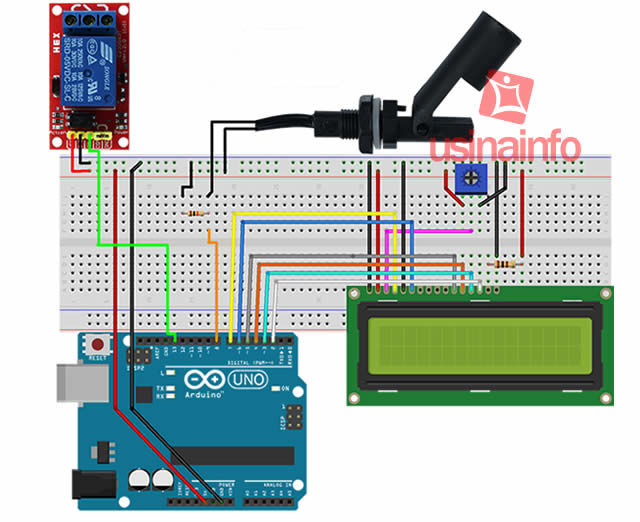
Sensor de Nível : <https://www.usinainfo.com.br/blog/sensor-de-nivel-de-agua-arduino-para-automacao-com-rele-e-display/>



1 Montagem no Arduino (apenas olhar para o sensor) (o Resistor é 10KΩ)

Usa: GND, VCC, RESISTOR E PORTA COMUM (NUMERO)

Código:

int PinSensor = 8;

int VarSensor = 0;

void setup()

{}

void loop(){

VarSensor = digitalRead(PinSensor);

//se var sensor for 0, ainda não chegou no nível do sensor e se for 1 chegou

}

Sensor de Gases Inflamáveis MQ-2: <https://www.usinainfo.com.br/blog/sensor-de-gas-arduino-mq-2-para-gases-inflamaveis-e-fumaca/>

Este sensor é um sensor desenvolvido com fins didáticos e para quem necessita de indicações de presença ou não de um dos gases listados, devido ao fato de não conseguir isolar cada gás que detecta, os seus valores não são de total precisão.

Usa: GND, VCC e V0

Código:

int pinSensor = A0; //Pino Sensor

int ValDesarm = 30; //Variável para selecionar a quantidade de Gás/Fumaça detectada

int valor = 0;

void setup()

{}

void loop()

{

valor = analogRead(pinSensor); //Faz a leitura da entrada do sensor

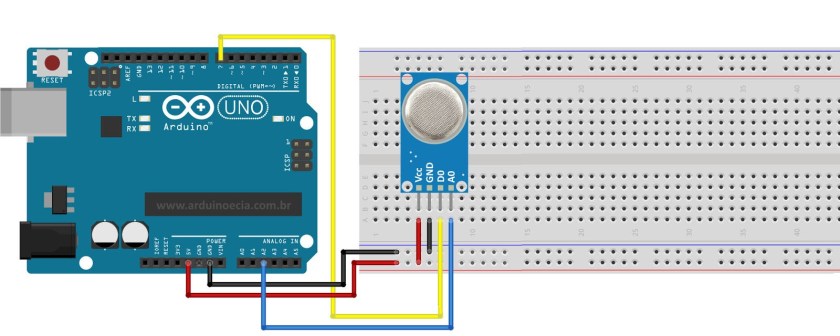
valor = map(valor, 0, 1023, 0, 100); //Faz a conversão da variável para porcentagem

Serial.println(valor); //Escreve o valor na porta Serial

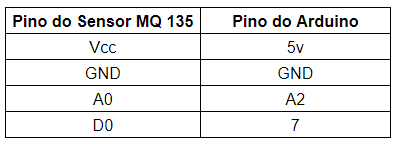
if (valor>=ValDesarm){ //Condição, se valor continuar maior que ValDesarm faça:

}

Sensor de Gases Inflamáveis MQ-135: <https://portal.vidadesilicio.com.br/sensor-de-gas-mq-135/>



O modulo MQ 135 possui 4 pinos: dois pinos para alimentação (Vcc e GND) e dois pinos para emissão dos sinais elétricos, sendo um sinal digital (D0) e o outro um sinal analógico (A0). Para realizar a montagem, iremos conectar os pinos na seguinte ordem:



Código:

#define MQ\_analog A2

#define MQ\_dig 7

int valor\_analog;

int valor\_dig;

void setup() {

Serial.begin(9600);

// Configura o sensor

pinMode(MQ\_analog, INPUT);

pinMode(MQ\_dig, INPUT);

}

void loop() {

valor\_analog = analogRead(MQ\_analog);

valor\_dig = digitalRead(MQ\_dig); // lê o valor do sensor

Serial.print(valor\_analog);

Serial.print(" || ");

if(valor\_dig == 0)

Serial.println("GAS DETECTADO !!!");

else

Serial.println("GAS AUSENTE !!!");

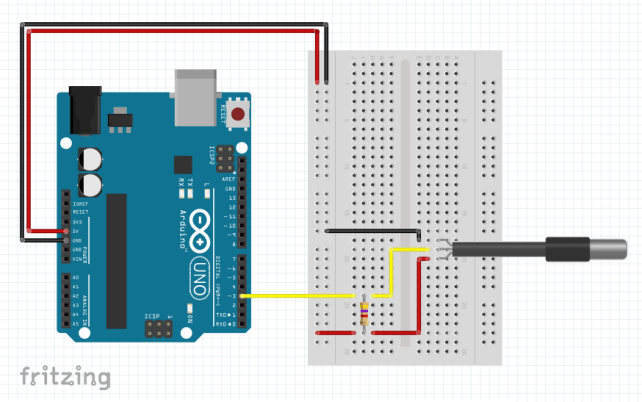
delay(500);

}

Sensor de Temperatura DS18B20: <https://portal.vidadesilicio.com.br/sensor-de-temperatura-ds18b20/>

o sensor DS18B20 possui sua própria inteligência. Ele é capaz de ler a temperatura, interpreta-la e enviar a informação do valor de temperatura em graus Celsius para o microcontrolador usando um barramento de apenar um fio (protocolo de comunicação One wire ou 1-wire ).

O sensor de temperatura DS18B20 pode medir temperaturas entre -55 ºC e 125 ºC com uma precisão de cerca de 0,5 ºC na faixa de -10 ºC e +85 ºC.



Código:

#include <OneWire.h>

#include <DallasTemperature.h>

OneWire pino(3); // transmissão de dados pro sensor

DallasTemperature barramento(&pino); // representa o barramento de

// sensores em nosso código

DeviceAddress sensor;

void setup(void)

{

Serial.begin(9600);

barramento.begin();

barramento.getAddress(sensor, 0);

}

void loop()

{

barramento.requestTemperatures();

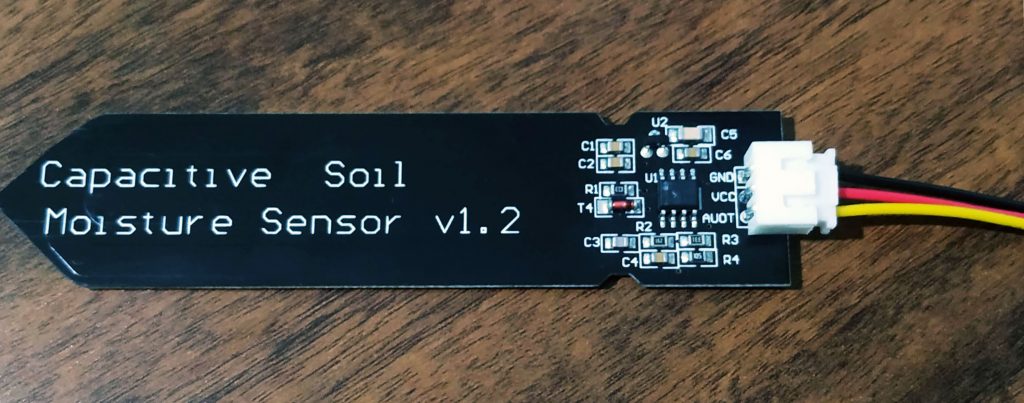
float temperatura = barramento.getTempC(sensor);

Serial.print(temperatura);

delay(500);

