

Electrónica Digital

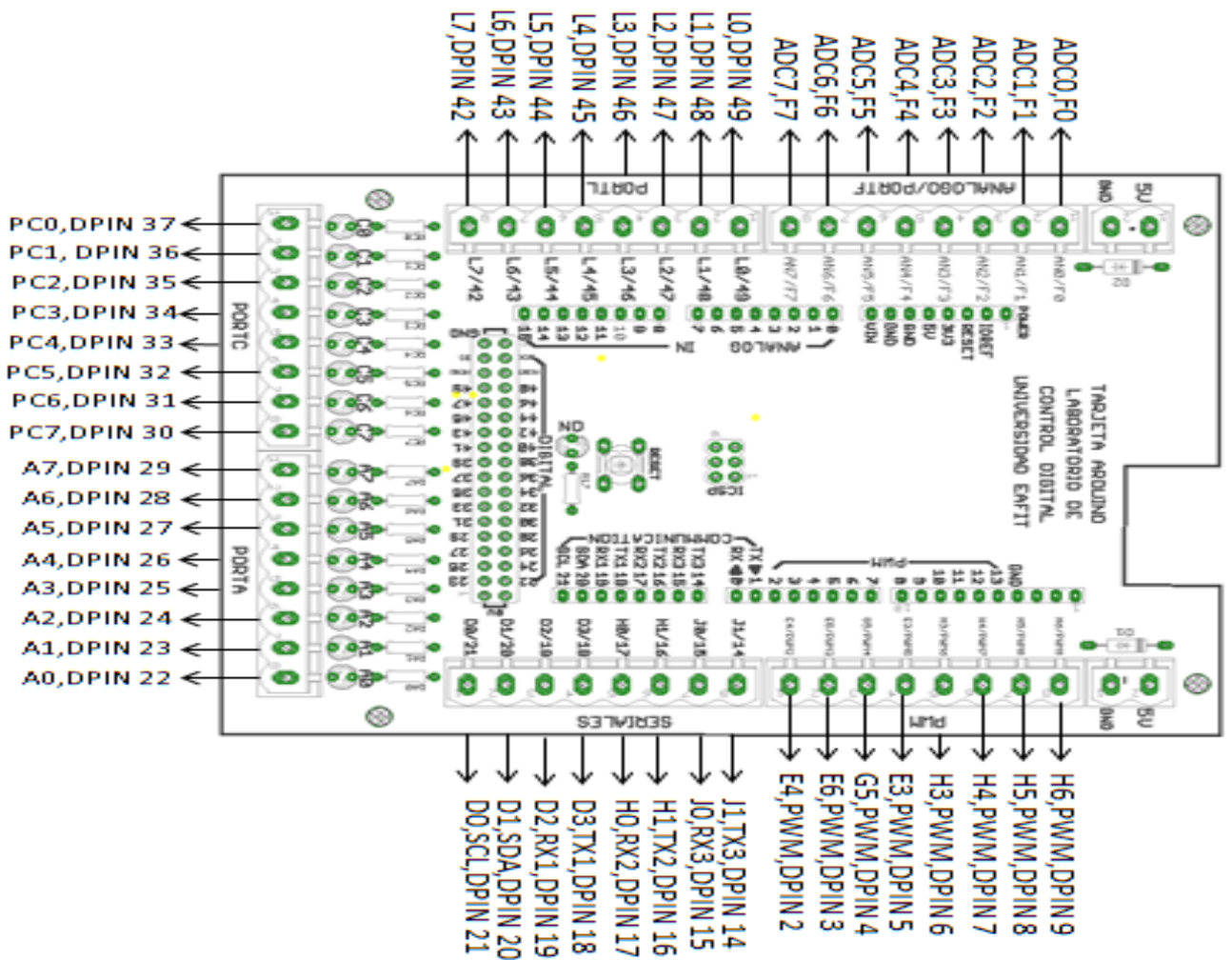
Clase 6

Clase Práctica con Entradas y Salidas Digitales

Ficha técnica tarjeta de desarrollo mega

Microcontrolador	<i>ATmega2560</i>
Voltaje de funcionamiento	5 V
Voltaje de entrada (recomendado)	7 – 12 V
Voltaje de entrada (los limites)	6 – 20 V
Pines de I/O digital	54 (<i>de los cuales 15 proporcionan una salida PWM</i>)
Pines de entrada análoga	16
Corriente DC I/O Pin	40 mA
corriente DC para 3.3V Pin	50 mA
Memoria flash	256 KB <i>de los cuales 8 KB utilizadas por gestor de arranque</i>
SRAM	8 KB
EEPROM	4 KB
Velocidad del cristal	16 MHz

