

UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA



Informe de Memoria de Título para Optar al Título de:
Ingeniero Civil Electrónico

Sistema de presencia académica a través de sensor
de movimiento y control de asistencia a través de
huellas dactilares usando un microcontrolador.

Felipe Alfonso Rogers Sufán

Ph.D. Alejandro Rojas Norman
Ph.D. Mario Medina Carrasco
Ph.D. Sergio Sobarzo Guzmán

Concepción, marzo de 2018

Tabla de contenidos

ABREVIACIONES	IV
1. RESUMEN	1
2. REVISIÓN BIBLIOGRÁFICA	2
2.1. INTRODUCCIÓN	2
2.2. TRABAJOS PREVIOS	2
2.2.1 <i>Sistemas digitales</i>	2
2.2.2 <i>Proyectos relacionados</i>	3
2.2.3 <i>Softwares y Manuales</i>	5
2.3. DISCUSIÓN	10
3. DEFINICIÓN DEL PROBLEMA	11
3.1. HIPÓTESIS	11
3.2. OBJETIVOS	11
3.2.1 <i>Objetivo general</i>	11
3.2.2 <i>Objetivos específicos</i>	11
3.3. ALCANCES Y LIMITACIONES.....	11
3.4. METODOLOGÍA.....	12
4. ESPECIFICACIONES TÉCNICAS DE LOS EQUIPOS	13
4.1. ARDUINO UNO MODELO R3	13
4.2. SENSOR BIOMÉTRICO DE HUELLA DIGITAL.....	14
4.3. MÓDULO BLUETOOTH HC-06	15
4.4. MÓDULO WI-FI ESP8266	16
4.5. SENSOR INFRARROJO DE MOVIMIENTO PIR HC-SR501	17
4.6. MÓDULO RELOJ RTC DS3231	18
4.7. ADAPTADOR TARJETA MICRO SD CATALEX v1.0	19
4.8. SOFTWARE ARDUINO	20
4.9. COMUNICACIÓN SERIE	21
5. SISTEMA DE PRESENCIA ACADÉMICA UTILIZANDO SENSOR DE MOVIMIENTO PIR HC-SR501	22
5.1. INTRODUCCIÓN	22
5.2. SENSORES PIR	23
5.3. CONEXIÓN MICROCONTROLADOR – SENSOR PIR HC-SR501	25
5.4. CONEXIÓN MICROCONTROLADOR – MÓDULO WIFI ESP-8266	25
5.5. SERVIDOR WEB.....	29
5.6. COMUNICACIÓN MICROCONTROLADOR – SERVIDOR WEB.....	31
6. SISTEMA DE REGISTRO DE ASISTENCIA A CLASES UTILIZANDO LECTOR DE HUELLA DACTILAR	34
6.1. INTRODUCCIÓN	34
6.2. SISTEMAS BIOMÉTRICOS	35
6.2.1 <i>Arquitectura de los sistemas biométricos</i>	36
6.3. HUELLA DACTILAR.....	38
6.3.1 <i>Introducción</i>	38
6.3.2 <i>Procesamiento de la huella digital</i>	39
6.4. LECTORES DE HUELLA DACTILAR.....	39
6.5. CONEXIÓN MICROCONTROLADOR – SENSOR DACTILAR.....	40
6.6. BASE DE DATOS DEL SENSOR DE HUELLA DACTILAR	40
6.7. RECONOCIMIENTO DE HUELLA Y ESCRITURA EN MEMORIA MICRO SD (DATALOGGER).....	42
7. RESULTADOS	45

8. CONCLUSIONES.....	52
BIBLIOGRAFÍA.....	54
ANEXO A. TABLAS COMPLEMENTARIAS	57
ANEXO B. TRABAJO COMPLEMENTARIO	58
B.1. CONEXIÓN MICROCONTROLADOR – MÓDULO BLUETOOTH HC-06	58
ANEXO C. DIAGRAMAS.....	59
C.1. DIAGRAMAS DE CONEXIÓN	59
ANEXO D. CÓDIGO	62
D.1. RUTINA CONEXIÓN ARDUINO – MÓDULO HC-06.....	62
D.2. RUTINA CONEXIÓN ARDUINO – MÓDULO ESP8266-01.....	63
D.3. RUTINA CONEXIÓN ARDUINO – LECTOR HUELLAS DACTILARES (ENROLAR)	64
D.4. RUTINA CONEXIÓN ARDUINO – LECTOR HUELLAS DACTILARES (DATALOGGER).....	68
D.5. RUTINA CONEXIÓN ARDUINO – SENSOR INFRARROJO PIR HC-SR501	73
D.6. RUTINA DEL CUERPO PRINCIPAL DE LA PÁGINA WEB.....	74
D.7. RUTINA DE CONDICIONES DE LA PÁGINA WEB.....	75
D.8. RUTINA COMUNICACIÓN MICROCONTROLADOR – SERVIDOR WEB	76
ANEXO E. DATASHEETS	80
E.1. ARDUINO UNO R3	80
E.2. MÓDULO WI-FI ESP8266-01	82
E.3. MÓDULO BLUETOOTH HC-06.....	87
E.4. SENSOR DE MOVIMIENTO PIR HC-SR501	91
E.5. SENSOR DE HUELLA DACTILAR - ADAFRUIT.....	95
E.6. RELOJ RTC DS3231	98



Abreviaciones

Mayúsculas

WI-FI	: fidelidad inalámbrica (Wireless fidelity).
S.A	: sociedad anónima.
SoC	: sistema en Chip (System on Chip).
HD	: alta definición (High definition).
RAM	: memoria de acceso aleatorio (Random Access Memory).
SD	: seguro digital (Secure Digital).
USB	: bus serie universal (Universal Serial Bus).
RJ-45	: conector Jack 45 registrado (Registered Jack 45).
GPIO	: entrada/salida de propósito general (General Purpose Input/Output).
SMSC	: central de servicio de mensajes cortos (Short Message Service Center).
HDMI	: interfaz multimedia de alta definición (High Definition Multimedia Interface).
RCA	: corporación de radio de américa (Radio Corporation of America).
MHz	: mega heartz.
GB	: giga bytes.
V	: volts.
A	: amperes.
Mbps	: mega bytes por segundos.
LED	: diodo emisor de luz. (Light Emitting Diode).
ROM	: memoria de sólo lectura (Read Only Memory).
EEPROM	: ROM programable y borrable eléctricamente (Electrically Erasable Programmable).
SRAM	: memoria estática de acceso aleatorio (Static Random Access Memory).
API	: interfaz de programación de aplicaciones (Application Programming Interface).
SDK	: kit de desarrollo de software (Software Development Kit).
IOT	: internet de las cosas (Internet Of Things).
IP	: protocolo de internet (Internet Protocol).
PCB	: placa de circuito impreso (Printed Circuit Board).
DIY	: hágallo usted mismo (Do It Yourself).
MIT	: instituto tecnológico de Massachusetts (Massachusetts Institute of Technology).
ISM	: industrial, científica y médica (Industrial, Scientific and Medical).
TxD	: transmitir datos (Transmit Data).
RxD	: recibir datos (Receive Data).
SCL	: línea de reloj en serie (Serial Clock Line).
SDA	: línea de datos en serie (Serial Data).
AREF	: referencia analógica (Analog Reference).
IOREF	: referencia de entrada y salida (Input/Output Reference).
WPAN	: red inalámbrica de área personal (Wireless Personal Area Network).
IDE	: entorno de desarrollo integrado (Integrated Development Environment).
GND	: filtro graduado de densidad neutra (Graduated Neutral Density).
OSI	: modelo de interconexión de sistemas abiertos (Open System Interconnection).
DSP	: procesador digital de señales (Digital Signal Processor).
PIR	: sensor infrarrojo pasivo (Passive Infrared).
TTL	: lógica de transistor a transistor (Transistor-Transistor Logic).
PLA	: ácido poli láctico (Poly lactic Acid).

Minúsculas

mm : mili metros.
mW : mili watts.
mA : mili amperes.
cm : centi metros.



1. Resumen

La problemática a resolver radica en la necesidad de los alumnos de la Universidad de Concepción de conocer la disponibilidad que tiene un profesor en un instante cualquiera. A pesar de que los profesores tienen horarios definidos por las clases que deben realizar en sus correspondientes asignaturas, fuera de estos horarios ya establecidos, los profesores no necesariamente se encuentran en sus oficinas o dentro de la Universidad, dificultando así la posibilidad de que los alumnos se puedan entrevistar con ellos según sus requerimientos. Es por esta problemática que nace la idea de implementar un sistema de presencia académica utilizando un microcontrolador conectado a un sensor de movimiento PIR HC-SR501 que cumpla la función de avisar vía Internet, por medio de la conexión de éste con un módulo WI-FI ESP8266, la presencia o ausencia de los docentes en sus respectivas oficinas a cada uno de los alumnos que lo requiera, lo que es complementado con el denominado tablero de presencia académica, el que informa a través de luces LED que el mismo docente manipula manualmente, su disponibilidad inmediata o futura, lo cual también es informado a través de una página web. Una vez implementado el sistema anteriormente descrito, los estudiantes pueden tener acceso a la disponibilidad, presencia o ausencia de un profesor en particular en un momento determinado, lo que se traduce en otorgar información relevante para que los alumnos organicen de mejor manera su tiempo.

En otra esfera del ámbito académico, se encuentra una problemática a resolver consistente en la modernización de los sistemas de control de asistencia a clases de los alumnos de la Universidad. El objetivo en este ámbito es evitar la pérdida de tiempo que conlleva la consulta individual de asistencia a clases realizada por el docente a cada alumno. Con el fin de resolver esta segunda problemática se implementó un sistema de control de asistencia utilizando un microcontrolador conectado a un lector de huellas dactilares que se encarga de leer las huellas a cada alumno y dejar el respectivo registro de asistencia. Para implementar dicho sistema se utilizaron los módulos de lector de tarjeta micro SD Catalex y de reloj RTC DS3231, ambos conectados al microcontrolador Arduino. Dichos dispositivos tienen por objeto guardar el registro de la huella de cada usuario y darle la capacidad al Arduino de obtener mediciones de tiempo en unidades temporales respectivamente. Luego de implementar el sistema propuesto se logra llevar un registro de asistencia automatizado, confiable y eficiente en relación al tiempo de registro del mismo, dándole más tiempo de cátedra al profesor, lo cual beneficia directamente a los alumnos.

2. Revisión bibliográfica

2.1. Introducción

La revisión bibliográfica permitirá la introducción y ambientación del tema de los sistemas digitales y automatización. Además, se obtendrá información relevante de las hojas de datos de los equipos utilizados.

2.2. Trabajos previos

Los trabajos previos a considerar se centran en los sistemas digitales propiamente tal, en proyectos relacionados, softwares necesarios y manuales u hojas de datos de los equipos a utilizar.

2.2.1 Sistemas digitales

- ♣ Thomas L. Floyd. “Fundamentos de Sistemas Digitales” Vol. 9, Pearson Educación S.A, Madrid, 2006.

En el documento se abordan temas básicos de la tecnología digital que son imprescindibles en miras al desarrollo de una carrera o algún proyecto relacionado a esta industria. Los primeros capítulos del documento contienen mayormente información básica y necesaria para introducirse en el ámbito de la programación de microprocesadores, se abordan temas relacionados con las operaciones lógicas básicas, funciones lógicas básicas e introduce el tema de la lógica programable. Además cuenta con una sección denominada “Aplicación a los sistemas digitales”, la cual se incluye al final de la mayoría de los capítulos del documento, en donde se presentan aplicaciones prácticas sobre los conceptos y dispositivos cubiertos en el capítulo respectivo.

La importancia del texto para el proyecto radica en la introducción en un ámbito general a la lógica programable, tratando temas como las computadoras, microprocesadores o el proceso digital de la señal. También contribuye a los fines del proyecto abordando temas relacionados con las operaciones y funciones lógicas. Se introduce el tema de la lógica programable de lleno, describiendo los distintos tipos de dispositivos lógicos programables y sus diferencias, lo que es fundamental para conocer y saber diferenciar los dispositivos disponibles para la implementación del proyecto y así poder tomar una buena decisión con respecto a cuál elegir.

- ♣ Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss. “Sistemas Digitales Principios y Aplicaciones” 10º Edición, 2007.

En este documento el autor busca abordar un estudio completo sobre los principios y las técnicas de los sistemas digitales modernos. Enseña los principios fundamentales sobre los sistemas digitales y cubre con amplitud los métodos tradicional y moderno para aplicar las técnicas de diseño y desarrollo digital, incluyendo como administrar un proyecto a nivel de sistemas. Menciona además, que aunque sería de gran utilidad tener conocimientos sobre electrónica básica, la mayor parte del material no lo requiere. Otra de las mejoras con respecto a otras ediciones es que en ésta se estudian en detalle los circuitos digitales simples, reforzando conceptos tales como las entradas y salidas binarias, la operación de un dispositivo físico y las limitaciones prácticas.

El objetivo que ayuda a cumplir el texto dentro del proyecto que se pretende realizar, dice relación con conocer en profundidad el mundo de los sistemas digitales, diferenciando entre la representación analógica, híbrida y la digital, ya que el texto menciona las ventajas, desventajas y diferencias de cada uno de estos sistemas. El documento también cuenta con un capítulo dedicado a los dispositivos lógicos programables y sus aplicaciones, en donde se permite comenzar a entender la arquitectura del hardware de los dispositivos típicos de lógica programable, usar instrumentos de software que permitan la programación de una gran variedad de estos dispositivos e implantar un diseño lógico utilizando dispositivos lógicos programables.

2.2.2 Proyectos relacionados

- ♣ Diego Gudiño C. “Diseño y construcción de un sistema de control de luces para una casa utilizando la placa Raspberry”. Sistemas Informáticos, Universidad Tecnológica Israel, Ecuador, 2014.

En este trabajo se realiza la implementación de un sistema de control de luces en una casa mediante la mini computadora Raspberry Pi utilizando un lenguaje de cuarta generación y además se elabora una interfaz de programación para controlar mediante software el control de luces. El objetivo principal que tiene el documento es presentar el concepto de hogar digital. Para lo anterior, se comienza dando una pincelada a la situación del mercado para luego describir las tecnologías, su integración y las implicancias para el equipamiento tradicional de la vivienda.

La importancia del documento para el proyecto radica principalmente en el énfasis que se le da al concepto de la domótica, tanto como en su historia, definición y los distintos tipos en los que se puede encontrar, debido a que el proyecto está totalmente ligado a este concepto.

- ♣ Emilio Lledó S. “Diseño de un sistema de control domótico basado en la plataforma Arduino”. Ingeniería Técnica en Informática de Sistemas. Universidad Politécnica de Valencia, Valencia, España, diciembre 2012.

En el documento se abarcan temas de domótica y se realiza una descripción del microcontrolador Arduino, además de dedicar una sección a dar a conocer el por qué se utiliza este microprocesador y no otro.

Las ventajas de contar con este documento corresponden al conocimiento previo de cómo se comporta y como se configura un microcontrolador de este tipo en aplicaciones de automatización como la que se busca implementar en este proyecto, ya que el documento contempla una sección denominada “prototipo de servidor domótico” en el cual se describe el montaje y el funcionamiento del microcontrolador como servidor web.

- ♣ Juan Almansa M. “Desarrollo e implementación utilizando Arduino y Zigbee con un sensor ultrasónico para control de nivel de llenado”. Departament d’Enginyeria Electrònica Elèctrica i Automàtica. Universitat Rovira i Virgili. Tarragona, España, septiembre 2013.

El documento consiste en el desarrollo e implementación de un sensor ultrasónico para el control de nivel de llenado utilizando el microcontrolador Arduino, en el que se detalla la tecnología empleada para la construcción de un prototipo del sistema, con comparaciones y pruebas realizadas. En un principio el documento se centra en describir los dos grandes bloques que lo componen, como son Arduino y Zigbee, se realiza un estudio técnico sobre las características respectivas, así como el funcionamiento en el entorno de programación y la configuración de los mismos. Luego se procede a describir el hardware y software utilizado en todas las frases del proyecto y en la implementación. Finalmente se explican una serie de casos experimentales, como la lectura del consumo en diferentes ciclos de trabajo, lecturas de señales de potencia recibida y distancias de comunicación inalámbrica y se adjuntan planos y fotografías de las partes fundamentales del proyecto a modo de resumen.

El aporte que significa este documento para el proyecto, tiene que ver con el énfasis que el autor le da al lenguaje de programación y el entorno de desarrollo del dispositivo, contemplando conceptos básicos relacionados como, por ejemplo, estructuras, variables, funciones, y uso de distintas librerías disponibles. Lo anterior implicará una gran ayuda en el diseño del proyecto que se requiere

implementar. El documento además cuenta con una sección denominada “Introducción a Arduino”, en la cual se describe el microcontrolador y se analizan las ventajas que implica la utilización de éste dispositivo en comparación a otro.

- ♣ Javier Colomer B. “Aplicación Android para reproducción de audio en plataforma Arduino”. Escuela Técnica Superior de Ingenieros de Telecomunicación. Universidad Politécnica de Valencia, Valencia, España, julio 2015.

El documento corresponde a un trabajo final de grado que consiste en la creación de un prototipo de un reproductor de audio mediante la plataforma Arduino que está controlado por una aplicación de sistema operativo Android desde un dispositivo móvil. En el texto se integran los campos de la programación, el de la electrónica y el de las telecomunicaciones. El control que se pretende implementar es a la reproducción de audio almacenado en la tarjeta micro SD que posee el Shield de Arduino y la comunicación que se implementa entre ambos dispositivos se establece mediante Bluetooth Low Energy.

El trabajo realizado por el autor de este documento es importante, puesto que en él, se diseñan conexiones y comunicaciones muy similares a la que se pretende realizar en este proyecto, como son las conexiones vía Bluetooth entre un dispositivo móvil y un microcontrolador Arduino y la conexión del microcontrolador con un dispositivo lector de tarjeta micro SD.

Además, el documento sirve de complemento a las literaturas anteriores para conocer en profundidad el microcontrolador Arduino, ya que el capítulo 6 de éste, está destinado a describir y definir el microcontrolador.

2.2.3 Softwares y Manuales

- ♣ Eben Upton and Garth Halfacree. “Raspberry Pi User guide”, John Wiley & Sons Ltd., The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom, 2012.

Este documento corresponde a la guía del usuario de la Raspberry Pi, elaborada por uno de los creadores del dispositivo. En el documento se elabora una detallada descripción de la Raspberry Pi en general (para todas las versiones). Consta de dos partes, la primera se denomina “Conexión de la Placa” y corresponde a la descripción del dispositivo propiamente tal. La segunda parte del documento se denomina “Utilizando la RasPi como un centro Multimedia, Máquina Productiva y Servidor Web” y corresponde a una guía de cómo utilizar la Raspberry Pi en diversas aplicaciones o funciones.

En los primeros capítulos, particularmente el 1 y 3, el documento se centra en la descripción del hardware del dispositivo, es decir, en las conexiones y solución de problemas de cada uno de los elementos externos al equipo. En los demás capítulos, el documento se centra en el software del equipo, en las configuraciones y aplicaciones de éste.

La importancia de este último documento, para el proyecto, radica en la posibilidad, como trabajo futuro o complementario, de utilizar la Raspberry Pi como servidor para el sistema de asistencia utilizando sensor dactilar.

- ♣ Isabel Gracia, Andrés Marzal. “Introducción a la programación con Python”. Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I, 2003.

Este documento tiene por objetivo permitir a los lectores aprender a programar en lenguaje Python y lenguaje C. Se diferencia de otros textos en el hecho de que permite estudiar dos lenguajes y en la forma en que se exponen y desarrollan los conocimientos de los autores. Lo que hace que además de estudiar los lenguajes, el lector sea capaz de compararlos y obtener ventajas y desventajas de ambos.

La importancia que tienes este documento para el proyecto radica en que el lenguaje a utilizar para la programación en Raspberry Pi es lenguaje Python y el lenguaje a utilizar para la programación del microcontrolador Arduino está basado y es muy similar al lenguaje C.

- ♣ Guido Van Rossum. “The Python Tutorial”. Director of PythonLabs at Zope Corporation, Febrero 2017. Disponible en: <http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>. (Last Access 05/05/2017).

El texto corresponde a una guía de cómo utilizar el software Python, el que corresponde a un intérprete del lenguaje con el mismo nombre. El autor de este documento es el creador del lenguaje Python y según sus propias palabras, se introduce de manera informal al lector, a los conceptos y características básicas del lenguaje y el sistema de Python. Sin embargo, para el que requiera una mirada más formal del lenguaje, se cuenta también con un apartado denominado “La referencia de la Biblioteca”.

En el texto introduce la mayoría de las características más notables de Python a través de ejemplo, empezando con expresiones, instrucciones y tipos de datos simples, pasando por funciones y módulos, y finalmente tocando conceptos avanzados como excepciones y clases definidas por el usuario.

Del documento, se obtienen conocimientos a través de los distintos ejemplos y aplicaciones que se describen en éste. La ventaja que se obtiene del documento es, como su autor lo dice, el lenguaje informal y básico que se emplea, ya que además de contemplar desde lo más básico hasta lo más complejo de este lenguaje de programación, se hace con simpleza y a través de ejemplos, con el fin de ser más amigable al lector. Cabe destacar que este documento hace referencia a la versión más reciente del Software Python elaborada por el autor, lo que implica una ventaja al momento de utilizarlo en el proyecto.

- ♣ Óscar Torrente A. “Arduino, curso práctico de formación”. Alfaomega Grupo Editor, México, febrero 2013.

El documento corresponde a una guía completa del microcontrolador Arduino, en donde se describe en su totalidad el dispositivo, tanto en composición como en funcionamiento. Sin embargo, el autor no se limita solo a la descripción del dispositivo propiamente tal, ya que en el texto se abordan temas con relación a la programación y a la electrónica en sus aspectos más básicos posibles, con el fin de incluir a lectores con nulos conocimientos de estos temas. Se incluyen conceptos teóricos sobre electricidad, tales como voltaje, corriente, ley de Ohm, entre otros. Además de describir los componentes eléctricos básicos relacionados a la electrónica como son las resistencias, potenciómetros, diodos y LEDs, etc. Luego de la introducción al ámbito de la electricidad y electrónica, se comienza a describir el Arduino a nivel de Hardware, iniciando con una descripción a los que son los sistemas electrónicos y microcontroladores en general. El texto se ha escrito, a palabras del autor, con el objetivo de facilitar al lector autodidacta una asimilación gradual de los conceptos y procedimientos necesarios para ir avanzando de a poco y con seguridad a lo largo de los diferentes capítulos.

El principal aporte del documento al proyecto viene dado por permitir conocer en profundidad la estructura y funcionamiento del dispositivo Arduino, para así saber si cumple con los requisitos para la implementación del proyecto. Además de la descripción del hardware del dispositivo, en el texto se describe el Arduino a nivel de Software, se explica detalladamente la instalación de éste y su configuración para distintos sistemas operativos, lo que hace más favorable su utilización.

De los capítulos 4 y 5 del documento, se extraerán los conocimientos necesarios para la programación propiamente tal del dispositivo, tales como, los distintos bloques para la creación de códigos y funciones dentro del lenguaje que el dispositivo utiliza y las distintas librerías que son necesarias.

- ♣ Hacedores – Maker Community. “Arduino o Raspberry Pi, ¿Cuál es la mejor herramienta para ti?”. Mayo, 2014. Disponible en: <http://hacedores.com/arduino-o-raspberry-pi-cual-es-la-mejor-herramienta-para-ti/>. (Last Access 08/05/2017).

