



6 Conclusiones

Con el fin de llevar a cabo los objetivos propuestos para el presente proyecto se han realizado las siguientes tareas y se han alcanzado los siguientes hitos:

- Estudio del micro PIC16F876.
- Estudio preliminar de la placa NYDIANET 3.0.
- Estudio del Compilador CCS PIC C Compiler.
- Estudio de las librerías de funciones de C para PIC.
- Estudio del entorno de desarrollo MPLAB IDE.
- Estudio de la posibilidad de integración entre CCS PIC C Compiler y MPLAB IDE.
- Estudio del hardware de programación MPLAB ICD 2.
- Estudio de diferentes protocolos domóticos: EIB, EHS, Konnex, etc.
- Conexión e instalación del panel de pruebas.
- Realización de un cable para programación en placa NYDIANET y su testeo.
- Estudio de los diversos modos de configuración de entradas de la placa.
- Primeras pruebas de lectura de entradas y activación de salidas de modo configurable.
- Primeras pruebas de comunicación entre placas.
- Pruebas de distinción de instrucciones y comunicación con el PC.
- Uso de interrupción externa para sincronismo con red eléctrica.
- Estudio del uso de la EEPROM del micro.
- Estudio de la posibilidad de ampliación del hardware.
- Ampliación del micro al modelo PIC18F2620.
- Inclusión de memoria EEPROM externa por I²C y su estudio.
- Estudio del micro PIC18F2620.
- Estudio de la generación de CRC con micros PIC.
- Definición del formato básico de las tramas.
- Estudio de la posibilidad de utilización de un RTOS (SO en Tiempo Real).
- Estudio del RTOS integrado con el compilador de CCS.
- Definición del mecanismo de asentimiento de tramas.
- Definición de los tipos de trama.



- Definición del mecanismo de generación de direcciones.
- Pruebas con el RTOS de CCS.
- Implementación de la detección de pulsaciones cortas y largas.
- Implementación del control de potencia mediante TRIAC's.
- Definición del formato de la configuración para almacenaje en EEPROM.
- Realización del código de la versión previa alfa 0.0 (Implementa la lectura de configuración de la EEPROM, la lectura de entradas y activación de salidas, la detección de pulsaciones largas, control de potencia de salidas, etc.).
- Depuración de la versión alfa 0.0.
- Depuración del RTOS.
- Depuración de la lectura de entradas.
- Depuración de la activación de salidas.
- Depuración de la regulación de potencia.
- Implementación de la versión alfa 1.0.
- Implementación de la interpretación de la configuración.
- Implementación de la transmisión y recepción de tramas.
- Implementación de la lógica de estados del bus.
- Implementación de la interrupción de recepción.
- Implementación de la gestión del buffer E/S.
- Implementación de la interpretación de tramas.
- Implementación del timeout de recepción.
- Generación de secuencia de transmisión.
- Desarrollo de un formato de configuración más flexible que el existente.
- Implementación del almacenaje y restauración del estado del nodo.

La última versión desarrollada es la alfa 2.0, que es la que se puede encontrar en el Anexo de la presente memoria.

Todas estas tareas realizadas han tenido como resultado la consecución con éxito de los objetivos planteados como punto de partida a la hora de abordar el presente proyecto fin de carrera. Partiendo de un hardware existente se ha conseguido implementar una arquitectura flexible y robusta a partir de la cual sea posible generar nuevos proyectos de desarrollo de aplicaciones basadas en dicha arquitectura.



La simplicidad y sencillez de instalación del hardware empleado han permitido dotar al sistema de cualidades tan importantes como el bajo coste (menos de 50€ por unidad local) y la flexibilidad de operación.

El sistema se ha diseñado de modo que sea independiente del número de elementos que componen la red domótica. Del mismo modo, es capaz de funcionar tanto si existe un nodo central como si no, aunque algunas capacidades se ven inevitablemente mermadas en este caso.

La programación de los elementos de la red es flexible y permite dotar a cada nodo de una configuración por defecto que se puede ajustar de manera sencilla gracias al formato común definido para la configuración.

Todo esto hace que el sistema desarrollado no solo sea funcionalmente independiente, si no que de igual modo permite su control de manera remota y centralizada así como la inclusión de nuevas funcionalidades, aspecto este último muy interesante a la hora de definir nuevas líneas de avance para la realización de futuros proyectos fin de carrera.

A título personal, el autor del presente proyecto ha tenido la oportunidad de enfrentarse al desarrollo de una arquitectura funcional completa desde cero, partiendo de un hardware *a priori* desconocido con el que ha tenido que familiarizarse y usando herramientas específicas que hasta el momento desconocía.

El conocimiento adquirido sobre elementos hardware como los microcontroladores de la familia PIC o los dispositivos I²C, así como de herramientas de desarrollo y programación, suponen sin lugar a dudas un enriquecimiento personal y de conocimientos cuya importancia y utilidad es más que patente, ya que, a fecha de hoy, está determinando de manera muy clara la actividad profesional que el autor ha comenzado a desarrollar recientemente.