ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA

Proyecto / Trabajo Fin de Carrera

Enlace Zigbee para la Gestión de un Sistema de Alumbrado Basado en LEDs

(Link ZigBee for the Management of a Lighting System Based on LEDs)

Para acceder al Título de

INGENIERO TÉCNICO INDUSTRIAL
ESPECIALIDAD ELECTRÓNICA INDUSTRIAL

Autor: Sergio Pérez Plaza

Septiembre 2015

ÍNDICE

INTRODUCCIÓN……..4

MEMORIA……..6

1. DETECTORES DE PRESENCIA……..7

1.1 ¿Qué es un detector de presencia?.....7

1.2 Tipos de detectores de presencia……7

1.2.1 Detectores por infrarrojos pasivos (PIR)……7

1.2.2 Detectores por microondas……7

1.2.3 Detectores duales o de doble tecnología……...8

1.3 Principales problemas de los detectores de presencia……8

1.4 Selección del detector de presencia adecuado……8

1.5 Funcionamiento……...8

2. MÓDULOS DE RADIO FRECUENCIA XBEE……9

2.1 ¿Qué son los módulos Xbee?.....9

2.2 Protocolo Zigbee……9

2.3 Características físicas y descripción de pines……10

2.4 Características generales……12

2.5 Modos de operación……13

2.5.1 Modo de reposo (Idle Mode)……14

2.5.2 Modo sleep o de bajo consumo……..14

2.5.3 Modo de recepción……...15

2.5.4 Modo de transmisión……16

2.5.5 Modo comandos……16

3. PLACA CONTROLADORA ARDUINO……..18

3.1 ¿Qué es Arduino?.....18

3.2 Historia……19

3.3 Hardware……20

3.4 Características generales……21

3.5 Tipos y características……..22

3.6 Estructura de programación……35

4. DRIVER……39

4.1 ¿Qué es un driver?.....39

4.2 Funcionamiento……39

5. ILUMINACIÓN LED……41

5.1 Historia……41

5.2 Protocolo DALI……...41

5.2.1 Introducción……41

5.2.2 Funcionamiento……..42

5.3 Actualidad y futuro de la tecnología LED……..44

5.4 Fundamentos del LED……..45

5.5 Eficiencia energética de la iluminación LED……46

5.6 Utilización de los LEDs……48

5.7 Retos de I+D en tecnología LED……49

5.8 Ventajas del LED frente a los halógenos y fluorescentes……..50

5.9 Estrategias de avance en LEDs……..52

6. IMPLEMENTACIÓN DE LA UNIDAD DE CONTROL E ILUMINACIÓN..55

6.1 Configuración de los módulos Xbee……....55

6.2 Envío y recepción de datos entre módulos Xbee……...61

6.3 Objetivo de la programación de Arduino……..66

6.4 Selección de pines……66

6.5 Programación de Arduino……67

6.5.1 Librería de frecuencia segura de PWM…….68

6.5.2 Código de programación……..70

6.6 Implementación de código en Arduino……..72

7. FUNCIONAMIENTO AUTÓNOMO Y CONTROLADO……75

7.1 Interconexión de los componentes……75

7.2 Consecución de ciclo de trabajo……77

7.3 Funcionamiento del sistema completo……..79

7.4 Diseño del PCB……80

PLIEGO DE CONDICIONES……82

PRESUPUESTO……85

BIBLIOGRAFÍA……..94

INTRODUCCIÓN

INTRODUCCIÓN

El proyecto que aquí se presenta trata de un sistema de iluminación LED, cuya intensidad será controlada, o bien, por un ordenador de forma inalámbrica por unos módulos de radiofrecuencia basados en protocolo Zigbee, o bien, por un detector de presencia.

Por lo que podemos funcionar de dos maneras: de manera autónoma, a través del detector de presencia, y regulando el flujo de luz inalámbricamente a través de nuestro ordenador con los módulos Xbee.

Con el detector de presencia la lámpara de LEDs se iluminará tan solo cuando, como bien dice su nombre, se detecte presencia humana en el radio de acción del sensor.

Con los módulos Xbee podremos ordenar a nuestro controlador que la lámpara emita luz a la intensidad que nosotros deseemos independientemente de que el detector sienta presencia humana o no, enviando unas órdenes por radiofrecuencia desde nuestro ordenador hasta una placa controladora.

También es reseñable el ahorro de energía que llevaremos a cabo, ya que, con la iluminación LED, obtendremos ahorros de hasta un 80% en comparación con las lámparas tradicionales, añadiendo también el hecho de que el detector de presencia nos proporciona otro 15% adicional de ahorro energético al no mantener la luminancia encendida en todo momento, sino sólo cuando se necesite.

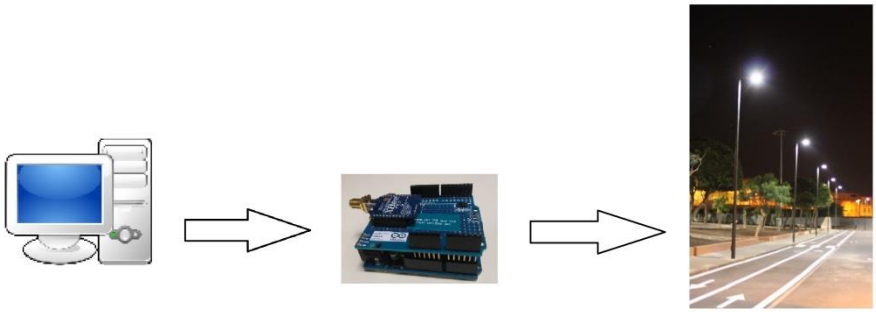


Figura 1.Esquema básico de funcionamiento

MEMORIA

1. DETECTORES DE PRESENCIA

1.1 ¿Qué es un detector de presencia?

Se trata de un sensor cuya función es informarnos de la existencia de personas en su radio de acción.

1.2 Tipos de detectores de presencia

Actualmente, en el mercado, existen detectores basados en distintas tecnologías para detectar la presencia de personas en un espacio determinado.

1.2.1 Detectores por infrarrojos pasivos (PIR):

Los detectores PIR (Passive Infrared) o Infrarrojo Pasivo reaccionan sólo ante determinadas fuentes de energía tales como el calor del cuerpo humano. Básicamente reciben la variación de las radiaciones infrarrojas del medio ambiente que cubre. Se les llama pasivos porque no emiten radiaciones, sino que las recibe. Estos captan la presencia detectando la diferencia entre el calor emitido por el cuerpo humano y el espacio alrededor.

Su componente principal es el sensor piroeléctrico. Se trata de un componente electrónico diseñado para detectar cambios en la radiación infrarroja recibida. Generalmente dentro de su encapsulado incorporan un transistor de efecto campo que amplifica la señal eléctrica que genera cuando se produce dicha variación de radiación recibida.

La información infrarroja llega al sensor piroeléctrico a través de una lente de Fresnel que divide el área protegida en sectores.

1.2.2 Detectores por microondas:

Los detectores por microondas transmiten (son activos) al ambiente una radiación de ondas de una frecuencia comprendida dentro del radio de las microondas.

Estas ondas son recibidas por el mismo dispositivo, el cual se encarga de verificar que se mantengan estables los parámetros de la onda recibida. Si una persona entra en su campo de acción, las ondas se verán reflejadas en ella y regresarán al dispositivo con un desfase de frecuencia (efecto Doppler).

1.2.3 Detectores duales o de doble tecnología:

Son denominados así porque utilizan de manera simultánea las dos tecnologías anteriores. (PIR y microondas).

La finalidad de combinarlos es la optimización máxima para eliminar falsos positivos, ya que solamente en el caso de que ambas tecnologías detectasen presencia de personas se activaría un positivo.

1.3 Principales problemas de los detectores de presencia

La problemática general de estos dispositivos es la generación de falsos positivos por parte del detector lo cual haría que nuestro sistema no funcionase como deseamos.

Las causas principales y más habituales son:

- Factores meteorológicos: tales como la lluvia, viento, nieve, etc.
- Animales salvajes o domésticos.
- Insectos
- Movimientos de elementos externos debidos al viento: ramas de árboles, bolsas de plástico u otros objetos ligeros.

1.4 Selección del detector de presencia adecuado

Dependiendo de cada instalación a realizar es necesario un tipo específico de detector. En este caso, nuestro proyecto está dirigido a instalaciones exteriores, por lo que necesitaremos un detector capaz de minimizar al máximo las falsas alarmas originadas por agentes externos que nos podemos encontrar en la calle.

Por este motivo, el dispositivo que hemos escogido es un detector de doble tecnología preparado para soportar todos los problemas citados en el anterior apartado, por lo que lleva un grado de protección IP 55.

1.5 Funcionamiento

Su funcionamiento es bastante fácil de entender. En el momento que el detector encuentra a alguna persona en su rango de acción envía una señal a su salida digital, la cual seguirá siendo enviada hasta que haya ausencia de detección en su alcance de visión. Una vez que esto último ocurre, se dejará de enviar la señal hasta que, de nuevo, el sensor detecte presencia humana.

2. MÓDULOS DE RADIO FRECUENCIA XBEE

2.1 ¿Qué son los módulos Xbee?

De acuerdo con su fabricante, los módulos Xbee son soluciones integradas que brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos.

Estos módulos utilizan el protocolo Zigbee para crear redes FAST POINT-TO-MULTIPOINT (punto a multipunto); o para redes PEER-TO-PEER (punto a punto). Fueron diseñados para aplicaciones que requieren de un alto tráfico de datos, baja latencia y una sincronización de comunicación predecible. En términos simples, los XBee son módulos inalámbricos fáciles de usar.

2.2 Protocolo Zigbee

Zigbee es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas de área personal (wireless personal area network, WPAN) IEE 802.15.4.Creado por Zigbee Alliance, una organización, teóricamente sin ánimo de lucro, de más de 200 grandes empresas (destacan Mitsubishi, Honeywell, Philips, Invensys, entre otras), muchas de ellas fabricantes de semiconductores. Zigbee permite que dispositivos electrónicos de bajo consumo puedan realizar sus comunicaciones inalámbricas. Es especialmente útil para redes de sensores en entornos industriales, médicos y, sobre todo, domóticos.

Las comunicaciones Zigbee se realizan en la banda libre de 2.4GHz. A diferencia de bluetooth, este protocolo no utiliza FHSS (Frequency hooping), sino que realiza las comunicaciones a través de una única frecuencia, es decir, de un canal. Normalmente puede escogerse un canal de entre 16 posibles. El alcance depende de la potencia de transmisión del dispositivo así como también del tipo de antenas utilizadas (cerámicas, dipolos, etc). La velocidad de transmisión de datos de una red Zigbee es de hasta 256kbps. Una red Zigbee la pueden formar, teóricamente, hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad enorme de dispositivos.

Entre las necesidades que satisface el módulo se encuentran:

- Bajo coste.
- Muy bajo consumo de potencia.
- Uso de bandas de radio libres y sin necesidad de licencias.

- Instalación barata y simple.
- Redes flexibles y extensibles.

El uso del protocolo Zigbee va desde reemplazar un cable por una comunicación serial inalámbrica, hasta el desarrollo de configuraciones punto a punto, multipunto, peer-to-peer (todos los nodos conectados entre sí) o redes complejas de sensores.

2.3 Características físicas y distribución de pines

En esta figura se pueden ver las características de tamaño y organización de pines de los módulos Xbee, tanto en su versión normal como PRO:

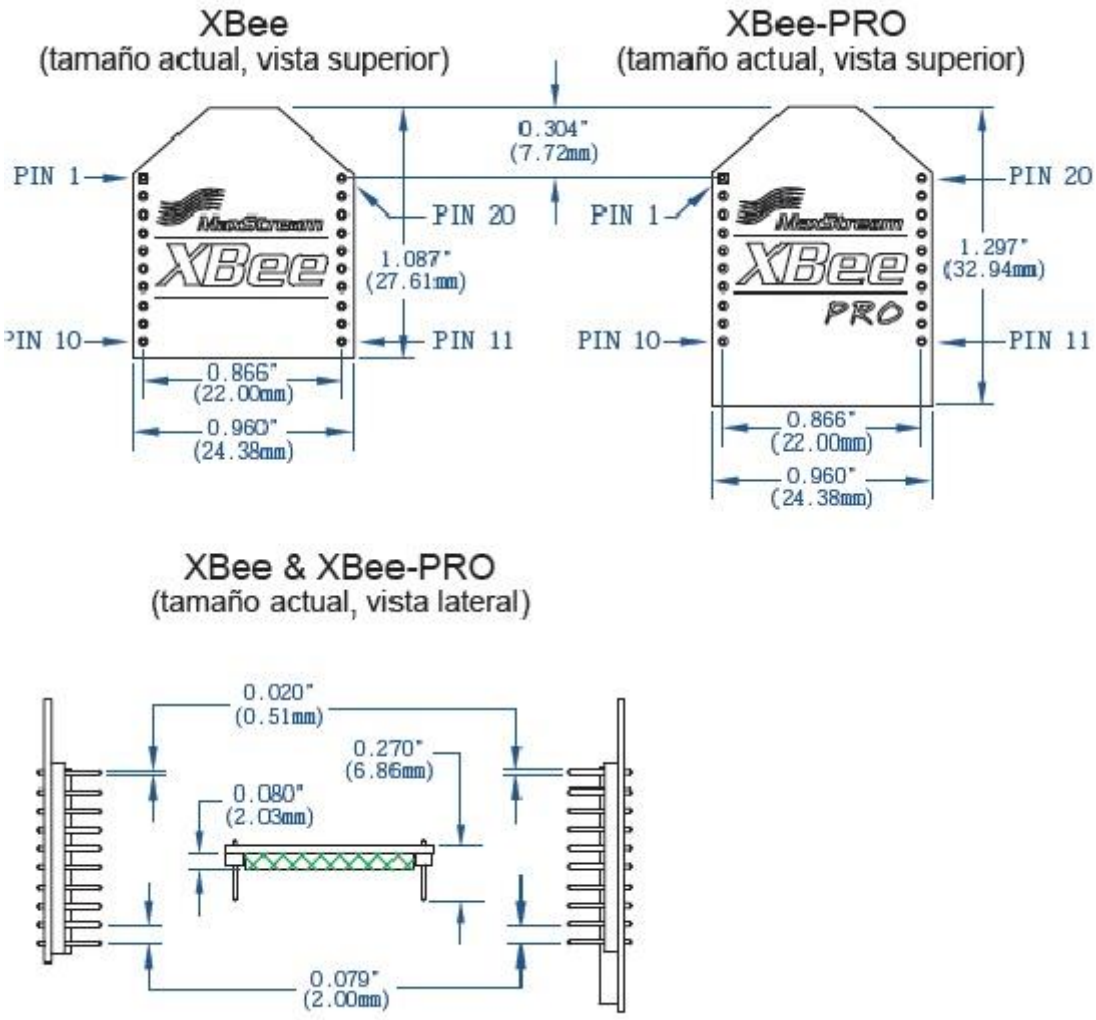


Figura 2.Descipción física del Xbee

Este módulo, a pesar de tener unas reducidas dimensiones (25mm x 27mm), proporciona hasta siete entradas analógicas, nueve entradas/ocho salidas digitales y dos salidas PWM, todo ello en 20 pines. Para ello, permite configurar por software la función de los pines, para que actúen de un modo u otro.

En la siguiente tabla se muestra una disposición de los pines:

Pin #	**Nombre**	**Dirección**	**Descripción**
1	VCC	-	Tensión de alimentación
2	DOUT	Salida	UART Salida de datos
3	DIN/$\overline{\text{CONFIG}}$	Entrada	UART Entrada de datos
4	DO8*	Salida	Salida digital 8
5	RESET	Entrada	Modo reset
6	PWM0/RSSI	Salida	Salida PWM0 / Indicador de señal recibida
7	PWM1	Salida	Salida PWM 1
8	[reservado]	-	No conectado
9	$\overline{\text{DTR}}$/ Sleep _ RQ/DI8	Entrada	Pin de control de línea de sleep o Entrada digital 8
10	GND	-	Tierra
11	AD4/DIO4	Entrada/Salida	Entrada analógica 4 o entrada/salida digital 4
12	$\overline{\text{CTS}}$ /DIO7	Entrada/Salida	Control de flujo o entrada/salida digital 7
13	ON / $\overline{\text{SLEEP}}$	Salida	Indicador módulo de estado
14	VREF	Entrada	Referencia de voltaje para entradas analógicas / digitales
15	ASSOCIATE/ AD5 / DIO5	Entrada/Salida	Indicador asociado / entrada analógica 5 o entrada/salidas digital 5
16	RTS / AD6 / DIO6	Entrada/Salida	Solicitud de envío de control de flujo / entrada analógica 6 o entrada digital 6
17	AD3 / DIO3	Entrada/Salida	Entrada analógica 3 o entrada/salida digital 3
18	AD2 / DIO2	Entrada/Salida	Entrada analógica 2 o entrada/salida digital 2
19	AD1 / DIO1	Entrada/Salida	Entrada analógica 1 o entrada/salida digital 1
20	AD0 / DIO0	Entrada/Salida	Entrada analógica 0 o entrada/salida digital 0

Tabla 1.Disposición de pines del Xbee

2.4 Características generales

ESPECIFICACIONES		XBEE	XBEE PRO
Rendimiento	Alcance en ambientes interiores /zonas urbanas	Hasta 30m	Hasta 100m
	Alcance de RF en zona de visión para ambientes exteriores	Hasta 100m	Hasta 1200m
	Potencia de salida de transmisión	1mW (0db)	60mW (18dB) ,100mW EIRP
	Régimen RF de datos	250000 bps	250000 bps
	Sensibilidad del receptor	-92 dBm (1% PER)	-100 dBm (1% PER)
Rendimientos de potencia	Rango de alimentación	2.8--3.4 V	2.8--3.4 V
	Corriente de transmisión	45 mA a 3.3V	270 mA a 3.3V
	Corriente de recepción	50 mA a 3.3V	55 mA a 3.3V
	Corriente Power-Down	<10 µA	<10 µA
Información general	Frecuencia	ISM 2.4 MHz	ISM 2.4 MHz
	Dimensiones	2.438 cm x 2.761 cm	2.438 cm x 3.292 cm
	Tª de operación	-40 a 85º (industrial)	-40 a 85º (industrial)
	Opciones de antena	Conector U.FL, Antena en chip o antena cable.	Conector U.FL, Antena en chip o antena cable.
Trabajo en red	Topologías permitidas en la red	Punto a Punto, Punto a multipunto, entre pares y Mesh	Punto a Punto, Punto a multipunto, entre pares y Mesh
	Número de canales	16 canales de secuencia directa	12 canales de secuencia directa
	Capas de filtración de la red	PAN ID y direcciones 64 bits	PAN ID y direcciones 64 bits

Redes y seguridad	Reintentos y confirmaciones DSSS (Direct Sequence Spread Spectrum) Tienen alrededor de 65000 direcciones disponible de red Direccionamiento Fuente/Sumidero Comunicaciones de Broadcast (todos dentro de la red) o a un destinatario.
Otras características	Soporte de entradas analógicas y digitales I/O Line parsing (Cableado virtual de entrada salidas) Fácil de usar. Preconfigurado para usarse como transmisor RF por el puerto serie. El software de configuración (X-CTU) es gratuito. Soporte de comandos AT y API para la configuración de los módulos

Tabla 2.Características generales del Xbee

2.5 Modos de operación

Los módulos XBee están preparados para funcionar en cinco modos de operación o estados, según la tarea que han de realizar: modo de transmisión, modo de recepción, modo de entrada de comandos, modo sleep, en el que el consumo de energía es bajo y modo de reposo (Idle mode).Este último modo es al que se retorna de los demás, como se muestra en la siguiente figura:

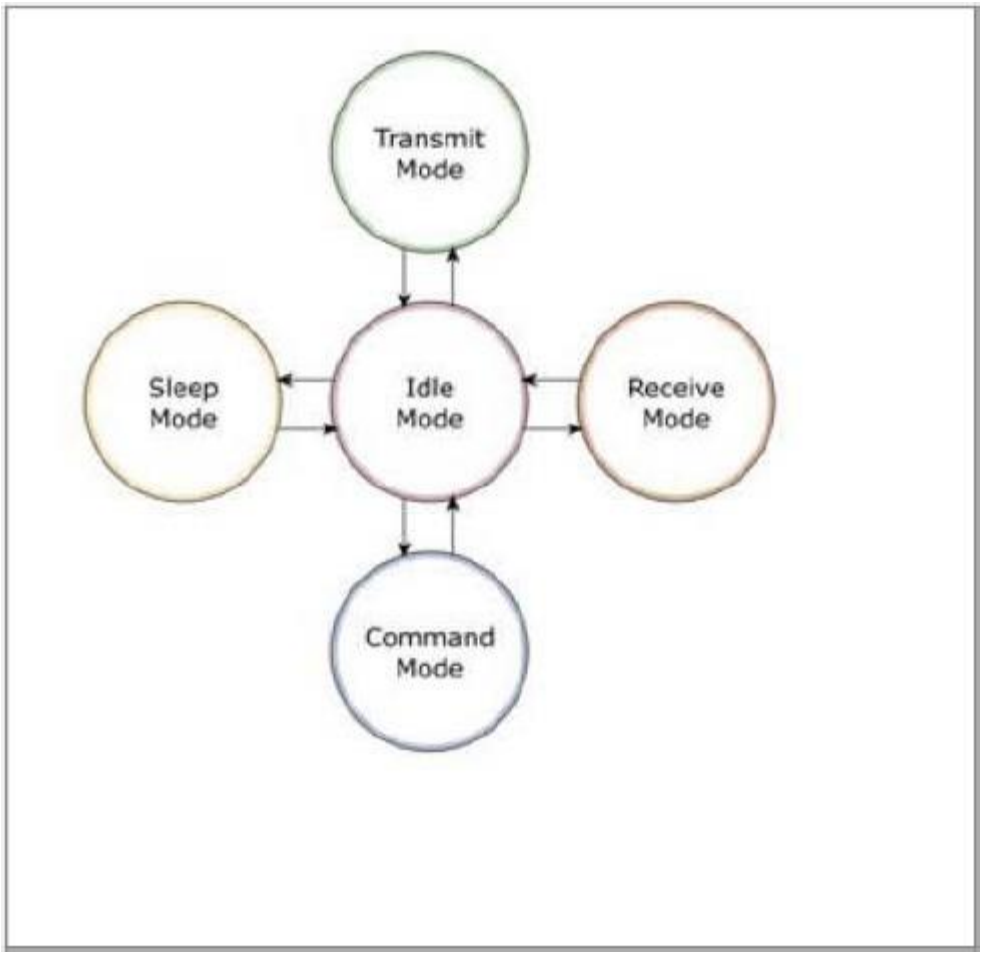


Figura 3.Diagrama de modos de operación del Xbee

A continuación se explica cada uno de los distintos modos del módulo Xbee:

2.5.1 Modo de reposo (Idle Mode): El módulo estará siempre en este estado cuando este activo y no esté en ninguno de los otros modos; ni transmitiendo, ni recibiendo, ni en el modo de comandos, ni ahorrando energía en modo sleep. Desde este modo, se puede cambiar a cualquiera de los demás modos, y es al que se vuelve siempre que termina cualquiera de los otros modos. Se consumen 50 mA en este modo.

2.5.2 Modo sleep o de bajo consumo: El módulo estará en un estado de bajo consumo cuando no es usado, dícese, cuando no se envía ni se transmite nada. Puede entrar a actuar en este modo por recibir una señal en el pin 9 (*Sleep RQ*), petición de bajo consumo o por software, si se ha cumplido el tiempo del parámetro ST (*Time before Sleep*).El consumo en este modo está por debajo de los 50µA.

A su vez estos módulos disponen de 3 modos de suspensión diferentes que se pueden configurar con el parámetro SM:

- Hibernación (SM=1):

Es el modo de menor consumo, menor de 10 µA. Este menor consumo tiene como contra el mayor tiempo que tarda en pasar al estado activo. Sólo se puede pasar de este modo a estado activo desactivando la entrada al pin 9 (*Sleep_RQ)*.

- Suspendido (SM=2):

En este modo es más rápido su vuelta a modo activo, pero su consumo es unas 5 veces más elevado que el anterior, alrededor de 50µA.

- Cíclico (SM=4,5):

El módulo entra y sale del modo suspendido según el tiempo definido en sus parámetros (ST- *Time before Sleep* y SP- *Cyclic Sleep Period*). En este modo tiene el mismo consumo y el tiempo de vuelta a modo activo que en el caso anterior, pues en los periodos de inactividad está en el mismo estado suspendido.

Hay que aclarar que el ahorro de energía en este modo depende enormemente de la tensión de alimentación. Cuanto mayor es la tensión (por encima de los 3V) el consumo en este modo sleep aumenta de manera exponencial con el aumento de la tensión de entrada.

En esta tabla se muestran los consumos de energía en cada uno de los modos anteriores:

Modo	Consumo alimentación			Modo Sleep	Modo Wake-up
	2.8 – 3 V	3.2 V	3.4V		
SM=1	<3 uA	32 uA	255 uA	Sleep_RQ	Sleep_RQ
SM=2	<35 uA	48 uA	170 uA	Sleep_RQ	Sleep_RQ
SM=3	(reservado)			(reservado)	(reservado)
SM=4	<34 uA	49 uA	240 uA	Comando ST	Comando SP
SM=5	<34 uA	49 uA	240 uA	Comando ST	Sleep RQ

Tabla 3.Consumos de energía en modo sleep

- **Modo de recepción:**

Se encuentra en este modo cuando el módulo recibe algún paquete de datos RF por la antena. Consume 50mA.

- **Modo de transmisión:**

Se encuentra en este modo cuando se envía información al pin 3 del módulo (UART Data In) que después será transmitida.

Los datos pueden ser enviados de dos maneras: método directo e indirecto.

- Método directo:

Los paquetes son enviados inmediatamente a su dirección de destino.

- Método indirecto:

El paquete es puesto a la espera de ser enviado y no es enviado hasta que el módulo de destino le realiza una petición de datos. Normalmente se usa para asegurar la recepción de paquetes por los dispositivos finales que pasan parte de su tiempo en modo suspendido.

Los módulos Xbee siguen un procedimiento por el cual antes de cada envío de paquetes, se realiza un escaneo de la energía del canal para ver el nivel de ocupación del canal, para que si este está ocupado esperarse a que quede libre antes de intentar enviar el paquete. Esta comprobación del canal es llamada CCA (*Clear Channel Asesessment*), y consiste en detectar la energía del canal, por el que se va a enviar el paquete y comparar su energía con el valor del parámetro CA en el XBee. Si el valor de la energía del canal es inferior a este, se retrasa el envío hasta que se descongestione el canal y su nivel de energía baje por debajo del parámetro CA.

También se puede pedir al receptor el envío de un chequeo o reconocimiento (*acknowledgement).* Esto consiste en que si el módulo que envía el paquete de datos no recibe una señal de vuelta (denominada ack), este enviará la información 3 veces seguidas hasta recibir dicha señal. Si en ninguna de las 3 ocasiones se recibe el ack, el módulo emisor avisará de un error de ack, lo que nos daría a entender que el módulo receptor no habría recibido los datos.

- Modo comandos**:**

Modo comando, utilizado para la configuración del módulo. En este modo se pueden leer y escribir los parámetros de configuración del módulo. En este estado los caracteres recibidos son interpretados como comandos. Los módulos soportan dos tipos de modo comando diferente, el modo AT y el modo API.

- Modo AT:

En este modo es posible configurar el módulo XBee mediante un terminal conectado por el puerto serie a la UART del XBee, enviando caracteres ASCII.

- Modo API:

Sirve para leer o escribir comandos en el módulo, además de ser el medio en que se envían las lecturas y escrituras de los pines de entrada/salida del módulo. Este modo tiene la peculiaridad que permite la configuración remota del módulo, y es el utilizado para el uso de los módulos desde aplicaciones informáticas, dado que permite el envío y recepción de paquetes dirigidos a un destinatario concreto y diferente sin tener que cambiar el parámetros de destino en el módulo, simplemente añadiendo al paquete a enviar la dirección del destinatario (DL, DH – *Destination Address Low, High*).

Para poder meter los comandos al módulo Xbee, es necesario utilizar o el Hyperterminal de Windows, o el programa X-CTU (que proporciona el fabricante), o algún microcontrolador que maneje UART y tenga los comandos guardados en memoria o los adquiera de alguna otra forma.

3 PLACA CONTROLADORA ARDUINO

3.1 ¿Qué es Arduino?

Arduino es una plataforma de prototipos electrónica de código abierto (open – source) basada en hardware y software flexibles y fáciles de usar. Está pensado e inspirado en artistas, diseñadores, y estudiantes de computación o robótica y para cualquier interesado en crear objetos o entornos interactivo, o simplemente por hobby. Arduino consta de una placa principal de componentes eléctricos, donde se encuentran conectados los controladores principales que gestionan los demás complementos y circuitos ensamblados en la misma. Además, requiere de un lenguaje de programación para poder ser utilizado y, como su nombre lo dice, programado y configurarlo a nuestra necesidad, por lo que se puede decir que Arduino es una herramienta completa en cuanto a las herramientas principales nos referimos, ya que sólo debemos instalar y configurar con el lenguaje de programación de esta placa los componentes eléctricos que queramos para realizar el proyecto que tenemos en mente, haciéndola una herramienta no sólo de creación, sino también de aprendizaje en el ámbito del diseño de sistemas electrónicos-automáticos y, además, fácil de utilizar. Arduino también simplifica el proceso de trabajo con micro controladores, ya que está fabricada de tal manera que viene pre-ensamblada y lista con los controladores necesarios para poder operar con ella una vez que la saquemos de su caja, ofreciendo una ventaja muy grande para profesores, estudiantes y aficionados interesados en el desarrollo de tecnologías. Las posibilidades de realizar proyectos basados en esta plataforma tienen como limite la imaginación de quien opera esta herramienta.

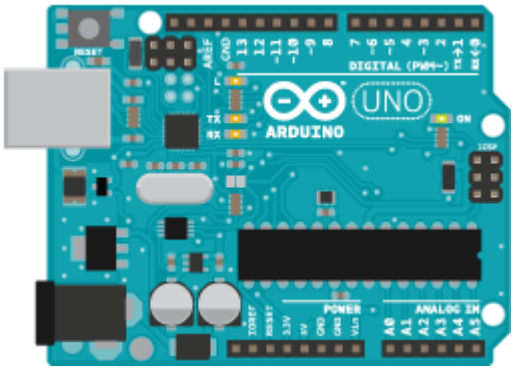


Figura 4. Arduino (Modelo UNO)

3.2 Historia

Arduino fue inventado en el año 2005 por el entonces estudiante del instituto IVRAE Massimo Banzi, quien, en un principio, pensaba en hacer Arduino por una necesidad de aprendizaje para los estudiantes de computación y electrónica del mismo instituto, ya que en ese entonces, adquirir una placa de micro controladores eran bastante caro y no ofrecían el soporte adecuado; no obstante, nunca se imaginó que esta herramienta se llegaría a convertir en años más adelante en el líder mundial de tecnologías DIY (Do It Yourself). Inicialmente fue un proyecto creado no solo para economizar la creación de proyectos escolares dentro del instituto, sino que además, Banzi tenía la intención de ayudar a su escuela a evitar la quiebra de la misma con las ganancias que produciría vendiendo sus placas dentro del campus a un precio accesible (1 euro por unidad).

El primer prototipo de Arduino fue fabricado en el instituto IVRAE. Inicialmente estaba basado en una simple placa de circuitos eléctricos, donde estaban conectados un micro controlador simple junto con resistencias de voltaje, además de que únicamente podían conectarse sensores simples como leds u otras resistencias, y es más, aún no contaba con el soporte de algún lenguaje de programación para manipularla.
Años más tarde, se integró al equipo de Arduino Hernando Barragán, un estudiante de la Universidad de Colombia que se encontraba haciendo su tesis, y tras enterarse de este proyecto, contribuyó al desarrollo de un entorno para la programación del procesador de esta placa: Wiring, en colaboración con David Mellis, otro integrante del mismo instituto que Banzi, quien más adelante, mejoraría la interfaz de software.
Tiempo después, se integró al "Team Arduino" el estudiante español David Cuartielles, experto en circuitos y computadoras, quien ayudó Banzi a mejorar la interfaz de hardware de esta placa, agregando los micro controladores necesarios para brindar soporte y memoria al lenguaje de programación para manipular esta plataforma.
Más tarde, Tom Igoe, un estudiante de Estados Unidos que se encontraba haciendo su tesis, escuchó que se estaba trabajando en una plataforma de open-source basada en una placa de microcontroladores pre ensamblada. Después se interesó en el proyecto y fue a visitar las instalaciones del Instituto IVRAE para averiguar en que estaban trabajando. Tras regresar a su país natal, recibió un e-mail donde el mismo Massimo Banzi invitó a Igoe a participar con su equipo para ayudar a mejorar Arduino. Aceptó la invitación y ayudó a mejorar la placa haciéndola más potente, agregando puertos USB para poder conectarla a un ordenador. Además, el le sugirió a Banzi la distribución de este proyecto a nivel mundial.
Cuando creyeron que la placa estaba al fin lista, comenzaron su distribución de manera gratuita dentro de las facultades de electrónica, computación y

diseño del mismo instituto. Para poder promocionar el proyecto Arduino dentro del campus, tuvieron que consultar con un publicista que más parte pasaría a formar parte del equipo Arduino: Gianluca Martino, quien la distribuyo dentro del instituto y promocionándola a algunos conocidos y amigos suyos. Al ver su gran aceptación por parte de los alumnos y maestros y tomando en cuenta el consejo de Igoe, pensaron en su distribución nivel mundial, para lo cual contactaron a un amigo y socio de Banzi, Natan Sadle, quien se ofreció a producir en masa las placas tras interesarse en el proyecto.

Un breve tiempo más tarde, al ver los grandes resultados que tuvo Arduino y las grandes aceptaciones que tuvo por parte del público, comenzó a distribuirse en Italia, después en España, hasta colocarse en el número uno de herramientas de aprendizaje para el desarrollo de sistemas autómatas.

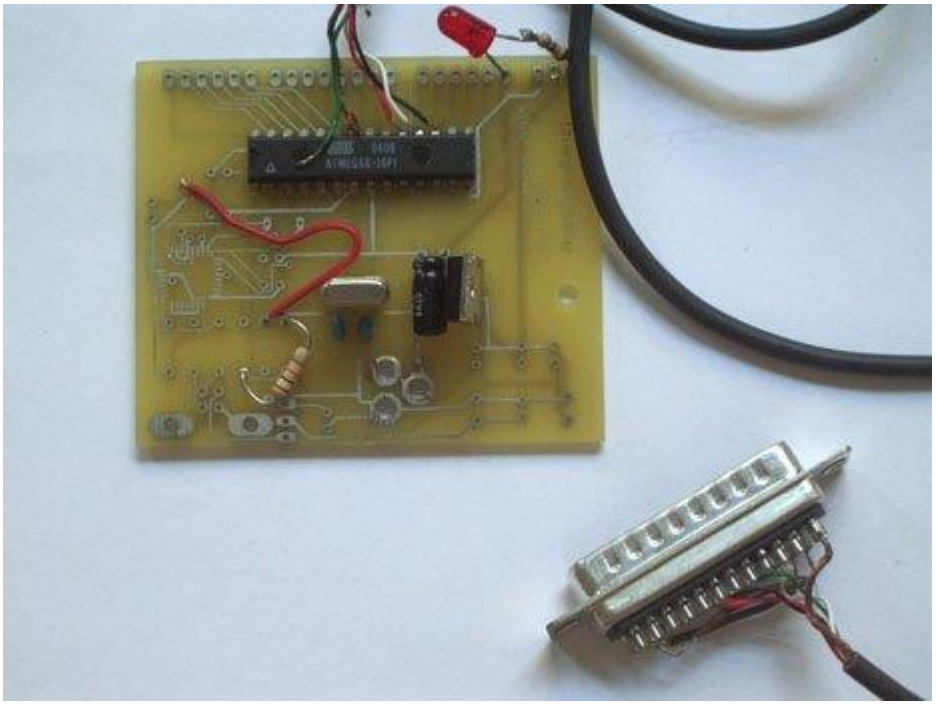


Figura 5. Primer prototipo de Arduino

3.3 Hardware

Consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, y Atmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software

consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa. Se programa en el ordenador para que la placa controle los componentes electrónicos.

Desde octubre de 2012, Arduino se utiliza también con microcontroladoras CortexM3 de ARM de 32 bits, que coexistirán con las más limitadas, pero también económicas AVR de 8 bits. ARM y AVR no son plataformas compatibles a nivel binario, pero se pueden programar con el mismo IDE de Arduino y hacerse programas que compilen sin cambios en las dos plataformas. Eso sí, las microcontroladoras CortexM3 usan 3,3V, a diferencia de la mayoría de las placas con AVR, que generalmente usan 5V. Sin embargo, ya anteriormente se lanzaron placas Arduino con Atmel AVR a 3,3V como la Arduino Fio y existen compatibles de Arduino Nano y Pro como Meduino en que se puede conmutar el voltaje.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software tal como Adobe Flash, Processing, Max/MSP, Pure Data. Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente.

Arduino puede tomar información del entorno a través de sus entradas analógicas y digitales, puede controlar luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador.

También cuenta con su propio software que se puede descargar de su página oficial que ya incluye los drivers de todas las tarjetas disponibles lo que hace más fácil la carga de códigos desde el computador.

3.4 Características generales

Lo principal que debemos saber es que tipo de proyectos vamos a implementar. Con esto nos da una idea de la cantidad de pines analógicos y digitales (normales y de tipo PWM o modulados por ancho de pulso para simular una salida analógica) que necesitamos para nuestro trabajo. Este primer escrutinio nos permite descartar algunas placas más simples que no tengan suficientes pines o, al contrario, descartar las de mayor número de ellos para reducir los costes puesto que con menos pines nos conformamos.

También podemos deducir el tamaño de código que vamos a generar para nuestros sketchs. Un programa muy largo, con muchas constantes y variables demandará una cantidad mayor de memoria flash para su

almacenamiento, por lo que se debe elegir una placa adecuada para no quedarnos cortos.

La RAM será la encargada de cargar los datos para su inmediato procesamiento, pero no es uno de los mayores escollos, puesto que esto solo afectaría a la velocidad de procesamiento. La RAM va ligada al microcontrolador, puesto que ambos afectan a la agilidad de procesamiento de Arduino.

En los Arduinos originales podemos diferenciar entre dos tipos fundamentales de microcontroladores, los de 8 y 32 bits basados en ATmega AVR y los SMART basados en ARM de 32 bits y con un rendimiento superior, ambos creados por la compañía Atmel.

Por último, en cuanto al voltaje, no importan demasiado a nivel electrónico, excepto en algunos casos, para tener en cuenta la cantidad de tensión que la placa puede manejar para montar nuestros circuitos. Esto no supone mayor problema, puesto que una placa de Arduino podría trabajar incluso con tensiones de 220v en alterna con el uso por ejemplo de relés. Pero cuando queremos prescindir de una fuente de alimentación externa, hay que tener en cuenta que este es el voltaje que se puede manejar. Y entre otras cosas marcar el límite para no destruir la placa con sobretensiones no soportadas.

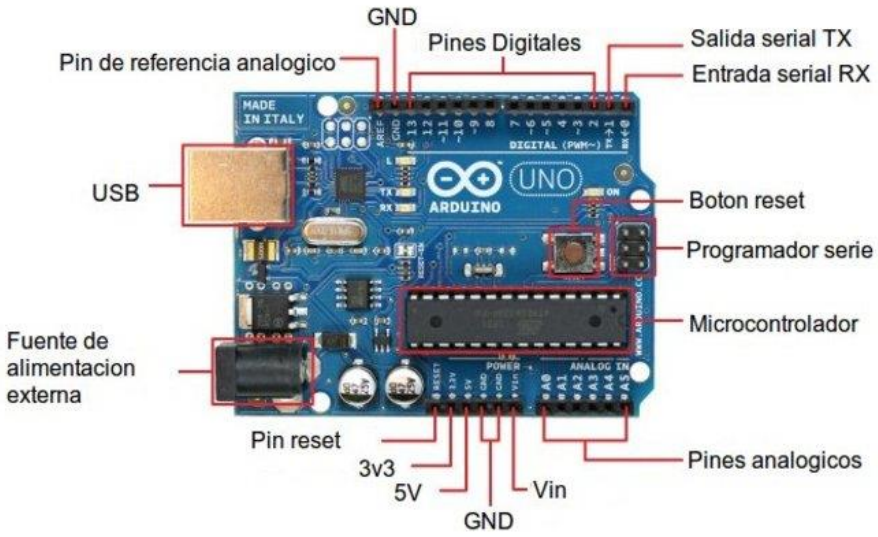


Figura 6.Descripción física de Arduino

3.5 Tipos y características

De entre las placas puedes encontrar multitud de modelos. Todos especialmente pensados para un fin, compatibles con los shields y módulos oficiales, así como con Arduino IDE (entorno de programación de Arduino):

- ***Arduino Uno:***

Es la plataforma más extendida y la primera que salió al mercado, por ello nos podemos basar en esta para hacer la comparativa con el resto de placas. Todas las características de esta placa estarán implementadas en casi todas las placas restantes, a excepción de algunas que ya veremos. Se basa en un microcontrolador Atmel ATmega320 de 8 bits a 16Mhz que funciona a 5v. 32KB son correspondientes a la memoria flash (0,5KB reservados para el bootloader), 2KB de SRAM y 1KB de EEPROM. En cuanto a memoria es una de las placas más limitadas, pero no por ello resulta insuficiente para casi todos los proyectos que rondan la red. Las salidas pueden trabajar a voltajes superiores, de entre 6 y 20v pero se recomienda una tensión de trabajo de entre 7 y 12v. Contiene 14 pines digitales, 6 de ellos se pueden emplear como PWM. En cuanto a pines analógicos se cuenta con hasta 6. Estos pines pueden trabajar con intensidades de corriente de hasta 40mA.

Figura 7.Arduino UNO

- ***Arduino Zero:***

En aspecto es similar a Arduino UNO,pero con alguna que otra diferencia. En vez del microcontrolador Atmel ATmega basado en arquitectura AVR de 8 bits, el Zero contiene un potente Atmel SAMD21 MCU de 48Mhz con un núcleo ARM Cortex M0 de 32 bits. Con 256 KB de memoria flash, 32 KB de SRAM y una EEPROM de más de 16KB por emulación. El voltaje en el que opera es de 3,3/5v (7mA) y contiene 14 pines E/S digitales, de los cuales 12 son PWM y UART. En el terreno analógico se dispone de 6 entradas para un canal ADC de 12 bits y una salida analógica para DAC de 10 bits. En definitiva, esta placa va destinada para los que Arduino UNO se les quede

corto y necesitan algo más de potencia de procesamiento.

Figura 8.Arduino ZERO

- ***Arduino Yun:***

se basa en el microcontrolador ATmega32u4 y en un chip Atheros AR9331 (que controla el host USB, el puerto para micro-SD y la red Ethernet/WiFi), ambos comunicados mediante un puente. El procesador Atheros soporta la distribución Linux basadas en OpenWrt llamada OpenWrt-Yun. Se trata de una placa similar a Arduino UNO pero con capacidades nativas para conexión Ethernet, WiFi, USB y micro-SD sin necesidad de agregar o comprar shields aparte. Contiene 20 pines digitales, 7 pueden ser usados en modo PWM y 12 como analógicos. El microcontrolador ATmega32u4 de 16Mhz trabaja a 5v y contiene una memoria de solo 32KB (4KB reservados al bootloader), SRAM de solo 2,5KB y 1KB de EEPROM. Como vemos, en este sentido queda corto. Sin embargo se complementa con el AR9331 a 400Mhz basado en MIPS y trabajando a 3,3V. Este chip además contiene RAM DDR2 de 64MB y 16MB flash para un sistema Linux embebido.

Figura 9.Arduino YUN

- ***Arduino Leonardo:***

Es una placa basada en un microcontrolador ATmega32u4 de bajo consumo y que trabaja a 16Mhz. La memoria flash tiene una capacidad de 32KB (4KB para el bootloader) y 2.5KB de SRAM. La EEPROM es de 1KB, también muy similar a Arduino UNO en cuanto a capacidades de almacenamiento. A nivel electrónico y de voltajes es igual al UNO. Pero este microcontrolador puede manejar 20 pines digitales (7 de ellos pueden ser manejados como PWM) y 12 pines analógicos. Como vemos, contiene los mismos pines que Yun, solo que prescinde de las funcionalidades de red. El volumen ocupado por Leonardo es inferior al de UNO, puesto que carece de las inserciones de los pines y en su lugar posee perforaciones con pads de conexión en la propia placa. Además las dimensiones del conector USB de la placa es mucho menor, ya que en vez de una conexión USB emplea una mini-USB para ahorrar espacio. Por eso es idóneo para proyectos en los que se requiera ahorrar algo de espacio, pero todo al mismo precio que UNO.

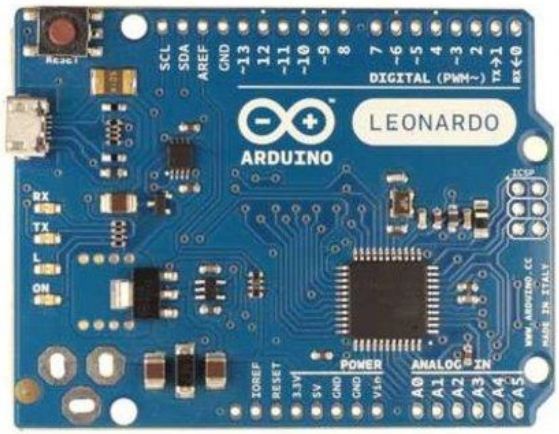


Figura 10.Arduino Leonardo

- ***Arduino Due:***

Es una placa con un microcontorlador Atmel SAM3X8E ARM Cortex-M3 de 32 bits. Este chip que trabaja a 84Mhz (3,3V) aporta una potencia de cálculo bastante superior a los anteriores microcontroladores vistos. Por eso es idóneo para todos aquellos que necesiten de un proyecto con alta capacidad de procesamiento. Al tener un núcleo a 32 bits permite realizar operaciones con datos de 4 bytes en un solo ciclo de reloj. Además, la memoria SRAM es de 96KB, superior al resto de placas vistas anteriormente e incorpora un controlador DMA para acceso directo a memoria que intensifica el acceso a memoria que puede hacer la CPU. Para el almacenamiento se dispone de 512KB de flash, una cantidad muy grande de memoria para cualquier código de programación. En cuanto a soporte de voltajes en intensidades es idéntica a UNO, solo que el amperaje de los pines se extiende hasta los 130-800mA (para 3,3V y 5vVrespectivamente). El sistema dispone de 54 pines de E/S digitales, 12 de ellos pueden ser usados como PWM. También tiene 12 analógicos, 4 UARTs (serie, frente a los dos de UNO), capacidades de conexión USB OTG, dos conexiones DAC (conversión digital a analógico), 2 TWI, un power jack, SPI y JTAG. Como vemos en cuanto a interfaz de conexionado está muy completo y permite multitud de posibilidades.

