



INSTITUTO DE INVESTIGACIÓN
TECNOLÓGICA

PROYECTO DE FIN DE GRADO

IMPLEMENTACIÓN DE UN CONTROL PREDICTIVO DE TEMPERATURA EN UNA VIVIENDA

Alumno:

GARCÍA AGUILAR, JAVIER

Directores

ZAMORA MACHO, JUAN LUIS

MUÑOZ FRÍAS, JOSÉ DANIEL

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Javier García Aguilar, como estudiante de la UNIVERSIDAD PONTIFICIA COMILLAS (ICAI, E.T.S.I. de Ingeniera Industrial)

DECLARA que es el titular de los derechos de propiedad intelectual de la obra: “Implementación de un control predictivo de temperatura en una vivienda”, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual como titular único.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia *Creative Commons*.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que

podieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 20 de Julio de 2016

ACEPTA



Fdo.



Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

***“IMPLEMENTACIÓN DE UN CONTROL PREDICTIVO DE TEMPERATURA
EN UNA VIVIENDA”***

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2016 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

PROYECTO REALIZADO POR EL ALUMNO

Javier García Aguilar

Fdo.:



Fecha: 16 / 07 / 2016

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Juan Luis Zamora Macho

Fdo.:



Fecha: 16 / 07 / 2016

José Daniel Muñoz Frías

Fdo.:

Fecha: 16 / 07 / 2016

Vº Bº DEL COORDINADOR DE PROYECTOS

Álvaro Sánchez Miralles

Fdo.:

Fecha: / /

IMPLEMENTACIÓN DE UN CONTROL PREDICTIVO DE TEMPERATURA EN UNA VIVIENDA

ALUMNO: GARCÍA AGUILAR, JAVIER

DIRECTOR: ZAMORA MACHO, JUAN LUIS

CODIRECTOR: MUÑOZ FRÍAS, JOSÉ DANIEL

RESUMEN DEL PROYECTO

En un entorno domótico, uno de los aspectos más importantes es el confort. Para conseguir un buen confort, el correcto control de la temperatura es vital. Actualmente, los controles de temperatura más utilizados en la industria de climatización son los controles todo/nada, que presentan una potencia máxima, o nula, en función de la temperatura respecto a un valor de referencia, con histéresis. Otros controles que empiezan a utilizarse son los PID.

Los controles predictivos, frente a los controles PID, presenta la ventaja de poder prever situaciones futuras y actuar en consecuencia con antelación, mejorando el consumo frente a los controles PID [1]. En contrapartida, requieren de una potencia de cálculo substancialmente mayor.

Objetivos

Este proyecto trata de implementar un control predictivo de temperatura en una vivienda de 4 habitaciones.

Estructura del sistema.

Este proyecto se engloba en un proyecto más grande para domotizar una vivienda que, por su extensión, fue dividido en dos. En la Figura 1 se observa la estructura del proyecto completo.

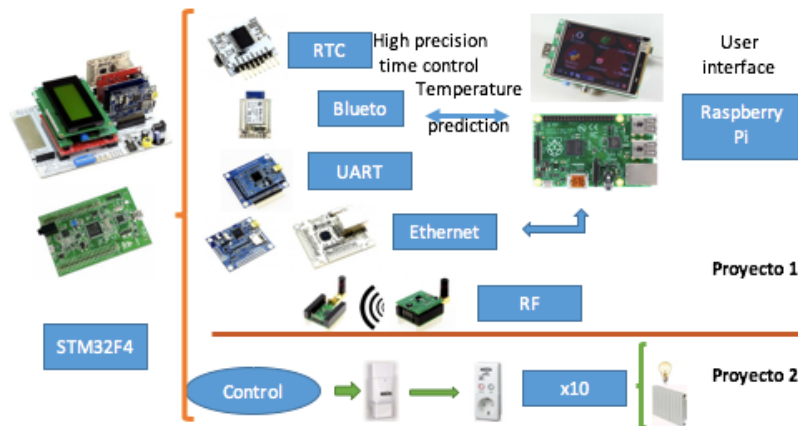


Figura 1. Estructura del proyecto.

El elemento común entre ambos es el microcontrolador STM32F407VG. Este microcontrolador presenta una gran capacidad de cálculo y cuenta con un set de bloques propio (Waijung) que permite programarlo fácilmente desde Simulink, además de presentar una amplia variedad de periféricos. Estas características hacen del STM32 idóneo para el proyecto global. En la Tabla I se muestra el reparto efectuado entre ambas partes del proyecto.

Domotización de una vivienda de 4 habitaciones	
Proyecto 1	Proyecto 2
Sistema de comunicación sensores Servidor web Interfaz usuario Obtención predicción meteorológica	Implementación protocolo x10 Diseño del control predictivo Hardware in the loop (HIL)

Tabla I. Reparto de tareas del proyecto.

Los apartados que siguen a continuación hacen referencia al Proyecto 2.

Sistema

En la Figura 2 puede observarse un esquema del sistema que se trata en este proyecto. A partir de las lecturas de las temperaturas por los sensores inalámbricos, la temperatura exterior predicha y las referencias impuestas, el control predictivo calcula los mandos (potencia) a aplicar. A continuación se efectúa una modulación por ancho de pulso (PWM) de éstos, por las limitaciones de los actuadores (radiadores eléctricos). Los actuadores son accionados por medio de comunicación x10.

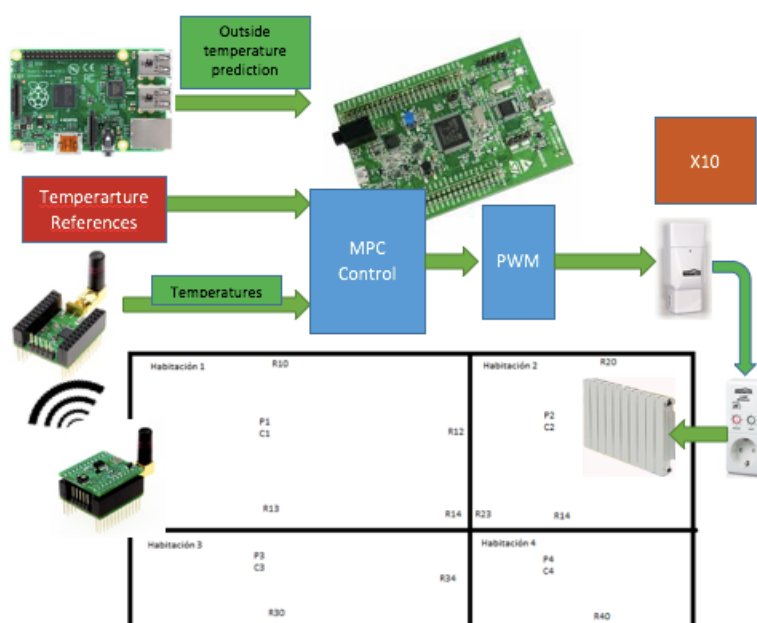


Figura 2. Esquema del Sistema

Tareas realizadas.

La primera tarea que se realizó en este proyecto fue gestionar el protocolo de comunicaciones x10 desde el STM32F407VG. Para ello, se diseñó un diagrama de bloques en Simulink, usando tanto bloques propios de Simulink y bloques del Waijung, y una máquina de estados para gestionar la lógica del protocolo. En la Figura 3 puede observarse la conexión entre ambos bloques.

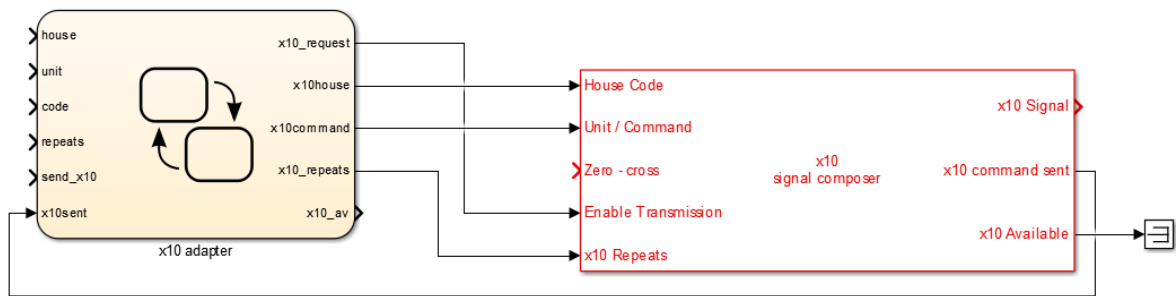


Figura 3. Diagrama de gestión de la comunicación x10.

La composición de la trama x10 se realiza en dos pasos. El primero es obtener la señal lógica, que se compone a partir de los datos enviados desde la máquina de estados. Por otro lado se compone una señal auxiliar, sincronizada con los pasos por cero de la senoidal de red (flancos en la señal digital 'Zero - cross'). La trama x10 a enviar consiste en el producto lógico de esas dos señales.

Por otro lado, se lleva la cuenta de las veces que se ha enviado la trama x10, información que necesita la máquina de estados para el correcto funcionamiento del protocolo. Cada mensaje tiene que enviarse varias veces para asegurar una comunicación correcta y salvar las limitaciones que presenta la red.

Por último, la máquina de estados se encarga de gestionar las distintas partes de un mismo mensaje y de señalar si hay una comunicación en curso o si, por otro lado, está disponible para una nueva comunicación.

Para la gestión de los cuatro mandos (radiadores), se diseñó una segunda máquina de estados, encargada de llevar cuenta de los estados en que se encuentran los actuadores, compararlos con los mandos de salida del control, y actuar en consecuencia. Su apariencia puede observarse en la Figura 4.

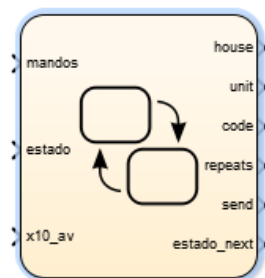


Figura 4. Máquina de estados de gestión de los mandos.

En la Figura 5 se muestra la captura con el osciloscopio de una trama de comunicación x10 (CH1), además de la señal ZC (CH2).

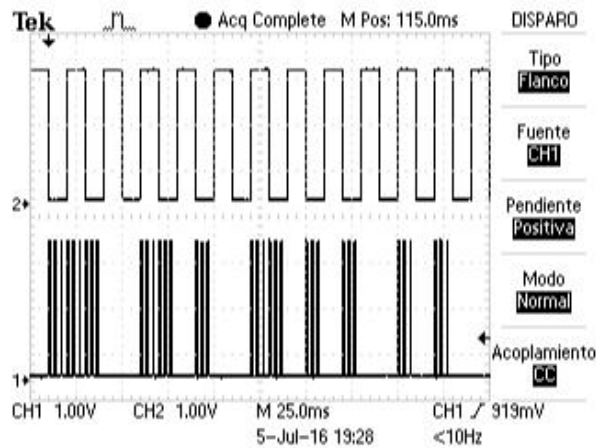


Figura 5. Trama de comunicación x10.

Fuente: Osciloscopio

La segunda tarea realizada fue el diseño de un control predictivo para el modelo de la planta disponible. Para ello se hizo uso de la herramienta 'MPC Tool' de Matlab [2], que permite diseñar controles predictivos basándose en un modelo matemático de la planta modificando los pesos de las variables a controlar y de los mandos. La comprobación del control se efectuó mediante simulación, haciendo uso del modelo matemático de la planta, y del control propiamente dicho.

Otra tarea realizada constituyó en determinar si el microcontrolador STM32F407VG era capaz de ejecutar el control correctamente. Para ese propósito se empleó una estrategia basada en el "Hardware In the Loop" (HIL), que consiste en simular en el ordenador el modelo de la planta en tiempo real, a la vez que en el STM32 se ejecuta el control. Por puerto serie se envían los valores de las temperaturas simuladas al microcontrolador cada cierto intervalo de tiempo. Del mismo modo, los mandos son enviados al ordenador para simular el comportamiento de la planta. En la Figura 6 puede apreciarse un esquema del funcionamiento del HIL.

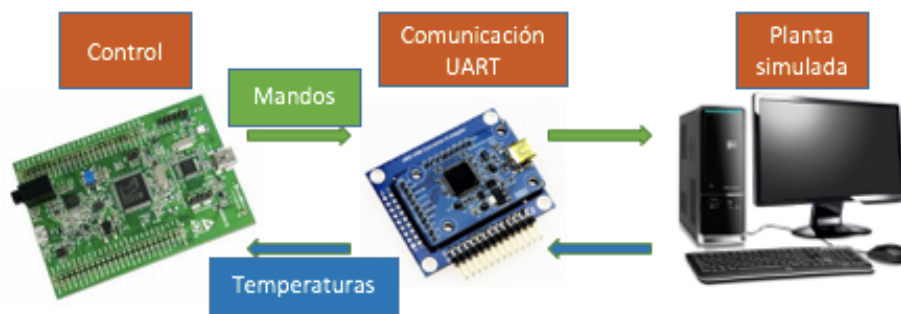


Figura 6. Sistema HIL

En la Figura 7 puede observarse la evolución de la temperatura simulada, sin modulación por ancho de pulso (PWM) en el mando, con él, y con los mandos del HIL, para una de las habitaciones.

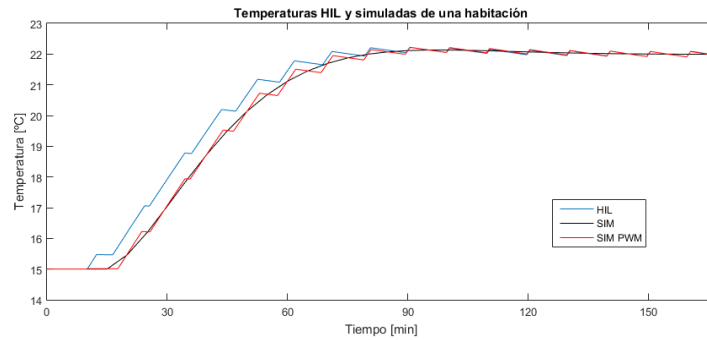


Figura 7. Temperaturas de una habitación, simuladas y obtenidas por HIL

Por último, se dejó preparado un sistema cuyo propósito es la obtención de parámetros más fiables de la planta en concreto. Para ello será necesario someter a un ensayo la vivienda en cuestión. El ensayo consta simplemente de una "Pseudo Random Binary Signal" (PRBS) por habitación, empleada como referencia de temperatura; y un control proporcional. Tanto los mandos como las temperaturas obtenidas a lo largo del ensayo son usados en un script de optimización para hallar los datos de resistencias y capacidades térmicas de la planta.

Resultados y conclusiones

La implementación del protocolo x10 en el STM32F407VG ha sido exitosa, siendo capaz de encender y apagar múltiples dispositivos, y de regular la intensidad luminica de las luces.

Por otro lado, el control predictivo diseñado cumple con las expectativas, presentando robustez y tiempos de alcance y establecimiento adecuados, teniendo en cuenta las constantes de tiempo del sistema. El HIL también funciona correctamente, se ha conseguido una semejanza con la simulación lo suficientemente buena, teniendo en cuenta las limitaciones del método.

Por último, el ensayo del script de optimización para la obtención de los parámetros de la planta fue exitoso, obteniendo unos parámetros muy parecidos a los originales, y una respuesta prácticamente idéntica para tras la aplicación de los mismos mandos.

Además, se efectuaron pruebas de integración junto al trabajo realizado en el Proyecto 1. Para ello, el primer ensayo fue comprobar si se es capaz de gestionar la luminaria desde el interfaz del servidor web, estableciendo una comunicación x10 correcta. El segundo ensayo consistió en establecer las referencias de temperatura de las habitaciones directamente desde el servidor web. Los resultados de ambos ensayos fueron satisfactorios.

Futuros desarrollos

Lo próximo que habría de hacerse es ensayar la planta para obtener unos parámetros precisos de la planta, para poder diseñar un control más preciso para ella, y dejar finalmente el sistema en funcionamiento.

Por otro lado, podría dotarse a la vivienda de un sistema de aire acondicionado para ser capaz de enfriarla en las épocas calurosas del año.

Bibliografía

- [1] P.-D. Morosan, R. Bourdais, D. Dumur and J. Buisson, "Distributed model predictive control for building temperature regulation," American Control Conference, Baltimore, 2010.
- [2] A. Bemporad, M. Morari and N. L. Ricker, "Model Predictive Control Toolbox. Getting Started Guide".

IMPLEMENTATION OF A TEMPERATURE MODEL PREDICTIVE CONTROL IN A HOUSE

ALUMNO: GARCÍA AGUILAR, JAVIER

DIRECTOR: ZAMORA MACHO, JUAN LUIS

CODIRECTOR: MUÑOZ FRÍAS, JOSÉ DANIEL

ABSTRACT

In an automated environment, comfort is one of the most important aspects. For achieving a good level of comfort, a correct temperature control is vital. Currently, the most used temperature control in home automation is the on/off with hysteresis control. PID controls are also starting to be used.

Model Predictive Controls (MPC), comparing to PID, has the advantage of being able to predict future situations and act consequently. They also improve power consumption [1]. In return, they need higher calculus speed and power.

Objectives

The aim of this project is to implement a temperature MPC in a 4 room building.

System structure

This project is included in a greater one whose aim is to automate a 4 room home. Attending to its extension, this project was divided in 2 smaller ones. In Figure 1 the global project structure is shown.

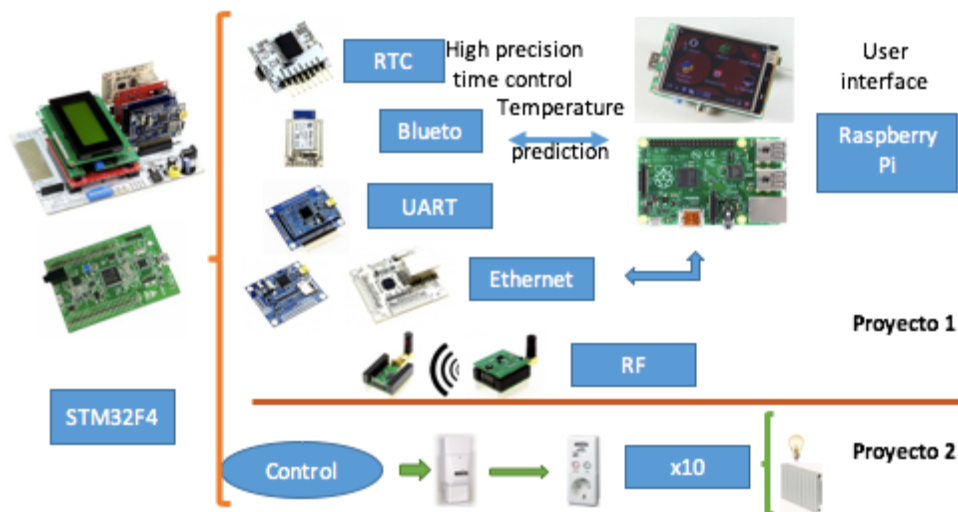


Figure 1. Project structure

The common element between both projects is the STM32F407VG microcontroller. This microcontroller presents a great calculus capacity, and counts with an own blockset (Waijung) which allows to program it easily from Simulink. Moreover, it has a huge amount of peripherals, with their own Waijung blocks. These characteristics make the STM32 perfect for its use in this project. The Table I shows the work that has been done in each part of the project.

4 room automated home	
Project 1	Project 2
Sensor communication system Web Server User Interface Meteorological prediction	x10 protocol implementation Predictive control design Hardware in the loop (HIL)

Table I. Work done in each project

The following is referred to work done in Project 2.

System

In the Figure 2 the schematic that has been used in this project is shown. The MPC controller calculates the manipulated variables (heat power) to apply from the temperature in each room (measured with wireless sensors), the outside temperature predicted and the temperature references. Then, a Pulse Width Modulation (PWM) is applied to the MV due to the limitations of the actuators (electric heaters). The actuators are controlled by x10 communication.

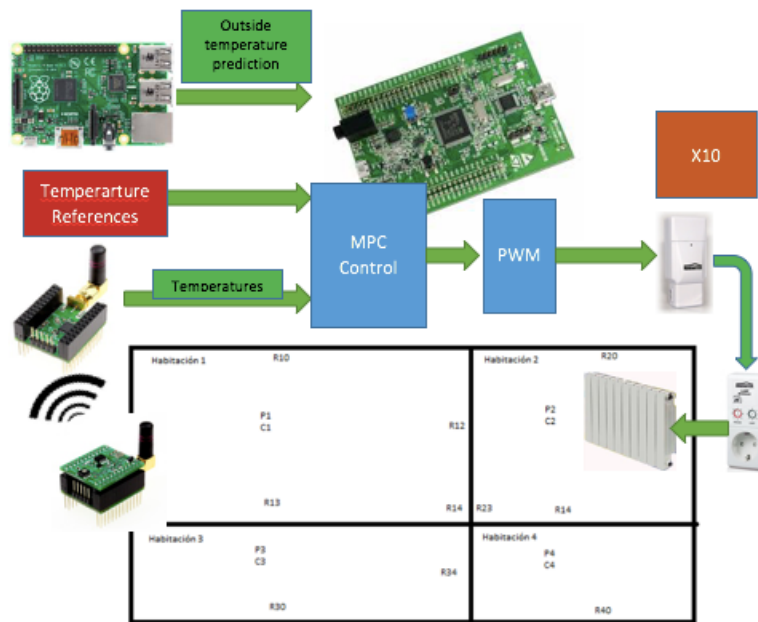


Figure 2. System schematic

Tasks done.

First task done in this project was to achieve a correct x10 communication from the STM32F407VG microcontroller. To achieve this goal, a Simulink block diagram subsystem was designed, combining both Simulink and Waijung blocks. A State Machine to manage the x10 protocol was also designed. In Figure 3 the proper connection between both systems is shown.

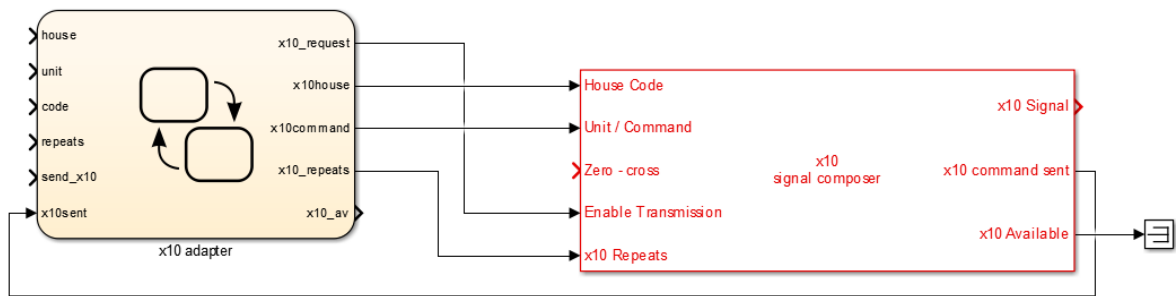


Figure 3. x10 communication manager block diagram

The composition of the x10 message is done in two steps. First is obtain the logic signal, composed from the data sent by the State Machine. Second, an auxiliary signal is composed. This signal is synchronized with the Zero-Cross (ZC) of the electrical network. The final x10 signal is obtained from the logical product of this two signals.

On the other hand, the subsystem counts the number of times the message has been sent, which is necessary for the correct operation of the protocol. Each message has to be sent at least three times to assure the communication and overcome the limitations of the electrical power.

Finally, the state machine manages the different parts of the same message, and it is the one in charge to indicate if the system is available to send another message or not.

For managing the four manipulated variables, a second state machine was designed. This one is the responsible of checking in which state is each heater, and compare it with its manipulated variable, and act consequently. Its appearance may be observed in Figure 4.

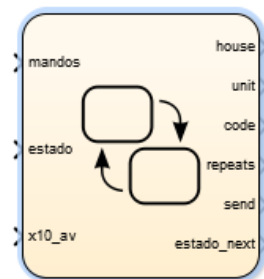


Figure 4. State Machine. Manipulated variables x10 manager

In Figure 5 an oscilloscope sample of a x10 communication (CH1) is shown, as well as the ZC signal (CH2).

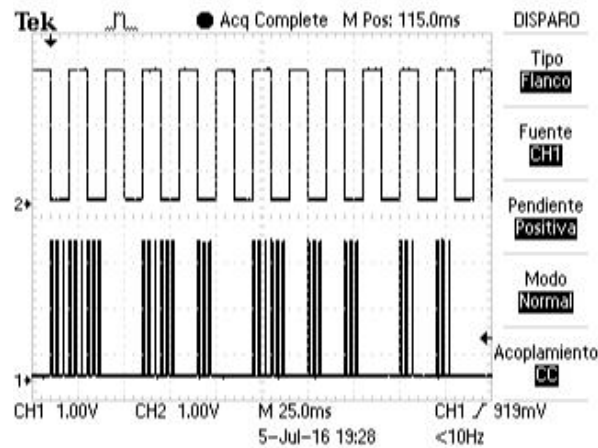


Figure 5. x10 communication example in the oscilloscope

The second task done was to design a MPC controller for the mathematical plant available. For that purpose, the Matlab 'MPC Tool' was used [2]. This tool allows to design a predictive control by modifying the weights of the controlled variables and the manipulated variables. The prove of the control was done by simulating its behaviour in Simulink.

Other task was to determine if the STM32F407VG microcontroller is able to run the MPC control correctly. With that purpose a hardware in the loop (HIL) strategy was used. The HIL consists in simulate in a computer the plant in real time, while in the STM32 is running the controller. The communication between both is done by UART. In Figure 6 can be appreciated an schematic of the HIL operation.



Figure 6. HIL system

In the Figure 7 may be observed the evolution of the simulated and HIL temperatures for oe of the rooms. Both PWM and non_PWM simulated control signals are shown to their comparison with the HIL one.

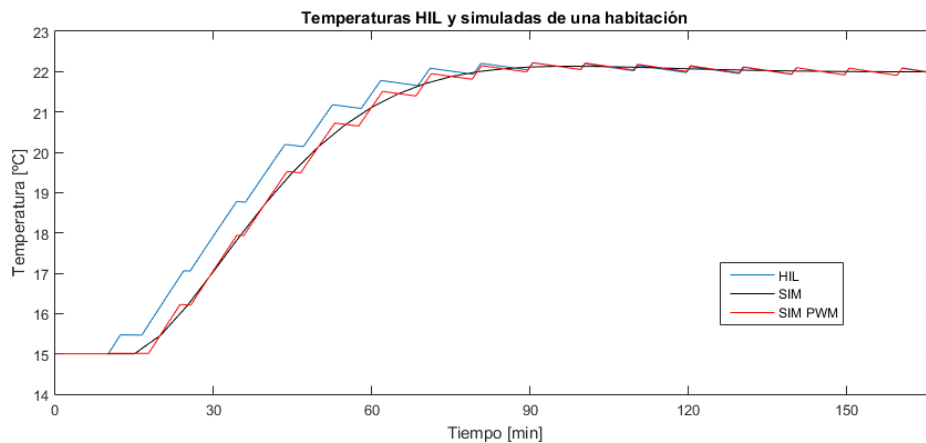


Figure 7. Simulated and HIL temperatures for a room

Lastly, there was designed a system whose purpose is to obtain more realistic parameters of the plant in order to design better controllers for the final implementation in the automated house. With this purpose, the real plat shall pass a test, which consists in applying a “Pseudo Random Digital Signal” (PRBS) as references for each room, and apply a proportional controller. Both temperature and manipulated variables must be saved in order to be processed in an optimization program which will be the responsible to obtain the thermic resistances and capacitances of the plant.

Results and conclusions.

Implementation of the x10 protocol in the STM32F407VG has been successful. It is able to turn on and off multiple devices, even to bright and dim lights.

On the other hand, the MPC control that has been designed has proved to work properly, presenting robustness and time constants that are appropriated to this application. The HIL has also work properly, and has proved that the microcontroller is able to run the controller.

Last, the optimization script that is intended to be used to obtain the correct parameters of the plant has been proved to work properly, obtaining parameters very similar to the original ones, and responses nearly identical to the application of the same manipulated variables.

In addition, integration tests have been done with the Project 1. For that, the first test was to prove that the lights can be controlled from the web server interface, establishing a correct x10 communication. The second test consisted in apply the temperature references directly from the web serves. The results of both tests were successful.

Future developments

First is to obtain the correct parameters of the plant, in order to design an accurate MPC and finally set the system to work.

Moreover, the plant could be given an air condition system in order to be able to cool it in summer time.

Bibliography

- [1] P.-D. Morosan, R. Bourdais, D. Dumur and J. Buisson, "Distributed model predictive control for building temperature regulation," American Control Conference, Baltimore, 2010.
- [2] A. Bemporad, M. Morari and N. L. Ricker, "Model Predictive Control Toolbox. Getting Started Guide".



INSTITUTO DE INVESTIGACIÓN
TECNOLÓGICA

PROYECTO DE FIN DE GRADO

MEMORIA

IMPLEMENTACIÓN DE UN CONTROL PREDICTIVO DE TEMPERATURA EN UNA VIVIENDA

Alumno:

GARCÍA AGUILAR, JAVIER

Directores

ZAMORA MACHO, JUAN LUIS

MUÑOZ FRÍAS, JOSÉ DANIEL



Índice de contenidos

Capítulo I: Planta a trabajar	13
I.1 Elementos de la planta	21
I.2 Actuadores.....	22
I.3 Modelado matemático de la planta	22
I.4 Ensayo y parametrización de la planta	26
Capítulo II: Sistema de control.....	31
II.1 Control predictivo.....	31
II.2 Diseño de controles	32
II.3 Simulación de controles.....	34
Capítulo III: Bloques Simulink	37
Capítulo IV: STM32F407VG.....	41
IV.1 Elección del microcontrolador	41
IV.2 Programación. Waijung blockset.....	41
Capítulo V: Hardware in the loop.....	47
V.1 Propósito del hardware in the loop.....	47
V.2 Elementos necesarios para el HIL	48
V.3 Simulación HIL	52
Capítulo VI: Comunicación x10	55
VI.1 Protocolo x10.....	55
VI.2 Hardware x10 empleado	59



VI.3 Obtención de la señal Zero - Cross	61
VI.4 Generación de una señal auxiliar	62
VI.5 Generación de la señal x10.....	63
VI.6 Lógica x10. Máquina de estados.....	69
VI.7 Actuación x10 sobre los mandos.....	71
Capítulo VII: Conclusiones.....	77
VII.1 Controles.....	77
VII.2 Simulación con HIL	77
VII.3 Comunicación x10	77
Capítulo VIII: Futuros desarrollos	79
Capítulo IX: Referencias y bibliografía.....	81



Índice de figuras

Figura 1. Esquema de la planta	21
Figura 2. Diagrama empleado para la optimización	27
Figura 3. Test optimización. Iteración 1	28
Figura 4. Test optimización. Iteración 2	28
Figura 5. Test optimización. Iteración 3	29
Figura 6. Test optimización. Iteración 4	29
Figura 7. Test optimización. Iteración 5. Final	30
Figura 8. Pesos de los mandos y las salidas	32
Figura 9. Restricciones propias del control MPC	33
Figura 10. Diagrama de bloques para la comparación de los controles	34
Figura 11. Simulación control MPC	35
Figura 12. Simulación control PI	36
Figura 13: Resettable subsystem	37
Figura 14. Discrete integrator (Backward Euler Accumulator)	37
Figura 15. Configuración del acumulador	38
Figura 16. Selector block. Select Rows	38
Figura 17. MPC block	39
Figura 18. Bloque PWM	39



Figura 19. Configuración del subsistema PWM	40
Figura 20. Bloque de configuración.....	42
Figura 21. Configuración entradas digitales	42
Figura 22. Configuración salidas digitales.....	43
Figura 23. Configuración UART STM32.....	43
Figura 24. Bloque Rx UART STM32.....	44
Figura 25. Bloque Tx. UART STM32	44
Figura 26. Configuración UART host	44
Figura 27. UART receive (host)	45
Figura 28. UART transmit (host).....	45
Figura 29. Esquema del HIL.....	47
Figura 30. Diagrama de bloques HIL (STM32).....	48
Figura 31. Enabled subsystem. Control MPC.....	48
Figura 32. HIL. Adaptación PWM de los mandos.....	49
Figura 33. Diagrama de bloques HIL (PC). Recepción HIL	50
Figura 34. Diagrama de bloques HIL (PC). Simulación MPC sin adaptación PWM.....	51
Figura 35. Diagrama de bloques HIL (PC). Simulación MPC con adaptación PWM.....	51
Figura 36. Diagrama de bloques HIL. Representación y captura de los datos	52
Figura 37. Simulación HIL habitación 1.....	53
Figura 38. Simulación HIL habitación 2.....	53
Figura 39. Simulación HIL habitación 3.....	54
Figura 40. Simulación HIL habitación 4.....	54



Figura 41. Paso por cero de una senoidal	55
Figura 42. Escucha de los receptores x10	56
Figura 43. Codificación x10 de un '1' binario	56
Figura 44. Codificación x10 de un '0' binario	57
Figura 45. Código de cabecera x10.....	57
Figura 46. Módulo de luz LM 12 de Marmitek	59
Figura 47. Módulo de interfaz x10 XM10FL de Marmitek	60
Figura 48: Senoidal red y ZC con resistencia de pull-up.....	61
Figura 49: Senoidal red y ZC sin resistencia de pull-up	62
Figura 50: Diagrama para generar la señal auxiliar	63
Figura 51: Señal auxiliar y señal ZC.....	63
Figura 52: Subsistema 'x10 signal composer'	64
Figura 53: Signal composer. Selección de trama.....	67
Figura 54: Signal composer. Generación de trama	68
Figura 55: Signal composer. Final de mensaje	69
Figura 56: Máquina de estados. Lógica x10	70
Figura 57: Máquina de estados. Gestión de mensaje x10	71
Figura 58: Conexionado entre la máquina de estados y el subsistema 'Signal composer'	71
Figura 59: Máquina de estados. Gestión de mandos.....	72
Figura 60: Máquina de estados. Gestión de mandos x10.....	73
Figura 61: Diagrama de Simulink. Simulación mandos x10	73
Figura 62: Montaje para el ensayo de la comunicación.....	74



Figura 63: Simulación mensaje x10 con dos mandos.....	75
Figura 64: Simulación trama x10. Detalle 1	76
Figura 65: Simulación trama x10. Detalle 2	76



Índice de tablas

Tabla I. Resistencias térmicas entre las habitaciones	22
Tabla II. Capacidades térmicas de las habitaciones	22
Tabla III. Potencias nominales de los radiadores.....	22
Tabla IV. Resultados de la optimización	30
Tabla V. Parámetros de los controles PI.....	33
Tabla VI. Datos simulación control MPC.....	35
Tabla VII. Datos simulación control PI	36
Tabla VIII. Tramas lógicas x10 para cada código	58
Tabla IX: Entradas y salidas del subsistema "x10 Signal Composer"	65
Tabla X: Datos de entrada al subsistema 'x10 signal composer'.....	66
Tabla XI. Máquina de estados. Lógica x10. I/O	70
Tabla XII: Máquina de estados. Gestión x10 de los mandos.....	72



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA ELECTROMECÁNICA





Índice de ecuaciones

Ecuación 1. Almacenamiento de energía térmica en un medio	23
Ecuación 2. Transmisión de energía térmica entre dos medios	23
Ecuación 3. Potencia generada en la habitación n	24
Ecuación 4. Potencia con la que se almacena energía en el medio n.....	24
Ecuación 5. Potencia con la que se transfiere energía del medio m al n	24
Ecuación 6. Balance de potencias habitación 1	24
Ecuación 7. Balance de potencias habitación 2	25
Ecuación 8. Balance de potencias habitación 3	25
Ecuación 9. Balance de potencias habitación 4	25
Ecuación 10. Entradas de la representación de estados	25
Ecuación 11. Variables de estado	25
Ecuación 12. Salidas de la representación de estado	25
Ecuación 13. Representación de estado del Sistema	25



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA ELECTROMECÁNICA





Capítulo I: INTRODUCCIÓN

Actualmente, la tecnología nos envuelve en todos los aspectos, mejorando nuestra calidad de vida tanto en términos de confort como en simplificación y optimización de tareas. Esto ha sido posible, sobre todo, al gran desarrollo que han experimentado las tecnologías TIC (tecnologías de la información y comunicación), lo que ha permitido coordinar diversos sistemas y dispositivos.

En éstos términos, aparece el concepto de domótica, que viene a ser un conjunto de sistemas automatizados y coordinados entre sí para aportar servicios de bienestar, gestión energética y seguridad dentro de una vivienda. Ésta tecnología no es nueva de ahora, pero hasta hace poco, el precio de su instalación requería de tener un poder adquisitivo elevado. Sin embargo, con el reciente abaratamiento de la ésta, y a la aparición de alternativas, ahora mismo la domótica está al alcance de prácticamente todos, y cada vez son más los que deciden hacer uso de ella en distintas medidas.

Entre los principales aspectos que se quieren automatizar, está la temperatura. Hoy en día existen múltiples sistemas de climatización, pero, en su mayoría, los sistemas de control que implementan son muy simples y pueden darse fluctuaciones de la temperatura que alteren notablemente el confort del usuario.

El objetivo de éste proyecto es diseñar e implementar un sistema de control de temperature para una vivienda de cuatro habitaciones, mediante el uso de controles predictivos. Esto se hará en un contexto de domotizar la vivienda, por lo cual el sistema de control deberá de presentar una manera cómoda de comunicarse con otros sistemas y con una interfaz con el usuario.



I.1 ESTADO DEL ARTE

Los controles de temperatura más empleados hoy en día son los controles "todo / nada" y los controles PID (Proporcional/Integral/Derivativo). Otros sistemas de control, como los predictivos, no son habituales de encontrar en sistemas de climatización.

Todo / Nada

Es el sistema más comúnmente implementado en termostatos para viviendas.

Este sistema de control se basa en mantener la temperatura en un margen de temperaturas determinado aplicando potencia máxima o ninguna en absoluto.

Es un control de temperatura muy sencillo de implementar.

En contramedida, las fluctuaciones de temperatura que permite son muy elevadas y pueden llegar a afectar a la comodidad del usuario.

PID

Los controles PID, o variantes suyas como el PI, también suelen utilizarse en domótica. Es un control mucho más preciso que el Todo / Nada, pero también presenta varias desventajas, como es la necesidad de ajustar muy finamente el control para evitar tanto fluctuaciones desmedidas (control demasiado rápido) como para atenuar el efecto de las perturbaciones (control demasiado lento).

Controles predictivos

Presentan una mayor complejidad con respecto a los dos anteriores, lo que también permite mejores prestaciones. Los controles predictivos calculan el valor estimado que deberá de tomar el mando (potencia de los radiadores, por ejemplo) en los siguientes intervalos de tiempo, al contrario que el PID, que solo lo hace para el mismo instante en el que trabaja. Son controles muy fiables, capaces de operar durante mucho tiempo sin intervenciones.



Los controles predictivos más extendidos son los basados en un modelo de la planta (MPC) y los adaptativos. Los basados en modelo requieren un modelo matemático de la planta muy preciso para funcionar correctamente, mientras que los adaptativos se van regulando a si mismos a lo largo del tiempo en función de los datos anteriores y son más fiables al haber pasado un cierto tiempo.

I.2 CONTROL PREDICTIVO

Los algoritmos de control predictivo están diseñados de forma que sean muy intuitivos de usar e implantar. La toolbox de 'Model Predictive Control' para Simulink ofrece una manera rápida y eficaz de implementar este tipo de controles, permitiendo modificar todos los parámetros relevantes de forma ordenada.

En el ámbito industrial, se ha probado también la efectividad de los controles predictivos [1], como en el caso de un control multivariable para un horno industrial [2]. En cuanto a la forma de proceder de los controles predictivos, puede esquematizarse de la siguiente forma:

1. Uso de un modelo de predicción (planta) para predecir el valor de la salida del sistema en los instantes siguientes dentro de un horizonte de predicción.
2. Cálculo de una secuencia de control para minimizar una función objetivo.
3. El uso de una 'estrategia móvil', en la cual en cada instante el horizonte de predicción se desplaza en el tiempo, de forma que se está continuamente optimizando el proceso.

A continuación se enumeran los conceptos más básicos a la hora de diseñar un control predictivo:



Periodo de muestreo:

Es el tiempo entre dos cálculos del mando a aplicar. Para el sistema relacionado con éste proyecto, un periodo de muestreo de varios minutos es suficiente, dadas las lentas dinámicas que presenta la temperatura. Horizonte de predicción: Define cuántos "periodos de muestreo" desde el momento actual va a tener el control cada vez que efectúe un ciclo de cálculos.

Referencia:

Valor deseado de la variable que se quiera controlar. El control trata de mantener la variable física lo más parecida posible a la referencia, dentro de unos límites.

Límites de los mandos:

Especifica los mandos máximos y mínimos del sistema. En éste proyecto, serían los límites de potencia para cada radiador, para la máxima potencia, y '0', para la mínima, ya que la planta dónde se va a diseñar sólo dispone de sistema de calefacción.

Modelo matemático de la planta:

Para el correcto funcionamiento del sistema, se tiene que definir un modelo matemático fiable de la planta, teniendo en cuenta también el efecto de posibles perturbaciones (temperatura exterior). Función de optimización: Función que se pretende minimizar a la hora de calcular los mandos del sistema. El control modificará la variable física para parecerse lo mejor posible a la referencia respetando ésta función. En este caso en concreto, podría ser una función cuadrática para minimizar el tiempo en que la variable física se estabiliza con la referencia.

Por último, la principal diferencia entre los controles predictivos adaptativos y los basados en modelo radica en que los primeros no hace falta que tengan un modelo matemático de qué ocurre con las perturbaciones en su sistema. Para tenerlas en cuenta, el propio algoritmo de control se autorregula en función de los resultados obtenidos a lo largo del tiempo de ejecución.

I.3 ARQUITECTURA DEL PROYECTO

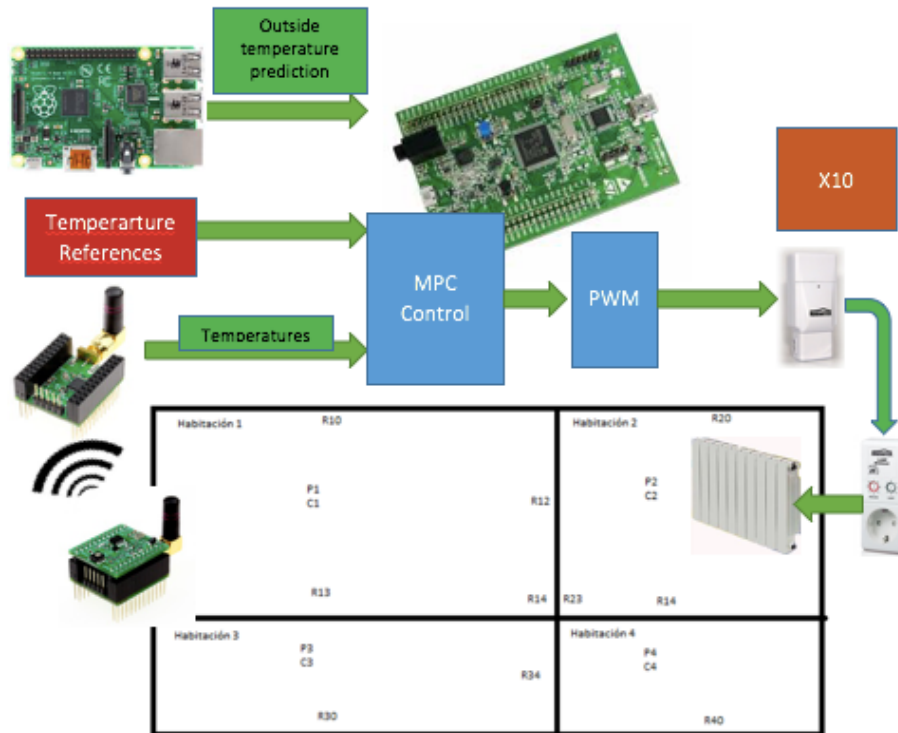


Figura 1. Estructura del Sistema

En la Figura 1 se muestra la estructura básica del sistema. Básicamente, consta del microcontrolador STM32, donde está ejecutándose el control, la planta física a trabajar, los sensores de temperatura y los actuadores. Los sensores de temperatura le envían los datos al microcontrolador por radiofrecuencia. Por otra parte, los actuadores son accionados por comunicación x10.

I.4 MOTIVACIÓN

La mayoría de los sistemas domóticos disponibles actualmente presenta algún inconveniente de los siguientes: su precio es excesivo, su implementación no es fácil o es demasiado simple y sus funciones, insuficientes. En este proyecto se pretende



implementar un sistema económico con buenas prestaciones, que sea instalable en cualquier recinto conectado a la red eléctrica, y permita un desarrollo de una interfaz con el usuario fiable y cómoda.

Por ello, se optará por el hardware de FiO, tecnología x10 y controles predictivos.

I.5 OBJETIVOS

El objetivo principal de este proyecto es implementar un sistema de control predictivo de temperatura en una vivienda, basándose en el hardware de Aimagin.

Para ello, se llevaron a cabo distintas tareas:

1. Conseguir una comunicación x10 desde el microcontrolador STM32.
2. Diseñar un control predictivo basándose en un modelo de la planta.
3. Conseguir una simulación hardware in the loop (HIL) que verifique que el microcontrolador es capaz de ejecutar el control.

I.6 METODOLOGÍA DE TRABAJO

A continuación se explica la metodología que se ha empleado en cada tarea del proyecto para llevarlas a cabo.

Por parte de la comunicación x10, se procedió a programar ésta a base de diagramas de bloques y máquinas de estados en Simulink. Para comprobar el correcto funcionamiento, se actuó sobre una lámpara de flexo.

Para el diseño del control predictivo, se empleó la herramienta ‘MPC tool’ de Matlab. El control se simuló con un modelo matemático de la planta para comprobar su correcto funcionamiento.



Para la estrategia HIL, se ejecutó el control en el microcontrolador, a la vez que en el ordenador se simulaba la planta en tiempo real. La comunicación entre ambos (temperaturas y mandos) se efectúa mediante comunicación UART.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA ELECTROMECÁNICA





Capítulo II: PLANTA A TRABAJAR

En este capítulo se introduce la planta con la que se va a trabajar. En el apartado II.1 se introducen los elementos y características físicas de la planta. En II.2 se explica cómo son los actuadores utilizados en ella. En II.3 se presenta el modelado matemático. Por último, en el apartado II.4 se explica cómo ensayar la planta para obtener unos parámetros más fiables de ésta.

II.1 ELEMENTOS DE LA PLANTA

La planta a trabajar consiste en una vivienda de 4 habitaciones. A cada habitación le corresponde una capacidad térmica C_n . Por otro lado, existe una resistencia térmica R_{mn} entre cada par de habitaciones y entre una habitación y el exterior de la vivienda. Además, en cada habitación hay una potencia de entrada (un radiador eléctrico) que proporciona una potencia térmica P_n para actuar sobre la temperatura de la habitación. No se consideran las variables correspondientes a la transmisión de calor por radiación al ser ésta no lineal y sólo ser significativa a muy alta temperatura. En la Figura 2 puede observarse el esquema de la planta.

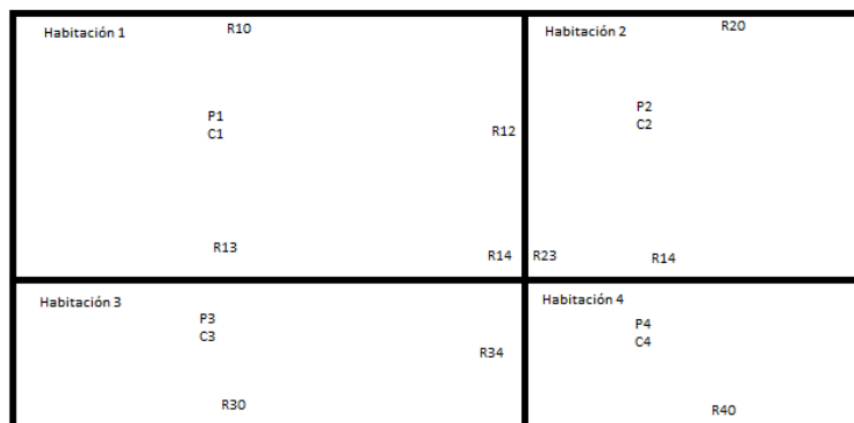


Figura 2. Esquema de la planta

En la Tabla I se muestran las resistencias térmicas empleadas para el modelado de la planta. Lo propio ocurre con las capacidades térmicas y la Tabla II.



Resistencias térmicas (W/K)					
	Exterior	Hab 1	Hab 2	Hab 3	Hab 4
Exterior	-----	0.03	0,06	0.085	0.085
Hab 1		-----	0.015	0.01	0.02
Hab 2			-----	0.015	0.015
Hab 3				-----	0.015
Hab 4					-----

Tabla I. Resistencias térmicas entre las habitaciones

Capacidades térmicas (W*s/K)			
Hab 1	Hab 2	Hab 3	Hab 4
7e5	3.5e5	3.5e5	3e5

Tabla II. Capacidades térmicas de las habitaciones

II.2 ACTUADORES

Los actuadores empleados son un radiador eléctrico cerámico por habitación. Estos radiadores proporcionan su potencia nominal cuando están encendidos, o nulo, cuando están apagados. Para proporcionar una potencia intermedia, tiene que modularse una señal PWM en el tiempo con valor medio la potencia que se quiera aplicar. En la Tabla III se observan las potencias térmicas nominales de los radiadores de cada habitación.

Potencias térmicas de los radiadores (W)			
Hab 1	Hab 2	Hab 3	Hab 4
2400	1200	1200	800

Tabla III. Potencias nominales de los radiadores

II.3 MODELADO MATEMÁTICO DE LA PLANTA

Se supone un sistema lineal e invariante en el tiempo (LTI) para el modelado de la planta (sistema térmico). Se considera que el medio exterior a la vivienda tiene la posibilidad de intercambiar energía térmica con el resto de medios del sistema sin afectar a su propia



temperatura. Es decir, las variaciones de temperatura del medio exterior no se deben a su interacción con el resto de medios de la planta [1].

Por otro lado, se supone un valor uniforme de temperatura en cada medio, para evitar tener que modelar las ecuaciones en derivadas parciales que marcan el comportamiento del almacenamiento de la energía térmica y la transmisión de la misma dentro de un mismo medio.

Para modelar el sistema, es necesario conocer las leyes que rigen el almacenamiento de energía en un medio y la transmisión de energía entre medios.

$$P_a = \frac{dE_a}{dt} = C \times \frac{dT}{dt}$$

Ecuación 1. Almacenamiento de energía térmica en un medio

La Ecuación 1 rige el almacenamiento de energía térmica en un medio físico, donde P_a es la potencia con la que se almacena energía E_a en el medio, C [W*s/K] es la capacidad térmica del medio, T la temperatura del medio, y t el tiempo.

$$P_t = \frac{T_1 - T_2}{R_{12}}$$

Ecuación 2. Transmisión de energía térmica entre dos medios

La Ecuación 2 rige el comportamiento de la transmisión de energía térmica entre dos medios. P_t es la potencia transferida entre ambos medios, T_1 y T_2 las temperaturas de los medios 1 y 2, respectivamente, y R_{12} [W/K] la resistencia térmica entre ambos medios.

Para modelar la planta se emplea una representación de estado lineal. Sus parámetros de mayor importancia son los siguientes:

Entradas: Potencias térmicas suministradas P_n

Variables de estado: Temperaturas T_n de las habitaciones.

Salidas: Coinciden con las variables de estados.



Perturbaciones: Temperatura exterior T_0 y cuatro potencias térmicas Pd_n , que modelan las perturbaciones internas al medio n .

Esta representación de estados se traduce en la siguiente ecuación (1 por habitación), obtenida por balance de potencias:

$$P_n = Pa_n + \sum_{\substack{m=0 \\ m \neq n}}^4 Pt_{nm} - Pd_n$$

Ecuación 3. Potencia generada en la habitación n

Donde P_n es la potencia generada en la habitación n , Pa_n , la potencia con la que se almacena energía en n , Pt_{nm} la potencia térmica transmitida desde el medio n hasta el m , y Pd_n , la potencia de perturbaciones en la habitación n . El cálculo de Pa_n y Pt_{nm} se muestra en la Ecuación 4 y la Ecuación 5, respectivamente.

$$Pa_n = C_n \times \frac{dT_n}{dt}$$

Ecuación 4. Potencia con la que se almacena energía en el medio n

Siendo C_n y T_n la capacidad térmica y la temperatura del medio n , respectivamente.

$$Pt_{nm} = \frac{T_n - T_m}{R_{nm}}$$

Ecuación 5. Potencia con la que se transfiere energía del medio m al n

Donde T_n y T_m son las temperaturas de los medios m y n , y R_{nm} la resistencia térmica entre ambos medios.

Se desarrolla la Ecuación 3 para cada habitación:

$$P_1 = C_1 \times \frac{dT_1}{dt} + \frac{T_1 - T_2}{R_{12}} + \frac{T_1 - T_3}{R_{13}} + \frac{T_1 - T_4}{R_{14}} + \frac{T_1 - T_0}{R_{10}} - Pd_1$$

Ecuación 6. Balance de potencias habitación 1



$$P_2 = C_2 \times \frac{dT_2}{dt} + \frac{T_2 - T_1}{R_{12}} + \frac{T_2 - T_3}{R_{23}} + \frac{T_2 - T_4}{R_{24}} + \frac{T_2 - T_0}{R_{20}} - Pd_2$$

Ecuación 7. Balance de potencias habitación 2

$$P_3 = C_3 \times \frac{dT_3}{dt} + \frac{T_3 - T_1}{R_{13}} + \frac{T_3 - T_2}{R_{23}} + \frac{T_3 - T_4}{R_{34}} + \frac{T_3 - T_0}{R_{30}} - Pd_3$$

Ecuación 8. Balance de potencias habitación 3

$$P_4 = C_4 \times \frac{dT_4}{dt} + \frac{T_4 - T_1}{R_{14}} + \frac{T_4 - T_2}{R_{24}} + \frac{T_4 - T_3}{R_{34}} + \frac{T_4 - T_0}{R_{40}} - Pd_4$$

Ecuación 9. Balance de potencias habitación 4

Si se despejan las derivadas parciales de las variables de estado y se definen los vectores siguientes,

$$U = [P_1 \ P_2 \ P_3 \ P_4 \ T_0 \ Pd_1 \ Pd_2 \ Pd_3 \ Pd_4]'$$

Ecuación 10. Entradas de la representación de estados

$$X = [T_1 \ T_2 \ T_3 \ T_4]'$$

Ecuación 11. Variables de estado

$$Y = X$$

Ecuación 12. Salidas de la representación de estado

Se obtiene la representación de estado siguiente:

$$\frac{dX}{dt} = [A] \times X + [B] \times U$$

$$Y = [C] \times X + [D] \times U$$

Ecuación 13. Representación de estado del Sistema

Siendo [A], [B], [C] y [D] las matrices de la representación de estado.



II.4 ENSAYO Y PARAMETRIZACIÓN DE LA PLANTA

Con el propósito de obtener unos valores más fiables de las resistencias y capacidades térmicas, se diseñó un script de optimización que, basándose en los datos obtenidos de un ensayo de la planta, sacase esos valores. El ensayo en cuestión consiste en, a lo largo de 24 h, someter a la vivienda a un control proporcional, y variar la referencia de temperaturas de las viviendas en función de una señal PRBS. Éste tipo de señales se emplea para forzar los sistemas. [2] Tienen la propiedad de presentar una ganancia constante en todo el espectro de frecuencia [3]. Han de guardarse tanto las temperaturas como los mandos aplicados a lo largo del ensayo para su uso en el script de optimización.

El sistema de optimización funciona de la siguiente manera: a partir de los mandos aplicados a lo largo del ensayo, y las temperaturas medidas, se optimizan los valores de las resistencias y capacidades térmicas por mínimos cuadrados. Para ello, se parten de unos valores teóricos, y se simula la planta con ellos, usando como entrada los mandos del ensayo. A continuación, se modifican los parámetros ligeramente y se vuelve a simular la planta. El proceso se repite hasta que los valores de las temperaturas simuladas se asemejan lo suficiente a las temperaturas obtenidas en el ensayo.

En la Figura 3 puede observarse el diagrama de Simulink empleado para las simulaciones durante la optimización. Cabe mencionar que las constantes [time,end] y [time,sal] corresponden, respectivamente, a los mandos aplicados y las temperaturas medidas durante el ensayo de la planta.

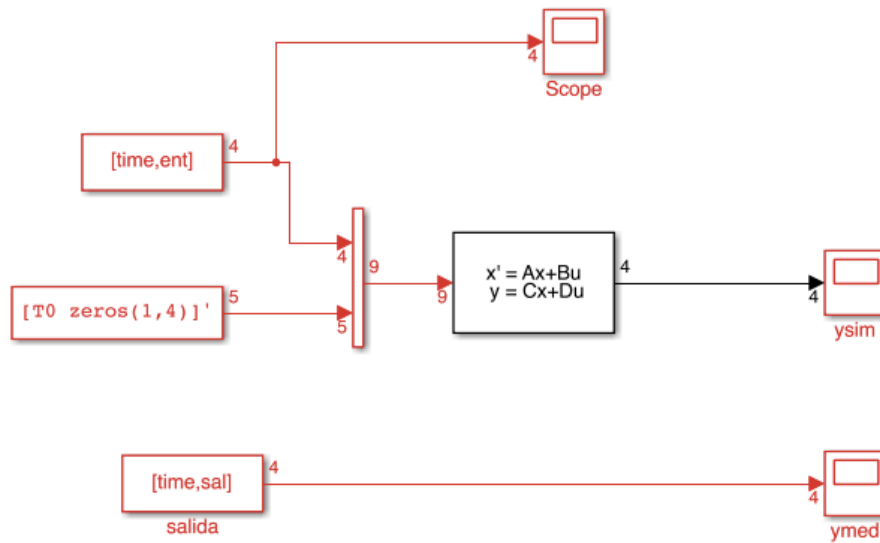


Figura 3. Diagrama empleado para la optimización

Cabe destacar que, para comprobar que el script funcionase correctamente, se consideró la temperatura exterior constante. A la hora de efectuar el ensayo real, habría que modificar ligeramente el diagrama para que la entrada de la temperatura exterior fuese la correcta a cada instante de tiempo.

En las figuras siguientes (Figura 4, Figura 5, Figura 6, Figura 7 y Figura 8) puede observarse una simulación del script de optimización, con todas sus iteraciones. En la Tabla IV se muestran los valores teóricos y los de inicio y final de la optimización, además del error obtenido.

