جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم

قسم الانظمة الطبية الذكية

Intelligent Medical Systems Department

Lap(5)

LCD & Sensors III

Subject: Embedded System

Class: Third

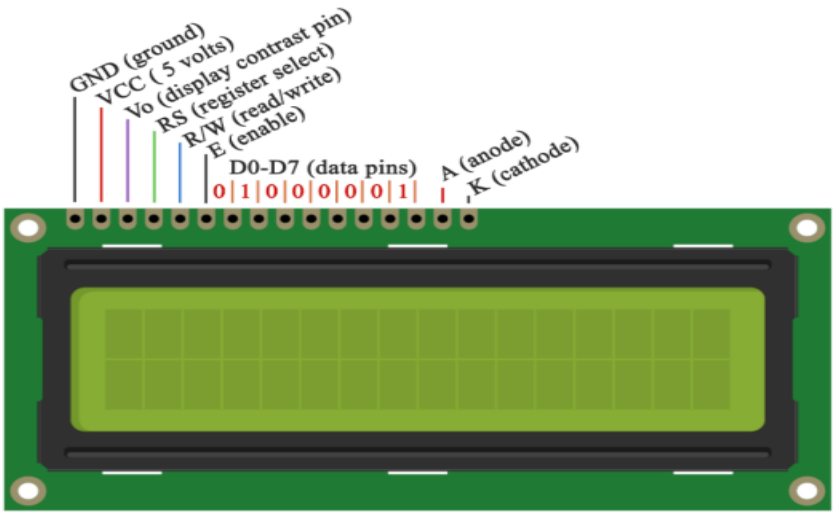
Lecturer: Prof.Dr. Mehdi Ebady Manaa

Programmer:- Fatima Hussein Jawad

Study Year: 2024-2025

LCD(Liquid Crystal Display):

A display screen used to display text, numbers, and symbols. This type of screen is the most common and most popular among the electronics, due to its low availability and ease of programming. It has a backlight and is available in different sizes and colors. There is 16x2, which are the most common months, 20x2, 20x4, and so on. We will use the screen 16 * 2, which means 16 columns with 2 rows. Its communication system is in parallel.



RS (Registrar Selection): This pin determined in which memory we write data.

R/W(Read/Write): This option determines whether to read or write.

E(enable): if enabled, can be written to the LCD monitor.

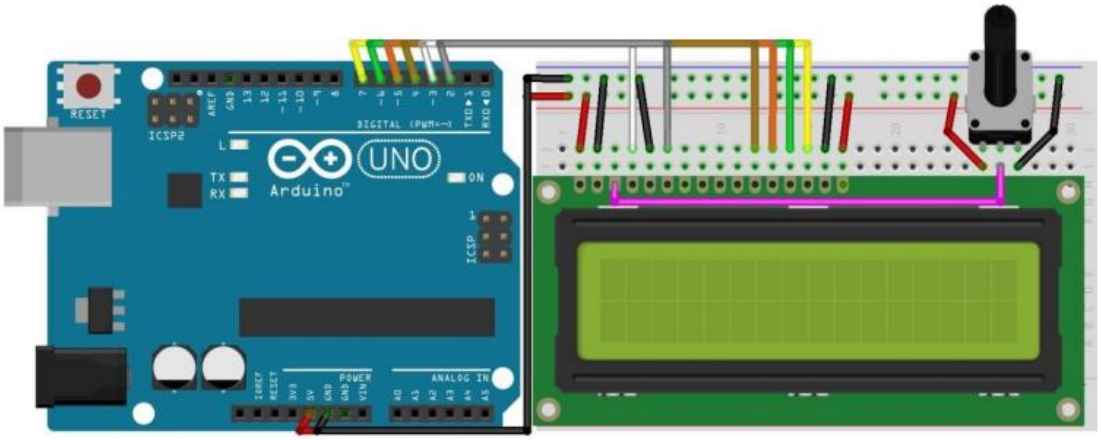
D0-D7(data pin): for data transition.

There are also pin for voltage, grounding and internal backlighting.

Example (Display Something)

Requirements: Arduino, BreadBoard, Resistor, LCD, wire.