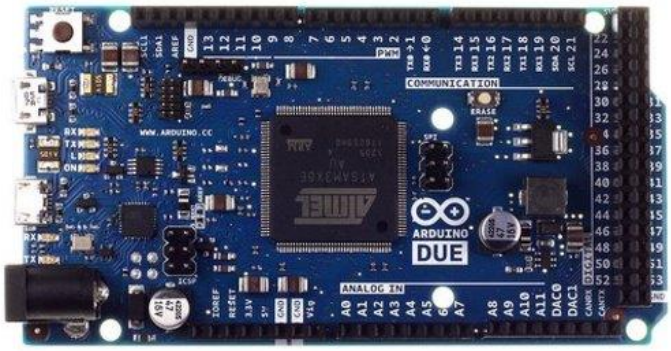


Figura 11.Arduino Due

- ***Arduino Mega:***

Su nombre proviene del microcontrolador que lo maneja, un ATmega2560. Este chip trabaja a 16Mhz y con un voltaje de 5V. Sus capacidades son superiores al ATmega320 del Arduino UNO, aunque no tan superiores como las soluciones basadas en ARM. Este microcontrolador de 8 bits trabaja conjuntamente con una SRAM de 8KB, 4KB de EEPROM y 256KB de flash (8KB para el bootloader). Como puedes apreciar, las facultades de esta placa se asemejan al Due, pero basadas en arquitectura AVR en vez de ARM. En cuanto a características electrónicas es bastante similar a los anteriores, sobre todo al UNO. Pero como se puede apreciar a simple vista, el número de pines es parecido al Arduino Due: 54 pines digitales (15 de ellos PWM) y 16 pines analógicos. Esta placa es idónea para quien necesita más pines y potencia de la que aporta UNO, pero el rendimiento necesario no hace necesario acudir a los ARM-based.

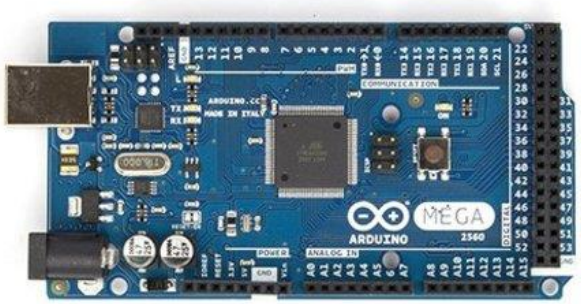


Figura 12.Arduino Mega

- ***Arduino Ethernet:***

Si se busca un Arduino UNO pero con capacidades Ethernet se tienen dos opciones, comprar un Arduino UNO y un shield Ethernet para integrarlo, o la otra opción sería adquirir un Arduino Ethernet. Esta placa es bastante similar a la UNO, incluso en el aspecto, pero tiene capacidades de red. Su microcontrolador es un ATmega328 que trabaja a 16Mhz (5V). Va acompañado de 2KB de SRAM, 1KB de EEPROM y 32KB de flash. El resto de características electrónicas son como las de UNO solo que añade capacidad para conexión Ethernet gracias a un controlador W5100 TCP/IP embebido y posibilidad de conectar tarjetas de memoria microSD. Los pines disponibles son 14 digitales (4 PWM) y 6 analógicos. Lo que hay que tener en cuenta es que Arduino reserva los pines 10-13 para ser usado para SPI, el 4 para la tarjeta SD y el 2 para el interruptor W5100.

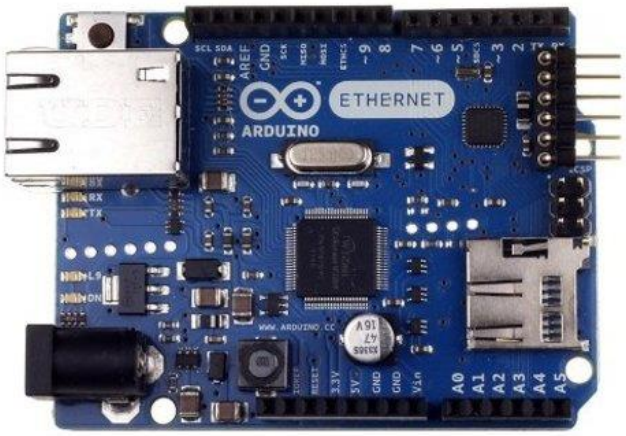


Figura 13.Arduino Ethernet

- ***Arduino Fio:***

Es una placa Arduino reducida a la mínima expresión. Por su tamaño es especialmente considerado para proyectos móviles inalámbricos o para ser insertados en espacios reducidos. Funciona con un microcontrolador ATmega328P, una versión similar a la del Ethernet pero que trabaja a una frecuencia inferior, 8Mhz. Al ser tan reducida carece de ciertas comodidades, por ejemplo, para subir los sketches hay que usar un cable FTDI o una placa adicional adaptadora Sparkfun. Igualmente, las tensiones

con las que se trabaja se ven mermadas hasta los 3.35-12V máximo. 14 pines digitales (6 PWM) y 8 pines analógicos serán los únicos disponibles en esta placa. Tampoco ayuda sus 2KB de SRAM, 32KB de flash y 1KB de EEPROM, todo esto limitará mucho el tamaño de los sketchs y del circuito del proyecto.

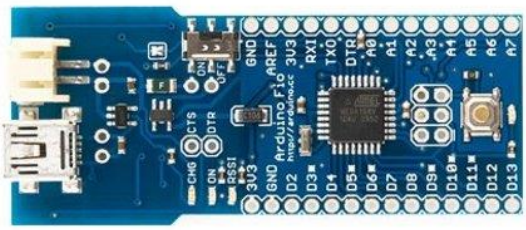


Figura 14.Arduino Fio

- ***Arduino Nano:***

Empezaron incorporando un ATMega328 como el de otras placas vistas anteriormente, pero tras la revisión 2.x se sustituyó por un ATmega168 a 16Mhz. Sus dimensiones son aún más reducidas que las de Fio, de tan solo 18,5×43.2mm. Su reducido tamaño no le quitan la posibilidad de ser una placa completa, pero sí que necesita de un cable mini-USB y no posee conector de alimentación externa. Esta versión fue diseñada y producida por la compañía Gravitech, especialmente pensado para aplicaciones de reducido costo y donde el tamaño importe. A nivel eléctrico se comporta como un UNO, con 14 pines digitales (6 PWM) y 8 analógicos. Pero sus capacidades han ido a menos con las nuevas revisiones en pos de un menor consumo. Por ello se ha pasado de 32 a 16KB de flash (2 reservados al cargador de arranque), de 2 a 1KB de SRAM y de 1KB a 512 bytes de EEPROM.

Figura 15.Arduino Nano

- ***Arduino LilyPad:***

Especial para ser integrado en prendas y textiles, es decir, es una versión de Arduino “ponible”. Fue desarrollado por Leah Buechley y SparkFun Electronics para ser empleado con los mismo fines que otros Arduinos solo que con ciertas limitaciones a cambio de sus capacidades de integración y su base flexible. Se basa en dos versiones de microcontrolador diferentes, ambas de bajo consumo denominadas Atmega168V y ATmega328V, esta segunda más potente. Ambos trabajan a 8Mhz, pero la primera trabaja a solo 2,7V y a segunda a 5,5V. Dispone de 14 pines digitales (6 PWM) y 6 analógicos a lo largo de su perímetro. Además integra 16KB de memoria flash para el código del programa, 1KB de SRAM y 512 bytes de EEPROM.

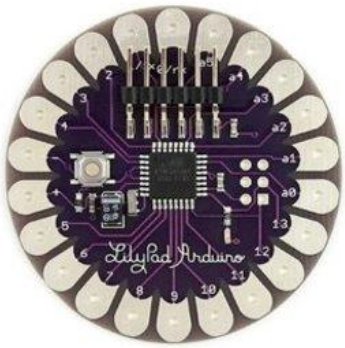


Figura 16.Arduino LilyPad

- ***Arduino Pro:***

Contiene un microcontrolador ATmega168 o Atmega328, con versiones de 3.3V y 8Mhz o 5V para 16Mhz. Contiene 14 pines de E/S digitales (6 de ellos son PWM) y 6 pines analógicos. Esta placa diseñada y construida por SparkFUn Electronics integra entre 32KB y 16KB de flash según el microcontrolador en el que se base (2KB reservados para el gestor de arranque). Lo que sí es común para todos los modelos es el KB de SRAM y los 512 bytes de EEPROM. A pesar de su nombre, la versión Pro no es una de las más potentes como podemos apreciar. Pero se ha concebido para usuarios avanzados que necesitan flexibilidad y precios bajos.

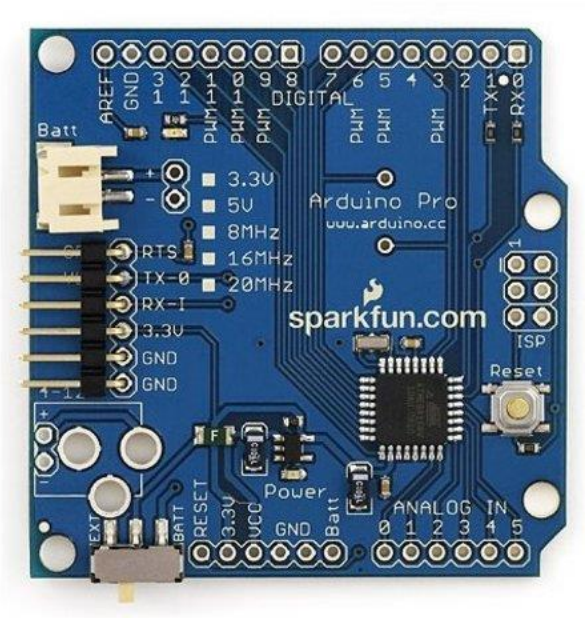


Figura 17.Arduino Pro

- ***Arduino Pro Mini:***

Es la hermana pequeña de la versión Pro. Además de la flexibilidad y precios bajos hay que añadirle su reducido tamaño. Para reducir coste y tamaño dispone de los componentes mínimos (sin conector USB Integrado ni conectores de pin). El resto de características son muy similares a la versión Pro.

Figura 18.Arduino Pro Mini

- ***Arduino Mega ADK:***

Placa basada en un ATmega2560 como la versión Mega vista anteriormente. Pero su principal ventaja es que dispone de una interfaz preparada para ser conectada mediante USB a dispositivos móviles basados en Android, gracias a su IC MAX3421e. Esto permite contar con todas las capacidades de una placa Arduino Mega (igual en cuanto al resto de

características) más las posibilidades de desarrollo conjuntas con una plataforma Android.

Figura 19.Arduino Mega ADK

- ***Arduino Esplora:***

Se diferencia del resto de placas, a parte de su reducido tamaño y por su forma, en que dispone de una serie de sensores onboard. Es bueno para las personas que comienzan a dar sus primeros pasos en el mundo de la electrónica y están aprendiendo, de ahí su nombre. Incluye unos sensores (acelerómetro, temperatura, luz), zumbador, botones, joystick, micrófono y un socket para conectar una pantalla a color TFT LCD. Funciona con un microcontorlador ATmega32u4 que trabaja a 16MHz y 5V, con SRAM de 2,5KB y 1KB de EEPROM. La flash llega a los 32KB con 4KB reservados al bootloader. El mayor problema es su capacidad de conectividad, casi nula, ya que todo lo tiene integrado para los principiantes.

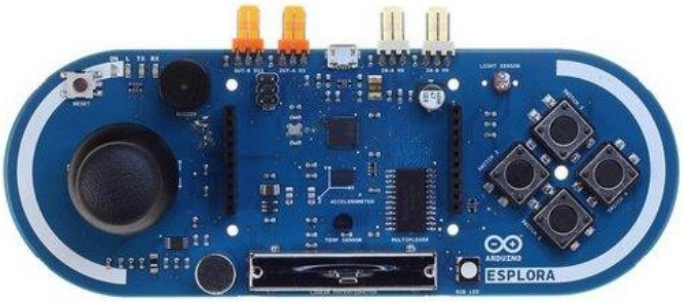


Figura 20.Arduino Esplora

- ***Arduino Micro:***

Diseñado por Adafruit y pensado para una autonomía elevada y con un reducido tamaño. Su precio es bajo con respecto a otros modelos. Sin embargo cuenta con características similares a otros diseños, como un microcontrolador ATmega32u4 a 16Mhz, 20 pines digitales (7 de ellos PWM) y 12 analógicos. En muchos aspectos es similar a Leonardo, pero con capacidad de comunicación USB built-in, eliminando la necesidad de un segundo procesador.

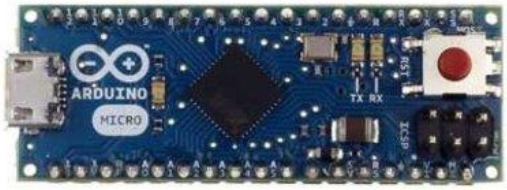


Figura 21.Arduino Micro

- ***Arduino BT:***

Es una placa Arduino con un módulo Bluetooth incorporado y que permite comunicación inalámbrica sin necesidad de comprar un shield independiente. El módulo bluetooth incorporado es un Bluegiga WT11. El resto de características son similares al Arduino UNO, con microcontrolador ATmega168 o 328 a 16MHz.

Figura 22.Arduino BT

- ***Arduino Duemilanove:***

Apareció en 2009, con microcontrolador ATmega168 o 368, 14 pines digitales (6 con posibilidad PWM) y 6 analógicos. Contiene conexión USB, entrada de alimentación, una cabecera ISCP, botón de reset… todo bastante parecido a UNO ya que se trata de una versión previa que fue sustituida por UNO en la actualidad.

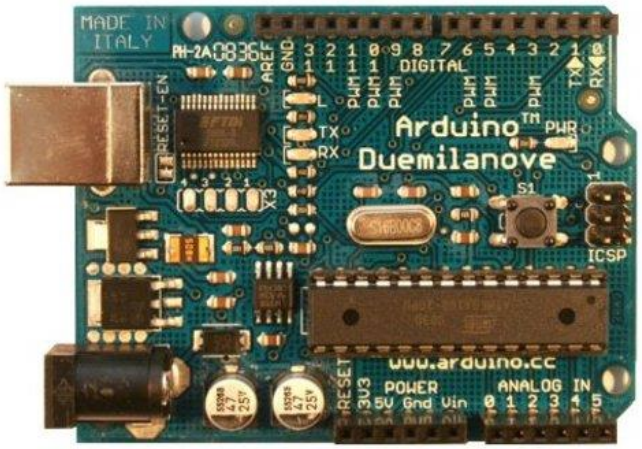


Figura 23.Arduino Duemilanove

- ***Arduino Diecimila:***

Otra placa antigua basada en chip DIP ATmega168 a 16Mhz, con 14 pines digitales (6 PWM) y 6 analógicos. El resto de características son similares al más moderno UNO que se ha sacado al mercado para sustituir a estos diseños más primitivos. El modelo “Diecimila” debe su nombre a “diez mil” en italiano, modo en el que se festejó el hecho de haber vendido más de 10.000 placas Arduino en la época en la que se sacó a la venta. Fue una de las placas más antiguas de Arduino, sacada en 2007.

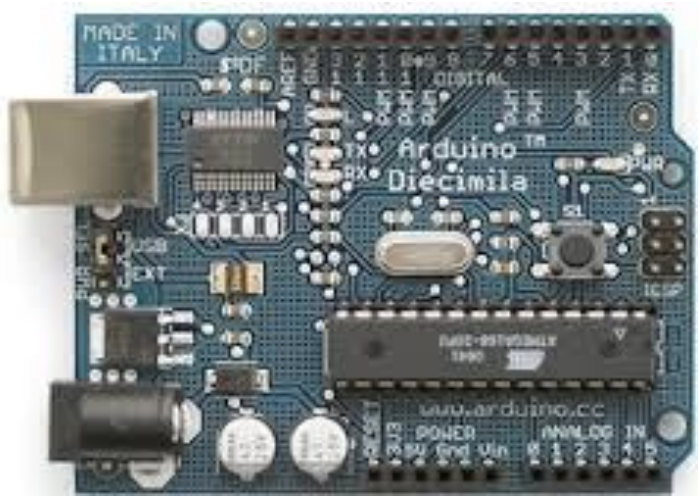


Figura 24.Arduino Diecimila

Existen placas compatibles con Arduino pero no pueden estar registradas bajo el nombre de Arduino. Son diseñadas y fabricadas por otras compañías ajenas. Estos desarrollos no aportan nada al desarrollo propio de Arduino, sino que son derivados que han salido para cubrir otras necesidades. Estas frecuentemente utilizan un nombre que integra el sufijo “duino” para identificarlas.

3.6 Estructura de programación

Todo programa para Arduino presenta una estructura básica:

1ª parte **int** x=0; Declarar las variables.
2ª parte **void setup() {…}** Configuración de Arduino.
3ª parte **void loop() {…}** Comandos que regirán el comportamiento de Arduino.

A. Declarar las variables

int x=0;

Una variable es un valor que Arduino puede almacenar en su memoria, y que posteriormente podrá ser utilizado o modificado.

Los tipos de variables más utilizados son:

- **int:** almacena un número entero entre -32769 y 32767 (2 bytes).
- **long:** almacena un número entero muy largo, entre -2147483648 y 2147483647 (4 bytes).

- **float:** almacena un número decimal con un rango entre $-3.4028235 \cdot 10^{38}$ y $3.4028235 \cdot 10^{38}$ (4 bytes).
- **const:** especifica que la variable definida no podrá ser cambiada durante el programa, siendo un siempre un valor constante:
  const float pi=3.1415;

Es importante saber que es posible declarar una variable sin asignarle un valor inicial, y hacerlo posteriormente durante el transcurso del programa:

```
int x;
…
x=4;
```

Dominio de una variable: si declaro una variable al comienzo del programa, podré emplear dicha variable en cualquier momento (dentro de cualquier función o bloque de programa), pero si declaro una variable dentro de una función, solo se podrá utilizar en dicha función.

Poner nombre a las variables: Por último, una última consideración: a la hora de poner un nombre a una variable es recomendable utilizar alguna palabra que nos ayude a reconocer que se está almacenando en ella, y en caso de utilizar dos o más palabras se suele emplear la notación de joroba de camello (poner en mayúscula la primera letra de las siguientes palabras).
Por ejemplo:

ledPin	estadoAnterior	cuentaPulsaciones
miVariable	lecturaSensor	ledPinAzul

B. Configuración de Arduino

void setup() {…}

En este bloque habrá que especificar:

Que pines van a ser empleados como entrada y cuales como salida.

```
pinMode(2,OUTPUT);      //utilizaré el pin 2 como salida Digital.
pinMode(3,OUTPUT);      //utilizaré el pin 3 como salida Digital
                        o Analógica.
pinMode(8,INPUT);       //utilizaré el pin 8 como entrada
                        Digital.
```

Las entradas analógicas no hacen falta incluirlas en el setup, puesto que esos pines (A0, A1, A2, A3, A4, A5) sólo pueden ser entradas analógicas.

Si vamos a querer establecer una conexión con el ordenador:

Serial.begin(9600); /*hay que especificar los baudios (bits por segundo) a la que va a realizarse dicha comunicación Arduino-PC */

Si vamos a querer utilizar número aleatorios:

randomSeed(0); //se inicia la generación de un número aleatorios.

C. Comandos que regirán el comportamiento de Arduino

void loop () {…}

En este bloque se deberá escribir todas aquellas instrucciones, órdenes, primitivas, comandos o funciones necesarias para que Arduino funcione según nuestro deseo (for, while ,case , if…).

Realmente, este bloque constituye un bucle infinito, ya que Arduino, mientras este alimentada con energía, funcionará haciendo el programa loop una y otra vez.

D. Entradas y salidas

- Entradas digitales:
-

Debemos indicar a Arduino que pin vamos a emplear como entrada digital. Dicha señal de entrada podrá tener dos valores: LOW o HIGH.

x = **digitalRead**(3); //asigna a x el valor logico que Arduino lee en el pin 3.

Cualquier sensor que tenga dos posibles valores distintos puede emplearse como entrada digital.

- Entradas analógicas:

Las entradas analógicas en Arduino no hay que configurarlas en el setup. Dicha señal analógica de entrada podrá tener valores comprendidos entre 0 y 1023, correspondientes a los valores intermedios de un rango de 0 a 5 V.

x = **analogRead**(A3); //asigna a x el valor analógico que Arduino lee en el pin A3.

Como los únicos pines válidos para una entrada analógica son los A0...A5, no hay posibilidad de confusión, y puede omitirse la A:

x = **analogRead**(3);

Evidentemente, para emplear una entrada analógica necesitamos un sensor analógico, es decir, que sus valores eléctricos varíen en un rango significativo, no limitándose a dos posibles valores. Sensores analógicos pueden ser: LDR (fotorresistencia), NTC (termorresistencia), potenciómetro, sensor de sonido (piezorresistencia), sensor de ultrasonido, etc…

- Salidas digitales:

Podemos indicar a Arduino que en un pin determinado coloque un "0" o un "1" lógico (que serán 0V ó 5V) mediante los siguientes comandos, respectivamente:

digitalWrite(12,LOW); **digitalWrite**(12,HIGH);

- Salidas analógicas:

Podemos indicar a Arduino que en un pin determinado coloque un valor de tensión comprendido entre 0V y 5V, pudiendo seleccionar entre 256 valores intermedios posibles (de 0 a 255), empleando la siguiente orden:

analogWrite(11,214); // coloca en el pin 11 un valor de salida de 214, que equivale a unos 4,2V.

Realmente, la señal analógica de salida no es analógica en sí, sino un PWM. Esto significa que la salida es una señal modulada por pulsos, formada por una serie de pulsos (de valor 5 V) repartidos durante un tiempo determinado de tal forma que el valor promedio de la señal de salida se hace coincidir con el de la señal analógica que se persigue imitar.

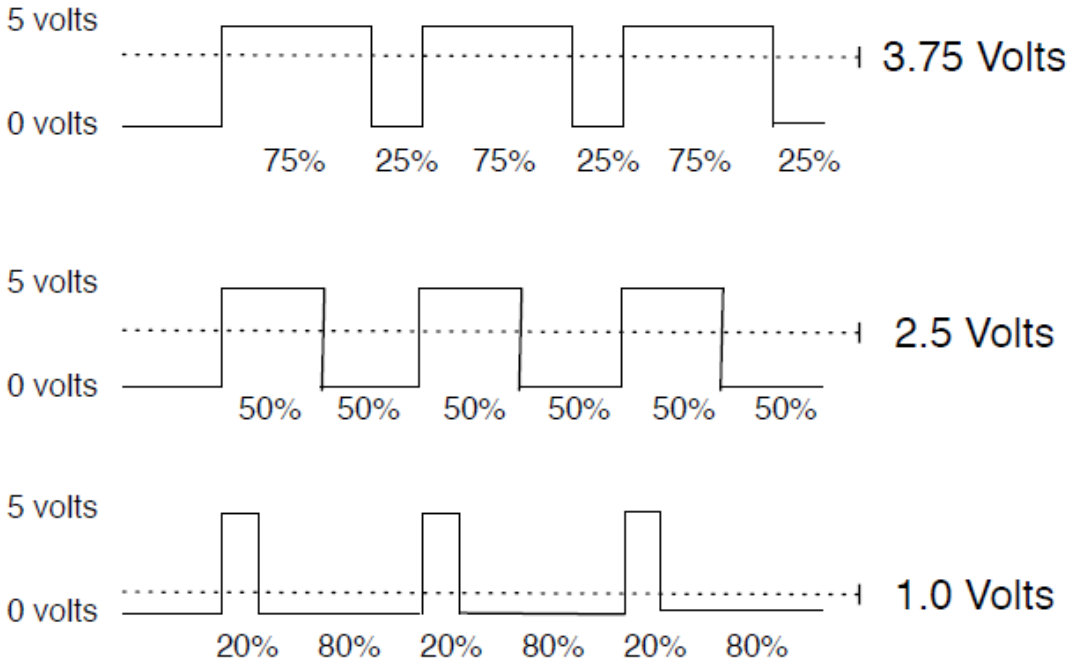


Figura 25.Señal PWM

4 DRIVER

4.1 ¿Qué es un driver?

Un driver es un dispositivo electrónico que regula el flujo de electricidad asegurando que un LED o una serie de LEDs se alimenta siempre con el voltaje y la corriente adecuada independiente de los cambios constantes en las propiedades eléctricas de los LEDs.

Los cambios de temperatura cambian la forma en que los semiconductores de los LEDs reaccionan a la electricidad. El driver ajusta el voltaje que suministra a los LEDs para alimentarlo a con una corriente constante.

El driver también se ajusta a las continuas fluctuaciones de voltaje de la red eléctrica y evita que los LEDs se vean perjudicados por un exceso de tensión.

Sin un driver adecuado los LEDs se pueden sobrecalentar o descontrolar provocando una pérdida de rendimiento o fallo irreparable.

4.2 Funcionamiento

El driver que se utilizará para este proyecto tiene el esquema de un inversor resonante como el siguiente:

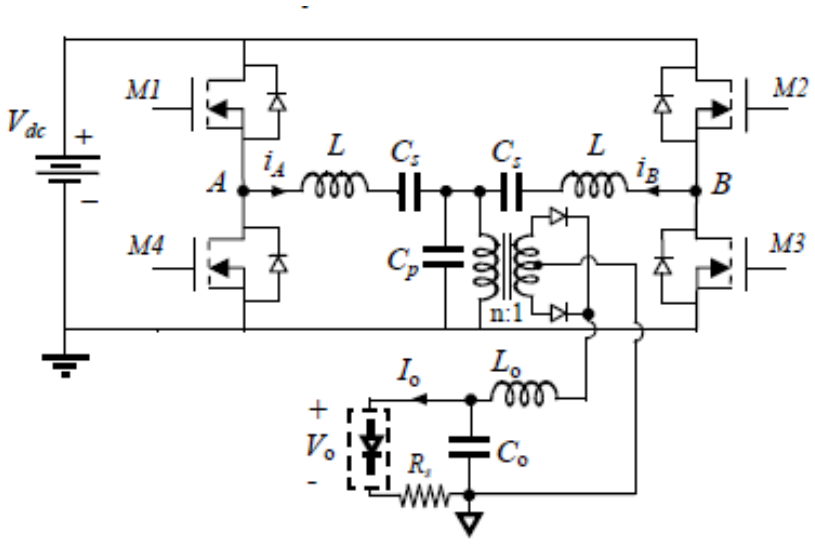


Figura 26.Circuito del driver

Como se aprecia en la ***Figura 26***, el circuito tiene una estructura simétrica donde se han establecido los puntos **A** y **B** para medir sus tensiones.

El objetivo es conseguir controlar la intensidad que sale del circuito para lograr tener poder sobre el flujo de iluminación de la lámpara LED a la que va conectado.

Esto se consigue teniendo control sobre el ángulo **ψ** que indica el desfase entre las tensiones en los dos puntos, A y B, del circuito según indica la siguiente fórmula.

$$\hat{I}_{ac} = \frac{4V_{dc}\sqrt{1+\frac{C_p}{2C_s}}}{\pi Z_p} \cdot Cos\left(\frac{\Psi}{2}\right)$$

De esta manera, controlando el ángulo **ψ** , la intensidad será máxima cuando los desfases entre las dos tensiones sea nulo, mientras que si existe un desfase de 180º entre las señales obtendremos una intensidad de salida 0,por lo que nuestra luminancia estaría apagada en ese momento.

Para conseguir los diferentes ángulos **ψ** que nos den control al circuito, buscaremos una señal PWM de entrada para lograr los desfases que nos interesan y de esta manera manejar la intensidad de la iluminación LED a nuestro gusto.

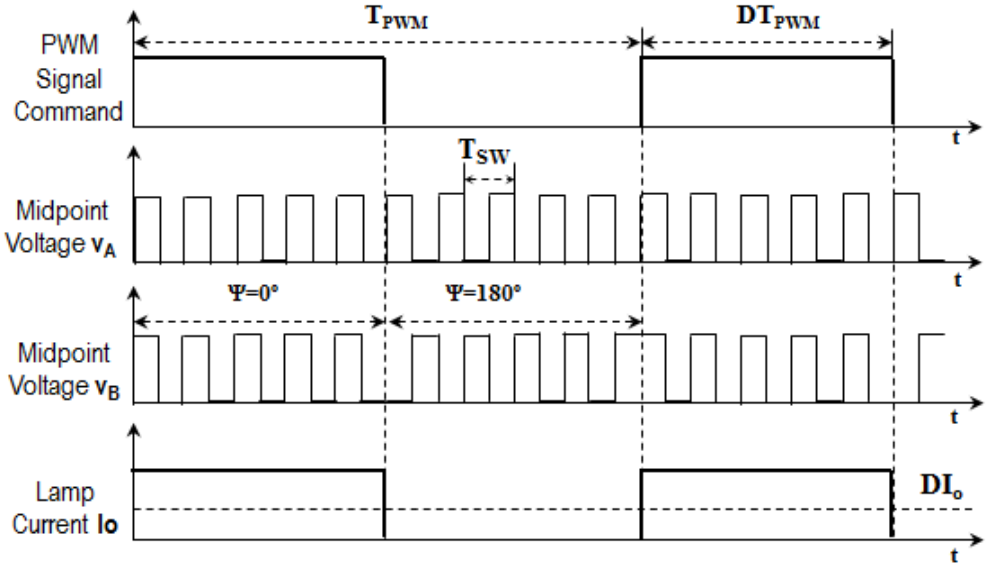


Figura 27.Desfase controlado por señal PWM

5. ILUMINACIÓN LED

5.1 Historia

En estos días, habiendo transcurrido más de 130 años de existencia (1879, Thomas Alva Edison), el empleo de las lámparas incandescentes empieza a desaparecer. En numerosos países desarrollados la fabricación, la importación o venta de las mismas será prohibida en un futuro muy próximo (en los 27 países de la Unión Europea ya lo fue en 2010, Australia en 2011, los Estados Unidos en 2012-14, en Japón en 2012, etc.), esto también ocurre en países en vías de desarrollo y en las nuevas economías emergentes.

La tecnología LED es relativamente antigua, va a cumplir una cincuentena de años, pero es de reciente aplicación para sistemas de iluminación general, desde que se crearon los LED de alta potencia o alto brillo, (High Brightness LED) que hicieron su aparición en 2001. Aún no se ha logrado la estandarización debido a que los sistemas actuales no tienen un rendimiento y cualidades mínimas estándar. Las expectativas se mantienen en conseguir un aumento en el rendimiento de unos 160lm/W a 200lm/W.

La tecnología y los sistemas de iluminación LED están rápidamente convirtiéndose en la iluminación del futuro, con un elevado ahorro de energía y una notable reducción de costos. Los LED (Light-emmiting diode ó diodos emisores de luz) son muy eficientes en el consumo de energía, compactos, de instalación fácil, de larga vida, bajo mantenimiento, y con extensas capacidades de diseño (en el color y en la atmósfera de iluminación).

5.2 Protocolo DALI

5.2.1 Introducción

DALI (Digital Addressable Lighting Interface) fue diseñado por los principales fabricantes de equipos de conexión electrónicos (ECE) con el fin de garantizar un estándar unificado en la industria de la iluminación. No hace referencia a un sistema concreto, sino a un estándar de comunicación entre un controlador y los distintos equipos de conexión electrónicos

Este sistema es un estándar internacional de acuerdo a la norma IEC 62386, que asegura la compatibilidad e intercambiabilidad entre equipos de diferentes fabricantes.

Figura 28.Logo de regulación DALI

5.2.2 Funcionamiento

Es un interfaz de regulación bidireccional con una estructura maestro-esclavo, donde la información fluye desde un controlador, que opera como maestro, hacia los equipos de iluminación que operan únicamente como esclavos, ejecutando los comandos o respondiendo a las solicitudes de información recibidas

La comunicación mediante las señales digitales se realiza a través de un bus o línea de control de dos hilos. Estos hilos de control pueden poseer polaridad positiva y negativa, aunque la mayoría de equipos están diseñados libres de polaridad para que la conexión sea indiferente.

No se necesitan cables especiales apantallados, pudiendo realizarse el cableado conjunto de la línea de alimentación y del bus DALI con una misma manguera estándar de 5 hilos.

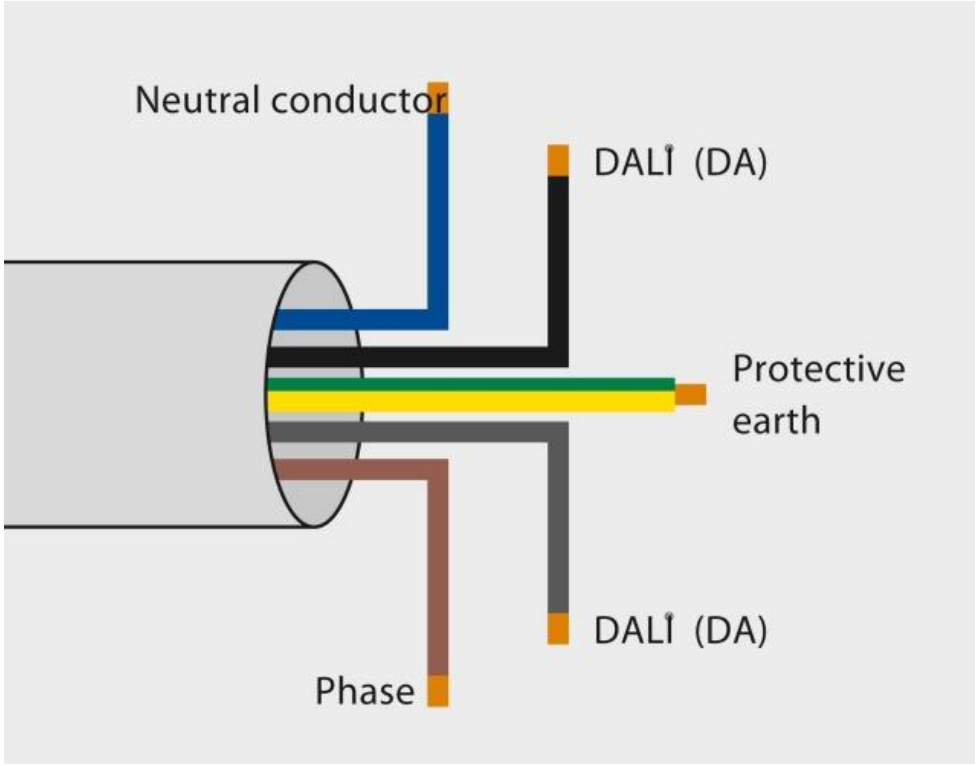


Figura 29.Diagrama simple de cableado

La máxima caída de tensión a lo largo de la línea de control no puede ser superior a 2V con la corriente máxima del bus de 250mA. Por tanto, la máxima distancia de cableado permitida depende de la sección del cable, pero en ningún caso debe ser superior a 300m.

Una vez realizado el cableado, se realiza la configuración del sistema de iluminación DALI vía software. Se pueden crear hasta 16 escenas diferentes, direccionando los equipos de forma individual hasta un máximo de 64 direcciones, por grupos hasta un máximo de 16, o de forma simultanea mediante un comando “broadcast”. La configuración puede ser cambiada en cualquier momento sin necesidad de volver a cablear.

La señal es digital en niveles nominales de 0 V y 16 V (actualmente entre -6.5 V y +22.5 V, ya que cada nivel de lógica es ±6.5 V) con transferencia de datos mediante un protocolo serial, asíncrono y de medio dúplex a 1200 bits/s. El uso de una señal diferencial garantiza un funcionamiento confiable incluso sin la presencia de un ruido eléctrico importante, permitiendo el cableado de DALI a lo largo de los cables de la red principal o incluso en un cable de núcleo múltiple que transporta energía de la red principal.

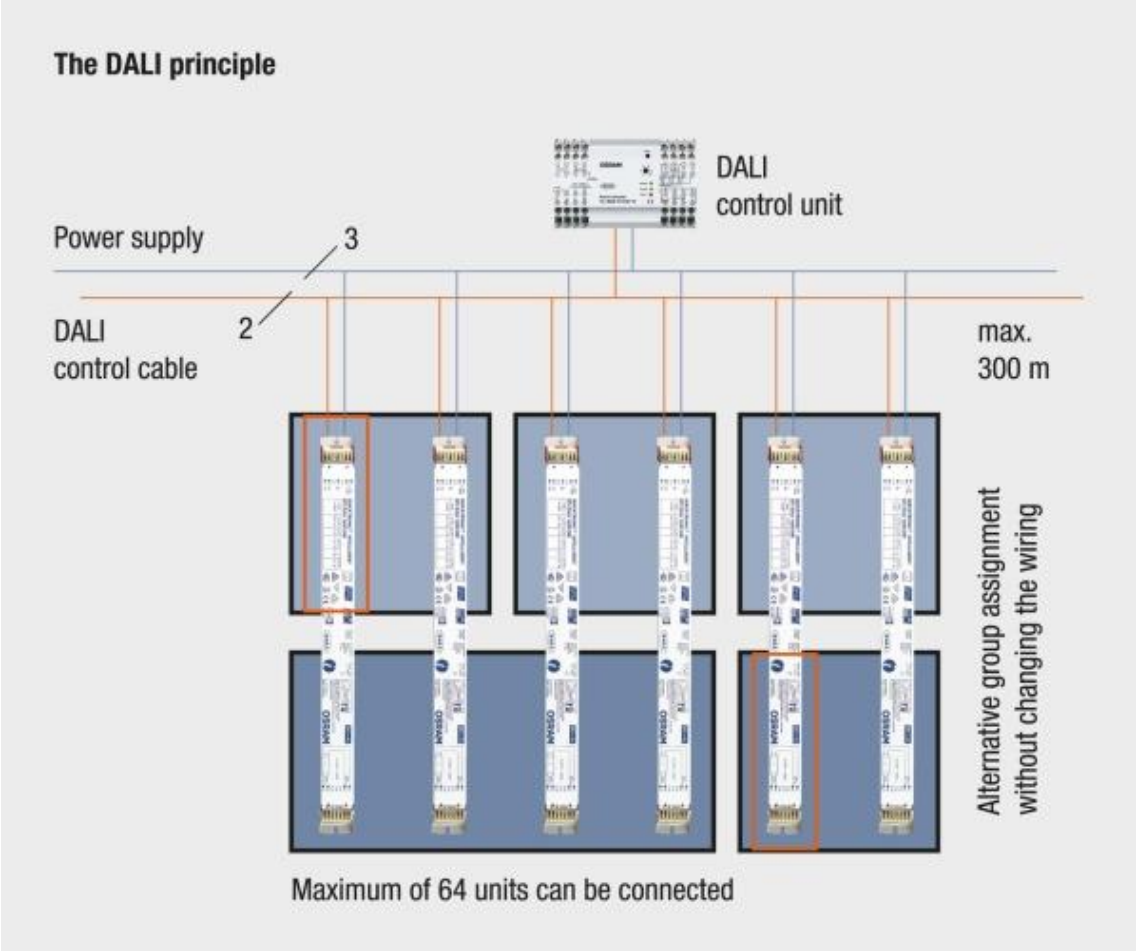


Figura 30.Diagrama de conexionado en protocolo DALI

5.3 Actualidad y futuro de la tecnología LED

Los avances tan extraordinarios en la tecnología LED están desencadenando la explosión de eficientes sistemas de iluminación de estado sólido (SSL-LEDs) de alto brillo como fuente luminosa, de gran eficiencia y rendimiento lumínico. Utilizada originalmente sólo en aplicaciones muy específicas (señales luminosas como testigos de la presencia de energía eléctrica), la tecnología LED se está empleando cada vez más y ha iniciado la conquista del mercado mundial de iluminación general de los espacios, sustituyendo las tradicionales lámparas incandescentes, halógenas y fluorescentes, y convirtiéndose en una fuente de luz que será dominante en el futuro.

Como siempre, el precio, los problemas derivados de la generación de calor y la inmediatez de las necesidades de los proyectos, retrasará un poco su utilización. La próxima década, el OLED (Organic LED) puede llegar a ser la gran estrella.

No está claro cuando el LED, la iluminación de estado sólido SSL, se convertirá en un estándar ni cuándo dejará de ser el “foco de atención”, pero según la AEO2006 (Anual Energy Outlook del 2006) el verdadero futuro de esta tecnología se sitúa hacia el año 2030, cuando esté completamente desarrollada y estandarizada, tanto los HBLEDs (High-Brightning LED) como los OLEDs, y alcance su máxima evolución, de forma que la repercusión sea tanto ambiental como económica.

Hay que pensar en el largo camino recorrido por otras tecnologías hasta posicionarse como fiables y asequibles para el usuario final. El HBLED con menos de 10 años en el mercado sigue en evolución y, actualmente, las normativas que le afectan son pocas (pero bastantes más que hace cinco años), y la mayoría han sido desarrolladas en Estados Unidos. En EEUU, en los últimos 5 años se ha puesto mucha más atención a los LEDs como un medio para obtener ahorro energético, así como en Asia, y sobre todo en China, que considera la SSL-LED como una prioridad nacional.

5.4 Fundamentos del LED

A diferencia de las lámparas incandescentes y fluorescentes, los LED no son fuentes de luz blanca por sí mismas. Los LED emiten luz casi monocromática, que los hace altamente eficientes para aplicaciones de color claro, como en los semáforos y las señales piloto. Sin embargo, para ser utilizado como una fuente de luz para iluminación la luz blanca es necesaria. Esta se puede lograr con LEDs de dos maneras principales:

- **Fósforo de conversión**, en el que un fósforo se utiliza en o cerca del LED para emitir luz blanca.

- **Sistemas RGB**, en el que se mezcla la luz de múltiples LEDs monocromático (rojo, verde y azul), lo que resulta en la luz blanca.

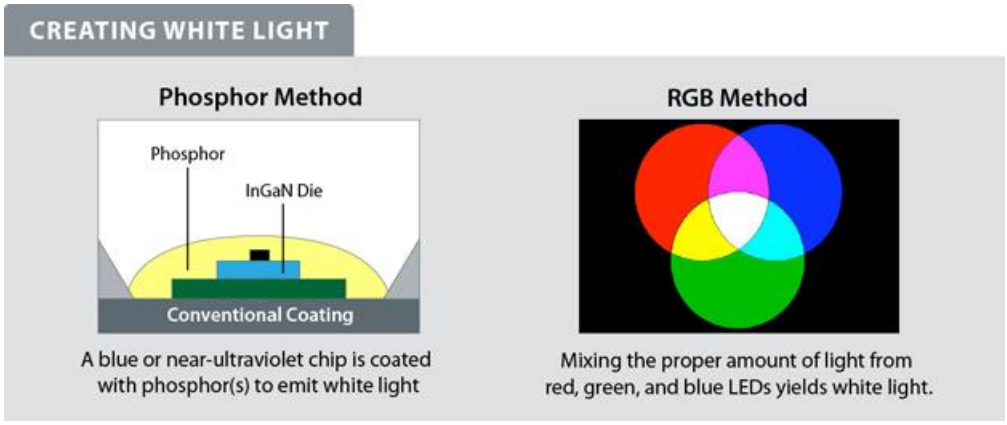


Figura 31. Creación de luz blanca en LEDs

5.5 Eficiencia energética de la iluminación LED

El uso de LEDs en el ámbito de la iluminación (incluyendo la señalización de tráfico e iluminación de viales) está alcanzando cotas de crecimiento de hasta un 300% anual, comenzando en el año 2007, en que se instaló iluminación LED en una vía pública por primera vez en Europa, en el paseo marítimo de Barbate (Cádiz) por un fabricante local. Sus prestaciones son superiores a las de la lámpara incandescente y la lámpara fluorescente, desde diversos puntos de vista. La iluminación con ledes presenta indudables ventajas: fiabilidad, mayor eficiencia energética, mayor resistencia a las vibraciones, mejor visión ante diversas circunstancias de iluminación, menor disipación de energía, menor riesgo para el medio ambiente, capacidad para operar de forma intermitente de modo continuo, respuesta rápida, etc. Asimismo, con LEDs se pueden producir luces de diferentes colores con un rendimiento luminoso elevado, a diferencia de muchas de las lámparas utilizadas hasta ahora que tienen filtros para lograr un efecto similar (lo que supone una reducción de su eficiencia energética).

Cabe destacar también que diversas pruebas realizadas por importantes empresas y organismos han concluido que el ahorro energético varía entre el 70 y el 80 % respecto a la iluminación tradicional que se utiliza hasta ahora. Todo ello pone de manifiesto las numerosas ventajas que los LEDs ofrecen en relación al alumbrado público.

Los LEDs de luz blanca son uno de los desarrollos más recientes y pueden considerarse como un intento muy bien fundamentado para sustituir los focos o bombillas actuales (lámparas incandescentes) por dispositivos mucho más ventajosos. En la actualidad se dispone de tecnología que consume el 92 % menos que las lámparas incandescentes de uso doméstico común y el 30 % menos que la mayoría de las lámparas fluorescentes; además, estos ledes pueden durar hasta 20 años. Estas características convierten a los ledes de luz blanca en una alternativa muy prometedora para la iluminación.

La eficiencia energética de las fuentes de luz se mide en lúmenes por vatio (lm/W), es decir, la cantidad de luz producida por cada vatio de electricidad que se consume. Esto se conoce como eficacia luminosa. A largo plazo el objetivo de la investigación y el desarrollo de la llamada de luz blanca LED está en llegar a los 160 lm/W en un umbral preparado para el mercado de los sistemas LED para el año 2025 e incluso mayor hasta 200 lm/W. Actualmente, los LEDs blancos más eficaces pueden llegar a ser considerados como similares a las lámparas fluorescentes. Sin embargo, hay varias consideraciones importantes, como se explica a continuación.

- Calidad del color:

Los LEDs tienen temperaturas más eficaces en corrección muy alta del color, a menudo por encima de 5000K (la temperatura del color se expresa en Kelvin), produciendo un "frío" luz azulada. Sin embargo, los LEDs

blancos cálido (2600K a 3500K) se han mejorado considerablemente, ahora su eficacia se acerca a la de las lámparas fluorescentes compactas. Además con una elevación en la apariencia del color, lo que hace que el color LED también esté mejorando: los principales LEDs blancos están ahora disponibles con un índice de reproducción cromática (CRI) de 80, equivalente a las lámparas fluorescentes compactas.

- Las pérdidas del conductor o driver:

Los fluorescentes y lámparas de descarga de alta intensidad (HID) con fuentes de luz no pueden funcionar sin un balasto, que proporciona un voltaje de partida y los límites de la corriente eléctrica a la lámpara. Del mismo modo, los LED requieren una fuente de alimentación (comúnmente llamado "driver").

La fuente de alimentación se convierte línea AC a la tensión de DC adecuada (habitualmente entre 2 y 4 voltios DC para LED de alto brillo) y actuales (por lo general 200 -1000 mA), y también puede incluir electrónica adicional para graduar y/o controlar la corrección del color.

Los Drivers LED disponibles en la actualidad suelen tener alrededor de 85% de eficiencia. Por lo tanto la eficacia del LED debe tener un descuento del 15% que se empleará por cuenta del conductor. Para una comparación aproximada, el rango de eficacias luminosas de las fuentes tradicionales y la disfunción eréctil de la lámpara, incluyendo la pérdida en el lastre del balasto o del conductor driver en su caso, se muestran a continuación.

