**PRÉ RELATÓRIO – INICIAÇÃO CIENTÍFICA: INTERNET DAS COISAS**

**João Vítor Fernandes Dias**

1. **Breve descrição do dispositivo pretendido.**

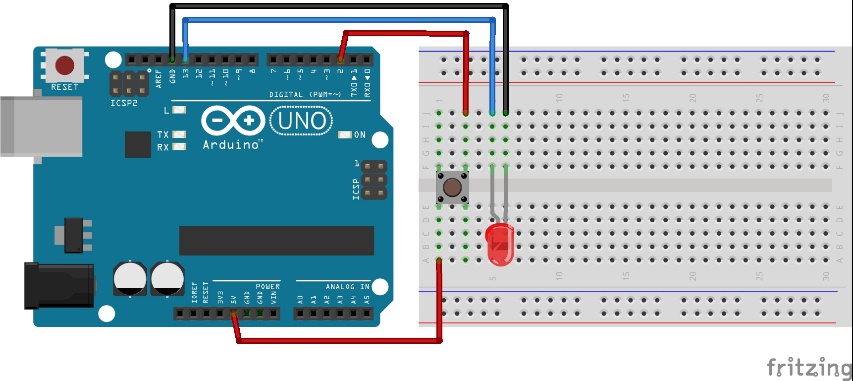
A ideia inicial era me familiarizar novamente com a plataforma Arduino, então elaborei diversos microprojetos, com a intenção de ir aprimorando-os ao ponto de alcançar o objetivo final que é o de controlar um braço robótico.

1. Ligar um LED ao pressionar um botão;
2. Ligar quatro LEDs em posição das quatro coordenadas cartesianas em loop;
3. Ligar os quatro LEDs de acordo com a posição de dois potenciômetros representando as coordenadas x e y, potenciômetro posteriormente substituído por um joystick;
4. Variar a intensidade luminosa dos quatro LEDs de acordo com a posição do joystick em relação as coordenadas cartesianas;
5. Variar a angulação do servo motor de acordo com a posição de um potenciômetro;
6. Uso do módulo Bluetooth para acender e apagar LED;
7. Uso do módulo Bluetooth para controlar a luminosidade de um LED;
8. [WIP] Uso do módulo Bluetooth para controlar a luminosidade de um LED com app específico desenvolvido por mim;
   1. [WIP] Uso de um app desenvolvido por mim para testar as formas de passagem de informação providas pelo App Inventor 2 através do módulo Bluetooth e a forma de recepção desses dados pelo Arduino;
9. [TODO] Uso do módulo Bluetooth para controlar a luminosidade de quatro LEDs;
10. [TODO] Uso do módulo Bluetooth para controlar a angulação de quatro Servo motores;
11. [TODO] Aplicar a programação do microprojeto 9 no projeto.

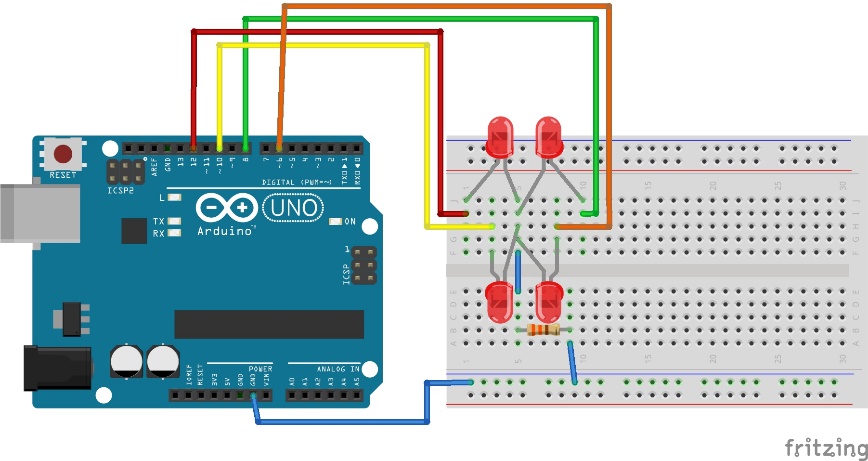
[WIP]: Work In Progress (trabalho em andamento);

[TODO]: To Do (a fazer” (Trabalho ainda a ser feito)).

