

en electrónica, sin embargo otro acierto de Arduino a sido socializar la tecnología acercar la electrónica y la programación a todos los niveles incluso a quienes no entienden de electrónica.

Programar Arduino consiste en traducir a líneas de código o líneas de programa, las tareas que tiene que realizar, y en función de las condiciones del entorno, información que le llega desde sensores y sistemas conectados al propio Arduino, interactuar con el mundo exterior mediante actuadores.

Arduino proporciona un entorno de programación sencillo y potente para programar, pero además incluye las herramientas necesarias para compilar el programa y programar la memoria flash o memoria de programa del microcontrolador.

Además el entorno de trabajo nos ofrece un sistema de gestión de librerías y placas muy práctico, si bien carece de funciones avanzadas típicas de otros entornos de programación es suficiente para programar y desarrollar aplicaciones simples y complejas con Arduino.

3.6 COMO TRABAJAR CON ARDUINO

Un programa de Arduino se denomina sketch y tiene la extensión .ino.

No es necesario que un sketch esté en un único archivo, pero si es imprescindible que todos los archivos estén dentro del mismo directorio que el archivo principal.

Para programar un Arduino tenemos dos caminos posibles.

1. *Programación clásica para Arduino.* Que es probablemente la forma en que la mayoría del simpatizante Arduino hace su trabajo. Casi no requiere conocimientos de electrónica y programación, las bibliotecas hacen todo el trabajo y diseñar una aplicación desde cero toma muy poco tiempo y los resultados son poco menos que sorprendentes.

La programación de Arduino es la programación de un microcontrolador y esto siempre ha sido algo más relacionado con ingeniería y técnicos avanzados en electrónica, sin embargo otro acierto de Arduino a sido socializar la tecnología, acercar la electrónica y la programación a todos los niveles incluso a quienes no entienden de electrónica.

Programar Arduino consiste en traducir a líneas de código o líneas de programa, las tareas que tiene que realizar, y en función de las condiciones del entorno, información que le llega desde sensores y sistemas conectados al propio Arduino, interactuar con el mundo exterior mediante actuadores.

Arduino proporciona un entorno de programación sencillo y potente para programar, pero además incluye las herramientas necesarias para compilar el programa y programar la memoria flash o memoria de programa del microcontrolador.

Además el entorno de trabajo nos ofrece un sistema de gestión de librerías y placas muy práctico, si bien carece de funciones avanzadas típicas de otros entornos de programación es suficiente para programar y desarrollar aplicaciones simples y complejas con Arduino.

3.6 COMO TRABAJAR CON ARDUINO

Un programa de Arduino se denomina sketch y tiene la extensión .ino.

No es necesario que un sketch esté en un único archivo, pero si es imprescindible que todos los archivos estén dentro del mismo directorio que el archivo principal.

Para programar un Arduino tenemos dos caminos posibles.

1. *Programación clásica para Arduino*. Que es probablemente la forma en que la mayoría del simpatizante Arduino hace su trabajo. Casi no requiere conocimientos de electrónica y programación, las bibliotecas hacen todo el trabajo y diseñar una aplicación desde cero toma muy poco tiempo y los resultados son poco menos que sorprendentes.

2. *Programación avanzada para Arduino*. No tenemos que olvidar que detrás de una placa Arduino hay un poderoso microcontrolador, una verdadera computadora en miniatura, la programación clásica obvia muchos aspectos técnicos para no abrumar al programador con engorrosos detalles del funcionamiento electrónico de la CPU con la que está trabajando.

La programación con algún conocimiento básico de la tecnología que se está usando permite diseñar aplicaciones, máquinas y sistemas de verdadera competencia profesional.

Arduino es sin duda la forma más sencilla de ingresar al mundo de la electrónica moderna, en estos tiempos la electrónica “habla”, se comunica, está conectada con el mundo.

En Arduino existen los shields que son placas de circuitos modulares que se montan unas encima de otras para ampliar el hardware y las capacidades de Arduino, los bajos costos de estas placas y la gran variedad disponible permite rápidamente llevar al plano real proyectos ambiciosos de robótica, domótica y sistemas electrónicos en general.

3.7 ARQUITECTURA DE UN SKETCH

Un programa Arduino se divide en tres partes principales.

```
#include "LiquidCrystal.h"      // Biblioteca para el LCD
#define ledPin 13              // Pin del LED de la placa Mega
// ..... Variables del programa .....
byte adc_Pin = 0;
volatile unsigned char muestras_0=0,muestras_1=0;
```

```
void setup () {
  lcd.begin(20,4);
  pinMode(ledPin, OUTPUT); // Pin del LED de la placa
  digitalWrite(13,LOW );   // LED inicia apagado
  ADCSRA = bit (ADEN);
}
```

```
void loop () {
  conversion = (conv_0 *5.0)/1024;
  dtostrf(conversion,2,2,buffn);
  sprintf(buffer,"%s", buffn);
  lcd.setCursor(11,0);
  lcd.print(buffer);
}
```

En la primera sección se declaran todas las variables, funciones y bibliotecas. Se puede decir que en esta sección se informa al programa de todos los elementos que necesitará para construir el proyecto.

Recuerde que no puede usar nada que no haya sido declarado previamente.

La segunda sección contiene la configuración del Hardware, toda la electrónica que usará el proyecto se configura en esta sección.

La sección tres contiene el bloque principal del programa, todo sucederá en esta sección, desde aquí el programa puede bifurcar a otras secciones y rutinas sin embargo siempre regresará a este “bucle”.

Anteriormente se hablaba del Stack o pila, cuando el programa sale del bucle loop() a ejecutar código en otra función o en una interrupción, saca una “radiografía” del estado de algunos de sus registros para saber exactamente donde estaba y que estaba haciendo, lo guarda en el Stack para recuperarlo luego y así conocer donde y como tiene que regresar cuando este llamado fuera del loop() termine.

3.8 CONCEPTOS BÁSICOS PARA UNA PROGRAMACIÓN EXITOSA

Para minimizar los posibles problemas a la hora de programar Arduino es bueno tener claro algunos conceptos.

- ▶ *Debe estar seguro que la fuente de alimentación no tiene problemas. El voltaje es el correcto, no hay falsos contactos ni cables sueltos. Si algo no funciona usted puede desconfiar de su código pero debe tener certeza que su electrónica funciona correctamente.*
- ▶ *El lenguaje de programación distingue mayúsculas de minúsculas, si escribe “Dato” y luego hace referencia a “dato” el compilador dará un error indicando que la variable no ha sido declarada o no sabe que es.*
- ▶ *No mezcle mayúsculas y minúsculas en un programa, es confuso y poco elegante.*
- ▶ *No utilice nombres con espacios, es preferible agregar un “_” para representar un espacio.*
- ▶ *No utilice nombres largos para variables o nombres de archivos.*
- ▶ *Trabaje prolijo, cree una carpeta donde contener todos sus proyectos que a su vez tendrán cada uno de ellos su propia carpeta individual. En estas*

carpetas podrá agregar notas explicativas, fotos y toda la información que le servirá en el futuro para recordar rápidamente como realizó el trabajo.

Trabajar con Arduino es realmente sencillo sin embargo relea los consejos anteriores y asegúrese de cumplirlos se evitará muchos dolores de cabeza y pérdidas de tiempo.

3.9 EL MICROCONTROLADOR DEL ARDUINO UNO

Un microcontrolador es un circuito integrado capaz de ser programado desde una computadora y seguir la secuencia de acciones escrita en el programa grabado en su memoria.

Cuando el microcontrolador ejecuta una instrucción que definimos en el sketch o programa. Arduino, internamente hace muchas operaciones y cada una de esas operaciones se ejecuta al ritmo de un reloj.

Para el ATmega 328p que tiene una frecuencia de 16 MHz, es decir, cada ciclo tarda 0,0000000625 segundos = 0,0625 microsegundos = 62,5 nanosegundos, este es el tiempo que se conoce como **un ciclo de CPU** y las instrucciones o comandos pueden llevar más de un ciclo de CPU.

Es importante entender que si bien Arduino es muy simple de usar, esto no significa que sea simple en su arquitectura.

Su electrónica es verdaderamente compleja, un microcontrolador ATmega328 es el “cerebro” de Arduino. Un microcontrolador se diferencia de un microprocesador en que el primero centra todos sus recursos en una tarea específica aquella para lo cual fue programado. Todos sus recursos están embebidos en el propio chip, no hay posibilidad de expandir memorias o mejorar sus desempeño a nivel de hardware, solo podemos “afinar” la forma de programar y así obtener un mejor rendimiento del chip.

En el caso de los microprocesadores los recursos de memoria son externos y por tanto se pueden expandir como es el caso de la memoria RAM en una computadora o cambiar el disco para aumentar a capacidad de almacenamiento. En un microcontrolador el disco duro es el equivalente a la memoria de programa o memoria FLASH.