جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم

قسم الأنظمة الطبية الذكية

Intelligent Medical Systems Department

Lap (4)

Ultrasonic sensor

Subject: Embedded System

Class: Third

Lecturer: Prof.Dr. Mehdi Ebady Manaa

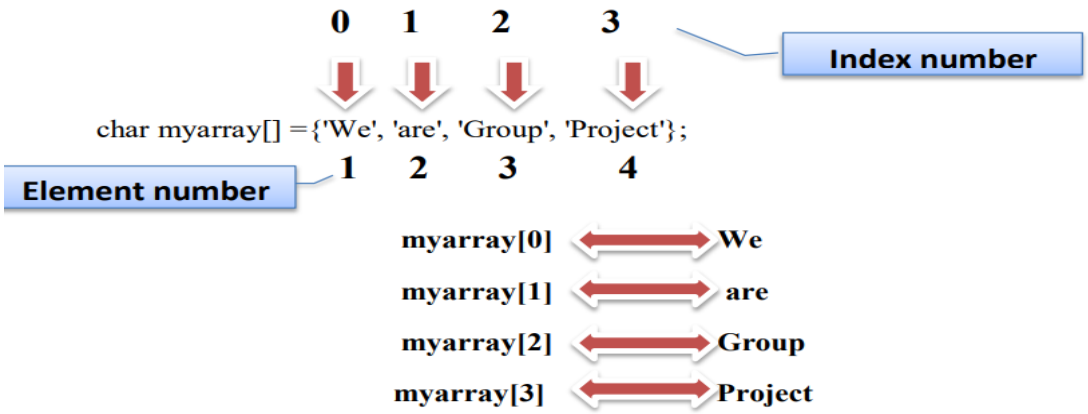
Programmer:- Fatima Hussein jawad

Study Year: 2024-2025

Array

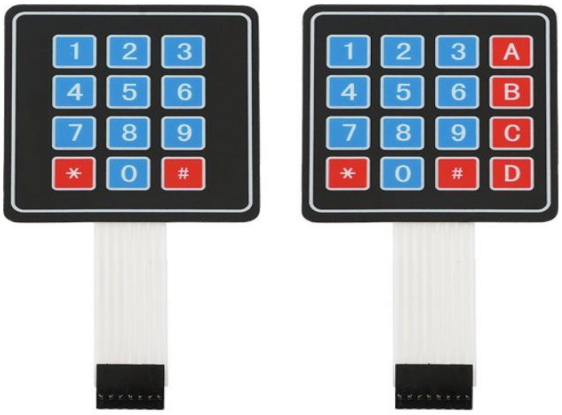
An array is reservoir for a value group, where each value can be accessed from its own index. The array must declare before it is used. First, specifies the data type, and then set any name for array, then putting square brackets (leaving the square brackets empty indicating that the length of the array is not specified but when determined, the length is placed between the square brackets). For the values of the array, it is determined either directly with the declared it or later determined within the program.

Example 1 :

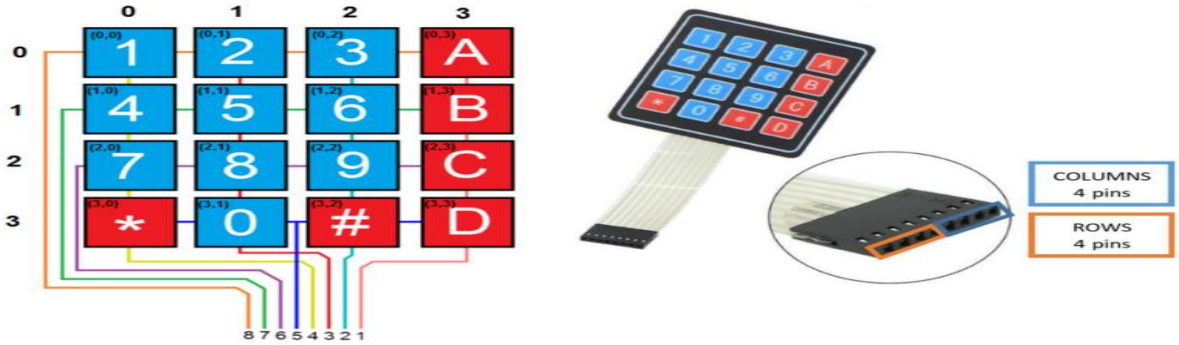


Keypad

is one of the most commonly used input devices. In a standard keypad wired as an X-Y switch matrix, normally-open switches connect a row to a column when pressed (where it is set of buttons arranged in a block). If a keypad has 12 keys, it is wired as 3 columns by 4 rows (4×3 array). A 16 key pad would have 4 columns by 4 rows (4×4 array).



