INTERFACING OF SERVO MOTOR USING BEAGLEBONE BLACK

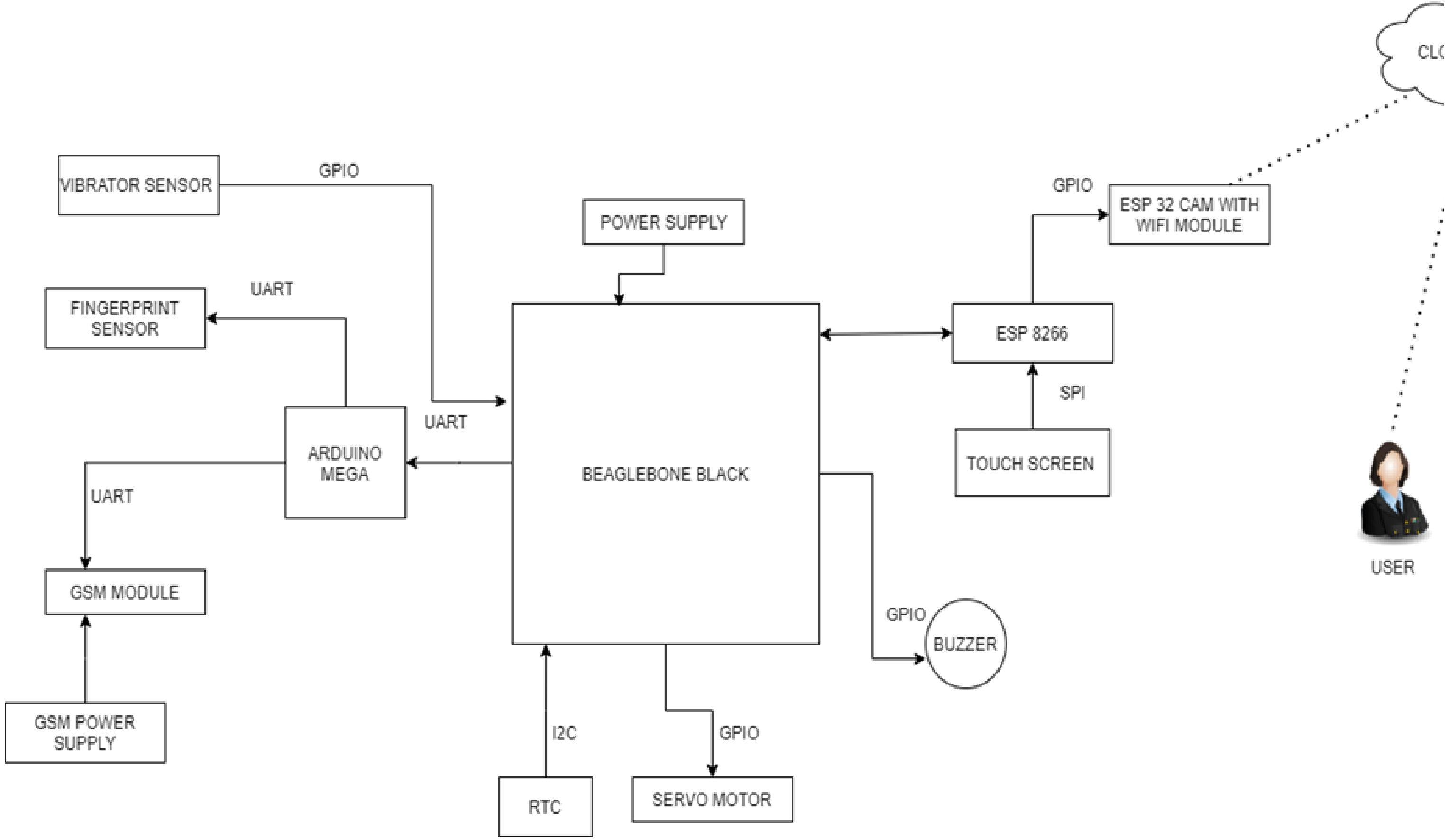
GURPREET SINGH

GROUP #2

CONTENT

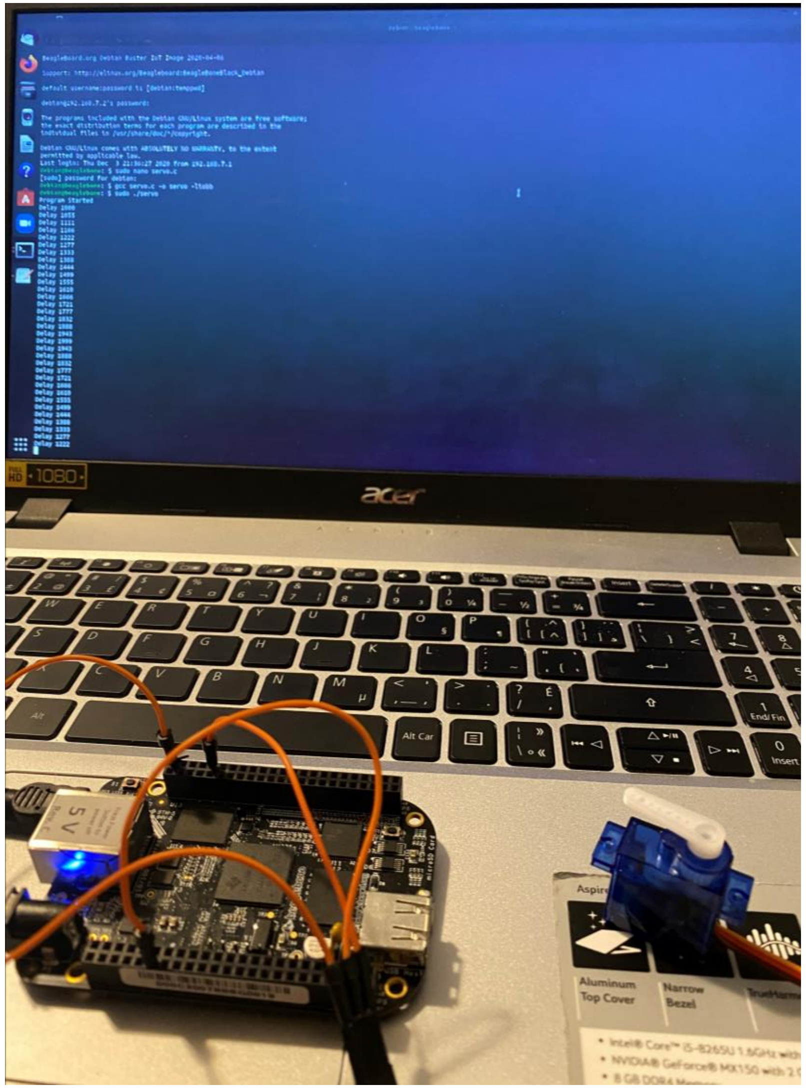
- PROJECT OVERVIEW
- REQUIREMENTS FOR INTERFACING
  - Hardware Requirements
  - Software Requirements
- ROLE OF BEAGLEBONE
  - Pin Layout
  - Features
- INTRODUCTION TO SERVO MOTOR
  - Features
  - Function
- INTERFACING OF SERVO MOTOR
  - Schematic layout
  - Connections of servo motor and beaglebone black
  - Libraries include in coding
  - Coding of Servo motor
- TERMINAL OUTPUT
- REFERENCES

PROJECT OVERVIEW: IOT BASED BANK LOCKER SECURITY



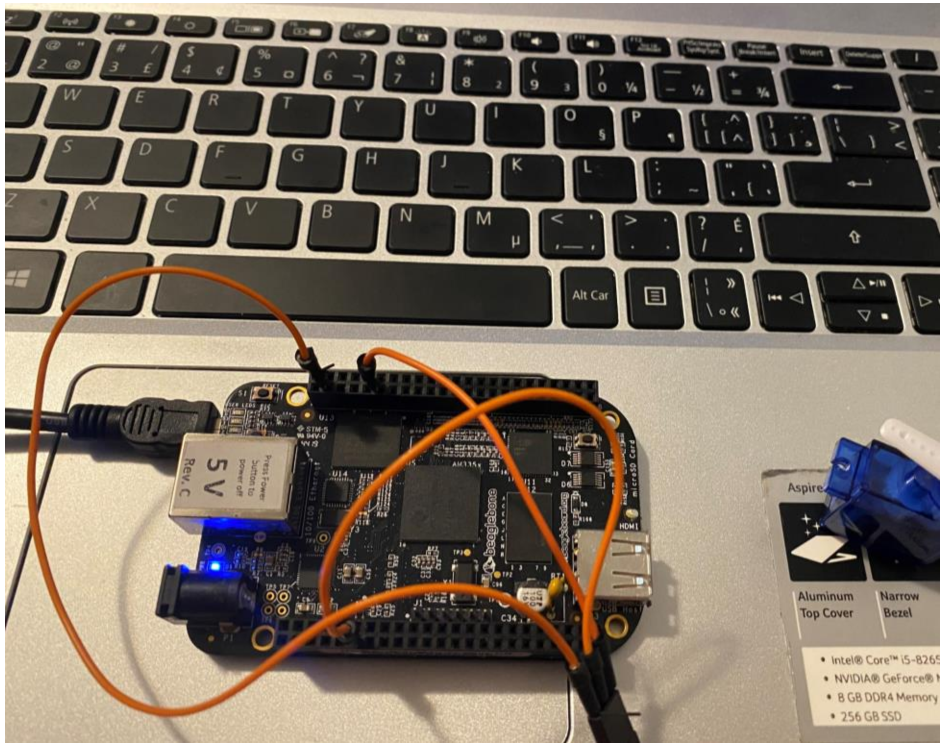
PROJECT OVERVIEW

- The main objective of this project is to effectively control and manage the bank locker security using fingerprint sensor, passcode and camera.
- The IOT based Bank locker security system uses an automated Safety vault with layered defense mechanism.



```
BeagleBoard.org Debian Buster IoT Image 2020-04-06

Support: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian

default username:password is [debian:temppwd]

debian@192.168.7.2's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Dec  3 21:30:27 2020 from 192.168.7.1
debian@beaglebone:~$ sudo nano servo.c
[sudo] password for debian:
debian@beaglebone:~$ gcc servo.c -o servo -liobb
debian@beaglebone:~$ sudo ./servo
Program Started
Delay 1000
Delay 1055
Delay 1111
Delay 1166
Delay 1222
Delay 1277
Delay 1333
Delay 1388
Delay 1444
Delay 1499
Delay 1555
Delay 1610
Delay 1666
Delay 1721
Delay 1777
Delay 1832
Delay 1888
Delay 1943
Delay 1999
Delay 1943
Delay 1888
Delay 1832
Delay 1777
Delay 1721
Delay 1666
Delay 1610
Delay 1555
Delay 1499
Delay 1444
Delay 1388
Delay 1333
Delay 1277
Delay 1222
```

REQUIREMENTS



- After interfacing of buzzer, interfacing of servo motor with beaglebone black is the next task.
- For sevo motor interfacing, there are some hardware and software requirements.

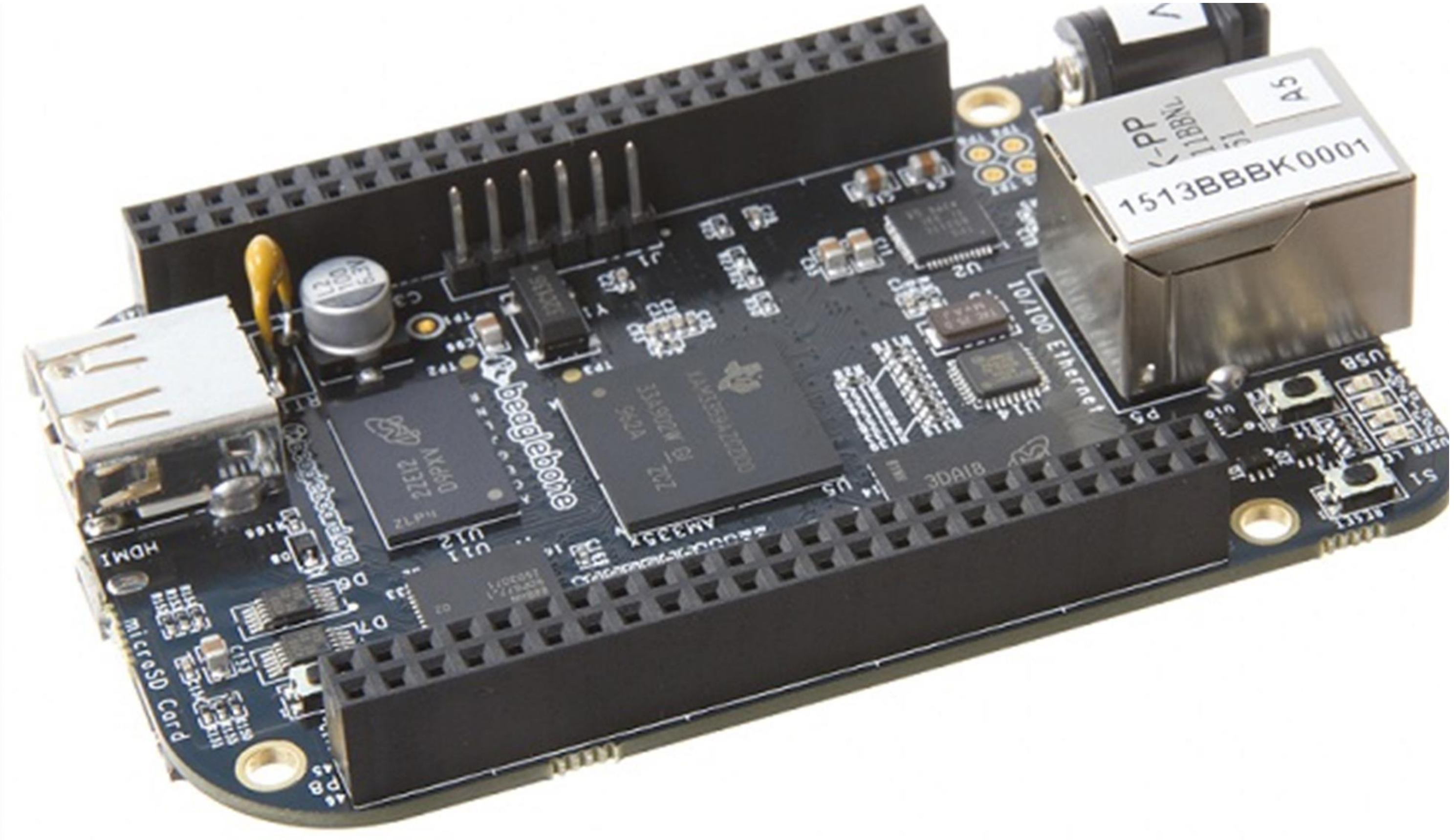
HARDWARE REQUIREMENTS

- Beaglebone black
- Servo Motor
- Jumper wires
- USB Cable

SOFTWARE REQUIREMENTS

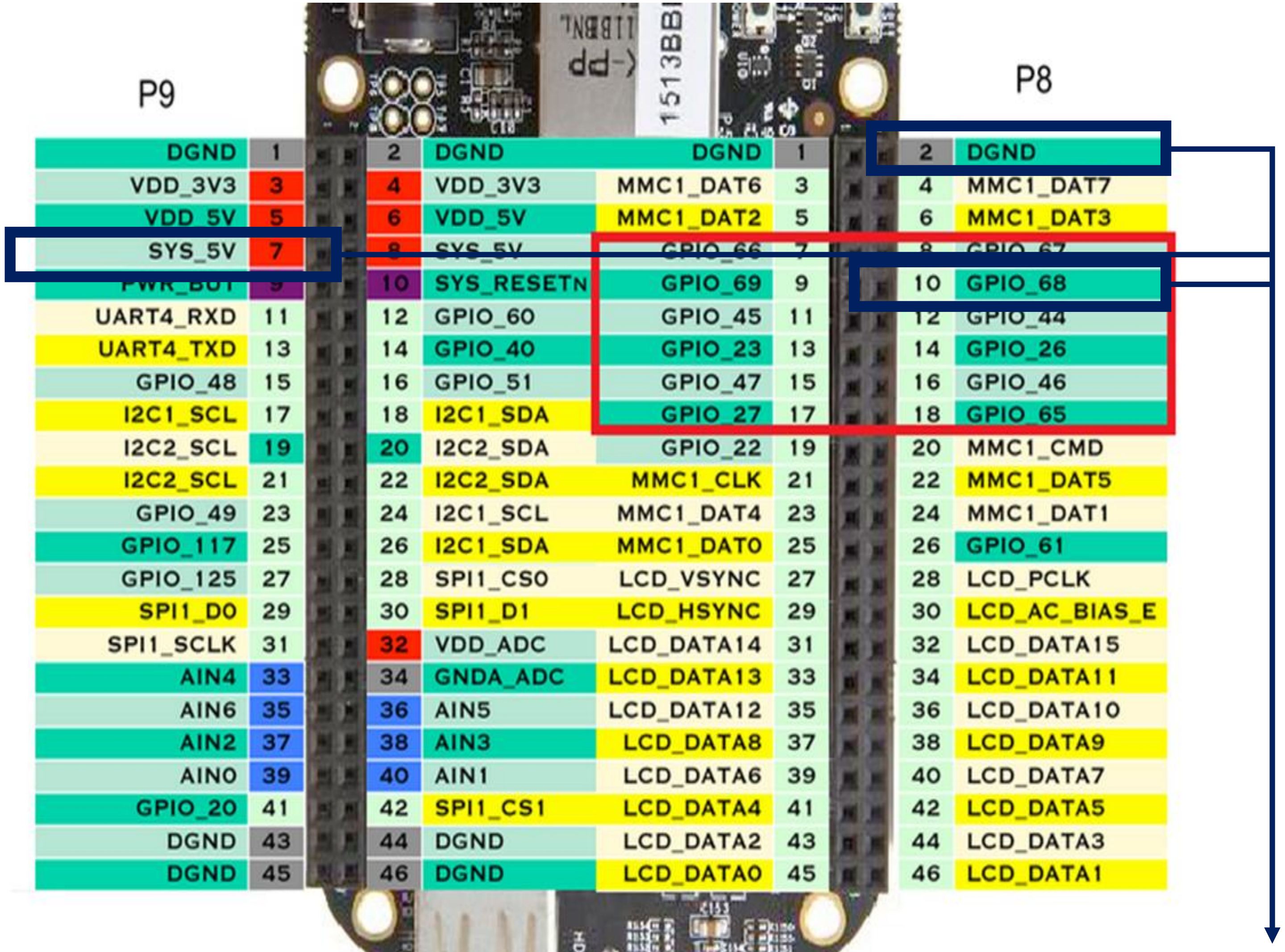
- Terminal
- GCC compiler
- GNU Nano editor

BEAGLEBONE BLACK



- Beaglebone Black is our main microcontroller unit as all the components are connected directly or indirectly to beaglebone black. Therefore, servo motor is connented to the digital pins of beaglebone directly.
- Also, it is a low-cost, community-supported development platform.

PIN LAYOUT OF BEAGLEBONE BLACK

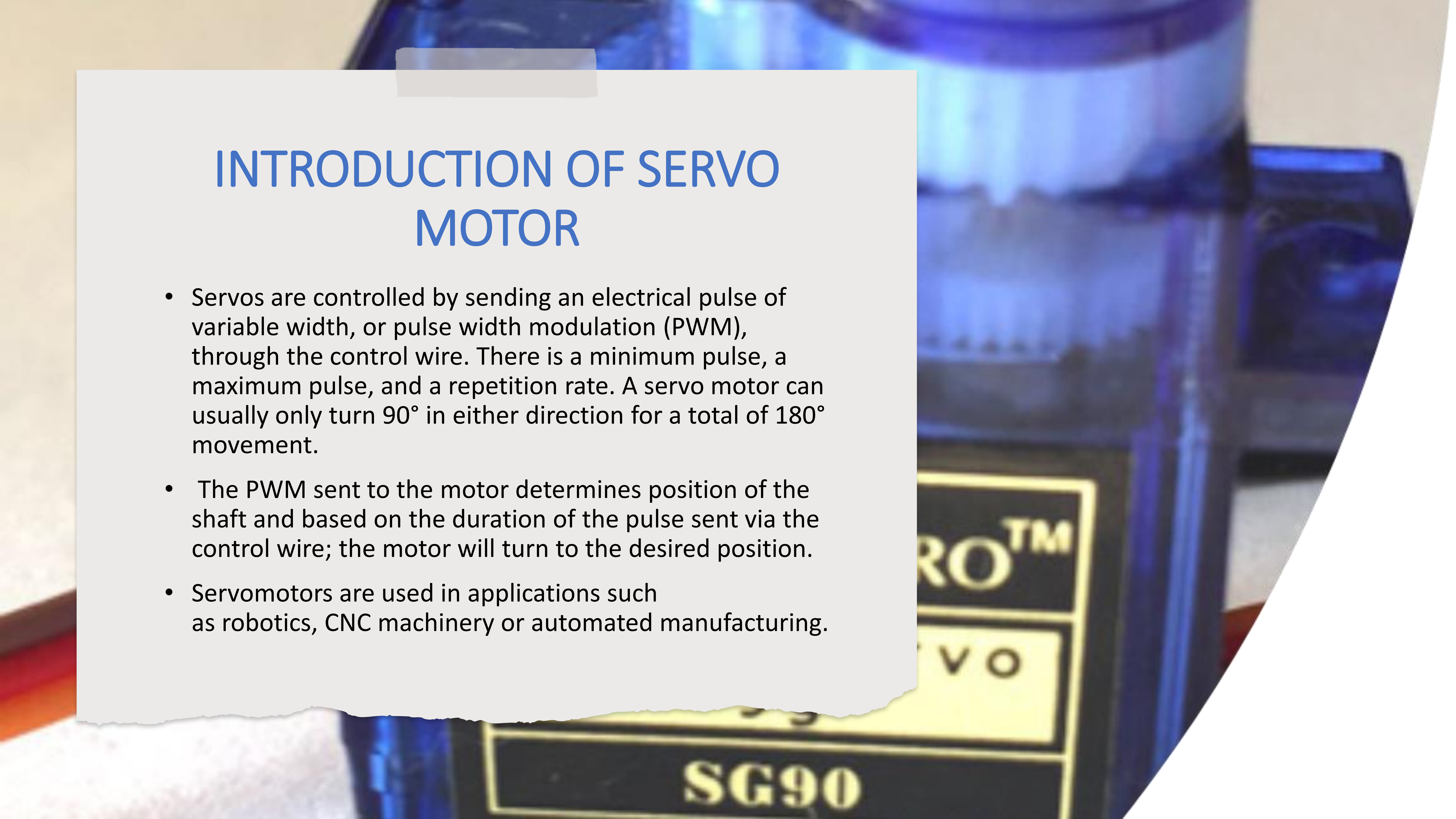


P9				P8			
DGND	1	2	DGND	DGND	1	2	DGND
VDD_3V3	3	4	VDD_3V3	MMC1_DAT6	3	4	MMC1_DAT7
VDD_5V	5	6	VDD_5V	MMC1_DAT2	5	6	MMC1_DAT3
SYS_5V	7	8	SYS_5V	GPIO_66	7	8	GPIO_67
PWR_BUT	9	10	SYS_RESETN	GPIO_69	9	10	GPIO_68
UART4_RXD	11	12	GPIO_60	GPIO_45	11	12	GPIO_44
UART4_TXD	13	14	GPIO_40	GPIO_23	13	14	GPIO_26
GPIO_48	15	16	GPIO_51	GPIO_47	15	16	GPIO_46
I2C1_SCL	17	18	I2C1_SDA	GPIO_27	17	18	GPIO_65
I2C2_SCL	19	20	I2C2_SDA	GPIO_22	19	20	MMC1_CMD
I2C2_SCL	21	22	I2C2_SDA	MMC1_CLK	21	22	MMC1_DAT5
GPIO_49	23	24	I2C1_SCL	MMC1_DAT4	23	24	MMC1_DAT1
GPIO_117	25	26	I2C1_SDA	MMC1_DAT0	25	26	GPIO_61
GPIO_125	27	28	SPI1_CS0	LCD_VSYNC	27	28	LCD_PCLK
SPI1_D0	29	30	SPI1_D1	LCD_HSYNC	29	30	LCD_AC_BIAS_E
SPI1_SCLK	31	32	VDD_ADC	LCD_DATA14	31	32	LCD_DATA15
AIN4	33	34	GNDA_ADC	LCD_DATA13	33	34	LCD_DATA11
AIN6	35	36	AIN5	LCD_DATA12	35	36	LCD_DATA10
AIN2	37	38	AIN3	LCD_DATA8	37	38	LCD_DATA9
AIN0	39	40	AIN1	LCD_DATA6	39	40	LCD_DATA7
GPIO_20	41	42	SPI1_CS1	LCD_DATA4	41	42	LCD_DATA5
DGND	43	44	DGND	LCD_DATA2	43	44	LCD_DATA3
DGND	45	46	DGND	LCD_DATA0	45	46	LCD_DATA1

PINS USED FOR INTERFACING

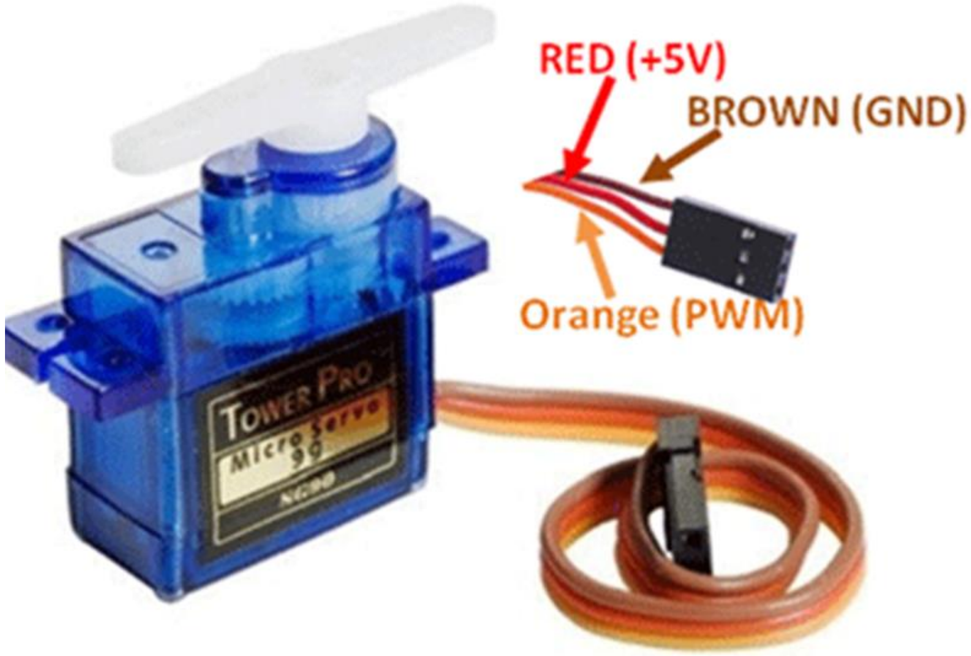
INTRODUCTION OF SERVO MOTOR

- Servos are controlled by sending an electrical pulse of variable width, or pulse width modulation (PWM), through the control wire. There is a minimum pulse, a maximum pulse, and a repetition rate. A servo motor can usually only turn 90° in either direction for a total of 180° movement.
- The PWM sent to the motor determines position of the shaft and based on the duration of the pulse sent via the control wire; the motor will turn to the desired position.
- Servomotors are used in applications such as robotics, CNC machinery or automated manufacturing.

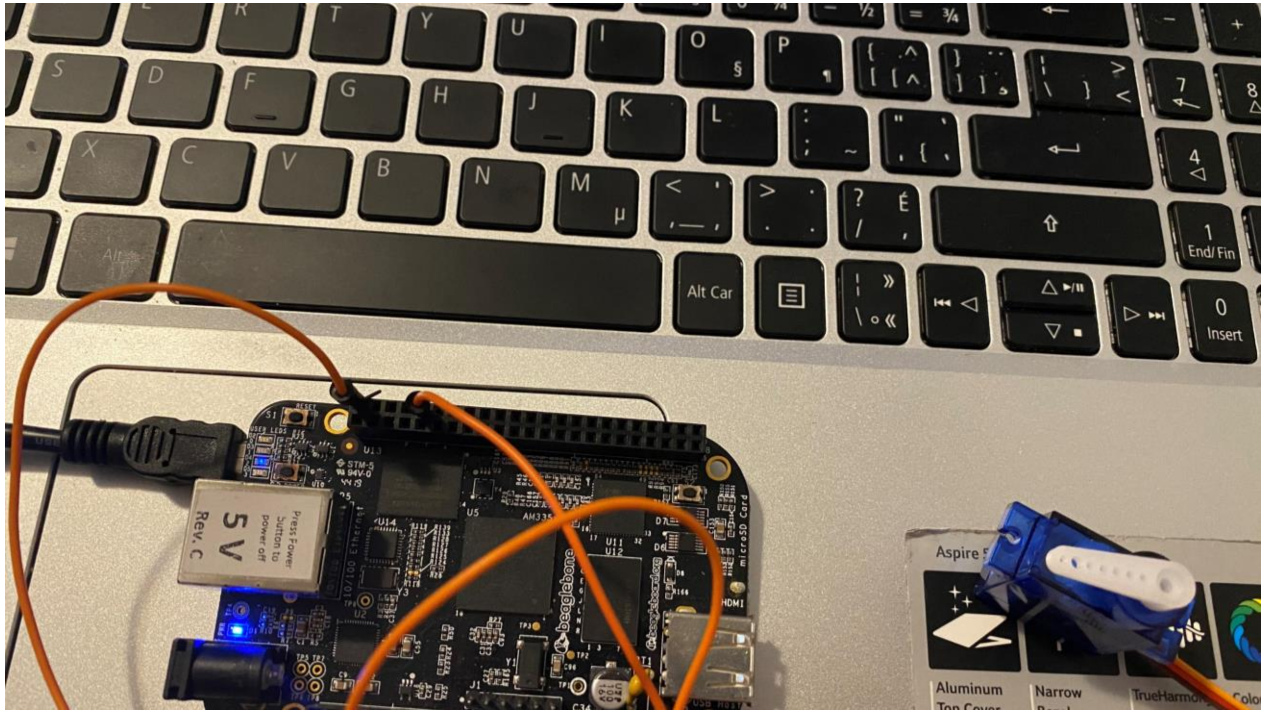


FEATURES OF SERVO MOTOR

- Operating Voltage is +5V typically
- Torque: 2.5kg/cm
- Operating speed is 0.1s/60°
- Gear Type: Plastic
- Rotation : 0°-180°
- Weight of motor : 9gm



FUNCTION OF SERVO MOTOR



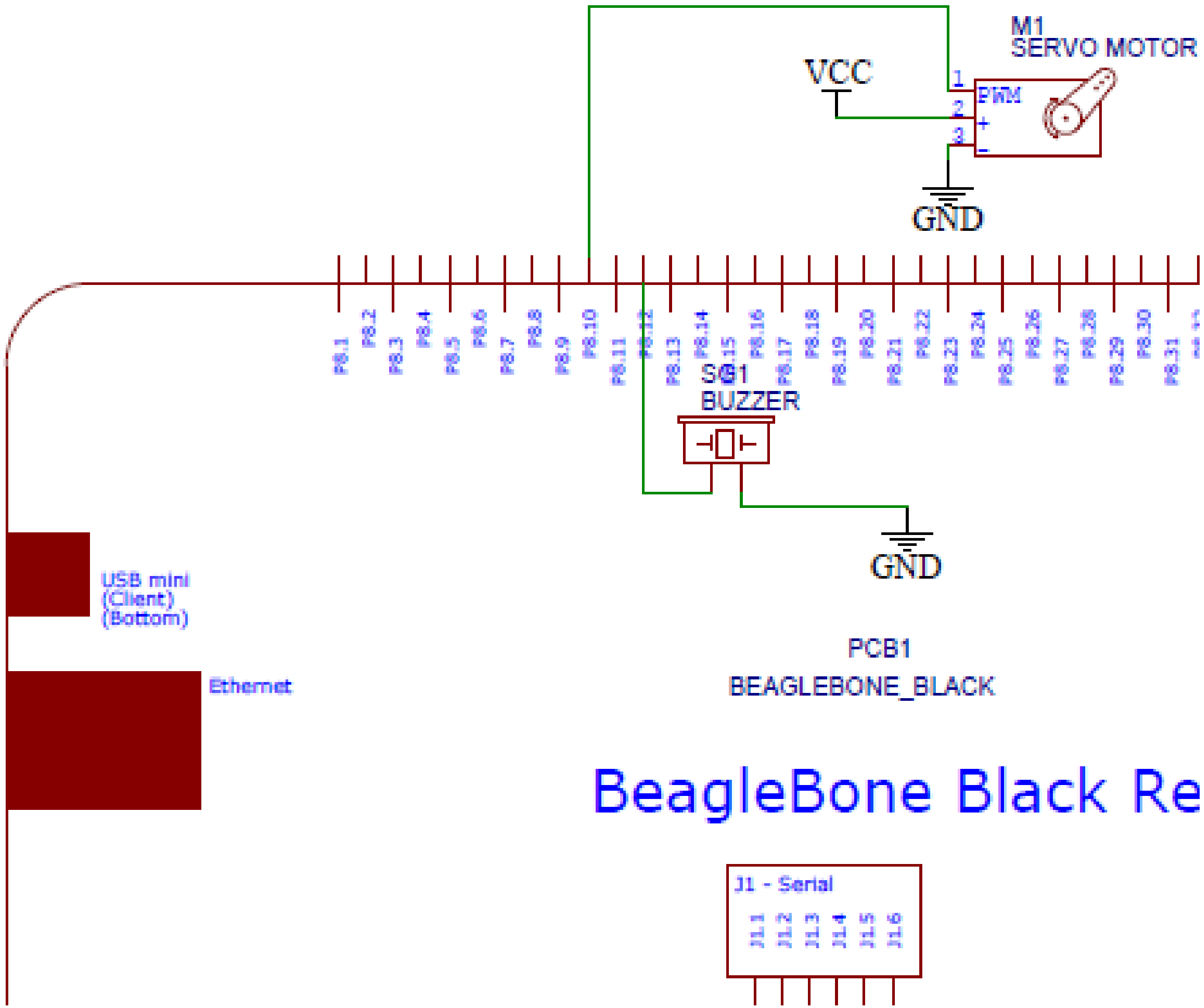
- It is used in the project for lock open and close mechanism.

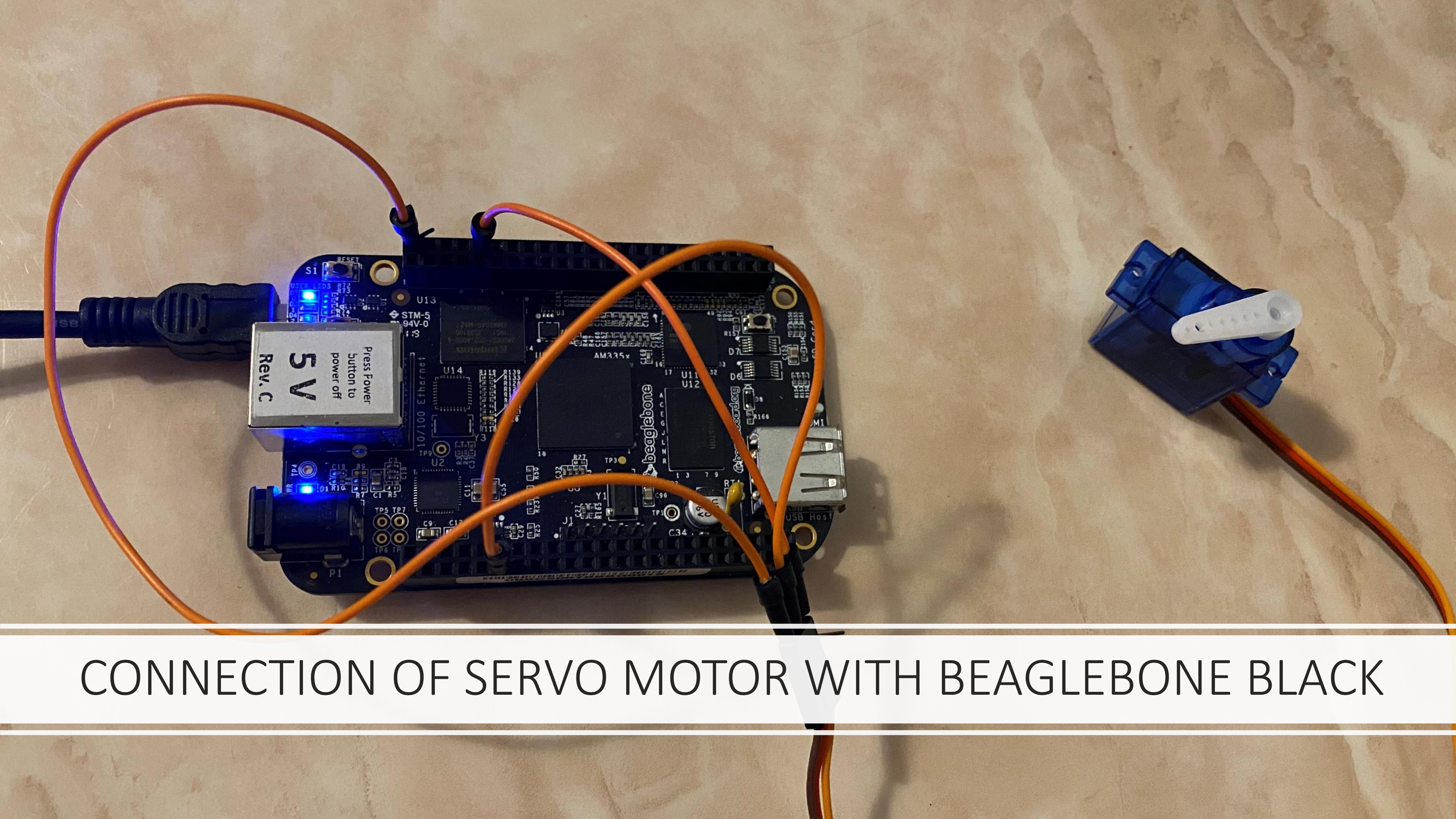
INTERFACING OF BUZZER TO BEAGLEBONE BLACK

Servo motor	Beaglebone Black
GND(BROWN)	P8.2
5V (RED)	P9.7
PWM(BROWN)	P8.10

- Servo motor is connected directly to the beaglebone black.
- Connections are made with the help of GPIO pins.

SCHEMATIC DIAGRAM





CONNECTION OF SERVO MOTOR WITH BEAGLEBONE BLACK

LIBRARIES USED FOR INTERFACING

- IOBB LIBRARY:
  - As the servo motor is directly connected to GPIO pins of beaglebone black we need IOBB library.
  - We already installed that library during the interfacing of buzzer.
- STDIO LIBRARY:
  - Standard C input output library.
- TIME LIBRARY:
  - The Time library provides the data structures and functions required to retrieve the system time, perform time calculations, and output formatted strings that allow the time to be displayed in a variety of common formats.

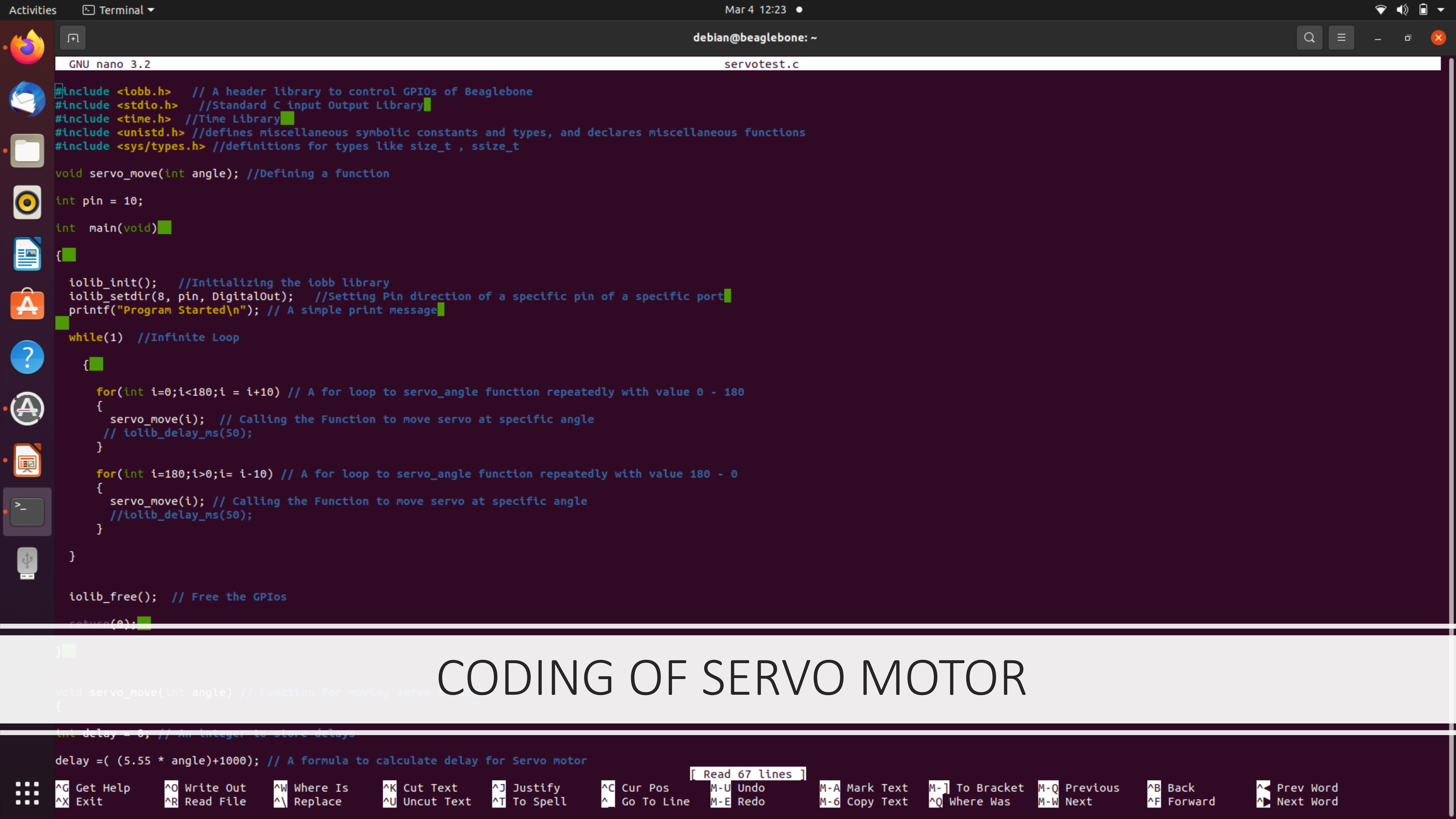
LIBRARIES USED FOR INTERFACING

unistd.h: standard symbolic constants and types

- The <unistd.h> header defines miscellaneous symbolic constants and types and declares miscellaneous functions.

sys/types.h: data types such as time_t used for time in seconds.

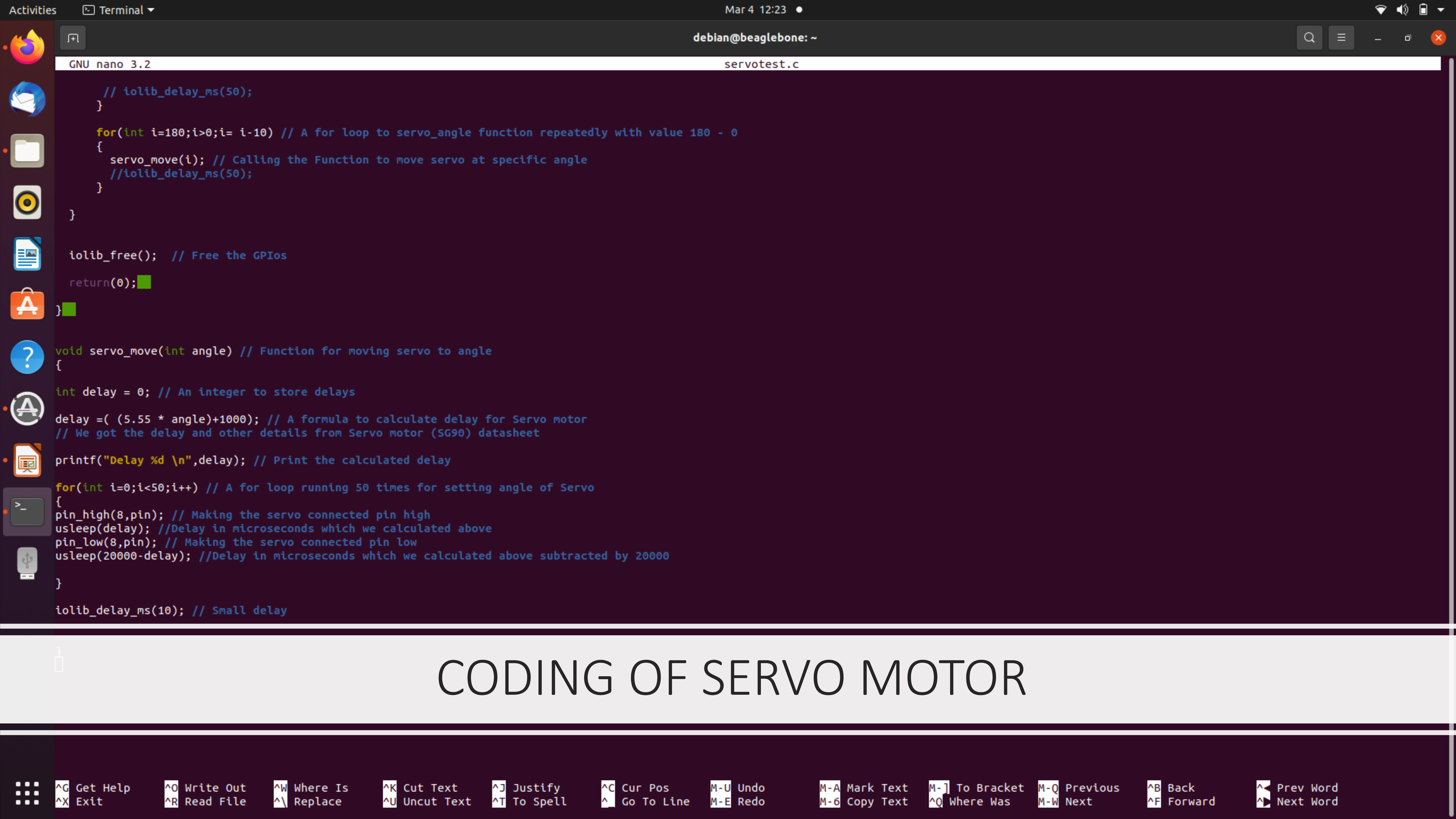
CODING OF SERVO MOTOR



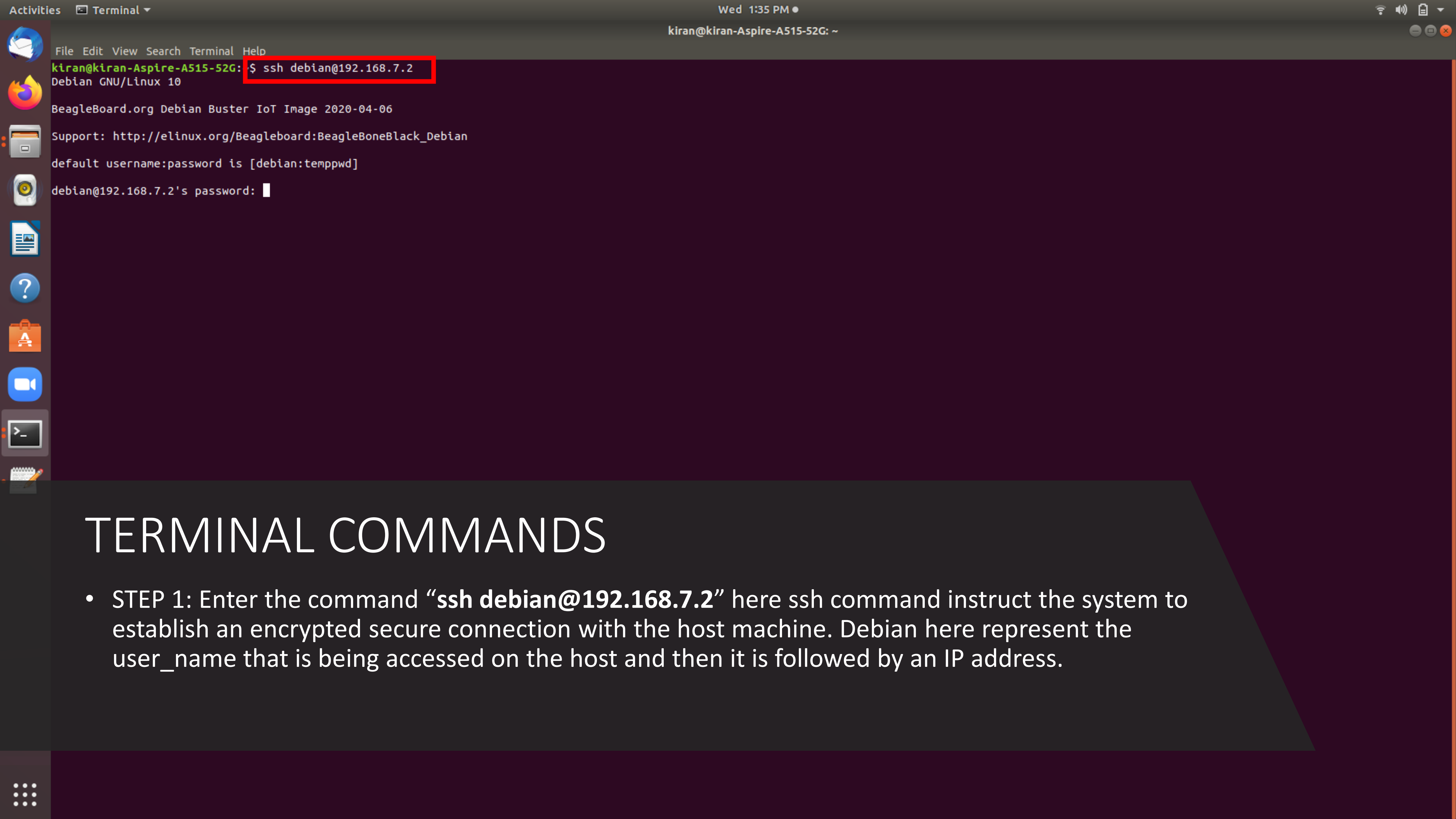
```
GNU nano 3.2                                servotest.c

#include <iobb.h>   // A header library to control GPIOs of Beaglebone
#include <stdio.h>   //Standard C input Output Library
#include <time.h>  //Time Library
#include <unistd.h> //defines miscellaneous symbolic constants and types, and declares miscellaneous functions
#include <sys/types.h> //definitions for types like size_t , ssize_t

void servo_move(int angle); //Defining a function

int pin = 10;

int  main(void)

{

  iolib_init();   //Initializing the iobb library
  iolib_setdir(8, pin, DigitalOut);   //Setting Pin direction of a specific pin of a specific port
  printf("Program Started\n"); // A simple print message

  while(1)  //Infinite Loop

    {

      for(int i=0;i<180;i = i+10) // A for loop to servo_angle function repeatedly with value 0 - 180
      {
        servo_move(i);  // Calling the Function to move servo at specific angle
       // iolib_delay_ms(50);
      }

      for(int i=180;i>0;i= i-10) // A for loop to servo_angle function repeatedly with value 180 - 0
      {
        servo_move(i); // Calling the Function to move servo at specific angle
        //iolib_delay_ms(50);
      }

  }


  iolib_free();  // Free the GPIOs

  return(0);

}


void servo_move(int angle) // Function for moving servo
{

int delay = 0; // An integer to store delays

delay =( (5.55 * angle)+1000); // A formula to calculate delay for Servo motor
```

[Read 67 lines]

CODING OF SERVO MOTOR



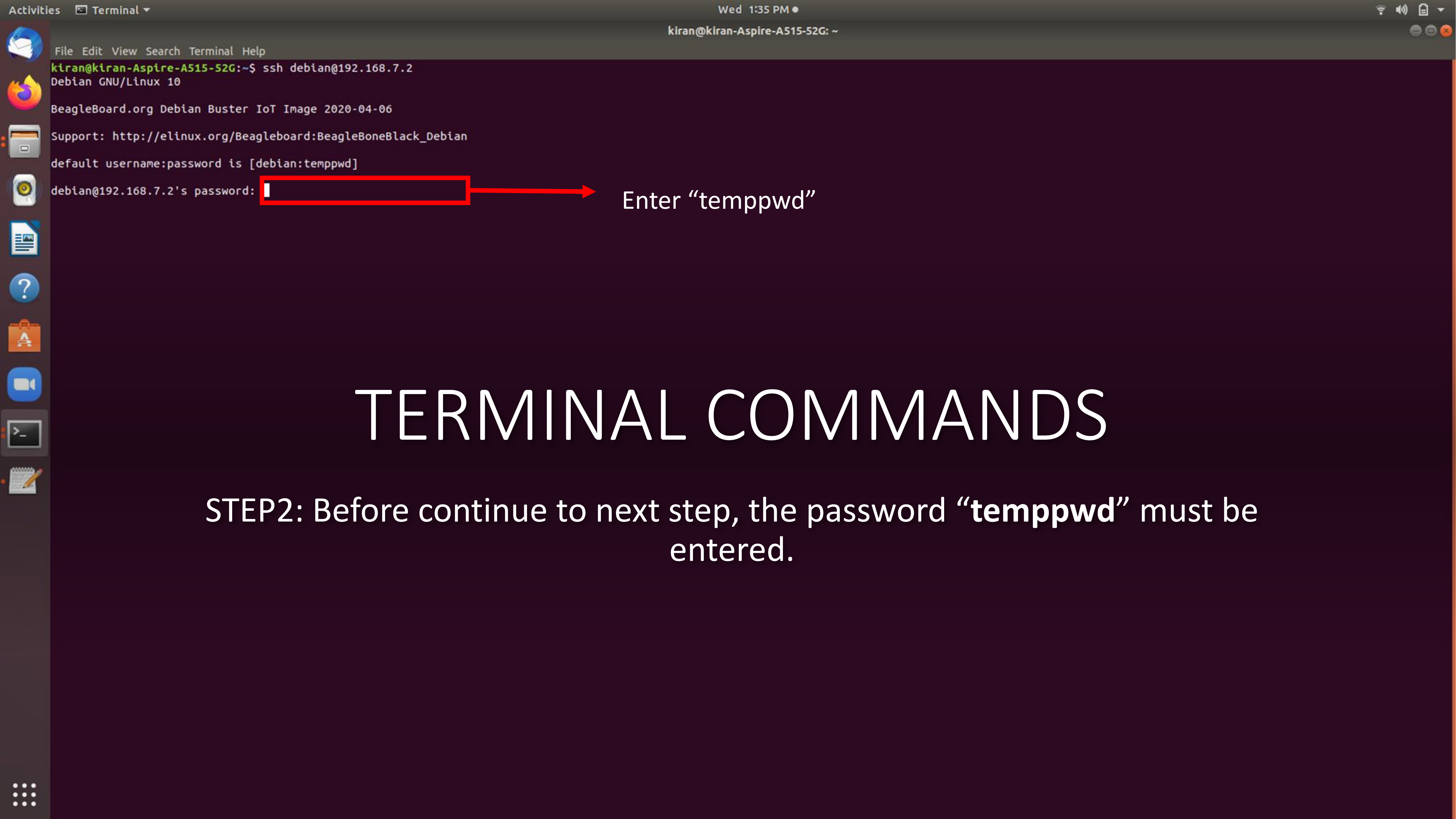
```
       // iolib_delay_ms(50);
      }

      for(int i=180;i>0;i= i-10) // A for loop to servo_angle function repeatedly with value 180 - 0
      {
        servo_move(i); // Calling the Function to move servo at specific angle
        //iolib_delay_ms(50);
      }

  }


  iolib_free();  // Free the GPIos

  return(0);

}


void servo_move(int angle) // Function for moving servo to angle
{

int delay = 0; // An integer to store delays

delay =( (5.55 * angle)+1000); // A formula to calculate delay for Servo motor
// We got the delay and other details from Servo motor (SG90) datasheet

printf("Delay %d \n",delay); // Print the calculated delay

for(int i=0;i<50;i++) // A for loop running 50 times for setting angle of Servo
{
pin_high(8,pin); // Making the servo connected pin high
usleep(delay); //Delay in microseconds which we calculated above
pin_low(8,pin); // Making the servo connected pin low
usleep(20000-delay); //Delay in microseconds which we calculated above subtracted by 20000

}

iolib_delay_ms(10); // Small delay

}
```


Activities
Terminal
Wed 1:35 PM
kiran@kiran-Aspire-A515-52G: ~
File Edit View Search Terminal Help

```
kiran@kiran-Aspire-A515-52G:~$ ssh debian@192.168.7.2
Debian GNU/Linux 10

BeagleBoard.org Debian Buster IoT Image 2020-04-06

Support: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian

default username:password is [debian:temppwd]

debian@192.168.7.2's password: 
```

TERMINAL COMMANDS

- STEP 1: Enter the command “**ssh debian@192.168.7.2**” here ssh command instruct the system to establish an encrypted secure connection with the host machine. Debian here represent the user_name that is being accessed on the host and then it is followed by an IP address.



Activities
Terminal
Wed 1:35 PM
kiran@kiran-Aspire-A515-52G: ~
File Edit View Search Terminal Help

```
kiran@kiran-Aspire-A515-52G:~$ ssh debian@192.168.7.2
Debian GNU/Linux 10

BeagleBoard.org Debian Buster IoT Image 2020-04-06

Support: http://elinux.org/Beagleboard:BeagleBoneBlack_Debian

default username:password is [debian:temppwd]

debian@192.168.7.2's password:
```

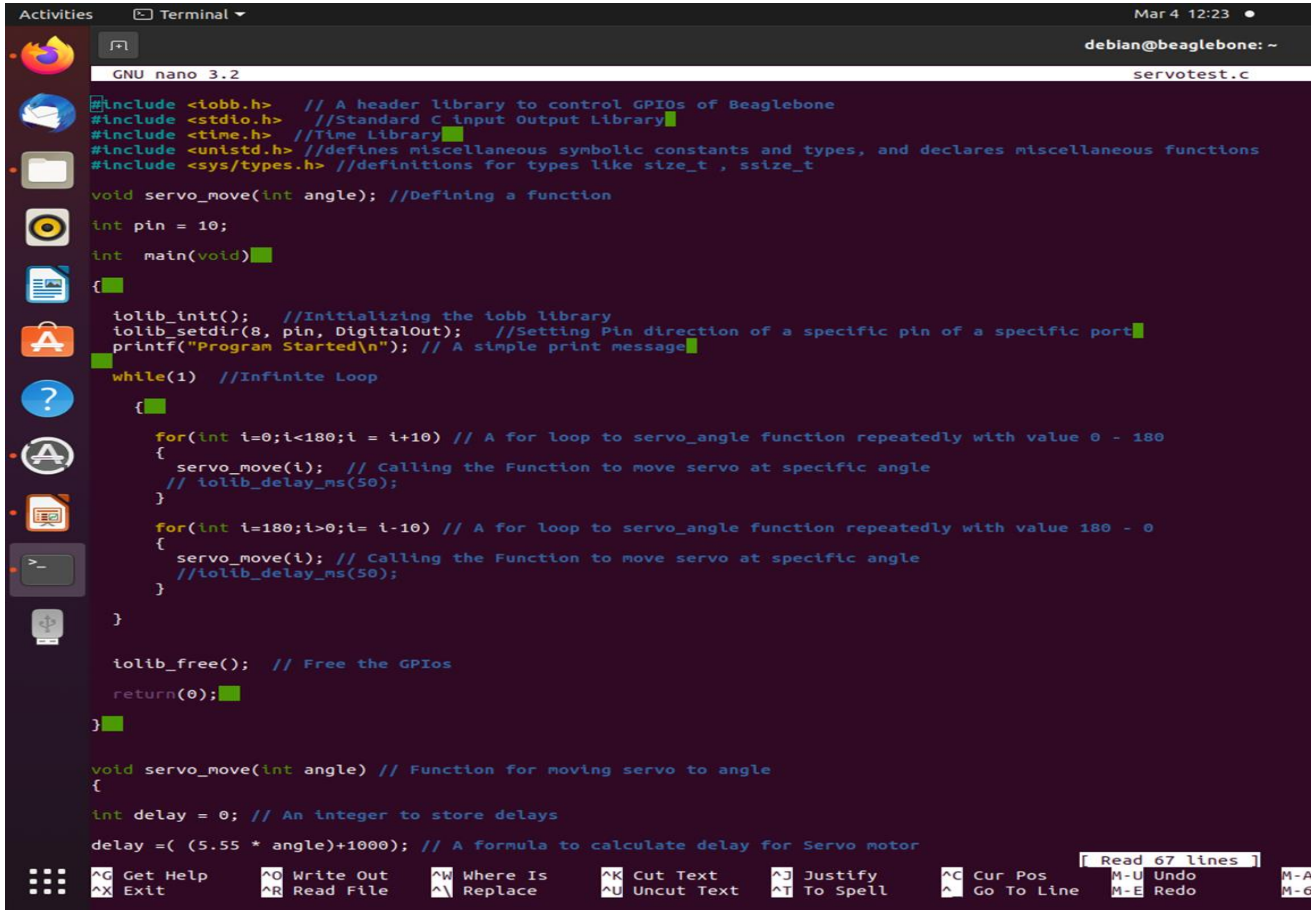
Enter “temppwd”

TERMINAL COMMANDS

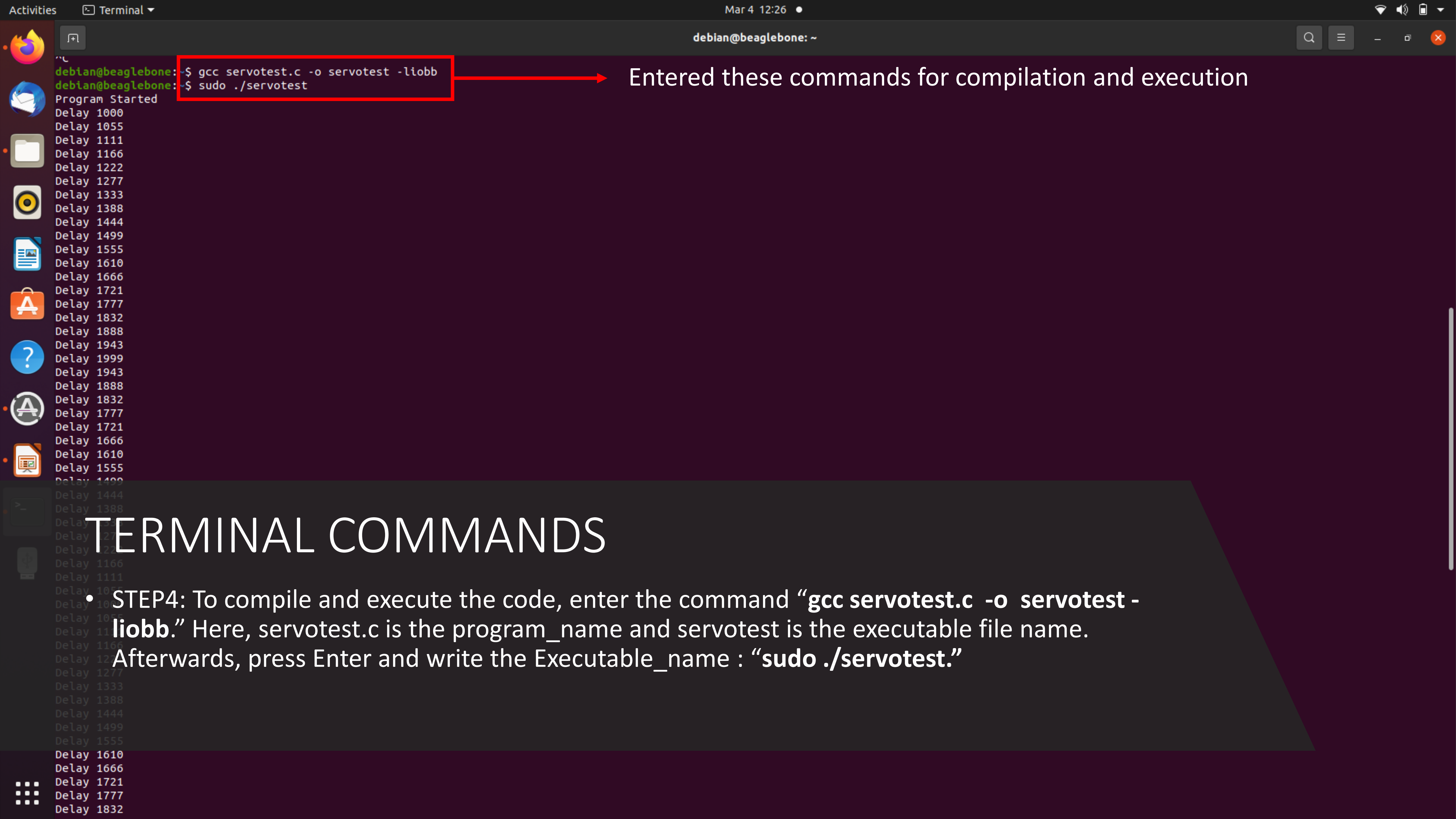
STEP2: Before continue to next step, the password “**temppwd**” must be entered.

TERMINAL COMMANDS

- STEP 3: Enter the command "**sudo nano servotest.c**" to open Nano text editor and to directly write, edit and navigate the code and to get immediate onscreen feedback. Here, servotest.c is the file name. Then enter the same password "**temppwd"** and this window will appear. Here CTRL+O : save the code; then press enter; CTRL+X : to exit.



```
GNU nano 3.2                                                                    servotest.c

#include <iobb.h>    // A header library to control GPIOs of Beaglebone
#include <stdio.h>    //Standard C input Output Library
#include <time.h>  //Time Library
#include <unistd.h> //defines miscellaneous symbolic constants and types, and declares miscellaneous functions
#include <sys/types.h> //definitions for types like size_t , ssize_t

void servo_move(int angle); //Defining a function

int pin = 10;

int  main(void)

{

  iolib_init();   //Initializing the iobb library
  iolib_setdir(8, pin, DigitalOut);    //Setting Pin direction of a specific pin of a specific port
  printf("Program Started\n"); // A simple print message

  while(1)  //Infinite Loop

    {

      for(int i=0;i<180;i = i+10) // A for loop to servo_angle function repeatedly with value 0 - 180
      {
        servo_move(i);  // Calling the Function to move servo at specific angle
       // iolib_delay_ms(50);
      }

      for(int i=180;i>0;i= i-10) // A for loop to servo_angle function repeatedly with value 180 - 0
      {
        servo_move(i); // Calling the Function to move servo at specific angle
        //iolib_delay_ms(50);
      }

  }


  iolib_free();  // Free the GPIos

  return(0);

}

void servo_move(int angle) // Function for moving servo to angle
{

int delay = 0; // An integer to store delays

delay =( (5.55 * angle)+1000); // A formula to calculate delay for Servo motor
                                                                          [ Read 67 lines ]
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos      M-U Undo        M-A
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell     ^_ Go To Line   M-E Redo        M-6
```

