

**UNIVERSIDAD AUTÓNOMA DE MADRID**

**ESCUELA POLITÉCNICA SUPERIOR**



**PROYECTO FIN DE CARRERA**

**Electrónica de control de un mini-robot para el  
posicionamiento micrométrico de una fibra óptica  
en el plano focal de un telescopio**

Nasib Fahim Fernández

Julio 2013



# Electrónica de control de un mini-robot para el posicionamiento micrométrico de una fibra óptica en el plano focal de un telescopio

AUTOR: Nasib Fahim Fernández  
TUTOR: Guillermo González de Rivera Peces

HCTLab  
Dpto. Tecnología Electrónica y de las Comunicaciones  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid

Julio 2013



*A mis padres  
y a Barcelona*



# Agradecimientos

*Agradece a la llama su luz, pero no olvides el pie del candil que paciente la sostiene.*

Rabindranath Tagore

A todas las personas del HCTLab, por prestarme su soporte informático y conocimiento cuando lo necesitaba, ayudarme en la resolución de todo tipo de problemas y prestarme su apoyo en la redacción y corrección de diversos artículos .

A Guillermo González de Rivera Peces, por todo el apoyo y ayuda prestados durante este último año, por todas las clases magistrales y consejos dados, por asesorarme y por la confianza depositada en mí durante este proyecto.

Al Instituto de Astrofísica de Andalucía (IAA/CSIC), por acogerme en Granada durante la realización de este proyecto. Especialmente a Justo Sánchez por llevar a cabo todos los test de funcionamiento con el robot y por ayudar de manera fundamental en la mejora del sistema.

Al Instituto de Física Teórica (IFT/CSIC), por el soporte a través del programa MICINNs Consolider-Ingenio 2010 bajo la subvención del proyecto MultiDark CSD2009-00064. Especialmente a Francisco Prada.

A mis compañeros de la Escuela Politécnica Superior (EPS/UAM), por estos años de penas y alegrías, especialmente a Fernando compañero de bibliotecas y fatigas. A todos los que por una razón u otra no pudieron acabar sus estudios.

A mis padres y abuelos, por toda una vida, por demostrarme que aunque flaqueen las fuerzas siempre se puede seguir hacia adelante. Por educarme en el respeto y el amor, por enseñarme que todo es posible, especialmente estos últimos años, por enseñarme a luchar y hacer de mí una persona crítica y curiosa.

A Olga, por estar siempre ahí cuando la necesitaba, por hacerme saber que cualquier problema es pequeño a su lado, por ayudarme y acompañarme durante todo este viaje.

A mi grupo scout Estrella Polar y a todos los bichos raros que lo componen, que me han dado fuerzas siempre para seguir adelante.

Y en definitiva, a todas y todos los que han hecho posible que llegue a este punto.





# Resumen

*¿Por qué ocultar ciertas verdades, hoy que son del dominio de la historia y pueden ser una enseñanza para el presente y para el porvenir?*

Errico Malatesta

Este PFC se enmarca en un proyecto de investigación a gran escala, dirigido por la división física del *DOE's Lawrence Berkeley National Laboratory* (Universidad de Berkeley). En él se pretende, en colaboración con multitud de entidades a escala global, llevar a cabo la creación del mayor mapa espectrográfico del universo. Poniendo a prueba teorías sobre la Materia y Energía Oscuras.

Para ello, se servirá del *NOAO 4-meter Mayall Telescope* (Kitt Peak, Arizona, EEUU), en el que se instalarán y desarrollarán diversos instrumentos de altas prestaciones.

El equipo español en este proyecto, en el que se encuentra el *Human Computer Technology Laboratory* (HCTLab) (Entorno en el que se desarrollará este PFC), se encargará del diseño del plano focal del telescopio. En éste se implementará una nube de 5000 mini-robots, encargados de posicionar el mismo número de fibras ópticas con el fin de captar la radiación procedente de diversos cuerpos celestes.

Este Proyecto de Fin de Carrera llevará a cabo la implementación y puesta a punto de los sistemas de control, tanto hardware como software de cada uno de los mini-robots. En el desarrollo de estos sistemas, se afrontará la problemática del control de motores paso a paso de gran precisión por medio de drivers de altas prestaciones, acotados por estrictas condiciones de consumo y superficie de diseño. Así como el desarrollo de una interfaz de usuario orientada a poder probar en tiempo real cada uno de los sistemas.

- **Palabras clave:** Motor paso a paso, sistemas de control, interface de usuario, drivers de motor



# Abstract

*Why to hide certain truths, today they are the domain of history and can be a lesson for the present and for the future?*

Errico Malatesta

This project is part of a worldwide research project, sponsored by the physical division of *DOE's Lawrence Berkeley National Laboratory (Berkeley Lab)*. In it, has been intended, in collaboration with many global organizations, carry out the creation of the largest spectroscopic map of the universe. Testing theories about the Dark Energy.

To do this, it will use the *NOAO 4-meter Mayall Telescope (Kitt Peak, Arizona, USA)*, where will install and develop different high-performance instruments.

The Spanish team in this project, in which is the *Human Computer Technology Laboratory (HCTLab)* (environment in which it will be developed this PFC) will design the focal plane of the telescope. In this, it will implement an array of 5000 mini-robots responsible for positioning optical fibers, in order to focus the radiation from galactic bodies.

This project will conduct the implementation and the development of control systems, both hardware and software of each of the mini-robots. During the development of these systems, it will face the problem of controlling high precision stepper motors with high performance drivers, bounded by strict conditions of use and design surface. As well as the development of an user-oriented real-time interface to test each of the systems.

- **Key words:** Stepper motor, control system, user interface, motor drivers.



# Índice general

<b>Agradecimientos</b>	<b>I</b>
<b>Resumen</b>	<b>III</b>
<b>Abstract</b>	<b>V</b>
<b>I Capítulos</b>	<b>1</b>
<b>1. Introducción</b>	<b>3</b>
1.1. Motivación y antecedentes . . . . .	3
1.1.1. Proyecto BigBOSS . . . . .	5
1.1.2. Diseño del plano focal . . . . .	6
1.1.3. Colaboración española . . . . .	6
1.2. Objetivos . . . . .	7
1.3. Marco de trabajo . . . . .	8
1.4. Organización de la Memoria . . . . .	9
<b>2. Fundamentos y estado del arte</b>	<b>11</b>
2.1. Posicionadores micrométricos en astrofísica . . . . .	11
2.1.1. Proyecto Hectospec . . . . .	12
2.1.2. Proyecto MEFOS . . . . .	13
2.1.3. Proyecto WFMOS, posicionador Cobra . . . . .	14
2.1.4. Proyecto LAMOST . . . . .	15
2.1.5. Proyecto BigBOSS, posicionador coplanar . . . . .	16
2.2. Introducción al control de motores . . . . .	19

---

2.2.1.	Motores DC . . . . .	19
2.2.2.	Motores piezo-eléctricos . . . . .	20
2.2.3.	Motores paso a paso . . . . .	21
2.2.3.1.	Fundamentos . . . . .	21
2.2.3.2.	<i>Modulación de corriente</i> . . . . .	22
2.2.3.3.	Control . . . . .	22
2.2.3.4.	Conclusiones . . . . .	23
<b>3.</b>	<b>Diseños preliminares</b>	<b>25</b>
3.1.	Planteamiento del diseño . . . . .	25
3.2.	Descripción del sistema . . . . .	26
3.3.	Estudio de componentes . . . . .	27
3.3.1.	Microcontrolador . . . . .	28
3.3.2.	Driver de motor . . . . .	28
3.3.3.	Potenciómetros digitales . . . . .	28
3.3.4.	Sensores de temperatura . . . . .	29
3.4.	Presupuesto . . . . .	29
3.4.1.	Modelo v.1 de la electrónica . . . . .	30
3.4.2.	Modelo v.2 de la electrónica . . . . .	30
3.4.3.	Modelo v.3 de la electrónica (una unidad) . . . . .	31
3.4.4.	Modelo v.3 de la electrónica (5000 unidades) . . . . .	31
3.5.	Conclusiones . . . . .	31
<b>4.</b>	<b>Diseño hardware</b>	<b>33</b>
4.1.	Componentes seleccionados . . . . .	33
4.1.1.	Microcontrolador . . . . .	33
4.1.1.1.	Encapsulados . . . . .	34
4.1.1.2.	Alimentación y consumo . . . . .	35
4.1.2.	Drivers de motor . . . . .	35
4.1.2.1.	Encapsulados . . . . .	37

4.1.2.2.	Alimentación y consumo . . . . .	37
4.1.3.	Potenciómetros digitales . . . . .	38
4.1.3.1.	Encapsulados . . . . .	38
4.1.3.2.	Alimentación y consumo . . . . .	39
4.1.4.	Sensores de temperatura . . . . .	39
4.1.4.1.	Encapsulados . . . . .	41
4.1.4.2.	Alimentación y consumo . . . . .	41
4.1.5.	Placa de desarrollo . . . . .	42
4.2.	Funcionamiento y características . . . . .	43
4.2.1.	Protocolos de comunicación . . . . .	43
4.2.1.1.	Comunicación interna / protocolo SPI . . . . .	43
4.2.1.2.	Comunicación externa / protocolo $I^2C$ . . . . .	44
4.2.2.	Funcionamiento . . . . .	46
4.2.2.1.	Esquema general . . . . .	46
4.2.2.2.	Control de la modulación de corriente . . . . .	49
4.2.2.3.	Prevención de fallos y control de temperatura . . . . .	50
4.2.2.4.	Control de posición . . . . .	51
4.2.3.	Alimentación y consumo total teórico . . . . .	52
4.3.	Fases del diseño . . . . .	53
4.3.1.	Modelo V.1 . . . . .	53
4.3.1.1.	Programación y comunicación . . . . .	58
4.3.2.	Modelo V.2 . . . . .	59
4.3.2.1.	Programación y comunicación . . . . .	62
4.3.3.	Modelo V.3 . . . . .	63
4.4.	Conclusiones . . . . .	64
<b>5.</b>	<b>Diseño software</b>	<b>65</b>
5.1.	Descripción del sistema de control software . . . . .	65
5.2.	Especificaciones y modos de funcionamiento . . . . .	66
5.3.	Software maestro, interface de usuario . . . . .	67

---

5.3.1.	Funcionamiento general . . . . .	68
5.3.1.1.	Comandos empleados . . . . .	70
5.3.2.	Control xy . . . . .	71
5.3.2.1.	Envío unitario de posición . . . . .	72
5.3.2.2.	Modo automático . . . . .	73
5.3.2.3.	Envío a cero . . . . .	77
5.3.2.4.	Control del modo de chopero . . . . .	77
5.3.3.	Control R1R2 . . . . .	78
5.3.3.1.	Envío unitario de posición . . . . .	79
5.3.3.2.	Envío en modo <i>cluster</i> . . . . .	80
5.3.3.3.	Configuración de los límites del sistema . . . . .	81
5.3.4.	Settings . . . . .	82
5.3.4.1.	Control de temperatura . . . . .	83
5.3.4.2.	Algoritmo de reducción de histéresis . . . . .	83
5.3.4.3.	Limitación de corriente entregada . . . . .	84
5.3.5.	Errores y alertas de ejecución . . . . .	85
5.4.	Software esclavo, código embarcado . . . . .	86
5.5.	Conclusiones . . . . .	88
<b>6.</b>	<b>Integración, pruebas y resultados</b>	<b>89</b>
6.1.	Laboratorios y centros de prueba . . . . .	89
6.2.	Pruebas realizadas en el IAA . . . . .	89
6.2.1.	Pruebas de repetitividad . . . . .	90
6.2.2.	Pruebas de histéresis . . . . .	91
6.3.	Pruebas realizadas en el LBNL . . . . .	93
6.3.1.	Sistema de cámara CCD . . . . .	94
6.3.2.	Sistema de reflexión CMOS . . . . .	94
6.3.3.	Pruebas de calibración . . . . .	95
6.3.4.	Pruebas de precisión (x,y) . . . . .	96
6.3.5.	Pruebas de variación en z . . . . .	97



---

6.3.6. Pruebas de alineamiento con el eje óptico ( <i>tip/tilt</i> ) . . . . .	98
6.4. Pruebas independientes . . . . .	99
6.5. Conclusiones . . . . .	100
<b>7. Conclusiones y trabajo futuro</b>	<b>101</b>
7.1. Conclusiones . . . . .	101
7.2. Trabajo futuro . . . . .	102
<b>Bibliografía</b>	<b>106</b>
<b>II Apéndices</b>	<b>109</b>
<b>A. Esquemáticos diseñados</b>	<b>111</b>
<b>B. Esquemáticos LPCXpresso</b>	<b>119</b>
<b>C. Algoritmos de transformación</b>	<b>127</b>
<b>D. Tráferencia tecnológica</b>	<b>131</b>
<b>E. Publicación</b>	<b>135</b>
<b>F. Presupuesto</b>	<b>145</b>
<b>G. Pliego de condiciones</b>	<b>149</b>
G.1. Pliego de condiciones . . . . .	151
G.1.1. Condiciones generales . . . . .	151
G.1.2. Condiciones particulares . . . . .	153



# Índice de figuras

1.1. Imágenes obtenidas por el proyecto SDSS-III. . . . .	4
1.2. Mayal Telescope. . . . .	5
1.3. Colocación de los posicionadores en el plano focal. . . . .	7
1.4. Sección vertical de la óptica del telescopio y detalle del plano focal. . . . .	8
2.1. Clasificación de posicionadores. . . . .	12
2.2. Posicionador del proyecto Hectospec. . . . .	13
2.3. Posicionador del proyecto MEFOS. . . . .	14
2.4. Detalle del posicionador empleado en el proyecto LAMOST. . . . .	16
2.5. Detalle frontal. . . . .	17
2.6. Detalle del posicionador coplanar. . . . .	17
2.7. Safe Operate Area (SOA). . . . .	18
2.8. Generación de rotación en motor DC. . . . .	20
2.9. Esquema de funcionamiento de un motor bipolar. . . . .	21
2.10. Circuito en puente H. . . . .	22
4.1. Encapsulados del microcontrolador LPC11U14 (mm). . . . .	34
4.2. Proceso interno de control de la modulación de corriente. . . . .	36
4.3. Encapsulados del driver de motor DRV8834. . . . .	37
4.4. Funcionamiento interno del potenciómetro AD5160. . . . .	38
4.5. SOT-23 3x3 mm. . . . .	39
4.6. Comunicación con los sensores de temperatura TC72. . . . .	40
4.7. Encapsulados del sensor de temperatura TC72. . . . .	41
4.8. Placa de prototipado LPCXpresso de NXP. . . . .	42

---

4.9. Interconexión de componentes en el protocolo SPI. . . . .	43
4.10. Interconexión de componentes en el protocolo $I^2C$ . . . . .	44
4.11. Diagrama general del control de la electrónica. . . . .	46
4.12. Señales implicadas en el movimiento de los motores (5 V/Div). . . . .	47
4.13. Flujo de control de $I_{chop}$ . . . . .	49
4.14. Flujo de alerta y detección de fallos. . . . .	50
4.15. Detalle del sistema de codificación de posición. . . . .	52
4.16. Diagrama de alimentación para el modelo v.1. . . . .	54
4.17. Modelo v.1 de la electrónica. . . . .	55
4.18. Terminales USB disponibles en la placa de prototipado LPCXpresso. . . . .	58
4.19. Diagrama de alimentación para el modelo v.2. . . . .	59
4.20. Modelo v.2 de la electrónica. . . . .	60
4.21. Etapa de potencia del modelo v.2 de la electrónica. . . . .	61
4.22. Conexión de programación para el modelo v.2 de la electrónica . . . . .	63
4.23. Imagen virtual de la versión final de la electrónica. . . . .	64
5.1. Diagrama de flujo del funcionamiento general del sistema. . . . .	68
5.2. Paquete de datos empleado. . . . .	70
5.3. Pestaña para el control en eje cartesiano. . . . .	71
5.4. Detalles de la sub-pestaña <i>Individual Mode Position</i> . . . . .	72
5.5. Ventana de monitorización tras el envío a x=1 mm e y=1 mm. . . . .	72
5.6. Detalle de las sub-pestañas implicadas en el modo automático. . . . .	73
5.7. Diagrama de flujo del funcionamiento modo automático. . . . .	75
5.8. Tres posibles advertencias mostradas tras la calibración. . . . .	76
5.9. Detalle de la sub-pestaña de envío a cero. . . . .	77
5.10. Detalle de la sub-pestaña de control del modo de modulación de corriente . . . . .	77
5.11. Pestaña para el control en modo R1R2. . . . .	78
5.12. Detalle de la sub-pestaña de envío unitario R1R2. . . . .	79
5.13. Ventana de monitorización tras el movimiento de R1. . . . .	79

5.14. Detalle de la sub-pestaña de envío en modo cluster. . . . .	80
5.15. Ventana de monitorización en un envío en modo cluster. . . . .	81
5.16. Detalle de la sub-pestaña de configuración de los límites del sistema. . .	81
5.17. Pestaña para el control de parámetros generales. . . . .	82
5.18. Detalle de la sub-pestaña de monitorización térmica. . . . .	83
5.19. Detalle de la sub-pestaña de selección del algoritmo de <i>histéresis</i> . . .	83
5.20. Detalle de la sub-pestaña de selección de la corriente entregada. . . .	84
5.21. Ventana de monitorización en la configuración de corriente. . . . .	84
5.22. Errores más comunes en la ejecución del software de control . . . . .	85
5.23. Diagrama de flujo del software esclavo. . . . .	87
6.1. Sistema de prueba en el IAA. . . . .	90
6.2. Extracto de un estudio sobre repetitividad (Justo Sánchez IAA/CSIC). . .	91
6.3. Experimento de histéresis (Justo Sánchez IAA/CSIC). . . . .	92
6.4. Mejora de los resultados tras una única iteración del algoritmo de reducción de histéresis (Justo Sánchez IAA/CSIC). . . . .	92
6.5. Otros test de histéresis(Justo Sánchez IAA/CSIC). . . . .	93
6.6. Sistema CCD de calibración y prueba de precisión. . . . .	94
6.7. Sistema CMOS para el estudio de la desviación vertical. . . . .	94
6.8. Sistema Keyence de emisión-recepción láser. . . . .	95
6.9. Resultados del proceso de calibración. . . . .	96
6.10. Errores calculados, donde N es el número de posicionados. . . . .	96
6.11. Test realizado con nueve correcciones sucesivas . . . . .	97
6.12. Variación de la componente z en función de la temperatura. . . . .	98
6.13. Estudio de la desviación respecto de la normal teórica. . . . .	98
6.14. Gráfica comparativa entre el movimiento teórico y los datos reales enviados al actuador desde el software de control. . . . .	99
A.1. Esquema del modelo v.1 (1ª mitad). . . . .	113
A.2. Esquema del modelo v.1 (2ª mitad). . . . .	114
A.3. Esquema del modelo v.2 (1ª hoja, 1ª mitad) . . . . .	115

---

A.4. Esquema del modelo v.2 (1ª hoja, 2ª mitad) . . . . .	116
A.5. Esquema del modelo v.2 (2ª hoja, 1ª mitad) . . . . .	117
A.6. Esquema del modelo v.2 (2ª hoja, 2ª mitad) . . . . .	118
B.1. Esquema general. . . . .	121
B.2. Etapa de potencia. . . . .	122
B.3. Interface USB. . . . .	123
B.4. Conexión entre placas. . . . .	124
B.5. Conectores de expansión. . . . .	125
B.6. Microcontrolador. . . . .	126

# Índice de cuadros

2.1. Características del sistema Hectospec. . . . .	12
2.2. Características del sistema MEFOS. . . . .	13
2.3. Características del sistema Cobra. . . . .	14
2.4. Características del proyecto LAMOST. . . . .	15
2.5. Características de los posicionadores coplanares. . . . .	16
3.1. Microcontroladores candidatos a formar parte de la electrónica. . . . .	28
3.2. Potenciómetros candidatos a formar parte de la electrónica. . . . .	28
3.3. Sensores de temperatura candidatos a formar parte de la electrónica. . . . .	29
3.4. Presupuesto para el modelo v.1 de la electrónica . . . . .	30
3.5. Presupuesto para el modelo v.2 de la electrónica . . . . .	30
3.6. Presupuesto para el modelo v.3 de la electrónica . . . . .	31
3.7. Presupuesto para el sistema completo de control . . . . .	31
4.1. Consumo medio ejecutando un <i>While(1)</i> desde la flash a 25 °C y 3.3 V. . . . .	35
4.2. Señales de configuración para el driver DRV8834. . . . .	36
4.3. Consumos medios del driver DRV8834. . . . .	37
4.4. Consumo medio del potenciómetro AD5160. . . . .	39
4.5. Registros internos del sensor de temperatura TC72. . . . .	40
4.6. Configuración de modos de funcionamiento del sensor TC72. . . . .	41
4.7. Consumos y alimentación del sensor TC72. . . . .	41
4.8. Consumo individual de los componentes. . . . .	52
4.9. Consumo global de la electrónica. . . . .	53
4.10. Consumo global del posicionador en mW. . . . .	53

---

F.1. Presupuesto global del proyecto desarrollado . . . . . 147



**Parte I**  
**Capítulos**



# Capítulo 1

## Introducción

*Si consigo ver más lejos es porque he conseguido  
auparme a hombros de gigantes.*

Isaac Newton

**Resumen:** en este capítulo se presentan las motivaciones y el contexto del proyecto. Se exponen los objetivos principales y se describe en detalle la organización y estructuración de la Memoria. Por otro lado, se describe el marco de trabajo en el que se encuentra inmerso el proyecto: el grupo de investigación HCTLab (Human Computer Technology Laboratory).

### 1.1. Motivación y antecedentes

En la Noche de los Tiempos, el hombre alzó la vista intentando dar respuesta a los innumerables misterios que le rodeaban.

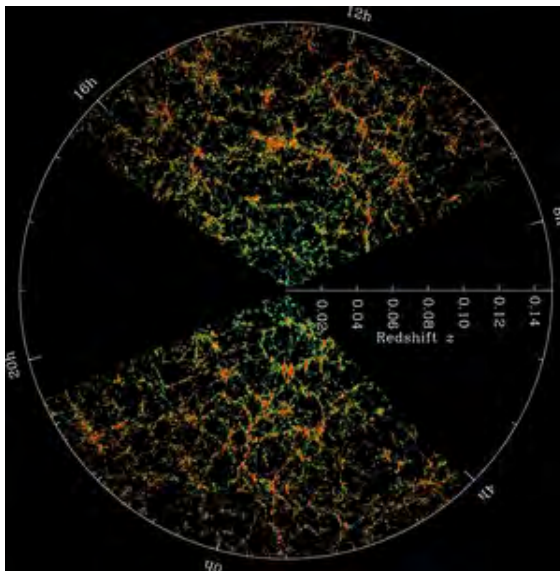
Tras incontables intentos por cartografiar el universo observable, la comunidad científica elaboró complejos mapas, detallando las propiedades de los cuerpos que podían observar en el firmamento.

Con el avance de la ciencia, los logros alcanzados en este campo han sido notable y constantemente mejorados. Surgiendo, durante los últimos años, multitud de proyectos cuyo objetivo principal no ha sido otro que dar respuesta a tantas cuestiones sin resolver. Entre las que destacan la creación del universo, su expansión y el estudio y comprobación de teorías entorno a la Energía y Materia Oscuras.

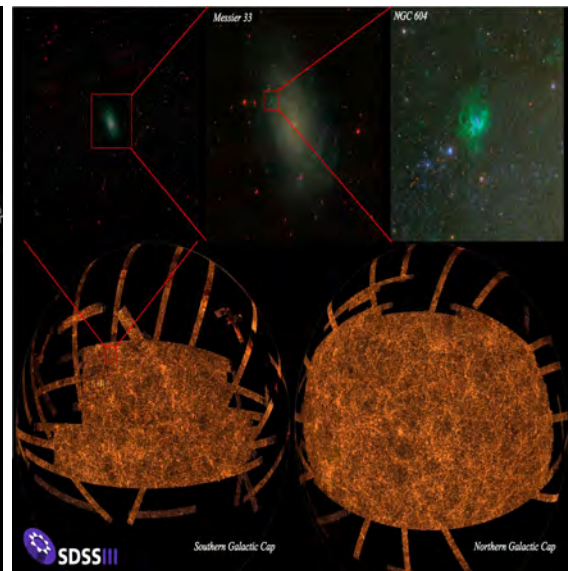
Uno de los proyectos más destacados en este campo es el SDSS-III [1],[2] (*Sloan Digital Sky Survey III*) (Berkeley Lab.). Los principales objetivos de éste son, entre otros, el mapeo exhaustivo de la Vía Láctea, la búsqueda de planetas extrasolares y el estudio de la Energía Oscura.

Sirviéndose del *Sloan Foundation 2.5-m Telescope* en el observatorio Apache (Nuevo México), este proyecto extenderá su actividad hasta finales de 2014, con cuatro líneas principales de investigación:

- **SEGUE**[3] (*Sloan Extension for Galactic Understanding and Exploration*) (2008-2009): recopila el espectro de más de 230000 estrellas cercanas con el objetivo de estudiar la estructura interna de la Vía Láctea.
- **APOGEE**[4] (*Apache Point Observatory Galactic Evolution Experiment*) (2011-2014): estudia el interior de la galaxia y sigue Gigantes Rojas mediante un espectrógrafo de infrarrojos.
- **MARVELS**[5] (*Multi-object APO Radial Velocity Exoplanet Large-area Survey*) (2008-2014): estudia las características de planetas extrasolares.
- **BOSS**[6] (*Baryon Oscillation Spectroscopic Survey*) (2009-2014): estudia la distribución espacial de las galaxias rojas (LRGs) y quásares, para detectar las características impresas por las oscilaciones acústicas de Baryon en el universo temprano.



(a) Distribución de galaxias.



(b) Mapeo del universo observable.



(c) Supernovas.



(d) Colisión de galaxias.

Figura 1.1: Imágenes obtenidas por el proyecto SDSS-III.

### 1.1.1. Proyecto BigBOSS

Tras más de 10 años de investigaciones y planteamiento de proyectos, la Universidad de Berkeley junto a multitud de universidades y organismos tanto públicos como privados a lo largo del planeta, han puesto en marcha la creación del mayor mapa espectrográfico del universo observable bajo el nombre de «*BigBOSS*»[7][8][9]. Éste, apoyándose en los avances realizados por el SDSS-III en su línea de investigación BOSS, toma como objetivo 50 millones de cuerpos y busca la localización más precisa hasta la fecha de, al menos, 20 millones de galaxias y quásares, para realizar un mapa con un volumen de objetivos 10 veces mayor al mejor mapa conseguido hasta la fecha. Ofreciendo así herramientas y datos de extrema precisión que puedan formar las bases de futuras investigaciones a científicos y público en general.

Este ambicioso proyecto se enmarca en el conjunto de investigaciones de la división física del *DOE's Lawrence Berkeley National Laboratory*, Universidad de Berkeley. Y responde a una evolución de la misión espacial *Joint Dark Energy* con el objetivo de estudiar desde la Tierra el efecto de *Corrimiento al Rojo* (*redshift*) en cientos de cuerpos, reduciendo los costes de una misión espacial. Este efecto es la constatación del decaimiento frecuencial hacia el rojo. Con él, haciendo una analogía con el efecto *Doppler* en las ondas sonoras, es posible conocer con precisión el desplazamiento de objetos en el espacio, comparando la radiación reflejada por éstos con la procedente de la fuente emisora.

Para alcanzar los objetivos descritos, BigBOSS se servirá del *NOAO 4-meter Mayall Telescope* (*Kitt Peak, Arizona, EEUU*) (figura 1.2) sobre el que se montará un espectrógrafo que podrá estudiar en profundidad las propiedades del universo observable.

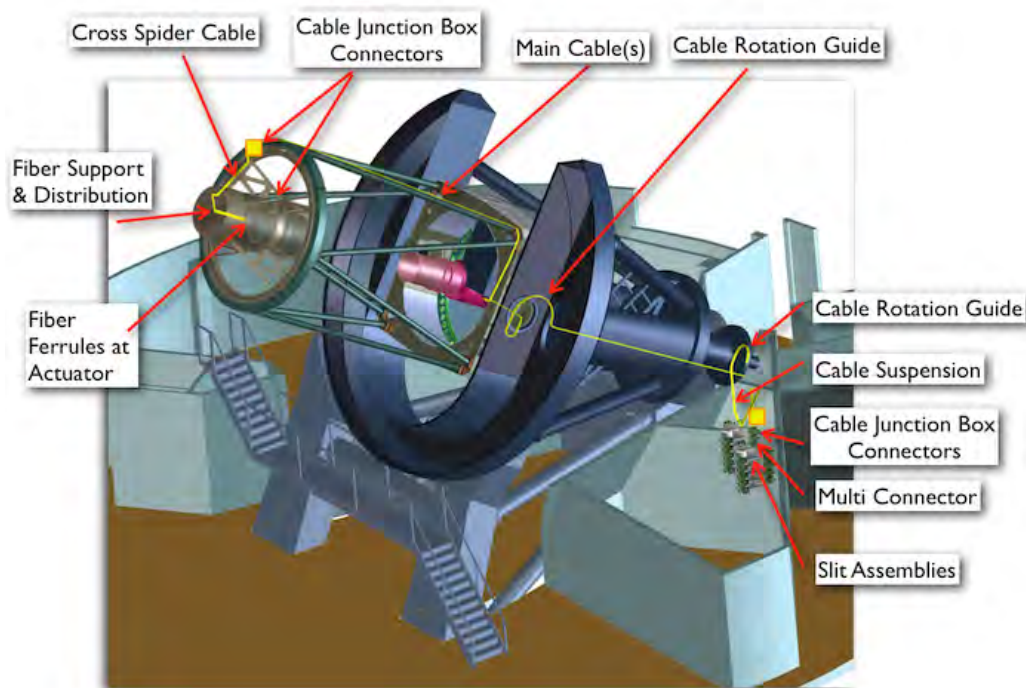


Figura 1.2: Mayal Telescope.

El diseño de la interfaz encargada de focalizar la radiación captada por el telescopio NOAO ha sido encargado a diferentes organismos a nivel mundial, entre ellos al Instituto de Física Teórica (Campus de Excelencia UAM/CSIC) y al Instituto Astrofísico de Andalucía (IAA/CSIC). Estos grupos han confiado el diseño técnico al grupo de investigación HCTLab de la Escuela Politécnica Superior (EPS/UAM). Y en este contexto, se encuentra enmarcado el Proyecto de Fin de Carrera que nos atañe.

### 1.1.2. Diseño del plano focal

Con el objetivo de captar la radiación emitida por los diferentes cuerpos celestes, focalizando la exposición en puntos discretos del plano focal del telescopio (estrellas, cúmulos, galaxias...), el proyecto BigBOSS se servirá de 5000 fibras ópticas. Éstas serán guiadas al punto exacto de exposición luminosa por medio de robots diseñados para tal fin (sección 2.1.5), dispuestos perpendicularmente al plano focal. La nube de robots se instalará a través de una malla metálica de 1 metro de diámetro que recorrerá la superficie focal del telescopio y captará una imagen cada 20 minutos.

El diseño de la nube de robots, así como de la electrónica y sistemas necesarios para controlarlos y conectar cada una de estas 5000 estructuras con el ordenador central, ha sido encargado paralelamente a organismos españoles, chinos y americanos, con el objetivo de optimizar los resultados. Y ésta, en Españ, es la labor principal del Instituto de Física Teórica (IFT) y el *Human Computer Technology Laboratory* (HCTLab), perteneciente a la Universidad Autónoma de Madrid (UAM).

### 1.1.3. Colaboración española

Los robots encargados del posicionamiento de la fibra en el plano focal basan su funcionamiento en el concepto  $\theta$ - $\theta$ , o posicionamiento angular de la cabeza de fibra, por medio de la rotación coplanar, perpendicular a la dirección de observación, de dos pequeños motores. Este posicionamiento será ordenado desde una unidad central que almacenará las posiciones relativas de los diferentes cuerpos a estudiar y generará un patrón de movimiento para cada uno de los robots. Durante la generación de este patrón se ejecutarán diversos algoritmos<sup>1</sup> de prevención de colisiones entre sistemas, dado que debido a la arquitectura del plano focal (figura 1.3) cada robot comparte cierta superficie de posicionado con los sistemas vecinos (con el objetivo de optimizar el espacio y mejorar el tiempo de colocación de la fibra) y con esto existe la posibilidad de colisión entre sistemas, hecho que desencadenaría una rotura de los mismos.

Tras la generación del patrón y la comprobación de la inexistencia de posibles colisiones entre sistemas, la unidad central dará la orden de acción, comunicando a cada robot la posición exacta en la que debe situar la fibra dentro de su rango de actuación. Después de unos segundos de posicionamiento, el sistema retroiluminará cada una de las 5000 fibras y comprobará la correcta posición de éstas por

---

<sup>1</sup>Estos algoritmos están siendo desarrollados en la actualidad por entidades suizas especializadas en el control de tráfico terrestre y aéreo.

medio del análisis de imagen realizado por una cámara CCD colocada en la parte frontal del telescopio, que comprobará la posición relativa de cada fibra respecto a unos patrones estáticos de posición.

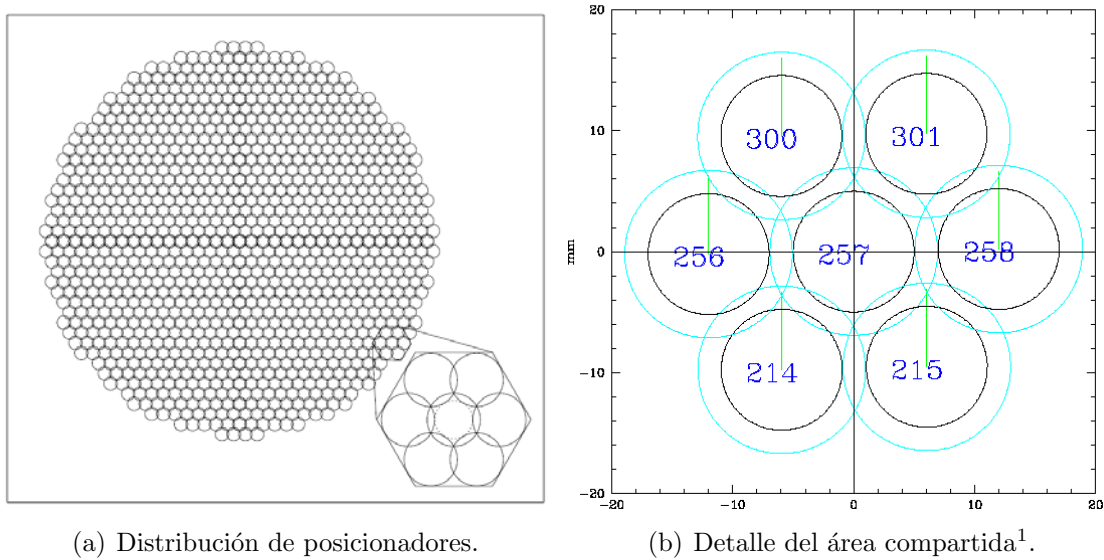


Figura 1.3: Colocación de los posicionadores en el plano focal.

Realizado el posicionamiento de las fibras en el plano focal del instrumento, dará comienzo una exposición aproximada de 20 minutos del campo observado. Tras esto, se ejecutará un nuevo reposicionamiento de las fibras, focalizando la nueva exposición en otros 5000 cuerpos.

El objetivo planteado al equipo español es el de conseguir el posicionamiento total de las 5000 fibras en menos de 60 segundos. Esto supone un gran reto, dado que los menores tiempos conseguidos por equipos extranjeros hasta la fecha son del orden de minutos con un número mucho menor de fibras. Todo esto, enmarcado en niveles de consumo de corriente eléctrica muy reducidos, dado el alto número de robots existente.

## 1.2. Objetivos

El objetivo pues, de este proyecto, será el planteamiento, diseño y prueba de un sistema electrónico funcional de control para el manejo de cada uno de los robots de esta nube de 5000 unidades (sección 4). Estos sistemas tendrán como principal objetivo la recepción de los datos procedentes de los nodos de comunicación superiores, el procesamiento de éstos y el control de cada uno de los dos mini-motores paso a paso instalados en el robot. A su vez, éstos monitorearán su temperatura interna y alertarán de posibles errores en el sistema con el objetivo de prevenir fallos fatales del mismo.

<sup>1</sup>Debido a la curvatura del plano focal, esta área compartida depende de la posición de los robots en el mismo.

Para cumplir lo anteriormente descrito, se partirá en un diseño de bajo consumo compuesto por elementos de reducidas dimensiones integrables en un circuito impreso que no exceda las dimensiones requeridas por el proyecto.

El sistema creado en este proyecto no tendrá que cumplir las restricciones en cuanto a tamaño se refiere, y servirá como plataforma experimental de pruebas para el diseño final. Éste ha de poder ser controlado por un ordenador personal, desde el que se darán las órdenes básicas para el posicionamiento de cada motor. Por lo que se diseñará, a su vez, una interface de usuario desde la que se podrá controlar todo lo anteriormente descrito (sección 5).

Una vez comprobado el buen funcionamiento y el cumplimiento de los requisitos en cuanto a consumo, tiempo de respuesta y protección frente a posibles fallos se refiere, el sistema será implementado en tamaño muy reducido, pudiendo instalarse en el lateral de cada uno de los 5000 mini-robots. Debido a esto, todos los elementos que se han seleccionado para este diseño pueden conseguirse en varios tamaños y tipos de encapsulados, con el fin de poder seguir fielmente el esquema hardware planteado desde el principio.

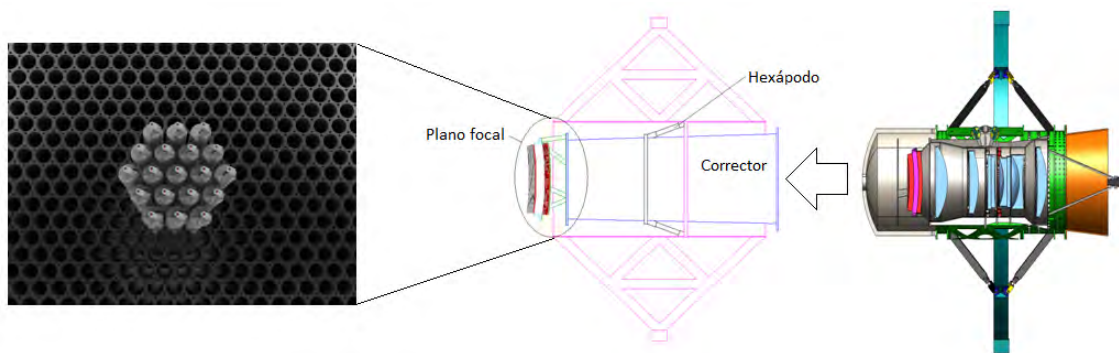


Figura 1.4: Sección vertical de la óptica del telescopio y detalle del plano focal.

### 1.3. Marco de trabajo

El entorno en el que se desarrollará el Proyecto será Grupo de Investigación HCTLab (*Human Computer Technology Laboratory*), adscrito al departamento de *Tecnología Electrónica y de las Comunicaciones* de la Escuela Politécnica Superior (EPS) en la Universidad Autónoma de Madrid (UAM).

Este laboratorio fue creado en Octubre de 2002 y centra sus líneas de investigación en ámbitos multidisciplinares, en torno a tecnologías que relacionan al usuario humano con las máquinas o, más concretamente, con los sistemas basados en procesador.

Este grupo de investigación está formado por varios profesores universitarios que desarrollan su actividad de I+D con preferencia en áreas muy diversas, yendo éstas desde el diseño de aplicaciones hardware y software basadas en sistemas embebidos (microcontroladores, FPGAs, etc.) hasta el desarrollo de tecnologías del habla y la accesibilidad.



## 1.4. Organización de la Memoria

Esta memoria recoge las etapas y procesos llevados a cabo en el desarrollo de la electrónica de los actuadores y el software de control de estos. Ésta se dividirá en dos partes principales independientes:

- **Diseño de la electrónica:** en esta parte se describirán las etapas y procesos de diseño de los actuadores, los diferentes prototipos desarrollados hasta conseguir el diseño definitivo y los componentes utilizados. Por otro lado, se describirá el funcionamiento físico del sistema y se expondrán previsiones teóricas de comportamiento y consumo.
- **Diseño del software:** en esta parte se expondrá tanto el software de control en *modo ingeniero* del sistema como el código embarcado en el microcontrolador instalado en el hardware. Se desarrollará el funcionamiento de éste y se explicarán en detalle los algoritmos usados para el control a bajo nivel del robot. A su vez, se explicará el protocolo utilizado para comunicarse desde el software de control con el actuador.

Por otro lado, en esta memoria se presentarán los resultados detallados tras la realización de los diferentes test de comportamiento del sistema, en cuanto a consumo, a disipación térmica, a tiempos y a precisión se refiere. Se expondrán los proyectos similares que precedieron a éste y se hablará de las posibles mejoras futuras en el sistema, las aplicaciones de éste y los futuros desarrollos.



# Capítulo 2

## Fundamentos y estado del arte

*Si has construido castillos en el aire, tu trabajo no se pierde;  
ahora coloca las bases debajo de ellos.*

Henry David Thoreau

**Resumen:** en este capítulo se presentan los diseños que precedieron al estado actual de los posicionadores, se desarrolla en detalle el posicionador coplanar y se introducen las bases del control de motores.

### 2.1. Posicionadores micrométricos en astrofísica

Durante los últimos años, en el campo de la astrofísica, se han desarrollado multitud de proyectos basados en espectrógrafos [10][11][12][13][14], cuya característica principal ha sido el posicionamiento de fibras ópticas en el plano focal de telescopios con el fin de focalizar la radiación extra-planetaria.

El objetivo principal de estos proyectos ha sido transformar telescopios con resoluciones moderadamente aceptables en sistemas capaces de analizar simultáneamente millares de objetos y estudiar sus propiedades en profundidad [14][15][16].

En una primera etapa, estos sistemas basaron su funcionamiento en el posicionamiento estático de las fibras por medio de anclajes fijos. Más adelante se desarrollaron prototipos *Pick and Place* basados en el *enganche y suelte* de las fibras dentro de una superficie pre-establecida (pegamentos o taladros), mediante procesos manuales o robotizados. En los desarrollos más modernos se están utilizando posicionadores independientes cuyos principios, grados de libertad, coste y funcionalidad varían significativamente de unos a otros.

En esta sección se expondrán los ejemplos más significativos que preceden al posicionador coplanar  $\theta$ - $\theta$  propuesto por el equipo español para el proyecto BigBOSS.

En la figura 2.1 puede observarse la clasificación de posicionadores realizada por Greg Smit (et al.) [17] y los diferentes proyectos en los que han sido integrados.

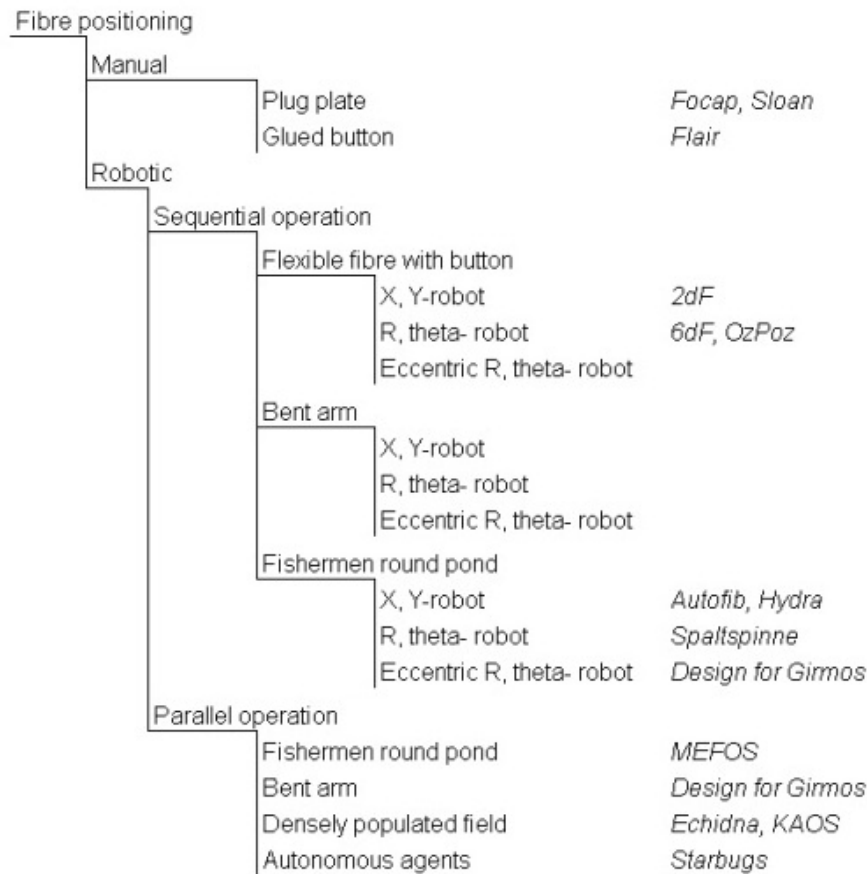


Figura 2.1: Clasificación de posicionadores.

### 2.1.1. Proyecto Hectospec

El proyecto Hectospec puede ser considerado uno de los exponentes del método *Pick and Place* robotizado. Al igual que en BigBoss, la herramienta principal en el proyecto Hectospec [18] es un espectrógrafo de resolución moderada basado en la recepción de radiación mediante fibras ópticas. Este instrumento se sirve del método *pick-and-place* para posicionar un total de 300 cabezales de fibra en el plano focal del telescopio *NMT* (La Silla, Chile).

Modelo	Número de fibras	Precisión	Tiempo de posicionado
Pick and Place	300	25 $\mu\text{m}$	300 s

Cuadro 2.1: Características del sistema Hectospec.

El elemento principal de este espectrógrafo es un posicionador robótico (figura 2.2) con seis grados de libertad: tres ejes lineales ( $X, Y, Z$ ), dos ejes de rotación ( $\Theta$  y  $\phi$ ); que causan a su vez movimiento en los ejes  $X$  e  $Y$  y un último sistema encargado de agarrar las fibras. Todo esto proveerá al posicionador de una precisión máxima de 25  $\mu\text{m}$  y hará posible la colocación de las 300 fibras en 300 s.

El método *Pick and Place* es un método que se ha utilizado comúnmente en este tipo de espectrógrafos debido a la precisión que puede alcanzar. Esta técnica consiste en

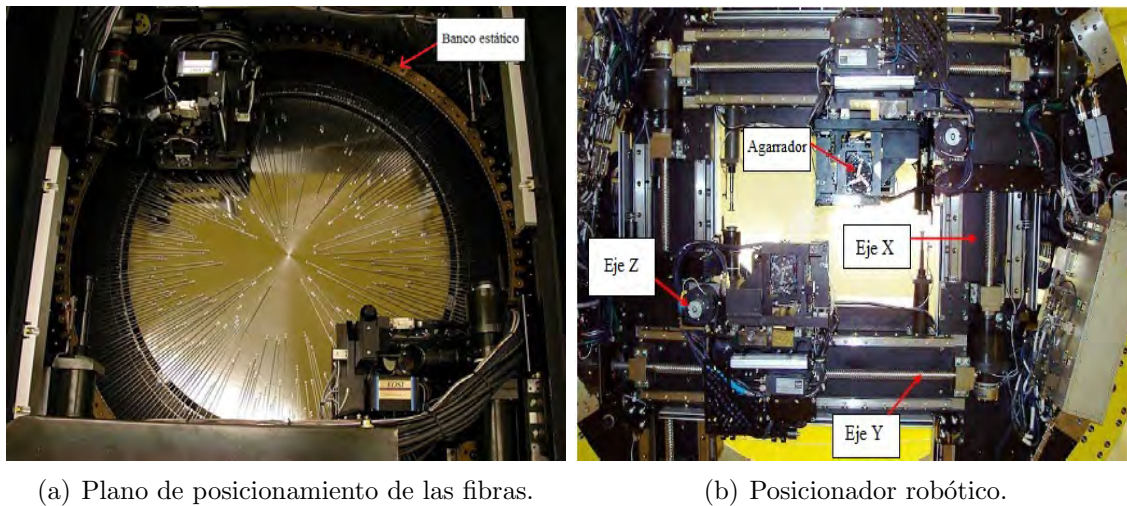


Figura 2.2: Posicionador del proyecto Hectospec.

agarrar (*pick*) la fibra de un banco estático y posicionarla (*place*) en el plano focal mediante el enganche de ésta en la superficie establecida para tal caso.

Esta técnica muestra innumerables desventajas frente al posicionador  $\theta$ - $\theta$  presentado por el equipo español para el proyecto BigBOSS. Por un lado, el hecho de llevar la fibra hasta el punto de exposición, dejando ésta extendida en el plano de posicionamiento, entorpece mucho el posicionado del resto de fibras e impide que se coloque ninguna en la línea que forma el punto de exposición de ésta, con su banco estático. Por otro lado, el posicionar las fibras de una en una retrasa mucho el proceso e implica un desgaste excesivo de éstas y una necesidad constante de mantenimiento del posicionador. No obstante, el hecho de contar con un único robot posicionador, independientemente de las características de éste, implica un coste del proyecto mucho menor al que se manejará en BigBOSS.

