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
Dept. of Information Technology

Date of Submission: 01.10.2021

Co-supervisor Name: Upulanka Premasiri
Lecturer
Dept. of Information Technology

Table of contents

Table of figures

Introduction

This project aims at designing an Automatic Coconut breaking, scraping, and grinding machine to enable easy and safe extraction of milk from the coconut. Automation is one of the vital developments which must be factored into 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 being unable to cut, scrape and grind the coconut in a single system. As we are concerned about these limitations, we have come up with this machine as a solution.

Literature Survey

Traditional and Electric coconut graters

Scraping mechanism

Figure 1

This figure shows the manual coconut scraper which is used 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 a current, and a switch is used to turn on and off the machine. The motor is used to rotate the scraper. When using this machine, the 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

Compared to the above two machines, the scraping part of our machine is very advantageous. Here we use two scrapers. They work simultaneously. So, the time needed to scrap two coconuts is less than the previous two machines. In those two machines, we should hold the coconut and the machine, but in our scraper part, there is no need to hold the coconut and the machine. Coconuts are held by the handles. This whole process is automated. Therefore, the operator's attention is not 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.

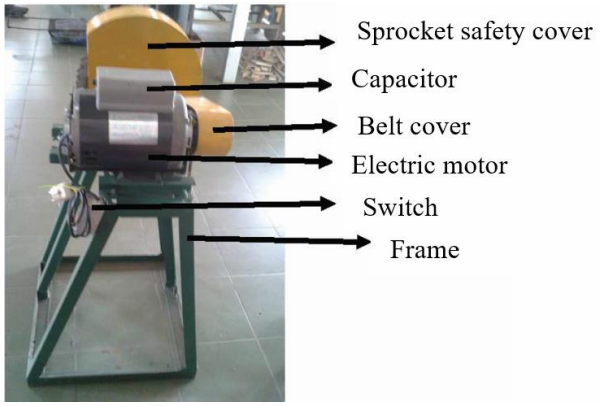


Figure 5

This coconut shell breaker is in the form of a box consisting of a breaking gear, and 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 coconut shell breaker blade is driven by an electric motor whose rotation has been reduced to produce a large torque. Coconut shells are the hardest part of coconut fruit. Therefore, this shell breaker has several parts, including a sprocket safety cover, capacitor, belt cover, electric motor as drive, switch on/off, coconut shell breaker sprocket, coconut shell breaker and breaker, gearbox, and screw adjuster.

Figure 6

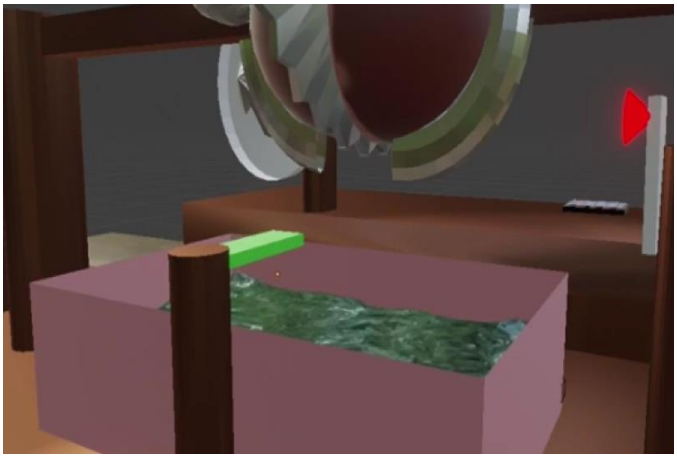


Figure 7

When compared to figure 4 the coconut breaking part of our machine is safer and easier. It is a 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. The start and end process in figure 5 is manual but in our machine, we automated the ending process using an IR sensor. When considering both figures 4 and 5 in our machine there is a systematic way to collect coconut water and there is a cover to avoid coconut dust.

Also in the scrapping process, we use a color sensor. It helps to prevent the scrapping of the coconut shell. That is also another advantage.

Summary

In our “automatic coconut breaking, scraping, and grinding machine” we improved coconut breaking and scraping processes both by safety side and easiness. The Blender system is also connected to the machine. Then the whole process can be done through the same machine, and it will be very easy in our day-to-day life. The innovative automatic coconut breaking, scraping, and grinding machine is time-saving, less power-consuming, has no need for manpower, and is 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.

Aim and objectives

Aim

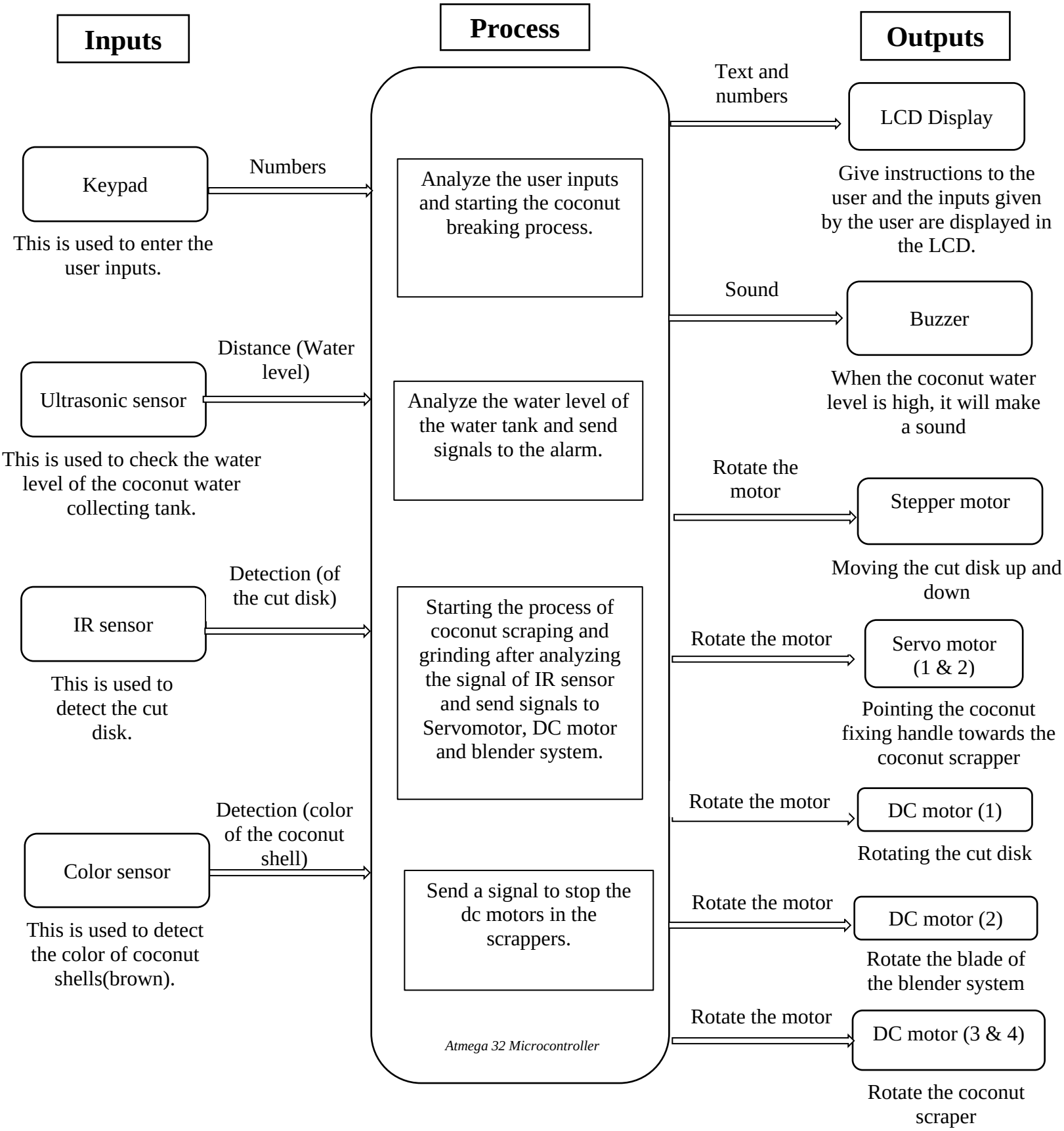
- Design and develop an automated machine for the process of coconut breaking, scraping & grinding with safety mechanisms.

