Introduction To Embedded System

INDEX

Sr No	Date	Practical
1		Introduction to Arduino
2		Program using Light Sensitive Sensors
3		Program using temperature sensors
4		Program using Humidity sensors
5		Program using Ultrasonic Sensors
6		Programs using Servo Motors
7		Programs using digital infrared motion sensors
8		Programs using Gas sensors
9		Programs making Joystick with Arduino

Practical 1: Introduction to Arduino

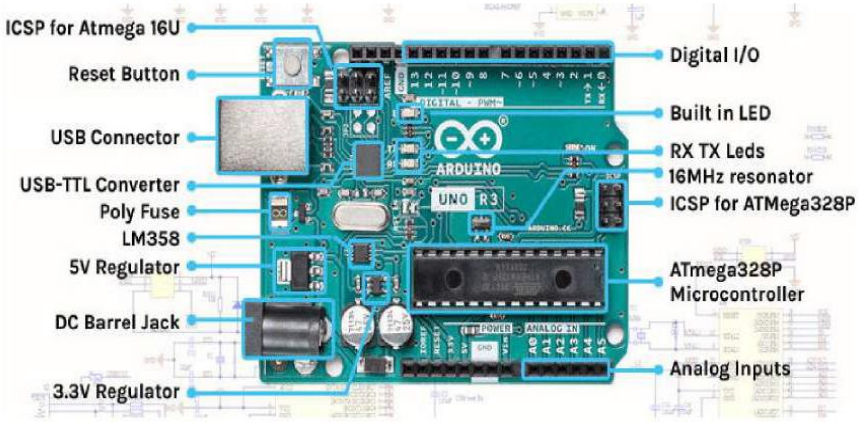
Aim:

1. To study the basics of Arduino circuits and bread-boarding
2. Blinking of LEDs

Simulation Environment: TinkerCAD (Free online simulator)

Part A: Basics of Arduino Circuits

Theory:



Arduino is an open-source electronics platform that has gained immense popularity for its ease of use and versatility. It was created in 2005 by a group of Italian engineers and is now maintained and developed by the Arduino community.

The heart of the Arduino platform is a microcontroller, which is a small, programmable computer on a single integrated circuit (IC) chip.

Arduino boards, which house these microcontrollers, provide a user-friendly environment for creating interactive electronic projects, prototypes, and various applications.

Key Components of Arduino:

1. Microcontroller: The core of an Arduino board is the microcontroller. The most commonly used microcontroller in Arduino is the ATmega series from Atmel (now a part of Microchip Technology). These microcontrollers come in different variations and are the brains behind your Arduino projects.

2. Input/output Pins: Arduino boards have a set of digital and analog pins that can be used to read data (inputs) or send data (outputs). Digital pins work with binary signals (0 or 1), while analog pins can read a range of values. The number and types of pins vary among different Arduino board models.

3. Power Supply: Arduino boards can be powered via USB, an external power supply, or a battery. Some boards have built-in voltage regulators, which make them compatible with a range of power sources.

4. USB Port: Arduino boards often feature a USB port for programming and power supply. This allows you to connect the board to your computer and upload code.

5. Reset Button: A reset button is provided to restart the Arduino, allowing you to upload new code or reset the program.

6. LED Indicator: Many Arduino boards include a built-in LED (Light Emitting Diode) on pin 13, which can be used for testing and basic visual feedback.

Arduino Software:

The Arduino platform comes with its integrated development environment (IDE). The Arduino IDE is a software tool that allows you to write, compile, and upload code to the Arduino board. Key features of the IDE include:

- Programming Language: Arduino uses a simplified version of the C/C++ programming language. It provides a set of libraries and functions tailored for easy interaction with the hardware.

- Code Library: Arduino has a vast library of pre-written code and functions that simplify common tasks, making it accessible to beginners.

- Serial Monitor: The IDE includes a serial monitor that allows you to communicate with the Arduino board and view debugging information.

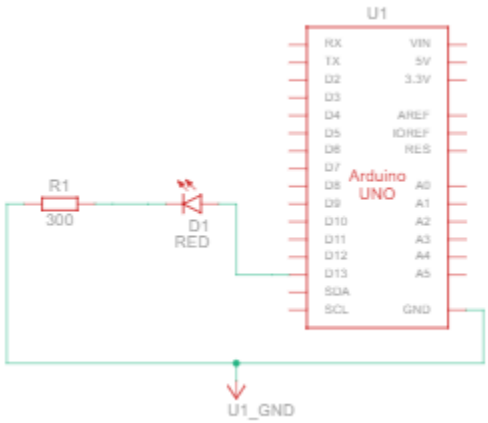
- Community Support: The Arduino community is large and active, offering forums, tutorials, and extensive documentation to help users troubleshoot issues and learn.

Part B: Blinking of LEDs

Components Used:

1. Arduino UNO
2. Breadboard
3. LED
4. Resistor (330 Ω)

Circuit Diagram:



CODE:

```
// C++ code
//
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
}


void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
  delay(100); // Wait for  millisecond(s)
```

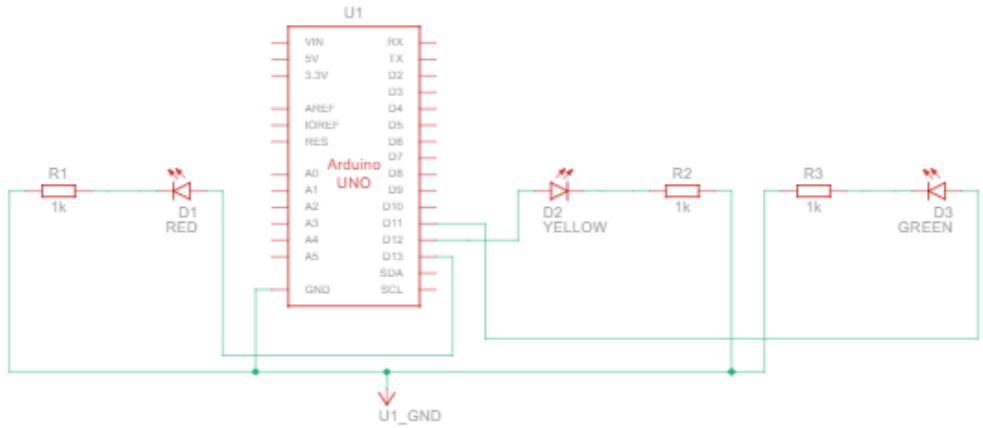
```
 digitalWrite(LED_BUILTIN, LOW);
 delay(100); // Wait for 1000 millisecond(s)
}
```

Part C: Blinking of Three LED's

Component Used

1. Arduino UNO
2. Breadboard
3. 3LEDs
4. 3 Resistors (330 ohm)

Circuit Digram



Code: int animationSpeed = 400; // Set initial speed

```
void setup() {
   pinMode(13, OUTPUT);
   pinMode(12, OUTPUT);
   pinMode(11, OUTPUT);
}
```

```
void loop() {
   digitalWrite(13, HIGH);
   delay(animationSpeed);
   digitalWrite(13, LOW);
  // delay(animationSpeed);



   digitalWrite(12, HIGH);
   delay(animationSpeed);
   digitalWrite(12, LOW);
 // delay(animationSpeed);

  digitalWrite(11, HIGH);
   delay(animationSpeed);
   digitalWrite(11, LOW);
//  delay(animationSpeed);
```

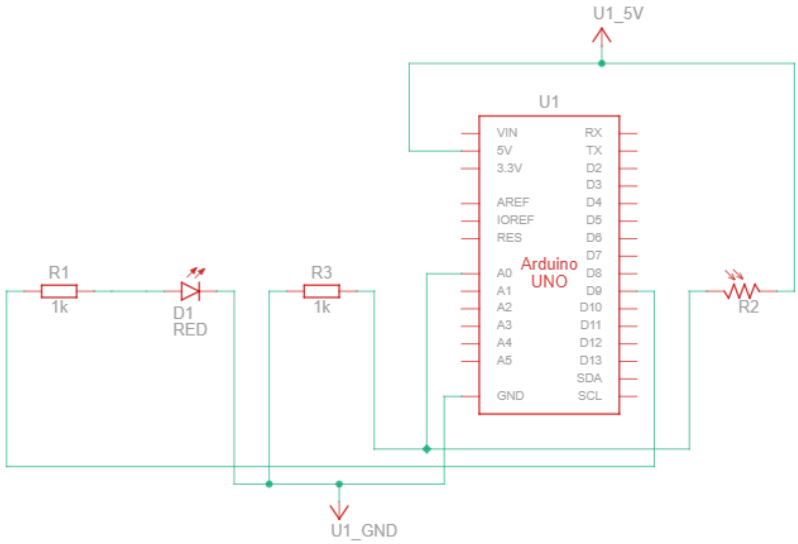
Practical No. 2: Program Using Light Sensitive Sensors

Aim: To study the working of Light sensor using Arduino

Simulation Environment: Tinker CAD

Components: Arduino, Photodiode and Resistors

Circuit Diagram



Pin Connections:

Arduino	Photoresistor	LED
5V	Right pin	
GND (Power)	Left pin through a Series Resistor	
A_0	Left pin	
Pin 9		Anode
GND (Digital)		Cathode through a Series Resistor

CODE:

```
int lightSensorValue = 0;


void setup() {
  pinMode(A0, INPUT);
  pinMode(9, OUTPUT);
  Serial.begin(9600);
}
```

```
void loop() {
  lightSensorValue = analogRead(A0);  // Read sensor value
  Serial.println(lightSensorValue);   // Print value to Serial Monitor

  int ledBrightness = map(lightSensorValue, 0, 1023, 0, 255); // Map sensor range to LED brightness
  analogWrite(9, ledBrightness); // Set LED brightness

  delay(100);  // Small delay to stabilize readings
}
```

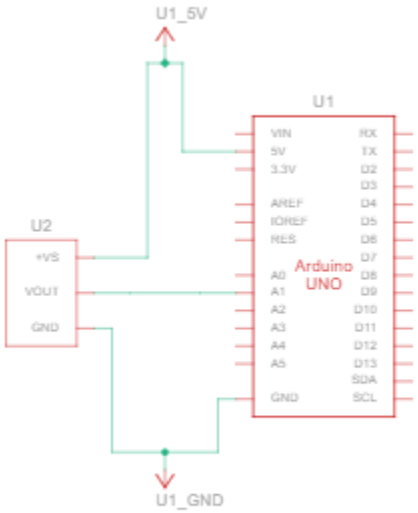
Practical No. 2: Program Using temperature Sensors

Aim: To study the working of Light sensor using Arduino

Simulation Environment: Tinker CAD

Components: Arduino, temperature Sensor TMP 36

Circuit Diagram:



Pin Connections:

Arduino	TMP36 Sensor
5V	Left pin
GND	Right pin
A_1	Center pin

Code: const int sensor = A1; // Define sensor pin

char degree = 176; // ASCII code for degree symbol

void setup() {

  pinMode(sensor, INPUT);

  Serial.begin(9600); // Corrected function name

}

void loop() {

  int tmp = analogRead(sensor); // Read sensor value

  float voltage = (tmp * 5.0) / 1024; // Convert to voltage

```
  float tempCelsius = voltage * 100;  // Convert voltage to temperature (LM35 sensor)

  Serial.print("Celsius: ");
  Serial.print(tempCelsius);
  Serial.print((char)degree);  // Print degree symbol
  Serial.println("C");

  delay(1000);  // Wait 1 second
}
```

Practical 4: Program using Humidity sensors

Aim:

To study the working of Humidity sensors using Arduino

Simulation Environment: TinkerCAD (Free online simulator)

Components: Arduino UNO, Potentiometer (wiper)

Pin Connections:

Arduino	Potentiometer
5V	Left pin
GND	Right pin
A_1	Center pin

Practical 5: Programs using Ultrasonic Sensors

Aim:

To study the working of Ultrasonic sensors using Arduino

Simulation Environment: TinkerCAD (Free online simulator)

Components: Arduino UNO, HC-SR04 sensor

Pin Connections:

Arduino	HC-SR 04 Sensor
5V	V_{CC}
GND	GND
Pin 9	TRIG
Pin 10	ECHO

Practical 6: Programs using Servo Motors

Aim:

To control the motion of a Servo motor using Arduino

Simulation Environment: TinkerCAD (Free online simulator)

Components: Arduino UNO, Servo motor

Pin Connections:

Arduino	Servo Motor
5V	Power
GND	Ground
Pin A_1	Signal

Code: