**Para o PIC12F629**

**Código para compilador mikroC**

//

// Apelidos dos pinos =======================

//

sbit start at gp5\_bit; //

sbit porta at gp4\_bit; // parte específica

sbit lamp at gp0\_bit; // para o PIC 12F629

sbit led at gp1\_bit; //

//

// Variáveis Globais ========================

//

int cont; // auxiliar para contagens

//

// Função pisca\_led =========================

//

void pisca\_led() {

led = 1;

delay\_ms(100);

led = porta;

delay\_ms(200);

led = 1;

delay\_ms(100);

led = porta;

delay\_ms(100);

led = 0;

delay\_ms(500);

}

//

// Função validar\_start =====================

//

void validar\_start() {

cont = 0;

while (cont<300) {

cont = cont + 1;

delay\_ms(10);

if (!start || !porta) {

cont = 0;

}

}

}

//

// Função temporizar ========================

//

void temporizar() {

cont = 0;

while (cont<30) {

cont = cont + porta;

lamp = porta;

pisca\_led(); // chama função pisca\_led()

}

lamp = 0;

led = 1;

}

//

// Função Setup =============================

//

void setup() {

intcon = 0x00; // específica

cmcon = 0x07; // para o

trisio = 0b111100; // PIC 12F629

led = 0;

lamp = 0;

cont = 0;

}

//

// Função loop ==============================

//

void loop() {

validar\_start(); // chama função validar\_start()

temporizar(); // chama função temporizar()

}

//

// Função main - principal ==================

//

void main() {

setup(); // chama função setup()

while (1) {

loop(); // chama função loop()

}

}

**Para o Arduino UNO ou ESP32**

**Código para IDE Arduino e TinkerCad**

**Obs.: para ESP32 tem que ajustar os números dos pinos**

//

// Apelidos dos pinos =======================

//

byte start = 2; //

byte porta = 3; // parte específica

byte lamp = 4; // para o Arduino UNO

byte led = 5; //

//

// Variáveis Globais ========================

//

int cont; // auxiliar para contagens

//

// Função pisca\_led =========================

//

void pisca\_led() {

digitalWrite(led, HIGH);

delay(100);

digitalWrite(led, digitalRead(porta));

delay(200);

digitalWrite(led, HIGH);

delay(100);

digitalWrite(led, digitalRead(porta));

delay(100);

digitalWrite(led, LOW);

delay(500);

}

//

// Função validar\_start =====================

//

void validar\_start() {

cont = 0;

while (cont<300) {

cont = cont + 1;

delay(10);

if (!digitalRead(start) || !digitalRead(porta)) {

cont = 0;

}

}

}

//

// Função temporizar ========================

//

void temporizar() {

cont = 0;

while (cont<30) {

cont = cont + digitalRead(porta);

digitalWrite(lamp, digitalRead(porta));

pisca\_led(); // chama função pisca\_led()

}

digitalWrite(lamp, LOW);

digitalWrite(led, HIGH);

}

//

// Função Setup =============================

//

void setup() {

pinMode(start, INPUT); // específico

pinMode(porta, INPUT); // para

pinMode(lamp, OUTPUT); // o

pinMode(led, OUTPUT); // Arduino

digitalWrite(led, LOW);

digitalWrite(lamp, LOW);

digitalWrite(cont, LOW);

}

//

// Função loop ==============================

//

void loop() {

validar\_start(); // chama função validar\_start()

temporizar(); // chama função temporizar()

}

//

// Função main - principal ==================

//

Não requer função main, já está implícita na compilação