MICROCONTROLLER

Presentation by Sashwat K

Microcontroller

- A microcontroller is a small computer on a single integrated circuit.
- A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals.
- Microcontrollers are designed for embedded applications.
- Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems.
- Eg: ATmega series.

ARDUINO UNO

ARDUINO UNO

- Microcontroller: ATmega328
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- NOTE: PWM - Pulse Width Modulation
- Analog Input Pins: 6
- Flash Memory: 32 KB of which 0.5 KB used by bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz

ARDUINO PROGRAMS

1. LED blink (BUILTIN)
2. LED blink (LED)
3. Blinking 2 LEDs alternatively
4. Analog read using variable resistance
5. Serial monitor
6. Ultrasonic sensor
7. Adafruit OLED display
8. Servo motor

1. LED blink (BUILTIN)

HARDWARE NEEDED :-

- Arduino Uno - 1

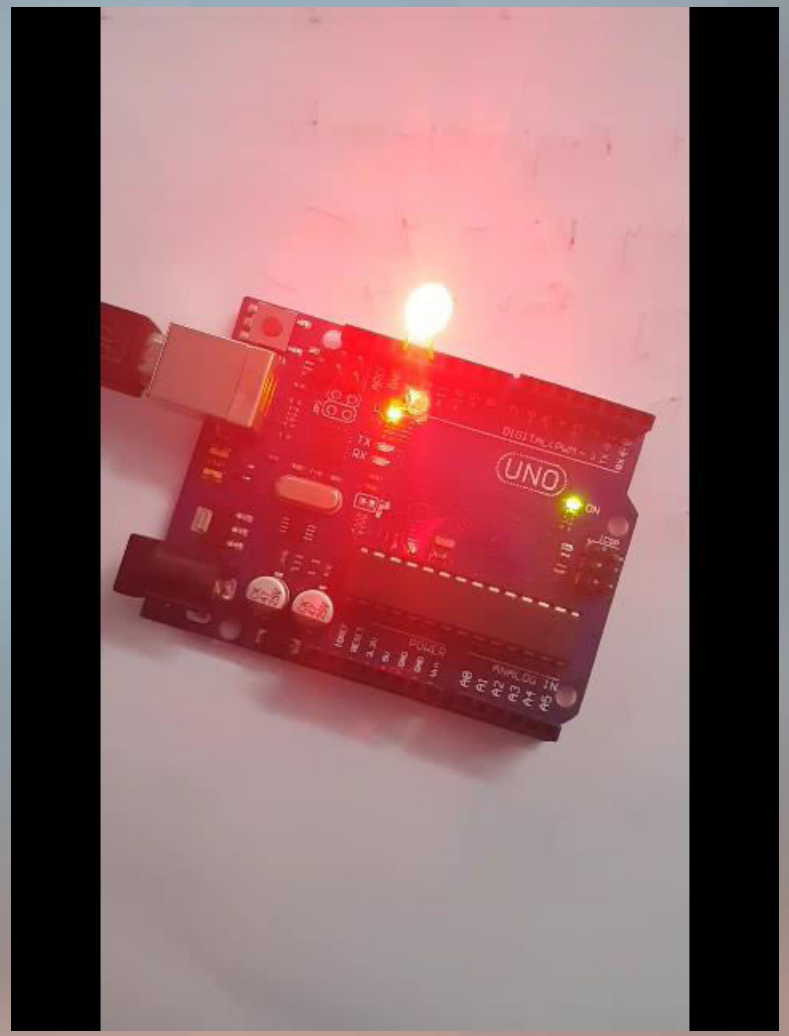
1. LED blink (BUILTIN) - CODE

```
1.  void setup() {
2.    // initialize digital pin LED_BUILTIN as an output.
3.    pinMode(LED_BUILTIN, OUTPUT);
4.  }

5.  // the loop function runs over and over again forever
6.  void loop() {
7.    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
8.    delay(1000);                     // wait for a second
9.    digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
10.   delay(1000);                    // wait for a second
11. }
```

2. LED blink (LED)

HARDWARE NEEDED :-

- Arduino Uno - 1
- LED - 1



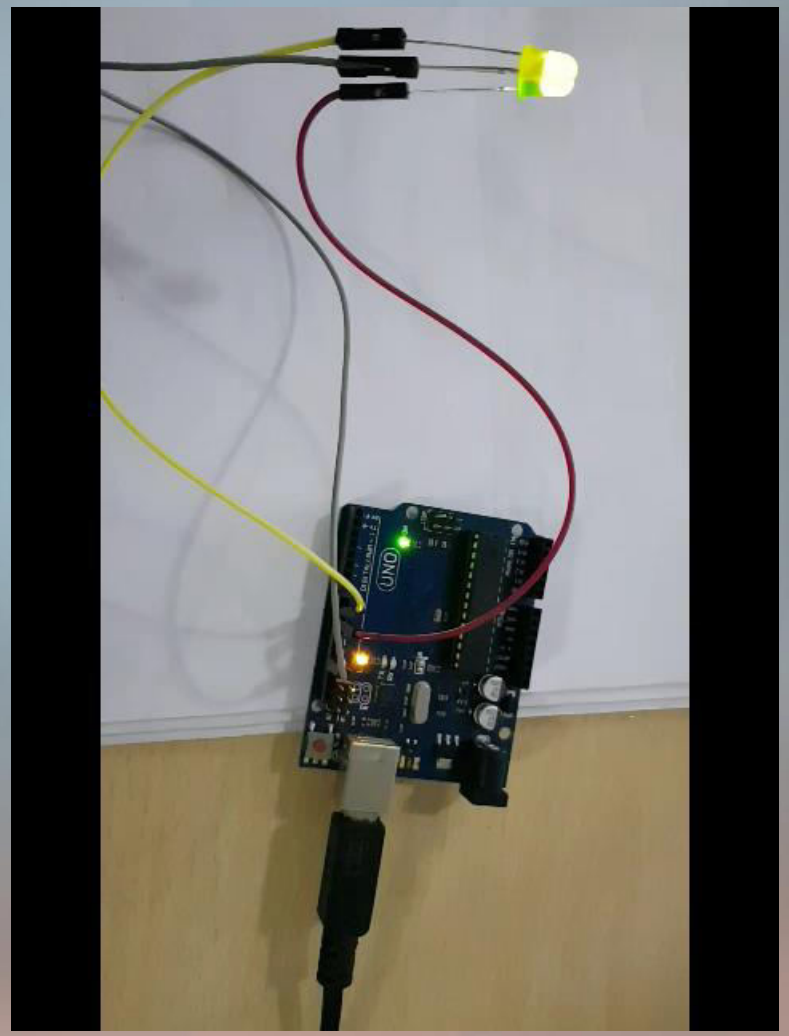
2. LED blink (LED) - CODE

```
1.  int LED1 = 13; // led 1 pin number

2.  // the setup function runs once when you press reset or power the board
3.  void setup() {
4.    // initialize digital pin LED_BUILTIN as an output.
5.    pinMode(LED1, OUTPUT);
6.  }

7.  // the loop function runs over and over again forever
8.  void loop() {
9.    digitalWrite(LED1, HIGH);   // turn the LED1 on (HIGH is the voltage level)
10.   delay(1000);                // wait for a second
11.   digitalWrite(LED1, LOW);    // turn the LED1 off by making the voltage LOW
12.   delay(1000);                // wait for a second
13. }
```

3. Blinking 2 LEDs alternatively

HARDWARE NEEDED :-

- Arduino Uno - 1
- LED – 2
- Jumper pin - 3



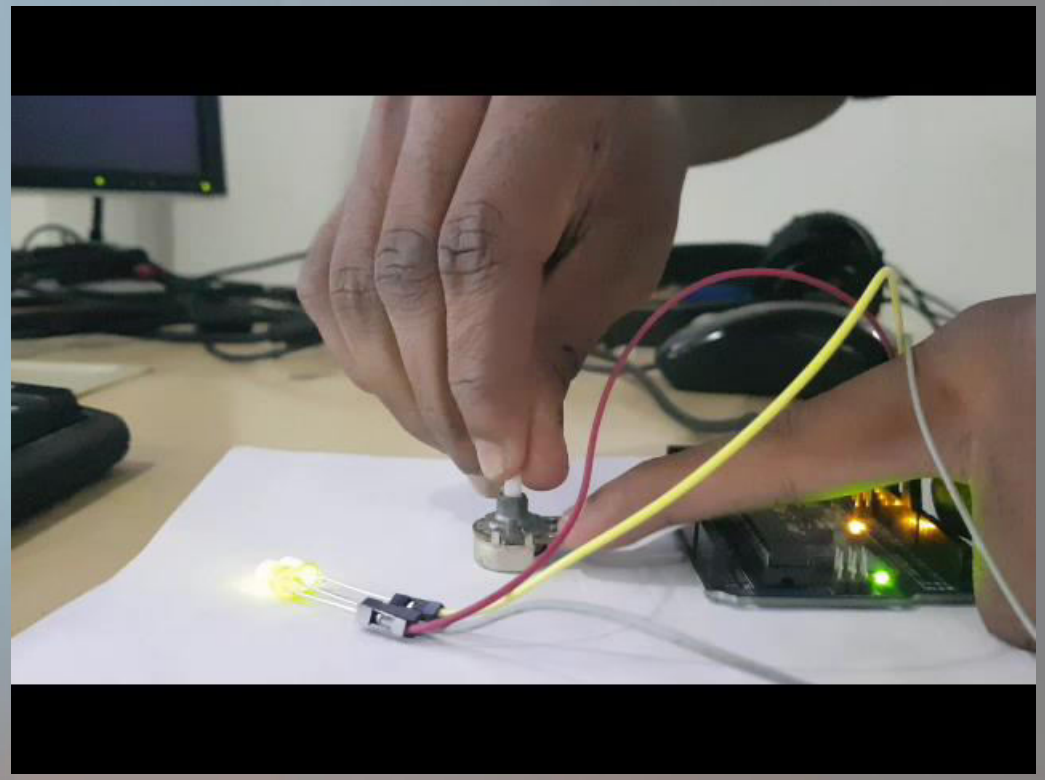
3. Blinking 2 LEDs alternatively - CODE

```
1.  int led1=5;
2.  int led2=7;
3.  void setup() {
4.    // initialize digital pin LED_BUILTIN as an output.
5.    pinMode(led, OUTPUT);
6.  }

7.  // the loop function runs over and over again forever
8.  void loop() {
9.    digitalWrite(led1, HIGH);   // turn the LED on (HIGH is the voltage level)
10.   digitalWrite(led2, LOW);    // turn the LED off by making the voltage LOW
11.   delay(1000);                // wait for a second
12.   digitalWrite(led1, LOW);    // turn the LED off by making the voltage LOW
13.   digitalWrite(led2, HIGH);   // turn the LED on (HIGH is the voltage level)
14.   delay(1000);                // wait for a second
15. }
```

4. Analog read using variable resistance

HARDWARE NEEDED :-

- Arduino Uno - 1
- LED – 2
- Jumper pin – 3
- Wire – 3
- Variable resistor - 1



4. Analog read using variable resistance - CODE

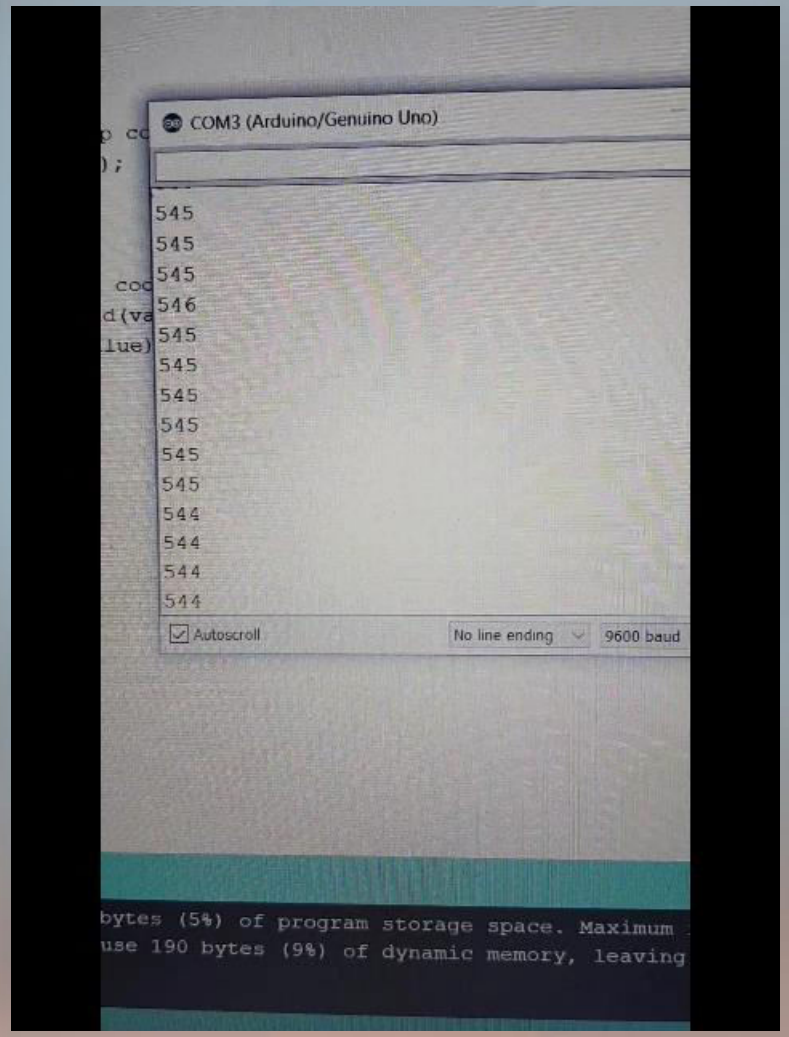
```
int varires = A0; // variable resistor pin number
int LED1 = 9; // LED 1 pin number
int LED2 = 10; // LED 2 pin number
int value; // store vale generated by variable resistor

void setup() {
  // put your setup code here, to run once:
 pinMode(varires,INPUT); // pinmode input to get value from resistor
 // initialize digital pin LED(s) as an output.
 pinMode(LED1, OUTPUT);
 pinMode(LED2, OUTPUT);
}
```

```
void loop() {
 // put your main code here, to run repeatedly:
 value = analogRead(varires); // generated value from variable resistor is assigned to "value"
 if(value>50) // if value greater than 50 , then set LED 1 as high and LED 2 as low
 {
  digitalWrite(LED1,HIGH);

 digitalWrite(LED2,LOW);
 }
 else // else, set LED 2 as High and LED 1 as low
 {
  digitalWrite(LED1,LOW);
  digitalWrite(LED2,HIGH);
 }
}
```

