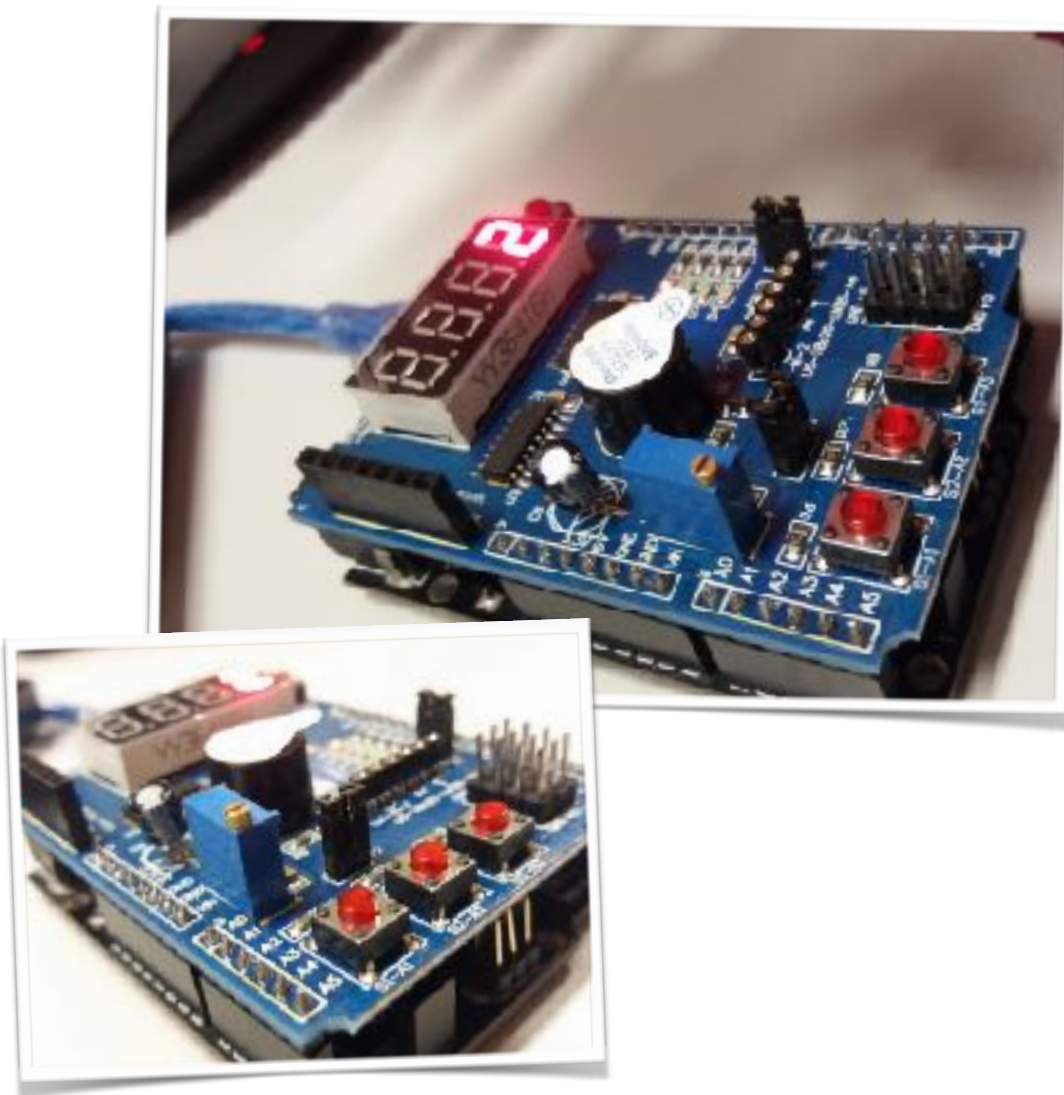


PROYECTO

Stefanía Antón y Paula García Arévalo



PROYECTO 1

DE 0...9

```
int i=0;
#define L 4
#define C 7
#define D 8
const byte MAP[] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82, 0xF8, 0x80, 0x90};
const byte POSICION[] = {0xF1, 0xF2, 0xF4, 0xF8};
void setup(){
  pinMode(L, OUTPUT);
  pinMode(C, OUTPUT);
  pinMode(D, OUTPUT);
  pinMode(A1, INPUT);
  Escribe(3, 0);
}
void loop(){
  Escribe(3, 0);
  if (digitalRead(A1) == LOW){

    delay(200);
    i++;

  }
}
void Escribe(byte Segmento, byte Valor){
  digitalWrite(L, LOW);
  shiftOut(D, C, MSBFIRST, MAP[Valor]);
  shiftOut(D, C, MSBFIRST, POSICION[Segmento]);
  digitalWrite(L, HIGH);
}
```

