CA326 AEP (Advanced Embedded Programming)
Lecture Notes:

Chapter-3 (Sensors and Actuators)

Sensor and Its Types

"A sensor is a device which reads the physical environment or environmental variable and tell you what happened." (temperature, pressure, humidity)

"A sensor is an object whose purpose is to detect change in environmental parameters and provide corresponding output."

"A sensor is a device which reads environmental parameters and convert it into electrical form which can be read by other devices."

Sensor provides input to embedded system and based on that input information system will make decision.

We need a sensor to feed data in a system and tells the system when to take a specific action.

Let's take an example of air condition system.

- You have set temperature to 18 degree.
- Initially the ac was off.
- So fan starts pumping the air until desired temperature is reached.
- How the ac system knows that desired temperature is reached?
- You must require a device which sense the temperature which is temperature sensor device.
- Based on that system will take a decision to operate or not.

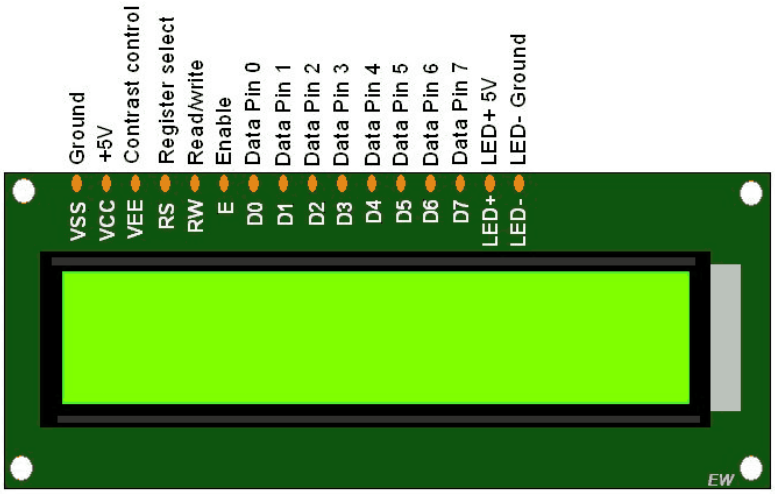
Types of sensors:

1. Temperature sensor.
2. Humidity sensor
3. Pressure sensor
4. Ultrasonic sensor
5. Motion sensor
6. IR (Infrared) Sensor
7. Optical sensor (photo detector)
   1. LED: Light Emitting Diode
   2. LASER: Light Amplification Through Stimulated Emission of Radiation
8. Sound sensor
9. Water sensor
10. Gas sensor
11. Line follower sensor
12. Wireless sensor (RFID)

13. Color sensor
14. Light intensity sensor
15. Barcode reader
16. Heart rate pulse sensor
17. Alcohol sensor
18. Fingerprint scanner
19. Flame sensor
20. Specific gas sensor (Mithane)

LCD pin configuration and interface with Arduino.

LCD: Liquid Crystal Display



(16 X 2) LCD Display

- It is used to display characters and numbers on its screen.
- (16 x 2) = 32 characters can be displayed on LCD.
- 16 columns (Vertical Line) and 2 rows (horizontal Line)

H (0,0)	E (1,0)	L (2,0)	L (3,0)	O		W	O	R	L	D	!	!			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

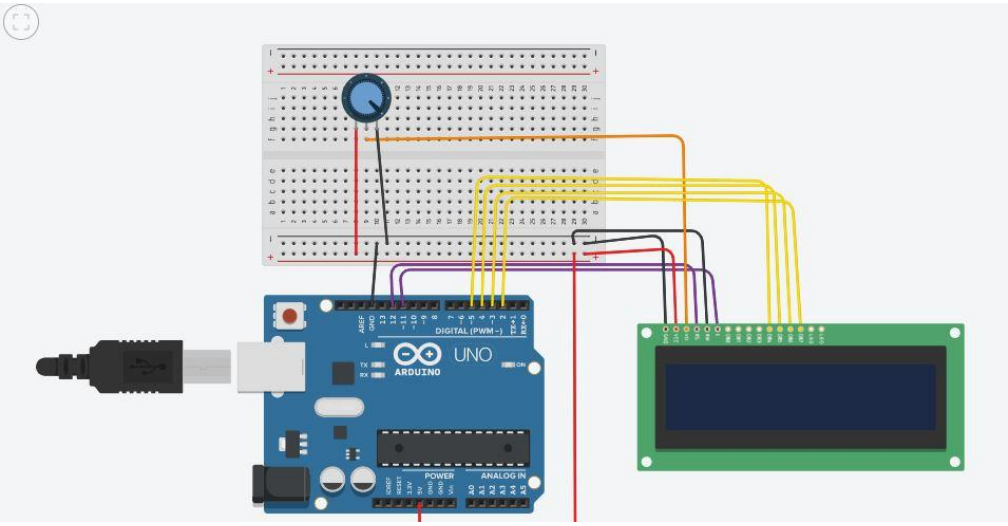
- It can be found in digital clock, washing machine, Microwave oven, Freeze, LCD Television and many more places.
- LCDs are used in embedded system for displaying some parameters or status information.
- (16 x 2) LCD display is having 16 Pins.
- It is having 2 rows. In each row 16 characters can be displayed.

Lcd.setCursor(3,0) will set cursor to **First** row and **Fourth** column.
Lcd.setCursor(13,1) will set cursor to **Second** row and **Fourteenth** column.

LCD Pin Configuration:

Pin NUmber	Name of The Pin	Function of the PIN
Pin 1	VSS	It is Ground pin of LCD display and it must be connected with ground pin of arduino.
Pin 2	VCC	It is connected with +5 V. It must be connected with +5 of arduino board.
Pin 3	VEE	It must be connected with potentiometer device which is used to vary the brightness / contrast of the display. Potentiometer is a device which is use to vary the resistance. It is having three terminals. First is connected with +5, second is connected with ground and third is connected with arduino pin and VEE.
Pin 4	RS	Register Select. There are two different register used with LCD. 1. Data Register 2. Instruction(command) Register RS pin is used to select any one of the above register. RS→1→+5v→ activate or use data register. RS→0→ground → activate or use Instruction register. It must be connected with digital pin of arduino.
Pin 5	RW	Read /Write It is used to specify whether you want to read or write on LCD display. It must be connected with digital pin of arduino. R/W→1→+5v→Read R/W→0→Ground→Write
Pin 6	E	E (Enable) is used to enable the LCD display for displaying data. E=1→enable the LCD E=0→disable the LCD It must be connected with any digital pin of arduino.
Pin 7	D0	Data Pin – 0 is used to send the data and it must be connected with any digital pin of arduino.

Pin 8	D1	Data Pin – 1 is used to send the data and it must be connected with any digital pin of arduino.
Pin 9	D2	Data Pin – 2 is used to send the data and it must be connected with any digital pin of arduino.
Pin 10	D3	Data Pin – 3 is used to send the data and it must be connected with any digital pin of arduino.
Pin 11	D4	Data Pin – 4 is used to send the data and it must be connected with any digital pin of arduino.
Pin 12	D5	Data Pin – 5 is used to send the data and it must be connected with any digital pin of arduino.
Pin 13	D6	Data Pin – 6 is used to send the data and it must be connected with any digital pin of arduino.
Pin 14	D7	Data Pin – 7 is used to send the data and it must be connected with any digital pin of arduino.
Pin 15	LED +5v	Used to enable LED Light and must be connected with +5 v.
Pin 16	LED gnd	It must be connected with Ground



List out various pin configurations for LCD.

1. 6 pin configuration→ LiquidCrystal lcd(rs, en, d4, d5, d6, d7)
2. 7 pin configuration→ LiquidCrystal lcd(rs, rw, en, d4, d5, d6, d7)
3. 10 pin configuration→ LiquidCrystal lcd(rs, en, d0, d1, d2,d3 d4, d5, d6, d7)
4. 11 pin configuration→ LiquidCrystal lcd(rs, rw, en, d0, d1, d2,d3 d4, d5, d6, d7)

Code

```
#include <LiquidCrystal.h>
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int thisChar = 97;
void setup()
{
lcd.begin(16, 2);
```

```
lcd.cursor();
}
void loop()
{
// print the character
lcd.write(thisChar);
delay(1000);
thisChar++;
// reverse directions at 'm':
if (thisChar == 'm') {
// go right for the next letter
lcd.rightToLeft();
}
// reverse again at 's':
if (thisChar == 's') {
// go left for the next letter
lcd.leftToRight();
}
// reset at 'z':
if (thisChar > 'z') {
// go to (0,0):
lcd.home();
lcd.clear();
// start again at 0
thisChar = 'a';
}
}
```

List out various functions / methods of LCD.

1. lcd.begin()
2. lcd.cursor()
3. lcd.clear()
4. lcd.home()
5. lcd.rightToLeft()
6. lcd.leftToRight()
7. lcd.scrolldisplayLeft()
8. lcd.scrolldisplayRight()

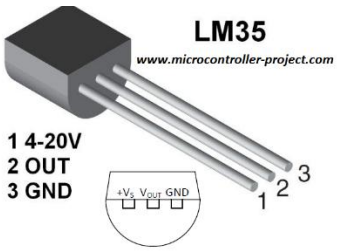
Uses of Temperature Sensor and LM35 sensor.

Temperature sensor:

- It detects the environmental temperature.
- Various types of temperature sensors are
  - Resistance Temperature Detectors (RTD)
  - Thermocouples
  - Thermistor
- LM35 is popular temperature sensor.

- TMP36 is also well known temperature sensor.

LM35 sensor:



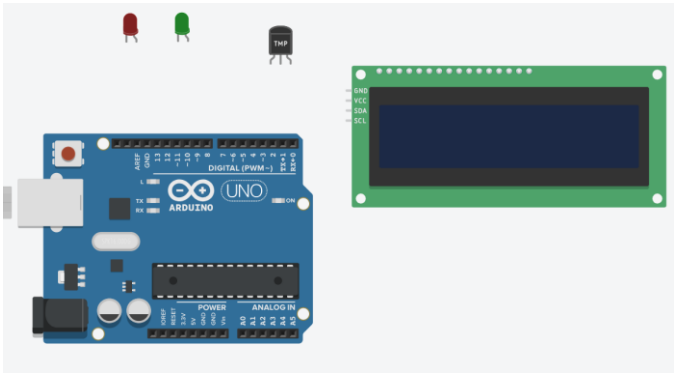
PIN 1 : Voltage Pin (range is 4 v to 20v and it is connected with vcc of arduino controller)

PIN 2: Output Pin (it is connected with A0 analog pin of arduino)

PIN 3 : Ground pin (connected with ground of arduino)

- LM35 reads the temperature from environment and generate the output electrical signal or voltage which is proportional to temperature.
- It can able to detect temperature from -55 degree to +150 degree.
- It operates between 4 to 20 v.
- It is having low cost with greater accuracy.
- Coating around LM35 protects it from self-heating.
- Lm35 has many family members a few names are LM35C, LM35CA, LM35D, LM135, LM135A, LM235, LM335.

Write steps (Algorithm) to interface temperature sensor with arduino and I2C LCD. Display the temperature value on LCD as well as Serial Monitor. If the temperature is > 40 then make red LED ON else, make green LED ON.



1. Declare LCD Library.
2. Initialize LCD.
3. Initialize Serial Monitor.
4. Connect temperature sensor with analog pin.
5. Read the value of temperature.
6. Convert the temperature into Celsius and Fahrenheit (in given range).

7. Display the temperature on serial monitor
8. Display the temperature on LCD.
9. Define threshold = 40 degree.
10. Compare the temperature with threshold.
11. If temperature > threshold value, then make RED led ON.
12. If temperature < threshold value, then make GREEN led ON.

Interface Ultrasonic sensor with Arduino.

Ultra: It refer to sound which cannot be listen by human but animals like bat and dog can hear it.

Sonic: It refers to sound

HC-SR04

- Generally human use the eye for finding the distance of an object which is not an accurate method. Birds like bat can emit the sound signal and receives it back and find the distance. If we estimate the time between sending and receiving the signal then we can able to find distance.
- Ultrasonic sensor can be used in many applications where we need to find the distance of an object, detecting motion, counting objects etc.
- Ultrasonic sensor must have two parts
    - A transmitter which transmits the signal which human cannot hear.
    - A Receiver which receives the reflected signal bounced back from nearby objects.
- If the object is very close to the sensor, the signal comes back quickly.
- If the object is far away from the sensor, the signal takes longer to come back
- If objects are too far away from the sensor, the signal takes so long to come back (or is very weak when it comes back) that the receiver cannot detect it.
- It is having four pins to interface with arduino controller.
    - **Vcc: +5v (power pin)**
    - **Gnd: Ground pin**
    - **Trig: Trigger pin to send the ultrasonic signal**
    - **Echo: Echo pin to receive the reflected signal**
- Total time required by the signal to move forward and reverse back is T.
- The time to recei
- ve back from object is (T/2).

Steps: (Algorithm)

1. Connect Trig pin and Echo pin with arduino.
2. Connect Red and Green LED with Arduino Pins
3. Initialize serial monitor.
4. Initialize LCD.
5. Create a pulse (low-high-low) signal on "Trig pin" of ultrasonic sensor.
6. Wait for "Echo pin" to become "HIGH".
7. When "Echo pin" is high then find the duration using pulseIn() function.
8. Calculate distance from duration value using the equation distance (cm) = 0.0172 x Duration (Microseconds).
9. Display the distance on LCD.
10. Display the distance on serial monitor.
11. Define threshold value = 50 cm
12. Compare distance with threshold value.
13. If the distance is higher then ON red led.
14. If the distance id lower then ON green led.

Prove that for ultrasonic sensor Distance (cm) = 0.0172 x T cm

We have **Velocity = distance /time**

Sound velocity is 343 m/s = 34300 cm/s.(1m = 100 cm.)

Now convert cm/s into cm/microsecond.
Micro = 10^{-6}

34300 x 10^{-6} cm/microsecond.

34300 / 1000000 =0.0343 cm / microsecond.

Velocity = distance /time
Distance = velocity x time
Distance = velocity x (T/2)
Distance = 0.0343 x T/2

Distance (cm) = 0.0172 x T cm (Proved)
Distance (inch) = distance(cm) / 2.54

Interface keypad with Arduino

A keypad is a device that can be used to receive input data from user. In this project, you will first learn how to interface a (4×4) keypad with Arduino, and then you will Build a simple calculator using a keypad.

