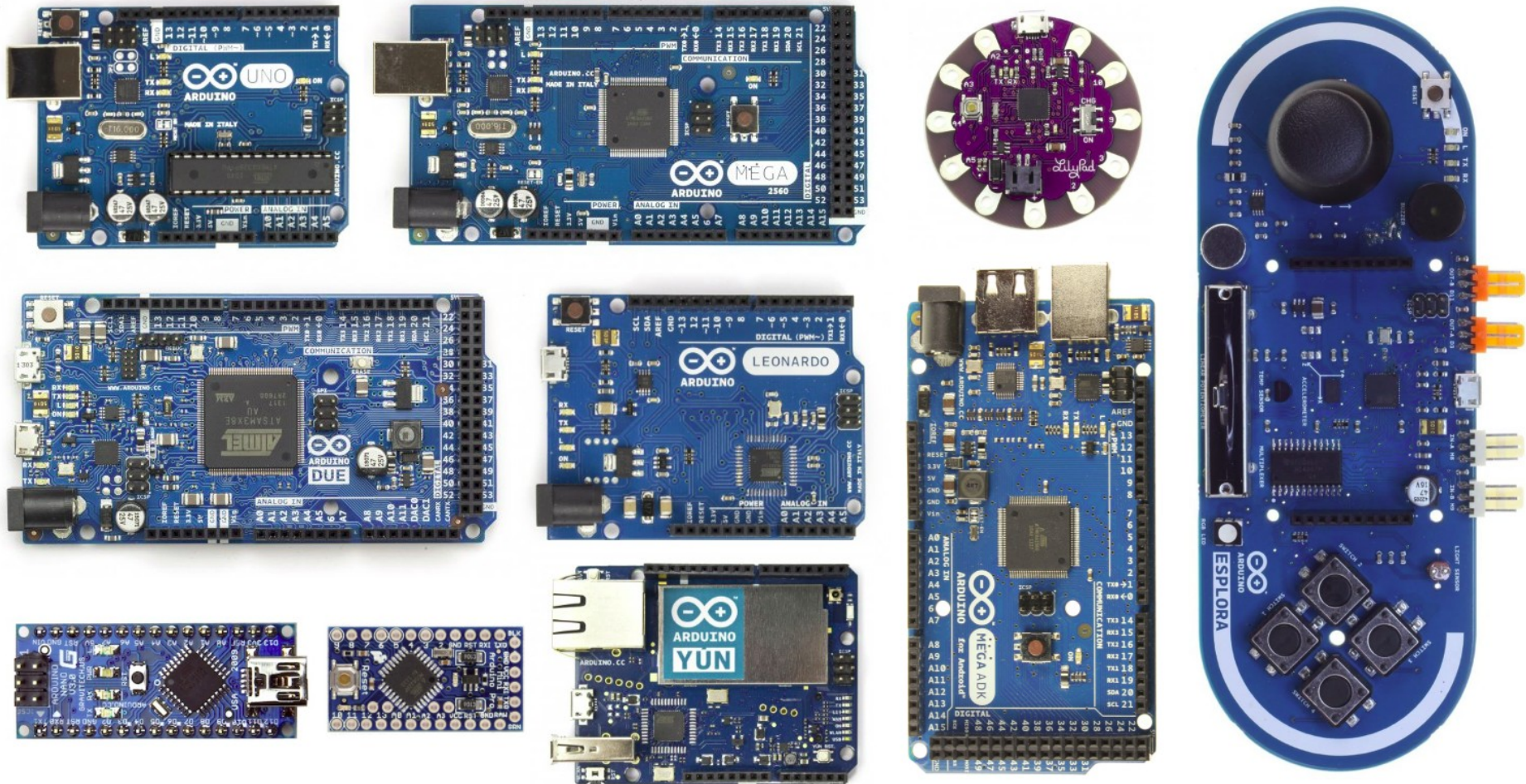


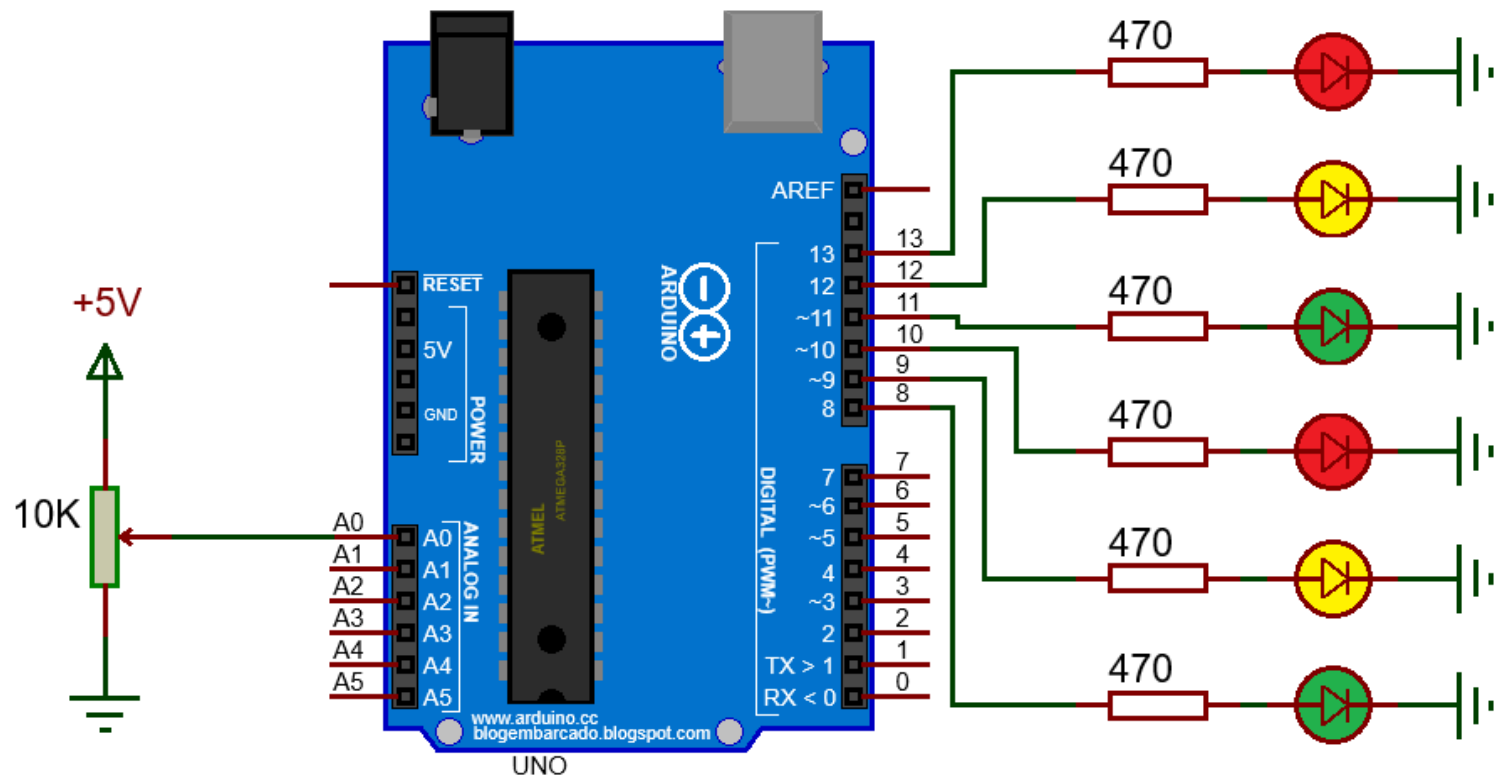


Curso Arduino – Aula 3

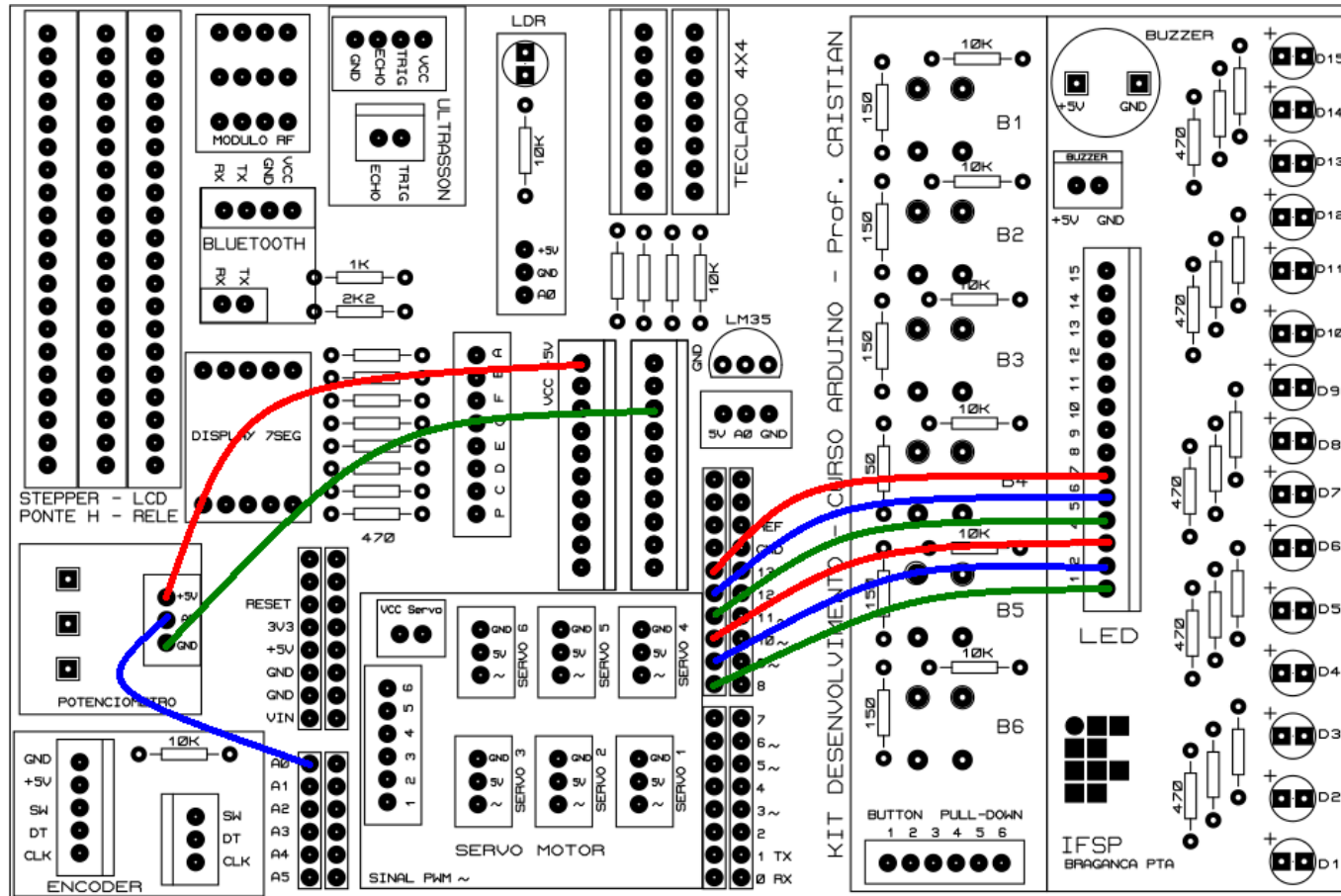


Prática 8 – Controle de acionamento de LEDs utilizando potenciômetro

Crie um programa onde ao variar o potenciômetro os LED's se acendam proporcionalmente.



Prática 8 – Controle de acionamento de LEDs utilizando potenciômetro



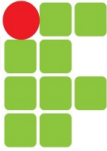


Comando **if ... else**

Neste último projeto você poderá implementar o controle do acionamento dos LED's através do potenciômetro utilizando o comando **if**.

Mas para facilitar e diminuir o número de linhas no programa podemos utilizar o comando **if ... else** .

O **else** faz a função complementar ao **if**, ou seja, se a condição **if** for falsa, o complemento verdadeira será dados pelo **else**.