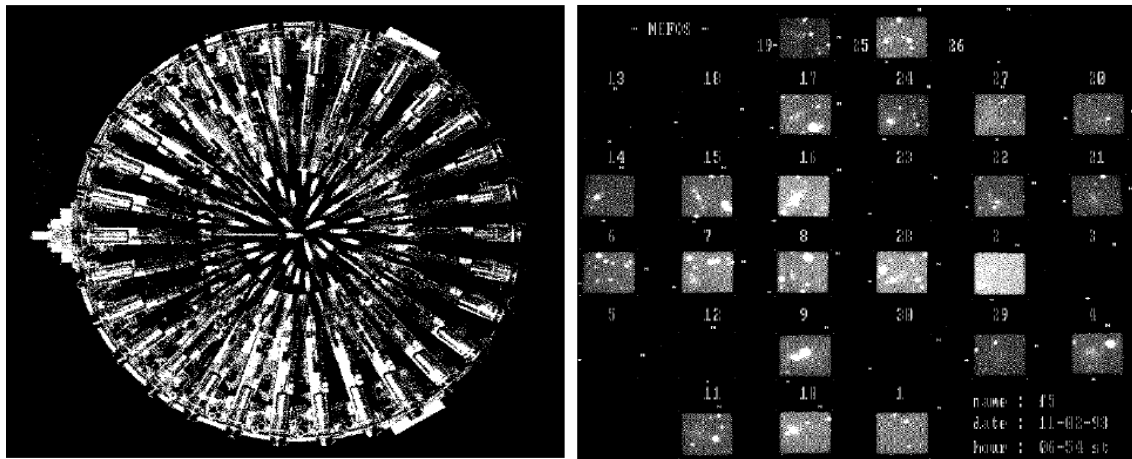
### 2.1.2. Proyecto MEFOS

Con el objetivo de disminuir el tiempo empleado en el posicionamiento secuencial de fibras en proyectos como el Hectospec, se diseñaron sistemas robotizados que pudiesen trabajar en paralelo (sacrificando resolución). Uno de los primeros proyectos que materializó este tipo de sistemas fue el MEFOS [19] (*Meudon-ESO Fibre Optical System*), que utilizó el método *Fisherman-round-the-pond* para posicionar 30 fibras en el plano focal del *3.6 m ESO telescope* (La Silla, Chile).

Modelo	Número de fibras	Precisión	Tiempo de posicionado
Fisherman-round-the-pond	30	? $\mu\text{m}$	4 min

Cuadro 2.2: Características del sistema MEFOS.

El método *Fisherman-round-the-pond* se sirve de un conjunto de brazos, instalados alrededor de la superficie de posicionado, cuyo único grado de libertad les permite rotar un determinado número de grados en su base. Éstos mantienen fijo un cabezal de fibra en su extremo, realizando la exposición cuando permanecen quietos.



(a) Posicionador.

(b) Resultado tras 5 minutos de exposición.

Figura 2.3: Posicionador del proyecto MEFOS.

Este sistema consta de 30 brazos posicionadores (uno usado como guía y 29 con propósitos astronómicos), en cuyo extremo se sitúan los cabezales de dos fibras. Una de estas fibras es usada para la calibración y la otra para la observación del cielo. Cada uno de estos brazos es capaz de cubrir un triángulo de  $15^\circ$  cuya base se encuentra en el centro del plano focal. Por otro lado, cada uno de ellos posee un sistema esclavo de control, cuyo manejo es centralizado en una unidad independiente al control del telescopio. Esto no supone un problema, en cuanto a coste se refiere, dado el reducido número de posicionadores. No obstante, introduce una mejora notable en tiempo de posicionado, respecto a proyectos puestos en marcha en la época (Hectospec, ARGUS, etc.) debido a la capacidad de sus 30 brazos de trabajar en paralelo.

Finalmente, este posicionador utiliza motores DC (cuyo funcionamiento se presenta en la sección 2.2.1) dado que durante el diseño se estimó que el calentamiento de los motores paso a paso (diseñados en el momento) introduciría errores fuera del margen admisible.

### 2.1.3. Proyecto WFMOS, posicionador Cobra

Similar a los proyectos expuestos hasta el momento, WFMOS (*Wide-Field Multi Object Spectrometer*), centra sus estudios en el mapeo del universo y la Componente Oscura. Para ello, este proyecto cuenta con un espectrógrafo montado en el espejo secundario del telescopio terrestre *Mauna Kea* (Hawaii, EEUU).

Modelo	Número de fibras	Precisión	Tiempo de posicionado
$\Theta$ - $\Phi$	2400	$5 \mu\text{m}$	? s

Cuadro 2.3: Características del sistema Cobra.

Al igual que en los proyectos anteriores, esta herramienta se sirve de posicionadores de fibra óptica que focalizan la radiación procedente del cuerpo a observar. En este caso, el diseño de estos posicionadores y sus sistemas de control forman parte del sub-proyecto Cobra [20].

Cobra utiliza 2400 posicionadores de fibra independientes basados en el modelo  $\Theta\text{-}\Phi$  con dos grados de libertad, capaces de conseguir precisiones de hasta  $5\ \mu\text{m}$ , con un ángulo de observación de  $1.5^\circ$  por fibra. Cada uno de ellos es capaz de abarcar una región de 9.5 mm de diámetro. Para ello cuentan con dos piezo-motores paso a paso (cuyas dimensiones los colocaban entre los motores más pequeños en el momento del planteamiento del proyecto), capaces de alcanzar precisiones de  $0.065^\circ$  por paso y 3 mNm de par. No obstante, para evitar colisiones con los posicionadores vecinos, aún siendo el diámetro máximo abarcable por cada actuador 9.5 mm, se limita éste a 7.7 mm consiguiendo igualmente dar cobertura al total del plano focal.

Estos actuadores no cuentan con *encoders* capaces de verificar la posición real de los 4800 motores, por lo que, al igual que en el proyecto BigBOSS, Cobra retroilumina las fibras tras cada movimiento y comprueba la situación de cada una por medio de la toma de imágenes con una cámara CCD y el posterior procesamiento de éstas con un software de control.

Por otro lado, este sistema utiliza motores piezo-eléctricos, que si bien cuentan con reducidas dimensiones, es necesaria su continua alimentación para evitar la pérdida de la posición. Este hándicap eleva considerablemente el consumo en cada actuador y niega la posibilidad del uso de una modulación de corriente que mantenga calientes los motores solo en los momentos necesarios como se expone en la sección 2.2.3.2.

#### 2.1.4. Proyecto LAMOST

Muy similar al proyecto liderado por la universidad de Berkeley, el proyecto chino LAMOST[21] (*Large sky area Muti-object fibre Spectroscopy Telescope*) dirigido por la Academia China de Ciencias, busca estudiar en profundidad las propiedades de la energía oscura, recopilando el espectro de cuerpos galácticos y extra-galácticos. Para ello se sirve de un telescopio instalado en el *Xinglong Station of national Astronomical Observatory*. Este telescopio consta de dos espejos reflectores masivos que focalizan la radiación captada hacia una superficie de exposición de 1.75 m, muy similar a la diseñada para el proyecto BigBOSS. Esta superficie está formada por 4000 mini-robots posicionadores de fibra (figura 2.4), que siguen el modelo  $\Theta\text{-}\Theta$  (tabla 2.4).

Modelo	Número de fibras	Precisión	Tiempo de posicionado
$\Theta\text{-}\Theta$	4000	$40\ \mu\text{m}$	600 s

Cuadro 2.4: Características del proyecto LAMOST.

Al igual que en el proyecto BigBOSS, LAMOST se sirve de posicionadores independientes con el objetivo de reducir el tiempo de posicionado y poder compensar los errores en tiempo de observación. Esto introduce el mismo problema que el aparecido en este proyecto, dado que al aumentar la densidad de posicionadores en el plano focal se introduce el riesgo de colisión entre unidades vecinas, en ciertas superficies de posicionado compartidas. Lo que aumenta el tiempo de reposicionado y la latencia total del sistema.

Por otro lado, LAMOST usa motores paso a paso similares a los incluidos por el equipo español en el proyecto BigBOSS. No obstante, la velocidad a los que éstos

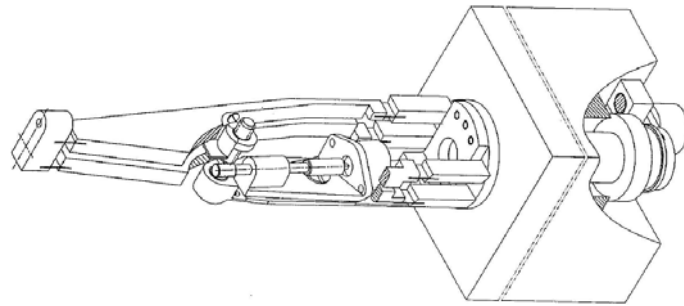


Figura 2.4: Detalle del posicionador empleado en el proyecto LAMOST.

operan (600-800 pasos/s) es mucho menor a la utilizada en este proyecto, lo que aumenta una vez más la latencia total del sistema, limitando las observaciones que pueden llevarse a cabo en un periodo acotado.

Finalmente, con el objetivo de reducir los costes de un proyecto de esta envergadura, LAMOST no dedica una electrónica de control individualizada para cada posicionador. Por lo que se han diseñado 63 unidades de control capaces de mover 64 posicionadores cada una. Esto impide tener un control preciso del estado de cada robot (alertando de posibles fallos fatales) y aumenta el tiempo de posicionado, dado que no se individualiza el movimiento de cada actuador.

### 2.1.5. Proyecto BigBOSS, posicionador coplanar

El sistema español presentado para el proyecto BigBOSS lo componen 5000 posicionadores robóticos independientes (figura 2.6). Cada uno de ellos posee dos mini-motores paso a paso que colocan cada fibra en su punto de exposición de forma coplanar, basándose en el modelo  $\Theta$ - $\Theta$ . Esto es: ambos motores hacen mover la fibra en el mismo plano, con dos grados de libertad, de forma angular.

Modelo	Número de fibras	Precisión	Tiempo de posicionado
$\Theta$ - $\Theta$	5000	5 $\mu\text{m}$	60 s

Cuadro 2.5: Características de los posicionadores coplanares.

El requerimiento del proyecto limita la colocación de las 5000 fibras a menos de 60 s. No obstante, este posicionador coloca su fibra en un máximo de 17 s, a los que hay que añadir el tiempo de re-posicionamiento requerido por el algoritmo de colocación para evitar colisiones con los sistemas vecinos.

Los motores utilizados en este posicionador son mini-motores paso a paso. Éstos poseen una clara ventaja frente a los utilizados en proyectos similares: el hecho de no ser motores piezo-eléctricos posibilita el uso de modulación de corriente. Este proceso consiste en alimentar los motores con señales pulsadas. Estas señales se generan haciendo subir la corriente hasta un máximo preconfigurado y llevándola a cero tras alcanzar este máximo, originando una señal PWM modulada a las necesidades de cada bobina presente en el motor. Este proceso se genera tanto en el movimiento de los motores como con el motor parado (pudiéndose generar solo en el primer caso),



con lo que garantizamos mantener el motor caliente en determinadas situaciones (condiciones climáticas adversas, etc.).

Por otro lado, cada uno de los motores puede generar un par máximo de 25 nNm, no obstante, el posicionador está diseñado para soportar un par total máximo de 10 nNm, que será suficiente para posicionar la fibra con errores menores a  $5 \mu\text{m}$ .

La colocación de la fibra en la circunferencia de posicionamiento de cada robot se realiza mediante la rotación de un eje principal R1 colocado en el centroide del posicionador (figura 2.5), desde el que se extiende un brazo principal L1 de 3,464 mm idealmente. Y un eje secundario R2 colocado en el extremo del brazo principal, desde el que se extiende el brazo secundario L2 de 3,464 mm idealmente, al final del cual se colocará el cabezal de la fibra encargado de captar la radiación.

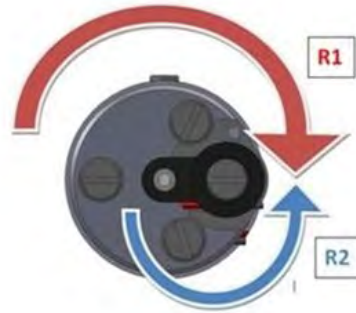


Figura 2.5: Detalle frontal.

La rotación del eje principal se realiza en sentido horario, con un ángulo máximo de  $360^\circ$  o 27520 pasos del motor 1. Por otro lado, la rotación del eje secundario se realiza en sentido antihorario con un ángulo máximo de  $180^\circ$  o 5120 pasos del motor 2.

No obstante, tanto el motor 1 como el motor 2 pueden llevarse más allá de  $360^\circ$  y  $180^\circ$  respectivamente, hasta llegar al límite físico del sistema ( $365^\circ$  y  $185^\circ$  respectivamente). Esta situación es extremadamente delicada, dado que un error en el paso puede llegar a ocasionar daños físicos en el posicionador. Por esta razón se guarda un ángulo de seguridad de  $5^\circ$  en cada motor y no se permite al software controlador superar los  $360^\circ$  y  $180^\circ$ .

Por último, la fibra óptica es guiada hasta su punto de exposición, en el extremo del brazo L2, mediante una canalización realizada en el interior del chasis del posicionador. Ésta garantiza que la fibra no sufra una torsión indeseada provocada por el movimiento de los ejes posicionadores, manteniendo en la medida de lo posible la señal limpia de distorsiones y minimizando las pérdidas.

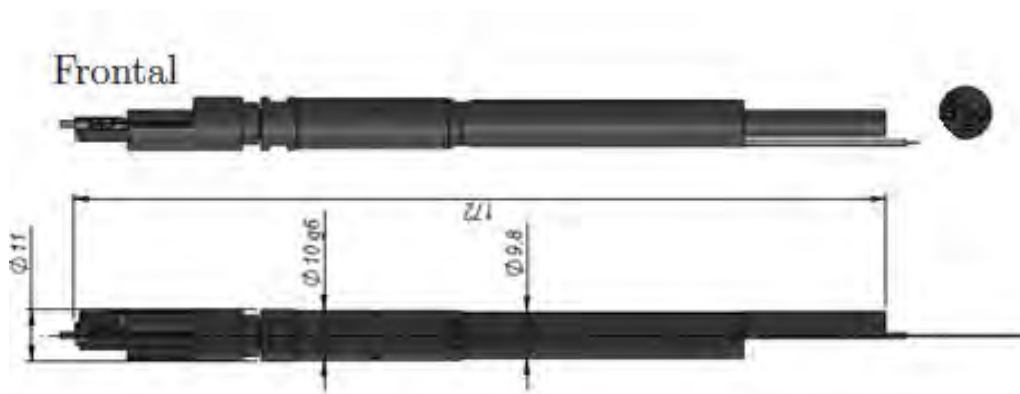


Figura 2.6: Detalle del posicionador coplanar.

Debido a la mecánica interna del motor, dado el mecanismo de posicionamiento, el movimiento del eje R1 induce una pequeña rotación residual de igual sentido en el eje R2. Por esto, el movimiento real del eje R2 (movimiento requerido de R2 más movimiento inducido por la rotación de R1) será denominado RR2. Esta rotación es proporcional al movimiento del eje principal, con un factor de proporción de 2.6875. En función de la siguiente expresión:

$$RR2 = R2 + R1/2,6875$$

Por esta razón se define un área de seguridad en la que puede operar el robot sin sufrir daños. Este área considera los pasos el eje R1 y los pasos verdaderos del eje RR2 según la figura 2.7.

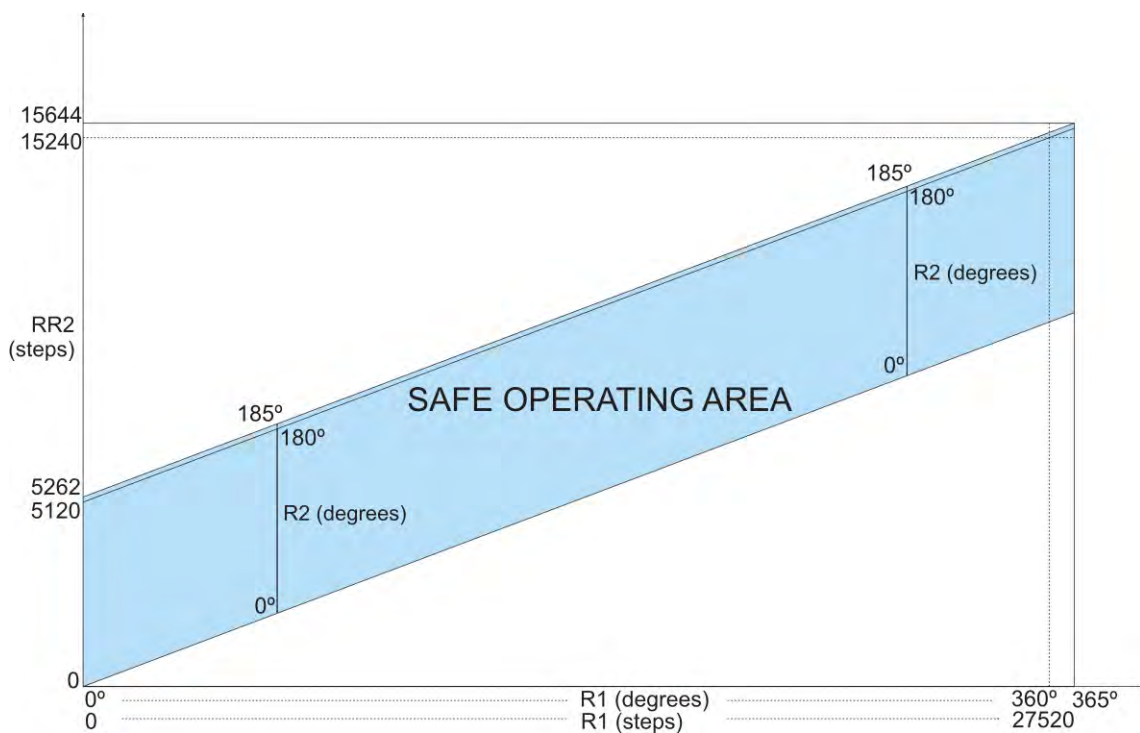


Figura 2.7: Safe Operate Area (SOA).

Esta figura representa el Área Segura de Operación, mostrando los pasos de R1 en el eje de abscisas y los pasos reales de R2 (RR2) en el eje de ordenadas. En primer lugar, en esta figura es posible observar una pendiente constante en el área de operación: ésta es producida por el crecimiento proporcional de RR2 a medida que crece R1 (en función de la constante de proporcionalidad 26875). Por otro lado, pueden observarse los pasos teóricos (sin inducción) de R2 en el origen (moviéndose estos de 0 a 5120 o 5262 pasos).

Por esta razón, todos los algoritmos de posicionamiento diseñados en este proyecto, irán encaminados a permitir el desplazamiento del robot a lo largo de esta área de seguridad, constatando en todo momento que no se exceden los límites definidos (27520 pasos de R1 y 15240 pasos de RR2), asegurando la integridad del posicionador en cualquier situación.

## 2.2. Introducción al control de motores

El objetivo de este proyecto es, entre otros, el diseño de la electrónica de control de los dos motores involucrados en cada posicionador. Éstos son motores paso a paso cuyo control y funcionamiento serán detallados más adelante. No obstante, a continuación se exponen los tipos de motores eléctricos cuyas características (dimensiones, movimiento rotacional, etc.) les hacen similares a los utilizados en los posicionadores.

### 2.2.1. Motores DC

Éstos son los motores comúnmente más conocidos. Son máquinas alimentadas con corriente continua (comúnmente pulsada), capaces de transformar la energía eléctrica en energía mecánica, provocando un movimiento rotatorio de un eje principal.

Estas máquinas, principalmente se componen de dos entidades:

- **Estátor:** da soporte mecánico al motor y posee un hueco cilíndrico que contendrá la segunda entidad.

En esta pieza se encuentran los *polos*. Éstos son comúnmente imanes o devanados de hilo de cobre sobre un núcleo de hierro, por los que se hará fluir una corriente eléctrica para generar un campo magnético.

- **Rotor:** es la pieza dinámica que comúnmente formará el eje del motor. En esta pieza se encuentran los devanados principales a los que llegará la corriente mediante escobillas.

Siguiendo la ley de la *Fuerza de Lorenz*, cuando un conductor por el que circula una corriente eléctrica se sumerge en un campo magnético, el conductor sufre una fuerza perpendicular al plano formado por el campo magnético y la corriente, según la siguiente expresión:

$$F = B \cdot l \cdot I$$

Donde:

$F$	—	Fuerza ( <i>newtons</i> ).
$B$	—	Densidad de campo magnético ( <i>teslas</i> ).
$I$	—	Intensidad eléctrica ( <i>amperios</i> ).
$l$	—	Longitud del conductor ( <i>metros</i> ).

Según ésta, cuando se hace circular una corriente por el rotor a través de las escobillas (figura 2.8), teniendo imantados los polos del estátor, esta energía eléctrica es transformada en movimiento rotatorio con un par proporcional a la tensión suministrada. No obstante, a la hora de alimentar estos motores se recurre a señales PWM (*Pulse-width modulation*). Éstas son trenes de pulsos de frecuencia y ancho variable, que evitan una alimentación continua de los polos, reduciendo considerablemente el

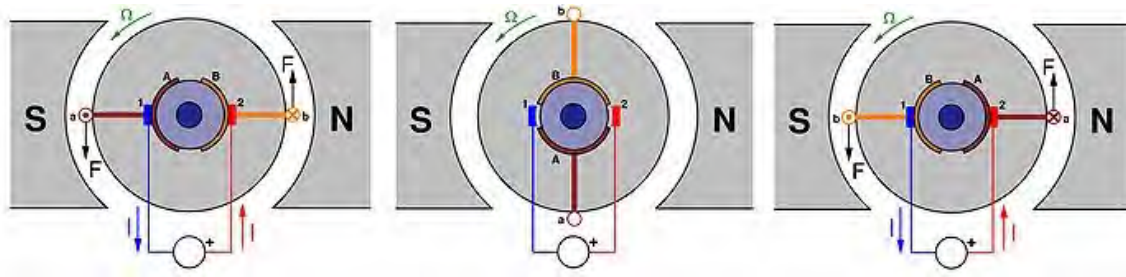


Figura 2.8: Generación de rotación en motor DC.

consumo de los motores y teniendo un control directo de la velocidad de rotación. Siendo esta velocidad proporcional al tiempo en que la señal está activa.

Por otro lado, el sentido de giro de estos motores lo dicta la polarización de las bobinas del rotor. Por ello, para cambiarlo, habrá que invertir la polarización de las mismas.

Por último, estos motores presentan varias desventajas en el campo del control de precisión frente al resto de motores que se presentan a continuación. En primer lugar, el control preciso de la posición en estos motores es muy tedioso, siendo necesaria mucha mecánica adicional. Por otro lado, estos motores no son capaces de mantener una posición fija mientras se les alimenta (con el objetivo de mantenerlos activos y calientes), lo que hará que no puedan ser aplicados en proyectos como el que nos concierne. Finalmente, para aumentar el par en este tipo de máquinas, habrá que aumentar proporcionalmente su tensión de alimentación, lo que nos aleja de nuestro propósito de mantener una tensión de alimentación constante.

### 2.2.2. Motores piezo-eléctricos

Estos motores se basan en el efecto piezo-eléctrico [22][23] de determinados materiales, con el fin de conseguir una vibración de frecuencias ultrasónicas[24], que generará un movimiento lineal o rotatorio sobre el rotor. Este efecto se produce cuando determinados materiales (comúnmente con estructura cristalina) son atravesados por un campo eléctrico, generándose dichas vibraciones.

Estos motores poseen una estructura similar a los motores DC (estátor y rotor). Comúnmente, existen tres componentes de material piezoeléctrico, dos de ellos fijos y uno móvil. El componente móvil se encuentra fijado al estátor y mediante su vibración (fases de dilatación y contracción) hace mover a los otros dos, que por medio de la fricción con el rotor, generarán el movimiento.

Este tipo de motores presenta una desventaja frente a los utilizados finalmente en el proyecto, dado que hay que mantenerlos constantemente alimentados para mantener una posición fija. Hecho que disparará el consumo del sistema y hará inviable la integración de 5000 unidades. No obstante, aun no siendo indicados para este proyecto, el campo de aplicación de este tipo de motores es muy extenso, dado que el efecto piezoeléctrico es capaz de generar movimientos lineales, ideales para otro tipo de aplicaciones (aplicaciones médicas, aeroespaciales, etc.).

## 2.2.3. Motores paso a paso

### 2.2.3.1. Fundamentos

El funcionamiento y principios de este tipo de motores, a diferencia de los motores piezo-eléctricos, es muy similar al de los sistemas DC. Éstos utilizan la combinación de inducción magnética de un cierto número de devanados, posicionados en el estátor, con el fin de colocar un rotor central en una cierta posición, manteniendo así ésta, sin necesidad de alimentar continuamente el sistema.

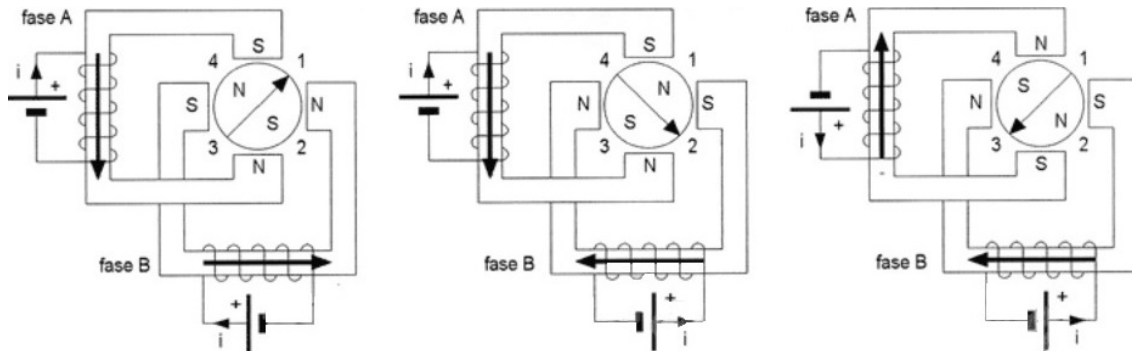


Figura 2.9: Esquema de funcionamiento de un motor bipolar.

Existen diferentes tipos de motores paso a paso: en función de su estructura, del número de devanados o de la forma de tratar éstos. Los más extendidos son los motores *bipolares*. Éstos poseen comúnmente un devanado sobre núcleo de hierro por fase, cuya polarización determina la orientación del imán integrado en el rotor central (figura 2.9). Por lo que será necesario invertir la corriente eléctrica en los devanados para invertir el polo magnético.

Este tipo de motores puede controlarse de varias formas:

- **Full stepping:** en este modo, se polariza exclusivamente un polo, lo que induce al rotor a moverse a una posición estable alineada con él. Este modo posee una menor resolución angular, no obstante, al mantener el rotor alineado con uno de los polos, cada posición será una posición estable, lo que evitará la pérdida de pasos tras la desconexión del sistema.
- **Half stepping:** en este modo, pueden polarizarse dos polos simultáneamente, lo que induce al rotor a colocarse en situaciones intermedias. Esta forma de control posee una mayor resolución angular, ya que dobla el número de posibles posiciones del rotor frente al modo anterior. No obstante, introduce una clara debilidad, dado que tras la desconexión del sistema, el rotor tiende a colocarse en la posición estable más cercana alineándose con los polos y perdiendo pasos.
- **Micro stepping:** en este modo se utilizan funciones sinusoidales para alimentar los devanados, pudiéndose solapar en mayor o menor medida la alimentación de más de un polo. De este modo se consigue un control a muy bajo nivel del paso dado y una resolución angular muy elevada.

### 2.2.3.2. Modulación de corriente

Al igual que en los motores DC, a la hora de alimentar cada devanado, independientemente del modo de funcionamiento deseado, se recurre a señales PWM. En el caso de los motores paso a paso, a este control se le denomina coloquialmente *chopeo de corriente*. En él se incrementa constantemente la corriente entregada a cada polo hasta llegar a un máximo predefinido, tras lo que se lleva de nuevo a 0 y pasado un cierto periodo vuelve a aumentarse.

Esta característica introduce multitud de ventajas. Por un lado, teniendo un control de la corriente máxima a entregar a cada devanado (modulación de corriente) tenemos un control exhaustivo del par generado. Por otro, controlando este proceso de modulación, pueden mantenerse los motores activos sin producir movimiento (frecuencia de modulación elevada), generando mayor o menor calor. Con esto, el sistema puede mantener una temperatura constante en condiciones adversas, reduciendo así el riesgo de rotura y la necesidad de otro tipo de mantenimiento.

### 2.2.3.3. Control

Para controlar este tipo de motores es habitual utilizar un circuito de potencia en *Puente H* (figura 2.10). Por un lado, estos circuitos protegen el sistema de una inversión de polarización o polarización errónea. Por otro, entregan la corriente necesaria para polarizar los devanados y hacer mover el motor, dado que la gran mayoría de los circuitos integrados utilizados para el control de movimiento de este tipo de motores (microcontroladores, circuitos lógicos, etc.) no son capaces de entregar la corriente suficiente y pueden romperse.

No obstante, en la actualidad, estos sistemas forman parte de circuitos integrados mucho más complejos, como los que se exponen en la sección 4.1.2, que generan un control y monitorización muy precisos del movimiento del motor.

En la figura 2.10, puede observarse la disposición de los dos devanados de un motor bipolar. En este caso, para polarizar la bobina «A», solo haría falta alimentar la base de los transistores T1 y T4. Por otro lado, si se quiere invertir la polaridad de esta bobina alimentaríamos los transistores T2 y T3. Sucediendo algo similar con la bobina «B» y los transistores T5 a T8.

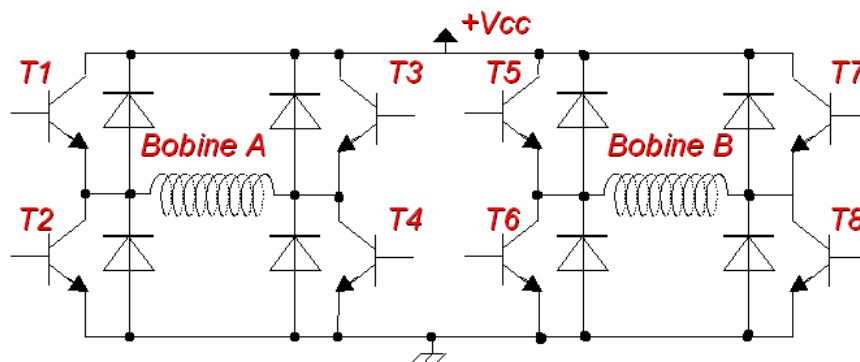


Figura 2.10: Circuito en puente H.

#### **2.2.3.4. Conclusiones**

Estos motores presentan multitud de ventajas frente a los expuestos con anterioridad. Por un lado, los motores paso a paso bipolares pueden ser dirigidos a la posición exacta deseada, con mayor o menor precisión, en función del modo de control al que estén sometidos (calidad que los hará apropiados para todo tipo de actividades de precisión). Por otro lado, éstos no necesitan estar permanentemente alimentados para no perder su posición, lo que reducirá considerablemente el consumo y por consiguiente el aumento de temperatura en ellos. Este último dato posee especial relevancia, considerando el elevado número de posicionadores que coexisten en el proyecto BigBOSS y la distorsión que la temperatura introduce en las imágenes obtenidas. Por último, estos motores son controlados por corriente, lo que permite el control exhaustivo del par entregado, manteniendo constante la tensión de alimentación.

Con todo esto, este tipo de motores serán seleccionados para la propuesta española de posicionadores en el proyecto BigBOSS.





# Capítulo 3

## Diseños preliminares

*Entre las dificultades se esconde una oportunidad.*

Albert Einstein

**Resumen:** en este capítulo se citan las razones por las que se ha optado por el diseño de sistemas de control individualizados a cada robot, se introduce la composición de cada sistema y se exponen los diferentes componentes que se han barajado para formar parte de ellos. Por otro lado, se indica cuales han sido seleccionados finalmente.

### 3.1. Planteamiento del diseño

Tras el estudio de los diferentes sistemas existentes fabricados para abordar la misma problemática (el posicionamiento automatizado de un conjunto de fibras ópticas en el plano focal de un telescopio) (sección 2.1), y la conclusión de que un sistema compuesto por un conjunto de posicionadores independientes que realicen su labor en paralelo, introduce notables mejorías respecto a posicionadores secuenciales (en cuanto a tiempo de posicionamiento, precisión y operatividad se refiere), se concluyó que la instalación de sistemas electrónicos independientes, encargados del control individualizado de cada uno de los robots introduce ventajas destacables.

Por un lado, el diseño de sistemas individualizados reduce significativamente la longitud de los cables que circulan desde los sistemas de control hasta los actuadores, dado que se precisan como mínimo 8 cables para controlar cada robot y esto, debido a la densidad de actuadores en 1 m de diámetro focal, supone un gran problema operativo.

Por otro lado, la dedicación de un sistema microcontrolado para cada actuador permite el trabajo en paralelo de los 5000 robots, reduciendo significativamente el tiempo de posicionado, dado que la única información que es preciso enviar a cada actuador es la posición absoluta de R1 y R2 (sección 4.2.2). El controlador individual se encargará de la traducción de esta posición y el control de cada motor. Esto libera al nodo superior de comunicación de la mayor parte del trabajo, haciendo que éste solo actúe como router (optimizando una vez más el tiempo de posicionado).

Finalmente, el diseño de sistemas individualizados posibilita una monitorización en tiempo real dedicada a cada robot, midiendo las diferentes constantes (temperatura, corriente y posición). Esto cede el control de alertas a cada actuador (sección 4.2.2.3), permitiendo que los nodos superiores únicamente trabajen como enrutadores de la comunicación.

No obstante, el único inconveniente destacable en la individualización del control de los posicionadores, es el coste total del proyecto. Sin embargo, el coste de estos sistemas es prácticamente despreciable en comparación con lo que se asumirá en la fabricación de las 5000 unidades robotizadas.

## 3.2. Descripción del sistema

La tarea principal de este Proyecto de Fin de Carrera es el diseño del hardware necesario para el control de cada uno de los posicionadores. Éste será el elemento central entorno al que girarán las diferentes variables de diseño, incluidas las aplicaciones software.

Durante el desarrollo del sistema definitivo, se ha pasado por tres etapas de prototipado diferenciadas, en las cuales se han ido añadiendo nuevos elementos que respondían a las necesidades eventuales del proyecto (sección 4.3), realizándose las diferentes pruebas de funcionamiento. No obstante, todos ellos presentan el mismo funcionamiento de base.

Este sistema, será el encargado de recibir y procesar los comandos enviados desde el ordenador central, a través de nodos intermedios de comunicación<sup>1</sup>. E irá encapsulado en el lateral de cada posicionador, llevando a cabo seis funciones fundamentales:

- Mover el posicionador.
- Monitorizar la temperatura de la electrónica.
- Monitorizar la temperatura de cada motor.
- Controlar la posición de cada eje.
- Controlar la modulación de corriente.
- Alertar al nodo superior de posibles fallos.

Las principales limitaciones presentadas en este proyecto son, entre otras, la superficie del hardware y el consumo de éste. Por ello, en primer lugar, debido al gran número de posicionadores que coexistirán en el plano focal del telescopio, el consumo de cada unidad no ha de ser muy elevado, dado que esto provocaría un consumo total considerable. Por otro lado, el aumento de la corriente que circula por el sistema,

---

<sup>1</sup>El sistema de comunicación, a través del cual la información es transportada desde la unidad central a cada uno de los actuadores, ha sido diseñado en el HCTLab, durante la realización del PFC «*Sistema de Comunicación para la gestión remota de un elevado número de Mini-Robots de posicionamiento en un entorno reducido*» por Jesús Castro Murillas.

conlleva un incremento de la temperatura en el plano focal, introduciendo distorsiones indeseables en la imagen obtenida. En segundo lugar, dada la proximidad de los actuadores, el hardware no deberá poseer una gran superficie, teniendo que ir encapsulado en el interior del chasis de cada robot. Por esta razón, el diseño final de la electrónica de control tendrá que ser inferior a 0.8 cm en una de sus dimensiones, respetando la envolvente del robot (figura 2.6).

Finalmente, una característica a tener en cuenta a la hora de seleccionar entre varios componentes barajados, es la fabricación de cada circuito integrado en los distintos tipos de encapsulados, que compondrán las diferentes fases de desarrollo del sistema. Teniendo que poder encontrarse en modelos de reducida superficie para la electrónica definitiva (sometiéndose estos a las limitaciones antes descritas) y modelos de mayor tamaño para los prototipos iniciales (facilitando su manipulación).

### 3.3. Estudio de componentes

Tras el estudio de diferentes soluciones para el diseño de la electrónica de control de los robots, se concluyó en el desarrollo de un sistema microcontrolado basado en el manejo de los robots por medio de la modulación de corriente en sus bobinas. Por esta razón, se hizo necesaria la integración de componentes capaces de controlar los motores [25][26][27], entregando la corriente necesaria y sistemas capaces de manejar la información recibida de los nodos superiores, respetando las limitaciones de consumo y superficie.

El componente principal de este sistema es un microcontrolador encargado de recibir los comandos del nodo superior, procesarlos y manejar la información que circula en el actuador. En esta electrónica se pueden destacar también dos drivers de bajo consumo y altas prestaciones encargados de entregar la corriente suficiente a cada motor, monitorizar la posible sobrecarga en éstos y controlar la modulación de corriente, así como los modos de funcionamiento (half, full o microstep). Estos drivers se apoyan en dos potenciómetros digitales que controlan la tensión que cae en dos de sus terminales de control, determinando la corriente máxima que cada uno de éstos entregará a los motores. Por otro lado, la electrónica incluye un sensor que monitoriza la temperatura del PCB, conectores para enlazar con los motores y los diferentes sensores instalados en estos y las resistencias y condensadores de desacople que permiten configurar el sistema.

La etapa de potencia que configura la alimentación de la electrónica no se encuentra en el diseño definitivo de ésta. No obstante, en prototipos intermedios (sección 4.3.2) se incluye para tener más control sobre el sistema global y realizar los estudios de consumo de las diferentes partes de la electrónica.

Finalmente, durante la fase de planteamiento del diseño, han sido barajados diferentes modelos para cada uno de los componentes citados anteriormente. La selección del componente definitivo que formará parte de la electrónica, ha estado sujeta a estrictas restricciones de consumo, operatividad y superficie. Por otro lado, la elección definitiva del protocolo de comunicación principal dentro de la electrónica (sección 4.2.1), ha tomado un papel protagonista en esta decisión.

A continuación se exponen los diferentes modelos estudiados para cada uno de los componentes:

### 3.3.1. Microcontrolador

	ATxmega32A4U	ATtiny40	LPC11U14
Fabricante	Atmel	Atmel	NXP
Dimensión mínima (mm)	5x5	3x3	<b>4.5x4.5</b>
Número de pines	44	20	48
Fast $I^2C$	No	No	<b>Sí(1)</b>
SPI	Sí(2)	Sí(1)	Sí(2)
Memoria Flash (kB)	32	4	32
Alimentación (V)	1.6-3.6	1.8-5.5	1.8-3.3
Frec. de trabajo (MHz)	0-32	0-12	<b>0-50</b>
Consumo mínimo en activa (mA)	9.6	3.2	<b>2</b>
Consumo mínimo durmiendo (mA)	3.8	0.8	<b>1</b>

Cuadro 3.1: Microcontroladores candidatos a formar parte de la electrónica.

Estos tres microcontroladores han sido seleccionados dadas sus dimensiones, reducido consumo y coste respecto a otros integrados similares en el mercado. No obstante, muy pocos microcontroladores cuyas dimensiones se sujeten a las requeridas poseen un bus fast  $I^2C$  (estando éste dedicado a procesadores de gran potencia y tamaño) por lo que finalmente el microcontrolador LPC11U14 fue escogido (sección 4.1.1).

### 3.3.2. Driver de motor

Para este componente se seleccionó desde un principio el circuito integrado descrito en la sección 4.1.2, dado su reducido consumo y el control interno que éste realizaba de la señal entregada, así como la capacidad de este dispositivo de alertar de problemas de sobre-corriente (sección 4.2.2.3).

### 3.3.3. Potenciómetros digitales

	MCP414X	ISL22429X	AD5160
Fabricante	Microchip	Intersil	Analog Devices
Dimensión mínima (mm)	4x4	5x3	<b>3x3</b>
Número de pines	8	8	8
SPI	Sí(1)	Sí(1)	Sí(1)
Resistencia máxima ( $k\Omega$ )	120	50	100
Resolución (bits)	8	7	8
Alimentación (V)	2.7-5.5	-0.3-6	2.7-5.5
Consumo con SPI activo ( $\mu A$ )	450	450	<b>5</b>
Consumo con SPI inactivo ( $\mu A$ )	5	2	<b>3</b>

Cuadro 3.2: Potenciómetros candidatos a formar parte de la electrónica.

Tras una selección de los potenciómetros digitales que mejor se adaptaban a los requerimientos de consumo y dimensiones, fue seleccionado el circuito integrado modelo AD5160 de Analog Devices (sección 4.1.3), debido a su notable ventaja en lo que a consumo se refiere, estando éste casi un orden de magnitud por debajo de sus semejantes.

### 3.3.4. Sensores de temperatura

	MCP9808	TMP125	TC72
Fabricante	Microchip	Texas Instruments	Microchip
Dimensión mínima (mm)	2x3	3x3	<b>3x3</b>
Encapsulados mayores	Sí	No	<b>Sí</b>
Número de pines	8	6	8
$I^2C$	Sí(1)	No	No
SPI	No	Sí(1)	<b>Sí(1)</b>
Límites (°C)	-40-125	-40-125	-55-125
Precisión Máxima (°C)	0.25	0.5	2
Resolución (bits)	8	10	10
Tiempo de conversión (ms)		60	150
Alimentación (V)	2.7-5.5	2.7-5.5	2.65-5.5
Consumo comunicando ( $\mu A$ )	200	50	250
Consumo en standby ( $\mu A$ )	0.1	36	<b>0.1</b>

Cuadro 3.3: Sensores de temperatura candidatos a formar parte de la electrónica.

Tras el proceso de selección fue escogido el circuito integrado modelo TC72 de Microchip (sección 4.1.4). Este componente no presenta tanta precisión como sus semejantes, no obstante, posee un consumo muy pequeño en modo inactivo, lo que reducirá en gran medida el consumo medio del circuito integrado. Por otro lado, este integrado posee encapsulados en varios tamaños, lo que ayudará a la hora de la fabricación manual de los diferentes prototipos.

## 3.4. Presupuesto

Una vez seleccionados los componentes se realizó un presupuesto dedicado a cada uno de los modelos de la electrónica (sección 4.3). Éstos fueron planificados por medio del coste de los componentes ofrecido por *Farnell España* (para productos en stock sin coste de envío) y ofrecen una visión general del coste de cada una de las unidades de control diseñadas en este PFC.

Estos presupuestos se diseñaron para la realización de un único prototipo de cada modelo (tablas 3.4, 3.5 y 3.6) en los que se seleccionaron componentes individuales. Por otro lado, se realizó un presupuesto general para la fabricación del modelo final de la electrónica (tabla 3.7) en el cual se consideró la compra de grandes cantidades de circuitos integrados con el objetivo de fabricar el sistema completo de 5000 unidades de control dedicado.

### 3.4.1. Modelo v.1 de la electrónica

#### MODELO v.1

Componente	Cantidad	Precio por unidad (€)	Coste total (€)
Resistencias	8	0.03	0.24
Condensadores cerámicos	11	0.06	0.66
Condensadores electrolíticos	6	0.4	2.4
LPCXpresso	1	18,57	18.57
DRV8834	2	2.66	5.32
AD5160	2	1.98	3.96
TC72	1	0.79	1.58
Conectores de 27 pines	2	6.5	13
Conectores de 4 pines	2	0.36	0.72
Conectores de 3 pines	2	0.3	0.6
Conectores de 2 pines	2	0.3	0.6
Cable USB	1	3	3
<b>TOTAL:</b>			<b>50.65</b>

Cuadro 3.4: Presupuesto para el modelo v.1 de la electrónica

### 3.4.2. Modelo v.2 de la electrónica

#### MODELO v.2

Componente	Cantidad	Precio por unidad (€)	Coste total(€)
Resistencias	19	0.03	0.57
Condensadores cerámicos	20	0.06	1.2
Condensadores electrolíticos	6	0.4	2.4
LPC11u14	1	3.33	3.33
DRV8834	2	2.66	5.32
AD5160	2	1.98	3.96
TC72	1	0.79	0.79
Reg103	1	3.98	3.98
Switch	1	1.57	1.57
LEDs	2	0.13	0.26
Jumpers	2	0.43	0.86
Conector USB	1	1.15	1.15
Conectores de 4 pines	4	0.36	1.44
Conectores de 2 pines	3	0.3	0.9
Cable USB	1	3	3
<b>TOTAL:</b>			<b>30.73</b>

Cuadro 3.5: Presupuesto para el modelo v.2 de la electrónica

### 3.4.3. Modelo v.3 de la electrónica (una unidad)

MODELO v.3 (1 UNIDAD)

Componente	Cantidad	Precio por unidad (€)	Coste total(€)
Resistencias	17	0.03	0.51
Condensadores cerámicos	18	0.06	1.08
Condensadores electrolíticos	5	0.4	2
LPC11u14	1	3.33	3.33
DRV8834	2	2.54	5.08
AD5160	2	1.98	3.96
TC72	1	0.79	0.79
Conectores de 4 pines	4	0.36	1.44
Conectores de 2 pines	3	0.3	0.9
<b>TOTAL:</b>			<b>19.09</b>

Cuadro 3.6: Presupuesto para el modelo v.3 de la electrónica

### 3.4.4. Modelo v.3 de la electrónica (5000 unidades)

MODELO v.3 (5000 UNIDADES)

Componente	Cantidad	Precio por unidad (€)	Coste total(€)
Resistencias	85000	0.009	765
Condensadores cerámicos	90000	0.009	810
Condensadores electrolíticos	25000	0.1	2500
LPC11u14	5000	1.81	9050
DRV8834	10000	1.53	15300
AD5160	10000	0.95	9500
TC72	5000	0.62	3100
Conectores de 4 pines	20000	0.16	3200
Conectores de 2 pines	15000	0.1	1500
<b>TOTAL:</b>			<b>45725</b>
<b>PRECIO POR UNIDAD:</b>			<b>9.145</b>

Cuadro 3.7: Presupuesto para el sistema completo de control

## 3.5. Conclusiones

Como se ha descrito, tras el proceso de selección, han sido elegidos los componentes que mejor se ajustan a las restricciones de consumo y superficie marcadas por el proyecto. No obstante, los diseños mostrados en este PFC no son más que prototipos muy próximos a la electrónica definitiva. Estos dan una idea de como será el diseño final, no obstante, están sujetos a los cambios que puedan solucionar problemáticas

---

eventuales. Un ejemplo de esto es la monitorización de las variables globales en los actuadores (temperatura y posición). El método que se seguirá para abordar este problema no está aun definido, por lo que la adaptación de la electrónica diseñada a los sensores que monitorizarán estas variables (y por lo tanto los circuitos integrados que la componen), está sujeta a constantes cambios.



# Capítulo 4

## Diseño hardware

*A fuerza de construir bien, se llega a buen arquitecto.*

Aristóteles

**Resumen:** en este capítulo se presentan los componentes finalmente seleccionados, se detalla el funcionamiento del sistema hardware, se desarrollan las etapas de diseño y se presentan los diferentes prototipos fabricados antes de llegar al sistema definitivo.

### 4.1. Componentes seleccionados

Tras el estudio de los diferentes componentes que pudiesen formar parte de la electrónica a diseñar, se realizó una selección basándose en criterios de consumo y superficie. No obstante, durante las primeras etapas de diseño, la consideración del uso de protocolos de comunicación serie de alta velocidad como *fast I<sup>2</sup>C* (sección 4.2.1) para la conexión con los nodos superiores, o el uso del protocolo SPI para la interconexión serie de algunos componentes, redujo la búsqueda.

A continuación se exponen los diferentes circuitos integrados que finalmente forman parte de la electrónica de control, presentándose sus propiedades fundamentales, características de consumo y el encapsulado seleccionado:

#### 4.1.1. Microcontrolador

Tras valorar los microcontroladores ATxmega32A4U y ATtiny40 (tabla 3.3) debido a su reducido consumo y su óptima superficie, se ha optado definitivamente por el microcontrolador *LPC1114* de NXP. Éste presenta unas dimensiones muy reducidas (en sus encapsulados de menor tamaño) y un consumo menor a muchos circuitos integrados similares. Además, este microcontrolador es compatible con los protocolos de comunicación elegidos para el diseño final, teniendo integrado un módulo de comunicación USB que facilita notablemente la realización de los primeros prototipos, en los que la comunicación externa se realizará a través de este puerto.

El LPC11U14 es un computador ARM *Cortex-M0* de 32 bits capaz de realizar tareas muy complejas con reducidos recursos que, además de poseer las prestaciones mostradas en la tabla 3.3, destaca por las siguientes características:

- Memoria RAM de 6kB.
- Memoria ROM para boot de 16kB.
- Programación mediante ISP<sup>1</sup> (*In-System Programming*).
- Interface JTAG.
- 40 puertos GPIO (*General Purposte Input Output*).
- Ocho puertos ADC (*Analog Digital CONverter*).
- Un WWDT (*Windowed WatchDog Timer*).
- Comunicación directa mediante USB<sup>2</sup> (*Universal Serial Bus*).