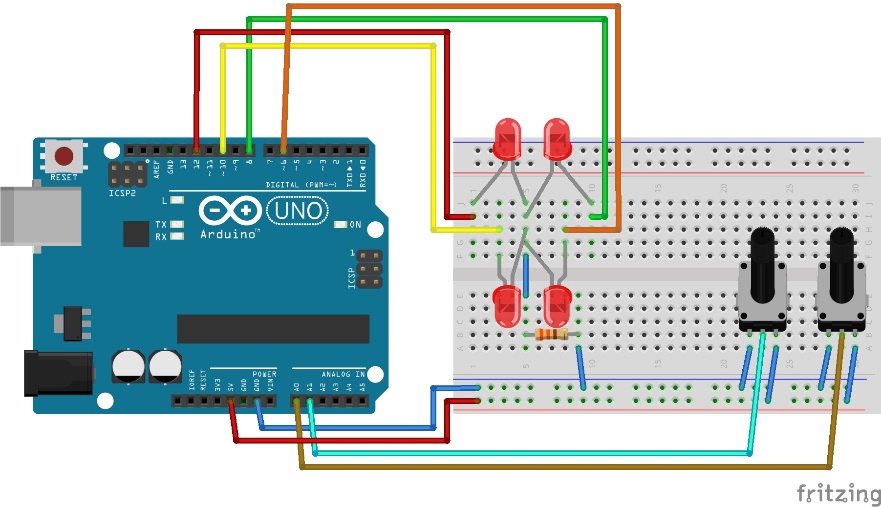
1. **Relação de componentes utilizados.**
2. Arduino Uno,
3. cabo USB,
4. protoboard,
5. diversos Jumpers,
6. um botão,
7. resistores de 330Ω,
8. quatro LEDs (Light Emitting Diode – Diodos Emissores de Luz),
9. dois potenciômetros,
10. um joystick,
11. servo motor,
12. conector bluetooth HC05,
13. celular com conexão Bluetooth e aplicativo de controle Bluetooth
14. **Diagrama de conexão.**
15. Ligar um LED ao pressionar um botão



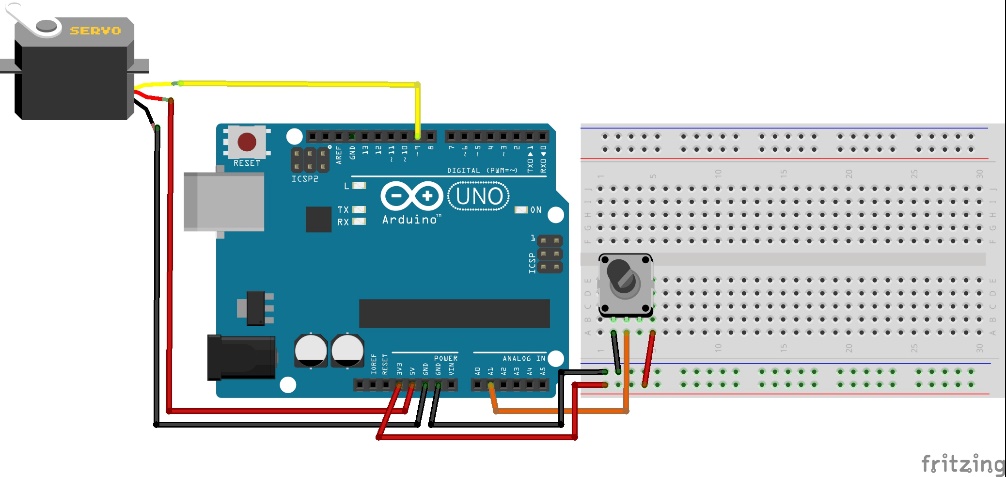
1. Ligar quatro LEDs em posição das quatro coordenadas cartesianas em loop



