Universidade de São Paulo
Brasil

Universidade de São Paulo
Escola Superior de Agricultura Luiz de Queiroz
Departamento de Engenharia de Biossistemas
LEB – 0400 – Ambiência, Zootecnia de Precisão e Bem-Estar animal.

Aula 2 – Sensores e atuadores no Arduino – Parte II

Para resolver os exemplos a seguir, utilize o Arduino UNO com a configuração no software do computador:

- Ferramentas -> Placa -> Arduino Genuino UNO
- Ferramentas -> Porta -> <A porta que o seu computador liberar quando o Arduino estiver conectado>.

Os módulos a serem utilizados nesta aula serão:

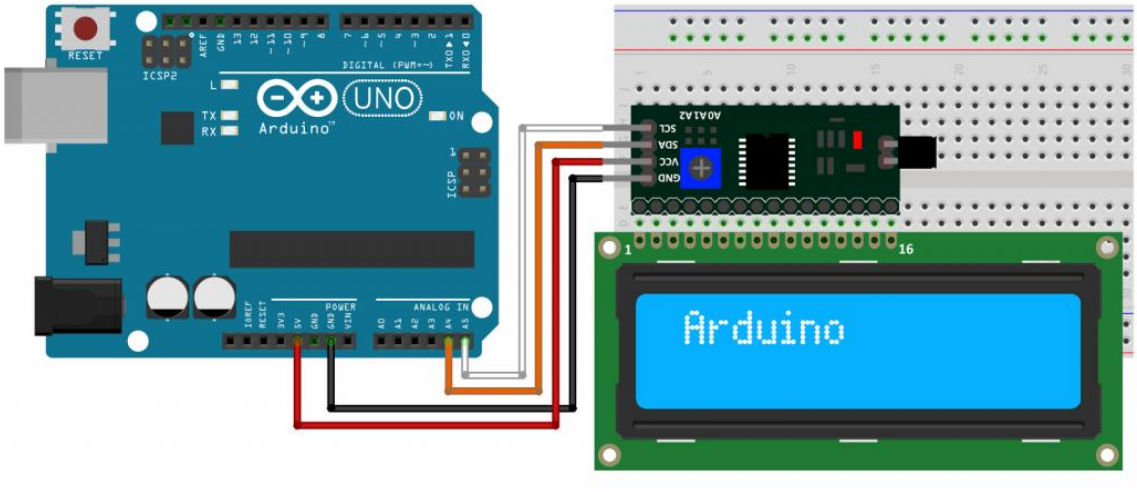
1. Módulo LED i2C
2. Display LCD (Display TFT 1.8" ST7735)
3. Display LCD (Display 0.91 OLED)
4. Sensor de temperatura - LM35
5. Sensor ultrassom para medir distâncias (HC-SR04)
6. Micro Servo Motor
7. Sensor de nível de água (Fuduino)
8. Sensor de umidade e temperatura DHT11
9. Módulo Relé Arduino (ligar uma lâmpada)
10. Teclado Membrana (KeyPad 4x4)
11. Sensor PIR (Detector) de Movimento
12. MQ-2 (gases inflamáveis (CO, LPG) e fumaça)
13. Umidade solo (higrômetro)

1. Módulo LED i2C

Possibilita a utilização de um LCD para exibição de caracteres. Será utilizado um segmento de 16 colunas por 2 linhas. Para tanto utilize o módulo i2C:

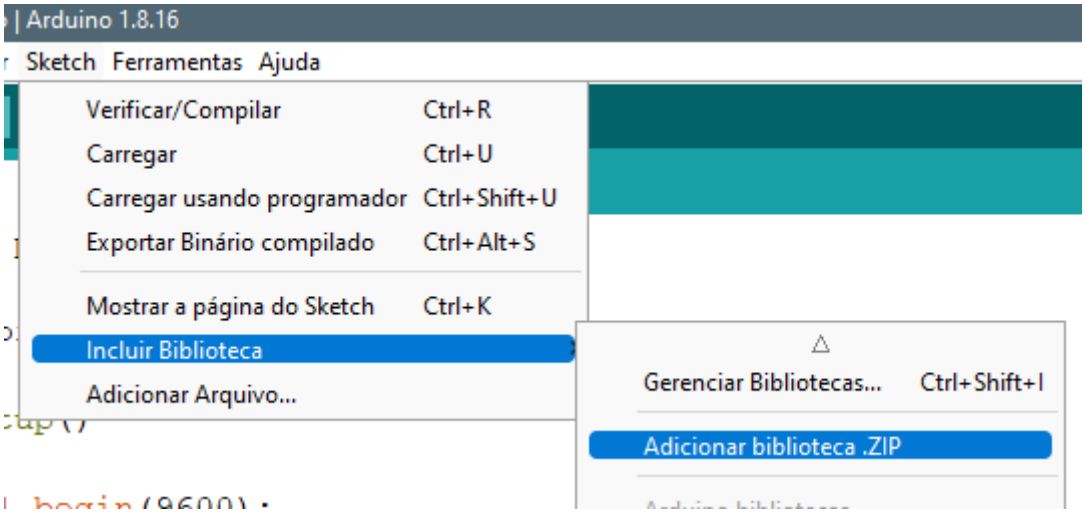
a. Monte o seguinte esquema

Obs.: ignore a protoboard, faça a conexão direta entre i2C e Arduino:



b. Baixar a Library:
https://www.arduino.cc/reference/en/libraries/liquidcrystal-i2c

c. Menu Sketch -> Incluir Bibliotecas -> Adicionar biblioteca .ZIP (e apontar para o arquivo ZIP do item anterior).



d. Em seguida, digite o seguinte código:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

