**Remote Control Your**

**Raspberry Pi**

Team Members: Elizabeth Crittenden, Emily Omelyanovsky, Tyler O’Brien, Trey Nenni

Submission date: Sixth of May, 2021

**Table Of Contents**

1. **Title Page**(Emily) **. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .** 1
2. **Table of Contents**(Emily)**. . . . . . . . . . . . . . . . . . . . . . . . . . . . .** 2
3. **Executive Summary** (Tyler)**. . . . . . . . . . . . . . . . . . . . . . . . . . .** 3
   1. Problem Solved 3
   2. Technical Challenges 3
   3. Solution 3
   4. Project Performance 3
4. **Physical Components**(Liz+Trey) **. . . . . . . . . . . . . . . . . . . . . .** 3
5. **Software Description**(Liz+Tyler)**. . . . . . . . . . . . . . . . . . . . . . .** 5
6. **Project Assessment . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .** 6

# Performance(Liz+Trey) 6

# Team Dynamics(Emily) 7

**Executive Summary**

*Problem Solved:*

The problem we wanted to solve was being able to access the Raspberry Pi Remotely, from our PCs, instead of having to plug in a keyboard/monitor/mouse into the Raspberry Pi to control it.

*Technical Challenges:*

To do this, we needed a way to proxy Video and Audio over a network connection to another computer.

The common issues with remote desktop applications are encryption and security, prevent people on the same network from seeing what you do on the machine, and prevent unauthorized access. As well as performance and bandwidth consumption, video tends to be very bandwidth intensive.

*Solution:*

We solved this problem by using RealVNC on the Raspberry Pi, to open up a service to be able to connect to, and VNC viewer on the client machine, to be able to connect to the service on the Raspberry Pi. This solution was optimal because RealVNC has direct support for the Raspberry Pi and has industry-standard encryption, as well as letting users sign in via Linux accounts directly.

*Project Performance:*

As mentioned above, one of the technical challenges was bandwidth. This solution only eats up to around 0.5 MBPS when doing basic tasks and up to around 2 MBPS when watching a video at 1080p. The solution also consumed very little system resources, has built-in encryption, and authentication using Linux accounts directly.

Overall, it performed well for day-to-day use and required zero configuration.

Physical Components

*Raspberry Pi 4:*

The main components the Raspberry Pi 4 uses in this project are the Internet/Bluetooth module used to connect to the PC, the USB-C port for power and the microSD card slot. The micro-HDMI and two USB slots were used to set up the Raspberry Pi 4 to be remotely controlled but are no longer needed after the fact. This gives the Raspberry Pi two extra USB slots if needed for other uses.

Internet/Bluetooth Module

USB 3.0 Ports

USB 2.0 Ports

Ethernet Port

MicroSD Card Slot

A/V Port

USB-C Port

(Power)

Secondary Micro HDMI

Primary Micro HDMI

A picture containing text, electronics, circuit

Description automatically generated

*Dell Inspiron 3180 running Lubuntu 20.04 with Panda N600 Wireless Dongle*

To remotely control the Raspberry Pi, we used a Dell Inspiron laptop. This device long had issues with Windows functioning correctly, so the owner (Liz) replaced it with Lubuntu 20.04. Unfortunately, this version does not have the correct wi-fi driver, and the easiest fix was using a compatible dongle (Panda N600).



**Software Description**

*# Linux commands used* on Raspberry Pi:

# Get Local IP to connect via VNC Viewer Later

hostname -I

# Write down the IP for later

# update package lists

sudo apt-get update

# install newest versions of packages

sudo apt-get upgrade

# download VNC Connect

sudo apt-get install realvnc-vnc-server realvnc-vnc-viewer

# this part needed an extra step

sudo apt-get install realvnc-vnc-viewer --fix-missing

# open Raspberry Pi configuration tool to enable VNC

sudo raspi-config

# Click on Interface Options => Click on VNC  => Click Yes to the prompt "Do you want the VNC Server to be enabled"

# (Optional) To set it up so you can instantly use VNC to access the PI, even after

# a reboot

# and no monitor attached:

sudo raspi-config

# Click on System Options => Click on Boot/Auto Login => Navigate to and

# Select Desktop AutoLogin (B4)

# Click on Display Options => Click on Resolution => Navigate to and

# Select DMT Mode 82 1920 x 1080 60hz => Select Finish at the bottom =>

# Answer Yes to the reboot prompt.

on Dell Inspiron:

# VNC Viewer Downloaded from www.realvnc.com

# Linux’s Discover software center could not install the app, so it was done from the

# command line

sudo dkpg -i /tmp/mozilla\_elizabeth0/VNC-Viewer-6.20.529-Linux-x64.deb

# On Windows:

# Visit: <https://www.realvnc.com/en/connect/download/viewer/>

# Download and install, launch application, enter the IP from earlier command

# “hostname -I”, along with username/password of an account on the Pi, login

*Python used*

# Due to that this project required only Linux commands, I remotely accessed the Pi’s

# Mu IDE from the laptop to write a small Python program and run it

answer = “”

while answer != “yes”:

print(“Are mermaids real?”)

answer = input()

if answer == “yes”:

print(“Correct! Goodbye.”)

else:

print(“Wrong! Try again.”)

**Project Assessment**

a)

I. The project “Remotely Control Your Raspberry Pi” was chosen from the provided PDF “Practical Raspberry Pi Projects 5th Edition” (page 94). This guide creates a static IP address and uses RaspCTL to access the Pi remotely. After several failed attempts to install RaspCTL, we uncovered that it is a discontinued service. The two currently recommended replacements are VNC Server/Viewer and Dataplicity. We chose VNC Server because it gives remote access to the full graphical interface, not just the command line.

Once that issue was resolved, the project performed as well as or better than expected. The VNC service is easy to use and installed beautifully on both the Pi and the Dell laptop. The only aspect that did not perform as well as anticipated is the connection was a little spotty. The VNC Viewer froze periodically for a few seconds. This made playing Minecraft remotely a little frustrating, but coding in the Mu IDE that comes with the Pi had no problems. This may be because the slow speed of the wi-fi used, or the Pi’s CPU being strained by Minecraft too much.

II. The great thing about remotely controlling the Raspberry Pi is that any other project could run on it because it was set up using Linux commands. Setting up another project will be easier as the Raspberry Pi and researching how to make the project are done through the same device, which means there would not be unnecessary swapping between the PC display and the Raspberry Pi display. Some examples of projects that could be done with would be getting a robot kit and controlling the Raspberry Pi from the PC or running a server for an online game. The possibilities are only expanded with making the Raspberry Pi controlled remotely.

b)

I. All members of this team had the chance to fully participate in this project. Everyone was willing to do their part in completing this project to the best of their abilities. All members of the group were able to reach each other through e-mails and figure out who will do what part of the assignment.

II. Under today’s circumstances our groups were not able to meet in person, this could have helped the team immensely. When everyone is one place, discussing all ideas, it is easier to come to an understanding and even think of better solutions to the problem. However, since that is impossible due to social distancing and online learning, having zoom chats with everyone at a specific time may have been helpful to get to know one another and have everyone explain their thoughts on the project. Although this may have helped improve the project, it would have been difficult because many students have work or other things occupying certain times during the day. All in all, this team did well communicating, planning, and completing this project.