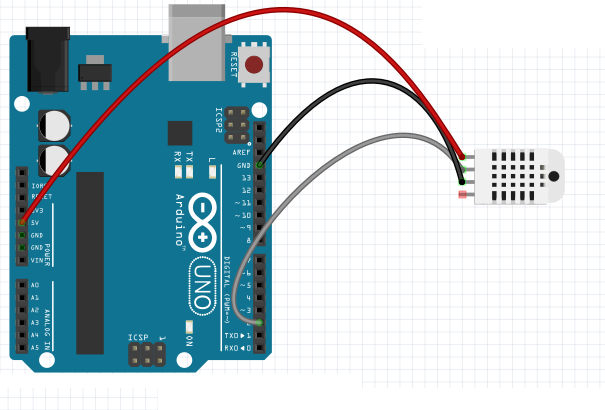
**Arduino – Modulo Ky – 015**

**DHT11 – Sensor de Temperatura e Humidade**

Este é o modulo de medição de temperatura e humidade, para isso não iremos usar a biblioteca padrão do Arduino, usaremos uma biblioteca com o nome do próprio modulo (DTH11) onde não há nenhum delay na execução do código, não afetando assim o funcionamento de outros possíveis dispositivos conectados à placa.

Como a biblioteca é especifica para o modulo, há diversas funções prontas para serem usadas no código ampliando a variedade de desenvolvimento, podendo apresentar temperatura em diversas medidas, por exemplo.

Foi usado no código os comandos, if, para verificar estados do sensor, e switch case, para a verificação e apresentação de erros, além do uso dos comandos direcionados para o monitor serial, onde tudo será exibido.

**OBS.** No Arduino Uno, apenas as portas 2 e 3 digitais devem ser usadas, pois são as únicas que possuem o recurso de interrupção necessário para o uso da biblioteca.

**Sketch**

**#include <idDHT11.h>**

int idDHT11pin = 2; //Porta Digital do Arduino onde o Sinal do Sensor DHT esta conectado

int idDHT11intNumber = 0; //Número da interrupção respectiva à porta definida no parametro anterior (veja tabela acima)

void dht11\_wrapper(); // Declaração da funcão de controle da interrupção.

void loopDHT(); // Atualiza a leitura do sensor

idDHT11 DHT11(idDHT11pin, idDHT11intNumber, dht11\_wrapper); //Instanciação do Objeto de Controle do Sensor

void setup()

{

Serial.begin(9600);

Serial.println("Inicio do Sketch");

}

//Variaveis que irao conter os valores lidos no Sensor DHT11

float temperaturaC;

float temperaturaF;

float temperaturaK;

float umidade;

float dewPoint;

float dewPointSlow;

void loop()

{

loopDHT();

Serial.print("Temperatura Celcius: ");

Serial.println( temperaturaC );

Serial.print("Umidade Relativa: ");

Serial.println( umidade );

Serial.print("Ponto de Orvalho: ");

Serial.println( dewPoint );

Serial.println();

}

void dht11\_wrapper() {

DHT11.isrCallback();

}

void loopDHT() {

#define tempoLeitura 1000

static unsigned long delayLeitura = millis() + tempoLeitura + 1;

static bool request = false;

if ((millis() - delayLeitura) > tempoLeitura) {

if (!request) {

DHT11.acquire();

request = true;

}

}

if (request && !DHT11.acquiring()) {

request = false;

int result = DHT11.getStatus();

switch (result)

{

case IDDHTLIB\_OK:

Serial.println("Leitura OK");

break;

case IDDHTLIB\_ERROR\_CHECKSUM:

Serial.println("Erro\n\r\tErro Checksum");

break;

case IDDHTLIB\_ERROR\_ISR\_TIMEOUT:

Serial.println("Erro\n\r\tISR Time out");

break;

case IDDHTLIB\_ERROR\_RESPONSE\_TIMEOUT:

Serial.println("Erro\n\r\tResponse time out");

break;

case IDDHTLIB\_ERROR\_DATA\_TIMEOUT:

Serial.println("Erro\n\r\tData time out erro");

break;

case IDDHTLIB\_ERROR\_ACQUIRING:

Serial.println("Erro\n\r\tAcquiring");

break;

case IDDHTLIB\_ERROR\_DELTA:

Serial.println("Erro\n\r\tDelta time to small");

break;

case IDDHTLIB\_ERROR\_NOTSTARTED:

Serial.println("Erro\n\r\tNao iniciado");

break;

default:

Serial.println("Erro Desconhecido");

break;

}

float valor = DHT11.getCelsius();

if (!isnan(valor)) {

temperaturaC = valor;

}

valor = DHT11.getHumidity();

if (!isnan(valor)) {

umidade = valor;

}

valor = DHT11.getFahrenheit();

if (!isnan(valor)) {

temperaturaF = valor;

}

valor = DHT11.getKelvin();

if (!isnan(valor)) {

temperaturaK = valor;

}

valor = DHT11.getDewPoint();

if (!isnan(valor)) {

dewPoint = valor;

}

valor = DHT11.getDewPointSlow();

if (!isnan(valor)) {

dewPointSlow = valor;

}

delayLeitura = millis();

}

}