Acionamento por chaves usando Comando **if ... else**

```
int botao = 2;
int led = 13;

int estado = 0;

void setup(){
  pinMode(led, OUTPUT);
  pinMode(botao, INPUT);
}

void loop(){
  estado=digitalRead(botao);
  if(estado == HIGH){
    digitalWrite(led, HIGH);
  }
  → if(estado == LOW){
    digitalWrite(led, LOW);
  }
}
```

```
int botao = 2;
int led = 13;

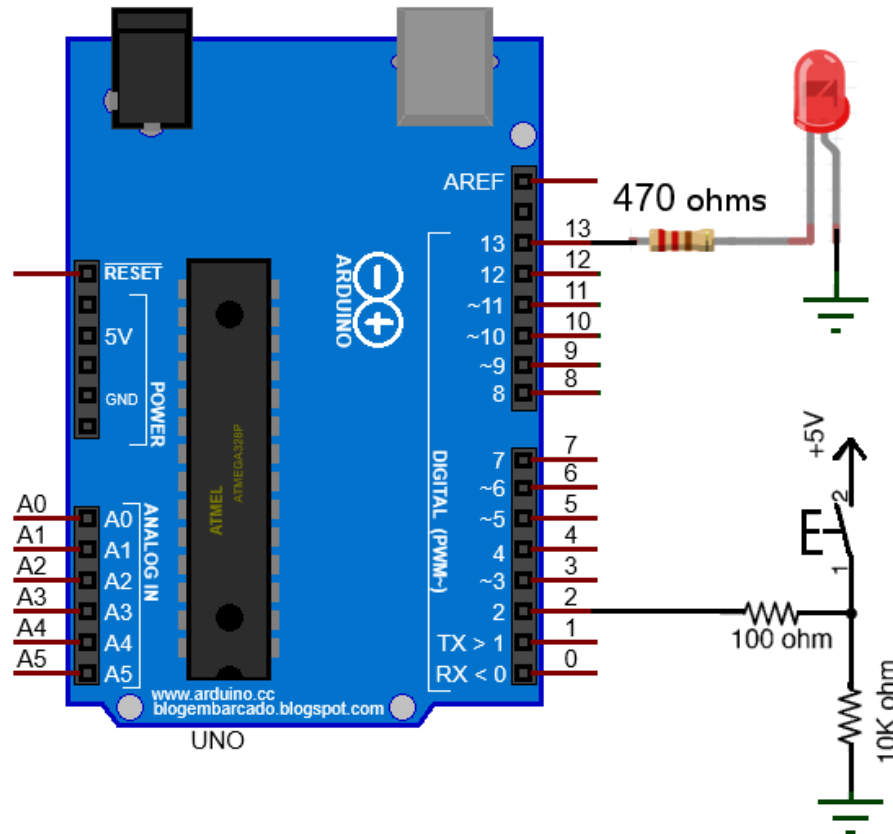
int estado = 0;

void setup(){
  pinMode(led, OUTPUT);
  pinMode(botao, INPUT);
}

void loop(){
  estado=digitalRead(botao);
  if(estado == HIGH){
    digitalWrite(led, HIGH);
  }
  → else{
    digitalWrite(led, LOW);
  }
}
```

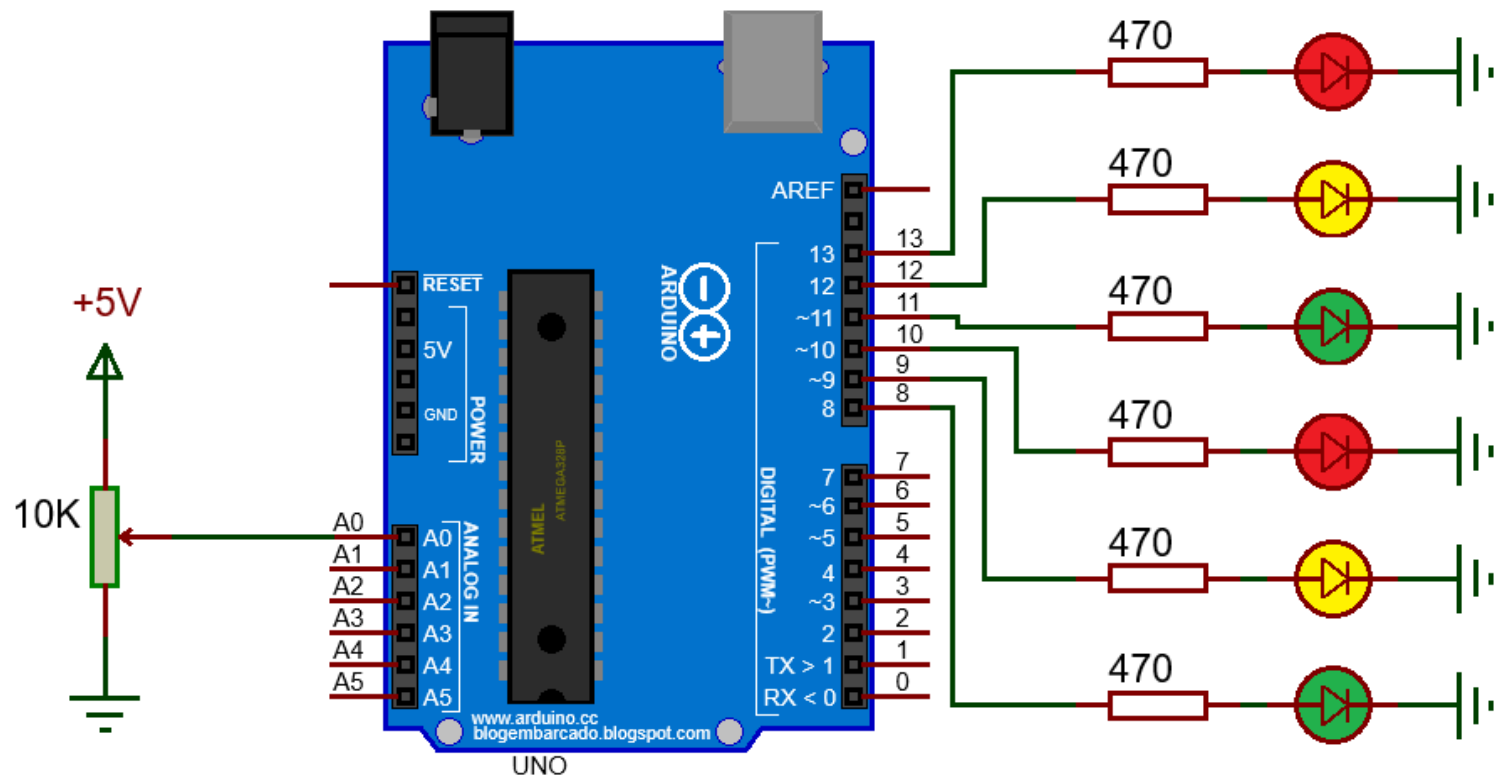

Acionamento por chaves usando Comando **if ... else**

Veja a ligação do circuito para relembrar.
Observe que agora estamos usando uma chave **pull down**.



Prática 8 – Controle de acionamento de LEDs utilizando potenciômetro

Crie um programa onde ao variar o potenciômetro os LED's se acendam proporcionalmente.



