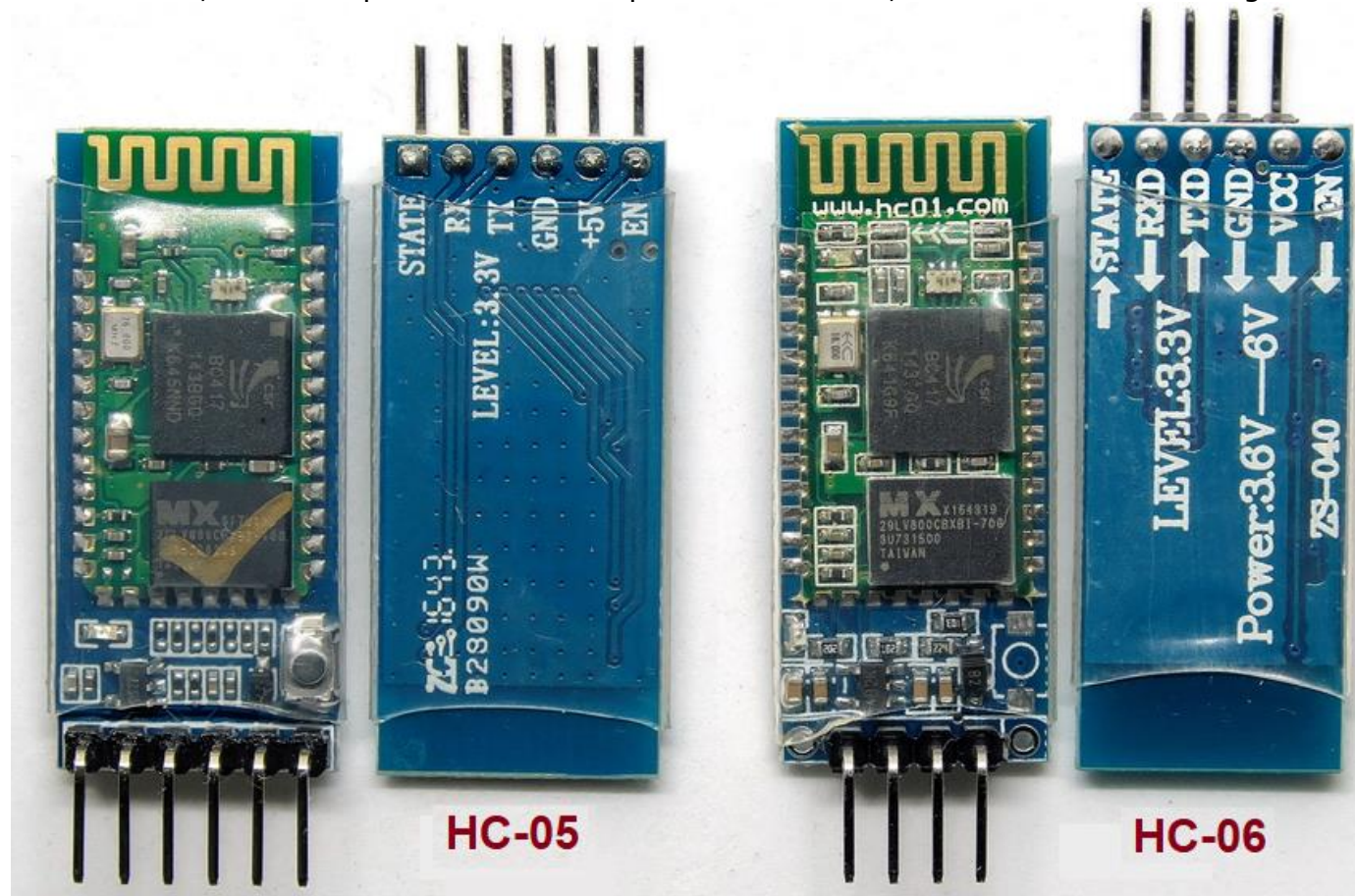


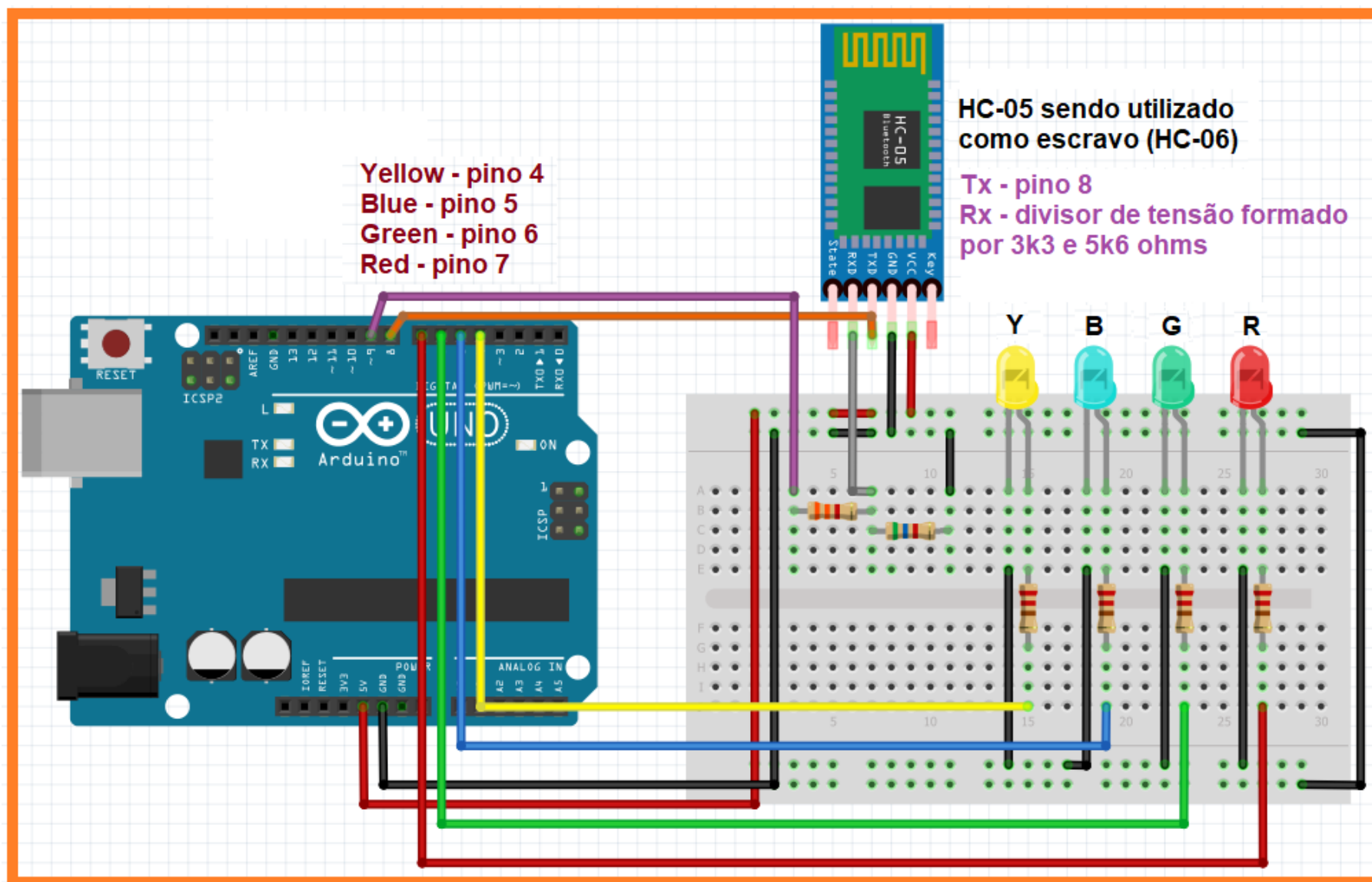
USANDO O BLUETOOTH COM O ARDUINO CONTROLADO POR SMARTPHONE

Vamos analisar a utilização do Bluetooth (BT) HC-06 com o Arduino UNO e como controlar dispositivos através do celular com a utilização de aplicativos. Vamos utilizar um desses aplicativos para controlar 4 leds nas cores amarela, azul, verde e vermelha.

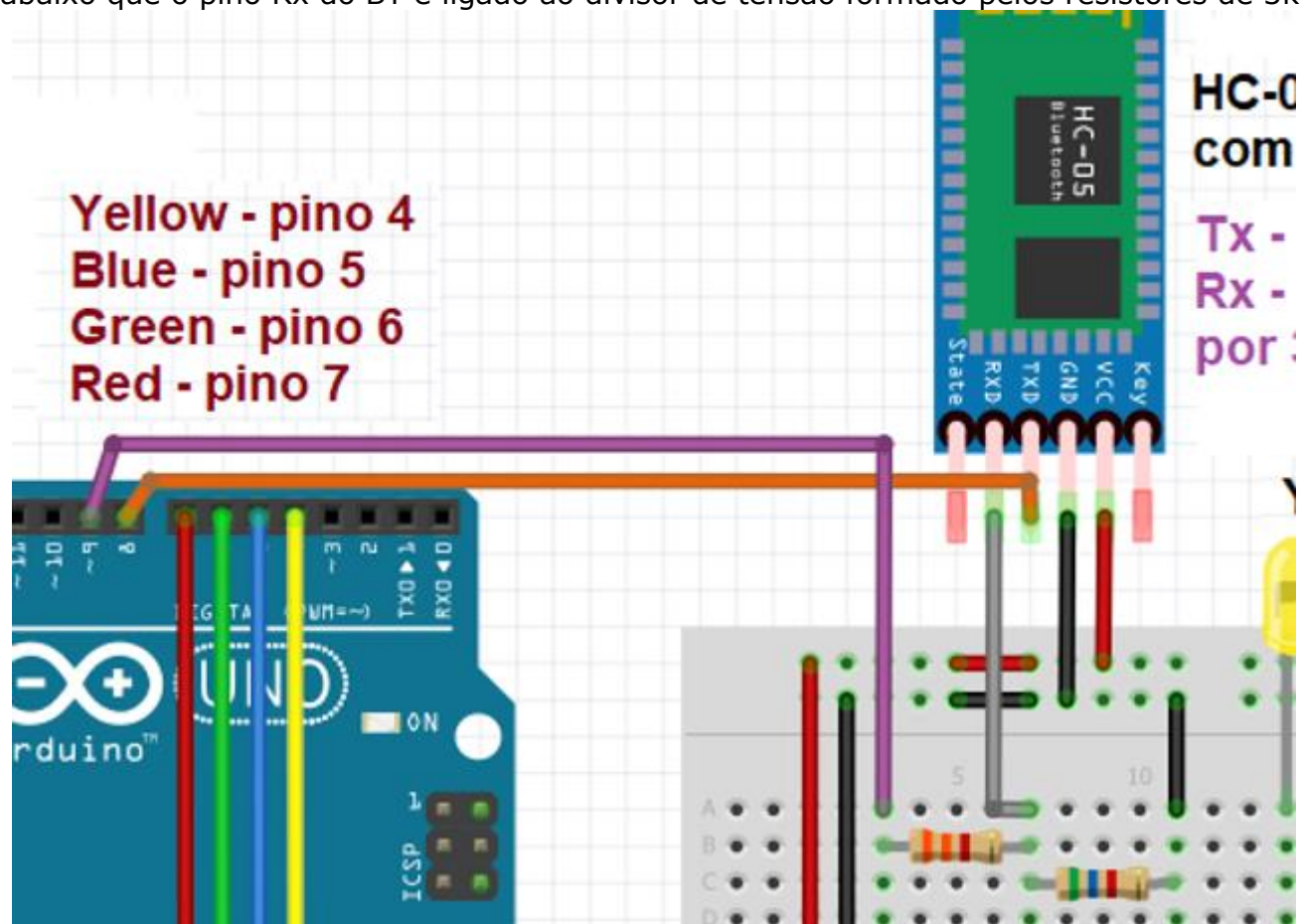
O HC-06 é um BT que opera como escravo, enquanto o HC-05 acumula as duas funções: mestre e escravo, no entanto, o HC-05 pode operar também como escravo, bastando para isso utilizar apenas 4 terminais, conforme ilustram as figuras abaixo:



Para utilizar o módulo BT HC-05 como escravo, basta ignorar os terminais State e EN. A figura abaixo ilustra o diagrama esquemático para controle de 4 leds:



Observe na figura abaixo que o pino Rx do BT é ligado ao divisor de tensão formado pelos resistores de 3k3 e 5k6 ohms.

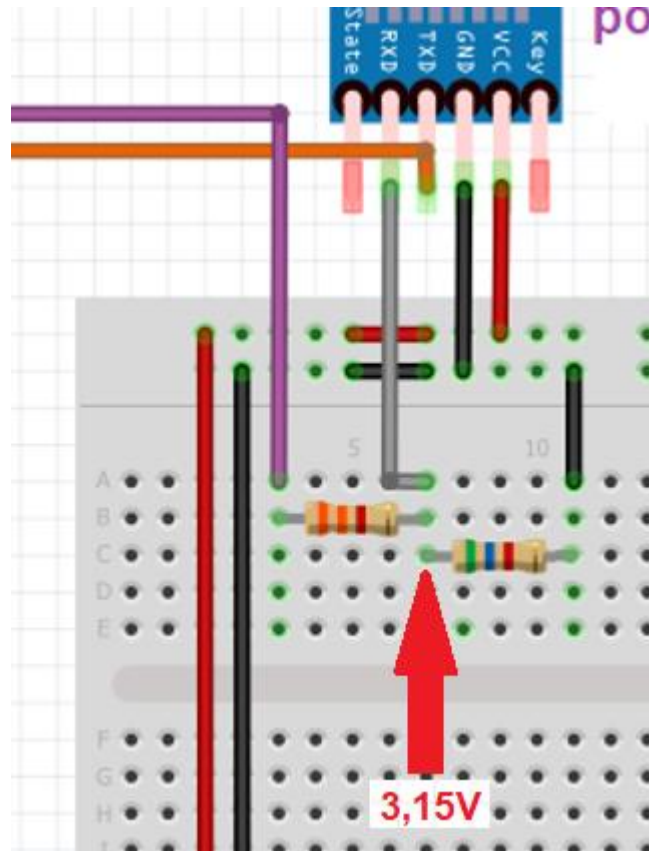


Isto se deve ao fato de que o terminal Rx deve ser alimentado com uma tensão máxima de 3,3 volts.

O divisor de tensão é alimentado pelo pino 9 do Arduino que fornece uma tensão de 5 volts, daí então:

$$5V \times 5k6 / 5k6 + 3k3 = 28.000 / 8.900$$

3,15 volts

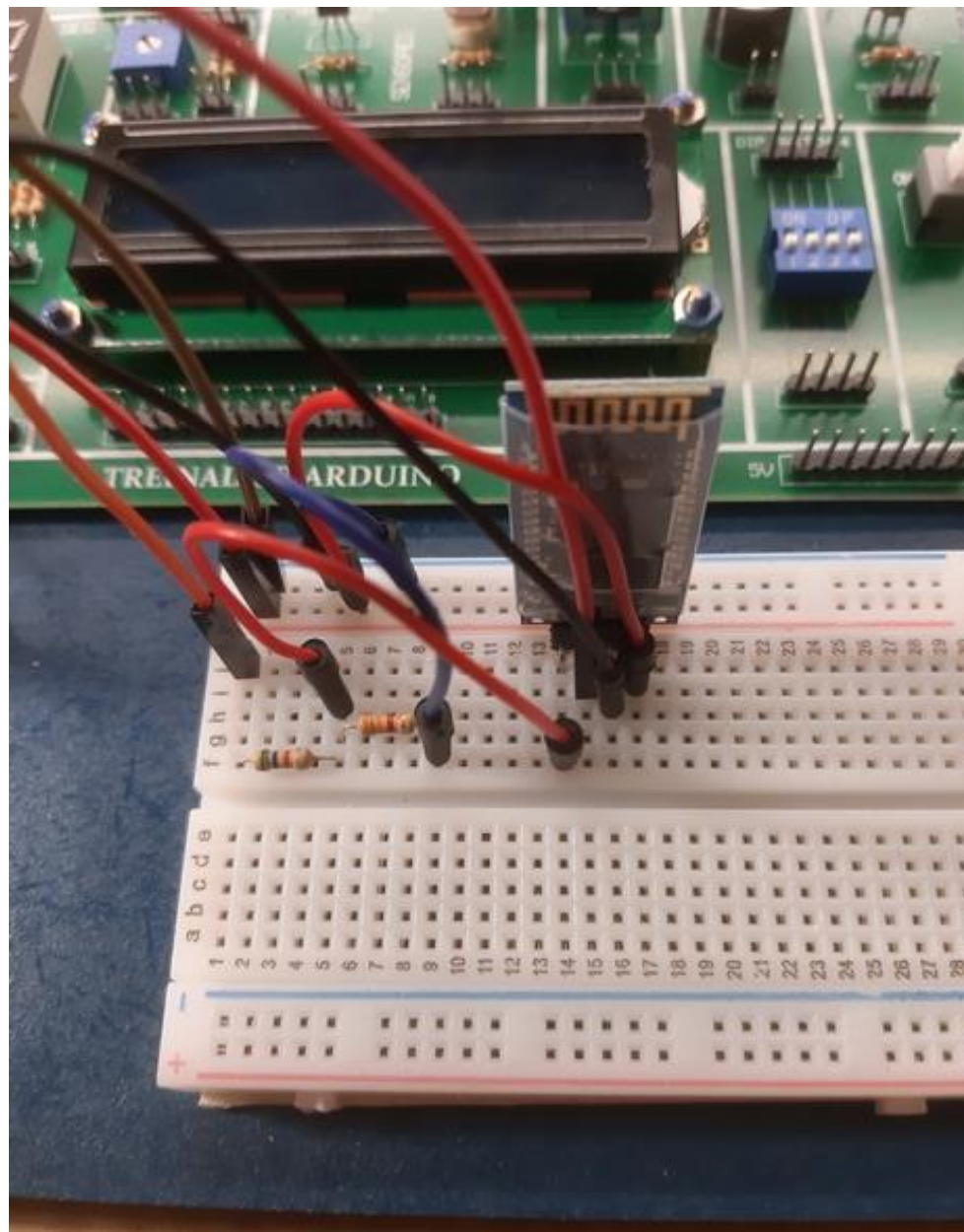


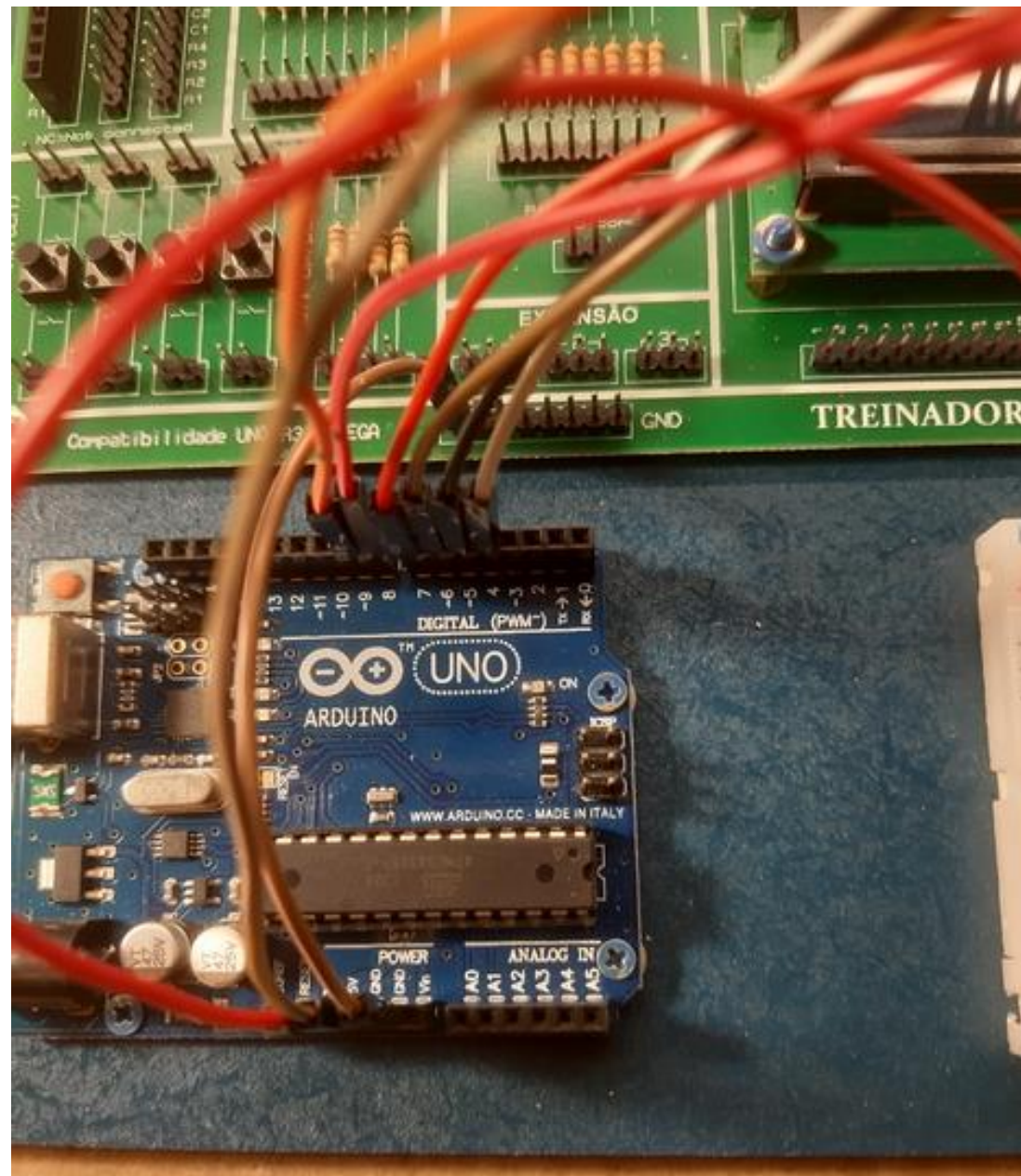
No entanto, os terminais Tx e Rx do BT podem ser conectados também aos terminais 0 e 1 do Arduino (Rx e Tx respectivamente) da seguinte forma:

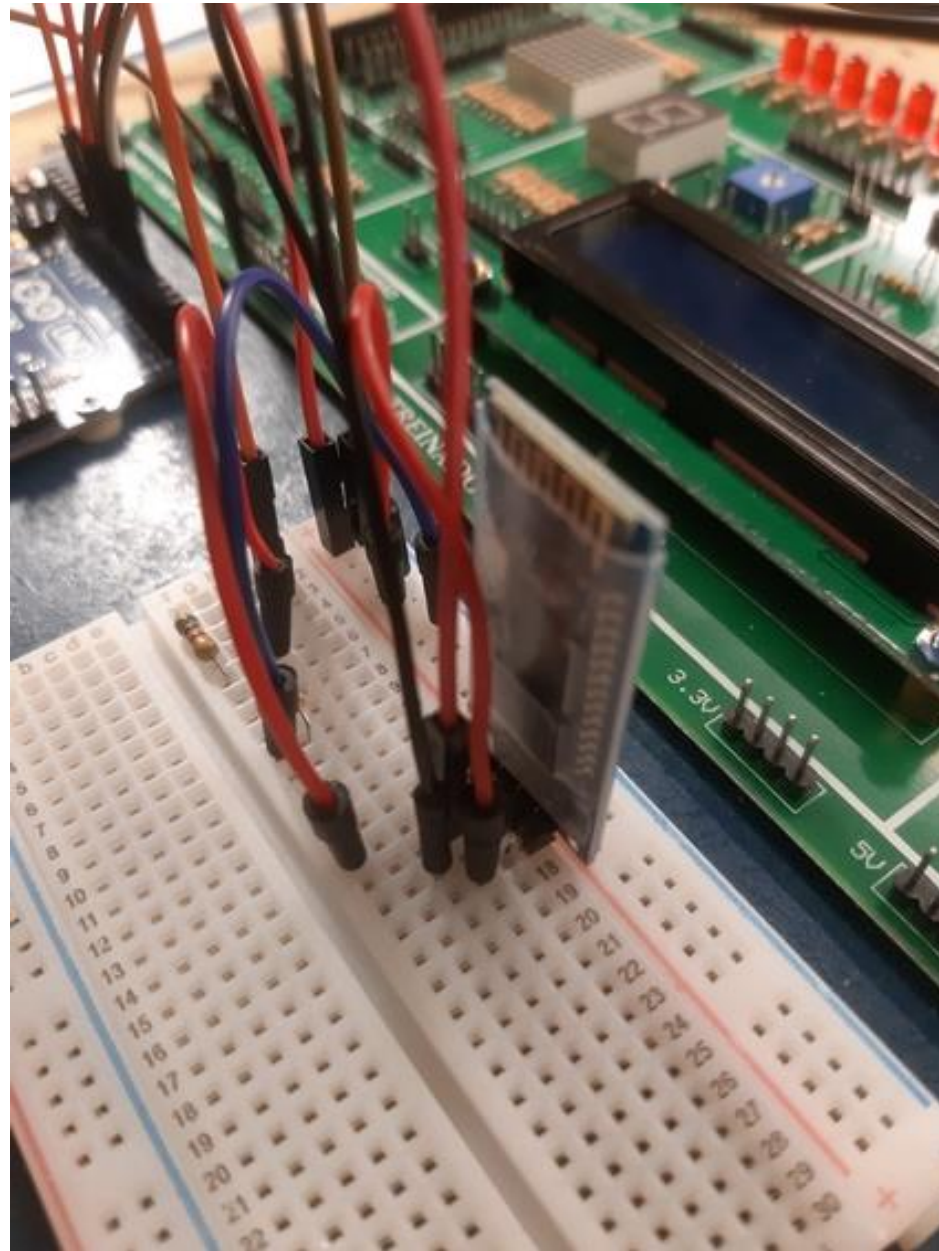
Tx do BT ao Rx do Arduino
Rx do BT ao Tx do Arduino

Como os terminais A0 e A1 do Arduino são usados para uploads dos códigos, para evitar conflitos torna-se mais seguro atribuir essas funções a outros terminais do Arduino, utilizando então o divisor de tensão para alimentar o terminal Rx e, através da programação dos códigos configurando esses pinos/terminais como veremos a seguir.

A figura a seguir mostra o aspecto da montagem no Treinador Arduino:







Códigos para o sketch:

```
#include <SoftwareSerial.h> //para conexão com o BT
SoftwareSerial serialBT(8,9); //portas seriais do BT (TX e RX)
#define led1Y 4 // define o pino digital para o led amarelo
#define led2B 5 // define o pino digital para o led azul
#define led3G 6 // define o pino digital para o led verde
#define led4R 7 // define o pino digital para o led vermelho
int valueBT; // variável para leitura do valor enviado pelo BT
void setup()
{
  serialBT.begin (9600); //inicia o serial do BT
  pinMode(led1Y, OUTPUT);
  pinMode(led2B, OUTPUT);
  pinMode(led3G, OUTPUT);
  pinMode(led4R, OUTPUT);
  digitalWrite (led1Y, LOW); //garantir leds desligados ao iniciar
  digitalWrite (led2B, LOW);
  digitalWrite (led3G, LOW);
  digitalWrite (led4R, LOW);
}
void loop() {
  if (serialBT.available()) //se BT estiver ok será lido o valor do serial
  {
    valueBT = serialBT.read();
  }
  if (valueBT=='A') // ligar o led amarelo
  {
    digitalWrite (led1Y, HIGH);
  }
  if (valueBT=='a') // desligar o led amarelo
  {
    digitalWrite (led1Y,LOW);
  }
}
```

```
    }  
    if (valueBT=='B') // ligar o led azul  
    {  
        digitalWrite (led2B, HIGH);  
    }  
    if (valueBT=='b') // desligar o led azul  
    {  
        digitalWrite (led2B,LOW);  
    }  
    if (valueBT=='C') // ligar o led verde  
    {  
        digitalWrite (led3G, HIGH);  
    }  
    if (valueBT=='c') // desligar o led verde  
    {  
        digitalWrite (led3G,LOW);  
    }  
    if (valueBT=='D') // ligar o led vermelho  
    {  
        digitalWrite (led4R,HIGH);  
    }  
    if (valueBT=='d') // desligar o led vermelho  
    {  
        digitalWrite (led4R,LOW);  
    }  
    if (valueBT=='E') // leds piscando  
    {  
        digitalWrite (led1Y,HIGH);  
        digitalWrite (led2B,HIGH);  
        digitalWrite (led3G,HIGH);  
        digitalWrite (led4R,HIGH);  
        delay(500);  
        digitalWrite (led1Y,LOW);
```

```

digitalWrite (led2B,LOW);
digitalWrite (led3G,LOW);
digitalWrite (led4R,LOW);
delay(500);
}
if (valueBT=='e') //somente os leds azul e vermelho piscam
{
    digitalWrite (led1Y, LOW);
    digitalWrite (led2B, HIGH);
    digitalWrite (led3G, LOW);
    digitalWrite (led4R, HIGH);
    delay(500);
    digitalWrite (led1Y, LOW);
    digitalWrite (led2B, LOW);
    digitalWrite (led3G, LOW);
    digitalWrite (led4R, LOW);
    delay(500);
}
}

```

A partir daí podemos acionar o aplicativo pelo celular.

O aplicativo que está sendo utilizado é o Arduino Bluetooth Control – “Giumig Apps”



Arduino bluetooth controller

Giumig Apps • Ferramentas

📌 Instalado

Abrindo o aplicativo:



Analisando as figuras a seguir após abrir o aplicativo:

FIGURA 1: dispositivo HC-06 pareado, aguardando a conexão

FIGURA 2: a opção de conexão foi por "Terminal Mode" (digitam-se parâmetros como letras, números ou caracteres)

FIGURA 3: abertura da tela de comando para digitar o comando "type command". Conforme programação dos códigos as letras "A" até "E" (maiúsculas) e "a" até "e" (minúsculas) serão responsáveis pelos comandos

FIGURA 4: foi digitada a letra "A", aguardando a inserção

FIGURA 5: após a inserção, teclando-se em "Concluído" o led amarelo deve acender

FIGURA 6: observa-se que o último comando válido é a letra "E" no qual todos os leds piscarão

Pode-se observar que o aplicativo utilizado como exemplo é bastante simples com várias opções de comando:

CONTROLLER MODE
SWITCH MODE
DIMMER MODE
TERMINAL MODE

A opção pelo "Terminal Mode" foi pela sua praticidade a simplicidade apenas. Outras formas de controle poderão ser utilizadas.



Figura 1

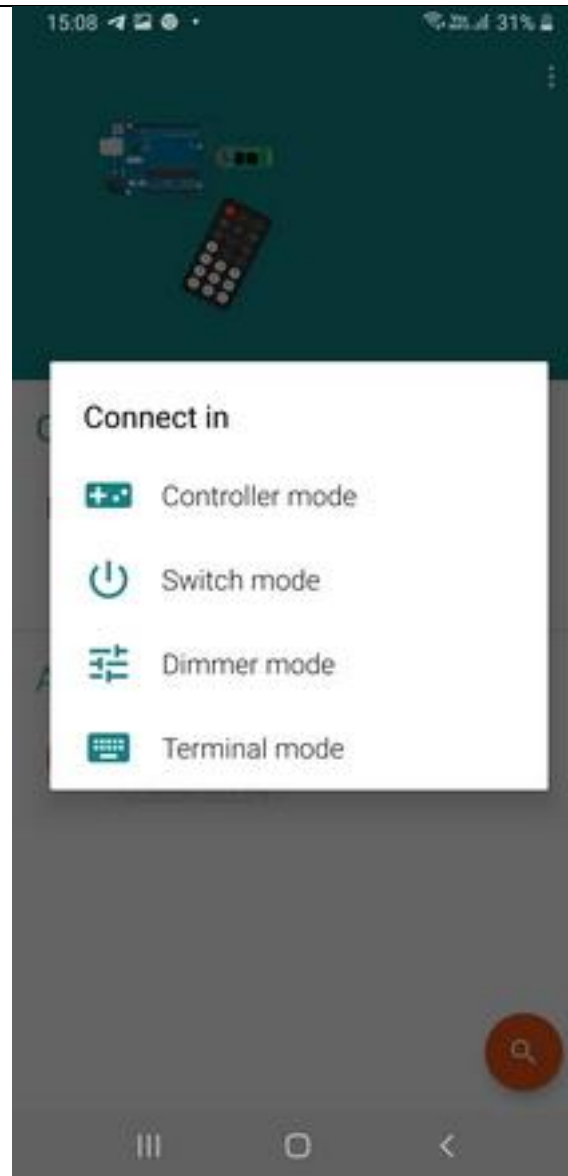


Figura 2



Figura 3

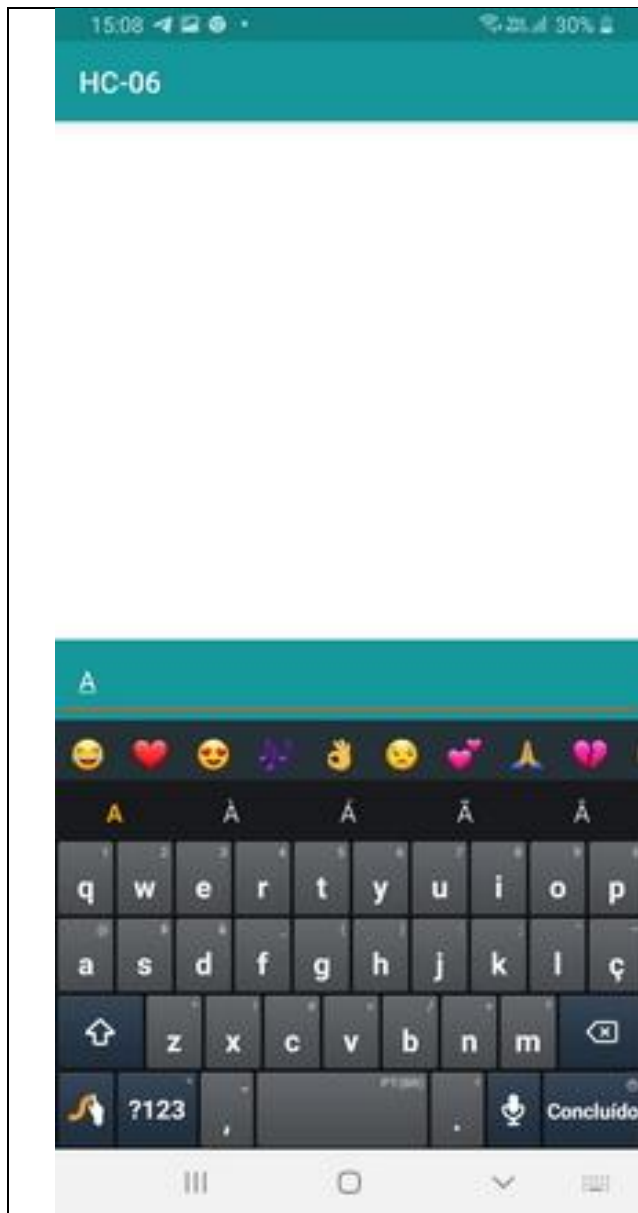


Figura 4

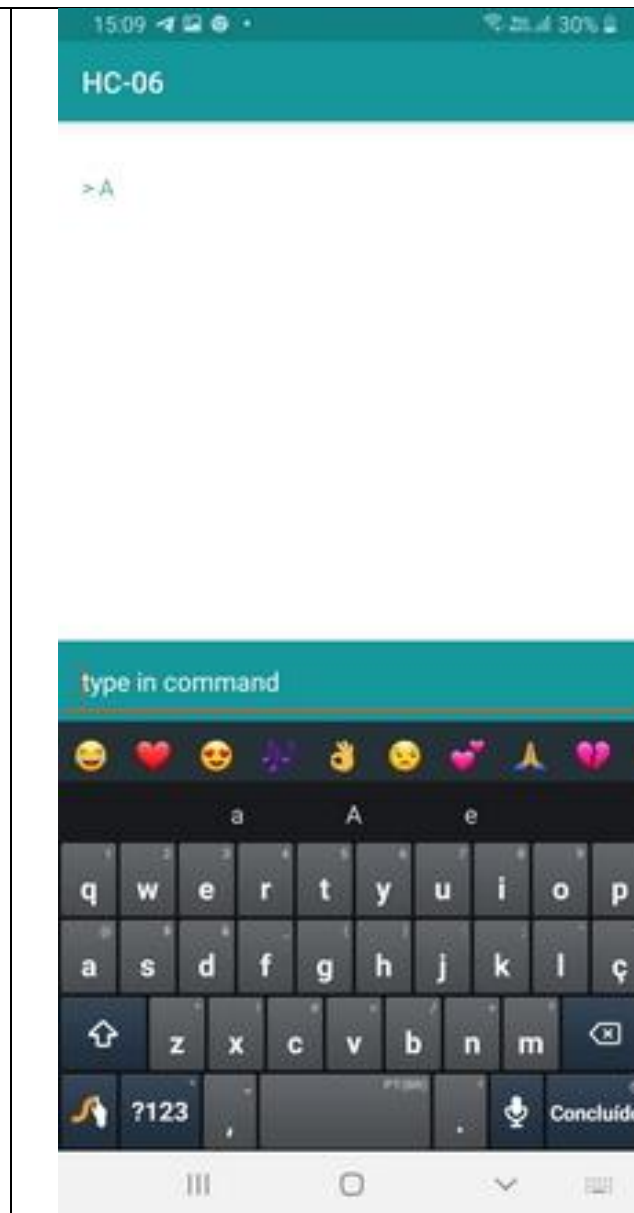


Figura 5

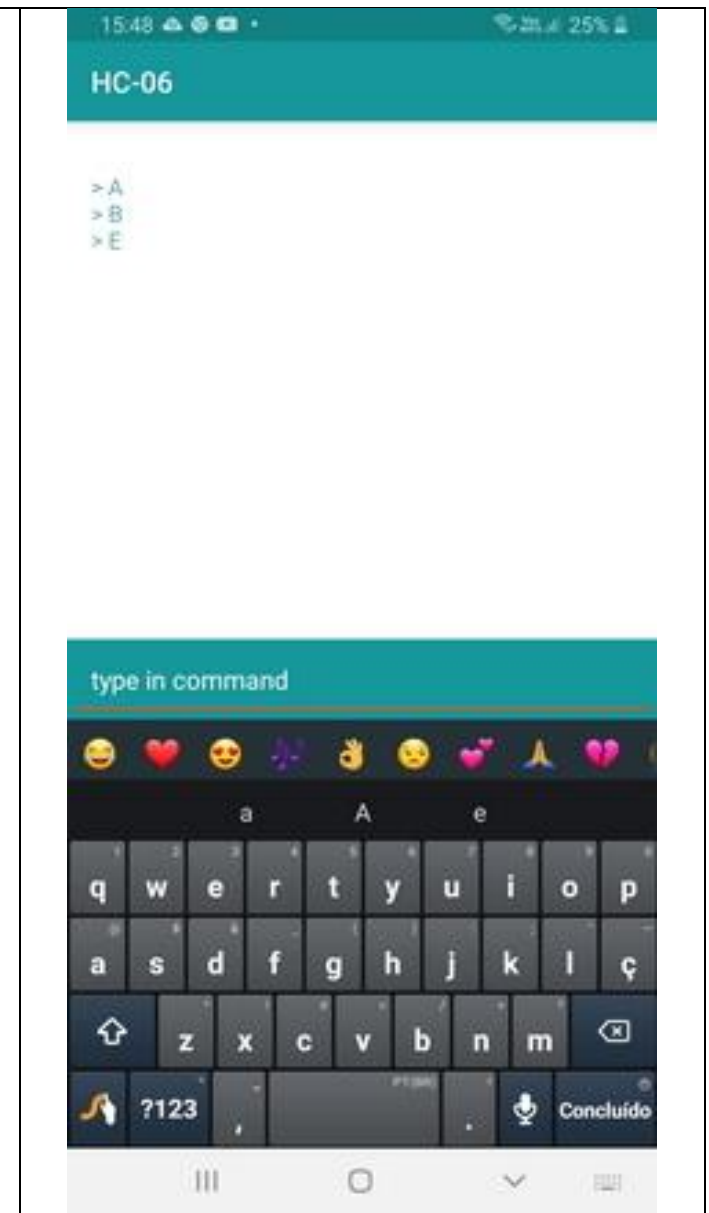


Figura 6

Quando se utiliza o modo terminal "Terminal Mode" para inserir comandos, várias combinações podem ser feitas para acionar o conjunto de leds.

É bom lembrar que no lugar dos leds podem ser utilizados quaisquer tipos de dispositivos, daí percebe-se então a enorme quantidade de opções para se desenvolver um projeto com módulo "bluetooth".

Veja abaixo uma tabela com algumas sugestões e respectivos resultados, utilizando as opções da programação dos códigos:

Comando	Resultado
DADO	Leds VM e AM acendem
dado	Todos os leds apagados
ALFACE	Todos os leds piscam
ALFACe	Apenas os leds AZ e VM piscam
ABCD	Todos os leds acendem
DCBA	Todos os leds acendem
ABCDE	Todos os leds piscam
EDCBA	Todos os leds acesos
ABCDe	Os leds AZ e VM piscam

Na primeira linha vemos que as letras A e D (maiúsculas) estão contidas na programação, daí então apenas os leds amarelo e vermelho acenderem.

```
if (valueBT=='A') // ligar o led amarelo
{
  digitalWrite (led1Y, HIGH);
}
```

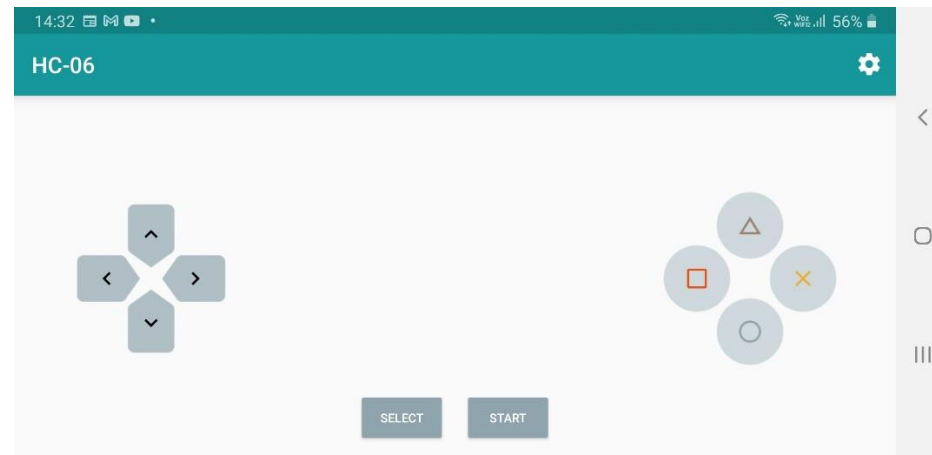
```
if (valueBT=='D') // ligar o led vermelho
```

```
{
  digitalWrite (led4R,HIGH);
}
```

Na última linha, com a combinação ABCDe, verifica-se que prevalece a letra “e” minúscula daí então teremos os leds azul e vermelho piscando.

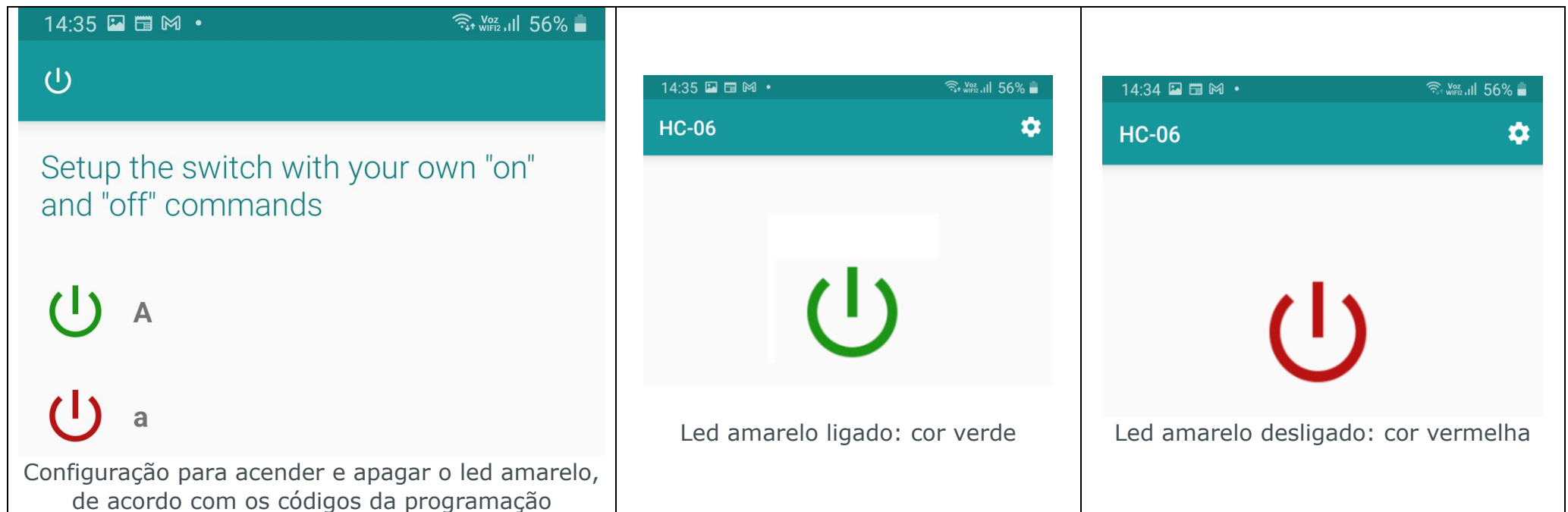
```
if (valueBT=='e') //somente os leds azul e vermelho piscam
{
  digitalWrite (led1Y, LOW);
  digitalWrite (led2B, HIGH);
  digitalWrite (led3G, LOW);
  digitalWrite (led4R, HIGH);
  delay(500);
  digitalWrite (led1Y, LOW);
  digitalWrite (led2B, LOW);
  digitalWrite (led3G, LOW);
  digitalWrite (led4R, LOW);
  delay(500);
}
```

Veja na figura a seguir os comandos disponíveis na opção “Controller Mode”

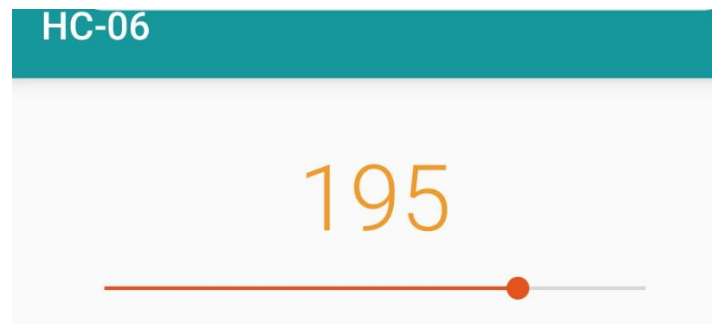


Observe que é uma opção que se assemelha aos comandos de um joystick.

A figura a seguir mostra a opção "Switch Mode" – apenas ligar e desligar um dispositivo



A figura a seguir mostra a opção "Dimmer Mode"



Essa opção permite o controle de qualquer dispositivo dimerizável, como por exemplo, controle de luminosidade de uma lâmpada ou de ambientes.

A figura mostra a grandeza 195, lembrando que essa variação pode ocorrer entre 0 e 255