

Etapas e dificuldades

Victor Vieira Paulino

Arthur Cicuto Pires

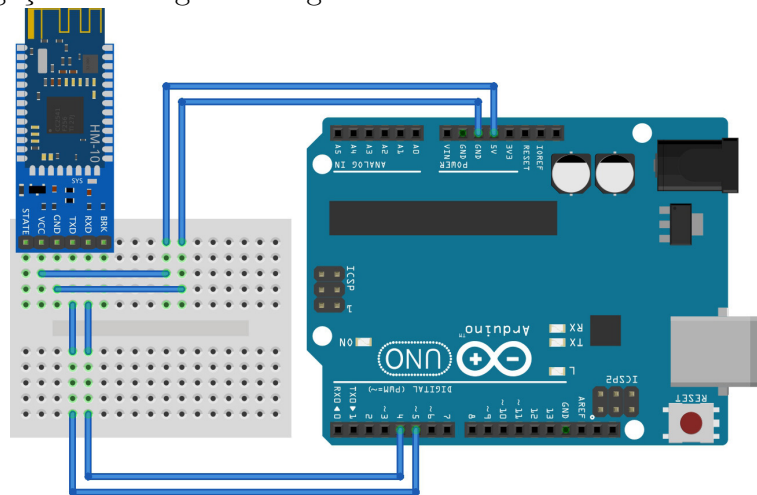
15 de agosto de 2017

1 Módulo do Ponto de 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("-----");
        }
    }
}
```



```

// 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

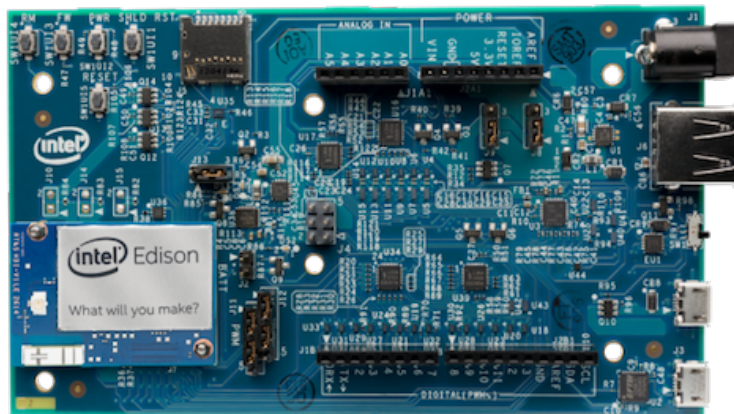
Repositório da Metractive - Como construir Beacons

2 Módulo do Ônibus

2.1 Hardware

- Intel Edison
- Raspberry Pi 3
- Tela LCD 7” (em breve)
- NEO u-blox 6 GPS Modules

2.1.1 Intel Edison



Inicialmente foi adotado o Intel Edison com placa de expansão arduino. Foi escolhido devido a fácil acesso a um exemplar e ótimo hardware. Ele conta com WiFi, Bluetooth, portas I/O, processador Intel Atom de 500 MHz, 1GB de memória RAM DDR3 e 4GB eMMC.

Sua utilização foi fácil e não obtivemos nenhuma dificuldade em instalar o sistema que escolhemos.

Problemas encontrados em adotar como solução:

Preço

Embora tenha um ótimo hardware e uma empresa séria por trás da sua construção, o preço, em 07/2017, que gira em torno de R\$ 600,00, não justifica

sua adoção como a melhor solução para o projeto já que existe alternativas com preços melhores e bom desempenho.

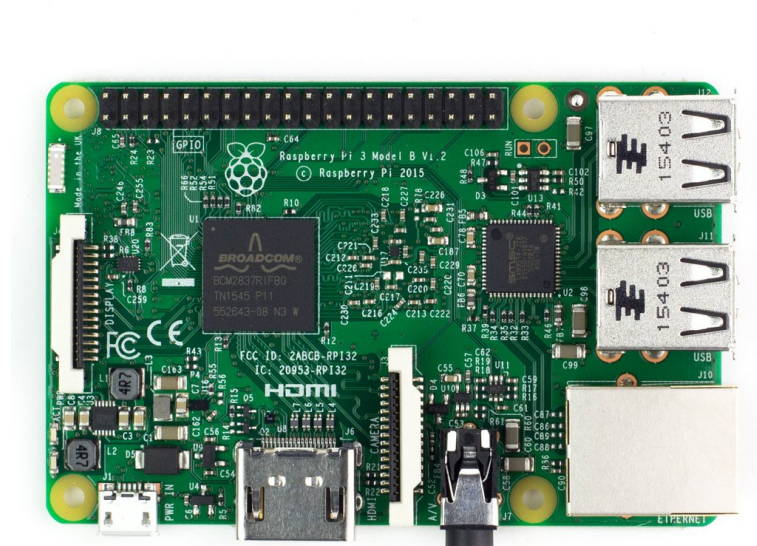
Ausência de controlador gráfico

Uma das features do projeto é emitir alertas visuais para o motorista por meio de telas LCDs. A placa Intel Edison só nos permite fazer alertas visuais utilizando LEDs e afins.

Descontinuidade da placa pela Intel

em 07/2017, a Intel anunciou a descontinuidade do desenvolvimento de algumas placas que fabrica. O Intel Edison foi uma delas.

2.1.2 Raspberry Pi 3



Testes realizados no Raspberry Pi 3 demonstraram ser uma boa alternativa ao Intel Edison. Foi fácil a instalação do sistema e a placa vem com saída HDMI permitindo utilizar telas LCD para fazer os alertas visuais. Seu preço, em 08/2017, gira em torno de R\$ 150,00, 1/4 do preço do Intel Edison. Seu hardware contém boas especificações:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU

- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

Embora ele tenha um hardware com especificações superiores ao Intel Edison, não houve ganho de desempenho ao rodar o sistema, devido a ausência de algoritmos complexos no sistema. Assim, a grande vantagem de se utilizar o Raspberry Pi 3 ao invés do Intel Edison, é seu baixo custo e recurso de chip gráfico.

2.1.3 NEO u-blox 6 GPS Modules



Para realizar o rastreamento do ônibus foi adotado o módulo NEO u-blox 6 GPS Modules, devido a compatibilidade com as placas que contém o sistema embarcado e preço acessível.

Localização

Uma característica desse módulo é trabalhar com GPS, fazendo comunicação direta com no mínimo 3 satélites para triangular sua posição com mais precisão. Alguns módulos disponíveis no mercado trabalham com A-GPS, que usam torres de telefonia móvel para conhecer sua posição. O uso do GPS trás maior precisão, porém demora mais para estabelecer conexão com satélites. O A-GPS fornece a localização com menor tempo, porém com menor precisão e a um custo mais alto.

Comunicação

O módulo realiza comunicação UART (Universal Asynchronous Receiver/-Transmitter), o que permite fácil comunicação com as placas utilizadas para testes.

Preço

Seu preço, em 08/2017, gira em torno de R\$ 60,00 e pode ser encontrado com facilidade na internet para venda.

2.2 Software

2.2.1 Sistema Operacional

Android Things Em 2016 o Google anunciou o Android Things, uma versão do Android voltada para IoT (Internet of Things). Ele é, atualmente, uma versão do Android Marshmallow reduzida. Sua escolha foi devido a facilidade de embarcar em placas como o Raspberry Pi e Intel Edison, e a variedade de recursos que já estão disponíveis no SO que facilitam o desenvolvimento do módulo, como o recurso LocationManager. **[Detalhar mais essa parte]**

2.2.2 IDE

Foi escolhido o Android Studio como IDE do projeto. Ela é desenvolvida pela IntelliJ e tida pelo Google como ferramenta oficial de desenvolvimento para aplicativos Android.

2.2.3 Linguagem

O Google tem duas linguagens de primeiro nível para desenvolvimento Android: Java e Kotlin. Para esse projeto adotamos a linguagem Kotlin, que possui sintaxe muito simplificado em comparação ao Java. Embora Java tenha sido a primeira linguagem oficial para desenvolvimento, Kotlin oferece acesso aos mesmos recursos do sistema. Algumas bibliotecas disponíveis, desenvolvidas por terceiros, ainda não migraram para o Kotlin, obrigando a implementar algumas classes em Java. Como Kotlin tem interoperabilidade com Java, não existe nenhum impeditivo de utilizar Kotlin e eventualmente alguma classe Java.

2.3 Referências

Site Oficial Intel Edison

Datasheet Intel Edison

Anúncio do fim da produção do Intel Edison

Site Oficial Raspberry Pi

Datasheet Raspberry Pi 3

Datasheet NEO u-blox 6 GPS Modules

Site Oficial Android Things

Configuração do Android Things no Intel Edison

Configuração do Android Things no Raspberry Pi 3

3 Aplicativo

3.0.1 IDE

Foi escolhido o Android Studio como IDE do projeto. Ela é desenvolvida pela IntelliJ e tida pelo Google como ferramenta oficial de desenvolvimento para aplicativos Android.

3.0.2 Linguagem

O Google tem duas linguagens de primeiro nível para desenvolvimento Android: Java e Kotlin. Para esse projeto adotamos a linguagem Kotlin, que possui sintaxe muito simplificado em comparação ao Java. Embora Java tenha sido a primeira linguagem oficial para desenvolvimento, Kotlin oferece acesso aos mesmo recursos do sistema. Algumas bibliotecas disponíveis, desenvolvidas por terceiros, ainda não migraram para o Kotlin, obrigando a implementar algumas classes em Java. Como Kotlin tem interoperabilidade com Java, não existe nenhum impeditivo de utilizar Kotlin e eventualmente alguma classe Java.

4 Web service

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