Universidad del Valle de Guatemala Cristopher René Sagastume 18640  
Electrónica Digital 2  
Sección 30 21/03/2021

**Pseudocódigo**

* Inicio
* Verificar si alguno de los botones de inicio fue presionado
  + sí fue presionado verificamos cual fue presionado y le damos valor a una bandera y volvemos a verificar esta vez si dejaron de presionar (anti rebote)
    - sí fue presionado inicia conteo de semáforo, sino no pasa nada.
      * iniciamos un ciclo que se cumplirá mientras una bandera tenga un valor especifico
        + revisamos si el botón de jugador 1 fue presionado con el respectivo anti rebote implementado

sí fue presionado verificamos si el contador del jugador 1 es =8

si fuese = 8 encendemos el led bandera que indica que el jugador 1 gano y cambiamos el valor de la bandera del ciclo while para que la carrera pueda iniciar de nuevo

si no fuese =8 aumentamos un contador en 1 y llamamos una función que encenderá los leds en formato de década.

* + - * + Revisamos si el botón del jugador 2 fue presionado con el respectivo anti rebote implementado

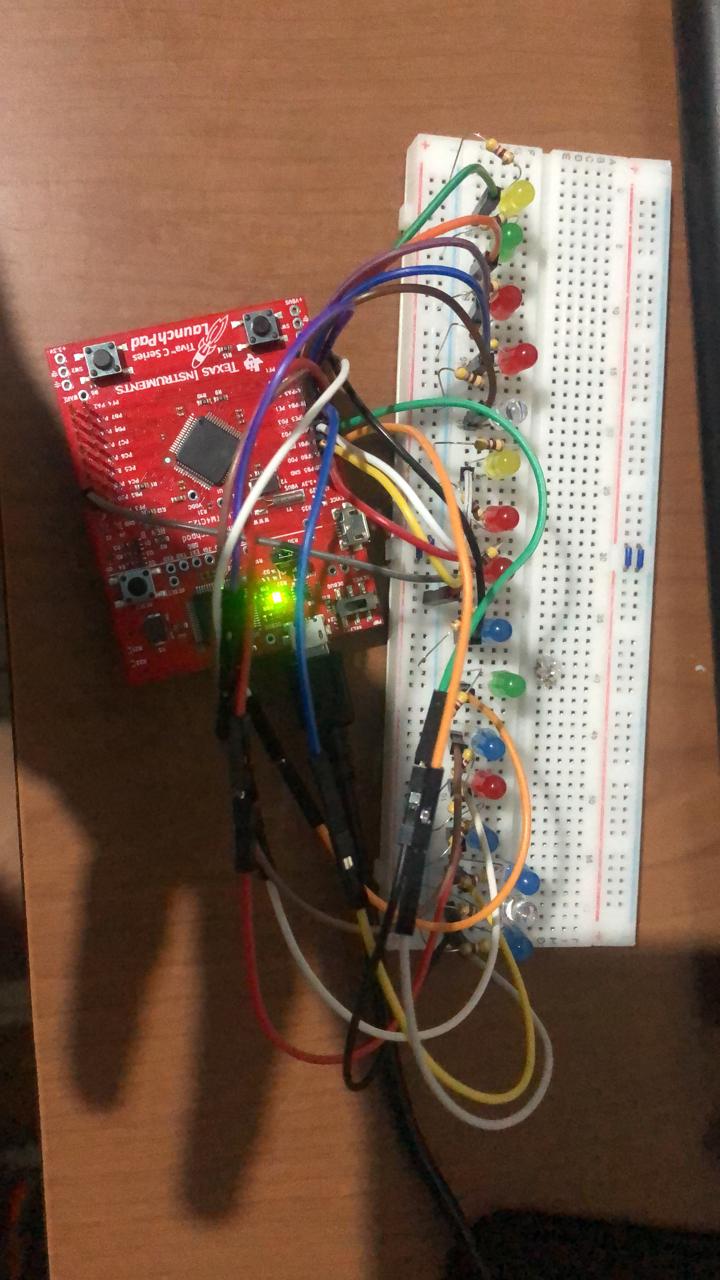
Si fue presionado verificamos si el contador del jugador 2 es =8

Si fuese =8 encendemos el led bandera que indica que el jugador 2 gano y cambiamos el valor de la bandera del ciclo while para que la carrera pueda iniciar de nuevo

Si no fuese =8 aumentamos el contador del jugador 2 en 1 y llamamos a la función que encenderá los leds en formato de década.