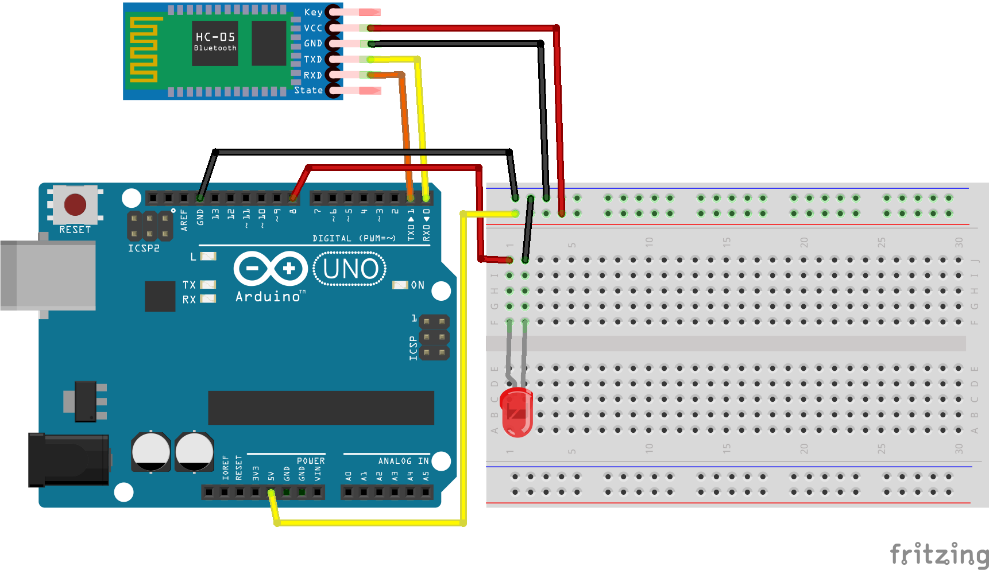
1. Ligar os quatro LEDs de acordo com a posição de dois potenciômetros representando as coordenadas x e y, potenciômetro posteriormente substituído por um joystick
2. Variar a intensidade luminosa dos quatro LEDs de acordo com a posição do joystick em relação as coordenadas cartesianas



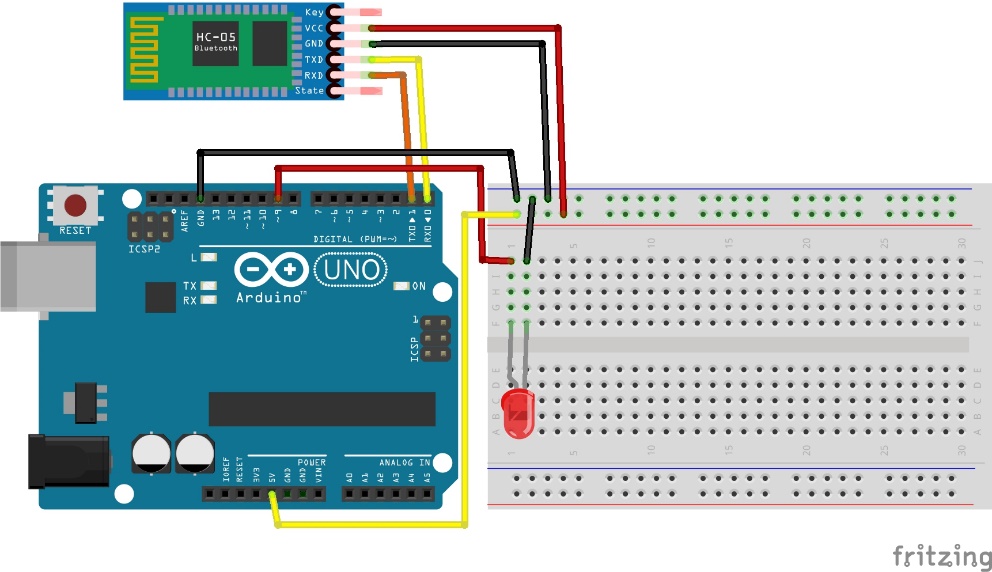
1. Variar a angulação do servo motor de acordo com a posição de um potenciômetro;



1. Uso do módulo Bluetooth para acender e apagar LED



1. Uso do módulo Bluetooth para controlar a luminosidade de um LED



1. [WIP] Uso do módulo Bluetooth para controlar a luminosidade de um LED com app específico desenvolvido por mim;
   1. [WIP] Uso de um app desenvolvido por mim para testar as formas de passagem de informação providas pelo App Inventor através do módulo Bluetooth e a forma de recepção desses dados pelo Arduino;
2. **Relação de ferramentas computacionais para programação.**

Dev. C++ 5.11;

Arduino 1.8.11;

MIT App Inventor 2.