Prática 8 – Controle de acionamento de LEDs utilizando potenciômetro

```
int Pot = 0;
int valor = 0;

void setup()
{
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(8, OUTPUT);
}

void loop()
{
  valor = analogRead(Pot);

  if(valor > 100){
    digitalWrite(8, HIGH); }
  else{
    digitalWrite(8, LOW); }
```

```
    if(valor > 300){
      digitalWrite(9, HIGH); }
    else{
      digitalWrite(9, LOW); }

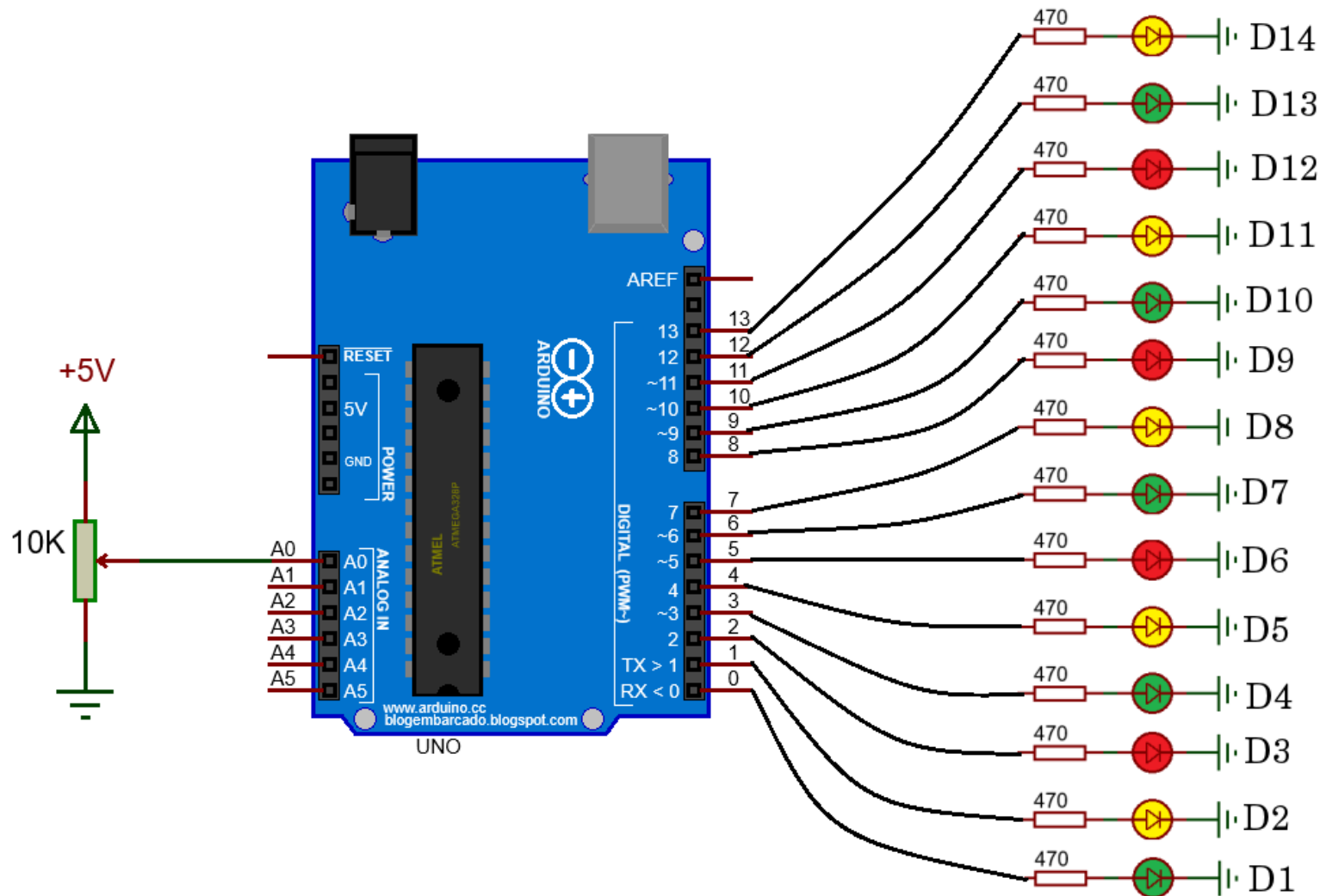
    if(valor > 500){
      digitalWrite(10, HIGH); }
    else{
      digitalWrite(10, LOW); }

    if(valor > 650){
      digitalWrite(11, HIGH); }
    else{
      digitalWrite(11, LOW); }

    if(valor > 850){
      digitalWrite(12, HIGH); }
    else{
      digitalWrite(12, LOW); }

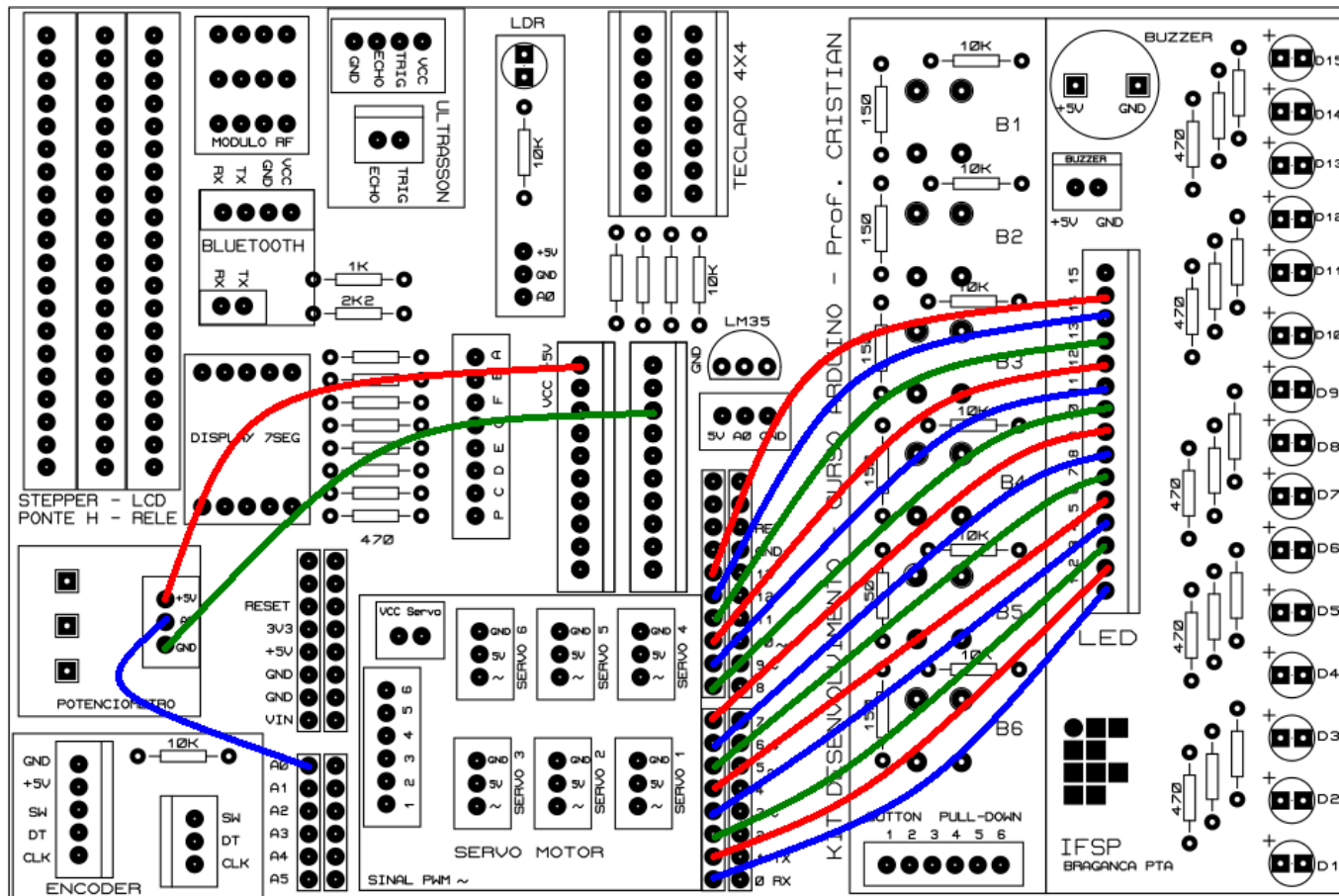
    if(valor > 1000){
      digitalWrite(13, HIGH); }
    else{
      digitalWrite(13, LOW); }
  }
```


Prática 9 – Circuito Sequencial de LEDs utilizando potenciômetro

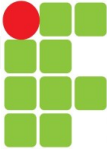


Prática 9 – Circuito Sequencial de LEDs utilizando potenciômetro

A ligação do circuito deve ser realizada como visto a seguir:



Prática 9 – Circuito Sequencial de LEDs utilizando potenciômetro



```
byte leds[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13}; // laço para declarar os pinos
int atraso;
int direction = 1;
int led_atual = 0;
int pot = 0; // pino analogico para o potenciometro

void setup(){
    for (int x=0; x<14; x++){ //declara os pinos de 0 a 13 como OUTPUT
        pinMode(leds[x], OUTPUT);
    }
}

void loop(){
    atraso = analogRead(pot);

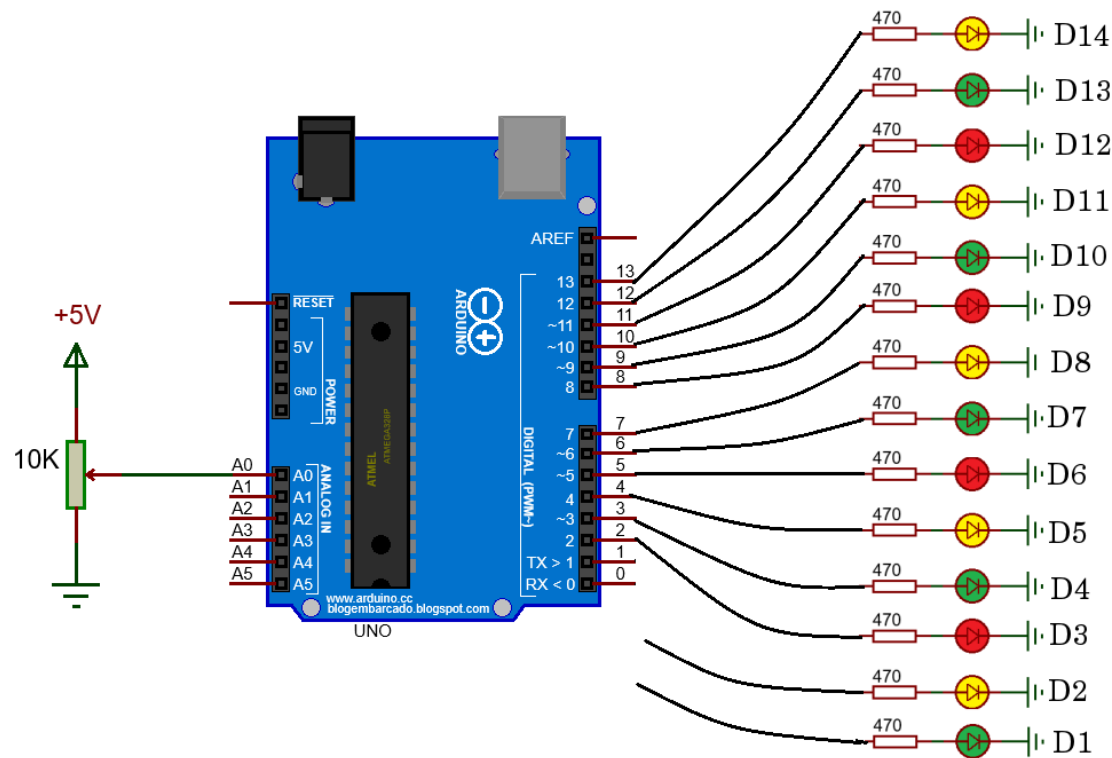
    delay(atraso);

    for (int x=0; x<14; x++){ // apaga todos os leds
        digitalWrite(leds[x], LOW);
    }

    digitalWrite(leds[led_atual], HIGH);
    led_atual += direction; // incrementa de acordo com o valor de direction
                           // x += y; equivale --> x = x + y;

    if(led_atual == 13) {direction = -1;} // altera a direção se tivermos atingido o fim
    if(led_atual == 0) {direction = 1;}
}
```

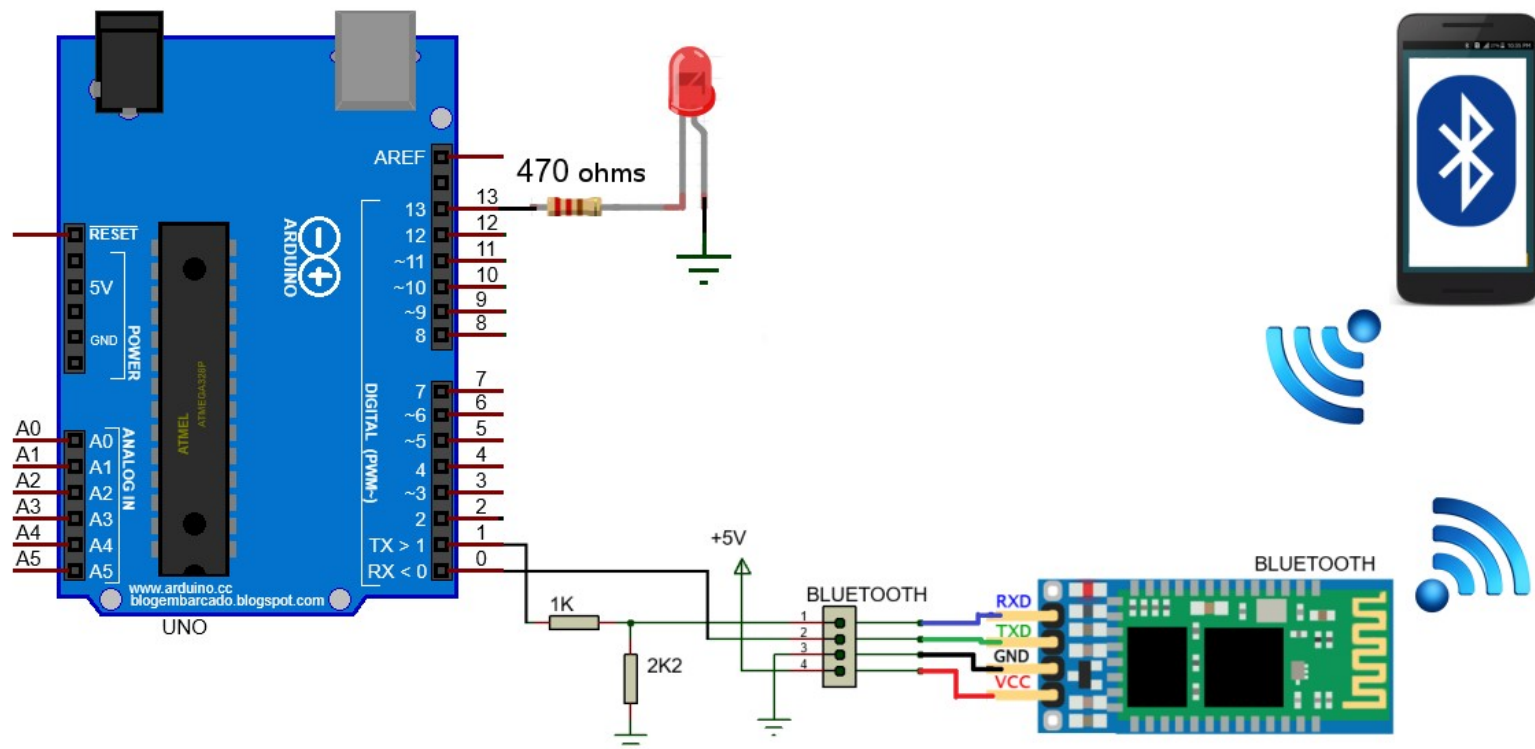
Prática 9 – Circuito Sequencial de LEDs utilizando potenciômetro



Observação: No momento de transmitir o programa para Arduino, os pinos 0 e 1 devem estar desconectados dos LEDs.

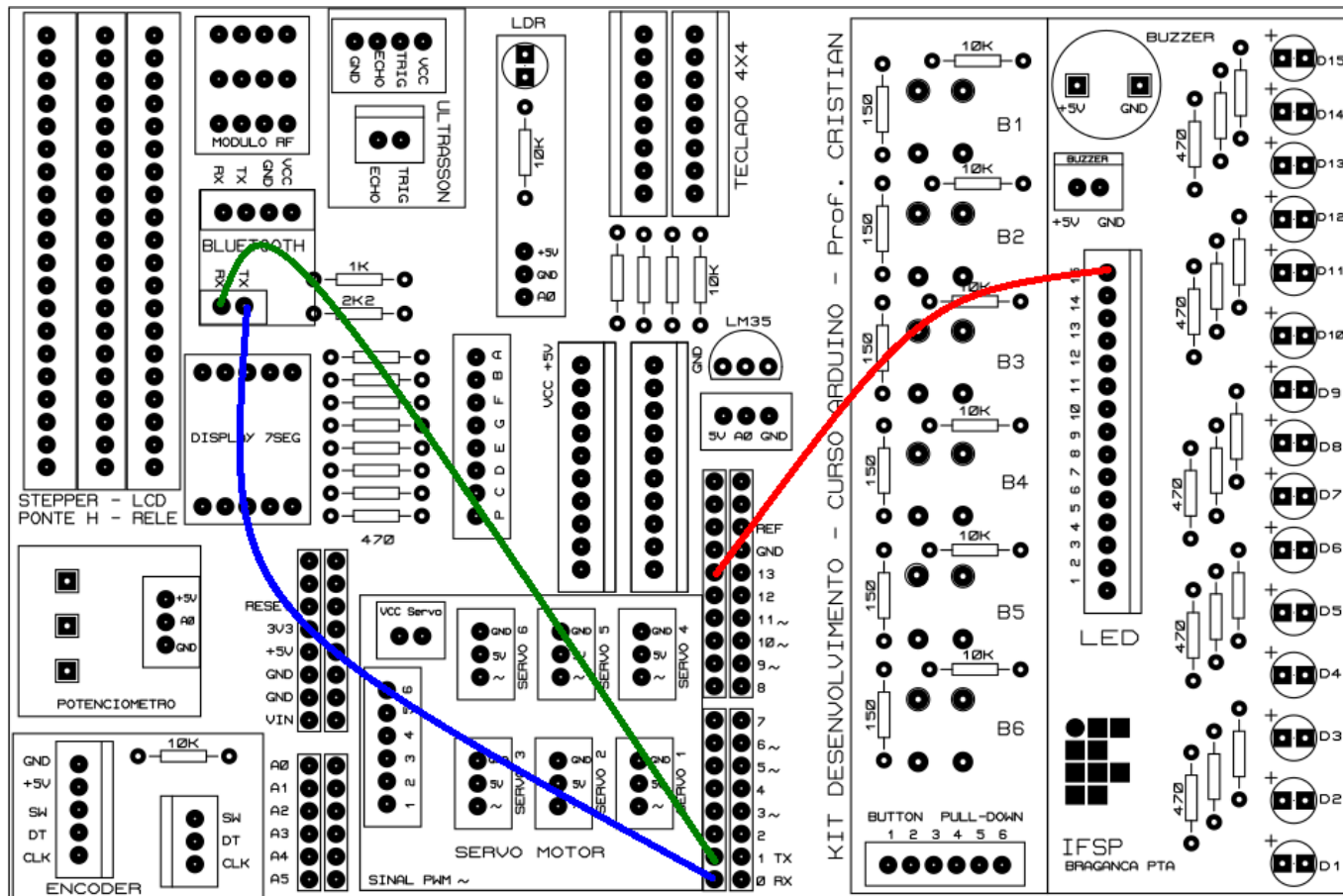
Prática 10 – Acionamento de LED com Celular via BlueTooth

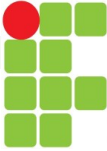
Abaixo temos o circuito para o acionamento do LED com comandos enviados com Celular via BlueTooth.





