



INSTITUTO TECNOLÓGICO DE CD. GUZMÁN

TESIS

Medición por elementos finitos vía wifi de la humedad relativa en un
Invernadero a través de un dispositivo móvil.

QUE PARA OBTENER EL GRADO DE:
MAESTRO EN INGENIERÍA ELECTRÓNICA

PRESENTA:
ING. JOSÉ ANTONIO CEJA GONZÁLEZ

DIRECTOR DE TESIS:
M.I.E. RAÚL RENTERÍA TORIZ

Cd. Guzmán Jalisco, México Agosto del 2018

AGRADECIMIENTOS

- ❖ Quiero agradecer a Dios que me permitió concluir con mis estudios de maestría y que me dio la fuerza necesaria para afrontar los problemas y dificultades que se me presentaron a lo largo de mis estudios.

- ❖ A mi familia que siempre me ha apoyado de manera incondicional durante todas las etapas de mi vida y de mi carrera profesional y en especial a mi Madre que ha hecho si sacrificado mucho por mí.

- ❖ A mi esposa que siempre está a mi lado en los momentos buenos y sobre todo en los momentos de adversidad y que me a apoyado para salir adelante en mi vida personal y profesional.

- ❖ A mis maestros de la maestría que han compartido sus conocimientos de manera profesional y me han ayudado a llevar este trabajo de la mejor manera posible.

- ❖ A mi Asesor y Director de Tesis que siempre ha estado apoyándome y guiándome para realizar un trabajo profesional durante este tiempo que desarrollamos este proyecto.

- ❖ A mis compañeros y amigos que siempre nos apoyamos y nos ayudamos mutuamente para llevar acabo las actividades de la maestría y que gracias su ayuda y consejos logramos terminar con éxito este paso importante de la carrera profesional.

Medición por elementos finitos vía wifi de la humedad relativa en un invernadero a través de un dispositivo móvil.

RESUMEN

Debido a la necesidad que se tiene de monitorear variables climatológicas en cultivos, donde es de suma importancia contar con sistemas de medición que permitan optimizar los procesos y la calidad de la producción, así como reaccionar de manera oportuna ante los riesgos que puede involucrar el impredecible comportamiento del clima, se diseñara un sistema automático de medición, registro y almacenamiento por elementos finitos en tiempo real vía wifi, para monitorear a través de un dispositivo móvil el comportamiento de la humedad relativa dentro de un invernadero.

Esta investigación desarrollara un sistema basado en tecnología inalámbrica wifi, para la medición, el registro y la transmisión de elemento finitos en tiempo real, sobre el comportamiento de la humedad relativa en un invernadero, para ser analizados en grados-día y así tener un manejo adecuado de la variable en los cultivos del invernadero. El proyecto consta en una red de sensores para la medición de la humedad y por medio de microcontroladores, localizados en cada área de cultivo dentro de un invernadero que entregaran los elementos finitos vía wifi a una computadora central de administración, utilizando las ventajas que ofrece la transmisión inalámbrica, en cuanto al bajo consumo de energía, así como la facilidad de implementación y el gran número de nodos que se pueden conectar, la cual hará la función de servidor que almacenara y controlara esta variable.

También constará de una aplicación móvil que podrá ejecutarse en las plataformas Android y IOS la cual tendrá el fin de visualizar de estas mediciones haciendo la función de extensión de monitor.

Un sistema de medición y monitoreo ágil, flexible y eficiente que se vuelva determinante para el manejo óptimo de la humedad relativa en el cultivo en un invernadero, y que contribuirá notablemente al mejoramiento de la calidad y en la optimización del proceso.

Palabras Clave: Internet de las cosas, ESP8266, Wifi,

Abstract

Due to the need to monitor climatic variables in crops, where it is of the utmost importance to have measurement systems that optimize the processes and the quality of production, as well as react in a timely manner to the risks that may involve the unpredictable climate behavior, an automatic system of measurement, recording and storage by finite elements will be designed in real time via wifi, to monitor, through a mobile device, the relative humidity behavior inside a greenhouse.

This research will develop a system based on Wi-Fi wireless technology, for the measurement, registration and transmission of finite elements in real time, on the behavior of relative humidity in a greenhouse, to be analyzed in degrees-day and thus have a management adequate of the variable in greenhouse crops. The project consists of a network of sensors for the measurement of humidity and by means of microcontrollers, located in each growing area in a greenhouse that delivers the finite elements via wifi to a central administration computer, using the advantages offered by the wireless transmission, in terms of low power consumption, as well as the ease of implementation and the large number of nodes that can be connected, which will serve as a server that will store and control this variable.

It will also consist of a mobile application that will be able to run on the Android and IOS platforms, which will have the purpose of displaying these measurements by performing the monitor extension function.

An agile, flexible and efficient measurement and monitoring system that becomes determinant for the optimal management of the relative humidity in the crop in a greenhouse, and that will contribute significantly to the improvement of the quality and in the optimization of the process.

Keywords: Internet of Things, ESP8266, Wifi.

Índice General

Índice de figuras	VI
Índice de tablas	VII
Capítulo I INTRODUCCION	1
1.1 Introducción	3
1.2 Objetivo General	4
1.3 Objetivos Específicos	4
1.4 Justificación	4
1.5 Hipótesis	5
1.6 Metas	5
1.7 Contribución del trabajo a la solución del problema	5
1.8 Principales Resultados	5
Capítulo II ANTECEDENTES	6
2.1 Estado del Arte	8
2.1.1 Sistema de monitoreo y control de invernaderos a través de una red inalámbrica mediante un servidor Web embebido en microcontroladores de alto rendimiento	8
2.1.2 Diseño de un sistema inalámbrico para el monitorio en tiempo real de temperatura y humedad relativa bajo invernadero	8
2.1.3 Sistema de Control Inalámbrico: Diseño, Construcción y Aplicación en Caldera Industrial	9
2.1.4 Diseño e implementación de un prototipo para monitoreo remoto con un robot móvil, a través de una cámara controlada vía internet por Wifi	9
2.1.5 Sistema de control de Humedad y Temperatura para invernaderos	10
Capítulo III MARCO TEÓRICO	11
3.1 Método de los Elementos Finitos	13
3.2 Internet de las cosas	13
3.3 Wifi	14
3.4 ESP8266	15
3.4.1 Placa Wemos basada en ESP8266	16
3.5 Plataforma de desarrollo Arduino	17
3.5.1 Arduino Mega 2560	18
3.5.2 Arduino Nano	19
3.6 Sensor DHT11	19
3.7 Protocolo de Comunicación I2C	20
3.8 Protocolo de comunicación 1-Wire	21
3.9 NRF24L01	22
3.10 Plataforma de Desarrollo Blynk	23
Capítulo IV METODOLOGÍA	25
4.1 Descripción general del Sistema	27
4.2 Descripción por etapas del sistema	28
4.2.1 Etapa de sensado de la humedad Relativa y temperatura	28
4.2.2 Etapa de Modulo Principal	37

Capítulo V RESULTADOS Y PANTALLAS DESARROLLADAS	59
5.1 Principales resultados alcanzados en el proyecto	61
5.1.1 Descripción de la aplicación móvil desarrollada en la plataforma de Blynk	61
5.1.2 Descripción de la interface de usuario en LabVIEW	62
5.1.3 Descripción del Módulo principal	63
Capítulo VI CONCLUSIONES Y APORTACIONES	64
6.1 Principales Conclusiones el proyecto	66
6.1.1 Costo beneficio del proyecto	66
6.1.2 Versatilidad del proyecto	66
6.2 Aportaciones del proyecto a la solución del problema	66
Capítulo VII ANEXOS	68

Índice de figuras

Figura 3.1 Procesador ESP8266EX	15
Figura 3.2 Interior de un ESP8266 con periféricos anotados	16
Figura 3.3 Arquitectura de Harvard	16
Figura 3.4 Wemos D1 Mini-Pro	17
Figura 3.5 WeMos D1 Mini	17
Figura 3.6 Arduino Mega 2560 R3	18
Figura 3.7 Arduino Nano	19
Figura 3.8 Sensor DHT 11	19
Figura 3.9 Pines de conexión I2C en Arduino UNO y Mega	20
Figura 3.10 NRF24L01 en sus dos versiones con Antena Impresa y Externa	23
Figura 3.11 Plataforma de desarrollo Blynk	23
Figura 3.12 Iconos de programación gráfica	24
Figura 3.13 Sistemas operativos compatibles con Blynk	24
Figura 4.1 Sistema de monitoreo implementado	27
Figura 4.2 Sensor Digital DHT11 y DHT22	28
Figura 4.3 Arduino Nano	28
Figura 4.4 Gestionar librería en el IDE de Arduino para el sensor DHT	29
Figura 4.5 Librería DHT instalada	29
Figura 4.6 Diagrama a bloques del sensor DHT11	30
Figura 4.7 Diagrama de Conexiones en fritzing	31
Figura 4.8 Gestionar librería en el IDE de Arduino para el Módulo NRF2401	33
Figura 4.9 Librería RF24-master instalada	33
Figura 4.10 Diagrama a bloques del circuito de sensado	36
Figura 4.11 Diagrama de Conexiones en fritzing	36
Figura 4.12 Diagrama a bloques del módulo principal	37
Figura 4.13 Diagrama a bloques del circuito de recepción de datos	40
Figura 4.14 Diagrama de conexiones del circuito de recepción de datos en fritzing	40
Figura 4.15 Diagrama a bloques de la etapa de conexión wifi del módulo principal	41
Figura 4.16 Diagrama de conexiones en fritzing	41
Figura 4.17 Ventana en el IDE de Arduino para descargar la tarjeta ESP8266	42
Figura 4.18 Diagrama de conexiones del Arduino y el ESP8266 en fritzing	44
Figura 4.19 Imagen del montaje del Arduino Mega y el Ethernet Shield	45
Figura 4.20 Conversor Paralelo a I2C para display LCD de 2 X 16	49
Figura 4.21 Búsqueda y descarga de la aplicación en PlayStore de Android	49
Figura 4.22 Pantalla de inicio al iniciar sesión en la aplicación de Blynk	50
Figura 4.23 Pantalla de Nuevo proyecto	51
Figura 4.24 Se asigna el nombre al proyecto y se selecciona la tarjeta Arduino para utilizar en el proyecto	51
Figura 4.25 Selección del Modo de conexión	52
Figura 4.26 Confirmación del envío del Auth Token	52
Figura 4.27 Correo recibido por parte de Blynk con el Auth Token	53
Figura 4.28 Ventana de comandos para crear aplicación	54
Figura 4.29 Selección del comando SuperChart	54
Figura 4.30 Cambio y edición del título del comando	55

Figura 4.31 Cambio y edición del título de la variable	-----	55
Figura 4.32 Edición del Grafico en la aplicación	-----	56
Figura 4.33 Asignación de Pin Virtual	-----	57
Figura 4.34 Agregar más variables a graficar	-----	57
Figura 4.35 Configuración de la resolución de los gráficos	-----	58
Figura 5.1 Variables Virtuales en la aplicación móvil y programa de Arduino	-----	61
Figura 5.2 Grafica de variables en la aplicación móvil	-----	61
Figura 5.3 Pantalla principal de la interfase de usuario en LabVIEW	-----	62
Figura 5.4 Diagrama a Bloques de la interfase de usuario en LabVIEW	-----	62
Figura 5.5 Fotografía del módulo principal y sus 3 tipos de visualización	-----	63

Índice de Tablas

Tabla I Especificaciones del microcontrolador ESP8266	-----	15
Tabla II Sensor DTH11	-----	20

GLOSARIO

IoT: Internet of things; Internet de las cosas es un concepto que se refiere a la interconexión digital de objetos cotidianos con internet.

Wi-Fi: (o Wi-fi, WiFi, Wifi, wifi) es un conjunto de estándares para redes inalámbricas basado en las especificaciones del estándar IEEE 802.11. Wi-Fi se creó para ser utilizada en redes locales inalámbricas.

BYTE: conjunto de ocho bits.

BIT: dígito binario. Unidad fundamental de la información.

BLUETOOTH: es la norma que define un estándar global de comunicación inalámbrica que posibilita la transmisión de voz y datos entre diferentes equipos mediante un enlace por radiofrecuencia. Diseñado para aplicaciones de corta distancia (<10m).

Capítulo I

Introducción

1.1 Introducción.

En el estudio de la interacción entre las condiciones climáticas, la agricultura, los recursos económicos, la calidad de los productos, la tecnología productiva y la eficiencia en los procesos, se generan indicadores que nos permiten generar respuestas a problemáticas en los procesos productivos para la eficiencia de los campos de cultivo controlando sus condiciones climáticas de una manera más fácil y eficiente.

Como consecuencia de ello, el registro y análisis del comportamiento climatológico dentro de un ambiente artificial es fundamental para la eficiencia y la mejora de calidad en los productos, por lo que herramientas de monitoreo y registro de variables se vuelven sumamente importantes a la hora de optimizar los procesos, lo que contribuye en la reducción de daños fisiológicos en los cultivos, lo que se traduce en pérdidas de producción, por consecuencia reducción de utilidades.

En esta investigación se presentará el diseño de un sistema de monitoreo en tiempo real por elementos finitos de la variable humedad relativa para ser aplicado en el análisis en grados-día, y serán transmitidos vía wifi a una computadora central de administración. El concepto de grados-día, es de gran utilidad en la agricultura permitiendo, entre otras cosas programar fechas de siembra y ciclos de cultivo, pronosticar fechas de cosecha, plagas y enfermedades, así como identificar el grado de desarrollo en sus diferentes etapas.

Las soluciones eficientes de comunicación inalámbricas para el registro y monitoreo de variables climatológicas han producido gran impacto en el sector agrícola, ya que representan en términos de reducción de costos en instalación y mantenimiento, sobre los sistemas tradicionales de cableado. Esto ha abierto la puerta para el desarrollo de nuevas tecnologías enfocadas a sistemas inalámbricos de control para diferentes usos, como lo son: Sistemas de control de riego, automatización y mejoramiento de procesos de empaque entre otros.

La versatilidad en el monitoreo en tiempo real a distancia nos permite tener un mejor control de las variables lo que es muy importante en ambientes artificiales controlados como son los invernaderos.

Por ello se ha decidido implementar un sistema que nos permita mediante una aplicación móvil, tener acceso a esta información en todo momento desde nuestra terminal celular y así contar con la capacidad de respuesta más rápida en el control climático, lo que será reflejado en la calidad de los cultivos, la eficiencia de los procesos y la rentabilidad en la producción costo-utilidad.

1.2 Objetivo General.

Diseñar un sistema de monitoreo y registro de la Medición por elementos finitos de la humedad relativa y transmitidos vía wifi a una computadora central bajo invernadero y que puedan ser visualizados a través de una aplicación dispositivo móvil (Terminal Celular o Tablet).

1.3 Objetivos Específicos.

- a) Identificar la viabilidad técnica de incorporar tecnología inalámbrica, para efectuar la transmisión de datos del invernadero, frente a soluciones tradicionales de transmisión de datos por cable.
- b) Definir la tecnología de transmisión de los datos, con base en las características y disposición geográfica de las áreas productivas (bloques en los que está segmentado el cultivo) donde se efectuarán las mediciones de la variable climatológica Humedad Relativa en cuestión, teniendo en cuenta factores tales como, las distancias al lugar de recepción de los datos, los obstáculos físicos, las instalaciones eléctricas disponibles y el uso que se le dará a los datos producto de las mediciones.
- c) Elegir los sensores adecuados para la medición de humedad relativa, teniendo en cuenta el uso que se le dará a las mediciones, las condiciones climatológicas de los lugares donde se implementará, y la aplicabilidad futura de la solución.
- d) Rediseñar el hardware necesario para efectuar la medición de la variable climatológica, incorporando el acondicionamiento de las señales obtenidas, el almacenamiento y la transmisión de los datos, basándose en módulos de medición comandados por microprocesadores.
- e) Rediseñar el software adecuado que permita: configurar los parámetros de medición del sistema (tasas de muestreo, fecha y hora de inicio de medición, umbrales para generación automática de alarmas); diagnosticar el estado técnico de funcionamiento (módulos activos o inactivos, estado de baterías o suministro eléctrico); recibir y administrar los datos obtenidos de las mediciones permitiendo el almacenamiento de estos en una base de datos o en forma de archivo plano o similar.
- f) Diseñar la aplicación para dispositivos móviles en las plataformas más usuales; Android y IOS que ofrecerá la información de muestreo en forma gráfica en todo momento y en todo lugar cuando se esté conectado a una red de datos móviles o wifi.

1.4 Justificación.

Debido a la necesidad que se tiene de monitorear variables climatológicas en cultivos bajo invernadero, donde es de suma importancia contar con sistemas de medición que permitan optimizar los procesos y la calidad de la producción, así como el ahorro de los elevados costos de instalación de redes alámbricas, se diseñara un sistema automático de medición, registro y almacenamiento por elementos finitos en tiempo real vía wifi, para monitorear a través de un dispositivo móvil y/o una computadora, el comportamiento de la humedad relativa dentro de un invernadero.

1.5 Hipótesis.

Los cultivos bajo invernadero aumentan su productividad si se monitorea la humedad relativa y controla adecuadamente, pero los elevados costos de instalación alámbrica no son viable para algunos productores pequeños.

1.6 Metas.

- 1.- Se diseñó un sistema hardware basado en una plataforma de desarrollo libre de licencias Arduino, la cual es de fácil acceso a información y compatible con diferentes tipos de hardware disponible en el mercado.
- 2.- Se realizó una aplicación móvil basada en una plataforma de desarrollo de aplicaciones libre Blink y que además nos proporciona espacio de almacenamiento en su servidor de manera gratuita.
- 3.- El sistema completo se desarrolló utilizando componentes de bajo costo y de fácil acceso para cualquier usuario, con el fin de abaratar costos y hacer el sistema más versátil y adaptativo para las diferentes necesidades de cada productor en cultivos bajo invernadero.

1.7 Contribución del Trabajo a la Solución del Problema.

Los pequeños productores podrán tener acceso a un sistema de bajo costo que se puede adaptar a cualquier necesidad ya que es un sistema de bajo costo y mantenimiento además de ser fácil instalar y configurar. Gracias a este sistema cualquier productor ya sea grande o pequeño podrá realizar una medición de variables en Humedad y temperatura de manera más eficiente lo que se traducirá en mejores resultados en sus cosechas.

1.8 Principales Resultados.

- 1.- Se logró realizar un sistema capaz de conectarse a internet a través de conexión Wifi haciendo uso de un pequeño microcontrolador ESP8266 que puede funcionar como una simple antena o hasta como un microcontrolador principal en un sistema embebido simple.
- 2.- Se logró realizar pequeños prototipos de monitoreo de Humedad Relativa y Temperatura utilizando el ESP8266 como microcontrolador principal dentro de un sistema completo embebido utilizando la plataforma de desarrollo Blink para el monitoreo a través de la aplicación móvil.
- 3.- Como resultado del estudio de las diferentes plataformas de desarrollo de aplicaciones móviles que están disponibles de manera libre se logró identificar los pros y contras de cada una de ellas lo que nos ayuda a saber cuál es la más adecuada dependiendo de la aplicación del proyecto.
- 4.- Se logró diseñar un sistema de conexión RF entre sensores lo que reduce en gran medida el costoso proceso de instalación de cableado lo que reduce en gran medida el costo de nuestro sistema.

Capítulo II

Antecedentes

2.1 Estado del Arte

Este trabajo de investigación tomo como punto de partida algunos trabajos e investigaciones relacionadas con el monitoreo y control a distancia a través de conexiones RF y Wifi. Estos trabajos están documentados y a continuación se mencionan.

2.1.1 Sistema de monitoreo y control de invernaderos a través de una red inalámbrica mediante un servidor Web embebido en microcontroladores de alto rendimiento.

Tesis Ingeniería.

Lugar de Publicación: Centro Universitario de Querétaro

Fecha: Agosto del 2010

Resumen:

En este proyecto se realizó un estudio de los sistemas Web Embebidos con el objetivo de diseñar e implementar un prototipo que permitiera controlar y monitorear la temperatura y la humedad de un invernadero desde cualquier parte de una red LAN. El estudio permitió la selección del sistema mas adecuado para el proyecto. Como dispositivo de procesamiento se optó por la utilización de los kits de desarrollo de Ethernet, los cuales cumplían con todos los requerimientos del sistema. En el microcontrolador se programó el firmware del sistema el lenguaje C en el software PIC C Compiler. El código incluye la implementación de parte del stack TCP/IP. Se eligió utilizar TCP en lugar de UDP como protocolo de transporte porque TCP es confiable. La página Web diseñada se envía con las mediciones de temperatura, humedad y el estado de los dispositivos de control.

Para más información de este documento revisar **Anexo 1**.

2.1.2 Diseño de un sistema inalámbrico para el monitorio en tiempo real de temperatura y humedad relativa bajo invernadero.

Tesis Ingeniería

Lugar de Publicación: Universidad de la Salle, Facultad de Ingeniería de Diseño y Automatización Electrónica

Año: 2006

Resumen:

El presente trabajo presenta el diseño de un sistema basado en tecnología inalámbrica, para efectuar la medición, registro, y transmisión de datos en tiempo real, sobre el comportamiento de la temperatura y humedad relativa en un invernadero, para ser aplicado en análisis de grados-día y el monitoreo para el manejo climatológico óptimo en un cultivo de flores. El diseño consiste en una red de dispositivos de medición, operados con baterías y por medio de microcontroladores, localizados en cada área productiva del cultivo, que entregan los datos de las mediciones vía radiofrecuencia a una computadora central de administración por software, aprovechando las ventajas que ofrece el nuevo estándar de transmisión inalámbrica IEEE 802.15.4, en cuanto a bajo costo y consumo de energía, así como facilidad de implementación y gran número de nodos.

Para más información de este documento revisar **Anexo 2**.

2.1.3 Sistema de Control Inalámbrico: Diseño, Construcción y Aplicación en Caldera Industrial.

Artículo Científico

Lugar de Publicación: Ximhai 10(3): 39-53.

Fecha: 22 de noviembre de 2013

Resumen:

Este trabajo de investigación y desarrollo tecnológico tiene como finalidad el diseño y construcción de una unidad de control con capacidad de comunicación en red inalámbrica y cableada, con la finalidad de desarrollar una unidad de control de procesos poderosa, fácil de utilizar, flexible y bajo costo, a través de la cual, sea posible el implementar sistemas de control distribuidos inalámbricos y cableados basados en topologías de red avanzadas como estrella, árbol y malla, dado las diferentes tecnologías de comunicación que hoy en día se manejan en el campo de la automatización de procesos industriales y tomando referencia de las más novedosas y prometedoras ventajas de las tecnologías de avanzada disponibles en el mercado internacional.

El sistema incluye el desarrollo de un driver de comunicación en dos capas, el cual fue desarrollado bajo la plataforma del microcontrolador ATmega 328 con bootloader Arduino y la capa superior en ambiente de desarrollo LabVIEW, con la finalidad de facilitar al usuario el desarrollo de la aplicación de control y lograr que las tareas de comunicación entre las unidades de control y cuarto de supervisión sean completamente transparentes para el programa del usuario.

Para más información de este documento revisar **Anexo 3**.

2.1.4 Diseño e implementación de un prototipo para monitoreo remoto con un robot móvil, a través de una cámara controlada vía internet por Wifi.

Tesis Ingeniería

Lugar de Publicación: Instituto Politécnico Nacional

Año: 2012

Resumen:

Este proyecto consiste en implementar un servicio de monitoreo. El cual proporciona la facultad de desplazar un robot móvil a través del ambiente o evento que se esté monitoreando. En este caso, con el término monitoreo, implica grabar o tomar video y cuando se menciona internet, quiere decir que este vídeo tomado, será enviado o transmitido por este medio de comunicación: ya sea a uno o varios usuarios. Por otra parte, se transmitirá video por internet a un usuario determinado y también este usuario será capaz de controlar la posición del carro quien es el portador de la cámara. Todo esto mediante una página Web.

Para más información de este documento revisar **Anexo 4**.

2.1.5 SISTEMA DE CONTROL DE HUMEDAD Y TEMPERATURA PARA INVERNADEROS

Artículo Científico:

Lugar de Publicación: Universidad Pedagógica y Tecnológica de Colombia

Resumen:

Este desarrollo se origina en la Universidad Pedagógica Y Tecnológica de Colombia (UPTC) Duitama-Colombia por iniciativa de los docentes de la asignatura Microcontroladores de las Escuelas de Licenciatura en Educación Industrial e Ingeniería electromecánica, el objetivo es controlar de forma precisa la humedad, la temperatura, la iluminación, la ventilación y demás variables relevantes para los invernaderos, con la intención de lograr habientes artificiales e ideales para el buen desarrollo de algunos cultivos. Específicamente se pensó en el cultivo de Shiitake seta muy exigente en cuanto a condiciones ambientales de cultivo. Se ha desarrollado inicialmente un sistema de control que consiste de módulos de sensores en base al sensor SHT11 para Temperatura, humedad relativa y punto de rocío, módulos actuadores en base a reles de estado sólido y un módulo de control con los algoritmos de control, fusiones gráficas de agradable y útil interfaz a usuario mediante pantalla LCD gráfica y funciones de datalogger para posterior descarga y análisis de datos en PC.

Todos los módulos se intercomunican de forma inalámbrica en la banda de 2.4Ghz mediante transceiver TRF-2.4GHZ, se eligió la comunicación inalámbrica porque disminuye en mucho los costos y el tiempo de instalación. En todos los módulos se eligió como centro del sistema μ controladores Microchip ya sea de la serie 16F8xx o 18F4xx. En la parte de simulación se eligió Proteus Isis por sus modelos de simulación VSM; el lenguaje de programación utilizado fue el C y como compilador el software PIC C COMPILER debido a que ofrece RTOS (Real Time Operating System).

Para más información de este documento revisar **Anexo 5**.

Capítulo III

Marco Teórico

Este proyecto inicialmente tiene en cuenta medir las variables humedad absoluta, humedad relativa y temperatura.

Se define HUMEDAD como la medida del contenido de agua en la atmósfera. La atmósfera contiene siempre algo de agua en forma de vapor. La cantidad máxima depende de la temperatura; crece al aumentar ésta: a 4,4 °C, 1.000 kg de aire húmedo contienen un máximo de 5 kg de vapor; a 37,8 °C 1.000 kg de aire contienen 18 kg de vapor.

El peso del vapor de agua contenido en un volumen de aire se conoce como HUMEDAD ABSOLUTA y se expresa en unidades de masa de agua por unidades de masa o de volumen de aire seco. Frecuentemente se utiliza la medida de gramos de vapor de agua por metro cúbico de aire.

La HUMEDAD RELATIVA, dada en los informes meteorológicos, es la razón entre el contenido efectivo de vapor en la atmósfera y la cantidad de vapor que saturaría el aire a la misma temperatura.

Si la temperatura atmosférica aumenta y no se producen cambios en el contenido de vapor, la humedad absoluta no varía mientras que la relativa disminuye. Una caída de la temperatura incrementa la humedad relativa produciendo ROCÍO por condensación del vapor de agua sobre las superficies sólidas.

La temperatura a la cual se empieza a formar el rocío en el aire que contiene una cantidad conocida de vapor de agua se llama PUNTO DE ROCÍO.

3.1 Método de los Elementos Finitos

El método de los elementos finitos es un método de aproximación de problemas continuos de tal forma que:

- El continuo se divide en un número finito de partes, “**elementos**”, cuyo comportamiento se especifica mediante un número finito de parámetros asociados a ciertos puntos característicos denominados “**nodos**”. Estos nodos son los puntos de unión de cada elemento con sus adyacentes.
- La solución del sistema completo sigue las reglas de los problemas discretos. El sistema completo se forma por ensamblaje de los elementos.
- Las incógnitas del problema dejan de ser funciones matemáticas y pasan a ser un valor de estas funciones en los nodos.
- El comportamiento en el interior de cada elemento queda definido a partir del comportamiento de los nodos mediante las adecuadas **funciones de interpolación o funciones de forma**.

3.2 Internet de las cosas.

El internet de las cosas (IOT) consiste en que diferentes cosas u objetos tengan la capacidad de conectarse a internet en cualquier momento y en cualquier lugar. En un sentido más técnico, consiste en la integración de sensores y dispositivos en objetos cotidianos que estén conectados a internet a través

de redes fijas e inalámbricas. De esta manera, cualquier objeto es susceptible de ser conectado y manifestarse en la red. Además, el IOT implica que todo objeto puede ser una fuente de información. [1].

El internet de las cosas está presente en nuestro día a día. En nuestro entorno laboral, en nuestra casa, en la escuela, en el supermercado, etc.

Un ejemplo de ello sería una máquina expendedora de refrescos que funciona con la publicación de un tweet, o con un hashtag en la página de la propia empresa o con un like en Facebook.

En definitiva, el objetivo ideal del IOT sería lograr que cualquier objeto tenga vida propia a través de internet y con ello una identidad. [1].

3.3 Wifi.

Wifi es una tecnología de comunicación inalámbrica que permite conectar a internet equipos electrónicos, como computadoras, tablet's, smartphones o celulares, etc., mediante el uso de radiofrecuencias o infrarrojos para la trasmisión de la información.

El funcionamiento aparente de una red Wi-Fi es muy similar al de una red Ethernet, solo que sin cables. No obstante, además de las direcciones IP y las direcciones MAC, en esta tecnología inalámbrica hay que tener en cuenta otros conceptos:

Estándar IEEE802.11: “Wi-Fi” se basa en este estándar, el cual es en realidad un conjunto de estándares. Dependiendo de la compatibilidad con uno o más de dichos estándares, encontraremos dispositivos que pueden formar parte de redes Wi-Fi 802.11b, 802.11g o 802.11n entre otros.

Punto de acceso (AP): un punto de acceso es un equipo de red inalámbrico (puede ser un computador con el software adecuado, o un dispositivo hardware específico) que se encarga de gestionar de forma centralizada las comunicaciones de todos los dispositivos que forman la red Wi-Fi. No solo se utiliza para controlar las comunicaciones internas de la red, sino que también hace de puente en las comunicaciones con las redes externas (redes Ethernet e Internet), a modo de “transformador de señal” entre redes inalámbricas y cableadas.

Modo: un dispositivo Wi-Fi puede tener un rol determinado dentro de la red, y esto se configura estableciendo su modo de funcionamiento. El modo Station o (“Managed”) es el modo en el que un dispositivo es un mero cliente que se conecta a un punto de acceso para tener conectividad. El modo AP o (“Master”) es el modo en el que un dispositivo puede trabajar él mismo como punto de acceso (si dispone del firmware adecuado).

SSID: es un dato emitido por el punto de acceso que identifica la red inalámbrica a la que pertenece. En otras palabras, es el “nombre de la red” que los terminales son capaces de ver para poderse conectar.

Canal: la banda de frecuencias electromagnéticas en la que trabaja una red Wi-fi (la banda de los 2,4 GHz, generalmente) se divide en varios canales. Concretamente, el estándar subdivide el rango de los 2,4 GHz en 14 canales separados entre sí por 5 MHz (aunque cada país aplica sus propias restricciones al número de canales disponibles. [2].

3.4 ESP8266

ESP8266 es el nombre de un microcontrolador diseñado por una compañía china llamada Espressif Systems en su sede en Shanghai. Pero su producción en masa inicio hasta principios del año 2014, donde se anunció que este chip sería una excelente solución automática de redes wifi que se ofrece como puente entre los microcontroladores que hasta ahora existen o que tiene la capacidad de ejecutar aplicaciones independientes. [3].

Un ESP8266 salido de fábrica no sería de mucha utilidad ya que su producción está basada en la compactación de un chip SMT (Tecnología de Montaje Superficial por sus siglas en inglés - Surface Mount Technology) el cual viene en un pequeño paquete de tan solo cinco milímetros cuadrados. La buena noticia es que gracias a que diversos fabricantes que construyen placas de circuito impreso prefabricadas adecuándolos y dejándolos listos para nuestro uso.

Esto nos permite trabajar con este dispositivo único acoplado a un microcontrolador, para desarrollar proyectos o como sistema autónomo para ciertas aplicaciones.

Características eléctricas de este microcontrolador son:

Tabla I: Especificaciones del Microcontrolador ESP8266

Voltaje	3.3 V
Consumo de corriente	10 μ A – 170 mA
Memoria Flash	16 MB máx. (512 k normal)
Procesador	Tensilica L106 32 bit
Velocidad del procesador	80 – 160 MHz
GPIOs	17
Analógico a digital	1 entrada con 10 bit de resolución (1024 valores)
Soporte de 802.11	b/g/n/d/e/i/k/r
Máximas conexiones simultáneas	5

El system on a chip (SoC) ESP9266EX usa un microcontrolador Tensilica Xtensa L106, que es un procesador de 32 bit con instrucciones de 16 bit.



Figura 3.1: Procesador ESP8266EX

CAPITULO III: Marco Teórico

El SoC describe la tendencia cada vez más frecuente de usar tecnologías de fabricación que integran todos o gran parte de los módulos que componen un computador o cualquier otro sistema informático o electrónico en un único circuito integrado o chip. Como podemos ver a continuación:

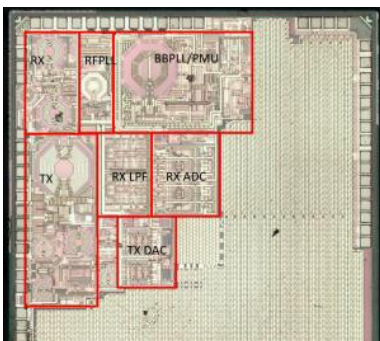


Figura 3.2: Interior de un ESP8266 con periféricos anotados.

El procesador funciona por defecto a 80 MHz, pero puede ir hasta 160 MHz, tiene ~ 80kB de DRAM (Data RAM), y ~ 35kB IRAM (Instruction RAM). La IRAM se carga en el arranque con lo que el usuario quiere mantener en el procesador, aunque el procesador puede ejecutar el código directamente fuera del flash externo a una velocidad más baja.

b) Arquitectura

Tiene una arquitectura de Harvard, con lo cual la CPU puede tanto leer una instrucción como realizar un acceso a la memoria de datos al mismo tiempo, incluso sin una memoria caché. En consecuencia, una arquitectura de computadores Harvard puede ser más rápida para un circuito complejo, debido a que la instrucción obtiene acceso a datos y no compete por una única vía de memoria. [4].

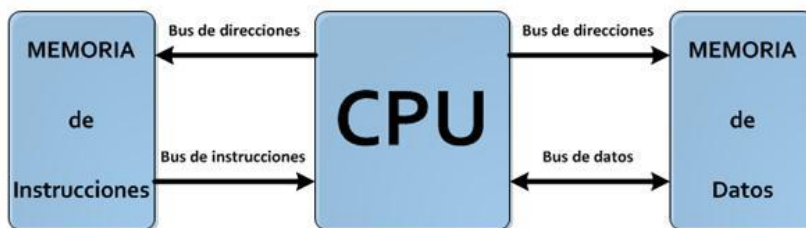


Figura 3.3: Arquitectura de Harvard

3.4.1 Placa Wemos basada en ESP8266

La D1 mini pro de WeMos es una placa de desarrollo equiparable a la NodeMCU, ya que dispone de un conector USB, puede usarse en una placa de pruebas, puede alimentarse a 5 V, tiene un botón de reset, 4 MB de memoria flash... aunque en un formato más reducido ya que tiene 16 patillas a una décima de pulgada.

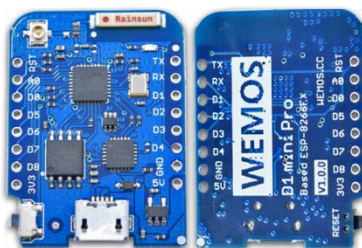


Figura 3.4: Wemos D1 Mini-Pro

Una Variante de este modelo es el D1 Mini que a diferencia del Pro este está basado en el ESP-14.

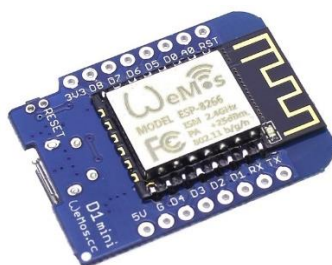


Figura 3.5: WeMos D1 Mini

3.5 Plataforma de desarrollo Arduino.

Arduino es una plataforma de desarrollo electrónica abierta (Open-Source) para la creación de prototipos basada en software y hardware libre.

Arduino puede tomar información y datos del entorno a través de sus pines de entrada/salida tanto analógicos como digitales haciendo uso de diferentes sensores que existen en el mercado. Gracias a ello también es posible realizar control y actuar sobre todo aquello que le rodea haciendo uso de Motores, luces y diferentes tipos de actuadores.

El micro controlador de la placa Arduino se programa mediante un sencillo código de programación de muy alto nivel basado en el lenguaje C++ y un entorno (IDE) que responde a las especificaciones de Open Software.

Los proyectos desarrollados en Arduino pueden ejecutarse sin necesidad de conectarlo a un ordenador, aunque si bien tiene la posibilidad de comunicarse con diferentes tipos de software, por ejemplo: LabVIEW, Proteus, Flash, Processing, etc.

En definitiva, las posibilidades de Arduino son inmensas, además todos los días aparecen en la red proyectos de lo más curioso y novedosos, aplicaciones y hardware compatible para desarrollar todo tipo de proyectos donde la imaginación es el límite.

3.5.1 Arduino Mega 2560.

Arduino Mega es una tarjeta de desarrollo open-source construida con un microcontrolador modelo Atmega2560 que posee pines de entradas y salidas (E/S), analógicas y digitales. Esta tarjeta es programada en un entorno de desarrollo que implementa el lenguaje Processing/Wiring. Arduino puede utilizarse en el desarrollo de objetos interactivos autónomos o puede comunicarse a un PC a través del puerto serial (conversión con USB) utilizando lenguajes como Flash, Processing, MaxMSP, etc. Las posibilidades de realizar desarrollos basados en Arduino tienen como límite la imaginación.

El Arduino Mega tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16 entradas análogas, 4 UARTs (puertos serial por hardware), cristal oscilador de 16MHz, conexión USB, jack de alimentación, conector ICSP y botón de reset. Arduino Mega incorpora todo lo necesario para que el microcontrolador trabaje; simplemente conéctalo a tu PC por medio de un cable

USB o con una fuente de alimentación externa (9 hasta 12VDC). El Arduino Mega es compatible con la mayoría de los shields diseñados para Arduino Duemilanove, diecimila o UNO.

Esta nueva versión de Arduino Mega 2560 adicionalmente a todas las características de su sucesor utiliza un microcontrolador ATmega8U2 en vez del circuito integrado FTDI. Esto permite mayores velocidades de transmisión por su puerto USB y no requiere drivers para Linux o MAC (archivo inf es necesario para Windows) además ahora cuenta con la capacidad de ser reconocido por el PC como un teclado, mouse, joystick, etc.

Para más información sobre este producto visita www.arduino.cl, allí encontraras tutoriales y ejemplos que te serán de utilidad si estás iniciándote con Arduino.[5].

Características:

Microcontrolador ATmega2560.

Voltaje de entrada de - 7-12V.

54 pines digitales de Entrada/Salida (14 de ellos son salidas PWM).

16 entradas análogas.

256k de memoria flash.

Velocidad del reloj de 16Mhz.

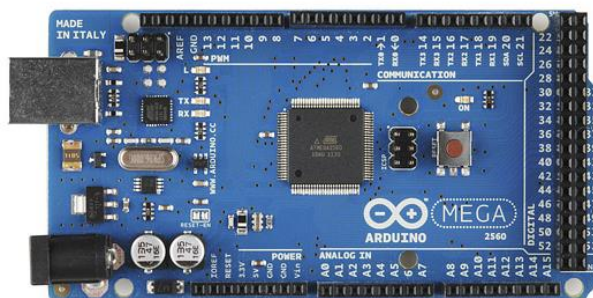


Figura 3.6: Arduino Mega 2560 R3

3.5.2 Arduino Nano.

El Arduino Nano es una pequeña, pero poderosa tarjeta basada en el ATmega328. Posee las mismas funcionalidades que un Arduino UNO, solo que en un tamaño reducido. Para programarla solo se necesita de un cable Mini USB.[5].

Características:

Microcontrolador: ATmega328

Voltaje de operación: 5V

Voltaje de alimentación (Recomendado): 7-12V

I/O Digitales: 14 (6 son PWM)

Memoria Flash: 32KB

EEPROM: 1KB

Frecuencia de trabajo: 16MHz

Dimensiones: 0.73" x 1.70"



Figura 3.7: Arduino Nano

3.6 Sensor DHT11.

DHT11 que nos va a servir para medir tanto la humedad relativa como la temperatura la resolución que presenta es de 1% para la humedad relativa y de 1°C para la temperatura.



Figura 3.8: Sensor DHT 11

CAPITULO III: Marco Teórico

Este sensor se caracteriza por tener la señal digital calibrada por lo que asegura una alta calidad y una fiabilidad a lo largo del tiempo, ya que contiene un microcontrolador de 8 bits integrado. Está constituido por dos sensores resistivos (NTC y humedad). Tiene una excelente calidad y una respuesta rápida en las medidas. Puede medir la humedad entre el rango 20% – aprox. 95% y la temperatura entre el rango 0°C – 50°C.

El protocolo de comunicación es a través de un único hilo (protocolo 1-wire), por lo tanto, hace que la integración de este sensor en nuestros proyectos sea rápida y sencilla. Además, presenta un tamaño reducido, un bajo consumo y la capacidad de transmitir la señal hasta 20 metros de distancia.

Características de dicho sensor:

Tabla II: Sensor DHT 11

Model	DHT11		
Power supply	3-5.5V DC		
Output signal	digital signal via single-bus		
Sensing element	Polymer resistor		
Measuring range	humidity 20-90%RH; temperature 0-50 Celsius		
Accuracy	humidity +-4%RH (Max +-5%RH); temperature +-2.0Celsius		
Resolution or sensitivity	humidity 1%RH;	temperature 0.1Celsius	
Repeatability	humidity +-1%RH;	temperature +-1Celsius	
Humidity hysteresis	+-1%RH		
Long-term Stability	+-0.5%RH/year		
Sensing period	Average: 2s		
Interchangeability	fully interchangeable		
Dimensions	size 12*15.5*5.5mm		

3.7 Protocolo de Comunicación I2C.

El bus I2C, también conocido como interfaz de dos hilos (Two Wire Interface) o abreviado TWI es una forma simple de comunicar dispositivos mediante dos cables.

El Arduino proporciona este tipo de comunicación a través de dos de sus patillas llamadas SDA y SCL.

Arduino Uno
Patillas A4 y A5



Arduino
Mega

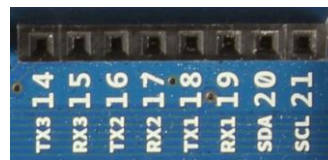


Figura 3.9: Pines de conexión I2C en Arduino UNO y Mega

I2C (Inter-Integrated Circuit, también conocido con el nombre de TWI –de “TWo-wire”, literalmente “dos cables” en inglés, es un sistema muy utilizado en la industria principalmente para comunicar circuitos integrados entre sí.

En el protocolo I2C del Arduino es considerado como dispositivo maestro y cada uno de los otros dispositivos externos I2C como esclavos.

Cada esclavo tiene su propia dirección expresada como un número hexadecimal que permite al Arduino direccionarlo para comunicarse con él. Cada dispositivo tiene un rango posible de direcciones dadas por su fabricante

Su principal característica es que utiliza dos líneas para transmitir la información: una (llamada línea “SDA”) sirve para transferir los datos (los 0s y los 1s) y otra (llamada línea “SCL”) sirve para enviar la señal de reloj.

En realidad, también se necesitarían dos líneas más: la de alimentación y la de tierra común, pero estas ya se presuponen existentes en el circuito.

La velocidad de transferencia de datos es de 100 Kbits por segundo en el modo estándar (aunque también se permiten velocidades de hasta 3,4 Mbit/s). No obstante, al haber una única línea de datos, la transmisión de información es “half duplex” (es decir, la comunicación solo se puede establecer en un sentido al mismo tiempo) por lo que en el momento que un dispositivo empiece a recibir un mensaje, tendrá que esperar a que el emisor deje de transmitir para poder responderle.

