

**FIAP GRADUAÇÃO**

# TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

Arquiteturas Disruptivas, IA, IoT e Big Data

PROF. ANTONIO SELVATICI

## SHORT BIO



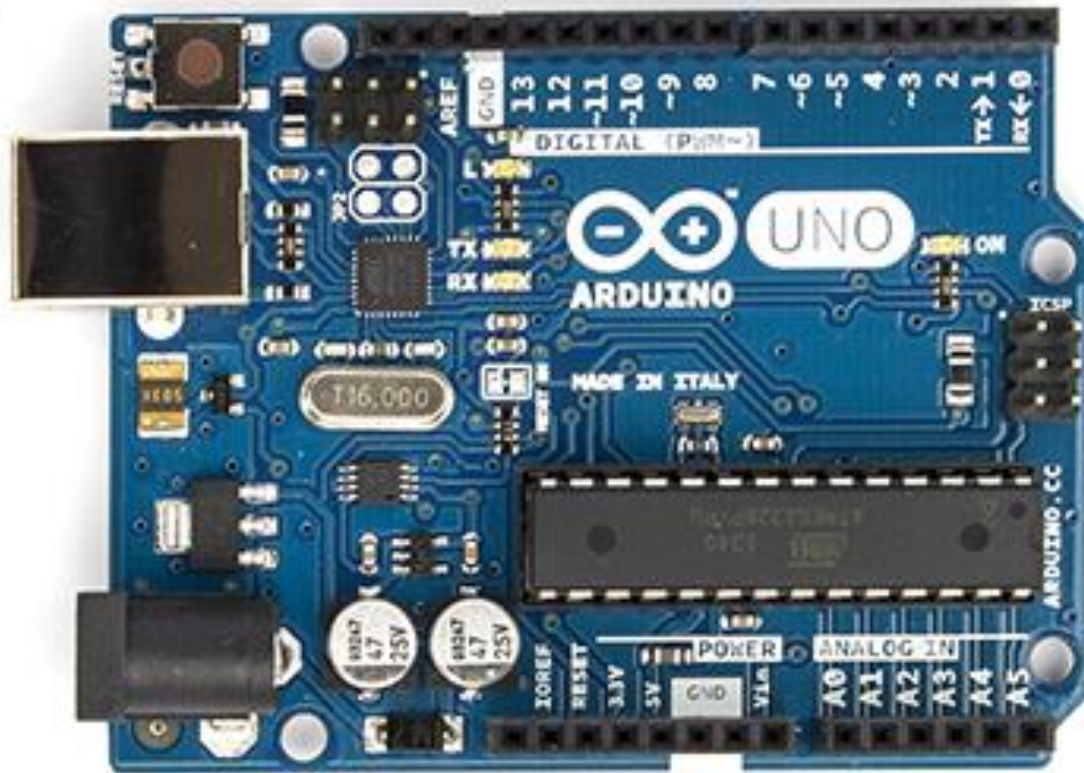
É engenheiro eletrônico formado pelo Instituto Tecnológico de Aeronáutica (ITA), com mestrado e doutorado pela Escola Politécnica (USP), e passagem pela Georgia Institute of Technology em Atlanta (EUA). Desde 2002, atua na indústria em projetos nas áreas de robótica, visão computacional e internet das coisas, aliando teoria e prática no desenvolvimento de soluções baseadas em Machine Learning, processamento paralelo e modelos probabilísticos. Desenvolveu projetos para Avibrás, IPT, CESP e Systax.

**PROF. ANTONIO SELVATICI**

[profantonio.selvatici@fiap.com.br](mailto:profantonio.selvatici@fiap.com.br)

# INTERNET DAS COISAS

## Arduino UNO



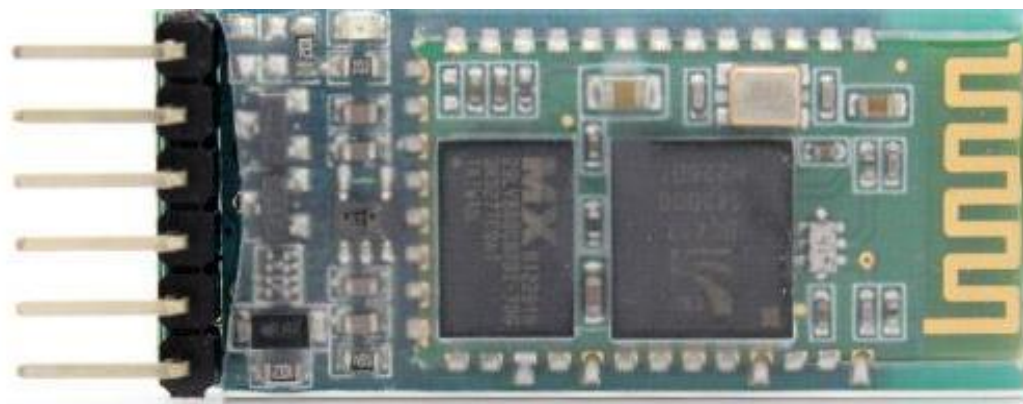
## A comunicação no Arduino

- A porta serial do Arduino Uno usa as portas 0 (Rx) e 1 (Tx) para receber e enviar dados respectivamente.
- Da mesma forma que esses dados trafegam pelo cabo USB, no caso do Arduino estar conectado ao computador, essas portas poderiam estar conectados a outros dispositivos de comunicação (como módulos ou shields) enviando dados para e recebendo dados desses dispositivos
- Caso a porta serial do Arduino já esteja ocupada, temos duas alternativas:
  - Usar outros modelos, como o Arduino Due, que possuem duas ou três portas seriais
  - Eleger outras portas digitais quaisquer como entrada e saída de dados através da biblioteca `SoftwareSerial`

