

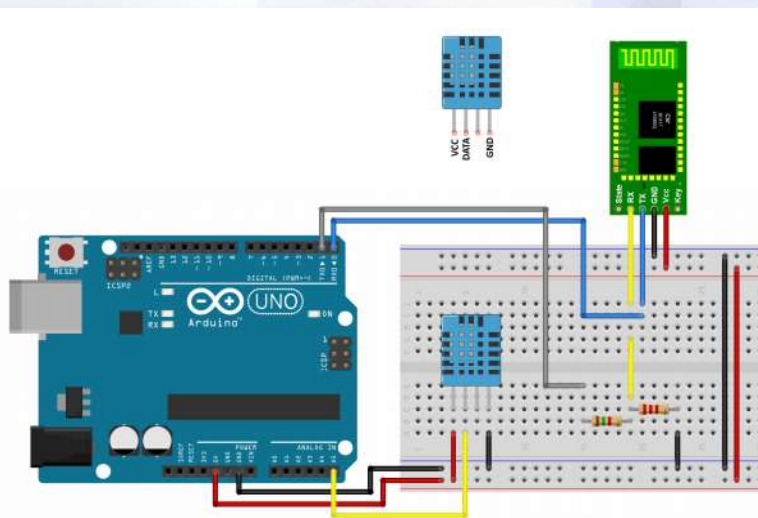
## ***Problem Based Learnd - PBL 02***

- **Problema:** Apresentar os valores de umidade e temperatura no Aplicativo Android, a cada 3 segundos, utilizando-se do Sensor DHT11
  - Para isto, deve-se enviar essas informações pelo Módulo Bluetooth HC-05 para o Aplicativo Bluetooth SPP

## **Componentes Utilizados**

- Arduino UNO / Arduino MEGA
- Cabo USB
- Protoboard
- Sensor DHT11
- Módulo Bluetooth HC-05
- 2 Resistores de 330 ohms
- Fonte de energia externa para alimentação do Arduino (Recomendado : 9 à 12 Vcc)
- Cabos de Jumpers Macho

## Construindo o Circuito no Protoboard



## Código Fonte para Solução (1)

```

1 #include "DHT.h"
2
3 #define DHTPIN A1 // pino que estamos conectado
4 #define DHTTYPE DHT11 // DHT 11
5
6 // Conecte pino 1 do sensor (esquerda) ao +5V
7 // Conecte pino 2 do sensor ao pino de dados definido em seu Arduino
8 // Conecte pino 4 do sensor ao GND
9 // Conecte o resistor de 10K entre pin 2 (dados)
10 // e ao pino 1 (VCC) do sensor
11 DHT dht(DHTPIN, DHTTYPE);
12
13 void setup()
14 {
15     Serial.begin(9600);
16     delay(1000);
17     Serial.println("DHT - OK");
18     dht.begin();
19 }

```

## Código Fonte para Solução (2)

```

20
21 void loop()
22 {
23   // A leitura da temperatura e umidade pode levar 250ms!
24   // O atraso do sensor pode chegar a 2 segundos.
25   float h = dht.readHumidity();
26   float t = dht.readTemperature();
27
28   // testa se retorno é valido, caso contrário algo está errado.
29   if (isnan(t) || isnan(h))
30   {
31     Serial.println("Failed to read from DHT");
32   }
33   else
34   {
35     Serial.print("Umidade: ");
36     Serial.print(h);
37     Serial.print(" ");
38     Serial.print("Temperatura: ");
39     Serial.print(t);
40     Serial.println(" C");
41   }
42
43   delay(3000);
44 }

```

## Verificação e Simulação do Código Desenvolvido (1)

- Embarcar o código-fonte na aplicação
  - Não esquecer de deixar o Módulo Bluetooth desconectado do VCC
- Execute o aplicativo Bluetooth SPP
  1. Será apresentada a tela principal
  2. Selecione o botão MENU do celular e escolha a opção Connected e depois Research
  3. O celular inicia uma varredura para detectar os dispositivos bluetooth, e o módulo Bluetooth que você pareou anteriormente será exibido na tela



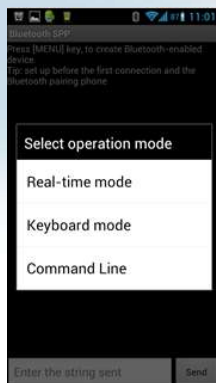
1

2

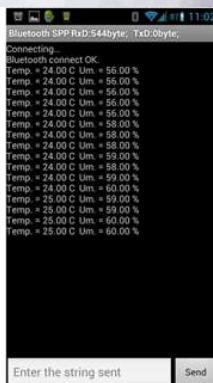
3

## Verificação e Simulação do Código Desenvolvido (2)

4. Clique no dispositivo detectado, e na tela de seleção de modo de operação, selecione REAL TIME MODE (Modo de tempo real)
5. Será mostrada, então, uma nova tela, com as informações de temperatura atualizadas a cada 3 segundos:



4



5

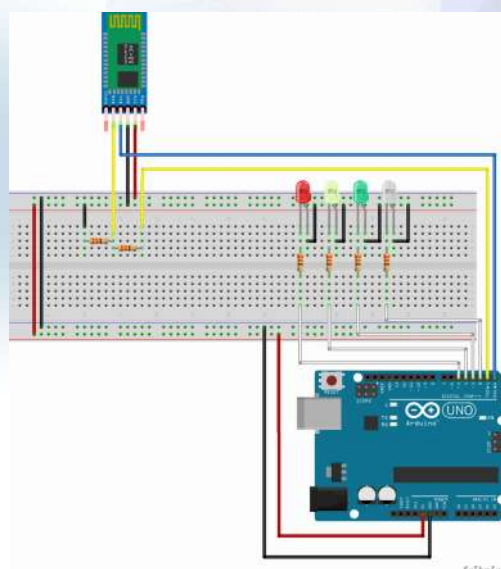
## Problem Based Learnd - PBL 03

- **Problema:** Acender e Apagar LEDs de cores diferentes utilizando como entrada informações do Usuário enviadas por Bluetooth pelo Dispositivo Móvel através do Aplicativo Bluetooth SPP

## Componentes Utilizados

- Arduino UNO / Arduino MEGA
- Cabo USB
- Protoboard
- Módulo Bluetooth HC-05
  - 2 Resistores de 330 ohms
- Fonte de energia externa 9 à 12V para Alimentação do Arduino
- 4 LEDs (vermelho, verde, amarelo e branco)
  - 4 Resistores 330 ohms
- Cabos Jumpers Macho

## Construindo o Circuito no Protoboard



## Código Fonte para Solução (1)

```

1 // The LEDs are connected to these pins
2 int LEDRedPin = 5;
3 int LEDYellowPin = 4;
4 int LEDGreenPin = 3;
5 int LEDWhitePin = 2;
6
7 void setup()
8 {
9     // set up serial at 9600 baud
10    Serial.begin(9600);
11
12    // set all four LED pins to output mode
13    pinMode(LEDRedPin, OUTPUT);
14    pinMode(LEDYellowPin, OUTPUT);
15    pinMode(LEDGreenPin, OUTPUT);
16    pinMode(LEDWhitePin, OUTPUT);
17 }
18
19 void toggleLED(int LEDPin)
20 {
21     // toggle the LED on the pin passed as an argument
22     digitalWrite(LEDPin, !digitalRead(LEDPin));
23 }

```

## Código Fonte para Solução (2)

```

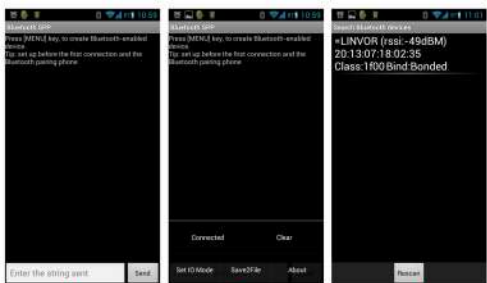
25 void loop()
26 {
27     if (Serial.available() > 0)
28     {
29         // read a single character over serial
30         int inByte = Serial.read();
31
32         switch (inByte)
33         {
34             // if we receive r, y, g, or w toggle the respective LED using our function
35             case 'r':
36                 toggleLED(LEDRedPin);
37                 break;
38             case 'y':
39                 toggleLED(LEDYellowPin);
40                 break;
41             case 'g':
42                 toggleLED(LEDGreenPin);
43                 break;
44             case 'w':
45                 toggleLED(LEDWhitePin);
46                 break;
47             default:
48                 // if we receive any other character turn all the LEDs off
49                 digitalWrite(LEDRedPin, LOW);
50                 digitalWrite(LEDYellowPin, LOW);
51                 digitalWrite(LEDGreenPin, LOW);
52                 digitalWrite(LEDWhitePin, LOW);
53                 break;
54         }
55     }
56 }

```

Instituto Federal de São Paulo – IFSP São João da Boa Vista

## Verificação e Simulação do Código Desenvolvido (1)

- Embarcar o código-fonte na aplicação
  - Não esquecer de deixar o Módulo Bluetooth desconectado do VCC
- Execute o aplicativo Bluetooth SPP
  1. Será apresentada a tela principal
  2. Selecione o botão MENU do celular e escolha a opção Connected
  3. O celular inicia uma varredura para detectar os dispositivos bluetooth, e o módulo Bluetooth que você parou anteriormente será exibido na tela

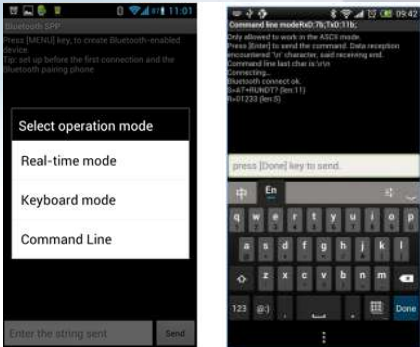


1                      2                      3

Instituto Federal de São Paulo – IFSP São João da Boa Vista

## Verificação e Simulação do Código Desenvolvido (2)

4. Clique no dispositivo detectado, e na tela de seleção de modo de operação, selecione REAL TIME MODE (Modo de tempo real)
5. Será mostrada, então, uma nova tela, que irá responder de acordo com as entradas do usuário:
  - Enviar um dos seguintes caracteres para acender um dos LEDs: r – Vermelho / y – Amarelo / g – Verde / w – Branco



4                      5

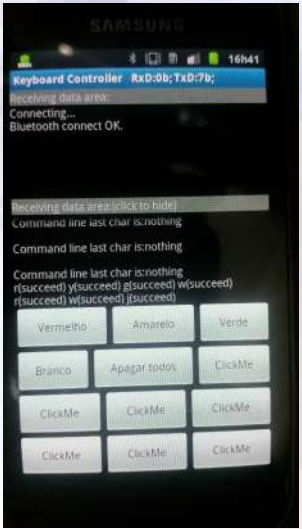


Instituto Federal de São Paulo – IFSP São João da Boa Vista

## Verificação e Simulação do Código Desenvolvido (3)

Apresentar o **Keyboard Mode** para Enviar Informações ao Arduino através de Botões:

- Clicar em configurar e em “**Set Key Func**”
- Clicar nos botões desejados e alterar as informações para:
  - Botão 01:
    - Button Name: Vermelho
    - Send value: r
  - Botão 02:
    - Button Name: Amarelo
    - Send value: y
  - Botão 03:
    - Button Name: Verde
    - Send value: g
  - Botão 04:
    - Button Name: Branco
    - Send value: w
  - Botão 05:
    - Button Name: Apagar todos
    - Send value: q
- Clicar em configurar e “**Complete set**”
- Interagir com o Arduino acionando os Botões



Instituto Federal de São Paulo – IFSP São João da Boa Vista

## Miniprojeto 07

- Acender e Apagar LEDs de cores diferentes utilizando como entrada informações do Usuário enviadas por Bluetooth pelo Dispositivo Móvel, através do Aplicativo Bluetooth SPP
  - Ao acender ou apagar um LED, deve ser apresentada uma mensagem dizendo qual foi o LED que ACENDEU ou APAGOU
    - Exemplo:
      - LED VERMELHO ACENDEU / LED VERMELHO APAGOU
      - LED BRANCO ACENDEU / LED BRANCO APAGOU
      - LED VERDE ACENDEU / LED VERDE APAGOU
      - LED AMARELO ACENDEU / LED AMARELO APAGOU



## Miniprojeto 08

- Criar um circuito com apenas um LED Branco e sua luminosidade pode ser alterada através da entrada do Usuário enviada por Bluetooth pelo Dispositivo Móvel, utilizando o Aplicativo Bluetooth SPP
  - Sugestão: Criar uma escala de quanto em quanto varia a luminosidade
  - Garantir que nenhum valor inválido seja entrado pelo usuário

## Miniprojeto 09

- Criar um circuito com dois LEDs Brancos e sua luminosidade pode ser alterada através da entrada do Usuário enviada por Bluetooth pelo Dispositivo Móvel, utilizando o Aplicativo Bluetooth SPP
  - Sugestões:
    - Criar uma regra de quando um dos LEDs é acionado
    - Criar uma escala de quanto em quanto varia a luminosidade
    - Garantir que nenhum valor inválido seja entrado pelo usuário

## Miniprojeto 10

- Simular o funcionamento das Setas (02 LED Vermelhos - 1 Seta Esquerda e 1 Seta Direita), Lanterna Dianteira (02 LEDs Amarelos) e Luz de ré (02 LEDs Brancos) de um protótipo de carro controlado remotamente pelo Usuário, com informações enviadas por Bluetooth pelo Dispositivo Móvel, através do Aplicativo Bluetooth SPP.
  - Ao acionar esquerda, o LED Vermelho da Esquerda deve ficar piscando. Só parar quando acionar novamente
  - Ao acionar direita, o LED Vermelho da Direita deve ficar piscando. Só parar quando acionar novamente
  - Ao acionar Lanterna Dianteira, os LEDs Amarelos devem acender. Só parar quando acionar novamente
  - Ao acionar Farol de Ré, os LEDs Brancos devem acender. Só parar quando acionar novamente

## PDM: Projeto para Dispositivos Móveis

### Aula 07a: Utilização do Módulo de Bluetooth HC-05

Breno Lisi Romano

**Obrigado!**

Instituto Federal de São Paulo – IFSP São João da Boa Vista  
Especialização em Desenvolvimento para Dispositivos Móveis



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
SÃO PAULO  
Campus São João da Boa Vista