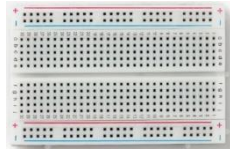


Componentes envolvidos:

Led Vermelho



Breadboard



Resistência



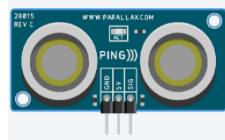
Arduino



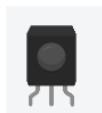
Cabos



Sensor ultrassônico



Sensor infravermelho

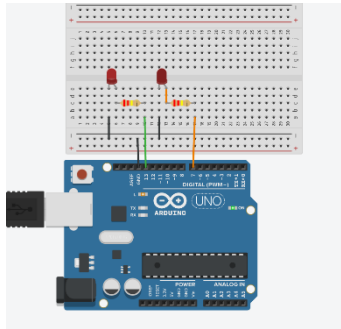


Comando infravermelho



Exercício 1

Montagem do circuito e respectivos testes:



Código Utilizado:

```
unsigned long previousMillisLED13=0;
unsigned long previousMillisLED7=0;

int intervalLED13 = 200;
int intervalLED7 = 400;

boolean LED7state = false;
boolean LED13state = false;

void setup() {
  pinMode(7, OUTPUT);
  pinMode(13, OUTPUT);
}

void loop() {
  unsigned long currentMillis = millis();

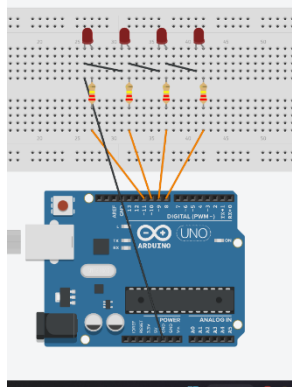
  if ((unsigned long) (currentMillis -
previousMillisLED13) >= intervalLED13) {
    LED13state = !LED13state;
    digitalWrite(13, LED13state);
    previousMillisLED13 = currentMillis;
  }

  if ((unsigned long) (currentMillis -
previousMillisLED7) >= intervalLED7) {
    LED7state = !LED7state;
    digitalWrite(7, LED7state);
    previousMillisLED7 = currentMillis;
  }
}
```

Neste exercício os LEDs tem um efeito de timer que consoante um certo intervalo de tempo os LEDs vão alternando entre ligado e desligado

Exercício 2

Montagem do circuito e respectivos testes:



Código Utilizado:

```
#include <Arduino_FreeRTOS.h>
void setup()
{
  Serial.begin(9600);
  Serial.println(F("In Setup function"));
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);

  xTaskCreate(MyTask1, "Task1", 100, NULL, 1, NULL);
  xTaskCreate(MyTask2, "Task2", 100, NULL, 2, NULL);
  xTaskCreate(MyTask3, "Task3", 100, NULL, 3, NULL);
  xTaskCreate(MyIdleTask, "IdleTask", 100, NULL, 0, NULL);
}

void loop()
{
  static void MyTask1(void* pvParameters)
  {
    while(1)
    {
      digitalWrite(8, HIGH);
      digitalWrite(9, LOW);
      digitalWrite(10, LOW);
      digitalWrite(11, LOW);
      Serial.println(F("Task1"));
      vTaskDelay(100/portTICK_PERIOD_MS);
    }
  }
}
```

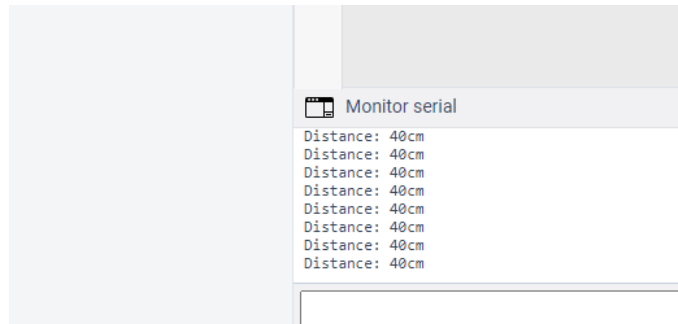
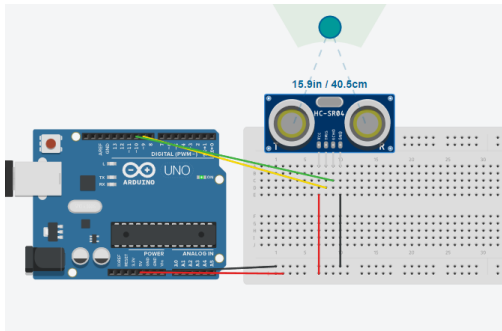
```
static void MyTask3(void* pvParameters)
{
  while(1)
  {
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    digitalWrite(10, HIGH);
    digitalWrite(11, LOW);
    Serial.println(F("Task3"));
    vTaskDelay(110/portTICK_PERIOD_MS);
  }
}

static void MyIdleTask(void* pvParameters)
{
  while(1)
  {
    digitalWrite(8, LOW);
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    digitalWrite(11, HIGH);
    Serial.println(F("Idle state"));
    delay(50);
  }
}
```

Neste exercício usamos a biblioteca FreeRTOS do Arduino para poder fazer 3 tipos diferentes de tarefas com os LEDs montados no breadboard.

Exercício 3

Montagem do circuito e respectivos testes:



Código Utilizado:

```
const int trigPin = 9;
const int echoPin = 10;
long duration;
int distance;

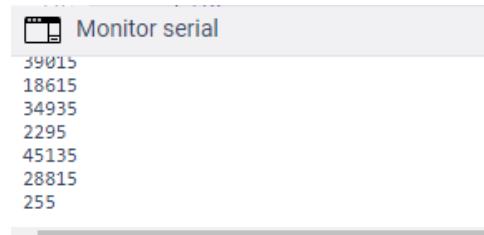
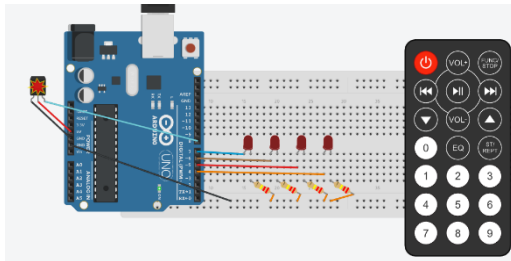
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration*0.034/2;
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println("cm");
}
```

Este exercício utiliza um sensor ultrassônico para ver a que distancia é que o objeto à frente está dos sensores

Exercício 4

Montagem do circuito e respectivos testes:



Código Utilizado:

```
#include <IRremote.h>

int receiver_pin = 8;
int first_light_pin = 7;
int second_light_pin = 6;
int third_light_pin = 5;
int fourth_light_pin = 4;
int led_case[] {0,0,0,0,0};
IRrecv receiver(receiver_pin);
decode_results output;

#define code1 48703
#define code2 58359
#define code3 539
#define code4 25979

void setup()
{
  Serial.begin(9600);
  receiver.enableIRIn();
  pinMode(first_light_pin, OUTPUT);
  pinMode(second_light_pin, OUTPUT);
  pinMode(third_light_pin, OUTPUT);
  pinMode(fourth_light_pin, OUTPUT);
}
```

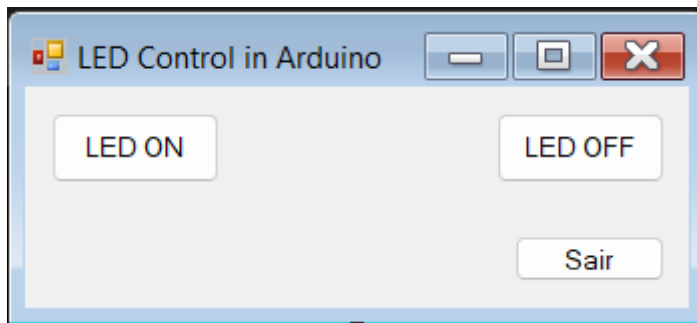
```
void loop()
{
  if (receiver.decode(&output))
  {
    unsigned int value = output.value;
    switch(value)
    {
      case code1:
        if(led_case[1] == 1)
        {
          digitalWrite(first_light_pin, LOW);
          led_case[1] = 0;
        }
        else
        {
          digitalWrite(first_light_pin, HIGH);
          led_case[1] = 1;
        }
        break;
      case code2:
        if(led_case[2] == 1)
        {
          digitalWrite(second_light_pin, LOW);
          led_case[2] = 0;
        }
        else
        {
          digitalWrite(second_light_pin, HIGH);
          led_case[2] = 1;
        }
        break;
      case code3:
```

```
        if(led_case[3] == 1)
        {
          digitalWrite(third_light_pin, LOW);
          led_case[3] = 0;
        }
        else
        {
          digitalWrite(third_light_pin, HIGH);
          led_case[3] = 1;
        }
        break;
      case code4:
        if(led_case[4] == 1)
        {
          digitalWrite(fourth_light_pin, LOW);
          led_case[4] = 0;
        }
        else
        {
          digitalWrite(fourth_light_pin, HIGH);
          led_case[4] = 1;
        }
        break;
    }
    Serial.println(value);
    receiver.resume();
  }
}
```

Neste exercício os LEDs são controlados através da comunicação entre um comando infravermelho e um sensor infravermelho.

Exercício 6

Design do Windows Form:



Código Utilizado:

```
//Ação botão LED ON
1 referência
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        serialPort1.Write("1");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

```
//Ação botão LED OFF
1 referência
private void button2_Click(object sender, EventArgs e)
{
    try
    {
        serialPort1.Write("0");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

```
//Botão fechar aplicação e Port
1 referência
private void button3_Click(object sender, EventArgs e)
{
    serialPort1.Close();
    Application.Exit();
}
```

```
//Abrir Port1 ao carregar o Form
1 referência
private void Form1_Load(object sender, EventArgs e)
{
    serialPort1.Open();
}
```

Ao pressionar os botões o LED liga-se ou desliga-se por causa dos valores inseridos no write do serialPort1.