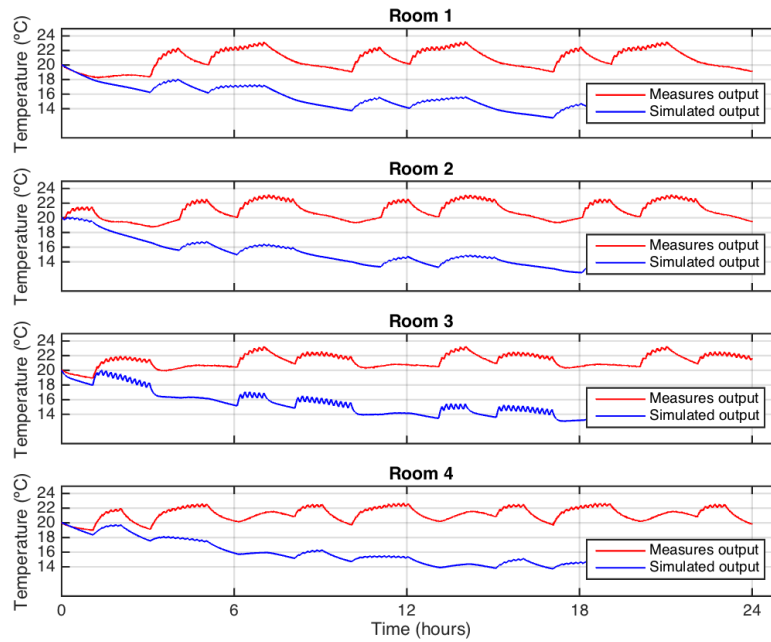


Figura 4. Test optimización. Iteración 1

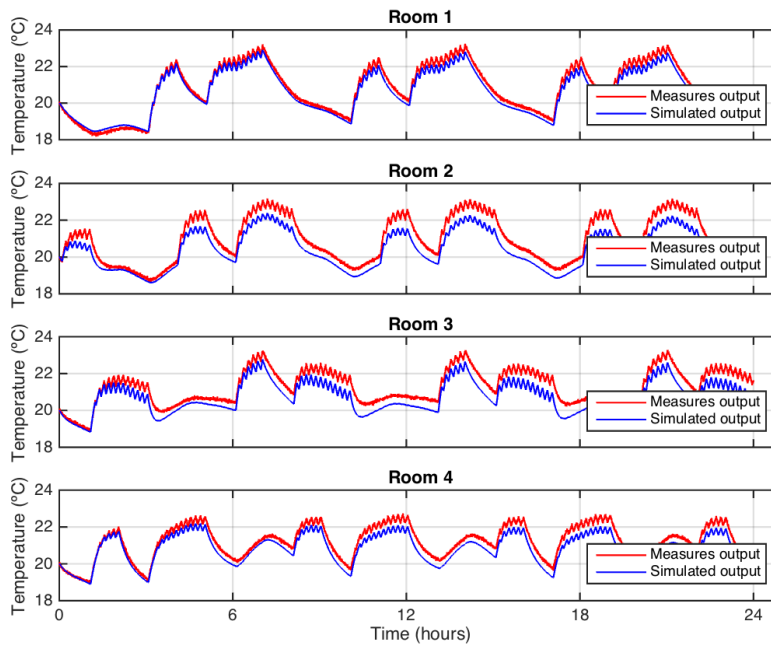


Figura 5. Test optimización. Iteración 2

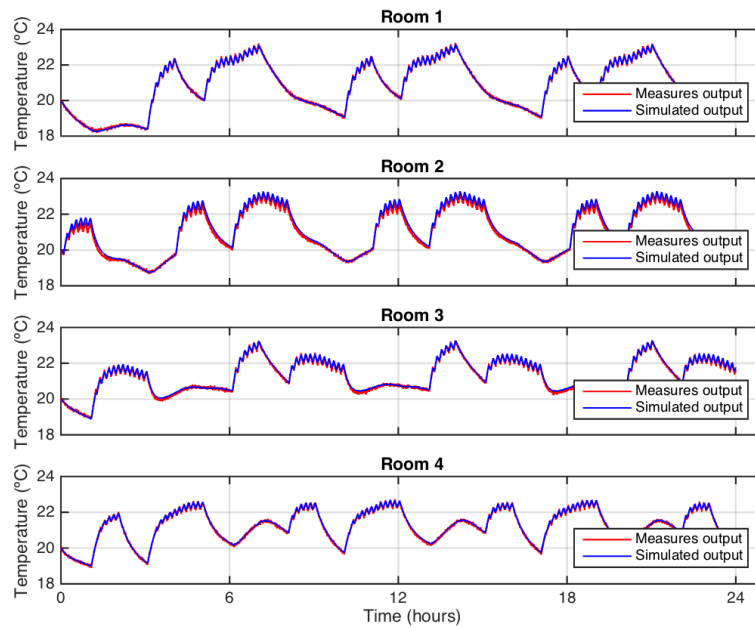


Figura 6. Test optimización. Iteración 3

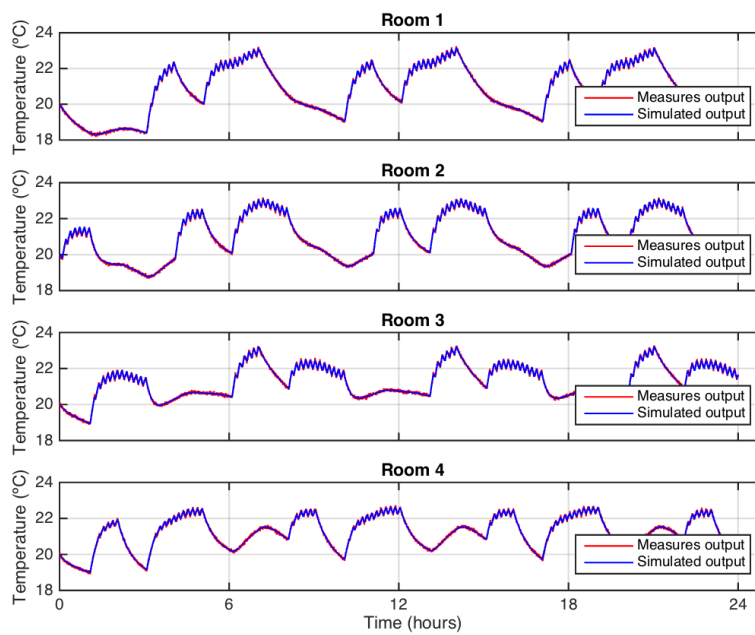


Figura 7. Test optimización. Iteración 4

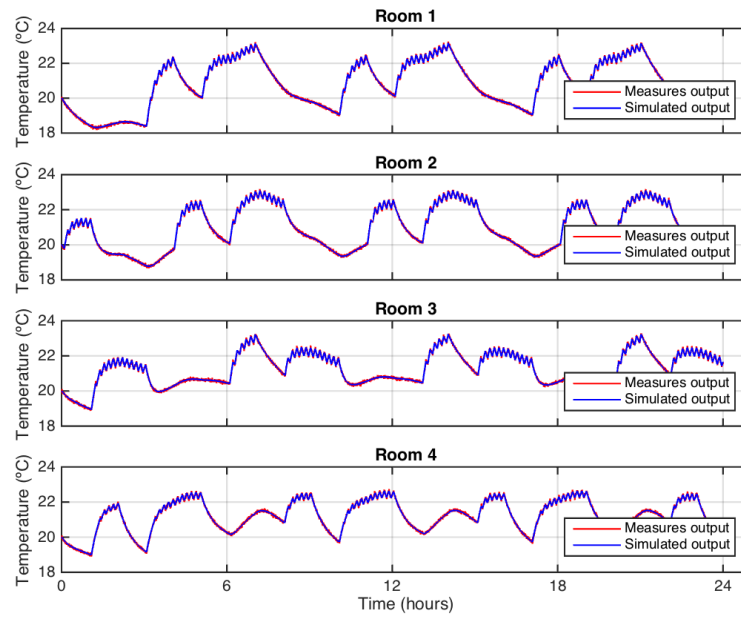


Figura 8. Test optimización. Iteración 5. Final

Resultados optimización				
Variable	Valor 'Real'	Valor inicial	Valor final	Error [%]
C_1	7e5	1.22e6	7e5	4.07e-3
C_2	3.5e5	2.37e5	3.503e5	4.53e-2
C_3	3.5e5	6.49e5	3.514e5	2.06e-1
C_4	3e5	1.11e5	3.015e5	2.1e-1
$1/R_{10}$	33.3	30	33.32	2.6e-6
$1/R_{12}$	66.7	26.8	66.35	4.45e-5
$1/R_{13}$	100	51.4	99.94	8.5e-6
$1/R_{14}$	50	105.6	50.18	2.5e-5
$1/R_{20}$	16.7	30.5	16.65	1.96e-6
$1/R_{23}$	66.7	66.5	66.72	8.51e-6
$1/R_{24}$	66.7	159.4	66.41	3.59e-5
$1/R_{30}$	17.8	4.42	11.8	4.62e-6
$1/R_{34}$	66.7	84.3	66.46	2.96e-6
$1/R_{40}$	17.8	13.4	17.7	1.17e-6

Tabla IV. Resultados de la optimización



Capítulo III: SISTEMA DE CONTROL

En este capítulo se comenta el sistema de control adoptado. En III.1 se introduce brevemente el control predictivo. En el apartado III.2 se enuncian los dos controles que se han empleado para su comparación. En III.3 se muestran las simulaciones de ambos controles. Por último, en el apartado **Error! Reference source not found.** se comparan los resultados de ambas simulaciones.

III.1 CONTROL PREDICTIVO

Para este proyecto, se ha adoptado una estrategia de control basada en un control predictivo basado en modelo (MPC). Los controles MPC se caracterizan por los siguientes aspectos [4]:

- a) Uso de un modelo matemático de la planta a trabajar para predecir la salida en instantes de tiempo futuros.
- b) Cálculo de secuencias de control que minimice una función objetivo.
- c) En cada instante, el horizonte de predicción se desplaza una muestra (estrategia deslizante).

Las principales diferencias entre los controles MPC radican en el modelo matemático que se emplea para simular la planta y en la función de coste asociada al control.

Las principales ventajas del control MPC son su gran aplicabilidad en múltiples cambios, la facilidad que presenta al tratar con sistemas MIMO y la capacidad de que el propio control tenga en cuenta las restricciones, tanto de la planta como de los mandos aplicados. Esto permite una mejor respuesta frente a otros controles. [5]

Además, gracias a la herramienta ‘MPCtool’ [6] de Matlab, su diseño se efectúa de forma simple e intuitiva.



Por otro lado, los controles MPC también presentan algunos inconvenientes. Los principales son la necesidad de contar con un modelo matemático del proceso (planta) que tiene que controlar y la mayor computación que requiere con respecto a otros controles, como son los PID.

III.2 DISEÑO DE CONTROLES

Para éste proyecto se diseñó un control MPC, y se comparó con un control PI, típicamente utilizado en aplicaciones de climatización de viviendas.

Para el control MPC, se optó por un periodo de muestreo de 5 minutos, un horizonte de control de 5 periodos de muestreo, y un horizonte de predicción de 25 periodos de muestreo. Esto significa que, cada 5 minutos, el control MPC hará un nuevo cálculo de mandos, basado en una predicción de 125 minutos (horizonte de predicción), pero calculando únicamente los mandos de los primeros 25 min (horizonte control). Para los valores del horizonte de predicción no calculados, el control supone que se aplica el último mando del horizonte de control. Esto es así para aliviar ligeramente los cálculos que tiene que efectuar el control a cada periodo de muestreo.

Overall
More robust Faster response

Value: 0.8

Input weights

Name	Description	Units	Weight	Rate Weight
P1		W	0	0.007
P2		W	0	0.015
P3		W	0	0.015
P4		W	0	0.02

Output weights

Name	Description	Units	Weight
T1		°C	1
T2		°C	1.3
T3		°C	1.5
T4		°C	1.25

Figura 9. Pesos de los mandos y las salidas



En la Figura 9 se muestran los pesos que se les ha dado a las temperaturas de salida y a los mandos. Estos pesos son los que utiliza la función objetivo que se minimiza en el control.

Constraints on manipulated variables					
Name	Units	Minimum	Maximum	Max Down Rate	Max Up Rate
P1	W	0	2400	-Inf	Inf
P2	W	0	1200	-Inf	Inf
P3	W	0	1200	-Inf	Inf
P4	W	0	800	-Inf	Inf

Constraints on output variables			
Name	Units	Minimum	Maximum
T1	°C	-Inf	Inf
T2	°C	-Inf	Inf
T3	°C	-Inf	Inf
T4	°C	-Inf	Inf

Figura 10. Restricciones propias del control MPC

En la Figura 10 puede contemplarse las restricciones impuestas. Se imponen restricciones de mando máximo y mínimo, únicamente, ya que la planta, como tal, no tiene ninguna salida restringida.

Para el control PI, por otro lado, se emplearon cuatro controles (uno por habitación) ya diseñados para ésta planta [1], cuyos parámetros se recogen en la Tabla V.

Controles PI				
	Hab 1	Hab 2	Hab 3	Hab 4
K	4095	1152	1222.5	1009.5
Ti	2956	4030	3630	3855

Tabla V. Parámetros de los controles PI

III.3 SIMULACIÓN DE CONTROLES

Ambos controles se sometieron a una simulación simultánea, en la que se variaba la referencia de temperatura, manteniendo constante la temperatura exterior a 5°C. En la Figura 11 puede apreciarse el diagrama de bloques empleado a tal efecto.

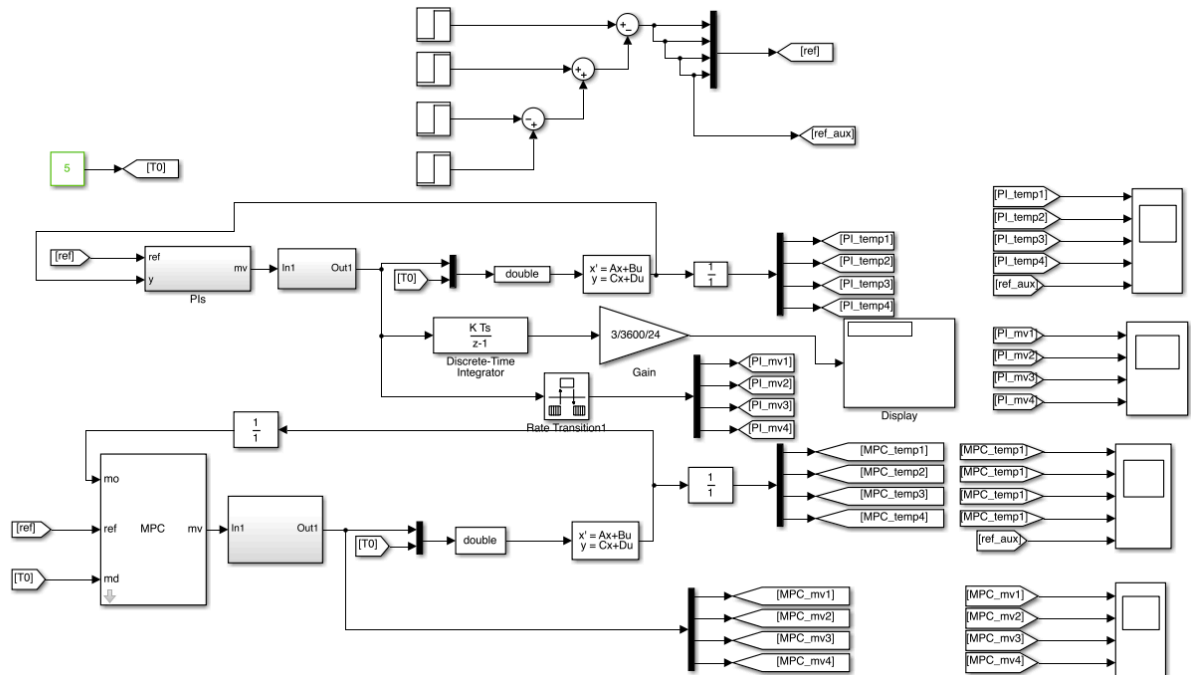


Figura 11. Diagrama de bloques para la comparación de los controles

En la Figura 12 y la Figura 13 pueden apreciarse las simulaciones de los controles MPC y PI, respectivamente. En la Tabla VI se recogen datos básicos de la simulación del control MPC. Lo propio ocurre con la Tabla VII y la simulación de los controles PI.

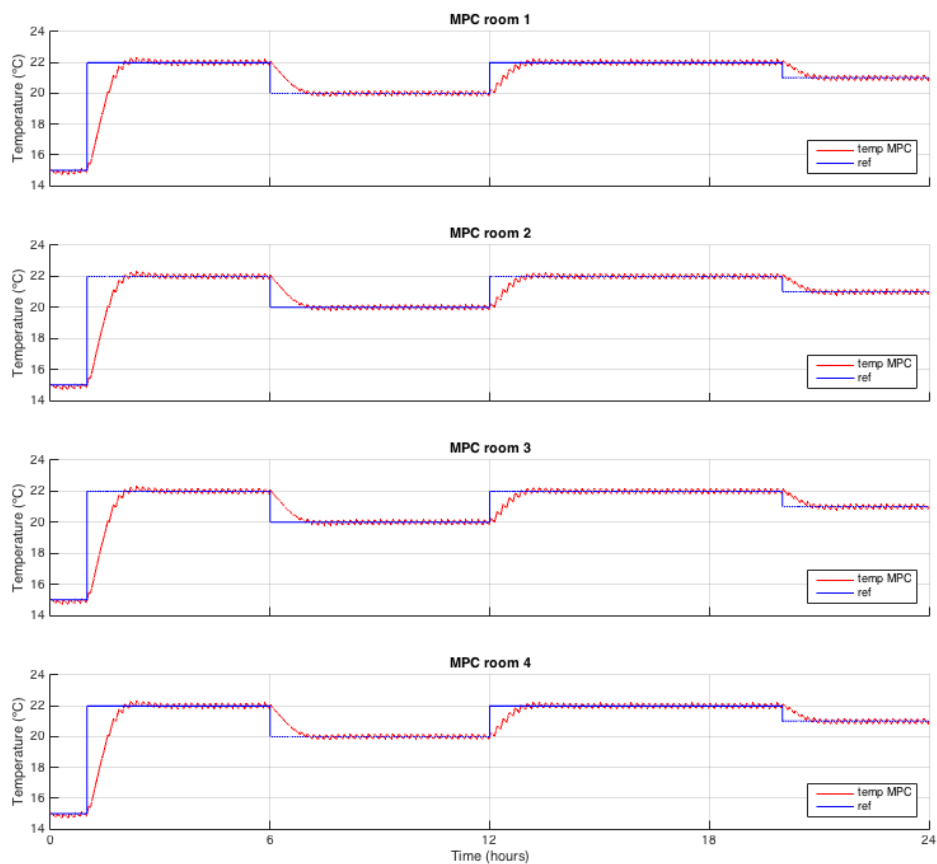


Figura 12. Simulación control MPC

Control MPC				
	Hab 1	Hab 2	Hab 3	Hab 4
Error absoluto medio (°C)	0.33	0.32	0.27	0.36
Potencia media (W)	1735	869.8	650.9	613.7

Tabla VI. Datos simulación control MPC

Potencia media total = 3869.4

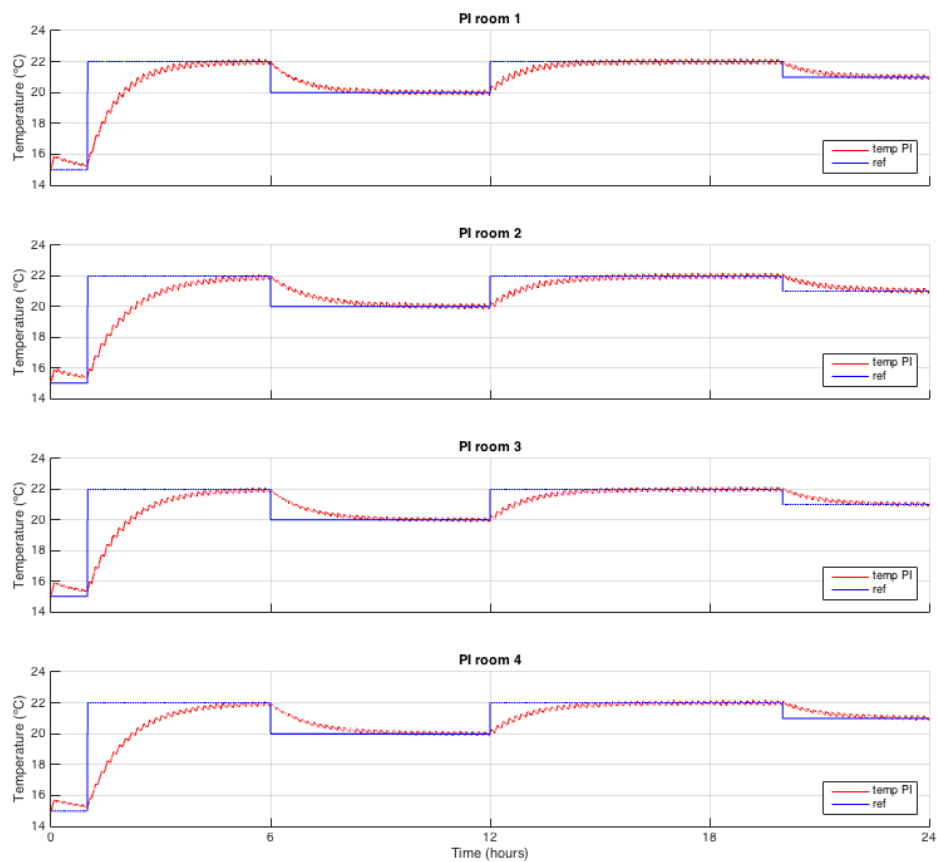


Figura 13. Simulación control PI

Control PI				
	Hab 1	Hab 2	Hab 3	Hab 4
Error absoluto medio (°C)	0.53	0.70	0.62	0.69
Potencia media (W)	1761	846.1	633.5	608.6

Tabla VII. Datos simulación control PI

Potencia media total = 3849.2

Capítulo IV: BLOQUES SIMULINK

En este apartado se comentan brevemente cuáles son los bloques menos comunes que se han empleado a lo largo del proyecto. [7]

RESETTABLE SUBSYSTEM

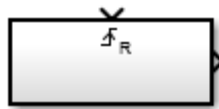


Figura 14: Resettable subsystem

Font: Simulink block

Éste bloque (Figura 14) tiene la peculiaridad de, cada vez que detecta un flanco de subida en su entrada 'R', resetea los valores de salida de todos los bloques a su valor por defecto. Es especialmente útil cuando quiere generarse una señal sincronizada con otra, como es el caso de la trama de la comunicación x10, que tiene que ir en sincronismo con la red para un correcto funcionamiento del protocolo.

ACUMULADOR

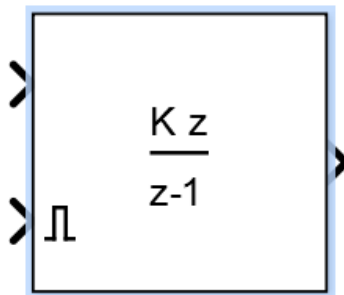


Figura 15. Discrete integrator (Backward Euler Accumulator)

Fuente: Simulink block



Este bloque (Figura 15), en su función de acumulador, permite llevar cuenta de las veces que se da un evento. Mediante una de las dos entradas del bloque, puede reiniciarse la cuenta. Este bloque se ha empleado para gestionar las tramas y la lógica del protocolo x10. En la Figura 16 puede apreciarse la configuración que se ha empleado para el funcionamiento del bloque.

Figura 16. Configuración del acumulador

SELECTOR BLOCK

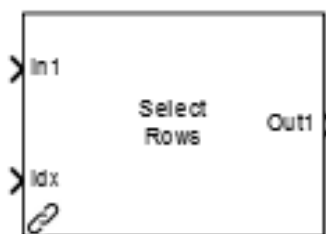


Figura 17. Selector block. Select Rows

Fuente: Simulink block



Este bloque permite seleccionar una fila o una columna de una matriz. La entrada 'Idx' es la que especifica qué fila o columna seleccionar. Usando su función de seleccionar filas (Figura 17), este bloque ha resultado extremadamente útil para facilitar la gestión del protocolo x10, tanto a la hora de seleccionar qué mensaje se tiene que enviar como en la composición temporal de la trama.

MPC BLOCK

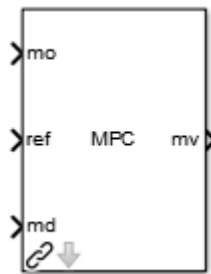


Figura 18. MPC block

Fuente: Simulink block

El MPC block (Figura 18) sirve para implementar un control predictivo en Simulink. Para ello, es necesario tener el control definido en el workspace de Matlab.

SUBSISTEMA PWM

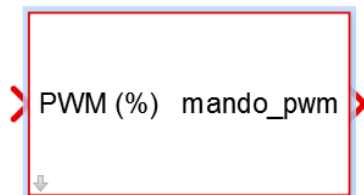


Figura 19. Bloque PWM

Fuente: Simulink subsystem



El subsistema PWM (Figura 19) permite modular una señal por ancho de pulso. Esta función es necesaria para adaptar los mandos de salida del control a los actuadores, que, al tratarse de radiadores eléctricos, solo permiten estar encendidos a máxima potencia.

En la Figura 20 puede observarse la configuración que se ha empleado del subsistema. ‘Número de puntos’ especifica la resolución que tiene el PWM dentro de su periodo. La opción ‘Alternancia de tramas’ sirve para solapar los estados activos entre dos periodos PWM consecutivos, reduciendo así el número de conmutaciones que presenta la señal.

Figura 20. Configuración del subsistema PWM



Capítulo V: STM32F407VG

En este capítulo se habla del microcontrolador que se ha usado en el proyecto, el STM32F407VG. En el apartado V.1 se explican los motivos por los que se ha optado por este microcontrolador. En el apartado V.2 se comenta la forma que se ha empleado para su programación.

V.1 ELECCIÓN DEL MICROCONTROLADOR

Se ha optado por utilizar un STM32F407VG como microcontrolador para este proyecto por su gran capacidad de cálculo, al tener un procesador de 32 bits que además trabaja en coma flotante, y por la gran cantidad de periféricos que presenta, de fácil uso y programación. Además, gracias a el blockset Waijung, puede ser programado directamente desde Simulink, facilitando así la implementación del control en el microcontrolador.

Frente a otros dispositivos de control, como podrían ser la plataforma Arduino o una Raspberry Pi, el STM32 presenta múltiples ventajas. Los Arduino, a pesar de tener muy buena gestión de periféricos, y de contar con numerosas librerías para su funcionamiento, presenta una potencia de cálculo muy limitada, no pudiendo ejecutar el control predictivo. La Raspberry Pi, en cambio, presenta serias dificultades a la hora de programarse correctamente desde Simulink.

V.2 PROGRAMACIÓN. WAIJUNG BLOCKSET

Para programar el STM32F407VG se ha empleado el blockset Waijung, un conjunto de bloques para Simulink específicamente diseñado para la familia de microcontroladores STM32. A continuación se nombra y se explica el funcionamiento de los bloques Waijung que se han empleado en éste proyecto. [8]



CONFIGURACIÓN

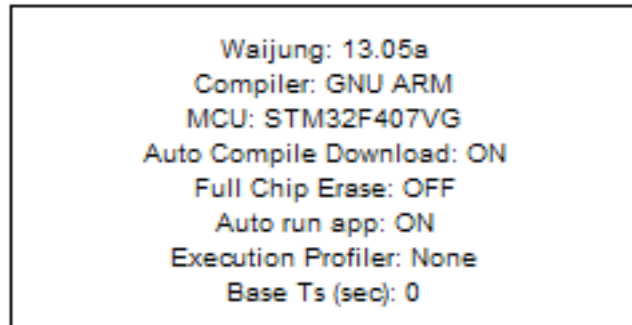


Figura 21. Bloque de configuración

Fuente: Waijung blockset

Este bloque (Figura 21) es el que permite programar el STM32 desde Simulink. Este bloque permite configurar las características del microcontrolador, permitiendo programar múltiples microcontroladores de la familia STM32.

ENTRADAS Y SALIDAS DIGITALES

Para gestionar las entradas digitales se hace uso del bloque de la Figura 22. Éste bloque permite seleccionar el puerto y los pines de entrada que se pretenden utilizar. Además, permite configurarlos para operar con resistencias de pull-up o pull-down internas, o sin ellas. La mayoría de puertos, si se opera sin resistencia de pull-up o pull-down interna, permiten trabajar con 5 V de entrada.

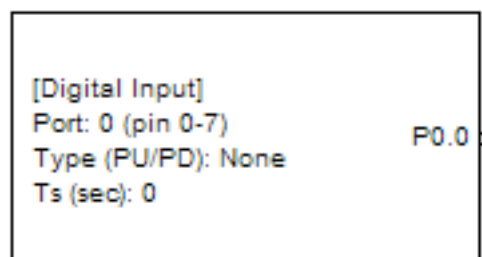


Figura 22. Configuración entradas digitales

Fuente: Waijung blockset



Las salidas digitales se configuran con el bloque de la Figura 23. Análogamente al bloque de entradas digitales, permite seleccionar los puertos y pines de salida.

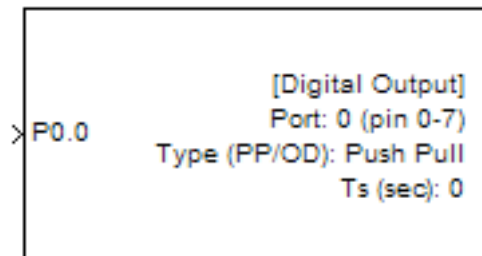


Figura 23. Configuración salidas digitales

Fuente: Waijung blockset

PUERTOS UART

Para la comunicación por UART se emplean seis bloques. Tres de ellos son la configuración, el de recepción y el de emisión de datos por parte del STM32 (Figura 24, Figura 25 y Figura 26, respectivamente). El bloque de configuración permite seleccionar qué puerto UART del microcontrolador se usa, la velocidad de transmisión, y los pines por los que se realiza la transmisión (2 posibilidades por puerto UART). En este proyecto se ha empleado el USART 3, que es el puerto rápido de comunicaciones que dispone el microcontrolador.

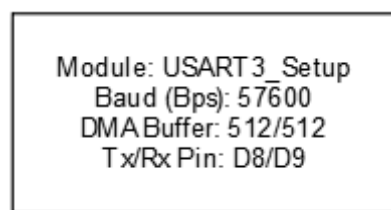


Figura 24. Configuración UART STM32

Fuente: Waijung blockset

Los bloques de emisión y recepción de datos, por su parte, permiten configurar el número y el tipo de datos a enviar.



Figura 25. Bloque Rx UART STM32

Fuente: Waijung blockset

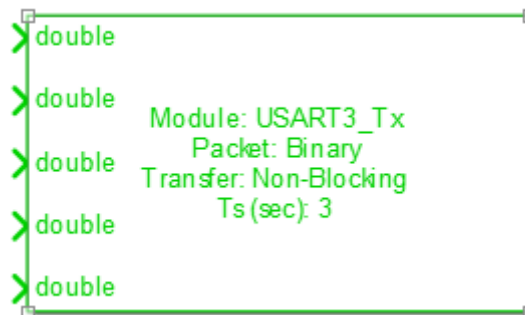


Figura 26. Bloque Tx. UART STM32

Fuente: Waijung blockset

Por otro lado, para su uso en el host (ordenador), Waijung dispone de otros tres bloques a emplear, análogos a los empleados para el microcontrolador (Figura 27, Figura 28 y Figura 29). Éstos bloques sólo han de emplearse si se emplea el puerto rápido del microcontrolador. Si se emplea otro tipo de puerto, los bloques a emplear son los propios de Simulink.

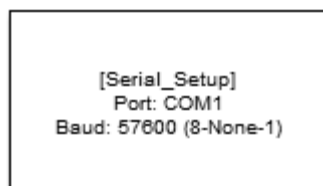


Figura 27. Configuración UART host

Fuente: Waijung blockset

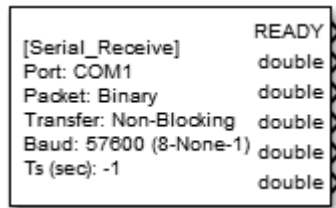


Figura 28. UART receive (host)

Fuente: Waijung blockset

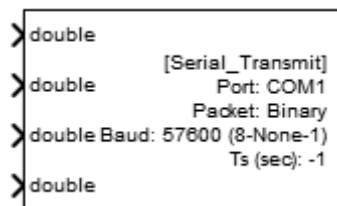


Figura 29. UART transmit (host)

Fuente: Waijung blockset



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA ELECTROMECÁNICA



Capítulo VI: HARDWARE IN THE LOOP

En este capítulo se presenta el trabajo realizado para asegurar que el microcontrolador era capaz de efectuar correctamente las operaciones del control predictivo. Para ello, se hizo uso de una estrategia de Hardware In the Loop (HIL). En el apartado VI.1 se comenta cuál es el propósito del HIL y cómo va a llevarse a cabo. En VI.2 se enumeran y explican todos los elementos que han sido necesarios para llevar a cabo un HIL con éxito. Por último, en el apartado VI.3 se muestran los resultados obtenidos con el HIL.

VI.1 PROPÓSITO DEL HARDWARE IN THE LOOP

El HIL tiene como propósito verificar si el microcontrolador STM32F407VG es capaz de gestionar y efectuar correctamente los cálculos que requiere el control predictivo [9] [10]. Para ello, se hace uso de un sistema como el representado en la Figura 30.

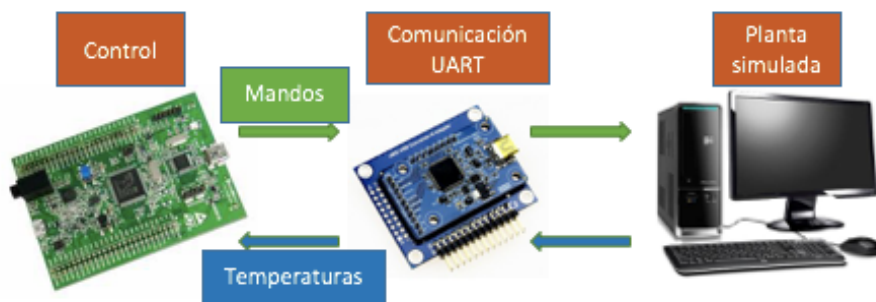


Figura 30. Esquema del HIL

En el HIL, el control está ejecutándose en el microcontrolador, mientras que en un ordenador se está simulando el modelo matemático de la planta a tiempo real. La comunicación entre ambos se efectúa mediante puerto serie (UART). El microcontrolador recibe las temperaturas simuladas del ordenador, que utiliza para ejecutar el control, y posteriormente envía de vuelta los mandos calculados. Estos mandos se aplican a la planta matemática, que simula las dinámicas del sistema.

VI.2 ELEMENTOS NECESARIOS PARA EL HIL

Para que funcione correctamente el HIL, se construyeron dos diagramas de bloques en Simulink. Uno de ellos se carga en el microcontrolador; el otro se ejecuta en el ordenador. A continuación se procede a explicar ambos diagramas.

El diagrama de bloques para el HIL correspondiente al STM32 puede apreciarse en la Figura 31. Para su correcto funcionamiento y carga en el microcontrolador, los bloques de Waijung de configuración del micro y del puerto USART 3 (puerto serie rápido) tienen que incluirse en el diagrama. El sistema de control (Figura 32), para evitar un funcionamiento inadecuado, se encuentra en un subsistema que se habilita cuando se reciba algo por el puerto serie (sumatorio de las señales recibidas mayor que 0).

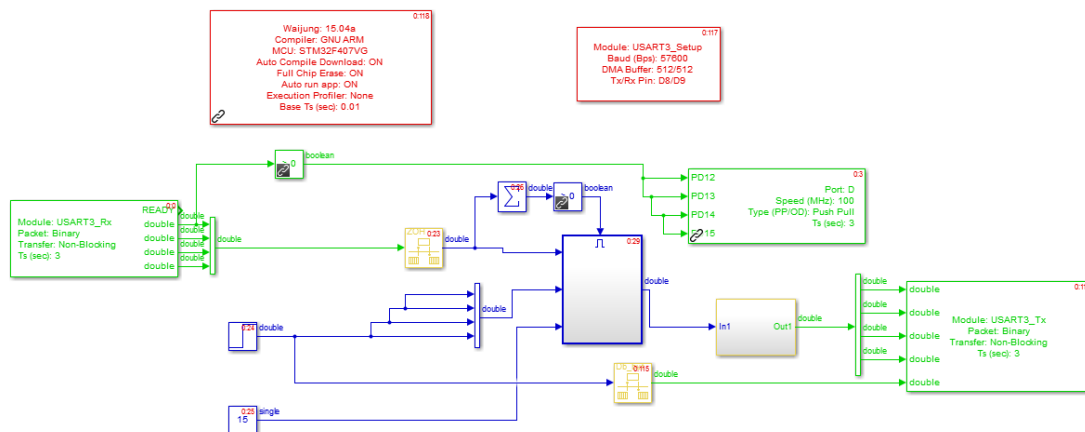


Figura 31. Diagrama de bloques HIL (STM32)

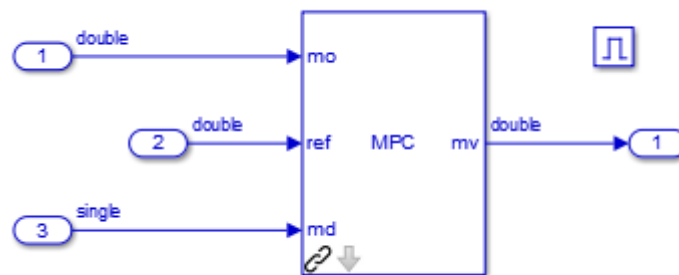


Figura 32. Enabled subsystem. Control MPC



A los mandos de salida del control, posteriormente, se les adapta a PWM (Figura 33). Posteriormente, los mandos resultantes son enviados al ordenador por medio del puerto serie.

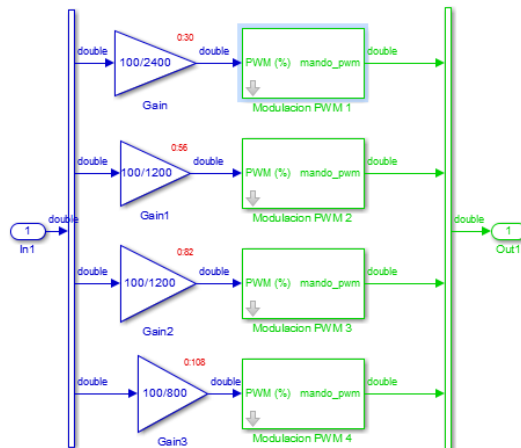


Figura 33. HIL. Adaptación PWM de los mandos

El diagrama de bloques que se ejecuta en el ordenador se compone de tres partes. La primera, representada en la Figura 34, es la parte que compete directamente al HIL. La segunda parte (Figura 35 y Figura 36) corresponde a la simulación de la planta, para su comparación con los resultados del HIL. La última parte recoge la monitorización y guardado de datos de la simulación (Figura 37).

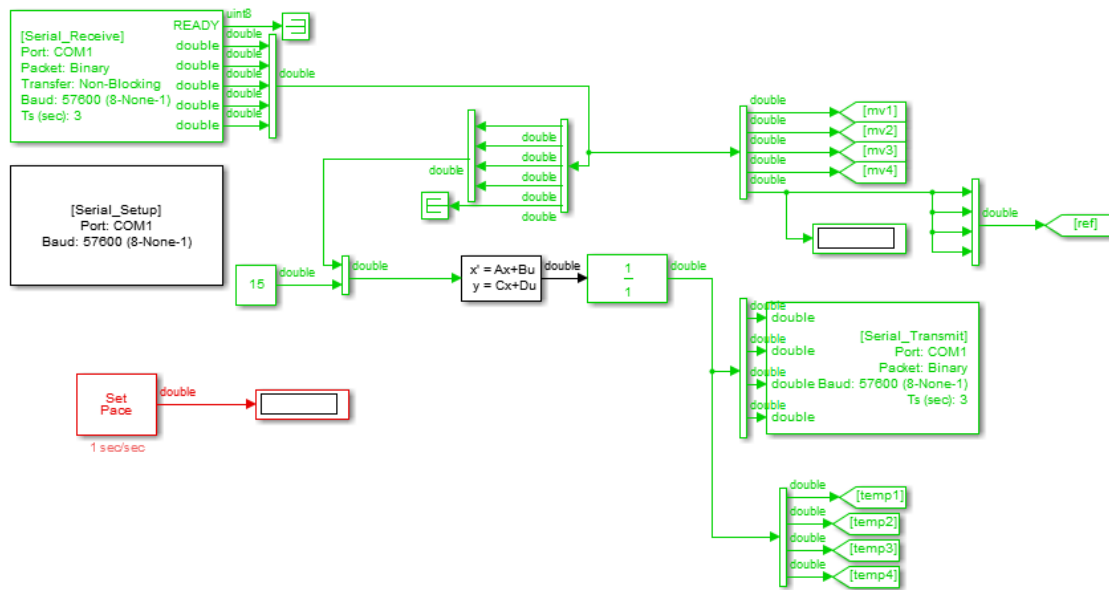


Figura 34. Diagrama de bloques HIL (PC). Recepción HIL

Lo único destacable en el diagrama de la Figura 34 es el el bloque de ‘Set Pace’ y el sistema de comunicaciones del puerto serie utilizado. Al hacer uso del puerto rápido del STM32, la configuración de la comunicación en el ordenador tiene que hacerse con los bloques de Waijung destinados a tal propósito, en vez de los bloques propios de Simulink. Por otro lado, el ‘Set Pace’ sirve para ralentizar la simulación en el ordenador y que se ejecuten las dinámicas de la planta a tiempo real.

De las dos simulaciones que se ejecutan para la verificación del HIL, la primera (Figura 35) no aplica el PWM en los mandos, al contrario que la segunda (Figura 36). Se operó de esta forma para poder modificar la precisión del PWM y comprobar cuánto desvaría de lo que sería el funcionamiento ideal.

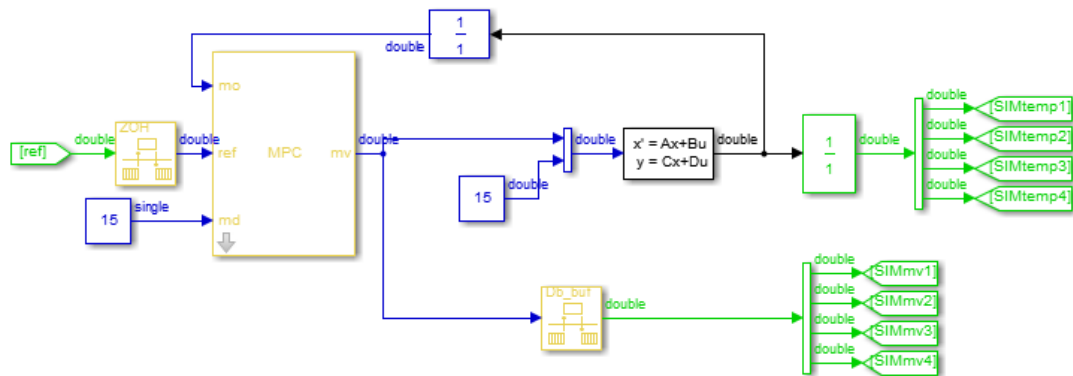


Figura 35. Diagrama de bloques HIL (PC). Simulación MPC sin adaptación PWM

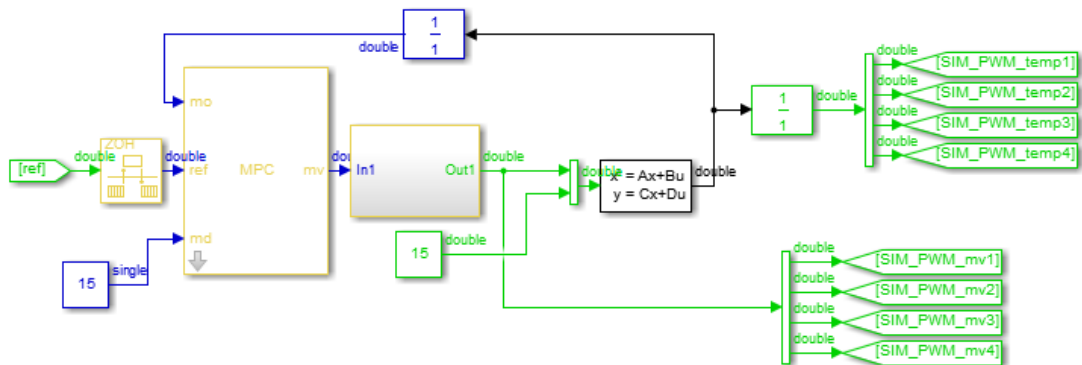


Figura 36. Diagrama de bloques HIL (PC). Simulación MPC con adaptación PWM

En la Figura 37 se observa el sistema de guardado de datos. Los ‘Scopes’ guardan las variables en el workspace de Matlab para su posterior uso y representación.

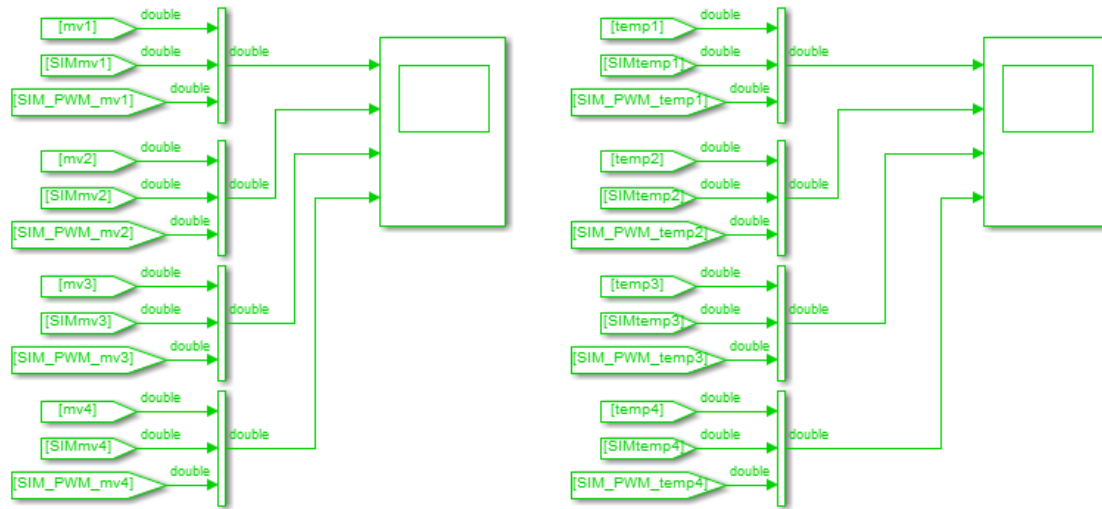


Figura 37. Diagrama de bloques HIL. Representación y captura de los datos

VI.3 SIMULACIÓN HIL

A continuación se muestran las gráficas resultantes de la simulación con HIL (Figura 38 para la habitación 1, Figura 39 para la 2, Figura 40, la 3, y Figura 41 para la cuarta habitación). Que la simulación del HIL comience antes se debe a el tiempo en el que cambia de valor la referencia de temperaturas: el cambio de referencia ocurre cuando el simulador ya ha calculado el mando a aplicar en el periodo de muestreo actual. Hasta que no llega el siguiente periodo de muestreo, no detecta el cambio de referencia.

Puede observarse que los resultados obtenidos con el HIL son muy parecidos a los simulados, sobre todo en las fases finales de la simulación. De esto puede extrapolarse que el microcontrolador STM32 es capaz de ejecutar el control correctamente, y cumplirá con las expectativas en la aplicación final.

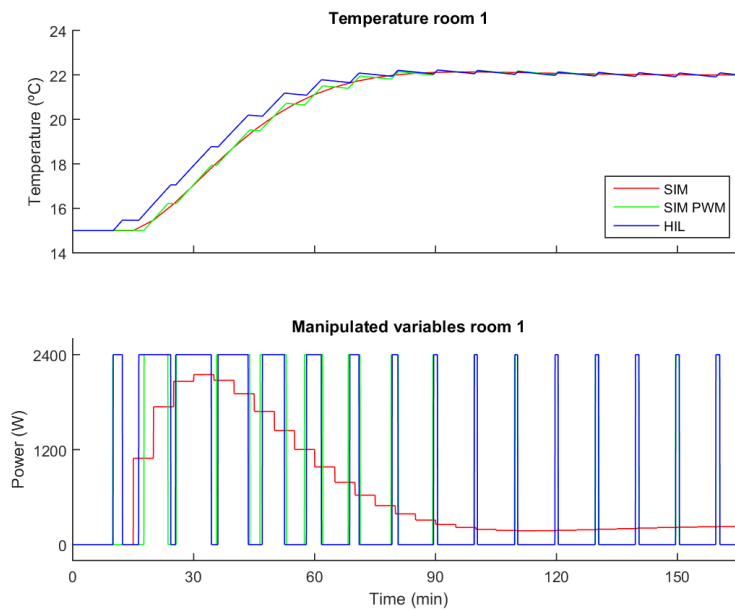


Figura 38. Simulación HIL habitación 1

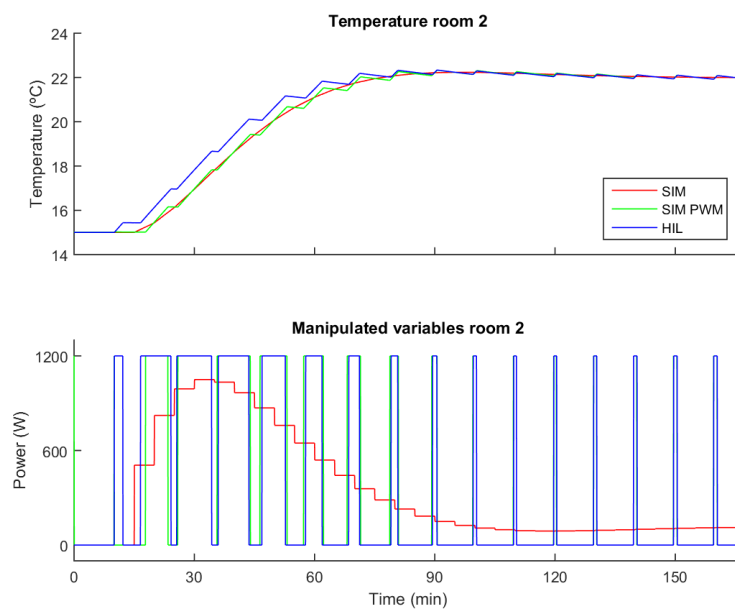


Figura 39. Simulación HIL habitación 2

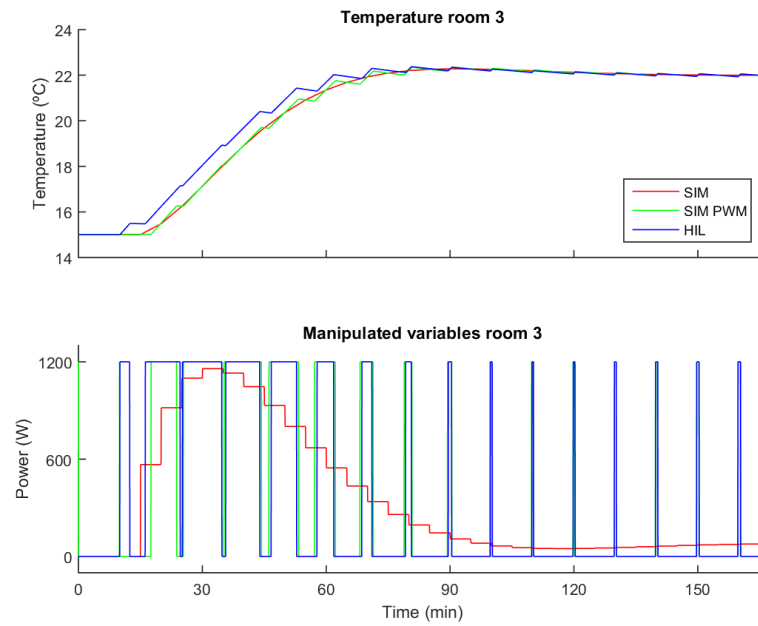


Figura 40. Simulación HIL habitación 3

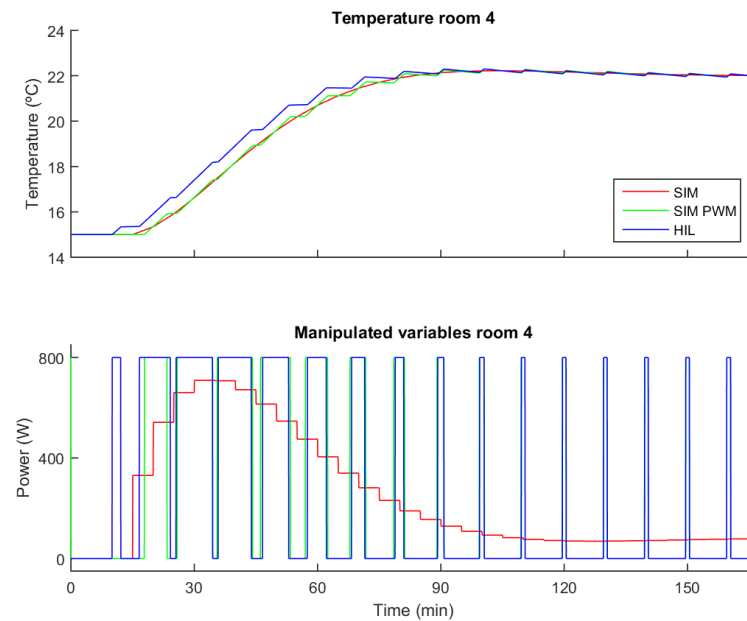


Figura 41. Simulación HIL habitación 4



Capítulo VII: COMUNICACIÓN x10

En este capítulo se aborda el trabajo realizado para gestionar la comunicación x10 directamente desde el STM32. En el apartado VII.1 se explica brevemente el funcionamiento del protocolo, de forma que sea más fácil entender los siguientes apartados. En VII.2 se comenta el hardware propio del protocolo que se ha empleado para conseguir la comunicación. En VII.3 se explica cómo se detecta el paso por cero de la red eléctrica. En el apartado VII.4 se comenta cómo se ha generado una señal auxiliar para que la gestión del protocolo fuese más sencilla. En VII.5 se explica cómo se han generado las tramas de la comunicación. En el apartado VII.6 se introduce la lógica que se ha empleado para evitar la colisión de mensajes. Por último, en VII.7 se muestra un ejemplo del funcionamiento del protocolo mediante el control de dos dispositivos x10.

VII.1 PROTOCOLO x10

El protocolo de comunicaciones x10 emplea la red eléctrica como elemento físico de la comunicación. Concretamente, emplea los pasos por cero (Figura 42) de la señal de red para enviar bits de información. Para ello introduce armónicos a 120 kHz de muy baja amplitud. [11]

Se va a explicar el protocolo para redes eléctricas monofásicas. En el caso de trabajar con redes eléctricas trifásicas, el procedimiento es el mismo, salvo que se repite el mensaje en las tres fases.

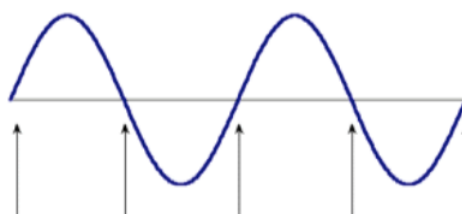


Figura 42. Paso por cero de una senoidal

Los receptores x10, cada vez que detectan un paso por cero, buscan durante el lapso de 0.6 ms la presencia de armónicos de 120 kHz en la red (Figura 43).

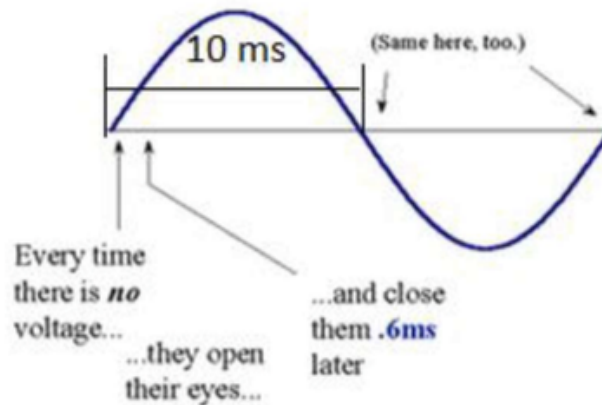


Figura 43. Escucha de los receptores x10.

La codificación de un bit de información requiere de un ciclo de red, ya que para codificar un '1' lógico se envía un pulso de 120 kHz en un paso por cero seguido de una ausencia de pulso en el siguiente (Figura 44).

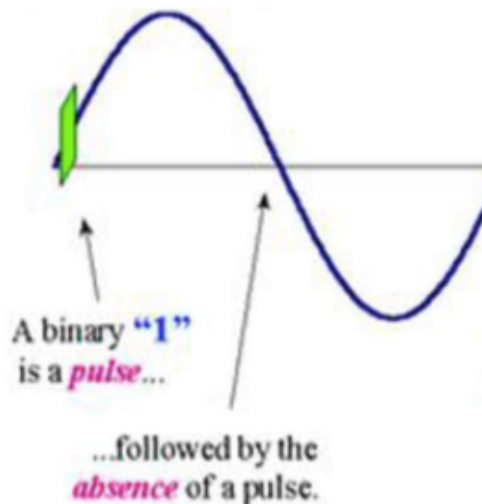


Figura 44. Codificación x10 de un '1' binario

Del mismo modo, para codificar un '0' binario, se envía primero una ausencia de pulso, seguida de un pulso en el siguiente paso por cero, tal y como puede observarse en la Figura 45.

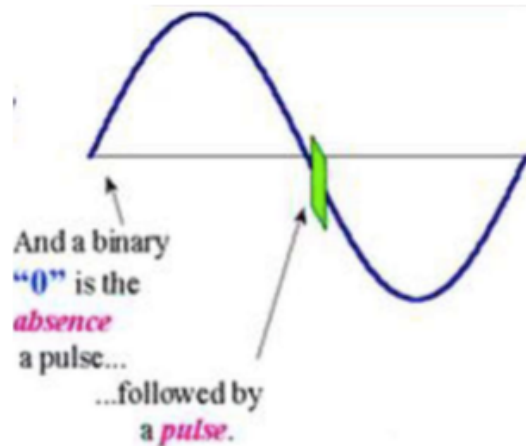


Figura 45. Codificación x10 de un '0' binario

Un mensaje de x10 completo se compone de un código de cabecera (pulso en 3 pasos por cero consecutivos, seguidos de la ausencia de pulso en el siguiente paso por cero '1110'), que puede observarse en la Figura 46; un código de casa (house code) y el comando x10, que bien puede ser la unidad sobre la que se quiere actuar (unit code) o la orden propiamente dicha (command code). Entre dos tramas distintas tiene que haber 3 ciclos de red (6 pasos por cero) de silencio. La excepción a esta regla son los comandos BRIGHT y DIM, que sirven para controlar la intensidad lumínica de las luces, y se envían de forma consecutiva, sin pasos por cero en silencio.

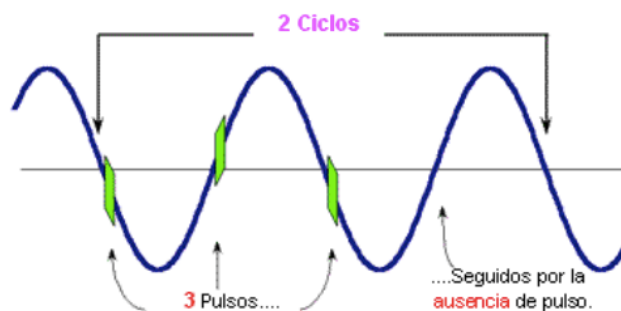


Figura 46. Código de cabecera x10



Códigos x10					
HOUSE		COMMAND		COMMAND	
House	Trama	Unit	Trama	Order	Trama
A	0110	Unit 1	01100	all units OFF	00001
B	1110	Unit 2	11100	all units ON	00011
C	0010	Unit 3	00100	ON	00101
D	1010	Unit 4	10100	OFF	00111
E	0001	Unit 5	00010	DIM	01001
F	1001	Unit 6	10010	BRIGHT	01011
G	0101	Unit 7	01010	All lights OFF	01101
H	1101	Unit 8	11010	Extended code	01111
I	0111	Unit 9	01110	hail request	10001
J	1111	Unit 10	11110	hail ACK	10011
K	0011	Unit 11	00110	pre-set dim	10101
L	1011	Unit 12	10110	extended data	11001
M	0000	Unit 13	00000	status ON	11011
N	1000	Unit 14	10000	status OFF	11101
O	0100	Unit 15	01000	status request	11111
P	1100	Unit 16	11000		

Tabla VIII. Tramas lógicas x10 para cada código

En la Tabla VIII se muestran las tramas lógicas x10 para cada código. Nótese que estas tramas tienen que seguir la lógica previamente indicada para comunicar correctamente (1: pulso seguido de ausencia de pulso; 0 al contrario).

VII.2 HARDWARE x10 EMPLEADO

Para efectuar las pruebas y comprobar el correcto funcionamiento de la comunicación, se empleó el siguiente hardware:

MARMITEK LM 12 LAMP MODULE

El módulo empleado como receptor de las señales x10 es el módulo de luz LM 12, cuya apariencia puede observarse en la Figura 47. Éste módulo en concreto está diseñado para trabajar en corriente alterna de 230 V a 50 Hz, y permite una potencia máxima de 300 W.

Éste módulo puede configurarse como una unidad x10, es decir, se le puede especificar a qué número de casa y unidad debe de responder. Para ello cuenta con dos conmutadores múltiples, con 16 posiciones cada uno, que permiten hacer la selección.



Figura 47. Módulo de luz LM 12 de Marmitek

Fuente: Web

MARMITEK XM10FL TWO WAY TTL INTERFACE

El módulo empleado como transmisor x10 ha sido el XM10FL (Figura 48). Está también diseñado para trabajar con una tensión de red alterna de 230 V a 50 Hz.

El módulo consta de 4 conexiones para su funcionamiento, accesibles a través de un cable RJ11. Éstas conexiones corresponden a GND, el detector de paso por cero de la red (ZC, salida), el pin de datos de entrada, y el de salida. El pin de datos de salida no se ha empleado en éste proyecto, y, por ello, se ha conectado a GND.



Figura 48. Módulo de interfaz x10 XM10FL de Marmitek

Fuente: Web



VII.3 OBTENCIÓN DE LA SEÑAL ZERO - CROSS

Para el correcto funcionamiento del protocolo, es necesario que el microcontrolador sea capaz de detectar los pasos por cero de la senoidal de la red. El interfaz XM10FL de Marmitek dispone de una salida con tal propósito, el ZC.

Es necesario conectar una resistencia de pull-up a la entrada digital del microcontrolador para recibir correctamente la señal. Para ello es suficiente con habilitar la resistencia interna de pull-up que dispone el STM32F4.

En la Figura 49 y la Figura 50 se observa perfectamente la diferencia entre la señal ZC con la presencia de una resistencia de pull-up o sin ella.

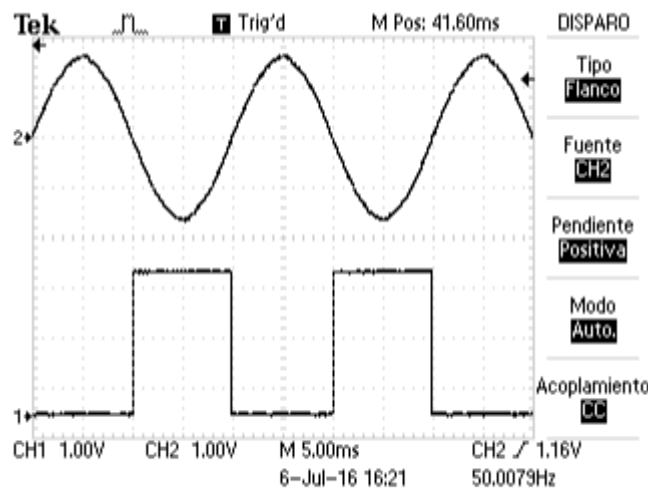


Figura 49: Senoidal red y ZC con resistencia de pull-up

CH1: ZC

CH2: red (x200)

Fuente: Osciloscopio

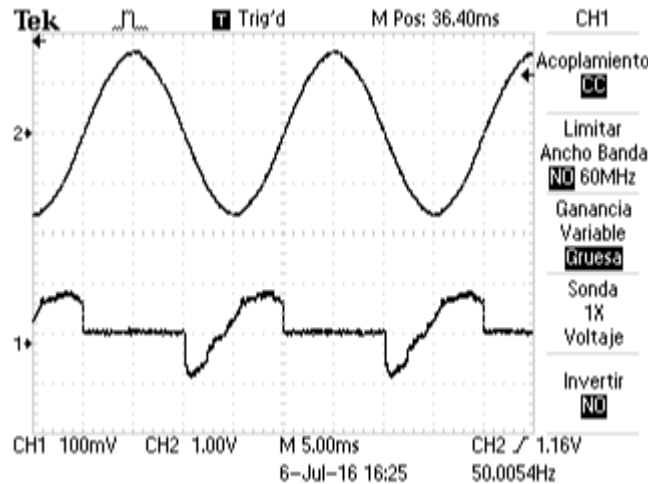


Figura 50: Senoidal red y ZC sin resistencia de pull-up

CH1: ZC

CH2: red (x200)

Fuente: Osciloscopio

Se observa que la señal ZC es digital $\{\sim 3\text{ V}\}$. Los flancos de subida y bajada de ZC coinciden con el paso por cero de la red. Por lo tanto, para detectar un paso por cero de la red, basta con muestrear la señal ZC y detectar sus flancos de subida y bajada.

El muestreo de la señal se efectúa a 10 kHz para poder cumplir con las restricciones de temporización del protocolo x10.

VII.4 GENERACIÓN DE UNA SEÑAL AUXILIAR

Para generar la señal x10, se hará uso de una señal auxiliar que se compondrá de tres pulsos de 900 μs distanciados 3,3 ms entre sí. Estos pulsos tienen que iniciarse a cada paso por cero de la red. A tal efecto, se utiliza un "Resettable subsystem".

En la Figura 51 se muestra el diagrama de bloques que se ha empleado para generar la señal. En la Figura 52 se muestra una comparativa entre la señal ZC y la auxiliar.

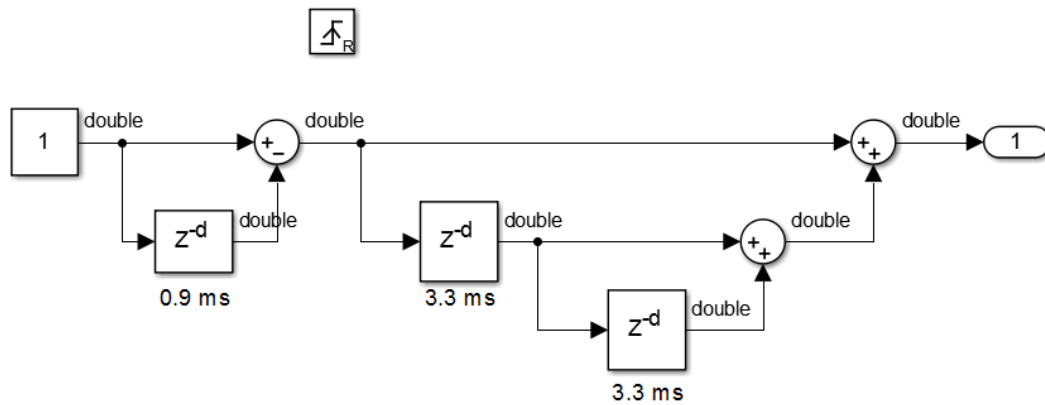


Figura 51: Diagrama para generar la señal auxiliar

Fuente: Simulink screenshot

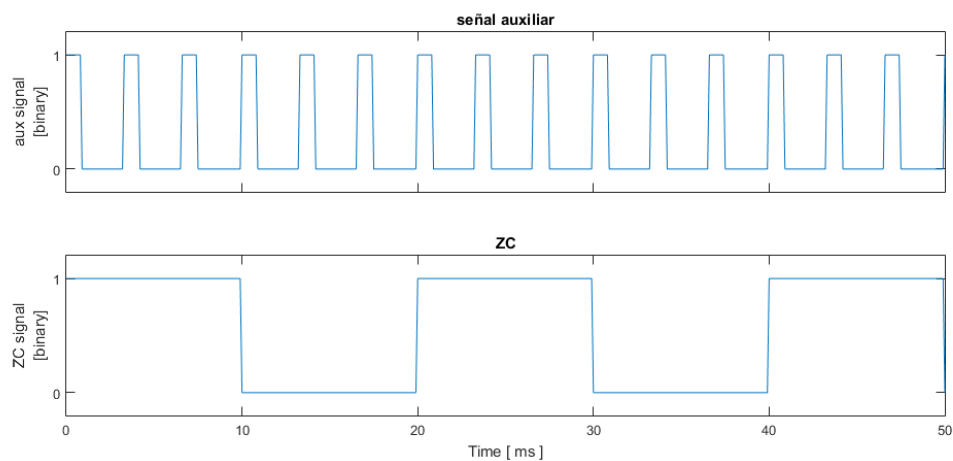


Figura 52: Señal auxiliar y señal ZC

Fuente: Matlab plot from Simulink

VII.5 GENERACIÓN DE LA SEÑAL x10

Para obtener la señal de x10 se ha diseñado un subsistema de Simulink. En la Figura 53 se puede observar la apariencia del subsistema. En la Tabla IX se resumen sus entradas y



salidas, así como el tipo de dato que maneja cada una. En la Tabla X se especifica qué datos tiene que recibir el subsistema para enviar cada código del protocolo.



Figura 53: Subsistema 'x10 signal composer'

Fuente: Simulink subsystem



x10 Signal Composer data			
Tipo de puerto	Puerto	Descripción	Tipo de datos
Entradas	House Code	x10 HOUSE	double {1 - 16}
	Unit/Command	x10 Message	double {1 - 31}
	Zero - cross	señal ZC	boolean
	EN Trans	habilita la transmisión	boolean
	x10 repeats	veces que se repite el mensaje	double
Salidas	x10 Signal	trama x10 a emitir	boolean
	command sent	indica cuándo se ha enviado la totalidad del mensaje	boolean
	Available	NOT USED	-----

Tabla IX: Entradas y salidas del subsistema "x10 Signal Composer"



Entradas HOUSE y COMMAND para el subsistema signal composer					
HOUSE		COMMAND		COMMAND	
House	Input	Unit	Input	Order	Input
A	1	Unit 1	1	all units OFF	17
B	2	Unit 2	2	all units ON	18
C	3	Unit 3	3	ON	19
D	4	Unit 4	4	OFF	20
E	5	Unit 5	5	DIM	21
F	6	Unit 6	6	BRIGHT	22
G	7	Unit 7	7	All lights OFF	23
H	8	Unit 8	8	Extended code	24
I	9	Unit 9	9	hail request	25
J	10	Unit 10	10	hail ACK	26
K	11	Unit 11	11	pre-set dim	27
L	12	Unit 12	12	extended data	28
M	13	Unit 13	13	status ON	29
N	14	Unit 14	14	status OFF	30
O	15	Unit 15	15	status request	31
P	16	Unit 16	16		

Tabla X: Datos de entrada al subsistema 'x10 signal composer'



A continuación se explica cómo funciona el subsistema. En la Figura 54 se muestra la parte del diagrama correspondiente a la selección, a partir de las entradas correspondientes, de la trama x10 a enviar. Para ello, se disponen de 2 matrices, HOUSE, de dimensiones [16x4], que contiene la información lógica de los códigos x10 de las casas {A - P} en sus filas; y X10_DATA, de dimensiones [31x5], que contiene lo propio para los códigos de unidad y comandos x10 (ver Tabla X).

Esas matrices son las entradas de dos bloques 'Select Rows', que seleccionan y filtran las en función de las señales 'HOUSE CODE' y 'UNIT / X10 COMMAND', obteniendo un vector fila en la salida. Cada vector es procesado en un subsistema, encargado de trasponerlo a un vector columna, duplicar éste, negar la señal duplicada e intercalar los elementos de la columna original y la negada uno a uno.

Las dos señales resultantes son combinadas con otras dos columnas, que contienen los códigos de inicio de la trama x10 y los ciclos de espera al final, obteniendo así la señal lógica de la trama x10 a enviar (señal 'x10_DATA').

Por otro lado, se obtiene la señal lógica FADE (TRUE cuando el código a enviar sea 'bright' o 'dim', y FALSE en caso contrario), que será necesaria cuando se desee modificar la intensidad lumínica de alguna luz.

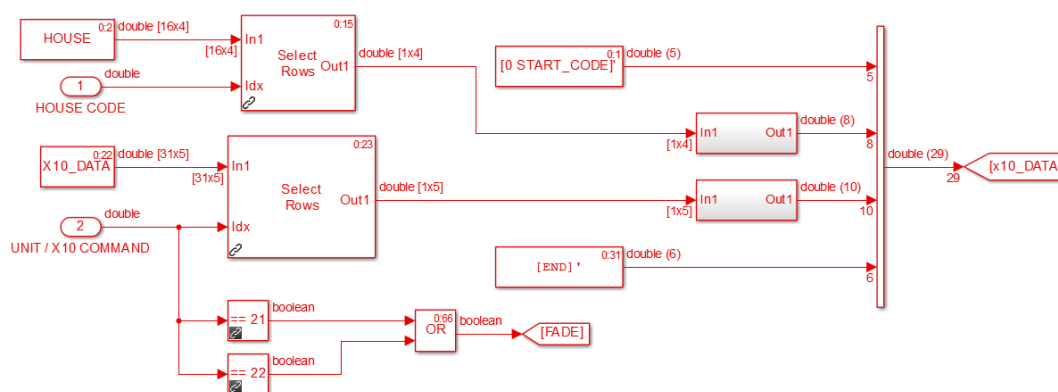


Figura 54: Signal composer. Selección de trama

Fuente: Simulink screenshot

En la Figura 55 se genera la trama x10. Para ello, se emplea un bloque 'Select Rows', que selecciona la señal lógica dependiendo de la salida de un acumulador. Este acumulador incrementa en ' 1 ' su salida a cada flanco de la señal '0-CROSS' (ZC). Además, el acumulador resetea su salida a 1 cuando se detecta un final de trama o un flanco de subida de la señal 'ENABLE TRANSMISSION'.

La señal lógica resultante es multiplicada por la señal auxiliar explicada en VII.4. Con ello se consigue una trama x10 válida para ser enviada ('X10 SIGNAL'). Sólo se enviará la trama cuando la comunicación esté habilitada (señal 'ENABLE').

Por último, cada vez que se llega a un final de trama, se activa la señal 'END_MESSAGE', que servirá para llevar la cuenta de las veces en que se envía la trama.

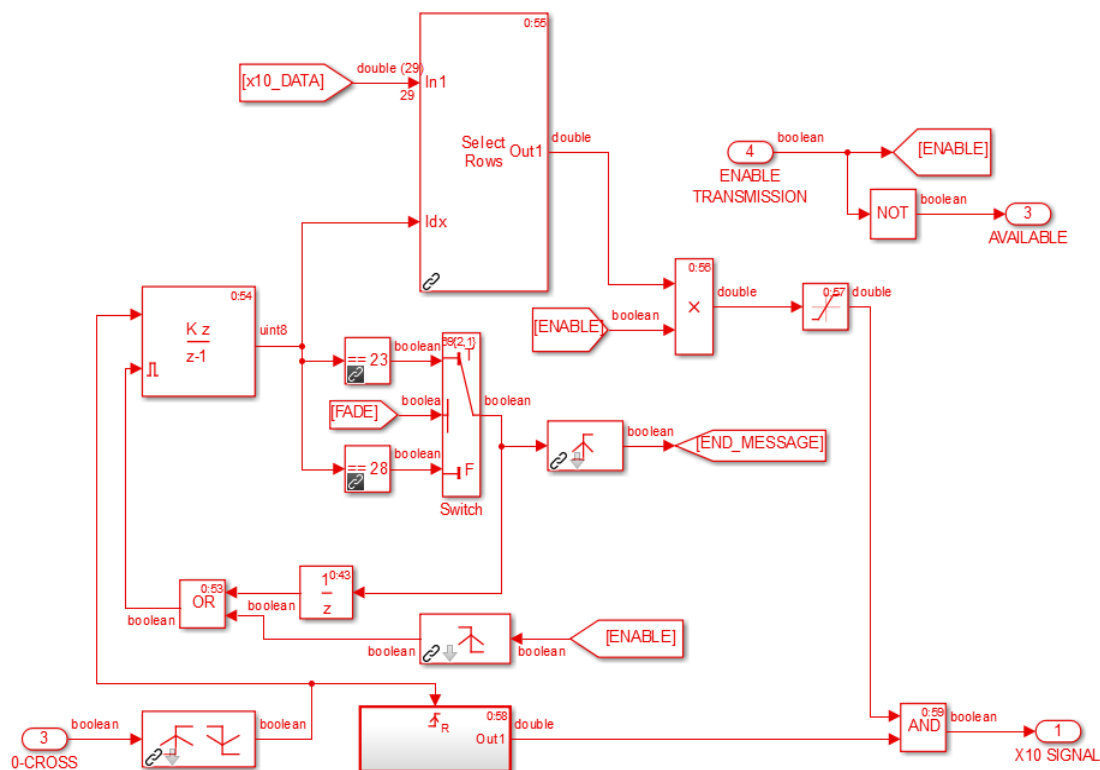


Figura 55: Signal composer. Generación de trama

Fuente: Simulink screenshot

Para terminar con el subsistema 'Signal composer', solo falta indicar cuándo se ha enviado el mensaje completo, con las repeticiones que se hayan especificado. Para ello, se incrementa un acumulador cada vez que se detecta un flanco de subida en la señal 'END_MESSAGE'. Este acumulador se resetea a '0' cuando se detecte un flanco de subida en el 'ENABLE' de la transmisión.

La salida del acumulador se compara con el número de repeticiones especificado en la entrada 'x10 repeats' y, cuando coinciden, se activa la salida 'X10 SENT', indicando a la máquina de estados de la Figura 57 que se ha terminado con la transmisión.

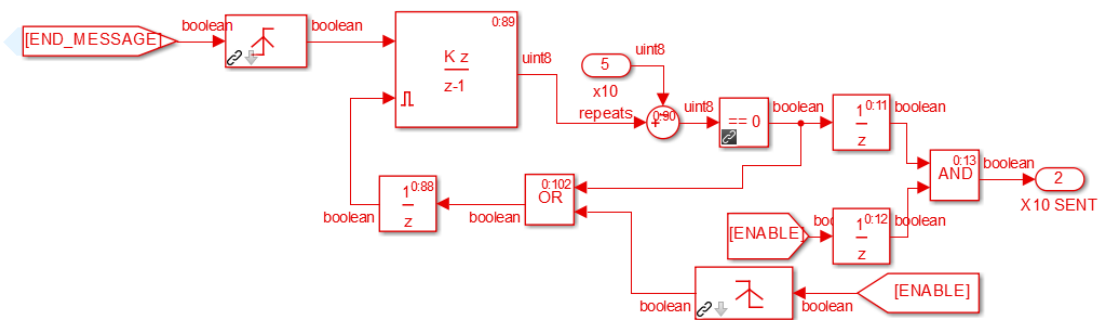


Figura 56: Signal composer. Final de mensaje

Fuente: Simulink screenshot

VII.6 LÓGICA x10. MÁQUINA DE ESTADOS

Para gestionar la lógica del protocolo x10, se ha empleado una máquina de estados, cuya apariencia puede verse en la Figura 57. En la Tabla XI se resumen las entradas y salidas de esta máquina de estados.

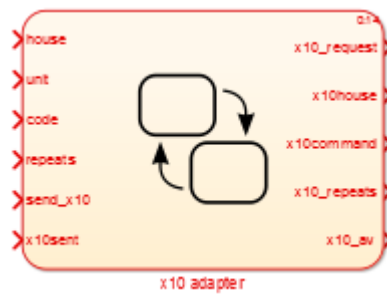


Figura 57: Máquina de estados. Lógica x10

Fuente: Simulink block

Máquina de estados. Lógica x10			
Entradas		Salidas	
house	double {1-16}	x10_request	boolean
unit	double {1-16}	x10house	double {1-16}
code	double {1-15}	x10command	double {1-31}
repeats	double	x10_repeats	double
send_x10	boolean	x10_av	boolean
x10sent	boolean		

Tabla XI. Máquina de estados. Lógica x10. I/O

En la Figura 58 se muestra la máquina de estados propiamente dicha. El funcionamiento de ésta es simple:

1. Espera a recibir una orden de envío.
2. Una vez recibida, ordena enviar los códigos x10 de HOUSE y UNIT y habilita la comunicación. Espera a que se haya enviado el mensaje.
3. Ordena enviar los códigos de HOUSE y el comando deseado. Espera a que se haya enviado el mensaje.
4. Deshabilita la comunicación y vuelve a esperar una orden de envío.

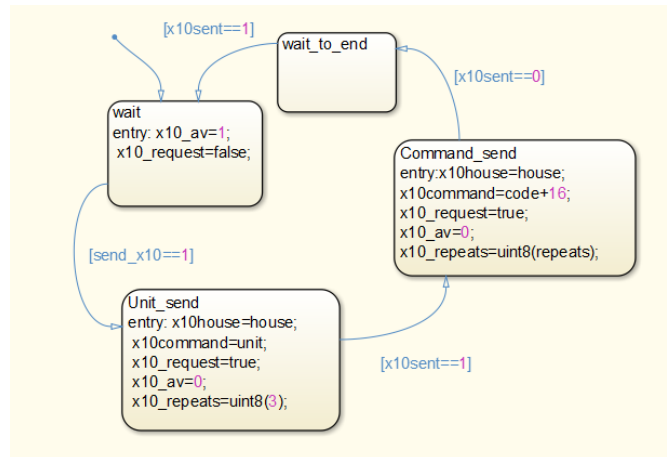


Figura 58: Máquina de estados. Gestión de mensaje x10

Fuente: Simulink screenshot

En la Figura 59 se muestra cómo ha de ser la conexión entre la máquina de estados y el subsistema 'x10 Signal composer' explicado en el apartado VII.5.

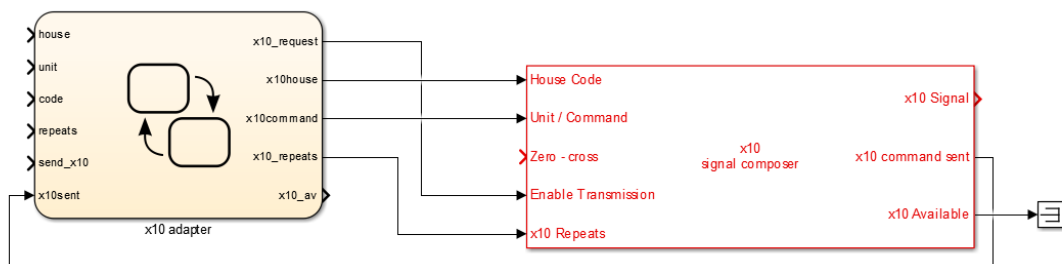


Figura 59: Conexión entre la máquina de estados y el subsistema 'Signal composer'

Fuente: Simulink screenshot

VII.7 ACTUACIÓN x10 SOBRE LOS MANDOS

Para gestionar los mandos (radiadores), se ha diseñado otra máquina de estados, cuya apariencia se observa en la Figura 60. En la Tabla XII se muestra un resumen de las entradas y salidas de la máquina de estados.

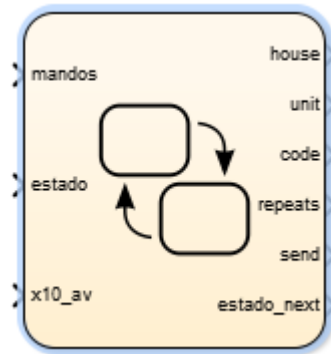


Figura 60: Máquina de estados. Gestión de mandos

Fuente: Simulink block

I/O Máquina de estados. Gestión de mandos					
Entradas		Salidas			
mandos	double(4) {0,1}	house	double{1,16}	repeats	double
estado	double(4) {0,1}	unit	double{1,16}	send	double{0,1}
x10_av	double {0,1}	code	double{1,15}	estado_next	double(4) {0,1}

Tabla XII: Máquina de estados. Gestión x10 de los mandos

En la Figura 61 se puede observar la máquina propiamente dicha. Presenta un funcionamiento cíclico, donde se van comparando uno a uno los mandos con sus respectivos estados. En el caso de ser iguales, pasa a comparar el siguiente mando con su estado. En caso de que difieran, envía el código x10 que corresponda, espera a que se haya enviado y vuelve a empezar a comparar los mandos.

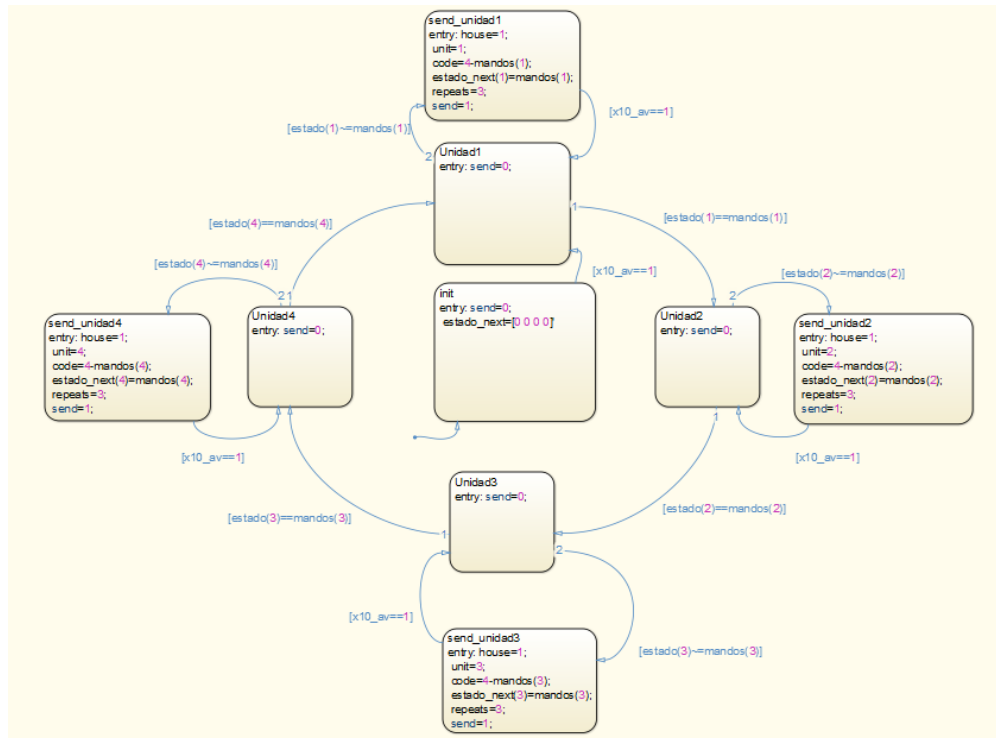


Figura 61: Máquina de estados. Gestión de mandos x10

Fuente: Simulink screenshot

En la Figura 62 se muestra un diagrama que se ha empleado para simular el comportamiento de los mandos. Se hizo uso de sólo dos mandos para facilitar la comprensión. En las figuras siguientes (Figura 64, Figura 65 y Figura 66) se muestra los resultados de la simulación. Se comprueba que funciona correctamente.

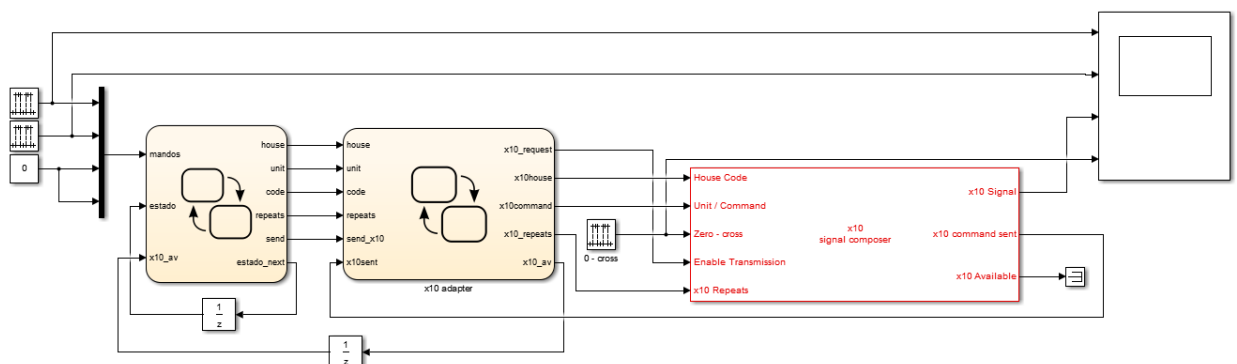


Figura 62: Diagrama de Simulink. Simulación mandos x10



Para comprobar el correcto funcionamiento de la comunicación, se ensayó sobre una luz, para simular de forma visible el comportamiento de los mandos. Para ello se utilizó el 'Lamp Module' de Marmitek. En la Figura 63 se muestra el montaje que se utilizó para comprobar la comunicación.



Figura 63: Montaje para el ensayo de la comunicación

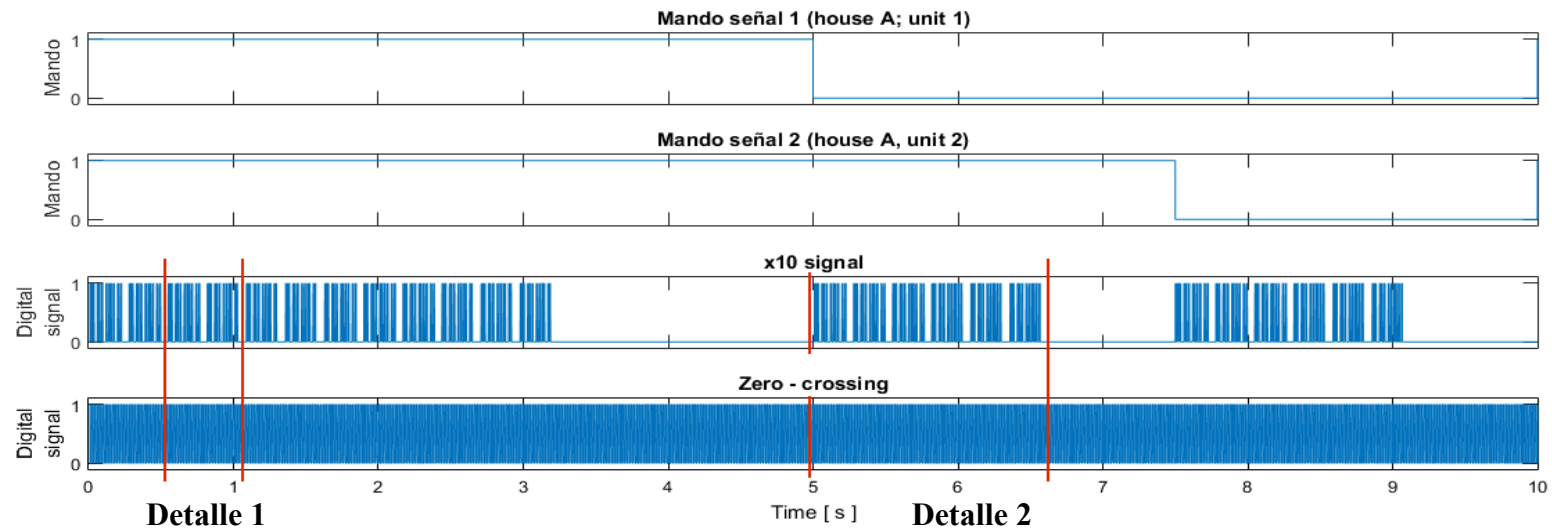


Figura 64: Simulación mensaje x10 con dos mandos

Fuente: Matlab plot from Simulink

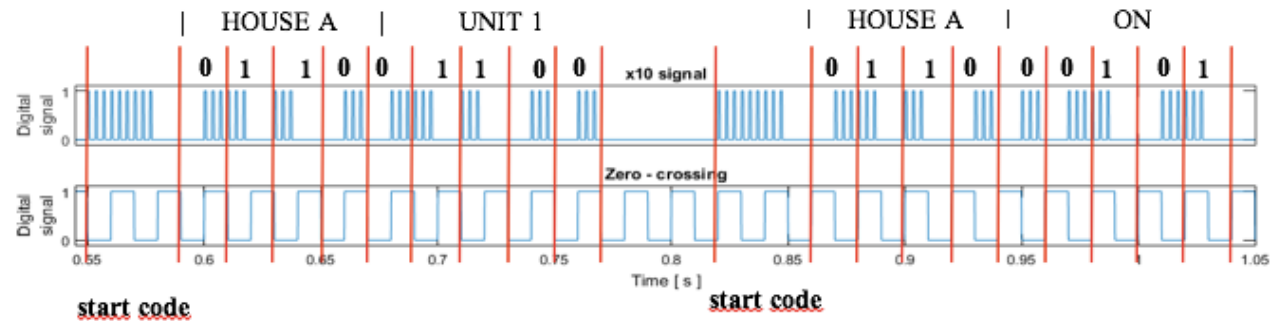


Figura 65: Simulación trama x10. Detalle 1

Fuente: Matlab plot from Simulink

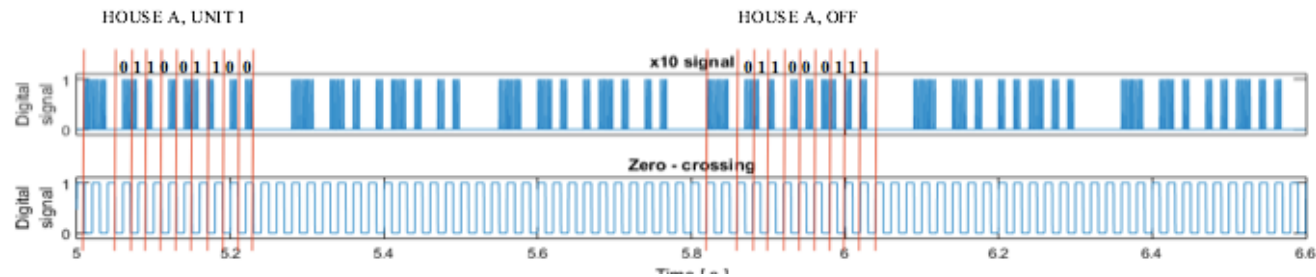


Figura 66: Simulación trama x10. Detalle 2

Fuente: Matlab plot from Simulink



Capítulo VIII: CONCLUSIONES

En este apartado se muestran las principales conclusiones de cada una de las principales tareas del proyecto. En el apartado VIII.1 se comentan las conclusiones derivadas de los controles. En VIII.2, se muestran las conclusiones del HIL. Por último, en el apartado VIII.3 se comentan las de la comunicación x10.

VIII.1 CONTROLES

A partir de las simulaciones que pueden apreciarse en el apartado III.3, se denota que ambos controles simulados requieren de una potencia eléctrica similar, pero el control MPC presenta menos desviaciones de temperatura, y unas constantes de tiempo más rápidas.

VIII.2 SIMULACIÓN CON HIL

La simulación HIL llevada a cabo ha permitido comprobar que el microcontrolador STM32F407 es capaz de ejecutar el control predictivo sin problemas.

VIII.3 COMUNICACIÓN x10

La adaptación de la comunicación x10 al microcontrolador STM32 ha sido exitosa, permitiendo controlar múltiples dispositivos a la vez sin colisionar los mensajes, así como controlar la intensidad lumínica de los módulos adaptados a tal efecto.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA ELECTROMECÁNICA





Capítulo IX: FUTUROS DESARROLLOS

De cara a futuros desarrollos, lo primero que habría de hacerse es ensayar la planta y obtener parámetros más precisos de ésta. A continuación, rediseñar los controles y ya dejar el sistema en funcionamiento.

Otra posible mejora es dotar a la vivienda de un sistema de aire acondicionado, y modificar el software para que se sea capaz de elegir si usar la calefacción o el aire en el control. Para ello, lo más probable es que se necesite diseñar otro control enfocado únicamente al aire acondicionado.

Por último, una vez se tengan claras las conexiones del microcontrolador con el resto de periféricos, se podría diseñar una PCB con el objetivo de disminuir el tamaño del sistema final.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA ELECTROMECÁNICA





Capítulo X: REFERENCIAS Y BIBLIOGRAFÍA

- [1] F. Navarro Pecci, Diseño e implementación de un sistema de control predictivo para una vivienda, Madrid: UP Comillas, 2015.
- [2] K. Tomlinson, «blog.kurttomlinson,» [En línea]. Available: <http://blog.kurttomlinson.com/posts/prbs-pseudo-random-binary-sequence>. [Último acceso: 30 January 2016].
- [3] «National Instruments,» [En línea]. Available: https://zone.ni.com/reference/en-XX/help/372458C-01/lvsysidconcepts/pseudo-random_bin_seq/. [Último acceso: 29 January 2016].
- [4] C. Bordons, Model Predictive Control, Sevilla: Escuela Superior de Ingenieros, 1998.
- [5] P.-D. Morosan, R. Bourdais, D. Dumur y J. Buisson, Distributed model predictive control for building temperature regulation, Baltimore: American Control Conference, 2010.
- [6] A. Bemporad, M. Morari y N. L. Ricker, Model Predictive Control Toolbox. Getting started guide, Mathworks.
- [7] «Mathworks,» Mathworks, [En línea]. Available: <http://es.mathworks.com/help/simulink/block-libraries.html>. [Último acceso: 2 April 2015].



- [8] «Waijung blockset,» Aimagin, [En línea]. Available: <http://waijung.aimagin.com>. [Último acceso: 20 May 2016].
- [9] «National Instruments,» [En línea]. Available: <http://www.ni.com/white-paper/10343/es/>. [Último acceso: 20 May 2016].
- [10] J.-N. Paquin, C. Dufour y J. Belanger, A Hardware-In-the-Loop Simulation Platform for Prototyping and Testing of Wind Generator Controllers, Winnipeg: Conference on Power Systems, 2008.
- [11] J. A. Infantes Diaz, Descripción de X10. Biblioteca de conexión de Arduinos con el protocolo X10, Málaga: Universidad de Málaga, 2009.



Anexo A CÓDIGO SCRIPT OPTIMIZACIÓN.

Param_prep_datos.m

```
clear;
clc;

load('SIM_Mandos.mat');
load('SIM_Temperaturas.mat');

%%
% grid
% disp('Hacer zoom en la zona de interés de la figura')
% disp('Después, pulsar cualquier tecla')
% pause
% [aux1,aux2]=ginput(2);
% aux1=round(aux1);
t_sim = 60*60*24;
t_adq = 3;
sal=[SIM_TEMP_adqPlanta.signals(1).values(:,2)
SIM_TEMP_adqPlanta.signals(2).values(:,2)
SIM_TEMP_adqPlanta.signals(3).values(:,2)
SIM_TEMP_adqPlanta.signals(4).values(:,2)];
%ent=ent-ent(1);
%ent=ent-mean(ent(1:30));
ent=[SIM_MV_adqPlantal.signals(1).values(:,1)
SIM_MV_adqPlantal.signals(2).values(:,1)
SIM_MV_adqPlantal.signals(3).values(:,1)
SIM_MV_adqPlantal.signals(4).values(:,1)];

N=size(ent,1);
time=(0:N-1)*t_adq;
time=time(:);
%sal=sal-sal(1);
%sal=sal-mean(sal(1:30));
% plot(time,[sal ent])
% xlabel('Tiempo (s)')
% ylabel('y(t) y u(t)')
% grid
```



Param_ajuste.m

[illegible]



```
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ALGORITMO DE AJUSTE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% El algoritmo se ejecuta mientras (bucle while) la variacion de
% la funcion objetivo en % es mayor que 'tol1' o el maximo incremento
% en % de los parametros es mayor que 'tol2'.
% Tambien se puede usar como criterio el numero de iteraciones.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

while (100*(V-Vaux)/Vaux>tol1 || 100*max(abs(dgn./theta))>tol2) %&
niter<10

    niter=niter+1; % Se incrementa el numero de iteraciones

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Actualizacion de los parametros
th=theta;
%h0=theta(2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulacion del sistema
%[t,x,ys,ym,error]=sim('sistema',tfin);
sim('param_sistema')
ym=[ymed(:,2);ymed(:,3);ymed(:,4);ymed(:,5)];
ys=[ysim(:,2);ysim(:,3);ysim(:,4);ysim(:,5)];

error=ys-ym;
% Funcion objetivo (error cuadratico medio)
V=sqrt(sum((error).^2)/Nd);
% Construccion de la matriz Jacobiana por diferencias finitas
for i=1:Np
    thaux=theta;
    h=.001*abs(theta(i)); % Incremento para las derivadas
    if abs(theta(i))<10*sqrt(eps)
        h=.01*sqrt(eps);
    end
    thaux(i)=theta(i)+h;
    % Actualizacion de los parametros
    th=thaux;
    % Simulacion
    sim('param_sistema')
    ym=[ymed(:,2);ymed(:,3);ymed(:,4);ymed(:,5)];
    yaux=[ysim(:,2);ysim(:,3);ysim(:,4);ysim(:,5)];
    error=yaux-ym;
    % Matriz de sensibilidades
    J(:,i)=(yaux-ys)/h;
end

% Representacion grafica de 'ys', 'ym' y del error=ys-ym
clf
figure(1);
subplot(4,1,1)
```



```
plot(ymed(:,1),ymed(:,2),'-r',ysim(:,1),ysim(:,2),'-b')
xlabel('Tiempo (s)')
legend('Salida medida','Salida simulada')
grid

subplot(4,1,2);
plot(ymed(:,1),ymed(:,3),'-r',ysim(:,1),ysim(:,3),'-b')
xlabel('Tiempo (s)')
legend('Salida medida','Salida simulada')
grid

subplot(4,1,3);
plot(ymed(:,1),ymed(:,4),'-r',ysim(:,1),ysim(:,4),'-b')
xlabel('Tiempo (s)')
legend('Salida medida','Salida simulada')
grid

subplot(4,1,4);
plot(ymed(:,1),ymed(:,5),'-r',ysim(:,1),ysim(:,5),'-b')
xlabel('Tiempo (s)')
legend('Salida medida','Salida simulada')
grid

% disp('PULSA UNA TECLA PARA CONTINUAR')
% pause

dgn=(J\ (ym-ys))'; % Direccion de Gauss-Newton

% BUSQUEDA DE 'MU' (ajuste del paso en la direccion de Gauss-
% Newton para garantizar una disminucion de la funcion objetivo
mu=2; % Inicializacion de mu
Vaux=V+10; % Inicializacion de la funcion objetivo en el nuevo
punto
stb=1; % Flag de estabilidad del sistema (0-inestable: 1-estable)

% Busqueda de 'mu' mientras no se reduzca la funcion objetivo o
% el sistema sea inestable
thaux=theta;

while Vaux > V %|| any(thaux<0)
    mu=mu/2; % Reduccion del paso
    thaux=theta+mu*dgn; % Actualizacion de parametros

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Actualizacion de los parametros del regulador
    th=thaux;
    %h0=thaux(2);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    % Simulacion con los nuevos parametros

    %[t,x,yaux,ym,error]=sim('sistema',tfin);
    if any(thaux<0)
```



```
Vaux=inf;
else
    sim('param_sistema')
    ym=[ymed(:,2);ymed(:,3);ymed(:,4);ymed(:,5)];
    yaux=[ysim(:,2);ysim(:,3);ysim(:,4);ysim(:,5)];
    error=yaux-ym;
    % Actualizacion de la funcion objetivo
    Vaux=stb*sqrt(sum((yaux-ym).^2)/Nd);
end
end

% Parametros y funcion objetivo en pantalla
theta=thaux;
disp('PARAMETROS')
disp(theta)
disp('FUNCIÓN DE COSTE')
disp(Vaux)

end

% Matriz de covarianzas de los parametros (Sensibilidades)
%P=Vaux^2*inv(J'*J);
%disp('Coeficientes de dispersión')
%disp(100*sqrt(diag(P))./theta)
```



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA ELECTROMECÁNICA





INSTITUTO DE INVESTIGACIÓN
TECNOLÓGICA

PROYECTO DE FIN DE GRADO

PRESUPUESTO

IMPLEMENTACIÓN DE UN CONTROL PREDICTIVO DE TEMPERATURA EN UNA VIVIENDA

Alumno:

GARCÍA AGUILAR, JAVIER

Directores

ZAMORA MACHO, JUAN LUIS

MUÑOZ FRÍAS, JOSÉ DANIEL

Capítulo I: RECURSOS EMPLEADOS

En este capítulo se resumen las cantidades de recursos empleados, tanto materiales, como humanos y software.

I.1 COMPONENTES

<i>Componente</i>	<i>Cantidad</i>
<i>STM32F407VG DISCO</i>	1
<i>aMG USB Converter</i>	1
<i>LM 12 Marmitek Lamp Module</i>	1
<i>XM10FL Marmitek 2 way interface</i>	1
<i>aMG F4 Connect</i>	1
<i>Cable USB – mini USB</i>	2
<i>Cable RJ11</i>	1

I.2 SOFTWARE

<i>Elemento</i>	<i>Cantidad</i>
<i>Licencia Matlab uso académico</i>	1
<i>Licencia Simulink uso académico</i>	1
<i>Microsoft Office</i>	1

I.3 MANO DE OBRA

<i>Elemento</i>	<i>Cantidad (h)</i>
<i>Estudio protocolo x10</i>	5
<i>Desarrollo comunicación x10</i>	100
<i>Pruebas de comunicación x10</i>	10
<i>Estudio controles MPC</i>	15
<i>Diseño de controles</i>	20
<i>Simulación y comprobación de controles</i>	10
<i>Desarrollo simulación HIL</i>	20
<i>Simulaciones HIL</i>	80
<i>Desarrollo script adquisición de datos</i>	5
<i>Simulaciones script adquisición de datos</i>	10
<i>Preparación presentación</i>	10
<i>Redacción de documentos</i>	100
<i>Total</i>	385

Capítulo II: PRECIOS UNITARIOS

II.1 COMPONENTES

<i>Componente</i>	<i>Precio / ud (€)</i>
<i>STM32F407VG DISCO</i>	12.75
<i>aMG USB Converter</i>	31.75
<i>LM 12 Marmitek Lamp Module</i>	26.65
<i>XM10FL Marmitek 2 way interface</i>	40.63
<i>aMG F4 Connect</i>	30
<i>Cable USB – mini USB</i>	2.5
<i>Cable RJ11</i>	9.5

II.2 SOFTWARE

<i>Elemento</i>	<i>Precio unitario (€)</i>
<i>Licencia Matlab uso académico</i>	400
<i>Licencia Simulink uso académico</i>	400
<i>Microsoft Office</i>	350

II.3 MANO DE OBRA

<i>Elemento</i>	<i>Precio unitario (€/h)</i>
<i>Estudio protocolo x10</i>	5
<i>Desarrollo comunicación x10</i>	15
<i>Pruebas de comunicación x10</i>	18
<i>Estudio controles MPC</i>	5
<i>Diseño de controles</i>	15
<i>Simulación y comprobación de controles</i>	17
<i>Desarrollo simulación HIL</i>	15
<i>Simulaciones HIL</i>	20
<i>Desarrollo script adquisición de datos</i>	13
<i>Simulaciones script adquisición de datos</i>	15
<i>Preparación presentación</i>	10
<i>Redacción de documentos</i>	25

Capítulo III: SUMAS PARCIALES

III.1 COMPONENTES

<i>Componente</i>	<i>Precio / ud (€)</i>	<i>Unidades</i>	<i>Total (€)</i>
<i>STM32F407VG DISCO</i>	12.75	1	12.75
<i>aMG USB Converter</i>	31.75	1	31.75
<i>LM 12 Marmitek Lamp Module</i>	26.65	1	26.65
<i>XM10FL Marmitek 2 way interface</i>	40.63	1	40.63
<i>aMG F4 Connect</i>	30	1	30
<i>Cable USB – mini USB</i>	2.5	2	5
<i>Cable RJ11</i>	9.5	1	9.5
<i>TOTAL</i>	---	---	156.28

III.2 SOFTWARE

<i>Elemento</i>	<i>Precio unitario (€)</i>	<i>Unidades</i>	<i>Total (€)</i>
<i>Licencia Matlab uso académico</i>	400	1	400
<i>Licencia Simulink uso académico</i>	400	1	400
<i>Microsoft Office</i>	350	1	350
<i>TOTAL</i>	---	---	1150

III.3 MANO DE OBRA

<i>Elemento</i>	<i>Precio unitario (€/h)</i>	<i>Tiempo (h)</i>	<i>Total (€)</i>
<i>Estudio protocolo x10</i>	5	5	25
<i>Desarrollo comunicación x10</i>	15	100	1500
<i>Pruebas de comunicación x10</i>	18	10	180
<i>Estudio controles MPC</i>	5	15	75
<i>Diseño de controles</i>	15	20	300
<i>Simulación controles</i>	17	10	170
<i>Desarrollo simulación HIL</i>	15	20	300
<i>Simulaciones HIL</i>	20	80	1600
<i>Desarrollo adquisición de datos</i>	13	5	65
<i>Simulaciones adquisición</i>	15	10	150
<i>Preparación presentación</i>	10	10	100
<i>Redacción de documentos</i>	15	100	1500
<i>TOTAL</i>	---	385	5965

Capítulo IV: SUMAS TOTALES

Apartado	Coste (€)
Hardware	156.28
Software	1150
Mano de obra	5965
PEM	7 271.28
Gastos generales (10%)	727.13
I.V.A (21%)	1 527
PEC	9 525.4