Prática 10 – Acionamento de LED com Celular via Bluetooth





Prática 10 – Acionamento de LED com Celular via BlueTooth

```
char comando; // função que armazena um caractere.
int LED1 = 13;

void setup() {
    Serial.begin(9600);
    pinMode(LED1, OUTPUT);
}

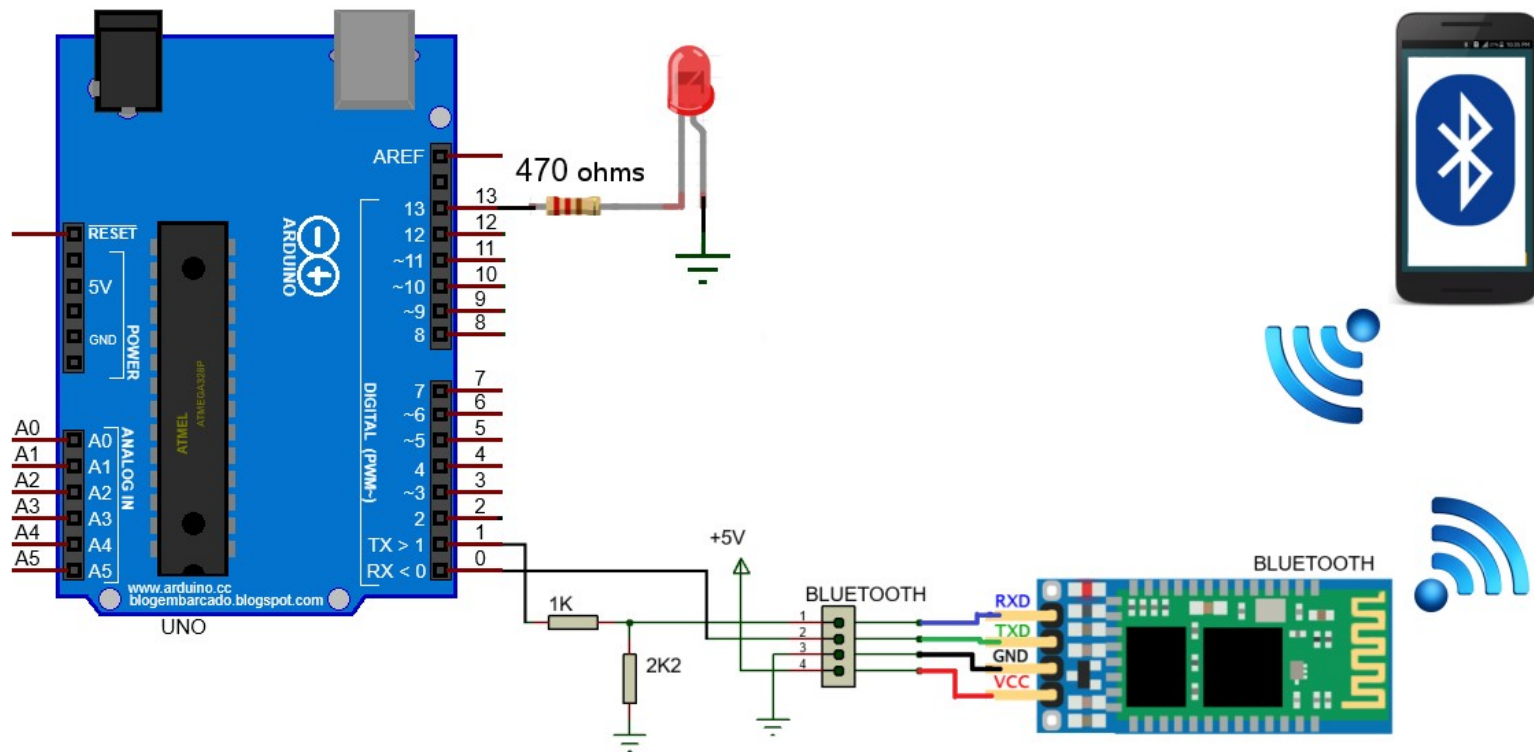
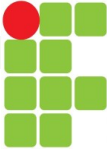
void loop() {

    while(Serial.available() > 0) { // Enquanto houverem bytes disponíveis;

        comando=Serial.read(); // Lê byte do buffer serial;

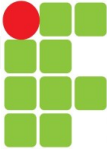
        if(comando=='L') {
            digitalWrite(LED1, HIGH);}
        if(comando=='D') {
            digitalWrite(LED1, LOW);}
    }
}
```

Prática 10 – Acionamento de LED com Celular via BlueTooth

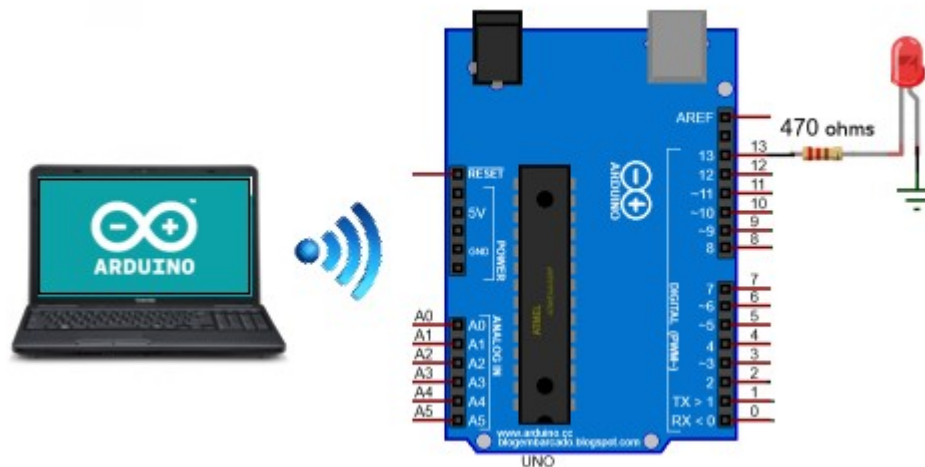


Observação: No momento de transmitir o programa para Arduíno, os pinos 0 e 1 devem estar desconectados do dispositivo BlueTooth.

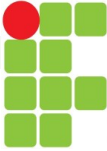
Prática 10 – Acionamento de LED com Celular via Bluetooth



Após concluído a ligação e o programa, antes de iniciarmos a comunicação via Bluetooth, deixe os pinos 0 e 1 desconectado do dispositivo Bluetooth e tente enviar as letra **L** , de liga, e a letra **D**, de desliga, via **monitor serial**.



Prática 10 – Acionamento de LED com Celular via BlueTooth



Para enviar um comando com Celular via BlueTooth, é necessário utilizar um aplicativo que execute essa função.

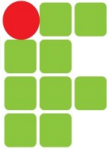
Há vários aplicativos BlueTooth gratuito que pode ser utilizado. No exemplo a seguir usaremos:



BlueTooth Serial Controller

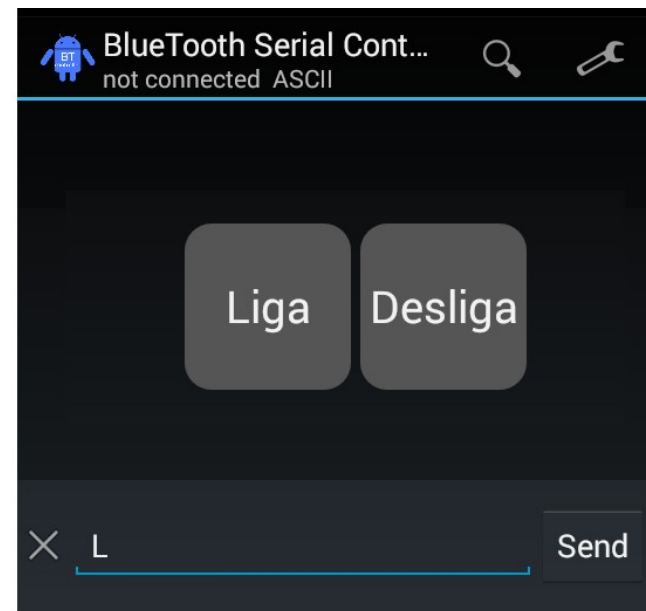
https://play.google.com/store/apps/details?id=nextprototypes.BTSerialController&hl=pt_BR

Prática 10 – Acionamento de LED com Celular via Bluetooth



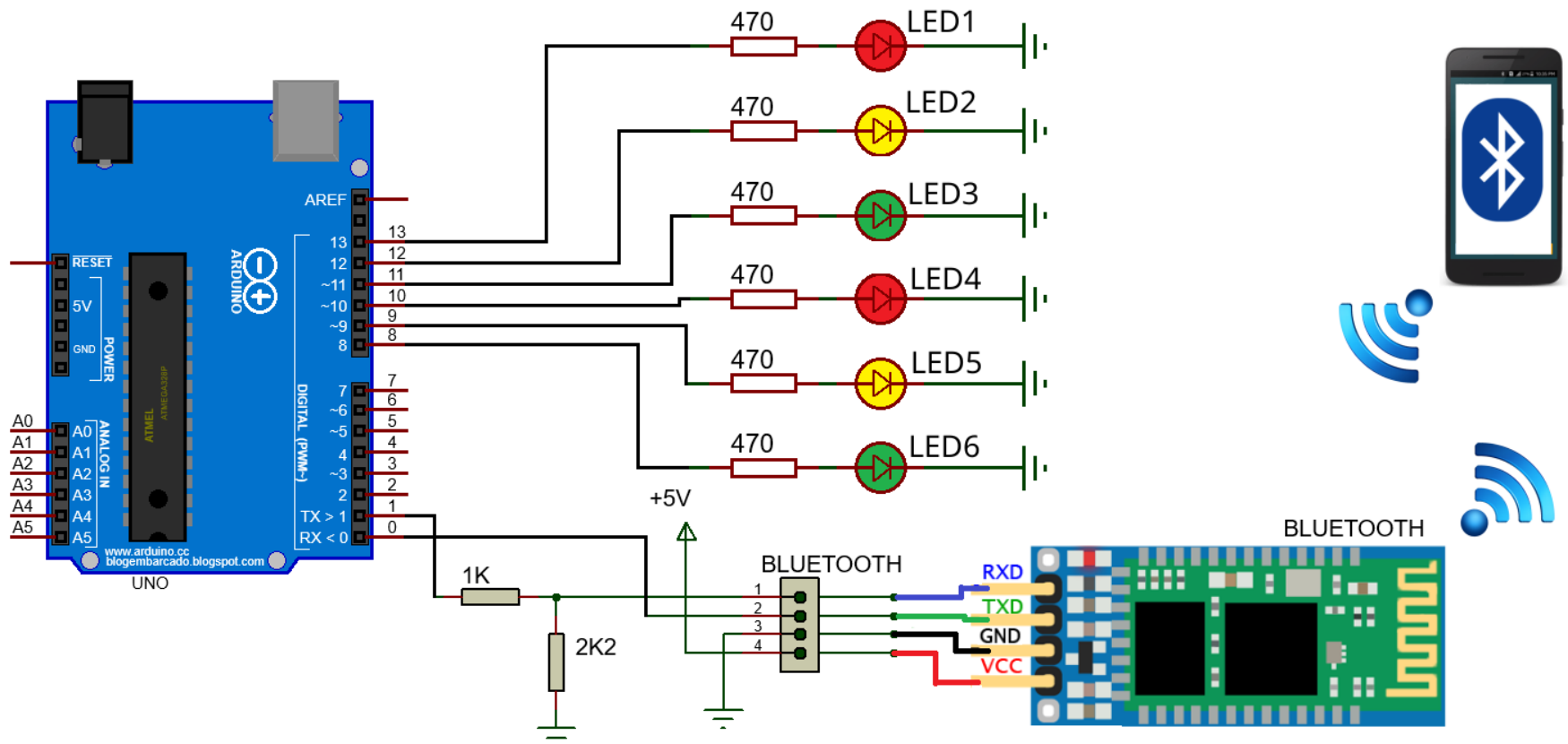
É possível executar o envio do comando L (liga) e o comando D (desliga) de duas forma:

- Escrevendo na tela o comando e enviar;
- Criando um botão que envie esse comandos;



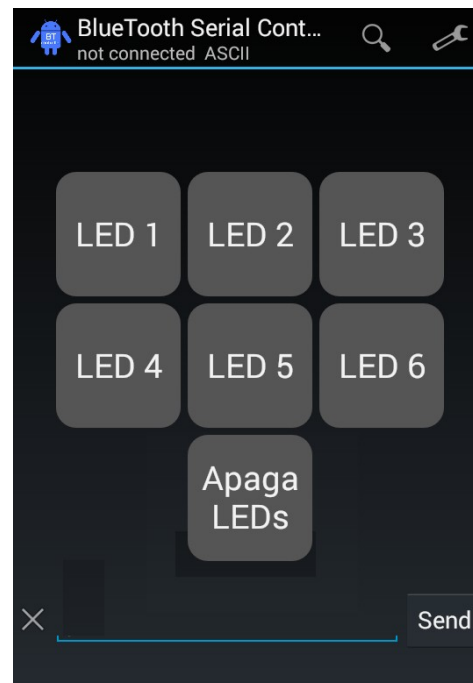
Prática 11 – Acionamento de varios LEDs com Celular via BlueTooth

Crie um programa que ligue um LED por vez, controlado pelo Celular via BlueTooth.



Prática 11 – Acionamento de varios LEDs com Celular via BlueTooth

Uma sugestão, é criar os botões para comanda os LEDs, inserindo os comandos correspondente a cada LED. Mas se preferir os comandos podem ser enviados diretamente via teclado.



Prática 11 – Acionamento de varios LEDs com Celular via BlueTooth

A ligação do circuito deve ser realizada como visto a seguir:

