Circuit Design and Analysis II
Final Technical Report

# Interfacing LED using Joystick Controller with Arduino Uno

ENR -202-01

Date of submission : 12/09/2019
Member: Hao Hoang Nguyen

# ***Problem Statement***

Throughout the history of video games, one of the most important aspects of console video games is the controller. This piece of hardware is the one with which the players interact the most and is by far the most memorable component.

The design of a video game controller should not be taken lightly. Aside from removing the barrier between the player and the virtual environment, the controller also specifies the type of experience the player will have by defining what types of games are best played on it due to its design.

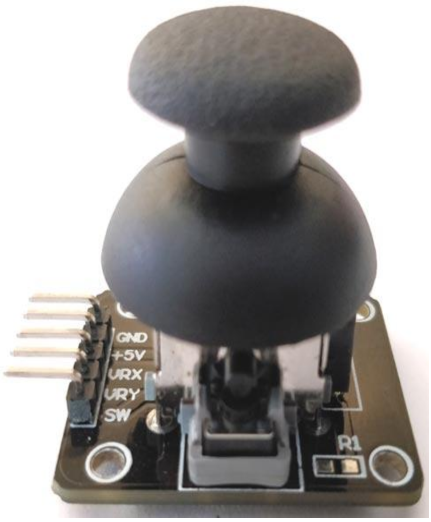
In this project, we came up with the simplest video game controller " Joystick Game Controller". Apart from gaming, this " Joystick Controller has many other applications in industrial electronics. Basically, this Joystick is nothing but a combination of the two potentiometers for X and Y direction. It reads the voltage through the potentiometer and give analog value to the board; and the analog values would be changed as the movement of the Joystick shaft changed.

In this circuit, we are interfacing Joystick with Arduino Uno board simply by controlling 5 LEDs as the movement of the joystick and switch button on the joystick. We have placed 4 LEDs in such a way that it represents the direction of the Joystick shaft movement. A single LED is connected to the switch , as the joystick is pressed, that LED will TURN ON. In this project, some materials is required. Arduino Uno, 5 LEDs , Joystick Module, 330 Ohm Resistors, Breadboard, and Connecting Wires.

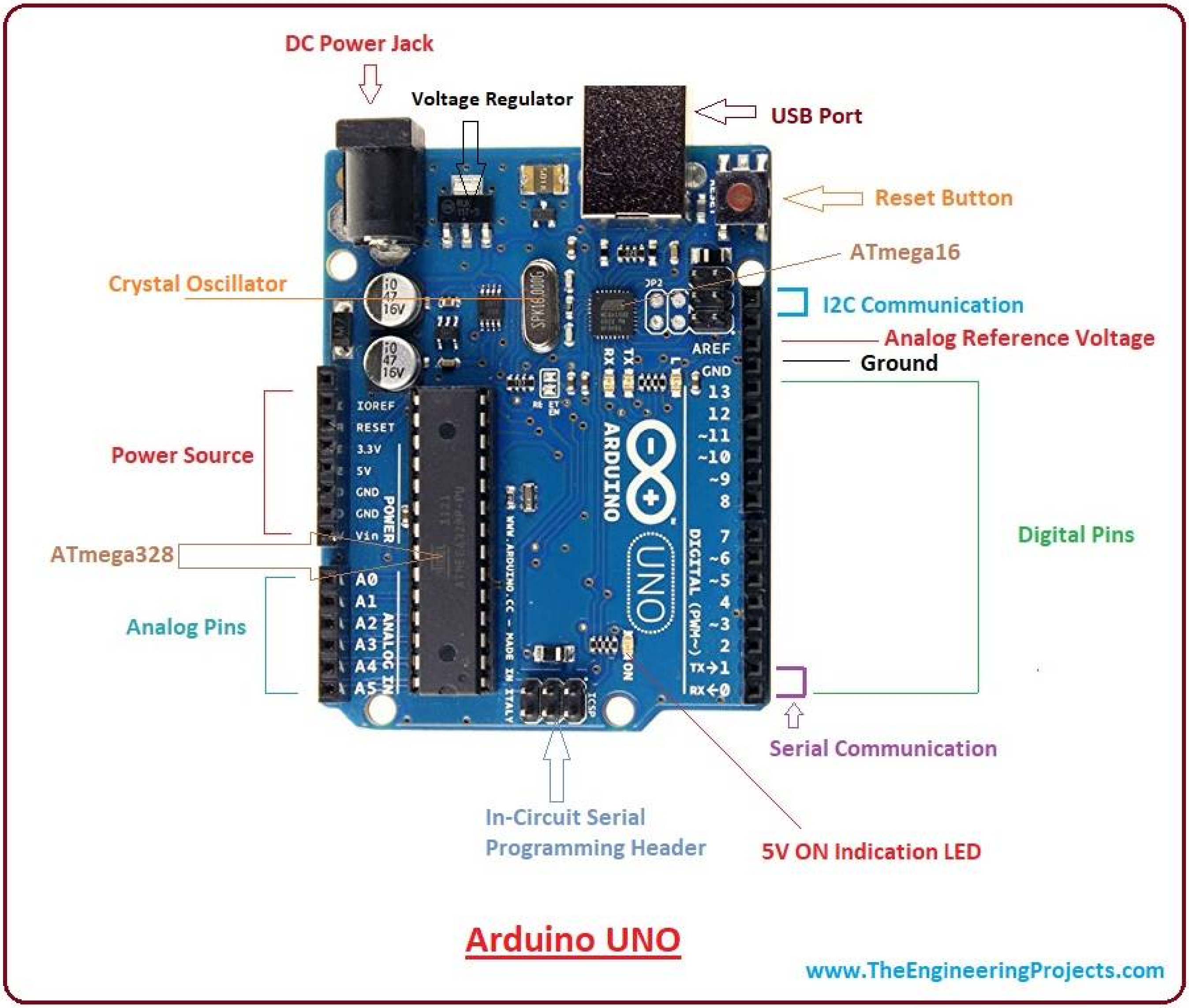
# ***Narrative***

## Joystick Module

Joysticks are in many different shapes and sizes. The joystick that was used in this project which is showing below, provides Analog outputs and output voltages provided by this module keep changing according to the direction in which we move the joystick. This joystick has 2 axes as X-axis and Y-axis. Each axis of the joystick is mounted to a potentiometer. The mid points of these pots are driven out as Rx and Ry. So Rx and Ry are variable Points. When the joystick standby, Rx and Ry act as a voltage divider. When the joystick is moving along the horizontal axis, the voltage at Rx pin changes. Similarly, when the joystick is moving along the vertical axis, the voltage at Ry pin will change. So we have totally 4 directions of joystick on 2 ADC outputs. When the stick is moved, the voltage on each pin goes high or low depending on direction.



## Arduino UNO


DC Power Jack
Voltage Regulator
USB Port
Reset Button
ATmega16
Crystal Oscillator
I2C Communication
Analog Reference Voltage
Ground
Power Source
Digital Pins
ATmega328
Analog Pins
Serial Communication
In-Circuit Serial
Programming Header
5V ON Indication LED
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
ARDUINO
UNO
ICSP
ON
TX
RX
Arduino UNO
www.TheEngineeringProjects.com

## What is Arduino UNO ?

Arduino UNO is a microcontroller board developed by Arduino.cc which is an open-source electronics platform mainly based on AVR microcontroller Atmega328. This board comes with all the features required to run the controller and can be directly connected to the computer through USB cable that is used to transfer the code to the controller using IDE software, mainly developed to program Arduino.

Atmega328 microcontroller is placed on the board that comes with a number of features like timers, counters, interrupts, PWM, CPU, I/O pins and based on a 16MHz clock that helps in producing more frequency and number of instructions per cycle.It is an open source platform where anyone can modify and optimize the board based on the number of instructions and task they want to achieve.

This board comes with a built-in regulation feature which keeps the voltage under control when the device is connected to the external device. Reset pin is added in the board that reset the whole board and takes the running program in the initial stage. This pin is useful when board hangs up in the middle of the running program; pushing this pin will clear everything up in the program and starts the program right from the beginning. There are 14 I/O digital and 6 analog pins incorporated in the board that allows the external connection with any circuit with the board. These pins provide the flexibility and ease of use to the external devices that can be connected through these pins. There is no hard and fast interface required to connect the devices to the board. Simply plug the external device into the pins of the board that are laid out on the board in the form of the header. The 6 analog pins are marked as A0 to A5 and come with a resolution of 10bits. These pins measure from 0 to 5V, however, they can be configured to the high range using analogReference() function and AREF pin. 13KB of flash memory is used to store the number of instructions in the form of code. Only 5 V is required to turn on the board, which can be achieved directly using USB port or external adapter, however, it can support external power source up to 12 V which can be regulated and limited to 5 V or 3.3 V based on the requirement of the project.

## Application

Arduino Uno comes with a wide range of applications. A larger number of people are using Arduino boards for developing sensors and instruments that are used in scientific research. Following are some main applications of the board.

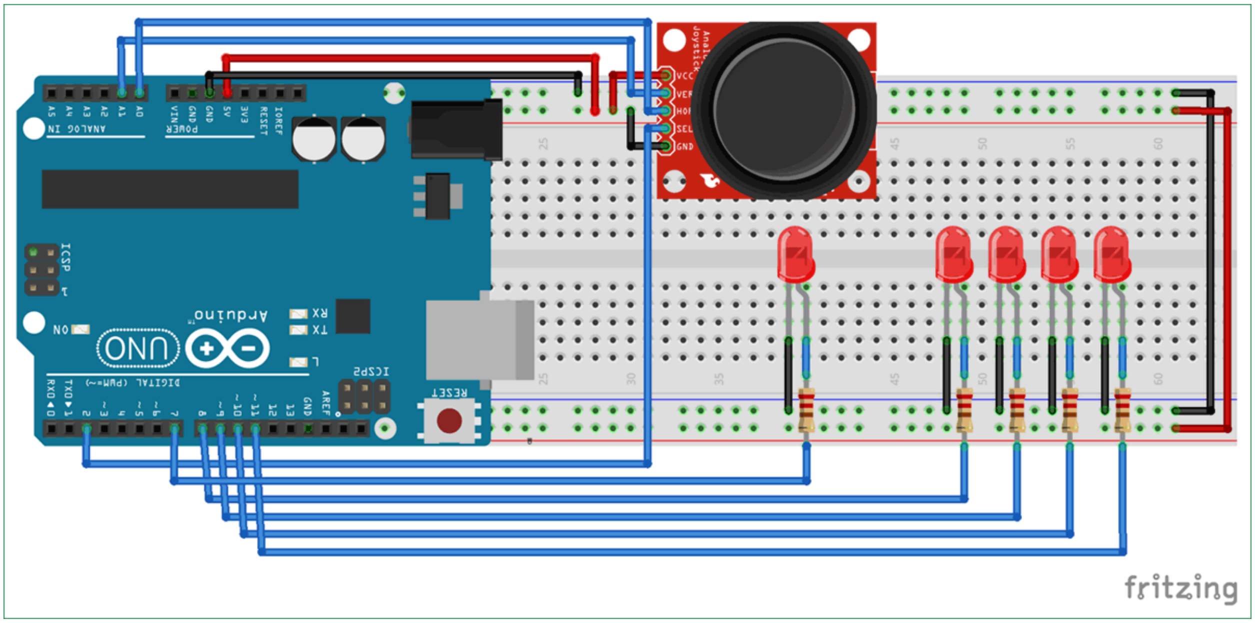
1. Security and defense system
2. Digital Electronic and Robotic
3. Parking Lot counter
4. Traffic light countdown timer
5. Medical Instrument
6. Industrial automation

## History of Arduino UNO

Arduino is an open-source hardware and software company, project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices. The Arduino project started in 2005 as a program for students at the Interaction Design Institute Ivrea in Ivrea, Italy. The purpose was to provide a low-cost and easy way for beginners and professionals to create devices that interact with their environment using sensors. Arduino project was invented by David Cuartielles, Gianluca Martino, Tom Igoe, David Mellis, and Massimo Banzi.

## Technical Analysis

In the system that we have been working on, Joystick Module is defined as an Input that sends Analog signal to the Arduino UNO, so the VRx and VRy pins of the Joystick module have to be connected to the analog input Pin A0 and A1; and the Switch pin has to be connected to Pin A2 on the Arduino. In this case, Arduino UNO board is defined as a receiver and ADC ( analog-Digital converter), which receives the Analog signal from the Joystick Module and convert the signal from Analog to Digital. After the signal goes through the process of the board, the Arduino will produce the output signals which are connected from pin A7 to A11 on the Arduino to LEDs. All of the connection in the system are shown in the Diagram below.



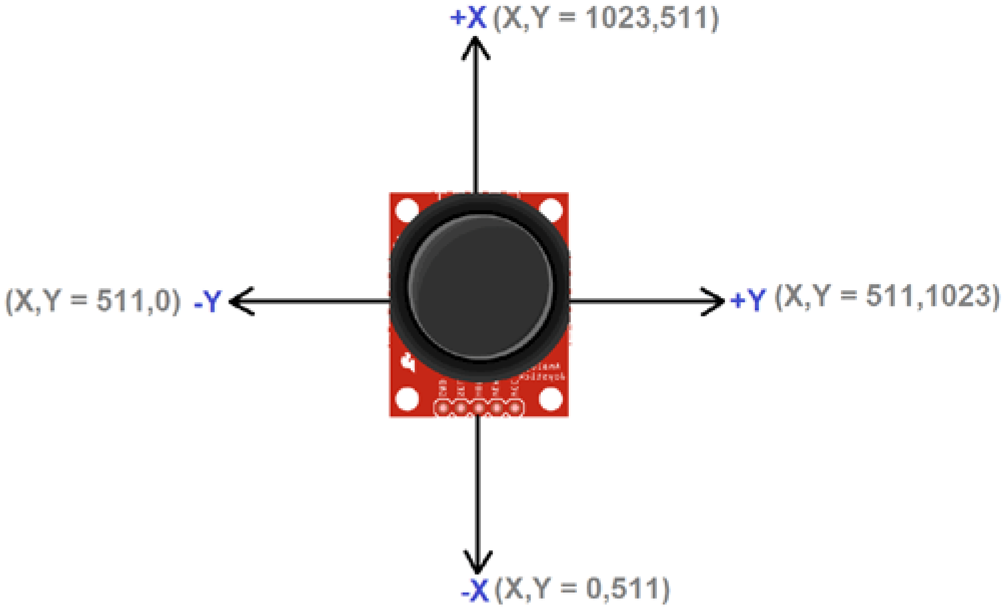
However, to the system to work, the board needs to be programming and uploaded.

## Coding Explanation.

```
#define joyX A0
#define joyY A1
int button=2;
int buttonState = 0;
int buttonState1 = 0;
void setup() {
  pinMode(7,OUTPUT);
pinMode(8,OUTPUT);
pinMode(9,OUTPUT);
pinMode(10,OUTPUT);
pinMode(11,OUTPUT);
  pinMode(button,INPUT);
  digitalWrite(button, HIGH);
  Serial.begin(9600);
}

void loop() {
 int xValue = analogRead(joyX);
 int yValue = analogRead(joyY);

  Serial.print(xValue);
  Serial.print("\t");
  Serial.println(yValue);

  buttonState = digitalRead(button);
```

```
Serial.println(buttonState);
if (xValue>=0 && yValue<=10)
{
  digitalWrite(10, HIGH);
}
else{digitalWrite(10, LOW);}
if (xValue<=10 && yValue>=500)
{
  digitalWrite(11, HIGH);
}
else{digitalWrite(11, LOW);}
if (xValue>=1020 && yValue>=500)
{
  digitalWrite(9, HIGH);
}
else{digitalWrite(9, LOW);}
if (xValue>=500 && yValue>=1020)
{
  digitalWrite(8, HIGH);
  }
  else{digitalWrite(8, LOW);}
  if (xValue>=1020 && yValue>=1020)
  {
    digitalWrite(9, LOW);
    digitalWrite(8, LOW);
  }
  if (buttonState == LOW)
  {
    Serial.println("Switch = High");
    digitalWrite(7, HIGH);
  }
  else{digitalWrite(7, LOW);}
  buttonState1 = digitalRead(7);
  Serial.println(buttonState1);
  delay(50);
}
```

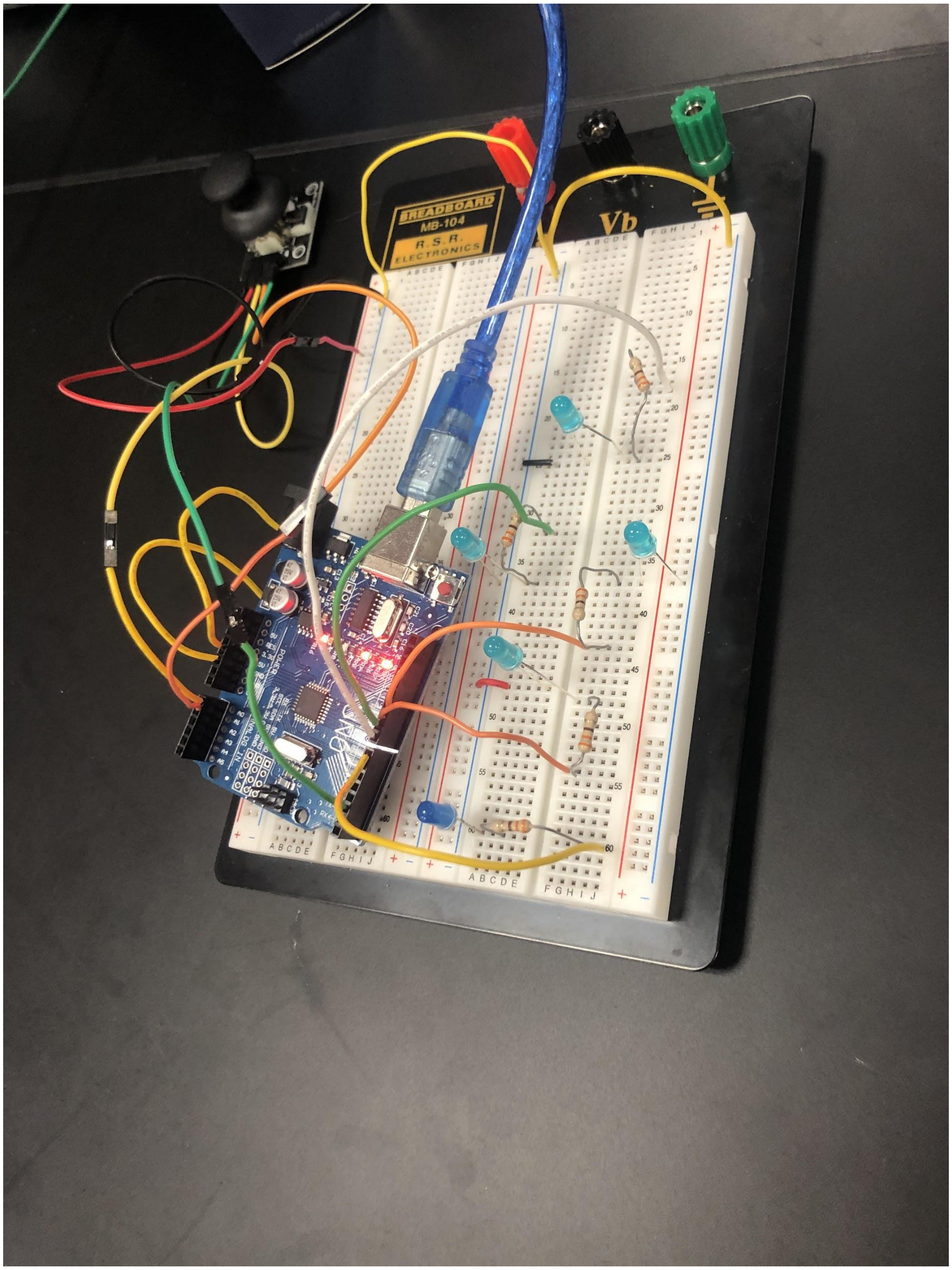
X and Y axis of the Joystick module are defined for analog pin A0 and A1 respectively. initializing PIN 2 of Arduino for the Switch of the Joystick module and the value of *buttonstate* and *buttonstate1* will be 0 at the start. Setting up the baud rate to 9600 and defined Pins 7,8,9,10,11 as an output pins and button pin as an input Pin. Initially, the button pin will remain high until the Switch will press. reading the values from the analog pin A0 and A1 and printing serially .Taking analog values of voltage at pin A0 and A1 of Arduino. These analog values will change as we move the joystick and LED will glow according to movement of joystick for turning LEDs on and off as per the movement of the Joystick shaft. The analog values will change based on the movement of Joystick showing below.



## Summary of Finding

In this project, which is called Interfacing LEDs using Joystick controller with Arduino UNO, all five LEDs will be turned ON and OFF based on the movement of Joystick. This project includes an Arduino UNO board with USB cable ( cost: 15.98$) , a Joystick Module (cost : 4,89$), 5 LEDs, bread board , Jumper wires ( cost : 5.49$), 5 Resistors - 100Ohm. First, using male - female jumper wires to connect the GND pin and 5V+ Pin on the joystick module to the (-) and (+) on the breadboard. The 5V+ pin and GND pin of Arduino Uno also have to be connected to the (+) and (-) of the breadboard using male to male Jumper Wires. Next, Connecting the VRx, VRy, and switch pins of Joystick Module to pins A0, A1 and 2 as Input Signals on the Arduino UNO board. Then, connecting pins A7 to A11 of the Arduino to the breadboard which is connected to the LEDs and 100 Ohm resistors, as Outputs. Finally, the Code has to be prepared and upload to the Arduino UNO board using USB cable. Moreover, Installing Arduino UNO to the computer also power up the board. Technically, the Joystick Module produces Analog Input Signal which is based on the movement of the joystick. Depending on the direction of the Joystick's movement, the different Analog Input Signals will be sent to the Arduino. Then from the Arduino UNO board, which is working as an ADC ( Analog - Digital Converter), convert the Analog signal and Produce the Digital signal that decides which LEDs will be ON and OFF.

**Final Result of the project.**



## References.

https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-uno.html

https://en.wikipedia.org/wiki/Arduino_Uno

https://www.electronicshub.org/arduino-joystick-interface-control-servo/

https://create.arduino.cc/projecthub/Treebug842/simple-joystick-with-led-64c6d1