Connection map:



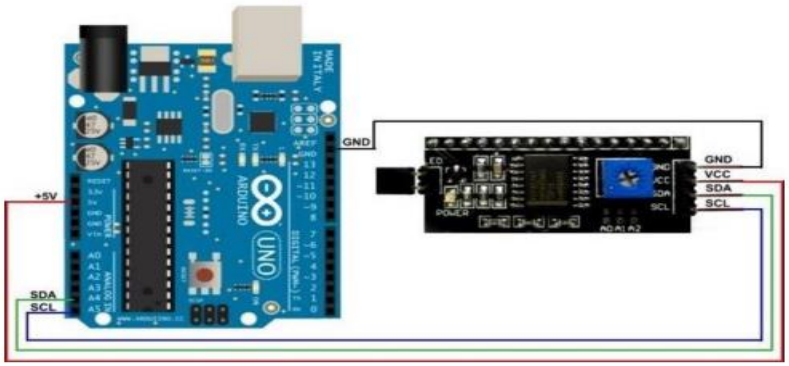
**(first must add (LiquidCrystal) librariy - can download it from github)

Code

```
#include <LiquidCrystal.h>
// LiquidCrystal lcd(RS, E, D4, D5, D6, D7);
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

void setup() {
  lcd.begin(16, 2);  // Initialize the LCD with 16 columns and 2 rows
}

void loop() {
  lcd.setCursor(7, 1);  // Set cursor to column 7, row 1 (second row)
  lcd.print("Hello World!");  // Print "Hello World!"
  delay(4000);  // Wait for 4 seconds (4000 milliseconds)
  lcd.clear();  // Clear the LCD screen

  lcd.setCursor(1, 1);  // Set cursor to column 1, row 1 (second row)
  lcd.print("Embedded systems");  // Print " Embedded systems"
  delay(4000);  // Wait for 4 seconds before repeating }}
```

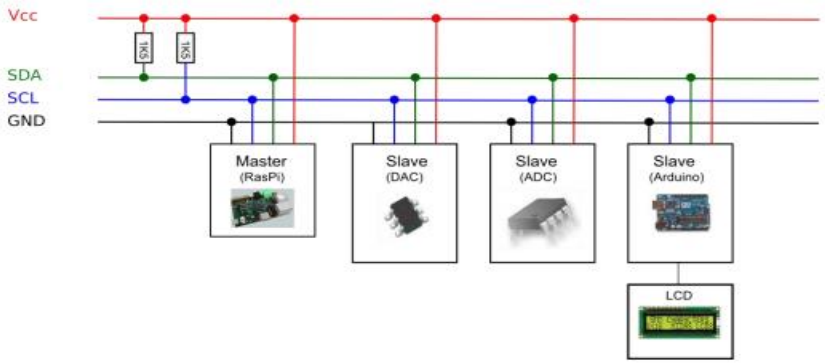
I2C:

The I2C communication bus is very popular and broadly used by many electronic devices because it can be easily implemented in many electronic designs which require communication between a master and multiple slave devices or even multiple master devices. The easy implementations comes with the fact that only two wires are required for communication between up to almost 128 (112) devices when using 7 bits addressing and up to almost 1024 (1008) devices when using 10 bits addressing.



How it work:

The two wires, or lines are called Serial Clock (or SCL) and Serial Data (or SDA). The **SCL** line is the clock signal which synchronizes the data transfer between the devices on the I2C bus and it's generated by the master device. The other line is the **SDA** line which carries the data.