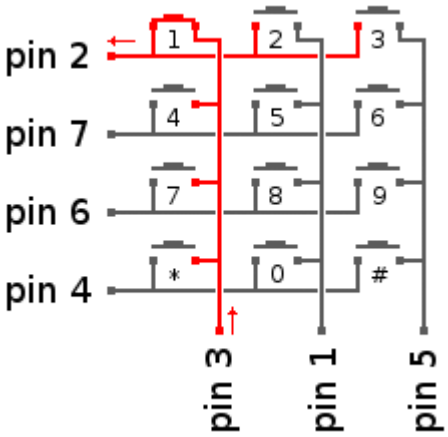
Different Kind of Keypads

Keypads are made in various types, but the most common sizes are (5 * 4) and (4 * 4).

These numbers indicate the number of rows and columns of the keypads. For example, A (3*4) keypad has 4 rows and 3 columns.

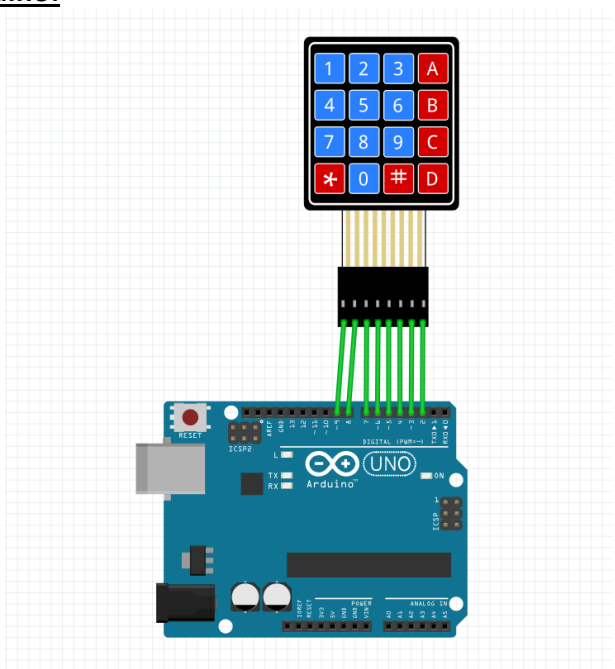
How Does A Keypad Work?

To receive data from 12 buttons, we need to use 12 digital pins from our microcontroller (Arduino), and this means wasting a lot of pins, but using the keypad, we only need 7 digital pins. You can see how the keypad works in the animation below.



Keypad pins are divided into two groups: row and column. you can see 2 examples of naming them in the image below.

Interfacing with Arduino:



Steps: (Algorithm)

1. Install keypad library. (Keypad.h)
2. Declare object for keypad. (first identify rows and columns of keypad, list the arduino pins where rows and columns are connected and declare Keymap)
3. Declare object for LCD.
4. Declare object for Servomotor.
5. Declare password.
6. Initialize LCD.
7. Initialize serial monitor.
8. Check if key is presses or not. If any key is pressed then store the value in "Key" variable which is an array.
9. Compare "Key" and "password" variable.
10. If "key" = "Password" then Rotate servomotor for 180 degree for 10 second.
11. Display key on LCD.
12. Display key on serial monitor.

UART and its working

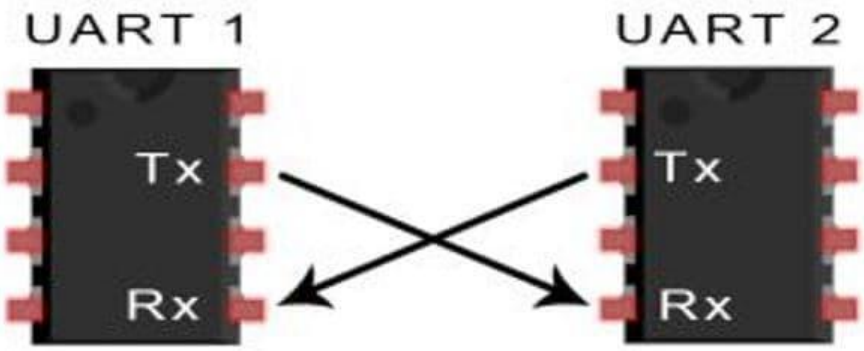
What is UART?

- It stands for **Universal Asynchronous Reception and Transmission (UART)**
- A simple serial communication protocol that allows the host communicates with the auxiliary device.
- UART supports bi-directional, asynchronous and serial data transmission.
- It has two data lines, one to transmit (TX) and another to receive (RX), which are used to communicate through digital pin 0, digital pin 1.

- TX and RX are connected between two devices. (eg. USB and computer)

How does it work?