#### 4.1.1.1. Encapsulados

Como se ha explicado en este capítulo, los circuitos integrados seleccionados han de poder adquirirse en diferentes encapsulados. Éstos han de ajustarse a las dimensiones de los diferentes prototipos, pudiendo entrar en las restricciones mínimas del prototipo final ( $< 8 \times 8 \text{mm}$ ). En este caso, el LPC11U14 puede encontrarse en cuatro tipos de encapsulados diferentes, todos dentro de las limitaciones de superficie:

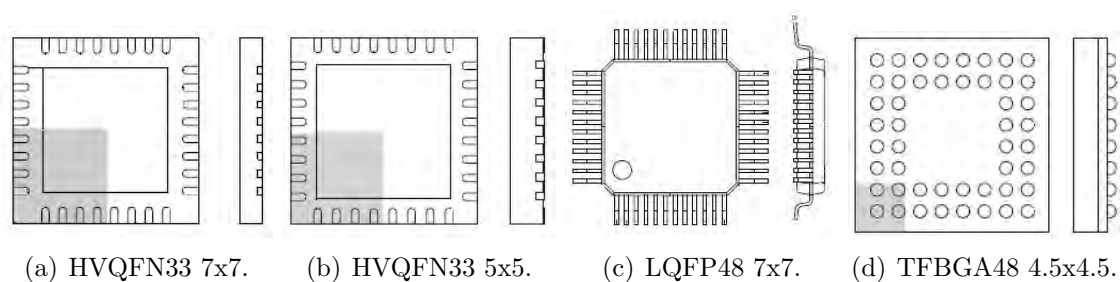


Figura 4.1: Encapsulados del microcontrolador LPC11U14 (mm).

El encapsulado que se ha utilizado para los prototipos previos al diseño final es el LQFP48 de  $49 \text{mm}^2$  con 48 pines y 0.5 mm entre cada uno, dado que éste facilita mucho el soldado manual y las labores de prototipado. No obstante, se ha optado por el modelo BGA para formar parte del diseño final, dado que sus dimensiones permiten entrar dentro de los 8 mm del PCB dejando espacio para el rutado, considerando que este último prototipo lo soldará una entidad ajena al HCTLab.

<sup>1</sup> Esta característica nos permitirá programar el micro una vez esté integrado en el sistema

<sup>2</sup> Esto evitará el uso de interfaces extra

#### 4.1.1.2. Alimentación y consumo

Uno de los criterios de selección de este microcontrolador ha sido su bajo consumo. El LPC11U14 presenta un consumo de aproximadamente 2 mA en modo activo (consumo muy inferior al presente en microcontroladores similares). Por otro lado, este micro presenta cinco modos de funcionamiento: *Active mode*, *Sleep mode*, *Deep Sleep mode*, *Power Down mode* y *Deep Power Down mode*. Estos modos permiten reducir aun más el consumo en momentos en los que no se requiera un nivel de procesamiento muy elevado.

Modo	Alimentación	Frecuencia del reloj	Consumo
Active mode	3.3 V	12MHz	2.2000 mA
		50MHz	7.0000 mA
Sleep mode	3.3 V	12 MHz	1.0000 mA
Deep Sleep mode	3.3 V	12 MHz	0.3600 mA
Power Down mode	3.3 V	12 MHz	0.0020 mA
Deep Power Down mode	3.3 V	12 MHz	0.0002 mA

Cuadro 4.1: Consumo medio ejecutando un *While(1)* desde la flash a 25 °C y 3.3 V.

En esta aplicación se ha configurado la frecuencia del reloj a 12 MHz y se utiliza una tensión de alimentación de 3.3 V. Por un lado, se ha utilizado el modo Active para realizar el procesamiento de datos y el movimiento de los motores. Por otro, se usa el modo Sleep para mantenerse a la espera de recibir nuevos comandos. Por lo que en ningún momento se superarán los 2.4 mA de consumo en el microcontrolador a temperatura ambiente.

#### 4.1.2. Drivers de motor

Para este diseño se ha utilizado el circuito integrado *DRV8834* de Texas Instruments. Este componente es capaz de realizar un control a muy bajo nivel del movimiento de un motor paso a paso o dos de continua, contando con las siguientes características:

- Corriente máxima de 1.5 A en continua y 2.2 A de pico por puente-H.
- Control de modos de funcionamiento hasta *1/32 Step Mode*.
- Monitorización y alerta de sobrecarga o sobretensión.
- Control de la modulación de corriente.
- Modo *Sleep* de funcionamiento en ausencia de movimiento.

Utilizando motores paso a paso, este circuito integrado controla el sentido de giro y configura automáticamente la polarización de cada devanado en función del paso en el que se encuentre, tomando como entrada únicamente el flanco de subida de un pulso, que representa cada paso y varias señales de control.

El control de este circuito integrado, para motores paso a paso, se realiza con 8 señales de configuración que manejan a bajo nivel los movimientos ordenados por el driver (tabla 4.2).

Pin	E/S	Descripción
nSLEEP	Entrada	Habilita el modo de funcionamiento Sleep
nEN	Entrada	Habilita los dos devanados del motor
STEP	Entrada	Señal de pasos, un flanco de subida representa un paso.
DIR	Entrada	Control del sentido de giro
nFAULT	Salida	Señal de alerta de sobre carga
CONFIG	Entrada	Configura el modo de control (STEP/PHASE)
M1	Entrada	Control junto a M0 del modo de movimiento (full step, half step, etc.)
M0	Entrada	Control junto a M1 del modo de movimiento (full step, half step, etc.)

Cuadro 4.2: Señales de configuración para el driver DRV8834.

Para el control de la modulación de corriente (figura 4.2), el DRV8834 cuenta con una entrada analógica de control y una resistencia de referencia que combinadas (sección 4.2.2.2), marcan el límite de corriente que este circuito integrado puede entregar a cada devanado, siguiendo la siguiente expresión:

$$I_{chop} = \frac{xV_{ref}}{5 \cdot R_{ISENSE}}$$

Pudiendo ser  $xV_{ref}$ : tanto  $AV_{ref}$  como  $BV_{ref}$ , siendo éstas las señales analógicas de control para cada uno de los devanados del puente-H y  $R_{ISENSE}$  la resistencia de referencia de cada puente.

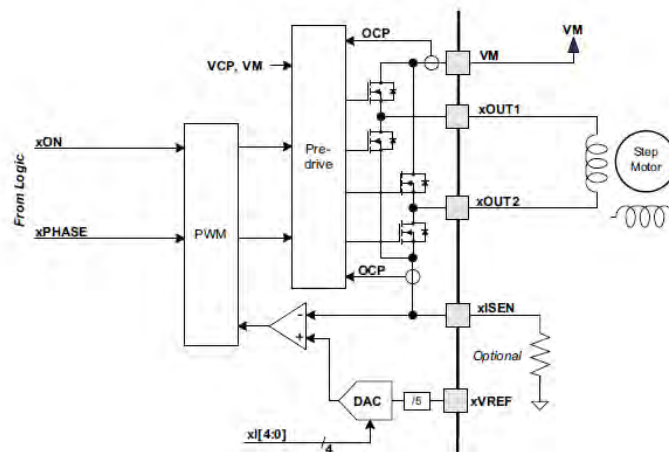


Figura 4.2: Proceso interno de control de la modulación de corriente.

En la figura 4.2 puede observarse el control interno y creación de la señal PWM que será la encargada de modular la corriente. Ésta es generada comparando la tensión que cae en bornes de la resistencia de referencia  $R_{ISENSE}$ , cuando por ella fluye la corriente de uno de los devanados del motor, con el valor analógico de la entrada  $xV_{ref}$ .

Las señales analógicas de control serán generadas por los potenciómetros digitales tomando como referencia una señal de 2 V entregada por los propios drivers como se explicará en la sección 4.2.2.2. Este proceso, controlando los potenciómetros digitales por medio del software del microcontrolador, permitirá la parametrización por parte del usuario de la modulación de corriente.

#### 4.1.2.1. Encapsulados

Al igual que todos los circuitos integrados seleccionados para este proyecto, el DRV8834 cuenta con varios encapsulados adaptados a cada uno de los prototipos realizados.

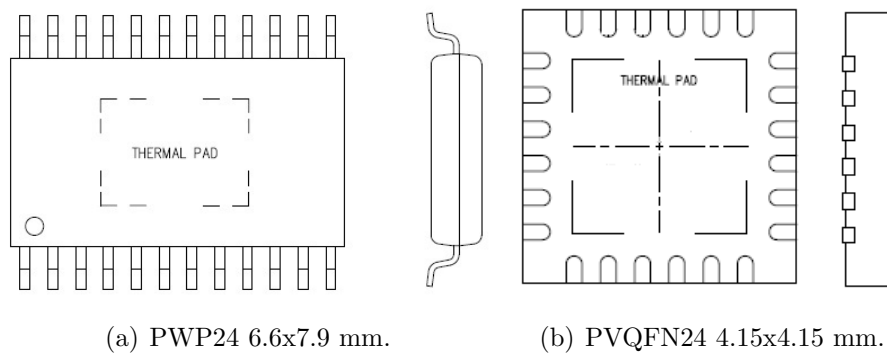


Figura 4.3: Encapsulados del driver de motor DRV8834.

El encapsulado que se ha seleccionado para los prototipos intermedios ha sido el PWP, dado su fácil manejo y la posibilidad de poder soldarlo a mano con las herramientas disponibles en el laboratorio. En el diseño final, sin embargo, ha sido seleccionado el modelo PVQFN debido a sus dimensiones.

#### 4.1.2.2. Alimentación y consumo

Al igual que el resto de componentes utilizados para este diseño, el DRV8834 ha sido seleccionado por su bajo consumo y la posibilidad de actuar en modo *Sleep*. Alcanzando consumos como los mostrados en la tabla 4.3.

Modo	Alimentación	Consumo
Active	5 V	2.40 mA
	10 V	2.75 mA
Sleep	5 V	0.6 $\mu$ A
	10 V	9.6 $\mu$ A

Cuadro 4.3: Consumos medios del driver DRV8834.

### 4.1.3. Potenciómetros digitales

Para configurar la corriente máxima modulada que los drivers entregarán a los motores, se utilizan dos potenciómetros digitales modelo AD5160 de Analog Devices. Éstos son sistemas adaptados al bus de comunicación SPI (*Serial Peripheral Interface*), que variarán la resistencia existente entre dos de sus terminales en función del código recibido por el bus serie.

Cada uno de estos potenciómetros tomará la tensión de 2 V cedida por uno de los drivers, como tensión de referencia (terminal A). Con ésta variarán la tensión entre sus terminales W y B en función del valor de la resistencia interna (figura 4.4). Tensión que tomará el driver para configurar la modulación de corriente.

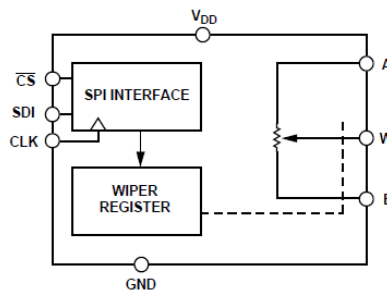


Figura 4.4: Funcionamiento interno del potenciómetro AD5160.

Éste es un potenciómetro de bajo consumo y reducidas dimensiones, con 256 posiciones y hasta 100 K $\Omega$  de resistencia máxima. Este dispositivo utiliza el bus de comunicación serie SPI como se expone en la subsección 4.2.1.

Para estimar la resistencia en bornes de los terminales W y B, el potenciómetro se basa en la siguiente expresión:

$$R_{WB}(D) = \frac{D}{256} \cdot R_{AB} + R_W$$

Donde:

- $D$  — Código serie recibido por el bus SPI (8bits).
- $R_{WB}$  — Resistencia en bornes de los terminales W y B.
- $R_{AB}$  — Resistencia máxima (5 K $\Omega$ , 10 K $\Omega$ , 50 K $\Omega$  o 100 K $\Omega$ )
- $R_W$  — Resistencia residual (60  $\Omega$  normalmente).

Con esta expresión, enviando únicamente un byte por el bus SPI, se varía la resistencia interna entre 60  $\Omega$  y 100 K $\Omega$ . Con lo que se consigue, tomando una tensión de 2 V en el terminal A y conectando el terminal B a tierra, una variación de tensión en el terminal W de entre 1 mV y 2 V.

#### 4.1.3.1. Encapsulados

Este dispositivo posee un encapsulado de dimensiones muy reducidas (figura 4.5), lo que posibilitará su integración tanto en los prototipos intermedios como en el diseño final de la electrónica.

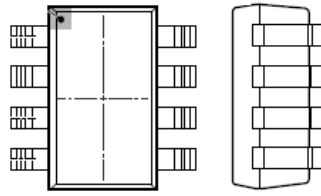


Figura 4.5: SOT-23 3x3 mm.

#### 4.1.3.2. Alimentación y consumo

Al igual que el resto de componentes utilizados en este proyecto, una de las características principales por las que se ha seleccionado el potenciómetro digital AD5160 ha sido su bajo consumo (tabla 4.4). Éste es el único dispositivo del diseño que carece de modo *Sleep* de funcionamiento, no obstante su consumo no es significativo respecto a dispositivos de mayor peso como los drivers de motor, siendo el de éstos un orden de magnitud superior.

Modo	Alimentación	Consumo
Active	3.3 V	3 $\mu$ A

Cuadro 4.4: Consumo medio del potenciómetro AD5160.

#### 4.1.4. Sensores de temperatura

Se usan para monitorizar la temperatura tanto de los motores como del circuito impreso, con el objetivo de percibir un posible fallo en el robot o un mal funcionamiento del sistema. En este diseño se ha contado con el sensor de temperatura TC72 de Microchip. Éste es un sensor de comunicación serie, que al igual que los potenciómetros AD5160, utiliza el bus de comunicación SPI como se explica en la subsección 4.2.1. Éste es un circuito integrado de micro-consumo y reducidas dimensiones, que cuenta con las siguientes características:

- Rango de medición de -55 °C a 125 °C.
- Resolución de 10bits (0.25 °C/bit).
- Modo de lectura continuo (*Active mode*).
- Modo de lectura por disparo (*Sleep mode*).

Este circuito integrado cuenta con dos modos de funcionamiento. En el primero es posible realizar una lectura continua del valor de temperatura leído por el sensor, teniendo en cuenta los 100 ms que tarda el dispositivo en realizar la conversión Analógico-Digital. Por otro lado, este circuito integrado cuenta con un modo de conversión por disparo, en el que se permanece en modo *shutdown* hasta que se da una orden de conversión, tras la cual se realiza una única lectura. Para configurar estos modos, el TC72 cuenta con unos registros de configuración que habrá que modificar en función del modo que interese utilizar. Tanto la modificación de los

registros de configuración, como la recepción de comandos de lectura se realizan por medio del bus serie SPI.

La comunicación con el sensor se realiza mediante el envío de un byte, indicando el registro al que se desea acceder. Si éste es un registro de configuración, se escribirá a continuación un byte con la información de configuración pertinente. Si por el contrario, éste es un registro de solo lectura, a continuación el sensor responderá devolviendo la información almacenada en él (figura 4.6).

El valor de la temperatura leída será almacenado en dos registros de un byte. El primero indicando la parte entera del valor leído (siendo el bit más significativo, el signo del dato) y el segundo, representando la parte decimal de éste.

Finalmente, el contenido del registro de configuración puede ser consultado en cualquier momento, leyendo el registro adecuado (figura 4.5).

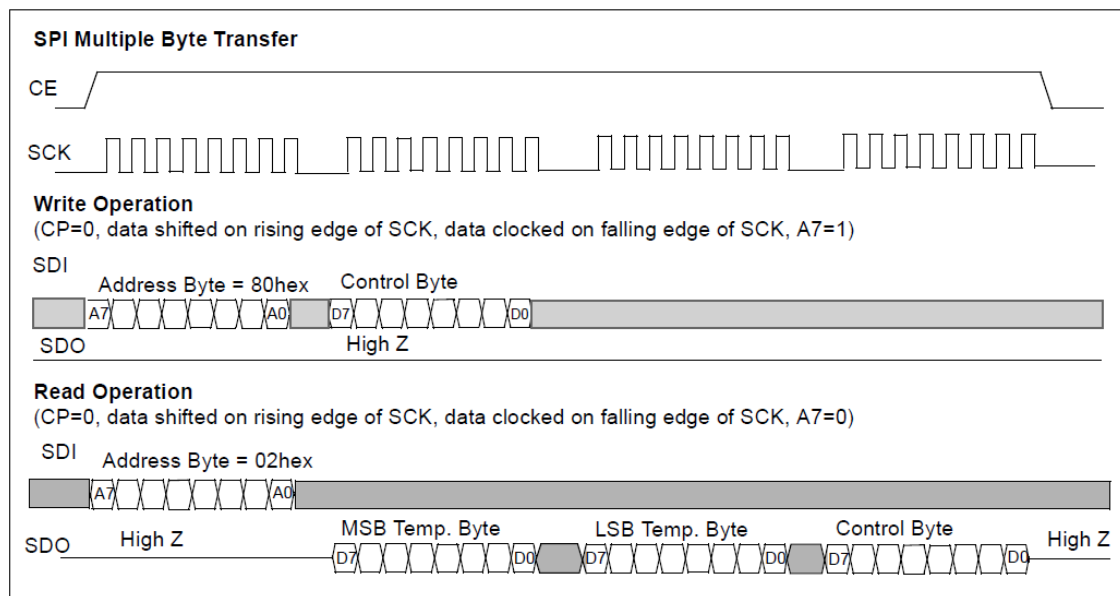


Figura 4.6: Comunicación con los sensores de temperatura TC72.

En la tabla 4.5 pueden observarse la dirección de memoria de los diferentes registros utilizados, así como su posible contenido. Siendo los bits del registro de control, OS (*One-Shot*) y SD (*Shutdown*) los encargados de configurar el modo de funcionamiento del dispositivo (tabla 4.6).

Para esta aplicación, el sensor permanecerá en modo shutdown, y se realizará una lectura cada cierto tiempo, reduciendo el consumo del sensor.

Registro	Dir. Lec.	Dir. Esc.	b7	b6	b5	b4	b3	b2	b1	b0
Control	0x00	0x80	0	0	0	OS	0	1	0	SD
Parte decimal	0x01	N/A	T1	T0	0	0	0	0	0	0
Parte entera	0x02	N/A	T9	T8	T7	T6	T5	T4	T3	T2
ID	0x03	N/A	0	1	0	1	0	1	0	0

Cuadro 4.5: Registros internos del sensor de temperatura TC72.



Modo	OS	SD
Lectura continua	-	0
Shutdown	0	1
Por disparo	1	0

Cuadro 4.6: Configuración de modos de funcionamiento del sensor TC72.

#### 4.1.4.1. Encapsulados

Como en el resto de componentes, el encapsulado MSOP (figura 4.7) del sensor TC72 entra dentro de las restricciones espaciales del diseño final. Además, debido a su fácil manipulación, este encapsulado ha sido utilizado en todas las fases de prototipado.

No obstante, este circuito integrado es fabricado también en el modelo DFN, lo que reduce aún más su tamaño y lo hace apropiado para formar parte de la electrónica final que irá encapsulada en el robot.

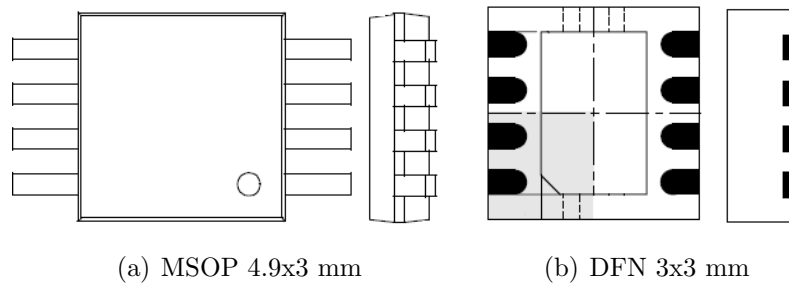


Figura 4.7: Encapsulados del sensor de temperatura TC72.

#### 4.1.4.2. Alimentación y consumo

Una de las características por las que este circuito integrado ha sido seleccionado, como el resto de componentes, es su reducido consumo. Dado que en el sistema coexisten tres sensores y la gran mayoría del tiempo permanecen en modo *shutdown*, el consumo medio total de los tres dispositivos será muy reducido.

Modo	Alimentación	Consumo
Lectura continua	3.3 V	250 $\mu$ A
Shutdown	3.3 V	0.1 $\mu$ A

Cuadro 4.7: Consumos y alimentación del sensor TC72.

No obstante, en función de la configuración requerida en el momento, es posible aumentar la frecuencia con la que la unidad central envía peticiones de medida (sección 4.2.2), esto incrementa el consumo medio del sistema. Sin embargo, el consumo medio total de los tres sensores instalados en cada actuador no es relevante, en comparación con la corriente necesaria para mover los motores o hacer funcionar el microcontrolador.

### 4.1.5. Placa de desarrollo

Para el diseño de los primeros prototipos, se ha utilizado el sistema de desarrollo LPCXpresso (figura 4.8) facilitado por NXP [28]. Ésta es una placa de prototipado que integra tanto el microcontrolador LPC11U14 <sup>1</sup> como las fases de potencia, el programador y el sistema de *debugger*. Esta placa ha sido utilizada como base de pruebas previa a la integración del microcontrolador con el resto de componentes. Con el fin de realizar diferentes estudios de consumo y operatividad.

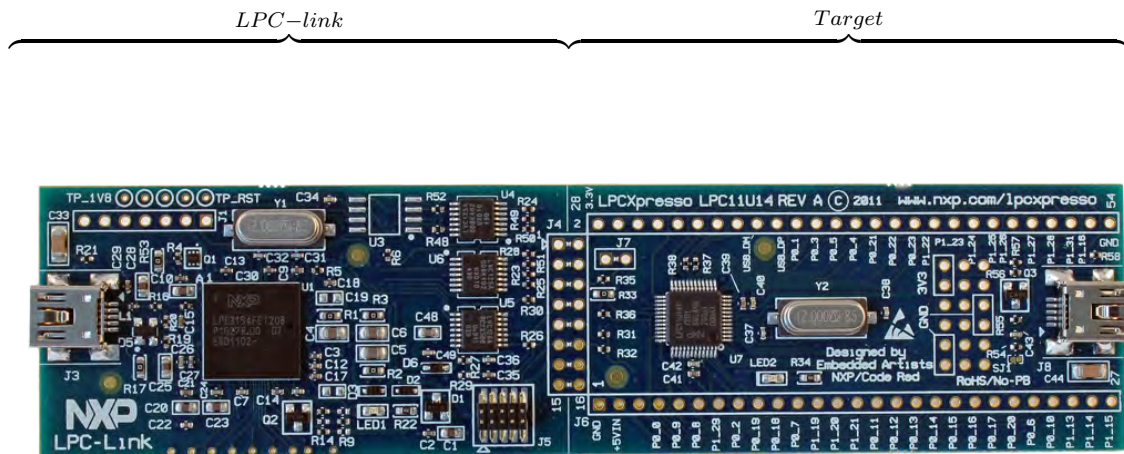


Figura 4.8: Placa de prototipado LPCXpresso de NXP.

El sistema de desarrollo LPCXpresso cuenta con 54 pines de E/S, entre los que se encuentran tanto los terminales de E/S del microcontrolador, como aquellos dedicados a la alimentación, programación, comunicación externa, etc.

Para el prototipado de la electrónica esta placa introduce una gran versatilidad, dado que puede cortarse y dividirse en dos partes. Una exclusiva para la programación y el *debugger* mediante JTAG (*LPC-link*) y otra que incluye únicamente el microcontrolador, los terminales de E/S y el conector para la comunicación USB (*Target*).

Por otro lado, este sistema puede alimentarse tanto a través del conector USB, como por medio de alimentación externa usando los pines dedicados para ello. No obstante, la sub-tarjeta que incluye la fase de programación y *debugger*, posee a su vez la etapa de potencia, que alimenta al microcontrolador con 3.3 V, obteniéndolos de los 5 V cedidos por el USB. Por lo que si se quiere diseñar una aplicación dedicada a la comunicación USB, por medio de la sub-tarjeta que posee el microcontrolador y se desea obtener la alimentación de 3.3 V sin necesidad de diseñar una nueva etapa de potencia, no es recomendable dividir la LPCXpresso en las dos sub-tarjetas.

<sup>1</sup>Actualmente puede encontrarse este sistema de prototipado, tanto con el microcontrolador LPC11U14 como con otros procesadores de la familia LPC: LPC1100, LPC1200, LPC1300, LPC1800, LPC2000, LPC3000 y LPC4000.

## 4.2. Funcionamiento y características

### 4.2.1. Protocolos de comunicación

En la electrónica diseñada pueden distinguirse dos procesos de comunicación: el llevado a cabo entre los propios circuitos integrados que forman parte del diseño y el realizado para la transferencia de información con el exterior.

#### 4.2.1.1. Comunicación interna / protocolo SPI

La comunicación entre la mayoría de los componentes de la placa de control se realiza mediante el protocolo serie SPI (*Serial Peripheral Interface*). Éste es un protocolo maestro-esclavo, serie, semi-duplex y síncrono que utiliza cuatro hilos para realizar el flujo de datos (figura 4.9):

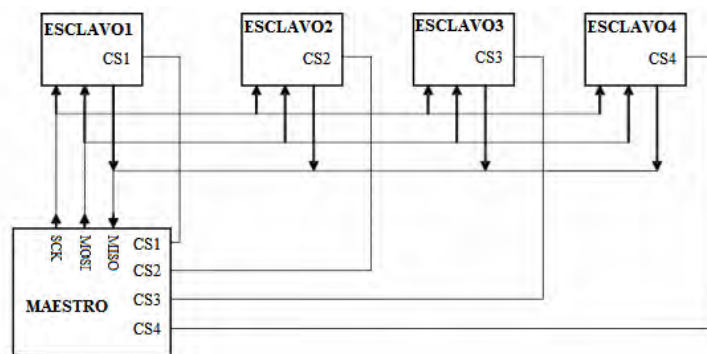


Figura 4.9: Interconexión de componentes en el protocolo SPI.

- **SCLK:** este hilo transporta la señal de reloj generada por el maestro. Ésta es usada como referencia de sincronización por todos los componentes esclavos conectados al bus.
- **MOSI (*Master Out Slave IN*):** este hilo transporta la comunicación serie desde el maestro a todos los esclavos.
- **MISO (*Master In Slave Out*):** este hilo transporta la comunicación serie desde un esclavo al maestro.
- **CS (*Chip Select*):** éste es el hilo de selección, activo bajo.

El intercambio de información a través de este protocolo se realiza sin cabeceras ni identificaciones de trama previos. Esto es, escribiendo el dato directamente en la línea MOSI o MISO. Por lo que su implementación resulta mucho más sencilla que la de otros protocolos serie.

Sin embargo, una de las desventajas presentes en SPI frente a otros protocolos serie como  $I^2C$ , es la necesidad de dedicar un hilo de selección por cada componente

esclavo que se enlace al bus, dado que este protocolo no utiliza cabeceras de identificación. Esto consume muchos recursos espaciales y dificulta el rutado final de la electrónica. No obstante, este protocolo ha sido seleccionado para la comunicación interna, dado que los componentes necesarios para este diseño presentan tanto un consumo mucho más reducido, como unas dimensiones y un coste menores, en comparación con los componentes  $I^2C$ .

En este diseño, se ha utilizado el bus serie SPI para realizar, por un lado, la comunicación entre el microcontrolador y los potenciómetros, enviando el primero el dato con el código de resistencia y por otro, la comunicación entre el microcontrolador y los sensores de temperatura.

#### 4.2.1.2. Comunicación externa / protocolo $I^2C$

En los primeros prototipos se ha utilizado un bus USB (*Universal Serial Bus*) para comunicar la electrónica con el ordenador maestro y enviar desde éste los comandos apropiados (como se explica en la sección 5.1). No obstante, el objetivo final del proyecto es agrupar células de 19 posicionadores conformando la malla de 5000 robots, haciendo llegar a éstos la información en el menor tiempo posible, con el menor consumo, el menor coste y los menores niveles de ruido. Para ello, y debido a que el protocolo USB no soporta la coexistencia de 5000 objetivos y el coste, consumo y tiempo introducidos por multiplexores y HUBs en este protocolo son muy elevados, se ha desarrollado un sistema en árbol mediante el bus serie  $I^2C$ [29] (*Inter-Integrated Circuit*), en el que la información llega hasta cada terminal por una serie de nodos multiplexores y repetidores.

Éste es un protocolo maestro-esclavo, serie, síncrono y duplex que utiliza solamente tres líneas (figura 4.10).

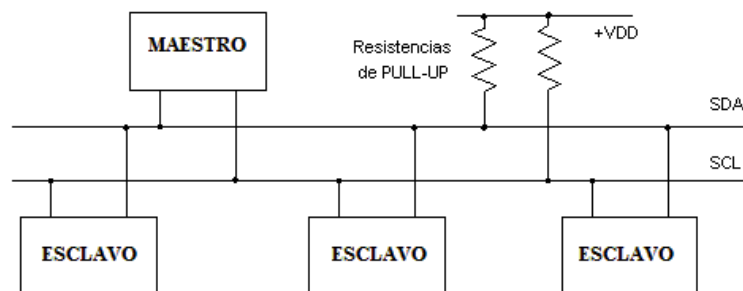


Figura 4.10: Interconexión de componentes en el protocolo  $I^2C$ .

- **SDA:** única línea de comunicación. Ésta transporta los datos serie del maestro al esclavo y viceversa.
- **SCL:** línea de sincronización. Transporta el reloj generado por el maestro.

Cabe destacar, que tanto la línea SDA como SCL se encuentran en *drenador abierto*, por lo que se hace necesario el uso de *pull-ups*. El valor de estas resistencias variará en función de la velocidad y la capacidad total del bus.

A diferencia del protocolo SPI y otros protocolos serie similares, en el bus  $I^2C$  cada componente posee un identificador único que o bien es impuesto por el fabricante, o bien es configurado externamente, sirviéndole al protocolo para rutar los datos. Esto evita el uso de líneas de *chip select* dedicadas a cada dispositivo, con lo que se optimiza el espacio utilizado y se facilita el rutado del sistema. Por otro lado, el protocolo de comunicación serie utilizado por  $I^2C$  es significativamente más complejo que los procedimientos utilizados en otras comunicaciones serie. No obstante, esta complejidad asegura el correcto flujo de información y la serialización.

Este protocolo utiliza siete bits de direccionamiento, por lo que pueden llegar a conectarse hasta 127 dispositivos sin necesidad de multiplexores ni HUBs, siguiendo el siguiente procedimiento:

**En primer lugar se realiza la configuración de la línea.**

1. El bus es activo bajo, por lo que queda libre cuando SDA y SCL están en estado lógico alto.
2. En este estado, cualquier dispositivo puede ejercer de maestro y comenzar una comunicación en el bus.
3. El maestro comienza la comunicación enviando un patrón llamado "start condition". Esto alerta a los dispositivos esclavos, poniéndolos a la espera de una transacción.
4. El maestro se dirige al dispositivo con el que quiere hablar, enviando un byte que contiene los siete bits que componen la dirección del dispositivo esclavo con el que se quiere comunicar. Correspondiéndose el octavo bit con la operación deseada (L/E), lectura=1 (recibir del esclavo) y escritura=0 (enviar al esclavo).
5. La dirección enviada es comparada con la de cada esclavo, el esclavo con coincidencia se considera direccionado como esclavo-transmisor o esclavo-receptor dependiendo del bit R/W.
6. El esclavo responde enviando un bit de ACK que le indica al dispositivo maestro que éste reconoce la solicitud y está en condiciones de comunicarse.

**Seguidamente da lugar el intercambio de información.**

1. El maestro envía la dirección del registro interno del dispositivo que se desea leer o escribir.
2. El esclavo responde con otro bit de ACK.
3. Ahora el maestro puede empezar a leer o escribir bytes de datos. Todos los bytes de datos deben constar de ocho bits, el número máximo de bytes que pueden ser enviados en una transmisión no está restringido, siendo el esclavo quien fija esta cantidad de acuerdo a sus características. Cada byte leído/escrito por el maestro debe ser obligatoriamente reconocido por un bit de ACK por el dispositivo maestro/esclavo.
4. Cuando la comunicación finaliza, el maestro transmite una «stop condition» para dejar libre el bus.

## 4.2.2. Funcionamiento

### 4.2.2.1. Esquema general

Como se expone en la sección 4.3, en este proyecto se han realizado varios prototipos intermedios. No obstante, el funcionamiento general es común para todos.

Este sistema se basa en la recepción y procesamiento de comandos de control (sección 5.3.1.1). Éstos se envían en los primeros diseños por medio de una interface USB, utilizándose en los prototipos finales el protocolo  $I^2C$  para reducir la superficie física empleada y conseguir el rutado de la información hacia las 5000 unidades. No obstante la forma y contenido de los comandos no varía en función del protocolo empleado, por lo que el funcionamiento del sistema es el mismo.

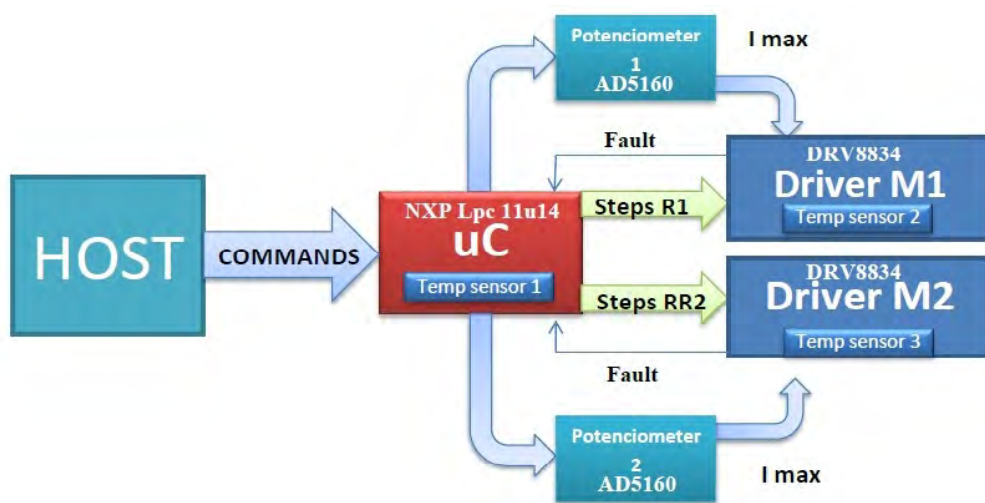


Figura 4.11: Diagrama general del control de la electrónica.

El funcionamiento general del sistema puede dividirse en las siguientes fases:

- **Espera de comandos:** tras el encendido del sistema y la inicialización de las variables empleadas para configurar todos los otros circuitos integrados que forman parte de la electrónica, el microcontrolador entra en modo de espera, reduciendo al máximo el consumo del sistema. Durante esta fase, el microcontrolador escucha permanentemente el terminal de comunicación externo utilizado ( $I^2C$  o USB), comprobando que no contenga información. Tras recibir un nuevo comando, el procesador sale del modo *Sleep* y comienza a procesar la información recibida.
- **Procesamiento de datos:** una vez recibido el comando, el microcontrolador procesa la información actuando en consecuencia:
  - **Posicionado del actuador:** esta acción se divide en dos envíos: la posición absoluta en pasos del motor R1 y la posición absoluta en pasos del motor R2. Tras la recepción de los dos datos, el microcontrolador envía la información necesaria a los drivers de motor para posicionar el actuador, según los algoritmos expuestos en la sección 5.4.

Debido a las características de los drivers utilizados en este proyecto, tras la recepción de un comando de posicionado y el procesamiento de éste, el microcontrolador únicamente configura el sentido de giro del motor (pin 12 / DIR en el encapsulado utilizado) siendo «0» el sentido de las agujas del reloj y «1» el contrario. Tras esto el microcontrolador envía al driver (pin 11 / STEP) un pulso por cada paso a dar (figura 4.12) hasta alcanzar el objetivo.

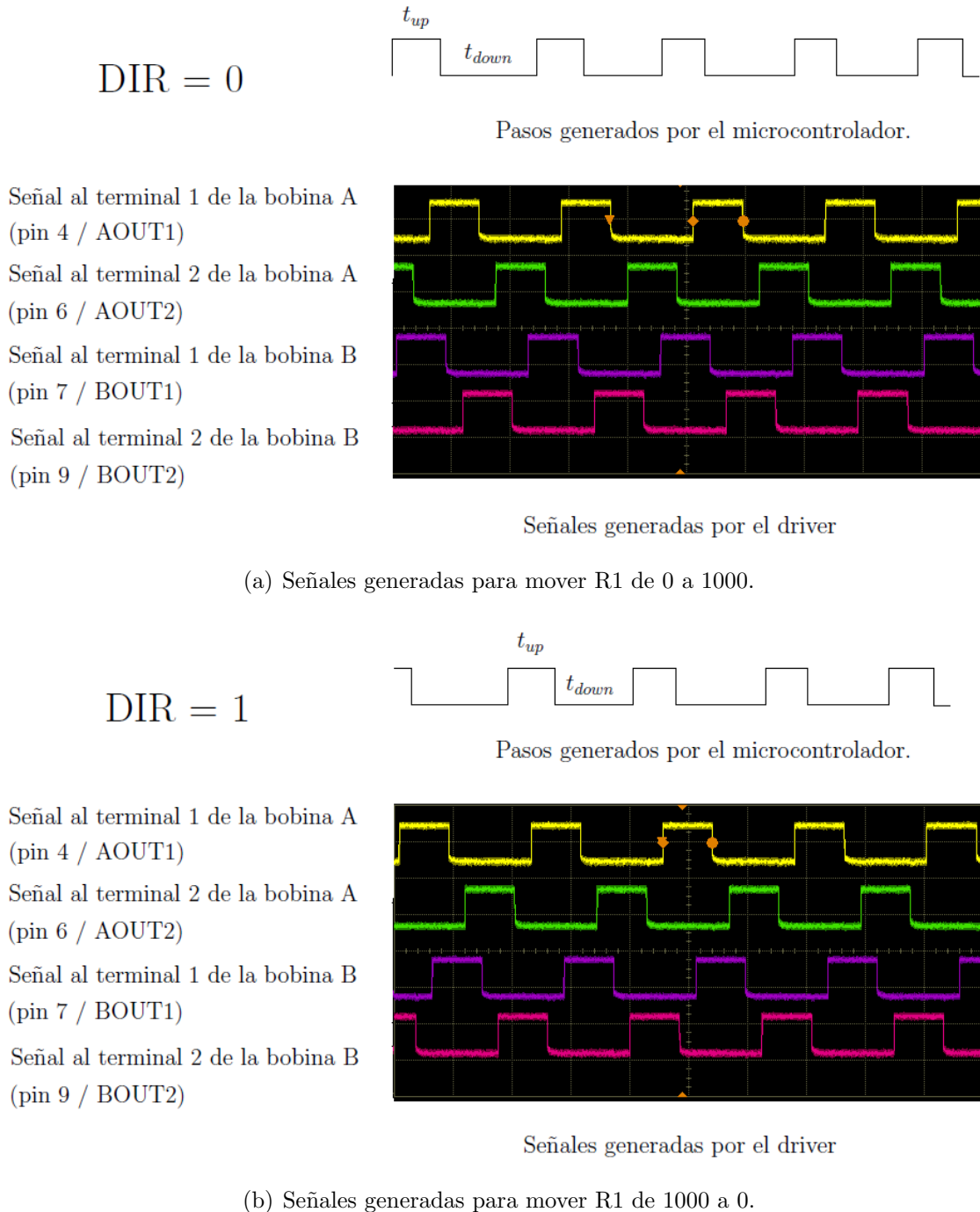


Figura 4.12: Señales implicadas en el movimiento de los motores (5 V/Div).

- **Configuración de la velocidad de posicionado:** con esta acción se configura el tiempo de espera entre los pulsos que el microcontrolador envía a cada driver en forma de pasos (figura 4.12). Siendo la velocidad

de posicionado mayor, cuanto menor sea este tiempo. No obstante, debido a las características físicas de los motores paso a paso utilizados, la velocidad no puede ser incrementada indefinidamente, por lo que ésta ha sido configurada (tras varias pruebas de laboratorio) a un máximo de 1666 pasos/s.

- **Configuración de la modulación de corriente:** en esta acción se envía una constante con valor entre 0 y 255 que se utiliza para configurar la corriente máxima de cada motor. El dato recibido es enviado por el microcontrolador a cada potenciómetro por medio del bus SPI (figura 4.11). Con éste los potenciómetros varían su resistencia interna, modificando por consiguiente la tensión presente en uno de los terminales. Tensión que los drivers utilizan para configurar el valor máximo de la corriente entregada.
- **Configuración del modo de funcionamiento:** esta acción permite seleccionar el modo de funcionamiento deseado; «*Full-chopping*» y «*Semi-Chopping*». El primero realiza una modulación continua de corriente, que permite mantener los motores activos como se ha expuesto en la sección 2.2.3.2. El segundo bloquea la corriente tras cada posicionado, inhabilitando los drivers y reduciendo al máximo la corriente consumida por los motores.
- **Envío de la posición actual:** esta acción es de vital importancia a la hora de mantener guardada la última posición movida. El ordenador maestro guarda las posiciones reales de cada motor para los 5000 actuadores. De esta forma en caso de un corte repentino de corriente o al apagar los dispositivos, los actuadores conocen su posición y pueden mover en consecuencia teniendo una referencia de inicio. Este envío se realiza para evitar incluir memorias ROM en la electrónica, reduciendo una vez más el tamaño de ésta. No obstante, a la hora de realizar este proceso es de vital importancia tener en cuenta la posición real respecto al cero físico (sección 4.2.2.4).

Tras realizar la tarea, el microcontrolador vuelve a escuchar permanentemente el terminal de comunicación a la espera de nuevos comandos, entrando por consiguiente en modo de espera.

- **Lectura de temperatura:** Con un periodo configurado por el sistema maestro, el microcontrolador deja de atender el terminal de comunicación durante unos microsegundos. En este tiempo el procesador sale del modo de espera y realiza una lectura de temperatura, tanto de los motores como de la propia electrónica, guardando el valor leído en una variable específica. Este valor es utilizado para alertar al ordenador maestro de una sobrettemperatura en el sistema, bloqueando los siguientes posicionados del actuador, para evitar así un posible daño. Un aumento en la temperatura de los motores, puede ser producido por un posible fallo en la mecánica que incremente la resistencia al movimiento de éstos. Este tipo de fallos puede ocasionar problemas fatales en el actuador, de no tratarlos convenientemente.

Por otro lado, la lectura de temperatura puede ser ordenada asíncronamente por el ordenador maestro, incluyéndose dentro de un comando. Esta lectura podrá ser utilizada para monitorizar el estado de los 5000 actuadores y estudiar la distribución térmica del plano focal.

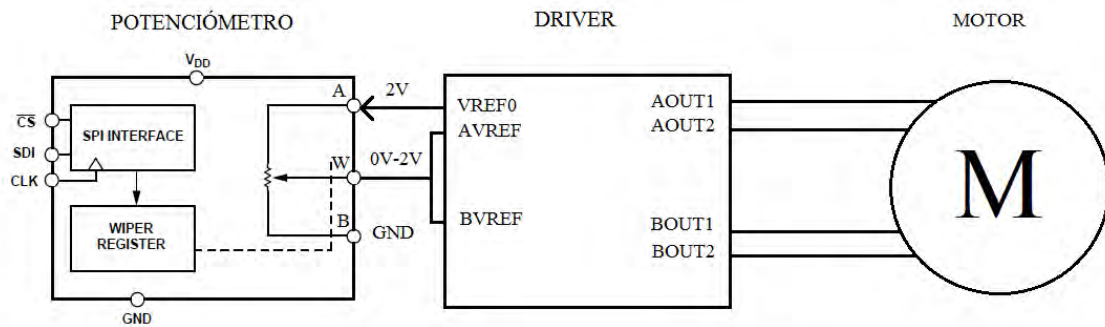


#### 4.2.2.2. Control de la modulación de corriente

Como se ha expuesto en secciones anteriores, la corriente modulada controlará el par generado por los motores y mantendrá éstos calientes evitando problemas de funcionamiento en condiciones climáticas adversas. Esta corriente es generada de manera pulsada por cada uno de los drivers de motor. Éstos entregan más o menos corriente en función de dos variables; la tensión (entre 0 V y 2 V) existente en dos de sus terminales de configuración (AVREF y BVREF, terminales 22 y 23 respectivamente) y la corriente que circula por dos resistencias de referencia conectadas a otros dos terminales de configuración (AISEN y BISEN, terminales 5 y 8 respectivamente), que monitorizan el flujo de corriente de los devanados del motor paso a paso, regulando el valor máximo entregado. Estas dos resistencias tienen un valor fijo que dependerá del prototipo desarrollado ( $5 \Omega$  o  $10 \Omega$  normalmente). Debido a la primera variable de configuración, en los prototipos diseñados se introducen dos potenciómetros digitales (uno para cada driver) que, alimentándolos con una tensión de referencia de 2 V entregada por el driver, generan un valor variable de entre 0 V y 2 V en función del código enviado (figura 4.14).

Finalmente, el terminal de control de cada potenciómetro se conecta a los pines de configuración AVREF y BVREF del driver asignado para cada uno. Esta tensión, junto a la corriente que fluye por las resistencias de referencia de cada driver (figura 4.2) son utilizadas para limitar la corriente que se entrega a cada devanado (A y B respectivamente), permitiendo finalmente el control detallado de la corriente entregada.

En la figura 4.14 puede observarse el flujo de control que configura finalmente la modulación de corriente.



$$V_{WB}(D) = I_A \cdot \left( \frac{D}{256} \cdot R_{AB} + R_W \right) \quad I_{chop} = \frac{xV_{ref}}{5 \cdot R_{ISENSE}}$$

Figura 4.13: Flujo de control de  $I_{chop}$

Donde:

- $I_A$  — Corriente de entrada en el terminal A.
- $D$  — Código serie recibido por el bus SPI (8bits).
- $R_{AB}$  — Resistencia máxima característica del circuito integrado.
- $R_W$  — Resistencia residual ( $60 \Omega$  normalmente).
- $xV_{ref}$  — Tensión de control del devanado (idéntica para ambas bobinas).
- $R_{ISENSE}$  — Resistencia de control de corriente de cada devanado.

En este diseño las resistencias de referencia de cada devanado tienen el mismo valor. Por otro lado, la tensión de control es la misma para cada devanado, por lo que la corriente entregada es idéntica en las dos bobinas de cada motor, y su modificación solo depende del código enviado a través del bus serie SPI mediante el software de control.

#### 4.2.2.3. Prevención de fallos y control de temperatura

Como se ha expuesto en la sección 4.2.2.1 y en secciones anteriores, el control del correcto funcionamiento de cada actuador es de vital importancia, dado que un fallo en la electrónica puede propiciar un fallo de mayor envergadura en la mecánica del robot, causando importantes pérdidas.

Los componentes mas propicios a sufrir fallos o roturas son los drivers de motor, éstos soportan el mayor estrés, dado que tienen que ceder la corriente necesaria a cada motor. Esto en condiciones de mal funcionamiento de los motores o fricción excesiva es éstos, puede llevar a la aparición de picos de corriente muy elevados que afecten al buen funcionamiento de estos componentes. Por ello los DRV8834 además de contar con un control preciso de la señal PWM generada y la corriente entregada, por medio de realimentaciones de las señales internas, cuenta con una señal de fallo (pin 16 en el encapsulado utilizado). Ésta permanece en estado alto en condiciones normales y es llevada a cero cuando se ha producido algún fallo interno en el driver. Esta señal de fallo es constantemente monitorizada por el microcontrolador, que detiene el funcionamiento del robot y alerta de ello mientras ésta se mantenga a cero.

En un caso extremo de rotura de un solo driver en ausencia de este control, el microcontrolador intentaría mover los dos motores, produciéndose el movimiento de únicamente uno de ellos. Sin una correcta detección de este suceso y su consiguiente comunicación al software maestro, éste no tendría en cuenta el problema y seguiría intentando mover el robot considerando que éste se ha posicionado correctamente, lo que produciría una probable colisión con los robots vecinos en futuros posicionados.

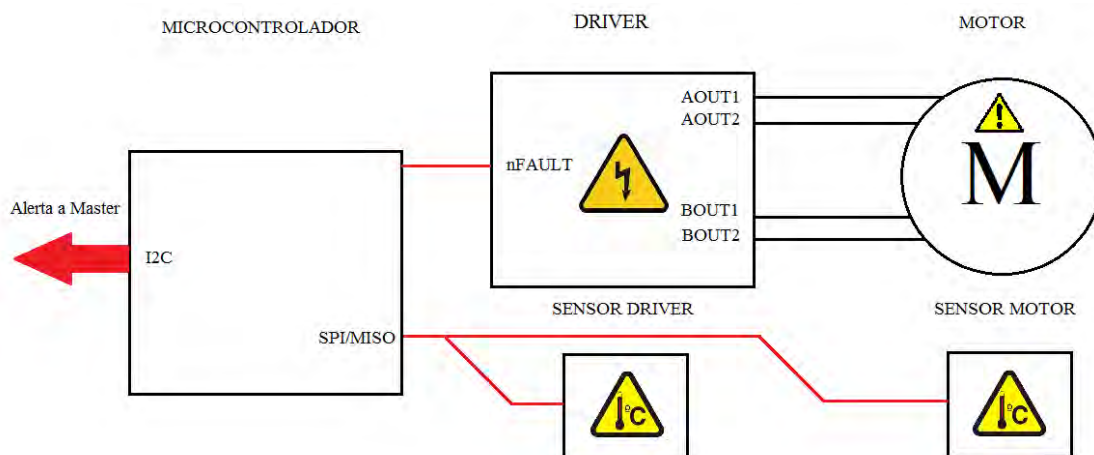


Figura 4.14: Flujo de alerta y detección de fallos.