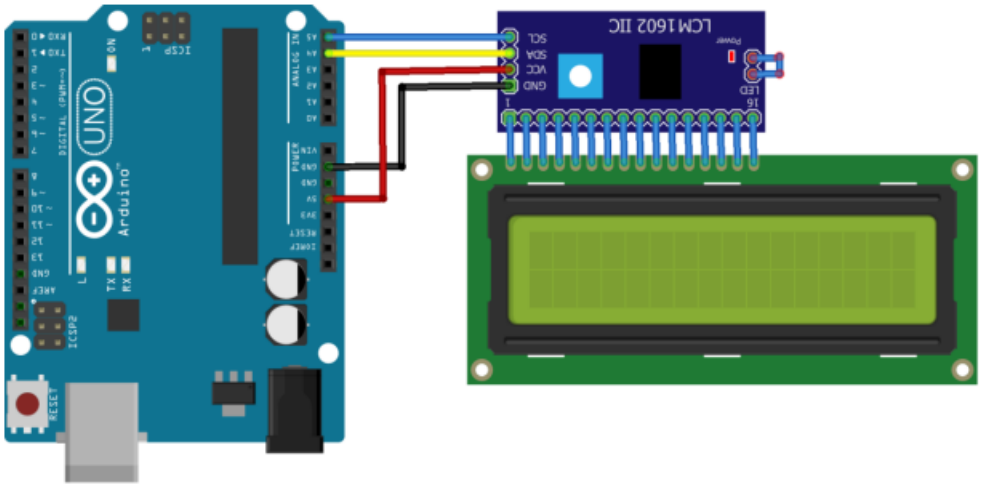


Example

(i2c with lcd)

Requirements: Arduino, LCD, i2c, wire.

Connection map:

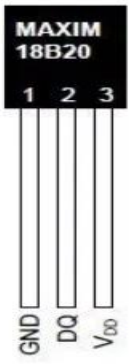


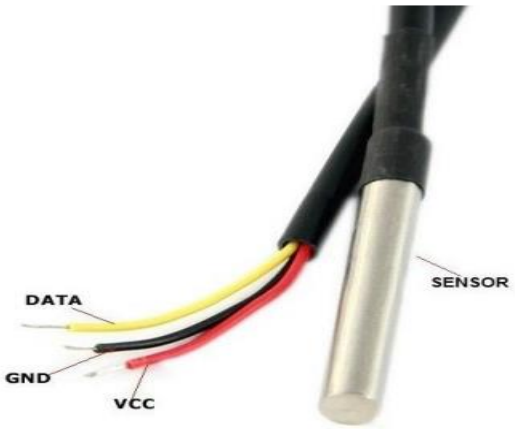
Code

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

LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup() {
  lcd.begin(16, 2); // Initialize the LCD with the specified dimensions
  lcd.backlight(); // Turn on the LCD backlight
}

void loop() {
  lcd.setCursor(0, 0);
  lcd.print("Embedded systems");
  delay(8000);
  lcd.clear();
  lcd.setCursor(0, 0); // Set cursor to the first row
  lcd.print("Lec6");
  delay(8000); // Add delay to allow second message to display
}
```

Waterproof (DS18b20) sensor:

The DS18b20 is a digital sensor that operates through a single wire and provides accurate temperature readings. It also has water resistance and the reading is not affected by the long distance between the sensor and the control board.



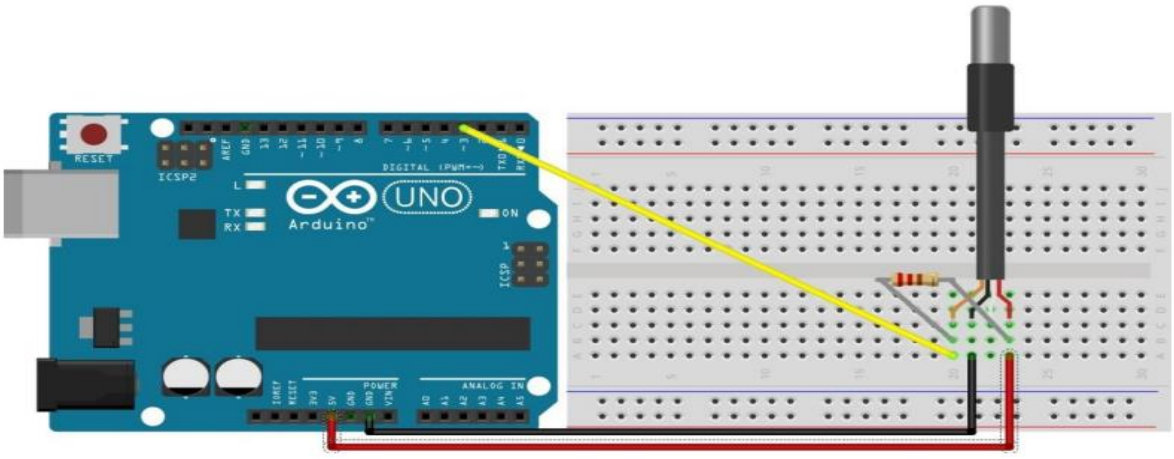


Example :

(Measurement of water temperature)

Requirements: Arduino, BreadBoard, Resistor, waterproof Sensor, wires.

Connection map:



**(first must add OneWire & DallasTemperature libraries - can download it from github)

Code:

```
#include <DallasTemperature.h>
#include <OneWire.h>
int data = 3;
OneWire oneWire(data);
DallasTemperature sensors(&oneWire);
float Celcius=0;
float Fahrenheit=0;
void setup(void)
{
 Serial.begin(9600);
 sensors.begin();
}
void loop(void)
{
 sensors.requestTemperatures();
 Celcius=sensors.getTempCByIndex(0);
 Fahrenheit=sensors.toFahrenheit(Celcius);
 Serial.print(Celcius);
 Serial.println(" C ");
 Serial.print(Fahrenheit);
 Serial.println(" F ");
 Serial.println("-----------------");
 delay(1000);
}
```

MQ sensor:

is a gas sensor. There are different types of it such as MQ2 for Combustible Gas and Smoke, MQ3 for Alcohol Vapor, MQ5 for LPG and Natural Gas and Town Gas, MQ9 for Carbon Monoxide and Coal Gas and Liquefied Gas, and so on… We will use the MQ2 type.

How does it Work:

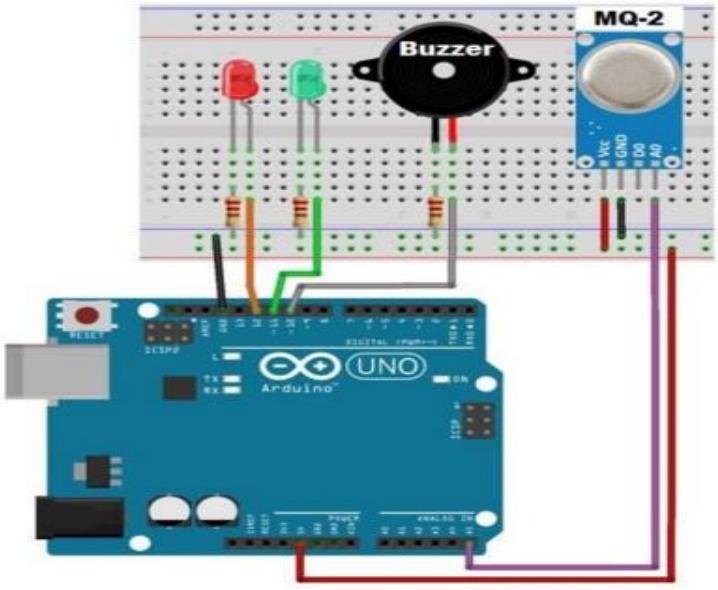
Use a small heater inside with an electro-chemical sensor. They are sensitive for a range of gasses and are used indoors at room temperature. The output is an analog signal and can be read with an analog input of the Arduino.

Example:

(Measurement of qaz)

Requirements :Arduino, BreadBoard, Resistor, LDR Sensor, Led, wires.

Connection map:



code

```
int redLed = 12;
int greenLed = 11;
int buzzer = 10;
int smoke = A5;
void setup() {
 pinMode(redLed, OUTPUT);
 pinMode(greenLed, OUTPUT);
 pinMode(buzzer, OUTPUT);
 pinMode(smoke, INPUT);
 Serial.begin(9600);
}
void loop() {
 int analogSensor = analogRead(smoke);
 Serial.print("Qaz: ");
 Serial.println(analogSensor);
 if (analogSensor > 120)
 {
 digitalWrite(redLed, HIGH);
```

```
 digitalWrite(greenLed, LOW);
 tone(buzzer, 1000);
 }
 else
 {
 digitalWrite(redLed, LOW);
 digitalWrite(greenLed, HIGH);
 noTone(buzzer);
 }
 delay(100);
}
```