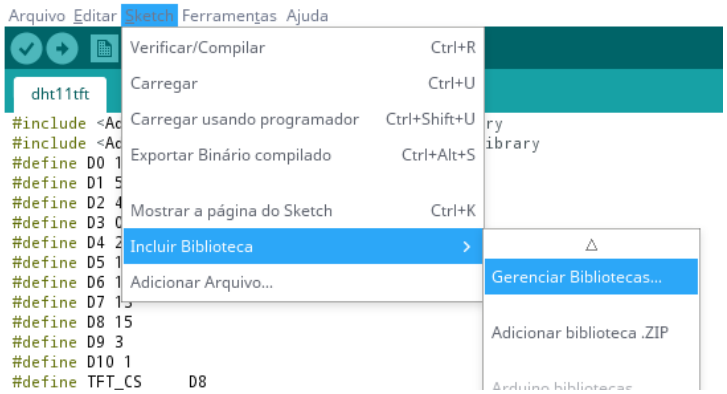
```
LiquidCrystal_I2C lcd(0x27,20,4);

void setup()
{
  lcd.init();
}


void loop()
{
  //acende a luz de fundo
  lcd.setBacklight(HIGH);
  //posiciona a digitacao na coluna 0 e linha 0
  lcd.setCursor(0,0);
  lcd.print("ESALQ");
  //posiciona a digitacao na coluna 0 e linha 0
  lcd.setCursor(0,1);
  lcd.print("Piracicaba");
  delay(4000);
  lcd.clear();

  //posiciona a digitacao na coluna 0 e linha 0
  lcd.setCursor(0,0);
  lcd.print("USP São Paulo");
  delay(4000);
  lcd.clear();


}
```

e. Execute!

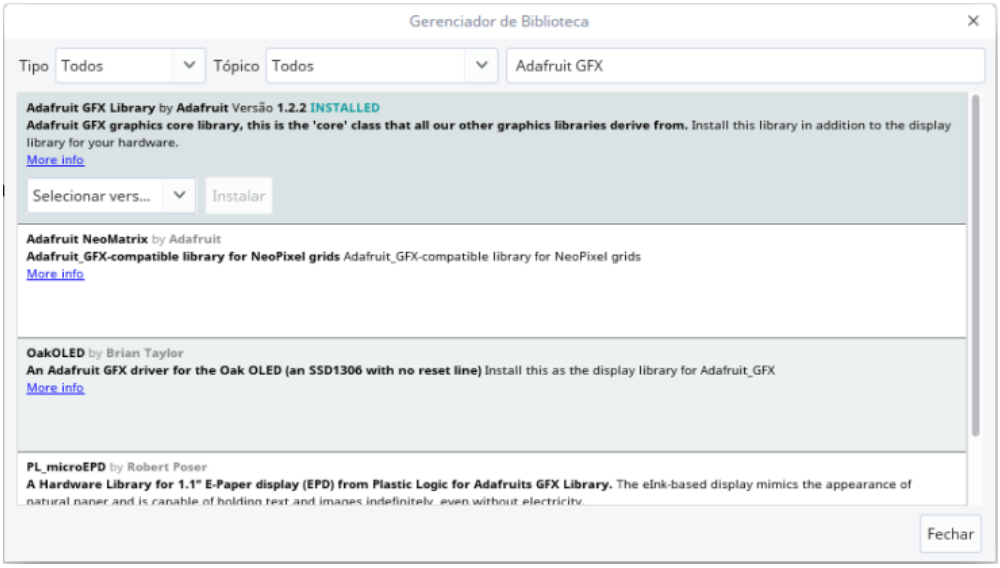
2. Display LCD (Display TFT 1.8” ST7735)

Fonte: https://portal.vidadesilicio.com.br/controlando-display-tft-st7735-nodemcu/

1. Adicione a biblioteca:
2. Menu Sketch -> Incluir biblioteca -> Gerenciar Bibliotecas



3. Procurar as bibliotecas Adafruit ST7735 e Adafruit GFX Library:



4. Pinagem no Arduino

Pino Display	Pino Arduino
VCC	5V
GND	GND
CS	10
RST	8
DC/A0	9
MOSI	11
SCK	13
LED	3.3V

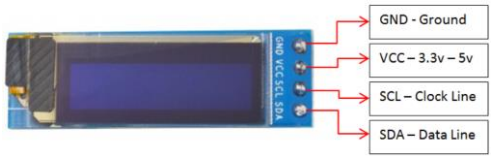
5. Codificação:

```
#include <Adafruit_GFX.h>    // Core graphics library
#include <Adafruit_ST7735.h> // Hardware-specific library

#define TFT_CS     10
#define TFT_RST    8
#define TFT_DC     9

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);

void setup(void){
  tft.setCursor(0,30); // Move o cursor para a coluna 30 na linha 0
  tft.initR(INITR_BLACKTAB); // Inicializa a tela com um fundo preto
  tft.setTextSize(2); // Seta o tamanho da fonte de texto para o tamanho 2
  tft.fillScreen(ST7735_BLACK); // Preenche a tela com a cor preta
  tft.setTextColor(ST7735_BLUE); // Seta a cor do texto para Azul
  tft.print("Vida"); // Escreve a palavra Vida Com a cor que foi setada na
linha acima
  tft.setTextColor(ST7735_WHITE); // Seta a cor do texto para Branco
  tft.print("De "); // Escreve a palavra Vida com a cor que foi setada na
linha acima
  tft.setTextColor(ST7735_BLUE); // Seta a cor do texto para Azul
  tft.print("Silicio"); // Escreve a palavra Silício, com a cor que foi setada
na linha acima

}
void loop(){
  tft.invertDisplay(true); // Inverte as cores que estão na tela
  delay(500); // Aguarda 0,5 segundos
  tft.invertDisplay(false); // Volta as cores originais
  delay(500); // Aguarda 0,5 segundos
}
```

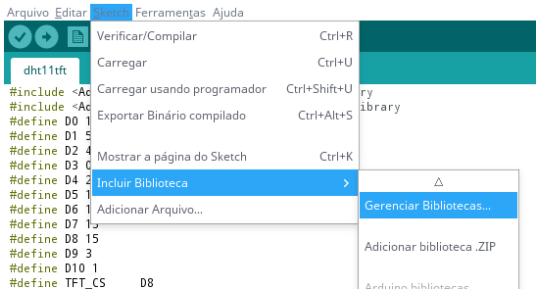
6. Execute!