Por otro lado, como se expone en la sección 4.2.2.1, uno de los indicadores característicos de mal funcionamiento o rotura de los motores es el incremento inesperado de su temperatura. Esto puede ser debido a una colisión no prevista entre posicionadores o a un fallo interno de la mecánica del robot, que produzca una excesiva fricción de los engranajes. A su vez, una rotura en los motores puede producir una sobrecarga en los drivers, incrementando la temperatura de éstos. Por esta razón el sistema realiza constantes mediciones de temperatura y detiene el funcionamiento del robot (alertando consecuentemente) para evitar daños mayores.

#### 4.2.2.4. Control de posición

El software de control y los algoritmos internos programados en el microcontrolador evitan que cualquier fallo en el envío, por parte del usuario, mande al robot a posiciones prohibidas ( $0 \leq R1 \leq 27520pasos$  o  $0 \leq R2 \leq 5120pasos$ , obviando la distancia de seguridad expuesta en la sección 2.1.5), dado que un envío a estas posiciones fuerza al mecanismo interno del robot a moverse contra su límite físico y esto puede ocasionar la rotura del mecanismo. No obstante, cada uno de los robots trabajará durante un periodo muy prolongado de tiempo una vez instalados en el telescopio, por lo que la aparición de errores en el envío de comandos de movimiento, o en la propia mecánica es probable. Por esta razón se hace necesaria la implementación de un sistema hardware que alerte al microcontrolador de la llegada al límite, evitando cualquier movimiento por encima de éste.

Por otro lado, durante el proceso de calibración de cada actuador, es necesario llevar a éstos hasta su límite físico inferior, y avanzando un cierto margen de seguridad (en función de la versión de robot presente), indicar que ese será su cero lógico ( $R1=0$  y  $R2 = 0$ ). Este proceso será automatizado y su realización se llevará a cabo de forma periódica, por lo que una vez más se hace necesaria la implementación de un sistema hardware que notifique de la llegada a un límite físico.

Finalmente, el movimiento de los motores se realiza de forma relativa respecto a un origen definido. Por esta razón, una vez más es imprescindible la introducción de un sistema hardware que indique la posición real del robot (contando en este caso las vueltas que ha dado cada eje motor), dado que un fallo en el cálculo interno de la posición del robot puede llevar a un error en cadena en siguientes posicionados y por consiguiente, en la prevención de colisiones con sistemas vecinos.

Durante las fases de diseño de la electrónica han sido barajados diferentes sistemas de codificación de la posición del robot («*encoders*»).

Por un lado, se ha proyectado la instalación de un sensor *hall* adjunto al eje trasero de cada motor (figura 4.15). Éste iría instalado sobre un PCB flexible, que sería atravesado por el eje del motor, sobre el que se ajustaría un imán de proporciones micrométricas. Este tipo de sensores generan una tensión en su terminal de salida cuando se aproximan a un campo magnético, por esta razón sería posible conocer la posición del motor cada vez que el imán instalado pasase sobre él. Siendo mayor la información de posición proporcionada, cuanto mayor sea el número de sensores existente en cada eje.

Esta primera solución queda en entredicho, dada la dificultad de acceder a los ejes

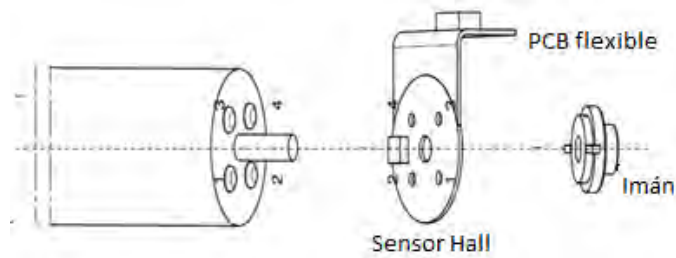


Figura 4.15: Detalle del sistema de codificación de posición.

motores dentro del robot. Por esta razón, actualmente se están implementando sistemas de codificación que, utilizando la inducción de las bobinas del propio robot, analicen la señal en dos hilos conectados a éstas y estudien la posición absoluta del robot respecto a su cero físico. Este trabajo queda fuera de los objetivos de este PFC.

No obstante, todos los prototipos realizados en este proyecto constan de los terminales correspondientes para adaptarse a la instalación de diferentes sistemas de codificación de dos hilos.

### 4.2.3. Alimentación y consumo total teórico

Como se ha expuesto en secciones anteriores, todos los componentes del diseño están preparados para trabajar tanto en modo activo como en diferentes modos de sueño. Por lo que el sistema en su totalidad se sirve de estos modos para funcionar, llevando así el consumo de la electrónica a niveles muy reducidos (tabla 4.9) mientras no se estén moviendo los motores y optimizando el consumo del sistema global (tabla 4.10). No obstante, el microcontrolador LPC11U14 posee un modo de funcionamiento *DeepSleep*, que es el utilizado finalmente en momentos de espera de comandos. De igual modo, los sensores de temperatura, poseen un modo propio *Shutdown*, que reduce el consumo mientras no se reciban órdenes de lectura (tabla 4.8).

Componente	Activo (mA)	Durmiendo (mA)
LPC11U14	2	0.36
DRV8834	2.4	0.0006
TC72	0.25	0.0001
AD5160	0.003	0.003

Cuadro 4.8: Consumo individual de los componentes.

Por esta razón, en el sistema global pueden distinguirse tres modos de funcionamiento (tabla 4.9), como se expone en la sección 4.2.2. El modo de movimiento, en el que los motores se encuentran realizando un posicionado. El modo de lectura de temperatura, en el que los motores están parados y toda la electrónica dormida menos el microcontrolador y los sensores de temperatura y el modo de latencia, en el que todos los componentes se encuentran durmiendo a la espera de nuevos comandos.

Por otro lado, las pruebas realizadas con los posicionadores y el par necesario para minimizar la pérdida de pasos, sitúan el consumo medio de los motores en 200 mW

Moviendo (mA)	Leyendo temperatura (mA)	Durmiendo (mA)
7.556	2.757	0.3675

Cuadro 4.9: Consumo global de la electrónica.

para el motor 1 (encargado de controlar el eje principal R1) y 170 mW para el motor 2 (encargado del control del eje secundario R2). De este modo y dado que se utiliza una tensión de alimentación de 5 V tanto para los motores como para los drivers de motor y 3.3 V para el resto de la electrónica, se consiguen consumos como los mostrados en la tabla 4.10.

Modo	Motor 1	Motor 2	Drivers	Electrónica	Total
Moviendo	200.000	170.000	24.000	9.095	403.095
Leyendo temperatura	0.000	0.000	0.006	9.095	9.101
Durmiendo	0.000	0.000	0.006	1.224	1.218

Cuadro 4.10: Consumo global del posicionador en mW.

### 4.3. Fases del diseño

En el transcurso de este proyecto se han realizado multitud de experimentos y pruebas de funcionamiento (sección 6) que han hecho necesaria la implementación de diferentes cambios y modificaciones en los diseños fruto de los primeros trabajos.

Con el objetivo de organizar y englobar estas mejoras, se han implementado dos prototipos previos a la electrónica definitiva que irá encapsulada en el robot. Éstos han sido diseñados a medida que se hacían necesarios grandes cambios en el diseño existente.

#### 4.3.1. Modelo V.1

Éste es el primer prototipo desarrollado para el proyecto. Este diseño basa su funcionamiento en la placa de desarrollo LPCxpresso que cuenta con un microcontrolador 11U14 del mismo fabricante. Esta placa servirá de nexo de comunicación con el software de control (ejecutado en el ordenador personal del ingeniero que manipule la electrónica)(sección 5). En esta electrónica pueden distinguirse dos partes fundamentales, la placa de desarrollo **LPCXpresso** y la **electrónica diseñada**.

La placa de desarrollo se ha utilizado en esta primera versión con el objetivo de estudiar el rendimiento del microcontrolador elegido, el consumo de éste y su capacidad para manejar la información requerida, sin necesidad de integrarlo completamente en un PCB (evitando el trabajo añadido, innecesario en las primeras etapas). Por otro lado, en esta primera fase de diseño ha sido recomendable el uso de un sistema que poseyese el componente más sensible de la electrónica, estando separado de la fase de «potencia» propiamente dicha. Esto ha sido de vital importancia, dado que el nivel de estrés al que se somete la electrónica en este primer periodo es muy

elevado. Por otro lado, el PCB que posee la electrónica diseñada ha sufrido multitud de modificaciones antes de evolucionar al modelo 2, por lo que ha resultado mucho más sencillo, a la hora de realizar cambios, el hecho de tener esta parte de la electrónica separada del microcontrolador, la fase de depuración y la interface de programación/comunicación.

Cabe destacar que la posibilidad de acceso a las diferentes señales del microcontrolador, sin necesidad de modificar la electrónica diseñada (accediendo a éstas directamente a través de los puertos de la placa de desarrollo) introduce varias ventajas. Por un lado, la posibilidad de estudiar diferentes cambios sin modificar la electrónica diseñada (introducción de nuevos sensores o dispositivos, lectura de puertos no utilizados en el diseño inicial, etc.). Por otro lado, la posibilidad de leer y estudiar las señales implicadas en la comunicación interna, lo que se puede realizar sin dedicar conectores específicos, ahorrando trabajo.

La electrónica diseñada para este modelo como parte del trabajo realizado en el PFC (figura 4.17) posee únicamente los drivers de motor, los potenciómetros digitales y el resto de componentes pasivos que configuran estos dispositivos <sup>1</sup>. Esta electrónica lleva integrada también un sensor de temperatura en la capa posterior. No obstante, éste es un componente vestigial, dado que en esta etapa de desarrollo el control de temperatura no está integrado y la instalación de este sensor como parte de la electrónica se ha realizado únicamente de forma preventiva.

Finalmente, la alimentación de 5 V es extraída del puerto de comunicación USB. Ésta es convertida a 3.3 V en la etapa de potencia de la placa de desarrollo LPCXpresso, tensión que es conducida a la electrónica diseñada para alimentar los diferentes integrados (figura 4.19). Este hecho hace dependiente la alimentación del puerto de comunicación externo, lo que impide la correcta medición del consumo de corriente total de la electrónica. Por lo que en esta etapa de diseño no se han realizado estudios del consumo de la electrónica, focalizando el esfuerzo en la realización de test de funcionamiento y precisión.

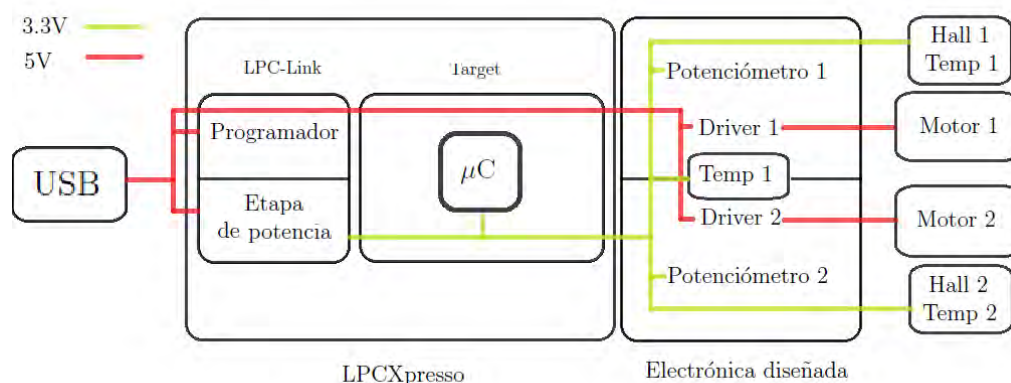
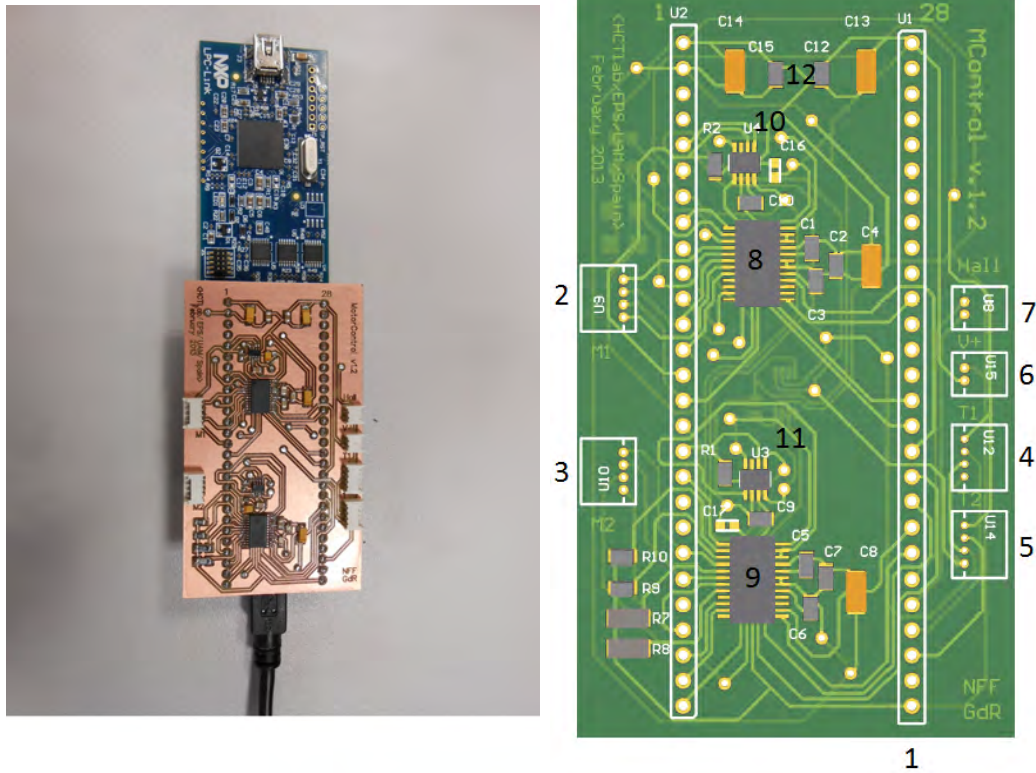


Figura 4.16: Diagrama de alimentación para el modelo v.1.

Este diseño posee los siguientes componentes destacables:

<sup>1</sup>Todos los componentes utilizados en este modelo han sido seleccionados en sus encapsulados de mayor tamaño, dado que la superficie de este diseño no es crítica y éstos presentan una menor dificultad al ser soldados con las herramientas disponibles en el laboratorio.



(a) Prototipo fabricado.

(b) Esquema del PCB.

Figura 4.17: Modelo v.1 de la electrónica.

- **(1) Conectores con la placa de desarrollo LPCXpresso:** éstos son dos conectores rectos de 27 pines, con los que la placa de desarrollo proporciona todas las entradas y salidas disponibles en el microcontrolador:
  - **Conector izquierdo numerado en orden ascendente desde el pin superior (según la figura 4.17):**
    - **Pin 1 / GND:** este pin junto con el pin 27 del segundo conector, une la masa de la electrónica diseñada con la de la placa de desarrollo.
    - **Pin 2 / VM:** este pin proporciona los 5 V necesarios para alimentar los drivers y mover los motores. Esta tensión se obtiene directamente del puerto USB, a través de la LPCXpresso.
    - **Pin 5 / MOSI:** éste es un pin de entrada a la electrónica diseñada correspondiente al protocolo SPI, con el que el microcontrolador (maestro) configura los potenciómetros digitales (esclavos).
    - **Pin 6 / MISO:** éste es un pin de salida de la electrónica diseñada, correspondiente al protocolo SPI, con el que los sensores de temperatura (esclavos) se comunican con el microcontrolador (maestro).
    - **Pin 7 / SCK:** esta es la señal de sincronismo generada por el maestro (microcontrolador) para la comunicación en el protocolo SPI.
    - **Pin 8 / nSLEPM1:** este pin proporciona la señal de configuración del modo de «sueño» del driver 1.
    - **Pin 9 / nENM1:** este pin proporciona la señal de ENABLE para el driver 1.

- **Pin 10 / STEPM1:** este pin proporciona los pasos al driver 1.
- **Pin 11 / DIRM1:** este pin dicta el sentido de giro del motor 1.
- **Pin 14 / CS0:** este pin proporciona el *chip select* del potenciómetro conectado al driver 1.
- **Pin 16 / CS2:** este pin proporciona el *chip select* del sensor de temperatura instalado en la placa.
- **Pin 17 / CS1:** este pin proporciona el *chip select* del potenciómetro conectado al driver 2.
- **Pin 18 / nSLEPM2:** este pin proporciona la señal de configuración del modo de «sueño» del driver 2.
- **Pin 25 / nENM2:** este pin proporciona la señal de ENABLE para el driver 2.
- **Pin 26 / STEPM2:** este pin proporciona los pasos al driver 2.
- **Pin 27 / DIRM2:** este pin dicta el sentido de giro del motor 2.
- **Conector derecho numerado en orden ascendente desde el pin superior (según la figura 4.17):**
  - **Pin 1 / VCC:** este pin proporciona la alimentación de 3.3 V a todos los dispositivos SPI conectados en la electrónica diseñada (potenciómetros, sensores de temperatura y conector para sensores externos). Esta tensión es generada por la LPCXpresso, a través de la alimentación de 5 V conseguida por el puerto USB.
  - **Pin 11 / M0M1:** este pin proporciona la primera de dos señales de configuración del modo de paso para el driver 1 (full step y half step).
  - **Pin 15 / M1M1:** este pin proporciona la segunda de dos señales de configuración del modo de paso para el driver 1.
  - **Pin 16 / CONFIGM1:** este pin proporciona la señal de configuración del modo de funcionamiento del driver 1. Éste puede trabajar en modo *step* o en modo *phase*<sup>1</sup>.
  - **Pin 17 / nFAULTM1:** éste es un pin de entrada al microcontrolador, que el driver 1 pone en estado bajo cuando se produce un fallo interno.
  - **Pin 18 / nFAULTM2:** éste es un pin de entrada al microcontrolador, que el driver 2 pone en estado bajo cuando se produce un fallo interno.
  - **Pin 19 / CONFIGM2:** este pin proporciona la señal de configuración del modo de funcionamiento del driver 2.
  - **Pin 20 / M1M2:** este pin proporciona la segunda de dos señales de configuración del modo de paso para el driver 2.
  - **Pin 21 / M0M2:** este pin proporciona la primera de dos señales de configuración del modo de paso para el driver 2.
  - **Pin 22 / HallM1**<sup>2</sup>: éste es un pin de entrada al microcontrolador,

<sup>1</sup>En esta aplicación se ha elegido el modo *step* de funcionamiento para el control preciso de motores paso a paso. No obstante, el modo *phase* proporciona un control a muy bajo nivel que no es necesario para este tipo de aplicaciones

<sup>2</sup>Como se expone en la sección 4.2.2.4, durante la realización de este proyecto se han barajado multitud de sistemas para codificar la posición real del robot. No obstante, en este periodo no se ha implementado ninguno



- procedente del encoder del motor 1. Éste indica la llegada al cero físico del motor 1.
- **Pin 24 / CS4:** este pin proporciona el *chip select* al sensor de temperatura del motor 1.
  - **Pin 25 / CS3:** este pin proporciona el *chip select* al sensor de temperatura del motor 2.
  - **Pin 26 / Alerta de temperatura:** éste es un pin de entrada al microcontrolador, utilizado para alertar de una sobre-temperatura. El proceso de alerta de sobre-temperatura depende del sensor de temperatura utilizado.
  - **Pin 27 /GND:** este pin junto con el pin 1 del primer conector, une la masa de la electrónica diseñada con la de la placa de desarrollo.
- **(2) Conector del motor 1:** este conector de ángulo recto y cuatro pines (AOUT1, AOUT2, BOUT1 y BOUT2) une la electrónica diseñada con cada uno de los devanados del motor 1.
  - **(3) Conector del motor 2:** este conector une la electrónica diseñada con el motor 2, de la misma manera que el conector 2.
  - **(4) Conector del sensor de temperatura 1:** éste es un conector de ángulo recto que enlaza los cuatro hilos del protocolo SPI (MOSI, MISO, SCK y CS) con el sensor de temperatura integrado en el motor 1.
  - **(5) Conector del sensor de temperatura 2:** este conector enlaza la electrónica diseñada con el sensor de temperatura integrado en el motor 2 de la misma manera que el conector (4).
  - **(6) Conector para la alimentación de los sensores:** éste es un conector de ángulo recto y dos pines (VCC y GND) que lleva la alimentación de 3.3 V a los sensores integrados en los motores.
  - **(7) Conector para el sensor Hall:** éste es un conector de ángulo recto y dos pines (HallM1 y GND) que une el sensor hall utilizado como encoder de posición del robot con la electrónica diseñada.
  - **(8),(9) Drivers de motor:** en este prototipo se han utilizado los drivers de motor DRV8834 de Texas Instruments en su encapsulado PWP de 24 pines (6.6 x 7.9 mm).
  - **(10),(11) Potenciómetros digitales:** en este prototipo se han utilizado los potenciómetros digitales AD5160 de Analog Devices en su encapsulado SOT-23 (3 x 3 mm).
  - **(12) Condensadores de desacoplo:** en esta electrónica se ha utilizado un condensador electrolítico de 10  $\mu$ F en paralelo con uno cerámico de 100 nF para configurar la red de desacoplo de la alimentación general y condensadores de 100 nF para el desacoplo de cada uno de los componentes (en adición a los especificados por cada fabricante). Estos condensadores han sido instalados con el objetivo de filtrar señales no deseadas introducidas por la alimentación (interferencias, picos de tensión, etc.).

## 4.3.1.1. Programación y comunicación

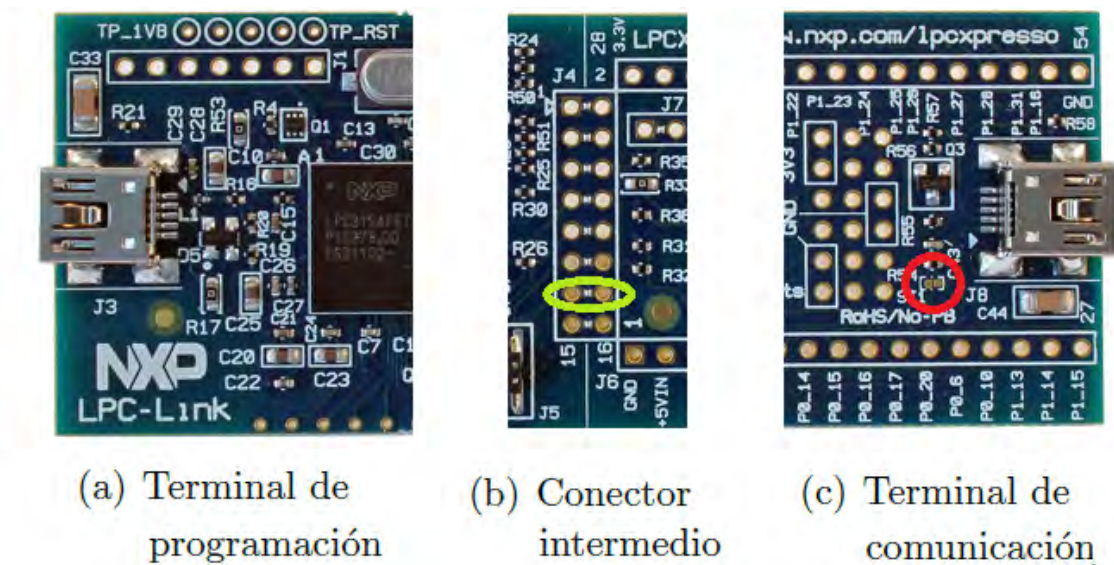


Figura 4.18: Terminales USB disponibles en la placa de prototipado LPCXpresso.

La placa de prototipado LPCXpresso posee dos puertos USB (figura 4.18), uno de ellos dedicado a la programación y el proceso de depuración del microcontrolador y otro utilizado para la comunicación propiamente dicha.

En primer lugar, el conector USB J3 (sub-figura a), es utilizado para programar el microcontrolador. En este proceso, la LPCXpresso se sirve de los componentes instalados en la sub-placa *LPC-link* (figura 4.8), para llevar a cabo el proceso de escritura en la memoria flash que posee el microcontrolador, programando así éste. En este proceso, el microcontrolador es alimentado con los 3.3 V que genera la etapa de potencia en esta sub-placa (figura 4.19).

Por otro lado, el conector USB J8 (sub-figura b), es utilizado como puerto de comunicación serie propiamente dicho. En este caso, la alimentación de 3.3 V también es conseguida a partir de los 5 V cedidos por el USB J8, a través de la etapa de potencia de la sub-placa *LPC-link*. Sin embargo, en este caso es preciso manipular la placa de prototipado, soldando el mini-jumper SJ1 (marcado con un círculo rojo en la figura 4.18), con el objetivo de guiar la tensión de 5 V a esta etapa.

En el caso de querer separar las dos placas, para crear un prototipo más realista del diseño realizado (independizando el microcontrolador de la sub-placa *LPC-Link*), sería preciso ceder al sistema directamente los 3.3 V sin soldar el mini-jumper, o bien, diseñar una etapa de potencia que tomando los 5 V cedidos por el USB, los transforme en 3.3 V a través del conector intermedio que une las sub-placas (marcado con un círculo verde). En este diseño no han sido independizadas las dos sub-placas, dado que se trata de un prototipo inicial, por lo que únicamente se ha soldado el mini-jumper, dejando la transformación de tensión a la etapa de potencia de la sub-placa *LPC-Link*.

### 4.3.2. Modelo V.2

Tras la realización de las diferentes pruebas de funcionamiento (recogidas en la sección 6.1) y la comprobación de la buena respuesta del microcontrolador LPC11U14 a las condiciones de uso (en cuanto a consumo y funcionalidad se refiere), se ha realizado un modelo completo de la electrónica, que incluye tanto el microcontrolador y los dispositivos para configurar éste, como la etapa de potencia que generará los 3.3 V necesarios para alimentar los circuitos integrados del prototipo. Sirviendo de antesala al diseño final que irá integrado en el robot.

La introducción de la etapa de potencia en este prototipo (figura 4.20), ha sido realizada con el objetivo de independizar la alimentación de la electrónica del puerto comunicación (USB), facilitando la medición del consumo de corriente y acercándose al diseño definitivo, en el que la alimentación será entregada desde los nodos superiores independizándola del puerto de comunicación. Por esta razón, han sido introducidos dos jumpers (figura 4.19) con el objetivo de poder medir la corriente que circula a través de la alimentación de 3.3 V (consumo de la mayor parte de la electrónica) y la que circula en la alimentación general de 5 V (consumo total de los motores y drivers). No obstante, en esta electrónica es posible seleccionar entre el puerto USB y un conector independiente como fuentes de alimentación, a través de un *switch* de selección.

Por otro lado, a diferencia del modelo 1 y con el objetivo de comprobar visualmente el correcto flujo de corriente de los componentes que son alimentados con 5 V y 3.3 V, han sido instalados dos LEDs (figura 4.20) de diferente color, que indican que estas dos etapas independientes están funcionando correctamente. Estos diodos forman parte de la etapa de potencia y su consumo ha de ser eliminado a la hora de calcular el consumo total de la electrónica.

Una vez más, en este prototipo se han utilizado los encapsulados de mayor tamaño existentes en los circuitos integrados seleccionados. Por un lado, esto facilita el trabajo manual y ayuda en el proceso de fabricación llevado a cabo en las instalaciones de la Escuela Politécnica Superior (EPS). Por otro, la superficie de la electrónica no es una variable crítica a tener en cuenta en esta etapa de prototipado, por lo que el uso de componentes de mayor tamaño no es un problema.

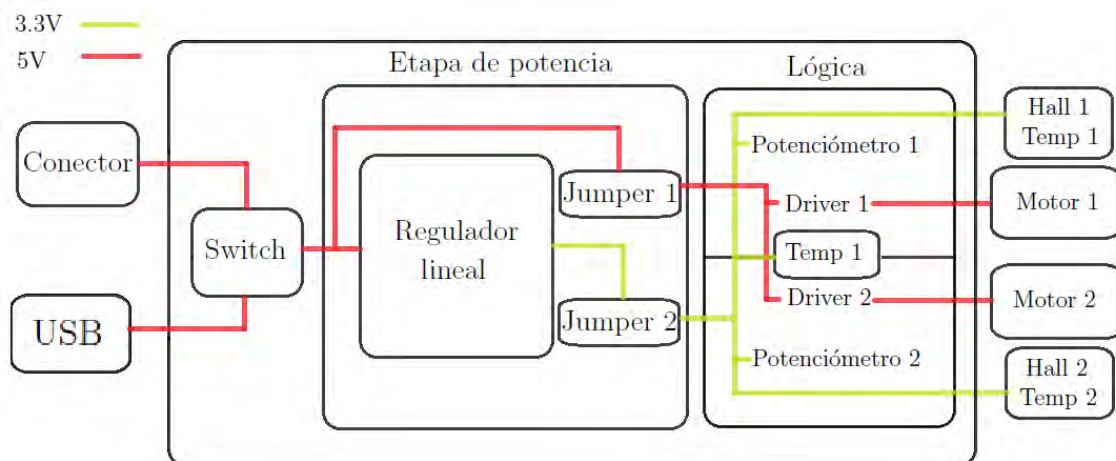


Figura 4.19: Diagrama de alimentación para el modelo v.2.

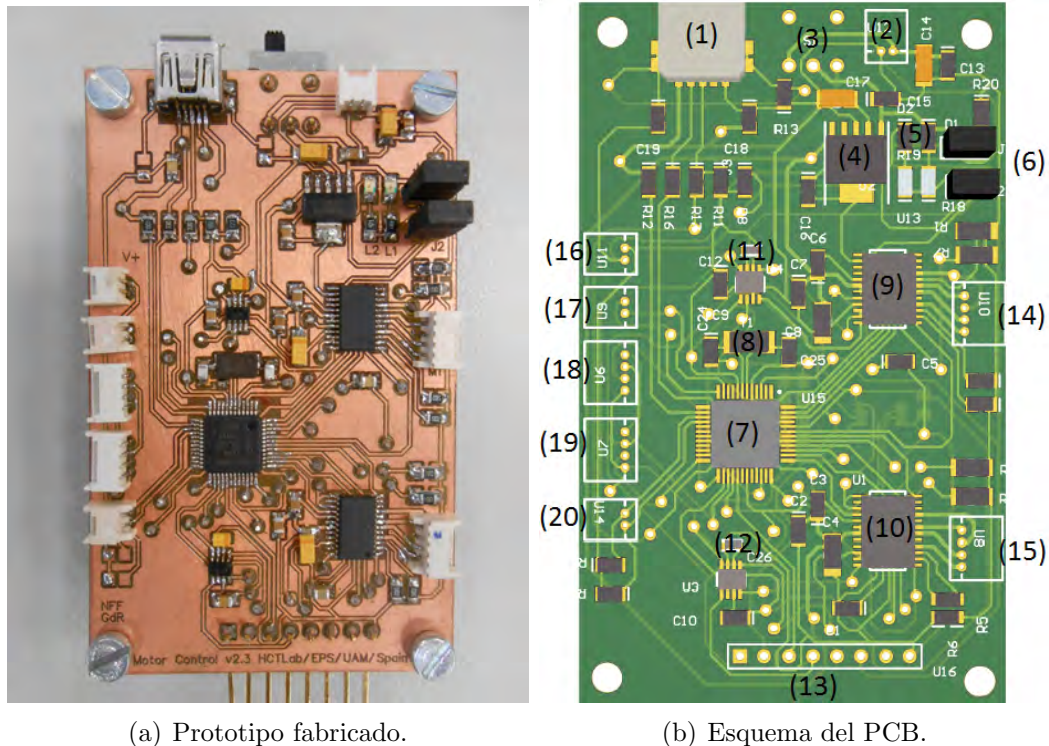


Figura 4.20: Modelo v.2 de la electrónica.

Este diseño posee los siguientes componentes destacables:

- **(1) Conector USB:** en este modelo, éste es el puerto a través del cual se realiza la comunicación con el ordenador. Por él pueden obtenerse los 5 V de alimentación general de la placa.
- **Etapa de potencia (figura 4.21):** esta etapa transforma los 5 V de alimentación general a 3.3 V. Por otro lado, permite medir la corriente que circula a través de la electrónica y comprobar visualmente el correcto funcionamiento del modelo.
  - **(2) Conector de alimentación:** este conector permite independizar la alimentación del puerto de comunicación USB.
  - **(3) Switch:** este componente permite seleccionar entre el conector (1) y el conector (2) como fuentes globales de alimentación.
  - **(4) Regulador lineal:** en este modelo se ha utilizado el regulador lineal REG103 de Texas Instruments, en su encapsulado plástico DCQ. Este circuito integrado transforma los 5 V generales en 3.3 V con los que se alimentan la gran mayoría de los integrados. Este componente presenta una tasa de ruido muy reducida en su salida, además posee una señal de error que alerta de un fallo interno debido, posiblemente, a una sobrecarga.
  - **(5) LEDs:** en este modelo se han utilizado dos LEDs de diferente color (rojo y verde) con el objetivo de comprobar visualmente el correcto flujo de corriente hacia los componentes alimentados con 5 V y hacia los que se alimentan con 3.3 V.

- **(6) Jumpers de medición:** estos componentes permiten conectar un amperímetro en las dos grandes etapas del diseño (componentes alimentados con 3.3 V y con 5 V), con el objetivo de realizar los diferentes estudios de consumo del modelo.

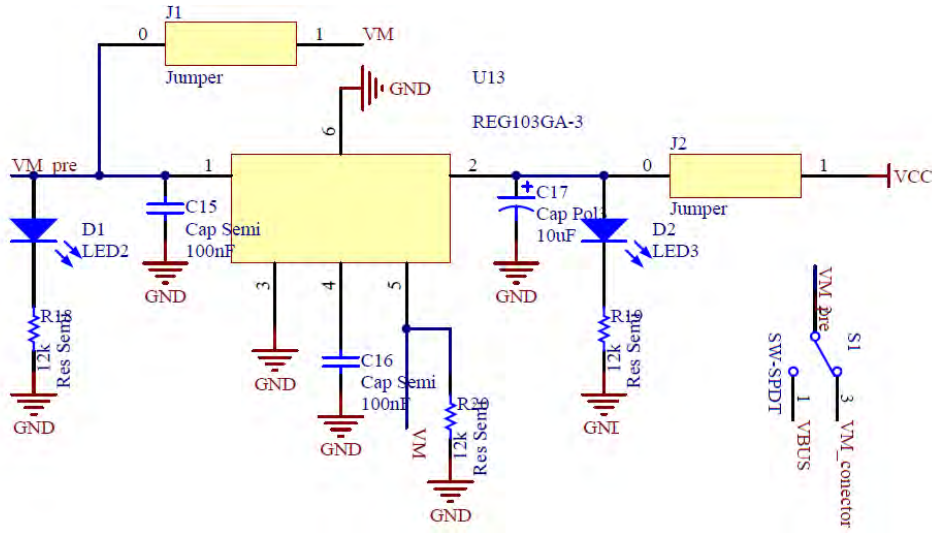


Figura 4.21: Etapa de potencia del modelo v.2 de la electrónica.

Donde:

$VM_{conector}$	—	Tensión de 5 V conseguida en el conector de alimentación.
$VMBUS$	—	Tensión de 5 V conseguida en el puerto USB.
$VM_{pre}$	—	Tensión de 5 V seleccionada por el switch.
$VM$	—	Tensión de 5 V cedida al circuito (trás el jumper de sensado).
$VCC$	—	Tensión de 3.3 V convertida por el regulador lineal.

- **(7) Microcontrolador:** al igual que en el modelo v.1 y dada su buena respuesta a todos los test realizados, en esta fase de la electrónica a vuelto ha utilizarse el microcontrolador LPC11U14 de NXP en su encapsulado LQFP48 (9x9 mm).
- **(8) Cristal:** para configurar el oscilador que genere el sincronismo interno del microcontrolador, formando parte de la etapa de filtrado, se utiliza un cristal de cuarzo de 12 MHz.
- **(9),(10) Drivers de motor:** debido al correcto funcionamiento en el modelo anterior de la electrónica, en este prototipo se han utilizado una vez más los drivers de motor DRV8834 de Texas Instruments en su encapsulado PWP de 24 pines (6.6x7.9 mm).
- **(11),(12) Potenciómetros digitales:** al igual que con los drivers de motor, en este prototipo se han utilizado una vez más los potenciómetros digitales AD5160 de Analog Devices en su encapsulado SOT-23 (3 x 3mm).

- **(13) Conector de programación:** este conector de ángulo recto y ocho pines, es utilizado para programar el microcontrolador a través de la sub-placa LPC-Link (figura 4.22). Siendo los pines de izquierda a derecha según la figura 4.20:
  - **VCC:** tensión de 3.3 V generada por la sub-placa *LPC-Link*.
  - **JTAG<sub>TMS</sub>** : selector de modo de testeo para el protocolo JTAG.
  - **JTAG<sub>TCLK</sub>** : reloj de testeo para el protocolo JTAG.
  - **JTAG<sub>TDO</sub>** : salida de datos de testeo para el protocolo JTAG.
  - **JTAG<sub>TDIX</sub>** : entrada de datos de testeo para el protocolo JTAG.
  - **JTAG<sub>RESETX</sub>** : señal de reset para el protocolo JTAG.
  - **VM:** tensión de entrada de 5 V a transformar en 3.3 V cuando se utiliza la sub-tarjeta *target* para la comunicación USB (figura 4.18).
  - **GND:** tierra.
- **(14) Conector del motor 1:** este conector de ángulo recto y cuatro pines (AOUT1, AOUT2, BOUT1 y BOUT2) es utilizado para enlazar con el motor 1 del robot.
- **(15) Conector del motor 2:** este conector de ángulo recto y cuatro pines es utilizado para enlazar con el motor 2 del robot, de igual modo que el conector (14).
- **(16) Conector de alimentación para los sensores:** este conector de ángulo recto y dos pines (VCC y GND) es utilizado para alimentar el sistema de sensores integrado en el robot.
- **(17) Conector sensor hall:** este conector de ángulo recto y dos pines (HallM1 y GND) es utilizado para enlazar con el sensor hall instalado en uno de los motores del robot.
- **(18) Conector sensor temperatura 1:** este conector de ángulo recto y cuatro pines (MOSI, MISO, SCK y CS) es utilizado para enlazar con el sensor de temperatura instalado en el motor 1, a través del protocolo serie SPI.
- **(19) Conector sensor temperatura 2:** este conector de ángulo recto y cuatro pines es utilizado para enlazar con el sensor de temperatura instalado en el motor 2, a través del protocolo serie SPI, de igual manera que el conector (18).
- **(20) Conector  $I^2C$ :** este conector de ángulo recto y dos pines (SDA y SCL) enlaza la electrónica diseñada con el nodo superior de comunicación a través del protocolo serie  $I^2C$ .

#### 4.3.2.1. Programación y comunicación

En primer lugar, la programación del microcontrolador en este modelo de la electrónica se realiza a través del puerto de programación (13) (figura 4.21). Este puerto es

una réplica del existente en la sub-placa *target* de la tarjeta de prototipado LPCXpresso (contenedora del microcontrolador). Esta arquitectura se ha aprovechado para utilizar la sub-placa *LPC-Link* a la hora de programar el microcontrolador, sin necesidad de tener que hacerlo de forma separada antes de instalarlo en el PCB, teniendo que utilizar programadores de mayor coste y complejidad. Por un lado, esto permite reprogramar el microcontrolador todas las veces que sea necesario sin modificar la electrónica. Por otro lado, el hecho de contar con la sub-tarjeta *LPC-link* separada, permite reprogramar la electrónica, utilizando únicamente un cable USB conectado al ordenador del programador.

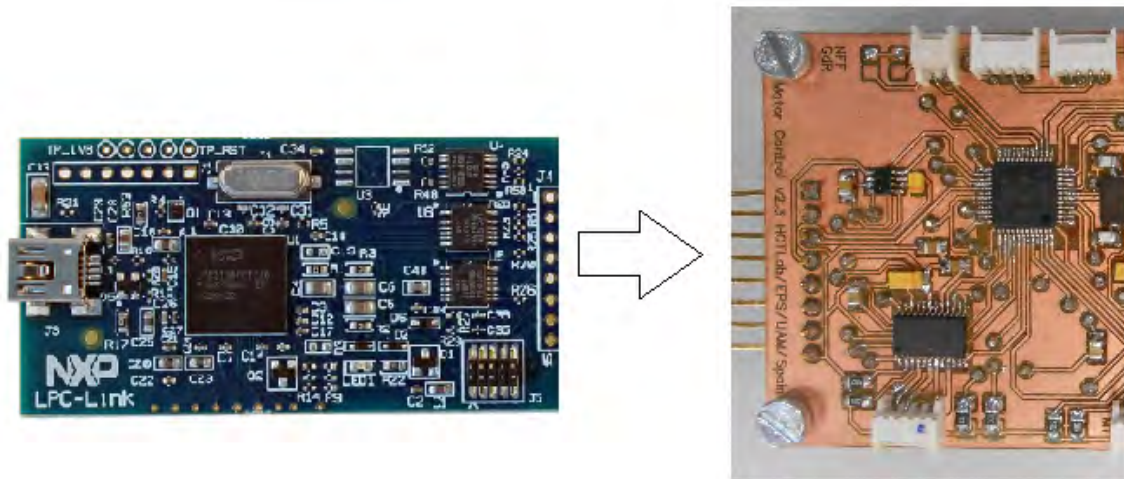


Figura 4.22: Conexión de programación para el modelo v.2 de la electrónica

Debido a las características del diseño, a la hora de programar este prototipo es preciso conectar la alimentación externa, dado que a través de ésta se consiguen los 3.3 V necesarios para alimentar el microcontrolador.

Por otro lado, en este prototipo la comunicación se realiza a través del puerto USB integrado en la electrónica diseñada. En este caso, la tensión de 3.3 V destinada a alimentar la mayoría de los componentes, se consigue a través de la etapa de potencia instalada en la propia electrónica, por lo que, a diferencia del prototipo v.1 no es necesario tener la sub-tarjeta *LPC-Link* conectada en el proceso de comunicación.

### 4.3.3. Modelo V.3

Durante la realización de este PFC se han diseñado los prototipos previos al diseño definitivo de la electrónica, el definitivo no ha sido construido por el momento, dado que los esfuerzos del equipo español están siendo focalizados en la realización de pruebas de consumo y funcionalidad que constaten el correcto comportamiento de éstos en todo tipo de situaciones. No obstante, este diseño ha de poder encapsularse en el chasis de los posicionadores, por lo que su superficie queda restringida a unos 8x50 mm. Por esta razón, para este último diseño se seleccionarán los encapsulados de menor tamaño para los componentes elegidos y se diseñará el circuito impreso en varias capas, reduciéndose la superficie dedicada al rutado de las pistas.

No obstante, el trabajo del equipo español implicado en este proyecto está encami-

nado a construir este último prototipo a corto plazo. Esto permitirá realizar los test de mayor importancia, en los que estará implicada una célula de varios posicionadores con la electrónica definitiva integrada, simulando el funcionamiento real del sistema una vez embarcado en el telescopio. De nuevo este trabajo esta fuera de los objetivos marcados en este PFC.

La figura 4.23 muestra un posible diseño final de esta electrónica :

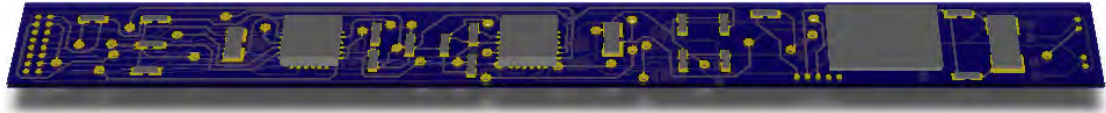


Figura 4.23: Imagen virtual de la versión final de la electrónica.

## 4.4. Conclusiones

La división del desarrollo de este proyecto en prototipos ha facilitado la realización de innumerables test de funcionamiento (sección 6.1), optimizando el proceso de diseño. Éstos han sido realizados en los diferentes centros de investigación del equipo español para el proyecto BigBOSS, sirviéndose del modelo v.1 para realizarlos. Esta división ha permitido acumular sucesivas mejoras a medida que se avanzaba de modelo, incluyendo la resolución de los fallos o cuestiones a mejorar detectados durante la fase de prueba. Se ha optimizado en gran parte la funcionalidad de la electrónica, gracias a este trabajo colaborativo.



# Capítulo 5

## Diseño software

*He viajado a lo largo y ancho de este país y hablado con la gente más preparada y puedo asegurar que el procesamiento de datos es una moda pasajera que acabará antes de fin de año*

Editor responsable de libros de negocios de Prentice Hall, 1957.

**Resumen:** por un lado, en este capítulo se describen las herramientas de control software diseñadas para el manejo del robot. Por otro, se exponen los diferentes algoritmos programados en el microcontrolador, que manejarán todas las funcionalidades de los prototipos desarrollados.

### 5.1. Descripción del sistema de control software

La malla de 5000 robots que componen el plano focal del telescopio NOAO será controlada por un ordenador central integrado en las instalaciones del observatorio. Éste será el encargado de enviar las órdenes de posicionado a los nodos intermedios de comunicación, calibrar los posicionadores por medio del estudio de las imágenes obtenidas por la cámara CCD de calibración, ejecutar los algoritmos predictores de colisiones y reposicionar los actuadores para evitar choques entre unidades vecinas. Por otro lado, este ordenador monitorizará las variables fundamentales de cada actuador (temperatura y posición) según sea necesario y detendrá el funcionamiento de aquellas unidades dañadas o que hayan alertado de un posible fallo, evitando la rotura de las mismas.

En esta etapa del diseño se pretende simular la funcionalidad de este computador central, reduciendo el rango de actuación a una única unidad robótica (con el objetivo de integrar una célula de actuadores a corto plazo). Para ello se ha diseñado un software de control que maneja todas las funcionalidades del hardware descritas en secciones anteriores, controlando el movimiento de un robot como lo haría la unidad central del proyecto finalizado. Este software otorga al ingeniero que manipule el robot (con la electrónica diseñada) todas las herramientas de control necesarias para la realización de multitud de test de funcionamiento (sección 6.1), a través de una interface agradable e intuitiva que facilite al usuario el uso del prototipo, sin la necesidad de conocer el funcionamiento interno de éste.

Por otro lado, en el desarrollo de los prototipos expuestos se han diseñado un conjunto de algoritmos que componen el código embarcado en el microcontrolador, encargados de hacer de interface entre el software de control y el posicionador. Este código incluye el conjunto de mecanismos para el control interno que manejan al detalle el posicionado del robot. Estos algoritmos son la base del proyecto y forman la parte más importante y compleja de éste.

## 5.2. Especificaciones y modos de funcionamiento

Durante el desarrollo del proyecto, debido a la necesidad de éste de ser testado en diferentes entornos (HCTLab, Instituto de Astrofísica de Andalucía [30], LBNL [31] de la Universidad de Berkeley, etc.) han sido diseñados distintos métodos de posicionado para facilitar la comprensión al usuario, aumentando la utilidad de la herramienta. Por esta razón, el software diseñado puede dividirse en tres modos de posicionado en función de su nivel de abstracción. No obstante, el objetivo final de éstos no es otro que mover el robot a una posición absoluta, en pasos de los ejes R1 y R2.

- **Manejo en (R1,R2) (sección 5.3.3):** éste es el método de más bajo nivel, en el cual el usuario puede mover el robot seleccionando directamente la posición absoluta del eje R1 (entre 0 y 27520 pasos) y del R2 (entre 0 y 5120 pasos). Este modo permite manejar el robot directamente sin ningún nivel de abstracción manipulando el movimiento real de los dos motores. Por ello, está reservado para los ingenieros más próximos al proyecto, con el objetivo de realizar los test más básicos en los que se estudia la posible pérdida de pasos y el error en cada uno de los ejes.
- **Manejo en (x,y) (sección 5.3.2):** a la hora de realizar los test de mayor relevancia en la Universidad de Berkeley, fue diseñado un método de posicionado por eje cartesiano. Este método, supone el centro de un eje cartesiano bidimensional en el centroide frontal del posicionador. Con este centro como referencia, se realizan todos los movimientos vectorialmente según coordenadas (x,y). Este modo de posicionamiento será el integrado en el sistema definitivo, siendo los pasos en R1 y R2 transparentes para el astrónomo que configure el software de la unidad central.

Cabe destacar la realización de varios algoritmos de transformación capaces de realizar las conversiones entre coordenadas cartesianas (x,y) y vectores de pasos (R1,R2), gracias a los cuales es posible alternar entre los modos R1R2 y xy indiferentemente, sin que esto afecte al correcto funcionamiento del posicionador.

- **Interoperatividad y modo automático (sección 5.3.2.2):** con el objetivo de aunar los dos modos de funcionamiento y realizar diferentes pruebas en las instalaciones del *Lawrence Berkeley National Laboratory*, se ha desarrollado un modo automático de prueba que interopere ente la rutina de prueba de este laboratorio y el software de control diseñado en este proyecto. Este modo es una rutina automática capaz de mover tanto en modo (R1,R2) como en modo (x,y), siguiendo las órdenes del software de prueba de Berkeley.

### 5.3. Software maestro, interface de usuario

Este software sirve de interface de usuario para el control y prueba de un único posicionador. Se ha desarrollado con el objetivo de poder realizar todos los test de funcionamiento tanto de la electrónica, como de la mecánica del robot propiamente dicha, mediante una interface de usuario agradable y de fácil manejo capaz de ser ejecutada en cualquier sistema operativo Windows.