1. It can operate between devices in 3 ways:
    - Simplex = data transmission in one direction
    - Half-duplex = data transmission in either direction but not simultaneously
    - Full-duplex = data transmission in both directions simultaneously
- Once connected, data flows from TX of the transmitting UART to RX of the receiving UART.
- As UART is an asynchronous serial transmission protocol = No clocks
- Transmitting UART converts parallel data from the master device (eg. CPU) into serial form and transmit in serial to receiving UART. It will then convert the serial data back into parallel data for the receiving device

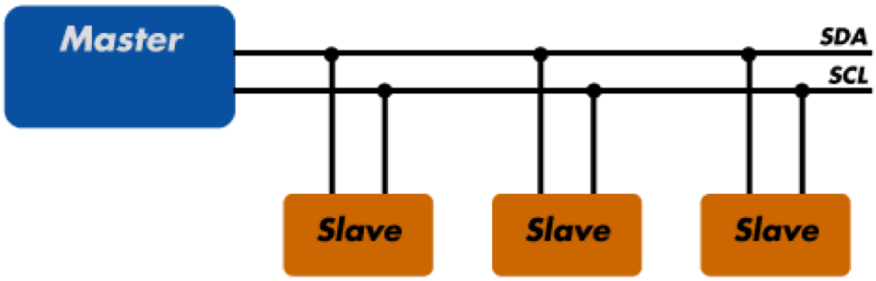


UART data transmission speed is referred to as BAUD Rate and is set to 9600 or 115,200 by default (BAUD rate is based on symbol transmission rate, but is similar to bit rate).

What is I2C ? How does it work? Explain interfacimg between I2C LCD and Arduino

What is I2C?

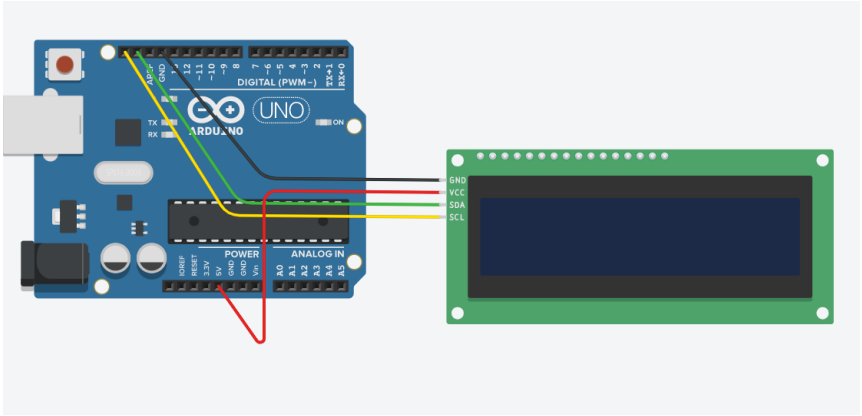
- It stands for Inter-integrated-circuit (I2C)
- It is a serial communications protocol similarly to UART. It is used with modules and sensors.
- It is a simple, bidirectional two-wire synchronous serial bus and requires only two wires to transmit information between devices connected to the bus.
- They are useful for projects that require many different parts (eg. sensors, pin, expansions and drivers) working together as they can connect up to 128 devices to the mainboard while maintaining a clear communication pathway!



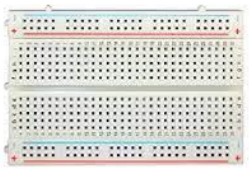
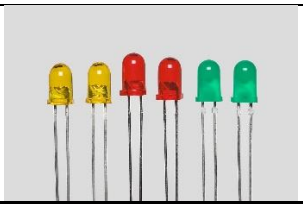
How does it work?

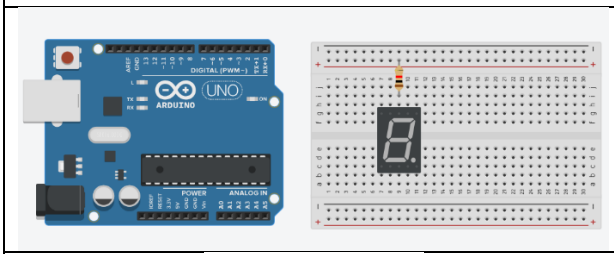
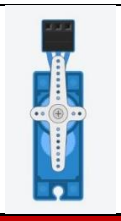
- It has 2 Lines which are SCL (serial clock line) and SDA (serial data line acceptance port)
- SCL is the clock line for synchronizing transmission.
- SDA is the data line through which bits of data are sent or received.
- The master device initiates the bus transfer of data and generates a clock to open the transferred device and any addressed device is considered a slave device.
- If the master wants to send data to the slave, the master must first address the slave before sending any data.
- If the master wants to receive data from the slave, the master must again address the slave first.

