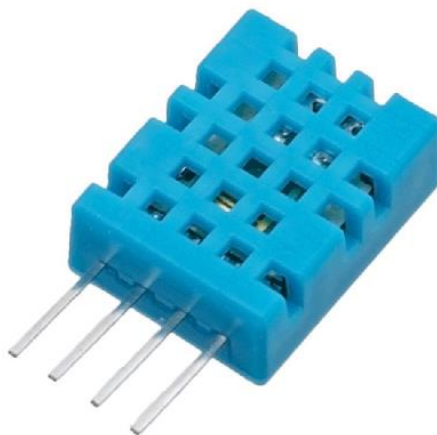




Universidade Federal de Sergipe - UFS
Departamento de Sistemas de Informação - Itabaiana - DSI/Ita
Sistema de Tempo Real
Docente: André Luis Meneses Silva
Discentes: Abraão Alves Pereira
Bruno Santana Santos
Vanessa Lima Santos

Tutoriais usados nas implementações dos sensores de temperatura e umidade, luz e presença e módulo Bluetooth

Como usar com Arduino – Sensor de Umidade e Temperatura DHT11



– Descrição:

O **Sensor de Umidade e Temperatura – DHT11** é um dos componentes mais utilizados em projetos que envolva medição de temperatura e umidade ambiente. Este sensor faz medições de temperatura de 0° até 50° celsius e mede a umidade do ar nas faixas de 20% a 90%. A precisão (margem de erro) do sensor para medição de temperatura é de aproximadamente 2° celsius e para umidade é de 5%.

– Especificações e características:

- Tensão de operação: 3,5 a 5,5VDC
- Corrente de operação: 0,3mA
- Corrente de operação (em stand by): 60μA
- Resolução: 16 bits
- Faixa de medição (umidade): 20 a 90%
- Faixa de medição (temperatura): 0° a 50° celsius
- Precisão (umidade): ±5%

- Precisão (temperatura): $\pm 2^{\circ}$ celsius
- Tempo de resposta: 2s

– **Datasheet:**

Datasheet DHT

– **Aplicações:**

Projetos com finalidade de medir umidade e temperatura ambiente utilizando Arduino ou outras plataformas microcontroladas.