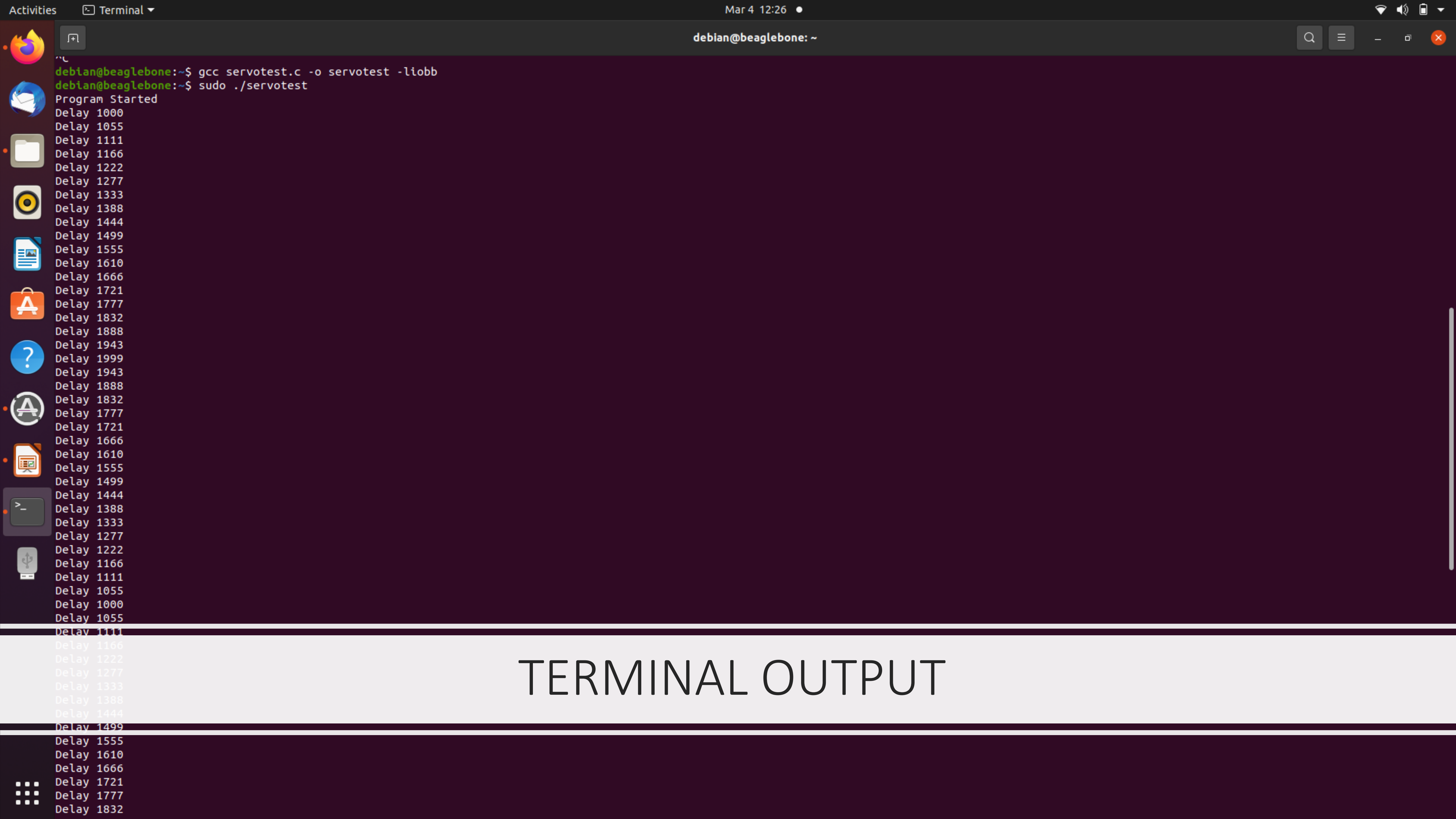


```
^C
debian@beaglebone:~$ gcc servotest.c -o servotest -liobb
debian@beaglebone:~$ sudo ./servotest
Program Started
Delay 1000
Delay 1055
Delay 1111
Delay 1166
Delay 1222
Delay 1277
Delay 1333
Delay 1388
Delay 1444
Delay 1499
Delay 1555
Delay 1610
Delay 1666
Delay 1721
Delay 1777
Delay 1832
Delay 1888
Delay 1943
Delay 1999
Delay 1943
Delay 1888
Delay 1832
Delay 1777
Delay 1721
Delay 1666
Delay 1610
Delay 1555
Delay 1499
Delay 1444
Delay 1388
Delay 1333
Delay 1277
Delay 1222
Delay 1166
Delay 1111
Delay 1055
Delay 1000
Delay 1055
Delay 1111
Delay 1166
Delay 1222
Delay 1277
Delay 1333
Delay 1388
Delay 1444
Delay 1499
Delay 1555
Delay 1610
Delay 1666
Delay 1721
Delay 1777
Delay 1832
```

Entered these commands for compilation and execution

TERMINAL COMMANDS

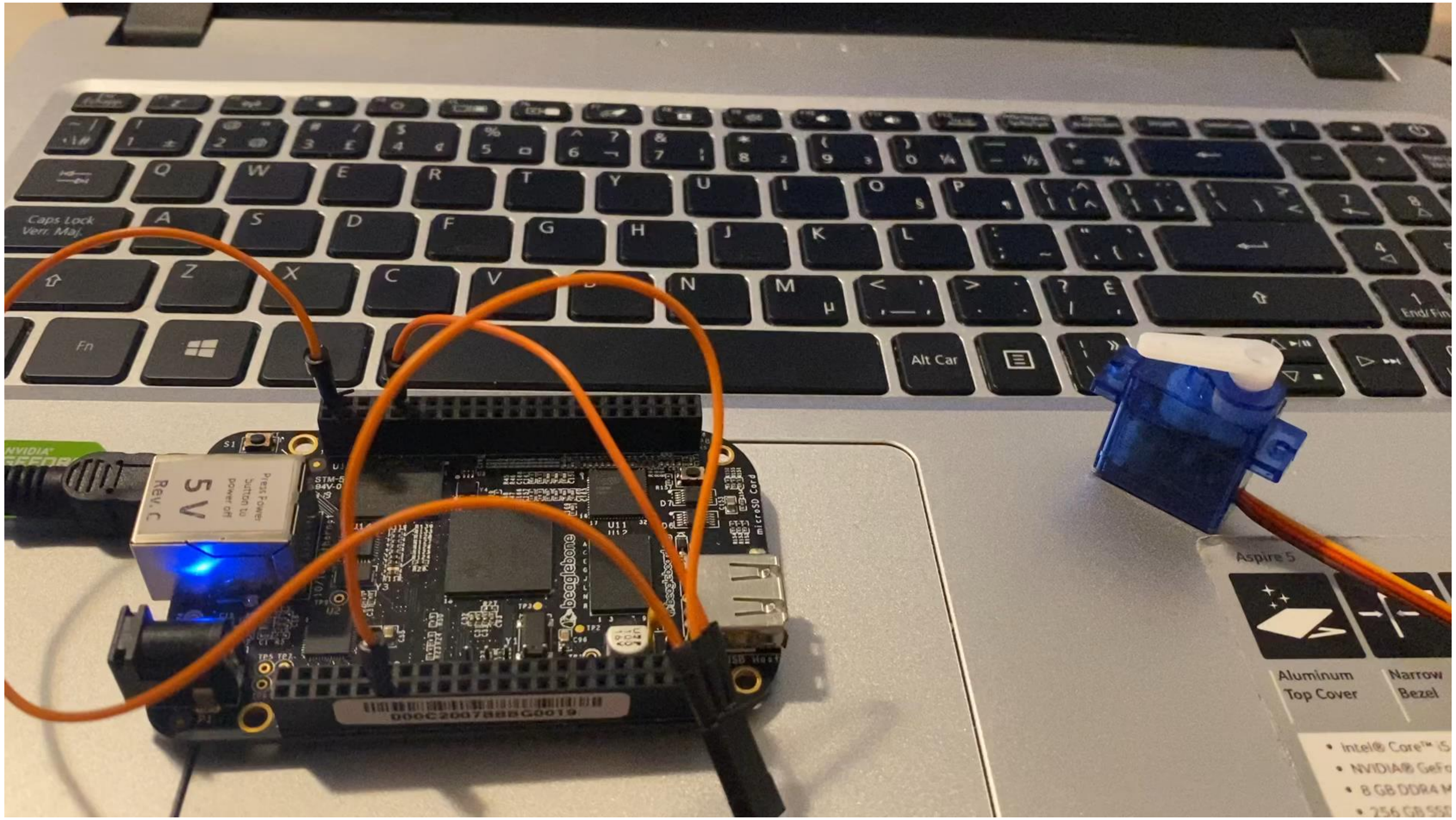
- STEP4: To compile and execute the code, enter the command “**gcc servotest.c -o servotest -liobb**.” Here, servotest.c is the program_name and servotest is the executable file name. Afterwards, press Enter and write the Executable_name : “**sudo ./servotest.”**



```
^C
debian@beaglebone:~$ gcc servotest.c -o servotest -liobb
debian@beaglebone:~$ sudo ./servotest
Program Started
Delay 1000
Delay 1055
Delay 1111
Delay 1166
Delay 1222
Delay 1277
Delay 1333
Delay 1388
Delay 1444
Delay 1499
Delay 1555
Delay 1610
Delay 1666
Delay 1721
Delay 1777
Delay 1832
Delay 1888
Delay 1943
Delay 1999
Delay 1943
Delay 1888
Delay 1832
Delay 1777
Delay 1721
Delay 1666
Delay 1610
Delay 1555
Delay 1499
Delay 1444
Delay 1388
Delay 1333
Delay 1277
Delay 1222
Delay 1166
Delay 1111
Delay 1055
Delay 1000
Delay 1055
Delay 1111
Delay 1166
Delay 1222
Delay 1277
Delay 1333
Delay 1388
Delay 1444
Delay 1499
Delay 1555
Delay 1610
Delay 1666
Delay 1721
Delay 1777
Delay 1832
```

TERMINAL OUTPUT

ROTATION OF SERVO MOTOR



REFERENCES

- Manual Control of a Servo on the Arduino for the Zipwhip TextSpresso Machine | Zipwhip
- C Library – <time.h> – The Geek Diary
- https://pubs.opengroup.org/onlinepubs/009696899/basedefs/sys/types.h.html
- https://www.element14.com/community/community/designcenter/single-board-computers/next-genbeaglebone/blog/2013/10/10/bbb--beaglebone-black-io-library-for-c

REFERENCES

- Servo Motors Work | How Servo Motors Work (jameco.com)
- Servo Motor SG-90 Basics, Pinout, Wire Description, Datasheet (components101.com)
- https://pubs.opengroup.org/onlinepubs/7908799/xsh/unistd.h.html

THANK YOU