El documento corresponde a una comparación técnica y de aplicación entre los microprocesadores Arduino y Raspberry Pi en un ámbito general, es decir, sin importar que versión o modelo sea, la que es orientada a lectores sin mucho conocimientos en el tema con el fin de hacer más fácil la comprensión del texto. Se evidencia que desde el objetivo de creación de los dispositivos existían diferencias, describiendo las respectivas definiciones de los dispositivos, la Raspberry Pi como una computadora completamente funcional y el Arduino como un microcontrolador. El documento además cuenta con tablas comparativas de los modelos a grandes rasgos, posibles utilidades de cada uno y en conjunto. Se hace especial énfasis en la diferencia de ambos equipos, basándose además, en el testimonio de expertos en temas relacionados con la electrónica y los dispositivos.

De este documento se extrae una mejor perspectiva de la comparación entre los microprocesadores a considerar en la realización del proyecto, con el fin de obtener la mayor información posible de éstos y así poder tomar una decisión segura de cuál es el más adecuado para cumplir dicha función. Lo que más aporta para fines del proyecto, es la comparación de los dispositivos que se realiza en el texto a nivel de las aplicaciones, ya que en éste, se describen las distintas aplicaciones para la que son mejores alternativas cada uno de los equipos en cuestión. Lo anterior se hace muy importante, ya que no basta con saber que dispositivo es mejor a nivel general, sino que se necesita conocer cual brinda mayores ventajas con respecto al otro a la hora de implementar el sistema.

- ♣ Arduino. “Arduino Uno Datasheet”. Disponible en: <http://arduino.cl/arduino-uno/>. (Last Access 08/05/2017).

El documento corresponde a la hoja de especificaciones del microcontrolador Arduino Uno, en todos sus modelos, tanto como el Arduino Uno R3, Arduino Uno R2 y el Arduino Uno SMD.

Del documento se obtienen las especificaciones del modelo Arduino Uno R3, sus características físicas, es decir, las medidas, los niveles de voltaje en los que puede o es recomendable que opere, las características de sus memorias internas, las características de los pines, tanto de entrada como de salida y algunas recomendaciones y características que se deben conocer con respecto a su comunicación y programación.

- ♣ Bluetooth Serial HC-06. “Product DataSheet”. Guangzhou HC Information Technology Co., Ltd. Disponible en: <https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf>. (Last Access 06/06/2017).

El documento corresponde a la hoja de especificaciones del módulo bluetooth HC-06 utilizado para brindarle la posibilidad al microprocesador de comunicarse vía Bluetooth con otros dispositivos.

Del documento se extraen las imágenes referenciales del producto, dimensiones, características principales, los parámetros y modos del producto, diagrama en bloques, las características de prueba, el diagrama de prueba, el conjunto de comandos, entre otros.

- ♣ ESP8266 Serial Esp-01 WIFI Wireless. “ESP8266-01 datasheet”.

El documento corresponde a la hoja de especificaciones del módulo WI-FI ESP8266-01 utilizado para brindarle la posibilidad al microprocesador de comunicarse vía WI-FI con otros dispositivos.

Del texto se pueden extraer imágenes referenciales del producto, las características principales de operación, características generales del hardware, características de memoria y CPU, características de las patillas de entradas y salidas digitales y sus características de operación, la potencia disipada en operación y en standby, entre otras características.

- ♣ Adafruit. “Adafruit Optical Fingerprint Sensor”. Disponible en: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-optical-fingerprint-sensor.pdf>. (Last Access 03/10/2017).

De este documento se extrae toda la información básica acerca de cómo utilizar de manera correcta el sensor dactilar fabricado por la marca Adafruit, con el que se pretende implementar el sistema de control de asistencia.

Dentro de la información que se puede extraer se considera las especificaciones técnicas, las precauciones que se debe tener a la hora de conectarlo, las dimensiones y un pequeño tutorial de cómo utilizarlo tanto a través de lenguajes de programación como a través del software “SFGDemo”, que corresponde a una aplicación que funciona como interfaz gráfica y de comunicación entre el dispositivo y un computador.

- ♣ Adafruit. “PIR Motion Sensor”. Disponible en: <https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>. (Last Access 03/10/2017).

El texto corresponde a la hoja de datos y especificaciones del sensor infrarrojo de movimiento PIR elaborado por la marca Adafruit.

Esta hoja de datos es importante para el proyecto ya que no solo contiene las especificaciones técnicas, sino que describe el sensor en sí, describe el funcionamiento, permite al lector conocer la correcta conexión que se debe utilizar para el debido funcionamiento. Describe además, las distintas configuraciones y sensibilidades de medición que contempla el sensor, utilizando ejemplos de usos.

2.3. Discusión

De la bibliografía analizada anteriormente se extraen, en general, los conocimientos básicos relacionados al mundo de la electrónica y sistemas digitales. Es del análisis de la literatura recabada que se permite comenzar a conocer e introducirse al tema de la programación de los microprocesadores.

Se realizan análisis bibliográficos de los manuales y softwares de los distintos componentes y equipos que son necesarios para desarrollar el proyecto, puesto que será muy importante tener en cuenta las precauciones y consideraciones que se deben tomar a la hora de la puesta en marcha del proyecto con el fin de no dañar los instrumentos.

Se analizan las características tanto de software como de hardware de la mini computadora “Raspberry Pi 2 modelo B” y del microcontrolador Arduino Uno modelo R3, puesto que son los posibles microprocesadores disponibles para la realización de la implementación física del proyecto.

La revisión de proyectos anteriores constituye un relevante aporte en el sentido de que permite conocer en detalle las distintas aplicaciones que se le pueden dar a los microprocesadores disponibles, lo que es de vital importancia a la hora de comparar ambos, para así tomar la decisión con respecto a cuál de los dos escoger para la implementación del proyecto. Por otro lado, se erigen como un importante incentivo, evidenciando que efectivamente es viable realizar proyectos de este tipo.

3. Definición del problema

3.1. Hipótesis

Es posible mejorar el sistema académico de la Universidad de Concepción a través de un sistema de bajo costo basado en un microcontrolador Arduino.

3.2. Objetivos

3.2.1 Objetivo general

Implementar un sistema de presencia académica y otro de control de asistencia, utilizando para ambos un microcontrolador Arduino.

3.2.2 Objetivos específicos

- Conocer las especificaciones técnicas necesarias de los componentes a utilizar.
- Implementar conexión de Arduino con sensor de movimiento HC-SR501.
- Diseñar una página web que cumpla la tarea de informar la presencia o ausencia de los docentes en oficina en tiempo real.
- Implementar conexión de Arduino con lector de huellas dactilares.
- Diseñar tablero de presencia académica, tablero de registro de asistencia a clases e implementar conexión de éstos con Arduino.

3.3. Alcances y limitaciones

El dispositivo utilizado para cumplir la función de controlar el sistema será el microcontrolador Arduino Uno Versión R3.

La importante carga de trabajo que supone el uso del lector de tarjeta micro SD para el microcontrolador se considera una limitación importante, debido a que sólo el programa de escritura en la memoria SD utiliza aproximadamente un 40% de la memoria flash y casi un 50% de la memoria dinámica del Arduino.

Como alcance, es posible mencionar el hecho de que el sistema de manipulación de los diodos LED del tablero es manual a través de interruptores para cada uno, con posibilidad de implementación de la conexión bluetooth para manipulación a distancia. Sin embargo, esta conexión llevaría consigo un pequeño retraso en el encendido y apagado de las luces, debido a que la consulta en el “loop” de la rutina quedaría determinada a la duración del envío de variables hacia el servidor web.

3.4. Metodología

El desarrollo del proyecto se realizará mediante la programación del microcontrolador Arduino Uno modelo R3 a través del software IDE de Arduino, que será de utilidad tanto para la configuración y creación de una rutina que configure el lector de huellas dactilares para la creación de la base de datos y para la manipulación de éste, como para la configuración del sensor de movimiento y la conexión de éste con el microcontrolador y el tablero de presencia académica.

Se utilizará el software Fritzing para llevar a cabo las demostraciones de las correctas configuraciones y conexiones físicas de los distintos módulos y sensores con el microcontrolador.

Para el diseño y posterior impresión en 3D del tablero de presencia académica y tablero de asistencia a través de lectura de huella dactilar se utilizará el software Autodesk Inventor 2018.



4. Especificaciones técnicas de los equipos

4.1. Arduino Uno modelo R3

El Arduino es una plataforma computacional física de fuente abierta basada en una simple tarjeta de I/O y un entorno de desarrollo que implementa el lenguaje Processing/Wiring. El Arduino Uno R3 puede ser utilizado para desarrollar objetos interactivos o puede ser conectado a Software en un computador.

Sus especificaciones son:

- Microcontrolador ATmega 328.
- Voltaje de operación de 5 V.
- Voltaje de entrada recomendada de 7-12 V.
- Límite de voltajes de entrada de 6-20 V.
- 14 pines digitales de I/O (6 salidas PWM).
- 6 entradas análogas.
- Corriente continua por pin de entrada/salida de 40 mA.
- Corriente continua para el pin de 3.3 V de 50 mA.
- 32k de memoria Flash.
- 2 kB de memoria SRAM.
- 1 kB de memoria EEPROM.
- Reloj de 16MHz de velocidad.

La tarjeta Arduino Uno R3 añade pins SDA y SCL en comparación a sus versiones anteriores cercanos al AREF. Además de dos nuevos pines cerca del pin RESET, uno es el IOREF que permite a los “shields” o placas externas adaptarse al voltaje brindado por la tarjeta.



Fig. 4.1 Arduino Uno modelo R3.

4.2. Sensor biométrico de huella digital

El sistema de este sensor realiza procesamiento digital de imágenes interno con un DSP, además, incluye capacidades de comparación en base de datos y actualización de ésta. El dispositivo funciona con el protocolo de comunicación serial, lo que hace más fácil su utilización y conexión con cualquier microcontrolador o tarjeta de desarrollo.

Sus especificaciones son:

- Voltaje de alimentación: 3.6 V – 6 V.
- Corriente de operación: 100 mA – 150 mA.
- Interfaz: TTL Serial.
- Modo de paridad de huella: 1:1 1:N.
- Tasa de baudios: 9600*N.
- Tiempo de adquisición: < 1 seg.
- Dimensión de la ventana: 14 mm x 18 mm.
- Entorno de trabajo: -20 °C – +50 °C.
- Dimensiones: 5.5 cm x 2.1 cm x 2.0 cm.
- Peso: 20 g.



Fig. 4.2 Sensor biométrico de huella digital.

4.3. Módulo bluetooth HC-06

Se escoge el módulo HC-06 debido a que gracias a sus puertos TxD y RxD, permite al equipo Arduino realizar una conexión inalámbricamente a 10 metros de distancia vía Bluetooth. El dispositivo se caracteriza por su fácil manejo y utilización, esto gracias a su SPP de bajo consumo, específicamente 3.3 V, lo cual permite ser alimentado con facilidad por el microcontrolador.

Sus principales características son:

- Funciona con cualquier adaptador Bluetooth USB.
- Velocidad de transmisión por defecto de 9600 bps.
- Construido en la antena.
- Versión Bluetooth 2.0 + EDR.
- Voltaje de funcionamiento de 3.3 V.
- Longitud del cable de 21.5 cm.
- Peso del artículo de 7g.
- Dimensiones del artículo de 3.5 cm x 1.5 cm.

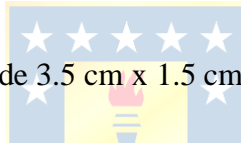


Fig. 4.3 Módulo Bluetooth HC-06.

4.4. Módulo WI-FI ESP8266

El módulo WI-FI ESP8266 es similar en arquitectura que el módulo Bluetooth descrito anteriormente e incluye toda la electrónica necesaria para la comunicación radio frecuencia, en este caso, en la banda WI-FI. Corresponde a un módulo sencillo, una plataforma inteligente de conectividad inalámbrica de alto rendimiento diseñada desde el principio con la IOT, y por eso, incluye todo lo necesario para conectarse a un punto de acceso WI-FI mediante comandos de texto AT, vía puerta serie que puede ser configurada a distintas velocidades. Una vez que se le instruye para que se conecte a la red WI-FI, el módulo es capaz de enviar información que se le remite vía puerta serie a una dirección IP y puerto que se desee.

Sus principales características son:

- Voltaje de operación entre 3 V y 3.6 V.
- Voltaje de operación recomendable de 3.3 V.
- Corriente de operación de pines de 12 mA.
- CPU de 32 bits de baja potencia incorporada.
- 8 patillas de conexión entre alimentación y GPIO.
- Antena impresa en placa PCB
- Dimensiones de 14.30 mm x 24.80 mm.
- Temperatura de operación entre -20 °C y 100 °C.
- Soporta IPv4 y protocolos TCP/UDP/HTTP/FTP.
- Consumo de energía en standby de menos de 1mW.

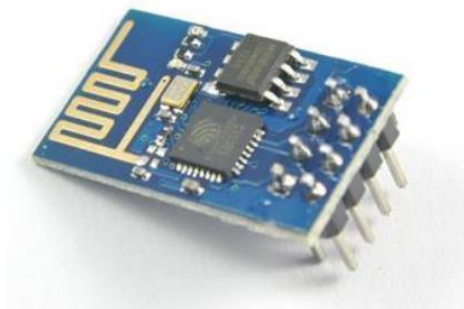


Fig. 4.4 Módulo WI-FI ESP8266.

4.5. Sensor infrarrojo de Movimiento PIR HC-SR501

El sensor PIR consta de dos elementos detectores separados, siendo la señal diferencial entre ambos la que permite activar la alarma de movimiento. En el caso específico del sensor HC-SR501, la señal generada ingresa a un circuito integrado, el cual contiene amplificadores operacionales e interfaces electrónicas adicionales.

Sus especificaciones son:

- Voltaje de alimentación: 5 V- 12 V.
- Consumo promedio: < 1 mA.
- Rango de distancia: 3 m a 7 m. Ajustable mediante potenciómetros.
- Ángulo de detección: 110°.
- Jumper para configuración de salida en modo mono-disparo o disparo repetitivo.
- Salida de alarma de movimiento con ajuste de tiempo entre 3 segundos y 5 minutos.
- Tiempo de inicialización: luego de alimentarlo, demora 1 minuto iniciar su operación normal.
- Tiempo de salida inactiva: Cada vez que la salida pase de activa a inactiva, tiene un tiempo de 3 segundos en donde ignorará cualquier movimiento, para luego empezar a detectar normalmente.
- Temperatura de operación: -15°C a 70°C.
- Dimensiones: 3.2 cm x 2.4 cm x 1.8 cm.



Fig. 4.5 Sensor de movimiento PIR HC-SR501.

4.6. Módulo reloj RTC DS3231

Un reloj de tiempo real es un dispositivo electrónico que permite obtener mediciones de tiempo en las unidades temporales que empleamos de forma cotidiana. El reloj DS3231 tiene como característica principal que lo diferencia de otros, la incorporación de medición y compensación de temperatura, lo que garantiza una precisión de al menos 2ppm, lo que equivale a un desfase máximo de 172 [ms/día] o 1 [s] cada día. La comunicación del reloj con el Arduino se realiza a través del bus I2C.

Sus especificaciones generales son:

- Voltaje de operación: 3.3V.
- Exactitud de $\pm 2\text{ppm}$ operando a una temperatura de entre 0°C a 40°C .
- Módulo cuenta con reloj DS3231 y memoria EEPROM I2C.
- Batería de respaldo.
- Registro de segundos, minutos, horas, día de la semana, fecha, mes y año.
- 2 alarmas programables por fecha y hora.
- Salida de señal cuadrada programable.

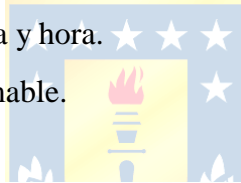


Fig. 4.6 Módulo reloj RTC DS3231.

4.7. Adaptador tarjeta micro SD Catalex v1.0

Un lector micro SD es un dispositivo que permite emplear como almacenamiento una tarjeta micro SD, lo que permite la incorporación de ésta en proyectos de Arduino y electrónica en general.

Una de las desventajas importantes que conlleva este adaptador es que supone una importante carga de trabajo del procesador del microcontrolador Arduino.

Sus especificaciones son:

- Voltaje de alimentación: 5 V.
- Voltaje de operación: 3.3V.
- Agujeros de posicionamiento: 4 tornillos M2 de posicionamiento, diámetro 2.2 mm del agujero.
- Interfaz de control: GND, VCC, MISO, MOSI, SCK, CS.
- Tamaño 45 mm x 28 mm.
- Peso neto: 6 g.

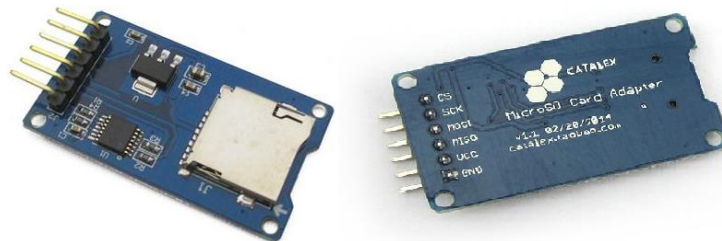


Fig. 4.7 Adaptador de tarjeta micro SD Catalex v1.0

4.8. Software Arduino

Cuando se dice que un microcontrolador es programable, se refiere a que permite grabar en su memoria de forma permanente el programa que se desee que dicho microcontrolador ejecute, en este caso, el Arduino Uno R3.

Antes de comenzar a programar el dispositivo es necesario introducir el concepto IDE, que significa en español, Entorno de Desarrollo Integrado. Esto es simplemente una forma de llamar al conjunto de herramientas software que permite a los programadores poder desarrollar sus propios programas con comodidad. En el caso de Arduino, se necesita un IDE que permita escribir y editar el programa, que permita comprobar que no se haya cometido ningún error.

El sistema operativo escogido para llevar a cabo la programación del microcontrolador es Windows, esto por conocimiento previo del sistema, lo que se traduce en mayor comodidad en su uso. El software utilizado y compatible con el sistema operativo Windows se denomina Arduino Software (IDE) que se obtuvo de [15]. La plataforma donde se crean las rutinas de código para las distintas implementaciones a realizar se muestra en la fig. 4.8.

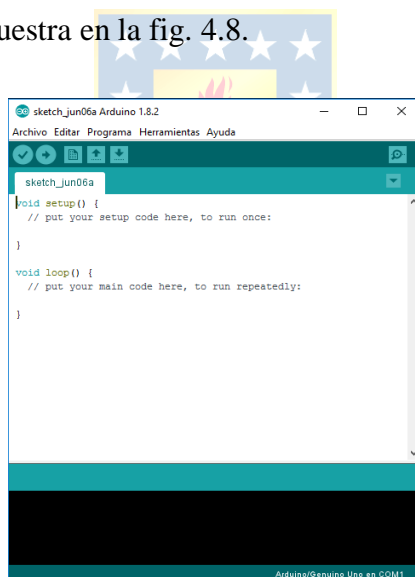


Fig. 4.8 Plataforma IDE de Arduino.

4.9. Comunicación Serie

La comunicación serie se define como una interfaz de comunicación de datos digitales que permite establecer transferencia de información entre varios dispositivos. Esto permite interactuar con el Arduino, recibiendo información y enviándosela cuando se necesite. Otra cosa que será muy útil de este tipo de comunicación es que permite darle las órdenes, previamente programadas, que se requieran.

Las funciones más importantes que se deben tener en cuenta para manejar el puerto serie de Arduino son:

- **begin():** Establece la velocidad de la UART en baudios para la transmisión serie, también es posible configurar el número de bits de datos, la paridad y los bits de stop. Por defecto es de 8 bits, sin paridad y un bit de stop.
- **read():** Lee primer byte entrante del buffer serie.
- **write():** Escribe datos en binario sobre el puerto serie. El dato es enviado como un byte o serie de bytes.
- **print():** Imprime datos al puerto serie como texto ASCII. También permite imprimir en otros formatos.
- **available():** Da el número de bytes (caracteres) disponibles para leer en el puerto serie, son datos que han llegado y se almacenan en el buffer serie que tiene un tamaño de 64 bytes.
- **end():** Deshabilita la comunicación serie permitiendo a los pines Rx y Tx ser usado como pines digitales.
- **if(Serial):** Especifica si el puerto serie está listo.
- **find():** Lee datos del buffer serie hasta encontrar el string buscado.
- **parseInt():** Busca el siguiente entero válido en el stream de datos del puerto serie.
- **readBytes():** Lee datos del buffer serie y lo guarda en una variable buffer.
- **setTimeout():** Configura el máximo de milisegundos de espera para la lectura del puerto serie. Por defecto es un segundo.

Buffer serial: Los puertos serie del microcontrolador tiene un buffer que se va llenando hasta que se lee con la función **read()**. El tamaño del buffer serie en el Arduino Uno es de 64 bytes, cuando se llena, el resto de elementos recibidos se pierde.

5. Sistema de presencia académica utilizando sensor de movimiento PIR HC-SR501

5.1. Introducción

El procedimiento a realizar en este capítulo consiste en subir los datos del microcontrolador extraídos de un sensor de movimiento y un tablero de luces controlado manualmente a un servidor web para que posteriormente, éste escriba los datos en una página web y así los alumnos tengan la posibilidad de conocer la presencia del docente y la disponibilidad presente y futura en su respectiva oficina.

El diagrama del sistema de presencia académico explicado anteriormente se presenta en la figura 5.1.

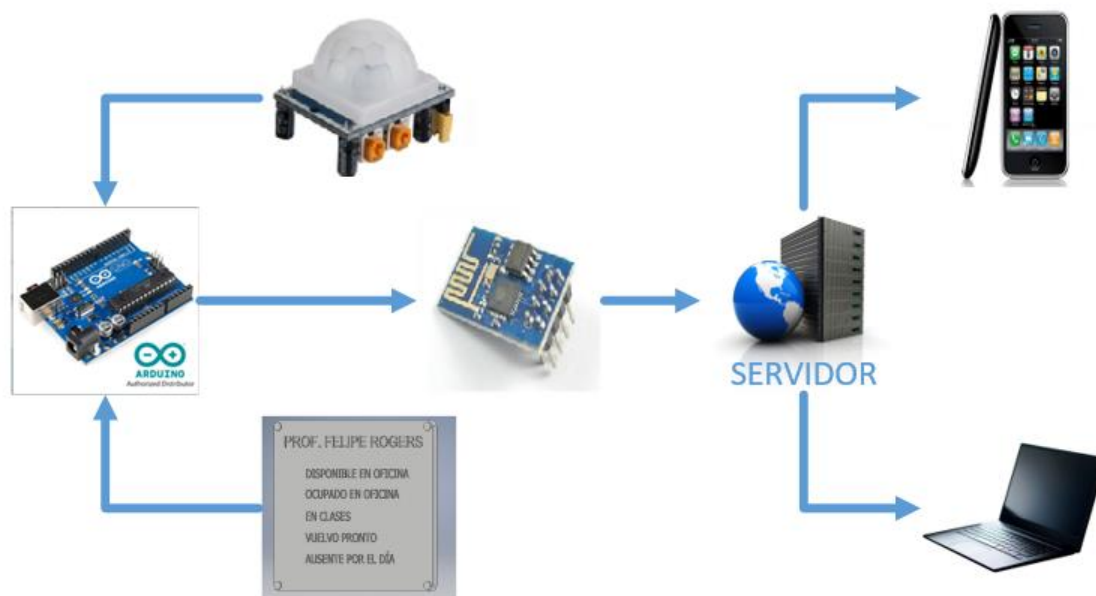


Fig. 5.1 Diagrama explicativo del sistema de presencia académica.

5.2. Sensores PIR

Los sensores PIR se basan en la medición de la radiación infrarroja. Todos los cuerpos emiten una cierta cantidad de energía infrarroja, mayor en cuanto mayor es su temperatura. Los dispositivos PIR cuentan con un sensor eléctrico capaz de captar esta radiación y convertirla en una señal eléctrica.

Cada sensor está dividido en dos campos y se utiliza un circuito eléctrico que compensa ambas mediciones. En el caso en que ambos campos reciban la misma cantidad de señal infrarroja, la señal eléctrica será nula. No así en el caso en que ambos campos reciban cantidades distintas de señal infrarroja, ya que si se generaría una señal eléctrica de salida.

