

Unit IV – Notes

Arduino:

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board. An Arduino is an open hardware development board that can be used by tinkerers, hobbyists, and makers to design and build devices that interact with the real world.

Arduino board consists of:

- A number of pins, which are used to connect with various components you might want to use with the Arduino. These pins come in two varieties:
- Digital pins, which can read and write a single state, on or off. Most Arduinos have 14 digital I/O pins.
- Analog pins, which can read a range of values, and are useful for more fine-grained control. Most Arduinos have six of these analog pins.

Arduino IDE:

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module. It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process. It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment. A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more.

Writing Sketches

Programs written using Arduino Software (IDE) are called sketches. These sketches are written in the text editor and are saved with the file extension .ino. The editor has features for cutting/pasting and for searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom righthand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

Some icons are as folloes:



Verify

Checks your code for errors compiling it.



Upload

Compiles your code and uploads it to the configured board.

See [uploading](#) below for details.

Note: If you are using an external programmer with your board, you can hold down the "shift" key on your computer when using this icon. The text will change to "Upload using Programmer"



New

Creates a new sketch.



Open

Presents a menu of all the sketches in your sketchbook. Clicking one will open it within the current window overwriting its content.

Note: due to a bug in Java, this menu doesn't scroll; if you need to open a sketch late in the list, use the File | Sketchbook menu instead.



Save

Saves your sketch.



Serial Monitor

Opens the [serial monitor](#).

Sensor

A sensor is a device that detects and responds to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena.

A sensor is a device used to measure a property, such as pressure, position, temperature, or acceleration, and respond with feedback.

Here are a few examples of the many different types of sensors:

In a mercury-based glass thermometer, the input is temperature. The liquid contained expands and contracts in response, causing the level to be higher or lower on the marked gauge, which is human-readable.

An oxygen sensor in a car's emission control system detects the gasoline/oxygen ratio, usually through a chemical reaction that generates a voltage. A computer in the engine reads the voltage and, if the mixture is not optimal, readjusts the balance.

Motion sensors in various systems including home security lights, automatic doors and bathroom fixtures typically send out some type of energy, such as microwaves, ultrasonic waves or light beams and detect when the flow of energy is interrupted by something entering its path.

A photosensor detects the presence of visible light, infrared transmission (IR), and/or ultraviolet (UV) energy.

Criteria to choose a Sensor

There are certain features which have to be considered when we choose a sensor. They are as given below:

1. Accuracy
2. Environmental condition – usually has limits for temperature/ humidity
3. Range – Measurement limit of sensor
4. Calibration – Essential for most of the measuring devices as the readings changes with time
5. Resolution – Smallest increment detected by the sensor
6. Cost
7. Repeatability – The reading that varies is repeatedly measured under the same environment

Classification of Sensors

The sensors are classified into the following criteria:

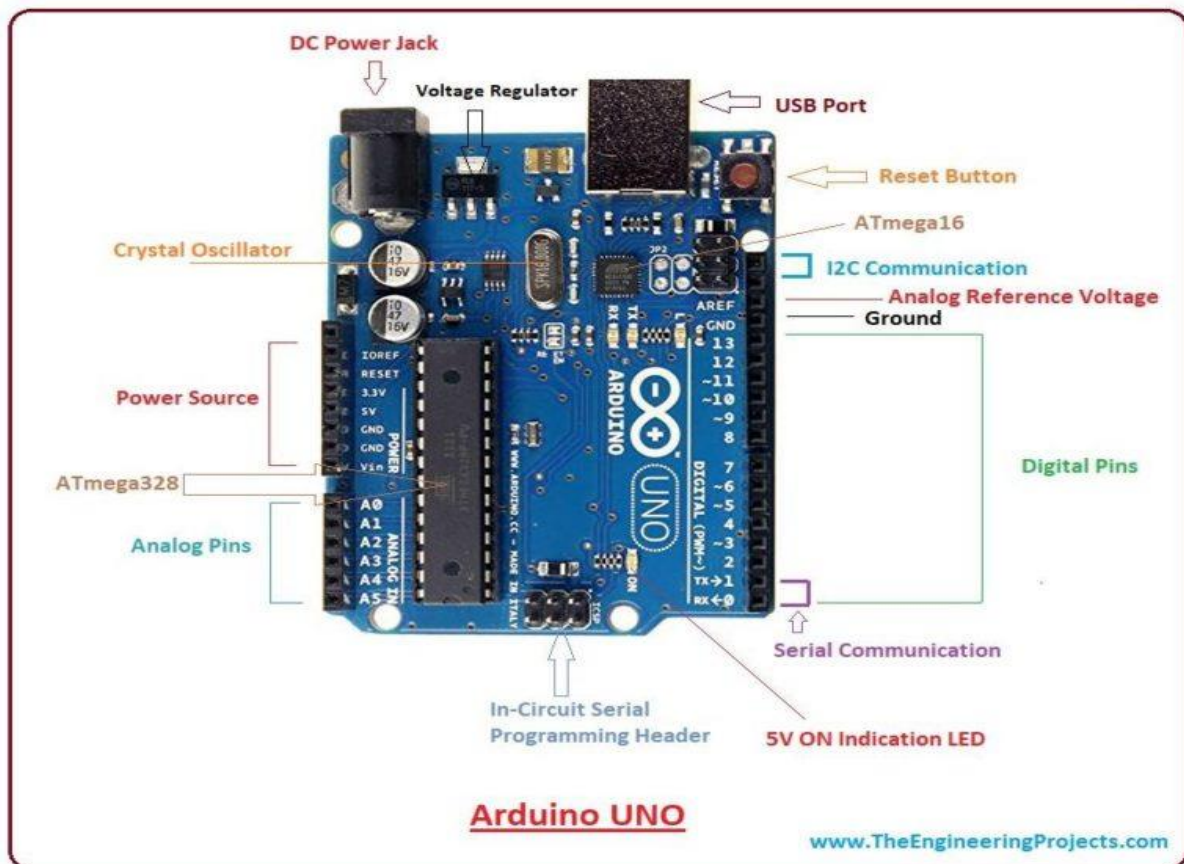
1. Primary Input quantity (Measurand)
2. Transduction principles (Using physical and chemical effects)
3. Material and Technology

4. Property
5. Application

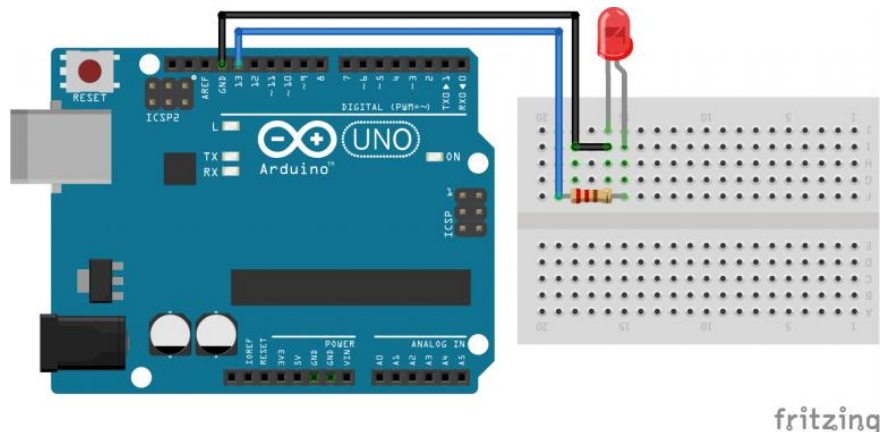
Classification based on property is as given below:

- Temperature – Thermistors, thermocouples, RTD's, IC and many more.
- Pressure – Fibre optic, vacuum, elastic liquid based manometers, LVDT, electronic.
- Flow – Electromagnetic, differential pressure, positional displacement, thermal mass, etc.
- Level Sensors – Differential pressure, ultrasonic radio frequency, radar, thermal displacement, etc.
- Proximity and displacement – LVDT, photoelectric, capacitive, magnetic, ultrasonic.
- Biosensors – Resonant mirror, electrochemical, surface Plasmon resonance, Light addressable potentiometric.
- Image – Charge coupled devices, CMOS
- Gas and chemical – Semiconductor, Infrared, Conductance, Electrochemical.
- Acceleration – Gyroscopes, Accelerometers.
- Others – Moisture, humidity sensor, Speed sensor, mass, Tilt sensor, force, viscosity.

Arduino Uno Photo:



Interfacing LED with Arduino



Arduino UNO has built in led available at pin no. 13. This built in led is very useful when you want to show some signal without connecting external led.

If you want to learn how to connect led to arduino. We can connect led externally by using the resistor in between the digital pin of arduino to the led. While connecting the led with the arduino make sure you know the resistor value required. We will also discuss how to calculate the led resistance. After finding the correct value resistor we can interface the led with arduino.

For a single led, we need one digital output from the arduino. We use the digital pin 13 for interfacing the led with arduino. We will connect the led to forward bias. Means LED's anode to the pin no. 13 and cathode to the GND of the arduino.

When we are using external Led to the arduino board, we should connect the proper value resistor to the led to limit the current coming from the arduino uno pin. A led requires around 10-20mA of current but arduino pin gives around 40mA, so due to excess current led may damage. For protecting the led we need a current limiting resistor.

You should try this programs to see what happen with the led blinking.

```
void setup() {  
  
    // initialize digital pin 13 as an output.  
  
    pinMode(13, OUTPUT);  
  
}  
  
// the loop function runs over and over again forever  
  
void loop() {  
  
    digitalWrite(13, HIGH);    // turn the LED on  
  
    digitalWrite(13, LOW);     // turn the LED off
```

```
}
```

Button:

Pushbuttons or switches connect two points in a circuit when you press them. This example turns on the built-in LED on pin 13 when you press the button.

Arduino Sketch

```
const int switchPin = 4; //Switch Connected to PIN 4

const int ledPin = 8; //LED Connected to PIN 8

int switchState = 0; //Variable for reading switch status

void setup()
{
    pinMode(ledPin, OUTPUT); //LED PIN is Output
    pinMode(switchPin, INPUT_PULLUP); //Switch PIN is Input
}

void loop()
{
    switchState = digitalRead(switchPin); //Read the status of the Switch
    if (switchState == LOW) //If the switch is pressed
    {
        digitalWrite(ledPin, HIGH); //LED ON
        delay(3000); //3 Second Delay
        digitalWrite(ledPin, LOW); //LED OFF
    }
}
```

DHT 11 sensor:

The DHT-11 is a digital-output relative humidity and temperature sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin. In this tutorial you will learn how to use this sensor with Arduino uno.

LDR:

An LDR is a component that has a (variable) resistance that changes with the light intensity that falls upon it. This allows them to be used in light sensing circuits.

MQ – 135:

MQ-135 is a gas sensor which is sensitive to CO₂. The change in CO₂ levels changes the resistance of sensor which results in the change in the output of the sensor. The output generated from the sensor is feed to Analog Input to Arduino board (In my case Arduino Uno).

LCD:

A liquid crystal display or LCD draws its definition from its name itself. It is combination of two states of matter, the solid and the liquid. LCD uses a liquid crystal to produce a visible image. Liquid crystal displays are super-thin technology display screen that are generally used in laptop computer screen, TVs, cell phones and portable video games. LCD's technologies allow displays to be much thinner when compared to cathode ray tube (CRT) technology.

Liquid crystal display is composed of several layers which include two polarized panel filters and electrodes. LCD technology is used for displaying the image in notebook or some other electronic devices like mini computers. Light is projected from a lens on a layer of liquid crystal. This combination of colored light with the grayscale image of the crystal (formed as electric current flows through the crystal) forms the colored image. This image is then displayed on the screen.

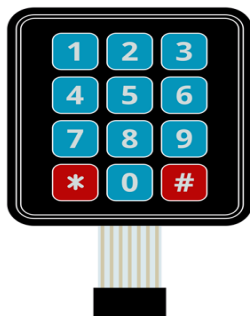
An LCD is either made up of an active matrix display grid or a passive display grid. Most of the Smartphone's with LCD display technology uses active matrix display, but some of the older displays still make use of the passive display grid designs. Most of the electronic devices mainly depend on liquid crystal display technology for their display. The liquid has a unique advantage of having low power consumption than the LED or cathode ray tube.

Liquid crystal display screen works on the principle of blocking light rather than emitting light. LCD's requires backlight as they do not emits light by them. We always use devices which are made up of LCD's displays which are replacing the use of cathode ray tube. Cathode ray tube draws more power compared to LCD's and are also heavier and bigger.

Working of LCD:

LCD displays don't just look different than bulky CRT monitors, the way they operate is significantly different as well. Instead of firing electrons at a glass screen, an LCD has backlight that provides light to individual pixels arranged in a rectangular grid. Each pixel has a red, green, and blue RGB sub-pixel that can be turned on or off. When all of a pixel's sub-pixels are turned off, it appears black. When all the sub-pixels are turned on 100%, it appears white. By adjusting the individual levels of red, green, and blue light, millions of color combinations are possible.

Keypad:



A keypad is one of the most commonly used input devices in microprocessor applications. In a standard keypad wired as an X-Y switch matrix, normally-open switches connect a row to a column when pressed. If a keypad has 12 keys, it is wired as 3 columns by 4 rows. A 16 key pad would have 4 columns by 4 rows. Some time ago, I bought a couple of 3x4 membrane keypads from eBay. As usual it's packed with zero documentation, thus it took couple of hours to get to work.

Bluetooth:

HC-05 is a Bluetooth device used for wireless communication with Bluetooth enabled devices (like smartphone). It communicates with microcontrollers using serial communication (USART).

Default settings of HC-05 Bluetooth module can be changed using certain AT commands.

As HC-05 Bluetooth module has 3.3 V level for RX/TX and microcontroller can detect 3.3 V level, so, there is no need to shift TX voltage level of HC-05 module. But we need to shift the transmit voltage level from microcontroller to RX of HC-05 module.

```

/* Bluetoooh Basic: LED ON OFF - Avishkar

* Coder - Mayoogh Girish

* Website - http://bit.do/Avishkar

* Download the App : https://github.com/Mayoogh/Arduino-Bluetooth-Basic

* This program lets you to control a LED on pin 13 of arduino using a bluetooth module

*/

char data = 0; //Variable for storing received data

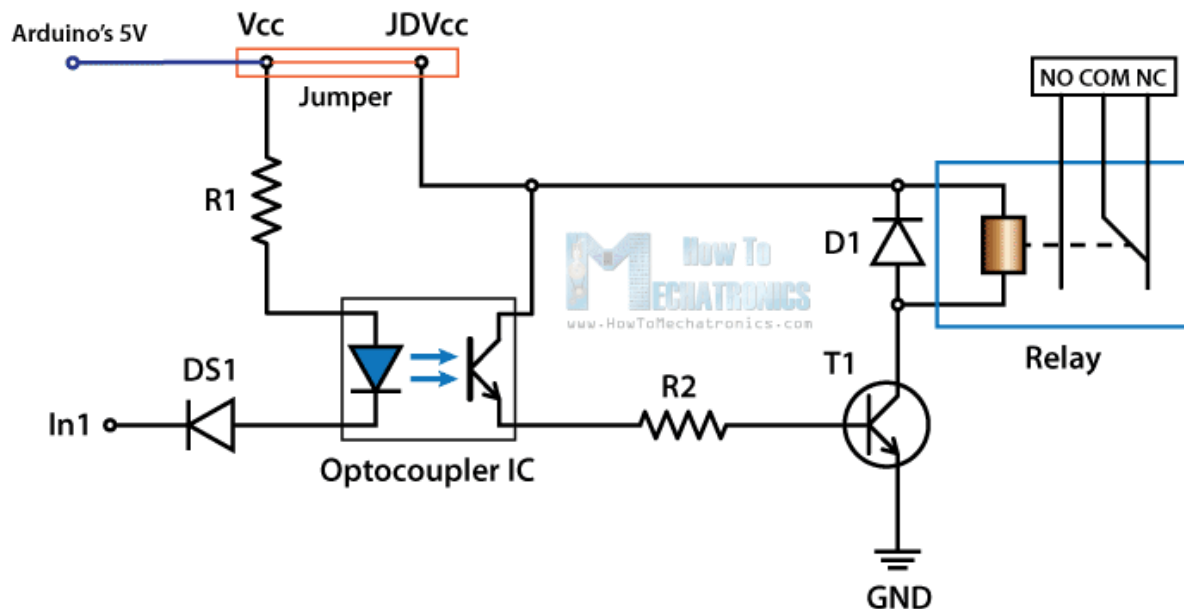
void setup()
{
    Serial.begin(9600); //Sets the baud for serial data transmission
    pinMode(13, OUTPUT); //Sets digital pin 13 as output pin
}

void loop()
{
    if(Serial.available() > 0) // Send data only when you receive data:
    {
        data = Serial.read();    //Read the incoming data and store it into variable data
        Serial.print(data);      //Print Value inside data in Serial monitor
        Serial.print("\n");      //New line
        if(data == '1')          // Checks whether value of data is equal to 1
            digitalWrite(13, HIGH); //If value is 1 then LED turns ON
        else if(data == '0')      // Checks whether value of data is equal to 0
            digitalWrite(13, LOW); //If value is 0 then LED turns OFF
    }
}

```

Relay:

We can control high voltage electronic devices using relays. A Relay is actually a switch which is electrically operated by an electromagnet. The electromagnet is activated with a low voltage, for example 5 volts from a microcontroller and it pulls a contact to make or break a high voltage circuit.



For better understanding let's see the circuit schematics of the relay module in this configuration. So we can see that the 5 volts from our microcontroller connected to the Vcc pin for activating the relay through the Optocoupler IC are also connected to the JDVcc pin which powers the electromagnet of the relay. So in this case we got no isolation between the relay and the microcontroller.