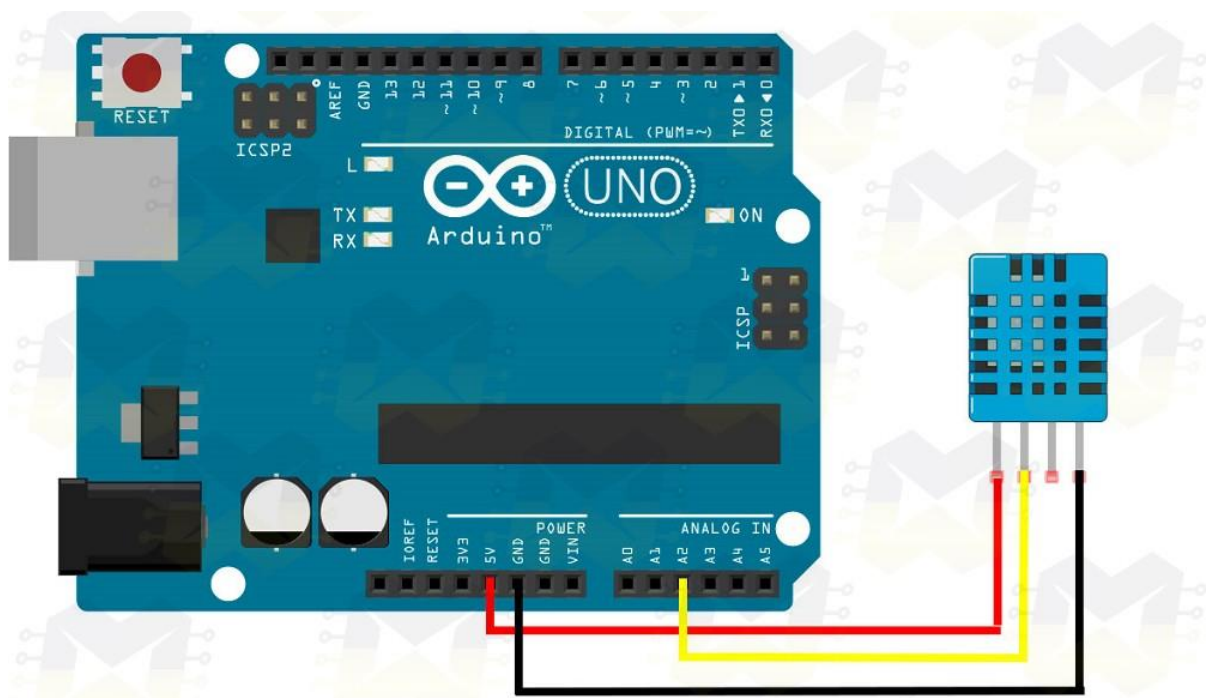
– **Proposta da prática:**

Utilizar o Sensor de Umidade e Temperatura DHT11 em conjunto com o Arduino e medir umidade e temperatura ambiente. O valor da umidade e da temperatura serão exibidos no monitor serial do ambiente de programação do Arduino.

– **Lista dos itens necessários:**

01. Arduino com Cabo USB
02. Sensor de Umidade e Temperatura – DHT11
03. Cabos Jumper macho-fêmea

– **Esquema de ligação da prática:**



– **Biblioteca:**

Biblioteca DHT11

– Código:

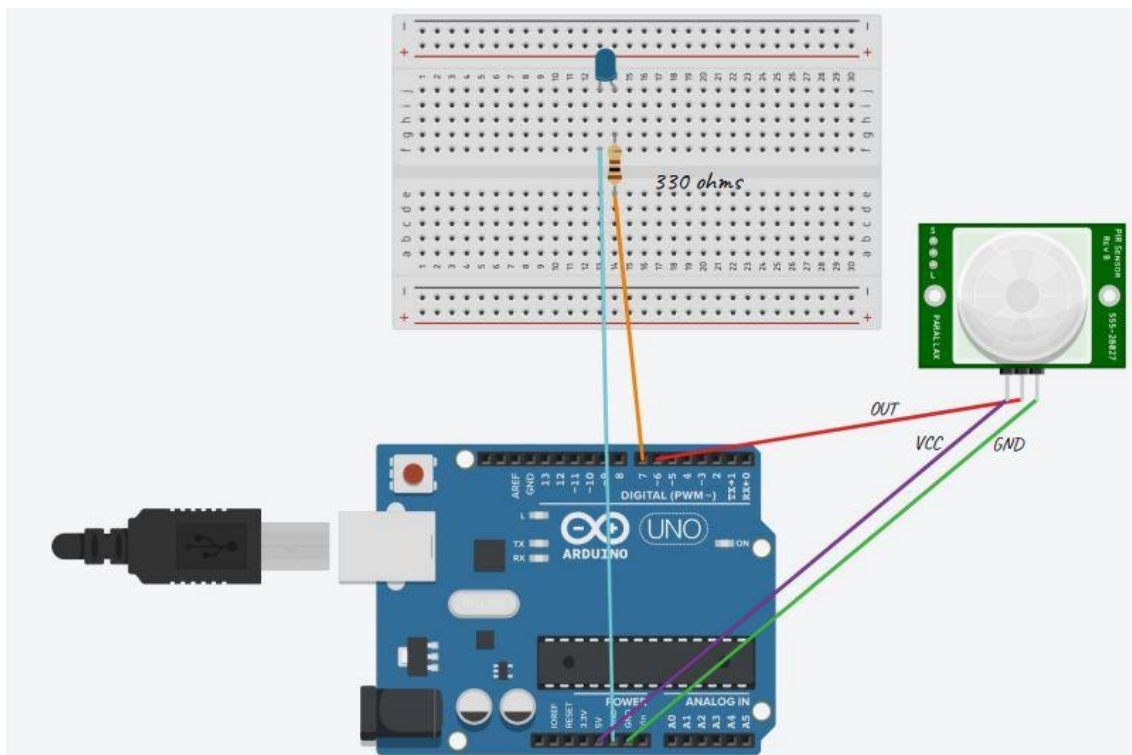
```
1 #include "dht.h" //INCLUSÃO DE BIBLIOTECA
2
3 const int pinoDHT11 = A2; //PINO ANALÓGICO UTILIZADO PELO DHT11
4
5 dht DHT; //VARIÁVEL DO TIPO DHT
6
7 void setup(){
8     Serial.begin(9600); //INICIALIZA A SERIAL
9     delay(2000); //INTERVALO DE 2 SEGUNDO ANTES DE INICIAR
10 }
11
12 void loop(){
13     DHT.read11(pinoDHT11); //LÊ AS INFORMAÇÕES DO SENSOR
14     Serial.print("Umidade: "); //IMPRIME O TEXTO NA SERIAL
15     Serial.print(DHT.humidity); //IMPRIME NA SERIAL O VALOR DE UMIDADE MEDIDO
16     Serial.print("%"); //ESCREVE O TEXTO EM SEGUIDA
17     Serial.print(" / Temperatura: "); //IMPRIME O TEXTO NA SERIAL
18     Serial.print(DHT.temperature, 0); //IMPRIME NA SERIAL O VALOR DE UMIDADE MEDIDA
19     Serial.println("°C"); //IMPRIME O TEXTO NA SERIAL
20     delay(2000); //INTERVALO DE 2 SEGUNDOS * NÃO DIMINUIR ESSE VALOR
21 }
```

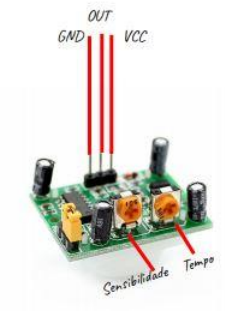
Trabalhando com o sensor de presença

– Descrição:

- ✚ Sensor de Movimento PIR DYP-ME003 detecta o movimento de objetos que estejam em uma área de até 7 metros
- ✚ Caso algo ou alguém se movimente nesta área o pino de alarme é ativado.
- ✚ Características:
 - ✚ Tensão de Operação: 4,5-20V
 - ✚ Tensão Dados: 3,3V (Alto) – 0V (Baixo)
 - ✚ Distância detectável: 3-7m (Ajustável)
 - ✚ Tempo de Delay: 5-200seg (Default: 5seg)
 - ✚ Tempo de Bloqueio: 2,5seg (Default)
 - ✚ Trigger: (L)-Não Repetível (H)-Repetível (Default: H)

– **Esquema de ligação da prática:**





– Código:

```
void setup() {  
  pinMode(6, INPUT);  
  pinMode(7, OUTPUT);  
}  
  
void loop() {  
  if (digitalRead(6) == 1)  
    digitalWrite(7, HIGH);  
  else  
    digitalWrite(7, LOW);  
}
```

Trabalhando com o Sensor de Luz LDR 5mm

– Descrição:

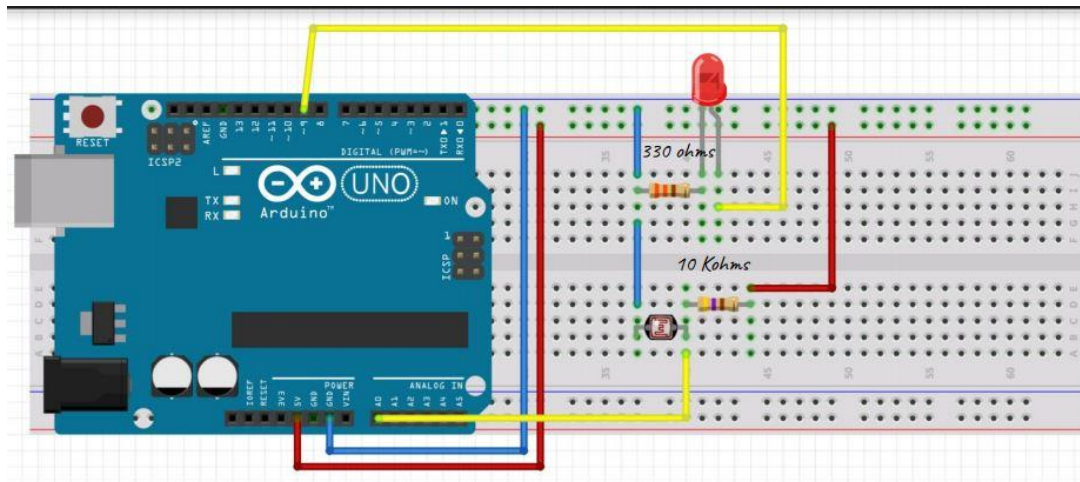
O Sensor de Luminosidade LDR (Light Dependent Resistor) é um componente cuja resistência varia de acordo com a intensidade da luz. Quanto mais luz incidir sobre o componente, menor a resistência.

– Características:

- Tensão máxima: 150VDC
- Resistência no escuro: 1 M Ω (Lux 0)
- Resistência na luz: 10-20 K Ω (Lux 10)



– Esquema de ligação da prática:



– Código:

*// Programa : LDR - Sensor de Iluminação
// Autor : Arduino e Cia*

```
int portaLed = 9; //Porta a ser utilizada para ligar o led
int portaLDR = A0; //Porta analógica utilizada pelo LDR

void setup()
{
  pinMode(portaLed, OUTPUT); //Define a porta do Led como saída
}

void loop()
{
  int estado = analogRead(portaLDR); //Lê o valor fornecido pelo LDR
  // Caso o valor lido na porta analógica seja maior do que
  // 800, acende o LED
  // Ajuste o valor abaixo de acordo com o seu circuito
  if (estado > 500)
  {
    digitalWrite(portaLed, HIGH);
  }
  else //Caso contrário, apaga o led
  {
    digitalWrite(portaLed, LOW);
  }
}
```

Controlling LED with BLE Module HM 10

This tutorial explains how to control LED with Android app using BLE Module HM10.

1. Introduction:

In this tutorial we explain, how to send and receive data from Arduino to the Android app without using Serial monitor.

For this, we have taken a simple example of LED. We will send command to get LED on and off.

1.1 Default Setting of BT Module:

The default setting of HM10 are:

Name- MLT-BT05

Password- 123456789

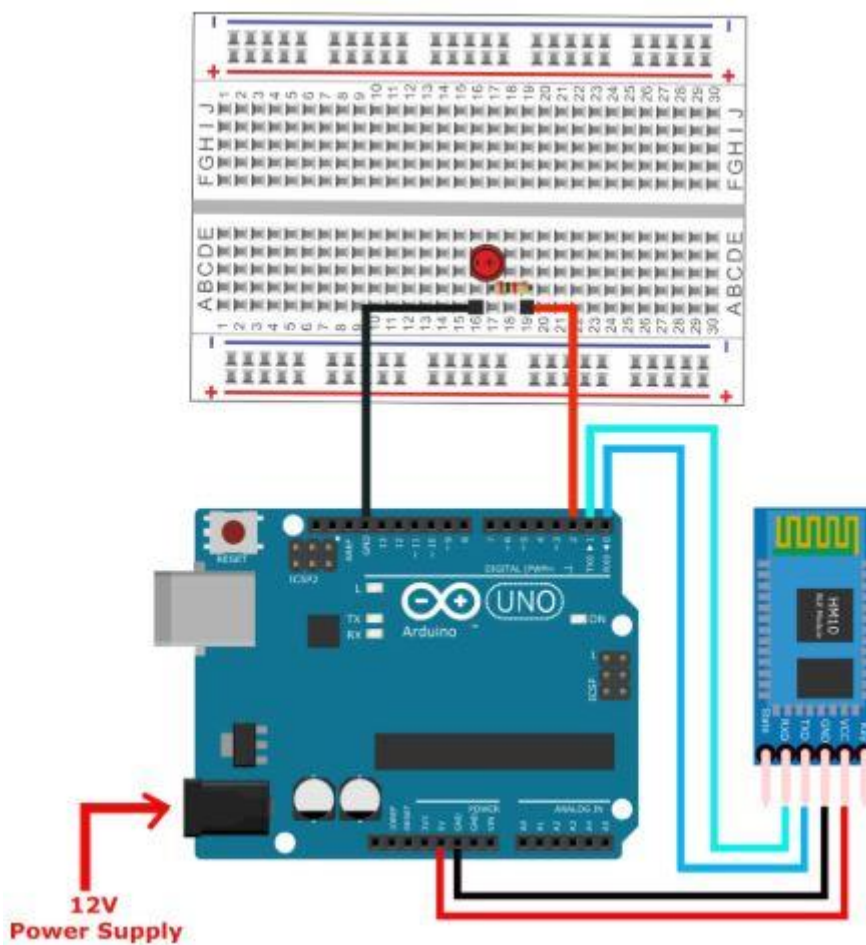
Baud Rate- 9600

All the modules have the same baud rate by default. Make sure to set Baud Rate to 9600 in your code. If you want to change the baud rate in your code, then you also need to change the baud rate of device by using AT Commands.