La aplicación ha sido programada para Windows en el lenguaje C# (C «Sharp»), mediante *Windows Framework 4*. Éste es un lenguaje de programación orientado a objetos, con el que se puede interoperar con todas las aplicaciones de windows (audio, video, comunicación con periféricos<sup>1</sup>, conectividad, etc.). Por un lado, éste es un lenguaje muy versátil, que permite realizar todo tipo de aplicaciones para este sistema operativo, consiguiendo entornos gráficos de gran calidad, agradables para el usuario y altamente interactivos. No obstante, la programación de esta aplicación en un lenguaje reservado a Windows, impide que usuarios de otros sistemas operativos (Mac, Linux, Gnu, etc.) puedan ejecutarlo en sus entornos, restringiendo el uso de esta aplicación a usuarios de Windows.

Requisitos de ejecución para la aplicación:

- Sistema operativo Windows 2000 o superior.
- Última versión de Microsoft Framework instalada.
- Instalación correcta de los controladores de dispositivos genéricos HID.
- Poseer al menos un puerto USB 2.0 (o superior)
- Poseer al menos una pantalla.
- Poseer al menos un teclado.

Esta aplicación ha sido testeada en ordenadores con sistemas operativos: Windows 2000, Windows XP, Windows Vista y Windows 7. Además, dadas las propiedades de interoperatividad del lenguaje utilizado entre diferentes sistemas operativos de Microsoft, esta aplicación puede ser ejecutada también en plataformas móviles como Windows mobile.

Para diseñar la aplicación se ha utilizado el entorno de programación de Microsoft *Visual Studio* en su versión de 2010. Esta plataforma permite compilar, enlazar y ejecutar el código escrito, dando acceso a todas las librerías fundamentales de Windows. Por otro lado, ésta es una herramienta muy potente a la hora de depurar el código, pudiendo ejecutar éste línea a línea desde una posición específica, facilitando la comprensión de errores o fragmentos de código con funcionamiento inesperado. Por otro lado, esta herramienta posee un modo de funcionamiento gráfico, que aun no habiendo sido utilizado en esta aplicación, facilita el diseño de interfaces gráficas, para usuarios no familiarizados con este tipo de lenguajes orientados a objetos.

---

<sup>1</sup>Una de las características por las que se ha utilizado este lenguaje de programación para crear la aplicación, ha sido la gran cantidad de documentación que existe en la red sobre aplicaciones dedicadas a la comunicación y específicamente a la comunicación USB

### 5.3.1. Funcionamiento general

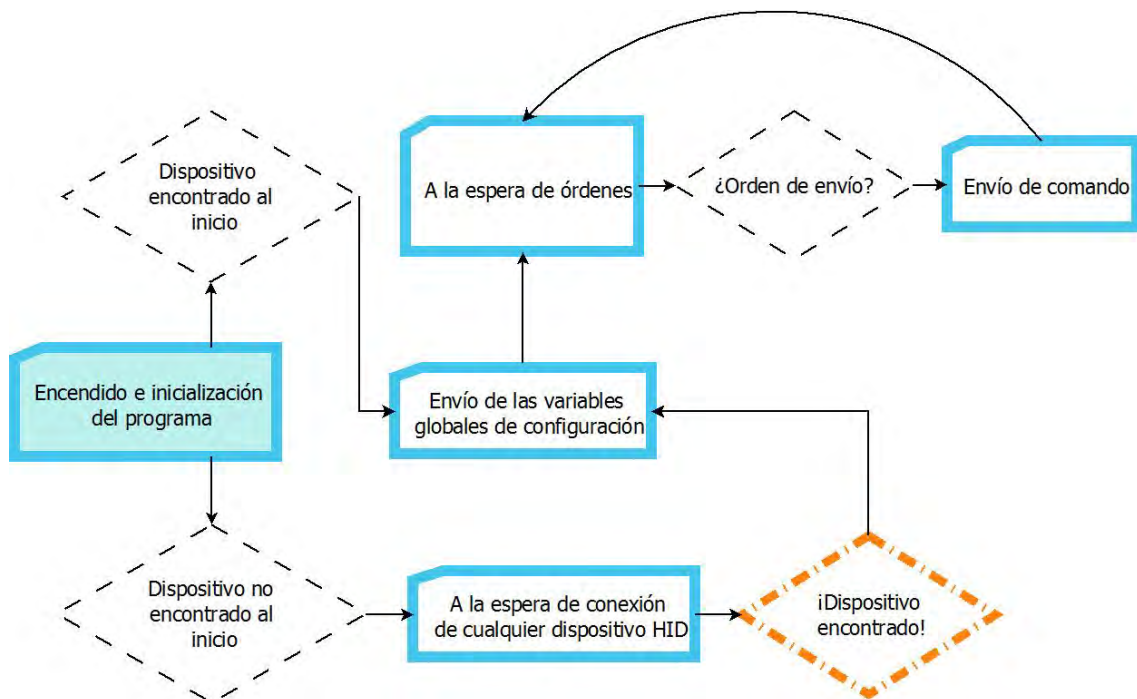


Figura 5.1: Diagrama de flujo del funcionamiento general del sistema.

Como se ha expuesto en secciones anteriores éste es un sistema basado en el envío y recepción de comandos, los cuales configuran la electrónica y envían órdenes de posicionado para mover el robot. Este software se sirve del grafo representado en la figura 5.1 para configurar el sistema y realizar el flujo de comunicación de la siguiente manera:

1. **Encendido e inicialización del programa:** tras el encendido de la aplicación, se inicializa la interface gráfica y se da valor a las variables globales usadas en el programa. Una vez iniciada la aplicación se comienza a *escuchar* los puertos USB con el objetivo de encontrar uno en el que haya conectado algún dispositivo *HID* (Human Interface Device). Estos son dispositivos genéricos de interface con el usuario como ratones, memorias externas, etc. Así ha sido definido el sistema hardware para facilitar su uso.

Los dispositivos USB son identificados a partir de dos códigos hexadecimales, el VID (*Vendor ID*) y el PID (*Product ID*). Estos códigos no son únicos, por lo que son representativos de determinado conjunto de dispositivos, pudiéndose repetir. No obstante, en el hardware diseñado estos códigos pueden ser reconfigurados desde el microcontrolador, en el caso de estar usando otro dispositivo con los mismos identificativos.

Finalmente, cada vez que se conecta un dispositivo HID a un puerto USB, el programa compara sus identificativos con los preconfigurados para el hardware. De haber un acierto se pasa al estado (2) del grafo, *Envío de las variables globales de configuración*. De no ser así, la aplicación se mantiene a la escucha de los puertos USB pasando al estado (5).

2. **Envío de las variable globales de configuración:** tras encontrarse la electrónica de control conectada al puerto USB, existiendo una coincidencia

entre los identificativos programados en la aplicación y los propios del hardware, se pasa a configurar éste.

En este estado se envían los siguientes parámetros de configuración al microcontrolador:

- **Posición actual:** este parámetro indica la última posición a la que se envió el robot. Esta posición es guardada en un fichero denominado «*current position.txt*» cada vez que se realiza un movimiento. Pudiendo ser recuperada y enviada al posicionador en caso de apagado de la electrónica o desconexión de ésta. Este envío se realiza en dos partes: el envío de la posición del eje R1 y el envío de la posición del eje R2. Estos parámetros son guardados en sendas variables por el microcontrolador y utilizados como referencia de movimiento para cada posicionado.

El documento «*current position.txt*» puede ser modificado, haciendo creer al robot que se encuentra en una determinada posición. Esto puede ser de utilidad en el estudio de errores de posicionado, dado que con la existencia de alguna pérdida de pasos, podemos recalibrar el robot «reenviándole» a la posición teórica verdadera, en la que el actuador debería encontrarse en ausencia de tal pérdida.

Este documento posee una columna y tres filas, representadas por los últimos valores alcanzados de R1, R2 y el valor de la posición inducida RR2.

- **Modo de «chopeo»:** Como se ha explicado en la sección 4.2.2.1 este parámetro selecciona el modo de modulación de corriente *full chopping* o *semi chopping*. Este modo podría configurarse en el microcontrolador por defecto, no obstante se ha preferido configurarlo desde el inicio mediante la aplicación de usuario, dado que esto introduce mayor versatilidad.

Tras la configuración de los parámetros fundamentales, el programa pasa al estado (3).

3. **A la espera de órdenes:** en este estado, el programa permanece a la espera de cualquier acción del usuario que implique la transferencia de información con el hardware (posicionado del robot, envío de parámetros de configuración, etc.). Cuando ésta es realizada y se precisa un envío se pasa al estado (4).
4. **Envío de comandos:** en este estado se realiza el envío de un comando al microcontrolador, éstos tienen el formato presentado en la sección 5.3.1.1. Tras el envío de un comando, el programa vuelve a permanecer a la espera de cualquier acción del usuario que implique la transferencia de información, volviendo de nuevo al estado (3).
5. **A la espera de conexión de cualquier dispositivo HID:** si en el encendido de la aplicación no se ha detectado ningún dispositivo HID cuyo VID y PID coincidan con los preconfigurados en el microcontrolador para la electrónica de control, el programa permanece escuchando el puerto USB, notificando de cualquier hallazgo.

Tras ser encontrado el dispositivo, la aplicación pasa al estado (2) configurando el sistema hardware con el envío de los parámetros fundamentales.

### 5.3.1.1. Comandos empleados

Como en la gran mayoría de los protocolos basados en mensajes, la estructura de los comandos empleados en este sistema consta de un byte de cabecera y dos bytes de datos (figura 5.2).



Figura 5.2: Paquete de datos empleado.

En función de su cabecera, estos comandos realizan las siguientes tareas:

- **E3:** este comando es el encargado de enviar la posición absoluta (en pasos) en la que se desea posicionar un motor. Siendo el dato una posición entre 0 y 65536.

El posicionamiento del robot se realiza moviendo simultáneamente los dos motores, para ello se envían dos comandos de esta clase, el primero conteniendo la posición absoluta del motor R1 y el segundo la del motor R2. Tras estos envíos se procede al posicionamiento del actuador (como se expone en la sección 5.4).
- **CC:** este comando se emplea para configurar el modo de modulación de corriente en función de su dato. Configurándose el modo «*full-chopping*» cuando el dato es «1», y el modo «*semi-chopping*» cuando éste es «0».
- **A1:** este comando envía la posición actual del motor R1. Siendo el dato una posición en pasos entre 0 y 27520 para el motor empleado.
- **A2:** este comando envía la posición actual del motor R2. Siendo el dato una posición en pasos entre 0 y 5120 para el motor empleado.
- **AC:** este comando especifica el tiempo que el microcontrolador espera entre pulsos enviados a cada driver. Con éste es posible configurar la velocidad de los motores, siendo la misma en ambos. El dato enviado en este comando es un valor entre «1» y «10». Correspondiendo a tiempos entre pasos de 0.1 ms a 1 ms respectivamente.
- **BB:** este comando especifica el tiempo que el posicionador ha de esperar entre movimientos, para el modo *cluster* de posicionado (sección 5.3.3.2).
- **F2:** este comando especifica el código que el microcontrolador enviará al potenciómetro 2 (encargado de configurar el driver del motor 2), configurando la corriente máxima entregada al motor 2. Este código se relaciona con la resistencia interna del potenciómetro según la expresión 4.1.3. El dato transmitido contiene un valor entre 0 (resistencia mínima) y 255 (resistencia máxima).
- **F1:** este comando especifica el código que el microcontrolador enviará al potenciómetro 1 (encargado de configurar el driver del motor 1), configurando la corriente máxima entregada al motor 1. Este código se relaciona con la resistencia interna del potenciómetro según la expresión 4.1.3. El dato transmitido contiene un valor entre 0 (resistencia mínima) y 255 (resistencia máxima).

El entorno gráfico de esta aplicación puede subdividirse en tres grandes partes, representadas por tres pestañas de control: control en modo R1R2, control en modo xy y Settings (además de una pestaña de ayuda y otra de contacto). A continuación se expone cada una de ellas:

### 5.3.2. Control xy

Este modo de control fue diseñado bajo las exigencias del Lawrence Berkeley National Laboratory (LBNL) en Berkeley, quienes estimaron de mayor utilidad la existencia de un modo de control cartesiano. En esta primera pestaña pueden distinguirse tanto el modo cartesiano de funcionamiento como el modo automático de interoperatividad con el sistema maestro del LBNL.

(1) Selector del modo de modulación de corriente.

(2) Sub-pestaña de envío cartesiano con selector de velocidad (pasos/segundo).

(3) Configuración de los parámetros usados en el modo automático.

(4) Botón de inicio del modo automático.

(5) Información de la posición actual ((x,y) en mm y (R1,R2) en pasos y grados).

(6) Sub-pestaña de información de errores y alertas en el modo automático.

(7) Envío a cero del eje R1, el R2 o ambos.

Figura 5.3: Pestaña para el control en eje cartesiano.

Por otro lado, y al igual que en todas las pestañas de esta aplicación, puede destacarse una ventana de monitorización, en la que se presentan todos los envíos realizados (mostrándose el comando completo enviado), así como todas las alertas y errores aparecidos durante la ejecución. En las siguientes secciones se expone el funcionamiento de la aplicación por medio de los datos mostrados en esta ventana.

### 5.3.2.1. Envío unitario de posición

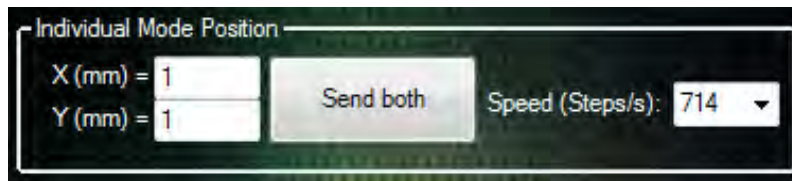


Figura 5.4: Detalles de la sub-pestaña *Individual Mode Position*.

Esta funcionalidad permite enviar al robot a una posición absoluta (x,y) en milímetros respecto a un centro de coordenadas marcado en el centroide frontal del posicionador. Este tipo de movimiento es traducido a una posición absoluta (R1,R2) mediante una transformación trigonométrica (dado que el envío se realiza con posiciones absolutas de R1 y R2). A la hora de realizar esta transformación es preciso conocer el valor exacto de los brazos posicionadores L1 y L2, éstos son considerados inicialmente perfectos (L1=L2=3.464 mm) en ausencia de calibración. Los detalles del funcionamiento de los algoritmos son mostrados en el anexo C

En la sub-pestaña mostrada en la figura 5.4, existen cuatro elementos característicos: dos «*textbox*» en los que puede introducirse la posición absoluta en milímetros de x e y a la que quiere mandarse el robot, un selector de velocidad en el que es posible configurar la velocidad de movimiento del posicionador (velocidad aplicable a toda la pestaña xy) y un botón de envío.

Tras pulsar sobre el botón SEND, la orden de posicionado constituida por el comando de configuración de la velocidad y dos envíos secuenciales con la posición R1 y R2 (tras la realización del algoritmo de transformación), son enviados al hardware.

En la figura 5.5 puede observarse el resultado sobre la ventana de monitorización, de un envío a la posición x=1 mm e y=1 mm:

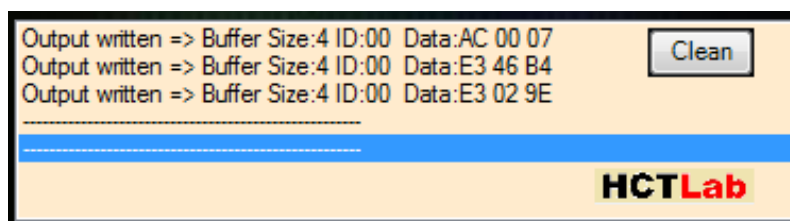


Figura 5.5: Ventana de monitorización tras el envío a x=1 mm e y=1 mm.

En este ejemplo puede observarse el envío del comando «AC» con el dato «7», que representa los 0.7 ms en que la señal de pasos, generada por el microcontrolador, está a «1» y a «0». Consiguiéndose una velocidad de 714 pasos por segundo (velocidad óptima configurada por defecto). Por otro lado pueden verse los comandos de posicionado «E3» con el dato en hexadecimal correspondiente a la posición absoluta de R1 (primer comando «E3») y a la posición absoluta de R2 (segundo comando «E3») a la que quiere enviarse el robot.



### 5.3.2.2. Modo automático

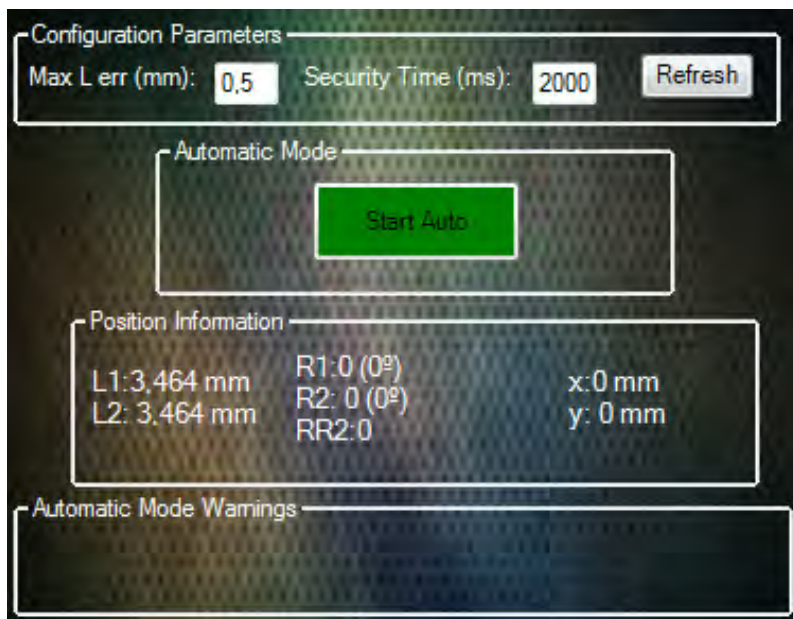


Figura 5.6: Detalle de las sub-pestañas implicadas en el modo automático.

Como se ha explicado en la sección 5.2 se ha incluido en el software un modo automático de funcionamiento, que interopere entre el posicionador y un software maestro en la Universidad de Berkeley. Este algoritmo se ha diseñado con el objetivo de realizar pruebas masivas de larga duración en las que el posicionador esté moviéndose durante horas. Estas pruebas se realizarán automáticamente mediante el envío de comandos de posicionado por el software maestro de Berkeley.

En esta sección se diferenciará entre **software maestro** y **software de control**, siendo el primero el diseñado por el LBNL de Berkeley y el segundo, el software de control desarrollado para este proyecto.

Este modo permite la realización de calibraciones periódicas y automáticas en las que se modifique el valor interno de la longitud de los brazos L1 y L2 (dato de principal relevancia a la hora de realizar las transformaciones de xy en R1R2 y viceversa).

En estas sub-pestañas del programa pueden distinguirse cuatro bloques de elementos característicos: una zona de emisión de alertas y errores, una zona de información general, un botón de comienzo/parada del algoritmo y dos cuadros de texto. En el primero de ellos puede especificarse el tiempo de seguridad, en milisegundos, que el programa debe esperar tras cada iteración del algoritmo automático (figura 5.7). Éste es usado para evitar el solapamiento de accesos a los ficheros entre el software maestro y la aplicación de control. Por otro lado, cabe destacar un segundo cuadro de texto en el que puede especificarse el error máximo permitido en la calibración de los brazos posicionadores.

La comunicación entre el software maestro de Berkeley y el software de control se realiza mediante la transferencia de mensajes por medio de la lectura y escritura en ciertos documentos de texto. A continuación se expone la estructura interna de las líneas incluidas en cada uno de ellos:

- **calibration.txt:** en este fichero se guardan los datos de calibración. Éste es escrito por el software maestro y leído por el software de control. Posee la siguiente estructura interna:

— *Timestamp Index LENGTH\_R1 Dato1 LENGTH\_R2 Dato2* —

**Timestamp:** marca de tiempo identificativa de cada calibración.

**Index:** índice de 0 a  $2^{64}$  que numera la calibración.

**LENGTH\_Rx:** texto de ayuda a la identificación visual del valor que le sigue (no es interpretable).

**Dato1:** dato correspondiente a la longitud en milímetros del brazo L1.

**Dato2:** dato correspondiente a la longitud en milímetros del brazo L2.

- **motion\_status.txt:** este fichero posee el estado de movimiento del robot. Éste es escrito por el software de control y leído por el software maestro. Posee la siguiente estructura interna:

— *Timestamp Index Status* —

**Timestamp:** marca de tiempo identificativa de cada movimiento del robot.

**Index:** índice de 0 a  $2^{64}$  que numera el movimiento.

**Status:** estado en el que se encuentra el robot, puede ser:

- «moving»: robot en proceso de posicionado.
- «stopped»: robot quieto a la espera de órdenes.
- «outofrange»: posición a mover fuera del rango válido del robot.

- **move\_cmd.txt:** este fichero guarda el comando de movimiento. Éste es escrito por el software maestro y leído por el software de control. Posee la siguiente estructura interna:

— *Timestamp Index Comando Dato1 Dato2* —

**Timestamp:** marca de tiempo identificativa de cada comando enviado.

**Index:** índice de 0 a  $2^{64}$  que numera el comando enviado.

**Comando:** tipo de movimiento a realizar.

**Dato1:** dato correspondiente a R1 (pasos) o x (mm).

**Dato2:** dato correspondiente a R2 (pasos) o y (mm).

Los comandos escritos pueden ser del tipo:

- «abs\_R1R2»: movimiento a una posición absoluta de R1 y R2 en pasos.
- «abs\_xy»: movimiento a una posición absoluta de x e y en milímetros.
- «rel\_dR1dR2»: movimiento relativo de R1 y R2 en pasos.
- «rel\_dx dy»: movimiento relativo de x e y en milímetros.

El *timestamp* sigue las recomendaciones del organismo regulador IEEE para este tipo de marcas. Éste posee 15 caracteres, en los que se plasma el año, mes, día, hora, minuto y segundo, que marca el reloj del sistema, en el que se está ejecutando el programa cuando el marcador es escrito. Por ejemplo, siendo las 13:45 y 10 segundos del 17 de Julio de 2013, el timestamp marcaría: 20130617T134510. Siendo la «T» el separador entre hora y fecha.

A la hora de escribir en estos ficheros se añade una nueva línea. De esta manera se mantiene un historial de actuación del posicionador con el que pueden realizarse multitud de estudios, entre ellos la posibilidad de conocer movimientos no dados debido al intento de ser posicionado fuera de rango. Por otro lado, estos ficheros pueden ser utilizados para representar diagramas de movimiento teórico frente a las posiciones realmente alcanzadas, dotando al ingeniero de herramientas de estudio adicionales.

Este modo de funcionamiento automático da comienzo tras ser pulsado el botón «Start Auto» y permanece ejecutándose hasta que éste vuelve a ser pulsado. En él, el software maestro escribe cada cierto tiempo comandos de posicionado, siguiendo ciertas bases de datos con experimentos a realizar. Por otro lado, realiza ciertas calibraciones periódicas por medio del estudio de imágenes frontales obtenidas durante los experimentos. Este modo sigue el siguiente algoritmo:

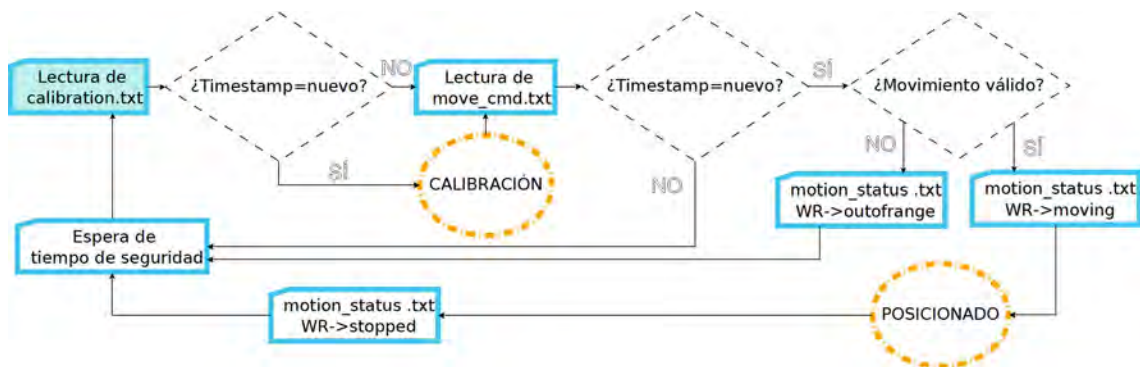
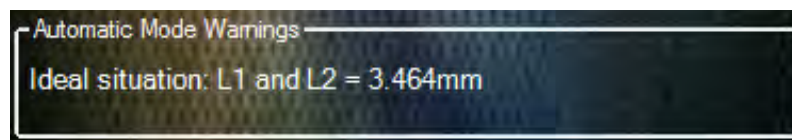


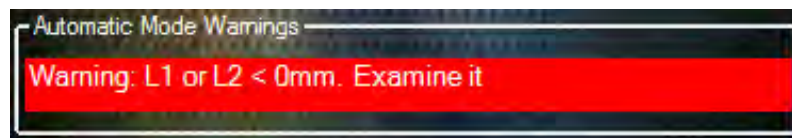
Figura 5.7: Diagrama de flujo del funcionamiento modo automático.

1. **Inicialización de variables:** una vez encendido el modo automático, si ésta es la primera vez en ser puesto en marcha en esta sesión, se escribe «stopped» en el fichero «motion\_status.txt» y se inicializan los valores de los brazos posicionadores, tomando antes de cualquier calibración el valor 3.464 mm para ambos brazos.
2. **Lectura de «calibration.txt»:** tras la inicialización de las variables globales y la primera escritura en «motion\_status.txt», se hace una primera calibración del sistema. En ésta se realiza una lectura de la última línea del fichero «calibration.txt». En primer lugar se compara el timestamp escrito, con el último valor de éste (siendo «0» si es la primera vez que se lee). En caso de coincidencia o de estar el fichero vacío, no se realiza calibración alguna. En caso de no haber coincidencia y el fichero no estar vacío, se dan nuevos valores a la longitud de los brazos posicionadores, asignándoles el valor leído en esta última línea.

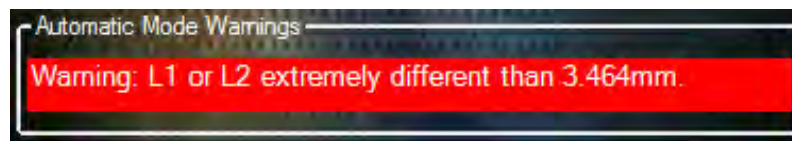
Durante el proceso de calibración pueden suceder tres situaciones destacables (figura 5.8). En primer lugar, si el valor de los brazos posicionadores leído es igual a  $L1=-1$  y  $L2=-1$ , se considera una situación ideal, en la que ambos brazos miden 3.464 mm. En segundo lugar, si en el valor leído existe un error mayor que el especificado en el cuadro de texto respecto a la situación ideal, se alerta de ello y se sale del modo automático, evitando posibles acumulaciones excesivas de errores. En tercer lugar, si el valor leído para una de las dos longitudes es negativo y distinto de -1, se alerta de ello y se sale del modo automático, dado que existe un error interno de medición y esto puede causar daños fatales en el sistema.



(a) Situación ideal.



(b) Calibración negativa.



(c) Margen de error excedido.

Figura 5.8: Tres posibles advertencias mostradas tras la calibración.

3. **Lectura de «move\_cmd.txt»:** tras la calibración se pasa a la lectura del fichero «move\_cmd.txt», con el objetivo de realizar un posicionado. De igual modo que en la lectura del fichero de calibración, se compara el timestamp con un valor anterior de éste (o «0» de ser la primera vez que se ejecuta el modo). En caso de coincidencia no se realiza ninguna acción. En caso de ser distintos, se procede a la comprobación del correcto envío. Si la posición a la que se desea mandar el robot está dentro de los márgenes aceptados ( $0 \leq R1 \leq 27520$  y  $0 \leq R2 \leq 5120$ ), se escribe en el fichero «motion\_status.txt» el estado «moving», se realiza el envío y se escribe de nuevo sobre el fichero de estado: «stopped». De no ser así se notifica debidamente mediante la escritura en el fichero de estado con el mensaje «outofrange».

Tras este proceso se vuelve al estado (2) y se continua el algoritmo hasta que el usuario detenga su proceso volviendo a pulsar el botón de inicio. El cual, tras comenzar la ejecución toma un color rojo y cambia su texto interno de «Start Auto» a «Stop Auto».

Este modo de funcionamiento ha sido de gran utilidad a la hora de ejecutar multitud de test (sección 6.1) en los que se realizaban miles de posicionados en un periodo de tiempo muy reducido (del orden de días). Estos test han sacado a la luz pequeños

errores de concepto, fallos de código y limitaciones de la mecánica del robot, que no habrían podido detectarse de no haberse realizado test masivos y esto habría sido imposible de no existir un software maestro y un modo de funcionamiento que automatizase el proceso de prueba.

### 5.3.2.3. Envío a cero

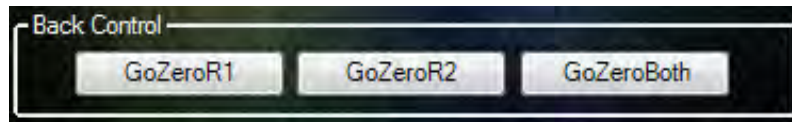


Figura 5.9: Detalle de la sub-pestaña de envío a cero.

En esta sub-pestaña se facilita el envío a cero de cada uno de los ejes posicionadores, mediante tres botones de envío (figura 5.9); uno para el eje R1, otro para el R2 y otro para ambos.

Esta herramienta, facilita el estudio de la repetitividad del movimiento de cada eje, mediante el envío a cero desde distintas posiciones. Por otro lado, en las primeras etapas del diseño, antes de definir correctamente los algoritmos de posicionado, apareció una pérdida de pasos en determinadas ocasiones. Esto generó un error entre la posición actual en la que el software consideraba al actuador y la posición real en la que éste estaba. Por esta razón la herramienta de ida a cero facilitó el trabajo, dado que tras la aparición de esta situación, era posible llevar ambos ejes a cero y, manipulando el fichero «current\_position.txt», configurar esa posición en función de los pasos perdidos.

### 5.3.2.4. Control del modo de chopper

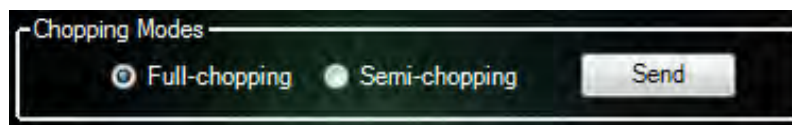


Figura 5.10: Detalle de la sub-pestaña de control del modo de modulación de corriente

En esta sub-pestaña se facilita la selección del modo de modulación de corriente entre «full-chopping» o «semi-chopping» (figura 5.10) con el envío de un comando «CC». Aquí pueden destacarse tres objetos: dos botones radiales con los que seleccionar el modo y un botón general de envío con los que mandar la configuración al microcontrolador.

Como se ha explicado en la sección 4.2.2.1, estos modos permiten mantener los devanados del motor con corriente permanentemente, sin necesidad de mover el actuador, controlando la temperatura de reposo de éste o únicamente en periodos de posicionado, reduciendo el consumo al mínimo.

### 5.3.3. Control R1R2

Este modo de funcionamiento está reservado para trabajar al nivel más bajo, mediante el control directo de los pasos en ambos motores. Este sistema permite independizar el movimiento de los motores, pudiendo mover únicamente uno de ellos, con el objetivo de estudiar el comportamiento de cada uno de los ejes del posicionador. Esto facilita el estudio del comportamiento de la mecánica y permite detectar errores a un nivel más detallado que el modo cartesiano.

(1) Selector del modo de modulación de corriente

(2) Sub-pestaña de envío de pasos con selector de velocidad (pasos/segundo).

(3) Sub-pestaña de envío en modo cluster.

(4) Configuración de los límites de movimiento y monitorización de la posición actual.

(5) Envío a cero del eje R1, el R2 o ambos.

Figura 5.11: Pestaña para el control en modo R1R2.

Al igual que en todas las pestañas, en este apartado existe una ventana de monitorización en la que se presentan todos los envíos realizados y los errores y advertencias aparecidos, facilitando al usuario el estudio de la transferencia de información entre la aplicación y la electrónica. Además, existe un botón de limpiado con el que éste puede borrar todos los mensajes anteriores que no interesen.

Por otro lado, en esta pestaña también existe un sub-apartado de envío a cero y uno de control del modo de modulación de corriente, con la misma funcionalidad que los expuestos en la sección 5.3.2.

### 5.3.3.1. Envío unitario de posición

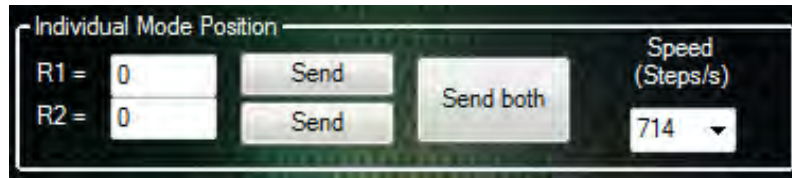


Figura 5.12: Detalle de la sub-pestaña de envío unitario R1R2.

Esta sub-pestaña facilita el posicionado del robot, controlando únicamente la posición absoluta de los ejes R1 y R2. Ésta es utilizada en movimientos y estudios puntuales, en los que se necesite posicionar una única vez el robot, teniendo que realizar cada orden de posicionado manualmente.

En esta parte pueden destacarse seis elementos (figura 5.12): dos cuadros de texto en los que especificar la posición absoluta en pasos de los ejes R1 (entre 0 y 27520) y R2 (entre 0 y 5120), un selector de velocidad (cuya velocidad es aplicable a toda la pestaña R1R2), y tres botones de envío de información. Por un lado, dos para realizar el movimiento de un único eje, por otro, uno para posicionar ambos motores.

Al igual que en el modo cartesiano, el envío de cada mensaje «E3» con la información de posición es precedido de un comando «AC» con la especificación de la velocidad de posicionado. En este caso, para posicionar un único eje se envían dos comandos «E3», uno con la posición a la que se quiere enviar el eje deseado y otro con la posición actual del eje que no se desea mover (respetando el protocolo de envío y moviendo únicamente un eje).

En la figura 5.13, puede observarse la información presentada en la ventana de monitorización, tras el envío de R1 al paso «5», estando R2 en el paso «1» y configurándose una velocidad de 714 pasos/segundo.

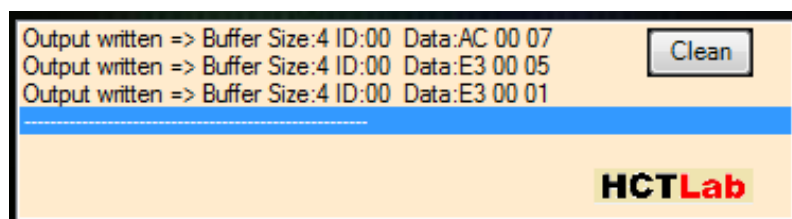


Figura 5.13: Ventana de monitorización tras el movimiento de R1.

Cabe destacar que el movimiento del motor R1, induce internamente movimiento en el eje R2 (sección 2.1.5), por lo que aun no enviando información de posicionado a este segundo eje, si se mueve el eje R1, el motor se moverá debido a los algoritmos internos de control de posición que contrarrestan la inducción interna del eje R2. No obstante, si la desinducción se realiza correctamente, externamente el robot solo posicionará el eje deseado.

### 5.3.3.2. Envío en modo *cluster*



Figura 5.14: Detalle de la sub-pestaña de envío en modo cluster.

En esta sub-pestaña se facilita el envío de un conjunto de posicionados, mediante la selección rápida de un máximo de cuatro movimientos o la utilización de una tabla *Excel* para realizar un amplio grupo de ellos. Con este método es posible realizar pequeñas pruebas con el objetivo de estudiar ciertos tipos de movimientos.

Esta sub-pestaña puede dividirse en dos partes: el envío en modo cluster y el envío masivo (figura 5.15). En la primera de ellas pueden destacarse cuatro conjuntos de elementos característicos: un grupo de ocho cuadros de texto con los que especificar las posiciones a las que queremos mover el robot, cuatro *checkboxes* con los que seleccionar los movimientos que queremos enviar, un cuadro de texto en el que especificar el tiempo que se desea esperar entre cada envío (segundos) y un botón general de envío de los movimientos. En la segunda parte podemos destacar dos elementos: un botón de búsqueda y selección, que despliega un buscador con el objetivo de seleccionar el archivo Excel a leer y un botón general de envío. Por último, cabe destacar una columna de texto en la que se especifica la posición real a la que se moverá el eje R2 tras la inducción en cada envío (posición RR2). Ésta se refrescará tras la finalización de cada envío.

El envío masivo mediante archivo Excel se realiza siguiendo la misma nomenclatura que la utilizada por el software maestro en el modo automático. En este caso se utilizan únicamente tres columnas, una con el tipo de movimiento a realizar (*abs\_R1R2*, *abs\_xy*, *rel\_dR1dR2* y *rel\_dxdy*) y las otras dos con el dato propiamente dicho (siguiendo las especificaciones del modo automático).

En este caso, a la transferencia del cluster de comandos «E3» de posicionado, le precede un comando «AC» con la información de velocidad de movimiento y un comando «BB» con la información del tiempo de espera entre movimientos. Este tiempo tiene importante relevancia, dado que durante la realización de determinados test, es necesario fotografiar en detalle el resultado del posicionado tras cada movimiento y esto, en función del sistema utilizado para realizar esta toma de imágenes, se lleva a cabo en mayor o menor tiempo, necesitando un periodo de pausa antes de realizar el siguiente movimiento.



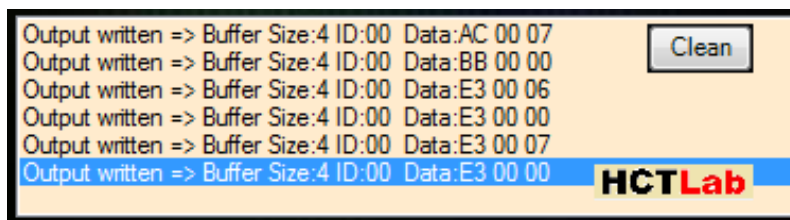


Figura 5.15: Ventana de monitorización en un envío en modo cluster.

En la figura 5.15 puede observarse el detalle de la ventana de monitorización tras el envío en cluster de dos posiciones, (6,0) y (7,0). En este ejemplo se distingue como se envía en primer lugar la información de velocidad (siendo el dato «7» el correspondiente a 714 pasos/segundo), en segundo lugar el comando con la información de espera entre movimientos (en este caso no se espera) y finalmente el cluster con los dos movimientos (R1, R2, R1, R2). En este sistema toda la información se envía en bloque, siendo el firmware programado en el microcontrolador el encargado de gestionarla e independizarla, realizando cada uno de los movimientos por separado y esperando el tiempo marcado por el comando BB.

### 5.3.3.3. Configuración de los límites del sistema

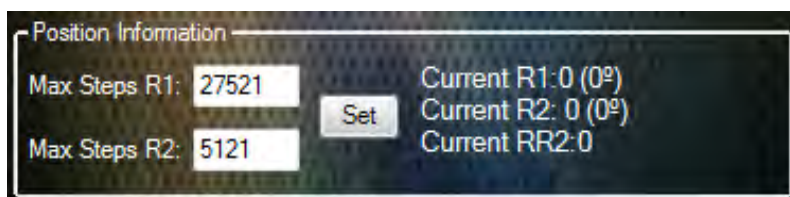


Figura 5.16: Detalle de la sub-pestaña de configuración de los límites del sistema.

Una funcionalidad importante en esta aplicación, es la reconfiguración interna de los parámetros del robot, con el objetivo de prevenir la necesidad de reprogramación del código tras una modificación del posicionador. Estos parámetros no son más que los pasos máximos de cada eje, siendo estos en la versión de robot actual: R1 = 27520 pasos y R2 = 5120 pasos. Éstos son parámetros globales y su reconfiguración afecta tanto al modo de funcionamiento xy como al R1R2.

Previo envío de cualquier comando de posicionado, ya sea en modo xy como en R1R2, se realiza una comprobación del dato, asegurando que éste se encuentre dentro de los parámetros definidos. De no ser así, se advierte adecuadamente (sección 5.3.5) y se cancela el envío del comando. Una mala configuración de estos parámetros permitiría moverse al robot en posiciones prohibidas, forzando la mecánica de este y rompiéndola en caso de sobrepasar notablemente el límite.

En esta sub-pestaña (figura 5.16) pueden distinguirse cuatro elementos fundamentales: dos cuadros de texto en los que se puede especificar el nuevo límite del sistema, un botón de reconfiguración y un texto informativo con la posición actual en pasos y grados de cada uno de los ejes.

### 5.3.4. Settings

Mediante esta pestaña se controlan todos los parámetros que no están relacionados directamente con el movimiento del posicionador, como la lectura de temperatura, el control de la modulación de corriente, o la selección de determinados algoritmos de aumento de precisión.

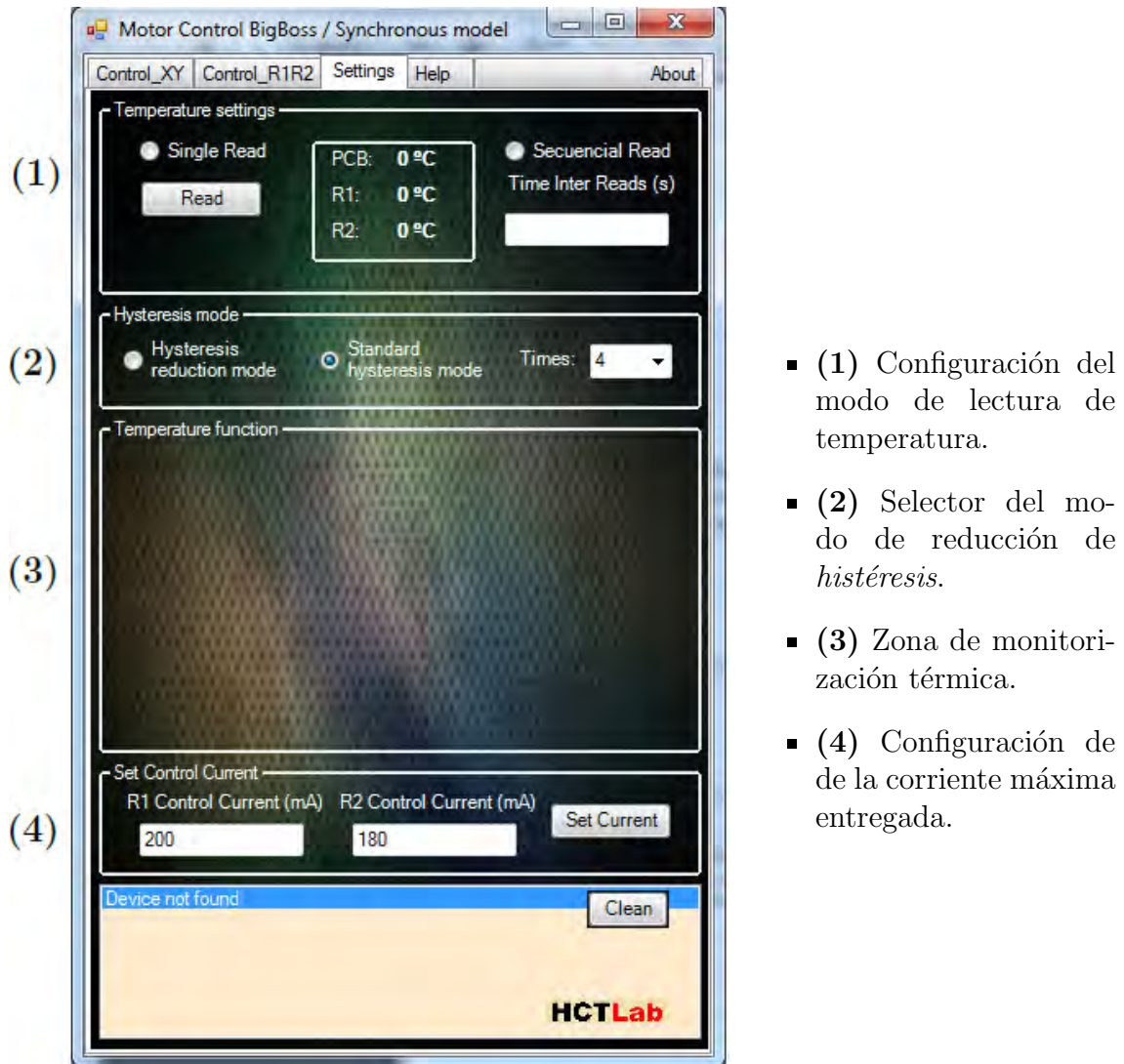


Figura 5.17: Pestaña para el control de parámetros generales.

Al igual que en el resto de pestañas, en ésta puede destacarse una ventana de monitorización en la que se presentan todos los envíos realizados, así como las alertas y errores acontecidos durante la ejecución. Esta ventana está enlazada con las dos expuestas anteriormente, por lo que todas las acciones llevadas a cabo en cualquiera de las pestañas, se verán reflejadas en las tres.

Como se ha expuesto con anterioridad, la lectura de la temperatura y la monitorización térmica no se han configurado para este proyecto, dado que el esfuerzo en esta fase del proyecto ha sido focalizado en la optimización de tiempos, precisión y consumo. No obstante, esta aplicación está preparada para enviar y recibir este tipo de comandos, procesarlos y presentar los resultados obtenidos por pantalla.

### 5.3.4.1. Control de temperatura



Figura 5.18: Detalle de la sub-pestaña de monitorización térmica.

Como se expone en la sección 4.2.2.1, en este proyecto tiene vital importancia la monitorización de la temperatura en tres puntos característicos: el motor 1, el motor 2 y el circuito impreso. En esta aplicación, puede realizarse la configuración del tipo de monitorización que el usuario desee llevar a cabo, pudiéndose seleccionar entre una lectura puntual o la realización de una lectura periódica de la temperatura.

En esta sub-pestaña (figura 5.18) pueden destacarse cinco elementos característicos: dos botones radiales con los que seleccionar el modo de lectura, un cuadro de texto en el que especificar el periodo de monitorización (segundos) con el que quiere realizarse la lectura continua, un botón de comienzo y una zona de texto en la que se exponen los resultados de cada lectura en grados centígrados.

Esta aplicación también incluye una funcionalidad en proceso de desarrollo (figura 5.17), en la que se presentará una gráfica que mostrará todos los valores de las muestras de temperatura recogidas, indistintamente al modo de monitorización seleccionado.

### 5.3.4.2. Algoritmo de reducción de histéresis

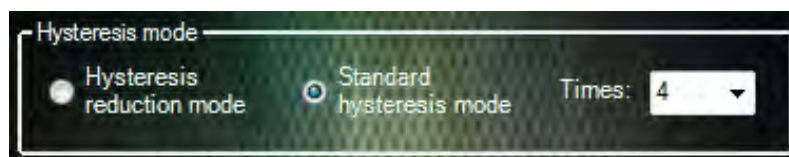


Figura 5.19: Detalle de la sub-pestaña de selección del algoritmo de *histéresis*.

La histéresis (sección 6.1) es un proceso a tener en cuenta dentro de la mecánica del robot. Este proceso consiste en la aparición de un determinado error en el posicionado del actuador a un punto concreto. Este error es diferente en función del sentido de giro de los motores, a la hora de alcanzar la posición deseada, debido probablemente a la mecánica interna del posicionador.

Para minimizar el error introducido por la histéresis e igualar la precisión del actuador, independientemente del sentido de giro de los motores, se ha desarrollado un algoritmo en colaboración con Justo Sánchez (IAA/CSIC, Granada). Este algoritmo realiza tres posicionados predefinidos, antes de alcanzar el objetivo marcado, siguiendo la siguiente secuencia:

1. Objetivo R1 + 27 pasos, Objetivo R2 - 17
2. Objetivo R1 - 27 pasos, Objetivo R2 - 37
3. Objetivo R1 - 27 pasos, Objetivo R2 + 17
4. Objetivo R1, Objetivo R2

Cabe destacar que este algoritmo sólo es ejecutable para posiciones de  $R1 \geq 27$  y de  $R2 \geq 54$ , dado que para objetivos inferiores a estos valores, se enviaría al posicionador a pasos negativos (hecho físicamente imposible).

En esta sub-pestaña pueden destacarse cuatro elementos: dos botones radiales con los que seleccionar el algoritmo o no, un selector con el que elegir el número de veces que el algoritmo ha de ejecutarse y un botón de confirmación.

#### 5.3.4.3. Limitación de corriente entregada

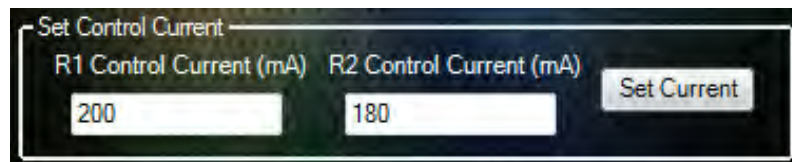


Figura 5.20: Detalle de la sub-pestaña de selección de la corriente entregada.

