

Laboratorio # 4

Juego de carreras TivaC

Link del video:

https://youtu.be/XUzU6N9HT_s

Link del GitHub:

https://github.com/nataliadlb/LABS_REPOSITORIO.git

Pseudocódigo

```
// LEDS J1 //
```

```
const byte ledJ1_1 = 3;
```

```
const byte ledJ1_2 = 4;
```

```
const byte ledJ1_3 = 5;
```

```
const byte ledJ1_4 = 6;
```

```
const byte ledJ1_5 = 7;
```

```
const byte ledJ1_6 = 8;
```

```
const byte ledJ1_7 = 9;
```

```
const byte ledJ1_8 = 10;
```

```
const byte ledJ2_1 = 32;
```

```
const byte ledJ2_2 = 29;
```

```
const byte ledJ2_3 = 28;
```

```
const byte ledJ2_4 = 27;
```

```
const byte ledJ2_5 = 26;
```

```
const byte ledJ2_6 = 25;
```

```
const byte ledJ2_7 = 24;
```

```
const byte ledJ2_8 = 23;
```

```
const byte ledJ1_WIN = 19;
```

```
const byte ledJ2_WIN = 18;
```

```
// LEDS SEMAFORO //
```

```
const byte led_VERDE = GREEN_LED;
```

```
const byte led_AZUL = BLUE_LED;
```

```
const byte led_ROJO = RED_LED;
```

```
//PUSH INTERRUPTIONES //
```

```
const byte interruptPin = 2; //para activar  
SEMAFORO
```

```
const byte interruptPin2 = PUSH1;
```

```
const byte interruptPin3 = PUSH2;
```

```
// BANDERAS //
```

```
volatile byte flag_semaf = LOW;
```

```
volatile byte flag_BEGIN = LOW;
```

```

volatile byte flag_DONE = HIGH;

volatile byte flag_J1_WIN = LOW;

volatile byte flag_J2_WIN = LOW;


// CONTADORES JUGADORES //
uint8_t contJ1 = 0;

uint8_t contJ2 = 0;

uint8_t debouncing1 = 0; //Variable que
controla debouncing de un push
uint8_t debouncing2 = 0;


void setup() {

  pinMode(led_VERDE, OUTPUT);
  pinMode(led_AZUL, OUTPUT);
  pinMode(led_ROJO, OUTPUT);
  pinMode(interruptPin, INPUT_PULLUP);
  pinMode(interruptPin2, INPUT_PULLUP);
  pinMode(interruptPin3, INPUT_PULLUP);


  //leds//
  pinMode(ledJ1_1, OUTPUT);
  pinMode(ledJ1_2, OUTPUT);
  pinMode(ledJ1_3, OUTPUT);
  pinMode(ledJ1_4, OUTPUT);
  pinMode(ledJ1_5, OUTPUT);
  pinMode(ledJ1_6, OUTPUT);
  pinMode(ledJ1_7, OUTPUT);
  pinMode(ledJ1_8, OUTPUT);

  pinMode(ledJ2_1, OUTPUT);
  pinMode(ledJ2_2, OUTPUT);
  pinMode(ledJ2_3, OUTPUT);
  pinMode(ledJ2_4, OUTPUT);
  pinMode(ledJ2_5, OUTPUT);
  pinMode(ledJ2_6, OUTPUT);
  pinMode(ledJ2_7, OUTPUT);
  pinMode(ledJ2_8, OUTPUT);

  pinMode(ledJ1_WIN, OUTPUT);
  pinMode(ledJ2_WIN, OUTPUT);


  //Interrupciones//

  attachInterrupt(digitalPinToInterrupt(interruptPin), semaforo, FALLING); //interrupcion

  attachInterrupt(digitalPinToInterrupt(interruptPin2), J1, FALLING);

  attachInterrupt(digitalPinToInterrupt(interruptPin3), J2, FALLING);


  Serial.begin(9600);

}

void loop() {

  sec_semaforo();

  if (flag_BEGIN == HIGH){

```

```

int      Estado_PUSHJ1      =
digitalRead(interruptPin2);

int      Estado_PUSHJ2      =
digitalRead(interruptPin3);

if (debouncing1 == 1 && Estado_PUSHJ1 ==
0){
    contJ1++;
    debouncing1 = 0;
    //Serial.println(contJ1);
}

if (contJ1 > 0 && contJ1 <= 8){
    aumento_J1();
}
else{
    ganadorJ1();
}

if (debouncing2 == 1 && Estado_PUSHJ2 ==
0){
    contJ2++;
    debouncing2 = 0;
}

if (contJ2 > 0 && contJ2 <= 8){
    aumento_J2();
}
else{
    ganadorJ2();
}
}