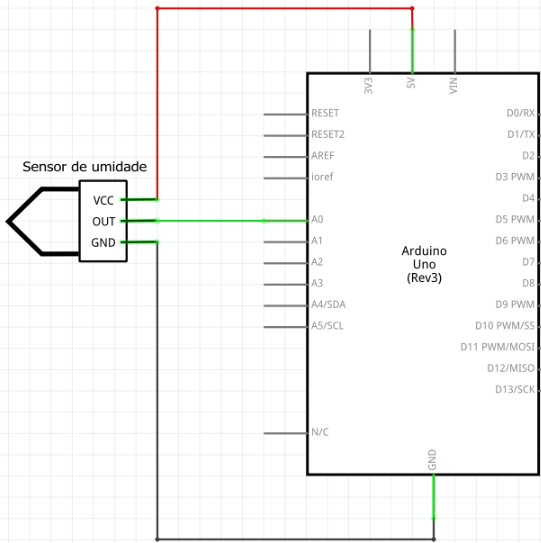
}

Sensor de Umidade Capacitivo Para Solo: <http://mundoprojetado.com.br/sensor-de-umidade-capacitivo-para-solo/>



O sensor de umidade capacitivo (imagem abaixo) é uma excelente solução para medir a umidade do solo. Isso porque ele evita o principal problema que existe em um sensor comum de umidade do solo: a corrosão.

Por outro lado, o sensor de umidade capacitivo é excelente, pois a medição da umidade não é feita em contato direto com nenhum elemento  metálico do sensor (abaixo explico como isso é feito). Logo, é possível aplicar uma camada externa anticorrosiva para proteger o sensor e fazer ele durar anos.

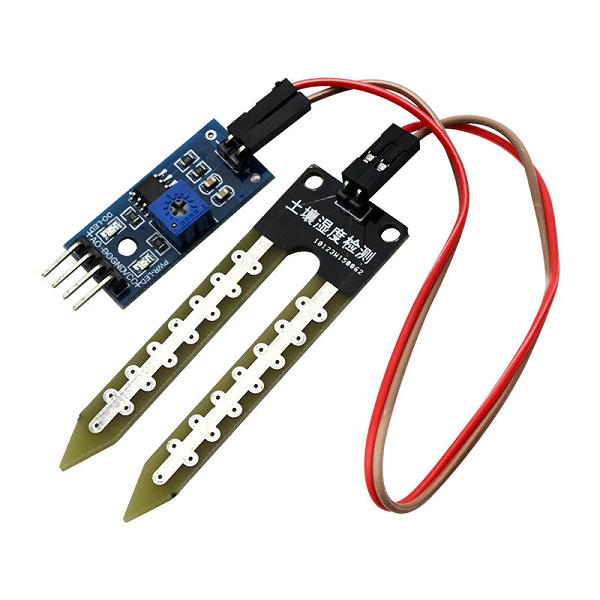


O sensor é muito fácil de utilizar, pois o circuito dele já converte a capacitância em tensão. Então, basta ler esse valor da tensão ligando o pino “OUT” ou “AVOT” em um pino analógico do Arduino (o A0 neste caso).

Em relação aos demais pinos: é só conectar o Vcc no 5v (ou 3.3v) e o GND no GND do Arduino.

void setup() {  
  // Inicia a comunicação serial  
  Serial.begin(9600);  
}  
  
void loop() {  
  // Faz a leitura do sensor  
  int leitura = analogRead(A0);  
  
  // Exibe o valor lido no monitor serial  
  Serial.println(leitura);  
  
