

Universidad Del Valle De Guatemala

01/02/2021

Digital 2

Helder Ovalle Barrios

Sección: 20

18349

Laboratorio #4

Link de github:

<https://github.com/Helder1121/Labsdigitaldos/tree/main/lab4>

Link de youtube:

https://www.youtube.com/watch?v=m_W9DQbhBYE

Progra comentada:

```
//Electronica Digital
```

```
//Laboratorio 4
```

```
//Helder Ovalle ;18349
```

```
//Seccion:20
```

```
//Pines del jugador 1
```

```
int led1 = PB_0;
```

```
int led2 = PB_1;
```

```
int led3 = PB_2;
```

```
int led4 = PB_3;
```

```
int led5 = PB_4;
```

```
int led6 = PB_5;
```

```
int led7 = PB_6;
```

```
int led8 = PB_7;
```

```
//Pines del jugador 2
```

```
int led9 = PA_5;
```

```
int led10 = PD_7;
```

```
int led11 = PD_2;
```

```
int led12 = PD_3;
```

```
int led13 = PC_4;

int led14 = PC_5;

int led15 = PC_6;

int led16 = PC_7;

//Pines del semaforo

int ledR = RED_LED;

int ledG = GREEN_LED;

int ledB = BLUE_LED;

//Indicadores de ganador (azules)

int led17 = PA_6;

int led18 = PA_7;


//Variables

int avanzar = PUSH2;

int avanzar2 = PUSH1;

//Puertos de la tiva push1 y push2

int presionado = 0;

int presionado2 = 0;

//variable para antirebote

int cont = 0;

int cont2 = 0;

//Variables de cont1 y cont2


//Funciones a llamar

void jugador1(void);

void jugador2(void);

void semaforo(void);

void ganador(void);
```

```
void setup() {  
  //Leds de ambos jugadores  
  pinMode(led1, OUTPUT);  
  pinMode(led2, OUTPUT);  
  pinMode(led3, OUTPUT);  
  pinMode(led4, OUTPUT);  
  pinMode(led5, OUTPUT);  
  pinMode(led6, OUTPUT);  
  pinMode(led7, OUTPUT);  
  pinMode(led8, OUTPUT);  
  pinMode(led9, OUTPUT);  
  pinMode(led10, OUTPUT);  
  pinMode(led11, OUTPUT);  
  pinMode(led12, OUTPUT);  
  pinMode(led13, OUTPUT);  
  pinMode(led14, OUTPUT);  
  pinMode(led15, OUTPUT);  
  pinMode(led16, OUTPUT);  
  //Semaforo  
  pinMode(ledR, OUTPUT);  
  pinMode(ledG, OUTPUT);  
  pinMode(ledB, OUTPUT);  
  //Indicador del ganador  
  pinMode(led17, OUTPUT);  
  pinMode(led18, OUTPUT);  
  //Push  
  pinMode(avanzar, INPUT_PULLUP);  
  pinMode(avanzar2, INPUT_PULLUP);  
}
```

```

semaforo();
}

void loop(){
  jugador1();
  jugador2();
  ganador();
}

//Funciones
void jugador1(void){
  //Aumentando el conta
  if (digitalRead(avanzar) == LOW){
    presionado = 1; //Cambiara el estado del push1
  }
  if (digitalRead(avanzar) == HIGH && presionado == 1){
    presionado = 0; //se reinicia la variable del antirrebote
    delay(100);
    cont++; //Incrementa el contador
  }
  //Casos posibles del jugador 1
  switch (cont){
  case 0:
    digitalWrite(led1, LOW); //INICIA TODO EN 0
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);
    digitalWrite(led4, LOW);
    digitalWrite(led5, LOW);
    digitalWrite(led6, LOW);
    digitalWrite(led7, LOW);

```

```
digitalWrite(led8, LOW);
```

```
break;
```

case 1:

```
digitalWrite(led1, HIGH); //Enciende el LED 1 y apaga los demas
```

```
digitalWrite(led2, LOW);
```

```
digitalWrite(led3, LOW);
```

```
digitalWrite(led4, LOW);
```

```
digitalWrite(led5, LOW);
```

```
digitalWrite(led6, LOW);
```

```
digitalWrite(led7, LOW);
```

```
digitalWrite(led8, LOW);
```

```
break;
```

case 2:

```
digitalWrite(led1, LOW); //Enciende el LED 2 y apaga los demas
```

```
digitalWrite(led2, HIGH);
```

```
digitalWrite(led3, LOW);
```

```
digitalWrite(led4, LOW);
```

```
digitalWrite(led5, LOW);
```

```
digitalWrite(led6, LOW);
```

```
digitalWrite(led7, LOW);
```

```
digitalWrite(led8, LOW);
```

```
break;
```

case 3:

```
digitalWrite(led1, LOW); //Enciende el LED 3 y apaga los demas
```

```
digitalWrite(led2, LOW);
```

```
digitalWrite(led3, HIGH);
```

```
digitalWrite(led4, LOW);  
digitalWrite(led5, LOW);  
digitalWrite(led6, LOW);  
digitalWrite(led7, LOW);  
digitalWrite(led8, LOW);  
break;
```

case 4:

```
digitalWrite(led1, LOW); //Enciende el LED 4 y apaga los demas  
digitalWrite(led2, LOW);  
digitalWrite(led3, LOW);  
digitalWrite(led4, HIGH);  
digitalWrite(led5, LOW);  
digitalWrite(led6, LOW);  
digitalWrite(led7, LOW);  
digitalWrite(led8, LOW);  
break;
```

case 5:

```
digitalWrite(led1, LOW); //Enciende el LED 5 y apaga los demas  
digitalWrite(led2, LOW);  
digitalWrite(led3, LOW);  
digitalWrite(led4, LOW);  
digitalWrite(led5, HIGH);  
digitalWrite(led6, LOW);  
digitalWrite(led7, LOW);  
digitalWrite(led8, LOW);  
break;
```

case 6:

```
digitalWrite(led1, LOW); //Enciende el LED 6 y apaga los demas
digitalWrite(led2, LOW);
digitalWrite(led3, LOW);
digitalWrite(led4, LOW);
digitalWrite(led5, LOW);
digitalWrite(led6, HIGH);
digitalWrite(led7, LOW);
digitalWrite(led8, LOW);
break;
```

case 7:

```
digitalWrite(led1, LOW); //Enciende el LED 7 y apaga los demas
digitalWrite(led2, LOW);
digitalWrite(led3, LOW);
digitalWrite(led4, LOW);
digitalWrite(led5, LOW);
digitalWrite(led6, LOW);
digitalWrite(led7, HIGH);
digitalWrite(led8, LOW);
break;
```

case 8:

```
digitalWrite(led1, LOW); //Enciende el LED 8 y apaga los demas
digitalWrite(led2, LOW);
digitalWrite(led3, LOW);
digitalWrite(led4, LOW);
digitalWrite(led5, LOW);
digitalWrite(led6, LOW);
```

```

    digitalWrite(led7, LOW);

    digitalWrite(led8, HIGH);

    break;
}
}

void jugador2(void){
    //secuencia para aumentar el contador2
    if (digitalRead(avanzar2) == LOW){
        presionado2 = 1;//cambia de estado si se presiona el pulsador 2
    }

    if (digitalRead(avanzar2) == HIGH && presionado2 == 1){
        presionado2 = 0;//se reinicia la variable del antirrebote
        delay(100);
        cont2++;//el contador aumenta una unidad
    }

    //Casos posibles del jugador 2
    switch (cont2){
        case 1:
            digitalWrite(led9, HIGH); //Enciende el LED 9 y apaga los demas
            digitalWrite(led10, LOW);
            digitalWrite(led11, LOW);
            digitalWrite(led12, LOW);
            digitalWrite(led13, LOW);
            digitalWrite(led14, LOW);
            digitalWrite(led15, LOW);
            digitalWrite(led16, LOW);

            break;

        case 2:

```



```
digitalWrite(led9, LOW); //Enciende el LED 10 y apaga los demas
digitalWrite(led10, HIGH);
digitalWrite(led11, LOW);
digitalWrite(led12, LOW);
digitalWrite(led13, LOW);
digitalWrite(led14, LOW);
digitalWrite(led15, LOW);
digitalWrite(led16, LOW);
break;
```

case 3:

```
digitalWrite(led9, LOW); //Enciende el LED 11 y apaga los demas
digitalWrite(led10, LOW);
digitalWrite(led11, HIGH);
digitalWrite(led12, LOW);
digitalWrite(led13, LOW);
digitalWrite(led14, LOW);
digitalWrite(led15, LOW);
digitalWrite(led16, LOW);
break;
```

case 4:

```
digitalWrite(led9, LOW); //Enciende el LED 12 y apaga los demas
digitalWrite(led10, LOW);
digitalWrite(led11, LOW);
digitalWrite(led12, HIGH);
digitalWrite(led13, LOW);
digitalWrite(led14, LOW);
digitalWrite(led15, LOW);
```

```
digitalWrite(led16, LOW);
```

```
break;
```

case 5:

```
digitalWrite(led9, LOW); //Enciende el LED 13 y apaga los demas
```

```
digitalWrite(led10, LOW);
```

```
digitalWrite(led11, LOW);
```

```
digitalWrite(led12, LOW);
```

```
digitalWrite(led13, HIGH);
```

```
digitalWrite(led14, LOW);
```

```
digitalWrite(led15, LOW);
```

```
digitalWrite(led16, LOW);
```

```
break;
```

case 6:

```
digitalWrite(led9, LOW); //Enciende el LED 14 y apaga los demas
```

```
digitalWrite(led10, LOW);
```

```
digitalWrite(led11, LOW);
```

```
digitalWrite(led12, LOW);
```

```
digitalWrite(led13, LOW);
```

```
digitalWrite(led14, HIGH);
```

```
digitalWrite(led15, LOW);
```

```
digitalWrite(led16, LOW);
```

```
break;
```

case 7:

```
digitalWrite(led9, LOW); //Enciende el LED 15 y apaga los demas
```

```
digitalWrite(led10, LOW);
```

```
digitalWrite(led11, LOW);
```

```
digitalWrite(led12, LOW);
digitalWrite(led13, LOW);
digitalWrite(led14, LOW);
digitalWrite(led15, HIGH);
digitalWrite(led16, LOW);
break;

case 8:
digitalWrite(led9, LOW); //Enciende el LED 16 y apaga los demas
digitalWrite(led10, LOW);
digitalWrite(led11, LOW);
digitalWrite(led12, LOW);
digitalWrite(led13, LOW);
digitalWrite(led14, LOW);
digitalWrite(led15, LOW);
digitalWrite(led16, HIGH);
break;
}
}
```

```
void semaforo(void){
digitalWrite(ledR, HIGH);
digitalWrite(ledG, LOW);
delay(1000);
digitalWrite(ledR, HIGH);
digitalWrite(ledG, HIGH);
delay(1000);
digitalWrite(ledR, LOW);
digitalWrite(ledG, HIGH);
```

```

delay(1000);

digitalWrite(ledG, HIGH);

digitalWrite(ledB, HIGH);

}

void ganador(void){

if (cont>8){

    digitalWrite(led17, HIGH);//Indicador J1

    digitalWrite(led8, LOW);//Ultima led del J1

}

if (cont2>8){

    digitalWrite(led18, HIGH);//Indicador J2

    digitalWrite(led16, LOW);//Ultima led del J2

}

}

```

Diagrama de flujo:

