

UNIVERSIDAD MAYOR DE SAN ANDRES
FACULTAD DE TECNOLOGIA
CARRERA DE ELECTRONICA Y TELECOMUNICACIONES



PROYECTO DE APLICACIÓN DE EXAMEN DE GRADO

**AUTOMATIZACION RESIDENCIAL: ACCIONAMIENTO DE
DISPOSITIVOS DOMESTICOS UTILIZANDO ARDUINO CON
INTEGRACION DEL SISTEMA OPERATIVO ANDROID**

POSTULANTE; ARIEL DAVID BRAVO PACHECO

JULIO 2016

LA PAZ- BOLIVIA

TABLA DE CONTENIDO

RESUMEN	1
1. INTRODUCCION.....	2
2. ANTECEDENTES	3
3. PLANTEAMIENTO DEL PROBLEMA	4
3.1. IDENTIFICACION DEL PROBLEMA	4
3.2. FORMULACION DEL PROBLEMA.....	5
4. OBJETIVOS	5
4.1. OBJETIVO GENERAL	5
4.2. OBJETIVOS ESPECÍFICOS.....	5
5. JUSTIFICACIÓN	5
6. DELIMITACIÓN	6
6.1. TEMÁTICA.....	6
6.2. TEMPORAL	6
6.3. ESPACIAL	6
7. ESTRATEGIA METODOLÓGICA.....	6
8. INGENIERÍA DEL PROYECTO.....	7
8.1. FUNDAMENTO TEORICO	7
8.1.1. Domótica	7
8.1.1.1. Domótica en Bolivia	8
8.1.1.2. Beneficios.....	8
8.1.1.3. Aplicaciones	9
8.1.1.4. Seguridad.....	14
8.1.1.5. Confort (Automatismos)	15
8.1.1.6. Previsión del futuro	16
8.1.2. Arduino	17

8.1.2.1.	Hardware.....	20
8.1.2.2.	Aplicaciones	21
8.1.2.3.	Esquemas de conexiones	22
8.1.2.4.	Lenguaje de programación Arduino	24
8.1.2.5.	Ventajas del IDE Arduino	25
8.1.2.6.	Primera ejecución del IDE Arduino y configuración inicial para el desarrollo del proyecto.....	25
8.1.2.7.	Opciones de la interfaz	27
8.1.2.8.	Arduino NANO	27
8.1.2.8.1.	Especificaciones	28
8.1.2.8.2.	Alimentación	28
8.1.2.8.3.	Memoria.....	29
8.1.2.8.4.	Entradas y Salidas	29
8.1.2.8.5.	Comunicaciones	30
8.1.2.8.6.	Programación	30
8.1.2.8.7.	Diagrama de flujo del programa de la tarjeta NANO.....	31
8.1.2.8.8.	Reinicio automático (Software)	32
8.1.2.8.9.	Referencias del lenguaje	33
8.1.2.8.10.	Librerías	35
8.1.2.8.11.	Librerías estándar	35
8.1.2.9.	Microcontrolador ATMEGA 328	36
8.1.2.9.1.	Parámetros fundamentales.....	37
8.1.2.9.2.	Aplicaciones.....	37
8.1.2.9.3.	Recursos especiales de los microcontroladores.....	37
8.1.2.9.4.	Temporizadores (Timers).....	38
8.1.2.9.5.	Perro guardián (Watchdog).....	38

8.1.2.9.6.	Protección ante fallos de alimentación (Brownout)	38
8.1.2.9.7.	Estado de reposo o de bajo consumo.....	39
8.1.2.9.8.	Convertor A/D (CAD)	39
8.1.2.9.9.	Convertor D/A (CDA)	40
8.1.2.9.10.	Puertas de Comunicación.....	40
8.1.2.9.11.	Correspondencia entre los pines ATMEGAS y ARDUINO. 40	
8.1.2.10.	ANDROID	41
8.1.2.10.1.	Introducción	41
8.1.2.10.2.	Historia.....	42
8.1.2.10.3.	Arquitectura y Características	42
8.1.2.10.4.	Aplicaciones en ANDROID	45
8.1.2.10.5.	Librerías ANDROID	45
8.1.2.10.6.	Librerías Avanzadas	47
8.1.2.10.7.	Multimedia	47
8.1.2.11.	Plataforma App Inventor.....	51
8.1.2.11.1.	Introducción	51
8.1.2.11.2.	Historia.....	52
8.1.2.11.3.	Características	52
8.1.2.12.	BLUETOOTH	53
8.1.2.12.1.	Introducción	53
8.1.2.12.2.	Aplicaciones.....	55
8.1.2.12.3.	Arquitectura Hardware	56
8.1.2.12.4.	Usos de BLUETOOTH.....	57
8.1.2.12.5.	Módulo BLUETOOTH H 05.....	57
8.1.2.12.6.	Características	58

8.2.	DESARROLLO PRACTICO/EXPERIMENTAL	60
8.2.1.	PROGRAMACION EN ARDUINO	60
8.2.1.1.	Sensor PIR	60
8.2.1.1.1.	Características Técnicas	61
8.2.1.1.2.	Forma de Operación.....	62
8.2.1.2.	Control de dirección de motores con C.I. L293 (PUENTE H) ...	62
8.2.1.2.1.	Características técnicas	63
8.2.1.2.2.	Tabla de verdad para el funcionamiento de los motores.....	64
8.2.2.	Programación de la aplicación android en App Inventor.....	64
8.2.3.	Circuito en Proteus	67
9.	ANÁLISIS DE COSTOS	67
9.1.	COSTOS FIJOS.....	67
9.2.	COSTOS VARIABLES	68
10.	CONCLUSIONES Y RECOMENDACIONES.....	68
11.	BIBLIOGRAFIA	69

ÍNDICE DE FIGURAS

Figura 1. Domótica	7
Figura 2. Beneficios de Domótica.....	10
Figura 3. Esquema conceptual VoIP	13
Figura 4. Dispositivo de mando asociado.....	16
Figura 5. Las casas del mañana	17
Figura 6. Placas Arduino	21
Figura 7. Primer paso de Ejecución del IDE Arduino como Administrador.....	25
Figura 8. Selección de tarjeta Arduino Nano	26
Figura 9. Selección del puerto a utilizar	26
Figura 10. IDE de desarrollo Arduino v.1.0.1.....	27
Figura 11. Diagrama de flujo del programa de la tarjeta NANO	32
Figura 12. Librerías de IDE de Arduino	35
Figura 13. Puertos E/S del ATMEGA328	36
Figura 14. Reseteo del dispositivo por Brownout	39
Figura 15. Correspondencia entre pines ATMEGA y ARDUINO	41
Figura 16. Arquitectura de un sistema Android	43
Figura 17. Logo de App Inventor	51
Figura 18. Desarrollo con App Inventor.....	53
Figura 19. Ejemplo con App Inventor	53
Figura 20. Módulo BLUETOOTH H 05.....	58
Figura 21. Sensor de movimiento PIR.....	60
Figura 22. Conexión PIR – ARDUINO.....	61
Figura 23. Resistencias variables de calibración.....	62
Figura 24. Circuito integrado L293 (PUENTE H).....	63
Figura 25. Diagrama de PINES L293.....	63

Figura 26. Tabla de verdad del motor	64
Figura 27. Editor en bloques en App inventor	65
Figura 28. Alcance de señal libre y con barreras	65
Figura 29. Simulador de App Inventor.....	66
Figura 30. Simulador del proyecto.....	66
Figura 31. Circuito en Proteus.....	67

ÍNDICE DE TABLAS

Tabla 1. Módulos Arduinos.....	23
Tabla 2. Especificaciones de Arduino Nano.....	28
Tabla 3. Variables utilizadas en Arduino	33
Tabla 4. Estructuras utilizadas en Arduino	34
Tabla 5. Funciones utilizadas en Arduino.....	34
Tabla 6. parámetros del microcontrolador ATMEGA 328.....	37
Tabla 7. Clasificación por Potencia	54
Tabla 8. Versiones BLUETOOTH.....	55
Tabla 9. Costos fijos del proyecto	67
Tabla 10. Costos variables del proyecto	68

RESUMEN

El presente proyecto de aplicación de examen de grado muestra los beneficios en cuanto a tiempo, ahorro de energía, seguridad, gestión remota y más que todo comodidad y bienestar de tener una casa domótica así también de las características de sensores y sus componentes empleados, interfaz entre teléfono móvil y usuario así como también un estudio al sistema operativo android.

El fondo del proyecto es permitir una automatización de luces a través de sensores de movimiento, seguridad a través de infrarrojos y sensores de luz para puertas y un vínculo entre el usuario y su hogar operado a través de un teléfono móvil.

En el hogar están incorporados tres sensores. El primero es un sensor de movimiento PIR, el cual permite saber de la existencia de un movimiento así sin la necesidad de un switch activar las luces del ambiente al que entra, al no tener movimiento alguno después de un tiempo considerable las luces automáticamente se apagaran, el segundo sensor es un LDR o sensor fotovoltaico, este permitirá controlar las luces así como el sensor del garaje, el tercero es un infrarrojo que nos ayudara en la detección de una ventana que se abra sin autorización

La comunicación entre usuario y el teléfono móvil será mediante bluetooth creando una aplicación para la conexión entre usuario y todo el sistema del hogar

1. INTRODUCCION

Un sistema de gestión remota es una aplicación electrónica la cual sirve como interfaz del usuario para el control de sistemas del hogar. Esta aplicación utilizara técnicas de procesamiento digital de comandos seleccionados en el teléfono móvil inteligente (Smartphone), ordenador o tableta móvil.

Es un hecho que en la actualidad la tecnología ha tenido un avance considerable a comparación de años anteriores, lo que ha permitido el desarrollo de diferentes aplicaciones y dispositivos, donde muchas de las actividades cotidianas son realizadas por medio de dispositivos electrónicos. Es un hecho que la domótica está generando gran impacto en la sociedad, y que en un futuro no muy lejano, gracias a las ventajas que ofrece, será la gran tendencia mundial. Y no solo la domótica se está posicionando fuerte, en nuestra sociedad hay un boom tecnológico por así decirlo, en cuanto al uso de celulares inteligentes y tabletas.

Como se conoce los celulares inteligentes, tienen gran impacto a nivel mundial, donde una parte considerable de la humanidad posee este tipo de dispositivo electrónico. En países como estados unidos se tiene cifras de que alrededor de más del 50% de los usuarios de Smartphone, poseen celulares con sistema operativo Android, el sistema operativo que sigue en la lista es iOS de iPhone, con un 15%, mostrando una alta tendencia por los dispositivos con el sistema operativo que provee Google.

Basándose en esta tendencia en los Smartphones, en cuanto a su preferencia por sistema operativo, y llegando a la conclusión que los usuarios también prefieren el sistema operativo Android en las Tablets, se decide crear una aplicación para los dispositivos que utilizan este sistema operativo. La aplicación tiene como fin combinar el uso de estos dispositivos electrónicos con la domótica, creando un prototipo de sistema de control domótico basado en Android, el cual permitirá controlar, luces, alarmas, puertas, entre otros

Actualmente los sistemas domóticos tienen un precio muy alto de instalación con lo cual solo es posible verlo en casas de lujo. Estos suelen utilizar buses de transmisión de información que posibilitan una domótica robusta como son el EIB, X10, CEBus, LonWorks/LongTalk y ZigBee. Una alternativa más barata y

casera consiste en la utilización de placas Arduino. En este proyecto utilizaremos la plataforma Arduino en la que nos apoyaremos con otros dispositivos para poder construir un sistema domótico simple. Arduino es una plataforma de hardware libre creada en 2005, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios. Para crear el sistema domótico han de tenerse en cuenta varios aspectos. Hay que conocer el capital que disponemos para invertir en el sistema y seleccionar los dispositivos que más se ajusten a nuestras necesidades. De poco sirve comprar un elemento con grandes prestaciones si luego no se va a aprovechar.

2. ANTECEDENTES

La domótica, es uno de los efectos de la globalización, donde en países desarrollados está teniendo un gran avance. En sus inicios presento grandes inconvenientes, debido al costo elevado que presentaba para los usuarios, hoy en día es mucho más asequible que en sus primeras fases, y las ventajas que ofrecen son mucho más evidentes hoy en día que hace unos años; entre estas ventajas se tiene el ahorro energético, confort, seguridad, accesibilidad, entre otras.

El avance de la domótica en cada país es diferente, en unos países ha tenido gran impacto y desarrollo, mientras que en otros países el desarrollo es más bien escaso. Colombia no se queda atrás en cuanto a este tema, en nuestro país el avance ha sido grande, tanto así que los proyectos de esta índole se está llevando hasta los estratos 3, con precios acordes asequibles y facilidades de pago.

Debido a la gran cantidad de empresas que están ejerciendo en este campo, cada proyecto que se relaciona con la domótica, para poder ser exitoso debe tener un valor agregado, para de esta manera sobresalga en el mercado. El uso de dispositivos como lo son los Smartphones y las Tablets, ha ido generando un consumismo excesivo en la sociedad, donde un gran porcentaje de esta, posee uno o ambos dispositivos. La preferencia por uno u otro dispositivo varía de las funcionalidades que quiera el usuario, o simplemente de la condición económica que se posea.

Existen diversos sistemas operativos que corren sobre estos dispositivos, entre los destacables se tienen Android de Google, iOS de Apple, WindowsMobile y BlackBerryOS. Los sistemas operativos que más se usan son el Android y el iOS, donde alrededor de más del 50% de los usuarios prefiere utilizar Android que iOS, debido a las diferentes ventajas que posee el sistema operativo de Google.

3. PLANTEAMIENTO DEL PROBLEMA

3.1. IDENTIFICACION DEL PROBLEMA

En la actualidad la dinámica productiva enfrenta a las personas e instituciones al desafío de la eficiencia, la velocidad y la ubicuidad; en el mundo moderno cualquier actividad que no pueda ser realizada en forma eficiente mediante herramientas remotas representa una enorme desventaja, tanto competitivamente como de calidad de vida. Dentro de esta perspectiva, la automatización de los espacios y su dinamización con la incorporación de herramientas y tecnologías de la información y las telecomunicaciones, es más que un capricho frívolo, una necesidad inminente.

Son múltiples los factores que pueden caracterizar las formas de vida de las personas hoy en día, como ejemplo, adultos mayores que viven solos, población con algún tipo de discapacidad y que viven de forma independiente, incluso los mismos avatares de la modernidad que requieren todo con prontitud y ahorro del tiempo que no les permite siquiera descender del vehículo.

Es así como, labores tan comunes como abrir o cerrar una puerta, encender y apagar las luces o seguridad del hogar se convierten en un problema; pero con la implementación de la domótica en el hogar se disminuye el grado de dificultad para realizar estas tareas. En el mercado actual se cuenta con un gran número de aplicaciones de muy alto nivel sobre el tema, las cuales en su mayoría tienen costos elevados y están pensadas más en generar una experiencia de lujo al cliente que en facilitar el desarrollo de una actividad específica a las personas.

Las características anteriormente descritas ponen en evidencia un desafío mayor: poder implementar un sistema de domótica orientado a facilitar

actividades domésticas o empresariales asociados al control de acceso y seguridad de los espacios contando con un bajo costo, alto nivel de calidad, confiabilidad y seguridad.

3.2. FORMULACION DEL PROBLEMA

Tomando en cuenta el problema detallado anteriormente, debemos atender y solucionar el problema desde un punto de vista técnico dando una solución práctica eficiente con bajo costo y que solucione esta problemática es así como recurrimos a formular la siguiente interrogante el cual será el problema de la investigación:

¿Cómo contribuir al aumento de calidad de vida de nuestra sociedad mediante un dispositivo remoto que interactúe entre el usuario y su hogar?

4. OBJETIVOS

4.1. OBJETIVO GENERAL

Diseñar e implementar un sistema de automatización residencial mediante un teléfono móvil que pueda controlar distintos áreas del hogar.

4.2. OBJETIVOS ESPECÍFICOS

- Simular un sistema domótico que controle la iluminación, puerta automática y seguridad con alarma.
- Diseñar una aplicación para el teléfono móvil con sistema operativo Android que contenga los parámetros para administrar los dispositivos del prototipo, controlando luces y motores.
- Utilizar tecnología Bluetooth para la transmisión y recepción de datos procesados por Arduino para el control de los diferentes dispositivos instalados en el hogar.

5. JUSTIFICACIÓN

La investigación gira en torno a la automatización y control de los espacios, no solamente como un status elevado de vida para algunas personas sino también algo necesario para personas de la tercera edad o personas

discapacitadas que mostraban hasta ahora una dependencia en segundas o terceras personas, no solo para sus necesidades vitales sino también para su seguridad.

6. DELIMITACIÓN

6.1. TEMÁTICA

Este proyecto de aplicación de examen de grado se desarrollará en el análisis y diseño de un sistema automatizado residencial que tendrá la particularidad de ser manipulado u operado por alguna persona mediante una conexión bluetooth utilizando un procesado por una placa Arduino.

