

Etapas e dificuldades

Victor Vieira Paulino

Arthur Cicuto Pires

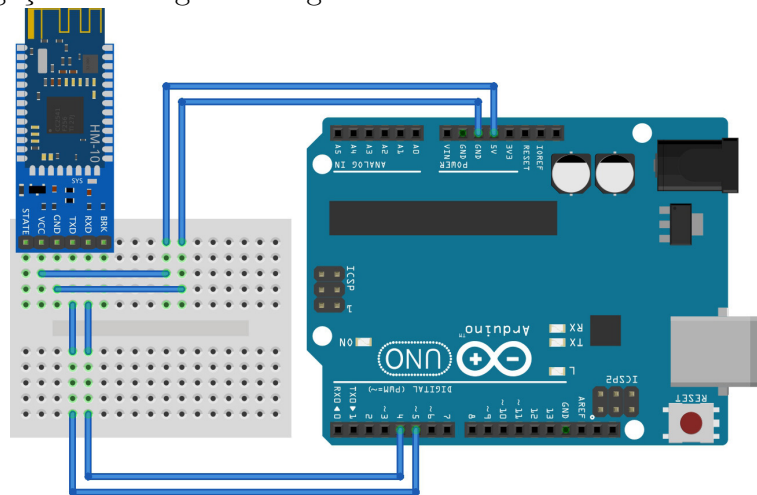
14 de agosto de 2017

1 Módulo do Ônibus

1.1 Hardware

1. HM-10 - Bluetooth 4.0 BLE module
2. Arduino Uno

Arduino Uno é utilizado apenas como ponte para configurar o módulo HM-10. A ligação deve seguir o diagrama abaixo.



1.2 Software

- Arduino IDE 1.8.3 ou superior.

1.3 Configuração

Conecte o arduino Uno ao computador e compile o código abaixo utilizando a IDE do arduino.

```
#include <SoftwareSerial.h>

// SoftwareSerial (TX / RX)
SoftwareSerial mySerial(7, 8);

// Serial value
String val;

// Receive info
boolean receiveInfo = false;

// Bauds for detectBleBaudRate function
long bauds[] = {
    // major
    9600, 57600, 115200,

    // others
    19200, 38400, 4800, 2400, 1200, 230400
};

// Detect BLE Baud Rate
bool detectBleBaudRate() {
    Serial.println("-----");
    Serial.println("Detecting BLE baud rate:");

    for (int i=0; i < (sizeof(bauds) / sizeof(long)); i++) {
        Serial.write("Checking baud rate: ");
        // Checking baud rate
        long cur_baud = bauds[i];
        Serial.println(cur_baud, DEC);

        mySerial.begin(cur_baud);
        mySerial.write("AT");
        mySerial.flush();

        delay(50);

        String response = mySerial.readString();

        if (response == "OK") {
            Serial.println("-----");
            Serial.print("BLE Baud Detected: ");
            Serial.println(cur_baud);
            Serial.println("-----");
        }
    }
}
```

```

        return true;
    } else {
        mySerial.end();
    }
}

detectBleBaudRate();
}

// Uncomment function to wake up bluetooth or discover BLE
// Baud Rate
void setup() {
    mySerial.begin(9600);
    Serial.begin(9600);
}

void wakeup() {
    Serial.println("-----");
    for (int i = 0; i < 30; i++) {
        mySerial.write("
12345678901234567890123456789012345678901234567890123456789012345678
");
    }

    Serial.println("Wake up command has been sent to HM-10");
    Serial.println("-----");
}

// Reads strings that are sent to mySerial
void loop() {
    while (Serial.available() > 0) {
        delay(10); // Delay needed to recognize key and value
                    properly

        String val = Serial.readString();
        Serial.print("Serial sent: ");
        Serial.println(val);

        // Receive info from HM-10
        receiveInfo = true;

        // If you need to wake up the HM-10 sensor, send wake
           command
        if (val == "wake") {
            wakeup();
            receiveInfo = false;
        }
    }
}

```

```

// If you want to detect BLE Baud Rate, send detect
// command
if (val == "detect") {
    detectBleBaudRate();
    receiveInfo = false;
}

mySerial.print(val);
mySerial.flush();

delay(50);

if (receiveInfo) {
    // Read string from HM-10 (Software Serial)
    String response = mySerial.readString();

    Serial.print("mySerial received: ");
    Serial.println(response);

    Serial.println("-----");
}
}
}

```

Após compilador, utilizando o Serial Monitor da IDE, execute os comandos AT na seguinte ordem:

Obs: Quanto menor o tempo de envio, maior a economia de energia.

1. AT+RENEW //Coloca nos padrões de fábrica
2. AT+RESET //Reinicia para aplicar os padrões de fábrica
3. AT+MARJ0xNNNN //Define o valor Marjor
4. AT+MINO0xNNNN //Define o valor Minor
5. AT+NAMEMeuBeacon //Define o nome do Beacon
6. AT+ADVI5 //Define tempo de envio. 5 = 546.25 milisegundos
7. AT+ADTY3 //Define como não pareável
8. AT+IBEA1 //Habilita como Beacon
9. AT+DELO2 //Configura para apenas emitir sinal
10. AT+PWRM0 //Habilita auto-sleep para economizar energia
11. AT+RESET

Após configurado, pode ser ligado em uma bateria 3v para utilização.

1.4 Referências

HM-10 Bluetooth 4.0 BLE module Datasheet

Arduino IDE

Repositório da Metractive - Como construir Beacons

2 Módulo do ponto de ônibus

teste

3 Aplicativo

teste

4 Web service

Falar sobre o MEAN stack e sobre as dificuldades que estou encontrando sobre como trabalhar com cada tecnologia da pilha.