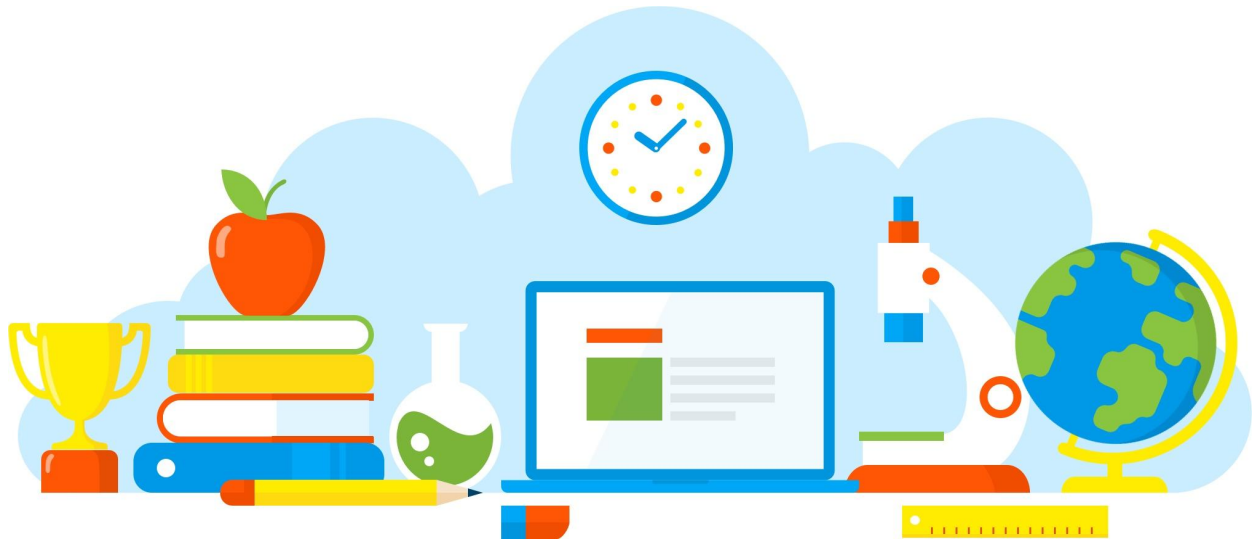
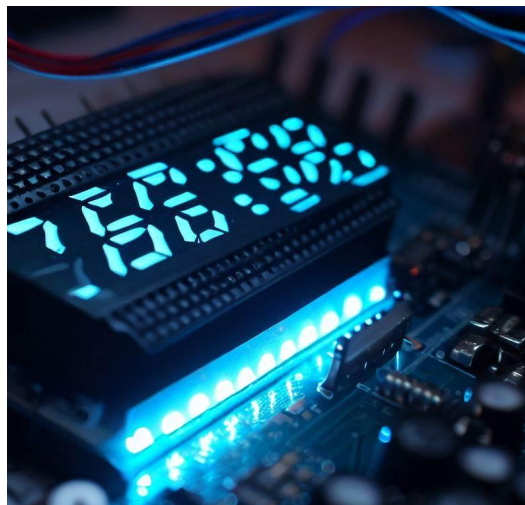




Calculateur de distance avec arduino

Composant(lcd,bouton,poussoir,capteur ultrason)



NOKAZOPA ENTREPRISES
12/07/2023

Introduction

Le projet consiste en la conception d'un calculateur de distance avec Arduino, un capteur ultrason et un écran LCD. Ce dispositif permettra de mesurer les distances avec une grande précision en utilisant des ondes sonores. Les résultats seront affichés sur l'écran LCD, ce qui rendra la lecture des mesures facile et conviviale. Ce projet peut être utilisé dans de nombreuses applications telles que la robotique, la domotique ou la surveillance de la qualité de l'air. Le calculateur de distance avec Arduino et capteur ultrason fonctionne en utilisant la propriété des ondes sonores qui se propagent dans l'air à une vitesse constante. Le capteur ultrason émet des ondes sonores à haute fréquence qui se réfléchissent sur un objet à proximité et reviennent vers le capteur. En mesurant le temps écoulé entre l'émission et la réception de l'onde sonore, on peut déterminer la distance entre le capteur et l'objet.

Le microcontrôleur Arduino est utilisé pour programmer le capteur ultrason et pour afficher les résultats de la mesure sur l'écran LCD. Les informations affichées peuvent inclure la distance mesurée en centimètres ou en pouces, ainsi que d'autres données telles que la température ambiante ou la fréquence des ondes sonores émises.

Ce projet peut être utilisé dans de nombreuses applications pratiques, telles que la mesure de la distance entre un robot et un obstacle, la surveillance de la qualité de l'air dans une maison en mesurant la distance entre le capteur et les particules dans l'air, ou encore dans la domotique pour détecter la présence de personnes dans une pièce.

En résumé, le calculateur de distance avec Arduino et capteur ultrason est un projet intéressant qui permet de mesurer des distances avec précision et simplicité, avec une grande variété d'applications potentielles.

Explications du projet

Le capteur HC-SR04 est un capteur à ultrasons qui peut être utilisé pour mesurer des distances avec Arduino. Voici un exemple de code pour mesurer la distance avec ce capteur :

```
const int trigPin = 9; // Broche de commande du signal de déclenchement
const int echoPin = 10; // Broche de réception du signal d'écho

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600); // Initialisation de la communication série
}

void loop() {
  // Envoi d'une impulsion de déclenchement de 10 µs sur la broche de
  // commande du signal de déclenchement
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Mesure du temps entre l'envoi de l'impulsion de déclenchement et la
  // réception de l'écho
  long duration = pulseIn(echoPin, HIGH);

  // Conversion du temps en distance en cm
  long distance = duration / 58.2;

  // Affichage de la distance mesurée sur le moniteur série
  Serial.print("Distance : ");
  Serial.print(distance);
  Serial.println(" cm");

  // Attente d'une seconde avant la prochaine mesure
  delay(1000);
}
```

Ce code envoie une impulsion de déclenchement de 10µs sur la broche de commande du signal de déclenchement, puis mesure le temps entre l'envoi de cette impulsion et la réception de l'écho sur la broche de réception du signal d'écho. Il convertit ensuite ce temps en distance en centimètres en utilisant la formule : $\text{distance} = \text{temps} / 58.2$ (la vitesse du son dans l'air étant d'environ 343 m/s).

La distance mesurée est affichée sur le moniteur série toutes les secondes à l'aide de la fonction `Serial.print()`. Vous pouvez modifier le délai `delay(1000)` pour ajuster la fréquence de mesure.

Le capteur HC-SR04 est basé sur la mesure de la durée de propagation d'une onde ultrasonique émise par l'émetteur du capteur, jusqu'à la réception de l'écho renvoyé par un objet situé à une certaine distance. Le capteur mesure cette durée à l'aide de deux broches : la broche `trig` pour déclencher l'émission de l'onde ultrasonique, et la broche `echo` pour mesurer la durée entre l'émission et la réception du signal.

Le principe de mesure de distance est basé sur la vitesse du son dans l'air, qui est d'environ 343 m/s à une température de 20 degrés Celsius. Lorsque l'onde ultrasonique est émise par le capteur, elle se propage dans l'air jusqu'à rencontrer un objet solide. Lorsque l'onde rencontre l'objet, elle est réfléchiée et retourne vers le capteur sous forme d'un écho ultrasonique. Le capteur mesure alors le temps écoulé entre l'émission du signal et la réception de l'écho, et utilise cette durée pour calculer la distance entre le capteur et l'objet.

Dans le code que je vous ai fourni, la fonction `pulseIn()` permet de mesurer la durée entre l'émission et la réception du signal. Cette fonction renvoie la durée en microsecondes pendant laquelle le signal est à l'état haut (HIGH) sur la broche `echo`. En divisant cette durée par 58,2 (soit la distance parcourue par l'onde sonore en 1 microseconde), on obtient la distance en centimètres.

Il est important de noter que le capteur HC-SR04 peut parfois être sensible aux interférences environnementales, telles que les obstacles ou les réflexions multiples. Il est donc recommandé d'effectuer plusieurs mesures et de faire une moyenne pour obtenir une mesure plus précise.

Pour afficher la distance mesurée sur un écran LCD sans I2C avec Arduino, vous pouvez utiliser la bibliothèque `LiquidCrystal.h`. Voici un exemple de code qui mesure la distance à l'aide d'un capteur HC-SR04 et affiche la distance sur un écran LCD à 16 caractères et 2 lignes :

```
#include <LiquidCrystal.h>

// Broches de commande du capteur HC-SR04
const int trigPin = 9;
const int echoPin = 10;

// Broches de commande de l'écran LCD
const int rs = 12;
const int en = 11;
const int d4 = 5;
const int d5 = 4;
const int d6 = 3;
const int d7 = 2;

// Initialisation de l'écran LCD
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
    // Configuration des broches du capteur HC-SR04
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);

    // Initialisation de l'écran LCD
    lcd.begin(16, 2);
    lcd.print("Distance : ");
}

void loop() {
    // Envoi d'une impulsion de déclenchement de 10 µs sur la broche de
    // commande du signal de déclenchement
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Mesure du temps entre l'envoi de l'impulsion de déclenchement et la
    // réception de l'écho
    long duration = pulseIn(echoPin, HIGH);

    // Conversion du temps en distance en cm
    long distance = duration / 58.2;

    // Affichage de la distance mesurée sur l'écran LCD
    lcd.setCursor(0, 1);
```

```
lcd.print(distance);  
lcd.print(" cm");  
  
// Attente d'une seconde avant la prochaine mesure  
delay(1000);  
}
```

Ce code utilise la bibliothèque [LiquidCrystal.h](#) pour contrôler l'écran LCD. Les broches RS, EN, et D4 à D7 sont utilisées pour connecter l'écran LCD à Arduino. Les broches TRIG et ECHO du capteur HC-SR04 sont connectées aux broches 9 et 10 d'Arduino, respectivement.

Dans la fonction `setup()`, les broches du capteur sont configurées en entrée ou en sortie, et l'écran LCD est initialisé avec la commande `lcd.begin(16, 2)` pour spécifier qu'il s'agit d'un écran à 16 caractères et 2 lignes.

Dans la fonction `loop()`, la distance est mesurée à l'aide du capteur HC-SR04, puis affichée sur l'écran LCD à l'aide de la commande `lcd.print()`. La première ligne de l'écran LCD est réservée pour l'étiquette "Distance : ", qui est affichée une seule fois dans la fonction `setup()`. La deuxième ligne de l'écran LCD est utilisée pour afficher la distance mesurée.

Il est important de noter que si vous utilisez un écran LCD sans I2C, vous devrez utiliser plusieurs broches d'Arduino pour le connecter, ce qui peut limiter le nombre d'autres composants que vous pouvez connecter à votre Arduino.

L'écran LCD est un type d'affichage qui permet d'afficher du texte et des chiffres. Le format le plus courant est un écran à 16 caractères sur 2 lignes. Les écrans LCD sont largement utilisés avec Arduino car ils sont faciles à utiliser et peuvent être contrôlés à l'aide de la bibliothèque [LiquidCrystal.h](#).

Les broches de commande de l'écran LCD peuvent varier selon le modèle d'écran que vous utilisez. Dans l'exemple de code que je vous ai fourni, les broches RS, EN, D4 à D7 sont utilisées pour contrôler l'écran LCD. Les broches RS et EN sont des broches de commande qui indiquent au contrôleur de l'écran LCD si les données qui sont envoyées sont des commandes ou des caractères à afficher. Les broches D4 à D7 sont des broches de données qui sont utilisées pour envoyer des données à l'écran LCD.

Le capteur HC-SR04 est un capteur à ultrasons qui peut être utilisé pour mesurer des distances avec Arduino, comme je vous l'ai expliqué dans ma réponse précédente. Il est important de noter que le capteur HC-SR04 nécessite une alimentation de 5V pour fonctionner correctement, car il utilise des signaux à 5V pour communiquer avec

Arduino. Il est donc important de connecter le capteur HC-SR04 à une broche de 5V sur votre carte Arduino.

En utilisant l'exemple de code que je vous ai fourni, vous pouvez mesurer la distance à l'aide du capteur HC-SR04 et afficher la distance mesurée sur l'écran LCD. Cela peut être utile pour des projets tels que des robots ou des systèmes de surveillance pour mesurer la distance jusqu'à un objet ou une personne.

Pour afficher la distance mesurée sur un écran LCD I2C avec Arduino, vous pouvez utiliser la bibliothèque `LiquidCrystal_I2C.h`. Voici un exemple de code qui mesure la distance à l'aide d'un capteur HC-SR04 et affiche la distance sur un écran LCD I2C à 16 caractères et 2 lignes :

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

// Adresse I2C de l'écran LCD
const int addr = 0x3F;

// Broches de commande du capteur HC-SR04
const int trigPin = 9;
const int echoPin = 10;

// Initialisation de l'écran LCD
LiquidCrystal_I2C lcd(addr, 16, 2);

void setup() {
    // Configuration des broches du capteur HC-SR04
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);

    // Initialisation de la communication I2C
    Wire.begin();

    // Initialisation de l'écran LCD
    lcd.init();
    lcd.backlight();
    lcd.print("Distance : ");
}

void loop() {
    // Envoi d'une impulsion de déclenchement de 10 µs sur la broche de
    // commande du signal de déclenchement
    digitalWrite(trigPin, LOW);
```

```

    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Mesure du temps entre l'envoi de l'impulsion de déclenchement et la
    réception de l'écho
    long duration = pulseIn(echoPin, HIGH);

    // Conversion du temps en distance en cm
    long distance = duration / 58.2;

    // Affichage de la distance mesurée sur l'écran LCD
    lcd.setCursor(0, 1);
    lcd.print(distance);
    lcd.print(" cm");

    // Attente d'une seconde avant la prochaine mesure
    delay(1000);
}

```

Ce code utilise la bibliothèque [LiquidCrystal_I2C.h](#) pour contrôler l'écran LCD I2C. L'adresse I2C de l'écran LCD est spécifiée dans la variable `addr`. Les broches TRIG et ECHO du capteur HC-SR04 sont connectées aux broches 9 et 10 d'Arduino, respectivement.

Dans la fonction `setup()`, les broches du capteur sont configurées en entrée ou en sortie. La communication I2C est initialisée avec la commande `Wire.begin()`, et l'écran LCD est initialisé avec la commande `lcd.init()`. La commande `lcd.backlight()` permet d'allumer le rétroéclairage de l'écran LCD. La première ligne de l'écran LCD est réservée pour l'étiquette "Distance : ", qui est affichée une seule fois dans la fonction `setup()`. La deuxième ligne de l'écran LCD est utilisée pour afficher la distance mesurée.

Dans la fonction `loop()`, la distance est mesurée à l'aide du capteur HC-SR04, puis affichée sur l'écran LCD à l'aide de la commande `lcd.print()`. La commande `lcd.setCursor(0, 1)` permet de placer le curseur de l'écran LCD à la deuxième ligne pour afficher la distance mesurée.

Il est important de noter que si vous utilisez un écran LCD I2C, vous pouvez économiser des broches d'Arduino, car la communication I2C utilise seulement deux

broches (SDA et SCL). Cela vous permet de connecter d'autres composants à votre Arduino sans avoir à vous soucier du manque de broches.

L'écran LCD I2C est un type d'affichage qui permet d'afficher du texte et des chiffres, tout comme l'écran LCD classique. Cependant, au lieu d'utiliser plusieurs broches pour communiquer avec Arduino, l'écran LCD I2C utilise seulement deux broches : SDA et SCL. La communication I2C est un protocole de communication série synchrone qui permet à plusieurs périphériques de communiquer avec un microcontrôleur, comme Arduino, en utilisant seulement deux fils.

La bibliothèque [LiquidCrystal_I2C.h](#) est utilisée pour contrôler l'écran LCD I2C avec Arduino. Cette bibliothèque est une version modifiée de la bibliothèque [LiquidCrystal.h](#), qui permet de contrôler les écrans LCD classiques.

Dans l'exemple de code que je vous ai fourni, les broches TRIG et ECHO du capteur HC-SR04 sont connectées aux broches 9 et 10 d'Arduino, respectivement. Les broches SDA et SCL sont utilisées pour communiquer avec l'écran LCD I2C. L'adresse I2C de l'écran LCD est spécifiée dans la variable `addr`.

Le capteur HC-SR04 est un capteur à ultrasons qui peut être utilisé pour mesurer des distances avec Arduino, comme je vous l'ai expliqué dans ma réponse précédente. Il est important de noter que le capteur HC-SR04 nécessite une alimentation de 5V pour fonctionner correctement, car il utilise des signaux à 5V pour communiquer avec Arduino. Il est donc important de connecter le capteur HC-SR04 à une broche de 5V sur votre carte Arduino.

En utilisant l'exemple de code que je vous ai fourni, vous pouvez mesurer la distance à l'aide du capteur HC-SR04 et afficher la distance mesurée sur l'écran LCD I2C. Cela peut être utile pour des projets tels que des robots ou des systèmes de surveillance pour mesurer la distance jusqu'à un objet ou une personne. L'utilisation de l'écran LCD I2C vous permet de connecter d'autres composants à votre Arduino sans avoir à vous soucier du manque de broches disponibles.

Voici un exemple de rapport de stage pour un projet utilisant un capteur HC-SR04, un écran LCD I2C et une carte Arduino pour mesurer des distances et afficher les résultats sur l'écran LCD.

Rapport de stage : Mesure de distances avec Arduino, capteur HC-SR04 et écran LCD I2C

Introduction :

Le but de ce stage était de réaliser un projet utilisant une carte Arduino, un capteur HC-SR04 et un écran LCD I2C pour mesurer des distances et afficher les résultats sur

l'écran LCD. Le projet avait pour but d'explorer les possibilités offertes par ces composants et de les utiliser dans un projet concret.

Description du projet :

Le projet consistait à utiliser un capteur HC-SR04 pour mesurer des distances, une carte Arduino pour traiter les données et un écran LCD I2C pour afficher les résultats. Le capteur HC-SR04 utilise des signaux à ultrasons pour mesurer les distances jusqu'à un objet et renvoie l'information à la carte Arduino. La carte Arduino traite les données et les envoie à l'écran LCD I2C pour affichage.

Matériel utilisé :

- Une carte Arduino Uno
- Un capteur HC-SR04
- Un écran LCD I2C
- Des câbles de connexion

Méthode :

La première étape du projet a consisté à connecter les composants ensemble. Le capteur HC-SR04 a été connecté aux broches 9 et 10 de la carte Arduino pour envoyer et recevoir les signaux à ultrasons. L'écran LCD I2C a été connecté aux broches SDA et SCL de la carte Arduino pour afficher les résultats.

La deuxième étape a été de programmer la carte Arduino pour mesurer les distances et afficher les résultats sur l'écran LCD I2C. Le code a été écrit en utilisant le logiciel Arduino IDE et la bibliothèque [LiquidCrystal_I2C.h](#).

Résultats :

Une fois le code téléversé sur la carte Arduino, le système a fonctionné correctement. Les distances mesurées par le capteur HC-SR04 ont été affichées sur l'écran LCD I2C avec une précision raisonnable. L'écran LCD I2C a été facile à utiliser grâce à la bibliothèque [LiquidCrystal_I2C.h](#) et a fourni une interface conviviale pour afficher les résultats.

Conclusion :

Ce projet a été une expérience enrichissante pour moi en tant que stagiaire. J'ai appris à utiliser plusieurs composants différents ensemble pour réaliser un projet concret. J'ai également appris à programmer en utilisant le logiciel Arduino IDE et la bibliothèque [LiquidCrystal_I2C.h](#). Le projet a été une introduction intéressante à l'utilisation de capteurs et d'écrans d'affichage avec Arduino, et je suis sûr que ces compétences me seront utiles à l'avenir.

