

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



PROYECTO FIN DE CARRERA

Sistema de comunicación para la gestión remota de un elevado número de Mini-Robots de posicionamiento en un entorno reducido.

Ingeniería de Telecomunicación

Jesús Castro Murillas

Junio 2013

**Sistema de comunicación para la gestión remota de un elevado
número de Mini-Robots de posicionamiento en un entorno
reducido**

**AUTOR: Jesús Castro Murillas
TUTOR: Guillermo González de Rivera Peces**

**Grupo HCTLab
Dpto. de Tecnología Electrónica y de las comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio de 2013**

Resumen:

En este proyecto se realiza un sistema de comunicación que permite conectar una unidad central con unos 5.000 dispositivos (Mini-Robots) en un entorno reducido. Para ello se han comparado algunas de las distintas tecnologías disponibles actualmente, con el fin de lograr una eficiencia tanto en el consumo de potencia como en el tiempo de configuración de los dispositivos, asegurando la detección y retransmisión en caso de errores. Se han comparado tanto soluciones Wireless (ZigBee y CyFi) como soluciones Wired (I2C, Ethernet y CAN). La configuración de los dispositivos se decide desde una unidad central, la cual enviará la información a los distintos dispositivos. Para ello la unidad central (PC) se conectará con seis módulos llamados Control Devices mediante Ethernet. Cada módulo es el encargado de configurar sus correspondientes dispositivos mediante I2C.

Para ello cada Control Device tendrá un microcontrolador y seis multiplexores I2C con el que se accederá a los diferentes canales de I2C donde cuelgan los dispositivos finales a configurar.

Abstract:

This project is performing a communication system that connects a central unit with around 5,000 devices (Mini-Robots) in a small environment. Some of the different technologies available today have therefor been compared, in order to achieve an efficiency of power consumption and configuration time of the devices, ensuring detection and retransmission in case of error. The solutions Wireless (ZigBee and CyFi) as Wired solutions (I2C, Ethernet and CAN) have been compared. The devices configuration is decided from a central unit, which will send the information to different devices. For this the central unit (PC) will be connected with six modules called Control Devices via Ethernet. Each module is responsible for configuring their devices via I2C.

Each Control Device has a microcontroller and six I2C multiplexers to access to the different channels of I2C, connected with the end devices to be configured.

Palabras Clave:

BigBOSS, I2C, Multiplexing, Ethernet, BeagleBone.

Agradecimientos:

En primer lugar dar las gracias a mi tutor Guillermo González de Rivera, por la posibilidad de incorporarme a este Proyecto Fin de Carrera y por transmitirme su conocimiento y dedicación por la electrónica. Porque sin todas esas horas debatiendo sobre el proyecto no habría sido posible completarlo de manera tan satisfactoria. También quiero agradecer a todos los profesores que he tenido, tanto en la universidad como en el colegio, por aportarme los conocimientos necesarios para completar esta etapa de mi vida y por ayudarme a ser la persona que soy.

A mis amigos y compañeros por todos los buenos y malos momentos, por aquellas noches en la biblioteca y por aquellos viajes que serán un gran recuerdo dentro de unos años.

Y por último a mi familia, por el apoyo recibido desde siempre en cualquier decisión que he tomado, por estar ahí cuando más lo he necesitado y por ayudarme a conseguir mis objetivos en la vida. Sin vosotros me habría sido imposible conseguir todo lo que he conseguido y todo lo que conseguiré.

INDICE DE CONTENIDOS

1	Introducción.....	1
1.1	Motivación.....	1
1.2	Objetivos.....	3
2	Estado del arte	5
2.1	Tipos de arquitecturas.....	7
2.1.1	Estrella.....	7
2.1.2	Malla.....	8
2.1.3	Árbol.....	9
2.1.4	Bus	10
2.1.5	Anillo	11
2.2	Tecnologías actuales de comunicación.....	12
2.2.1	Tecnologías Wireless.....	12
2.2.1.1	Zigbee	12
2.2.1.2	Estudio sobre viabilidad.....	12
2.2.2	Tecnologías Wired.....	15
2.2.2.1	Bus I2C.....	15
2.2.2.2	Bus CAN	17
2.2.2.3	Ethernet.....	18
2.2.3	Tecnologías Mixtas.....	21
3	Análisis de soluciones	23
3.1	Comparativa de tiempos	25
3.1.1	I2C	28
3.1.1.1	Tiempo OffLine:.....	29
3.1.1.2	Tiempo Live. Tabla Entera:	31
3.1.1.3	Tiempo Live. Posición:	33
3.1.2	BUS CAN	34
3.1.2.1	Tiempo OffLine	35
3.1.2.2	Tiempo Live. Tabla Entera	37
3.1.2.3	Tiempo Live. Posición.	39
3.1.3	CyFi + I2C.	40
3.1.4	Zigbee	42
3.1.5	Ethernet + I2C:	43
3.2	Resumen	45
4	Diseño.....	47
4.1	Solución Final.....	49
4.1.1	Control Device.....	51
4.1.2	Multiplexor I2C	52
4.1.3	Regulador.....	52
4.2	Consumo.....	53
4.3	Herramientas.....	53
5	Desarrollo	55
5.1	Elementos del prototipo.....	57
5.1.1	BeagleBone.....	57

5.1.2 Multiplexor I2C	58
5.1.3 Regulador.....	60
5.1.4 Control Device.....	60
5.1.5 Periféricos I2C	61
5.1.5.1 Puerto entrada/salida de 8 bits	61
5.1.5.2 Sensor de temperatura	62
5.2 Diseño	63
5.2.1 Placa Multiplexores I2C v1	63
5.2.2 Placa Multiplexores I2C v2	65
5.2.3 Puerto entrada/salida de 8 bits	67
5.2.4 Sensor de temperatura	69
6 Integración, pruebas y resultados	71
6.1 Conexión entre los módulos	73
6.2 Software.....	74
6.3 Pruebas.....	75
7 Conclusiones y trabajo futuro.....	83
7.1 Conclusiones.....	85
7.2 Trabajo futuro	85
7.2.1 Integración microprocesador	85
7.2.2 Integración módulos	85
7.2.2.1 Repetidor I2C.....	86
7.2.2.2 Diseño.....	87
7.2.3 Futuras tecnologías	89
Referencias	- 91 -
Glosario	- 92 -
Anexos	93
A Esquemáticos	95
B Presupuesto.....	108
8 Certificación de transferencia tecnológica	115

INDICE DE FIGURAS

FIGURA 1 : TELESCOPIO BIGBOSS	1
FIGURA 2 : SISTEMA DE FIBRAS ÓPTICAS, BIGBOSS	1
FIGURA 3 : DIVISIÓN PLANO FOCAL.....	2
FIGURA 4 : DISPOSITIVO POSICIONADOR	2
FIGURA 5 : TIPOS DE ARQUITECTURA DE RED.....	7
FIGURA 6 : ARQUITECTURA EN ESTRELLA.....	7
FIGURA 7 : ARQUITECTURA EN MALLA.....	8
FIGURA 8 : ARQUITECTURA EN ÁRBOL	9
FIGURA 9 : ARQUITECTURA TIPO BUS.....	10
FIGURA 10 : ARQUITECTURA ANILLO	11
FIGURA 11 : ESQUEMA ANTENAS EN PLANO FOCAL.....	13
FIGURA 12 : DISTRIBUCIÓN DE LAS ANTENAS	13
FIGURA 13 : COEFICIENTE DE REFLEXIÓN DE ELEMENTO CENTRAL EN RETÍCULA DE 19. RADIANDO SOLO EL ELEMENTO CENTRAL.....	14
FIGURA 14 : COEFICIENTE DE REFLEXIÓN DEL ELEMENTO CENTRAL EN RETÍCULA DE 19. RADIANDO TODOS LOS ELEMENTOS.	14
FIGURA 15 : TRAMA I2C.....	16
FIGURA 16 : TRAMA BUS CAN.....	17
FIGURA 17 : COMPARATIVA I2C Y BUS CAN	18
FIGURA 18 : TRAMA ETHERNET	19
FIGURA 19 : TIPOS DE ETHERNET	20
FIGURA 20 : COMPARATIVA ZIGBEE Y CYFI.....	21
FIGURA 21 : FIBER VIEW CAMERA	25
FIGURA 22 : POSICIONADOR DE FIBRA.....	26
FIGURA 23 : INTERBLOQUEOS.....	26
FIGURA 24 : DIAGRAMA I2C	28

FIGURA 25 : TIEMPOS I2C.....	29
FIGURA 26 : DIAGRAMA BUS CAN.....	34
FIGURA 27 : TIEMPOS BUS CAN.....	35
FIGURA 28: DIAGRAMA CYFi+I2C.....	40
FIGURA 29 : TIEMPO CYFi+I2C.....	40
FIGURA 30 : DIAGRAMA ZIGBEE.....	42
FIGURA 31 : TIEMPOS ZIGBEE.....	42
FIGURA 32 : DIAGRAMA Y TIEMPOS ETHERNET + I2C.....	44
FIGURA 33 : COMPARATIVA DE TIEMPOS.....	45
FIGURA 34 : COMPARATIVA I2C Y BUS CAN. TIEMPO OFFLINE.....	45
FIGURA 35 : COMPARATIVA I2C Y BUS CAN. TIEMPO LIVE ENVIANDO TABLA DE POSICIONES.....	46
FIGURA 36 : COMPARATIVA I2C Y BUS CAN. TIEMPO LIVE ENVIANDO UNA POSICIÓN.....	46
FIGURA 37 : DISTRIBUCIÓN PLANO FOCAL.....	49
FIGURA 38 : DIAGRAMA ETHERNET + I2C.....	50
FIGURA 39 : DIAGRAMA CONTROL DEVICE.....	51
FIGURA 40 : DIAGRAMA MULTIPLEXOR I2C.....	52
FIGURA 41 : DIAGRAMA CONSUMOS SOLUCIÓN FINAL.....	53
FIGURA 42 : BEAGLEBONE.....	57
FIGURA 43 : ENCAPSULADOS PCA9548.....	58
FIGURA 44 : DIRECCIÓN MUX I2C FIGURA 45 : CONFIGURACIÓN CANALES I2C.....	59
FIGURA 46 : TRAMA CONFIGURACIÓN CANAL I2C.....	59
FIGURA 47 : DIAGRAMA REGULADOR DE POTENCIA.....	60
FIGURA 48 : DIAGRAMA CONTROL DEVICE.....	60
FIGURA 49 : DIAGRAMA PCA9554.....	61
FIGURA 50 : DIAGRAMA TCN75A.....	62
FIGURA 51 : ESQUEMÁTICO PCB MULTIPLEXOR I2C v1.....	63

FIGURA 52 : DISEÑO PCB MULTIPLEXOR I2C v1	64
FIGURA 53 : PCB MULTIPLEXORES I2C v1	64
FIGURA 54 : SECCIÓN ESQUEMÁTICO PCB MULTIPLEXOR I2C V2.....	65
FIGURA 55 : DISEÑO PCB MULTIPLEXOR I2C v2	66
FIGURA 56 : PCB MULTIPLEXORES I2C v2	66
FIGURA 57 : ESQUEMÁTICO PUERTO DE ENTRADA Y SALIDA DE 8BITS	67
FIGURA 58 : DISEÑO PCB PUERTO ENTRADAS Y SALIDAS DE 8 BITS	68
FIGURA 59 : ESQUEMÁTICO PCB SENSOR DE TEMPERATURA	69
FIGURA 60 : DISEÑO PCB SENSOR DE TEMPERATURA	70
FIGURA 61 : HEADER P9 BEAGLEBONE.....	73
FIGURA 62 : CONEXIÓN ENTRE LOS MÓDULOS	73
FIGURA 63 : CONEXION DE LOS PCBs	75
FIGURA 64 : CONEXIÓN ETHERNET	76
FIGURA 65 : CONFIGURACIÓN GPIO1_16 COMO RESET	77
FIGURA 66 : BUS I2C -3.....	77
FIGURA 67 : CONFIGURACIÓN PRIMER MULTIPLEXOR	78
FIGURA 68 : TRAMA I2C CONFIGURACIÓN MULTIPLEXOR 1	78
FIGURA 69 : CONFIGURACIÓN SEGUNDO MULTIPLEXOR.....	79
FIGURA 70 : TRAMA I2C CONFIGURACIÓN MULTIPLEXOR 2	79
FIGURA 71 : LECTURA DEL SENSOR DE TEMPERATURA	79
FIGURA 72 : TRAMA I2C LECTURA DEL SENSOR TEMPERATURA	79
FIGURA 73 : DIAGRAMA REPETIDOR PCA9517	86
FIGURA 74 : ESQUEMÁTICO PCB REPETIDOR I2C	87
FIGURA 75 : PCB REPETIDOR I2C	88
FIGURA 76 : PROTOTIPO CON MÓDULO REPETIDOR I2C.....	88

1 Introducción

1.1 Motivación

Desde la comunidad astrofísica internacional se desea conseguir una mejor comprensión del cosmos y del universo observable.

Debido al constante avance de las tecnologías cada vez se pueden desarrollar mejores herramientas las cuales permiten un constante avance técnico en este campo.

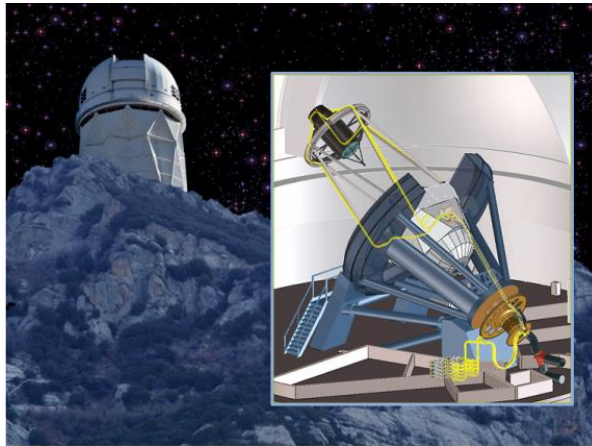


Figura 1 : Telescopio BigBOSS

Desde la universidad de Berkeley, junto a diversas universidades y organismos tanto públicos como privados a lo largo del planeta, han puesto en marcha la creación del mayor mapa espectrográfico del universo. Este proyecto se denomina “BigBoss”.

Los resultados obtenidos por BigBOSS abrirán nuevas líneas de investigación futura en el campo de la astronomía. Para alcanzar los objetivos del BigBOSS se utilizará el NOAO (National Optical Astronomy Observatory) 4-meter Mayall Telescope (Kitt Peak, Arizona, EEUU) sobre el que se montará un espectrógrafo que podrá estudiar en profundidad las propiedades del universo observable.

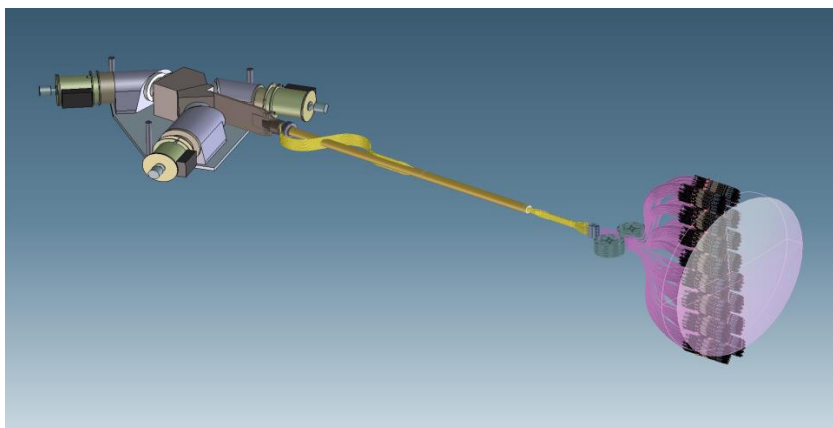


Figura 2 : Sistema de fibras ópticas, BigBOSS

El diseño de la interfaz encargada de la focalización y concentración de la radiación captada por el telescopio NOAO ha sido encargado a diferentes organismos a nivel mundial.

Con el objetivo de captar la radiación emitida, el proyecto BigBOSS se servirá de unas 5.000 fibras ópticas. Éstas, serán guiadas al punto exacto de exposición luminosa por medio de robots diseñados para tal caso, para cada una de las fibras. La nube de robots se instalará a través de una malla metálica de 1 metro de diámetro que abarcará 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 5.000 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 Madrid, es la labor principal del IFT (Instituto de Física Teórica) y el Grupo de Investigación HCTLab (Human Computer Technology Laboratory, grupo perteneciente a la Escuela Politécnica Superior), ambas entidades de la Universidad Autónoma de Madrid.

Los dispositivos del plano focal se agruparán en celdas hexagonales de 19 dispositivos con el fin de dividir el plano focal en sectores lógicos.

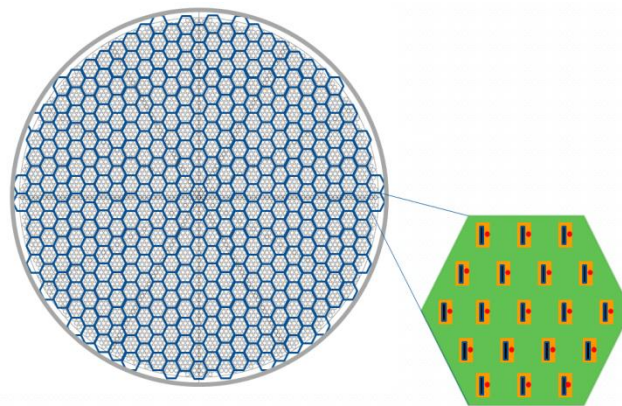


Figura 3 : División plano focal

Los dispositivos se encargarán mediante dos motores, de posicionar correctamente la fibra. Estos motores se controlarán mediante la electrónica del propio dispositivo.

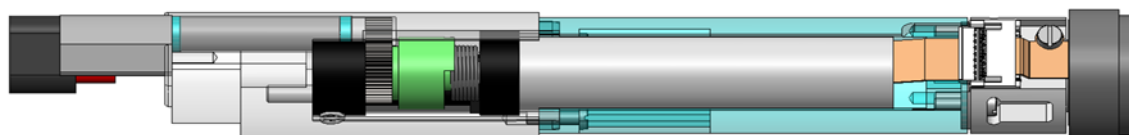


Figura 4 : Dispositivo posicionador

1.2 Objetivos

El objetivo planteado en el proyecto *“Sistema de Comunicación para la gestión remota de un elevado número de Mini-Robots de posicionamiento en un entorno reducido”* al Grupo de Investigación HCTLAB es el de diseñar un sistema de comunicación entre una unidad central y cada uno de los 5.000 Mini – Robots posicionadores de cada fibra que permita que los Mini – Robots este posicionados en un tiempo menor de 60s para la adquisición de la siguiente imagen.

La tarea de configuración consiste en dos etapas. En una primera etapa se configurarán y se moverán todos los dispositivos.

En una segunda etapa se detectará que todos los dispositivos se hayan posicionado correctamente, en caso negativo, los dispositivos erróneamente posicionados se volverán a configurar y posicionar. Se repetirá esta etapa hasta que todos los dispositivos estén correctamente posicionados.

Además también se podrá recibir información sobre los dispositivos (como la temperatura de los motores) con el fin de una mayor seguridad.

Para ello se estudiarán distintas tecnologías:

- Tipo radio:
 - Zigbee: Es un protocolo de comunicaciones inalámbricas, está pensado para comunicaciones a baja velocidad entre dos o varios dispositivos. Se pueden formar redes con miles de dispositivos comunicándose entre sí.
 - CyFi: CyFi es un ISM de 2,4 GHz de radio transmisor-receptor, capaz de soportar velocidades de datos de hasta 1 Mbps. Cuenta con una radio de 1-Mbps GFSK, buffering de paquetes de datos, packet framer, DSSS y RSSI.
- Tipo bus:
 - I2C: Es un bus de comunicaciones en serie que utiliza dos líneas para transmitir la información: una para los datos y por otra la señal de reloj.
 - Bus Can: Es un protocolo de comunicaciones basado en una topología bus para la transmisión de mensajes en entornos distribuidos.
 - Ethernet: Es un estándar de transmisión de datos para redes de área local.
- Estructura Mixta:
 - CyFi + I2C: Con una configuración híbrida se intenta solucionar el problema de alta densidad de antenas.

Con este estudio se verificará la viabilidad de cada una de ellas, al igual que las posibles ventajas y desventajas de cada una de ellas. El desarrollo e implementación de BigBOSS abrirá futuras líneas de investigación a nivel internacional.

2 Estado del arte

2.1 Tipos de arquitecturas

Como ya se ha comentado anteriormente, el objetivo es comunicar una unidad central con aproximadamente 5.000 dispositivos, para ello se estudiará todos los tipos de arquitecturas con el fin de encontrar la arquitectura que ofrece una mejor solución a nuestro problema. Se puede destacar cinco tipos de arquitecturas: Malla, estrella, árbol, bus y anillo [1].

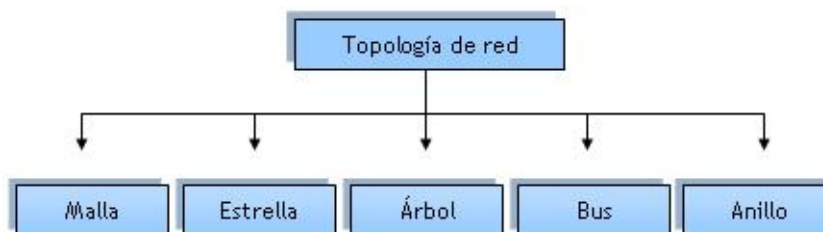


Figura 5 : Tipos de arquitectura de red

2.1.1 Estrella

En la topología en estrella cada nodo solamente tiene un enlace punto a punto dedicado con el controlador central, habitualmente llamado concentrador. Los nodos no están directamente enlazados entre sí.

Las principales ventajas de esta arquitectura son:

1. Si un nodo se desconecta o se rompe el cable solo queda fuera de la red aquel nodo.
2. Posee un sistema que permite agregar nuevos nodos fácilmente.
3. Reconfiguración rápida.
4. Fácil de prevenir daños y/o conflictos.
5. Centralización de la red.

La principal desventaja de esta arquitectura es que si el controlador central falla, toda la red deja de transmitir.

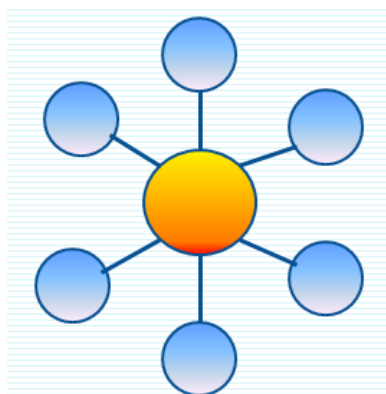


Figura 6 : Arquitectura en estrella

2.1.2 Malla

En una arquitectura en malla, cada nodo tiene un enlace punto a punto con cualquier otro nodo. De esta manera es posible llevar los mensajes de un nodo a otro por distintos caminos. Si la red de malla está completamente conectada, no puede existir absolutamente ninguna interrupción en las comunicaciones. Por tanto, una red en malla completamente conectada necesita $n(n-1)/2$ canales físicos para enlazar n dispositivos. Para acomodar tantos enlaces, cada dispositivo de la red debe tener tanto puertos de entrada como de salida.

Las principales ventajas de una arquitectura en malla son:

1. Es posible llevar los mensajes de un nodo a otro por diferentes caminos.
2. No puede existir absolutamente ninguna interrupción en las comunicaciones.
3. Cada servidor tiene sus propias comunicaciones con todos los demás servidores.
4. Si falla un cable el otro se hará cargo del tráfico.
5. No requiere un nodo o servidor central lo que reduce el mantenimiento.
6. Si un nodo desaparece o falla no afecta en absoluto a los demás nodos.
7. Si desaparece un nodo no afecta tanto a los nodos de redes.

La principal desventaja es el costo de implementar tantos enlaces y el mantenimiento de los mismos.

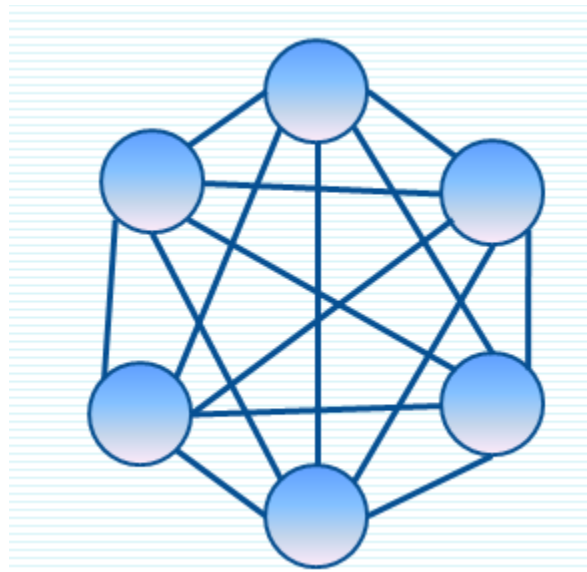


Figura 7 : Arquitectura en malla

2.1.3 Árbol

La red en árbol es una topología de red en la que los nodos están colocados en forma de árbol. Desde una visión topológica, es parecida a una serie de redes en estrella interconectadas salvo en que no tiene un nodo central. En cambio, tiene un nodo de enlace troncal, generalmente ocupado por un hub o switch, desde el que se ramifican los demás nodos. Las ramificaciones se extienden a partir de un punto raíz, a tantas ramificaciones como sean posibles, según las características del árbol.

Las principales ventajas de esta arquitectura:

1. El Hub central, al retransmitir las señales, amplifica la potencia e incrementa la distancia a la que puede viajar la señal.
2. Se permite conectar más dispositivos gracias a la inclusión de concentradores secundarios.
3. Permite priorizar y aislar las comunicaciones de distintas computadoras.
4. Cableado punto a punto para segmentos individuales.

La principal desventaja de esta arquitectura es que si el hub de un segmento falla, todas la comunicaciones con los nodos de ese segmento fallan.

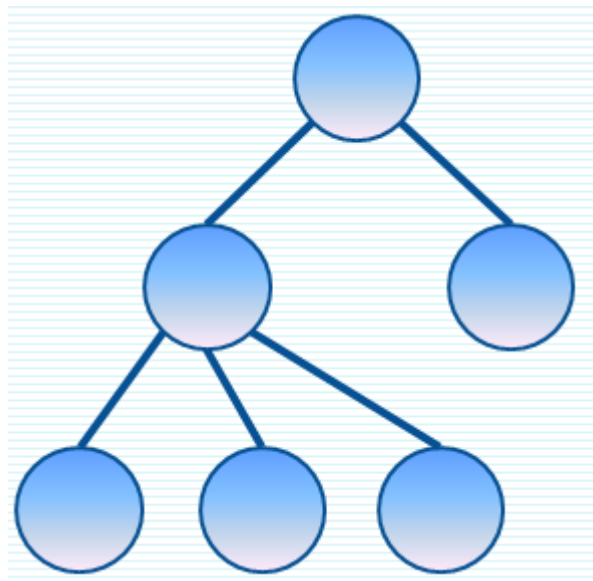


Figura 8 : Arquitectura en árbol

2.1.4 Bus

Una topología de bus es multipunto. Un cable largo actúa como una red troncal que conecta todos los dispositivos en la red.

Las principales ventajas de esta arquitectura son:

1. Facilidad de implementación y crecimiento.
2. Simplicidad en la arquitectura.

Por el contrario para un gran número de nodos esta arquitectura presenta bastantes desventajas:

1. Hay un límite de equipos dependiendo de la calidad de la señal.
2. Puede producirse degradación de la señal.
3. Complejidad de reconfiguración y aislamiento de fallos.
4. Limitación de las longitudes físicas del canal.
5. Un problema en el canal usualmente degrada toda la red.
6. El canal requiere ser correctamente cerrado (caminos cerrados).
7. Altas pérdidas en la transmisión debido a colisiones entre mensajes.

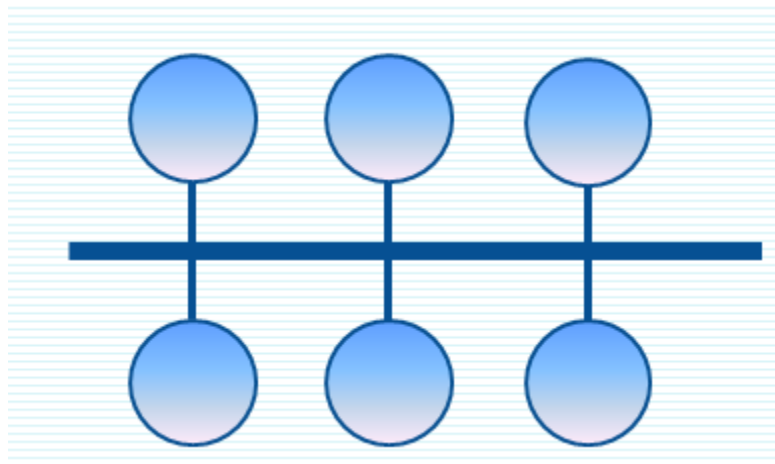


Figura 9 : Arquitectura tipo bus

2.1.5 Anillo

En una topología en anillo cada dispositivo tiene una línea de conexión dedicada y punto a punto solamente con los dos dispositivos que están a sus lados. La señal pasa a lo largo del anillo en una dirección, o de dispositivo a dispositivo, hasta que alcanza su destino. Cada dispositivo del anillo incorpora un repetidor.

Una mejora de esta arquitectura es implementar un anillo doble, el cual permite que la comunicación sea bidireccional.

Las principales ventajas de esta arquitectura son:

1. El sistema provee un acceso equitativo para todas las computadoras.
2. El rendimiento no decae cuando muchos usuarios utilizan la red.
3. Arquitectura muy sólida.

Las principales desventajas de esta arquitectura:

1. Longitudes de canales
2. El canal usualmente se degradará a medida que la red crece.
3. Difícil de diagnosticar y reparar los problemas.
4. Si una estación o el canal falla, las restantes quedan incomunicadas (Circuito unidireccional).

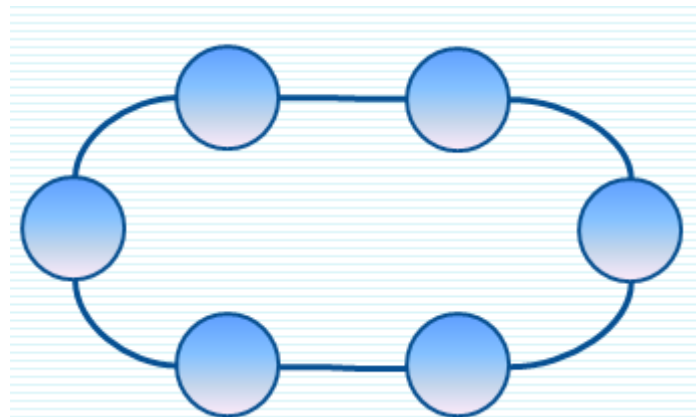


Figura 10 : Arquitectura anillo

2.2 Tecnologías actuales de comunicación

A continuación se estudiarán algunas de las distintas tecnologías (tanto *Wireless* como cableadas) con el fin de encontrar la tecnología cuyas características planteen la mejor solución para nuestro problema (consumo, velocidad, viabilidad...).

2.2.1 Tecnologías Wireless

2.2.1.1 Zigbee

Una de las opciones más interesantes es la opción mediante tecnología Zigbee. ZigBee es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación [2]. Las características que lo diferencian de otras tecnologías son:

1. Su bajo consumo.
2. Su topología de red en malla.
3. Su fácil integración (se pueden fabricar nodos con muy poca electrónica).

ZigBee permite tres topologías de red:

1. Topología en estrella: el coordinador se sitúa en el centro.
2. Topología en árbol: el coordinador será la raíz del árbol.
3. Topología de malla: al menos uno de los nodos tendrá más de dos conexiones.

La topología más interesante de Zigbee (y una de las causas por las que parece que puede triunfar ZigBee) es la topología de malla. Ésta permite que si, en un momento dado, un nodo del camino falla y se cae, pueda seguir la comunicación entre todos los demás nodos debido a que se rehacen todos los caminos. La gestión de los caminos es tarea del coordinador.

2.2.1.2 Estudio sobre viabilidad

Al estudiar la solución mediante Zigbee, el principal problema que se encuentra es la alta densidad de antenas (5.000 antenas en un entorno reducido). Para ello se realizó una simulación cuyo resultado fue la inviabilidad de esta solución debido a la alta distorsión que provocarían las antenas entre sí [3].

Se trata de una disposición en forma de nido de abeja, dispuesta en hexágonos contiguos. La distancia entre centros de hexágonos es de 12 mm. En cada uno de esos hexágonos se sitúa una fibra óptica sujeta a una estructura mecanizada de posicionamiento, para cuyo control se ha establecido la utilización de un módulo radio. La posición del módulo radio se identifica con el círculo color azul emplazado en el centro de cada hexágono.

La frecuencia de utilización de las antenas se centra en 2.4 GHz, donde existen sistemas radio de comunicaciones para controlar dispositivos robotizados capaces de desplazar las fibras ópticas según los requisitos del radio-telescopio.

