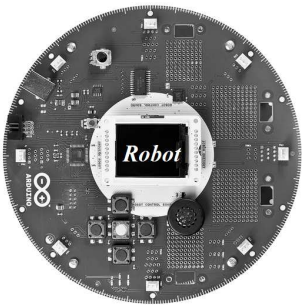
Capítulo 3

Plataforma electrónica



Capítulo Web

3.1 Introducción

3.2 Arquitectura AVR

3.3 Plataforma electrónica Arduino

3.4 Modelos de tarjetas Arduino

3.5 Resumen

3.6 Referencias selectas

3.7 Problemas propuestos

Competencias

Presentar los conceptos fundamentales sobre la arquitectura AVR de los microcontroladores Atmel; plataforma electrónica Arduino, clasificación y sus potenciales aplicaciones.

Desarrollar habilidades en:

- Descripción de la arquitectura AVR.
- Tabla y manejo de interrupciones.
- Organización de la memoria para programas y datos.
- Tipos de periféricos disponibles en las tarjetas Arduino.
- Formas de comunicación serial.
- Gestor de arranque y cargador de código de máquina.
- Aspectos funcionales del modelo Arduino UNO.
- Descripción técnica de las señales para interface electrónica de la tarjeta Arduino UNO.
- Tipos de tarjetas de la plataforma electrónica Arduino y sus potenciales aplicaciones.

3.1 Introducción

Los sistemas empotrados (*embedded systems*) son plataformas electrónicas cuya base lo constituye un microprocesador con el lenguaje de programación adecuado y el soporte de un conjunto de periféricos como puertos I/O, timers, contadores, PWM's, FPGA's e instrumentación electrónica para acoplar señales, sensores, motores y actuadores (convertidores analógica/digital, digital/analógico) dedicado exclusivamente al procesamiento, control y automatización de procesos físicos en tiempo real.

Atmel introduce en 1993 el primer controlador con memoria flash utilizando la arquitectura del popular microcontrolador 8051 de 8 bits y en 1997 coloca en el mercado el primer microcontrolador AT90S1200 con arquitectura AVR. A partir de este momento inicia una clase de microcontroladores Atmel de alto desempeño y eficiencia para utilizarse como sistemas empotrados con aplicaciones diversas en ingeniería robótica y mecatrónica. Los microcontroladores Atmel forman las tarjetas Arduino, cuya versatilidad aumenta debido a que se programan con los lenguajes C/C++, desde un ambiente de programación orientado al desarrollo e implementación de aplicaciones en ingeniería.

Microcontroladores AVR

El nombre AVR deriva de sus desarrolladores **A**lf-Egil Bogen **V**egard Wollan para microcontroladores con arquitectura **R**ISC (*reduced instruction set computing*) que en 1996 trabajaban en la compañía Atmel. El primer microcontrolador prototipo con arquitectura AVR fue el AT90S8515 (en el año 1996) y el primer microcontrolador comercial fue el AT90S1200 en el año 1997.

Arduino es un sistema empotrado dedicado al control y automatización de procesos físicos; toma como plataforma electrónica una familia de microcontroladores Atmel de 8 bits (Atmega 48PA/ 88PA/168PA/328PA) con memoria flash integrada para almacenar el código de los programas, memoria SRAM para datos y un conjunto de circuitos periféricos que le permiten adquirir datos, procesamiento y control de información, así como comunicación serie con la computadora.

Los microcontroladores Atmel de 8 y 32 bits de arquitectura AVR mezclan alta eficiencia y desempeño, optimizando su forma de trabajo con baja potencia, alta velocidad. La arquitectura AVR contiene dos tipos de buses: uno para instrucciones donde el CPU (*central processing unit*) lee y ejecuta instrucciones y otro bus para leer y escribir datos. Esto asegura que una instrucción puede ser ejecutada en cada ciclo de reloj, eliminando estados de espera cuando no hay instrucciones para ejecutar.

La arquitectura AVR tiene múltiples registros de trabajo de propósito general los cuales pueden almacenar datos en forma dinámica dentro del CPU, eliminando la necesidad de mover temporalmente los datos del CPU hacia la SRAM y posteriormente regresar la información (ocupando varios ciclos de reloj). Los registros del CPU AVR son extremadamente rápidos, permitiendo que el CPU lea, ejecute y almacene la información en un registro de trabajo en un simple ciclo de reloj, reduciendo de esta forma la potencia consumida.

La tabla 3.1 muestra las características de diversos modelos de microcontroladores de la compañía Atmel (número de bits que utiliza y frecuencia de operación).

Tabla 3.1 Microcontroladores Atmel

Modelos	Num. de bits	Frecuencia
Atmega48PA	8 bits	8 MHz
Atmega88PA	8 bits	8 MHz
Atmega168PA	8 bits	8 MHz
Atmega328P	8 bits	8 MHz
At32UC3A0128	32 bits	66 MHz
At32UC3A1256	32 bits	66 MHz

Cada instrucción de la arquitectura AVR es múltiplo de 16 bits y la parte clave es que reduce la complejidad del sistema digital requerido para decodificar cada instrucción. Generalmente cuando un CPU ejecuta un programa requiere de accesos frecuentes a un conjunto limitado de datos, apuntadores, lazos de programas, monitoreo de estados de bits, arreglos, índices, etc. Atmel desarrolló un compilador AVR en lenguaje C eficiente y optimizado para que las aplicaciones se realicen con pequeño código y bajo consumo de potencia.

La nueva generación de microcontroladores AVR de 32 bits expanden su potencialidades y aplicaciones, ya que cuentan con un conjunto mayor de instrucciones para realizar operaciones con enteros y punto fijo; además contienen instrucciones de punto flotante tipo DSP (*digital signal processing*) incremento el desempeño y eficiencia del microcontrolador.

3.2 Arquitectura AVR

La plataforma Arduino tiene como base la familia de microprocesadores Atmega 48PA/88PA/168PA/328P con arquitectura AVR-RISC (ver diagrama a bloques en la figura 3.1) fabricados por la empresa ATMEL con tecnología CMOS para obtener bajo consumo, alta flexibilidad en aplicaciones de sistemas embebidos (*embedded systems*) de bajo costo y potente mecanismo de programación.

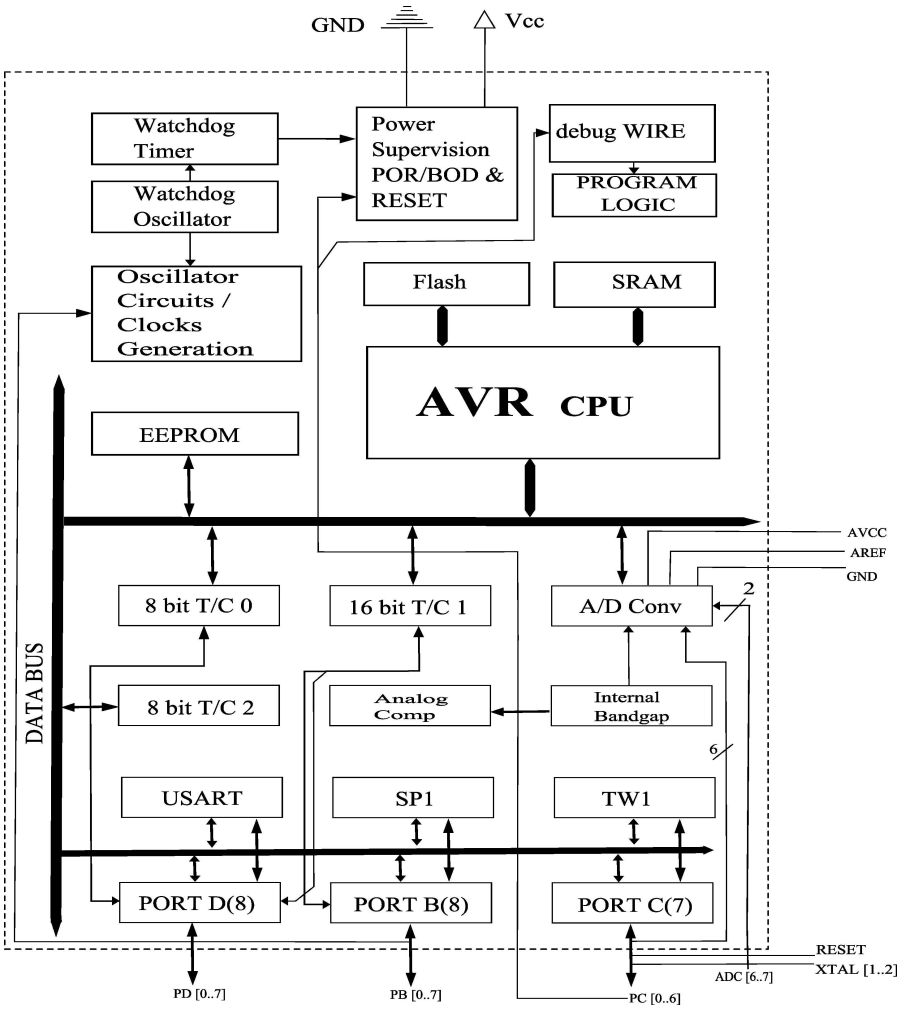


Figura 3.1 Diagrama a bloques de la arquitectura Arduino.

La familia de microcontroladores Atmega se basa en la arquitectura AVR-RISC, CMOS (baja potencia), memoria no volátil de alta densidad y como principal característica puede realizar 1 MIPS (un millón de instrucciones por segundo) en un MHz, este mecanismo de programación le permite obtener alto desempeño y eficiencia en aplicaciones de ingeniería robótica y mecatrónica.

La arquitectura AVR tiene como cerebro electrónico a una unidad central de proceso CPU (*central processing unit*, comúnmente conocida como *processor*), la cual es la parte encargada de ejecutar las instrucciones de un programa y controlar que dicha ejecución se realice correctamente, como parte fundamental del CPU se encuentra el contador de programa (*program counter* PC o *instruction pointer* IP) el cual indica la instrucción actual que se está ejecutando y al terminar ésta, apunta a la siguiente instrucción.

La arquitectura AVR de los microcontroladores Atmega tienen las siguientes características técnicas: memoria programable flash no volátil con mecanismo de lectura mientras escribe (*read-while-write*), la memoria tipo flash almacena instrucciones del programa o sketch del usuario y puede ser programada usando interface serial; memoria estática volátil exclusiva para almacenar datos SRAM (separada de la memoria flash), memoria no volátil EEPROM (*electrically erasable programmable read-only memory*) con un pequeño programa grabado para inicializar y definir el modo de operación de los circuitos periféricos. También contiene 23 puertos digitales (I/O) de propósito general, 32 registros de trabajo de propósito general de 8 bits, funcionan como espacios de memoria dentro del microcontrolador, pueden acceder a la memoria de datos SRAM o EEPROM, almacenan resultados y realizan determinadas operaciones con la unidad lógica aritmética (ALU).

Como parte de la arquitectura AVR hay tres temporizadores/contadores (*timers/counters*) con diferentes modos de comparación y manejo de interrupciones para eventos internos y externos, un dispositivo periférico programable de comunicación serial USART (*universal asynchronous receiver/transmitter*), bus serial con protocolo TWI (*two-wire* también conocido como I^2C *inter-integrated circuit*), un puerto serial SPI (*serial peripheral interface*), convertidor analógico/digital (ADC o *analog to digital converter*) de 10 bits multiplexado en 6 canales de 10 bits (depende del tipo de encapsulado), los dispositivos ADCs tienen una terminal de alimentación

AVCC y emplean la técnica de conversión por aproximaciones sucesivas, por lo que requieren de una señal de referencia (AREF); incorporan también, un dispositivo programable llamado guardián de eventos (*watchdog timer*) con oscilador interno y 5 modos de programación para establecer el consumo eléctrico. Todos estos dispositivos están conectados al bus de datos y por lo tanto tienen asignado direcciones de memoria para los dispositivos flash, SRAM y EEPROM y direcciones I/O para los puertos y circuitos periféricos como timers, USART, TWI, SPI y ADCs.

La señal de RESET es un comando de control considerada como un tipo de interrupción con la mayor prioridad del microcontrolador, generalmente el contador de programa inicia en la localidad 0x0000 de la memoria EEPROM para ejecutar el programa que inicializa registros, modo de operación de puertos y dispositivos periféricos y define el modo de operación de los timers, protocolos de comunicación serial (USART, TWI, SPI), de tal forma que deja al sistema listo para recibir y ejecutar un programa del usuario.

3.2.1 Secciones de memoria

La arquitectura AVR de los microcontroladores Atmega 48PA/88PA/168PA/328P contienen tres espacios diferentes de memoria (lineales y regulares) divididas en: memoria para datos (SRAM), memoria flash para almacenar el código de máquina de las instrucciones de un programa e información del sistema (constantes, direcciones de memoria y puertos I/O, etc.), y el programa pregrabado en la memoria EEPROM que inicializa, define modos de operación y funcionalidad de los puertos y periféricos del sistema; se denomina gestor de arranque y cargador de código de usuario (cargador/arrancador *boot-loader*) también conocido como *firmware*.

Memoria flash para instrucciones de programas flash

Los microcontroladores Atmega 48PA/88PA/168PA/328P contienen chips o circuitos integrados de memoria flash de 4/8/16/32 Kbytes, respectivamente para almacenar el código de máquina de los programas o sketchs de usuario. Sin embargo, debido a que todas las instrucciones tienen formato de 16 y 32 bits, la memoria flash se encuentra direccionada en espacios de 2 Kbytes (0x000 a 0x07FF, 48PA), 4

Kbytes (0x000 a 0x0FFF, 88PA), 8 Kbytes (0x000 a 0x1FFF, 168PA) y 16 Kbytes (0x000 a 0x3FFF, 328P).

La memoria flash de los microcontroladores Atmega 88PA/168PA/328P se divide en dos secciones: un área específica para el arrancador/cargador de programas (*Boot Loader Section*), la cual se encuentra en la parte alta de la memoria flash y otra sección destinada para almacenar el código de máquina del sketch (*Application Program Section*); inicia en la localidad de memoria 0x000. Mientras que para el microprocesador Atmega 48PA solo contiene la sección (*Application Program Section*), con un espacio de memoria de 0x000 a 0x7FF (ver figura 3.2).

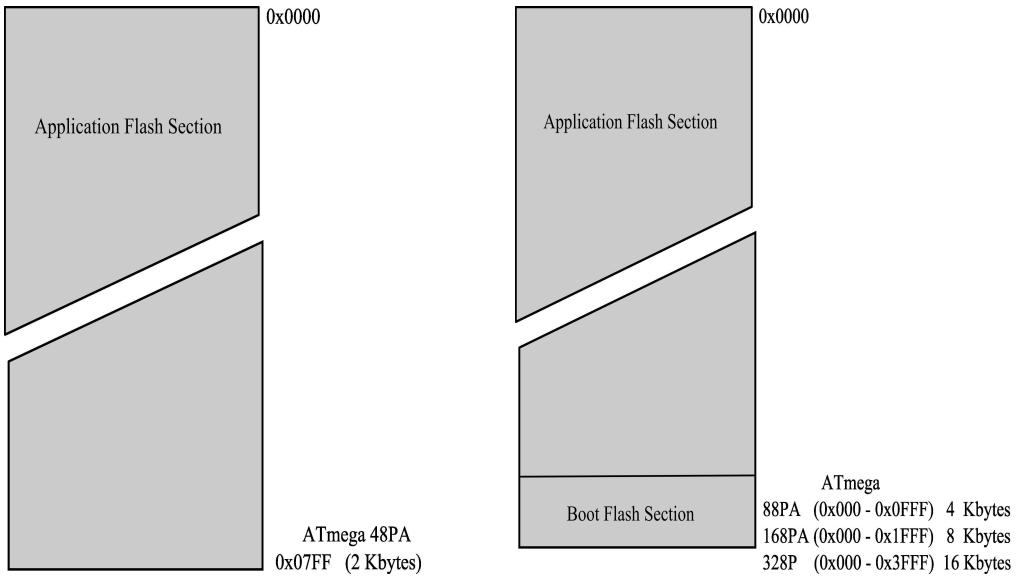
Memoria flash

Memoria flash es un dispositivo electrónica que almacena datos en sistemas digitales, como pueden ser microprocesadores, DSPs o computadoras. Cuando se apaga o se retira la fuente de alimentación guarda la información sin pérdida de datos (memoria no volátil). La memoria flash se borra eléctricamente y de la misma forma se puede reprogramar (es una variante de EEPROM).
La memoria flash de los microprocesadores Atmega tiene una razón de pérdida mucho menor a 1 PPM (parte por millón) en 20 años; la vida útil puede alcanzar 20 años si se opera el dispositivo a 85 °C o 100 años a temperatura de 25 °C.

Flash lectura-escritura-lectura

Los microcontroladores 88PA, 168PA y 328P cuentan con un mecanismo para leer datos mientras escriben (*read-while-write*); el arrancador/cargador de código se encuentra en sección (*Application Program Section*) y se encuentra separada del área de instrucciones de programas. El microcontrolador 48PA no cuenta con dicho mecanismo, y no tiene dividida el área de memoria flash.

El contador del programa PC (*program counter*) de los microprocesadores Atmega 48PA/88PA/168PA/328P tiene una longitud respectivamente de 11/12/13/14 bits, lo que le permite direccionar localidades de memoria de 2/4/8/16 Kbytes. La forma de trabajar es direccionar la localidad de memoria donde se encuentra la instrucción, decodificarla y ejecutarla; mientras realiza esta actividad, un subsistema del CPU tiene decodificada la siguiente instrucción del programa para que sea ejecutada en forma secuencial.



(a) Atmega 48PA **(b)** Atmega 88PA/168PA/328P

Figura 3.2 Mapa de memoria de los microcontroladores 48PA/88PA/168PA/328P.

3.2.2 Memoria de datos SRAM

La memoria SRAM sirve para almacenar datos, variables, constantes o información que requiera el programa del usuario. Los microcontroladores Atmega 48PA/88PA/168PA/328P tienen asignados puertos de entrada/salida (I/O) en espacios de memoria, así como para dispositivos periféricos.

En la memoria SRAM existe el espacio extendido de puertos entrada/salida (I/O) el cual comprende de la dirección 0x60 a la 0xFF, en este espacio de memoria sólo las instrucciones de lenguaje ensamblador ST/STS/STD y LD/LDS/LDD pueden ser usadas. El espacio de periféricos contiene 64 localidades reservadas para manejo de puertos por medio de instrucciones IN/OUT.

La tabla 3.2 muestra el mapa de memoria asignado al manejo de datos y puertos y en la tabla 3.3 presenta las diferencias técnicas que hay entre los diversos modelos de microcontroladores Atmega.

Tabla 3.2 Mapa de memoria SRAM de datos.

Descripción	dirección de memoria
32 registros de trabajo de propósito general	0x0000 - 0x001F
64 registros de entrada/salida (I/O)	0x0020 - 0x005F
160 registros externos entrada/salida (I/O)	0x0060 - 0x00FF
SRAM interna Atmega 48PA 512 bytes x 8 bits Atmega 88PA 1024 bytes x 8 bits Atmega 168PA 1024 bytes x 8 bits Atmega 328P 2048 bytes x 8 bits	SRAM inicia en 0x0100 Atmega 48PA 0x0100 - 0x1FF Atmega 88PA 0x0100 - 0x3FF Atmega 168PA 0x0100 - 0x3FF Atmega 328P 0x0100 - 0x7FF

Tabla 3.3 Detalles técnicos de 48PA/88PA/168PA/328P.

Dispositivo	Flash	EEPROM	RAM	Tamaño del vector de interrupción
Atmega 48PA	4 Kbytes	256 bytes	512 bytes	1 word/vector
Atmega 88PA	8 Kbytes	512 bytes	1 Kbytes	1 word/vector
Atmega 168PA	16 Kbytes	512 bytes	1 Kbytes	2 word/vector
Atmega 328P	32 Kbytes	1 Kbytes	2 Kbytes	2 word/vector

Puertos I/O

Todos los puertos de la arquitectura AVR son puertos de entrada/salida (I/O), cada puerto tiene resistencias individuales de *pull-up* (inicialmente desconectadas y que puede ser configuradas por software) y diodos de protección para la fuente de alimentación VCC y la referencia o tierra GND. A través de los puertos entrada/salida es la forma en que el microcontrolador se puede comunicar con el mundo exterior. Cuando un puerto es configurado como salida (OUT) puede suministrar la corriente necesaria para alimentar a indicadores luminosos como LEDs.

La arquitectura electrónica AVR de la figura 3.1 contiene un total de 23 líneas de puertos digitales entrada/salida agrupadas en 3 puertos B, C y D. Sin embargo, varias líneas de puertos tienen funciones alternas.

El puerto B contiene 8 líneas (PB0:PB7) bidireccionales, es decir programables como

entrada/salida con resistencias internas *pull-up*, cada línea tiene 3 estados (entrada, salida y alta impedancia). Adicionalmente, las líneas PB6 y PB7 tienen funciones alternas, como entrada y salida del circuito oscilador, respectivamente.

El puerto C (PC0:PC6) es un conjunto de 7 líneas configurables como entrada/salida. Adicionalmente la línea PC6 puede ser programada como una entrada de la señal de reset. El puerto D (PD0:PD7) de 8 bits donde todas sus líneas son bidireccionales (entrada/salida), con resistencias internas seleccionadas por software.

Otros modelos ATMEGA como el microcontrolador Atmega16 contiene 4 grupos de puertos A, B, C y D, cada grupo de puertos son de 8 bits, con un total de 32 líneas digitales de entrada/salida.

3.2.3 Unidad lógica aritmética ALU

La arquitectura AVR tiene alto desempeño en procesamiento de operaciones aritméticas debido que la ALU trabaja directamente con 32 registros propósito general de 8 bits; utiliza un ciclo de reloj para ejecutar operaciones aritméticas con los registros de propósito general, también se auxilia para realizar operaciones entre registros y registros con constantes. El tipo de operaciones que realiza la ALU son: aritméticas, lógicas y funciones de bits. También soporta operaciones en formatos con signo, sin signo y fraccional.

3.2.4 Temprizadores/Contadores

La arquitectura AVR de la figura 3.1 muestra 3 temporizadores/contadores (timers/counters). Los circuitos Timer/Counter0 y Timer/Counter1 son temporizadores/contadores de propósito general de 8 bits con dos unidades de comparación de coincidencias independientes y con soporte para generar señales electrónicas PWM (*pulse width modulation*), así como la unidad programable bi-direccional (incremento/decremento). Las señales de entrada a los circuitos Timer/Counter pueden ser internas o fuentes externas. Adicionalmente tiene un circuito temporizador/contador de 16 bits para manejo de eventos, generación de ondas, y medición de tiempo en

señales de instrumentación electrónica.

3.2.5 Interface periférica serial SPI

La interface periférica serial SPI (*serial peripheral interface*) permite transferencia de datos con alta velocidad en forma síncrona entre la familia de microcontroladores Atmega 48PA/88PA/168PA/328P y dispositivos periféricos o dispositivos AVR.

3.2.6 Comunicación serial UART

El dispositivo receptor y transmisor serial universal síncrono/asíncrono USART (*universal synchronous and asynchronous serial receiver and transmitter*) es un sistema de comunicación serial. Contiene unidades de recuperación de datos para recepción de información asíncrona; el receptor y transmisor incluyen el protocolo de comunicación definiendo la detección de paridad, bit de control, registro de corrimiento, mensaje de error y sobreflujo de datos. El USART puede ser programado para funcionar en modo SPI-maestro.

3.2.7 Bus de interface serial

El bus de interface serial TWI (*2-wire serial interface bus*) permite conectar hasta 128 diferentes dispositivos usando dos líneas bidireccionales, cada línea con resistencias de *pull-up*; una línea se emplea para la señal del reloj y la otra para transferencia de datos SDA. Los dispositivos conectados a este bus tienen direcciones diferentes y el mecanismo para resolver el tráfico de información es el protocolo TWI.

3.2.8 Convertidor analógico/digital

La familia de microprocesadores Atmega 48PA/88PA/168PA/328P tiene un convertidor analógico/digital ADC (*analog-to-digital converter*) usando la técnica de conversión por aproximaciones sucesivas con resolución de 10 bits. El ADC

está conectado a un multiplexor analógico de 8 canales, permitiendo de esta forma 8 entradas de voltaje de señales analógicas (referidas a tierra GND) directamente de los pins del puerto A. El ADC contiene un circuito muestreador y retenedor (*sampler and hold*) que asegura mantener constante el nivel de voltaje durante el proceso de conversión.

El ADC tiene un pin para su propia fuente de alimentación AVCC (separada de la fuente VCC), la cual puede variar hasta ±0.3 V con respecto a la fuente de alimentación VCC. Como parte del circuito integrado del microcontrolador, un voltaje de referencia interno de 1.1 V es proporcionado AVCC. El mínimo valor de conversión del ADC es la tierra o GND y el máximo valor lo representa el voltaje del pin de AREF menos un bit. Opcionalmente AVCC o la referencia interna 1.1 V pueden ser conectados al pin AREF

3.2.9 Interrupciones

Una interrupción es una solicitud de servicio para que el microcontrolador realice una tarea de cómputo, suspendiendo momentáneamente todas las actividades que está procesando, cuando el microprocesador finaliza la actividad solicitada, entonces retorna al programa principal para continuar ejecutando las instrucciones en forma secuencial.

La forma en que funcionan las interrupciones en los microcontroladores Atmel es por prioridad o gerarquía, la figura 3.3 muestra el diagrama de flujo de interrupciones en los microcontroladores Atmega. La interrupción de mayor prioridad es RESET (por hardware: ya sea por el botón de reset o al conectar la fuente de alimentación; también se puede generar por software). Si el proceso 1 solicita servicio de interrupción mientras se encuentra el microcontrolador atendiendo la interrupción del proceso 0, entonces su ejecución dependerá del nivel de prioridad, es decir si la interrupción del proceso 1 es de menor prioridad, tendrá que esperar a

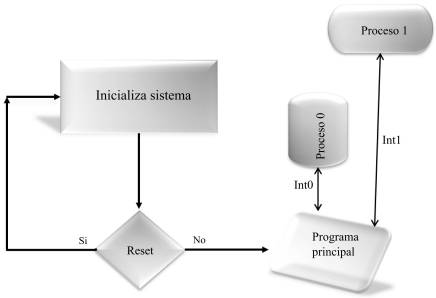


Figura 3.3 Reset.

que termine la interrupción del proceso 0, de otra manera la interrupción del proceso 1 se ejecutará inmediatamente, suspendiendo momentáneamente la interrupción del proceso 0, al terminar la tarea del proceso 1, retornará a las actividades pendientes de la interrupción anterior.

En los microcontroladores Atmega hay dos tipos de interrupciones, el primer tipo corresponde a eventos externos que generan un estado lógico o una transición como un pulso electrónico (*triggered*, se detecta por flanco bajo hacia alto o viceversa), esto habilita una bandera de interrupción (*interrupt flag*), dependiendo de la prioridad de interrupción, el contador del programa (*program counter*) toma la dirección de memoria de la tabla de vectores y salta a la localidad de memoria correspondiente donde se encuentra la rutina de la interrupción solicitada (*interrupt handling routine*).

De manera automática, por hardware se limpia la bandera de interrupción (*interrupt flag*); también se puede limpiar las banderas de interrupción por software, ya que tienen asociados sus respectivos bits de habilitación en el registro de estado (*status register*).

Similarmente, si más solicitudes de interrupción ocurren mientras se encuentra en proceso alguna interrupción, permanecerán en espera por orden de prioridad. Cuando el contador de programa sale de una interrupción, retornará al programa principal y ejecutará una o más instrucciones antes de atender alguna interrupción pendiente. El segundo tipo de interrupciones corresponden a las interrupciones que pueden ser cambiadas o reasignadas por software en los pins del microcontrolador, estas interrupciones no necesariamente tienen banderas de interrupción.

Los microprocesadores Atmega tienen diferentes tipos de interrupciones, y las direcciones (vector de interrupción) donde residen las rutinas de servicio de interrupción (programas que atienden la solicitud de interrupción) se encuentran en las primeras localidades de memoria del espacio de programa.

El vector de interrupciones de los microcontroladores Atmega 168PA y 328P tiene una longitud de 32 bits y 16 bits para los microcontroladores Atmega 48PA y 88PA.

Las tablas 3.4, 3.6, 3.6 y 3.7 muestran los vectores de interrupción de los microcontroladores Atmega48PA/88PA/168PA/328P, respectivamente.

Tabla de vectores

A partir de la dirección 0x000 del espacio de memoria flash se almacena la tabla de vectores (direcciones de la memoria flash donde se encuentran las rutinas de servicio de interrupción), la lista completa de interrupciones para el microcontrolador Atmega48PA se muestra en la tabla 3.4, esta lista se ordena en base a los niveles de prioridad de los diferentes tipos de interrupciones, por ejemplo una interrupción generada por RESET tiene asignada la dirección 0x000 y por lo tanto, representa la interrupción de mayor prioridad, la segunda interrupción en jerarquía se encuentra en la dirección 0x001 corresponde a la solicitud de interrupción externa 0 (INT0), así sucesivamente hasta llegar a la de menor prioridad que corresponde a la interrupción 26 ubicada en la dirección 0x019.

Vector de interrupciones

El vector de interrupciones es una lista indexada de localidades de memoria de cada interrupción, cada localidad de memoria contiene la dirección de memoria donde se ubica la rutina de servicio de interrupción, es decir el programa que atiende el proceso de solicitud de un evento o proceso. Cuando una interrupción es generada, se salva la dirección del contador de programa, para saltar a la correspondiente rutina de interrupción indicada por la tabla de vectores, al finalizar la rutina de interrupción, entonces el contador de programa regresa a la localidad de memoria donde estaba ejecutando el programa principal.

Actualmente los microcontroladores AVR Atmega 48PA/88PA/168PA/328P tienen paquetes de cómputo con herramientas integradas que permiten realizar la programación mediante compiladores de lenguaje C, macro ensambladores, debugger/simuladores, emuladores de circuitos, y kits de evaluación disponibles en el siguiente sitio Web:

http://www.atmel.com/avr

Tabla 3.4 Reset y vector de interrupciones de Atmega 48PA.

Número de vector	Localidad	Fuente	Características
1	0x000	RESET	Pin externo, reset al encender (*power-on*), apagón parcial (*brown-out*), reset generado por *watchdog system.*
2	0x001	INT0	Solicitud de interrupción externa 0.
3	0x002	INT1	Solicitud de interrupción externa 1.
4	0x003	PCINT0	Solicitud de interrupción cambio de pin 0.
5	0x004	PCINT1	Solicitud de interrupción cambio de pin 1.
6	0x005	PCINT2	Solicitud de interrupción cambio de pin 2.
7	0x006	WDT	Interrupción tiempo-muerto del Watchdog.
8	0x007	TIMER2 COMPA	Timer/Contador2
9	0x008	TIMER2 COMPB	Timer/Contador2 compara Match B.
10	0x009	TIMER2 OVF	Sobreflujo del Timer/Contador2.
11	0x00A	TIMER1 CAPT	Captura de evento del Timer/Contador1.
12	0x00B	TIMER1 COMPA	Compara Match A del Timer/Contador1.
13	0x00C	TIMER1 COMPB	Compara Match B del Timer/Contador1.
14	0x00D	TIMER1 OVF	Sobreflujo del del Timer/Contador1.
15	0x00E	TIMER0 COMPA	Compara Match A del Timer/Contador0.
16	0x00F	TIMER0 COMPB	Compara Match B del Timer/Contador0.
17	0x010	TIMER0 OVF	Sobreflujo del Timer/Contador0.
18	0x011	SPI, STC	Transferencia serial completa SPI.
19	0x012	USART, RX	Recepción serial completa RX.
20	0x013	USART, UDRE	Registro vacío de datos.
21	0x014	USART, TX	Transmisión serial completa TX.
22	0x015	ADC	Conversión completa del ADC.
23	0x016	EE READY	Memoria lista EEPROM.
24	0x017	ANALOG COMP	Comparador analógico.
25	0x018	TWI	Interface serial 2-cables.
26	0x019	SPM READY	Listo almacena programa en memoria.

Tabla 3.5 Reset y vector de interrupciones de Atmega 88PA.

Número de vector	Localidad[2]	Fuente	Características
1	0x000[1]	RESET	Pin externo, reset al encender (*power-on*), apagón parcial (*brown-out*), reset generado por *watchdog system.*
2	0x001	INT0	Solicitud de interrupción externa 0.
3	0x002	INT1	Solicitud de interrupción externa 1.
4	0x003	PCINT0	Solicitud de interrupción cambio de pin 0.
5	0x004	PCINT1	Solicitud de interrupción cambio de pin 1.
6	0x005	PCINT2	Solicitud de interrupción cambio de pin 2.
7	0x006	WDT	Interrupción tiempo-muerto del Watchdog.
8	0x007	TIMER2 COMPA	Timer/Contador2
9	0x008	TIMER2 COMPB	Timer/Contador2 compara Match B.
10	0x009	TIMER2 OVF	Sobreflujo del Timer/Contador2.
11	0x00A	TIMER1 CAPT	Captura de evento del Timer/Contador1.
12	0x00B	TIMER1 COMPA	Compara Match A del Timer/Contador1.
13	0x00C	TIMER1 COMPB	Compara Match B del Timer/Contador1.
14	0x00D	TIMER1 OVF	Sobreflujo del del Timer/Contador1.
15	0x00E	TIMER0 COMPA	Compara Match A del Timer/Contador0.
16	0x00F	TIMER0 COMPB	Compara Match B del Timer/Contador0.
17	0x010	TIMER0 OVF	Sobreflujo del Timer/Contador0.
18	0x011	SPI, STC	Transferencia serial completa SPI.
19	0x012	USART, RX	Recepción serial completa RX.
20	0x013	USART, UDRE	Registro vacío de datos.
21	0x014	USART, TX	Transmisión serial completa TX.
22	0x015	ADC	Conversión completa del ADC.
23	0x016	EE READY	Memoria lista EEPROM.
24	0x017	ANALOG COMP	Comparador analógico.
25	0x018	TWI	Interface serial 2-cables.
26	0x019	SPM READY	Listo almacena programa en memoria.

Tabla 3.6 Reset y vector de interrupciones de Atmega 168PA.

Número de vector	Localidad[2]	Fuente	Características
1	0x000[1]	RESET	Pin externo, reset al encender (*power-on*), apagón parcial (*brown-out*), reset generado por *watchdog system.*
2	0x002	INT0	Solicitud de interrupción externa 0.
3	0x004	INT1	Solicitud de interrupción externa 1.
4	0x006	PCINT0	Solicitud de interrupción cambio de pin 0.
5	0x008	PCINT1	Solicitud de interrupción cambio de pin 1.
6	0x00A	PCINT2	Solicitud de interrupción cambio de pin 2.
7	0x00C	WDT	Interrupción tiempo-muerto del Watchdog.
8	0x00E	TIMER2 COMPA	Timer/Contador2
9	0x010	TIMER2 COMPB	Timer/Contador2 compara Match B.
10	0x012	TIMER2 OVF	Sobreflujo del Timer/Contador2.
11	0x014	TIMER1 CAPT	Captura de evento del Timer/Contador1.
12	0x016	TIMER1 COMPA	Compara Match A del Timer/Contador1.
13	0x018	TIMER1 COMPB	Compara Match B del Timer/Contador1.
14	0x01A	TIMER1 OVF	Sobreflujo del del Timer/Contador1.
15	0x01C	TIMER0 COMPA	Compara Match A del Timer/Contador0.
16	0x01E	TIMER0 COMPB	Compara Match B del Timer/Contador0.
17	0x020	TIMER0 OVF	Sobreflujo del Timer/Contador0.
18	0x022	SPI, STC	Transferencia serial completa SPI.
19	0x024	USART, RX	Recepción serial completa RX.
20	0x026	USART, UDRE	Registro vacío de datos.
21	0x028	USART, TX	Transmisión serial completa TX.
22	0x02A	ADC	Conversión completa del ADC.
23	0x02C	EE READY	Memoria lista EEPROM.
24	0x02E	ANALOG COMP	Comparador analógico.
25	0x030	TWI	Interface serial 2-cables.
26	0x032	SPM READY	Listo almacena programa en memoria.

Tabla 3.7 Reset y vector de interrupciones de Atmega 328P.

Número de vector	Localidad[2]	Fuente	Características
1	0x000[1]	RESET	Pin externo, reset al encender (*power-on*), apagón parcial (*brown-out*), reset generado por *watchdog system.*
2	0x002	INT0	Solicitud de interrupción externa 0.
3	0x004	INT1	Solicitud de interrupción externa 1.
4	0x006	PCINT0	Solicitud de interrupción cambio de pin 0.
5	0x008	PCINT1	Solicitud de interrupción cambio de pin 1.
6	0x00A	PCINT2	Solicitud de interrupción cambio de pin 2.
7	0x00C	WDT	Interrupción tiempo-muerto del Watchdog.
8	0x00E	TIMER2 COMPA	Timer/Contador2
9	0x010	TIMER2 COMPB	Timer/Contador2 compara Match B.
10	0x012	TIMER2 OVF	Sobreflujo del Timer/Contador2.
11	0x014	TIMER1 CAPT	Captura de evento del Timer/Contador1.
12	0x016	TIMER1 COMPA	Compara Match A del Timer/Contador1.
13	0x018	TIMER1 COMPB	Compara Match B del Timer/Contador1.
14	0x01A	TIMER1 OVF	Sobreflujo del del Timer/Contador1.
15	0x01C	TIMER0 COMPA	Compara Match A del Timer/Contador0.
16	0x01E	TIMER0 COMPB	Compara Match B del Timer/Contador0.
17	0x020	TIMER0 OVF	Sobreflujo del Timer/Contador0.
18	0x022	SPI, STC	Transferencia serial completa SPI.
19	0x024	USART, RX	Recepción serial completa RX.
20	0x026	USART, UDRE	Registro vacío de datos.
21	0x028	USART, TX	Transmisión serial completa TX.
22	0x02A	ADC	Conversión completa del ADC.
23	0x02C	EE READY	Memoria lista EEPROM.
24	0x02E	ANALOG COMP	Comparador analógico.
25	0x030	TWI	Interface serial 2-cables.
26	0x032	SPM READY	Listo para almacenar programa en memoria.

3.2.10 Núcleo de la arquitectura AVR Atmega

El núcleo de la arquitectura AVR de los microprocesadores Atmega utiliza bloques memoria flash y SRAM separada para programas y datos, respectivamente; con la finalidad de maximizar desempeño y paralelismo de procesamiento de instrucciones por ciclo de reloj y asegurar la correcta ejecución del programa, acceso a memoria, cálculos matemáticos, comunicación y control de señales electrónicas con los periféricos, así como manejo de interrupciones.

La figura 3.4 muestra el diagrama a bloques de la arquitectura AVR del CPU.

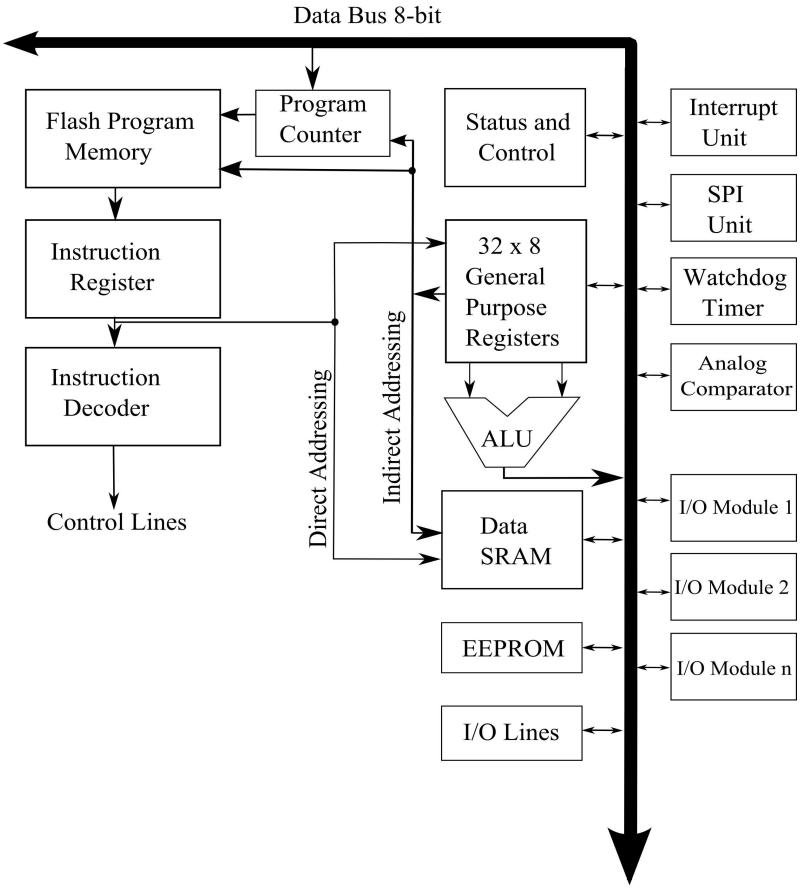


Figura 3.4 Diagrama a bloques del núcleo AVR Atmega.

Las instrucciones en la memoria flash del programa son ejecutadas con un simple nivel de *pipelining*, esto significa que mientras una instrucción está siendo ejecutada, la siguiente instrucción ya está siendo seleccionada de la memoria del programa, decodificada y preparada para su ejecución, con esta técnica es posible ejecutar instrucciones cada ciclo de reloj, es decir un MIPS (un millón de instrucciones por segundo) en un MHz; algunos microprocesadores corren a 16 MHz, entonces pueden ejecutar 16 MIPS.

La arquitectura AVR combina instrucciones de programación con un conjunto de 32 registros de trabajo de propósito general de 8 bits conectados a la unidad lógica aritmética (ALU *arithmetic logic unit*), para realizar operaciones matemáticas en un ciclo de reloj, permitiendo mayor eficiencia y rapidez que otros microcontroladores convencionales.

Seis de los 32 registros de propósito general son usados como apuntadores de 16 bits (registros X, Y, Z) para direccionar localidades de memoria de datos. Uno de estos registros se emplea para acceder a la tabla de vectores de la memoria flash. Cierto tipo de operaciones aritméticas pueden ser realizadas con estos registros a través de ALU.

La mayoría de las instrucciones de programación tienen un formato de 16 bits (*16-bit word format*). Cada dirección de memoria de programa contiene instrucciones de 16 o 32 bits.

El espacio de memoria flash para el programa es dividida en dos secciones: la sección del programa arrancador (*boot program*) y la sección de la aplicación del programa La sección del arrancador tiene bits de protección para escritura y lectura/escritura; la instrucción SPM que escribe dentro de la sección de memoria flash de aplicación del programa reside dentro de la sección del programa arrancador.

La memoria Stack o pila se encuentra localizada en la sección de datos de la memoria estática SRAM y por lo tanto su tamaño es limitado. Durante las interrupciones y llamda a subrutinas, la dirección de retorno del contador del programa PC (*program counter*) se almacena en la memoria del Stack.

El espacio de memoria en la arquitectura AVR es lineal y compuesta de mapas regulares de memoria. El espacio de memoria para puertos de entrada/salida (I/O) contiene 64 direcciones para funciones que manejan dispositivos periféricos como registros de control, SPI, puertos. El acceso de la memoria I/O puede ser directo o como localidades de espacio de datos (0x20 a la 0x5F) usando el registro de archivo (*Register file*). Adicionalmente los microprocesadores Atmega 48PA/88PA/168PA/328P tienen un espacio extendido de (I/O) para las direcciones 0x60 a la 0xFF en memoria estática SRAM donde sólo las instrucciones ensamblador ST/STS/STD y LD/LDS/LDD puede ser usadas.

La unidad de atención y control de solicitudes de interrupción del CPU le da versatilidad al microcontrolador, las interrupciones representan un atributo importante para todo sistema empotrado que le permite atender procesos del mundo exterior sin descuidar actividades relevantes en la ejecución del programa del usuario; contiene una tabla de vectores indexados en forma prioritaria con la direcciones de la memoria flash donde se ubican las rutinas de servicio de interrupción.

Por ejemplo, el control de robots manipuladores requiere adquisición de datos, saber cada 2.5 mseg la posición actual del robot, esto significa que un conjunto de encoders (sensores de posición digitales optoelectrónicos) son conectados a los puertos digitales del microcontrolador y no tendría sentido que el microcontrolador estuviera sensando (*polling*) a cada encoder para detectar que ya tienen listo el dato de posición. Esto sería un proceso ineficiente, y pérdida de tiempo innecesario. En su lugar, se utiliza la interrupción INT0 que después del RESET es la de mayor prioridad, garantizando la adquisición de datos y procesamiento de la información en tiempo y forma. Un disparo electrónico o señal de control del encoder indica al microcontrolador el momento exacto para realizar la lectura de posición del robot.

Para trabajar control de procesos en tiempo real, los timers del microcontrolador pueden ser programados para generar una onda cuadrada periódica y producir una frecuencia de muestreo adecuada para realizar las siguientes actividades: adquisición de datos de los sensores de posición, generar la señal de error de posición, cálculos de las operaciones matemáticas que involucra la ley de control (ganancias proporciona, derivativa, funciones trigonométricas e hiperbólicas) y envío de la información como señal de comando a los servomotores del robot para producir desplazamiento

mecánico. Este ciclo se repite cada periodo de muestreo, es decir control en tiempo real.

La incorporación de convertidores analógico/digital, puertos entrada/salida, timers, dispositivos de comunicación serial como USART y diversos periféricos a los microcontroladores Atmel con arquitectura AVR, adicional a las características tecnológicas que poseen los hacen ideales para utilizarse como sistemas empotrados con amplias aplicaciones en ingeniería robótica y mecatrónica.

3.3 Plataforma electrónica Arduino

La tarjeta electrónica más utilizada y popular es el modelo Arduino UNO la cual emplea arquitectura AVR utilizando el microcontrolador ATMega328P y un conjunto de periféricos que la hacen un sistema de instrumentación electrónica de propósito general ideal para ingeniería robótica y mecatrónica.

A continuación se describen las características técnicas y ventajas principales del modelo Arduino UNO.

3.3.1 Modelo Arduino UNO

La tarjeta electrónica más utilizada del sistema Arduino es el modelo UNO la cual utiliza el microcontrolador ATMega328P del fabricante ATMEL. La letra "P" representa la tecnología "picopower" que incorpora el chip del microcontrolador, teniendo como ventaja un menor consumo de energía. El modelo Arduino UNO es la base para entender otras tarjetas Arduino, por lo que, resulta clave entender la parte operativa. La figura 3.5 muestra la forma que tiene la tarjeta electrónica modelo Arduino UNO.

La tabla 3.8 presenta la nomenclatura y características de las señales que presenta los conectores tipo pin-hembra que se encuentran en la cara superior de la tarjeta Arduino UNO (ver figura 3.5). La forma de interaccionar la tarjeta electrónica con el mundo exterior es a través de estos conectores, en cuyos pins-hembra se encuentran

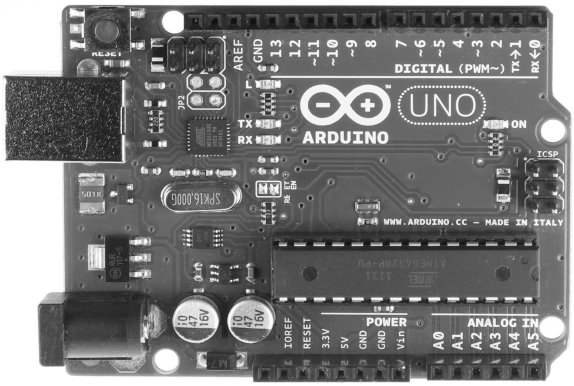


Figura 3.5 Tarjeta electrónica modelo Arduino UNO.

los puertos digitales entrada/salida, señales analógicas, PWM, comunicación serial Rx/Tx, señales de alimentación y representan la forma de colocar sensores, motores, actuadores e interfaces electrónicas.

La alimentación eléctrica de esta tarjeta electrónica es de 5 V y se puede suministrar en dos formas distintas: fuentes externas de alimentación y cable USB.

Las fuentes externas pueden ser por medio de un adaptador eléctrico AC/DC o utilizando pilas o baterías.

A continuación se describe las dos formas de conectar las fuentes externas a la tarjeta Arduino UNO:

- La primera opción de las fuentes externas es por medio de un adaptador eléctrico AC/DC de 7 a 12 V (de 500 mA a 1 A) directamente al conector *jack power* (ver figura 3.5). La polaridad del adaptador es con el positivo al centro, es decir la parte externa del cilindro metálico del conector del adaptador representa el borne negativo.

- La segunda forma es por medio de una pila o batería de 9 V conectándola en los pines-hembra marcados como "V_{IN}" (borne positivo de la pila) y el borne negativo a "GND".

El voltaje de las fuentes externas (7 a 12 V) será regulado a 5 V por un circuito integrado (regulador de voltaje) ya incluido en el modelo Arduino UNO.

Tabla 3.8 Conector pins-hembra del modelo Arduino UNO.

Grupo de señales	Pin-hembra	Señal	Función
"POWER"	5V	5 V	Salida: proporciona 5 V regulados (40 mA). Entrada: alimentación de la tarjeta 7 a 20 V.
	3.3V	3.3 V	Salida: proporciona 3.3 V con margen de error del 1 % (50 mA).
	V_{IN}	V_{IN}	Salida: suministra el mismo nivel de voltaje de la fuente externa (no regulado a 5 V) para alimentar a otros circuitos. En este caso se requiere una fuente de alimentación externa (de 7 a 12 V) acoplada al conector *jack power*. Si la fuente de alimentación es por cable USB, la salida será de 5 V regulados. Entrada: cuando se desea alimentar la tarjeta electrónica mediante pilas de 9 V (sin utilizar conector *jack power* o cable USB).
	GND	GND	Referencia o tierra.
"Digital"	pins 0 al 13	I/O	puertos digitales entrada/salida.
"Analog in"	pins A0 al A5	A0-A5	6 canales analógicos (10 bits por canal).
	pin AREF	AREF	Voltaje de referencia para la conversión analógica/digital.
"PMW∼"	pins 3∼, 5∼, 6∼, 9∼, 10∼ y 11∼	Salidas PWM	6 salidas de señal PWM (8 bits por señal).
"Rx/Tx"	pins 0 y 1	Rx/Tx	Comunicación serial Rx/Tx.
"Interruociones"	pins 2 y 3	INT0/INT1	Interrupciones INT0 y INT 1.
"SPI"	pins 10 al 13	SPI	Comunicación SPI.
"LED"	pin 13	LED	Indicador luminoso integrado LED.
"I^2C/TWI"	pins A4 y A5	LED	Protocolo de comunicación I^2C/TWI.
"RESET"	pin RESET	RESET	Interrupción de RESET.
"IOREF"	pin IOREF	RESET	Similar a la función de salida de V_{IN}.

La segunda opción de alimentación es por medio de un cable estándar USB;

en este caso el voltaje aplicado ya está regulado a 5 V, proporcionando una corriente de 500 mA, es decir una potencia de 2.5 W. Es importante indicar al usuario que la tarjeta electrónica ya viene protegida con un polifusible que al sobrepasar la magnitud de la corriente indicada, se abre automáticamente hasta que las condiciones eléctricas sean normales.

Sin importar el tipo de fuente de alimentación, la tarjeta electrónica Arduino UNO detecta automáticamente una u otra forma de suministrar el voltaje requerido y por lo tanto, no se requiere ningún proceso adicional.

Señales pins-hembra "POWER"

En la tarjeta electrónica Arduino UNO hay una serie de señales (pines-hembra) en la zona del circuito impreso con la leyenda "POWER" (ver figura 3.5) relacionados con la alimentación eléctrica de la siguiente manera:

- El pin-hembra "5 V" puede ser salida/entrada. Funciona como **salida** si se cuenta con una fuente externa de alimentación, donde el adaptador AC/DC se encuentre conectado eléctricamente al conector *jack power* o la alimentación de la tarjeta electrónica sea proporcionada por la computadora por medio del cable USB; entonces, este pin "5 V" suministra un voltaje de salida de 5 V regulados para alimentar a otros circuitos o componentes dentro de los márgenes de seguridad, la corriente máxima generada será de 40 mA. También puede funcionar como **entrada** para alimentar a la tarjeta electrónica Arduino UNO conectando a este pin una **fuente regulada de 5 V** (y al pin "GND") sin utilizar el cable USB, ni la fuente de alimentación externa acoplada al conector *jack power*.

- "GND" es la referencia o tierra eléctrica de la tarjeta electrónica y en este pin-hembra deben ser conectados todas las tierras de los componentes o circuitos acoplados a la tarjeta electrónica (para tener una tierra en común).

- El pin-hembra "3.3 V" es un voltaje regulado de salida que se obtiene de la fuente de alimentación externa o del cable USB. La tarjeta electrónica tiene un circuito regulador integrado LP2985 para proporcionar 3.3 V a 50 mA (con

un margen de error del 1 %) para alimentar a otros componentes o circuitos. No conectar a este pin ninguna fuente de alimentación.

El pin-hembra "V_{IN}" es un pin de entrada/salida. Funciona como **salida** sólo en el caso que tengamos una fuente de alimentación externa (de 7 a 12 V) acoplada al conector *jack power*, entonces podemos alimentar a circuitos u otros componentes con el mismo nivel de voltaje que esté proporcionando la fuente externa (sin regular a 5 V); en el caso de que la fuente de alimentación sea por el cable USB, entonces este pin-hembra suministrará 5 V regulados. Sin importar la forma de alimentación de la tarjeta electrónica, la corriente máxima que suministra este pin-hembra es de 40 mA. Funciona como **entrada** cuando se desea alimentar la tarjeta electrónica mediante una fuente externa utilizando pilas o baterías de 9 V (sin utilizar el adaptador acoplado al conector *jack power* o cables USB). El borne positivo de la pila irá conectado a este pin-hembra "V_{IN}" y el borne negativo al pin-hembra "GND". El regulador interno de la tarjeta 5 V limitará el voltaje de entrada de 9 V a 5 V.

Puertos digitales entrada/salida

En los pins-hembra se disponen de 14 líneas de puertos digitales entrada/salida numeradas del pin 0 al pin 13. En estos pins se pueden conectar sensores, actuadores e interfaces electrónicas para comunicarse con el mundo exterior. El voltaje que pueden proveer o recibir estas líneas digitales es de 5 V a 40 mA. Por software se puede configurar una resistencia interna de *pull-up* de 20 KΩ a 50 KΩ que inicialmente está desconectada.

Entradas analógicas

Un total de 6 entradas analógicas etiquetadas con la nomenclatura A0$\cdots$A5 en los pins-hembra. Cada entrada puede recibir voltaje continuo en el rango de 0 a 5 V, con una resolución de 10 bits (2^{10}=1024 valores diferentes), para un voltaje de 5V la mínima medición que puede discernir es: $\frac{5\text{ V}}{1024}$=4.882 mV. Sin embargo, en el pin "AREF" (*Analogue REFerence*) se puede aplicar un voltaje externo como referencia para realizar la conversión de las entradas analógicas: $\frac{\text{AREF}}{1024}$.

Para señales de entrada con componentes negativos y positivos se tiene que acondicionar por medio de instrumentación electrónica, por ejemplo, amplificadores operacionales de tal forma que tomando como referencia 5 V, si utiliza por convención 2.5 V para representar 0 V de la señal de entrada, entonces el voltaje comprendido entre 2.5 a 5 V es la parte positiva de la señal y de 0 a 2.5 representa la componente negativa de la señal.

Los canales analógicos pueden ser configurados por software como puertos digitales entrada/salida (numerados del pin 14 al pin 19).

Salidas PWM

En el conector pin-hembra, algunos puertos digitales pueden ser configurados por software como señales de salida PWM (*pulse width modulation*), específicamente los pins marcados con PWM∼ en la tarjeta Arduino UNO (pins: 3∼, 5∼, 6∼, 9∼, 10∼ y 11∼) proporcionan salidas PWM, es una señal cuadrada formada por pulsos de frecuencia constante (490 Hz o aproximadamente 2 mseg de periodo) y con amplitud de voltaje de 0 y 5 V; esta señal se aplica en control de hornos mediante optoacopladores produciendo un porcentaje de tiempo encendido el horno y otro porcentaje apagado. También se emplea en control la velocidad de un motor, variación de la frecuencia de un sonido emitido por un zumbador o bocina, o variar la intensidad de luz de un LED. La duración del pulso es variable, es decir el ancho del pulso puede ser modulado; cada pin PWM tiene una resolución de 8 bits, es decir 256 posibles valores, el cero corresponde a 0 V, con un valor de 128 significa que se genera al 50 % de energía activada con un pulso de un mseg en alto o 5 V y 0 V o en bajo durante un mseg; el valor 255 representa un pulso de máxima duración de 5 V. La señal de PWM de los pins 3 y 11 son controlados por el Timer1, pins 5 y 6 por el Timer2 y pins 9 y 10 por el Timer3.

Comunicación serial Rx/Tx: pins 0 y 1

Los puertos digitales pins 0 y 1 pueden ser configurados como funciones especializadas de transmisión serial Rx y Tx, respectivamente. El pin 0 (Rx) puede recibir y el pin 1 (Tx) puede transmitir datos serialmente (sin pasar por el formato de conversión USB-serie que realiza el chip Atmega16U2). Por diseño electrónico

los pins Rx y Tx se encuentran internamente conectados mediante resistencias de 1 KΩ al chip Atmega16U2, por lo que la información en el USB también se encuentra disponible en los pins 0 y 1.

Es importante aclarar que en la tarjeta electrónica Arduino UNO se encuentran un par de indicadores luminosos LEDs etiquetados con Rx y Tx y que se encienden únicamente cuando reciben información por medio de la comunicación USB a través del circuito integrado Atmega16U2; estos LEDs no se encienden cuando hay transmisión/recepción de datos en los pins 0 y 1.

Interrupciones INT0/INT1: pins 2 y 3

Mediante programación, los pins 2 y 3 se pueden configurar como interrupciones externas INT0/INT1.

Comunicación SPI: pins 10, 11, 12 y 13

Usando programación los pins 10, 11, 12 y 13 pueden ser configurados para realizar comunicación serial mediante el protocolo SPI (*serial peripheral interface*).

LED pin 13

El pin 13 del conector pins-hembra contiene un indicador luminoso LED (con su resistencia limitadora de corriente) identificado con la letra "L" en la tarjeta electrónica Arduino UNO, de forma que cuando el pin 13 funciona como puerto digital, si este puerto tiene un valor ALTO (HIGH), el LED encenderá y con un bajo (LOW), el LED se apagará.

Este LED funciona como un detector de señales sin la necesidad de conectar algún indicador adicional.

Comunicación I^2C/TWI: pins A4 y A5

Los pins A4 y A5 que normalmente se utilizan para conversión de señales analógicas

a digitales, mediante programación pueden ser configuradas para trabajar con el bus serial bajo el protocolo TWI (*two-wire* también conocido como I^2C *inter-integrated circuit*).

Los pins A4 y A5 están duplicados en los últimos pins del conector pins-hembra, justo después del pin AREF (sin etiquetar, debido a que ya no hay más espacio físico).

Pin RESET

Si el voltaje en el pin RESET es 0 V (bajo o LOW), el microcontrolador se reinicializará y se pondrá en marcha el *bootloader*, es decir de manera electrónica se puede controlar la interrupción de RESET. Ésta es una función adicional al botón de RESET incluido en la tarjeta Arduino UNO.

Pin IOREF

El pin IOREF es un pin que duplica la función del pin V_{IN}; puede utilizarse en las tarjetas de interface conectadas al modelo Arduino UNO para indicar el voltaje que trabajan los puertos digitales entrada/salida (en el caso Arduino UNO es 5 V).

Pin sin conexión

El pin que se encuentra al final del conector pins-hembra, después del pin IOREF (no cuenta con etiqueta) no tiene ninguna función y se considera sin conexión (para futuras mejoras en el modelo Arduino UNO).

Conector ICSP

El conector ICSP (*in circuit serial programming*) se utiliza para programar a los microcontroladores con arquitectura AVR, PICs y Parallax Propeller, ya que carecen de un *bootloader* previamente instalado, como puede ser el caso de esos dispositivos nuevos o recién salidos de fábrica.

Para programar un microcontrolador nuevo o sustituir el *bootloader* de alguna

tarjeta Arduino se necesita un programador ISP con su cable USB para la computadora y un cable para el conector ICSP hacia la tarjeta Arduino. Información sobre compatibilidad y modelos programadores se puede consultar en el archivo programmers.txt el cual fue descargado junto con el ambiente de programación Arduino; suponiendo por ejemplo que el usuario instaló el paquete Arduino en la siguiente trayectoria: c:\arduino\, entonces generalmente se ubica en la subcarpeta c:\arduino\hardware\arduino\programmers.txt.

Botón de RESET

El modelo Arduino UNO dispone de un botón para RESET (libre de rebotes), lo que permite el reinicio del sistema electrónico cuando se pulsa enviando una señal en bajo (LOW) al pin RESET del microcontrolador Atmega328P, al momento del arranque se activa la ejecución del *bootloader* permitiendo cargar un nuevo sketch en la memoria flash eliminando el sketch o programa anterior.

La tarjeta Arduino UNO contiene un sistema de auto-reset que se genera desde el ambiente de programación Arduino en la computadora permitiendo la activación del *bootloader* en la tarjeta cuando se oprime el icono para descargar código ➡.

3.3.2 Señales electrónicas del microcontrolador Atmega328P

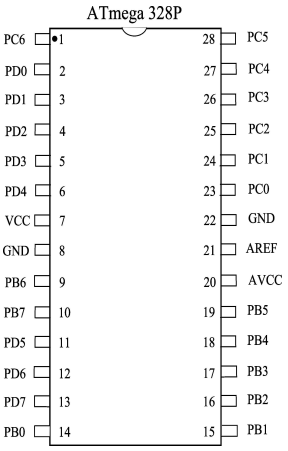


Figura 3.6 328P.

La figura 3.6 presenta la distribución de pins alrededor del tipo de encapsulado DIP del chip que tiene el circuito integrado del microcontrolador Atmega328P del modelo Arduino UNO, mientras que la tabla 3.9 muestra la descripción de señales. La alimentación eléctrica del microprocesador ATMega328P se denota por VCC y utiliza un voltaje de 5 V (pin 7 de la figura 3.6). Contiene dos pins (8 y 22) referidos a tierra GND. La alimentación se puede obtener conectando el conector *jack power* a una fuente externa utilizando un adaptador eléctrico AC/DC o una pila conectandola directamente a los pines-hembra marcados como “VIN” y “GND”. El pin 1 se identifica

por un pequeño círculo negro en la esquina superior del chip y la numeración de pins crece en el sentido contrario a las manecillas del reloj. AVCC representa el voltaje de referencia para el convertidor analógico/digital interno y AREF es la referencia analógica para dicho convertidor. El puerto B contiene 8 líneas bidireccionales (I/O) con resistencias internas *pull-up*, cada línea tiene 3 estados (entrada, salida y alta impedancia) DPI (*dual in-line package*).

Tabla 3.9 Señales del microcontrolador Atmega 328P.

Pin	Señal	Descripción	Pin	Señal	Descripción
1	PC6	PCINT14/Reset	28	PC5	ADC5/SCL/PCINT13
2	PD0	PCINT16/RXD	27	PC4	ADC4/SDA/PCINT12
3	PD1	PCINT17/TXD	26	PC3	ADC3/PCINT11
4	PD2	PCINT18/INT0	25	PC2	ADC2/PCINT10
5	PD3	PCINT19/OC2B/INT1	24	PC1	ADC1/PCINT9
6	PD4	PCINT20/XCK/T0	23	PC0	ADC0/PCINT8
7	VCC	Alimentación	22	GND	Tierra
8	GND	Tierra	21	AREF	Referencia analógica
9	PB6	PCINT6/XTAL1/TOSC1	20	AVCC	Alimentación
10	PB7	PCINT/XTAL2/TOSC2	19	PB5	SCK/PCINT5
11	PD5	PCINT21/OC0B/T1	18	PB4	MISO/PCINT4
12	PD6	PCINT22/OC0A/AIN0	17	PB3	MOSI/OC2A/PCINT3
13	PD7	PCINT23/AIN1	16	PB2	SS/OC1B/PCINT2
14	PB0	PCINT0/CLK0/ICP1	15	PB1	OC1A/PCINT1

En el modelo Arduino UNO, el *bootloader* se ejecuta en el primer segundo de cada reinicio (por generar una señal de reset, al encender la fuente de alimentación externa o mediante el cable USB); el gestor de arranque espera recibir el código de máquina desde el ambiente de programación Arduino, si no hay envío de información durante ese lapso de tiempo, termina la ejecución del *bootloader* transfiriendo al contador de programa (*program counter*) la ejecución de instrucciones del último programa que se descargó en la memoria flash.

Atmel utiliza en la programación de sus microcontroladores el protocolo STK500 para el diseño del *bootloader* o gestor de arranque, como el del kit de desarrollo del *starter kit* STK500 (ver el siguiente sitio Web):

http://www.atmel.com/tools/STK500.aspx

Ejemplos de protocolos de *bootloader* son los protocolos Arduino que utilizan las tarjetas Arduino UNO, Mini, Nano, Ehternet; wiring para los modelos Arduino Mega 2560 o el protocolo avr109 para los modelos Leonardo, Esplora, Arduino Robot Motor, Arduino Robot Control. Esta información se encuentra disponible en el archivo boards.txt (ver subcarpeta hardware/arduino de la carpeta donde tiene instalado el ambiente de programación Arduino). También, consultar el sitio Web: http://code.google.com/p/arduino/wiki/Plat

Gestor de arranque Optiboot

El programa Optiboot es un ejemplo de un programa gestor de arranque/cargador (*bootlaoder* o *firmware*) que funciona para el modelo Arduino UNO (Atmega328P), el cual logra velocidades de descarga de código de máquina en la memoria flash de 115 Kilobaudios, este programa es (*firmware* libre) que se puede obtener en el siguiente sitio Web: http://code.google.com/p/optiboot

Hay que tomar en cuenta que si tenemos un microcontrolador nuevo Atmel, por ejemplo el modelo Atmega328P, entonces este dispositivo no tendrá pregrabado un *bootloader*, en este caso se requiere de un sistema programador denominado ISP (*in system programmer*), el cual se debe conectar a la computadora y al microcontrolador (ver ambiente de programación Arduino, opción Herramientas, Programador y elegir una opción adecuada).

Importante

Cuando Usted descargó e instaló el paquete de cómputo del ambiente de programación Arduino, éste ya viene con respaldos (copias exactas) de los *bootloaders* de los microcontroladores Arduino; se pueden identificar con la extensión .hex (*Intel-hex-Format*). Por ejemplo, supóngase que la trayectoria de instalación es: c:\Arduino, entonces verifique el archivo AtmegaBOOT.hex se encuentra en la subcarpeta: c:\Arduino\hardware\arduino\bootloaders\atmega8\. Normalmente estos archivos no se utilizan, salvo que tengamos un microcontrolador nuevo o se requiera reponer porque sufrió algún daño el *bootloader* de la tarjeta, entonces podemos grabarlo en la memoria flash en el ambiente de programación Arduino, con el menú Herramientas, opción Programador y seleccionar el ISP más adecuado.

3.4 Modelos de tarjetas Arduino

ARDUINO contiene una gama amplia de tarjetas electrónicas con diferentes características tecnológicas y aplicaciones específicas en ingeniería y ciencias exactas. La selección de las tarjetas electrónicas depende de los parámetros del microcontrolador, puertos entrada/salida, ADCs (convertidores analógico/digital), DACs (convertidores digital/analógico), velocidad de comunicación serie, periféricos, señales disponibles para interface, protocolos de comunicación inalámbrica (WiFi, bluetooth) y ethernet.

En función de las características tecnológicas y del tipo de aplicación, a los modelos Arduino los podemos clasificar como:

- De propósito general para instrumentación y automatización de procesos físicos e ingeniería robótica y mecatrónica se utilizan los modelos: Arduino UNO, Leonardo, Due, Mega, Mega 2560, Mega ADK, Mini, Pro, Pro mini, Micro, Nano, FIO y Galileo.
- Corresponden a los modelos Esplora y Arduino Robot la integración de sensores para monitorear, interaccionar y procesar variables ambientales, así como aplicaciones en robótica móvil.
- Los modelos LilyPad (LilyPad Arduino, USB, Simple y SimpleSnap) se especializan en aplicaciones de electrónica textil (**e-textil**) y prendas de vestir inteligentes dinámicas e interactivas fabricados con fibras textiles de materiales especiales de hilos conductores e integradas con sensores, electrónica digital, microcontroladores y componentes eléctricos.
- Conexión a WiFi, bluetooth y ethernet se puede realizar por medio de las tarjetas electrónicas Yún, BT y ethernet, respectivamente que no sólo permiten procesamiento de señales digitales y acoplamiento de sensores, también contienen la integración de módulos para enlazar tecnología inalámbrica y protocolos de redes.

A continuación se describen los modelos de tarjetas del sistema Arduino.

3.4.1 Modelos de propósito general para ingeniería

Arduino UNO

El modelo Arduino UNO es una tarjeta electrónica que se basa en el microcontrolador Atmega28, tiene 14 puertos digitales de entrada/salida, de los cuales 6 puertos pueden ser utilizados como salidas PWM; 6 canales analógicos de 10 bits cada uno, frecuencia de operación de 16 MHz, conexión USB, conectores *power jack* y ICSP, botón de reset. La alimentación de la tarjeta puede ser mediante el cable USB o por medio de una fuente externa de 5 V. La figura 3.7 muestra el modelo Arduino UNO y en la tabla 3.11 se presentan sus principales características.

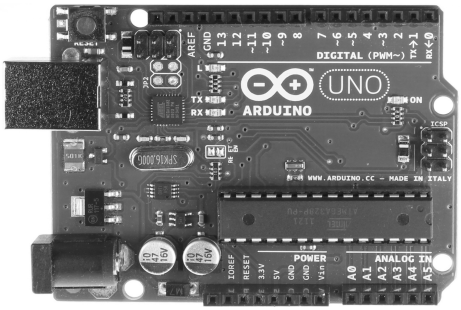


Figura 3.7 Arduino UNO.

Arduino Leonardo

El modelo Arduino Leonardo es una tarjeta electrónica que utiliza el microcontrolador Atmega32u4, con 20 puertos digitales entrada/salida, de los cuales 7 puertos pueden ser configurados como salidas PWM y 12 pins como entradas analógicas; la frecuencia de operación es de 16 MHz, botón para reset; la alimentación de la tarjeta puede ser por medio del conector micro USB conector o por una fuente de alimentación externa (usando el conector *power jack*); también contiene un conector tipo ICSP. Este modelo tiene incorporado un módulo propio de comunicación USB, eliminado la necesidad de un procesador secundario, lo que permite ser visto por una

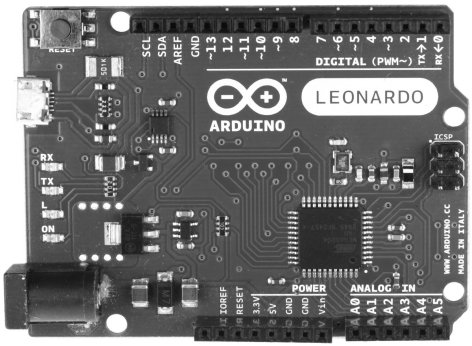


Figura 3.8 Arduino Leonardo.

computadora como un *mouse* o teclado. La figura 3.8 muestra la tarjeta electrónica del modelo Arduino Leonardo y en la tabla 3.11 se resumen sus principales características tecnológicas.

3.4.2 Arduino Due

El modelo Arduino Due es una tarjeta electrónica que utiliza el microcontrolador Atmel SAM3X8E ARM Cortex-M3 de 32 bits, tiene 54 puertos digitales de entrada/salida, de los cuales 12 puertos pueden ser utilizados como salidas PWM, 12 entradas analógicas de 10 bits, 4 UARTs, frecuencia de operación de 84 MHz; también tiene 2 convertidores digital/analógico, 2 TWI, un conector SPI y JTAG, cuenta con botón de reset y otro más para borrado. De manera diferente a otros modelos, Arduino Due sólo puede correr a 3.3 V siendo esta magnitud su máximo voltaje, ya que excederlo puede dañar a los componentes electrónicos. La alimentación de la tarjeta se puede realizar mediante cable micro USB o alimentación externa usando el conector *power jack*. La figura 3.9 muestra el modelo Arduino Due y en la tabla 3.11 se detalla las principales características técnicas.

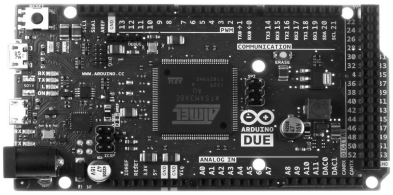


Figura 3.9 Arduino Due.

3.4.3 Arduino Mega

El modelo Arduino Mega emplea el microcontrolador Atmega1280 contiene 54 puertos digitales (I/O), de los cuales 14 puertos pueden ser configurados como salidas para señal PWM, 16 entradas analógicas con 10 bits de resolución cada una, 4 dispositivos de comunicación serial UARTs, oscilador de cristal con frecuencia de 16 MHz, conexión a la computadora vía USB, voltaje de alimentación VCC de 5V, (puede usar alimentación USB o con una fuente externa utilizando el conector *jack power*); la fuente de alimentación es seleccionada

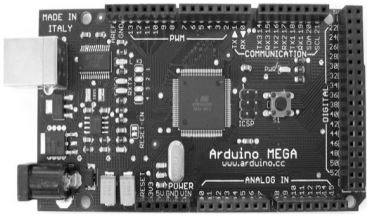


Figura 3.10 Arduino Mega.

automáticamente; también cuenta con conector ICSP y botón de reset. La figura 3.10 muestra el modelo Arduino Mega y en la tabla 3.12 se concentran las características claves de este modelo.

3.4.4 Arduino Mega 2560

El modelo Arduino Mega2560 es una versión actualizada y mejorada de la versión Arduino Mega (ver figura 3.11). Usa el microcontrolador Atmega2560, tiene 54 pins de puertos digitales entrada/salida (I/O), de los cuales 15 puertos pueden ser usados como salidas PWM, 16 entradas analógicas, cada uno con 10 bits de resolución (1024 valores diferentes), 4 UARTs, oscilador de 16 MHz, conexión a la computadora USB, memoria flash de 256 Kbytes, de los cuales 8 Kbytes están reservados para el gestor de arranque (*bootloader*), memoria para datos SRAM de 8 Kbytes y memoria EEPROM de 4 Kbytes, botón de reset y conector ICSP. El voltaje de alimentación VCC es de 5V, se puede suministrar vía USB o con una fuente externa (*jack power*). En la tabla 3.12 muestra se detallan sus características.

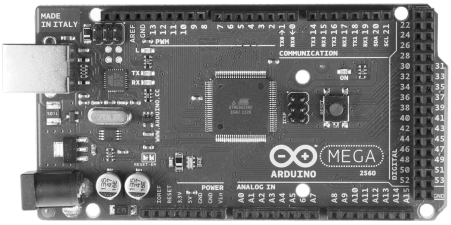


Figura 3.11 Arduino Mega 2560.

3.4.5 Arduino Mega ADK

Arduino Mega ADK emplea el microcontrolador Atmega2560, cuenta con una interface USB para conectarse con teléfonos que tienen el sistema Android (utilizando el circuito integrado MAX3421e). Además contiene 54 puertos digitales entrada/salida, de los cuales 15 puertos pueden ser usados como salidas PWM, 16 entradas analógicas de 10 bits cada

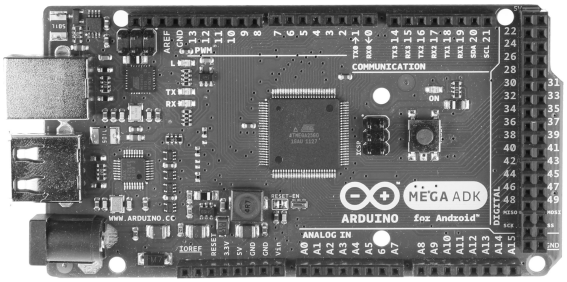


Figura 3.12 Arduino Mega ADK.

una, 4 dispositivos UARTs, frecuencia de operación a 16 MHz, conexión USB, conectores *power jack*, ICSP y botón de reset. El modelo Arduino ADK se basa en la tarjeta Mega 2560. La figura 3.12 muestra el modelo Arduino Mega ADK y en la tabla 3.12 se muestran sus características técnicas.

3.4.6 Arduino Micro

Arduino micro utiliza el microcontrolador Atmega32u4 desarrollado de manera conjunta entre Atmel y Adafruit. Tiene 20 puertos digitales de entrada/salida, de los cuales 7 pueden ser utilizados como salidas PWM y 12 como entradas analógicas; la frecuencia de operación es de 16 MHz, conexión micro USB, botón de reset, conector ICSP. La figura 3.13 muestra el modelo Arduino Micro y la tabla 3.13 describe las características técnicas.

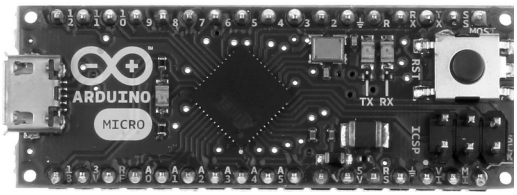


Figura 3.13 Arduino Micro.

Arduino Pro

Este modelo se basa en los microcontroladores Atmega168 o Atmega328 y viene en dos versiones: 3.3 V proporcionado por baterías, el cual corre a 8 MHz o 5 V proporcionado por fuente de alimentación externa y puede correr a 16 MHz. El modelo Arduino Pro tiene 14 puertos digitales de entrada/salida, de los cuales 6 puertos pueden ser configurables como salidas PWM; 6 entradas analógicas de 10 bits por canal, un conector tipo *jack power* e interruptor o switch para la alimentación por medio de pilas o baterías, botón de reset, conector ICSP y pins para interface. Adicionalmente, un conector tipo pin puede ser soldado

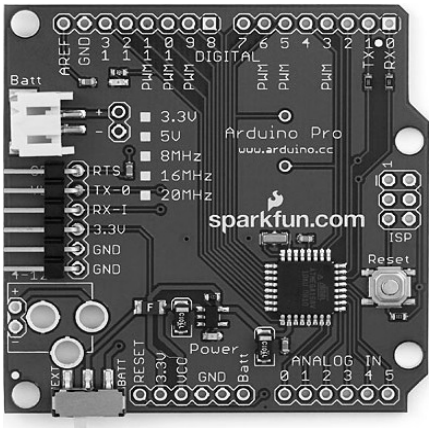


Figura 3.14 Arduino Pro.

a la tarjeta para cable FTDI y habilitar comunicación USB y proveer alimentación a la tarjeta, la cual viene sin conectores, permitiendo soldar diversos tipos de conectores o directamente cables. Este modelo fue manufacturado por Sparkfun Electronics. La figura 3.14 describe el modelo Arduino Pro y en tabla 3.13 describe sus características claves.

Arduino Mini

La figura 3.15 muestra el modelo Arduino Mini, el cual utiliza el microcontrolador Atmega168, contiene 14 puertos digitales de entrada/salida, 6 de estos puertos pueden ser configurados como salidas PWM, 8 canales analógicos de 10 bits y un cristal oscilador de 16 MHz. Esta tarjeta puede ser programada usando un adaptador serial USB o RS232. La tabla 3.14 presenta los detalles técnicos de este modelo.

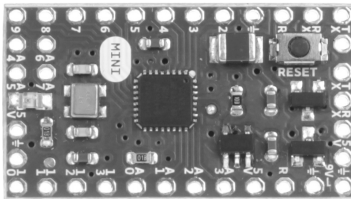


Figura 3.15 Arduino Mini.

Arduino Pro Mini

El modelo Arduino Pro Mini utiliza el microcontrolador Atmega168 (ver figura 3.16); esta tarjeta electrónica cuenta con 14 puertos entrada/salida, de los cuales 6 puertos pueden ser utilizados como salida PWM; 8 entradas analógicas de 10 bits por canal, botón de reset; esta tarjeta viene sin el montaje de conectores, por ejemplo para comunicación USB (usando cable FTDI) y para proveer alimentación a la tarjeta se requiere soldar un conector con 6 pins. También tiene varios hoyos o contactos disponibles en la placa electrónica para soldar cables o pins.

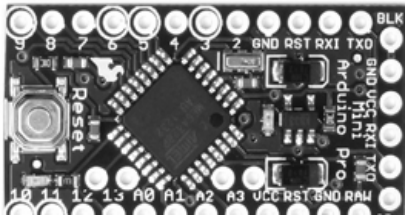


Figura 3.16 Arduino Pro Mini.

Este modelo presenta dos versiones: una de ellas corresponde a Pro Mini que se alimenta a 3.3 V y corre a 8 MHz y la otra versión utiliza 5 V de voltaje de alimentación y opera a 16 MHz. La tabla 3.13 concentra sus características técnicas.

Arduino Nano

Arduino Nano es un modelo que presenta dos tipos de versiones dependiendo del tipo de microcontrolador a usar, el cual puede ser: Atmega168 (utiliza 16 Kb de memoria flash) o Atmega328 (con 32 Kb de memoria flash); ambas versiones cuentan con auto-reset y trabajan con mini USB cable en lugar del conector estándar; además tiene 8 entradas analógicas de 10 bits por canal, memoria EEPROM de 512 bytes (Atmega168) y 1 Kb (Atmega328); en ambas versiones destinan 2 Kb de la memoria flash para almacenar el *bootloader*. La memoria de datos SRAM es de 1 Kb para Atmega168 o 2 Kb para la versión Atmega328. Este modelo es manufacturado por Gravitechno y no cuenta con conector de alimentación (*power jack*), la figura 3.17 muestra el modelo Arduino Nano y en la tabla 3.14 se presentan las características técnicas.

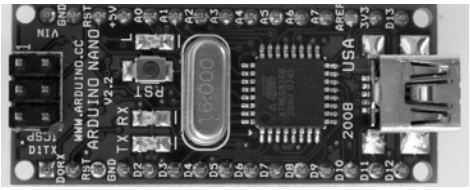


Figura 3.17 Arduino Nano.

Arduino FIO

Arduino FIO es una tarjeta electrónica para aplicaciones inalámbricas (*wireless applications*) que se basa en el microcontrolador Atmega328P, la velocidad de trabajo es 8 MHz empleando 3.3 V; tiene 14 puertos digitales entrada/salida, de los cuales 6 puertos pueden ser programados como salidas PWM; 8 entradas analógicas de 10 bits cada una, botón para reset, socket para XBee y un conjunto de hoyos para soldar terminales, cables o conectores. Además contiene conexiones para alimentación por baterías de lithium e incluye cargador vía USB.

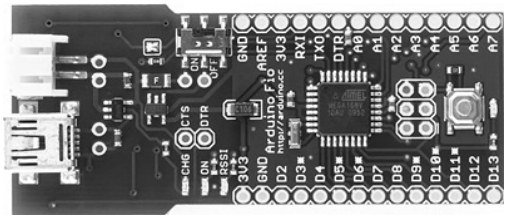


Figura 3.18 Arduino FIO.

Este modelo es manufacturado por SparkFun Electronics. El usuario puede descargar sketchs usando cable FTDI. En la figura 3.18 se muestra el modelo Arduino FIO y en la tabla 3.14 se concentran sus principales características técnicas.

3.4.7 Modelo Intel Galileo

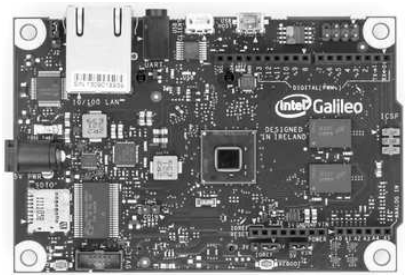


Figura 3.19 Tarjeta Galileo.

El modelo Galileo contiene un microcontrolador Intel Quark SoC X1000, es un sistema Intel-Pentium de 32 bits y 400 MHz. Para el sistema Arduino, este modelo representa el primer diseño con arquitectura Intel, compatible con las interfaces shields del modelo UNO R3 y también contiene la misma localización de pins en los conectores. Los puertos digitales se encuentran sobre los pins 0 al 13, tiene 6 entradas analógicas, conectores de alimentación, ICSP, puertos serial UART trabajan en los pins digitales 0 y 1. La tarjeta Galileo acepta shields que operan con 3.3 V o 5 V. Sin embargo, el voltaje de operación de Galileo es de 3.3 V, sobre el circuito impreso de la tarjeta, existe un puente de configuración (jumper) que permite habilitar 5 V en los puertos digitales, también por medio de ese jumper se puede regresar al modo de 3.3 V para definir el nivel de operación de los puertos digitales.

Además del hardware compatible de la tarjeta Galileo, también tiene puertos, capacidades y características de la computadoras industriales, tales como un mini slot PCI express, un puerto Ethernet de 100 Mb, un slot para Micro-SD, puerto serial RS-232, puerto USB host, puerto cliente USB y memoria RAM flash de 8 Mbytes. La tabla 3.10 muestra sus características técnicas.

La tarjeta Galileo requiere la versión IDE Arduino 1.5.3-Intel.10.4, la cual se puede descargar gratuitamente de cualquiera de los siguientes enlaces:

https://communities.intel.com/docs/DOC-22226

http://arduino.cc/en/Main/Software

Tabla 3.10 Modelo de propósito general: Intel Galileo.

Descripción	Características
Microcontrolador	Intel Quark SoC X1000 Pentium de 32 bits, frecuencia de operación: 400 MHz.
Voltaje de alimentación	Voltaje nominal de operación: 3.3 V. Utilizando el jumper instalado en el circuito impreso de la tarjeta, se pueden configurar los puertos I/O para trabajar con 5 V.
Shields	Acepta shields compatibles con la tarjeta Arduino UNO (también pueden ser configurados a trabajar con 3.3 V o 5 V).
Puertos digitales I/O	Contiene 14 puertos digitales I/O, de los cuales 6 puertos pueden ser programados como salidas para modulación de ancho de pulso (PWM). Los puertos digitales I/O aceptan las funciones: pinMode(), digitalWrite() y digitalRead(). Cada pin digital puede proporcionar 10 mA y recibir un máximo de 25 mA. Los pins digitales tienen resistencias de pull-up (desconectadas por default) de 5.6 KΩ a10 KΩ.
Entradas analógicas	6 entradas analógicas (etiquetadas A0-A5) utilizando el convertidor analógico digital AD7298, con 12 bits de resolución (4096 valores de cuantización), por default las mediciones son referidas de tierra a 5 V.
Bus TWI	Contiene un bus TWI, 12 C por medio de los pins A4 (SDA) y A5 (SCL). Para soportar la comunicación TWI se emplea la librería Wire.
UART	Puertos seriales UART en los pins 0 para Rx y 1 para Tx
ICSP	Contiene un circuito (6 pins) para programación serial con un concetor tipo header (ICSP) para conectar shields; para la comunicación SPI se utiliza la librería SPI.
V_{in}	Cuando se alimenta a la tarjeta Galileo por medio de una fuente externa usando el conector tipo jack, se puede acceder al pin V_{in} o también se puede aplicar voltaje a V_{in} por medio de una fuente regulada de voltaje, en este caso, debe tener cuidado en no exceder 5 V, debido a que puede quemar o dañar a la tarjeta.
5 V	Este pin proporciona 5 V de salida, que puede venir de una fuente externa de alimentación o del conector USB; la máxima corriente que puede suministrar son 800 mA.
3.3 V	Este pin proporciona 3.3 V de salida regulado por un circuito integrado que se encuentra en el circuito impreso de la tarjeta; la máxima corriente que puede suministrar son 800 mA.

Tabla 3.11 Modelo de tarjetas de propósito general.

Características	UNO	Leonardo	Arduino Due
Microcontrolador	Atmega328 (8bits)	Atmega32u4 (8 bits)	AT91SAM3X8E (32 bits)
Voltaje de alimentación	5 V	5 V	3.3 V
Voltaje de entrada	7 a 12 V	7 a 12 V	7 a 12 V
Límites de voltaje de entrada	6 a 20 V	6 a 20 V	6 a 16 V
Puertos digitales (I/O)	14 pins, de los cuales 14 se pueden configurar como salidas PWM	20 pins I/O, de los cuales 7 pins pueden ser salidas PWM y 12 como entradas analógicas	54 pins I/O, de los cuales 12 pins se pueden configurar como salidas PWM
Entradas analógicas	6	12, configurables de los puertos digitales	12, configurables de los puertos digitales
Salidas analógicas DACs	No	No	2
Corriente DC por pin (I/O)	40 mA	40 mA	Corriente total en todos los pins: 130 mA
Corriente DC por pin 3.3 V	50 mA	50 mA	800 mA
Memoria flash	32 Kb (de los cuales 0.5 Kb son para *bootloader*)	32 Kb (de los cuales 4 Kb son para *bootloader*)	512 Kb disponibles para aplicaciones del usuario
SRAM	2 Kb	2.5 Kb	96 Kb
EEPROM	1 Kb	1 Kb	
Frecuencia de operación	16 MHz	16 MHz	84 MHz
Contiene integrado un LED en el pin 13 (puerto digital configurable como salida) con resistencia limitadora de corriente (enciende en "1=HIGH", se apaga con "0=LOW")	Si	Si	Si

Tabla 3.12 Modelos Mega de propósito general.

Características	Mega ADK	Mega	Mega2560
Microcontrolador	Atmega2560 8 bits	Atmega1280 8 bits	Atmega2560 8 bits
Voltaje de alimentación	5 V	5 V	5 V
Voltaje de entrada	7 a 12 V	7 a 12 V	7 a 12 V
Límites de voltaje de entrada	6 a 20 V	6 a 20 V	6 a 20 V
Puertos digitales (I/O)	54 pins, de los cuales 15 se pueden configurar como salidas PWM	54 pins I/O, de los cuales 15 pins pueden ser salidas PWM	54 pins I/O, de los cuales 15 pins se pueden configurar como salidas PWM
Entradas analógicas	16	16	16
Corriente DC por pin (I/O)	40 mA	40 mA	40 mA
Corriente DC por pin 3.3 V	50 mA	50 mA	50 mA
Memoria flash	256 Kb (de los cuales 8 Kb son para *bootloader*)	128 Kb (de los cuales 4 Kb son para *bootloader*)	256 Kb (de los cuales 8 Kb son para *bootloader*)
SRAM	8 Kb	8 KB	8 Kb
EEPROM	4 Kb	4 KB	4 Kb
Frecuencia de operación	16 Mhz	16 Mhz	16 Mhz
Contiene integrado un LED en el pin 13 (puerto digital configurable como salida) con resistencia limitadora de corriente (enciende en “1=HIGH”, se apaga con “0=LOW”)	Si	Si	Si

Tabla 3.13 Modelos Micro/Pro/Pro-mini de propósito general.

Características	Micro	Pro	Pro mini
Microcontrolador	Atmega32u4 8 bits	Atmega168/Atmega328 8/8 bits	Atmega168 8 bits
Voltaje de alimentación	5 V	3.3/5 V	3.3/5 V
Voltaje de entrada	7 a 12 V	3.35 a 12 V/5 a 12 V	3.35 a 12 V/5 a 12 V
Límites de voltaje de entrada	6 a 20 V	3.35 a 12 V/5 a 12 V	3.35 a 12 V/5 a 12 V
Puertos digitales (I/O)	20 pins, de los cuales 7 se pueden configurar como salidas PWM y 12 como entradas analógicas	14 pins I/O, de los cuales 6 pins pueden ser salidas PWM	14 pins I/O, de los cuales 6 pins pueden ser salidas PWM
Entradas analógicas	12	6	8
Corriente DC por pin (I/O)	40 mA	40 mA	40 mA
Corriente DC por pin 3.3 V	50 mA		
Memoria flash	32 Kb (de los cuales 4 Kb son para *bootloader*)	16/32 Kb (en ambas versiones, 4 Kb son para *bootloader*)	16 Kb (de los cuales 2 Kb son para *bootloader*)
SRAM	2.5 Kb	1/2 Kb	1 Kb
EEPROM	1 Kb	0.5/1 Kb	512 bytes
Frecuencia de operación	16 MHz	8/16 MHz	8/16 MHz
Contiene integrado un LED en el pin 13 (puerto digital configurable como salida) con resistencia limitadora de corriente (enciende en “1=HIGH”, se apaga con “0=LOW”)	Si	Si	Si

Tabla 3.14 Modelos Mini/Nano/FIO de propósito general.

Características	Mini	Nano	FIO
Microcontrolador	Atmega328 8 bits	Atmega168/Atmega328 8/8 bits	Atmega328 8 bits
Voltaje de alimentación	5 V	5 V	3.3 V
Voltaje de entrada	7 a 9 V	7 a 12 V	3.35 a 12 V
Límites de voltaje de entrada	6 a 20 V	6 a 20 V	Voltaje de entrada para carga: 3.7 a 7 V
Puertos digitales (I/O)	14 pins, de los cuales 6 se pueden configurar como salidas PWM	14 pins I/O, de los cuales 6 pins pueden ser salidas PWM	14 pins I/O, de los cuales 6 pins pueden ser salidas PWM
Entradas analógicas	8, cuatro de las cuales se encuentran disponibles en los pins.	8	8
Corriente DC por pin (I/O)	40 mA	40 mA	40 mA
Memoria flash	32 Kb (de los cuales 2 Kb son para *bootloader*)	16/32 Kb (en ambas versiones, 2 Kb son para *bootloader*)	32 Kb (de los cuales 2 Kb son para *bootloader*)
SRAM	2 Kb	1/2 Kb	2 Kb
EEPROM	1 Kb	0.5/1 Kb	1 Kb
Frecuencia de operación	16 MHz	16 MHz	8 MHz
Contiene integrado un LED en el pin 13 (puerto digital configurable como salida) con resistencia limitadora de corriente (enciende en "1=HIGH", se apaga con "0=LOW")	Si	Si	Si

3.4.8 Modelos de integración de sensores y robótica móvil

Existen determinados modelos Arduino que tienen integrados diversos componentes y sensores para que el usuario puede interactuar con su entorno como son los casos de las tarjetas electrónicas: Esplora y Arduino Robot.

El modelo Esplora incluye sensores de temperatura, acelerómetro, potenciómetro deslizable, sensor de luz, interruptores, micrófono, palanca de juegos (*joystick*), Led RGB, buzzer, etc., así como capacidad para expandirse ya que cuenta con un conjunto de conectores para colocar un display o pantalla LCD táctil y realizar interface con el mundo exterior.

El modelo Arduino Robot dedicado a las aplicaciones de robótica móvil integra en una placa electrónica a dos tipos de microcontroladores, uno de ellos dedicado al control de los motores y el otro para desarrollar algoritmos de control. Cuenta con pantalla LCD táctil, 2 llantas para desplazamiento, bocinas, indicadores de luz, compás digital, sensores de proximidad y ultrasonido. Extensas librerías especializadas (Robot.digitalRead(), Robot.digitalWrite(), etc.) permiten realizar y desarrollar aplicaciones desde el entorno de programación Arduino.

Esplora

Arduino Esplora (ver figura 3.20) es una tarjeta electrónica que utiliza el microcontrolador Atmega32U4, oscilador con cristal de 16 Mhz, conexión micro-USB, botón de reset, 4 LED's indicadores, voltaje de alimentación de 5 V, memoria flash de 32 Kb, SRAM 2.5 Kb, EEPROM 1 Kb. Esta tarjeta tiene características técnicas parecidas al modelo Arduino Leonardo. Contiene un conjunto importante de sensores para interacción: sensores de temperatura, LED RGB, joystick, potenciómetro lineal deslizable, acelerómetro, micrófono de ambiente, sensor de luz, socket para pantalla SCD TFT (*touch*), 4 botones permanentemente abiertos (*push button*). La tabla 3.15 presenta los detalles técnicos del modelo Esplora.

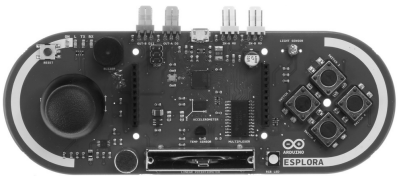


Figura 3.20 Arduino Esplora.

Arduino Robot

La figura 3.21 presenta al modelo Arduino Robot, el cual es un prototipo de robot móvil que se desplaza por medio de llantas. Este modelo contiene dos tarjetas con sus respectivos microcontroladores (ambas tarjetas utilizan el microcontrolador Atmega32u4). Una tarjeta es de control de motores y la otra tarjeta es para lectura de sensores y algoritmos de control. El robot móvil tiene varios de sus pins configurados como lectura de sensores y envío de información a los actuadores. Cada una de las tarjetas (motores y control) es programable desde el ambiente de programación Arduino con comunicación USB. Las características técnicas de Arduino Robot se detallan en la tabla 3.15.

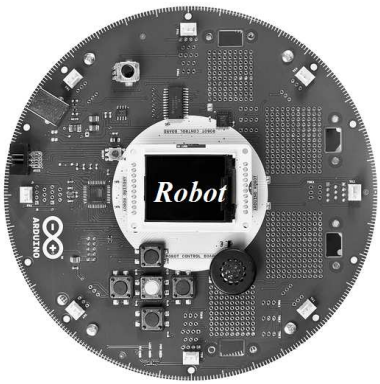


Figura 3.21 Arduino Robot.

Tabla 3.15 Modelos Esplora y Arduino Robot.

Características	Esplora	Arduino Robot
Microcontrolador	Atmega32u4 (8 bits)	Atmega32u4 (8 bits)
Voltaje de alimentación	5 V	5 V usando cable plano
Voltaje de entrada	7 a 9 V	7 a 12 V
Límites de voltaje de entrada	6 a 20 V	6 a 20 V
Puertos digitales (I/O)	14 pins, de los cuales 6 se pueden configurar como salidas PWM	5; 6 Canales PWM; 4 canales de entrada analógica.
Memoria flash	32 Kb (de los cuales 4 Kb son reservados para *bootloader*)	32 Kb (de los cuales 4 Kb son para *bootloader*)
SRAM	2.5 Kb	3.5 Kb
EEPROM	1 Kb	Interna 1 Kb; externa 512 bytes
Frecuencia de operación	16 MHz	16 MHz

3.4.9 Modelos para comunicación WiFi, ethernet y bluetooth

Arduino Yún

El microcontrolador Atmega32U4 es la plataforma electrónica del modelo Arduino Yún incorpora comunicación WiFi y aplicaciones en Linux. Tiene 14 puertos digitales entrada/salida, de los cuales 7 puertos pueden ser utilizados como salidas PWM, 6 entradas analógicas de 10 bits cada una; frecuencia de operación a 16 MHz. El voltaje de alimentación es de 5 V, el cual puede ser proporcionado por cable micro USB.

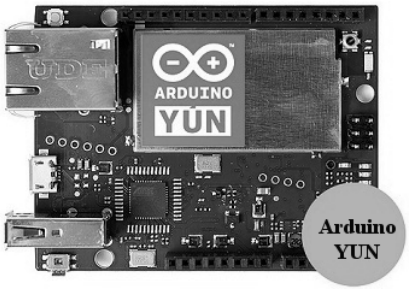


Figura 3.22 Arduino Yún.

La figura 3.22 muestra el modelo Arduino Yún y en la tabla 3.16 se describen las principales prestaciones técnicas con las que cuenta.

Arduino Ethernet

El modelo Arduino Ethernet es una tarjeta electrónica que se basa en el microcontrolador Atmega328, contiene 14 puertos digitales entrada/salida, de los cuales 4 son usados como salidas PWM; 6 entradas analógicas de 10 bits, frecuencia de operación a 16 Mhz, conexión RJ45, conector *power jack* para alimentación, conector ICSP, botón de reset. Los pins 10, 11, 12 y 13 se encuentran reservados para interface con ethernet. También, en forma opcional un sistema de alimentación sobre el módulo Ethernet se puede agregar a la tarjeta. Este modelo no contiene un dispositivo de conversión USB a serial, en su lugar contiene interface Wiznet ethernet usando el pin 10; en la tarjeta existe un lector de memoria microSD (se programa con la librería SD) para almacenar archivos de la red (pin 4). El pin 6 para programación serial es compatible con el

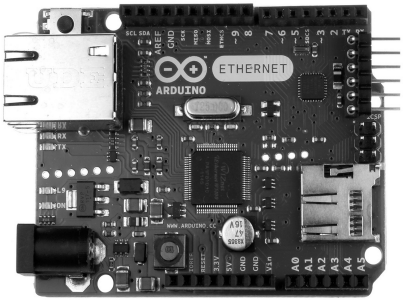


Figura 3.23 Arduino Ethernet.

adaptador serial USB y también con los cables FTDI USB. La figura 3.23 muestra el modelo Arduino Ethernet y en la tabla 3.16 se describen sus características técnicas.

Arduino BT

El modelo Arduino BT emplea el microcontrolador Atmega328 y el módulo de bluetooth Bluegiga WT11. Esta tarjeta tiene amplias aplicaciones de comunicación inalámbrica usando bluetooth. Tiene 14 puertos digitales entrada/salida, de los cuales 6 puertos pueden ser usados como salidas PWM y uno como salida para generar pulso de reset al módulo WT11; 6 entradas analógicas de 10 bits cada una, frecuencia de trabajo a 16 MHz, terminales SCREW para alimentación, conector ICSP y botón de reset. La tarjeta puede ser programada usando la comunicación bluetooth.

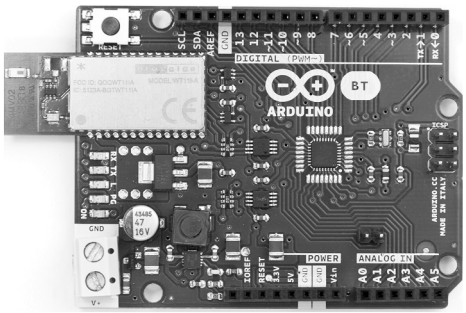


Figura 3.24 Arduino BT.

La figura 3.24 muestra el modelo Arduino BT para comunicación inalámbrica bluettoth y en la tabla 3.16 se presentan sus principales características técnicas.

Aplicaciones de tarjetas Arduino

En los capítulos 4 al 14, el lector encontrará varios ejemplos en lenguaje C debidamente documentados con aplicaciones de los modelos Arduino en: adquisición de datos, sistemas disctreos, control digital, control de motores a pasos, procesos de temperatura (hornos), sistemas dinámicos lineales y no lineales.

Modelo Arduino Zero

El sistema Arduino renueva y perfecciona continuamente los modelos de tarjetas; actualmente se encuentra en proceso el modelo **Arduino zero**, pero ahora con 32 bits, utilizando el microcontrolador Atmel SAMD21 MCU (32 bits ARM Cortex). En cuanto se encuentre disponible este nuevo modelo, se actualizará el material Web con sus aplicaciones potenciales.

Tabla 3.16 Modelos para WiFi, ethernet y bluetooth.

Características	Yún (WiFi)	Ethernet	BT (bluetooth)
Microcontrolador	Atmega32u4 8 bits	Atmega328 8 bits	Atmega328 8 bits
Voltaje de alimentación	5 V	5 V	5 V
Voltaje de entrada	5 V via micro USB o PoE 802.3af	7 a 12 V	2.5 a 12 V
Límites de voltaje de entrada	6 a 20 V	6 a 20 V (36 a 57 V para PoE)	
Puertos digitales (I/O)	14 pins, de los cuales 7 se pueden configurar como salidas PWM y 6 canales analógicos utilizando los pin de puertos digitales	14; de los cuales 4 canales PWM. Los pins 10 al 13 están reservados para SPI. 4 pins para tarjeta SD y 2 para interrupción W5100	14 pins, de los cuales 6 son salidas PWM
Entradas analógicas	6 (con 6 entradas analógicas adicionales multiplexeadas con 6 puertos digitales)	6	6
Corriente DC por puerto digital (pin I/O)		40 mA	40 mA
Corriente DC del pin: 3.3 V/5 V	50 mA para el pin 3.3 V	50 mA para el pin 3.3 V	500 mA/1000 mA; fuente de alimentación a 1.5 A
Memoria flash	32 Kb (de los cuales 4 Kb son reservados para *bootloader*)	32 Kb (de los cuales 0.5 Kb son para *bootloader*)	32 Kb de los cuales 2 Kb son reservados para el *bootloader*
SRAM	2.5 Kb	2 Kb	2 Kb
EEPROM	1 Kb	1 Kb	1 Kb
Frecuencia de operación	16 MHz	16 MHz	16 MHz
Módulo integrado	MicroSD Poe compatible 802.3af	W5100 TCP/IP	BT: 2.1 WT11i-A-AI4

3.4.10 Modelos LilyPad Arduino

LilyPad es un concepto diseñado por Leah Buechley que relaciona la integración de componentes electrónicas mediante fibra textil (con propiedades parecidas a los semiconductores) para dar origen a la ropa inteligente. El concepto comercial fue diseñado conjuntamente por Leah Buechley y Sparkfun Electrónica, se refiere al conjunto de componentes y elementos electrónicos que utilizan fibras o tejido textil (**e-textil**) en prendas de vestir. Con el empleo de materiales de fibra orgánica y textil lavable se puede construir ropa inteligente, dinámicas e interactiva tejida o cocida con hilos conductores para llevar información de variables ambientales, luminosidad, humedad, temperatura a un microcontrolador para su análisis y procesamiento.

Las prendas de vestir pueden incorporar sensores de algodón para detectar los movimientos anatómicos y fisiológicos del cuerpo e incorporar esta información a un modelo dinámico y producir efectos interactivos en la ropa mediante una red de LEDs, motores vibradores y altavoces.

Las áreas que estudian y analizan la clase de materiales de fibra textil para fabricar ropa inteligente (prendas de vestir electrónicas) se les denomina **e-textil** y **fibertronics**. El sistema Arduino contempla 4 tipos de modelos LilyPad con las siguientes características:

LilyPad Arduino

El modelo LilyPad Arduino es una tarjeta electrónica diseñada para prendas y tela con hilo textil conductor (*e-textiles*). Utiliza el microcontrolador Atmega168V (versión de baja potencia del microcontrolador Atmega168) y puede ser instalada sobre tela (cocida con hilo conductor). El modelo LilyPad Arduino fue diseñado y desarrollado por Leah Buechley y SparkFun Electronics. En la figura 3.25 se muestra dicho modelo y en la tabla 3.17 sus principales características técnicas.

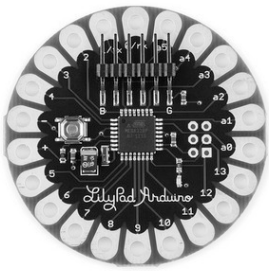


Figura 3.25 LilyPad Arduino.

LilyPad Arduino USB

El modelo LilyPad Arduino USB utiliza el microcontrolador Atmega32u4, su montaje es sobre una placa con geometría circular, contiene 9 puertos digitales de entrada/salida, de los cuales 4 puertos pueden ser configurados como salidas PWM y 4 como canales analógicos de 10 bits cada uno; trabaja a una frecuencia de 8 MHz, incluye botón de reset. Los conectores que tiene son del tipo micro USB, JST para baterías LiPo de 3.7 V. Este tipo de modelo de tarjeta electrónica también tiene amplias aplicaciones en interiores para la automatización de casas habitación y edificios inteligentes (domótica).

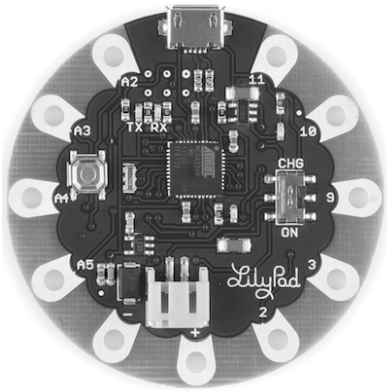


Figura 3.26 LilyPad Arduino USB.

La figura 3.26 muestra el modelo LilyPad Arduino USB y en la tabla 3.17 se describen sus principales características técnicas.

LilyPad Arduino Simple

LilyPad Arduino Simple es una tarjeta electrónica que se basa en el microcontrolador Atmega328 con aplicaciones en la industria textil de tela o ropa. Puede ser cosida en la tela e incorporar conexiones de sensores y actuadores con hilo conductivo. Tiene 9 puertos digitales entrada/salida, un conector tipo JST y circuito cargador para batería de Lithium. Este modelo fue diseñado y desarrollado por Leah Buechley y SparkFun Electronics. La figura 3.27 muestra a LilyPad Arduino Simple y la tabla 3.17 tiene sus características técnicas.

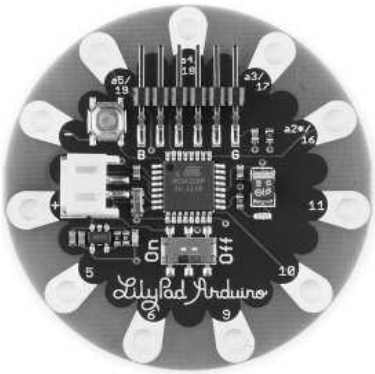


Figura 3.27 LilyPad Arduino Simple.

LilyPad Arduino SimpleSnap

El modelo LilyPad Arduino SimpleSnap utiliza el microcontrolador Atmega328 a 8 MHz. Este modelo es similar a LilyPad Arduino Simple, excepto que tiene una batería de lithium con circuito cargador y en lugar de orificios conductores (*through-holes*), tiene broches conductores de presión para instalarse en ropa y prendas textiles con hilo conductor y 9 puertos digitales de entrada/salida. El modelo LilyPad Arduino SimpleSnap fue diseñado y desarrollado por Leah Buechley y SparkFun Electronics. En la figura 3.28 se muestra este modelo y en la tabla 3.17 sus principales características técnicas.

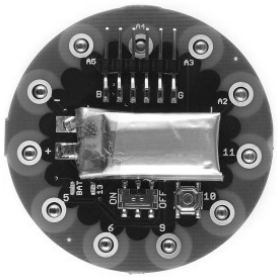
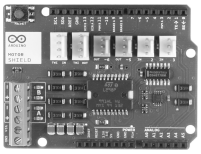


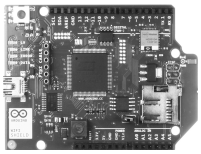
Figura 3.28 LilyPad Arduino SimpleSnap.

Interfaces Arduino Shields

Los modelos Arduino utilizan interfaces electrónicas especificas denominadas *shields* para acoplar la impedancia y potencia eléctrica de motores, aspectos de comunicación inalámbrica WiFI y protocolos de internet con amplia gama de aplicaciones. Por ejemplo, la figura 3.29a muestra el Arduino motor shield se basa en el puente HL298, el cual es ampliamente utilizado para acoplamiento de impedancia y suministro de potencia entre sistemas digitales y motores de corriente directa; permite controlar la dirección de giro y velocidad de movimiento del motor. El Arduino WiFi shield de la figura 3.29b conecta una tarjeta Arduino a internet en forma inalámbrica. En el sitio Web: http://arduino.cc/en/Main/Products puede encontrar mayor información de interfaces shield para varias aplicaciones.



(a) Motor shield



(b) WiFi shield

Figura 3.29 Interfaces electrónicas shield para motores y comunicación WiFi.

e-textiles

El término e-textiles (textiles electrónico o textiles inteligentes) se refiere a los productos que emplean telas o prendas de vestir con hilo conductor que permiten integrar textiles como elementos electrónicos, circuitos integrados, LEDs, sensores, actuadores, baterías convencionales, sistemas digitales y microcontroladores dando origen a pequeñas computadoras; también se les denomina tecnología portátil o ropa inteligente.
Este tipo de sistemas tiene amplias aplicaciones en el diseño de interiores de casas habitación o edificios inteligentes para monitoreas variables como temperatura, presión ambiental, condiciones de iluminación, humedad (domótica).
La tecnología actual hace posible tener electrónica integrada directamente en los sustratos textiles. La integración de componentes pueden ser desde elementos pasivos como resistencias, capacitancias, inductancias, conductores y componentes activos tales como diodos, transistores y células solares.
Algunos sensores que se utilizan son fabricados de algodón que determinan patrones fisiológicos y proporcionar información de variables ambientales.
Los proyectos de investigación e-textiles incluyen botones táctiles fabricados con tejidos textiles y mediante hilos conductivos son conectados a microcontroladores para procesar la información enviándola a un conjunto de LEDs que forman redes de fibra tejida para formar pantallas.

Fibertronics

Fibertronics es un campo de la tecnología moderna que estudia la funcionalidad electrónica y computacional sobre la integración en fibras textiles que emplean hilo conductor.
El diseño y fabricación de elementos conductores, capacitores y resistencias requiere del estudio y análisis de materiales textiles con capacidades conductivas, resistivas y con la adecuada flexibilidad para que puedan ser cocidas o tejidas. Hoy en día, existen una gama amplia de semimateriales textiles con propiedades conductivas que que incluyen fibras metálicas mezcladas con fibras textiles para ser utilizados como hilos conductores y de esta forma integrar conexiones a sistemas digitales o microprocesadores.
Algunos materiales de adecuados para e-textiles lo son la clase electrónica orgánica construidas a base de pinturas y tintes con propiedades muy similares a los semiconductores. Algunas funciones a nivel laboratorio son: transistores de fibra orgánica-textil sin utilizar elementos metálicos, células solares orgánicas de fibras textiles.
Nuevos conceptos surgen con estas áreas, tales como: **e-costura** significa la técnica de coser circuitos electrónicos usando hilo conductor.
Otro aspecto importante a considerar, es que una prenda textil electrónica puede ser lavable, esto significa que deben incorporar elementos aislantes al contacto con agua o ambientes húmedos.

Tabla 3.17 Modelos Arduino LilyPad para e-textil y prendas de vestir inteligentes.

Características	USB	Simple	SimpleSnap	Arduino
Microcontrolador	Atmega32u4 8 bits	Atmega328 8 bits	Atmega328 8 bits	Atmega168V 8 bits Atmega328V 8 bits
Voltaje de alimentación	3.3 V	2.7 a 5.5 V	2.7 a 5.5 V	2.7 a 5.5 V
Voltaje de entrada	3.8 a 5 V	2.7 a 5.5 V	2.7 a 5.5 V	2.7 a 5.5 V
Puertos digitales (I/O)	9 de los cuales 4 son salisdas PWM	9 de los cuales 5 son salidas PWM	9 de los cuales 5 son salidas PWM	14 de los cuales 6 son salidas PWM
Entradas analógicas	4	4	4	6
Corriente DC por pin (I/O)	40 mA	40 mA	40 mA	40 mA
Memoria flash	32 Kb (de los cuales 4 Kb son para *bootloader*)	32 Kb (de los cuales 2 Kb son para *bootloader*)	32 Kb (de los cuales 2 Kb son para *bootloader*)	16 Kb (de los cuales 2 Kb son para *bootloader*)
SRAM	2.5 Kb	2 Kb	2 Kb	1 Kb
EEPROM	1 Kb	1 Kb	1 Kb	512 bytes
Frecuencia de operación	8 MHz	8 MHz	8 MHz	8 MHz
Contiene integrado un LED en el pin 13 (puerto digital configurable como salida) con resistencia limitadora de corriente (enciende en "1=HIGH", se apaga con "0=LOW")	Si	No	No	No

3.5 Resumen

Los microcontroladores Atmel de 8 y 32 bits con arquitectura AVR integran alta eficiencia, alta velocidad de procesamiento y desempeño, optimizando su forma de trabajo para consumir baja potencia.

La arquitectura AVR obtiene beneficio de los dos tipos de buses con los que cuenta, uno para instrucciones para leer y ejecutar instrucciones en la memoria flash y el otro bus para manipulador datos en la memoria SRAM. Esta tecnología asegura que una instrucción puede ser ejecutada en cada ciclo de reloj, eliminando estados de espera y optimizando tiempo de procesamiento.

El gestor de arranque y cargador de código de usuario (*bootloader*), también conocido como *firmware* es el programa que recibe el código de máquina del programa (sketch) del usuario, el cual es enviado desde la computadora por comunicación USB, usando el ambiente de programación Arduino. Una vez que finaliza el proceso de descarga del código, el microcontrolador inicia la ejecución en forma secuencial de las instrucciones del sketch en forma permanente (mientras tenga fuente de alimentación externa o utilizando el cable USB).

El sitio Web del fabricante Atmel de microcontroladores con arquitectura AVR

http://www.atmel.com

Arduino es un sistema empotrado dedicado al control y automatización de procesos físicos, así como amplias aplicaciones de robótica y mecatrónica. La plataforma electrónica de Arduino se deriva de la familia de microcontroladores Atmel de 8 bits (Atmega 48PA/ 88PA/168PA/328PA) con memoria flash integrada especialmente para almacenar programas y otra memoria separada para manipulador datos SRAM; adicionalmente cuenta con un conjunto de circuitos periféricos (timers, convertidores analógico/digital, puertos de entrada/salida, UARTs, etc.) que le permiten comunicarse con el mundo exterior.

En este capítulo se ha estudiado los aspectos operativos del modelo Arduino UNO el

cual es la tarjeta electrónica base para entender el funcionamiento de otros modelos más avanzados. Se ha descrito la formas en las que se puede suministrar alimentación eléctrica hasta los detalles de las señales eléctricas que el usuario tiene acceso en los conectores pin-hembra montados directamente sobre la tarjeta, haciendo hincapié sobre las múltiples funciones que pueden tener una línea digital.

Arduino contiene varios tipos de tarjetas electrónicas para cubrir diversas aplicaciones: desde propósito general para instrumentación electrónica, automatización de procesos físicos e ingeniería robótica y mecatrónica (Arduino UNO, Leonardo, Due, Mega, Mega 2560, Mega ADK, Mini, Pro, Pro mini, Micro, Nano, FIO y Galileo); integración de sensores y robótica móvil (Esplora y Arduino Robot); electrónica textil (**e-textil**); y prendas de vestir inteligentes dinámicas e interactivas fabricados con fibras textiles de materiales especiales de hilos conductores (modelos LilyPad: Arduino, USB, Simple y SimpleSnap), comunicación inalámbrica considerando WiFi y bluetooth, hasta protocolos para enlace de ethernet y redes. y protocolos de redes.

El sitio Web de las tarjetas Arduino donde el usuario puede acceder a información técnica, documentación y aplicaciones es la siguiente:

www.arduino.cc

3.6 Referencias selectas

Esta sección proporciona al usuario enlaces electrónicos y sitios Web con información técnica, notas, manuales de usuario, aplicaciones foro y sociedades que utilizan los microcontroladores con arquitectura AVR de la compañía Atmel, los cuales constituyen la plataforma electrónica de los diversos modelos de tarjetas Arduino.

Web http://www.atmel.com/avr

Web http://www.atmel.com/products/microcontrollers/avr/default.aspx

http://www.atmel.com/avr

El diseño electrónico de las primeras tarjetas Arduino fue inspirado de otra tarjeta con la filosofía de arquitectura abierta y software libre denominada Wiring:

http://www.wiring.com

Información sobre prendas electrónicas y ropa inteligente, productos **e-textil** y **e-costura** puede ser encontrada en:

http://lilypadarduino.org/

http://www.fabrick.it/

http://www.aniomagic.com/

http://www.adafruit.com/category/92

http://www.textilefutures.co.uk/

Kits de desarrollo y sistemas de evaluación (*starter kits*) de microcontroladores AVR Atmel pueden ser consultados en el siguiente sitio:

http://www.atmel.com/tools/STK500.aspx

Un ejemplo de un programa gestor de arranque (arrancador/cargador, *firmware* libre) del modelo Arduino UNO es el *bootloader* `optiboot` disponible en el siguiente sitio Web:

http://code.google.com/p/optiboot

3.7 Problemas propuestos

Los conceptos básicos de la arquitectura electrónica AVR y sus aspectos cualitativos son elementos claves para entender el funcionamiento y aplicaciones de los diferentes modelos Arduino. Esta sección está destinada a valorar los conocimiento adquiridos por el usuario sobre las plataformas electrónicas Arduino.

3.7.1 Describa las características principales de la arquitectura AVR.

3.7.2 Proporcione una descripción cualitativa de las interrupciones de los microcontroladores AVR.

3.7.3 ¿Qué ventajas tiene usar interrupciones?

3.7.4 Describa el diagrama a bloques de la arquitectura electrónica que utiliza las tarjetas Arduino.

3.7.5 ¿Cuáles son los principales periféricos que utilizan las tarjetas Arduino?

3.7.6 Si un convertidor analógico/digital emplea 10 bits de resolución:

a) ¿Cuál es la mínima lectura que puede discernir, tomando en cuenta que el voltaje de referencia es de 5 V?

b) ¿Es suficiente para propósitos de control y automatización contar con 10 bits de resolución en el convertidor analógico/digital?

3.7.7 ¿Cómo puede clasificar a grandes rasgos por sus aplicaciones a las tarjetas Arduino?

3.7.8 ¿Qué tipos de tarjetas electrónicas Arduino de propósito general se emplean en aplicaciones de ingeniería robótica y mecatrónica?

3.7.9 ¿Qué significa el concepto LilyPad?

3.7.10 ¿Qué tipo de modelos Arduino se emplean en aplicaciones inalámbricas?

3.7.11 Describa los modelos Arduino que utilizan sensores integrados y manejo de robots móviles.