6.2. TEMPORAL

De acuerdo a los objetivos planteados anteriormente, se prevé desarrollar el proyecto en un tiempo de 2 meses.

6.3. ESPACIAL

El desarrollo del proyecto de aplicación se llevara a cabo de forma local, en domicilios paceños que cuenten con todas las herramientas necesarias que se utilizaran a lo largo del proyecto.

7. ESTRATEGIA METODOLÓGICA

El proyecto de aplicación de examen de grado está dirigido principalmente al estudio de sistemas inalámbricos y automatizados, los cuales tienen gran aplicación en proyectos de interés común referido a la ayuda y seguridad hacia personas de la tercera edad y personas con discapacidad. Este tema de redes inalámbricas es de interés actual tanto por la necesidad de desechar todo cableado así como por la comodidad en la movilidad, no solo en domótica también para televisión, telefonía, seguridad, etc.

El trabajo se divide básicamente en tres partes:

- La electrónica
- El software
- El desarrollo y la aplicación en android y la comunicación bluetooth

También necesitamos; Investigar acerca de parámetros técnicos que ayuden al proyecto y cotizar en los almacenes de componentes electrónicos la aproximación del costo total del proyecto de aplicación.

8. INGENIERÍA DEL PROYECTO

8.1. FUNDAMENTO TEORICO

8.1.1. Domótica

La domótica (del latín *domus* (hogar) es el conjunto de sistemas automatizados de una vivienda que aportan servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación. Se centra en los servicios de bienestar, seguridad y comunicaciones que pueden facilitarse en la vivienda a sus habitantes.

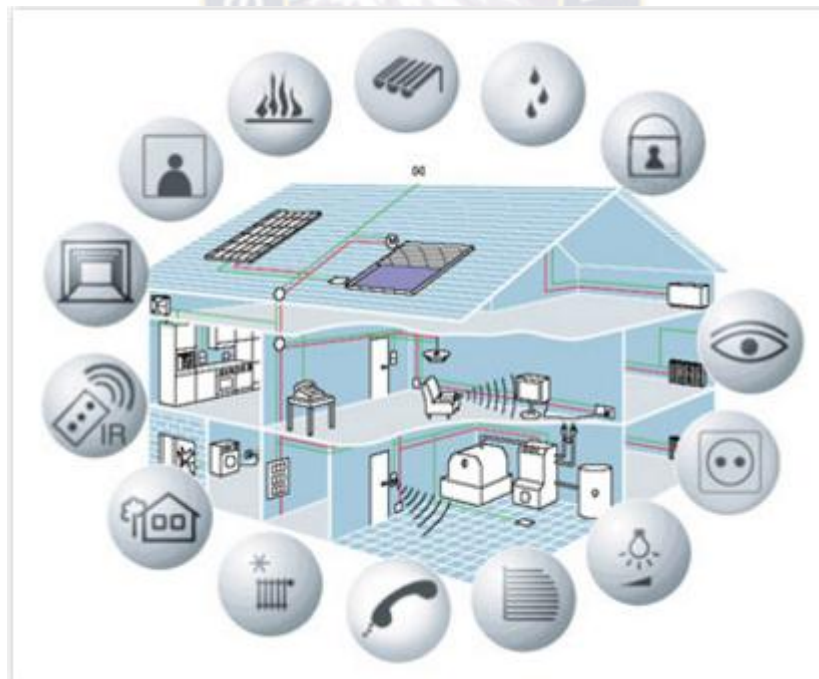


Figura 1. Domótica

8.1.1.1. Domótica en Bolivia

En Bolivia solo 50 hogares cuentan con tecnología domótica (casa inteligente) a comparación de otras ciudades metropolitanas como Sao Paulo (40% de las viviendas) o Santiago (20%), que incluso han ingresado a la tecnología inalámbrica, con la premisa de ahorrar energía eléctrica.

Pocas casas inteligentes. "Según nuestro estudio de mercado, en el país solo existen 50 casas inteligentes, de las cuales Santa Cruz cuenta con el 70% y el restante 30% están ubicadas en la zona sur de La Paz y la ciudad de Cochabamba. Pero cabe hacer énfasis que estas cuentan con tecnología domótica a través de cableado, a comparación de la última tendencia que es lo inalámbrico y que controla todo en el hogar por medio de módulos", comentó Cristóbal Gutiérrez, gerente manager de Fibaro Bolivia.

Gutiérrez también indicó que esta tecnología requiere una inversión del 12 al 18 por ciento del valor total de la vivienda.

"Para un departamento de 70 m² la inversión para obtener la tecnología domótica será entre 5 a 6 mil dólares", explicó el representante al destacar que con la tecnología inalámbrica el usuario tendrá el dominio de las puertas electrónicas, cámaras de seguridad, luces, cortinas, electrodomésticos y aires acondicionados.

Por su parte, Roxana Pérez, gerente de Control House, destacó que la correcta utilización de la innovación permite el ahorro de electricidad, que al cabo de tres años compensa la inversión de esta tecnología.

8.1.1.2. Beneficios

Los beneficios que aporta la domótica son múltiples, y se podría afirmar que cada día surgen nuevos. Por ello los agruparemos en los siguientes apartados:

- 1- El ahorro energético gracias a una gestión tarifaria e "inteligente" de los sistemas y consumos.
- 2- La potenciación y enriquecimiento de la propia red de comunicaciones.
- 3- La más contundente seguridad personal y patrimonial.
- 4- La tele-asistencia.

5- La gestión remota (vía teléfono, radio, Internet, etc.) de instalaciones y equipos domésticos.

6- Aumento del bienestar y en definitiva, del confort.

8.1.1.3. Aplicaciones

La domótica busca el aprovechamiento al máximo de la energía y luz solar adecuando su comportamiento a nuestras necesidades.

Las posibles aplicaciones son innumerables dadas las posibilidades de la domótica, podemos decir tranquilamente que las posibilidades son tan extensas como puedan ser las pretensiones de los propios usuarios, por ello trataremos de agruparlas en las más comunes:

En el ámbito del ahorro energético:

a) Programación y zonificación de la climatización. El usuario personaliza a qué hora y que zonas de la vivienda desea que estén gestionadas por el control central

b) Racionalización de cargas eléctricas: desconexión de equipos de uso no prioritario en función del consumo eléctrico en un momento dado. (Reduce la potencia contratada).

c) Gestión de tarifas, derivando el funcionamiento de algunos aparatos a horas de tarifa reducida.

En el ámbito del nivel de confort:

a) Control de todos los dispositivos instalados y operativos desde un dispositivo central simplificando su gestión y optimizando su uso.

b) Apagado general de todas las luces de la vivienda.

c) Automatización del apagado/encendido en cada punto de luz. La forma de encender y apagar la iluminación de la vivienda puede ser automatizada y controlada de formas complementarias al control tradicional a través del interruptor clásico. Se puede en esta manera conseguir un incremento del confort y ahorro energético.

La iluminación puede ser regulada en función del nivel de luminosidad ambiente, evitando su encendido innecesario o adaptándola a las necesidades

del usuario. La activación de ésta se realiza siempre cuando el nivel de luminosidad pasa un determinado umbral, ajustable por parte del usuario. Esto garantiza un nivel de iluminación mínima, que puede ser especialmente útil para por ejemplo un pasillo o la iluminación exterior.



Figura 2. Beneficios de Domótica

La iluminación puede ser activada en función de la presencia de personas en la estancia. Se activa la iluminación cuando un sensor detecta presencia. Esto garantiza una buena iluminación para por ejemplo zonas de paso como pasillos. Asegura que luces no se quedan encendidas en habitaciones cuando no hace falta.

- d)** Regulación de la iluminación según el nivel de luminosidad ambiente.
- e)** Automatización de todos los distintos sistemas/ instalaciones / equipos dotándolos de control eficiente y de fácil manejo. El hecho de que los sistemas de la vivienda se pueden programar ya sea para que realicen ciertas funciones con sólo tocar un botón o que las lleven a cabo en función de otras condiciones del entorno (hora, temperatura interior o exterior, etc.) produce un aumento del confort y un ahorro de tiempo.
- f)** Integración del portero al teléfono, o del video-portero al televisor. La señal de audio y control del portero automático se puede integrar en la red de telefonía interior de la vivienda, para permitir utilizar el teléfono en lugar de la habitual consola de control de esta instalación.

Cualquier llamada desde el portero automático puede ser atendida desde un terminal telefónico, entablando conversación con la persona visitante y, si es preciso, abrirle la puerta. La señal de vídeo y control del video-portero automático se puede integrar en la red de televisión de la vivienda y edificio, para permitir utilizar el televisor en lugar de la habitual consola de control de esta instalación. Cualquier llamada desde el video-portero automático puede ser atendida desde el televisor, reconociendo la persona visitante y, si es preciso, abrirle la puerta mediante el propio mando a distancia del televisor (u otro de uso específico).

Opcionalmente, y cuando no hay nadie en la vivienda, podría pensarse en desviar la llamada desde el portero automático a un número de abonado telefónico, simulando la presencia de un usuario en casa o abrirle la puerta de acceso de la calle a por ejemplo un mensajero.

g) El riego automático es una aplicación muy utilizada por la gente que vive en viviendas unifamiliares. El riego puede ser gestionado por un controlador que normalmente se limita a regar según la programación horaria. Pero el riego puede ser más desarrollado y avanzado que eso. Puede ser activado de forma automática según programación horaria, pero también según la humedad en el césped, el día de la semana o cualquier otro valor. Además si el riego está integrado en el sistema de domótica se puede controlar el riego de forma remota o según otros eventos como incendios o robos.

Además existe la posibilidad de realizar actuaciones puntuales y personalizadas como por ejemplo regar por la tarde en vez de por la noche si el dueño planifica una barbacoa con los amigos por la noche.

En el ámbito de la protección personal y patrimonial:

a) Detección de un posible intruso. En caso de intruso el control central se encarga de hacer saltar las alarmas, a la vez que avisa al propietario del inmueble y las autoridades.

b) Simulación de presencia. Gestión del control de acceso y control de presencia, así como la simulación de presencia.

c) Detección de conatos de incendio, fugas de gas, escapes de agua. Mediante el nodo telefónico, se puede tener acceso (mediante un pulsador radio-

frecuencia que se lleve encima por ejemplo) a los servicios de Policía, etc. A través del nodo telefónico es posible desviar la alarma hacia los bomberos, por ejemplo.

d) Servicios de información, tele-compra, tele-banco, alerta médica (Tele-asistencia), etc. Para ciertos colectivos estos servicios pueden ser de gran utilidad (por ejemplo, unidades familiares donde ambos cónyuges trabajan) ya que producen un ahorro de tiempo.

e) Cerramiento de persianas puntual y seguro.

f) Se puede detectar averías en los accesos, en los ascensores, etc.

En el ámbito de las comunicaciones.

a) Control Remoto.

- **Dentro de la vivienda:** a través de un esquema de comunicación con los distintos equipos (mando a distancia, bus de comunicación, etc.). Reduce la necesidad de moverse dentro de la vivienda, este hecho puede ser particularmente importante en el caso de personas de la tercera edad o minusválidos.

- **Fuera de la vivienda:** presupone un cambio en los horarios en los que se realizan las tareas domésticas (por ejemplo: la posibilidad de que el usuario pueda activar la cocina desde el exterior de su vivienda, implica que previamente ha de preparar los alimentos) y como consecuencia permite al usuario un mejor aprovechamiento de su tiempo.

b) Transmisión de alarmas.

c) Intercomunicaciones entre las habitaciones.

d) Telefonía IP. Las comunicaciones de voz por Internet utiliza la conexión a Internet como red de transporte de los datos, para realizar una comunicación *VoIP* (Voice Over IP). Se puede realizar las llamadas desde el ordenador personal hasta otro PC remoto o bien hasta cualquier tipo de teléfono, basta con disponer de un PC, conexión a Internet, un equipo multimedia (altavoces y micrófono) y el software necesario para ello. Pero también existe la posibilidad de integrar, o hasta sustituir la telefonía tradicional

con la telefonía IP. Como *terminal* para realizar las llamadas se puede utilizar por parte del que tiene contratado el servicio:

- El PC, aprovechando los altavoces y micrófonos internos o externos.
- El PC, con un teléfono especial conectado al puerto USB.
- Un teléfono normal conectado a un hub que a su vez está conectado a un router.
- Un Teléfono o SmartPhone dotado de tecnología WiFi que directamente integra el software de telefonía IP.



Figura 3. Esquema conceptual VoIP

Tal vez la ventaja más tangible para los usuarios finales radique en el método de facturación de estas llamadas. Mientras que las operadoras telefónicas tradicionales suelen tarificar las comunicaciones según la distancia y el tiempo de conexión, el coste de una llamada por Internet puede no depender de la lejanía del interlocutor. Pero aun siendo dependiente de su destino, la comunicación tendrá siempre un valor considerablemente más reducido que el de una llamada telefónica habitual, pudiendo en algunos casos ser gratis.

8.1.1.4. Seguridad

Las aplicaciones de seguridad contemplan tanto la seguridad de las personas como la seguridad de los bienes materiales. Hay que destacar:

- El *sistema anti-intrusión*: existen dos tipos de sistemas anti-intrusión que son los siguientes:

1) Protección perimetral: protege de accesos a la parcela y a la misma vivienda a través de puertas y ventanas. Principalmente se utiliza barreras infrarrojas de exterior en vallas, el jardín y ventanas y puertas; y sensores de contacto magnético de puerta/ventana y sensores de rotura de cristal.

2) Protección del interior: protege de intrusión dentro de la misma vivienda. Se utiliza normalmente sensores de detección de movimiento con tecnologías infrarroja y ultrasónica.

- La *alarma técnica* consiste en un conjunto de detectores (incendio, gas, CO, inundación, etc.) que en caso de producirse fuegos o escapes de fluidos, posibles errores en la instalación o mal uso involuntario detectan la anomalía y proceden al corte del suministro mediante la válvula de fluidos, la desconexión de los circuitos eléctricos y otras actuaciones previstas. También pueden configurarse para que den un aviso de entre los siguientes:

1) Local: sirenas, timbres, luces, mensajes hablados etc.

2) Remoto: a las Centrales Receptoras de Alarmas y/o al usuario final directamente, a través del teléfono convencional, móvil, correo electrónico o similar.

Además la casa puede estar preparada para actuar automáticamente según la alarma, adicionalmente del aviso, como por ejemplo:

- Si hay un escape de agua corta el suministro de agua con la electro-válvula de agua.
- Si hay un escape de gas corta el suministro de gas con la electro-válvula de gas.

- Si hay humo puede subir o bajar persianas según necesidad preprogramada.

También pueden avisar de fallo de suministro eléctrico, fallo de línea telefónica, etc. Y con algunos sistemas modernos se puede realizar una *reconfiguración* automática para volver a activar algunos sistemas.

- La *simulación de presencia* consiste en la activación aleatoria de algunos elementos como la iluminación y las persianas en los periodos en que la vivienda no está ocupada, creando una sensación de actividad en su interior.
- Las *alarmas personales*: existen dos tipos de alarmas personales destinadas al hogar personal.
 - *Aviso SOS o pánico*: se utiliza en casos de emergencias graves tipo en casos que hay intrusos, de robo o ataques personales realizados dentro o justo fuera de la vivienda.
 - *Avisos de asistencia*: se utiliza para llamar la atención de necesidad de asistencia personal principalmente para personas de tercera edad o gente discapacitada.

Las interfaces suelen ser botones en las mismas centrales de seguridad o botones en los llaveros, y para avisos de asistencia hay pulsadores de forma reloj o colgante que manda una señal vía radio en caso de caída o al encontrarse mal.

8.1.1.5. Confort (Automatismos)

El campo de automatización de sistemas e instalaciones eléctricas es muy amplio, dependiendo de las necesidades e ideas del usuario, como más significativos destacamos:

- Accionamiento automático de persianas para evitar los efectos atmosféricos (viento, lluvia, nieve, etc.) sobre estos elementos para limitar sus pérdidas térmicas.
- Accionamiento automático de la iluminación en función del nivel de iluminación y de la presencia de personas.

- Centralización y supervisión de la información del estado de los sistemas instalados.
- Posibilidad de agrupar un conjunto de acciones en una sola orden. Por ejemplo: apagar todas las luces, cerrar las persianas y cortar el suministro de agua, gas y poner el sistema de alarma en estado de vigilancia con un mismo interruptor con llave.
- Utilización de mandos a distancia para las diferentes instalaciones



Figura 4. Dispositivo de mando asociado

8.1.1.6. Previsión del futuro

La penetración imparable de la tecnología en los hogares, como consecuencia de la disminución de precios, la competencia y el ciclo de vida de estos productos, hará mucho más posible y accesible la domótica para todos.