3.8 Protocolo de comunicación 1-Wire

El BUS 1-Wire, permite realizar una comunicación serial asincrónica entre un dispositivo maestro y uno o varios dispositivos esclavos, utilizando un único pin de E/S del microcontrolador. Podemos señalar algunas características de este bus:

- Utiliza niveles de alimentación CMOS/TTL con un rango de operación que abarca desde 2.8V hasta 6V.
- Tanto el maestro como los esclavos transmiten información de forma bidireccional, pero, sólo en una dirección a la vez, de ésta manera la comunicación es realizada en forma “half duplex”.
- Toda la información es leída o escrita comenzando por el bit menos significativo (LSB).
- No se requiere del uso de una señal de reloj, ya que, cada dispositivo 1-Wire posee un oscilador interno que se sincroniza con el del maestro cada vez que en la línea de datos aparezca un flanco de bajada.
- La alimentación de los esclavos se puede hacer utilizando el voltaje propio del BUS. Para ello, cada circuito esclavo posee un rectificador de media onda y un condensador, durante los períodos en los cuales no se efectúa ninguna comunicación, la línea de datos se encuentra en estado alto debido a la resistencia de “Pull Up”; en esa condición, el diodo entra en conducción y carga al condensador. Cuando el voltaje de la red cae por debajo de la tensión del condensador, el diodo se polariza en inverso evitando que el condensador se descargue. La carga almacenada en el condensador alimentará al circuito esclavo.
- Las redes de dispositivos 1-Wire pueden tener fácilmente una longitud desde 200m y contener unos 100 dispositivos.

- Todas las tensiones mayores que 2,2Voltios son consideradas un 1 lógico mientras que como un 0 lógico se interpreta cualquier voltaje menor o igual a 0,8 V.
- La transferencia de información es a 16.3Kbps en modo "Standard" y hasta a 142Kbps en modo "Over-Drive".[6].

3.9 NRF24L01

El NRF24L01 es un chip de comunicación inalámbrica fabricado por Nordic Semiconductor que podemos conectar a un procesador como Arduino

El NRF24L01 integra un transceptor RF (transmisor + receptor) a una frecuencia entre 2.4GHz a 2.5GHz, una banda libre para uso gratuito. La velocidad de transmisión es configurable entre 250 Kbps, 1Mbps, y 2 Mbps y permite la conexión simultánea con hasta 6 dispositivos.

El NRF24L01 también incorpora la lógica necesaria para que la comunicación sea robusta, como corrección de errores y reenvío de datos si es necesario, liberando de esta tarea al procesador. El control del módulo se realiza a través de bus SPI, por lo que es sencillo controlarlo desde un procesador como Arduino.

La banda de frecuencia es de 2400 a 2525 MHz, pudiendo elegir entre 125 canales espaciados a razón de 1MHz. Se recomienda usar las frecuencias de 2501 a 2525 MHz para evitar interferencias con las redes Wifi.

La tensión de alimentación del NRF24L01 es de 1.9 a 3.6V, aunque los pines de datos son tolerantes a 5V. El consumo eléctrico en Stand By es bajo, y de unos 15mA durante el envío y recepción.

Existen dos versiones de módulos que montan el NRF24L01, uno con antena integrada en forma de zig-zag y un alcance máximo de 20-30 metros, y la versión de alta potencia que incorpora amplificador y antena externa, con un alcance máximo de 700-1000 metros.

Sin embargo, el alcance real se ve limitado por muchos factores, incluso en condiciones visibilidad directa sin obstáculos. Con el módulo de antena integrada y alimentación desde Arduino y velocidad de transmisión de 2 Mbps el alcance será apenas de 2-3 metros.

Un factor de gran impacto en el alcance es la alimentación del módulo. Para conseguir el máximo alcance conviene alimentar el módulo con una fuente externa de 3.3V, estable y con potencia suficiente.

Los módulos NRF24L01 son ampliamente empleados por su bajo precio y buenas características. Podemos emplearlos, por ejemplo, para recepción remota de sensores como temperatura presión, aplicaciones de domótica y edificios inteligentes, activación remota de dispositivos como iluminación, alarmas, y control o monitorización de robots en el rango de hasta 700 metros. [7].



Figura 3.10: NRF24L01 en sus dos versiones con Antena Impresa y Externa.

3.10 Plataforma de Desarrollo Blynk

¿Qué es Blynk? A diferencia de plataformas vistas anteriormente blynk es una solución para la construcción de aplicaciones IoT disponible para Android y iOS basada en plataformas conocidas como Arduino, Raspberry pi, ESP8266, Intel Galileo, Particle, dispositivos Seriales y Wifi entre otros, en pocos minutos puedes crear un dashboard de gran calidad, Resaltamos la gran cantidad de widgets disponibles, Control, display, Notificación y Video steaming.

Esta plataforma es de desarrollo libre y nos proporciona tutoriales y ejemplos en su página de internet con las que se puede practicar para desarrollar proyectos futuros.

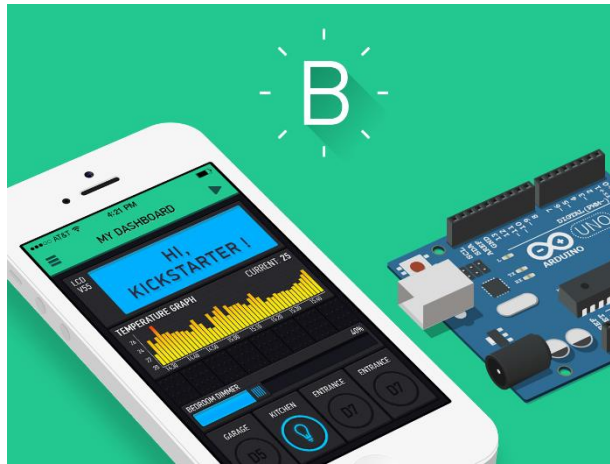


Figura 3.11: Plataforma de desarrollo Blynk.

La plataforma Blynk cuenta con una programación grafica para la creación de las aplicaciones móviles.



Figura 3.12: Iconos de programación gráfica.

La desventaja de esta plataforma es que no cuenta con visualización en la PC únicamente vía dispositivo móvil en los dos sistemas operativos más usados Android y IOS.



Figura 3.13: Sistemas operativos compatibles con Blynk.

Capítulo IV

Metodología

CAPITULO IV Metodología

En este capítulo se describen los componentes del sistema implementado, así como los criterios que se tomaron en cuenta para su diseño, basados en la teoría planteada en el Capítulo III. Esto con la finalidad que se tenga una referencia para definir proyectos a futuro en los cuales se facilite la implementación de mejoras o de nuevos proyectos que tengan como base este trabajo.

De igual manera se definen las etapas utilizadas a lo largo de la investigación, describiendo su funcionamiento y sus características principales.

4.1 Descripción general del Sistema

Se desarrollo un sistema basado en tecnología inalámbrica, para la medición, el registro y la transmisión de elemento finitos en tiempo real, sobre el comportamiento de la humedad relativa bajo invernadero, cuáles serán analizados en porcentaje y así tener un manejo adecuado de la variable en los cultivos.

El proyecto consta de una red de sensores inalámbricos de Humedad relativa que realizarán la medición y promedio de estas mediciones. Estos estarán localizados en un área específica dentro del invernadero y enviarán los datos a través de conexión RF a un módulo central con conexión Serial, wifi y Ethernet a una computadora central de administración que almacenara, controlara la variable según la necesidad que se tenga además el módulo central se conectara a internet, para el envío de la información a un dispositivo móvil compatible para Android y IOS.

Se creó una aplicación móvil utilizando una plataforma de desarrollo libre la cual proporciona un espacio de manera gratuita y segura de información en sus servidores lo que nos permite tener acceso a la información en todo momento en tiempo real.

En la siguiente figura se muestra un esquema general de la arquitectura del proyecto, cada etapa se describe de manera más detallada a lo largo del capítulo.

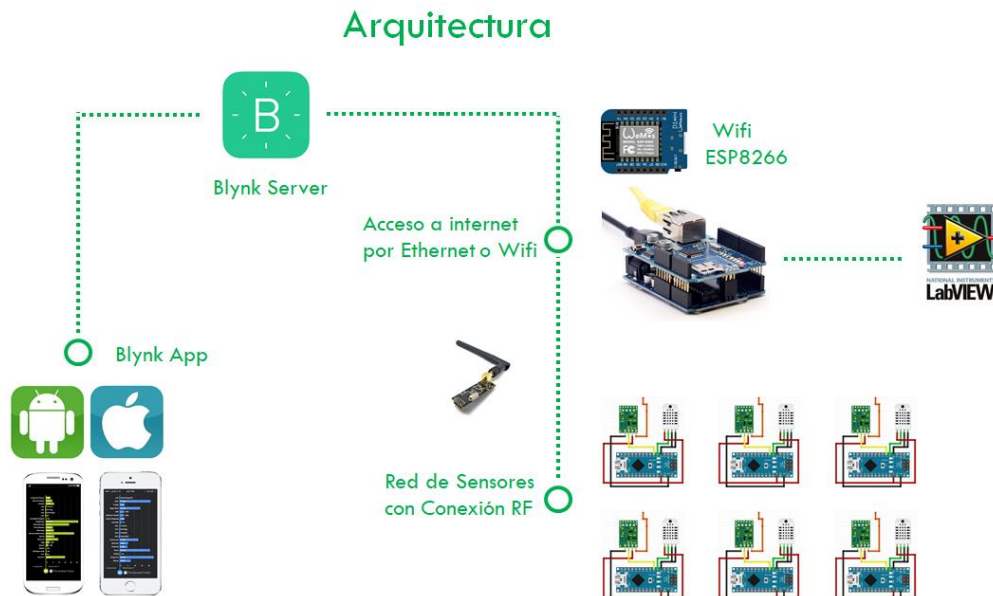


Figura 4.1: Sistema de monitoreo implementado

En este proyecto se requirió diseñar una red de sensores que permitieran hacer mediciones por elementos infinitos de humedad relativa y a la par de la temperatura para promediar las lecturas en un cierto tiempo. Estas mediciones son enviadas a través de antenas RF de mediano alcance a un módulo central el cual tiene la capacidad de presentar las mediciones en elementos finitos graficarlos y enviarlos a través de internet utilizando conexión Wifi o Ethernet a la plataforma Blynk para ser monitoreados de manera remota por medio de una aplicación móvil.

4.2 Descripción por etapas del sistema.

4.2.1 Etapa de sensado de la humedad Relativa y temperatura.

La etapa de sensado consta de un sensor de humedad relativa y temperatura modelo DHT11 o DHT22 ambos modelos tienen el mismo funcionamiento con la diferencia que el DHT22 entrega decimales en sus mediciones lo que permitirá si la aplicación de uso lo requiere tener más precisión en los datos.

La programación no cambia si decidimos utilizar un sensor u otro lo que hace versátil la etapa de sensado.

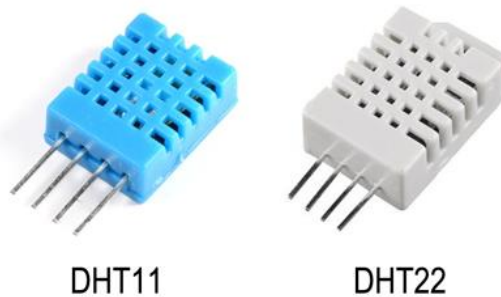


Figura 4.2: Sensor Digital DHT11 y DHT22

Para el procesamiento de la información que el sensor entrega es procesada a través de un microcontrolador que en este caso escogimos un Arduino Nano que cuenta con un microcontrolador ATmega 328 de la marca Atmel de 10 bit's el cual es suficiente para esta aplicación ya que cuenta con los recursos necesarios para procesar y enviar la información del sensado de la Temperatura y la Humedad Relativa.

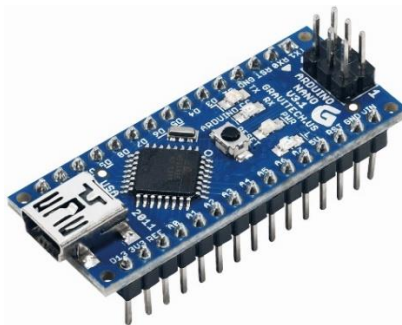


Figura 4.3: Arduino Nano

CAPITULO IV Metodología

Ahora analizaremos la parte de programación donde realizamos la parte del sensado de la temperatura y la humedad relativa.

La programación se realizó con el IDE de Arduino lo cual nos permitirá utilizar muchas de sus herramientas como el monitor serie y las librerías para los diferentes sensores y módulos que utilizaremos en el desarrollo del proyecto.

Primero necesitamos descargar la librería necesaria para trabajar con los sensores DHT, para ello entramos al IDE de Arduino y nos dirigimos a la pestaña de programa donde se abrirá una pestaña, ahí escogeremos la opción de incluir librería, se abrirá otra pestaña donde elegiremos la opción gestionar librería.

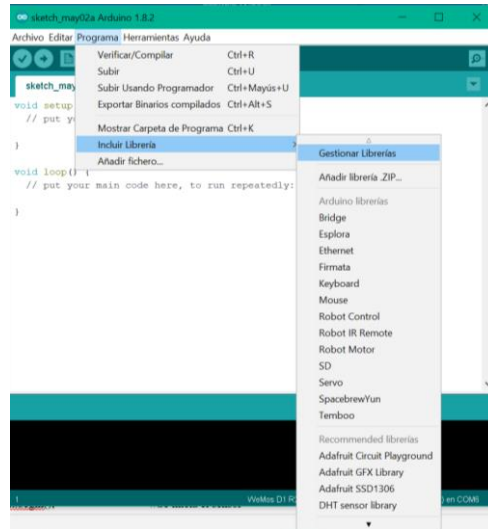


Figura 4.4: Gestionar librería en el IDE de Arduino para el sensor DHT

Se desplegará una pantalla donde buscaremos la librería simplemente escribiendo en la barra de búsqueda la palabra DHT, ahí escogeremos la primera librería y daremos instalar. La librería se descargará y se incluirá de manera automática en nuestro IDE.

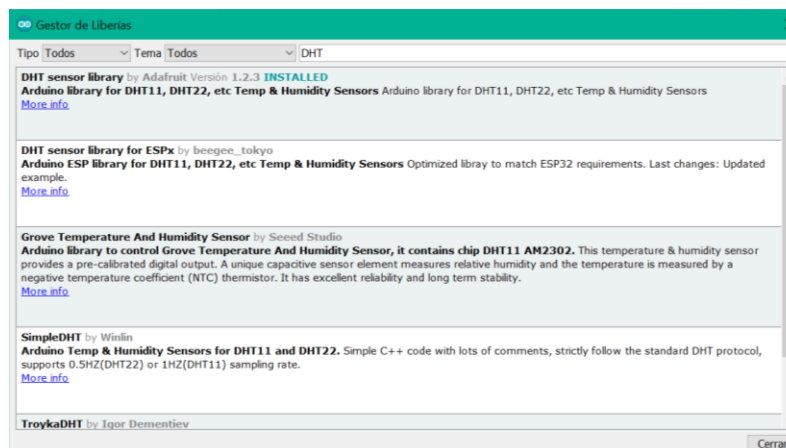


Figura 4.5: Librería DHT instalada

CAPITULO IV Metodología

Una vez descargada la librería podremos incluirla en el código de programación y ahorraremos muchas líneas de código.

A continuación, la fracción de código donde se realiza el sensado de las variables.

```
/////////librerias del DHT

#include "DHT.h"
#define DHTPIN 22
#define DHTTYPE DHT11 // Si queremos utilizar el DHT22 solo hay que cambiarlo en esta parte
DHT dht(DHTPIN, DHTTYPE);

void setup()
{
  Serial.begin(9600); //Se inicia la comunicación serial
  dht.begin(); //Se inicializa el sensor DHT
}

void loop(){
  float h = dht.readHumidity(); //Se lee la Humedad Relativa y se guarda en la variable flotante h
  float t= dht.readTemperature(); // Se lee la Temperatura y se guarda en la variable flotante t
  Serial.print("Humidity: "); // Se imprime el valor de la humedad en el monitor serie
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: "); //Se imprime el valor de la temperatura en el monitor serie
  Serial.print(t);
  Serial.print(" *C ");
  delay(500); //Se repite el ciclo cada medio segundo
}
```

Diagrama a bloques del circuito y sus conexiones

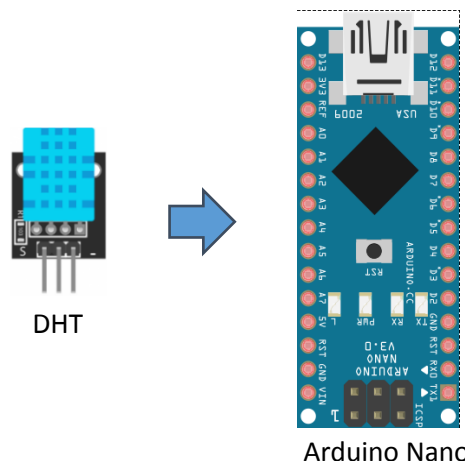


Figura 4.6: Diagrama a bloques del sensor DHT11

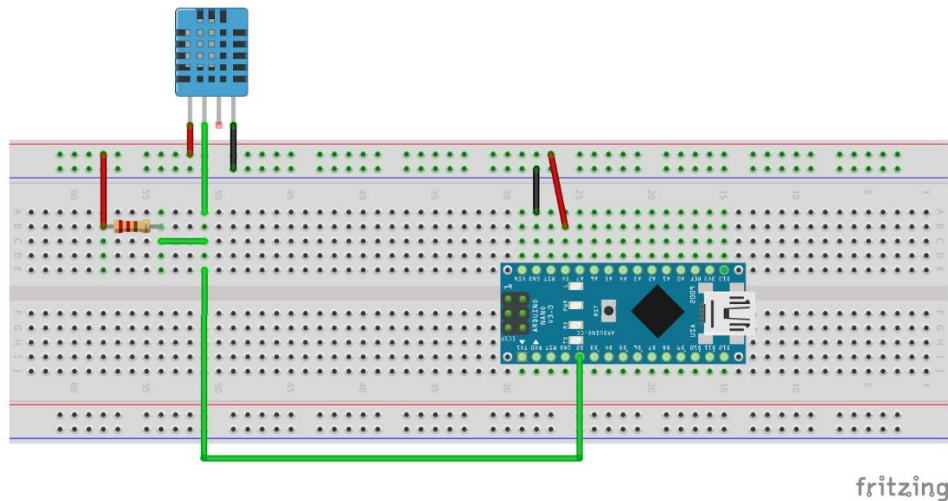


Figura 4.7: Diagrama de Conexiones en fritzing

El tiempo de muestreo está seteado a 500 milisegundos lo que nos entrega 120 muestras por minuto. Para la aplicación donde los cambios de variable son muy rápidos este tiempo de muestreo es muy bajo por lo que sería insuficiente. Por el contrario, el muestreo de Temperatura y Humedad Relativa este tiempo de muestreo suele ser demasiado alto, ya que la Humedad y temperatura en el ambiente puede llegar a tardar en mostrar un cambio en cuestión de varios minutos.

Por lo que nos es muy útil implementar un promedio de la variable de para al cabo de varias muestras nos entregue un dato ya promediado sin tener que tener miles de datos iguales por minuto.

Para implementar el promedio de la variable haremos uso de un filtro de media Móvil el cual por definición nos dice que es un cálculo utilizado para analizar un conjunto de datos en modo de puntos para crear series de promedios. Así las medias móviles son una lista de números en la cual cada uno es el promedio de un subconjunto de los datos originales. La fórmula del filtro se expresa de la siguiente manera:

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$$

El código de un filtro de media móvil desde Arduino es muy simple:

```
#include "MeanFilterLib.h"
// Simula la captura de un sensor o la lectura de un sensor
float values[] = 0;
size_t valuesLength = sizeof(values) / sizeof(values[0]);

float getMeasure()
{
    size_t static index = 0;
    index++;
    return values[index - 1];
}
```

```
// Instanciar filtro media movil con ventana tamaño 5
MeanFilter<float> meanFilter(5);

void setup()
{
    Serial.begin(115200);

    for (size_t index = 0; index < valuesLength; index++)
    {
        // Obtener medicion simulada
        float rawMeasure = getMeasure();

        // Calcular filtro
        float mean = meanFilter.AddValue(rawMeasure);

        // Mostrar resultados
        // Emplear Serial Plotter para visualización gráfica
        Serial.print(rawMeasure);
        Serial.print("\t");
        Serial.println(mean);
    }
}

void loop()
{
}
```

Hasta este punto nuestro sistema es capaz de realizar las mediciones de las variables de Temperatura y Humedad Relativa cada 500mseg. Y por medio de un filtro de media móvil podemos ajustar la cantidad de muestra a promediar para tener menos valores en un lapso de tiempo más amplio y a su vez eliminar los ruidos generados al realizar las lecturas en nuestro sensor.

Ahora el paso siguiente es saber cómo enviar esta información de manera inalámbrica a través de una antena de radio frecuencia para así poder realizar un monitoreo a distancia entre sensores y el módulo principal que funge como la parte que sube los datos a internet para ser monitoreados de manera remota o de manera local.

Para ello utilizamos un módulo de antena RF de fácil acceso y programación, el módulo NRF2401 el cual ya cuenta con sus propias librerías de trabajo en el IDE de Arduino y que en su variante de amplificador y antena exterior nos brinda un alcance de hasta 1000 mtrs a la redonda lo que nos será muy útil para establecer conexiones a distancia entre los sensores y el módulo principal.

Para hacer uso de este módulo lo primero que tenemos que hacer es descargar desde el IDE de Arduino la librería del módulo RF. Haremos el mismo procedimiento que utilizamos cuando descargamos la librería del Sensor DHT11.

Primero nos dirigimos a la pestaña de programa donde se abrirá una pestaña, ahí escogeremos la opción de incluir librería, se abrirá otra pestaña donde elegiremos la opción gestionar librería.

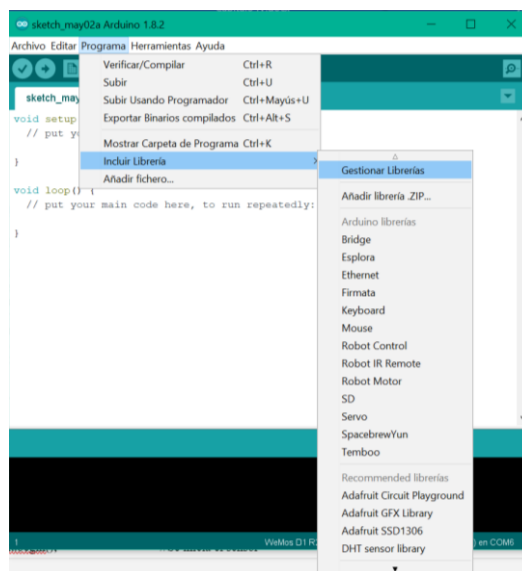


Figura 4.8: Gestionar librería en el IDE de Arduino para el Módulo NRF2401

Se desplegará una pantalla donde buscaremos la librería simplemente escribiendo en la barra de búsqueda la palabra RF24-master, ahí escogeremos la primera librería y daremos instalar. La librería se descargará y se incluirá de manera automática en nuestro IDE.

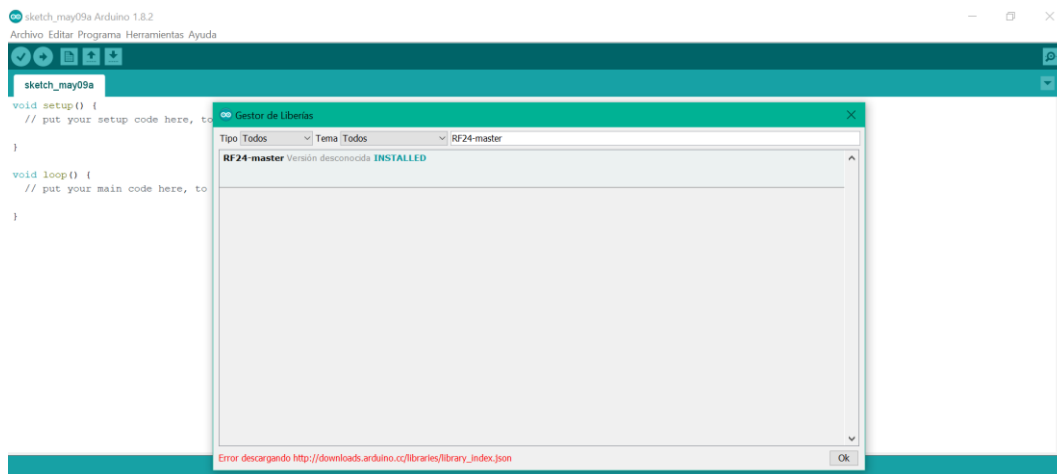


Figura 4.9: Librería RF24-master instalada

Una vez descargada la librería necesaria para el módulo NRF2401 analizaremos la sección del código de programación para la transmisión de los datos con obtenidos y promediados hacia el módulo principal.

```
#include <SPI.h>
#include <nRF24L01.h> //Librerias del modulos nRF24L01+
#include <RF24.h>
```

RF24 radio(9, 10); //Declaración de los pines de control CE y CSN para el módulo, se define "radio"

// Definicion de los canales de comunicacion

```
const uint64_t canal[2] = {0xF0F0F0F0E1LL, 0xF0F0F0F0D2LL};
char fun[2] = {'t', 'h'};
```

void setup(void)

```
{
  pinMode(10, OUTPUT);
  Serial.begin(9600);
  radio.begin(); //Inicio del modulo nRF24L01+
  radio.setRetries(15, 15); // Configuración del número máximo de reintentos
  //Apertura de los canales de comunicación
  radio.openWritingPipe(canal[1]);
  radio.openReadingPipe(1, canal[0]);
  Serial.println("Sistema de comunicación con nRF24L01");
  Serial.println("Digite la letra de la variable a consultar al receptor");
  Serial.println("\h"-Humedad, \t"-Temperatura");
}
```

void loop(void)

```
{
  radio.stopListening(); // Paramos la escucha para poder escribir
  if (Serial.available())
  {
    int data = Serial.read();
    if (data == 't')
    {
      Serial.print("Leyendo Temperatura");
      bool ok = radio.write(&fun[0], sizeof(fun[0])); //Envio de la funcion seleccionada
      if (ok)
        Serial.println("ok...");
      else
        Serial.println("failed");
      radio.startListening(); //Volvemos a la escucha
      unsigned long started_waiting_at = millis();
      bool timeout = false;
      while ( ! radio.available() && ! timeout ) // Esperamos 200ms
        if (millis() - started_waiting_at > 300 )
          timeout = true;
    }
  }
}
```

```
if ( timeout )
    Serial.println("Error, No ha habido respuesta a tiempo");
else
    { // Leemos el mensaje recibido
      float temp;
      radio.read( &temp, sizeof(temp)); //Función de lectura de datos
      Serial.print("Temperatura = ");
      Serial.println(temp);
      Serial.println("°C");
    }
}
else if (data == 'h')
{
  Serial.print("Leyendo Humedad");
  bool ok = radio.write(&fun[1], sizeof(fun[1])); //Envío de la función seleccionada
  if (ok)
    Serial.println("ok...");
  else
    Serial.println("failed");

  radio.startListening(); //Volvemos a la escucha
  unsigned long started_waiting_at = millis();
  bool timeout = false;
  while ( ! radio.available() && ! timeout ) // Esperamos 200ms
    if (millis() - started_waiting_at > 300 )
      timeout = true;
  if ( timeout )
    Serial.println("Error, No ha habido respuesta a tiempo");
  else
    { // Leemos el mensaje recibido
      float humi;
      radio.read( &humi, sizeof(humi)); //Función de lectura de datos
      Serial.print("Humedad = ");
      Serial.println(humi);
      Serial.println("%");
    }
}
}
}
```

Diagrama a bloques del circuito de sensado:

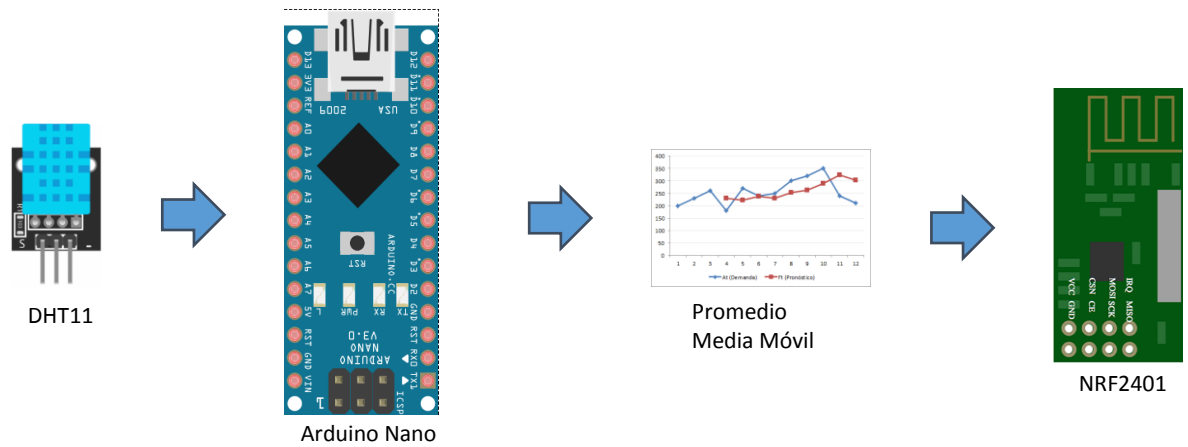


Figura 4.10: Diagrama a bloques del circuito de sensado

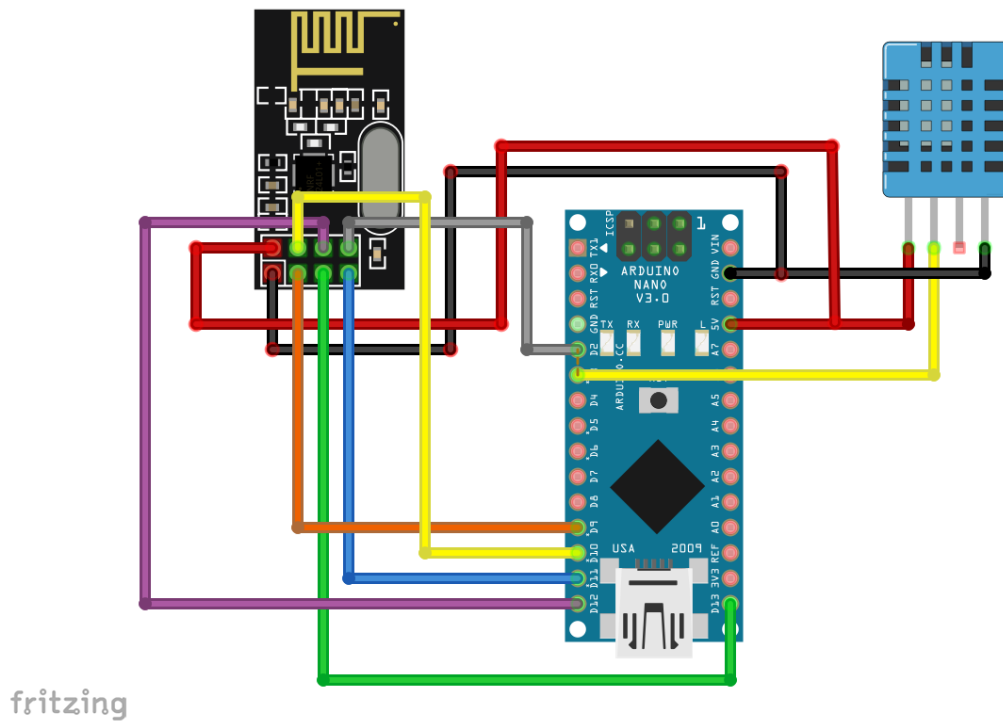


Figura 4.11: Diagrama de Conexiones en fritzing

4.2.2 Etapa de Modulo Principal.

El módulo principal de nuestro sistema tiene la función de recibir las señales enviadas por los módulos de sensores por medio de antenas RF, guarda esa información y la muestra a través de una pantalla LCD y a su vez envía esa información a la nube de la plataforma Blynk para poder tener acceso a ella en cualquier lugar utilizando de una aplicación móvil. Además, cuenta con conexión serial para conectar una computadora y poder monitorear el comportamiento de la variable mediante de un programa de monitoreo desarrollado en LabVIEW.

A continuación, se muestra un diagrama a bloques del módulo principal.

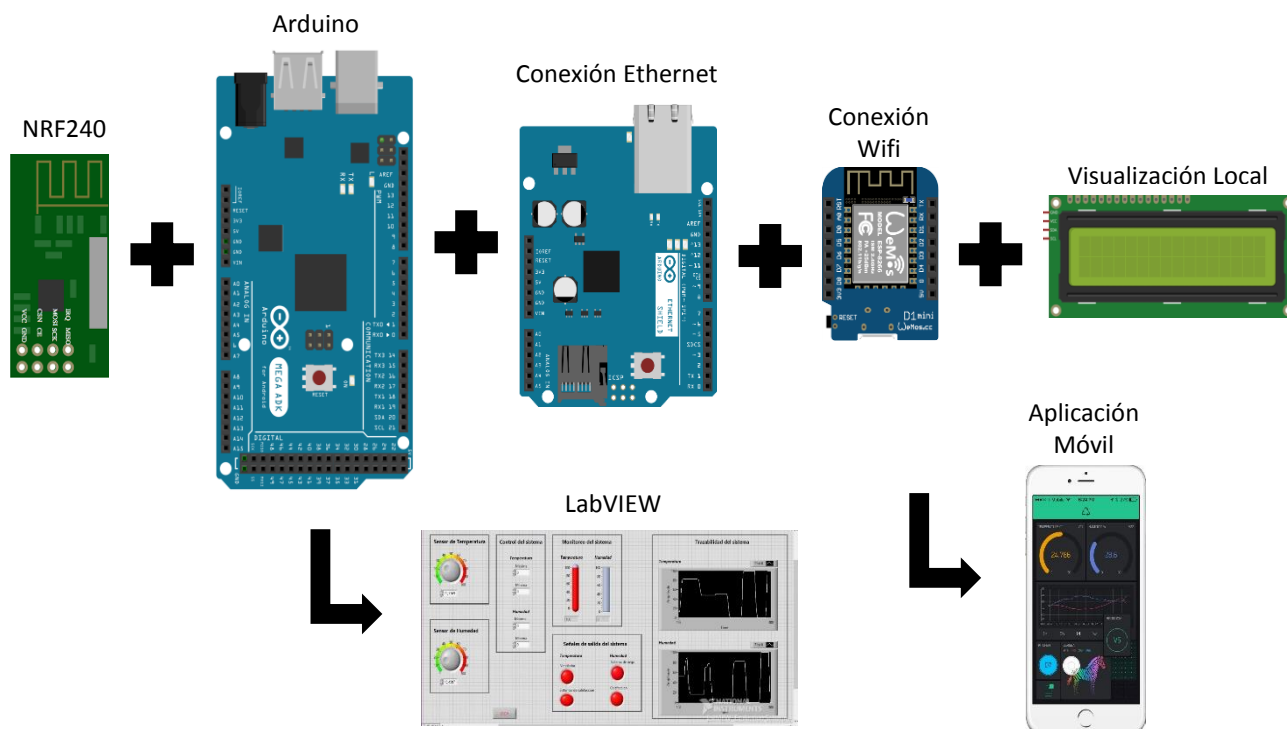


Figura 4.12: Diagrama a bloques del módulo principal

Este módulo principal tiene como característica la versatilidad de poderse conectar por medio de una señal Wifi o Ethernet a internet dependiendo de cuál es el tipo de conexión disponible en el lugar donde se instalará el módulo principal en el invernadero.

Lo primero que analizaremos de este módulo principal es la etapa de recepción de información que envían los módulos de sensores instalados a través de las antenas RF.

En esta etapa utilizamos una placa de desarrollo de Arduino modelo Mega y el módulo NRF2401

La programación de recepción es muy similar a la programación de envío solo cambiamos algunas líneas de código.

A continuación, se muestra la sección del código de la recepción de los datos.

```
#include <SPI.h> //librería para el uso del módulo SPI
#include <Wire.h> //Librería para el uso del módulo I2C
#include <LiquidCrystal_I2C.h> //Librería para el control de la LCD por I2C
#include <nRF24L01.h> //Librería del módulos nRF24L01+
#include <RF24.h>
#include "DHT.h"
#define DHTPIN 22
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);

LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); /* Configuración de la dirección i2c
* y dimensiones del módulo LCD */
RF24 modulo(9, 10); //Declaración de los pines de control CE y CSN para el módulo, se define el objeto
"modulo"
const uint64_t canal[2] = {0xF0F0F0F0E1LL,0xF0F0F0F0D2LL};
void setup(void)
{
  dht.begin();
  pinMode(10, OUTPUT); //Iniciar como salida el pin de control CSN
  modulo.begin();
  modulo.setRetries(15, 15); //Configuración del número de intentos
  modulo.openWritingPipe(canal[0]); //Apertura de los canales de comunicación
  modulo.openReadingPipe(1, canal[1]); //
  lcd.begin(16,2); //Iniciación de la LCD
  lcd.backlight();
  //Mensaje inicial
  lcd.setCursor(0,0);
  lcd.print("Comunicacion con ");
  lcd.setCursor(0,1);
  lcd.print("los modulos nRF24L01");
  modulo.startListening(); // Activando el módulo en modo de escucha
  delay(1000);
  lcd.clear();
  lcd.print("modulo iniciado.");
  delay(1000);
  lcd.clear();
  Serial.begin(9600);
}
void loop(void)
{
  //lcd.clear();
  //delay(500);
  float temp = dht.readTemperature();
  float humi = dht.readHumidity();
```



```
lcd.setCursor(0,0);
lcd.print("Temp=");
lcd.print(temp);
lcd.print(" C");
lcd.setCursor(0,1);
lcd.print("Humidity=");
lcd.print(humi);
lcd.print("%");
//Espera de peticiones
if ( modulo.available() ) // Comprobación de datos recibidos
{
  char op; // Declara la variable "op" como tipo character, puede ser "h" o "t"
  bool done = false; // Declara la variable "done" como Boolean y la inicializa en "falso"
  while (!done) // Se va a ejecutar mientras no sea verdadera o sea falsa negada = verdadera
  {
    modulo.read(&op, sizeof(op));
    done = true; // Si "op" es mayor que "0" significa que recibio un caracter
    lcd.clear();
    lcd.setCursor(0,0);
    Serial.print("Dato Recibido =");
    lcd.print(String(op));
    delay(20); // Para dar tiempo al emisor
    lcd.clear();
  }
  modulo.stopListening(); // Paramos el modo de escucha para poder escribir
  if (op == 't') //Solicitud de valor del Potenciómetro
  {
    modulo.write(&temp, sizeof(temp)); //Envío de variable solicitada
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Enviando valor de la");
    lcd.setCursor(0,1);
    lcd.print("Temperatura=");
    lcd.print(temp);
    lcd.print("°C");
    delay(300); // Espera para que se puede visualizar en la LCD
    lcd.clear();
  } else if (op == 'h') //Solicitud de valor de LDR
  {
    modulo.write(&humi, sizeof(humi)); //Envío de variable solicitada
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Enviando valor de");
    lcd.setCursor(0,1);
    lcd.print("la Humedad");
    lcd.print(humi);
    lcd.print("%");
  }
}
```

CAPITULO IV Metodología

```
delay(300); // Espera para que se puede visualizar en la LCD
lcd.clear();
} else //En el caso de recibir un dato erróneo
{
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Error!");
  lcd.setCursor(0,1);
  lcd.print("operacion no validad!");
  delay(1000); // Espera para que se puede visualizar en la LCD
  lcd.clear();
}
// Volvemos a la escucha para recibir más paquete
modulo.startListening();
}
}
```

A continuación de muestra el diagrama a bloques del Circuito:

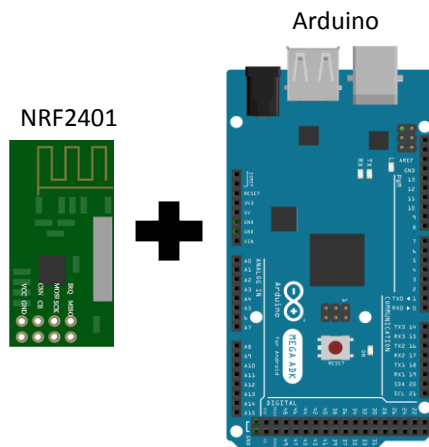


Figura 4.13: Diagrama a bloques del circuito de recepción de datos

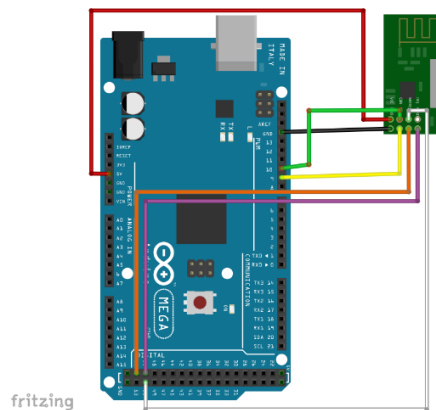


Figura 4.14: Diagrama de conexiones del circuito de recepción de datos en fritzing.

CAPITULO IV Metodología

Una vez que se reciben los datos de los promedios de las mediciones provenientes de los módulos de sensores estos se enrutan al tipo de conexión en la que esté conectado el módulo principal a internet.

El módulo cuenta con 2 tipos de conexión de internet; Wifi y Ethernet.

A continuación, vamos a describir cada uno de los tipos de conexión a internet con los que cuenta el módulo principal.

La conexión Wifi la realizamos a través de un ESP8266 el cual programamos de manera independiente lo que aligera la programación del microprocesador principal y los tiempos de ejecución en el módulo principal.

A continuación, se presenta un diagrama a bloques de la etapa de conexión wifi del módulo principal.

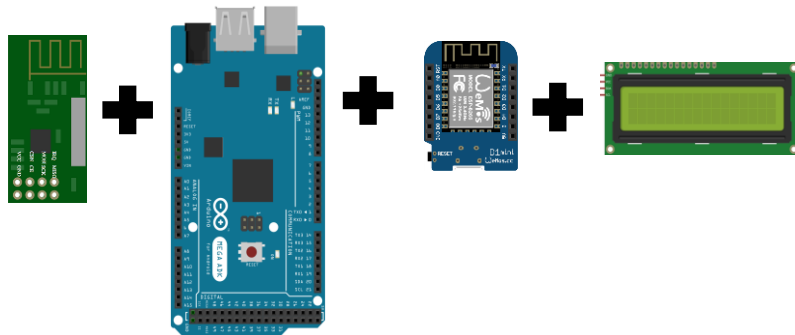


Figura 4.15: Diagrama a bloques de la etapa de conexión wifi del módulo principal

El ESP8266 es una tarjeta de desarrollo la cual puede trabajar de manera independiente como un microcontrolador principal, como un microcontrolador en paralelo o como una simple antena de transmisión de datos vía Wifi. Su versatilidad nos permite utilizarlo en múltiples aplicaciones.

En este caso utilizamos el ESP8266 de marca Wemos modelo D1Mini como un microcontrolador en paralelo el cual tiene la función conectarse de manera inalámbrica a internet y enviar los datos a la plataforma Blynk cada 500 milisegundos.

La comunicación entre el microcontrolador principal un Arduino Mega y el microcontrolador secundario ESP8266 se realiza a través de conexión serial TX y RX.

A continuación, se muestra el diagrama de conexiones entre el Arduino Mega y el ESP8266.

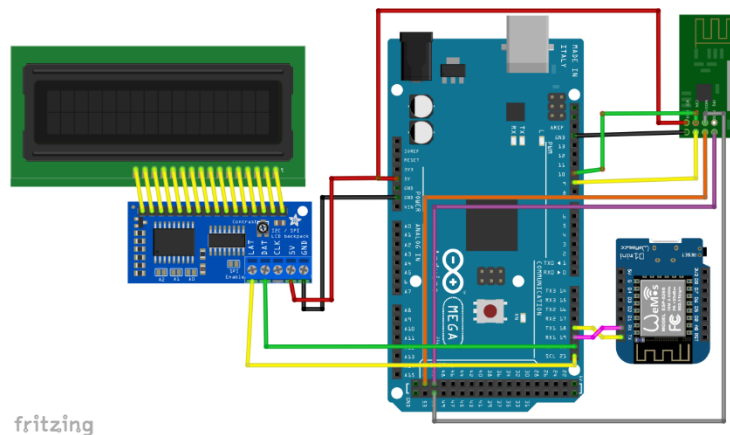


Figura 4.16: Diagrama de conexiones en fritzing.

Los datos se transmiten del Arduino Mega al ESP8266 en forma de 2 variables las cuales se guardan en este último para subirse a la nube de la plataforma Blynk donde se guardan en el servidor una base de datos que se proporciona de manera gratuita para aplicaciones de internet de las cosas.

La programación del ESP8266 de la marca Wemos nos permite desarrollar un código muy simple sin necesidad de utilizar comandos AT los cuales se utilizan para la configuración de la conexión inalámbrica. Esto nos ahorra una cantidad importante de líneas de código lo que se traduce en un tiempo de ejecución más corto.

Para poder programar nuestra placa ESP8266 directamente desde el IDE de Arduino debemos descargar algunos archivos necesarios para que el software reconozca el ESP8266 como una placa de programación compatible. Para ello debemos primero descargar la placa de desarrollo en el IDE de Arduino siguiendo la siguiente ruta:

- Abrir el IDE
- Abrir la pestaña de Herramientas
- Abrir la pestaña de Placa:
- Seleccionar Gestor de Tarjetas
- Cuando se abre la ventana del gestor Buscar la tarjeta ESP8266 y descarga

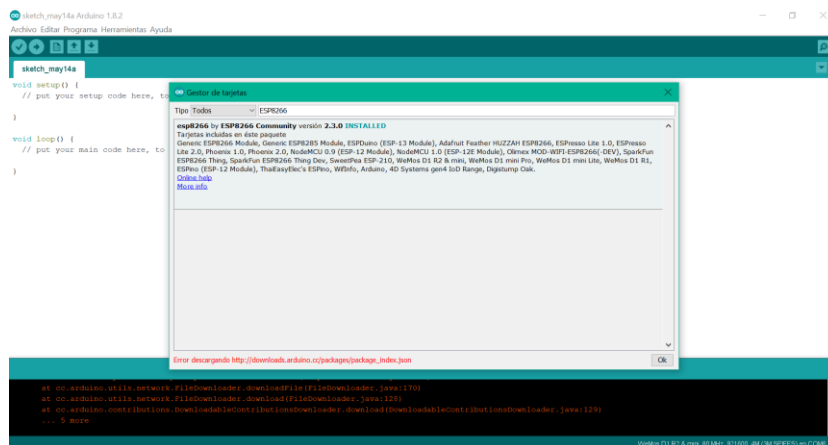


Figura 4.17: Ventana en el IDE de Arduino para descargar la tarjeta ESP8266

Una vez descargada la tarjeta ahora si podemos programar directamente en el IDE de Arduino sin problema.

Ahora solo es cuestión de cargar el código al ESP8266 el cual se muestra a continuación.

Es necesaria una librería para poder utilizar la plataforma de Blynk de la cual hablaremos más adelante, por el momento solo nos enfocaremos en la parte del código donde se realiza la transmisión de información a internet a través de una conexión Wifi.

A continuación, se muestra el código de programación del ESP8266.

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
```

```
char auth[] = "Token que la aplicación Blynk";
char ssid[] = "Nombre de tu conexión Wifi";
char pass[] = "Contraseña de tu conexión";

#define DHTPIN 2
#define DHTTYPE DHT11 // DHT 11

DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;
void sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t))
  {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  Blynk.virtualWrite(V1, t);
  Blynk.virtualWrite(V2, h);
}
void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  dht.begin();
  timer.setInterval(1000L, sendSensor);
}
void loop()
{
  Blynk.run();
  timer.run();
}
```

A continuación, el diagrama de conexiones en fritzing

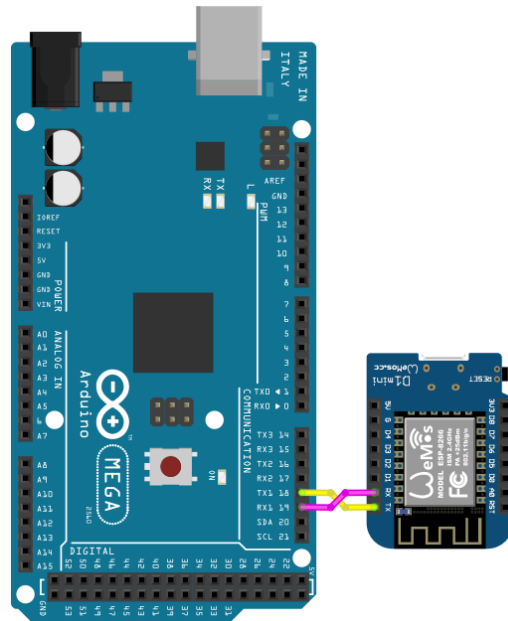


Figura 4.18: Diagrama de conexiones del Arduino y el ESP8266 en fritzing

El otro medio de conectividad es un módulo Ethernet Shield de Arduino el cual su programación al igual que el módulo del ESP8266 es muy sencilla.

Este Ethernet Shield maneja un protocolo de comunicación SPI el cual se conecta de manera directa con el Arduino Mega sin necesidad de cables.

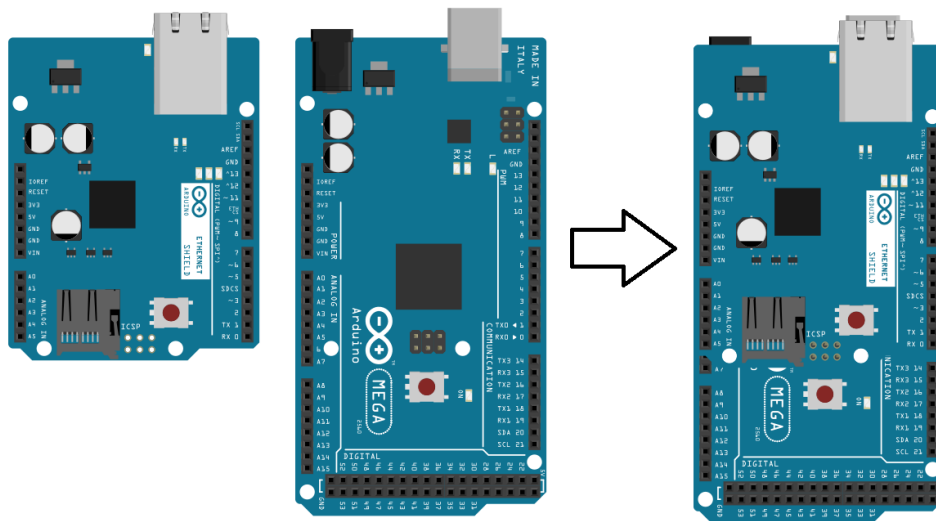


Figura 4.19: Imagen del montaje del Arduino Mega y el Ethernet Shield

La programación no requiere descargar ninguna librería especial para utilizar el Ethernet Shield solo necesitamos desarrollar el programa que nos permitió realizar la conexión a la plataforma de Blynk.

CAPITULO IV Metodología

El código que se requirió fue desarrollo utilizando como base los ejemplos que la plataforma Blynk nos proporciona lo que ayuda a aprender de una manera muy didáctica y sencilla el funcionamiento y programación en su plataforma de Internet de las cosas.

A continuación, se muestra el código de que se desarrolló para el funcionamiento del Ethernet Shield.

```
#define BLYNK_PRINT Serial
#include <SPI.h>
#include <Ethernet.h>
#include <BlynkSimpleEthernet.h>
#include <DHT.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "2d1bc61852c04b639a907e93340a7af2";

#define W5100_CS 10
#define SDCARD_CS 4

#define DHTPIN 2 // What digital pin we're connected to

// Uncomment whatever type you're using!
#define DHTTYPE DHT11 // DHT 11
// #define DHTTYPE DHT22 // DHT 22, AM2302, AM2321
// #define DHTTYPE DHT21 // DHT 21, AM2301

DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;

// This function sends Arduino's up time every second to Virtual Pin (5).
// In the app, Widget's reading frequency should be set to PUSH. This means
// that you define how often to send data to Blynk App.
void sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit
  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  // You can send any value at any time.
  // Please don't send more that 10 values per second.
  Blynk.virtualWrite(V2, h);
  Blynk.virtualWrite(V1, t);
}

void setup()
```

```
{
// Debug console
Serial.begin(9600);

pinMode(SDCARD_CS, OUTPUT);
digitalWrite(SDCARD_CS, HIGH); // Deselect the SD card

Blynk.begin(auth);
// You can also specify server:
//Blynk.begin(auth, "blynk-cloud.com", 8442);
//Blynk.begin(auth, IPAddress(192,168,1,100), 8442);

dht.begin();

// Setup a function to be called every second
timer.setInterval(1000L, sendSensor);
}

void loop()
{
  Blynk.run();
  timer.run();
}
```

El código de programación es muy similar tanto para el módulo ESP8266 como para el Ethernet Shield debido a que utilizamos la plataforma de desarrollo de aplicaciones móviles Blynk y esto nos facilita y ahorra muchas líneas de código.

A este módulo principal se le integro una pantalla LCD para contar con visualización de manera local sin necesidad de conectar una computadora nuestro sistema si lo que se pretende es solo dar una inspección rápida de los valores actuales de Humedad Relativa y Temperatura que estamos recibiendo de los diferentes sensores que tenemos en nuestra red.

Por lo que solo es necesario agregarla en las líneas de código que acabamos de revisar.

La LCD que se integró cuanta ya con una interfase que convierte el protocolo de comunicación en paralelo con la cuenta el LCD en un protocolo de comunicación I2C. Este cambio de protocolo nos permite eliminar el conflicto entre protocolos de comunicación con el Arduino y los demás módulos conectados a el.

A continuación, se muestra el Código con la implementación de la LCD

```
#define BLYNK_PRINT Serial
#include <SPI.h>
#include <Ethernet.h>
#include <BlynkSimpleEthernet.h>
#include <DHT.h>

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```


CAPITULO IV Metodología

```
// Inicializa lcd
LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7,3, POSITIVE);

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "2d1bc61852c04b639a907e93340a7af2";

#define W5100_CS 10
#define SDCARD_CS 4

#define DHTPIN 22      // What digital pin we're connected to

int LEDpin = 13; // Output

// Uncomment whatever type you're using!
#define DHTTYPE DHT11 // DHT 11
// #define DHTTYPE DHT22 // DHT 22, AM2302, AM2321
// #define DHTTYPE DHT21 // DHT 21, AM2301

DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;

// This function sends Arduino's up time every second to Virtual Pin (5).
// In the app, Widget's reading frequency should be set to PUSH. This means
// that you define how often to send data to Blynk App.
void sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature(); // or dht.readTemperature(true) for Fahrenheit

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  if (t >= 22) //Encender el Led arriba de 22°C de temperatura
  {
    digitalWrite(LEDpin, HIGH);
  }
  else
  {
    digitalWrite(LEDpin, LOW);
  }
  lcd.setBacklight(HIGH);
  lcd.setCursor(0,0);
  lcd.print("Humedad: ");
  lcd.print(h);
}
```

```
lcd.setCursor(0,1);
lcd.print("Temp:");
lcd.print(t);
  Serial.print(h); //Para LabView
  Serial.print("\t");
  Serial.print(t);
  Serial.print("\t");

// You can send any value at any time.
// Please don't send more that 10 values per second.
Blynk.virtualWrite(V0, h);
Blynk.virtualWrite(V1, t);
}

void setup()
{
  // Debug console
  Serial.begin(9600);

  pinMode(SDCARD_CS, OUTPUT);
  digitalWrite(SDCARD_CS, HIGH); // Deselect the SD card

  lcd.begin (16,2);
  pinMode(LEDpin, OUTPUT);
  digitalWrite (LEDpin, LOW);

  Blynk.begin(auth);
  // You can also specify server:
  //Blynk.begin(auth, "blynk-cloud.com", 8442);
  //Blynk.begin(auth, IPAddress(192,168,1,100), 8442);

  dht.begin();

  // Setup a function to be called every second
  timer.setInterval(1000L, sendSensor);
}

void loop()
{
  Blynk.run();

  timer.run();
}
```

CAPITULO IV Metodología

En la siguiente imagen observamos cómo es físicamente el módulo convertidor de protocolo de comunicación paralelo a I2C.



Figura 4.20: Conversor Paralelo a I2C para display LCD de 2 X 16

Una vez terminado de analizar la parte de Hardware y Software del sistema analizaremos la plataforma que desarrollo de aplicaciones móviles Blynk y como realizar paso a paso una aplicación en nuestro teléfono celular lo que nos da una gran variedad de aplicaciones en el entorno del Internet de las Cosas. Blynk es una plataforma para la creación de aplicaciones móviles cuyo ambiente de desarrollo es de programación grafica por lo que lo hace más sencillo y amigable de utilizar.

Su compatibilidad con los sistemas operativos más utilizados en el mercado de teléfonos y tables (IOS de Apple y Android de Google) nos permite tener un mayor alcance a nivel usuario.

En este proyecto se optó en utilizarla porque cuenta con mucha información disponible y guías disponibles que permiten el aprendizaje del manejo y programación de una manera más rápida y sencilla. Como primer paso necesitamos descargar la aplicación de desarrollo Blynk en nuestro teléfono móvil. Para ello solo basta con entrar en el PlayStore en caso de Andriod o en el AppStore en caso de Apple y hacer una búsqueda de esta app y descargarla.

La aplicación de desarrollo no tiene ningún costo para cualquiera de los sistemas operativos que manejemos.

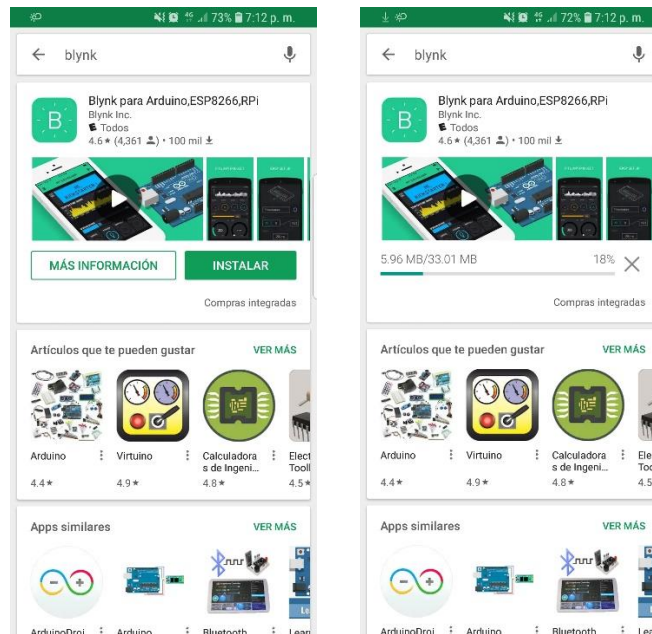


Figura 4.21: Búsqueda y descarga de la aplicación en PlayStore de Android.

CAPITULO IV Metodología

Una vez descargada e instalada la app en nuestro teléfono celular nos pedirá que nos registremos para crear una cuenta y contraseña de acceso, así como daremos de alta un correo electrónico en caso de contar con uno donde se nos proporcionara un Auth Token o número de identificación y vinculación de cada proyecto de aplicación móvil que deseemos crear. Por cada aplicación que creemos debemos recibir un Auth Token ya que este número es el que la nube de la plataforma relaciona y vincula parte virtual con la parte física del sistema.

Una vez terminado el registro entramos a la pantalla de inicio donde podemos empezar a crear aplicaciones.

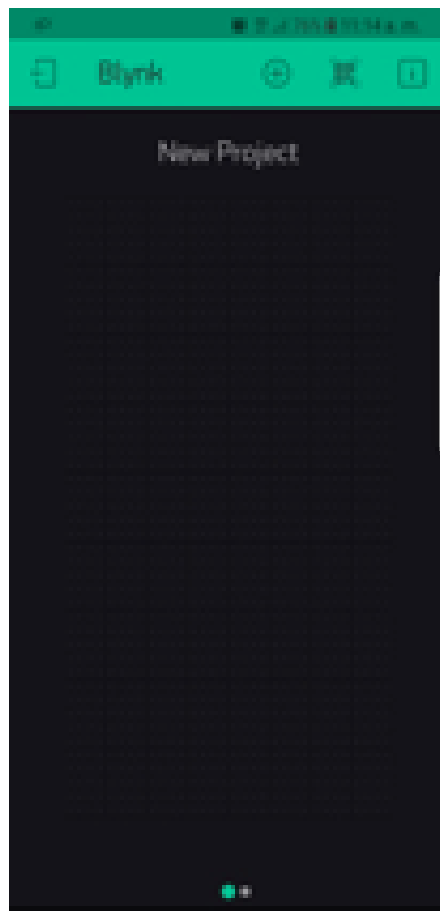


Figura 4.22: Pantalla de inicio al iniciar sesión en la aplicación de Blynk

Como segundo paso nos encontramos en la pantalla de nuevo proyecto donde iniciaremos a con la creación de nuestra aplicación.

Primero damos un click en New Project.

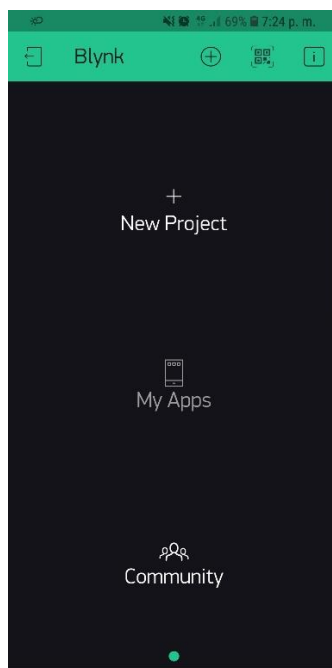


Figura 4.23: Pantalla de Nuevo proyecto

Nos desplegara una pantalla donde podemos dar el nombre al proyecto y escoger el tipo el modelo y versión del microcontrolador o tarjeta de desarrollo que vamos a utilizar.

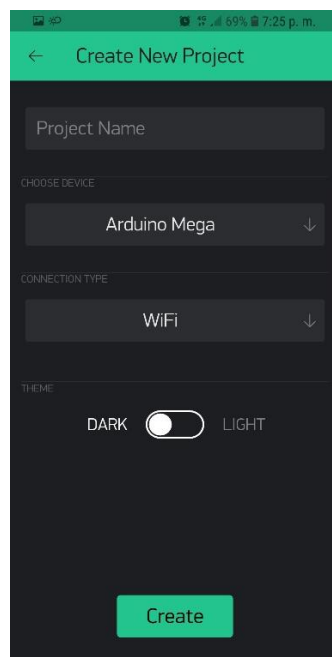


Figura 4.24: Se asigna el nombre al proyecto y se selecciona la tarjeta Arduino para utilizar en el proyecto.

CAPITULO IV Metodología

Una vez nombrado el proyecto y seleccionada la tarjeta de desarrollo, continuamos con la selección del modo de conexión a internet damos un click en Create.

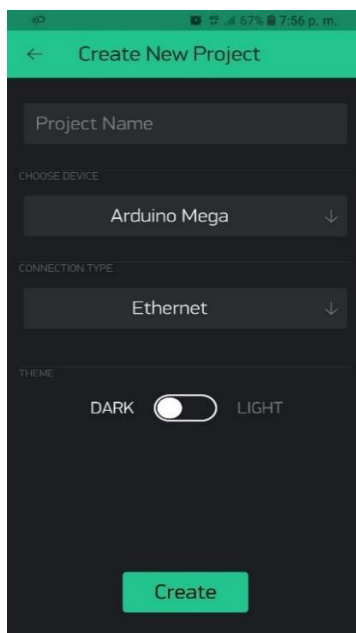


Figura 4.25: Selección del Modo de conexión.

Al crear el nuevo proyecto nos aparece una ventana donde nos confirma el envío del Auth Token al Correo que damos de alta el cual utilizamos en nuestro código de programación de Arduino.

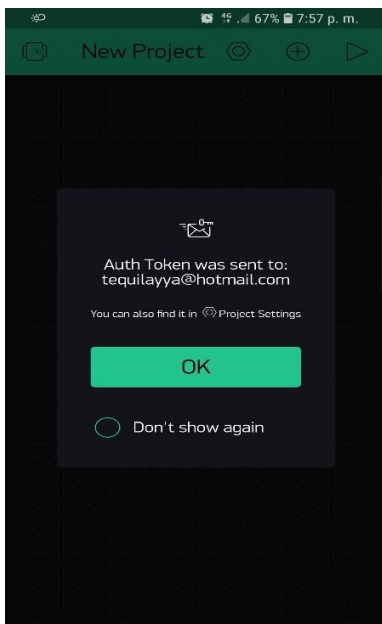


Figura 4.26: Confirmación del envío del Auth Token

CAPITULO IV Metodología

El Auth Token es una serie de caracteres alfanuméricos el cual nos llega en forma de correo electrónico y que podemos copiar y pegar dentro del código de programación de nuestro Arduino el cual nos sirve como una llave de identidad que reconoce la nube de la plataforma Blynk para identificar el proyecto, en palabras más coloquiales es la dirección virtual que nuestro proyecto físico tendrá una vez

conectada a internet.

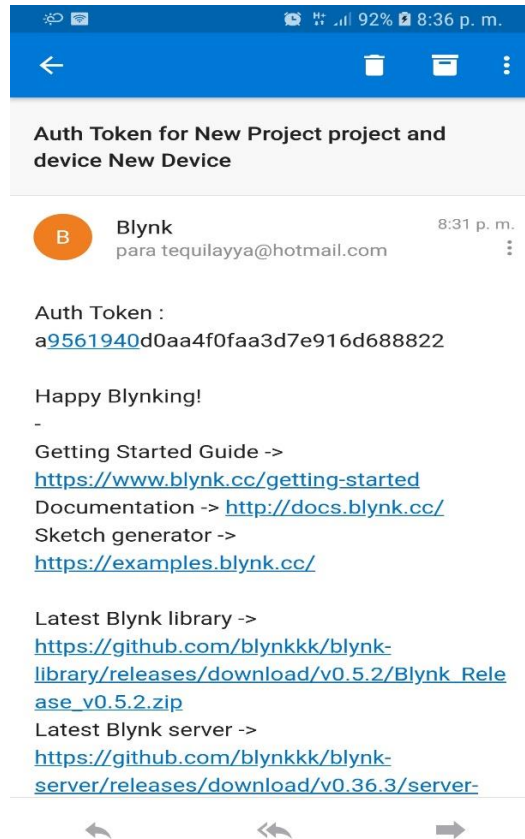


Figura 4.27: Correo recibido por parte de Blynk con el Auth Token

Como primera pantalla para iniciar con la creación de nuestra aplicación móvil tenemos la ventana donde podemos seleccionar los comandos que vamos a utilizar en nuestra aplicación. Dichos comandos tienen un valor monetario el cual al ser seleccionados para utilizarlos se nos descuenta de nuestra bolsa de dinero virtual que la aplicación nos ofrece de manera gratuita. Una versión más completa con costo para el usuario desarrollador ofrece mayor dinero virtual para crear aplicaciones más completas o más proyectos de manera simultánea.

En este caso utilizaremos la versión gratuita para generar una sola aplicación sin costo.

Esta venta de comandos está dividida en siete secciones; Controles, Displays, Notificaciones, Manejo de dispositivos, Interfaces y Otros.

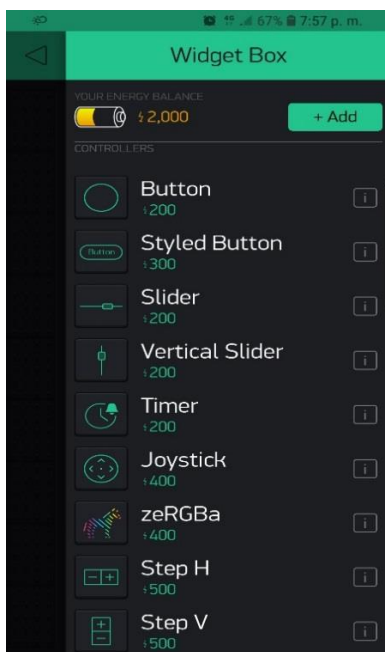


Figura 4.28: Ventana de comandos para crear aplicación

Iniciaremos Seleccionando de la sección de Displays el comando de SuperChart de tipo streaming el cual es un graficador de variables que nos registra, nos muestra y guarda la información por un periodo de tiempo que nosotros queramos.

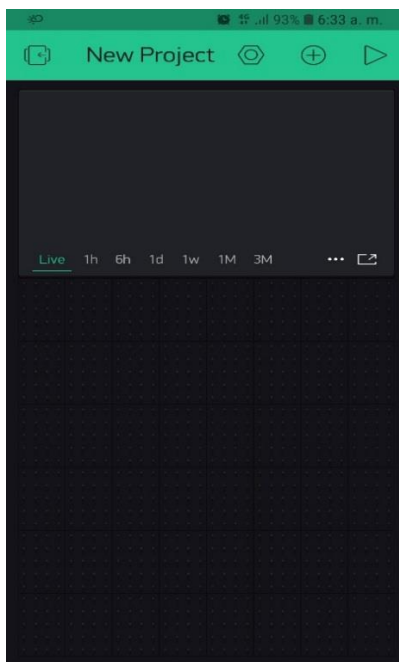


Figura 4.29: Selección del comando SuperChart

CAPITULO IV Metodología

Una vez seleccionado damos dos click sobre el comando para entrar a editar las configuraciones. En esta sección podremos asignarle un título al comando SuperChart que nos ayude a identificar el tipo de variables que vamos a graficar y registrar. Podemos editar el tamaño de la fuente del título, el color y la posición donde queremos colocar el título del gráfico.

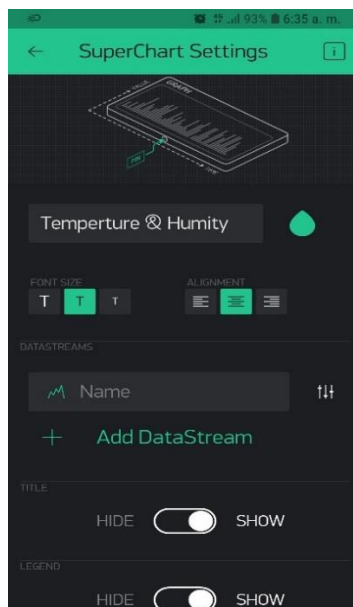


Figura 4.30: Cambio y edición del título del comando

El siguiente paso es dar de alta las variables a graficar y editar el tipo de grafico que se mostrara para ello damos Click en el apartado de DataStreams en la barra de cursor para asignarle un nombre a nuestra variable.

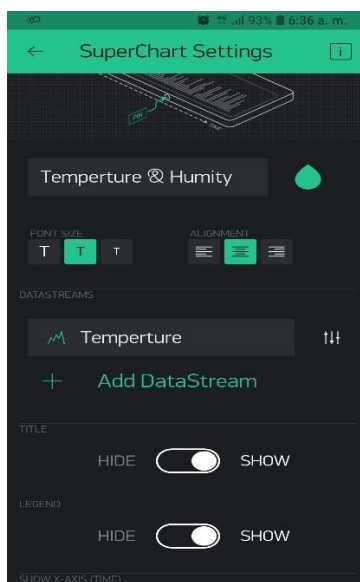


Figura 4.31: Cambio y edición del título de la variable

Después damos un click en el icono de tres barras para realizar la edición de gráfico donde podremos cambiar el tipo de gráfico y color.

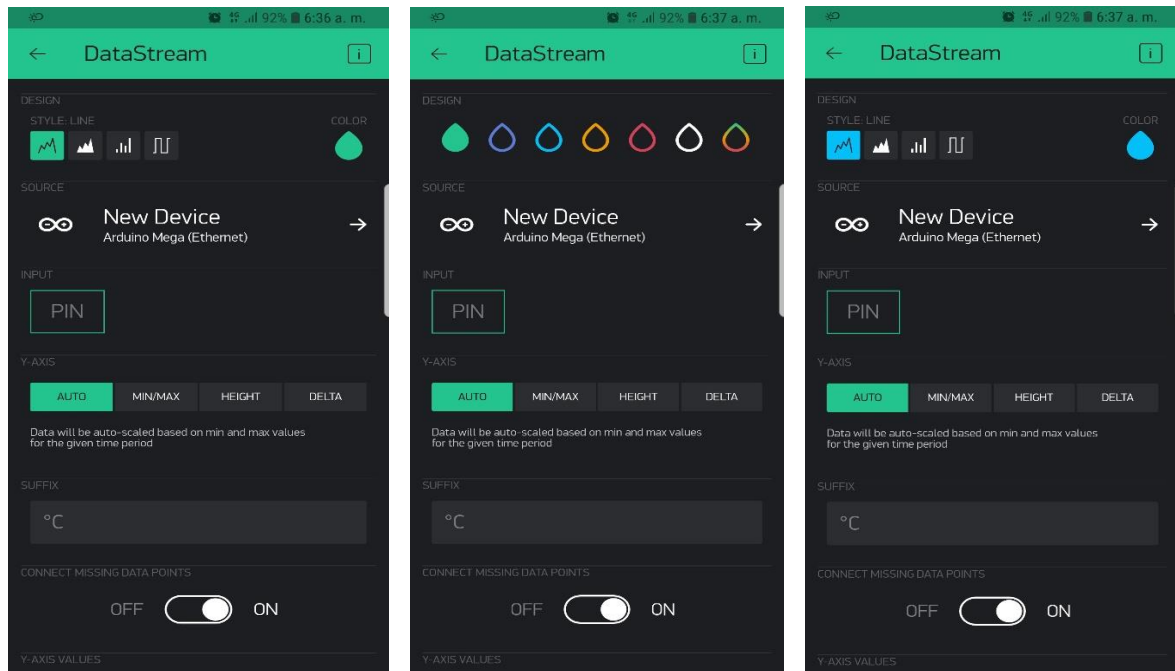


Figura 4.32: Edición del Gráfico en la aplicación

Lo siguiente es asignarle a nuestro gráfico la variable virtual que se enviara a través de la nube desde nuestro dispositivo físico Hardware a la aplicación móvil. Para lograr esta vinculación damos un click en la opción Pin y seleccionamos la opción Virtual y asignamos un número V_x , además en la opción SUFFIX podemos agregar una leyenda que queramos imprimir al final de la variable, Ejemplo, para nuestra variable a graficar de temperatura asignamos el Pin Virtual V1 y queremos que se imprima el tipo de unidad que es $^{\circ}\text{C}$.

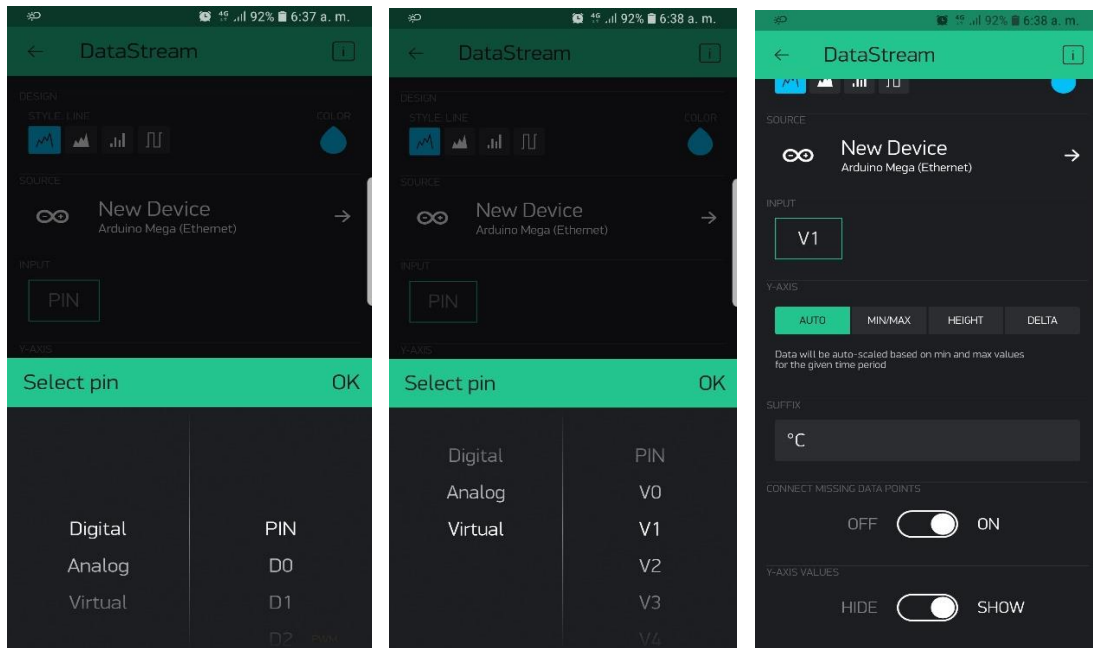


Figura 4.33: Asignación de Pin Virtual

Para asignar más variables a graficar solo es necesario dar un click en la opción + Add DataStream y repetir los pasos anteriores para editar las demás variables a graficar. Es importante señalar que no es posible repetir el mismo Pin Virtual para 2 o más variables diferentes.

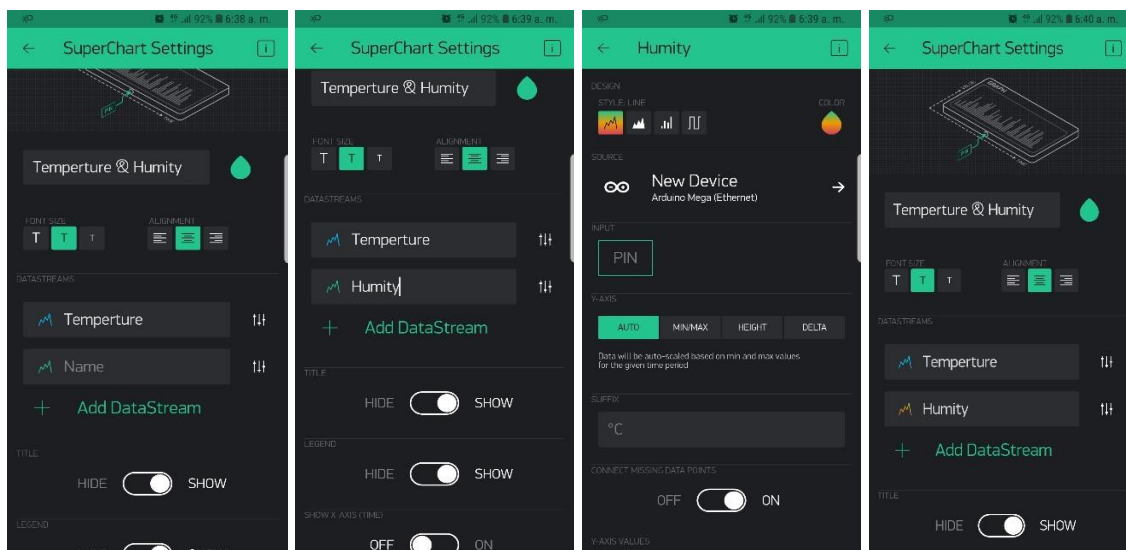


Figura 4.34: Agregar más variables a graficar.

CAPITULO IV Metodología

El último paso es definir los lapsos de tiempo que el graficador mostrara durante el tiempo que está corriendo en modo OnLine para ello editamos en la opción de TIME RANGES PICKER la resolución en tiempo real (Live) la resolución para 6 horas (6h) y la resolución para semana(1Week) además que podemos tener resoluciones de gráficos muy altos hasta de 1 año si se requiere.

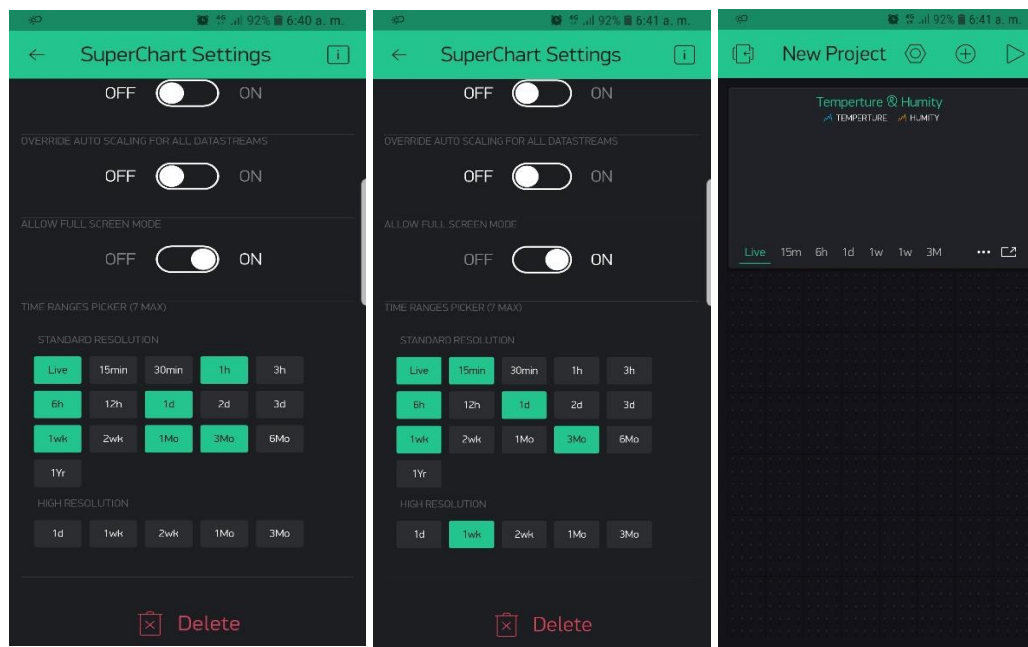


Figura 4.35: Configuración de la resolución de los gráficos.

Capítulo V

Resultados y Pantallas Desarrollas

5.1 Principales resultados alcanzados en el proyecto.

5.1.1 Descripción de la aplicación móvil desarrollada en la plataforma de Blynk

La plataforma de Desarrollo de Blynk para aplicaciones móviles es el enlace entre la parte física del proyecto y la parte virtual, además de almacenar la información en una base de datos de manera remota.

La aplicación fue diseñada y creada en un ambiente de programación 100% grafico lo que nos permitió simplificar la implementación y agregar nuevos comandos para un futuro control de variables.

La aplicación móvil obtiene la información a través de la conexión a internet del teléfono el cual se conecta al servidor utilizando un número de identificación único para cada aplicación desarrollada llamado AuthToken. Este sirve como una dirección virtual de nuestro teléfono para que el servidor sepa a qué información debe dar acceso.

Cada variable declarada en la aplicación móvil es vinculada a un numero de pin virtual el cual enlazamos en la programación del Arduino para que el servidor pueda guardarla y entregarla en el momento en que el usuario la solicita mediante la aplicación de monitoreo.

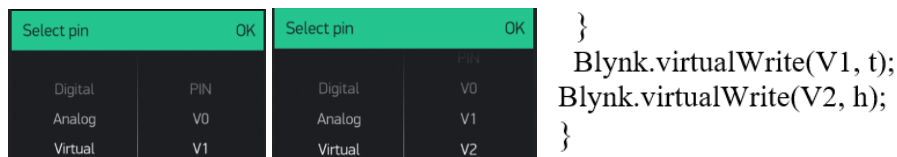


Figura 5.1: Variables Virtuales en la aplicación móvil y programa de Arduino

En la siguiente imagen podemos apreciar como la aplicación móvil obtiene la información de los sensores, la gráfica en tiempo real y genera un historial disponible hasta por un año.



Figura 5.2: Grafica de variables en la aplicación móvil.

5.1.2 Descripción de la interfase de usuario en LabVIEW

Se logró el desarrollo de una interfase de usuario en la plataforma de LabVIEW en la cual se muestra las variables graficadas en tiempo real de una manera local. La comunicación entre la interfase y el modulo principal se lleva acabo vía Serial.

La pantalla de monitoreo cuenta con dos gráficas que muestran las variables de Temperatura y Humedad Relativa de manera separada, así como un display donde se nos muestra de forma numérica el valor de cada una de las variables.

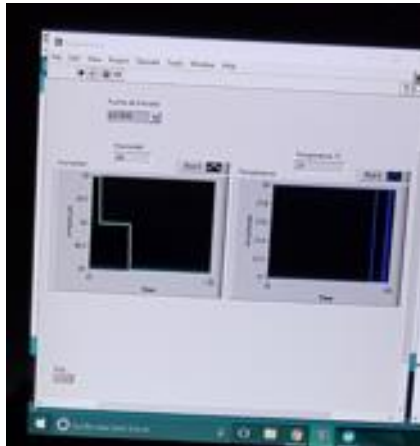


Figura 5.3: Pantalla principal de la interfase de usuario en LabVIEW



Figura 5.4: Diagrama a Bloques de la interfase de usuario en LabVIEW

5.1.3 Descripción del Módulo principal

El módulo principal tiene la capacidad de presentar los datos de las variables en 3 diferentes formas de manera local mediante un display LCD necesidad de conectarse con la interface de usuario, de manera Local con la interface de usuario a través del puerto Serial o vía remota a través internet mediante la plataforma Blynk como servidor a una aplicación móvil.

En estos dos últimos casos se puede consultar el histórico de las variables para analizar su comportamiento a través del tiempo y así poder determinar acciones de control más adecuadas según las necesidades los cultivos.

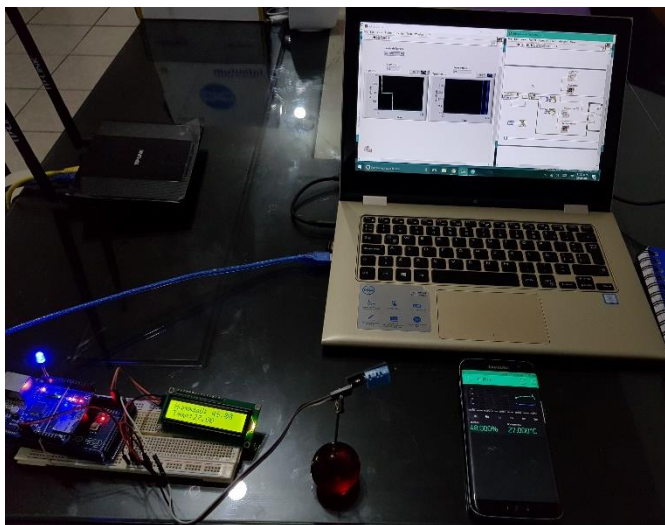


Figura 5.5: Fotografía del módulo principal y sus 3 tipos de visualización

Capítulo VI

Conclusiones y Contribuciones

6.1 Principales Conclusiones el proyecto

Se logró implementar un sistema inalámbrico de transmisión y comunicación de las variables de Temperatura y Humedad Relativa desarrollado en plataformas de desarrollo libres como lo son Arduino y Blynk donde se llegó a las siguientes conclusiones:

6.1.1 Costo beneficio del proyecto

Se concluyó que es viable desarrollar un sistema de monitoreo construido en plataformas de desarrollo libre ya que significa un muy bajo costo con referencia a los sistemas que actualmente existen en el mercado.

También se concluyó que al ser un sistema construido sobre este tipo de plataformas no es lo suficientemente robusto ya que no soporta capacidad de manejo de muchas variables además de la limitante de usuarios para la aplicación móvil, esto en comparativa con los sistemas que se ofrecen actualmente en el mercado por lo que nos limita a ser implementados sistemas de producción de cultivos más pequeños que cuentan con una menor capacidad de producción.

A pesar de esta limitante, contamos con una gran ventaja, que los micro, pequeños y medianos productores que no tienen acceso a recursos más elevados pueden implementar este sistema para la mejora de su sistema de producción sin que esto signifique que el costo beneficio sea muy bajo.

Se logró resolver el problema en la distribución e instalación de cableado para formar redes de sensores dentro de los cultivos lo que nos ahorra tiempo y sobre todo dinero en la implementación bajo invernadero ya que se utilizó un método de comunicación RF entre los sensores y el módulo principal lo aminora en gran medida pérdidas de señal por malas conexiones entre terminales alámbricas o atenuaciones por distancias muy largas.

6.1.2 Versatilidad del proyecto.

Este prototipo tiene la capacidad de rediseñarse en el sentido que podría realizar acciones de control como según convenga en los cultivos que se estén produciendo, esto hace un proyecto versátil y con futuro de mayor desarrollo para continuar con un producto más completo. La red de sensores también se puede migrar a tecnología de mayor capacidad y alcance para aplicaciones de mayor impacto para productores más grandes sin elevar considerablemente el costo de implementación.

