



INSTITUTO POLITECNICO NACIONAL

**ESCUELA SUPERIOR DE INGENIERÍA MECANICA Y
ELECTRICA**

UNIDAD PROFESIONAL “ADOLFO LOPEZ MATEOS”

INGENIERIA EN COMUNICACIONES Y ELECTRONICA

TESIS

**“Sistema de domótica controlado por voz para personas
con deficiencias motrices”**

**QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMUNICACIONES Y ELECTRONICA**

PRESENTAN:

**Vidkar Anibal Delgado Gallardo
Alejandro Trujillo Castellanos
Daniel Valdez Simón**

ASESORES:

**M. en C. Eduardo Gabriel Balderas
M. en C. Pedro Gustavo Magaña del Río**



MEXICO, D.F. 2014

INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
UNIDAD PROFESIONAL "ADOLFO LÓPEZ MATEOS"

TEMA DE TESIS

QUE PARA OBTENER EL TÍTULO DE
POR LA OPCIÓN DE TITULACIÓN
DEBERÁ (N) DESARROLLAR

INGENIERO EN COMUNICACIONES Y ELECTRÓNICA
TESIS COLECTIVA Y EXAMEN ORAL INDIVIDUAL
C. VIDKAR ANIBAL DELGADO GALLARDO
C. ALEJANDRO TRUJILLO CASTELLANOS
C. DANIEL VALDEZ SIMON

**"SISTEMA DE DOMÓTICA CONTROLADO POR VOZ PARA PERSONAS CON DEFICIENCIAS
MOTRICES"**

DISEÑAR E IMPLEMENTAR UN SISTEMA DE COMUNICACIÓN INALÁMBRICO POR MEDIO DE
COMANDOS DE VOZ QUE PERMITA MANEJAR 3 ELEMENTOS EN UNA CASA HABITACIÓN.

- INTRODUCCIÓN
- ANTECEDENTES
- DOMÓTICA
- ANÁLISIS Y DISEÑO DEL SISTEMA
- PRUEBAS DEL SISTEMA Y RESULTADOS
- CONCLUSIONES
- RECOMENDACIONES
- BIBLIOGRAFÍA
- GLOSARIO
- ANEXOS

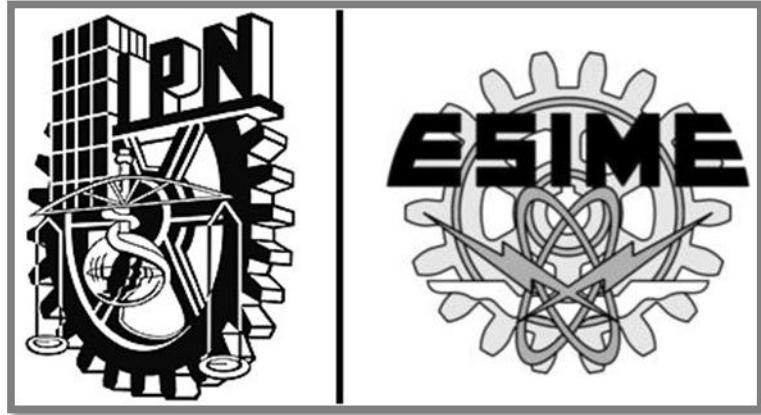
MÉXICO D.F. A 29 DE MAYO DE 2014

ASESORES

M. EN C. EDUARDO GABRIEL BALDERAS

ING. PEDRO GUSTAVO MAGAÑA DEL RIO


ING. PATRICIA LORENA RAMIREZ RANGEL
JEFE DEL DEPARTAMENTO DE
INGENIERÍA EN COMUNICACIONES Y ELECTRONICA



Agradecimientos

Agradecimientos

Desde el comienzo de nuestra carrera no teníamos contemplado aún todo lo que viviríamos dentro de nuestra apreciada escuela, sin embargo, el cursar cada uno de los semestres nos fue dando las herramientas necesarias para poder enfrentarnos al mundo de afuera. Estos largos años, como los veíamos al principio, han culminado con la preparación de profesionistas de bien, es ahora cuando nos toca a nosotros regresarle un poco a nuestra escuela por todo aquello que nos brindó. También aprendimos muchas cosas dentro de ella, a aceptar puntos de vista, a reconocer triunfos ajenos, aprendimos a escucharnos, a ser más responsables y a dar nuestro mejor esfuerzo en cada trabajo realizado, también aprendimos a ser más pacientes y creamos las mejores amistades dentro de ella. Sin lugar a dudas, haber sido parte de tan prestigiosa escuela nos hace sentir orgullosos, no solo por los logros obtenidos, si no por haber formado parte de ella. Gracias ESIME ZACATENCO. Gracias IPN.

También queremos agradecer a nuestros profesores, quienes han sido siempre un ejemplo a seguir, por los conocimientos, experiencias y enseñanzas que han compartido con nosotros a lo largo de estos años, y por la amistad que nos han brindado como personas.

Al profesor Eduardo Gabriel Balderas por habernos llevado por un camino responsable y que a pesar de los regaños recibidos, siempre tomamos sus críticas para mejorar nuestro trabajo.

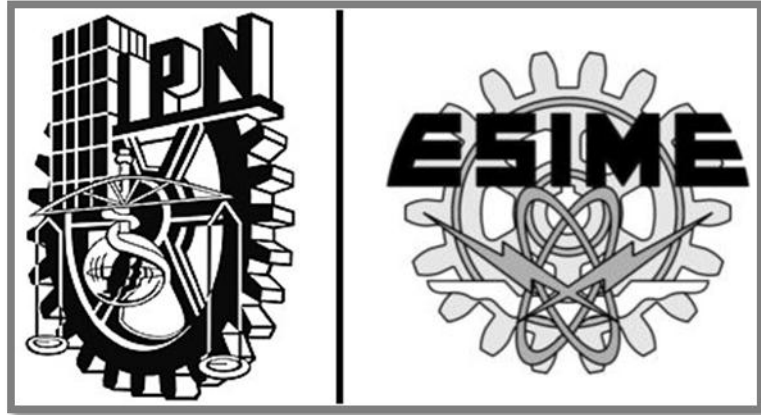
Al profesor Jaime Pedro Abarca Reyna por habernos motivado a trabajar arduamente, por habernos enseñado que el valor de las cosas radica en lo que obtenemos al final, no en los límites que nos ponemos.

A la profesora Nancy, sin sus numerosos regaños no nos habríamos exigido lo suficiente para realizar un buen trabajo, al final valió la pena.

Al profesor Pedro Gustavo, sin su guía y asesoramiento no lo habríamos logrado, aun cuando se nos juntó el trabajo siempre tuvo tiempo para atendernos y darnos sus puntos de vista para que nosotros lográramos entregar un buen reporte en tiempo y forma, también por sus consejos y motivaciones para entregar una investigación de calidad.

Gracias a nuestras metas hemos sido afortunados de ser parte de un gran logro, una buena institución y profesores de calidad nos han permitido desprendernos de limitaciones, para poder anhelar siempre ser mejores. El aprendizaje y las numerosas lecciones que vivimos fueron duras, pero sabíamos que tendríamos que esforzarnos al máximo para lograr un buen resultado.

A nuestros amigos y todas las personas que con su apoyo nos dieron la motivación para seguir adelante, muchas gracias.



Dedicatorias

Dedicatorias

A mi madre Ma. Dolores por ese apoyo incondicional, por esa confianza, sacrificio y por siempre estar a mi lado. Por ese amor que siempre me ha dado la fortaleza para seguir adelante.

A mi abuela Dolores por ser el pilar de mi familia y por brindarme una oportunidad de estudiar una carrera profesional.

A mis hermanos Antonio y Lucero, que siempre me han guiado por el buen camino y me han dado consejos para tomar buenas decisiones.

A mi madrina Julia, por todas sus muestras de cariño y por educarme desde pequeño para ser una persona de bien.

Vidkar

Dedicatorias

A mis padres:

Gracias por su apoyo. Por cada segundo que me brindaron su confianza y cariño, por cada paso que avanzaron junto a mí y me ayudaron a no ceder ante las adversidades. A ustedes dedico este trabajo, en él plasmo los conocimientos que con arduo esfuerzo fui construyendo con el paso de los años en la que hoy es mi alma mater. Gracias por estar conmigo siempre.

A mis hermanos:

Gracias por todo el apoyo recibido, aunque pasamos grandes retos siempre estuvieron presentes brindándome su apoyo incondicional. Gracias por todas sus palabras de aliento y muestras de afecto cuando más necesite su ayuda, sé que siempre serán mi pilar en la vida.

A mis amigos:

Si estoy aquí es gracias a ustedes, porque me ayudaron, me apoyaron y me dieron los mejores años de mi vida. Sus consejos siempre guiarán mi camino. Gracias por cada recuerdo que con gusto he atesorado en mi corazón, siempre los tendré presentes.

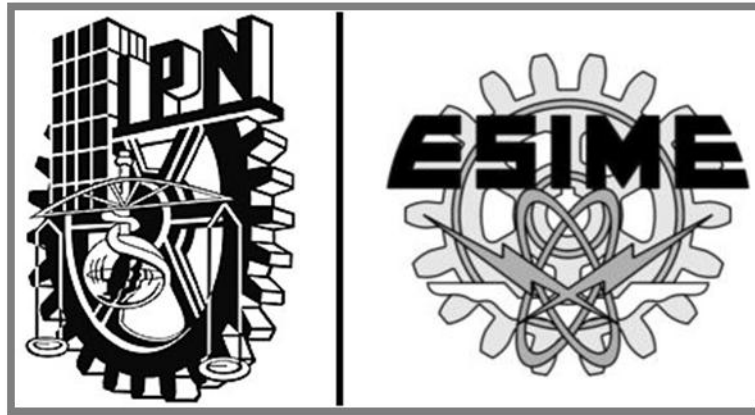
Daniel

Dedicatorias

A mi familia:

Agradezco su apoyo y consejos. A mis hermanos que siempre estuvieron ahí en las buenas y en las malas, a todos ellos que me brindaron la mano cuando más la necesitaba y su apoyo incondicional. Sobre todas las cosas le agradezco a aquella mujer que nunca dudó de mí, que fue cómplice de mis momentos más importantes como los más necesarios, a aquella persona que se volvió mi amiga, mi cómplice y el soporte en mi vida, a mi madre que realizó de esta meta un sueño.

Alejandro



“Sistema de domótica controlado por voz para personas con deficiencias motrices”

OBJETIVO

“Diseñar e implementar un sistema de comunicación inalámbrico por medio de comandos de voz que permita manejar 3 elementos en una casa habitación”

Índice

Introducción	15
Antecedentes	17
Capítulo 1. Domótica	18
1.1 Introducción	19
1.2 El hogar inteligente	20
1.3 La red dentro del hogar inteligente	21
1.4 Dispositivos del hogar inteligente	23
1.5 La comunicación dentro del hogar inteligente	25
Capítulo 2. Análisis y diseño del sistema	27
2.1 Análisis del sistema	28
2.2 EasyVR Shield	32
2.2.1 Características	33
2.2.2 Micrófono	34
2.2.3 Protocolo de comunicación	34
2.2.4 EasyVR Commander	35
2.3 Módulo Xbee	36
2.3.1 Estándar IEEE 802.15.4	36
2.3.2 Características del Xbee	37
2.4 Manejo de timers y PWM en microcontroladores AVR de Atmel	39
2.4.1 Temporizadores	40
2.4.2 Partes de los timers	41
2.4.2.1 Registro TCNTX	41
2.4.2.2 Registro OCRXA y OCRXB	42
2.4.3 Modulación por ancho de pulso PWM	44
2.4.4 PWM en el Atmega 1248	44
2.4.5 PWM rápida	45
2.5 Diseño del sistema	47
2.5.1 Generador de comandos	48

2.5.2 Etapa de reconocimiento de voz	48
2.5.3 Comandos de voz	51
2.6 Transmisor y receptor inalámbrico	53
2.7 Etapa de control	55
2.8 Etapa de potencia	58
2.8.1 Circuito PWM	58
Capítulo 3. Pruebas del sistema y resultados	60
3.1 Introducción	61
3.2 Prueba en la etapa de reconocimiento de voz	61
3.3 Pruebas de transmisión y recepción inalámbrica	65
3.4 Pruebas de la etapa de control	67
3.4.1 Modo de inicialización	70
3.4.2 Encendido, apagado y regulación del ventilador	71
3.4.2.1 Modo “enciende ventilador”	71
3.4.2.2 Modo “regula ventilador”	71
3.4.2.3 Modo “apaga ventilador”	71
3.4.3 Encendido, apagado, regulación y temporizador del cuarto	72
3.4.3.1 Modo “regula cuarto”	72
3.4.3.2 Modo “enciende cuarto”	72
3.4.3.3 Modo “apaga cuarto”	73
3.4.3.4 Modo “cuarto en 30 segundos”	73
3.4.4 Encendido, apagado y regulación de la sala	73
3.4.4.1 Modo “enciende sala” y “apaga sala”	73
3.5 Pruebas en la etapa de potencia	74
3.6 Pruebas en un edificio y en una casa	75
Conclusiones	79
Recomendaciones	80
Bibliografía	82
Glosario	85
Anexos	88



Introducción

INTRODUCCIÓN

Los hogares han evolucionado conforme ha pasado el tiempo, desde aquellos tiempos de grutas y cuevas utilizadas como un hogar, hasta las maravillosas casas que además de resguardarnos de la intemperie ofrecen ventajas tecnológicas.

Este espacio del hogar toma unas determinadas formas en función de la tecnología disponible en cada época. Es por ello que se adopta una herramienta para lograr comodidad y brindar seguridad dentro de nuestro hogar. La domótica consiste en introducir la tecnología y los elementos desarrollados por ella para los hogares y así mejorar la calidad de vida de sus habitantes y ampliar sus posibilidades de comunicación automatizando procesos domésticos. La domótica puede automatizar un simple proceso para encender o apagar un foco o un aparato a una hora determinada, hasta los más complejos sistemas capaces de interactuar con cualquier elemento eléctrico de la casa, e intercomunicando tanto estos procesos con los residentes del hogar entre sí y con el exterior.

Un aspecto adicional a considerar es el dotar de independencia funcional a los usuarios de estas casas inteligentes con elementos que les permitan emitir sus órdenes a los dispositivos que se desee controlar sin estar en la cercanía de estos; para ello se deben considerar elementos inalámbricos.



Antecedentes

ANTECEDENTES

Los avances tecnológicos marcan el ritmo de la vida del hombre y sus hogares no están exentos de ellos. Y uno de los avances que más ha influido en el hogar es la electricidad.

La electricidad es el punto clave que ha permitido elevar el confort en los hogares, ya que da paso a los electrodomésticos tales como: la lavadora, horno de microondas, licuadora, refrigerador, etc., los cuales simplifican tareas de la vida cotidiana y ayudan a hacer cómoda la vida del hombre en el hogar.

Estos electrodomésticos no serían capaces de realizar sus funciones sin la ayuda de otro avance: la electrónica, la cual permite que a través del diseño de circuitos electrónicos o programación de los mismos, se pueda regular cada proceso de los electrodomésticos, como regular la potencia de un microondas y una licuadora o elegir un modo de lavado como rápido o delicado.

El siguiente avance es la domótica, la cual es capaz de integrar y controlar todos los sistemas eléctricos y electrónicos del hogar a través de sensores como los de presencia, de temperatura, de nivel de luz, etc. y hacer que la casa actúe por sí sola ante los cambios detectados por los sensores, y que también sea capaz de interactuar con los usuarios a través de un celular, PC, LCD, etc.

En una casa existe un sin fin de elementos que pueden ser controlados pero ¿es posible diseñar un sistema domótico para una persona con deficiencias motrices, de tal forma que desde cualquier lugar de su casa pueda controlar los elementos que desee? Ese es el problema a resolver.



CAPÍTULO 1 | “Domótica”

1.1 Introducción

Durante la década de los 80 y a finales de ésta, surgieron en Estados Unidos, las primeras ideas para la automatización de los aparatos eléctricos y electrónicos en un hogar. Durante los finales de esa década se crearon diferentes métodos para poder lograr ese propósito, para ello se implementó la primera tecnología llamada X-10, ésta consistía en controlar diferentes equipos del hogar a través del envío de señales en formato digital, con la característica de que los dispositivos controlados utilizaran la línea eléctrica. Algunos de los primeros sistemas que utilizaron esta tecnología fueron instalados en las oficinas de edificios, en su primera etapa sólo controlaban el aire acondicionado.

Años después se comenzó a utilizar el sistema de cableado estructurado, el cual facilitaba la conexión de diferentes dispositivos de control y de seguridad, fue por ello que con este sistema se logró controlar un mayor número de elementos. Una vez que los edificios comenzaron a utilizar los nuevos sistemas para la automatización de diferentes tareas, se les comenzó a llamar edificios inteligentes.

De la misma manera que sucedió con los edificios, se comenzó a buscar la manera de facilitar las tareas dentro de un hogar. Para ello se empezaron a enumerar las primeras necesidades que se cubrirían utilizando estas tecnologías para satisfacerlas, esto dio origen a la vivienda domótica. La domótica consiste en la automatización de diferentes aparatos eléctricos y electrónicos a través de diferentes instrucciones que pueden activarse automáticamente o al ser ingresadas por algún otro elemento que pueda interactuar con el sistema.

Una vivienda domótica es aquella que integra los elementos eléctricos y electrónicos con el propósito de asegurar un mejor confort y calidad de vida para los usuarios. La vivienda domótica es capaz de interactuar con los diferentes

elementos que se comunican dentro de ella, además de brindar mayor seguridad, ahorro de energía y aprovechamiento al máximo de todos sus recursos, brindando así una mínima intervención por parte del usuario.

1.2 El hogar inteligente

Algunas ventajas de vivir dentro de un hogar inteligente son:

- ✓ Encendido y apagado de los elementos eléctricos y electrónicos (aire acondicionado, alarmas, etc.).
- ✓ Control del tiempo de encendido de los aparatos eléctricos y electrónicos (luces, calefacción, etc.).
- ✓ Aviso de una intrusión o averías dentro de la casa (alarmas).
- ✓ Control de acceso a la vivienda.
- ✓ Control de los elementos hidráulicos del hogar (bomba de agua).

En la figura 1.1 se muestran algunas de las principales variables que permiten ser controladas dentro del hogar inteligente. Algunas de ellas son seguridad, comodidad, ahorro de energía, comunicación y ocio, éstas pueden ser controladas a través de instrucciones o por medio de redes de comunicaciones, por ejemplo: bluetooth, wi-fi, etc.



FIGURA 1.1 Variables controlables en un hogar inteligente.

1.3 La red dentro del hogar inteligente

Para que los elementos del hogar puedan interactuar entre sí y con el exterior se utiliza una red domótica, ésta permite controlar la automatización de actividades cotidianas, ya sea mediante un control remoto o a través de otros elementos dentro de la vivienda.

Cuando es posible controlar los elementos dentro de la vivienda a través de un sistema de control se reduce el consumo de energía dentro del hogar digital, también se incrementa el confort y mejora la calidad de vida de los usuarios. Es por ello que la red que controla todos estos elementos no debe interferir con señales ajenas. Una red de uso no comercial es la que suele usarse en este tipo de aplicaciones, esto se debe a la gran capacidad de adaptabilidad por parte de

las necesidades del sistema al cual se vaya a controlar, de lo contrario tendría que usarse otro tipo de red.

Para el caso del hogar inteligente suele usarse una red WPAN (Red de área personal), ya que permite una cobertura personal en donde pueden interactuar diferentes tecnologías basadas en radiofrecuencia y en el estándar HOME RF, el cual utiliza un aparato central para recibir todas las señales, por ejemplo: el bluetooth (que sigue la especificación IEEE 802.15.1), ZigBee (basado en la especificación IEEE 802.15.4) y RFID (sistema remoto de almacenamiento y recuperación de datos, que permite transmitir la identidad de un objeto mediante ondas de radio).

Aunque el alcance típico de este tipo de redes es de algunos metros, mucho depende del tipo de tecnología que esté utilizando, pero la principal finalidad de estas redes es comunicar cualquier dispositivo que ocupe algunas de las tecnologías descritas anteriormente.

Una vez que la red domótica está en funcionamiento, suelen presentarse diferentes servicios dentro de ella: los servicios del hogar inteligente, los servicios de datos y los servicios multimedia.

En la figura 1.2 se muestran algunos servicios proporcionados por la red domótica y su conexión con el exterior a través de un enrutador o aparato central que recibe todas las instrucciones de los elementos del interior, éste se encarga de interpretar todas las instrucciones con las cuales se comunican los elementos del interior para así ejecutar las tareas que son programadas por un usuario dentro de la vivienda.

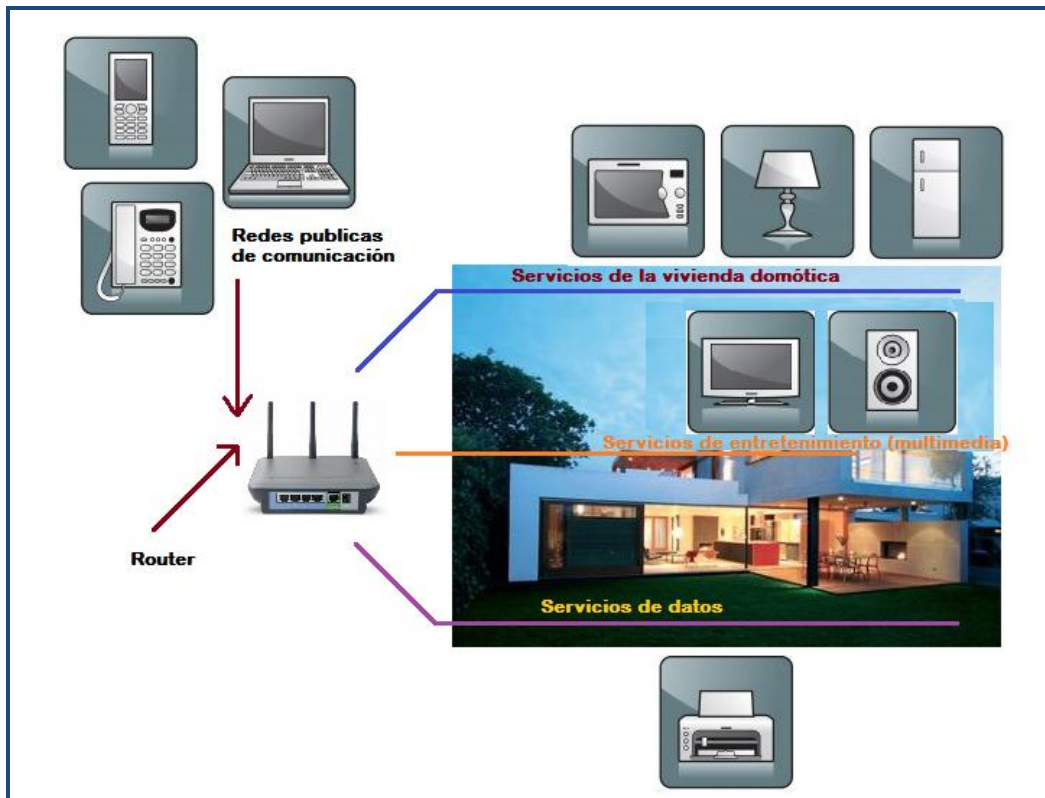


FIGURA 1.2 Comunicación de los servicios internos con el exterior.

1.4 Dispositivos del hogar inteligente

Los dispositivos que se deben instalar para automatizar y controlar los elementos del hogar inteligente son:

- ✓ El Enrutador es el dispositivo que interconecta los servicios de los elementos del interior con el exterior, permitiendo así el control local y remoto de los diferentes dispositivos utilizados en la automatización del hogar inteligente.
- ✓ El sistema de control es el conjunto de elementos que se encarga de controlar todos los dispositivos que intervienen en la automatización del hogar digital.