  //Cria um pequeno atraso entre cada medição  
  delay(1000);  
}

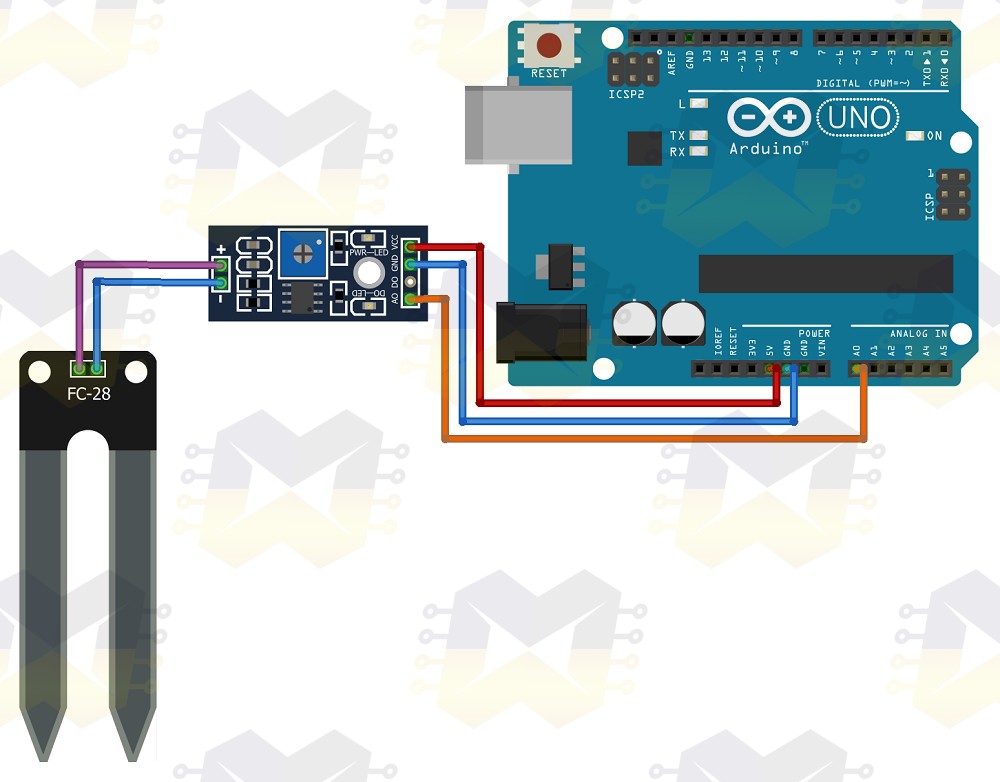
Sensor de Umidade do Solo (Higrômetro): <https://blogmasterwalkershop.com.br/arduino/como-usar-com-arduino-sensor-medidor-de-umidade-do-solo-higrometro/#:~:text=%E2%80%93%20Descri%C3%A7%C3%A3o%3A,se%20mant%C3%A9m%20em%20n%C3%ADvel%20baixo.>



Caso o solo esteja seco o sensor mantém a saída (digital) em nível alto e quando o solo estiver úmido a saída (digital) se mantém em nível baixo. Para medir a variação da umidade no solo, recomenda-se o uso do pino analógico disponível no sensor em conjunto com um microcontrolador que possua conversor analógico digital.

**– Lista dos itens necessários:**

01 – [Arduino com Cabo USB](https://www.masterwalkershop.com.br/arduino-uno-r3-com-cabo-usb-ab)  
01 – [Sensor (Medidor) de Umidade do Solo (Higrômetro)](https://www.masterwalkershop.com.br/sensor-medidor-de-umidade-do-solo-higrometro)  
03 – [Cabos Jumper macho-fêmea](https://www.masterwalkershop.com.br/cabo-jumper-macho-femea-20cm-kit-com-40pcs)  
02 – [Cabos Jumper fêmea-fêmea](https://www.masterwalkershop.com.br/cabo-jumper-femea-femea-20cm-kit-com-40pcs)  
01 – Vaso com terra seca



const int pinoSensor = A0; //PINO UTILIZADO PELO SENSOR

int valorLido; //VARIÁVEL QUE ARMAZENA O PERCENTUAL DE UMIDADE DO SOLO

int analogSoloSeco = 400; //VALOR MEDIDO COM O SOLO SECO (VOCÊ PODE FAZER TESTES E AJUSTAR ESTE VALOR)

int analogSoloMolhado = 150; //VALOR MEDIDO COM O SOLO MOLHADO (VOCÊ PODE FAZER TESTES E AJUSTAR ESTE VALOR)

int percSoloSeco = 0; //MENOR PERCENTUAL DO SOLO SECO (0% - NÃO ALTERAR)

int percSoloMolhado = 100; //MAIOR PERCENTUAL DO SOLO MOLHADO (100% - NÃO ALTERAR)

void setup(){

Serial.begin(9600); //INICIALIZA A SERIAL

Serial.println("Lendo a umidade do solo..."); //IMPRIME O TEXTO NO MONITOR SERIAL

delay(2000); //INTERVALO DE 2 SEGUNDOS

}

void loop(){

valorLido = constrain(analogRead(pinoSensor),analogSoloMolhado,analogSoloSeco); //MANTÉM valorLido DENTRO DO INTERVALO (ENTRE analogSoloMolhado E analogSoloSeco)

valorLido = map(valorLido,analogSoloMolhado,analogSoloSeco,percSoloMolhado,percSoloSeco); //EXECUTA A FUNÇÃO "map" DE ACORDO COM OS PARÂMETROS PASSADOS