1. **Código ou programa de controle.**
2. **Ligar um LED ao pressionar um botão**

const int buttonPin = 2; // Pino em que o Botão está conectado

const int ledPin = 13; // Pino em que o LED está conectado

int buttonState = 0; // Variável para o estado do botão: 0 (desligado) ou 1(ligado)

void setup(){

pinMode(ledPin, OUTPUT); // Inicializa o Pino 13 (LED) como sendo Output

pinMode(buttonPin, INPUT); // Inicializa o Pino 2 (Botão) como sendo Input

}

void loop(){

buttonState = digitalRead(buttonPin); // Lê o valor vindo do botão

if (buttonState == HIGH) // Compara o valor do botão {

digitalWrite(ledPin, HIGH); // se 1, então liga o led

}

else {

digitalWrite(ledPin, LOW); // senão, desliga o led;

}

}

1. **Ligar quatro LEDs em posição das quatro coordenadas cartesianas em loop**

const int led1 = 8;

const int led2 = 12;

const int led3 = 10;

const int led4 = 6;

void setup()

{

pinMode(led1, OUTPUT); // Define o pino 8 como Output

pinMode(led2, OUTPUT); // Define o pino 12 como Output

pinMode(led3, OUTPUT); // Define o pino 10 como Output

pinMode(led4, OUTPUT); // Define o pino 6 como Output

}

void loop()

{

digitalWrite(led1, HIGH); // Liga led

delay(200); // Espera 200 ms

digitalWrite(led1, LOW); // Desliga LED

digitalWrite(led2, HIGH); // Liga led

delay(200); // Espera 200 ms

digitalWrite(led2, LOW); // Desliga LED

digitalWrite(led3, HIGH); // Liga led

delay(200); // Espera 200 ms

digitalWrite(led3, LOW); // Desliga LED

digitalWrite(led4, HIGH); // Liga led

delay(200); // Espera 200 ms

digitalWrite(led4, LOW); // Desliga LED

}

1. **Ligar os quatro LEDs de acordo com a posição de dois potenciômetros representando as coordenadas x e y, potenciômetro posteriormente substituído por um joystick**

const int led1 = 3;

const int led2 = 5;

const int led3 = 6;

const int led4 = 9;

const int X = A0;

const int Y = A1;

int EntradaX = 0;

int EntradaY = 0;

int MapX = 0;

int MapY = 0;

int MeioX = 512;

int MeioY = 512;

void setup() {

pinMode(led1, OUTPUT);

pinMode(led2, OUTPUT);

pinMode(led3, OUTPUT);

pinMode(led4, OUTPUT);

Serial.begin(9600);

MeioX = analogRead(X);

MeioY = analogRead(Y);

}

void loop(){

EntradaX = analogRead(X);

EntradaY = analogRead(Y);

if (EntradaX > MeioX) {

MapX = map(EntradaX, 0, 1023, 0, 255);

}

else

{

if (EntradaX < MeioX) {

MapX = map(EntradaX, 0, 1023, 255, 0);

}

};

if (EntradaY > MeioY) {

MapY = map(EntradaY, 0, 1023, 0, 255);

}

else {

if (EntradaY < MeioY)

{

MapY = map(EntradaY, 0, 1023, 255, 0);

};

}

if (EntradaX > MeioX && EntradaY > MeioY) {

if (MapX > MapY) {

analogWrite(led1, MapX);

}

else {

analogWrite(led1, MapY);

}

digitalWrite(led2, LOW);

digitalWrite(led3, LOW);

digitalWrite(led4, LOW);

}

else if (EntradaX < MeioX && EntradaY > MeioY)

{

digitalWrite(led1, LOW);

if (MapX > MapY) {

analogWrite(led2, MapX);

}

else {

analogWrite(led2, MapY);

}

digitalWrite(led3, LOW);

digitalWrite(led4, LOW);

}

else if (EntradaX < MeioX && EntradaY < MeioY) {

digitalWrite(led1, LOW);

digitalWrite(led2, LOW);

if (MapX > MapY) {

analogWrite(led3, MapX);

}

else {

analogWrite(led3, MapY);

}

digitalWrite(led4, LOW);

}

else if (EntradaX > MeioX && EntradaY < MeioY)

{

digitalWrite(led1, LOW);

digitalWrite(led2, LOW);

digitalWrite(led3, LOW);

if (MapX > MapY) {

analogWrite(led4, MapX);

}

else {

analogWrite(led4, MapY);

}

}

}

1. **Variar a intensidade luminosa dos quatro LEDs de acordo com a posição do joystick em relação as coordenadas cartesianas**

const int led1 = 8;

const int led2 = 12;

const int led3 = 10;

const int led4 = 6;

const int PotX = A0;

const int PotY = A1;

int EntradaX = 0;

int EntradaY = 0;

int MostrarX = 0;

int MostrarY = 0;

void setup(){

pinMode(led1, OUTPUT);

pinMode(led2, OUTPUT);

pinMode(led3, OUTPUT);

pinMode(led4, OUTPUT);

Serial.begin(9600);

}

void loop(){

EntradaX = analogRead(PotX);

MostrarX = map(EntradaX, 0, 1023, 0, 255);

EntradaY = analogRead(PotY);

MostrarY = map(EntradaY, 0, 1023, 0, 255);

if (EntradaY > 512 && EntradaX > 512) {

digitalWrite(led1, HIGH);

digitalWrite(led2, LOW);

digitalWrite(led3, LOW);

digitalWrite(led4, LOW);

}

else if (EntradaY > 512 && EntradaX < 512) {

digitalWrite(led1, LOW);

digitalWrite(led2, HIGH);

digitalWrite(led3, LOW);

digitalWrite(led4, LOW);

}

else if (EntradaY < 512 && EntradaX < 512) {

digitalWrite(led1, LOW);

digitalWrite(led2, LOW);

digitalWrite(led3, HIGH);

digitalWrite(led4, LOW);

}

else if (EntradaY < 512 && EntradaX > 512) {

digitalWrite(led1, LOW);

digitalWrite(led2, LOW);

digitalWrite(led3, LOW);

digitalWrite(led4, HIGH);

}

}

1. **Variar a angulação do servo motor de acordo com a posição de um potenciômetro**

#include <Servo.h>

Servo servo1;

int pinX1 = A1;

int X1;

void setup()

{

servo1.attach(9);

}

void loop()

{

X1 = analogRead(pinX1);

X1 = map(X1, 0, 1023, 0, 180);

servo1.write(X1);

delay(15);

}

1. **Uso do módulo Bluetooth para acender e apagar LED**

void setup(){

Serial.begin(9600);

pinMode(8, OUTPUT);

}

void loop(){

if (Serial.available() > 0) {

char data = Serial.read(); // Lê o valor recebido via Bluetooth

Serial.print(data);

Serial.print("\n");

switch (data) {

case 'a': digitalWrite(8, HIGH); break; // Caso envie o valor a, liga led.

case 'b': digitalWrite(8, LOW); break; // Caso envie o valor b, desliga led.

default : break;

}

}

delay(40);

}

1. **Uso do módulo Bluetooth para controlar a luminosidade de um LED**

int led1 = 9;

int dimmer;

String stringGeral;

void setup(){

Serial.begin(9600);

pinMode(led1, OUTPUT);

}

void loop(){

if (Serial.available()) {

stringGeral = String("");

while (Serial.available()) {

stringGeral = stringGeral + char(Serial.read());

delay(1);

}

dimmer = stringGeral.toInt();

if ( dimmer >= 0 && dimmer <= 255) {

analogWrite(led1, dimmer);

}

}

}

1. **[WIP] Uso do módulo Bluetooth para controlar a luminosidade de um LED com app específico desenvolvido por mim**
   1. **[WIP] Uso de um app desenvolvido por mim para testar as formas de passagem de informação providas pelo App Inventor através do módulo Bluetooth e a forma de recepção desses dados pelo Arduino**