Identificación de pines de practicas



Configuración de pines I/O

- En el **setup()** se configuran los pines del Arduino como **entrada** o **salida**, se configura el puerto serial para luego hacer el monitoreo de programas y se desactivan todas las salidas.
 - Con la instrucción `pinMode(PIN, OUTPUT);` // Configura el pin como salida
 - Con la instrucción `pinMode(PIN, INPUT);` // Configura el pin como entrada
 - Con la instrucción `Serial.begin(9600);` // Configura la comunicación a 9600 baudios
 - Con la instrucción `digitalWrite(PIN, LOW);` // Desactiva la salida pin
- Ejemplo
 - Configure un programa en Arduino para un botón pulsador y un motor



```
//Definición de pines de I/O
#define P1 37 //Pulsador P1 conectado en el pin 37
#define M1 22 //Motor M1 conectado en el pin 22

void setup() {
  //Definición de que pin es entrada y que es salida
  pinMode(P1, INPUT); //P1 como entrada
  pinMode(M1, OUTPUT); //M1 como salida

  //Limpieza de salidas
  digitalWrite(M1, LOW); //Apago M1
}
```

Instrucciones básicas

- ¿Cómo preguntar si una entrada esta activada?

```
if(digitalRead(Pulsador) == HIGH) { //Si el pulsador esta en HIGH  
    //Sentencia;  
}
```

La “**sentencia**” corresponde a la **decisión** que se va a tomar en caso de que la **entrada** Pulsador este **activada**.

- ¿Cómo preguntar si una entrada esta desactivada?

```
if(digitalRead(Pulsador) == LOW) { //Si el pulsador esta en LOW  
    //Sentencia;  
}
```

- ¿Cómo activar o desactivar una salida?

```
digitalWrite(L1, HIGH); //Activa el LED L1  
digitalWrite(L1, LOW); //Desactiva el LED L1
```

Instrucciones básicas

- ¿Como preguntar si dos entradas están activadas simultáneamente?

```
if((digitalRead(Pulsador1) == HIGH) && (digitalRead(Pulsador2) == HIGH)) {  
    //Sentencia;  
}
```

El “&&” corresponde a la pregunta “**AND**”

- ¿Como preguntar si cualquiera de las dos entradas esta activada?

```
if((digitalRead(Pulsador1) == HIGH) || (digitalRead(Pulsador2) == HIGH)) {  
    //Sentencia;  
}
```

El “||” corresponde a la pregunta “**OR**”

Ejemplo 1

- Encender y apagar un led L1 conectado al Pin 22 cada medio segundo si el suiche SW1 conectado al pin 37 esta activado.

```
//Definición de pines de I/O
#define L1 22 //LED L1 conectado en el pin 22
#define SW1 37 //Suiche SW1 conectado en el pin 37

//Definición de constantes
const unsigned long TIT = 500; //Constante de titileo TIT iniciada en 500 ms

void setup() {
    //Definición de que pin es entrada y que es salida
    pinMode(SW1, INPUT); //SW1 como entrada
    pinMode(L1, OUTPUT); //L1 como salida

    //Limpieza de salidas
    digitalWrite(L1, LOW); //Apago L1
}

void loop() {
    if(digitalRead(SW1) == HIGH) { //Si el suiche SW1 esta en HIGH
        digitalWrite(L1, HIGH); //Prendo L1
        delay(TIT); //Retardo de TIT milisegundos (500 ms)
        digitalWrite(L1, LOW); //Apago L1
        delay(TIT); //Retardo de TIT segundo (500 ms)
    }
}
```

Ejemplo 2

- Encender y apagar un led conectado al Pin 22 cada medio segundo si el suiche conectado al pin 37 esta activado. Use el monitor serial para monitorear la ejecución del programa.

```
//Definición de pines de I/O
#define L1 22 //LED L1 conectado en el pin 22
#define SW1 37 //Suiche SW1 conectado en el pin 37

//Definición de constantes
const unsigned long TIT = 500; //Constante de titileo TIT iniciada en 500 ms

void setup() {
    //Definición de que pin es entrada y que es salida
    pinMode(SW1, INPUT); //SW1 como entrada
    pinMode(L1, OUTPUT); //L1 como salida

    //Comunicaciones
    Serial.begin(9600); //Configura el monitor serial

    //Limpieza de salidas
    digitalWrite(L1, LOW); //Apago L1
}

void loop() {
    if(digitalRead(SW1) == HIGH) { //Si el suiche SW1 esta en HIGH
        digitalWrite(L1, HIGH); //Prendo L1
        Serial.println("Led encendido"); //Muestra Led encendido
        delay(TIT); //Retardo de TIT milisegundos (500 ms)
        digitalWrite(L1, LOW); //Apago L1
        Serial.println("Led apagado"); //Muestra Led apagado
        delay(TIT); //Retardo de TIT segundo (500 ms)
    }
}
```