Como se expone en la sección 4.2.2.2, la limitación de la corriente entregada se realiza por medio de la configuración de los potenciómetros digitales instalados en la electrónica. Estos configuran su resistencia interna en función del código que el microcontrolador les manda a través del bus serie SPI. En esta sub-pestaña, es posible indicar el código deseado para cada uno de los potenciómetros. Siendo 255, la resistencia máxima (100 k $\Omega$ ) y 0 la mínima (60 $\Omega$ ).

En esta sub-pestaña pueden destacarse tres elementos característicos: dos cuadros de texto en los que indicar el código a enviar y un botón de envío.

La transmisión de información de este tipo de configuración se realiza mediante el envío de los comandos «F1» (para el potenciómetro 1) y «F2» (para el potenciómetro 2). En la figura 5.20 puede observarse el envío secuencial de estos comandos, con el código 200 para el potenciómetro 1 y 180 para el potenciómetro 2. Estos valores se han definido por defecto, tras un proceso de prueba, en el que se constató que esta limitación de corriente era suficiente para mover el robot, sin tener pérdida de pasos, optimizando el consumo medio.

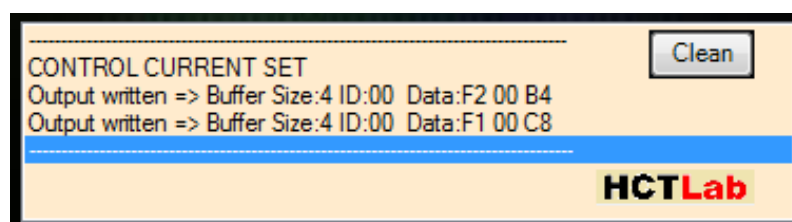
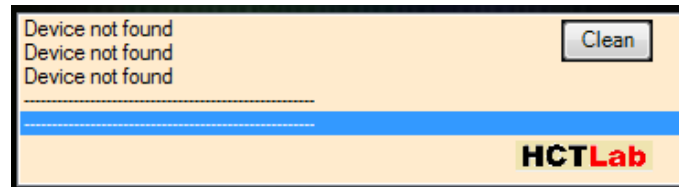


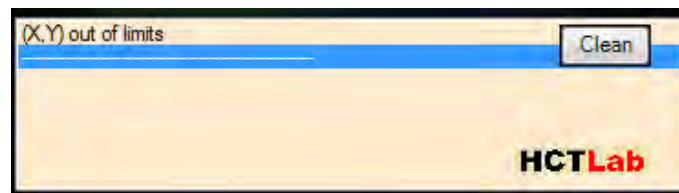
Figura 5.21: Ventana de monitorización en la configuración de corriente.

### 5.3.5. Errores y alertas de ejecución

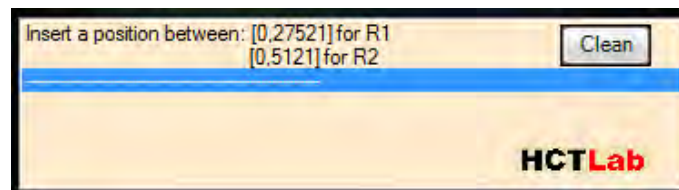
Como se ha explicado en esta sección, el software de control imprime en la ventana de monitorización todos los errores acontecidos durante la ejecución del programa. Siendo los más característicos los cinco siguientes:



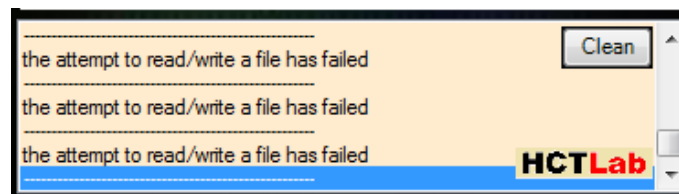
(a) Error 1: dispositivo no conectado.



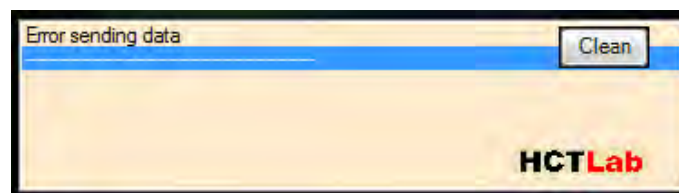
(b) Error 2: posición xy no permitida.



(c) Error 3: posición R1R2 fuera de límites.



(d) Error 4: error en el formato de los ficheros.



(e) Error 5: caracteres prohibidos en el cuadros de texto.

Figura 5.22: Errores más comunes en la ejecución del software de control

- **Error 1:** este error es mostrado cuando el usuario intenta enviar información de cualquier tipo al dispositivo, estando éste desconectado.
- **Error 2:** este error se muestra cuando el sistema intenta acceder a posiciones prohibidas de xy. El área máxima en el plano cartesiano que puede abarcar el posicionador es una circunferencia de  $L1+L2$  de radio, estando todos los puntos exteriores, fuera del alcance del robot.

- **Error 3:** este error es mostrado cuando el usuario intenta mover el robot, a posiciones fuera de los límites marcados en el modo R1R2.
- **Error 4:** este error se muestra cuando se ha detectado un fallo de formato en cualquiera de los ficheros implicados en la ejecución (ficheros de comunicación del modo automático y fichero de posición actual).
- **Error 5:** este error es mostrado cuando el usuario introduce caracteres prohibidos (cualquier carácter no numérico distinto del punto o la coma) en cualquiera de los cuadros de texto de la aplicación.

## 5.4. Software esclavo, código embarcado

El software esclavo es el código con el que el microcontrolador está programado. Éste es el encargado de recibir los comandos enviados por el software de control, procesarlos y actuar en consecuencia. Este código es muy similar al definitivo que irá embarcado en la electrónica final, estando preparado para interactuar con cualquier software de control que utilice el set de comandos especificado en la sección 5.3.1.1.

Para la realización de este software se ha utilizado el famoso entorno de programación Eclipse, adaptado por Code Red para la interoperatividad con todos los dispositivos de NXP. Éste es un entorno orientado a objetos, en el que pueden desarrollarse diferentes aplicaciones escritas en distintos lenguajes (C, C++, Android, VHDL, etc.). Optándose, para la programación del software esclavo, la utilización del lenguaje básico «C». Este entorno está únicamente desarrollado para Windows, por lo que, una vez más, la realización de este proyecto se ve dedicada a este sistema operativo.

Durante este proceso, han sido utilizadas multitud de librerías puestas a la disposición del público general por los desarrolladores de aplicaciones de NXP. Las más utilizadas han sido, entre otras, las librerías genéricas para la comunicación SPI, las librerías genéricas para la utilización de los puertos de propósito general (GPIO) y la librería específica, en su última versión de desarrollo, para la comunicación USB: «npxUSBlib».

A la hora de programar la placa de desarrollo LPCXpresso, o el microcontrolador (a través de la sub-tarjeta *LPC-Link*), no es preciso la instalación de drivers adicionales, dado que los necesarios para llevar a cabo este proceso son encontrados automáticamente por el sistema operativo. No obstante, con la instalación del entorno de programación se descargan todos los drivers necesarios para llevar a cabo la programación y comunicación con componentes de NXP.

A continuación se expone el funcionamiento general del código embarcado en el microcontrolador. La descripción en detalle de las funcionalidades y uso de cada uno de los comandos implicados en este código, se exponen en las secciones 4.2.2.1, 5.3.1.1 y 4.2.2, dado que estas funciones implican tanto al software de control como a componentes hardware.

Este firmware posee la siguiente estructura de funcionamiento:

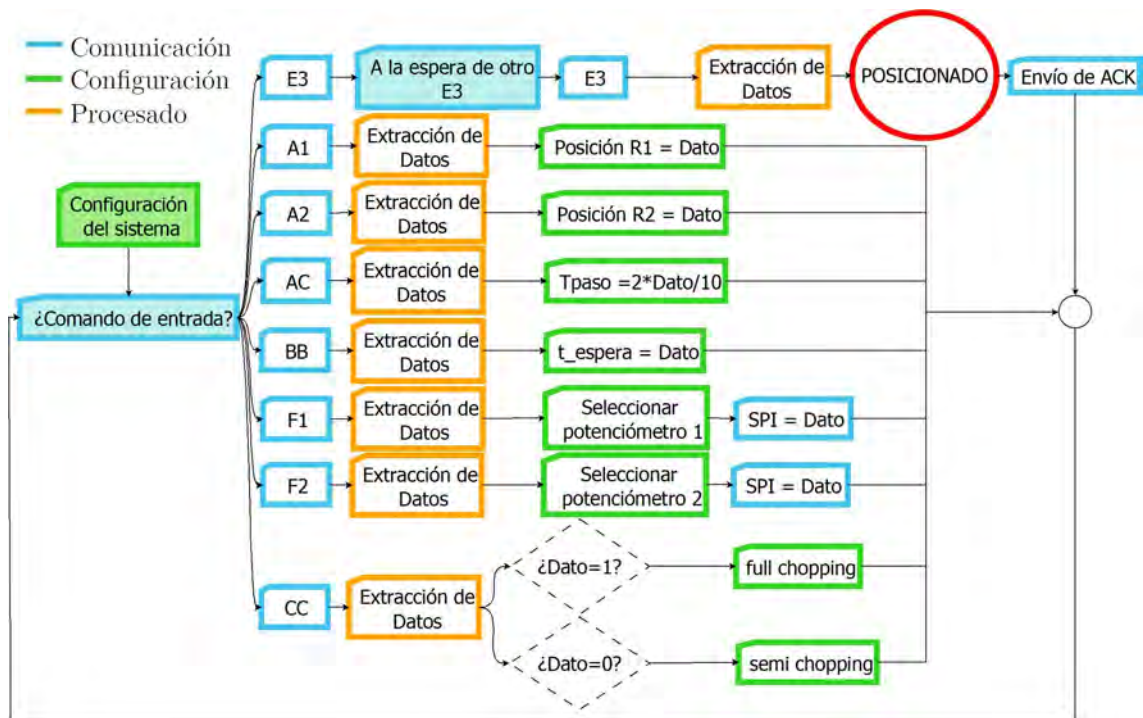


Figura 5.23: Diagrama de flujo del software esclavo.

En este diagrama pueden distinguirse cuatro grandes etapas: la configuración del sistema, el intercambio de datos, la extracción y procesado de datos y la configuración interna. Siendo esta última la más relevante.

1. **Configuración del sistema:** tras el encendido del sistema, se configuran todos los parámetros implicados en el funcionamiento de éste. En primer lugar se configuran todos los puertos de entrada y salida utilizados para la comunicación entre el microcontrolador y todos los dispositivos implicados en el diseño. En este proceso, se especifica el tipo de uso que se le va a dar a cada puerto (dado que existe un gran número de ellos que poseen varias funcionalidades), se activa su *pull-up* o *pull-down* interno y se especifica si éste va a ser de entrada o de salida. En segundo lugar, se configura la velocidad del reloj (dependiendo de la velocidad de procesado que interese) y se configuran los diferentes periféricos a utilizar (USB, SPI e  $I^2C$ ), indicando la velocidad, el tamaño de palabra y especificando cual va a ser utilizado en caso de existir más de uno (SPI). Finalmente se inicializan las diferentes variables globales implicadas en el sistema.
2. **Espera de comandos:** tras la configuración general del sistema, se entra en el bucle de funcionamiento del microcontrolador, en el que se permanecerá continuamente escuchando el puerto USB o  $I^2C$  (en función de la versión del hardware) y se actuará en consecuencia tras la recepción de un comando correcto.
3. **Extracción de datos:** tras la recepción correcta de un comando, se decodifica y se extrae la cabecera, conociendo de que comando se trata. Una vez especificado el comando, se extraen los dos bytes de datos que éste posee guardándolos en una variable específica de cada comando.

4. **Procesado de información:** una vez conocido el comando recibido y separados los datos que éste portaba, se realizan las funciones específicas de cada uno como se especifica en las secciones 4.2.2.1, 5.3.1.1 y 4.2.2.
5. **Posicionado:** si el comando recibido es un comando de posicionado «E3», se guarda el dato encapsulado en una variable reservada al posicionado del eje R1 y se espera la recepción de un nuevo comando de posicionado. Tras esta segunda recepción, se almacena el nuevo dato recibido en una variable reservada al posicionado del eje R2. Una vez recibidos y almacenados los dos datos, se ejecuta el algoritmo de posicionado que realiza el movimiento del robot. Este algoritmo se ha desarrollado junto a Justo Sánchez (IAA/CSIC, Granada) y conforma la parte más importantes del proyecto. Su exposición y desarrollo está sujeto a términos de confidencialidad, por lo que no puede mostrarse en detalle en este documento.

## 5.5. Conclusiones

El software de control realizado para testar los diferentes modelos de la electrónica y los distintos prototipos de robot fabricados, está sujeto a continuos cambios y modificaciones. Ésta es una herramienta útil para el ingeniero, a la hora de probar el funcionamiento tanto de la electrónica diseñada, como del robot. No obstante, está programado para actuar específicamente con un único dispositivo. En un periodo corto de tiempo, se realizarán mejoras sobre esta aplicación para permitir la interoperatividad entre varios dispositivos funcionando simultáneamente.

Por otro lado, el firmware diseñado es capaz de operar con cualquier software de control que utilice el set de comandos mostrado, por esta razón éste podría ser el código que irá definitivamente embarcado en cada uno de los 5000 posicionadores.



# Capítulo 6

## Integración, pruebas y resultados

*Jamás se descubrirá nada si nos considerásemos satisfechos con las cosas descubiertas.*

Lucio Anneo Séneca.

**Resumen:** en este capítulo se presentan los diferentes test, realizados en distintos centros de investigación, mediante el software de control y el modelo v.1 de la electrónica con la última versión del posicionador.

### 6.1. Laboratorios y centros de prueba

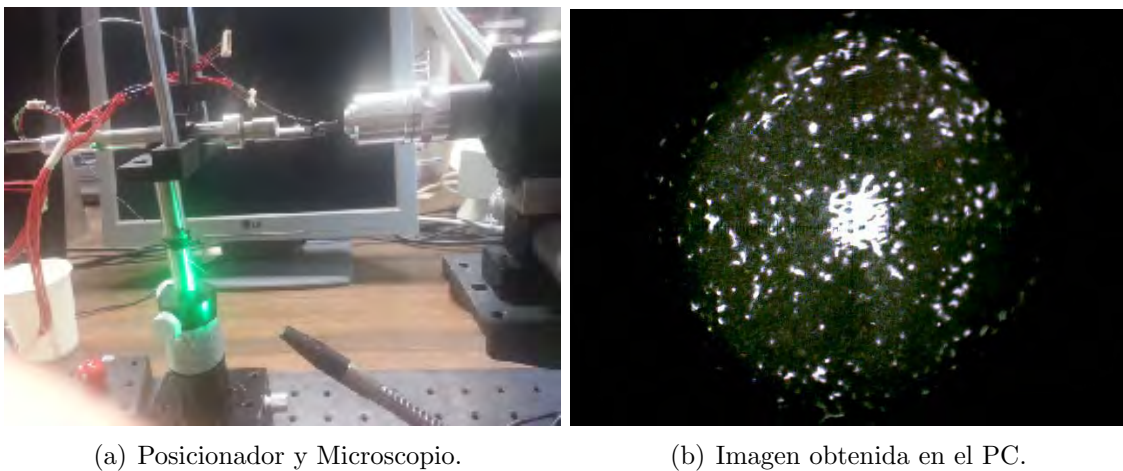
Tras el desarrollo completo del software de control y la programación del firmware embarcado en el microcontrolador, se han realizado un conjunto de test de funcionamiento para estudiar todo tipo de comportamientos del robot diseñado. Estos han sido llevado a cabo en diferentes centros de investigación. Por un lado, la mayor parte de los test de funcionamiento que han ayudado en el desarrollo de la electrónica, descubriendo errores de diseño, *bugs* en el código y sirviendo de soporte fundamental en la aplicación de mejoras y adaptaciones en los prototipos, han sido llevadas a cabo en las instalaciones granadinas del *Instituto Astrofísico de Andalucía* (IAA/CSIC), bajo el soporte y dirección de Justo Sanchez. Por otro lado, los test que implicaban una mayor inversión tecnológica y un proceso de prueba más prolongado, fueron realizados en las instalaciones del *Lawrence Berkeley National Laboratory* (LBNL) perteneciente a la Universidad de Berkeley (California).

### 6.2. Pruebas realizadas en el IAA

Tras la realización del primer prototipo de la electrónica y la comprobación básica de la buena respuesta de éste al envío de los comandos fundamentales del set (sección 5.3.1.1), se envió el prototipo v.1 junto a la primera versión del software de control a las instalaciones del IAA en Granada. Allí se realizaron los primeros test de integración entre la electrónica diseñada y el robot. Estos test pusieron en evidencia la buena respuesta de la electrónica y algoritmos diseñados. No obstante,

fueron fundamentales para la realización de mejoras de vital importancia que hicieron del modelo v.1 un prototipo con resultados mucho mejores a los alcanzados por los robots diseñados en el proyecto LAMOST (sección 2.1.4).

Para la realización de estos test, se ha utilizado un fragmento de fibra óptica de unos 40 cm, conectada en uno de sus extremos al posicionador y en el otro a una fuente luminosa. En el frontal del robot se ha instalado un microscopio con una webcam, encargada de tomar una imagen tras cada posicionado. Para facilitar el análisis de las imágenes obtenidas se ha instalado un filtro plástico en el cabezal a posicionar de la fibra, que reduce la luminosidad de ésta y limita su superficie. A la hora de analizar las imágenes, se han utilizado ciertos puntos de referencia de la fibra (manchas o marcas de la misma), con el objetivo de estudiar su posición antes y después del posicionado.



(a) Posicionador y Microscopio.

(b) Imagen obtenida en el PC.

Figura 6.1: Sistema de prueba en el IAA.

El conjunto de pruebas realizadas en el IAA fueron llevadas a cabo en condiciones no controladas, por lo que este sistema de medición está sujeto a multitud de diferentes factores de ruido impredecibles. Por un lado, la deriva térmica que sufre tanto la fibra como diversos componentes internos del robot, hace que la variación de temperatura de la habitación en la que se realizan los diferentes experimentos, cause un incremento significativo del error. Por otro lado, las posibles vibraciones del entorno ruidoso invalidan multitud de resultados. Por lo que se hace necesaria la realización de un conjunto de test de mayor precisión, en un entorno controlado al detalle. Por esta razón, los test llevados a cabo en el IAA son pruebas de corta duración que intentan demostrar la buena respuesta del diseño al más bajo nivel, dando una idea de la capacidad de repetitividad y precisión del sistema formado por el posicionador, la electrónica y el software de control.

### 6.2.1. Pruebas de repetitividad

Este conjunto de test es de vital importancia, dado que con ellos se pretende estudiar los errores del sistema (posicionador, software y electrónica) en el posicionado de la fibra en un cierto punto. Para la realización de estos test, se manda mover el posicionador de un punto «A» (cuya posición es bien conocida) a un punto «B» (pasando por un conjunto de puntos intermedios) y de nuevo al punto de partida «A», es-

tudiando los errores de posición de este último punto respecto a la posición inicial conocida.

R1	R2	RR2
6880	2560,00	7680
6881	2560,26	7681
6882	2560,51	7682
6883	2560,77	7683
6884	2561,02	7684
6885	2561,28	7685
6886	2561,53	7686
6887	2561,79	7687
6888	2562,05	7688
6889	2562,30	7689
6888	2562,05	7688
6887	2561,79	7687
6886	2561,53	7686
...	...	...

(b) Pasos dados

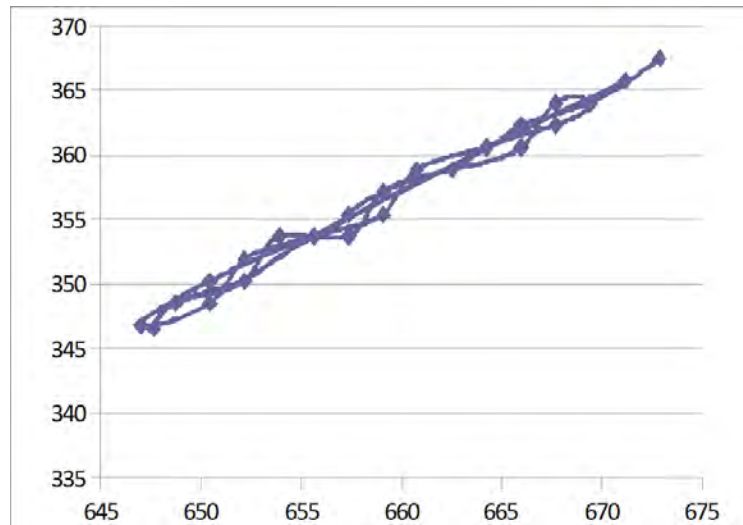
(b) Eje cartesiano en  $\mu\text{m}$ 

Figura 6.2: Extracto de un estudio sobre repetitividad (Justo Sánchez IAA/CSIC).

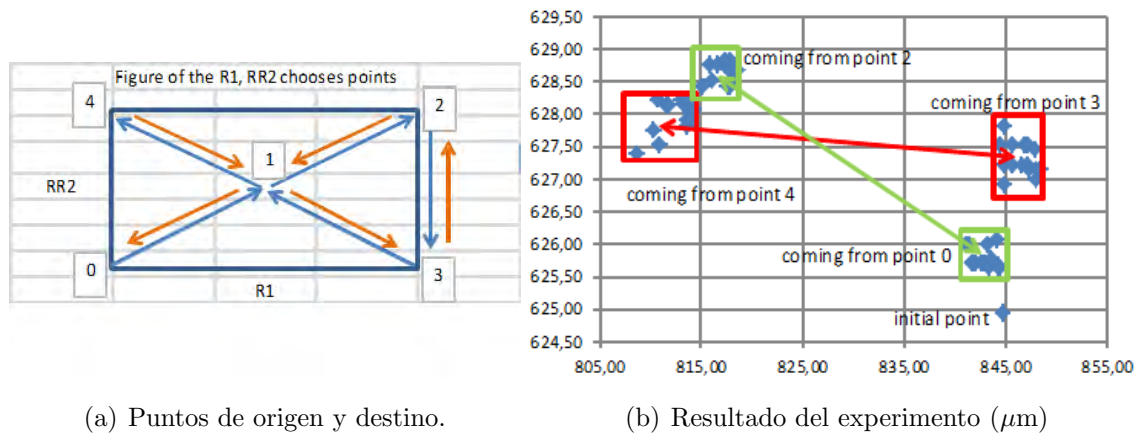
En el experimento realizado (figura 6.2) se ejecuta el posicionado secuencial de los ejes R1 y RR2 (pasos R2 más los inducidos por el movimiento de R1) desde el punto R1=6880 y RR2=7680 al punto R1=6889 y RR2=7689, realizando tres veces un movimiento de subida y bajada en escalera. Tras este experimento es posible destacar los buenos resultados de repetitividad del sistema, consiguiendo una caja de error total de  $7.38 \times 5.33 \mu\text{m}$ .

### 6.2.2. Pruebas de histéresis

Como se ha expuesto en la sección 5.3.4.2, debido al diseño del robot, éste presenta un efecto de histéresis en el posicionado. Este proceso consiste en la aparición de un determinado error en el movimiento del actuador. Este error es diferente en función del sentido de giro de los motores a la hora de alcanzar la posición deseada, debido probablemente a la mecánica interna del posicionador. Confirmada la buena repetitividad y precisión del sistema, éste es el error más destacable.

Para constatar este hecho, se han realizado diferentes test de posicionado cuyo objetivo es alcanzar un mismo punto desde posiciones diferentes. En la figura 6.3 pueden observarse los resultados del test de histéresis realizado en ausencia del algoritmo de reducción de error (sección 5.3.4.2). En este experimento se llega a un punto intermedio, partiendo de los cuatro extremos del área operativa del posicionador. En él, es posible observar como existe un claro error repetitivo en el movimiento hacia el punto central, procediendo de diferentes orígenes. No obstante, este experimento constata la buena repetitividad del sistema, dado que casi la totalidad de los puntos caen en una caja de error predecible y acotada de hasta  $4 \times 0.75 \mu\text{m}$ . Por otro lado, los resultados de histéresis varían entre los  $20 \mu\text{m}$  y los  $40 \mu\text{m}$ , siendo estos

valores comunes en todos los experimentos, hecho que constata el error de histéresis sistemático y repetitivo del mecanismo.



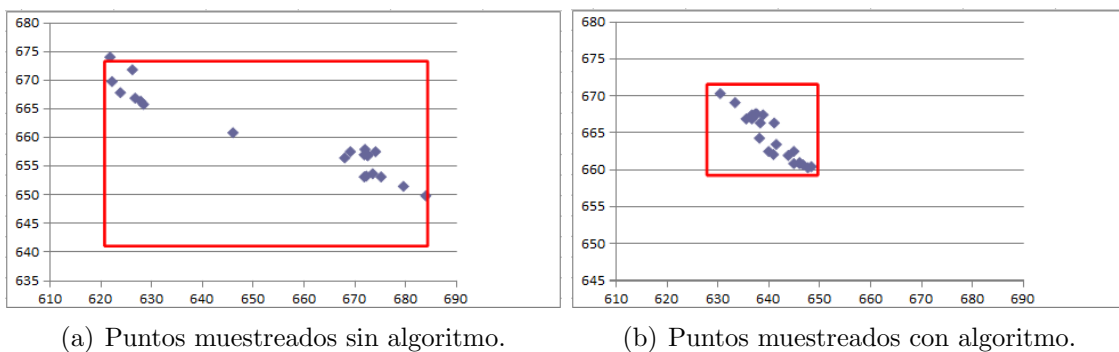
(a) Puntos de origen y destino.

(b) Resultado del experimento ( $\mu\text{m}$ )

Figura 6.3: Experimento de histéresis (Justo Sánchez IAA/CSIC).

Tras el desarrollo del algoritmo de reducción de histéresis, se han realizado varios experimentos que constatan la introducción de una clara mejoría respecto al modo estándar de funcionamiento. Este algoritmo puede ser ejecutado varias veces antes de alcanzar la posición objetivo, reduciendo aun más los errores introducidos por la mecánica.

En el experimento expuesto en la figura 6.4 se realiza el posicionado cartesiano desde diferentes puntos de un grid de  $2 \times 2$  mm centrado en el origen, hasta el punto  $x = -4$  mm e  $y = 4$  mm. En este test puede verse la clara reducción del error que aporta una única iteración del algoritmo.



(a) Puntos muestreados sin algoritmo.

(b) Puntos muestreados con algoritmo.

Figura 6.4: Mejora de los resultados tras una única iteración del algoritmo de reducción de histéresis (Justo Sánchez IAA/CSIC).

Tras la realización de este test, se recogen cajas de error sin algoritmo de  $60.1 \times 20.9 \mu\text{m}$ , siendo ésta después de la utilización del algoritmo de  $17.88 \times 8.8 \mu\text{m}$ , pudiéndose apreciar la clara mejoría que este método introduce en el sistema, respecto al modo estándar de funcionamiento.

Por otro lado se han realizado test de características similares (figura 6.5) en los que se mueve el robot desde diferentes posiciones enteras, hasta un cierto objetivo situado en los ejes cartesianos, comparándose los resultados obtenidos con y sin algoritmo.

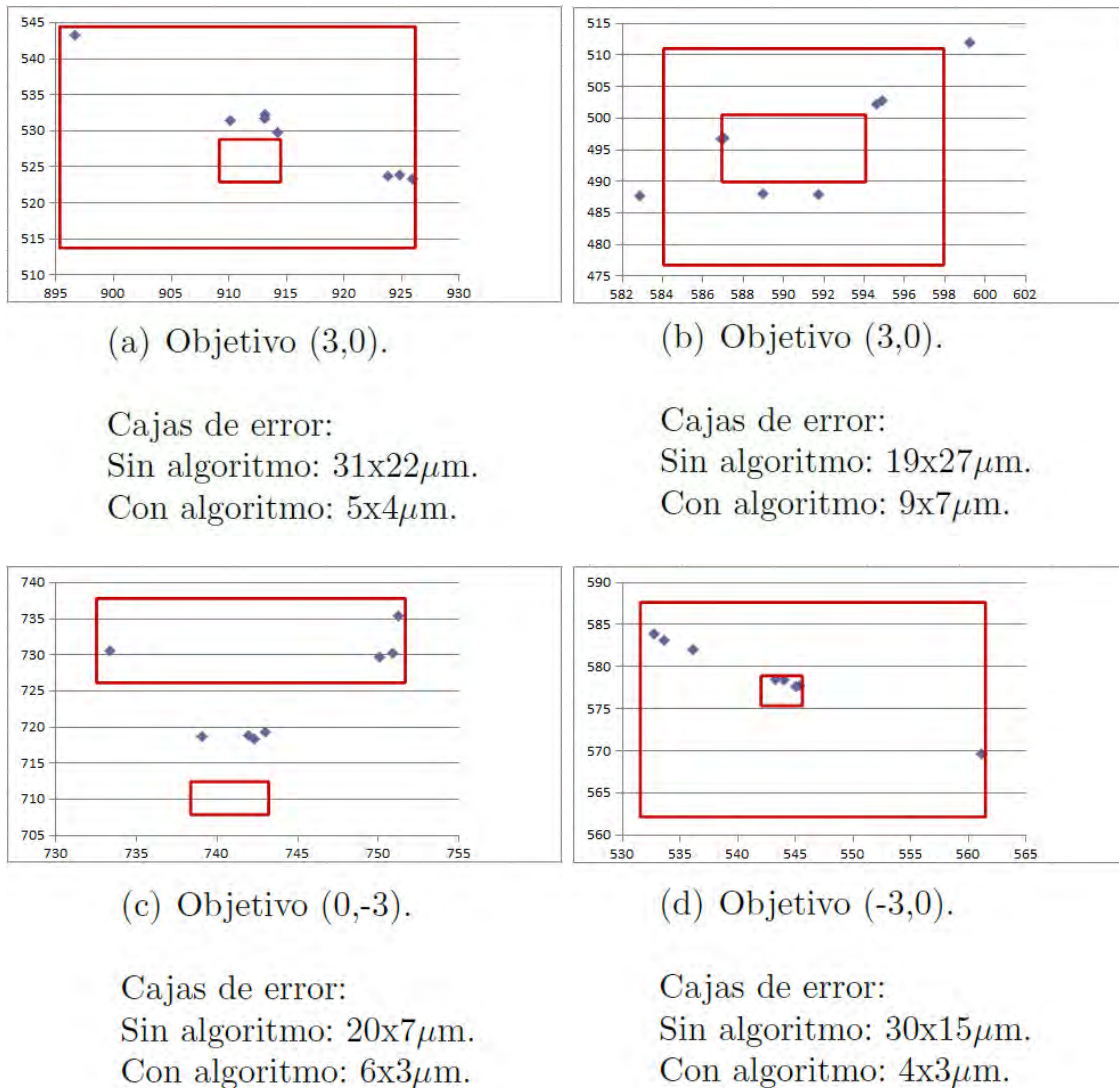


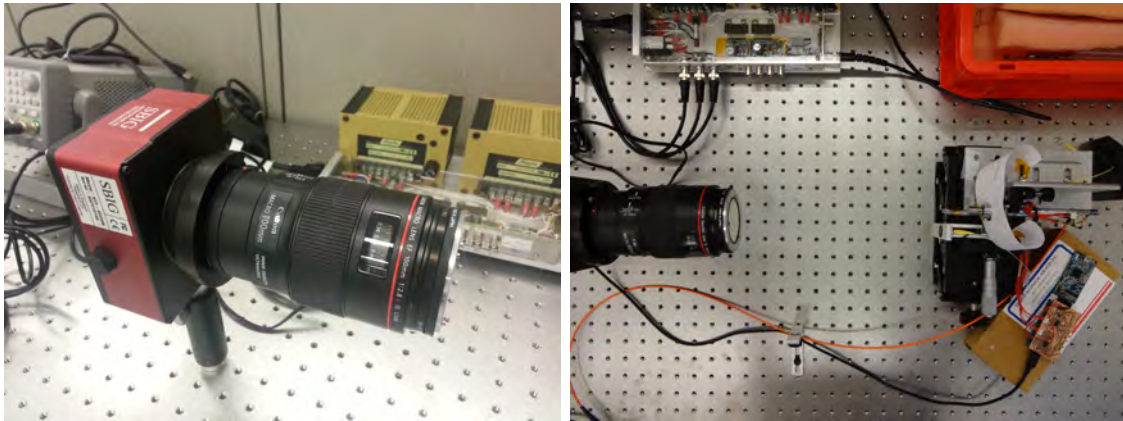
Figura 6.5: Otros test de histéresis(Justo Sánchez IAA/CSIC).

### 6.3. Pruebas realizadas en el LBNL

Con el objetivo de realizar pruebas de larga duración, en las que se sometiese el robot a largos periodos de posicionado, llevando éste a límites de estrés no alcanzados hasta el momento. Se trasladó el posicionador a las instalaciones californianas del LBNL de la Universidad de Berkeley. Allí se realizaron un conjunto de pruebas enmarcadas en entornos controlados al detalle, en los que la temperatura, vibración y otros factores de ruido estaban estrictamente acotados.

En estas instalaciones se realizaron cuatro conjuntos de pruebas: pruebas de calibración, pruebas de precisión (x,y), pruebas de variación en la componente z y pruebas de alineamiento con el eje óptico (*tip/tilt*). Estos test fueron llevados a cabo mediante el uso del algoritmo de automatización expuesto en la sección 5.3.2.2, por medio del estudio de las imágenes obtenidas a través de una cámara CCD (figura 6.6) o de la luz reflejada hacia un componente CMOS en el caso de las pruebas de alineamiento (figura 6.7).

### 6.3.1. Sistema de cámara CCD



(a) Cámara CCD.

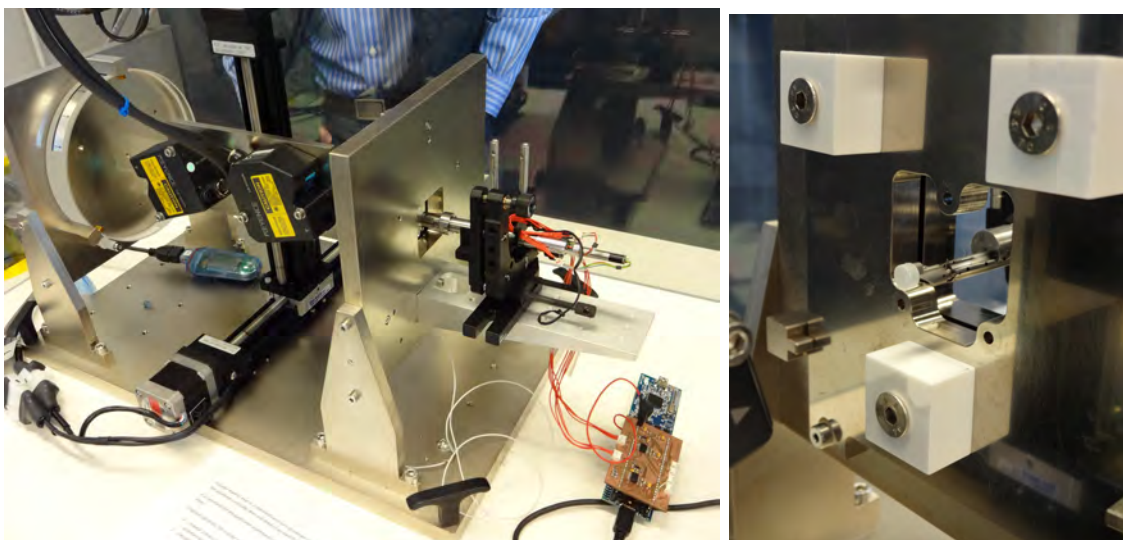
(b) Posicionador y electrónica.

Figura 6.6: Sistema CCD de calibración y prueba de precisión.

Este sistema es utilizado para llevar a cabo las pruebas de precisión y calibración del posicionador. Está compuesto por una cámara de alta definición CCD (con un objetivo macro de Canon), montada sobre un raíl móvil. Por otro lado, el posicionador va instalado en frente de la cámara, portando una fibra iluminada por una fuente LED de 632 nm y 78 mA. La cámara y la electrónica diseñada se conectan a un ordenador central en el que se ejecutan, tanto el software de control, como el software maestro de Berkeley.

Tras cada posicionado, la cámara toma una imagen y la guarda en una base de datos para su posterior análisis.

### 6.3.2. Sistema de reflexión CMOS



(a) Sistema laser.

(b) Detalle del posicionador.

Figura 6.7: Sistema CMOS para el estudio de la desviación vertical.

Este sistema es utilizado para medir la desviación respecto a la normal que sufre el robot en diferentes posicionados. El principal componente de este sistema es un emisor-receptor láser del fabricante Keyence (figura 6.8). Éste está compuesto por un emisor láser de intensidad ajustable, un sistema de lentes focalizadoras y un receptor RS-CMOS que analiza la señal reflejada en el objeto a estudiar. El posicionador es ajustado a una distancia predefinida del sistema láser. En el extremo del brazo L2 se ajusta un espejo en el que incide la señal láser emitida por el sistema Keyence, el cual analiza la recepción tras cada posicionado, con gran precisión.

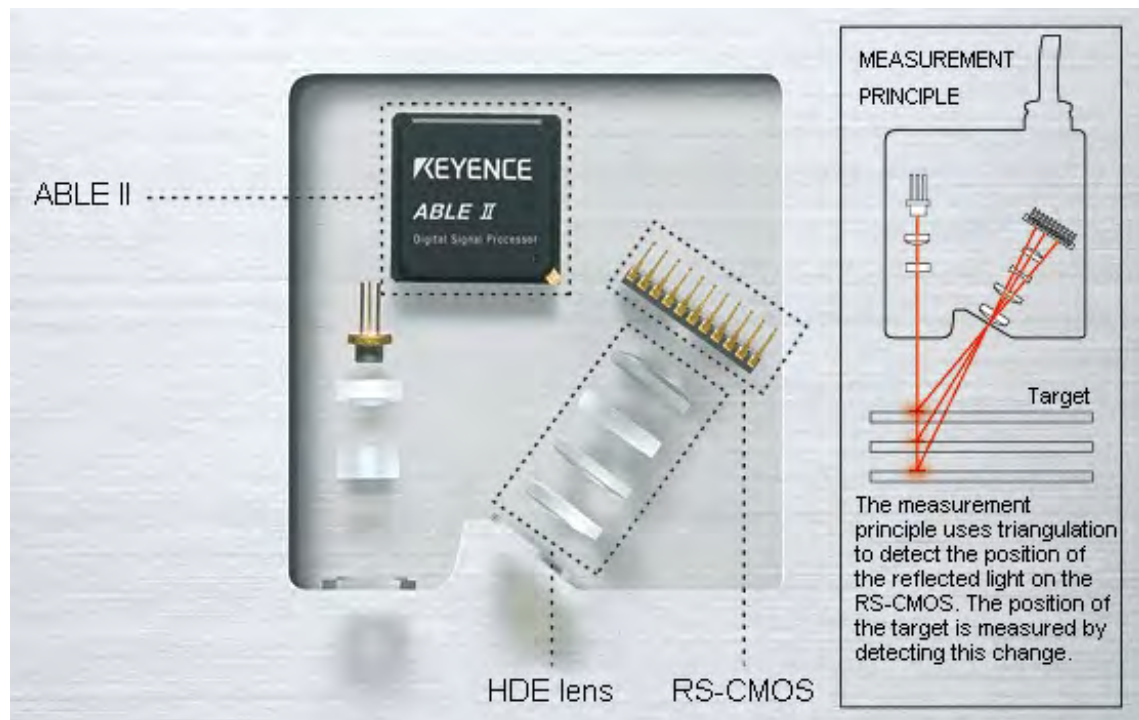
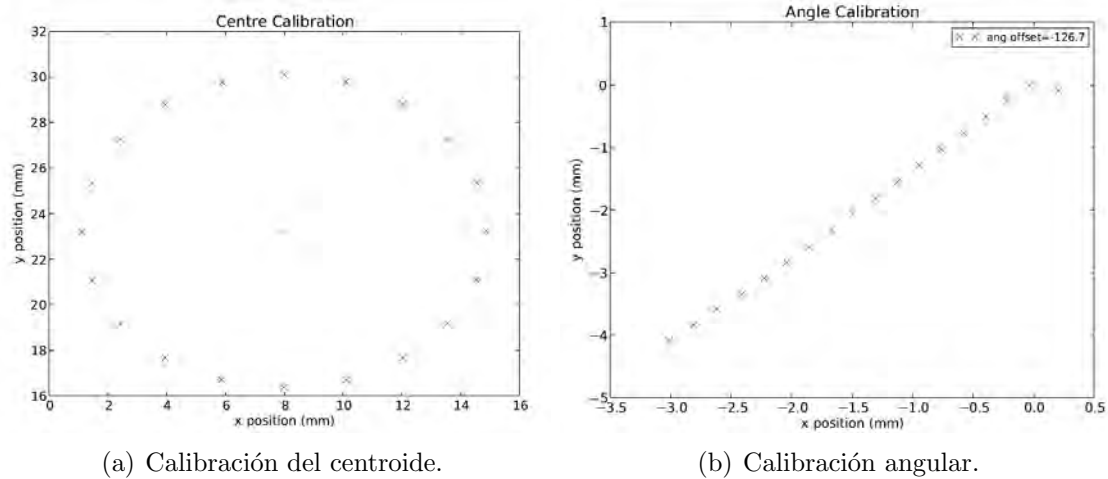


Figura 6.8: Sistema Keyence de emisión-recepción láser.

### 6.3.3. Pruebas de calibración

Estas pruebas se realizan de forma periódica con el sistema CCD tras conocer la precisión del robot. Dado que cada actuador es diferente, es preciso conocer determinadas variables que caracterizan a cada uno de ellos. En primer lugar, es necesario conocer el centroide real de éste (desviación respecto al centro teórico), con el objetivo de calcular el tamaño verdadero de cada uno de los brazos posicionadores y calibrar el robot. Para ello, se hace mover el robot a través de  $n$  posiciones a lo largo de los  $360^\circ$  de una circunferencia de radio  $\frac{3}{4}r_{max}$ . En segundo lugar, es necesario calcular la desviación angular (en  $\theta$ ) de éste. Para ello, se hace mover el posicionador, desde  $r_{min}$  a  $r_{max}$  para un  $\theta$  constante.

En la figura 6.9 puede observarse el resultado de un test de calibración con el posicionador, la electrónica diseñada, el software de control y la ejecución del modo automático de prueba.



(a) Calibración del centroide.

(b) Calibración angular.

Figura 6.9: Resultados del proceso de calibración.

### 6.3.4. Pruebas de precisión (x,y)

Estas pruebas se realizan con el objetivo de estudiar la precisión del posicionador (característica fundamental en aplicaciones astronómicas). Para ello se hace mover el robot a determinados puntos y se estudian los diferentes errores del posicionado real respecto de los puntos teóricos.

A la hora de realizar este tipo de test, se efectúan cierto número de correcciones, éstas se realizarán también una vez embarcados los robots en el plano focal. El número de correcciones consecutivas, efectuadas antes de alcanzar la posición definitiva, depende en gran medida de la limitación en tiempo de posicionado. Para realizar estas correcciones se hace mover el posicionador a un cierto punto y se estudian los diferentes tipos de errores respecto el punto deseado (figura 6.10), recalibrando el robot y aumentando su precisión.

$$\text{rms}_x = \sqrt{\frac{1}{N} \sum \text{error}_x^2} \quad \text{rms}_y = \sqrt{\frac{1}{N} \sum \text{error}_y^2} \quad \text{rms}_{xy} = \sqrt{\text{rms}_x^2 + \text{rms}_y^2}.$$

Figura 6.10: Errores calculados, donde N es el número de posicionados.

En la figura 6.11 puede observarse el resultado de la ejecución de un test en el que se realizan 53 posicionados del robot. En este experimento se efectúan nueve correcciones consecutivas, representándose el error de posicionado respecto a los objetivos teóricos. Este test muestra los errores máximos y *rms* en cada una de las correcciones. En él puede observarse un error acumulativo en las primeras iteraciones del experimento, este error es del orden de mm (inaceptable en esta aplicación), causado por un problema de pérdida de pasos existente en el momento del posicionado<sup>1</sup>.

En la representación del posicionado, pueden verse en verde los puntos teóricos y en azul las posiciones realmente alcanzadas. Por otro lado, en la gráfica (d) se representa en rojo el error xy, en azul el error en la dimensión x y en verde el error en la y.

<sup>1</sup>Este error apareció con el uso del modo *semi-chopping*, en el que se deshabilitaban los drivers de motor tras cada movimiento, perdiendo el paso inestable en el que se encontraba el motor



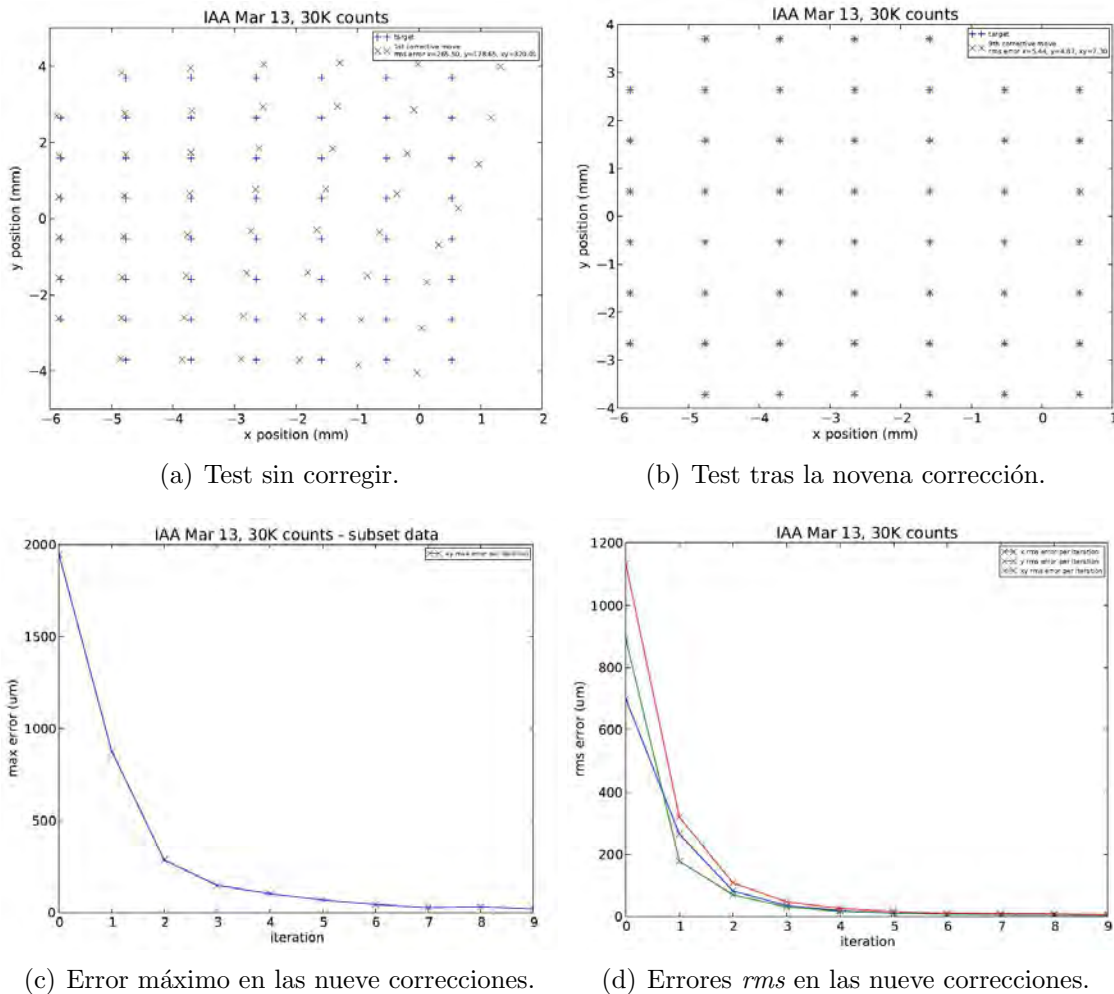


Figura 6.11: Test realizado con nueve correcciones sucesivas

Tras la realización de nueve correcciones consecutivas, se consiguen errores medios de hasta  $1.6 \mu\text{m}$  (*rms* de  $7.3 \mu\text{m}$ ), resultados mucho mejores a los alcanzados por el proyecto chino LAMOST ( $40 \mu\text{m}$ ) de características similares.

### 6.3.5. Pruebas de variación en $z$

A la hora de estudiar el error de posicionado en la dimensión  $z$ , se centran los esfuerzos en el análisis de la variación de esta componente en función de la temperatura. El robot está fabricado en materiales metálicos, muy susceptibles a dilataciones en presencia de cambios bruscos de temperatura. Debido a esto se han diseñado los mecanismos descritos en la sección 4.2.2.2 con el objetivo de estabilizar la temperatura del posicionador reduciendo al mínimo los cambios de temperatura introducidos por diferentes factores (condiciones atmosféricas, temperatura ambiente, temperatura de los motores, etc.).

Para analizar este fenómeno, se ha realizado un posicionamiento continuo durante un periodo de cuatro horas, estudiando la variación de temperatura en el mismo y el error en la dimensión  $z$  del posicionador en función de esta temperatura. En la figura 6.12 puede observarse como durante este periodo la temperatura ha variado

cuasi-linealmente, desde los 23.2°C hasta los 24.3°C, produciendo la variación en  $z$  expuesta en la gráfica. Tras la realización del experimento puede concluirse que en ausencia de la variación de temperatura, el error máximo en la componente  $z$ , no superaría los 6.4  $\mu\text{m}$ .