I2C LCD Interfaced with Arduino:



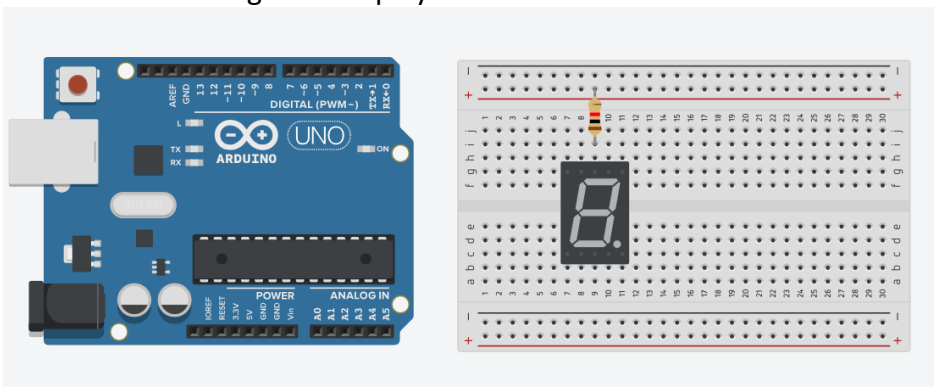
Que 9: Identify following devices.

	**Breadboard**
	**Integrated circuit**
	**LEDs**

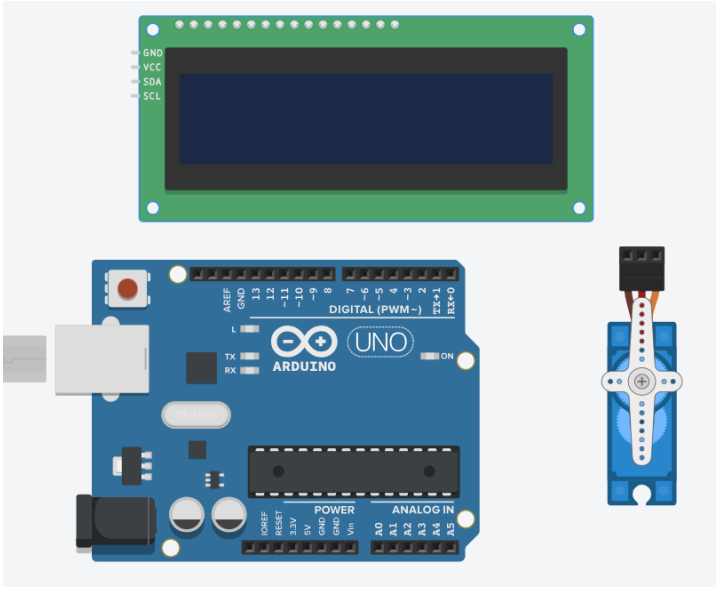
	**Fuse**
	**Resistors**
	**Pushbutton switches**
	**Arduino and 7 segment display**
	**Ultrasonic sensor**
	**Servomotor**

Interface following components with arduino.

