Faculty of Information Technology

IN -1900 – ICT Project

Automatic Pipe Cutting Machine

Group No: 39

1. 204178K	**Rathnasinghe D.S**
2. 204027A	Chathurangi P.K.D.P
3. 204117B	Madhubhashini P.H.A.E
4. 204200A	Shalinda J.D.A.
5.204215A	Tharangana K.M.

Supervisors' Names : Mr.B.H.Sudantha,
Dean/Senior Lecturer,
Dept. of Information Technology.

Co-Supervisor Name : Ms. I. A. Wijethunga,
Lecturer,
Dept. of Information Technology.

Signatures of Supervisors:

Date of Submissoin :08/06/2022

1. ……………………………

2 ……………………………

Table of Content

Page

1. Introduction

There are different types of machines used in construction sites for different purposes. Complex machinery and new techniques are continuously being developed for cost-effective manufacturing. Simultaneously, care should be taken to ensure that quality and accuracy have not been compromised. Machines have become an intrinsic component of human beings in the age of automation. The utilization of an automation machine demonstrates that it has a higher production rate than a manual machine. Today, automation has made a significant impact on the industrial manufacturing process, allowing for the production of identical and accurate products while reducing human involvement.

A pipe cutting machine is one of the essential machines that is used in mass manufacturing. New machinery and processes are constantly being developed to create various items at lower costs while maintaining excellent quality. For cutting PVC pipes, metal pipes and box bars there exist separate machines which are manual or semi-automatic. But there is an industrial deficit for a fully automatic machine that can reduce human participation to boost productivity and accuracy.

2. Literature Survey

Jeet Machine Tools Construction [5] has developed a Pipe Cutter Machine to cut metal pipes. They have developed semi-automatic systems to make their machinery more efficient and productive. This machine can cut and separate metals that are tough to cut by hand.

But in this machine, there is no automatic method to lower the blade to the cutting point. It requires manpower to direct the cutting arm to its intended location. Because of this manual part, it is not possible to carry out the cutting without an operator. It is a limitation of this machine.

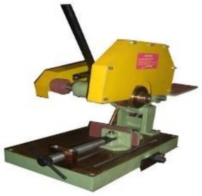


Figure 2.1-Pipe cutter machine

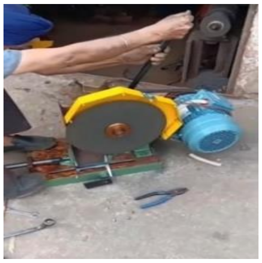


Figure 2.2-Manual Cutting

Sohit Engineering Works [4] has manufactured a semi-automatic pipe cutting machine named "Sohit 385 Automatic Pipe Cutting Machine" which is more suitable for medium-scale industries. For reducing human effort, there is an automatic clamping and cutting process in this machine that helps to save 70% manpower. Since there is no automatic length measuring method is used in this machine, the operator must align the pipe to the cutting point according to his requirement. This machine weighs about 425kg and that causes less portability.

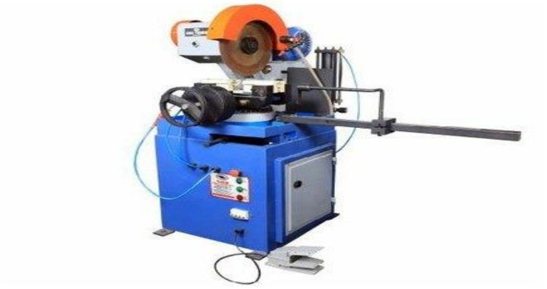


Figure 2.3-Sohit 385 Automatic Pipe Cutting Machine

3. Aim & Objectives

3.1 Aim:

Design and develop a fully automatic pipe cutting machine to measure and cut PVC pipes, steel pipes and box bars.

3.2 Objectives:

- To measure and cut pipes automatically according to the different measurements.
- To cut pipes of different materials by using one machine.
- To reduce manpower by limiting human intervention only to place the pipe on the machine.
- To increase the safety level of the operator by reducing human involvement.

4. Analysis and Design

4.1 Block Diagram

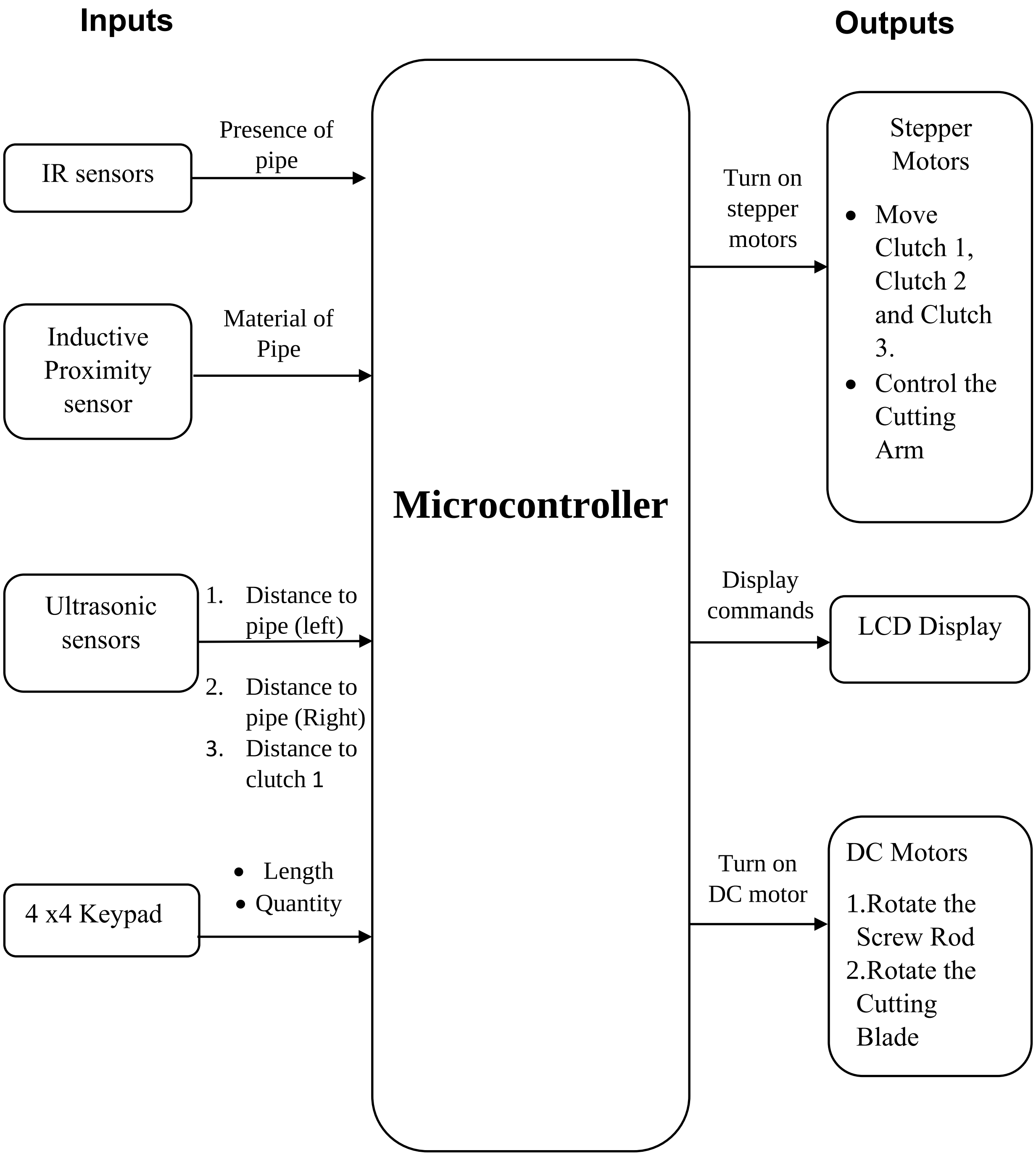


Figure 4.1-Block diagram

4.2 3D Diagram

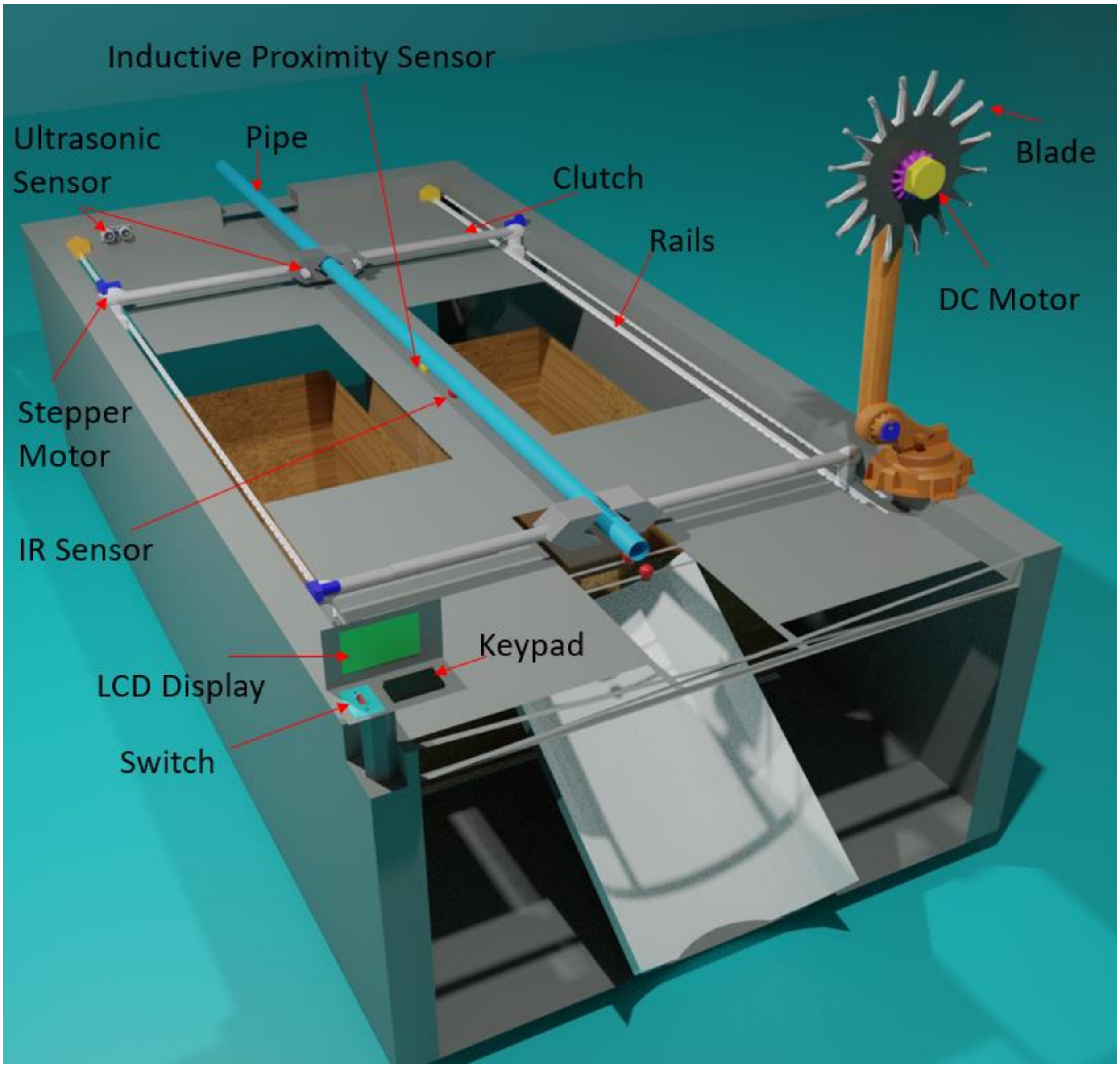


Figure 4.2-3D Model of the machine

4.3 Schematic Diagram

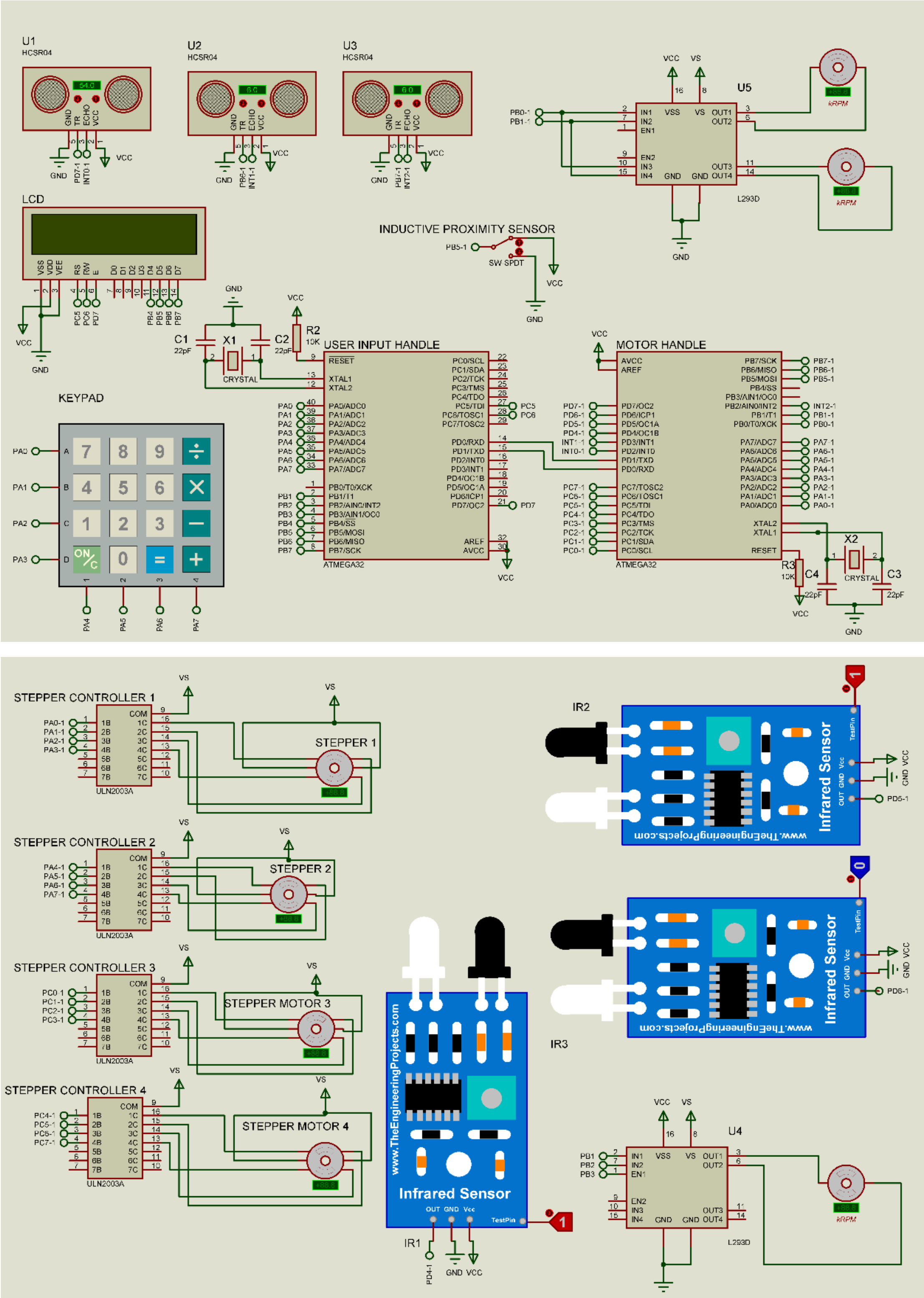


Figure 4.3-Schematic diagram of the machine

4.4 PCB Design

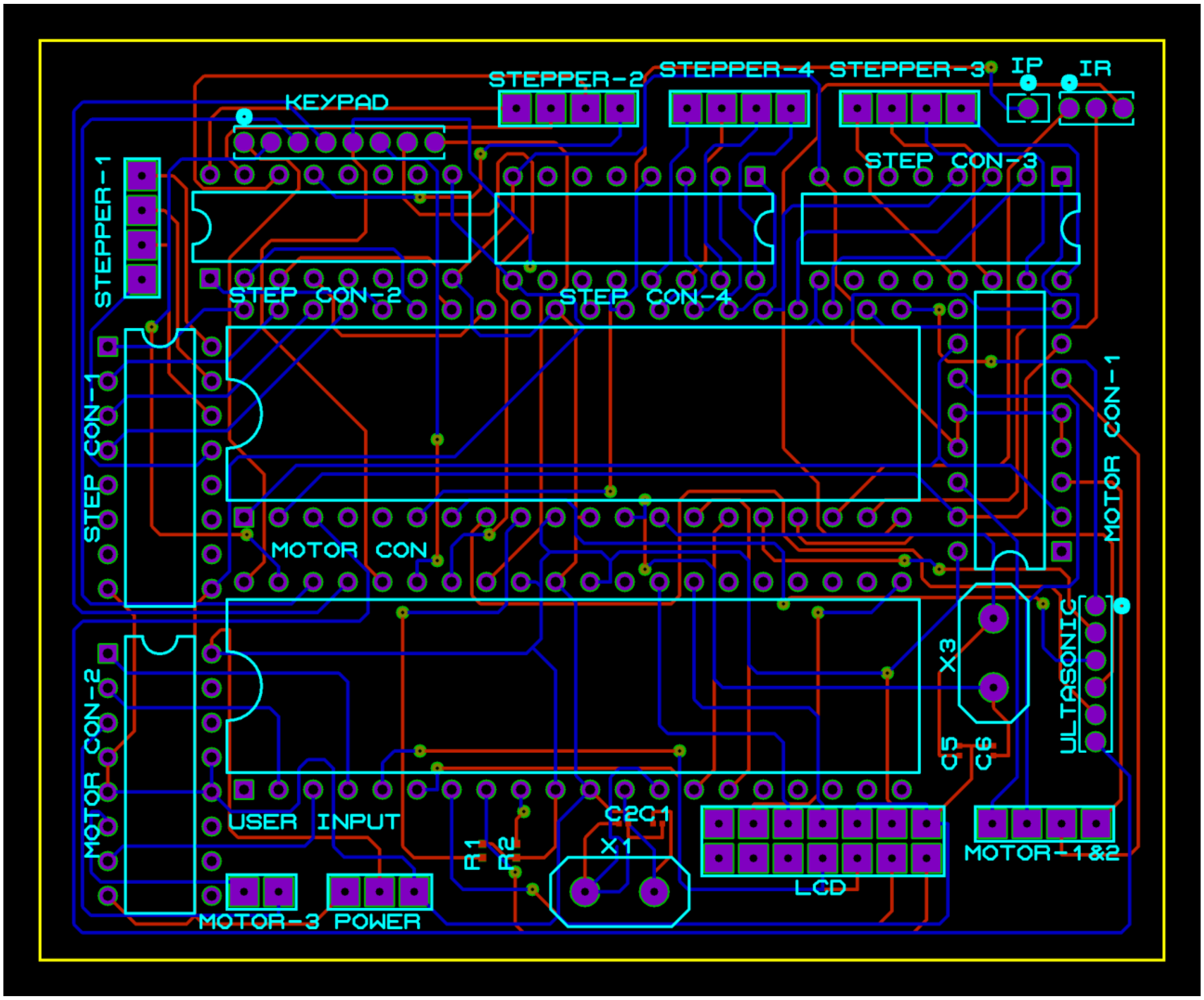
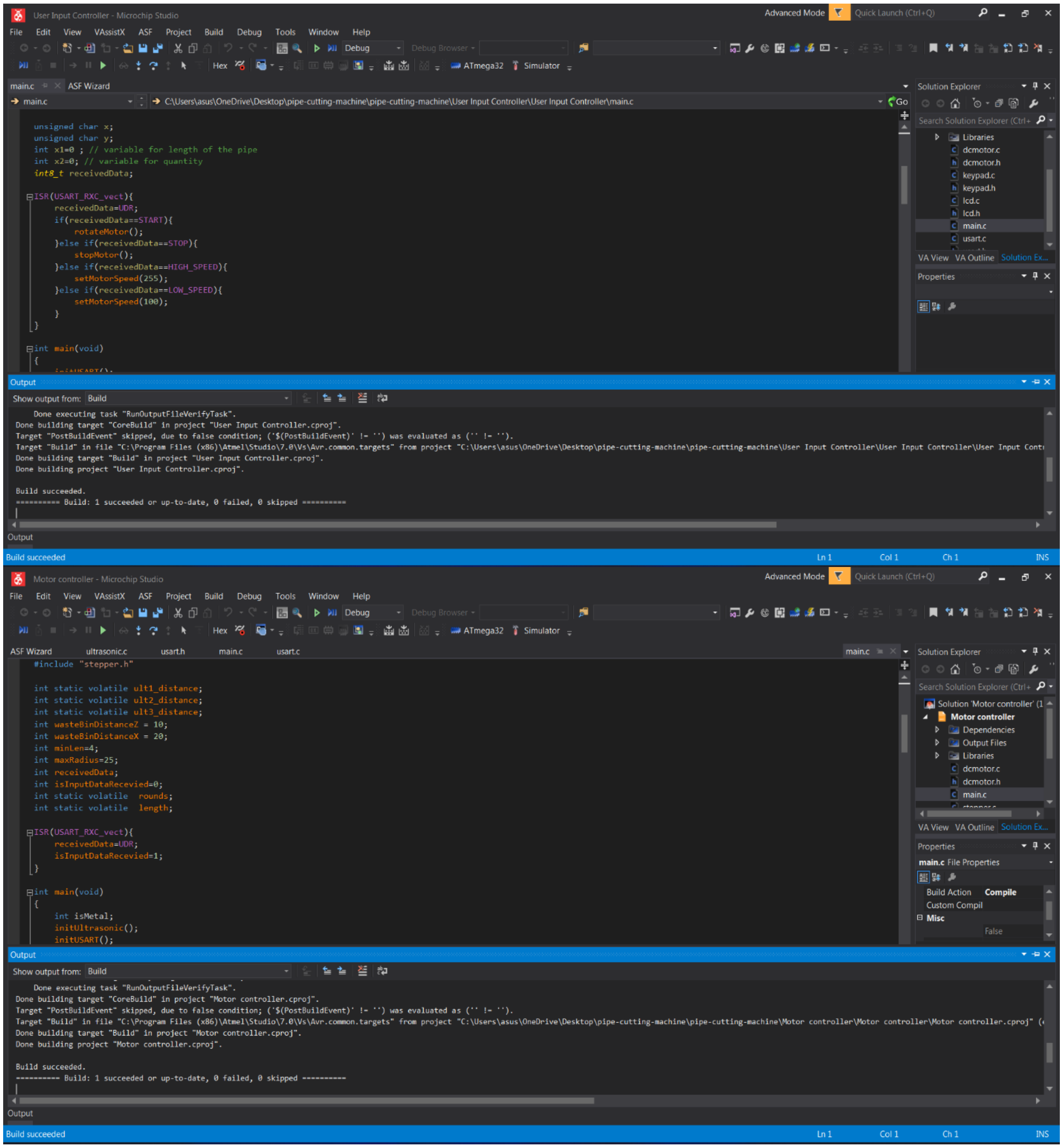


Figure 4.4-PCB design of the machine

5. Testing and Implementation

According to need of our machine, we have used two microcontrollers to interface all the sensors and motors. Also, we have initialized them as ‘User input controller’ and ‘Motor controller’. Then we have developed the code for the machine by connecting these microcontrollers in USART communication and included header and c files for all the sensors, LCD, keypad, and motors. Both codes built successfully.



```c
unsigned char x;
unsigned char y;
int x1=0 ; // variable for length of the pipe
int x2=0; // variable for quantity
int8_t receivedData;

ISR(USART_RXC_vect){
    receivedData=UDR;
    if(receivedData==START){
        rotateMotor();
    }else if(receivedData==STOP){
        stopMotor();
    }else if(receivedData==HIGH_SPEED){
        setMotorSpeed(255);
    }else if(receivedData==LOW_SPEED){
        setMotorSpeed(100);
    }
}

int main(void)
{
```

```
    Done executing task "RunOutputFileVerifyTask".
Done building target "CoreBuild" in project "User Input Controller.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets" from project "C:\Users\asus\OneDrive\Desktop\pipe-cutting-machine\pipe-cutting-machine\User Input Controller\User Input Controller\User Input Contr
Done building target "Build" in project "User Input Controller.cproj".
Done building project "User Input Controller.cproj".

Build succeeded.
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
```

```c
#include "stepper.h"

int static volatile ult1_distance;
int static volatile ult2_distance;
int static volatile ult3_distance;
int wasteBinDistanceZ = 10;
int wasteBinDistanceX = 20;
int minLen=4;
int maxRadius=25;
int receivedData;
int isInputDataRecevied=0;
int static volatile  rounds;
int static volatile  length;

ISR(USART_RXC_vect){
    receivedData=UDR;
    isInputDataRecevied=1;
}

int main(void)
{
    int isMetal;
    initUltrasonic();
    initUSART();
```

```
    Done executing task "RunOutputFileVerifyTask".
Done building target "CoreBuild" in project "Motor controller.cproj".
Target "PostBuildEvent" skipped, due to false condition; ('$(PostBuildEvent)' != '') was evaluated as ('' != '').
Target "Build" in file "C:\Program Files (x86)\Atmel\Studio\7.0\Vs\Avr.common.targets" from project "C:\Users\asus\OneDrive\Desktop\pipe-cutting-machine\pipe-cutting-machine\Motor controller\Motor controller\Motor controller.cproj" (
Done building target "Build" in project "Motor controller.cproj".
Done building project "Motor controller.cproj".

Build succeeded.
---------- Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ----------
```

Figure 5.1- Microchip studio code file

Then we have simulated the code with schematic diagram which we have designed with proteus software. After that we tested functioning of each sub process of the machine.

5.1 Entering user inputs

At the beginning, machine must capture needed length and quantity entered by the user.

Result:

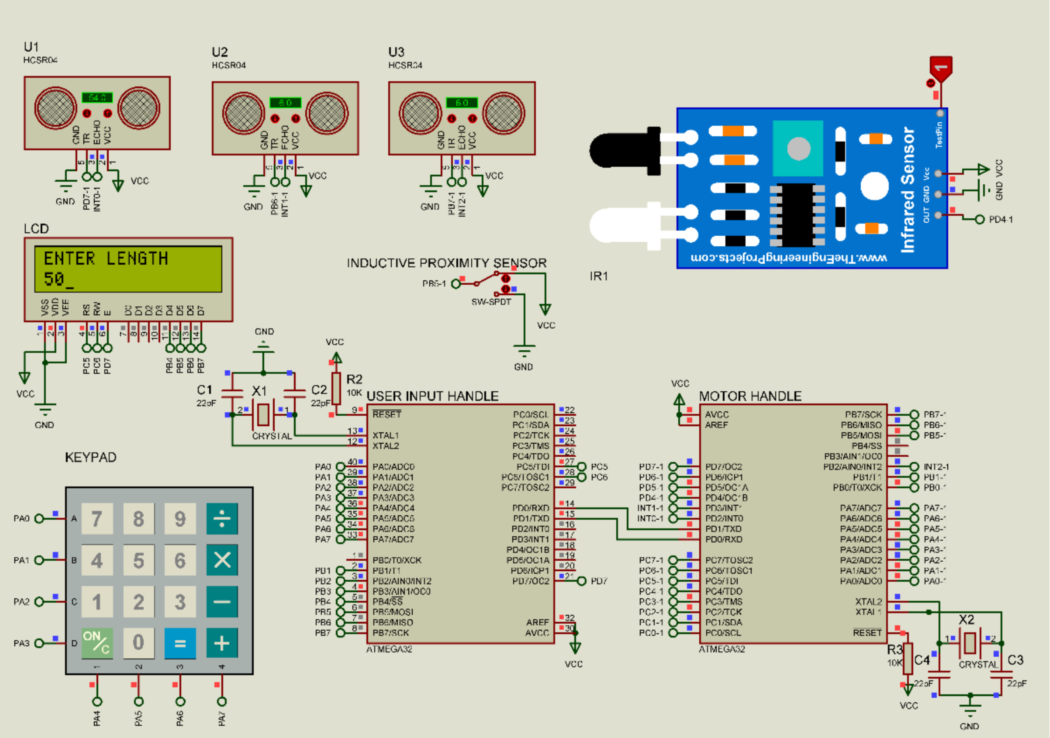


Figure 5.2 – Functioning of keypad and lcd

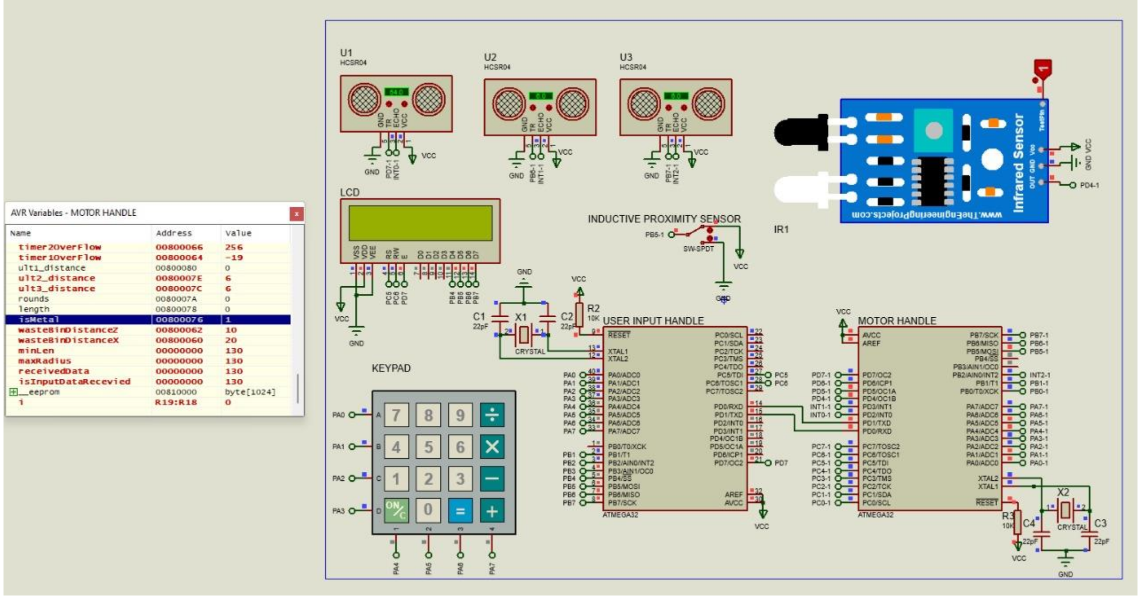
5.2 Detecting and identifying the material

After that, the machine will detect the presence of the pipe by the IR sensor and will identify the material of the pipe by the Inductive Proximity sensor.

- Both positive signals by IR sensor and Inductive proximity sensor can be considered as ‘metal object detected’. Which means it is a metal pipe.
- Positive signal by IR sensor and negative signal by Inductive proximity sensor can be considered as ‘nonmetal object detected’. Which means it is a PVC pipe.

As it is in simulation mode, we have hard coded inputs of the sensors and got the out as below.

Result:



AVR Variables - MOTOR HANDLE

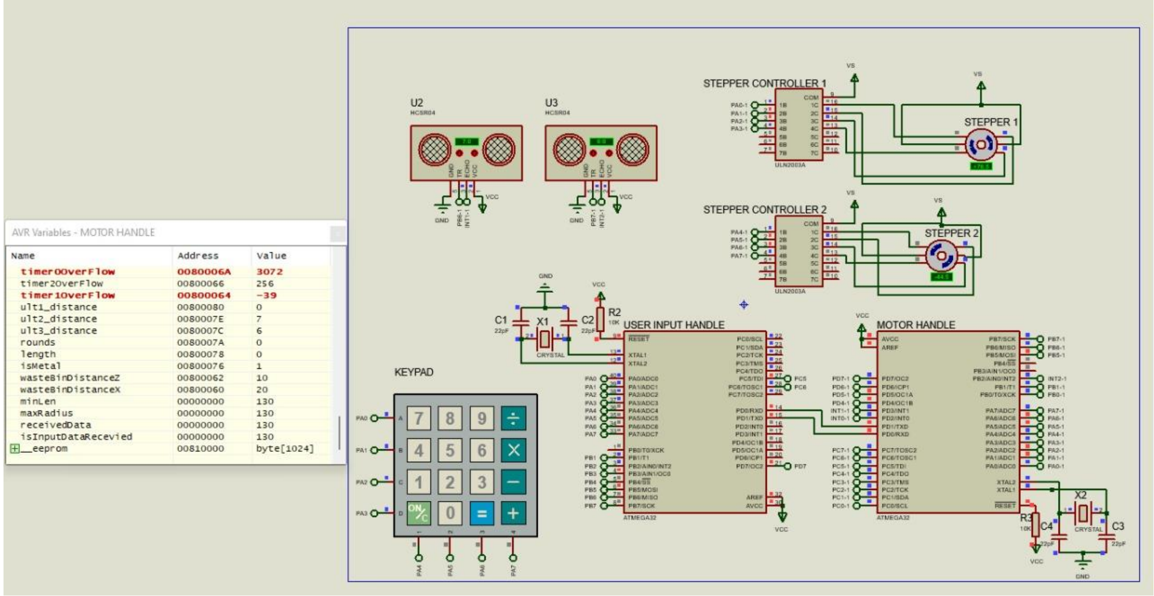
Name	Address	Value
timer2OverFlow	00800066	256
timer1OverFlow	00800064	-19
ult1_distance	00800080	0
ult2_distance	0080007E	6
ult3_distance	0080007C	6
rounds	0080007A	0
length	00800078	0
isMetal	00800076	1
wasteBinDistanceZ	00800062	10
wasteBinDistanceX	00800060	20
minLen	00000000	130
maxRadius	00000000	130
receivedData	00000000	130
isInputDataRecevied	00000000	130
⊞__eeprom	00810000	byte[1024]
i	R19:R18	0

Figure 5.3- identifying the material of pipe

5.3 Tightening Process

After detecting and identifying the pipe, clutches will automatically come forward and the pipe will be tightened. So, stepper motor 1,2 & 3 must rotate accordingly.

Result:



AVR Variables - MOTOR HANDLE

Name	Address	Value
timer0OverFlow	0080006A	3072
timer2OverFlow	00800066	256
timer1OverFlow	00800064	-39
ult1_distance	00800080	0
ult2_distance	0080007E	7
ult3_distance	0080007C	6
rounds	0080007A	0
length	00800078	0
isMetal	00800076	1
wasteBinDistanceZ	00800062	10
wasteBinDistanceX	00800060	20
minLen	00000000	130
maxRadius	00000000	130
receivedData	00000000	130
isInputDataRecevied	00000000	130
⊞__eeprom	00810000	byte[1024]

Figure 5.4-tighning the pipe

5.4 Aligning Process

Two IR sensors will be used for this process. When one IR sensor is giving a positive signal and next IR sensor is giving a negative signal,it can be considered as an aligned position. So, microcontroller has to check the inputs of these sensors continuosly and rotate the DC motors of rail system.

Result:

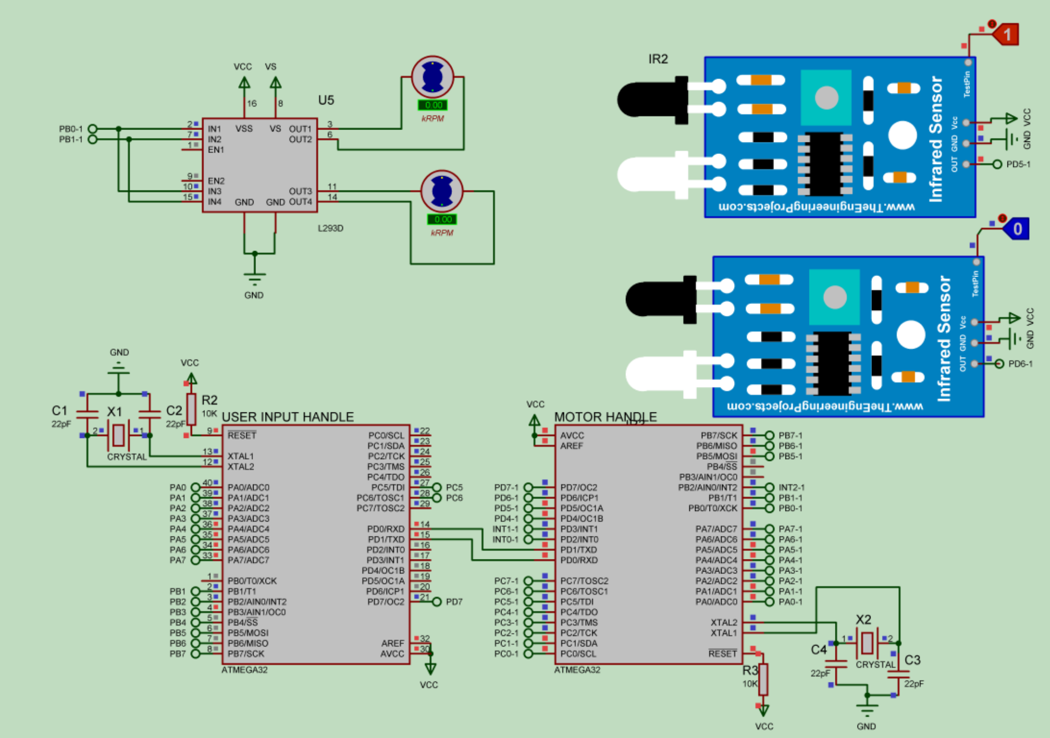


Figure 5.5-Aligning process

5.5 Length Measuring Process

After aligning, the measuring process will be started by the machine. Another Ultra-sonic sensor will be used here. The pipe will be moved forward by the clutches which are connected to the rail system. The ultrasonic sensor will measure the distance to clutch 1. Clutches 1 &2 will move forward and backward, and it will bring the pipe forward until the given length is satisfied.

So, this time micro controller has to check the inputs of ultrasonic sensor 1 and rotate the DC motors

Result:

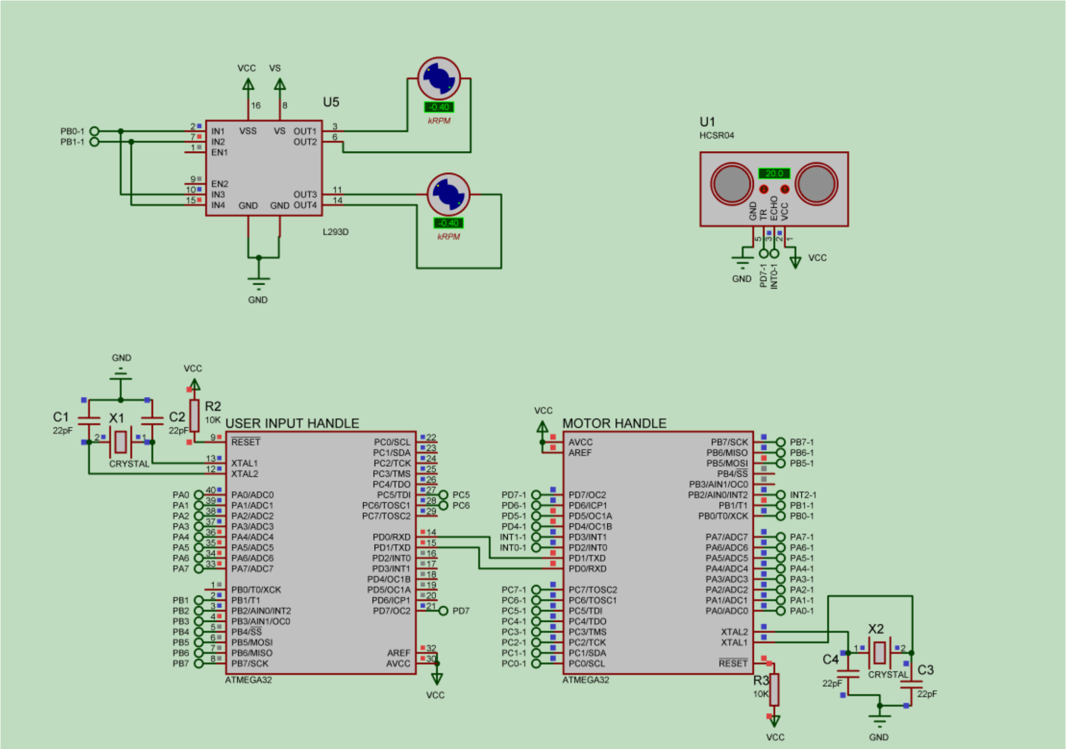


Figure 5.6- length measuring process

5.6 Cutting Process

After identifying the cutting point of the pipe, the cutting process will be started. Here the cutting arm, which is controlled by the stepper motor, will come down and cutting blade will start rotating and cutting the pipe.

So, the stepper motor and DC motor of cutting arm must rotate.

Result:

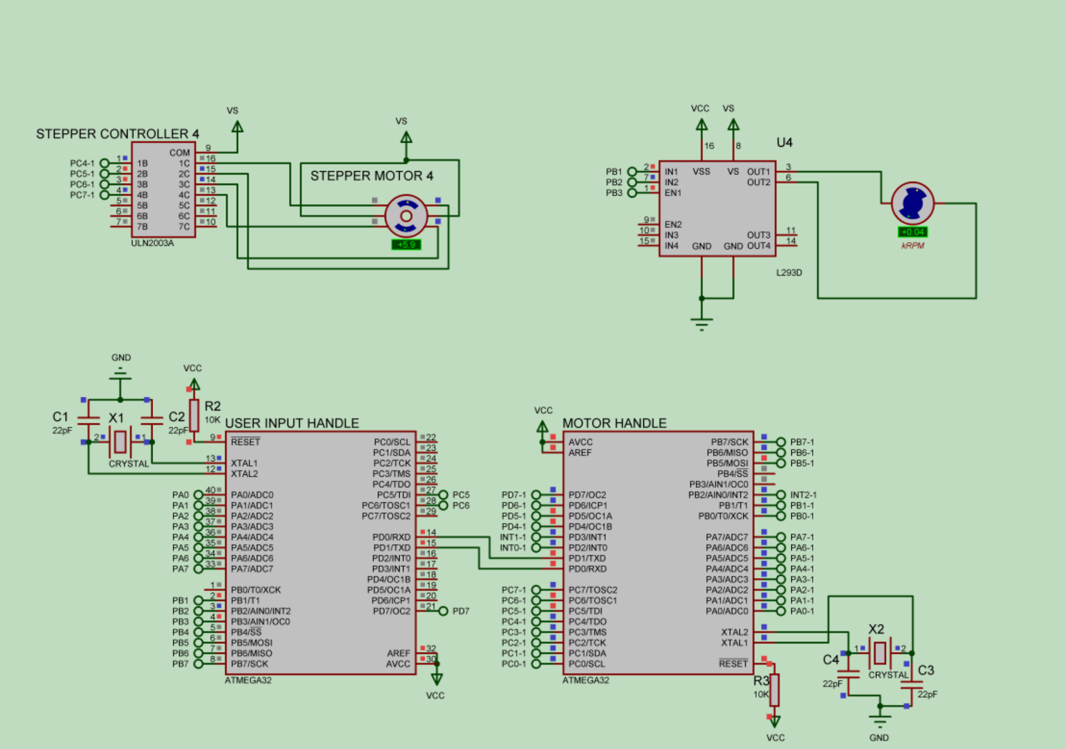

STEPPER CONTROLLER 4
ULN2003A
COM
1B 2B 3B 4B 5B 6B 7B
1C 2C 3C 4C 5C 6C 7C
PC4-1
PC5-1
PC6-1
PC7-1
VS
STEPPER MOTOR 4
+5.9
VCC
U4
L293D
IN1 IN2 EN1 EN2 IN3 IN4
VSS VS GND GND
OUT1 OUT2 OUT3 OUT4
PB1
PB2
PB3
+0.04
kRPM
GND
C1
22pF
X1
CRYSTAL
C2
22pF
R2
10K
USER INPUT HANDLE
ATMEGA32
RESET
XTAL1
XTAL2
PA0/ADC0 PA1/ADC1 PA2/ADC2 PA3/ADC3 PA4/ADC4 PA5/ADC5 PA6/ADC6 PA7/ADC7
PB0/T0/XCK PB1/T1 PB2/AIN0/INT2 PB3/AIN1/OC0 PB4/SS PB5/MOSI PB6/MISO PB7/SCK
PC0/SCL PC1/SDA PC2/TCK PC3/TMS PC4/TDO PC5/TDI PC6/TOSC1 PC7/TOSC2
PD0/RXD PD1/TXD PD2/INT0 PD3/INT1 PD4/OC1B PD5/OC1A PD6/ICP1 PD7/OC2
AREF
AVCC
PA0 PA1 PA2 PA3 PA4 PA5 PA6 PA7
PB1 PB2 PB3 PB4 PB5 PB6 PB7
PC5
PC6
PD7
MOTOR HANDLE
ATMEGA32
AVCC
AREF
PD7-1 PD6-1 PD5-1 PD4-1 INT1-1 INT0-1
PC7-1 PC6-1 PC5-1 PC4-1 PC3-1 PC2-1 PC1-1 PC0-1
PB7-1 PB6-1 PB5-1 INT2-1 PB1-1 PB0-1
PA7-1 PA6-1 PA5-1 PA4-1 PA3-1 PA2-1 PA1-1 PA0-1
R3
10K
X2
CRYSTAL
C4
22pF
C3
22pF

Figure 5.7-pipe cutting process

5.7 Wastage Separating and Collecting Process

At the end of the cutting process, the remaining piece of pipe will be kept between clitch 3 and clutch 4. If the length of this piece is less than 10cm, the machine will consider it as a wastage. IR sensor will be used to identify this. If the remaining piece is identified as a waste, it will be grabbed by clutch 1 and clutch 2 and will be released to the relavant bin. There are two bins placed inside the machine for this purpose.

First, DC motors of rail system must rotate anticlock wise and the stepper motors of clutches must rotate.

Result:

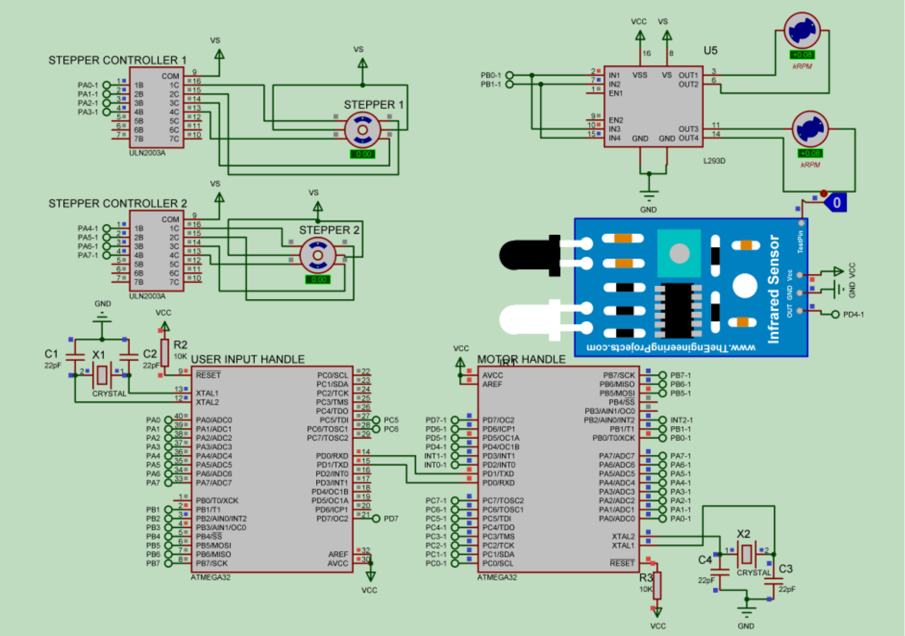
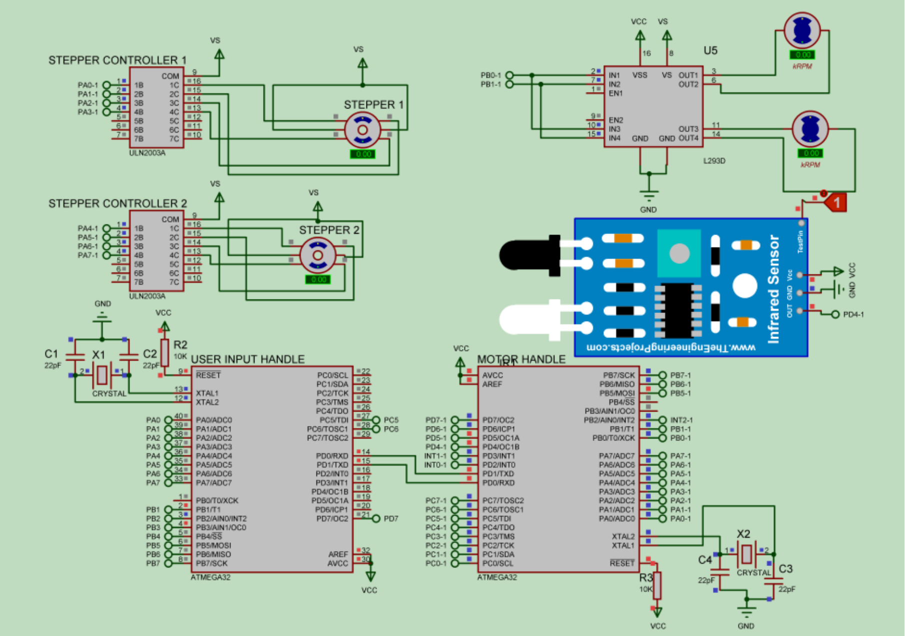


Figure 5.8-remaining piece identified as wastage and separated



.

Figure 5.9-remaining piece is not identified as wastage

6. Cost Estimation

Figure 6.1-Cost Estimation

Components	**Units**	**Unit Price**	**Price**
Microcontroller	2	Rs: 550.00	Rs: 1100.00
IR Sensor	3	Rs: 120.00	Rs: 360.00
Inductive Proximity Sensor	1	Rs: 550.00	Rs: 550.00
Ultrasonic Sensor	3	Rs: 400.00	Rs: 1200.00
Keypad	1	Rs: 160.00	Rs: 160.00
LCD	1	Rs: 800.00	Rs: 800.00
Stepper Motor	4	Rs: 3200.00	Rs:12800.00
DC Motor	3	Rs: 950.00	Rs: 2850.00
Grinding Disc	1	Rs: 2000.00	Rs: 2000.00
Drivers A4988 (Stepper motors)	4	Rs: 460.00	Rs: 1840.00
Drivers L1293 (DC motor)	3	Rs: 400.00	Rs: 1200.00
Other expenses			Rs: 5000.00
Total			Rs:29860.00

10. References

1. Components101. 2021. *16x2 LCD Module*. [online] Available at: <https://components101.com/displays/16x2-lcd-pinout-datasheet>
2. Cuidevices.com. 2021. [online] Available at: <https://www.cuidevices.com/product/resource/nema17-amt112s.pdf>
3. Electronicwings.com. 2021. *ElectronicWings - Hardware Developers Community*. [online] Available at: <https://www.electronicwings.com/ >
4. indiamart.com. 2021. *Sohit 385 Semi-Automatic Pipe Cutting Machine*. [online] Available at: <https://www.indiamart.com/proddetail/sohit-385-semi-automatic-pipe-cutting-machine-21986504233.html>
5. indiamart.com. 2021. *Pipe Cutter Machine*. [online] Available at: <https://www.indiamart.com/proddetail/pipe-cutter-machine-15005718191.html>
6. Ijarse.com. 2018. semi-automatic pipe cutting machine. [online] Available at: <http://www.ijarse.com/images/fullpdf/1519811949_NMCOE4025ijarse.pdf>
7. Tronic.lk. 2021. *TRONIC.LK*. [online] Available at: <https://tronic.lk/>
8. Youtu.be. 2021. [online] Available at: <https://youtu.be/UHLXMufJuk4>
9. Youtu.be. 2021. [online] Available at: <https://youtu.be/ILqOWe3zAbk>
10. Youtu.be. 2021. [online] Available at: <https://youtu.be/w7keb73nNok>

Individual Contribution to the Project

Name of the Student – P.K.D.P.Chathurangi (204027A)

Responsible Components: 1). Ultrasonic Sensors

2). Stepper Motors

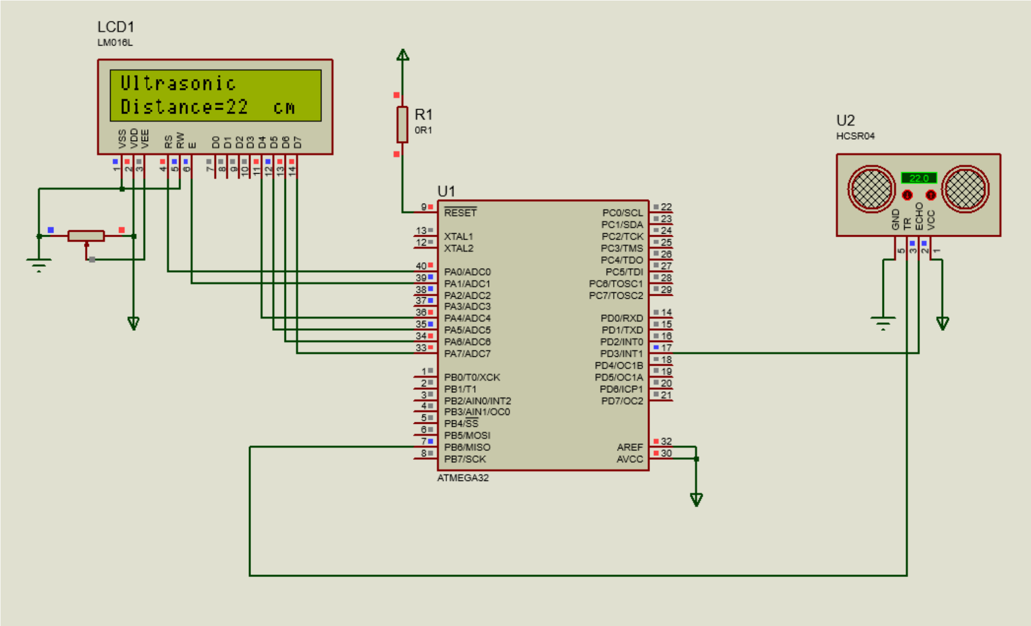
1). Ultrasonic sensor

Ultrasonic sensor is the first component that I am responsible for in our project. We have used two ultrasonic sensors in this machine for tightening process. That are placed in both clutch 1 and clutch 2 to identify the distance to the pipe each side.

Specifications:

- Maximum current: 15mA
- Maximum voltage: 5V
- Range : 2cm-400cm
- Measuring angle : 15 degrees

Schematic diagram for ultrasonic sensor:



Code:

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <string.h>
#include <stdlib.h>
#include "LCD.h"
#include "ultrasonic.h"

int main(void)
{
       LCD_Init();
       initUltrasonic();
 int16_t count_a = 0;
 char show_a[16];
sei();

       while(1)
       {
              count_a =getUltrasonic_2_Distance();
              LCD_String_xy(0, 0, "Ultrasonic");
              itoa(count_a,show_a,10);
              LCD_String_xy(1, 0, "Distance= ");
              LCD_String_xy(1, 9, show_a);
              LCD_String_xy(1, 13, "cm");
              _delay_ms(200);
       }}

static volatile int ult2_count = 0; // integer to access all though the program

static volatile int i = 0;

void initUltrasonic(){
       DDRB |=(1<<PB6);
       _delay_ms(50);
       GICR |= (1<<INT1); //enabling interrupt 0, General Interrupt Control Register
       MCUCR|= (1<<ISC10); //setting interrupt triggering logic change
       TCCR1A = 0;
}

int getUltrasonic_2_Distance(){
       int pulse;
       PORTB |= (1<<PB6);
       _delay_us(20); //triggering the sensor for 15usec
       PORTB &= (~(1<<PB6));
       while(ult2_count==NULL);
       pulse=ult2_count;
       ult2_count=NULL;
       return (pulse/928)+1;
}
```

```
ISR(INT1_vect) //interrupt service routine when there is a change in logic level
{
       if (i==1)//when logic from HIGH to LOW

       {      TCCR1B=0; //disabling counter
              ult2_count=TCNT1;//count memory is updated to intege
              TCNT1=0;//resetting the counter memory
              i=0;}
       if (i==0)//when logic change from LOW to HIGH

       {
              TCCR1B|=(1<<CS10);//enabling counter
              i=1;
       }

}
```

```
//Header files

#ifndef ULTRASONIC_H_
#define ULTRASONIC_H_
#define F_CPU 16000000UL

#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdlib.h>

void initUltrasonic();
int getUltrasonic_2_Distance();

#endif /* ULTRASONIC_H_ */
```

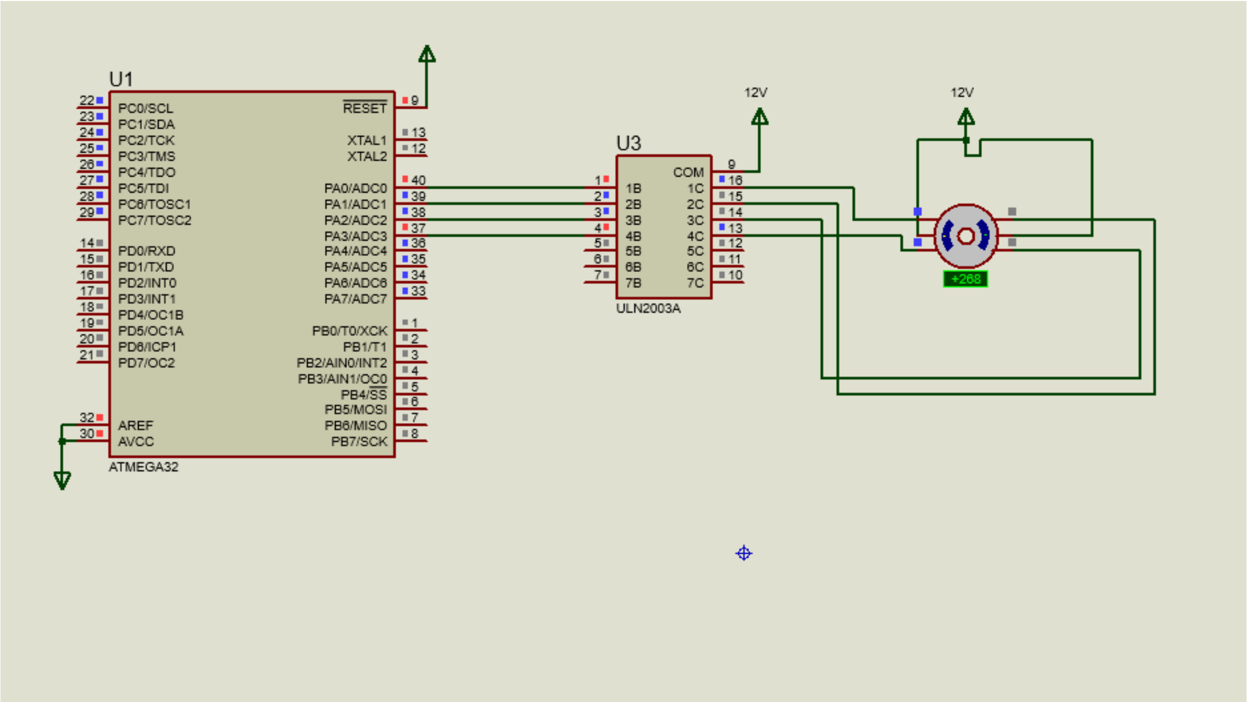
2). Stepper Motor

My second component is Stepper Motor. There are four clutches which are used in the tightening process, and I am responsible for clutch 1 and clutch 2. Here I have used two stepper motors in clutch 1 and clutch 2.

Specifications:

- Maximum current: 1.5A-1.7A
- Maximum voltage: 12 VDC
- Step Angle : 1.8 degree
- Unipolar

Schematic diagram for Stepper Motor:



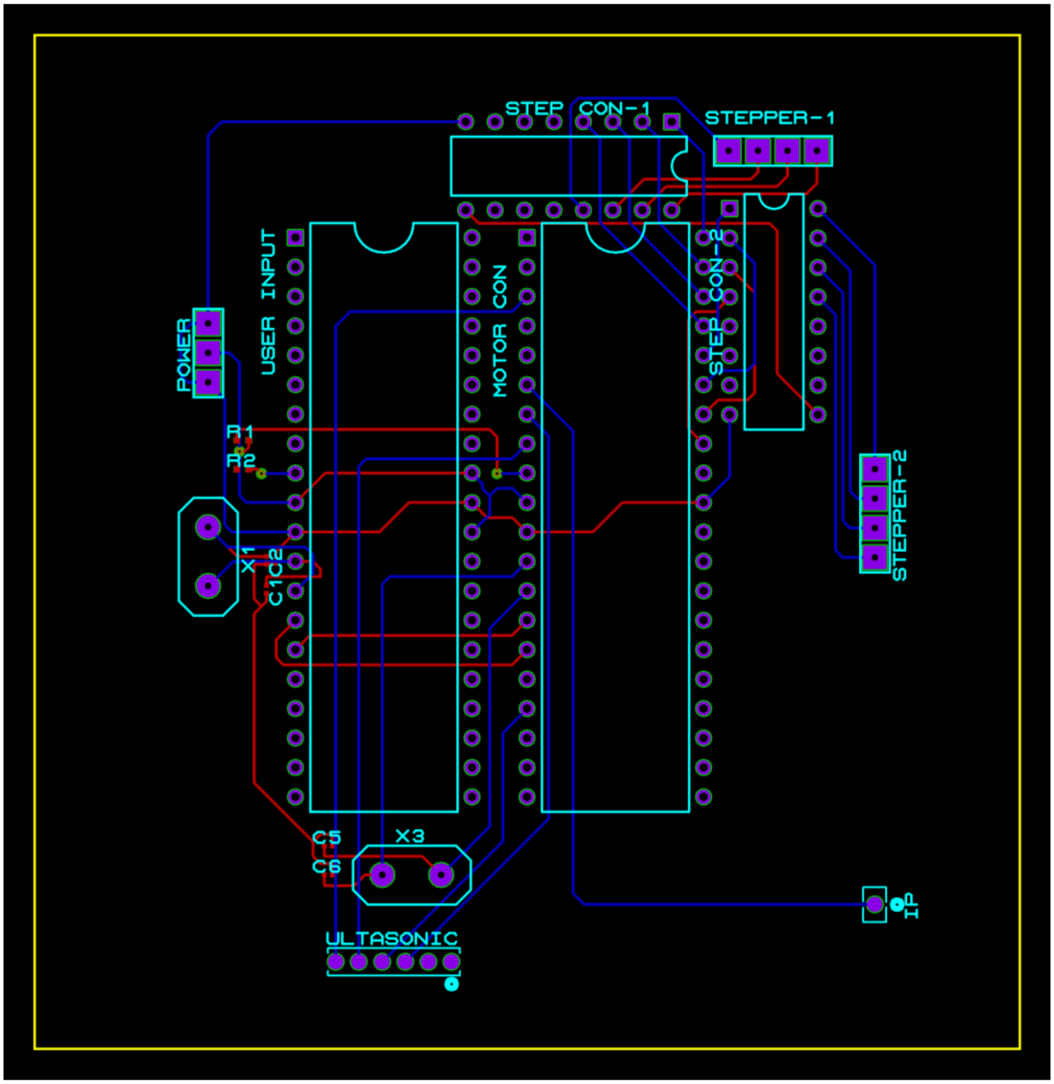
Code:

```
#define F_CPU 16000000UL
#include "stepper.h"

int main(void)
{
        initSteppers();
        rotateStepper_1(1,4);
}
```

```
#include "stepper.h"

void initSteppers(){
        DDRA = 0xFF;
        DDRC = 0xFF;
}

void rotateStepper_1(int direction, float distance)
{
// clockwise-1, Counterclockwise-0, 1 distance unit = 1 rotating

        int period;
        period = 100;
        if (direction == 1)
        {
                for (int i = 0; i < round(distance); i++)
                {
                        PORTA = 0x03;
                        _delay_ms(period);
                        PORTA = 0x06;
                        _delay_ms(period);
                        PORTA = 0x0c;
                        _delay_ms(period);
                        PORTA = 0x09;
                        _delay_ms(period);
                        PORTA = 0x03;
                        _delay_ms(period);
                }
        }
        else
        {
                for (int i = 0; i < round(distance); i++)
                {
                        PORTA = 0x03;
                        _delay_ms(period);
                        PORTA = 0x09;
                        _delay_ms(period);
                        PORTA = 0x0c;
                        _delay_ms(period);
                        PORTA = 0x06;
                        _delay_ms(period);
                        PORTA = 0x03;
                        _delay_ms(period);
                }
        }
}
```

```
// header files

#ifndef STEPPER_H_
#define STEPPER_H_

#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>

void initSteppers();
void rotateStepper_1(int direction, float distance);

#endif /* STEPPER_H_ */
```

PCB for above components:



Name of the Student: P.H.A.E.Madhubhashini (204117B)

Responsible Components: 1). 4 x 4 Keypad

2). IR Sensor

3). Power Supply Unit

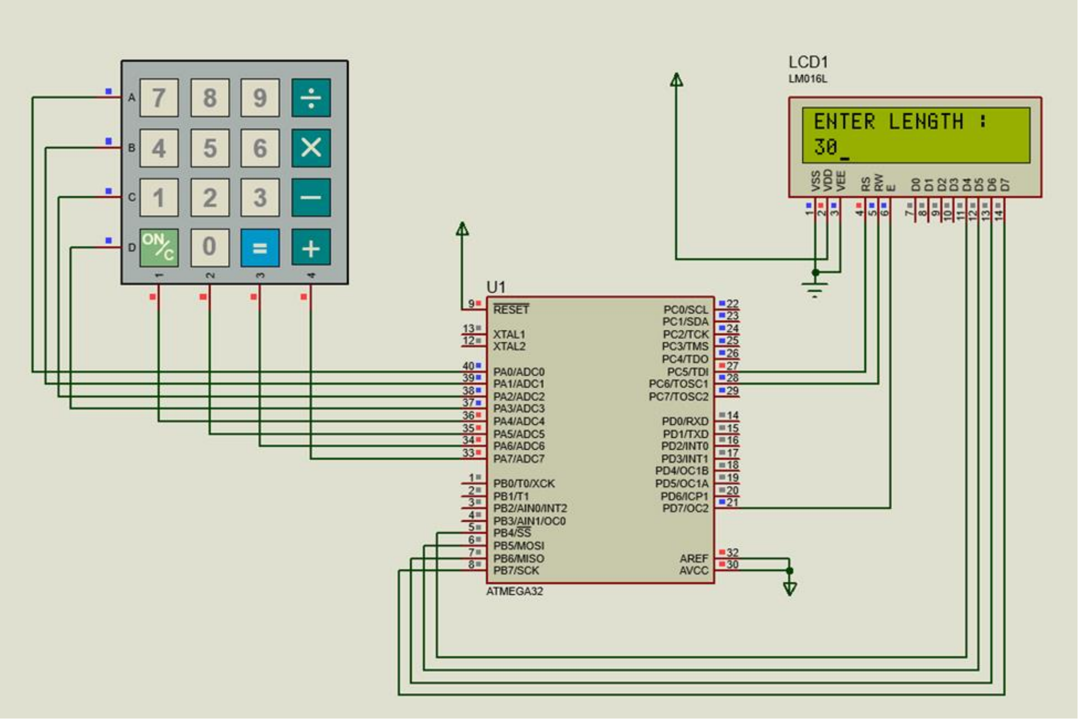
1). 4 x4 Keypad

Keypad is the first component that I am responsible in our project. We have used keypad module in this machine to get two inputs from the user. They are required length of the pipe and the required quantity.

Specifications:

- Maximum current : 30 mA
- Maximum voltage : 24 VDC
- Operating temperature : 0 - 50 ºC

Schematic Diagram (Circuit)



Code:

keypad.h

```
#ifndef KEYPAD_H_
#define KEYPAD_H_

#define F_CPU 16000000UL

#include <avr/io.h>
#include <util/delay.h>

unsigned char keypad();

#endif /* KEYPAD_H_ */
```

keypad.c

```
unsigned char keypad()

{
        PORTA=0b11111110;//check first row
        if((PINA &(1<<PINA4))==0)
        {
                _delay_ms(10);
                return '7';
        }
        else if((PINA & (1<<PINA5))==0)
        {
                _delay_ms(10);
                return '8';
        }
        else if((PINA & (1<<PINA6))==0)
        {
                _delay_ms(10);
                return '9';
        }
        else if((PINA & (1<<PINA7))==0)
        {
                _delay_ms(10);
                return '/';
        }

        PORTA=0b11111101;//check second row
        if((PINA &(1<<PINA4))==0)
        {
                _delay_ms(10);
                return '4';
        }
        else if((PINA & (1<<PINA5))==0)
        {
                _delay_ms(10);
                return '5';

        }
        else if((PINA & (1<<PINA6))==0)
        {
                _delay_ms(10);
                return '6';
        }
        else if((PINA & (1<<PINA7))==0)
        {
                _delay_ms(10);
                return '*';
        }PORTA=0b11111011;//check 3rd row

        if((PINA &(1<<PINA4))==0)
        {
                _delay_ms(10);
                return '1';
        }
        else if((PINA & (1<<PINA5))==0)
        {
                _delay_ms(10);
                return '2';
        }
        else if((PINA & (1<<PINA6))==0)
        {
                _delay_ms(10);
                return '3';
        }
        else if((PINA & (1<<PINA7))==0)
        {
                _delay_ms(10);
                return '-';
        }

        PORTA=0b11110111;//check 4th row
        if((PINA &(1<<PINA4))==0)
        {
                _delay_ms(10);
                return 'C';
        }
        else if((PINA & (1<<PINA5))==0)
        {
                _delay_ms(10);
                return '0';
        }
        else if((PINA & (1<<PINA6))==0)
        {
                _delay_ms(10);
                return 'E';
        }
        else if((PINA & (1<<PINA7))==0)
        {
                _delay_ms(10);
                return '+';
        }
}
```

main.c

```
#define F_CPU 16000000UL

 #include <avr/io.h>
 #include <util/delay.h>
 #include "lcd.h"
 #include "keypad.h"

 unsigned char x;
 unsigned char y;
 unsigned char z;
 int x1=0 ; // variable for length
 int x2=0; // variable for quantity

int main(void)
{
   DDRA=0x0F;
   lcd_init();
   lcd_print("  HELLO USER ");
   _delay_ms(30);
   lcdcmd(0x01);
   lcdcmd(0x0E);
   _delay_ms(30);
   lcd_print("-- ICT PROJECT --");
   _delay_ms(30);
   lcdcmd(0x01);
   lcdcmd(0x0E);
   _delay_ms(30);
   lcd_print("  GROUP - 39");
   label1:
   _delay_ms(100);
   lcdcmd(0x01);
   lcdcmd(0x0E);
   _delay_ms(100);
   lcd_print("ENTER LENGTH :");
   lcdcmd(0xC0);

    do
     {
       PORTA=0xF0;
       if(PINA!=0xF0)
       {
        x=keypad();
        if(x=='/'||x=='*'||x=='-'||x=='C'||x=='+')
          {
             lcdcmd(0x01);
             lcdcmd(0x0E);
             _delay_ms(30);
             lcd_print("INVALID INPUT");
              _delay_ms(30);
              goto label1;
              }
        else if(x!='E')
           {
               x1=(x1*10)+(x-'0');//read length      }
lcddata(x);
              }
       }
}while(x!='E');

if(x1<=10 || x1>150)
 {
        lcdcmd(0x01);
        lcdcmd(0x0E);
        _delay_ms(30);
        lcd_print("INVALID LENGTH");
        _delay_ms(30);
}
    label2:
    lcdcmd(0x01);
    lcdcmd(0x0E);
    _delay_ms(30);
    lcd_print("ENTER QUANTITY :");
    _delay_ms(30);
    lcdcmd(0xC0);
do
   {
       PORTA=0xF0;
       if(PINA!=0xF0)
       {
       y=keypad();
if(y=='/'||y=='*'||y=='-'||y=='+')
   {
       lcdcmd(0x01);
       lcdcmd(0x0E);
      _delay_ms(30);
       lcd_print("INVALID INPUT");
      _delay_ms(30);
       goto label2;
       }
else if(y!='E')//read the quantity
       {
          x2=y-'0';
          lcddata(y);
       }
       }
   }while(y!='E');

if(y=='E')
{
       lcdcmd(0x01);
       lcdcmd(0x0E);
      _delay_ms(30);
       lcd_print("    DONE");
      _delay_ms(100);
       lcdcmd(0x01);
      _delay_ms(500);
}

return 0;
```

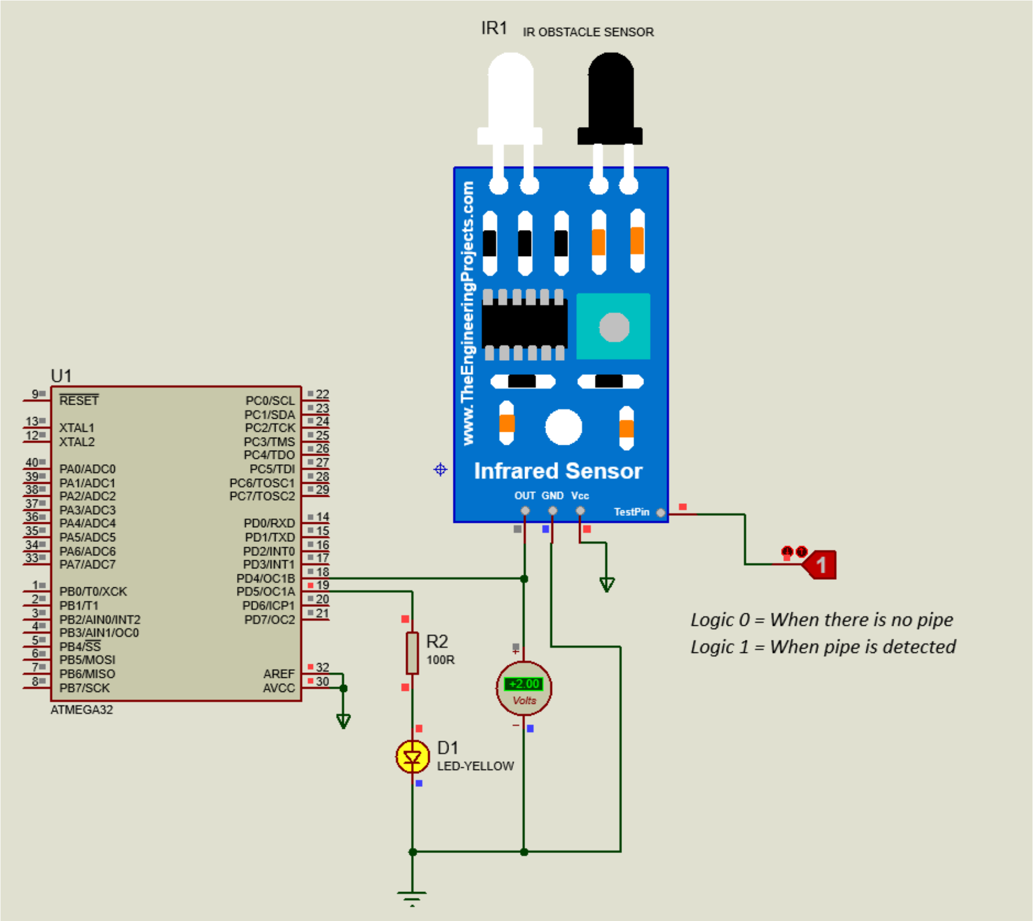
2). IR Sensor

My next component is IR sensor. We have used this IR sensor in two different stages of this machine. IN the Identification process, this IR sensor is used to detect the presence of the pipe and at the final stage which is wastage separating and collecting process we have used this same IR sensor to identify the wastage.

Specifications:

- Supply current : 20 mA
- Operating voltage : 5 VDC
- Range : up to 20 cm

Schematic Diagram (Circuit)

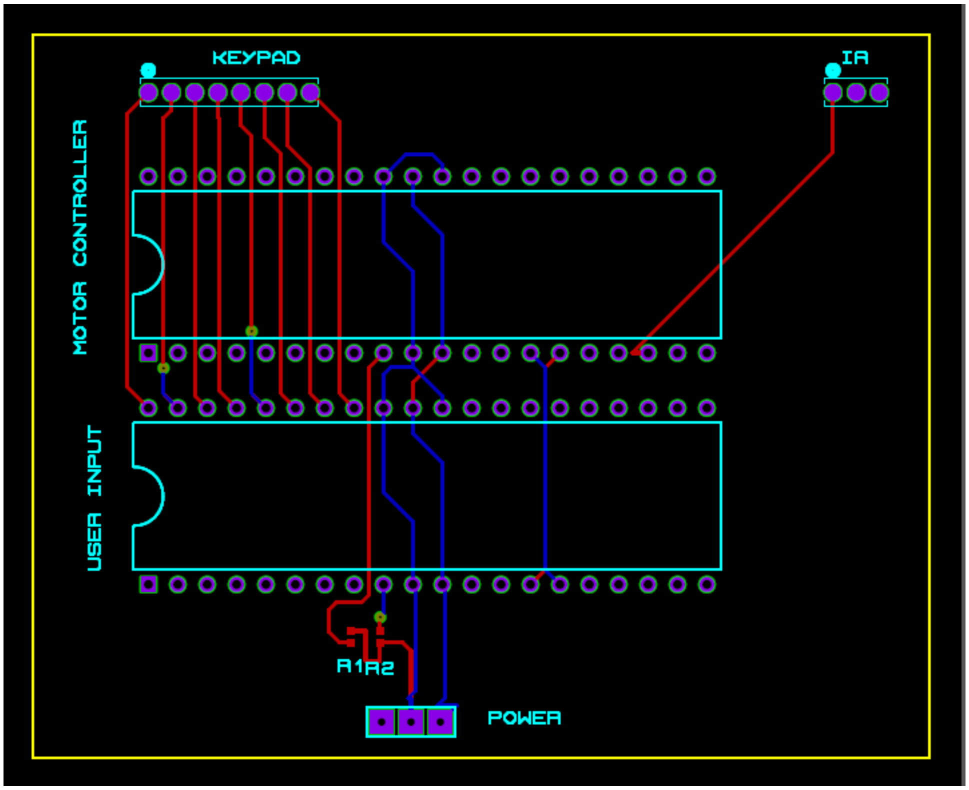


Logic 0 = When there is no pipe
Logic 1 = When pipe is detected

Code :

```
#define F_CPU 16000000UL
#include <avr/io.h>
int main()
{
PORTD = 0x00;
DDRD  = 0x20; //00100000 set PIND5 as output pin
 while (1)
     {
              if((PIND & 0x10) != 0) //read the logic level on PIND4
                  PORTD |= 0b00100000;
              else
                  PORTD &= 0b11011111;
     }
}
```

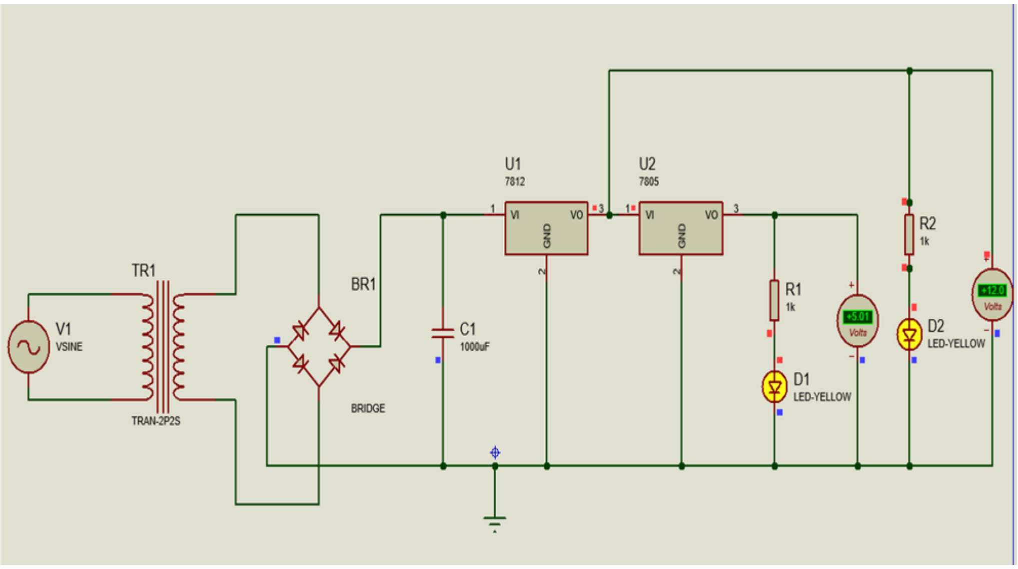
PCB Design (Keypad + IR sensor)



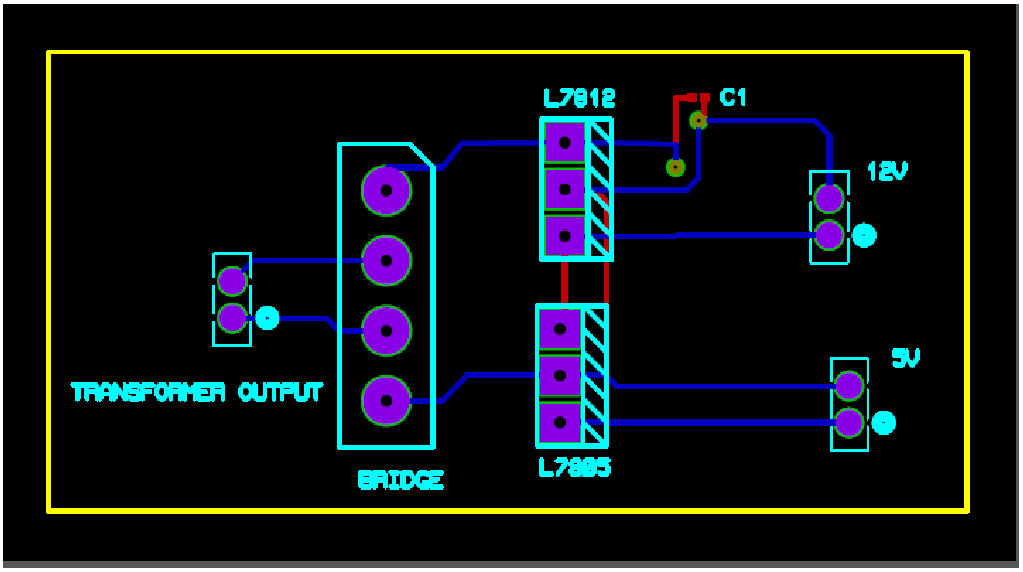
3). Power Supply Unit

Designing a power supply unit for our machine is my next responsibility. Since we have used several sensors and different types of motors for this machine we have to convert 230 VAC domestic power supply to various DC values such as 5VDC and 12VDC according to their need. I have used a step-down transformer, bridge, capacitors and regulators to design the power supply circuit and the whole process happens in the circuit is known as "Full Rectification".

Schematic Diagram



PCB Design



Name of the Student: D.S.Rathnasinghe (204178K)

Responsible Components: 1). DC Motor
2). IR Sensor
Others: 3). USART Communication

1). DC Motor

DC motor is the first component which I am responsible in our project. We have used this DC motor in the rail system of our machine.

Technique: we have designed a rail system by using two dc motors at both sides of machine.

Specifications

- Maximum current: 1.5A
- Maximum voltage: 20 VDC
- Motor Type :775
- No Load Speed : 12,000 RPM @ 12V.

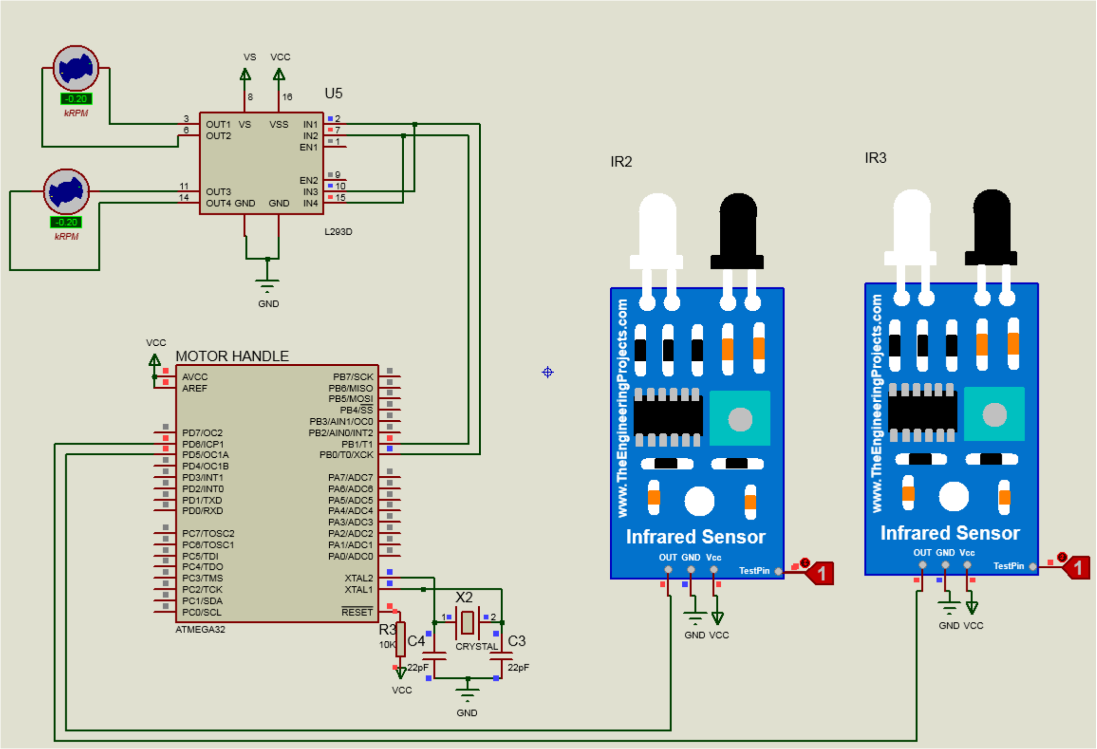
2). IR sensor

My second component is IR Sensor which is used for aligning process. Here we are using two IR sensors to get the align the pipe edge to the cutting point.

Specifications:

- Maximum current: 20 mA
- Maximum voltage: 5 VDC

Schematic diagram for above components:



Code for above components:

Header file:

```
#ifndef DCMOTOR_H_
#define DCMOTOR_H_
#define F_CPU 16000000UL
#include <avr/io.h>

void rotateDCMotors(int direction);
void stopDCMotors();
void initDCMotors();

#endif /* DCMOTOR_H_ */
```

.c file:

```
#define F_CPU 16000000UL
#include "dcmotor.h"

void initDCMotors(){
       DDRB |= (1<<PB0) | (1<<PB1);
}
```

```
void rotateDCMotors(int direction)
{
       if (direction != 0)
       {// Rotate DC motor Clockwise
              PORTB &= ~(1 << PB0);
              PORTB |= (1 << PB1);
       }
       else
       {// Else rotate DC motor Counterclockwise
              PORTB |= (1 << PB0);
              PORTB &= ~(1 << PB1);
       }
}

void stopDCMotors()
{
       // stop Rotation DC motor
       PORTB &= ~(1 << PB0);
       PORTB &= ~(1 << PB1);
}
```

Main file:

```
/*
 * dasith.c
 *
 * Created: 5/24/2022 2:56:44 AM
 * Author : Dasith
 */

#include <avr/io.h>

#include "dcmotor.h"

int main(void)
{
    initDCMotors();
       while (1)
    {

              if ((PIND & 1 << PD5) && (PIND & 1 << PD6)){ // if both Ir sensor-2,3 are detected pipe
                     rotateDCMotors(1);
                     while (PIND & (1 << PD6));
              }
              else if (!(PIND & 1 << PD5) && !(PIND & 1 << PD6)){ // else if both Ir sensor-2,3 are not detected pipe
                     rotateDCMotors(0);
                     while (!(PIND & (1 << PD5)));
              }
              stopDCMotors();
    }
}
```

3). USART Communication

As we are using two microcontrollers it is necessary to build a communication between them. we have used USART communication for that. I have developed the code to satisfy our requirement here.

Code in ‘user input’ microcontroller:

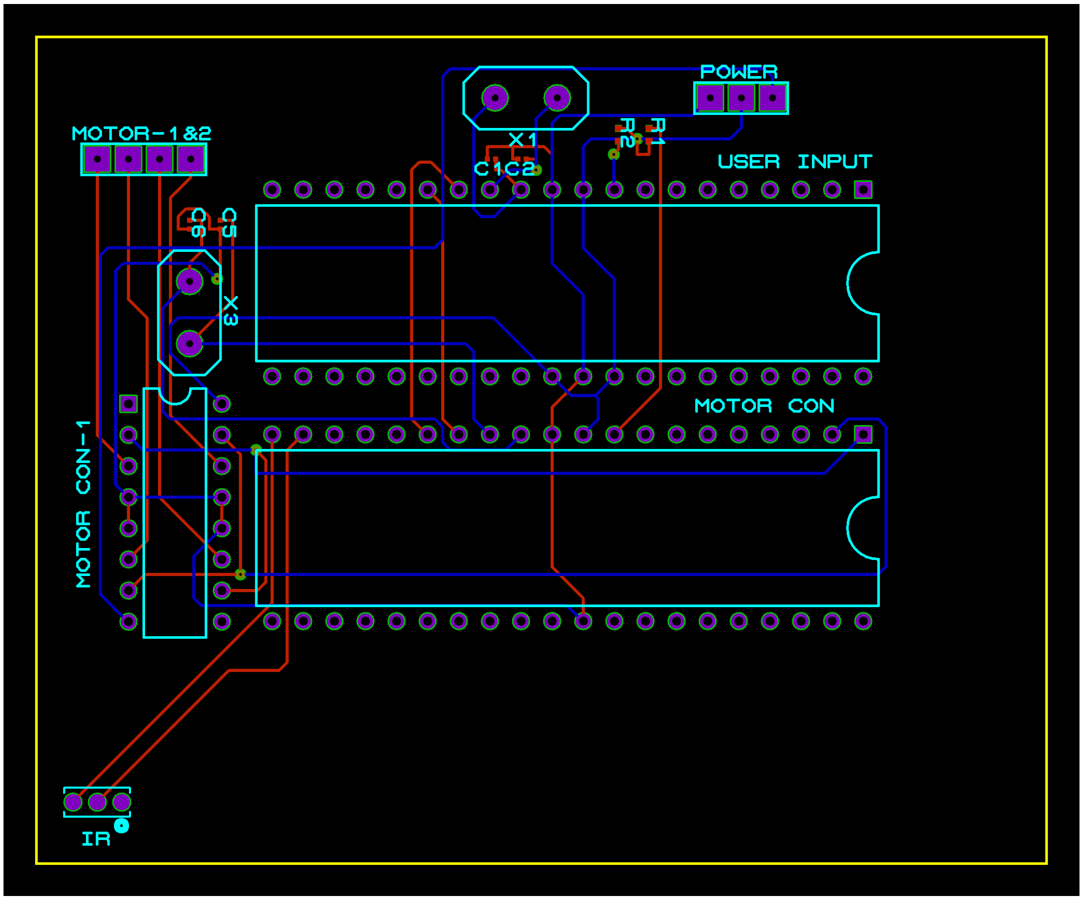
Usart.h file:

```
#ifndef USART_H_
#define USART_H_

#define F_CPU 16000000UL
#define BAUD_PRESCALE(USART_BAUDRATE) (((F_CPU / (USART_BAUDRATE * 16UL))) - 1)

#include <avr/io.h>
#include <util/delay.h>
#include <stdlib.h>

void initUSART(long USART_BAUDRATE);
void sendData(uint8_t byte);
uint8_t getReceivedData();

#endif /* USART_H_ */
```

Usart.c file:

```
#include "usart.h"

void initUSART(long USART_BAUDRATE){

       DDRD |= 1 << PIND1;//pin1 of portD as OUTPUT
       //Put the upper part of the baud number here (bits 8 to 11)
       UBRRH = (unsigned char) (BAUD_PRESCALE(USART_BAUDRATE) >> 8);
       //Put the remaining part of the baud number here

       UBRRL = (unsigned char) BAUD_PRESCALE(USART_BAUDRATE);
       //Enable the receiver, transmitter
       UCSRB = (1 << RXEN) | (1 << TXEN) | (1<<RXCIE);
       //Set 2 stop bits and data bit length is 8-bit
       UCSRC = (1 << USBS) | (1 << UCSZ0) | (1<<UCSZ1);

}

void sendData(uint8_t byte){
       while (! (UCSRA & (1 << UDRE)) );

       {
              UDR = byte;//once transmitter is ready sent eight-bit data
       }
       // Get that data out here!
       _delay_ms(5);
}
```

PCB design for the above components:



Name of the Student: J.D.A. Shalinda (204200A)

Responsible Components: 1). Stepper Motor (For Cutting Arm)
2). DC Motor (For Cutting Blade)
3). Inductive Proximity Sensor

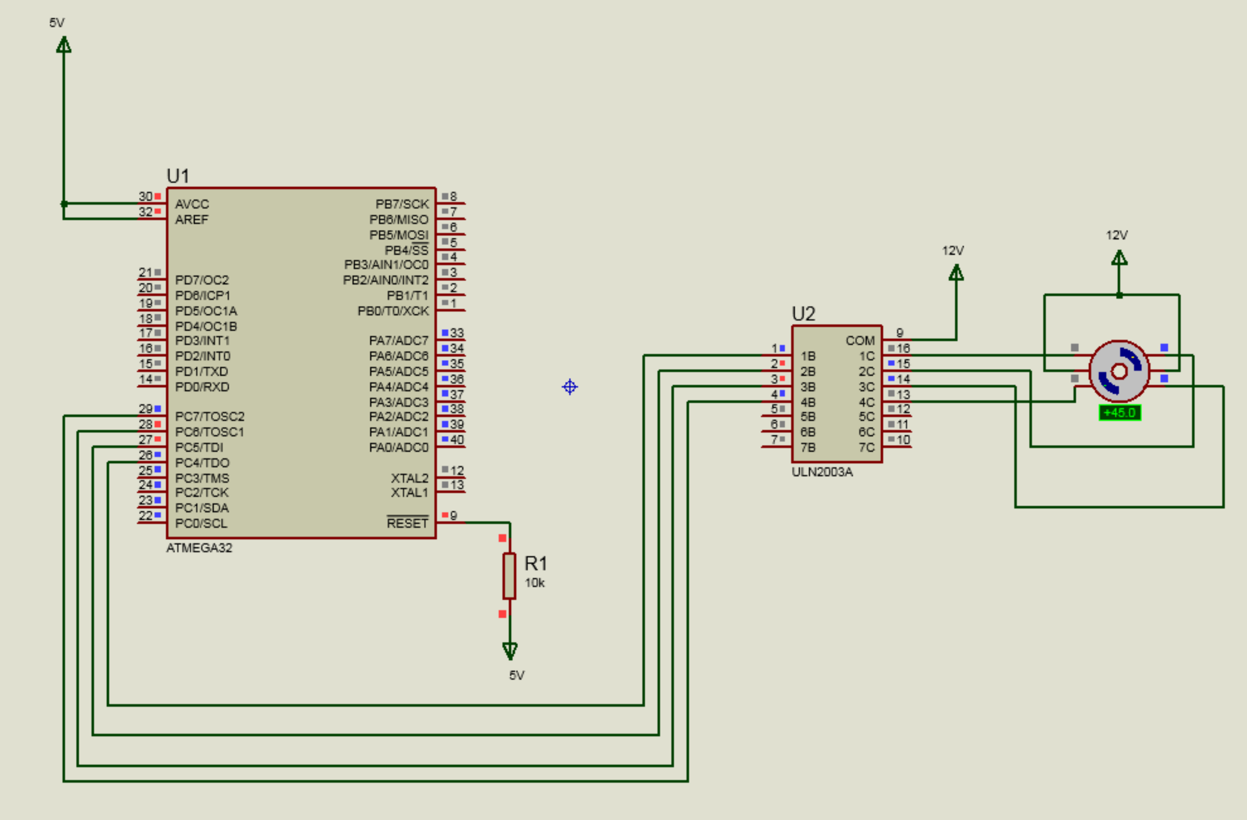
1). Stepper Motor: NEMA 17

Stepper Motor is the first component which I am responsible in our project. We have used this stepper motor in our machine to move the cutting arm up and down at the cutting point.

Specifications:

- Maximum voltage: 12 VDC
- Maximum current: 1.5A-1.7A
- Step angle: 1.8º degree
- Holding torque: 3.2 kg - cm
- Unipolar

Schematic diagram for Stepper Motor:



Code:

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>

void initSteppers(){

      DDRC = 0xFF;
}

void rotateStepper_4(int direction){
      // clockwise-1
      // Counterclockwise-0
      // 1 distance unit = 1 rotating

      int period;
      period = 100;
      if (direction == 1)
      {
            /* Rotate  clockwise */
            PORTC = 0x30;
            _delay_ms(period);
            PORTC = 0x60;
            _delay_ms(period);
      }
      else
      {
            /* Rotate  Counterclockwise  */
            PORTC = 0x60;
            _delay_ms(period);
            PORTC = 0x30;
            _delay_ms(period);
      }
}

int main(void)
{
      initSteppers();
      rotateStepper_4(1);
      _delay_ms(2000);
      rotateStepper_4(0);
}
```

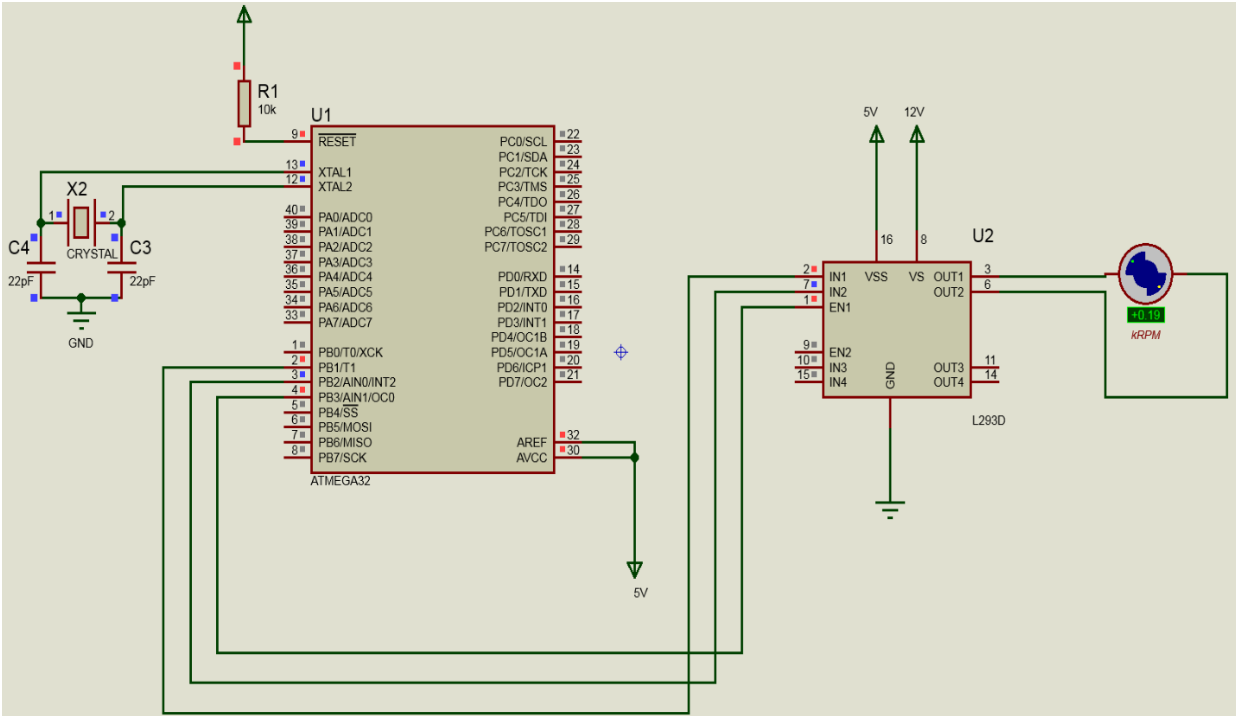
2). DC Motor

My second component is DC motor which is placed at the upper part of the cutting arm. We have used this DC motor to control the rotation of the cutting blade.

Specifications:

- Maximum current: 1.5A
- Maximum voltage: 20VDC
- No load speed: 12 000 RPM @12V
- Stall torque: 79Ncm @14.4 V

Schematic diagram for DC Motor:



Code:

DC motor.h file:

```
#ifndef DCMOTOR_H_
#define DCMOTOR_H_
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>

void initDCMotor();
void setMotorSpeed(int bladeMotorSpeed);
void rotateMotor();
void stopMotor();

    #endif /* DCMOTOR_H_ */

#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>

#include "dcmotor.h"
```

```
DCmotor.c file:

#include "dcmotor.h"

void initDCMotor(){
       DDRB |= (1 << PB3) | (1<<PB2) | (1<<PB1);
       TCCR0  = (1 << WGM00) | (1 << WGM01) | (1 << COM01) | (1 << CS00);
       OCR0 = 255;
}
void setMotorSpeed(int bladeMotorSpeed){
       OCR0 = bladeMotorSpeed;
}

void rotateMotor(){
       PORTB |=(1<<PB1);
}
void stopMotor(){
       PORTB &=~(1<<PB1);
}
```

```
main.c file:

int main()
{
       initDCMotor();
       _delay_ms(2000);
       rotateMotor();
       _delay_ms(2000);
       stopMotor();
       _delay_ms(2000);
       setMotorSpeed(10);
       rotateMotor();
       _delay_ms(2000);
       stopMotor();
       _delay_ms(2000);
}
```

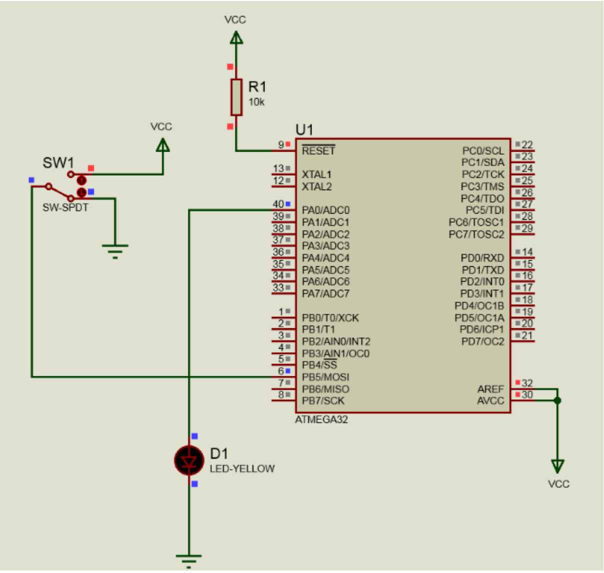
3). Inductive Proximity sensor

Inductive Proximity sensor is my next component and we have used it in the detecting and identification process to identify the material of the pipe. If the material of the pipe is metal, then its output becomes high and set a higher speed to the DC motor and if the material of the pipe is non-metal (PVC) its output becomes low and sets a lower speed to the DC motor.

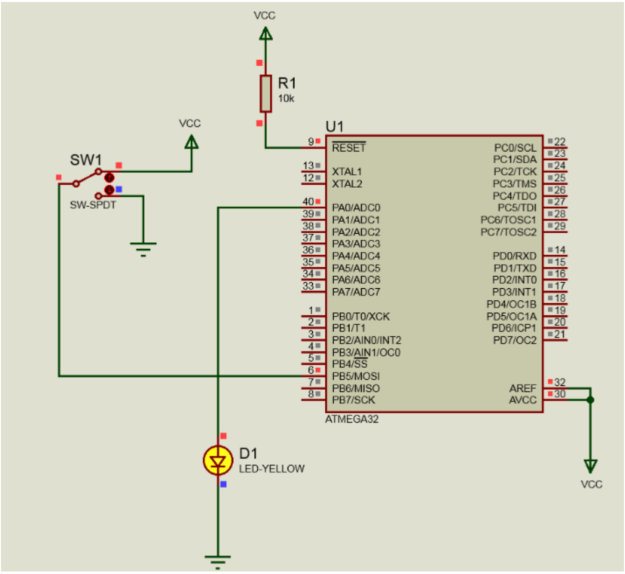
Specifications:

- Maximum voltage: 30VDC
- Maximum current: 200 mA
- Output type: PNP
- Temperature: -25……70 ºc

Schematic diagram for **Inductive Proximity sensor**:



For PVC Pipe

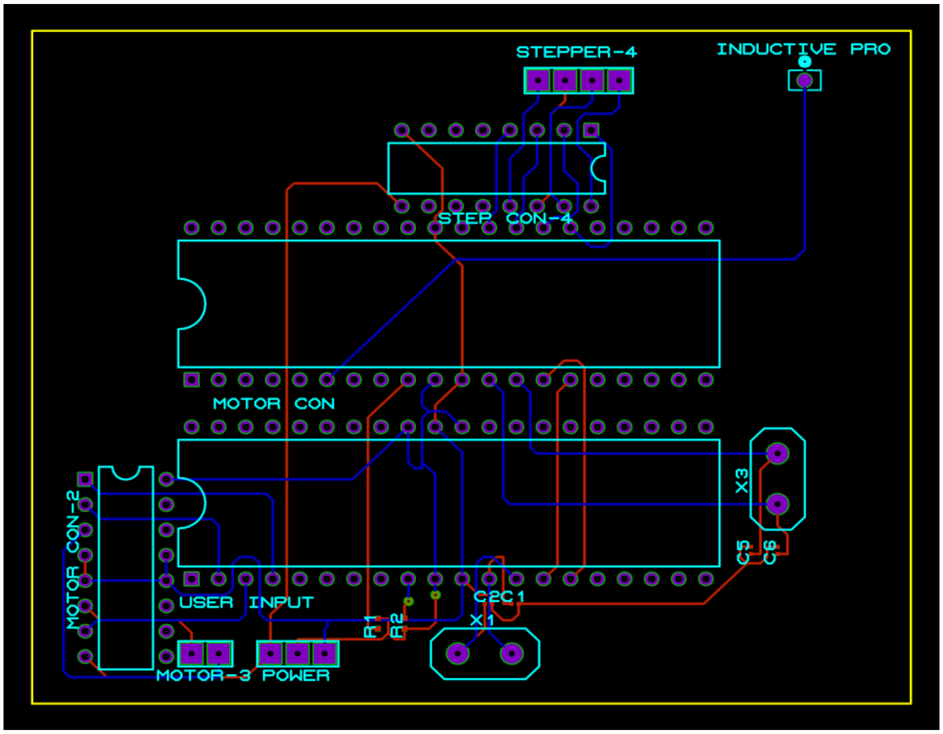


For Metal Pipe

Code:

```
#define F_CPU 16000000UL
#include <avr/io.h>
int main()
{

        DDRA=(1<<PA0);
        int isMetal;

        while (1)
        {
                if (PINB & 1 << PB5){ // Change DC motor speed
                        // metal
                        isMetal = 1;
                        //sendData(HIGH_SPEED);

                        }else{
                        // PVC
                        isMetal = 0;
                        //sendData(LOW_SPEED);
                }
                //for demonstration only
                if(isMetal){
                        PORTA |=(1<<PA0);
                }else{
                        PORTA &=~(1<<PA0);
                }
        }
}
```

PCB design for above components:



Name of the Student: K.M.Tharangana (204215A)

Responsible Components: 1). 16x2 LCD Display

2). Ultrasonic Sensor

3). Stepper Motor

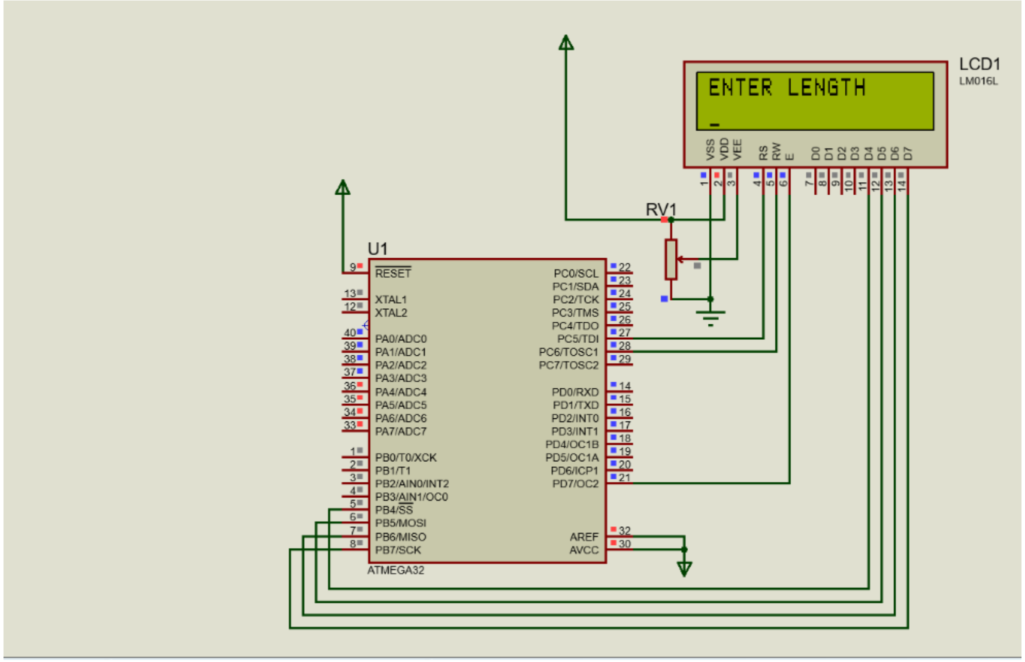
1). LCD Display

16 x2 LCD Display is the first component that I am responsible in our project. This has been used to display commands to the user. It directs the user to enter the needed pipe length and amount.

Specifications:

- Current range : 1.5 - 2.5 mA
- Voltage range : 4.8 - 5.3 VDC
- Operating temperature: 0 - 50 ⁰C

Schematic diagram for LCD Display:



Code:

Lcd.h file:

```
#ifndef LCD_H_
#define LCD_H_
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>

#define LCD PORTB
#define EN 7
#define RW 6
#define RS 5

void lcd_init();
void lcdcmd(unsigned char cmd);
void lcddata(unsigned char data);
void lcd_print(char *str);

#endif /* LCD_H_ */
```

Lcd.c file:

```
#include "lcd.h"

void lcdcmd(unsigned char cmd)
{
       PORTC&=~(1<<RS);
       PORTC&=~(1<<RW);
       LCD=cmd & 0xF0;
       PORTD|=(1<<EN);
       _delay_ms(20);
       PORTD&=~(1<<EN);

       LCD=cmd<<4;
       PORTD|=(1<<EN);
       _delay_ms(20);
       PORTD&=~(1<<EN);
}

void lcd_print(char *str)//function to
print a given string
{
       unsigned char i=0;
       while(str[i]!=0)
       {
              lcddata(str[i]);
              i++;
       }
}
```

```
void lcddata(unsigned char data){
//function to display input data
       PORTC|=(1<<RS);
       PORTC&=~(1<<RW);
       LCD=data & 0xF0;
       PORTD|=(1<<EN);
       _delay_ms(20);
       PORTD&=~(1<<EN);

       LCD=data << 4;
       PORTD|=(1<<EN);
       _delay_ms(20);
       PORTD&=~(1<<EN);}
void lcd_init()//function to LCD
initialization{
       DDRC=0xFF;
       DDRD|=(1<<EN);
       PORTD&=~(1<<EN);
       lcdcmd(0x33);
       lcdcmd(0x32);
       lcdcmd(0x28);
       lcdcmd(0x0E);
       lcdcmd(0x01);
       _delay_ms(20);
}
```

Main,c file:

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#include "lcd.h"

int main(void)
{
DDRA=0x0F;
lcd_init();
lcd_print("  HELLO USER ");
_delay_ms(100);
lcdcmd(0x01);//clear display
lcdcmd(0x0E);//blink the cursor
_delay_ms(100);
lcd_print("ENTER LENGTH");
lcdcmd(0xC0);//bring cursor to second
row
delay_ms(500);
lcdcmd(0x08);//off the lcd

return 0;
}
```

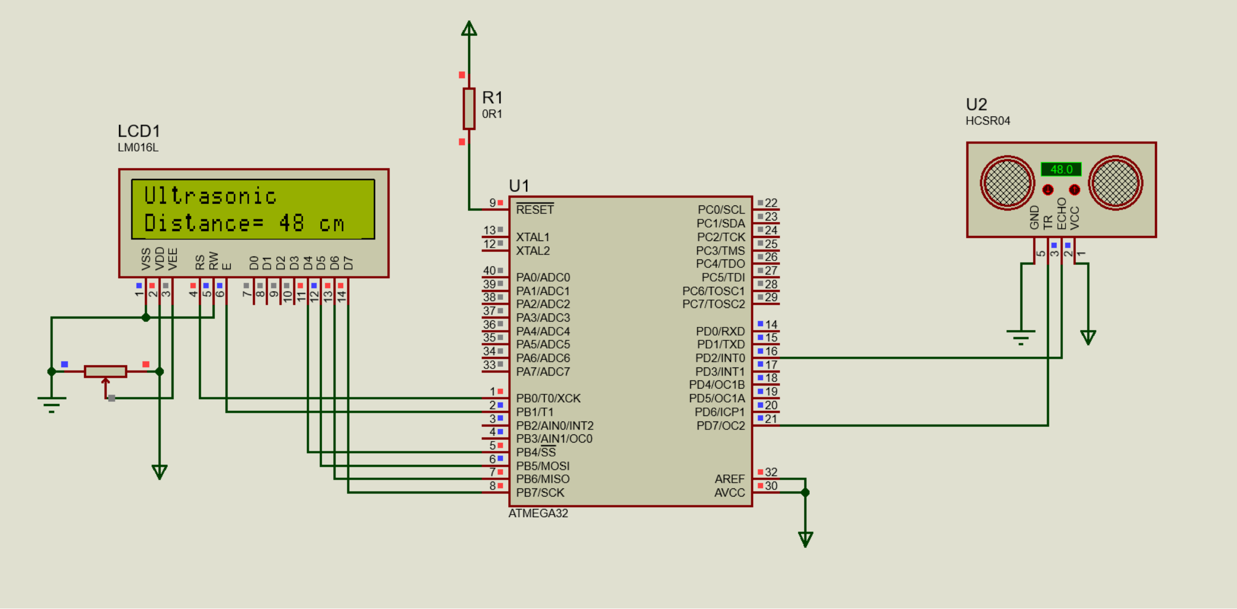
2.Ultrasonic sensor – HCSR04 module

Ultrasonic sensor is the second component of our project for which I am accountable. This sensor is used in the pipe length measurement procedure. We're looking at the distance between clutch 1 and one of the machine's edges. (It gives the pipe length which went ahead in an indirect way) When the specified length is reached, the sensor will interrupt to the microcontroller.

Specifications:

- Maximum current: 15mA
- Maximum voltage: 5 VDC
- Range : 2cm-400cm
- Measuring angle : 15 degrees

Schematic diagram for ultrasonic sensor:



(LCD is only used to check the correct results)

Code:

Ultrasonic.h file:

```
#ifndef ULTRASONIC_H_
#define ULTRASONIC_H_

#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdlib.h>

void initUltrasonic();
int getUltrasonic_1_distance();
#endif /* ULTRASONIC_H_ */
```

Ultrasonic.c file:

```
#include "ultrasonic.h"

static volatile int pulse = 0; // integer to access all though the program
static volatile int i = 0;
int timerOverFlow=0;
static volatile int ult_count=NULL;

void initUltrasonic(){
GICR |= (1<<INT0); //enabling interrupt 0-General Interrupt Control Register
MCUCR|= (1<<ISC00); //setting interrupt triggering logic change- MCU control Register
TIMSK|=(1<<TOIE0);
TCCR0 = 0;
}

int getUltrasonic_1_distance(){
       int pulse;
       PORTD |= (1<<PIND7);
       _delay_us(15); //triggering the sensor for 15usec
       PORTD &= (~(1<<PIND7));
       while(ult_count==NULL);
       pulse=ult_count;
       ult_count=NULL;
       return (pulse/58)+1;
}

ISR(INT0_vect) //interrupt service routine when there is a change in logic level

{
if (i==1)//when logic from HIGH to LOW
       {
TCCR0=0; //disabling counter
ult_count=TCNT0+timerOverFlow;//count memory is updated to integer
TCNT0=0;//resetting the counter memory
i=0;

}
```

```
       if (i==0)//when logic change from LOW to HIGH
       {
            TCCR0|=(1<<CS10);//enabling counter
            i=1;
       }
       timerOverFlow=0;
}
ISR(TIMER0_OVF_vect)
{
       timerOverFlow+=256;
}
```

main.c file:

```
/*
 * ultra-sonic_01.c
 *
 * Created: 12/3/2021 11:21:51 AM
 * Author : milanka
 */

#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <string.h>
#include <stdlib.h>
#include "ultrasonic.h"
#include "LCD.h"

int main(void)
{
       LCD_Init();
       DDRD = 0b11111011;
       _delay_ms(50);
       initUltrasonic();
    static int16_t count_a = 0;
    char show_a[16];
       sei();

       while(1)
       {

              count_a =getUltrasonic_1_distance();
              LCD_String_xy(0, 0, "Ultrasonic");
            itoa(count_a,show_a,10);
              LCD_String_xy(1, 0, "Distance= ");
              LCD_String_xy(1, 10, show_a);
              LCD_String_xy(1, 13, "cm");
              _delay_ms(200);

       }
}
```

3). Stepper Motor – Nema17

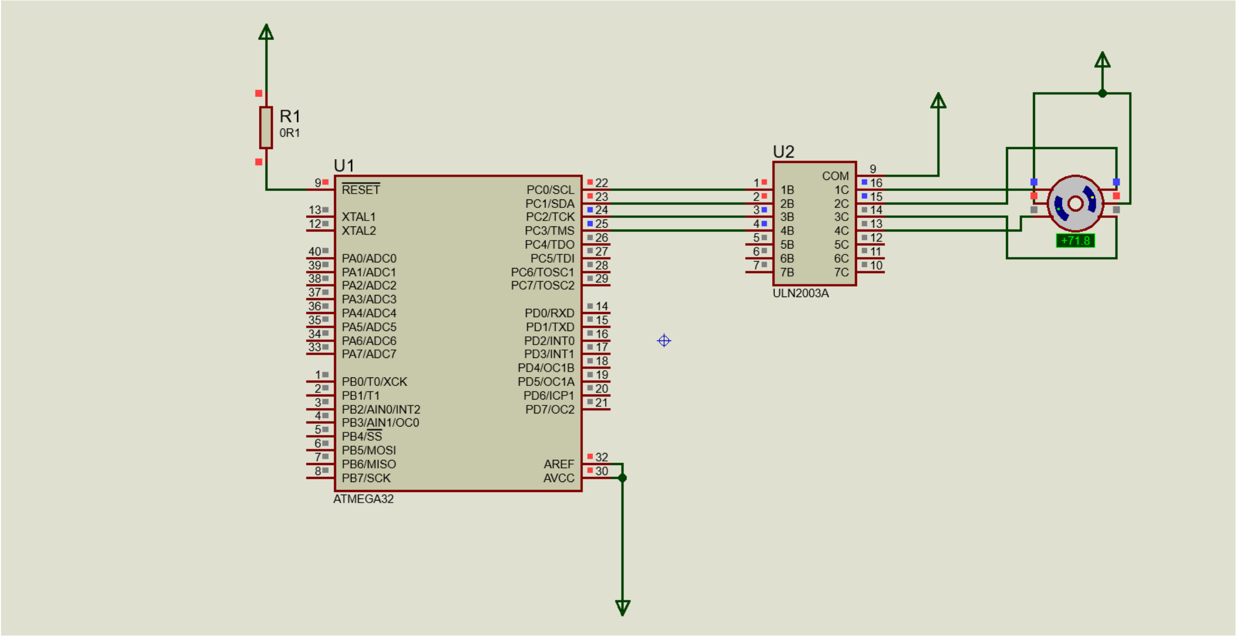
Stepper motor is the next component which I am responsible in our project. We have used this stepper motor in clutch 3 and it will be used in tightening process. This motor takes distance to the pipe as an input and rotate accordingly.

Technique: The stepper motor is used to rotate a screw rod. The clutch will slide in or out with the screw rod due to the internal structure's design.

Specifications:

- Current range: 1.5A-1.7A
- Maximum voltage: 12 VDC
- Step angle: 1.8⁰ degree
- Holding torque: 3.2 kg - cm
- Unipolar

Schematic diagram for stepper motor:



Code:

Stepper.h file:

```
#ifndef STEPPER_H_
#define STEPPER_H_
#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

void initSteppers();
void rotateStepper_3(int direction, float distance);
#endif /* STEPPER_H_ */
```

Stepper.c file:

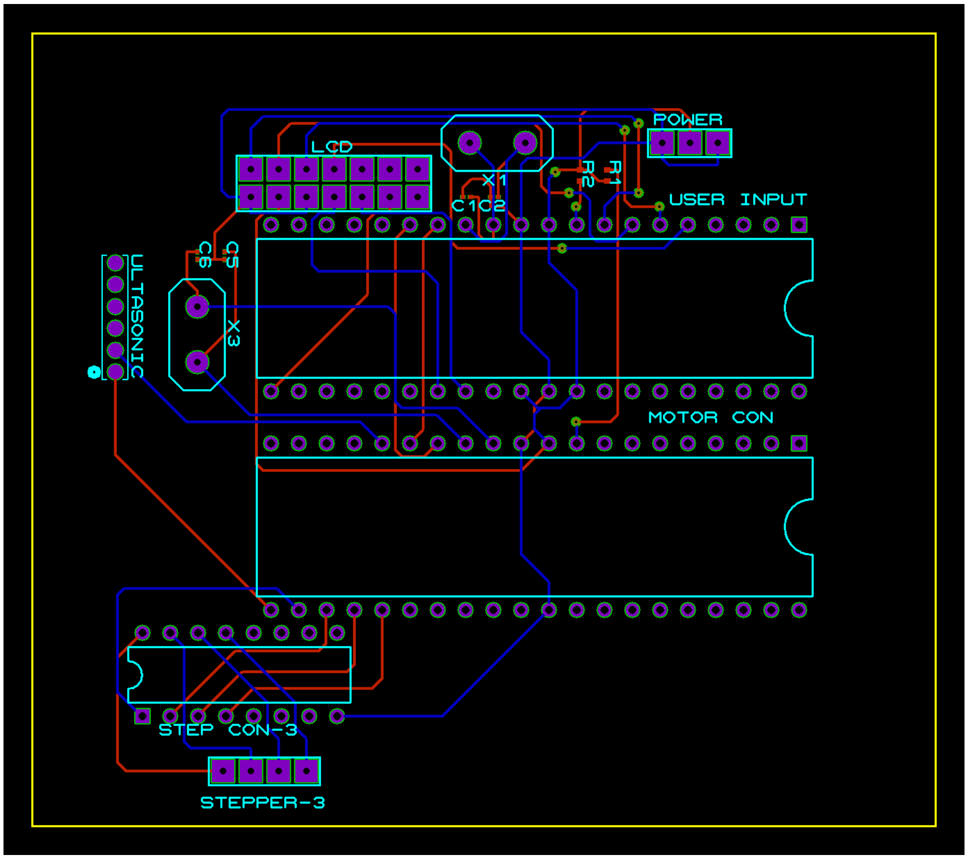
```
#include "stepper.h"
void initSteppers(){
       DDRA = 0xFF;
       DDRC = 0xFF;
}
void rotateStepper_3(int direction, float distance){
       // clockwise-1, Counterclockwise-0, 1 distance unit = 1 rotating
       int period;
       period = 100;
       if (direction == 1)
       {
              for(int i=0;i<distance;i++)//Rotate  clockwise
              {
                     PORTC = 0x03;
                     _delay_ms(period);
                     PORTC = 0x06;
                     _delay_ms(period);
                     PORTC = 0x0c;
                     _delay_ms(period);
                     PORTC = 0x09;
                     _delay_ms(period);
                     PORTC = 0x03;
                     _delay_ms(period);
              }

              _delay_ms(10);
       }
       else
       {
              for (int i = 0; i < distance; i++)//Rotate Counterclockwise
              {
                     PORTC = 0x03;
                     _delay_ms(period);
                     PORTC = 0x09;
                     _delay_ms(period);
                     PORTC = 0x0c;
                     _delay_ms(period);
                     PORTC = 0x06;
```

```
                    _delay_ms(period);
                    PORTC = 0x03;
                    _delay_ms(period);
              }
              _delay_ms(10);
        }
}
```

Main.c file:

```
#define F_CPU 16000000UL
#include <avr/io.h>

#include "stepper.h"

int main(void){

       initSteppers();
       rotateStepper_3(1,3);
       rotateStepper_3(0,1);
}
```

PCB design for above components:


POWER
LCD
X1
C1C2
R2
R1
USER INPUT
C6
C5
X3
ULTASONIC
MOTOR CON
STEP CON-3
STEPPER-3