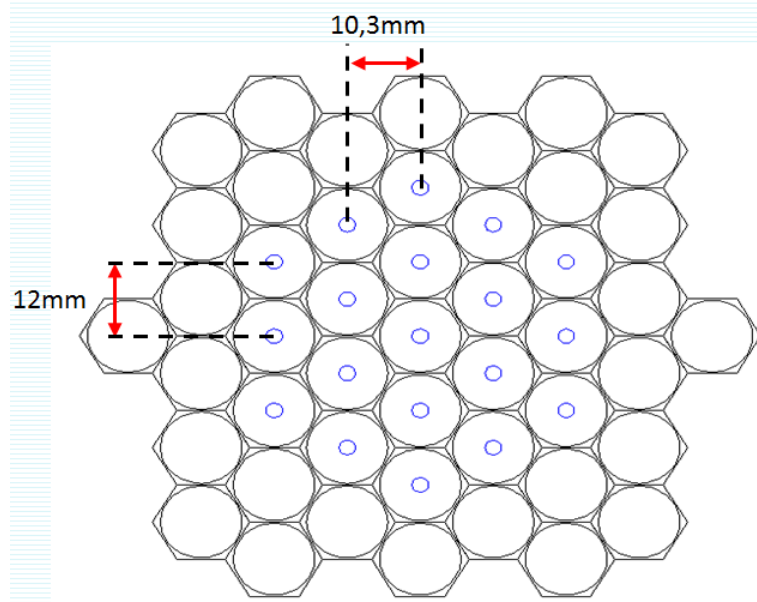


Figura 11 : Esquema antenas en plano focal

Como se puede observar en las figuras, la principal problemática es la distancia entre las antenas. Al encontrarse en un entorno reducido (1m de diámetro), las antenas están muy cerca entre ellas.

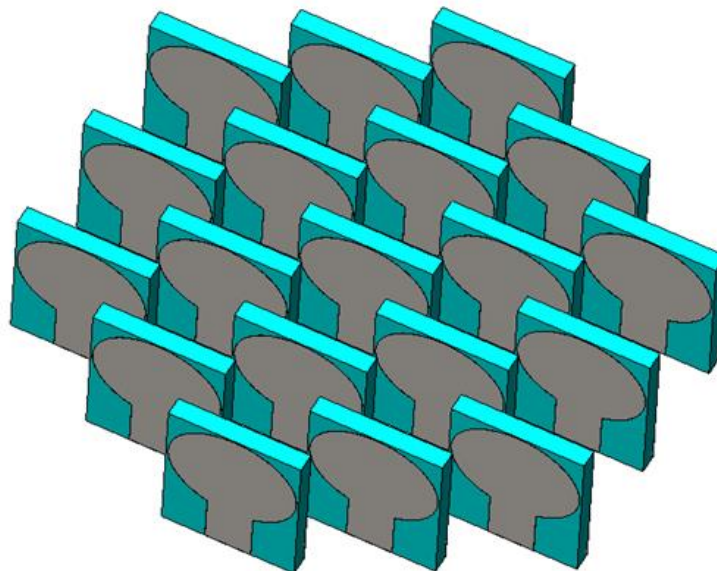


Figura 12 : Distribución de las antenas

La separación entre antenas es muy reducida, de aproximadamente 0.1λ . Si se representa el coeficiente de reflexión que tiene el elemento central simplemente por la presencia física del resto de antenas, se puede apreciar que existe una modificación en la respuesta en la reflexión del mismo comparada con su situación aislada.

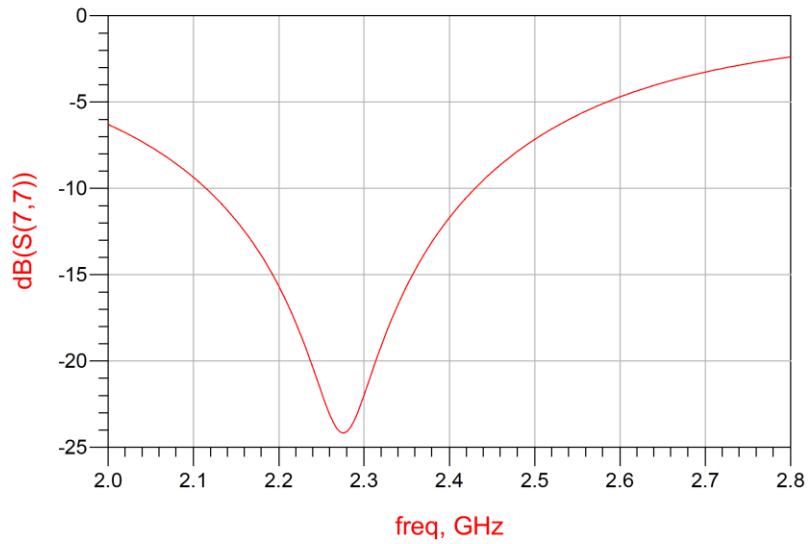


Figura 13 : Coeficiente de reflexión de elemento central en retícula de 19. Radiando solo el elemento central

Esta situación empeorará cuando los otros 18 elementos estén radiando, ya que, lo que se recoge en Figura 7 implica que únicamente elemento central radiaría, y que la presencia de los otros 18 (en condiciones de no radiación), perturban al elemento bajo estudio. Obviamente esta situación se puede agravar todavía más si se aumenta el número de elementos a considerar en la simulación.

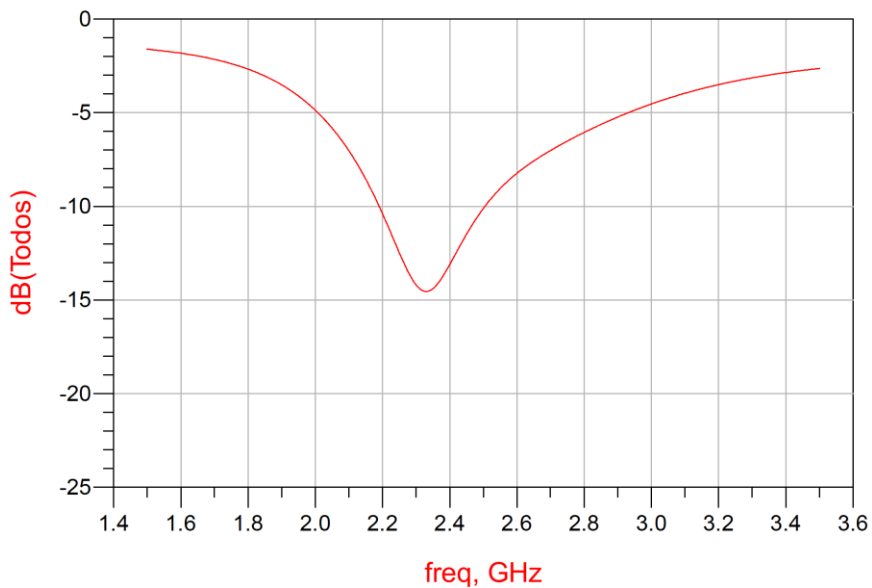


Figura 14 : Coeficiente de reflexión del elemento central en retícula de 19. Radiando todos los elementos.

2.2.2 Tecnologías Wired

Tras demostrar la inviabilidad de la arquitectura mediante tecnología Wireless se estudiarán las soluciones Wired. Para ello se estudiarán los tiempos, las ventajas e inconvenientes de las tecnologías I2C y Bus CAN, realizando una comparativa entre ambas.

2.2.2.1 Bus I2C

I2C es un bus de comunicaciones en serie. Su nombre viene de Inter-Integrated Circuit (Inter-Circuitos Integrados). La velocidad es de 100 kbit/s en el modo estándar, aunque también permite velocidades de hasta 3.4 Mbit/s [4].

I2C permite cuatro tipos de velocidades:

- ✓ 100 Kbit/s Standart
- ✓ 400 Kbit/s Fast mode
- ✓ 1 Mbit/s Fast mode Plus
- ✓ 3.4 Mbit/s High Speed Mode

La principal característica de I2C es que utiliza dos líneas para transmitir la información: una para los datos (SDA) y por otra la señal de reloj (SCL). También es necesaria una tercera línea, pero esta sólo es la referencia (GND).

Características I2C:

1. El bus está libre cuando SDA y SCL están en estado lógico alto.
2. En estado bus libre, cualquier dispositivo puede ocupar el bus I2C como maestro.
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 (A7-A1) que componen la dirección del dispositivo esclavo con el que se quiere comunicar, y el octavo bit (A0) de menor peso se corresponde con la operación deseada (L/E), lectura=1 (recibir del esclavo) y escritura=0 (enviar al esclavo).
5. La dirección enviada es comparada por cada esclavo del bus con su propia dirección, si ambas coinciden, el esclavo 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 el esclavo reconoce la solicitud y está en condiciones de comunicarse.
7. Seguidamente comienza el intercambio de información entre los dispositivos.
8. El maestro envía la dirección del registro interno del dispositivo que se desea leer o escribir.

9. El esclavo responde con otro bit de ACK.
10. Ahora el maestro puede empezar a leer o escribir bytes de datos. Todos los bytes de datos deben constar de 8 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.
11. Cada byte leído/escrito por el maestro debe ser obligatoriamente reconocido por un bit de ACK por el dispositivo maestro/esclavo.
12. Se repiten los 2 pasos anteriores hasta finalizar la comunicación entre maestro y esclavo.
13. Aun cuando el maestro siempre controla el estado de la línea del reloj, un esclavo de baja velocidad o que deba detener la transferencia de datos mientras efectúa otra función, puede forzar la línea SCL a nivel bajo. Esto hace que el maestro entre en un estado de espera, durante el cual, no transmite información esperando a que el esclavo esté listo para continuar la transferencia en el punto donde había sido detenida.
14. Cuando la comunicación finaliza, el maestro transmite una "stop condition" para dejar libre el bus.

La trama enviada en I2C contiene una condición de START, la dirección del dispositivo ADDRESS, un bit para leer o escribir en el dispositivo, una condición de STOP y un bit de ACK por cada Byte de datos.

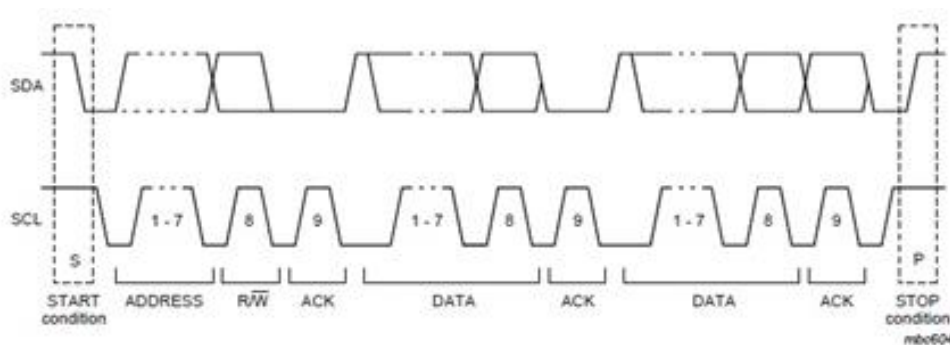


Figura 15 : Trama I2C

2.2.2.2 Bus CAN

Es una tecnología Multi – Master *broadcast serial bus* en el que todos los nodos reciben los datos, que permite velocidades de hasta 3.4 Mbps. Cada nodo es capaz de enviar y recibir mensajes, de manera semiduplex [5].

Cada nodo contiene:

- I. *Host processor*: Decide que mensajes leer y cuales enviar.
- II. *Can controller*: Para controlar como enviar y recibir mensajes. Necesita un reloj asíncrono.
- III. *Transceiver*: Adapta la señal para enviar y recibir.

Sus principales características son:

1. Prioridad de mensajes.
2. Garantía de tiempos de latencia.
3. Flexibilidad en la configuración.
4. Recepción por multidifusión (multicast) con sincronización de tiempos.
5. Sistema robusto en cuanto a consistencia de datos.
6. Sistema multimaestro.
7. Detección y señalización de errores.
8. Retransmisión automática de tramas erróneas
9. Distinción entre errores temporales y fallas permanentes de los nodos de la red, y desconexión autónoma de nodos defectuosos.

El mensaje consiste en un Identificador (11 bits), datos a enviar y bits de detección y corrección de errores.

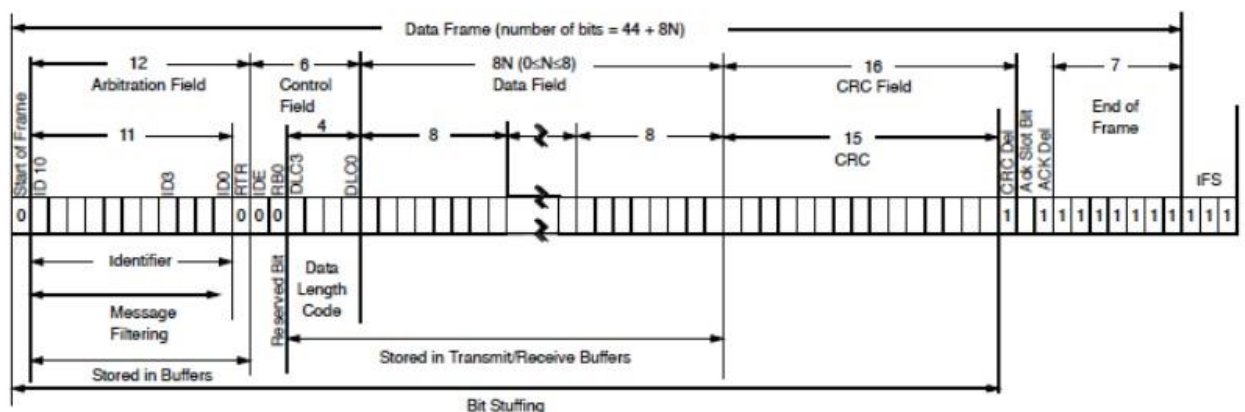


Figura 16 : Trama Bus CAN.

Para tomar una decisión sobre la tecnología a usar, se tiene que tener en cuenta tanto el tiempo de configuración, la seguridad de la trama enviada y otros factores. Ambas tecnologías ofrecen altas velocidades y sistemas de seguridad de transmisión de información.

	CAN Bus	I²C Bus
Velocidad	Hasta 1Mbps	Hasta 3.4Mbps
Acknowledgement	Si	Si
Detección de error	Si	No
Datos ilimitados por trama	No	Si
Bus diferencial	Si	No
Interrupciones del esclavo	Si	No
Retransmisión automática	Si	No
CSMA-CD	Si	No

Figura 17 : Comparativa I2C y Bus CAN

2.2.2.3 Ethernet

Es un estándar de redes de área local para computadores con acceso al medio por contienda CSMA/CD. CSMA/CD (Acceso Múltiple por Detección de Portadora con Detección de Colisiones [6].

Los elementos de una red Ethernet son: tarjeta de red, repetidores, concentradores, puentes, los conmutadores, los nodos de red y el medio de interconexión. Los nodos de red pueden clasificarse en dos grandes grupos: equipo terminal de datos (DTE) y equipo de comunicación de datos (DCE).

Los DTE son dispositivos de red que generan el destino de los datos: los PC, routers, las estaciones de trabajo, los servidores de archivos, los servidores de impresión; todos son parte del grupo de las estaciones finales. Los DCE son los dispositivos de red intermediarios que reciben y retransmiten las tramas dentro de la red pueden ser: conmutadores (switch), concentradores (hub), repetidores o interfaces de comunicación [7]. Por ejemplo: un módem o una tarjeta de interfaz.

Elementos de una red Ethernet:

1. NIC o Tarjeta de Interfaz de Red - permite que una computadora acceda a una red local. Cada tarjeta tiene una única dirección MAC que la identifica en la red. Una computadora conectada a una red se denomina nodo.
2. Repetidor o repeater - aumenta el alcance de una conexión física, recibiendo las señales y retransmitiéndolas, para evitar su degradación, a través del medio de transmisión, lográndose un alcance mayor. Usualmente se usa para unir dos áreas locales de igual tecnología y sólo tiene dos puertos. Opera en la capa física del modelo OSI.
3. Concentrador o hub - funciona como un repetidor pero permite la interconexión de múltiples nodos. Su funcionamiento es relativamente

simple pues recibe una trama de ethernet, por uno de sus puertos, y la repite por todos sus puertos restantes sin ejecutar ningún proceso sobre las mismas. Opera en la capa física del modelo OSI.

4. Puente o bridge - interconecta segmentos de red haciendo el cambio de frames (tramas) entre las redes de acuerdo con una tabla de direcciones que le dice en qué segmento está ubicada una dirección MAC dada. Se diseñan para uso entre LAN's que usan protocolos idénticos en la capa física y MAC (de acceso al medio). Aunque existen bridges más sofisticados que permiten la conversión de formatos MAC diferentes (Ethernet-Token Ring por ejemplo).
5. Conmutador o Switch - funciona como el bridge, pero permite la interconexión de múltiples segmentos de red, funciona en velocidades más rápidas y es más sofisticado. Los switches pueden tener otras funcionalidades, como Redes virtuales, y permiten su configuración a través de la propia red. Funciona básicamente en la capa 2 del modelo OSI (enlace de datos). Por esto son capaces de procesar información de las tramas; su funcionalidad más importante es en las tablas de dirección. Por ejemplo, una computadora conectada al puerto 1 del conmutador envía una trama a otra computadora conectada al puerto 2; el switch recibe la trama y la transmite a todos sus puertos, excepto aquel por donde la recibió; la computadora 2 recibirá el mensaje y eventualmente lo responderá, generando tráfico en el sentido contrario; ahora el switch conocerá las direcciones MAC de las computadoras en el puerto 1 y 2; cuando reciba otra trama con dirección de destino de alguna de ellas, sólo transmitirá la trama a dicho puerto disminuyendo así el tráfico de la red y contribuyendo al buen funcionamiento de la misma.

Dentro de la trama Ethernet se encuentran las cabeceras de IP, TCP y los datos a enviar.

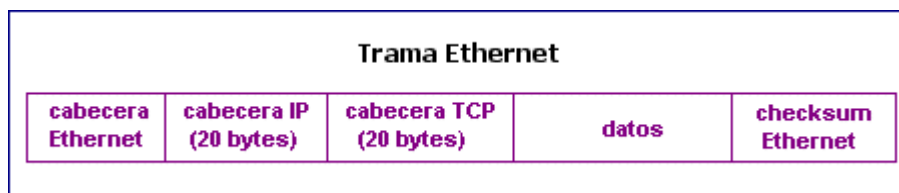


Figura 18 : Trama Ethernet

Tipos de Ethernet

Tipo de Ethernet	Ancho de banda	Tipo de cable	Duplex	Distancia máxima
10Base-5	10 Mbps	Coaxial thicknet	Half	500 m
10Base-2	10 Mbps	Coaxial thinnet	Half	185 m
100Base-TX	10 Mbps	UTP Cat3/Cat5	Half	100 m
100Base-TX	100 Mbps	UTP Cat5	Half	100 m
100Base-TX	200 Mbps	UTP Cat5	Full	100 m
100Base-TX	100 Mbps	Fibra multimodo	Half	400 m
1000Base-T	200 Mbps	Fibra multimodo	Full	2 km
1000Base-TX	1 Gbps	UTP Cat5e	Full	100 m
1000Base-SX	1 Gbps	UTP Cat6	Full	100 m
1000Base-LX	1 Gbps	Fibra multimodo	Full	550 m
10GBase-CX4	1 Gbps	Fibra monomodo	Full	2 km
10GBase-T	10 Gbps	Twinaxial	Full	100 m
10GBase-LX4	10 Gbps	UTP Cat6a/Cat7	Full	100 m
10GBase-LX4	10 Gbps	Fibra multimodo	Full	300 m
10 Mbps	10 Gbps	Fibra monomodo	Full	10 km

Figura 19 : Tipos de Ethernet

En todas las soluciones propuestas el tiempo de configuración de los dispositivos aumenta sobre todo debido al cuello de botella producido desde la unidad central. Ethernet permite enviar información a 10Gbps con seguridad en distancias medias, con lo que se podría solucionar el cuello de botella entre la unidad central y el primer nivel.

2.2.3 Tecnologías Mixtas

Como la solución con tecnologías *Wireless* es inviable debido a la alta densidad de antenas, se estudia también una solución mixta.

- CyFi + I2C: Se estudia una opción con arquitectura de tipo árbol compuesta por tecnología CyFi (*Wireless*) entre la unidad central y el primer nivel. Desde el primer nivel se configurarán los dispositivos mediante una solución cableada por I2C.

CyFi es una tecnología *Wireless* diseñada por Cypress Semiconductor capaz de transmitir a 1 Mbps mediante una radio de 2.4 GHz. Es capaz de controlar mediante una topología estrella hasta 250 nodos.

Las principales características de CyFi son:

1. Velocidad desde 260 Kbps hasta 1 Mbps mediante GFSK.
2. Almacenamiento de datos.
3. Tensión desde 1.8 V hasta 3.6 V.
4. Consumo de corriente (modo *sleep*) menor de 1 μ A.
5. Cambio rápido de canal.

Ofrece seguridad y menos consumo respecto a Zigbee [8].

	ZigBee	CyFi
Velocidad	Hata 250Kbps	Hasta 1Mbps
Canales	16	80
Ancho del canal	5 Mhz	1 Mhz
DSSS	Si	Si
Semi-Duplex	Si	Si
Consumo operativo	30 mA	21 mA
Consumo mínimo	3 μ A	1 μA

Figura 20 : Comparativa Zigbee y CyFi

3 Análisis de soluciones

En este capítulo se analizará las distintas soluciones tecnológicas bajo una arquitectura de tipo árbol. Se ha elegido una arquitectura árbol debido a que reduce el tiempo final, ya que en una arquitectura árbol es posible paralelizar.

Además se usará una arquitectura de tipo bus para comunicarse con los dispositivos de una celda ya que los dispositivos están agrupados en celdas de 19 dispositivos. Usando ambas arquitecturas se reduce el tiempo final de configuración y la complejidad de las conexiones.

3.1 Comparativa de tiempos

El objetivo es calcular el tiempo final de configuración de todos los dispositivos según las diferentes tecnologías bajo arquitectura árbol.

En primer lugar, se configurarán todos los dispositivos (5.000). Se llamará Tiempo OffLine, ya que es el tiempo en enviar a los dispositivos la información de posicionamiento, y esto se hará cuando las fibras estén estáticas observando el universo.

En segundo lugar, se configurarán los dispositivos que se hayan posicionado de manera errónea, se llamará Tiempo Live. Este tiempo es crítico, ya que cuanto más tiempo se tarde en posicionar, más tiempo se tardará en empezar a capturar las imágenes del espacio.

Para detectar si las fibras se han posicionado correctamente se utilizará una cámara situada en el foco del telescopio (llamada *Fiber View Camera*). Se aplica una fuente de luz por la parte posterior de las fibras ópticas y éstas convertirán en emisoras de luz, siendo la posición de cada una de las fibras ópticas detectada por la cámara. Tras un rápido análisis de la imagen, una vez detectada la posición real de cada una de las fibras, se procederá a configurar las fibras ópticas erróneamente posicionadas.

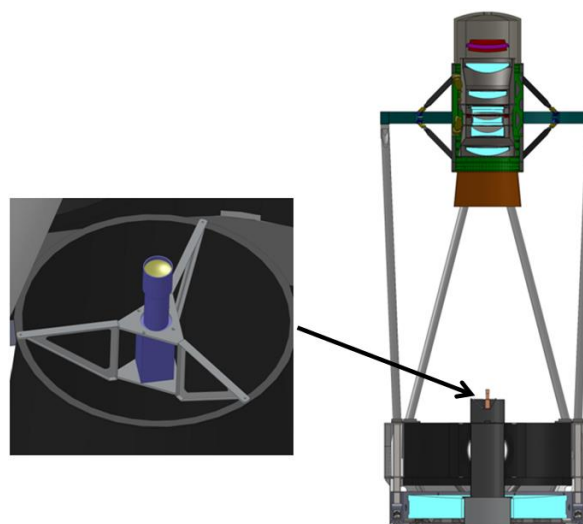


Figura 21 : Fiber View Camera

Debido al diseño de los posicionadores de fibra, es posible que se produzcan interbloques cuando los dispositivos se están posicionando. Cada Mini Robot contiene dos motores, un primer motor es capaz de girar la estructura 360 grados. Un segundo motor es capaz de mover el brazo que sujeta la fibra óptica 180 grados. Existen zonas comunes en las trayectorias de los brazos de distintos Mini Robots que sujetan las fibras ópticas, formándose zonas de posibles interbloques [16].

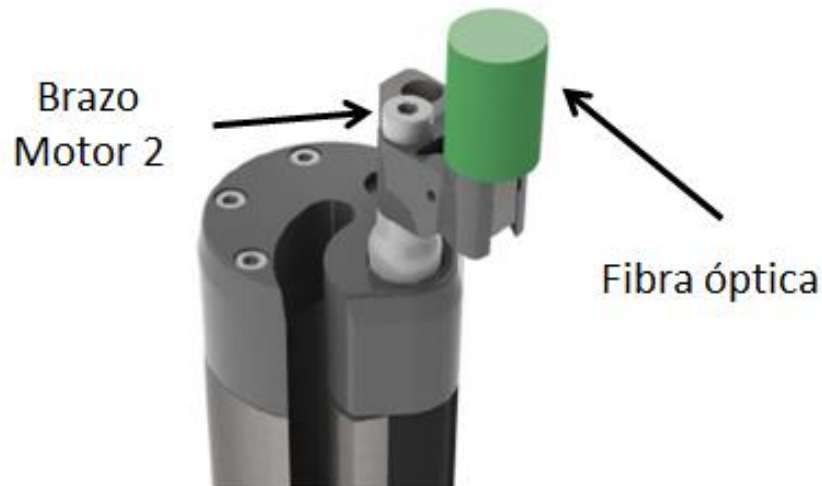


Figura 22 : Posicionador de fibra

Con el fin de solucionar estos interbloques, se utilizarán algoritmos software mediante los cuales se asegura que mediante siete etapas se resuelvan los interbloques. Cada Mini Robot recibirá siete posiciones, cada una corresponde a cada etapa del algoritmo.

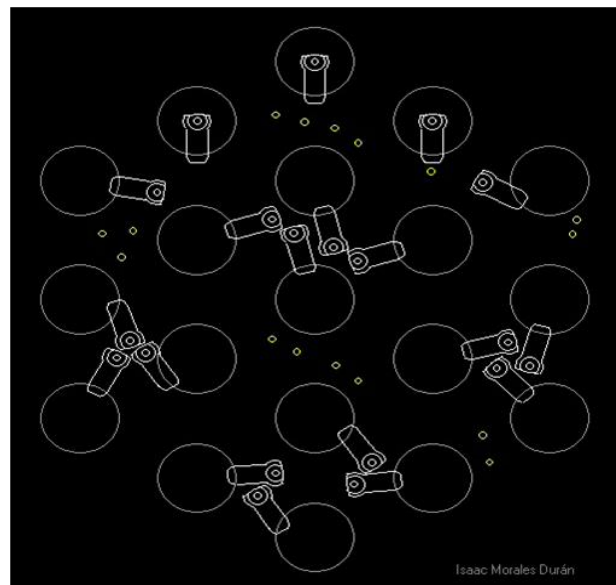


Figura 23 : Interbloques

- Offline: Se realizará la carga de posiciones para cada robot. Debido a los posibles interbloqueos, habrá que mandar siete posiciones, ya que se asegura que en máximo siete pasos se resuelven los interbloqueos.

- Live: Se realizará la carga de posiciones para cada robot cuyo posicionamiento haya sido erróneo. Para ello se realizará el cálculo en función de cuantos robots hayan sido erróneamente posicionados. Se realizará el cálculo con 100, 1.500 y 5.000 robots erróneamente posicionados.

En el caso de que el posicionamiento sea erróneo, se estudiarán dos posibilidades. Una primera sería enviarle la tabla entera (28 bytes), de posiciones, y una segunda sería enviarle al dispositivo un dato de posición para corregir el error (4 bytes).

En una posibilidad a cada dispositivo erróneamente posicionado se le enviará una tabla de posiciones.

Tiempo 1	Posición 1 Motor 1 2 bytes	Posición 1 Motor 2 2 bytes
Tiempo 2	Posición 2 Motor 1 2 bytes	Posición 2 Motor 2 2 bytes
Tiempo 3	Posición 3 Motor 1 2 bytes	Posición 3 Motor 2 2 bytes
Tiempo 4	Posición 4 Motor 1 2 bytes	Posición 4 Motor 2 2 bytes
Tiempo 5	Posición 5 Motor 1 2 bytes	Posición 5 Motor 2 2 bytes
Tiempo 6	Posición 6 Motor 1 2 bytes	Posición 6 Motor 2 2 bytes
Tiempo 7	Posición 7 Motor 1 2 bytes	Posición 7 Motor 2 2 bytes

En la otra posibilidad a cada dispositivo erróneamente posicionado se le enviarán dos posiciones.

Tiempo 1	Posición 1 Motor 1 2 bytes	Posición 1 Motor 2 2 bytes
----------	-------------------------------	-------------------------------

Para este cálculo se supondrá un tamaño de datos para cada dispositivo de 28 bytes. También se estudiará los tiempos en función de las distintas velocidades que ofrecen cada tecnología, y posibles modificaciones en la arquitectura.

3.1.1 I2C

La arquitectura que se propone con esta tecnología es de tipo árbol de tres niveles, el primer nivel con cuatro Control Device, cada uno controlando 65 Control Devices, las cuales controlarían una celda de 19 dispositivos.

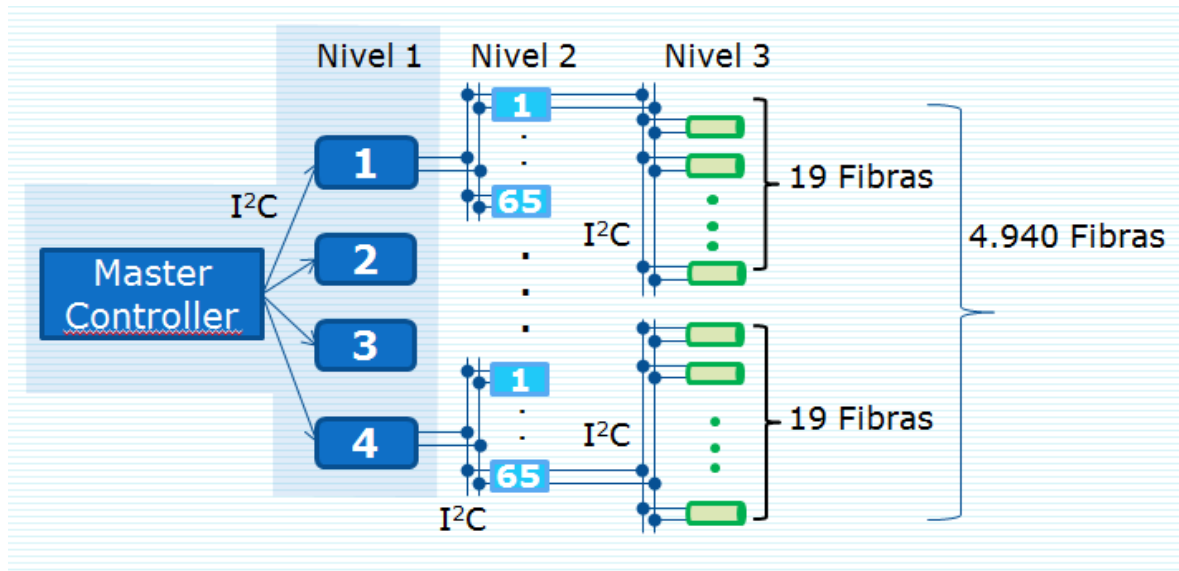


Figura 24 : Diagrama I2C

Cada nivel funcionará como un router para el siguiente nivel [9].

El principal problema de esta estructura es que se tendrían 260 placas de control con lo que se tendría muchísimo hardware, elevando así el problema del consumo.

Otro problema de esta arquitectura es que entre la unidad central y el primer nivel de la arquitectura se produce un cuello de botella, lo cual aumenta de manera considerable el tiempo de configuración final de todos los dispositivos. Esto es debido a que la unidad central se comunica con el primer nivel mediante I2C.

El tiempo aproximado de configuración para todos los dispositivos sería de 2s, utilizando una velocidad de 1Mbps.

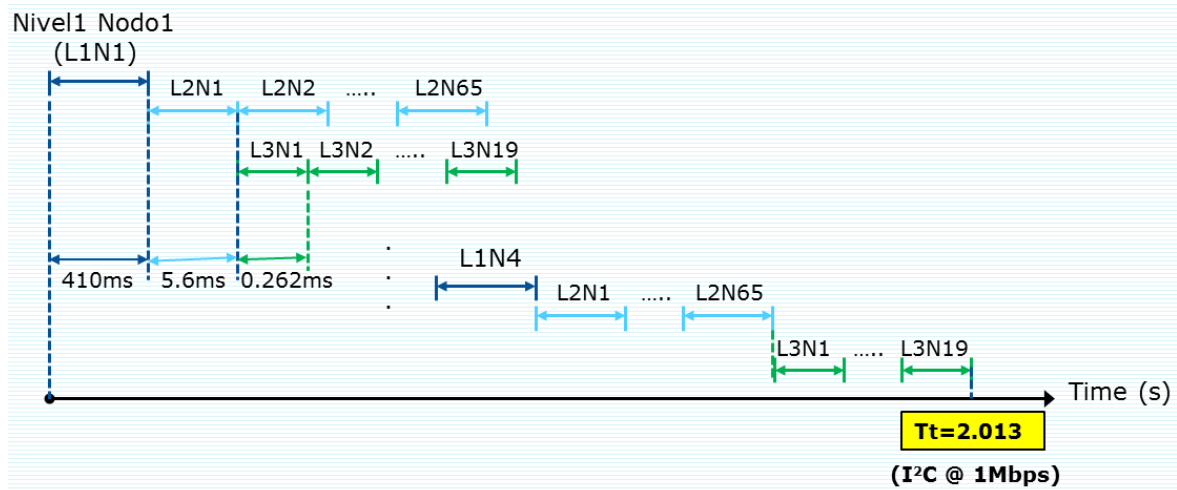


Figura 25 : Tiempos I2C

Para el análisis se tendrá en cuenta que el bloque de datos a enviar es de 28 bytes de posicionamiento.

Por lo que teniendo en cuenta la trama I2C habrá que enviar: $10\text{bits} * (\text{Adress} + \text{Start} + \text{stop}) + 8 * 28\text{bits} (\text{datos}) + 28\text{bits} (\text{ack}) = 262 \text{ bits}$.

3.1.1.1 Tiempo OffLine:

3.1.1.1.1 Opción 4 Control Device:

Se implementarán 4 Control Device que se conectarán a 65 micros cada uno. Cada uno de estos micros tendrá dos I2C, uno para recibir (esclavo) de la capa superior y otro para enviar (Master) a la capa inferior. La capa inferior tendrá 19 micros que son los que se comunicaran con los drivers de motor. Con este sistema se da soporte para $4 * 65 * 19 = 4940$ robots posicionadores. El primer nivel tiene que enviar todos los datos por el bus.

Las transmisiones se harán en paralelo, por lo que cuando el Control Device 1 ha recibido sus datos, los empezara a mandar. Entonces cuando el Control Device 4 está recibiendo los datos, Control Device 1, Control Device 2 y Control Device 3 ya están enviando. En el segundo nivel pasa lo mismo.

Un nodo del segundo nivel tendrá que enviar $262 \text{ bits} * 19 = 4978 \text{ bits}$.

$$T_3 = 4.978 \text{ bits} / 1.000.000 \text{ bits} = 0.004978 \text{ s.}$$

Estos 4978 bits (622.25 bytes) se empaquetan para que los envíe el primer nivel. $10\text{bits} + 8 * 623 + 623 = 5617 \text{ bits}$.

El primer nivel enviará $5617 \text{ bits} * 65 = 365.105 \text{ bits}$.

$$T_2 = 365.105 \text{ bits} / 1.000.000 \text{ bits} = 0.365105 \text{ s.}$$

Estos 365.105 (45.638,1) se empaquetan para que los envíe el primer nivel. $10\text{bits} + 8*639+639= 410.761 \text{ bits}$.

El primer nivel recibirá $410.761 \text{ bits} * 4 = 1.643.044 \text{ bits}$.

$$T_1 = 1.643.044 \text{ bits} / 1.000.000 \text{ bits} = 1,643044 \text{ s.}$$

$$T_t = T_1 + T_2 + T_3 = 2,013127 \text{ s.}$$

3.1.1.1.2 Opción 120 Control Device:

Se implementarán 120 micros operando como Control Device que se conectarán a 41 micros cada uno. Con este sistema se da soporte para $120*41 = 4920$ robots posicionadores.

Un nodo del primer nivel tendrá que enviar $262 \text{ bits} * 41 = 10.742 \text{ bits}$.

$$T_2 = 10.742 \text{ bits} / 1.000.000 \text{ bits} = 0.010742 \text{ s.}$$

Estos 10.742 bits (1342,75) se empaquetan para que los envíe el primer nivel.

$$10\text{bits} + 8*1343+1343= 12.097 \text{ bits.}$$

El primer nivel recibirá $12.097 \text{ bits} * 120 = 1.451.640 \text{ bits}$.

$$T_1 = 1.451.640 \text{ bits} / 1.000.000 \text{ bits} = 1,45164.$$

$$T_t = T_1 + T_2 = 0.010742 \text{ s} + 1,45164 = 1,462382 \text{ s.}$$

3.1.1.2 Tiempo Live. Tabla Entera:

3.1.1.2.1 Opción 4 Control Device:

- ✓ 114 Mini- Robots posicionados erróneamente:

Tiempo máximo será cuando se puedan agrupar en el menor número de Control Device posibles. $114/19 = 6$ nodos del segundo nivel. Un nodo del segundo nivel tendrá que enviar $262 \text{ bits} * 19 = 4978 \text{ bits}$.

$$T3 = 4.978 \text{ bits} / 1.000.000 \text{ bits} = 0.004978 \text{ s.}$$

Estos 4978 bits (622.25 bytes) se empaquetan para que los envíe el primer nivel. $10 \text{ bits} + 8 * 623 + 623 = 5617 \text{ bits}$. El primer nivel enviará $5617 \text{ bits} * 6 = 28.085 \text{ bits}$

$$T2 = 28.085 \text{ bits} / 1.000.000 \text{ bits} = 0.028085 \text{ s.}$$

Estos 28.085 bits (3510,6) se empaquetan para que los envíe el primer nivel. $10 \text{ bits} + 8 * 3511 + 3511 = 31.609 \text{ bits}$. El primer nivel recibirá 31.609 bits.

$$T1 = 31.609 / 1.000.000 \text{ bits} = 0,031069 \text{ s.}$$

$$Tt = T1 + T2 + T3 = 0,064672 \text{ s.}$$

- ✓ 2470 Mini- Robots posicionados erróneamente:

Tiempo máximo será cuando se puedan agrupar en el menor número de Control Device posibles. $2470/19 = 130$ Control Devices. $130/65 = 2$ nodos del segundo nivel. El nodo del segundo nivel tendrá que enviar $262 \text{ bits} * 19 = 4978 \text{ bits}$.

$$T3 = 4.978 \text{ bits} / 1.000.000 \text{ bits} = 0.004978 \text{ s.}$$

Estos 4978 bits (622.25 bytes) se empaquetan para que los envíe el primer nivel. $10 \text{ bits} + 8 * 623 + 623 = 5617 \text{ bits}$. El primer nivel enviará $5617 \text{ bits} * 65 = 365.105 \text{ bits}$.

$$T2 = 365.105 \text{ bits} / 1.000.000 \text{ bits} = 0.365105 \text{ s.}$$

Estos 365.105 (45.638,1) se empaquetan para que los envíe el primer nivel. $10 \text{ bits} + 8 * 639 + 639 = 410.761 \text{ bits}$. El primer nivel recibirá $410.761 \text{ bits} * 2 = 821.522 \text{ bits}$.

$$T1 = 821.522 \text{ bits} / 1.000.000 \text{ bits} = 0,821522 \text{ s.}$$

$$Tt = T1 + T2 + T3 = 1,191605 \text{ s.}$$

- ✓ 4940 Mini- Robots posicionados erróneamente:

Este caso corresponde a que todos los Mini – Robots se han posicionado erróneamente por lo que el tiempo es igual al tiempo Offline.

3.1.1.2.2 Opción 120 Control Devices:

- ✓ 123 Mini- Robots posicionados erróneamente:

Tiempo máximo será cuando se puedan agrupar en el menor número de Control Devices posibles. $123/41 = 3$ Control Devices. El nodo del primer nivel tendrá que enviar $262 \text{ bits} * 41 = 10.742 \text{ bits}$.

$$T2 = 10.742 \text{ bits} / 1.000.000 \text{ bits} = 0.010742 \text{ s}$$

Estos 10.742 bits (1342,75) se empaquetan para que los envíe el primer nivel.

$$10 \text{ bits} + 8 * 1343 + 1343 = 12.097 \text{ bits.}$$

El primer nivel recibirá $12.097 \text{ bits} * 3 = 36.291 \text{ bits}$.

$$T1 = 36.291 / 1.000.000 \text{ bits} = 0,036291$$

$$Tt = T1 + T2 = 0.010742 \text{ s} + 0,036291 \text{ s} = 0,047033 \text{ s.}$$

- ✓ 2501 Mini- Robots posicionados erróneamente:

Tiempo máximo será cuando se puedan agrupar en el menor número de Control Device posibles. $2501/41 = 61$ nodos del primer nivel.

El nodo del primer nivel tendrá que enviar $262 \text{ bits} * 41 = 10.742 \text{ bits}$.

$$T2 = 10.742 \text{ bits} / 1.000.000 \text{ bits} = 0.010742 \text{ s}$$

Estos 10.742 bits (1342,75) se empaquetan para que los envíe el primer nivel.

$$10 \text{ bits} + 8 * 1343 + 1343 = 12.097 \text{ bits.}$$

El primer nivel recibirá $12.097 \text{ bits} * 61 = 737.917 \text{ bits}$.

$$T1 = 737.917 / 1.000.000 \text{ bits} = 0,737917 \text{ s.}$$

$$Tt = T1 + T2 = 0.010742 \text{ s} + 0,737917 \text{ s} = 0,748659 \text{ s}$$

- ✓ 4920 Mini- Robots posicionados erróneamente

Este caso corresponde a que todos los Mini – Robots se han posicionado erróneamente por lo que el tiempo es igual al tiempo Offline.

3.1.1.3 Tiempo Live. Posición:

El bloque de datos a enviar es de 4 bytes de posicionamiento.

Por lo que habrá que enviar $10 (\text{Adress} + \text{Start} + \text{stop}) + 8 * 4 (\text{datos}) + 4 (\text{ack}) = 46 \text{ bits}$.

3.1.1.3.1 Opción 4 Control Device:

- ✓ 114 Mini- Robots posicionados erróneamente:

El tiempo de configuración es de 0,013636 s.

- ✓ 2470 Mini- Robots posicionados erróneamente:

El tiempo de configuración es de 0,212144 s.

- ✓ 4940 Mini- Robots posicionados erróneamente:

El tiempo de configuración es de 0,358414 s.

3.1.1.3.2 Opción 120 Control Device:

- ✓ 123 Mini- Robots posicionados erróneamente:

El tiempo de configuración es de 0,008288 s.

- ✓ 2501 Mini- Robots posicionados erróneamente:

El tiempo de configuración es de 0,13206 s.

- ✓ 4920 Mini- Robots posicionados erróneamente:

El tiempo de configuración es de 0,257966 s.

3.1.2 BUS CAN

Bajo una arquitectura de tipo árbol se podrían configurar todos los dispositivos en un tiempo aproximado de 3.3 s, con una velocidad de 1Mbps [10].

La arquitectura tendría dos niveles, el primer nivel con 260 Control Device, las cuales controlarían una celda de dispositivos.

Al igual que en el planteamiento con I2C el problema es se tendrían 260 placas de control, elevando el consumo.

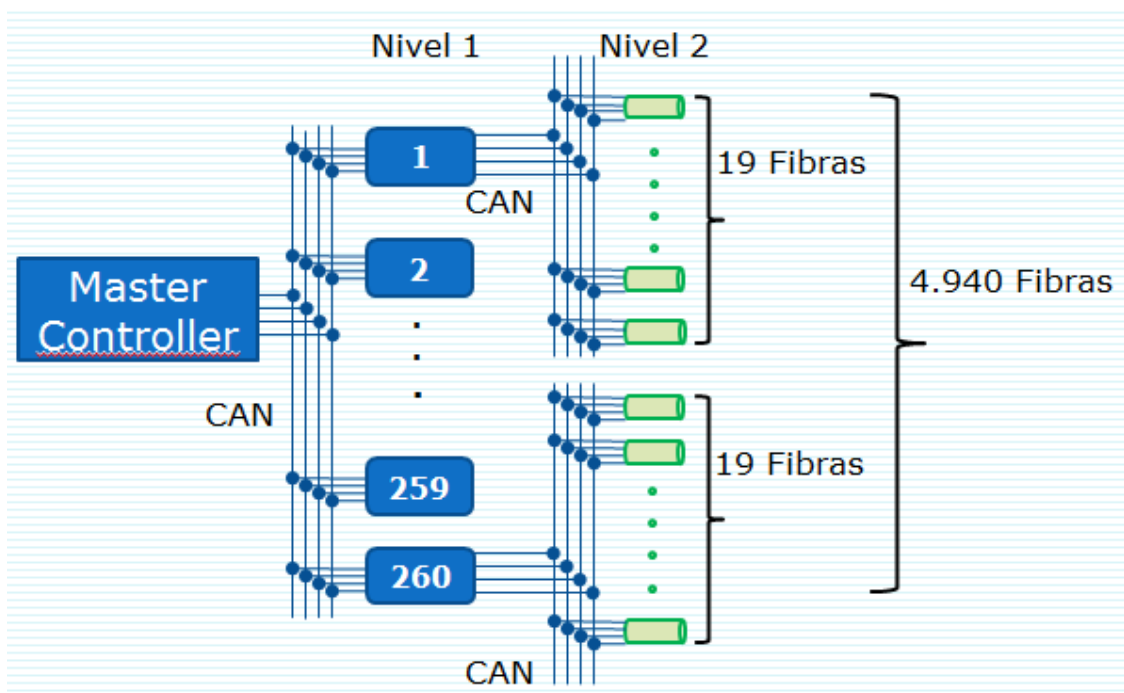


Figura 26 : Diagrama Bus CAN

Para el cálculo de tiempo, se tendrá en cuenta la trama del Bus Can, donde por cada 8 bytes de Datos habrá que enviar 44bits + 8bytes = 108 bits.

El tiempo aproximado de configuración para todos los dispositivos sería de 3.34s.

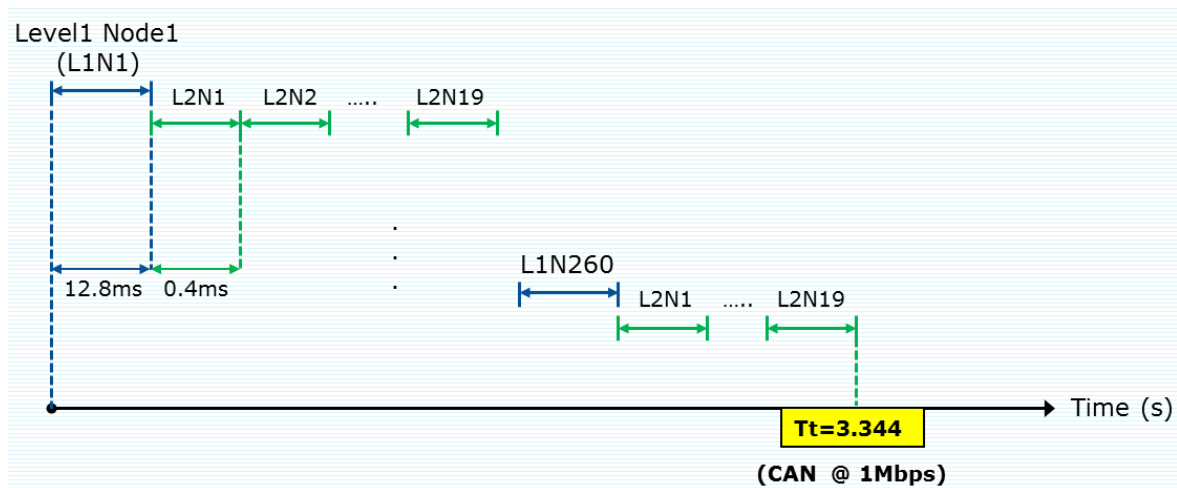


Figura 27 : Tiempos Bus CAN

3.1.2.1 Tiempo OffLine

3.1.2.1.1 Opción 260 Control Devices:

Cada Control Device enviará la información a 19 mini - robots, por lo que se podrá controlar 4940 Mini – Robots. Para enviar los 28 bytes hay que dividirlo en cada uno de los frames. Por lo que se tendrán 3 frames de 8bytes de datos y 1 frame de 4 bytes de datos.

$$(44 \text{ bits} + (8 * 8 \text{ bits})) * 3 + (44 \text{ bits} + 4 * 8 \text{ bits}) = 400 \text{ bits} = 50 \text{ bytes.}$$

Cada nodo del primer nivel enviará $19 * 400 \text{ bits} = 7600 \text{ bits}$.

$$T_2 = 7600 \text{ bits} / 1.000.000 \text{ bits} = 0,0076 \text{ s.}$$

Para enviar los 7.600 bits (950 bytes) hay que dividirlo en cada uno de los frames, ya que se tendría 118 frames de 8bytes de datos y 1 frame de 6 bytes de datos.

$$(44 \text{ bits} + (8 * 8 \text{ bits})) * 118 + (44 \text{ bits} + 6 * 8 \text{ bits}) = 12.836 \text{ bits.}$$

Cada nodo del primer nivel recibiría 12.836 bits, por lo que el bus del primer nivel recibiría $12.836 \text{ bits} * 260 = 3.337.360 \text{ bits}$.

$$T_1 = 3.337.360 \text{ bits} / 1.000.000 \text{ bits} = 3,33736 \text{ s.}$$

$$T_t = T_1 + T_2 = 3,33736 \text{ s} + 0,0076 \text{ s} = 3,34496 \text{ s.}$$

3.1.2.1.2 Opción 8 Control Devices:

Cada nodo enviará la información a 620 mini - robots, por lo que se podrá controlar 4960 Mini – Robots. Para enviar los 28 bytes hay que dividirlo en cada uno de los frames. Por lo que se tendrá 3 frames de 8bytes de datos y 1 frame de 4 bytes de datos.

$$(44 \text{ bits} + (8 * 8 \text{ bits})) * 3 + (44 \text{ bits} + 4 * 8 \text{ bits}) = 400 \text{ bits} = 50 \text{ bytes.}$$

Cada nodo del primer nivel enviará $620 * 400 \text{ bits} = 248.000 \text{ bits}$.

$$T2 = 248.000 \text{ bits} / 1.000.000 \text{ bits} = 0,248 \text{ s.}$$

Para enviar los 248.600 bits (31.000 bytes) hay que dividirlo en cada uno de los frames.

Se tendrán 3.875 frames de 8bytes de datos.

$$(44 \text{ bits} + (8 * 8 \text{ bits})) * 3.875 = 418.500 \text{ bits.}$$

Cada nodo del primer nivel recibiría 418.500 bits, por lo que el bus del primer nivel recibiría $418.500 \text{ bits} * 8 = 3.348.000 \text{ bits}$.

$$T1 = 3.348.000 \text{ bits} / 1.000.000 \text{ bits} = 3,348 \text{ s.}$$

$$Tt = T1 + T2 = 3,348 \text{ s} + 0,248 \text{ s} = 3,596 \text{ s.}$$

3.1.2.2 *Tiempo Live. Tabla Entera*

3.1.2.2.1 *Opción 260 Control Devices:*

- ✓ 114 Mini- Robots posicionados erróneamente:

Tiempo máximo será cuando se puedan agrupar en el menor número de nodos posibles. $114/19 = 6$ Control Devices. Cada nodo del primer nivel enviará $19 * 400 \text{ bits} = 7600 \text{ bits}$.

$$T_2 = 7.600 \text{ bits} / 1.000.000 \text{ bits} = 0,0076 \text{ s.}$$

Para enviar los 7.600 bites (950 bytes) hay que dividirlo en cada uno de los frames.

Se tendrán 118 frames de 8 bytes de datos y 1 frame de 6 bytes de datos.

$$(44 \text{ bits} + (8 * 8 \text{ bits})) * 118 + (44 \text{ bits} + 6 * 8 \text{ bits}) = 12.836 \text{ bits.}$$

Cada nodo del primer nivel recibiría 12.836 bits, por lo que el bus del primer nivel recibiría $12.836 \text{ bits} * 6 = 77.016 \text{ bits}$.

$$T_1 = 77.016 \text{ bits} / 1.000.000 \text{ bits} = 0,077016 \text{ s.}$$

$$T_{t \text{ max}} = T_1 + T_2 = 0,077016 \text{ s} + 0,0076 \text{ s} = 0,084616 \text{ s.}$$

- ✓ 2,508 Mini- Robots posicionados erróneamente:

Tiempo máximo será cuando se puedan agrupar en el menor número de nodos posibles.

$2508/19 = 132$ Control Devices. Cada nodo del primer nivel enviará $19 * 400 \text{ bits} = 7600 \text{ bits}$.

$$T_2 = 7.600 \text{ bits} / 1.000.000 \text{ bits} = 0,0076 \text{ s.}$$

Para enviar los 7.600 bites (950 bytes) hay que dividirlo en cada uno de los frames.

Se tendrán 118 frames de 8 bytes de datos y 1 frame de 6 bytes de datos.

$$(44 \text{ bits} + (8 * 8 \text{ bits})) * 118 + (44 \text{ bits} + 6 * 8 \text{ bits}) = 12.836 \text{ bits.}$$

Cada nodo del primer nivel recibiría 12.836 bits, por lo que el bus del primer nivel recibiría $12.836 \text{ bits} * 132 = 1.694.352 \text{ bits}$.

$$T_1 = 1.694.352 \text{ bits} / 1.000.000 \text{ bits} = 1,694352 \text{ s.}$$

$$T_{t \text{ max}} = T_1 + T_2 = 1,694352 \text{ s} + 0,0076 \text{ s} = 1,701952 \text{ s.}$$

- ✓ 4940 Mini- Robots posicionados erróneamente

Este caso corresponde a que todos los Mini – Robots se han posicionado erróneamente por lo que el tiempo es igual al tiempo Offline.

3.1.2.2.2 Opción 8 Control Devices

- ✓ 100 Mini- Robots posicionados erróneamente:

Tiempo máximo será cuando se puedan agrupar en el menor número de nodos posibles.

Los 100 nodos se agruparan en el mismo Control Devices. Cada nodo del primer nivel enviará $100 \cdot 400 \text{ bits} = 40.000 \text{ bits}$.

$$T_2 = 40.000 \text{ bits} / 1.000.000 \text{ bits} = 0,04 \text{ s.}$$

Para enviar los 40.000 bites (5.000 bytes) hay que dividirlo en cada uno de los frames.

Se tendrán 625 frames de 8bytes de datos. $(44 \text{ bits} + (8 \cdot 8 \text{ bits})) \cdot 625 = 67.500 \text{ bits}$.

$$T_1 = 67.500 \text{ bits} / 1.000.000 \text{ bits} = 0,0675 \text{ s.}$$

$$T_t \text{ max} = T_1 + T_2 = 0,0675 \text{ s} + 0,04 \text{ s} = 0,1075 \text{ s.}$$

- ✓ 2,480 Mini- Robots posicionados erróneamente:

Tiempo máximo será cuando se puedan agrupar en el menor número de nodos posibles.
 $2.480 / 620 = 4 \text{ nodos}$.

Cada nodo del primer nivel enviará $620 \cdot 400 \text{ bits} = 248.000 \text{ bits}$.

$$T_2 = 248.000 \text{ bits} / 1.000.000 \text{ bits} = 0,248 \text{ s.}$$

Para enviar los 248.600 bits (31.000 bytes) hay que dividirlo en cada uno de los frames.
Se tendrán 3.875 frames de 8bytes de datos. $(44 \text{ bits} + (8 \cdot 8 \text{ bits})) \cdot 3.875 = 418.500 \text{ bits}$.

Cada nodo del primer nivel recibiría 418.500 bits, por lo que el bus del primer nivel recibiría $418.500 \text{ bits} \cdot 4 = 1.674.000 \text{ bits}$.

$$T_1 = 1.674.000 \text{ bits} / 1.000.000 \text{ bits} = 1,674 \text{ s.}$$

$$T_t \text{ max} = T_1 + T_2 = 1,674 \text{ s} + 0,248 \text{ s} = 1,922 \text{ s.}$$

- ✓ 4940 Mini- Robots posicionados erróneamente:

Este caso corresponde a que todos los Mini – Robots se han posicionado erróneamente por lo que el tiempo es igual al tiempo Offline.

3.1.2.3 Tiempo Live. Posición.

El tamaño de paquete para cada Mini – Robot será de 76 bits.

3.1.2.3.1 Opción 260 Control Devices:

- ✓ 114 Mini- Robots posicionados erróneamente:

El tiempo de configuración es de 0,011284 s.

- ✓ 2,508 Mini- Robots posicionados erróneamente:

El tiempo de configuración es de 0,326164 s.

- ✓ 4940 Mini- Robots posicionados erróneamente

El tiempo de configuración es de 0,641044 s.

3.1.2.3.2 Opción 8 Control Devices:

- ✓ 100 Mini- Robots posicionados erróneamente:

El tiempo de configuración es de 0,020436 s.

- ✓ 2,480 Mini- Robots posicionados erróneamente:

El tiempo de configuración es de 0,379732 s.

- ✓ 4940 Mini- Robots posicionados erróneamente:

El tiempo de configuración es de 0,710484 s.

3.1.3 CyFi + I2C.

La unidad central se conectaría mediante CyFi a 250 Control Devices, las cuales configurarían los dispositivos mediante I2C. Sigue apareciendo el mismo problema de consumo debido a la gran densidad de hardware.

Con esta solución se configurará todos los dispositivos en un tiempo aproximado de 1.1 s, trabajando ambas tecnologías a 1Mbps.

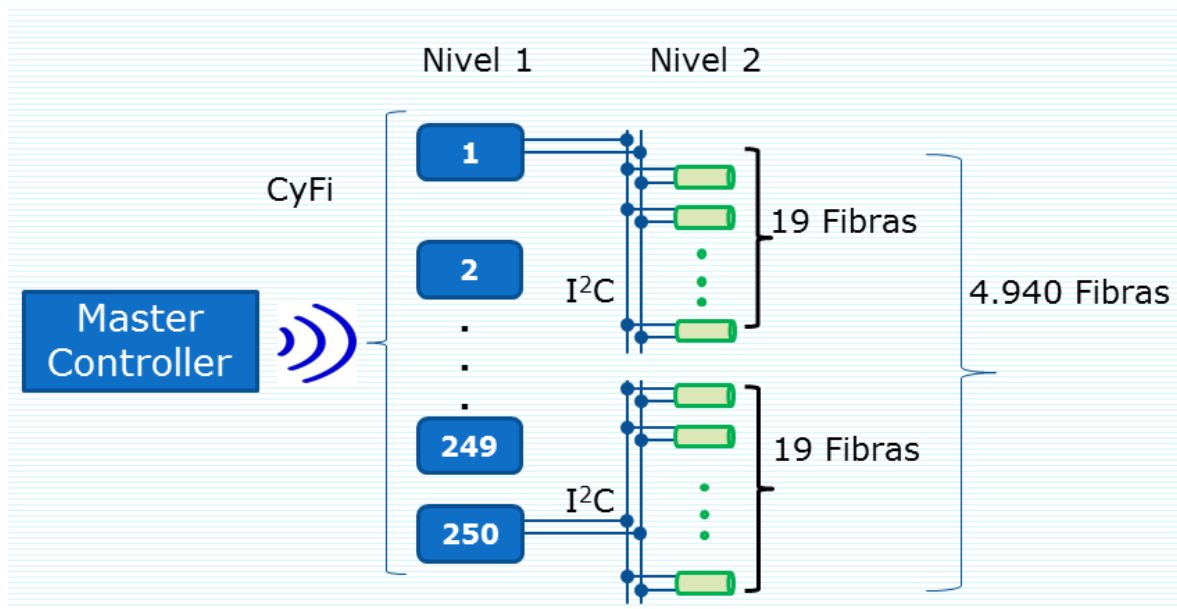


Figura 28: Diagrama CyFi+I2C.

El tiempo aproximado de configuración para todos los dispositivos sería de 1.1s.

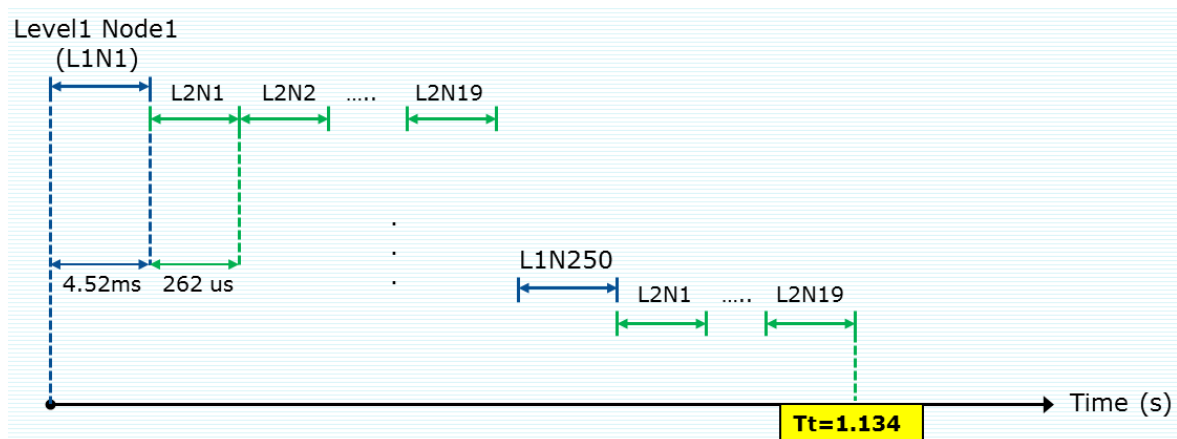


Figura 29 : Tiempo CyFi+I2C

Se supondrá que el bloque de datos a enviar es de 28 bytes por nodo.

- Etapa CyFi:

Se define:

1. T_s : Tiempo de establecimiento.
2. T_t : Tiempo de transmisión.
3. T : Tiempo total= T_s+T_t .

Cada CyFi tendrá que recibir $28 \text{ Bytes} * 19 = 532 \text{ Bytes}$.

$$T_s = 250 * 270 \mu\text{s} = 0.0675 \text{ s.}$$

$$T_t = 250 * (532 \text{ bytes} / 1 \text{ Mbps}) = 250 * (4256 \text{ bits} / 1000 \text{ kbps}) = 1.064 \text{ s}$$

$$T = T_s + T_t = 0.0675 \text{ s} + 1.064 \text{ s} = 1.1315 \text{ s}$$

Tiempo etapa CyFi = 1.1315 s.

- Etapa I2C:

El bloque de datos a enviar es de 28 bytes de posicionamiento.

Por lo que habrá que enviar $10 (\text{Adress} + \text{Start} + \text{stop}) + 8 * 28 (\text{datos}) + 28 (\text{ack}) = 262 \text{ bits}$.

$$T_t = 19 * (262 \text{ bits} / 1.000.000 \text{ bits}) = 0.004978 \text{ s.}$$

Tiempo etapa I2C 0.004978 s.

$$T_t = T_{\text{CyFi}} + T_{\text{I2C}} = 1.1315 \text{ s} + 0.004978 \text{ s} = 1,136478 \text{ s.}$$

3.1.4 Zigbee

Como se puede observar en la siguiente figura, la arquitectura propuesta sería en árbol, actuando cada Control Device como un router capaz de llegar a 250 dispositivos [11].

Con esta configuración se tendrían 20 nodos intermedios en el primer nivel, paralelizando el sistema de comunicación, con lo que se reduciría el tiempo final de configuración de todos los dispositivos. El tiempo aproximado de configuración sería de unos ≈ 4.7 s.

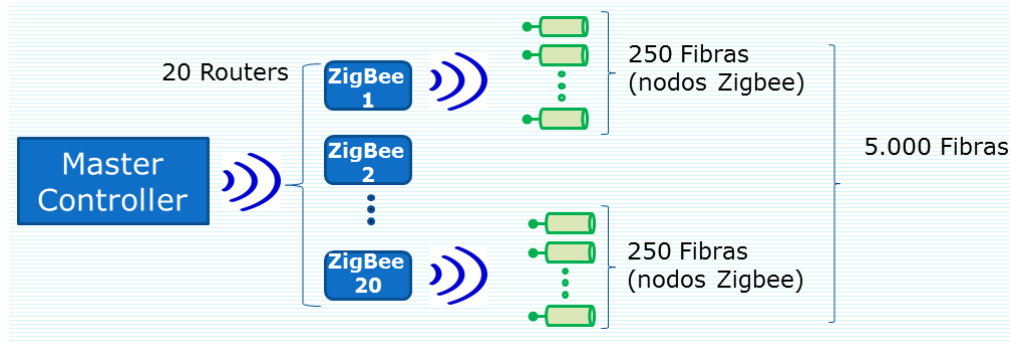


Figura 30 : Diagrama Zigbee

El tiempo aproximado de configuración de todos los dispositivos es de 4.7s.

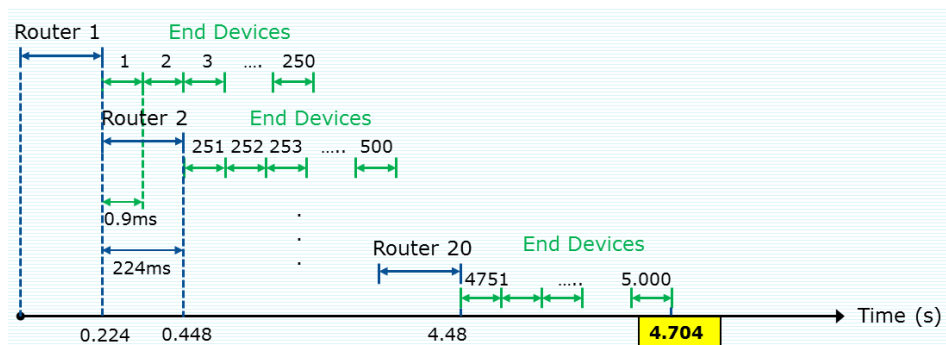


Figura 31 : Tiempos Zigbee

Se tendrá un primer nivel de 20 nodos y otro segundo nivel de 250 nodos. Total 5.000 fibras. Cada nodo del primer nivel tendrá que enviar $28 \text{ bytes} * 250 = 7.000 \text{ bytes}$.