Objectives

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

Analysis and Design

Block diagram



Schematic diagram

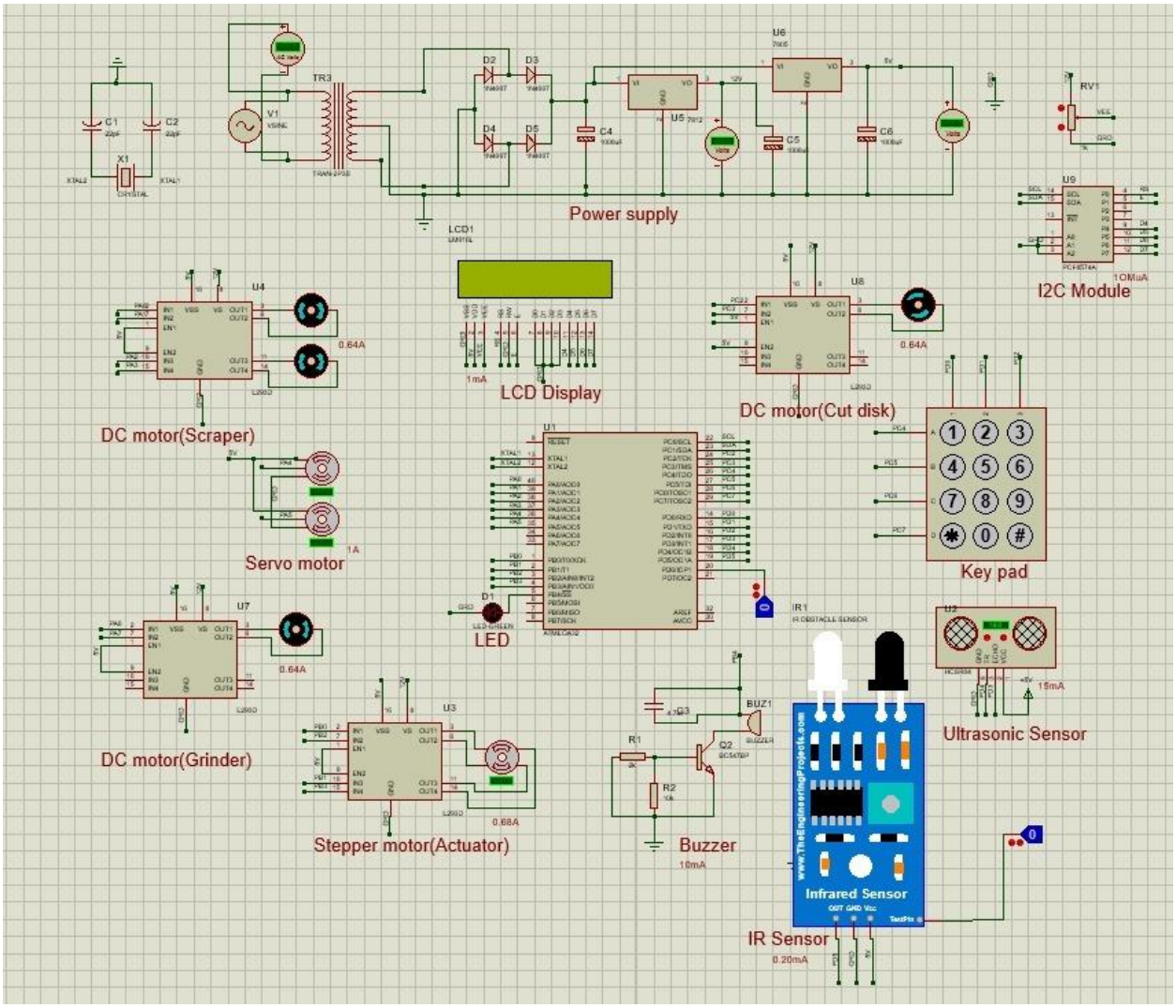


Figure 8 - Full schematic diagram

3D images

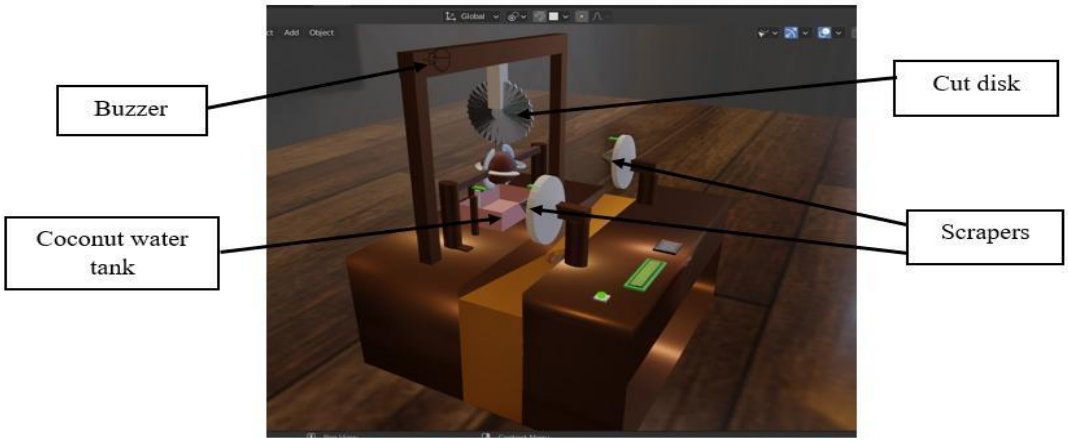


Figure 9 - 3D diagram left side view

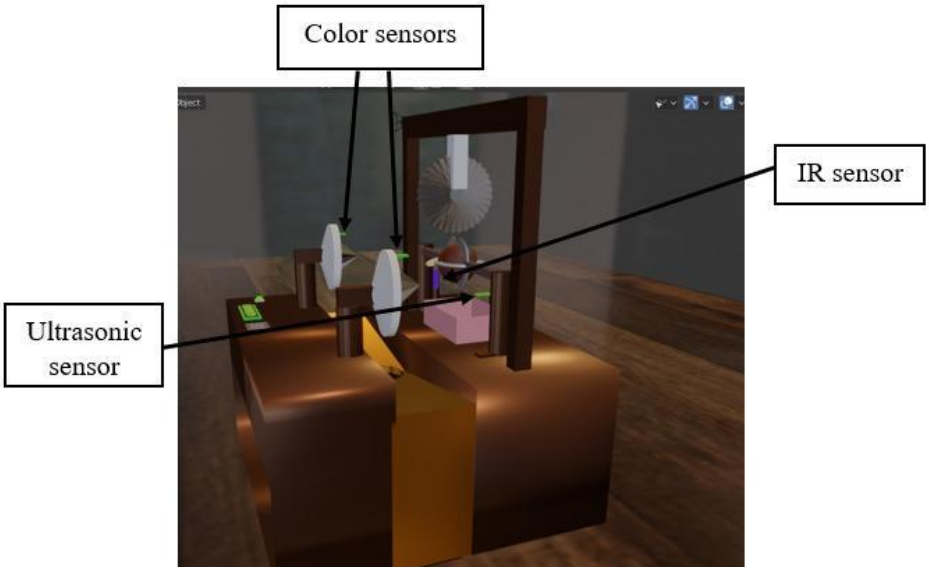


Figure 10 - 3D diagram right side view

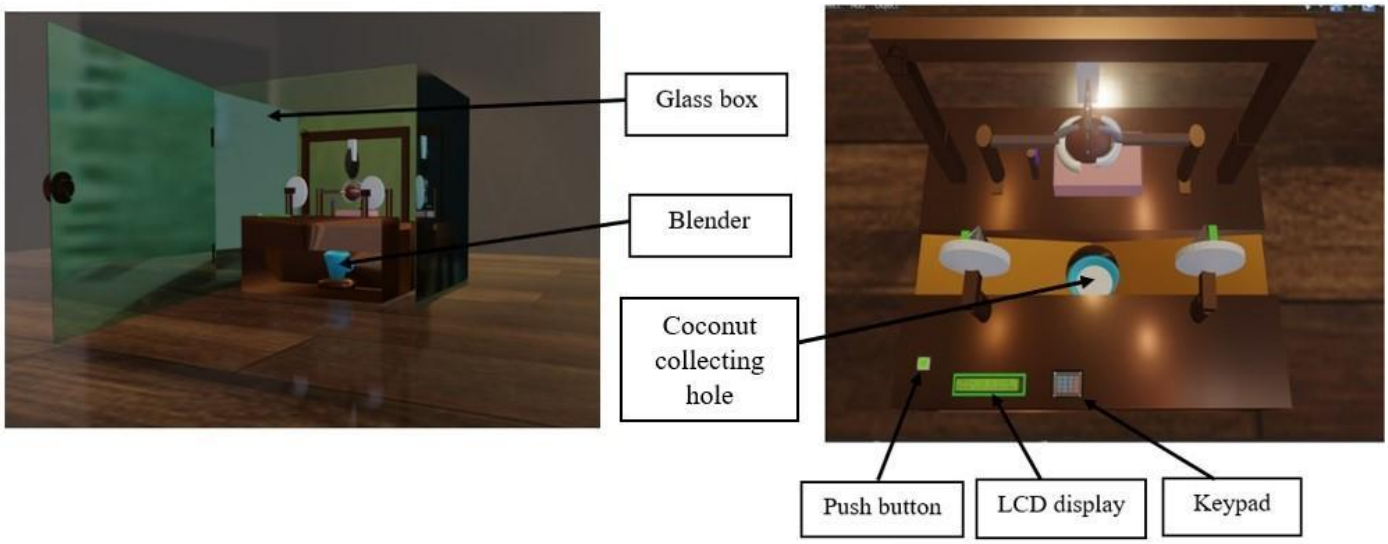


Figure 11 - 3D diagram front & top view

Testing and Implementation

First, we should connect the machine to the power supply to power up the machine.

Using the push button user can start the machine. Once the machine starts the LCD displays the below message.

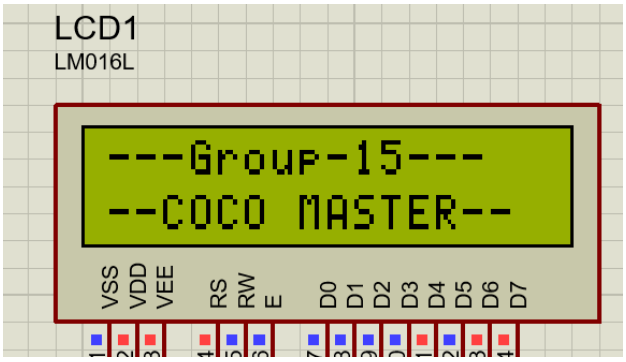


Figure 10 - LCD introductory message

Then the stepper motor and the DC motor begin to rotate clockwise simultaneously to break the coconut. After the breaking part process is done, the IR sensor detects the cut disk and sends signals to the stepper motor and the DC motor. Then only the stepper motor rotates anti-clockwise direction and the dc motor stops rotating.

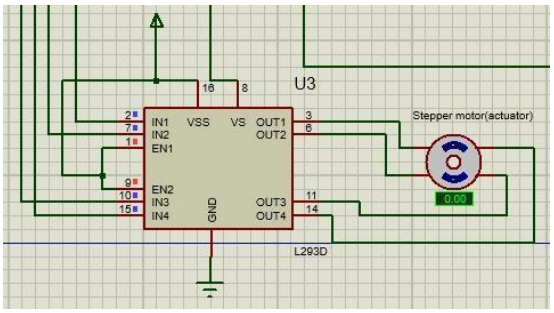


Figure 12 - Before rotating the stepper motor

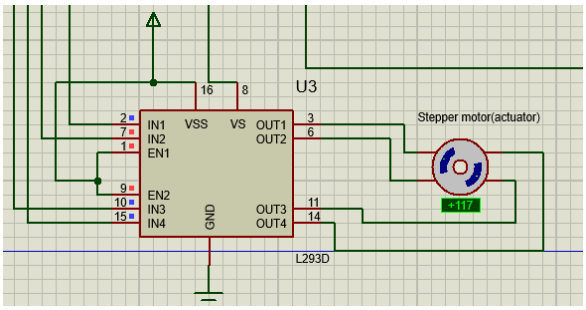


Figure 11 - After rotating the stepper motor

Then the LCD will ask for the need for scraping. If scrapping is needed, the user should input number 1 using the keypad. If not, the user should input number 2.

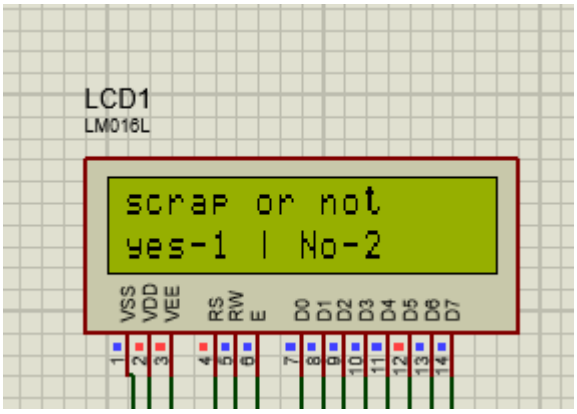


Figure 13 - LCD message 2

When the user inputs number 1, the display will ask whether one coconut part or both parts need to be scrapped.

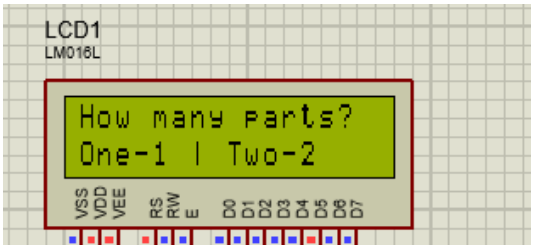


Figure 14 - LCD message 3

If the user gives number 1, only one servo motor connected to the handle will turn by 90°. After the servo motor rotates, one DC motor connected to the scraper will rotate.

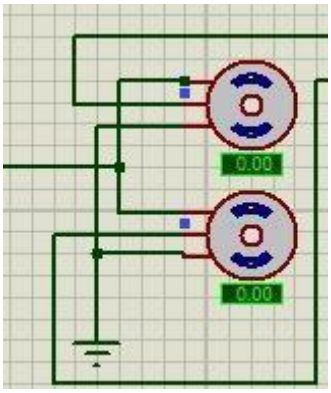


Figure 16 - Before rotating the servo motor

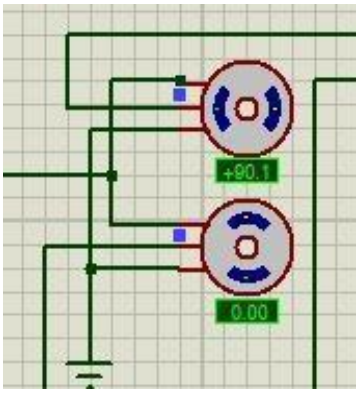


Figure 15 - After rotating only one servo motor

If the user gives number 2, both servo motors connected to the handles will turn by 90°. After the servo motors rotate, both DC motors connected to the scrapers will rotate.

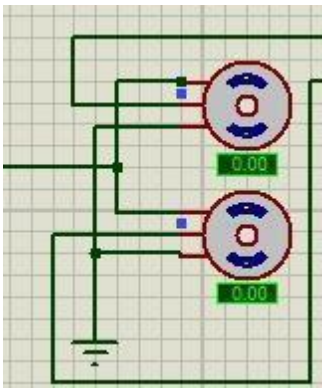


Figure 18 - Before rotating both servo motors

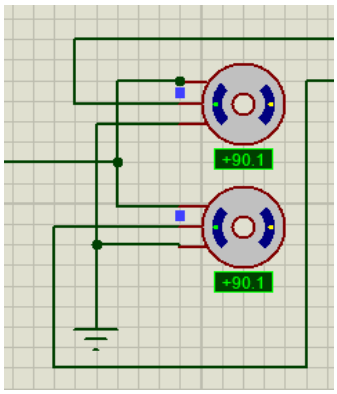


Figure 17 - After rotating both servo motors

While rotating the scrapers, the color sensor connected to the scraper will detect the coconut shell color(brown). If the brown color is detected DC motors connected to the scrappers will stop rotating. In our simulation we used a testing pin to show the process of the color sensor.

for the brown color we assigned binary 1. for the white color we assigned binary 0.

After the scrapping part is done, the LCD displays the below message.

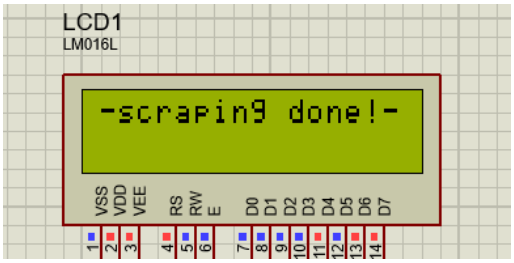


Figure 19 - LCD message 4

After the DC motor stops, the LCD displays the below message asking the need of grinding.

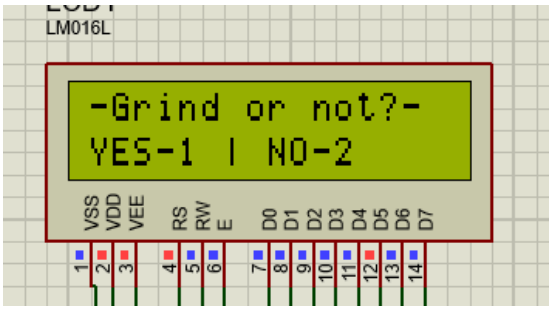


Figure 20 - LCD message 5

If the user inputs number 1, the LCD shows the below message. The user will be given time to add water to the grinder.

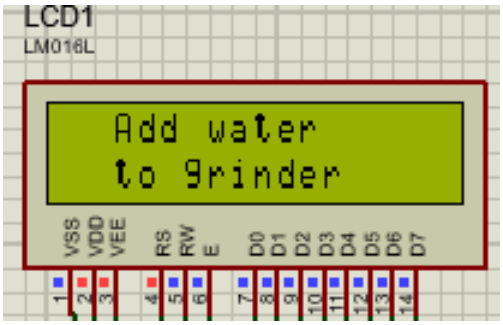


Figure 22 - LCD message 6

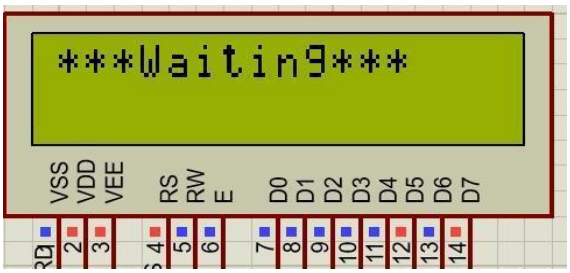


Figure 21 - LCD message 7

According to the number of scraped coconut parts, the rotating time of the DC motor in the blender system will automatically be set by the machine. After the given time, the DC motor rotating stops.

After the grinding process the ultrasonic sensor detects the coconut water level of the coconut water collecting tank. If the distance between the ultrasonic sensor and coconut water level is very low, the LED will blink, and the buzzer will make a sound. The LCD also displays the below message. Then the machine will stop.

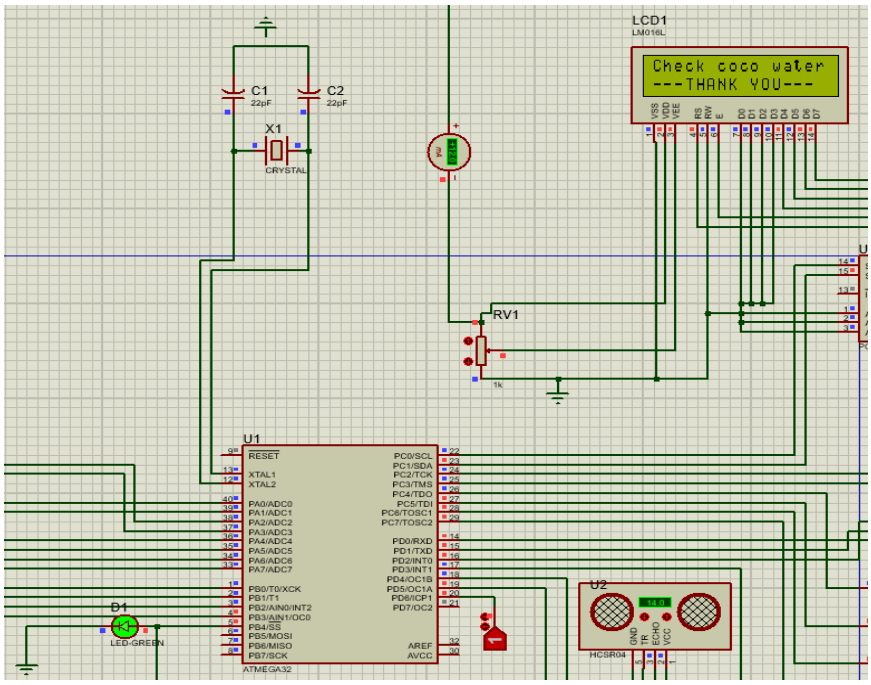


Figure 23 - End of the process (LCD message and LED)

Total cost estimation

Item	Quantity	Price (Rs)	Subtotal (Rs)
Atmega32	1	1100	1100
IR sensor	1	150	150
Ultrasonic sensor	1	350	350
Buzzer	1	400	400
Keypad (4*3)	1	260	260
Display (2*16)	1	840	840
Push-button	1	40	40
Servo motor	2	600	1200
Stepper motor	1	1000	1000
DC motor	4	1250	5000
Finishing	1	3000	3000
Color sensor	1	1500	1500
Motor driver	4	780	3120
I2C module	1	340	340
Total			18300

Further work

- In the scrapping part, color sensor detects only the coconut shell color and stops scrappers because sometimes coconut can be left inside the coconut shell after stopping the scrappers. So, coconut can be wasted. As a solution we propose another automated process for the scrappers using servo motors. If the color sensor detects the coconut shell, scrappers will change its angle and do the scraping process until the coconut runs out.

- Sometimes users will face difficulties using the keypad to enter inputs. So, we propose a remote-control system as further work to enter inputs more easily.

- Because of the plastic cover dust will collect inside the machine. As a solution, we propose an automated cleaning process to remove the dust which collects inside the machine.

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].

Individual contribution

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

Responsible Part:

My responsible components are the Ultrasonic sensor module and LCD Display module. I have done the designing of schematic diagrams, PCB designs, and C codes for the above two modules. And I made the full code of our machine. Also, I contributed to making the reports, 3D design of our machine and presentations as well.

1. **Ultrasonic sensor**

Technique:

In our project, we used ultrasonic sensor to identify whether coconut water tank overflow or not. We fixed ultrasonic sensors in the 10 cm height above the tank. When a distance (height) below 15 cm, an ultrasonic sensor is given a signal to the buzzer, and it will be given to signal to the display.

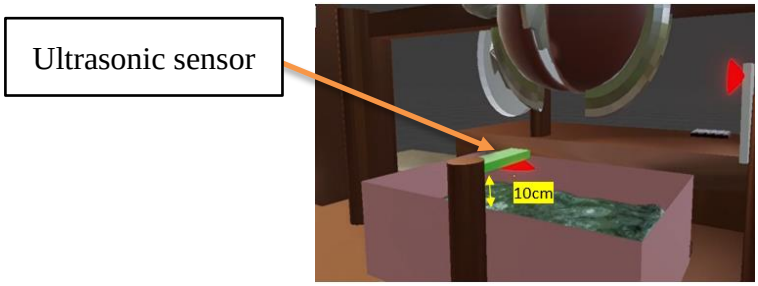


Figure 24 - Ultrasonic sensor

Specifications:

- Working voltage: 5V DC
- Working current: 15mA
- Distance range: 2 - 400cm
- Measuring angle: 15 degrees
- Trigger input pulse: 10 micro-seconds

Schematic Diagram:

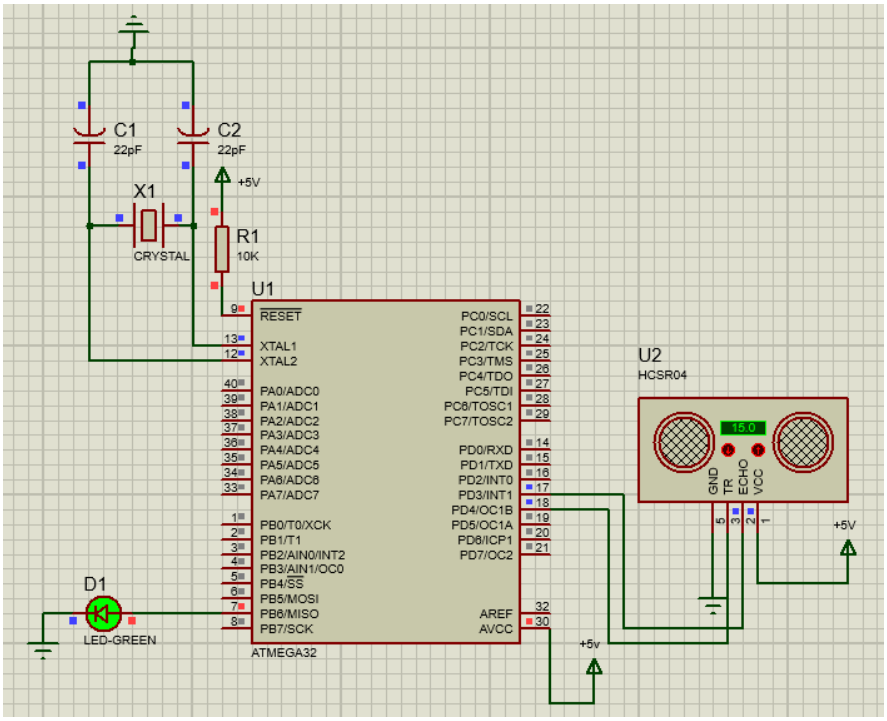


Figure 25 - Ultrasonic sensor schematic diagram

PCB design:

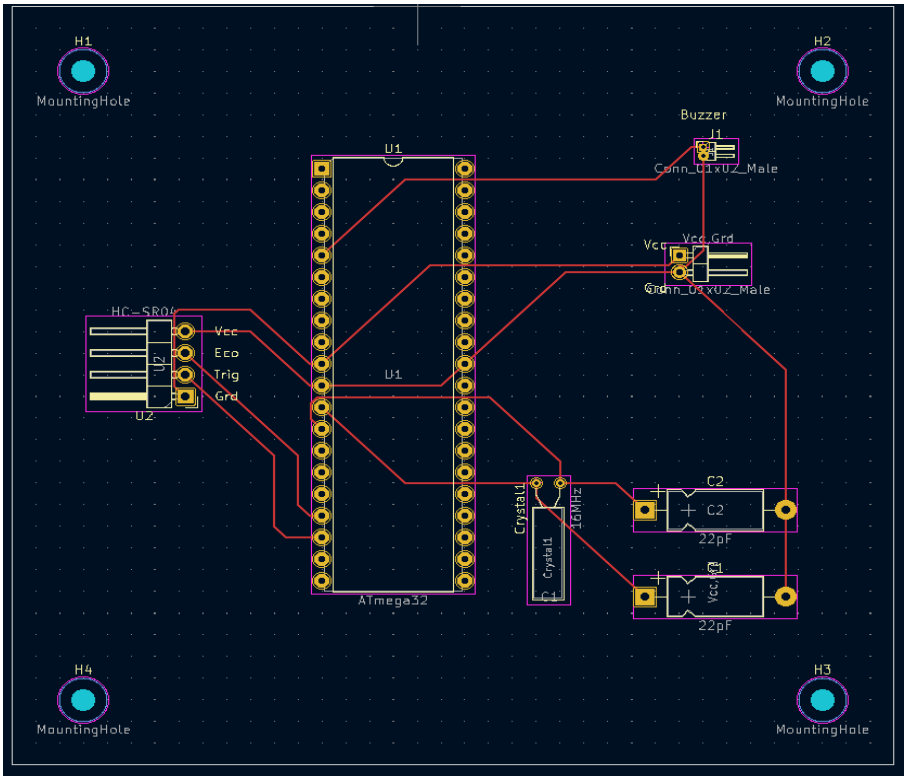


Figure 26 - Ultrasonic sensor PCB diagram

Code:

```
#define F_CPU 8000000UL
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdlib.h>
#include <avr/io.h>
static volatile unsigned int tCount = 0;
static volatile int i = 0;
double distance = 0;
int main(void)
{

	UBRRH=0x00;
	UBRRL=51;
	DDRB = DDRB | 1 << DDB6;
	DDRD = DDRD | 1 << DDD4;
	DDRD = DDRD & ~(1 << DDD3) ;
	PORTD = PORTD & ~(1 << PD3);
	_delay_ms(50);
	GICR |= 1<<INT1;
	MCUCR |= 1<<ISC10;
	TIMSK |= 1 << TOIE1;
	sei();
	while(1)
	{
		PORTD = PORTD | 1<<PD4;
		_delay_us(15);
		PORTD = PORTD &
~(1<<PD4);
		_delay_us(15);
	distance =
tCount*1000000.0/F_CPU/58;
		if (distance < 15)
		{
			PORTB = PORTB | 1 <<
PB6;
		}
else
	{
	PORTB = PORTB & ~(1 <<
PB6);
	}
	_delay_ms(200);
	}
}
ISR(INT1_vect)
{
	if(i == 0)
	{
		TCCR1B |= 1<<CS10;
		i = 1;
	}
	else
	{
		TCCR1B = 0;
		tCount = TCNT1;
		TCNT1 = 0;
		i = 0;
	}
}

ISR(TIMER1_OVF_vect){
	TCCR1B = 0;
	tCount = 200;
	TCNT1 = 0;
	i = 0;
}
```

2. LCD display

Technique:

In our project we use a LCD display (16*2) for displaying what the process is at the time on our machine and get the input from the user at a specific time.

E.g. 1. Do you want to scrape?
2.How many parts scrape?
3.Do you want to grind?

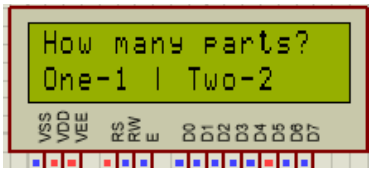


Figure 27

Specification:

- Operating voltage of the LCD is 4.7V – 5.3V
- The utilization of current is 1mA with no backlight
- The display bezel is 72 x 25mm
- Number of columns – 16
- Number of rows – 2.
- It can display 32 characters at time

3. I2c Module (PCF8574A)

Technique:

In our project we used I2c module (PCF8574A) to reduce number of LCD pins which are connected to the microcontroller with 2 pins.

Specification:

- Number of input outputs -8
- Features -Interrupt pin
- Supply voltage -2.5V-6V
- Low standby-current consumption of 10 µA max.
- Addresses -8
- Frequency (Max) -100kHz
- Operating temperature range (C) -40 to 85

Schematic Diagram:

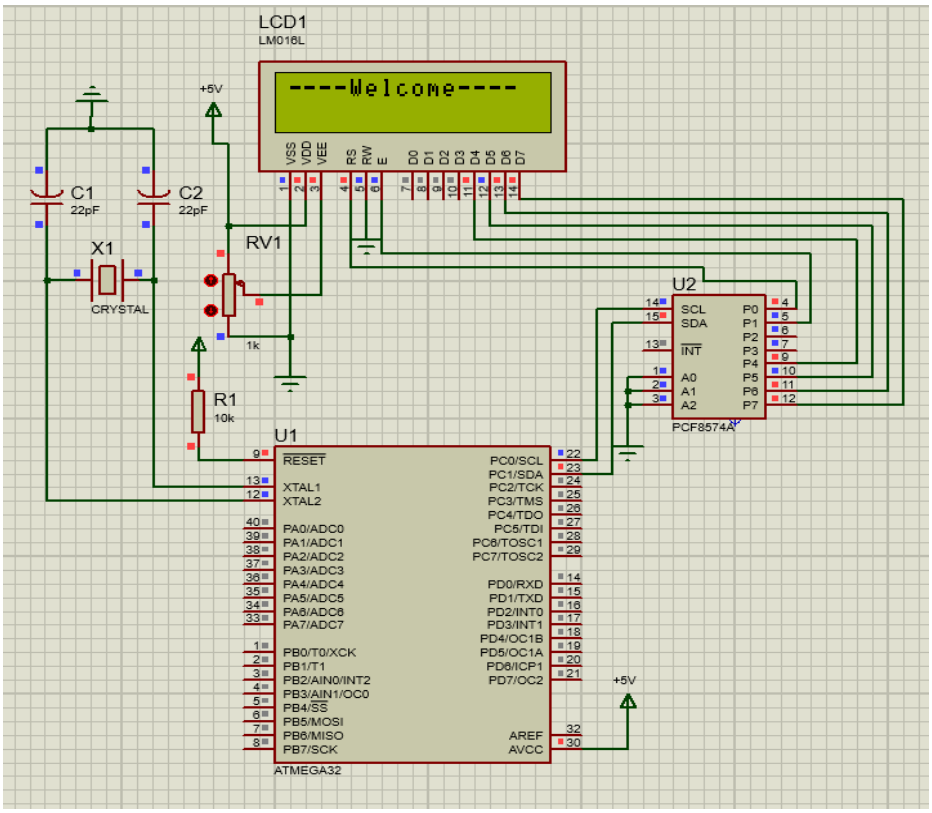


Figure 28 - LCD and I2c schematic diagram

PCB design:

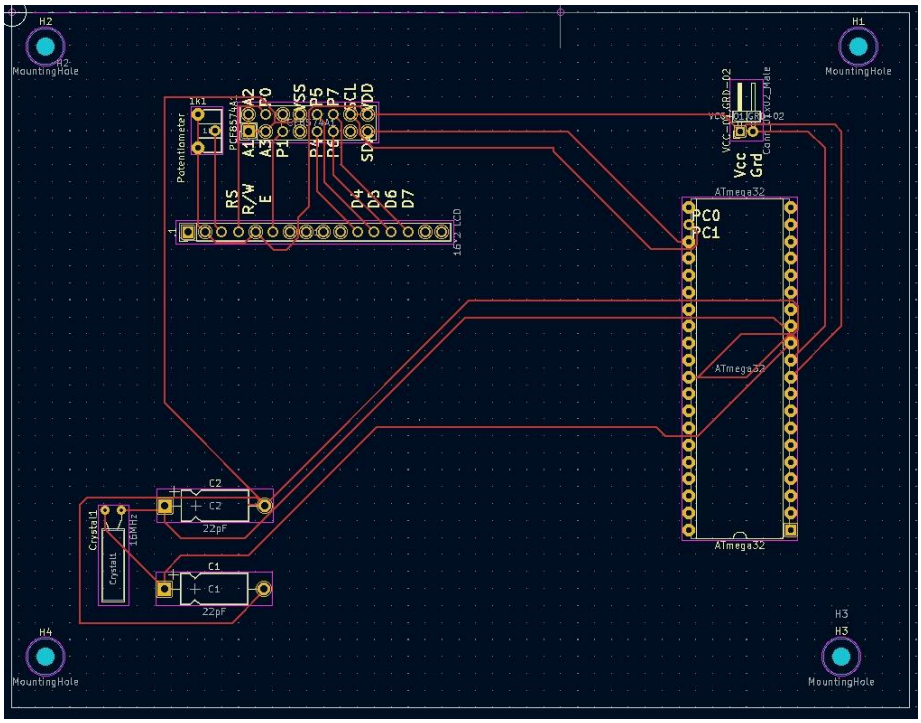


Figure 29 - LCD and I2c PCB diagram

Code:

```
#define F_CPU 8000000UL
#include <avr/io.h>
#include <util/delay.h>
#include "i2c.h"
#include "LCD_I2C.h"
int main(void)
{
	i2c_init();
	i2c_start();
	i2c_write(0x70);
	lcd_init();
		while (1)

			{
				lcd_cmd(0x80);
				lcd_msg("----Welcome----");
				_delay_ms(1000);
				lcd_cmd(0x01);
				_delay_ms(100);
				lcd_msg("scrap or not");
				lcd_cmd(0xC0);
				_delay_ms(200);
				lcd_msg("yes-1 | No-2");
				_delay_ms(1000);
				lcd_cmd(0x01);
				_delay_ms(200);
				lcd_msg("How many parts?");
				lcd_cmd(0xC0);
				_delay_ms(200);
				lcd_msg("One-1 | Two-2");
				_delay_ms(1000);
				lcd_cmd(0x01);
				lcd_msg("--Thank you!--");
				_delay_ms(1000);
				lcd_cmd(0x01);
				_delay_ms(1000);

			}
}
```

Member 2: Banu AGS (204018X)

Responsible part:

- Designed, tested, and implemented steppe motor systems.
- designed the power supply system
- used the blender software to create the animation of the machine

In our project I have took the responsibility of moving the cut disk by using a stepper motor and a motor driver. I've used stepper motor as it has Flexibility and supplied a constant holding torque without the need for the motor to be powered. I've designed the schematic diagram of power supply and stepper motor using proteus software. And used Atmel studio to program the stepper motor. To create the PCB designs for my modules I've used the Kicad software. And I've helped in 3D design and animation of the machine using the Blender software.

1. **Stepper motor**

Technique:

We're using a NEMA-17 stepper motor to move the cut disk. When the stepper motor rotates clockwise the cut disk system attached to the stepper motor shaft moves down towards the coconut. After breaking the coconut, the IR sensor gives a signal to the stepper motor, and it will rotate anti-clockwise. Then the cut disk moves upwards to the starting position.

Specifications:

- Working voltage – 12V
- Max current – 1.58A
- Holding torque -6.0kg-cm(85oz-in)
- step angle -200 steps/revolution
- Bi-polar stepper motor

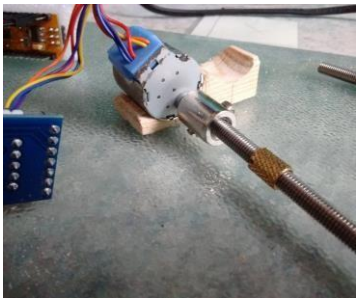


Figure 30 - Stepper motor

Circuit:

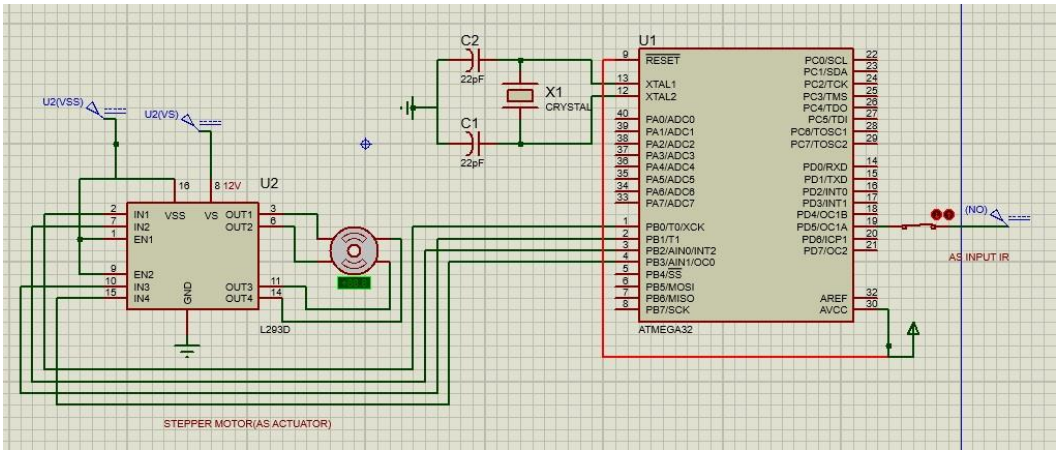


Figure 31 - Stepper motor schematic diagram

PCB Design:

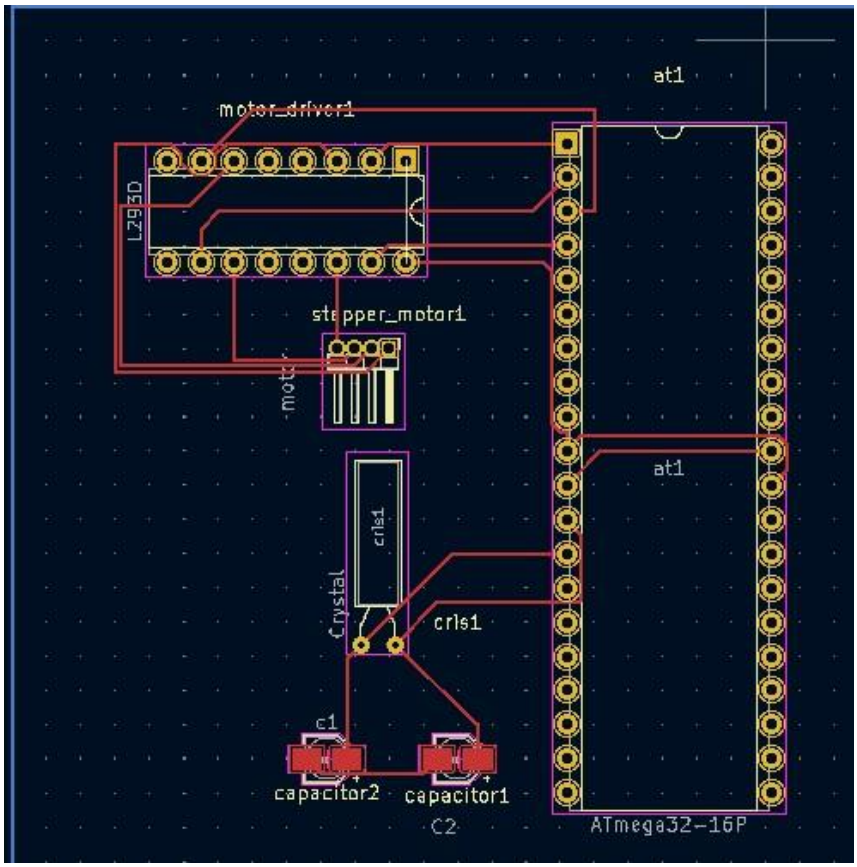


Figure 32 - Stepper motor PCB diagram

Code:

```
#define F_CPU 8000000UL    /* Define CPU Frequency 8MHz */
#include <avr/io.h>    /* Include AVR std. library file */
#include <util/delay.h>    /* Include delay header file */

int main(void)
{
  int period=100;
  DDRB |= 0B00001111;// making pins as output for stepper motor
  DDRD &= ~(1<<5) ; //AS IR INPUT
  while (1)
  {
    /* Rotate Stepper Motor clockwise*/
    if(!(PIND & 0b00100000)) // ir not detected
    {
      //power each winding in order to rotate
      PORTB = 0B00001000;
      _delay_ms(period);
      PORTB = 0B00000100;
      _delay_ms(100);
      PORTB = 0B00000010;
      _delay_ms(100);
      PORTB = 0B00000001;
      _delay_ms(period);
    }
```

```
/* Rotate Stepper Motor Anticlockwise */
    else if((PIND & 0b00100000)) // ir detected
    {
      for(int i=0;i<25;i++)
      {PORTB = 0B00000001;
      _delay_ms(period);
      PORTB = 0B00000010;
      _delay_ms(period);
      PORTB = 0B00000100;
      _delay_ms(100);
      PORTB = 0B00001000;
      _delay_ms(period);}
    }
  }
}
```

1. **L293D Driver**

Technique:

I have used an l293d driver to control the current flow through the stepper motor. Also, the direction of this motor can be controlled independently.

Specification:

- Motor voltage Vcc2 (VS): 4.5V to 36V
- Supply Voltage to Vcc1(VSS): 4.5V to 7V
- Maximum Peak motor current: 1.6A
- Speed and Direction control is possible

1. Power supply

Technique:

In our project we're using a power supply system to power up the components. I have used an ac step down transformer to convert ac 240v to 20v as our system requires a maximum voltage of 12V (to overcome current fluctuation issues I've taken 20V). Then used a rectifier circuit with 4 diodes to rectify the AC output from the transformer to DC. The capacitor is used as a filter to remove the ripples from the rectified DC signal. Ripples are AC components in the DC signal. Capacitor filters those AC components and provides a stable DC. I've used voltage regulators to provide the regulated output. In our project we need only 5v and 12v, so I selected 7805,7812 regulators to fill the requirements.

Circuit:

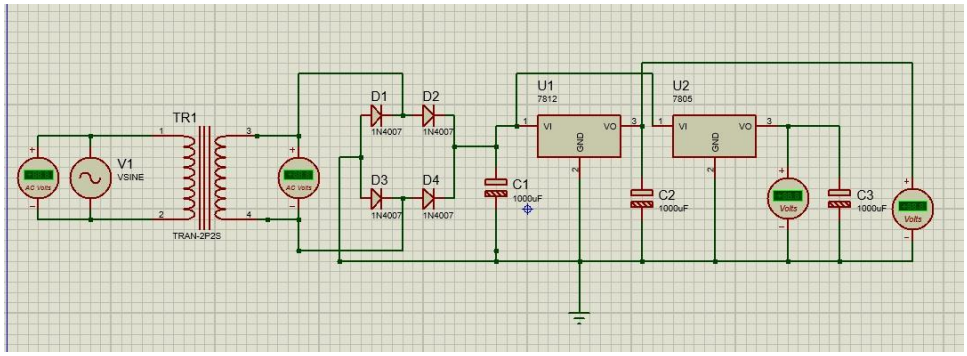


Figure 33 - Power supply schematic diagram

PCB Design:

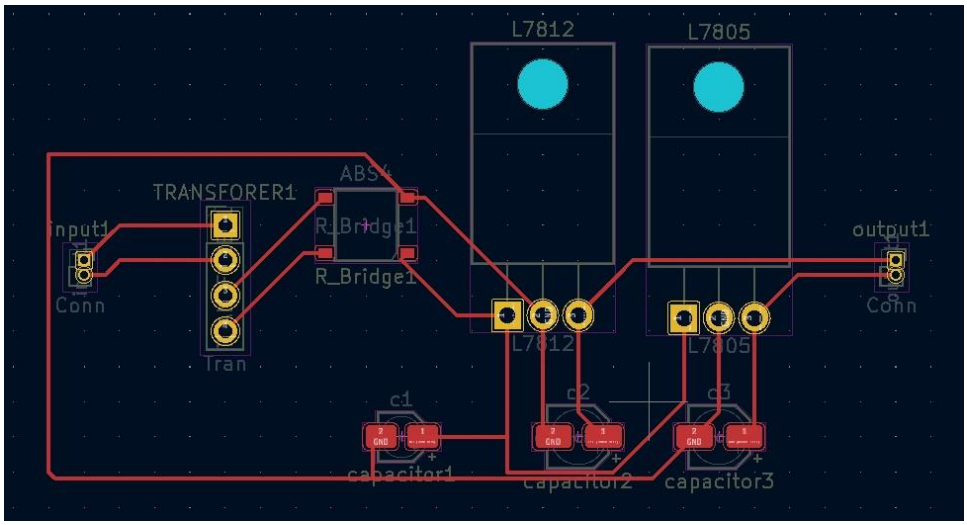


Figure 34 - Power supply PCB diagram

Member 3: Karunaweera R.L. (204096G)

Responsible Part:

My responsible components are the servo motor module and buzzer module. I have done the designing of schematic diagrams, PCB designs, and C codes for the above two modules. And I made the full PCB design of our machine. Also, I contributed to making the reports, presentations, and presenting as well.

1. **Servo Motor**

Technique:

Servo motor controls the fixing handles and points the coconut fixing handle towards the scraper by turning it by 90 degrees and holding it thereafter coconut breaks till the coconut scraping is finished as you can see in the below figures. And I selected SG Tower pro 90 Motor for our project.

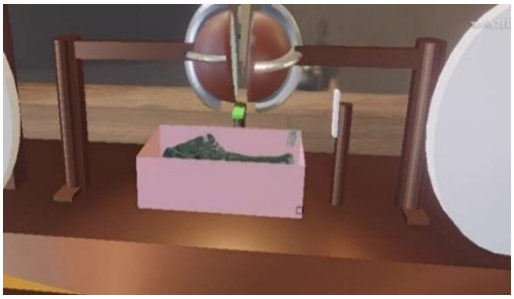


Figure 36

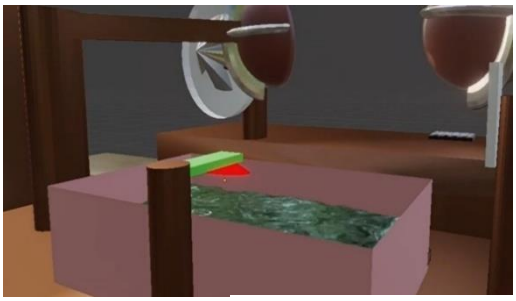


Figure 35

Specifications:

- The operating Voltage is +5V typically.
- Torque is 2.5kg/cm.
- The operating speed is 0.1s/60°.
- Gear Type is Plastic.
- Rotation is 0°-180°.
- Weight of motor is 9gm.
- The package includes gear horns and screws.

Schematic Diagram:

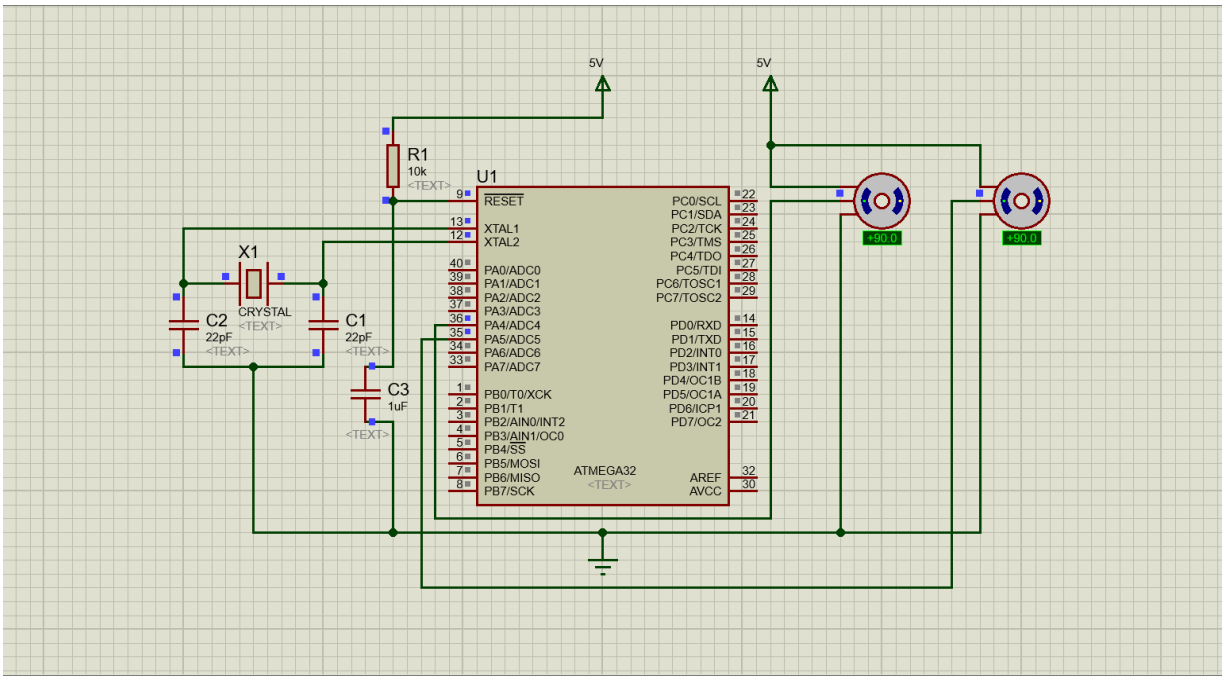


Figure 37 - Servo motor Schematic diagram

PCB design:

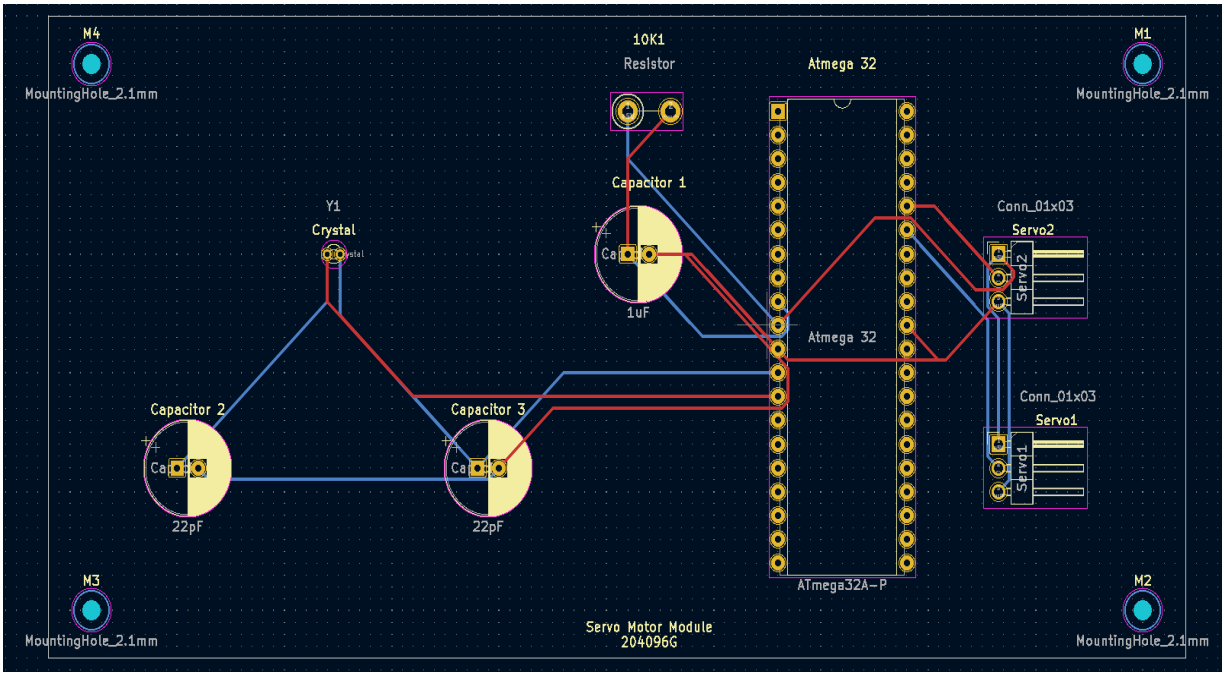


Figure 38 - Servo motor PCB diagram

Codes:

To rotate one servo motor

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

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

int main(void)
{
	DDRA= 0x10; //Makes PA4 output pin
	PORTA = 0x00;

	while(1)
	{
		//Rotate Motor to 0 degree
		PORTA = 0x10;
		_delay_us(1000);
		PORTA = 0x00;

		_delay_ms(2000);

		//Rotate Motor to 90 degree
		PORTA = 0x10;
		_delay_us(1500);
		PORTA = 0x00;

		_delay_ms(2000);
	}
}
```

To rotate both servo motors

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

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

int main(void)
{
	DDRA= 0x30; //Makes PA4 output pin
	PORTA = 0x00;

	while(1)
	{
		//Rotate Motor to 0 degree
		PORTA = 0x30;
		_delay_us(1000);
		PORTA = 0x00;

		_delay_ms(2000);

		//Rotate Motor to 90 degree
		PORTA = 0x30;
		_delay_us(1500);
		PORTA = 0x00;

		_delay_ms(2000);
	}
}
```

2. **Buzzer**

Technique:

The purpose of using the buzzer in our machine is to notify the user if the coconut water tank is full. So based on the signal of the ultrasonic sensor it will ring. If the ultrasonic sensor senses the water level is high, then it will send a signal through the microcontroller and the buzzer will start to ring. That is shown in the below picture.

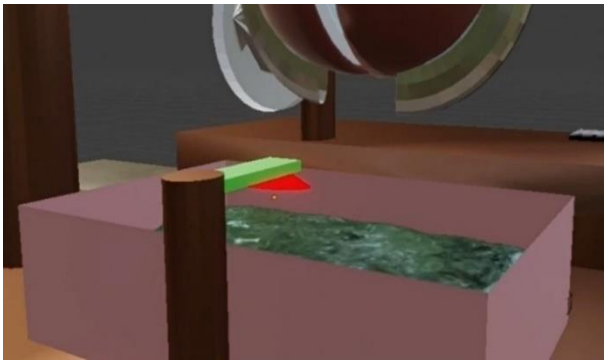


Figure 39

Specifications:

- Rated Voltage is 6V DC
- Operating Voltage is 4-8V DC
- Rated current is less than 30mA
- Sound Type is Continuous Beep
- Resonant Frequency is 2300 Hz

Schematic Diagram:

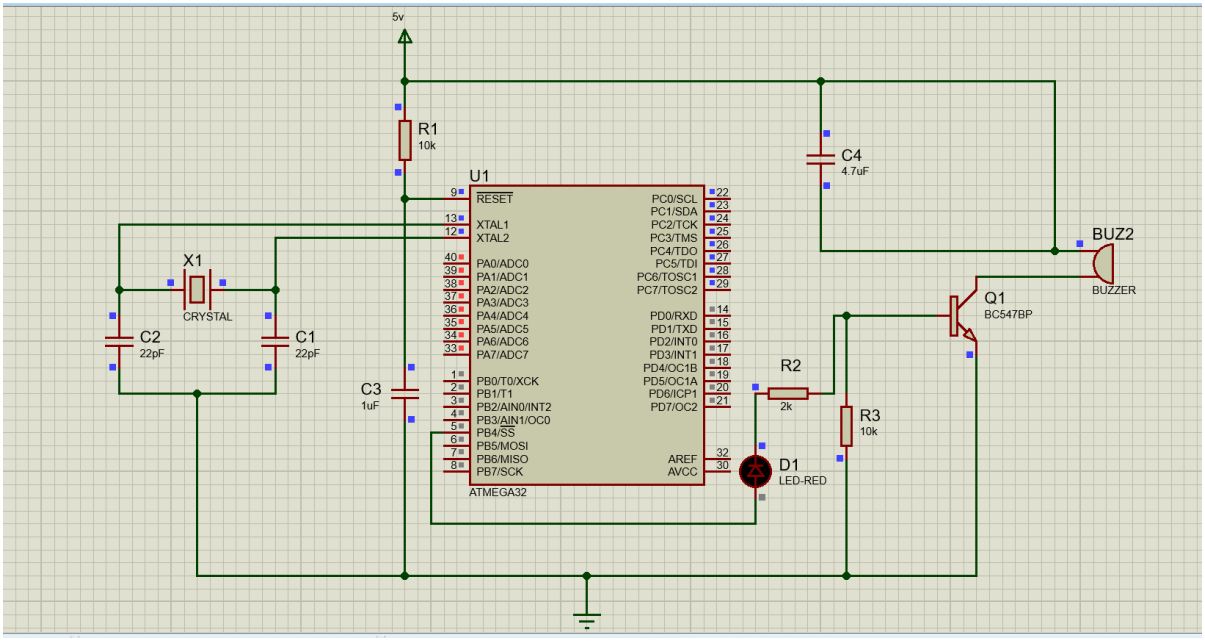


Figure 40 - Buzzer schematic diagram

PCB design:

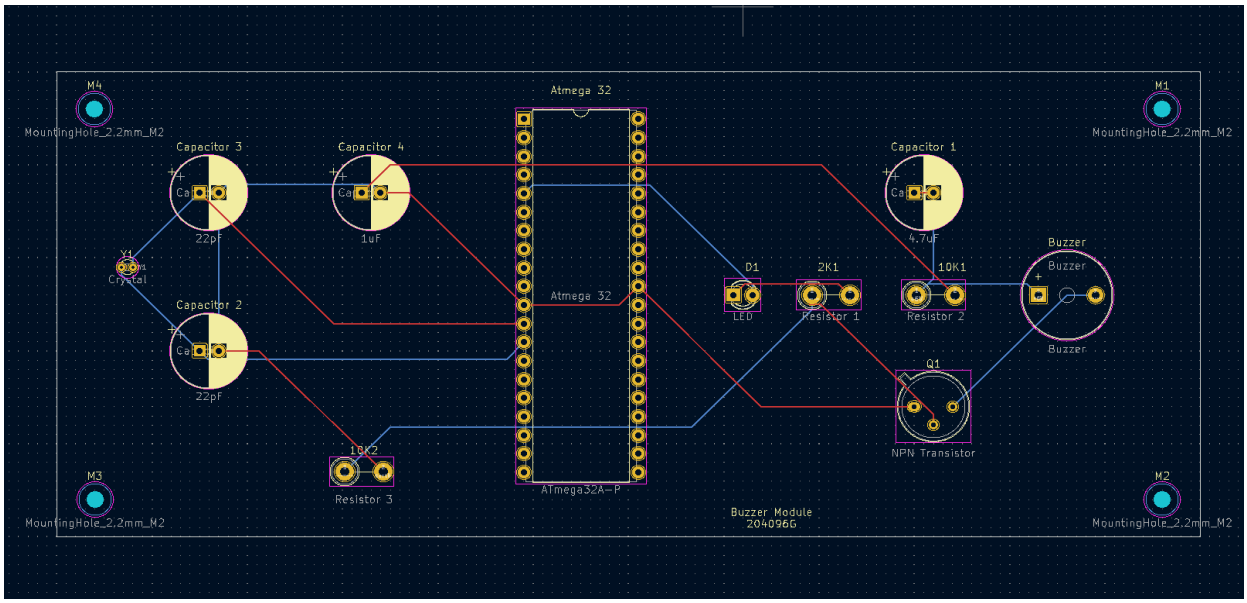


Figure 41 - Buzzer PCB diagram

Compiled Code:

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

int main()
{
	DDRB = 0x20; //Makes PORTC as output
	DDRD = 0x00; //Makes PORTD as Input to get signal from Ultrasonic sensor
	PORTD = 0xFF; // Enable The PullUps of PORTD.
	while(1)
	{
		if(PIND==0x01) // Read the signal from Ultrasonic sensor and Turn ON/OFF the Buzzer
		{
			PORTB=0x20;
		}
		else
		{
			PORTB=0x00;
		}
	return 0;
}
}
```

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

Responsible Part:

My responsible components are the DC motor module and color sensor module. I have done the designing of schematic diagrams, PCB designs, and C codes for the above two modules. Also, I contributed to making the reports, presentations, and presenting as well.

1. **DC motor**

Technique:

A dc motor converts direct current electrical energy into mechanical energy. The machine uses dc motors to rotate the cut disk, blender, and the scrappers by 360˚.There are 3 main types of dc motors. dc shunt motor, dc series motor and dc compound motor. The shunt motor is more suitable for the scrappers because the speed of a dc shunt motor is sufficiently constant. A compound dc motor is more suitable for cut disk because compound motors provide high starting torque and good speed regulation.

Specifications:

- Maximum Voltage - 12V
- Maximum Current - 0.64A
- Rated Speed (RPM) -200
- Rated Torque(kg-cm) -1.5
- Stall Torque(kg-cm) - 5.4

Figure 42

Circuit:

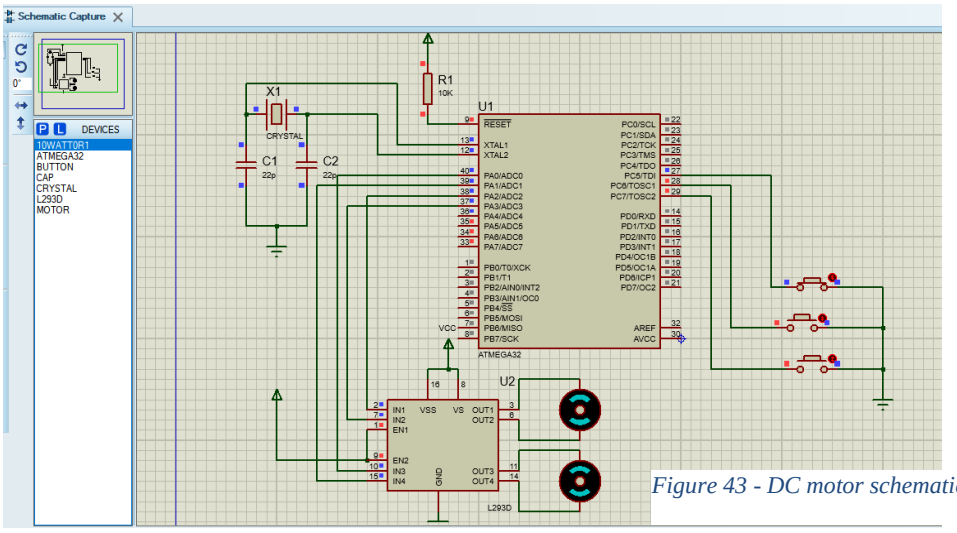


Figure 43 - DC motor schematic diagram of scrapers

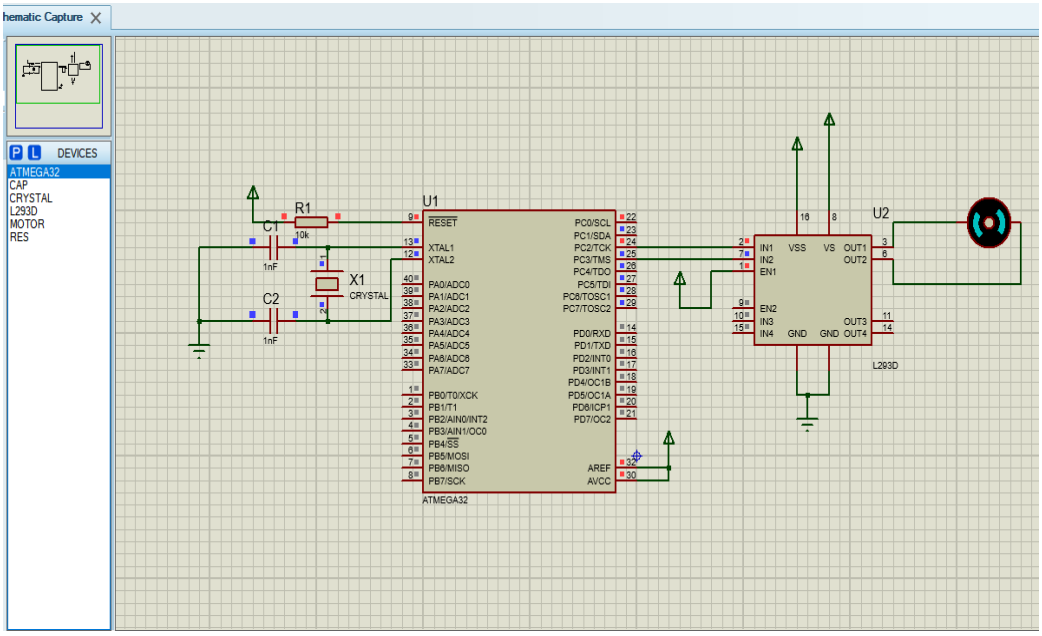


Figure 44 - DC motor schematic diagram of cut disk and blender

PCB design:

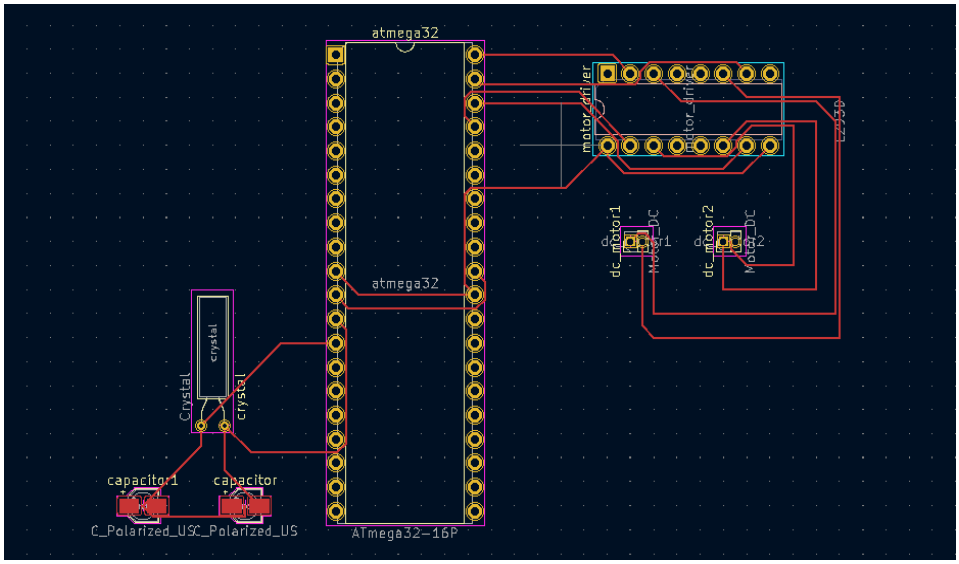


Figure 45 - DC motor PCB diagram of scrapers

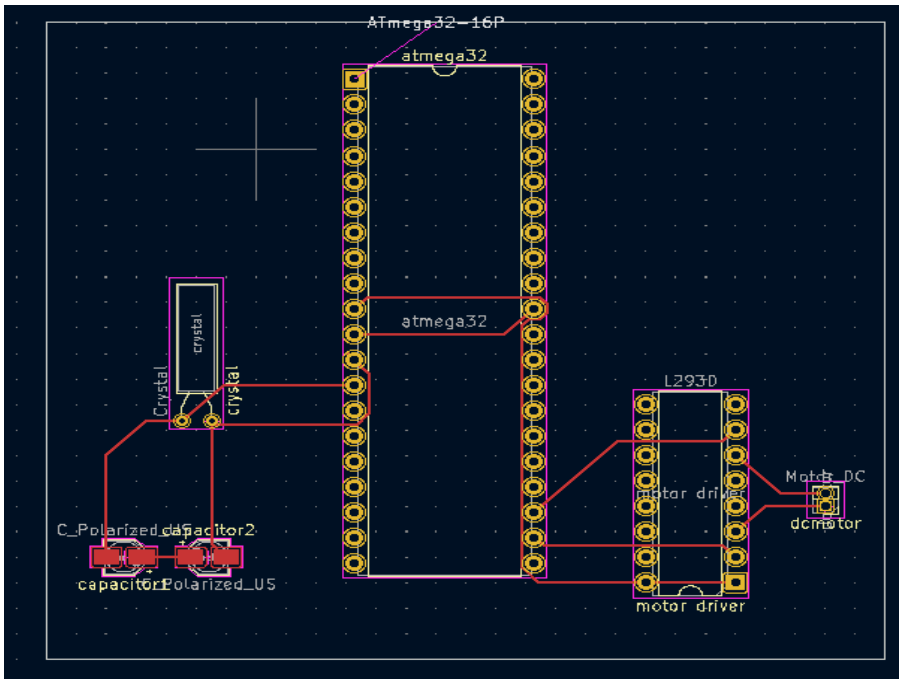


Figure 46 - DC motor PCB diagram of cut disk and blender

Code:

```
// code for the scrapers

#include <avr/io.h>
#define F_CPU 16000000UL
#include "util/delay.h"
//Test the  PINC7
#define white_both (PINC&0b10000000)
//Test the PINC6
#define brown    (PINC&0b01000000)
//Test the PINC5
#define white_one (PINC&0b00100000)
//OUTPUT DATA FOR MOTOR
#define run_both 0b11100101
#define stop_running 0b11100000
#define run_one_motor 0b11100001 int
main(void){
   DDRA=0b00011111;//PINC7.5 INPUT
   PORTC=0b11100000; //PC7.5 SET TO
HIGH while (1) {
      if (white_both==0){
         PORTA=stop_running;
         _delay_ms(100);
         PORTA=run_both;
      }
```

```
else if (brown==0){
         PORTA=stop_running;
         _delay_ms(100);
         PORTA=stop_running;
      }
      else if (white_one==0){
         PORTA=stop_running;
         _delay_ms(100);
PORTA=run_one_motor;
      }
      }
```

```
// code for the cut disk and blender

#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){
DDRC= 0XFF; //direction of port B as output
   while(1) // infinite loop{
    PORTC= 0X05; //motor rotation in clockwise direction
            _delay_ms(2000); //delay of 2  sec

            PORTC = 0X00; // motor stopped
            _delay_ms(2000); // delay of 2  sec
              }
}
```

2. Color sensor

Technique:

It can Detect color of an object. Color converts the intensity of incident radiation into frequency. This is used to detect the color of coconut shells(brown) in the machine.

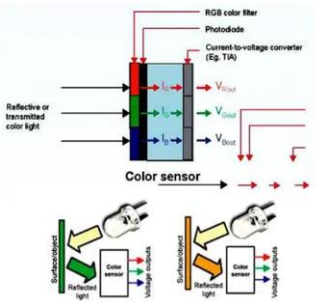


Figure 47 - Color sensor

Specifications:

- Working voltage: 2.7 - 5.5v
- Working current: 1.4 mA
- Dimension: 28.4 x 28.4 mm
- Interface: digital TTL
- Weight: 4.17g
- Programmable color

Circuit:

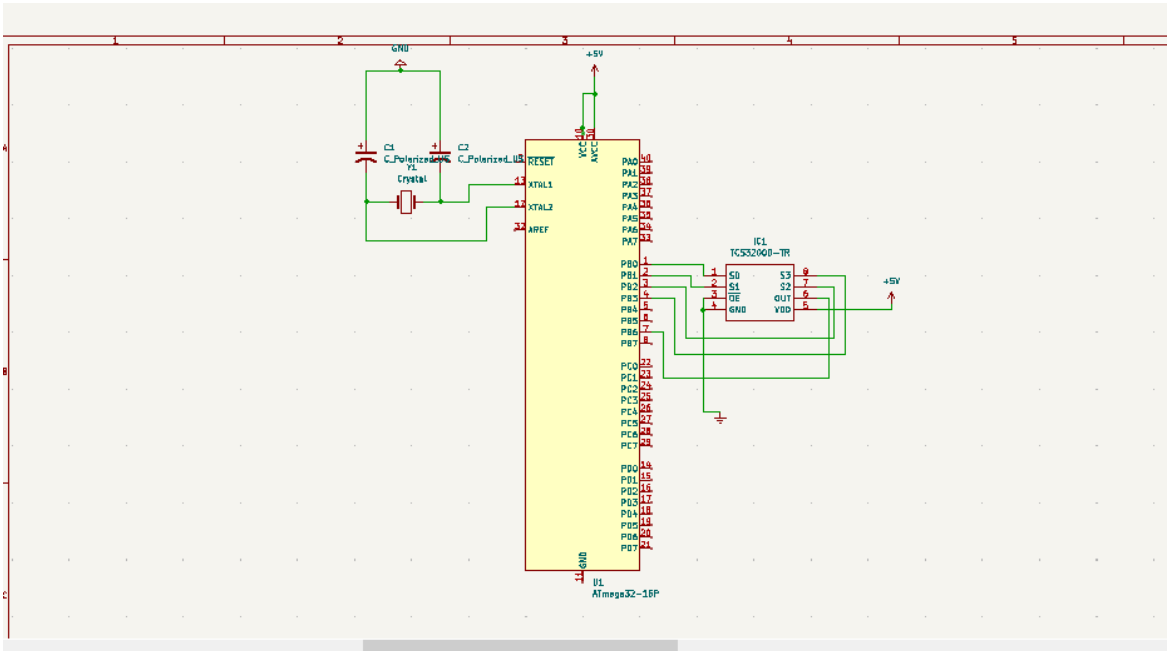


Figure 48 - Color sensor schematic diagram

PCB design:

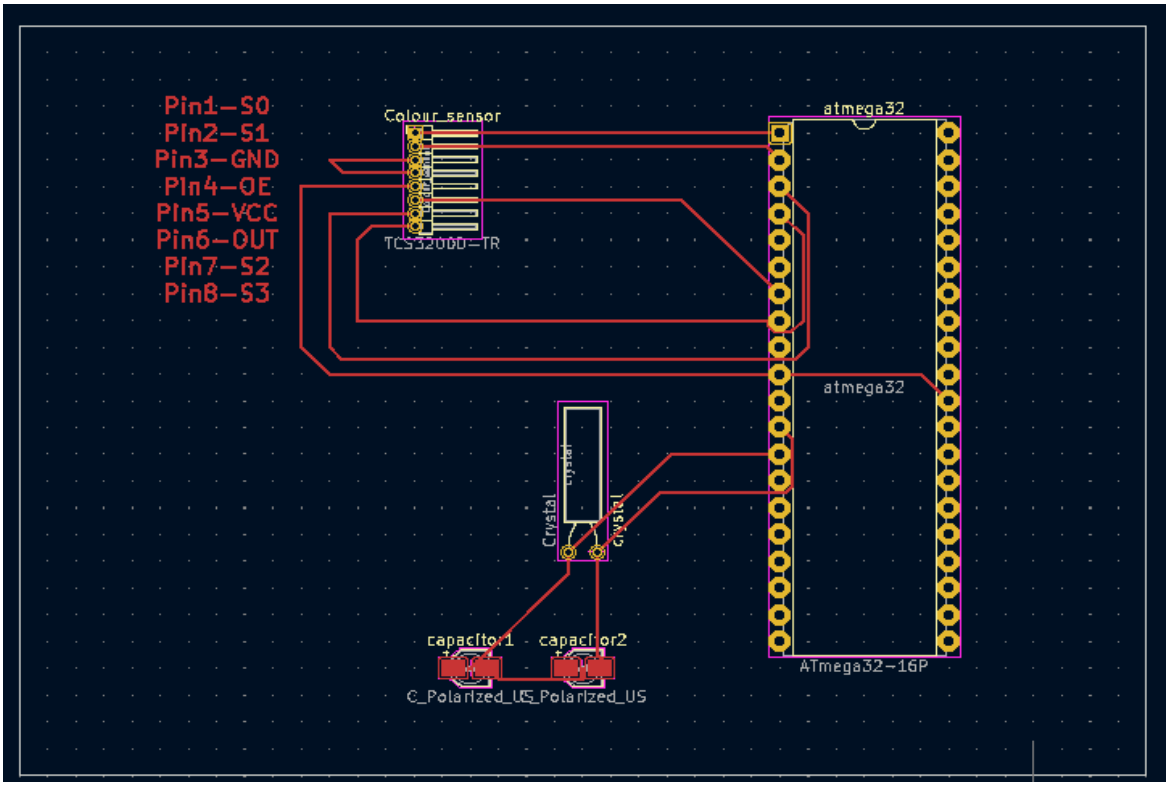


Figure 49 - Color sensor PCB diagram

Code:

```
#include <avr/io.h>
#include <util/delay.h>
#include "lib/colour_sensor/tcs3200.h"

uint32_t MeasureR();  //measure frequency with red filter
uint32_t MeasureG(); //measure frequency with green filter
uint32_t MeasureB(); //measure frequency with blue filter
uint32_t MeasureC();  //measure frequency without filter
int main(void){
InitTCS3200();  //Initialize TCS Library
uint8_t x=0;
int8_t vx=1;
while(1){
  TCSLEDOn();
    uint32_t v1=MeasureC();
    delay_ms(100);

     TCSLEDOff();
     uint32_t v2=MeasureC();

     uint32_t d=v1-v2;

     if(d>8000) //detect colour change{
     if(colour==red){
//stop motor
      }
       else if (colour==white){
 //rotate motor
      }
                uint32_t r,g,b;   //Show
TCSLEDOn();
r=MeasureR();
g=MeasureG();
b=MeasureB();
TCSLEDOff();
 uint32_t smallest;

        if(r<b){
        if(r<g)
            smallest=r;
          else
            smallest=g;
        }
        Else{
       if(b<g)
            smallest=b;
          else
            smallest=g;
       }
uint32_t _r,_g,_b;
smallest=smallest/10;  //decrease the
smallest value with factor of 10
 _r=r/smallest;   //r, g, b are the value which
have scaled
       _g=g/smallest;
       _b=b/smallest;
 }   } }
   uint32_t MeasureR(){
     TCSSelectRed();
      uint32_t r;
     _delay_ms(10);
      r=TCSMeasure();
     _delay_ms(10);
      r+=TCSMeasure();
     _delay_ms(10);
      r+=TCSMeasure();
return r/3.3;
}
uint32_t MeasureG(){
      TCSSelectGreen();
      uint32_t r;
     _delay_ms(10);
     r=TCSMeasure();
     _delay_ms(10);
     r+=TCSMeasure();
     _delay_ms(10);
    r+=TCSMeasure();
return r/3;
}
uint32_t MeasureG(){
    TCSSelectGreen();
    uint32_t r;
   _delay_ms(10);
  r=TCSMeasure();
  _delay_ms(10);
  r+=TCSMeasure();
 _delay_ms(10);
  r+=TCSMeasure();
return r/3; }
```

```
uint32_t MeasureB(){
    TCSSelectBlue();
    uint32_t r;
    _delay_ms(10);
  r=TCSMeasure();
   _delay_ms(10);
  r+=TCSMeasure();
    _delay_ms(10);
  r+=TCSMeasure();
return r/4.2; }

uint32_t MeasureC(){
    TCSSelectClear();
    uint32_t r;
   _delay_ms(10);
   r=TCSMeasure();
   _delay_ms(10);
   r+=TCSMeasure();
   _delay_ms(10);
   r+=TCSMeasure();
return r/3;
}
}
```

3. L293D motor driver

Technique:

Output voltage and current of microcontroller is not sufficient to drive a dc motor. So, in the machine l293d motor drivers provide up to 600mA current in the voltage range from 4.5 to 36V to drive a dc motor.

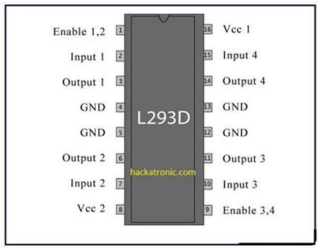


Figure 50 - Motor driver

Specifications:

- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Output Current :600 mA Per Channel
- High-Noise-Immunity Inputs

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

Responsible part:

My responsible components are the servo motor module and buzzer module. I have done the designing of schematic diagrams, PCB designs, and C codes for the above two modules. Also, I contributed to making the reports, presentations, and presenting as well.

1. IR sensor

Technique:

There are two LEDs. one of them emits IR Ray. the other receives IR rays. if that IR ray touches any surface, it reflects. Then the reflected IR ray is caught by the IR receiver. then the sensor sends a signal to fulfill our need. In our project, the IR sensor is used to detect the cut disk.

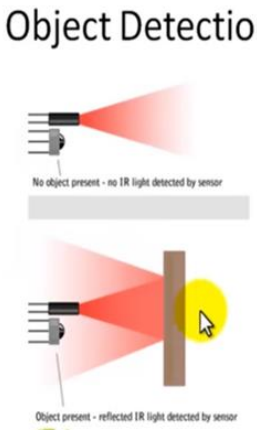


Figure 51

Specifications:

- Working voltage: 3.6 - 5v
- Working current: 20mA
- Distance range: 2cm - 30cm
- Dimensions: 48 * 14 * 8 mm
- Weight: 15g

Circuit:

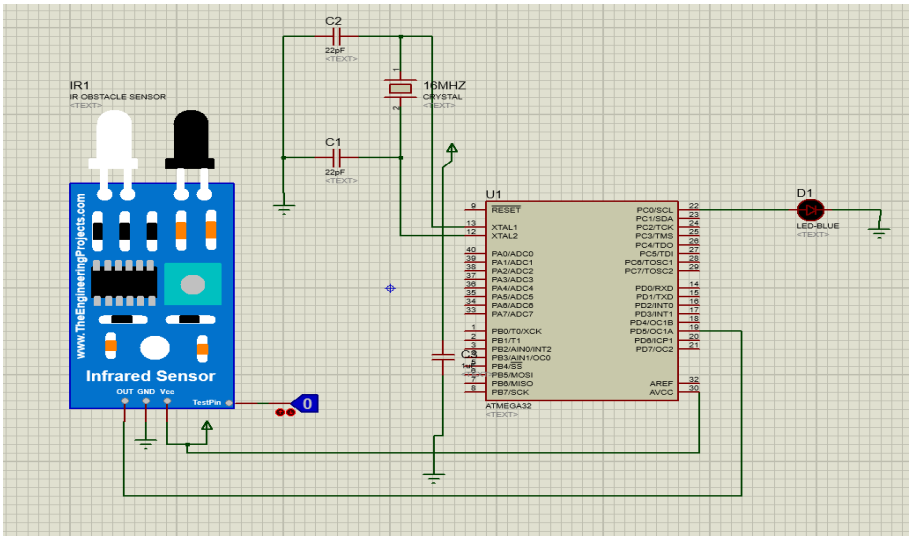


Figure 52 - IR sensor schematic diagram

PCB design:

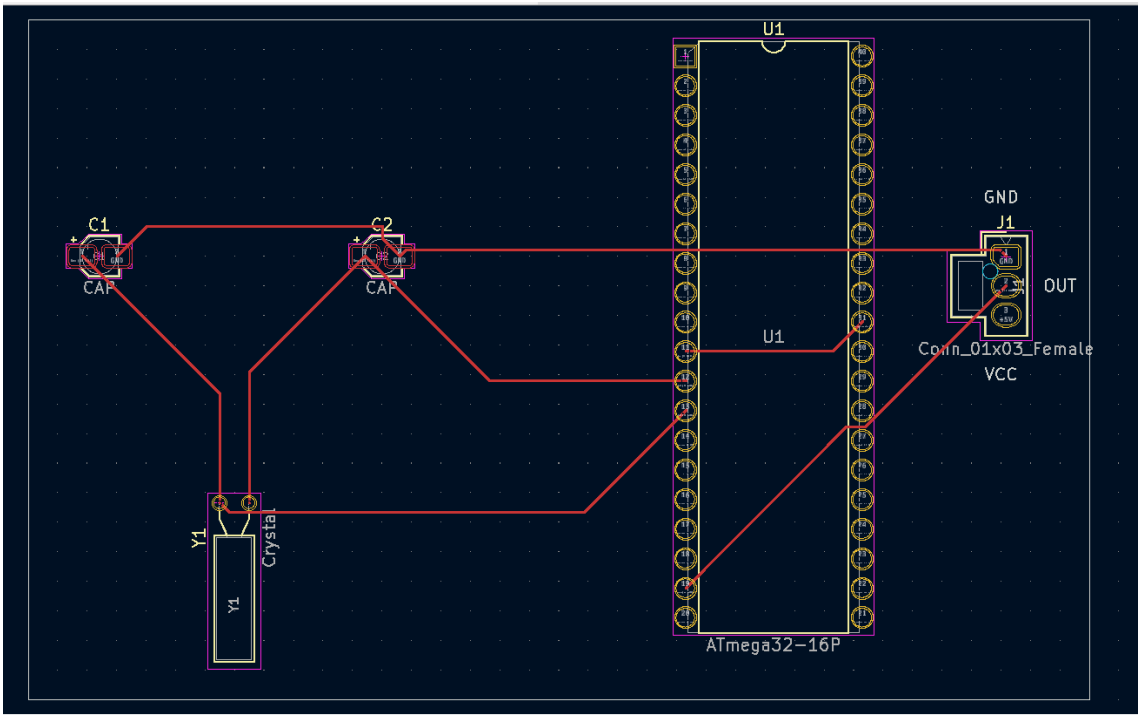


Figure 53 - IR sensor PCB diagram

Code:

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

	}
}
```

2. 4*3 Keypad

Technique:

This is a keypad containing 12 membrane buttons which can give inputs. In our project keypad is used to give inputs which should be displayed on the LCD display. We give numbers as inputs using the keypad.

*Figure 54 - 4*3 Keypad*

Specifications:

- Working voltage: 24v
- Working current: 20mA
- Dimensions: 9 * 8 * 4 mm
- Weight: 10g
- Ribbon cable length: 90mm

Circuit:

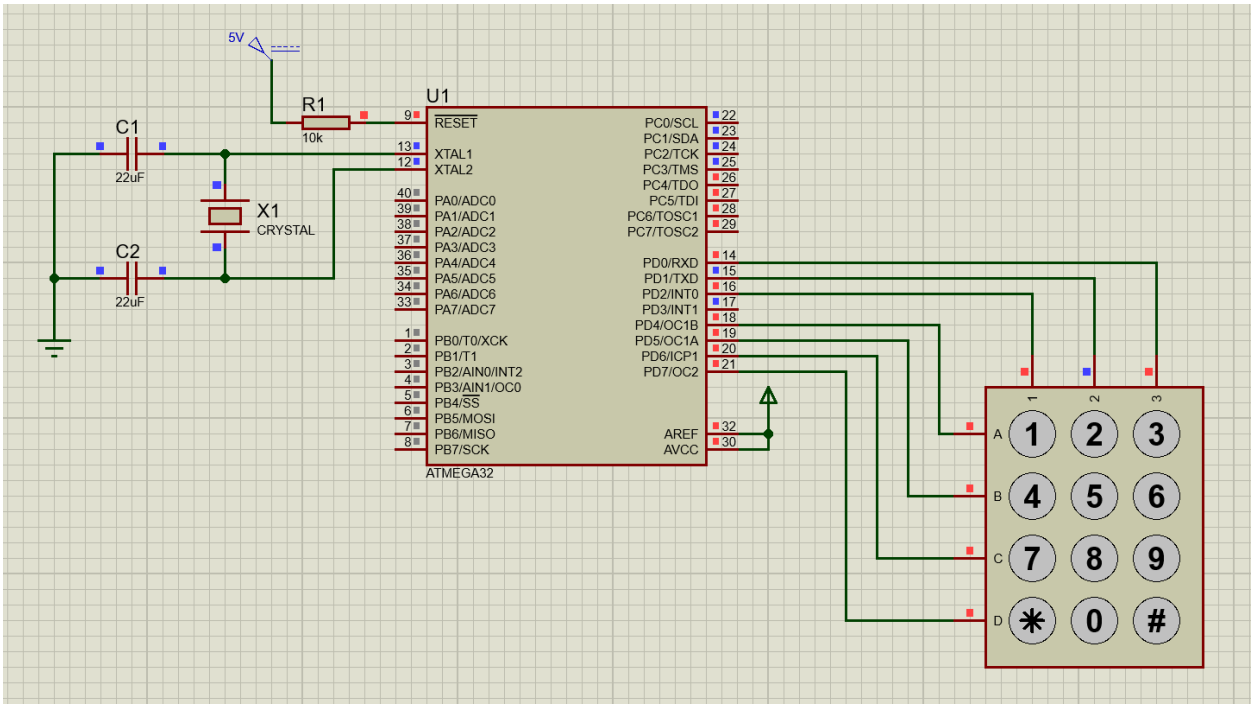


Figure 55 - Keypad schematic diagram

PCB design:

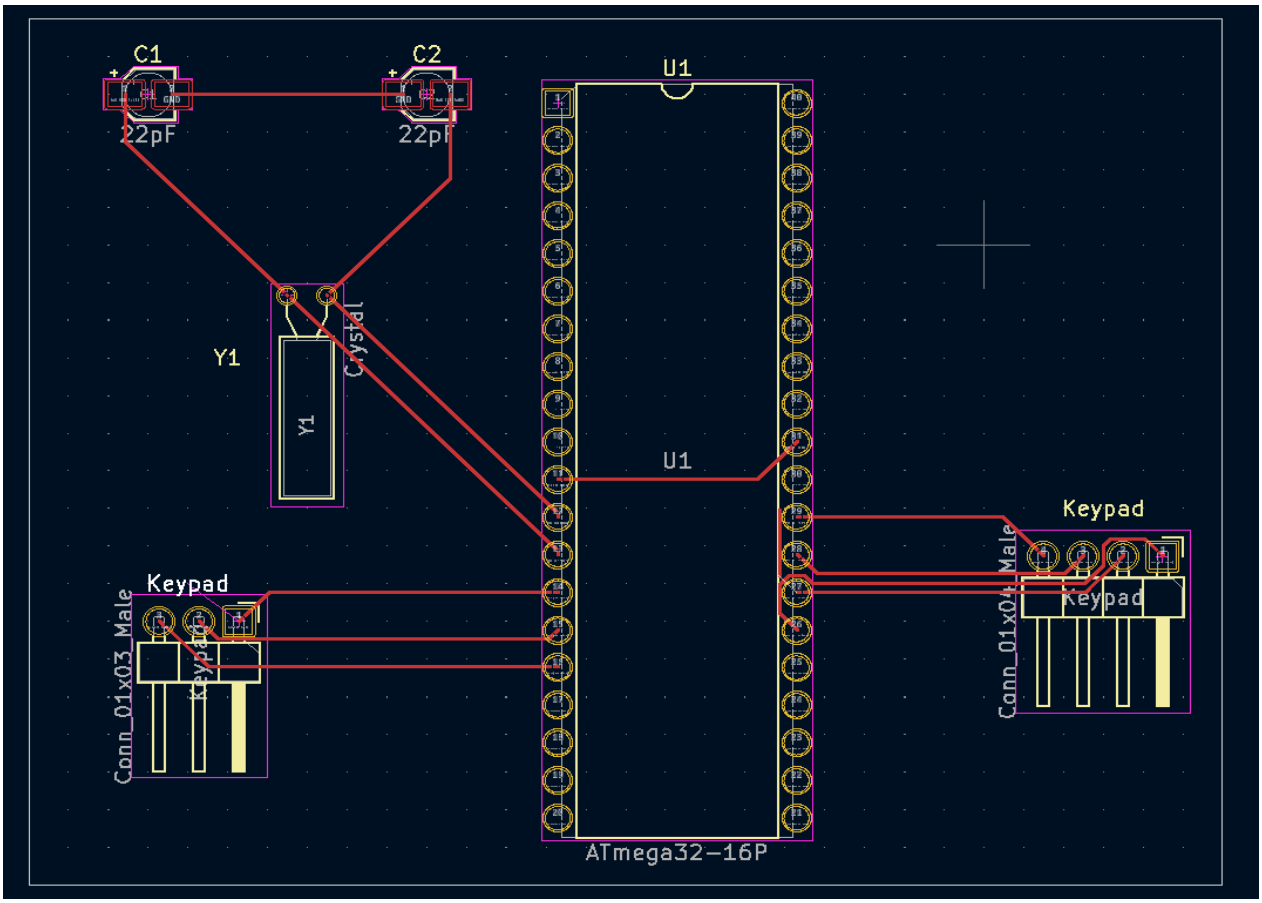


Figure 56 - Keypad PCB diagram

Code:

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

int main(void)
{
   PORTC &=~ 0xf0 ; //PIN4-7 of port c is low
        PORTD |= ( 1<<0) ; //PIN0 of port D is
high
        PORTD |= ( 1<<1) ;
        PORTD |= ( 1<<2) ;

        PORTC |= 0b11100000; // make 1st row
0
        PORTC &= ~ 0b00010000;
        if((PIND & (1<<PIND0))==0){
                _delay_ms(30);
                return '1';
                }else if((PIND &
(1<<PIND1))==0){
                _delay_ms(30);
                return '2';
                }else if((PIND &
(1<<PIND2))==0){
                _delay_ms(30);
                return '3';
        }
        PORTC |=0b11010000;
        PORTC &= ~ 0b00100000;
        if((PIND & (1<<PIND0))==0){
                _delay_ms(30);
                return '4';
                }else if((PIND &
(1<<PIND1))==0){
                _delay_ms(30);
                return '5';
                }else if((PIND &
(1<<PIND2))==0){
                _delay_ms(30);
                return '6';
        }
PORTC|=0b10110000;
        PORTC &= ~ 0b01000000;
        if((PIND & (1<<PIND0))==0){
                _delay_ms(30);
                return '7';
                }else if((PIND &
(1<<PIND1))==0){
                _delay_ms(30);
                return '8';
                }else if((PIND &
(1<<PIND2))==0){
                _delay_ms(30);
                return '9';
        }
        PORTC|=0b01110000;
        PORTC &= ~ 0b10000000;
        if((PIND & (1<<PIND0))==0){
                _delay_ms(30);
                return 'C';
                }else if((PIND &
(1<<PIND1))==0){
                _delay_ms(30);
                return '0';
                }else if((PIND &
(1<<PIND2))==0){
                _delay_ms(30);
                return '=';
        }
}

void delay(int ms)
{
        int i,j;
        for(i=0;i<=ms;i++)
        for(j=0;j<=120;j++);
}
```