2. Hardware required

S.No.	Item	Quantity
1.	Arduino	1
2.	Breadboard	1
3.	BT Module HM10	1
4.	LED	1
5.	Resistor 1K	1
6.	Male to Female Jumper Wires	4
7.	Male to Male Jumper Wires	2

3. Building Circuit



4. Programming:

Once the circuit part is done, Arduino is needed to be programmed.

You may download this code (Arduino Sketch) from here.

```
//Robo India tutorial on Controlling LED with Android
//Hardware Required: BLE HM10 & Arduino
//https://www.roboindia.com/tutorials/

byte LED = 2;           // device to control
char BT_input=' ';      // to store input character received via BT.

void setup()
{
    Serial.begin(9600);   // default baud rate of module
    pinMode(LED, OUTPUT); // device to control
    while (!Serial)
    {

    }
}

void loop()
{
    if (Serial.available())
    {
        BT_input = Serial.read();
        if ( BT_input== 48)      //ascii code for 0 is dec 48
        {
            digitalWrite(LED, LOW);
            Serial.println(BT_input);
            Serial.println("LED is OFF");
        }
        if ( BT_input== 49)
        {
            digitalWrite(LED, HIGH);
            Serial.println(BT_input);
            Serial.println("LED is ON"); //ascii code for 1 is dec 49
        }
    }
}
```

Before uploading the code, remove the RX and TX wires from Arduino's TX and RX pins. After uploading the code, connect them again and plug out the USB cable and supply external power using adaptor.

5. Mobile app

You may download the Android control App on Playstore from here.

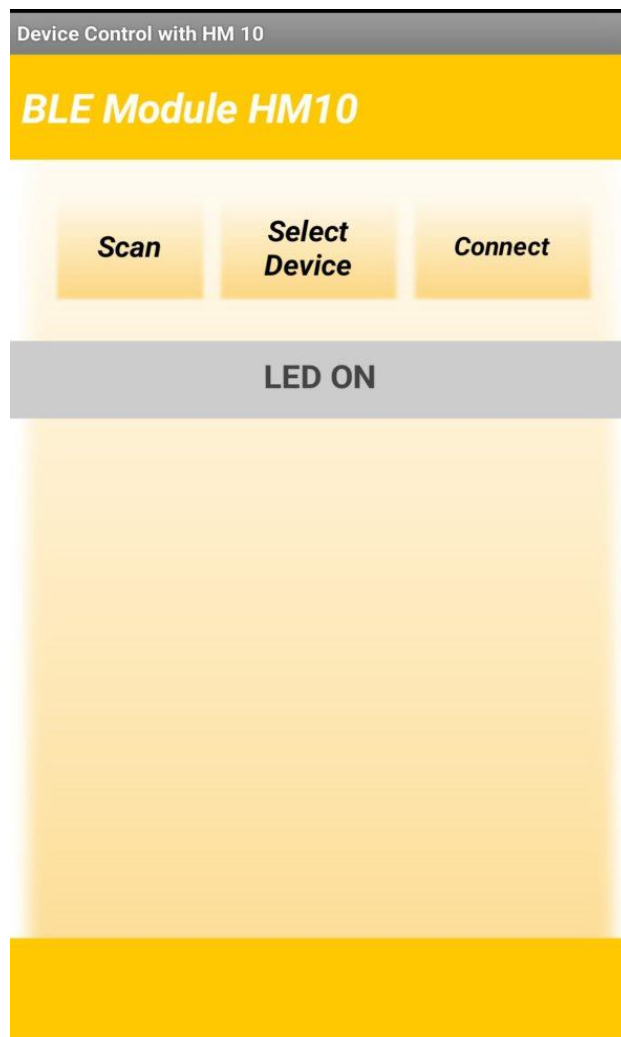
Connecting the Android device to the HM10 creates a serial communication channel very similar to the serial monitor in the Arduino IDE. This means we need a Bluetooth version of the serial monitor.

1. Download this Application.
2. Pair your phone with HM10. for doing this go to *Settings->Bluetooth->Scan device->select MLT-BT05* and pair it. Pass code to pair is '123456789'.
3. Now, open the app and connect the HM10 module.

After Connection, the blinking LED on the MODULE will remain ON.

6. Output:

Press on the button "LED ON" to turn on the connected LED.



If you have any query please write us at support@roboindia.com

Referências:

1. <http://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-sensor-de-umidade-e-temperatura-dht11/>
2. <https://roboindia.com/tutorials/ble-4-0-arduino-led-control/>
3. Slides de aula do professor André Luis