Fuente de luz	**Rango típico de eficacia luminosa en lm/W (varía en función de la potencia y el tipo de lámpara)**
Incandescente (sin balasto)	10 - 18
Halógenas (sin balasto)	15 - 20
Fluorescente compacto (CFL) (incl. reactancia y cebador)	35 - 60
Tubo fluorescente (incl. reactancia y cebador)	50 - 100
Halogenuro metálico (incl. balasto)	50 - 90
LED Blanco frío < 4000K (incl. driver)	60 - 92
LED Blanco cálido < 4000K (incl.	27 - 54

Fuente de luz	Rango típico de eficacia luminosa en lm/W (varía en función de la potencia y el tipo de lámpara)
driver)	

Figura 32. Rango de eficacia luminosa

- Los efectos térmicos:

Las cifras de flujo luminoso citado por los fabricantes de LED se basan en un funcionamiento del LED a temperatura de servicio (**Tj**) de 25 ° C. Los LEDs se ponen a prueba durante la fabricación en condiciones que difieren de la realidad en un dispositivo o sistema. En general, el flujo luminoso se mide en un servicio instantáneo (20 o 25 milisegundos de pulso aproximadamente) al aire libre. **Tj** puede siempre ser más alto cuando se opera bajo una corriente constante de un aparato o sistema. Los LEDs incluso funcionando en un elemento bien diseñado con luminaria adecuada y buena disipación de calor pueden producir el 10% -15% menos de luz que el indicado en la especificación que le corresponde por calificación del "típico flujo luminoso".

5.6 Utilización de los LEDs

Hoy en día, un flujo constante de lámparas para utilizarlas como repuesto en sustitución de las antiguas y diversos accesorios de iluminación LED están entrando en el mercado doméstico de iluminación en general.

Los LEDs son a menudo elogiados por la eficiencia energética y larga vida. Si bien estas son consideraciones importantes, la selección de la iluminación se debe basar en otros muchos factores también.

Los productos LED son más competitivos en aquellas aplicaciones en las que sus características de rendimiento son superiores a sus costos de primera instalación. Además, desde el punto de vista de la rentabilidad llevada a la gestión térmica efectiva es más difícil de lograr en aquellas pequeñas luminarias LED y en las lámparas de repuesto que en las grandes operaciones de cambio, tales como el área de LED de la iluminación exterior que han sido los pioneros en la penetración en el mercado. No obstante, su utilización para la pequeña aplicación de los repuestos no es nada desdeñable en términos de ahorro energético directo.

El rendimiento de SSL-LED ha ido en constante aumento y en sintonía con la investigación y el desarrollo de los productos. Hoy en día, muchos productos bien diseñados pueden lograr una distribución de luz muy apropiada, además de una alta eficacia y potencia de luz adecuada. Segmentos de mercado donde los LED han hecho los mayores avances hasta el momento son los residenciales downlights empotrados (lámparas de luces descendentes), las dicroicas y halógenas de cuarzo en focos, la iluminación portátil de escritorio, y la iluminación del área al aire libre. El siguiente sector con gran potencial es el de la iluminación de grandes áreas de oficinas. Además de su mayor eficacia, los productos de SSL-LED pueden ser más controlables que las tecnologías tradicionales de iluminación y en su vida no están afectados por frecuentes tareas de mantenimiento.

5.7 Retos de I+D en tecnología LED

Como cualquier nueva tecnología, la iluminación de estado sólido debe demostrar un alto valor antes de que comience a ganar cuota de mercado que pueda desplazar las tecnologías anteriores. Durante la última década, la investigación y desarrollo han producido mejoras notables en el costo, el rendimiento del color, la salida de luz, la eficacia, la fiabilidad, duración y capacidad de fabricación de productos LED. En cuanto a los costos en concreto, la tasa de disminución ha sido de alrededor del 25% por año desde 2005. Como resultado, algunos productos de iluminación LED se han convertido en muy competentes si se los compara con sus homólogos convencionales en el mercado para ciertas aplicaciones

A pesar del rápido ritmo de su desarrollo, la I+D no ha llegado todavía a punto de alcanzar su pleno potencial. Queda aún mucho por hacer para mejorar aún más el rendimiento y reducir costes.

5.8 Ventajas del LED frente a los halógenos y fluorescentes

En el mercado existen múltiples sistemas iluminación, pero los más relevantes con los que el LED debe competir en sistemas de iluminación artificial son los halógenos y los fluorescentes. El LED es un producto relativamente novedoso y desconocido en este campo, frente a otros que llevan años en el mercado.

A continuación se muestran sus principales ventajas:

- **Bajo consumo:**

Las lámparas LED tienen un gran rendimiento por lo que necesitan menos energía para producir igual cantidad de luz.
Una lámpara halógena consigue rendimiento del orden de 20 lumens/W y las fluorescentes del orden de los 60 lumens/W mientras que una iluminación LED consigue fácilmente rendimientos de más de 100 lumens/W lo que supone un rendimiento 5 veces mayor que los halógenos y casi dos veces mayor que los fluorescentes.

- **Baja Tensión:**

La gran mayoría de lámparas LED se alimentan a 24V de corriente continua, adaptándose perfectamente a la mayoría de las fuentes de alimentación de los equipos, y reduciendo al mínimo los posibles riesgos de electrocución.

- **Pequeña anchura espectral:**

Los LEDs emiten luz en una pequeña banda espectral, esto es, emiten la luz en un único color que depende del tipo de LED, por lo que no necesitan filtros para obtener un determinado color, lo que evita la pérdida de energía. La iluminación monocromática evita la aberración cromática de las ópticas, mejora el rendimiento de los CCDs (Dispositivos de carga acoplada) y puede ser utilizado para evitar la influencia de iluminaciones exteriores.

- **Amplia gama espectral:**

El LED es un dispositivo de longitud de onda fija pero que está disponible para una amplia gama que nos permiten cubrir una gran banda del espectro que va desde la luz infrarroja hasta la UV.

- **Mayor rapidez de respuesta:**

El LED tiene una respuesta de funcionamiento mucho más rápida que el halógeno y el fluorescente, del orden de algunos microsegundos, ello lo hace ideal para funcionar con un estrobo (sistemas estroboscópicos). Al ser un semiconductor de estado sólido no sufre daños ni desgastes por el encendido y el apagado continuo, al contrario que los otros tipos de iluminación.

- **Luz más brillante:**

En las mismas condiciones de luminosidad que sus rivales, la luz que emite el LED es mucho más nítida y brillante.

- **Sin fallos de iluminación:**

Soporta las vibraciones a las que pueda estar sometido el equipo sin producir fallos ni variaciones de iluminación. Esto es debido a que el LED carece de filamento luminiscente evitando de esta manera las variaciones de luminosidad del mismo y su posible rotura.

- **Baja emisión de calor:**

Al consumir poca energía, las bombillas LED emiten poco calor. Es la llamada luz fría. Por ejemplo, una bombilla halógena gasta de 50W, 45 aproximadamente en emisión de calor, esto supone un gasto extraordinario en aire acondicionado, siendo necesarios unos 70W adicionales para deshacerse del calor generado por esa bombilla.

- **Ecológicos:**

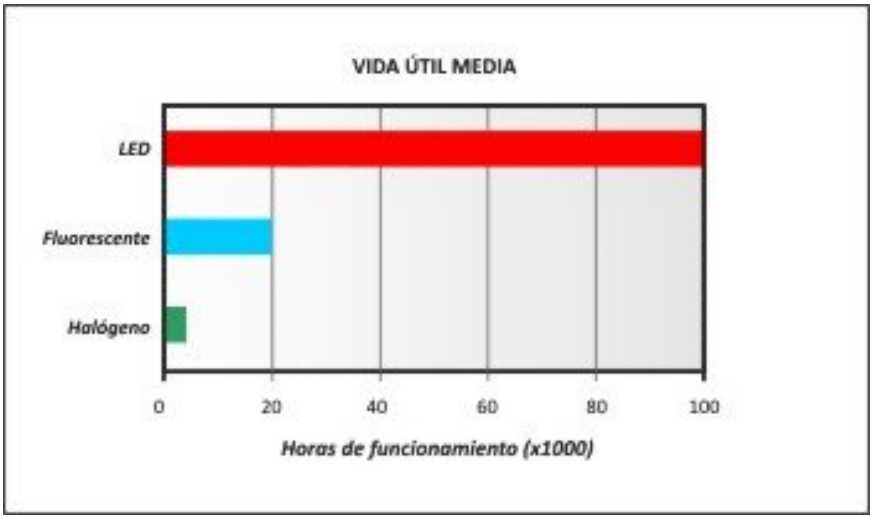
Las bombillas LED son totalmente reciclables y ecológicas ya que no contienen mercurio, ni materiales tóxicos como las lámparas fluorescentes, o CFL.

- **Alta rentabilidad:**

El mayor inconveniente que tiene el LED sin duda es su precio, pero si evaluamos sus múltiples e inmejorables condiciones de funcionamiento, y sobre todo su larga vida en comparación con los demás sistemas de iluminación, estamos en condiciones de afirmar que es la inversión más eficaz y rentable que se puede hacer.

- **Mayor duración y fiabilidad:**

La vida de un LED es muy larga en comparación con los demás sistemas de iluminación y su fiabilidad es mucho mayor ya que la degradación de la luz es mínima en relación a la de halógenos y fluorescentes, como se muestra en las siguientes figuras:



VIDA MEDIA	HORAS
LED	100.000
Fluorescente	20.000
Halógeno	4.000

PÉRDIDA DE LUMINOSIDAD	-20%	-30%
LED	45.000 h.	100.000 h.
Fluorescente	5.000 h.	20.000 h.
Halógeno	1.500 h	4.000 h.

Figura 33.Gráficas comparativas de vida media y pérdida de luminosidad

5.9 Estrategias de avance en LEDs

La eficiencia del alumbrado se basa en convertir el máximo de la energía que se consume en luz y el mínimo en calor, de esta manera se reduce el gasto energético para enfriar el edificio, que no es poco, ya que se puede cuantificar entre el 10 y el 20%. Además de realizar el cambio de luminarias y/o lámparas se deberá incorporar un sistema que regule el flujo luminoso de los equipos en función del nivel de luz natural que existe en las estancias. La combinación de los dos tipos de iluminación, natural y artificial aumenta el confort visual y reduce el gasto energético.

Acelerar el ritmo de comercialización de SSL-LED dependerá de la mejora en la competitividad de los costes de producción y de la calidad del producto. Para ello se cuenta con la experiencia de los aciertos y errores cometidos anteriormente en el lanzamiento al mercado de las lámparas fluorescentes compactas (CFL) en la década de 1990.

Los fabricantes mundiales LED esperan beneficiarse de un aumento de la demanda en los próximos años. Grandes empresas como Osram, OS, KLA Tencor, ULVAC, y fabricantes de equipos "(metal organic chemical vapor deposition)" MOCVD, Veeco y Aixtron, están empeñados en hacer que la tecnología empleada en el recubrimiento del flujo laminar de los chip LED pueda contribuir a la mejora de la calidad del LED reduciendo los costos. Con respecto al material y los componentes, los taiwaneses principales fabricantes, Epistar y FOREPI nos presentaran las últimas novedades y tendencias de futuro de los dispositivos para el LED en la iluminación.

Los principales objetivos para lograr avances son:

- Mejora de la competitividad de costes:

Un obstáculo importante para SSL es el costo. Hoy en día, el precio de compra de productos de iluminación LED es generalmente más alto que la de sus contrapartes convencionales, a menudo por un amplio margen, y el ahorro de energía a menudo no son suficientes para compensar la diferencia dentro de un plazo de amortización atractivo. Sin embargo, se está llevando por la industria una reducción de costes agresiva, y la tasa actual de disminución de costos en sistemas de iluminación LED es de aproximadamente 20 por ciento al año. Así, mientras que aún se piense que los LED no resultan rentables hoy en día, hay una gran cantidad de información que muestra que seguirá disminuyendo rápidamente su precio.

- Mejora de la calidad:

Existen aún problemas de bajo rendimiento en productos que todavía no están preparados para competir con las tecnologías de titular, incluidos los productos que no cumplen con las demandas de rendimiento del fabricante. Se debería crear una opinión pública respecto al grado de excelencia exigible del producto LED educando a los consumidores, así como a la comunidad de iluminación sobre las reclamaciones habidas y otros riesgos acaecidos en el mercado SSL.
No hay duda de que la iluminación de estado sólido SSL-LED en última instancia, va a surgir como la tecnología preferida para una variedad de aplicaciones sin precedentes, ya que, en igualdad de condiciones, todos queremos ahorrar energía y proteger el medio ambiente. De todas maneras, no podemos generalizar, por lo que debemos enfatizar repetidamente que el SSL es el elemento adecuado para algunas aplicaciones pero no para todas, ya que el profundo conocimiento del caso y la debida diligencia a aplicar deben ser los elementos clave en cualquiera de las especificaciones de iluminación y en el proceso de compra.

- La creciente innovación tecnológica:

Cada día que pasa la innovación tecnológica de las empresas del sector está impulsando el crecimiento de una industria muy competitiva en la iluminación SSL-LED. Esto es algo que continuará sucediendo durante los próximos años. Importantes inversiones de las empresas en I + D reflejan un amplio crecimiento en esta tecnología. Ello abarca una amplia gama de tecnologías digitales y algoritmos relacionados con la unidad, de redes y control de aplicaciones de iluminación LED del estado sólido.

6 IMPLEMENTACIÓN DE LA UNIDAD DE CONTROL E ILUMINACIÓN

6.1 Configuración de los módulos Xbee

Lo primero que hacemos es configurar cada uno de los módulos Xbee para la emisión y recepción a través de radiofrecuencia con el fin de transmitir datos de un módulo al otro de forma inalámbrica.

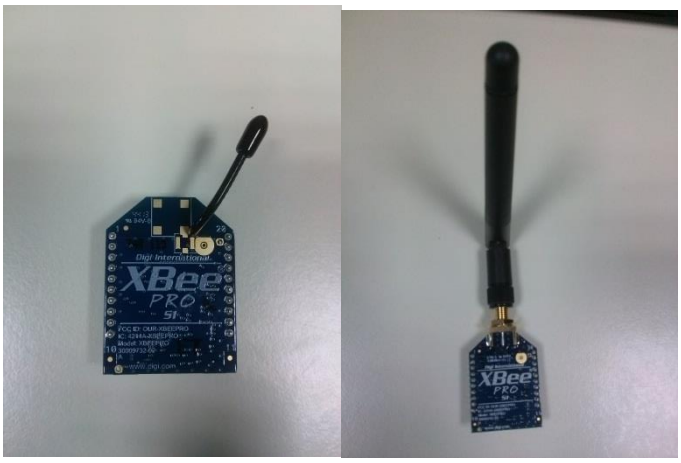


Figura 34. Módulos Xbee

Para ello usaremos el software facilitado por el fabricante en su página web, denominado X-CTU, donde podremos, de manera más sencilla, modificar los parámetros necesarios para conseguir dicha transmisión de información.

En el kit adquirido para la comunicación de los módulos Xbee se encuentran también 2 placas denominadas *Interface board USB* cuya utilidad es la comunicación a través de puerto USB de los módulos al ordenador para proceder de esta manera a la configuración de los Xbee utilizando el programa X-CTU.

Figura 35. Interface USB Board

Conectamos los módulos sobre las placas en los pines habilitados para ello de la manera que se muestra en la *Figura 3*:

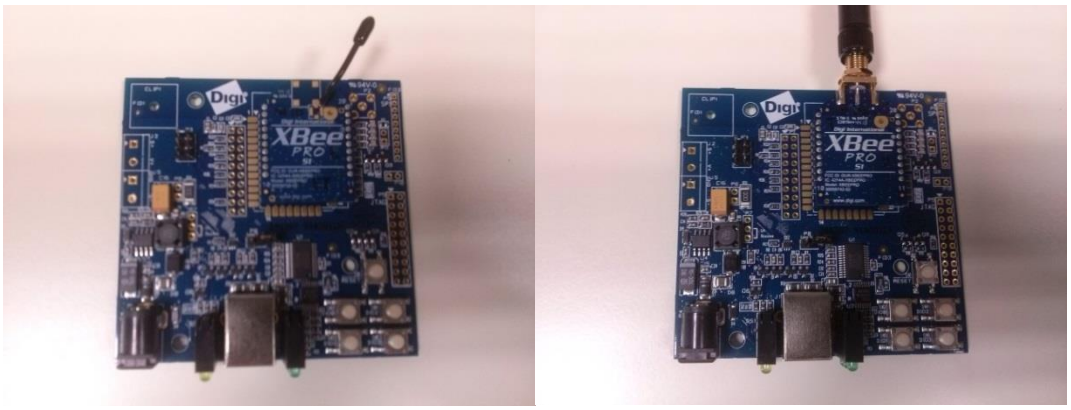


Figura 36. Xbee sobre Interface USB Board

El siguiente paso sería conectar las placas, (con los Xbee en conexión a ellas) mediante cable USB, al ordenador en donde tenemos instalado el software X-CTU en disposición para empezar la configuración de los módulos.

Figura 37.Módulos conectados por cable USB al PC

Una vez nuestro ordenador encuentre los módulos, comenzaremos con la configuración de parámetros en el software para conseguir la comunicación deseada.

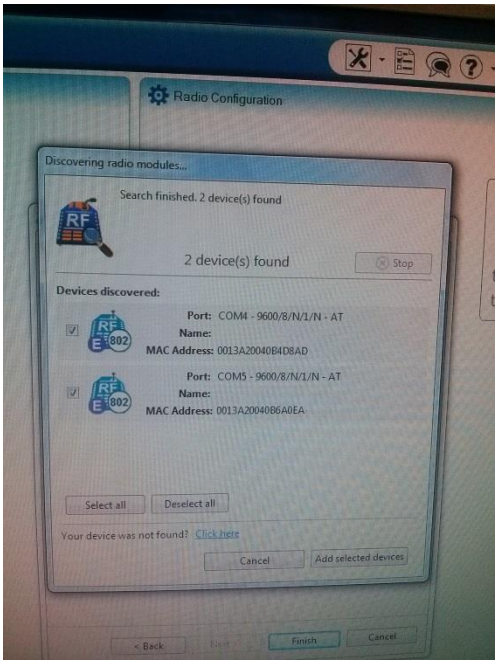


Figura 38. PC encuentra los módulos

Los módulos Xbee vienen, cada uno, con una dirección MAC de fábrica, que está situada en la parte trasera de cada unidad y que debemos tener en cuenta ya que a través de estas direcciones podremos conseguir, de manera sencilla, la comunicación inalámbrica entre los módulos.

Figura 39. Dirección MAC de los módulos

También el software X-CTU nos proporciona dicha información una vez reconocido cada Xbee por el ordenador.

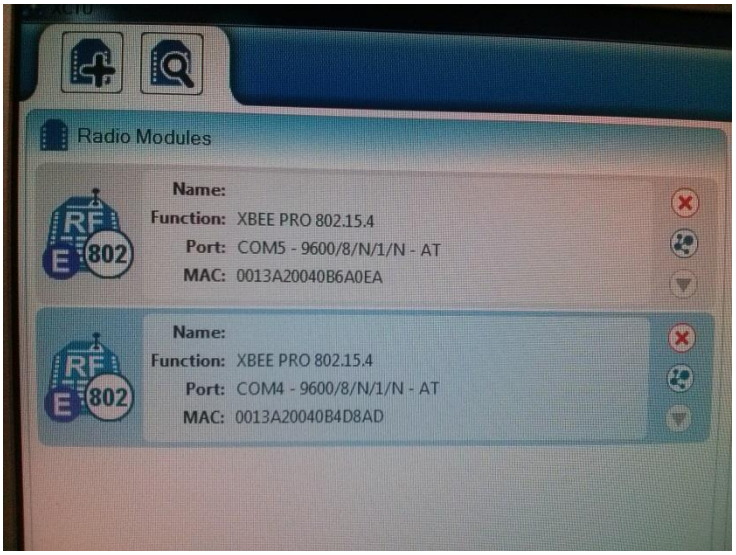


Figura 40. Dirección MAC en X-CTU

Como vemos en las ***Figuras 39 y 40***, las direcciones MAC de nuestros módulos son:

Emisor: 0013A20040B6A0EA

Receptor: 0013A20040B4D8AD

Ahora que ya tenemos ambos Xbee localizados en el software, deben configurarse los parámetros para que los módulos se reconozcan entre si y, de esta manera, enviar y recibir información entre ellos.

En este paso, configuraremos los parámetros DH (Destination Address High) y DL (Destination Address Low) de cada módulo, como se muestra en las ***Figuras 41*** **y** ***42***, modificando los valores de fábrica por las direcciones MAC que obtuvimos anteriormente. En el Xbee que actuará como emisor, debemos introducir los parámetros del receptor y viceversa, insertando los 8 primeros dígitos de la dirección MAC en el recuadro correspondiente a DH y los 8 posteriores en la ubicación que corresponde a DL.

De esta manera, en el apartado DH del emisor deberemos introducir **“0013A200”** y el DL **“40B4D8AD”**.

De igual forma en el lugar correspondiente a DH del receptor escribiremos **“0013A200”** y en DL **“40B6A0EA”**.

En estas figuras se muestra la configuración de los parámetros DH y DL tanto en emisor como en receptor:

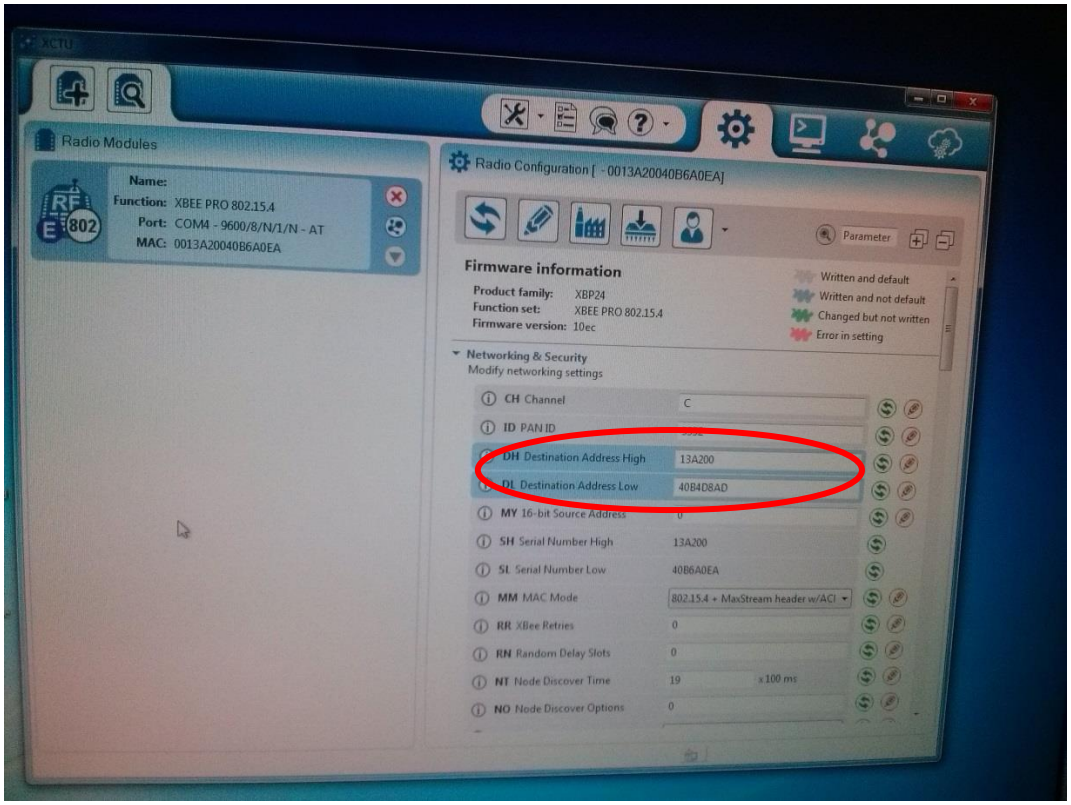


Figura 41. Parámetros DH y DL del emisor

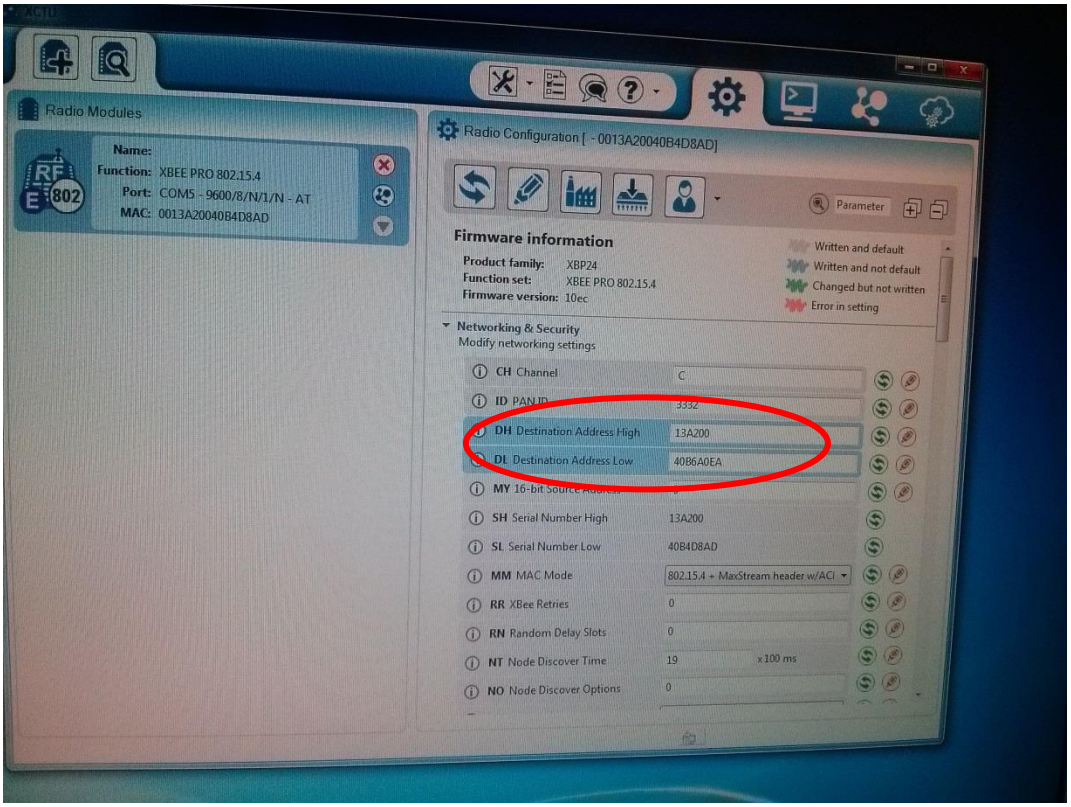


Figura 42. Parámetros DH y DL del receptor

Debemos asegurarnos que los datos quedan registrados pulsando en el círculo rojo (con la figura de un lápiz en su interior), en cada apartado modificado para que el programa realice los cambios correctamente.

