Programming Traffic Signal

Implementation using QEMU –

We began the implementation by using the command,

```
qemu-img create -f qcow2 ubuntu_disk1.qcow2 30G
```

This helps in generating new disk image, named as **ubuntu_disk1.qcow2**

As a next step, we've implemented the following command to run the qemu system emulator

```
qemu-system-x86_64 -m 8G -hda ubuntu_disk1.qcow2 -cdrom ubuntu-20.04.1-desktop-amd64.iso -boot d
```

After implementing the above command, we've opened the terminal and verified the APT cache which helps in identifying the cross build essential tools for ARM by using the following command,

```
apt-cache search cross-build-essential
```

Later we installed cross-compiler tools with the following command,

```
sudo apt install crossbuild-essential-armhf
```

As a follow up for the previous command, to check for the cross-compiler installation we used,

```
arm-linux-gnueabihf-gcc -v
```

We tried to install user mode qemu for ARM command by using the following,

```
sudo add-apt-repository universe
sudo apt update
sudo apt install qemu-user-static
```

Adding the armhf packages and cross checking if it’s added by using this command,

```
sudo dpkg –add-architecture armhf
```

```
dpkg –print-foreign-architectures
```

To update the packages, we used

```
sudo apt update
```

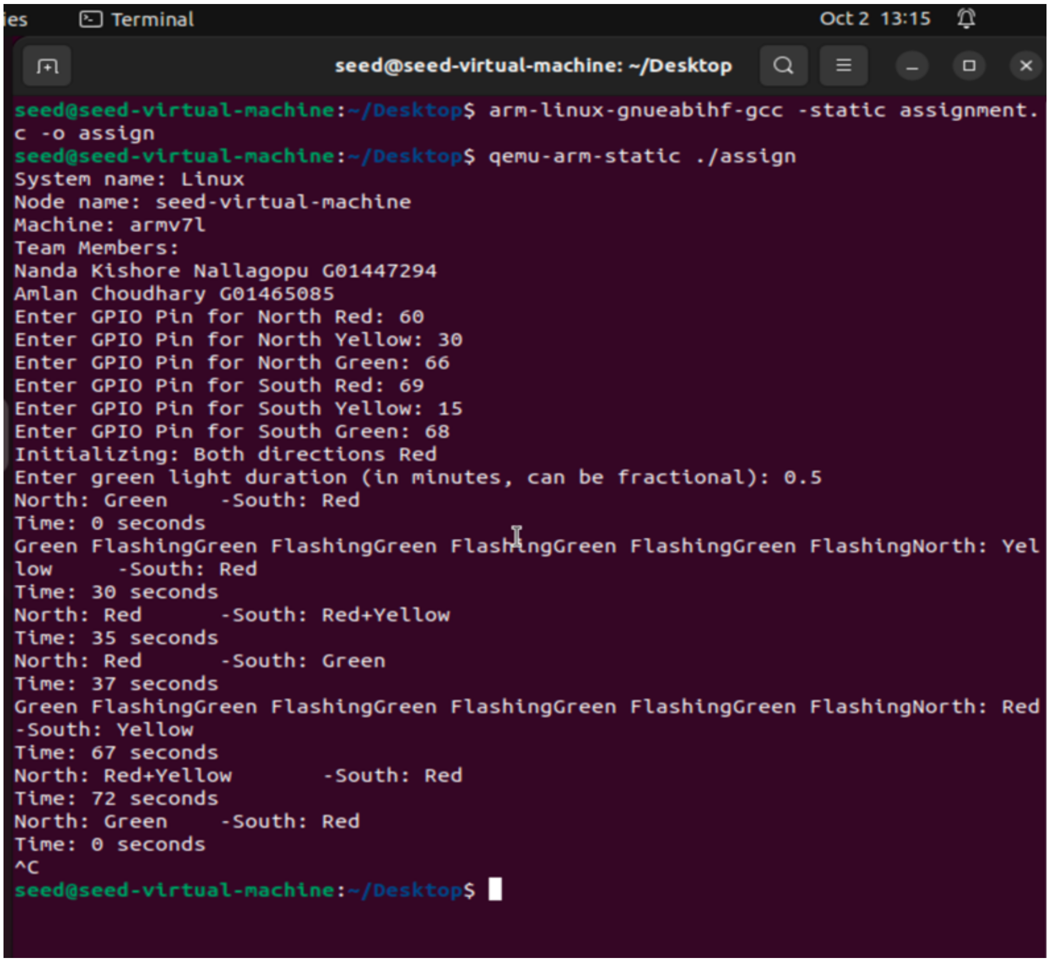
To compile the C code, we used

```
arm-linux-gnueabihf-gcc –static assignment.c -o assign
```

