Project Group 08

Spring 2023

Submitted By:

Name	Id
Md. Arafat Akash	1911799642
Nafizur Rahman	1911255642
Ibna Sina Shaun	1921125642
Maria Rahman Jui	1922236642

Course : CSE 331 (Microprocessor Interfacing & Embedded System)

Section : 04

Submitted To:

Name : Dr. Dihan Md. Nuruddin Hasan (DMH)

Date : 19 June, 2023

Contents

Project Report

1. Introduction

1.1 Project Title

Implement the given encryption table using microcontroller. Use single pole, double throw switch to configure the inputs for high and low conditions. Use LEDs to represent the corresponding output statuses.

1.2 Main Objective

- The main objective of this project was to analyze the power consumption of different microcontroller boards (which are commercially available in the market.) for a special case.
- The special case is to apply a 4 bit logic input in the microcontrollers.
- he microcontroller then would apply an encryption algorithm according a given truth table (Table 2.1.1) and give a 4 bit output which can be displayed with LEDs.
- The scope of this experiment is to observe the current drawn by the circuit for each possible logic applied to the input and the total system. As for 4 bit input, there are 16 possible inputs (from 0000 to 1111) and current in the circuit is measured for all these combinations. This is done for a minimum of 2 microcontroller boards with different specification.
- The data is collected for all the boards and a graphical analysis was done to build a comparison between the three boards.

- To get valid results all the input conditions of the microcontrollers must be kept constant. For example, a constant power supply to the circuit.

2. Method of Derivation

2.1 Truth Table

Inputs				Outputs			
I3	**I2**	**I1**	**I0**	**O3**	**O2**	**O1**	**O0**
0	0	0	0	0	1	0	0
1	0	0	0	1	1	0	1
0	1	0	0	0	0	1	0
1	1	0	0	1	1	0	0
0	0	1	0	1	0	1	1
1	0	1	0	1	0	1	0
0	1	1	0	1	0	0	1
1	1	1	0	0	0	0	0
0	0	0	1	0	1	0	1
1	0	0	1	0	0	1	1
0	1	0	1	1	0	1	0
1	1	0	1	1	1	1	1
0	0	1	1	0	1	0	0
1	0	1	1	1	0	0	0
0	1	1	1	1	1	1	0
1	1	1	1	0	1	0	0

2.2 Boolean Expression using Karnaugh Map (K-Map)

K- Map for O0(Output 0):

	I2’ I3’	I2’ I3	I2 I3	I2 I3’
I0’ I1’	0	1	0	0
I0’ I1	1	0	0	1
I0 I1	0	0	0	0
I0 I1’	1	1	1	0

O0 = I1’I2’I3 + I0’I1 I3’ + I0 I1’I2’ + I0 I1’I3

K- Map for O1(Output 1):

	I2’ I3’	I2’ I3	I2 I3	I2 I3’
I0’ I1’	0	0	0	1
I0’ I1	1	1	0	0
I0 I1	0	0	0	1
I0 I1’	0	1	1	1

O1 = I1’I2 I3’ + I0’I1 I2’ + I0 I1’I3 + I0 I2 I3’

K- Map for O2(Output 2):

