Ahmedabad University

Sensors, Instruments and Experiments

Section 3

Embedded Systems Design

Section 2

Project Report

The Four-Legged Rover

Submitted to.: Prof. Sanket Patel
Submitted on: 7th December, 2022

Name	**Enrollment Number**
Astha Bhalodiya	AU2040067
Mrugen Fichadia	AU2040075
Shreya Karia	AU2040076
Yesha Dhivar	AU2040215
Twinkle Popat	AU2040216

The Four-Legged Rover

Contents

I. INTRODUCTION

The Four-Legged Rover is a modified version of a simple rover where it walks on four legs instead of wheels which helps in better movement and overall balance of the structure. It works on 12 servo motors where each leg comprises 3 motors each. The rover can have forward, backward, left and right movements based on the code. An ultrasound sensor is also used to detect obstacles and turn the rover in the left direction when done so.

II. CIRCUIT DIAGRAM

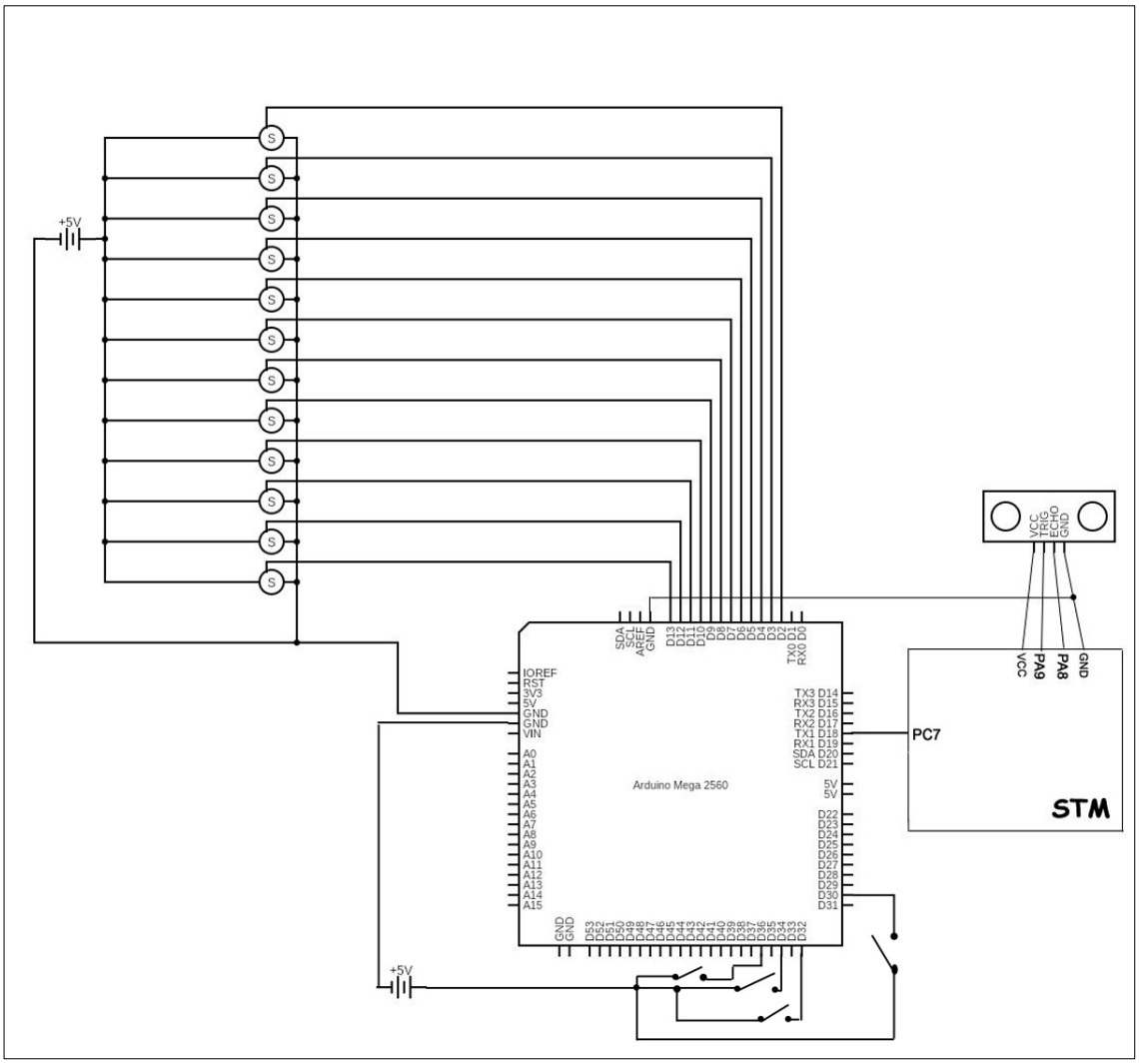

Figure 1: Circuit Diagram

III. List of Components

Component	Quantity	Cost per piece	Acquired From
Servo Motors (SG-90)	12	100	University
Ultrasound Sensor	1	52	University
Arduino Mega	1	800	University
STM32F0RB	1	280	University
3D Printing	660 gm	1000	University

IV. The Process of Building

Firstly, various parts like the base and leg components of the rover were designed using Fusion360, a 3D design software. These components were 3D printed in the fabrication shop within the next week. After receiving the components, they were connected and tested along with all the technical components and their fitting. There are a total four legs in the rover and each leg is controlled with the help of 3 servo motors resulting in a total of 12 servo motors.

Figure 2: 3D Design in Fusion360

Following this the servos were configured to a base condition for the rover to stand. After the configuration, these servos were tested against different conditions for ideating its movement. Post ideation, code for movement was written using Arduino IDE to control Arduino Mega, a microcontroller. These movements were done by changing the base angles of the servos according to the movement required.

Theory of support polygon is followed for the movement of the rover. The rover is considered stable when the self-weight projection of the rover is within the support polygon. The support polygon is obtained by connecting all the surface contact points of the rover.

Along with the movement, the rover also does obstacle detection. For this purpose, an ultrasonic sensor is used, which stops the movement of the rover when an obstacle is detected. This sensor is interfaced with STM32F070RB, through input capture mode of Timer 1. The trigger pulse is sent through the transmitter and the echo wave is received and captured through input capture mode. The time between the transmission and receiving of the pulse is used to measure the distance. When this obtained distance is less than a threshold value, a signal is sent to the Arduino and an interrupt is generated. This interrupt will change the direction of the rover to avoid the obstacle.

V. PROBLEMS ENCOUNTERED

During the whole journey of building the rover, there were several issues that were faced and some of them were tackled. At the initial stage, some issues were faced with the 3D printed parts of the rover. Some designs were not foresighted in terms of sections from where we would wire the structure. So, the pieces were manually carved to meet the requirements.

Furthermore, when the base and the legs of the rover were connected, a new issue surfaced as it was difficult to carry the weight of the whole rover for the legs. So, we connected a wheel to the base for providing support and stability to the model.

There were multiple failures related to the servo motors that were faced throughout the journey. This problem was handled by replacing the old servo with the new ones.

After the connection of servo motors with the Arduino Mega, it was observed that the Arduino Mega faced problems in providing sufficient voltage and current to the motors. This resulted in lagging of the motors. An external power source was used to solve this issue.

After the leg movement started, it was brought to notice that the movements of the legs were very aggressive when the legs came in contact with the surface. A layer of glue with the glue gun was applied to the base of the legs to reduce the impact and provide some friction to the movements.

VI. CODE

A. STM32F070RB Code:

Note: Only main snippets of code are included.

```
/* USER CODE BEGIN 0 */
void delay(uint16_t time){
	__HAL_TIM_SET_COUNTER(&htim1,0);
	while(__HAL_TIM_GET_COUNTER(&htim1)<time);
}
```

```
uint32_t IC_Val1 = 0;
uint32_t IC_Val2 = 0;
uint32_t Difference = 0;
uint8_t Is_First_Captured = 0;  // is the first value captured ?
uint8_t Distance  = 0;
int cnt_ccb=0, cnt_hscr=0;

#define TRIG_PIN GPIO_PIN_9
#define TRIG_PORT GPIOA

// Let's write the callback function
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)
{
	cnt_ccb++;
	if (htim->Channel == HAL_TIM_ACTIVE_CHANNEL_1)  // if the interrupt source is
channel1
	{
		if (Is_First_Captured==0) // if the first value is not captured
		{
			IC_Val1		=		HAL_TIM_ReadCapturedValue(htim,
TIM_CHANNEL_1); // read the first value
			Is_First_Captured = 1;  // set the first captured as true
			// Now change the polarity to falling edge
			__HAL_TIM_SET_CAPTUREPOLARITY(htim,
TIM_CHANNEL_1, TIM_INPUTCHANNELPOLARITY_FALLING);
		}

		else  // if the first is already captured
		{
			IC_Val2		=		HAL_TIM_ReadCapturedValue(htim,
TIM_CHANNEL_1);  // read second value
			__HAL_TIM_SET_COUNTER(htim, 0);  // reset the counter

			if (IC_Val2 > IC_Val1)
			{
				Difference = IC_Val2-IC_Val1;
			}

			else if (IC_Val1 > IC_Val2)
			{
				Difference = (0xffff - IC_Val1) + IC_Val2;
			}

			Distance = Difference * .034/2;
			if(Distance<=20){
				HAL_GPIO_TogglePin( GPIOC, GPIO_PIN_7);
			}
			Is_First_Captured = 0; // set it back to false

			// set polarity to rising edge
```

```
                        __HAL_TIM_SET_CAPTUREPOLARITY(htim,
TIM_CHANNEL_1, TIM_INPUTCHANNELPOLARITY_RISING);
                        __HAL_TIM_DISABLE_IT(&htim1, TIM_IT_CC1);
                }
        }
}
void HCSR04_Read (void)
{
        HAL_GPIO_WritePin(TRIG_PORT, TRIG_PIN, GPIO_PIN_RESET);
        HAL_Delay(10);
        HAL_GPIO_WritePin(TRIG_PORT, TRIG_PIN, GPIO_PIN_SET);  // pull the TRIG
pin HIGH
        delay(10);  // wait for 10 us
        HAL_GPIO_WritePin(TRIG_PORT, TRIG_PIN, GPIO_PIN_RESET);  // pull the
TRIG pin low
        __HAL_TIM_ENABLE_IT(&htim1, TIM_IT_CC1);
                cnt_hscr++;
}

/* USER CODE END 0 */
```

In the Main Function:

```
/* USER CODE BEGIN 2 */
HAL_TIM_IC_Start_IT(&htim1, TIM_CHANNEL_1);
  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
                HAL_GPIO_TogglePin(led_GPIO_Port,led_Pin);
                HCSR04_Read();
                HAL_Delay(500);
  }
  /* USER CODE END 3 */
}
```

B. Arduino Code:

```
#include <Servo.h>
Servo servo[4][3];
//define servos' ports
const int servo_pin[4][3] = { {3,4,2}, {12,13,11}, {9,10,8}, {6, 7, 5} }; //1b 1c 1a 4b 4c 4a
3b 3c 3a 2b 2c 2a
```

```
int
a1=90,b1=160,c1=160,a2=90,b2=160,c2=160,a3=90,b3=160,c3=140,a4=90,b4=160,c4=160;
int val[4][3] = {{160,130,90},{155,90,90},{170,120,90},{160,170,90}};
int cnt=1;

void hi(){
  Serial.println("hi");
  val[0][0]=180;
  val[0][1]=100;
  servo[0][0].write(val[0][0]);
  servo[0][1].write(val[0][1]);
  for(int j=0;j++;j<30){
    servo[0][1].write(val[0][1]++);
    delay(35);
  }
  for(int j=0;j++;j<30){
    servo[0][1].write(val[0][1]--);
    delay(35);
  }
  for(int j=0;j++;j<30){
    servo[0][1].write(val[0][1]++);
    delay(35);
  }
  for(int j=0;j++;j<30){
    servo[0][1].write(val[0][1]--);
    delay(35);
  }
  servo[0][1].write(val[0][1]+=50);
  delay(30);
  for(int k=0;k<20;k++){
    servo[0][0].write(val[0][0]--);
    delay(15);
  }
  reset();
}

void reset()
{
 val[0][0] = 160;
 val[0][1] = 130;
 val[0][2] = 90;
 val[1][0] = 155;
 val[1][1] = 90;
 val[1][2] = 90;
 val[2][0] = 170;
 val[2][1] = 120;
 val[2][2] = 90;
 val[3][0] = 160;
 val[3][1] = 170;
 val[3][2] = 90;
```

```
for (int i = 0; i < 4; i++)
 {
   for (int j = 0; j < 3; j++)
   {
     servo[i][j].write(val[i][j]);
     delay(20);
   }
 }

 delay(2000);

}

void reset1()
{
 val[0][0] = 175;
 val[0][1] = 160;
 val[0][2] = 90;
 val[1][0] = 155;
 val[1][1] = 90;
 val[1][2] = 90;
 val[2][0] = 170;
 val[2][1] = 120;
 val[2][2] = 90;
 val[3][0] = 160;
 val[3][1] = 170;
 val[3][2] = 90;

 for (int i = 0; i < 4; i++)
 {
   for (int j = 0; j < 3; j++)
   {
     servo[i][j].write(val[i][j]);
     delay(20);
   }
 }

 delay(2000);

}

void pushforward(){
  val[0][2]+=10;
  val[2][2]-=10;
  val[1][2]+=10;
  val[3][2]-=10;
  val[0][0]+=10;
  val[0][1]+=15;
  val[1][0]-=20;
```

```
  val[1][1]-=40;
  val[3][0]-=15;
  val[3][1]-=25;
  val[2][0]+=17;
  val[2][1]+=15;
  servo[0][2].write(val[0][2]);
  servo[1][2].write(val[1][2]);
  servo[2][2].write(val[2][2]);
  servo[3][2].write(val[3][2]);
  servo[0][0].write(val[0][0]);
  servo[0][1].write(val[0][1]);
  servo[1][0].write(val[1][0]);
  servo[1][1].write(val[1][1]);
  servo[3][0].write(val[3][0]);
  servo[3][1].write(val[3][1]);
  servo[2][0].write(val[2][0]);
  servo[2][1].write(val[2][1]);
}

void pushbackward(){
  val[0][2]-=10;
  val[2][2]+=10;
  val[1][2]-=10;
  val[3][2]+=10;
  val[0][0]-=10;
  val[0][1]-=15;
  val[1][0]+=20;
  val[1][1]+=40;
  val[3][0]+=15;
  val[3][1]+=25;
  val[2][0]-=17;
  val[2][1]-=15;
  servo[0][2].write(val[0][2]);
  servo[1][2].write(val[1][2]);
  servo[2][2].write(val[2][2]);
  servo[3][2].write(val[3][2]);
  servo[0][0].write(val[0][0]);
  servo[0][1].write(val[0][1]);
  servo[1][0].write(val[1][0]);
  servo[1][1].write(val[1][1]);
  servo[3][0].write(val[3][0]);
  servo[3][1].write(val[3][1]);
  servo[2][0].write(val[2][0]);
  servo[2][1].write(val[2][1]);
}

void moveForward(){
  //Movement of first leg, far from leg 3
      reset();
       servo[0][0].write(val[0][0]+=20);
```

```
    delay(200);
    servo[0][1].write(val[0][1]-=30);
    delay(200);
    servo[0][2].write(val[0][2]-=30);
    for(int i=0; i<30;i++){
     servo[0][0].write(val[0][0]--);
     delay(20);
    }
    //delay(1000);
    Serial.println("Leg 1 moved");
    delay(200);
    pushforward();
    Serial.println("Pushed...");
    delay(500);


  //Move leg 2 inside i.e towards leg 4
    servo[3][0].write(val[3][0]+=40);
    delay(200);
    servo[3][2].write(val[3][2]+=30);
    servo[3][1].write(val[3][1]+=60);
    delay(200);
    Serial.println("Leg 2 moved");
    delay(200);
    pushforward();
    Serial.println("Pushed");
    delay(500);

 //Move the leg 3 outside i.e 3a move far from leg 1
    servo[2][0].write(val[2][0]+=20);
    delay(200);
    servo[2][1].write(val[2][1]-=30);
    delay(200);
    servo[2][2].write(val[2][2]+=30);
    delay(200);
    for(int i=0; i<30;i++){
     servo[2][0].write(val[2][0]--);
     delay(20);
    }
    delay(200);
    pushforward();
    Serial.println("Pushed");
    delay(500);

  //Move leg 4 inside i.e towards 2
    servo[1][0].write(val[1][0]+=110);
    Serial.println("P-1");
    delay(200);
    servo[1][1].write(val[1][1]+=135);
    servo[1][2].write(val[1][2]-=30);
```

```
      Serial.println("P-2");
      delay(200);
     pushforward();
      Serial.println("Done...!!!!");
      delay(500);
      reset();
  }

void moveBackward(){
  reset1();
  //Move leg 4 inside i.e towards 2
      servo[1][0].write(val[1][0]+=20);
      Serial.println("P-1");
      delay(1000);
      servo[1][1].write(val[1][1]-=30);
      servo[1][2].write(val[1][2]-=30);
      Serial.println("P-2");
      delay(1000);
     for(int i=0; i<30;i++){
       servo[1][0].write(val[1][0]--);
       delay(20);
      }
     pushbackward();
      Serial.println("leg 1");
      delay(1000);
      //Move the leg 3 outside i.e 3a move far from leg 1
      servo[2][0].write(val[2][0]+=40);
      delay(1000);
      servo[2][1].write(val[2][1]+=30);
      delay(1000);
      servo[2][2].write(val[2][2]+=60);
      delay(1000);
      pushbackward();
      Serial.println("Pushed");
      delay(1000);

    //Move leg 2 inside i.e towards leg 4
      servo[3][0].write(val[3][0]+=20);
      delay(1000);
      servo[3][2].write(val[3][2]-=30);
      servo[3][1].write(val[3][1]+=30);
      delay(1000);
      for(int i=0; i<30;i++){
       servo[3][0].write(val[3][0]--);
       delay(20);
      }
      Serial.println("Leg 2 moved");
      delay(1000);
      pushbackward();
      Serial.println("Pushed");
```

```
       delay(1000);
       servo[3][2].write(val[3][2]+=20);
       delay(1000);

//Movement of first leg, far from leg 3
      reset();
       servo[0][0].write(val[0][0]+=110);
       delay(1000);
       servo[0][1].write(val[0][1]+=135);