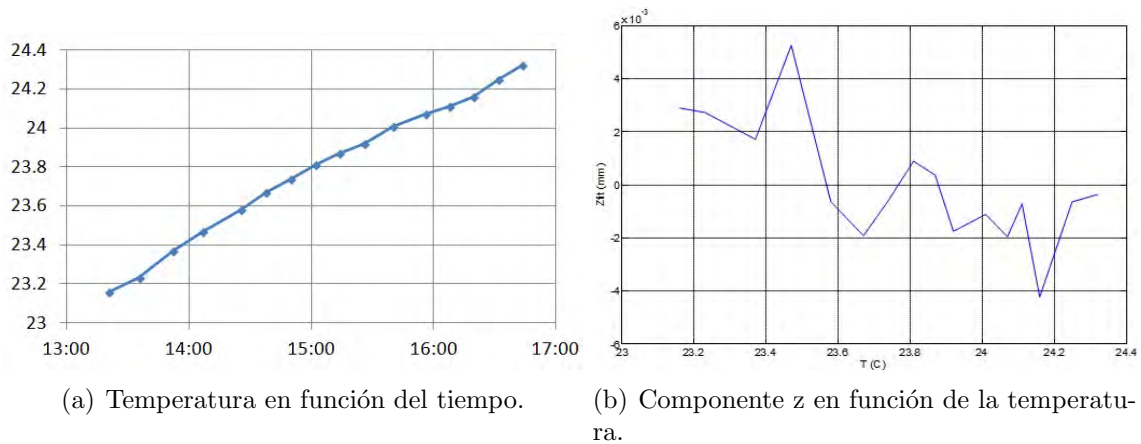


Figura 6.12: Variación de la componente  $z$  en función de la temperatura.

### 6.3.6. Pruebas de alineamiento con el eje óptico (*tip/tilt*)

Este tipo de pruebas forman, junto con los experimentos de precisión y corrección, el conjunto más importante. En esta prueba se estudia la desviación de la normal del posicionador frente a la dirección normal teórica de observación, tras la realización de un número significativo de posicionados.

Como se ha expuesto en la sección 6.3.2, este experimento se realiza por medio de la reflexión láser por parte de un espejo instalado en el frontal del posicionador. Este espejo refleja el láser proyectado por el analizador, hacia un sensor RS-CMOS que estudia la desviación del láser recibido, respecto al emitido.

En la figura 6.13 pueden observarse los resultados de uno de los experimentos realizados siguiendo este método. En él se recogen errores medios de desviación de hasta 0.322° antes de la corrección. Tras la realización de una corrección del sistema, gracias al estudio de los errores realizado, se consiguen resultados medios de hasta 0.035° de desviación respecto de la normal.

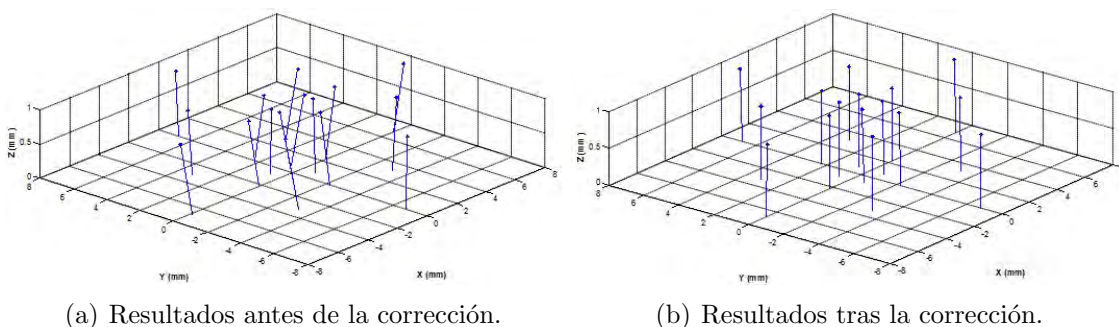
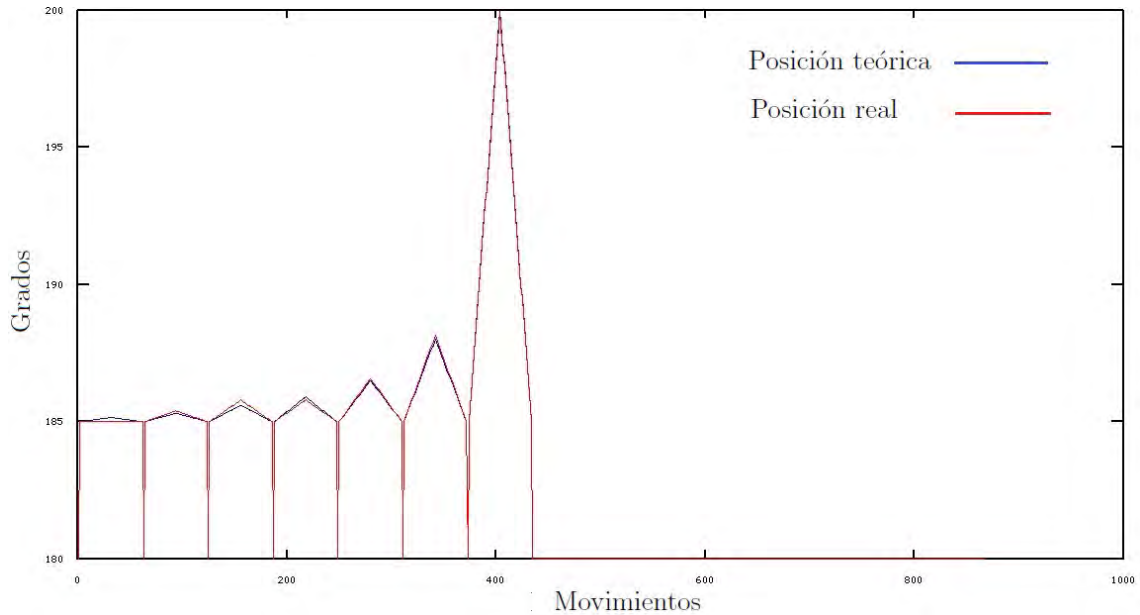


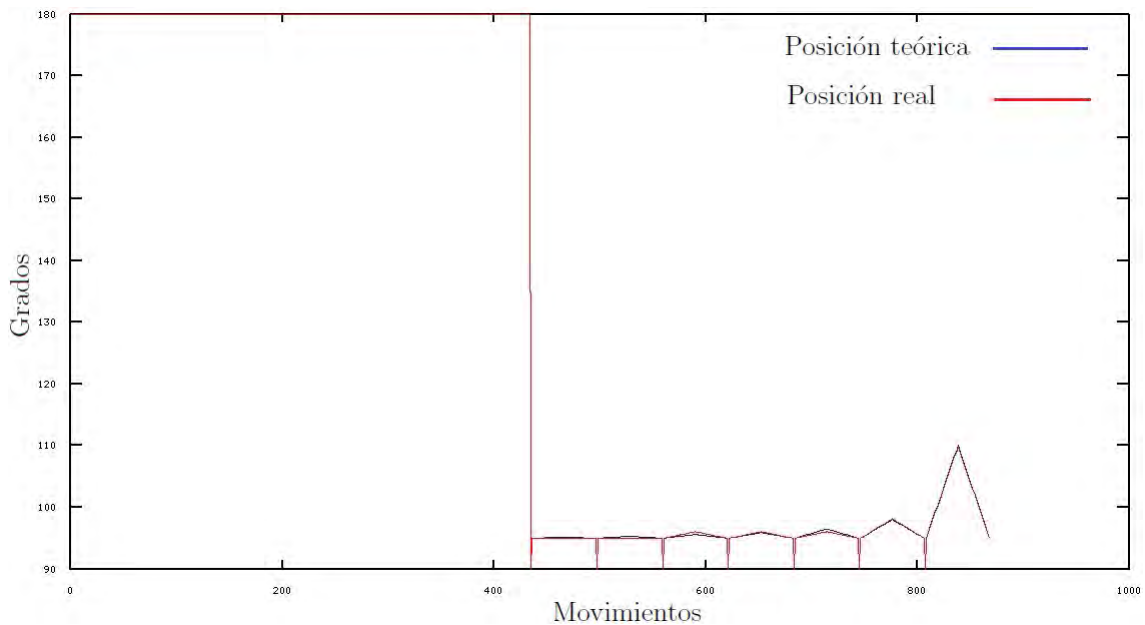
Figura 6.13: Estudio de la desviación respecto de la normal teórica.

## 6.4. Pruebas independientes

Analizando los datos de envíos en determinados test llevados a cabo en el LBNL, se ha realizado una comparación entre las posiciones teóricas requeridas por los tests y el dato que el software de control manda al actuador, recogiendo los resultados que se muestran en la figura 6.14 con el objetivo de analizar los errores de redondeo introducidos por el software de control.



(a) Datos para R1.



(b) Datos para R2.

Figura 6.14: Gráfica comparativa entre el movimiento teórico y los datos reales enviados al actuador desde el software de control.

En estas figuras puede apreciarse claramente como para ciertas etapas del movimiento, el dato enviado difiere ligeramente del dato teórico esperado. Esta situación es ocasionada por la acumulación de errores en el redondeo del dato a enviar.

El dato de posición generado por el software maestro en el modo automático, es presentado en grados con parte decimal. Por otro lado, el actuador trabaja en pasos enteros. Debido a esto, a la hora de realizar la transformación de grados a pasos, se ha de redondear a la unidad, por medio de la siguiente expresión:

$$Pasos_{R1} = Round(Grados_{R1} \cdot 27520/360)$$

$$Pasos_{R2} = Round(Grados_{R2} \cdot 5120/180)$$

Este hecho hace que como máximo pueda cometerse un error de 0.5 pasos. Por lo que el umbral de error en este caso será de  $0.0065^\circ$  para R1 y  $0.0176^\circ$  para R2.

Dividiendo la figura 6.14 en los siete tramos reconocibles (para el movimiento en R1), puede concluirse el siguiente resultado, siendo análogo para el movimiento en R2:

- **Primer tramo (ausencia de movimiento):** en este primer tramo puede apreciarse lo anteriormente expuesto. Los movimientos relativos ordenados por el software maestro son de  $0.005^\circ$ , siendo éstos inferiores al umbral de redondeo, por lo que el software de control no manda nada al actuador.
- **Tercer tramo (movimientos superiores):** en este tercer tramo se mandan movimientos relativos de  $0.02^\circ$ , correspondientes a 1.529 pasos. Con este dato el software redondea los pasos a enviar a «2», sumando 0.471 pasos por envío, lo que generará un movimiento por encima de lo esperado.
- **Cuarto tramo (movimientos inferiores):** en este cuarto tramo, el software maestro envía pasos relativos de  $0.03^\circ$ , correspondientes a 2.29 pasos. Con este dato, el software de control redondea los pasos a enviar a «2», 0.29 pasos por debajo de lo esperado, lo que generará un envío inferior al teórico.

Éste es un fenómeno inevitable causado por la adaptación del sistema cartesiano al sistema angular R1R2. No obstante, el error introducido no es muy elevado, por lo que puede ser asumido.

## 6.5. Conclusiones

Tras la realización de todo este conjunto de experimentos, y el estudio de los resultados obtenidos, puede concluirse que el sistema formado por el posicionador, la electrónica diseñada y el software de control desarrollado, aporta resultados muy satisfactorios en cuanto a precisión, tiempo de posicionado y funcionalidad de los algoritmos se refiere. Consiguiendo resultados mucho mejores a los aportados por proyectos similares, que implican posicionadores de fibra independientes, para aplicaciones astronómicas.

# Capítulo 7

## Conclusiones y trabajo futuro

*Conclusión es el lugar donde llegaste cansado de pensar.*

Anónimo.

**Resumen:** este capítulo concluye el proyecto exponiendo las conclusiones obtenidas tras la realización del mismo y citando el trabajo futuro que puede realizarse para mejorar el sistema propuesto.

### 7.1. Conclusiones

Tras la realización de este proyecto, se han puesto de manifiesto los buenos resultados que aporta a un sistema de posicionado, formado por robots independientes, la dedicación de una electrónica de control para cada posicionador. Esto introduce innumerables ventajas en cuanto a tiempo de posicionado se refiere, permitiendo realizar un número significativo de observaciones en relativamente poco tiempo. Sin embargo, el uso de electrónicas independientes para cada posicionador introduce un aumento del consumo total del sistema. No obstante, la selección de componentes realizada y el diseño dedicado al *micro-consumo*, reducen esta variable a niveles de potencia despreciables, frente a los consumidos por los motores del posicionador. Por otro lado, la introducción de funcionalidades, como el control de temperatura y par de los motores, por medio de la modulación de corriente y la monitorización de las variables fundamentales (temperatura, posición y corriente) dedicadas a cada uno de los motores del robot, hacen de este diseño un sistema mucho más preciso y dedicado que los actualmente utilizados en aplicaciones similares.

Por otro lado, el desarrollo software realizado para la ejecución de pruebas y experimentos con un único posicionador, ha facilitado el estudio preciso del comportamiento del robot en innumerables situaciones. En primer lugar, este desarrollo ha permitido reducir el efecto de la histéresis mecánica, introducida por el diseño interno del posicionador. En segundo lugar, la parametrización de la corriente máxima entregada a cada motor, ha permitido estudiar los valores óptimos de esta variable para el movimiento de los motores y la optimización de la precisión. Por último, gracias a la programación del modo automático de funcionamiento, ha sido posible la realización de experimentos y pruebas de larga duración que han permitido estudiar

el diseño planteado, con mayor profundidad.

Finalmente, el conjunto de pruebas realizadas en las instalaciones del LBNL de la Universidad de Berkeley, han puesto de manifiesto las óptimas condiciones del sistema, presentando resultados (en cuanto a precisión y robustez del sistema se refiere) que superan otros diseños como el LAMOST, cuyo objetivo es el mismo.

Todo esto pone de manifiesto los buenos resultados de la solución presentada por el equipo español para el proyecto BigBOSS, haciendo de ésta una propuesta firme y seria que supera las metas planteadas por el proyecto internacional dirigido por la Universidad de Berkeley.

## 7.2. Trabajo futuro

Éste es un proyecto en continua evolución, que plantea metas y nuevos retos continuamente. Por ello, actualmente se sigue trabajando con el objetivo de optimizar el sistema. Por esta razón, en un periodo corto de tiempo se pretende abordar las siguientes líneas de trabajo:

- **Desarrollo, fabricación y prueba del diseño definitivo:** uno de los temas clave a abordar llegados a este punto es el diseño y la fabricación del prototipo final de la electrónica. Éste irá encapsulado en el chasis de un posicionador, por lo que sus dimensiones están estrictamente acotadas. La fabricación de este sistema se llevará a cabo con tecnologías muy superiores a las utilizadas para la realización de los prototipos v.1 y v.2 de la electrónica, por lo que éste ha de realizarse en un organismo exterior al HCTLab. Esto introduce nuevos retos, dado que este tipo de fabricación incrementa notablemente el coste del proyecto y éste es un dato a considerar.
- **Diseño definitivo de un sistema de codificación de posición:** como se ha expuesto en este proyecto, el sistema definitivo de codificación de la posición del robot, se encuentra actualmente en fase de diseño y prueba, dado que el principal objetivo, es reducir al mínimo la instalación de dispositivos entorno al posicionador. Esto introduce nuevos retos en el diseño, planteando la posibilidad de utilizar los propios devanados del motor como indicador de posición.
- **Diseño definitivo de un sistema de monitorización térmica:** el diseño del sistema de monitorización y regulación térmica, se encuentra sujeto a continuos cambios, dado que éste introduce necesariamente nuevos dispositivos entorno al posicionador, violando las especificaciones del proyecto. Este punto está siendo estudiado en detalle para reducir la superficie utilizada por los dispositivos que componen el sistema. Por otro lado, en este proyecto queda pendiente el diseño del sistema software de monitorización, sistema que podría ser incluido en cualquiera de las versiones a realizar del software de control.
- **Adaptación al sistema de comunicación  $I^2C$ :** en paralelo a la realización de este proyecto, otra línea de investigación del HCTLab, ha desarrollado el árbol  $I^2C$  con el que se comunicará la electrónica con el ordenador central.

Los prototipos de la electrónica diseñados, están adaptados a este protocolo de comunicación, no obstante, por el momento no se han diseñado rutinas internas de comunicación  $I^2C$ , por lo que este es un punto importante a abordar.

- **Estudio de nuevas posibles soluciones para la mejora de precisión:** en la actualidad, los algoritmos de movimiento diseñados, presentan buenos resultados en cuanto a precisión y velocidad de movimiento se refiere. No obstante, una nueva línea de trabajo a desarrollar, sería el diseño de nuevos algoritmos que optimicen el proceso de posicionamiento de la fibra, mejorando las marcas conseguidas por el momento.
- **Estudio de nuevos posibles mecanismos de reducción del consumo:** la electrónica ha sido diseñada en la línea del pico-consumo, por esta razón todos los componentes seleccionados presentan unas tasas de consumo de potencia muy reducidas. Por otro lado, la mayoría de estos componentes pueden ser manejados en varios modos de sueño. Esto posibilita el planteamiento de algoritmos de control que manejen los diferentes modos de sueño del sistema, en función del estado del mismo, optimizando las tasas de consumo y mejorando el rendimiento de la electrónica.

Finalmente, tras el proceso de prueba y testado del sistema diseñado para un único posicionador, se está trabajando en el desarrollo de un cluster de 19 robots para la simulación de los diferentes algoritmos de posicionado y prevención de colisiones, con el objetivo de simular el comportamiento final del sistema diseñado. Por esta razón una nueva línea de trabajo a abordar sería, el diseño e implementación de un software de control capaz de manejar este cluster simulando definitivamente, el software maestro que dirigirá el telescopio.





*La gran victoria que hoy parece fácil fue el resultado de pequeñas victorias que pasaron desapercibidas.*

Paulo Coelho



# Bibliografía

*En Egipto se llamaba a las bibliotecas el tesoro de los remedios del alma. En efecto, curábase en ellas de la ignorancia, la más peligrosa de las enfermedades y el origen de todas las demás.*

Jacques Benigne Bossuet

- [1] E. Neilsen *et al.*, “The sloan digital sky survey data archive server,” *Computing in Science Engineering*, vol. 10, no. 1, pp. 13–17, 2008.
- [2] R. E. Owen and others., “Fiber feed for the sdss spectrograph,” pp. 110–114, 1994.
- [3] C. M. Rockosi *et al.*, “The Sloan Extension for Galactic Understanding and Exploration,” in *American Astronomical Society Meeting Abstracts*, vol. 37 of *Bulletin of the American Astronomical Society*, p. 147.01, Dec. 2005.
- [4] J. C. Wilson *et al.*, “Performance of the apache point observatory galactic evolution experiment (apogee) high-resolution near-infrared multi-object fiber spectrograph,” pp. 84460H–84460H–15, 2012.
- [5] N. M. De Lee *et al.*, “The SDSS-III MARVELS Survey: The First Year,” in *American Astronomical Society Meeting Abstracts #215*, vol. 42 of *Bulletin of the American Astronomical Society*, p. 472.05, Jan. 2010.
- [6] D. J. Schlegel *et al.*, “SDSS-III: The Baryon Oscillation Spectroscopic Survey (BOSS),” in *American Astronomical Society Meeting Abstracts*, vol. 39 of *Bulletin of the American Astronomical Society*, p. 132.29, Dec. 2007.
- [7] D. Schlegel *et al.*, “The BigBOSS Experiment,” 2011.
- [8] D. Schlegel *et al.*, “The Baryon Oscillation Spectroscopic Survey: Precision measurements of the absolute cosmic distance scale,” 2009.
- [9] D. J. Schlegel *et al.*, “BigBOSS: The Ground-Based Stage IV Dark Energy Experiment,” 2009.
- [10] J. H. Silber *et al.* pp. 845038–845038–13, 2012.
- [11] S. C. Barden and others., “Kaos: kilo-aperture optical spectrograph,” pp. 364–372, 2004.
- [12] A. M. Moore *et al.*, “Extending the echidna concept: a proposal for a 2000+ multi-fiber positioner,” pp. 299–305, 2002.

- 
- [13] S. C. Barden *et al.*, “Instrumentation at the anglo-australian observatory,” pp. 75–81, 2004.
- [14] P. R. Gillingham *et al.*, “The fiber multi-object spectrograph (fmos) project: the anglo-australian observatory role,” pp. 985–996, 2003.
- [15] F. G. Watson *et al.*, “6df: a very efficient multiobject spectroscopy system for the uk schmidt telescope,” pp. 834–843, 1998.
- [16] S. Miziarski *et al.*, “Concepts for multi-ifu robotic positioning systems,” pp. 845018–845018–14, 2012.
- [17] G. Smith *et al.*, “A survey of fiber positioning technologies.”
- [18] Fabricant *et al.*, “Hectospec: a 300-optical-fiber spectrograph for the converted mmt,” pp. 251–263, 1994.
- [19] P. Felenbok *et al.*, “The performance of mefos, the eso multi-object fibre spectrograph,” *Experimental Astronomy*, vol. 7, no. 2, pp. 65–85, 1997.
- [20] Fisher *et al.*, “Cobra: A two-degree of freedom fiber optic positioning mechanism,” in *Aerospace conference, 2009 IEEE*, pp. 1–11, 2009.
- [21] H. Hu *et al.*, “New type optical fiber positioning unit device for lamost,” pp. 548–555, 2003.
- [22] B. Koc *et al.*, “A piezoelectric motor using two orthogonal bending modes of a hollow cylinder,” *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, vol. 49, no. 4, pp. 495–500, 2002.
- [23] S. Sherrit *et al.*, “Acoustic mechanical feedthroughs,” pp. 86920P–86920P–9, 2013.
- [24] B. He *et al.*, “A multi-modes cylindrical piezoelectric rotary motor with a single-vibrator,” in *Piezoelectricity, Acoustic Waves and Device Applications (SPAW-DA), 2012 Symposium on*, pp. 101–104, 2012.
- [25] B. Arango *et al.*, “Stepper motor drives for robotic applications,” in *Power Engineering and Optimization Conference (PEDCO) Melaka, Malaysia, 2012 Ieee International*, pp. 361–366, 2012.
- [26] G. Athalye *et al.*, “A stepper motor and serial communication interface daughter board for educational use,” in *SoutheastCon, 2007. Proceedings. IEEE*, pp. 328–332, 2007.
- [27] T. C. Chin *et al.*, “A stepper motor controller,” in *Control, 1988. CONTROL 88., International Conference on*, pp. 500–505, 1988.
- [28] The Next eXPerience semiconductors. url: <http://www.nxp.com>.
- [29] F. Zha *et al.*, “Design of schematic mapping system based on i2c and usb bus,” in *Instrumentation, Measurement, Computer, Communication and Control, 2011 First International Conference on*, pp. 180–183, 2011.
- [30] Instituto de Astrofísica de Andalucía. url: <http://www.iaa.es/es>.
- [31] Lawrence Berkeley National Laboratory. url: <http://www.lbl.gov/>.

# Parte II

## Apéndices



# Apéndice A

## Esquemáticos diseñados





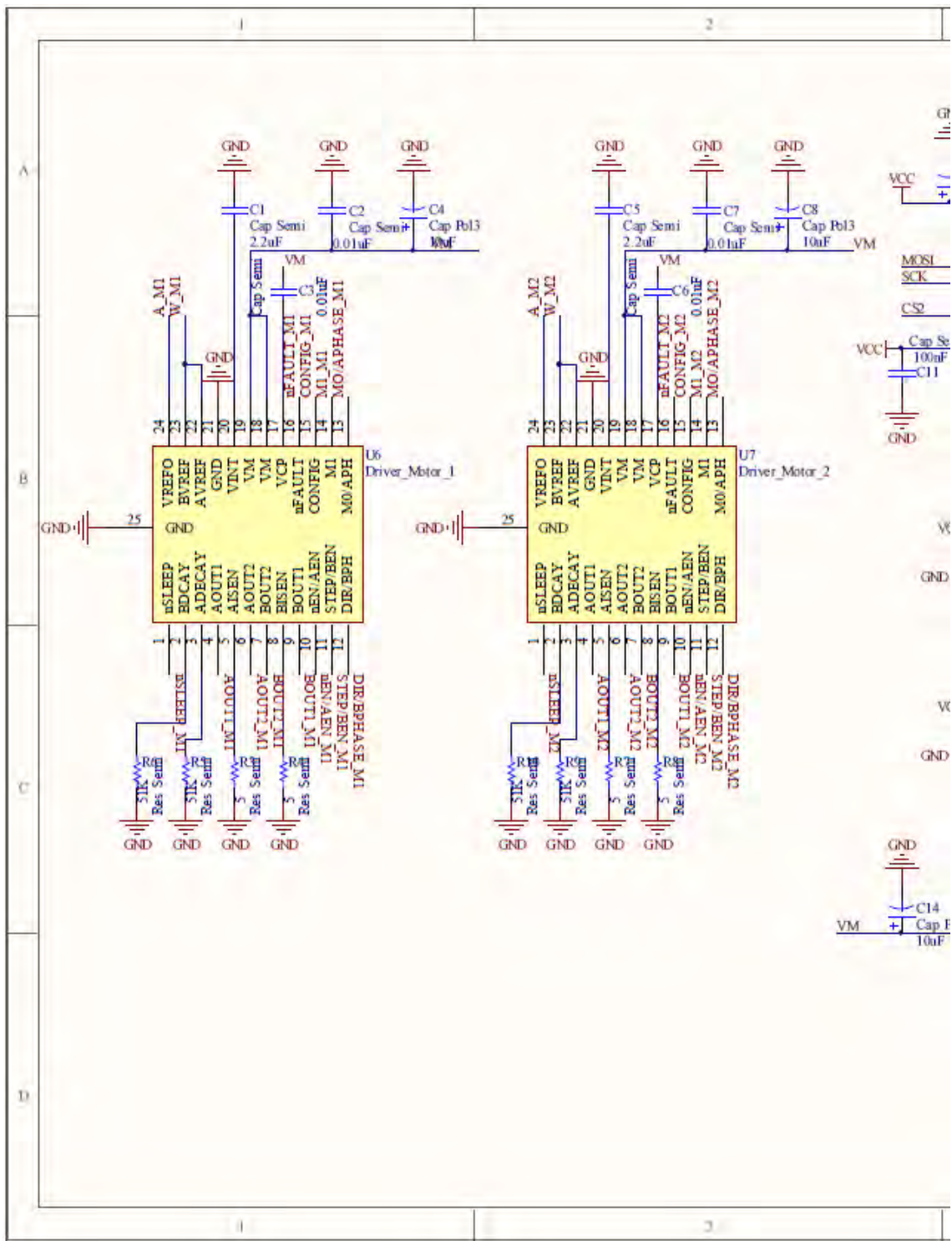


Figura A.1: Esquema del modelo v.1 (1ª mitad).

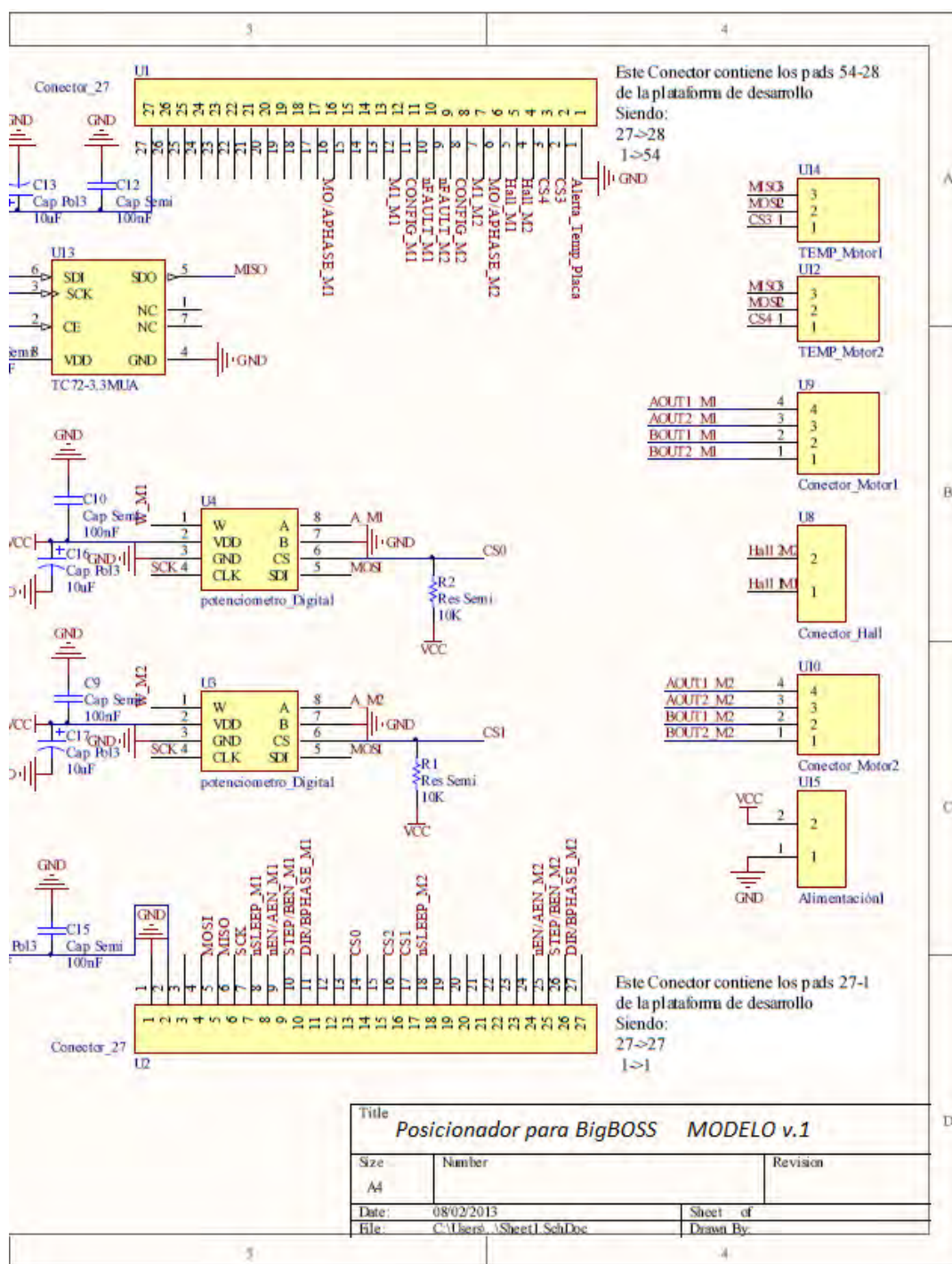


Figura A.2: Esquema del modelo v.1 (2ª mitad).

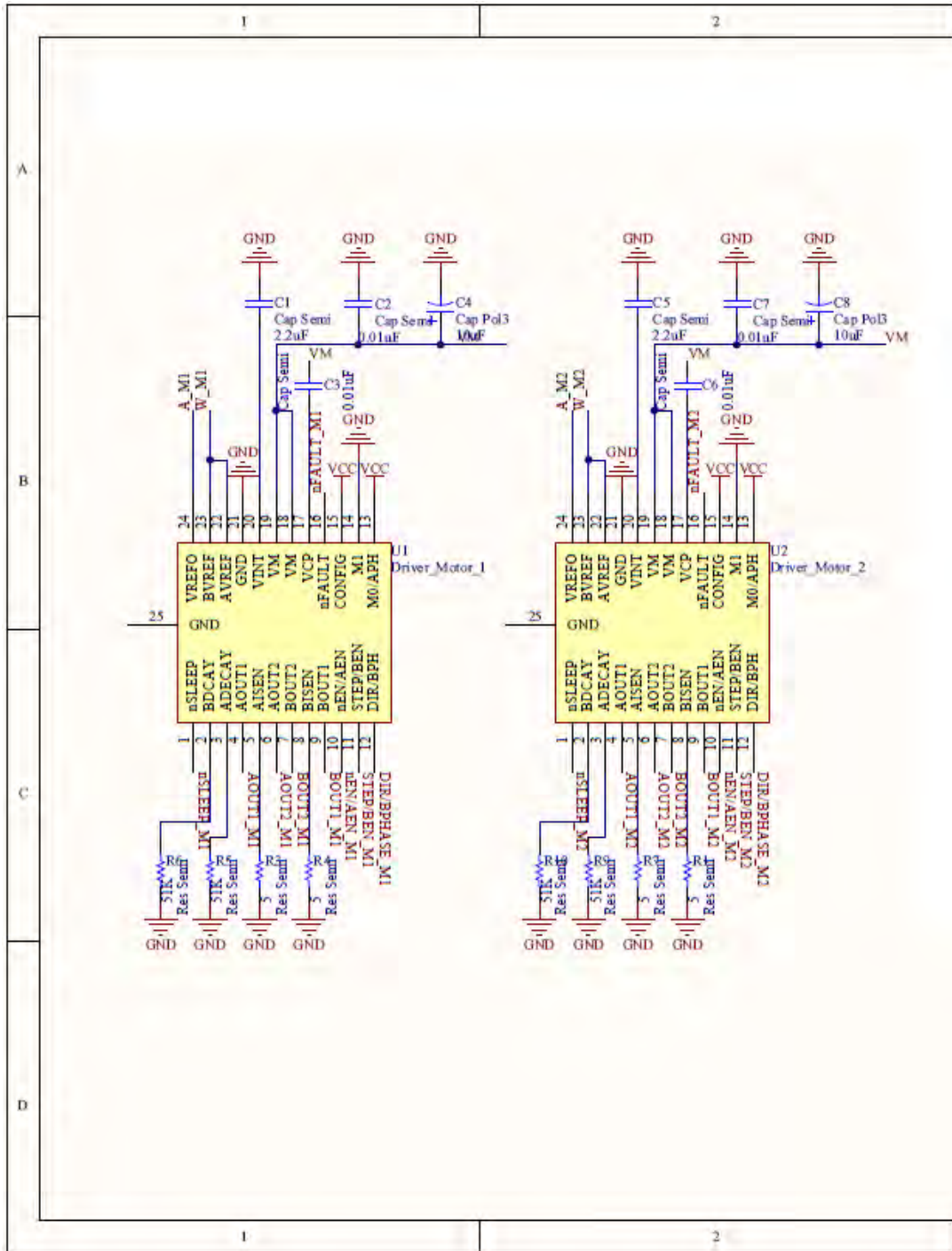


Figura A.3: Esquema del modelo v.2 (1ª hoja, 1ª mitad)

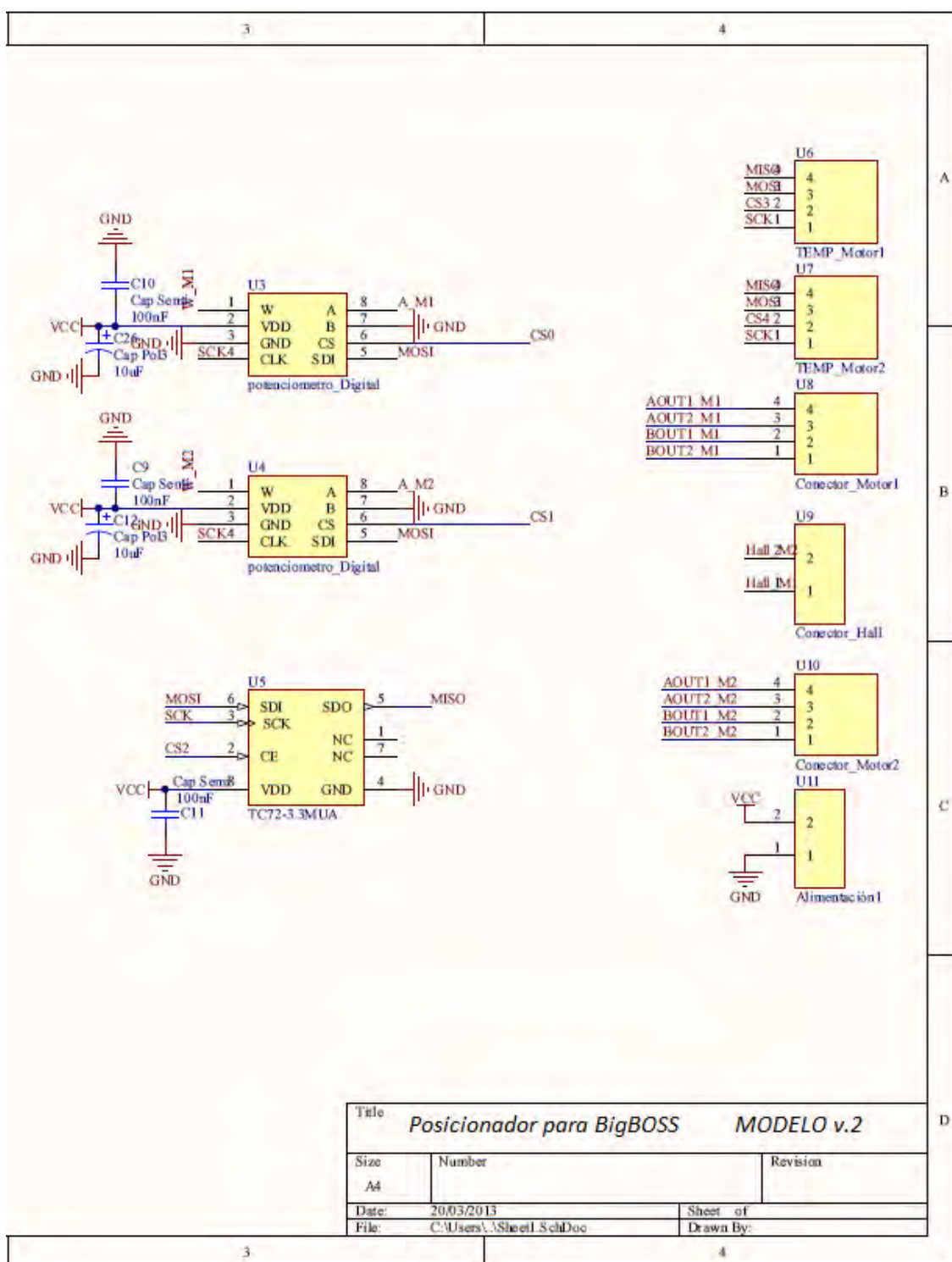


Figura A.4: Esquema del modelo v.2 (1ª hoja, 2ª mitad)

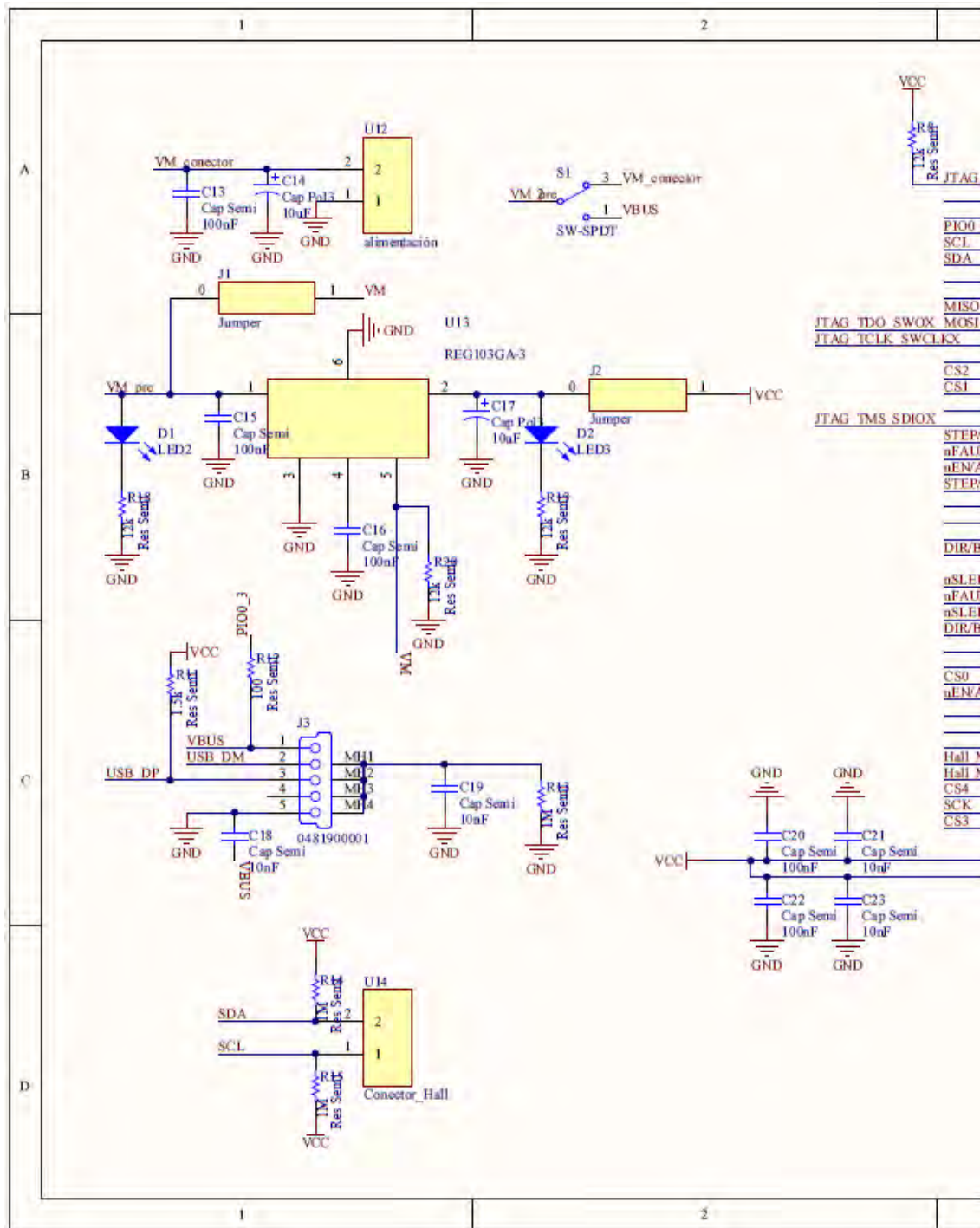


Figura A.5: Esquema del modelo v.2 (2ª hoja, 1ª mitad)

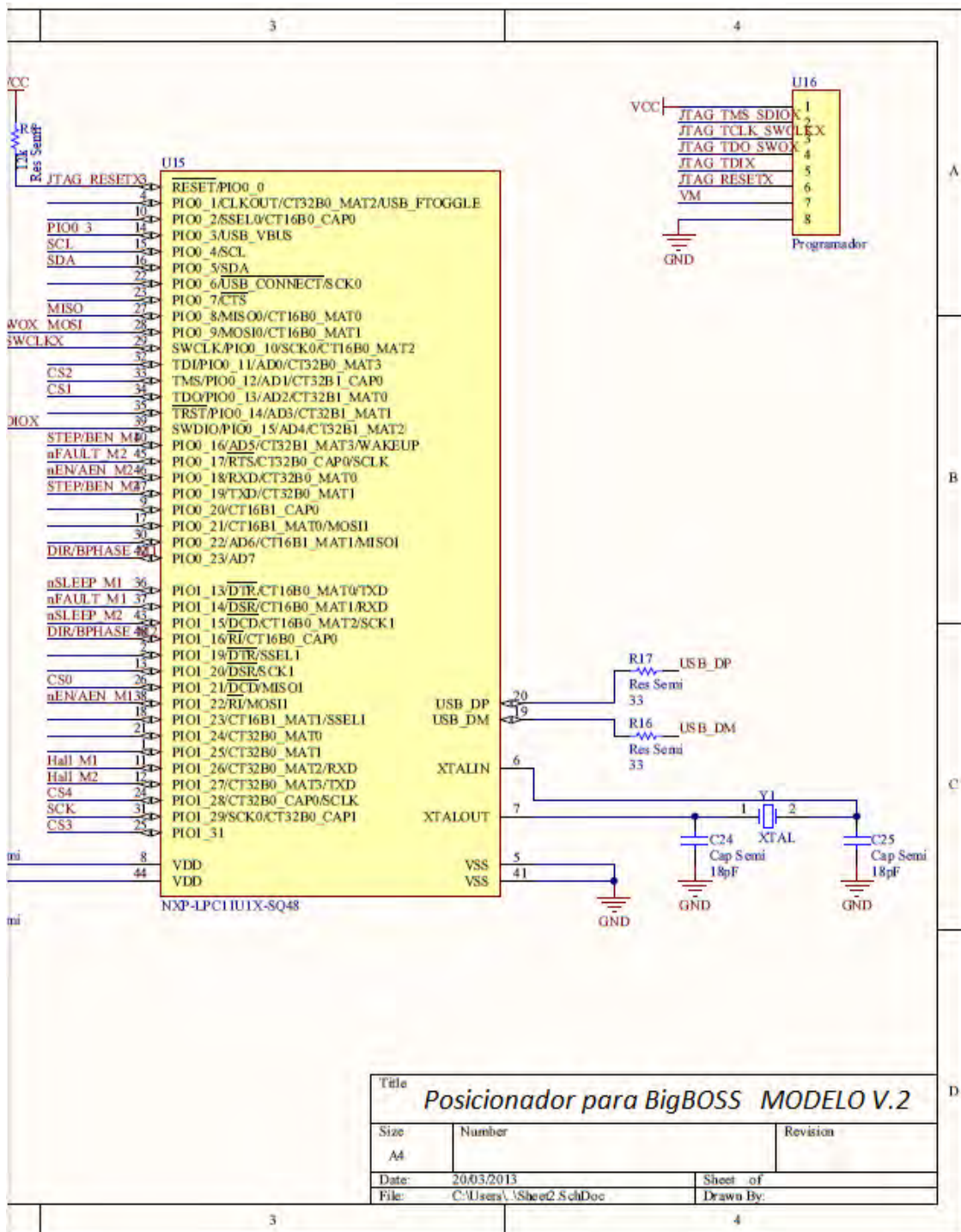


Figura A.6: Esquema del modelo v.2 (2ª hoja, 2ª mitad)

# Apéndice B

## Esquemáticos LPCXpresso





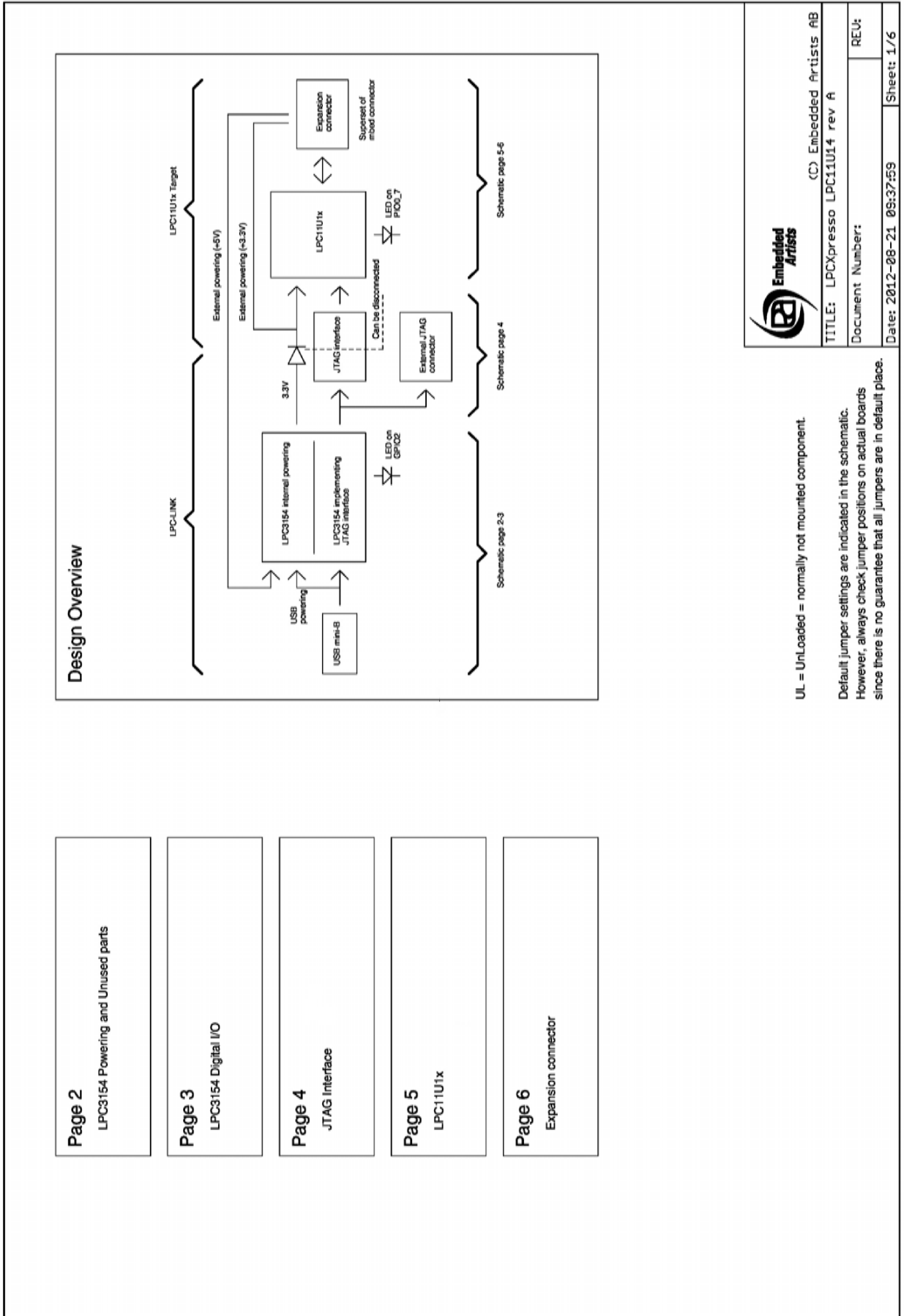


Figura B.1: Esquema general.

