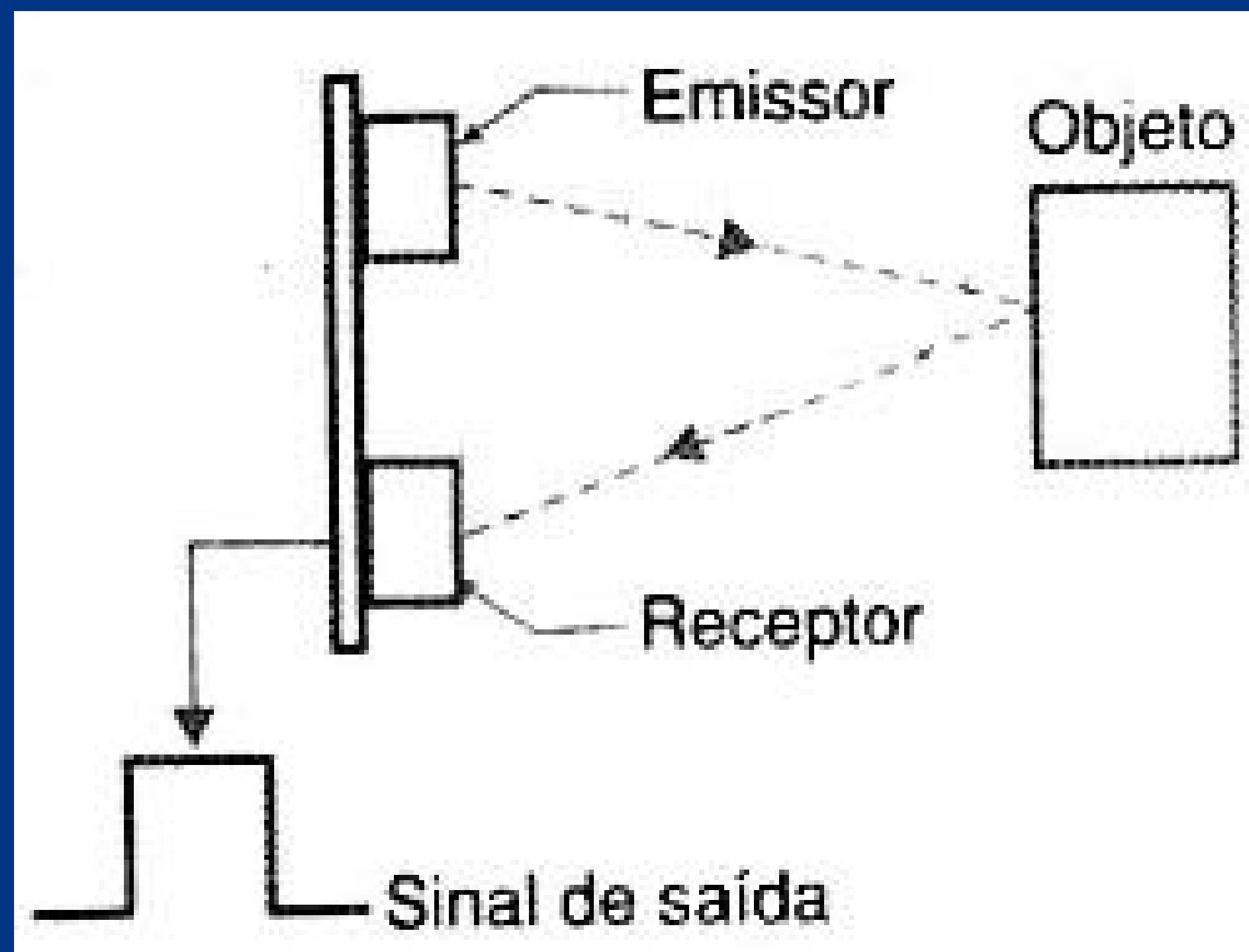


Sensor Ultrassônico



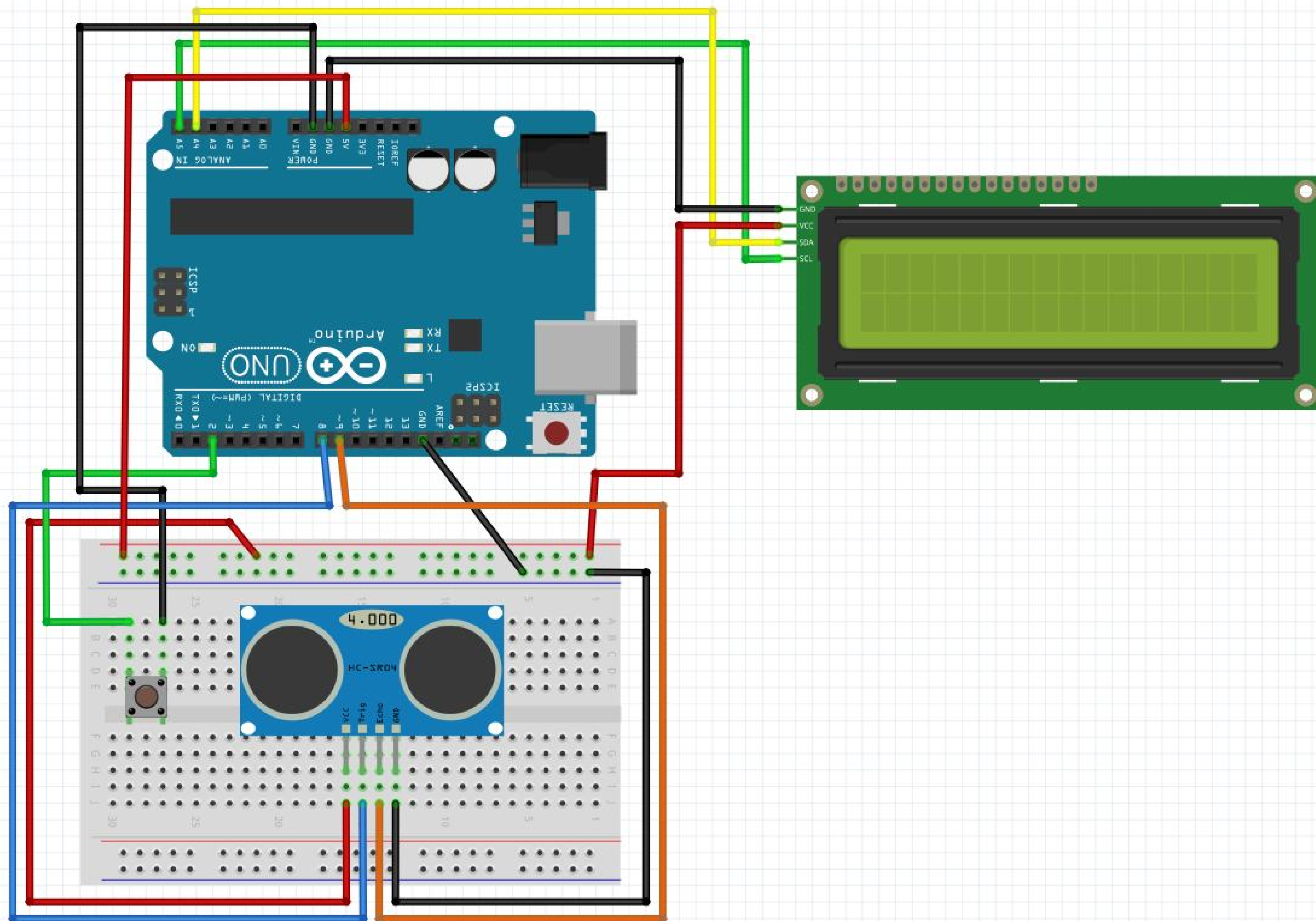
Como funciona?



Display LCD



Vamos fazer uma régua



Início do código

```
1  #include <NewPing.h>           // Inclui a biblioteca NewPing para o módulo HC-SR04
2  #include <LiquidCrystal_I2C.h> // Inclui a biblioteca LCD com I2C
3
4  // Define pinos de entrada e saída do Arduino
5  const byte pinTrig = 8; // Pino usado para disparar os pulsos do sensor
6  const byte pinEcho = 9; // Pino usado para ler a saída do sensor
7  const byte pinBut = 2;  // Pino usado para o push button
8
9  #define col 16 // Define o número de colunas do display utilizado
10 #define lin 2  // Define o número de linhas do display utilizado
11 #define ende 0x27 // Define o endereço do display
12 #define MAX_DISTANCE 200 // Distância máxima em cm que será medida
13
14 // Instancia objetos das bibliotecas
15 LiquidCrystal_I2C lcd(ende, col, lin);
16 NewPing sonar(pinTrig, pinEcho, MAX_DISTANCE); // Cria o objeto sonar usando NewPing
17
18 // Define variáveis globais
19 byte state = 1;           // Estado do menu
20 byte decimal = 0;         // Número de casas decimais
21 String unity = "cm";      // Descrição das unidades
22 double result = 0;        // Resultado da medida
```

void setup

```
void setup(){
  // Pin mode - define entradas e saídas digitais
  pinMode(pinBut, INPUT_PULLUP);  // Pullup interno do Arduino

  lcd.init(); // Inicializa a comunicação com o display já conectado
  lcd.clear(); // Limpa a tela do display
  lcd.backlight(); // Aciona a luz de fundo do display
  lcd.setCursor(0, 0); // Coloca o cursor do display na coluna 1 e linha 1
  lcd.print("Bora medir?"); // Exibe a mensagem na primeira linha do display
  lcd.setCursor(0, 1); // Coloca o cursor do display na coluna 1 e linha 2
  lcd.print("Ass: Tio Kiwi"); // Exibe a mensagem na segunda linha do display
  delay(1000);
  lcd.clear();

  // Inicializa a porta serial
  Serial.begin(9600);
}
```

void loop e readPushButton

```
void loop(){
    readPushButton(); // Verifica se o pushbutton foi acionado
    sowDisplay();     // Exibe a distância no display lcd
    delay(150); // Delay de 150ms
}

void readPushButton() { // Verifica se o botão foi acionado - altera para cm, mm ou pol
    if (!digitalRead(pinBut)) { // Verifica se o botão foi acionado (LOW)
        state = (state % 3) + 1; // Alterna entre 1, 2 e 3
        while (!digitalRead(pinBut)) {} // Aguarda soltar o botão
        lcd.clear();
        delay(150); // Delay para reduzir o efeito bouncing
    }
}
```

Show Display

```
void showDisplay() { // Exibe resultados no display LCD
    switch (state) {
        case 1:
            result = sonar.ping_cm(); // Medida em cm
            decimal = 0;
            unity = " cm";
            break;

        case 2:
            result = sonar.ping_cm() * 10; // Converte para mm
            unity = " mm";
            decimal = 0;
            break;

        case 3:
            result = sonar.ping_in(); // Medida em polegadas
            unity = " pol";
            decimal = 1;
            break;
    }
    lcd.setCursor(0, 0);
    lcd.print(" ");
    lcd.print(result, decimal); // Exibe no display as medidas de distância em cm, mm e polegadas
    lcd.print(unity);
    lcd.print(" ");
    lcd.setCursor(4, 1);
    lcd.print("Robotics");
}
```


Tarefa