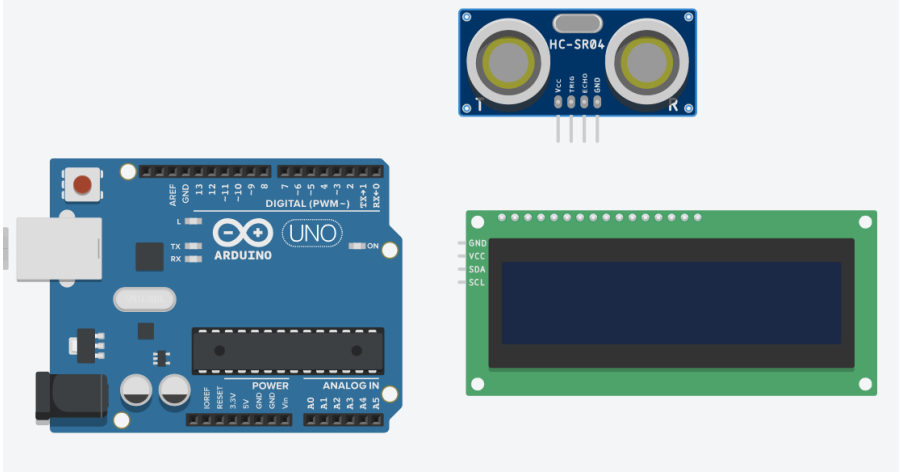
1.Arduino and 7 segment display



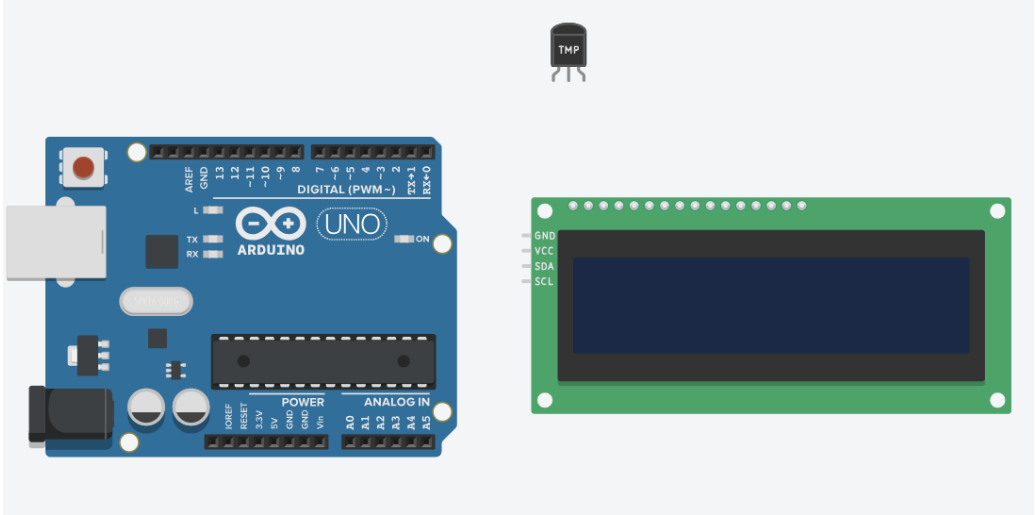
2.Arduino and I2C LCD and servomotor



3. Arduino and I2C LCD and Ultrasonic sensor



4. Arduino and I2C LCD and Temperature sensor



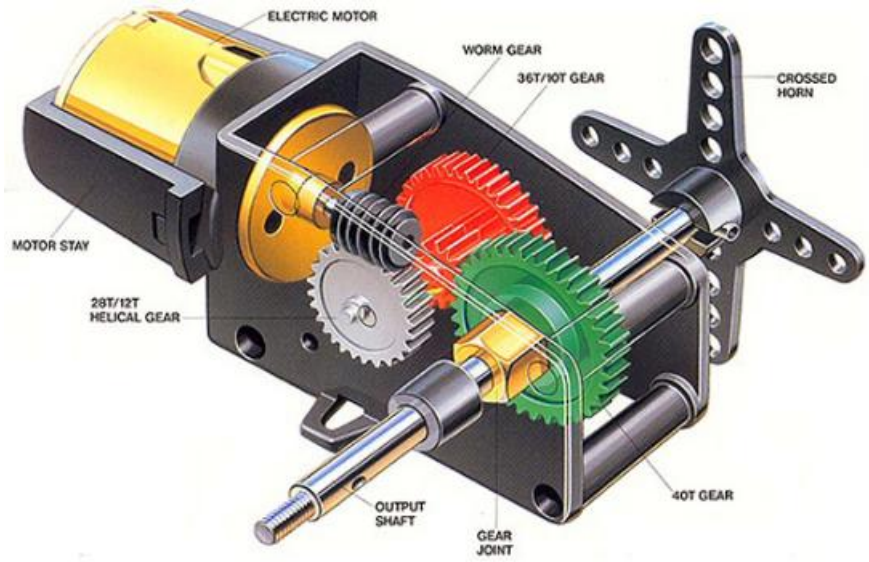
Explain DC Motor interfacing using L293D Motor Driver IC

- L293D is 16 pin IC.
- It is used to connect two DC Motors with arduino.
- Left side pins 1 to 8 are used for motor 1.
- Right side pins 9 to 16 are used for motor 2.

Pin Number	Pin Name	Pin Description
PIN 1	Enable 1,2	• Pin 1 is known as the enable pin. • If HIGH voltage I given to Pin 1 then it enables motor 1. • When enable pin is high then all input and output pins (2,3,6,7) are ready to work.
PIN 2	Input 1	• It is a pin which reads the input from arduino or any other controller.
PIN 3	Output 1	• It is a Pin which writes the data or output to DC motor input • It must be connected with DC Motor.
PIN 4	Ground	• The ground pin will be attached to the ground of the circuit or arduino.
PIN 5	Ground	• The ground pin will be attached to the ground of the circuit or arduino.
PIN 6	Output 2	• It is a Pin which writes the data or output to DC motor input • It must be connected with DC Motor.
PIN 7	Input 2	• It is a pin which reads the input from arduino or any other controller.
PIN 8	Vcc	• It must be connected with +5v to +16 v
PIN 9	Enable 3,4	• Pin 9 is known as the enable pin. • If HIGH voltage I given to Pin 9 then it enables motor 2. • When enable pin is high then all input and output pins (10,11,14,15) are ready to work.

PIN 10	Input 3	• It is a pin which reads the input from arduino or any other controller.
PIN 11	Output 3	• It is a Pin which writes the data or output to DC motor input • It must be connected with DC Motor.
PIN 12	Ground	• The ground pin will be attached to the ground of the circuit or arduino.
PIN 13	Ground	• The ground pin will be attached to the ground of the circuit or arduino.
PIN 14	Output 4	• It is a Pin which writes the data or output to DC motor input • It must be connected with DC Motor.
PIN 15	Input 4	• It is a pin which reads the input from arduino or any other controller.
PIN 16	Vss	• It must be connected with +5v to +16 v

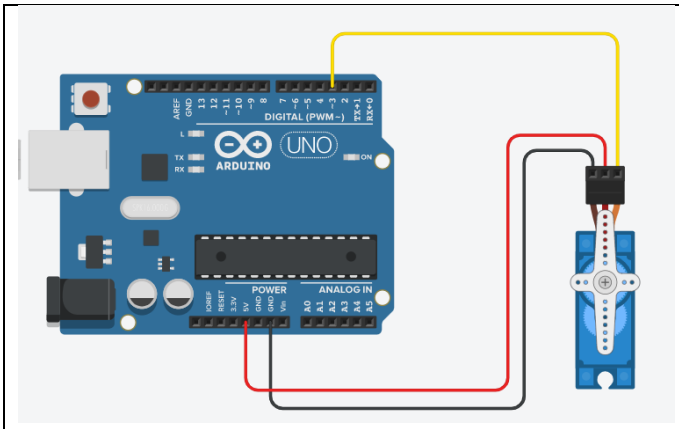
Interface servomotor with Arduino and write steps / algorithm to rotate the servomotor from 0 to 180 degree. Once it touch to 180 then reverse the rotation back to 0 degree.



Four main parts of servomotor

1. Dc motor
2. Gear Mechanism (4 gears)
3. Shaft with Horn
4. Position sensor

Servomotor works on principle of PWM technique.



Servomotor is having three connections

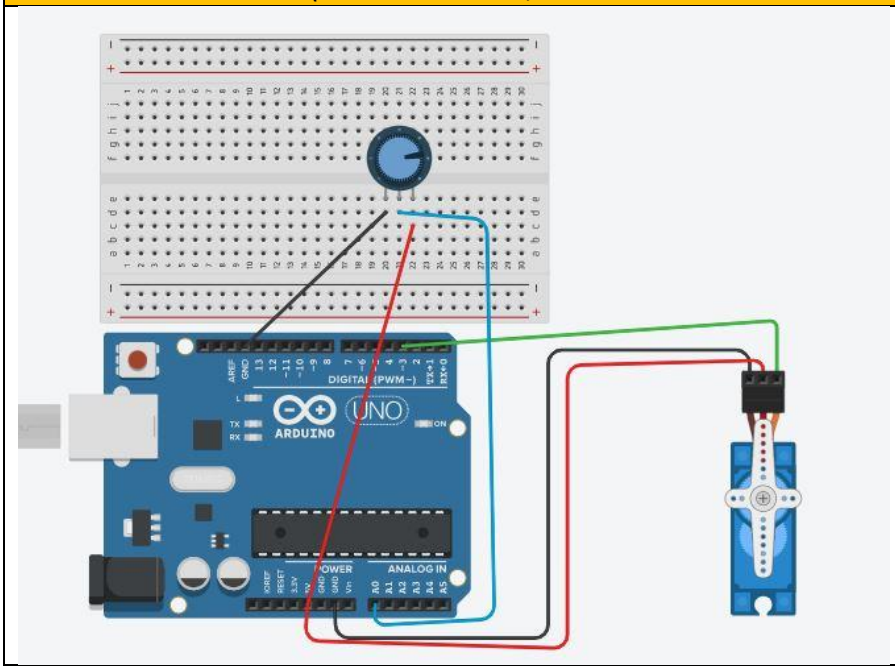
1. Power
2. Ground
3. Signal

Signal pin can be connected to any of the PWM pins. (3,5,6,9,10,11)

Steps:

1. Interface servomotor with arduino.
2. Use power, ground and signal pins.
3. Include servomotor library.
4. Declare object for servomotor.
5. Initialize serial monitor.
6. Set a variable **position** and initialize its value to 0.
7. Write the value 0 to 180 on servomotor using any looping method.
8. Once it reaches to 180 rotate the servomotor back from 180 to 0 using another loop.
9. Write the position value on serial monitor.

Interface potentiometer and servomotor with arduino. Rotate the knob of potentiometer to rotate servomotor. (Knob=minimum, rotation = 0 and Knob=maximum, rotation=180)



Steps:

1. Interface servomotor with arduino using ground, power and signal pins.
2. Interface potentiometer with arduino using ground, power and signal pins.
3. Connect signal pin of servomotor to PWM pin. (3,5,6,9,10,11)
4. Connect signal pin of potentiometer to analog pin. (A0 to A5)
5. Declare library of servomotor.
6. Create an object of servomotor.
7. Initialize serial monitor.
8. Read the value of potentiometer from analog pin. (0 to 1023)
9. Convert / Map the above value into range of 0 to 180.
10. Write this value on servomotor.
11. Display the potentiometer value as well as servomotor degree value on serial monitor.