## A comunicação Bluetooth

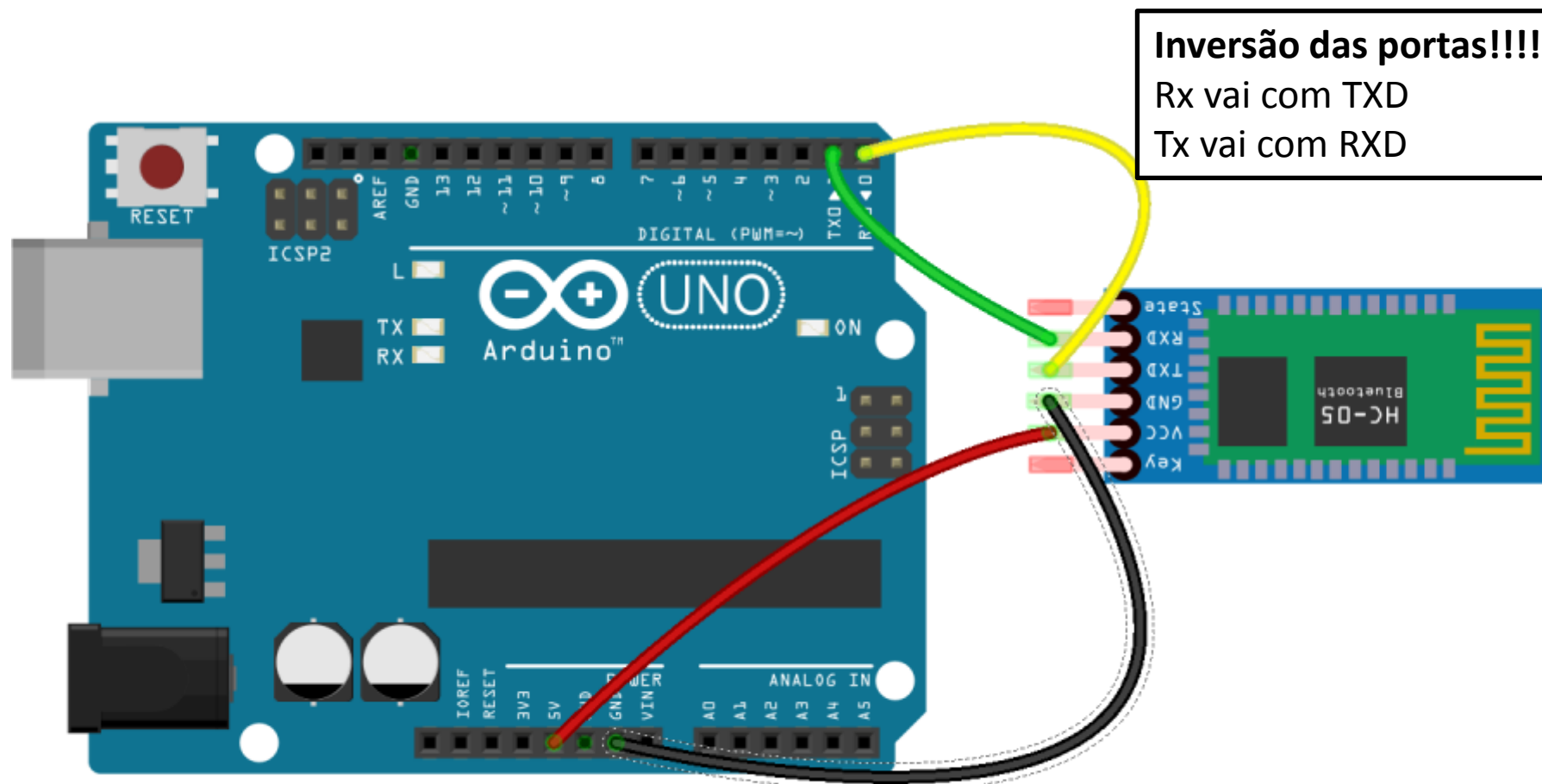
- O código que vimos até agora lê a porta serial para enviar e receber dados.
- Isso significa que esses dados serão enviados via cabo USB, no caso deste estar conectado, ou via Bluetooth, caso o Arduino possua um módulo Bluetooth instalado
- O módulo que usaremos aqui é o HC-05, que pode operar tanto no modo master (central) quanto slave (periférico), mas já vem configurado para operar como slave.
- Além de ligar os pinos de tensão, ele deve ter seus pinos TXD e RXD conectados às portas RX (0) e TX(1) do Arduino, respectivamente.
  - Não estranhe a aparente inversão: a porta serial do Arduino envia dados pelo terminal TX, que são recebidos pelo módulo Bluetooth no terminal RXD, e vice versa

## O Módulo Bluetooth HC-05, frente e verso

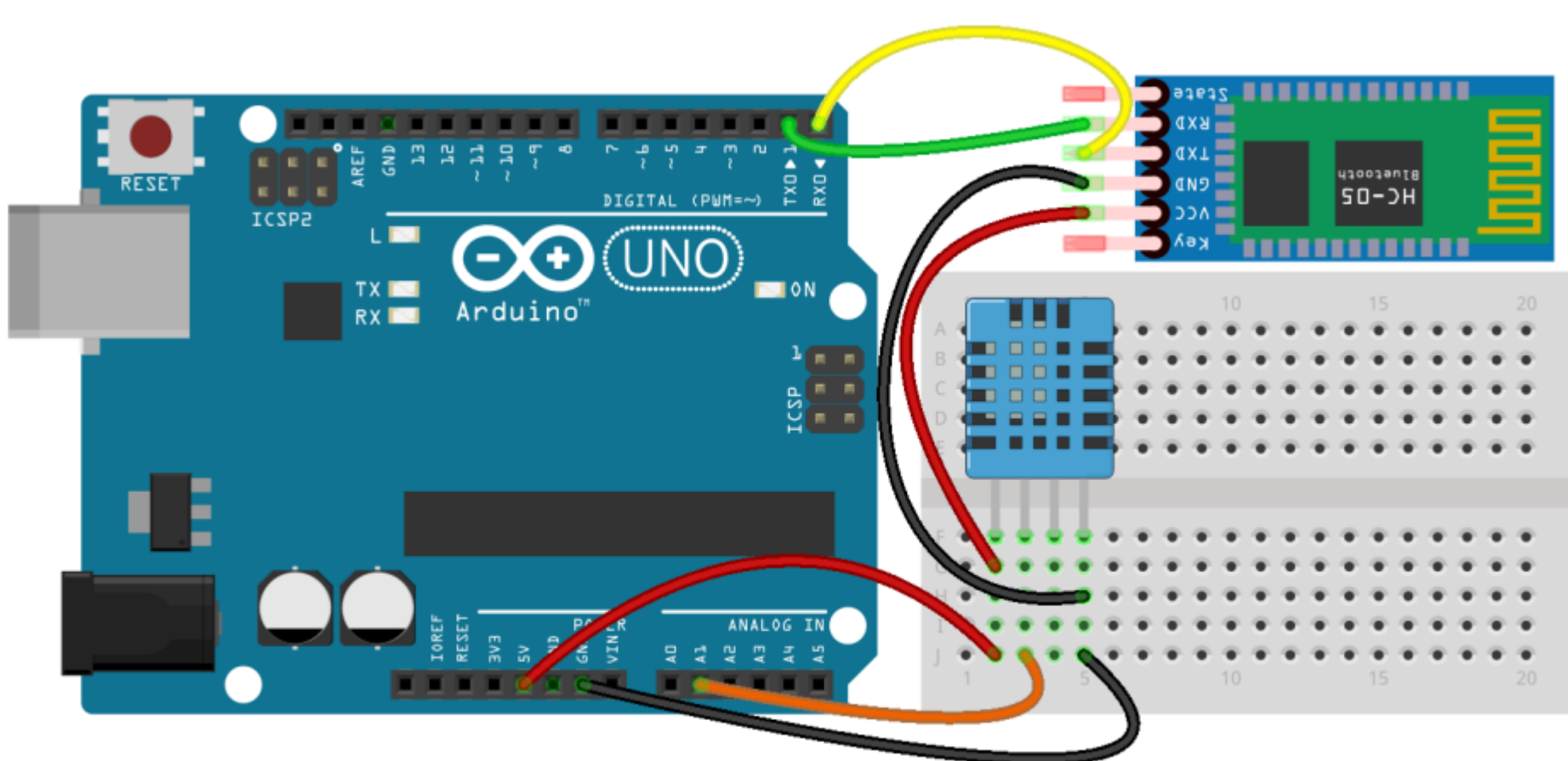




## Conexão do HC-05 para uso normal (substituição da porta serial)



## Exemplo: usando o HC-05 com o DHT-11



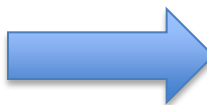
## Lendo o sensor DHT-11: nada muda

```
#include "DHT.h"
#define DHTPIN A1 // pino que estamos conectado
#define DHTTYPE DHT11 // DHT 11
DHT dht(DHTPIN, DHTTYPE); //Instanciação do objeto do sensor
void setup() {
    Serial.begin(9600);
    dht.begin();
}
void loop() {
    // A leitura da temperatura e umidade pode levar 250ms!
    float h = dht.readHumidity(); //Valor da umidade
    float t = dht.readTemperature(); //Valor da temperatura
    if (isnan(t) || isnan(h)) {
        Serial.println("Erro ao ler do DHT");
    } else {
        Serial.print("Umidade: ");
        Serial.print(h); Serial.print(" %\t");
        Serial.print("Temperatura: ");
        Serial.print(t); Serial.println(" °C");
    }
}
```

## Arduino Bluetooth Controller



```
> 1  
HC-06: LED: on  
> 0  
HC-06: LED: off  
> 0  
HC-06: LED: off  
> 0  
HC-06: LED: off
```