Así, si un objeto atraviesa uno de los campos se genera una señal eléctrica diferencial, que es captada por el sensor, el cual emite una señal digital.

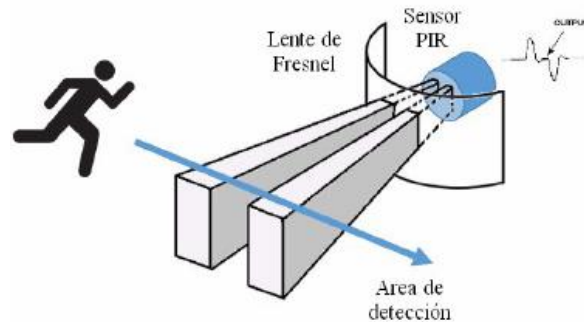


Fig. 5.2 Diagrama de principio de detección de un sensor PIR. [27]

Otro elemento importante en la funcionalidad del sensor es la óptica de éste. Básicamente corresponde a una cúpula de plástico formada por lentes de Fresnel, que divide el espacio en zonas y enfoca la radiación infrarroja a cada uno de los campos del PIR. De esta forma, cada uno de los sensores capta un promedio de la radiación infrarroja del entorno. Al momento de que un objeto entre en el rango del sensor, alguna de las zonas marcadas por la óptica recibirá una cantidad distinta de radiación, lo que será captado por uno de los campos del sensor, haciendo que éste dispare la señal digital.

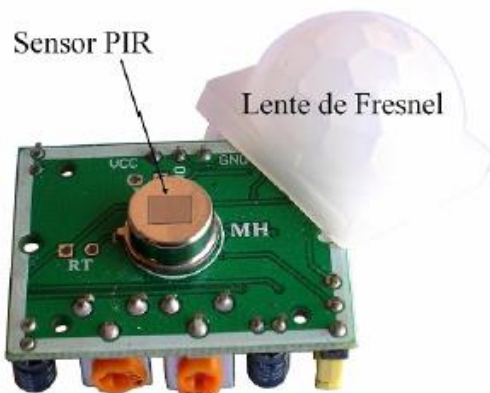


Fig. 5.3 Interior del sensor PIR y lente Fresnel. [27]

El ángulo de detección de movimiento de estos sensores es ajustable y generalmente funcionan con alcances de hasta 7 metros y aperturas de 90° a 110° , como se muestra en la figura 5.4.



Fig. 5.4 Rango de detección de movimiento de los sensores PIR. [27]

El sensor, además, cuenta con dos potenciómetros que tienen como finalidad permitir al usuario ajustar tanto el tiempo de disparo de la señal de alarma de movimiento, como la distancia de detección. El funcionamiento y sentido de los respectivos potenciómetros se muestra en la figura 5.5.

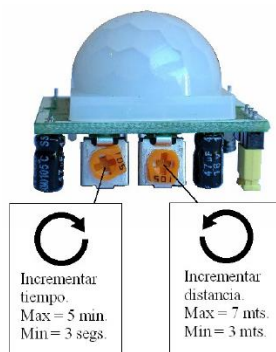


Fig. 5.5 Potenciómetros del sensor PIR. [27]

5.3. Conexión Microcontrolador – Sensor PIR HC-SR501

La idea de esta conexión es alertar, mediante una luz LED, si hay o no movimiento en alguna zona específica. Para fines del proyecto se utilizará específicamente en las oficinas del docente del Departamento de Ingeniería Civil Eléctrica de la Universidad de Concepción. La rutina necesaria para completar dicha acción se muestra en el anexo D.5, la cual prenderá un LED al momento de detectar movimiento en un rango y tiempo especificado por el usuario. Se establece como pin de entrada, el pin 2 del microcontrolador Arduino, que tiene la función de recibir la señal digital emitida por el sensor al detectar movimiento.

Si el Arduino detecta la salida del sensor a través del pin 2, activará la salida del pin 13, definido previamente así en la rutina. El pin 13 a su vez, estará conectado a un diodo LED y una resistencia para limitar su voltaje. De esta forma se consigue prender una luz LED al momento de detectar movimiento mediante un sensor PIR. La correcta conexión de este sistema está mostrada en la figura C.5 del anexo C.

Se pretende incluir esta función a través de una página web, con la finalidad de que cualquier alumno pueda tener conocimiento previo de la presencia del docente dentro de su oficina.

5.4. Conexión Microcontrolador – Módulo WIFI ESP-8266

La finalidad de utilizar esta conexión es permitirle al Arduino la posibilidad de conectarse a Internet, con el fin de que pueda dar a conocer el estado del sensor de movimiento en tiempo real a través de una página web.

De la hoja de especificaciones del módulo WI-FI se extrae que el voltaje de alimentación recomendado es de 3.3 V, por esta razón, se utilizará el pin de 3.3 V del Arduino para alimentar las patillas Vcc y CH_PD del módulo. Luego, se procede a conectar las patillas UTXD y URXD a los pines Tx y Rx (2 y 4) respectivamente. La última patilla que es necesario conectar, pero no menos importante, es la denominada GND perteneciente al módulo, que será conectada con el pin de GND del Arduino. Cabe mencionar que las correctas conexiones de tierra, son de mucha importancia, puesto que de olvidarlo podría dañar algún componente electrónico que se esté utilizando. La correcta conexión recién explicada se puede ver gráficamente en la figura C.3 del anexo C.

Ahora, cuando ya se implementa la conexión física del módulo y el microcontrolador, se procede a configurar el módulo para que se conecte a una red WI-FI. Lo primero es cargar la rutina correspondiente que permite dicha configuración rápidamente. Se comienza creando un nuevo objeto serie llamado BT1 a través del comando **SoftwareSerial**, el cual estará conectado a los pines 4 y 2

del microcontrolador, luego se continúa definiendo y ajustando las velocidades del microcontrolador y sus puertos de transmisión Tx y recepción Rx mediante los comandos **Serial.begin()** y **BT1.begin()** respectivamente. El siguiente paso es verificar si en el monitor serie no haya textos, a través del comando **Serial.available()**, para luego proceder a definir caracteres que serán leídos y recibidos por el Arduino y el módulo WI-FI. La rutina completa recién descrita se puede apreciar en el anexo D.2.

Ahora se procede a cargar el código al Arduino, luego de esto, se abre el monitor serie de la plataforma. Primero se debe configurar el monitor serie, cambiando las opciones de la barra inferior de éste, como se muestra en la figura 5.6. En el monitor serie debería aparecer un texto previo a mostrar la palabra “ready” (si esto no se cumple, se deberá reiniciar el módulo quitándole la alimentación y volviéndola a conectar). Ya habiendo aparecido la palabra “ready” se procede a configurar el módulo mediante los comandos **AT**, que son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un terminal módem. La tabla resumen de los comandos a utilizar se puede apreciar como tabla A.1 en el anexo A.

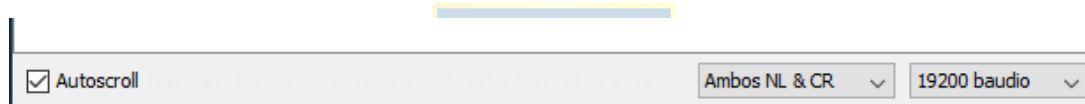


Fig. 5.6 Configuración necesaria del monitor serie para la configuración del módulo WI-FI.

El primer comando AT que se debe escribir en el monitor serial es precisamente “AT”, el cual habilita al módulo para recibir estos comandos, al escribirlo se deberá recibir del módulo en el monitor la palabra “ok” si se recibió satisfactoriamente. El siguiente comando tiene la función de reiniciar el módulo WI-FI por si previamente tenía una configuración distinta, que corresponde al comando **AT+RST**, al hacerlo correctamente se recibe la palabra “ok”. Para permitirle al módulo funcionar en modo estación se define el comando **AT+CWMODE=1**. Si se realiza el cambio correctamente se imprimirá la palabra “ok” y si es que el módulo ya estaba en este modo, se imprimirá la palabra “no change” que en inglés significa “sin cambios”. Ahora se procede a conectar el módulo a una red WI-FI, para ello se utiliza el comando **AT+CWJAP=”nombre de la red WI-FI”,”contraseña de la red WI-FI”** en el cual se debe rellenar con el nombre y la contraseña de la red a la que se desea conectar. El siguiente comando que puede ser de utilidad es **AT+CIFSR** el cual permite conocer la dirección IP que se le asignó al módulo WI-FI, para así luego, trabajar con ella. Finalmente se tiene el comando **AT+CIPMUX=0**, el cual tiene la función de deshabilitar las conexiones múltiples. Con esto el módulo queda configurado correctamente. Las imágenes 5.7, 5.8, 5.9, 5.10 y 5.11 muestran el proceso recién descrito.

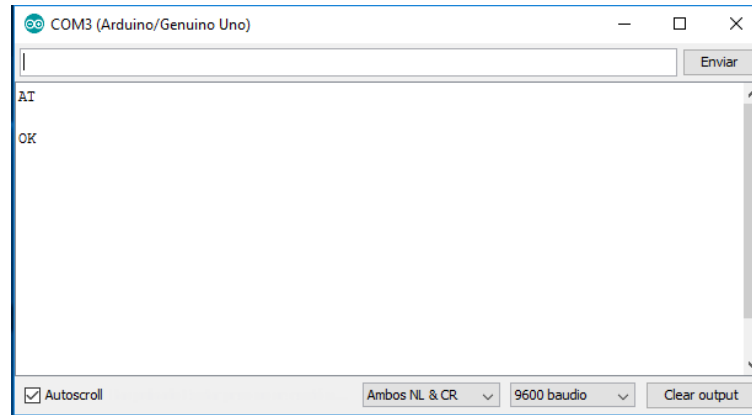


Fig. 5.7 Monitor serial luego de recibir el comando AT correctamente.

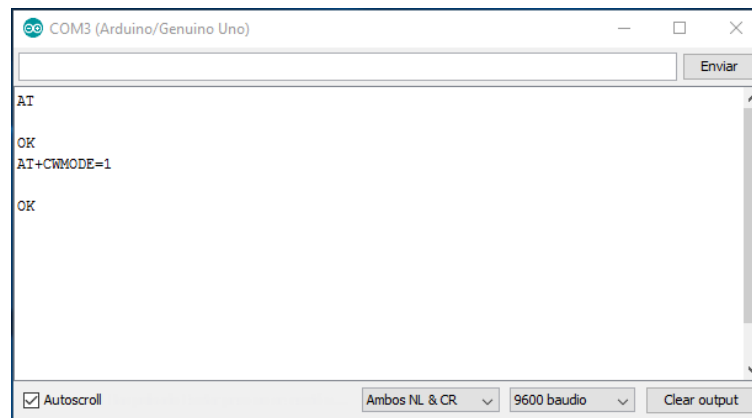


Fig. 5.8 Monitor serial luego de recibir el comando AT+CWMODE=1 correctamente.

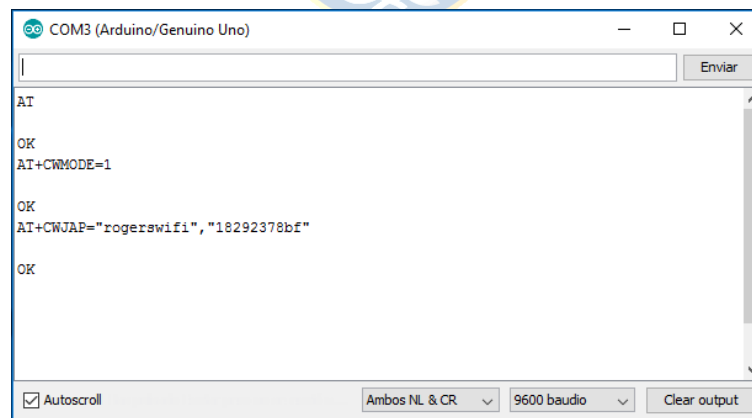
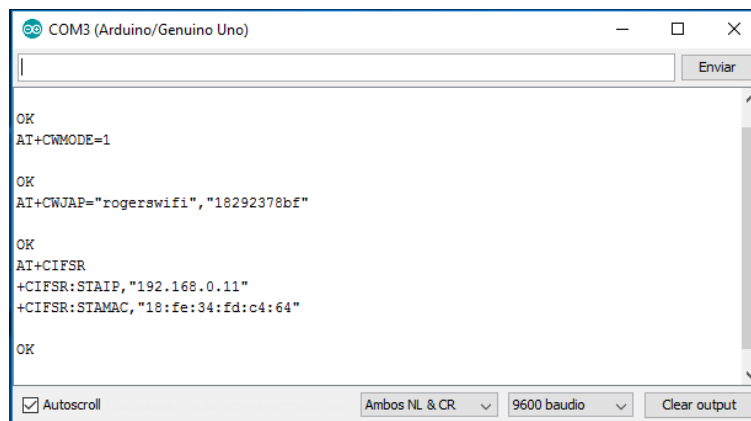


Fig. 5.9 Monitor serial luego de recibir el comando AT+CWJAP="nombre de red", "contraseña de red" correctamente.

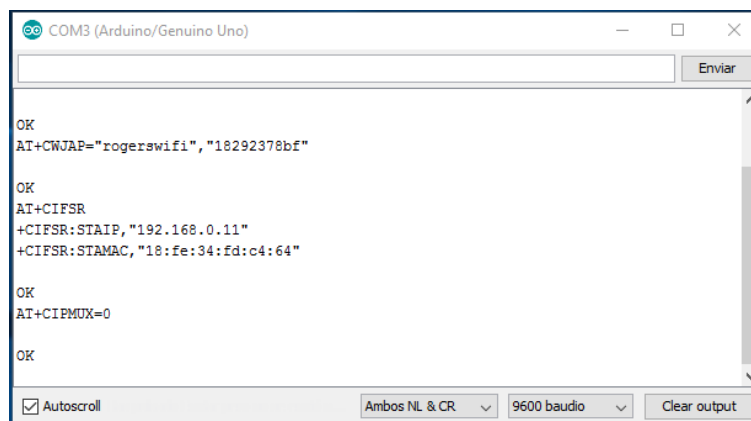


```

COM3 (Arduino/Genuino Uno)
Enviar
OK
AT+CWJAP=1
OK
AT+CWJAP="rogerswifi","18292378bf"
OK
AT+CIFSR
+CIFSR:STAIP,"192.168.0.11"
+CIFSR:STAMAC,"18:fe:34:fd:c4:64"
OK
Autoscroll Ambos NL & CR 9600 baudio Clear output

```

Fig. 5.10 Monitor serial luego de recibir el comando AT+CIFSR correctamente, por lo que entrego la dirección IP asignada al módulo.



```

COM3 (Arduino/Genuino Uno)
Enviar
OK
AT+CWJAP="rogerswifi","18292378bf"
OK
AT+CIPMUX=1
+CIPMUX:0
OK
AT+CIPMUX=0
OK
Autoscroll Ambos NL & CR 9600 baudio Clear output

```

Fig. 5.11 Monitor serie luego de recibir los comandos AT+CIPMUX=1 correctamente.

Cabe destacar que todo el proceso anteriormente descrito, se realizará de manera automática en el código final del sistema.

La combinación de las dos conexiones y configuraciones anteriores nos permitirá informar el estado del sensor a través de Internet, a esto, se le agrega la creación de una página web que se encargará de ser la interfaz gráfica de información hacia los estudiantes y que estará dentro de un servidor web gratuito.

La página web se configura de tal manera que muestre una imagen del escudo de la Universidad de Concepción y bajo éste, se informe textualmente dos estados del sensor. Si el sensor detecta movimiento (estado=1), se imprimirá en la página web “Profesor presente en la oficina”. En caso contrario (estado=0), la página web mostrará bajo el escudo “Profesor no está presente en la oficina”. Además de esto, se incorpora la opción de dar a conocer la disponibilidad que el propio docente desea dar a conocer a sus alumnos en un instante dado, es decir, la disponibilidad que estará siendo informada en el tablero de presencia académica fuera de la oficina del docente. Esta

información dependerá del diodo LED que el profesor deje encendido manualmente a través de switches para cada luz, los que determinarán el valor de una constante llamada “**state**” que posteriormente será subida al servidor web para que éste imprima el texto correspondiente en la página web. El código D.6 es el que da origen al cuerpo principal de la página web, mientras que la configuración de los textos está dada por la rutina D.7 en el anexo D.

5.5. Servidor Web

El sistema de envío de información desde el servidor web y la página web que ven los usuarios del sistema se puede representar mediante el diagrama de la figura 5.12.

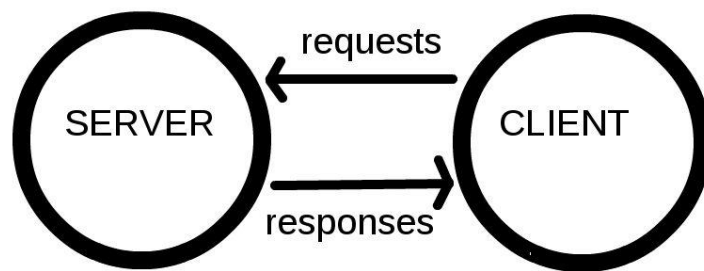


Fig. 5.12 Diagrama de la interacción cliente-servidor. [29]

Los clientes son dispositivos conectados a Internet mediante algún software o navegador que permite el acceso a Internet en dichos dispositivos, en este caso, corresponde a los usuarios del sistema que ven la información de los sensores del Arduino a través de la página web. En cambio, los servidores, son computadoras que almacenan páginas web, sitios o aplicaciones. Cuando un dispositivo cliente quiere acceder a una página web, una copia de la página web se descarga desde el servidor en el equipo cliente y se muestra en el navegador del usuario.

Además de los clientes y servidores, existen otros factores necesarios para establecer la comunicación dentro de la web, los que son:

- **Conexión a Internet:** Permite el envío y recepción de datos en la web.
- **Protocolo de control de transmisión y protocolo de Internet (TCP/IP):** Protocolos de comunicación que definen cómo deben ser transportarse los datos a través de la web.
- **Servidores del sistema de nombres de dominio (DNS):** Libreta de direcciones de sitios web. Cuando se escribe una dirección web en el navegador, el navegador busca

los DNS antes de recuperar el sitio web. El navegador averigua en qué servidor está alojado el sitio web antes de enviar los mensajes HTTP.

- **Protocolo de transferencia de hipertexto (HTTP):** Protocolo de aplicación que define un idioma para que los clientes y servidores se puedan comunicar.
- **Archivos componentes:** Un sitio web se compone de muchos archivos diferentes. Estos archivos se dividen principalmente en dos tipos, los archivos de código, los que corresponden a la constitución principal de un sitio web (HTML, CSS y Javascript) y los recursos, que corresponde al nombre colectivo del resto de materiales que conforman un sitio web como, por ejemplo, las imágenes, música, video, etc.

Las direcciones web reales no son las agradables y fácilmente recordables secuencias que se teclea en la barra de direcciones con el fin de encontrar los sitios web. En realidad, se trata de números como, por ejemplo, 63.245.217.105. Lo anterior se denomina dirección IP y representa un lugar único en la web. Sin embargo, como no es muy fácil de recordar, se crearon los servidores de nombres de dominio. Éstos, son servidores especiales que hacen coincidir una dirección web tecleada desde tu navegador con la dirección real del sitio web.

Finalmente, cuando se escribe una dirección web en el navegador de algún dispositivo electrónico, en primer lugar, el navegador va al servidor DNS y encuentra la dirección real del servidor donde se encuentra el sitio web, luego envía un mensaje de petición HTTP al servidor, pidiéndole que envíe una copia de la página web para el cliente a través de una conexión a Internet usando TCP/IP. En el caso de que el servidor apruebe la solicitud del cliente, enviará el mensaje de “200 OK”, para luego comenzar a enviar los archivos de la página web al navegador como una serie de pequeños trozos llamados paquetes de datos. Finalmente, el navegador reúne los pequeños trozos, forma un sitio web completo y lo muestra al usuario.

5.6. Comunicación Microcontrolador – Servidor Web

Después de verificar las conexiones físicas del microcontrolador con los diodos LED, el sensor de movimiento y el módulo WIFI, se utiliza la rutina D.8 del anexo D para establecer la comunicación con el servidor web y así concluir la realización del sistema propuesto.

Lo primero que se establece en la rutina son las librerías, en este caso solo se agregan las librerías necesarias para utilizar el monitor serial y el módulo WIFI ESP8266. Posteriormente se definen las variables que se utilizarán en el código, de las cuales, las que tomarán mayor importancia son las denominadas “variable1” y “state”, que contienen la información del sensor y del diodo LED encendido respectivamente, que será entregada al servidor web.

Luego se pasa a la configuración de la rutina, en donde se establecen las velocidades de comunicación tanto del módulo WIFI como del monitor serial. Los pines digitales a utilizar se configuran como entrada y se procede a configurar el módulo WIFI automáticamente a través de los comandos AT, para poder establecer correctamente la comunicación con el servidor web.

En el “loop” de la rutina, se comienza con la lectura del sensor de movimiento a través del comando “**digitalRead()**”, luego se procede a establecer las condiciones que debe tener la variable “state” dependiendo de la luz LED encendida, las que se resumen en la tabla 5.1.

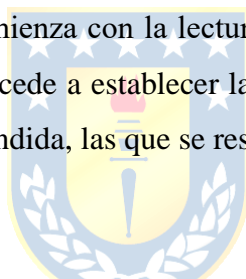


Tabla 5.1 Estados de las luces LED.

<i>“state”</i>	<i>Luz LED(pin digital)</i>	<i>Disponibilidad del docente</i>
1	Verde (8)	Disponible en oficina
2	Amarillo (7)	Ocupado en oficina
3	Azul (11)	En clases
4	Naranja (12)	Vuelvo pronto
5	Rojo (10)	Ausente por el día

El valor de la variable “state” depende de los 5 interruptores que dejarán pasar corriente hacia los cinco diodos LED y hacia los respectivos pines digitales (como entrada) asignados. Una pequeña limitación que es importante mencionar, es la dependencia manual de cada interruptor, es decir, que para realizar el cambio de un LED encendido a otro primero el docente deberá apagar el primero y luego encender el otro. En cualquier caso, de haber dos diodos encendidos en un mismo instante, la variable “state” sólo tomará el valor de uno de los dos dependiendo del momento en el que se encuentre el loop.

Ahora, teniendo definida la información que será enviada al servidor, se inicia la comunicación con éste. A través del comando **“SerialESP8266.println()”** perteneciente a la librería incluida en la rutina, se procede a enviar un comando AT al módulo WIFI, preparándolo para establecer la comunicación a través del protocolo TCP y el puerto 80. Posteriormente se forma el encabezado de la petición http que será enviada al servidor, definiendo la ruta a seguir por las variables para llegar a la API y así, al servidor web. Luego, nuevamente se escribe en el módulo WIFI, pero esta vez el comando **“AT+CIPSEND”** que es el encargado de enviar la longitud de caracteres de la petición http.

Se define una variable de tipo booleana con el objetivo de dar por finalizado el ciclo de envío de respuesta, que puede terminar antes de ser enviada por dos razones, o la respuesta excede el máximo de caracteres definidos o después de transcurrirse 10 segundos, también definidos por código. Si no pasa lo anterior descrito y en la respuesta se obtiene, a través del comando **“.indexOf()”**, la palabra **“CLOSED”**, significa que la información fue recibida satisfactoriamente y se finaliza la comunicación. Ahora, se tiene acceso a la información de la página web, utilizando cualquier navegador de Internet y escribiendo en la barra de navegación la ruta **“Project.felipe.saidgeek.io”**. La correcta conexión del sistema completo recién descrito se muestra en el diagrama de la figura C.6 del anexo C.

Para la implementación de este sistema, se diseña un tablero con la capacidad de albergar los elementos y módulos utilizados. Las imágenes mostradas en la figura 5.13 muestran el diseño del tablero que se realizó a través del software Autodesk Inventor 2018.