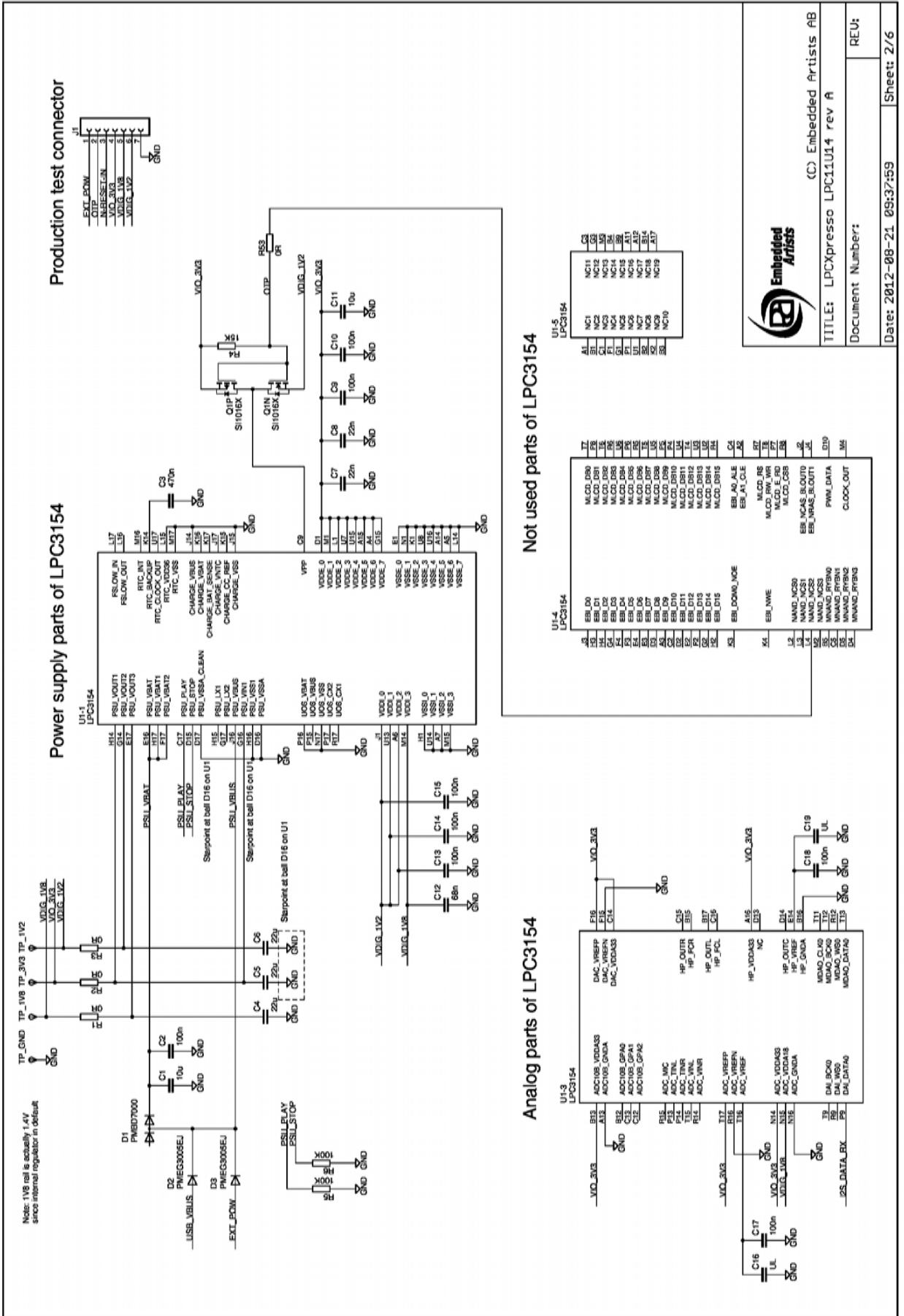


Figura B.2: Etapa de potencia.

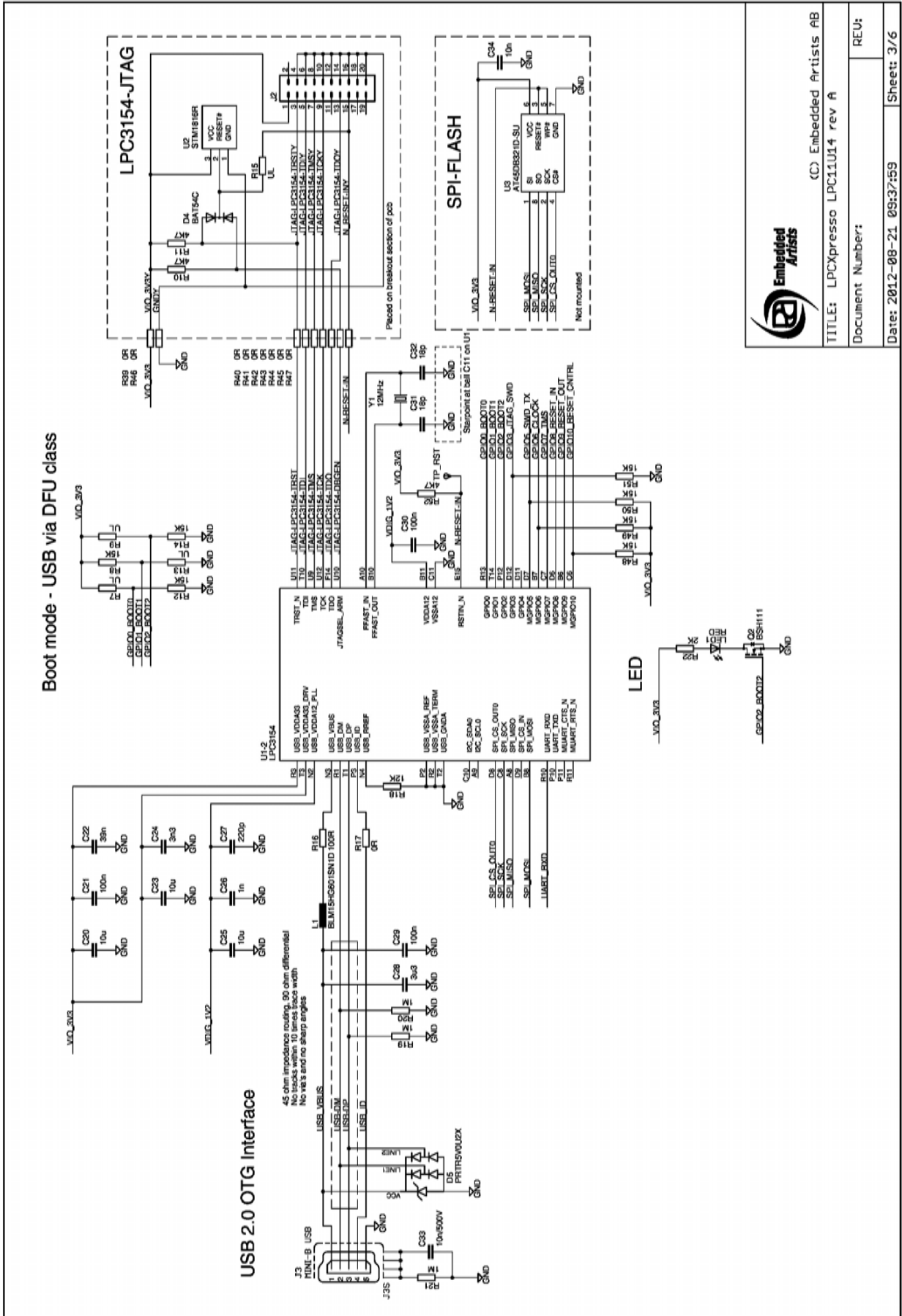


Figura B.3: Interface USB.

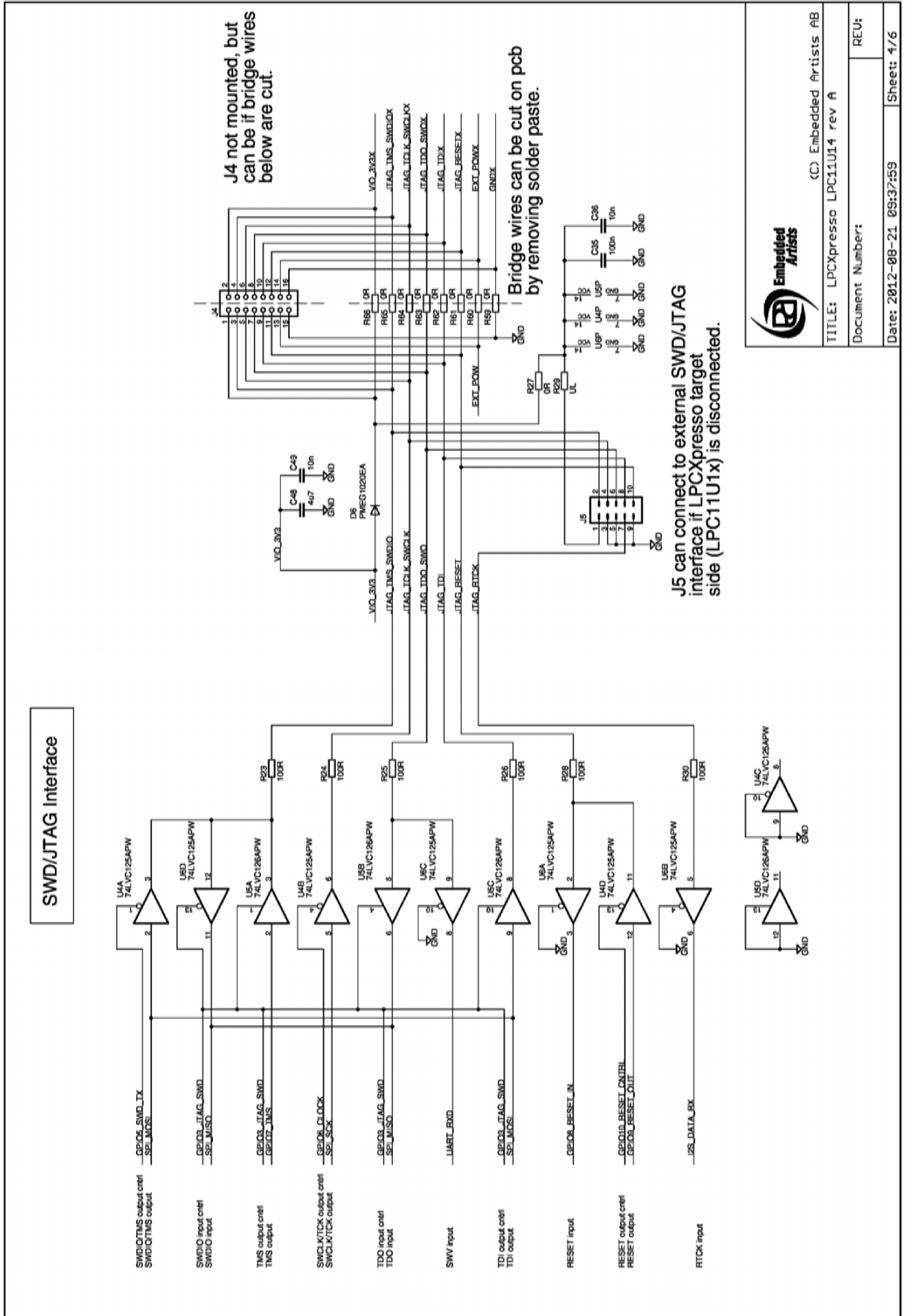
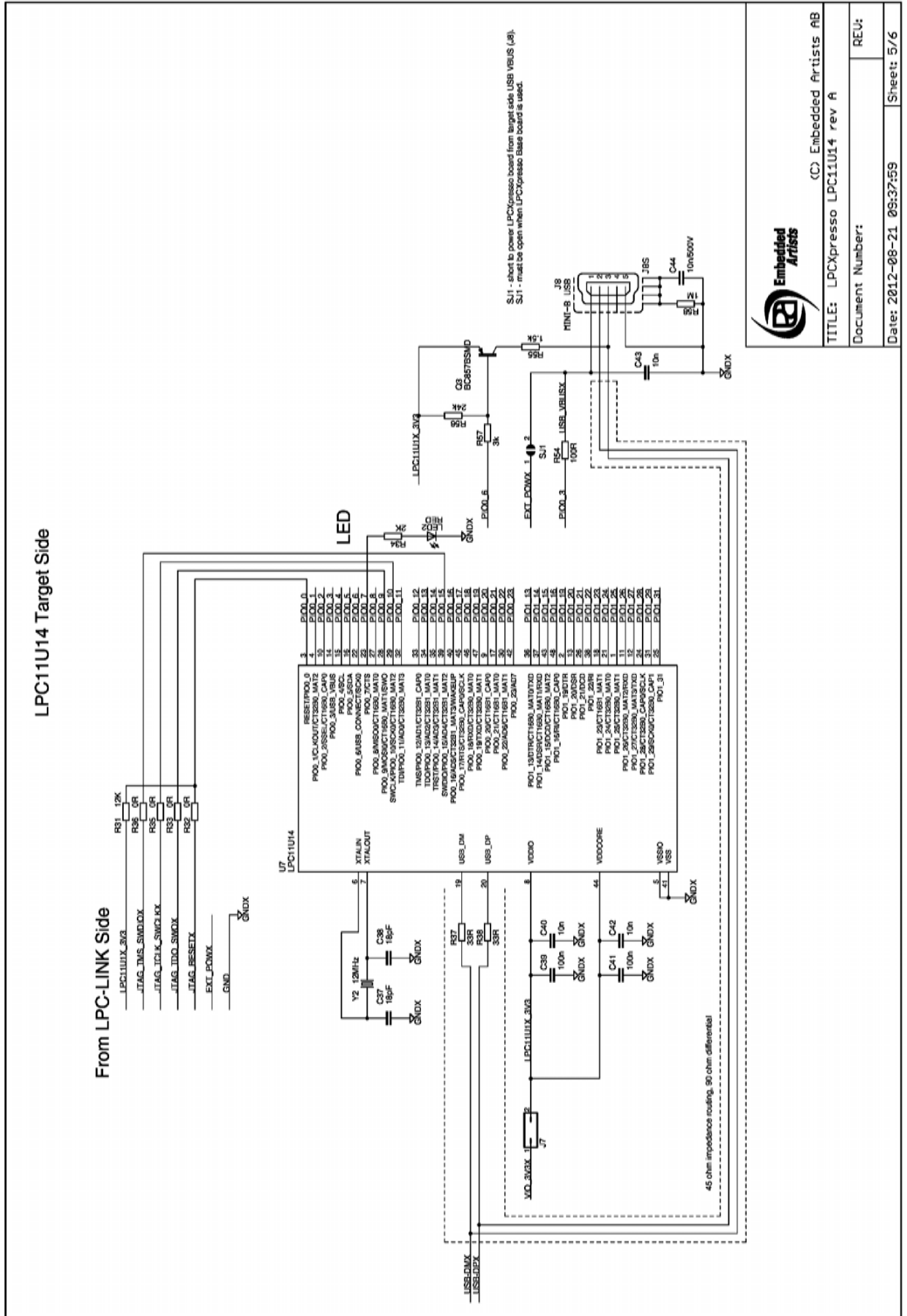


Figura B.4: Conexión entre placas.




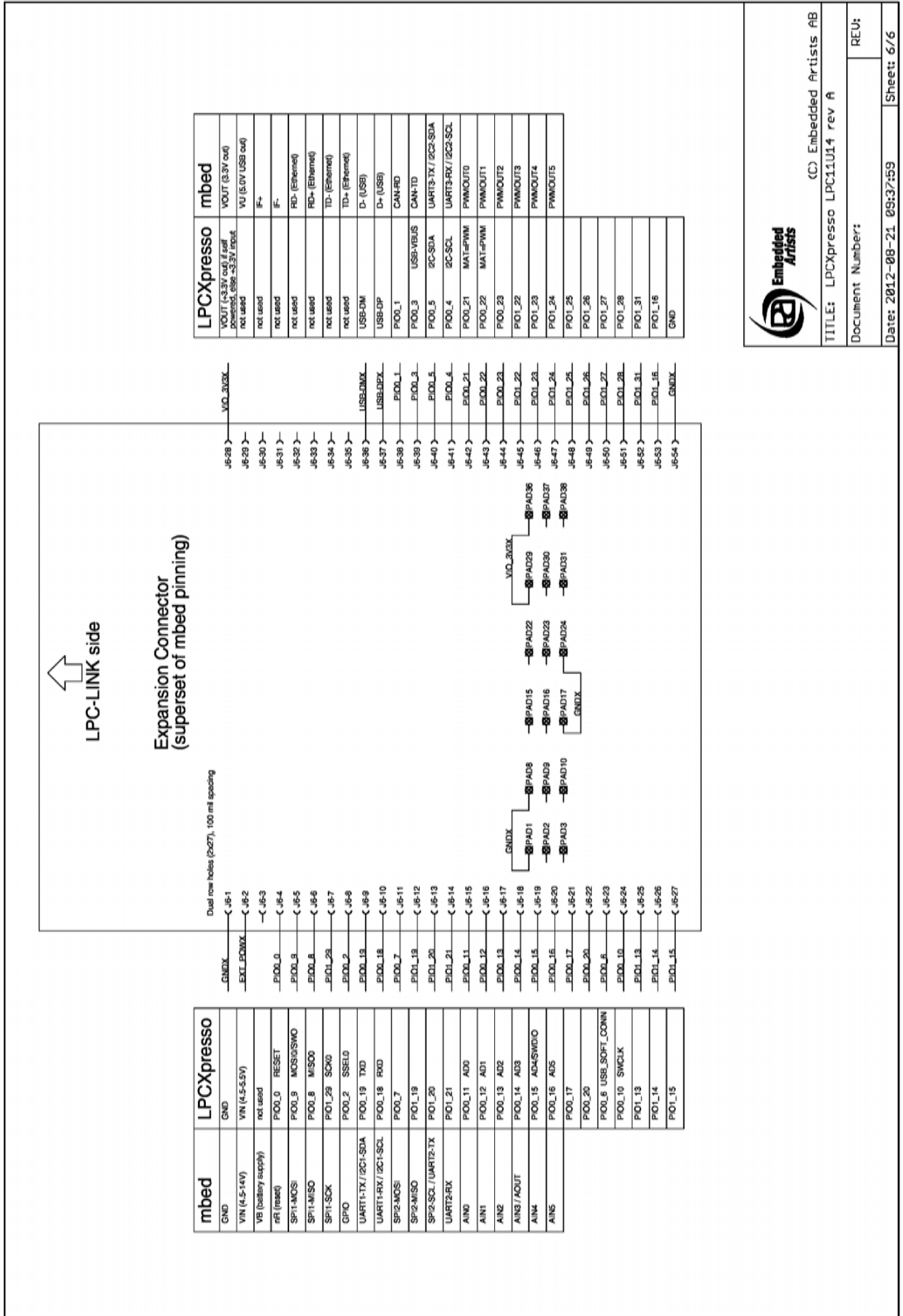

  
 (C) Embedded Artists AB  
 TITLE: LPCXpresso LPC111U14 rev A  
 Document Number:  
 Date: 2012-08-21 09:37:59  
 Sheet: 5/6

Figura B.5: Conectores de expansión.



Embedded Artists

(C) Embedded Artists AB

TITLE: LPCXpresso LPC1114 rev A

Document Number:

REV:

Date: 2012-08-21 09:37:59 Sheet: 6/6

Figura B.6: Microcontrolador.

# Apéndice C

## Algoritmos de transformación





```

/*****
/*
/*      ALGORITMO DE TRASFORMACIÓN R1R2 en xy      */
/*
/*      in:  R1 y R2 (grados)                      */
/*      out: x e y (mm)                            */
/*
/*****

double [] R1R2enxy(double R1, double R2)
{
    double R1d, R2d, alpha = 0, beta = 0, r = 0, x=0, y=0;
    double [] xy = { 0, 0 };

    R1    = Math.Round(R1, 3);
    R2    = Math.Round(R2, 3);
    R1d   = R1*2*Math.PI / 360;
    R2d   = R2*2*Math.PI / 360;
    r     = Math.Sqrt(Math.Pow(L1, 2)+Math.Pow(L2, 2) -2*L1*L2*Math.Cos(R2d)
    );
    beta  = Math.Asin(L2*Math.Sin(R2d)/r);
    alpha = R1_d+beta;

    if (alpha >= 2*Math.PI)
        alpha=alpha -2*Math.PI;

    if (Math.Pow(L1, 2)+Math.Pow(L2, 2) == 2*L1*L2*Math.Cos(R2d)) {
        x = 0;
        y = 0;
    }

    else{
        x = r*Math.Cos(alpha);
        y = -r*Math.Sin(alpha);
    }

    xy[0] = x;
    xy[1] = y;
    return xy;
}

```

```

/*****
/*
/*      ALGORITMO DE TRASFORMACIÓN xy en R1R2      */
/*
/*      in:  x e y      (mm)                        */
/*      out: R1 y R2   (pasos)                      */
/*
/*
/*****

int [] xy_en_R1R2(double x1, double y1)
{
    int [] R1R2 = {0,0 };
    double alpha=0,beta=0,r=0,R1=0,R2=0,R1_intermedio=0,R2_intermedio=0;

    r = Math.Sqrt(Math.Pow(x1,2)+Math.Pow(y1,2));
    if (r<=L1+L2&& r>=L1-L2)
    {
        if (x1==0&&y1==0){
            R1 = 0;
            R2 = 0;
        }
        else
        {
            alpha = Math.Atan(Math.Abs(y1)/Math.Abs(x1));
            beta = Math.Acos((Math.Pow(L1,2)+Math.Pow(r,2)-Math.Pow(L2,2))
                /(2*L1*r));
            R2 = Math.Acos((Math.Pow(L1,2)-Math.Pow(r,2)+Math.Pow(L2,2))
                /(2*L1*L2));
            if (x1>=0&&y1>=0){ //1º CUADRANTE
                R1=(2*Math.PI)-beta-alpha;
            }
            else if (x1<0&&y1>=0){ //2º CUADRANTE
                R1 = Math.PI+alpha-beta;
            }
            else if (x1<0&&y1<0){ //3º CUADRANTE
                R1 = Math.PI-alpha-beta;
            }
            else if (x1>=0&&y1<0){ //4º CUADRANTE
                R1 = alpha-beta;
            }
        }
    }
    else{
        R1 = 100000;
        R2 = 100000;
    }

    R1 = R1*360/(2*Math.PI);
    R2 = R2*360/(2*Math.PI);
    if (R1<0) R1 = 360+R1;
    R1_intermedio = Math.Round((R1*27520/360),0);
    R2_intermedio = Math.Round((R2*5120/180),0);
    R1R2[0] = Convert.ToInt32(R1_intermedio);
    R1R2[1] = Convert.ToInt32(R2_intermedio);
    return R1R2;
}

```

## Apéndice D

### Trasferencia tecnológica



A QUIEN CORRESPONDA, certifico que

El Proyecto Fin de Carrera realizado por D. Nasib Fahim Fernández en la Escuela Politécnica Superior de la UAM, dirigido por el profesor Guillermo González de Rivera Peces, titulado "*Electrónica de Control de un Mini-Robot para el posicionamiento micrométrico de una fibra óptica en un punto determinado del plano focal de un telescopio*" ha servido como diseño de la solución propuesta por el grupo español que participa en el proyecto BigBOSS como respuesta al problema presentado para el control electrónico de cada uno de los micro-robots que se van a encargar del posicionado micrométrico del conjunto de las fibras ópticas situadas en el plano focal de un telescopio.

Tras la realización de varios prototipos, el modelo actual ya ha superado la mayoría de las pruebas a las que ha sido sometido tanto en el laboratorio del grupo de investigación de la EPS donde se está desarrollando, como en el Lawrence National Berkeley Laboratory por el grupo de investigación que lidera el proyecto. Por lo tanto, se puede considerar que se ha realizado una contribución tecnológica a un proyecto internacional. Una vez superada todas las pruebas, se va a realizar la propuesta de instalarlo en cada uno de los 5.000 robots que componen el conjunto de posicionadores del plano focal.

El proyecto BigBOSS está liderado por el Lawrence National Berkeley Laboratory y forman parte varias instituciones de EEUU y diferentes grupos de investigación de China, Francia, Corea, España y el Reino Unido. El objetivo del proyecto es la creación del mayor mapa espectrográfico del universo para el estudio de la naturaleza de la energía oscura.

Y para que así conste y a petición del interesado, expido el presente certificado en Madrid, a 31 de Mayo de 2013.



Fdo.: Francisco Prada Martínez  
 Investigador Científico CSIC y Profesor CEI UAM  
 Instituto de Física Teórica UAM/CSIC



# Apéndice E

## Publicación

Nasib Fahim, Guillermo Glez.-de-Rivera, Javier Garrido,  
Justo Sánchez, Francisco Prada

**«Mechatronic for micrometric optical fiber positioning in a  
telescope focal plane»**

Enviado a International Conference on Design of Circuits and  
Integrated Systems (DCIS'13)  
San Sebastian, Spain, Nov. 2013.





DCIS 2013 Submission 74

<https://www.easychair.org/conferences/submission.cgi?a=4445323;su...>**DCIS 2013 (author)**[Paper 74](#) [DCIS 2013](#) [Alerts](#) [EasyChair](#)[Help](#) [Sign out](#)**DCIS 2013 Submission 74**[Update information](#)[Update authors](#)[Withdraw](#)

If you want to **change any information** about your paper or withdraw it, use links in the upper right corner.

**Paper 74**

Title: Mecatronics for micrometric optical fiber positioning in a telescope focal plane

Paper: [PDF](#)

Motor control  
Micrometric positioning

Keywords: Mechatronics

Low power design  
Sensors and actuators

Topics: Industrial Applications, Integrated Sensors & Actuators, Low Power Design

Abstract: The NOAO 4-meter Mayall Telescope (Kitt Peak, Arizona, EEUU) will be the basis of the worldwide research project BigBOSS [1], witch allows the simultaneous observation of thousands of galaxies and quasars and provide significant advances in the study of the Dark Energy. To achieve this, BigBOSS uses an array of 5000 mini-robots responsible for positioning optical fibers. These, will focalice the radiation of each target into a three spectrograph system that will analyse the obtained signals. To do this, these positioners are able to place the fibers with an accuracy of 5 m, within a maximum time of 16 s and with very low consumption levels. This Surpasses the performance of all mechatronic positioners designed for this type of applications.

Time: May 10, 13:36 GMT

**Authors**

first name	last name	email	country	organization	Web site	corr
Guillermo	Glez-De-Rivera	guillermo.gdrivera@uam.es	Spain	Universidad Autónoma de Madrid	<a href="http://www.hctlab.com">http://www.hctlab.com</a>	✓
Nasib	Fahim	nasib.fahim@estudiante.uam.es	Spain	Universidad Autónoma de Madrid	<a href="http://www.hctlab.com">http://www.hctlab.com</a>	✓
Javier	Garrido	javier.garrido@uam.es	Spain	Universidad Autónoma de Madrid	<a href="http://www.hctlab.com">http://www.hctlab.com</a>	✓
Justo	Sánchez	justo@iaa.es	Spain	Instituto de Astrofísica de Granada - CSIC		
Francisco	Prada	fprada@iaa.es	Spain	Campus Internacional de Excelencia UAM-CSIC		

## Mechatronics for micrometric optical fiber positioning in a telescope focal plane

Nasib Fahim, Guillermo Glez-de-Rivera,  
Angel de Castro, Javier Garrido  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
Madrid, Spain  
Email: guillermo.gdrivera@uam.es

Justo Sanchez  
Instituto de Astrofísica de Andalucía  
Granada, Spain  
Email: justo@iaa.es

Francisco Prada  
Campus of International Excellence  
UAM + CSIC  
Cantoblanco  
Madrid, Spain  
Email: fprada@iaa.es

**Abstract**—The *NOAO 4-meter Mayall Telescope (Kitt Peak, Arizona, EEUU)* will be the basis of the worldwide research project *BigBOSS* [1], [2], which allows the simultaneous observation of thousands of galaxies and quasars and provide significant advances in the study of the Dark Energy. To achieve this, BigBOSS uses an array of 5000 mini-robots responsible for positioning optical fibers. These will focus the radiation of each target into a three spectrograph system that will analyse the obtained signals. To do this, these positioners are able to place the fibers with an accuracy of  $5\ \mu\text{m}$ , within a maximum time of 16 s and with very low consumption levels. This surpasses the performance of all mechatronic positioners designed for this type of applications.

### I. INTRODUCTION

After more than 10 years of research and project approach, the creation of the largest spectroscopic map of the deep universe has been launched under the name of "*BigBOSS*" (*Big Baryon Oscillation Spectroscopic Survey* [1]). This, based on the progress made by the *SDSS-III* into his survey *BOSS*, will collect 20 million galaxies and quasars redshifts, to make a map with a volume 10 times higher than the best achieved to date. This offers important elements and extreme precision distances for the scientific community.

To achieve these objectives, BigBOSS will use the *NOAO 4-meter Mayall Telescope (Kitt Peak, Arizona, USA)* on which will be mounted an spectrograph system that will study in depth the properties of the universe and shed light about the nature and properties of the Dark Energy.

In order to study simultaneously the radiation emitted by thousands of celestial bodies, focusing on discrete points of the focal plane of the telescope (stars, galaxies and quasars), the project BigBOSS will use 5000 optical fibers. They will be guided to the exact point of light exposure through ad hoc robots. The array of robots will be installed in a 1 m in diameter metal mesh (figure 1) that is the focal surface of the telescope.

The position of each robot will be decided in a central unit that stores the relative positions of the different galactic bodies to study and generates a motion pattern for each robot.

Because of the structure of the robot's arms, it exists an specific shared position area between each robot and

its neighbours. For this reason, after pattern generation and verification of the absence of collisions, the central unit notifies each robot the exact position where the fiber must be placed within its range of action. After sending this command, the master software will run a repositioning algorithm to avoid collision between the positioner arms of two or more adjacent robots, sending, if necessary, a new position order. After the placement, the central unit backlights each one of the 5000 fibers using an internal light system and checks the correct position of these, through an image analysis performed by a CCD camera attached to the front of the telescope, which checks the relative position of each fiber according to a static pattern of positions. This will used to calibrate each robot for future uses and to study the error in the image obtained.

Once the positioning of the fibers in the focal plane of the instrument is made, it will begin an exposure of approximately 20 minutes from observed field. After this, a new repositioning of the fibers will start, in order to focus the fibers for a new exposure of 5000 new targets.

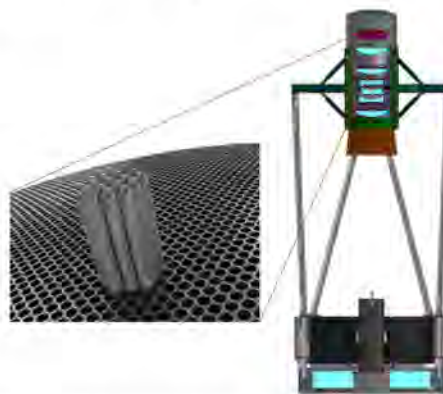


Fig. 1. Telescope optical lay-out and focal plane detail with a cluster of 19 robots (courtesy of AVS)

The stated goal of this project is to achieve the overall

positioning of the 5000 fibers in less than 60 seconds. This involves a challenge, because the smaller times achieved to date, in similar positioners, are in the order of several minutes, with a much smaller number of fibers. Apart of that, very low levels of power consumption are necessary, because the high number of existing robots in the focal plane.

This paper will describe the mechatronics of the coplanar positioner developed by the Spanish team for the BigBOSS project. On the other hand, it will explain the innovations from the electronics point of view that will be provided to the project, and also will present preliminary experimental results obtained during the last performed test ran with the positioner.

## II. COPLANAR POSITIONER

The 5000 actuators responsible for positioning the fiber at the point of exposure are mechatronic systems (figure 2) that should be installed in the focal plane mesh. They are composed of two mini-stepper motors that put every fiber in his exposure point coplanarly, based on the model  $\Theta - \Theta$ . That is, both engines move the fiber in the same plane, angularly, with two degrees of freedom.



Fig. 2. Coplanar Positioner designed by AVS in collaboration with the IAA-CSIC

The project requirement limits the positioning of the 5000 fibers in less than 60 seconds. However, this fiber positioner places in up to 16 seconds. Apart of that, it must be added the repositioning time, required for the placement algorithm (to avoid collisions between neighbour systems), and the latency time it takes for the information to arrive from the central unit to each 5000 actuators. In table I, the performance increase

TABLE I. COMPARATIVE TABLE OF SIMILAR PROJECTS

Project	Model	Number of fibers	Precision	Time
LAMOST [4]	$\Theta - \Theta$	4000	40 $\mu\text{m}$	600 s
Cobra [5]	$\Theta - \Phi$	2400	5 $\mu\text{m}$	7 s
BigBOSS-Spain [1]	$\Theta - \Theta$	5000	5 $\mu\text{m}$	<60 s
BigBOSS-Berkeley [3]	$R - \Theta$	5000	5 $\mu\text{m}$	<60 s

achieved in this new system is compared to similar projects [4], [3], [5], [1].

The engines used in this positioner are mini-stepper motors. They have a clear advantage over those used in similar projects; The fact of not being piezoelectric motors such as those used in the Cobra project [5], enables the use of *chopping current* (section III). This process consists to send to the engines pulsed signals. These signals are generated increasing the current to a pre-configured maximum and making it zero after reaching this maximum, creating a modulated PWM signal to the needs of each coil in this motor. This process is generated either in movement or when the motor is stopped (being able to generate only in the first case), guaranteeing to keep the engine warm in certain situations (bad weather, etc.). All this introduces a great innovation, and will be detailed in section III.

Furthermore, each motor can generate 25 nNm of maximum torque. However, the positioner is designed to support a maximum total torque of 10 nNm, which will be sufficient to position the fiber with errors of 5  $\mu\text{m}$ .

The fiber placement on the positioning circumference of each robot is performed by rotating a main axis R1 placed at the positioner centroid (figure 3), from which extends a main arm L1 of 3,464 mm ideally. And a secondary axis R2 placed at the end of the main arm, from which extends the secondary arm L2 of 3,464 mm ideally, in the end of which will place the fiber head responsible for capturing the radiation.

The main axis rotation is performed counter-clockwise, with a maximum angle of  $360^\circ$  or 27520 steps on the motor 1. Furthermore, the secondary axis rotation is performed clockwise with an angle of  $180^\circ$  or 5120 steps on the motor 2.

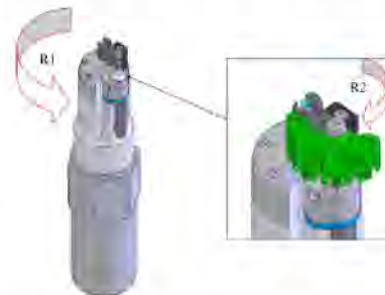


Fig. 3. Detail of the coplanar positioner arm

However, both motor 1 and motor 2 can be sent beyond

360° and 180° respectively, up to the physical limit of the system (365° and 185°). This situation is extremely delicate, because an error in the step can potentially cause physical damage on the positioner. For this reason, a 5° security angle is kept on each motor and the controller software does not allow to exceed 360° and 180°.

Finally, the optical fiber is guided to the exposure point, on the end of arm L2 through a pipe located inside the robot chassis. It ensures that the fiber will not suffer undesired twisting caused by the movement of the positioner axes, maintaining as far as possible the signal clean out of distortion and minimizing losses.

### III. CONTROL ELECTRONICS

All placement orders will be sent from superior communication nodes through an  $I^2C$  bus tree [6]. This receives commands from the host computer and distributes them to each actuator, through multiplexers and repeaters systems. The control electronics designed for these positioners will be responsible for, on the one hand, processing received commands and act accordingly, and on the other hand, constantly monitoring the temperature and position of the actuator, alerting of any unexpected value, in order to avoid fatal damage.

The main limitations presented in this project are, among others, the surface of the hardware and the current consumption of it. Therefore, due to the large number of positioners that coexist in the focal plane of the telescope, the consumption of each unit must not be very high. On the other hand, because the proximity of the actuators, the hardware should be as smaller as possible, having to be encapsulated in the chassis of each robot, so its surface will not exceed 1.5 cm<sup>2</sup>, respecting the 0.8 cm internal diameter of the cylindrical chassis. Furthermore, the temperature of both actuators and the electronics must be very accurately monitored, because an increase on these will introduce distortions in the obtained image. For all this, the design of these electronics is based on low power and the ability to integrate its components in the specified area.



Fig. 4. Prototype to final electronics design

The main component of the system is the LPC1114, a 32-bit microcontroller from NXP, in charge of receiving commands from the top node, processing them and managing

information flowing into the actuator. In this electronics can also highlight two low power and high performance drivers DRV8834 from Texas Instruments, responsible of giving sufficient current to each motor, monitoring possible overload in them and controlling the chopping current as well as operating modes (half, full or microstep). These drivers are supported on two digital potentiometers AD5160 from Analog Devices, which control the voltage drop across two of its control terminals, determining the maximum chopping current each driver will deliver to the motor. Finally, the electronics include three temperature sensors TC72 from Microchip, one embedded in the PCB board and one installed in each motor.

To optimize the consumption of the electronics, the control system can run in two modes, Active and Sleep. The first one is used to process data, to monitor constants and to position the engine, whereas the second one is used in waiting command periods, reducing consumption to a minimum when not required. Tables II and III show the consumption of each part of the system.

Considering the motors and drivers supply of 5 V and 3.3 V for the other electronics, reached levels of total power consumption are shown in table IV.

TABLE II. COMPONENTS CURRENT CONSUMPTION IN mA

Component	Active	Sleep
LPC1114	2	0.36
DRV8834	2.4	0.0006
TC72	0.25	0.0001
AD5160	0.003	0.003

TABLE III. OVERALL CURRENT CONSUMPTION IN mA

Active	Sleep
7.56	0.37

TABLE IV. OVERALL POWER CONSUMPTION IN mW

Mode	Motor 1	Motor 2	Drivers	Electronics	Total
Running	200.000	170.000	24.000	9.095	403.095
Sleeping	0.000	0.000	0.006	1.224	1.218

The first innovative feature of this system is to control the temperature by software limiting the chopping current. This can be modulated to the needs of each motor windings over periods of movement or steadily in standby moments when the system is waiting for commands. In the second case, limiting the current to low values, it is able to maintain a constant temperature in engines without moving, adapting it to environmental characteristics. Thus, under conditions of very low ambient temperature, the chopping current is limited in stationary periods, at high values, increasing the temperature in each engine and offsetting the damage in mobility that freezing may cause in the actuators mechanism, while they remain unemployed waiting for new moves. Moreover, handling the chopping current, allows to control the torque generate for each motor, offering solutions to problems occurring in freezing conditions.

Finally, the microcontroller and mechanics used allow to optimize positioning time versus other control systems that

preceded this. However, we have implemented a decisive algorithm, that allows to achieve the best times results, respecting the robot integrity, reducing the position errors and ensuring not to lose steps.

Due to the design of the mechanics of each actuator, the R2 axis is induced a number of steps with the R1 axis movement. This could make, during rotation of R1, to overcome the physical limit of R2. To avoid this situation, we have designed control algorithms embedded in electronics. They continuously monitor the position of the two axis and move accordingly, correcting undesired movement in the R2 axis and optimizing axis positioning time.



Fig. 5. Safe Operating Area

Figure 5 represents the SOA (Safe Operating Area). This is the operation area within the robot can perform, without damaging the mechanism. This figure represents the actual steps (RR2) in the R2 axis (steps in R2 with the induced ones), versus the steps in the R1 axis.

In order to fill all the functions described above, this system bases its performance on the commands management received from superior communications nodes. During the middle stages of the electronics development, control commands that managed in detail all system operation have been programmed. As in other communication protocols, these commands are formed by three bytes, one with the header and two with data. The command set presented in table V shows the basic commands to control all functions. However, the final design of these electronics does not have all these features, because they have been integrated into the internal system control algorithms.

TABLE V. COMMAND SET

Command	Data	Description
MOVE	Absolute position	Sends an absolute position
TEMP	Sensor number	Read a temperature value
CCHOP	Potenciometer voltage value	Set the maximum chopping current
SCHOP	Control mode	Set the chopping mode
VEL	Steps/second value	Set the motor speed

#### IV. EXPERIMENTS AND RESULTS

For this system, several preliminary performance tests have been run at the *Instituto de Astrofísica de Andalucía* [9] (IAA-

CSIC) and the *Lawrence Berkeley National Laboratory* [8] (LBNL), proving that the positioner does not lose steps and places the fiber in a bounded box error. These were performed in environments with temperature and vibration controlled by operating the system for hours taking it to high stress levels. These tests have been carried out in two phases, the first one, without performing a previous calibration and the second one, calibrating the positioner according to the actual length of the arms L1 and L2 in order to find the robot exact centroid.

First, positioning tests (figure 6 and 7) have been performed to estimate the existing error between the positions reached by the actuator and the theoretical one. On the one hand, in blue, target movements are represented, on the other, in green the real position reached by the robot.

After performing nine times the same positioning test shown in figure 6, calibrating the variables involved after the completion of each test, significant improvements are achieved (figure 7), coming to get rms errors of about  $7 \mu\text{m}$  (table VI).

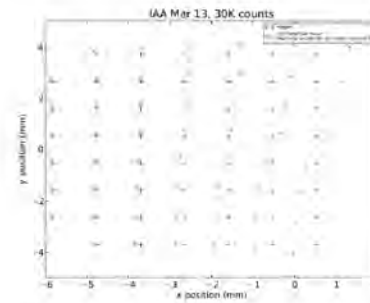


Fig. 6. Position test without calibration

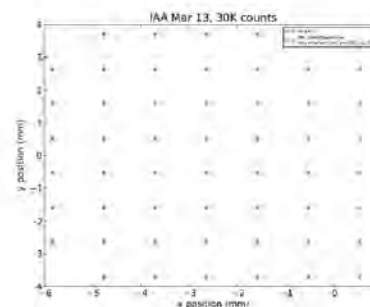


Fig. 7. Position test after calibration

TABLE VI. ERRORS IN  $\mu\text{m}$  REACHED IN THE TESTS

Test	RMS	Mean	Maximum
1	320.0	99.6	880.2
2	108.8	10.9	287.4
3	47.9	2.8	149.6
4	26.8	5.0	105.0
5	16.6	3.0	70.3
6	11.4	1.3	46.3
7	10.4	1.3	29.4
8	9.6	2.6	33.9
9	7.5	1.6	21.7

On the other hand, it is important to stand out the effect of hysteresis that appears in the position of the fiber, when it is placed at the same point coming from different positions. This effect is caused by the inner workings of mechanics. However, the error introduced is framed within the accepted limits.

Several tests have been conducted to study the effects of hysteresis (figure 9) and the system repeatability, bringing the robot to a same central point from four equidistant points (figure 8). In them, error boxes are obtained by movement up to  $4 \mu\text{m} \times 0.75 \mu\text{m}$  and errors introduced by hysteresis between  $20 \mu\text{m}$  and  $40 \mu\text{m}$ . However, after the completion of these tests, it is observed the good repeatability of the system.

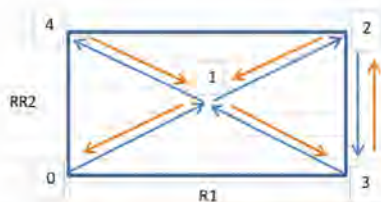


Fig. 8. Repeatability test points



Fig. 9. Repeatability test

Finally, different tests have been performed to study the

positioner tilt errors respect to the light incidence plane of the fiber head (figure 10). Again, after performing a calibration of the positioner variables to get the real value of the robot centroid, a significant improvement in these errors achieving values as  $0.06^\circ$  of maximum error,  $0.035^\circ$  of mean error and  $0.019^\circ$  of RMS error can be seen.

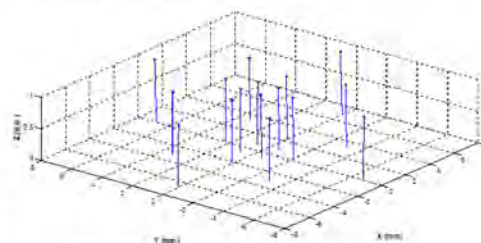


Fig. 10. Tilt test after calibration

## V. CONCLUSIONS

Making a comparison with similar positioners that currently exist, the system designed for the BigBOSS project introduces clear improvements.

On the one hand, the positioning time achieved with the electronics and the coplanar system allows to optimize the process, substantially increasing the performance of the system, allowing to take a larger number of images in a shorter time. Moreover, at present, systems involving independent robotic positioners dedicated to each fiber have a very high consumption. However, the orientation of this design to micro-consumption, allows to achieve very low levels, limiting the overall system power consumption of 5000 units and minimizing the heat dissipation in the focal plane. Finally, the design of the temperature control by controlling the chopping current guarantees better performance in adverse conditions, reducing the need for maintenance of the system and securing the capacity for positioning and taking images in all situations.

On the other hand, studies with the coplanar positioner and control electronics reveal the need for positioners calibration prior to final installation in the telescope's focal plane, due to the non-ideality of the components used. However, the results obtained by performing various tests stand out the good system repeatability and the low value of the errors obtained in different positions.

## VI. ACKNOWLEDGMENTS

This work has been supported by the Spanish Ministerio de Ciencia e Innovación under project TEC2009-09871.

We thank the support of the spanish MICINN's Consolider-Ingenio 2010 Programme under grant MultiDark CSD2009-00064. Also we want to acknowledge the collaboration of Faulhaber and AVS, and the special support in the performance tests of the LBNL-Berkeley.

## REFERENCES

- [1] D. Schlegel, et al. *The BigBOSS Experiment*, arXiv 1106.1706 , 2011.
- [2] D. J. Schlegel, C. Bebek, H. Heetderks, S. Ho, M. Lampton, et al. *BigBOSS: The Ground-Based Stage IV Dark Energy Experiment*, arXiv 0904.0468 , 2009.
- [3] Silber, Joseph H. et al. *Design and performance of an R-0 fiber positioner for the BigBOSS instrument*, Proc. SPIE 8450, Modern Technologies in Space- and Ground-based Telescopes and Instrumentation II, 845038 (September 13, 2012); doi:10.1117/12.926457
- [4] X. Xing, C. Zhai and H. Du. *Parallel controllable optical fiber positioning system for LAMOST*. Proc. SPIE 3352, Advanced Technology Optical/IR Telescopes VI, 839 (August 25, 1998); doi:10.1117/12.319309
- [5] C. Fisher, D. Braun, J. Kaluzny, and T. Haran, *Cobra: A two-degree of freedom fiber optic positioning mechanism*, in Aerospace conference, 2009 IEEE, pp. 111, IEEE, 2009.
- [6] Fang Zhao, Daquan Deng, Zhensheng Wang, Huanzhao Liu. *Design of Schematic Mapping System Based on I2C and USB Bus*, Instrumentation, Measurement, Computer, Communication and Control, 2011 First International Conference on, pp.180,183, 21-23 Oct. 2011.
- [7] *FAULHABER Miniature Drive Systems*. <http://www.faulhaber.com/>
- [8] *Lawrence Berkeley National Laboratory*. <http://www.lbl.gov/>
- [9] *Instituto de Astrofísica de Andalucía*. <http://www.iaa.es/es>





**Apéndice F**

**Presupuesto**



PRESUPUESTO GLOBAL

Concepto	Coste (€)
<b>Ejecución Material</b>	
Compra de ordenador personal	600
Compra de licencia del software de diseño	1200
Fabricación del modelo V.1 de la electrónica	50.6
Fabricación del modelo V.2 de la electrónica	30.7
Total ejecución material	1881.3
<b>Gastos generales</b>	
16 % sobre Ejecución Material	301.0
<b>Beneficio industrial</b>	
6 % sobre Ejecución Material	112.9
<b>Honorarios de Proyecto</b>	
1200 h a 15 €/h	19000.0
<b>Material fungible</b>	
Gastos de impresión	50.0
Encuadernación	50.0
<b>Subtotal del Presupuesto</b>	
Subtotal del Presupuesto	<b>21395.2</b>
<b>I.V.A aplicable</b>	
21 % subtotal del Presupuesto	4493
<b>Total del Presupuesto</b>	<b>25888.2</b>

Cuadro F.1: Presupuesto global del proyecto desarrollado

Madrid, Julio 2013

El Ingeniero Jefe de Proyecto

Fdo.: Nasib Fahim Fernández

Ingeniero Superior de Telecomunicación



# Apéndice G

## Pliego de condiciones



## **G.1. Pliego de condiciones**

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un «Electrónica de control de un mini-robot para el posicionamiento micrométrico de una fibra óptica en el plano focal de un telescopio». En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

### **G.1.1. Condiciones generales**

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.
2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.
3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.
4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.
5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.
6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.
7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades

sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.
9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.
10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.
11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.
12. Las cantidades calculadas para obras accesorias, aunque figuren por partidaalzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.
13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.
14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.
15. La garantía definitiva será del 4 % del presupuesto y la provisional del 2 %.
16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.
17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la



provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.
19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.
20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.
21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.
22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.
23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad «Presupuesto de Ejecución de Contrata» y anteriormente llamado «Presupuesto de Ejecución Material» que hoy designa otro concepto.

### **G.1.2. Condiciones particulares**

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.
3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.
4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.
5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.
6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.
7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.
8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.
9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.
10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.
11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.
12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar