Faculty of Information Technology

IN 1900-ICT Project

Automatic Coconut breaking, scraping, and grinding machine

Group No: 15

Nusky M.N.N. -204146L

Banu A.G.S. -204018X

Karunaweera R.L. -204096G

Kumarasingha H.J.A. -204108A

Lamahewage D.R. -204114M

Supervisor's Name: BH Sudantha
Dean/Senior lecturer
Faculty of Information Technology

Date of Submission: 01.10.2021

Signature of Supervisor:

Table of Contents

Table of figures

Section 1: Introduction

This project aims at designing an Automatic Coconut breaking, scraping, and grinding machine to enable easy and safe extraction of milk from coconut. Automation is one of the vital developments which must be factored in our daily lives. The involvement of automation in the cutting, scraping, and grinding of coconut is to improve the quality of these processes with less or no human effort to save time, and reduce the possibility of danger while cutting. There are some existing products related to scrapping coconut in the market. In these products there are some limitations. Such as unable to cut, scrape and grind the coconut in a single system. As we concern about these limitations, we have come up with this machine as a solution.

Section 2: Literature Survey

Traditional and Electric coconut graters

Scraping mechanism

Figure 1

This figure shows the manual coconut scraper which is used in nowadays. Manually – operated coconut scraper machines are portable and may be used effectively in households, using the clamping screw to clamp the entire mechanism securely on a table. As one rotates the manual handle, the rotation is transferred to the scraping bit The DE husked coconut half-shell is pressed against the sharp bit while in rotation. This device requires a fair amount of effort to grate a coconut. Attention is required by the operator because if a slip occurs, serious injuries may result.

Figure 2

This figure shows the semi-automated coconut scraper. This machine is powered by current, and a switch is used to on and off the machine. The motor is used to rotate the scraper. When using this machine operator should hold the coconut and the machine. So, in this machine also attention is required by the operator because if a slip occurs, serious injuries may result.

Figure 3

In compared to above two machines the scraping part of our machine is very advantageous. Here we use two scrapers. They work simultaneously. So, time needed to scrap two coconuts is less than previous two machines. In those two machines we should hold coconut and the machine, but in our scraper part no need to hold coconut and the machine. Coconuts are held by the handles. This whole process is automated. Therefore, operator's attention is no needed. So, the user's safety is confirmed.

Cutting mechanism

Figure 4

This figure shows the way we usually use to break the coconut. This is an unsafe and hard mechanism. Also, here coconut water is wasted, and dust is spread everywhere.

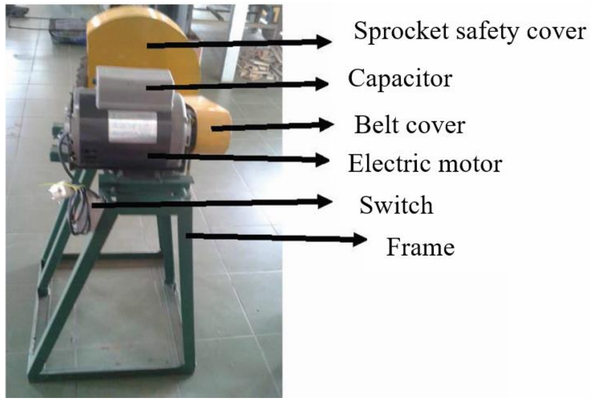


Figure 5

This coconut shell breaker is in the form of a box consisting of a breaking gear, a coconut clamp that can be adjusted according to the size of the coconut. The working principle of this tool is the incorporation of the clamping, pressing, and shifting processes according to the coconut fiber cutting path. The drive of the coconut shell breaker blade is driven by an electric motor which rotation has been reduced to produce a large torque. Coconut shells are the hardest part of a coconut fruit. Therefore, this shell breaker has several parts, including sprocket safety cover, capacitor, belt cover, electric motor as drive, switch on/off, coconut shell breaker sprocket, coconut shell breaker and breaker, gear box and screw adjuster.

Figure 6

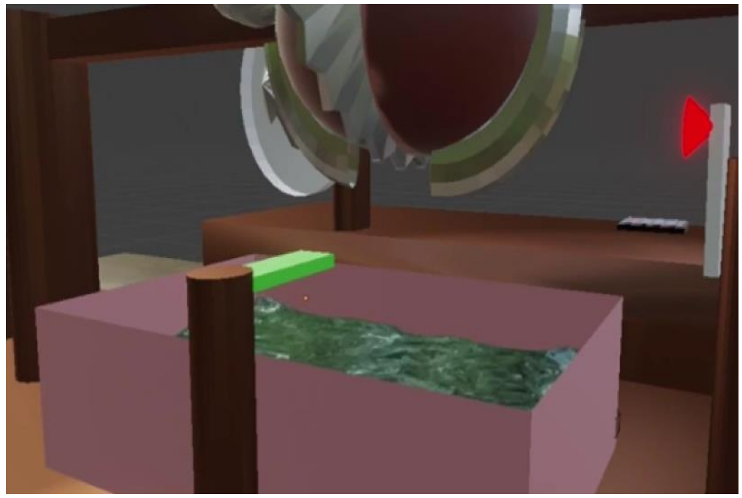


Figure 7

When compared to figure 4 the coconut breaking part of our machine is safer and easier. It is fully automated process. When compared to figure 5 the coconut breaking part of our machine takes less space. Also, our breaking part is less complex. Start and end process in figure 5 is manual but in our machine, we automated ending process using IR sensor. When considering both figure 4 and 5 in our machine there is a systematic way to collect coconut water and there is a cover to avoid coconut dust.

Summary

In our “automatic coconut breaking, scraping, and grinding machine” we improved coconut breaking and scraping processes both by safety side and easiness. Blender system is also connected to the machine. Then the whole process can be done through same machine, and it will be very easy in our day today life. The innovative automatic coconut breaking, scraping, and grinding machine is really time saving, less power consuming, no need of manpower and safe. It can break, scrape and grind coconuts within minutes. Also, it is portable and simple. The coconut will be broken, scraped and grinded automatically. The materials used for making this machine are safe and cheap. So, among all these machines we can say our one is the most effective machine.

Section 3: Problem in brief

- Hard & risky to break the coconut using knife with our hand.
- Difficult with scraping the coconut for a long time holding with the hands.
- Taking time to cut, scrape and grind the coconut separately.
- Have risk with handling the coconut through an unsafe machine.

Section 4: Aim and objectives

Aim

- Design and develop an automated machine for process of coconut breaking, scraping & grinding with safety mechanism.

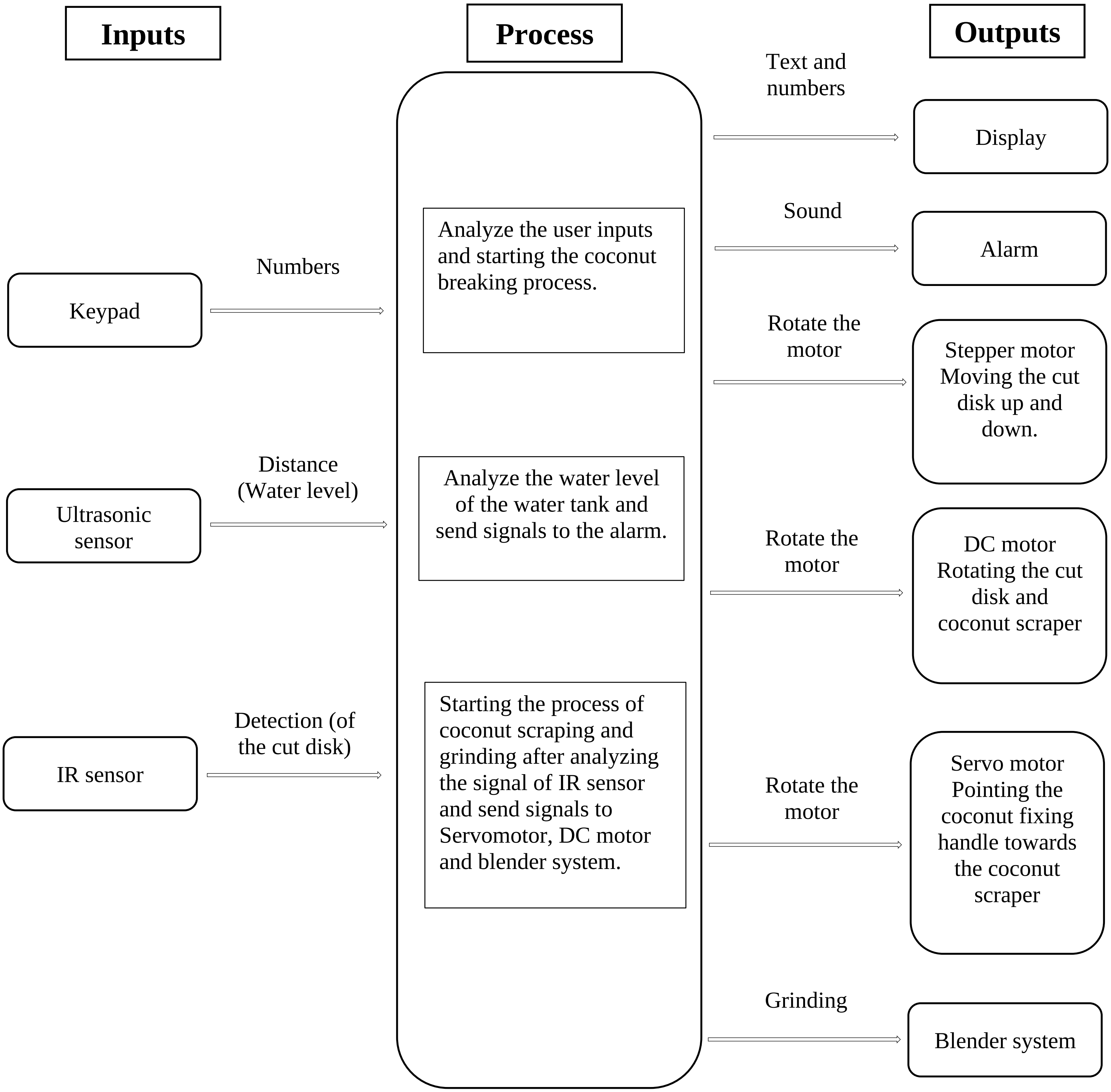
Objectives

- To break the coconut and get the water separately.
- To scrape the coconut.
- To grind the coconut.
- To maintain the process safely.

Section 5: System Description

We are using keypad, IR sensor, Ultrasonic sensor as the inputs. We are using keypad to input how many parts user need to scrape using the machine, to input the need of grinding, to input the time of grinding and resetting the details. We are using ultrasonic sensor to sense the water level of the coconut water tank. Moreover, we are using the IR sensor to sense the cut disk and make sure the coconut is broken properly. To get this done, we are using Atmega32 microcontroller. When we consider about the outcomes, there are 2*16 digital display, alarm, servo motors, DC motors, stepper motor and a grinder. We are using digital display to display the input data, alarm system to give an alarm at finishing time. When we consider about the motors, DC motors will do the coconut breaking alongside with a cut disk, scraping with a coconut scraper and also grinding. And stepper motor will do the movement of the cut disk. And servo motor will make the process of rotation of coconut fixing handle. Finally, there is a grinder to grind the scraped coconut automatically.

Block diagram



Atmega 32 Microcontroller

3D images

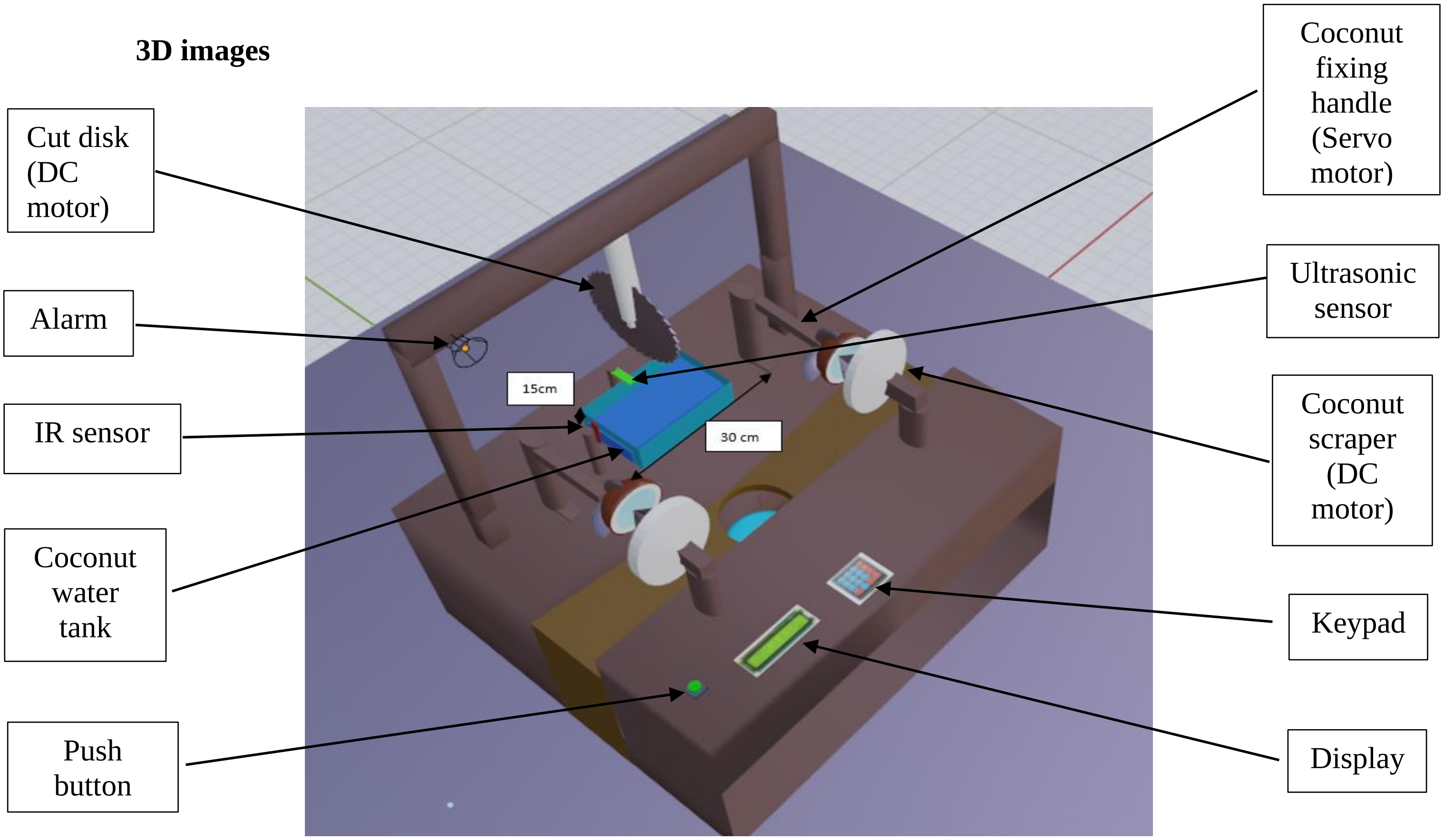


Figure 8 - Overview

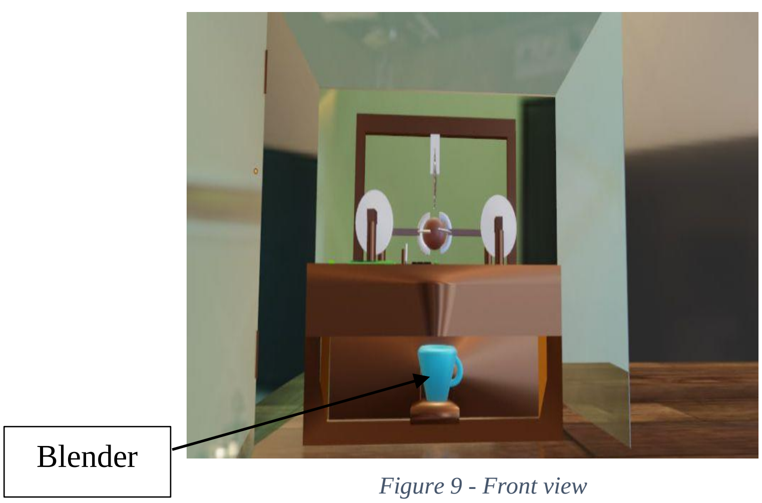


Figure 9 - Front view

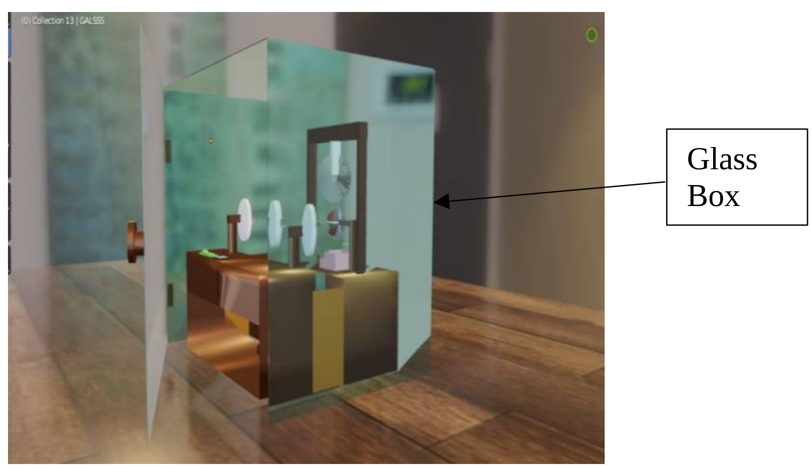


Figure 10 - Right side view

Section 6: Testing and Implementation

Member 1 - Nusky M.N.N. (204146L)

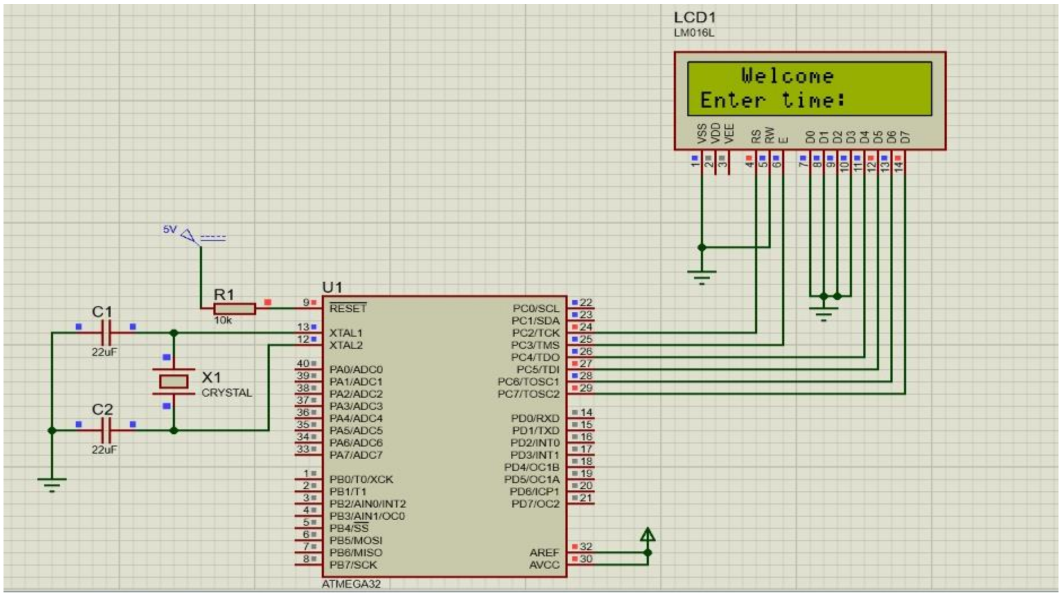


Figure 11-LCD circuit diagram

```c
#define F_CPU 8000000UL
#include<avr/io.h>
#include<util/delay.h>
#include "IncFile1.h"

#define rs PC2
#define en PC3

void lcd_init();
void dis_cmd(char);
void dis_data(char);
void lcdcmd(char);
void lcddata(char);

int main(void)
{
    unsigned char data0[]="Welcome";
    unsigned char data1[]="Enter time:";

    int i=0,x;
    DDRC=0xFF;
    lcd_init();

    while(data0[i]!='\0')
    {
        dis_data(data0[i]);
        _delay_ms(10);
        i++;
    }

    dis_cmd(0xC0);

    i=0;
    while(data1[i]!='\0')
    {
        dis_data(data1[i]);
        _delay_ms(10);
        i++;
    }
}
```

```c
void lcd_init()
{
    dis_cmd(0x02);
    dis_cmd(0x28);
    //dis_cmd(0x01);
    dis_cmd(0x0C);
    dis_cmd(0x06);
    dis_cmd(0x83);
}

void dis_cmd(char cmd_value)
{
    char cmd_value1;

    cmd_value1 = cmd_value & 0xF0;
    lcdcmd(cmd_value1);

    cmd_value1 = ((cmd_value<<4) & 0xF0)
    lcdcmd(cmd_value1);
}

void dis_data(char data_value)
{
    char data_value1;

    data_value1=data_value&0xF0;
    lcddata(data_value1);

    data_value1=((data_value<<4)&0xF0);
    lcddata(data_value1);
}

void lcdcmd(char cmdout)
{
    PORTC=cmdout;
    PORTC&=~(1<<rs);
    //PORTC&=~(1<<rw);
    PORTC|=(1<<en);
    _delay_ms(1);
    PORTC&=~(1<<en);
}

void lcddata(char dataout)
{
    PORTC=dataout;
    PORTC|=(1<<rs);
    //PORTC&=~(1<<rw);
    PORTC|=(1<<en);
    _delay_ms(1);
    PORTC&=~(1<<en);
}
```

Figure 12 - LCD C code

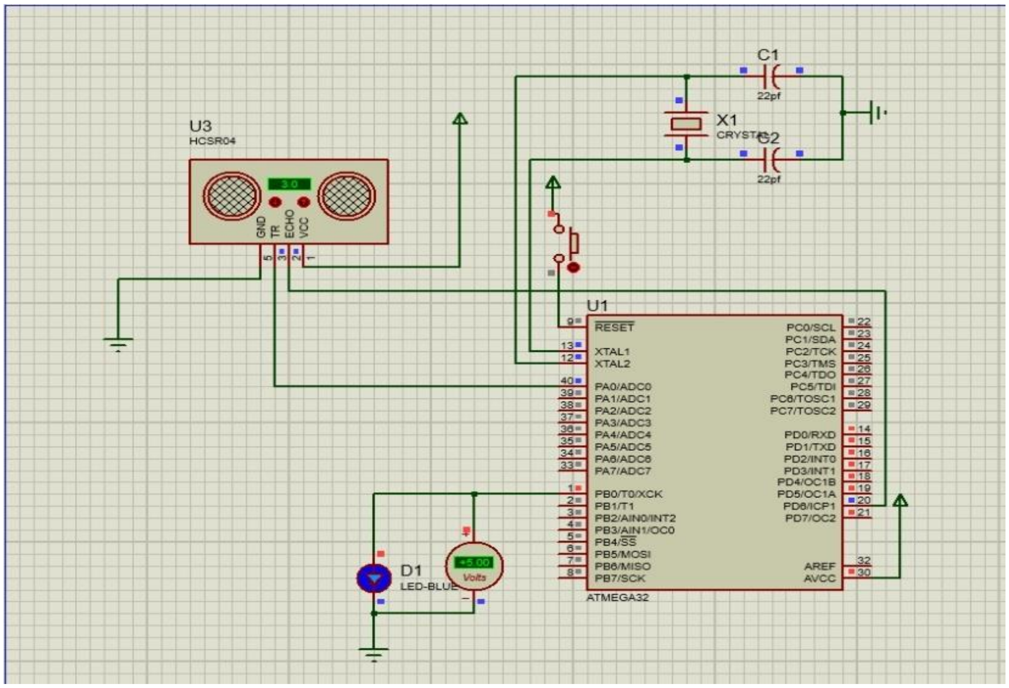


Figure 13 - Ultrasonic sensor circuit diagram

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

#define  Trigger_pin    PA0
#define Echo_pin     PD6
int TimerOverflow = 0;

ISR(TIMER1_OVF_vect)
{
    TimerOverflow++;
}

int main(void)
{
    long count;
    double distance;

DDRA = 0x01;
    PORTD = 0xFF;
    sei();
    TIMSK = (1 << TOIE1);
    TCCR1A = 0;
```

```c
    while(1)
    {

        PORTA |= (1 << Trigger_pin);
        _delay_us(10);
        PORTA &= (~(1 << Trigger_pin));

        TCNT1 = 0;
        TCCR1B = 0x41;
        TIFR = 1<<ICF1;
        TIFR = 1<<TOV1;

        while ((TIFR & (1 << ICF1)) == 0);
        TCNT1 = 0;
        TCCR1B = 0x01;
        TIFR = 1<<ICF1;
        TIFR = 1<<TOV1;
        TimerOverflow = 0;

        while ((TIFR & (1 << ICF1)) == 0);
        count = ICR1 + (65535 * TimerOverflow);
        distance = (double)count / 466.47;

        if(distance<5){
            PORTB=0xFF;
        }
        else(distance>5)
        {
            PORTB=0x00;
        }
    }
}
```

Figure 14 - Ultrasonic sensor C code

Member 2 - Banu A.G.S. (204018X)

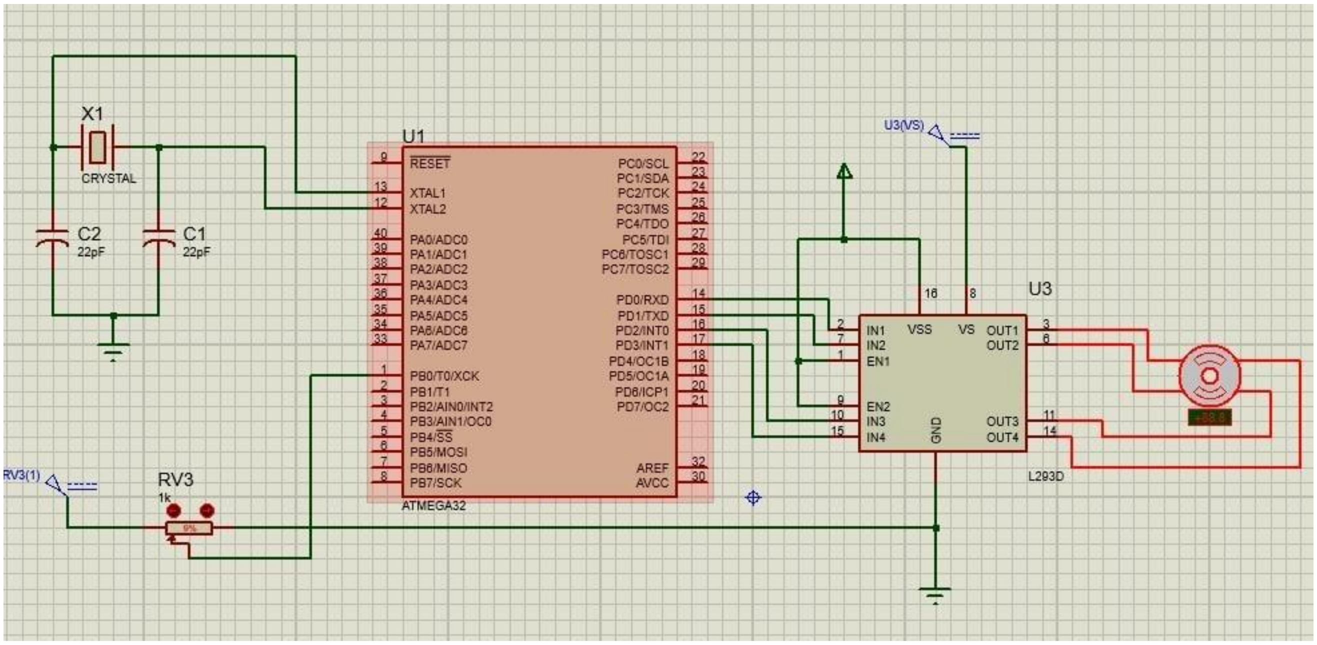


Figure 15 - Stepper motor circuit diagram

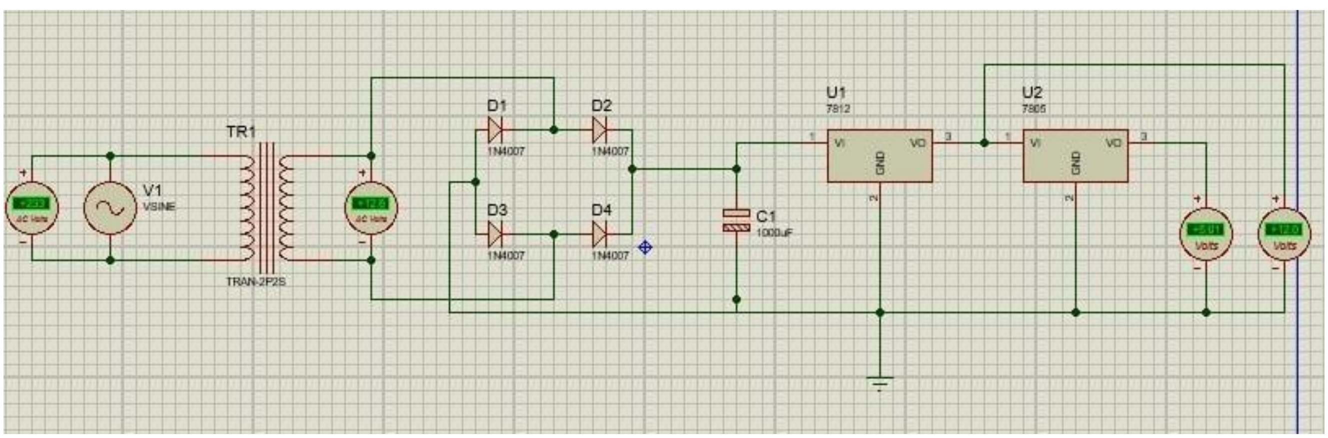


Figure 16 - Power supply

```c
#ifndef F_CPU
#define F_CPU 8000000UL // 8 MHz clock speed
#endif

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

int main(void)
{
    DDRC = 0x01; //Makes RC0 output pin
    PORTC = 0xFF;
    while(1)
    {
        //Rotate Motor to 0 degree
        PORTC = 0x01;
        _delay_us(1000);
        PORTC = 0x00;

        _delay_ms(2000);

        //Rotate Motor to 90 degree
        PORTC = 0x01;
        _delay_us(1500);
        PORTC = 0x00;

        _delay_ms(2000);

        //Rotate Motor to 180 degree
        PORTC = 0x01;
        _delay_us(2000);
        PORTC = 0x00;

        _delay_ms(2000);
    }
}
```

Figure 17 - Stepper motor C code

Member 3 – Karunaweera R.L. (204096G)

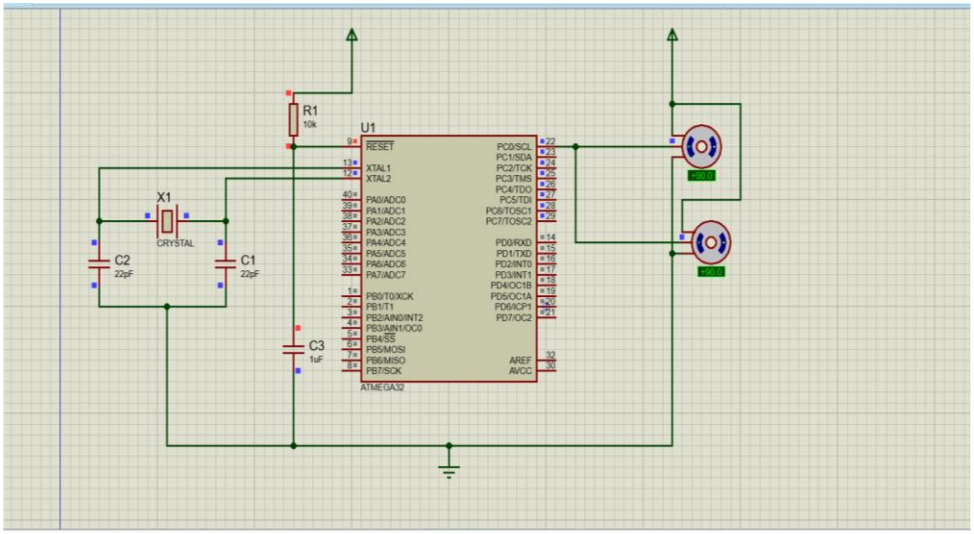


Figure 18 - Servo motor circuit diagram

```c
#ifndef F_CPU
#define F_CPU 8000000UL // 16 MHz clock sp
#endif

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

int main(void)
{
    DDRC = 0x01; //Makes PC0 output pin
    PORTC = 0x00;
    DDRD = 0x01; //Makes PD0 output pin
    PORTD = 0x00;

    while(1)
    {
        //the point coconut is fixed to the handle
        //rotate towards the scraper
        PORTC = 0x01;
        PORTD = 0x01;
        _delay_us(1000);
```

```c
        PORTC = 0x00;
        PORTD = 0x00;

        _delay_ms(5000);

        //returning to the starting point
        PORTC = 0x01;
        PORTD = 0x01;
        _delay_us(1500);
        PORTC = 0x00;
        PORTD = 0x00;

        _delay_ms(5000);
    }
}
```

Figure 19 - Servo motor C code

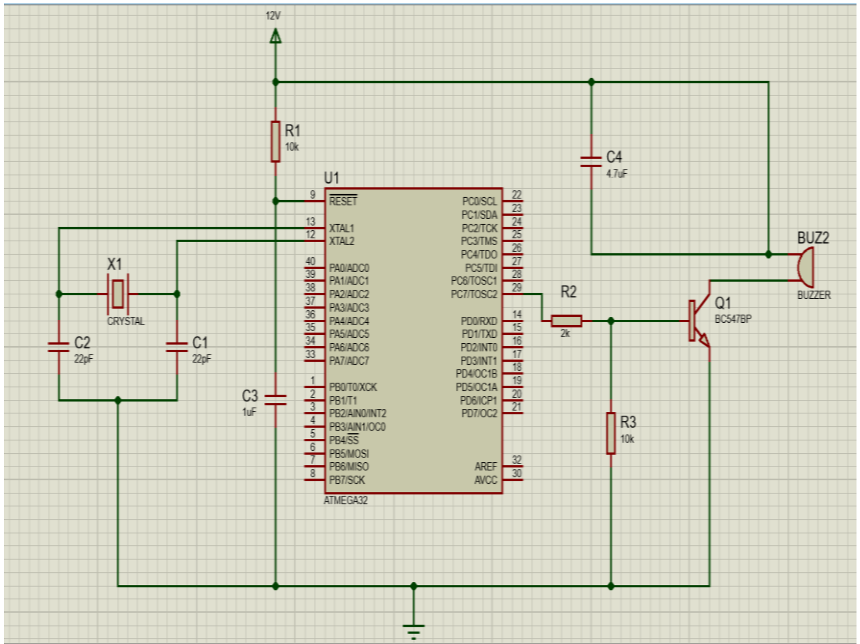


Figure 20 - Buzzer circuit diagram

```c
#include <avr/io.h>

int main()
{
    DDRC = 0xff; //Makes PORTC as output
    DDRA = 0x00; //Makes PORTA as Input to get signal from Ultrasonic sensor
    PORTA = 0xff; // Enable The PullUps of PORTA.
    while(1)
    {
        if(PINA==0x01) // Read the signal from Ultrasonic sensor and Turn ON/OFF the Buzzer
        {
            PORTC=0x01;
        }
        else
        {
            PORTC=0x00;
        }
    return 0;
}
```

Figure 21 - Buzzer C code

Member 4 - Kumarasingha H.J.A. (204108A)

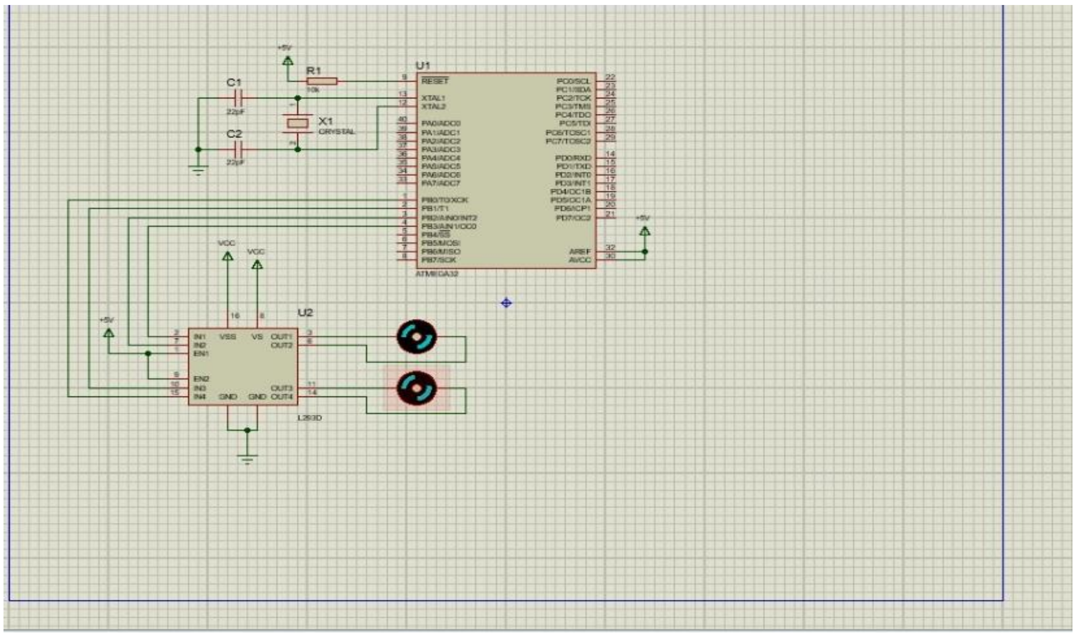


Figure 22 - DC motor circuit diagram

```c
#ifndef F_CPU
#define F_CPU 16000000UL // 16MHz clock speed
#endif
#include <avr/io.h> //standard AVR library
#include <util/delay.h> // delay library


int main(void)
{
    DDRB= 0XFF; //direction of port B as output
    while(1) // infinite loop
    {
        _delay_ms(500); // until the cut disk comes down and the coconut breaks
      PORTB = 0X05; //motor rotation in clockwise direction
      _delay_ms(2000); //delay of 2  sec

      PORTB = 0X00; // motor stopped
      _delay_ms(2000); // delay of 2  sec

    }
}
```

Figure 23 - DC motor C code

Member 5 - Lamahewage D. R. (204114M)

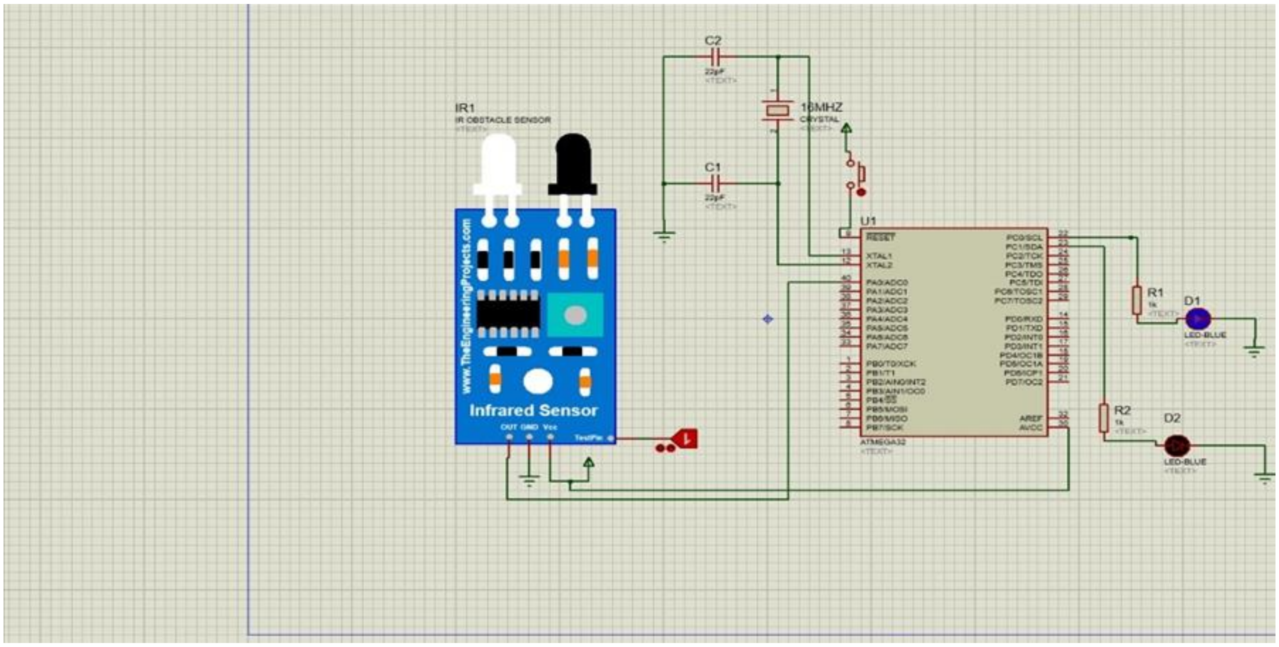


Figure 24 - IR sensor circuit diagram

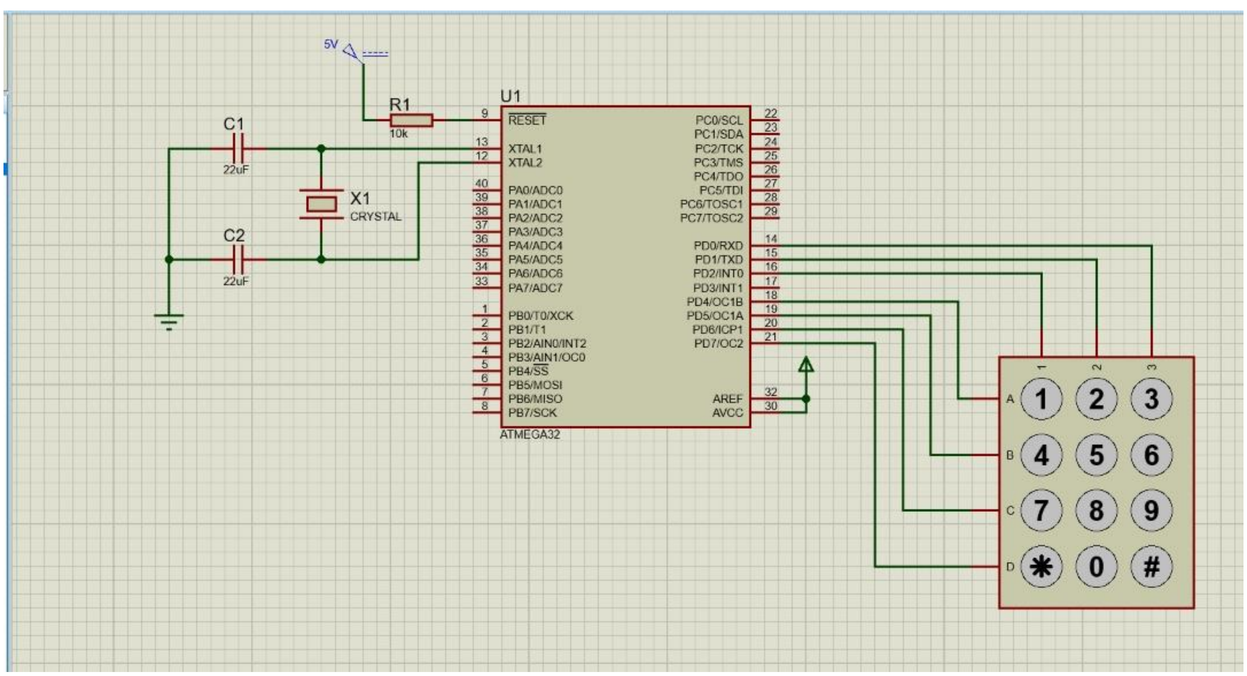


Figure 25 - Keypad circuit diagram

```c
#define F_CPU 16000000UL
#include <avr/io.h>
int main(void)
{
    DDRA=0x00;
    DDRC=0xff;
    while (1)
    {
        if(PINA==0x01)
        {
            PORTC=0x01;
        }
        else
        {
            PORTC=0x00;
        }