Área de saída  
de texto



Área de comandos

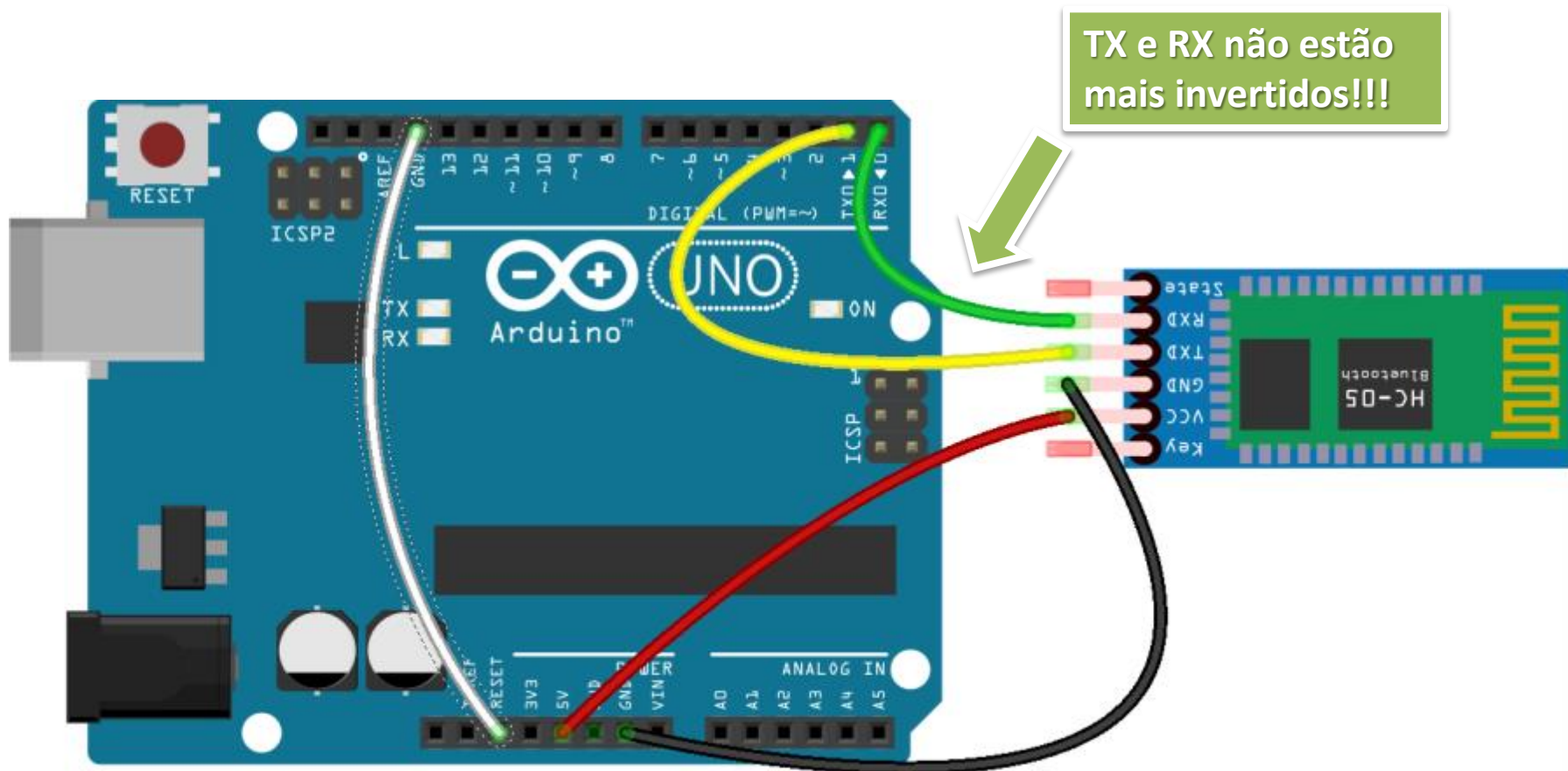
## Usando o módulo HC-05

- Depois que foi feita a configuração, podemos usar o HC-05 no modo *slave* sem alterar a programação do Arduino.
- Para iniciar a comunicação com seu computador ou celular Android, você deve ligar o módulo e parear com seu dispositivo, usando o PIN 1234.
  - Obs: A programação do smartphone Android para usar o Bluetooth é objeto de outra disciplina. Aqui vamos usar o aplicativo **Arduino Bluetooth Controller**
- **Importante!!!** Note que não será possível usar o cabo USB e o módulo Bluetooth simultaneamente nesta configuração.
  - Caso estiver usando o cabo USB para programar o Arduino, desconecte os seus pinos RX e TX do HC-05.

## ■ Configurando o módulo HC-05

- Se vários dispositivos estiverem presentes, não conseguimos saber quem é quem no App do Android
- Antes de usar o módulo HC-05, podemos configurá-lo para trocar o nome do dispositivo, por exemplo, e para isso precisamos usar o Arduino como **ponte** entre o computador e o HC-05
- Primeiro, devemos ligar a porta RESET do Arduino ao GND. Isso garante que o Arduino não irá interferir na comunicação do computador com o HC-05
- Então devemos ligar o RX do módulo ao RX do Arduino, e o TX do módulo ao TX do Arduino, pois para o chip FTDI o HC-05 fará o **papel** de Arduino
- Por fim, pressionamos o botão do HC-05 ao mesmo tempo que conectamos o cabo serial ao computador. Dessa forma, o HC-05 entrará no modo de comandos AT.

## Conexão do HC-05 para sua configuração



fritzing

## ■ Configurando o HC-05

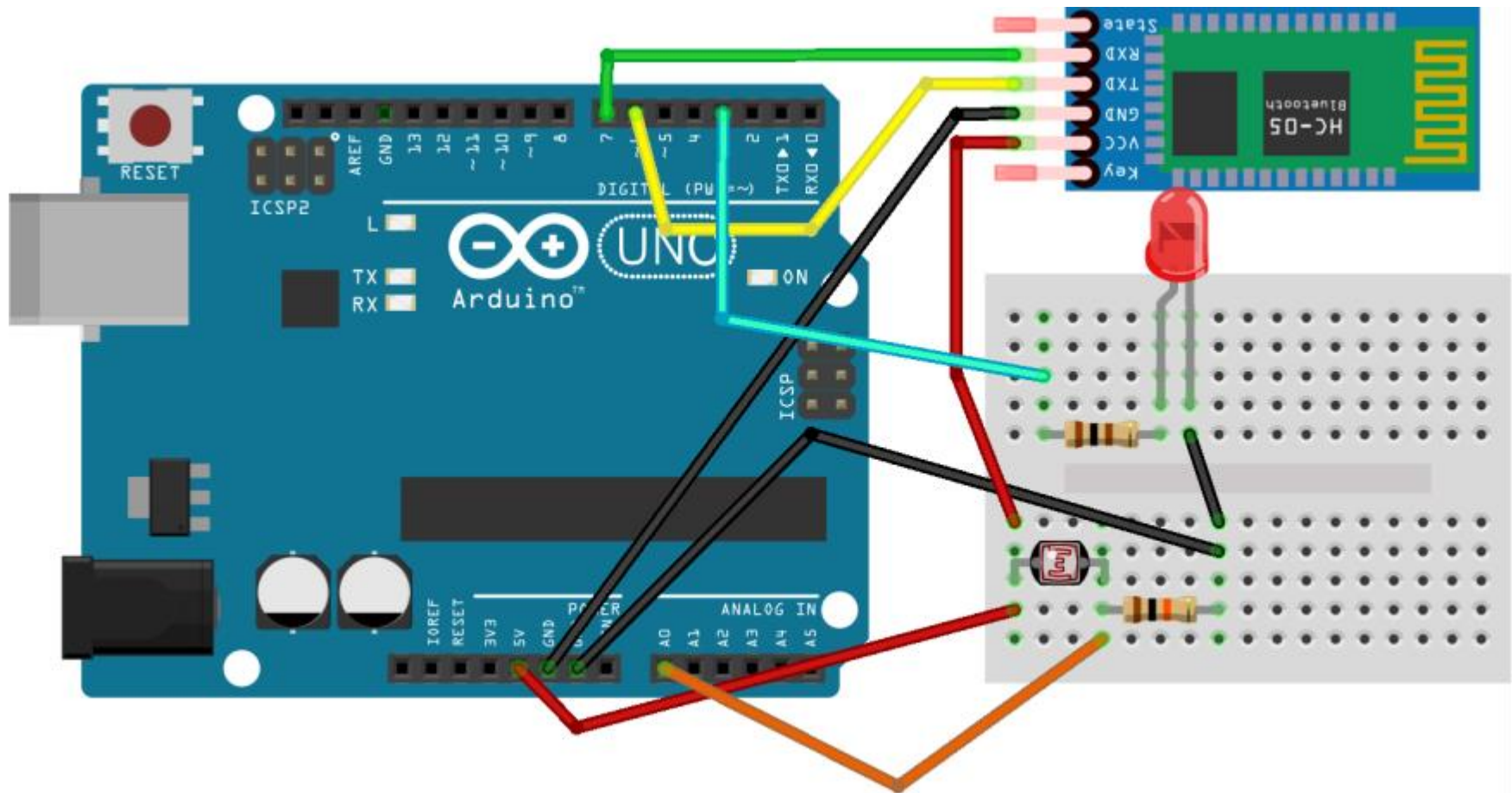
- Para configurar o módulo bluetooth, usamos o próprio monitor serial do Arduino, porém configuramos a velocidade para 38400 baud, com o envio de NL & CR
- Para testar a comunicação, digitamos o comando de teste “AT”, que deve ter “OK” como resposta.
- Alguns comandos que podem ser dados:
  - AT+VERSION (busca a versão do firmware)
  - AT+NAME=“Nome” (muda o nome do dispositivo para Nome)
  - AT+PSWD=qwerty123 (muda a senha de pareamento para qwerty123)
  - AT+STATE (verifica o estado do dispositivo)
- Uma lista de comandos pode ser encontrada em [https://s3-sa-east-1.amazonaws.com/robocore-lojavirtual/709/HC-05\\_ATCommandSet.pdf](https://s3-sa-east-1.amazonaws.com/robocore-lojavirtual/709/HC-05_ATCommandSet.pdf)