6.2 Envío y recepción de datos entre módulos Xbee.

Una vez que tenemos los módulos configurados para la transmisión de información entre ellos, debemos asegurarnos de que, efectivamente, los módulos intercambian información.

Para ello, utilizaremos el modo *Consolers Working* del X-CTU, situado en la parte superior del programa:

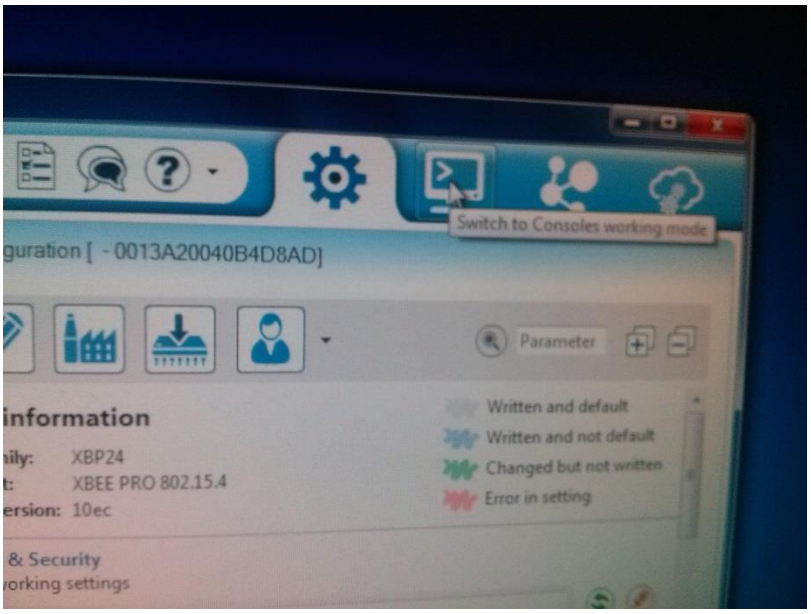


Figura 43. Modo Consoles working

Nuestro objetivo en este paso es enviar la información que deseamos de un módulo a otro.

En primer lugar debemos indicarle al programa que abra la conexión con los módulos pulsando el recuadro que aparece en la ***Figura 44***:

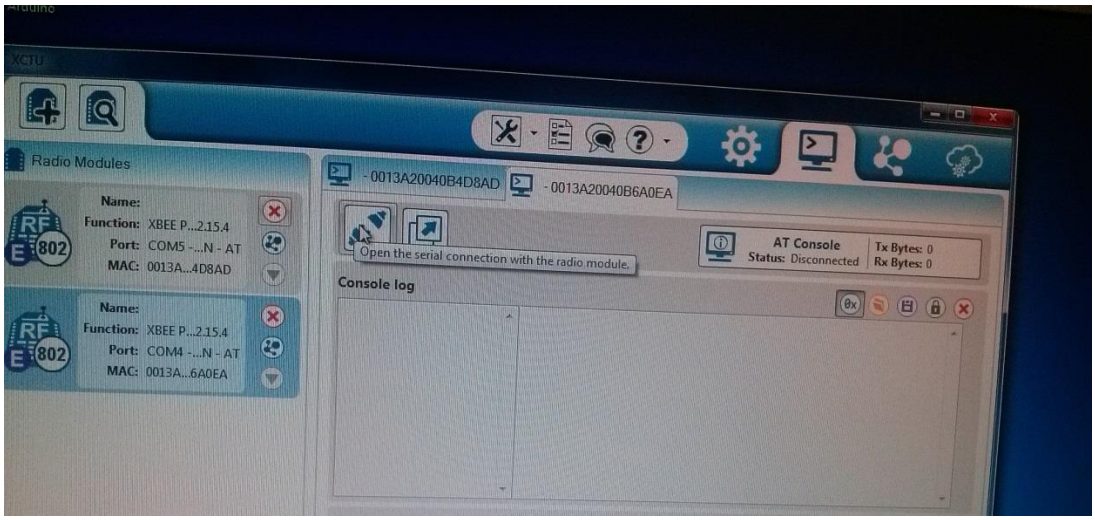


Figura 44. Conexión en serie entre PC y Xbee

Una vez realizado este paso, nos iremos a la parte inferior de la pantalla, en el apartado "*Send a single packet*", donde aparece un círculo verde en cuyo interior se encuentra el signo "+" como aparece en la ***Figura 45***:

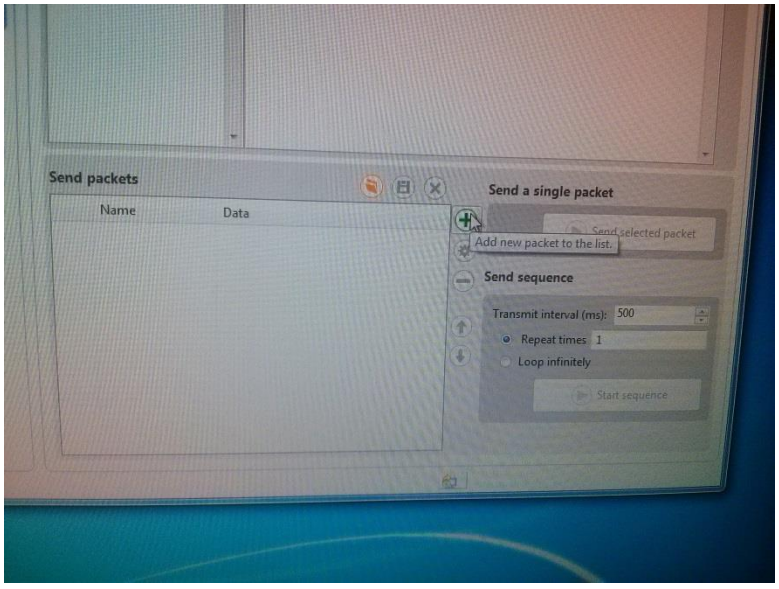


Figura 45. Añadir paquete de datos a enviar

Debemos pulsarlo y a continuación nos aparecerá un lugar en blanco con las opciones "ASCII" y "HEX" (hexadecimal), que son las dos formas de datos que permite enviar Xbee. En ese espacio en blanco escribiremos los datos que queremos enviar de un módulo a otro de manera inalámbrica.

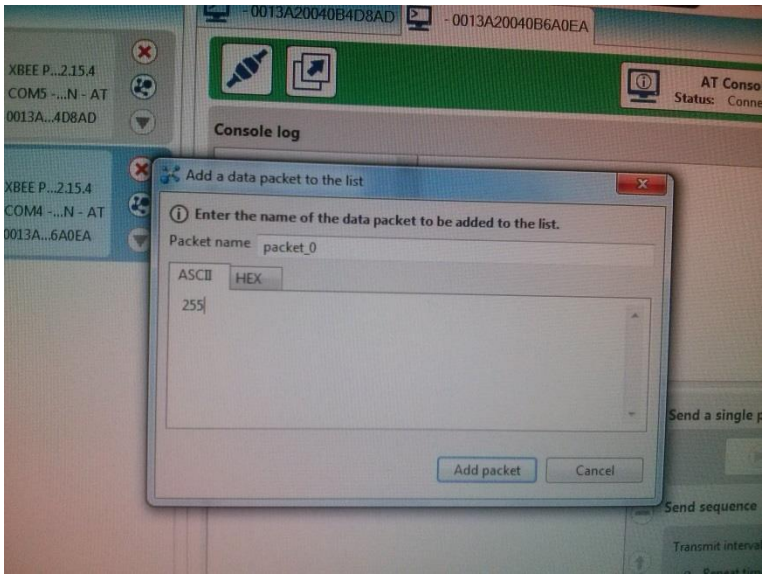


Figura 46. Paquete de datos a enviar

Cuando hayamos finalizado de introducir los datos que queremos, pulsaremos la pestaña "*Add packet*". A continuación se activará la pestaña "*Send selected packet*" como se muestra en la ***Figura 47***, y pulsándola estaremos enviando los datos que introdujimos anteriormente al receptor:

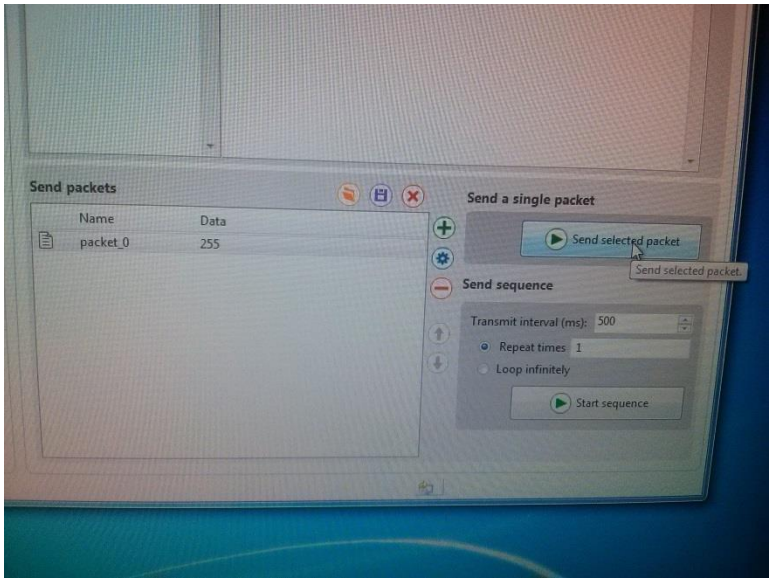


Figura 47. Envío de paquete de datos

Comprobamos que tanto el envío, como la recepción de datos es correcta. En el apartado "*Console log*" comprobamos como están en azul los datos que acabamos de enviar, tanto en ASCII como en hexadecimal. Esto confirma el envío de datos.

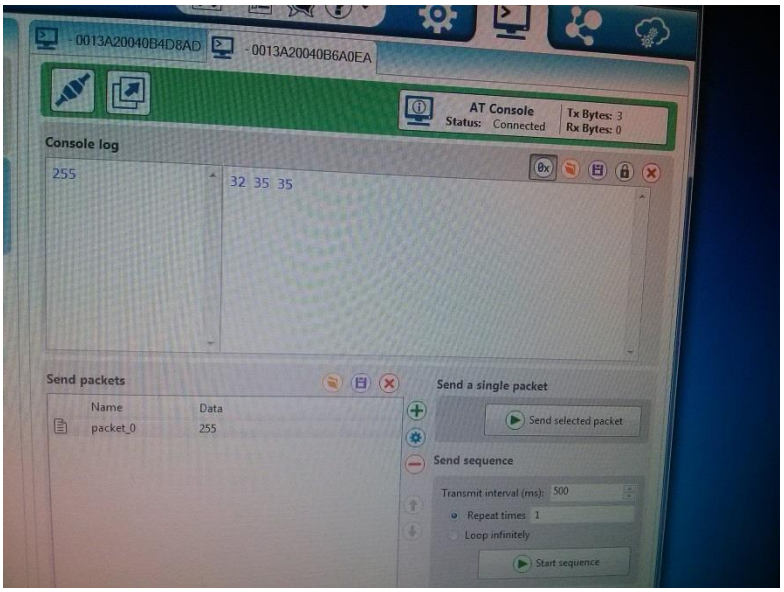


Figura 48. Confirmación de envío de datos

Si seleccionamos ahora la pestaña del receptor, comprobaremos que los mismos datos nos aparecen en rojo. Eso significa que la recepción de datos de un dispositivo a otro fue correcta.

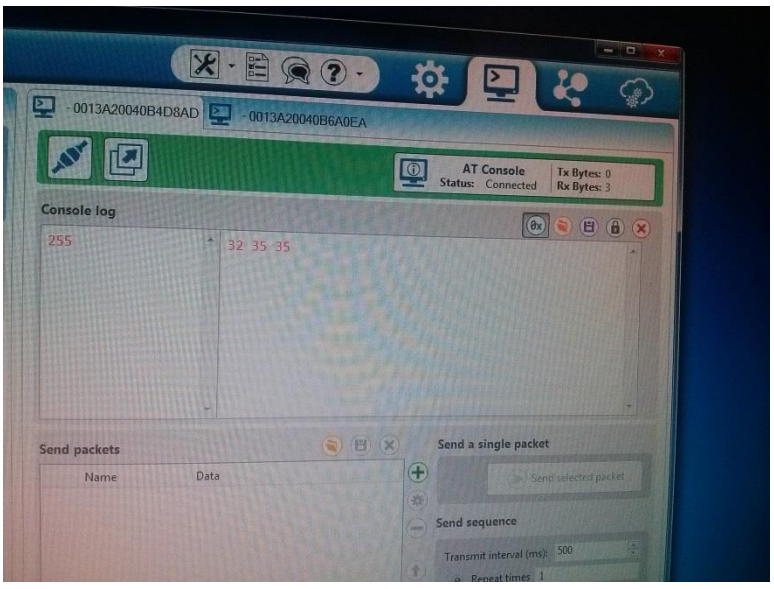


Figura 49. Confirmación de recepción de datos

En este proyecto fin de carrera el receptor no estará conectado al ordenador por lo que no podemos utilizar el programa X-CTU para comprobar que los datos han sido enviados correctamente.

Para confirmar esta recepción, la placa *Interface USB Board* sobre la que irá situado el Xbee receptor, dispone de 6 leds (3 a cada lado del puerto USB), que indicarán actividad en el módulo.

Las 3 luces situadas a la izquierda del puerto USB nos darán información sobre el estado en el que está en ese momento:

Led amarillo (superior) parpadeando: Se están recibiendo datos

Led verde (medio) parpadeando: Se están enviando datos

Led rojo (inferior) parpadeando: La placa está alimentada

Las 3 luces situadas a la derecha del puerto USB nos dan información sobre la recepción correcta de datos. De esta forma, si después de enviar un paquete de datos desde el emisor, las 3 luces del *Interface USB Board* del receptor están iluminadas durante unos segundos, significará que la información ha sido recibida.

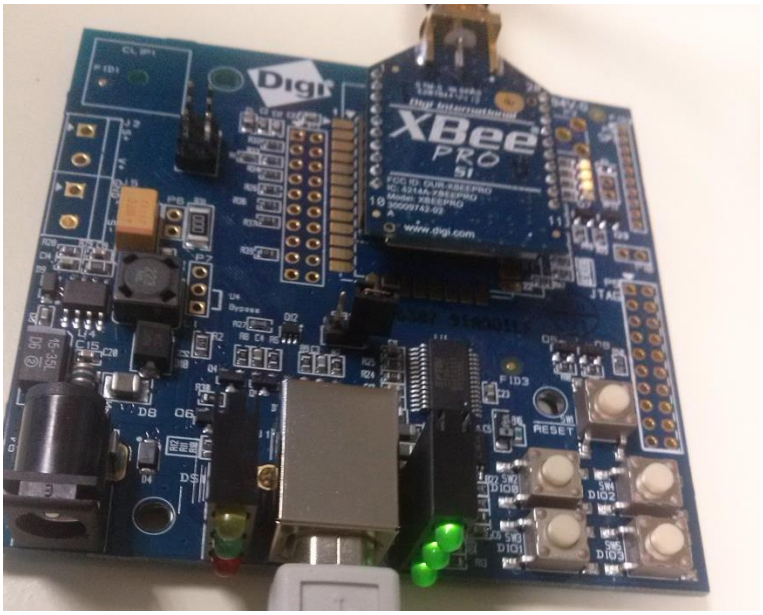


Figura 50. Leds encendidos confirmando la recepción de datos

6.3 Objetivo de la programación de Arduino

Para poder controlar la iluminación de la lámpara LED correctamente necesitaremos tener control sobre la tensión de salida del controlador (Arduino) con el fin de regular la intensidad de luz que se emitirá.

Para ello debemos programar una salida PWM (Pulse Width Modulation ó Modulación por Ancho de Pulsos) y tener la capacidad de poderlo manejar a nuestra conveniencia para lograr los distintos niveles de intensidad lumínica que queramos en cada momento.

Utilizaremos una de las salidas PWM que nos ofrece Arduino para conseguir dicha señal y modificaremos el valor de la tensión que nos interese en cada momento a través del módulo de radio frecuencia (XBee).

Es importante también establecer una frecuencia adecuada, para evitar interferencias y que no sea molesta al oído humano.

Conectaremos también el detector de presencia al Arduino y programaremos el controlador para que el circuito funcione de manera autónoma cuando no queramos tener un control continuo sobre él y de esta manera el funcionamiento sea acorde a lo que se necesita en cada momento.

6.4 Selección de pines

Utilizaremos Arduino UNO para la realización de nuestro proyecto, que nos permitirá disponer de las entradas y salidas pertinentes en cada caso para que todo funcione correctamente.

- Para el detector de presencia utilizaremos el pin 2 de Arduino (entrada digital).
- Para la conexión con el receptor de radio frecuencia utilizaremos los pines 0 y 1 (Rx,Tx) que actuarán de puerto serie para el envío y recepción de datos entre la placa Arduino y el módulo Xbee.
- Seleccionaremos el pin 9 del controlador ya que nos ofrece una salida PWM variable que nos permitirá controlar el ciclo de trabajo deseado en cada momento.
- Los pines de tensión de 5V, 3,3V y GND serán utilizados también para alimentar los circuitos.

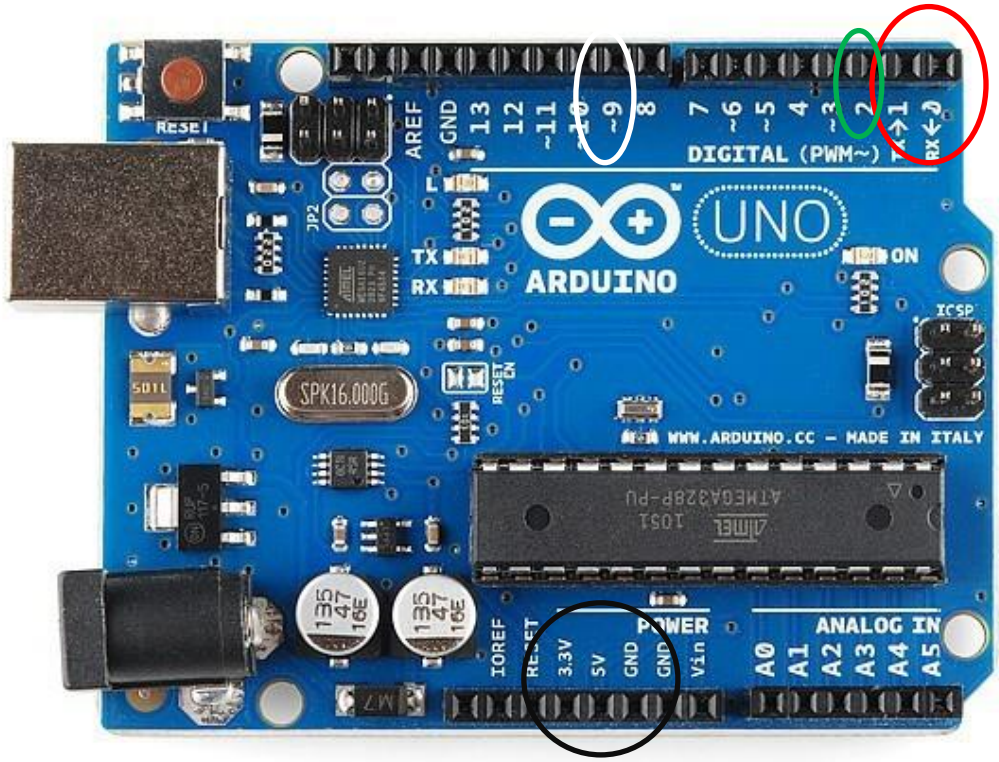


Figura 51.Pines utilizados

6.5 Programación de Arduino

Para realizar el programa o *sketch* necesario para que la placa funcione como queremos se utilizará *Arduino IDE* que es el software de programación de código abierto proporcionado por el fabricante en su página web para la programación del controlador.

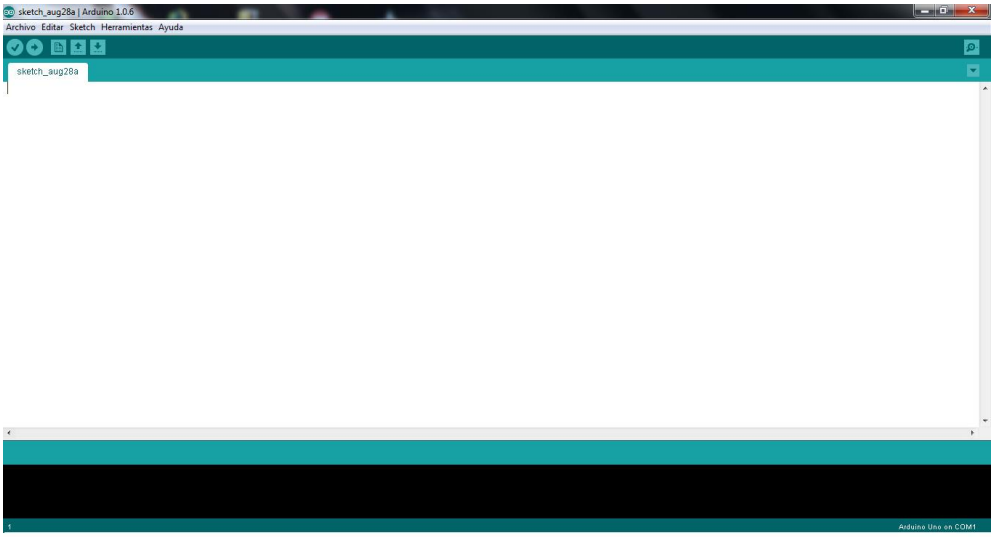


Figura 52.Entorno programación Arduino

Una de las premisas que utilizaremos para la realización del proyecto es que la frecuencia sea adecuada para no producir sonido alguno al oído humano. Por ello utilizaremos una frecuencia de 25 KHz. Sin embargo la frecuencia de las salidas PWM de Arduino UNO es de 490 Hz. Por esta razón deberemos conseguir de alguna manera la frecuencia que nos hemos propuesto como objetivo que es de 25 KHz.

6.5.1 Librería de frecuencia segura de PWM:

El comportamiento en frecuencia de las salidas PWM está limitado por componentes integrados llamados temporizadores. Estos temporizadores necesitan ser manipulados para conseguir la frecuencia de salida que necesitamos.

Sin embargo, nosotros utilizaremos la biblioteca “PWM.h” con el fin de conseguir nuestra frecuencia de 25 KHz, sin entrar en modificaciones de temporizadores del microcontrolador ATMega328 externas a las que realice la biblioteca que podría ocasionarnos algún conflicto interno a la hora de realizar nuestra programación.

La biblioteca cuenta con 5 funciones globales:

InitTimers () **//** Inicializa todos los temporizadores

InitTimersSafe () **//** Igual que **InitTimers ()** excepto temporizador 0 no se inicializa con el fin de preservar las funciones de tiempo de mantenimiento.

pwmWrite(pin,val)	***//*** Igual que 'analogWrite ()', pero sólo funciona con temporizadores inicializados.
SetPinFrequency (pin, frecuencia)	***//*** Establece la frecuencia del pin (en Hz) y devuelve una comprobación de que se la frecuencia se estableció correctamente.
SetPinFrequencySafe (pin, frecuencia)	***//*** Igual que SetPinFrequency excepto que no afecta temporizador 0.

Una vez que tenemos la librería que nos asegurará una frecuencia de 25 KHz deberemos instalarla en el *Arduino IDE* con el fin de que Arduino la reconozca y podamos utilizarla para nuestro propósito.

Para ello abriremos el IDE de Arduino y seleccionaremos *Sketch* en las pestañas superiores, a continuación seleccionaremos *Importar Librería* y en ese sub-apartado escogeremos la opción *Add Library.* Lo siguiente es ir a la ruta donde tengamos el archivo .zip de la librería y seleccionarlo.

Si regresamos al Arduino IDE y volvemos a *Sketch>>Importar Librería,* en la parte inferior observamos como aparece nuestra librería instalada pudiéndose seleccionar y ser utilizada.

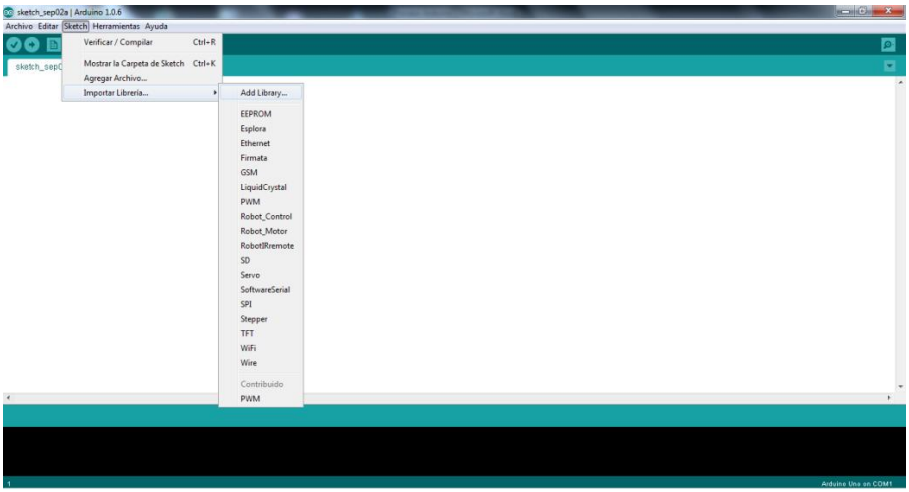


Figura 53.Instalación de librería PWM

6.5.2 Código de programación

Una vez tenemos instalada la librería procedemos a realizar el código para posteriormente implementarlo en nuestro Arduino Uno.

Nuestro propósito es conseguir el ciclo de trabajo que queramos siempre y cuando se lo ordenemos.

Cuando le enviemos el valor “01” (escogido para realizar esta función), el sistema funcionará de manera autónoma, siendo el detector de presencia el que le dará la orden de apagado y encendido cuando detecte a alguna persona.

El código es el siguiente:

```
#include <PWM.h>                          //lamamos a la librería PWM

int32_t frequency = 25000;                //Declaramos la frecuencia de
                                          25000Hz


void setup ()                             //Inicializa la configuración inicial del
                                          programa
{
  Serial.begin(9600);                     //Inicializa comunicación serie a
                                          9600 baudios
  InitTimersSafe();                       //Inicializa los temporizadores
  bool success = SetPinFrequencySafe (9,frequency);   //Establece la
                                                      frecuencia de
                                                      25000 Hz en el
                                                      pin 9
  if(success) {                           //Si se ha establecido la frecuencia
                                          correctamente nos envía al pin 13
                                          una señal de confirmación
    pinMode(13,OUTPUT);
    digitalWrite(13,HIGH);
  }
  pinMode(9,OUTPUT);                      //Declara el pin 9 como salida
  pinMode(2,INPUT);                       //Declara el pin 2 como entrada
```

```
}

void loop()                          //Inicializa en programa
{ int Valor;                         //Declaramos el entero “Valor”
int detector;                        //Declaramos el entero “detector”
if (Serial.available())              //Si hay datos en el puerto serie
  { if (Valor==1)                    //Si “Valor” tiene el valor 1
  {detector = digitalRead(2);        //La variable “detector” toma el valor
                                     de la entrada digital del pin 2
  if (detector == HIGH) {            // Si detector está en la posición
                                     HIGH
    pwmWrite(9, 255);                //Escribe en el pin 9 la salida 255
                                     (5V).
  }
 else {                              //Si detector no está en HIGH
   pwmWrite(9, 0);                   //Escribe en el pin 9 el valor 0 (0V)
  }
 }
else {                               //Si “Valor” no tiene el valor 1
  Valor = Serial.read();             //”Valor” toma el valor de los datos
                                     que estén entrando por el puerto
                                     serie
  pwmWrite(9,Valor);                 //Se escribe en el pin 9 el valor que
                                     tenga la variable “Valor”
  delay(30);                         //Pausa de 30 ms
       }
   }
}
}
```

6.6 Implementación de código en Arduino

Ya tenemos el código programado y ahora debemos pasarle ese programa a Arduino que será el encargado de interpretarle y efectuar las funciones que se le ordenan.

Para ello necesitaremos conectar la placa Arduino Uno al PC para poder traspasarle la información al controlador. Esta conexión se realizará a través de un cable USB, que además de proporcionarnos el vínculo con la computadora nos dará la alimentación necesaria para el funcionamiento de la placa.

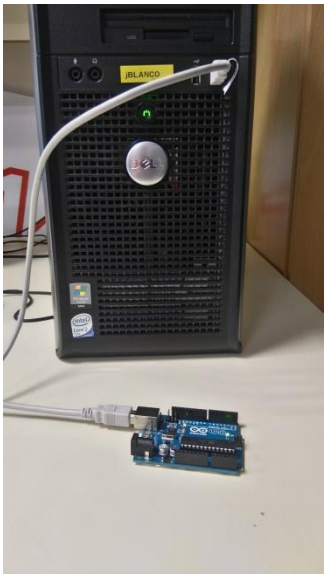


Figura 54.Conexión Arduino-PC

El PC reconocerá el Arduino y también lo hará el IDE aunque le tendremos que indicar en que puerto está y la placa con la que estamos trabajando.

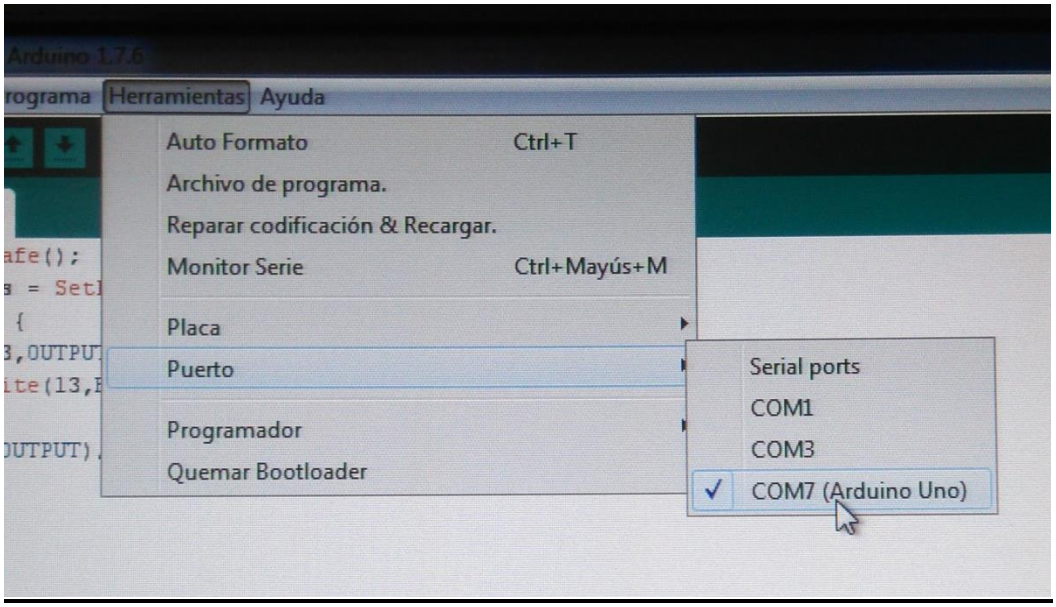


Figura 55.Selección de puerto

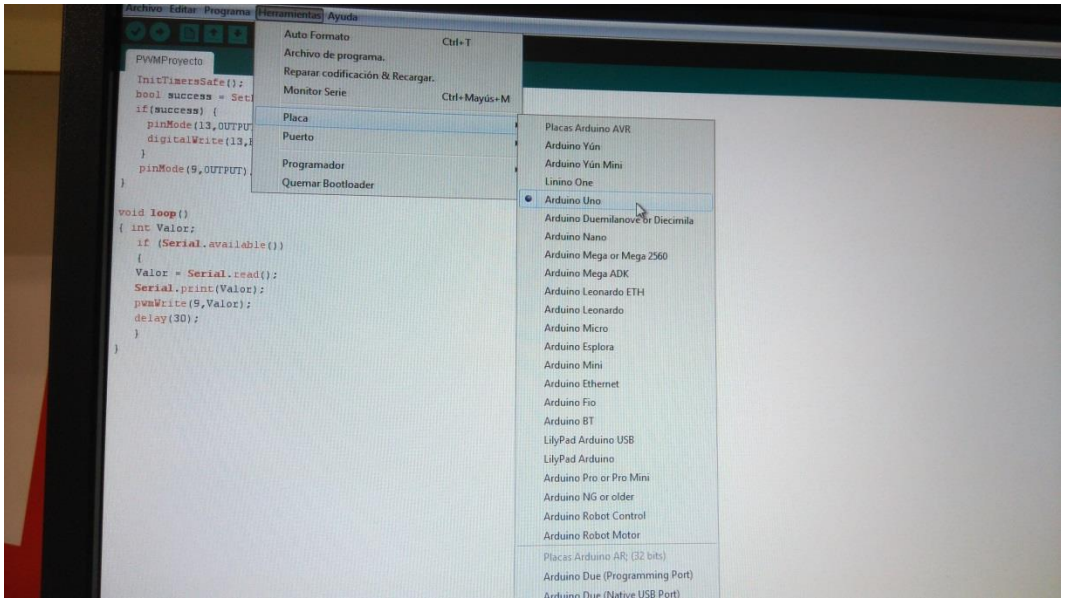


Figura 56.Selección de placa

Una vez que el PC nos reconozca el controlador, abrimos nuestro programa y debemos verificar que el código está correctamente realizado y posteriormente lo cargaremos a nuestra placa para, de esta forma, tener preparado ya el Arduino para realizar las funciones que le pedimos en esa programación.

Esto se lleva a cabo en la parte superior del IDE de Arduino donde encontramos la tecla de “Verificar” y a su lado la de “Cargar” que son las que debemos indicar para completar el proceso. Si todo está correcto en la parte

inferior hay una consola que nos indica que tanto la verificación como el vuelco del programa a la placa han sido satisfactorios. Si no es así, nos mostrará el error y la línea en la que se ha producido el fallo.

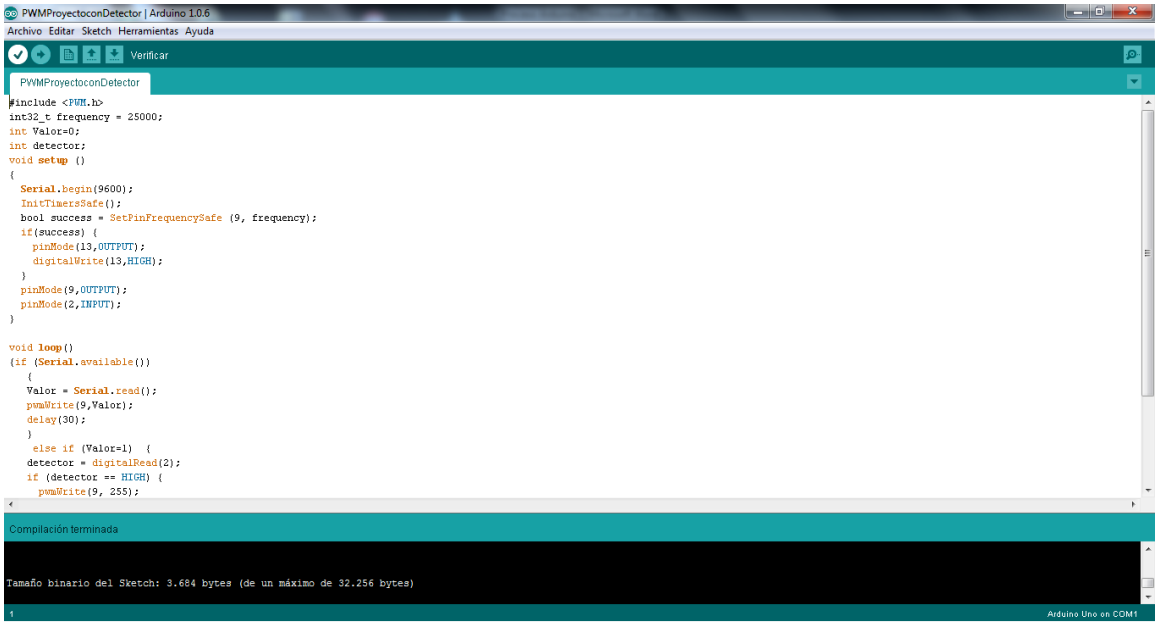


Figura 57.Verificación y carga del código a placa

Una vez que todo esto esté correcto ya podremos interconectar todos los componentes y realizar la comprobación de que se realiza lo que buscamos.

7. FUNCIONAMIENTO AUTÓNOMO Y CONTROLADO

7.1 Interconexión de los componentes

Ya tenemos configurados y programados todos los componentes del proyecto y ahora debemos conectarlos entre sí para que cada uno realice la función esperada y conseguir una iluminación controlada.

Lo primero que haremos será interconectar el módulo de radiofrecuencia Xbee con la placa Arduino.

El objeto de esta conexión es la de enviar información del módulo emisor del Xbee al módulo receptor, el cual estará conectado por el puerto serie al Arduino Uno. La placa, a través de la programación, nos proporcionará la salida PWM que le indiquemos por medio del Xbee. De esta manera podremos regular el ciclo de trabajo que nosotros queramos en cada momento con el fin de modificar la intensidad lumínica de la lámpara LED dependiendo de nuestras necesidades.

Para todo esto hemos de conectar de forma serial los módulos receptor del Xbee y el Arduino Uno.

Nos ayudaremos de un módulo que hará de enlace entre las dos placas y nos proporcionará una conexión segura además de comodidad para trabajar de manera adecuada con ellas. Este módulo se denomina Xbee Shield.

Figura 58.Xbee Shield

Como se puede ver en la ***Figura 58*** se trata de una placa que hace de puente entre los dos módulos.

El siguiente paso es montar el módulo receptor Xbee sobre el Xbee Shield y este a su vez montarlo sobre el Arduino haciendo que sus pines encajen en sus zócalos correspondientes.

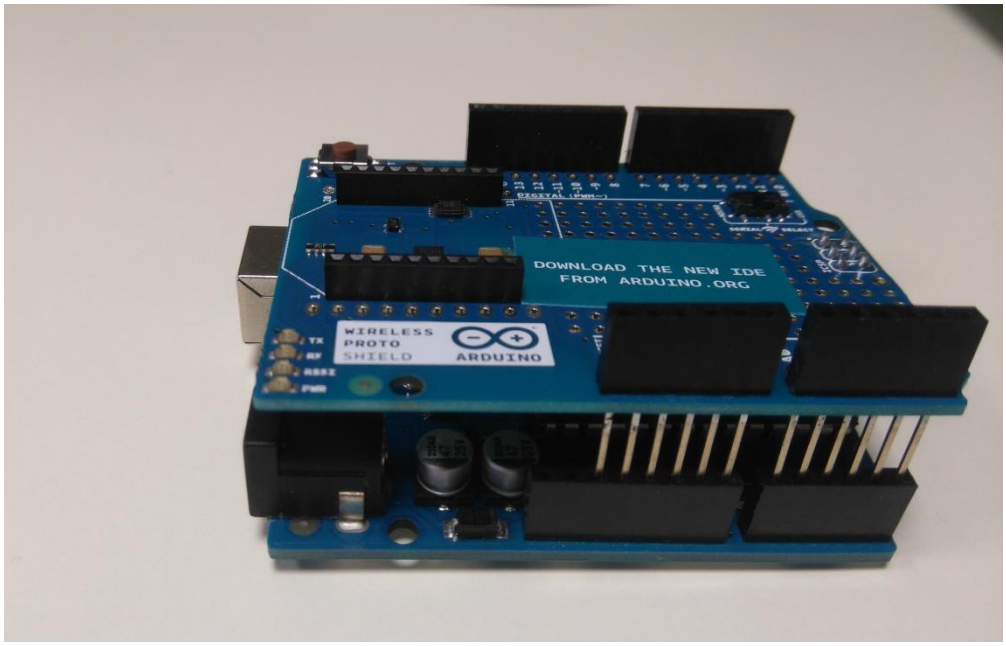


Figura 59.Xbee Shield sobre Arduino

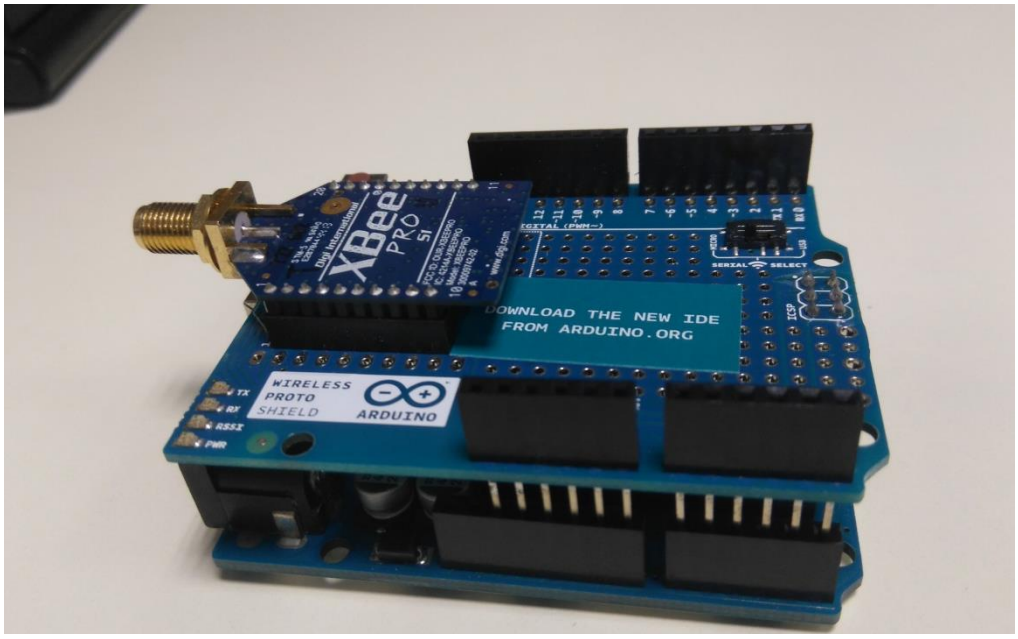


Figura 60 .Xbee Shield + Arduino + Xbee receptor

Ahora debemos conectar la salida del detector de presencia a una de las entradas digitales de la placa Arduino.

Posteriormente conectaremos la salida PWM de nuestro controlador al driver que a su vez se conectará a la lámpara de LEDs.

7.2 Consecución de ciclo de trabajo

Una vez preparadas todas las conexiones físicas entre componentes del proyecto y configuradas todas las partes, procedemos a la comprobación de que todo funciona como deseamos y tenemos la respuesta esperada del circuito.

Previamente y para comprobar que, exactamente, tenemos el ciclo de trabajo que buscamos, conectaremos a la salida del pin 9 del Arduino (salida PWM configurada) a un osciloscopio para visualizar tanto la frecuencia que programamos para dicha salida como el valor del ciclo de trabajo obtenido en cada momento.

Procederemos a conectar el módulo emisor del Xbee (a través de la placa de *Interface USB Board*, nombrada en apartados anteriores) a un ordenador mediante cable USB y abriremos el software XCTU para poder enviar la información al módulo receptor.

Alimentamos al Arduino a la red eléctrica con un transformador DC de 12V y este se encargará, por mediación de la placa Xbee Shield, de proporcionar la tensión necesaria al Xbee receptor para su funcionamiento.

Posteriormente entraremos en el modo de envío de datos del XCTU de la manera que se explicó en apartados anteriores del proyecto y mandamos el valor de intensidad deseado a través del Xbee emisor al Xbee receptor en el modo HEX del envío de paquete de datos ya que así está programado para el Arduino.

La salida PWM configurada para proporcionarnos el ciclo de trabajo está comprendida entre valores de 0 a 255 (divide de 0 a 5V en 255 partes), por lo que debemos enviar a través del Xbee emisor un valor comprendido entre dichos valores.

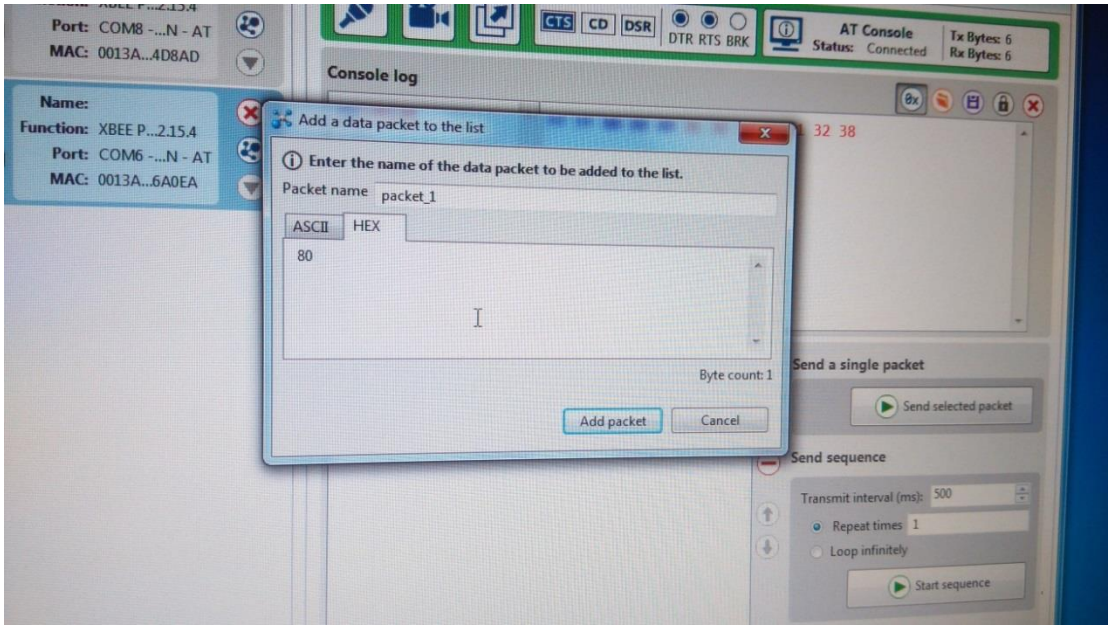


Figura 61 .Envío de número hexadecimal a Xbee receptor

De esta manera, si queremos obtener un ciclo de trabajo del 100% deberemos enviar a través del Xbee emisor “FF” que equivale al valor “255” en decimal y que corresponde con el 100% del ciclo de trabajo.

De la misma forma podemos probar con el 50% del ciclo de trabajo cuyo valor decimal sería “128”.Para hacernos con él enviamos el valor “80”.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Tabla 4.Equivalencia hexadecimal-decimal

Podemos observar en el osciloscopio como la frecuencia en la salida PWM del Arduino es de 25KHz tal y como habíamos programado y la consecución de los ciclos de trabajo deseados dependiendo del valor enviado a través del Xbee emisor.

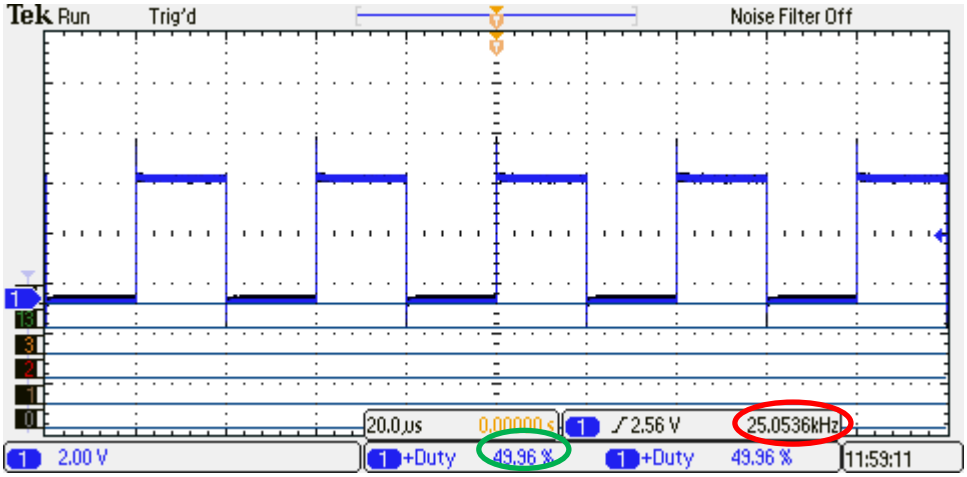


Figura 62 .Consecución de valores deseados (frecuencia y ciclo de trabajo)

7.3 Funcionamiento autónomo

Ya tenemos el ciclo de trabajo que nos permitirá controlar la intensidad de luz que emitirá nuestro alumbrado, pero ahora debemos asegurarnos de que nuestro sistema es capaz de funcionar de manera autónoma.

Para ello habíamos conectado un detector de presencia de doble tecnología en la entrada digital 2 del Arduino. Esta entrada será la encargada de gobernar los cambios de señal digital enviados por el sensor a nuestra placa y otorgarle su intensidad a la lámpara de LEDs.

Como hemos visto anteriormente en la programación, reservamos el valor "1" del ciclo de trabajo que podemos manejar con el fin de que cuando introduzcamos ese valor en nuestro ordenador y se lo enviemos de manera inalámbrica a la placa automáticamente lea la entrada digital del Arduino donde está conectado el detector y de esta manera funcione autónomamente sin necesidad de que nosotros tengamos que controlar la intensidad de la luminaria.

De esta manera, cuando el detector note presencia humana enviará una señal que el Arduino interpretará como un "1" digital y, gracias a la

programación cargada previamente en nuestro controlador, la salida PWM tomará su valor máximo, otorgando la intensidad máxima de luz a nuestra lámpara de LEDs.

Si por el contrario el detector de presencia no enviase ninguna señal, el Arduino tendrá un valor digital de “0” por lo que la salida PWM de nuestra placa sería 0 y no tendríamos iluminación saliente de la lámpara de LEDs.

De esta manera conseguimos un funcionamiento totalmente autónomo, sin necesidad de control por nuestra parte, y llegaremos a conseguir un ahorro energético de alrededor del 15%.

7.4 Diseño del PCB

Ya tenemos comprobado en nuestro laboratorio que el sistema funciona como nosotros queremos, por lo que ahora diseñaremos un circuito impreso de manera que utilicemos específicamente aquellos componentes y conexiones imprescindibles para el funcionamiento correcto de nuestro sistema y el ahorro de materiales innecesarios.

Para ello utilizaremos el programa “Fritzing” que contiene un apartado para diseñar PCBs teniendo además a nuestra disposición librerías con los componentes que nosotros utilizamos en nuestro proyecto.

Como se muestra en la ***Figura 63***, escogeremos los componentes necesarios para la realización de nuestro circuito impreso en un menú situado a la derecha del programa donde estarán todas las partes.

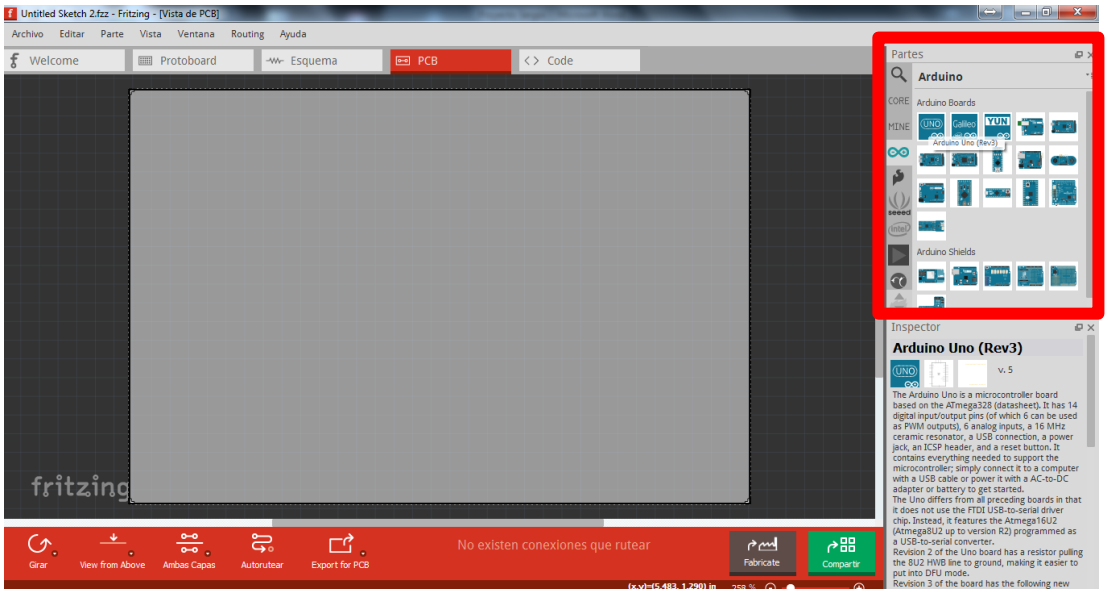


Figura 63.Selección de componentes en Fritzing

Sencillamente seleccionaremos el componente que necesitemos y lo arrastraremos a nuestro PCB.

En nuestro caso necesitaremos:

- Un Xbee Pro S1
- Un Atmega328
- Un Oscilador de Cristal de 16 MHz
- 2 Condensadores de 22pF
- 1 Regulador de tensión de 5V.
- 1 Regulador de tensión de 3,3V.

Los situaremos todos en nuestra placa y realizaremos las pistas de unión pulsando en un punto de conexión y dejando pulsado llevarlo hasta el contacto que nosotros queramos unir eléctricamente.

Con el fin de que ninguna pista interfiera con otra creando conexiones erróneas para nuestro circuito, debemos buscar la forma de evitar que las pistas se crucen entre sí.

Una vez diseñado, nuestro circuito final quedaría de la siguiente manera:

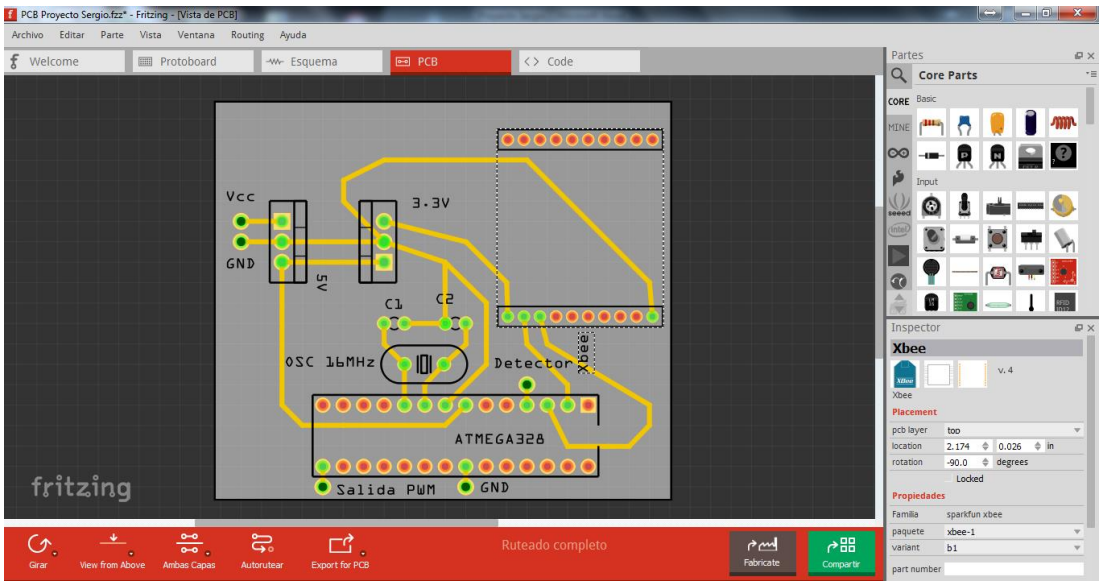


Figura 64.Circuito PCB completo en Fritzing

PLIEGO DE CONDICIONES

PLIEGO DE CONDICIONES

1. Especificaciones de diseño

Los dispositivos deben cumplir las siguientes especificaciones:

- Conectar el driver a una red eléctrica de 230V 50Hz
- Alimentar detector de presencia con adaptador de corriente continua de 12V.
- Utilizar un adaptador de corriente continua de entre 12V y 7V para la placa de control.

2. Revisión de precios

El precio del prototipo depende del coste de sus componentes en el momento de su compra, además del coste de la mano de obra y de la tecnología empleada en su diseño, producción y verificación.

El presupuesto general total que supone la redacción y ejecución del proyecto puede sufrir un incremento mensual que se elevará aplicando a dicho presupuesto la tasa internacional de inflación y el índice de precios al consumo que se establezca en el mes transcurrido.

3. Garantía

El periodo de garantía se estipulará en el contrato y empezará a contar desde la fecha de aprobación del acta de recepción. Hasta que tenga lugar la recepción definitiva, el Ingeniero Técnico responsable del correcto montaje y funcionamiento, tendrá a su cuenta y riesgo las reparaciones por defecto de ejecución o mala calidad de los materiales.

Durante este periodo, el Ingeniero Técnico garantizará al contratista contra toda reclamación de terceros fundada en causa y por ocasión de la fabricación del producto.

El Ingeniero Técnico no se hará responsable de los desperfectos ocasionados por el mal uso.

4. Normativa

La realización del proyecto se regirá, además de lo descrito anteriormente en el pliego de condiciones, por lo especificado en las siguientes normas de construcción:

- Se realizará el montaje del dispositivo con los elementos descritos en el proyecto, ya que con ellos se garantiza su calidad y correcto funcionamiento.
- Se aplicará el Reglamento Electrotécnico para Baja Tensión, aprobado por el Decreto 2413/1973,el 20 de Septiembre, así como las Instrucciones Complementarias a este reglamento, aprobadas por la Orden del 31 de Octubre de 1973 y las modificaciones posteriores.
- Se aplicará el Reglamento de Eficiencia Energética en Instalaciones de Alumbrado Exterior contemplados en el Real Decreto 1890/2008 del 14 de Noviembre del Ministerio de Industria y Trabajo.
- Se aplicará también la Ordenanza General de Seguridad e Higiene en el Trabajo aprobada por la Orden Ministerial del 9 de Marzo de 1971,del Ministerio de Trabajo.
- Se aplicará el Reglamento General de Contratación, según decreto 3410/75 del 25 de Noviembre.
- Norma tecnológica de la edificación NTE-IEB/1974 Instalaciones de Baja Tensión aparecida en el BOE del 20 de Abril de 1974.
- Orden general de Seguridad e Higiene en el trabajo aprobada por orden ministerial el 11 de Marzo de 1971, del Ministerio de Trabajo.
- Las normas UNE, DIN y las propias de la empresa, también son de aplicación.

5. Uso de los dispositivos

- La temperatura de funcionamiento de los dispositivos está entre 0º y 50º.
- No poner en contacto con agua o lugares de alta humedad.
- Conectar entradas y salidas conectores especificados.
- No manipular las tapas que cierran las carcasas donde se encuentran las placas de los dispositivos.

PRESUPUESTO

PRESUPUESTO

1. Introducción

En este apartado llevaremos a cabo la elaboración de un presupuesto para la fabricación de 1000 unidades de nuestro prototipo en una línea de fabricación.

2. Coste de mano de obra directa

Para poder fabricar nuestro prototipo necesitaremos un oficial de primera, un oficial de segunda y un oficial de tercera, que desempeñarán los siguientes cometidos:

El oficial de primera será el responsable de la supervisión de los trabajos y del control y suministro del material necesario para la fabricación y montaje de las placas.

El oficial de segunda se encargará de la fabricación de las PCBs y de realizar el control de calidad antes de que se monten los componentes sobre estas.

El oficial de tercera se encargará del montaje de los componentes y de comprobar el funcionamiento del producto final.

El coste de la mano de obra depende de la categoría del oficial y está regulado por el convenio del sector de Industria Siderometalúrgica. En el caso del convenido de Cantabria para este año los costes se mueven en torno a:

Oficial de primera: 6,71 Euros/h

Oficial de segunda: 6,19 Euros/h

Oficial de tercera: 5,73 Euros/h

El tiempo necesario para la fabricación de las 1000 unidades, para cada uno de los trabajadores lo podemos situar entorno a las siguientes horas:

Oficial de primera: 70 horas

Oficial de segunda: 330 horas

Oficial de tercera: 500 horas

Con lo que el coste de la mano de obra es el siguiente:

Oficial de primera: 6,71 Euros/h * 70h = 469,7 Euros

Oficial de segunda: 6,19 Euros/h * 250h = 1547,5 Euros

Oficial de tercera: 5,73 Euros/h * 500h = 2865 Euros

Coste de mano de obra directa = 469,7 Euros + 1547,5 Euros + 2865 Euros = 4882,2 Euros

Coste total de la mano de obra directa: 4882,2 Euros

3. Coste de materiales

A continuación se mostrará el precio de los materiales necesarios para el montaje de nuestro prototipo.

COMPONENTE	**UNIDADES**	**PRECIO UNITARIO**	**PRECIO TOTAL**
Placa de cobre virgen	1	1,02 €	1,02 €
Condensador 22pF	2	0,40 €	0,80 €
Oscilador Cristal 16MHz	1	1,80 €	1,80 €
XBP24-DKS Kit de desarrollo Xbee Pro	1	190,48 €	190,48 €
ATMEGA328	1	6,60 €	6,60 €
Driver para lámpara LEDs	1	80 €	80 €
Lámpara LEDs 4 matrices	1	102 €	102 €
Regulador de tensión a 5V	1	1,60 €	1,60 €
Regulador de tensión a 3,3V	1	1,70 €	1,70 €
Detector de presencia de doble tecnología BOSCH OD850-F1	1	103,60 €	103,60 €
Caja plástico	1	7,50 €	7,50 €
TOTAL			**497,10 €**

Por lo tanto, el coste total de la materia prima para una unidad de nuestro prototipo completo es de 497,10 Euros. Lo que hace un total de 497.100 Euros para la fabricación de las 1000 unidades.

Coste final de materiales : 497.100 Euros

4. Puesto de trabajo

Para realizar el prototipo hemos utilizado diferente maquinaria y herramientas, para poder producir nuestras unidades. Se especifican a continuación:

DESCRIPCIÓN	**PRECIO**
Fresadora CNC PCB	8.680 €
Ordenador a 2,4 GHz	319 €
Osciloscopio TDS724A	1.400 €
Arduino UNO Rev3	24,95 €
Estación de Soldadura AOUYUE 2702	277,22 €
Puntas y repuestos del soldador	150 €
Puntas y repuestos de la fresadora	320 €
Cubeta para preparación PCB	338,20 €
Rack para PCB	31,51 €
Insoladora UV Doble Capa	980 €
Tanque triple tratamiento	744 €
Aplicaciones y herramientas PCB	198,20 €
TOTAL	**13.463,08 €**

Calcularemos los gastos de amortización de los aparatos y equipos utilizados, sabiendo que su coste de instalación es de 13.463,08 Euros. Teniendo en cuenta un período de amortización de 6 años:

Coste amortización anual = 13.463,08 / 6 años = 2.243,85 Euros/año

Para calcular el coste de amortización por hora tomamos una jornada laboral de 8 horas diarias,5 días laborables a la semana, en 52 semanas al año, descontando 16 días festivos en Cantabria, nos salen 1.952 horas laborables al año.

Coste amortización por hora = 2.243,85 Euros / 1.952 horas = 1,15 Euros /h

Si para fabricar mil unidades necesitamos 900 horas, tenemos que el coste de amortización es de:

Coste de amortización = 1,15 Euros/h * 900 h = 1.034,56 Euros

Junto con el coste de amortización debemos tener en cuenta el mantenimiento de las máquinas, que estará compuesto por las horas empleadas en su realización y por el material utilizado.

La cantidad de horas empleadas en el mantenimiento de nuestra maquinaria se sitúa en torno a las 100 horas anuales, con un coste del material empleado de 300 euros.

Por lo que el coste de mantenimiento realizado por un oficial de segunda en un año es:

Coste mantenimiento anual = 6,19 Euros / h * 100 h = 619 €

Tomando las 1952 horas laborables de un año, tenemos un coste por hora de:

Coste de mantenimiento por hora = 619 Euros / 1952 h = 0,32 Euros/h

En las 100 horas que necesitamos, tendremos un coste de mantenimiento de:

Coste de mantenimiento = 0,32 Euros/h * 100 h = 32 Euros

Además debemos considerar el consumo de electricidad que ejercemos durante las 900 horas que empleamos para la fabricación de las 1000 unidades:

Coste electricidad consumida = (20 KW * 0,09 Euros-KW/h) * 900 h=

1.620 Euros

Por lo tanto el coste total del puesto de trabajo para la fabricación de 1000 unidades es de:

Coste de Mantenimiento = Coste de Mano de Obra + Coste del material

Coste total puesto de trabajo =

= Coste de Mantenimiento + Coste de amortización + Coste electricidad consumida

Coste total puesto de trabajo = 332 Euros + 1.034,56 Euros + 1.620 Euros

Coste total puesto de trabajo: 2.986,56 Euros

5. Coste de fabricación

Con los valores anteriormente obtenidos podemos calcular el coste de fabricación para 1000 unidades.

Coste de fabricación = Coste total de la mano de obra directa + Coste de la materia prima + Coste total puesto de trabajo

Coste de fabricación = 4.882,2 Euros + 497.100 Euros + 2.986,56 Euros

Coste de fabricación: 504.968,76 Euros

6. Coste en fábrica

En este tipo de proyectos de desarrollo de prototipos, lo que interesa es tener un presupuesto aproximado que nos permita tener una idea del precio de venta que puede alcanzar nuestro producto. Lo que se suele hacer en estos casos es utilizar aproximaciones de los costes de MOD y MOI anuales de años anteriores de la empresa. Por lo tanto, para obtener el coste en fábrica debemos tener en cuenta lo siguiente:

Debemos tener en cuenta el trabajo en el diseño del prototipo. Este ha sido realizado por un Ingeniero Técnico Industrial. El tiempo empleado en el diseño podemos situarlo en 60 horas a 9,78Euros / h, lo que hace un total de:

Mano de obra indirecta del diseño = 586,8 Euros

Una vez calculados los costes de MOI correspondientes al diseño, debemos obtener los costes de la mano de obra indirecta relativos a las tareas administrativas, etc. Estas se evalúan en un 38% del coste de la mano de obra directa del proyecto:

Coste mano de obra indirecta empresa = 0,38 * 4.882,2 = 1.855,24 Euros

Para obtener el coste total de la mano de obra indirecta, debemos sumar el coste de la MOI relativa al diseño más el del MOI de la empresa:

Coste total de la MOI = 586,8 Euros + 1855,24 Euros = 2442,04 Euros

Además debemos añadir unos gastos generales que se estiman en un 16% del coste final de la mano de obra directa:

Gastos generales = 0,16 * MOD = 781,152 Euros

Como gastos sociales tomamos un 7% del total de la mano de obra:

Gastos sociales = 0,07 * (MOD+MOI) = 512,7 Euros

Ahora podemos calcular el coste en fábrica para 1000 unidades:

Coste en fábrica = Coste de fabricación + MOI + GG + GS

Coste en fábrica = 504.968,76 € + 2442,04 € + 781,152 € + 512,7 €

Coste en fábrica: 508.704,649 Euros

7. Coste total

El coste total se define como:

Coste total = Coste total en fábrica + Gastos comerciales + Beneficio

Si tomamos como gastos comerciales el 2% del coste en fábrica:

Gastos comerciales = 10.174,09 Euros

Si deseamos obtener un beneficio del 6%:

Beneficio = 30.522,28 Euros

Tenemos un total de:

Coste total: 549.401,018 Euros

8. Precio de venta del producto

Para hallar el presupuesto de fabricación final, se aplica al coste total un 16% de I.V.A.:

Presupuesto facturable = Coste total + 0,16 * Coste total

Presupuesto facturable = 637.305,18 Euros

Por último calculamos el precio de venta al público que se define de la siguiente forma:

Precio de venta al público = Presupuesto facturable / nº de unidades

Precio de venta al público = 637.305.18 Euros / 1000

Precio Venta al Público = 637,31 Euros

BIBLIOGRAFÍA

- http://www.idae.es
- http://www.olimex.cl
- http://www.ledsday.com

- http://www.instalacionesyeficienciaenergetica.com
- http://www.digitalavmagazine.com
- http://www.onoff-av.es
- http://www.cogaprel.com
- http://www.digikey.com
- http://forum.arduino.cc/index.php
- http://www.theben.es
- http://www.faciled.es
- http://www.andresduarte.com
- http://tutorialfritzing.blogspot.com.es
- http://www.atmel.com
- http://www.digi.com/
- http://www.emb.cl
- http://www.ambientum.com
- https://code.google.com/p/arduino-pwm-frequency-library/downloads/list (librería PWM Arduino)