5. Serial Monitor

HARDWARE NEEDED :-

- Arduino Uno - 1
- Wire – 3
- Variable resistor - 1



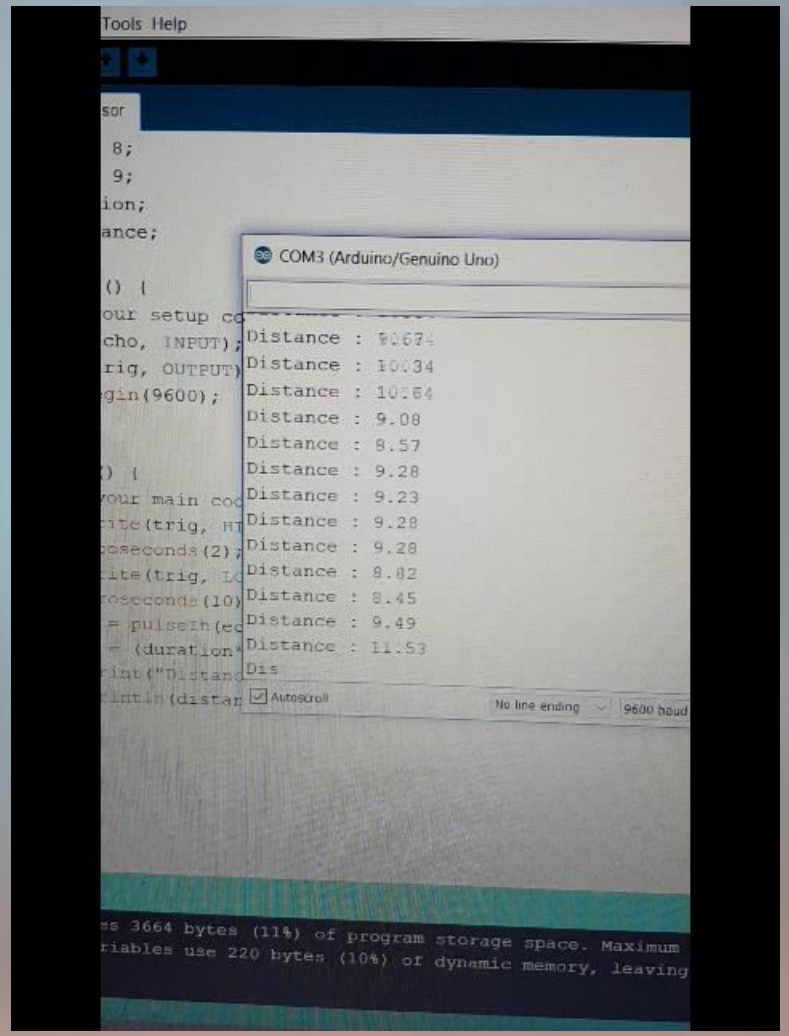
5. Serial Monitor - CODE

```
1.  Int led=5;
2.  Int led=7;
3.  void setup() {
4.    // initialize digital pin LED_BUILTIN as an output.
5.    pinMode(led, OUTPUT);
6.  }

7.  // the loop function runs over and over again forever
8.  void loop() {
9.    digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)

10.   delay(1000);                  // wait for a second
11.   digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
12.   delay(1000);                        // wait for a second
13. }
```

6. Ultrasonic Sensor

HARDWARE NEEDED :-

- Arduino Uno
- Ultrasonic sensor - 1
- Jumper pin - 4



6. Ultrasonic Sensor - CODE

```
1.  int trig = 8; // trigger pin number of ultrasonic sensor in arduino
2.  int echo = 9; // echo pin number of ultrasonic sensor in arduino
3.  long duration; // duration calculated from ultrasonic sensor
4.  float distance; // distance calculated from duration

5.  void setup() {
6.    // put your setup code here, to run once:
7.   pinMode(echo, INPUT);
8.   pinMode(trig, OUTPUT);
9.   Serial.begin(9600);
10. }

11. }
```

```
1.  void loop() {
2.    // put your main code here, to run repeatedly:
3.    // next 5 statements will create a digital wave
4.   digitalWrite(trig, HIGH);
5.   delayMicroseconds(2);
6.   digitalWrite(trig, LOW);
7.   delayMicroseconds(10);
8.   duration = pulseIn(echo,HIGH);
9.
10.  distance = (duration*0.034)/2; // distance = speed *time
11.
12.  Serial.print("Distance : ");
13.  Serial.println(distance);
```

7. Adafruit OLED Display

HARDWARE NEEDED :-

- Arduino Uno - 1
- Adafruit OLED display - 1
- Jumper pin - 4

7. Adafruit OLED Display - CODE

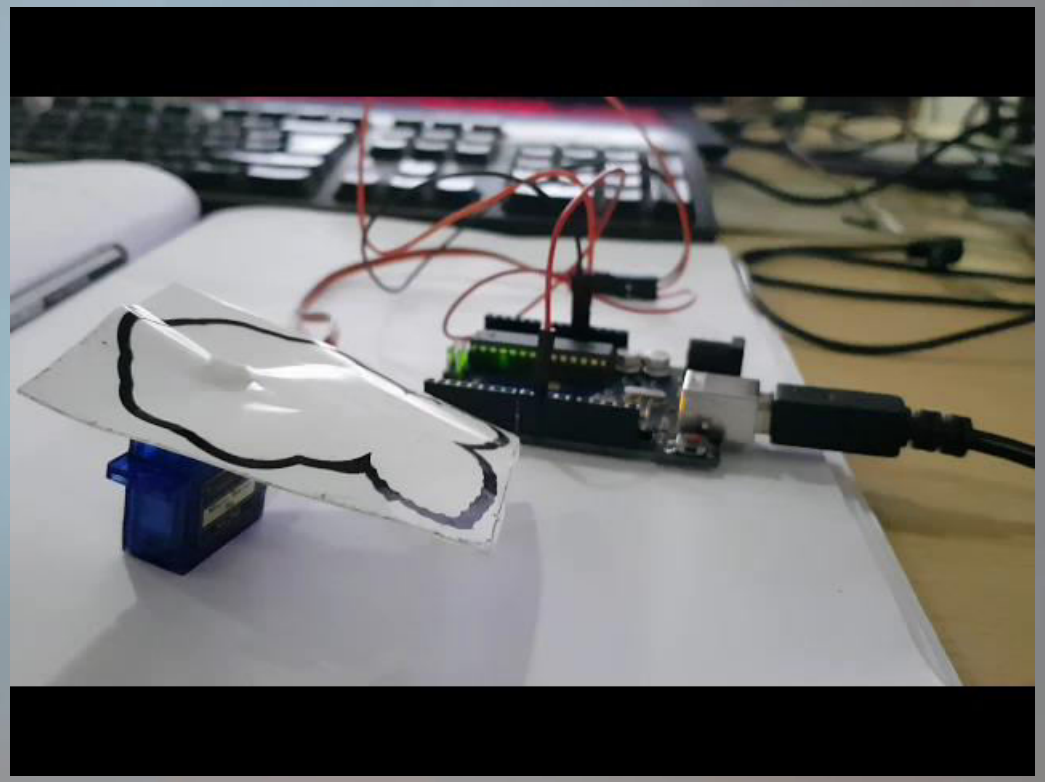
```
#include <SPI.h>  // adafruit display driver header file
#include <Wire.h> // adafruit display driver header file
#include <Adafruit_GFX.h> // adafruit display driver header file
#include <Adafruit_SSD1306.h> // adafruit display model header file

#define OLED_RESET 4 //OLED
Adafruit_SSD1306 display(OLED_RESET); //OLED

void setup() {
  // put your setup code here, to run once:
 display.begin(SSD1306_SWITCHCAPVCC, 0x3C); //OLED
 display.display(); //initialize display
 delay(2000);
 display.clearDisplay(); //clear display
}
```

```
void loop() {
  // put your main code here, to run repeatedly:
 display.clearDisplay();
 display.display();
 display.setTextSize(2); // setting text size
 display.setTextColor(WHITE); // setting text color
 display.setCursor(0,0); // setting position for OLED display
 display.println("IOT LAB");
 display.println("CET");
 display.display();
}
```

8. Servo Motor

HARDWARE NEEDED :-

- Arduino Uno – 1
- Servo Motor - 1
- Jumper pin - 3



8. Servo Motor - CODE

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}
```

```
void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```

THANK YOU

The presentation and program codes will be available in the following links. Follow the following links for more contents.