* Fin.

**Foto del circuito**

****

**Código comentado**

//Cristopher Sagastume 18640

//Lab4\_Carrera

//21/03/2021

// Creación de variables y se definen los Led RGB

#define LED RED\_LED

#define LED1 GREEN\_LED

const int SW1 = PUSH1;

const int SW2 = PUSH2;

int SW1State = 0;

int SW2State = 0;

char flag = 0;

char flagJ1 = 0;

char flagJ2 = 0;

char flagJuego = 0;

char count = 0;

char count2 = 0;

// Se configuran los pines y los led como salidas y botones como PULLUP

void setup() {

pinMode(LED, OUTPUT);

pinMode(LED1, OUTPUT);

pinMode(SW1, INPUT\_PULLUP);

pinMode(SW2, INPUT\_PULLUP);

pinMode(PE\_3, OUTPUT);

pinMode(PE\_2, OUTPUT);

pinMode(PE\_1, OUTPUT);

pinMode(PD\_3, OUTPUT);

pinMode(PD\_2, OUTPUT);

pinMode(PD\_1, OUTPUT);

pinMode(PD\_0, OUTPUT);

pinMode(PE\_0, OUTPUT);

pinMode(PA\_7, OUTPUT);

pinMode(PA\_6, OUTPUT);

pinMode(PA\_5, OUTPUT);

pinMode(PB\_4, OUTPUT);

pinMode(PE\_5, OUTPUT);

pinMode(PE\_4, OUTPUT);

pinMode(PB\_1, OUTPUT);

pinMode(PB\_0, OUTPUT);

}

// Rutina de loop infinito

void loop() {

//lectura del estado del boton

SW1State = digitalRead(SW1);

SW2State = digitalRead(SW2);

flag = 0;

count = 0;

count2 = 0;

J1();

J2();

flagJ1 = 0;

flagJ2 = 0;

//antirebote para semaforo

if (SW1State == LOW or SW2State == LOW) {

if (SW1State == LOW) {

flagJ1 = 1;

}

else if (SW2State == LOW) {

flagJ2 = 1;

}

while (flag == 0) {

SW1State = digitalRead(SW1);

SW2State = digitalRead(SW2);

if (flagJ1 == 1 and SW1State == HIGH) {

flag = 1;

}

else if (flagJ2 == 1 and SW2State == HIGH) {

flag = 1;

}

}

semaforo();

flagJuego = 0;

//ciclo del juego

while (flagJuego == 0) {

flag = 0;

SW1State = digitalRead(SW1);

SW2State = digitalRead(SW2);

//antirebote del jugador 1

if (SW1State == LOW) {

while (flag == 0) {

SW1State = digitalRead(SW1);

SW2State = digitalRead(SW2);

if (SW1State == HIGH) {

flag = 1;

}

}

//aumento de contador y encender leds

count++;

J1();

//revisión si ganó

if (count == 8) {

digitalWrite(LED, HIGH);

delay(2000);

digitalWrite(LED, LOW);

flagJuego = 1;

}

}

//antirebote jugador 2

else if (SW2State == LOW) {

while (flag == 0) {

SW1State = digitalRead(SW1);

SW2State = digitalRead(SW2);

if (SW2State == HIGH) {

flag = 1;

}

}

//aumento de contador jugador 2 y encendido de leds

count2++;

J2();