## Usando a biblioteca Software Serial

- Após a configuração do módulo HC-05, provavelmente desejamos realizar uma série de modificações no programa **enquanto** usamos o módulo Bluetooth
- Como já vimos, **não** é possível **programar** o Arduino **enquanto** o HC-05 estiver conectado em Rx/Tx (portas 0/1)
- Para evitar o inconveniente de remover e reconectar o HC-05 toda hora, podemos conectá-lo via biblioteca `SoftwareSerial`
- Para utilizá-la, basta instanciar a classe `SoftwareSerial`, passando as portas Rx e TX, respectivamente
  - Manual: <https://www.arduino.cc/en/Reference/SoftwareSerial>

## Montagem



```
#include <ArduinoJson.h>
#include <SoftwareSerial.h>
const int LED = 3;
const int LUZ = A1;
const int TAMANHO = 200;
SoftwareSerial mySerial(6, 7); //Rx, Tx
void setup() {
    mySerial.begin(9600);
    mySerial.setTimeout(10); //1000ms é muito tempo
    pinMode(LED, OUTPUT);
}
void loop() {
    if (mySerial.available() > 0) {
        //Lê o texto disponível na porta serial:
        char texto[TAMANHO];
        mySerial.readBytesUntil('\n', texto, TAMANHO);
        //Grava o texto recebido como JSON
        StaticJsonBuffer<TAMANHO> jsonBuffer;
        JsonObject& json = jsonBuffer.parseObject(texto);
        if(json.success() && json.containsKey("led")) {
            analogWrite(LED, json["led"]);
        }
    }
    StaticJsonBuffer<TAMANHO> jsonBuffer;
    JsonObject& json = jsonBuffer.createObject();
    json["luz"] = analogRead(LUZ);
    json.printTo(mySerial); mySerial.println();
    delay(1000);
}
```

# REFERÊNCIAS



- <https://www.robocore.net/tutoriais/configurando-bluetooth-hc-05-via-arduino.html>
- <https://www.arduino.cc/en/Reference/SoftwareSerial>



Copyright © 2019 Prof. Antonio Selvatici

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).