VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belgavi-590014, Karnataka

EMERGING TECHNOLOGIES

MINI PROJECT REPORT

Under Bachelor of Engineering in Electrical & Electronics

"Create a smart lock which can be unlocked by entering a unique password format designed by your app"

Name of the Candidates:	**AYUSH KUMAR**	**USN:**	**1DS18EE082**
	AYUSH KASHYAP		**1DS17EE024**
	DHAVAL PEERSAB NADAF		**1DS18EE405**
	SRINIVASA BJ		**1DS18EE428**

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
DAYANANDA SAGAR COLLEGE OF ENGINEERING
(Affiliated to VTU, Belagavi - 590018)
Accredited by National Assessment & Accreditation Council (NAAC) with 'A' grade
Shavige Malleshwara Hills, Kumaraswamy Layout, Bengaluru-560078 2020-21

CONTENT PAGE

INTRODUCTION TO PYTHON

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- System scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

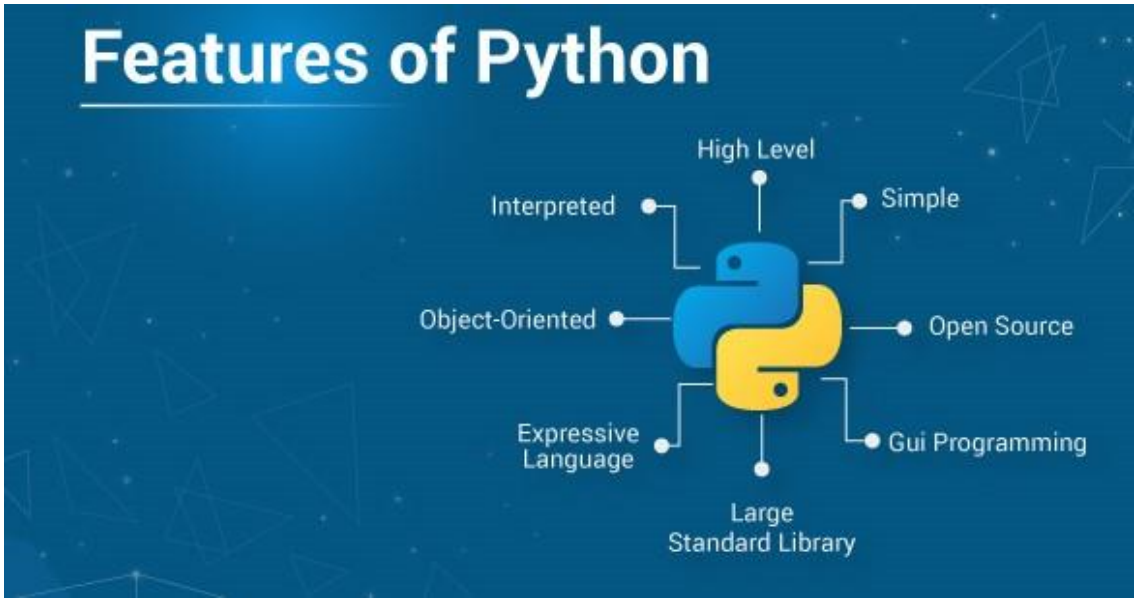
- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.



INTRODUCTION TO LINUX

Linux is a community of open-source UNIX like operating systems that are based on the LINUX KERNEL. It was initially released by **Linus Torvalds** on September 17, 1991. It is a free and open-source operating system and the source code can be modified and distributed to anyone commercially or non-commercially under the GNU General Public License.
Initially, Linux was created for personal computers and gradually it was used in other machines like servers, mainframe computers, supercomputers, etc. Nowadays, Linux is also used in embedded systems like routers, automation controls, televisions, digital video recorders, video game consoles, smartwatches, etc. The biggest success of Linux is Android (operating system) it is based on the Linux kernel that is running on smartphones and tablets. Due to android Linux has the largest installed base of all general-purpose operating systems. Linux is generally packaged in a Linux distribution.

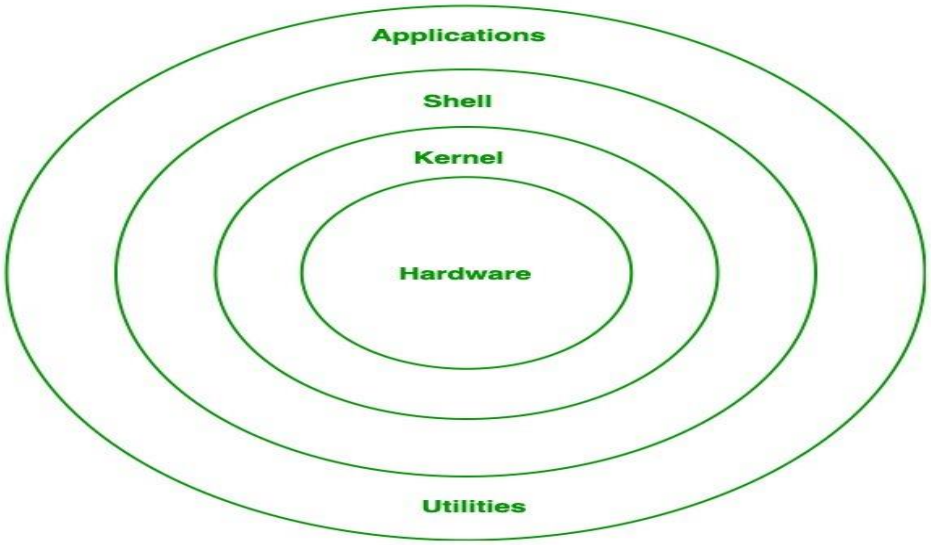
Linux Distribution

Linux distribution is an operating system that is made up of a collection of software based on Linux kernel or you can say distribution contains the Linux kernel and supporting libraries and software. And you can get Linux based operating system by downloading one of the Linux distributions and these distributions are available for different types of devices like embedded devices, personal computers, etc. Around **600 + Linux Distributions** are available and some of the popular Linux distributions are:

- MX Linux
- Manjaro
- Linux Mint
- elementary
- Ubuntu
- Debian
- Solus
- Fedora
- openSUSE
- Deepin

Architecture of Linux

Linux architecture has the following components:



1. **Kernel:** Kernel is the core of the Linux based operating system. It virtualizes the common hardware resources of the computer to provide each process with its virtual resources. This makes the process seem as if it is the sole process running on the machine. The kernel is also responsible for preventing and mitigating conflicts between different processes. Different types of the kernel are:
   - Monolithic Kernel
   - Hybrid kernels
   - Exo kernels
   - Micro kernels
2. **System Library:** it is the special types of functions that are used to implement the functionality of the operating system.
3. **Shell:** It is an interface to the kernel which hides the complexity of the kernel's functions from the users. It takes commands from the user and executes the kernel's functions.
4. **Hardware Layer:** This layer consists all peripheral devices like RAM/ HDD/ CPU etc.
5. **System Utility:** It provides the functionalities of an operating system to the user.