- ✓ Los sensores son los elementos encargados de reunir la información de las diferentes variables utilizadas por el sistema de control (el control de temperatura, los detectores de presencia, etc.), su finalidad es reunir los datos suficientes para enviárselos al enrutador y que así ejecute las tareas programadas. Existen de diferentes tipos: infrarrojos, de contacto, de presencia, de microondas y de vibración.
- ✓ Los actuadores son los dispositivos empleados por el sistema de control, sirven para modificar el estado de los diferentes elementos o de las instalaciones en el hogar (el corte o suministro de agua o gas, el envío de una alarma en caso de intrusión a una central de seguridad, avisar a una línea de emergencia, cerrar puertas, etc.).

En la figura 1.3 se muestran los diferentes elementos que pueden interactuar entre sí dentro del hogar digital, también se muestran algunas de las principales aplicaciones que pueden activarse siendo programadas, por ejemplo: que se encienda el aire acondicionado a una determinada hora, que la lavadora comience a lavar la ropa en cuanto el usuario cierre la tapa, que los servicios de wi-fi y cable se activen al notar la presencia del usuario, etc.

Por ejemplo, una persona de la tercera edad, puede necesitar un monitoreo constante del estado de su salud, para ello necesitaría acudir a un hospital, programar una previa cita con el médico que lo atenderá, si necesita el traslado al hospital necesitaría pedir un taxi, etc. Para ello podría hacer uso de un sistema inteligente que le ayudara a realizar todo en cuanto un sensor detectara que su salud está mal, con ayuda del sistema el usuario podría obtener un diagnóstico previo sin la necesidad de acudir directamente al hospital, lo cual le permitiría saber en qué momento realmente lo necesitaría y así llamar a una ambulancia.



FIGURA 1.3 Automatización de los diferentes servicios dentro del hogar inteligente.

1.5 La comunicación dentro del hogar inteligente

Para poder enlazar entre sí los diferentes elementos eléctricos y electrónicos del hogar inteligente se usa el enrutador, éste se encargará de coleccionar la información recibida por los sensores y a través de los actuadores ejecutará las tareas programadas.

Existen algunos electrodomésticos capaces de interpretar este tipo de comandos, una vez que son enviadas las diferentes instrucciones, el enrutador o módulo central puede ser capaz de ejecutar una acción u otra mientras una está activada, esto permite que todas las diferentes tareas sean realizadas sin que interfieran entre sí.

En la actualidad sólo algunas oficinas de edificios, universidades y hospitales suelen usar sistemas de automatización en la iluminación, lo cual puede controlarse a través de más elementos de vanguardia o a través de un módulo central que sea capaz de ejecutar acciones previamente programadas.

La figura 1.4 muestra algunas funciones que pueden automatizarse dentro del hogar inteligente, también se muestran los servicios de uso común que pueden encontrarse en la vivienda y que pueden ejecutarse con tareas previamente programadas.



FIGURA 1.4 Algunas funciones básicas a instalar en un hogar inteligente.



CAPÍTULO 2 | “Análisis y diseño del sistema”

2.1 Análisis del sistema

Se toma como punto inicial el mejorar la calidad de vida de una persona con problemas motrices, ya que pueden presentar dificultades para realizar actividades de la vida diaria como por ejemplo encender y apagar un foco de una habitación, ya que si la persona está en silla de ruedas, le resultará complicado acercarse hasta el lugar donde se encuentra el interruptor, y más si hay obstáculos de por medio, como una silla, un sillón, una puerta, etc., o tal vez el espacio de maniobra para moverse esta reducido, lo que puede traer como consecuencia algún accidente.

Para evitar esas dificultades, se decide que el sistema a diseñar sea inalámbrico, para que el usuario desde cualquier lugar de su casa tenga la facilidad de controlar elementos del hogar, y así evitar la necesidad de desplazarse para accionarlos, de tal forma que se le brinda confort y se mejora su seguridad.

Otro factor a tomar en cuenta es como el usuario va a controlar los elementos, ya que existen muchas alternativas, como un mando de control accionado por botones, una aplicación para un teléfono inteligente, comandos de voz, etc. Pero en la mayoría de estas opciones se tienen que utilizar las manos, y como el usuario final tiene problemas motrices, es posible que presente dificultad para usar sus manos, por lo tanto la opción más viable es usar comandos de voz, así mientras el usuario tenga la capacidad del habla podrá controlar lo que el quiera.

Entonces el proyecto debe tener 2 características esenciales: ser inalámbrico y controlado por voz.

En la figura 2.1 se muestra el diagrama de los bloques necesarios para cumplir con las características principales del proyecto, en el cual se muestra que los elementos a controlar son el foco de un cuarto y de una sala, así como un ventilador. El primer bloque es un sistema de reconocimiento de voz el cual se encarga de procesar y decidir cuál es el comando hablado por el usuario final para posteriormente enviarlo inalámbricamente al segundo bloque que es el sistema de control que se encarga de realizar la orden que emitió el usuario.

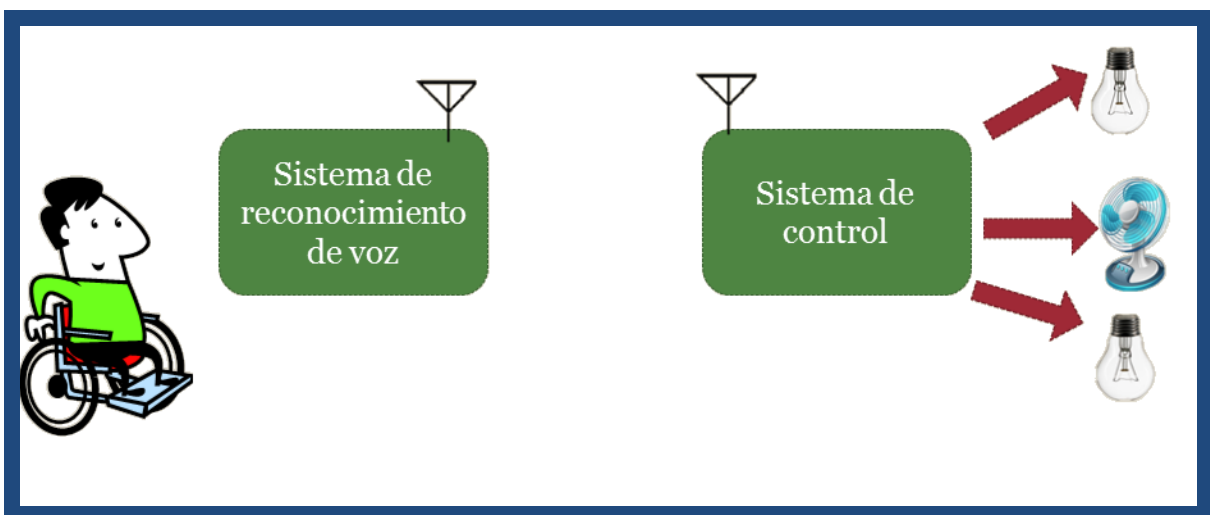


FIGURA 2.1 Estructura lógica del diseño.

Las características necesarias del bloque del sistema de reconocimiento de voz son:

- ✓ Reconocer las órdenes habladas por el usuario de forma eficiente.
- ✓ Generar comandos de control a partir del punto anterior.
- ✓ Tener un transmisor inalámbrico de bajo consumo ya que este sistema va a estar continuamente escuchando la voz del usuario.

Las características del bloque del sistema de control son:

- ✓ Tener un receptor inalámbrico de bajo consumo.
- ✓ Contar con los suficientes puertos de salida para controlar diversos elementos.
- ✓ Capacidad de mostrar a través de un display la orden que ha emitido el usuario.

Para cumplir con las características del primer bloque es posible utilizar un módulo de voz como el EasyVR Shield, el cual es capaz de reconocer comandos grabados previamente y a través de una **comunicación serial** enviar la información correspondiente del comando de voz escuchado.

Para procesar la información enviada del módulo de voz, es necesario tener un microcontrolador, como el atmega328, el cual es idóneo ya que se puede utilizar con la **plataforma** de hardware libre “**Arduino**” y también cuenta con una interfaz fácil de programar para que trabaje en conjunto con el módulo de voz.

Para el segundo bloque también es indispensable usar otro microcontrolador para que pueda procesar la información que reciba y tome una decisión acerca del control de los elementos, el microcontrolador atmega1284, se elige porque tiene las siguientes características que satisfacen las del sistema de control:

- ✓ 1 puerto para enviar información a un LCD, para mostrar al usuario los comandos que se están ejecutando.
- ✓ Al menos 5 Salidas PWM (Modulación por ancho de pulso) para controlar focos y ventiladores, esto se debe a que en una casa o departamento, por lo menos habrá un cuarto, una sala, un baño, una cocina y posiblemente aire acondicionado. Las salidas deben ser PWM ya que con este tipo de modulación se puede tener un control de potencia en los elementos a

controlar, y así por ejemplo regular la intensidad de luz de un foco o regular la velocidad de un ventilador.

- ✓ Entradas para convertidor analógico-digital, para la detección de fallos en los focos o en el ventilador.

Para la comunicación inalámbrica se emplean dos chips Xbee, uno es conectado al atmega328 para funcionar como transmisor, y el otro chip es conectado con el atmega1284 para trabajar como receptor.

La razón para emplear los chips Xbee, es porque proporcionan una **comunicación serial** que es la misma que se utiliza en el microcontrolador, lo único que se debe configurar es la tasa de transmisión de datos, el número de bits a enviar, el bit de paridad y el número de bits de parada. Además el alcance del Xbee es de 90 metros al aire libre y 30 metros en el interior de una casa. Tomando en cuenta que un departamento tiene por lo mínimo un área de 20 metros cuadrados y cuenta con un cuarto, sala, baño y cocina, se toma un aproximado de 30 metros cuadrados para el estudio, aunque estas medidas pueden cambiar por el diseño y/o alcances económicos que se tengan para adquirir una casa, entonces el alcance del chip Xbee es adecuado y suficiente para la transmisión inalámbrica del prototipo.

El funcionamiento propuesto para el sistema se basa en que cuando el usuario habla, el micrófono trabajando con el módulo de voz le alerta al microcontrolador que hay una nueva instrucción, el atmega328 la convierte en un comando para que pueda ser transmitido a través del chip Xbee, los comandos son caracteres tales como "a", "b", "c", "d", etc., que representan instrucciones como: encender el foco del cuarto, apagar el foco del cuarto, encender el ventilador, entre otras que serán explicadas en el apartado de diseño del prototipo.

El Xbee del atmega328 envía la señal correspondiente al Xbee receptor con la instrucción deseada, y la señal recibida es procesada por el microcontrolador atmega1284 para que controle los dispositivos de la casa habitación.

Para el control de los dispositivos de la casa, se configuran 2 timers del atmega1284, el timer0 y el timer2 para tener 4 salidas PWM, y con ellas se pueden implementar las funciones de los niveles de cambio, que son encender, apagar y regular la intensidad del foco o ventilador. Y la función del temporizador la realiza el timer0, de tal forma que el usuario mediante un comando de voz, pueda controlar el apagado automático de la luz o del ventilador, y en el timer se hace la cuenta para apagarlo. La detección de errores, es para que el usuario se percate de que alguno de sus dispositivos no funciona al momento de activarlos.

En general, estas funciones de control las puede realizar el microcontrolador simultáneamente.

2.2 EasyVR Shield

El módulo de reconocimiento de voz Easy VR Shield de la marca Veeear, la cual fabrica una línea de productos de alta calidad para equipar capacidades de reconocimiento de voz a casi cualquier proyecto, está diseñado para trabajar en conjunto con la plataforma Arduino. En la figura 2.2 se muestra el EasyVR Shield que se usa en este proyecto.

Este módulo de voz es capaz de comunicarse con cualquier microcontrolador, siempre y cuando el microcontrolador tenga una **interfaz UART**, por eso puede comunicarse con el microcontrolador atmega328, ya que es posible programar esta interfaz en él.

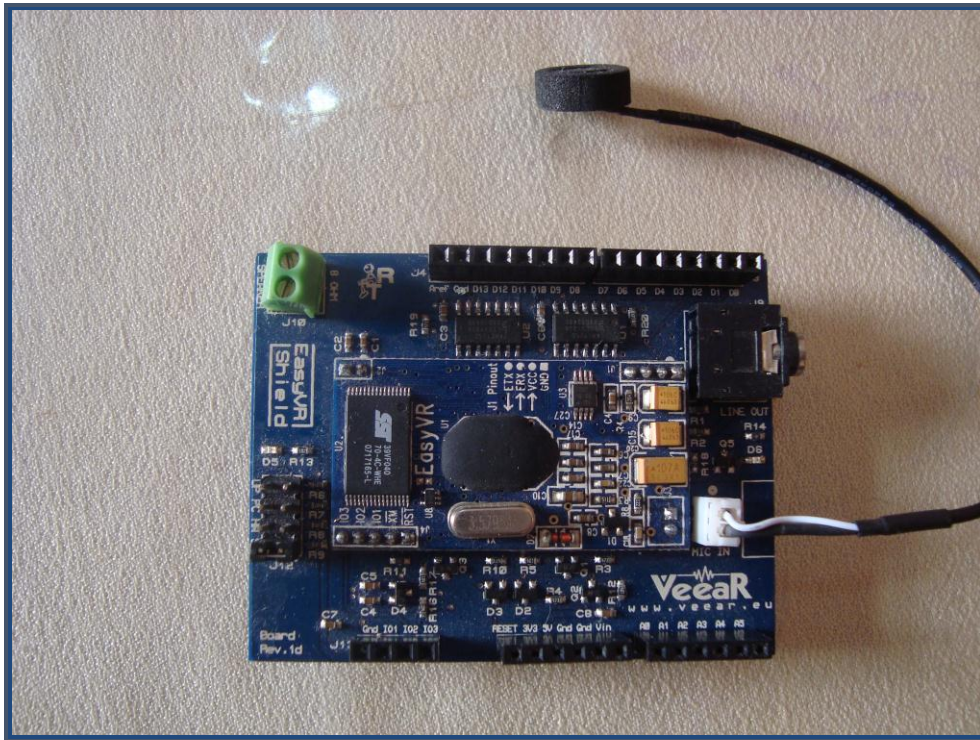


FIGURA 2.2 Placa EasyVR con micrófono.

2.2.1 Características

- ✓ Comandos SI (speaker independent) pregrabados y listos para ejecutar. Están en los siguientes idiomas:
 - Inglés.
 - Italiano.
 - Alemán.
 - Francés.
 - Español.
 - Japonés.
- ✓ Soporta hasta 32 comandos de activación definidos por el usuario (SD, speaker dependent) o también contraseñas de voz.

- ✓ Fácil de usar y simple interfaz gráfica de usuario para programar los comandos de voz y audio.
- ✓ El modulo puede ser utilizado con un microcontrolador que tenga interfaz UART (alimentado a 3.3 -5v).
- ✓ Función de reproducción de sonido.

2.2.2 Micrófono

El micrófono suministrado con el módulo EasyVR es un micrófono de condensador electret omnidireccional (Horn EM9745P-382) con las siguientes características:

- ✓ Sensitividad de 38 dB (0 dB = 1V/Pa @ 1 KHz).
- ✓ Impedancia de carga 2.2 K.
- ✓ 3V, Voltaje de funcionamiento.
- ✓ Respuesta en frecuencia casi plana en el intervalo de 100 Hz - 20 kHz.

2.2.3 Protocolo de comunicación

La comunicación con el módulo de EasyVR utiliza una interfaz UART estándar compatible con los niveles lógicos 3.3, 5v TTL/CMOS, de acuerdo a su tensión de Vcc suministrada.

Una conexión típica a un micro controlador es la que se muestra en la figura 2.3, donde se puede apreciar que únicamente se requiere que el microcontrolador cuente con la salida TX y entrada RX que deben ser programadas para una **comunicación serial**, y suministrar un voltaje en el intervalo de 3.3 a 5 v para alimentar al módulo Easy VR.

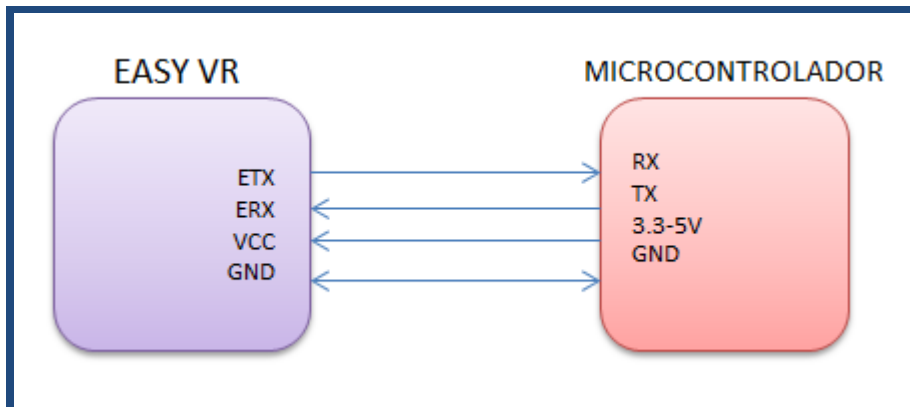


FIGURA 2.3 Conexión del EasyVR con un microcontrolador.

La configuración por defecto del módulo de voz es de 9600 bps, 8 bits de datos, no paridad, 1 bit de parada. La velocidad de transmisión se puede cambiar y puede estar en un intervalo entre los 9600 y 115200 bps.

El protocolo de comunicación sólo utiliza caracteres ASCII, que se dividen en dos grupos principales:

- ✓ Caracteres de comando y estado, respectivamente en las líneas RX y TX, elegidos entre letras minúsculas.
- ✓ Argumentos de comandos o datos de estado, una vez más en las líneas RX y TX, que abarca el rango de letras mayúsculas.

2.2.4 EasyVR Commander

El software EasyVR Commander se utiliza para configurar el módulo EasyVR conectado a la computadora, siempre y cuando sea controlado por un

microcontrolador con el programa "puente" que trabaja como intermediario entre el modulo y la computadora.

El usuario puede definir grupos de comandos o contraseñas y generar una plantilla de código para su manejo. Es necesario modificar el código generado para implementar la lógica de la aplicación, pero la plantilla contiene todas las funciones o subrutinas para manejar las tareas de reconocimiento de voz.

2.3 Modulo Xbee

El Xbee está diseñado para trabajar con los estándares IEEE 802.15.4 y proporcionar un bajo consumo de energía y bajo costo.

Los módulos operan en la banda ISM (Industrial, Científica y Médica) que son bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica.

Las bandas ISM son definidas por la ITU en el artículo 5 de las Regulaciones Radio (RR). Concretamente puntos 5.138 y 5.150. El uso de estas bandas de frecuencia está abierto a todo el mundo sin necesidad de licencia, los Xbees operan en la frecuencia de 2.4 GHz.

2.3.1 Estándar IEEE 802.15.4

El módulo Xbee utilizado en este proyecto trabaja con el estándar IEEE 802.15.4 que define el nivel físico y el de control de acceso al medio para redes LR-WPAN (Redes de área personal de baja transmisión de datos). El grupo de trabajo IEEE 802.15 es el responsable del desarrollo de este estándar.

Las principales características de este estándar es su bajo costo, bajo consumo de energía, gran flexibilidad de red y se puede utilizar en muchas aplicaciones en el hogar que requieran una tasa baja en la transmisión de datos, lo que lo hace ideal para la domótica.

En la figura 2.4 se observan los 16 canales de la banda de 2.4 GHz de la especificación IEEE 802.15.4, los canales están numerado del 11 al 26 , los canales del 0 al 10 están definidos para comunicaciones con frecuencias menores a 1 GHz. Para la banda de 2.4 GHz, debe existir una separación entre canales de 5 MHz.

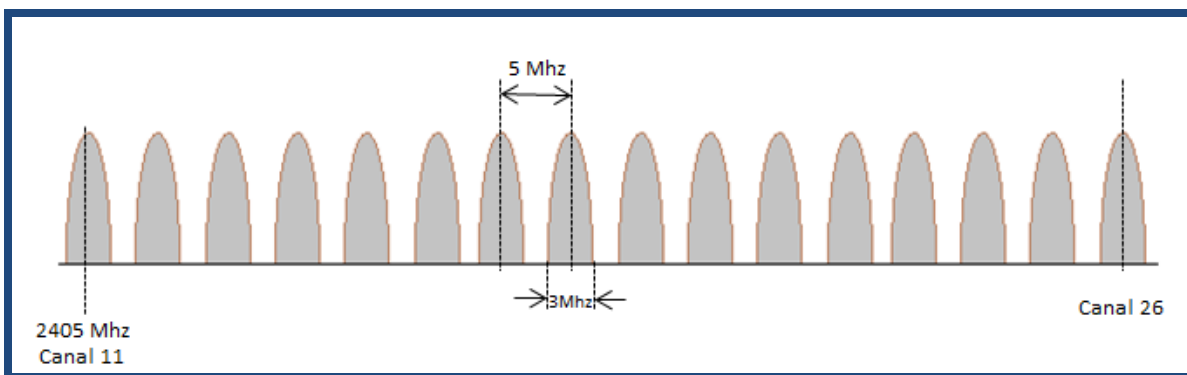


FIGURA 2.4 Canales disponibles para el protocolo IEEE 802.15.4 en la banda de 2.4 GHz

2.3.2 Características del Xbee

Los módulos Xbee son económicos y fáciles de utilizar. Algunas de sus principales características son:

- ✓ Buen Alcance: hasta 100 mts en línea de vista para los módulos Xbee y hasta 1.6 Km para los módulos Xbee Pro.
- ✓ 9 entradas/salidas con entradas analógicas y digitales.

- ✓ Bajo consumo <50mA cuando están en funcionamiento y <10uA cuando están en modo dormido.
- ✓ Interfaz serial.
- ✓ 65,000 direcciones para cada uno de los 16 canales disponibles. Se pueden tener muchos de estos dispositivos en una misma red.
- ✓ Fáciles de integrar.

El alcance en interiores se reduce a 30 metros, pero este alcance puede variar dependiendo del material de construcción de los obstáculos de la señal, en la tabla 2.1 se muestra los materiales y atenuación que le hará a la señal transmitida.

Objeto en la trayectoria de la señal.	Atenuación de la señal a través del objeto.
Pared recubierta	3 dB
Pared de vidrio con marco de metal	6 dB
Bloque de pared	4 dB
Ventana de oficina	3 dB
Puerta de metal	6 dB
Puerta de metal en pared de ladrillo	12 dB
Cuerpo humano.	3 dB

TABLA 2.1 Atenuaciones de las señales dependiendo el material.

En cada sitio donde se realice un estudio, se tendrán diferentes niveles de interferencia, pérdida de señal y ruido, la tabla 2.2 muestra la influencia de algunos materiales sobre la señal transmitida.

Material	Ejemplo	Interferencia
Madera	Tabiques	Baja
Vidrio	Ventanas	Baja
Amianto	Techos	Baja
Yeso	Paredes interiores	Baja
Ladrillo	Paredes interiores y exteriores	Media
Hojas	Árboles y plantas	Media
Agua	Lluvia / Niebla	Alta
Cerámica	Tejas	Alta
Papel	Rollos de papel	Alta
Vidrio con alto contenido en plomo	Ventanas	Alta
Metal	Vigas, armarios	Muy Alta

TABLA 2.2 Interferencia ocasionada dependiendo el material.

Una casa habitación está construida normalmente por tabique o ladrillo y aplanado por yeso o mezcla, y de acuerdo a las tablas estos materiales provocan poca atenuación e interferencia lo que hace al Xbee ideal para la transmisión en interiores de una casa habitación.

Existen 2 series de estos módulos. La serie 1 y la serie 2 ó también conocida como 2.5. Los módulos de la Serie 1 y la Serie 2 tienen el mismo pin-out, sin embargo, no son compatibles entre sí ya que utilizan distintos chipset y trabajan con protocolos diferentes.

La serie 1 está pensada para ser utilizado en redes punto a punto y punto a multipunto. Los módulos de la serie 2 están diseñados para ser utilizados en aplicaciones que requieren repetidores.

2.4 Manejo de timers y PWM en microcontroladores AVR de Atmel

Los microcontroladores son dispositivos programables de tecnología CMOS de 8,16 y 32 bits basados en tecnología RISC, capaces de ejecutar la mayoría de sus

instrucciones en cada ciclo de reloj. Tienen temporizadores, osciladores internos, USART, PWM, ADC, comparadores analógicos, entre otras cosas:

- ✓ Soportan programación en ensamblador y lenguaje C.
- ✓ Alto desempeño y bajo consumo de energía (menos de 1uA en estado apagado, y 1.1mA en activo).
- ✓ Cuentan con una gama de instrucciones sencillas en lenguaje ensamblador.
- ✓ Cuenta con 32 registros de propósito general.

Para el prototipo a diseñar, se necesita conocer el funcionamiento de los timers y las salidas PWM que tendrán la función de controlar los dispositivos de la casa.

2.4.1 Temporizadores

Un temporizador o timer, es un reloj especializado, que regularmente se ocupa para controlar la secuencia de un evento o proceso.

Un temporizador en un microcontrolador AVR, es un registro que almacena el número de pulsos de reloj que ocupa como base de tiempo y que, al llegar a un valor de sobre flujo, regularmente genera una interrupción.

El AVR atmega1248 posee cuatro temporizadores, dos de ellos, timer 0 y timer 2, son de 8 bits, mientras que el timer 1 y 3, son de 16 bits.

El microcontrolador ATMEGA1248 tiene 4 timers. El timer0 es de 8 bits, el timer1 es de 16bits, el timer2 es de 8 bits y el timer3 es de 16 bits. El timer0 está asociado a los pines B3 y B4 (4 y 5), el timer1 está asociado a los pines D4 y D5 (18 y 19), el Timer2 está asociado a los pines D6 y D7 (20 y 21) y el timer 3 está asociado a los pines B6 y B7 (7 y 8).

Se indica que determinados pines están asociados a un timer porque cuando sucede algún evento en el timer se pueden modificar esos pines poniéndose a 1 o a 0 cambiando de valor. Esto es que el Timer puede modificar directamente dos pines asociados a él.

2.4.2 Partes de los timers

Al trabajar con los timers, se usan registros que tienen tareas específicas, los cuales son los siguientes:

2.4.2.1 Registro TCNTX

Es el corazón del timer que se utilice y básicamente, lleva la cuenta de los pulsos de reloj o los eventos externos. Para los timers 0 y 2 el registro es de 8 bits que es el mostrado en la figura 2.5 y para los timers 1 y 3 es de 16 bits mostrado en la figura 2.6. De tal forma que los de 8 bits pueden contar desde 0 hasta 255 y los de 16 bits desde 0 hasta 65535.

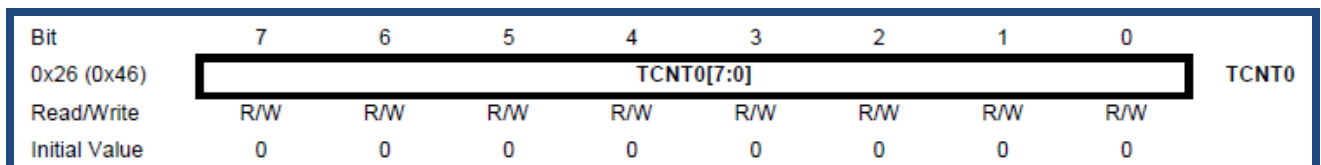


FIGURA 2.5 Registro de 8 bits para timers 0 y 2.

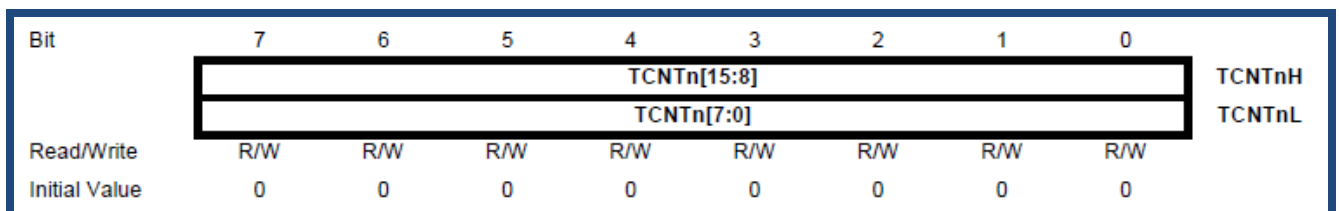


FIGURA 2.6 Registro de 16 bits para timers 1 y 3.

La cuenta de estos timers se realiza de acuerdo al pre escalamiento de la frecuencia interna del microcontrolador, que puede ser de 1, 864, 256 o 1024 el factor de pre escalamiento. Es importante señalar que el pre escalador divide únicamente la frecuencia de operación del timer y no la frecuencia del micro controlador.

2.4.2.2 Registro OCRXA y OCRXB

OCRXA : Es el registro de comparación del timerX salida A, es un registro de 8 o 16 bits según el timer que se esté usando y es accesible para el programador que continuamente se compara con el registro TCNTX y cuando son idénticos se puede generar una acción al pin que está asociado a la salida A del timerX.

OCRXB : Es el registro de comparación del timerX salida B, es un registro de 8 o 16 bits según el timer que se esté usando y es accesible para el programador que continuamente se compara con el registro TCNTX y cuando son idénticos se puede generar una acción al pin que está asociado a la salida B del timerX.

En la figura 2.7 se muestra como trabajan las partes esenciales del timer en conjunto, es un ejemplo para el timer0. La frecuencia del cristal interno, que es de 8 MHz, entra a un pre escalamiento de 8, para obtener la frecuencia a la cual trabajara todo el microcontrolador (1 MHz), posteriormente esta frecuencia puede ser dividida para el uso exclusivo del timer0, y está frecuencia es la que hace incrementar al contador TCNT0, el cual es comparado con los registros OCR0A y OCR0B y cuando alguno de ellos es idéntico con TCNT0 se pueden efectuar acciones sobre los pines de salida A y B (pin B3 y pin B4 respectivamente).

El timer va incrementándose en una unidad cada determinada cantidad de tiempo, según la frecuencia y el pre-escalamiento, pero el timer tiene un determinado ancho en bits por lo que el número máximo que puede almacenar está limitado.

Por ejemplo el timer0 es de 8 bits por lo que puede tener un número desde 0 hasta 255 (0 a FFh) pero cuando tiene 254 y cuenta una unidad más, el timer se incrementa a 255, y si cuenta un número más ya no cabe el 256 en 8 bits, por lo que el timer se regresa a 0 esto se conoce como overflow, es decir llegar al valor máximo y contar un pulso más que hace que el timer regrese a cero. Este overflow se puede utilizar para activar una interrupción en el microcontrolador y hacer la acción que deseemos.

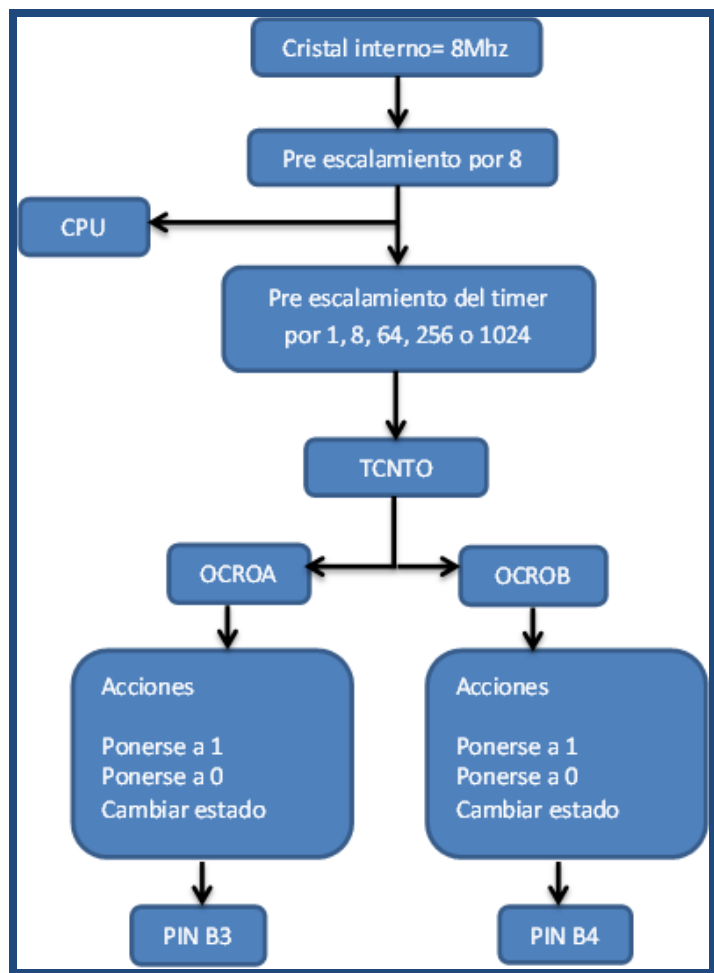


FIGURA 2.7 Diagrama de las partes fundamentales de un timer en los microcontroladores AVR.

2.4.3 Modulación por ancho de pulso PWM

El control analógico no siempre es la mejor opción en la industria pues es más propenso a alteraciones por ruido y con el tiempo suele presentarse cambios lo cual complica la sincronización de los mismos. Los controles analógicos de alta precisión suelen ser muy caros, y voluminosos.

La modulación por ancho de pulso, o PWM, es una técnica para control de circuitos analógicos utilizando una señal digital. En este caso la señal digital del microcontrolador.

Vista de una forma sencilla la PWM es una forma de codificar señales analógicas de una forma digital. A través del uso de contadores una señal cuadrada es modulada para obtener un determinado nivel analógico.

2.4.4 PWM en el Atmega1248

Señales PWM. Una señal PWM o modulación por ancho de pulso tiene las siguientes características: el periodo de la señal es fijo, lo que cambia es el ancho de pulso.

PWM no invertida, en la figura 2.8 se aprecia que el periodo de la señal es fijo y lo que se modifica es el ancho de pulso, en el primer ciclo la modulación está al 50%, esto es que durante todo el 50% del periodo de la señal estará en 1 el pulso, en el siguiente ciclo se encuentra la modulación es del 75%, así que estará en 1 durante $\frac{3}{4}$ partes del periodo de la señal, y en el último ciclo está al 100% la modulación, así que está en 1 todo el tiempo que dura el periodo de la señal.

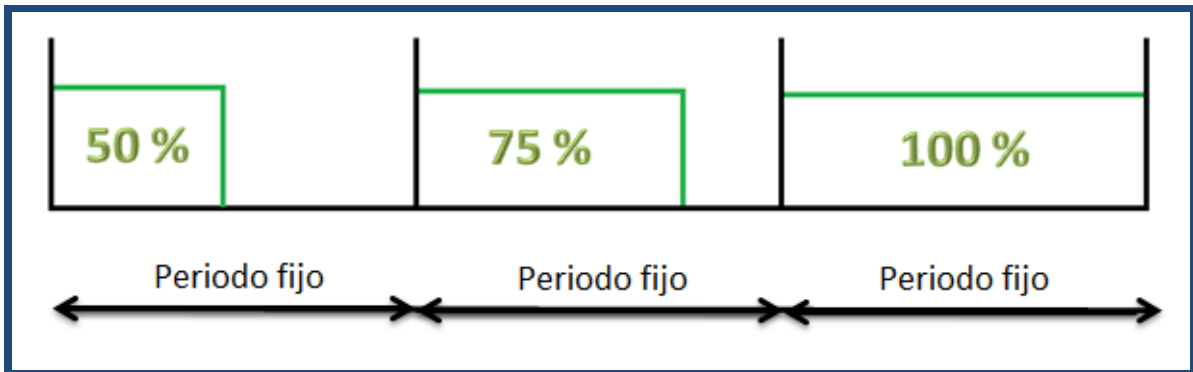


FIGURA 2.8 Diferentes niveles de modulación en PWM.

Las señales PWM son muy utilizadas en control de energía de CA y CD. Si se conecta un LED con un PWM al 50% se notaría que está ligeramente encendido, si se aumenta el ancho de pulso el LED prendería con mayor intensidad, y si la modulación está al 100% el LED prendería al máximo de intensidad luminosa.

El microcontrolador Atmega1248 posee 8 canales de PWM (OC1A, OC1B, OC3A, OC3B, OC2A, OC2B, OCOB y OC0A), las 4 primeros son canales de 16 bits mientras que los otros cuatro de 8 bits.

2.4.5 PWM rápida

Esta señal PWM no invertida es la que se programará en el atmega1284 y tiene las siguientes características:

- PWM de alta frecuencia.
- Modo de operación de una sola pendiente o cuesta simple.
- Modo de operación invertido y no invertido.
- Resolución de 8, 9, y 10 bits.

La PWM rápida proporciona la opción de una generación de forma de onda PWM de alta frecuencia. Debido a su pendiente sencilla, la frecuencia de operación de la

PWM rápida puede ser el doble de rápida que la PWM de fase correcta o PWM de fase y frecuencia correcta que emplean doble pendiente. Esta alta frecuencia permite que la PWM rápida sea ideal para el control de potencia, rectificación, aplicaciones de DAC.

Su funcionamiento está relacionado con el uso de los timers, en la figura 2.9 se muestra como se genera la señal PWM rápida usando los timers como referencia para regular la modulación de la señal.

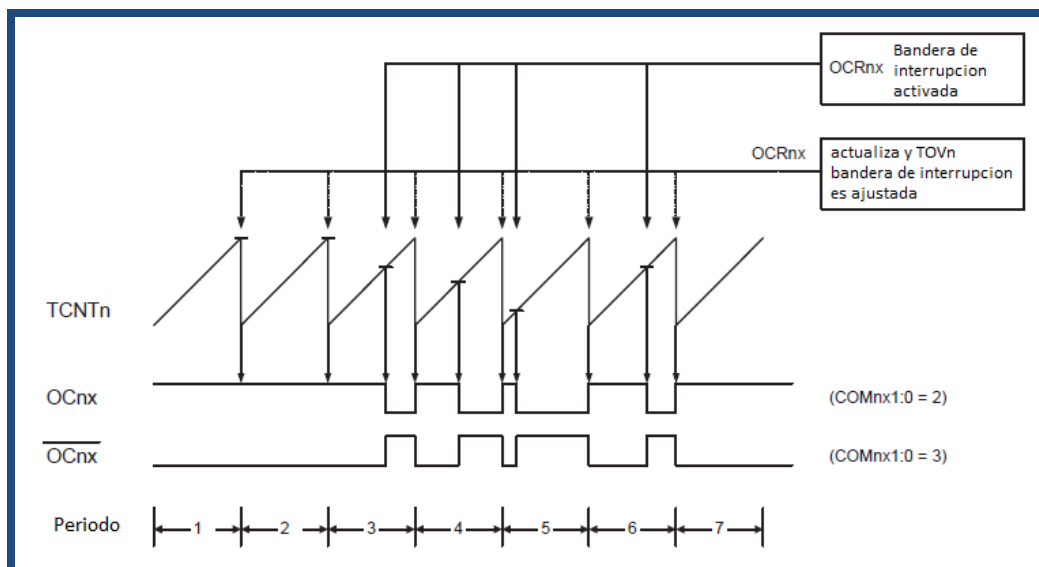


FIGURA 2.9 Diagrama de funcionamiento de la PWM rápida.

El periodo de la señal viene definido por la cantidad máxima de conteo del timer en uso TCNTn y en función del pre escalamiento de la señal de reloj del microcontrolador, por esta razón, se dice que esta PWM es de pendiente sencilla.

Para elegir el valor de la modulación en la salida OCnx, se debe ajustar el valor del registro OCRnx, y así cuando el valor del timer TCNTn sea igual al valor de OCRnx, se efectúa un cambio de nivel en la salida de la señal. De tal forma que

cuando OCRnx está configurado al valor de la cuenta máxima de TCNTn, la modulación es al 100%, como se puede apreciar en el periodo 1 y 2 de la figura.

En los siguientes periodos el valor de OCRnx está por debajo del valor máximo de TCNTn, lo que provoca que existan cambios de nivel en el periodo de la señal, y entre más bajo sea el valor de OCRnx, más bajo es el porcentaje de modulación de la señal PWM, como se aprecia en el periodo 5.

Usando este tipo de modulación es posible controlar la iluminación de un cuarto, siendo posible efectuar las funciones de niveles de cambio tales como: Encender el foco (modulación al 100%), apagar el foco (modulación al 0%) y regular la intensidad del mismo (por ejemplo modulación al 30%).

2.5 Diseño del sistema

A partir del análisis el prototipo se divide en 2 grandes bloques que son el generador de comandos y el modulo central mostrados en la figura 2.10.

- ✓ En el primer bloque se generan comandos de control a través de un reconocimiento de voz y estos comandos son transmitidos inalámbricamente.
- ✓ En el segundo bloque los comandos son recibidos y procesados para controlar a través de la etapa de potencia las variables.

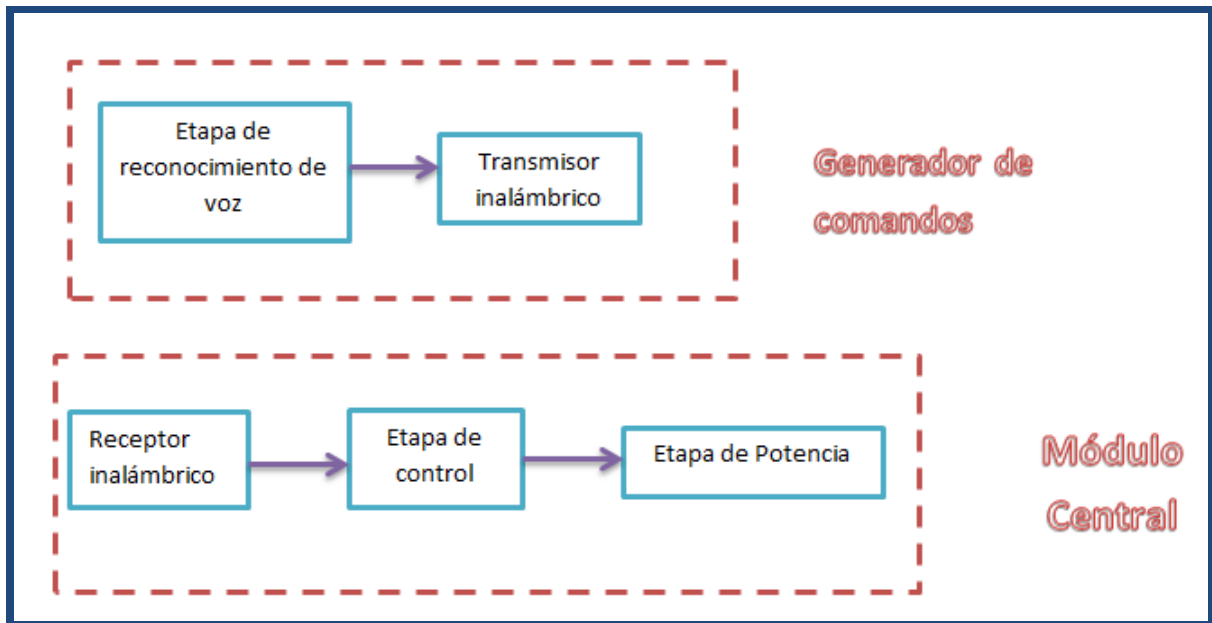


FIGURA 2.10 Diagrama a bloques del prototipo.

2.5.1 Generador de comandos

El objetivo de esta etapa es que a través de un reconocimiento de voz se generen comandos de control que van a ser enviados inalámbricamente al módulo central.

2.5.2 Etapa de reconocimiento de voz

Para este bloque se emplea el módulo de voz EasyVr Shield en conjunto con el microcontrolador atmega328 como se muestra en la figura 2.11.

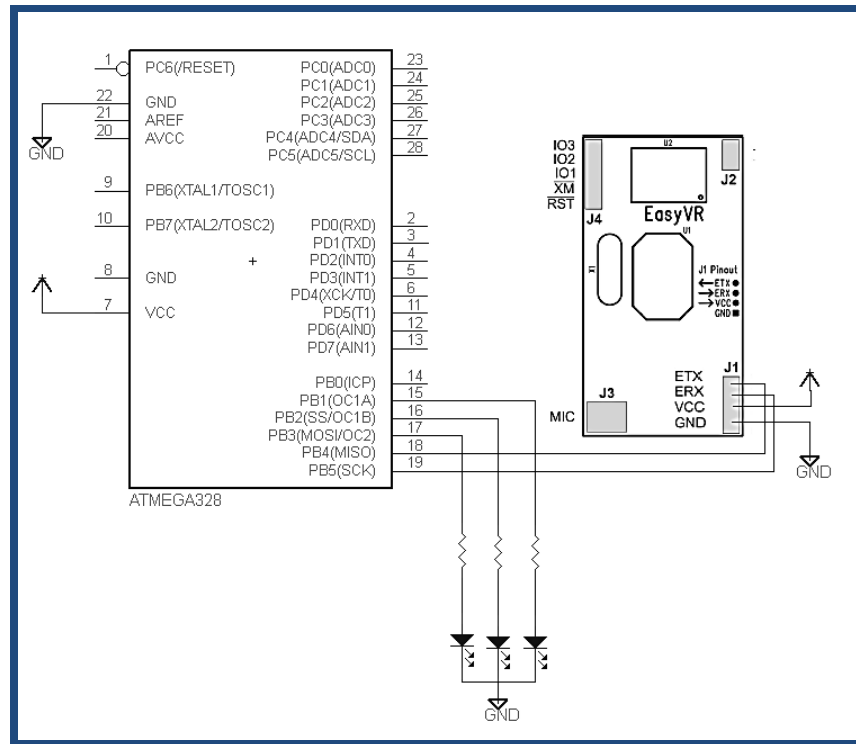


FIGURA 2.11 Conexión del atmega328 con el módulo EasyVr.

Una plataforma de desarrollo para usar este microcontrolador es “Arduino”, el fabricante proporciona la placa para poderlo programar y tener acceso a sus entradas y salidas, en la figura 2.12 se muestra la placa de Arduino adquirida para el proyecto.



FIGURA 2.12 Placa de desarrollo Arduino.

La placa EasyVr shield mostrada en la figura 2.13 está diseñada para trabajar en conjunto con las placas de “Arduino”, lo que facilita el uso del hardware, ya que ambas placas se conectan colocando una encima de la otra a través de los headers que tienen en sus costados lo cual se puede apreciar en la figura 2.14.

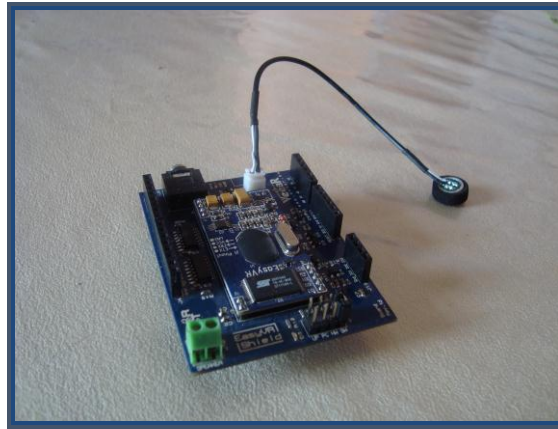


FIGURA 2.13 Modulo de voz EasyVR Shield.

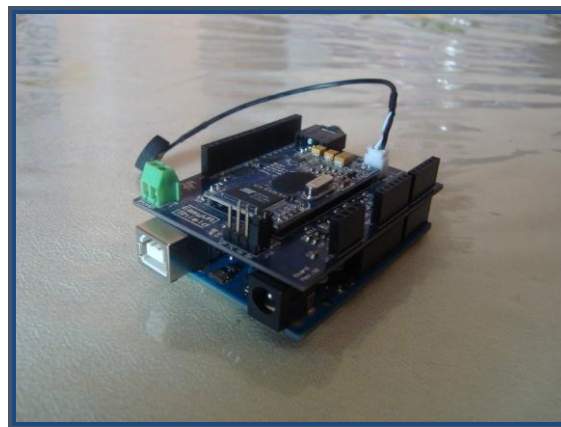


FIGURA 2.14 Montaje del EasyVR Shield con la placa Arduino.

La combinación de ambas placas forma la etapa de reconocimiento de voz, que además debe proporcionar comandos de control, y esto lo procesará el microcontrolador una vez que el EasyVR le informa que comando de voz dijo el usuario.

2.5.3 Comandos de voz

En el módulo EasyVR se deben grabar los comandos que son hablados por el usuario final. Para el prototipo, se propuso tener control de la iluminación de una sala, un cuarto y un ventilador. Las funciones de control son las siguientes:

- ✓ Temporizador: Se establece un tiempo (30 seg para fines demostrativos) para que uno de los dispositivos permanezca encendido durante ese tiempo.
- ✓ Niveles de cambio: Encender, apagar y regular la intensidad.
- ✓ Detección de errores: Esta función se realiza automáticamente al encender alguno de los dispositivos a controlar, si el dispositivo no sirve, se mostrará en la pantalla un mensaje haciendo referencia a que no funciona.

En base a estas funciones, los comandos propuestos son los que se muestran en la tabla 2.3 , además incluye que letra será la que envía el microcontrolador para la etapa de transmisión:

Número	Comando hablado	Letra	Descripción
1	Enciende cuarto	a	Enciende el foco de un cuarto
2	Apaga cuarto	b	Apaga el foco de un cuarto
3	Regula cuarto	c	Regula la intensidad de un cuarto
4	Enciende ventilador	d	Enciende un ventilador
5	Apaga ventilador	e	Apaga un ventilador
6	Regula ventilador	f	Regula la intensidad de un ventilador
7	Enciende sala	g	Enciende el foco de la sala
8	Apaga sala	h	Apaga el foco de la sala
9	Cuarto en 30 segundos	i	El foco del cuarto se mantiene enciendo durante 30 segundos

TABLA 2.3 Descripción de los comandos de voz.

Estos comandos son enviados a través del pin PD1 del atmega328 que es para transmisión de datos asíncrona.

Para acceder a estos comandos, es necesario decir una palabra de activación, que por default es “robot” ya que una de las aplicaciones más comunes del módulo de voz es usarlo como interfaz para interactuar con un robot y darle ordenes de movimiento, pero en este caso esa palabra no es adecuada para el prototipo.

Por lo que se propuso, usar una palabra corta y fácil de recordar, que haga sentir al usuario que esta interactuando con un sistema que ejecuta las ordenes que él le diga, por lo que se propuso la palabra “Modulo”, haciendo referencia al módulo de voz, ya que decir “Módulo de voz” toma más tiempo y puede ocasionar más probabilidad de error, se redujo la palabra solamente a “Modulo”

Por lo tanto “Modulo” sería el equivalente al nombre del sistema, así el usuario puede sentir que le está dando órdenes al “Modulo”, ya que por ejemplo, para encender el cuarto el usuario debe decir: “Modulo, enciende cuarto”.

El programa para el atmega328 está basado en el lenguaje de programación de Arduino, además de que el software EasyVR commander tiene la opción de generar una plantilla de código en Arduino para solamente agregar las funciones que se quieran ejecutar por cada comando de voz hablado.

Básicamente lo que hace el programa es estar esperando el momento en el que el usuario mencione la palabra clave, una vez que es reconocida por el microcontrolador, se pasa a la siguiente etapa, que es el reconocimiento del comando de control que puede ser cualquiera de los descritos en la tabla 2.3.

Por último cada uno de estos comandos tiene una referencia en el programa y se usa como parámetro para elegir una letra que será enviada a través del puerto de transmisión del microcontrolador para ser transmitida a la siguiente etapa.

2.6 Transmisor y receptor inalámbrico

Para estas dos etapas, se emplean los chips Xbee trabajando en una comunicación punto a punto. Y la transmisión de datos es asíncrona con las características de transmisión que se eligieron en la programación del atmega328 que son:

- ✓ Tasa de transmisión de bits de 9600 bps.
- ✓ 1 bit de parada.
- ✓ 8 bits de información.
- ✓ No paridad.

Esta configuración se establece con el software X-CTU. El diagrama básico de conexión de los chips Xbee con sus respectivos microcontroladores se muestra en la figura 2.15.

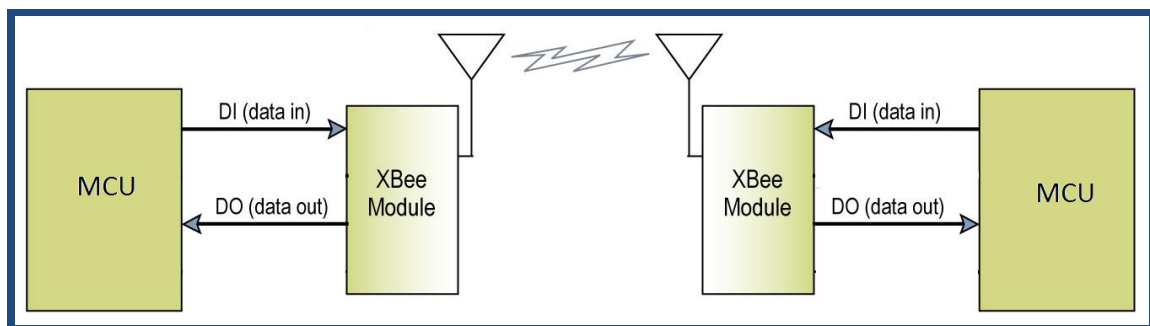


FIGURA 2.15 Conexión básica para los Xbee.

Para el prototipo, en el transmisor sólo será necesario emplear el pin DI para que el módulo Xbee envíe datos, y en su contraparte en el receptor solo se emplea el pin DO para que el dato transmitido sea procesado por el microcontrolador.

Para hacer pruebas de la transmisión inalámbrica se usa el circuito ya acoplado al módulo de voz de la figura 2.16:

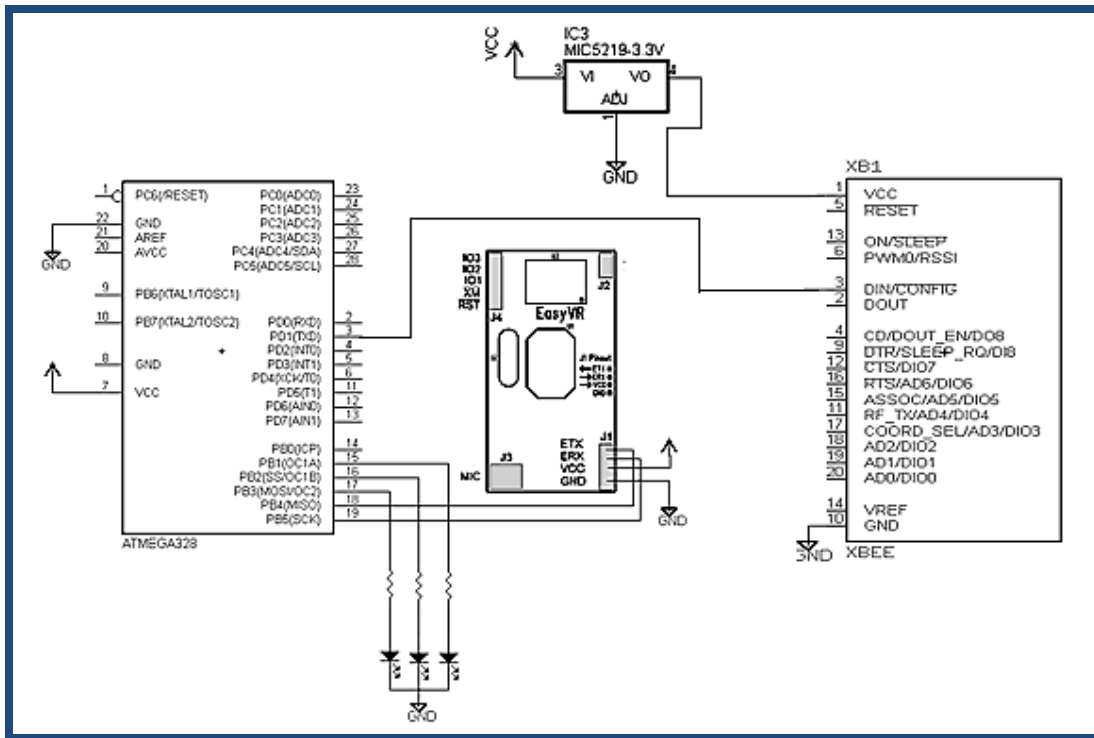


FIGURA 2.16 Circuito transmisor de comandos de control.

El Xbee receptor se conecta a la placa “Xbee Explorer USB” como se muestra en la figura 2.17, la cual es una placa que adapta los comandos seriales para que puedan ser transmitidos por cable USB y pueda la placa Xbee conectarse a una computadora y con el software X-CTU observar los comandos que son recibidos.

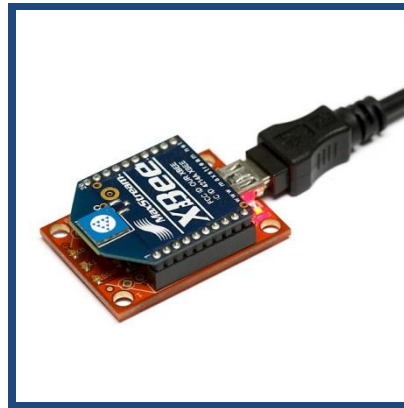


FIGURA 2.17 Xbee receptor conectado a la placa "Xbee explorer USB".

2.7 Etapa de control

El objetivo de esta etapa es recibir los comandos enviados por el transmisor en la etapa de generación de comandos y realizar la tarea que se le asigne de acuerdo a la letra recibida. El diagrama de esta etapa incluyendo su receptor inalámbrico se aprecia en la figura 2.18.

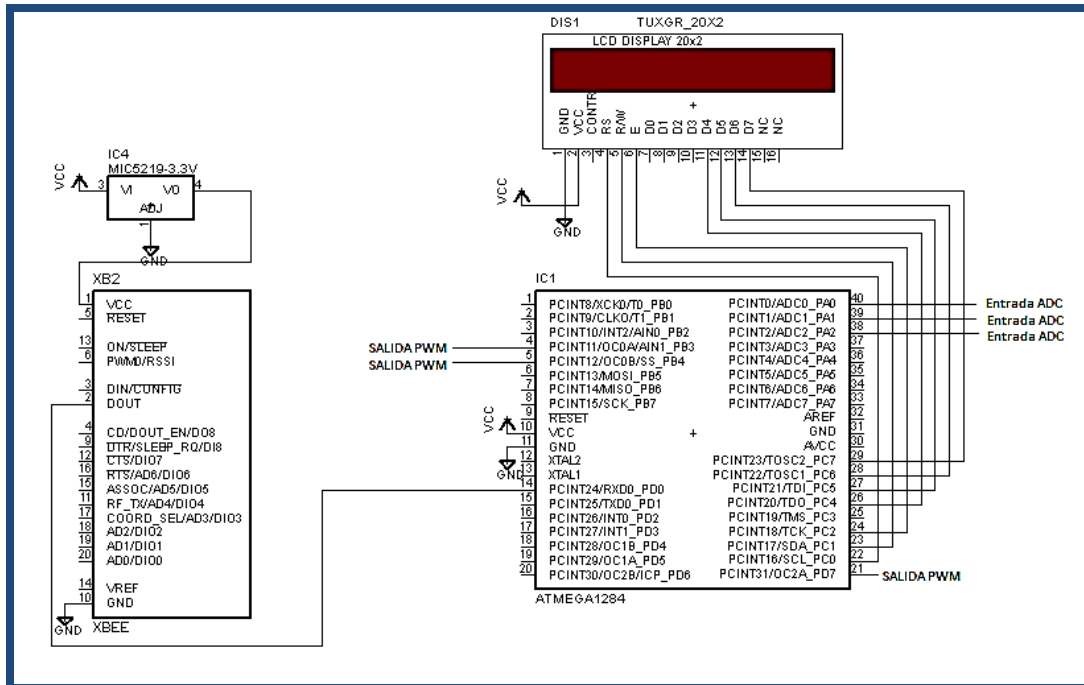


FIGURA 2.18 Diagrama de la etapa de control.

Para la recepción de comandos, se configura la comunicación asíncrona con las mismas características con las que se programan los Xbee y el atmega328. De tal forma que cuando un carácter se reciba, se active la **interrupción vectorizada** “USART1_RX_vect” para que la letra recibida se guarde en el registro UDR1 y posteriormente de acuerdo a esa letra, se ejecute alguna de las funciones de control que puede realizar el prototipo.

Para la función de niveles de cambio, se configuran salidas PWM para tener una PWM de alta frecuencia, ideal para el control de potencia.

A través de los registros OCR0A, OCR0B y OCR2A correspondientes a los pines PB3, PB4 y PD7 que son salidas PWM se lleva a cabo el encendido, apagado o regulación de intensidad de los dispositivos a controlar. Dependiendo del valor que se les asigne, comprendido entre 0 y 255 ya que son de 8 bits y son comparados con los timers 0 y 2.

Cuando alguno de estos registros sea igual a cero, su salida PWM estará modulada al 0%, lo que provocará que el dispositivo que se esté controlando permanezca apagado.

Cuando el registro sea igual a 255, la salida PWM esta modulada al 100%, lo que provoca que el dispositivo permanezca encendido.

Y para la regulación de intensidad, se propone un valor intermedio como 125, para tener una modulación aproximada del 50% para que la intensidad del foco o del ventilador disminuya.

Para la función de temporizador, se hace uso de la interrupción vectorizada “TIMER0_COMPA_vect” la cual se activa cuando el timer 0 tiene el mismo valor que el registro OCR0A. El valor máximo de conteo del timer 0 es de 255, y si la

preescala del reloj es de 64 y su frecuencia es de 1Mhz, la frecuencia para el timer será de 15625 Hz, con un periodo de 64 microsegundos, lo que dura un ciclo de reloj preescalado para el timer0. Cuando el timer llegue a la cuenta máxima se tendrá un tiempo de 0.01632 segundos y ahí es cuando se activa la interrupción si OCROA es igual a 255.

Para llegar a 1 segundo completo, esta interrupción se debe activar 61 veces, por lo que hay un contador en el programa para llevar el conteo de veces que se ha activado la interrupción y así ejecutar una subrutina en la cual se actualiza la información del LCD por si ha llegado algún nuevo comando o si hay algún fallo en los dispositivos, y en esta subrutina se incluye un contador que lleva la cuenta de cuantas veces se ha llamado esta subrutina que es lo equivalente a contar cada segundo para que dure el foco prendido un determinado tiempo, si son 30 segundos, será que la subrutina fue llamada 30 veces.

Para la detección de errores, se usa el convertidor analógico digital, ya que en la siguiente etapa, la de potencia, se usan circuitos PWM para manejar una fuente de 12 V. Lo que hará el microcontrolador es que a través de sus entradas ADC compara el voltaje convertido con un valor de referencia que es de 2 volts, si es mayor el voltaje convertido el dispositivo está funcionando correctamente, en caso contrario habrá una falla. Así en caso de que este encendido el foco, es porque el circuito está cerrado y si se pone una resistencia en serie y se mide su voltaje con el ADC, la comparación debe dar un número mayor a 2, lo que significa que el foco está funcionando correctamente.

Pero en caso de que se halla activado el foco y este no sirva, el circuito estará abierto, y como consecuencia no habrá voltaje en la resistencia de prueba, lo que hará que el valor del ADC sea menor al valor de referencia.

2.8 Etapa de potencia

2.8.1 Circuito PWM

Se usan tres circuitos PWM como el de la figura 2.19 para controlar los dispositivos del sistema que son los 2 focos led y el ventilador. El opto acoplador 4N35 se utiliza para proteger al microcontrolador, ya que en esta etapa habrá flujo de corriente que se regresa hacia el microcontrolador y dañará el pin de E/S del microcontrolador.

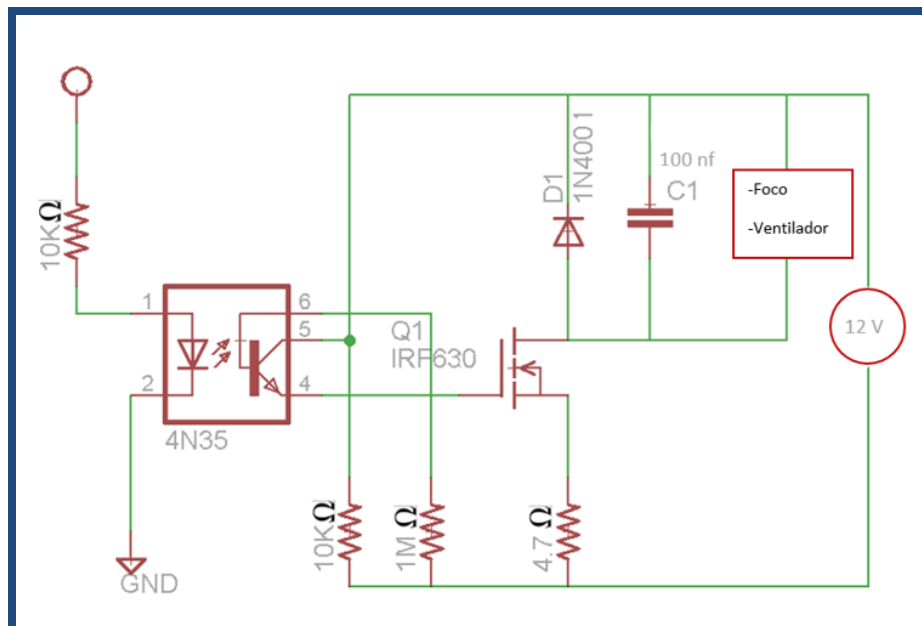


FIGURA 2.19 Circuito PWM.

El IRF630 es un transistor N-MOSFET, que funciona como un interruptor, cuando $V_{GS} > 2V$ (valor lógico alto) el dispositivo conectado estará encendido y cuando $V_{GS} < 2V$ (valor lógico bajo) el dispositivo estará apagado.

El diodo se utiliza para absorber la carga almacenada en el ventilador, cuando la tensión de alimentación del motor se reduce repentinamente, habrá un riesgo de que la tensión vuelva y puede perjudicar al microcontrolador, por lo que el diodo puede eliminar el riesgo.

La resistencia de $1\text{M } \Omega$, se recomienda utilizar para que el transistor en el 4N35 tenga una mejor respuesta a la señal PWM. En el BUZ73, se añadió una resistencia para hacer la función de detección de fallos. Se utiliza el atmega1284 para medir la tensión entre la fuente y tierra para comprobar si el dispositivo funciona de una manera correcta, de lo contrario, la pantalla LCD mostrará que el dispositivo tiene un error.



CAPÍTULO 3 | “Pruebas del sistema y resultados”

3.1 Introducción

Las pruebas se realizan paso a paso, desde el grabar los comandos de voz en el módulo de voz, enviar información de Xbee a Xbee observando que se reciba la información, que los dispositivos a controlar realizaran sus funciones, pruebas en una casa y observar qué factores interviene la transmisión para un buen funcionamiento.

3.2 Prueba en la etapa de reconocimiento de voz

El primer paso es grabar los comandos de voz a utilizar en el prototipo, primero se emite la palabra “Modulo” para accionar el grupo, y dentro de ese grupo se tienen todas las funciones a realizar, como “apaga cuarto”, “enciende sala”, “regula ventilador”, etc.

Estos comandos de voz se graban con la herramienta “esasyVR commander”, la cual es un software. Se graban los comandos de voz de distintas formas, a distintas velocidades al emitir la palabra, con diferente tono de voz y diferentes voces, se puede apreciar en la figura 3.1 el programa donde se pueden grabar editar los comandos.

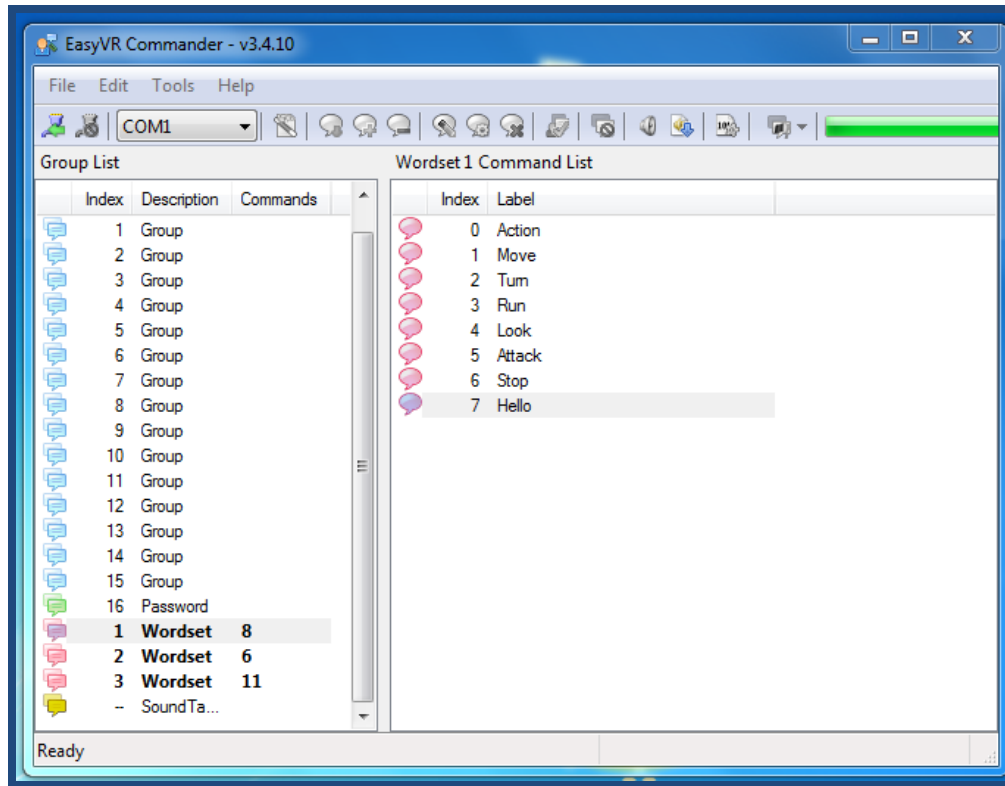


FIGURA 3.1. (EsasyVR commander) Software a utilizar para grabar y probar los comandos de voz.

La primera complicación que se obtiene, es en el momento en que uno de los 3 integrantes del equipo graba todos los comandos de voz, ya grabados los comandos, las otras 2 personas intentan activar los comandos, a pesar de decir la palabra correctamente, es complicado activar el comando por la forma y velocidad de emitir las palabras

Lo que hace que active al comando, es la similitud en tiempo con la que se emite la palabra respecto a las palabras ya grabadas, es decir, es capaz de diferenciar y activar el comando con cualquier tipo de voz pero lo más parecido al emitir la palabra con las palabras grabadas, si una persona tiene la voz muy grave y otra muy aguda, el sistema lo reconocerá bajo las condiciones de mencionar la palabra lo más cercano al tiempo en que se graba, también si se resaltan algunas letras o partes de la palabra, que sea lo más similar posible para obtener el mínimo número de errores.

Se observa que su principal limitante es reconocer la palabra de tal forma que se parezca en velocidad, más que en tono, a la que ya está grabada, bajo esta limitante se selecciona palabras cortas para su fácil activación de los comandos.

Las palabras se graban de una forma sencilla, es decir, una persona comúnmente no dice “apaga sala” muy pausadamente o resaltando alguna palabra, sino solo apaga sala, de una forma en la que la gente comúnmente lo hace, con la finalidad de que cualquier persona que pueda interactuar con el equipo pueda accionar las funciones fácilmente sin temor a que no se realice dicha acción.

En esta etapa se trabaja en conjunto el ATmega328 con el módulo de voz EasyVr Shield, empleando a la plataforma de desarrollo Arduino como se muestra en la figura 3.2.

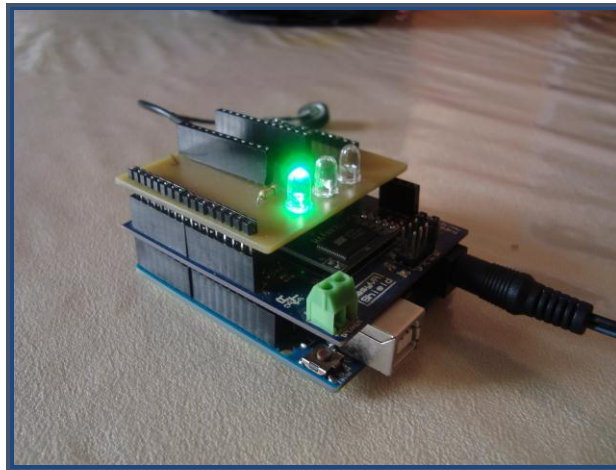


FIGURA 3.2 Prueba de comunicación del ATmega328 y el módulo de voz.

El software que utiliza Arduino es muy amigable, permite saber si los comandos de voz son correctos, muestra en pantalla lo que recibe por los comandos de voz , muestra el carácter que va a ser enviado a la etapa de transmisión y permite avisar al usuario si fue exitosa la comunicación, en la figura 3.3 y 3.4 se muestra como empieza interactuar el módulo de voz con Arduino.

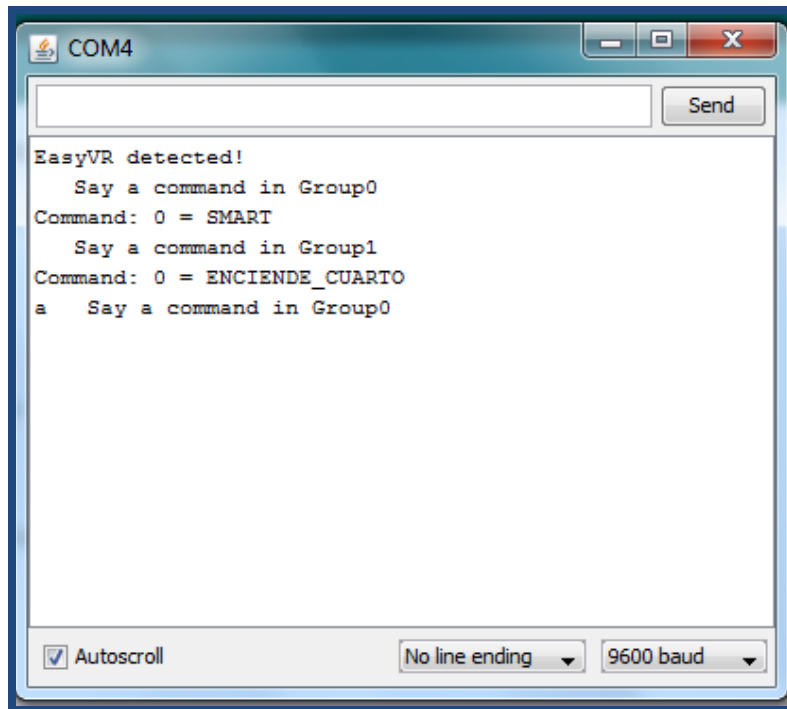


FIGURA 3.3 Software de Arduino mostrando comando "enciende cuarto" y el carácter que se enviará.

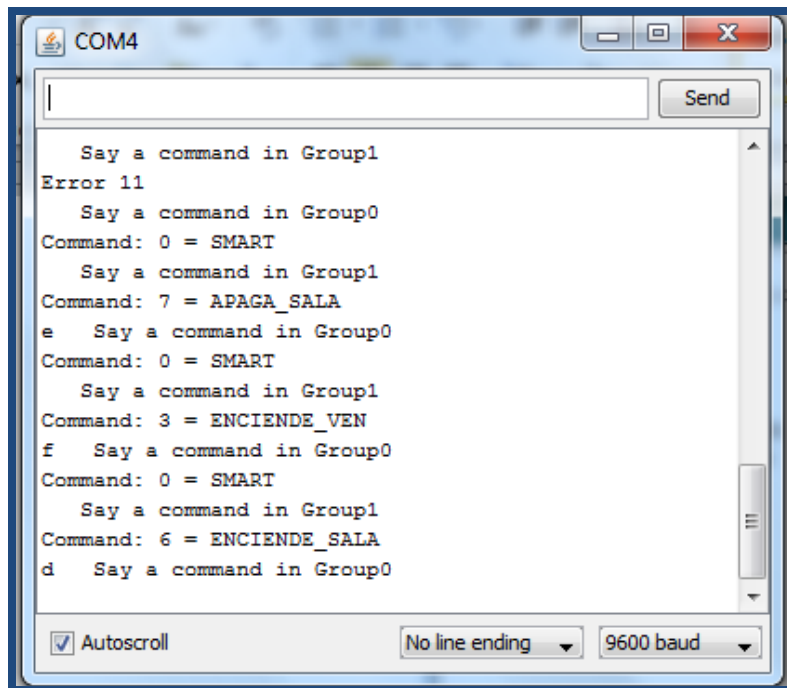


FIGURA 3.4 Software de Arduino mostrando varios comandos y los caracteres que se envían.

3.3 Pruebas de transmisión y recepción inalámbrica

Ahora que se ha grabado los comandos de voz, se ha de seguir con él envío de la información, para esto, se transmite mediante los Xbee, ocupando primeramente el Xbee de transmisión, esto se lleva acabo activando los comandos por medio de la voz, estos los envía el Xbee de transmisión en forma de caracteres (a, b, c, d, e, f, g, h, i.) al receptor, para asegurarse de que la información se envía correctamente se conecta el Xbee de recepción a la PC para observar que es lo que recibe con la ayuda de la herramienta X-CTU.

X-CTU permite, aparte de configurar el Xbee, observar que es lo que recibe de información, en este caso solo recibe información de un Xbee, ya que se utiliza la configuración punto a punto. Observamos la configuración en el X-CTU en la figura 3.5.

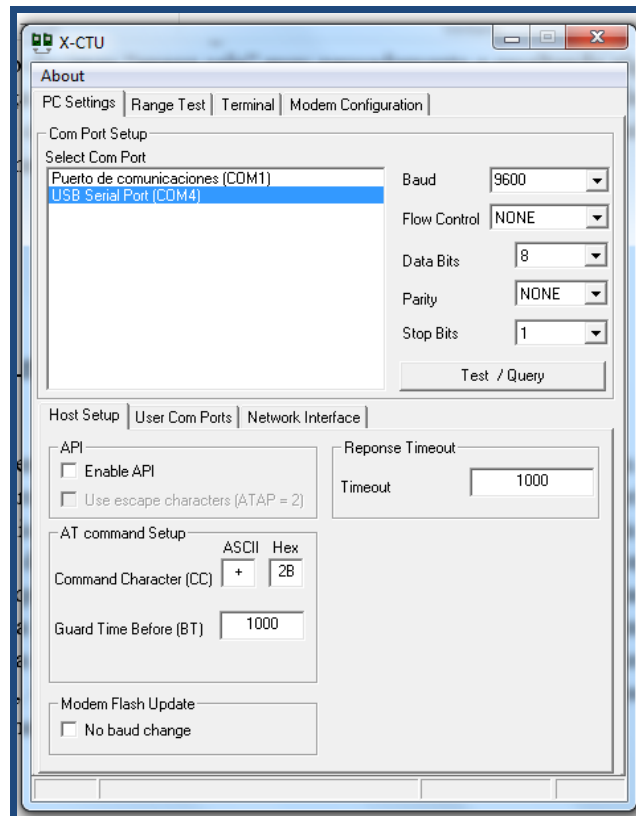


FIGURA 3.5 (X-CTU) Software a utilizar para configurar los Xbee y hacer pruebas de la transmisión de caracteres.

Cada carácter tiene su representación en binario, el Xbee receptor recibe el comando y lo interpreta como carácter, lo cual el software X-CTU imprimirá en pantalla el carácter con su representación en hexadecimal, por ejemplo; el comando “apaga cuarto”, está dada por el carácter “c” y su representación en hexadecimal será 63 (la representación del hexadecimal lo da por defecto el software), el microcontrolador debe recibir ese carácter y él lo interpreta como “0110 0011” como se muestra en la figura 3.6:

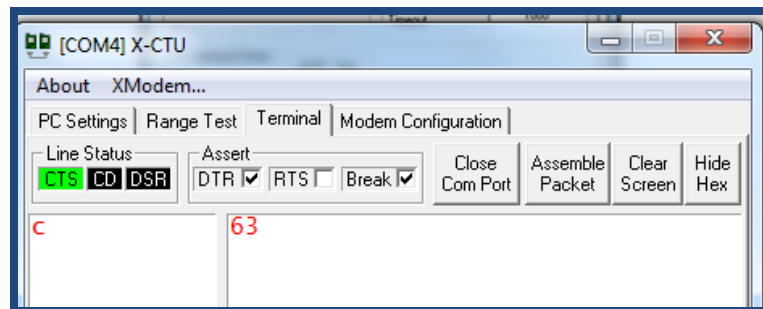


FIGURA 3.6 Pruebas de recepción del comando “apaga cuarto” y mostradas por el software X-CTU.

Este procedimiento se realiza para todos los comandos grabados, por ejemplo el comando “enciende ventilador” representa el carácter “f”, el número 66 en hexadecimal y por lo tanto el número “0110 0110” en binario como lo muestra la figura 3.7.

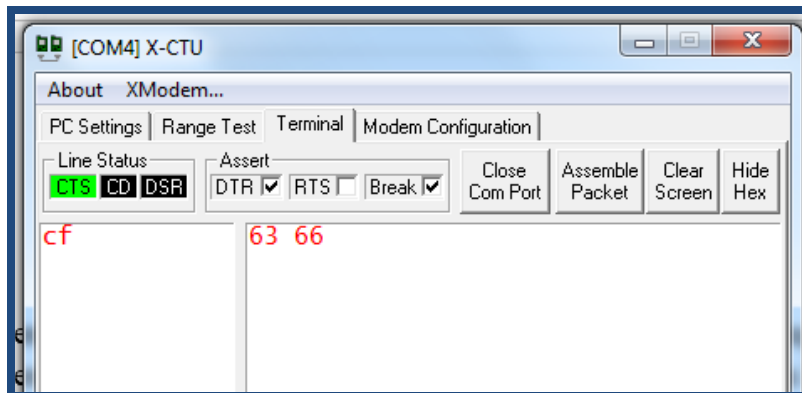


FIGURA 3.7 Pruebas de recepción del comando “apaga cuarto” y “enciende ventilador” mostradas por el software X-CTU.

El software ayuda a corroborar que lo que se manda es correcto, ya que si el prototipo tiene errores, conocer hasta ese punto si el sistema tiene o no fallas.

3.4 Pruebas de la etapa de control

Hasta ahora se ha logrado transmitir la información hasta el Xbee receptor, corroborando con el X-CTU que la información que se envía es la correcta. Ahora es turno del Atmega1284 que reciba esa información y realizar su trabajo, para ello se realizan pruebas preliminares, el LCD sirve de apoyo para corroborar que la acción deseada sea cumplida, si se manda desde el módulo de voz que apague la sala, el LCD nos mostrara que el foco de la sala ha sido apagada, y el microcontrolador apagara el foco de la sala, este paso se realiza primeramente con diodos led's a las salidas del microcontrolador antes de conectarlos a los circuitos PWM, con el fin de corroborar nuevamente que la información a las salidas del ATmega1284 se ejecutan con éxito, como se muestra en la figura 3.8.



FIGURA 3.8 Pruebas de la etapa de control.

En esta etapa está configurado asíncronamente con las mismas características que el chip de Xbee transmisor. Se enciende el foco de la sala, el LCD indica que acción se produce. El prototipo es capaz de realizar varias acciones a la vez, se puede encender el cuarto y a la vez el ventilador, quizás se quiere regular la luz y el ventilador a la vez para sentir más comodidad, se realizan pruebas respectivamente de todas las combinaciones posibles para observar que efectivamente el prototipo esta en óptimas condiciones, se observa el funcionamiento en la figura 3.9.



FIGURA 3.9 Pruebas de la etapa de control realizando varias funciones.

En la figura 3.10 se muestra la salida PWM que genera el ATmega1284 mediante un osciloscopio, con la finalidad de ver la estabilidad de los pulsos que genera el microcontrolador.

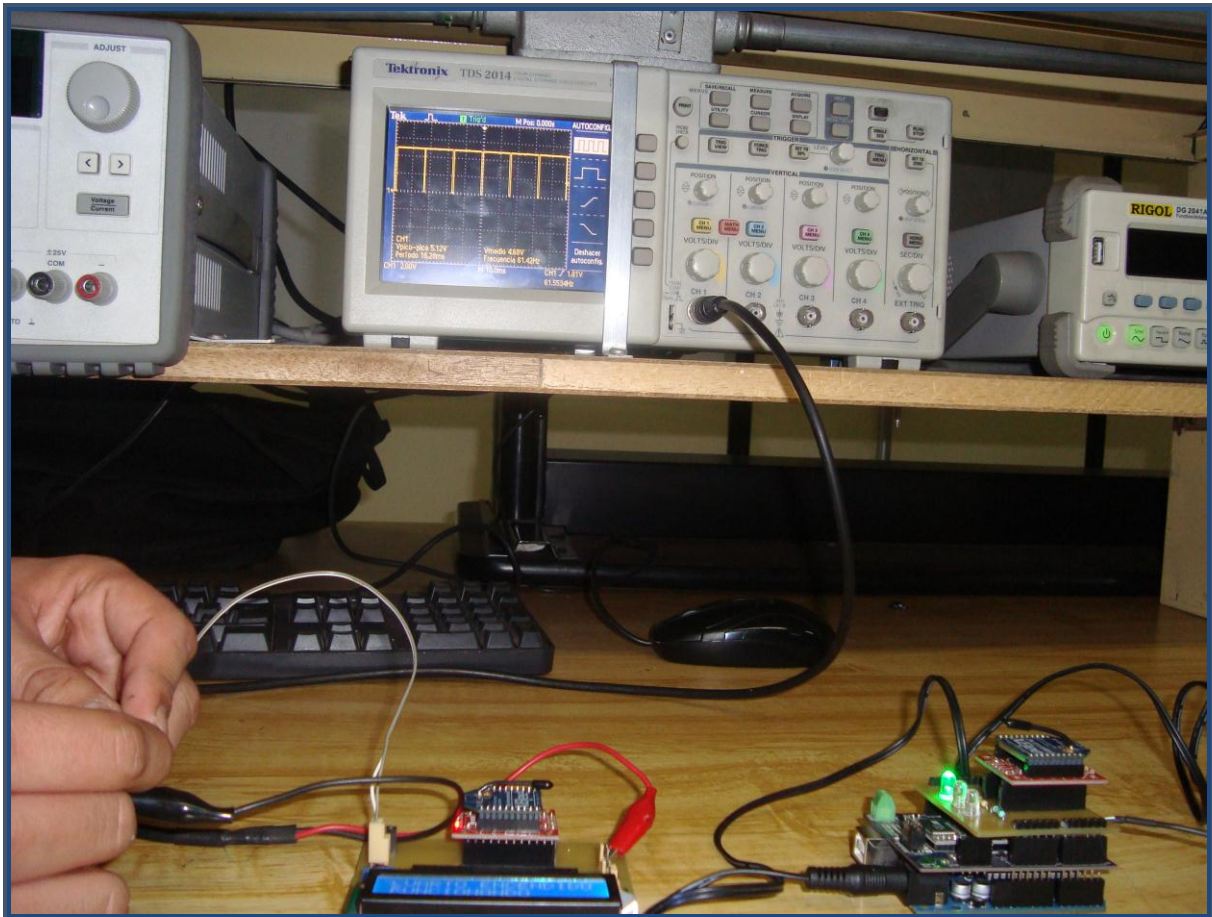


FIGURA 3.10 Se observa las salidas del ATmega1284 para ver el comportamiento de los pulsos PWM.

Se observa que la salida que muestra el osciloscopio es estable, el ancho del pulso cambia si es que se regula el ventilador o algún foco, ya que la intensidad de voltaje es menor que cuando esta encendido en su totalidad. En la figura 3.11 se muestra en el osciloscopio que es estable la salida, proporcionando una confianza y la estabilidad del sistema.

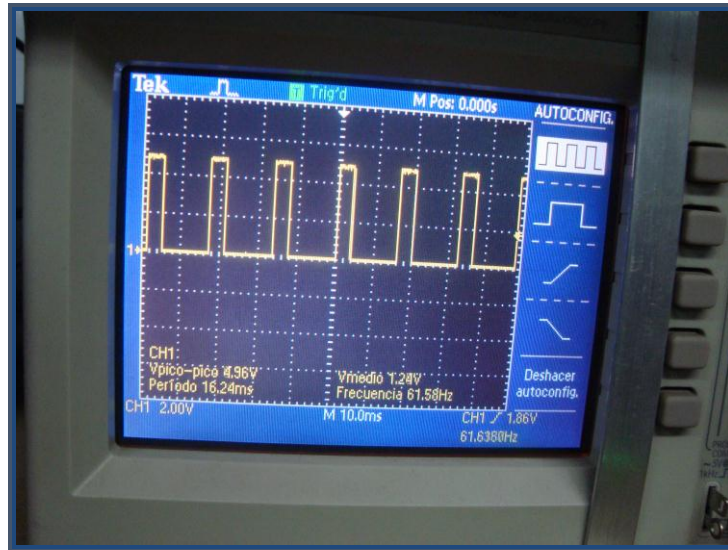


FIGURA 3.11 Se observa la salida PWM en el osciloscopio cuando se regula alguna acción para ver el ancho del pulso.

3.4.1 Modo de inicialización

Una vez se ha conectado el ATmega1284, es capaz de observar la inicialización mostrada en la pantalla del LCD con la palabra "Modulo". Si LCD muestra un mensaje de FUNCIONANDO podemos asegurarnos de que el elemento a controlar esta en óptimas condiciones, si en mensaje es ERROR significa que el foco de la habitación no está encendido. Aquí, se tiene una detección de realimentación para probar si un aparato en casa está en buen estado o no. Así que si el sistema todavía muestre "ERROR", como el circuito está abierto, se notificará al usuario de que algo anda mal con los focos o el ventilado, en la figura 3.12 se observa el mensaje de inicialización en el LCD.



FIGURA 3.12 Mensaje de inicialización.

3.4.2 Encendido, apagado y regulación del ventilador

3.4.2.1 Modo “enciende ventilador”

Este modo se ejecuta con el comando "enciende ventilador". Una vez que la voz menciona "enciende ventilador", es reconocido por el chip de reconocimiento de voz, el comando se envía con un carácter que es "f" a través del Xbee del ATmega328 y recibida por ATmega1284. Una vez que la rutina de servicio de interrupción está encendida, el carácter se almacena en UDR. OCR2A se dirigió a 0 por lo que el ventilador funciona a toda velocidad. Mientras tanto "ventilador apagado" aparece en la pantalla LCD.

3.4.2.2 Modo “regula ventilador”

Este modo se activa por recibir el comando "regula ventilador". Una vez que nuestra voz de "ventilador disminuye" se reconoce por el chip de reconocimiento de voz, el comando se envía con un carácter que es "j" a través del Xbee del ATmega328 y recibida por ATmega1284. Una vez que la rutina de servicio de interrupción está encendida, el carácter se almacena en UDR. OCR2A se dirigió a 100 de manera que el ventilador disminuye la velocidad. Mientras tanto "ventilador regulado" aparece en la pantalla LCD.

3.4.2.3 Modo “apaga ventilador”

Este modo se ejecuta con el comando "apaga ventilador". Una vez que la voz menciona "apaga ventilador", es reconocido por el chip de reconocimiento de voz, el

comando se enviará con un carácter que es "g" a través del Xbee del ATmega328 y recibida por ATmega1284. Una vez que la rutina de servicio de interrupción está encendida, el carácter se almacena en UDR. OCR2A se dirige a 255 por lo que el ventilador se apaga. Mientras tanto, "ventilador apagado" se muestra en la pantalla LCD.

3.4.3 Encendido, apagado, regulación y temporizador del cuarto

3.4.3.1 Modo "regula cuarto"

Este se activa por recibir el comando "regula cuarto". Una vez que la voz menciona "regula cuarto" es reconocido por el chip de reconocimiento de voz, el comando se enviará con un carácter que es "c" a través del Xbee del ATmega328 y recibida por ATmega1284. Una vez que la rutina de servicio de interrupción está encendida, el carácter se almacena en UDR. OCR0A tiene el valor 200, esto implica que el foco se ve más tenue. Mientras tanto el modo "cuarto regulado" aparece en la pantalla LCD.

3.4.3.2 Modo "enciende cuarto"

Este se activa por recibir el comando "enciende cuarto". Una vez que la voz menciona "enciende cuarto" es reconocido por el chip de reconocimiento de voz, el comando se enviará con un carácter que es "a" a través del Xbee del ATmega328 y recibida por ATmega1284.

Una vez que la rutina de servicio de interrupción está encendida, el carácter se almacena en UDR. OCR0A tiene el valor 0, esto implica que el foco se enciende al máximo. Mientras tanto el modo "cuarto encendido" aparece en la pantalla LCD.

3.4.3.3 Modo “apaga cuarto”

Este se activa por recibir el comando "apaga cuarto". Una vez que la voz menciona "apaga cuarto" es reconocido por el chip de reconocimiento de voz, el comando se enviará con un carácter que es "b" a través del Xbee del ATmega328 y recibida por ATmega1284. Una vez que la rutina de servicio de interrupción está encendida, el carácter se almacena en UDR. OCR0A tiene el valor 255, esto implica que el foco se apaga. Mientras tanto el modo "cuarto apagado" aparece en la pantalla LCD.

3.4.3.4 Modo “cuarto en 30 segundos”

Este se activa por recibir el comando "cuarto en 30 segundos". Una vez que la voz menciona "cuarto en 30 segundos" es reconocido por el chip de reconocimiento de voz, el comando se enviará con un carácter que es "i" a través del Xbee del ATmega 328 y recibida por ATmega 1284. Una vez que la rutina de servicio de interrupción está encendida, el carácter se almacena en UDR. Se activa un temporizador de 30 segundos para que el foco se apague en el respectivo tiempo. Mientras tanto "T=30 seg" aparece en la pantalla LCD.

3.4.4 Encendido, apagado y regulación de la sala

3.4.4.1 Modo “enciende sala” y “apaga sala”

La temática de la sala es similar a las anteriores de las de cuarto, para encender la sala se usa el carácter “g” y para apagar la sala se usa el carácter “h”.

Después de guardarse los caracteres en UDR, para el modo “apaga sala” OCR1A tiene el valor 255, esto implica que el foco se apaga. Mientras tanto el modo "sala apagada" aparece en la pantalla LCD.

Para el modo “enciende sala” OCR1A tiene el valor 0, esto implica que el foco se enciende. Mientras tanto el modo "sala encendida" aparece en la pantalla LCD.

3.5 Pruebas en la etapa de potencia

Realizando las pruebas con diodos led`s en las salidas PWM y corroborando las salidas del Atmega1284 sean lo deseado, se puede agregar los circuitos PWM de configuración rápida para poder accionar los focos y el ventilador, repitiendo las mismas pruebas que se realizó con los diodos led. El propósito es corroborar que no se tenga fallas con el encendido y control de estos dispositivos de forma separada y en conjunto como lo muestra la figura 3.13.



FIGURA 3.13 Pruebas de la etapa de recepción con los circuitos PWM.

La primera complicación en la etapa de recepción se refleja al mal armado de los circuitos PWM en el protoboard, ya que un diodo ocasionó un corto circuito al estar mal conectado, y provocó que un foco se funda. A consecuencia de este error se repitió al armado del circuito y en este segundo intento funciono correctamente.

3.6 Pruebas en un edificio y en una casa

Ya asegurado de que el sistema funciona correctamente, se realizaron pruebas de alcance. Primeramente el edificio 3 de la ESIME ZACATENCO se transmitió la información que viaja de Xbee a Xbee, la principal obstrucción de la señal es la pared que separa a los salones, se transmite correctamente la información de un salón a un salón alado sin obtener perdidas, se realiza la misma prueba en un salón más alejado, así hasta llegar a un tercer salón, el cual se empezó a complicar, ya que la perdida fue bastante, no se trasmite adecuadamente.

La distancia aproximada de un salón a 2 salones mas es de 15 m, los obstáculos principales eran 3 paredes, con una atenuación de la señal muy alta, bajo en esas condiciones no será recomendable más distancia.

El propósito de esta prueba es someter al sistema en condiciones difíciles para ver su funcionamiento, ya que el propósito del proyecto no está enfocado a ese tipo de estructuras, podemos decir que es suficiente la transmisión en una casa para que en cualquier punto se pueda transmitir la señal. La pérdida de potencia de la señal es mucho mayor en el edificio 3 por el tipo de material de los salones y las columnas que tiene, en una casa se cuenta con paredes de ladrillo o tabique, la cual esta aplanado por mezcla o yeso que normalmente se emplea, la cual producen mucho menor perdida de la señal obteniendo más alcance de transmisión y recepción de la señal.

La prueba que se realiza en la casa se obtiene mucho más alcance como se espera, como el fabricante lo expresa en las hojas de especificaciones de Xbee que su alcance máximo en una línea de transmisión libre es de 90 m y de 30 m en interiores aproximadamente, se corrobora esa información en el interior de la casa. Tomando en cuenta que la superficie es aproximadamente de 220 m^2 , 22 m de largo por 10 m de ancho como se muestra en la figura 3.14, se pone al Xbee trasmisor en un

extremo de la casa y el Xbee receptor en el otro extremo de la casa, se manda la información y el sistema si puede realizar las tareas solicitadas.

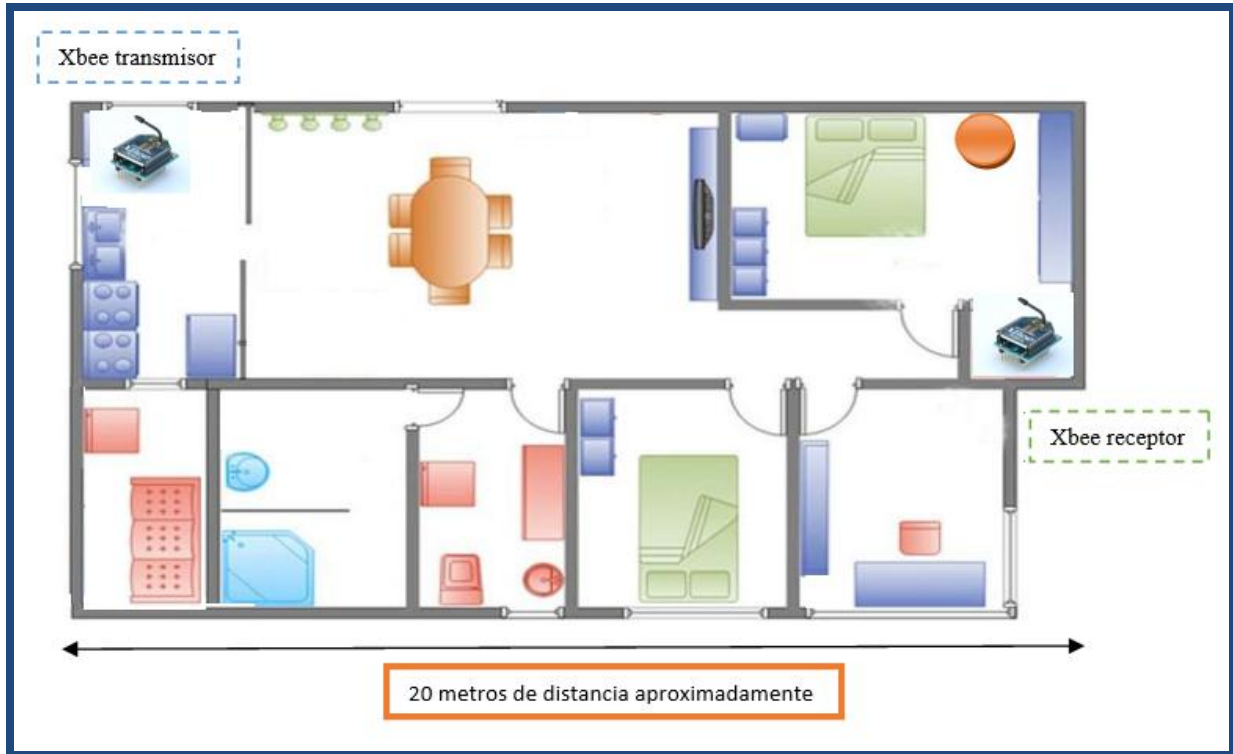


FIGURA 3.14 Transmisión de información de extremo a extremo de la casa.

Para tener menos posibilidad de error a la hora de transmitir la información se puede, en teoría, duplicar la distancia. No es necesario ni eficiente tener el módulo de recepción en un extremo de la casa, es más factible que se coloque en un lugar estratégico, es decir, si se pone el modulo receptor lo más centrado que se pueda de la casa.

Las pruebas que se realizan con el modulo receptor centrado en la casa obtiene más respuesta, e incluso fuera de la casa se pudo transmitir sin ningún problema, ya que el alcance es aproximadamente de 30 m, si lo vemos desde el punto de vista en que el modulo tendrá 30 m de un lado de la casa y 30 metros del otro lado de la casa estaríamos hablando que puede tener un alcance a lo largo de la casa de hasta casi 60 metros bajo esta condición.

La figura 3.15 muestra un ejemplo de cuando una casa rebaza más de 30 m que es la distancia en interiores que alcanza el Xbee tomando de referencia las pruebas ya realizadas, colocando el modulo receptor lo más centrado posible evitamos el problema del alcance en los interiores de una casa, ya que es poco común encontrar casas de más de 50 o 60 m de largo, podemos decir entonces que bajo estas condiciones es suficiente el Xbee para la realización de la transmisión de la información.

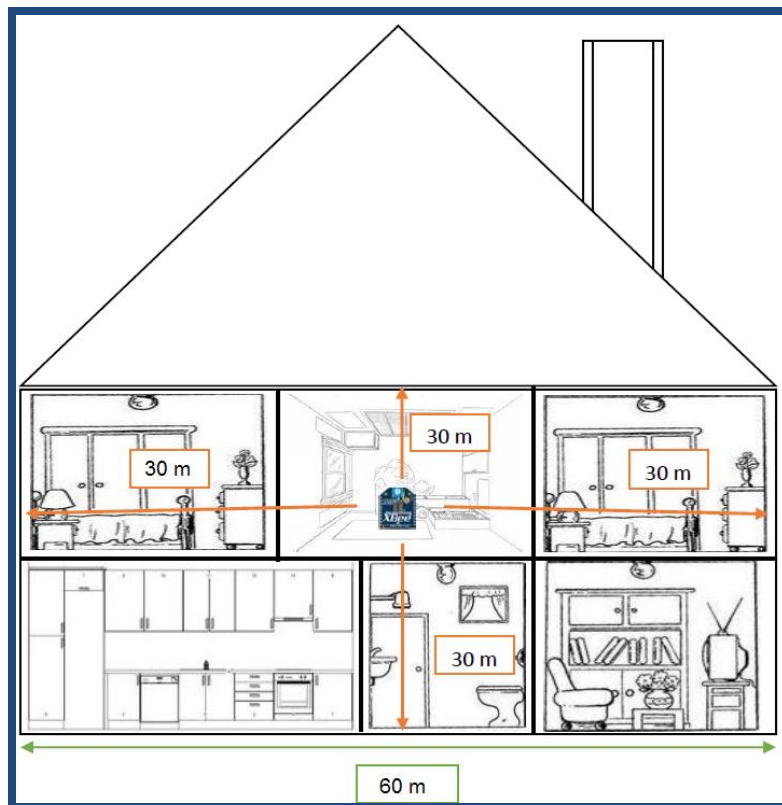


FIGURA 3.15 Bosquejo de alcance de recepción centrando el Xbee receptor en la casa.

Las pruebas sólo se realizan en 2 casas, ya que son las únicas a disposición, aun con la limitante de realizar pruebas en más casas, con la prueba en el edificio la cual tiene como propósito poner al sistema en condiciones más complicadas, podemos suponer que el sistema funcionara en cualquier otra casa con las limitantes antes vistas, y así tener nosotros un sistema de casa inteligente.



Conclusiones

Conclusiones

Se logró diseñar un prototipo de control de distintas variables de una casa habitación a través de un sistema de comunicación inalámbrica.

A través del módulo de voz “EasyVR shield”, utilizando la configuración diseñada para el reconocimiento de voz se permite emitir las ordenes en forma oral de tal manera que los dispositivos a ser controlados respondan de forma adecuada al recibir esas instrucciones, mismas que son transportadas desde el usuario al dispositivo utilizando los Xbee que previamente fueron configurados para esta finalidad.

Los microcontroladores Atmega328 y Atmega1284 son confiables para comunicarse entre sí a través de los módulos XBEE por medio del protocolo “USART” para la transmisión y recepción de comandos de control. Estos microcontroladores tienen una versatilidad que puede ayudar a los dispositivos del proyecto. El Atmega1284 fue utilizado por contar con varias salidas, permitiendo controlar de forma simultánea los dispositivos deseados en el proyecto, el Atmega328 porque interactúa bien con la plataforma de Arduino, siendo este micro suficiente para interactuar con el módulo de voz.

El circuito de potencia permite que las salidas de control del microcontrolador Atmega1284 sean protegidas de la etapa de potencia, además de que los dispositivos a ser controlados tengan buena respuesta a los comandos que emite el usuario.

Recomendaciones:

En caso de que alguien quiera darle continuidad a este prototipo, se pueden tomar en cuenta las siguientes recomendaciones:

Si se desea tener una mayor cobertura en caso de que la instalación del prototipo sea en una casa grande, se pueden utilizar los módulos “XBEE PRO” que tienen un alcance de 90 metros en interiores para mejorar el alcance de la comunicación inalámbrica. Además de que el diseño puede ser implementado usando una red punto-multipunto con los mismos “XBEE”, de tal forma que existan varios módulos centrales que controlen un determinado número de elementos.

El diseño puede ser complementado con un control vía internet, para poder controlar cualquier dispositivo desde cualquier lugar donde se tenga acceso a esta red. También para tener más posibilidades de control además de la voz, se puede diseñar un programa en android para interactuar con el sistema de control desde un teléfono celular o tablet.



Bibliografía

Bibliografía

1. Huidobro Moya, José Manuel. Millán Tejedor, Ramón Jesús. “Domótica. Edificios inteligentes”. Editorial: Limusa Noriega Editores.
2. Romero Morales, Cristóbal. “Domótica e inmótica”. Editorial: Alfaomega Grupo Editor.
3. Romero Morales, Cristóbal. Vázquez Serrano, Francisco. Castro Iozano, Carlos de. “Domótica e inmótica. Viviendas y edificios inteligentes”. Editorial: Alfaomega Grupo Editor.
4. Harke, Werner. “Domótica para viviendas y edificios”. Editorial: Marcombo Ediciones Técnicas.
5. Floyd, Thomas. “Dispositivos electrónicos”. Editorial: Pearson Educación.
6. Boylestad, Robert. Nashelsky, Louis. “Electrónica, teoría de circuitos y dispositivos electrónicos”. Editorial: Prentice Hall.

Páginas de internet

1. www.atmel.com
2. <http://examples.digi.com/get-started/configuring-xbee-radios-with-x-ctu/>
3. <http://www.digi.com/support/productdetail?pid=3352>
4. <http://bildr.org/2011/04/arduino-xbee-wireless/>
5. <http://www.instructables.com/id/Changing-Xbee-Baud-Rates/>
6. http://www.adrirobot.it/EasyVR_Shield/EasyVR_Shield-decrizione_modulo.htm



Glosario

Glosario

Arduino: es un microcontrolador de una placa para el uso de la electrónica en proyectos multidisciplinarios más accesibles. El hardware consiste en una placa de hardware de código abierto diseñado en torno a un microcontrolador Atmel AVR de 8 bits, o un Atmel ARM de 32 bits. El software consiste en un compilador de lenguaje de programación estándar y un gestor de arranque que se ejecuta en el microcontrolador.

Comunicación serial: consiste en el envío de un bit de información de manera secuencial, esto es, un bit a la vez y a un ritmo acordado entre el emisor y el receptor.

Edificios inteligentes: son aquellos que integran cuatro categorías: eficiencia del consumo de energía, sistema de aseguramiento de vidas, sistemas de telecomunicaciones y automatización de áreas de trabajo en un sistema computarizado.

IEEE: corresponde a las siglas de Institute of Electrical and Electronics Engineers (Instituto de Ingenieros Eléctricos y Electrónicos) en español Instituto de Ingenieros Eléctricos y Electrónicos, una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas. Es la mayor asociación internacional sin ánimo de lucro formada por profesionales de las nuevas tecnologías.

Interfaz UART: son las siglas de "Universal Asynchronous Receiver-Transmitter" (en español, Transmisor-Receptor Asíncrono Universal). Éste controla los puertos y dispositivos serie. Se encuentra integrado en la placa base o en la tarjeta adaptadora del dispositivo.

Interrupción vectorizada: es un vector que almacena la dirección de atención a la interrupción. En muchas arquitecturas de computación típicas, los vectores de interrupción se almacenan en una tabla en una zona de memoria, la llamada tabla de vectores de interrupción, de modo que cuando se atiende una petición de interrupción de número “n”, el sistema, tras realizar eventualmente algunas tareas previas (tales como salvar el valor de ciertos registros) transfiere el control a la dirección indicada por el elemento n - ésimo de dicha tabla.

Paridad: procedimiento de verificación basado en que el número de 1 deberá ser siempre par o impar, para cada grupo de bits que se transmitan.

Plataforma: es un sistema que sirve como base para hacer funcionar determinados módulos de hardware o de software con los que es compatible. Dicho sistema está definido por un estándar alrededor del cual se determina una arquitectura de hardware y una plataforma de software (incluyendo entornos de aplicaciones). Al definir plataformas se establecen los tipos de arquitectura, sistema operativo, lenguaje de programación o interfaz de usuario compatibles.

Vivienda domótica: es el conjunto de sistemas capaces de automatizar los aparatos eléctricos, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación y neuronales, cableadas o inalámbricas desde dentro y fuera del hogar.



Anexos

ANEXO I. PROGRAMA PARA EL ATMEGA328

```
//CODIGO PARA ATMEGA 328
```

```
//ESTE PROGRAMA ES PARA RECONOCIMIENTO DE VOZ Y TRANSMISION //DE  
COMANDOS
```

```
#include "Arduino.h"
```

```
#include "SoftwareSerial.h"
```

```
SoftwareSerial port(12,13);
```

```
#include "EasyVR.h"
```

```
EasyVR easyvr(port);
```

```
//GRUPOS Y COMANDOS
```

```
enum Groups
```

```
{
```

```
    GROUP_0 = 0,
```

```
    GROUP_1 = 1,
```

```
};
```

```
enum Group0
```

```
{
```

```
    G0_SMART = 0,
```

```
};
```

```
enum Group1
```

```
{
```

```
    G1_ENCIENDE_CUARTO = 0,
```

```
    G1_APAGA_CUARTO = 1,
```

```
    G1_REGULA_CUARTO = 2,
```

```
    G1_ENCIENDE_VEN = 3,
```

```
    G1_APAGA_VEN = 4,
```

```
    G1_DISMINUYE_VEN = 5,
```

```
    G1_ENCIENDE_SALA = 6,
```



```
G1_APAGA_SALA = 7,
G1_CUARTO_30_SEG = 8,
};
EasyVRBridge bridge;
int8_t group, idx;
void setup()
{
  // MODO BRIDGE?
  if (bridge.check())
  {
    cli();
    bridge.loop(0, 1, 12, 13);
  }
  // PROCESO NORMAL
  Serial.begin(9600);
  port.begin(9600);
  if (!easyvr.detect())
  {
    //Serial.println("EasyVR not detected!");
    for (;;)
  }
  easyvr.setPinOutput(EasyVR::IO1, LOW);
  //Serial.println("EasyVR detected!");
  easyvr.setTimeout(5);
  easyvr.setLanguage(5);
  group = EasyVR::TRIGGER;
  pinMode(11, OUTPUT);
  digitalWrite(11, LOW); // LED APAGADO
}
void action();
```

```
void loop()
{
  easyvr.setPinOutput(EasyVR::IO1, HIGH); // LED ENCENDIDO (PUEDE HABLAR
  EL USUARIO)
  //Serial.print("Say a command in Group");
  //Serial.println(group);
  easyvr.recognizeCommand(group);
do
{
  // SE PUEDE HACER ALGO EN LA ESPERA DE UN COMANDO DE VOZ
}
while (!easyvr.hasFinished());
easyvr.setPinOutput(EasyVR::IO1, LOW); // LED APAGADO
idx = easyvr.getWord();
if (idx >= 0)
{
  // built-in trigger (ROBOT)
  // group = GROUP_X; &lt;-- jump to another group X
  return;
}
idx = easyvr.getCommand();
if (idx >= 0)
{
  // PARA IMPRIMIR EL COMANDO
  uint8_t train = 0;
  char name[32];
  //Serial.print("Command: ");
  //Serial.print(idx);
  if (easyvr.dumpCommand(group, idx, name, train))
  {
```

```
// Serial.print(" = ");
//Serial.println(name);
}
else
// Serial.println();
easyvr.playSound(0, EasyVR::VOL_FULL);
// REALIZAR ALGUNA ACCION
action();
}
else // ERRORES O TIEMPO DE ESPERA AGOTADO
{
/*
if (easyvr.isTimeout())
//Serial.println("Timed out, try again...");
int16_t err = easyvr.getError();
if (err >= 0)
{
// Serial.print("Error ");
// Serial.println(err, HEX);
}*/
group = GROUP_0;
}
}
void action()
{
switch (group)
{
case GROUP_0:
switch (idx)
{
```

```
    case G0_SMART:
        group = GROUP_1;
        break;
    }
    break;
case GROUP_1:
    switch (idx)
    {
    case G1_ENCIENDE_CUARTO:
        group = GROUP_0;
        Serial.print('a');
        break;
    case G1_APAGA_CUARTO:
        group = GROUP_0;
        Serial.print('b');
        break;
    case G1_REGULA_CUARTO:
        group = GROUP_0;
        Serial.print('c');
        break;
    case G1_ENCIENDE_SALA:
        group = GROUP_0;
        Serial.print('d');
        break;
    case G1_APAGA_SALA:
        group = GROUP_0;
        Serial.print('e');
        break;
    case G1_ENCIENDE_VEN:
        group = GROUP_0;
```

```
Serial.print('f');
break;
case G1_APAGA_VEN:
group = GROUP_0
Serial.print('g');
break;
case G1_CUARTO_30_SEG:
group = GROUP_0;
Serial.print('i');
break;

case G1_DISMINUYE_VEN:
group = GROUP_0
Serial.print('j');
break;
}
break;
}
```

ANEXO II. PROGRAMA PARA EL ATMEGA1284

```
/*
 * Smart_atmega1284.cpp
 * Funcion: Receptor inalambrico, control de luces y ventilador
 */
#include <inttypes.h>
#include <util/delay.h>
#include <avr/io.h>
#include <avr/interrupt.h>
#include "lcd_lib.h"
#include <avr/pgmspace.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define BAUDRATE 9600
#define UBRRVAL ((F_CPU/(BAUDRATE*16UL))-1)
#define USART_DDR    DDRD
#define USART_PORT    PORTD
#define USART_RX_PIN  2
#define USART_TX_PIN  3
#define t1 61
//variables auxiliares en el programa
char lcd_buffer[17]; // buffer del display LCD
volatile unsigned int lcdtime=0;
volatile unsigned int time0,time1=0;
volatile char index='z';
volatile unsigned int pinvalue=0;
//*****
// Configuracion del LCD
void init_lcd(void)
```

```
{
    MCUCR = (1<<JTD); //Desactiva interfaz JTAG para poder
    MCUCR = (1<<JTD); //usar el Puerto C como I/O para el display
    LCDinit(); //inicializacion del display
    LCDcursorOFF();
    LCDclr(); //Limpia la pantalla
    LCDGotoXY(4,0);
}
//*****
//Interrupcion para timer/counter 0 igualdad en comparacion A
ISR (TIMER0_COMPA_vect)
{
    //Actualizar el tiempo de la tarea
    if (time1>0) --time1;
}
//*****
// Task 1 escribe en el LCD cada segundo
void task1 (void)
{
    if(lcdtime>0)
    {
        lcdtime-=1;
    }
    else
    {
        lcdtime=0;
        if (index=='i')
        {
            OCR0A=255;
            index='z';
        }
    }
}
```

```
    }
}
//Deteccion de fallos
pinvalue=ADCH*0.96/5.00;
ADCSRA|=(1<<ADSC);
if(pinvalue<2.00) // Prueba
{
    LCDGotoXY(4,1);
    LCDstring("Error",strlen("Error"));
}
if(pinvalue>2.00)
{
    LCDGotoXY(4,1);
    LCDstring("FUNCIONANDO",strlen("FUNCIONANDO"));
}
}
//*****
void USART_Init(void)
{
    USART_DDR|=((1<<USART_TX_PIN)&~(1<<USART_RX_PIN));//0x00000010 Tx
    salida Rx entrada
    USART_PORT = 0X00;
//Configurar tasa de transmision de datos
    UBRR1L=(F_CPU/16/(BAUDRATE+1))%256;
    UBRR1H=(F_CPU/16/(BAUDRATE+1))/256;
//Ajustar el formato de la trama de datos:Modo asincrono ,no paridad, 1 bit de parada,
8 bits
    UCSR1C=(0<<UMSEL11)|(0<<UMSEL10)|(0<<UPM11)|(0<<UPM10)|
(0<<USBS1)|(0<<UCSZ12)|(1<<UCSZ11)|(1<<UCSZ10)|(0<<UCPOL1);
//Habilitar receptor e interrupcion en recepcion completa
```



```
UCSR1B=(1<<RXEN1)|(1<<RXCIE1)|(1<<TXEN1);
}
//*****
uint8_t USART_vReceiveByte(void)
{
    // Esperar hasta que un byte sea recibido
    while((UCSR1A&(1<<RXC1)) == 0);
    // regresa el dato recibido
    return UDR1;
}
ISR(USART1_RX_vect)
{
    UDR1=0;//limpia el buffer del UART
    uint8_t data;//unsigned char
    data=USART_vReceiveByte();//dato recibido
    //OCR0A--P3 Cuarto---"a" encendido/"b"apagado/"c"regulador
    //OCR0B--P4 Sala---"d"encendido/"e"apagado
    //OCR2A--D7 Ventilador ---"f"encendido/"g"apagado/"h"regulador
    LCDGotoXY(4,0);
    data-=64;
    //FOCO DEL CUARTO ENCENDIDO
    if(data=='a')
    {
        OCR0A = 10;
        //LCD muestra "CUARTO ENCENDIDO"
        LCDGotoXY(4,0);
        LCDstring("CUARTO ENCENDIDO",strlen("CUARTO ENCENDIDO"));

        LCDGotoXY(4,1);
        LCDstring(" ",strlen(" "));
    }
}
```

```
}  
//FOCO DEL CUARTO APAGADO  
if(data=='b')  
{  
    OCR0A = 255;  
    LCDGotoXY(4,0);  
    LCDstring("CUARTO APAGADO",strlen("CUARTO APAGADO"));  
    LCDGotoXY(4,1);  
    LCDstring(" ",strlen(" "));  
}  
//REGULADOR DEL FOCO DEL CUARTO  
if(data=='c')  
{  
    OCR0A = 192;  
    LCDGotoXY(4,0);  
    LCDstring("FOCO REGULADO",strlen("FOCO REGULADO"));  
    LCDGotoXY(4,1);  
    LCDstring(" ",strlen(" "));  
}  
//FOCO DE LA SALA ENCENDIDA  
if(data=='d')  
{  
    OCR0B = 0;  
    LCDGotoXY(4,0);  
    LCDstring("SALA ENCENDIDA",strlen("SALA ENCENDIDA"));  
    LCDGotoXY(4,1);  
    LCDstring(" ",strlen(" "));  
}  
//FOCO DE LA SALA DESACTIVADO  
if(data=='e')
```

```
{
  OCR0B = 255;
  LCDGotoXY(4,0);
  LCDstring("SALA APAGADA ",strlen("SALA APAGADA "));
  LCDGotoXY(4,1);
  LCDstring(" ",strlen(" "));
}

//VENTILADOR ENCENDIDO
if(data=='f')
{
  OCR2A = 0;
  LCDGotoXY(4,0);
  LCDstring("VENTILADOR ON ",strlen("VENTILADOR ON "));
  LCDGotoXY(4,1);
  LCDstring(" ",strlen(" "));
}

//BAJA VELOCIDAD DEL VENTILADOR
if(data=='j')
{
  OCR2A = 192;
  LCDGotoXY(4,0);
  LCDstring("VENTILADOR REGULADO ",strlen("VENTILADOR REGULADO "));
  LCDGotoXY(4,1);
  LCDstring(" ",strlen(" "));
}

//APAGA VENTILADOR
if(data=='g')
{
  OCR2A = 255;
  LCDGotoXY(4,0);
```

```
LCDstring("VENTILADOR OFF ",strlen("VENTILADOR OFF "));
LCDGotoXY(4,1);
LCDstring(" ",strlen(" "));
}
//MODO DE TIEMPO PARA EL FOCO DEL CUARTO
if(data=='i')
{
OCR0A=10;
lcdtime=30;
index=data;
//LCD MOSTRARA EL TIEMPO
LCDGotoXY(4,0);
LCDstring("T=",strlen("T="));
sprintf(lcd_buffer,"%d",lcdtime);
LCDGotoXY(6, 0);
// tiempo
LCDstring(lcd_buffer, strlen(lcd_buffer));
LCDGotoXY(6+strlen(lcd_buffer),0);
LCDstring("sec ", strlen("sec "));
}
}
//*****
int main(void)
{
// Inicializa el LCD
init_lcd();
LCDGotoXY(4,0);
LCDstring("SMART HOME ",strlen("SMART HOME "));
ADMUX=(1<<ADLAR)|(1<<REFS0);//Ajuste por la izquierda y activa la referencia
interna
```

```
ADCSRA=(1<<ADEN)|(1<<ADSC)+7;//Activa el ADC y comienza la conversion,  
configura el preescalar a 1MHz
```

```
PORTA=0x00;
```

```
DDRA=0x00;
```

```
//*****
```

```
//Configura el timer 0 para instantes de 1 mSec
```

```
TIMSK0 = 2;           //Activa timer 0 cmp con ISR
```

```
OCR0A = 249; //Registro comparador a 250 instantes
```

```
TCCR0A = 0b00000010; // Enciende clear-on-match
```

```
TCCR0B = 0b00000011; // clock prescalar a 64 clk/O/64 (del prescaler)
```

```
//*****
```

```
//pwm
```

```
DDRB = (1<<PINB3)|(1<<PINB4);
```

```
DDRD = (1<<PINB6)|(1<<PINB7);
```

```
// timer 0 funciona a velocidad maxima
```

```
TCCR0B = 3 ; //clk/O/64 (From prescaler)=011
```

```
TCCR2B = 3; //clkT2S/64 (From prescaler)=100 /32=011
```

```
// Enciende el PWM
```

```
// Enciende PWM rapida and salida del OC0A
```

```
// a max velocidad de reloj, cambia OC0A (pin B3)
```

```
// 16 microsec para ciclos PWM
```

```
TCCR0A = (1<<COM0A0) | (1<<COM0A1) | (1<<WGM00) | (1<<WGM01)  
|(1<<COM0B0) | (1<<COM0B1);
```

```
OCR0A =255; //P3
```

```
OCR0B =255; //p4
```

```
TCCR2A = (1<<COM2A0) | (1<<COM2A1) | (1<<WGM20) | (1<<WGM21)  
|(1<<COM2B0) | (1<<COM2B1);
```

```
OCR2A =255; //D7
```

```
OCR2B =255; //D6
```

```
//*****
```

```
sei();//activa interrupciones globales
USART_Init();
while(1)
{
  if (time1==0){ time1=t1; task1();}
}
}
```

**ANEXO III. HOJAS DE ESPECIFICACIÓN DEL MICROCONTROLADOR
ATMEGA328**

Features

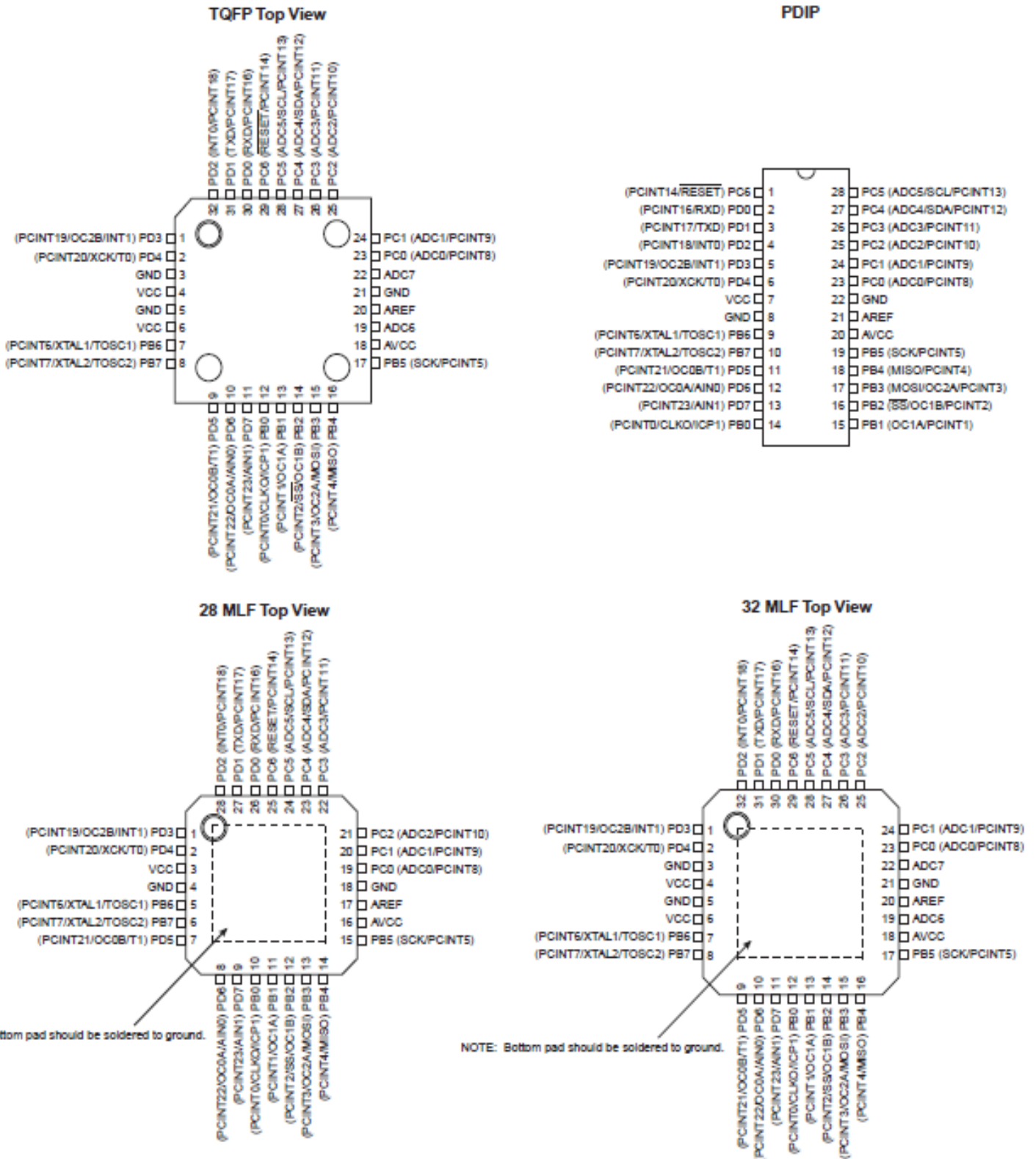
- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48PA/88PA/168PA/328P)
 - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
 - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
 - Active Mode: 0.2 mA
 - Power-down Mode: 0.1 µA
 - Power-save Mode: 0.75 µA (Including 32 kHz RTC)



**8-bit AVR®
Microcontroller
with 4/8/16/32K
Bytes In-System
Programmable
Flash**

**ATmega48PA
ATmega88PA
ATmega168PA
ATmega328P**

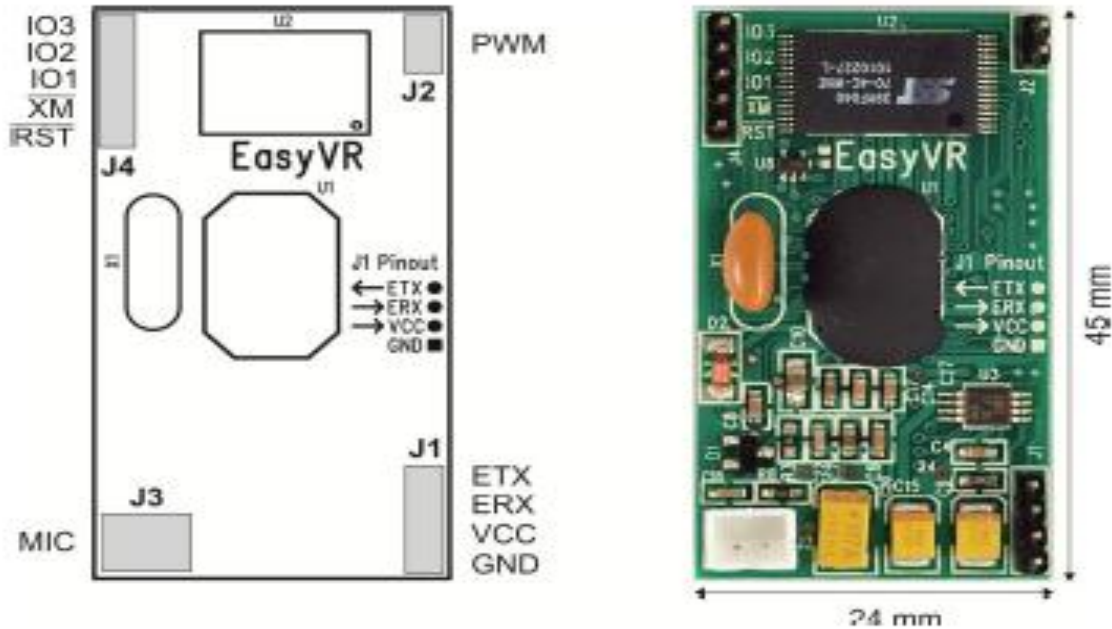
Figure 1-1. Pinout ATmega48PA/88PA/168PA/328P



ANEXO IV. HOJAS DE ESPECIFICACIÓN DEL MÓDULO DE VOZ

Technical specifications

Physical dimensions and pin assignment



Connector	Number	Name	Type	Description
J1	1	GND	-	Ground
	2	VCC	I	Voltage DC input
	3	ERX	I	Serial Data Receive (TTL level)
	4	ETX	O	Serial Data Transmit (TTL level)
J2	1-2	PWM	O	Differential audio output (can directly drive 8Ω speaker)
J3	1	MIC_RET	-	Microphone reference ground
	2	MIC_IN	I	Microphone input signal
J4	1	/RST	I	Active low asynchronous reset (internal 100K pull-up)
	2	/XM	I	Boot select (internal 1K pull-down)
	3	IO1	I/O	General purpose I/O (3.0 VDC TTL level)
	4	IO2	I/O	General purpose I/O (3.0 VDC TTL level)
	5	IO3	I/O	General purpose I/O (3.0 VDC TTL level)

Note: the GPIO (J4.3, J4.4, and J4.5) are at nominal 3.0VDC level. Do not connect 5VDC directly to these pins!

Recommended Operating Conditions

Symbol	Parameter	Min	Typ	Max	Unit
VCC	Voltage DC Input	3.3	5.0	5.5	V
Ta	Ambient Operating Temperature Range	0	25	70	°C
ERX	Serial Port Receive Data	0	-	VCC	V
ETX	Serial Port Transmit Data	0	-	VCC	V

Electrical Characteristics

These are applicable to J4 pins only, including IO1-3, /XM and /RST.

Symbol	Parameter	Min	Typ	Max	Unit
V _{IH}	Input High Voltage	2.4	3.0	3.3	V
V _{IL}	Input Low Voltage	-0.1	0.0	0.75	V
I _{IL}	Input Leakage Current (0 < V _{IO} < 3V, Hi-Z Input)		<1	10	µA
R _{PU}	Pull-up Resistance	Strong	10		kΩ
		Weak	200		kΩ
V _{OH}	Output High Voltage (I _{OH} = -5 mA)	2.4			V
V _{OL}	Output Low Voltage (I _{OL} = 8 mA)			0.6	V

Power Supply Requirements

Symbol	Parameter	Min	Typ	Max	Unit
I _{sleep}	Sleep current		< 1		mA
I _{oper}	Operating current		12		mA
I _{speaker}	Audio playback current (with 8ohm speaker)		180		mA _(RMS)

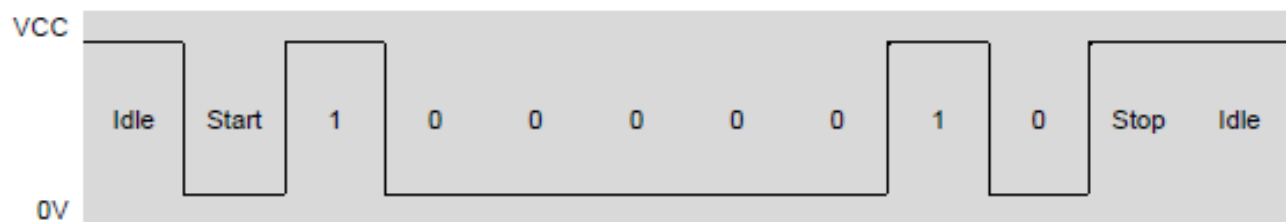
Serial Interface

The EasyVR is a "slave" module communicating via an asynchronous serial interface (commonly known as UART interface), with the following features:

- Baud Rate: 9600 (default), 19200, 38700, 57600, 115200
- Frame: 8 Data bits, No parity, 1 Stop bit

The receiver input data line is ERX, while the transmitter output data line is ETX. No handshake lines are used.

Example of a serial data frame representing character "A" (decimal 65 or hexadecimal 41):



See also chapter [COMMUNICATION PROTOCOL](#) later on this manual for communication details.

ANEXO V. HOJAS DE ESPECIFICACIÓN DEL MICRÓFONO

Microphone

The microphone provided with the EasyVR module is an omnidirectional electret condenser microphone (Horn EM9745P-382):

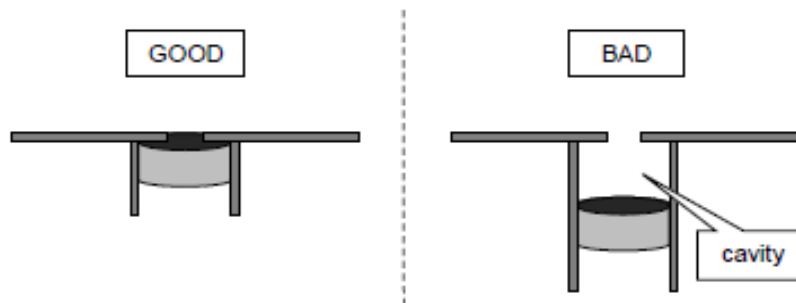
- Sensitivity -38dB (0dB=1V/Pa @1KHz)
- Load Impedance 2.2K
- Operating Voltage 3V
- Almost flat frequency response in range 100Hz – 20kHz

If you use a microphone with different specifications the recognition accuracy may be adversely affected. No other kind of microphone is supported by the EasyVR.

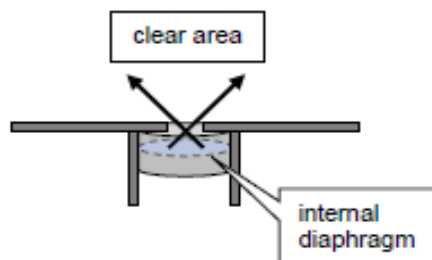
Note: Vocal commands should be given from about 60cm from the microphone, but you can try at greater distances by talking louder.

Please note that improper acoustic positioning of the microphone will reduce recognition accuracy. Many mechanical arrangements are possible for the microphone element, and some will work better than others. When mounting the microphone in the final device, keep in mind the following guidelines:

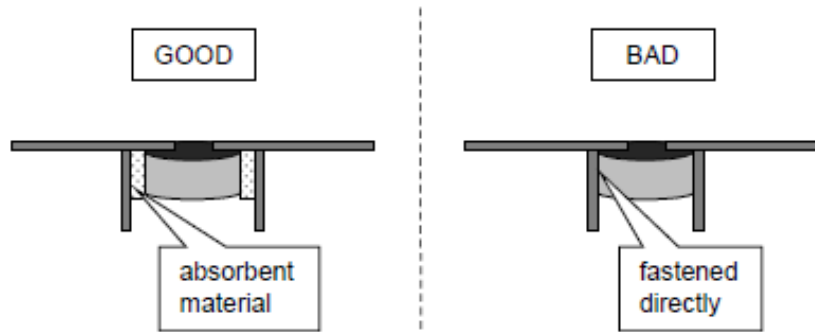
1. **Flush Mounting** - The microphone element should be positioned as close to the mounting surface as possible and should be fully seated in the plastic housing. There must be no airspace between the microphone element and the housing. Having such an airspace can lead to acoustic resonance, which can reduce recognition accuracy.



2. **No Obstructions, Large Hole** - The area in front of the microphone element must be kept clear of obstructions to avoid interference with recognition. The diameter of the hole in the housing in front of the microphone should be at least 5 mm. Any necessary plastic surface in front of the microphone should be as thin as possible, being no more than 0.7 mm, if possible.



3. **Insulation** - The microphone should be acoustically isolated from the housing if possible. This can be accomplished by surrounding the microphone element with a spongy material such as rubber or foam. The provided microphone has this kind of insulating foam. The purpose is to prevent auditory noises produced by handling or jarring the device from being "picked up" by the microphone. Such extraneous noises can reduce recognition accuracy.



4. **Distance** - If the microphone is moved from 15 cm to 30 cm from the speaker's mouth, the signal power decreases by a factor of four. The difference between a loud and a soft voice can also be more than a factor of four. Although the internal preamplifier of the EasyVR compensates for a wide dynamic range of input signal strength, if its range is exceeded, the user application can provide feedback to the speaker about the voice volume (see appendix [ERROR CODES](#)).

Audio Output

The EasyVR audio output interface is capable of directly driving an 8Ω speaker. It could also be connected to an external audio amplifier to drive lower impedance loudspeakers.

Note: Connecting speakers with a lower impedance directly to the module may permanently damage the EasyVR audio output or the whole module.

It is possible to connect higher impedance loads such as headphones, provided that you scale down the output power according to the speaker ratings, for example using a series resistor. The exact resistor value depends on the headphone power ratings and the desired output volume (usually in the order of $10k\Omega$).

Note: Connecting headphone speakers directly to the EasyVR audio output may damage your hearing.

General Purpose I/O

Since the EasyVR communication interface takes two pins of the host controller, a few spare I/O pins are provided, that can be controlled with the communication protocol, to get those pins back for basic tasks, such as lighting an LED.

The three I/O pins IO1–IO3 are connected directly to the embedded microcontroller on the EasyVR module, so they are referenced to the internal 3.0V regulated power supply. If you need to interface to circuits using a different supply, there is a number of solutions you can adopt. Some of these are outlined below (here IO_n indicates any one of the three I/O pins of the EasyVR).

Use a pin as an output

All the I/O pins are inputs with weak internal pull-up after power on. You must explicitly configure a pin before you can use it as an output (see the example code [USE GENERAL PURPOSE I/O PINS](#)).

ANEXO VI. HOJAS DE ESPECIFICACIÓN DEL XBEE

1. XBee®/XBee-PRO® RF Modules

The XBee and XBee-PRO RF Modules were engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices.

The modules operate within the ISM 2.4 GHz frequency band and are pin-for-pin compatible with each other.



Key Features

Long Range Data Integrity

XBee

- Indoor/Urban: up to 100' (30 m)
- Outdoor line-of-sight: up to 300' (90 m)
- Transmit Power: 1 mW (0 dBm)
- Receiver Sensitivity: -92 dBm

XBee-PRO

- Indoor/Urban: up to 300' (90 m), 200' (60 m) for International variant
- Outdoor line-of-sight: up to 1 mile (1600 m), 2500' (750 m) for International variant
- Transmit Power: 63mW (18dBm), 10mW (10dBm) for International variant
- Receiver Sensitivity: -100 dBm

RF Data Rate: 250,000 bps

Advanced Networking & Security

- Retries and Acknowledgements
- DSSS (Direct Sequence Spread Spectrum)
- Each direct sequence channels has over 65,000 unique network addresses available
- Source/Destination Addressing
- Unicast & Broadcast Communications
- Point-to-point, point-to-multipoint and peer-to-peer topologies supported

Low Power

XBee

- TX Peak Current: 45 mA (@3.3 V)
- RX Current: 50 mA (@3.3 V)
- Power-down Current: < 10 µA

XBee-PRO

- TX Peak Current: 250mA (150mA for international variant)
- TX Peak Current (RPSMA module only): 340mA (180mA for international variant)
- RX Current: 55 mA (@3.3 V)
- Power-down Current: < 10 µA

ADC and I/O line support

- Analog-to-digital conversion, Digital I/O
- I/O Line Passing

Easy-to-Use

- No configuration necessary for out-of box RF communications
- Free X-CTU Software (Testing and configuration software)
- AT and API Command Modes for configuring module parameters
- Extensive command set
- Small form factor

Worldwide Acceptance

FCC Approval (USA) Refer to Appendix A [p64] for FCC Requirements. Systems that contain XBee®/XBee-PRO® RF Modules inherit Digi Certifications.

ISM (Industrial, Scientific & Medical) 2.4 GHz frequency band

Manufactured under **ISO 9001:2000** registered standards

XBee®/XBee-PRO® RF Modules are optimized for use in the United States, Canada, Australia, Japan, and Europe. Contact Digi for complete list of government agency approvals.