Figura 5. Las casas del mañana

Si bien existe normativa suficiente para distintos ámbitos del hogar digital y conectado que pretende la domótica (REBT, reciente ICT, etc.), no existen normas específicas que guíen la actuación de este conglomerado de actores que se dan cita en el sector, lo que, en opinión de algunos expertos, se solucionaría con un reglamento concreto.

El concepto de *ambiente inteligente* muestra una visión de la *sociedad de la información* en el que se enfatiza la facilidad de uso, el soporte eficiente de los servicios y la posibilidad de mantener interacciones naturales con el ser humano. El objeto central se materializa a grandes rasgos en un individuo rodeado de interfaces inteligentes e intuitivas que se encuentran integradas en partes y objetos corrientes, todo esto en un entorno que sea capaz de reconocer y responder a la presencia y necesidades de diferentes individuos, de una forma completamente discreta e imperceptible más que a través de los resultados. Un entorno por otra parte, que no se limita a ningún lugar físico determinado, sino que comprende a todos ellos: la casa, el coche, el lugar de trabajo, etc.

8.1.2. Arduino

Arduino es una compañía de hardware libre, y comunidad tecnológica, que diseña y manufactura placas de desarrollo de hardware y software compuesta respectivamente por circuitos impresos que integran un microcontrolador, y un entorno de desarrollo (IDE) en donde se programa cada placa. Arduino se

enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios. Toda la plataforma, tanto para sus componentes de hardware como de software son liberados bajo licencia de código abierto que permite libertad de acceso a los mismos.

El hardware consiste en una placa de circuito impreso con un microcontrolador, usualmente Atmel AVR, puertos digitales y analógicos de entrada/salida, los cuales pueden conectarse a placas de expansión (shields) que amplían las características de funcionamiento de la placa Arduino. Asimismo posee un puerto de conexión USB desde donde se puede alimentar la placa y establecer comunicación serial con el computador.

Por otro lado, el software consiste en un entorno de desarrollo (IDE) basado en el entorno de Processing y lenguaje de programación basado en Wiring, así como en el cargador de arranque (bootloader) que es ejecutado en la placa.⁴ El microcontrolador de la placa se programa a través de un computador, haciendo uso de comunicación serial mediante un convertidor de niveles RS-232 a TTL serial.

La primera placa Arduino fue introducida en el 2005, ofreciendo un bajo costo y facilidad de uso para novatos y profesionales buscando desarrollar proyectos interactivos con su entorno mediante el uso de actuadores y sensores. A partir de octubre de 2012, se incorporaron nuevos modelos de placas de desarrollo que hacen uso de microcontroladores Cortex M3, ARM de 32 bits, que coexisten con los originales modelos que integran microcontroladores AVR de 8 bits. ARM y AVR no son plataformas compatibles en cuanto a su arquitectura y por lo tanto su set de instrucciones, pero se pueden programar y compilar bajo el IDE predeterminado de Arduino sin ningún cambio.

Las placas Arduino están disponibles de forma ensambladas o en forma de Kits "Hazlo tú mismo" (por sus siglas en inglés <<DIY>>). Los esquemáticos de diseño del Hardware están disponibles bajo licencia Libre, permitiendo a cualquier persona crear su propia placa Arduino sin necesidad de comprar una prefabricada. Adafruit Industries estimó a mediados del año 2011 que alrededor de 300,000 placas arduinos habían sido producidas comercialmente, y en el año 2013 estimó que alrededor de 700.000 placas oficiales de la empresa Arduino estaban en manos de los usuarios.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software tal como Adobe Flash, Processing, Max/MSP, Pure Data, etc. Una tendencia tecnológica es utilizar Arduino como tarjeta de adquisición de datos desarrollando interfaces en software como JAVA, Visual Basic y LabVIEW. Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente.

¿Por qué Arduino?

Hay muchos otros microcontroladores y plataformas con microcontroladores Disponibles para la computación física. Parallax Basic Stamp, BX-24 de Netmedia, Phidgets, Handyboard del MIT, y muchos otros ofrecen funcionalidades similares.

Todas estas herramientas organizan el complicado trabajo de programar un Microcontrolador en paquetes fáciles de usar. Arduino, además de simplificar el Proceso de trabajar con microcontroladores, ofrece algunas ventajas respecto a otros sistemas a profesores, estudiantes y amateurs:

- **Asequible** - Las placas Arduino son más asequibles comparadas con otras plataformas de microcontroladores. La versión más cara de un módulo de Arduino puede ser montada a mano, e incluso ya montada cuesta bastante menos de 50\$us. Lo único que “vale” en la placa son los componentes, ya que no debemos pagar el costo de la licencia de su creador, por el hecho de ser libre.
- **Multi-Plataforma** - El software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux. La mayoría de los entornos para microcontroladores están limitados a Windows.
- **Entorno de programación simple y directo** - El entorno de programación

de Arduino es fácil de usar para principiantes y lo suficientemente flexible para los usuarios avanzados. Pensando en los profesores, Arduino está basado en el entorno de programación de Processing con lo que el estudiante que aprenda

a programar en este entorno se sentirá familiarizado con el entorno de desarrollo Arduino.

- **Software ampliable y de código abierto** - El software Arduino está publicado bajo una licencia libre y preparado para ser ampliado por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++, y si se está interesado en profundizar en los detalles técnicos, se puede dar el salto a la programación en el lenguaje AVR C en el que está basado. De igual modo se puede añadir directamente código en AVR C en tus programas si así lo deseas.
- **Hardware ampliable y de Código abierto** - Arduino está basado en los microcontroladores ATMEGA168, ATMEGA328 y ATMEGA1280. Los planos de los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo, ampliándolo u optimizándolo. Incluso usuarios relativamente inexpertos pueden construir la versión para placa de desarrollo para entender cómo funciona y ahorrar algo de dinero.

8.1.2.1. Hardware

Los modelos de Arduino se categorizan en: placas de desarrollo, placas de expansión (*shields*), kits, accesorios e impresoras 3d.

Placas

Arduino Nano, Arduino Uno, Arduino Leonardo, Arduino Due, Arduino Yún, Arduino Tre (En Desarrollo), Arduino Zero, Arduino Micro, Arduino Esplora, Arduino Mega ADK, Arduino Ethernet, Arduino Mega 2560, Arduino Robot, Arduino Mini, LilyPad Arduino Simple, LilyPad Arduino SimpleSnap, LilyPad Arduino, LilyPad Arduino USB, Arduino Pro Mini, Arduino Fio, Arduino Pro, Arduino MKR1000/Genuino MKR1000, Arduino MICRO/Genuino MICRO, Arduino 101/Genuino 101, Arduino Gemma.

Placas de expansión (*shields*)

Arduino GSM Shield, Arduino Ethernet Shield, Arduino WiFi Shield, Arduino Wireless SD Shield, Arduino USB Host Shield, Arduino Motor Shield, Arduino Wireless Proto Shield, Arduino Proto Shield.

Kits

The Arduino Starter Kit, Arduino Materia 101.

Accesorios

TFT LCD Screen, USB/Serial Light Adapter, Arduino ISP, Mini USB/Serial Adapter.

Impresoras 3d

Arduino Materia 101.

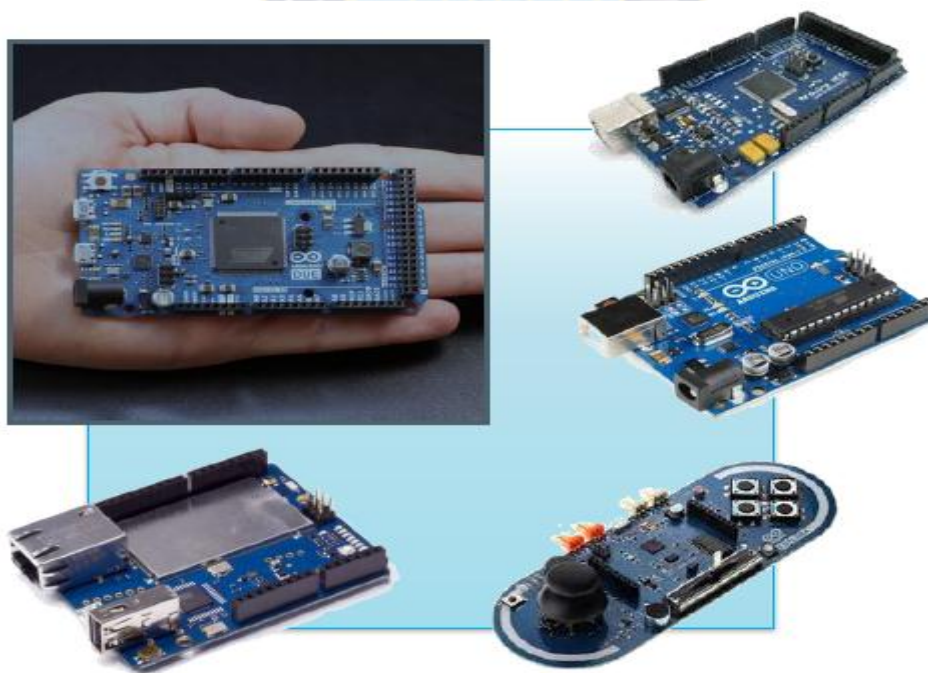


Figura 6. Placas Arduino

8.1.2.2. Aplicaciones

La plataforma Arduino ha sido usada como base en diversas aplicaciones electrónicas:

- Xoscillo: Osciloscopio de código abierto.
- Equipo científico para investigaciones.
- Arduinome: Un dispositivo controlador MIDI.
- OBDuino: un económetro que usa una interfaz de diagnóstico a bordo que se halla en los automóviles modernos.
- SCA-ino: Sistema de cómputo automotriz capaz de monitorear sensores como el TPS, el MAP y el O2S y controlar actuadores automotrices como la bobina de ignición, la válvula IAC y aceleradores electrónicos.
- Humane Reader: dispositivo electrónico de bajo coste con salida de señal de TV que puede manejar una biblioteca de 5000 títulos en una tarjeta microSD.

The Humane PC: equipo que usa un módulo Arduino para emular un computador personal, con un monitor de televisión y un teclado para computadora.

- Ardupilot: software y hardware de aeronaves no tripuladas.
- ArduinoPhone: un teléfono móvil construido sobre un módulo Arduino.
- Máquinas de control numérico por computadora (CNC).
- Open Theremín Uno: Versión digital de hardware libre del instrumento Theremín.
- Impresoras 3D.

8.1.2.3. Esquemas de conexiones

Entradas y salidas

Poniendo de ejemplo al módulo Diecimila, este consta de 14 entradas digitales configurables como entradas y/o salidas que operan a 5 voltios. Cada contacto puede proporcionar o recibir como máximo 40 mA. Los contactos 3, 5, 6, 9, 10 y 11 pueden proporcionar una salida PWM (Pulse Width Modulation). Si se conecta cualquier cosa a los contactos 0 y 1, eso interferirá con la comunicación USB. Diecimila también tiene 6 entradas analógicas que proporcionan una resolución de 10 bits. Por defecto, aceptan de 0 hasta 5

voltios, aunque es posible cambiar el nivel más alto, utilizando el contacto Aref y algún código de bajo nivel.

Tabla 1. Módulos Arduinos

PLACAS ARDUINO	CARACTERÍSTICAS
	<p>Arduino UNO R3: Es un módulo que se basa en el ATMEGA328. Tiene 14 entradas/salidas, de las cuales 6 se pueden usar como PWM, 6 salidas analógicas, un cristal de 16 MHz, una conexión USB, un jack de alimentación, un botón de reset y un "header" ICSP. El Arduino UNO R3 ya no lleva el FTDI para la conexión USB, pero usa el ATMEGA16U2.</p>
	<p>Arduino Leonardo: Se basa en el micro ATMEGA32u4. Cuenta con 20 entradas/salidas, de las cuales 7 pueden ser salidas PWM y 12 entradas analógicas. Como oscilador tiene un cristal de 16MHz.</p>
	<p>Arduino Diecimila: Esta es la placa Arduino más popular. Se conecta al ordenador con un cable estándar USB y contiene todo lo que necesitas para programar y usar la placa. Puede ser ampliada con variedad de dispositivos y/o escudos (shields) con características específicas.</p>
	<p>Arduino Nano: Una placa compacta diseñada para uso como tabla de pruebas, el Nano se conecta al ordenador usando un cable USB Mini-B.</p>

	<p>Arduino Bluetooth: <i>Arduino BT contiene un módulo bluetooth que permite comunicación y programación sin cables. Es compatible con los dispositivos Arduino.</i></p>
	<p>Arduino LilyPad: <i>Diseñada para aplicaciones listas para llevar, esta placa puede ser conectada en fábrica, y un estilo sublime.</i></p>
	<p>Arduino Mini: <i>Esta es la placa más pequeña de Arduino. Trabaja bien en tabla de pruebas o para aplicaciones en las que prima el espacio. Se conecta al ordenador usando el cable Mini USB.</i></p>
	<p>Arduino Serial: <i>Es una placa básica que usa RS232 como un interfaz con el ordenador para programación y comunicación. Esta placa es fácil de ensamblar incluso como ejercicio de aprendizaje.</i></p>
	<p>Arduino Esplora: <i>Es un controlador listo para usarse. Te permite explorar las infinitas posibilidades que tiene en el mundo de sensores y se puede comenzar a trabajar con una palanca de mando tipo joystick y con los sensores de luz y temperatura.</i></p>
	<p>Arduino Mega ADK: <i>Esta placa está basada en el Arduino MEGA 2560 y dispone de un puerto USB Host incorporado. Arduino MEGA ADK es una placa basada en microcontrolador ATMEGA2560. Dispone de un puerto USB host interfaz para conectarlo con dispositivos basados en Android basados en el chip MAX3421e. Dispone de 54 pines I/O (14 de los cuales con PWM), 16 entradas analógicas, 4 puertos UARTs (por hardware), un cristal de 16 MHz, un conector USB, un conector de alimentación Jack, pines para ICSP y un botón de RESET.</i></p>

8.1.2.4. Lenguaje de programación Arduino

Arduino está basado en C y soporta todas las funciones del estándar C y algunas de C++.

El entorno de programación IDE programa la tarjeta Arduino desde un ambiente gráfico que permite para fines didácticos avanzar con mayor velocidad.

8.1.2.5. Ventajas del IDE Arduino

Tiene un lenguaje simple, basado en C/C++.

- Permite desde un primer contacto estar programando directamente al hardware.
- Es un proyecto open-source, por lo que debido a su precio podemos probar y experimentar sobre la misma tarjeta.
- Tiene una comunidad de desarrollo alrededor que permite un acceso a referencias, ejemplos, proyectos de gran ayuda.

8.1.2.6. Primera ejecución del IDE Arduino y configuración inicial para el desarrollo del proyecto

Tras instalar el controlador de Arduino, ahora realizaremos una primera ejecución para

configurar el IDE de desarrollo, pulsaremos con el botón derecho del ratón sobre

el ejecutable "arduino.exe" de la carpeta descomprimida anteriormente y

seleccionaremos "Ejecutar como administrador" (no es necesario, pero así

evitamos posibles problemas de limitaciones de Windows 7):

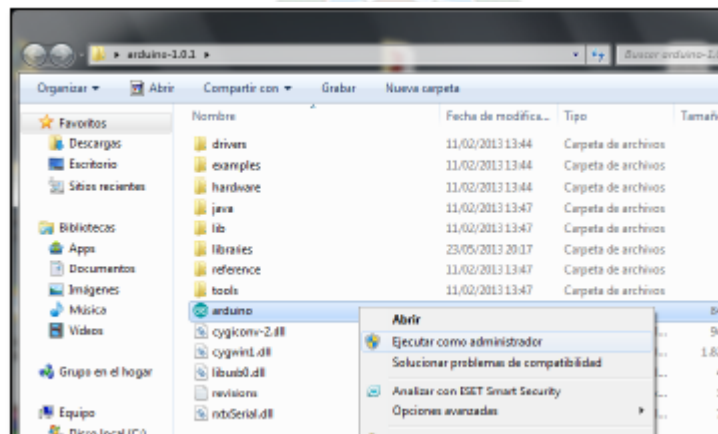


Figura 7. Primer paso de Ejecución del IDE Arduino como Administrador

En el IDE de desarrollo de Arduino, en primer lugar seleccionaremos el tipo de dispositivo (Tarjeta). Para ello pulsaremos en el menú "Herramientas" - "Tarjeta" y seleccionaremos "Arduino Nano":

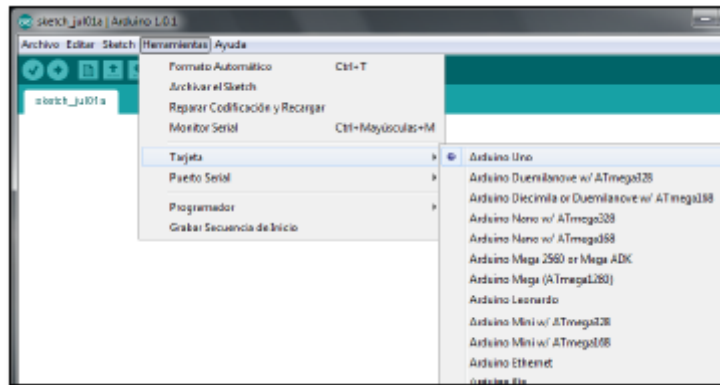


Figura 8. Selección de tarjeta Arduino Nano

Seleccionaremos también el puerto serie asignado al controlador de Arduino (en nuestro caso COM1), para ello accederemos al menú "Herramientas" - "Puerto Serial" - "COM1":

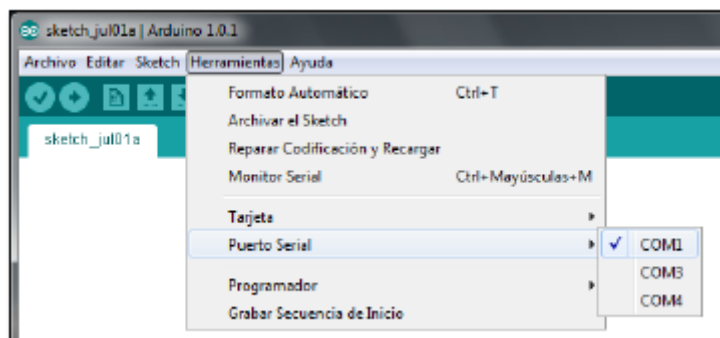


Figura 9. Selección del puerto a utilizar

8.1.2.7. Opciones de la interfaz



Verificar/Compilar. Revisa el código, identificando los errores



Crea un nuevo proyecto



Presenta un menú para abrir proyectos previamente guardados y también muestra los proyectos de ejemplos.



Guarda el proyecto actual dentro de la carpeta sketch/. Se puede guardarlo también con un nombre distinto por el menú Archivo → Guardar como



Descarga el programa compilado desde el PC hasta la tarjeta Arduino



Realiza el monitoreo del puerto serial, visualiza la data enviada desde la tarjeta Arduino.

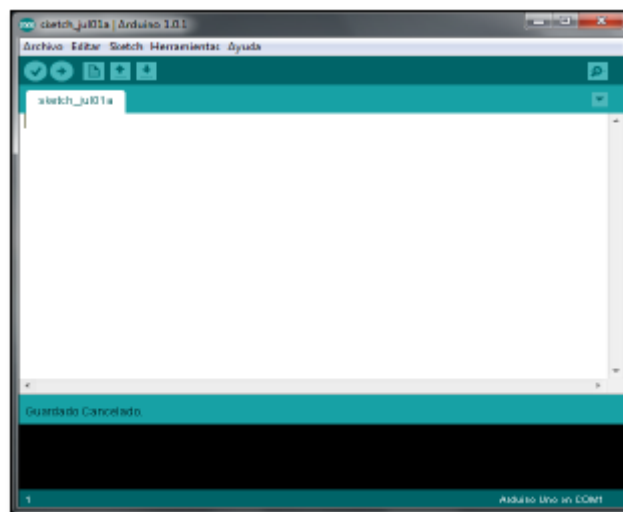


Figura 10. IDE de desarrollo Arduino v.1.0.1.

8.1.2.8. Arduino NANO

Es un modelo pequeño pero completo, que incorpora el mismo controlador que la Arduino UNO, el ATmega328. Carece de adaptador de corriente Jack, ya que se alimenta a través del cable mini-B USB, que sirve también de

comunicación (llevada a cabo por el chip USB FTDI). Las características de memoria son las mismas que la placa UNO, exceptuando el tamaño del bootloader que se ve incrementado a 2 KB debido al uso del chip FTDI USB

Podemos considerarla una placa Arduino UNO de reducido tamaño, 4,30 cm de largo por 1,85 cm de ancho.

Cuenta con 14 entradas digitales (4 de PWM) y 6 analógicas, además de pines, interrupciones y otros puertos.

8.1.2.8.1. Especificaciones

Tabla 2. Especificaciones de Arduino Nano

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz

8.1.2.8.2. Alimentación

La tarjeta Arduino Nano puede ser alimentada a través de la conexión USB tipo mini B, una fuente de alimentación externa no regulada de 6- 20V (pin 30), o una fuente de alimentación externa regulada de 5V (pin 27). La fuente de alimentación se selecciona automáticamente por el voltaje más alto de la fuente. El chip FTDI FT232RL en el Nano sólo se enciende si la tarjeta está siendo alimentada a través de USB. Como resultado, cuando está funcionando con una alimentación externa (no USB), la salida de 3.3V (la cual es soportada

por el chip FTDI) no está disponible y los leds RX y TX parpadearán si los pines digitales 0 o 1 están en alto.

8.1.2.8.3. Memoria

El ATmega168 tiene 16 KB de memoria flash para almacenar el código (de los cuales 2 KB se utiliza para el gestor de arranque), el ATmega328 tiene 32 KB, (también con 2 KB utilizados para el gestor de arranque). El ATmega168 tiene 1 KB de SRAM y 512 bytes de EEPROM (puede ser leído y escrito con la librería EEPROM), el ATmega328 tiene 2 KB de SRAM y 1 KB de memoria EEPROM.

8.1.2.8.4. Entradas y Salidas

Cada uno de los 14 pines digitales de la tarjeta Arduino Nano puede ser utilizado como una entrada o salida, usando las funciones `pinMode()`, `digitalWrite()`, y `digitalRead()`. Los pines operan a 5 voltios. Cada pin puede proporcionar o recibir una corriente máxima de 40 mA y tienen una resistencia pull-up interna (desconectado por defecto) de 20 a 50 kΩ. Además, algunos pines tienen funciones específicas:

- a) Serie: 0 (RX) y 1 (TX). Se utiliza para recibir (RX) y transmitir (TX) datos serie TTL. Estos pines están conectados a los pines correspondientes del chip FTDI USB-to-TTL.
- b) Interrupciones externas: 2 y 3. Estos pines pueden ser configurados para disparar una interrupción en un valor bajo, un borde ascendente o descendente o un cambio en el valor. Para realizar esto se utiliza la función `attachInterrupt()`.
- c) PWM: 3, 5, 6, 9, 10 y 11. Proporcionar 8 bits de salida PWM con la función `analogWrite()`.
- d) SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines soportan la comunicación SPI, que, a pesar de ser proporcionado por el hardware fundamental, actualmente no está incluido en el lenguaje de Arduino.
- e) LED: 13. Hay un LED incorporado conectado al pin digital 13. Cuando el pin está en ALTO, el LED está encendido, cuando el pin está en BAJO, se apaga el LED.

La tarjeta Arduino tiene 8 entradas analógicas, cada una de las que proporcionan 10 bits de resolución (es decir 1024 valores diferentes). Por defecto miden desde 0 a 5 voltios, aunque es posible cambiar el extremo superior de su rango utilizando la función `analogReference()`. Además, algunos de estos pines tienen funciones específicas:

a) I2C: 4 (SDA) y 5 (SCL). Soportan la comunicación I2C (TWI) utilizando la librería `Wire`.

b) AREF: Voltaje de referencia para las entradas analógicas. Se utiliza la función `analogReference()`.

c) RESET: Provoca un valor BAJO en la línea que reinicia al microcontrolador. Normalmente se utiliza para añadir un botón de reinicio en la placa.

8.1.2.8.5. Comunicaciones

La tarjeta Arduino Nano tiene una serie de facilidades para comunicarse con un ordenador a diferencia de otras tarjetas Arduino u otros microcontroladores. El ATmega168 y ATmega328 ofrecen una comunicación serie UART TTL (5V), que está disponible en los pines digitales 0 (RX) y 1 (TX). Un FT232RL FTDI en la placa proporciona los canales de comunicación serie a través de USB y los controladores de FTDI (incluido con el software de Arduino) proporcionan un puerto COM virtual para el software del ordenador. El software de Arduino incluye un monitor serie que permite visualizar los datos de texto que se envía desde y hacia la placa Arduino. Los leds RX y TX en la placa parpadean cuando se están transmitiendo datos a través del chip FTDI y de la conexión USB a la computadora (pero no parpadean para la comunicación serie de los pines 0 y 1). La biblioteca `SoftwareSerial` permite la comunicación serie en cualquiera de los pines digitales de la tarjeta Arduino Nano. El ATmega168 y ATmega328 también soportan I2C (TWI) y la comunicación SPI. El software de Arduino incluye una librería `Wire` para simplificar el uso del bus I2C.

8.1.2.8.6. Programación

La tarjeta Arduino Nano puede ser programada con el software de Arduino. Seleccione en el menú Herramientas > Placas "Arduino Diecimila, Duemilanove o Nano w/ATmega168" o "Arduino Duemilanove Nano o w/ATmega328" (de

acuerdo con el microcontrolador de la placa). El ATmega168 o ATmega328 en la Arduino Nano viene pre-grabado con un gestor de arranque que le permite cargar nuevo código a la misma sin el uso de un programador de hardware externo. Se comunica utilizando el protocolo original STK500 (de referencia, archivos de cabecera C). También se puede pasar por alto el gestor de arranque y programar el microcontrolador a través de la cabecera ICSP.

El programa que se diseñó en la tarjeta Arduino Nano realiza las siguientes tareas:

- Graba la hora y la fecha actualizada en un Real-Time-Clock DS1307 mediante la interfaz de comunicación I2C.
- Muestra los datos de los sensores de corriente ACS712 20A tomando 5000 muestras cada 500 milisegundos.

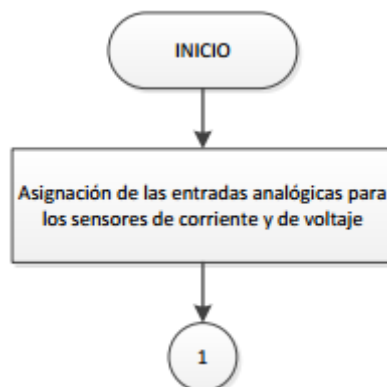
Acondiciona digitalmente la señal de los sensores de corriente ACS712 20 A en un rango de 0A a 20A.

- Acondiciona digitalmente la señal del sensor de voltaje en un rango que va desde 30V hasta 170V.

- Calcula los kWh.

- Presenta en la pantalla LCD los valores del voltaje, la corriente y los kWh. Además presenta la hora y fecha actualizada.

8.1.2.8.7. Diagrama de flujo del programa de la tarjeta NANO



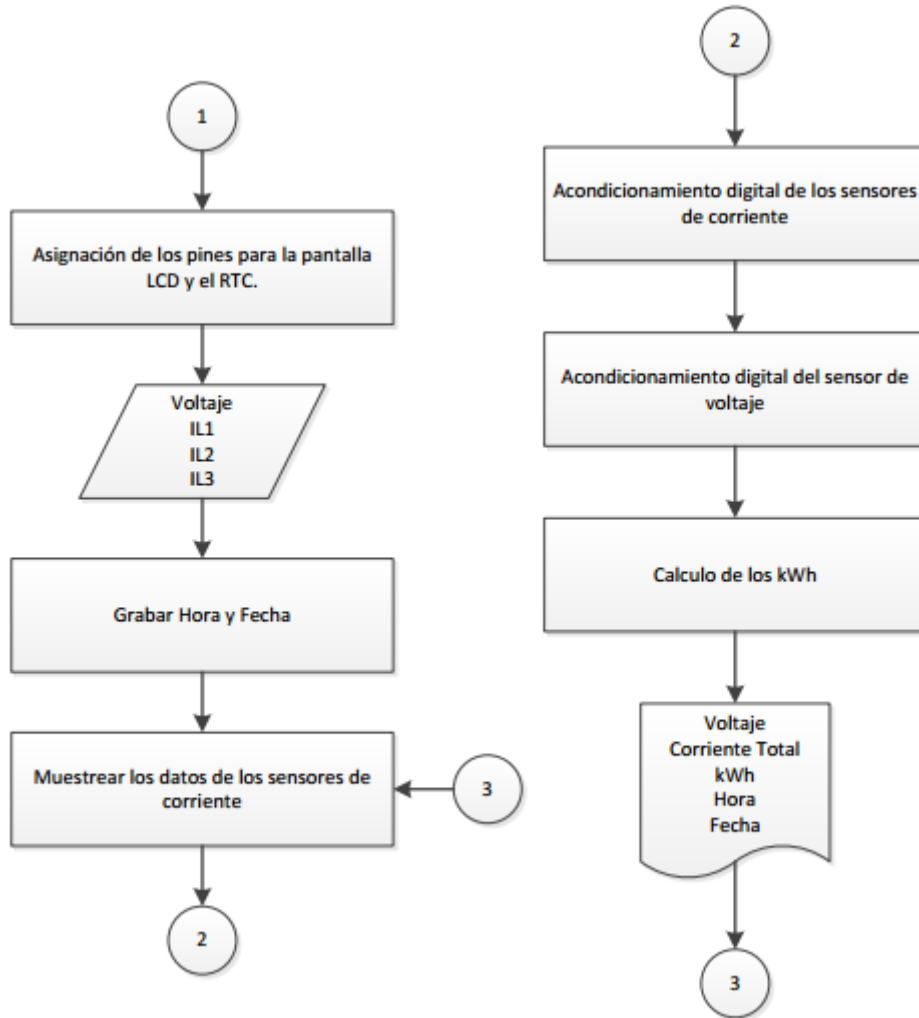


Figura 11. Diagrama de flujo del programa de la tarjeta NANO

8.1.2.8.8. Reinicio automático (Software)

En lugar de requerir un pulso físico del botón de reinicio antes de que se cargue el sketch en la placa, La tarjeta Arduino Nano está diseñada de una manera que le permite ser reiniciada mediante el software que se ejecuta cuando está conectada a un ordenador. Una de las líneas de control de flujo de hardware (DTR) del FT232RL está conectado a la línea de reinicio del ATmega168 o ATmega328 a través de un condensador de 100 nano faradios. Cuando se asegura esta línea, la línea de reinicio cae lo suficiente como para restablecer el chip. El software de Arduino utiliza esta capacidad de la tarjeta para cargar el código con sólo pulsar el botón "UPLOAD" en el entorno de Arduino. Esto significa que el gestor de arranque puede tener un tiempo de

espera más corto, como la reducción de DTR puede ser bien coordinada con el inicio de la carga. Esta configuración tiene otras implicaciones. Cuando la tarjeta Arduino Nano se conecta a un ordenador con Mac OS X o Linux, se pone a cero cada vez que se realiza una conexión a ella desde el software (a través de USB). Para el siguiente medio segundo más o menos, el gestor de arranque se ejecuta en la tarjeta Arduino Nano. La tarjeta está programada para ignorar los datos con formato incorrecto interceptando los primeros bytes de datos enviados a la placa después de que una conexión se abre. Si un sketch se está ejecutando en la placa recibe una configuración en tiempo u otros datos cuando se inicia por primera vez, asegúrese de que el software con el que se comunica espere un segundo después de abrir la conexión y antes de enviar estos datos.

8.1.2.8.9. Referencias del lenguaje

Los programas hechos con Arduino se dividen en tres partes principales: *estructura*, *valores* (variables y constantes) y *funciones*. El Lenguaje de programación Arduino se basa en C/C++.