How does it work?



1. **Matrix Design:** Keypads are built in a matrix of rows and columns. For example, a 4x4 keypad has 4 rows and 4 columns, resulting in 16 keys. Each key connects one row to one column.

2. **Scanning the Matrix:** A microcontroller (such as Arduino or other microcontrollers) regularly "scans" the rows and columns to detect if any key is pressed. It does this by setting each row high (voltage on) one by one, while the other rows remain low (voltage off), and then checking the columns.

3. **Detecting a Key Press:** When a key is pressed, it connects a specific row to a specific column, creating a closed circuit. The microcontroller detects which column is active during each row scan, so it knows the exact key that's been pressed by matching the row and column.

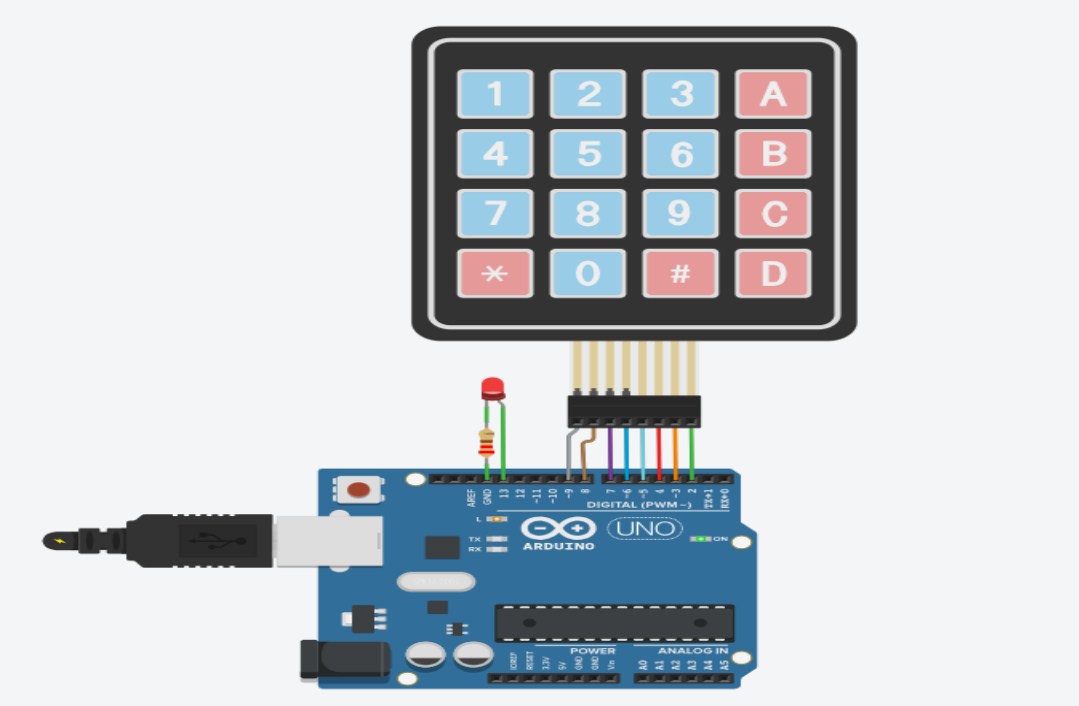
4. **Interpreting the Signal:** The microcontroller then interprets the pressed key based on the row and column it detects. For example, if row 2 and column 3 are connected, it could mean key "5" in a standard keypad layout.

5. **Software Processing:** Finally, the microcontroller's software can process the detected key press to perform actions, like entering a password, dialing a number, or other tasks depending on the application.

Example 2 (Password Luck) :

Requirements: Arduino, BreadBoard, Resistor, Led, Keypad, wire.

Connection map:



Code:

```
#include <Keypad.h>

#define ROWS 4
#define COLS 4
```

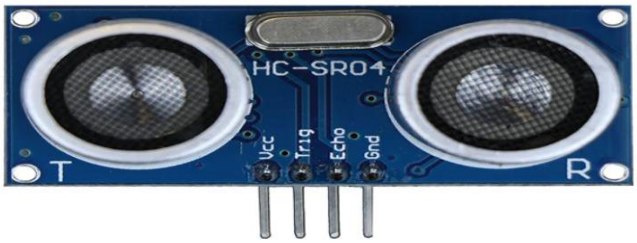
```
char mapKeys[ROWS][COLS] = {
  { '1', '2', '3', 'A' },
  { '4', '5', '6', 'B' },
  { '7', '8', '9', 'C' },
  { '*', '0', '#', 'D' }
};

byte rowPins[ROWS] = { 9, 8, 7, 6 };
byte colPins[COLS] = { 5, 4, 3, 2 };
Keypad kpd = Keypad(makeKeymap(mapKeys), rowPins, colPins, ROWS, COLS);

void setup() {
  Serial.begin(9600);
  pinMode(13, OUTPUT);
}

void loop() {
  char key = kpd.getKey();
  if (key) {
    Serial.println(key);
    if (key == '1')
      digitalWrite(13, 1);
    if (key == '0')
      digitalWrite(13, 0);
  }
}
```

Circle work online:

https://www.tinkercad.com/things/8meIXS7lALU/editel

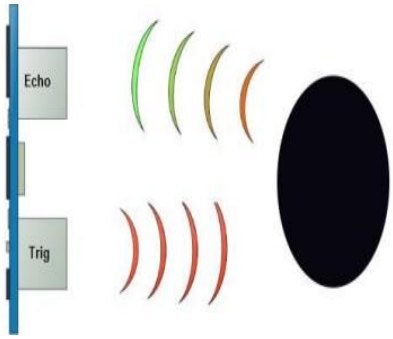
Ultrasonic sensor (HC-SR04) :

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. Work from 2cm to 400 cm or 1 to 13 feet. Its operation is not affected by sunlight or black material. It comes complete with ultrasonic transmitter and receiver module.



How does it Work:

a- The transmitter (trig pin) sends a signal: a high-frequency sound.
b- When the signal finds an object, it is reflected and…
c- The transmitter (echo pin) receives it.



The equation of distance is distance =duration* speed Here the speed is a speed of air 340µm in one second. But this sensor used cm, so we need to convert speed to cm to become 0.0340cm.

The output distance represents the distance for Send & Receive, so we need to divide it on 2 .The equation becomes:

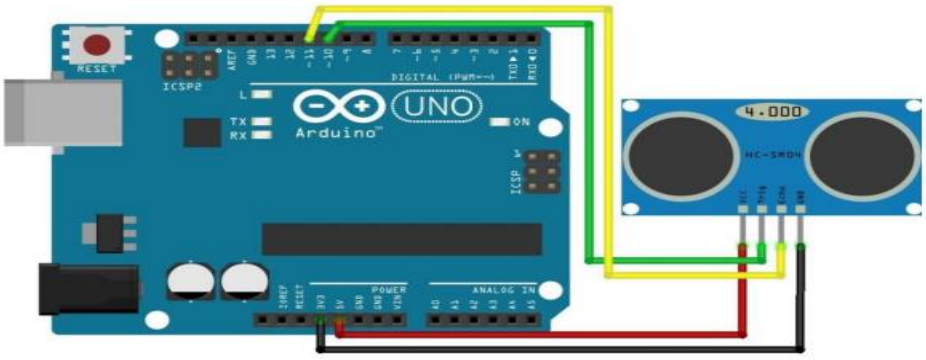
distance =duration* 0.034/2

Example 3:

(Measurement of distance)

Requirements :Arduino, Ultrasonic Sensor, wire.

Connection map:

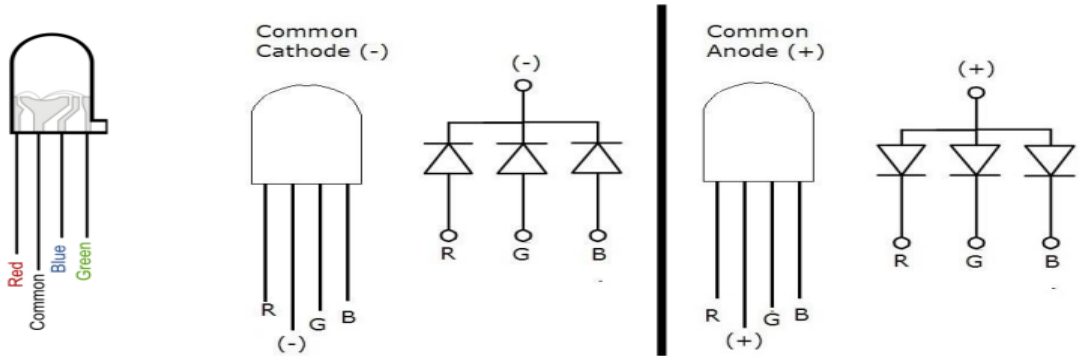


Code

```
int TRIG = 10;
int ECHO = 11;
void setup() {
  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);
  Serial.begin(9600);
}
void loop() {
  digitalWrite(TRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);
  long duration = pulseIn(ECHO, HIGH);
  int distance = duration * 0.034 / 2;
  if (duration == 0) {
    Serial.println("Warning: no pulse from sensor");
  } else {
    Serial.print("distance to nearest object:");
    Serial.print(distance);
    Serial.println(" cm");
  }
  delay(1000);}
```

RGB LED :

The RGB LED can emit different colors by mixing the 3 basic colors red, green and blue. So it actually consists of 3 separate LEDs red, green and blue packed in a single case. That's why it has 4 pins, one pin for each of the 3 colors and one common cathode or anode depending of the RGB LED type.

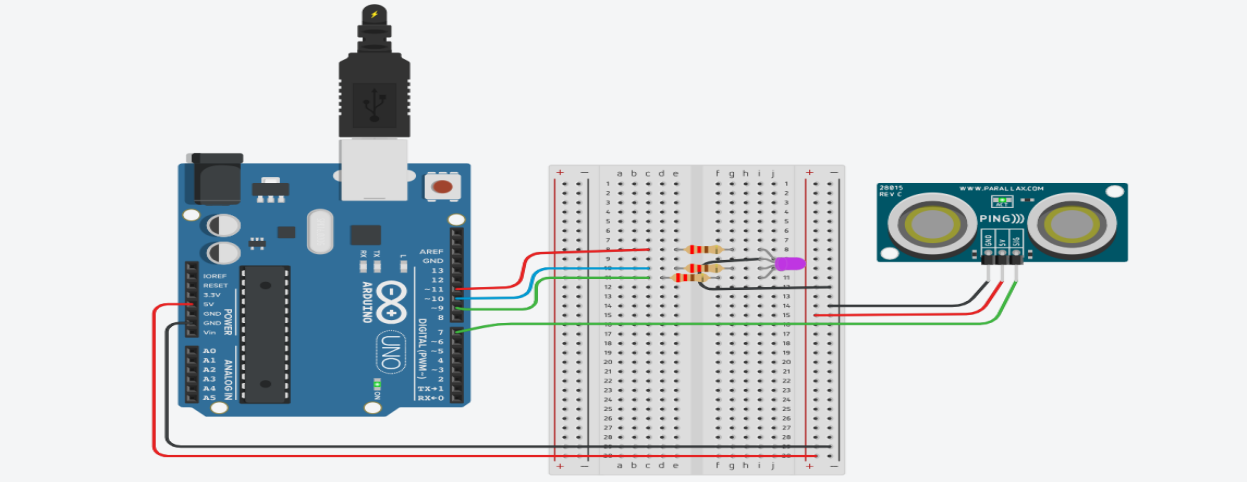


Example 4:

(Ultrasonic sensor with RGB)

Requirements: Arduino, BreadBoard, RGB Led, Ultrasonic sensor, 3Resistor, wires.

Connection map:



Code with explain :

```
const int pingPin = 7;
const int red=11;
const int blue=10;
int green=9;
void setup() {
  // initialize serial communication:
  Serial.begin(9600);
  pinMode(red,OUTPUT);
  pinMode(blue,OUTPUT);
  pinMode(green,OUTPUT);
}

void loop() {
  // establish variables for duration of the ping, and the distance result
  // in inches and centimeters:
  long duration, inches, cm;

  // The PING))) is triggered by a HIGH pulse of 2 or more microseconds.
  // Give a short LOW pulse beforehand to ensure a clean HIGH pulse:
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);

  // The same pin is used to read the signal from the PING))): a HIGH pulse
  // whose duration is the time (in microseconds) from the sending of the ping
  // to the reception of its echo off of an object.
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);

  // convert the time into a distance
  inches = microsecondsToInches(duration);
  cm = microsecondsToCentimeters(duration);

  Serial.print(inches);
  Serial.print("in, ");
  Serial.print(cm);
```

```
  Serial.print("cm");
  Serial.println();

  if(cm<256){
  analogWrite(red,cm);
  analogWrite(blue,255-cm);
  //analogWrite(green,inches);
  }
  else{analogWrite(red,0);analogWrite(blue,0);analogWrite(green,0);}
  delay(100);
}

long microsecondsToInches(long microseconds) {
  // According to Parallax's datasheet for the PING))), there are 73.746
  // microseconds per inch (i.e. sound travels at 1130 feet per second).
  // This gives the distance travelled by the ping, outbound and return,
  // so we divide by 2 to get the distance of the obstacle.
  // See: http://www.parallax.com/dl/docs/prod/acc/28015-PING-v1.3.pdf

  return microseconds / 74 / 2;
}

long microsecondsToCentimeters(long microseconds) {
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
  // The ping travels out and back, so to find the distance of the object we
  // take half of the distance travelled.
  return microseconds / 29 / 2;
}
```

Circle work online:

https://www.tinkercad.com/things/1uNbIdipyk2/editel