6.2 Aportaciones del proyecto a la solución del problema.

Los pequeños productores podrán tener acceso a un sistema completo que cuenta con monitoreo a distancia a través de una aplicación móvil, así como de manera local ya sea con la ayuda de una interface de usuario o directamente del sistema en el display de visualización del módulo principal.

Este sistema se puede adaptar a cualquier necesidad ya que es un sistema de bajo costo en instalación y mantenimiento, además de ser fácil instalar y configurar.

Gracias a este sistema cualquier productor ya sea grande o pequeño podrá realizar una medición de variables en Humedad y temperatura de manera más eficiente su cultivo lo que se traducirá en mejores resultados en sus cosechas y mayor beneficio económico.

Bibliografía

- [1] H. W. A. Karl, *Protocols And Architectures For Wireless Sensor Networks*, West: John Wiley & Sons., 2005.
- [2] B. Forouzan, *Transmisión de datos y redes de comunicaciones*, Aravaca: McGraw-Hill, 2002.
- [3] F. Zhao, *Wireless Sensor Networks: An Information Processing*, Massachusetts,: Morgan Kaufmann., 2004.
- [4] P. KAMP, *Computerizer enviromental control in greenhouses:*, Massachusetts,: Guibas, L. J., 2004.
- [5] G. T. Calaza, *Taller de Arduino un enfoque práctico para principiantes*, 1er ed., Marcombo, 2014, pp. 261-262.
- [6] Ó. T. Artero, *Arduino Curso Practco de Formación*, 1ra ed., Alfaomega, 2013, pp. 534-535.
- [7] Arduino, «GitHub,» 27 06 2016. [En línea]. Available: <https://github.com/esp8266/Arduino>. [Último acceso: 01 2018].
- [8] N. Kolban, *Kolban´s Book on ESP8266*, Leanpub, 2016.
- [9] Arduino, «Arduino.CL,» 13 02 2016. [En línea]. Available: <http://arduino.cl/category/arduinoboards/arduino/>. [Último acceso: 09 2017].
- [10] F. D. S. P., «scribd.com,» 03 12 2009. [En línea]. Available: <https://www.scribd.com/doc/24421918/Protocolo-1-Wire>. [Último acceso: 18 04 2017].
- [11] L. Llamas, «luisllamas.es,» 08 12 2016. [En línea]. Available: <https://www.luisllamas.es/comunicacion-inalambrica-a-2-4ghz-con-arduino-y-nrf24101/>. [Último acceso: 22 10 2017].

ANEXO 1

Mendoza Mondragón
Fortino

Sistema de monitoreo y control de invernaderos a
través de una red inalámbrica mediante un servidor
web embebido en microcontroladores de alto
rendimiento.

2010



Universidad Autónoma de Querétaro
Facultad de Ingeniería

Sistema de monitoreo y control de invernaderos a través
de una red inalámbrica mediante un servidor web
embebido en microcontroladores de alto rendimiento

Tesis

Que como parte de los requisitos para
Obtener el grado de

INGENIERO EN AUTOMATIZACIÓN

Presenta

Mendoza Mondragón Fortino

Dirigida por

Dr. Genaro Martin Soto Zarazúa

C.U. Santiago de Querétaro, Qro. Agosto 2010

RESUMEN

En el presente proyecto se realizó un estudio de los Sistemas Web Embebidos con el objetivo de diseñar e implementar un prototipo que permita controlar y monitorear la temperatura y humedad de un invernadero desde cualquier parte de una red LAN. El estudio permitió la selección del sistema más adecuado para el proyecto. Como dispositivo de procesamiento se optó por la utilización de los kits de desarrollo de Ethernet, los cuales cumplían con todos los requerimientos del sistema. En el microcontrolador se programó el firmware del sistema en lenguaje C en el software PIC C Compiler. El código incluye la implementación de parte del stack TCP/IP. Se eligió utilizar TCP en lugar del UDP como protocolo de transporte porque TCP es confiable. La página Web diseñada se envía con las mediciones de temperatura, humedad y el estado de los dispositivos de control.

Palabras clave: LAN, TCP, IP, UDP, microcontrolador, invernadero

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar un sistema de automatización basado en microcontroladores de alto desempeño y tecnología ZigBee para monitoreo y control de invernaderos de forma local y remota a través de internet.

1.3.2 Objetivo particulares

1. Implementar un servidor web embebido en un microcontrolador de alto rendimiento para control remoto en invernaderos.
2. Desarrollar sistemas de monitoreo y control inalámbrico utilizando ZigBee para integrar una red de estaciones para medición en invernaderos.

ANEXO 2

DISEÑO DE UN SISTEMA INALÁMBRICO
PARA EL MONITOREO EN TIEMPO REAL
DE TEMPERATURA Y HUMEDAD RELATIVA BAJO INVERNADERO

JAIME ARTURO RODRÍGUEZ RAMÍREZ
CÓDIGO: 44951055

UNIVERSIDAD DE LA SALLE
FACULTAD DE INGENIERÍA DE DISEÑO Y
AUTOMATIZACIÓN ELECTRÓNICA
BOGOTÁ D.C.
2008

RESUMEN

Debido a la necesidad de monitorear variables climatológicas en cultivos, donde es de vital importancia contar con sistemas de medición que permitan optimizar los procesos y la calidad de la producción, así como responder de manera oportuna ante los riesgos que puede involucrar el impredecible comportamiento climatológico, se ha diseñado un sistema automático de medición, registro y almacenamiento de datos en tiempo real, para monitorear el comportamiento de la temperatura y la humedad relativa dentro de un invernadero.

El presente trabajo presenta el diseño de un sistema basado en tecnología inalámbrica, para efectuar la medición, registro, y transmisión de datos en tiempo real, sobre el comportamiento de la temperatura y humedad relativa en un invernadero, para ser aplicado en análisis de grados-día y el monitoreo para el manejo climatológico óptimo en un cultivo de flores. El diseño consiste en una red de dispositivos de medición, operados con baterías y por medio de microcontroladores, localizados en cada área productiva del cultivo, que entregan los datos de las mediciones vía radiofrecuencia a una computadora central de administración por software, aprovechando las ventajas que ofrece el nuevo estándar de transmisión inalámbrica IEEE 802.15.4, en cuanto a bajo costo y consumo de energía, así como facilidad de implementación y gran número de nodos.

Un sistema de medición y registro ágil, flexible y eficiente se vuelve determinante para el manejo óptimo del cultivo y contribuye de manera notable en el análisis de grados-día y otras actividades de mejoramiento desde el punto de vista de la agronomía.

ANEXO 3

SISTEMA DE CONTROL INALÁMBRICO: DISEÑO, CONSTRUCCIÓN Y APLICACIÓN EN CALDERA INDUSTRIAL

WIRELESS CONTROL SYSTEM: DESIGN, DEVELOPMENT AND APPLICATION ON INDUSTRIAL BOILER

Gerardo Cázarez-Ayala¹, Jesús Sallas-Armenta², Hugo Castillo-Meza¹, Antonio Rodríguez-Beltrán¹, Sócrates Lugo-Zavala¹, Miguel Ramírez-Montenegro¹

¹Profesor Investigador, Instituto Tecnológico de Los Mochis, Departamento de Ingeniería Eléctrica y Electrónica, Blvd. Juan de Dios Batiz y 20 de Noviembre S/N, Los Mochis, Sinaloa. ²Alumno residente profesional del Instituto Tecnológico de Los Mochis.

RESUMEN

Este trabajo de investigación y desarrollo tecnológico tiene como finalidad el diseño y construcción de una unidad de control con capacidad de comunicación en red inalámbrica y cableada, con la finalidad de desarrollar una unidad de control de procesos poderosa, fácil de utilizar, flexible y bajo costo, a través de la cual, sea posible el implementar sistemas de control distribuidos inalámbricos y cableados basados en topologías de red avanzadas como estrella, árbol y malla, dado las diferentes tecnologías de comunicación que hoy en día se manejan en el campo de la automatización de procesos industriales y tomando referencia de las más novedosas y prometedoras ventajas de las tecnologías de avanzada disponibles en el mercado internacional.

El sistema incluye el desarrollo de un driver de comunicación en dos capas, el cual fue desarrollado bajo la plataforma del microcontrolador ATmega 328 con bootloader *Arduino* y la capa superior en ambiente de desarrollo LabVIEW, con la finalidad de facilitar al usuario el desarrollo de la aplicación de control y lograr que las tareas de comunicación entre las unidades de control y cuarto de supervisión sean completamente transparentes para el programa del usuario.

Palabras Clave: Red de sensores inalámbricos, topología en Malla, wifi, zigbee.

SUMMARY

This paper has as objective to describe the design and development of a control unit with capacity to communicate with similar others units in a wireless and wired way, this with the finality of achieve to develop a powerful control unit, useful, flexible and cheap, through it be possible the implementation of distributed control systems in a wireless and wired way based in advance network topologies, like star, cluster tree and mesh, in function of the diverse communication technologies that in these times are manage in the industry process automation field and having as reference the most novelties and promising advantages of the last technologies available in the international market.

The system includes a software communication driver: develop to operate in the control unit and the PC level. This driver was implemented in two layers structure, the first one, was develop under the platform of the microcontroller based the electronic board implemented, an ATmega 328 microcontroller with an *Arduino* bootloader. The second layer of the driver is based in LabVIEW platform under Windows operating system, and this work in a personal computer. All this, with the main objective of to prove to the user a technology tool to develop task communication between the control units network and the control room and do easier the implementation of the network communications and graphical users interfaces GUI's, and transparent for the developer of the system or the programmer of the network.

Keywords: Wireless sensor network, mesh topology, wifi, zigbee.

INTRODUCCIÓN

En la unidad de control multiprotocolo desarrollada se cuenta con comunicación vía enlaces Ethernet, WIFI y Zigbee, estos han sido seleccionados por tener un gran campo de aplicaciones no solo en las industrias, sino también en cualquier lugar donde se requiera una comunicación efectiva de transferencia de datos, control y monitoreo de procesos en diversas aplicaciones como: invernadero, líneas de producción, espacios inteligentes, etc.

Hoy en día existen muchos dispositivos que presentan características muy alentadoras y demasiado llamativas para la comunicación, control y monitoreo de procesos industriales, pero están limitados muchos de ellos a solo tener la ventaja de manejar un protocolo de comunicación y algunos no tienen la opción de manipular variables, es decir, solo operan como sistemas de monitoreo y adquisición de datos y no actúan sobre un proceso. Este sistema para el control y monitoreo de procesos industriales, ofrece una amplia opción de protocolos de comunicación inalámbricos y cableados y sobre todo estandarizados en la industria del control y

ANEXO 4



INSTITUTO POLITÉCNICO NACIONAL



ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA UNIDAD ZACATENCO

” Diseño e implementación de un prototipo para monitoreo remoto con un robot móvil, a través de una cámara controlado vía Internet por WiFi”

TESIS

Que para obtener el título de Ingeniero en
Comunicaciones y Electrónica

Presentan:

**Alejandro Aranda Arsaluz
Juan Pablo Cruz Gutiérrez**

Asesores Técnicos:

**Ing. Armando Mancilla León
M. en C. Roberto Galicia Galicia**

Asesor Metodológico:

M. en C. Genaro Zavala Mejia

Objetivos

Objetivo General

Diseñar e implementar un sistema de monitoreo móvil, controlado vía Internet por medio de una página Web interactiva, que muestre al usuario lo que capta la cámara de vídeo, en tiempo real.

Objetivos Particulares

- Mostrar en tiempo real la ubicación del móvil, mediante vídeo.
- Implementar una capa de servicio para controlar el móvil desde la página Web.
- Diseñar una página Web que permita visualizar el entorno móvil.
- Realizar conexión inalámbrica al móvil y al servidor mediante el uso de la tecnología Wi-Fi.
- Agregar sobre la comunicación Wi-Fi los bits de control del móvil.
- Desarrollar un móvil electromecánico para superficies regulares que sea capaz de soportar la cámara.

Descripción Breve

Este proyecto consiste en implementar un servicio de monitoreo. El cual proporciona la facultad de desplazar un robot móvil a través del ambiente o evento que se esté monitoreando. En este caso, con el término monitoreo, implica grabar o tomar vídeo y cuando se menciona Internet, quiere decir que este vídeo tomado, será enviado o transmitido por este medio de comunicación: ya sea a uno o varios usuarios. Por otra parte se transmitirá vídeo por Internet a un usuario determinado, y también este usuario será capaz de controlar la posición del carro quien es el portador de la cámara. Todo esto mediante una página Web.

ANEXO 5

SISTEMA DE CONTROL DE HUMEDAD Y TEMPERATURA PARA INVERNADEROS

Eje temático: Telecomunicaciones.

Autores:

- Ing. Libardo Enrique Hernández Rangel, Universidad Pedagógica y Tecnológica de Colombia (UPTC), kikeher7777@yahoo.es. - Colombia
- Ing. Wilman Pineda - Universidad Pedagógica y Tecnológica de Colombia (UPTC), wilmanpineda@hotmail.com. – Colombia.
- Ing. Daríel Alejandro Bayona Ruiz- Universidad Pedagógica y Tecnológica de Colombia (UPTC) - dalbaruy@yahoo.es. – Colombia.

Resumen

Este desarrollo se origina en la Universidad Pedagógica Y Tecnológica de Colombia (UPTC) Duitama-Colombia por iniciativa de los docentes de la asignatura Microcontroladores de las Escuelas de Licenciatura en Educación Industrial e Ingeniería electromecánica, el objetivo es controlar de forma precisa la humedad, la temperatura, la iluminación, la ventilación y demás variables relevantes para los invernaderos, con la intención de lograr ambientes artificiales e ideales para el buen desarrollo de algunos cultivos. Específicamente se pensó en el cultivo de *Shiitake* seta muy exigente en cuanto a condiciones ambientales de cultivo.

Se ha desarrollado inicialmente un sistema de control que consiste de módulos de sensores en base al sensor SHT11 para Temperatura, humedad relativa y punto de rocío, módulos actuadores en base a reles de estado sólido y un módulo de control con los algoritmos de control, fusiones gráficas de agradable y útil interfaz a usuario mediante pantalla LCD gráfica y funciones de datalogger para posterior descarga y análisis de datos en PC.

Todos los módulos se intercomunican de forma inalámbrica en la banda de 2.4Ghz mediante transceiver *TRF-2.4GHZ*, se eligió la comunicación inalámbrica porque disminuye en mucho los costos y el tiempo de instalación.

En todos los módulos se eligió como centro del sistema μ controladores Microchip ya sea de la serie 16F8xx o 18F4xx.

ANEXO 6

Módulo ESP8266 y sus aplicaciones en el internet de las cosas.

CEJA-José, RENTERIA-Raúl, RUELAS-Rubén, OCHOA-Gustavo

jantoniocg13@gmail.com, rrtoriz91@gmail.com, ruben.ruelas@cucei.udg.mx, gustavo_8a58@hotmail.com

(Indicar fecha de envío: mes, día, año); Aceptado (indicar fecha de aceptación: uso exclusivo de ECORFAN)

Resumen

En la actualidad existen infinidad de dispositivos electrónicos que interactúan con nosotros en nuestra vida cotidiana y la necesidad de comunicarnos con ellos, de compartir información y realizar funciones a distancia mediante el uso de internet es cada vez mayor. Para lograr esta conexión, es necesario el uso de una interfase entre estos dispositivos y el mundo del internet. Por lo que esta investigación nos llevara a conocer más a fondo un módulo Wifi ESP8266 de bajo costo, ideal para aplicaciones de internet de las cosas (IOT) y saber cómo elegir el más adecuado para nuestros futuros proyectos, enfocados en el internet de las cosas.

Se obtuvo como resultado de este análisis, que existen factores importantes para la elección de estos dispositivos, como lo son la normativa de comunicación en cada país, la cantidad de memoria y sobre todo el costo beneficio, dependiendo del hardware con que se cuenta en el proyecto. Concluimos que antes de adquirir un módulo ESP8266 debemos realizar un análisis, tomando en cuenta estos factores para una correcta elección.

Internet de las cosas, ESP8266, Wifi.

Abstract

Nowadays, there are countless electronic devices that interact with us in our daily lives and the need to communicate with them, to share information and perform functions at a distance through the use of the internet is increasing. To achieve this connection is necessary the use of an interface between these devices and the world of the internet. So, this research will lead us to know more about an inexpensive ESP8266 Wifi module ideal for internet applications of things (IOT) and know how to choose the most suitable for our future projects focused on the internet of things.

It was obtained as results of this analysis that there are important factors for the choice of these devices, such as the communication regulations in each country, the amount of memory and especially the cost benefit depending on the hardware that is counted in the project. We conclude that before acquiring an ESP8266 module we must carry out an analysis taking these factors into account for a correct choice.

Internet of Things, ESP8266, Wifi.

Cita: CEJA-José, RENTERIA-Raúl, RUELAS-Rubén, OCHOA-Gustavo

Módulo ESP8266 y sus aplicaciones en el internet de las cosas. Revista de Investigación y Desarrollo

Introducción

Internet de las cosas

El internet de las cosas (IOT) consiste en que diferentes cosas u objetos tengan la capacidad de conectarse a internet en cualquier momento y en cualquier lugar. En un sentido más técnico, consiste en la integración de sensores y dispositivos en objetos cotidianos que estén conectados a internet a través de redes fijas e inalámbricas. De esta manera, cualquier objeto es susceptible de ser conectado y manifestarse en la red. Además, el IOT implica que todo objeto puede ser una fuente de información. (Tojeiro, 2014).

El internet de las cosas está presente en nuestro día a día. En nuestro entorno laboral, en nuestra casa, en la escuela, en el supermercado, etc.

Un ejemplo de ello sería una máquina expendedora de refrescos que funciona con la publicación de un tweet, o con un hashtag en la página de la propia empresa o con un like en Facebook.

En definitiva, el objetivo ideal del IOT sería lograr que cualquier objeto tenga vida propia a través de internet y con ello una identidad. (Tojeiro, 2014)

Desarrollo

Hipótesis

Para desarrollar aplicaciones IOT es necesario conocer e identificar las ventajas y desventajas de los dispositivos que nos llevarían a lograr nuestro propósito, la conexión a internet, por lo que iniciaremos definiendo cual será nuestra manera de conectarnos.

Planteamiento del problema

Cuando pretendemos desarrollar un proyecto de aplicación IOT debemos considerar un aspecto muy importante dentro del diseño del prototipo, que es la manera con la cual nos vamos a

conectar a la red de internet para la transmisión y/o recepción de datos vía remota.

Si la opción para dicha conexión es mediante una señal de Wifi debemos conocer a detalle que componente sería el más adecuado a las necesidades de nuestro proyecto. Por lo que podría resultar muy poco complicado ya que existen demasiados módulos en el mercado que nos podrían facilitar la parte de nuestro diseño.

La pregunta es: ¿Cuál de todos estos módulos podría ser el más adecuado?

A continuación, analizaremos una gran cantidad de estos módulos y desarrollaremos una metodología para elegir un modo Wifi adecuado a nuestras necesidades.

¿Que es Wifi?

Wifi es una tecnología de comunicación inalámbrica que permite conectar a internet equipos electrónicos, como computadoras, tablet's, smartphones o celulares, etc., mediante el uso de radiofrecuencias o infrarrojos para la transmisión de la información.

El funcionamiento aparente de una red Wi-Fi es muy similar al de una red Ethernet, solo que sin cables. No obstante, además de las direcciones IP y las direcciones MAC, en esta tecnología inalámbrica hay que tener en cuenta otros conceptos:

Estándar IEEE802.11: "Wi-Fi" se basa en este estándar, el cual es en realidad un conjunto de estándares. Dependiendo de la compatibilidad con uno o más de dichos estándares, encontraremos dispositivos que pueden formar parte de redes Wi-Fi 802.11b, 802.11g o 802.11n entre otros.

Punto de acceso (AP): un punto de acceso es un equipo de red inalámbrico (puede ser un computador con el software adecuado, o un dispositivo hardware específico) que se encarga de gestionar de forma centralizada las comunicaciones de todos los dispositivos que forman la red Wi-Fi. No solo se utiliza para

controlar las comunicaciones internas de la red, sino que también hace de puente en las comunicaciones con las redes externas (redes Ethernet e Internet), a modo de “transformador de señal” entre redes inalámbricas y cableadas.

Modo: un dispositivo Wi-Fi puede tener un rol determinado dentro de la red, y esto se configura estableciendo su modo de funcionamiento. El modo Station o (“Managed”) es el modo en el que un dispositivo es un mero cliente que se conecta a un punto de acceso para tener conectividad. El modo AP o (“Master”) es el modo en el que un dispositivo puede trabajar él mismo como punto de acceso (si dispone del firmware adecuado).

SSID: es un dato emitido por el punto de acceso que identifica la red inalámbrica a la que pertenece. En otras palabras, es el “nombre de la red” que los terminales son capaces de ver para poderse conectar.

Canal: la banda de frecuencias electromagnéticas en la que trabaja una red Wi-Fi (la banda de los 2,4 GHz, generalmente) se divide en varios canales. Concretamente, el estándar subdivide el rango de los 2,4 GHz en 14 canales separados entre sí por 5 MHz (aunque cada país aplica sus propias restricciones al número de canales disponibles. (Torrente, 2013)

ESP8266

ESP8266 es el nombre de un microcontrolador diseñado por una compañía china llamada Espressif Systems en su sede en Shanghai. Pero su producción en masa inicio hasta principios del año 2014, donde se anunció que este chip sería una excelente solución automática de redes wifi que se ofrece como puente entre los microcontroladores que hasta ahora existen o que tiene la capacidad de ejecutar aplicaciones independientes. (ESP8266, 2016)

Un ESP8266 salido de fábrica no sería de mucha utilidad ya que su producción está basada en la compactación de un chip SMT (Tecnología de Montaje Superficial por sus siglas en inglés - Surface Mount Technology) el cual viene en un

pequeño paquete de tan solo cinco milímetros cuadrados. La buena noticia es que gracias a que diversos fabricantes que construyen placas de circuito impreso prefabricadas adecuándolos y dejándolos listos para nuestro uso.

Esto nos permite trabajar con este dispositivo único acoplado a un microcontrolador, para desarrollar proyectos o como sistema autónomo para ciertas aplicaciones.

Metodología

Como ya se mencionó antes analizaremos el módulo ESP8266 ideal para aplicaciones IOT por su bajo costo, sus características, sus variantes y aplicaciones.

Iniciaremos analizando sus características generales:

Voltaje	3.3 V
Consumo de corriente	10 μ A – 170 mA
Memoria Flash	16 MB max. (512 k normal)
Procesador	Tensilica L106 32 bit
Velocidad del procesador	80 – 160 MHz
GPIOs	17
Analogico a digital	1 entrada con 10 bit de resolución (1024 valores)
Soporte de 802.11	b/g/n/d/e/i/k/r
Máximas conexiones simultáneas	5

Tabla 1: Especificaciones del Microcontrolador ESP8266

a) Procesador

El system on a chip (SoC) ESP9266EX usa un microcontrolador Tensilica Xtensa L106, que es un procesador de 32 bit con instrucciones de 16 bit.



Figura 1: Procesador ESP8266EX

El SoC describe la tendencia cada vez más frecuente de usar tecnologías de fabricación que integran todos o gran parte de los módulos que componen un computador o cualquier otro sistema informático o electrónico en un único circuito integrado o chip.

Como podemos ver a continuación:



Figura 2: Interior de un ESP8266 con periféricos anotados.

El procesador funciona por defecto a 80 MHz, pero puede ir hasta 160 MHz, tiene ~ 80kB de DRAM (Data RAM), y ~ 35kB IRAM (Instruction RAM). La IRAM se carga en el arranque con lo que el usuario quiere mantener en el procesador, aunque el procesador puede ejecutar el código directamente fuera del flash externo a una velocidad más baja.

b) Arquitectura

Tiene una arquitectura de Harvard, con lo cual la CPU puede tanto leer una instrucción como realizar un acceso a la memoria de datos al mismo tiempo, incluso sin una memoria caché. En consecuencia, una arquitectura de computadores Harvard puede ser más rápida para un circuito complejo, debido a que la instrucción obtiene acceso a datos y no compite por una única vía de memoria. (Kolban, 2016)

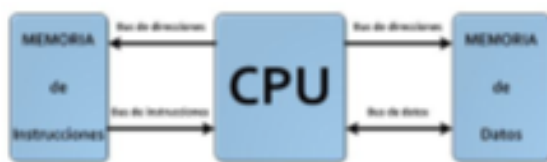


Figura 3: Arquitectura de Harvard

c) Variantes

Como ya lo hemos dicho el ESP8266 es solo un procesador, pero su versión varía a la hora de construirlo sobre una placa impresa ya que sus características de construcción difieren en diferentes aspectos. Existen diferentes marcas fabricantes de estas excelentes variantes basadas en ESP8266. AI-Thinker la empresa China es una de las más importantes, con una extensa variedad de módulos de una excelente calidad a nivel global. Wemos (Compañía China) y Olimex (Europa) también aportan sus propias versiones. Las compañías Norte Americanas Adafruit y SparkFun no se pueden quedar atrás, fabricando dos modelos más de estas poderosas tarjetas Wi-fi. A continuación, revisaremos cada una de estas variantes de la global AI-Thinker.

ESP-01:

Dimensiones: 14,30 mm × 24,80 mm
Conexiones: 8 patillas entre alimentación y GPIO
Antena impresa en la PCB sin apantallar
Alimentación: 3,3 V
Para ser precisos, las versiones más nuevas incluyen el ESP8266EX y las primitivas el modelo inicial del ESP8266 (sin EX).

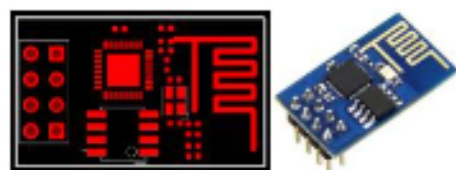


Figura 4: ESP-01

ESP-02:

Dimensiones: 14,20 mm × 14,20 mm
Conexiones: 8 conexiones de superficie (es viable soldar patillas de 0,1 ")
Sin antena en la placa, pero con un conector para antena externa sin apantallar
Alimentación: 3,3 V

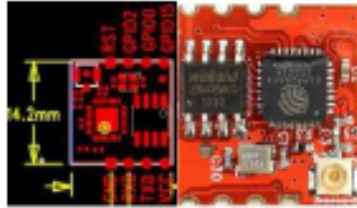


Figura 5: ESP-02

ESP-03:

Dimensiones: 17,30 mm × 12,10 mm
 Conexiones: 14 conexiones de superficie en los dos lados mayores
 Antena de tipo cerámico sin apantallar
 Alimentación: 3,3 V

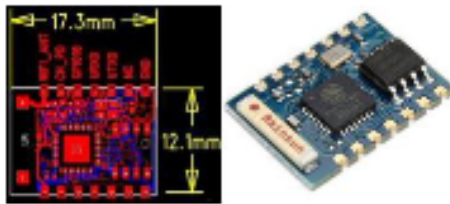


Figura 6: ESP-03

ESP-04:

Dimensiones: 14,70 mm × 12,10 mm
 Conexiones: 14 conexiones de superficie en los dos lados mayores
 Sin antena
 Apantallado
 Alimentación: 3,3 V

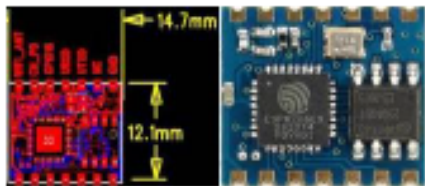


Figura 7: ESP-04

ESP-05:

Dimensiones: 14,20 mm × 14,20 mm
 Conexiones: 8 patillas separadas una décima de pulgada en una única tira
 Sin antena en placa, con un conector para antena externa
 Apantallado

Alimentación: 3,3 V

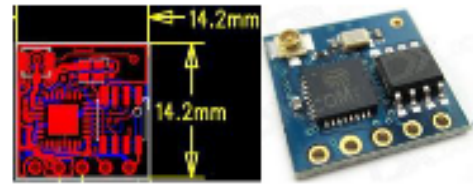


Figura 8: ESP-05

ESP-06:

Dimensiones: 14,20 mm × 14,70 mm
 Conexiones: 12 conexiones bajo la placa
 Sin antena
 Apantallado
 Alimentación: 3,3 V

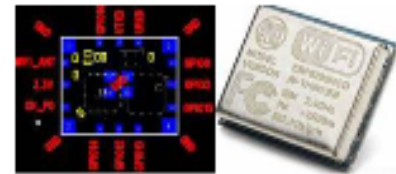


Figura 9: ESP-06

ESP-07:

Dimensiones: 20,00 mm × 16,00 mm
 Conexiones: 16 conexiones de superficie en los lados mayores de la placa
 Antena cerámica u conector para antena externa
 Apantallado
 Alimentación: 3,3 V

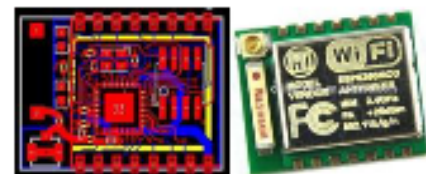


Figura 10: ESP-07

ESP-08:

Dimensiones: 17,00 mm × 16,00 mm
 Conexiones: 16 conexiones de superficie en los lados mayores de la placa
 Sin antena
 Apantallado

Alimentación: 3,3 V

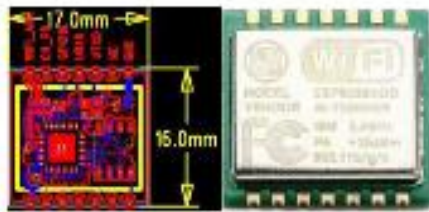


Figura 11: ESP-08

ESP-09:

Dimensiones: 10,00 mm × 10,00 mm
Conexiones: 18 conexiones bajo la placa (6 de las cuales van a masa)
Sin antena
Sin apantallar
Alimentación: 3,3 V



Figura 12: ESP-09

ESP-10:

Dimensiones: 14,20 mm × 10,00 mm
Conexiones: 5 conexiones de superficie en uno de los lados menores
Sin antena
Sin apantallar
Alimentación: 3,3 V



Figura 13: ESP-10

ESP-11:

Dimensiones: 14,20 mm × 10,00 mm
Conexiones: 5 conexiones de superficie en uno de los lados menores

Sin antena
Sin apantallar
Alimentación: 3,3 V

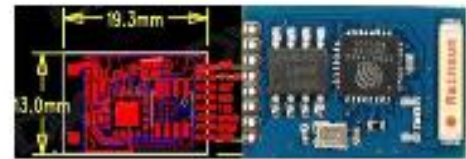


Figura 14: ESP-11

ESP-12:

Dimensiones: 24,00 mm × 16,00 mm
Conexiones: 16 conexiones de superficie dispuestas en los dos lados mayores
Antena impresa en la PCB
Apantallado
Alimentación: 3,3 V

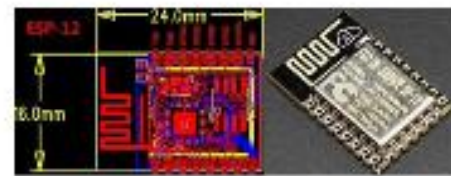


Figura 15: ESP-12

ESP-12-E — ESP-12-F:

Dimensiones: 24,00 mm × 16,00 mm
Conexiones: 22 conexiones de superficie dispuestas en tres lados (8+8+6)
Antena impresa en la PCB
Apantallado
Alimentación: 3,3 V

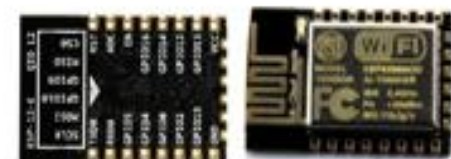


Figura 16: ESP-12-E, ESP-12-F

ESP-13:

Dimensiones: 18,00 mm × 20,00 mm
Conexiones: 18 conexiones de superficie situadas en los dos lados menores

Antena impresa en la PCB
Apantallado
Alimentación: 3,3 V

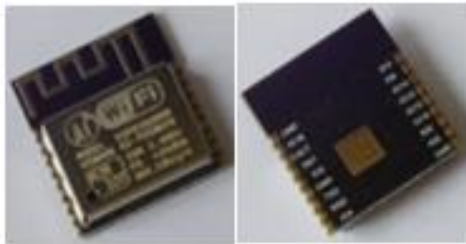


Figura 17: ESP-13

ESP-14:

Incluye un STM8 que se encarga de controlar el ESP8266 por medio de órdenes AT
Dimensiones: 24,30 mm × 16,20 mm
Conexiones: 22 conexiones de superficie dispuestas en tres lados (8+8+6)
Antena impresa en la PCB
Apantallado
Alimentación: 3,3 V

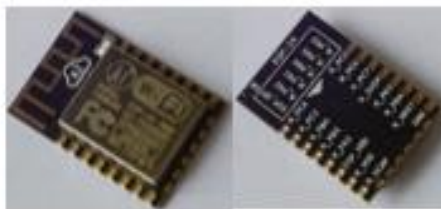


Figura 18: ESP-14

ESP-WROOM:

Dimensiones: 18,00 mm × 20,00 mm
Conexiones: 18 conexiones de superficie dispuestas en los lados mayores
Antena impresa en la PCB
Apantallado
Alimentación: 3,3 V



Figura 19: ESP-WROOM

NodeMCU:

Basado en ESP-12
Dimensiones: 30,85 mm × 47,35 mm
Conexiones: 30 patillas separadas una décima de pulgada y USB
Antena impresa en la PCB
Apantallado
Alimentación: 3,3 V y 5 V
Pulsadores user y programación (flash)



Figura 20: NodeMCU

ESP-201:

Dimensiones: 26,00 mm × 33,50 mm
Conexiones: 26 patillas separadas una décima de pulgada y distribuidas en dos tiras de 11 a cada lado (que encajan en una placa de pruebas) y 4 perpendiculares que, al estar soldadas en el mismo lado, no es posible usar en una placa de pruebas.

Antena impresa en la placa y conector para una antena externa. Debe elegirse entre una opción u otra desoldando un puente (una resistencia de 0 Ω, por defecto utiliza el conector externo por lo que la mayoría de las placas ESP-201 se venden con una pequeña (especie de) antena. Algunas versiones antiguas tienen el conector de antena mal soldado (girado 180°) y es necesario recolocarlo o la antena (externa) no sirve de nada.

Sin apantallar

Alimentación: 3,3 V

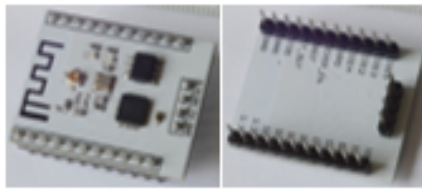


Figura 21: ESP-201

Otras compañías fabricantes de placas basadas en microcontroladores ESP son las compañías Wemos proveniente de china y la europea Olimex.

Wemos:

La D1 mini pro de WeMos es una placa de desarrollo equiparable a la NodeMCU, ya que dispone de un conector USB, puede usarse en una placa de pruebas, puede alimentarse a 5 V, tiene un botón de reset, 4 MB de memoria flash... aunque en un formato más reducido ya que tiene 16 patillas a una décima de pulgada



Figura 22: Wemos D1 Mini-Pro

Una Variante de este modelo es el D1 Mini que a diferencia del Pro este está basado en el ESP-14.

Olimex:

Olimex comercializa dos productos basados en el ESP8266: el MOD-WIFI-ESP8266 y el MOD-WIFI-ESP8266-DEV, que vienen a ser, como sus prosaicos nombres sugieren, un módulo (10 patillas) y una placa de desarrollo (22 patillas que además encajan en una placa de pruebas). Ambos dispositivos tienen 2 MB de memoria flash y una antena impresa en la placa

de circuito, pero permiten fácilmente añadir una antena externa. Olimex también ofrece un par de placas para hacer pruebas con la versión de desarrollo, pero por su precio-componentes no son muy competitivos comparándolos con otros módulos genéricos.

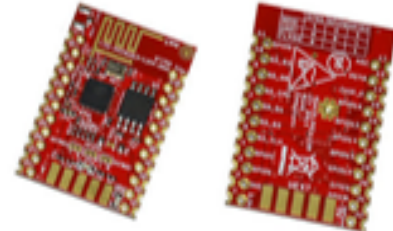


Figura 23: MOD-WIFI-ESP8266

AdaFruit:

La serie de productos de Adafruit que incluyen el SoC ESP8266 está formada por la placa HUZAZH ESP8266 breakout y la placa Feather HUZAZH (con wifi ESP8266). La primera es su opción para desarrollo y la segunda, aunque es bastante versátil, está diseñada principalmente para producción, como sugiere su conexión (más o menos estándar) a batería externa, que en la placa HUZAZH ESP8266 breakout es sólo una patilla. Además, en la opción para desarrollo, pueden encontrarse tanto un pulsador de reset, presente también en la de producción, como otro de programación que las placas Feather HUZAZH no incluyen. La placa de producción cuenta también con un conector USB

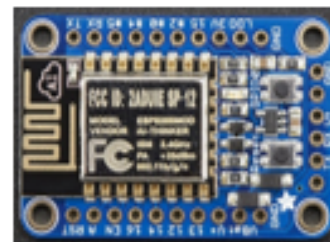


Figura 24: HUZAZH ESP8266 breakout de AdaFruit



Figura 25: Feather HUZAZH ESP8266 de AdaFruit

Ambos productos de Adafruit se pueden utilizar con patillas de un décimo de pulgada de separación y cuentan con 26 conectores en el caso de la placa Adafruit HUZAZH ESP8266 breakout y 32 conectores la placa Adafruit Feather HUZAZH. Las dos están basadas en el módulo ESP-12, así que, salvo por las características del circuito en el que se dispone (como la conexión de batería) las prestaciones son las mismas que tiene este dispositivo.

SparkFun:

La casa SparkFun comercializa cuatro productos basados en el SoC wifi ESP8266. Por un lado, ofrece el escudo para Arduino SparkFun WiFi Shield cuya principal característica es, precisamente esa: poder disponerse fácilmente sobre una placa Arduino.

Por otra parte, SparkFun también cuenta con la placa Blynk Board, cuyo objetivo principal es desarrollar productos para teléfonos móviles como una opción de entrada a la Internet de las cosas. Sus conectores son bastante sencillos de utilizar, incluso por usuarios que no estén habituados al montaje de componentes electrónicos: se pueden conectar con pinzas de cocodrilo además de en una placa de pruebas (protoboard o breadboard) y dispone de conectores verticales con guías para los sensores además de una conexión USB. Existen varios juegos de sensores para pruebas y se vende con una suscripción a una especie de nube para las aplicaciones desarrolladas con este dispositivo.

Las opciones de SparkFun que seguramente resultarán más interesantes a los desarrolladores cercanos a la electrónica son las de la serie Thing: la placa ESP8266 Thing para explotación y la placa ESP8266 Thing – Dev Board, para desarrollo. Una de las diferencias más destacables que existe entre ellas es la posibilidad de alimentar cómodamente la

versión de producción con una batería. Ambas disponen de un interruptor (no recuerdo ninguna otra placa ESP8266 que lo tenga), conexión USB, 20 espacios (10+10) para pines de una décima de pulgada y un conector para una antena externa. (Ventura, 2016)

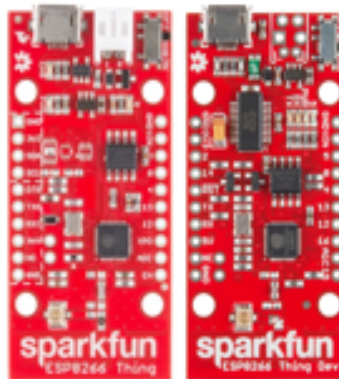


Figura 26: ESP8266 Thing (Izquierda) ESP8266 Thing-DEV (Derecha) de Sparkfun

Diferentes lenguajes de programación:

Arduino IDE

Uno de los entornos de desarrollo integrado (IDE) más usado es en definitiva Arduino ya que es una plataforma libre lo que nos permite descargar desde Internet de manera gratuita.

Además de proporcionar un editor de lenguaje C de Alto nivel, más herramientas para compilar e implementar, Arduino IDE proporciona bibliotecas pre-suministradas de rutinas C que "ocultan" complejos detalles de la implementación que de otra manera podrían ser necesarios cuando se programa.

Arduino entra en conexión con ESP8266 gracias a una cantidad de personas que han construido un proyecto Open Source Github que proporciona un "plug-in" o "extensión" a la herramienta IDE. Lo que hace esta extensión es permitir escribir bocetos en el IDE de Arduino que aprovechan las interfaces de la biblioteca Arduino que, en tiempo de compilación y despliegue, generan código que se ejecutará en el ESP8266.

El ESP8266 Arduino además incluye librerías para comunicar con WiFi utilizando TCP y UDP, crear servidores, usar un sistema de

archivos en la memoria flash, trabajar con tarjetas SD, servos, y demás periféricos.

Lua NodeMCU

Es un firmware para el ESP8266 basado en el Espressif Non-OS SDK y usa el lenguaje de programación Lua.

Lua es un potente lenguaje de scripting disponible en entornos de ESP8266. La aplicación

más popular de Lua para el ESP8266 se conoce como el NodeMCU Lua Firmware y está disponible en su repositorio GitHub.

JavaScript

El JavaScript es un lenguaje de alto nivel interpretado. Algunas de sus construcciones centrales son mecanografía suelta, orientación a objetos, soporte de funciones lambda, soporte de cierres y, lo que es más importante, se ha convertido en el lenguaje de la web.

Espruino es un proyecto de código abierto para proporcionar un tiempo de ejecución de JavaScript para dispositivos integrados. Se ha implementado para los procesadores ARM Cortex M3 / M4 y otros. Actualmente está en desarrollo para implementarlo en el ESP8266.

ESP8266 Basic

ESP8266 Basic proporciona un editor de lenguaje Basic en la propia página, ESP8266 Basic, proporciona bibliotecas pre-suministradas de rutinas que "ocultan" complejos detalles de la implementación que de otra manera podrían ser necesarios.

El entorno de desarrollo como bien se ha dicho es vía web, donde accederemos al propio ESP8266 y podremos escribir nuestro código para flashearlos. Además, dispone de un sistema de archivos en la memoria flash. (Ubach, 2017)

Ejemplo de aplicación utilizando un ESP8266

Monitoreo de temperatura utilizando una placa de Wemos, un Sensor DHT11 y la

plataforma de desarrollo de aplicaciones móviles Blynk

Un ejemplo de aplicación utilizando el microcontrolador ESP8266 montado en la placa de desarrollo de la marca Wemos modelo D1 mini, es el monitoreo de una variable de temperatura el cual se puede realizar programando directamente el microcontrolador por medio de la plataforma de desarrollo libre de Arduino.

Como primer paso descargaremos el instalador del IDE de desarrollo de Arduino directo de la siguiente página de internet oficial de Arduino: <https://www.arduino.cc/en/main/software>

Una vez descargado el instalador procedemos a instalarlo siguiendo los sencillos pasos de instalación.

Ya instalado nuestro IDE de programación para Arduino procederemos a descargar la interfase entre el IDE de Arduino y la placa de desarrollo Wemos D1 mini siguiendo la siguiente ruta:

- Abrir el IDE
- Abrir la pestaña de Herramientas
- Abrir la pestaña de Placa:
- Seleccionar Gestor de Tarjetas
- Cuando se abra la ventana del gestor Buscar la tarjeta ESP8266 y descargar.

Ahora ya podemos programar de manera directa nuestra placa de desarrollo D1 mini como si fuera una placa de Arduino.

Para este ejemplo de aplicación utilizaremos la plataforma de desarrollo de aplicaciones móviles Blynk la cual es libre en su versión de uso básico. Para desarrollo esta aplicación debemos descargar algunas librerías de Blynk y del sensor a utilizar, siguiendo la siguiente ruta en el IDE de Arduino:

- Abrir la pestaña de Programa
- Abrir la pestaña de incluir librería
- Abrir la pestaña de gestionar librería

- Cuando se abre la ventana del gestor de librerías buscamos la librería Blynk y la descargamos
- Después descargamos la librería de BlynkESP8266
- Y para finalizar descargamos la librería DHT11.

