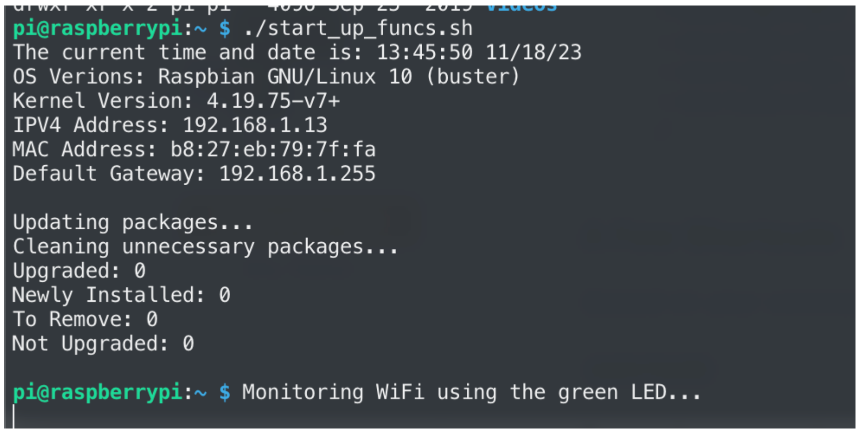
Michael Starks & Sovann Chak
Project 2 Module 3
ECEN 5803-002

1. We created a script using Bash.

2. To “convert” the script to an executable, we included the shebang at the top of the file to indicate that it is a bash executable. See Appendix for source code.

We developed 3 functions for the script:

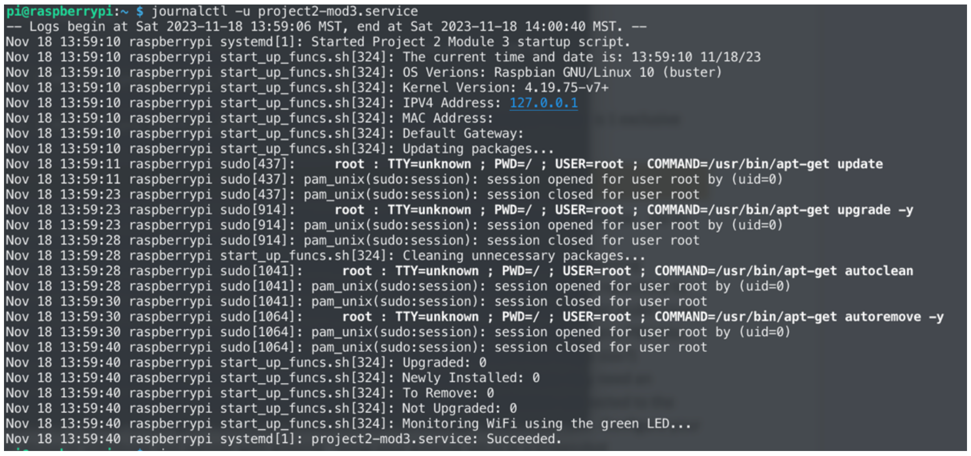
   a. console_display which displays the most recent/current information in terms of the date, time, operating system, kernel version, IPv4 address, MAC address, and Default Gateway.
   b. An update function which automatically updates necessary/removes unnecessary packages, and displays the accumulated counts of packages that were upgrade, newly installed, removed, or not upgraded.
   c. A WiFi monitor which will print a message of the current WiFi status as well as turn on the green LED. If WiFi is connected, the green LED is held on. If the WiFi is disconnected or disabled, the green LED is turned off. See the attached video in the submission package.


```
pi@raspberrypi:~ $ ./start_up_funcs.sh
The current time and date is: 13:45:50 11/18/23
OS Verions: Raspbian GNU/Linux 10 (buster)
Kernel Version: 4.19.75-v7+
IPV4 Address: 192.168.1.13
MAC Address: b8:27:eb:79:7f:fa
Default Gateway: 192.168.1.255

Updating packages...
Cleaning unnecessary packages...
Upgraded: 0
Newly Installed: 0
To Remove: 0
Not Upgraded: 0

pi@raspberrypi:~ $ Monitoring WiFi using the green LED...
```