```

```

void semaforo() { //INTERRUPCION
SEMAFORO
    if(flag_DONE == HIGH){
        flag_semaf = !flag_semaf;
    }
}

void J1() { //INTERRUPCION PUSH1
    if (flag_BEGIN == HIGH){
        debouncing1 = 1;
    }
}

void J2() { //INTERRUPCION PUSH2
    if (flag_BEGIN == HIGH){
        debouncing2 = 1;
    }
}

void sec_semaforo(){
    if (flag_semaf == HIGH){
        digitalWrite(ledJ2_WIN, LOW);
        digitalWrite(ledJ1_WIN, LOW);
        digitalWrite(led_ROJO, HIGH);
        digitalWrite(led_AZUL, LOW);
        digitalWrite(led_VERDE, LOW);
    }
}

```

```

Serial.print("3 ");
delay(1000);
digitalWrite(led_ROJO, LOW);
digitalWrite(led_AZUL, HIGH);
digitalWrite(led_VERDE, LOW);
Serial.print("2 ");
delay(1000);
digitalWrite(led_ROJO, LOW);
digitalWrite(led_AZUL, LOW);
digitalWrite(led_VERDE, HIGH);
Serial.print("1 ");
delay(1000);
digitalWrite(led_ROJO, LOW);
digitalWrite(led_AZUL, LOW);
digitalWrite(led_VERDE, LOW);
Serial.println("GO ");
flag_semaf = LOW;
flag_BEGIN = HIGH;
flag_DONE = LOW;
contJ1 = 0;
contJ2 = 0;
}
}

void aumento_J1(){
  if (contJ1 == 1){
    digitalWrite(ledJ1_1, HIGH);
    //Serial.println(contJ1);
  }
  else if (contJ1 == 2){
    digitalWrite(ledJ1_2, HIGH);
    //Serial.println(contJ1);
  }
  else if (contJ1 == 3){
    digitalWrite(ledJ1_3, HIGH);
    //Serial.println(contJ1);
  }
  else if (contJ1 == 4){
    digitalWrite(ledJ1_4, HIGH);
    //Serial.println(contJ1);
  }
  else if (contJ1 == 5){
    digitalWrite(ledJ1_5, HIGH);
    //Serial.println(contJ1);
  }
  else if (contJ1 == 6){
    digitalWrite(ledJ1_6, HIGH);
    //Serial.println(contJ1);
  }
  else if (contJ1 == 7){
    digitalWrite(ledJ1_7, HIGH);
    //Serial.println(contJ1);
  }
  else{
    digitalWrite(ledJ1_8, HIGH);
    //Serial.println(contJ1);
  }
}

```

```

delay(500);
digitalWrite(ledJ1_1, LOW);
digitalWrite(ledJ1_2, LOW);
digitalWrite(ledJ1_3, LOW);
digitalWrite(ledJ1_4, LOW);
digitalWrite(ledJ1_5, LOW);
digitalWrite(ledJ1_6, LOW);
digitalWrite(ledJ1_7, LOW);
digitalWrite(ledJ1_8, LOW);
contJ1 = 0;
//Serial.println(contJ1);
flag_J1_WIN = HIGH;
}
}

void aumento_J2(){
  if (contJ2 == 1){
    digitalWrite(ledJ2_1, HIGH);
    //Serial.println(contJ2);
  }
  else if (contJ2 == 2){
    digitalWrite(ledJ2_2, HIGH);
    //Serial.println(contJ2);
  }
  else if (contJ2 == 3){
    digitalWrite(ledJ2_3, HIGH);
    //Serial.println(contJ2);
  }
  else if (contJ2 == 4){
    digitalWrite(ledJ2_4, HIGH);
    //Serial.println(contJ2);
  }
  else if (contJ2 == 5){
    digitalWrite(ledJ2_5, HIGH);
    //Serial.println(contJ2);
  }
  else if (contJ2 == 6){
    digitalWrite(ledJ2_6, HIGH);
    //Serial.println(contJ2);
  }
  else if (contJ2 == 7){
    digitalWrite(ledJ2_7, HIGH);
    //Serial.println(contJ2);
  }
  else{
    digitalWrite(ledJ2_8, HIGH);
    //Serial.println(contJ2);
    delay(500);
    digitalWrite(ledJ2_1, LOW);
    digitalWrite(ledJ2_2, LOW);
    digitalWrite(ledJ2_3, LOW);
    digitalWrite(ledJ2_4, LOW);
    digitalWrite(ledJ2_5, LOW);
    digitalWrite(ledJ2_6, LOW);
    digitalWrite(ledJ2_7, LOW);
    digitalWrite(ledJ2_8, LOW);
    contJ2 = 0;
    flag_J2_WIN = HIGH;
  }
}