3. Display LCD (Display 0.91 OLED)

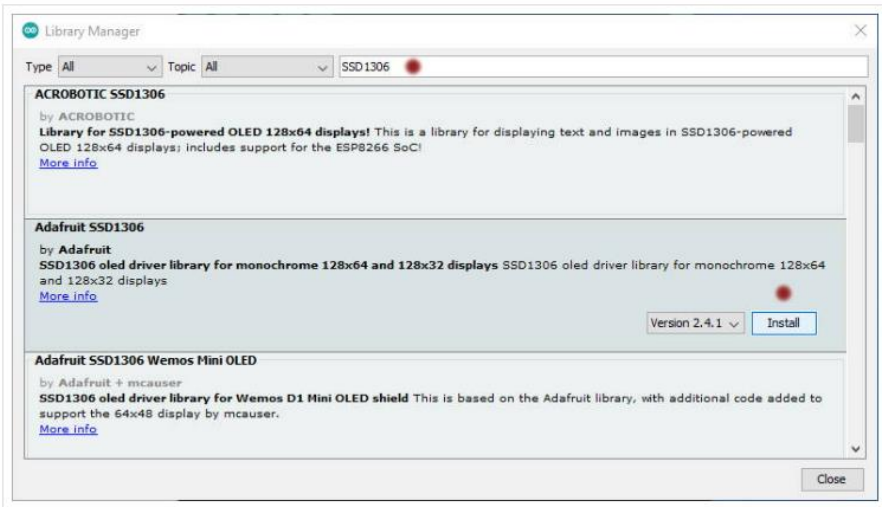
1. Pinagem:

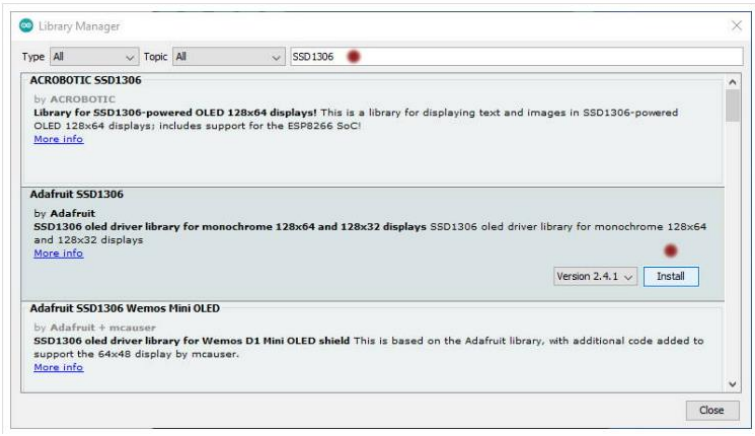


2. Menu Sketch -> Incluir biblioteca -> Gerenciar Bibliotecas



3. Procurar as bibliotecas SSD1306 e Adafruit GFX Library:





4. Copie o código:

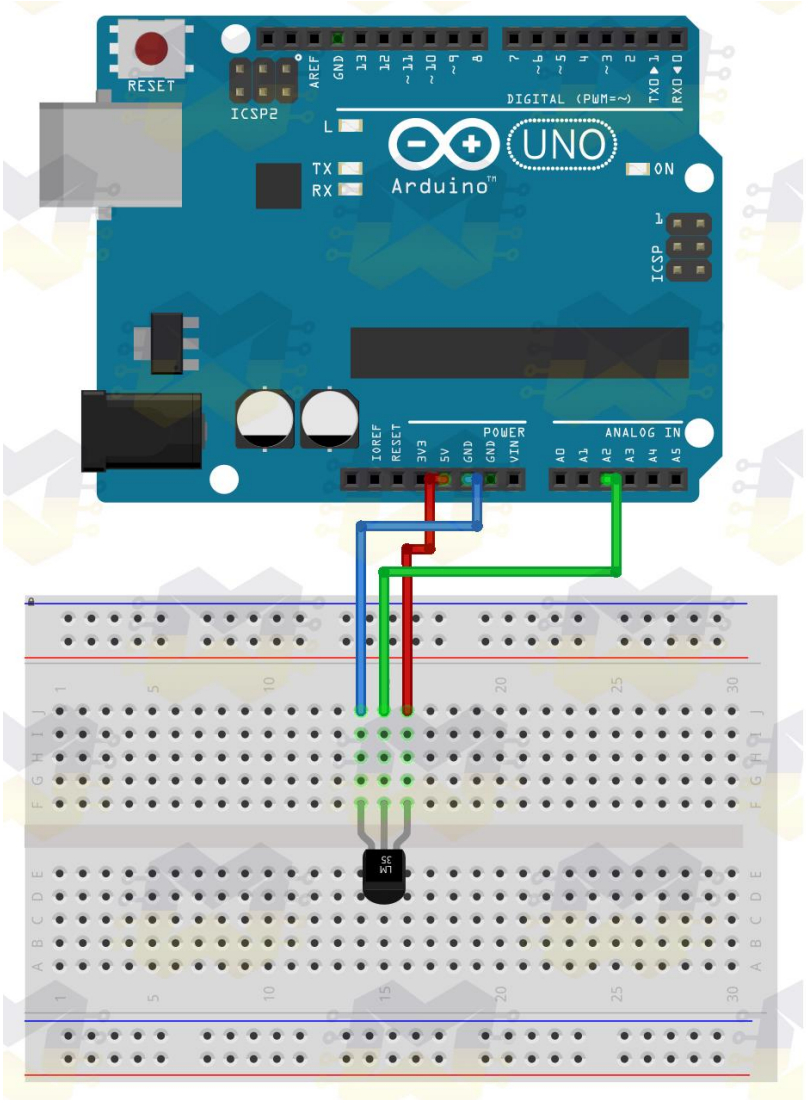
```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

Adafruit_SSD1306 display = Adafruit_SSD1306();

void setup() {
  Serial.begin(9600);

  Serial.println("OLED intialized");
  display.begin(SSD1306_SWITCHCAPVCC, 0x3C); // Address 0x3C for 128x32

  display.display();
  delay(1000);

  // Clear the buffer.
  display.clearDisplay();
  display.display();

  // text display tests
  display.setTextSize(1);
  display.setTextColor(WHITE);
}

void loop() {
    display.clearDisplay();
    display.setCursor(0, 0);
    display.println("ESALQ - USP");
    display.print("Piracicaba");
    display.display();
    delay(2000);
    display.clearDisplay();
    display.setCursor(0, 0);
    display.println("Curso de Arduino");
    display.println("OLED 0.91");
    display.print("Visualizacao das informacoes");
    display.display();
    delay(2000);
}
```

5. Execute!

4. Sensor de temperatura - LM35

Permite que você receba dados da temperatura ambiente. Para tanto você irá utilizar o seguinte sensor:

a. Monte a seguinte configuração no Arduino (observe que a parte reta – com a especificação do LM35 – deverá ficar voltada para a pinagem):



b. Em seguida, digite o seguinte código:

```
const int sensorPin = A2; //pino analogico do sensor
float temperatura = 0; //o valor lido serah armazenado nesta variavel

void setup(){
  Serial.begin(9600);
}

void loop(){
   temperatura = (analogRead(sensorPin) * 0.0048828125 * 100);
//varivel recebe a temperatura
   Serial.println("Temperatura = " +String(temperatura)+"C"); //exibe
o valor da temperatura na serial
   delay(2000); //aguarda 2 segundos para a proxima leitura
 }
```

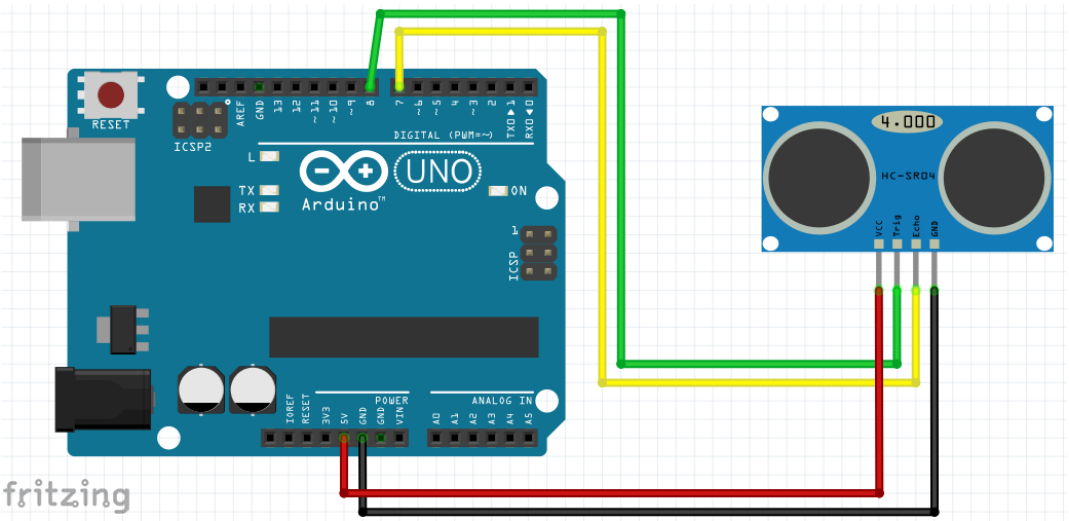
c. Carregue no Arduino. Em seguida exiba os dados no Monitor Serial:

- Ferramentas -> Monitor Serial (garanta que a leitura dos dados esteja em 9600).

5. Sensor ultrassom para medir distâncias (HC-SR04)

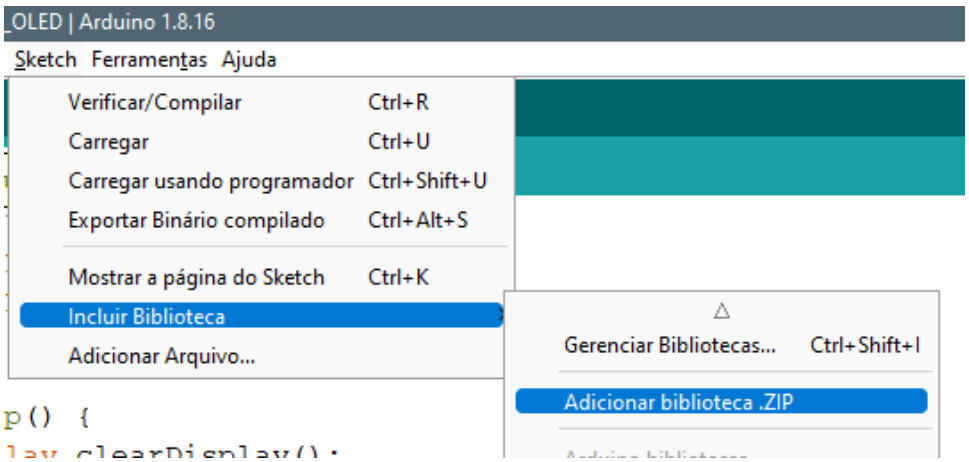
Permite medir distâncias entre o sensor e um objeto. Isso pode ser utilizado para evitar colisões de um robô com uma parede, por exemplo. O sensor utilizado será:

a. Monte o esquema como mostrado abaixo:



b. Faça download da library deste sensor:

- Feche o software do Arduino no computador
- Acesse o site: https://portal.vidadesilicio.com.br/wp-content/uploads/2017/05/Ultrasonic.zip
- Faça o download deste arquivo e inclua a biblioteca:



c. Digite o código abaixo:

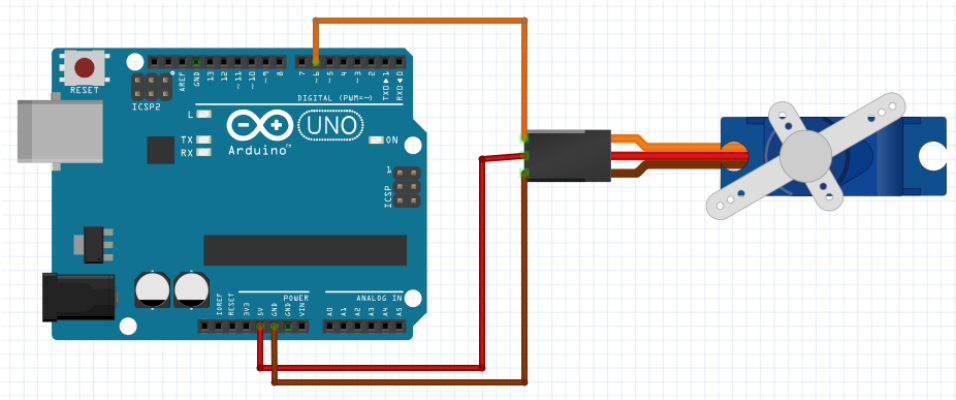
```
#include <Ultrasonic.h>

Ultrasonic ultrassom(8,7);

long distancia;

 void setup() {
 Serial.begin(9600);
 }

 void loop()
 {
   distancia = ultrassom.Ranging(CM);//define cm como medida da distancia
   Serial.print(distancia); //mostra o valor da medida realizada
   Serial.println(" cm");
   delay(1000);//dorme 1 segundo
 }
```

d. Execute!

6. Micro Servo Motor

O Micro Servo 9g Sg90 TowerPro é um micro servo motor que possui ângulos de rotação de até 180º. O motor a ser usado será este:

a. Monte o esquema como mostrado abaixo:



b. Digite o seguinte código:

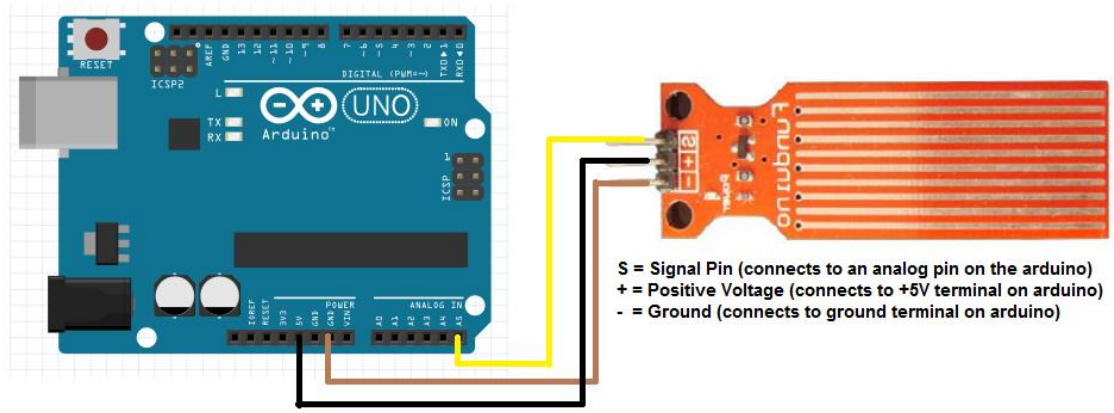
```
#include <Servo.h>
#define SERVO 6 // utiliza a porta digital 6 para ligar o motor

Servo s; // servo
int pos; // posicao de 0 a 180 graus

void setup ()
{
  s.attach(SERVO);
  Serial.begin(9600);
  s.write(0); // reseta o motor (coloca na posicao 0)
}

void loop()
{
  //faz girar para esquerda - avanca 180 graus
  for(pos = 0; pos < 180; pos++)
  {
    s.write(pos);//avanca
    Serial.println(pos);
    delay(5);//tempo para o motor girar. > mais lento e < mais rapido
  }

  //faz girar para direita - retorna 180 graus
  for(pos = 180; pos >= 0; pos--)
  {
    s.write(pos);//retorna
    Serial.println(pos);
    delay(5);//tempo para o motor girar. > mais lento e < mais rapido
  }
}
```

c. Execute o código. Abra o Monitor Serial para ver a saída (Ferrramentas -> Monitor Serial).

7. Sensor de nível de água (Fuduino)

Este módulo foi desenvolvido principalmente para os amadores de aquarismo e fornece a eles um esquema de detecção de nível de água de baixo custo e fácil de usar. O sensor que vou usar neste tutorial pode medir o nível da água até 40 mm (4 cm). Este é um sensor analógico e os dados que leremos terão valores de 0 a 1024. O sensor a ser utilizado é este:

a. Monte o esquema como mostrado abaixo:

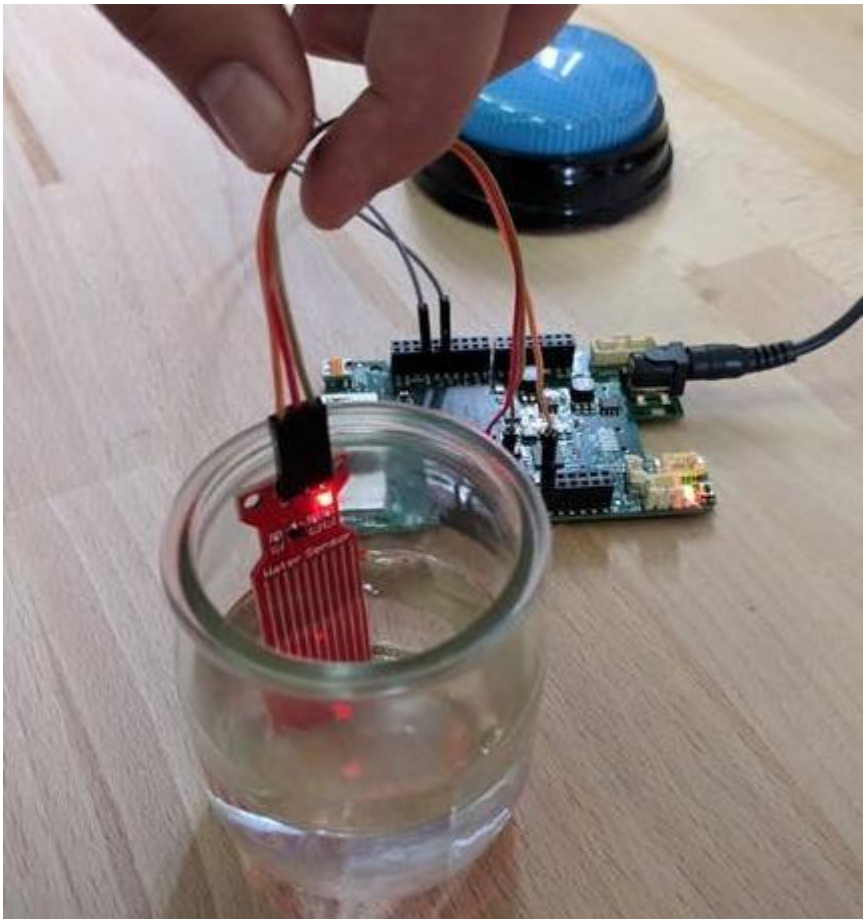


b. Em seguida, digite o código (obs.: o código carece de calibração):

```
int valorLeitura;

void setup(){
  Serial.begin(9600);
  Serial.println("Sensor de nivel de agua:");
}

void loop(){
  valorLeitura= analogRead(A5);
  if(valorLeitura>=0 && valorLeitura <480){
    Serial.println("Nivel da agua: 0mm - Vazio!");
  }else if(valorLeitura>=480 && valorLeitura <530){
    Serial.println("Nivel da agua: 0mm - 5mm");
  }else if(valorLeitura>=530 && valorLeitura <615){
    Serial.println("Nivel da agua: 5mm - 10mm");
  }else if(valorLeitura>=615 && valorLeitura <660){
    Serial.println("Nivel da agua: 10mm - 15mm");
  }else if(valorLeitura>=660 && valorLeitura <680){
    Serial.println("Nivel da agua: 15mm - 20mm");
  }else if(valorLeitura>=680 && valorLeitura <690){
    Serial.println("Nivel da agua: 20mm - 25mm");
  }else if(valorLeitura>=690 && valorLeitura <700){
    Serial.println("Nivel da agua: 25mm - 30mm");
  }else if(valorLeitura>=700 && valorLeitura <705){
    Serial.println("Nivel da agua: 30mm - 35mm");
  }else if(valorLeitura>=705 && valorLeitura <710){
    Serial.println("Nivel da agua: 35mm - 40mm");
  }
  delay(1000);
}
```

c. Execute. Abra o Monitor Serial (Ferramentas -> Monitor Serial).

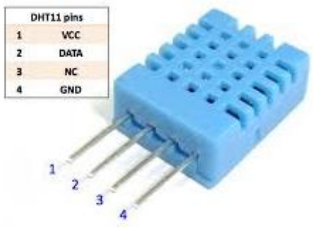
d. Pegue um copo com água e introduza o sensor (obs.: não deixe a água se aproximar e ultrapassar a palavra Fuduino).



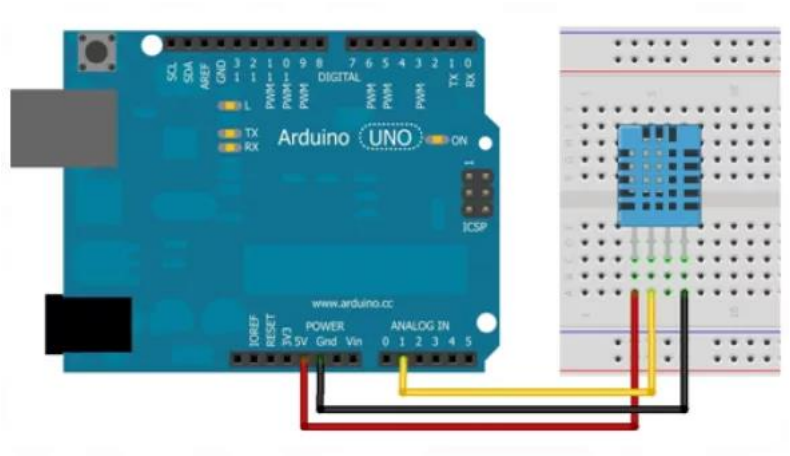
8. Sensor de umidade e temperatura DHT11

Fonte: https://portal.vidadesilicio.com.br/dht11-dht22-sensor-de-umidade-e-temperatura/

O sensor DHT11 é um sensor básico e de baixo custo que utiliza um termistor e um sensor capacitivo para medir a temperatura e a umidade do ar ambiente. Esse sensor é bastante simples de usar, mas requer cuidado com o tempo entre duas leituras consecutivas, uma vez que é necessário um intervalo de, no mínimo, 1 segundo entre uma leitura e outra. O sensor a ser utilizado é este:

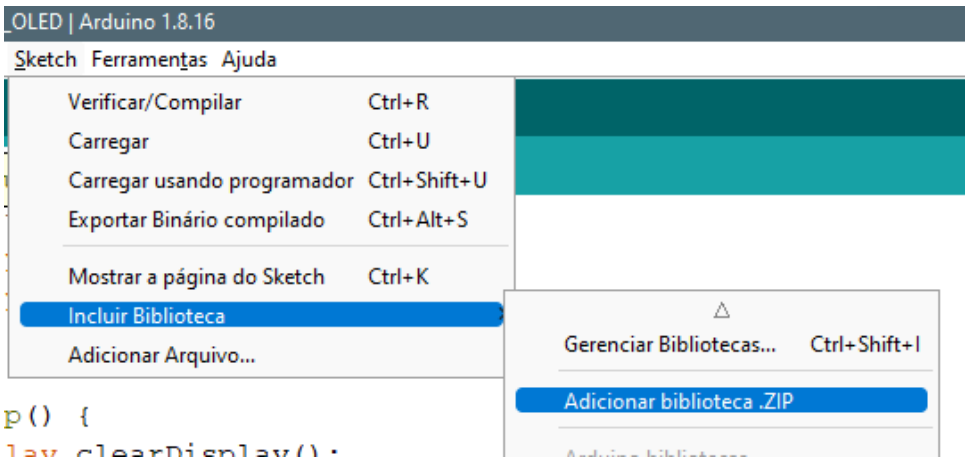


a. Monte o esquema como mostrado abaixo:



b. Faça download da library deste sensor:

- Feche o software do Arduino no computador
- Acesse o site: https://portal.vidadesilicio.com.br/wp-content/uploads/2017/05/DHTlib.zip
- Faça o download deste arquivo e inclua a biblioteca:



c. Digite o seguinte código:

```
#include <dht.h>

dht DHT; // Cria um objeto da classe dht
uint32_t timer = 0;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  delay(2000);
    DHT.read11(A1); // chama o metodo de leitura da classe dht com o pino de
transmissao de dados ligado no pino A1

    // Exibe na serial o valor de umidade
    Serial.print("Umidade: ");
    Serial.print(DHT.humidity);
    Serial.print("%");

    // Exibe na serial o valor da temperatura
    Serial.print(" - Temperatura: ");
    Serial.print(DHT.temperature);
    Serial.println(" Celsius");
}
```

d. Execute. Abra um Monitor Serial para acompanhar a visualização dos dados (Ferramentas -> Monitor Serial).

9. Módulo Relé Arduino (ligar uma lâmpada)

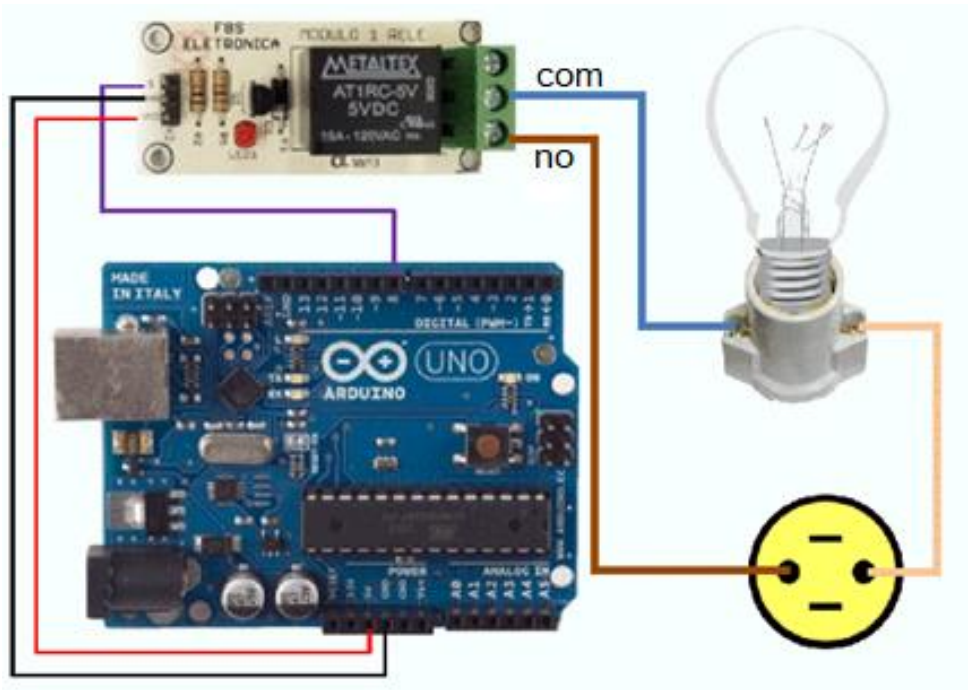
Fonte: https://athoselectronics.com/ligando-lampada-com-arduino/

O relé é um dispositivo eletromecânico que possui a capacidade de comutar sinais diferentes de forma isolada. Ele faz isso de uma forma bem simples, a partir de uma bobina, que quando energizada gera um campo magnético, que atrai um contato, e então um terminal que estava normalmente aberto passa a ser fechado e fluir corrente por ele, assim como, inversamente, o terminal que era normalmente fechado passa a ser aberto. O relé a ser utilizado será este:

Obs.: por ser um exemplo que irá estar energizado, algumas regras de segurança deverão obrigatoriamente serem seguidas:

- este exemplo somente poderá ser realizado com acompanhamento do professor;
- somente ligar na tomada quando for testar o circuito, do contrário ele deverá estar desconectado da tomada (esta regra se descumprida acarretará em advertência formal);
- ao conectar o fio ao relé tenha o cuidado de não deixar nenhuma rebarba de fio exposto (o cabo plástico do fio deverá estar encostado na boca de entrada);
- cuidado ao manipular a lâmpada, especialmente na base que conecta ao bocal pois parte dela poderá ficar exposta se for mal enroscado;

a. Monte o esquema como mostrado abaixo:



b. Digite o seguinte código:

```
const byte RELE = 8;

void setup()
{
   pinMode(RELE,OUTPUT);//define o pino do rele (8) como saida
}

void loop(){
  digitalWrite(RELE, HIGH); //Energiza o circuito: liga a lampada
  delay(2000); //Espera 2 segundos
  digitalWrite(RELE, LOW); // Desenergiza o circuito: desliga lampada
  delay(2000); //Espera 2 segundos
}
```

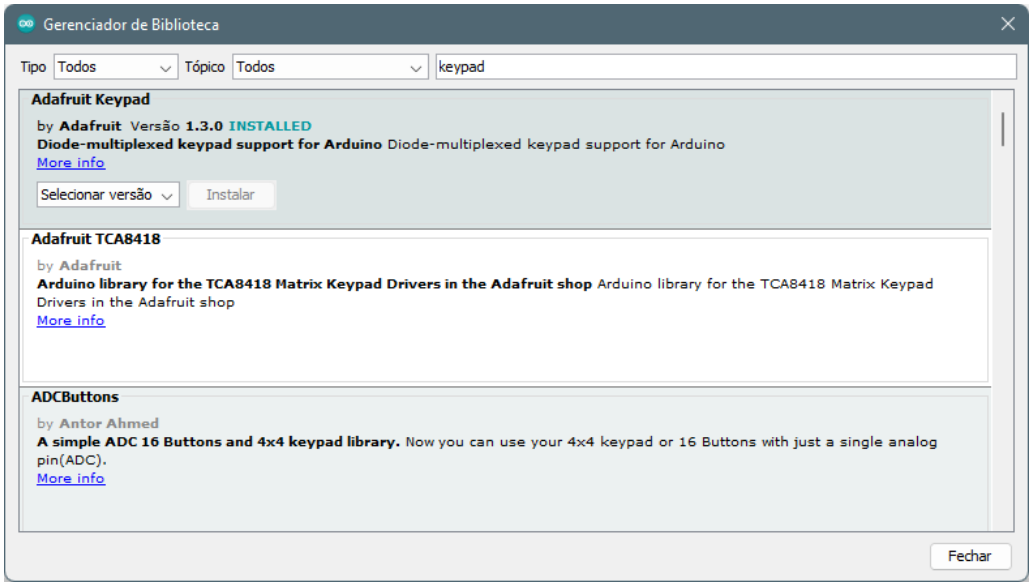
c. Energize o circuito (o professor deverá estar ao seu lado neste momento).

d. Ligue o Arduino ao computador via USB. Compile e carregue o código.

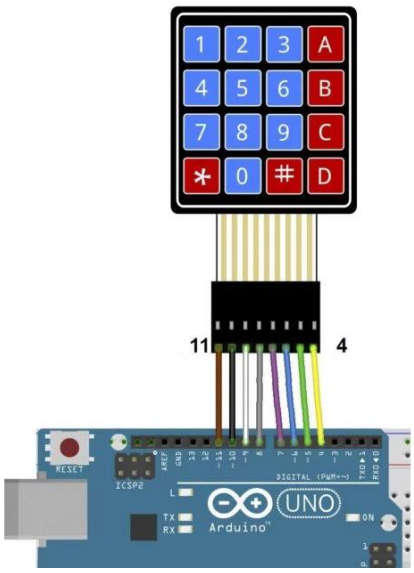
10. Teclado Membrana (KeyPad 4x4)

1. O teclado a ser utilizado será este:

2. Adicione a biblioteca:
    a. Sketch -> Incluir Biblioteca -> Gerenciar Biblioteca
    b. Pesquise por keypad
    c. Baixe a versão da Adafruit



3. Monte o seguinte esquema:



4. Digite o código:

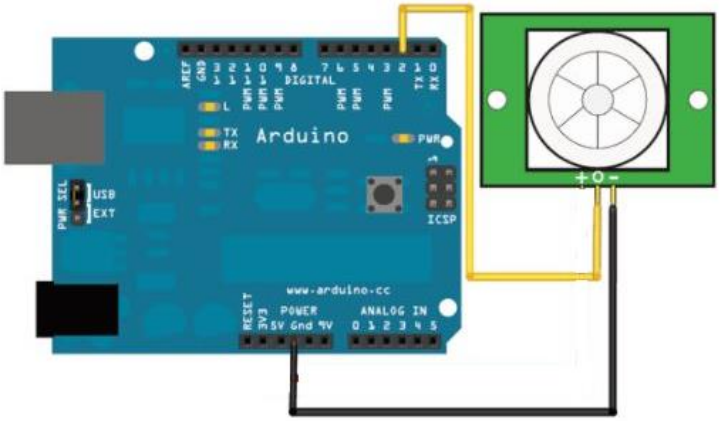
```
#include <Keypad.h>

const byte n_rows = 4;
const byte n_cols = 4;

char keys[n_rows][n_cols] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};

byte colPins[n_rows] = {7,6,5,4};
byte rowPins[n_cols] = {11,10,9,8};

Keypad myKeypad = Keypad( makeKeymap(keys), rowPins, colPins, n_rows, n_cols);

void setup(){
  Serial.begin(9600);
}

void loop(){
  char myKey = myKeypad.getKey();

  if (myKey != NULL){
    Serial.print("Key pressed: ");
    Serial.println(myKey);
  }
}
```

5. Execute!

11. Sensor PIR (Detector) de Movimento

1. O sensor a ser utilizado será este:

2. A configuração será esta:



3. Copie o seguinte código:

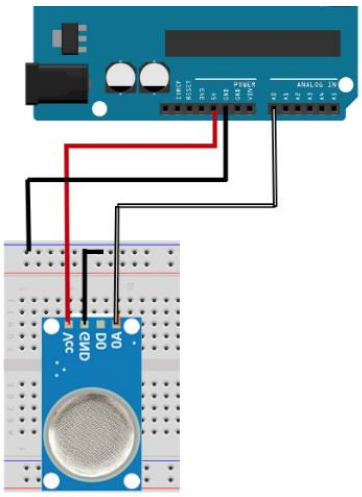
```
int inputPin = 2;
int val = 0;

void setup() {
  Serial.begin(9600);
  pinMode(inputPin, INPUT);
}

void loop(){
  val = digitalRead(inputPin);

  if (val == HIGH) {
    Serial.println("Movimento detectado");
  }else{
    Serial.println("Sem movimento");
  }

  delay(500);
}
```

4. Execute!

12. MQ-2 (gases inflamáveis (CO, LPG) e fumaça)

1. O sensor a ser utilizado será este:

2. A configuração será esta:



3. Copie o código abaixo:

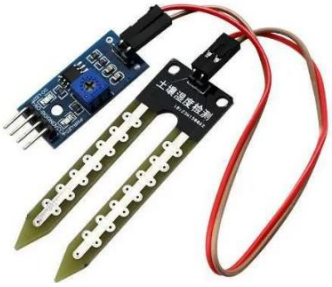
```
int PinA0 = A0;//PINO UTILIZADO PELO SENSOR DE GÁS MQ-2

void setup(){
  Serial.begin(9600);
  pinMode(PinA0, INPUT);
}

void loop(){
  int valor_analogico = analogRead(PinA0);
  Serial.print("Leitura: ");
  Serial.println(valor_analogico);
  delay(100);
}
```

4. Execute!
5. Observação: o código anterior apresenta apenas o valor da resistividade do sensor mediante a exposição aos gases por ele detectados. É necessário realizar curva de calibração com sensores comerciais para ter precisão aos valores lidos. Contudo a biblioteca abaixo faz isso, mas sem garantias:

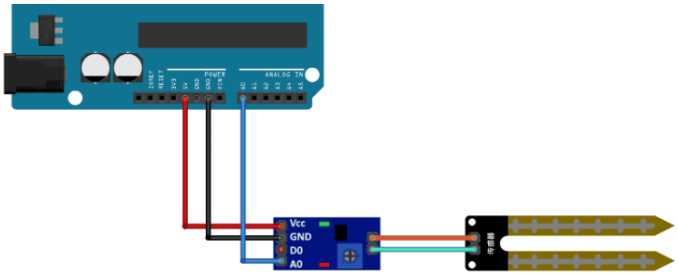
   https://github.com/labay11/MQ-2-sensor-library

13. Umidade solo (higrômetro)

1. O sensor a ser utilizado será este:



2. A configuração será esta:



3. Copie o código abaixo:

```
#define pino_higrometro A0

int valor_analogico;

void setup()
{
  Serial.begin(9600);
  pinMode(pino_higrometro, INPUT);
}

void loop()
{
  valor_analogico = analogRead(pino_higrometro);

  Serial.print("Valor: ");
  Serial.println(valor_analogico);

  delay(250);

}
```

4. Execute!

5. Observação: o código anterior apresenta apenas o valor da resistividade do sensor mediante a exposição a umidade. É necessário realizar curva de calibração com sensores comerciais para ter precisão aos valores lidos. Contudo o site abaixo apresenta um modelo algoritmo que faz isso, mas sem garantias:

https://portal.vidadesilicio.com.br/sensor-de-umidade-do-solo-higrometro/