El primer nivel tardará: $T_1 = 20 * (7.000 \text{ Bytes} / 250 \text{ Kbps}) = 4.48 \text{ s}$.

Cada nodo del segundo nivel tendrá que enviar 28 bytes.

$$T_2 = 250 * 28 \text{ Bytes} / 250 \text{ Kbps} = 0.224 \text{ s}$$

$$T_t = T_1 + T_2 = 4.48 \text{ s} + 0.224 \text{ s} = 4.704 \text{ s}$$

3.1.5 Ethernet + I2C:

La principal ventaja de esta solución es que al introducir Ethernet en el diseño se soluciona el cuello de botella producido entre la unidad central y los Control Device, ya que usando Ethernet se puede utilizar velocidades de hasta 10 Gbps. Los multiplexores utilizados son capaces de soportar velocidades de hasta 400 Kbps por el bus I2C.

Para el análisis de tiempo se distingue:

1. T1: Tiempo entre la unidad central y el Control Device.
2. T2: Tiempo de configuración del primer multiplexor.
3. T3: Tiempo de configuración del segundo multiplexor.
4. T4: Tiempo de configuración de una celda de dispositivos.
5. Nº Bytes a cada dispositivo: 20 Bytes.
6. Velocidad Ethernet: 1.000 Mbps.
7. Velocidad I2C: 400 Kbps.

Debido a que la velocidad de Ethernet es mucho mayor que la velocidad de I2C, el tiempo T1 es muy inferior al tiempo total, por lo que se asumirá T1 despreciable.

Además al dividir la estructura con seis Control Device, se paralelizará la configuración de dispositivos, por lo que el tiempo de configuración de todos los dispositivos será aproximadamente el tiempo de configuración de los dispositivos configurados por un Control Device (836 dispositivos).

Cada canal I2C representa una celda de 19 dispositivos, por lo que para configurar los primeros 19 dispositivos habría que configurar dos multiplexores. Configurar dos multiplexores supone dos tramas de 20 bits. Para configurar la siguiente celda de dispositivos habría que configurar solo el segundo multiplexor, lo que supone 10 bits de configuración.

Para configurar todos los dispositivos controlados por un Control Device (836 dispositivos) se configurará 6 veces el primer multiplexor (60 bits), y 44 veces los multiplexores del segundo nivel (440 bits). En total se enviarán 500 bits para la configuración de los multiplexores. Estos 500 bits a una velocidad de transmisión de 400 Kbps suponen una penalización al tiempo total de configuración de 1.25 ms debido a la utilización de multiplexores, suponiendo un 0,32% del tiempo total.

Es una penalización asumible ya que al incorporar los multiplexores se reduce la complejidad del hardware y el consumo de potencia.

Para configurar cada dispositivo se necesitarán 20 bytes, que empaquetados en una trama I2C son 190 bits para cada dispositivo. Como se quiere configurar 836 dispositivos, se tienen $836 \times 190 \text{ bits} = 158.840 \text{ bits}$.

Por lo que el tiempo total de configurar los multiplexores y configurar los dispositivos será aproximadamente de 400 ms. Este tiempo sería el tiempo Offline de configuración.

Para los tiempos Live de configuración de los dispositivos erróneamente posicionados, el tiempo será siempre inferior al tiempo OffLine, se envíe una tabla de posiciones o una posición.

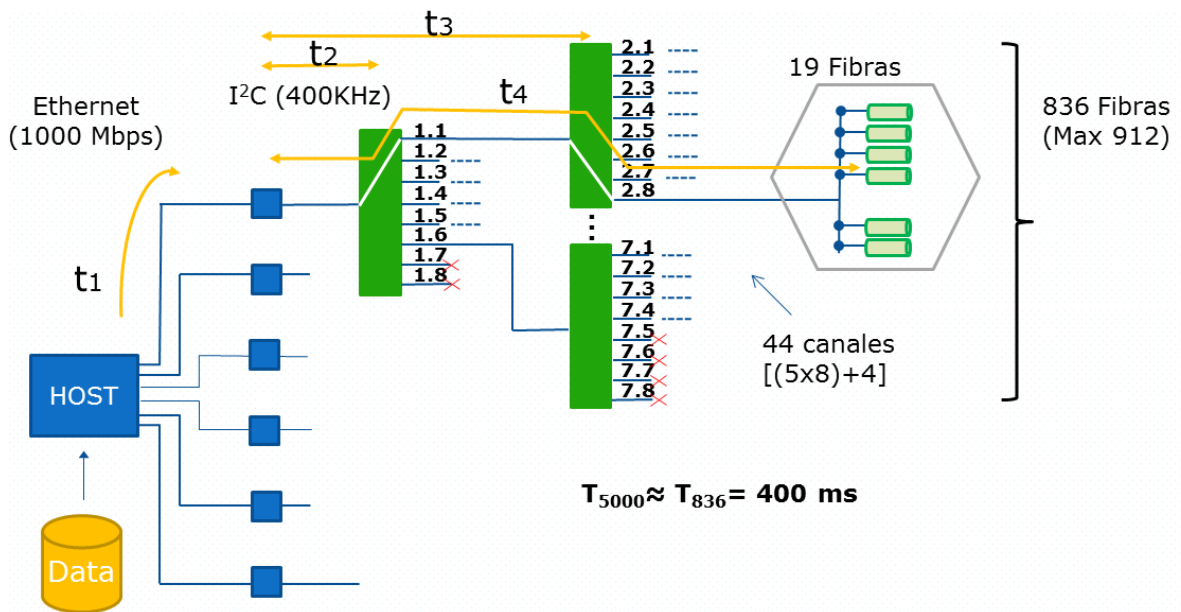


Figura 32 : Diagrama y tiempos Ethernet + I2C

3.2 Resumen

En la siguiente figura se observa los tiempos de configuración OffLine dependiendo de la tecnología. Se puede observar que utilizar Zigbee reporta tiempos más altos que el resto de soluciones. En cuanto a tiempos, una de las mejores soluciones es la mixta usando tecnologías wireless y wired. El problema de esta solución es que aunque se haya reducido la densidad de antenas, se sigue teniendo 250 antenas en un entorno reducido.

La solución con menos delay es utilizando Ethernet junto I2C.

Technology	Timing
Zigbee	4.704 s
Bus CAN @ 1 Mbps	3.345 s
I ² C @ 1 Mbps	2.013 s
CyFi + I ² C	1.137 s
Ethernet + I ² C	0.399 s

Figura 33 : Comparativa de tiempos

El resto de las soluciones cableadas ofrecen unos tiempos intermedios. Estas opciones ofrecen más seguridad de transmisión e integridad de la señal.

Se va a comparar los tiempos entre I2C y Bus CAN:

Tiempo OffLine de configuración de todos los dispositivos:

Bus CAN – 1 Mbit/s	Tiempos	I2C - 1 Mbit/s Fast mode	Tiempos
260 master * 19 = 4940 Mini- Robots	3,345 s	4 master * 65 * 19 = 4940 Mini- Robots	2,013 s
8 master * 620 = 4960 Mini - Robots	3,596 s	120 master * 41 = 4920 Mini - Robots	1,462 s

Figura 34 : Comparativa I2C y Bus CAN. Tiempo OffLine

El tiempo más crítico es el tiempo de configuración Live. Este tiempo dependerá del número de dispositivos que se hayan posicionado erróneamente. Estos tiempos son muy reducidos con cualquiera de las dos tecnologías cableadas.

Si se envía una tabla de posiciones (28 bytes) a los dispositivos erróneamente posicionados:

Bus CAN – 1 Mbit/s	Tiempos	I2C - 1 Mbit/s Fast mode	Tiempos
260 master * 19 = 4940 Mini-Robots		4 master *65*19 = 4940 Mini- Robots	
✓ 114 erróneos	0,085 s	✓ 114 erróneos	0,065 s
✓ 2.508 erróneos	1,702 s	✓ 2.470 erróneos	1,192 s
✓ 4940 erróneos	3,345 s	✓ 4940 erróneos	2,013 s
8 master * 620 = 4960 Mini - Robots		120 master * 41 = 4920 Mini - Robots	
✓ 100 erróneos	0,1075 s	✓ 123 erróneos	0,047 s
✓ 2.480 erróneos	1,922 s	✓ 2.501 erróneos	0,749 s
✓ 4960 erróneos	3,596 s	✓ 4920 erróneos	1,462 s

Figura 35 : Comparativa I2C y Bus CAN. Tiempo Live enviando Tabla de posiciones

Si se envía una posición (4 bytes) a los dispositivos erróneamente posicionados:

Bus CAN – 1 Mbit/s	Tiempos	I2C - 1 Mbit/s Fast mode	Tiempos
260 master * 19 = 4940 Mini-Robots		4 master *65*19 = 4940 Mini- Robots	
✓ 114 erróneos	0,011 s	✓ 114 erróneos	0,014 s
✓ 2.508 erróneos	0,326 s.	✓ 2.470 erróneos	0,212 s
✓ 4940 erróneos	0,641s.	✓ 4940 erróneos	0,358 s
8 master * 620 = 4960 Mini - Robots		120 master * 41 = 4920 Mini - Robots	
✓ 100 erróneos	0,021 s	✓ 123 erróneos	8,288 ms
✓ 2.480 erróneos	0,380 s	✓ 2.501 erróneos	0,132 s
✓ 4960 erróneos	0,711 s	✓ 4920 erróneos	0,258 s

Figura 36 : Comparativa I2C y Bus CAN. Tiempo Live enviando una posición

4 Diseño

4.1 Solución Final

Ante la dificultad de incorporar soluciones Wireless se decide realizar una solución cableada combinando Ethernet e I2C. Combinando ambas tecnologías se conseguirá reducir el tiempo de configuración de todos los dispositivos evitando el cuello de botella producido desde la unidad central al primer nivel.

También se introducirán multiplexores I2C, reduciendo considerablemente el problema de consumo de potencia al simplificar el hardware involucrado.

El plano focal del telescopio se divide en 6 sectores. Cada uno de estos sectores contará con un módulo Control Device para configurar los dispositivos de su sector.

Dentro de cada sector los dispositivos están agrupados en celdas de 19 dispositivos. Cada celda estará conectada a un canal I2C.

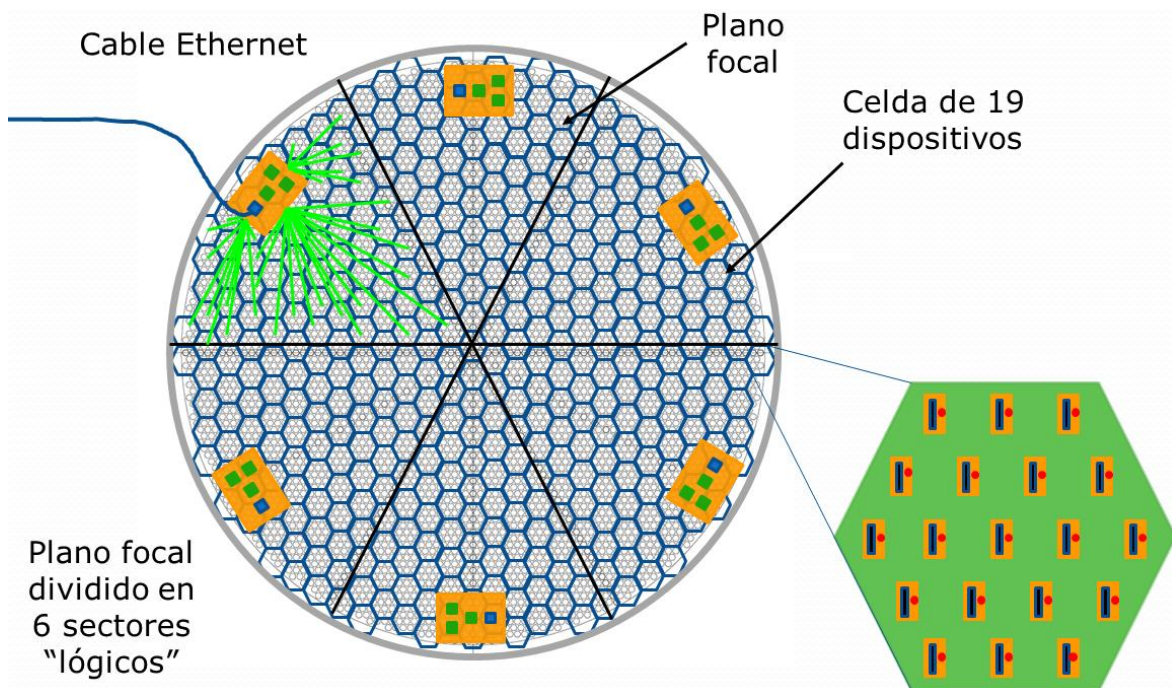


Figura 37 : Distribución plano focal

Desde la unidad central (PC) se conectará seis Control Devices mediante Ethernet, evitando así el cuello de botella. Con ello se conseguirá una velocidad de transmisión a cada Control Device de 1000 Mbps [12]. Cada Control Device será capaz de configurar hasta 912 dispositivos. En total la unidad central será capaz de configurar hasta 5.472 dispositivos, que al ser superior a nuestro objetivo (5.000), no hará falta conectar todas las salidas de los Control Devices.

Ethernet + I²C bus:

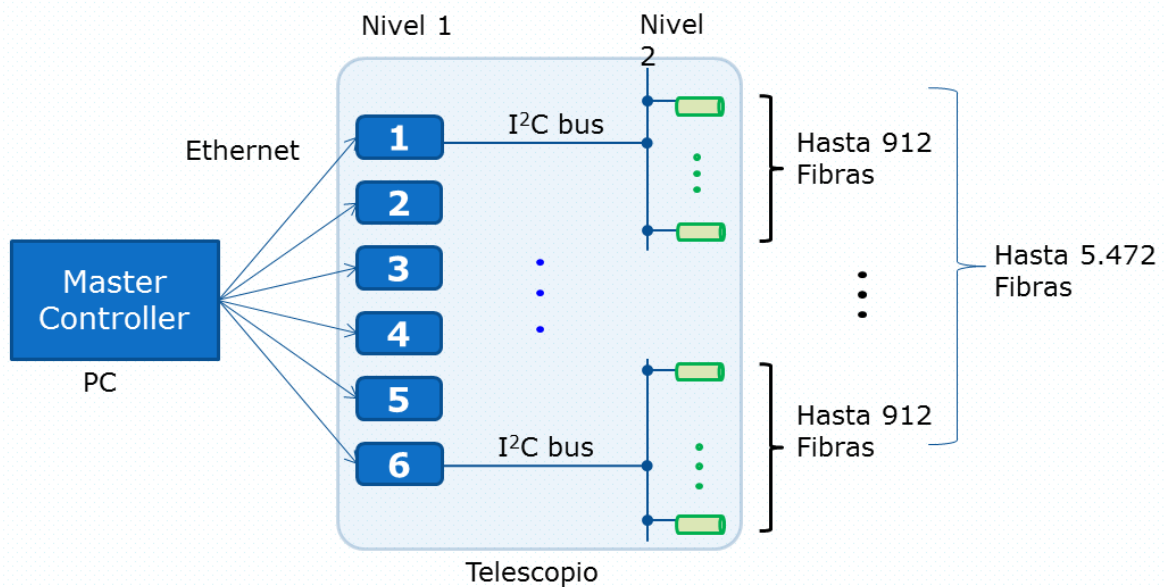


Figura 38 : Diagrama Ethernet + I2C

4.1.1 Control Device

Este módulo se encargará de configurar hasta 912 dispositivos. Para ello contará con un microprocesador (con un puerto Ethernet y un puerto I2C), y siete multiplexores I2C [13].

Cada uno de estos multiplexores tiene ocho canales I2C, donde de cada canal cuelga una celda de 19 dispositivos. Con esta configuración se podrá configurar hasta 5.472 dispositivos en total.

En nuestro caso particular, se quiere configurar aproximadamente 5.000 dispositivos, por lo que no hay necesidad de usar todos los canales de los multiplexores. Realizando dos niveles de multiplexores, con tan solo 7 multiplexores I2C se es capaz de multiplexar hasta 48 celdas de dispositivos posicionadores (912 dispositivos). Por lo tanto si cada Control Device es capaz de multiplexar 44 celdas de dispositivos, se configurarán 836 dispositivos por Control Device.

Los canales I2C libres se podrán utilizar en un futuro si se quisieran manejar más dispositivos posicionadores.

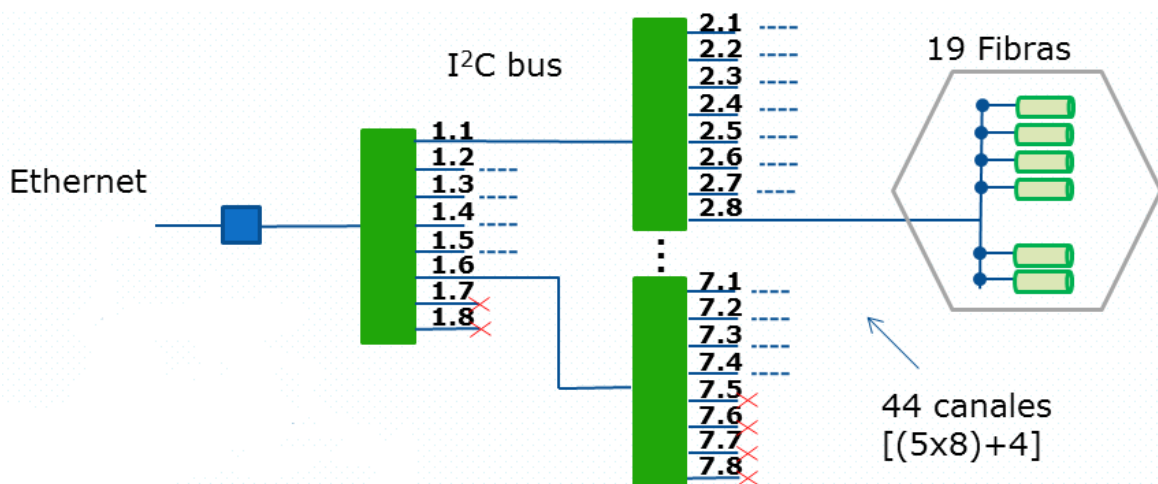


Figura 39 : Diagrama Control Device

4.1.2 Multiplexor I2C

Este componente permite multiplexar varios canales I2C. Utilizando multiplexores en vez de microprocesadores se simplifica el hardware, reduciendo el consumo total de nuestro sistema de comunicación.

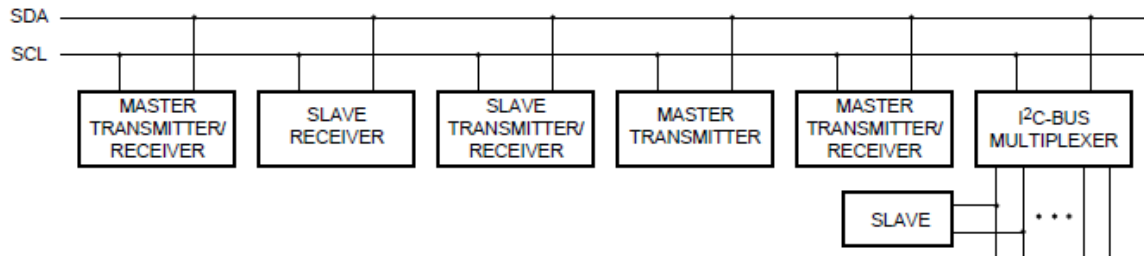


Figura 40 : Diagrama Multiplexor I2C

4.1.3 Regulador

Para simplificar la alimentación de toda la electrónica del BigBOSS (sistema de comunicación, electrónica para el control de los motores...) habrá una única alimentación de 5 V que recorrerá transversalmente el plano focal del telescopio. Todos los componentes electrónicos funcionan con una tensión de 3.3 V por lo que es necesario introducir un regulador de potencia que transforme la tensión de entrada de 5 V a 3.3 V.

4.2 Consumo

En cuanto al consumo de potencia este diseño reduce la complejidad del hardware, el cual era una de las problemáticas de otras soluciones. Como para el prototipo se usará una plataforma de desarrollo basada en un microcontrolador no se puede decir con exactitud el consumo del microprocesador encargado de recibir la información por el puerto Ethernet y enviarla por el puerto I2C. El consumo promedio de microprocesadores con estas características está en torno a los 25 mA. El regulador de potencia tiene un consumo de 0.2 μ A.

Los multiplexores tienen un consumo de 50 μ A cada uno. Como en cada Control Device hay siete multiplexores, se tendrá un consumo total por Control Device de 25.3 mA.

La estructura tiene seis unidades Control Device, por lo que el consumo total de todo el sistema de comunicación es de 152.1 mA. Como la alimentación de la electrónica es de 3.3 V se tendrá un consumo de potencia de 500 mW.

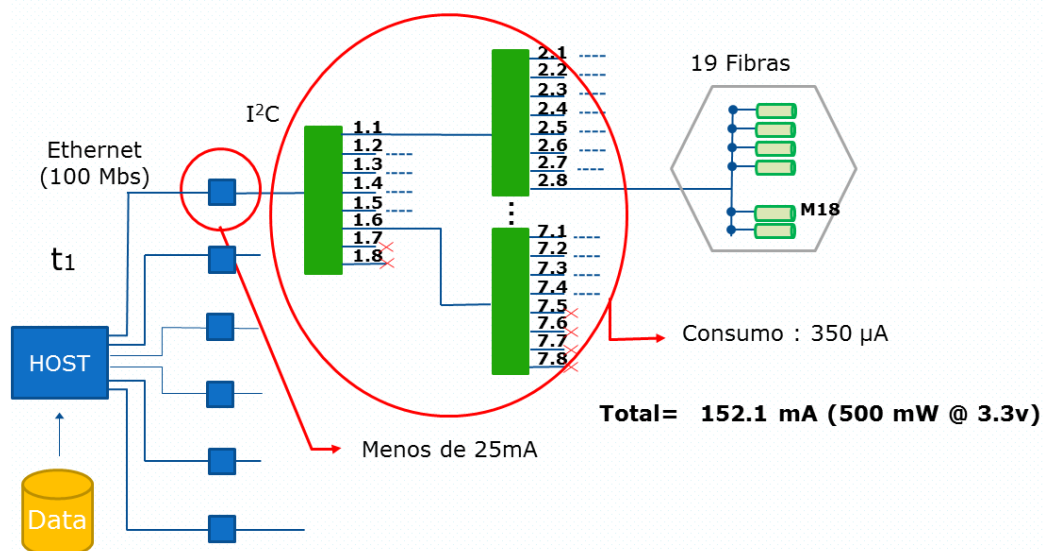


Figura 41 : Diagrama consumos solución final

4.3 Herramientas

Para el diseño de los prototipos se usarán las siguientes herramientas:

1. Altium Designer: Esta herramienta permitirá diseñar los PCB (Printed Circuit Board) de los prototipos a fabricar.
2. NetBeans IDE 7.2.1: Con esta herramienta se desarrollará el programa para el prototipo.

5 Desarrollo

5.1 Elementos del prototipo

5.1.1 BeagleBone

La plataforma de desarrollo elegida para el prototipado es la BeagleBone. BeagleBone es una placa computadora (single – board computer) de hardware libre. Es un ordenador en pequeño pero con el que también se puede controlar hardware externo a través de sus entradas/salidas. Su procesador es un ARM A8 Cortex que funciona a unos 700 MHz y cuenta con 256MB de RAM.

En cuanto a su tamaño, es de unos 9cm de largo y 5,5 de ancho. Cuenta con un lector de tarjetas microSD, un host USB, un conector RJ45 y un micro USB para conectarlo al PC.

En cuanto al sistema operativo se usará Debian (Linux).

Para configurar la placa se conectará por red usando el conector RJ45 incorporado.

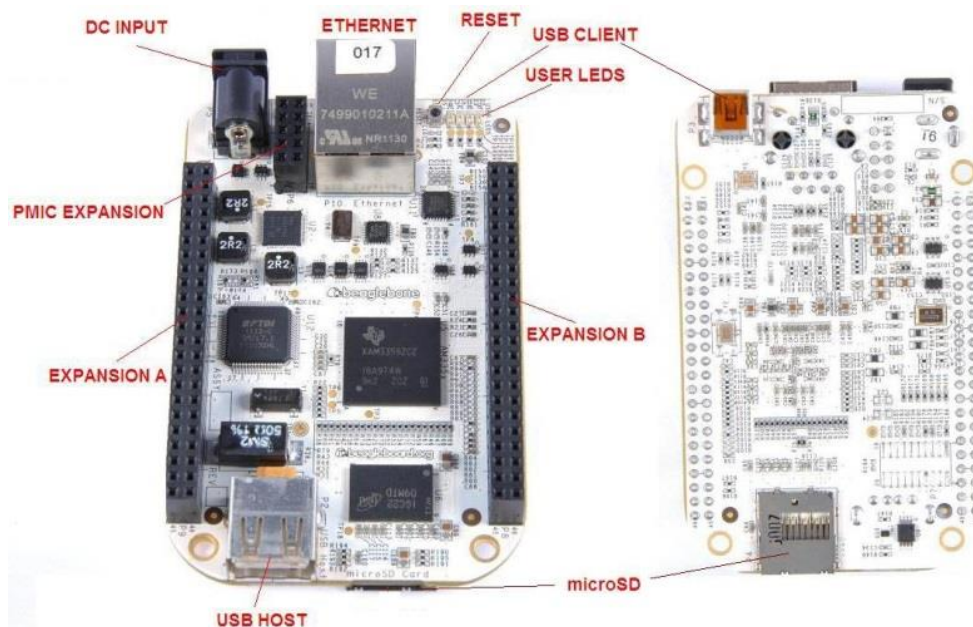


Figura 42 : BeagleBone

Se utilizará la BeagleBone ya que permite simular parte de la placa Control Device. BeagleBone posee Ethernet y dos puertos I2C, uno de los cuales se utilizará para conectar a los multiplexores.

5.1.2 Multiplexor I2C

Se ha elegido el modelo PCA9548A de NXP. Este componente posee una entrada I2C y es capaz de multiplexar hasta 8 canales I2C.

Se destacan estas otras características:

1. Entrada de Reset activa a nivel bajo.
2. 3 pines de direccionamiento permiten controlar la dirección del multiplexor.
3. Selección de uno o varios canales mediante el bus I2C.
4. Los canales se encuentran desactivados por defecto.
5. Permite variar el nivel de tensión del bus I2C entre 1.8 V y 5V.
6. Bajo consumo de corriente.
7. Tensión entre 2.3 V y 5.5 V.
8. Velocidad desde 0 Hz hasta 400 kHz.

El fabricante ofrece tres posibles encapsulados. Para el prototipo se ha elegido el encapsulado PCA9548D debido a su mayor tamaño y su mayor facilidad de montaje.

Type number	Package		
	Name	Description	Version
PCA9548ABS	HVQFN24	plastic thermal enhanced very thin quad flat package; no leads; 24 terminals; body 4 × 4 × 0.85 mm	SOT616-1
PCA9548AD	SO24	plastic small outline package; 24 leads; body width 7.5 mm	SOT137-1
PCA9548APW	TSSOP24	plastic thin shrink small outline package; 24 leads; body width 4.4 mm	SOT355-1

Figura 43 : Encapsulados PCA9548

Por su configuración interna permite seleccionar uno o varios canales I2C, teniendo siempre en cuenta la capacitancia. Esta opción es muy interesante ya que a la hora de configurar los dispositivos permite configurar varias celdas de dispositivos de manera continuada sin tener que reconfigurar el multiplexor.

Cada multiplexor tiene una dirección I2C de 7 bits, 4 bits son fijos y 3 configurables por hardware. Para selección un canal, hay que escribir por el bus I2C en el registro 0 del multiplexor el canal que se desea seleccionar.

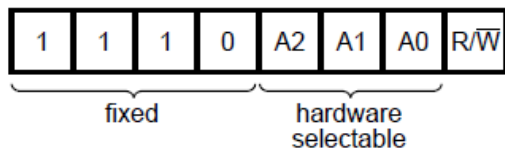


Figura 44 : Dirección Mux I2C

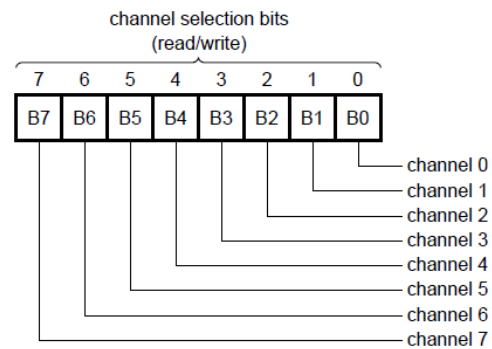


Figura 45 : Configuración canales I2C

La siguiente figura muestra una trama para seleccionar uno o varios canales del multiplexor. Se empieza la comunicación I2C con la condición de START, la dirección del multiplexor y si se desea lectura o escritura. El esclavo (multiplexor I2C) enviará un ACK, y se escribirá en el registro del multiplexor el canal o canales seleccionados. Para terminar la configuración el master enviará la condición de STOP.

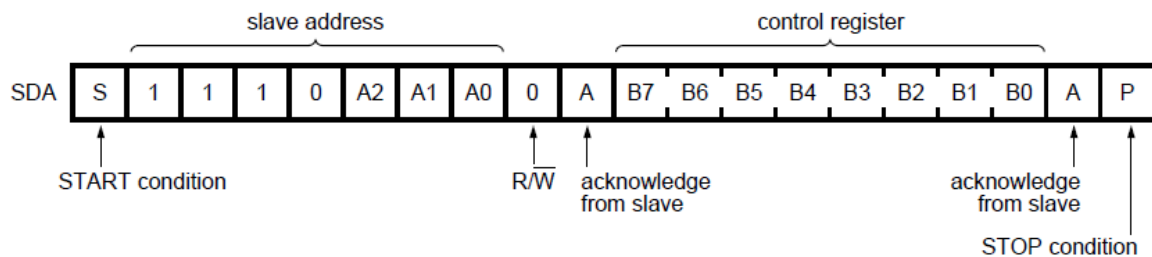


Figura 46 : Trama configuración canal I2C

Para un correcto funcionamiento, tanto las líneas del bus I2C (SDA y SCL) de entrada al multiplexor como las líneas de salida, tendrán unas resistencias de pull-up para limitar la corriente. El estudio sobre los valores de estas resistencias se detallará más adelante.

El consumo del multiplexor es muy reducido, 50 μ A.

5.1.3 Regulador

Para el regulador de tensión se ha utilizado el componente REG103-3.3, ya que directamente permite con una entrada de 5 V sacar una tensión de 3.3 V. Este regulador de bajo consumo posee un pin de ENABLE (activo alto). Ya que toda la alimentación de la electrónica del sistema de comunicaciones es alimentada mediante el regulador, controlando el pin ENABLE se podrá controlar la alimentación de toda la electrónica. Así cuando no se esté utilizando el sistema de comunicaciones se podrá mediante software apagar toda la electrónica del sistema de comunicación reduciendo su consumo.

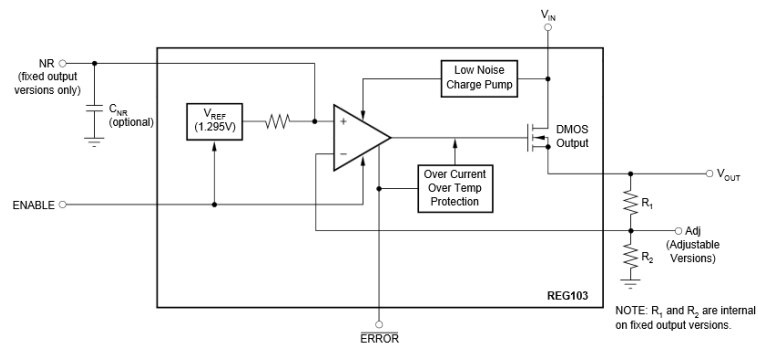


Figura 47 : Diagrama regulador de potencia

5.1.4 Control Device

En el prototipo desarrollado, la placa Control Device se compone de :

1. BeagleBone: Es el encargado de recibir los datos por Ethernet, enviar por I2C la información y configurar los multiplexores.
2. Multiplexores: Encargados de seleccionar el canal de I2C donde cuelga la celda de dispositivos que se quiere configurar.

En la siguiente figura el cuadrado azul representa la BeagleBone, siendo los rectángulos verdes los multiplexores.

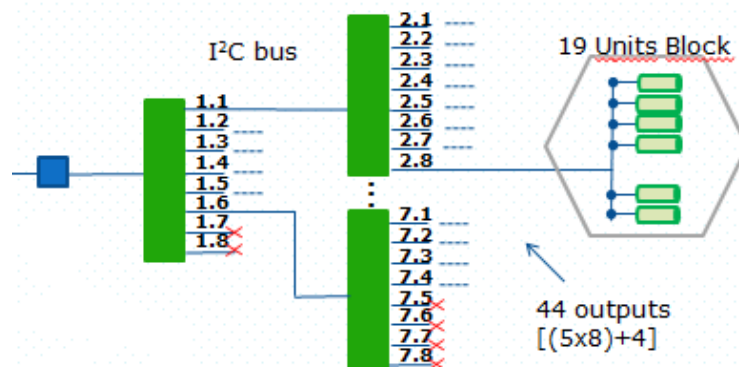


Figura 48 : Diagrama Control Device

5.1.5 Periféricos I2C

Para comprobar el funcionamiento del Control Device, se han diseñado dos dispositivos I2C sencillos.

5.1.5.1 Puerto entrada/salida de 8 bits

Uno de los dispositivos para comprobar el funcionamiento del Control Device es un puerto de entradas y salidas configurable por I2C. Se ha elegido el modelo PCA9554 ya que tiene hasta 8 bits configurables de entradas y salidas, y tiene la posibilidad de configurar hasta 16 direcciones diferentes por hardware.

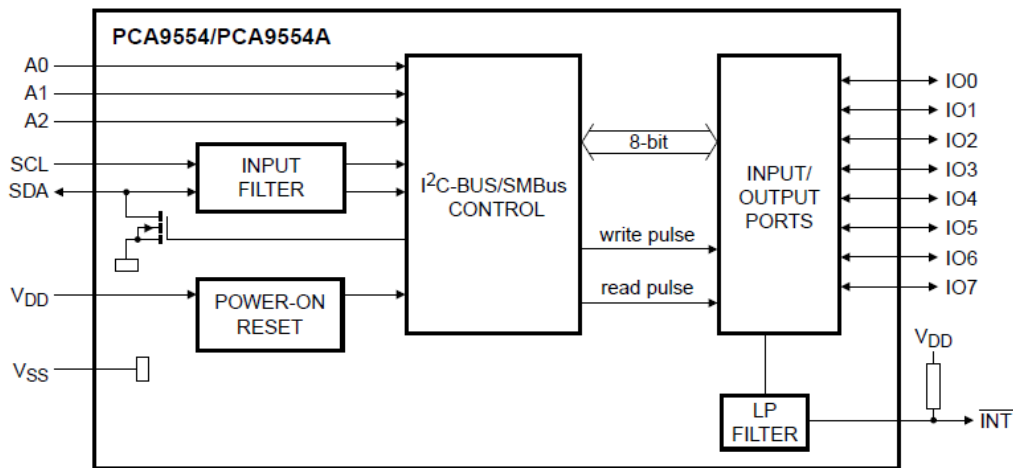


Figura 49 : Diagrama PCA9554

5.1.5.2 Sensor de temperatura

El otro dispositivo diseñado es un sensor de temperatura. Se ha elegido el dispositivo TCN75A. Este dispositivo permite configurar su dirección mediante 3 bits (8 direcciones diferentes) por hardware. Además este dispositivo tiene una salida de alerta cuando se pasa una temperatura establecida por el usuario.

Este dispositivo posee varios registros.

1. Registro de configuración: En este registro se configurará la resolución de la temperatura, el tipo de alerta ...
2. Registro de temperatura: Registro de lectura, leyendo este registro, se obtiene la temperatura.
3. Registro de histéresis: Cuando se activa la salida de alerta, no se desactivará hasta que la temperatura sea inferior al valor establecido por el usuario en este registro.
4. Registro de temperatura límite: Cuando la temperatura supera la temperatura establecida en este registro activa la salida de alerta.

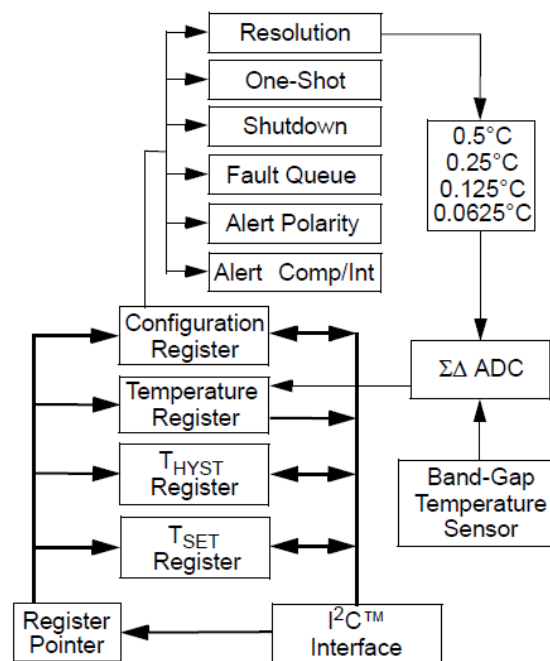


Figura 50 : Diagrama TCN75A

5.2 Diseño

Para el diseño de los PCB se ha utilizado Altium Designer.

5.2.1 Placa Multiplexores I2C v1

Para un primer prototipo se realizó una placa con 3 multiplexores en topología árbol pudiendo seleccionar hasta cuatro canales I2C. El PCB tiene un conector de cinco entradas, dos para tensión (VCC y GND), dos para el bus I2C (SDA y SCL) y un último para la señal de RESET de los multiplexores. El PCB posee cuatro conectores de tres salidas (SDA, SCL y GND). Para las resistencias de pull-up se decidieron valores de 10k (valor estándar para resistencias de pull-up para buses de I2C). Tanto cerca de la toma de tensión del conector de entrada como cerca de la entrada de tensión de los multiplexores hay condensadores de 100nF para evitar picos de tensión indeseados.

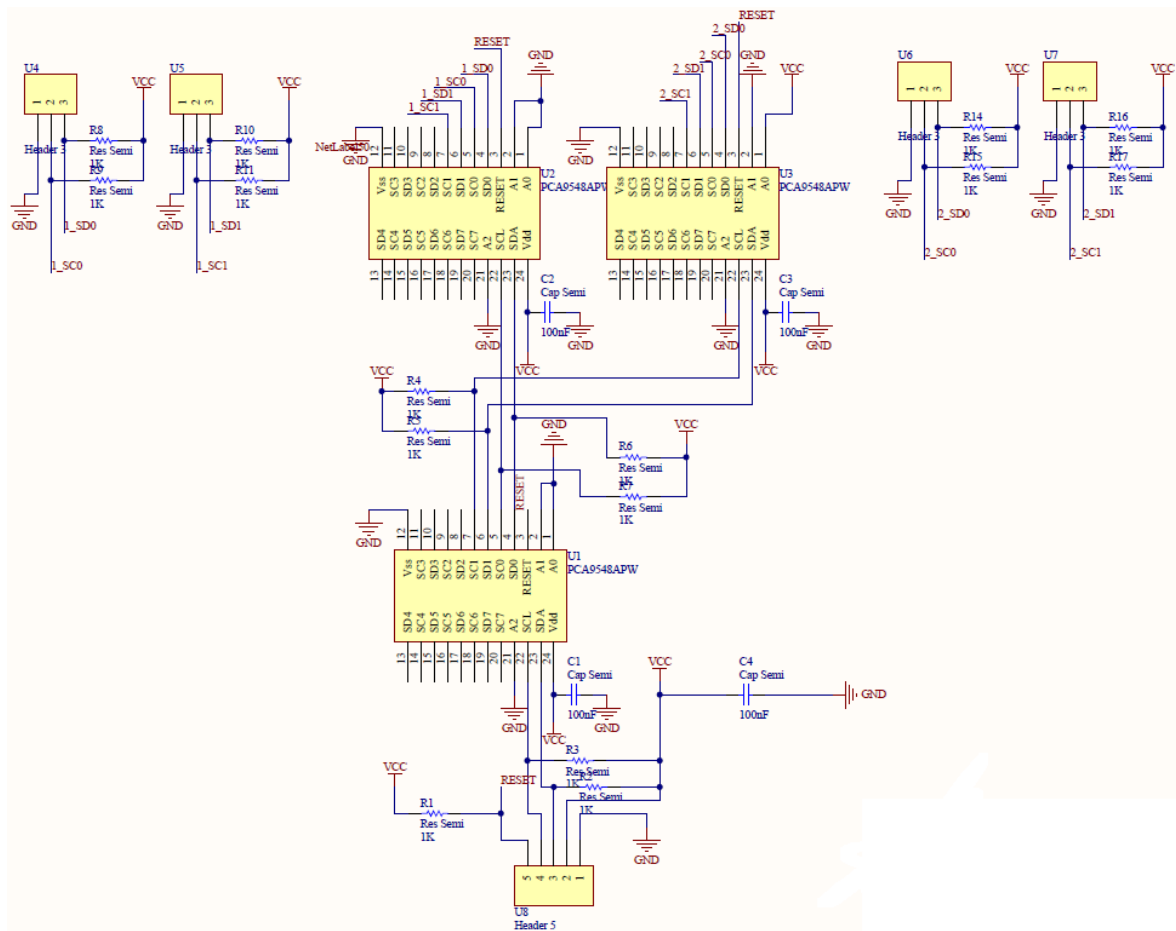


Figura 51 : Esquemático PCB Multiplexor I2C v1

Para la fabricación del PCB se diseña teniendo en cuenta las limitaciones de fabricación de la herramienta de la EPS. Se evitan vías debajo de los componentes electrónicos, se utiliza un ancho de pista de 12 mils, y se ruta en dos capas.

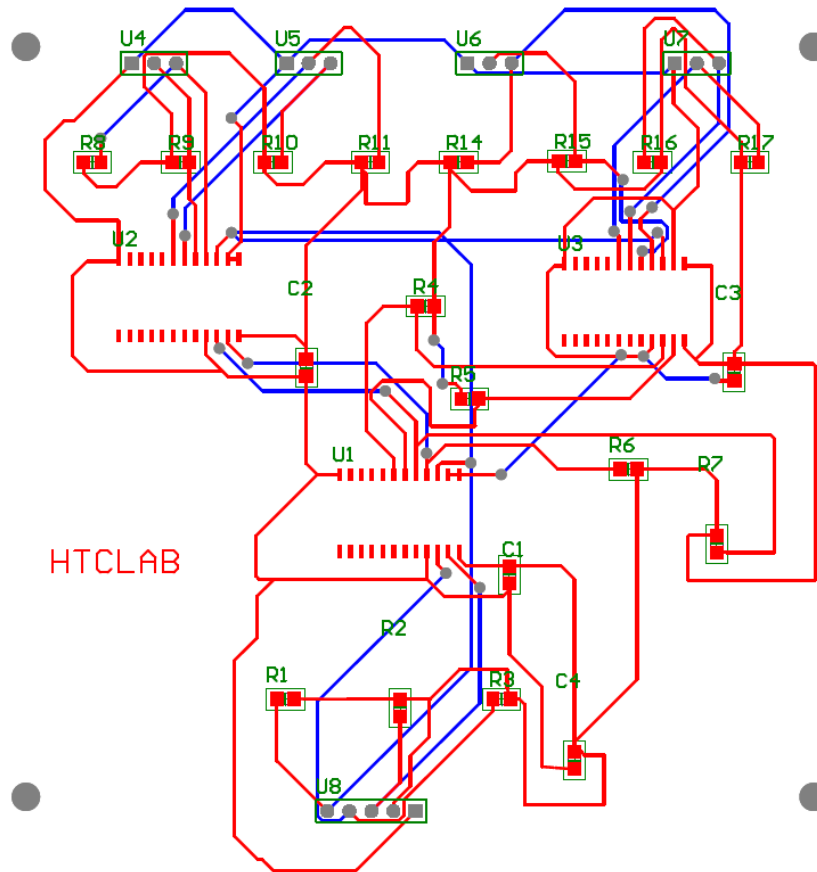


Figura 52 : Diseño PCB Multiplexor I2C v1

Tras la fabricación y montaje, se tiene cuatro conectores que representan cuatro canales I2C:

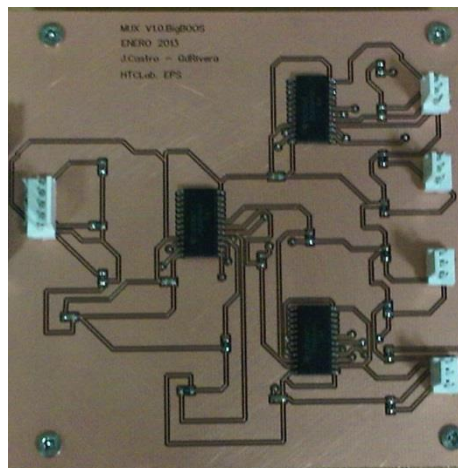


Figura 53 : PCB Multiplexores I2C v1

5.2.2 Placa Multiplexores I2C v2

En esta versión se integran todos los multiplexores, así como un regulador de potencia. Todos los componentes electrónicos se alimentarán con una tensión de 3.3 V, pero la tensión de la que se dispone es de 5 V. Por ello se introduce un regulador de potencia el cual además permite apagar toda la electrónica mediante un pin de ENABLE, reduciendo el consumo cuando no se esté utilizando. Como entrada tiene un conector de seis pines, dos para la tensión, dos para el bus I2C, uno para el RESET de los multiplexores y otro para el ENABLE del regulador. Como salida, hay 48 conectores de tres pines (SDA, SCL y GND). Cada uno de estos conectores es un canal I2C que podrá configurar una celda de 19 dispositivos. También se incorporan condensadores cerca del regulador de tensión como recomendación del fabricante [14].

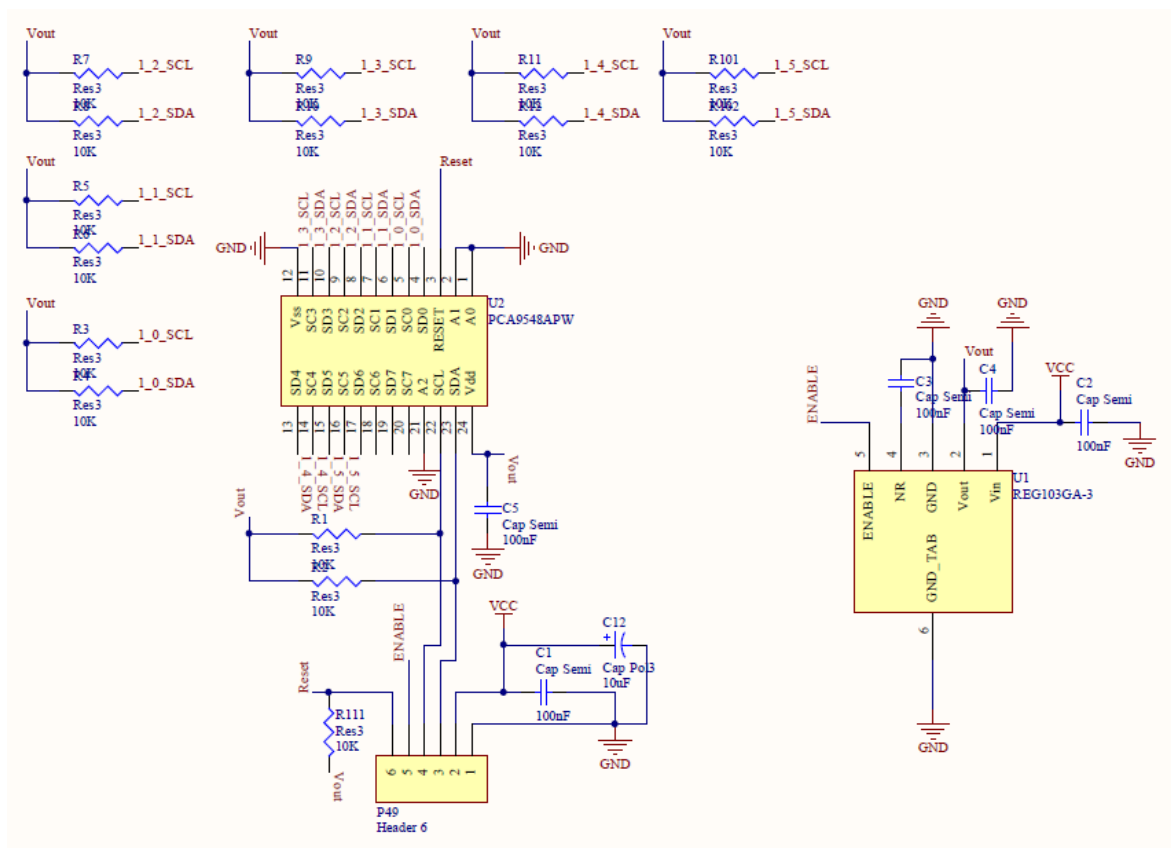


Figura 54 : Sección esquemático PCB Multiplexor I2C V2

El diseño de este PCB es muy complejo debido al alto número de conexiones en el PCB. El tamaño del PCB es de 12.7 cm x 7.9 cm. La fabricación en la EPS (Escuela Politécnica Superior) es inviable, ya que para realizar todas las conexiones rutando por dos caras, habría que realizar vías debajo de los componentes. Para una fabricación de prototipo se ha decidido habilitar un canal de I2C, soldando únicamente los componentes necesarios para que ese canal funcione. Para ello se ha restringido el uso de vías debajo de los componentes necesarios para la funcionalidad del canal. Para una fabricación con todos

los componentes, conectores y la posibilidad de todos los canales I2C habría que fabricar fuera de la EPS.

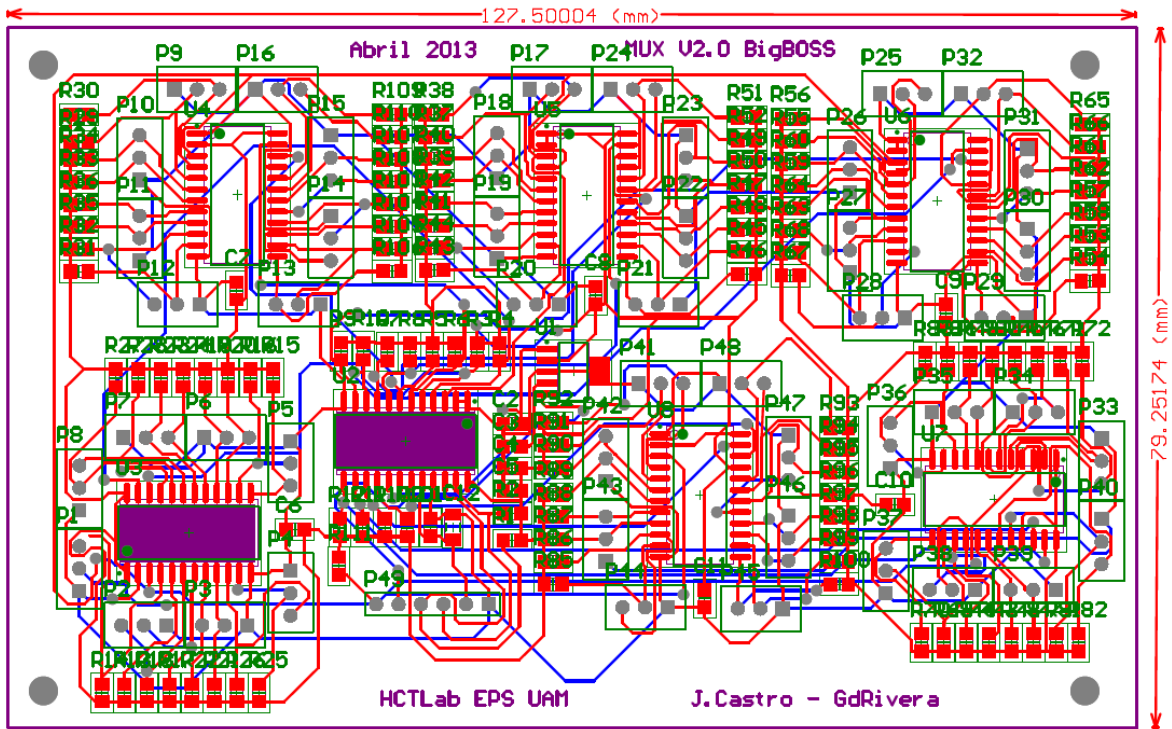


Figura 55 : Diseño PCB Multiplexor I2C v2

Tras la fabricación y montaje de uno de los conectores I2C:

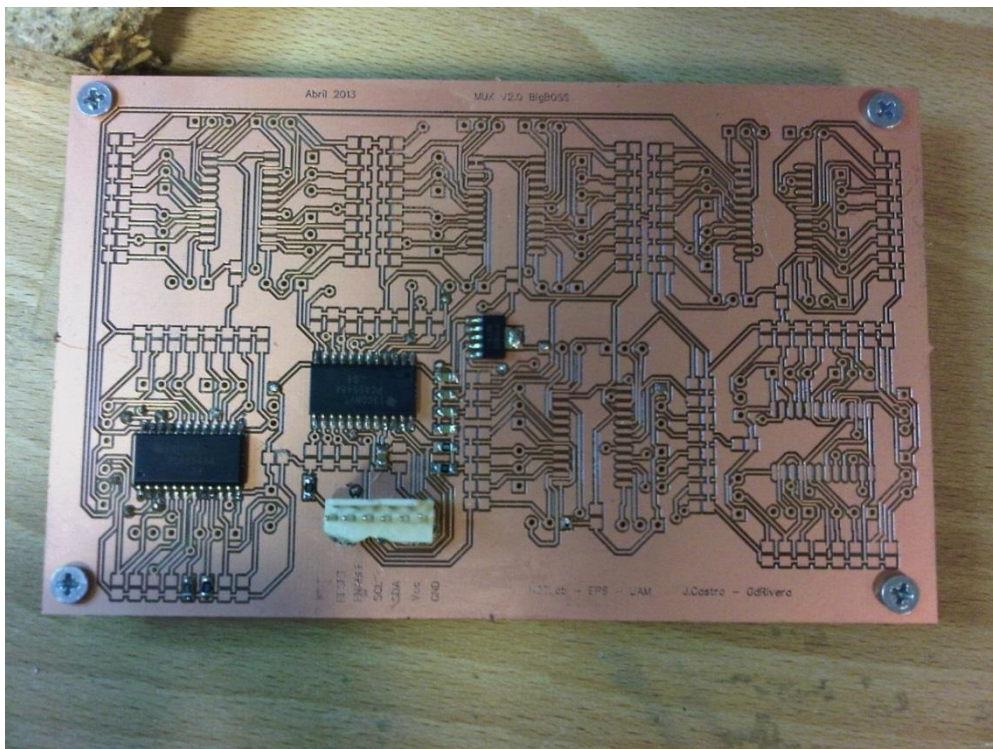


Figura 56 : PCB Multiplexores I2C v2

5.2.3 Puerto entrada/salida de 8 bits

En este PCB se tiene como entrada un conector de cuatro pines, dos de tensión y dos para el bus I2C. De las ocho entradas y salidas del dispositivo se utilizarán dos, una como entrada y otra como salida. La entrada se conectará a un pulsador, y la salida a un diodo led. En el diseño se ha colocado un conector de ocho pines para poder realizar pruebas con las resistencias de pull-up. Además, es posible añadirle condensadores para poder simular una mayor capacitancia en el bus I2C, simulando así que otros dispositivos están en el mismo bus I2C. También se ha dejado un conector de seis pines para poder configurar por hardware la dirección I2C del dispositivo. Se han colocado condensadores para evitar los picos de tensión.

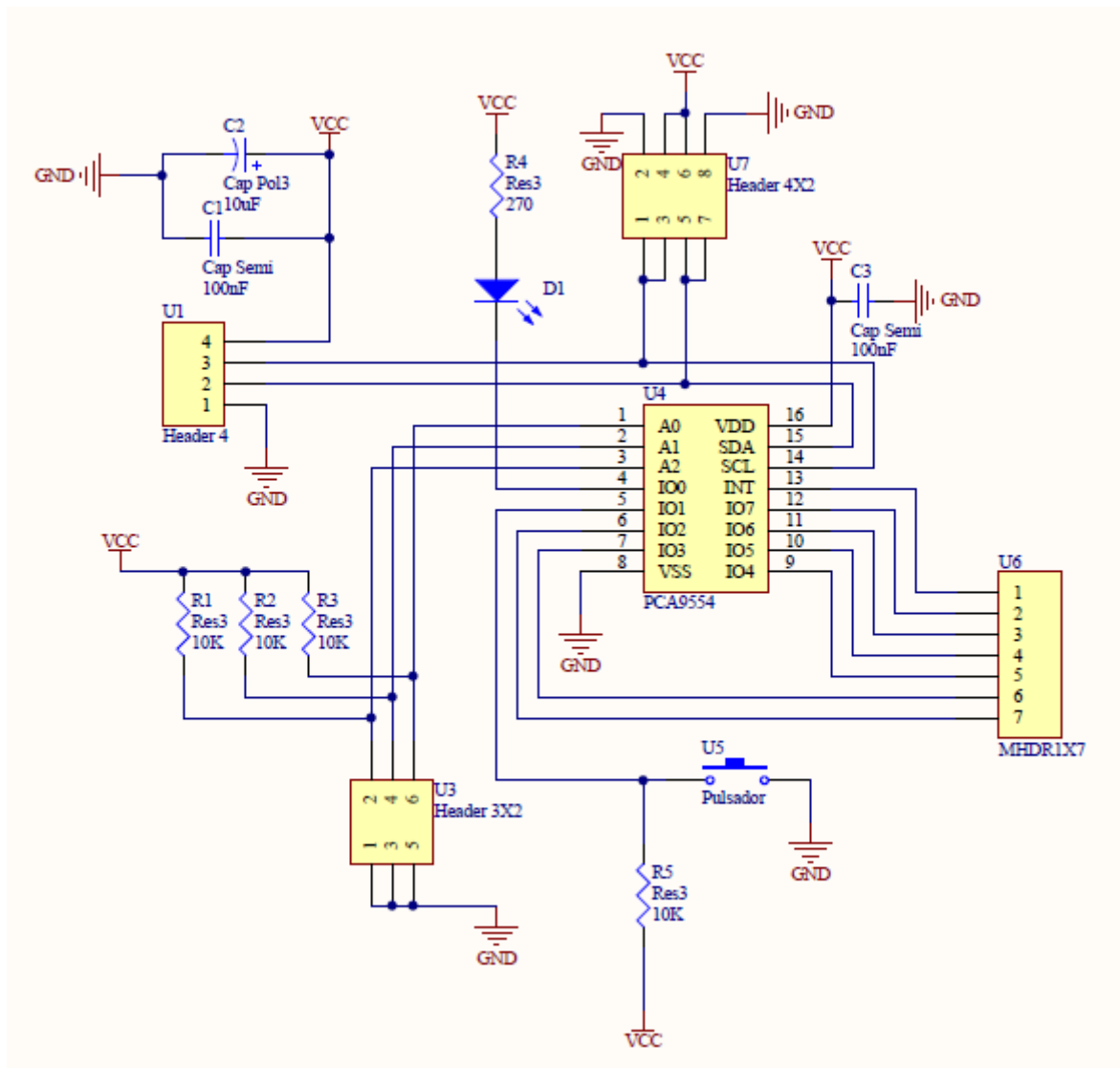


Figura 57 : Esquemático puerto de entrada y salida de 8bits

El tamaño de la placa es de 3.8 cm x 3.3 cm. Se ha utilizado pistas de 12 mils y rutado a dos caras. Se ha dejado un conector de siete pines para las entradas y salidas no utilizadas y para la salida de interrupción.

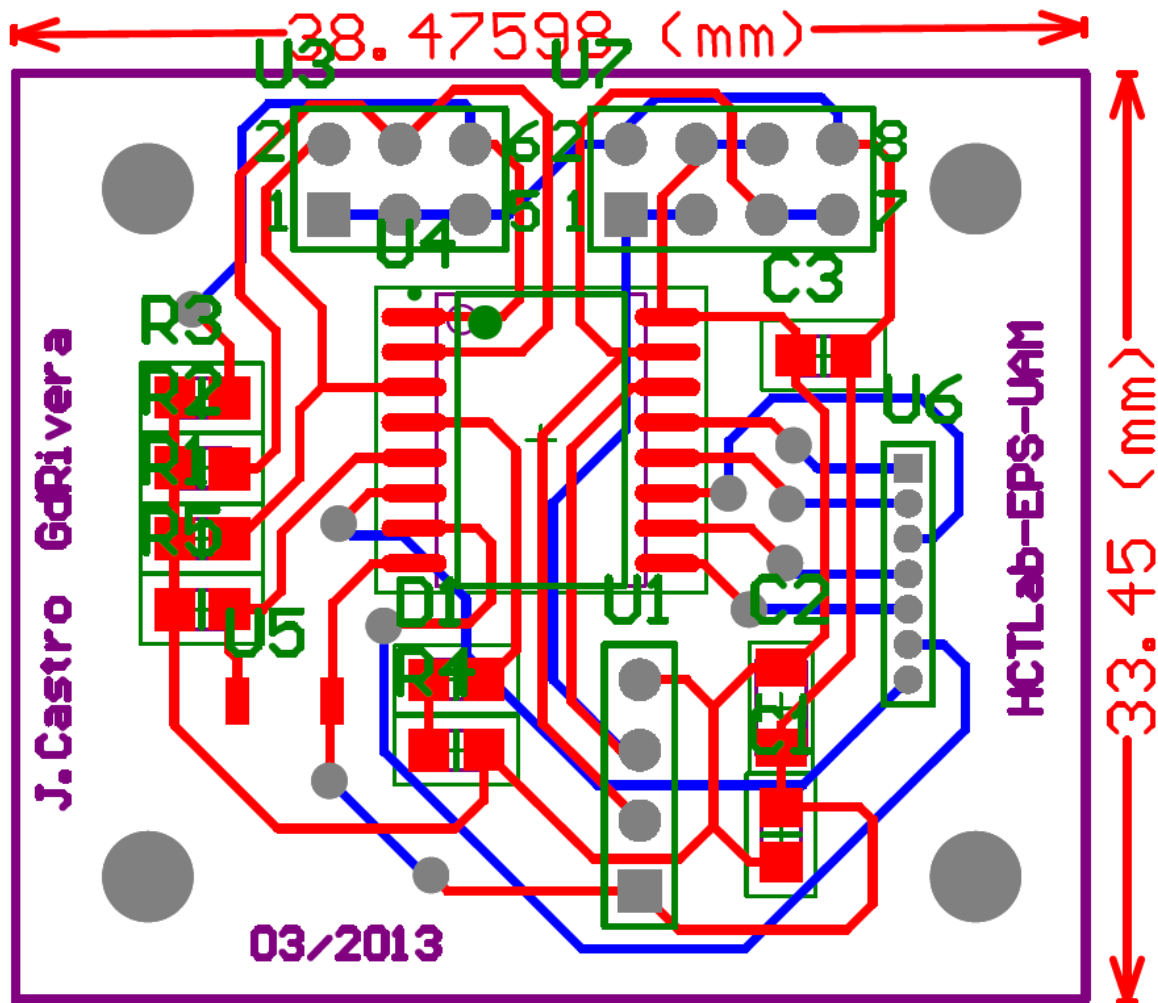


Figura 58 : Diseño PCB Puerto entradas y salidas de 8 bits

5.2.4 Sensor de temperatura

En este PCB se tiene como entrada un conector de cuatro pines, dos de tensión y dos para el bus I2C. En el diseño se han colocado un conector de ocho pines para poder realizar pruebas con las resistencias de pull-up. Además es posible añadirle condensadores para poder simular una mayor capacitancia en el bus I2C, simulando así que otros dispositivos están en el mismo bus I2C. También se ha dejado un conector de seis pines para poder configurar por hardware la dirección I2C del dispositivo. Se han colocado condensadores para evitar los picos de tensión. La salida ALERT se ha conectado a un diodo LED con el fin de cuando se active (a nivel bajo) se encienda dicho LED.

