

EXPERIMENT NUMBER : 5

DATE : 05/04/2022, TUESDAY

ARDUINO INTERFACING WITH PROCESSING IDE

* AIM :

To design and develop embedded systems using Arduino and Processing IDE.

* CODE :

(A) Serial Reception :-

① Arduino Source Code -

// The setup function runs once when you press reset or power the board :

```
void setup ()  
{
```

```
  serial.begin (9600); // start serial communications at 9,600  
  bits per second
```

```
}
```

// The loop function runs over and over again forever :

```
void loop ()
```

```
{
```

```
  serial.println ("Hello, World!");  
  delay (100); // Delay in between reads for stability
```

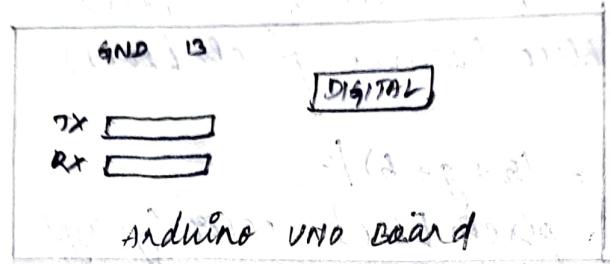
```
}
```

② Processing source Code -

```
import processing.serial.*;
```

```
Serial myPort; // Create object from Serial class
```

(a) Serial Reception:



Processing IDE - Output

Hello, world!

Hello, world!

Hello, world!

Hello, world!

Components Required

- ① Arduino Uno Board
- ② Battery / Power Supply Cable
- ③ Connecting Wires
as required

String val; // Variable to store the data received from the serial port

void setup ()

{

// On Windows machine, this generally opens in com1.

println (Serial.list());

String portName = serial.list()[2]; // Change the 0-201 or 2 etc., to match your port

myPort = new Serial (this, portName, 9600);

}

void draw ()

{

if (myPort.available () > 0)

{
val = myPort.readStringUntil ('\n'); // Read the value and store it in a variable "val"

}
println (val); // Print it out in the console

(b) Device Control using Button:

(i) Arduino source code:

// The setup function runs over once when you press reset or power the board:

void setup ()

{

pinMode (13, OUTPUT); // Set pin D13 (RED-LED) as output

Serial.begin (9600); // Start serial communications at 9,600 bits per second

}

```

// The loop function runs over and over again forever:
void loop()
{
    if (serial.available())
    {
        // If data is available to read now:
        char val = serial.read();
        if (val == '1')
        {
            digitalWrite (13, HIGH); // Turn ON Arduino LED
        }
        else if (val == '0')
        {
            digitalWrite (13, LOW); // Turn OFF Arduino LED
        }
    }
}

```

③ Processing source code

```

import controlP5.*;
import processing.serial.*;
Serial myPort;
ControlP5 cp5; // Create ControlP5 object
PFont font;

// same as setup in Arduino program:
void setup()
{
    size (300, 250); // Windows size (Width, Height)
    printArray (Serial.list()); // Prints all the available
                                // serial ports
    myPort = new Serial (this, "com7", 9600);
}

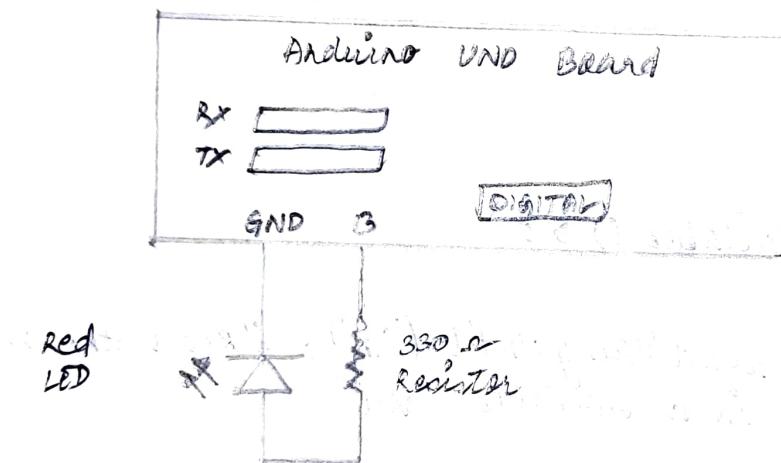
```

(b) Device Control using Button :-

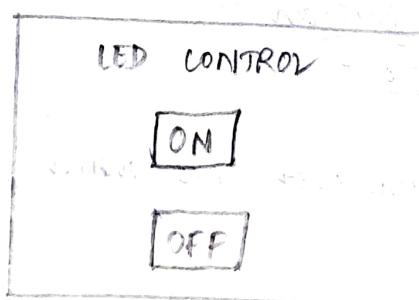
→ Components Required -

- ① Arduino UNO Board
- ② Red - LED
- ③ 330 Ω Resistor
- ④ Battery / Power Supply Cable
- ⑤ Connecting wires, as required

→ Circuit Diagram -



→ Processing IDE Output -



1	→ ON
0	→ OFF
1	→ ON
0	→ OFF
1	→ ON
0	→ OFF

```
cp5 = new ControlP5(this); // Add button to empty window  
font = createFont("calibri light bold", 20); // custom fonts  
for buttons and title
```

```
cp5.addButton("on") // "blue" is the name of button  
.setPosition(100, 50) // x and y coordinates of upper left corner  
.setSize(120, 70) // (width, height) of button  
.setFont(font);
```

```
cp5.addButton("off") // "OFF" is the name of button  
.setPosition(100, 150) // x and y coordinates of upper left corner  
.setSize(120, 70) // (width, height) of button  
.setFont(font);
```

}

// same as loop in Arduino Program:

```
void draw()
```

{

```
background(150, 0, 150); // Background color of window  
(r, g, b) or (0 to 255)
```

// Title to the window:

```
fill(0, 255, 0); Text color (r, g, b)
```

```
textFont(font);
```

```
text("LED CONTROL", 80, 30), // ("TEXT", x-coordinate,  
y-coordinate)
```

}

// Add functions to buttons: when the button is pressed, it
sends a particular char over serial port:

```
void on()
```

```
{ myPort.write('1');  
println("1");
```

}

```
void off ()  
{  
    myPort.write ('0');  
    printf ("0");  
}
```

(ii) Device Control using Mouse Click :-

(i) Arduino Source Code -

```
char val; // Variable to store the data received from the  
           serial port
```

```
int ledPin = 13; // LED Pin of Arduino
```

// The setup function runs once when you press reset or power
the board:

```
void setup ()
```

```
{
```

```
pinMode (ledPin, OUTPUT); // Set pin13 (RED-LED) as output  
Serial.begin (9600); // Start serial communications at 9,600  
                     bits per second
```

```
}
```

// The loop function runs over and over again forever:

```
void loop ()
```

```
{
```

```
if (serial.available ())
```

```
{
```

// If data is available to read now:

```
val = serial.read (); // Read the value and store it in a  
                     variable "val"
```

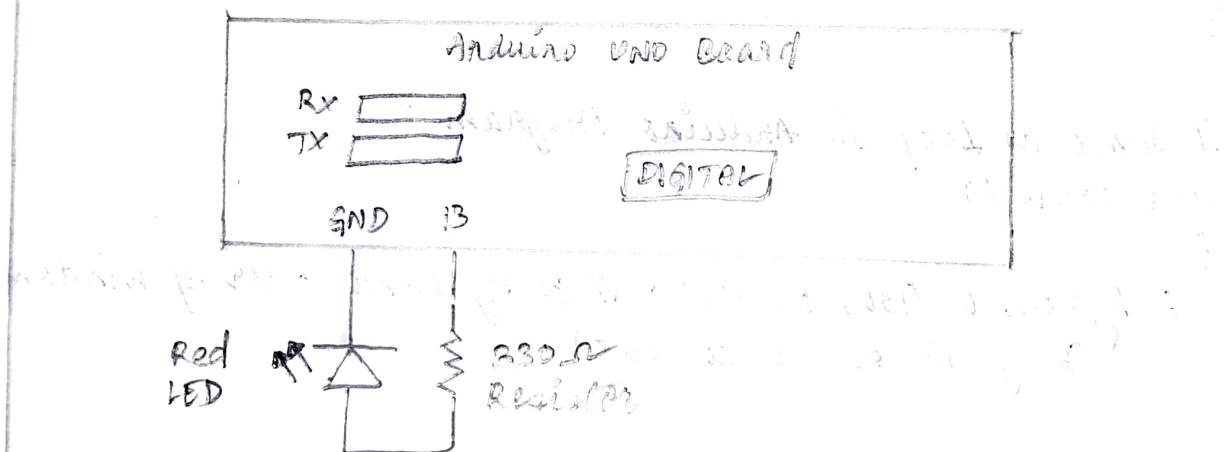
```
}
```

(c) Device Control using Mouse Clicks

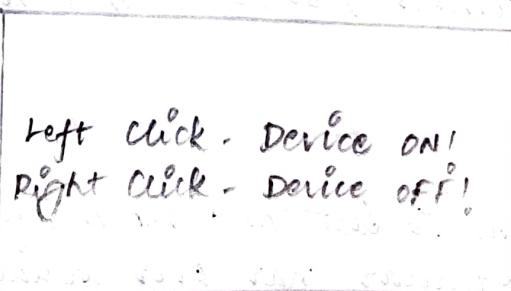
→ Components Required -

- ① Arduino UNO Board
- ② Red - LED
- ③ 330 Ω Resistor
- ④ Battery / Power supply cable
- ⑤ Connecting Wires, as required

→ Circuit Diagram -



→ Processing IDE Output -



1 → Left Click

0 → Right Click

1 → Left Click

0 → Right Click

```

if (val == '0')
{
    digitalWrite (ledPin, LOW); // Turn OFF Arduino LED
}
else if (val == '1')
{
    digitalWrite (ledPin, HIGH); // Turn ON Arduino LED
}
delay (10); // Delay in between reads for stability
}

```

② Processing source code

```

import processing.serial.*;
PFont f;
Serial myPort; // Create object from Serial class
String val; // Variable to store the data received from the
            // serial port

void setup()
{
    size (200, 200); // Output window size (width, height)
    f = createFont ("Arial", 16, true); // Create font
    String portName = serial.list () [2]; // Change the 0 to 1 or
                                         // 2 etc., to match your port
    myPort = new Serial (this, portName, 9600);
}

void draw()
{
    background (255); // Set background color
    textFont (f, 14); // Specify font to be used
    fill (0); // Specify font color
}

```

```

text ("Left click - Device ON!", 20, 100); // Display Text
text ("Right click - Device OFF!", 200, 100); // Display Text

if (mousePressed == true)
{
    // If clicked in the window:
    if (mouseButton == LEFT)
    {
        myPort.write ('1'); // Send a 1 (HIGH)
        println ("1");
    }
    else if (mouseButton == RIGHT)
    {
        myPort.write ('0'); // Send a 0 (LOW)
        println ("0");
    }
}

```

(d) Device Control using Keyboard-

① Arduino source code -

```

char val; // Variable to store the data received from the
          // serial port
int ledPin = 13; // LED Pin of Arduino

// The setup function runs once when you press reset or power
// the board:
void setup ()
{
    pinMode (ledPin, OUTPUT); // Set pin D8 (RED - LED) as output
    serial.begin (9600); // Start serial communications at 9,600
                         // bits per second
}

```

// The loop function runs over and over again forever:

```

void loop () {
    if (serial.available ()) {
        // If data is available to read now:
        val = serial.read (); // Read the value and store it in
        // a variable "val"
    }

    if (val == '1') {
        digitalWrite (ledPin, HIGH); // Turn ON Arduino LED
    } else if (val == '0') {
        digitalWrite (ledPin, LOW); // Turn OFF Arduino LED
    }

    delay (10); // Delay in between reads for stability
}

```

③ Processing source code-

```

import processing.serial.*;
PFont f; // declare PFont variable
Serial myPort; // Create object from Serial class
String val; // Variable to store the data received from the
// serial port

```

```
void setup ()
```

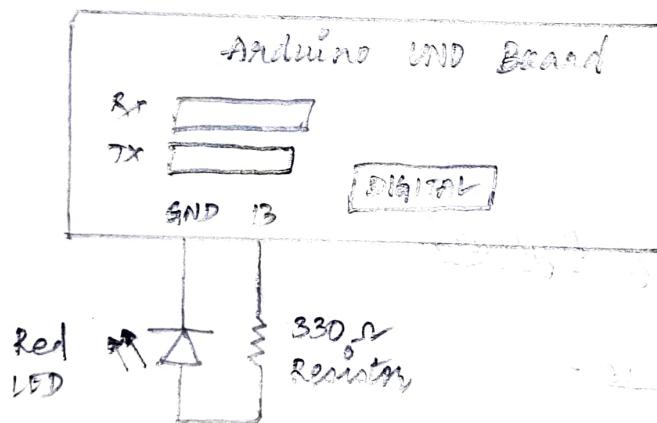
```
{
    size (220, 200); // Output Window size (width, height)
    f = createFont ("Arial", 16, true); // Create font
}
```

(d) Device Control using Keyboard -

→ Components Required -

- ① Arduino UNO Board
- ② Red LED
- ③ 330 Ω Resistor
- ④ Battery / Power supply cable
- ⑤ Connecting wires, as required

→ Circuit Diagram -



→ Processing IDE Output -

'A' in Keyboard - Device ON!
'B' in Keyboard - Device OFF!

1 → 'A' in Keyboard

0 → 'B' in Keyboard

1 → 'A' in Keyboard

```
String portName = serial.list()[2]; // Change the 0 to 1 or 2 etc  
to match your port  
myPort = new Serial(this, portName, 9600);  
}
```

```
void draw()  
{  
background(255); // Set background color to black  
textFont(f, 14); // Specify font to be used  
fill(0); // Specify font color  
text("A" in Keyboard - Device ON!, 20, 100); // Display Text  
text("B" in Keyboard - Device OFF!, 20, 120); // Display Text  
}
```

```
void keyPressed()  
{  
if (key == 'A' || key == 'a')  
{  
myPort.write('1'); // Send a 1 (HIGH)  
println("1");  
}  
else if (key == 'B' || key == 'b')  
{  
myPort.write('0'); // Send a 0 (LOW)  
println("0");  
}  
}
```

(e) Servo Motor Control using Knob -

① Arduino Source Code -

```
#include <Servo.h>  
Servo myservo; // Create servo object to control a servo
```

```
int servoPin = 2; // Connect yellow servo wire to a digital GPIO  
pin 2 (D2 of Arduino), must be PWM (Pulse Width Modulation)  
int val = 0; // Variable to store the data received from the  
serial port
```

// The setup function runs once when you press reset or power the board:

```
void setup ()  
{
```

```
myservo.attach (servoPin); // Attach the servo to the PWM pin  
serial.begin (9600); // serial communications start at 9,600  
bits per second
```

```
}
```

// The loop function runs over and over again forever:

```
void loop ()  
{
```

```
if (serial.available ())  
{
```

// If data is available to read now:

```
val = serial.read (); // Read the value and store it in a  
variable "val"
```

```
}
```

```
myservo.write (val); // Set the servo position
```

```
delay (15); // Wait for the servo to get there
```

```
}
```

② Processing source code -

```
import processing.serial.*;
```

```
Serial myPort; // Create object from serial class
```

```
import controlP5.*; // Import controlP5 library
```

(c) Servo Motor Control using Knob:

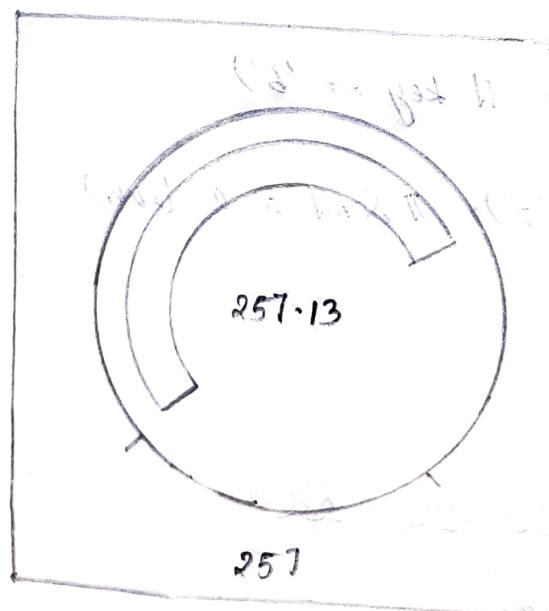
→ Components Required -

- ① Arduino UNO Motor
- ② Servo Motor
- ③ Battery / Power Supply Cable
- ④ Connecting Wires, as required

→ Circuit Diagram -



→ Processing IDE output -



Control event from : knob1, value : 257

```
ControlP5 controlP5; // ControlP5 object
PFont f; // Declare PFont variable

void setup()
{
    size(200, 200); // Output Window size (width, height)
    String portName = serial.list()[2]; // Change the 0 to 1 or 2
    // etc., to match your port
    myPort = new Serial(this, portName, 9600);

    smooth();
    f = createFont("Arial", 16, true); // Create Font
    controlP5 = new ControlP5(this);

    // Description: Round turning dial knob
    // Parameters: name, minimum, maximum, default value [float]
    // x, y, diameter
    controlP5.addKnob("knob1", 0, 360, 80, 50, 40, 100);
}

void draw()
{
    background(255); // Set background to black
    textFont(f, 16); // Specify font to be used
    fill(0); // Specify font color
    text("Knob", 310, 250);
    text(int(controlP5.getController("knob1").getValue()), 90, 190);
}

void controlEvent(ControlEvent theEvent)
{
    if (theEvent.isController())
    {
```

```
print ("control event from: " + theEvent.getController().  
      getName());  
println (" value: " + int(theEvent.getController().getValue())  
      );  
myPort.write (int (theEvent.getController().getValue()));  
}  
}  
}
```

* INFERENCES:

Designed and developed embedded systems using Arduino and Processing IDE and all simulation results were verified successfully.