Tabla 3. Variables utilizadas en Arduino

VARIABLES	unsigned int (entero sin signo)	array (cadena)	Conversión
Tipos de Datos	long (entero 32b)	void (vacío)	char()
boolean (booleano)	unsigned long (entero 32b sin signo)	Constantes	byte()
char (carácter)	float (en coma flotante)	HIGH LOW	int()
byte	double (en coma flotante de 32b)	INPUT OUTPUT	long()
int (entero)	string (cadena de caracteres)	true false	float()

Tabla 4. Estructuras utilizadas en Arduino

ESTRUCTURAS	<code>continue</code> (continuación en bloque de código)	<code>*</code> (multiplicación)	<code>></code> (mayor que)
<code>setup()</code> (inicialización)	<code>return</code> (devuelve valor a programa)	<code>/</code> (división)	<code><=</code> (menor o igual que)
<code>loop()</code> (bucle)	Sintaxis	<code>%</code> (resto)	<code>>=</code> (mayor o igual que)
Estructuras de control	<code>;</code> (punto y coma)	Operadores Booleanos	Operadores de Composición
<code>if</code> (comparador si-entonces)	<code>{}</code> (llaves)	<code>&&</code> (y)	<code>++</code> (incrementa)
<code>if...else</code> (comparador si...sino)	<code>//</code> (comentarios en una línea)	<code> </code> (o)	<code>--</code> (decrementa)
<code>for</code> (bucle con contador)	<code>/* */</code> (comentarios en múltiples líneas)	<code>!</code> (negación)	<code>+=</code> (composición suma)
<code>switch case</code> (comparador múltiple)	Operadores Aritméticos	Operadores Comparativos	<code>-=</code> (composición resta)
<code>while</code> (bucle por comparación booleana)	<code>=</code> (asignación)	<code>==</code> (igual a)	<code>*=</code> (composición multiplicación)
<code>do... while</code> (bucle por comparación booleana)	<code>+</code> (suma)	<code>!=</code> (distinto de)	<code>/=</code> (composición división)
<code>break</code> (salida de bloque de código)	<code>-</code> (resta)	<code><</code> (menor que)	

Tabla 5. Funciones utilizadas en Arduino

FUNCIONES		Matemáticas	Trigonometría
<code>delay()</code>			
E/S Avanzadas	<code>delayMicroseconds()</code>	<code>min()</code> (mínimo)	<code>sin()</code> (seno)
<code>tone()</code>	E/S Digitales	<code>max()</code> (máximo)	<code>cos()</code> (coseno)
<code>noTone()</code>	<code>pinMode()</code>	<code>abs()</code> (valor absoluto)	<code>tan()</code> (tangente)
<code>shiftOut()</code>	<code>digitalWrite()</code>	<code>constrain()</code> (limita)	Números Aleatorios
<code>pulseIn()</code>	<code>digitalRead()</code>	<code>map()</code> (cambia valor de rango)	<code>randomSeed()</code>
Tiempo	E/S Analógicas	<code>pow()</code> (eleva a un número)	<code>random()</code>
<code>millis()</code>	<code>analogRead()</code>	<code>sq()</code> (eleva al cuadrado)	Comunicación
<code>micros()</code>	<code>analogWrite()</code> - PWM(modulación por ancho de pulso)	<code>sqrt()</code> (raíz cuadrada)	Serial

8.1.2.8.10. Librerías

Las librerías proveen funcionalidad extra a nuestro programa, por ejemplo: al trabajar con hardware o al manipular datos. Para usar una librería dentro de una programación, puedes seleccionarla desde **Sketch > Importar Librería**

Si deseas usar librerías que no vienen junto con Arduino, necesitarás instalarlas. Para hacerlo, descarga la librería y descomprímela. Debería localizarse en una carpeta propia, y normalmente, contener dos archivos, uno con sufijo ".h" y otro con sufijo ".cpp". Abra su carpeta de Arduino, existe una carpeta llamada "*libraries*», coloca la carpeta de la librería ahí dentro.

Reiniciar el IDE de Arduino para encontrar la nueva librería en el menú **Sketch > Importar Librería**.

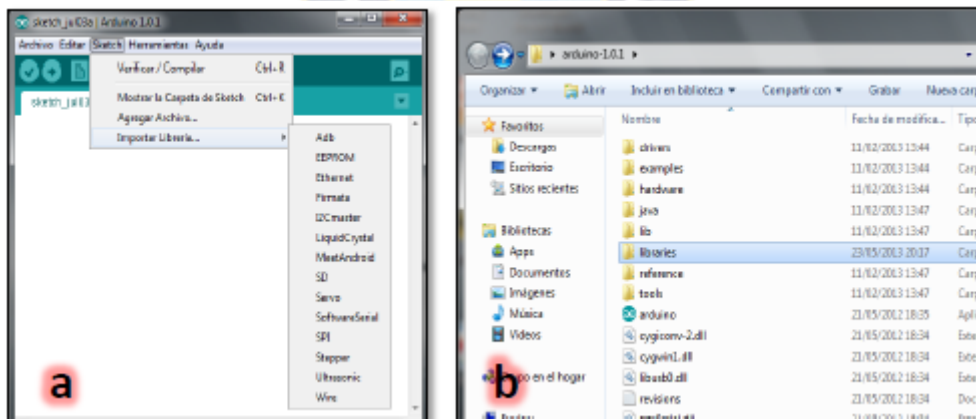


Figura 12. Librerías de IDE de Arduino

8.1.2.8.11. Librerías estándar

- EEPROM - Para leer y escribir en memorias "permanentes".
- Ethernet - Para conectar a internet usando el *Ethernet Shield*.
- Firmata - Para comunicarse con aplicaciones en la computadora usando un protocolo estándar *Serial*.
- LiquidCrystal - Para controlar Displays de cristal líquido (*LCD*)
- Servo - Para controlar *servomotores*
- SoftwareSerial - Para la comunicación serial de cualquier pin digital.
- Stepper - Para controlar motores paso a paso (*Stepper motors*)

- Wire - Interfaz de dos cables, o *Two Wire Interface (TWI/I2C)*, para enviar y recibir datos a través de una red de dispositivos y sensores.

8.1.2.9. Microcontrolador ATMEGA 328

El ATmega328p es un chip microcontrolador creado por Atmel y pertenece a la serie megaAVR.

El Atmega328 AVR 8-bit es un Circuito integrado de alto rendimiento que está basado un microcontrolador RISC, combinando 32 KB ISP flash una memoria con la capacidad de leer-mientras-escribe, 1 KB de memoria EEPROM, 2 KB de SRAM, 23 líneas de E/S de propósito general, 32 registros de proceso general, tres temporizadores flexibles/contadores con modo de comparación, interrupciones internas y externas, programador de modo USART, una interface serial orientada a byte de 2 cables, SPI puerto serial, 6-canales 10-bit Conversor A/D (canales en TQFP y QFN/MLF packages), "watchdog timer" programable con oscilador interno, y cinco modos de ahorro de energía seleccionables por software. El dispositivo opera entre 1.8 y 5.5 voltios. Por medio de la ejecución de poderosas instrucciones en un solo ciclo de reloj, el dispositivo alcanza una respuesta de 1 MIPS, balanceando consumo de energía y velocidad de proceso.

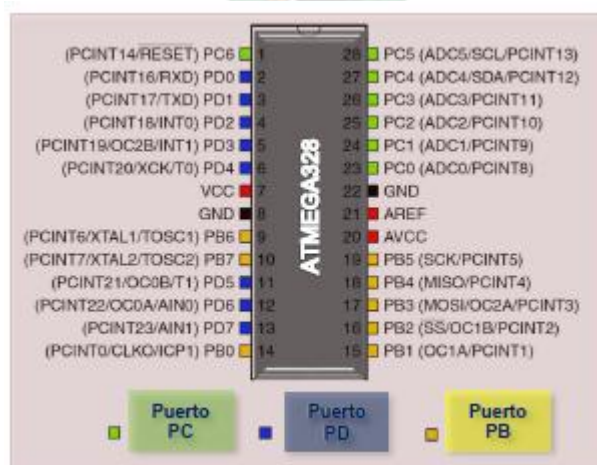


Figura 13. Puertos E/S del ATMEGA328

8.1.2.9.1. Parámetros fundamentales

Tabla 6. parámetros del microcontrolador ATMEGA 328

PARÁMETROS	VALORES
Flash	32 Kbytes
SRAM	2 Kbytes
Cantidad Pines	28
Frecuencia máxima de operación	20 MHz
CPU	8-bit AVR
Número de Canales variables	16
Pines máximos de E/S	26
Interrupciones internas	24

8.1.2.9.2. Aplicaciones

Hoy el ATmega328 se usa comúnmente en múltiples proyectos y sistemas autónomos donde un micro controlador simple, de bajo consumo, bajo costo es requerido. Tal vez la implementación más común de este chip es en la popular plataforma Arduino, en sus modelos Uno y Nano.

8.1.2.9.3. Recursos especiales de los microcontroladores

Cada fabricante oferta numerosas versiones de una arquitectura básica de microcontrolador. En algunas amplía las capacidades de las memorias, en otras incorpora nuevos recursos, en otras reduce las prestaciones al mínimo para aplicaciones muy simples, etc. La labor del diseñador es encontrar el modelo mínimo que satisfaga todos los requerimientos de su aplicación. De esta forma, minimizará el coste, el hardware y el software. Los principales recursos específicos que incorporan los microcontroladores son:

- Temporizadores o Timers.
- Perro guardián o Watchdog.
- Protección ante fallo de alimentación o Brownout.

- Estado de reposo o de bajo consumo (Sleep mode).
- Conversor A/D (Analógico ->Digital).
- Conversor D/A (Digital ->Analógico).
- Modulador de ancho de pulso o PWM
- Puertas de comunicación.

A continuación pasamos a ver con un poco más de detalle cada uno de ellos.

8.1.2.9.4. Temporizadores (Timers)

Se emplean para controlar periodos de tiempo (temporizadores) y para llevar la cuenta de acontecimientos que suceden en el exterior (contadores).

Para la medida de tiempos se carga un registro con el valor adecuado y a continuación dicho valor se va incrementando o decrementando al ritmo de los impulsos de reloj o algún múltiplo hasta que se desborde y llegue a 0, momento en el que se produce un aviso. Cuando se desean contar acontecimientos que se materializan por cambios de nivel o flancos en alguna de las patillas del microcontrolador, el mencionado registro se va incrementando o decrementando al ritmo de dichos impulsos.

8.1.2.9.5. Perro guardián (Watchdog)

En electrónica, un perro guardián (*watchdog*) es un mecanismo de seguridad que provoca un reset del sistema en caso de que éste se haya bloqueado.

Consiste en un temporizador que irá continuamente decrementando un contador, inicialmente con un valor alto. Cuando este contador llegue a cero, se reiniciará el sistema, así que se debe diseñar una subrutina en el programa de manera que refresque o reinicie al perro guardián antes de que provoque el reset.

Si el programa falla o se bloquea, al no actualizar el contador del perro guardián a su valor de inicio, éste llegará a decrementarse hasta cero y se reiniciará el sistema.

8.1.2.9.6. Protección ante fallos de alimentación (Brownout)

Se trata de un circuito que resetea al microcontrolador cuando el voltaje de alimentación (VDD) es inferior a un voltaje mínimo (brownout). Mientras el

voltaje de alimentación sea inferior al de brownout el dispositivo se mantiene reseteado, comenzando a funcionar normalmente cuando sobrepasa dicho valor. Esto es muy útil para evitar datos erróneos por transiciones y ruidos en la línea de alimentación.

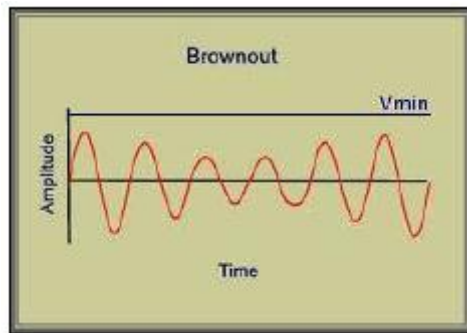


Figura 14. Reseteo del dispositivo por Brownout

8.1.2.9.7. Estado de reposo o de bajo consumo

Son abundantes las situaciones reales de trabajo en que el microcontrolador debe esperar, sin hacer nada, a que se produzca algún acontecimiento externo que le ponga de nuevo en funcionamiento. Para ahorrar energía, (factor clave en los aparatos portátiles), los microcontroladores disponen de una instrucción especial (SLEEP en los PIC), que les pasa al estado de reposo o de bajo consumo, en el cual los requerimientos de potencia son mínimos. En dicho estado se detiene el reloj principal y se congelan sus circuitos asociados, quedando sumido en un profundo sueño. Al activarse una interrupción ocasionada por el acontecimiento esperado, el microcontrolador se despierta y reanuda su trabajo. Para hacernos una idea, esta función es parecida a la opción de suspender en el menú para apagar el equipo (en aquellos PCs con administración avanzada de energía).

8.1.2.9.8. Conversor A/D (CAD)

Los microcontroladores que incorporan un conversor A/D (Analógico/Digital) pueden procesar abundantes señales analógicas provenientes de aplicaciones. Suelen disponer de un multiplexor que permite aplicar a la entrada del CAD diversas señales analógicas desde las patillas del circuito integrado.

8.1.2.9.9. Conversor D/A (CDA)

Transforma los datos digitales obtenidos del procesamiento del computador en su correspondiente señal analógica que saca al exterior por una de las patillas del chip. Existen aún muchos circuitos que siguen trabajando con señales analógicas.

8.1.2.9.10. Puertas de Comunicación

Con objeto de dotar al microcontrolador de la posibilidad de comunicarse con otros dispositivos externos, otros buses de microprocesadores, buses de sistemas, buses de redes y poder adaptarlos con otros elementos bajo otras normas y protocolos. Algunos modelos disponen de recursos que permiten directamente esta tarea, entre los que destacan:

- *UART*, adaptador de comunicación serie asíncrona.(Ej: Puerto Serie).
- *USART*, adaptador de comunicación serie síncrona y asíncrona.
- Puerta paralela esclava para poder conectarse con los buses de otros microprocesadores.
- *USB (Universal Serial Bus)*, que es un moderno bus serie para los PC.
- *Bus I2C*, que es un interfaz serie de dos hilos desarrollado por Philips.
- *CAN (Controller Area Network)*, para permitir la adaptación con redes, desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles.

8.1.2.9.11. Correspondencia entre los pines ATMEGAS y ARDUINO.

El Arduino Mini está basado en un formato físicamente más pequeño que incluye dos pines para el conversor Analógico-Digital que no están disponibles para las versiones de Arduino que implementan el chip en formato DIP.

A continuación se encuentra la correspondencia de pines del ATmega8, el chip original utilizados en los primeros Arduinos. La correspondencia de los pines para el ATmega168 y el 328, es aproximadamente la misma.

Arduino function				Arduino function
reset	(PCINT14/RESET) PC6	1	26	PC5 (ADC5/SCL/PCINT13) analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12) analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	28	PC3 (ADC3/PCINT11) analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	29	PC2 (ADC2/PCINT10) analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	30	PC1 (ADC1/PCINT9) analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	31	PC0 (ADC0/PCINT8) analog input 0
VCC	VCC	7	32	GND GND
GND	GND	8	33	AREF analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	34	AVCC VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	35	PB5 (SCK/PCINT5) digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	36	PB4 (MISO/PCINT4) digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	37	PB3 (MOSI/OC2A/PCINT3) digital pin 11(PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	38	PB2 (SS/OC1B/PCINT2) digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/CP1) PB0	14	39	PB1 (OC1A/PCINT1) digital pin 9 (PWM)

Figura 15. Correspondencia entre pines ATMEGA y ARDUINO

Nota: Los pines digitales 11, 12 y 13 son utilizados por el ISCP para las conexiones MISO, MOSI y SCK (ATmega168 pin 17, 18 y 19). Estos pines evitan cargas de baja impedancia cuando se utiliza ISCP.

8.1.2.10. ANDROID

8.1.2.10.1. Introducción

Android es un sistema operativo con una plataforma abierta para dispositivos móviles adquirido por Google y la Open Handset Alliance, su finalidad es satisfacer la necesidad de los operadores móviles y fabricantes de dispositivos, además de fomentar el desarrollo de aplicaciones, cualidad que ningún otro sistema operativo incluye en sus conceptos.

Es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, *tablets* o *tabletfonos*; y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, la compró. Android fue presentado en 2007 junto la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles.

El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008. Los dispositivos de Android venden más que las ventas combinadas de Windows Phone e IOS.

El éxito del sistema operativo se ha convertido en objeto de litigios sobre patentes en el marco de las llamadas «Guerras por patentes de teléfonos inteligentes» (en inglés, *Smartphone patent wars*) entre las empresas de tecnología. Según documentos secretos filtrados en 2013 y 2014, el sistema operativo es uno de los objetivos de las agencias de inteligencia internacionales.

La versión básica de Android es conocida como Android Open Source Project (AOSP).

El 25 de junio de 2014 en la Conferencia de Desarrolladores Google I/O, Google mostró una evolución de la marca Android, con el fin de unificar tanto el hardware como el software y ampliar mercados.

8.1.2.10.2. Historia

Fue desarrollado por Android Inc., empresa que en 2005 fue comprada por Google, aunque no fue hasta 2008 cuando se popularizó, gracias a la unión al proyecto de Open Handset Alliance⁹, un consorcio formado por 48 empresas de desarrollo hardware, software y telecomunicaciones, que decidieron promocionar el software libre. Pero ha sido Google quien ha publicado la mayor parte del código fuente del sistema operativo, gracias al software Apache, que es una fundación que da soporte a proyectos software de código abierto.

Dado que Android está basado en el núcleo de Linux, tiene acceso a sus recursos, pudiendo gestionarlo, gracias a que se encuentra en una capa por encima del Kernel¹⁰, accediendo así a recursos como los controladores de pantalla, cámara, parlantes, memoria flash, entre otros.

8.1.2.10.3. Arquitectura y Características

Para empezar con el desarrollo de aplicaciones en Android es importante conocer cómo está estructurado este sistema operativo. A esto le llamamos arquitectura y en el caso de Android está formada por varias capas que facilitan al desarrollador la creación de aplicaciones. Además, esta distribución permite

acceder a las capas más bajas mediante el uso de librerías para que así el desarrollador no tenga que programar a bajo nivel las funcionalidades necesarias para que una aplicación haga uso de los componentes de hardware de los teléfonos.

Cada una de las capas utiliza elementos de la capa inferior para realizar sus funciones, es por ello que a este tipo de arquitectura se le conoce también como *pila*.

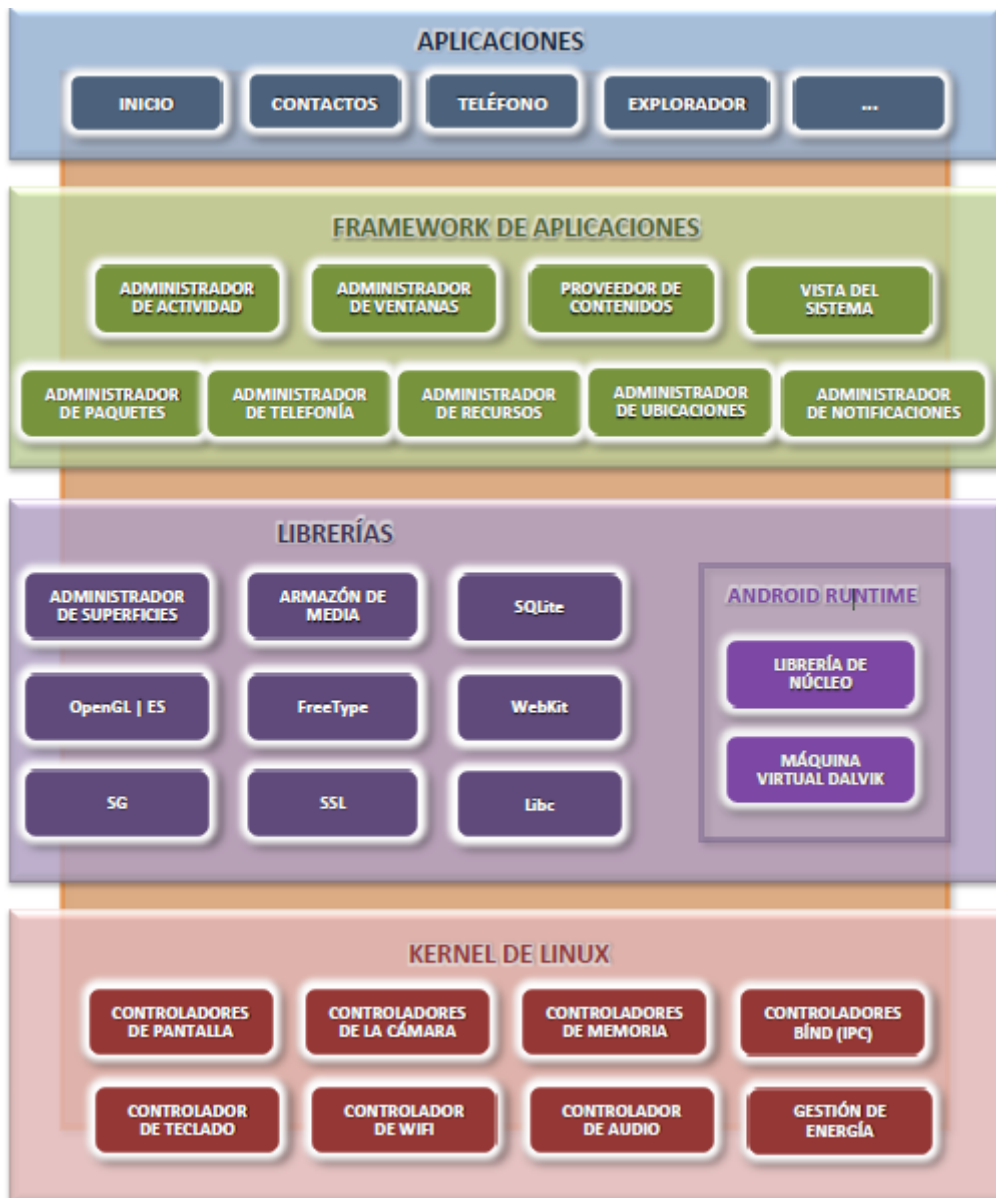


Figura 16. Arquitectura de un sistema Android

Los componentes principales del sistema operativo de Android (cada sección se describe en detalle):

- **Aplicaciones:** las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.
- **Marco de trabajo de aplicaciones:** los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.
- **Bibliotecas:** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.
- **Runtime de Android:** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecutaba hasta la versión 5.0 archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx". Desde la versión 5.0 utiliza el ART, que compila totalmente al momento de instalación de la aplicación.

- **Núcleo Linux:** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

8.1.2.10.4. Aplicaciones en ANDROID

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK), pero están disponibles otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C oC++, Google App Inventor, un entorno visual para programadores novatos y varios marcos de aplicaciones basadas en la web multiteléfono. También es posible usar las bibliotecas Qt gracias al proyecto *Necesitas SDK*.

El desarrollo de aplicaciones para Android no requiere aprender lenguajes complejos de programación. Todo lo que se necesita es un conocimiento aceptable de Java y estar en posesión del kit de desarrollo de software o «SDK» provisto por Google el cual se puede descargar gratuitamente.

Todas las aplicaciones están comprimidas en formato APK, que se pueden instalar sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos.

8.1.2.10.5. Librerías ANDROID

Android ofrece una serie de interfaces de programación de aplicaciones o **APIs** para el desarrollo de sus aplicaciones.

La siguiente lista proporciona una visión de lo que está disponible. Todos los dispositivos Android ofrecen al menos estas APIs:

- **android.util** El paquete básico de servicios públicos contiene las clases de bajo nivel, como contenedores especializados, formateadores de cadenas, y de análisis XML de servicios públicos.
- **android.os** El paquete de sistema operativo permite el acceso a los servicios básicos como el paso de mensajes, la comunicación entre procesos y funciones de reloj.

- **android.graphics** La API de gráficos, es el suministro de las clases de bajo nivel como lienzos de apoyo, colores y las primitivas de dibujo. También le permite dibujar sobre lienzos.
- **android.text** Las herramientas de procesamiento de texto para mostrarlo y analizarlo.
- **android.database** Proporciona las clases de bajo nivel necesario para la manipulación de cursores cuando se trabaja con bases de datos.
- **android.content** El contenido de la API se utiliza para admirar el acceso a los datos y a la publicación, proporcionando los servicios para hacer frente a los recursos, los proveedores de contenido y los paquetes.
- **android.view** Las vistas son un núcleo de la interfaz de usuario. Todos los elementos de la interfaz se construyen utilizando una serie de vistas que proporcionan los componentes de interacción con el usuario.
- **android.widget** Construido sobre el paquete de vista, están las clases widget y los elementos de la interfaz de usuario para su uso en las aplicaciones. Se incluyen listas, botones y diseños.
- **com.google.android.maps** API de alto nivel que proporciona acceso a los controles de mapas que usted puede utilizar en su aplicación. Incluye el control MapView así como la superposición y la clase Map Controller utilizados para anotar y controlar dichos mapas.
- **android.app** Paquete de alto nivel que proporciona el acceso al modelo de solicitud. Este, incluye la actividad de servicios y las API que forman la base de todas sus aplicaciones.
- **android.provider** Para facilitar el acceso a los desarrolladores a determinados proveedores de contenidos estándar, el paquete proveedor ofrece clases para todas sus distribuciones.
- **android.telephony** Las APIs de telefonía le dan la posibilidad de interactuar directamente con el dispositivo de Teléfono, permitiéndole realizar, recibir y controlar las llamadas de teléfono, su estado y mensajes SMS.
- **android.webkit** Ofrece funciones para trabajar con contenido basado en web, incluyendo un control WebView para incrustar los navegadores en sus actividades y un administrador de cookies.

8.1.2.10.6. Librerías Avanzadas

Las bibliotecas ofrecen toda la funcionalidad que necesita para comenzar a crear aplicaciones en Android, pero no pasará mucho tiempo antes de que esté dispuesto a profundizar en APIs avanzadas que ofrecen una funcionalidad más emocionante.

Android pretende abarcar una gran variedad de equipos móviles, de modo que tenga en cuenta que la adecuación y aplicación de las siguientes APIs variará en función del dispositivo a que se apliquen:

- **android.location** Da las aplicaciones de acceso a la ubicación física del dispositivo actual. Los servicios de localización ofrecen acceso genérico a información de localización utilizando cualquier posición de hardware xing- Fi o tecnología disponible en el dispositivo.
- **android.media** Las APIs de los medios de comunicación proporcionan soporte para reproducción y grabación de audio.
- **android.opengl** Android ofrece un potente motor 3D que utiliza la API de OpenGL ES que usted puede utilizar para crear interfaces de usuario en 3D dinámico para sus aplicaciones.
- **android.hardware** Cuando sea posible, el hardware de la API expone un sensor incluyendo la cámara, acelerómetro, sensores y una brújula.
- **android.bluetooth, android.net.wifi, android and telephony** Provee el acceso a la plataforma hardware, incluyendo Bluetooth, Wi-Fi y el hardware de telefonía

8.1.2.10.7. Multimedia

Android ofrece una biblioteca multimedia capaz de reproducir y grabar una amplia gama de formatos de medios de comunicación, proporcionando una amplia funcionalidad para multimedia, incluyendo grabación y reproducción de audio y vídeo.

También Android tiene diversas librerías para aplicaciones multimedia, el documento presentado se centrara en SensorManager, la cual sirve para manejar el sensor de orientación, y el acelerómetro. La razón de explicar esta API en concreto es porque el uso de cualquier otra librería de multimedia es muy parecido a la librería SensorManager.

La librería `SensorManager` se utiliza para administrar el hardware y los sensores disponibles en un dispositivo de Android. Para obtener una referencia al Servicio de sensor tenemos que usar `getSystemService` como se muestra a continuación:

```
String service_name = Context.SENSOR_SERVICE;

SensorManager sensorManager =
(SensorManager) getSystemService(service_name);
```

Ahora se explicará cómo utilizar el Administrador de sensores para controlar la orientación y la aceleración. La forma que se muestra aquí se puede utilizar para controlar cualquier sensor de hardware disponible:

```
SensorListener mySensorListener = new SensorListener() {
public void onSensorChanged(int sensor, float[] values) {
// Cambios del Valor del Sensor
}
public void onAccuracyChanged(int sensor, int accuracy) {
// Método auxiliar
}
};
```

La interfaz de `SensorListener` se utiliza para obtener el valor del sensor y los cambios de precisión. El método o evento `onSensorChanged` reacciona a los cambios de valor. El parámetro `sensor` es el identificador del sensor que activa el evento, mientras que la matriz contiene los nuevos valores detectados por ese sensor.

El evento `onAccuracyChanged` se produce cuando hay cambios en la precisión de un sensor. El parámetro `sensor`, de nuevo es el identificador del sensor que

activa el evento, mientras que el parámetro accuracy indica la nueva precisión del sensor usando una de las siguientes constantes:

- **SensorManager.SENSOR_STATUS_ACCURACY_HIGH:** Indica que el sensor tiene la precisión más alta posible.
- **SensorManager.SENSOR_STATUS_ACCURACY_LOW:** Indica que el sensor tiene la precisión más baja posible y necesita ser calibrado.
- **SensorManager.SENSOR_STATUS_ACCURACY_MEDIUM:** Indica que los datos del sensor tienen precisión media, y que puede mejorar la calibración de las lecturas.
- **SensorManager.SENSOR_STATUS_UNRELIABLE:** Indica que los datos del sensor no son fiables, lo que significa que se requiere calibración o lecturas que no son posibles en ese momento.

El Administrador de sensores incluye constantes para ayudar a identificar el sensor que disparó el evento de cambio. La siguiente lista incluye los sensores para los que las constantes están definidas. Algunos o todos de estos sensores estarán disponibles para las aplicaciones en función del hardware disponible en el dispositivo:

- **SensorManager.SENSOR_ACCELEROMETER:** Se trata de un sensor *acelerómetro* que devuelve la corriente de aceleración a lo largo de tres ejes en (m/s²).
- **SensorManager.SENSOR_ORIENTATION:** Es un sensor de orientación que devuelve la orientación actual en tres ejes y se mide en grados.
- **SensorManager.SENSOR_LIGHT:** Es un sensor de *luz* ambiental que devuelve un valor único para describir la iluminación ambiental.
- **SensorManager.SENSOR_MAGNETIC_FIELD:** Sirve para determinar la *corriente magnética* medida en microteslas (μT) a lo largo de tres ejes.
- **SensorManager.SENSOR_PROXIMITY:** Es un sensor de *proximidad* que devuelve un valor único y describe la distancia entre el dispositivo y el objeto de destino en metros (m).

- **SensorManager.SENSOR_TEMPERATURE:** Es un sensor de *temperatura* que devuelve la temperatura ambiente en grados Celsius (°C).

Para recibir notificaciones de los cambios en un sensor en particular, hay que crear un `SensorListener`, y registrarlo con el `SensorManager` especificando el sensor y la velocidad a la que el sensor se debe actualizar, como se muestra a continuación:

```
sensorManager.registerListener(mySensorListener,  
SensorManager.SENSOR_TRICORDER,  
SensorManager.SENSOR_DELAY_FASTEST);
```

El `SensorManager` incluye las siguientes constantes, las cuales permiten seleccionar una frecuencia de actualización adecuada:

- **SensorManager.SENSOR_DELAY_FASTEST:** Especifica la frecuencia más rápida posible de actualización del sensor.
- **SensorManager.SENSOR_DELAY_GAME:** Selecciona una tasa de actualización adecuada para su uso en el control de juegos.
- **SensorManager.SENSOR_DELAY_NORMAL:** Especifica la tasa de actualización por defecto.
- **SensorManager.SENSOR_DELAY_UI:** Especifica un índice adecuado para la actualización de las características de interfaz de usuario.

La tasa que se seleccione no es tan relevante ya que el `sensorManager` puede devolver resultados más rápido o más lento de lo que usted especifique, aunque tiende a ser más rápido. Para minimizar el coste de los recursos asociados al utilizar el sensor en su aplicación, usted debe tratar de seleccionar la velocidad mínima adecuada.

8.1.2.11. Plataforma App Inventor

Para nuestro diseño de casa Domótica, la cual controlaremos mediante un teléfono celular con sistema android utilizaremos una plataforma de Google Labs que es App Inventor para realizar nuestro software.



Figura 17. Logo de App Inventor

8.1.2.11.1. Introducción

Google App Inventor es una plataforma de Google Labs para crear aplicaciones de software para el sistema operativo Android. De forma visual y a partir de un conjunto de herramientas básicas, el usuario puede ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web. Las aplicaciones fruto de App Inventor están limitadas por su simplicidad, aunque permiten cubrir un gran número de necesidades básicas en un dispositivo móvil.

Con Google App Inventor, se espera un incremento importante en el número de aplicaciones para Android debido a dos grandes factores: la simplicidad de uso, que facilitará la aparición de un gran número de nuevas aplicaciones; y Google Play, el centro de distribución de aplicaciones para Android donde cualquier usuario puede distribuir sus creaciones libremente.

8.1.2.11.2. Historia

La plataforma se puso a disposición del público el 12 de julio de 2010 y está dirigida a personas que no están familiarizadas con la programación informática. En la creación de App Inventor, Google se basó en investigaciones previas significativas en informática educativa.

8.1.2.11.3. Características

El editor de bloques de la plataforma App Inventor, utiliza la librería Open Blocks de Java para crear un lenguaje visual a partir de bloques. Estas librerías están distribuidas por Massachusetts Institute of Technology (MIT) bajo su licencia libre (MIT License). El compilador que traduce el lenguaje visual de los bloques para la aplicación en Android utiliza Kawa como lenguaje de programación, distribuido como parte del sistema operativo GNU de la Free Software Foundation

App Inventor pueden tener su primera aplicación en funcionamiento en una hora o menos, y se pueden programar aplicaciones más complejas en mucho menos tiempo que con los lenguajes más tradicionales, basados en texto. Inicialmente desarrollado por el profesor Hal Abelson y un equipo de Google Educación, mientras que Hal pasaba un año sabático en Google, App Inventor se ejecuta como un servicio Web administrado por personal del Centro del MIT para el aprendizaje móvil –una colaboración de MIT de Ciencia Computacional e Inteligencia Artificial de laboratorio (CSAIL) y el Laboratorio de Medios del MIT–. Inventor MIT App es compatible con una comunidad mundial de casi dos millones de usuarios que representan a 195 países en todo el mundo. Más de 85 mil usuarios semanales activos de la herramienta han construido más de 4,7 millones de aplicaciones de Android. Una herramienta de código abierto que pretende realizar la programación y la creación de aplicaciones accesibles a una amplia gama de audiencias.