Una vez realizada la descarga de las librerías, descargamos en nuestro celular la aplicación Blynk del Appstore para Apple o PlayStore en Android según sea nuestro teléfono celular y creamos una cuenta y contraseña para obtener un número Token que incluiremos en el programa del microcontrolador.

El siguiente paso es construir el circuito utilizando el siguiente diagrama de conexiones de Fritzing.

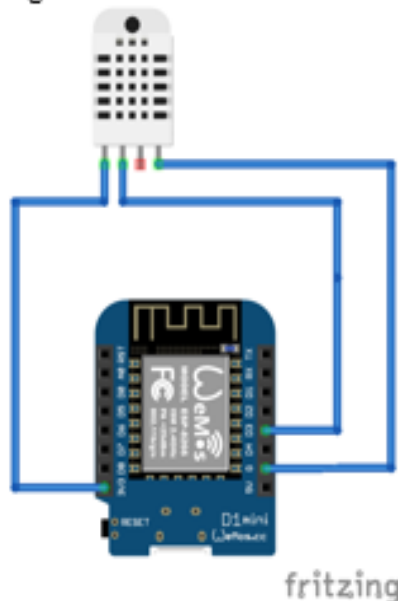


Figura 27: Diagrama de conexiones

Ahora creamos un nuevo Sketch de programación en el IDE y escribimos el siguiente código:

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <DHT.h>
```

```
char auth[] = "Token que la aplicación Blynk";
char ssid[] = "Nombre de tu conexión Wifi";
```

```
char pass[] = "Contraseña de tu conexión";

#define DHTPIN 2
#define DHTTYPE DHT11 // DHT 11

DHT dht(DHTPIN, DHTTYPE);
BlynkTimer timer;
void sendSensor()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  if (isnan(h) || isnan(t))
  {
    Serial.println("Failed to read from DHT
sensor!");
    return;
  }
  Blynk.virtualWrite(V6, t);
}
void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
  dht.begin();
  timer.setInterval(1000L, sendSensor);
}
void loop()
{
  Blynk.run();
  timer.run();
}
```

Y se lo cargamos a nuestra placa de desarrollo Wemos D1 mini de la misma manera como cargamos un programa en una placa de Arduino.



Figura 28: Cargando el programa al DI mini utilizando IDE de Arduino

A continuación, creamos la aplicación móvil para el monitoreo con la plataforma Blynk. Podemos utilizar un graficador en tiempo real y un display para que nos muestre el dato de manera numérica como se muestra en la siguiente imagen.



Figura 29: Aplicación Móvil creada con la plataforma Blynk

Una vez implementado nuestro sistema de monitoreo podremos tener las lecturas de la temperatura en cualquier lugar a través de internet casi en tiempo real ya que se genera un pequeño retraso en la transferencia de información ocasionada por la velocidad de la conexión a internet la cual puede variar entre 1 y 6 segundo.

Resultados

Una vez que analizamos cada una de las marcas más reconocidas a nivel mundial y sus variantes en cada una de ellas al igual de cada uno de los lenguajes disponibles de programación, nos pudimos dar cuenta en principio que cualquiera de los módulos wifi ESP8266 puede programarse utilizando los lenguajes que hay disponibles para ellos (también cualquier IDE) así que la elección depende principalmente de la implementación hardware.

El matiz más relevante que podría hacerse a esto sería, en todo caso, la cantidad de memoria disponible.

Aunque inicialmente algunos módulos estaban estrechamente vinculados con ciertos lenguajes o entornos de desarrollo, el ejemplo más claro de esto es NodeMCU y el lenguaje Lua, actualmente es relativamente sencillo usar cualquiera de los disponibles con cualquier módulo.

Otro criterio relevante que considerar es la normativa de uso ya que es diferente en cada país lo que puede ocasionar problemas si se quiere comercializar.

Un punto importante para tener en cuenta es también es el lenguaje de programación a utilizar. Hemos visto que Lua ModeMCU es una opción muy completa IDE de Arduino es un lenguaje más simple para el momento de realizar la programación y su IDE esta complementada disponible desde el internet.

Discusión

Una buena Opción para nuestros proyectos es el la ESP-201 de AI-Thinker es una versión barata de la placa de desarrollo NodeMCU, si no tenemos problemas para conseguir un adaptador USB-UART, la versión que se comercializa con las patillas soldadas a la placa suele disponer una de las tiras de manera que no se puede insertar en una breadboard salvo que se retire y se suelde en la superficie contraria.

Cuando sea necesario desarrollar una aplicación donde se requiera una buena cantidad de memoria, más de los 4 Mbit habituales, puede utilizarse una versión especial del ESP8266, el

módulo ESP-12F de AI-Thinker, que dispone de una memoria flash de 32 Mbit (además de 96 Kbytes de RAM) además es el más barato que he encontrado con esa memoria.

Si en el montaje ya hay un MCU, es decir, el ESP8266 no es el microcontrolador principal del dispositivo, la primera versión del módulo, la ESP-01, es perfectamente funcional e imbatible en precio, especialmente si se compran lotes de varios módulos ya que su costo va desde los 2 dólares.

Agradecimiento

Los autores de este trabajo queremos extender nuestro agradecimiento al departamento de posgrado del Instituto Tecnológico de Ciudad Guzmán por el apoyo brindado para llevar a cabo esta investigación.

Conclusiones

Como conclusión determinamos que existen diferentes aspectos importantes a considerar antes de escoger un módulo para nuestro proyecto. Inicialmente debemos revisar la normativa que rige dentro del país donde queremos implementar y producir si ese es el caso nuestro producto a desarrollar.

Otro punto importante es cual es mejor lenguaje de programación dependiendo de nuestra experiencia y habilidad para la elaboración del software.

Otro punto importante es la memoria que se requiere en nuestro módulo ya que dependiendo de la aplicación puede ser importante que no nos falte capacidad.

Y por último es determinar es el costo-beneficio del módulo a comprar, ya que si nuestro proyecto ya cuenta con un microcontrolador principal podemos adquirir un módulo ESP más pequeño para que solo trabaje a la par con nuestro microcontrolador principal.

Referencias

Tojeiro Calaza, German. (2014) *Taller de Arduino un enfoque práctico para*

principiantes".(1ra ed.), MARCOMBO, pp 261-262.

Torrente Artero, Oscar. (Febrero 2013) *Arduino Curso práctico de formación*, (1ra ed.), Alfaomega, pp. 534-535.

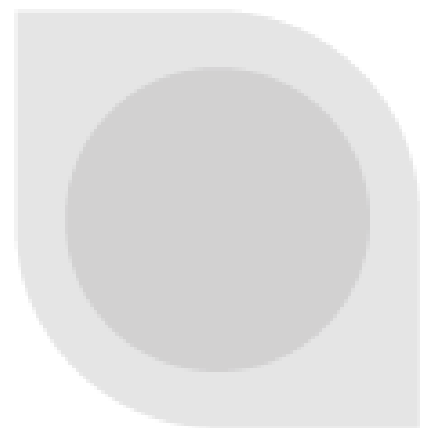
ESP8266. (27/06/2016) [Artículo], España, <https://github.com/esp8266/Arduino>

Kolban, Neil. (2016) *Kolban's Book on ESP8266*, (1ra ed.).

Ventura, Víctor (2016 junio 2).[en línea]. Disponible en: <https://polaridad.es/esp8266-modulo-wifi-elegircaracteristicas/>

Ubach Granados, Carles. (abril 2017). *Programando directamente un ESP8266*, Memoria de tesis Ingeniería Electrónica,

ANEXO 7



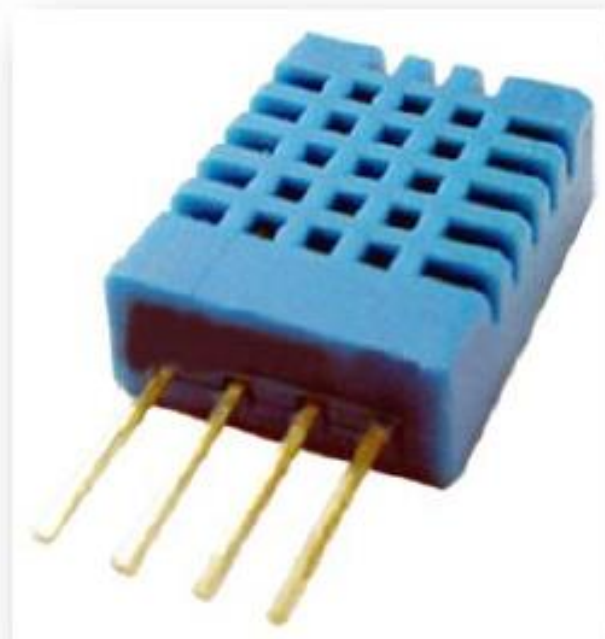
DHT11 Humidity & Temperature Sensor

DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output.

DHT 11 Humidity & Temperature Sensor

1. Introduction

DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.



Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package. It is convenient to connect and special packages can be provided according to users' request.

2. Technical Specifications:

Overview:

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 °C	±5%RH	±2°C	1	4 Pin Single Row

Detailed Specifications:

Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			±1%RH	
Accuracy	25°C		±4%RH	
	0-50°C			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0°C	30%RH		90%RH
	25°C	20%RH		90%RH
	50°C	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25°C, 1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
Temperature				
Resolution		1°C	1°C	1°C
		8 Bit	8 Bit	8 Bit
Repeatability			±1°C	
Accuracy		±1°C		±2°C
Measurement Range		0°C		50°C
Response Time (Seconds)	1/e(63%)	6 S		30 S

3. Typical Application (Figure 1)

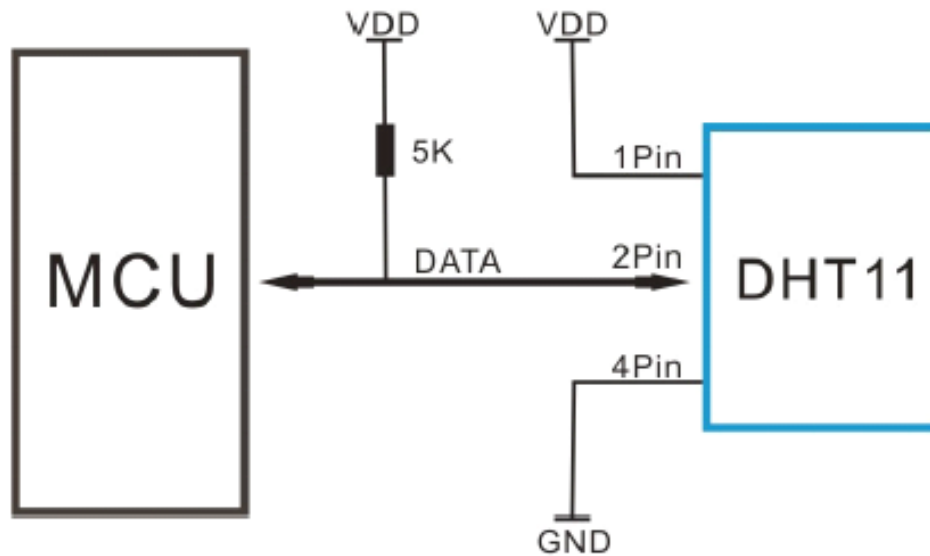


Figure 1 Typical Application

Note: 3Pin – Null; MCU = Micro-computer Unite or single chip Computer

When the connecting cable is shorter than 20 metres, a 5K pull-up resistor is recommended; when the connecting cable is longer than 20 metres, choose a appropriate pull-up resistor as needed.

4. Power and Pin

DHT11's power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor in within one second in order to pass the unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

5. Communication Process: Serial Interface (Single-Wire Two-Way)

Single-bus data format is used for communication and synchronization between MCU and DHT11 sensor. One communication process is about 4ms.

Data consists of decimal and integral parts. A complete data transmission is 40bit, and the sensor sends higher data bit first.

Data format: 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum. If the data transmission is right, the check-sum should be the last 8bit of "8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data".

5.1 Overall Communication Process (Figure 2, below)

When MCU sends a start signal, DHT11 changes from the low-power-consumption mode to the running-mode, waiting for MCU completing the start signal. Once it is completed, DHT11 sends a response signal of 40-bit data that include the relative humidity and temperature information to MCU. Users can choose to collect (read) some data. Without the start signal from MCU, DHT11 will not give the response signal to MCU. Once data is collected, DHT11 will change to the low-power-consumption mode until it receives a start signal from MCU again.

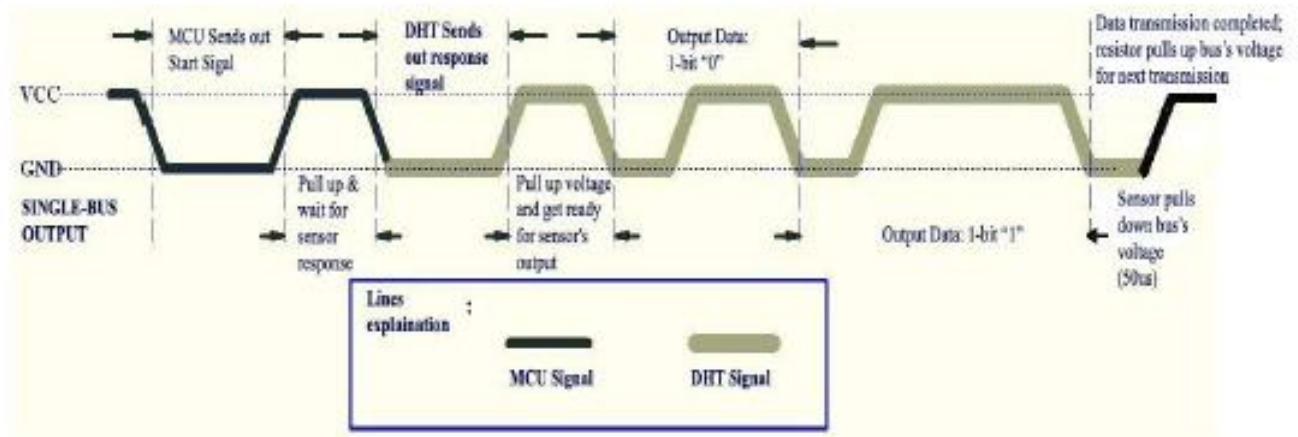


Figure 2 Overall Communication Process

5.2 MCU Sends out Start Signal to DHT (Figure 3, below)

Data Single-bus free status is at high voltage level. When the communication between MCU and DHT11 begins, the programme of MCU will set Data Single-bus voltage level from high to low and this process must take at least 18ms to ensure DHT's detection of MCU's signal, then MCU will pull up voltage and wait 20-40us for DHT's response.

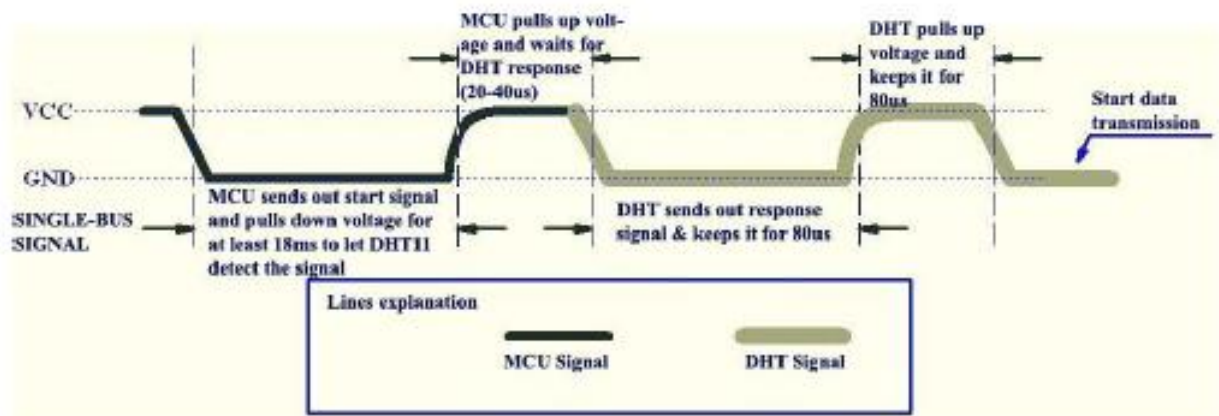


Figure 3 MCU Sends out Start Signal & DHT Responses

5.3 DHT Responses to MCU (Figure 3, above)

Once DHT detects the start signal, it will send out a low-voltage-level response signal, which lasts 80us. Then the programme of DHT sets Data Single-bus voltage level from low to high and keeps it for 80us for DHT's preparation for sending data.

When DATA Single-Bus is at the low voltage level, this means that DHT is sending the response signal. Once DHT sent out the response signal, it pulls up voltage and keeps it for 80us and prepares for data transmission.

When DHT is sending data to MCU, every bit of data begins with the 50us low-voltage-level and the length of the following high-voltage-level signal determines whether data bit is "0" or "1" (see Figures 4 and 5 below).

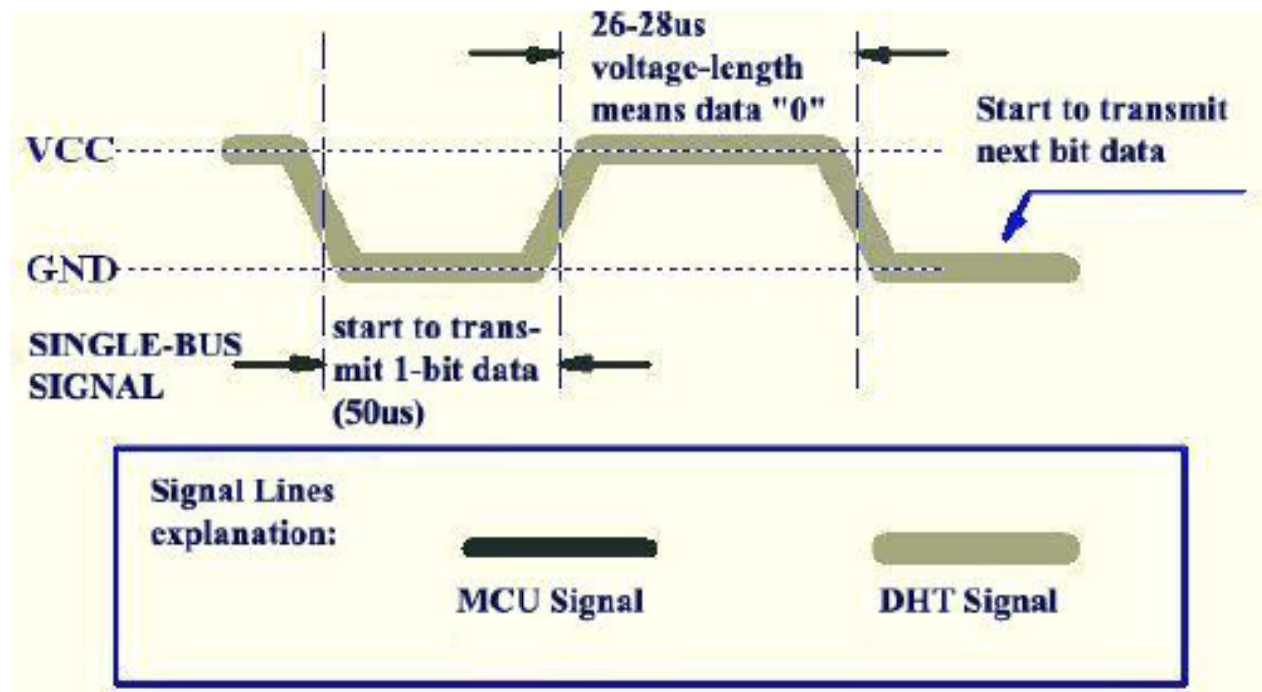


Figure 4 Data "0" Indication

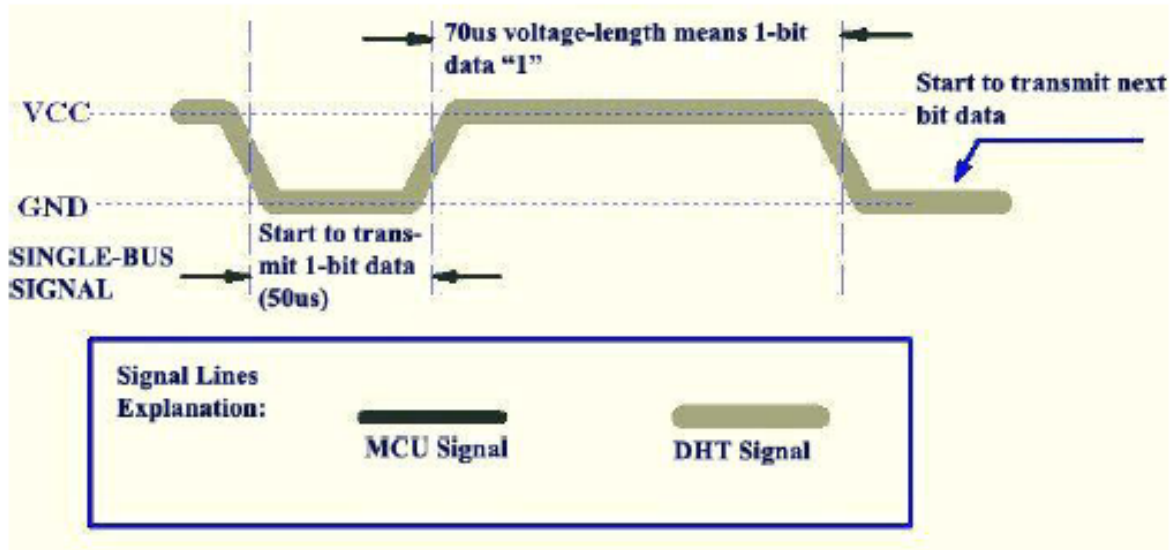


Figure 5 Data "1" Indication

If the response signal from DHT is always at high-voltage-level, it suggests that DHT is not responding properly and please check the connection. When the last bit data is transmitted, DHT11 pulls down the voltage level and keeps it for 50µs. Then the Single-Bus voltage will be pulled up by the resistor to set it back to the free status.

6. Electrical Characteristics

VDD=5V, T = 25°C (unless otherwise stated)

	Conditions	Minimum	Typical	Maximum
Power Supply	DC	3V	5V	5.5V
Current Supply	Measuring	0.5mA		2.5mA
	Average	0.2mA		1mA
	Standby	100µA		150µA
Sampling period	Second	1		

Note: Sampling period at intervals should be no less than 1 second.

7. Attentions of application

(1) Operating conditions

Applying the DHT11 sensor beyond its working range stated in this datasheet can result in 3%RH signal shift/discrepancy. The DHT11 sensor can recover to the calibrated status gradually when it gets back to the normal operating condition and works within its range. Please refer to (3) of

this section to accelerate its recovery. Please be aware that operating the DHT11 sensor in the non-normal working conditions will accelerate sensor's aging process.

(2) Attention to chemical materials

Vapor from chemical materials may interfere with DHT's sensitive-elements and debase its sensitivity. A high degree of chemical contamination can permanently damage the sensor.

(3) Restoration process when (1) & (2) happen

Step one: Keep the DHT sensor at the condition of Temperature 50~60Celsius, humidity <10%RH for 2 hours;

Step two:K keep the DHT sensor at the condition of Temperature 20~30Celsius, humidity >70%RH for 5 hours.

(4) Temperature Affect

Relative humidity largely depends on temperature. Although temperature compensation technology is used to ensure accurate measurement of RH, it is still strongly advised to keep the humidity and temperature sensors working under the same temperature. DHT11 should be mounted at the place as far as possible from parts that may generate heat.

(5) Light Affect

Long time exposure to strong sunlight and ultraviolet may debase DHT's performance.

(6) Connection wires

The quality of connection wires will affect the quality and distance of communication and high quality shielding-wire is recommended.

(7) Other attentions

* Welding temperature should be bellow 260Celsius and contact should take less than 10 seconds.

* Avoid using the sensor under dew condition.

* Do not use this product in safety or emergency stop devices or any other occasion that failure of DHT11 may cause personal injury.

* Storage: Keep the sensor at temperature 10-40°C, humidity <60%RH.

Disclaimer

This is a translated version of the manufacturer's data sheet. OSEPP is not responsible for the accuracy of the translated information.

ANEXO 8



ESP8266EX Datasheet

Version 4.3

Espressif Systems IOT Team

<http://bbs.espressif.com/>

Copyright © 2015



Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice. THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

The Wi-Fi Alliance Member Logo is a trademark of the WiFi Alliance.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.

Copyright © 2015 Espressif Systems. All rights reserved.



Table of Contents

1.	General Overview	6
1.1.	Introduction	6
1.2.	Features	7
1.3.	Parameters	7
1.4.	Ultra Low Power Technology	9
1.5.	Major Applications.....	9
2.	Hardware Overview.....	11
2.1.	Pin Definitions	11
2.2.	Electrical Characteristics.....	13
2.3.	Power Consumption.....	13
2.4.	Receiver Sensitivity.....	14
2.5.	MCU.....	15
2.6.	Memory Organization	15
2.6.1.	Internal SRAM and ROM.....	15
2.6.2.	External SPI Flash.....	15
2.7.	AHB and AHB Blocks.....	16
3.	Pins and Definitions.....	17
3.1.	GPIO	17
3.1.1.	General Purpose Input/Output Interface (GPIO)	17



- 3.2. Secure Digital Input/Output Interface (SDIO)18
- 3.3. Serial Peripheral Interface (SPI/HSPI).....18
 - 3.3.1. General SPI (Master/Slave).....18
 - 3.3.2. SDIO / SPI (Slave).....19
 - 3.3.3. HSPI (Master/Slave).....19
- 3.4. Inter-integrated Circuit Interface (I2C).....19
- 3.5. I2S20
- 3.6. Universal Asynchronous Receiver Transmitter (UART).....20
- 3.7. Pulse-Width Modulation (PWM).....21
- 3.8. IR Remote Control22
- 3.9. ADC (Analog-to-digital Converter)22
- 3.10. LED Light and Button24
- 4. Firmware & Software Development Kit26
 - 4.1. Features.....26
- 5. Power Management27
- 6. Clock Management28
 - 6.1. High Frequency Clock.....28
 - 6.2. External Reference Requirements29
- 7. Radio.....29
 - 7.1. Channel Frequencies30
 - 7.2. 2.4 GHz Receiver30
 - 7.3. 2.4 GHz Transmitter30



7.4.	Clock Generator.....	30
8.	Appendix: QFN32 Package Size	31



1. General Overview

1.1. Introduction

Espressif Systems' Smart Connectivity Platform (ESCP) is a set of high performance, high integration wireless SOCs, designed for space and power constrained mobile platform designers. It provides unsurpassed ability to embed WiFi capabilities within other systems, or to function as a standalone application, with the lowest cost, and minimal space requirement.

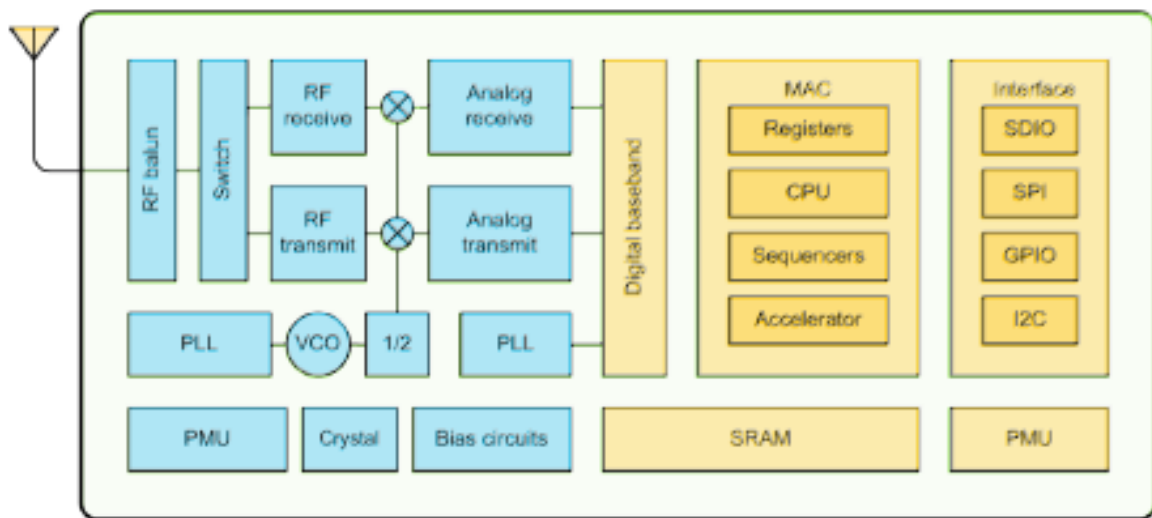


Figure 1 ESP8266EX Block Diagram

ESP8266EX offers a complete and self-contained WiFi networking solution; it can be used to host the application or to offload WiFi networking functions from another application processor.

When ESP8266EX hosts the application, it boots up directly from an external flash. It has integrated cache to improve the performance of the system in such applications.

Alternately, serving as a WiFi adapter, wireless internet access can be added to any micro controller-based design with simple connectivity (SPI/SDIO or I2C/UART interface).

ESP8266EX is among the most integrated WiFi chip in the industry; it integrates the antenna switches, RF balun, power amplifier, low noise receive amplifier, filters, power management modules, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

ESP8266EX also integrates an enhanced version of Tensilica's L106 Diamond series 32-bit processor, with on-chip SRAM, besides the WiFi functionalities. ESP8266EX is often integrated with external sensors and other application specific devices through its GPIOs; sample codes for such applications are provided in the software development kit (SDK).



Espressif Systems' Smart Connectivity Platform (ESCP) demonstrates sophisticated system-level features include fast sleep/wake context switching for energy-efficient VoIP, adaptive radio biasing for low-power operation, advance signal processing, and spur cancellation and radio co-existence features for common cellular, Bluetooth, DDR, LVDS, LCD interference mitigation.

1.2. Features

- 802.11 b/g/n
- Integrated low power 32-bit MCU
- Integrated 10-bit ADC
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Supports antenna diversity
- WiFi 2.4 GHz, support WPA/WPA2
- Support STA/AP/STA+AP operation modes
- Support Smart Link Function for both Android and iOS devices
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IR Remote Control, PWM, GPIO
- STBC, 1x1 MIMO, 2x1 MIMO
- A-MPDU & A-MSDU aggregation & 0.4s guard interval
- Deep sleep power <10uA, Power down leakage current < 5uA
- Wake up and transmit packets in < 2ms
- Standby power consumption of < 1.0mW (DTIM3)
- +20 dBm output power in 802.11b mode
- Operating temperature range -40C ~ 125C
- FCC, CE, TELEC, WiFi Alliance, and SRRC certified

1.3. Parameters

Table 1 Parameters



Categories	Items	Values
WiFi Parameters	Certificates	FCC/CE/TELEC/SRRC
	WiFi Protocols	802.11 b/g/n
	Frequency Range	2.4G-2.5G (2400M-2483.5M)
	Tx Power	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
	Rx Sensitivity	802.11 b: -91 dbm (11 Mbps)
		802.11 g: -75 dbm (54 Mbps)
802.11 n: -72 dbm (MCS7)		
Types of Antenna	PCB Trace, External, IPEX Connector, Ceramic Chip	
Hardware Parameters	Peripheral Bus	UART/SDIO/SPI/I2C/I2S/IR Remote Control
		GPIO/PWM
	Operating Voltage	3.0~3.6V
	Operating Current	Average value: 80mA
	Operating Temperature Range	-40°~125°
	Ambient Temperature Range	Normal temperature
	Package Size	5x5mm
	External Interface	N/A
Software Parameters	WiFi mode	station/softAP/SoftAP+station
	Security	WPA/WPA2
	Encryption	WEP/TKIP/AES
	Firmware Upgrade	UART Download / OTA (via network)
	Software Development	Supports Cloud Server Development / SDK for custom firmware development
	Network Protocols	IPv4, TCP/UDP/HTTP/FTP



	User Configuration	AT Instruction Set, Cloud Server, Android/ iOS App
--	--------------------	---

1.4. Ultra Low Power Technology

ESP8266EX has been designed for mobile, wearable electronics and Internet of Things applications with the aim of achieving the lowest power consumption with a combination of several proprietary techniques. The power saving architecture operates mainly in 3 modes: active mode, sleep mode and deep sleep mode.

By using advance power management techniques and logic to power-down functions not required and to control switching between sleep and active modes, ESP8266EX consumes about than 60uA in deep sleep mode (with RTC clock still running) and less than 1.0mA (DTIM=3) or less than 0.5mA (DTIM=10) to stay connected to the access point.

When in sleep mode, only the calibrated real-time clock and watchdog remains active. The real-time clock can be programmed to wake up the ESP8266EX at any required interval.

The ESP8266EX can be programmed to wake up when a specified condition is detected. This minimal wake-up time feature of the ESP8266EX can be utilized by mobile device SOCs, allowing them to remain in the low-power standby mode until WiFi is needed.

In order to satisfy the power demand of mobile and wearable electronics, ESP8266EX can be programmed to reduce the output power of the PA to fit various application profiles, by trading off range for power consumption.

1.5. Major Applications

Major fields of ESP8266EX applications to Internet-of-Things include:

- Home Appliances
- Home Automation
- Smart Plug and lights
- Mesh Network
- Industrial Wireless Control
- Baby Monitors
- IP Cameras
- Sensor Networks
- Wearable Electronics



- WiFi Location-aware Devices
- Security ID Tags
- WiFi Position System Beacons

2. Hardware Overview

2.1. Pin Definitions

The pin assignments for 32-pin QFN package is illustrated in Fig.2.

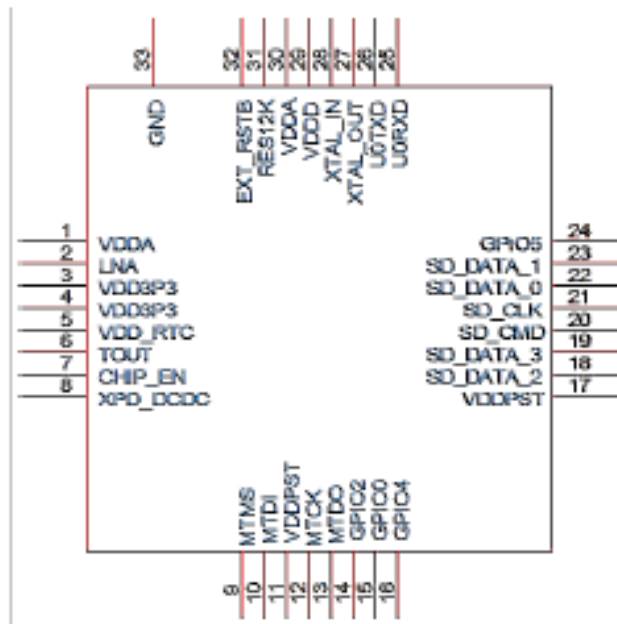


Figure 2 Pin Assignments

Table 2 below presents an overview on the general pin attributes and the functions of each pin.

Table 2 Pin Definitions

Pin	Name	Type	Function
1	VDDA	P	Analog Power 3.0~3.6V
2	LNA	I/O	RF Antenna Interface. Chip Output Impedance=50Ω No matching required but we recommend that the n-type matching network is retained.
3	VDD3P3	P	Amplifier Power 3.0~3.6V
4	VDD3P3	P	Amplifier Power 3.0~3.6V
5	VDD_RTC	P	NC(1.1V)



6	TOUT	I	ADC Pin (note: an internal pin of the chip) can be used to check the power voltage of VDD3P3 (Pin 3 and Pin4) or the input voltage of TOUT (Pin 6). These two functions cannot be used simultaneously.
7	CHIP_EN	I	Chip Enable. High: On, chip works properly; Low: Off, small current
8	XPD_DCDC	I/O	Deep-Sleep Wakeup; GPIO16
9	MTMS	I/O	GPIO14; HSPL_CLK
10	MTDI	I/O	GPIO12; HSPL_MISO
11	VDDPST	P	Digital/IO Power Supply (1.8V~3.3V)
12	MTCK	I/O	GPIO13; HSPL_MOSI; UART0_CTS
13	MTDO	I/O	GPIO15; HSPL_CS; UART0_RTS
14	GPIO2	I/O	UART Tx during flash programming; GPIO2
15	GPIO0	I/O	GPIO0; SPL_CS2
16	GPIO4	I/O	GPIO4
17	VDDPST	P	Digital/IO Power Supply (1.8V~3.3V)
18	SDIO_DATA_2	I/O	Connect to SD_D2 (Series R: 200Ω); SPIHD; HSPiHD; GPIO9
19	SDIO_DATA_3	I/O	Connect to SD_D3 (Series R: 200Ω); SPIWP; HSPiWP; GPIO10
20	SDIO_CMD	I/O	Connect to SD_CMD (Series R: 200Ω); SPL_CS0; GPIO11
21	SDIO_CLK	I/O	Connect to SD_CLK (Series R: 200Ω); SPL_CLK; GPIO6
22	SDIO_DATA_0	I/O	Connect to SD_D0 (Series R: 200Ω); SPL_MSIO; GPIO7
23	SDIO_DATA_1	I/O	Connect to SD_D1 (Series R: 200Ω); SPL_MOSI; GPIO8
24	GPIO5	I/O	GPIO5
25	U0RXD	I/O	UART Rx during flash programming; GPIO3
26	U0TXD	I/O	UART Tx during flash programming; GPIO1; SPL_CS1
27	XTAL_OUT	I/O	Connect to crystal oscillator output, can be used to provide BT clock input
28	XTAL_IN	I/O	Connect to crystal oscillator input
29	VDDD	P	Analog Power 3.0V~3.6V
30	VDDA	P	Analog Power 3.0V~3.6V
31	RES12K	I	Serial connection with a 12 kΩ resistor and connect to the ground
32	EXT_RSTB	I	External reset signal (Low voltage level: Active)



Note: GPIO2, GPIO0, MTDO can be configurable as 3-bit SDIO mode.

2.2. Electrical Characteristics

Table 3 ESP8266EX Electrical Characteristics

Parameters		Conditions	Min	Typical	Max	Unit
Storage Temperature Range			-40	Normal	125	°C
Maximum Soldering Temperature		IPC/JEDEC J-STD-020			260	°C
Working Voltage Value			3.0	3.3	3.6	V
I/O	V_{IL}/V_{IH}		-0.3/0.75 V_{IO}		0.25 $V_{IO}/3.6$	V
	V_{OL}/V_{OH}					
	I_{MAX}					12
Electrostatic Discharge (HBM)		TAMB=25°C			2	KV
Electrostatic Discharge (CDM)		TAMB=25°C			0.5	KV

2.3. Power Consumption

The following current consumption is based on 3.3V supply, and 25°C ambient, using internal regulators. Measurements are done at antenna port without SAW filter. All the transmitter's measurements are based on 90% duty cycle, continuous transmit mode.

Table 4 Description on Power Consumption

Parameters	Min	Typical	Max	Unit
Tx 802.11b, CCK 11Mbps, P OUT = +17dBm		170		mA
Tx 802.11g, OFDM 54Mbps, P OUT = +15dBm		140		mA
Tx 802.11n, MCS7, P OUT = +13dBm		120		mA
Rx 802.11b, 1024 bytes packet length, -80dBm		50		mA
Rx 802.11g, 1024 bytes packet length, -70dBm		56		mA
Rx 802.11n, 1024 bytes packet length, -65dBm		56		mA
Modem-Sleep①		15		mA
Light-Sleep②		0.9		mA
Deep-Sleep③		10		uA
Power Off		0.5		uA



①: Modem-Sleep requires the CPU to be working, as in PWM or I2S applications. According to 802.11 standards (like U-APSD), it saves power to shut down the WiFi Modem circuit while maintaining a WiFi connection with no data transmission. E.g. in DTIM3, to maintain a sleep 300ms-wake 3ms cycle to receive AP's Beacon packages, the current is about 15mA

②: During Light-Sleep, the CPU may be suspended in applications like WiFi switch. Without data transmission, the WiFi Modem circuit can be turned off and CPU suspended to save power according to the 802.11 standard (U-APSD). E.g. in DTIM3, to maintain a sleep 300ms-wake 3ms cycle to receive AP's Beacon packages, the current is about 0.9mA.

③: Deep-Sleep does not require WiFi connection to be maintained. For application with long time lags between data transmission, e.g. a temperature sensor that checks the temperature every 100s, sleep 300s and waking up to connect to the AP (taking about 0.3~1s), the overall average current is less than 1mA.

2.4. Receiver Sensitivity

The following are measured under room temperature conditions with 3.3V and 1.1V power supplies.

Table 5 Receiver Sensitivity

Parameters	Min	Typical	Max	Unit
Input frequency	2412		2484	MHz
Input impedance		50		Ω
Input reflection			-10	dB
Output power of PA for 72.2Mbps	15.5	16.5	17.5	dBm
Output power of PA for 11b mode	19.5	20.5	21.5	dBm
Sensitivity				
DSSS, 1Mbps		-98		dBm
CCK, 11Mbps		-91		dBm
6Mbps (1/2 BPSK)		-93		dBm
54Mbps (3/4 64-QAM)		-75		dBm
HT20, MCS7 (65Mbps, 72.2Mbps)		-72		dBm
Adjacent Channel Rejection				
OFDM, 6Mbps		37		dB
OFDM, 54Mbps		21		dB
HT20, MCS0		37		dB
HT20, MCS7		20		dB



2.5. MCU

ESP8266EX is embedded with Tensilica L106 32-bit micro controller (MCU), which features extra low power consumption and 16-bit RSIC. The CPU clock speed is 80MHz. It can also reach a maximum value of 160MHz. Real Time Operation System (RTOS) is enabled. Currently, only 20% of MIPS has been occupied by the WiFi stack, the rest can all be used for user application programming and development. The following interfaces can be used to connect to the MCU embedded in ESP8266EX:

- Programmable RAM/ROM interfaces (iBus), which can be connected with memory controller, and can also be used to visit external flash;
- Data RAM interface (dBus), which can connected with memory controller;
- AHB interface, can be used to visit the register.

2.6. Memory Organization

2.6.1. Internal SRAM and ROM

ESP8266EX WiFi SoC is embedded with memory controller, including SRAM and ROM. MCU can visit the memory units through iBus, dBus, and AHB interfaces. All memory units can be visited upon request, while a memory arbiter will decide the running sequence according to the time when these requests are received by the processor.

According to our current version of SDK provided, SRAM space that is available to users is assigned as below:

- **RAM size < 36kB**, that is to say, when ESP8266EX is working under the station mode and is connected to the router, programmable space accessible to user in heap and data section is around 36kB.)
- There is no programmable ROM in the SoC, therefore, user program must be stored in an external SPI flash.

2.6.2. External SPI Flash

An external SPI flash is used together with ESP8266EX to store user programs. Theoretically speaking, up to 16 Mbyte memory capacity can be supported.

Suggested SPI Flash memory capacity:

- OTA is disabled: the minimum flash memory that can be supported is 512 kByte;
- OTA is enabled: the minimum flash memory that can be supported is 1 Mbyte.

Several SPI modes can be supported, including Standard SPI, Dual SPI, DIO SPI, QIO SPI, and Quad SPI.



Therefore, please choose the correct SPI mode when you are downloading into the flash, otherwise firmwares/programs that you downloaded may not work in the right way.

2.7. AHB and APB Blocks

The AHB block performs the function of an arbiter, controls the AHB interfaces from the MAC, SDIO (host) and CPU. Depending on the address, the AHB data requests can go into one of the two slaves: APB block, or

flash controller (usually for standalone applications).

Data requests to the memory controller are usually high speed requests, and requests to the APB block are usually register access.

The APB block acts as a decoder. It is meant only for access to programmable registers within ESP8266's main blocks. Depending on the address, the APB request can go to the radio, SI/SPI, SDIO (host), GPIO, UART, real-time clock (RTC), MAC or digital baseband.

3. Pins and Definitions

The chipset encapsulates variable analog and data transmission I/Os, descriptions and definitions of which are explained below in detail.

3.1. GPIO

3.1.1. General Purpose Input/Output Interface (GPIO)

There are up to 17 GPIO pins. They can be assigned to various functions by the firmware. Each GPIO can be configured with internal pull-up (except XPD_DCDC, which is configured with internal pull-down), input available for sampling by a software register, input triggering an edge or level CPU interrupt, input triggering a level wakeup interrupt, open-drain or push-pull output driver, or output source from a software register, or a sigma-delta PWM DAC.

These pins are multiplexed with other functions such as I2C, I2S, UART, PWM, IR Remote Control, etc. Data I/O soldering pad is bidirectional and tri-state that include data input and output controlling buffer. Besides, I/O can be set as a specific state and remains like this. For example, if you intend to lower the power consumption of the chip, all data input and output enable signals can be set as remaining low power state. You can transport some specific state into the I/O. When the I/O is not powered by external circuits, the I/O will remain to be the state that it was used the last time. Some positive feedback is generated by the state-remaining function of the pins, therefore, if the external driving power must be stronger than the positive feedback. Even so, the driving power that is needed is within 5 μ A.

Table 6 Pin Definitions of GPIOs

Variables	Symbol	Min	Max	Unit
Input Low Voltage	V_{IL}	-0.3	$0.25 \times V_{IO}$	V
Input High Voltage	V_{IH}	$0.75 \times V_{IO}$	3.3	V
Input Leakage Current	I_{IL}		50	nA
Output Low Voltage	V_{OL}		$0.1 \times V_{IO}$	V
Output High Voltage	V_{OH}	$0.8 \times V_{IO}$		V
Input Pin Resistance Value	C_{pad}		2	pF
VDDIO	V_{IO}	1.8	3.3	V
Maximum Driving Power	I_{MAX}		12	mA
Temperature	T_{amb}	-40	125	$^{\circ}$ C

All digital IO pins are protected from over-voltage with a snap-back circuit connected between the pad and ground. The snap back voltage is typically about 6V, and the holding voltage is 5.8V. This



provides protection from over-voltages and ESD. The output devices are also protected from reversed voltages with diodes.

3.2. Secure Digital Input/Output Interface (SDIO)

One Slave SDIO has been defined by ESP8266EX, the definitions of which are described in Table 7 below. 4bit 25MHz SDIO v1.1 and 4bit 50MHz SDIO v2.0 are supported.

Table 7 Pin Definitions of SDIOs

Pin Name	Pin Num	IO	Function Name
SDIO_CLK	21	IO6	SDIO_CLK
SDIO_DATA0	22	IO7	SDIO_DATA0
SDIO_DATA1	23	IO8	SDIO_DATA1
SDIO_DATA_2	18	IO9	SDIO_DATA_2
SDIO_DATA_3	19	IO10	SDIO_DATA_3
SDIO_CMD	20	IO11	SDIO_CMD

3.3. Serial Peripheral Interface (SPI/HSPI)

Currently, one general Slave/Master SPI, one Slave SDIO/SPI, and one general Slave/Master HSPI have been defined by ESP8266EX. Functions of all these pins can be implemented via hardware. The pin definitions are described below:

3.3.1. General SPI (Master/Slave)

Table 8 Pin Definitions of General SPIs

Pin Name	Pin Num	IO	Function Name
SDIO_CLK	21	IO6	SPICLK
SDIO_DATA0	22	IO7	SPIQ/MISO
SDIO_DATA1	23	IO8	SPID/MOSI
SDIO_DATA_2	18	IO9	SPIHD
SDIO_DATA_3	19	IO10	SPIWP
SDIO_CMD	20	IO11	SPICS0
U0TXD	26	IO1	SPICS1
GPIO0	15	IO0	SPICS2



3.3.2. SDIO / SPI (Slave)

Table 9 Pin Definitions of SDIO / SPI (Slave)

Pin Name	Pin Num	IO	Function Name
SDIO_CLK	21	IO6	SPI_SLAVE_CLK
SDIO_DATA0	22	IO7	SPI_SLAVE_MISO
SDIO_DATA1	23	IO8	SPI_SLAVE_INT
SDIO_DATA_2	18	IO9	NC
SDIO_DATA_3	19	IO10	SPI_SLAVE_CS
SDIO_CMD	20	IO11	SPI_SLAVE_MOSI

3.3.3. HSPI (Master/Slave)

Table 10 Pin Definitions of HSPI (Master/Slave)

Pin Name	Pin Num	IO	Function Name
MTMS	9	IO14	HSPICLK
MTDI	10	IO12	HSPIQ/MISO
MTCK	12	IO13	HSPID/MOSI
MTDO	13	IO15	HPSICS

Note:

- SPI mode can be implemented via software programming. The clock frequency can reach up to a maximum value of 80MHz.
- Function of Slave SDIO/SPI interface can be implemented via hardware, and linked list DMA (Direct Memory Access) is supported, software overheads are smaller. However, there is no linked list DMA on general SPI and HSPI, and the software overheads are larger, therefore, the data transmitting speed will be restrained by software processing speed.

3.4. Inter-integrated Circuit Interface (I2C)

One I2C, which is mainly used to connect with micro controller and other peripheral equipment such as sensors, is defined by ESP8266EX. The present pin definition of I2C is as defined below:



Table 11 Pin Definitions of I2C

Pin Name	Pin Num	IO	Function Name
MTMS	9	IO14	I2C_SCL
GPIO2	14	IO2	I2C_SDA

Both I2C-Master and I2C-Slave are supported. I2C interface functionality can be realized via software programming, the clock frequency can be up to around 100KHz at most. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

3.5. I2S

Currently one I2S data input interface and one I2S data output interface are defined. I2S interface is mainly used in applications such as data collection, processing, and transmission of audio data, as well as the input and output of serial data. For example, LED lights (WS2812 series) are supported. The pin definition of I2S is as defined below:

Table 12 Pin Definitions of I2S

I2S Data Input:			
Pin Name	Pin Num	IO	Function Name
MTDI	10	IO12	I2SI_DATA
MTCK	12	IO13	I2SI_BCK
MTMS	9	IO14	I2SI_WS
I2S Data Output:			
Pin Name	Pin Num	IO	Function Name
MTDO	13	IO15	I2SO_BCK
U0RXD	25	IO3	I2SO_DATA
GPIO2	14	IO2	I2SO_WS

I2S functionality can be realized via software programming, the GPIOs that will be used are multiplexed, and linked list DMA is supported.

3.6. Universal Asynchronous Receiver Transmitter (UART)

Two UART interfaces, UART0 and UART1, have been defined by ESP8266EX, the definitions are as below:



Table 13 Pin Definitions of UART Interfaces

Pin Type	Pin Name	Pin Num	IO	Function Name
UART0	U0RXD	25	IO3	U0RXD
	U0TXD	26	IO1	U0TXD
	MTDO	13	IO15	U0RTS
	MTCK	12	IO13	U0CTS
UART1	GPIO2	14	IO2	U1TXD
	SD_D1	23	IO8	U1RXD

Data transfers to/from UART interfaces can be implemented via hardware. The data transmission speed via UART interfaces can reach 115200*40 (4.5Mbps).

UART0 can be for communication. It supports fluid control. Since UART1 features only data transmit signal (Tx), it is usually used for printing log.

Notes: By default, UART0 will output some printed information when the device is powered on and is booting up. The baud rate of the printed information is closely related to the frequency of the external crystal oscillator. If the frequency of the crystal oscillator is 40MHz, then the baud rate for printing is 115200; if the frequency of the crystal oscillator is 26MHz, then the baud rate for printing is 74880. If the printed information exerts any influence on the functionality of your device, you'd better block the printing during the power-on period by changing (U0TXD, U0RXD) to (MTDO, MTCK).

3.7. Pulse-Width Modulation (PWM)

Four PWM output interfaces have been defined by ESP8266EX. They can be extended by users themselves. The present pin definitions of the PWM interfaces are defined as below:

Table 14 Pin Definitions of PWM Interfaces

Pin Name	Pin Num	IO	Function Name
MTDI	10	IO12	PWM0
MTDO	13	IO15	PWM1
MTMS	9	IO14	PWM2
GPIO4	16	IO4	PWM3

The functionality of PWM interfaces can be implemented via software programming. For example, in the LED smart light demo, the function of PWM is realized by interruption of the timer, the minimum resolution can reach as much as 44 ns. PWM frequency range is adjustable from 1000 us to 10000 us,



i.e., between 100Hz and 1KHz. When the PWM frequency is at 1 KHz, the duty ratio will reach 1/22727, and over 14 bit resolution will be achieved at 1KHz refresh rate.