3. We were able to get the script to run at boot time by
    a. Adding a “service” for systemctl. See Appendix for the project2-mod3.service file.
    b. Configuring using the following commands:
       sudo systemctl enable project2-mod3.service
       sudo systemctl start project2-mod3.service
    c. After a reboot, we confirmed our script was run using journalctl -u project2-mod3.service. See screenshot below.



```
pi@raspberrypi:~ $ journalctl -u project2-mod3.service
-- Logs begin at Sat 2023-11-18 13:59:06 MST, end at Sat 2023-11-18 14:00:40 MST. --
Nov 18 13:59:10 raspberrypi systemd[1]: Started Project 2 Module 3 startup script.
Nov 18 13:59:10 raspberrypi start_up_funcs.sh[324]: The current time and date is: 13:59:10 11/18/23
Nov 18 13:59:10 raspberrypi start_up_funcs.sh[324]: OS Verions: Raspbian GNU/Linux 10 (buster)
Nov 18 13:59:10 raspberrypi start_up_funcs.sh[324]: Kernel Version: 4.19.75-v7+
Nov 18 13:59:10 raspberrypi start_up_funcs.sh[324]: IPV4 Address: 127.0.0.1
Nov 18 13:59:10 raspberrypi start_up_funcs.sh[324]: MAC Address:
Nov 18 13:59:10 raspberrypi start_up_funcs.sh[324]: Default Gateway:
Nov 18 13:59:10 raspberrypi start_up_funcs.sh[324]: Updating packages...
Nov 18 13:59:11 raspberrypi sudo[437]:     root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND=/usr/bin/apt-get update
Nov 18 13:59:11 raspberrypi sudo[437]: pam_unix(sudo:session): session opened for user root by (uid=0)
Nov 18 13:59:23 raspberrypi sudo[437]: pam_unix(sudo:session): session closed for user root
Nov 18 13:59:23 raspberrypi sudo[914]:     root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND=/usr/bin/apt-get upgrade -y
Nov 18 13:59:23 raspberrypi sudo[914]: pam_unix(sudo:session): session opened for user root by (uid=0)
Nov 18 13:59:28 raspberrypi sudo[914]: pam_unix(sudo:session): session closed for user root
Nov 18 13:59:28 raspberrypi start_up_funcs.sh[324]: Cleaning unnecessary packages...
Nov 18 13:59:28 raspberrypi sudo[1041]:     root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND=/usr/bin/apt-get autoclean
Nov 18 13:59:28 raspberrypi sudo[1041]: pam_unix(sudo:session): session opened for user root by (uid=0)
Nov 18 13:59:30 raspberrypi sudo[1041]: pam_unix(sudo:session): session closed for user root
Nov 18 13:59:30 raspberrypi sudo[1064]:     root : TTY=unknown ; PWD=/ ; USER=root ; COMMAND=/usr/bin/apt-get autoremove -y
Nov 18 13:59:30 raspberrypi sudo[1064]: pam_unix(sudo:session): session opened for user root by (uid=0)
Nov 18 13:59:40 raspberrypi sudo[1064]: pam_unix(sudo:session): session closed for user root
Nov 18 13:59:40 raspberrypi start_up_funcs.sh[324]: Upgraded: 0
Nov 18 13:59:40 raspberrypi start_up_funcs.sh[324]: Newly Installed: 0
Nov 18 13:59:40 raspberrypi start_up_funcs.sh[324]: To Remove: 0
Nov 18 13:59:40 raspberrypi start_up_funcs.sh[324]: Not Upgraded: 0
Nov 18 13:59:40 raspberrypi start_up_funcs.sh[324]: Monitoring WiFi using the green LED...
Nov 18 13:59:40 raspberrypi systemd[1]: project2-mod3.service: Succeeded.
```

Appendix.

start_up_funcs.sh:

```
#!/bin/bash


# Authors: Michael Starks and Sovann Chak

# ECEN 5803 - 002

# Project 2 Module 3




# Displays some information on the

# current time/date and relevant

# HW and SW information

console_display() {

    current_time_date=$(date +%T)

    current_time_date+=" "

    current_time_date+=$(date +%D)

    os_version=$(cat /etc/os-release | sed -n 's/^PRETTY_NAME="\(.*\)"/\1/p')

    kernel_version=$(uname -r)

    ipv4_address=$(ip -f inet addr show $(ip addr | awk '/state UP/ {print $2}' | sed 's/.$//') | grep -
Po 'inet \K[\d.]+')

    default_gateway=$(ip -f inet addr show $(ip addr | awk '/state UP/ {print $2}' | sed 's/.$//') |
grep -Po 'brd \K[\d.]+')

    mac_address=$(ifconfig  | grep -A1 eth0 | grep ether | awk '{print $2}')

    echo "The current time and date is: $current_time_date"

    echo "OS Verions: $os_version"

    echo "Kernel Version: $kernel_version"

    echo "IPV4 Address: $ipv4_address"

    echo "MAC Address: $mac_address"

    echo "Default Gateway: $default_gateway"

    echo ""
```

```
}

# Updates the system and accumulates the counts
# to display a total count
update(){

    echo "Updating packages..."
    # silence output from update, I don't want it
    sudo apt-get update > /dev/null 2>&1
    upgrades=$(sudo apt-get upgrade -y)

    echo "Cleaning unnecessary packages..."
    # silence output from clean, I don't want it
    sudo apt-get autoclean > /dev/null 2>&1
    removes=$(sudo apt-get autoremove -y)



    # Collect the counts from each so that
    # we an add them up as a single print
    # message
    upgrade_counts=$(echo $upgrades | sed 's/[^0-9]\+/ /g')
    remove_counts=$(echo $removes | sed 's/[^0-9]\+/ /g')

    upgrade_count0=$(echo $upgrade_counts | awk '{print $1}')
    upgrade_count1=$(echo $upgrade_counts | awk '{print $2}')
    upgrade_count2=$(echo $upgrade_counts | awk '{print $3}')
    upgrade_count3=$(echo $upgrade_counts | awk '{print $4}')

    remove_count0=$(echo $remove_counts | awk '{print $1}')
    remove_count1=$(echo $remove_counts | awk '{print $2}')
    remove_count2=$(echo $remove_counts | awk '{print $3}')
    remove_count3=$(echo $remove_counts | awk '{print $4}')

    count0=$((upgrade_count0 + remove_count0))
    echo "Upgraded: $((upgrade_count1 + remove_count1))"
    echo "Newly Installed: $((upgrade_count2 + remove_count2))"
    echo "To Remove: $((upgrade_count2 + remove_count2))"
    echo "Not Upgraded: $((upgrade_count3 + remove_count3))"
    echo ""
}

# Monitors the WiFi status and
# if the WiFi is connected to a
# network then the green LED will
# stay lit
monitor_wifi(){
#!/bin/bash

echo "Monitoring WiFi using the green LED..."

# Assuming these are static
INTERFACE="wlan0"
STATE_FILE="/tmp/wifi_state"

state_string="off"
# Initializes the LED and STATE_FILE by
# checking if the interface is up via ip command and
# by using grep, if grep returns something then we
# know the WiFi interface is up and can enable the green LED
```

```
ip link show wlan0 | grep 'state UP' > /dev/null && (echo "up" > /tmp/wifi_state && sudo echo
default-on > /sys/class/leds/led0/trigger && echo "WiFi is on") \
                                    || (echo "down" > /tmp/wifi_state && sudo echo none >
/sys/class/leds/led0/trigger && echo "WiFi is off")


# Start the monitoring loop
while true; do

    ip link show $INTERFACE | grep 'state UP' > /dev/null
    if [ $? -eq 0 ]; then
        curr_state=1
        trigger=default-on
                state=on
    else
        curr_state=0
        trigger=none
                state=off
    fi

    # Read previous state from state file
    prev_state=$(cat $STATE_FILE)

    # Compare the current state with the previous state
    if [ "$curr_state" != "$prev_state" ]; then
        echo $curr_state > $STATE_FILE
        echo $trigger > /sys/class/leds/led0/trigger
                echo "WiFi is currently $state"
    fi
    # Wait, don't want to each too much processing
    sleep 1
```

```
done
}

# Raspbian Linux is greedy when it comes
# to control, so we must usurp control
chmod 666 /sys/class/leds/led0/trigger

# Calling the functions
console_display
update
monitor_wifi

# end script
```

project2-mod3.service:

```
[Unit]
Description=Project 2 Module 3 startup script

[Service]
Type=simple
ExecStart=/home/pi/start_up_funcs.sh

[Install]
WantedBy=multi-user.target
```