    }
}
```

Figure 26 - IR sensor C code

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

void cmd(char c);
void data(char d);
void lcd_init();

int main(void)
{
    lcd_init();
    DDRD = 0x0F;
    PORTD = 0xF0;

    while(1){

        PORTD&=~(1<<PD2);
        PORTD|=(1<<PD0)|(1<<PD1);

        if (!(PIND&(1<<PD4))){
            data('1');
            _delay_ms(30);
        }
        else if (!(PIND&(1<<PD5))){
            data('4');
            _delay_ms(30);
        }
        else if (!(PIND&(1<<PD6))){
            data('7');
            _delay_ms(30);
        }
        else if (!(PIND&(1<<PD7))){
            data('*');
            _delay_ms(30);
        }

        PORTD&=~(1<<PD1);
        PORTD|=(1<<PD2)|(1<<PD0);

        if (!(PIND&(1<<PD4))){
            data('2');
            _delay_ms(30);
        }
        else if (!(PIND&(1<<PD5))){
            data('5');
            _delay_ms(30);
        }
        else if (!(PIND&(1<<PD6))){
            data('8');
            _delay_ms(30);
        }
        else if (!(PIND&(1<<PD7))){
            data('0');
            _delay_ms(30);
        }

        PORTD&=~(1<<PD0);
        PORTD|=(1<<PD2)|(1<<PD1);

        if (!(PIND&(1<<PD4))){
            data('3');
            _delay_ms(30);
        }
        else if (!(PIND&(1<<PD5))){
            data('6');
            _delay_ms(30);
        }
        else if (!(PIND&(1<<PD6))){
            data('9');
            _delay_ms(30);
        }
        else if (!(PIND&(1<<PD7))){
            cmd(0x01);
            _delay_ms(30);
        }
    }
}
```

Figure 27 - Keypad C code

Section 7: Resource Requirements

Software:

- Atmel Studio
- Blender
- Simulation software

Hardware:

- Microcontroller: Atmega32
- IR Sensor
- Ultrasonic sensor
- Keypad
- Alarm
- Display
- Stepper motor
- Servo motor
- DC motor
- Push button
- Blender system

Section 8: Cost estimation

Item	Quantity	Price (Rs)	Subtotal (Rs)
Atmega32	1	540	540
IR sensor	1	25	25
Ultrasonic sensor	1	200	200
Alarm	1	300	300
Keypad (4*4)	1	160	160
Display (2*16)	1	600	600
Push button	1	10	10
Servo motor	2	500	1000
Stepper motor	2	1050	2100
DC motor	3	1250	3750
Blender system	1	3500	3500
Finishing	1	4000	4000
Total			16185

Table 1 - Estimated cost

Section 9: References

[1] ADMIN, "atmega32avr," atmega32avr, 2015. [Online]. Available: https://atmega32-avr.com/speed-direction-control-stepper-motor-using-avr-microcontroller/amp/. [Accessed 15 September 2021].

[2] B. Ayob, "Atmega32-avr," [Online]. Available: https://atmega32-avr.com/servo-motor-control-using-avr-atmega32-microcontroller/.

[3] B. Ayob, "atmega32-avr," atmega32-avr, 25 April 2012. [Online]. Available: https://atmega32-avr.com/atmega-32-pinout/. [Accessed 29 September 2021].

[4] B. Earl, "ADAFRUIT," ADAFRUIT, 5 May 2014. [Online]. Available: https://learn.adafruit.com/all-about-stepper-motors. [Accessed 15 October 2021].

[5] E. Funda, "Engineering Funda," Engineering Funda, 24 May 2021. [Online]. Available: https://www.youtube.com/watch?v=awS93t-uX4Q. [Accessed 29 October 2021].

[6] Github, "Embedotronics," Embedotronics, 2021. [Online]. Available: https://github.com/Embedotronics/Atmega16-interfacing-with-LCD-in-4-bit-mode. [Accessed 26 October 2021].

[7] l. chandak, "ElectronicWings Homepage," ElectronicWings , 2018. [Online]. Available: https://www.electronicwings.com/avr-atmega/ultrasonic-module-hc-sr04-interfacing-with-atmega1632. [Accessed 23 October 2021].

[8] l. chandak, "ElectronicWings Homepage," ElectronicWings Homepage, 2018. [Online]. Available: https://www.electronicwings.com/avr-atmega/interfacing-lcd-16x2-in-4-bit-mode-with-atmega-16-32-. [Accessed 16 September 2021].

[9] TUSHAR, "Embedds," Embedds, 2018. [Online]. Available: https://embedds.com/interfacing-dc-motor-to-atmega32/. [Accessed 25 September 2021].

[10] Unknown, "Wikipedia," Wikipedia, 21 August 2021. [Online]. Available: https://en.wikipedia.org/wiki/DC_motor. [Accessed 15 October 2021].

Section 10: Individuals Contribution to the Project

Name of student: Nusky M.N.N. (204146L)

1. Study basic microcontroller programming
2. Studying LCD and Ultrasonic sensor module
3. Programming LCD and Ultrasonic sensor module
4. Design simulation
5. Designing 3D image

Name of student: Banu A.G.S. (204018X)

1. Study basic microcontroller programming
2. Studying stepper motor module and power supply
3. Programming stepper motor module
4. Design simulation
5. Designing 3D image

Name of student: Karunaweera R.L. (204096G)

1. Study basic microcontroller programming
2. Studying servo motor module and buzzer module
3. Programming servo motor module and buzzer module
4. Design simulation
5. Designing 3D image

Name of student: Kumarasingha H.J. A. (204108A)

1. Study basic microcontroller programming
2. Studying DC motor module
3. Programming DC motor module
4. Design simulation
5. Designing 3D image

Name of student: Lamahewage D. R. (204114M)

1. Study basic microcontroller programming
2. Studying IR sensor and Keypad modules
3. Programming IR sensor and Keypad modules
4. Design simulation
5. Designing 3D image

Section 11: Action Plan (Appendix C)

	Responsible group member	Task	Duration	Start Date	End Date	Completion date					
						24/01/2022	10/02/2021	10/02/2022	24/02/2022	10/03/2022	24/03/2022
1	204146L	Studying more about components and design circuit.	4 weeks	10/01/2022	10/02/2021						
		Design PCB and compile the code	2 weeks	10/02/2022	24/02/2022						
		Programming and testing	4 weeks	24/02/2022	24/03/2022						
2	204018X	Studying more about components and design circuit.	4 weeks	10/01/2022	10/02/2021						
		Design PCB and compile the code	2 weeks	10/02/2022	24/02/2022						
		Programming and testing	4 weeks	24/02/2022	24/03/2022						
3	204096G	Studying more about components and design circuit.	4 weeks	10/01/2022	10/02/2021						
		Design PCB and compile the code	2 weeks	10/02/2022	24/02/2022						
		Programming and testing	4 weeks	24/02/2022	24/03/2022						
4	204108A	Studying more about components and design circuit.	4 weeks	10/01/2022	10/02/2021						
		Design PCB and compile the code	2 weeks	10/02/2022	24/02/2022						
		Programming and testing	4 weeks	24/02/2022	24/03/2022						
5	204114M	Studying more about components and design circuit.	4 weeks	10/01/2022	10/02/2021						
		Design PCB and compile the code	2 weeks	10/02/2022	24/02/2022						
		Programming and testing	4 weeks	24/02/2022	24/03/2022						

Table 2 - Action plan