

# ARDUINO CONTROLLER PROGRAMMING & ITS APPLICATIONS

---

## Day-Three

Zuhaib A. Shaikh,  
Asst. Prof., CSE Deptt.,QUEST  
Web: [zuhaib-shaikh.neocities.org](http://zuhaib-shaikh.neocities.org)

# Outline

Day	Activity
1	i. Introduction to Arduino development board and Arduino IDE ii. C/C++ language overview iii. Basic input / output with Arduino iv. Overview of Proteus simulation software for Arduino simulation
2	i. Interfacing and glowing LEDs with different pattern ii. Interfacing push button and piezo buzzer iii. Interfacing a temperature sensor with Arduino
3	i. Familiarization with Serial Monitor for input and for output ii. Interfacing LDR sensor with Arduino iii. Interfacing PIR motion sensor with Arduino
4	i. Interfacing Arduino with LCD (16x2), relay and Servo motor ii. Interface Arduino with Sonic Sensor for obstacle detection
5	i. Interfacing shift register and 7-segment display with Arduino ii. Interfacing HC-05 Bluetooth module with Arduino iii. Driving GSM modem with Arduino
Prerequisites: <ul style="list-style-type: none"> <li>- Knowledge of C++</li> <li>- Knowledge of basic electronic components</li> </ul>	

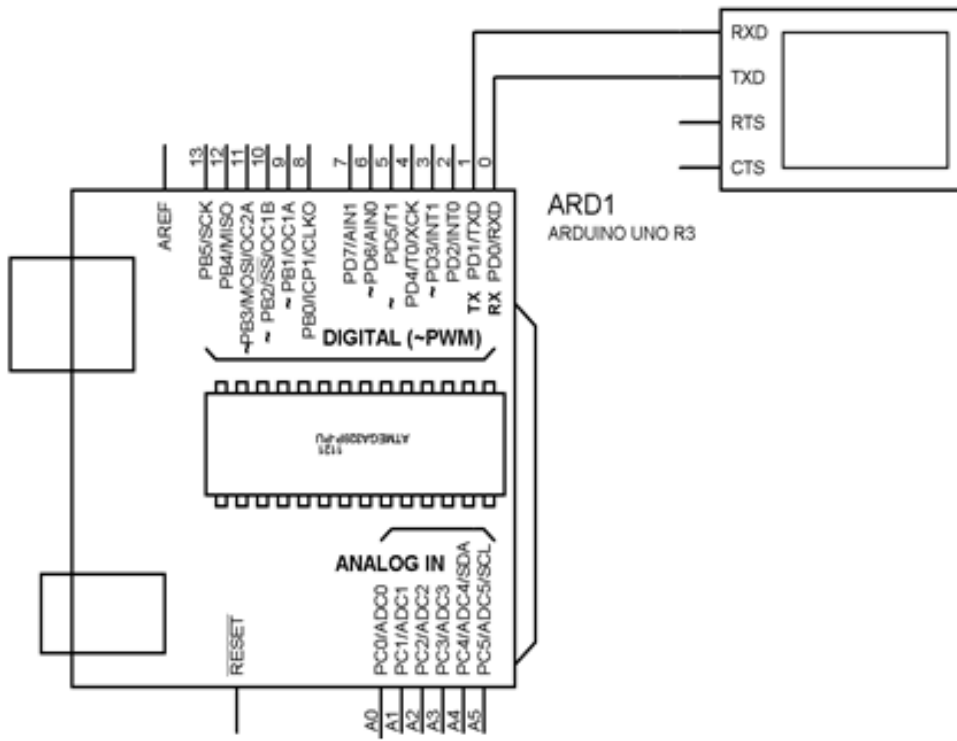
# Serial Monitor

- Serial Monitor allows to communicate with Arduino (if connected via USB cable)
- In Proteus, Virtual Terminal plays same role
- By default, Arduino Digital Pin 0 and 1 are reserved for Serial Monitor
- If any device is connected to these pins, then serial monitor is inaccessible
- Otherwise, other digital pins can be used for serial communication using SoftwareSerial library (except **pin 2 and 3** in [Arduino Mega](#))
- Basic functions for Serial Monitor:
  - **Serial.begin(*baud\_rate*)** – To set serial baud rate for serial monitor
  - Output functions:
    - **Serial.print(*string\_and/or\_variable*)** – prints string (enclosed in double quotation) and/or variable value on serial monitor
    - **serial.println(*string\_and/or\_variable*)** – prints string (enclosed in double quotation) and/or variable value on serial monitor with new line
  - Input functions:
    - **Serial.available( )** – Returns TRUE if data is available on serial monitor (from Arduino), else returns FALSE
    - **Serial.read( )** – Returns first byte from serial buffer. It is necessary to cast the returned byte data into desired format using explicit casting functions

# Example 1: Communicating with Serial Monitor

- In virtual terminal, data can be inputted by enabling “Echo, Typed Characters” option
  - Right-click on virtual terminal window and click on “Echo, Typed Characters” check
- Tx pin of Arduino is connected with Rx pin of any serial communication device (e.g. virtual terminal) and Rx of Arduino with Tx of serial device

Circuit



Code

```
String str;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  Str=" ";
  Serial.print("Enter any value: ");
  while(Serial.available())
    str+=(char) Serial.read();
  Serial.print("you have entered: ");
  Serial.println(str.toInt());
  delay(10000);
}
```

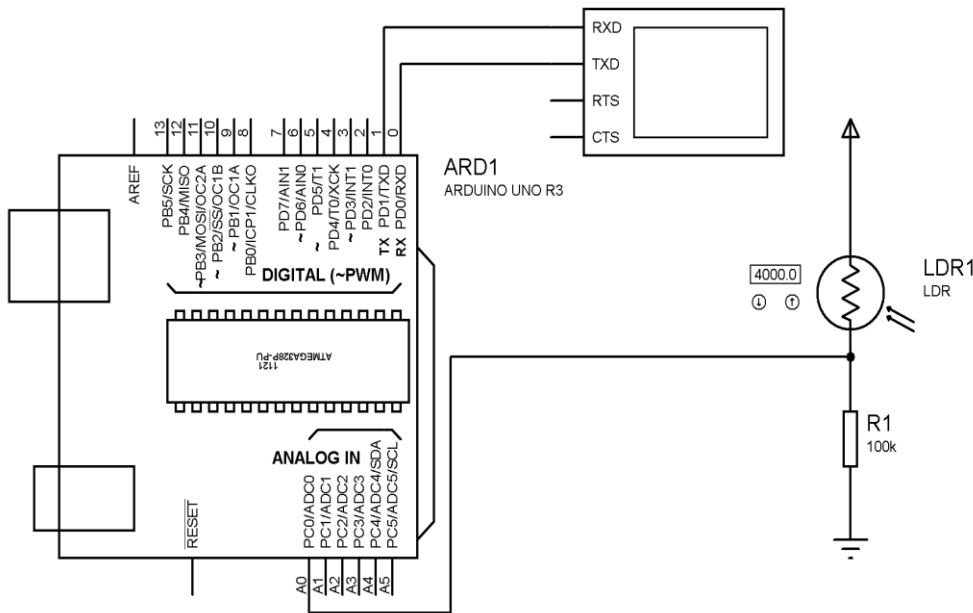
# Example 2: LDR

- Bilateral component
- Resistance is changed on change of light intensity
- Resistance is inversely proportional to light intensity in LDR
  - Less precise, thus can be used for street light control or enclosure detection



Circuit

Code



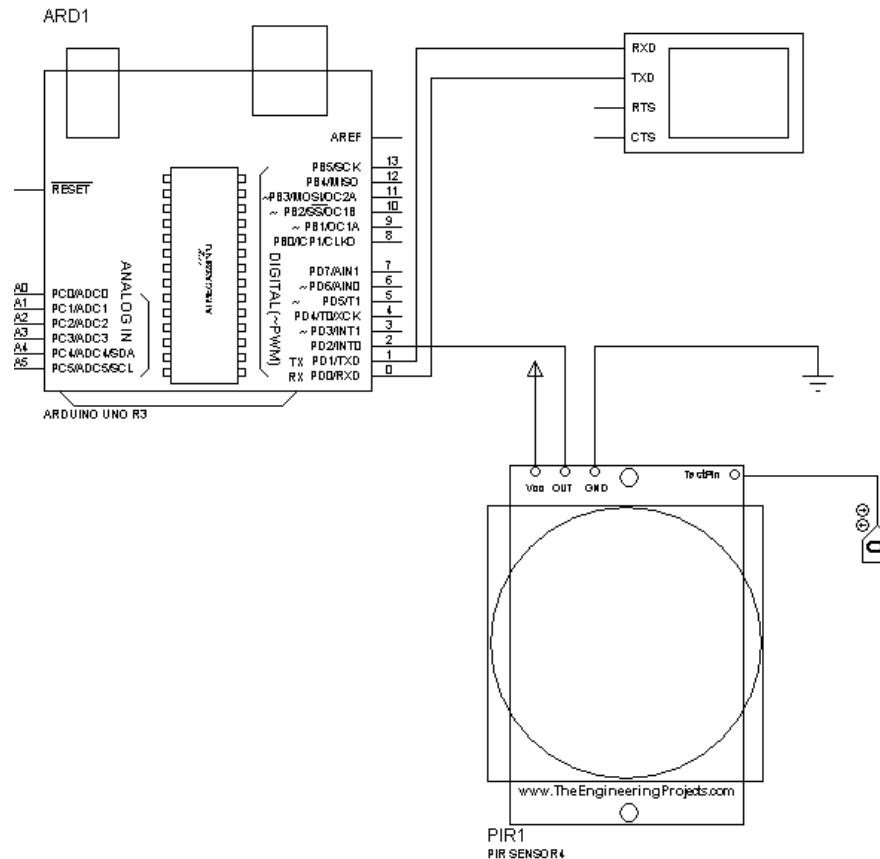
```
int sensorPin = A0;
int sensorValue = 0;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  sensorValue = analogRead(sensorPin);
  Serial.println(sensorValue);
  delay(100);
}
```

# Example 3: PIR sensor for Motion Detection

- Passive Infra Red sensor
- Detects IR radiations of objects in its field of view
- Provides HIGH if receives IR waves else provides LOW



## Circuit



# Example 3: PIR sensor for Motion Detection

## Code

```
#define PIRpin 2
int count;
void setup()
{
  pinMode(PIRpin,INPUT);
  Serial.begin(9600);
}
void loop() {
  count = 0;
  if(digitalRead(PIRpin)==HIGH)
  {
    Serial.println("Motion Detected");
    while(digitalRead(PIRpin))
    {
      count++;
      delay(10);
    }
    if(count>10)
      Serial.println("Montion Ended");
  }
}
```

# Questions