Ejemplo 3

- Controlar el encendido de tres motores que funcionen de la siguiente manera: Al oprimir el pulsador de inicio enciende el motor1, pasados 5 segundos enciende el motor 2 y 5 segundos después enciende el motor3. Si se oprime el pulsador detener, los motores se apagan en orden contrario al que fueron encendidos y con la misma diferencia de tiempo.

```
#define motor1 22           // motor1 en el pin 22
#define motor2 23           // motor2 en el pin 22
#define motor3 24           // motor2 en el pin 24
#define inicio 37           // inicio en el pin 37
#define detener 36          // detener en el pin 36

void setup() // configura los pines usados
{
  pinMode(motor1, OUTPUT);    // configura el motor1 como salida
  pinMode(motor2, OUTPUT);    // configura el motor2 como salida
  pinMode(motor3, OUTPUT);    // configura el motor3 como salida
  pinMode(inicio, INPUT);     // configura el pulsador inicio como entrada
  pinMode(detener, INPUT);    // configura el pulsador detener como entrada
  digitalWrite(motor1, LOW);  // desactiva el motor1
  digitalWrite(motor2, LOW);  // desactiva el motor2
  digitalWrite(motor3, LOW);  // desactiva el motor3
  Serial.begin(9600);         //Configura el monitor serial
}
```

Ejemplo 3

```
void loop()                // inicia el bucle del programa
{
  if(digitalRead(inicio)==HIGH)
  {
    digitalWrite(motor1, HIGH);    // activa el motor1
    Serial.println ("motor1 encendido");
    delay(5000);                   // espera medio segundo
    digitalWrite(motor2, HIGH);    // activa el motor2
    Serial.println ("motor2 encendido");
    delay(5000);                   // espera medio segundo
    digitalWrite(motor3, HIGH);    // activa el motor3
    Serial.println ("motor3 encendido");
  }
  if(digitalRead(detener)==HIGH)
  {
    digitalWrite(motor3, LOW);     // apaga el motor3
    Serial.println ("motor3 apagado");
    delay(5000);                   // espera medio segundos
    digitalWrite(motor2, LOW);     // apaga el motor2
    Serial.println ("motor2 apagado");
    delay(5000);                   // espera medio segundo
    digitalWrite(motor1, LOW);     // apaga el motor1
    Serial.println ("motor1 apagado");
  }
}
```

Ejemplo 4

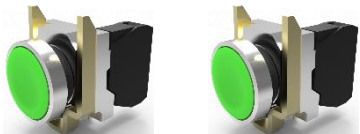
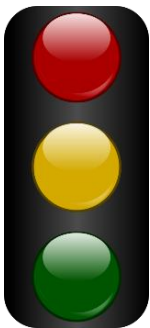
- Controlar el encendido y apagado de un bombillo de tal manera que al oprimir el pulsador el bombillo se encienda y apague 5 veces cada medio segundo vez que encienda el led, envíe el valor del contador al monitor serial.

```
#define led 22          // led en el pin 22
#define pulsador 37    // pulsador en el pin 37
unsigned int cont = 0;  // Contador en cero

void setup() // configura los pines usados
{
    pinMode(led, OUTPUT);    // configura el led como salida
    pinMode(pulsador, INPUT); // configura el pulsador como entrada
    digitalWrite(led, LOW);  // desactiva el led
    Serial.begin(9600);
}

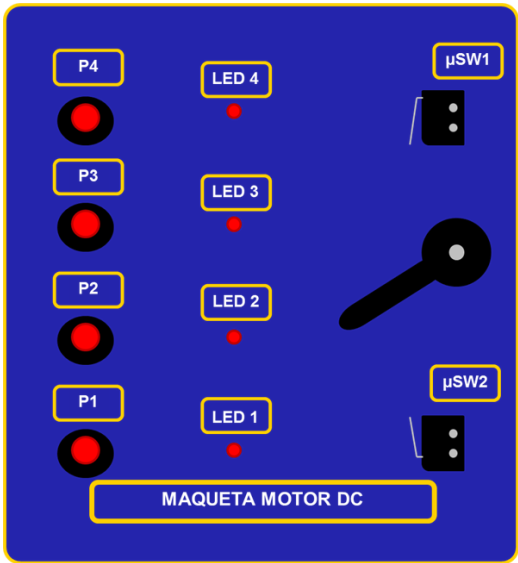
void loop() // inicia el bucle del programa
{
    if(digitalRead(pulsador)==HIGH)
    {
        cont=0;
        Serial.println(cont);
        while(cont<5)
        {
            digitalWrite(led, HIGH); // activa el led
            delay(500); // espera medio segundo
            digitalWrite(led, LOW); // desactiva el led
            delay(500); // espera 1 segundo
            cont++;
            Serial.println(cont);
        }
    }
}
```

- (Op 1) Escriba un programa que controle un semáforo y una vez se activen dos suiches de mantenimiento (ambos) el semáforo se desactive y quede solo el Led Rojo intermitente.



Pulsadores de Mantenimiento

- (Op 2) Escriba un programa que controle la maqueta MOTOR DC, haciendo que el motor suba y baje 3 veces, y luego que se quede titilando todos los LED hasta que se presione algún pulsador



PIN DEL ARDUINO	FUNCION
37	Microsuiche arriba(μSW1)
36	Microsuiche abajo (μSW2)
35	Pulsador 1 (P1)
34	Pulsador 2 (P2)
33	Pulsador 3 (P3)
32	Pulsador 4 (P4)
22	Subir Motor (SM)
23	Bajar Motor (BM)
24	Led 1 (L1)
25	Led 2 (L2)
26	Led 3 (L3)
27	Led 4 (L4)

MUCHAS GRACIAS