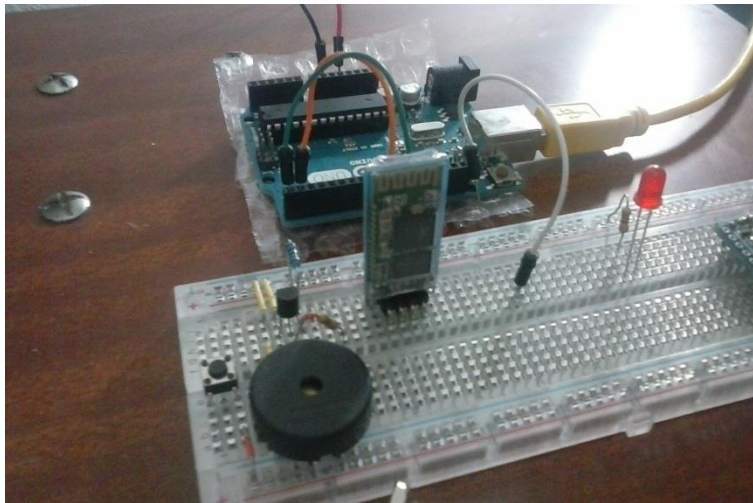


Smartphone diz On/Off via Bluetooth ao Arduino



Neste tutorial é explorado os básicos para colocar um Android Smartphone a comunicar com o Arduino via bluetooth. Uma simples solução de enviar um comando on/off a partir de uma App que corre em Android para um terminal de Bluetooth que por sua vez está ligado ao Arduino. O comando enviado poderá servir para activar uma rotina que irá comutar um relé (ver tutorial de relays), acender um LED, ligar o sistema de som, ect..

A intenção do tutorial é apenas explicar e fornecer algumas ferramentas básicas que poderão ser aplicadas como bloco na construção de um projecto DIY

É de facto interessante conseguir pôr um microcontrolador a agir com o mundo exterior. Mas agora, isso já não é o suficiente. Queremos igualmente conseguir pôr o microcontrolador a comunicar com o mundo! Existem várias opções: via Ethernet (RJ45 ou um simples “cabo USB”), Wi-Fi (equipamento ainda dispendioso), Zigbee (um pouco overkill para aplicações caseiras e um preço pouco convidativo) e finalmente o Bluetooth. Sim, o Bluetooth além de barato (4-7€) é uma tecnologia de radio frequência (wireless), que nos permite pôr o smartphone a comunicar com o Arduino.

1ª Ponto: A estratégia

- Uma App Android envia uma “palavra” via bluetooth para o módulo JY-MCU
- Módulo bluetooth reencaminha para a porta série do Arduino (pino RX)
- Arduino lê a palavra e uma estrutura de decisão decide a acção a efectuar

2ª Ponto: Hardware

Para isso necessitamos de um terminal Bluetooth, eu utilizo o JY-MCU Bluetooth que comprei no Dx.com por 4.90 euros (também disponível no ebay). O restante material é quase que dispensável referir: o Arduino (eu recomendo sempre a implementação da solução “[Aergia Arduino](#)”), breadboard, alguns [cabos dupont](#), resistências e um LED para podermos testar código.

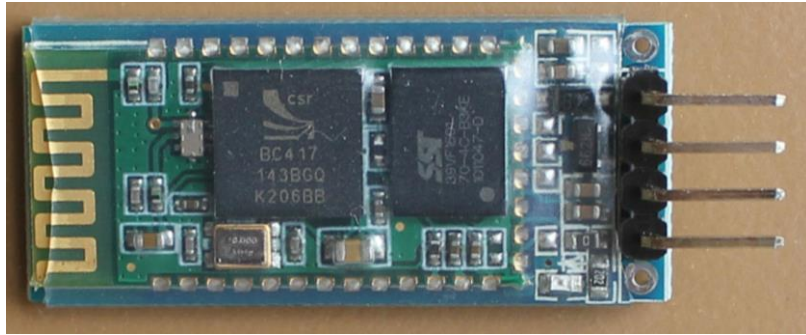


Ilustração 1 e 2 - Módulo JY-MCU

A comunicação do terminal Bluetooth com o Arduino é realizada através de “comunicação série”, usualmente denominada por USART (*universal asynchronous receiver/transmitter*) utilizando os pinos RX e TX. Não irei perder muito tempo neste tópico, é contudo valioso possuir alguns conhecimentos básicos do funcionamento da comunicação série. Temos pelo menos de saber que a ligação entre terminais é feita da seguinte forma:

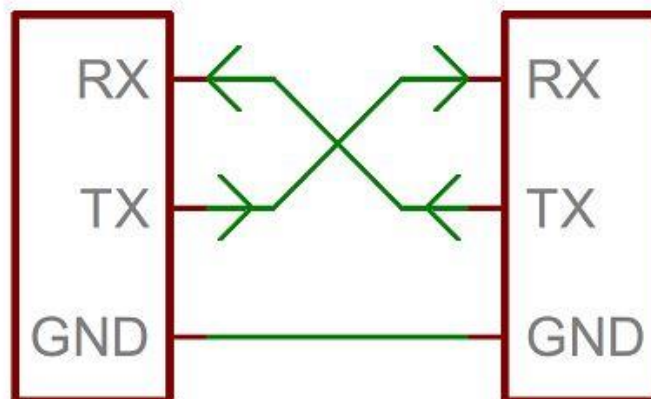


Ilustração 2 - Esquema de ligação

É necessário ter especiais cuidados na utilização do JY-MCU nomeadamente na tensão de alimentação visto haver no mercado versões a 3.3V e 5V. É de referir que o JY-MCU tem a marcação de 3.3V Level no pino TX (Transceiver), o que não haverá problemas visto que o Arduino lida com TTL (mais do que 2.2V corresponde a HIGH). Antes de efectuar a ligação do JY-MCU é necessário fazer o upload do Sketch para o Arduino, visto que enquanto um módulo estiver ligado a RX/TX não é possível o upload de código via USB. As funções dos pinos KEY e STATE não serão exploradas neste tutorial.

A ligação pode ser efectuada da seguinte forma:

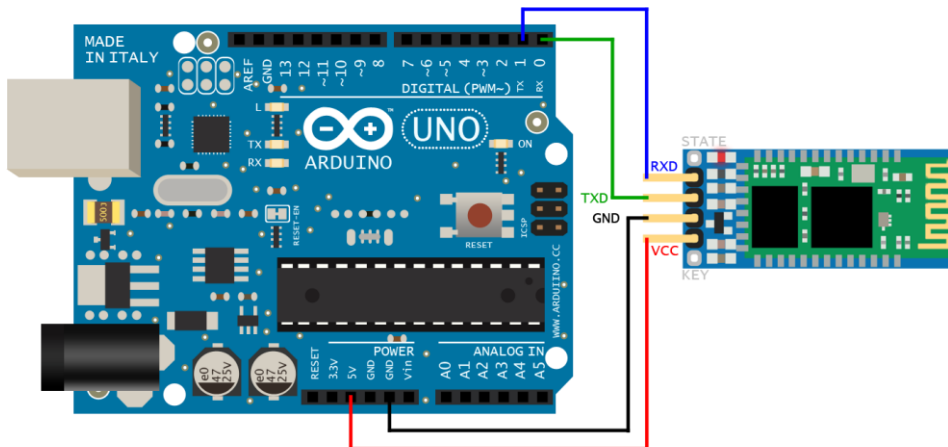


Ilustração 3 - Esquemático da ligação Arduino - Módulo Bluetooth

3ª Ponto: O código para o Arduino

O código é extremamente simples, e não é necessário qualquer adaptação por usarmos um terminal Bluetooth, é exactamente o mesmo que utilizaríamos caso a comunicação fosse feita através do "Processing" ou da janela de comando do Windows (via python).

```
/* Código de serial read */
```

```
int ledPin = 7; // pino ligado ao led, relay, ect..
```

```
String readString; // string com as instruções recebidas via Serial
```

```
void setup() {
```

```
  Serial.begin(9600); // iniciar Serial e definir vel. 9600 bits per second
```

```
  pinMode(ledPin, OUTPUT);
```

```
}
```

```
void loop() {
```

```
  while (Serial.available()) { // Quando recebe info via serial
```

```
    delay(3); // delay necessário sendo que 2ms é o suficiente
```

```
    char c = Serial.read(); // ler char recebido via serial
```

```
    readString += c; // construir string a partir do char recebido
```

```
  }
```

```
  if (readString.length() > 0) { // quando o tamanho da string é maior que zero
```

```
    Serial.println(readString); // print da string na janela de comando (não necessário)
```

```
    if (readString == "on") // caso tenhamos recebido a palavra "on"
```

```
    {
```

```
      digitalWrite(ledPin, HIGH); // coloca 5V no pino 7
```

```
    }
```

```
if (readString == "off")    // caso tenhamos recebido a palavra "off"
{
  digitalWrite(ledPin, LOW);    // coloca 0V no pino 7
}
readString=""; // esvazia o conteúdo, aka flush!
}
}
```

[Link do ficheiro de código *.ino](#)

De forma sucinta o que acontece é a leitura por parte do Arduino da palavra recebida via serial (pino RX). Caso a palavra seja "on" ele coloca o pino 7 no estado high, caso a palavra recebida seja "off" ele coloca o pino no estado "low". No pino poderá ser ligado por exemplo um relay que acende e apaga a luz da sala. O código pode ser estendido para aceitar mais e diferentes conjuntos de ordens. Agora só precisamos de nos certificar que enviamos as palavras correctas via App do Android para o módulo bluetooth.