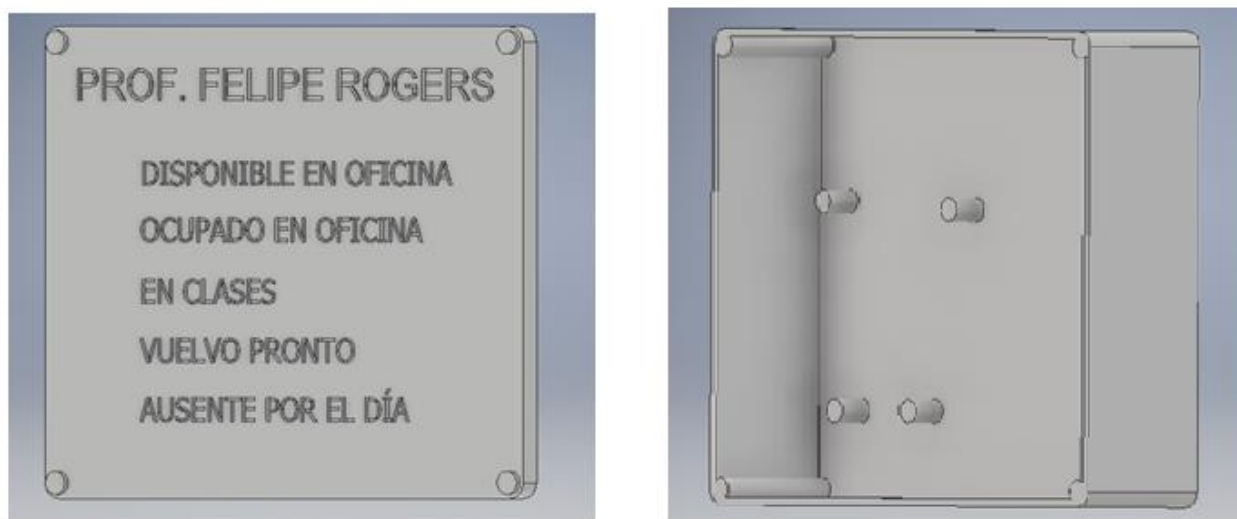


Fig. 5.13 Vista frontal y lateral de la simulación de la tapa del tablero de presencia académica.

Luego de terminado el diseño, se procede a realizar la impresión mediante la utilización de una de las impresoras que el Departamento de Ingeniería Eléctrica dispone. El material utilizado para la impresión del tablero corresponde a ácido poli láctico. Los resultados de lo anterior se pueden apreciar en las imágenes mostradas en la figura 5.14.



Fig. 5.14 Vista frontal y lateral de la tapa del tablero de presencia académica impreso.

Las medidas del tablero y tapa del sistema son de 12 cm de largo, 14.5 cm de ancho. El espesor de la tapa es de 0.4 cm mientras que la altura del tablero es de 3.2 cm. Se contempla un espacio en cada esquina de la tapa y el tablero para ubicar círculos de 6 mm de diámetro y 1 mm de alto que cumplirán la labor de ser perforados para atornillar la tapa y el tablero.

6. Sistema de registro de asistencia a clases utilizando lector de huella dactilar

6.1. Introducción

El procedimiento a realizar en este capítulo consiste en diseñar un sistema de registro de asistencia a clases a través de la conexión del microcontrolador Arduino y un lector de huellas dactilares. La idea es que el lector de huellas dactilares reconozca una huella previamente guardada en su memoria y por medio del microcontrolador, que tendrá la capacidad de llevar registro de la hora y fecha actual gracias a la conexión con un reloj RTC DS3231, guarde dentro de una memoria micro SD, mediante un lector de tarjeta micro SD Catalex para Arduino, el registro de la hora y fecha exacta en la que fue reconocida dicha huella.

El sistema completo se presenta en el diagrama de la figura 6.1.

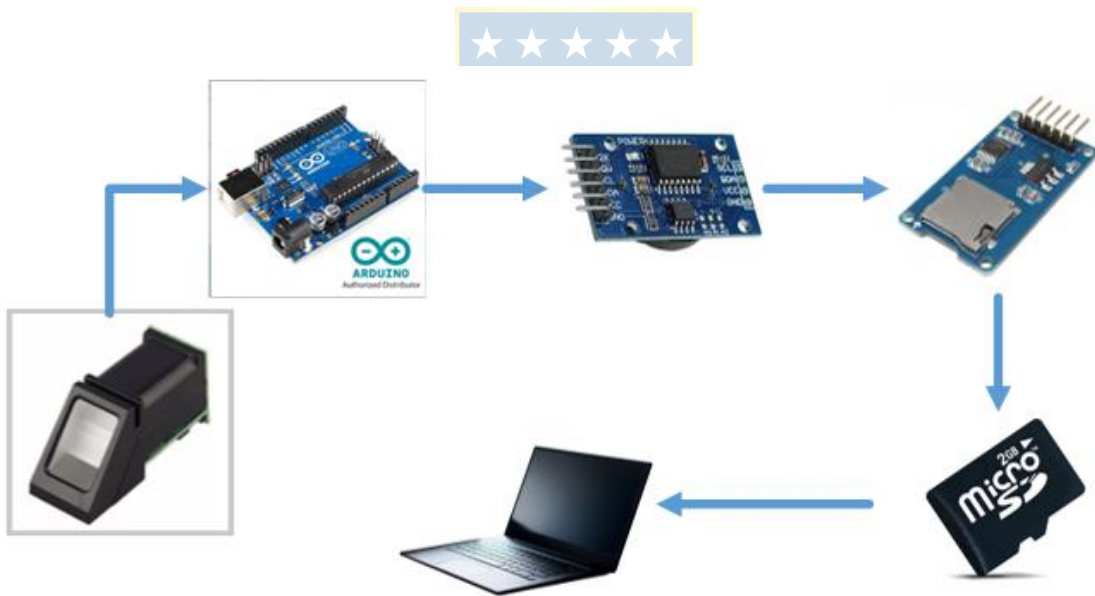


Fig. 6.1 Diagrama explicativo del sistema de registro de asistencia a clases.

6.2. Sistemas biométricos

El concepto proviene del griego “bios” que significa “vida” y “metron” que significa medida. Consiste en la toma de medidas estandarizadas de los seres vivos o de procesos biológicos. En materia de la tecnología de la información, la biometría hace referencia a las tecnologías que miden y analizan las características del cuerpo humano, como el ADN, las huellas dactilares, los patrones faciales o de la voz, las medidas de las manos a efectos de autenticación de identidades y la retina y el iris de los ojos.

En resumen, biometría se define como “la parte de la biología que estudia en forma cuantitativa la variabilidad individual de los seres vivos por medio de métodos estadísticos. Al aplicar técnicas matemáticas y estadísticas sobre los rasgos anteriormente mencionados, se logra verificar la identidad del individuo. Estas características físicas o estáticas y de comportamiento o dinámicas conforman los diferentes tipos de sistemas biométricos.

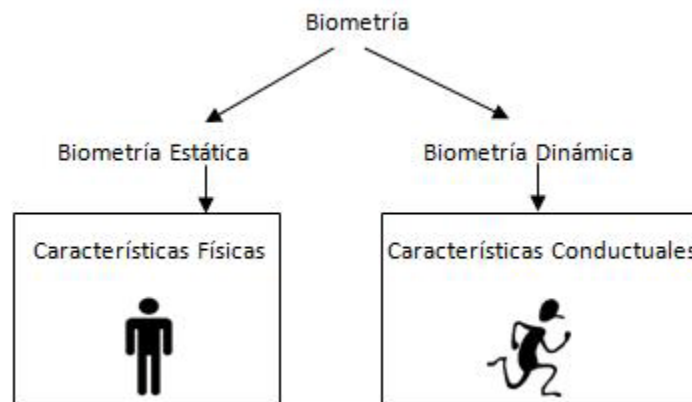


Fig. 6.2 Clasificación de los sistemas biométricos por tipo. [19]

La medición de las características físicas de un individuo corresponde a la biometría estática. Los principales estudios y aplicaciones de esta rama de la biometría están basados en los sistemas biométricos de huellas dactilares, geometría de la mano, reconocimiento facial y análisis de iris y retina. Por el contrario, las mediciones de los rasgos de comportamiento de un individuo forman parte de la biometría dinámica y dentro de esta rama los principales estudios y aplicaciones están basados en los sistemas de reconocimiento de voz y firma manuscrita principalmente.

En conclusión, la medición biométrica es considerada como el método ideal de identificación humana y control de acceso. Esta identificación consiste en la comparación de las características específicas de cada individuo con un patrón conocido y almacenado en una base de datos.

A continuación, se presentan las ventajas y desventajas que presentan los sistemas biométricos:

Ventajas:

- En el ámbito de la informática doméstica o laboral, la tecnología biométrica permite al usuario identificarse en distintas plataformas sin necesidad de recordar contraseñas o llevar consigo una tarjeta o llave de acceso, lo que se traduce en una mayor comodidad.
- Los sistemas biométricos son robustos y proporcionan un control de acceso físico confiable en entornos que requieran alta seguridad, puesto que la biometría garantiza uno de los niveles de autenticación menos franqueables hoy en día.
- Las características biométricas de cada persona no son transferibles a otra.
- Los costos de administración se reducen puesto que solo se debe realizar mantenimiento del lector que se utilice.

Desventajas:

- Algunos sistemas biométricos utilizan tecnologías muy caras.
- Como todo sistema informático, puede ser susceptible a errores en la identificación o que pueda ser burlado mediante suplantación de identidad como sería el caso en que alguien consiga la propia huella del dueño que tiene acceso al sistema.

6.2.1 Arquitectura de los sistemas biométricos

La arquitectura típica de un sistema biométrico puede entenderse conceptualmente como dos módulos:

- **Módulo de inscripción:** Se encarga de obtener y guardar la información que provenga del indicador biométrico con el fin de contrastar a ésta con la proporcionada en ingresos posteriores al sistema. Las tareas que ejecuta el módulo de inscripción son posibles gracias a la acción del lector biométrico y del extractor de características.

Con respecto al proceso que tiene que ver con el módulo de inscripción, los dispositivos biométricos poseen tres componentes básicos:

- El primero se encarga de la adquisición análoga o digital de algún indicador biométrico de un individuo, como por ejemplo, la obtención de la imagen de una huella dactilar mediante un escáner.

- El segundo maneja la compresión, procesamiento, almacenamiento y comparación de los datos adquiridos (en el ejemplo anterior, a partir de la salida del lector, extraemos una imagen) con los datos almacenados.
- El tercer componente tiene la función de establecer una interfaz con aplicaciones que se ubican en el mismo sistema u otro.
- **Módulo de reconocimiento:** Es el encargado de reconocer a los individuos en la aplicación de control de acceso. El proceso de identificación se inicia cuando el lector biométrico captura la característica única del individuo y la convierte a un formato digital, con el fin de que el extractor de características produzca una representación compacta con el mismo formato que un “template”, el que se define como la información representativa del indicador biométrico que se encuentra almacenada y será usada en las tareas de reconocimiento al ser comparada con la información proveniente del indicador biométrico en el punto de acceso. La representación consultante, llamada “query”, se envía al comparador de características que confronta a éste con uno o varios “templates” con el objetivo de establecer la identidad.
 - Fase operacional: Los procesos que realiza el módulo de identificación.



Fig. 6.3 Procesos de Inscripción y Reconocimiento. [28]

La información que proveen los templates permite particionar la base de datos de acuerdo a la presencia o ausencia de ciertos patrones particulares para cada indicador biométrico. Las clases así generadas permiten la reducción del rango de búsqueda de algún template en la base de datos. Sin embargo, los templates que pertenecen a una misma clase también presentarán diferencias, las que son conocidas como “variaciones intraclase”. Estas variaciones implican que la identidad de un individuo puede ser establecida solo con un cierto nivel de confianza.

Una decisión tomada por un sistema biométrico reconoce una característica biométrica de su base de datos o no hacerlo, a lo cual llamaremos persona autorizada o persona no autorizada. Para cada tipo de decisión, existen dos salidas posibles, las que son verdadero o falso. Por lo tanto existe un total de cuatro posibles respuestas del sistema:

- 1) Una persona autorizada es aceptada (salida correcta).
- 2) Una persona autorizada es rechazada (salida incorrecta).
- 3) Una persona no autorizada es rechazada (salida correcta).
- 4) Una persona no autorizada es aceptada (salida incorrecta).

6.3. Huella dactilar

6.3.1 Introducción

La ciencia que tiene por objetivo estudiar la huella dactilar se denomina “Dactiloscopia” y todos los sistemas dactiloscópicos cuentan con tres principios fundamentales, la perennidad, inmutabilidad y la diversidad infinita.

La inmutabilidad, el desarrollo físico y las enfermedades no afectan a las características de una huella dactilar. En caso de un desgaste involuntario, por ejemplo, una herida, el tejido epidérmico que la conforma se regenera conservando la forma original.

En el caso de la diversidad infinita, se puede mencionar que con las características que contiene cada huella, constituye un patrón único para cada persona con la combinación de rasgos que se verá posteriormente. Se estima así, que la probabilidad de que dos individuos tengan las mismas huellas dactilares es 1 en 64.000.000.000. Incluso tratándose del caso de gemelos monocoriales monoamnióticos, los cuales comparten tanto la placenta como la bolsa, las sutiles diferencias de la posición que mantienen en el útero y las distintas experiencias intrauterinas de los fetos determinan diferencias inconfundibles en las huellas dactilares.

6.3.2 Procesamiento de la huella digital

El procedimiento a seguir para el procesamiento de la huella dactilar a través de un sistema automatizado es:

- **Mejora de la imagen:** Consiste en eliminar las zonas confusas o borrosas de la imagen original.
- **Binarización:** Convertir la imagen original a tonos de gris a blanco y negro.
- **Adelgazamiento:** Se procesan las líneas dactilares para que tengan el mismo grosor, con el objetivo de facilitar la identificación de los puntos característicos.
- **Extracción de puntos característicos:** Se detectan y extraen las posiciones exactas de cada punto característico. Dentro de esta etapa se destacan:
 - **Construcción de un índice o vector:** Corresponde al proceso final que mediante algoritmos matemáticos completa la creación de un índice matemático, el cual constituye la esencia de la huella dactilar analizada, según las características que se consideran, almacenándolo en forma de fichero.
 - **Identificación y verificación:** Luego de tener el índice o vector de varias huellas, el sistema es capaz de realizar búsquedas comparativas para verificar la identidad de un individuo.

La extracción de puntos característicos es por lo tanto, el proceso final que completa la obtención de la plantilla de la huella o patrón biométrico dactilar. La cantidad mínima de puntos característicos necesarios para proceder a comparaciones eficaces entre imágenes dactilares es de 15.

6.4. Lectores de huella dactilar

Un lector de huella dactilar contempla por principio de funcionamiento la obtención de una imagen de la huella, de la que extrae un patrón y lo compara con los patrones de las huellas que tiene almacenado en su memoria.

Luego de que un lector ha leído la huella digital, necesita analizar la imagen obtenida. Con el fin de reducir el consumo del procesador, sólo se comparan rasgos específicos de la imagen, las anteriormente mencionadas minucias.

El software del lector utiliza algoritmos muy complejos para reconocer y analizar las minucias. El procedimiento consiste en medir las posiciones relativas de las minucias, y si dos imágenes poseen terminaciones de crestas y bifurcaciones que forman la misma figura dentro de la misma dimensión, significa que existe una gran probabilidad de que pertenezcan al mismo individuo.

En conclusión, el lector no necesita encontrar el patrón completo de una minucia en la muestra y en la imagen almacenada para obtener una coincidencia, sino que simplemente debe encontrar un número suficiente de patrones que tengan en común.

6.5. Conexión Microcontrolador – Sensor dactilar

El Sensor dactilar le permite al microcontrolador Arduino tener la capacidad de leer o escanear huellas dactilares. Se utilizará el sensor de “Adafruit” de la figura 4.2, que tiene todas las características para realizar un sistema capaz de procesar, analizar y reconocer una huella digital.

Para comenzar con la implementación, se tiene que tener en consideración las respectivas conexiones físicas del sensor y el Arduino. En la figura C.4 del anexo C se muestra la correcta conexión de la forma:

- Cable verde (Tx): Pin 2.
- Cable blanco (Rx): Pin 3.
- Cable rojo: Pin 5V.
- Cable negro: Pin GND.



6.6. Base de datos del sensor de huella dactilar

Luego de establecer la conexión se configura el sensor para establecer la base de datos dentro de éste, es decir, incorporar imágenes de huellas digitales en la memoria del sensor para luego ser comparadas y realizar alguna acción específica según se haya programado. Lo anterior se denomina “enrolar”. Para cumplir con el objetivo especificado, se tiene una librería de uso abierto a los usuarios del sensor, creada por “Adafruit”, que establece el enrolamiento de una huella digital en el sensor. Dicha librería de uso libre sirvió de base para la creación de la rutina D.3 presentada en el anexo D, modificada para fines del proyecto y que tiene por objetivo guardar en memoria las huellas de los usuarios.

El principio de acción de esta rutina contempla el uso del monitor serial anteriormente descrito, el cual sirve de interfaz de comunicación entre el usuario y el microcontrolador. Lo primero que hace la rutina es reconocer la correcta conexión del sensor con el Arduino, en caso contrario, se mostrará un mensaje de error en el monitor serial. Luego de esto, se requiere automáticamente un número de identificación en el cual asignar la imagen de la huella que se almacenará posteriormente.

Al momento de escribir un número de identificación para la imagen a almacenar, el monitor serial requiere al usuario acercar la huella dactilar al lente del sensor, a continuación, se requiere

remover el dedo y volver a acercarlo, con el fin de tomar dos imágenes y así asegurar una buena impresión. En el caso de que el enrolamiento sea exitoso, el monitor serial arrojará un mensaje expresando primero, que las dos imágenes tomadas coinciden y procederá finalmente a almacenarla en el número de identificación definido, este procedimiento se muestra en las figuras 6.4 y 6.5 para, en el caso de este ejemplo, el número de identificación 7. Luego de un proceso de enrolamiento exitoso o fallido, el monitor serial requerirá un nuevo número de identificación para repetir el proceso.

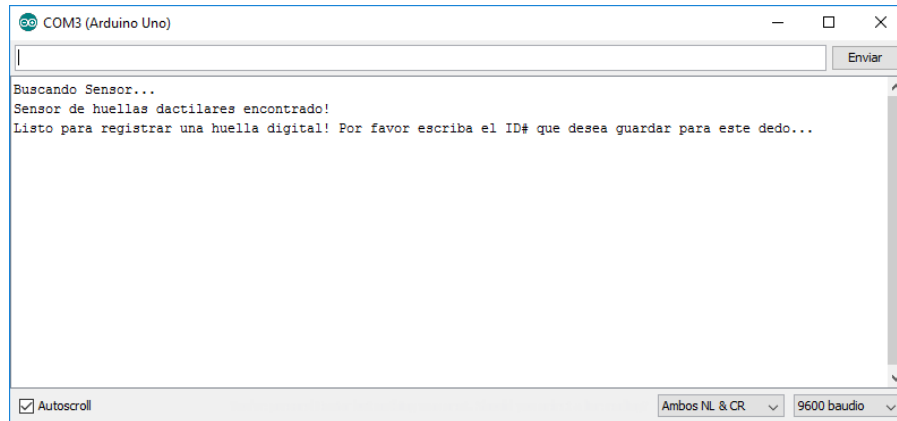


Fig. 6.4 Monitor serial luego de detectar la correcta conexión del sensor dactilar para luego pedir al usuario un número de identificación para el usuario a enrolar.

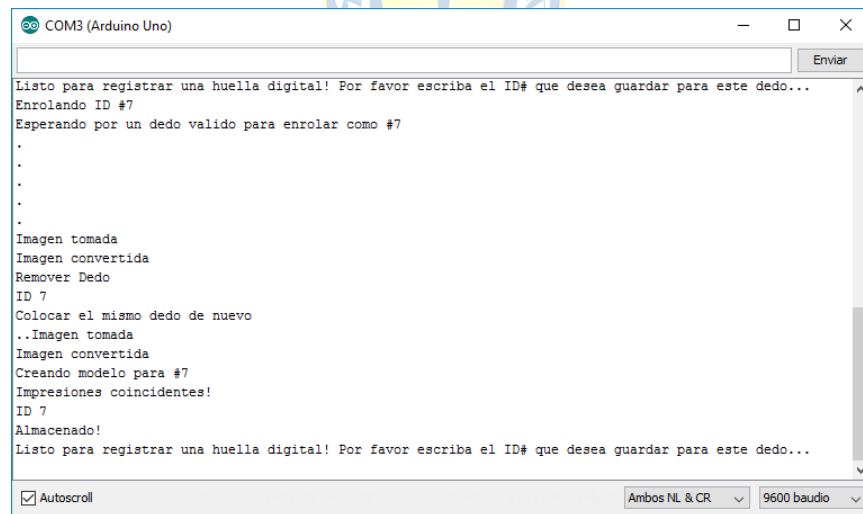


Fig. 6.5 Monitor serial luego de enrolar exitosamente un usuario a través de su huella dactilar.

6.7. Reconocimiento de huella y escritura en memoria micro SD (datalogger)

Luego de establecer la base de datos dentro del sensor dactilar, se utiliza la rutina presentada en el anexo D.4, la cual reconoce las huellas dactilares de los usuarios almacenados en la memoria del sensor. De esta forma se pueden establecer distintas acciones al aceptar o rechazar a un usuario, como por ejemplo, activar un servomotor para abrir una puerta, prender un LED si es aceptado o si es rechazado, entre otros. El objetivo que tiene la rutina es registrar en la memoria microSD, la hora y el nombre de cada individuo dueño de una huella dactilar que sea reconocida por el sensor.

La rutina comienza añadiendo las librerías necesarias para la utilización del reloj RTC, el lector de memoria micro SD y el lector de huella dactilar, luego se establece una variable tipo **“File”**, la que será manipulada por el lector micro SD para la apertura y cierre del documento de texto en el que se procederá a escribir dentro de la memoria. Luego se establecen variables tipo **“int”** y **“String”**, la primera llamada **“getFingerprintIDez()”** que será una función encargada de obtener el valor de retorno en el reconocimiento de la huella dactilar y la segunda llamada **“DaysOfTheWeek”** que permitirá crear un tipo de variable capaz de ser reconocida por el reloj RTC con el fin de conocer el día de la semana correspondiente a la hora en tiempo real que escriba éste.

En el ciclo de configuración, se procede a verificar la correcta conexión de los tres módulos utilizados y se define un pin digital como salida destinado a un LED que permitirá al usuario saber si su huella dactilar fue reconocida por el sensor.

Luego de esto, se procede a crear una función llamada **“printDate”**, la cual tiene una variable tipo **“DateTime”** de retorno que contiene la hora y fecha exacta que escribe el reloj RTC en tiempo real que posteriormente es escrita en la memoria micro SD. En la función loop se inicia la función **“getFingerprintIDez()”** que se encarga del reconocimiento de la huella dactilar y la posterior acción que se requiera, en este caso, la escritura en memoria micro SD del nombre del individuo y la hora en la que se efectúa el registro. También se procede al encendido de un LED al momento de detectar una huella, con el fin de dar a conocer al usuario que el registro se realizó efectivamente y finalmente se agrega un retraso de 50 ms con el fin de hacer parpadear el sensor al momento de estar listo para reconocer una huella.

La función arroja como retorno, el número de identificación de la huella que se registró en la rutina de base de dato, por otra parte, retorna un -1 mientras no se reconozca ninguna huella dactilar.

Es importante destacar que luego de guardar una nueva en la base de datos del lector dactilar, se proceda a escribir la siguiente parte de rutina en el datalogger con el número de registro (ID) de dicha huella.

```

if (finger.fingerID == 2) {
  archivo = SD.open("datos.txt", FILE_WRITE);

  if (archivo){
    archivo.print("Felipe Rogers :");
    DateTime now = rtc.now();
    printDate(now);
    archivo.close();

  }
  else{
    Serial.println("el archivo no se pudo abrir");
  }
}

```



Esto con el fin de que el datalogger cumpla correctamente la función de registrar al alumno con su nombre en la memoria micro SD. La correcta conexión del sistema completo se muestra en la imagen C.7 del anexo C.

Para la implementación del sistema, se diseña un tablero con la capacidad de albergar los elementos y módulos utilizados. Las imágenes mostradas en la figura 6.6 muestran el diseño del tablero que se realizó a través del software Autodesk Inventor 2018.

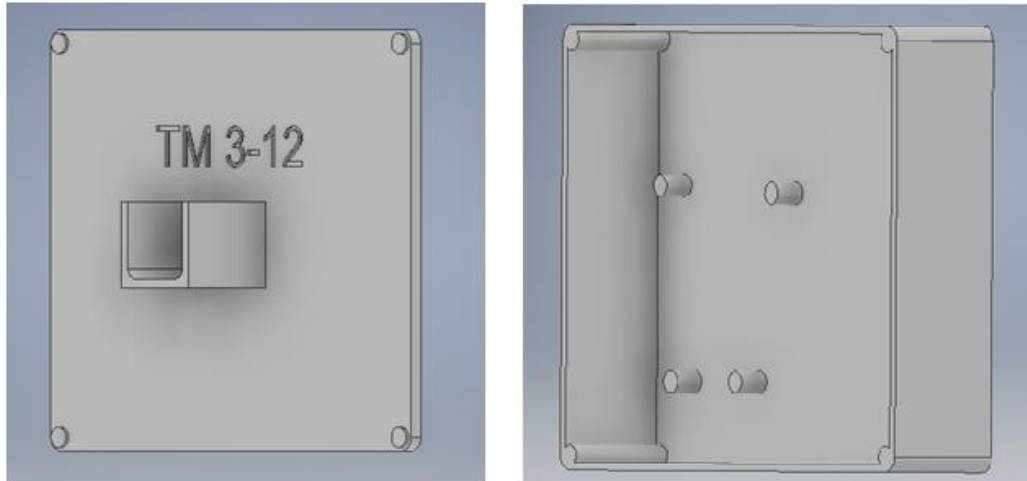


Fig. 6.6 Vista frontal y lateral de la simulación de la tapa del tablero de registro de asistencia.

Luego de terminado el diseño, se procede a realizar la impresión mediante la utilización de una de las impresoras que el Departamento de Ingeniería Eléctrica dispone. El material utilizado para la impresión del tablero corresponde a ácido poli láctico. Los resultados de lo anterior se pueden apreciar en las imágenes mostradas en la figura 6.7.

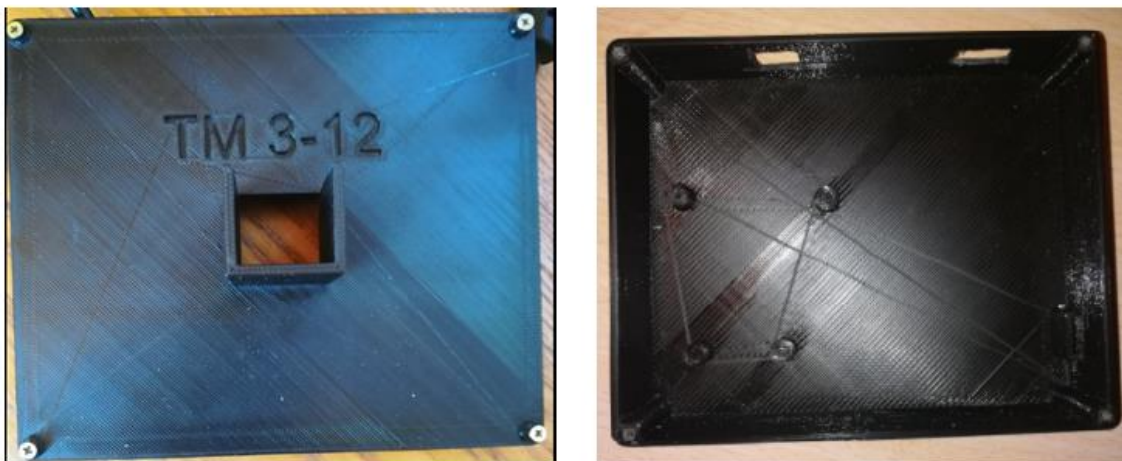


Fig. 6.7 Vista frontal y lateral de la tapa del tablero de registro de asistencia.

Las medidas del tablero y tapa del sistema son de 12 cm de largo, 14.5 cm de ancho. El espesor de la tapa es de 0.4 cm mientras que la altura del tablero es de 3.2 cm. Las medidas del espacio destinado al lector de huellas dactilares de la tapa del sistema son 2.2 cm de largo, 2.2 cm de ancho y 3.1 cm de alto. Se contempla un espacio en cada esquina de la tapa y el tablero para ubicar círculos de 6 mm de diámetro y 1 mm de alto que cumplirán la labor de ser perforados para atornillar la tapa y el tablero.

7. Resultados

Este capítulo consiste en establecer los resultados que arrojan los dos sistemas antes descritos. Respecto al sistema de presencia académica, se realizó satisfactoriamente su implementación, basada en la conexión de un microcontrolador y un sensor de movimiento. Se presentan a continuación las imágenes del sistema implementado.



Fig. 7.1 Tablero de presencia académica en su etapa final.

Se creó un tablero para el sistema de interruptores, con el fin de hacer más amigable la implementación y a la vista del propio docente. La imagen 7.2 muestra dicho tablero.

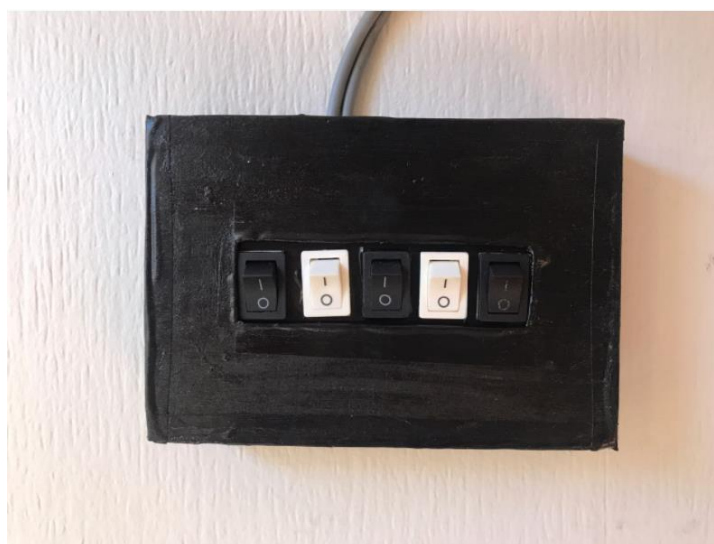


Fig. 7.2 Tablero sistema de interruptores en su etapa final.

A su vez, se implementó un pequeño tablero para la instalación del sensor de movimiento PIR. Esto con la finalidad de hacer más fácil su ubicación en las distintas zonas estratégicas dentro de la oficina de un docente y así, lograr captar de buena manera la presencia de éste. La imagen 7.3 muestra lo recién descrito.



Fig. 7.3 Tablero para el sensor de movimiento en su etapa final.

Así, finalmente en las imágenes 7.4 y 7.5 se muestra el sistema completo implementado en su etapa final.



Fig. 7.4 Sistema de presencia académica utilizando sensor de movimiento implementado en su etapa final.

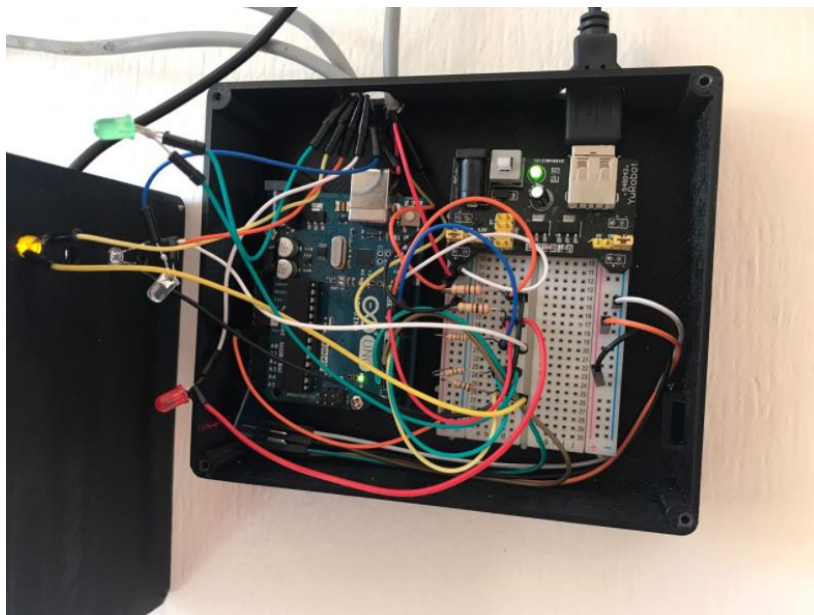


Fig. 7.5 Interior del sistema de presencia académica utilizando sensor de movimiento implementado en su etapa final.

Es importante mencionar que las imágenes recién vistas, corresponden al prototipo final y para el caso de la implementación en alguna oficina docente, el sistema de interruptores y el sensor de movimiento están diseñados para permanecer dentro de la oficina, a diferencia del tablero, que tiene la función de dar a conocer la disponibilidad docente a los alumnos por lo que debe permanecer fuera de ésta. Por otro lado, respecto al tablero sistema de interruptores se contempla la individualización de cada interruptor con la finalidad de que el docente pueda diferenciar la función que selecciona en cada caso. Además, se tiene pensado utilizar perforaciones de concreto con el fin de reducir al máximo la cantidad de cables a la vista.

Ahora, se procede a mostrar el resultado del sistema funcionando, se demostrará el caso del sistema informando que el docente está ocupado en su oficina. Lo anterior se demuestra en las imágenes 7.6 y 7.7 considerando, además, lo que se muestra en la página web implementada.



Fig. 7.6 Sistema de presencia académica informando a los alumnos que el docente se encuentra ocupado en la oficina.



Fig. 7.7 Página web de sistema de presencia académica informando a los alumnos que el docente se encuentra ocupado en la oficina.

Es importante notar el estado de los interruptores, en el cual sólo el segundo está en estado de encendido, enciende el LED amarillo que informa la presencia, pero no disponibilidad del docente en su oficina. Los costos totales asociados a la implementación de este sistema están presentados en la tabla A.2 del anexo A.

En cuanto al sistema de registro de asistencia a clases basado en la conexión de un microcontrolador y un lector de huellas dactilares, también se obtuvieron resultados satisfactorios, logrando así dejar registro escrito dentro de la tarjeta micro SD cada vez que el lector reconoce una huella previamente registrada. A continuación, se presentan las imágenes del sistema ya implementado.



Fig. 7.8 Sistema de registro de asistencia a clases mediante el uso de lector de huella dactilar implementado en su etapa final.

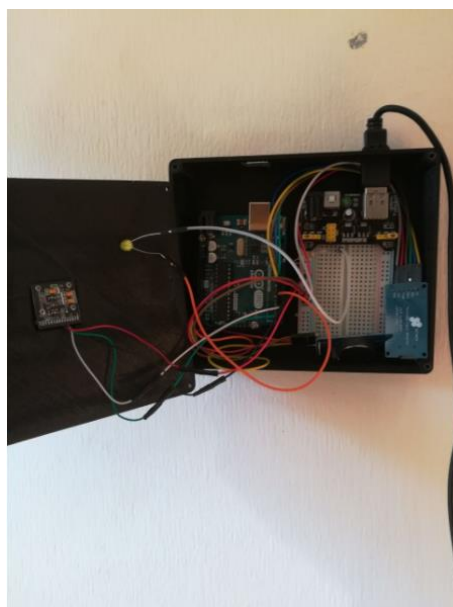


Fig. 7.9 Interior del sistema de registro de asistencia a clases mediante el uso de lector de huella dactilar implementado en su etapa final.

La implementación del sistema está pensada para dejar registros en una sala de clases, sin discriminar asignaturas ni docentes y se ubicaría en una posición estratégica dentro de cada una de ellas, con el fin de hacer fácil al alumno dejar registro de su asistencia.

A continuación, se presentan las imágenes del sistema ya en funcionamiento y los resultados de éste.



Fig. 7.10 Sistema de registro de asistencia a clases mediante el uso de lector de huella dactilar en funcionamiento a la espera de una huella dactilar.



Fig. 7.11 Sistema de registro de asistencia a clases mediante el uso de lector de huella dactilar en funcionamiento al momento de reconocer una huella dactilar dentro de su memoria (enciende la luz LED).



Fig. 7.12 Registro dentro de la memoria micro SD luego de reconocer ciertas huellas dactilares.

De la figura 7.12 se puede apreciar repeticiones consecutivas en los registros de un mismo alumno, esto debido a que la función del sistema implementado se limita solo a dejar registro de la asistencia de cada alumno en la memoria micro SD. Para evitar esta clase de eventos, se pensó el encendido del diodo LED y así dar a conocer al alumno que su huella ya fue registrada correctamente. Se propone como trabajo complementario, la implementación de un limitante de registro de huellas repetidas cada determinado rango de tiempo. Los costos totales asociados a la implementación del sistema están presentados en la tabla A.3 del anexo A.

8. Conclusiones

- El microcontrolador Arduino Uno modelo R3, en software y capacidad puede llegar a ser menor si se compara con la microcomputadora Raspberry Pi 2 modelo B, ya que posee menor memoria, velocidad de reloj y no tiene la capacidad de realizar múltiples tareas, además de no contar con un sistema operativo propio. Sin embargo, dependiendo del uso que se le desee dar, puede llegar a ser mucho más efectivo que la minicomputadora debido a su simplicidad, capacidad analógica y en tiempo real que la Raspberry no tiene. Esta flexibilidad le permite trabajar con casi cualquier tipo de sensor o chip sin la necesidad de softwares adicionales, como sucedería en el caso de la minicomputadora. De esta forma se concluye que, para fines de la implementación del proyecto, el microcontrolador Arduino en su funcionamiento y utilización es más simple y eficiente en comparación a la Raspberry.

- La corriente de alimentación que otorga el microcontrolador Arduino Uno modelo R3 fue insuficiente para satisfacer simultáneamente todos los módulos utilizados en la implementación de los dos sistemas, debido a esto, se optó por utilizar un sistema de alimentación independiente utilizando una fuente de alimentación externa de 3.3 V y 5.0 V y alimentándola mediante la conexión a la red eléctrica del puerto USB que ésta dispone.

- El excesivo uso de la memoria del Arduino que requiere el lector de tarjeta micro SD es un factor importante a considerar para la posible mejora del sistema de asistencia a clases mediante huellas dactilares. Se propone como posible trabajo futuro un reemplazo de dicho lector como sistema de registro de asistencia, por ejemplo, por algún sistema con un módulo WI-FI ESP8266 que envíe el registro a una base de datos dentro de la web a través de un servidor web, disponible para lectura del docente.

- Los costos presentados en el anexo A, asociados a la implementación de cada uno de los sistemas descritos en el documento son relativamente bajos, de lo que se extrae la escasa barrera económica en la puesta en marcha de éstos.

- Mediante la creación de los sistemas de bajo costo propuestos en el proyecto, se concluye que sí es posible mejorar el sistema académico de la Universidad de Concepción. Primero, en el ámbito de relación docente/alumno, la implementación del sistema de presencia académica constituye un importante aporte en cuanto a la transmisión de información desde el docente hacia los alumnos, la cual se realiza en tiempo real, lo que finalmente se traduce en una optimización significativa del tiempo y planificación para ambas partes. En segundo lugar, mediante la implementación del sistema de registro de asistencia a clases se mejora el sistema académico actual, específicamente respecto del desarrollo de las cátedras llevadas a cabo dentro de la institución, por cuanto la omisión de la actual consulta de asistencia individual por parte del docente se traduce en una importante optimización del tiempo disponible para impartir la asignatura.



Bibliografía

- [1] Thomas L. Floyd. “Fundamentos de Sistemas Digitales” Vol. 9, Pearson Educación S.A, Madrid, 2006.
- [2] Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss. “Sistemas Digitales Principios y Aplicaciones” 6° Edición, 2007.
- [3] Eben Upton and Garth Halfacree. “Raspberry Pi User guide”, John Wiley & Sons Ltd., The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom, 2012.
- [4] Isabel Gracia, Andrés Marzal. “Introducción a la programación con Python”. Departamento de Lenguajes y Sistemas Informáticos, Universitat Jaume I, 2003.
- [5] Guido Van Rossum. “The Python Tutorial”. Director of PythonLabs at Zope Corporation, febrero 2017. Disponible en: <http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>. (Last Access 05/05/2017).
- [6] Diego Gudiño C. “Diseño y construcción de un sistema de control de luces para una casa utilizando la placa Raspberry”. Sistemas Informáticos, Universidad Tecnológica Israel, Quito, Ecuador, 2014.
- [7] Óscar Torrente A. “Arduino, curso práctico de formación”. Alfaomega Grupo Editor, México, febrero 2013.
- [8] Hacedores - Maker Community. “Arduino o Raspberry Pi, ¿Cuál es la mejor herramienta para ti?”. Mayo, 2014. Disponible en: <http://hacedores.com/arduino-o-raspberry-pi-cual-es-la-mejor-herramienta-para-ti/>. (Last Access 08/05/2017).
- [9] Atmel. “ATMEL 8-BIT MICROCONTROLLER WITH 4/8/16/32KBYTES IN-SYSTEM PROGRAMMABLE FLASH”. Disponible en: http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf. (Last Access 08/05/2017).
- [10] Bluetooth Serial HC-06. “Product DataSheet”. Guangzhou HC Information Technology Co., Ltd. Disponible en: <https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf>. (Last Access 06/06/2017).

- [11] Emilio Lledó S. “Diseño de un sistema de control domótico basado en la plataforma Arduino”. Ingeniería Técnica en Informática de Sistemas. Universidad Politécnica de Valencia, Valencia, España, diciembre 2012.
- [12] Juan Almansa M. “Desarrollo e implementación utilizando Arduino y Zigbee con un sensor ultrasónico para control de nivel de llenado”. Departament d’Enginyeria Electrònica Elèctrica i Automàtica. Universitat Rovira i Virgili. Tarragona, España, septiembre 2013.
- [13] Javier Colomer B. “Aplicación Android para reproducción de audio en plataforma Arduino”. Escuela Técnica Superior de Ingenieros de Telecomunicación. Universidad Politécnica de Valencia, Valencia, España, julio 2015.
- [14] ESP8266 Serial Esp-01 WIFI Wireless. “ESP8266-01 datasheet”.
- [15] Arduino. “Software Arduino Web Editor”. Last Update 26/04/2017. Disponible en: <https://www.arduino.cc/en/Main/Software>. (Last Access 10/05/2017).
- [16] Arduino. “Arduino Uno Datasheet”. Disponible en: <http://arduino.cl/arduino-uno/>. (Last Access 08/05/2017).
- [17] “Sistemas biométricos”. Disponible en: <http://seginfitzelestrada.blogspot.cl/p/sistemas-biometricos.html>. (Last Access 28/09/2017).
- [18] SEGURED. “Reconocimiento por Voz”. Disponible en: <http://segured.com/reconocimiento-por-voz/>. (Last Access 28/09/2017).
- [19] Avril Hernández Briones. UNAM- Facultad de ingeniería biométrica informática. “Clasificación de los sistemas biométricos”. Disponible en: <http://redyseguridad.fi-p.unam.mx/proyectos/biometria/clasificacionsistemas/procesamientohuella.html>. (Last Access 02/10/2017).
- [20] TECNOSeguro.com. “¿Que es un detector de movimiento pasivo o PIR? Y ¿cómo funcionan los sensores de movimiento?”. Disponible en: <https://www.tecnoseguro.com/faqs/alarma/que-es-un-detector-de-movimiento-pasivo-o-pir.html>. (Last Access 03/10/2017).
- [21] Adafruit. “Adafruit Optical Fingerprint Sensor”. Disponible en: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-optical-fingerprint-sensor.pdf>. (Last Access 03/10/2017).

- [22] Adafruit. “PIR Motion Sensor”. Disponible en: <https://cdn-learn.adafruit.com/downloads/pdf/pir-passive-infrared-proximity-motion-sensor.pdf>. (Last Access 03/10/2017).
- [23] Fritzing. “Fritzing software download”. Disponible en: <http://fritzing.org/download/>. (Last Access 06/01/2018).
- [24] Maxim Integrated. “Reloj RTC DS3231 Datasheet”. Disponible en: <https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>. (Last Access 13/01/2018).
- [25] Luis Llamas. “Leer y escribir en una tarjeta SD o micro SD con Arduino”. Disponible en: <https://www.luisllamas.es/tarjeta-micro-sd-arduino/>. (Last Access 13/01/2018).
- [26] Luis Llamas. “Reloj y calendario en Arduino con los RTC DS1307 y DS3231”. Disponible en: <https://www.luisllamas.es/reloj-y-calendario-en-arduino-con-los-rtc-ds1307-y-ds3231/>. (Last Access 13/01/2018).
- [27] Punto flotante S.A 2017. “Manual del usuario sensor de movimiento PIR HC-SR501”. Disponible en: <https://puntoflotante.net/MANUAL-DEL-USUARIO-SENSOR-DE-MOVIMIENTO-PIR-HC-SR501.pdf>. (Last Access 17/01/2018).
- [28] Monografias.com. “Aplicación de nuevas tecnologías al sistema electoral – Biometría y voto electrónico”. Disponible en <http://www.monografias.com/trabajos82/biometria-y-voto-electronico/biometria-y-voto-electronico2.shtml>. (Last Access 17/01/2018).
- [29] MDN Web Docs. “Cómo funciona la web”. Disponible en: https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/C%C3%B3mo_funciona_la_Web. (Last Access 09/02/2018).

Anexo A. Tablas complementarias

Tabla A.1 Comandos AT utilizados en la configuración del Módulo WI-FI ESP8266.

Comando AT	Función
AT	Habilita el módem para recibir los comandos AT
AT+GMR	Muestra la versión del firmware que tiene cargado el módulo
AT+RST	Reinicia el módulo
AT+CWJAP="nombre de la red WIFI", "contraseña de la red WIFI"	Permite la conexión a una red WIFI específica
AT+CWMODE=?	Muestra el punto de acceso
AT+CWMODE=3	Convierte el módulo en modo estación y como punto de acceso
AT+CWLAP	Muestra las redes WIFI cercanas
AT+CIFSR	Muestra la dirección IP que se le asignó al módem
AT+CIPMUX=1	habilita el módem para permitir múltiples conexiones
AT+CIPSERVER=1,80	Permite al módulo funcionar como servidor web
AT+CWSAP="nombre de red", "contraseña"	Permite definir el nombre y la contraseña de la red para el punto de acceso

Tabla A.2 Costos aproximados del sistema de presencia académica en Chile.

Componente	Precio aproximado
Arduino Uno versión R3	\$ 9.800
Módulo WI-FI 8266	\$ 6.500
Sensor PIR HC-SR501	\$ 3.500
Diodo LED (x5)	\$ 500
Cables	\$ 4.000
Fuente de alimentación para protoboard	\$ 3.600
Protoboard 400 puntos	\$ 3.000
TOTAL	\$ 30.900

Tabla A.3 Costos aproximados del sistema de registro de asistencia a clases en Chile.

