# Arduino-GFS

Von Jonas Heilig

2.2

VCC -> 5V

GND -> GND

SDA -> A4

SCL -> A5

Code 2.2

#include <Wire.h>

#include <LiquidCrystal\_I2C.h> -> importiert Module: Wire.h für I2C, LiquidCrystal\_I2C.h

Für LCD

const int LCD\_COLUMNS = 16; -> Felder einer Zeile

const int LCD\_ROWS = 2; -> Wie viel Zellen hat das Display

const int LCD\_ADDRESS = 0x27. -> Adresse der I2C Schnittstelle

Bei jeder Art von Gerät anders. Bei allen

LCD I2C 0x27

* Die Pins des LCD müssen nicht angegeben werden, da sie schon in der Bibliothek angegeben sind.

LiquidCrystal\_I2C lcd(LCD\_ADDRESS, LCD\_COLUMNS, LCD\_ROWS); -> Erstellt Variabel lcd mit den Parametern die die Bibliotheken brauchen. (LiquidCrystal\_I2C heißt erbt von der Bibliothek LiquidCrystal\_I2C.h)

void setup() { -> wird nur einmal beim Start ausgeführt

  Serial.begin(9600); -> Zum Log schereiben, wir benötigen es nicht

  Wire.begin(); -> initialisiert das I2C -> Bestimmte Einstellungen werden geladen

  lcd.init(); -> initialisiert das LCD -> Bestimmte Einstellungen werden geladen

  lcd.backlight(); -> schaltet das Hintergrundlicht am LCD an

}

void loop() { -> wird als loop ausgeführt

  lcd.clear(); -> löscht alle Felder

  lcd.setCursor(0, 0); -> setzt den Ort des Schreibens. 0 = Feld 1; 0 = Zeile 1. (Feld, Zeile)

  lcd.print("Hello,"); -> schreibt ab dem angegeben Punkt

  lcd.setCursor(0, 1); -> setzt den Ort des Schreibens. 0 = Feld 1; 1 = Zeile 2. (Feld, Zeile)

  lcd.print("World!"); -> schreibt ab dem angegeben Punkt

  delay(2000); -> wartet 2 Sekunden

  lcd.clear(); -> löscht alle Felder

  lcd.setCursor(0, 0); -> setzt den Ort des Schreibens. 0 = Feld 1; 0 = Zeile 1. (Feld, Zeile)

  lcd.print("LiquidCrystal"); -> schreibt ab dem angegeben Punkt

  lcd.setCursor(0, 1); -> setzt den Ort des Schreibens. 0 = Feld 1; 1 = Zeile 2. (Feld, Zeile)

  lcd.print("Display mit I2C"); -> schreibt ab dem angegeben Punkt

  delay(2000);

} -> wartet 2 Sekunden

Aufgaben:

1. Ändere den Text des LCD
2. Setze einen Text mittig.
3. Mache das der Text Länger bei einem Textblock bleibt
4. Füge einen 3 Textblock hinzu. -> Nach delay immer lcd.clear()

2.3

#include <Servo.h>

const int TASTER1\_PIN = 2; -> definiere Pin als Variable

const int TASTER2\_PIN = 3; -> definiere Pin als Variable

const int SERVO\_PIN = 9; -> definiere Pin als Variable

Servo servo; -> definiere das Objekt servo aus Bibliothek

void setup() { -> wird nur einmal beim Start ausgeführt

  Serial.begin(9600); -> Zum Log schereiben

  pinMode(TASTER1\_PIN, INPUT); -> definiere Pin für den Code

  pinMode(TASTER2\_PIN, INPUT); -> definiere Pin für den Code

  servo.attach(SERVO\_PIN);

}

void loop() { -> wird als loop ausgeführt

  if (digitalRead(TASTER1\_PIN) == HIGH) { -> schaut ob gedrückt

    Serial.println("Taster 1 gedrückt - Servo fährt vorwärts"); -> Log schreiben

    servo.write(180); -> setze servo auf 180°

    delay(1000); -> Warte 1 Sekunde

    servo.write(90); -> Servo auf Ausgang

    Serial.println("Servo fährt zu Ausgangsposition"); -> Log schreiben

  }

  if (digitalRead(TASTER2\_PIN) == HIGH) { -> schaut ob gedrückt

    Serial.println("Taster 2 gedrückt - Servo fährt rückwärts"); -> Log schreiben

    servo.write(0); -> setze servo auf 0°

    delay(1000); -> Warte 1 Sekunde

    servo.write(90); -> Servo auf Ausgang

    Serial.println("Servo fährt zu Ausgangsposition"); -> Log schreiben

  }

}

LED blinken lassen

int ledPin = 12; // definiert die Pin-Nummer, an der die LED angeschlossen ist

void setup() {

  pinMode(ledPin, OUTPUT); // legt den Pin als Ausgang fest

}

void loop() {

  digitalWrite(ledPin, HIGH); // schaltet die LED an

  delay(1000); // wartet für 1 Sekunde

  digitalWrite(ledPin, LOW); // schaltet die LED aus

  delay(1000); // wartet für 1 Sekunde

}

Mit Knopf:

const int buttonPin = 11; // definiert die Pin-Nummer, an der der Knopf angeschlossen ist

const int ledPin = 12; // definiert die Pin-Nummer, an der die LED angeschlossen ist

void setup() {

  pinMode(buttonPin, INPUT); // legt den Knopf-Pin als Eingang fest

  pinMode(ledPin, OUTPUT); // legt den LED-Pin als Ausgang fest

}

void loop() {

  int buttonState = digitalRead(buttonPin); // liest den aktuellen Zustand des Knopfes

  if (buttonState == HIGH) { // prüft, ob der Knopf gedrückt wurde

    digitalWrite(ledPin, HIGH); // schaltet die LED an

    delay(1000); // wartet für 1 Sekunde

    digitalWrite(ledPin, LOW); // schaltet die LED aus

    delay(1000); // wartet für 1 Sekunde

  } else {

    digitalWrite(ledPin, LOW); // schaltet die LED aus, wenn der Knopf nicht gedrückt wurde

  }

}

Die Relais Schalten

const int buttonPin = 11; // definiert die Pin-Nummer, an der der Knopf angeschlossen ist

const int relaisPin = 12; // definiert die Pin-Nummer, an der die Relais angeschlossen ist

void setup() {

  pinMode(buttonPin, INPUT); // legt den Knopf-Pin als Eingang fest

  pinMode(relaisPin, OUTPUT); // legt den LED-Pin als Ausgang fest

}

void loop() {

  int buttonState = digitalRead(buttonPin); // liest den aktuellen Zustand des Knopfes

  if (buttonState == HIGH) { // prüft, ob der Knopf gedrückt wurde

    digitalWrite(relaisPin, HIGH); // schaltet die LED an

  } else {

    digitalWrite(relaisPin, LOW); // schaltet die LED aus, wenn der Knopf nicht gedrückt wurde

  }

}