```

```

}
}

```

```

void ganadorJ1(){
if(flag_J1_WIN == HIGH){
    digitalWrite(ledJ1_WIN, HIGH);
    digitalWrite(ledJ2_WIN, LOW);
    digitalWrite(ledJ2_1, LOW);
    digitalWrite(ledJ2_2, LOW);
    digitalWrite(ledJ2_3, LOW);
    digitalWrite(ledJ2_4, LOW);
    digitalWrite(ledJ2_5, LOW);
    digitalWrite(ledJ2_6, LOW);
    digitalWrite(ledJ2_7, LOW);
    digitalWrite(ledJ2_8, LOW);
    flag_J1_WIN = LOW;
    flag_J2_WIN = LOW;
    contJ1 = 0;
    contJ2 = 0;
    flag_DONE = HIGH;
    flag_BEGIN = LOW;
    Serial.println("---GANADOR---");
    Serial.println("  J1  ");
}
}

```

```

}

```

```

void ganadorJ2(){
    if(flag_J2_WIN == HIGH){
        digitalWrite(ledJ2_WIN, HIGH);
        digitalWrite(ledJ1_WIN, LOW);
        digitalWrite(ledJ1_1, LOW);
        digitalWrite(ledJ1_2, LOW);
        digitalWrite(ledJ1_3, LOW);
        digitalWrite(ledJ1_4, LOW);
        digitalWrite(ledJ1_5, LOW);
        digitalWrite(ledJ1_6, LOW);
        digitalWrite(ledJ1_7, LOW);
        digitalWrite(ledJ1_8, LOW);
        flag_J1_WIN = LOW;
        flag_J2_WIN = LOW;
        contJ1 = 0;
        contJ2 = 0;
        flag_DONE = HIGH;
        flag_BEGIN = LOW;
        Serial.println("---GANADOR---");
        Serial.println("  J2  ");
    }
}

```

Código

/*

Laboratorio # 4

Natalia de León Bercián

carné: 18193

Digital 2

Marzo 2021

*/

// LEDS J1 //

```
const byte ledJ1_1 = 3;
const byte ledJ1_2 = 4;
const byte ledJ1_3 = 5;
const byte ledJ1_4 = 6;
const byte ledJ1_5 = 7;
const byte ledJ1_6 = 8;
const byte ledJ1_7 = 9;
const byte ledJ1_8 = 10;
```

```
const byte ledJ2_1 = 32;
const byte ledJ2_2 = 29;
const byte ledJ2_3 = 28;
const byte ledJ2_4 = 27;
const byte ledJ2_5 = 26;
const byte ledJ2_6 = 25;
const byte ledJ2_7 = 24;
const byte ledJ2_8 = 23;
```

```
const byte ledJ1_WIN = 19;
const byte ledJ2_WIN = 18;
```

// LEDS SEMAFORO //

```
const byte led_VERDE = GREEN_LED;
const byte led_AZUL = BLUE_LED;
```

```
const byte led_ROJO = RED_LED;
```

//PUSH INTERRUPTIONES //

```
const byte interruptPin = 2; //para activar
SEMAFORO
```

```
const byte interruptPin2 = PUSH1;
const byte interruptPin3 = PUSH2;
```

// BANDERAS //

```
volatile byte flag_semaf = LOW;
volatile byte flag_BEGIN = LOW;
volatile byte flag_DONE = HIGH;
volatile byte flag_J1_WIN = LOW;
volatile byte flag_J2_WIN = LOW;
```