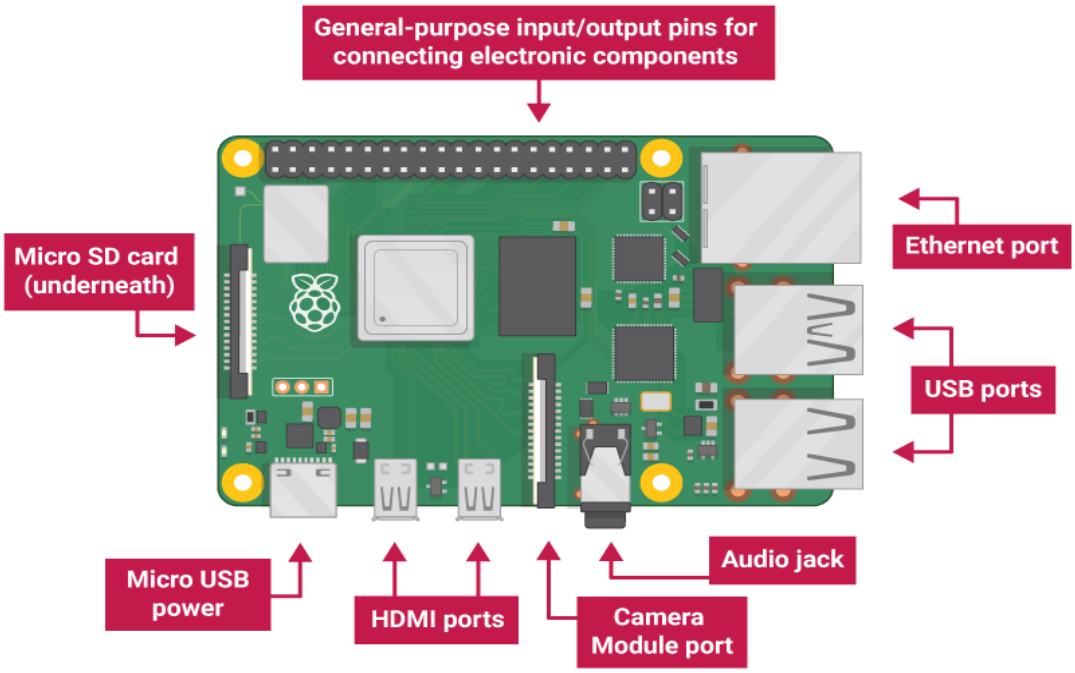
Advantages of Linux

- The main advantage of Linux, is it is an open-source operating system. This means the source code is easily available for everyone and you are allowed to contribute, modify and distribute the code to anyone without any permissions.
- In terms of security, Linux is more secure than any other operating system. It does not mean that Linux is 100 percent secure it has some malware for it but is less vulnerable than any other operating system. So, it does not require any anti-virus software.
- The software updates in Linux are easy and frequent.
- Various Linux distributions are available so that you can use them according to your requirements or according to your taste.
- Linux is freely available to use on the internet.
- It has large community support.
- It provides high stability. It rarely slows down or freezes and there is no need to reboot it after a short time.
- It maintain the privacy of the user.
- The performance of the Linux system is much higher than other operating systems. It allows a large number of people to work at the same time and it handles them efficiently.
- It is network friendly.
- The flexibility of Linux is high. There is no need to install a complete Linux suit; you are allowed to install only required components.
- Linux is compatible with a large number of file formats.
- It is fast and easy to install from the web. It can also install on any hardware even on your old computer system.
- It performs all tasks properly even if it has limited space on the hard disk.

Disadvantages of Linux

- It is not very user-friendly. So, it may be confusing for beginners.
- It has small peripheral hardware drivers as compared to windows.

INTRODUCTION TO RASPBERRY PI

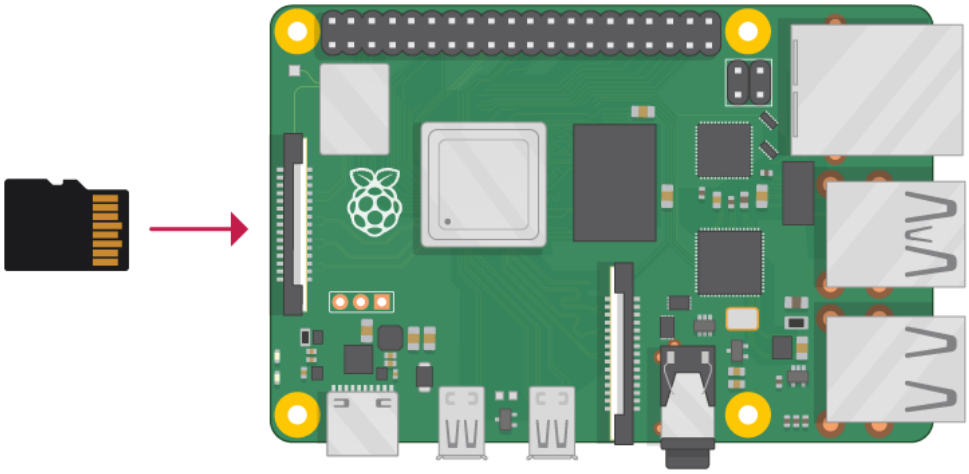


- **USB ports** — these are used to connect a mouse and keyboard. You can also connect other components, such as a USB drive.
- **SD card slot** — you can slot the SD card in here. This is where the operating system software and your files are stored.
- **Ethernet port** — this is used to connect Raspberry Pi to a network with a cable. Raspberry Pi can also connect to a network via wireless LAN.
- **Audio jack** — you can connect headphones or speakers here.
- **HDMI port** — this is where you connect the monitor (or projector) that you are using to display the output from the Raspberry Pi. If your monitor has speakers, you can also use them to hear sound.
- **Micro USB power connector** — this is where you connect a power supply. You should always do this last, after you have connected all your other components.
- **GPIO ports** — these allow you to connect electronic components such as LEDs and buttons to Raspberry Pi

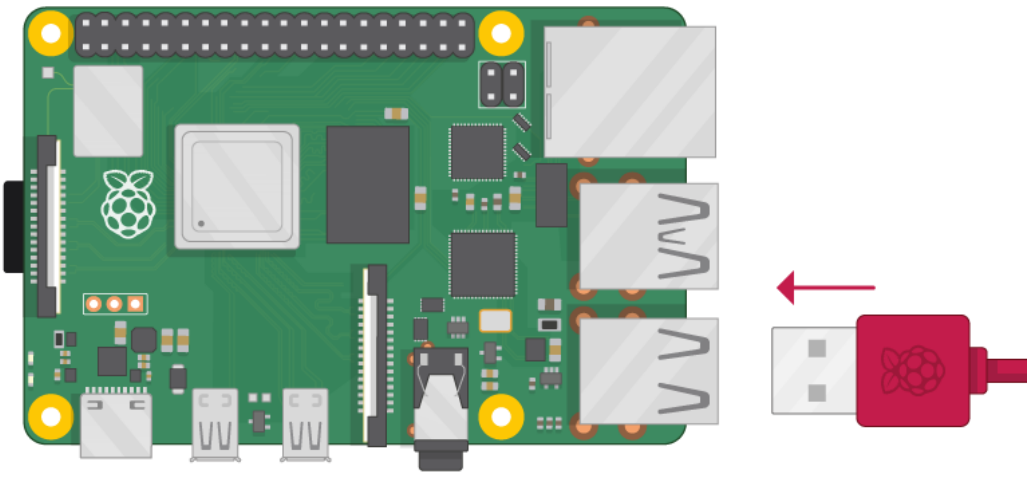
Connect your Raspberry Pi

Let’s connect up your Raspberry Pi and get it running.

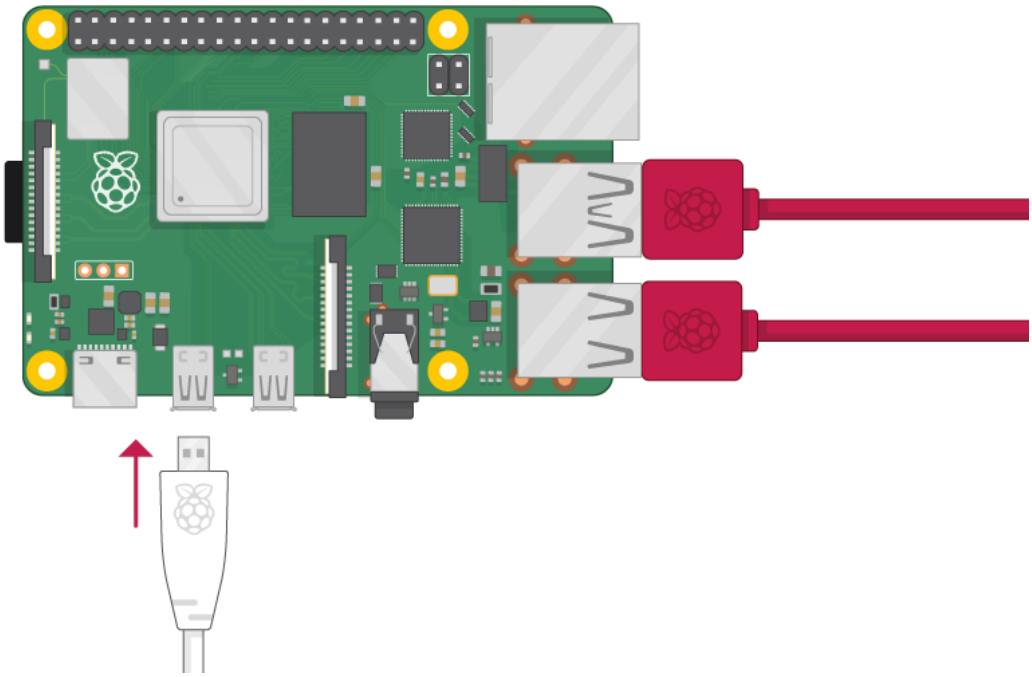
- Check the slot on the underside of your Raspberry Pi to see whether an SD card is inside. If no SD card is there, then insert an SD card with Raspbian installed (via NOOBS).



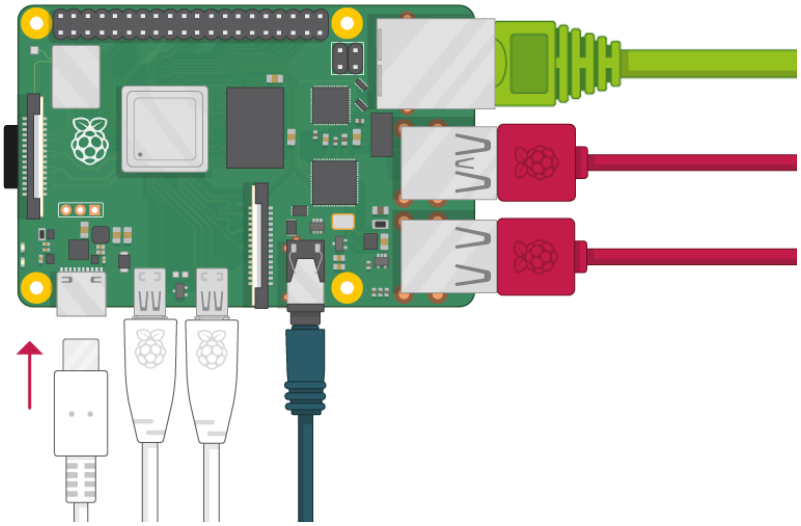
Find the USB connector end of your mouse’s cable, and connect the mouse to a USB port on your Raspberry Pi (it doesn’t matter which port you use). Connect the keyboard in the same way.



Connect your screen to the first of Raspberry Pi 4’s HDMI ports, labelled HDMI0.



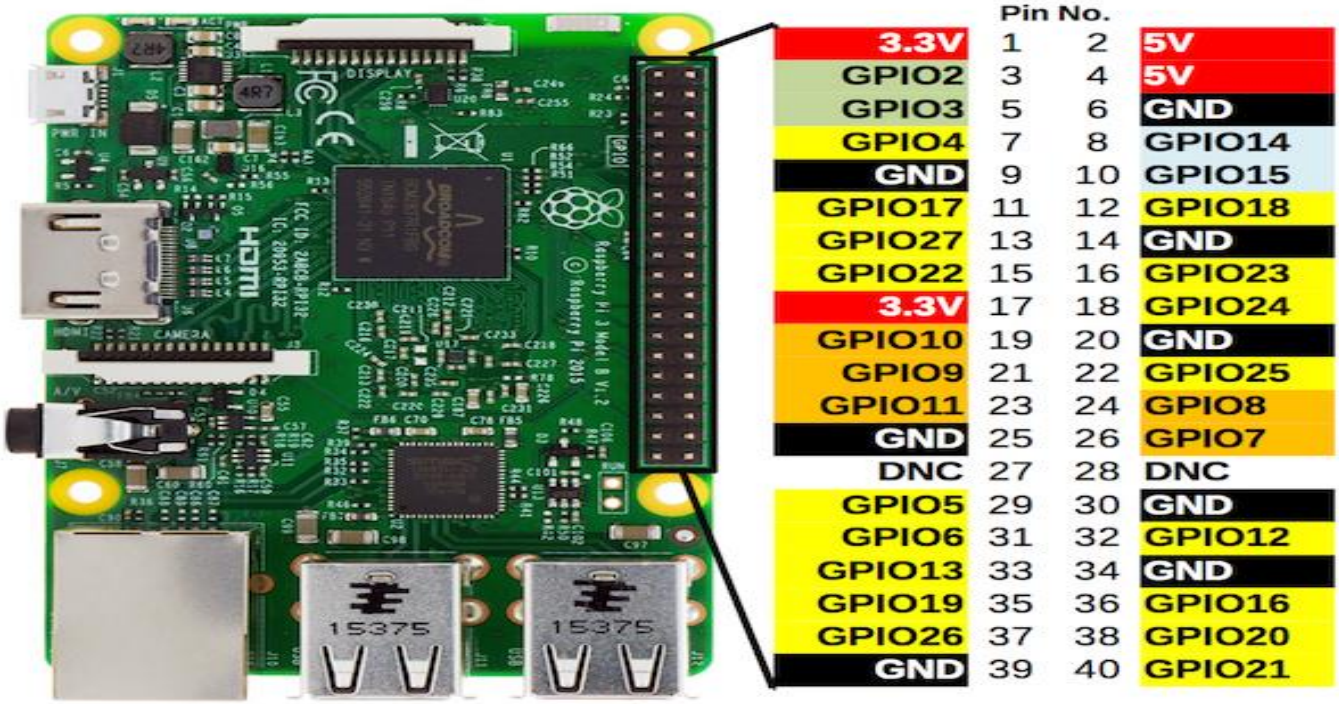
- Plug the power supply into a socket and then connect it to your Raspberry Pi's USB power port.



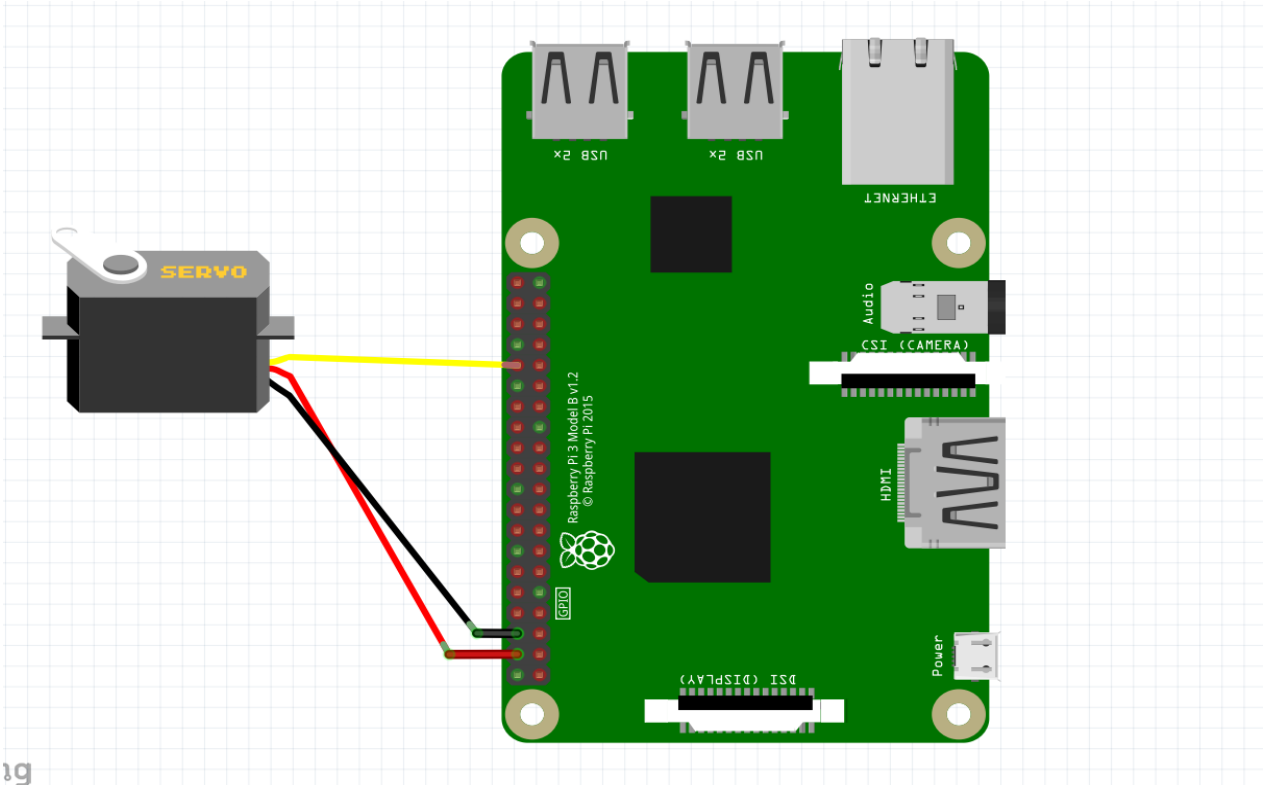
You should see a red light on your Raspberry Pi and raspberries on the monitor.

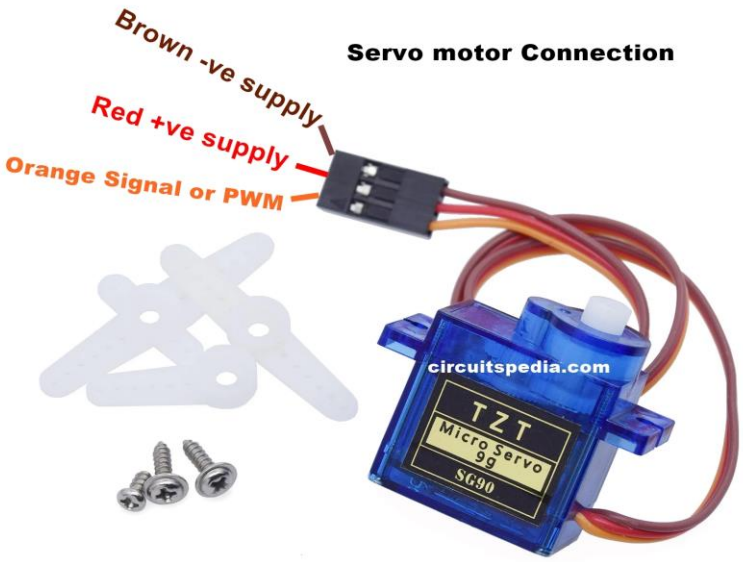
Your Raspberry Pi then boots up into a graphical desktop.

GPIO PINS OF RASPBERRY PI



Circuit diagram of the project:





Code used along with comments and explanations:

```
import RPi.GPIO as gpio          # importing library to use GPIO pins
import time
from tkinter import *            #importing Tkinter for creating GUI
servo=32                         #assignment of signal pin of servo to  pin32 which is a PWM pin


root=Tk()
root.geometry('1280x720')        # assigning dimensions of the window using geometry library
root.title("SMART LOCK")         # title is given
def Lock():                      #function called lock is defined
  print("Lock")
  gpio.setmode(gpio.BOARD)
  gpio.setwarnings(False)
  gpio.setup(servo,gpio.OUT)     # pin 32 is set as output pin.
  pwm=gpio.PWM(servo,50)         # 50 hz of frequency is assigned to servo
  pwm.start(0)
  gpio.output(servo,True)        #condition true makes it rotate   by 90 degrees
  pwm.ChangeDutyCycle(7)
  time.sleep(1)                  # delay
  gpio.output(servo,False)       # if condition not met the servo does not rotate
  pwm.ChangeDutyCycle(0)
  pwm.stop()
  gpio.cleanup()

def Unlock():                   # function called unlock is defined
  global wrong

  unlock['state']=DISABLED      # state of unlock button is disabled till the following code is executed
  Pass=inp.get()                # creating a variable pass to get input
  if Pass == "pi":              # password to unlock the door is created
    gpio.setmode(gpio.BOARD)
    gpio.setwarnings(False)
    gpio.setup(servo,gpio.OUT)       # pin 32 is set as output pin.
    pwm=gpio.PWM(servo,50)           # 50hz of frequency is assigned to servo
    pwm.start(0)                     # start PWM
    gpio.output(servo,True)          # If true, servo rotates to 90 degrees
    pwm.ChangeDutyCycle(2)
    time.sleep(1)                    # delay created
    gpio.output(servo,False)         # if condition not met the servo does not rotate

    pwm.ChangeDutyCycle(2)
    pwm.stop()                       #stop PWM
    gpio.cleanup()
                                     #clears the port 32 and makes it ready to receive next instruction
    succ = Label(root,text="Unlocked")
    succ.pack(padx=1,pady=3)
    unlock['state']=NORMAL
# after password is entered, the unlock button is bought to normal state #so that on the press of the but
ton the door is unlocked
  else:
```

```
    wrong = Label(root,text="Wrong Passcode try again")
# it prints the text if the password entered  is wrong
    wrong.pack(padx=1,pady=3)
    time.sleep(2)

    unlock['state']=NORMAL
lock= Button(root,text="Lock", command=Lock, padx=30,pady=30,width=30,bg='violet')
# button is a widget in tkinter which has paramters like command, #dimensions, background color, text col
or etc.
lock.pack(padx=1,pady=2)
inp=Entry(root,width=20)
# entry is a another widget in Tkinter that creates an input entry field for the user
inp.pack(padx=0,pady=1)
inp.insert(0,"")

unlock= Button(root,text="Unlock",command=Unlock,padx=20,
pady=30,width=30,bg="green")
# unlock button is created with its parametres defined
unlock.pack(padx=1,pady=1)
root.mainloop()
# this code keeps looping
```

We have used tkinter for creating our GUI for the password entering and controlling of the lock:

Tkinter Programming:

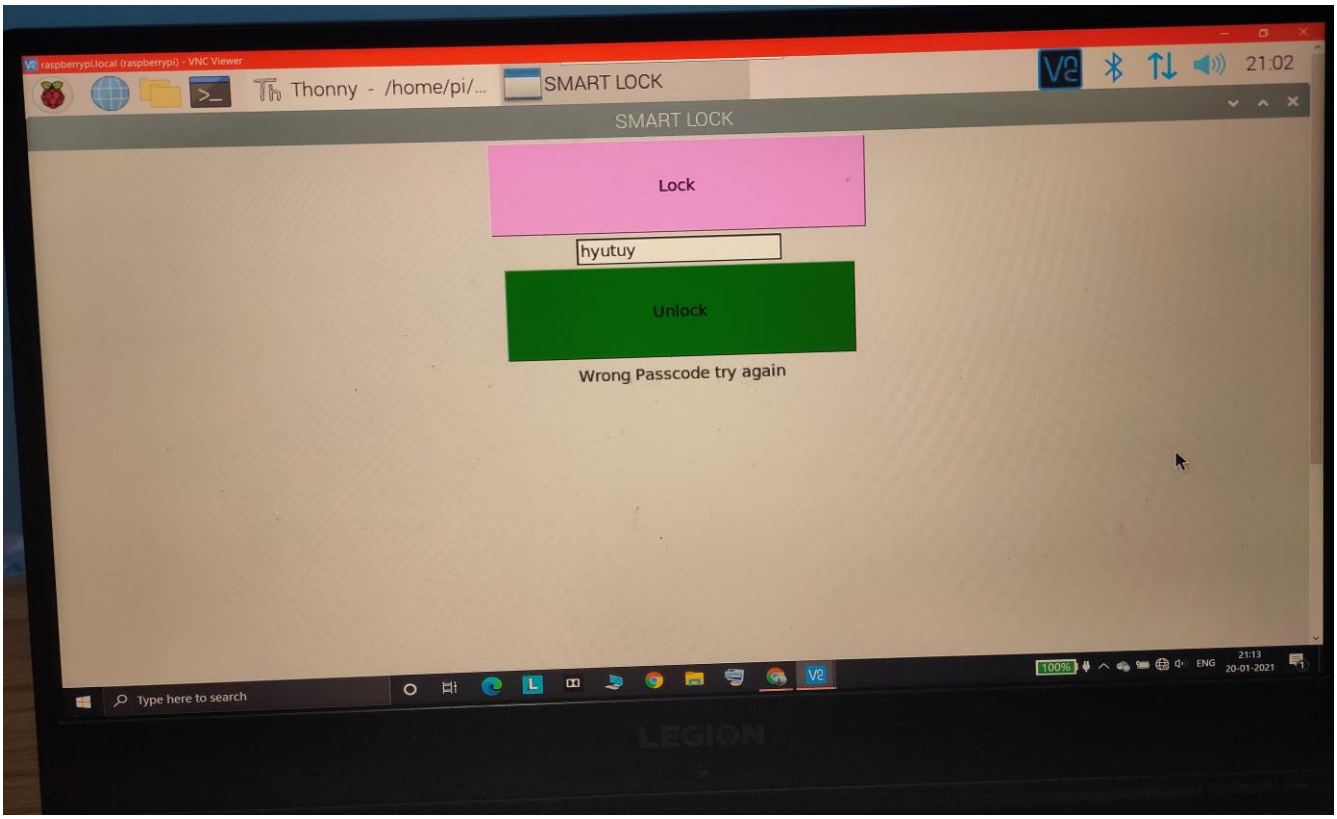
Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