//revisión si ganó jugador 2

if (count2 == 8) {

digitalWrite(LED1, HIGH);

delay(2000);

digitalWrite(LED1, LOW);

flagJuego = 1;

}

}

}

}

}

//formato de semaforo ROJO-AMARILLO-VERDE

void semaforo() {

digitalWrite(LED, HIGH);

delay(1000);

digitalWrite(LED, LOW);

delay(1000);

digitalWrite(LED1, HIGH);

digitalWrite(LED, HIGH);

delay(1000);

digitalWrite(LED, LOW);

digitalWrite(LED1, LOW);

delay(1000);

digitalWrite(LED1, HIGH);

delay(1000);

digitalWrite(LED1, LOW);

delay(1000);

}

void J1() {

//encender leds por decada jugador 1

switch (count) {

case 0:

digitalWrite(PE\_3, LOW);

digitalWrite(PE\_2, LOW);

digitalWrite(PE\_1, LOW);

digitalWrite(PD\_3, LOW);

digitalWrite(PD\_2, LOW);

digitalWrite(PD\_1, LOW);

digitalWrite(PD\_0, LOW);

digitalWrite(PE\_0, LOW);

break;

case 1:

digitalWrite(PE\_3, HIGH);

digitalWrite(PE\_2, LOW);

digitalWrite(PE\_1, LOW);

digitalWrite(PD\_3, LOW);

digitalWrite(PD\_2, LOW);

digitalWrite(PD\_1, LOW);

digitalWrite(PD\_0, LOW);

digitalWrite(PE\_0, LOW);

break;

case 2:

digitalWrite(PE\_3, LOW);

digitalWrite(PE\_2, HIGH);

digitalWrite(PE\_1, LOW);

digitalWrite(PD\_3, LOW);

digitalWrite(PD\_2, LOW);

digitalWrite(PD\_1, LOW);

digitalWrite(PD\_0, LOW);

digitalWrite(PE\_0, LOW);

break;

case 3:

digitalWrite(PE\_3, LOW);

digitalWrite(PE\_2, LOW);

digitalWrite(PE\_1, HIGH);

digitalWrite(PD\_3, LOW);

digitalWrite(PD\_2, LOW);

digitalWrite(PD\_1, LOW);

digitalWrite(PD\_0, LOW);

digitalWrite(PE\_0, LOW);

break;

case 4:

digitalWrite(PE\_3, LOW);

digitalWrite(PE\_2, LOW);

digitalWrite(PE\_1, LOW);

digitalWrite(PD\_3, HIGH);

digitalWrite(PD\_2, LOW);

digitalWrite(PD\_1, LOW);

digitalWrite(PD\_0, LOW);

digitalWrite(PE\_0, LOW);

break;

case 5:

digitalWrite(PE\_3, LOW);

digitalWrite(PE\_2, LOW);

digitalWrite(PE\_1, LOW);

digitalWrite(PD\_3, LOW);

digitalWrite(PD\_2, HIGH);

digitalWrite(PD\_1, LOW);

digitalWrite(PD\_0, LOW);

digitalWrite(PE\_0, LOW);

break;

case 6:

digitalWrite(PE\_3, LOW);

digitalWrite(PE\_2, LOW);

digitalWrite(PE\_1, LOW);

digitalWrite(PD\_3, LOW);

digitalWrite(PD\_2, LOW);

digitalWrite(PD\_1, HIGH);

digitalWrite(PD\_0, LOW);

digitalWrite(PE\_0, LOW);

break;

case 7:

digitalWrite(PE\_3, LOW);

digitalWrite(PE\_2, LOW);

digitalWrite(PE\_1, LOW);

digitalWrite(PD\_3, LOW);

digitalWrite(PD\_2, LOW);

digitalWrite(PD\_1, LOW);

digitalWrite(PD\_0, HIGH);

digitalWrite(PE\_0, LOW);

break;

case 8:

digitalWrite(PE\_3, LOW);

digitalWrite(PE\_2, LOW);

digitalWrite(PE\_1, LOW);

digitalWrite(PD\_3, LOW);

digitalWrite(PD\_2, LOW);

digitalWrite(PD\_1, LOW);

digitalWrite(PD\_0, LOW);

digitalWrite(PE\_0, HIGH);

break;

}

}

void J2() {

//encender leds por decada jugador 2

switch (count2) {

case 0:

digitalWrite(PA\_7, LOW);

digitalWrite(PA\_6, LOW);

digitalWrite(PA\_5, LOW);

digitalWrite(PB\_4, LOW);

digitalWrite(PE\_5, LOW);

digitalWrite(PE\_4, LOW);

digitalWrite(PB\_1, LOW);

digitalWrite(PB\_0, LOW);

break;

case 1:

digitalWrite(PA\_7, HIGH);

digitalWrite(PA\_6, LOW);

digitalWrite(PA\_5, LOW);

digitalWrite(PB\_4, LOW);

digitalWrite(PE\_5, LOW);

digitalWrite(PE\_4, LOW);

digitalWrite(PB\_1, LOW);

digitalWrite(PB\_0, LOW);

break;

case 2:

digitalWrite(PA\_7, LOW);

digitalWrite(PA\_6, HIGH);

digitalWrite(PA\_5, LOW);

digitalWrite(PB\_4, LOW);

digitalWrite(PE\_5, LOW);

digitalWrite(PE\_4, LOW);

digitalWrite(PB\_1, LOW);

digitalWrite(PB\_0, LOW);

break;

case 3:

digitalWrite(PA\_7, LOW);

digitalWrite(PA\_6, LOW);

digitalWrite(PA\_5, HIGH);

digitalWrite(PB\_4, LOW);

digitalWrite(PE\_5, LOW);

digitalWrite(PE\_4, LOW);

digitalWrite(PB\_1, LOW);

digitalWrite(PB\_0, LOW);

break;

case 4:

digitalWrite(PA\_7, LOW);

digitalWrite(PA\_6, LOW);

digitalWrite(PA\_5, LOW);

digitalWrite(PB\_4, HIGH);

digitalWrite(PE\_5, LOW);

digitalWrite(PE\_4, LOW);

digitalWrite(PB\_1, LOW);

digitalWrite(PB\_0, LOW);

break;

case 5:

digitalWrite(PA\_7, LOW);

digitalWrite(PA\_6, LOW);

digitalWrite(PA\_5, LOW);

digitalWrite(PB\_4, LOW);

digitalWrite(PE\_5, HIGH);

digitalWrite(PE\_4, LOW);

digitalWrite(PB\_1, LOW);

digitalWrite(PB\_0, LOW);

break;

case 6:

digitalWrite(PA\_7, LOW);

digitalWrite(PA\_6, LOW);

digitalWrite(PA\_5, LOW);

digitalWrite(PB\_4, LOW);

digitalWrite(PE\_5, LOW);

digitalWrite(PE\_4, HIGH);

digitalWrite(PB\_1, LOW);

digitalWrite(PB\_0, LOW);

break;

case 7:

digitalWrite(PA\_7, LOW);

digitalWrite(PA\_6, LOW);

digitalWrite(PA\_5, LOW);

digitalWrite(PB\_4, LOW);

digitalWrite(PE\_5, LOW);

digitalWrite(PE\_4, LOW);

digitalWrite(PB\_1, HIGH);

digitalWrite(PB\_0, LOW);

break;

case 8:

digitalWrite(PA\_7, LOW);

digitalWrite(PA\_6, LOW);

digitalWrite(PA\_5, LOW);

digitalWrite(PB\_4, LOW);

digitalWrite(PE\_5, LOW);

digitalWrite(PE\_4, LOW);

digitalWrite(PB\_1, LOW);

digitalWrite(PB\_0, HIGH);

break;

}

}

**Link del repositorio**

<https://github.com/sag18640/Electronica_Digital_2.git>