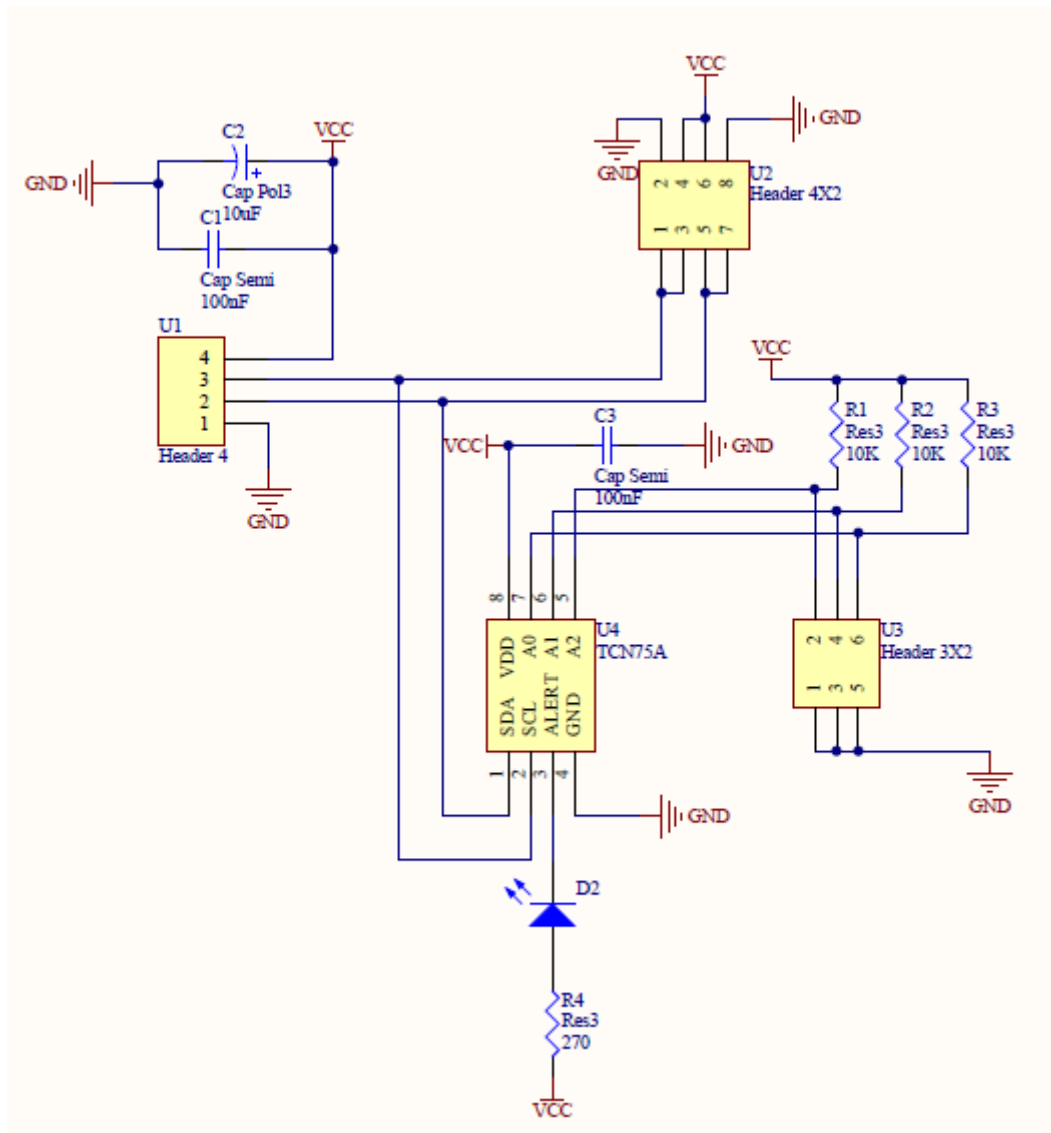


Figura 59 : Esquemático PCB sensor de temperatura

El tamaño de la placa es de 3.7 cm x 2.3 cm. Se ha utilizado pistas de 12 mils y rutado a dos caras.

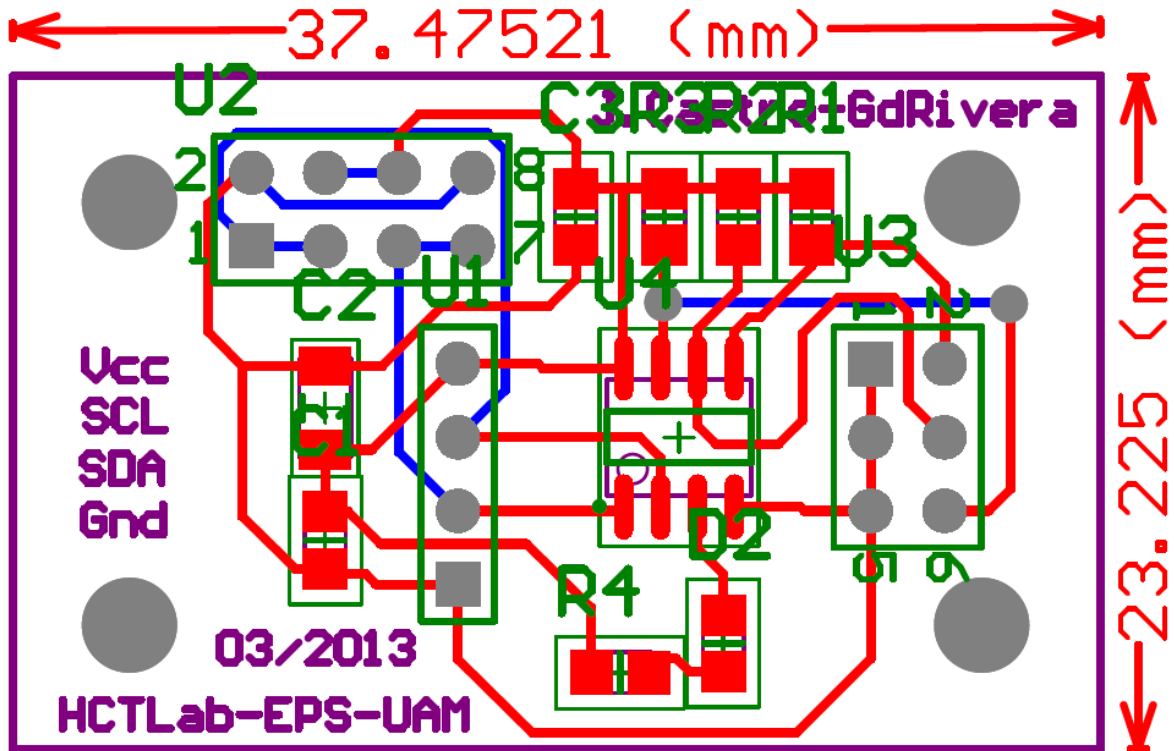


Figura 60 : Diseño PCB sensor de temperatura

6 Integración, pruebas y resultados

6.1 Conexión entre los módulos

La BeagleBone tiene el puerto I2C en el conector P9. Además este conector permite coger toma de tensión y configurar los pines destinados a estradas o salidas.

Con la salida de tensión de 3.3 V de la BeagleBone se alimenta el PCB Multiplexores I2C v1 y los periféricos I2C (Puertos I/O y sensor de temperatura).

Se configura el pin GPIO1_16 como salida, el cual servirá como RESET para los multiplexores.

SIGNAL NAME	PIN	CONN	PIN	SIGNAL NAME
	GND	1	2	GND
	VDD_3V3EXP	3	4	VDD_3V3EXP
	VDD_5V	5	6	VDD_5V
	SYS_5V	7	8	SYS_5V
PWR_BUT*		9	10	A10
UART4_RXD	T17	11	12	U18
UART4_TXD	U17	13	14	U14
GPIO1_16	R13	15	16	T14
I2C1_SCL	A16	17	18	B16
I2C2_SCL	D17	19	20	D18
UART2_TXD	B17	21	22	A17
GPIO1_17	V14	23	24	D15

Figura 61 : Header P9 BeagleBone

Los periféricos I2C se alimentarán desde la Beaglebone y se conectará al canal deseado de la placa de los multiplexores.

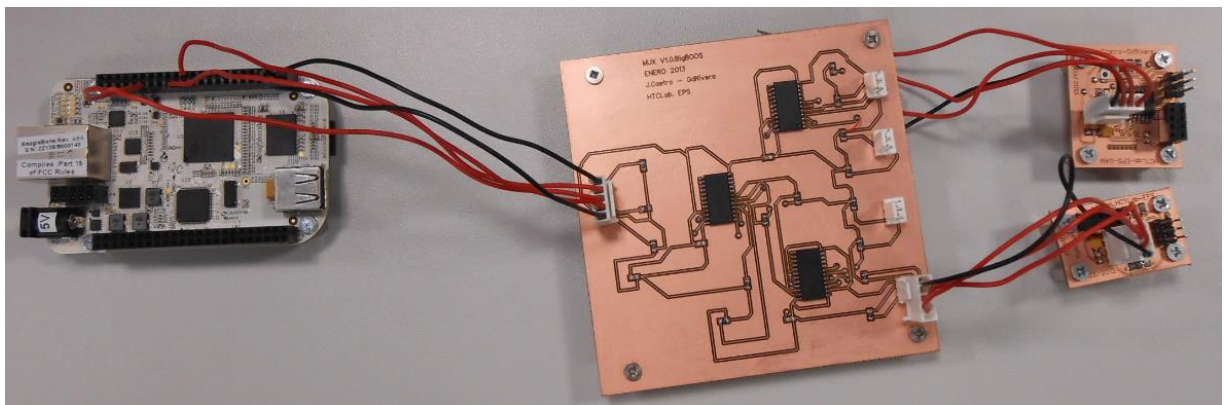


Figura 62 : Conexión entre los módulos

6.2 Software

La BeagleBone está desarrollada para que sea de un manejo fácil y dinámico. Existen librerías para I2C y el propio S.O trae unas herramientas para manejar I2C (I2C Tools).

Mediante unos sencillos comandos se puede manejar el protocolo I2C:

1. I2Cdetect: Este comando detecta todos los componentes I2C conectados al bus. Indica que direcciones están conectadas al bus. Ej: "i2cdetect -y -r 3 "Con este comando se lee los dispositivos conectados al bus I2C 3.
2. I2Cget: Este comando permite leer sobre un dispositivo conectado al bus.
3. I2Cset: Este comando permite escribir en un registro de un dispositivo conectado al bus.

Con este pequeño programa se configura los dos multiplexores y un periférico I2C (Puerto entradas y salidas).

```
/*
 * File:   main.c
 * Author: e109767
 *
 * Created on 5 de marzo de 2013, 10:15
 */

#include <glib.h>
#include <glib/gprintf.h>
#include <errno.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <linux/i2c-dev.h>
#include <sys/ioctl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
/*
 *
 */
int main(int argc, char** argv) {

    system("cd /sys/class/gpio/");
    system("echo 48 > export");
    system("cd /sys/class/gpio/gpio48");
    system("echo out > direction");//Configuramos GPIO1_16 como
Output
    system("echo 1 > value");// Configuramos GPIO1_16 con salida
a 3.3 V
    system("i2cdetect -y -r 3");//Leemos
    system("i2cbus=3,400");
```

```

system("dmesg | grep i2c");
system("i2cset -y 3 0x70 0x02");// Mux 1 Canal 1
system("i2cset -y 3 0x71 0x02");//Mux 2 Canal 1
system("i2cset -y 3 0x27 0x03 0x02");//IO1 como OUTPUT
system("i2cset -y 3 0x27 0x01 0x00");//Todos INPUT a 0

sleep(0.1);

return (EXIT_SUCCESS);
}

```

6.3 Pruebas

Se conecta el PC a la BeagleBone por Ethernet. Se conecta la alimentación (5 V), el RESET y el ENABLE y se conecta el bus I2C del PCB Multiplexores I2C v2 con el Header 9 de expansión de la BeagleBone. La BeagleBone ofrece tres buses I2C. Para nuestra prueba se conecta con el bus I2C - 3 de la BeagleBone. Se conecta el PCB Sensor de temperatura en el canal I2C deseado. Se alimenta el PCB Sensor de temperatura con la BeagleBone con la salida de 3.3 V. Ahora se deberá configurar por software los multiplexores.

En la figura siguiente se puede observar el PCB Sensor de temperatura a la izquierda, el PCB Multiplexores I2C v2 en el centro y la BeagleBone a la derecha.

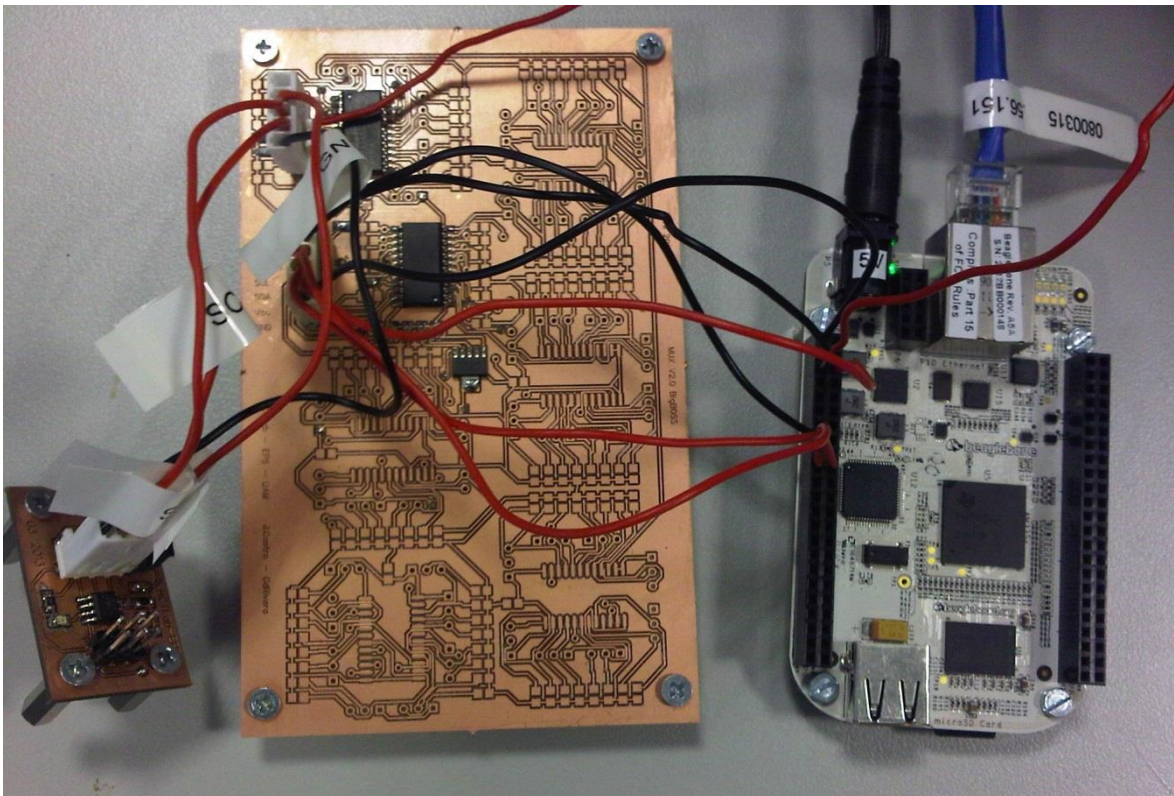


Figura 63 : Conexion de los PCBs

Para esta prueba se conectará por Ethernet estableciendo una sesión SSH desde el PC con la BeagleBone utilizando el programa Putty para realizar la conexión. La dirección IP por defecto de la BeagleBone es 192.168.01.02 puerto 22. Se conectará usando el perfil root para poder configurar las entradas y salidas de la BeagleBone y para poder controlar el bus I2C. Para controlar el bus I2C con la Beagle se utilizará las herramientas I2C del propio sistema operativo (I2C Tools).

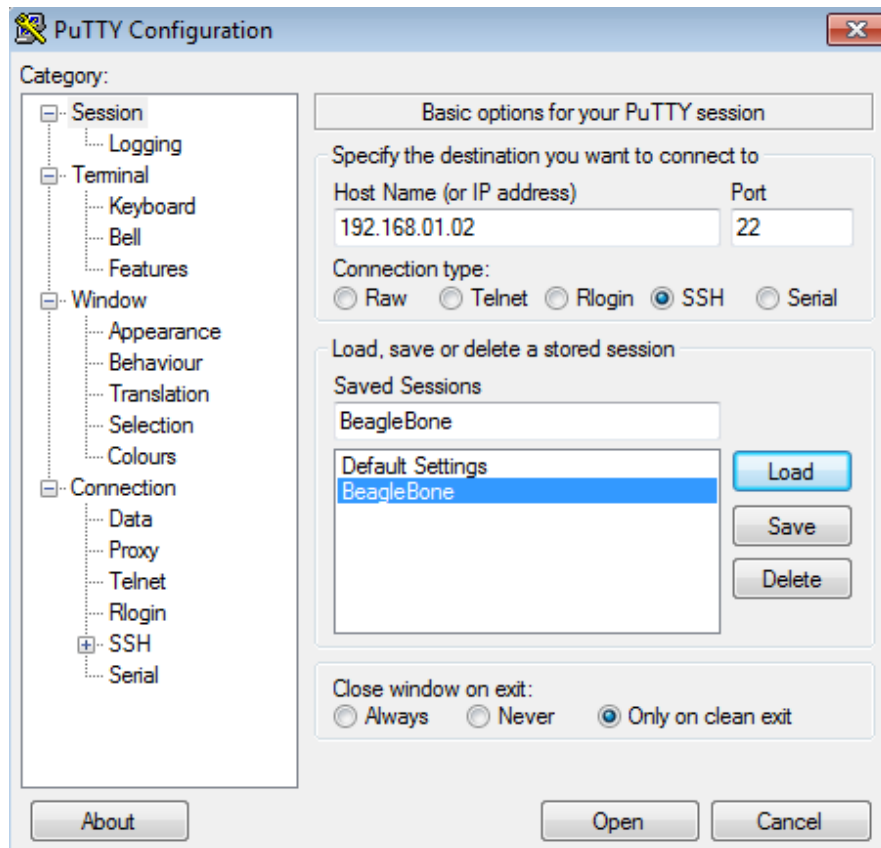
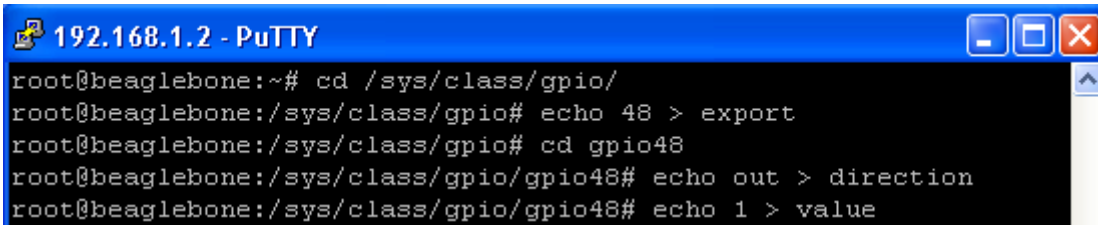


Figura 64 : Conexión Ethernet

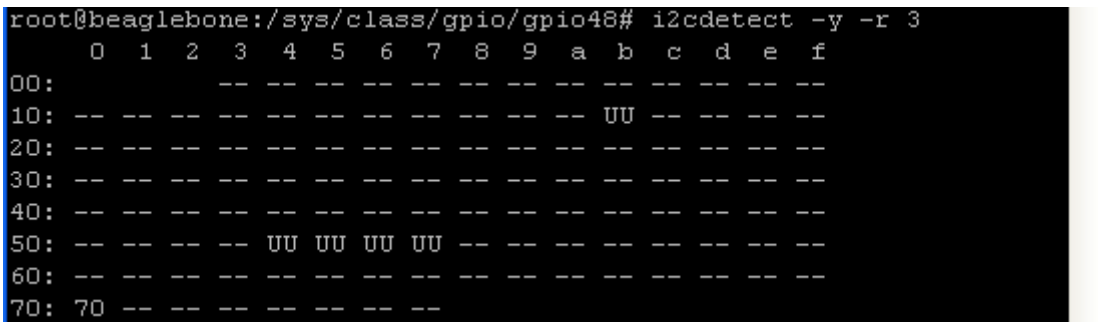
El RESET del PCB de los multiplexores I2C se ha conectado al Gpio1_16 del Header 9. Para configurarlo se tendrá que ir a la carpeta `"/sys/class/gpio/"`. Los cálculos para saber el pin que se quiere configurar depende de la numeración, cada Header contiene 32 pines, y como se quiere configurar el pin 16 del Header 2, por lo tanto se tiene que modificar el pin 48. Se accede a la carpeta de este pin y se modifica el fichero de dirección. Como se quiere que sea una salida controlada se ejecuta el comando `"echo out > direction"`. Según las especificaciones el RESET es activo a nivel bajo, por lo que cuando se quiera resetear la configuración se enviará el comando `"echo 0 > value"`. Este comando modificada el fichero con el valor deseado para esa salida lógica. La tensión de los pines de salida puede ser 0 V o 3.3 V.



```
192.168.1.2 - PuTTY
root@beaglebone:~# cd /sys/class/gpio/
root@beaglebone:/sys/class/gpio# echo 48 > export
root@beaglebone:/sys/class/gpio# cd gpio48
root@beaglebone:/sys/class/gpio/gpio48# echo out > direction
root@beaglebone:/sys/class/gpio/gpio48# echo 1 > value
```

Figura 65 : Configuración Gpio1_16 como RESET

Mediante el comando `i2cdetect`, se observa qué dispositivos están conectados al bus. Como se puede observar en la figura siguiente, hay un dispositivo conectado al bus I2C con dirección `0x70`. Este dispositivo es el primero de nuestros multiplexores. Por defecto todos los canales del multiplexor se encuentran desactivados, por lo que se deberá configurarlo para activar el canal o canales deseados.



```
root@beaglebone:/sys/class/gpio/gpio48# i2cdetect -y -r 3
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  UU  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  UU  UU  UU  UU  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  70  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Figura 66 : Bus I2C -3

Para seleccionar el canal deseado del multiplexor hay que escribirlo en el registro del multiplexor. Para eso se usa la instrucción `"i2cset -y 3 0x70 0x01"`. Esta instrucción indica que se desea escribir `0x01` en el registro del dispositivo con dirección `0x70` del bus I2C -3 de la BeagleBone. En nuestro ejemplo al escribir `0x01` en el registro del dispositivo se está seleccionando el canal 0 del multiplexor, donde está conectado el siguiente multiplexor. Tras configurar el primer multiplexor se ejecuta otra vez el comando `"i2cdetect -y -r 3"` para ver los dispositivos conectados al bus I2C- 3 y se observa como aparecen las direcciones `0x70` y `0x71`. La dirección `0x70` corresponde al multiplexor 1 y la dirección `0x71` corresponde al multiplexor 2. Si se quisiera seleccionar varios canales a la vez, habría que escribir en el registro del multiplexor los canales que se quieran seleccionar. Ej: Si se quisiera seleccionar los canales 0 y 1 del multiplexor, habría que escribir en el registro el dato `"0x03"`.

```

root@beaglebone:/sys/class/gpio/gpio48# i2cset -y 3 0x70 0x01
root@beaglebone:/sys/class/gpio/gpio48# i2cdetect -y -r 3
   0 1 2 3 4 5 6 7 8 9 a b c d e f
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- UU -- -- -- --
20:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
50:  -- -- -- -- UU UU UU UU -- -- -- -- -- --
60:  -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: 70 71 -- -- -- -- -- --

```

Figura 67 : Configuración primer multiplexor

En la siguiente figura se observa el envío de la trama I2C. La señal de arriba corresponde a la señal SDA (datos), siendo la señal de abajo SCL (reloj). Se recuerda que la trama I2C consta de: Condición START + Dirección + Lectura/Escritura + ACK (activo bajo) + Datos + ACK+ Condición de STOP.

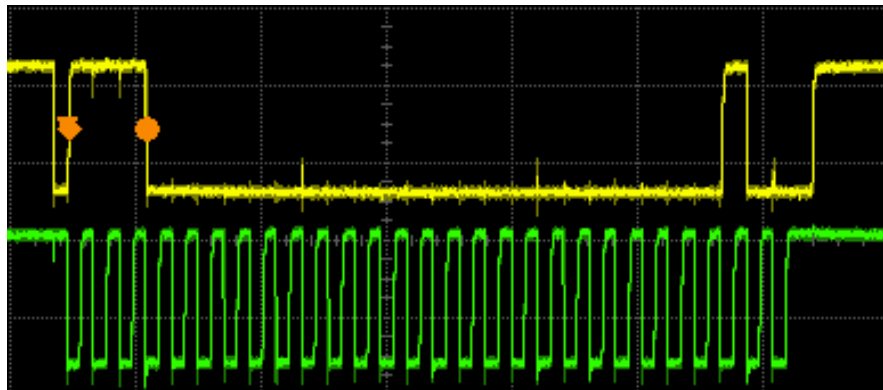


Figura 68 : Trama I2C configuración multiplexor 1

Por defecto en el segundo multiplexor también están desactivados todos los canales. Como en el multiplexor anterior hay que escribir en el registro el canal deseado. En nuestra prueba se quiere habilitar el canal 2, por lo que se usa la instrucción “i2cset -y 3 0x70 0x04”. Tras habilitar el canal 2 del segundo multiplexor se accederá a los dispositivos conectados a ese canal. Se ejecuta la instrucción “i2cdetect -y -r 3” para ver los dispositivos conectados al I2C-3 y se observa que aparece un nuevo dispositivo con dirección “4d”. El dispositivo con dirección “4d” corresponde al sensor de temperatura. Una vez establecida la configuración de los multiplexores, para la BeagleBone el sensor de temperatura está conectado a su mismo bus I2C por lo que ya se podrá leer/escribir del dispositivo.

```

root@beaglebone:/sys/class/gpio/gpio48# i2cset -y 3 0x71 0x04
root@beaglebone:/sys/class/gpio/gpio48# i2cdetect -y -r 3
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- UU -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- 4d -- -- --
50: -- -- -- -- UU UU UU UU -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: 70 71 -- -- -- -- -- -- -- -- -- -- -- -- --
root@beaglebone:/sys/class/gpio/gpio48#

```

Figura 69 : Configuración segundo multiplexor

El envío de esta segunda trama para configurar el segundo multiplexor ha sido correcto.

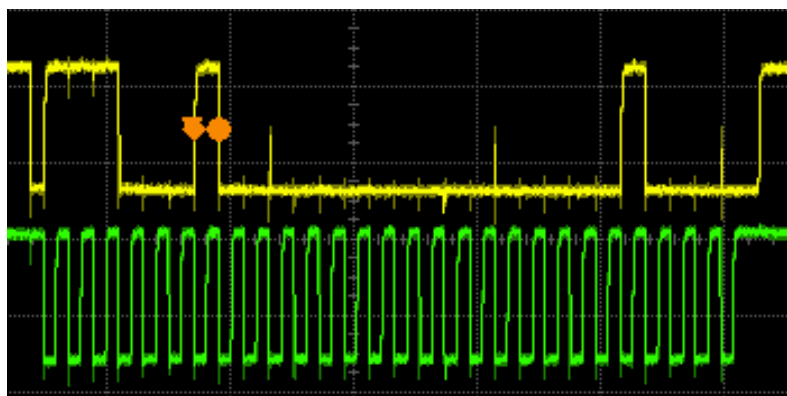


Figura 70 : Trama I2C configuración multiplexor 2

Para comprobar la conexión con el dispositivo final se leerá la temperatura. La temperatura se almacena en el registro 0. Por ello se ejecuta la instrucción “i2cget -y 3 0x4d 0x00”.

```

root@beaglebone:~# i2cget -y 3 0x4d 0x00
0x17

```

Figura 71 : Lectura del sensor de temperatura

La trama I2C para lectura es más larga debido a que el master tiene que señalar sobre que registro desea leer, y a continuación acepta los datos de ese registro.

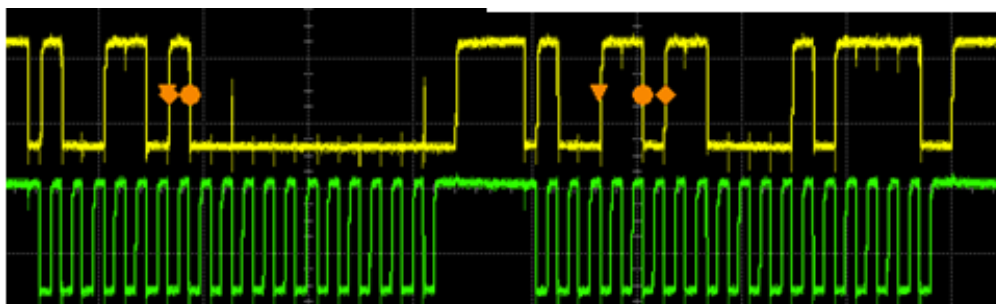
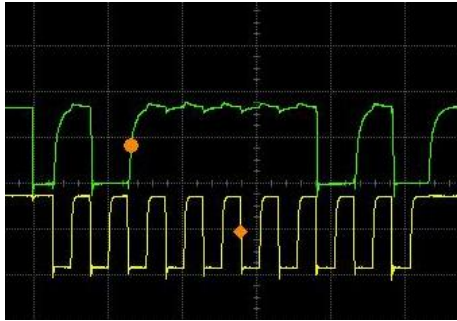
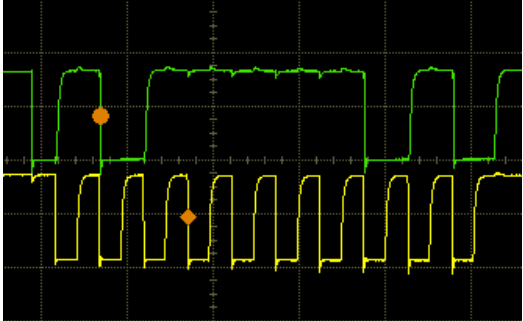
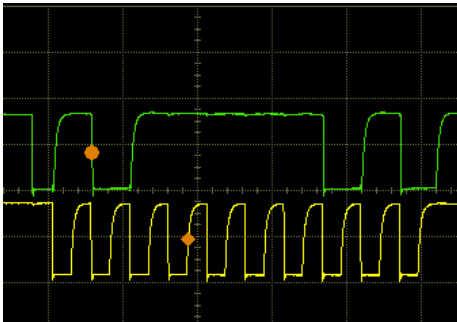
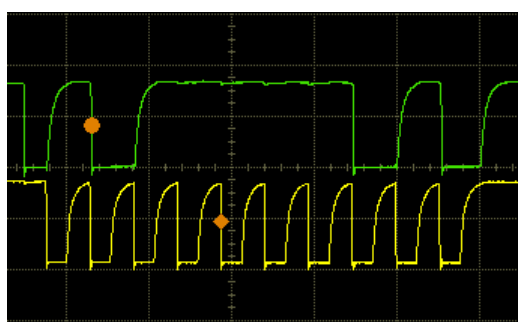
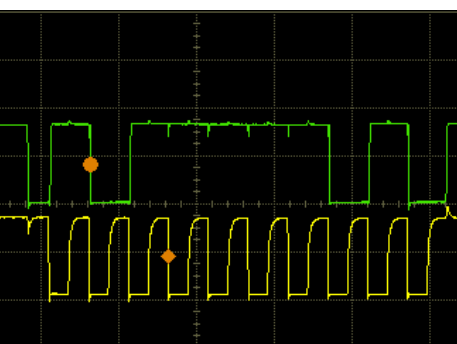
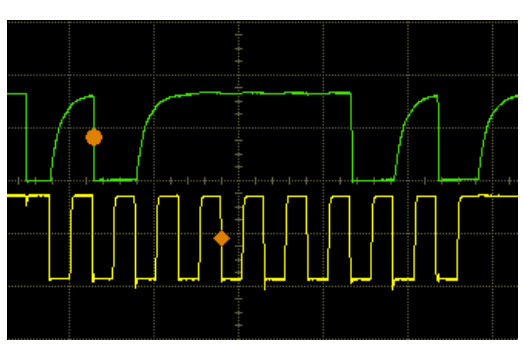


Figura 72 : Trama I2C lectura del sensor temperatura

Para comprobar la integridad de la señal se realizan diversas pruebas. La primera prueba es la distancia del bus I2C. Se probará el comportamiento de la señal si la longitud del bus es de 0m o de 1m.

Las entradas de los microcontroladores que van a manejar los motores tienen una capacitancia de 20 pF. Para ello se simulará cargando el bus con 100pF ,400pF y 1000pF. La simulación con una capacitancia añadida de 400pF, que es la que más se acerca a nuestro caso, ya que se simularía 20 dispositivos conectados al bus I2C (cada canal I2C se conecta con una celda de 19 dispositivos).

Por último se realizará las pruebas con diversas resistencias de pull-up, con valores de 2kΩ, 4.7kΩ y 10kΩ. Las pruebas representadas a continuación están realizadas con una longitud de bus de 1m.

	2kΩ	4.7kΩ
100pF		
400pF		
1000pF		

	10k Ω
100pF	
400pF	
1000pF	

Tras estas pruebas se observa que a mayor capacitancia en el bus I2C más se distorsiona la señal. En la simulación con mayor capacitancia, 1000pf (50 dispositivos), la señal I2C se comporta correctamente. Con este estudio se afirma la viabilidad para que un canal de I2C del multiplexor pueda configurar todos los dispositivos de una celda.

En cuanto a las resistencias de pull-up aunque se observa una calidad de la señal suficiente para cualquiera de los valores propuestos, se observa una mayor calidad de la señal para resistencias pull-up de 2k Ω .

7 Conclusiones y trabajo futuro

7.1 Conclusiones

1. Tras el estudio de las tecnologías actuales la solución propuesta presenta eficiencia tanto en consumo de potencia como en tiempo total de configuración de los dispositivos.
2. Se ha comprobado la viabilidad de solución propuesta para el problema planteado, y para extensiones de mismo (mayor número de dispositivos por celda, mayor número de dispositivos a configurar, cambio en la estructura lógica del plano focal ...)
3. La fabricación de un prototipo final es muy complejo para la fabricación en la EPS, pero las pruebas realizadas con un prototipo simplificado han dado validez al diseño.
4. Se resuelve la configuración de un alto número de dispositivos con un hardware simple y de bajo consumo.

7.2 Trabajo futuro

7.2.1 Integración microprocesador

Este proyecto deja varias líneas de trabajo para un futuro. La primera es desarrollar el prototipo final integrando el microprocesador con puerto Ethernet y puertos I2C en la placa de los multiplexores. Integrando el microprocesador se conseguiría el módulo Control Device en una PCB de dimensiones reducidas.

7.2.2 Integración módulos

Otra posible mejora es la integración de módulos que aseguren la integridad de la señal en largas distancias. Para ello existen repetidores de I2C los cuales amplifican la señal. Para grandes distancias entre el Control Device y los dispositivos a configurar se podría conectar un módulo con un repetidor I2C. Como parte del proyecto se ha diseñado un prototipo para un módulo de repetidor I2C.

7.2.2.1 Repetidor I2C

Un repetidor I2C es un componente que amplifica la señal de un bus I2C. Es muy útil para situaciones en las que puede haber un deterioro en la señal (largas distancias, interferencias electromagnéticas...). Un ejemplo de repetidor I2C es el componente PCA9517 del fabricante NXP. Este componente además permite cambiar los niveles de tensión del bus I2C. Esto es muy útil si se está implementando un bus I2C a 5 V desde el master, pero el dispositivo esclavo trabaja a 3.3 V. El repetidor PCA9517 permite trabajar a velocidades de hasta 400 Kbps.

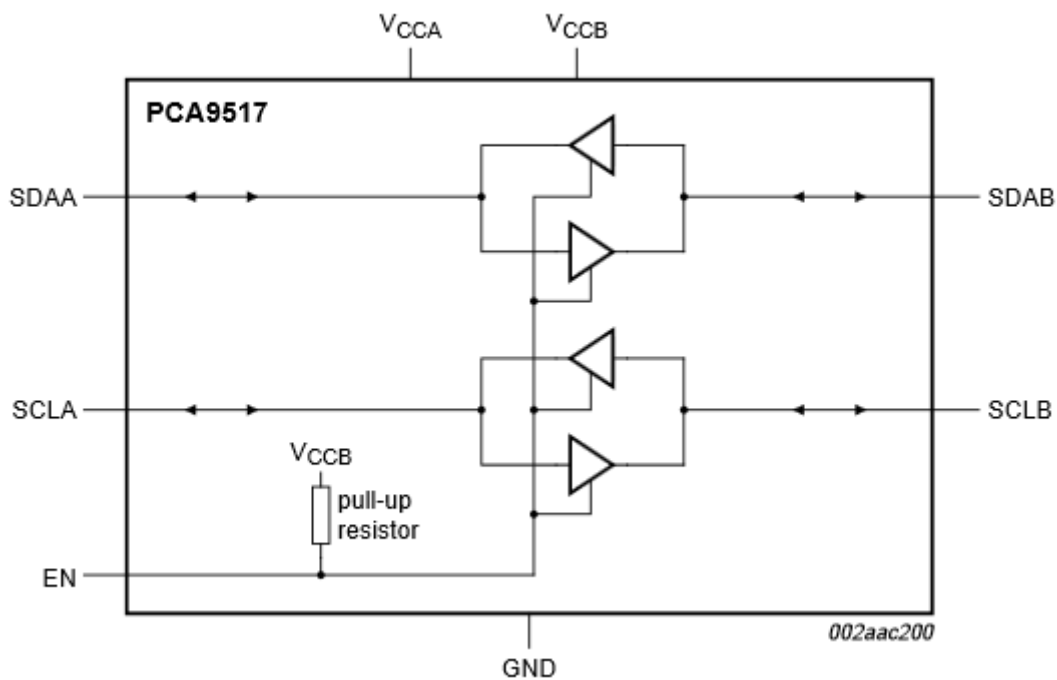


Figura 73 : Diagrama repetidor PCA9517

7.2.2.2 Diseño

En el diseño del esquemático aparece el repetidor PCA9517 y el regulador REG103GA-3. Como entrada se tiene dos conectores, uno para la alimentación (Vcc y GND) y otro para el bus I2C (SDA, SCL y GND). Como la alimentación en el plano focal es de 5 V se necesita un regulador que transforme la tensión en 3.3 V. Como salida se tiene un conector de 5 pines. Estos 5 pines corresponden al bus I2C (SDA, SCL y GND), y la alimentación (3.3 V y 5V) por si se quisiera alimentar los dispositivos mediante este módulo. Se han incorporado condensadores en la entrada de alimentación de la PCB y de los chips para prevenir picos de tensión. También se incorporan condensadores cerca del regulador de tensión como recomendación del fabricante.

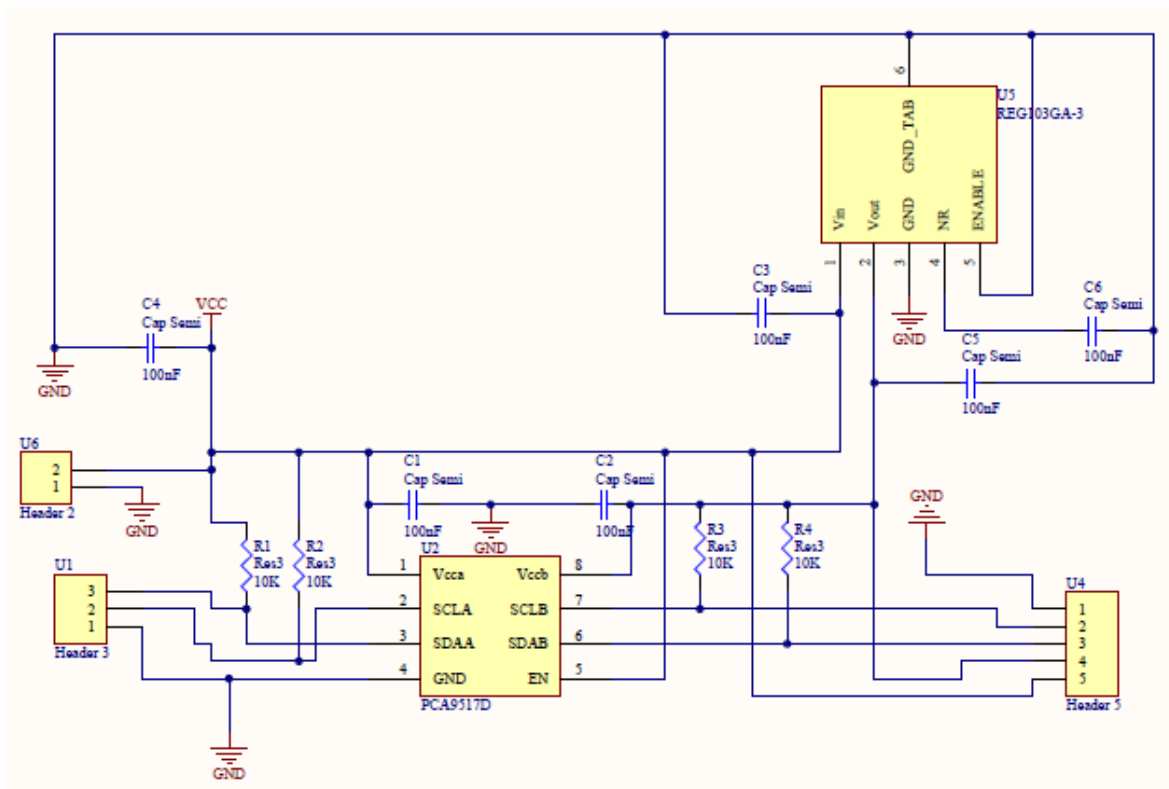


Figura 74 : Esquemático PCB Repetidor I2C

Para el diseño de este PCB se han utilizado pista de 12 mils y rutado a dos caras.

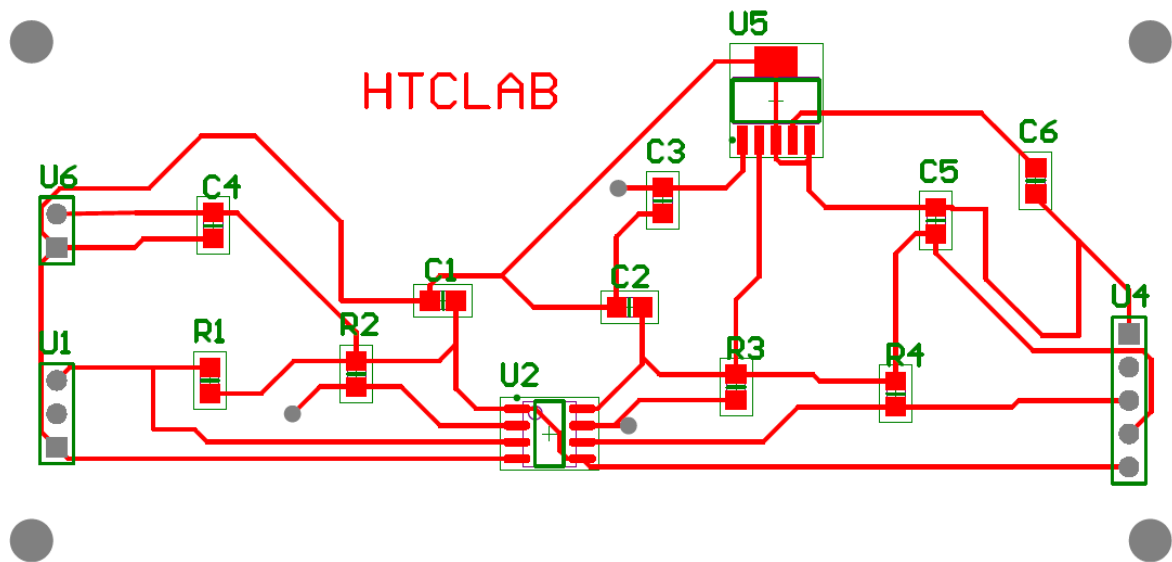


Figura 75 : PCB Repetidor I2C

En la siguiente figura se puede observar la BeagleBone a la izquierda, el PCB Multiplexores I2C v1 en el centro y el PCB Repetidor I2C a la derecha.

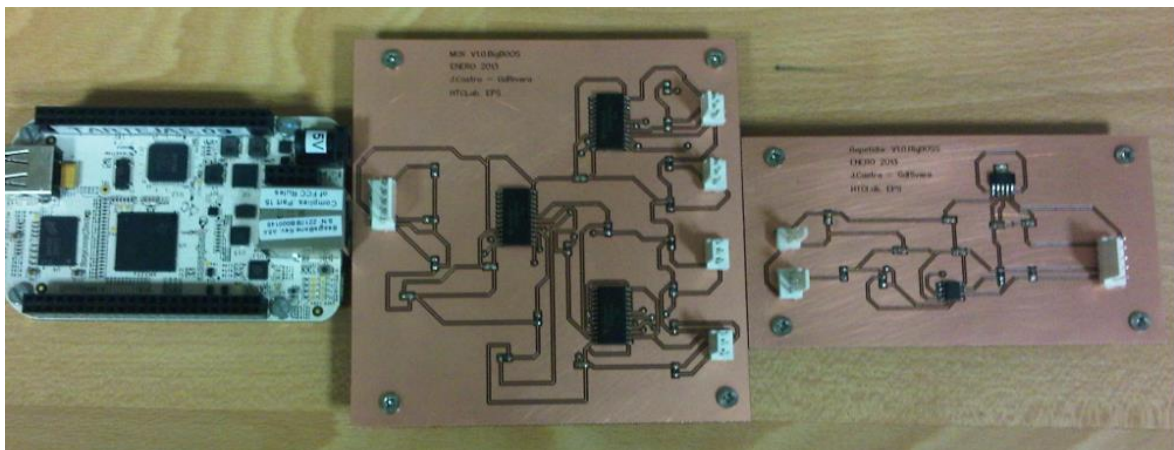


Figura 76 : Prototipo con módulo repetidor I2C

7.2.3 Futuras tecnologías

A medida que avanza las tecnologías este proyecto abre las puertas a alguna tecnología que consiga reducir el tiempo y el consumo de la solución propuesta, ya que la arquitectura propuesta se puede adoptar con facilidad.

Referencias

- [1] Bicsi, B., (2002). Network Design Basics for Cabling Professionals. City: McGraw-Hill Professional.