// CONTADORES JUGADORES //

```
uint8_t contJ1 = 0;
uint8_t contJ2 = 0;
uint8_t debouncing1 = 0; //Variable que
controla debouncing de un push
uint8_t debouncing2 = 0;
```

void setup() {

```
    pinMode(led_VERDE, OUTPUT);
    pinMode(led_AZUL, OUTPUT);
    pinMode(led_ROJO, OUTPUT);
    pinMode(interruptPin, INPUT_PULLUP);
    pinMode(interruptPin2, INPUT_PULLUP);
    pinMode(interruptPin3, INPUT_PULLUP);
```

```

//leds//
pinMode(ledJ1_1, OUTPUT);
pinMode(ledJ1_2, OUTPUT);
pinMode(ledJ1_3, OUTPUT);
pinMode(ledJ1_4, OUTPUT);
pinMode(ledJ1_5, OUTPUT);
pinMode(ledJ1_6, OUTPUT);
pinMode(ledJ1_7, OUTPUT);
pinMode(ledJ1_8, OUTPUT);

pinMode(ledJ2_1, OUTPUT);
pinMode(ledJ2_2, OUTPUT);
pinMode(ledJ2_3, OUTPUT);
pinMode(ledJ2_4, OUTPUT);
pinMode(ledJ2_5, OUTPUT);
pinMode(ledJ2_6, OUTPUT);
pinMode(ledJ2_7, OUTPUT);
pinMode(ledJ2_8, OUTPUT);

pinMode(ledJ1_WIN, OUTPUT);
pinMode(ledJ2_WIN, OUTPUT);

//Interrupciones//

attachInterrupt(digitalPinToInterrupt(interruptPin), semaforo, FALLING); //interrupcion
semaforo

attachInterrupt(digitalPinToInterrupt(interruptPin2), J1, FALLING); //interrupción J1

attachInterrupt(digitalPinToInterrupt(interruptPin3), J2, FALLING); //interrupción J2

Serial.begin(9600);
}

void loop() {
    sec_semaforo();
    if (flag_BEGIN == HIGH){

        if (contJ1 > 0 && contJ1 <= 8){ //Que se
mantenga dentro de los 8 bits J1
            aumento_J1(); //Encender los leds del
J1
        }
        else{
            ganadorJ1(); //Cuando gana J1
        }

        if (contJ2 > 0 && contJ2 <= 8){ //Que se
mantenga dentro de los 8 bits J2
            aumento_J2(); //Encender los leds del
J2
        }
        else{
            ganadorJ2(); //Cuando gana J2
        }
    }
}

```



```
void semaforo() { //INTERRUPCION  
SEMAFORO
```

```
    if(flag_DONE == HIGH){  
        flag_semaf = !flag_semaf;  
    }  
}
```

```
void J1() { //INTERRUPCION PUSH1
```

```
    if (flag_BEGIN == HIGH){  
        int Estado_PUSHJ1 =  
digitalRead(interruptPin2);  
        if (Estado_PUSHJ1 == 0){  
            contJ1++;  
        }  
    }  
}
```

```
void J2() { //INTERRUPCION PUSH2
```

```
    if (flag_BEGIN == HIGH){  
        int Estado_PUSHJ2 =  
digitalRead(interruptPin3);  
        if (Estado_PUSHJ2 == 0){  
            contJ2++;  
        }  
    }  
}
```

```
void sec_semaforo(){
```

```
    if (flag_semaf == HIGH){  
        digitalWrite(ledJ2_WIN, LOW);
```

```
digitalWrite(ledJ1_WIN, LOW);  
digitalWrite(led_ROJO, HIGH);  
digitalWrite(led_AZUL, LOW);  
digitalWrite(led_VERDE, LOW);  
Serial.print("3 ");
```

```
delay(1000);  
digitalWrite(led_ROJO, LOW);  
digitalWrite(led_AZUL, HIGH);  
digitalWrite(led_VERDE, LOW);  
Serial.print("2 ");
```

```
delay(1000);  
digitalWrite(led_ROJO, LOW);  
digitalWrite(led_AZUL, LOW);  
digitalWrite(led_VERDE, HIGH);  
Serial.print("1 ");
```

```
delay(1000);  
digitalWrite(led_ROJO, LOW);  
digitalWrite(led_AZUL, LOW);  
digitalWrite(led_VERDE, LOW);  
Serial.println("GO ");
```

```
flag_semaf = LOW;  
flag_BEGIN = HIGH;  
flag_DONE = LOW;  
contJ1 = 0;  
contJ2 = 0;
```

```
}
```

```
}
```

```

void aumento_J1(){
    if (contJ1 == 1){
        digitalWrite(ledJ1_1, HIGH);
        //Serial.println(contJ1);
    }
    else if (contJ1 == 2){
        digitalWrite(ledJ1_2, HIGH);
        //Serial.println(contJ1);
    }
    else if (contJ1 == 3){
        digitalWrite(ledJ1_3, HIGH);
        //Serial.println(contJ1);
    }
    else if (contJ1 == 4){
        digitalWrite(ledJ1_4, HIGH);
        //Serial.println(contJ1);
    }
    else if (contJ1 == 5){
        digitalWrite(ledJ1_5, HIGH);
        //Serial.println(contJ1);
    }
    else if (contJ1 == 6){
        digitalWrite(ledJ1_6, HIGH);
        //Serial.println(contJ1);
    }
    else if (contJ1 == 7){
        digitalWrite(ledJ1_7, HIGH);
        //Serial.println(contJ1);
    }
    else{
        digitalWrite(ledJ1_8, HIGH);
        //Serial.println(contJ1);
        delay(500);
        digitalWrite(ledJ1_1, LOW);
        digitalWrite(ledJ1_2, LOW);
        digitalWrite(ledJ1_3, LOW);
        digitalWrite(ledJ1_4, LOW);
        digitalWrite(ledJ1_5, LOW);
        digitalWrite(ledJ1_6, LOW);
        digitalWrite(ledJ1_7, LOW);
        digitalWrite(ledJ1_8, LOW);
        contJ1 = 0;
        //Serial.println(contJ1);
        flag_J1_WIN = HIGH;
    }
}

void aumento_J2(){
    if (contJ2 == 1){
        digitalWrite(ledJ2_1, HIGH);
        //Serial.println(contJ2);
    }
    else if (contJ2 == 2){
        digitalWrite(ledJ2_2, HIGH);
        //Serial.println(contJ2);
    }
    else if (contJ2 == 3){
        digitalWrite(ledJ2_3, HIGH);