4ª Ponto: App Android

Num futuro tutorial será explicado como criar e programar uma App para dispositivos Android (link). Para poder instalar uma aplicação (*.apk) sem ser via App Store é necessário permitir a instalação de "non-market software", ou seja software cuja segurança não foi verificada pela Apple.

Uma das formas de instalar uma App não verificada é através de um QR Code. Após a leitura do QR da ilustração 4 será reencaminhado para um ficheiro na Dropbox, efectue o download e clique no ficheiro "The_Aergia_App" e aceite as permissões clicando em "Instalar". A App ficará assim disponível no menu "aplicações" do seu dispositivo Android.



Ilustração 4 - QR Code para download da App

No contexto deste tutorial a App criada foi do mais simples possível, bastando clicar em “Ligar Bluetooth” para ligar ao módulo JY-MCU e depois basta clicar em “ON” ou “OFF” para que a palavra “on” ou “off” ser enviada.

Layout gráfico da Aplicação Android disponível para download (vista de um emulador):

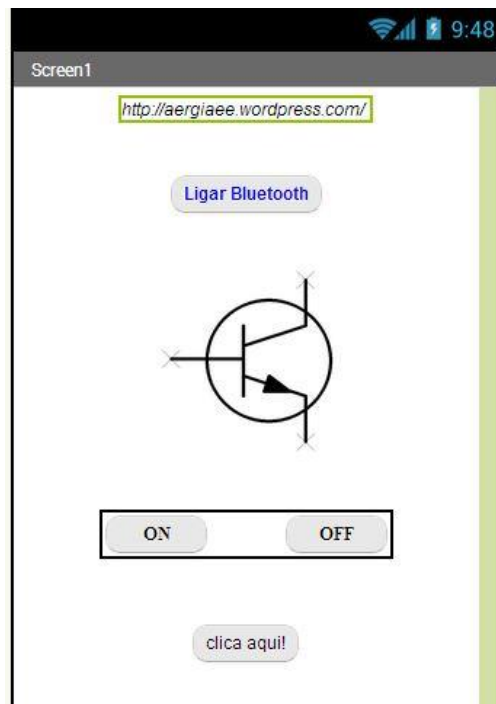


Ilustração 5 - Layout gráfico da Aplicação

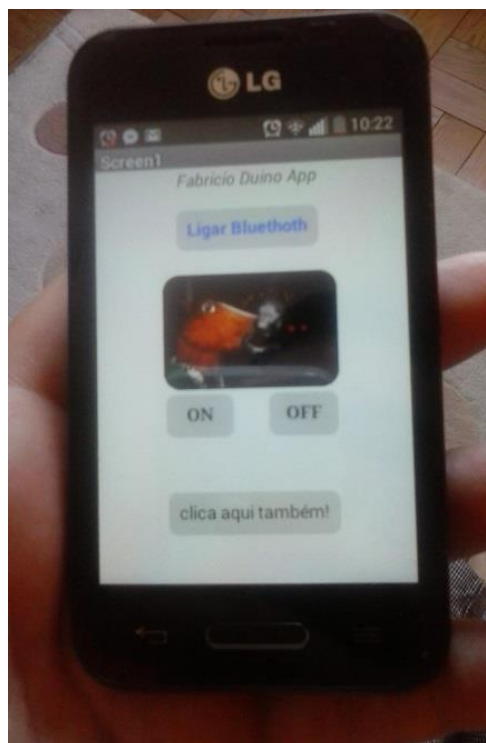


Ilustração 6 - Aplicação Android v2.0 (mais personalizada) no meu smartphone

5ª Ponto: Profit

Com recurso ao tutorial e à aplicação disponibilizada poderá facilmente integrar soluções wireless a qualquer projecto que já tenha ou venha a realizar.

Questões ou contribuições para o tutorial envie para:

fabricao.1137@gmail.com

Fabício Ferreira Antunes – Setembro de 2014

(<http://aergiaee.wordpress.com/>)