



UNIVERSIDAD FRAY LUCA PACCIOLI

CON RECONOCIMIENTO DE VALIDEZ OFICIAL DE
ESTUDIOS DE LA SECRETARÍA DE EDUCACIÓN PÚBLICA
SEGÚN ACUERDO 2008PO2375 CON FECHA DEL 05 DE DICIEMBRE DEL 2008

TESIS

“DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE SISTEMA DOMÓTICO
PARA EL CONTROL DE UNA LUMINARIA POR MEDIO DE UNA COMPUTADORA
O DISPOSITIVO MÓVIL UTILIZANDO LA RED ELÉCTRICA”

QUE PARA OBTENER EL TÍTULO DE
INGENIERO EN MECATRÓNICA

PRESENTA

RAFAEL ABAD PÉREZ RENDÓN

DIRECTOR DE TESIS: MTRO. JUAN MANUEL RODRÍGUEZ GONZÁLEZ

CUERNAVACA, MORELOS

20/10/2014

Contenido

Resumen

Abstract

Introducción..... I

Problemática..... I

Hipótesis II

Justificación II

Objetivo General..... III

Objetivos Particulares III

Alcance III

Limitaciones IV

Capítulo 1 Marco Teórico..... 1

 1.1 Antecedentes 2

 1.2 Estado del Arte 2

 1.2.1 Sistema “My home” de BTicino..... 3

 1.2.2 Sistema “Myhome” de Control4 3

 1.2.3 Proyecto Ubi 6

 1.3 Servicios 8

 1.3.1 Ahorro energético 8

 1.3.2 Confort 8

 1.3.3 Seguridad 9

 1.3.4 Comunicaciones 9

 1.3.5 Plusvalía 10

 1.4 Medios de Transmisión 10

| | |
|---|-----------|
| 1.4.1 Cableados | 10 |
| 1.4.2 Inalámbricos | 11 |
| 1.5 Protocolo X-10 | 13 |
| 1.5.1 Características..... | 13 |
| 1.5.2 Nivel Físico | 17 |
| 1.5.3 Comunicación..... | 18 |
| 1.5.4 Módulos de Potencia | 21 |
| 1.5.6 Limitaciones..... | 23 |
| 1.6 Microcontrolador Arduino | 23 |
| 1.6.1 Entorno de Desarrollo Arduino | 24 |
| 1.6.2 Gama de Arduinos..... | 25 |
| 1.7 Raspberry Pi modelo B | 26 |
| 1.7.1 Especificaciones de la computadora Raspberry Pi modelo B..... | 27 |
| 1.7.2 Sistema Operativo | 27 |
| 1.7.3 Dimensiones | 28 |
| 1.8 Python..... | 29 |
| 1.9 Django..... | 30 |
| 1.10 Firmata..... | 31 |
| Capítulo 2 Diseño del Sistema Domótico..... | 32 |
| 2.1 Modelo del Sistema..... | 33 |
| 2.1.1 Servidor | 34 |
| 2.1.2 Interfaz..... | 35 |
| 2.1.3 Actuadores | 35 |
| 2.2 Software..... | 35 |
| 2.2.1 Aplicación Web | 35 |

| | |
|---|-----------|
| 2.2.2 Arduino | 41 |
| 2.3 Hardware | 42 |
| 2.3.1 Módulo PSC05 | 42 |
| 2.3.2 Módulo para lámpara LM465..... | 43 |
| 2.3.3 Conexión Arduino-PSC05..... | 44 |
| 2.3.4 Diagrama Electrónico del Sistema Domótico..... | 45 |
| Capítulo 3 Integración y Pruebas..... | 46 |
| 3.1 Instalación del Software en Raspberry Pi | 47 |
| 3.1.1 Sistema Operativo | 47 |
| 3.1.2 Instalación del entorno de programación Arduino | 49 |
| 3.1.3 Instalación de Django y complementos de Python | 50 |
| 3.2 Prueba del sistema | 51 |
| Conclusiones..... | 59 |
| Recomendaciones..... | 60 |
| Bibliografía..... | 62 |
| Mesografía | 63 |
| Anexos | 68 |
| Anexo A | 69 |
| Anexo B | 70 |
| Anexo C Tabla comparativa de precios | 71 |

Índice de Ilustraciones

| | |
|--|----|
| Ilustración 1. Mando Touch Screen de pared y comando de pared, escenarios y difusión sonora..... | 3 |
| Ilustración 2. Sistema Myhome en diferentes equipos inteligentes | 4 |
| Ilustración 3. Diferentes controladores Control4..... | 4 |
| Ilustración 4. Interruptores y enchufes Control4 | 5 |
| Ilustración 5. Parlantes, Amplificadores, Docks de iPod Control4 | 5 |
| Ilustración 6. Control de temperatura | 6 |
| Ilustración 7. Cerraduras electrónicas | 6 |
| Ilustración 8. Ubi conectada a la corriente de la casa..... | 7 |
| Ilustración 9. Características de Ubi. | 7 |
| Ilustración 10. Logos de identificación X10 | 16 |
| Ilustración 11. Ejemplo de un Sistema X10..... | 16 |
| Ilustración 12. Colocación de las señales de información en la red C.A. | 17 |
| Ilustración 13. Puntos donde se comprueba si existe la señal de 120KHz | 18 |
| Ilustración 14. Forma de enviar la información con el protocolo X10 a la línea eléctrica | 20 |
| Ilustración 15. Transmisión estándar X10 | 21 |
| Ilustración 16. Módulos de X10 | 22 |
| Ilustración 17. Entorno de programación Arduino | 24 |
| Ilustración 18. Arduino Mega 2560..... | 26 |
| Ilustración 19. Raspberry Pi Modelo B | 27 |
| Ilustración 20. Distribución de Elementos Raspberry Pi (Raspberry Pi, 2012)..... | 28 |
| Ilustración 21. Diagrama a bloques del Sistema Domótico. | 33 |
| Ilustración 22. Imagen del Servidor (Raspberry Pi, modelo B)..... | 34 |
| Ilustración 23. Clase “ <i>Poll</i> ”..... | 36 |
| Ilustración 24. Clase "Choice" | 36 |
| Ilustración 25. Views.py..... | 37 |
| Ilustración 26. Entrando al sistema (index.html)..... | 37 |
| Ilustración 27. DetailView (detail.html) | 38 |
| Ilustración 28. ResultsView (results.html)..... | 38 |

| | |
|--|----|
| Ilustración 29. Función "submit" | 39 |
| Ilustración 30.Urls.py | 39 |
| Ilustración 31. Modelo-Vista-Controlador (MVC) | 40 |
| Ilustración 32. Diseño de la plantilla HTML del sistema | 40 |
| Ilustración 33. Código escrito en lenguaje Arduino | 42 |
| Ilustración 34. Módulo PSC05 | 43 |
| Ilustración 35. Módulo LM465 | 43 |
| Ilustración 36. Lámpara conectada al LM465..... | 43 |
| Ilustración 37. Diagrama de conexión | 44 |
| Ilustración 38. Arduino con los hilos del cable telefónico conectados | 44 |
| Ilustración 39. Diagrama de Conexión del Sistema Domótico..... | 45 |
| Ilustración 40. Espacio para tarjeta SD | 47 |
| Ilustración 41. Menú de selección de S.O. | 48 |
| Ilustración 42. Menú de configuración Raspbian..... | 49 |
| Ilustración 43. Entorno Arduino instalado en Raspbian..... | 49 |
| Ilustración 44. Directorio del proyecto | 51 |
| Ilustración 45. Conexión Arduino – Transmisor PSC05 | 52 |
| Ilustración 46. Módulo PSC05 con cable telefónico | 52 |
| Ilustración 47. Lámpara y configuración del módulo LM465 | 53 |
| Ilustración 48. Raspberry Pi conectada como servidor | 53 |
| Ilustración 49. Fuente de alimentación y cable USB - micro USB | 54 |
| Ilustración 50. Raspberry Pi y Arduino conectados a un televisor digital con entrada HDMI..... | 54 |
| Ilustración 51. Ruta seguida para obtener el programa "StandardFirmata"..... | 55 |
| Ilustración 52. Programa en Arduino con Firmata y X10 cargado a la placa | 56 |
| Ilustración 53. Servidor ejecutándose en la terminal de Raspbian | 56 |
| Ilustración 54. Vista desde teléfono inteligente | 57 |
| Ilustración 55. Pantalla de selección | 57 |
| Ilustración 56. Selección y registro de la opción elegida..... | 58 |
| Ilustración 57. Diagrama Esquemático Arduino Mega 2560..... | 69 |
| Ilustración 58. Diagrama Esquemático PSC05..... | 70 |

Índice de Tablas

| | |
|--|----|
| Tabla 1. Modelos de Arduino y sus características. | 25 |
| Tabla 2. Características generales del Arduino Mega 2560. | 26 |
| Tabla 3. Tabla Comparativa de precios..... | 71 |

Resumen

En esta tesis se presenta el diseño y construcción de un sistema **domótico**, para controlar el encendido y apagado de luminarias en una casa habitación por medio de algún dispositivo móvil que incorpore red WiFi, como puede ser una Tablet o un teléfono inteligente y usando el protocolo de comunicación **X10**.

Para lograr lo anterior, se utilizaron dispositivos con tecnologías actuales, como un microcontrolador **Arduino** para establecer comunicación X10 con los módulos de transmisión y recepción utilizando como medio, la red eléctrica. Es decir, se utilizaron los mismos cables de la red eléctrica para controlar remotamente, en este caso, una lámpara.

También se utilizó una computadora “**Raspberry Pi**” que funciona como servidor para permitir que los dispositivos móviles se puedan conectar y manipular este sistema domótico.

Palabras clave: **Domótica, Raspberry Pi, Arduino, X10.**

Abstract

The design and construction of a home automation system is presented in this thesis, to control the on and off of lights in a house through any mobile device that incorporates WiFi technology, such as a tablet or a mobile phone and also the X10 communication protocol is used in this project.

To achieve this, devices with existing technologies were used, like an Arduino microcontroller for communication with X10 transmitter and receiver modules, using the electrical grid. That is, the same wires of the electrical network were used to remotely control, in this case, a lamp.

A computer "Raspberry Pi" was enabled as a server to allow mobile devices were able to connect and manipulate the system.

Keywords: **Automated, Raspberry Pi, Arduino, X10.**

Introducción

En el primer capítulo se define el concepto de domótica, tomando términos de diferentes autores; además, se hace un pequeño recorrido por las características que componen un sistema domótico, una descripción del protocolo de comunicación X10 y se brinda información acerca de los elementos que vendrán en el prototipo. También se realiza una pequeña investigación de los sistemas que existen en el mercado.

El segundo capítulo habla sobre el diseño y describe los elementos que componen el sistema. Se explica la manera en cómo están realizados los programas que conforman la interfaz web y se muestra un diagrama de todo el sistema.

En el tercer capítulo se habla acerca de la instalación del sistema operativo del servidor, también la instalación de los programas necesarios para hacer funcionar el sistema en el servidor y sobre la manera en que se realizaron las pruebas. Después de este capítulo se realizan conclusiones y se emiten recomendaciones para mejorar el sistema domótico.

Finalmente se presentan dos anexos que muestran diagramas sobre la manera en que están construidos el microcontrolador Arduino y los módulos comerciales X10 y un tercer anexo que muestra una tabla comparativa de los costos de este proyecto y los diferentes sistemas del mercado que se investigaron.

Problemática

Conforme la edad de una persona sigue avanzando, las capacidades físicas van disminuyendo paulatinamente, cada vez se vuelve más difícil levantarse del asiento y desplazarse a ciertos lugares dentro del hogar.

Pero no solamente los adultos mayores tienen dificultades al realizar ciertas actividades, también están las personas con capacidades diferentes y las que han sufrido un accidente, además de las personas que padecen algún tipo de enfermedad degenerativa.

La artrosis de rodilla es una enfermedad degenerativa que se caracteriza por el deterioro paulatino de la articulación de la rodilla. Es una de las principales causas de incapacidad en Estados Unidos (Centers for Disease Control and Prevention). El desarrollo de la enfermedad es lento y el dolor empeora con el paso del tiempo.

Debido a que esta enfermedad limita el movimiento en las personas que lo padecen. Por ejemplo: el simple hecho de caminar para “prender” o “apagar” una luminaria o foco, resulta en una problemática que necesita solución.

¿De qué forma se podría desarrollar un sistema que sea utilizado por personas que padecen artrosis de rodilla y ayudarlos a activar o desactivar las luminarias en una casa sin la necesidad de levantarse de su lugar de reposo?

Hipótesis

Se podría realizar el sistema mediante la utilización de un microcontrolador para comunicar señales de control a dos módulos “X10” (transmisor y receptor) que estarán conectados a una lámpara por medio de la línea eléctrica de la casa.

También se utilizaría una mini computadora que podría funcionar como servidor para permitir que dispositivos móviles controlen remotamente el sistema.

Justificación

El proyecto pretende mejorar la calidad de vida de una persona con artrosis de rodilla, que tiene la dificultad para trasladarse de un lugar a otro a causa del dolor en su articulación. En México en una muestra de 2500 individuos se encontró una prevalencia de artrosis del 2.3% en la población adulta (Cardiel, 2002). En cuanto a su impacto, la artrosis en el Instituto Mexicano del Seguro Social (IMSS) constituye uno de los 10 principales motivos de consulta al médico familiar (DTIES, 2007).

A los pacientes de esta enfermedad se les indica que deben tener mucho reposo pero hay casos en los que estas personas se encuentran solas y tienen que interrumpir su tratamiento para manipular elementos a su alrededor como por

ejemplo las luces de su vivienda, que llegando cierta hora se deben encender y podría ser causa de un accidente.

Con el sistema propuesto se permitirá el manejo del encendido y apagado de luminarias, gracias a que cualquier dispositivo capaz de interpretar código HTML (Computadora, Smartphone, Tablet, etc.) se podrá utilizar como controlador remoto.

Como principal beneficio del proyecto se encuentra la poca inversión económica que se tendría que hacer en la adquisición de los diversos elementos tecnológicos, además, se pretende que ocupe poco espacio, y que sea de fácil instalación, manejo y mantenimiento. Dependiendo de las necesidades del usuario final, el proyecto se puede ampliar no solamente a luminarias, también a otros electrodomésticos que utilicen la red eléctrica prácticamente sin ninguna modificación, resultando versátil.

Objetivo General

Diseñar un sistema domótico que utilice el protocolo X10 para controlar el encendido y apagado de luminarias en una casa habitación, por medio de una computadora o dispositivo WiFi, conectado a la red local de la vivienda.

Objetivos Particulares

- Comprender el protocolo X10 (Instalación, operación, hardware necesario, etc.).
- Utilizar los aparatos necesarios para un entorno de comunicación X10 entre los dispositivos de transmisión y recepción colocados en la red eléctrica.
- Seleccionar una computadora que controle todo el sistema domótico, con bajo costo de adquisición.
- Programar una aplicación con los elementos necesarios para que el usuario pueda manipular el sistema.

Alcance

Se diseñará el sistema domótico y posteriormente se construirá un prototipo para el control de una luminaria, utilizando para ello, los módulos comerciales

(transmisor y receptor) X10. Se podrán controlar más luminarias simplemente adquiriendo y colocando otro módulo receptor X10.

Se habilitará una computadora Raspberry Pi como servidor (utilizando programas libres enfocados al ambiente Linux), la cual se encargará de enviar los comandos de encendido y apagado al sistema de domótico.

Se podrá utilizar cualquier dispositivo móvil (cuyo navegador interprete código HTML) como dispositivo de control, es decir, tabletas, teléfonos inteligentes, etc.

Limitaciones

- Para este prototipo solo se controlará una lámpara.
- Los focos deben ser incandescentes con una potencia máxima de 300 W.
- El transmisor X10 debe estar en la misma línea o fase eléctrica que el receptor X10.
- La tensión de la red eléctrica debe ser de 120V nominales.

CAPÍTULO 1

MARCO TEÓRICO

1.1 Antecedentes

En el diccionario de la real Academia Española se indica que la palabra domótica proviene del latín *domus* – casa – y del término informática, siendo “*el conjunto de sistemas que automatizan las diferentes instalaciones de la vivienda*” (Solánis, 2005, pág. 6). También se puede definir como “*la integración de las nuevas tecnologías y el diseño en los espacios habitables, a fin de obtener una mayor funcionalidad y confort*” (Maestro, 2009, pág. 2).

Para los autores J. M. Huidobro y Ramón Millán, la vivienda domótica es:

Aquella que integra un conjunto de automatismos en materia de electricidad, electrónica, robótica, informática y telecomunicaciones, con el objetivo de asegurar al usuario un aumento de confort, la seguridad, el ahorro energético, las facilidades de comunicación y las posibilidades de entretenimiento (Martín & Sáez, 2006, pág. 16).

El origen de la domótica se remonta a los años setenta, cuando en Estados Unidos aparecieron los primeros dispositivos de automatización de edificios basados en la aún hoy exitosa tecnología X10. Estas incursiones primerizas alternaron con la llegada de nuevos sistemas de calefacción y climatización orientados al ahorro de energía. Los primeros equipos comerciales se limitaban a la colocación de sensores y termostatos que regulaban la temperatura ambiente. La disponibilidad y proliferación de la electrónica de bajo costo favoreció la expansión de este tipo de sistemas, despertando así el interés de la comunidad internacional por la búsqueda de la casa ideal (Martín & Sáez, 2006)

1.2 Estado del Arte

Todo parece indicar que no se está lejos de hacer real lo que mostraban las series de ciencia ficción, ya que hoy en día es posible que la casa responda directamente a las órdenes de nuestra voz. Incluso podrá dialogar con nosotros y hasta aprender de nuestros gustos y necesidades.

Los inmuebles se van volviendo más inteligentes al incorporar otras tecnologías como la telefonía IP (protocolo de Internet) o el BMS (sistema de gerenciamiento de edificios) que se utiliza en los sistemas HVAC (Heating, Ventilating, Air Conditioning, por sus siglas en inglés) que en español significa calefacción, ventilación y aire acondicionado.

1.2.1 Sistema “My home” de BTicino.

Supera el concepto tradicional de “instalación eléctrica”, que está pensada como simple instalación de interruptores y toma corrientes.

La domótica de BTicino (Legrand, 2014) ha sido pensada para integrar las funciones de automatización, termorregulación, difusión sonora, etc. (ver Ilustración 1) de forma inmediata o en el transcurso del tiempo, según las necesidades y el presupuesto (un simple toma corriente puede costar hasta \$3 mil pesos mexicanos). (Elektron, 2014)



Ilustración 1. Mando Touch Screen de pared y comando de pared, escenarios y difusión sonora.

1.2.2 Sistema “Myhome” de Control4

Control4 es una empresa dedicada a la automatización y soluciones de control personalizadas; con sus productos permiten controlar prácticamente todos los dispositivos del hogar o negocio, además ofrecen soluciones para controlar

iluminación, reproducción de música, videos y seguridad en cualquier parte de la casa, a través de sus aplicaciones para computadora, teléfonos inteligentes, televisores y tabletas (ver Ilustración 2).



Ilustración 2. Sistema Myhome en diferentes equipos inteligentes

1.2.2.1 Productos Control4

- Controladores: Son el centro integral de cualquier sistema Control4, permite agregar dispositivos a controlar dependiendo de las necesidades del usuario (ver Ilustración 3).



Ilustración 3. Diferentes controladores Control4

- Iluminación: los atenuadores e interruptores de Control4 reemplazan fácilmente los interruptores del hogar; también se pueden enchufar

luminarias y equipos electrónicos sin tener que hacer modificaciones al hogar (ver Ilustración 4).



Ilustración 4. Interruptores y enchufes Control4

- Audio/Video: Control4 ofrece productos para ambientar musicalmente todas las habitaciones (ver Ilustración 5).



Ilustración 5. Parlantes, Amplificadores, Docks de iPod Control4

- Temperatura: Permite controlar la temperatura ambiente de las diferentes habitaciones del hogar (ver Ilustración 6).



Ilustración 6. Control de temperatura

- Seguridad: Control4 ofrece cerraduras que se instalan en minutos y que son fáciles de controlar desde el hogar o desde cualquier conexión a internet (ver Ilustración 7).



Ilustración 7. Cerraduras electrónicas

1.2.3 Proyecto Ubi

Ubi es una mini computadora que se activa por voz, y solo se tiene que conectar a un tomacorriente en la pared (ver Ilustración 8) y de inmediato accede a internet a través de una conexión WiFi (Team Ubi, 2012).



Ilustración 8. Ubi conectada a la corriente de la casa.

Entre las aplicaciones más destacadas se encuentran las siguientes:

- Búsqueda en internet mediante voz.
- Luces indicadoras que van cambiando dependiendo de los eventos que ocurren (cambio de clima, email nuevo, etc.).
- Asistente virtual (calendario, lector, radio).
- Sistema de intercomunicación.
- Monitor para bebés.
- Control del clima.

En la Ilustración 9, se pueden observar algunas de sus características.

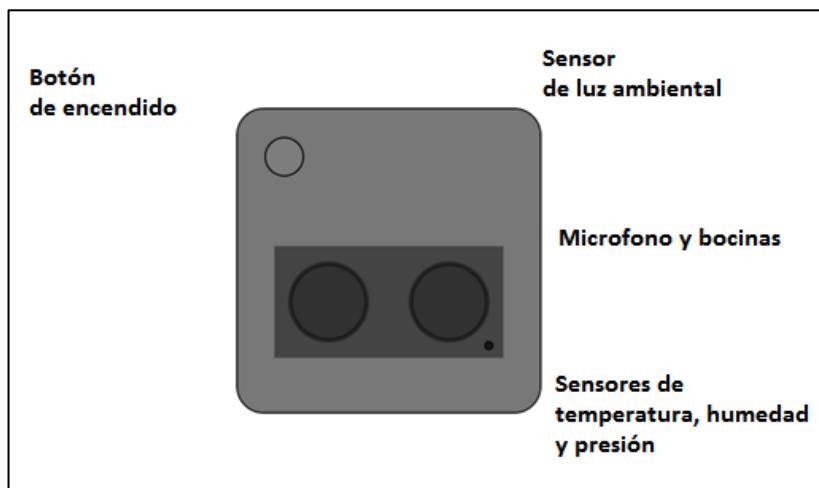


Ilustración 9. Características de Ubi.

Este proyecto se encuentra en fase de desarrollo y recibe aportaciones del público y futuros clientes por medio de la página web “Kickstarter” que es un sitio para financiar proyectos creativos (Chen & Strickler, 2008).

1.3 Servicios

Los servicios que ofrece la domótica se pueden agrupar según cinco aspectos: Ahorro energético, confort, seguridad, comunicación y plusvalía.

A continuación se describe cada uno de ellos.

1.3.1 Ahorro energético

El ahorro energético no es algo tangible, sino un concepto al que se puede llegar de muchas maneras. En muchos casos no es necesario sustituir los aparatos o sistemas del hogar por otros que consuman menos, sino una gestión eficiente de los mismos. En un principio tal vez haya que invertir para instalar el sistema, pero a la larga el usuario comprueba que sus recibos de electricidad, agua, etc. bajan, y con ello recupera el capital invertido (Redolfi, 2013).

Al igual que los cristales dobles¹ ahorran energía durante la temporada de frío en Europa al mantener el calor proporcionado por las chimeneas y hornos (Lindes, 2013), un sistema que supervisa y controla las luces y electrodomésticos apagándolos cuando no son necesarios también ahorra energía.

1.3.2 Confort

Proporciona al usuario una mejora considerable en la calidad de vida, ya que reduce el trabajo doméstico, aumentando el bienestar y aportando una serie de comodidades. Como aspectos importantes se pueden mencionar (Roqué, 2005):

- Climatización: Control de la calefacción o el aire acondicionado desde cualquier PC, teléfono inteligente, Tablet.

¹ Es un arreglo en el que al menos dos láminas de vidrio se unen por un separador perimetral dejando una cámara de aire o gas en el interior que le brinda propiedades aislantes térmicas y acústicas.

- Gestión de la iluminación: Automatización del apagado/encendido de las luces de la vivienda.
- Encendido y apagado remoto de electrodomésticos.

1.3.3 Seguridad

Se gestionan las alarmas que se presentan por problemas causados por intrusos o por alguna falla en la instalación (por ejemplo, fugas de gas). Se divide en detección de intrusos y en seguridad de la instalación (Redolfi, 2013).

- Detección perimetral de intrusos: Se activan las alarmas cuando se detecta una persona en los alrededores de la vivienda. Se puede utilizar barreras infrarrojas en jardines y rejas, sensores de magnéticos en puertas y ventanas.
- Detección interna de intrusos: Se utilizan sensores de movimiento para detectar personas dentro de la casa.
- Seguridad de la instalación: Utiliza sensores de humo, gas, agua, etc. Al detectar una anomalía, estos sensores pueden activar una alarma y cortar los suministros afectados mediante el uso de electroválvulas.

1.3.4 Comunicaciones

De manera general, un sistema domótico dispone de una red de comunicación y diálogo que permite la interconexión de una serie de equipos a fin de obtener información sobre el entorno doméstico y, realizar determinadas acciones sobre dicho entorno (Roqué, 2005).

Los elementos de campo (detectores, sensores, etc.), transmiten las señales a una unidad central que trata y elabora la información que recibe. En función de dicha información y de una determinada programación, la unidad central actúa sobre los circuitos relacionados con las señales que se recogen por los elementos de campo correspondientes.

1.3.5 Plusvalía.

Una casa con un sistema domótico se cotiza más alto en el mercado inmobiliario. La casa es más fácil de vender. Incorpora características únicas que no tiene la competencia. Es un valor añadido que le da mayor categoría (Active Homes, 2013).

1.4 Medios de Transmisión

El transporte de la información necesita de medios de transmisión, que pueden ser físicos (cableados) o a través del aire (inalámbricos). En el caso que se aborda, el diseño de una instalación domótica para una vivienda ya construida, es claramente comprensible que se opte por los medios inalámbricos; aunque hay que tener en cuenta el hecho de que no se dispone de todo tipo de dispositivos inalámbricos como para cubrir las necesidades del hogar, por lo que también se estudia el uso de medios físicos.

1.4.1 Cableados

Los medios de transmisión cableados están constituidos por un cable que se encarga de la conducción (o guiado) de las señales desde un extremo al otro.

Las principales características de los medios guiados son el tipo de conductor utilizado, la velocidad máxima de transmisión, las distancias máximas que puede ofrecer entre repetidores, la inmunidad frente a interferencias electromagnéticas, la facilidad de instalación y la capacidad de soportar diferentes tecnologías de nivel de enlace.

Entre los medios de transmisión más usados, se pueden mencionar: Al cable coaxial, el par trenzado, la fibra óptica y las señales de radio utilizando la red de energía eléctrica.

A continuación se describen brevemente cada una de ellas.

1.4.1.1. Coaxial.

Consta de un núcleo de hilo de cobre rodeado por un aislante, un apantallamiento de metal trenzado y una cubierta externa. Transporta señales eléctricas de alta frecuencia (Guanoluisa, 2012).

1.4.1.2. Par Trenzado.

Forma de conexión en la que dos cables aislados son entrelazados para tener menores interferencias y aumentar la potencia y la diafonía de los cables adyacentes (Guanoluisa, 2012).

1.4.1.3. Fibra Óptica.

La información se transmite en forma de señales lumínicas en lugar de señales eléctricas. Se trata de un hilo de cristal o polímero especial (núcleo) rodeado por otra capa del mismo material (revestimiento). La transmisión de la luz se basa en el fenómeno físico óptico de la refracción del rayo de luz (Redolfi, 2013).

1.4.1.4 Power Line Communications.

Se refiere a diferentes tecnologías que utilizan las líneas de energía eléctrica convencionales para transmitir señales de radio para propósitos de comunicación (Beaskoetxea, 2011).

1.4.2 Inalámbricos

En este tipo de medios tanto la transmisión como la recepción de información se lleva a cabo mediante antenas. A la hora de transmitir, la antena irradia energía electromagnética en el medio. Por el contrario, en la recepción la antena capta las ondas electromagnéticas del medio que la rodea.

La configuración para las transmisiones no guiadas puede ser direccional y omnidireccional. En la direccional, la antena transmisora emite la energía electromagnética concentrándola en un haz, por lo que las antenas emisora y receptora deben estar alineadas. En la omnidireccional, la radiación se hace de manera dispersa, emitiendo en todas direcciones, pudiendo la señal ser recibida

por varias antenas. Generalmente, cuanto mayor es la frecuencia de la señal transmitida es más factible confinar la energía en un haz direccional (Guanoluisa, 2012).

Respecto a los medios inalámbricos, los protocolos de comunicación más conocidos son el WiFi, GPRS, Bluetooth, Radio Frecuencia, Infrarrojo y Zigbee. A continuación se describen brevemente cada uno de ellos.

1.4.2.1 *Wi-Fi*.

La organización a la que pertenece la marca registrada, define WiFi como “cualquier red inalámbrica de acceso local que esté basada en los estándares 802.11 del Instituto de Ingenieros Eléctricos y Electrónicos”.

WiFi trabaja sin ninguna conexión de cables entre el emisor y receptor, utilizando tecnología de radio frecuencia. Cuando una corriente de RF se suministra a una antena, un campo electromagnético es creado, para después propagarse por el espacio.

1.4.2.2. *GPRS (General Packet Radio Service)*.

Es una extensión del Global System for Mobile Communications, o GSM, para la transmisión de datos no conmutada (o por paquetes).

1.4.2.3. *Bluetooth*.

Especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,5 GHz.

1.4.2.4. *Radiofrecuencia*.

Banda menos energética del espectro electromagnético, situada entre unos 3 Hz y unos 300 GHz cuyas ondas se pueden transmitir aplicando la corriente alterna originada en un generador a una antena.

1.4.2.5. Infrarrojos.

Radiación electromagnética cuyo rango de longitudes de onda va desde unos 700 nanómetros hasta 1 micrómetro.

1.4.2.6. ZigBee.

Es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radios digitales de bajo consumo. Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de la vida útil de sus baterías.

1.5 Protocolo X-10

X-10 es uno de los protocolos más antiguos que se están usando en aplicaciones domóticas. Diseñado en Escocia entre los años 1976 y 1978 con el objetivo de transmitir datos por las líneas de baja tensión a muy baja velocidad (60 bps en EEUU y 50 bps en Europa) y costos muy bajos. Al usar las líneas de eléctricas de la vivienda, no es necesario tender nuevos cables para conectar dispositivos.

El formato de codificación X-10 es un estándar usando transmisión de corrientes portadoras (Power Line Carrier = P.L.C) que, desde que se introdujo, ha desarrollado y manufacturado versiones O.E.M (Original Equipment Manufacturer) de su Sistema de Control del Hogar para muchas compañías incluyendo Leviton Manufacturing Co., Stanley Health / Zenith Co., Honeywell, Norweb y Busch Jaeger. Actualmente X10 es un protocolo que está presente en el mercado mundial, sobre todo en Norteamérica y Europa (España y Gran Bretaña fundamentalmente).

1.5.1 Características

El protocolo X-10, en sí, no es propietario, es decir, cualquier fabricante puede producir dispositivos X-10 y ofrecerlos en su catálogo, eso sí, está obligado a usar los circuitos del fabricante escocés que diseñó esta tecnología. Esto no resulta una gran desventaja ya que los circuitos integrados que implementan el X-10 tienen regalías muy bajas.

Es un sistema domótico de instalación sencilla y fácil manejo. No hay necesidad de conocimientos especiales para la instalación de los productos X10. Todos los productos son compatibles entre sí, por lo que se pueden combinar para formar el sistema que más se adecue a las necesidades.

Se caracteriza principalmente por:

- Ser un sistema descentralizado y configurable.
- De instalación sencilla (conectar y funcionar).
- De fácil manejo por el usuario.
- Compatibilidad casi absoluta con los productos de la misma gama, obviando fabricante y antigüedad.
- Flexible y ampliable.

Su considerable gama de productos permite aplicaciones diversas en los campos de:

- Seguridad: Intrusión, fugas de gas, inundaciones, incendio, alarma médica, simulación de presencia.
- Confort: Control centralizado / descentralizado de iluminación y aparatos así como persianas. Manejo con mando a distancia. Facilidades para audio y video. Posibilidad de gestión a través de computadora personal.
- Ahorro energético: Programación nocturna y optimización de recursos.
- Comunicación: Control telefónico remoto. Aviso de la vivienda ante incidentes (control telefónico bidireccional).
- Visión de Futuro: Una de las cosas que más preocupa cuando se invierte en tecnología hoy en día es su vida útil. El sistema domótico X10 sigue vigente después de más de 25 años y más de cien millones de aparatos funcionando por todo el mundo. Actualmente ya

se ha adaptado el sistema a 220V y se usa por toda Europa, además de EEUU.

- Inteligencia: El sistema X10 es inteligente y los aparatos interactúan entre sí. Por ejemplo, al entrar en casa por la tarde, el sensor de presencia puede encender secuencialmente el recibidor, el pasillo, el dormitorio, apagar el recibidor, conectar el calentador y encender dos lámparas del salón al 50%, todo de forma automática y con un solo botón para restablecer un nuevo modo de funcionamiento.

X10 se comunica entre los transmisores y receptores mediante el envío y recepción de señales a través de los cables de línea de alimentación. Estas señales son pulsos de RF (120 kHz) que representan la información digital. También se define un protocolo de transporte basado en radio inalámbrica.

Existen tres tipos de dispositivos X-10: los que sólo pueden transmitir órdenes, los que sólo pueden recibirlas y los que pueden enviar/recibir estas.

Los transmisores pueden direccionar hasta 256 receptores.

Los receptores vienen dotados de dos pequeños conmutadores giratorios, uno con 16 letras y el otro con 16 números) que permiten asignar una dirección de las 256 posibles. En una misma instalación puede haber varios receptores configurados con la misma dirección, todos realizarán la función pre-asignada cuando un transmisor envíe una trama con esa dirección. Evidentemente cualquier dispositivo receptor puede recibir órdenes de diferentes transmisores.

Los dispositivos bidireccionales, tienen la capacidad de responder y confirmar la correcta realización de una orden, lo cual puede ser muy útil cuando el sistema X-10 está conectado a un programa de computadora que muestre los estados en que se encuentra la instalación domótica de la vivienda.

Los logos que identifican a qué tipo pertenece un dispositivo se muestran en la Ilustración 10.



Ilustración 10. Logos de identificación X10

La red de la instalación es la base de todo el sistema de corrientes portadoras (X-10). El elemento básico y fundamental de la técnica de corrientes portadoras es el aprovechamiento doble de la instalación eléctrica ya existente, como conductor de energía y de información. Con los componentes X-10 la red, además de suministro de corriente, se encarga también de la transmisión de señales de mando para los diversos aparatos eléctricos. Con ello se puede enviar señales de corrientes portadoras a cualquier punto de la instalación que se desee, y a su vez pueden solicitarse de dicho punto las informaciones pertinentes. En la Ilustración 11, se muestra un ejemplo típico de un sistema X10.

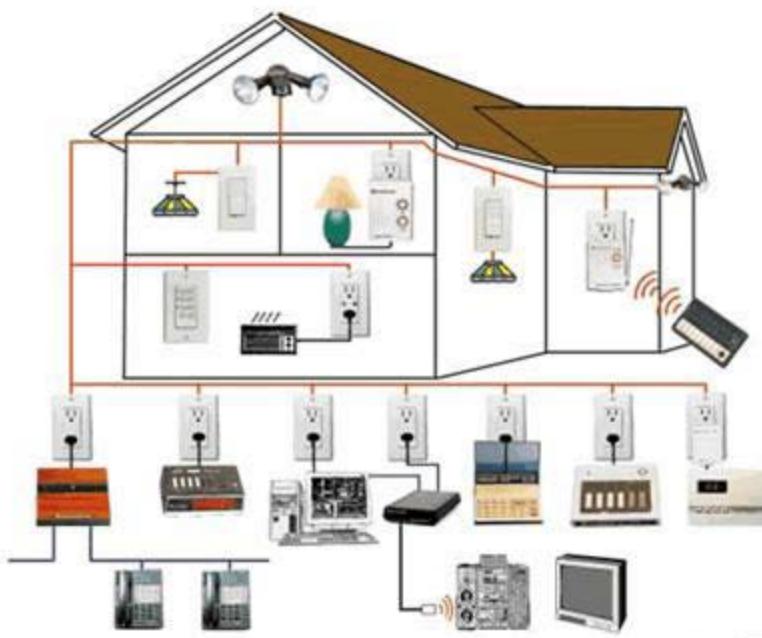


Ilustración 11. Ejemplo de un Sistema X10

El sistema permite el accionamiento a distancia y control remoto de diversos receptores eléctricos, desde uno o desde varios puntos.

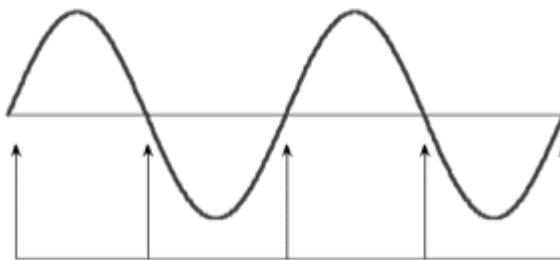
El sistema de corrientes portadoras trabaja tanto en redes de corriente alterna monofásica como trifásica.

1.5.2 Nivel Físico

El protocolo X-10 usa una modulación muy sencilla, comparado con las que usan otros protocolos de control por ondas portadoras. El transmisor X-10 está pendiente de los pasos por cero de la onda senoidal de 60 Hz típica de la alimentación eléctrica para insertar en un instante después una ráfaga muy corta de señal en una frecuencia fija. Las señales de control de X10 se basan en la transmisión de ráfagas de pulsos de RF (120 KHz) que representan información digital.

Se puede insertar esta señal en el semiciclo positivo y el negativo de la onda senoidal, un pulso sincronizado con el cruce por cero de la señal de la red (ver Ilustración 12). La codificación de un bit 1 o de un bit 0, depende de cómo se inyecte esta señal en los dos semiciclos. Un 1 binario se representa por un pulso de 120 KHz durante 1 milisegundo y el 0 binario se representa por la ausencia de ese pulso de 120 KHz (ver Ilustración 13). En un sistema trifásico el pulso de 1 milisegundo se transmite tres veces para que coincida con el paso por el cero en las tres fases.

Transmisores y receptores son sincronizados por el paso por cero de la tensión de red, de esta manera los transmisores saben cuando enviar los datos y los receptores cuando buscarlos.

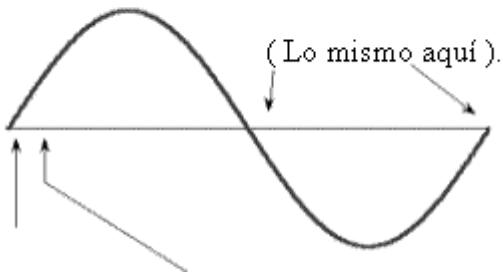


Los dispositivos X-10 no distinguen entre el paso por cero cuando la señal va de positivo a negativo que cuando va de negativo a positivo, ambos pasos por cero son interpretados de igual modo por el dispositivo.

Ilustración 12. Colocación de las señales de información en la red C.A.

Cada orden involucra 11 ciclos de red (220 ms para 50 Hz y 183.33, para 60Hz). A su vez, cada orden se transmite 2 veces, con lo cual toda la información transmitida tiene cuádruple redundancia. La relativamente alta frecuencia de la portadora de frecuencia que lleva la señal no puede pasar a través de un transformador o través de las fases de un sistema de múltiples fases.

Los receptores al igual que los transmisores detectan cada paso por cero y buscan la señal de 120 KHz durante un periodo de 1ms.



Cada paso por cero
ellos comprueban durante 1ms si está
..... presente la señal de 120 KHz

Ilustración 13. Puntos donde se comprueba si existe la señal de 120KHz

Para permitir el funcionamiento de los teclados inalámbricos, conmutadores remotos, y similares, un protocolo de radio también está definido. Operando a una frecuencia de 310 MHz en los EE.UU. y 433 MHz en los sistemas europeos, los dispositivos inalámbricos envían paquetes de datos que son muy similares a los paquetes normales de alimentación de la línea de control X10.

1.5.3 Comunicación

La transmisión completa de una orden X-10 necesita once ciclos de corriente. Primero se transmite una orden con el Código de Casa y el Número de Módulo que direccionan el módulo en cuestión. Luego se transmite otro orden con el código de función a realizar (Function Code). Hay 256 direcciones soportadas por el protocolo.

Esta trama se divide en tres campos de información:

1. Dos ciclos representan el Código de Inicio.
2. Cuatro ciclos representan el Código de Casa (letras A-P).
3. Cinco ciclos representan o bien el Código Numérico (1-16) o bien el Código de Función (Encender, Apagar, Aumento de Intensidad, etc.).

Para aumentar la fiabilidad del sistema, esta trama (Código de Inicio, Código de Casa y Código de Función o Numérico) se transmite siempre dos veces, separándolas por tres ciclos completos de corriente. Hay una excepción: en funciones de regulación de intensidad, se transmiten de forma continuada (por lo menos dos veces) sin separación entre tramas.

Los dispositivos están generalmente conectados en módulos X10 (receptores). X10 distingue entre módulos de lámparas y módulos de dispositivos. Los módulos de dispositivos proporcionan energía a los dispositivos eléctricos y aceptan órdenes X-10. Los módulos de dispositivos son capaces de gestionar cargas grandes (ej. máquinas de café, calentadores, motores...) simplemente encendiéndolos y apagándolos.

Un mensaje completo en X-10 está compuesto por el código de comienzo (1110), seguido por la letra de la casa y por un código de control. El código de control puede ser o una dirección de unidad o un código de comandos, dependiendo de si el mensaje es una dirección o un comando.

Un bloque completo de datos o paquete de información se compone de código de comienzo, código de la letra, código de control y sufijo.

El protocolo puede transmitir un mensaje que diga "Selección de código A3", seguido de "Encender". Varias unidades pueden ser tratadas antes de dar la orden, lo que permite a un comando actuar en varias unidades de forma simultánea. Por ejemplo, "Selección de código A3", "Selección de código A15", "Selección de código A3", y, finalmente, "Encender".

La Petición de Saludo se transmite para ver si existen otros transmisores X-10 dentro del rango de escucha. Esto permite al OEM asignar un Código de Casa diferente si se recibe un mensaje de Aceptación de Saludo.

El código de comienzo (1110) es el único que no se envía de forma complementaria. Inmediatamente después del código de comienzo se transmite la dirección de casa o letra. Después de enviar el código de la letra se envía la dirección de unidad o número.

Debido al medio de transmisión utilizado, los diseñadores del código X-10 decidieron transmitir dos veces cada bloque de información para que el sistema ganara en fiabilidad (ver Ilustración 14). Cada once ciclos se transmite un bloque de datos:

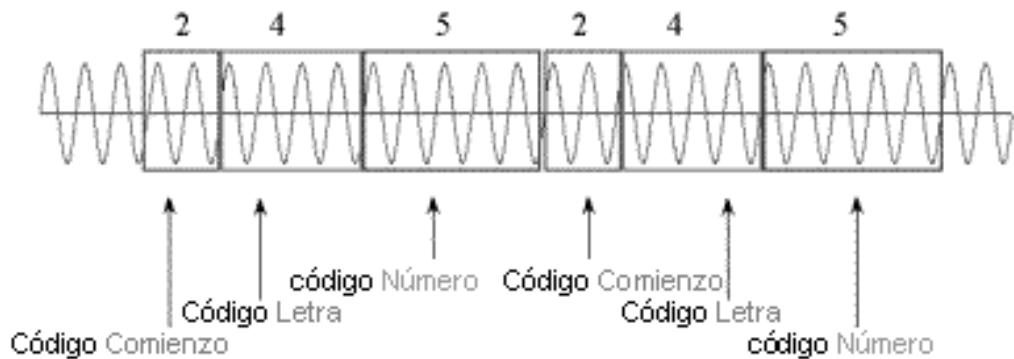


Ilustración 14. Forma de enviar la información con el protocolo X10 a la línea eléctrica

Cada par de bloques de información deben estar precedidos por 6 pasos por cero.

Estos 3 ciclos de margen son necesarios para que el receptor mueva los datos de sus registros en cada uno de los seis pasos por cero. Una transmisión estándar X-10 normal necesita 47 ciclos de la señal de red, como se observa en la Ilustración 15.

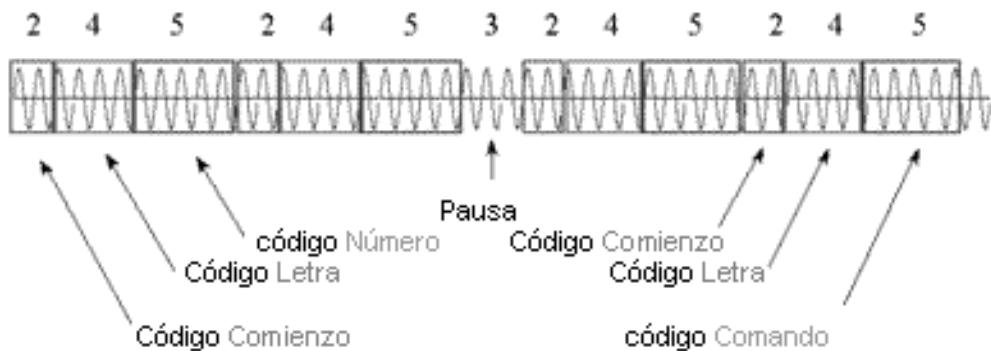


Ilustración 15. Transmisión estándar X10

A una frecuencia de 50 Hz ello supone un tiempo igual a 0,94 segundos en transmitir una orden completa. Hay excepciones a esta regla. Por ejemplo, el código de Aumentar Intensidad (Bright) y Atenuar intensidad (Dim) no requiere los tres ciclos de espera entre comandos consecutivos Dim o comandos consecutivos Bright. Sin embargo sí son necesarios los tres ciclos de espera entre códigos diferentes.

1.5.4 Módulos de Potencia

Dependiendo de la carga a controlar, los módulos a utilizar son diferentes. Para cargas de lámpara incandescente se puede utilizar un módulo de la lámpara o el módulo de interruptor de pared. Estos módulos conmutan la energía mediante un interruptor TRIAC de estado sólido y también son capaces de atenuar la carga de la lámpara. Los módulos de lámpara son bastante silenciosos al operar, y, en general soportan cargas que van desde los 40 watts a 500 watts.

Para cargas distintas a las lámparas incandescentes, como las lámparas fluorescentes, lámparas de descarga de alta intensidad, y los aparatos electrodomésticos eléctricos, el módulo de la lámpara es inadecuado y en su lugar se debe utilizar un módulo de aplicación. Estos módulos conmutan la energía mediante un relé de impulso.

Muchos de los módulos ofrecen una función llamada control local. Si el módulo está apagado, la función de encendido del interruptor de la lámpara o

aparato hará que el módulo se encienda. De esta manera, una lámpara todavía puede ser encendida sin necesidad de utilizar un controlador X10. Los módulos de conmutación de pared no ofrecen esta característica.

Algunos módulos de interruptor de pared ofrecen una característica llamada atenuación local. Normalmente, el pulsador de un interruptor de pared simplemente ofrece on / off, sin posibilidad de oscurecimiento local de la lámpara controlada. Si la atenuación local es posible, manteniendo pulsado el botón hará que la lámpara varíe su rango de brillo de forma cíclica.

Hay módulos que detectan e informan de temperatura, luz, infrarrojo o cierres y aperturas de contacto. Los módulos de dispositivos incluyen termostatos, alarmas audibles y mandos para interruptores de baja tensión.

La Ilustración 16 resume los módulos de aparato de X10.

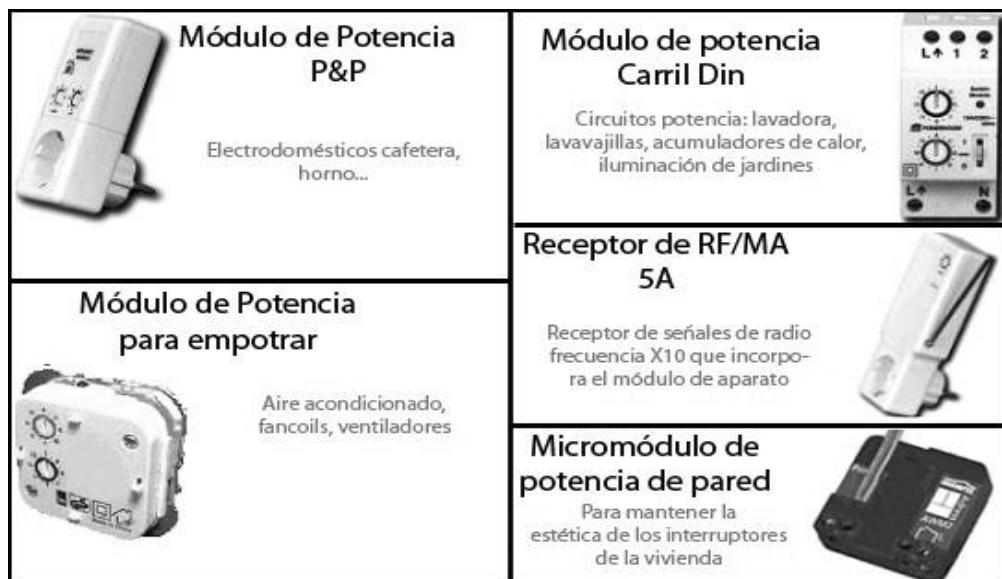


Ilustración 16. Módulos de X10

Todos pueden recibir órdenes X10 locales o a través de teléfono procedentes de la instalación eléctrica de la vivienda. Estas órdenes pueden proceder de programadores, mandos a distancia, detectores de presencia, PC, Internet, etc.

1.5.6 Limitaciones

- Un RCD (Dispositivo de Corriente Residual) o disyuntor puede atenuar las señales X10 que pasen por el dispositivo. Esto significa que las señales X10 que pasen por un disyuntor, pueden no ser lo suficientemente fuertes como para proporcionar una comunicación fiable.
- Los televisores o los dispositivos móviles pueden causar falsos o desactivar las señales. Los filtros de ruido pueden ayudar a evitar el ruido externo de la señal de X10.
- Además, cierto tipo de fuentes de alimentación utilizadas en equipos electrónicos (computadoras, televisiones y receptores de señal por satélite) “se comen” las señales X10 proporcionando una baja impedancia para señales de alta frecuencia.
- Algunos módulos X10 no funcionan del todo con los dispositivos de baja potencia (50 Watts) o dispositivos como los focos fluorescentes que no presentan cargas resistivas.
- Las señales X10 sólo pueden transmitir un comando a la vez, primero indicando la dirección del dispositivo a controlar, y luego la operación que ese dispositivo debe llevar a cabo. Si dos señales X10 se transmiten al mismo tiempo pueden chocar o intercalarse, lo que lleva a los comandos a que, o bien no pueden ser descifrados, o que indiquen operaciones incorrectas.
- El protocolo X10 es también lento. Se tarda aproximadamente tres cuartas partes de un segundo para transmitir una dirección de dispositivo y un comando.

1.6 Microcontrolador Arduino

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares.

1.6.1 Entorno de Desarrollo Arduino

El entorno de desarrollo Arduino está constituido por un editor de texto para escribir código de programación, un área de mensajes, una consola de texto, una barra de herramientas con botones para las funciones comunes, y una serie de menús. Permite la conexión con el hardware Arduino para cargar los programas y comunicarse con ellos (ver Ilustración 17).

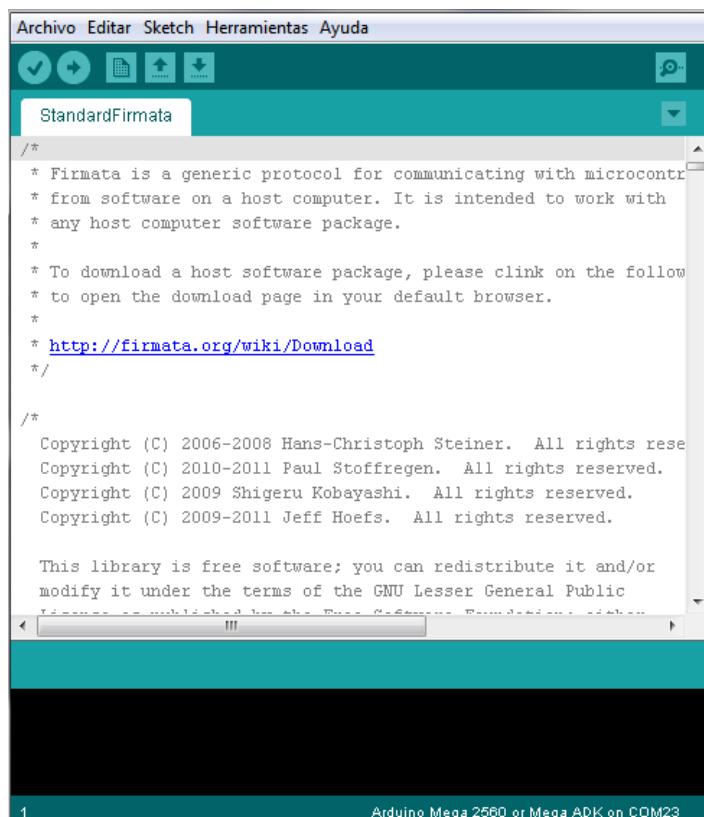


Ilustración 17. Entorno de programación Arduino

El software en Arduino es utilizado para escribir lo que se denomina “Sketch”. Estos sketches son escritos en el editor y se guardan con la extensión “.ino”. Existe la posibilidad de cortar, pegar, buscar y reemplazar texto. En el área de mensajes se muestra información mientras se cargan los programas y también muestra errores. La consola muestra el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones. La barra de herramientas permite verificar el proceso de carga, creación, apertura y guardado de programas, y la monitorización serie.

La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el lenguaje de programación de alto nivel “Processing”. Es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino, debido a que éste usa la transmisión serial de datos soportada por la mayoría de lenguajes, tales como: C, C++, C#, Java, Matlab, PHP, Perl, Processing, Python, Scratch.

1.6.2 Gama de Arduinos

Las especificaciones de las placas Arduino más populares se resumen en la siguiente tabla:

| Modelo | Micro controlador | Voltaje de entrada | Voltaje del sistema | Frecuencia de Reloj | Digital I/O | Entradas Analógicas | PWM | UART | Memoria Flash |
|------------------|-------------------|--------------------|---------------------|---------------------|-------------|---------------------|-----|------|---------------|
| Arduino Due | AT91SAM3X8E | 5-12V | 3,3V | 84MHz | 54 | 12 | 12 | 4 | 512Kb |
| Arduino Leonardo | ATmega32U4 | 7-12V | 5V | 16MHz | 20 | 12 | 7 | 1 | 32Kb |
| Arduino Uno - R3 | ATmega328 | 7-12V | 5V | 16MHz | 14 | 6 | 6 | 1 | 32Kb |

Tabla 1. Modelos de Arduino y sus características.

1.6.2.4 Arduino Mega 2560

El Arduino Mega 2560 es una placa basada en el microcontrolador ATmega2560 y es el elegido para ser utilizado en este proyecto. Tiene 54 pines de entrada/salida digital (15 de los cuales pueden ser utilizados como salidas PWM), 16 entradas analógicas, 4 UART's (Puerto serial), un cristal oscilador de 16 MHz, conexión USB, entrada para alimentación, conector ICSP y un botón de reinicio. Difiere de otras placas al no utilizar el chip FTDI USB-to-serial. En su lugar utiliza el ATmega16U2 programado como convertidor USB a puerto serial.

En la Tabla 2 se resumen las características relevantes de este micro controlador.

| | |
|---|--|
| Micro controlador | ATmega2560 |
| Voltaje de operación | 5V |
| Voltaje de entrada (recomendado) | 7-12V |
| Voltaje de entrada (límites) | 6-20V |
| Pines Digitales I/O | 54 (de los cuales 15 pueden ser salidas PWM) |
| Entradas análogas | 16 |
| Corriente DC pines I/O | 40 mA |
| Corriente DC para voltaje 3.3V | 50 mA |
| Memoria Flash | 256 KB (8 KB utilizados por el gestor de arranque) |
| SRAM | 8 KB |
| EEPROM | 4 KB |
| Velocidad del reloj | 16 MHz |

Tabla 2. Características generales del Arduino Mega 2560.

En la Ilustración 18, se muestra el Arduino Mega 2560 que debido a sus características sobresalientes fue el que se utilizó en el proyecto.



Ilustración 18. Arduino Mega 2560

1.7 Raspberry Pi modelo B

La Raspberry Pi es una computadora del tamaño de una tarjeta de crédito que se puede conectar a una TV y a un teclado. Desarrollada por una fundación

que lleva el mismo nombre, es capaz de operar la mayoría de las cosas que una computadora normal, como hojas de cálculo, procesar palabras y juegos. También reproduce video de alta definición (ver Ilustración 19).

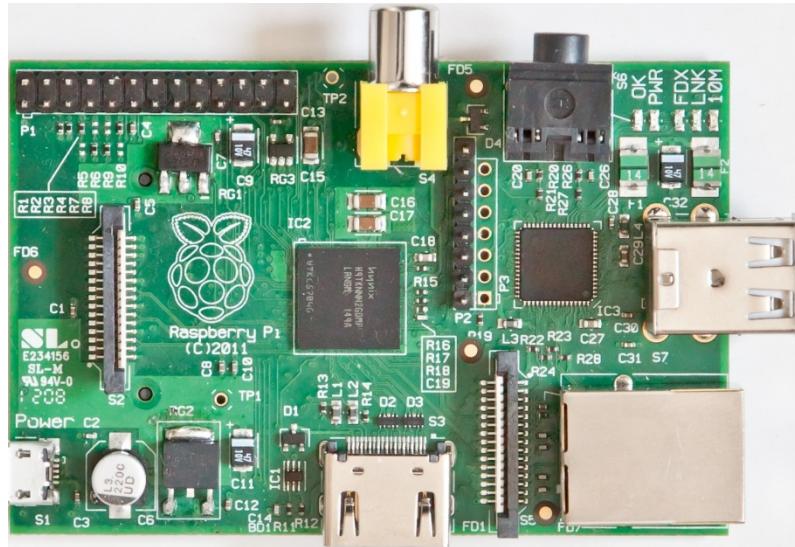


Ilustración 19. Raspberry Pi Modelo B

1.7.1 Especificaciones de la computadora Raspberry Pi modelo B

El Raspberry Pi modelo B, está construido en base al microprocesador Broadcom BCM2835. El BCM2835 es un procesador de aplicaciones multimedia orientado a dispositivos móviles y embebidos. Además de esto, varios otros componentes se han incluido para soportar 2 puertos USB, salida de video RCA, tarjeta de almacenamiento SD, Ethernet y tiene una memoria SDRAM de 512MB.

El modelo A, no contaba con puerto Ethernet, tenía 256 MB de memoria SDRAM y un solo puerto USB.

1.7.2 Sistema Operativo

Se utilizan mayoritariamente sistemas operativos basados en el núcleo Linux. Raspbian, una distribución derivada de Debian que está optimizada para el hardware de Raspberry Pi, es la distribución recomendada por la fundación para iniciarse.

Existe la aplicación NOOBS (New Out of Box Software), utilidad que facilita la instalación de diferentes sistemas operativos para Raspberry Pi. Se distribuye gratuitamente desde la página web de la fundación en forma de archivo zip que se copia descomprimido a una tarjeta SD de 4 GB o superior, y una vez arrancada la placa con la tarjeta por primera vez, aparece un menú en que se da la opción de instalar una de las diferentes distribuciones en el espacio libre de la tarjeta de memoria, o acceder a internet con el navegador integrado. Más adelante si se desea, es posible acceder a este menú apretando la tecla shift durante el arranque para reinstalar el sistema operativo, elegir otro, o editar el archivo config.txt. NOOBS contiene las distribuciones Linux de carácter general Raspbian, Arch Linux ARM y Pidora; las distribuciones Linux para media center con XBMC OpenELEC y RaspBMC.

1.7.3 Dimensiones

La Raspberry Pi es un dispositivo pequeño que mide 85.60mm x 53.98mm x 17mm y pesa solamente 45g. Estas medidas son perfectas para la automatización del hogar, donde un dispositivo pequeño puesto en una caja puede ser montado dentro de una caja eléctrica, o para reemplazar algún objeto en una pared (ver Ilustración 20).

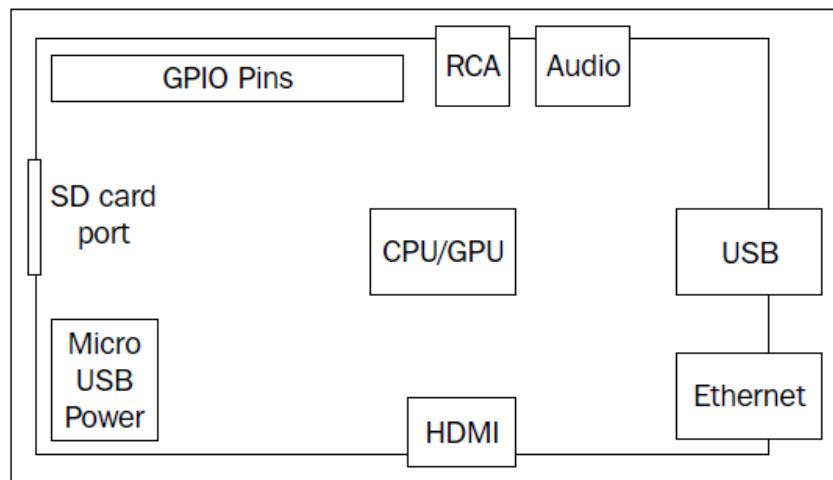


Ilustración 20. Distribución de Elementos Raspberry Pi (Raspberry Pi, 2012).

1.8 Python

Python presenta una serie de ventajas que lo hacen muy atractivo, tanto para su uso profesional como para el aprendizaje de la programación. Entre las más interesantes desde el punto de vista didáctico se tiene:

- Es un lenguaje muy expresivo, es decir, los programas Python son muy compactos: un programa en Python suele ser bastante más corto que su equivalente en lenguajes como C. (Python es considerado por muchos como un lenguaje de muy alto nivel.)
- Es muy legible. La sintaxis Python es muy elegante y permite la escritura de programas cuya lectura resulta más fácil que si se utilizaran otros lenguajes de programación.
- Ofrece un entorno interactivo que facilita la realización de pruebas y ayuda a despejar dudas acerca de ciertas características del lenguaje.
- El entorno de ejecución detecta muchos de los errores de programación que escapan al control de compiladores y proporciona información detallada para detectarlos y corregirlos.
- Se puede utilizar como lenguaje imperativo procedural o como lenguaje orientado a objetos.
- Posee estructuras de datos que se pueden manipular de modo sencillo.

Estas características hacen que sea relativamente fácil traducir métodos de cálculo a programas Python.

Fue diseñado por Guido Van Rossum y está en un proceso de continuo desarrollo por una gran comunidad de desarrolladores.

Una ventaja fundamental es que el intérprete se puede descargar directamente desde su página web sin costo alguno y tiene versiones para prácticamente cualquier plataforma en uso. (Windows, Linux, Mac, etc.)

1.9 Django

Django es un framework de desarrollo web que ahorra tiempo y hace que este proceso sea divertido. Utilizando Django se puede crear y mantener aplicaciones web de alta calidad con un mínimo esfuerzo.

En el mejor de los casos, el desarrollo web es un acto entretenido y creativo; en el peor, puede ser una molestia repetitiva y frustrante. Django permite enfocarse en la parte divertida al mismo tiempo que mitiga el esfuerzo de las partes repetitivas. De esta forma, provee un alto nivel de abstracción de patrones comunes en el desarrollo web, atajos y tareas frecuentes de programación y convenciones claras sobre cómo solucionar problemas. Al mismo tiempo Django intenta no entrometerse, dejando trabajar fuera del ámbito del framework según sea necesario.

Django nació de aplicaciones de la vida real escritas por un equipo de desarrolladores web en Lawrence, Kansas. Nació en el otoño de 2003, cuando los programadores del diario *Lawrence Journal World*, Adrian Holvaty y Simon Willison, comenzaron a usar Python para crear sus aplicaciones. El equipo de The World Online, responsable de la producción y mantenimiento de varios sitios locales de noticias, prosperaba en un entorno de desarrollo dictado por las fechas límite del periodismo. Para los sitios – incluidos LJWorld.com, Lawrence.com y KUsports.com – los periodistas (y directivos) exigían que se agregaran nuevas características y que aplicaciones enteras se crearan a una velocidad vertiginosa, a menudo con solo días u horas de preaviso. Es así que Adrián y Simón desarrollan por necesidad un framework de desarrollo web que les ahorrara tiempo – era la única forma en que podían crear aplicaciones operables en tan poco tiempo - .

En el verano de 2005, luego de haber desarrollado este framework hasta el punto en que estaba haciendo funcionar la mayoría de los sitios World Online, el equipo que ahora incluía a Jacop Kaplan-Moss, decidió liberar el framework como

software de código abierto. Lo liberaron en julio de 2005 y lo llamaron Django, por el guitarrista de jazz Django Reinhardt (Dinkins).

A pesar de que Django ahora es un proyecto de código abierto con colaboraciones por todo el mundo, los desarrolladores originales de Word Online todavía aportan una guia centralizada para el crecimiento del framework, también colaboran con otros aspectos importantes tales como tiempo de trabajo, materiales de marketing, y hosting/ancho de banda para el sitio web del framework.

1.10 Firmata

Es un protocolo genérico para comunicación con microcontroladores desde el software en una computadora “host”. El objetivo central es hacer del microcontrolador una extensión del entorno de programación, de una manera que se sienta natural.

Fue diseñado para ser flexible y abierto de modo que cualquier entorno de programación pueda soportarlo y que sea fácil de implementar en ambos, el microcontrolador y la computadora host.

CAPÍTULO 2

DISEÑO DEL SISTEMA DOMÓTICO

2.1 Modelo del Sistema

En este capítulo se observa el diseño de los elementos principales para la construcción del sistema.

El sistema pretende ayudar a una persona con artrosis de rodilla, facilitando la acción de encender y/o apagar luminarias. La persona afectada por esta condición degenerativa necesita de mucho reposo y con el uso del sistema, mantiene su posición de descanso, minimizando así, la necesidad de levantarse constantemente para accionar la luminaria.

Para lograr esto, se necesita un sistema que sea fácil de instalar, de mantenimiento sencillo, que el manejo del mismo no sea complicado para el usuario y que se pueda controlar desde dispositivos móviles.

En la Ilustración 21, se presenta el diagrama a bloques del sistema a diseñar.

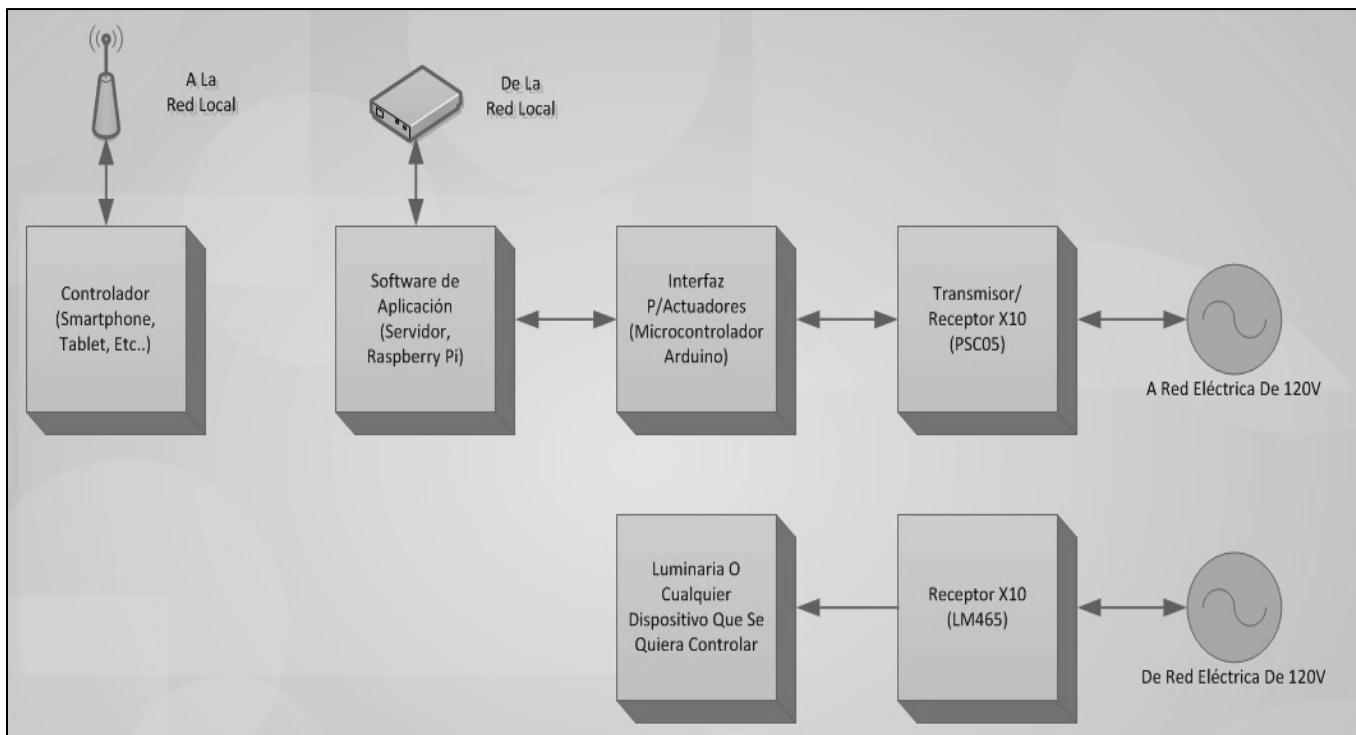


Ilustración 21. Diagrama a bloques del Sistema Domótico.

El sistema tiene una arquitectura centralizada, es decir, todas las instrucciones del usuario están dirigidas a un servidor que se encarga de procesar y enviar las órdenes correspondientes a los actuadores.

2.1.1 Servidor

Se utiliza una minicomputadora Raspberry Pi modelo B; por su tamaño y versatilidad (Sistema operativo, aplicaciones, puertos de comunicación, etc.), para cumplir con las funciones necesarias y funcionar como servidor del sistema (ver Ilustración 22).

- La placa del microcontrolador Arduino estará conectada a través del puerto USB.
- Para visualizar el sistema se puede utilizar un televisor o monitor con entrada HDMI o uno con entrada RCA (video compuesto).
- Se conecta a la red local por medio del puerto Ethernet.
- Un adaptador AC/DC comercial que da una salida de voltaje de 5V a 1A, junto con un cable USB a Micro USB se utilizan como fuente de alimentación.

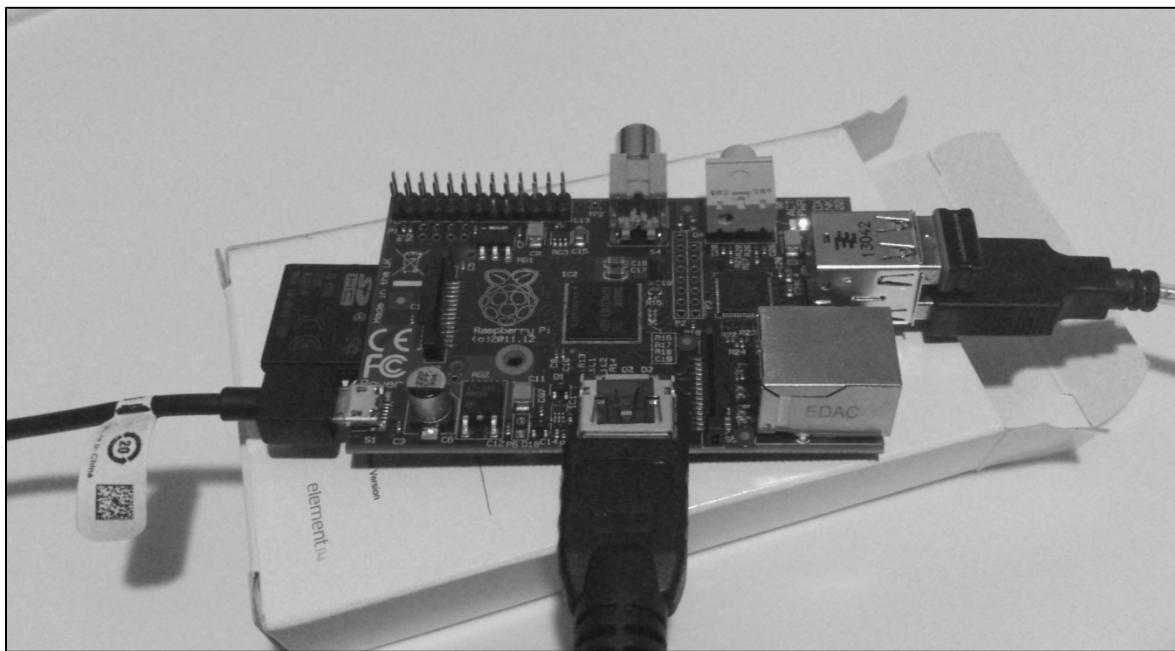


Ilustración 22. Imagen del Servidor (Raspberry Pi, modelo B).

2.1.2 Interfaz

Se utilizó una aplicación web como interfaz del sistema. Esta aplicación se aloja en el servidor, y permite al usuario elegir si enciende o se apaga la luminaria.

La interfaz fue escrita en Python, utilizando el framework “Django” que genera una vista final para el usuario utilizando plantillas HTML.

2.1.3 Actuadores

Cuando el usuario elija una opción, el sistema envía una señal a una tarjeta Arduino, la cual tendrá cargado un programa que interpretará la orden, la transformará en código X10 y la comunicará a los módulos transmisor y receptor comerciales X10, para que estos a su vez se encarguen de encender o apagar la luminaria, según sea el caso.

2.2 Software

El software del sistema está integrado por los programas realizados tanto en Django, como en el entorno de desarrollo Arduino.

Es decir, se diseñó un programa para la interfaz del usuario, y otro programa para que el microcontrolador Arduino se comunique con los módulos de transmisión y recepción, utilizando el protocolo X10.

2.2.1 Aplicación Web

Para crear esta aplicación se utilizó el tutorial que viene en el sitio web de Django.

Este tutorial ayuda a crear una aplicación de encuestas básicas, aunque no es precisamente el tipo de aplicación que se buscaba crear, se pensó que algunos elementos se podrían conservar al momento de crear la aplicación del sistema (los nombres de las clases y funciones por ejemplo); también se mantienen para aprovechar al máximo la información que suministra la fundación Django (Django Software Foundation, 2013).

Cuando se utiliza Django se generan varios archivos, los cuales modifican cada una de las propiedades de la aplicación web que se está desarrollando.

2.2.1.1 Archivo Models.py

El archivo “models.py” contiene una descripción de la base de datos, como una clase de Python. A esto se le llama modelo. Usando esta clase, se pueden crear, buscar, actualizar y borrar entradas de la base de datos usando código sencillo en lugar de escribir declaraciones SQL repetitivas.

En la Ilustración 23 se observa cómo se crea una clase llamada “Poll” (encuesta en inglés) que define el campo “question” (pregunta en inglés), esta instancia de la clase es un campo en la base de datos de la aplicación, por ejemplo, “CharField” se utilizó para indicar que los elementos son caracteres.

```
class Poll(models.Model):
    question = models.CharField(max_length=200)
```

Ilustración 23. Clase “Poll”

A la hora de utilizar la aplicación este campo tuvo como contenido: “Encendido y apagado de una lámpara”.

También se creó la clase “Choice” (opción en inglés) que definió los elementos que se tendrían para seleccionar una vez que la aplicación estuviera funcionando (ver Ilustración 24).

En este caso las opciones fueron: “Encender” y “Apagar”.

```
class Choice(models.Model):
    choice_text = models.CharField(max_length=200)
```

Ilustración 24. Clase "Choice"

2.2.1.2 Archivo Views.py

El archivo “views.py” (ver Ilustración 25) tuvo la lógica de la página y se le denominó “vista”.

```
class IndexView(generic.ListView):
    context_object_name = 'latest_poll_list'
    template_name = 'polls/index.html'

class DetailView(generic.DetailView):
    model = Poll
    template_name = 'polls/detail.html'

class ResultsView(generic.DetailView):
    model = Poll
    template_name = 'polls/results.html'

class Results2View(generic.DetailView):
    model = Poll
    template_name = 'polls/results2.html'
```

Ilustración 25. Views.py

- En la clase “IndexView” se manda llamar a los elementos de la clase Poll, para que sean mostrados en la plantilla principal del sistema (index.html en este caso, ver Ilustración 26).

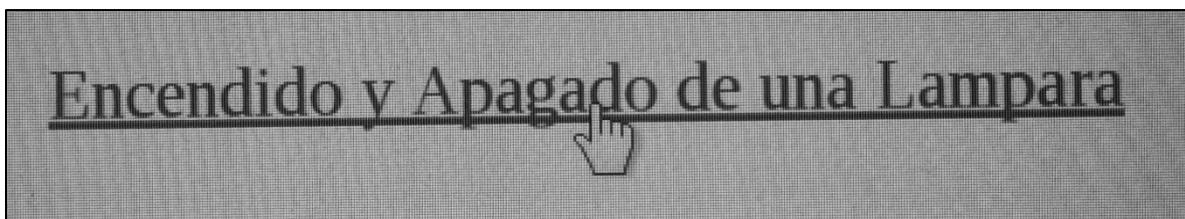


Ilustración 26. Entrando al sistema (index.html)

- “DetailView” es la vista en donde se muestran los elementos que puede elegir el usuario (Encender Lampara y Apagar Lampara, ver Ilustración 27).

Encendido y Apagado de una Lampara

Encender Lampara

Apagar Lampara

[Elegir](#)

Ilustración 27. DetailView (detail.html)

- Cuando la opción “Encender lámpara” sea seleccionada (ver Ilustración 28), se muestra una vista con la leyenda “Escogiste encender lámpara” (ResultsView); en caso contrario, se muestra “Escogiste apagar lámpara” (Results2View).

Escogiste encender la Lampara

Encender Lampara -- No. de veces seleccionado: 42
 Apagar Lampara -- No. de veces seleccionado: 38

Ilustración 28. ResultsView (results.html)

- La función “submit” se encarga de verificar que elemento fue elegido por el usuario al oprimir el botón “Elegir”. Se envía un mensaje de error cuando no se haya seleccionado ninguna opción.

Se envía una señal para cambiar el estado lógico de un led conectado en el pin número 13 del micro controlador Arduino (se enciende o se apaga), en este paso se lleva a cabo la función del programa cargado en el micro controlador comparando el estado lógico del led para comunicarlo a los módulos X10 (Ilustración 29).

Una vez hecho esto, se muestra en la pantalla la vista de resultados e indica si se desea repetir el proceso utilizando el enlace “¿Volver a Elegir?”.

```

def submit(request, poll_id):
    p = get_object_or_404(Poll, pk=poll_id)
    try:
        selected_choice = p.choice_set.get(pk=request.POST['choice'])
    except (KeyError, Choice.DoesNotExist):
        return render(request, 'polls/detail.html', {
            'poll': p,
            'error_message': "No seleccionaste una opción.",
        })
    else:
        selected_choice.votes += 1
        selected_choice.save()

        a = Arduino('/dev/ttyACM0')
        if selected_choice.id == 1:
            a.digital[13].write(1)
            return HttpResponseRedirect(reverse('polls:results', args=(p.id,)))
        else:
            a.digital[13].write(0)
            return HttpResponseRedirect(reverse('polls:results2', args=(p.id,)))

```

Ilustración 29. Función "submit"

2.2.1.3 Archivo *Urls.py*

El archivo “urls.py” especifica que vista es llamada a ejecución y cuál es la ruta que seguirá (ver Ilustración 30).

```

urlpatterns = patterns '',
    url(r'^$', views.IndexView.as_view(), name='index'),

    url(r'^^(?P<pk>\d+)/$', views.DetailView.as_view(), name='detail'),

    url(r'^^(?P<pk>\d+)/results/$', views.ResultsView.as_view(), name='results'),
    url(r'^^(?P<pk>\d+)/results2/$', views.Results2View.as_view(), name='results2'),

    url(r'^^(?P<poll_id>\d+)/vote/$', views.vote, name='vote'),
)

```

Ilustración 30. Urls.py

2.2.1.4 Templates

Los “templates” son plantillas escritas en HTML que describen el diseño de la página.

Tomadas en su conjunto, estas piezas se aproximan al patrón de diseño Modelo-Vista-Controlador (MVC) mostrado en la Ilustración 31. Dicho de manera

más sencilla, se especifica una forma de desarrollar software en la que el código para definir y acceder a los datos (el modelo) está separado de la solicitud lógica de la asignación de ruta (el controlador), que a su vez está separado de la interfaz del usuario (la vista).

Una ventaja clave de este enfoque es que los componentes tienen un acoplamiento flexible (Loosely coupled) entre sí. Esto significa que cada pieza de la aplicación web que funciona sobre Django tiene un único propósito y puede ser modificado de forma independiente sin afectar a las otras piezas. Por ejemplo, un desarrollador puede cambiar la URL de cierta parte de la aplicación sin afectar la implementación subyacente.

En esta aplicación la plantilla muestra las opciones que el usuario tiene para escoger (ver Ilustración 32).

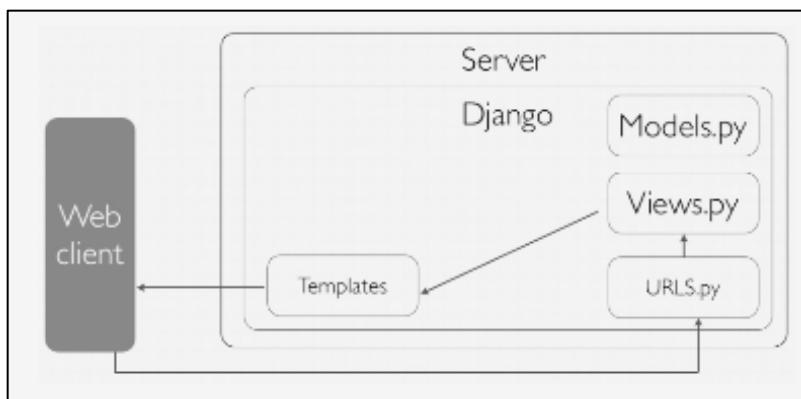


Ilustración 31. Modelo-Vista-Controlador (MVC)

Ilustración 32 muestra la interfaz de usuario de la aplicación. La cabecera indica "Encendido y Apagado de una Lampara". Abajo de la cabecera, hay dos botones radio:

- Encender Lampara
- Apagar Lampara

Un botón "Elegir" se encuentra debajo de los radios. Una línea horizontal separa la sección anterior de la siguiente, que contiene el texto:

Contactame en el Tuister!! ;)
Rafael Perez @_RufisTofeles

Ilustración 32. Diseño de la plantilla HTML del sistema

2.2.2 Arduino

Para que la aplicación web pueda encender un led en la placa Arduino Mega 2560, debe tener cargado un programa con el protocolo Firmata. Dicho programa viene incluido en las librerías del entorno de programación Arduino, bajo el nombre de “StandardFirmata”.

Como su licencia es de código libre, se puede modificar su contenido para que se ajuste a las necesidades del proyecto e incluya el uso de la librería X10.

El propósito de este programa será el de leer el estado lógico del led que por medio del protocolo Firmata, la aplicación web controla; una vez leído ese estado el programa evalúa, si se envía el código X10 para encender o apagar la luminaria, que está conectada a su vez los módulos comerciales de comunicación X10.

Se muestra en la Ilustración 33 el diseño del programa en lenguaje Arduino (similar al lenguaje C) para que el microcontrolador se comunique por medio del protocolo X10 con los módulos de transmisión y recepción que se usaron en la red eléctrica de la casa para enviar a estos las ordenes que el usuario requiera.

```
#include <x10.h>
#include <x10constants.h>

/* X10 Constantes */
#define zcPin 2 // zero crossing pin
#define dataWPin 3 // pin de transmisión
#define RPT_SEND 2 // número de veces que se reenvía un código X10

/* Variable X10 */
x10 myHouse = x10(zcPin, dataWPin); //Crea una nueva instancia de X10
void setup()
{
    /* Configuración X10 */

    pinMode(zcPin, INPUT); //Configura el pin especificado como una entrada
    pinMode(dataWPin, OUTPUT); //Configura el pin especificado como una salida
    digitalWrite(zcPin, HIGH); //Se polariza la entrada con 5v(HIGH)
}
```

```
void loop()
{
    int ledPin = 13;
    int var = 0;

    var = digitalRead(ledPin);

    if (ledPin == HIGH){ // Se evalua si el led controlado por la web app está encendido
        //Envia datos al modem PSC05 conectado al Arduino
        myHouse.write(HOUSE_A,UNIT_1,RPT_SEND);
        myHouse.write(HOUSE_A,ON,RPT_SEND);
    }

    if (ledPin == LOW){ // Se evalua si el led controlado por la web app está apagado
        //Envia datos al modem PSC05 conectado al arduino
        myHouse.write(HOUSE_A,UNIT_1,RPT_SEND);
        myHouse.write(HOUSE_A,OFF,RPT_SEND);
    }
}
```

Ilustración 33. Código escrito en lenguaje Arduino

Este código se agregó en cada una de las partes correspondientes en el programa principal StandardFirmata.

2.3 Hardware

El hardware del sistema estará formado por el servidor (minicomputadora Raspberry Pi), Arduino y los módulos de transmisión y recepción con protocolo X10.

A continuación se describen los elementos del hardware del sistema.

2.3.1 Módulo PSC05

El módulo PSC05 de dos vías permite interconectar el equipo con la línea eléctrica de 120 V mediante el protocolo X10. Se conecta por medio de un cable RJ11 (ver Ilustración 34).



Ilustración 34. Módulo PSC05

2.3.2 Módulo para lámpara LM465

Este módulo es un receptor que responde a señales X10 generadas por controladores conectados en la línea eléctrica. Simplemente se tiene que seleccionar el código en el módulo (ver Ilustración 36). En la Ilustración 35 se muestra una lámpara conectada al módulo.

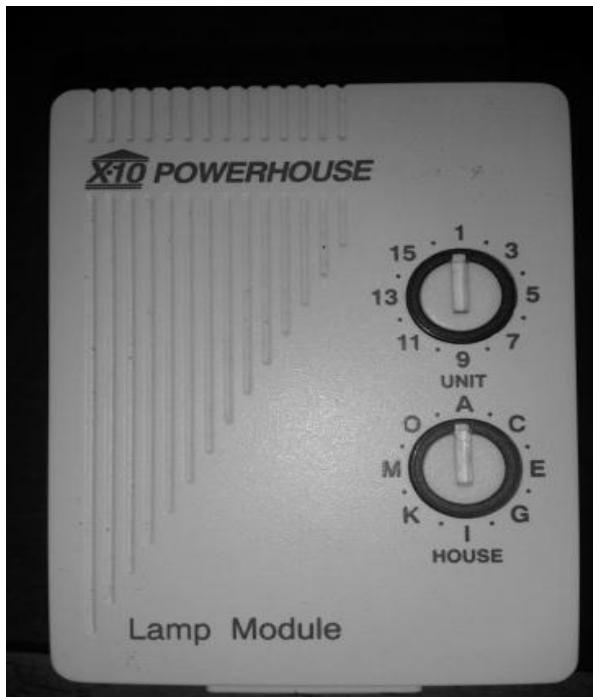


Ilustración 36. Módulo LM465



Ilustración 35. Lámpara conectada al LM465

2.3.3 Conexión Arduino-PSC05

Para realizar la conexión eléctrica entre el microcontrolador Arduino y el módulo PSC05, se toma un cable telefónico de 4 hilos y cortando uno de los extremos se conecta de acuerdo al siguiente esquema (ver Ilustración 37).

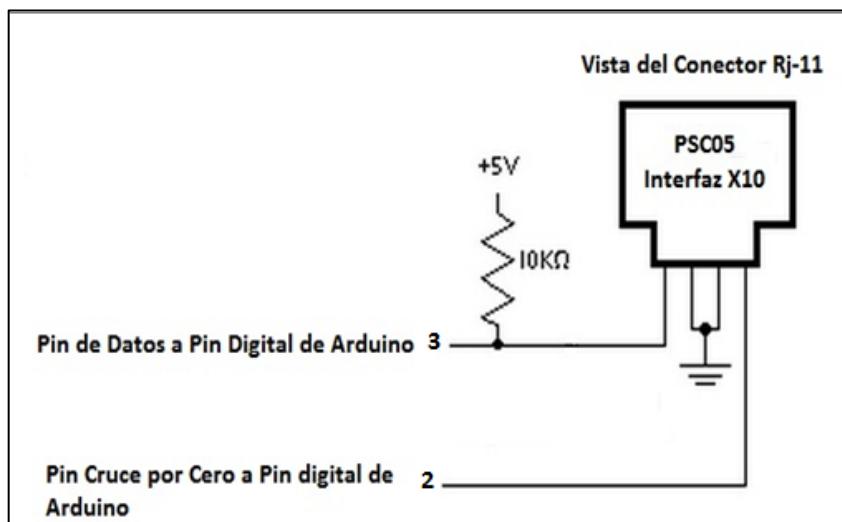


Ilustración 37. Diagrama de conexión

Se puede apreciar en la Ilustración 38 como se realizó la conexión física con el micro controlador Arduino Mega 2560.

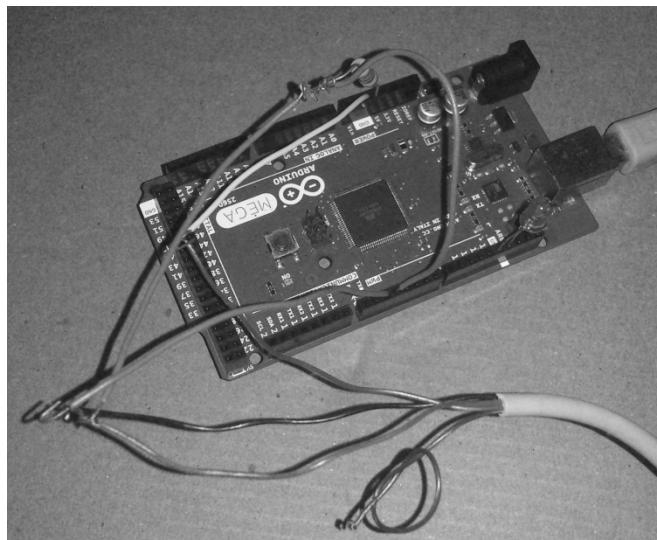


Ilustración 38. Arduino con los hilos del cable telefónico conectados

A continuación se muestra un diagrama de todo el sistema domótico.

2.3.4 Diagrama Electrónico del Sistema Domótico

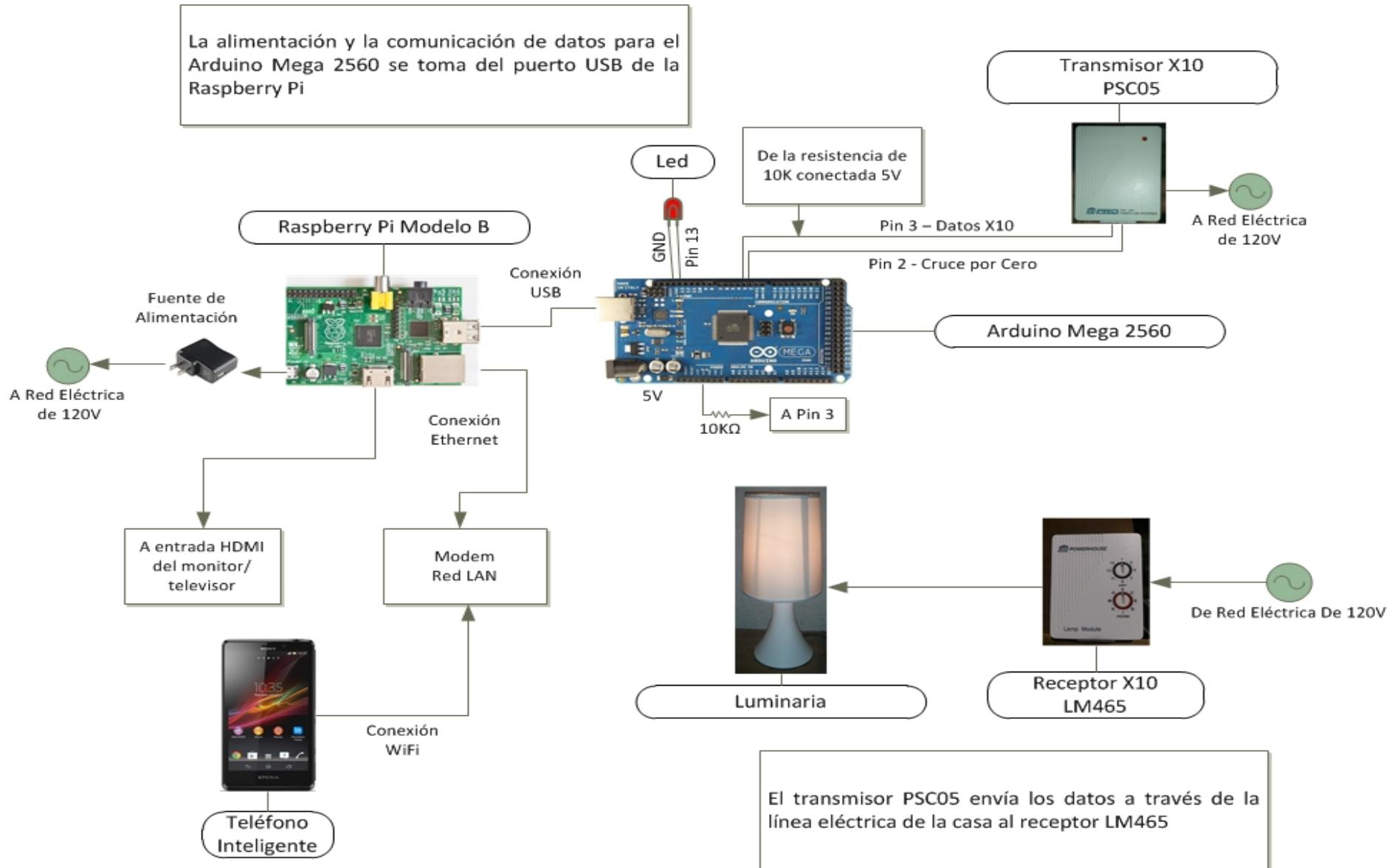


Ilustración 39. Diagrama de Conexión del Sistema Domótico

CAPÍTULO 3

INTEGRACIÓN Y PRUEBAS

3.1 Instalación del Software en Raspberry Pi

En este capítulo se verá la instalación del software necesario para utilizar los programas creados para el proyecto en la microcomputadora Raspberry Pi.

Además se mostrará la prueba del sistema completo.

3.1.1 Sistema Operativo

Para comenzar con Raspberry Pi se necesita un sistema operativo. El Noobs (New Out Of the Box Software – Recién salido de la caja) es un administrador de instalación de sistemas operativos fácil de utilizar.

Se puede conseguir comprando una tarjeta SD con el software preinstalado.

También se puede bajar gratuitamente desde la página de descargas de la fundación (Raspberry Pi Foundation, 2014).

3.1.1.1. Instalando Noobs en una tarjeta SD

Cuando se termine de descargar el archivo ZIP que contiene el programa; se extrae el contenido a una tarjeta SD formateada con FAT 32.

Una vez que los archivos estén en la tarjeta SD se inserta en la ranura de la Raspberry Pi (ver Ilustración 40).

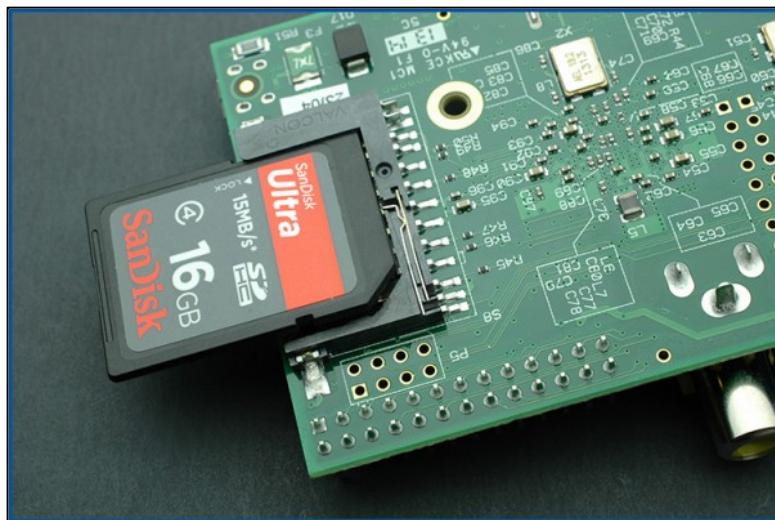


Ilustración 40. Espacio para tarjeta SD

3.1.1.2. Primer Arranque.

- Se conecta el teclado, ratón y el monitor a la computadora.
- Se conecta la fuente de poder en la ranura micro-USB.
- La Raspberry Pi se inicializa, y aparece una ventana con una lista de los diferentes sistemas operativos que se pueden instalar (ver Ilustración 41).

Para este proyecto se utilizó “Raspbian” por recomendación de la fundación Raspberry.

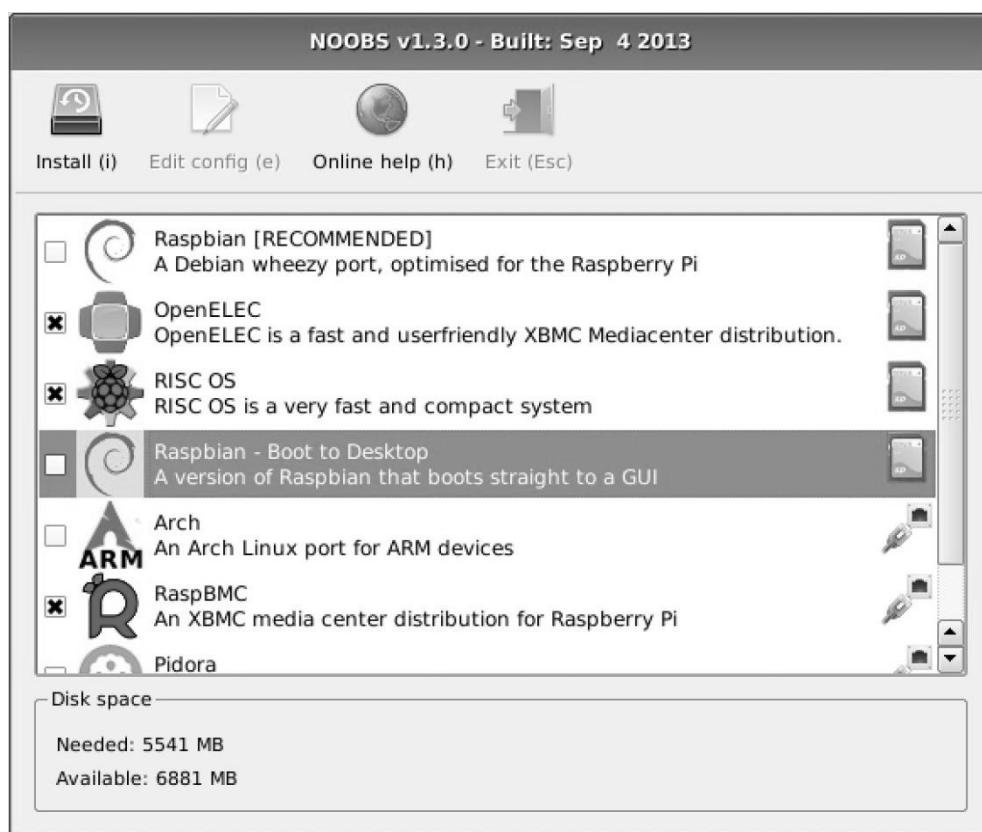


Ilustración 41. Menú de selección de S.O.

- Raspbian corre su proceso de instalación.
- Cuando el proceso de instalación se completa, la Raspberry Pi carga un menú de configuración (raspi-config). Aquí se puede cambiar la hora y la fecha de acuerdo a la región en la que se encuentre (ver Ilustración 42).
- El usuario por default es “pi” (sin las comillas) y la contraseña “raspberry” (también sin comillas).

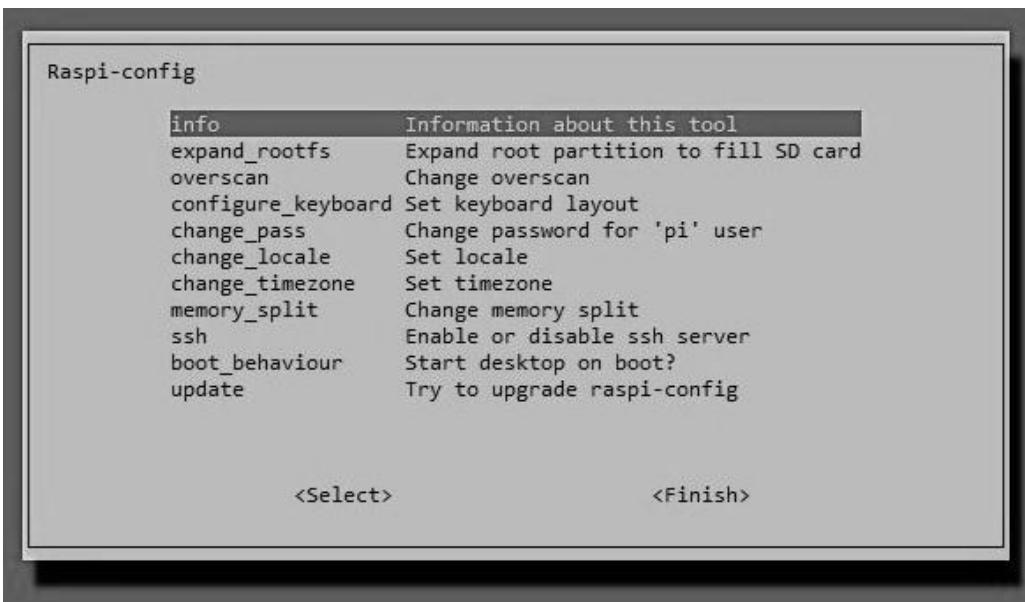


Ilustración 42. Menú de configuración Raspbian

3.1.2 Instalación del entorno de programación Arduino

Asumiendo que se cuenta con una conexión a internet, se escribe lo siguiente en la ventana de comandos o terminal:

```
sudo apt-get install arduino
```

Este proceso instalará lo necesario para que el entorno Arduino funcione (ver Ilustración 43).

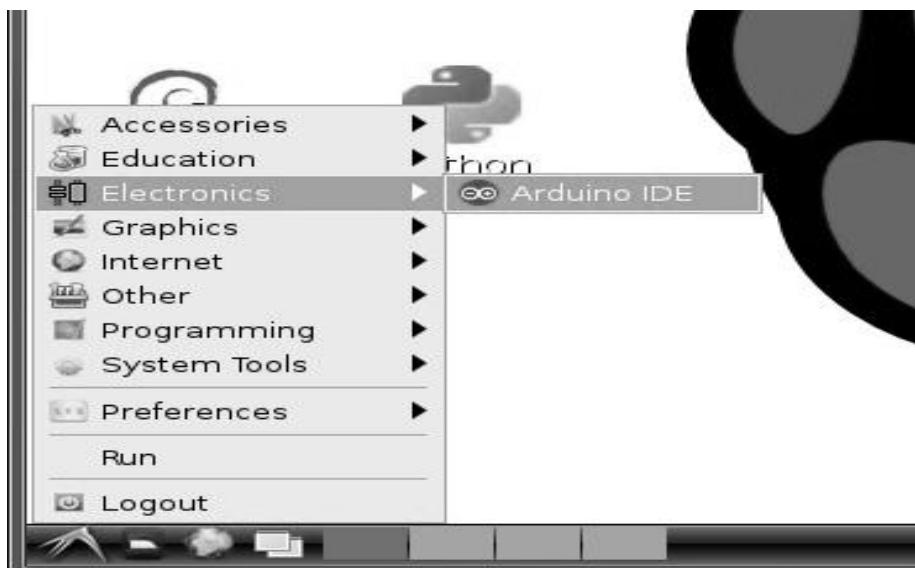


Ilustración 43. Entorno Arduino instalado en Raspbian

3.1.3 Instalación de Django y complementos de Python

El lenguaje de programación Python viene instalado por default en el sistema operativo Raspbian (Así como en todas las distribuciones Linux), pero hay componentes que se ocupan para instalar Django (instalador de paquetes) y también se necesitan para que la aplicación web se pueda comunicar con el micro controlador Arduino por medio del puerto serial.

3.1.3.1 Instalador de paquetes PIP

PIP es un instalador de paquetes Python escrito por Ian Bicking (Bicking). Puede instalar paquetes, listar paquetes instalados y actualizarlos.

Es un reemplazo para “easy_install”, que es el instalador de paquetes que viene instalado por default en Linux.

Para instalarlo simplemente se escribe en la terminal:

```
sudo apt-get install python-pip
```

3.1.3.2 Complementos de Python: Pyserial y Pyfirmata

- Pyserial es un módulo de Python que encapsula el acceso al puerto serial.

Para instalarlo se utiliza el instalador PIP y se escribe lo siguiente en la terminal:

```
pip install pyserial.
```

- Pyfirmata es una interfaz para Python del protocolo Firmata.

Se instala con PIP y se escribe lo siguiente:

```
pip install pyfirmata.
```

3.1.3.3. Instalación de Django.

Como ya se ha comentado, Django es un framework escrito en Python que permite la realización de aplicaciones web de alta calidad con un mínimo esfuerzo.

- Para instalarlo se utiliza PIP y se escribe lo siguiente en la terminal:

```
pip install Django
```

- Con la siguiente instrucción se crea un proyecto en Django:

```
django-admin.py startproject mysite
```

Donde “mysite” es el nombre del proyecto.

- Para crear una aplicación web se debe entrar en la ubicación del proyecto desde la terminal y escribir la siguiente instrucción:

```
python manage.py startapp polls
```

Donde “polls” es el nombre de la aplicación.

- Viendo el directorio queda de la siguiente manera (ver Ilustración 44).

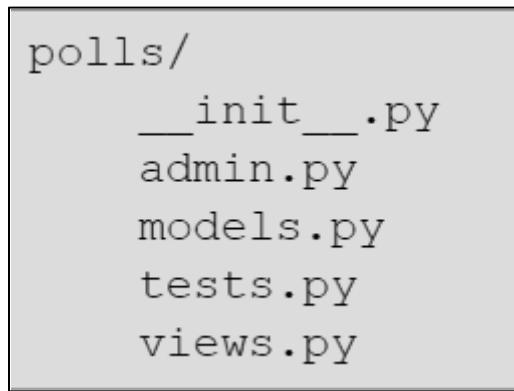


Ilustración 44. Directorio del proyecto

3.2 Prueba del sistema

Para esta prueba se utilizó una placa Arduino Mega 2560 (se puede utilizar cualquier placa, en esta prueba se utilizó este microcontrolador porque es el más apropiado). En la Ilustración 45 se observa como conectan los hilos del cable telefónico de acuerdo al diagrama (ver capítulo anterior).

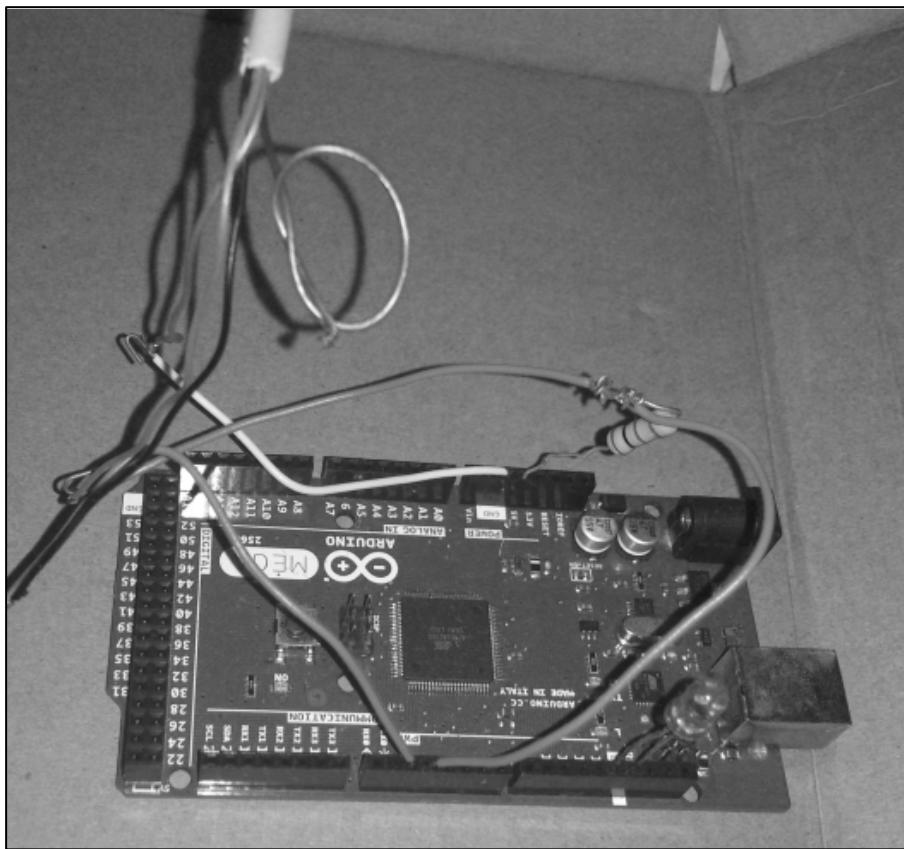


Ilustración 45. Conexión Arduino – Transmisor PSC05

El otro extremo del cable se conecta al módulo PSC05 y se enchufa a la corriente eléctrica (ver Ilustración 46).



Ilustración 46. Módulo PSC05 con cable telefónico

Se conecta el módulo LM465 a la lámpara (Se ocupó una lámpara metálica con pantalla de tela y un foco de 40W) como se muestra en la Ilustración 47, se ajustó el código X10 (Unidad 1 - Casa A para esta prueba) y también se enchufó a

la línea eléctrica (debe ser la misma fase para que exista una correcta comunicación).



Ilustración 47. Lámpara y configuración del módulo LM465

Se toma la Raspberry Pi y se le conectan los cables necesarios (HDMI, Ethernet, etc.). Ver Ilustración 48.

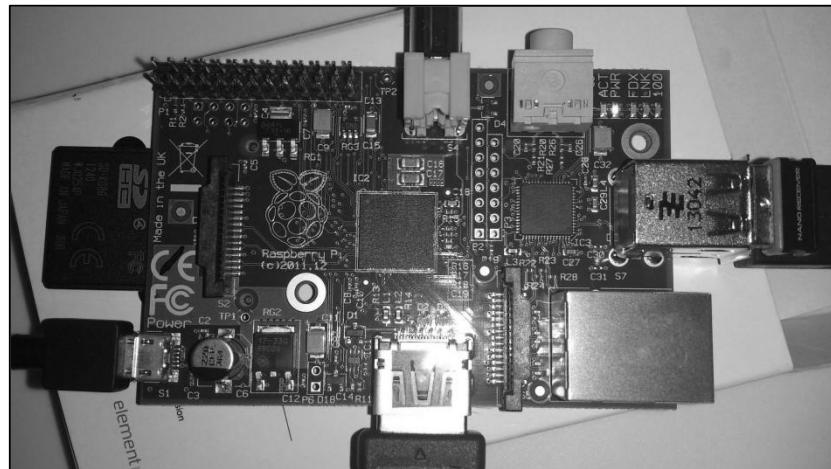


Ilustración 48. Raspberry Pi conectada como servidor

Se conecta al televisor y se energiza con el cable micro-USB (ver Ilustración 49).



Ilustración 49. Fuente de alimentación y cable USB - micro USB

Una vez que el sistema operativo este ejecutándose, se procede a conectar la placa Arduino por medio de un cable USB (ver Ilustración 50, modo gráfico).

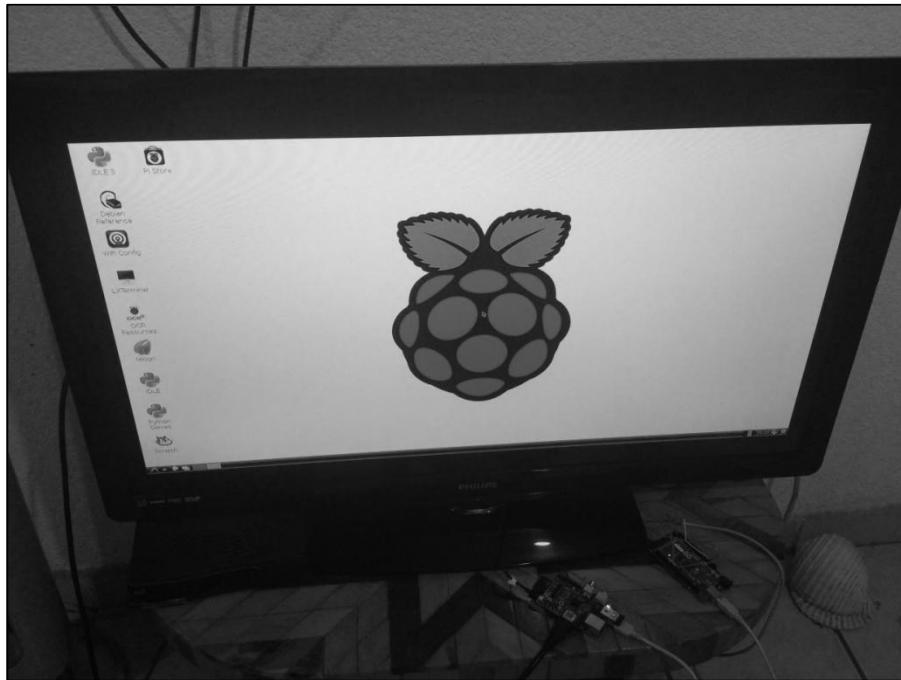


Ilustración 50. Raspberry Pi y Arduino conectados a un televisor digital con entrada HDMI

Una vez que se conectó la tarjeta, se procedió a cargar el programa que contiene el protocolo Firmata y los códigos X10 por medio del entorno de programación Arduino. En el menú “Archivo”, se siguió la ruta “Ejemplos/Firmata/StandardFirmata” (ver Ilustración 51), de esta manera se pudo obtener el programa que contiene el protocolo del mismo nombre y se realizó la modificación de tal forma que pudiera ser capaz de enviar y recibir códigos X10. (Ver Ilustración 52).

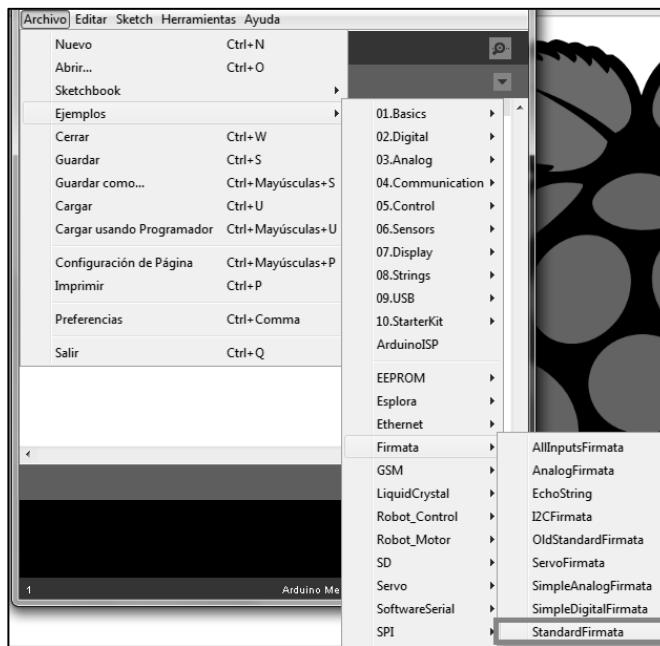


Ilustración 51. Ruta seguida para obtener el programa "StandardFirmata"

Cargado el programa, se abrió la carpeta donde se guardó el proyecto escrito en Django desde la ventana de comandos y se tecleó lo siguiente:

```
python manage.py runserver dirección_ip_raspberrypi:puerto
```

Para esta prueba la dirección IP es: 192.168.1.67 y el puerto que se ocupó fue el 8000.

Se mostró un mensaje que indicó que el servidor se estaba ejecutando y los archivos que se estuvieron mostrando vía web (ver Ilustración 53).

```

control_lampara_3
checkDigitalInputs();

/* SERIALREAD - processing incoming message as soon as possible,
 * checking digital inputs. */
while(Firmata.available())
    Firmata.processInput();

var = digitalRead(ledPin);

if (var == HIGH){

    //Envia datos al modem PSC05 conectado al Arduino
    myHouse.write(HOUSE_A,UNIT_1,RPT_SEND);
    myHouse.write(HOUSE_A,ON,RPT_SEND);

}
if (var == LOW){}

//Envia datos al modem PSC05 conectado al arduino
myHouse.write(HOUSE_A,UNIT_1,RPT_SEND);
myHouse.write(HOUSE_A,OFF,RPT_SEND);


```

Carga terminada.

Tamaño binario del Sketch: 18,026 bytes (de un máximo de 258,048 bytes)

Arduino Mega 2560 or Mega ADK on /dev/ttyACM0

Ilustración 52. Programa en Arduino con Firmata y X10 cargado a la placa

```

root@Nebuchadnezzar:/media/Rufistofeles/Users/Rufistofeles!!!/Dropbox/Documentos/proyecto de tesis/mysite# python manage.py runserver 192.168.1.67:8000
Validating models...
0 errors found
June 21, 2014 - 05:36:23
Django version 1.6.5, using settings 'mysite.settings'
Starting development server at http://192.168.1.67:8000/
Quit the server with CONTROL-C.
[21/Jun/2014 05:37:22] "GET /polls HTTP/1.1" 301 0
[21/Jun/2014 05:37:22] "GET /polls/ HTTP/1.1" 200 1297
[21/Jun/2014 05:37:23] "GET /static/polls/estilos.css HTTP/1.1" 200 3080
[21/Jun/2014 05:37:23] "GET /static/polls/logo.png HTTP/1.1" 200 29131
[21/Jun/2014 05:37:23] "GET /static/polls/normalize.css HTTP/1.1" 200 7546
[21/Jun/2014 05:38:01] "GET /polls/1/ HTTP/1.1" 200 1788
[21/Jun/2014 05:38:18] "POST /polls/1/vote/ HTTP/1.1" 302 0
[21/Jun/2014 05:38:19] "GET /polls/1/results/ HTTP/1.1" 200 1520

```

Ilustración 53. Servidor ejecutándose en la terminal de Raspbian

Después, se tomó un dispositivo móvil (en esta prueba se utilizó un teléfono inteligente Sony Xperia T - Android) y se tecleó la dirección IP del sistema en el navegador del teléfono inteligente, de la siguiente manera:

192.168.1.67:8000/polls (ver Ilustración 54).

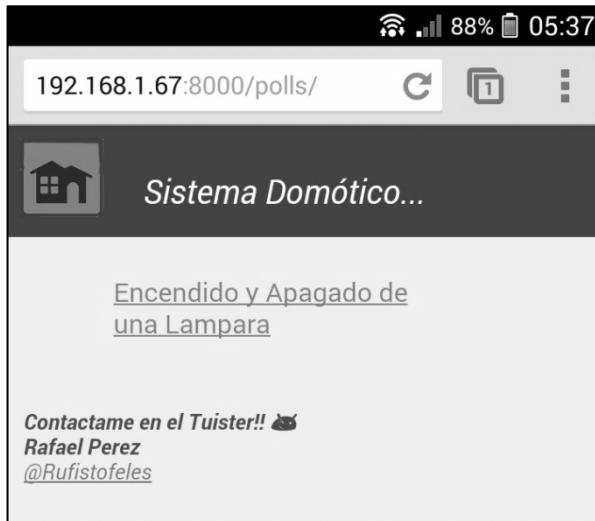


Ilustración 54. Vista desde teléfono inteligente

Se presentó el inicio de la aplicación y se eligió entrar en el enlace “Encendido y apagado de una lámpara”.

Una vez que se cargó la opción que se eligió, se muestran las opciones que se tienen para controlar la lámpara (encender y apagar, ver Ilustración 55).



Ilustración 55. Pantalla de selección

Cuando se seleccionó una opción y se dio clic en el botón “enviar”, el sistema muestra la opción que se eligió y el número de veces que ha sido escogida (solo para llevar un registro), ver Ilustración 56.

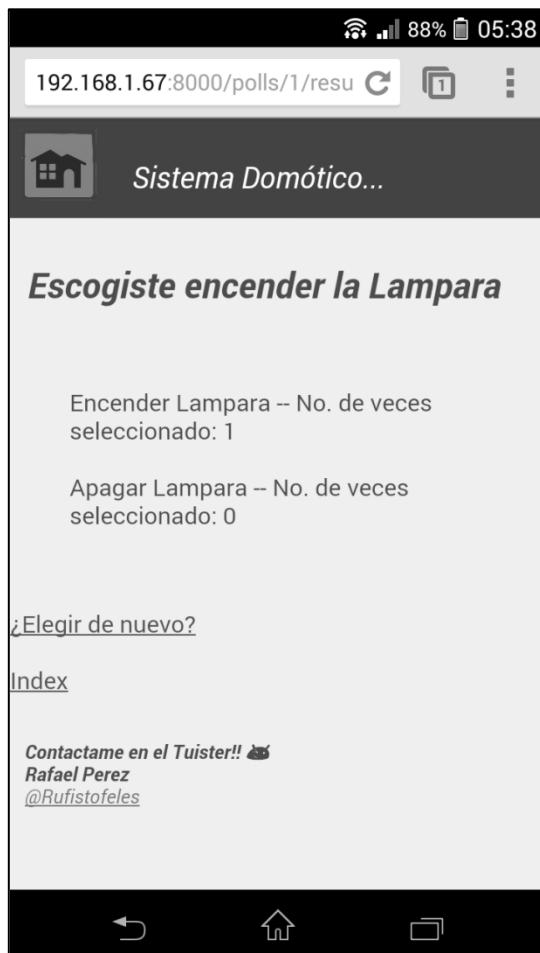


Ilustración 56. Selección y registro de la opción elegida.

Los archivos del sistema están disponibles desde el repositorio GitHub del autor de esta tesis.

<https://github.com/Rufistofeles>

Conclusiones

Viendo las dificultades que tiene una persona que padece artrosis de rodilla para trasladarse de un sitio a otro y encender las luminarias de su casa, surgió la necesidad de querer crear algo que ayudase a realizar esta acción. De esta manera se dio inicio a este proyecto.

Al principio no se contaba con la seguridad de tener los conocimientos suficientes para empezar a diseñar un proyecto, pero investigando en internet sobre distintos temas tecnológicos, se comenzó a ganar más confianza y deseos de lograr producir algo; fue entonces cuando se tomó la decisión de aprender más sobre lenguajes de programación, manejo de aplicaciones web y hardware, de esta manera, se podría tener un panorama más amplio de lo que se podría construir.

Para este proyecto se investigaron los siguientes temas:

- Programación de microcontroladores Arduino.
- Manejo de sistemas operativos basados en GNU/Linux.
- Uso de la mini computadora Raspberry Pi y su distribución Linux “Raspbian”
- Lenguaje de programación Python.
- Manejo de módulos comerciales X10 (transmisores y receptores).
- Creación de aplicaciones web con Django.

Poco a poco y mediante la búsqueda de información en libros, manuales y cursos en línea se pudo obtener más destreza y habilidad en los temas mencionados antes.

Los resultados de las pruebas fueron satisfactorios ya que se logró el control de una luminaria a través del sistema que se desarrolló a lo largo de este proyecto. El sistema se desempeña muy bien tanto en computadoras como en dispositivos móviles, ajustándose a la gran variedad de anchos de pantalla y a los diferentes navegadores.

El sistema está diseñado para funcionar en una red local, pero también se puede ingresar por una vía remota a través de internet. Lo único que se necesita hacer, es asignar la dirección IP pública del modem a la minicomputadora Raspberry Pi. De esta manera la dirección a ingresarse en el navegador del dispositivo es la siguiente:

Dirección_IP_publica:puerto/pagina_del_sistema

Ejemplo: *201.145.110.178:8000/polls*

Al final, se logró diseñar y construir un sistema domótico sencillo, pero que tiene capacidades muy amplias, aun se pueden desarrollar muchas utilidades, no solo en el control de luminarias, también se podría sumar el control de aparatos electrodomésticos, alarmas de seguridad, etc.

Este sistema se construyó con la pretensión de ayudar a una persona en específico y mejorar su calidad de vida, pero se seguirá trabajando para hacerlo crecer y que pueda ser de gran ayuda para muchas personas más. Cabe señalar la importancia que tuvieron los conocimientos previos generales y que se adquirieron a lo largo de la carrera.

Recomendaciones

Se observó que la Raspberry Pi, solo cuenta con 2 puertos USB y se necesita mínimo 3 para conectar de forma satisfactoria los aditamentos (tales como, la tarjeta Arduino, teclado y mouse), indudablemente se producirán modelos en un futuro no muy lejano que incluirán más puertos, pero hasta que ese momento llegue, se recomienda el uso de un adaptador de puertos USB para aumentar esta cantidad.

También se recomienda incluir más luminarias a controlar, así el usuario tendrá más comodidad y no tendrá que esforzarse y desplazarse a otros lugares de su hogar solo para apagar o encender una luminaria.

Se necesita idear una forma sencilla para que al encender el servidor (mini computadora) el sistema se ejecute automáticamente y que no necesite de la manipulación del usuario cada vez que se reinicie el servidor.

No obstante lo anterior, este prototipo es la punta de lanza para ser de ayuda a la comunidad donde vivimos.

Bibliografía

- Beaskoetxea, U. (26 de Enero de 2011). Domótica para Viviendas Construidas. *Titulación*. Pamplona, España: Universidad Pública de Navarra.
- Centro Nacional de Excelencia Tecnológica en Salud. (2009). Guía de Práctica clínica para el Diagnóstico y Tratamiento de la Osteoartritis de Rodilla. México, DF.
- DTIES, D. T. (2007). Motivos de Demanda de Consulta Externa. Unidad de Investigación, Educación y Políticas de Salud. División de Prestaciones Médicas IMSS 2007. México DF.
- Enríquez Herrador, R. (13 de Noviembre de 2009). Guía de Usuario de Arduino. Córdoba, España: Universidad de Córdoba.
- Guanoluisa, L. (Enero de 2012). Implementación de Sistemas de Telecomunicaciones para las Redes de Datos de las Empresas Públicas y Privadas del País. Quito.
- Hijano, A. (12 de Enero de 2011). Proyecto de Instalación Eléctrica y Domótica en una Vivienda Unifamiliar. *Proyecto Final de Carrera*. Cataluña, España: Universitat Politècnica de Catalunya.
- Holovaty, A., & Kaplan-Moss, J. (2007). *The Django Book*. The Django Foundation.
- Huidobro Moya, J. M., & Millán Tejedor, R. J. (2008). *Domótica: Edificios Inteligentes*. Creaciones Copyright.
- Infantes, J. (25 de Enero de 2009). Descripción de X10. Malaga, España.
- Maestro, J. A. (2009). Domótica e Inmótica. (pág. 2). Madrid: Universidad Antonio de Nebrija.
- Martín, H., & Sáez, F. (2006). Domótica: Un Enfoque Sociotécnico. (pág. 16). Madrid, España: Universidad Politécnica de Madrid.

- Marzal, A., & Gracia, I. (2009). *Introducción a la programación con Python*. Universitat Jaume I.
- McAlindon, T., Cooper, C., & Dieppe, P. (1993). Determinants of Disability in Osteoarthritis of the Knee. *Annals of the Rheumatic Diseases*, 258-262.
- Raspberry Pi. (9 de Agosto de 2012). Documentos Técnicos. *Raspberry Pi AAB*.
- Redolfi, L. (2013). Domótica. *Users*, 104.
- Roqué, A. (20 de Febrero de 2005). Diseño y Desarrollo Parcial de un Sistema Domótico para Facilitar la Movilidad de Minusválidos. *Trabajo Final de Carrera*. Cataluña, España: Universitat Politècnica de Catalunya.
- Solánis, D. (Mayo de 2005). Las Nuevas Tecnologías al Servicio de los Mayores. Castellón de la Plana, Castellón de la Plana, España: Universitat Jaume I.
- Steiner, H.-C. (2009). Firmata: Towards making microcontrollers act like extensions of the computer.

Mesografía

Active Homes. (2013). *Active Homes*. Recuperado el 30 de Junio de 2014, de Domótica: Ventajas: <http://activehomes.info/ventajas.htm>

American Academy of Orthopaedic Surgeons. (s.f.). *Arthritis of the Knee*. Recuperado el 13 de Septiembre de 2013, de OrtholInfo: <http://orthoinfo.aaos.org/topic.cfm?topic=a00212>

Arduino.cc. (s.f.). *Arduino*. Recuperado el 5 de Abril de 2013, de Arduino.cc: <http://www.arduino.cc/es/>

Bicking, I. (s.f.). *Projects*. Recuperado el 30 de Junio de 2014, de <http://www.ianbicking.org/projects.html>

Blitz. (2012, Abril 27). ¿Ud. Sábe qué es la Domótica? Retrieved Abril 5, 2013, from blitz Ingeniería: <http://www.blitzingenieria.com/noticia.php?id=8>

Canal AR. (23 de Mayo de 2007). *Expo Casa Domótica, la Automatización del Hogar*. Recuperado el 5 de Abril de 2013, de Canal AR: <http://www.canal-ar.com.ar/Nota.asp?Id=4504>

Cardiel, M. (2002). *Community based study to estimate prevalence, burden of illness and help seeking behavior in rheumatic diseases in Mexico City*. Recuperado el 30 de Junio de 2014, de PubMed.gov: <http://www.ncbi.nlm.nih.gov/pubmed/12412191?dopt=Abstract>

Centers for Disease Control and Prevention. (s.f.). *Arthritis - Related Statistics*. Recuperado el 13 de Septiembre de 2013, de Centers for Disease Control and Prevention: http://www.cdc.gov/arthritis/data_statistics/arthritis_related_stats.htm#6

Chen, P., & Strickler, Y. (2008). *Kickstarter*. Recuperado el 1 de Julio de 2014, de Kickstarter: <https://www.kickstarter.com/>

Control4. (n.d.). *Automatización del hogar y control del hogar inteligente*. Retrieved Junio 30, 2014, from Control4: <http://es-la.control4.com/>

Dinkins, J. (s.f.). *Django Reinhardt*. Recuperado el 1 de Julio de 2014, de Red Hot Jazz: <http://www.redhotjazz.com/django.html>

Django Software Foundation. (2013). *Writing your first Django app*. Recuperado el 30 de Junio de 2014, de Django documentation: <https://docs.djangoproject.com/en/1.6/intro/tutorial01/>

Django Software Foundation. (s.f.). *Django*. Recuperado el 20 de Octubre de 2013, de Django Project: <https://www.djangoproject.com/>

DougC. (s.f.). *arduino-x10*. Recuperado el 5 de Abril de 2013, de GitHub: <https://github.com/DougC/arduino-x10>

Elektron. (2014). *Lista de precios*. Recuperado el 30 de Junio de 2014, de Elektron - Materiales Electricos: <http://www.elektron.com.mx/precios.php>

- Free Software Foundation. (s.f.). *GNU General Public License*. Recuperado el 1 de Julio de 2014, de <http://www.gnu.org/licenses/gpl.html>
- Igoe, T. (s.f.). *X10 Library*. Recuperado el 5 de Abril de 2013, de Arduino: <http://arduino.cc/en/Tutorial/x10>
- King, W. D. (s.f.). *Treatment Options for Osteoarthritis*. Recuperado el 13 de Septiembre de 2013, de Palo Alto Medical Foundation: <http://www.pamf.org/sports/king/osteoarthritis.html>
- La Casa Inteligente: Domótica en México*. (s.f.). Recuperado el 5 de Abril de 2013, de <http://domoticamexico.blogspot.mx/>
- Legrand. (2014). *Domótica My Home: Automatización*. Recuperado el 30 de Junio de 2014, de bticino: <http://www.bticino.es/productos/DomoticaMyHome/Automatizacion.php>
- Liechti, C. (s.f.). *pySerial*. Recuperado el 10 de Octubre de 2013, de pySerial 2.7 Documentation: <http://pythonhosted.org/pyserial/pyserial.html>
- Lindes. (2013). *Vidrio Aislante*. Recuperado el 1 de Julio de 2014, de Lindes: <http://www.lindes.mx/vidrio-aislante.html>
- Módulos. (s.f.). Recuperado el 20 de Septiembre de 2013, de X10 México: <http://www.x10.com.mx/13-Modulos>
- Python Software Foundation. (s.f.). *The Python Language Reference*. Recuperado el 20 de Octubre de 2013, de Python 2.7.6 Documentation: <http://docs.python.org/2/reference/>
- Python Software Foundation. (s.f.). *The Python Standard Library*. Recuperado el 20 de Septiembre de 2013, de Python, 2.7.6 Documentation: <http://docs.python.org/2/library/>

Python Software Foundation. (s.f.). *The Python Tutorial*. Recuperado el 20 de Septiembre de 2013, de Python, 2.7.6 Documentation: <http://docs.python.org/2/tutorial/>

Raspberry Pi Foundation. (2014). *Raspberry Pi Foundation*. Recuperado el 30 de Junio de 2014, de Downloads: <http://www.raspberrypi.org/downloads/>

Robodacta.mx. (s.f.). *Arduino*. Recuperado el 20 de Agosto de 2013, de Robodacta. Robótica Didáctica: http://www.robodacta.mx/index.php?dispatch=categories.view&category_id=194

Robodacta.mx. (n.d.). *Raspberry Pi*. Retrieved Octubre 20, 2013, from Robodacta. Robótica Didáctica: http://www.robodacta.mx/index.php?dispatch=categories.view&category_id=314

Steren. (s.f.). *Cable Telefónico de 4 Vías*. Recuperado el 20 de Septiembre de 2013, de Steren México: <http://www.steren.com.mx/catalogo/category.asp?f=3&sf=33&c=358>

Team Ubi. (2012). *Ubi - The Ubiquitous Computer - Voice-Activated & Always On*. Recuperado el 30 de Junio de 2014, de Kickstarter: <https://www.kickstarter.com/projects/607691307/ubi-the-ubiquitous-computer-voice-activated-and-al-0>

Wikipedia. (s.f.). *Domótica*. Recuperado el 5 de Abril de 2013, de <http://es.wikipedia.org/wiki/Dom%C3%B3tica>

Wikipedia. (s.f.). *X10*. Recuperado el 5 de Abril de 2013, de <http://es.wikipedia.org/wiki/X10>

Wikipedia. (s.f.). *ZigBee*. Recuperado el 5 de Abril de 2013, de <http://es.wikipedia.org/wiki/ZigBee>

X - 10. (s.f.). Recuperado el 10 de Abril de 2013, de
<http://www.aquihayapuntes.com/x-10.html>

X10 Home Gadgets. (s.f.). Recuperado el 5 de Abril de 2013, de
<http://www.x10.com/homepage.htm>

ZigBee Alliance. (s.f.). Recuperado el 5 de Abril de 2013, de
<http://www.zigbee.org/Home.aspx>

Anexos

Anexo A

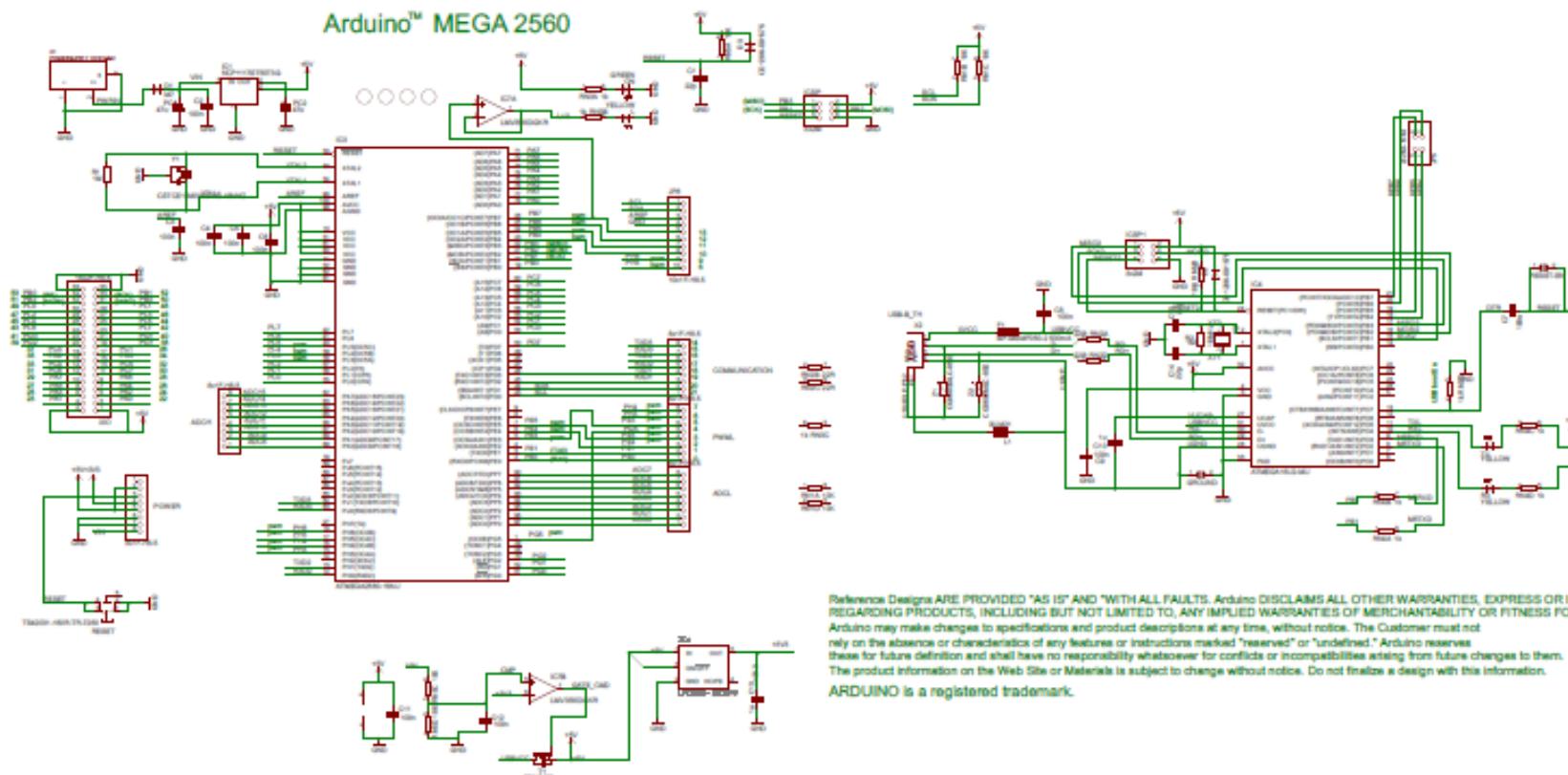


Ilustración 57. Diagrama Esquemático Arduino Mega 2560

Anexo B

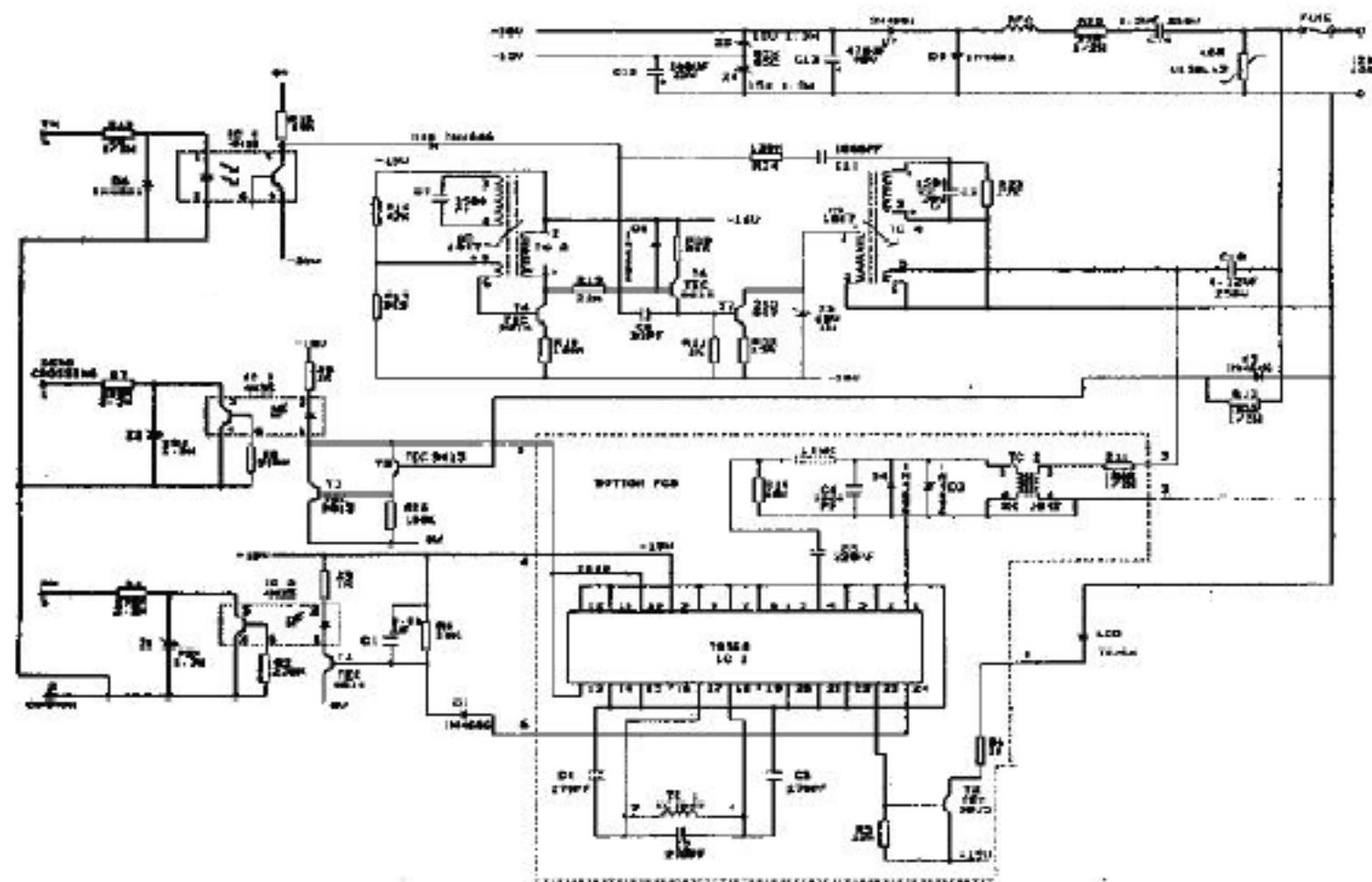


Ilustración 58. Diagrama Esquemático PSC05

Anexo C Tabla comparativa de precios

| Bticino Sistema “My home” | | |
|-----------------------------------|--|---------------------|
| Modelo | Descripción | Precio |
| L4001N | Interruptor sencillo 1 mód. 16 AX, 127-277V | \$58.95 |
| L4915N | Cubre tecla 1 mód. | \$15.83 |
| MH200N | Programador de escenarios | \$10,423.89 |
| F411/1N | Actuador DIN ON/OFF | \$862.95 |
| | TOTAL | \$ 11,361.62 |
| Control4 Sistema “Myhome” | | |
| HC-200 | Controlador | \$3,837.10 |
| C4-DIM1-Z | Wireless Dimmer | \$1,736.79 |
| | Software Licencia 1 equipo | \$2,679.24 |
| | TOTAL | \$8,253.13 |
| Prototipo Sistema Domótico | | |
| MEGA 2560 Rev 3 | Arduino | \$800 |
| B | Raspberry Pi | \$945 |
| LM45 | Modulo para lámpara (receptor) | \$425 |
| PSC05 | Modulo Transmisor | \$899 |
| | TOTAL | \$3,069 |

Tabla 3. Tabla comparativa de precios.

Nota: Los precios de los sistemas Bticino y Control4 se obtuvieron de una búsqueda en internet en diferentes sitios tales como... ebay, amazon, etc.

Para el prototipo se utilizaron los precios de las páginas de internet de donde se adquirieron los productos: robodacta.com.mx y x10-mexico.com.