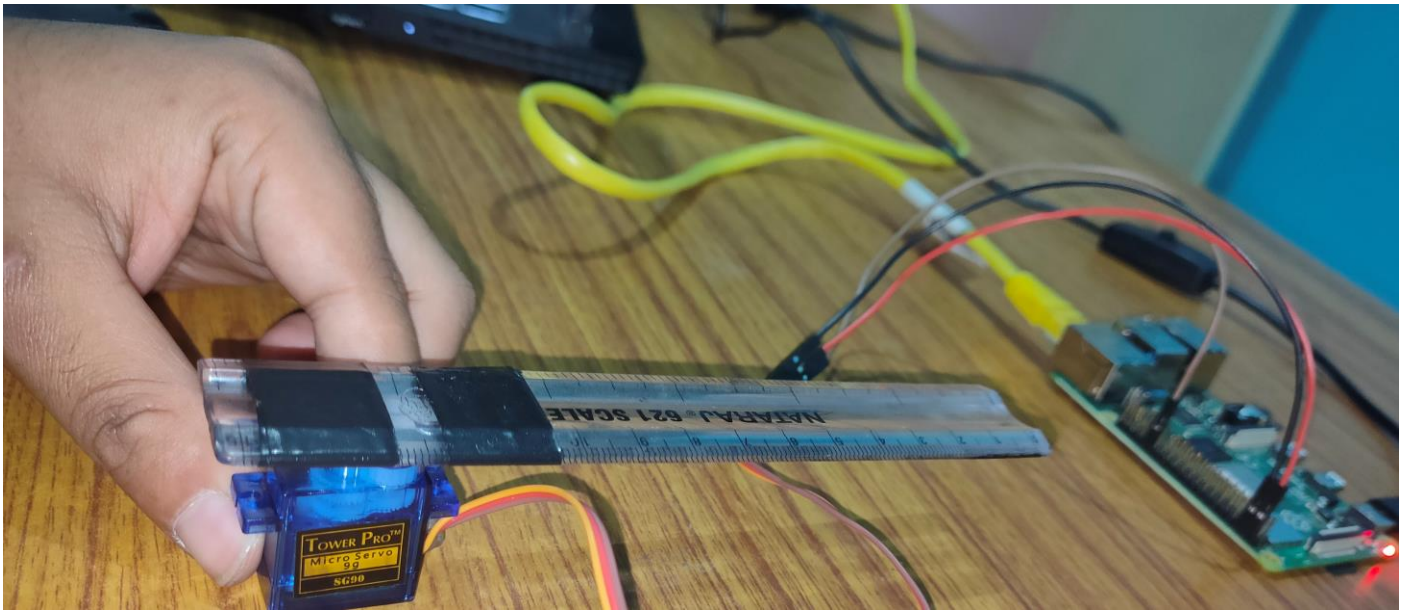
- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

Output along with supporting images:

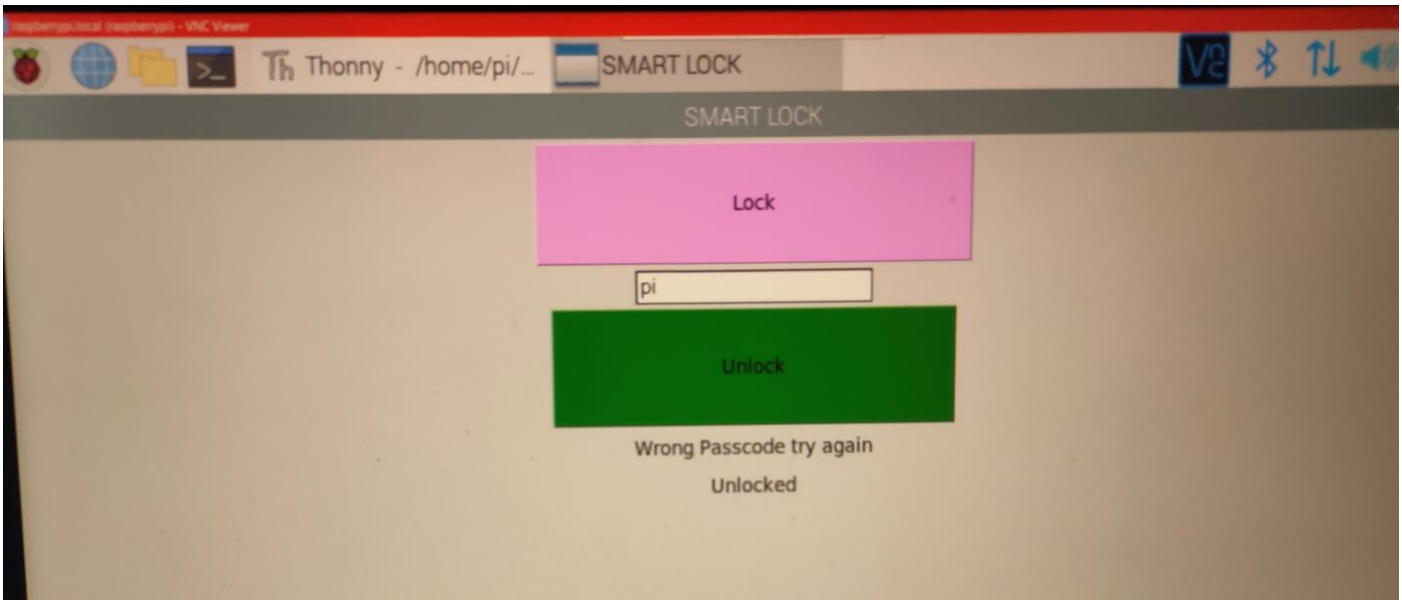
Entered wrong password:



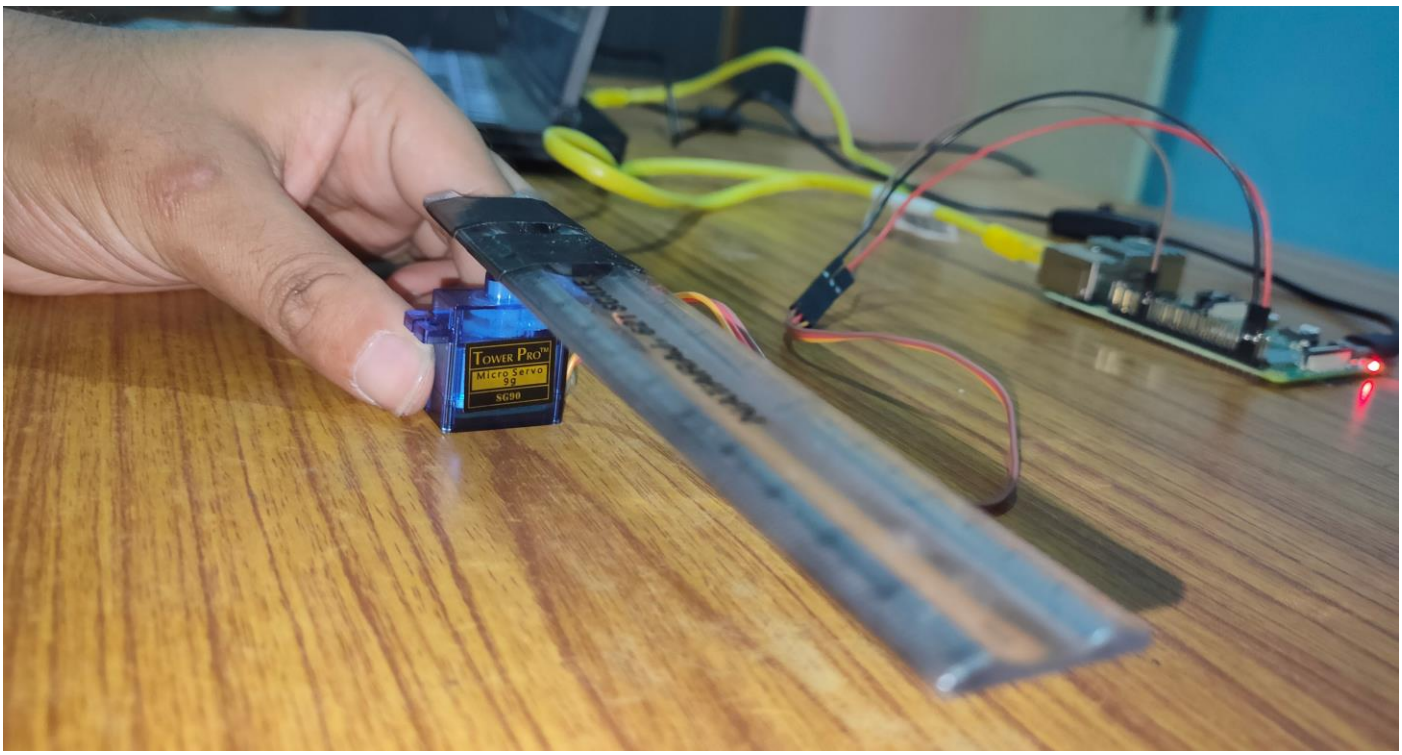
SERVO MOTOR DIDN'T ROTATES:



ENTERED CORRECT PASSWORD:

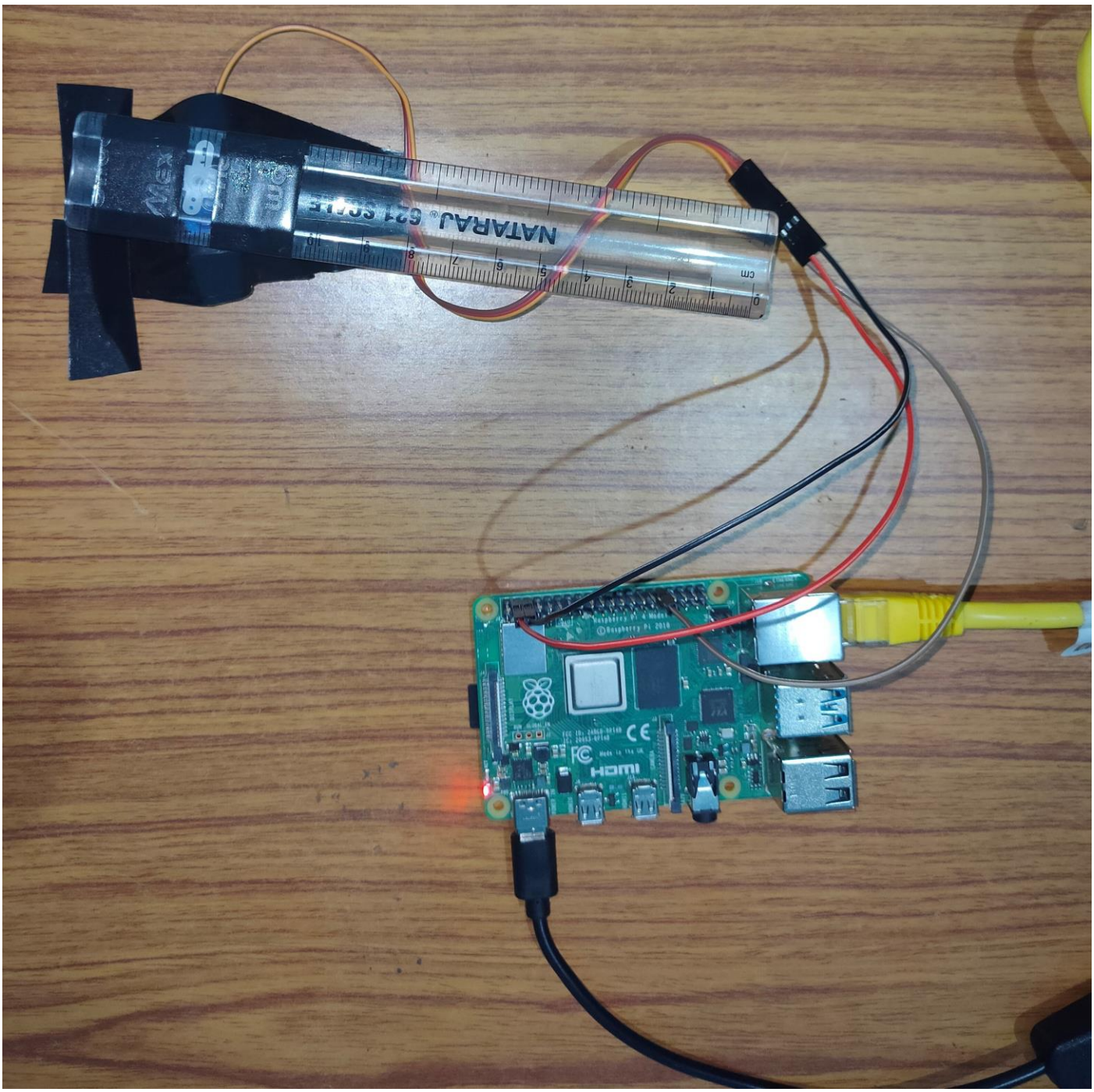


SERVO MOTOR ROTATES BY 90 DEGREE:

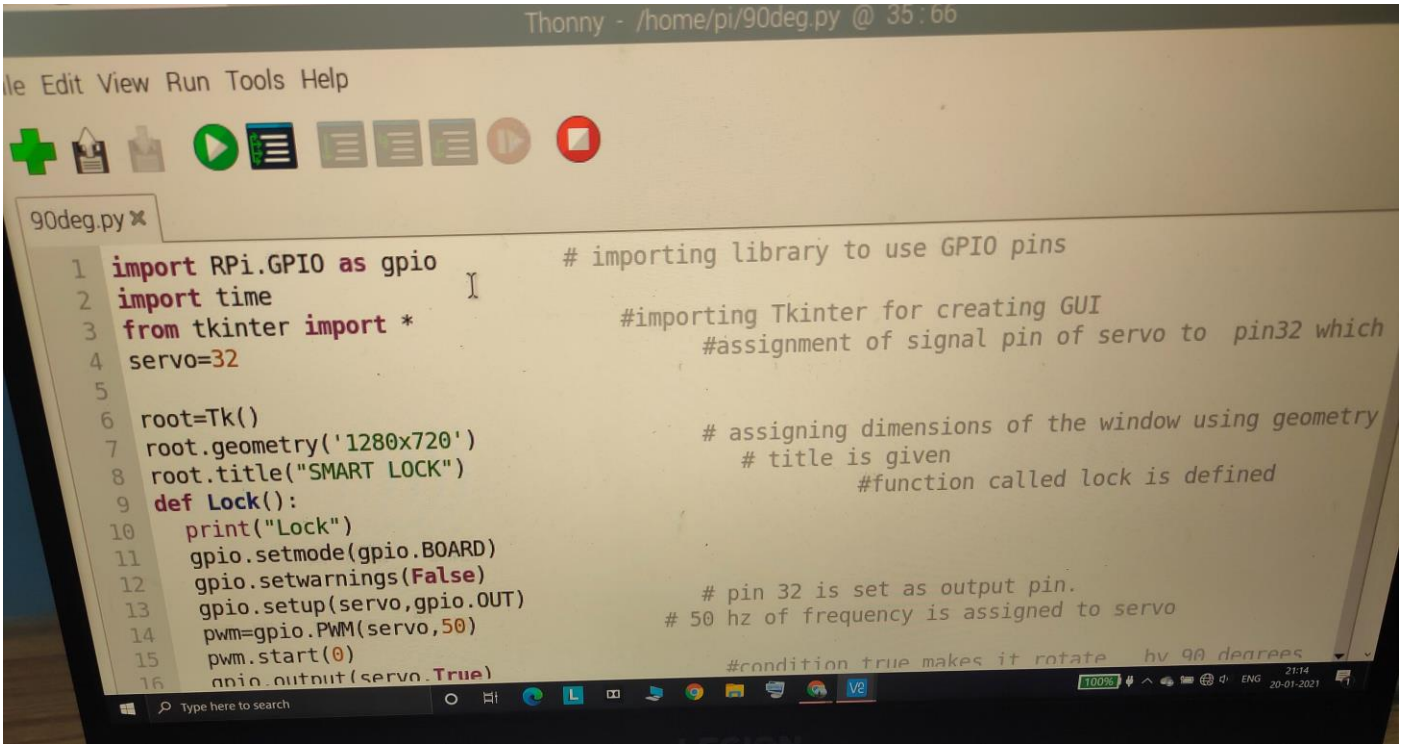


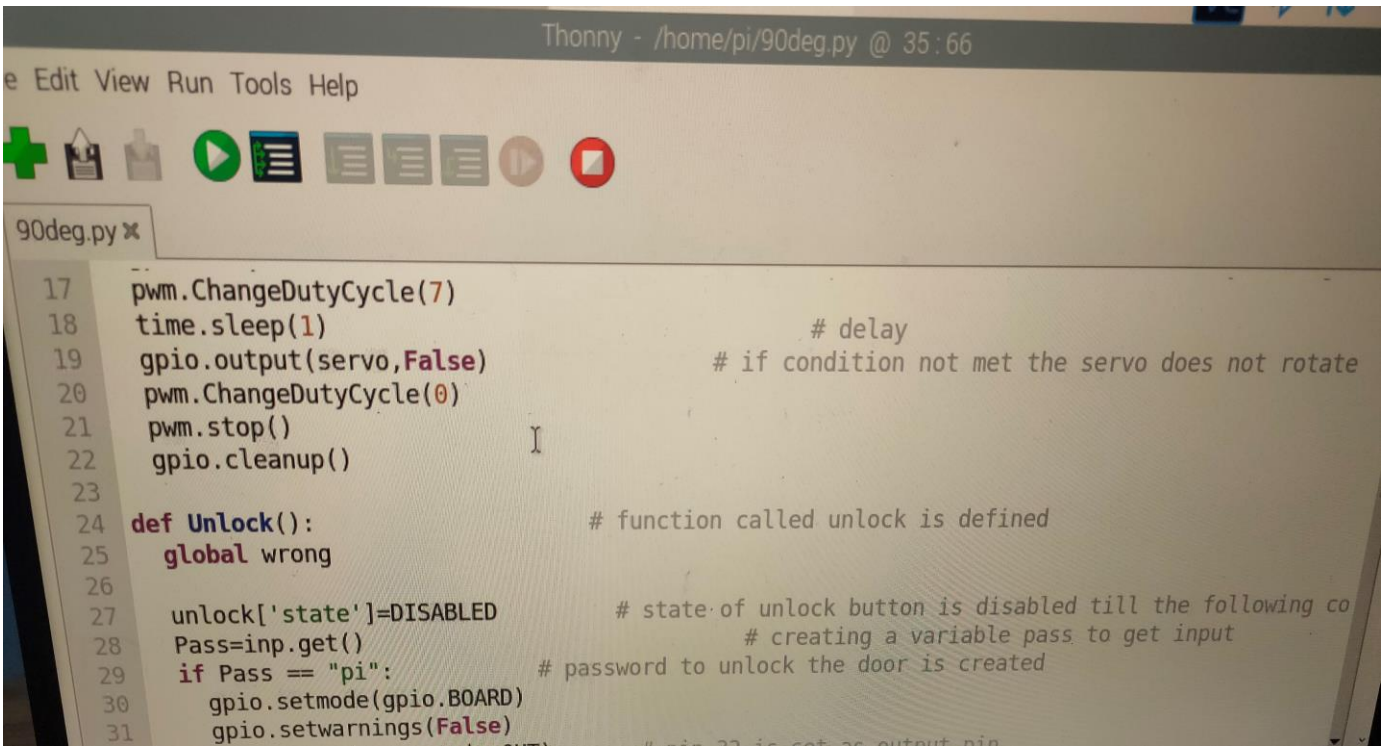
CIRCUIT DIAGRAM:

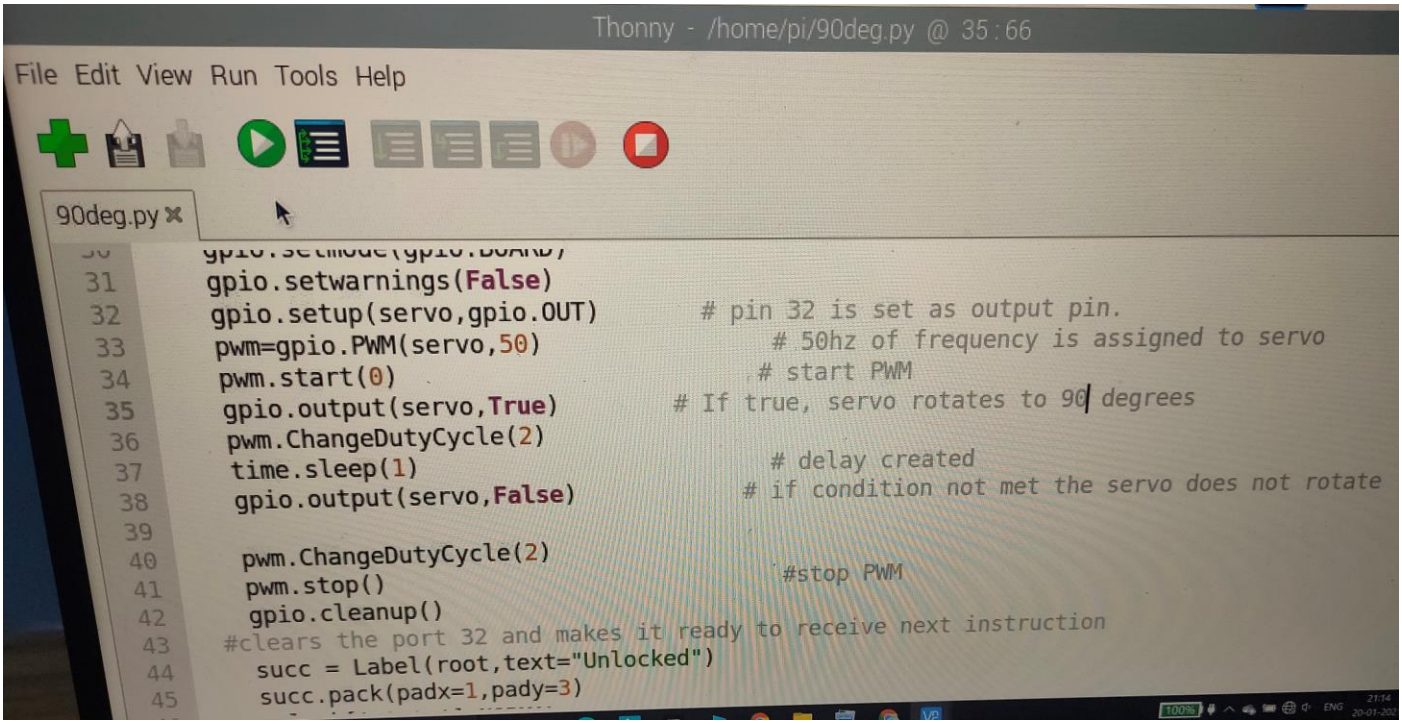
CONNECTED RASPBERRY PI TO SERVO MOTOR THROUGH 3 WIRES AVAILABLE WITH THE SERVO MOTOR 1 FOR VCC, 1 FOR GROUND , 1 FOR SIGNAL CONNECTED TO BOARD PIN NUMBER 32 WHICH IS A PWM PIN IN ORDER TO SEND HIGH/LOW PULSES:

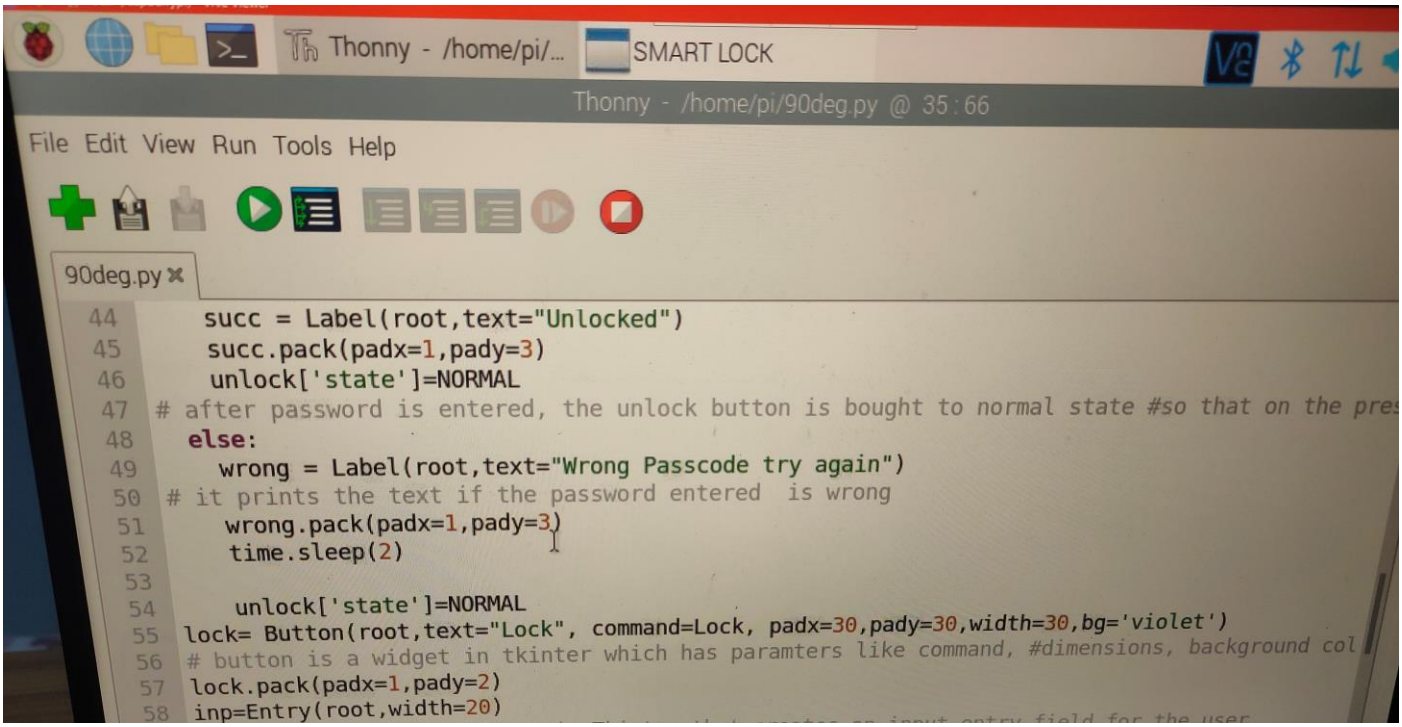


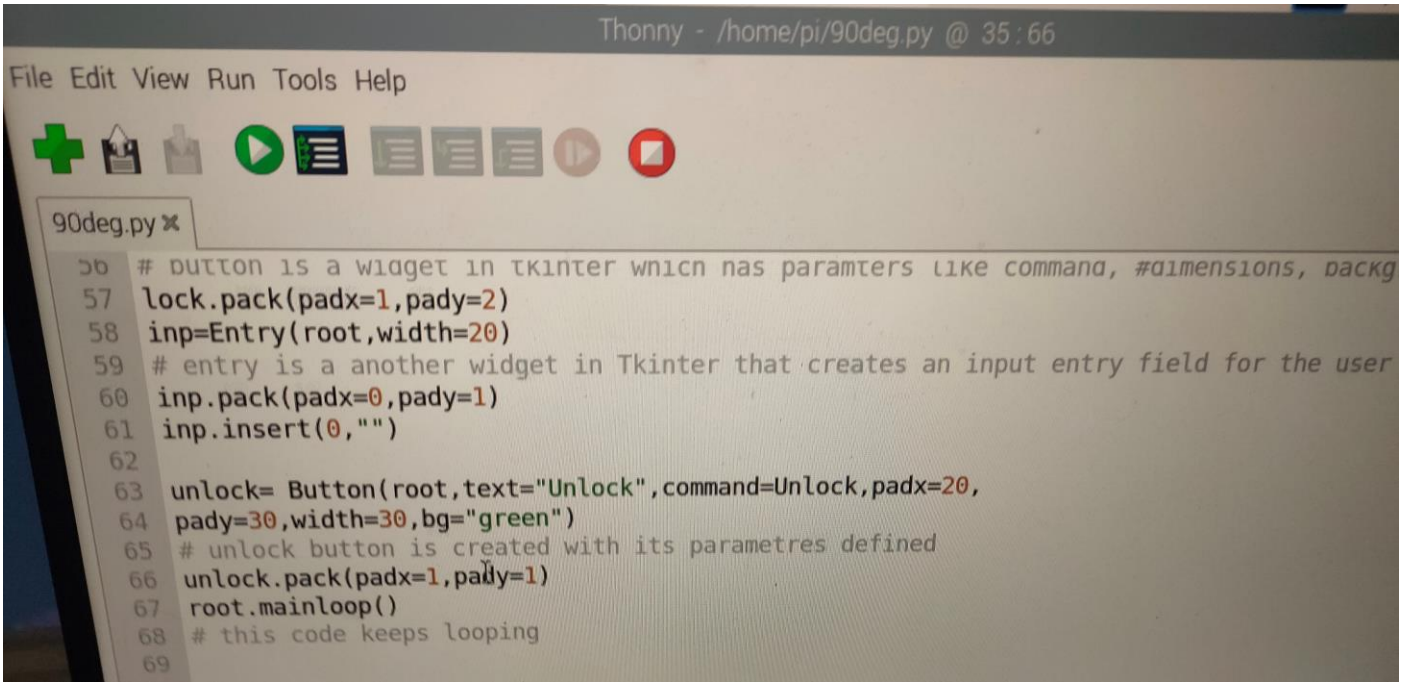
SNAPSHOTS OF CODE:






Thonny - /home/pi/90deg.py @ 35 : 66
File Edit View Run Tools Help
90deg.py
gpio.setwarnings(False)
gpio.setup(servo,gpio.OUT) # pin 32 is set as output pin.
pwm=gpio.PWM(servo,50) # 50hz of frequency is assigned to servo
pwm.start(0) # start PWM
gpio.output(servo,True) # If true, servo rotates to 90 degrees
pwm.ChangeDutyCycle(2)
time.sleep(1) # delay created
gpio.output(servo,False) # if condition not met the servo does not rotate
pwm.ChangeDutyCycle(2)
pwm.stop() #stop PWM
gpio.cleanup()
#clears the port 32 and makes it ready to receive next instruction
succ = Label(root,text="Unlocked")
succ.pack(padx=1,pady=3)


Thonny - /home/pi/...
SMART LOCK
Thonny - /home/pi/90deg.py @ 35 : 66
File Edit View Run Tools Help
90deg.py
succ = Label(root,text="Unlocked")
succ.pack(padx=1,pady=3)
unlock['state']=NORMAL
# after password is entered, the unlock button is bought to normal state #so that on the pres
else:
wrong = Label(root,text="Wrong Passcode try again")
# it prints the text if the password entered is wrong
wrong.pack(padx=1,pady=3)
time.sleep(2)
unlock['state']=NORMAL
lock= Button(root,text="Lock", command=Lock, padx=30,pady=30,width=30,bg='violet')
# button is a widget in tkinter which has paramters like command, #dimensions, background col
lock.pack(padx=1,pady=2)
inp=Entry(root,width=20)


Thonny - /home/pi/90deg.py @ 35 : 66
File Edit View Run Tools Help
90deg.py
# button is a widget in tkinter which has paramters like command, #dimensions, packg
lock.pack(padx=1,pady=2)
inp=Entry(root,width=20)
# entry is a another widget in Tkinter that creates an input entry field for the user
inp.pack(padx=0,pady=1)
inp.insert(0,"")
unlock= Button(root,text="Unlock",command=Unlock,padx=20,
pady=30,width=30,bg="green")
# unlock button is created with its parametres defined
unlock.pack(padx=1,pady=1)
root.mainloop()
# this code keeps looping

FUTURE SCOPE OF PROJECT:

- **Many companies have a main location and several remote locations, with remote locations not necessarily having cellular or internet connectivity. often there are rooms in remote locations that are locked for variety of security reasons so to save money on human interface equipment, otp(one time password) can be entered by a smartphone , the list of otp's , and therefore access to the locked room, can be controlled from a central location.**

- **We can add fingerprints so that entry will restricted for unauthorized persons.**

- **By interfacing fire and LPG sensors we can open the gate automatically in emergency situations.**

Regardless of the size of your commercial operation we have a solution for you and can provide standalone or networkable access control systems with user friendly custom made software. Our electronic door lock solutions include both electronic locking handles unitizing Mifare card credentials and Electronic key operated cylinders.

- **Office Buildings**
- **Shopping Centres**
- **Banks**
- **Server Rooms**
- **Hotels & Apartments**