3.8. IR Remote Control

Currently, only one Infrared remote control interface is defined, the pin definition is as below:

Table 14 Pin Definition of IR Remote Control

Pin Name	Pin Num	IO	Function Name
MTMS	9	IO12	IR Tx
GPIO5	24	IO5	IR Rx

The functionality of Infrared remote control interface can be implemented via software programming. NEC coding, modulation, and demodulation are used by this interface. The frequency of modulated carrier signal is 38KHz, while the duty ratio of the square wave is 1/3. The length of data transmission, which is around 1m, is determined by two factors: one is the maximum value of rated current, the other is internal current-limiting resistance value in the infrared receiver. The larger the resistance value, the lower the current, so is the power, and vice versa. The transmission angle is between 15° and 30°, and is mainly determined by the radiation direction of the infrared receiver.

Notes: Among the eight interfaces mentioned above, most of them can be multiplexed. Pin definitions that can be defined is not limited to the eight ones herein mentioned, customers can self customise the functions of the pins according to their specific application scenarios. Functions of these pins can be implemented via software programming and hardware.

3.9. ADC (Analog-to-digital Converter)

ESP8266EX is embedded with a 10-bit precision SARADC. Currently, TOUT (Pin6) is defined as ADC interface, the definition of which is described below:

Pin Name	Pin Num	Function Name
TOUT	6	ADC Interface

Table 16 Pin Definition of ADC

The following two applications can be implemented using ADC (Pin6). However, these two applications cannot be implemented concurrently.

- Test the power supply voltage of VDD3P3 (Pin 3 and Pin 4).

The function used to test the power supply voltage on PA_VDD pin is: `uint16 system_get_vdd33(void)`

- Test the input voltage of TOUT (Pin 6):



The function used to test the input voltage of TOUT is: `uint16 system_adc_read(void)`

`RF-init` parameter in the following passage refers to `esp_init_data_default.bin`

Application One:	Test the power supply voltage of VDD3P3 (Pin 3 and Pin 4).
Hardware Design:	TOUT must be dangled.
RF-init Parameter:	The 107th byte of <code>esp_init_data_default.bin</code> (0 - 127 byte), "vdd33_const", must set to be 0xFF, i.e., the value of "vdd33_const" is 255.
RF Calibration Process:	Optimize the RF circuit conditions based on the testing results of VDD3P3 (Pin 3 and Pin 4).
User Programming:	Use <code>system_get_vdd33</code> instead of <code>system_adc_read</code> .
Application Two:	Test the input voltage of TOUT (Pin 6).
Hardware Design:	The input voltage range is 0 to 1.0 V when TOUT is connected to external circuit.
RF-init Parameter:	The value of the 107th byte of <code>esp_init_data_default.bin</code> (0 - 127 byte), "vdd33_const", must be set to be the real power supply voltage of Pin 3 and Pin 4. The working power voltage range of ESP8266EX is between 1.8V and 3.6V, while the unit of "vdd33_const" is 0.1V, therefore, the effective value range of "vdd33_const" is 18 to 36.
RF Calibration Process:	Optimize the RF circuit conditions based on the value of "vdd33_const". The permissible error is $\pm 0.2V$.
User Programming:	Use <code>system_adc_read</code> instead of <code>system_get_vdd33</code> .

Note One:

In `RF_init` parameter `esp_init_data_default.bin` (0 - 127 byte), the 107th byte is defined as "vdd33_const". Definitions of "vdd33_const" is described below:

(1) If `vdd33_const = 0xff`, the power voltage of Pin 3 and Pin 4 will be tested by the internal self-calibration process of ESP8266EX chipset itself. RF circuit conditions should be optimized according to the testing results.



(2) If $18 \leq vdd33_const \leq 36$, ESP8266EX RF Calibration and optimization process is implemented via $(vdd33_const/10)$.

(3) If $vdd33_const < 18$ or $36 < vdd33_const < 255$, ESP8266EX RF Calibration and optimization process is implemented via the default value 3.0V.

Note Two:

Function `system_get_vdd33` is used to test the power supply voltage of VDD3P3 (Pin 3 and Pin 4). Details on this function are described below:

(1) Pin Tout must be dangled. The 107th byte of `esp_init_data_default.bin` (0 - 127 byte), "vdd33_const", must set to be 0xFF.

(2) If the 107th byte of `esp_init_data_default.bin` (0 - 127 byte), "vdd33_const", is equal to 0xff, the returned value of function `system_get_vdd33` will be an effective value, otherwise 0xffff will be returned.

(3) The unit of the returned value is: 1/1024 V.

Note Three:

Function `system_adc_read` is defined to test the input voltage of Pin TOUT (Pin 6). Details on this function are described below:

(1) The value of the 107th byte of `esp_init_data_default.bin` (0 - 127 byte), "vdd33_const", must be set to be the real power supply voltage of Pin 3 and Pin 4.

(2) If the 107th byte of `esp_init_data_default.bin` (0 - 127 byte), "vdd33_const", is NOT equal to 0xff, the returned value of `system_adc_read` will be an effective value of the input voltage of Pin TOUT, otherwise 0xffff will be returned.

(3) The unit of the returned value is: 1/1024 V.

3.10. LED Light and Button

ESP8266EX features up to 17 GPIOs, all of which can be assigned to realise various functions of LED lights and buttons. Definitions of some GPIOs that are assigned with certain functions in our demo application design are shown below:

Table 17 Pin Definitions of LED and Button

Pin Name	Pin Num	IO	Function Name
MTCK	12	IO13	Button (Reset)
GPIO0	15	IO0	WiFi Light
MTDI	10	IO12	Link Light



Altogether three interfaces have been defined, one is for the button, and the other two is for LED light. Generally, **MTCK** is used to control the reset button, **GPIO0** is used as an signal to indicate the WiFi working state, **MTDI** is used as a signal light to indicate communication between the device and the server.

Note: Among the nine interfaces mentioned above, most of them can be multiplexed. Pin definitions that can be defined is not limited to the eight ones herein mentioned, customers can self customise the functions of the pins according to their specific application scenarios. Functions of these pins can be implemented via software programming and hardware.



4. Firmware & Software Development Kit

The application and firmware is executed in on-chip ROM and SRAM, which loads the instructions during wake-up, through the SDIO interface, from the external flash.

The firmware implements TCP/IP, the full 802.11 b/g/n/e/i WLAN MAC protocol and WiFi Direct specification. It supports not only basic service set (BSS) operations under the distributed control function (DCF) but also P2P group operation compliant with the latest WiFi P2P protocol. Low level protocol functions are handled automatically by ESP8266:

- RTS/CTS
- acknowledgement
- fragmentation and defragmentation
- aggregation
- frame encapsulation (802.11h/RFC 1042)
- automatic beacon monitoring / scanning, and
- P2P WiFi direct

Passive or active scanning, as well as P2P discovery procedure is performed autonomously once initiated by the appropriate command. Power management is handled with minimum host interaction to minimize active duty period.

4.1. Features

The SDK includes the following library functions:

- 802.11 b/g/n/d/e/i/k/r support;
- WiFi Direct (P2P) support:
- P2P Discovery, P2P Group Owner mode, P2P Power Management
- Infrastructure BSS Station mode / P2P mode / softAP mode support;
- Hardware accelerators for CCMP (CBC-MAC, counter mode), TKIP (MIC, RC4), WAPI (SMS4), WEP (RC4), CRC;
- WPA/WPA2 PSK, and WPS driver;
- Additional 802.11i security features such as pre-authentication, and TSN;
- Open Interface for various upper layer authentication schemes over EAP such as TLS, PEAP, LEAP, SIM, AKA, or customer specific;
- 802.11n support (2.4GHz);
- Supports MIMO 1x1 and 2x1, STBC, A-MPDU and A-MSDU aggregation and 0.4 μ s guard interval;



- WMM power save U-APSD;
- Multiple queue management to fully utilize traffic prioritization defined by 802.11e standard;
- UMA compliant and certified;
- 802.11n/RFC1042 frame encapsulation;
- Scattered DMA for optimal CPU off load on Zero Copy data transfer operations;
- Antenna diversity and selection (software managed hardware);
- Clock/power gating combined with 802.11-compliant power management dynamically adapted to current connection condition providing minimal power consumption;
- Adaptive rate fallback algorithm sets the optimum transmission rate and Tx power based on actual SNR and packet loss information;
- Automatic retransmission and response on MAC to avoid packet discarding on slow host environment;
- Seamless roaming support;
- Configurable packet traffic arbitration (PTA) with dedicated slave processor based design provides flexible and exact timing Bluetooth co-existence support for a wide range of Bluetooth Chip vendors;
- Dual and single antenna Bluetooth co-existence support with optional simultaneous receive (WiFi/Bluetooth) capability.

5. Power Management

The chip can be put into the following states:

- **OFF:** CHIP_PD pin is low. The RTC is disabled. All registers are cleared.
- **DEEP_SLEEP:** Only RTC is powered on - the rest of the chip is powered off. Recovery memory of RTC can keep basic WiFi connecting information.
- **SLEEP:** Only the RTC is operating. The crystal oscillator is disabled. Any wakeup events (MAC, host, RTC timer, external interrupts) will put the chip into the WAKEUP state.
- **WAKEUP:** In this state, the system goes from the sleep states to the PWR state. The crystal oscillator and PLLs are enabled.
- **ON:** the high speed clock is operational and sent to each block enabled by the clock control register. Lower level clock gating is implemented at the block level, including the CPU, which can be gated off using the WAITI instruction, while the system is on.

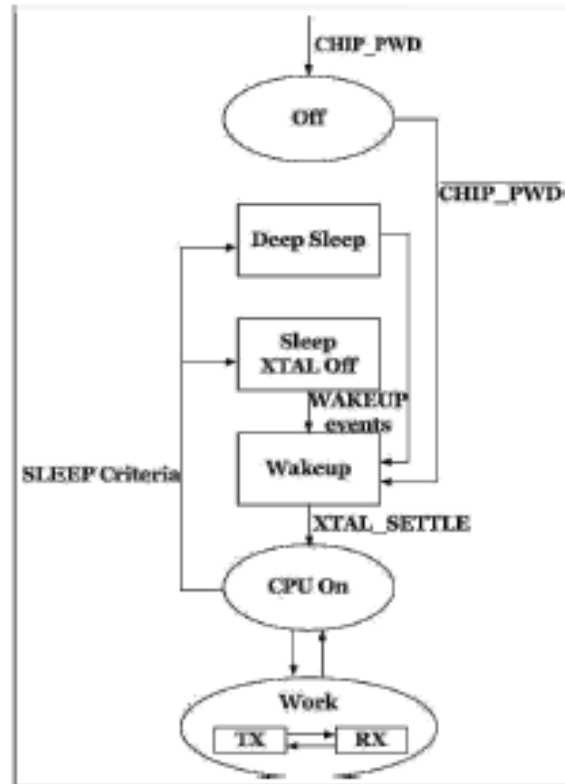


Figure 3 Illustration of Power Management

6. Clock Management

6.1. High Frequency Clock

The high frequency clock on ESP8266EX is used to drive both transmit and receive mixers. This clock is generated from the internal crystal oscillator and an external crystal. The crystal frequency can range from 26MHz to 52MHz.

While internal calibration of the crystal oscillator ensures that a wide range of crystals can be used, in general, the quality of the crystal is still a factor to consider, to have reasonable phase noise that is required for good performance. When the crystal selected is sub-optimal due to large frequency drifts or poor Q-factor, the maximum throughput and sensitivity of the WiFi system is degraded. Please refer to the application notes on how the frequency offset can be measured.



Table 18 High Frequency Clock

Parameter	Symbol	Min	Max	Unit
Frequency	FXO	26	52	MHz
Loading capacitance	CL		32	pF
Motional capacitance	CM	2	5	pF
Series resistance	RS	0	65	Ω
Frequency tolerance	Δ FXO	-15	15	ppm
Frequency vs temperature (-25°C ~ 75°C)	Δ FXO,Temp	-15	15	ppm

6.2. External Reference Requirements

For an externally generated clock, the frequency can range from 26MHz to 52MHz can be used. For good performance of the radio, the following characteristics are expected of the clock:

Table 19 External Clock Reference

Parameter	Symbol	Min	Max	Unit
Clock amplitude	VXO	0.2	1	Vpp
External clock accuracy	Δ FXO,EXT	-15	15	ppm
Phase noise @1kHz offset, 40MHz clock			-120	dBc/Hz
Phase noise @10kHz offset, 40MHz clock			-130	dBc/Hz
Phase noise @100kHz offset, 40MHz clock			-138	dBc/Hz

7. Radio

The ESP8266EX radio consists of the following main blocks:

- 2.4GHz receiver
- 2.4GHz transmitter
- High speed clock generators and crystal oscillator
- Real time clock
- Bias and regulators
- Power management



7.1. Channel Frequencies

The RF transceiver supports the following channels according to the IEEE802.11b/g/n standards.

Table 20 Frequency Channel

Channel No	Frequency (MHz)	Channel No	Frequency (MHz)
1	2412	8	2447
2	2417	9	2452
3	2422	10	2457
4	2427	11	2462
5	2432	12	2467
6	2437	13	2472
7	2442	14	2484

7.2. 2.4 GHz Receiver

The 2.4GHz receiver downconverts the RF signal to quadrature baseband signals and converts them to the digital domain with 2 high resolution high speed ADCs. To adapt to varying signal channel conditions, RF filters, automatic gain control (AGC), DC offset cancelation circuits and baseband filters are integrated within ESP8266EX.

7.3. 2.4 GHz Transmitter

The 2.4GHz transmitter up-converts the quadrature baseband signals to 2.4GHz, and drives the antenna with a high powered CMOS power amplifier. The use of digital calibration further improves the linearity of the power amplifier, enabling a state of art performance of delivering +19.5dBm average power for 802.11b transmission and +16dBm for 802.11n transmission.

Additional calibrations are integrated to cancel any imperfections of the radio, such as:

- carrier leakage,
- I/Q phase matching, and
- baseband nonlinearities

This reduces the amount of time required and test equipment required for production testing.

7.4. Clock Generator

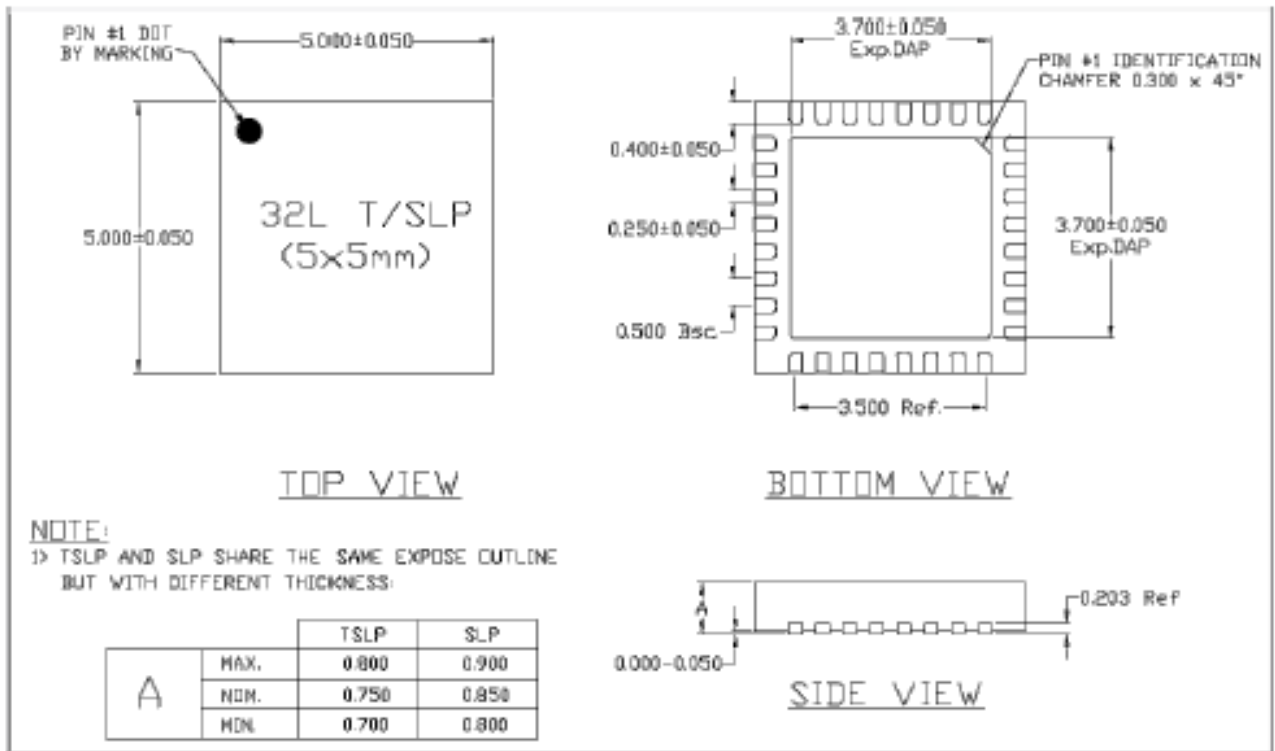
The clock generator generates quadrature 2.4 GHz clock signals for the receiver and transmitter. All components of the clock generator are integrated on-chip, including:



- inductor,
- varactor, and
- loop filter

The clock generator has built-in calibration and self test circuits. Quadrature clock phases and phase noise are optimized on-chip with patented calibration algorithms to ensure the best receiver and transmitter performance.

8. Appendix: QFN32 Package Size



ANEXO 9

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Schematic & Reference Design

EAGLE files: [arduino-mega2560-reference-design.zip](#)

Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

✦ **VIN**. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

✦ **5V**. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

✦ **3V3**. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

✦ **GND**. Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

✦ **Serial**: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

✦ **External Interrupts**: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.

✦ **PWM**: 0 to 13. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

✦ **SPI**: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.

✦ **LED**: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

✦ **I²C**: 20 (SDA) and 21 (SCL). Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- ✦ **AREF**. Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- ✦ **Reset**. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available [in the Arduino repository](#). The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can

have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

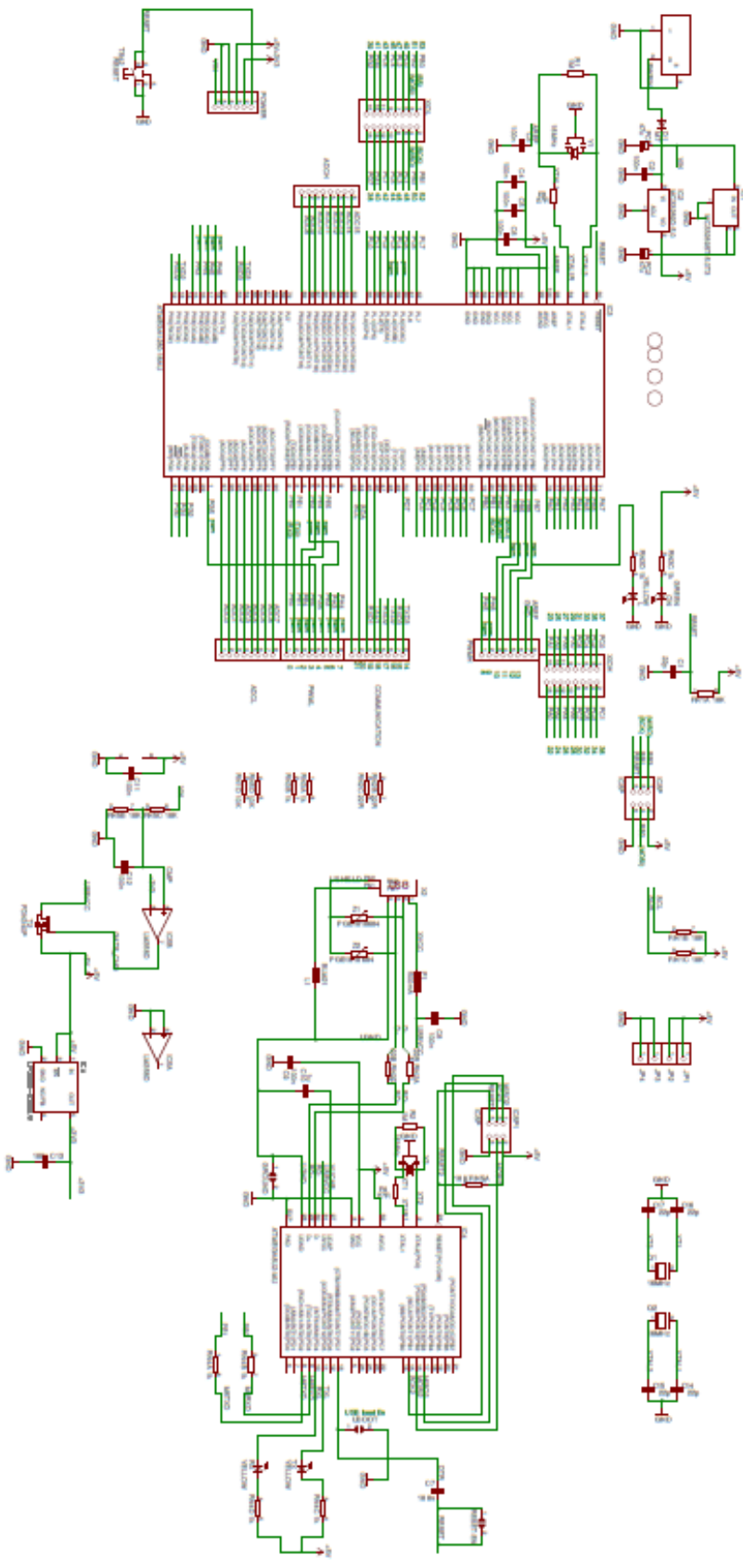
Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Diecimila. *Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).*

Arduino™ Mega 2560 Reference Design

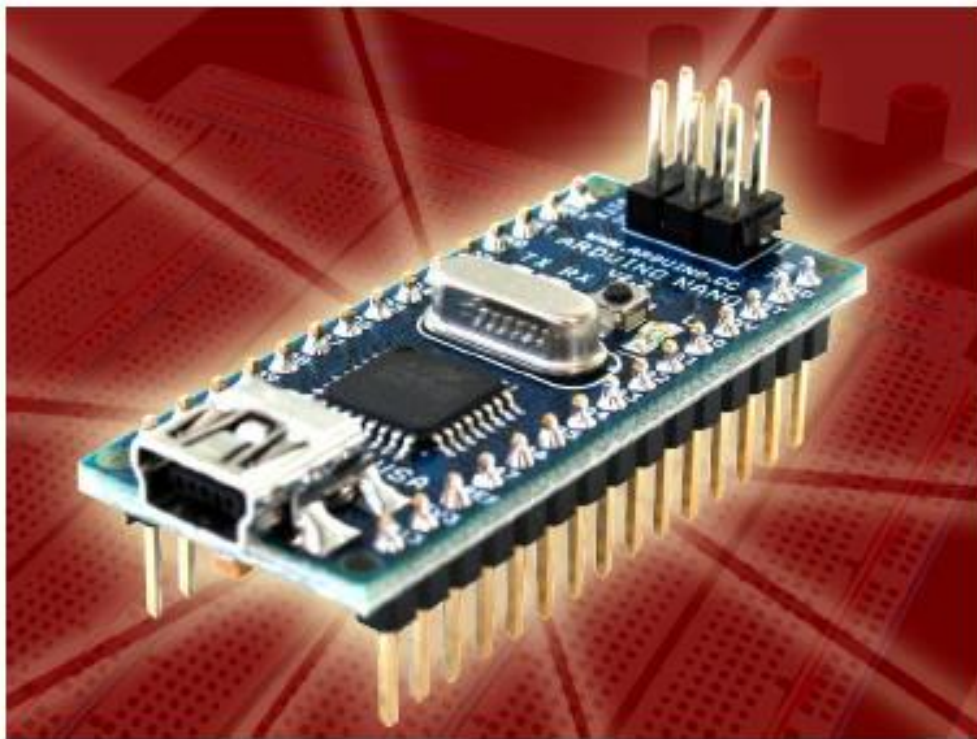
Arduino™ Mega 2560 Reference Design is a reference design for the Arduino Mega 2560. It is a PCB design that provides a complete reference design for the Arduino Mega 2560. The design is based on the Arduino Mega 2560 and includes all the components and connections needed to build a complete reference design. The design is available for free download and is a valuable resource for anyone interested in building an Arduino Mega 2560.



ANEXO 10

Arduino Nano (V2.3)

User Manual



Released under the Creative Commons Attribution Share-Alike 2.5 License

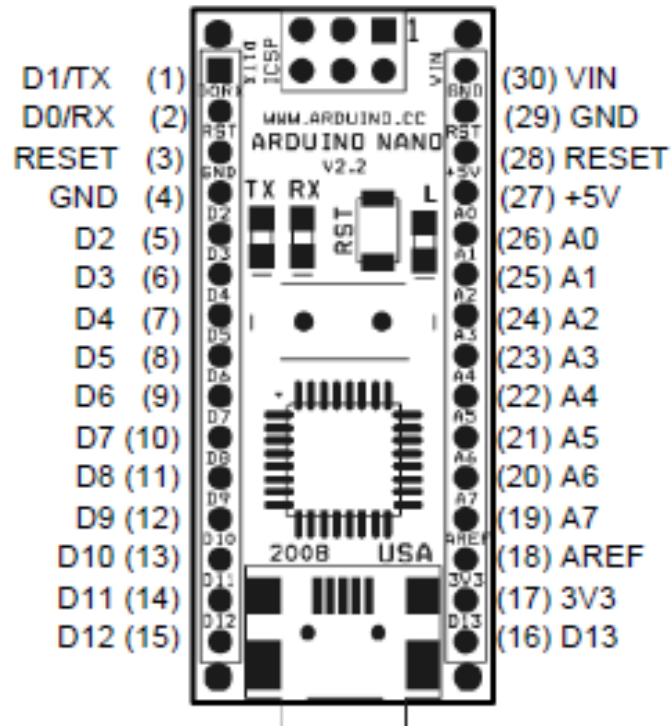
<http://creativecommons.org/licenses/by-sa/2.5/>

More information:

www.arduino.cc

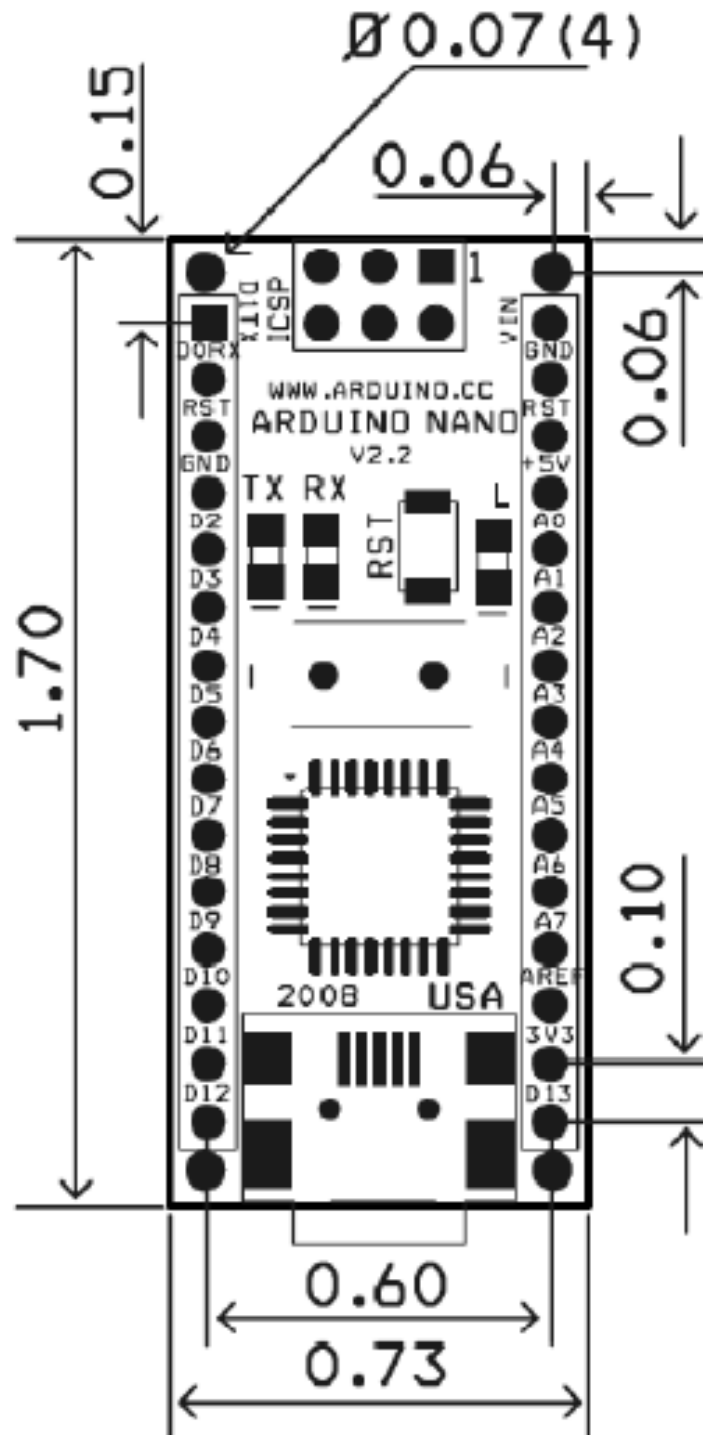
Rev. 2.3

Arduino Nano Pin Layout



Pin No.	Name	Type	Description
1-2, 5-16	D0-D13	I/O	Digital input/output port 0 to 13
3, 28	RESET	Input	Reset (active low)
4, 29	GND	PWR	Supply ground
17	3V3	Output	+3.3V output (from FTDI)
18	AREF	Input	ADC reference
19-26	A7-A0	Input	Analog input channel 0 to 7
27	+5V	Output or Input	+5V output (from on-board regulator) or +5V (input from external power supply)
30	VIN	PWR	Supply voltage

Arduino Nano Mechanical Drawing



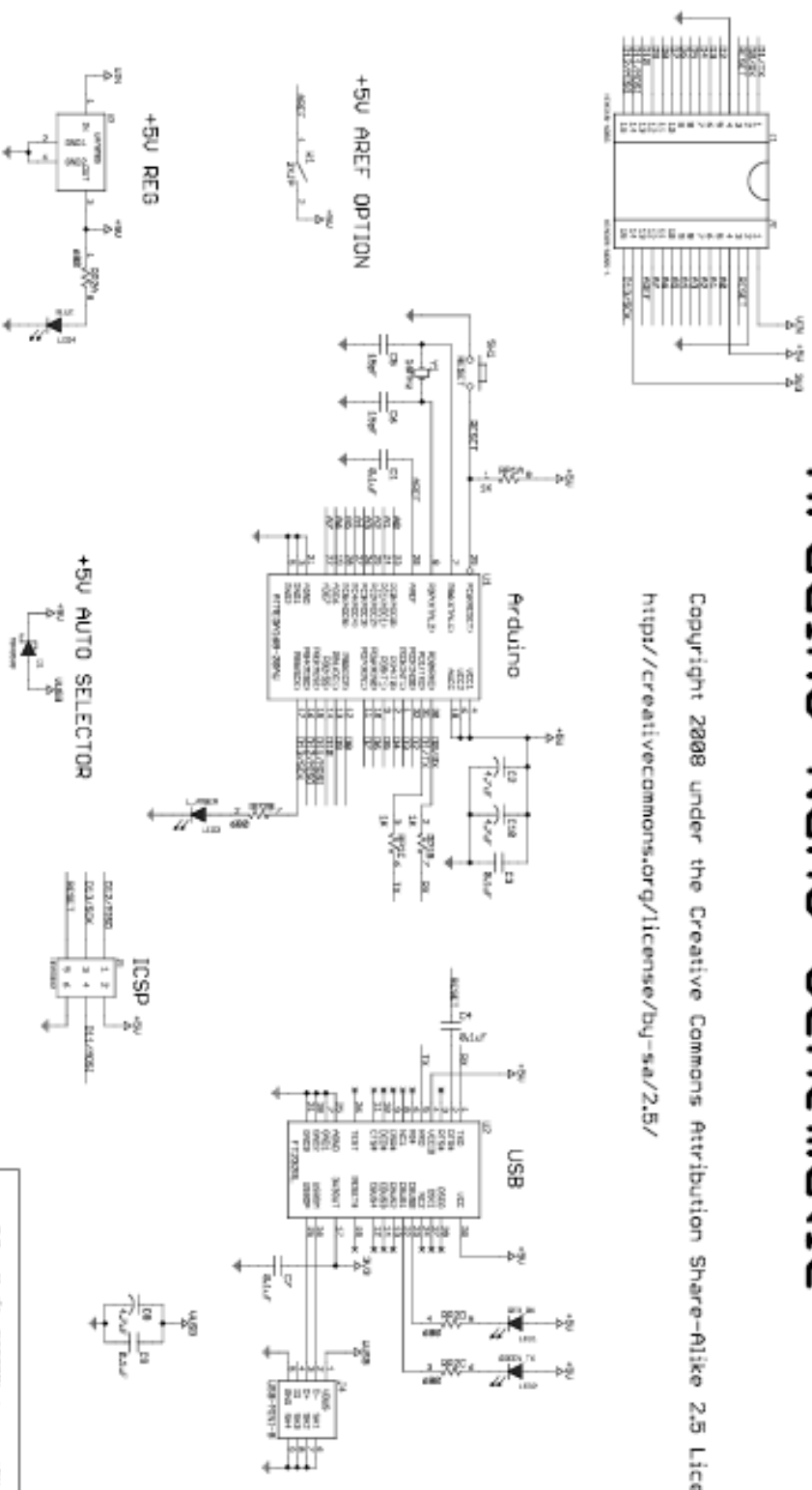
ALL DIMENTIONS ARE IN INCHES

Arduino Nano Bill of Material

Item Number	Qty.	Ref. Dest.	Description	Mfg. P/N	MFG	Vendor P/N	Vendor
1	5	C1,C3,C4,C7,C9	Capacitor, 0.1uF 50V 10% Ceramic X7R 0805	C0805C104K5RACTU	Kemet	80-C0805C104K5R	Mouser
2	3	C2,C8,C10	Capacitor, 4.7uF 10V 10% Tantalum Case A	T491A475K010AT	Kemet	80-T491A475K010	Mouser
3	2	C5,C6	Capacitor, 18pF 50V 5% Ceramic NOP/COG 0805	C0805C180J5GACTU	Kemet	80-C0805C180J5G	Mouser
4	1	D1	Diode, Schottky 0.5A 20V	MBR0520LT1G	ONsemi	863-MBR0520LT1G	Mouser
5	1	J1,J2	Headers, 36PS 1 Row	68000-136HLF	FCI	649-68000-136HLF	Mouser
6	1	J4	Connector, Mini-B Recept Rt. Angle	67503-1020	Molex	538-67503-1020	Mouser
7	1	J5	Headers, 72PS 2 Rows	67996-272HLF	FCI	649-67996-272HLF	Mouser
8	1	LD1	LED, Super Bright RED 100mcd 640nm 120degree 0805	APT2012SRCPRV	Kingbright	604-APT2012SRCPRV	Mouser
9	1	LD2	LED, Super Bright GREEN 50mcd 570nm 110degree 0805	APHCM2012CGCK-F01	Kingbright	604-APHCM2012CGCK	Mouser
10	1	LD3	LED, Super Bright ORANGE 160mcd 601nm 110degree 0805	APHCM2012SECK-F01	Kingbright	04-APHCM2012SECK	Mouser
11	1	LD4	LED, Super Bright BLUE 80mcd 470nm 110degree 0805	LTST-C170TBKT	Lite-On Inc	160-1579-1-ND	Digikey
12	1	R1	Resistor Pack, 1K +/-5% 62.5mW 4RES SMD	YC164-JR-071KL	Yageo	YC164J-1.0KCT-ND	Digikey
13	1	R2	Resistor Pack, 680 +/-5% 62.5mW 4RES SMD	YC164-JR-07680RL	Yageo	YC164J-680CT-ND	Digikey
14	1	SW1	Switch, Momentary Tact SPST 150gf 3.0x2.5mm	B3U-1000P	Omron	SW102OCT-ND	Digikey
15	1	U1	IC, Microcontroller RISC 16kB Flash, 0.5kB EEPROM, 23 I/O Pins	ATmega168-20AU	Atmel	556-ATMEGA168-20AU	Mouser
16	1	U2	IC, USB to SERIAL UART 28 Pins SSOP	FT232RL	FTDI	895-FT232RL	Mouser
17	1	U3	IC, Voltage regulator 5V, 500mA SOT-223	UA78M05CDCYRG3	TI	595-UA78M05CDCYRG3	Mouser
18	1	Y1	Crystal, 16MHz +/-20ppm HC-49/US Low Profile	ABL-16.000MHZ-B2	Abracon	815-ABL-16-B2	Mouser

Arduino Nano Schematic

Copyright 2008 under the Creative Commons Attribution Share-Alike 2.5 License
<http://creativecommons.org/licenses/by-sa/2.5/>



NOT USED
 5V
 5V

v2.3 - Modify FT232RL to use +5V	
TITLE: Arduino Name	REV: 2.3
DOCUMENT NUMBER:	Sheet 1/1
Date: 6/26/2008 10:26:41 PM	

ANEXO 11

Single chip 2.4 GHz Transceiver

nRF24L01

FEATURES

- True single chip GFSK transceiver
- Complete OSI Link Layer in hardware
- Enhanced ShockBurst™
- Auto ACK & retransmit
- Address and CRC computation
- On the air data rate 1 or 2Mbps
- Digital interface (SPI) speed 0-8 Mbps
- 125 RF channel operation
- Short switching time enable frequency hopping
- Fully RF compatible with nRF24XX
- 5V tolerant signal input pads
- 20-pin package (QFN20 4x4mm)
- Uses ultra low cost +/- 60 ppm crystal
- Uses low cost chip inductors and 2-layer PCB
- Power supply range: 1.9 to 3.6 V

APPLICATIONS

- Wireless mouse, keyboard, joystic
- Keyless entry
- Wireless data communication
- Alarm and security systems
- Home automation
- Surveillance
- Automotive
- Telemetry
- Intelligent sports equipment
- Industrial sensors
- Toys

GENERAL DESCRIPTION

nRF24L01 is a single chip radio transceiver for the world wide 2.4 - 2.5 GHz ISM band. The transceiver consists of a fully integrated frequency synthesizer, a power amplifier, a crystal oscillator, a demodulator, modulator and Enhanced ShockBurst™ protocol engine. Output power, frequency channels, and protocol setup are easily programmable through a SPI interface. Current consumption is very low, only 9.0mA at an output power of -6dBm and 12.3mA in RX mode. Built-in Power Down and Standby modes makes power saving easily realizable.

QUICK REFERENCE DATA

Parameter	Value	Unit
Minimum supply voltage	1.9	V
Maximum output power	0	dBm
Maximum data rate	2000	kbps
Supply current in TX mode @ 0dBm output power	11.3	mA
Supply current in RX mode @ 2000 kbps	12.3	mA
Temperature range	-40 to +85	°C
Sensitivity @ 1000 kbps	-85	dBm
Supply current in Power Down mode	900	nA

Table 1 nRF24L01 quick reference data

PRELIMINARY PRODUCT SPECIFICATION



nRF24L01 Single Chip 2.4 GHz Radio Transceiver

Type Number	Description	Version
nRF24L01	20 pin QFN 4x4, RoHS & SS-00259 compliant	D
nRF24L01 IC	Bare Dice	D
nRF24L01-EVKIT	Evaluation kit (2 test PCB, 2 configuration PCB, SW)	1.0

Table 2 nRF24L01 ordering information

BLOCK DIAGRAM

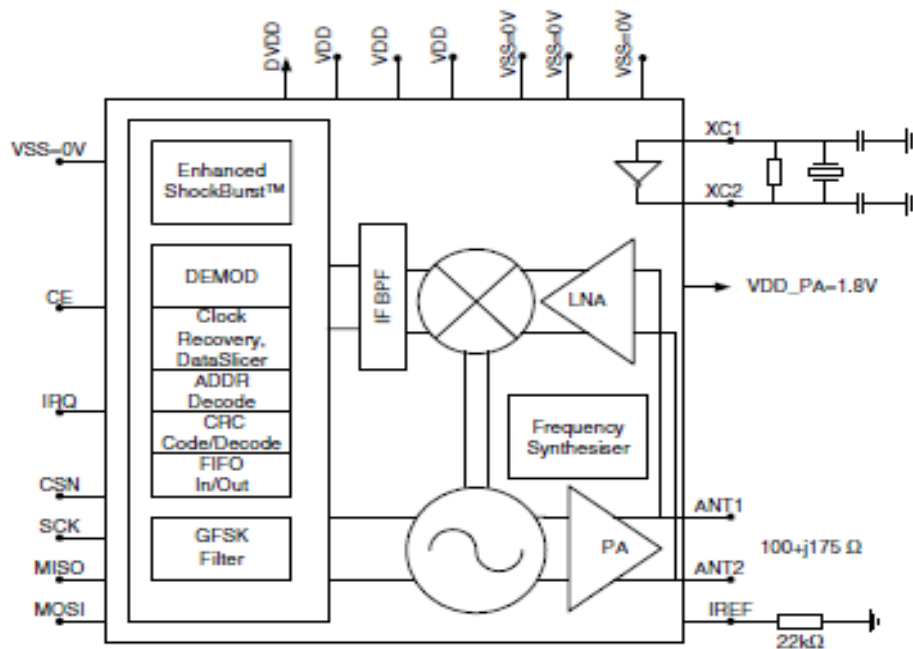


Figure 1 nRF24L01 with external components.



PIN FUNCTIONS

Pin	Name	Pin function	Description
1	CE	Digital Input	Chip Enable Activates RX or TX mode
2	CSN	Digital Input	SPI Chip Select
3	SCK	Digital Input	SPI Clock
4	MOSI	Digital Input	SPI Slave Data Input
5	MISO	Digital Output	SPI Slave Data Output, with tri-state option
6	IRQ	Digital Output	Maskable interrupt pin
7	VDD	Power	Power Supply (+3V DC)
8	VSS	Power	Ground (0V)
9	XC2	Analog Output	Crystal Pin 2
10	XC1	Analog Input	Crystal Pin 1
11	VDD_PA	Power Output	Power Supply (+1.8V) to Power Amplifier
12	ANT1	RF	Antenna interface 1
13	ANT2	RF	Antenna interface 2
14	VSS	Power	Ground (0V)
15	VDD	Power	Power Supply (+3V DC)
16	IREF	Analog Input	Reference current
17	VSS	Power	Ground (0V)
18	VDD	Power	Power Supply (+3V DC)
19	DVDD	Power Output	Positive Digital Supply output for de-coupling purposes
20	VSS	Power	Ground (0V)

Table 3 nRF24L01 pin function

PIN ASSIGNMENT

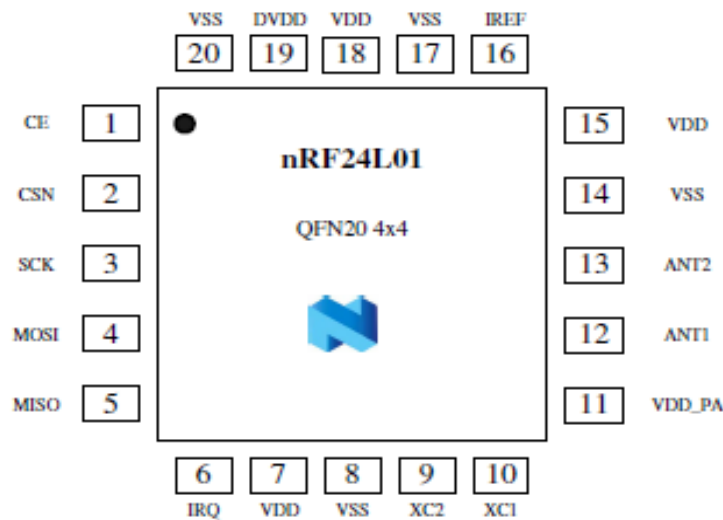


Figure 2 nRF24L01 pin assignment (top view) for a QFN20 4x4 package.

PRELIMINARY PRODUCT SPECIFICATION



nRF24L01 Single Chip 2.4 GHz Radio Transceiver

ELECTRICAL SPECIFICATIONS

Conditions: VDD = +3V, VSS = 0V, T_A = - 40°C to + 85°C

Symbol	Parameter (condition)	Notes	Min.	Typ.	Max.	Units
Operating conditions						
VDD	Supply voltage		1.9	3.0	3.6	V
TEMP	Operating Temperature		-40	+27	+85	°C
Digital input pin						
V _{HI}	HIGH level input voltage	¹	0.7VDD		5.25	V
V _{IL}	LOW level input voltage		VSS		0.3VDD	V
Digital output pin						
V _{OHI}	HIGH level output voltage (I _{OH} =-0.25mA)		VDD- 0.3		VDD	V
V _{OOL}	LOW level output voltage (I _{OL} =0.25mA)		VSS		0.3	V
General RF conditions						
f _{CP}	Operating frequency	²	2400		2525	MHz
f _{XTAL}	Crystal frequency			16		MHz
Δf _{1M}	Frequency deviation @ 1000kbps			±160		kHz
Δf _{2M}	Frequency deviation @ 2000kbps			±320		kHz
R _{GBURST}	Data rate ShockBurst™		>0		2000	kbps
F _{CHANNEL1}	Channel spacing @ 1000kbps			1		MHz
F _{CHANNEL2}	Channel spacing @ 2000kbps			2		MHz
Transmitter operation						
P _{RF}	Maximum Output Power	³		0	+4	dBm
P _{RF CR}	RF Power Control Range		16	18	20	dB
P _{RF CR}	RF Power Accuracy				±4	dB
P _{BW}	20dB Bandwidth for Modulated Carrier (2000kbps)			1800	2000	kHz
P _{RF1}	1 st Adjacent Channel Transmit Power 2MHz				-20	dBm
P _{RF2}	2 nd Adjacent Channel Transmit Power 4MHz				-50	dBm
I _{VDD}	Supply current @ 0dBm output power	⁴		11.3		mA
I _{VDD}	Supply current @ -18dBm output power			7.0		mA
I _{VDD}	Average Supply current @ -6dBm output power, Enhanced ShockBurst™	⁵		0.05		mA
I _{VDD}	Supply current in Standby-I mode	⁶		32		μA
I _{VDD}	Supply current in power down			900		nA

¹ All digital inputs handle up to 5.25V signal inputs. Keep in mind that the VDD of the nRF24L01 must match the V_{HI} of the driving device for output pins.

² Usable band is determined by local regulations

³ Antenna load impedance = 15Ω+j88Ω

⁴ Antenna load impedance = 15Ω+j88Ω. Effective data rate 1000kbps or 2000 kbps

⁵ Antenna load impedance = 15Ω+j88Ω. Effective data rate 10kbps and full packets

⁶ Given for a 12pF crystal. Current when using external clock is dependent on signal swing.

PRELIMINARY PRODUCT SPECIFICATION



nRF24L01 Single Chip 2.4 GHz Radio Transceiver

Receiver operation						
I_{VDD}	Supply current one channel 2000kbps			12.3		mA
I_{VDD}	Supply current one channel 1000kbps			11.8		mA
RX_{SENS}	Sensitivity at 0.1%BER (@2000kbps)			-82		dBm
RX_{SENS}	Sensitivity at 0.1%BER (@1000kbps)			-85		dBm
C/I_{CD}	C/I Co-channel (@2000kbps)	⁷		7 ⁸ /11 ⁹		dB
C/I_{1ST}	1 st Adjacent Channel Selectivity C/I 2MHz			1/4		dB
C/I_{2ND}	2 nd Adjacent Channel Selectivity C/I 4MHz			-21/-20		dB
C/I_{3RD}	3 rd Adjacent Channel Selectivity C/I 6MHz			-27/-27		dB
C/I_{CD}	C/I Co-channel (@1000kbps)	¹⁰		9 ¹¹ /12 ¹²		dB
C/I_{1ST}	1 st Adjacent Channel Selectivity C/I 1MHz			8/8		dB
C/I_{2ND}	2 nd Adjacent Channel Selectivity C/I 2MHz			-22/-21		dB
C/I_{3RD}	3 rd Adjacent Channel Selectivity C/I 3MHz			-30/-30		dB

Table 4 nRF24L01 RF specifications

⁷ Data rate is 2000kbps for the following C/I measurements

⁸ According to ETSI EN 300 440-1 V1.3.1 (2001-09) page 27

⁹ nRF24L01 equal modulation on interfering signal

¹⁰ Data rate is 1000kbps for the following C/I measurements

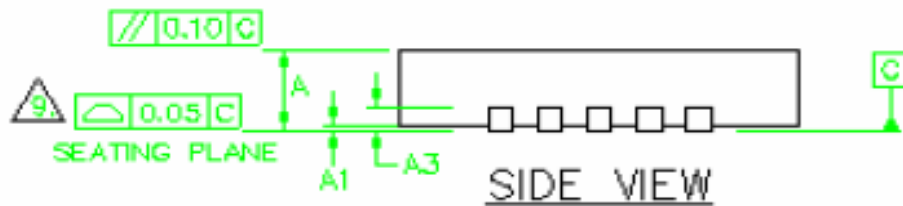
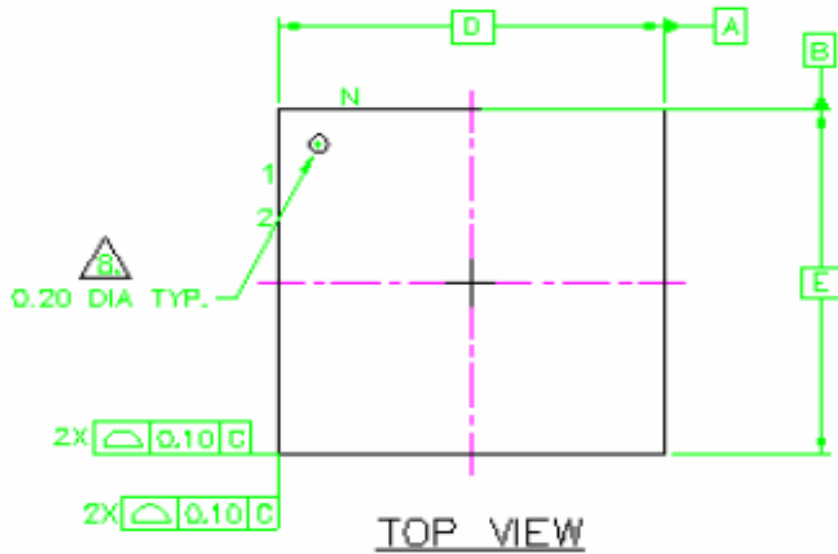
¹¹ According to ETSI EN 300 440-1 V1.3.1 (2001-09) page 27

¹² nRF24L01 equal modulation on interfering signal



PACKAGE OUTLINE

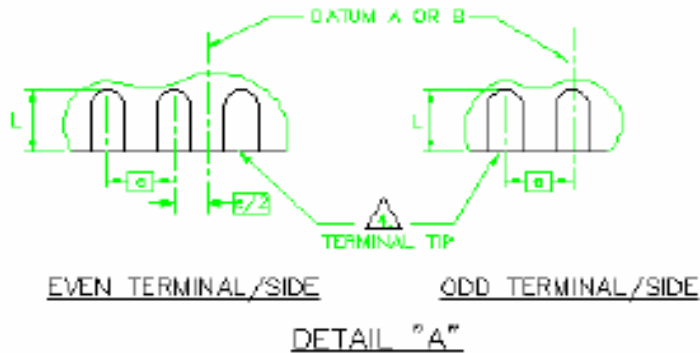
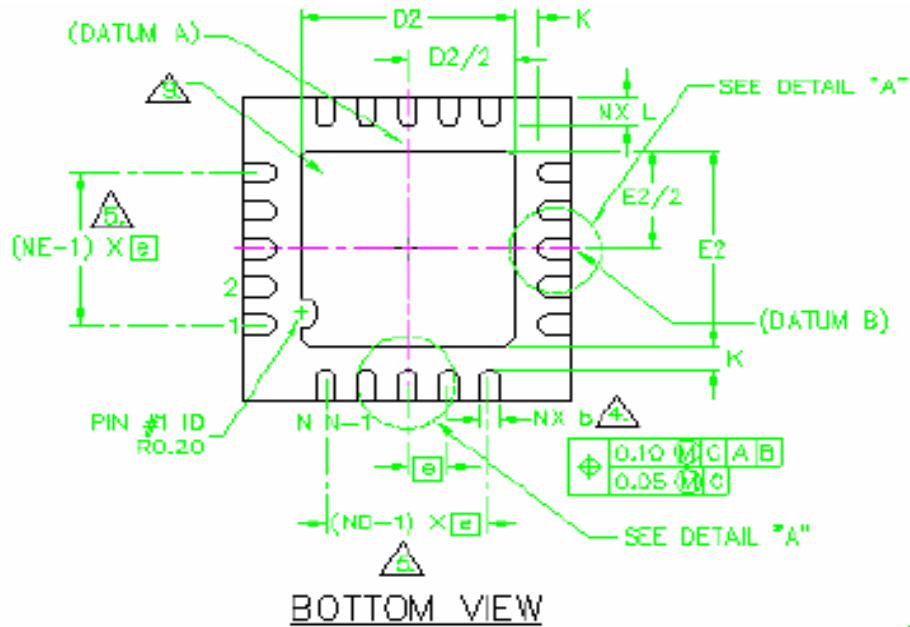
nRF24L01 uses the QFN20 4x4 package, with matt tin plating.



PRELIMINARY PRODUCT SPECIFICATION



nRF24L01 Single Chip 2.4 GHz Radio Transceiver



Package Type		A	A1	A3	K	D/E	e	D2/E2	L	L1	b
Saw QFN20 (4x4 mm)	Min	0.80	0.00	0.20	0.20	4.0 BSC ¹³	0.5 BSC	2.50	0.35	0.15	0.18
	Typ.	0.85	0.02	REF.	min			2.60	0.40	max	0.25
	Max	0.95	0.05					2.70	0.45		0.30

Figure 3 nRF24L01 Package Outline.

¹³ BSC: Basic Spacing between Centers, ref. JEDEC standard 9.5, page # 17-11/A



Package marking:

n	R	F		B	X
2	4	L	0	1	
Y	Y	W	W	L	L

Abbreviations:

- B – Build Code, i.e. unique code for production sites, package type and test platform
 X – "X" grade, i.e. Engineering Samples (optional)
 YY – 2 digit Year number
 WW – 2 digit Week number
 LL – 2 letter wafer lot number code

Absolute Maximum Ratings

Supply voltages

VDD - 0.3V to + 3.6V

VSS 0V

Input voltageV_I - 0.3V to 5.25V**Output voltage**V_O VSS to VDD**Total Power Dissipation**P_D (T_A=85°C) 60mW**Temperatures**

Operating Temperature.... - 40°C to + 85°C

Storage Temperature..... - 40°C to + 125°C

Note: Stress exceeding one or more of the limiting values may cause permanent damage to the device.

ATTENTION!

Electrostatic Sensitive Device
 Observe Precaution for handling.

**Glossary of Terms**



Term	Description
ACK	Acknowledgement
ART	Auto Re-Transmit
CE	Chip Enable
CLK	Clock
CRC	Cyclic Redundancy Check
CSN	Chip Select NOT
ESB	Enhanced ShockBurst™
GFSK	Gaussian Frequency Shift Keying
IRQ	Interrupt Request
ISM	Industrial-Scientific-Medical
LNA	Low Noise Amplifier
LSB	Least Significant Bit
LSByte	Least Significant Byte
Mbps	Megabit per second
MCU	Micro Controller Unit
MISO	Master In Slave Out
MOSI	Master Out Slave In
MSB	Most Significant Bit
MSByte	Most Significant Byte
PCB	Printed Circuit Board
PER	Packet Error Rate
PID	Packet Identity Bits
PLD	Payload
PRX	Primary RX
PTX	Primary TX
PWR_DWN	Power Down
PWR_UP	Power Up
RoHS	Restriction of use of Certain Hazardous Substances
RX	Receive
RX_DR	Receive Data Ready
SPI	Serial Peripheral Interface
TX	Transmit
TX_DS	Transmit Data Sent

Table 5 Glossary



FUNCTIONAL DESCRIPTION

Modes of operation

The nRF24L01 can be set in the following main modes depending on the level of the following primary I/Os and configuration registers:

Mode	PWR_UP register	PRIM_RX register	CE	FIFO state
RX mode	1	1	1	-
TX mode	1	0	1	Data in TX FIFO
TX mode	1	0	1 → 0	Stays in TX mode until packet transmission is finished
Standby-II	1	0	1	TX FIFO empty
Standby-I	1	-	0	No ongoing packet transmission
Power Down	0	-	-	-

Table 6 nRF24L01 main modes

An overview of the nRF24L01 I/O pins in different modes is given in Table 7.

Pin functions in the different modes of nRF24L01

Pin Name	Direction	TX Mode	RX Mode	Standby Modes	Power Down
CE	Input	High Pulse >10µs	High	Low	-
CSN	Input	SPI Chip Select, active low			
SCK	Input	SPI Clock			
MOSI	Input	SPI Serial Input			
MISO	Tri-state Output	SPI Serial Output			
IRQ	Output	Interrupt, active low			

Table 7 Pin functions of the nRF24L01

Standby Modes

Standby-I mode is used to minimize average current consumption while maintaining short start up times. In this mode, part of the crystal oscillator is active. In Standby-II mode some extra clock buffers are active compared to Standby-I mode. Standby-II occurs when CE is held high on a PTX device with empty TX FIFO. The configuration word content is maintained during Standby modes. SPI interface may be activated. For start up time see Table 13.

Power Down Mode

In power down nRF24L01 is disabled with minimal current consumption. When entering this mode the device is not active, but all registers values available from the SPI interface are maintained during power down and the SPI interface may be activated (CSN=0). For start up time see Table 13. The power down is controlled by the PWR_UP bit in the CONFIG register.



Packet Handling Methods

nRF24L01 has the following Packet Handling Methods:

- ShockBurst™ (compatible with nRF2401, nRF24E1, nRF2402 and nRF24E2 with 1Mbps data rate, see page 26)
- Enhanced ShockBurst™

ShockBurst™

ShockBurst™ makes it possible to use the high data rate offered by nRF24L01 without the need of a costly, high-speed microcontroller (MCU) for data processing/clock recovery. By placing all high speed signal processing related to RF protocol on-chip, nRF24L01 offers the application microcontroller a simple SPI compatible interface, the data rate is decided by the interface-speed the micro controller itself sets up. By allowing the digital part of the application to run at low speed, while maximizing the data rate on the RF link, ShockBurst™ reduces the average current consumption in applications.

In ShockBurst™ RX, IRQ notifies the MCU when a valid address and payload is received respectively. The MCU can then clock out the received payload from an nRF24L01 RX FIFO.

In ShockBurst™ TX, nRF24L01 automatically generates preamble and CRC, see Table 12. IRQ notifies the MCU that the transmission is completed. All together, this means reduced memory demand in the MCU resulting in a low cost MCU, as well as reduced software development time. nRF24L01 has a three level deep RX FIFO (shared between 6 pipes) and a three level deep TX FIFO. The MCU can access the FIFOs at any time, in power down mode, in standby modes, and during RF packet transmission. This allows the slowest possible SPI interface compared to the average data-rate, and may enable usage of an MCU without hardware SPI.

Enhanced ShockBurst™

Enhanced ShockBurst™ is a packet handling method with functionality that makes bi-directional link protocol implementation easier and more efficient. In a typical bi-directional link, one will let the terminating part acknowledge received packets from the originating part in order to make it possible to detect data loss. Data loss can then be recovered by retransmission. The idea with Enhanced ShockBurst™ is to let nRF24L01 handle both acknowledgement of received packets and retransmissions of lost packets, without involvement from the microcontroller.

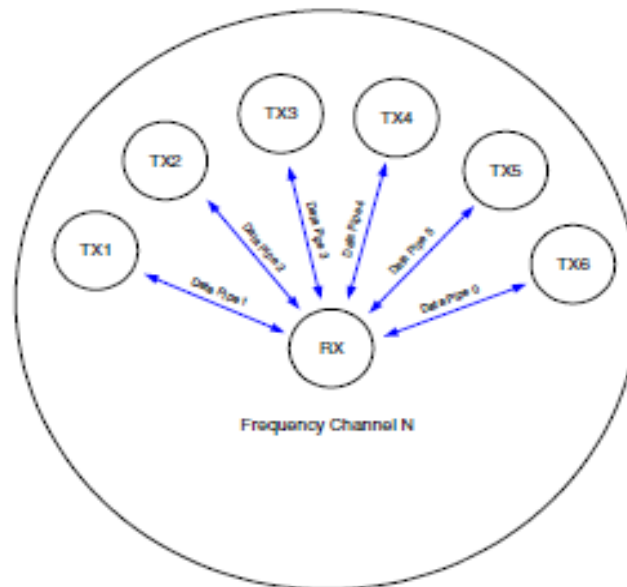


Figure 4: nRF24L01 in a star network configuration

An nRF24L01 configured as primary RX (PRX) will be able to receive data through 6 different data pipes, see Figure 4. A data pipe will have a unique address but share the same frequency channel. This means that up to 6 different nRF24L01 configured as primary TX (PTX) can communicate with one nRF24L01 configured as PRX, and the nRF24L01 configured as PRX will be able to distinguish between them. Data pipe 0 has a unique 40 bit configurable address. Each of data pipe 1-5 has an 8 bit unique address and shares the 32 most significant address bits. All data pipes can perform full Enhanced ShockBurst™ functionality.

nRF24L01 will use the data pipe address when acknowledging a received packet. This means that nRF24L01 will transmit ACK with the same address as it receives payload at. In the PTX device data pipe 0 is used to receive the acknowledgement, and therefore the receive address for data pipe 0 has to be equal to the transmit address to be able to receive the acknowledgement. See Figure 5 for addressing example.

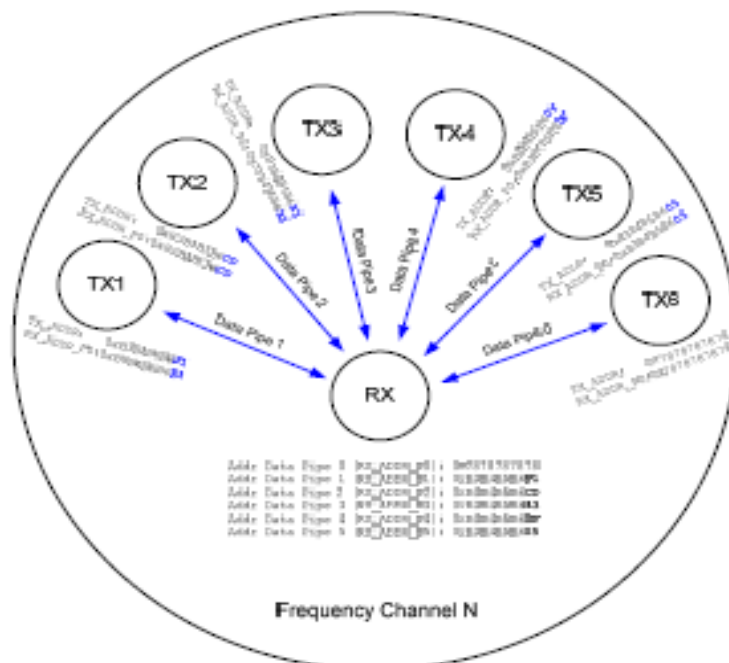


Figure 5: Example on how the acknowledgement addressing is done

An nRF24L01 configured as PTX with Enhanced ShockBurst™ enabled, will use the ShockBurst™ feature to send a packet whenever the microcontroller wants to. After the packet has been transmitted, nRF24L01 will switch on its receiver and expect an acknowledgement to arrive from the terminating part. If this acknowledgement fails to arrive, nRF24L01 will retransmit the same packet until it receives an acknowledgement or the number of retries exceeds the number of allowed retries given in the SETUP_RETR_ARC register. If the number of retries exceeds the number of allowed retries, this will be showed by the STATUS register bit MAX_RT which gives an interrupt.

Whenever an acknowledgement is received by an nRF24L01 it will consider the last transmitted packet as delivered. It will then be cleared from the TX FIFO, and the TX_DS IRQ source will be set high.

With Enhanced ShockBurst™ nRF24L01 offers the following benefits:

- Highly reduced current consumption due to short time on air and sharp timing when operating with acknowledgement traffic
- Lower system cost. Since the nRF24L01 handles all the high-speed link layer operations, like re-transmission of lost packet and generating acknowledgement to received packets, it is no need for hardware SPI on the system microcontroller to interface the nRF24L01. The interface can be done by using general purpose IO pins on a low cost microcontroller where the SPI is emulated in firmware. With the nRF24L01 this will be sufficient speed even when running a bi-directional link.
- Greatly reduced risk of “on-air” collisions due to short time on air
- Easier firmware development since the link layer is integrated on chip

**Enhanced ShockBurst™ Transmitting Payload:**

1. The configuration bit PRIM_RX has to be low.
2. When the application MCU has data to send, the address for receiving node (TX_ADDR) and payload data (TX_PLD) has to be clocked into nRF24L01 via the SPI interface. The width of TX-payload is counted from number of bytes written into the TX FIFO from the MCU. TX_PLD must be written continuously while holding CSN low. TX_ADDR does not have to be rewritten if it is unchanged from last transmit. If the PTX device shall receive acknowledgement, data pipe 0 has to be configured to receive the acknowledgement. The receive address for data pipe 0 (RX_ADDR_P0) has to be equal to the transmit address (TX_ADDR) in the PTX device. For the example in Figure 5 the following address settings have to be performed for the TX5 device and the RX device:
TX5 device: TX_ADDR = 0xB3B4B5B605
TX5 device: RX_ADDR_P0 = 0xB3B4B5B605
RX device: RX_ADDR_P5 = 0xB3B4B5B605
3. A high pulse on CE starts the transmission. The minimum pulse width on CE is 10 µs.
4. nRF24L01 ShockBurst™:
 - Radio is powered up
 - 16 MHz internal clock is started.
 - RF packet is completed (see the packet description)
 - Data is transmitted at high speed (1 Mbps or 2 Mbps configured by MCU).
5. If auto acknowledgement is activated (EN_AA_P0=1) the radio goes into RX mode immediately. If a valid packet has been received in the valid acknowledgement time window, the transmission is considered a success. The TX_DS bit in the status register is set high and the payload is removed from TX FIFO. If a valid acknowledgement is not received in the specified time window, the payload is resent (if auto retransmit is enabled). If the auto retransmit counter (ARC_CNT) exceeds the programmed maximum limit (ARC), the MAX_RT bit in the status register is set high. The payload in TX FIFO is NOT removed. The IRQ pin will be active when MAX_RT or TX_DS is high. To turn off the IRQ pin, the interrupt source must be reset by writing to the status register (see Interrupt chapter). If no acknowledgement is received for a packet after the maximum number of retries, no further packets can be sent before the MAX_RX interrupt is cleared. The packet loss counter (PLOS_CNT) is incremented at each MAX_RT interrupt. I.e. ARC_CNT counts the number of retries that was required to get a single packet through. PLOS_CNT counts the number of packets that did not get through after maximum number of retries.
6. The device goes into Standby-I mode if CE is low. Otherwise next payload in TX FIFO will be sent. If TX FIFO is empty and CE is still high, the device will enter Standby-II mode.
7. If the device is in Standby-II mode, it will go to Standby-I mode immediately if CE is set low.

Enhanced ShockBurst™ Receive Payload:

1. RX is selected by setting the PRIM_RX bit in the configuration register to high. All data pipes that shall receive data must be enabled (EN_RXADDR register),



- auto acknowledgement for all pipes running Enhanced ShockBurst™ has to be enabled (EN_AA register), and the correct payload widths must be set (RX_PW_Px registers). Addresses have to be set up as described in item 2 in the Enhanced ShockBurst™ transmit payload chapter above.
2. Active RX mode is started by setting CE high.
 3. After 130µs nRF24L01 is monitoring the air for incoming communication.
 4. When a valid packet has been received (matching address and correct CRC), the payload is stored in the RX-FIFO, and the RX_DR bit in status register is set high. The IRQ pin will be active when RX_DR is high. RX_P_NO in status register will indicate what data pipe the payload has been received in.
 5. If auto acknowledgement is enabled, an acknowledgement is sent back.
 6. MCU sets the CE pin low to enter Standby-I mode (low current mode).
 7. MCU can clock out the payload data at a suitable rate via the SPI interface.
 8. The device is now ready for entering TX or RX mode or power down mode.

Two way communication with payload in both directions

If payload shall be sent in both directions, the PRIM_RX register must be toggled by redefining the device from PRX to PTX or vice versa. The controlling processors must handle the synchronicity between a PTX and a PRX. Data buffering in both RX FIFO and TX FIFO simultaneously is possible, but restricted to data pipes 1 to 5. The third level in TX FIFO shall only be written in RX, TX or Standby-II mode if data is stored in RX FIFO

Auto Acknowledgement (RX)

The auto acknowledgement function reduces the load of the external microcontroller, and may remove the need for dedicated SPI hardware in a mouse/keyboard or comparable systems, and hence reduce cost and average current consumption. Auto acknowledgement can be configured individually for each data pipe via the SPI interface.

If auto acknowledgement is enabled and a valid packet (correct data pipe address and CRC) is received, the device will enter TX mode and send an acknowledgement packet. After the device has sent the acknowledgement packet, normal operation resumes, and the mode is determined by the PRIM_RX register and CE pin.

Auto Re-Transmission (ART) (TX)

An auto retransmission function is available. It will be used at the TX side in an auto acknowledgement system. In the SETUP_RETR register it will be possible to state how many times the data in the data register will be resent if data is not acknowledged. After each sending, the device will enter RX mode and wait a specified time period for acknowledgement. When the acknowledgement packet is received, the device will return to normal transmit function. If there is no more unsent data in the TX FIFO and the CE pin is low, the device will go into Standby-I mode. If the acknowledgement is not received, the device will go back to TX mode and resend the data. This will continue until acknowledgment is received, or a time out occurs


nRF24L01 Single Chip 2.4 GHz Radio Transceiver

(i.e. the maximum number of sending is reached). The only way to reset this is to set the PWR_UP bit low or let the auto retransmission finish. A packet loss counter will be incremented each time a packet does not succeed to reach the destination before time out. (Time out is indicated by the MAX_RT interrupt.) The packet loss counter is reset when writing to the RF channel register.

Packet Identity (PID) and CRC used by Enhanced ShockBurst™

Each packet contains a two bit wide PID field to detect if the received packet is new or resent. The PID will prevent that the PRX device presents the same payload more than once to the microcontroller. This PID field is incremented at the TX side for each new packet received via the SPI interface. The PID and CRC field is used by the PRX device to determine whether a packet is resent or new. When several data is lost on the link, the PID fields may in some cases become equal to last received PID. If a packet has the same PID as the previous packet, nRF24L01 will compare the CRC sums from both packets. If they also are equal, the last received packet is considered as a copy of the previous and is discarded.

1: PRX device:

The PRX device compares the received PID with the last PID. If the PID fields are different, the packet is considered to be new. If the PID is equal to last received PID, the received packet might be the same as last time. The receiver must check if the CRC is equal to the previous CRC. If the CRC is equal to the previous one, the packet is probably the same, and will be discarded.

2: PTX device:

The transmitter increments the PID field each time it sends a new packet.

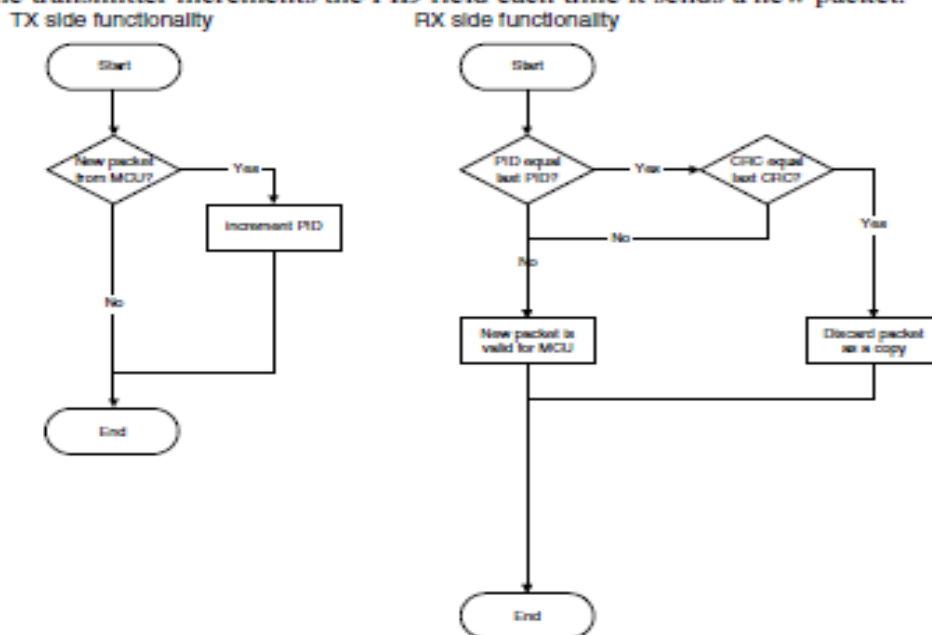


Figure 6 PID generation/detection



nRF24L01 Single Chip 2.4 GHz Radio Transceiver

The length of the CRC is configurable through the SPI interface. It is important to notice that the CRC is calculated over the whole packet including address, PID and payload. No packet is accepted as correct if the CRC fails. This is an extra requirement for packet acceptance that is not illustrated in the figure above.

Stationary Disturbance Detection – CD

Carrier Detect (CD) is set high when an in-band RF signal is detected in RX mode, otherwise CD is low. The internal CD signal is filtered before presented to CD register. The internal CD signal must be high for at least 128 μ s.

In Enhanced ShockBurst™ it is recommended to use the Carrier Detect functionality only when the PTX device does not succeed to get packets through, as indicated by the MAX_RT interrupt for single packets and by the packet loss counter (PLOS_CNT) if several packets are lost. If the PLOS_CNT in the PTX device indicates to high rate of packet losses, the device can be configured to a PRX device for a short time ($T_{stab2a} + \text{CD-filter delay} = 130\mu\text{s} + 128\mu\text{s} = 258\mu\text{s}$) to check CD. If CD was high (jam situation), the frequency channel should be changed. If CD was low (out of range), it may continue on the same frequency channel, but perform other adjustments. (A dummy write to the RF_CH will clear the PLOS_CNT.)

Data Pipes

nRF24L01 configured as PRX can receive data addressed to 6 different data pipes in one physical frequency channel. Each data pipe has its own unique address and can be configured to have individual behavior.

The data pipes are enabled with the bits in the EN_RXADDR register. By default only data pipe 0 and 1 are enabled.

The address for each data pipe is configured in the RX_ADDR_Px registers. Always ensure that none of the data pipes have the exact same address.

Data pipe 0 has a unique 40 bit configurable address. Data pipes 1-5 share the 32 most significant address bits and have only the LSByte unique for each data pipe. Figure 7 shows an example of how data pipes 0-5 are addressed. All pipes can have up to 40 bit address, but for pipe 1-5 only the LSByte is different, and the LSByte must be unique for all pipes.



	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Data pipe 0 (RX_ADDR_P0)	0xE7	0xD3	0xF0	0x35	0x77
Data pipe 1 (RX_ADDR_P1)	0xC2	0xC2	0xC2	0xC2	0xC2
Data pipe 2 (RX_ADDR_P2)	0xC2	0xC2	0xC2	0xC2	0xC3
Data pipe 3 (RX_ADDR_P3)	0xC2	0xC2	0xC2	0xC2	0xC4
Data pipe 4 (RX_ADDR_P4)	0xC2	0xC2	0xC2	0xC2	0xC5
Data pipe 5 (RX_ADDR_P5)	0xC2	0xC2	0xC2	0xC2	0xC6

Figure 7: Addressing data pipes 0-5

When a packet has been received at one of the data pipes and the data pipe is setup to generate acknowledgement, nRF24L01 will generate an acknowledgement with an address that equals the data pipe address where the packet was received.

Some configuration settings are common to all data pipes and some are individual. The following settings are common to all data pipes:

- CRC enabled/disabled (CRC always enabled when ESB is enabled)
- CRC encoding scheme
- RX address width
- Frequency channel
- RF data rate
- LNA gain
- RF output power



DEVICE CONFIGURATION

All configuration of nRF24L01 is defined by values in some configuration registers. All these registers are writable via the SPI interface.

SPI Interface

The SPI interface is a standard SPI interface with a maximum data rate of 10Mbps. Most registers are readable.

SPI Instruction Set

The available commands to be used on the SPI interface are shown below. Whenever CSN is set low the interface expects an instruction. Every new instruction must be started by a high to low transition on CSN.

In parallel to the SPI instruction word applied on the MOSI pin, the STATUS register is shifted serially out on the MISO pin.

The serial shifting SPI commands is on the format:

<**Instruction word:** MSBit to LSBit (one byte)>

<**Data bytes:** LSByte to MSByte, MSBit in each byte first>

See Figure 8 and Figure 9.

Instruction Name	Instruction Format [binary]	# Data Bytes	Operation
R_REGISTER	000A AAAA	1 to 5 LSByte first	Read registers. AAAAA = 5 bit Memory Map Address
W_REGISTER	001A AAAA	1 to 5 LSByte first	Write registers. AAAAA = 5 bit Memory Map Address <i>Executable in power down or standby modes only.</i>
R_RX_PAYLOAD	0110 0001	1 to 32 LSByte first	Read RX-payload: 1 – 32 bytes. A read operation will always start at byte 0. Payload will be deleted from FIFO after it is read. Used in RX mode.
W_TX_PAYLOAD	1010 0000	1 to 32 LSByte first	Used in TX mode. Write TX-payload: 1 – 32 bytes. A write operation will always start at byte 0.
FLUSH_TX	1110 0001	0	Flush TX FIFO, used in TX mode
FLUSH_RX	1110 0010	0	Flush RX FIFO, used in RX mode Should not be executed during transmission of acknowledge, i.e. acknowledge package will not be completed.
REUSE_TX_PL	1110 0011	0	Used for a PTX device Reuse last sent payload. Packets will be repeatedly resent as long as CE is high. TX payload reuse is active until W_TX_PAYLOAD or FLUSH TX is executed. TX payload reuse must not be activated or deactivated during package transmission
NOP	1111 1111	0	No Operation. Might be used to read the STATUS register

Table 8 Instruction set for the nRF24L01 SPI interface.

The W_REGISTER and R_REGISTER may operate on single or multi-byte registers. When accessing multi-byte registers one will read or write MSBit of LSByte first. The



nRF24L01 Single Chip 2.4 GHz Radio Transceiver

writing can be terminated before all bytes in a multi-byte register has been written. In this case the unwritten MSByte(s) will remain unchanged. E.g. the LSByte of RX_ADDR_P0 can be modified by writing only one byte to the RX_ADDR_P0 register. The content of the status register will always be read to MISO after a high to low transition on CSN.

Interrupt

The nRF24L01 has an active low interrupt pin (IRQ). The interrupt pin is activated when TX_DS, RX_DR or MAX_RT is set high in status register. When MCU writes '1' to the interrupt source, the IRQ pin will go inactive. The interrupt mask part of the CONFIG register is used to mask out the interrupt sources that are allowed to set the IRQ pin low. By setting one of the MASK bits high, the corresponding interrupt source will be disabled. By default all interrupt sources are enabled.

SPI Timing

The interface supports SPI. SPI operation and timing is given in Figure 8 to Figure 10 and in Table 9 and Table 10. The device must be in one of the standby modes or power down mode before writing to the configuration registers. In Figure 8 to Figure 10 the following notations are used:

Cn – SPI Instruction Bit

Sn – Status Register Bit

Dn – Data Bit (note: LSByte to MSByte, MSBit in each byte first)



Figure 8 SPI read operation.



Figure 9 SPI write operation.



Figure 10 SPI NOP timing diagram.

PARAMETER	SYMBOL	MIN	MAX	UNITS
Data to SCK Setup	Tdc	2		ns
SCK to Data Hold	Tdh	2		ns
CSN to Data Valid	Tcsd		38	ns
SCK to Data Valid	Tcd		55	ns
SCK Low Time	Tcl	40		ns
SCK High Time	Tch	40		ns
SCK Frequency	Fsck	0	8	MHz
SCK Rise and Fall	Tr,Tf		100	ns
CSN to SCK Setup	Tcc	2		ns
SCK to CSN Hold	Tcch	2		ns
CSN Inactive time	Tcwh	50		ns
CSN to Output High Z	Tcdz		38	ns

Table 9 SPI timing parameters ($C_{Load} = 5pF$).

PARAMETER	SYMBOL	MIN	MAX	UNITS
Data to SCK Setup	Tdc	2		ns
SCK to Data Hold	Tdh	2		ns
CSN to Data Valid	Tcsd		42	ns
SCK to Data Valid	Tcd		58	ns
SCK Low Time	Tcl	40		ns
SCK High Time	Tch	40		ns
SCK Frequency	Fsck	0	8	MHz
SCK Rise and Fall	Tr,Tf		100	ns
CSN to SCK Setup	Tcc	2		ns
SCK to CSN Hold	Tcch	2		ns
CSN Inactive time	Tcwh	50		ns
CSN to Output High Z	Tcdz		42	ns

Table 10 SPI timing parameters ($C_{Load} = 10pF$).



MEMORY MAP

All undefined bits in the table below are redundant. They will be read out as '0'.

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
00	CONFIG				Configuration Register
	Reserved	7	0	R/W	Only '0' allowed
	MASK_RX_DR	6	0	R/W	Mask interrupt caused by RX_DR 1: Interrupt not reflected on the IRQ pin 0: Reflect RX_DR as active low interrupt on the IRQ pin
	MASK_TX_DS	5	0	R/W	Mask interrupt caused by TX_DS 1: Interrupt not reflected on the IRQ pin 0: Reflect TX_DS as active low interrupt on the IRQ pin
	MASK_MAX_RT	4	0	R/W	Mask interrupt caused by MAX_RT 1: Interrupt not reflected on the IRQ pin 0: Reflect MAX_RT as active low interrupt on the IRQ pin
	EN_CRC	3	1	R/W	Enable CRC. Forced high if one of the bits in the EN_AA is high
	CRCO	2	0	R/W	CRC encoding scheme '0' - 1 byte '1' - 2 bytes
	PWR_UP	1	0	R/W	1: POWER UP, 0:POWER DOWN
	PRIM_RX	0	0	R/W	1: PRX, 0: PTX
01	EN_AA Enhanced ShockBurst™				Enable 'Auto Acknowledgment' Function Disable this functionality to be compatible with nRF2401, see page 26
	Reserved	7:6	00	R/W	Only '00' allowed
	ENAA_P5	5	1	R/W	Enable auto ack. data pipe 5
	ENAA_P4	4	1	R/W	Enable auto ack. data pipe 4
	ENAA_P3	3	1	R/W	Enable auto ack. data pipe 3
	ENAA_P2	2	1	R/W	Enable auto ack. data pipe 2
	ENAA_P1	1	1	R/W	Enable auto ack. data pipe 1
	ENAA_P0	0	1	R/W	Enable auto ack. data pipe 0
02	EN_RXADDR				Enabled RX Addresses
	Reserved	7:6	00	R/W	Only '00' allowed
	ERX_P5	5	0	R/W	Enable data pipe 5.
	ERX_P4	4	0	R/W	Enable data pipe 4.
	ERX_P3	3	0	R/W	Enable data pipe 3.
	ERX_P2	2	0	R/W	Enable data pipe 2.
	ERX_P1	1	1	R/W	Enable data pipe 1.
	ERX_P0	0	1	R/W	Enable data pipe 0.
03	SETUP_AW				Setup of Address Widths (common for all data pipes)
	Reserved	7:2	000000	R/W	Only '000000' allowed
	AW	1:0	11	R/W	RX/TX Address field width '00' - Illegal '01' - 3 bytes '10' - 4 bytes '11' - 5 bytes LSByte will be used if address width below 5 bytes
04	SETUP_RETR				Setup of Automatic Retransmission
	ARD	7:4	0000	R/W	Auto Re-transmit Delay '0000' - Wait 250+86µS

PRELIMINARY PRODUCT SPECIFICATION



nRF24L01 Single Chip 2.4 GHz Radio Transceiver

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
					'0001' – Wait 500+86µS '0010' – Wait 750+86µS '1111' – Wait 4000+86µS (Delay defined from end of transmission to start of next transmission) ¹⁴
	ARC	3:0	0011	R/W	Auto Retransmit Count '0000' – Re-Transmit disabled '0001' – Up to 1 Re-Transmit on fail of AA '1111' – Up to 15 Re-Transmit on fail of AA
05	RF_CH				RF Channel
	Reserved	7	0	R/W	Only '0' allowed
	RF_CH	6:0	0000010	R/W	Sets the frequency channel nRF24L01 operates on
06	RF_SETUP				RF Setup Register
	Reserved	7:5	000	R/W	Only '000' allowed
	PLL_LOCK	4	0	R/W	Force PLL lock signal. Only used in test
	RF_DR	3	1	R/W	Data Rate '0' – 1 Mbps '1' – 2 Mbps
	RF_PWR	2:1	11	R/W	Set RF output power in TX mode '00' – -18 dBm '01' – -12 dBm '10' – -6 dBm '11' – 0 dBm
	LNA_HCURR	0	1	R/W	Setup LNA gain
07	STATUS				Status Register (In parallel to the SPI instruction word applied on the MOSI pin, the STATUS register is shifted serially out on the MISO pin)
	Reserved	7	0	R/W	Only '0' allowed
	RX_DR	6	0	R/W	Data Ready RX FIFO interrupt. Set high when new data arrives RX FIFO ¹⁵ . Write 1 to clear bit.
	TX_DS	5	0	R/W	Data Sent TX FIFO interrupt. Set high when packet sent on TX. If AUTO_ACK is activated, this bit will be set high only when ACK is received. Write 1 to clear bit.
	MAX_RT	4	0	R/W	Maximum number of TX retries interrupt. Write 1 to clear bit. If MAX_RT is set it must be cleared to enable further communication.
	RX_P_NO	3:1	111	R	Data pipe number for the payload

¹⁴ Accurate formula for delay from start of transmission, to start of re-transmission:

$$TRD (\mu s) = 250\mu s * (ARD+1) + 4\mu s *(AW + PW + CRCW) + 138,5\mu s.$$

TRD= total retransmit delay, AW=Address Width (#bytes), PW=Payload Width(#bytes)

, CRCW= CRC Width (#bytes)

¹⁵ The Data Ready interrupt is set by a new packet arrival event. The procedure for handling this interrupt should be: 1) read payload via SPI, 2) clear RX_DR interrupt, 3) read FIFO_STATUS to check if there are more payloads available in RX FIFO, 4) if there are more data in RX FIFO, repeat from 1).

PRELIMINARY PRODUCT SPECIFICATION



nRF24L01 Single Chip 2.4 GHz Radio Transceiver

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
					available for reading from RX_FIFO 000-101: Data Pipe Number 110: Not Used 111: RX FIFO Empty
	TX_FULL	0	0	R	TX FIFO full flag. 1: TX FIFO full. 0: Available locations in TX FIFO.
08	OBSERVE_TX PLOS_CNT	7:4	0	R	Transmit observe register Count lost packets. The counter is overflow protected to 15, and discontinues at max until reset. The counter is reset by writing to RF_CH. See page 14 and 17.
	ARC_CNT	3:0	0	R	Count resent packets. The counter is reset when transmission of a new packet starts. See page 14.
09	CD				
	Reserved	7:1	000000	R	
	CD	0	0	R	Carrier Detect. See page 17.
0A	RX_ADDR_P0	39:0	0xd7E7E7E7E7	R/W	Receive address data pipe 0. 5 Bytes maximum length. (LSByte is written first. Write the number of bytes defined by SETUP_AW)
0B	RX_ADDR_P1	39:0	0xC2C2C2C2C2	R/W	Receive address data pipe 1. 5 Bytes maximum length. (LSByte is written first. Write the number of bytes defined by SETUP_AW)
0C	RX_ADDR_P2	7:0	0xC3	R/W	Receive address data pipe 2. Only LSB. MSBytes will be equal to RX_ADDR_P1[39:8]
0D	RX_ADDR_P3	7:0	0xC4	R/W	Receive address data pipe 3. Only LSB. MSBytes will be equal to RX_ADDR_P1[39:8]
0E	RX_ADDR_P4	7:0	0xC5	R/W	Receive address data pipe 4. Only LSB. MSBytes will be equal to RX_ADDR_P1[39:8]
0F	RX_ADDR_P5	7:0	0xC6	R/W	Receive address data pipe 5. Only LSB. MSBytes will be equal to RX_ADDR_P1[39:8]
10	TX_ADDR	39:0	0xd7E7E7E7E7	R/W	Transmit address. Used for a PTX device only. (LSByte is written first) Set RX_ADDR_P0 equal to this address to handle automatic acknowledge if this is a PTX device with Enhanced ShockBurst™ enabled. See page 14.
11	RX_PW_P0				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P0	5:0	0	R/W	Number of bytes in RX payload in data pipe 0 (1 to 32 bytes). 0 Pipe not used 1 = 1 byte ... 32 = 32 bytes
12	RX_PW_P1				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P1	5:0	0	R/W	Number of bytes in RX payload in data pipe 1 (1 to 32 bytes). 0 Pipe not used

PRELIMINARY PRODUCT SPECIFICATION



nRF24L01 Single Chip 2.4 GHz Radio Transceiver

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
					1 = 1 byte ... 32 = 32 bytes
13	RX_PW_P2				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P2	5:0	0	R/W	Number of bytes in RX payload in data pipe 2 (1 to 32 bytes). 0 Pipe not used 1 = 1 byte ... 32 = 32 bytes
14	RX_PW_P3				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P3	5:0	0	R/W	Number of bytes in RX payload in data pipe 3 (1 to 32 bytes). 0 Pipe not used 1 = 1 byte ... 32 = 32 bytes
15	RX_PW_P4				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P4	5:0	0	R/W	Number of bytes in RX payload in data pipe 4 (1 to 32 bytes). 0 Pipe not used 1 = 1 byte ... 32 = 32 bytes
16	RX_PW_P5				
	Reserved	7:6	00	R/W	Only '00' allowed
	RX_PW_P5	5:0	0	R/W	Number of bytes in RX payload in data pipe 5 (1 to 32 bytes). 0 Pipe not used 1 = 1 byte ... 32 = 32 bytes
17	FIFO_STATUS				FIFO Status Register
	Reserved	7	0	R/W	Only '0' allowed
	TX_REUSE	6	0	R	Reuse last sent data packet if set high. The packet will be repeatedly resent as long as CE is high. TX_REUSE is set by the SPI instruction REUSE_TX_PL, and is reset by the SPI instructions W_TX_PAYLOAD or FLUSH_TX.
	TX_FULL	5	0	R	TX FIFO full flag. 1: TX FIFO full. 0: Available locations in TX FIFO.
	TX_EMPTY	4	1	R	TX FIFO empty flag. 1: TX FIFO empty. 0: Data in TX FIFO.
	Reserved	3:2	00	R/W	Only '00' allowed
	RX_FULL	1	0	R	RX FIFO full flag. 1: RX FIFO full. 0: Available locations in RX FIFO.
	RX_EMPTY	0	1	R	RX FIFO empty flag. 1: RX FIFO empty. 0: Data in RX FIFO.
N/A	TX_PLD	255:0	X	W	Written by separate SPI command TX data payload register 1 - 32 bytes. This register is implemented as a FIFO with 3 levels. Used in TX mode only
N/A	RX_PLD	255:0	X	R	Written by separate SPI command RX data payload register. 1 - 32 bytes. This register is implemented as a FIFO



nRF24L01 Single Chip 2.4 GHz Radio Transceiver

Address (Hex)	Mnemonic	Bit	Reset Value	Type	Description
					with 3 levels. All receive channels share the same FIFO

Table 11 Memory map of nRF24L01

Configuration for compatibility with nRF24XX

How to setup nRF24L01 to receive from an nRF2401/nRF2402/nRF24E1/nRF24E2:

- Use same CRC configuration as the nRF2401/nRF2402/nRF24E1/nRF24E2 uses
- Set the PRIM_RX bit to 1
- Disable auto acknowledgement on the data pipe that will be addressed
- Use the same address width as the PTX device
- Use the same frequency channel as the PTX device
- Select data rate 1Mbit/s on both nRF24L01 and nRF2401/nRF2402/nRF24E1/nRF24E2
- Set correct payload width on the data pipe that will be addressed
- Set PWR_UP and CE high

How to setup nRF24L01 to transmit to an nRF2401/nRF24E1:

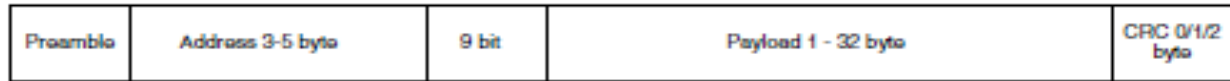
- Use same CRC configuration as the nRF2401/nRF2402/nRF24E1/nRF24E2 uses
- Set the PRIM_RX bit to 0
- Set the Auto Retransmit Count to 0 to disable the auto retransmit functionality
- Use the same address width as the nRF2401/nRF2402/nRF24E1/nRF24E2 uses
- Use the same frequency channel as the nRF2401/nRF2402/nRF24E1/nRF24E2 uses
- Select data rate 1Mbit/s on both nRF24L01 and nRF2401/nRF2402/nRF24E1/nRF24E2
- Set PWR_UP high
- Clock in a payload that has the same length as the nRF2401/nRF2402/nRF24E1/nRF24E2 is configured to receive
- Pulse CE to send the packet



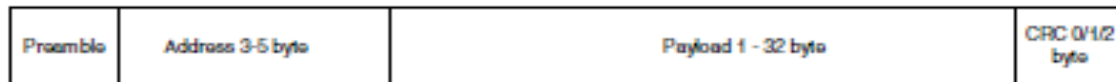
nRF24L01 Single Chip 2.4 GHz Radio Transceiver

PACKET DESCRIPTION

An Enhanced ShockBurst™ packet with payload (1-32 bytes).



A ShockBurst™ packet compatible to nRF2401/nRF2402/nRF24E1/nRF24E2 devices.



Preamble	<ul style="list-style-type: none"> Preamble is used to detect 0 and 1 levels. It is stripped off (RX) and added (TX) by nRF24L01.
Address	<ul style="list-style-type: none"> The address field contains the receiver address. The address can be 3, 4 or 5 bytes wide The address fields can be individually configured for all RX channels and the TX channel Address is automatically removed from received packets.¹⁶
Flags	<ul style="list-style-type: none"> PID: Packet Identification. 2 bits that is incremented for each new payload 7 bits reserved for packet compatibility with future products Not used when compatible to nRF2401/nRF24E1
Payload	<ul style="list-style-type: none"> 1 - 32 bytes wide.
CRC	<ul style="list-style-type: none"> The CRC is optional. 0-2 bytes wide CRC The polynomial for 8 bits CRC check is $X^8 + X^2 + X + 1$ The polynomial for 16 bits CRC check is $X^{16} + X^{12} + X^5 + 1$.

Table 12 Data packet description

¹⁶ Suggested use of addresses. In general more bits in the address gives less false detection, which in the end may give lower data Packet-Error-Rate (PER).

- The address made by (5, 4, or 3) equal bytes are not recommended because it in general will make the packet-error-rate increase.
- Addresses where the level shift only one time (i.e. 000FFFFFFF) could often be detected in noise that may give a false detection, which again may give raised packet-error-rate.
- Addresses as a continuation of the preamble (hi-low toggling) will raise the Packet-Error-Rate (PER).



IMPORTANT TIMING DATA

The following timing applies for operation of nRF24L01.

nRF24L01 Timing Information

nRF24L01 timing	Max.	Min.	Name
Power Down → Standby mode	1.5ms		Tpd2stby
Standby modes → TX/RX mode	130µs		Tstby2a
Minimum CE high		10µs	Tbce
Delay from CE pos. edge to CSN low		4µs	Tpcc2csn

Table 13 Operational timing of nRF24L01

When the nRF24L01 is in power down it must always settle in Standby for 1.5ms before it can enter one of the TX or RX modes. Note that the configuration word will be lost if VDD is turned off and that the device then must be configured before going to one of the TX or RX mode.

Enhanced ShockBurst™ timing

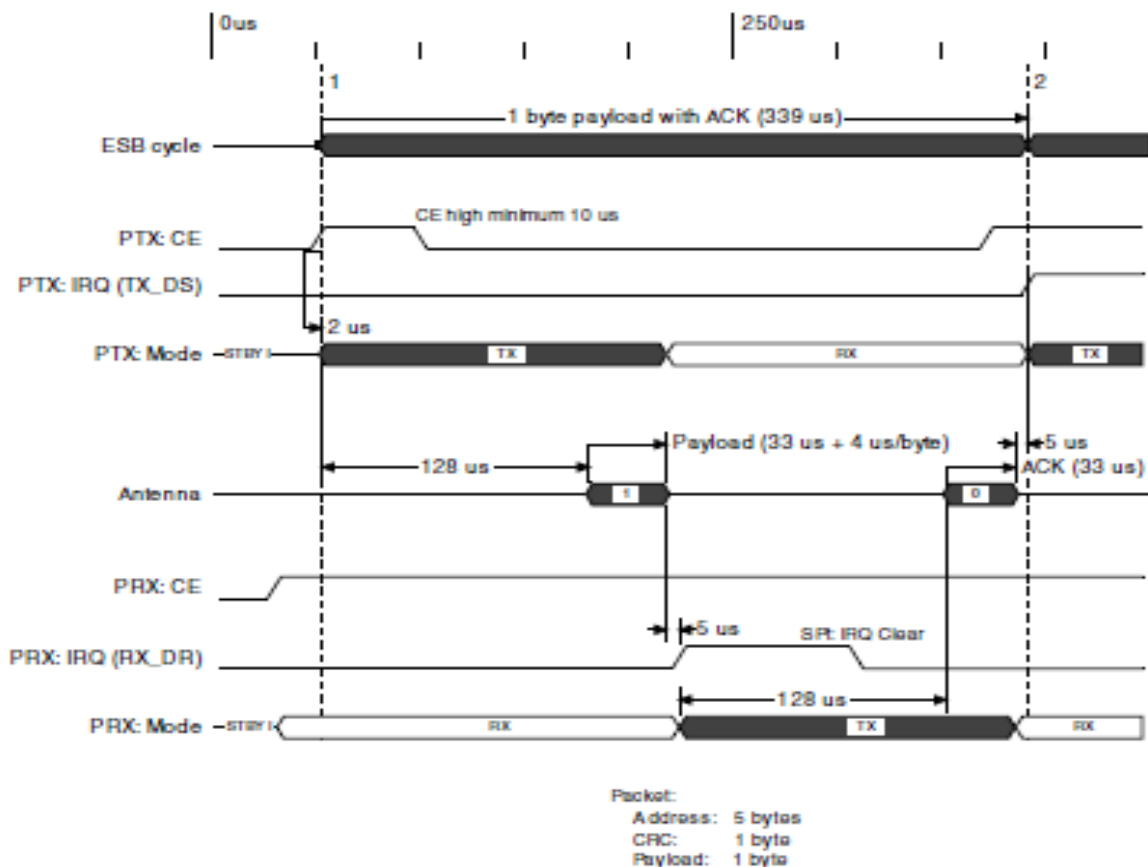


Figure 11 Timing of Enhanced ShockBurst™ for one packet upload (2Mbps).



nRF24L01 Single Chip 2.4 GHz Radio Transceiver

In Figure 11 the sending of one packet and the acknowledgement of this packet is shown. The loading of payload to the PTX device is not shown in the figure. The PRX device is turned into RX mode (CE=1), and the PTX device is set into TX mode (CE=1 for minimum 10 μ s). After 130 μ s the transmission starts and is finished after another 37 μ s (1 byte payload). The transmission ends, and the PTX device is automatically turned around to RX mode to wait for the acknowledgement from the PRX device. After the PTX device has received the acknowledgement it gives an interrupt to the MCU (IRQ (TX_DS) =>TX-data sent). After the PRX device has received the packet it gives an interrupt to the MCU (IRQ (RX_DR) =>RX-data ready).



PERIPHERAL RF INFORMATION

Antenna output

The ANT1 & ANT2 output pins provide a balanced RF output to the antenna. The pins must have a DC path to VDD, either via a RF choke or via the center point in a dipole antenna. A load of $15\Omega + j88\Omega$ (simulated values) is recommended for maximum output power (0dBm). Lower load impedance (for instance 50Ω) can be obtained by fitting a simple matching network between the load and ANT1 and ANT2.

Output Power adjustment

SPI RF-SETUP (RF_PWR)	RF output power	DC current consumption
11	0 dBm	11.3 mA
10	-6 dBm	9.0 mA
01	-12 dBm	7.5 mA
00	-18 dBm	7.0 mA

Conditions: VDD = 3.0V, VSS = 0V, T_A = 27°C, Load impedance = $15\Omega + j88\Omega$.

Table 14 RF output power setting for the nRF24L01.

Crystal Specification

Frequency accuracy includes initial accuracy (tolerance) and stability over temperature and aging.

Frequency	C _L	ESR max	C _{0max}	Frequency accuracy
16MHz	8 – 16 pF	100 Ω	7.0pF	±60ppm

Table 15 Crystal specification of the nRF24L01

To achieve a crystal oscillator solution with low power consumption and fast start-up time, it is recommended to specify the crystal with a low value of crystal load capacitance. Specifying a lower value of crystal parallel equivalent capacitance, C₀ will also work, but this can increase the price of the crystal itself. Typically C₀=1.5pF at a crystal specified for C_{0max}=7.0pF.

The crystal load capacitance, C_L, is given by:

$$C_L = \frac{C_1 \cdot C_2'}{C_1' + C_2'}, \text{ where } C_1' = C_1 + C_{PCB1} + C_{\Pi} \text{ and } C_2' = C_2 + C_{PCB2} + C_{\Pi}$$

C₁ and C₂ are SMD capacitors as shown in the application schematics. C_{PCB1} and C_{PCB2} are the layout parasitic on the circuit board. C_Π and C_Π are the capacitance seen into the XC1 and XC2 pin respectively; the value is typical 1pF.


nRF24L01 Single Chip 2.4 GHz Radio Transceiver

nRF24L01 sharing crystal with a micro controller.

When using a micro controller to drive the crystal reference input XC1 of the nRF24L01 transceiver some rules must be followed.

Crystal parameters:

When the micro controller drives the nRF24L01 clock input, the requirement of load capacitance C_L is set by the micro controller only. The frequency accuracy of ± 60 ppm is still required to get a functional radio link. The nRF24L01 will load the crystal by 0.5pF at XC1 in addition to the PBC routing.

Input crystal amplitude & Current consumption

The input signal should not have amplitudes exceeding any rail voltage, but any DC-voltage within this is OK. Exceeding rail voltage will excite the ESD structure and the radio performance is degraded below specification. If testing the nRF24L01 with a RF source with no DC offset as the reference source, the input signal will go below the ground level, which is not acceptable.

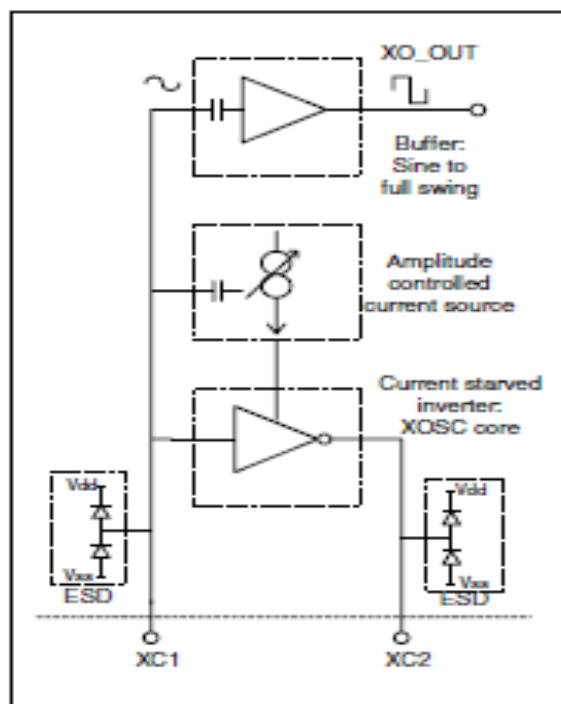


Figure 12 Principle of crystal oscillator

The nRF24L01 crystal oscillator is amplitude regulated. To achieve low current consumption and also good signal-to-noise ratio when using an external clock, it is recommended to use an input signal larger than 0.4 V-peak. When clocked externally, XC2 is not used and can be left as an open pin.



PCB layout and de-coupling guidelines

A well-designed PCB is necessary to achieve good RF performance. Keep in mind that a poor layout may lead to loss of performance, or even functionality, if due care is not taken. A fully qualified RF-layout for the nRF24L01 and its surrounding components, including matching networks, can be downloaded from www.nordicsemi.no.

A PCB with a minimum of two layers including a ground plane is recommended for optimum performance. The nRF24L01 DC supply voltage should be de-coupled as close as possible to the VDD pins with high performance RF capacitors, see Table 16. It is preferable to mount a large surface mount capacitor (e.g. 4.7 μ F tantalum) in parallel with the smaller value capacitors. The nRF24L01 supply voltage should be filtered and routed separately from the supply voltages of any digital circuitry.

Long power supply lines on the PCB should be avoided. All device grounds, VDD connections and VDD bypass capacitors must be connected as close as possible to the nRF24L01 IC. For a PCB with a topside RF ground plane, the VSS pins should be connected directly to the ground plane. For a PCB with a bottom ground plane, the best technique is to have via holes as close as possible to the VSS pads. At least one via hole should be used for each VSS pin.

Full swing digital data or control signals should not be routed close to the crystal or the power supply lines.



nRF24L01 Single Chip 2.4 GHz Radio Transceiver

APPLICATION EXAMPLE

nRF24L01 with single ended matching network crystal, bias resistor, and decoupling capacitors.

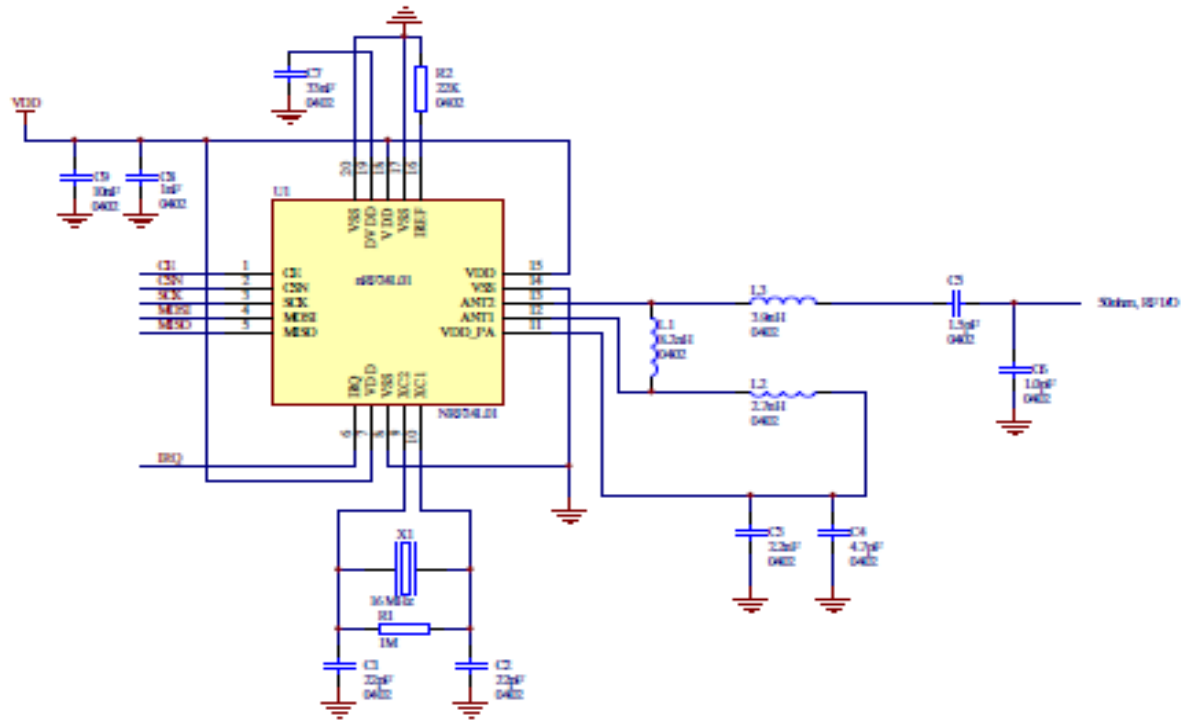


Figure 13 nRF24L01 schematic for RF layouts with single ended 50Ω RF output.

Part	Designator	Footprint	Description
22pF ¹⁷	C1	0402	NPO, +/- 2%, 50V
22pF ¹⁷	C2	0402	NPO, +/- 2%, 50V
2.2nF	C3	0402	X7R, +/- 10%, 50V
4.7pF	C4	0402	NPO, +/- 0.25 pF, 50V
1.5pF	C5	0402	NPO, +/- 0.1 pF, 50V
1,0pF	C6	0402	NPO, +/- 0.1 pF, 50V
33nF	C7	0402	X7R, +/- 10%, 50V
1nF	C8	0402	X7R, +/- 10%, 50V
10nF	C9	0402	X7R, +/- 10%, 50V
8,2nH	L1	0402	chip inductor +/- 5%
2.7nH	L2	0402	chip inductor +/- 5%
3,9nH	L3	0402	chip inductor +/- 5%
1M	R1	0402	+/-10%
22K	R2	0402	+/- 1 %
nRF24L01	U1	QFN20 4x4	
16MHz	X1		+/-60ppm, C _L =12pF ¹⁷

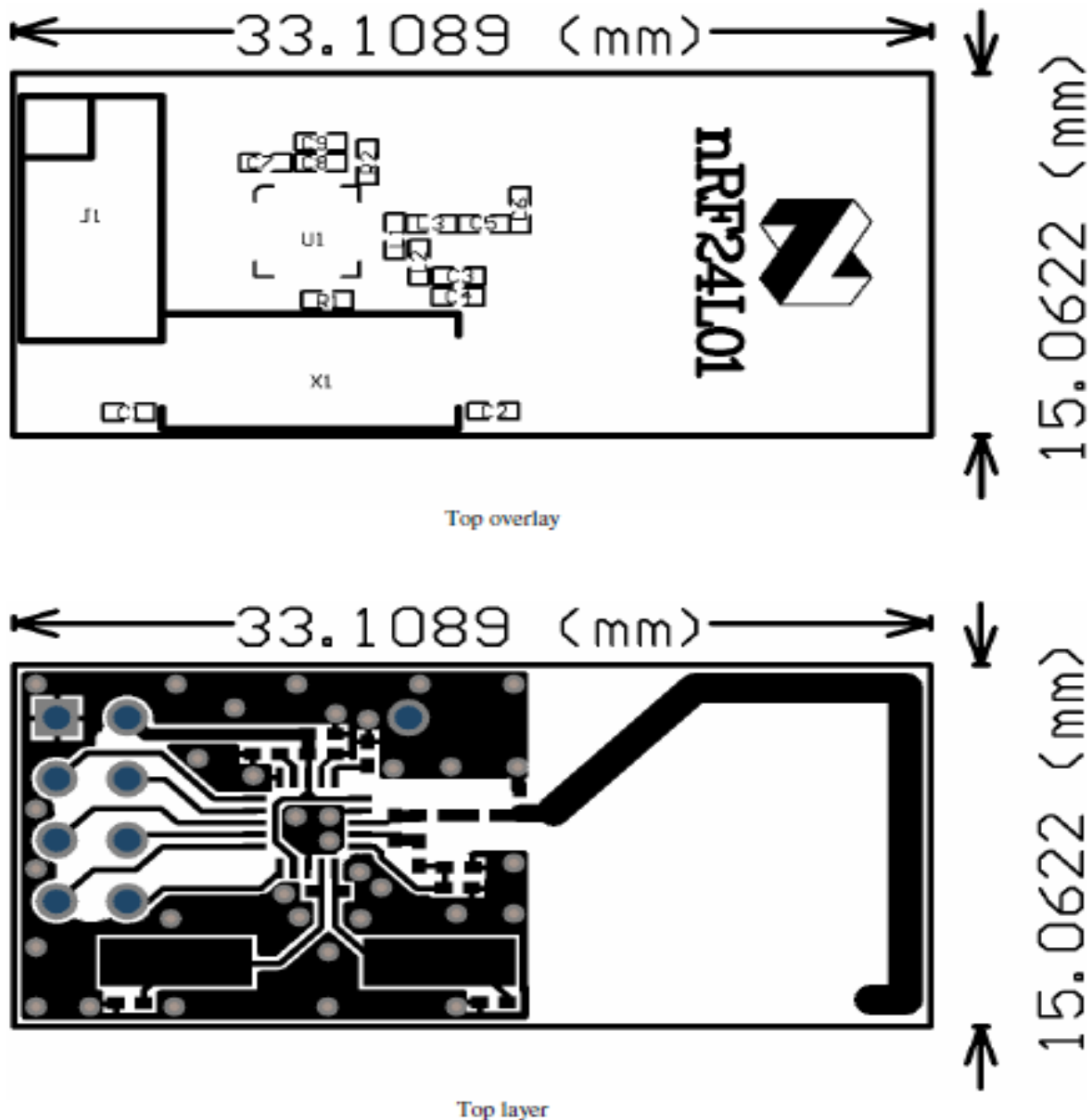
Table 16 Recommended components (BOM) in nRF24L01 with antenna matching network

¹⁷ C1 and C2 must have values that match the crystals load capacitance, C_L.



PCB layout examples

Figure 14 shows a PCB layout example for the application schematic in Figure 13. A double-sided FR-4 board of 1.6mm thickness is used. This PCB has a ground plane on the bottom layer. Additionally, there are ground areas on the component side of the board to ensure sufficient grounding of critical components. A large number of via holes connect the top layer ground areas to the bottom layer ground plane.



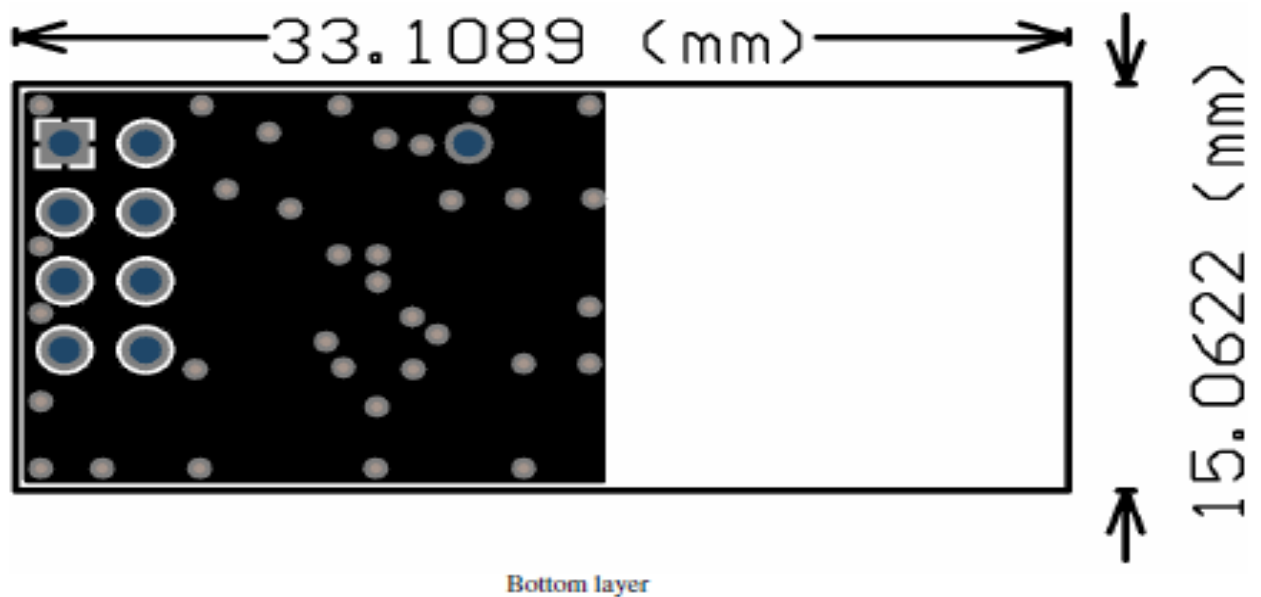
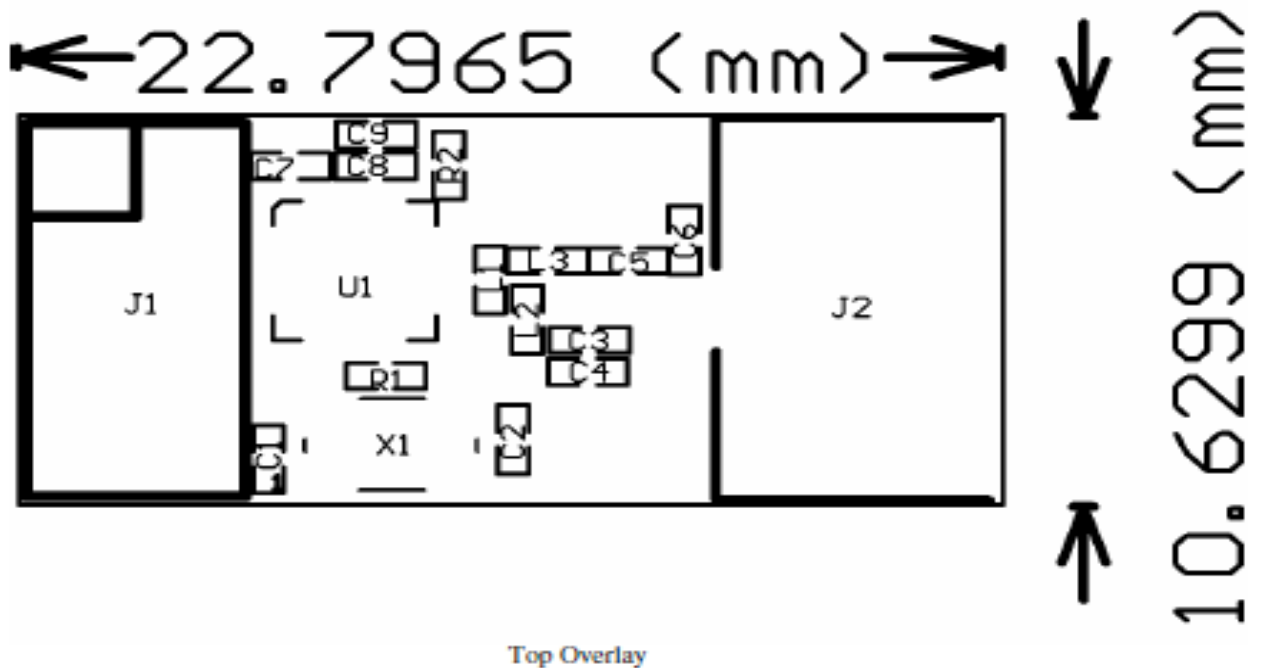


Figure 14 nRF24L01 RF layout with single ended connection to PCB antenna and 0603 size passive components

The next figure (Figure 15) is for the SMA output to have a board for direct measurements at a 50Ω SMA connector.



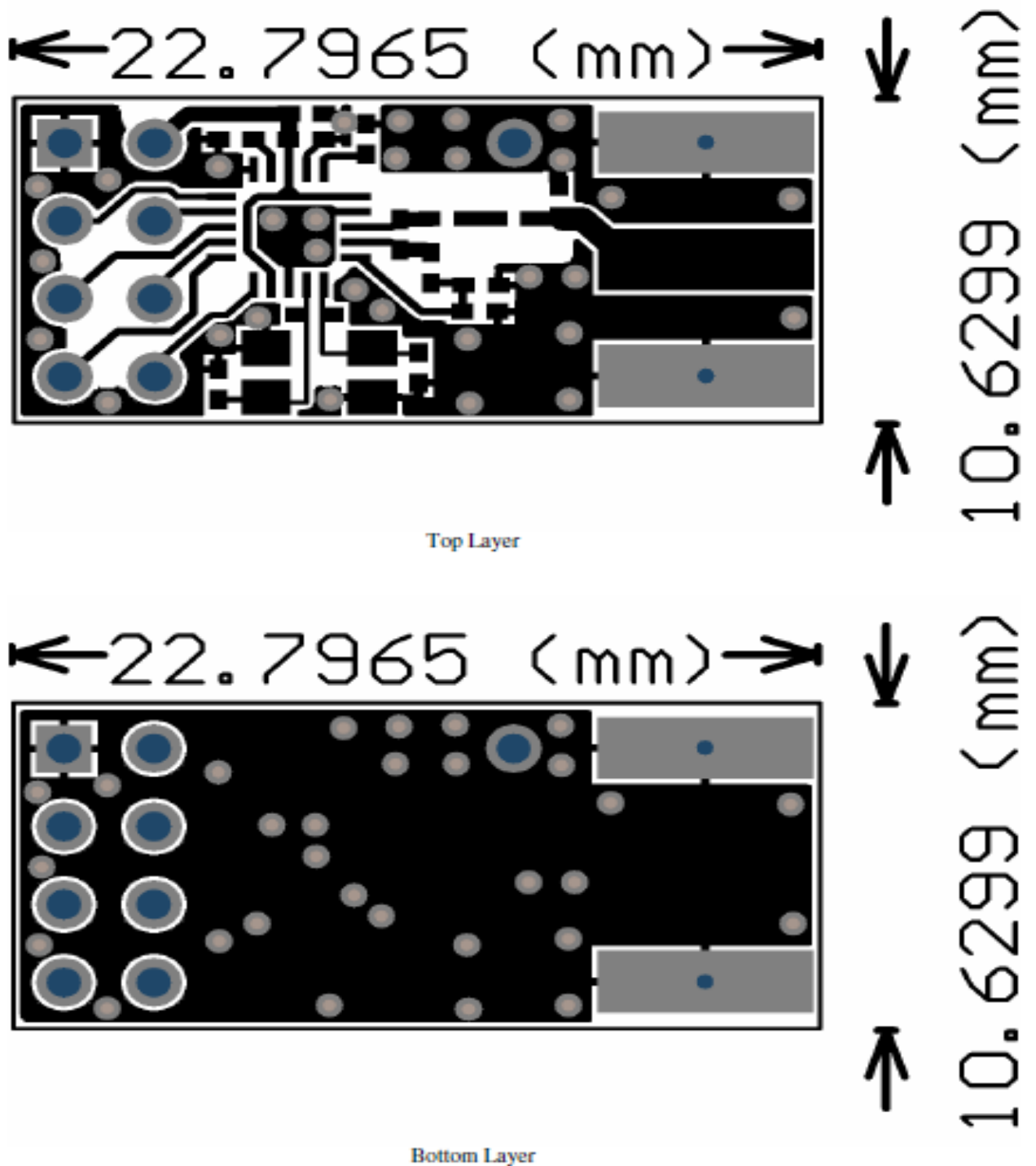


Figure 15 Module with OFM crystal and SMA connector



DEFINITIONS

Data sheet status	
Objective product specification	This data sheet contains target specifications for product development.
Preliminary product specification	This data sheet contains preliminary data; supplementary data may be published from Nordic Semiconductor ASA later.
Product specification	This data sheet contains final product specifications. Nordic Semiconductor ASA reserves the right to make changes at any time without notice in order to improve design and supply the best possible product.
Limiting values	
Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Specifications sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.	
Application information	
Where application information is given, it is advisory and does not form part of the specification.	

Table 17. Definitions

Nordic Semiconductor ASA reserves the right to make changes without further notice to the product to improve reliability, function or design. Nordic Semiconductor ASA does not assume any liability arising out of the application or use of any product or circuits described herein.

LIFE SUPPORT APPLICATIONS

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Nordic Semiconductor ASA customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nordic Semiconductor ASA for any damages resulting from such improper use or sale.

Preliminary Product Specification: Revision Date: 08.03.2006.

Data sheet order code: 080306-nRF24L01

All rights reserved ©. Reproduction in whole or in part is prohibited without the prior written permission of the copyright holder.



Nordic Semiconductor ASA – World Wide Distributors

For Your nearest dealer, please see <http://www.nordicsemi.no>



Main Office:

Vestre Rosten 81, N-7075 Tiller, Norway
Phone: +47 72 89 89 00, Fax: +47 72 89 89 89

Visit the Nordic Semiconductor ASA website at <http://www.nordicsemi.no>