- [2] Zigbee -Zigbee Alliance <http://www.zigbee.org/>.
- [3] Simulación fue realizada por José Luis Masa Campos, Grupo de Radiofrecuencias, Dpto. de Tecnología Electrónica y de las Comunicaciones de la Universidad Autónoma de Madrid.
- [4] I2C – NXP Semiconductor www.nxp.com.
- [5] Bus CAN – Bosch Semiconductor www.bosch-semiconductors.de.
- [6] Ethernet - <http://www.ieee802.org/3/>.
- [7] Charles E. Spurgeon, "Ethernet: The Definitive Guide".
- [8] CyFi - Cypress Semiconductor <http://www.cypress.com>
- [9] Vincent Himpe, "Mastering the I2C Bus".
- [10] Sudeep Kumar, R.; Ganesan, P.; Poddar, G., "Bus paralleling controller with CAN interface for High Power Converter modules," Sustainable Energy Technologies (ICSET), 2010 IEEE International Conference on , vol., no., pp.1,5, 6-9 Dec. 2010.
- [11] Seong-Hoon Kim; Jeong-Seok Kang; Hong Seong Park; Daeyoung Kim; Young-Joo Kim, "UPnP-ZigBee internetworking architecture mirroring a multi-hop ZigBee network topology," Consumer Electronics, IEEE Transactions on , vol.55, no.3, pp.1286,1294, August 2009 .
- [12] Rindos, A.; Woolet, S.; Nicholson, L.; Vouk, M., "A performance evaluation of emerging Ethernet technologies: switched/high-speed/full-duplex Ethernet and Ethernet LAN emulation over ATM," Southeastcon '96. Bringing Together Education, Science and Technology., Proceedings of the IEEE , vol., no., pp.401,404, 11-14 Apr 1996 .
- [13] Righetti, X.; Thalmann, D., "Proposition of a modular I2C-based wearable architecture," MELECON 2010 - 2010 15th IEEE Mediterranean Electrotechnical Conference , vol., no., pp.802,805, 26-28 April 2010.
- [14] 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 , vol., no., pp.180,183, 21-23 Oct. 2011.
- [15] <http://beagleboard.org/bone><http://beagleboard.org/bone>
- [16] Documentación aportada por Francisco Prada.

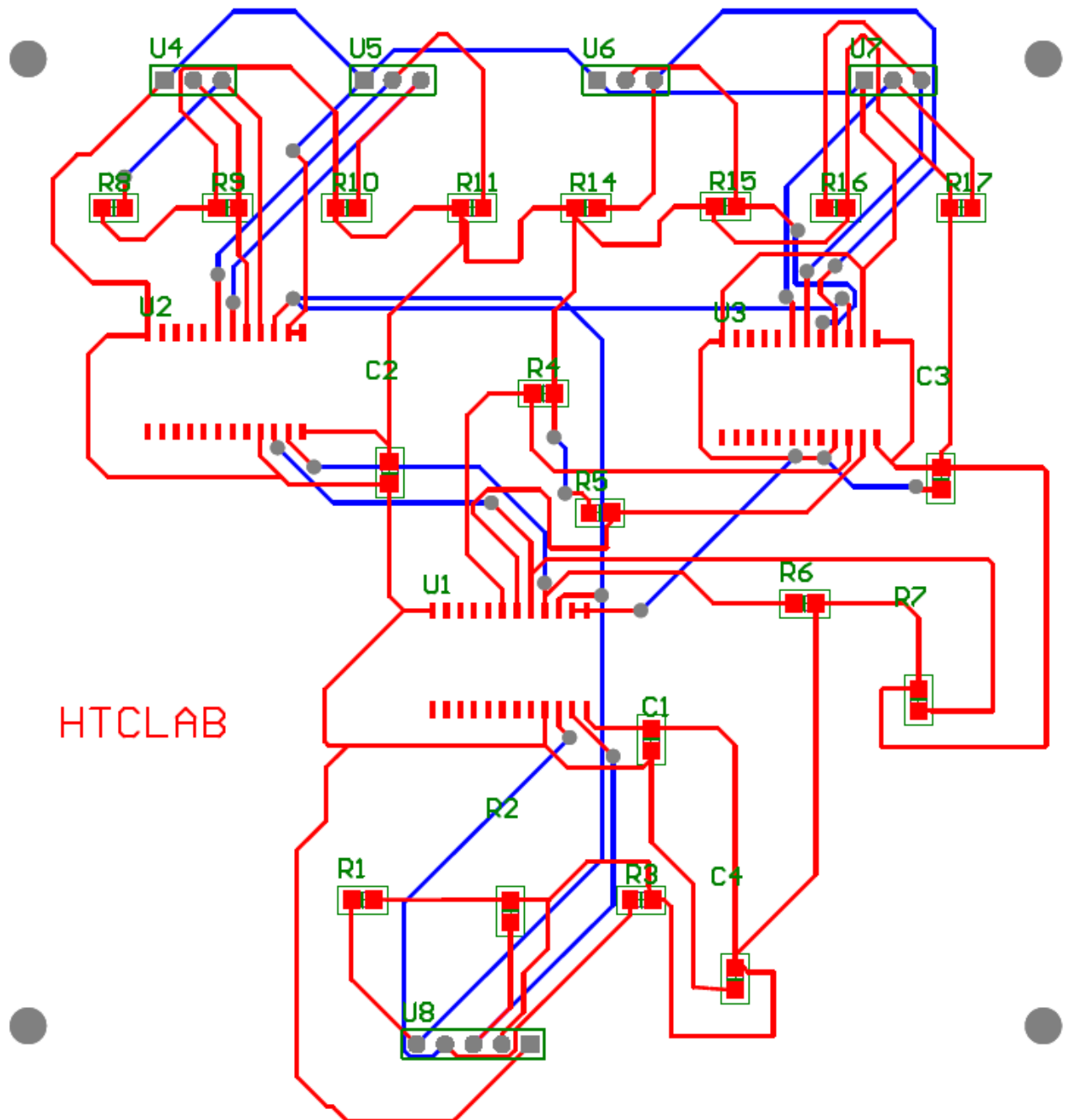
Glosario

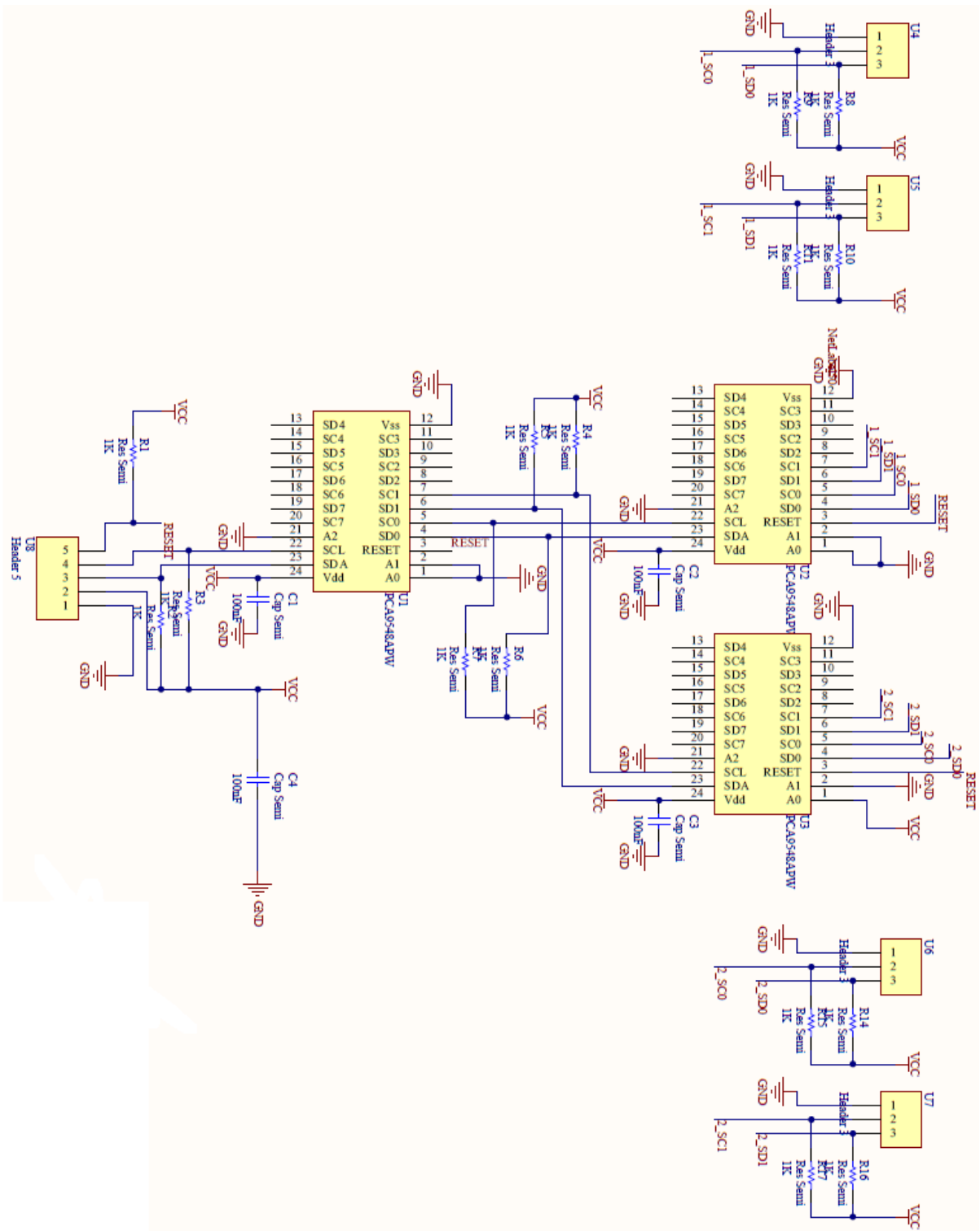
ACK	Acknowledgement
CAN	Controller Area Network
CSMA/CD	Carrier sense multiple access with collision detection
DSSS	Direct-sequence spread spectrum
GFSK	Gaussian Frequency-Shift Keying
HCTLab	Human Computer Technology Laboratory
I2C	Inter-Integrated Circuit
IDE	Integrated development environment
IFT	Instituto de Física Teórica
NOAO	National Optical Astronomy Observatory
OSI	Open System Interconnection
PCB	Printed Circuit Board
RSSI	Received Signal Strength Indication
USB	Universal Serial Bus

Anexos

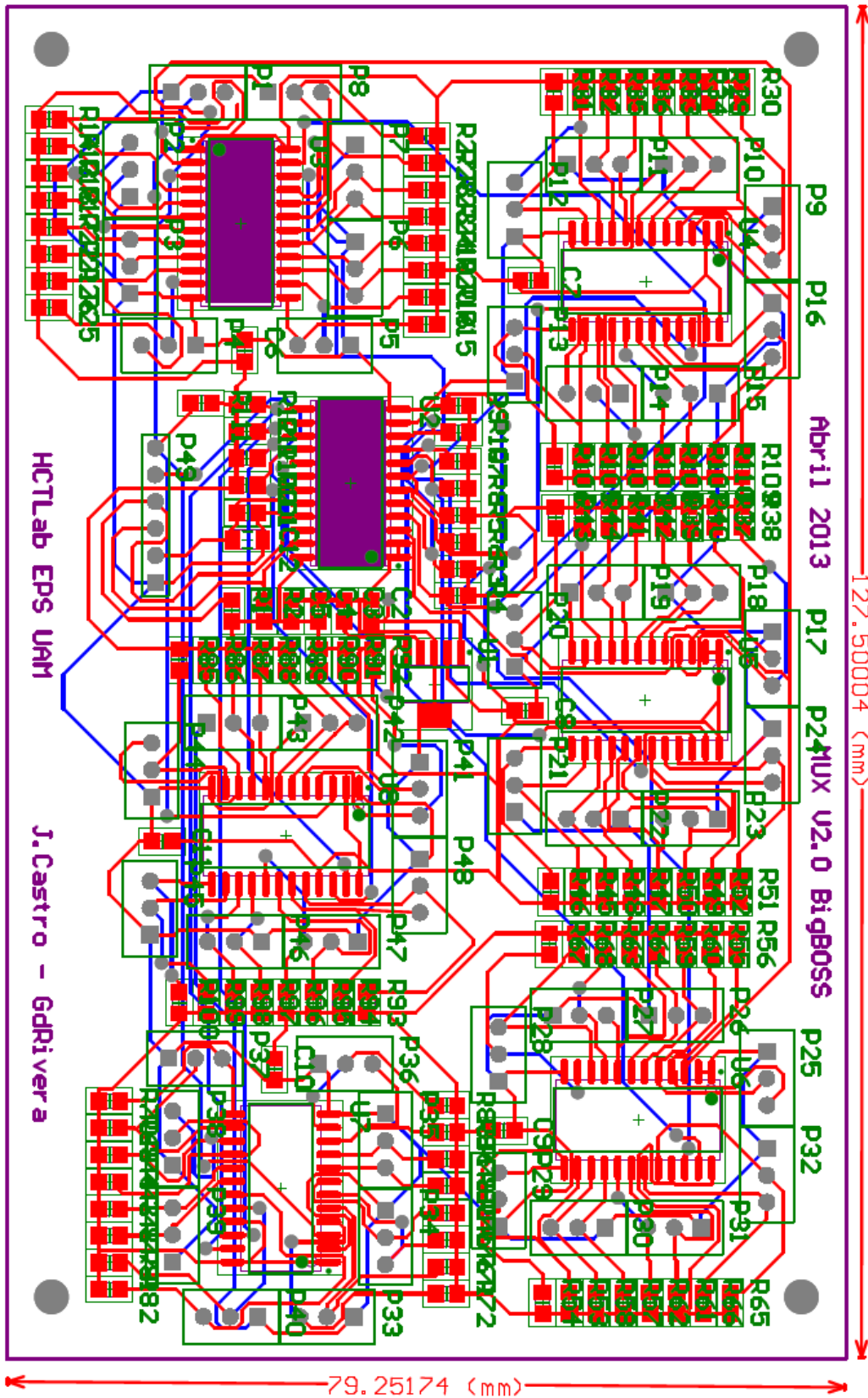
A Esquemáticos

PCB Multiplexor v1:

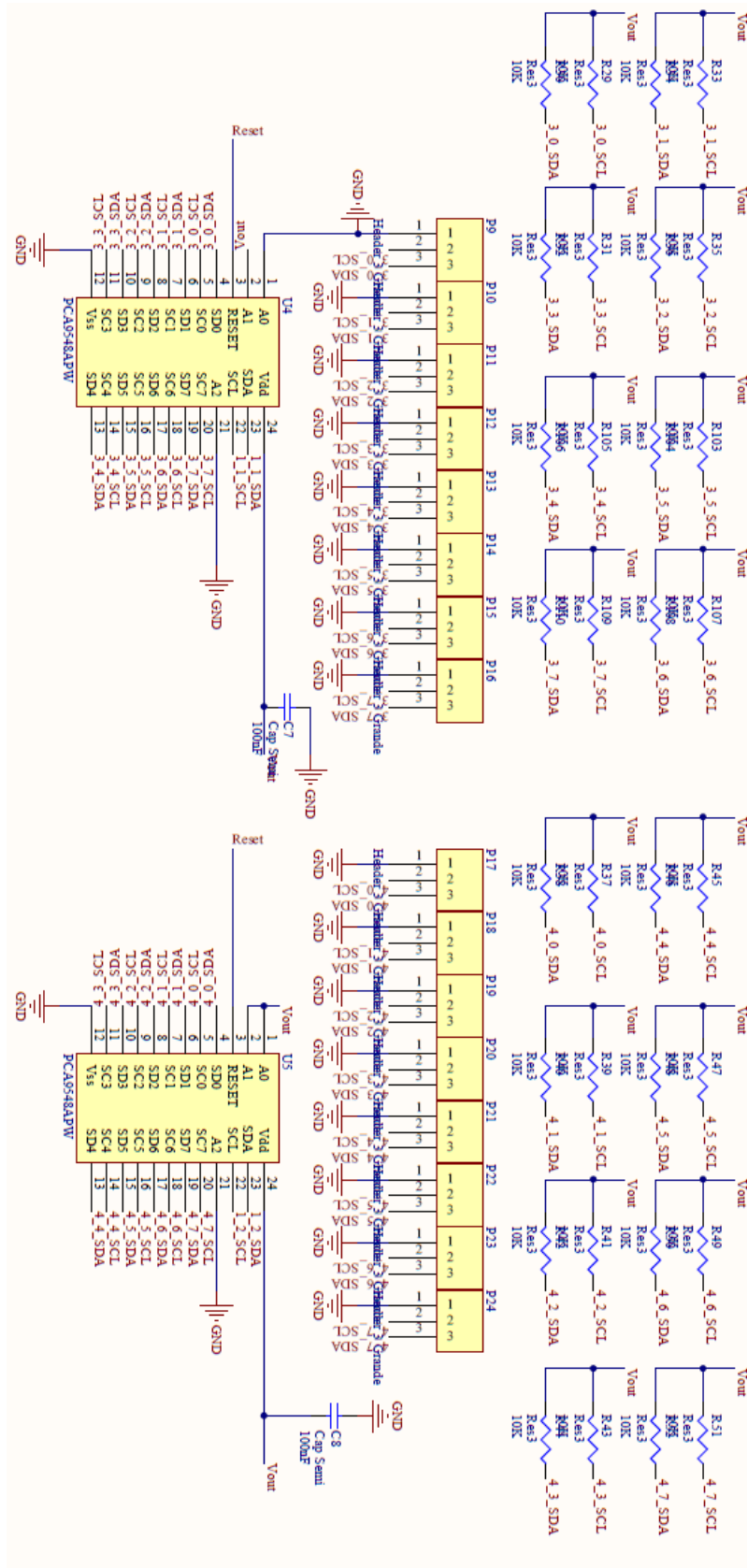




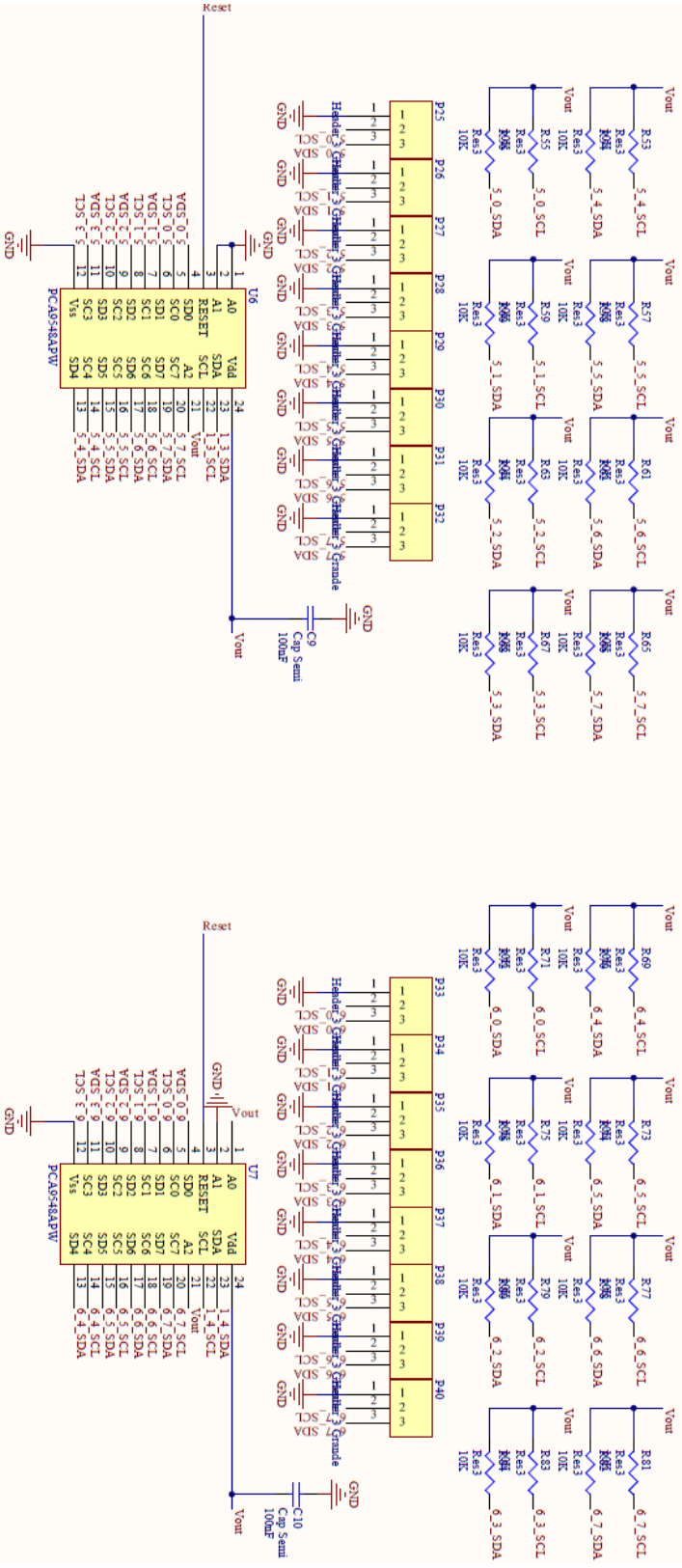
PCB Multiplexor v2:



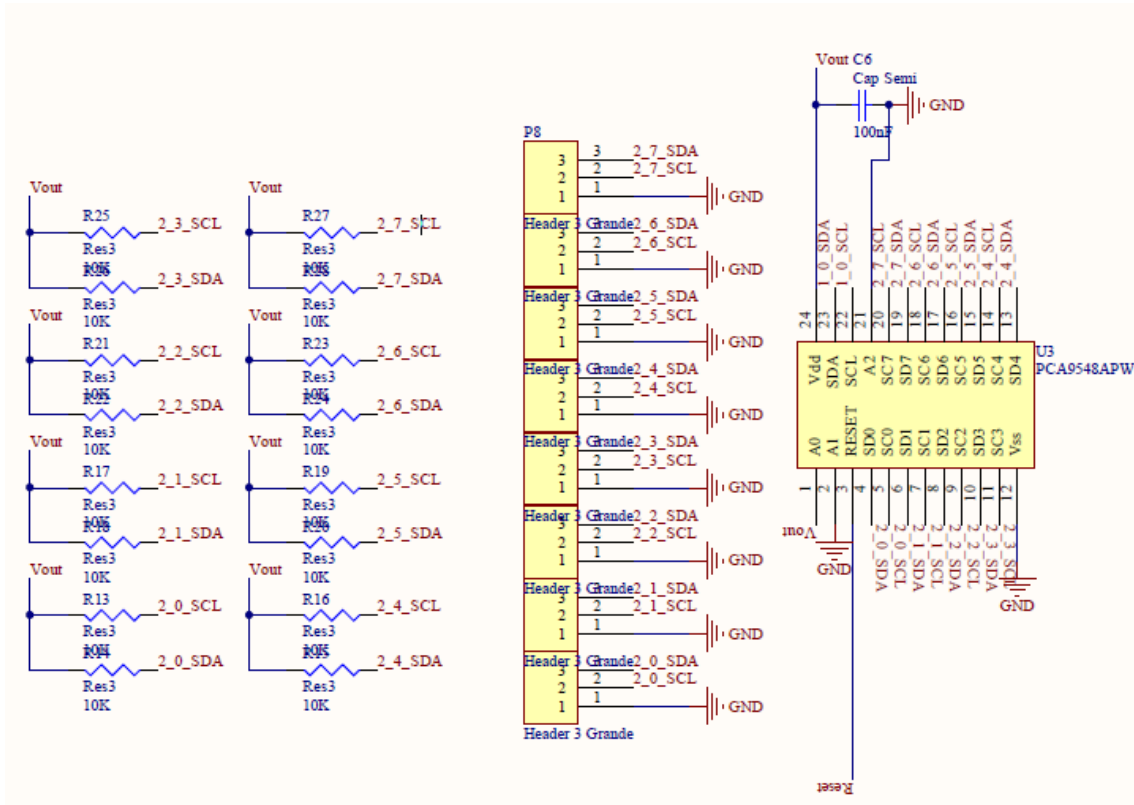
Sección 1:



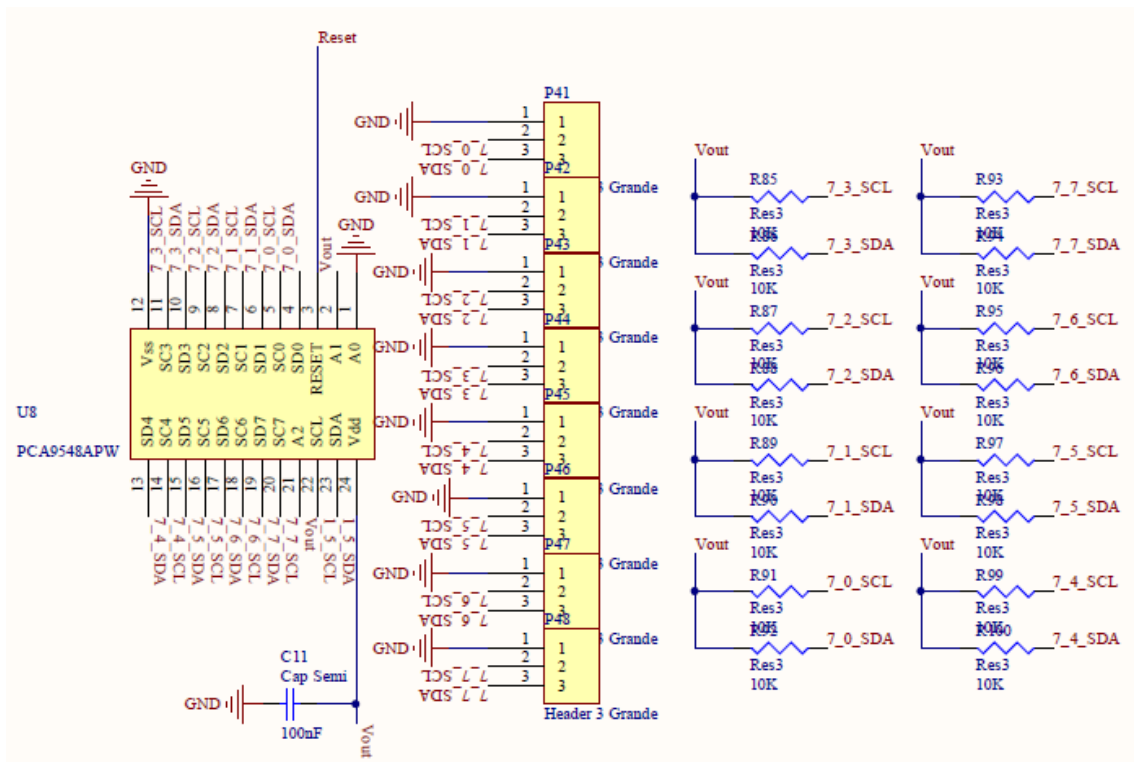
Sección 2:



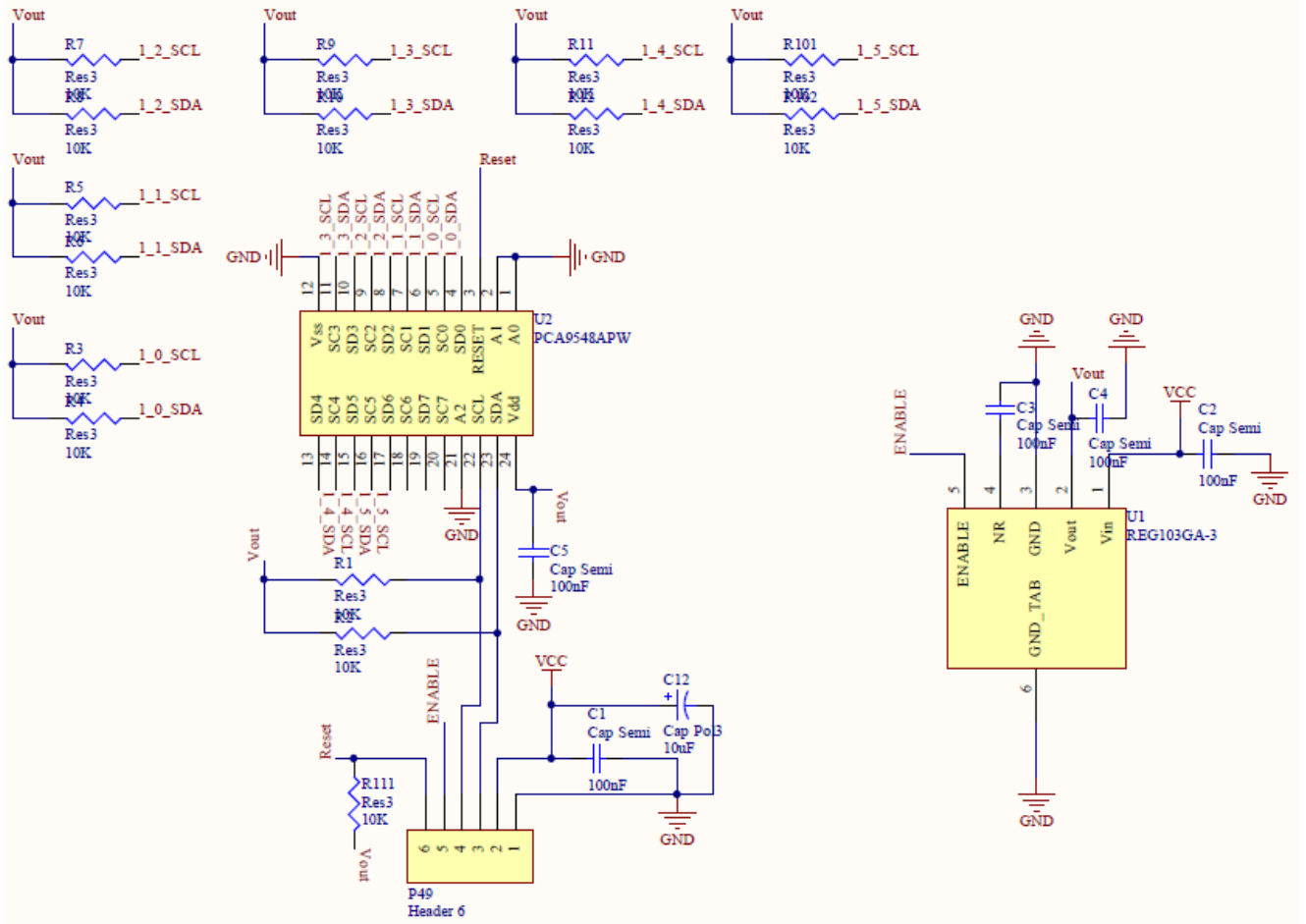
Sección 3:



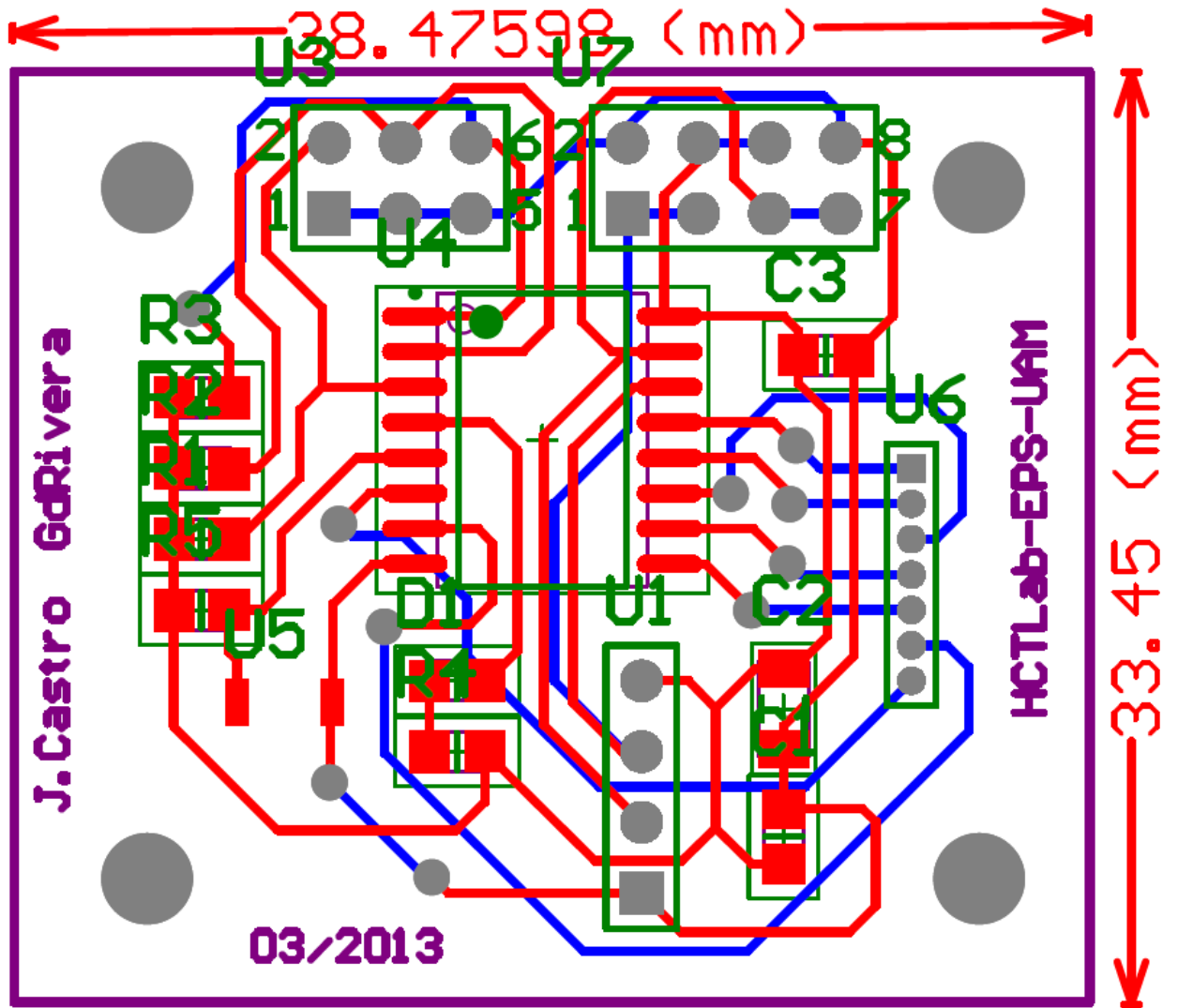
Sección 4:

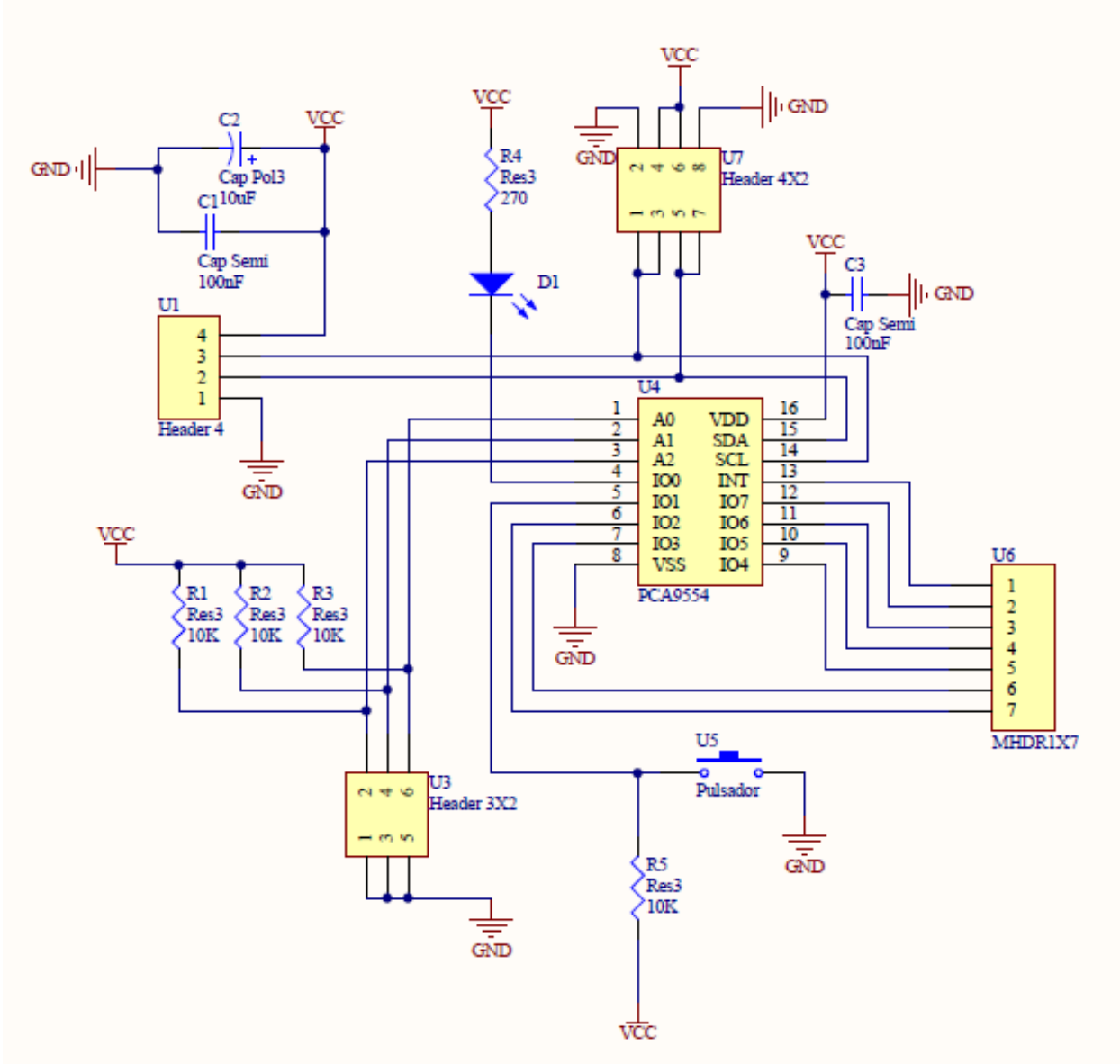


Sección 5:

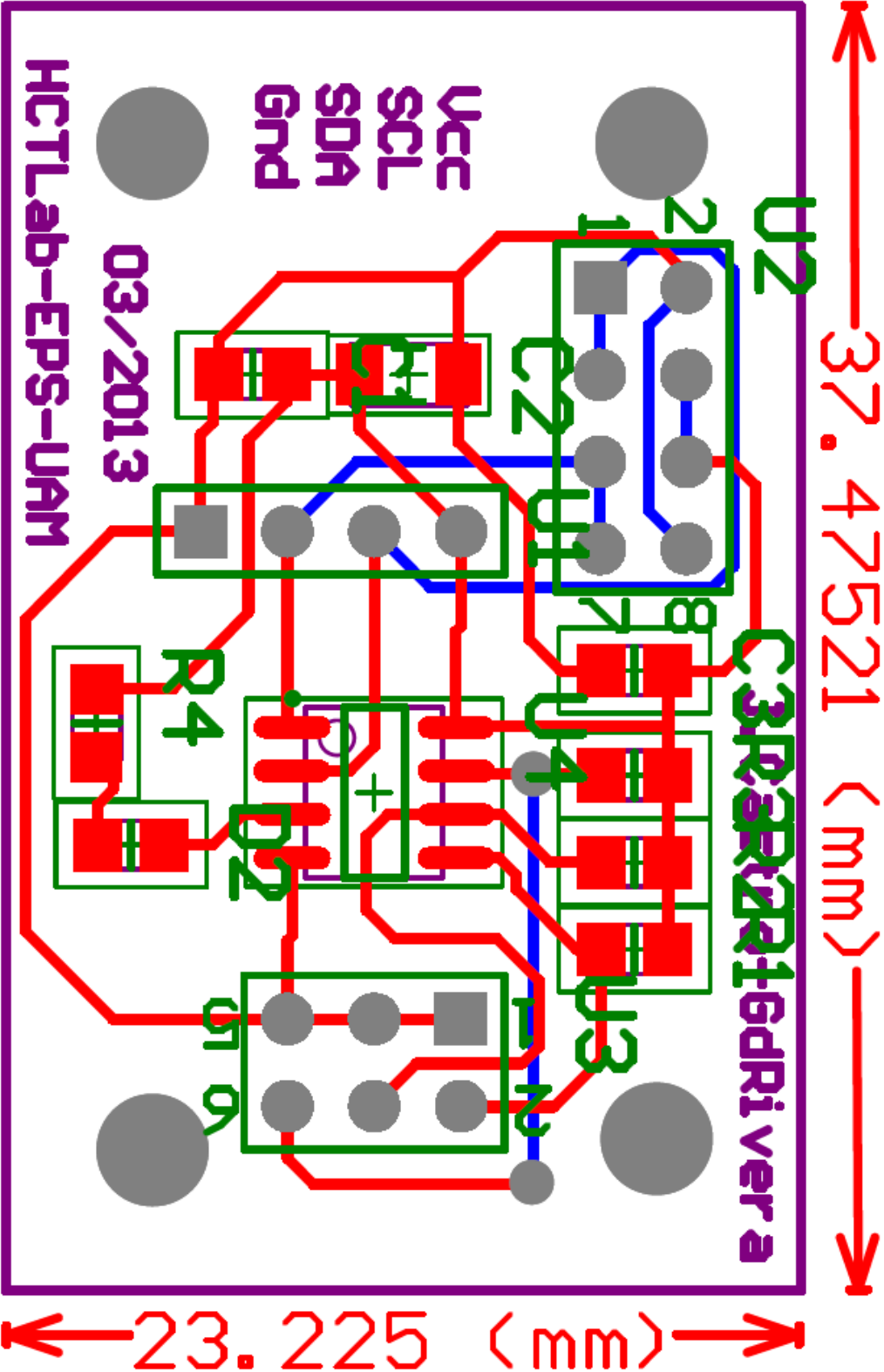


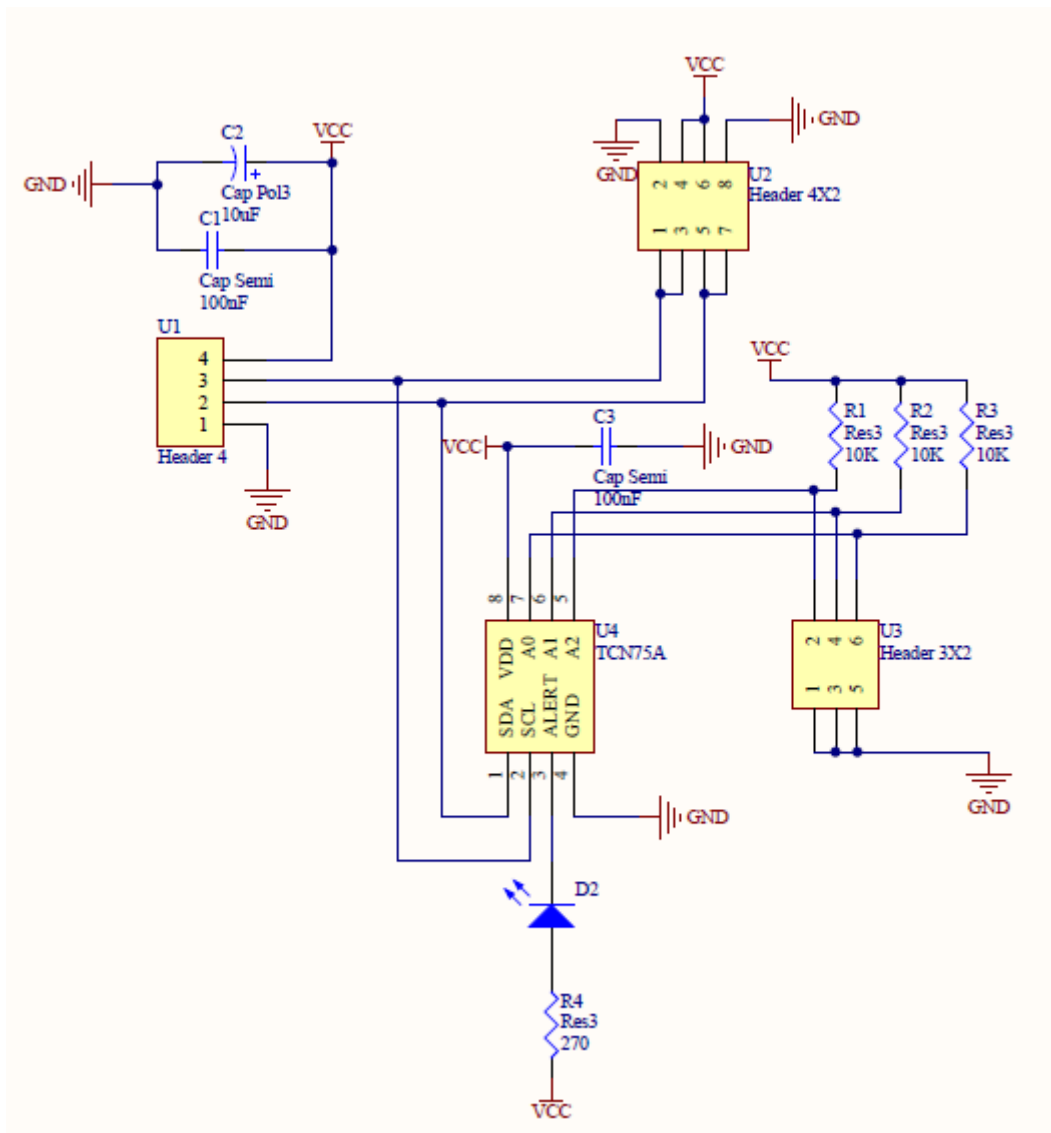
PCB Puerto entradas y salidas:

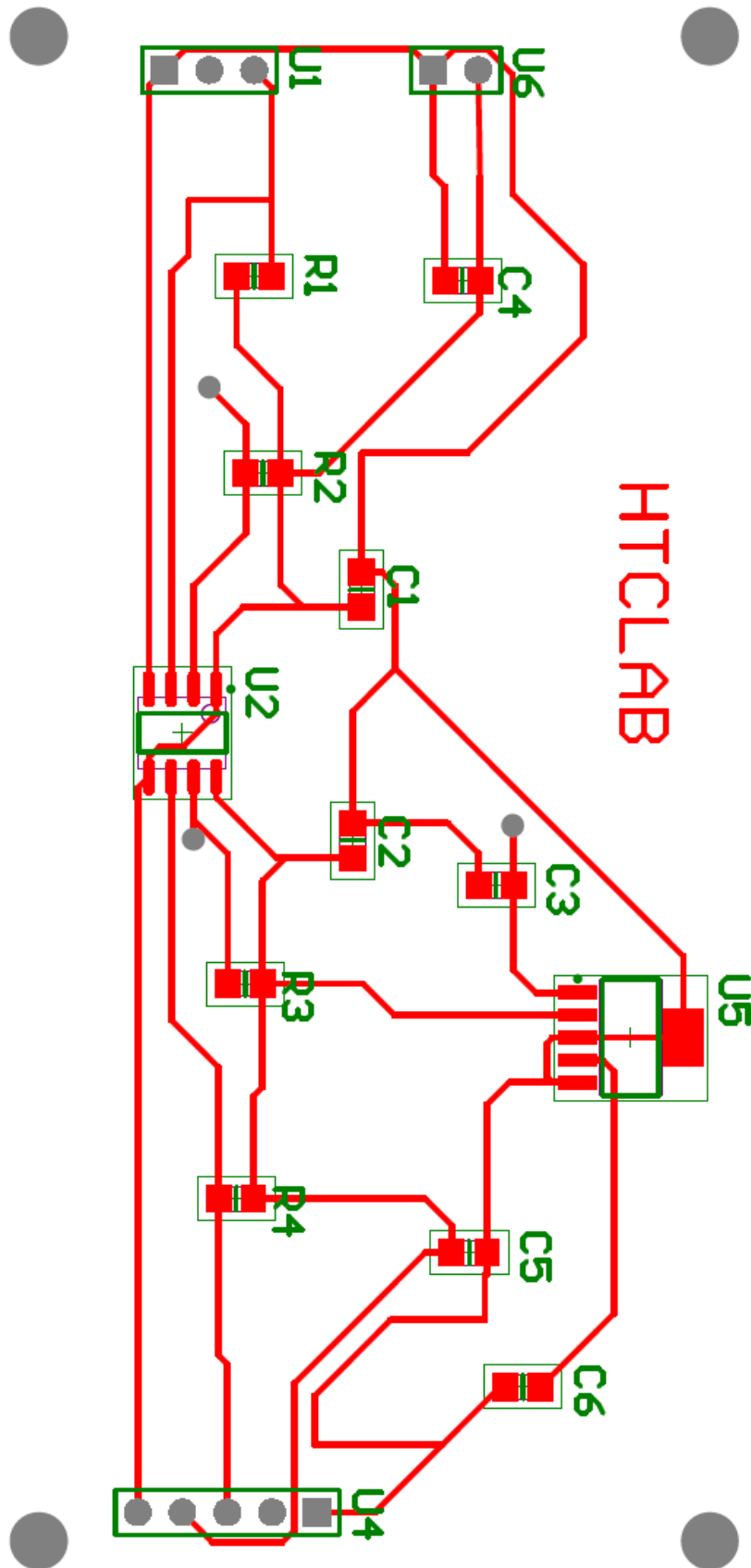


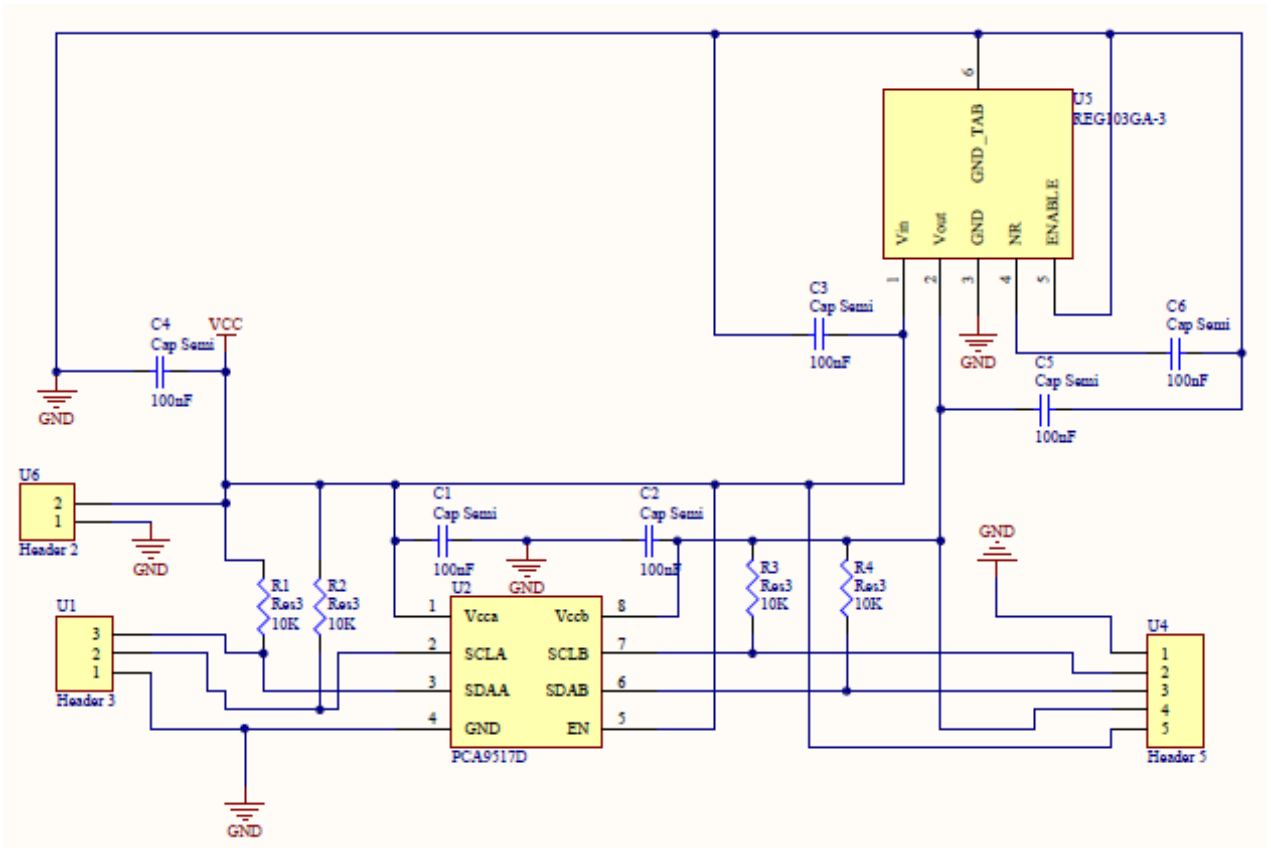


PCB Sensor de temperatura:









B Presupuesto

1) Ejecución Material

- Compra de ordenador personal (Software incluido)..... 2.000 €
- Componentes electrónicos.....300 €
- Fabricación de las PCB.....300 €
- Total de ejecución material 2.600 €

2) Gastos generales

- 16 % sobre Ejecución Material 416 €

3) Beneficio Industrial

- 6 % sobre Ejecución Material 156 €

4) Honorarios Proyecto

- 200 horas a 30 € / hora 6000 €

5) Material fungible

- Gastos de impresión 60 €
- Encuadernación 200 €

6) Subtotal del presupuesto

- Subtotal Presupuesto 9432 €

7) I.V.A. aplicable

- 21% Subtotal Presupuesto 1980,72 €

8) Total presupuesto

- **Total Presupuesto 11412,72 €**

Madrid, Junio de 2013

El Ingeniero Jefe de Proyecto

Fdo.: Jesús Castro Murillas

Ingeniero Superior de Telecomunicación

PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un “*Sistema de Comunicación para la gestión remota de un elevado número de Mini-Robots de posicionamiento en un entorno reducido*”. 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:

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 correspondiera 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.

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 con la autorización del mismo, quien delegará en él las responsabilidades que ostente.

8 Certificado de transferencia tecnológica

A QUIEN CORRESPONDA, certifico que

El Proyecto Fin de Carrera realizado por D. Jesús Castro Murillas en la Escuela Politécnica Superior de la UAM, dirigido por el profesor Guillermo González de Rivera Peces, titulado *"Sistema de comunicación para la gestión remota de un elevado número de Mini-Robots de posicionamiento en un entorno reducido"* ha servido para el diseño de la solución propuesta por el Grupo español como respuesta al problema presentado para la comunicación de los datos de posicionamiento desde una unidad central a cada uno de los 5.000 robots que contempla el desarrollo del proyecto BigBOSS. Esta solución, implementada en un primer prototipo de unos 250 elementos, ha sido probada con éxito.

Durante la ejecución del PFC no sólo se ha realizado una transferencia tecnológica, sino que de los estudios realizados se ha demostrado que otras soluciones propuestas por diferentes grupos de investigación, incluido el Lawrence Berkeley National Laboratory de la Universidad de Berkeley, no eran viables

El proyecto BigBOSS está liderado por la Universidad de Berkeley y forman parte 16 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 entender la naturaleza de la energía oscura.

Y para que así conste y a petición del interesado, expido el presente certificado en Madrid, a 24 de Mayo de 2013.

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