Primero definimos las variables al principio de cada función, al declarar la variable indicamos el tipo de variable y por lo tanto su valor. En este caso comenzamos con la variable “int”, que almacena valores enteros (1,2,0,-1...), a continuación utilizamos la palabra “#define”, para definir las constantes, después utilizamos la variable “const”, que indica que la variable no puede ser cambiada en tiempo de ejecución. “void setup” y “void loop” son dos partes necesarias para que el programa funcione, “void setup” recoge la configuración y el “void loop” es la función bucle.

Dentro del void setup encontramos la función “pinMode” que es el modo de trabajo de las variables, con ello configuramos las entradas y salidas. Después aparece el “void loop” donde le ordenamos al programa que donde aparece “i” escriba lo que le digamos pero además le añadimos la función “if”, donde en el caso de que se de una circunstancia, el programa escribirá lo que nosotros le digamos (en nuestro proyecto lo definimos como LOW, debido a que nuestras placas tienen una lógica inversa).

Para concluir le añadimos un Delay, que pausa el programa lo que nosotros le indiquemos. Algo que tenemos que tener en cuenta, es que después de cada función debemos poner al final “;”.



PROYECTO 2

SE DETIENE EN 9 Y COMIENZA OTRA VEZ

```
int i=0;
#define L 4
#define C 7
#define D 8
const byte MAP[] = {0x00,0x0F,0x04,0x00,0x09,0x02,0x82,0x0F,0x80,0x90};
const byte POSICION[] = {0x01,0x02,0x04,0x0F};
void setup () {
  pinMode(L,OUTPUT);
  pinMode(C,OUTPUT);
  pinMode(D,OUTPUT);
  pinMode(A1,INPUT);
  Escribe(1,0);
}
void loop() {
  if (digitalRead(A1)==LOW){
    Escribe(1,i);
    delay(200);
    i++;
    if (i==10){
      delay(200);
      i=0;
    }
  }
}
void Escribe(byte Segmento, byte Valor){
  digitalWrite(L,LOW);
  shiftOut(D, C, MSBFIRST, MAP[Valor]);
  shiftOut(D, C, MSBFIRST, POSICION[Segmento]);
  digitalWrite(L,HIGH);
}
```

Escribimos en Arduino la misma configuración a la anterior, sin embargo, añadimos un nuevo elemento en el que se le dice al programa que “i+”, lo que significa que la i debe ir aumentando y además le añadimos la condicional que la i debe aumentar hasta el 10, pero no llega a 10, se parará en el 9 y empezará otra vez.



PROYECTO 3

CUENTA ATRÁS

```

int i=0;
int j=1;
#define L 4
#define C 7
#define D 8
const byte MAP[] = {0xC0,0xF9,0xA4,0xBD,0x99,0xB2,0x12,0xF0,0x80,0x9C};
const byte POSICION[] = {0xF1,0xF2,0xF4,0xF8};
void setup() {
  pinMode(L,OUTPUT);
  pinMode(C,OUTPUT);
  pinMode(D,OUTPUT);
  pinMode(A1,INPUT);
  Serial.begin(9600);
}
void loop() {

  if (digitalRead(A1)==LOW){
    Serial.println(i);

    if (digitalRead(A1)==LOW){

      delay(200);
      i++;

    }
  }
  void describe(byte segmento, byte valor){
    digitalWrite(L,LOW);
    shiftOut(C, MSBFIRST, MAP[valor]);
    shiftOut(D, MSBFIRST, POSICION[segmento]);
    digitalWrite(L,HIGH);
  }
}

```

Con la misma configuración del programa, realizaremos una cuenta atrás, en este caso debemos añadir una nueva variable, “j”, que será igual a 1.

El programa avanzará debido a la función “i++” pero además al llegar a 9, comenzará la cuenta atrás.



