

Lab. Dispositivos de Almacenamiento y de E/S.

Previo °N 8: "Utilizando Arduino como dispositivo de entrada y salida."

1. Explique cuáles son las partes que componen la tarjeta Arduino UNO y para qué sirven las terminales (pines y conectores).

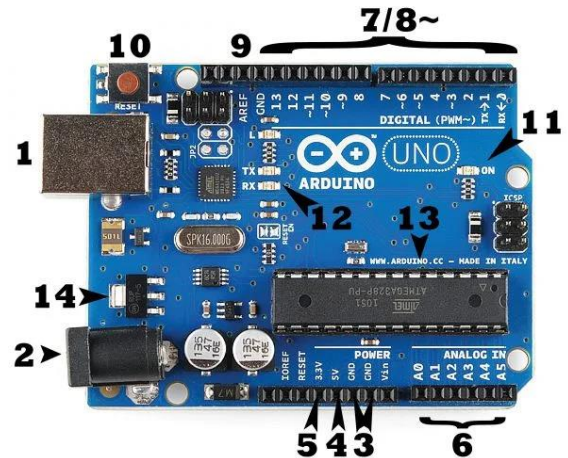
Potencia - USB (1) / Conector de Adaptador (2)

Alimentación de corriente directa a 5V desde un cable USB a el ordenador o de un adaptador a una batería de 9V como máximo.

Pines (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

Los pines en la placa Arduino es donde se conectan los cables de un circuito. El Arduino tiene varios tipos diferentes de entradas, cada uno de las cuales está marcado en el tablero y utilizan para diferentes funciones:

- **GND (3):** Hay varios pines GND en el Arduino, cualquiera de los cuales pueden ser utilizados para conectar a tierra el circuito.
- **5V (4) y 3.3V (5):** Suministran 5 voltios de energía y 3.3 voltios de potencia respectivamente.
- **Analógico (6):** Estos pines (A0 a A5) pueden leer la señal de un sensor analógico (sensor de temperatura) y convertirlo en un valor digital que podemos leer.
- **Digital (7):** Son los pines digitales (0 al 13), se pueden utilizar tanto para la entrada digital (oprimir un botón) y salida digital (encender un LED).
- **PWM (8):** Estos pines (~3, ~5, ~6, ~9, ~10 y ~11) actúan como pines digitales normales, pero también se pueden usar para algo llamado Modulación por ancho de pulsos (PWM).
- **AREF (9):** Soportes de referencia analógica. La mayoría de las veces se puede dejar este pin solo. A veces se utiliza para establecer una tensión de referencia externa (entre 0 y 5 voltios) como el límite superior para los pines de entrada analógica.



Botón de reinicio (10): Empujando este botón se conectará temporalmente el pin de reset a tierra y reinicie cualquier código que se carga en el Arduino.

Indicador LED de alimentación (11): Este LED debe encenderse cada vez que conecte la placa Arduino a una toma eléctrica.

LEDs RX TX (12): Los pines digitales (0, 1), y los indicadores LED de TX y RX (12) son los responsables de la comunicación en serie. Estos LEDs nos darán indicaciones visuales siempre que nuestro Arduino está recibiendo o transmitiendo datos (como cuando estamos cargando un nuevo programa).

Microcontrolador (13): El cerebro del Arduino. Circuito integrado ATmega de la empresa ATMEL).

Regulador de Voltaje (14): Controla la cantidad de tensión que se suministra a la placa Arduino.

2. Explique qué es el IDE de Arduino y en qué lenguaje de programación está basado.

Arduino IDE es un entorno de desarrollo y en él se realiza la programación de cada una de las placas de Arduino. Tiene como base el entorno de Processing al igual que un lenguaje de programación fundamentado en Wiring. El lenguaje de programación de Arduino está basado en C++.

3. Crear un código en Arduino (comentado) que habilite 8 pines digitales como salida y 4 pines digitales como entrada, el dato de entrada será leído, para seleccionar los datos de salida de acuerdo a sus 3 últimos números de su número de cuenta (de ambos alumnos). Ejemplo: 304179958, por lo tanto se tomarán el 9, 5 y 8, estos números deberán aparecer en forma individual en un Display de 7 segmentos, las entradas serán seleccionadas por medio de los pines configurados como entradas.

ENTRADA				SALIDA	
S3	S2	S1	S0	Display 7 segmentos	
1	0	0	1	9
0	1	0	1	5
1	0	0	0	8

<pre>//Definicion de constantes para la asignacion de pines const int pinesSalida[7] = { 2, 3, 4, 5, 6, 7, 8 }; const int pinesEntrada[4] = { 9, 10, 11, 12 }; //Definicion de la tabla para el despliegue en un display de 7 //segmentos con catodo comun const byte numerosDisplay[10] = {0b0111111, //0 0b0000110, //1 0b1011011, //2 0b1001111, //3 0b1100110, //4 0b1101101, //5 0b1111101, //6 0b0000111, //7 0b1111111, //8 0b1101111}; //9 //Variable que va a guardar el numero que se va a leer int numero = 0; void setup() { //Asignacion de pines de salida for(int i = 0; i < 7; i++) { pinMode(pinesSalida[i], OUTPUT); } //Asignacion de pines de entrada for(int i = 0; i < 4; i++) { pinMode(pinesEntrada[i], INPUT); } }</pre>	<pre>void loop() { //Entradas binarias bool x = digitalRead(pinesEntrada[0]); bool y = digitalRead(pinesEntrada[1]); bool z = digitalRead(pinesEntrada[2]); bool w = digitalRead(pinesEntrada[3]); //Se guarda la entrada binaria en su respectivo orden xyzw numero = x << 3; numero = y << 2; numero = z << 1; numero = w << 0; prenderSegmentos(numero); } //Funcion para desplegar el numero en el display void prenderSegmentos(int numero) { byte numeroBinario = numerosDisplay[numero]; for (int i = 0; i < 7; i++) { int bit = bitRead(numeroBinario, i); digitalWrite(pinesSalida[i], bit); } }</pre>
--	--

Referencias:

- **Arduino - Tutorial 0: Introducción** "<https://www.instructables.com/id/Arduino-Tutorial-0-Introducci%C3%B3n/>".
- **Curso de Arduino 2013** "<https://codigofacilito.com/cursos/Arduino>".