Serial.print("Umidade do solo: "); //IMPRIME O TEXTO NO MONITOR SERIAL

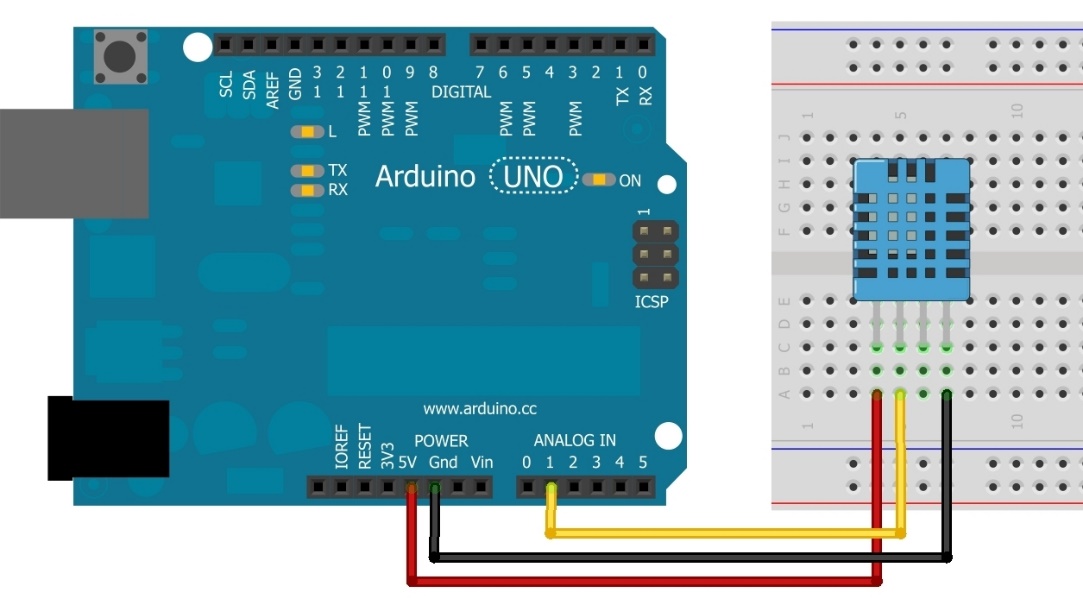
Serial.print(valorLido); //IMPRIME NO MONITOR SERIAL O PERCENTUAL DE UMIDADE DO SOLO

Serial.println("%"); //IMPRIME O CARACTERE NO MONITOR SERIAL

delay(1000); //INTERVALO DE 1 SEGUNDO

}

Temperatura e Umidade com o sensor DHT11: <https://www.filipeflop.com/blog/monitorando-temperatura-e-umidade-com-o-sensor-dht11/>



O DHT11 possui 4 terminais sendo que somente 3 são usados: GND, VCC e Dados. Se desejar, pode-se adicionar um resistor pull up de 10K entre o VCC e o pino de dados.

#include "DHT.h"

#define DHTPIN A1 // pino que estamos conectado

#define DHTTYPE DHT11 // DHT 11

// Conecte pino 1 do sensor (esquerda) ao +5V

// Conecte pino 2 do sensor ao pino de dados definido em seu Arduino

// Conecte pino 4 do sensor ao GND

// Conecte o resistor de 10K entre pin 2 (dados)

// e ao pino 1 (VCC) do sensor

DHT dht(DHTPIN, DHTTYPE);

void setup()

{

  Serial.begin(9600);

  Serial.println("DHTxx test!");

  dht.begin();

}

void loop()

{

  // A leitura da temperatura e umidade pode levar 250ms!

  // O atraso do sensor pode chegar a 2 segundos.

  float h = dht.readHumidity();

  float t = dht.readTemperature();

  // testa se retorno é valido, caso contrário algo está errado.

  if (isnan(t) || isnan(h))

  {

    Serial.println("Failed to read from DHT");

  }

  else

  {

    Serial.print("Umidade: ");

    Serial.print(h);

    Serial.print(" %t");

    Serial.print("Temperatura: ");

    Serial.print(t);

    Serial.println(" \*C");

  }

}