	I2' I3'	I2' I3	I2 I3	I2 I3'
I0' I1'	1	1	1	0
I0' I1	0	0	0	0
I0 I1	1	0	1	1
I0 I1'	1	0	1	0

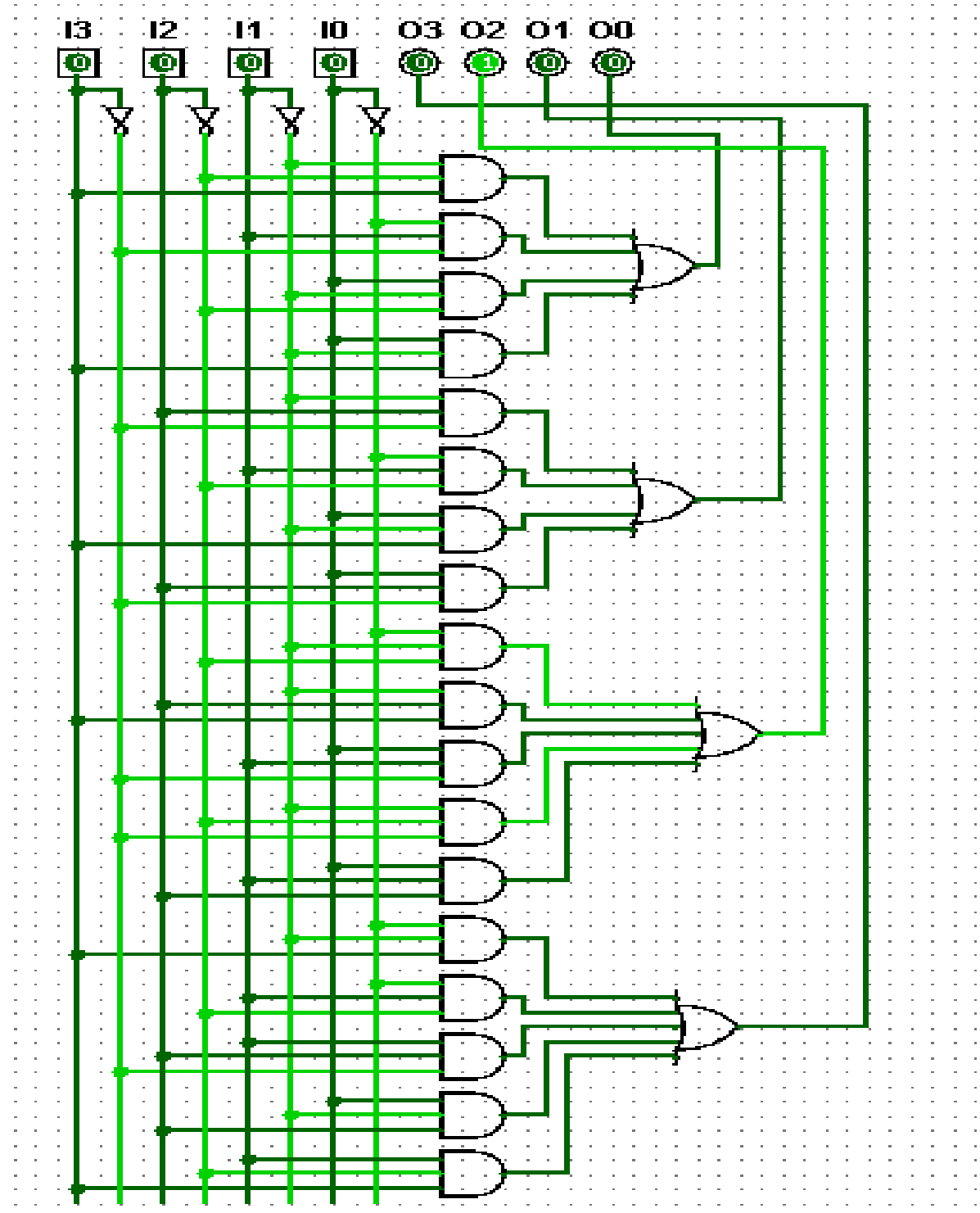
O2 = I0'I1'I2' + I1'I2 I3 + I0 I1 I3' + I1'I2'I3'+ I0 I1 I2

K- Map for O3(Output 3):

	I2' I3'	I2' I3	I2 I3	I2 I3'
I0' I1'	0	1	1	0
I0' I1	1	1	0	1
I0 I1	0	1	0	1
I0 I1'	0	0	1	1

O3 = I0'I1'I3 + I0'I1 I2' + I1 I2 I3' + I0 I1'I2 + I1 I2'I3

2.3 Logisim Simulation



3. Circuit Design & Simulation

3.1 Circuit Design Procedure

We have done the experiment using 2 microcontroller boards:

1. Arduino Uno R3 (16 MHz)
2. Arduino Nano (16MHz)

3.2 Simulation Procedure

Simulation is done using Proteus 8 Professional. We cannot simulate Arduino directly in Proteus, first we need to install some specific libraries which include the simulation libraries for both the Arduino Uno and Nano.

We have to write our code (all codes are given in Appendix C) to implement the Boolean expression that we have derived. This code will be converted into a hex file by the Arduino IDE (Appendix A) to be used to simulate the circuit in Proteus.

After performing the simulation of the two boards (Uno & Nano) we saw that our code was working fine and we were getting our desired output. The 2 snapshots of our circuit diagram for the two boards is given in the following section.

Now we will see the logic behind the codes.

3.3 Program Logic

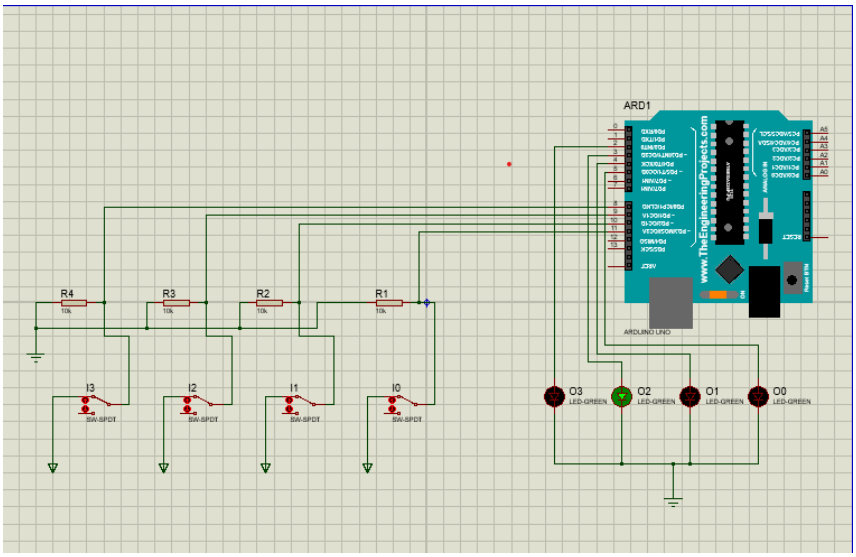
All the codes for the design are given in Appendix C. All the codes for the two boards use this same concept and the codes for Arduino Uno and Nano are exactly same.

3.4 Program Code For Both Arduino Uno & Nano

```
int I0=11;
int I1=10;
int I2=9;
int I3=8;
int O0=5;
int O1=4;
int O2=3;
int O3=2;
void setup() {
  pinMode(I0, INPUT);
  pinMode(I1, INPUT);
  pinMode(I2, INPUT);
  pinMode(I3, INPUT);
  pinMode(O0, OUTPUT);
  pinMode(O1, OUTPUT);
  pinMode(O2, OUTPUT);
  pinMode(O3, OUTPUT);
}
void loop() {
  boolean I0State = digitalRead(I0);
  boolean I1State = digitalRead(I1);
  boolean I2State = digitalRead(I2);
  boolean I3State = digitalRead(I3);
  boolean O0State;
  boolean O1State;
  boolean O2State;
  boolean O3State;
  O0State = (!I1State & !I2State & I3State) | (!I0State & I1State & !I3State) |
```

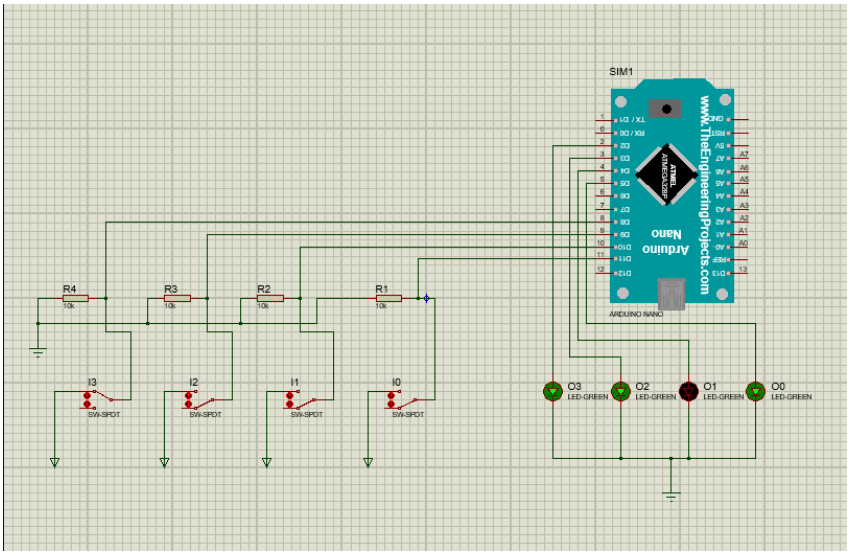
```
    (I0State & !I1State & !I2State) | (I0State & !I1State & I3State);
  O1State = (!I1State & I2State & !I3State) | (!I0State & I1State & !I2State) |
    (I0State & !I1State & I3State) | (I0State & I2State & !I3State);
  O2State = (!I0State & !I1State & !I2State) | (!I1State & I2State & I3State) |
    (I0State & I1State & !I3State) | (!I1State & !I2State & !I3State) | (I0State
    & I1State & I2State);
  O3State = (!I0State & !I1State & I3State) | (!I0State & I1State & !I2State) |
    (I1State & I2State & !I3State) | (I0State & !I1State & I2State) | (I1State &
    !I2State & I3State);
  digitalWrite(O0, O0State);
  digitalWrite(O1, O1State);
  digitalWrite(O2, O2State);
  digitalWrite(O3, O3State);
}
```

3.5 Arduino Uno Simulation



Circuit 3.5.1: Circuit diagram for Arduino Uno R3.

3.6 Arduino Nano Simulation



Circuit 3.6.1: Circuit diagram for Arduino Uno R3.

4. Appendix A: Software Specifications

4.1 Arduino IDE

Arduino IDE is an open-source tool that makes it possible for users to write as well as upload code to a work environment in real-time. Since the written code will be moved to the cloud, it's frequently used by those who need an additional level of redundancy. Arduino IDE offers full compatibility to any Arduino-based software board. The software can easily be deployed in any Linux, Mac or Windows operating systems. Most of its parts are written within JavaScript for seamless compilation and editing. While the tool's main aim is based on code writing, it offers several noteworthy functionalities. For instance, Arduino IDE lets users share important project information to company stakeholders. Users are given the freedom to make internal layouts and schematic modifications when needed. Comprehensive guides are available for those who need help in the installation process. Tutorials are present for users who have little experience dealing with the tool's framework. Arduino IDE is highly rated by users for its ease of use. It can conduct complex processes while keeping computer resources to a minimum. The tool makes it easy for users to access their libraries. At the same time, it offers updated support for the latest Arduino boards, which can help users with their sketches using the latest IDE version.

Website: https://www.arduino.cc/en/software

4.2 Proteus 8 Professional

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

Website: https://www.labcenter.com/downloads/

4.3 Logisim

When learning computer architecture and logic circuits, you will need a real-world, graphical example of what you are studying. Text and diagrams only go so far. A helpful tool for designing and simulating logic circuits is Logisim. Because the tool lets you create large circuits from smaller circuits, you can design entire CPUs using Logisim. Further, the tool will run on any computer! The interface itself is very intuitive and the use of color-coding of wires and elements allows for easy analysis and testing of circuits. You can also save the completed file as an image, or as a .circ file (core to Logisim).

Website: http://www.cburch.com/logisim/download.html

4.4 Truth Table

A truth table is a mathematical table used in logic specifically in connection with Boolean algebra, Boolean functions, and propositional calculus which sets out the functional values of logical expressions on each of their functional arguments, that is, for each combination of values taken by their logical variables.

4.5 K-Map

The Karnaugh map is a method of simplifying Boolean algebra expressions. Maurice Karnaugh introduced it in 1953 as a refinement of Edward W. Veitch's 1952 Veitch chart, which was a rediscovery of Allan Marquand's 1881 logical diagram aka Marquand diagram but with a focus now set on its utility for switching circuits.

5. Appendix B: Hardware Specification

5.1 Arduino Uno R3

The Arduino UNO R3 is frequently used microcontroller board in the family of an Arduino. This is the latest third version of an Arduino board and released in the year 2011. The main advantage of this board is if we make a mistake. We can change the microcontroller on the board. The main features of this board mainly include, it is available in DIP (dual-inline-package), detachable and ATmega328 microcontroller. The programming of this board can easily be loaded by using an Arduino computer program. This board has huge support from the Arduino community, which will make a very simple way to start working in embedded electronics, and many more applications. Please refer the link to know about Arduino – Basics, and Design.

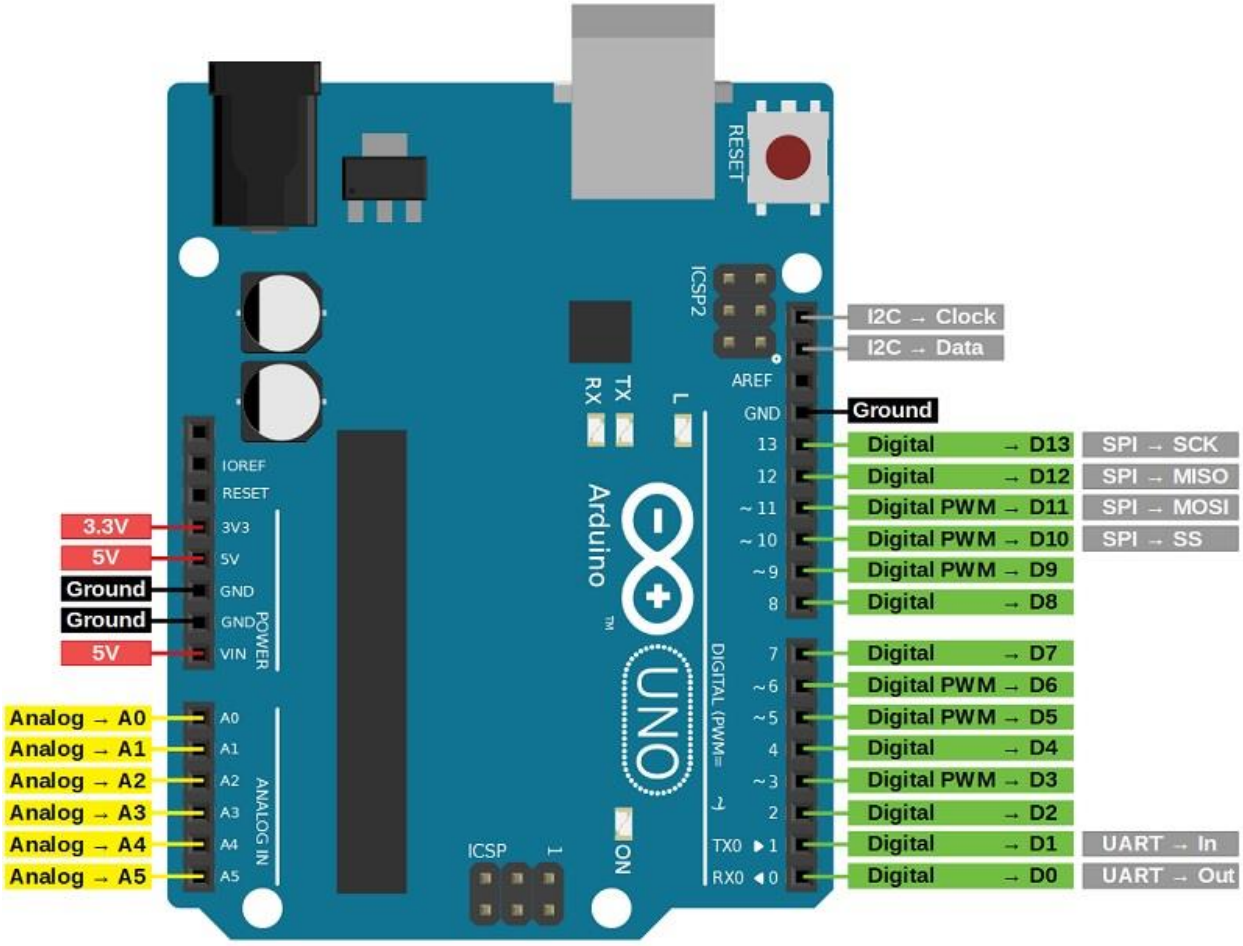
5.2 Arduino Uno R3 Specifications

The Arduino Uno R3 board includes the following specifications.

- It is an ATmega328P based Microcontroller
- The Operating Voltage of the Arduino is 5V
- The recommended input voltage ranges from 7V to 12V
- The i/p voltage (limit) is 6V to 20V
- Digital input and output pins-14
- Digital input & output pins (PWM)-6
- Analog i/p pins are 6
- DC Current for each I/O Pin is 20 mA
- DC Current used for 3.3V Pin is 50 mA

- Flash Memory -32 KB, and 0.5 KB memory is used by the boot loader
- SRAM is 2 KB
- EEPROM is 1 KB
- The speed of the CLK is 16 MHz
- In Built LED
- Length and width of the Arduino are 68.6 mm X 53.4 mm
- The weight of the Arduino board is 25 g
-

Pin Diagram:



5.3 Arduino Nano

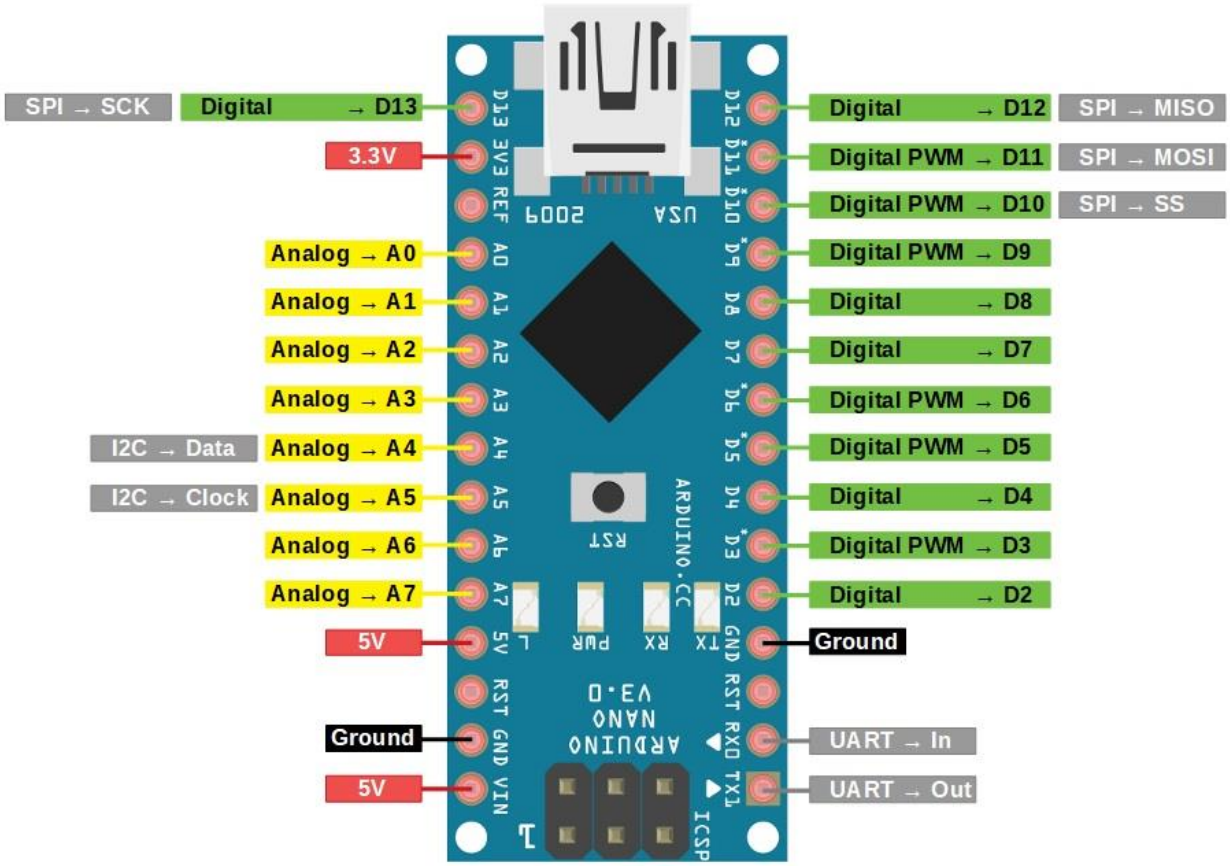
Arduino Nano is one type of microcontroller board, and it is designed by Arduino.cc. It can be built with a microcontroller like Atmega328. This microcontroller is also used in Arduino UNO. It is a small size board and also flexible with a wide variety of applications. Other Arduino boards mainly include Arduino Mega, Arduino Pro Mini, Arduino UNO, Arduino YUN, Arduino Lilypad, Arduino Leonardo, and Arduino Due. And other development boards are AVR Development Board, PIC Development Board, Raspberry Pi, Intel Edison, MSP430 Launchpad, and ESP32 board. This board has many functions and features like an Arduino Duemilanove board. However, this Nano board is different in packaging. It doesn't have any DC jack so that the power supply can be given using a small USB port otherwise straightly connected to the pins like VCC & GND. This board can be supplied with 6 to 20volts using a mini USB port on the board.

5.4 Arduino Nano Specifications

- ATmega328P Microcontroller is from 8-bit AVR family
- Operating voltage is 5V
- Input voltage (Vin) is 7V to 12V
- Input/Output Pins are 22
- Analog i/p pins are 6 from A0 to A5
- Digital pins are 14
- Power consumption is 19 mA
- I/O pins DC Current is 40 mA
- Flash memory is 32 KB
- SRAM is 2 KB
- EEPROM is 1 KB
- CLK speed is 16 MHz
- Weight-7g

- Size of the printed circuit board is 18 X 45mm
- Supports three communications like SPI, IIC, & USART

Pin Diagram:



Source: Fritzing

6. Question & Answers

6.1 Arduino Uno R3

1. What is the clock frequency of the microcontroller used?
Answer: 16 MHz.

2. What is the data bus width of the microcontroller used?
Answer: 8 bit.

3. What is the size of your hex file generated? Attach the hex codes in your report.
Answer: 2.59 KB.

4. Can the project be implemented by using interrupt?
Answer: Yes, since all the pins of the Arduino Uno R3 can be used as interrupts.

5. Is the main routine required to be an infinite loop? Provide explanation in favor of your answer.
Answer: Yes the main routine is required to be an infinite loop. This is because we need to continuously check the input status and at the same time provide a constant output logic all the time. If it were not an infinite loop the program would execute and the finish and after that if you change the input logic no output will be given.

6. Is the project referring encryption or decryption from input to output?
Answer: This is like an encryption circuit. We are applying an encryption algorithm to an information we have. This algorithm can be used to decrypt and find the original information.

6.2 Arduino Nano

1. What is the clock frequency of the microcontroller used?
Answer: 16 MHz.

2. What is the data bus width of the microcontroller used?
Answer: 8 bit.

3. What is the size of your hex file generated? Attach the hex codes in your report.
Answer: 2.59 KB.

4. Can the project be implemented by using interrupt?
Answer: Yes, since all the pins of the Arduino Nano can be used as interrupts.

5. Is the main routine required to be an infinite loop? Provide explanation in favor of your answer.

Answer: Yes the main routine is required to be an infinite loop. This is because we need to continuously check the input status and at the same time provide a constant output logic all the time. If it were not an infinite loop the program would execute and the finish and after that if you change the input logic no output will be given.

6. Is the project referring encryption or decryption from input to output?
Answer: This is like an encryption circuit. We are applying an encryption algorithm to an information we have. This algorithm can be used to decrypt and find the original information.

7. Resources

7.1 Project Demonstration Video Link

https://drive.google.com/file/d/1oQ8QxgwZoHqyotJ6qKwX-x-eUoPR7PGj/view?usp=drive_link

7.2 Project Related Github Link

https://github.com/ArafatAkashAkku/CSE-331-Project-Group-08