And for the output,

```
qemu-arm-static ./assign
```

Output Screenshot –



```
seed@seed-virtual-machine:~/Desktop$ arm-linux-gnueabihf-gcc -static assignment.
c -o assign
seed@seed-virtual-machine:~/Desktop$ qemu-arm-static ./assign
System name: Linux
Node name: seed-virtual-machine
Machine: armv7l
Team Members:
Nanda Kishore Nallagopu G01447294
Amlan Choudhary G01465085
Enter GPIO Pin for North Red: 60
Enter GPIO Pin for North Yellow: 30
Enter GPIO Pin for North Green: 66
Enter GPIO Pin for South Red: 69
Enter GPIO Pin for South Yellow: 15
Enter GPIO Pin for South Green: 68
Initializing: Both directions Red
Enter green light duration (in minutes, can be fractional): 0.5
North: Green    -South: Red
Time: 0 seconds
Green FlashingGreen FlashingGreen FlashingGreen FlashingGreen FlashingNorth: Yel
low     -South: Red
Time: 30 seconds
North: Red      -South: Red+Yellow
Time: 35 seconds
North: Red      -South: Green
Time: 37 seconds
Green FlashingGreen FlashingGreen FlashingGreen FlashingGreen FlashingNorth: Red
-South: Yellow
Time: 67 seconds
North: Red+Yellow       -South: Red
Time: 72 seconds
North: Green    -South: Red
Time: 0 seconds
^C
seed@seed-virtual-machine:~/Desktop$
```

Implementation using Beagle Bone Black –

As a first step, we imported the beagle bone black image using this command,

bone-debian-10.3-iot-armhf-2020-04-06-4gb.img

After running this code successfully, we tried to connect the micro-SD card to the system.

To burn the image on to the SD card, we used Balena Etcher.

We ejected the card from the system after this step and connected it to the beagle bone board.

Now we connected the beagle bone to the system through a cable. The folder was included in the system.

We've opened the terminal to run it on board by using the following commands,

```
ssh debain@192.168.6.2
```

And the password was set to **temppwd**

The above steps help in connecting it to the beagle bone.

As a next step, we are running this command,

```
vi assignment.c
```

We wrote the C code over there and we compiled it using the command,

```
gcc -o assignment assignment.c
```

And to run the program:

```
sudo ./assignment
```

GPIO Pins:

For North:

Red – 60

Yellow – 30

Green – 66

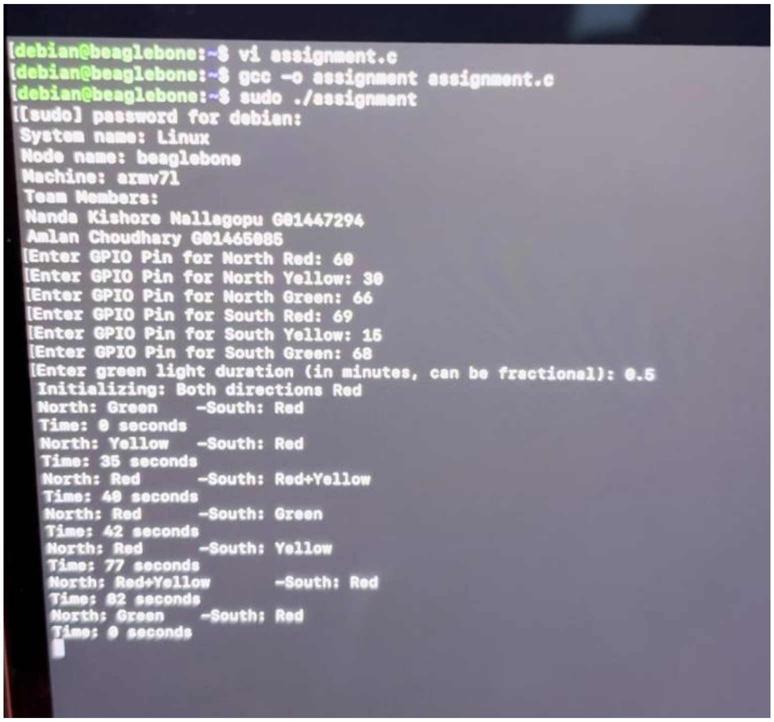
For South:

Red – 69

Yellow – 15

Green – 68

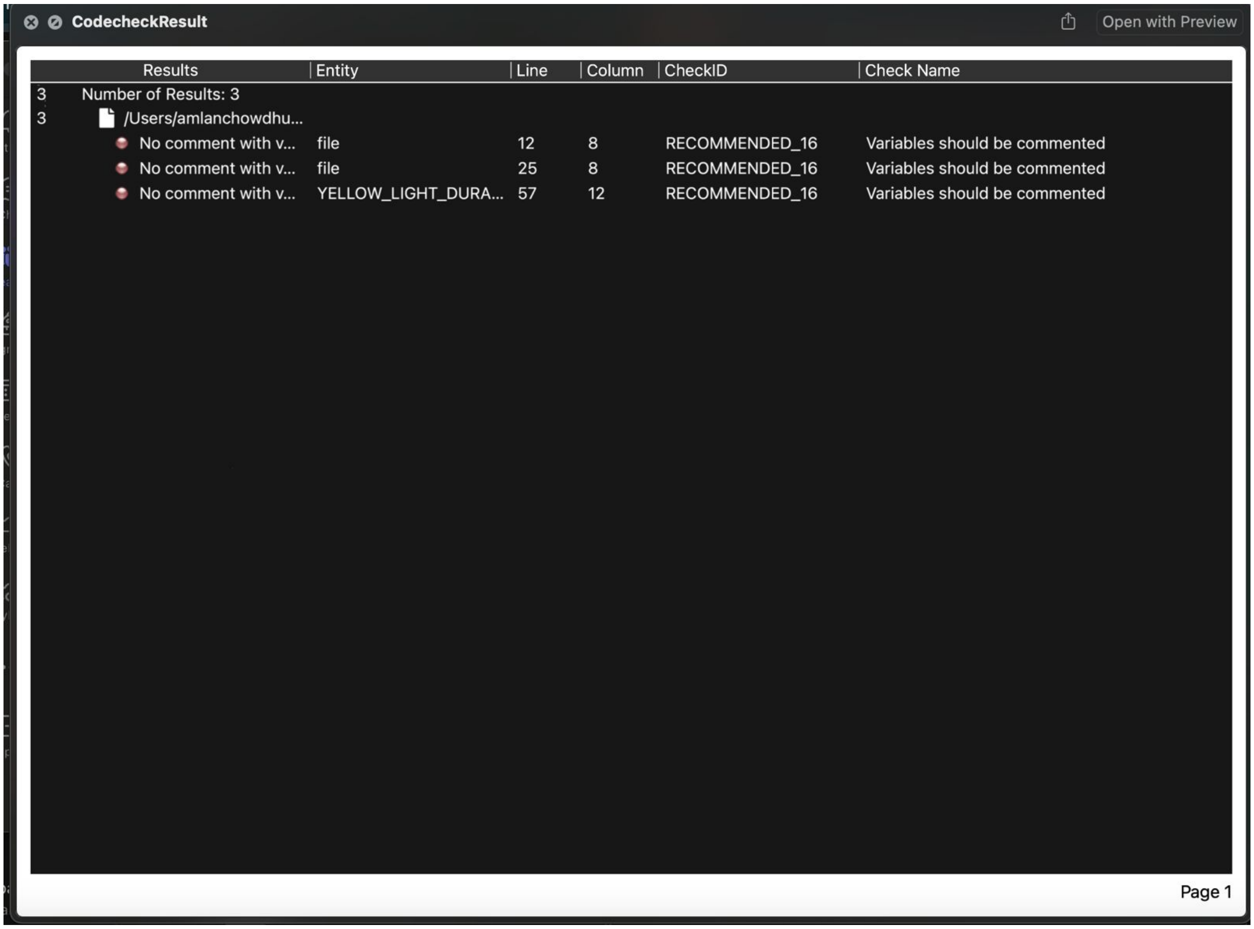
Output Screenshot –



```
[debian@beaglebone:~$ vi assignment.c
[debian@beaglebone:~$ gcc -o assignment assignment.c
[debian@beaglebone:~$ sudo ./assignment
[[sudo] password for debian:
 System name: Linux
 Node name: beaglebone
 Machine: armv7l
 Team Members:
 Nanda Kishore Nallagopu G01447294
 Amlan Choudhary G01465085
[Enter GPIO Pin for North Red: 60
[Enter GPIO Pin for North Yellow: 30
[Enter GPIO Pin for North Green: 66
[Enter GPIO Pin for South Red: 69
[Enter GPIO Pin for South Yellow: 15
[Enter GPIO Pin for South Green: 68
[Enter green light duration (in minutes, can be fractional): 0.5
 Initializing: Both directions Red
 North: Green    -South: Red
 Time: 0 seconds
 North: Yellow   -South: Red
 Time: 35 seconds
 North: Red      -South: Red+Yellow
 Time: 40 seconds
 North: Red      -South: Green
 Time: 42 seconds
 North: Red      -South: Yellow
 Time: 77 seconds
 North: Red+Yellow       -South: Red
 Time: 82 seconds
 North: Green    -South: Red
 Time: 0 seconds
```

MISRA – Reflection

We tried to solve most of the MISRA warnings. We are running out of time to complete the 3 warnings left. They are trivial errors as these blocks of codes were not sufficiently commented and does not affect the overall integrity of the code.



	Results	Entity	Line	Column	CheckID	Check Name
3	Number of Results: 3					
3	/Users/amlanchowdhu...					
	● No comment with v...	file	12	8	RECOMMENDED_16	Variables should be commented
	● No comment with v...	file	25	8	RECOMMENDED_16	Variables should be commented
	● No comment with v...	YELLOW_LIGHT_DURA...	57	12	RECOMMENDED_16	Variables should be commented

Page 1