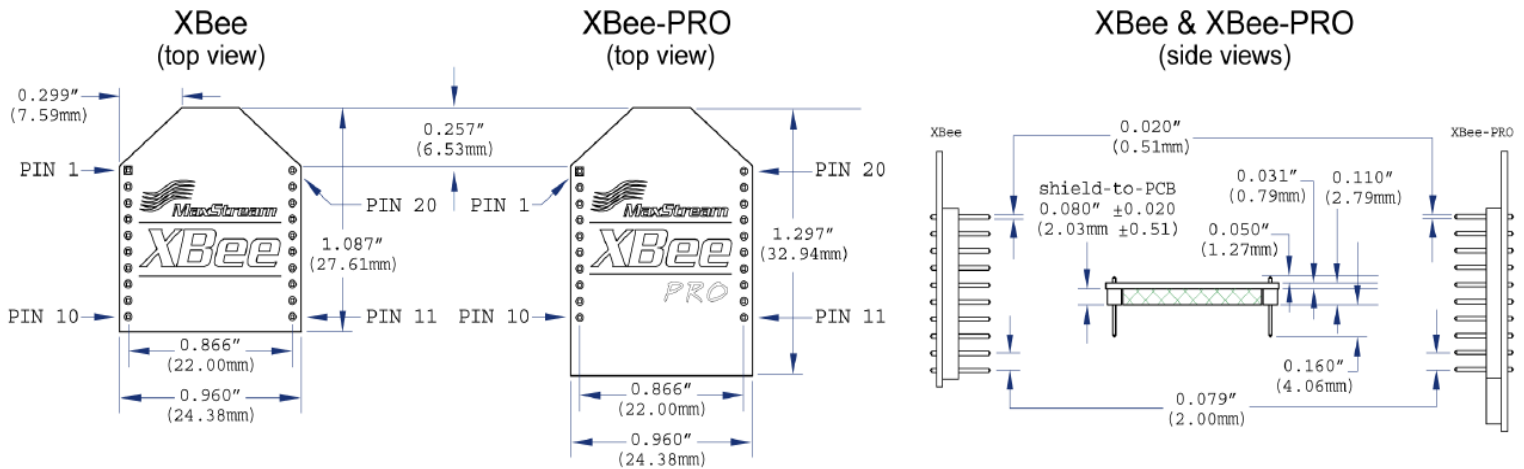


Specifications

Table 1-01. Specifications of the XBee®/XBee-PRO® RF Modules

Specification	XBee	XBee-PRO
Performance		
Indoor/Urban Range	Up to 100 ft (30 m)	Up to 300 ft. (90 m), up to 200 ft (60 m) International variant
Outdoor RF line-of-sight Range	Up to 300 ft (90 m)	Up to 1 mile (1600 m), up to 2500 ft (750 m) international variant
Transmit Power Output (software selectable)	1mW (0 dBm)	63mW (18dBm)* 10mW (10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
Power Requirements		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	250mA (@3.3 V) (150mA for international variant) RPSMA module only: 340mA (@3.3 V) (180mA for international variant)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 μ A	< 10 μ A
General		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector, RPSMA Connector	Integrated Whip, Chip or U.FL Connector, RPSMA Connector
Networking & Security		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses
Agency Approvals		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max. 10 dBm transmit power output)*
Japan	R201WW07215214	R201WW08215111 (Max. 10 dBm transmit power output)*
Australia	C-Tick	C-Tick

The XBee and XBee-PRO RF Modules are pin-for-pin compatible.



ignals

Figure 1-03. XBee®/XBee-PRO® RF Module Pin Numbers

(top sides shown - shields on bottom)

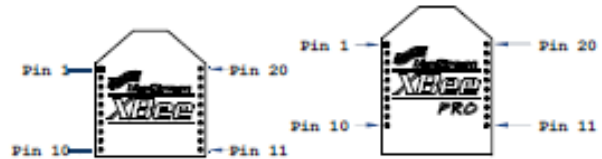


Table 1-02. Pin Assignments for the XBee and XBee-PRO Modules

(Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	D08*	Output	Digital Output 8
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

ANEXO VII. HOJAS DE ESPECIFICACIÓN DEL MICROCONTROLADOR ATMEGA1284

Features

- High-performance, Low-power AVR[®] 8-bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
 - 128K Bytes of In-System Self-Programmable Flash
Endurance: 10,000 Write/Erase Cycles
 - Optional Boot Code Section with Independent Lock Bits
In-System Programming by On-chip Boot Program
True Read-While-Write Operation
 - 4K Bytes EEPROM
Endurance: 100,000 Write/Erase Cycles
 - 16K Bytes Internal SRAM
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - Two 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel, 10-bit ADC
Differential mode with selectable gain at 1x, 10x or 200x
 - Byte-oriented Two-wire Serial Interface
 - Two Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
 - 1.8 - 5.5V for ATmega1284P
- Speed Grades
 - 0 - 4 MHz @ 1.8 - 5.5V
 - 0 - 10 MHz @ 2.7 - 5.5V
 - 0 - 20 MHz @ 4.5 - 5.5V
- Power Consumption at 1 MHz, 1.8V, 25°C
 - Active: 0.4 mA
 - Power-down Mode: 0.1 μ A
 - Power-save Mode: 0.7 μ A (Including 32 kHz RTC)



8-bit AVR[®]
Microcontroller
with 128K Bytes
In-System
Programmable
Flash

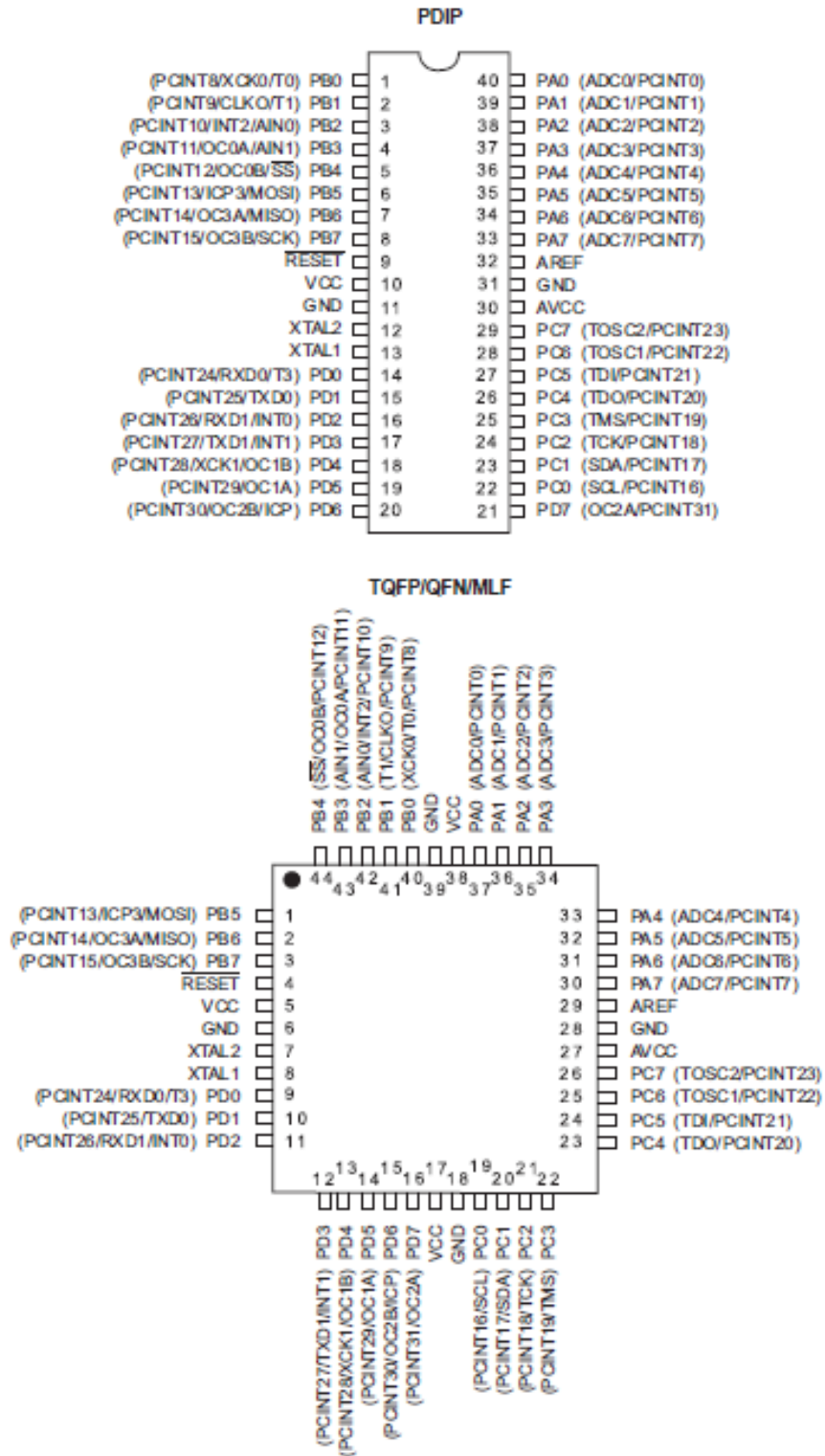
ATmega1284P

Preliminary

Summary

1. Pin Configurations

Figure 1-1. Pinout ATmega1284P



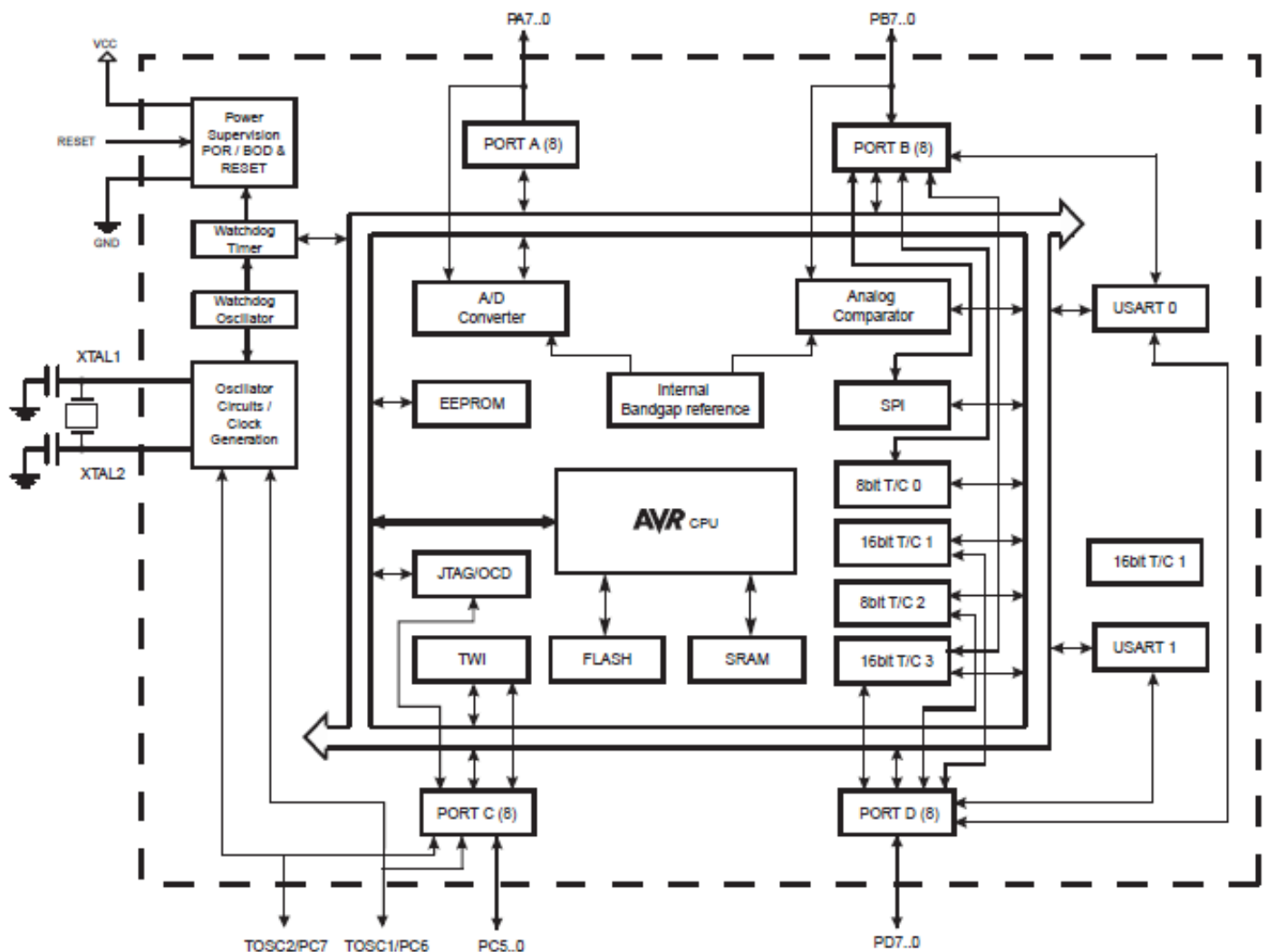
Note: The large center pad underneath the QFN/MLF package should be soldered to ground on the board to ensure good mechanical stability.

2. Overview

The ATmega1284P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega1284P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega1284P provides the following features: 128K bytes of In-System Programmable Flash with Read-While-Write capabilities, 4K bytes EEPROM, 16K bytes SRAM, 32 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), three flexible Timer/Counters with compare modes and PWM, 2 USARTs, a byte oriented 2-wire Serial Interface, a 8-channel, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega1284P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega1284P AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

2.2 Pin Descriptions

2.2.1 VCC

Digital supply voltage.

2.2.2 GND

Ground.

2.2.3 Port A (PA7:PA0)

Port A serves as analog inputs to the Analog-to-digital Converter.

Port A also serves as an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

2.2.4 Port B (PB7:PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega1284P as listed on

2.2.5 Port C (PC7:PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port C also serves the functions of the JTAG interface, along with special features of the ATmega1284.

2.2.6 Port D (PD7:PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega1284.

2.2.7 $\overline{\text{RESET}}$

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running.

2.2.8 XTAL1

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

2.2.9 XTAL2

Output from the inverting Oscillator amplifier.

2.2.10 AVCC

AVCC is the supply voltage pin for Port F and the Analog-to-digital Converter. It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.

2.2.11 AREF

This is the analog reference pin for the Analog-to-digital Converter.

ANEXO VIII. HOJAS DE ESPECIACIÓN DEL OPTOACOPLADOR 4N35

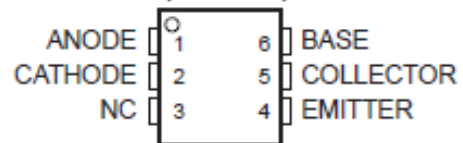
4N35, 4N36, 4N37
OPTOCOPLERS

SOES021C – NOVEMBER 1981 – REVISED APRIL 1998

COMPATIBLE WITH STANDARD TTL INTEGRATED CIRCUITS

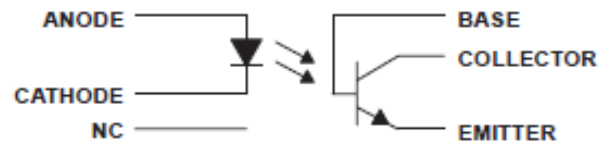
- Gallium-Arsenide-Diode Infrared Source
Optically Coupled to a Silicon npn Phototransistor
- High Direct-Current Transfer Ratio
- High-Voltage Electrical Isolation
1.5-kV, 2.5-kV, or 3.55-kV Rating
- High-Speed Switching
 $t_r = 7 \mu s$, $t_f = 7 \mu s$ Typical
- Typical Applications Include Remote Terminal Isolation, SCR and Triac Triggers, Mechanical Relays and Pulse Transformers
- Safety Regulatory Approval
UL/CUL, File No. E65085

DCJT OR 6-TERMINAL DUAL-IN-LINE PACKAGE
(TOP VIEW)



†4N35 only
NC – No internal connection

schematic



absolute maximum ratings at 25°C free-air temperature (unless otherwise noted)†

Input-to-output peak voltage (8-ms half sine wave):	4N35	3.55 kV
	4N36	2.5 kV
	4N37	1.5 kV
Input-to-output root-mean-square voltage (8-ms half sine wave):	4N35	2.5 kV
	4N36	1.75 kV
	4N37	1.05 kV
Collector-base voltage		70 V
Collector-emitter voltage (see Note 1)		30 V
Emitter-base voltage		7 V
Input-diode reverse voltage		6 V
Input-diode forward current:		
Continuous		60 mA
Peak (1 μs , 300 pps)		3 A
Phototransistor continuous collector current		100 mA
Continuous total power dissipation at (or below) 25°C free-air temperature:		
Infrared-emitting diode (see Note 2)		100 mW
Phototransistor (see Note 3)		300 mW
Continuous power dissipation at (or below) 25°C lead temperature:		
Infrared-emitting diode (see Note 4)		100 mW
Phototransistor (see Note 5)		500 mW
Operating temperature range, T_A		-55°C to 100°C
Storage temperature range, T_{stg}		-55°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds		260°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these conditions is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

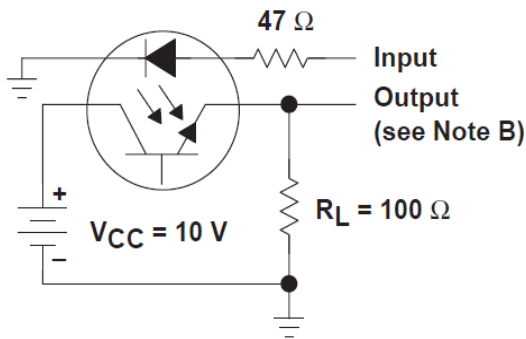
4N35, 4N36, 4N37 OPTOCOUPLERS

SOES021C – NOVEMBER 1981 – REVISED APRIL 1998

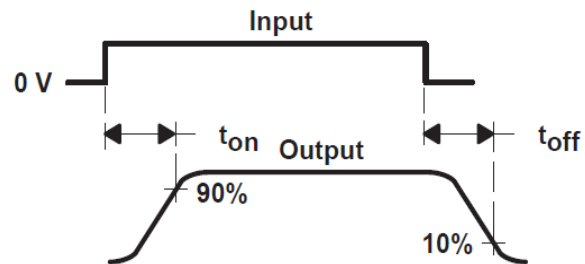
electrical characteristics at 25°C free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
$V_{(BR)CBO}$	Collector-base breakdown voltage	$I_C = 100 \mu A, I_E = 0, I_F = 0$	70†			V
$V_{(BR)CEO}$	Collector-emitter breakdown voltage	$I_C = 10 mA, I_B = 0, I_F = 0$	30†			V
$V_{(BR)EBO}$	Emitter-base breakdown voltage	$I_E = 100 \mu A, I_C = 0, I_F = 0$	7†			V
I_R	Input diode static reverse current	$V_R = 6 V$			10†	μA
I_{IO}	Input-to-output current	$V_{IO} = \text{rated peak value}, t = 8 ms$			100	mA
$I_{C(on)}$	On-state collector current	$V_{CE} = 10 V, I_F = 10 mA, I_B = 0$	10†			mA
		$V_{CE} = 10 V, I_F = 10 mA, I_B = 0, T_A = -55^\circ C$	4†			
		$V_{CE} = 10 V, I_F = 10 mA, I_B = 0, T_A = 100^\circ C$	4†			
$I_{C(off)}$	Off-state collector current	$V_{CE} = 10 V, I_F = 0, I_B = 0$		1	50	nA
		$V_{CE} = 30 V, I_F = 0, I_B = 0, T_A = 100^\circ C$			500†	μA
h_{FE}	Transistor static forward current transfer ratio	$V_{CE} = 5 V, I_C = 10 mA, I_F = 0$		500		
V_F	Input diode static forward voltage	$I_F = 10 mA$	0.8†		1.5†	V
		$I_F = 10 mA, T_A = -55^\circ C$	0.9†		1.7†	
		$I_F = 10 mA, T_A = 100^\circ C$	0.7†		1.4†	
$V_{CE(sat)}$	Collector-emitter saturation voltage	$I_C = 0.5 mA, I_F = 10 mA, I_B = 0 mA$			0.3†	V
r_{IO}	Input-to-output internal resistance	$V_{IO} = 500 V, \text{ See Note 6}$	10 ¹¹ †			Ω
C_{io}	Input-to-output capacitance	$V_{IO} = 0, f = 1 MHz, \text{ See Note 6}$		1	2.5†	pF

PARAMETER MEASUREMENT INFORMATION



TEST CIRCUIT



VOLTAGE WAVEFORMS

- A. The input waveform is supplied by a generator with the following characteristics: $Z_O = 50 \Omega, t_r \leq 15 ns, \text{ duty cycle } \approx 1\%, t_w = 100 \mu s.$
- B. The output waveform is monitored on an oscilloscope with the following characteristics: $t_f \leq 12 ns, R_{in} \geq 1 M\Omega, C_{in} \leq 20 pF.$

ANEXO IX. HOJAS DE ESPECIFICACIÓN DEL IRF630A

Advanced Power MOSFET

IRF630A

FEATURES

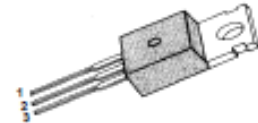
- Avalanche Rugged Technology
- Rugged Gate Oxide Technology
- Lower Input Capacitance
- Improved Gate Charge
- Extended Safe Operating Area
- Lower Leakage Current : 10 μ A (Max.) @ $V_{DS} = 200V$
- Low $R_{DS(ON)}$: 0.333 Ω (Typ.)

$$BV_{DSS} = 200 V$$

$$R_{DS(on)} = 0.4 \Omega$$

$$I_D = 9 A$$

TO-220



1.Gate 2. Drain 3. Source

Absolute Maximum Ratings

Symbol	Characteristic	Value	Units
V_{DSS}	Drain-to-Source Voltage	200	V
I_D	Continuous Drain Current ($T_c=25^\circ C$)	9	A
	Continuous Drain Current ($T_c=100^\circ C$)	5.7	
I_{DM}	Drain Current-Pulsed ①	36	A
V_{GS}	Gate-to-Source Voltage	± 30	V
E_{AS}	Single Pulsed Avalanche Energy ②	162	mJ
I_{AR}	Avalanche Current ①	9	A
E_{AR}	Repetitive Avalanche Energy ①	7.2	mJ
dv/dt	Peak Diode Recovery dv/dt ③	5.0	V/ns
P_D	Total Power Dissipation ($T_c=25^\circ C$)	72	W
	Linear Derating Factor	0.57	
T_J, T_{STG}	Operating Junction and Storage Temperature Range	- 55 to +150	$^\circ C$
T_L	Maximum Lead Temp. for Soldering Purposes, 1/8" from case for 5-seconds	300	

Thermal Resistance

Symbol	Characteristic	Typ.	Max.	Units
$R_{\theta JC}$	Junction-to-Case	--	1.74	$^\circ C/W$
$R_{\theta CS}$	Case-to-Sink	0.5	--	
$R_{\theta JA}$	Junction-to-Ambient	--	62.5	

IRF630A

N-CHANNEL POWER MOSFET

Electrical Characteristics ($T_c=25^\circ\text{C}$ unless otherwise specified)

Symbol	Characteristic	Min.	Typ.	Max.	Units	Test Condition
BV_{DSS}	Drain-Source Breakdown Voltage	200	--	--	V	$V_{GS}=0V, I_D=250\mu A$
$\Delta BV/\Delta T_J$	Breakdown Voltage Temp. Coeff.	--	0.21	--	V/ $^\circ\text{C}$	$I_D=250\mu A$ See Fig 7
$V_{GS(th)}$	Gate Threshold Voltage	2.0	--	4.0	V	$V_{DS}=5V, I_D=250\mu A$
I_{GSS}	Gate-Source Leakage, Forward	--	--	100	nA	$V_{GS}=30V$
	Gate-Source Leakage, Reverse	--	--	-100		$V_{GS}=-30V$
I_{DSS}	Drain-to-Source Leakage Current	--	--	10	μA	$V_{DS}=200V$
		--	--	100		$V_{DS}=160V, T_c=125^\circ\text{C}$
$R_{DS(on)}$	Static Drain-Source On-State Resistance	--	--	0.4	Ω	$V_{GS}=10V, I_D=4.5A$ ④
g_{fs}	Forward Transconductance	--	3.87	--	Ω	$V_{DS}=40V, I_D=4.5A$ ④
C_{iss}	Input Capacitance	--	500	650	pF	$V_{GS}=0V, V_{DS}=25V, f=1\text{MHz}$ See Fig 5
C_{oss}	Output Capacitance	--	95	110		
C_{rss}	Reverse Transfer Capacitance	--	45	55		
$t_{d(on)}$	Turn-On Delay Time	--	13	40	ns	$V_{DD}=100V, I_D=9A,$ $R_S=12\Omega$ See Fig 13 ④ ⑤
t_r	Rise Time	--	13	40		
$t_{d(off)}$	Turn-Off Delay Time	--	30	70		
t_f	Fall Time	--	18	50		
Q_g	Total Gate Charge	--	22	29	nC	$V_{DS}=160V, V_{GS}=10V,$ $I_D=9A$ See Fig 6 & Fig 12 ④ ⑤
Q_{gs}	Gate-Source Charge	--	4.3	--		
Q_{gd}	Gate-Drain("Miller") Charge	--	10.9	--		

Source-Drain Diode Ratings and Characteristics

Symbol	Characteristic	Min.	Typ.	Max.	Units	Test Condition
I_S	Continuous Source Current	--	--	9	A	Integral reverse pn-diode in the MOSFET
I_{SM}	Pulsed-Source Current ①	--	--	36		
V_{SD}	Diode Forward Voltage ④	--	--	1.5	V	$T_J=25^\circ\text{C}, I_S=9A, V_{GS}=0V$
t_{rr}	Reverse Recovery Time	--	137	--	ns	$T_J=25^\circ\text{C}, I_F=9A$
Q_{rr}	Reverse Recovery Charge	--	0.68	--	μC	$di_F/dt=100A/\mu\text{s}$ ④