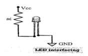
EXP 1

Sensor: A sensor is an object whose purpose is to detect events or changes in its environment, and then provide a corresponding output. A sensor is a type of transducer; sensors may provide various types of output, but typically use electrical or optical signals.

Actuator: An actuator is a component of machines that is responsible for moving or controlling a mechanism or system. An actuator requires a control signal and source of energy. The control signal is relatively low energy and may be electric voltage or current, pneumatic or hydraulic pressure, or even human power

I/O subsystem: The I/O sub system of the embedded system facilitates the interaction of embedded system with the external world

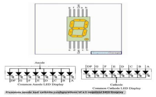
LED (Light Emitting Diode): LED is an important output device for visual indication in any embedded system.LED used as an indicator for the status of various signals or situations. LED is a p-n junction diode and contains a CATHODE and ANODE



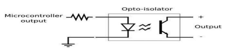
There are two ways to interface an LED to a microprocessor/microcontroller: The Anode of LED is connected to the port pin and cathode to Ground and The Cathode of LED is connected to the port pin and Anode to Vcc

Segment LED display: • It is the most basic electronic display device that can display digits from 0-9. • They find wide application in devices that display numeric information like digital clocks, radio, microwave ovens, electronic meters etc.

Pin diagram:



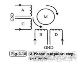
Optocoupler: Optocoupler is , also called photocoupler, or optical isolator, is a component that transfers electrical signals between two isolated circuits by using light.



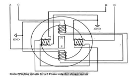
STEPPER MOTOR: A stepper motor 1s an electro-mechanical device which generates discrete displacement (motion) in response, to de electrical signals. Based on the coil winding arrangements, a two-phase stepper motor is classified into two. They are: 1. Unipolar 2. Bipolar

1. Unipolar

A unipolar stepper motor contains two windings per phase. The direction of rotation (clockwise or anticlockwise) of a stepper motor is controlled by changing the direction of current flow.



2. Bipolar:
A bipolar stepper motor contains single winding per phase.For reversing the motor rotation the current flow through the winding is reversed dynamically



Full Step: In the full step mode both the phases are energised simultaneously. The coils A, B, C and D are energised in the following order:

STEP	COIL A	COIL B	COIL C	COIL D
1	H	H	L	L
2	L	H	H	L
3	L	L	H	H

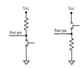
Wave Step: In the wave step mode only one phase is energized at a time and each coils of the phase is energized alternatively .The coils A,B,C and D are energized in the following order

STEP	COIL A	COIL B	COIL C	COIL D
1	H	H	L	L
2	L	H	L	L
3	L	L	H	L
4	L	L	L	H

Half Step: It uses the combination of wave & full step.It has the highest torque and stability. The coil energizing sequence for half step is given below

STEP	COIL A	COIL B	COIL C	COIL D
1	H	L	L	L
2	L	H	L	L
3	L	H	L	L
4	L	H	H	L
5	L	L	H	L
6	L	L	H	H
7	L	L	L	H
8	H	L	L	H

Push Button: Push Button Switch It is an input device. Push button switch comes in two configurations, namely “Push to Make’ and ‘Push to Break’.



EXP 2

A cyber-physical system (CPS) or intelligent system is a computer system in which a mechanism is controlled or monitored by computer based algorithms. CPS is also similar to the Internet of Things (IoT), sharing the same basic architecture; nevertheless.

Examples of automobile systems, medical CPS include smart monitoring, industrial grid, autonomous control systems, robotics systems, and automatic pilot avionics.

INDUSTRY 4.0

Industry 4.0 refers to the fourth industrial revolution.

Fourth Industrial Revolution

1st – Steam & Water, 2nd – Electricity, 3rd- IoT, 4th – CPS

Industry 4.0 Technologies

Industry 4.0 describes the growing trend towards automation and data exchange in technology and processes within the manufacturing industry, including: • The internet of things (IoT) • The industrial internet of things (IIoT) • Cyber-physical systems (CPS) • Smart manufacture • Smart factories • Cloud computing • Cognitive computing • Artificial intelligence

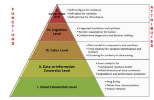


Figure 3. Industry 4.0 Revolution

CYBER PHYSICAL SYSTEM OPPORTUNITIES

Smart city management – CPS enables intelligent traffic, emergency response, and public safety systems, improving operational efficiency and citizen welfare

Infrastructure – CPS with IoT sensors and cameras helps upgrade outdated infrastructure for safer, smarter cities

Automotive – CPS-driven smart car features like blind-spot monitoring and lane-departure warnings enhance road safety

Agriculture – CPS tools such as drones and smart sensors improve farming efficiency and crop management

Smart city management – CPS enables intelligent traffic, emergency response, and public safety systems, improving operational efficiency and citizen welfare

Infrastructure – CPS with IoT sensors and cameras helps upgrade outdated infrastructure for safer, smarter cities

Automotive – CPS-driven smart car features like blind-spot monitoring and lane-departure warnings enhance road safety.

Agriculture – CPS tools such as drones and smart sensors improve farming efficiency and crop management

Sustainability – CPS supports eco-friendly solutions like public electric transport and renewable energy systems

Security – CPS powers intelligent surveillance and remote monitoring for improved safety

Health care – CPS enables advanced patient monitoring systems for better diagnosis and treatment tracking

Challenges and risks in Cyber-Physical Systems related to Industry 4.0 are: • Data protection and data security. • Lack of benefit quantification. • Lack of prioritization by top management. • Industrial broadband structure.

EXP 12

Introduction: MQTT stands for Message Queuing Telemetry Transport. It is an extremely lightweight and publish-subscribe messaging transport protocol.

Characteristics:

Machine-to-machine communication., Lightweight publish/subscribe model., Works without simultaneous client-server connection., Supports real-time messaging., Allows subscription to specific topics.

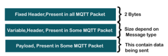
Components:

Message: Data sent over the network with payload, QoS, properties, and topic name., **Client**: Publishes or subscribes to topics., **Server/Broker**: Manages connections, subscriptions, and message forwarding., **Topic**: Label used to route messages to subscribers.

MQTT Message Format:

Uses a command–acknowledgment structure, where each command has an associated acknowledgment.

Packet Structure: Fixed header (2 bytes) – in all packets., Variable header – optional., Payload – optional, contains the data.



CPS LAB PGMS

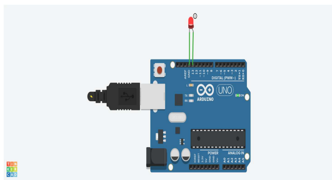
Ex. 3.1. Create a program that blinks the LED on the development board using MBED software

Components Used:

- Arduino Uno
- LED

Procedure:

- LED has two side which is positive and Negative.
- Negative side is connected to the Ground. Positive side is connected to Digital pin 13 of Arduino.



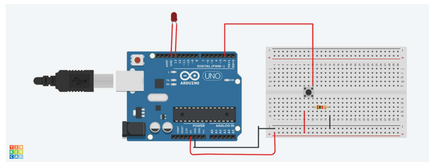
Code:

```
void setup()
{pinMode(13, OUTPUT);}
void loop()
{digitalWrite(13, HIGH);
delay(1000);
digitalWrite(13, LOW);
delay(1000);}
```

Ex. 3.2 Through Button blink LED

Components Used:

- LED
- Pushbutton
- 220Ω resistor (for LED)
- Arduino UNO board.
- Breadboard



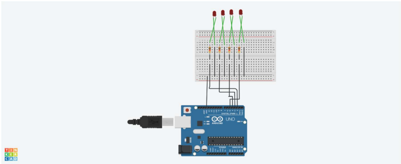
Code:

```
void setup()
{pinMode(2, INPUT);
pinMode(13,OUTPUT);}
void loop()
{if(digitalRead(2)==1)
{digitalWrite(13,HIGH);}
 else
 {digitalWrite(13,LOW); }}
```

Ex. 3.3 Create a program that blinks the LED with digital counter

Components Used:

- 4-LED
- 4-220Ω resistor (for LED)
- Arduino UNO board.
- Breadboard



Code:

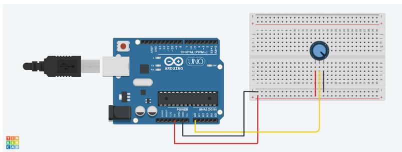
```
int pin2=2;
int pin3=3;
int pin4=4;
int pin5=5;
int stime=500;
void setup()
{pinMode(pin2,OUTPUT);
pinMode(pin3,OUTPUT);
pinMode(pin4,OUTPUT);
pinMode(pin5,OUTPUT);}
void loop()
{digitalWrite(pin2,LOW);
digitalWrite(pin3,LOW);
digitalWrite(pin4,LOW);
digitalWrite(pin5,LOW);
delay(stime);
digitalWrite(pin2,LOW);
digitalWrite(pin3,LOW);
digitalWrite(pin4,LOW);
digitalWrite(pin5,HIGH);
delay(stime);
digitalWrite(pin2,LOW);
digitalWrite(pin3,LOW);
digitalWrite(pin4,HIGH);
digitalWrite(pin5,LOW);
  delay(stime);
digitalWrite(pin2,LOW);
digitalWrite(pin3,LOW);
digitalWrite(pin4,HIGH);
digitalWrite(pin5,HIGH);
  delay(stime);
digitalWrite(pin2,LOW);
digitalWrite(pin3,HIGH);
digitalWrite(pin4,LOW);
digitalWrite(pin5,LOW);
  delay(stime);
digitalWrite(pin2,LOW);
digitalWrite(pin3,HIGH);
digitalWrite(pin4,LOW);
digitalWrite(pin5,HIGH);
  delay(stime);
digitalWrite(pin2,LOW);
digitalWrite(pin3,HIGH);
digitalWrite(pin4,HIGH);
digitalWrite(pin5,LOW);
  delay(stime);
digitalWrite(pin2,LOW);
digitalWrite(pin3,HIGH);
digitalWrite(pin4,HIGH);
digitalWrite(pin5,HIGH);
  delay(stime);
digitalWrite(pin2,HIGH);
digitalWrite(pin3,LOW);
digitalWrite(pin4,LOW);
digitalWrite(pin5,LOW);
  delay(stime);
digitalWrite(pin2,HIGH);
digitalWrite(pin3,LOW);
digitalWrite(pin4,LOW);
digitalWrite(pin5,HIGH);
  delay(stime);
digitalWrite(pin2,HIGH);
digitalWrite(pin3,LOW);
digitalWrite(pin4,HIGH);
digitalWrite(pin5,LOW);
  delay(stime);
digitalWrite(pin2,HIGH);
digitalWrite(pin3,LOW);
digitalWrite(pin4,HIGH);
digitalWrite(pin5,HIGH);
  delay(stime);
digitalWrite(pin2,HIGH);
digitalWrite(pin3,HIGH);
digitalWrite(pin4,LOW);
digitalWrite(pin5,LOW);
  delay(stime);
digitalWrite(pin2,HIGH);
digitalWrite(pin3,HIGH);
digitalWrite(pin4,LOW);
digitalWrite(pin5,HIGH);
  delay(stime);
digitalWrite(pin2,HIGH);
digitalWrite(pin3,HIGH);
digitalWrite(pin4,HIGH);
digitalWrite(pin5,LOW);
  delay(stime);
digitalWrite(pin2,HIGH);
digitalWrite(pin3,HIGH);
digitalWrite(pin4,HIGH);
digitalWrite(pin5,HIGH);
  delay(stime);}
```

4. Pick one-one from the available sensors and actuators and find or create code that will display the sensed data on the PC.

4.1 Using Potentiometer

Components Used:

- Arduino Uno
- Potentiometer
- Breadboard and jumper wires



Code:

```
int pot=A0;
void setup()
{Serial.begin(9600);}
void loop()
{int potvalue=analogRead(pot);
Serial.print("pot value");
Serial.println(potvalue);
delay(1);}
```

Output:

pot value818

pot value777

pot value716

pot value696

pot value61

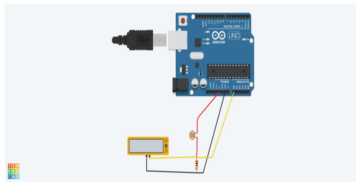
pot value20

pot value0

4.2 Using Photoresistor

Components Used:

- Arduino Uno
- Photoresistor
- Breadboard and jumper wires
- Multi meter



Code:

```
void setup()
{pinMode(A0, INPUT);
Serial.begin(9600);}
void loop()
{int lightvalue=analogRead(A0);
Serial.println(lightvalue);
delay(1000);}
```

Output:

6

6

379

526

640

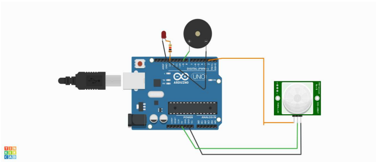
658

6

5.Create a program that displays data from the sensor in regular intervals in a compact format.

Components Used:

- Arduino Uno
- LED
- Resistor
- PIR Sensor
- Piezo

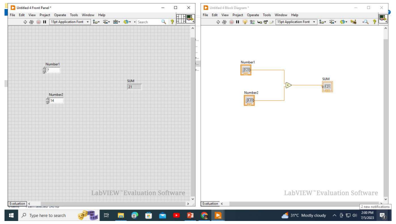


Code:

```
void setup()
{pinMode(2, INPUT);
 pinMode(13, OUTPUT);
 pinMode(9, OUTPUT);}
void loop()
{ if (digitalRead(2) >= HIGH) {
   digitalWrite(13, HIGH);
   tone(9, 523, 1000); }
else { digitalWrite(13, LOW);
   noTone(9); }
 delay(1);
}
```

6. To perform a vi for simple addition using embedded components in Lab view software.

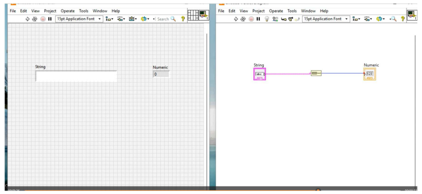
- Open lab view software. Open blank VI.
- Insert the numeric controls as an input from the control palette.
- Add the Adder tool from the numeric sub palette from the function palette.
- Insert the numeric indicator as an output from the control palette.
- Connect all the terminals in the block diagram window.
- Enter inputs in the front panel and click run to display the result.



7. To perform string operations using embedded components in Lab view software.

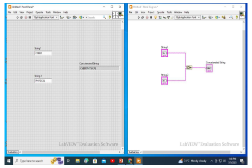
a) String Length

- Launch the LabVIEW software and create a new VI
- Place a String Control from the Controls palette
- Place a Numeric Indicator from the Controls palette
- Locate the String Length function
- Wire the String Control output to the input of the String Length function.
- Connect the output of the String Length function to the Numeric Indicator.
- Click the Run button.



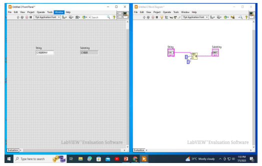
7b. String Concatenation.

- Open lab view software. Open blank VI.
- **Front Panel Controls Needed:**
- String Control – Label it: String 1
- String Control – Label it: String 2
- String Indicator – Label it: Concatenated String
- **Block Diagram Elements Needed:**
- "Concatenate Strings" function
- Terminals of String Controls
- Terminal of String Indicator
- Connect all the terminals in the block diagram window.
- Enter inputs in the front panel and click run to display the result.



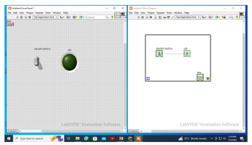
7c. Sub String

- Open lab view software. Open blank VI.
- **Front Panel Controls Needed:**
- String Control
- String Indicator
- **Block Diagram Elements Needed:**
- String Control
- String subset
- String Indicator
- Connect all the terminals in the block diagram window.
- Enter inputs in the front panel and click run to display the result.



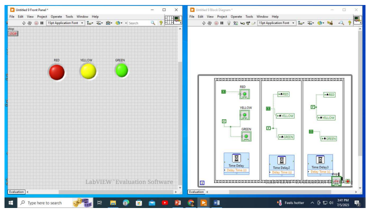
8. To perform led on/off

- Open lab view software. Open blank VI.
- **Front Panel elements needed:**
- a) Toggle Switch:
- b) Round LED:
- **Block Diagram Elements Needed:**
- a) While Loop
- b) Toggle Switch
- c) Round LED
- d) STOP Button
- Connect all the terminals in the block diagram window.
- Enter inputs in the front panel and click run to display the result.



9. Traffic Signal Light

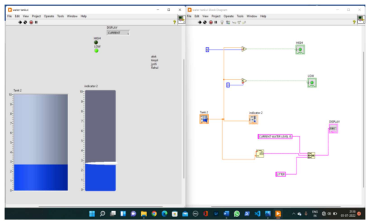
- Open lab view software. Open blank VI.
- **Front Panel elements needed: 3 LEDs**
- **Block Diagram elements needed: FLAT SQUENCE**
- From the Boolean section, choose three led's and place them in the control panel in vertical order.
- Encapsulate the led's within the „flat sequence„ in the block diagram panel. Add two more frames after that.
- In the frames, place the local variables of the led's and change the constants to T and F.
- Then add the time delay block to each frame and set delay as 2 sec.
- Now, run the program.



10. Water Tank Indicator

Procedure

- Open lab view software. Open blank VI.
- **Front Panel elements needed:**
- Tank
- Vertical pointer
- LED for high- and low-level indication
- Display-String indicator
- **Block Diagram elements needed:**
- 2 High/Low indicators
- 2 Tank
- 2 Indicator
- String Constant
- String indicator
- Numerical constants
- Comparison operator
- Bundle function
- Go to the "Front Panel" and press "Right Click" from your personal computer or laptop.
- Now, go to the Controls-> Modern-> Numeric-> Tank.
- Now go to the "Front Panel" and then go to the Controls-> Modern-> Numeric-> Vertical Pointer.
- We connect the water tank to the indicator.
- Now, go to the Controls-> Modern->comparison->greater or equal and connect the output to the high and create a constant. After that go to the Controls-> Modern- >comparison- >less or equal and connect the output to the low and create a constant.
- Now we use a function which will convert number to string for that go to function->string- >number to string->number to decimal string.
- For display the information current water level we take a concatenate string.
- Now run the program.



11. Real Time Temperature Converter

Procedure

- Open lab view software. Open blank VI.
- **Front Panel elements needed:**
- 1.Numeric up down control
- 2.Numeric control
- 3. Thermometer
- **4. Block Diagram elements needed:**
- Celsius-Numerical Control
- 9, 5 and 32 – Numeric Constants
- Multiplication symbol
- Division Symbol
- Addition/Subtraction symbol
- Fahrenheit- Numerical indicator
- Thermometer indicator
- Go to the Front Panel and right click on it.
- Now, go to the Controls -> Modern -> Numeric -> Numeric Control.
- Select this block and placed it on the front panel.
- Changed in name to Celsius.
- Now, go to Controls -> Modern -> Numeric -> Numeric Indicator.
- Selected this block and place it on the Front Panel
- Named it as a Fahrenheit.
- Select another Numeric Control and place it similarly on the Front Panel and name it Kelvin.
- Go to Functions -> Express -> Arithmetic & Comparison -> Formula.
- Select this block and place it on the "Front Panel" window.
- Now, I have changed the formula in order to convert "Celsius" to "Fahrenheit".
- After changing the formula pressed "OK".
- Now, connected the "Celsius" with the "X1" terminal of the "Formula" block and connect "Result" with the "Fahrenheit" block.
- Now, run the program.