Componente	Precio aproximado
Arduino Uno versión R3	\$ 9.800
Lector de huella dactilar Adafruit	\$ 28.000
Módulo reloj RTC DS3231	\$ 4.200
Adaptador tarjeta micro SD Catalex v1.0	\$ 4.900
Tarjeta micro SD	\$ 3.000
Adaptador tarjeta micro SD/USB	\$ 2.990
Diodo LED (x1)	\$ 100
Cables	\$ 4.000
Fuente de alimentación para protoboard	\$ 3.600
Protoboard 400 puntos	\$ 3.000
TOTAL	\$ 63.590

Anexo B. Trabajo complementario

B.1. Conexión microcontrolador – módulo Bluetooth HC-06

Se establece la conexión serie entre el microcontrolador Arduino Uno R3 con el módulo Bluetooth HC-06, con el fin de permitirle al dispositivo la posibilidad de conectarse vía Bluetooth a cualquier dispositivo que cuente con este tipo de conexión, para fines del proyecto se experimentará la conexión con un teléfono móvil.

Como el módulo Bluetooth es básicamente un nodo BT conectado a una interface serie, se podría en principio conectar los pines Rx y Tx de éste a los equivalentes de Arduino en los pines 0 y 1 digitales, más que cruzarlos (BT Tx a Arduino Rx y BT Rx a Arduino Tx). Sin embargo, esta configuración no es recomendable puesto que los pines 0 y 1 del Arduino se utilizan en la comunicación serie con un computador a través del puerto USB y por tanto, si se utilizan para la comunicación con el módulo Bluetooth, se perdería la posibilidad de conexión con un computador.

Por lo anterior, se utilizan otro par de pines del Arduino, aunque para ello se debe importar una librería que habilite la comunicación serie con otros pines, como es la librería denominada **SoftwareSerial** (disponible en el Software a utilizar para la programación). Finalmente, como el módulo HC-06 funciona a 3,3 V, no es necesario agregar divisores de tensión para acceder a los pines de éste, por lo tanto, la conexión se establece conectando tensión y GND al módulo para posteriormente proceder a conectar directamente los pines Tx y Rx del módulo a los pines 2 y 4 respectivamente del Arduino. Para comprobar que la conexión fue realizada con éxito, se debe verificar que el LED del módulo HC-06 parpadea continuamente, esto indica que está conectado aunque no está pareado o vinculado. Al conectar algo vía Bluetooth al módulo, dicha luz se quedará fija sin parpadear, lo cual indica que la conexión Bluetooth fue exitosa. La correcta conexión del dispositivo y el módulo se presenta en el anexo C.2. El código necesario para realizar la conexión a nivel de software entre el microcontrolador y el módulo se presenta en el anexo D.1.

Anexo C. Diagramas

C.1. Diagramas de conexión

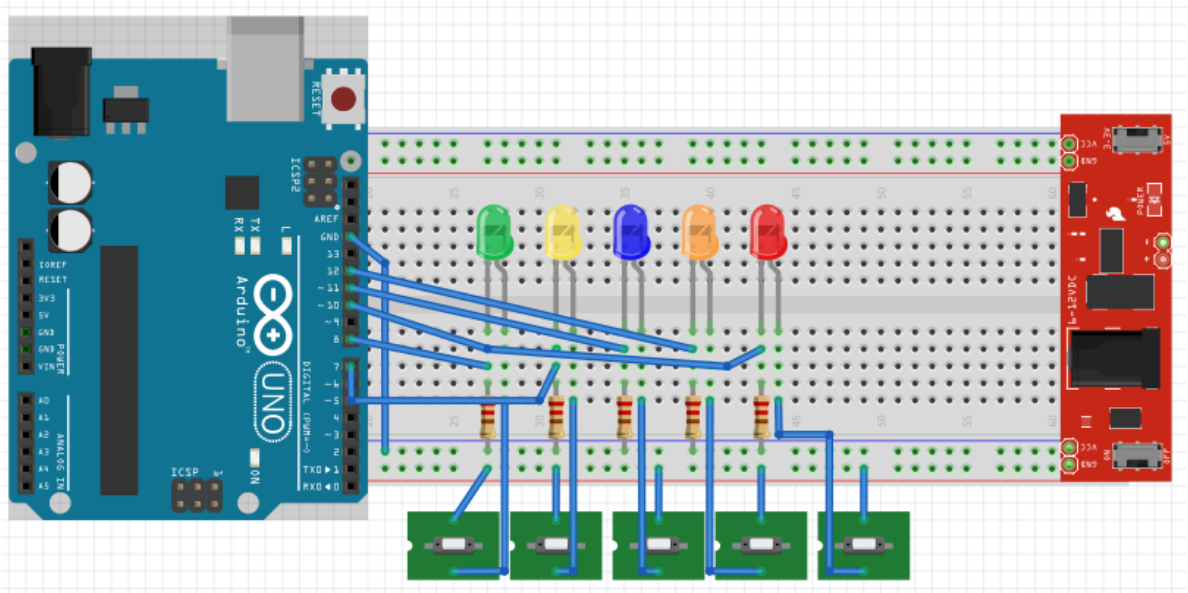


Fig. C.1 Conexión entre el microcontrolador Arduino Uno R3 y los diodos LED con sus respectivos interruptores.

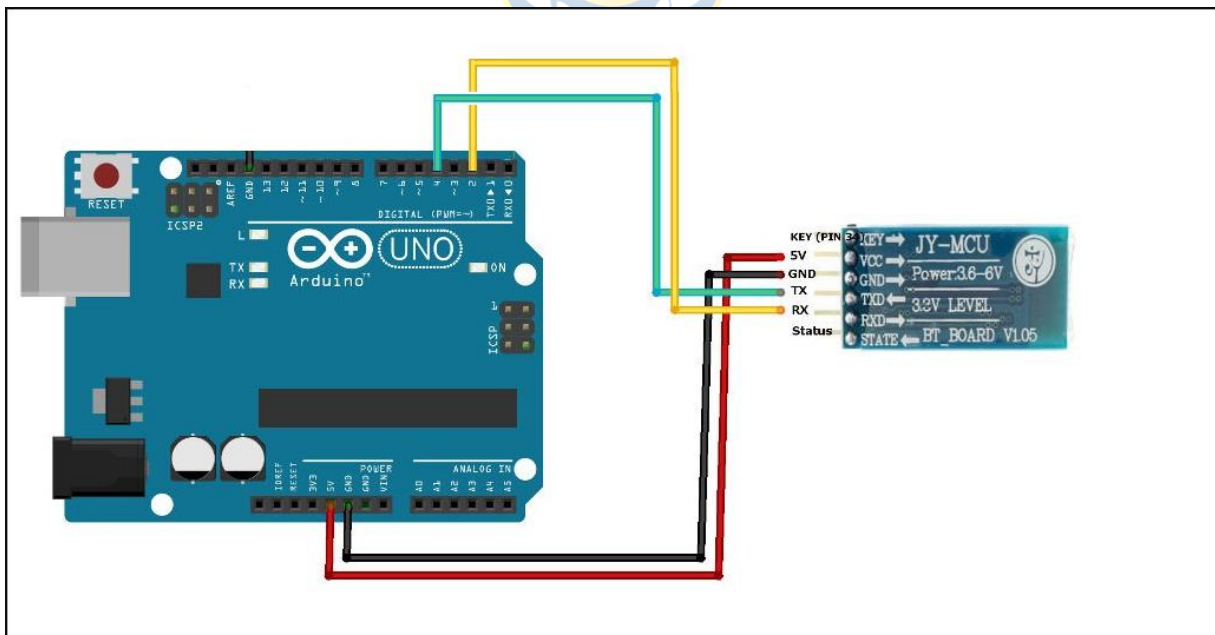


Fig. C.2 Conexión entre el Microcontrolador Arduino Uno R3 y el módulo Bluetooth HC-06.

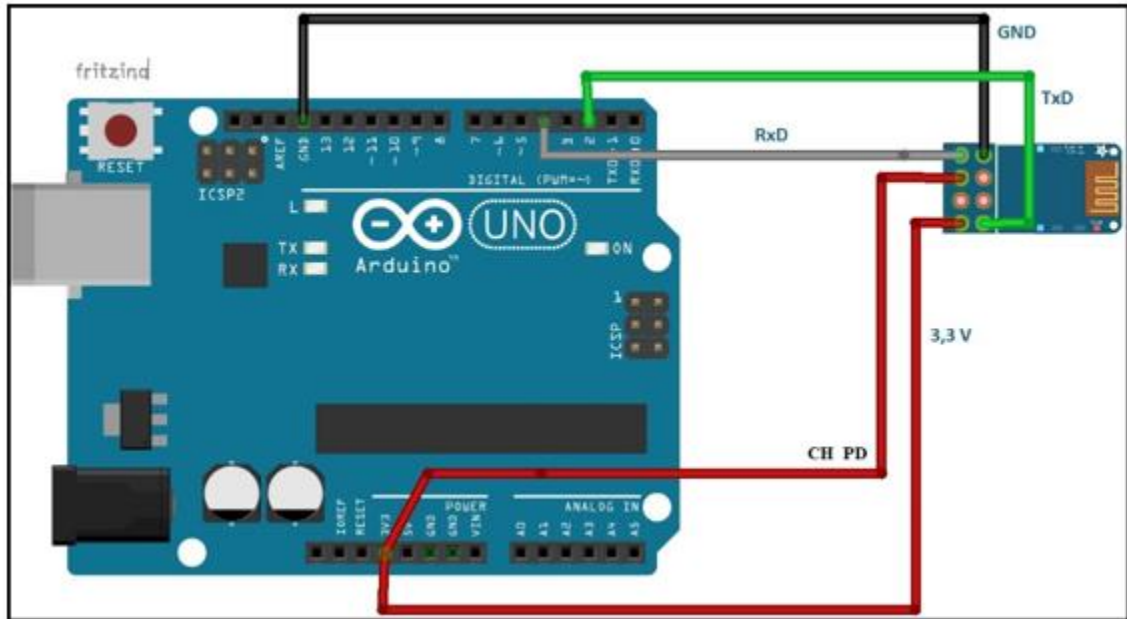


Fig. C.3 Conexión entre el Microcontrolador Arduino Uno R3 y el módulo WI-FI ESP8266.

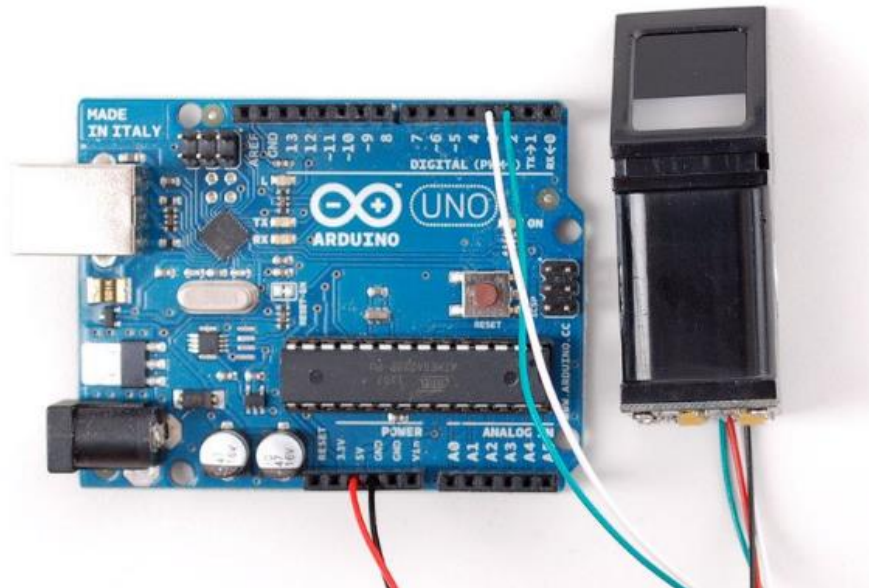


Fig. C.4 Conexión entre el Microcontrolador Arduino Uno R3 y el lector de huellas digitales.

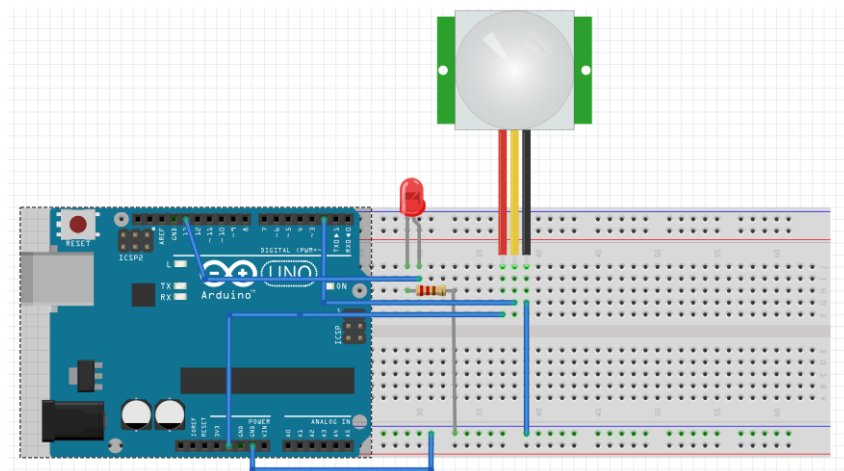


Fig. C.5 Conexión entre el Microcontrolador Arduino Uno R3 y sensor de movimiento PIR HC-SR501.

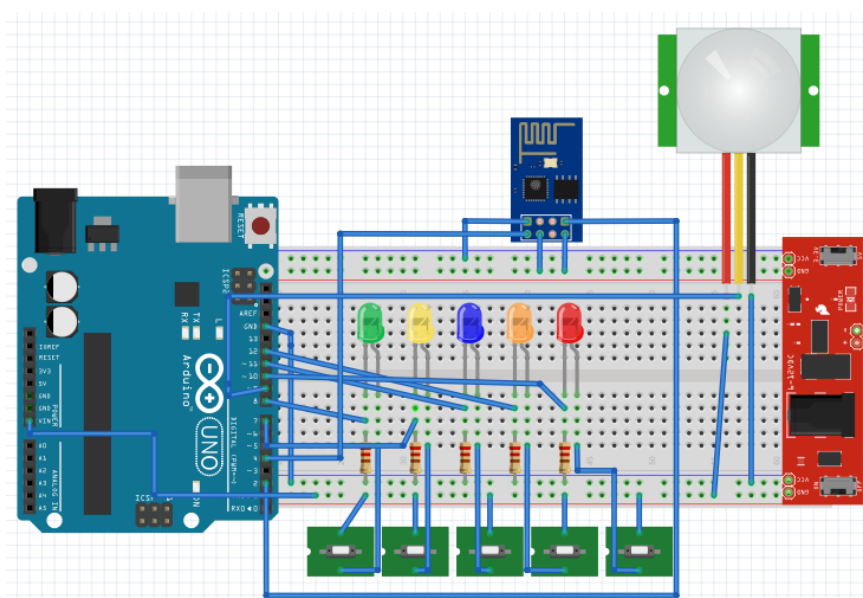


Fig. C.6 Conexión completa del sistema de presencia académica utilizando sensor de movimiento PIR.

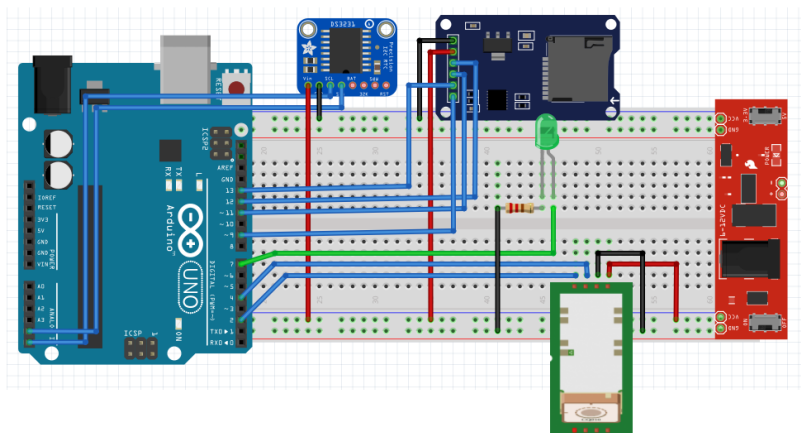


Fig. C.7 Conexión completa del sistema de registro de asistencia utilizando lector de huella dactilar.

Anexo D. Código

D.1. Rutina conexión Arduino – Módulo HC-06

```
#include <SoftwareSerial.h>
SoftwareSerial BT1(4,2);
void setup() {

    Serial.begin(9600);
    Serial.println("Enter AT commands:");
    BT1.begin(9600);
}

void loop() {

    if (BT1.available())
        Serial.write(BT1.read());

    if (Serial.available())
    {String S = GetLine();
    BT1.print(S);
    Serial.println("--->" + S);
    }
}

String GetLine()
{ String S= "";
  if (Serial.available())
  { char c= Serial.read(); ;
    while ( c!= '\n')
    { S = S + c;
      delay(25);
      c=Serial.read();
    }
    return( S + '\n');
  }
}
```



D.2. Rutina conexión Arduino – Módulo ESP8266-01

```
#include <SoftwareSerial.h>
SoftwareSerial BT1(3, 2); // RX | TX

void setup()
{ Serial.begin(19200);
  BT1.begin(19200);
}

void loop()
{ String B= ".";
  if (BT1.available())
  { char c = BT1.read();
    Serial.print(c);
  }
  if (Serial.available())
  { char c = Serial.read();
    BT1.print(c);
  }
}
```



D.3. Rutina conexión Arduino – Lector huellas dactilares (enrolar)

```

#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

uint8_t id;

uint8_t getFingerprintEnroll();

SoftwareSerial Huellas(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Huellas);

void setup()
{
  Serial.begin(9600);
  Serial.println("Buscando Sensor...");

  // ajuste de velocidad de datos para el puerto serie del sensor
  finger.begin(57600);
  if (finger.verifyPassword()) {
    Serial.println("Sensor de huellas dactilares encontrado!");
  } else {
    Serial.println("Sensor no encontrado");
    while (1);
  }
}

uint8_t readnumber(void) {
  uint8_t num = 0;
  boolean validnum = false;
  while (1) {
    while (! Serial.available());
    char c = Serial.read();
    if (isdigit(c)) {
      num *= 10;
      num += c - '0';
      validnum = true;
    } else if (validnum) {
      return num;
    }
  }
}

void loop()
{
  Serial.println("Listo para registrar una huella digital! Por favor escriba el ID# que desea guardar para este
dedo...");
  id = readnumber();
  Serial.print("Enrolando ID #");
  Serial.println(id);
}

```



```

while (! getFingerprintEnroll() );
}

uint8_t getFingerprintEnroll() {

int p = -1;
Serial.print("Esperando por un dedo valido para enrolar como #"); Serial.println(id);
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Imagen tomada");
    break;
  case FINGERPRINT_NOFINGER:
    Serial.println(".");
    break;
  case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Error de Comunicacion");
    break;
  case FINGERPRINT_IMAGEFAIL:
    Serial.println("Error de generacion de imagen");
    break;
  default:
    Serial.println("Error desconocido");
    break;
  }
}

// OK success!

p = finger.image2Tz(1);
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Imagen convertida");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Imagen demasiado borrosa");
    return p;
  case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Error de Comunicacion");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("No se pudieron encontrar las características de huella");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("No se pudieron encontrar las características de huella");
    return p;
  default:
    Serial.println("Error desconocido");
    return p;
}
}

```



```

Serial.println("Remover Dedo");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
  p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Colocar el mismo dedo de nuevo");
while (p != FINGERPRINT_OK) {
  p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Imagen tomada");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.print(".");
      break;
    case FINGERPRINT_PACKETRECEIVEERR:
      Serial.println("Error de Comunicacion");
      break;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Error de generacion de imagen");
      break;
    default:
      Serial.println("Error desconocido");
      break;
  }
}

// OK success!
p = finger.image2Tz(2);
switch (p) {
  case FINGERPRINT_OK:
    Serial.println("Imagen convertida");
    break;
  case FINGERPRINT_IMAGEMESS:
    Serial.println("Imagen demasiado borrosa");
    return p;
  case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Error de comunicacion");
    return p;
  case FINGERPRINT_FEATUREFAIL:
    Serial.println("No se pudieron encontrar las características de huella");
    return p;
  case FINGERPRINT_INVALIDIMAGE:
    Serial.println("No se pudieron encontrar las características de huella");
    return p;
  default:
    Serial.println("Error desconocido");
    return p;
}

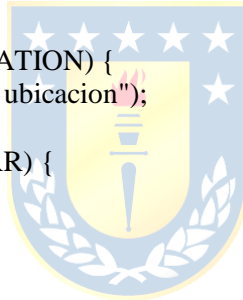
```



```

// OK converted!
Serial.print("Creando modelo para #"); Serial.println(id);
p = finger.createModel();
if (p == FINGERPRINT_OK) {
  Serial.println("Impresiones coincidentes!");
} else if (p == FINGERPRINT_PACKETRECIIVEERR) {
  Serial.println("Error de comunicacion");
  return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
  Serial.println("Huellas dactilares no coincidentes");
  return p;
} else {
  Serial.println("Error desconocido");
  return p;
}
Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
  Serial.println("Almacenado!");
} else if (p == FINGERPRINT_PACKETRECIIVEERR) {
  Serial.println("Error de comunicacion");
  return p;
} else if (p == FINGERPRINT_BADLOCATION) {
  Serial.println("No se pudo guardar en esa ubicacion");
  return p;
} else if (p == FINGERPRINT_FLASHERR) {
  Serial.println("Error al escribir en flash");
  return p;
} else {
  Serial.println("Error desconocido");
  return p;
}
}
}

```



D.4. Rutina conexión Arduino – Lector huellas dactilares (datalogger)

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
#include <Wire.h>
#include "RTClib.h"
#include <SD.h>
```

```
File archivo;
RTC_DS3231 rtc;
```

```
int getFingerprintIDez();
String daysOfTheWeek[7] = { "Domingo", "Lunes", "Martes", "Miercoles", "Jueves", "Viernes",
"Sabado" };
```

```
SoftwareSerial Huellas(2, 3);
```

```
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Huellas);
```

```
void setup()
{
  Serial.begin(9600);
  pinMode(7, OUTPUT);
  if (!rtc.begin()) {
    while (1);
  }
  if (!SD.begin(9))
  {
    return;
  }
  finger.begin(57600);

  if (finger.verifyPassword()) {
  } else {
    while (1);
  }
}
```



```
void printDate(DateTime date)
{
  archivo.print(date.year(), DEC);
  archivo.print('/');
  archivo.print(date.month(), DEC);
  archivo.print('/');
  archivo.print(date.day(), DEC);
  archivo.print(" ");
  archivo.print(daysOfTheWeek[date.dayOfTheWeek()]);
  archivo.print(" ");
}
```

```

archivo.print(date.hour(), DEC);
archivo.print(':');
archivo.print(date.minute(), DEC);
archivo.print(':');
archivo.print(date.second(), DEC);
archivo.println();
}

```

```
void loop()
```

```

{
  getFingerprintIDez();

  if (getFingerprintIDez() > -1) {
    digitalWrite(7,HIGH);
    delay(2000);
    digitalWrite(7,LOW);
  }
  delay(50);
}

```

```

uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      break;
    case FINGERPRINT_NOFINGER:
      return p;
    case FINGERPRINT_PACKETRECEIVEERR:
      return p;
    case FINGERPRINT_IMAGEFAIL:
      return p;
    default:
      return p;
  }
}

```



```
// OK logrado!
```

```

p = finger.image2Tz();
switch (p) {
  case FINGERPRINT_OK:
    break;
  case FINGERPRINT_IMAGEMESS:
    return p;
  case FINGERPRINT_PACKETRECEIVEERR:
    return p;
  case FINGERPRINT_FEATUREFAIL:
    return p;
  case FINGERPRINT_INVALIDIMAGE:

```

```

    return p;
default:
    return p;
}

// OK convertido!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    return p;
} else {
    return p;
}
// coincidencia encontrada!
Serial.print("Encontrado! ID #"); Serial.println(finger.fingerID);
//Serial.print(" with confidence of "); Serial.println(finger.confidence);
}

// devuelve -1 si falla, de lo contrario devuelve ID #
int getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.image2Tz();
    if (p != FINGERPRINT_OK) return -1;

    p = finger.fingerFastSearch();
    if (p != FINGERPRINT_OK) return -1;

    // coincidencia encontrada!
    Serial.print("Econtrado! ID #"); Serial.println(finger.fingerID);

    if (finger.fingerID == 2) {
        archivo = SD.open("datos.txt", FILE_WRITE);

        if (archivo){
            archivo.print("Felipe Rogers:");
            DateTime now = rtc.now();
            printDate(now);
            archivo.close();
        }
        else{
            Serial.println("el archivo no se pudo abrir");
        }
    }
}

```



```

}

if (finger.fingerID == 1) {
archivo = SD.open("datos.txt", FILE_WRITE);

if (archivo){
archivo.print("Alfonso Miranda:");
DateTime now = rtc.now();
printDate(now);
archivo.close();

}
else{
Serial.println("el archivo no se pudo abrir");
}
}

if (finger.fingerID == 7) {
archivo = SD.open("datos.txt", FILE_WRITE);

if (archivo){
archivo.print("Bryan Rogers:");
DateTime now = rtc.now();
printDate(now);
archivo.close();

}
else{
Serial.println("el archivo no se pudo abrir");
}
}

if (finger.fingerID == 11) {
archivo = SD.open("datos.txt", FILE_WRITE);

if (archivo){
archivo.print("Cristian Sagredo :");
DateTime now = rtc.now();
printDate(now);
archivo.close();

}
else{
Serial.println("el archivo no se pudo abrir");
}
}

if (finger.fingerID == 3) {
archivo = SD.open("datos.txt", FILE_WRITE);

```



```
if (archivo){
    archivo.print("Pedro Carrasco :");
    DateTime now = rtc.now();
    printDate(now);
    archivo.close();

}
else{
    Serial.println("el archivo no se pudo abrir");
}

if (finger.fingerID == 9) {
archivo = SD.open("datos.txt", FILE_WRITE);

if (archivo){
    archivo.print(" Alfonso Sufan:");
    DateTime now = rtc.now();
    printDate(now);
    archivo.close();

}
else{
    Serial.println("el archivo no se pudo abrir");
}

return finger.fingerID;
}
```



D.5. Rutina conexión Arduino – Sensor infrarrojo PIR HC-SR501

```
void setup()
{
  pinMode(13, OUTPUT);
  pinMode(2, INPUT);
}

void loop()
{
  int value= digitalRead(2);

  if (value == HIGH)
  {
    digitalWrite(13, HIGH);
    delay(50);
    digitalWrite(13, LOW);
  }
  else
  {
    digitalWrite(13, LOW);
  }
}
```



D.6. Rutina del cuerpo principal de la página web

```
<html>
  <head>
    <meta charset="utf8">
    <link rel="stylesheet" type="text/css" href="assets/style.css" />
    <title>botones arduino esp8266 html</title>
  </head>
  <body>
    <h1>
      SISTEMA DE PRESENCIA ACADÉMICA UDEC
    </h1>
    <center>
      
    </center>

    <p id="status_busy" style="font-size:30px;text-align:center;width:100%"></p>
    <p id="status_state" style="font-size:26px;text-align:center;width:100%;color:#232323"></p>

    <script src="https://code.jquery.com/jquery-3.2.1.min.js"></script>
    <script src="assets/scripts/sensor.js">
    </script>
  </body>
</html>
```

D.7. Rutina de condiciones de la página web

```
$(document).ready(function()
{
  getSensorData();
  setInterval(function() {
    getSensorData();
  }, 5000);
})

function getSensorData() {
$.get("/api/sensor/busy", function(data) {
  if (data.busy) {
    $('#status_busy')
    .text('Profesor presente en la oficina.')
    .css({
      color: 'green'
    });
  } else {
    $('#status_busy')
    .text('Profesor no esta presente en la oficina.')
    .css({
      color: 'red'
    });
  }
  var stateText = '';
  switch(data.state) {
    case 1:
      stateText = 'Disponible en oficina';
      break;
    case 2:
      stateText = 'Ocupado en oficina';
      break;
    case 3:
      stateText = 'En clases';
      break;
    case 4:
      stateText = 'Vuelvo pronto';
      break;
    case 5:
```



```

stateText = 'Ausente por el día';
break;
default:
stateText = 'Ausente por el día';
break;
}
$('#status_state')
.text(stateText);
});
}

```

D.8. Rutina comunicación Microcontrolador – Servidor web

```

#include <SoftwareSerial.h>
SoftwareSerial SerialESP8266(4,2); // RX, TX

```

```

//variables para enviar al servidor
String server = "felipe.saidgeek.io";
int variable1;
int valor=0;
int x=0;
int state;

```

```
String cadena="";
```

```
void setup() {
```

```

SerialESP8266.begin(9600);
Serial.begin(9600);
SerialESP8266.setTimeout(2000);
pinMode(9, INPUT);
pinMode(12,INPUT);
pinMode(7,INPUT);
pinMode(8,INPUT);
pinMode(11,INPUT);
pinMode(10,INPUT);
//Verificamos si el ESP8266 responde
SerialESP8266.println("AT");
if(SerialESP8266.find("OK"))
  Serial.println("Respuesta AT correcto");
else
  Serial.println("Error en ESP8266");

```



```

//-----Configuración de red-----//
//ESP8266 en modo estación (puesto que se efectuará la conexión a una red existente)
SerialESP8266.println("AT+CWMODE=1");
if(SerialESP8266.find("OK"))
  Serial.println("ESP8266 en modo Estacion");

//Conexión a una red wifi
SerialESP8266.println("AT+CWLAP=\"VTR-3755555\", \"ytc3vvrhYvyg\");
Serial.println("Conectandose a la red ...");
SerialESP8266.setTimeout(10000); //Aumentar si demora la conexion
if(SerialESP8266.find("OK"))
  Serial.println("WIFI conectado");
else
  Serial.println("Error al conectarse en la red");
SerialESP8266.setTimeout(2000);
//Se deshabilita las conexiones multiples
SerialESP8266.println("AT+CWMUX=0");
if(SerialESP8266.find("OK"))
  Serial.println("Multiconexiones deshabilitadas");

//-----fin de configuración-----

delay(1000);
}
void loop() {

//--- programa-----
// Variable que contiene el estado del sensor de movimiento
variable1= digitalRead(9);
// Estados de los diodos LED
if (digitalRead(8)){
  state=1;
  }
if (digitalRead(7)){
  state=2;
  }
if (digitalRead(11)){
  state=3;
  }
if (digitalRead(12)){
  state=4;
  }
if (digitalRead(10)){
  state=5;
  }
}

```



```
//-----se envian las variables al servidor-----
```

```
//Se establece la conexión con el servidor:
```

```
SerialESP8266.println("AT+CIPSTART=\\"TCP\\","\\" + server + "\",80");
if( SerialESP8266.find("OK"))
{
  Serial.println();
  Serial.println();
  Serial.println();
  Serial.println("ESP8266 conectado con el servidor...");
}
```

```
//Se arma el encabezado de la petición http
```

```
String peticiónHTTP= "GET /api/sensor/push?busy=";
peticiónHTTP=peticiónHTTP+String(variable1)+"&state="+String(state)+" HTTP/1.1\r\n";
peticiónHTTP=peticiónHTTP+"Host: project.felipe.saidgeek.io\r\n\r\n";
```

```
//Se envía el tamaño en caracteres de la petición http:
```

```
SerialESP8266.print("AT+CIPSEND=");
SerialESP8266.println(peticiónHTTP.length());
```

```
//Se espera a ">" para enviar la petición http
```

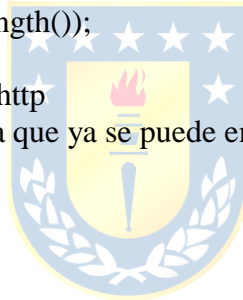
```
if(SerialESP8266.find(">")) // ">" indica que ya se puede enviar la petición http
```

```
{
  Serial.println("Enviando HTTP . . .");
  SerialESP8266.println(peticiónHTTP);
  if( SerialESP8266.find("SEND OK"))
  {
    Serial.println("Petición HTTP enviada:");
    Serial.println();
    Serial.println(peticiónHTTP);
    Serial.println("Esperando respuesta...");
  }
}
```

```
boolean fin_respuesta=false;
long tiempo_inicio=millis();
cadena="";
```

```
while(fin_respuesta==false)
```

```
{
  while(SerialESP8266.available(>0)
  {
    char c=SerialESP8266.read();
    Serial.write(c);
    cadena.concat(c); //Se guarda la respuesta en el string "cadena"
  }
  // Se finaliza si la respuesta es mayor a 800 caracteres
  if(cadena.length(>800)
```



```

{
  Serial.println("La respuesta a excedido el tamaño maximo");

  SerialESP8266.println("AT+CIPCLOSE");
  if( SerialESP8266.find("OK"))
    Serial.println("Conexion finalizada");
  fin_respuesta=true;
}
if((millis()-tiempo_inicio)>10000) //Se finaliza si ya han transcurrido 10 seg
{
  Serial.println("Tiempo de espera agotado");
  SerialESP8266.println("AT+CIPCLOSE");
  if( SerialESP8266.find("OK"))
    Serial.println("Conexion finalizada");
  fin_respuesta=true;
}
if(cadena.indexOf("CLOSED")>0) // Si se recibe un CLOSED significa que ha finalizado la
respuesta
{
  Serial.println();
  Serial.println("Cadena recibida correctamente, conexion finalizada");
  fin_respuesta=true;
}
}

}
else
{
  Serial.println("No se ha podido enviar HTTP.....");
}
}
}
else
{
  Serial.println("No se ha podido conectar con el servidor");
}

//-----

delay(1000); //pausa de 1 seg antes de conectarse nuevamente al servidor
}

```



Anexo E. Datasheets

E.1. Arduino Uno R3



Arduino Uno SMD Rev3

The Arduino Uno SMD R3 is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, 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 Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip.

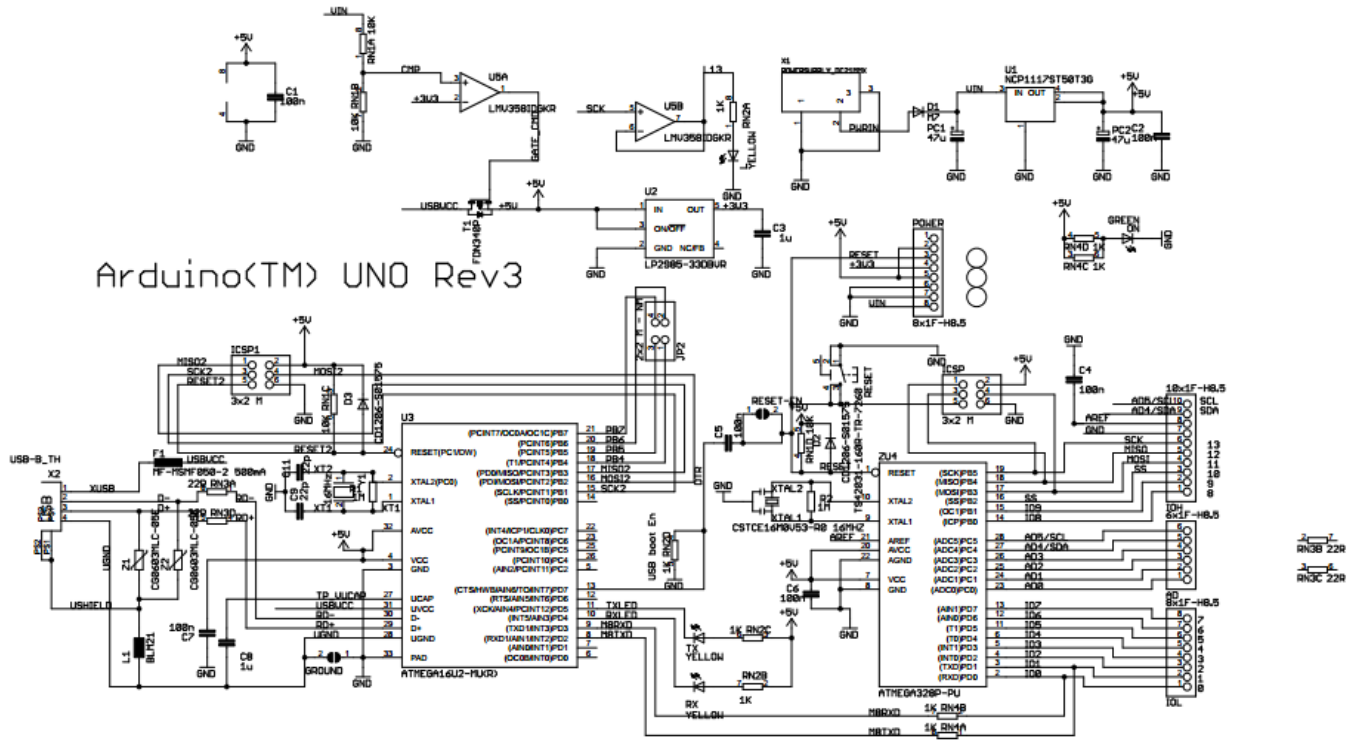
Additional features coming with the R3 version are:

- ATmega16U2 instead 8U2 as USB-to-Serial converter.
- 1.0 pinout: added SDA and SCL pins for TWI communication placed near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board and the second one is a not connected pin, that is reserved for future purposes.
- stronger RESET circuit.

"Uno" means "One" in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Technical Specifications

Microcontroller	ATmega328
Operating Voltage	5V
Supply Voltage (recommended)	7-12V
Maximum supply voltage (not recommended)	20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz



E.2. Módulo WI-FI ESP8266-01

ESP8266 Serial Esp-01 WIFI Wireless



Building the gcc toolchain

have a look at the [github wiki](https://github.com/esp8266/esp8266-wiki/wiki) <https://github.com/esp8266/esp8266-wiki/wiki>

Code examples

have a look at the [github wiki](https://github.com/esp8266/esp8266-wiki/wiki) <https://github.com/esp8266/esp8266-wiki/wiki>

Running the module

The modules pins only allow 3.3v (use a multi meter to check your serial lines if you are not sure)

Connect CH_PD to VCC to make it boot

Uploading code

The modules pins only allow 3.3v (use a multi meter to check your serial lines if you are not sure)

see <https://github.com/esp8266/esp8266-wiki/wiki/Uploading>

links

Internal space links

https://git.nurdspace.lan/esp8266/led_example/source/

External

SDK documentation (all chinese) [DOCS](#)

VM file [\[1\]](#) Password: i90l

Forum about the module <http://www.esp8266.com/>

[seeedstudio.com/depot/WiFi-Serial-Transceiver-Module-w-ESP8266-p-1994.html](https://www.seeedstudio.com/depot/WiFi-Serial-Transceiver-Module-w-ESP8266-p-1994.html)

ESP8266 ROM Bootloader utility <https://github.com/themadinventor/esp8266>

Datasheet

[English Datasheet](#)

<http://www.seeedstudio.com/document/pdf/ESP8266%20Specifications%20Chinese.pdf> (Chinese)

Introduction

Technical Overview

ESP8266 is a complete and self-contained Wi-Fi network solutions that can carry software applications, or through Another application processor uninstall all Wi-Fi networking capabilities. ESP8266 when the device is mounted and as the only application of the application processor, the flash memory can be started directly from an external Move. Built-in cache memory will help improve system performance and reduce memory requirements. Another situation is when wireless Internet access assume the task of Wi-Fi adapter, you can add it to any microcontroller-based design, the connection is simple, just by SPI / SDIO interface or central processor AHB bridge interface. Processing and storage capacity on ESP8266 powerful piece, it can be integrated via GPIO ports sensors and other applications specific equipment to achieve the lowest early in the development and operation of at least occupy system resources. The ESP8266 highly integrated chip, including antenna switch balun, power management converter, so with minimal external circuitry, and includes front-end module, including the entire solution designed to minimize the space occupied by PCB. The system is equipped with ESP8266 manifested leading features are: energy saving VoIP quickly switch between the sleep / wake patterns, with low-power operation adaptive radio bias, front-end signal processing functions, troubleshooting and radio systems coexist characteristics eliminate cellular / Bluetooth / DDR / LVDS / LCD interference.

Characteristics

802.11 b / g / n

Wi-Fi Direct (P2P), soft-AP

Built-in TCP / IP protocol stack

Built-in TR switch, balun, LNA, power amplifier and matching network

Built-in PLL, voltage regulator and power management components

802.11b mode + 19.5dBm output power

Built-in temperature sensor

Support antenna diversity

off leakage current is less than 10uA

Built-in low-power 32-bit CPU: can double as an application processor

SDIO 2.0, SPI, UART

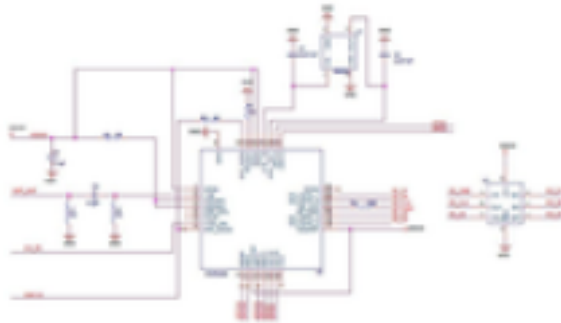
STBC, 1x1 MIMO, 2x1 MIMO

A-MPDU, A-MSDU aggregation and the 0.4 Within wake

2ms, connect and transfer data packets

standby power consumption of less than 1.0mW (DTIM3)

Schema



Ultra-low power technology

ESP8266 specifically for mobile devices, wearable electronics and networking applications design and make the machine to achieve the lowest energy consumption, together with several other patented technology. This energy-efficient construction in three modes: active mode, sleep mode and deep sleep mode type. When ESP8266 using high-end power management technology and logic systems to reduce non-essential functions of the power conversion regulate sleep patterns and work modes, in sleep mode, it consumes less than the current 12uA, is connected, it consumes less power to 1.0mW (DTIM = 3) or 0.5mW (DTIM = 10). Sleep mode, only calibrated real-time clock and watchdog in working condition. Real-time clock can be programmed to wake ESP8266 within a specific period of time. Through programming, ESP8266 will automatically wake up when detected certain to happen. ESP8266 automatic wake-up in the shortest time, this feature can be applied to the SOC for mobile devices, so before you turn Wi-Fi SOC are in a low-power standby mode. To meet the power requirements of mobile devices and wearable electronics products, ESP8266 at close range when the PA output power can be reduced through software programming to reduce overall power consumption in order to adapt to different applications.

Maximum integration

ESP8266 integrates the most critical components on the board, including power management components, TR switch, RF balun, a peak power of + 25dBm of PA, therefore, ESP8266 only guarantee the lowest BOM cost, and easy to be embedded in any system. ESP8266 BOM is the only external resistors, capacitors, and crystal.

ESP8266 application subject

Smart Power Plug

Home Automation

mesh network

industrial wireless control

Baby Monitor

Network Camera

Power

The following data are based on a 3.3V power supply, ambient temperature 25C and use the internal regulator measured. [1] All measurements are made in the absence of the SAW filter, the antenna interface is completed. [2] all transmit data based on 90% duty cycle, continuous transmission mode in the measured.

Mode	Min	Typical	Max	Units
802.11b, CCK 1Mbps, POUT=+19.5dBm		215		mA
802.11b, CCK 11Mbps, POUT=+18.5dBm		197		mA
802.11g, OFDM 54Mbps, POUT=+16dBm		145		mA
802.11n, MCS7, POUT =+14dBm		135		mA
802.11b, packet size of 1024 bytes, -80dBm		60		mA
802.11b, packet size of 1024 bytes, -70dBm		60		mA
802.11b, packet size of 1024 bytes, -65dBm		62		mA
Standby		0.9		uA
Deep sleep		10		mA
Saving mode DTIM 1		1.2		mA
Saving mode DTIM 3		0.86		mA
Shutdown		0.5		uA

RF specifications

The following data is at room temperature, the voltage of 3.3V and 1.1V, respectively, when measured

Description	Min	Typical	Max	Units
Input Frequency	2412		2484	MHz
Input resistance		50		Ω
Input reflection			-10	dB
At 72.2Mbps, PA output power	14	15	16	dBm
11b mode, PA output power	17.5	18.5	19.5	dBm
Sensitivity				
CCK, 1Mbps <small>(CC^2DFSK)</small>		-98		dBm
CCK, 11Mbps <small>(CC^2DFSK)</small>		-91		dBm
6Mbps (1/2 BPSK) <small>(CC^2DFSK)</small>		-93		dBm
54Mbps (3/4 64-QAM) <small>(CC^2DFSK)</small>		-75		dBm
HT20, MCS7 (65Mbps, 72.2Mbps) <small>(CC^2DFSK)</small>		-71		dBm
Adjacent suppression				
OFDM, 6Mbps		37		dB
OFDM, 54Mbps		21		dB
HT20, MCS0		37		dB
HT20, MCS7		20		dB

E.3. Módulo Bluetooth HC-06

<u>Product Data Sheet</u>										
Module Data Sheet										
Rev 1										
1. 0	2.0	2.1	2.2							
2006/6/18	2006/9/6	2010/4/22	2011/4/6							
DRAWN BY :	Ling Xin		MODEL : HC-06							
CHECKED BY :	Eric Huang		Description : BC04 has external 8M Flash and EDR module HC-06 is industrial, and compatible with civil HC-04							
APPD. BY :	Simon Mok		REV: 2.0				Page :			
Former version introduction	HC-06 is the higher version of LV_BC_2.0. Linvor is the former of wavesen.									

1. Product's picture

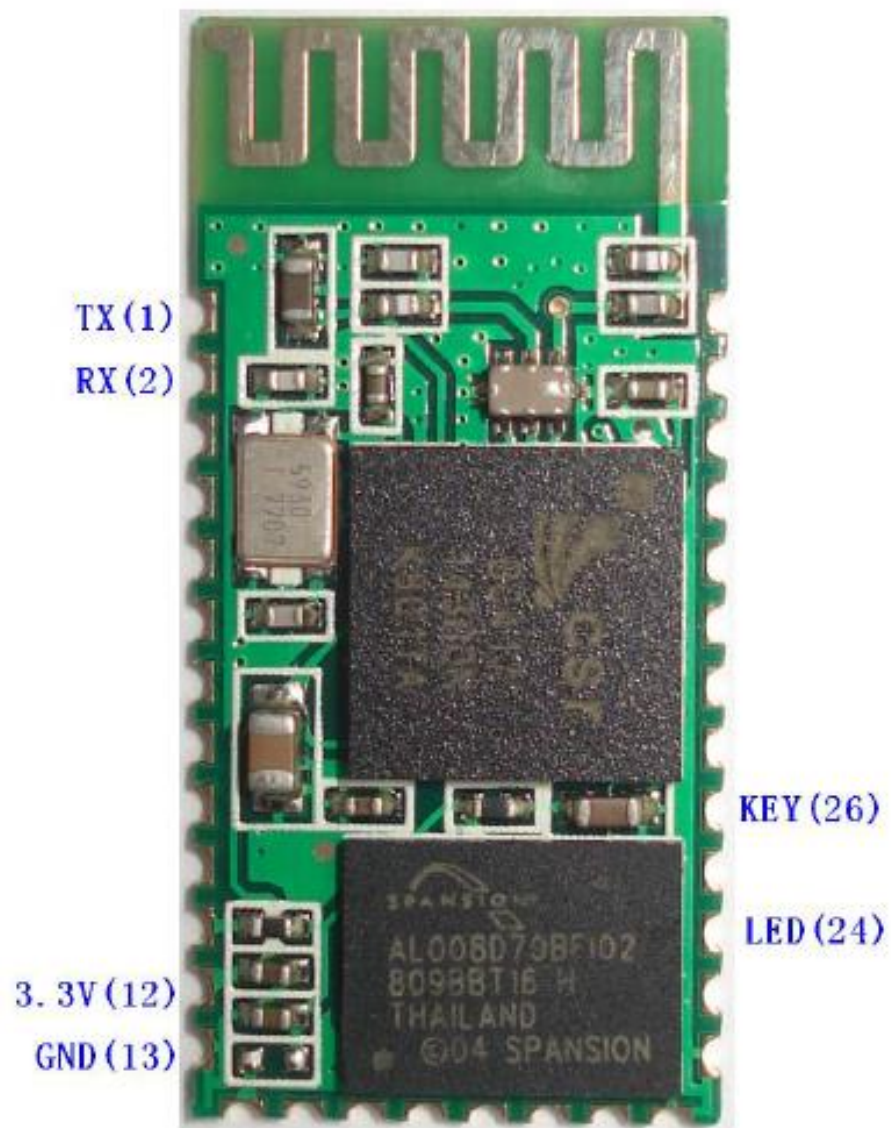


Figure 1 A Bluetooth module

2. Feature

- Wireless transceiver
 - Sensitivity (Bit error rate) can reach -80dBm.
 - The change range of output's power: -4 - +6dBm.
- Function description (perfect Bluetooth solution)
 - Has an EDR module; and the change range of modulation depth: 2Mbps - 3Mbps.
 - Has a build-in 2.4GHz antenna; user needn't test antenna.
 - Has the external 8Mbit FLASH
 - Can work at the low voltage (3.1V~4.2V). The current in pairing is in the range of 30~40mA. The current in communication is 8mA.
 - Standard HCI Port (UART or USB)
 - USB Protocol: Full Speed USB1.1, Compliant With 2.0
 - This module can be used in the SMD.
 - It's made through RoHS process.
 - The board PIN is half hole size.
 - Has a 2.4GHz digital wireless transceiver.
 - Bases at CSR BC04 Bluetooth technology.
 - Has the function of adaptive frequency hopping.
 - Small (27mm×13mm×2mm)
 - Peripherals circuit is simple.
 - It's at the Bluetooth class 2 power level.
 - Storage temperature range: -40 °C - 85°C, work temperature range: -25 °C - +75°C
 - Any wave inter Interference: 2.4MHz, the power of emitting: 3 dBm.
 - Bit error rate: 0. Only the signal decays at the transmission link, bit error may be produced. For example, when RS232 or TTL is being processed, some signals may decay.

- Low power consumption
- Has high-performance wireless transceiver system
- Low Cost
- Application fields:
 - Bluetooth Car Handsfree Device
 - Bluetooth GPS
 - Bluetooth PCMCIA , USB Dongle
 - Bluetooth Data Transfer
- Software
 - CSR

3. PINs description

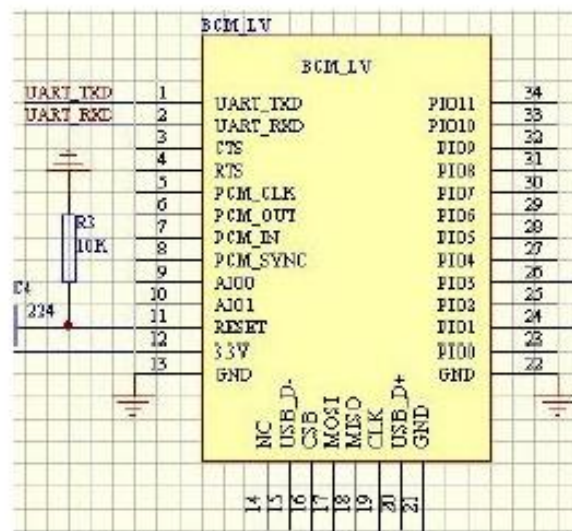


Figure 3 PIN configuration

The PINs at this block diagram is as same as the physical one.

PIN Name	PIN #	Pad type	Description	Note
GND	13 21 22	VSS	Ground pot	
1V8	14	VDD	Integrated 1.8V (+) supply with On-chip linear regulator output within 1.7-1.9V	
VCC	12	3.3V		
AIO0	9	Bi-Directional	Programmable input/output line	
AIO1	10	Bi-Directional	Programmable input/output line	

www.wavesen.com Phone: 020-84083341 Fax: 020-84332079 QQ:1043073574
 Address: Room 527, No.13, Jiangong Road, Tianhe software park, Tianhe district, Guangzhou Post: 510660
 Technology consultant: support@wavesen.com Business consultant: sales@wavesen.com
 Complaint and suggestion: sunbirdit@hotmail.com

E.4. Sensor de movimiento PIR HC-SR501

★ adafruit learning system

PIR Motion Sensor

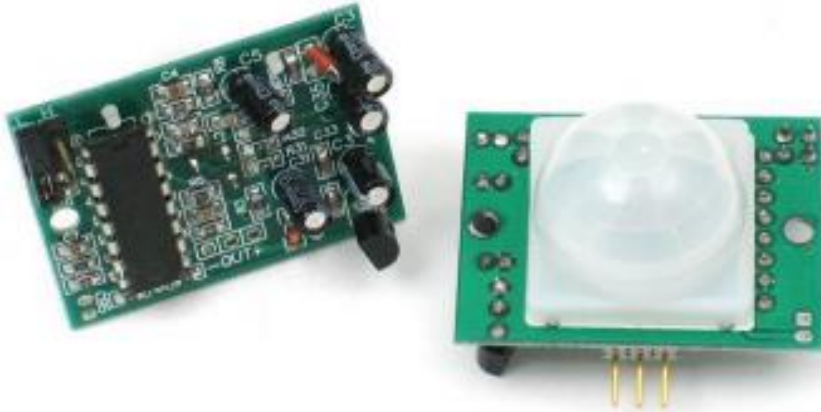
Created by lady ada



Last updated on 2017-12-26 10:07:42 PM UTC

Overview

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors.



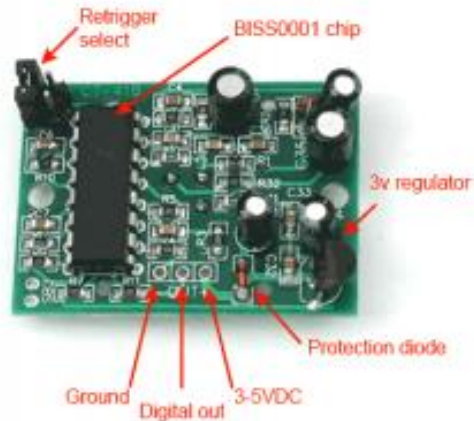
PIRs are basically made of a [pyroelectric sensor](#) (which you can see below as the round metal can with a rectangular crystal in the center), which can detect levels of infrared radiation. Everything emits some low level radiation, and the hotter something is, the more radiation is emitted. The sensor in a motion detector is actually split in two halves. The reason for that is that we are looking to detect motion (change) not average IR levels. The two halves are wired up so that they cancel each other out. If one half sees more or less IR radiation than the other, the output will swing high or low.



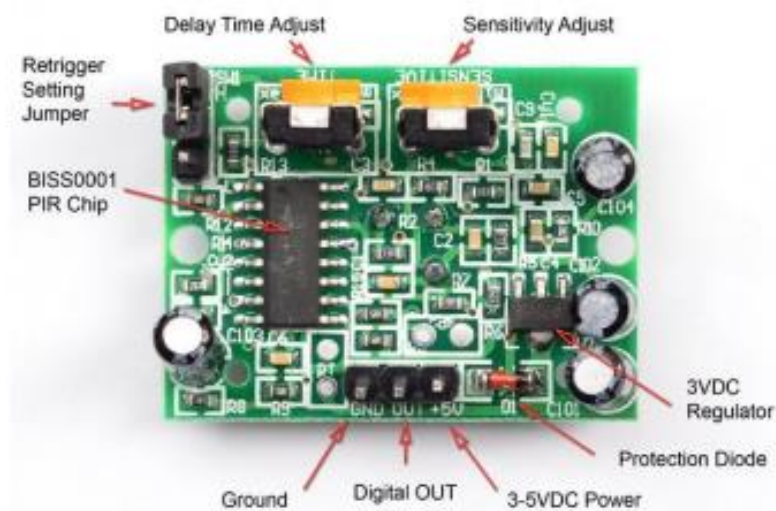
Along with the pyroelectric sensor is a bunch of supporting circuitry, resistors and capacitors. It seems that most small hobbyist sensors use the [BISS0001 \("Micro Power PIR Motion Detector IC"\)](#), undoubtedly a very inexpensive chip. This

chip takes the output of the sensor and does some minor processing on it to emit a digital output pulse from the analog sensor.

Our older PIRs looked like this:



Our new PIRs have more adjustable settings and have a header installed in the 3-pin ground/out/power pads



For many basic projects or products that need to detect when a person has left or entered the area, or has approached, PIR sensors are great. They are low power and low cost, pretty rugged, have a wide lens range, and are easy to interface with. Note that PIRs won't tell you how many people are around or how close they are to the sensor, the lens is often fixed to a certain sweep and distance (although it can be hacked somewhere) and they are also sometimes set off by housepets. Experimentation is key!

Some Basic Stats

These stats are for the PIR sensor in the Adafruit shop which is very much [like the Parallax one](#) . Nearly all PIRs will have slightly different specifications, although they all pretty much work the same. If there's a datasheet, you'll want to refer to it

- Size: Rectangular
- Price: [\\$10.00 at the Adafruit shop](#)
- Output: Digital pulse high (3V) when triggered (motion detected) digital low when idle (no motion detected). Pulse lengths are determined by resistors and capacitors on the PCB and differ from sensor to sensor.
- Sensitivity range: up to 20 feet (6 meters) 110° x 70° detection range
- Power supply: 5V-12V input voltage for most modules (they have a 3.3V regulator), but 5V is ideal in case the regulator has different specs
- [BIS0001 Datasheet](#) (the decoder chip used)
- [RE200B datasheet](#) (most likely the PIR sensing element used)
- [NL11NH datasheet](#) (equivalent lens used)
- [Parallax Datasheet on their version of the sensor](#)

More links!

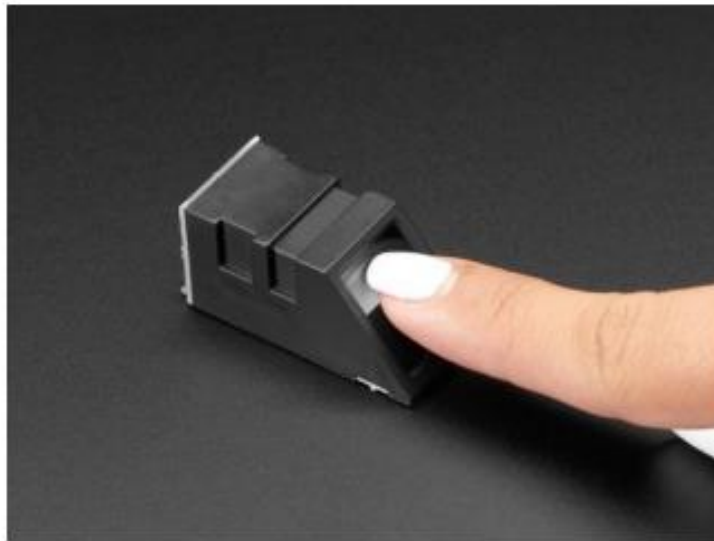
- [A great page on PIR sensors from GLOLAB \](#)

E.5. Sensor de huella dactilar - Adafruit



Adafruit Optical Fingerprint Sensor

Created by lady ada



Last updated on 2017-12-31 09:39:09 PM UTC

Overview



Secure your project with biometrics • this all-in-one optical fingerprint sensor will make adding fingerprint detection and verification super simple. These modules are typically used in safes • there's a high powered DSP chip that does the image rendering, calculation, feature-finding and searching. Connect to any microcontroller or system with TTL serial, and send packets of data to take photos, detect prints, hash and search. You can also enroll new fingers directly • up to 162 finger prints can be stored in the onboard FLASH memory. There's a red LED in the lens that lights up during a photo so you know its working.

We like this particular sensor because not only is it easy to use, it also comes with fairly straight-forward Windows software that makes testing the module simple • you can even enroll using the software and see an image of the fingerprint on your computer screen. But, of course, we wouldn't leave you a datasheet and a "good luck!" • [we wrote a full Arduino library so that you can get running in under 10 minutes. The library can enroll and search so its perfect for any project.](#) We've also [written a detailed tutorial on wiring and use.](#) This is by far the best fingerprint sensor you can get.

- Supply voltage: 3.6 • 6.0VDC
- Operating current: 120mA max
- Peak current: 150mA max
- Fingerprint imaging time: <1.0 seconds
- Window area: 14mm x 18mm
- Signature file: 256 bytes
- Template file: 512 bytes
- Storage capacity: 162 templates
- Safety ratings (1-5 low to high safety)
- False Acceptance Rate: <0.001% (Security level 3)
- False Reject Rate: <1.0% (Security level 3)
- Interface: TTL Serial

- Baud rate: 9600, 19200, 28800, 38400, 57600 (default is 57600)
- Working temperature rating: -20C to +50C
- Working humidity: 40%-85% RH
- Full Dimensions: 56 x 20 x 21,5mm
- Exposed Dimensions (when placed in box): 21mm x 21mm x 21mm triangular
- Weight: 20 grams

E.6. Reloj RTC DS3231

DS3231

Extremely Accurate I²C-Integrated RTC/TCXO/Crystal

General Description

The DS3231 is a low-cost, extremely accurate I²C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal resonator enhances the long-term accuracy of the device as well as reduces the piece-part count in a manufacturing line. The DS3231 is available in commercial and industrial temperature ranges, and is offered in a 16-pin, 300-mil SO package.

The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. Two programmable time-of-day alarms and a programmable square-wave output are provided. Address and data are transferred serially through an I²C bidirectional bus.

A precision temperature-compensated voltage reference and comparator circuit monitors the status of V_{CC} to detect power failures, to provide a reset output, and to automatically switch to the backup supply when necessary. Additionally, the RST pin is monitored as a pushbutton input for generating a μ P reset.

Benefits and Features

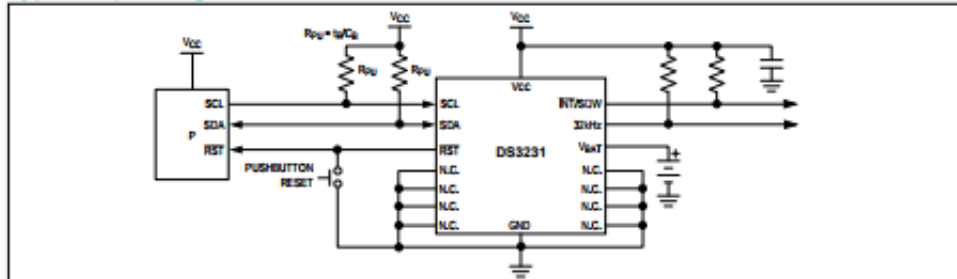
- Highly Accurate RTC Completely Manages All Timekeeping Functions
 - Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year, with Leap-Year Compensation Valid Up to 2100
 - Accuracy ± 2 ppm from 0°C to +40°C
 - Accuracy ± 3.5 ppm from -40°C to +85°C
 - Digital Temp Sensor Output: ± 3 °C Accuracy
 - Register for Aging Trim
 - RST Output/Pushbutton Reset Debounce Input
 - Two Time-of-Day Alarms
 - Programmable Square-Wave Output Signal
- Simple Serial Interface Connects to Most Microcontrollers
 - Fast (400kHz) I²C Interface
- Battery-Backup Input for Continuous Timekeeping
 - Low Power Operation Extends Battery-Backup Run Time
 - 3.3V Operation
- Operating Temperature Ranges: Commercial (0°C to +70°C) and Industrial (-40°C to +85°C)
- Underwriters Laboratories® (UL) Recognized

Applications

- Servers
- Telematics
- Utility Power Meters
- GPS

Ordering Information and Pin Configuration appear at end of data sheet.

Typical Operating Circuit



Underwriters Laboratories is a registered certification mark of Underwriters Laboratories Inc.

DS3231

Extremely Accurate I²C-Integrated
RTC/TCXO/Crystal

Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground-0.3V to +6.0V
 Junction-to-Ambient Thermal Resistance (θ_{JA}) (Note 1) 73°C/W
 Junction-to-Case Thermal Resistance (θ_{JC}) (Note 1)23°C/W
 Operating Temperature Range
 DS3231S0°C to +70°C
 DS3231SN+40°C to +85°C

Junction Temperature+125°C
 Storage Temperature Range+40°C to +85°C
 Lead Temperature (soldering, 10s)+260°C
 Soldering Temperature (reflow, 2 times max)+260°C
 (see the Handling, PCB Layout, and Assembly section)

Note 1: Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to www.maximintegrated.com/thermal-tutorial.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only; and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Recommended Operating Conditions

($T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{CC}		2.3	3.3	5.5	V
	V_{BAT}		2.3	3.0	5.5	V
Logic 1 Input SDA, SCL	V_{IH}		0.7 x V_{CC}		$V_{CC} + 0.3$	V
Logic 0 Input SDA, SCL	V_{IL}		-0.3		0.3 x V_{CC}	V

Electrical Characteristics

($V_{CC} = 2.3V$ to $5.5V$, V_{CC} = Active Supply (see Table 1), $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.) (Typical values are at $V_{CC} = 3.3V$, $V_{BAT} = 3.0V$, and $T_A = +25^\circ C$, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Active Supply Current	I_{CCA}	(Notes 4, 5)	$V_{CC} = 3.63V$		200	μA
			$V_{CC} = 5.5V$		300	
Standby Supply Current	I_{CCS}	I ² C bus inactive, 32kHz output on, SQW output off (Note 5)	$V_{CC} = 3.63V$		110	μA
			$V_{CC} = 5.5V$		170	
Temperature Conversion Current	I_{CCCONV}	I ² C bus inactive, 32kHz output on, SQW output off	$V_{CC} = 3.63V$		575	μA
			$V_{CC} = 5.5V$		650	
Power-Fail Voltage	V_{PF}		2.45	2.575	2.70	V
Logic 0 Output, 32kHz, INT/SQW, SDA	V_{OL}	$I_{OL} = 3mA$			0.4	V
Logic 0 Output, RST	V_{OL}	$I_{OL} = 1mA$			0.4	V
Output Leakage Current 32kHz, INT/SQW, SDA	I_{LO}	Output high impedance	-1	0	+1	μA
Input Leakage SCL	I_{UI}		-1		+1	μA
RST Pin IO Leakage	I_{OL}	RST high impedance (Note 6)	-200		+10	μA
V_{BAT} Leakage Current (V_{CC} Active)	I_{BATLKG}			25	100	nA

DS3231

Extremely Accurate I²C-Integrated
RTC/TCXO/Crystal**Electrical Characteristics (continued)**(V_{CC} = 2.3V to 5.5V, V_{CC} = Active Supply (see Table 1), T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Typical values are at V_{CC} = 3.3V, V_{BAT} = 3.0V, and T_A = +25°C, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output Frequency	f _{OUT}	V _{CC} = 3.3V or V _{BAT} = 3.3V		32.768		kHz
Frequency Stability vs. Temperature (Commercial)	Δf/f _{OUT}	V _{CC} = 3.3V or V _{BAT} = 3.3V, aging offset = 00h	0°C to +40°C		±2	ppm
			>40°C to +70°C		±3.5	
Frequency Stability vs. Temperature (Industrial)	Δf/f _{OUT}	V _{CC} = 3.3V or V _{BAT} = 3.3V, aging offset = 00h	-40°C to <0°C		±3.5	ppm
			0°C to +40°C		±2	
			>40°C to +85°C		±3.5	
Frequency Stability vs. Voltage	Δf/V			1		ppm/V
Trim Register Frequency Sensitivity per LSB	Δf/LSB	Specified at:	-40°C		0.7	ppm
			+25°C		0.1	
			+70°C		0.4	
			+85°C		0.8	
Temperature Accuracy	Temp	V _{CC} = 3.3V or V _{BAT} = 3.3V	-3		+3	°C
Crystal Aging	Δf/%	After reflow, not production tested	First year		±1.0	ppm
			0-10 years		±5.0	

Electrical Characteristics(V_{CC} = 0V, V_{BAT} = 2.3V to 5.5V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Active Battery Current	I _{BATA}	EO _{SC} = 0, BBSQW = 0, SCL = 400kHz (Note 5)	V _{BAT} = 3.63V		70	μA
			V _{BAT} = 5.5V		150	
Timekeeping Battery Current	I _{BATT}	EO _{SC} = 0, BBSQW = 0, EN32kHz = 1, SCL = SDA = 0V or SCL = SDA = V _{BAT} (Note 5)	V _{BAT} = 3.63V	0.84	3.0	μA
			V _{BAT} = 5.5V	1.0	3.5	
Temperature Conversion Current	I _{BATTC}	EO _{SC} = 0, BBSQW = 0, SCL = SDA = 0V or SCL = SDA = V _{BAT}	V _{BAT} = 3.63V		575	μA
			V _{BAT} = 5.5V		650	
Data-Retention Current	I _{BATDR}	EO _{SC} = 1, SCL = SDA = 0V, +25°C			100	nA

DS3231

Extremely Accurate I²C-Integrated
RTC/TCXO/Crystal**AC Electrical Characteristics**(V_{CC} = V_{CC(MIN)} to V_{CC(MAX)} or V_{BAT} = V_{BAT(MIN)} to V_{BAT(MAX)}; V_{BAT} > V_{CC}; T_A = T_{MIN} to T_{MAX}; unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f _{SCL}	Fast mode	100		400	kHz
		Standard mode	0		100	
Bus Free Time Between STOP and START Conditions	t _{BUF}	Fast mode	1.3			μs
		Standard mode	4.7			
Hold Time (Repeated) START Condition (Note 7)	t _{HD:STA}	Fast mode	0.6			μs
		Standard mode	4.0			
Low Period of SCL Clock	t _{LOW}	Fast mode	1.3			μs
		Standard mode	4.7			
High Period of SCL Clock	t _{HIGH}	Fast mode	0.6			μs
		Standard mode	4.0			
Data Hold Time (Notes 8, 9)	t _{HD:DAT}	Fast mode	0		0.9	μs
		Standard mode	0		0.9	
Data Setup Time (Note 10)	t _{SU:DAT}	Fast mode	100			ns
		Standard mode	250			
START Setup Time	t _{SU:STA}	Fast mode	0.6			μs
		Standard mode	4.7			
Rise Time of Both SDA and SCL Signals (Note 11)	t _r	Fast mode	20 + _____		300	ns
		Standard mode	0.1C _B		1000	
Fall Time of Both SDA and SCL Signals (Note 11)	t _f	Fast mode	20 + _____		300	ns
		Standard mode	0.1C _B		300	
Setup Time for STOP Condition	t _{SU:STO}	Fast mode	0.6			μs
		Standard mode	4.7			
Capacitive Load for Each Bus Line	C _B	(Note 11)			400	pF
Capacitance for SDA, SCL	C _{I/O}			10		pF
Pulse Width of Spikes That Must Be Suppressed by the Input Filter	t _{SP}			30		ns
Pushbutton Debounce	PBDB			250		ms
Reset Active Time	t _{RST}			250		ms
Oscillator Stop Flag (OSF) Delay	t _{OSF}	(Note 12)		100		ms
Temperature Conversion Time	t _{CONV}			125	200	ms

Power-Switch Characteristics(T_A = T_{MIN} to T_{MAX})

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V _{CC} Fall Time; V _{PF(MAX)} to V _{PF(MIN)}	t _{VCCF}		300			μs
V _{CC} Rise Time; V _{PF(MIN)} to V _{PF(MAX)}	t _{VCCR}		0			μs
Recovery at Power-Up	t _{REC}	(Note 13)		250	300	ms