```

```

//Serial.println(contJ2);
}
else if (contJ2 == 4){
    digitalWrite(ledJ2_4, HIGH);
    //Serial.println(contJ2);
}
else if (contJ2 == 5){
    digitalWrite(ledJ2_5, HIGH);
    //Serial.println(contJ2);
}
else if (contJ2 == 6){
    digitalWrite(ledJ2_6, HIGH);
    //Serial.println(contJ2);
}
else if (contJ2 == 7){
    digitalWrite(ledJ2_7, HIGH);
    //Serial.println(contJ2);
}
else{
    digitalWrite(ledJ2_8, HIGH);
    //Serial.println(contJ2);
    delay(500);
    digitalWrite(ledJ2_1, LOW);
    digitalWrite(ledJ2_2, LOW);
    digitalWrite(ledJ2_3, LOW);
    digitalWrite(ledJ2_4, LOW);
    digitalWrite(ledJ2_5, LOW);
    digitalWrite(ledJ2_6, LOW);
    digitalWrite(ledJ2_7, LOW);

```

```

digitalWrite(ledJ2_8, LOW);
contJ2 = 0;
flag_J2_WIN = HIGH;
}
}

void ganadorJ1(){
    if(flag_J1_WIN == HIGH){
        digitalWrite(ledJ1_WIN, HIGH);
        digitalWrite(ledJ2_WIN, LOW);
        digitalWrite(ledJ2_1, LOW);
        digitalWrite(ledJ2_2, LOW);
        digitalWrite(ledJ2_3, LOW);
        digitalWrite(ledJ2_4, LOW);
        digitalWrite(ledJ2_5, LOW);
        digitalWrite(ledJ2_6, LOW);
        digitalWrite(ledJ2_7, LOW);
        digitalWrite(ledJ2_8, LOW);
        flag_J1_WIN = LOW;
        flag_J2_WIN = LOW;
        contJ1 = 0;
        contJ2 = 0;
        flag_DONE = HIGH;
        flag_BEGIN = LOW;
        Serial.println("----GANADOR----");
        Serial.println("    J1    ");
    }
}

```

```
void ganadorJ2(){
  if(flag_J2_WIN == HIGH){
    digitalWrite(ledJ2_WIN, HIGH);
    digitalWrite(ledJ1_WIN, LOW);
    digitalWrite(ledJ1_1, LOW);
    digitalWrite(ledJ1_2, LOW);
    digitalWrite(ledJ1_3, LOW);
    digitalWrite(ledJ1_4, LOW);
    digitalWrite(ledJ1_5, LOW);
    digitalWrite(ledJ1_6, LOW);
    digitalWrite(ledJ1_7, LOW);
    digitalWrite(ledJ1_8, LOW);
    flag_J1_WIN = LOW;
    flag_J2_WIN = LOW;
    contJ1 = 0;
    contJ2 = 0;
    flag_DONE = HIGH;
    flag_BEGIN = LOW;
    Serial.println("---GANADOR---");
    Serial.println("  J2  ");
  }
}
```