       delay(1000);
       servo[0][2].write(val[0][2]-=30);
       Serial.println("Leg 1 moved");
       delay(1000);
       Serial.println("Pushed...");
       pushbackward();
       Serial.println("Done...!!!!");
       delay(1000);
       reset();
  }

void turnAround1(){
  reset();
  //Turn leg 3
  Serial.println("1");
  val[2][0]+=20;
    servo[2][0].write(val[2][0]);
    Serial.println("2");
    delay(200);
    val[2][2]-=40;
    servo[2][2].write(val[2][2]);
    Serial.println("2");
    delay(200);
    for(int i=0;i<30;i++){
      val[2][0]-=1;
      servo[2][0].write(val[2][0]);
      delay(15);
    }
    Serial.println("3");
    delay(200);
    val[1][2]+=20;
    val[3][2]+=20;
    val[0][2]+=20;
    val[2][2]+=20;
    servo[1][2].write(val[1][2]);
    Serial.println("4");
    servo[3][2].write(val[3][2]);
    Serial.println("5");
    servo[0][2].write(val[0][2]);
    Serial.println("6");
    servo[2][2].write(val[2][2]);
```

```
    delay(200);

  //Turn leg 2
  val[3][0]+=20;
    servo[3][0].write(val[3][0]);
    delay(200);
    val[3][2]-=40;
    servo[3][2].write(val[3][2]);
    delay(200);
    for(int i=0;i<30;i++){
      val[3][0]-=1;
      servo[3][0].write(val[3][0]);
      delay(15);
    }
    delay(200);
    val[1][2]+=20;
    val[3][2]+=20;
    val[0][2]+=20;
    val[2][2]+=20;
    servo[1][2].write(val[1][2]);
    servo[3][2].write(val[3][2]);
    servo[0][2].write(val[0][2]);
    servo[2][2].write(val[2][2]);
    delay(200);

  //Turn leg 4
 val[1][0]+=20;
    servo[1][0].write(val[1][0]);
    delay(200);
    val[1][2]-=40;
    servo[1][2].write(val[1][2]);
    delay(200);
    for(int i=0;i<30;i++){
      val[1][0]-=1;
      servo[1][0].write(val[1][0]);
      delay(15);
    }
    delay(200);
    val[1][2]+=20;
    val[3][2]+=20;
    val[0][2]+=20;
    val[2][2]+=20;
    servo[1][2].write(val[1][2]);
    servo[3][2].write(val[3][2]);
    servo[0][2].write(val[0][2]);
    servo[2][2].write(val[2][2]);
    delay(200);

  //Turn leg 1
   val[0][0]+=20;
```

```
    servo[0][0].write(val[0][0]);
    delay(200);
    val[0][2]-=40;
    servo[0][2].write(val[0][2]);
    delay(200);
    for(int i=0;i<30;i++){
      val[0][0]-=1;
      servo[0][0].write(val[0][0]);
      delay(15);
    }
    delay(200);
    val[1][2]+=20;
    val[3][2]+=20;
    val[0][2]+=20;
    val[2][2]+=20;
    servo[1][2].write(val[1][2]);
    servo[3][2].write(val[3][2]);
    servo[0][2].write(val[0][2]);
    servo[2][2].write(val[2][2]);
    delay(500);
    reset();
}

//Turn around 2
void turnAround2(){
  reset();
  //Turn leg 3
  Serial.println("1");
  val[2][0]+=20;
    servo[2][0].write(val[2][0]);
    Serial.println("2");
    delay(200);
    val[2][2]+=40;
    servo[2][2].write(val[2][2]);
    Serial.println("2");
    delay(200);
    for(int i=0;i<30;i++){
      val[2][0]-=1;
      servo[2][0].write(val[2][0]);
      delay(15);
    }
    Serial.println("3");
    delay(200);
    val[1][2]-=20;
    val[3][2]-=20;
    val[0][2]-=20;
    val[2][2]-=20;
    servo[1][2].write(val[1][2]);
    Serial.println("4");
    servo[3][2].write(val[3][2]);
```

```
    Serial.println("5");
    servo[0][2].write(val[0][2]);
    Serial.println("6");
    servo[2][2].write(val[2][2]);
    delay(200);

  //Turn leg 2
  val[3][0]+=20;
    servo[3][0].write(val[3][0]);
    delay(200);
    val[3][2]+=40;
    servo[3][2].write(val[3][2]);
    delay(200);
    for(int i=0;i<30;i++){
      val[3][0]-=1;
      servo[3][0].write(val[3][0]);
      delay(15);
    }
    delay(200);
    val[1][2]-=20;
    val[3][2]-=20;
    val[0][2]-=20;
    val[2][2]-=20;
    servo[1][2].write(val[1][2]);
    servo[3][2].write(val[3][2]);
    servo[0][2].write(val[0][2]);
    servo[2][2].write(val[2][2]);
    delay(200);

  //Turn leg 4
 val[1][0]+=20;
    servo[1][0].write(val[1][0]);
    delay(200);
    val[1][2]+=40;
    servo[1][2].write(val[1][2]);
    delay(200);
    for(int i=0;i<30;i++){
      val[1][0]-=1;
      servo[1][0].write(val[1][0]);
      delay(15);
    }
    delay(200);
    val[1][2]-=20;
    val[3][2]-=20;
    val[0][2]-=20;
    val[2][2]-=20;
    servo[1][2].write(val[1][2]);
    servo[3][2].write(val[3][2]);
    servo[0][2].write(val[0][2]);
    servo[2][2].write(val[2][2]);
```

```
    delay(200);

 //Turn leg 1
  val[0][0]+=20;
    servo[0][0].write(val[0][0]);
    delay(200);
    val[0][2]+=40;
    servo[0][2].write(val[0][2]);
    delay(200);
    for(int i=0;i<30;i++){
      val[0][0]-=1;
      servo[0][0].write(val[0][0]);
      delay(15);
    }
    delay(200);
    val[1][2]-=20;
    val[3][2]-=20;
    val[0][2]-=20;
    val[2][2]-=20;
    servo[1][2].write(val[1][2]);
    servo[3][2].write(val[3][2]);
    servo[0][2].write(val[0][2]);
    servo[2][2].write(val[2][2]);
    delay(500);
   reset();
}
void blink()
{
  reset();
  for(int i=0;i<5;i++){
  digitalWrite(19, HIGH); // Toggle LED
  Serial.println("interrupt");
  delay(500);
  digitalWrite(19, LOW);
  delay(500);
 }
Serial.println("I");
turnAround1();
reset();
}

void setup()
{
//Initialize the interrupt
pinMode(18, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(18), blink, CHANGE);
pinMode(19, OUTPUT);
pinMode(30, INPUT);
pinMode(32, INPUT);
pinMode(34, INPUT);
```

```
pinMode(36, INPUT);

//initialize all servos
 for (int i = 0; i < 4; i++)
 {
   for (int j = 0; j < 3; j++)
   {
     servo[i][j].attach(servo_pin[i][j]);
     delay(20);
   }
 }
 for (int i = 0; i < 4; i++)
 {
   for (int j = 0; j < 3; j++)
   {
     servo[i][j].write(val[i][j]);
     delay(20);
   }
 }
digitalWrite(30,LOW);
digitalWrite(32,LOW);
digitalWrite(34,LOW);
digitalWrite(36,LOW);
 Serial.begin(9600);
}

void loop(void)
{
    if(digitalRead(30)==HIGH){
        //reset();
        moveForward();//blue
          Serial.println("L1");
     }
     if(digitalRead(32)==HIGH){
        Serial.println("L2");
        //reset();
        moveBackward();//red
     }
     if(digitalRead(34)==HIGH){
      Serial.println("L3");
        //reset();
        turnAround2();//green
     }
     if(digitalRead(36)==HIGH){
      Serial.println("L4");
        //reset();
        turnAround1();//yellow
     }
     else{
        reset();
```

```
        moveForward();
      }
  }
```

VII. TECHNOLOGIES USED

A. Hardware Components:

i. STM32F070RB Microcontroller
ii. Arduino Mega Microcontroller
iii. HC-SR04 Ultrasonic Sensor

B. Software Components:

i. Keil uVision 5
ii. STM32CubeMX
iii. Arduino IDE

VIII. OUTPUT

Video of working model: https://drive.google.com/file/d/1gcSEh2CBVezo1jTxYsw4-pe5Ayo66bRN/view

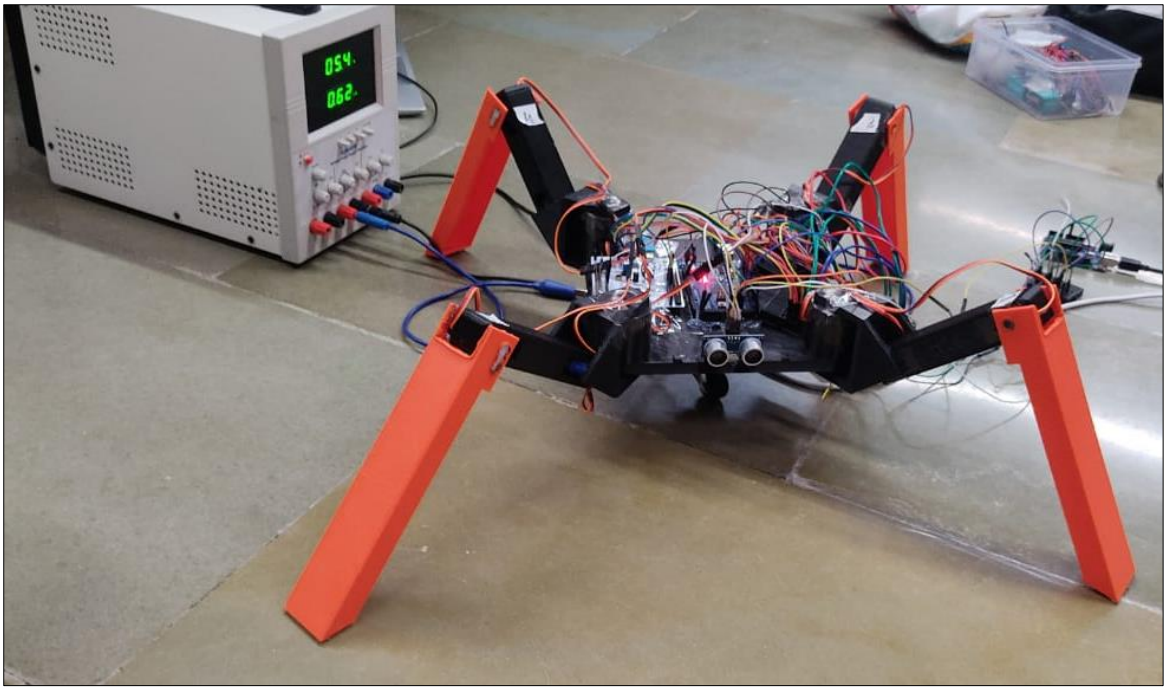

Figure 3: The Working Model of Rover

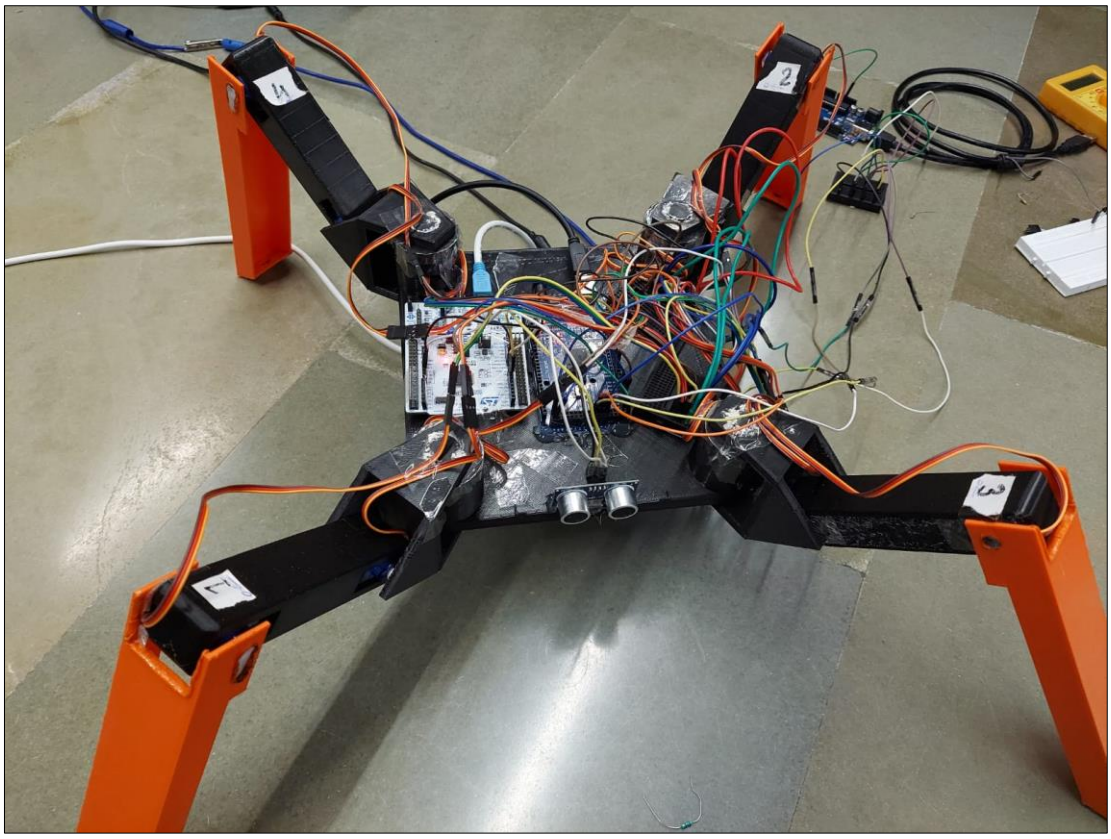

Figure 4: The Working Model of Rover

IX. ANALYSIS

i. The overall weight of the rover is 660gm. Here, the weight of each leg is 115gm and weight of the base is 200gm.

ii. The rover can make 4 different movements turn right, turn left, move forward and move backward.

iii. The rover requires 5.3V voltage and 1-1.5A range of current.

iv. The ultrasonic sensor has a range of 0-33cm. The distance greater than 33 cm generates a garbage value and does not give an accurate answer. In our project, an interrupt is generated and a green-colored led is turns on, if an obstacle is found at a distance of less than 20cm.

v. The setup position of the servo motor in the rover is as follows:
Note: Each leg of the rover has 3 servo motor, which is termed as 1A, 1B, 1C for the first leg.

- The servo motors (1A, 1B, 1C) of the first leg are configured at angle of 90°, 160°, 130°, respectively.
- The servo motors (2A, 2B, 2C) of the first leg are configured at angle of 90°, 160°, 170°, respectively.
- The servo motors (3A, 3B, 3C) of the first leg are configured at angle of 90°, 170°, 120°, respectively.
- The servo motors (4A, 4B, 4C) of the first leg are configured at angle of 90°, 155°, 90°, respectively.

X. CONCLUSION

Four-legged rover built by us is a preliminary prototype of a rover which can be used for archaeological excavation as well over surfaces of the moon and other planets. This working model is achieved with the help of microcontrollers and sensors. Many issues that rose during the process were tackled efficiently. And some problems which were difficult to cater to are mentioned in the future scope below.

XI. FUTURE SCOPE