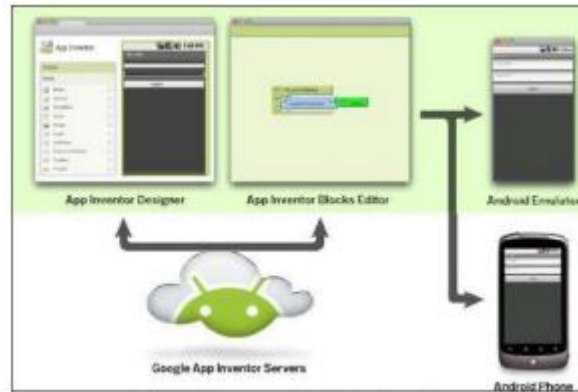


Figura 18. Desarrollo con App Inventor

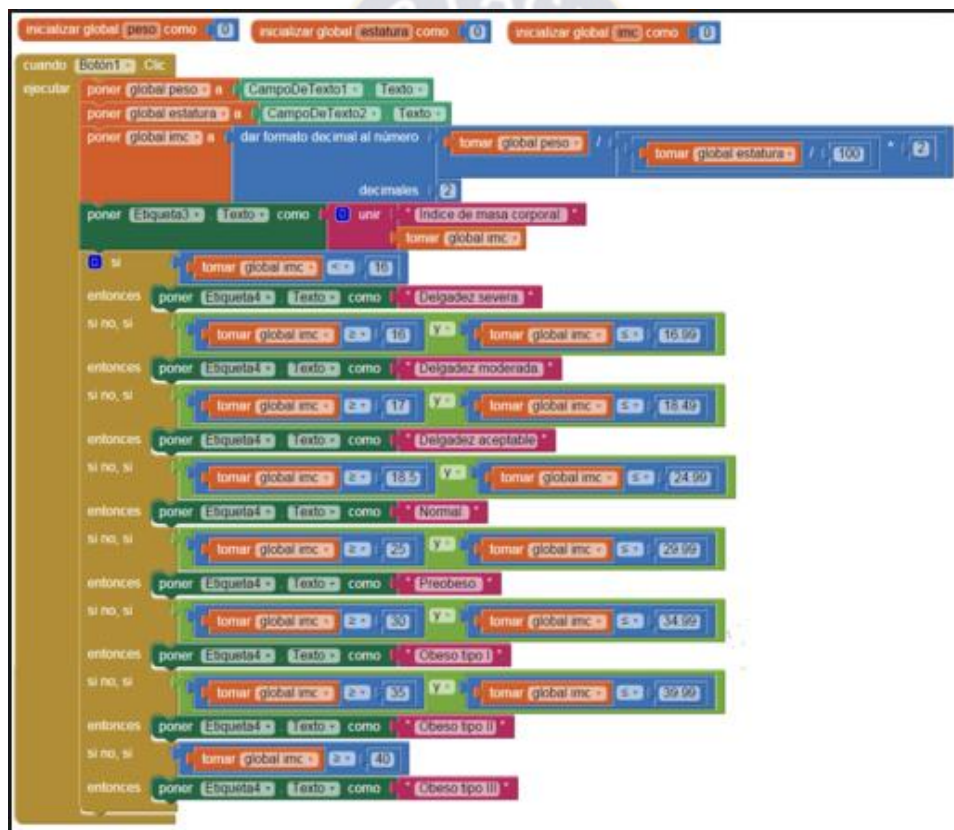


Figura 19. Ejemplo con App Inventor

8.1.2.12. BLUETOOTH

8.1.2.12.1. Introducción

Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los

2.4 GHz. Los principales objetivos que se pretenden conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles.
- Eliminar los cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

Los dispositivos que con mayor frecuencia utilizan esta tecnología pertenecen a sectores de las telecomunicaciones y la informática personal, como PDA, teléfonos móviles, computadoras portátiles, ordenadores personales, impresoras o cámaras digitales.

Se denomina Bluetooth al protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, que requieren corto alcance de emisión y basados en transceptores de bajo costo.

Los dispositivos que incorporan este protocolo pueden comunicarse entre sí cuando se encuentran dentro de su alcance. Las comunicaciones se realizan por radiofrecuencia de forma que los dispositivos no tienen que estar alineados y pueden incluso estar en habitaciones separadas si la potencia de transmisión es suficiente. Estos dispositivos se clasifican como "Clase 1", "Clase 2" o "Clase 3" en referencia a su potencia de transmisión, siendo totalmente compatibles los dispositivos de una caja de ordenador

Tabla 7. Clasificación por Potencia

Clase	Potencia máxima permitida (mW)	Potencia máxima permitida (dBm)	Alcance (aproximado)
Clase 1	100 mW	20 dBm	~100 metros
Clase 2	2.5 mW	4 dBm	~5-10 metros
Clase 3	1 mW	0 dBm	~1 metro

En la mayoría de los casos, la cobertura efectiva de un dispositivo de clase 2 se extiende cuando se conecta a un transceptor de clase 1. Esto es así gracias a la mayor sensibilidad y potencia de transmisión del dispositivo de clase 1, es decir, la mayor potencia de transmisión del dispositivo de clase 1 permite que la señal llegue con energía suficiente hasta el de clase 2. Por otra parte la mayor

sensibilidad del dispositivo de clase 1 permite recibir la señal del otro pese a ser más débil.

Los dispositivos con Bluetooth también pueden clasificarse según su capacidad de canal:

Tabla 8. Versiones BLUETOOTH

Versión	Ancho de banda
Versión 1.2	1 Mbit/s
Versión 2.0 + EDR	3 Mbit/s
Versión 3.0 + HS	24 Mbit/s
Versión 4.0	32 Mbit/s

8.1.2.12.2. Aplicaciones

- Conexión sin cables vía OBEX.
- Transferencia de fichas de contactos, citas y recordatorios entre dispositivos vía OBEX.
- Reemplazo de la tradicional comunicación por cable entre equipos GPS y equipamiento médico.
- Controles remotos (tradicionalmente dominado por el infrarrojo).
- Enviar pequeñas publicidades desde anunciantes a dispositivos con Bluetooth. Un negocio podría enviar publicidad a teléfonos móviles cuyo Bluetooth (los que lo posean) estuviera activado al pasar cerca.
- Las consolas [Sony](#) PlayStation 3, PlayStation 4, Microsoft Xbox 360 y Xbox One incorporan Bluetooth, lo que les permite utilizar mandos inalámbricos, aunque los mandos originales de la Wii funcionan mezclando la tecnología de infrarrojos y Bluetooth.
- Enlace inalámbrico entre sistemas de audio y los altavoces (o altoparlantes) correspondientes

8.1.2.12.3. Arquitectura Hardware

El hardware que compone el dispositivo Bluetooth está compuesto por dos partes:

- **un dispositivo de radio**, encargado de modular y transmitir la señal.
- **un controlador digital**, compuesto por una CPU, un procesador de señales digitales (DSP - Digital Signal Processor) llamado Link Controller (o controlador de Enlace) y de las interfaces con el dispositivo anfitrión.

El LC o Link Controller se encarga del procesamiento de la banda base y del manejo de los protocolos ARQ y FEC de la capa física; además, se encarga de las funciones de transferencia tanto asíncrona como síncrona, la codificación de audio y el cifrado de datos.

La CPU del dispositivo se encarga de las instrucciones relacionadas con Bluetooth en el dispositivo anfitrión, para así simplificar su operación. Para ello, sobre la CPU corre un software denominado Link Manager cuya función es la de comunicarse con otros dispositivos por medio del protocolo LMP.

DIPOSITIVO DE RADIO BLUETOOTH GENERIC

Entre las tareas realizadas por el LC y el Link Manager, destacan las siguientes:

- Envío y Recepción de Datos.
- Paginación y Peticiones.
- Establecimiento de conexiones.
- Autenticación.
- Negociación y establecimiento de tipos de enlace.
- Establecimiento del tipo de cuerpo de cada paquete.
- Establecer el dispositivo en modo sniff o hold: El primero, sniff, significa olfatear, pero en castellano y en informática se traduce por escuchar (el medio): en este caso es la frecuencia o frecuencias en la que está funcionando el dispositivo. Así, cualquier paquete de datos enviado en esa frecuencia será "leído" por el dispositivo, aunque no vaya dirigido a

él. Leerá todos los datos que se envíen en esa frecuencia por cualquier otro dispositivo Bluetooth, es lo que se denomina rastreo de paquetes. Una técnica parecida pero a nivel de frecuencias es la que se utiliza para detectar redes wi-fi, generalmente para encontrar redes abiertas (sin contraseña), al escanear todas las frecuencias se obtiene información de cada frecuencia o canal de las redes wi-fi disponibles. Hold por su parte significa mantener, retener; esto quiere decir que el dispositivo se mantendrá en esa frecuencia aunque no emita ni reciba nada, manteniendo esa frecuencia siempre disponible aunque otros dispositivos la utilicen.

8.1.2.12.4. Usos de BLUETOOTH

Bluetooth se utiliza principalmente en un gran número de productos tales como teléfonos, impresoras, módems y auriculares. Su uso es adecuado cuando puede haber dos o más dispositivos en un área reducida sin grandes necesidades de ancho de banda. Su uso más común está integrado en teléfonos y PDA, bien por medio de unos auriculares Bluetooth o en transferencia de ficheros. además se puede realizar y confeccionar enlaces o vincular distintos dispositivos entre sí.

Bluetooth simplifica el descubrimiento y configuración de los dispositivos, ya que estos pueden indicar a otros los servicios que ofrecen, lo que permite establecer la conexión de forma rápida (solo la conexión, no la velocidad de transmisión).

8.1.2.12.5. Módulo BLUETOOTH H 05

El módulo de bluetooth HC-05 es el que ofrece una mejor relación de precio y características, ya que es un módulo Maestro-Esclavo, quiere decir que además de recibir conexiones desde una PC o tablet, también es capaz de generar conexiones hacia otros dispositivos bluetooth. Esto nos permite por ejemplo, conectar dos módulos de bluetooth y formar una conexión punto a punto para transmitir datos entre dos microcontroladores o dispositivos.

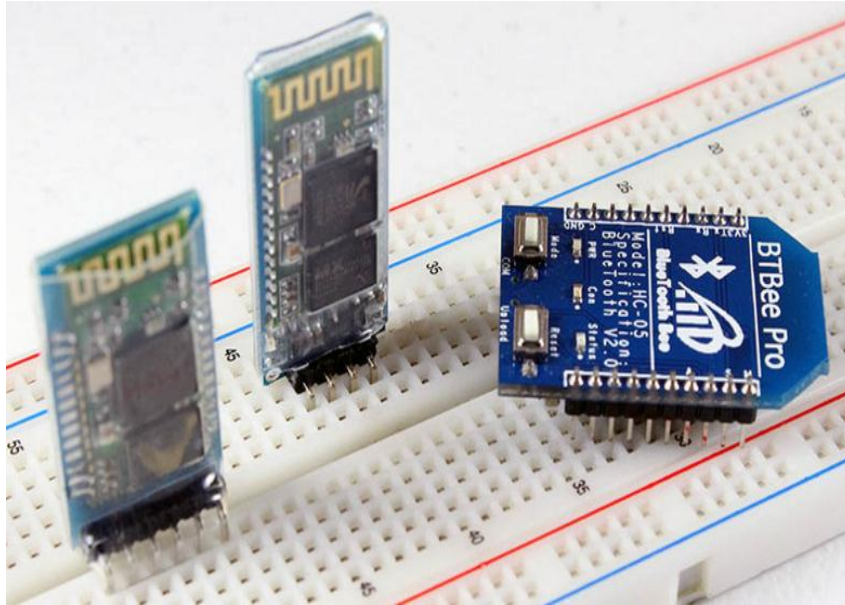


Figura 20. Módulo BLUETOOTH H 05

8.1.2.12.6. Características

- Chipset CSR BC417143
- Bluetooth versión V2.0+EDR
- Tensión de alimentación: 3.3V
- Frecuencia: 2.4GHz banda ISM
- Modulación: GFSK (Gaussian Frequency Shift Keying)
- Seguridad: Autenticación y encriptación.
- Velocidad-> Asíncrono: 2.1Mbps (Max) / 160 kbps ; Síncrono: 1Mbps/1Mbps
- Soporta comandos AT para configuración a través de un puerto serie.
- Configuración por defecto para el puerto COM: 9600, N, 8,1
- Temperatura de trabajo: -20 °C a +75 °C
- Dimensiones: 26.9mm x 13mm x 2.2 mm

El HC-05 tiene un modo de comandos AT que debe activarse mediante un estado alto en el PIN34 mientras se enciende (o se resetea) el módulo. En las versiones para protoboard este pin viene marcado como “Key”. Una vez que estamos en el modo de comandos AT, podemos configurar el módulo bluetooth y cambiar parámetros como el nombre del dispositivo, password, modo maestro/esclavo, etc.

Para comunicarnos con el módulo y configurarlo, es necesario tener acceso al módulo mediante una interfaz serial. Podemos usar un arduino con un par de cables (aprovechando el puente USB-Serial del Arduino), un kit para XBee o un simple MAX3232 en el puerto serie de la PC. Para este artículo estaremos usando un módulo Bluetooth Bee Pro que comercializamos en nuestra tienda web, así como una interfaz USB-Serial con socket Xbee “Foca” que también vendemos. Este es el hardware que utilizamos en nuestro taller, pero también es posible hacer los ensayos con el módulo HC-05 suelto o en formato para insertar en el protoboard sin mayor problema.

El puerto serie en modo de configuración para el HC-05 debe configurarse de la siguiente manera: 34800 bps, 8 bits de datos, Sin paridad, Sin control de flujo. Para entrar al modo de comandos AT seguimos los siguientes pasos:

1. Poner a estado alto en el pin 34 (PIO11)
2. Conectar la alimentación del módulo (o resetearlo de preferencia)
3. Enviar un comando AT\r\n para comprobar que estemos en modo de comando AT.

La siguiente lista es una compilación de los comandos que consideramos importantes

- **AT\r\n** Comando de prueba, debe responder con OK\r\n
- **AT+ROLE=1\r\n** Comando para colocar el módulo en modo Maestro (Master)
- **AT+ROLE=0\r\n** Comando para colocar el módulo en modo Esclavo (Slave)
- **AT+VERSION?\r\n** Obtener la versión del firmware

- **AT+UART=115200,1,2** Configurar el modo de funcionamiento del puerto serie en “modo puente”
- **AT+PIO=10,1** Colocar el pin de IO de propósito general a nivel alto

8.2. DESARROLLO PRACTICO/EXPERIMENTAL

8.2.1. PROGRAMACION EN ARDUINO

Como vimos anteriormente Arduino nos presente un entorno sumamente amigable al momento de realizar la programación de módulos electrónicos externos. Este proyecto de aplicación de Domótica debe ser muy práctica y sencilla en su utilización ya que será automatizada y en el caso de requerirlo operarla mediante un teléfono móvil con la sencillez que solo con un toque de pantalla un niño pueda ser capaz de operar.

8.2.1.1. Sensor PIR

El sensor PIR “Passive Infra Red” es un dispositivo piroeléctrico¹⁴ que mide cambios en los niveles de radiación infrarroja emitida por los objetos a su alrededor a una distancia máxima de 7 metros. Como respuesta al movimiento, el sensor cambia de nivel lógico de un “pin”, por lo cual, su uso es extremadamente simple. Adicionalmente es un sensor de bajo costo y reducido tamaño muy utilizado en sistemas de alarmas, iluminación controlada por movimiento y aplicaciones robóticas.



Figura 21. Sensor de movimiento PIR

8.2.1.1.1. Características Técnicas

- Voltaje de alimentación = 4.5 – 20 VDC.
- Rango de medición = 5 – 7 mt.
- Angulo de operación = 100°
- Simple calibración.

El sensor PIR cuenta solamente con tres terminales. Dos de ellos se lo utiliza para la alimentación y el restante es la salida de detección de movimiento.

La conexión al microcontrolador requiere del uso de este solo terminal. La figura muestra cómo se conecta el sensor PIR a la placa Arduino.

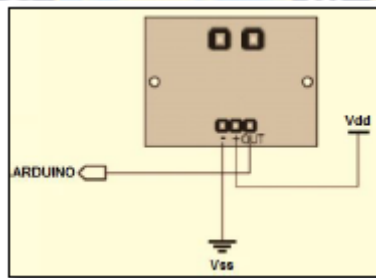


Figura 22. Conexión PIR – ARDUINO

Al sensor PIR también le acompañan dos resistencias variables de calibración Ch1 y RL2:

- Ch1: Con esta resistencia podemos establecer el tiempo que se va a mantener activa la salida del sensor. Una de las principales limitaciones de este módulo es que el tiempo mínimo que se puede establecer es de más o menos 3s. Si cambiamos la resistencia por otra de 100K, podemos bajar el tiempo mínimo a más o menos 0,5 s.
- RL2: Esta resistencia variable nos permite establecer la distancia de detección que puede variar entre 3-7m.



Figura 23. Resistencias variables de calibración

8.2.1.1.2. Forma de Operación

Los dispositivos piroelectricos, como el PIR, poseen elementos fabricados de un material cristalino que genera una carga eléctrica cuando se expone a la radiación infrarroja. Los cambios en la cantidad de radiación producen cambios de voltaje los cuales son medidos por un amplificador. El PIR contiene unos filtros especiales llamados lentes de Fresnel que enfocan las señales infrarrojas sobre el elemento sensor. Cuando las señales infrarrojas del ambiente donde se encuentra el sensor cambia rápidamente, el amplificador activa la salida para indicar movimiento. Esta salida permanece activa durante algunos segundos permitiendo al microprocesador saber si hubo algún movimiento.

8.2.1.2. Control de dirección de motores con C.I. L293 (PUENTE H)

El circuito integrado *L293D* incluye cuatro circuitos para manejar cargas de potencia media, en especial motores medianos, con la capacidad de controlar corriente hasta 600 mA en cada circuito y una tensión entre 4,5 V a 36 V.

Los circuitos individuales se pueden usar de manera independiente para controlar cargas de todo tipo y, en el caso de ser motores, manejar un único sentido de giro. Pero además, cualquiera de estos cuatro circuitos sirve para configurar la mitad de un *punte H*.

El integrado permite formar, entonces, dos puentes H completos, con los que se puede realizar el manejo de dos motores. En este caso el manejo

será bidireccional, con frenado rápido y con posibilidad de implementar fácilmente el control de velocidad.



Figura 24. Circuito integrado L293 (PUENTE H)

Las salidas tienen un diseño que permite el manejo directo de cargas inductivas tales como relés, solenoides, motores de corriente continua y motores por pasos, ya que incorpora internamente los diodos de protección de contracorriente para cargas inductivas.

Las entradas son compatibles con niveles de lógica TTL. Para lograr esto, incluso cuando se manejen motores de voltajes no compatibles con los niveles TTL, el chip tiene pines de alimentación separadas para la lógica (V_{CC2} , que debe ser de 5V) y para la alimentación de la carga (V_{CC1} , que puede ser entre 4,5V y 36V).

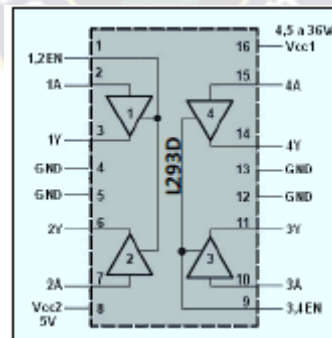


Figura 25. Diagrama de PINES L293

8.2.1.2.1. Características técnicas

- Total de pines: 16
- Corriente de salida: 600mA (por canal)
- Canales: 2
- Marca: ST

- Control de motores: Pasos y DC
- Diodos internos: Si
- Encapsulado: DIP

8.2.1.2.2. Tabla de verdad para el funcionamiento de los motores

Como se pudo advertir, el integrado L293D nos da la posibilidad de poder manipular o controlar el giro de cada uno de los motores de corriente continua que se utiliza en el proyecto. A continuación se detalla la tabla de verdad acerca del funcionamiento que tiene cada uno de los motores.

	MOTOR 1	
	GIRO DERECHA	GIRO IZQUIERDA
ADELANTE	1	0
ATRÁS	0	1

Figura 26. Tabla de verdad del motor

8.2.2. Programación de la aplicación android en App Inventor

Para que el móvil teleoperado llegue a ser operado o controlado por un operador, previamente se debe contar con una aplicación Android ya instalada en el equipo Smartphone; esta aplicación nos servirá como un interfaz de comunicación entre el usuario y el móvil, y mediante esta tendremos la facultad de operar al sistema robótico en su totalidad.

Como se mencionó anteriormente, Android es el sistema operativo de Google para dispositivos móviles, y para desarrollar aplicaciones en este sistema, tendremos que tener un conocimiento básico de Java, ya que este es el lenguaje que utiliza Android en sus aplicaciones.

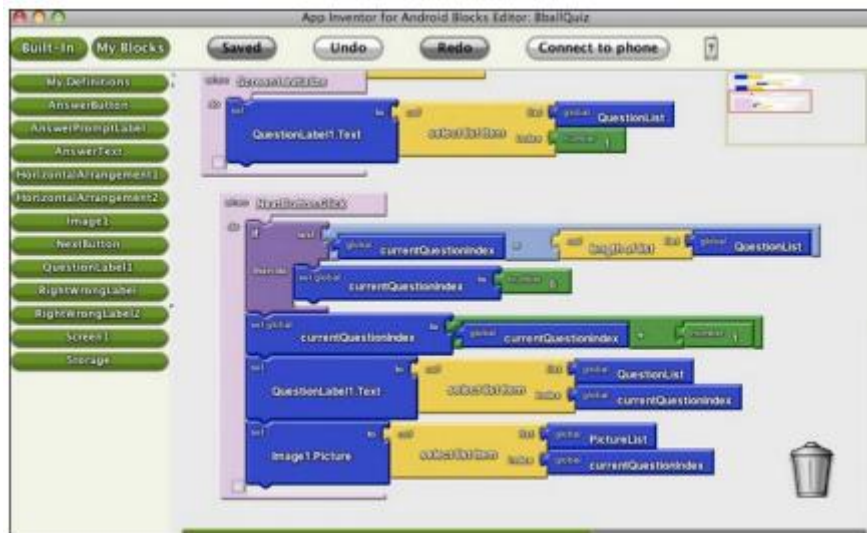


Figura 27. Editor en bloques en App inventor

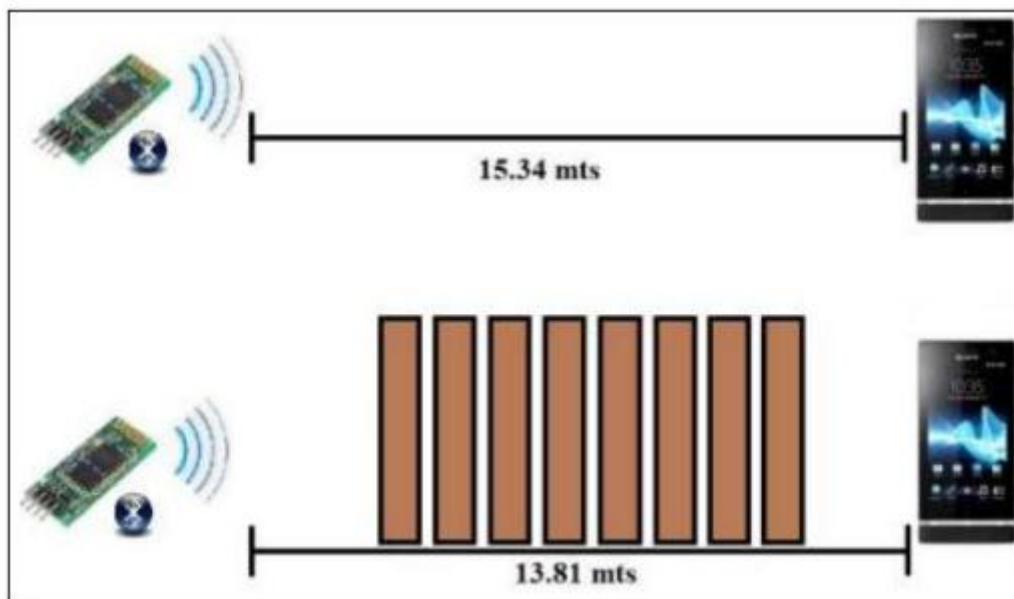


Figura 28. Alcance de señal libre y con barreras

App inventor también tiene un emulador que realiza una conexión con un teléfono virtual para simular las funciones que se programaron en la aplicación, esta función es muy útil ya que permite poner a prueba de funcionamiento del sistema creado, una vez descargada en el teléfono se instala como cualquier otra aplicación para android.



Figura 29. Simulador de App Inventor

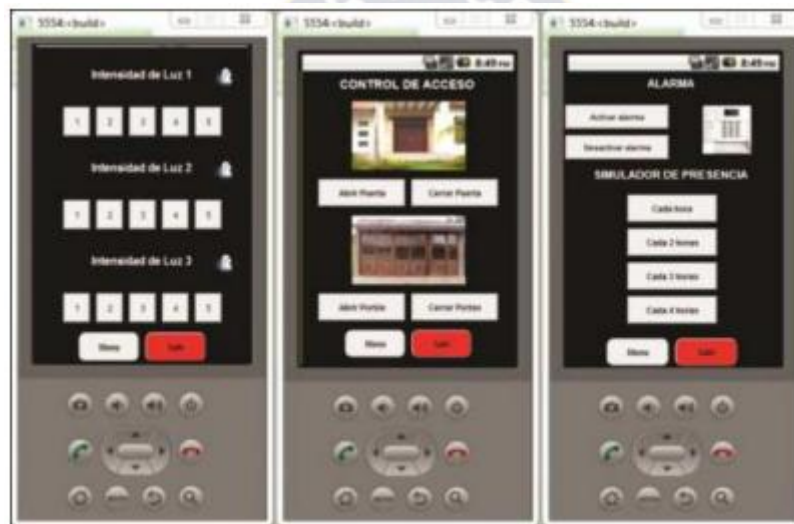


Figura 30. Simulador del proyecto

8.2.3. Circuito en Proteus

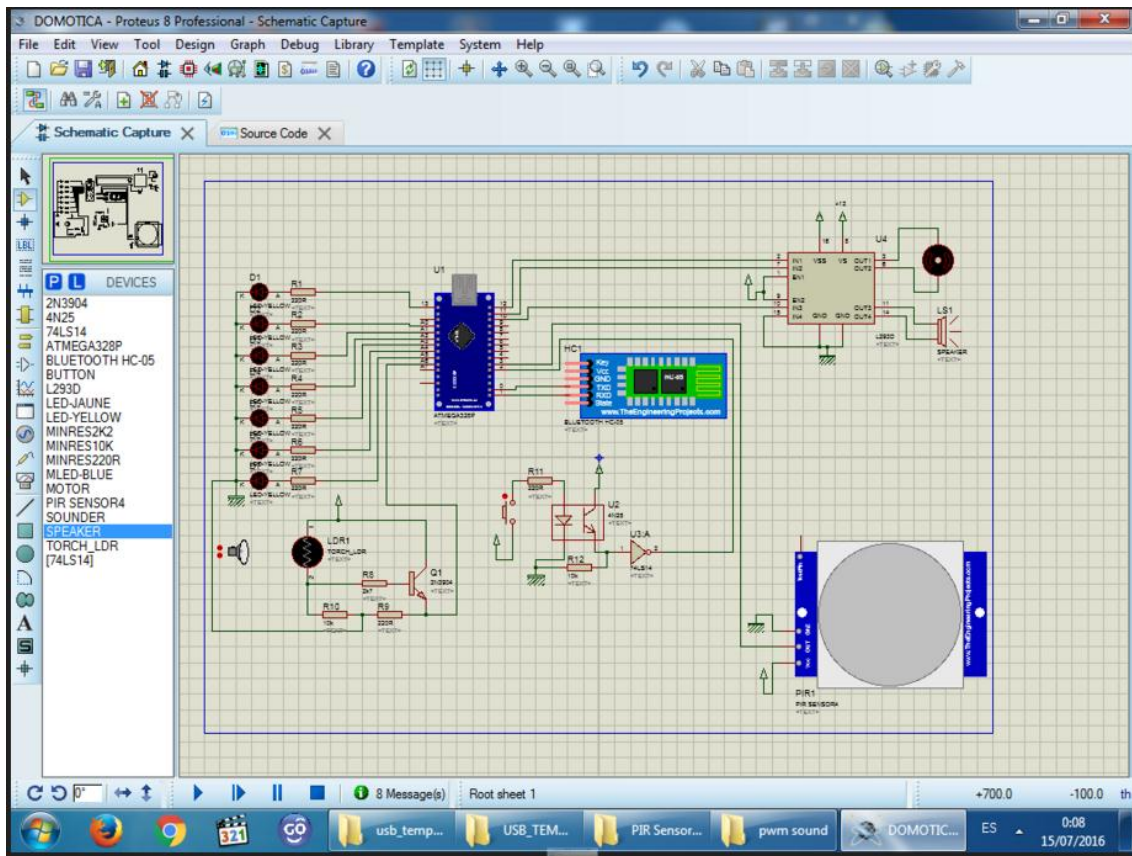


Figura 31. Circuito en Proteus

9. ANÁLISIS DE COSTOS

9.1. COSTOS FIJOS

El proyecto de grado expuesto cuenta con los siguientes costos fijos; vale la pena hacer notar que los costos de estos componentes **no tienen un efecto directo en cuanto al costo final del proyecto** debido a que se presume que el usuario final (operador del móvil teleoperado), ya cuenta con estos dispositivos electrónicos

Tabla 9. Costos fijos del proyecto

DESCRIPCION	COSTO Bs.
SMARTPHONE	1300
TOTAL	1300

9.2. COSTOS VARIABLES

Debido que los componentes electrónicos sufren constantes variaciones en su precio final; el costo total del proyecto presentado puede llegar a tener ligeros cambios con el pasar de los días. A continuación se detalla el costo final de cada uno de los componentes **al momento de ser adquiridos** tanto de proveedores locales como de proveedores en el interior del país.

Tabla 10. Costos variables del proyecto

DESCRIPCIÓN	COSTO Bs.
Arduino NANO	120
Sensor de Movimiento PIR	50
Motores DC	20
Fuente de alimentacion	40
Componentes Electronicos	10
Relay Spin	8
Modulo Bluetooth HC 05	130
leds	4
TOTAL	382

10. CONCLUSIONES Y RECOMENDACIONES

Los avances en la tecnología electrónica, de control y comunicaciones, nos van proporcionando cada vez más elementos disponibles y accesibles para desarrollar aplicaciones que puedan beneficiar a más partes de la sociedad y generar conocimiento para estudiantes y docentes. La integración de distintos elementos en hardware y software de la aplicación mostrada se puede extender a más elementos para generar un sistema de más capacidad que proporciona las distintas ventajas que se pretenden con la domótica. En los resultados obtenidos se demostró que para cierta clase de aplicaciones domóticas de bajos recursos se tiene un sistema suficientemente estable y confiable. Se describió una aplicación que puede potencializarse pero representa una alternativa económica. Se puede mejorar significativamente al tener elementos de mayor calidad y prestaciones, como en éste caso el alcance de la señal de comunicación y la seguridad en el sistema de control

que son las áreas de oportunidad a mejorar en trabajos futuros en este mismo tema.

El proyecto fue muy enriquecedor en cuanto a la adquisición de conocimiento, ya que trasegamos en áreas como la electrónica y la programación de dispositivos hardware; pero aún más ampliamente en la planeación y desarrollo de proyectos de mayor envergadura. Para nuestros alcances no se pudo realizar una implementación real debido a los altos costos generados por la infraestructura e incluyendo un método de acceso de mayor calidad y seguridad. Con el apoyo y los recursos necesarios, se podría obtener mayor beneficio de este sistema en el mercado.

En todo caso, se ha podido demostrar que es posible instalar un sistema domótico apoyándonos en la plataforma Arduino, con un coste muy inferior al que se utiliza en las viviendas de lujo, a cambio de dedicarle un poco de tiempo.

11. BIBLIOGRAFIA

- Huidobro J. M., Millán R. J. “Domótica: edificios inteligentes”, Creaciones COPIRIGHT España, 2004.
- Junestrand S., Passaret X. Vázquez D. “Domótica y Hogar Digital”, Thomson Paraninfo, España, 2005.
- Andaluz Ortiz, Víctor, Yépez Rodríguez, Juan. “Diseño y construcción de un control demótico utilizando
- bluetooth por medio de un pda”, Tesis, Escuela Politécnica Nacional, Quito, Ecuador. 2008.
- S.L.Jin, W.S. Yu, S.C. Chung, A. “Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee and
- Wi-Fi, Industrial Electronics Society”. IECON 2007. 33rd Annual Conference of the IEEE. 2007.
- Huidobro J. M., Millán R. J. “Manual de Domótica”, Creaciones COPIRIGHT España, 2010.
- Brian W. Evans. “Arduino Programming Handbook: a Beginner’s Reference”, Editorial, USA, 2 edición, 2008.

- Comunicación serie entre Arduinos:
<http://www.youtube.com/watch?v=FiDaNkuwgQM>
- Datasheet Ethernet Shield:
http://www.nuelectronics.com/estore/index.php?main_page=project_eth
- Información sobre sensores:
<http://www.ladyada.net/learn/sensors/index.html>
- Librería IRremote:
http://www.pjrc.com/teensy/td_libs_IRremote.html
- Librería NECIRrcv:
<http://www.sherkhan.net/blogs/frikadas/?p=331>
- Página oficial de Arduino:
<http://www.arduino.cc/>
- Tutorial Ethernet Shield 1:
<http://www.instructables.com/id/Arduino-Ethernet-Shield-Tutorial/>
- Tutorial Ethernet Shield 2:
<http://bildr.org/2011/06/arduino-ethernet-pin-control/>
- Tutorial Ethernet Shield 3:
<http://arduino.cc/forum/index.php/topic,6595.0.html#0>