i. Introduction of dynamic movements in the rover: In the given time constraints, the code developed was mostly hard-coded as in the rover made less decisions itself and most of them were written by us. Moving forward the rover can be developed to take decisions dynamically regarding rerouting and detection of obstacles.
ii. Reducing weight of the base to make the movement of the rover faster: As mentioned earlier, the rover's weight was more than the capacity of the motors. Design can be modified to reduce the weight, for better performance and the wheel can be removed in this condition.
iii. Making it more power efficient so that the batteries can be used: Currently, power is provided by a power supply, moving forward efficient batteries can be used to provide sufficient current for all the motors to work simultaneously.
iv. Providing a better grip to the contact surface of the Arduino legs for smooth movement: The legs can be designed to increase the friction further and to assist a smoother movement.

XII. USES

The Four-Legged Rover can be used on surfaces of moon and other such celestial body as well as in archaeological excavation. It can move individual legs to specific spots and can also tilt their bodies. Hence, they work better on inclined planes than wheeled rovers. The same advantage helps the rover to recover its position when toppled over. The symmetrical design enables the rover to move in all directions equally efficiently and thus makes rerouting easy. The ultrasound sensors help to detect the obstacles to signal the rover to change its directions.

XIII. REFERENCES

[1] *HCSR04 Ultrasonic sensor and STM32 »*. (2022, October 17). ControllersTech. Retrieved December 7, 2022, from https://controllerstech.com/hcsr04-ultrasonic-sensor-and-stm32/

[2] *Input Capture in STM32 »*. (2022, October 19). ControllersTech. Retrieved December 7, 2022, from https://controllerstech.com/input-capture-in-stm32/

[3] Lesics. (2022, January 28). *Four Legged Rovers | The future of space rover technolgy* [Video]. YouTube. https://www.youtube.com/watch?v=3WHGTAYmy1o