Byeongchan Gwak, Student ID : 501026

Washington University in St.Louis

McKelvey School of Engineering

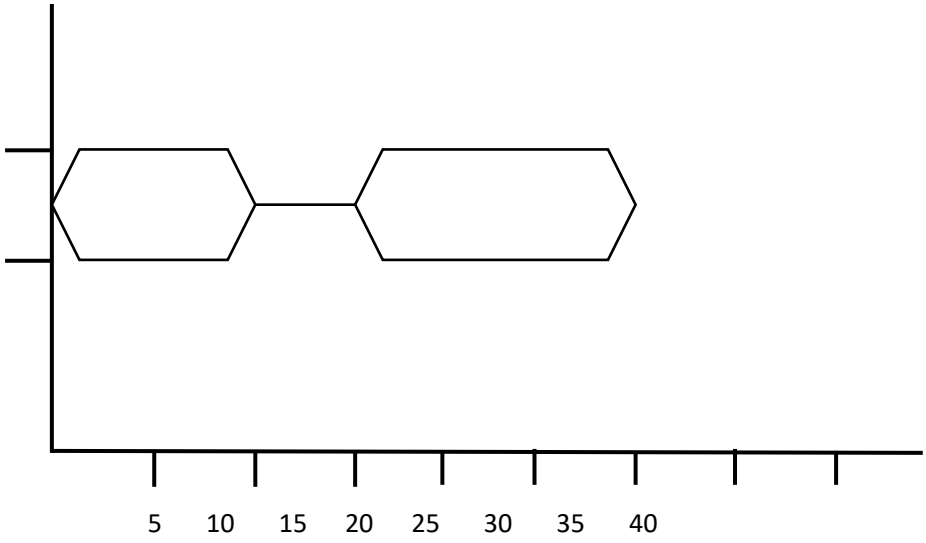
Fall Semester 2021

CSE467M: Embedded Computing Systems

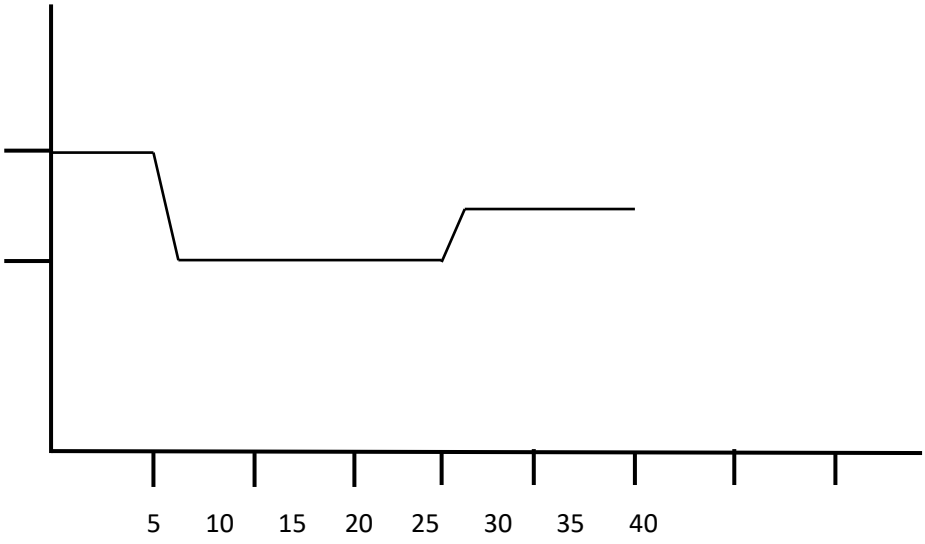
Homework #3

Chapter 4 problems:

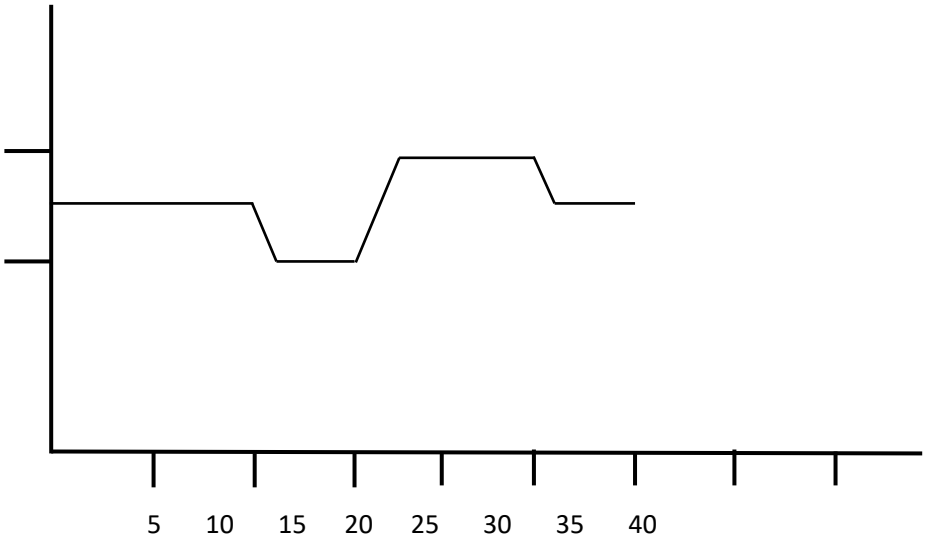
1) **Q4-1 (3 points)**
   a. **CPU, Memory, I/O Devices**
2) **Q4-7 (10 points)**
   a. **Signal A is stable [0, 10], changing [10,15], stable [15, 30]**



b. Signal B is 1 [0,5], falling [5,7], 0 [7,20], changing [20,30]

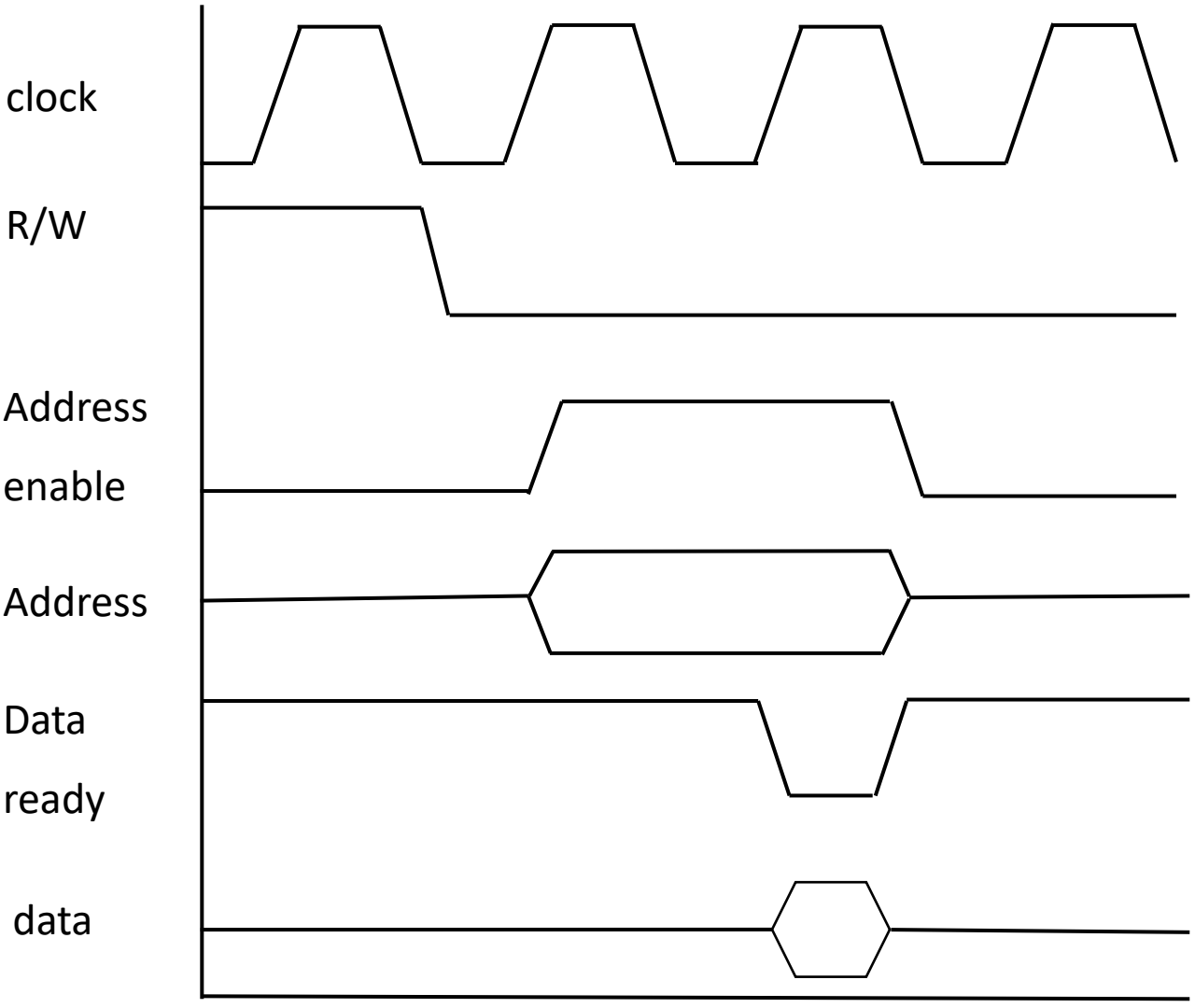


c. Signal C is changing [0,10], 0 [10,15], rising [15,18], 1 [18,25], changing [25,30]



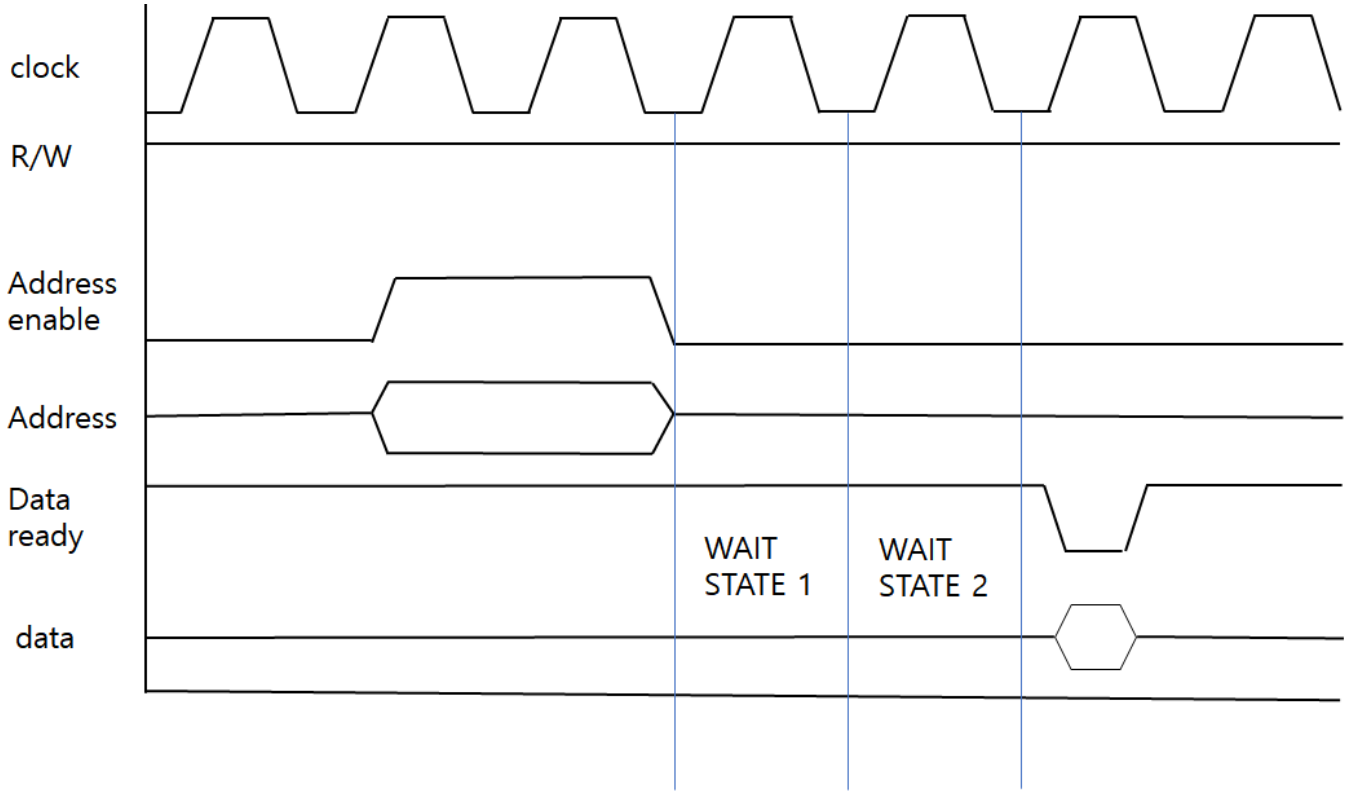
Byeongchan Gwak, Student ID : 501026

3) Q4-8 (10 points)

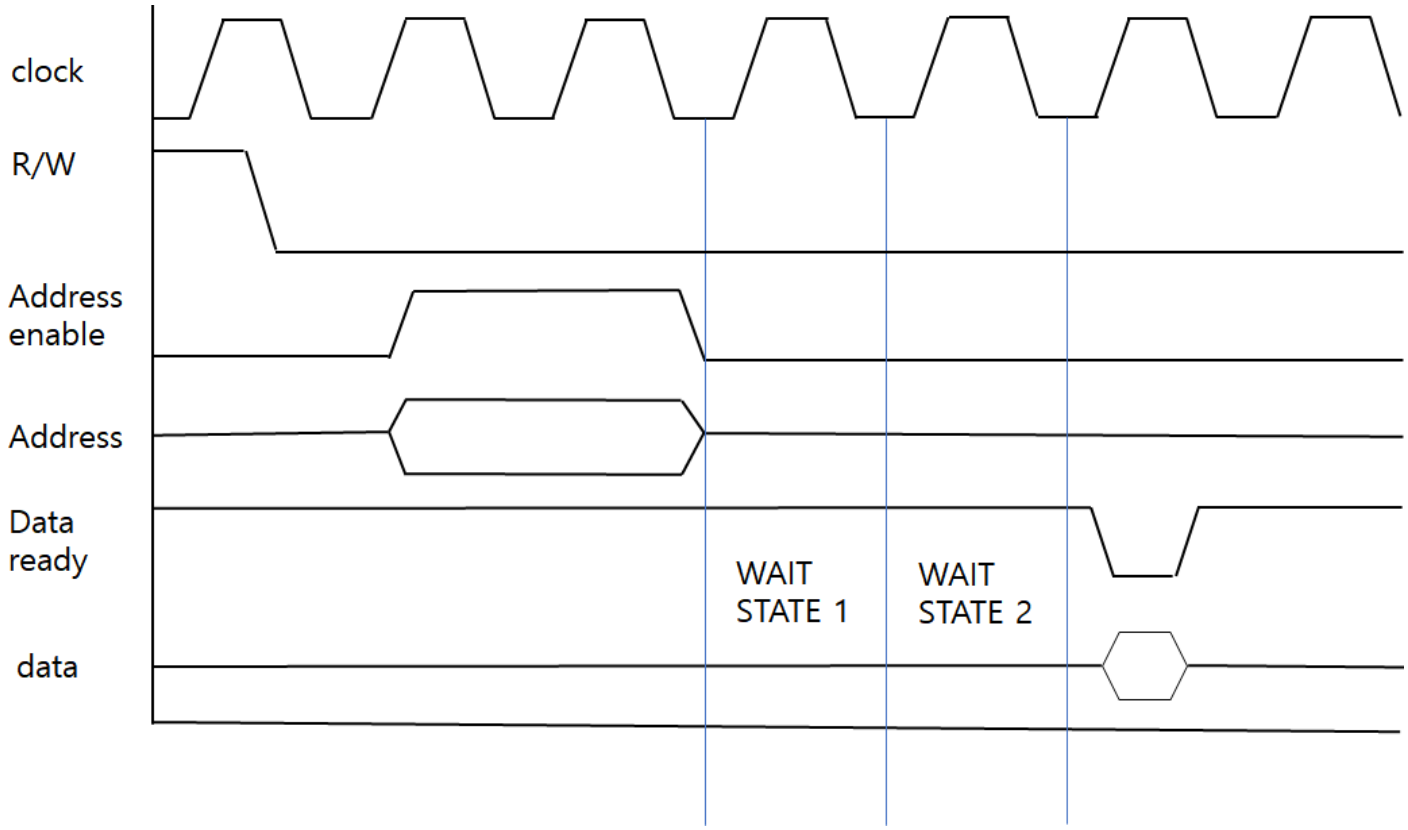


Byeongchan Gwak, Student ID : 501026

4) Q4-9 (10 points)

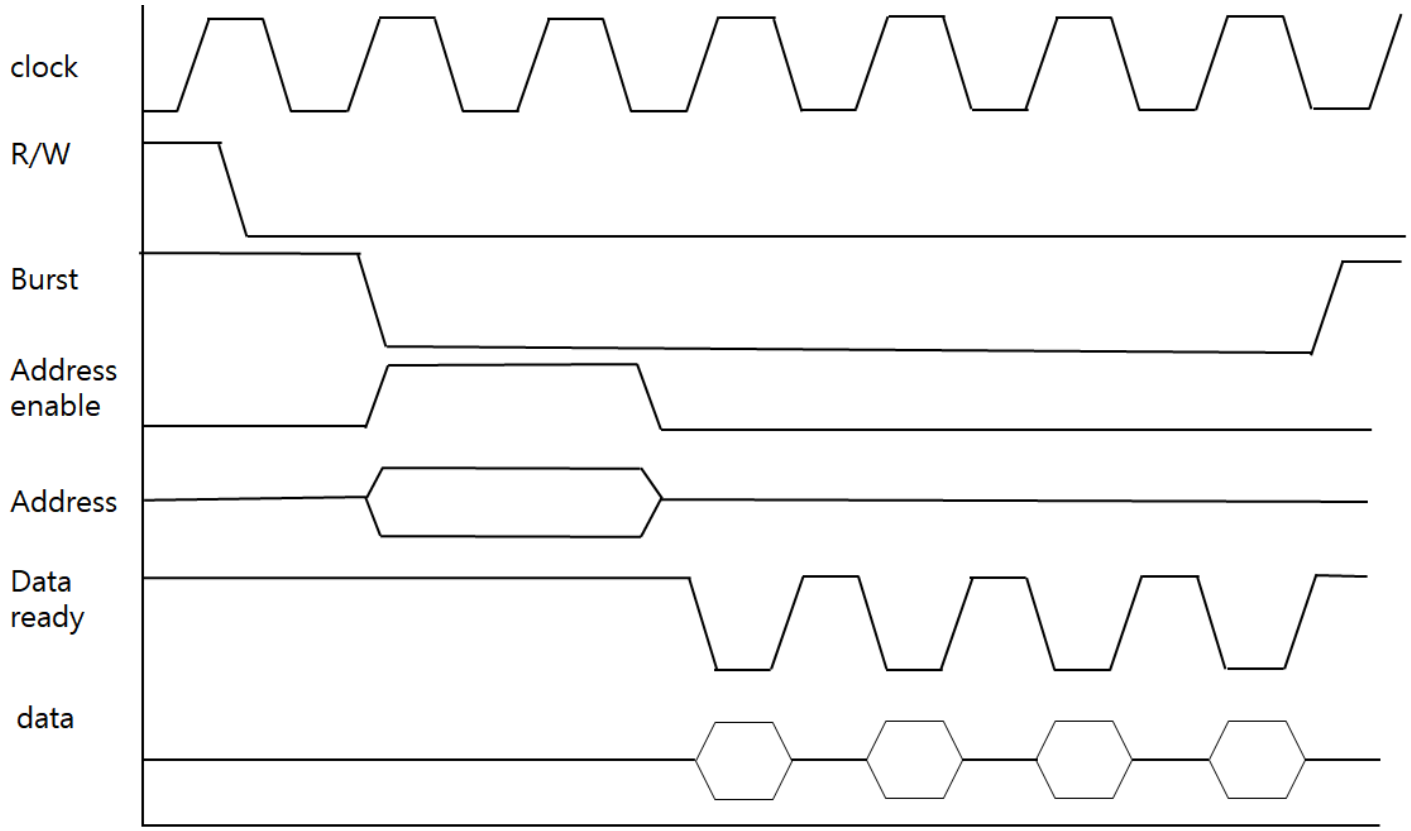


5) Q4-10 (10 points)



Byeongchan Gwak, Student ID : 501026

6) Q4-11 (10 points)



7) Q4-28 (15 points)

a. 1/44100 Hz = 0.0000226757 sec ≈ 22.68 µs
b. Get a time per instruction: 1/ 20 MHz = 0.0000000500 = 50 ns
c. During obtaining the sample, the RISC process can handle 22.68 µs / 50 ns = 22680 / 50 = 453.6 instructions
d. Since the interrupt handler execute 100 instructions, so 453.6 – 100 = 353.6 ≈ 353.
e. As a result 353 instructions can be executed with the RISC microprocessor.

Byeongchan Gwak, Student ID : 501026

8) **Q4-37 (32 points)**

a. **Data rate**

= 2byte * 2 channels * 44.1 kHz

= 2 * 2 * 44.1 * 1000

= 176400 bytes/sec

9) Raspberry Pi Circuit Card Assembly (CCA) Laboratory #1 – Blinking LED (600 points)

For this laboratory you need to do the following tasks:

9.1) Set up Raspberry Pi CCA as it is explained in the CanaKit Raspberry Pi Quick-Start Guide.

9.2) Obtain LED, resistor R = 220 Ω, some wires and some clips so that you can connect wires to the GPIO connect without soldering. If you have a breadboard then you can use it as well but clips will do the job even if you do not have a breadboard.

9.3) Connect LED and Resistor to the GPIO connector as it shown here in Figure 1. You can use breadboard as it is shown in Figure 1, if you have it. If you do not have the breadboard you can use the clips to connect wires to 220 Ω resistor, LED and GPIO connector pins on the Raspberry Pi CCA.

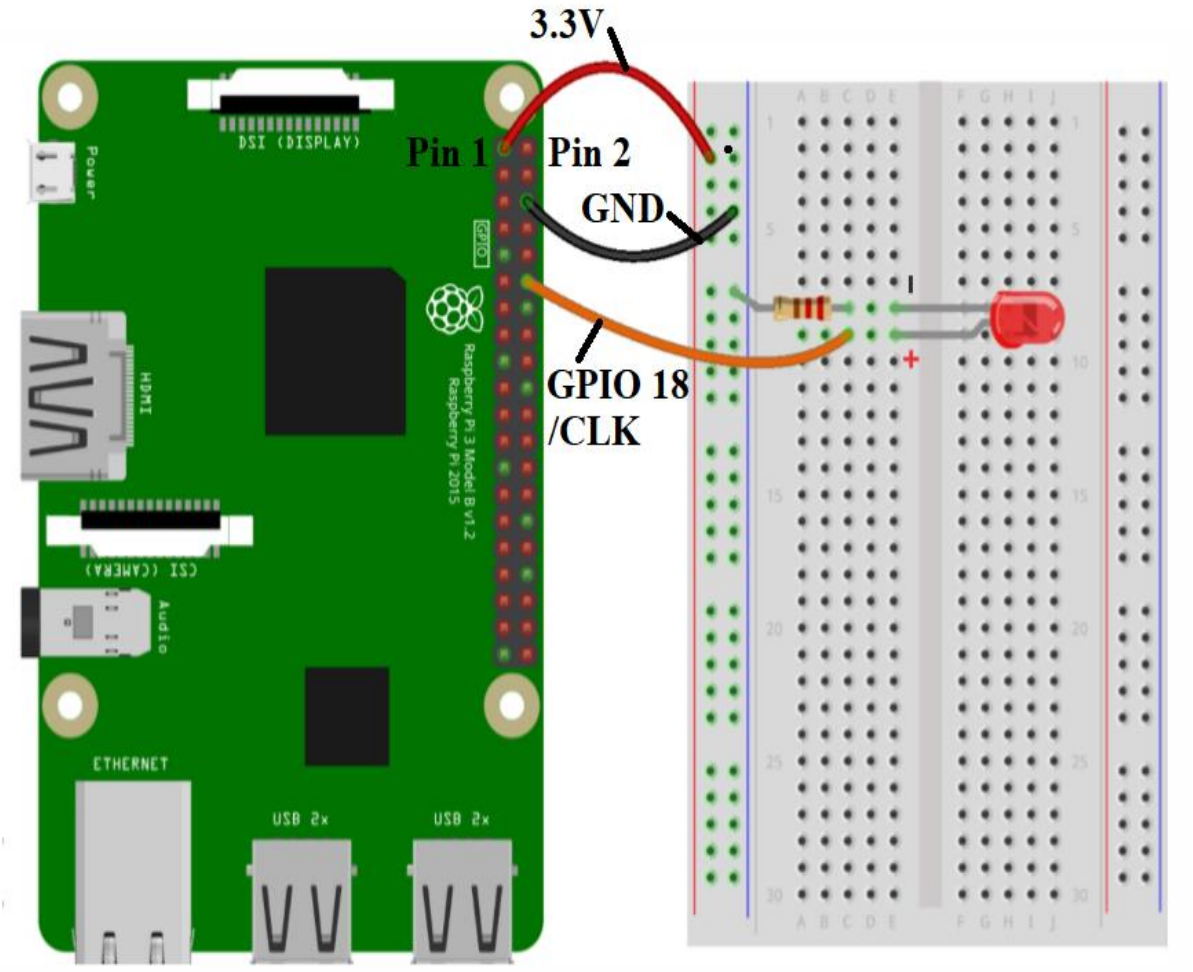


Figure 1. Blinking LED Laboratory 1

GPIO connector on the Raspberry Pi CCA is shown in Figure 2.

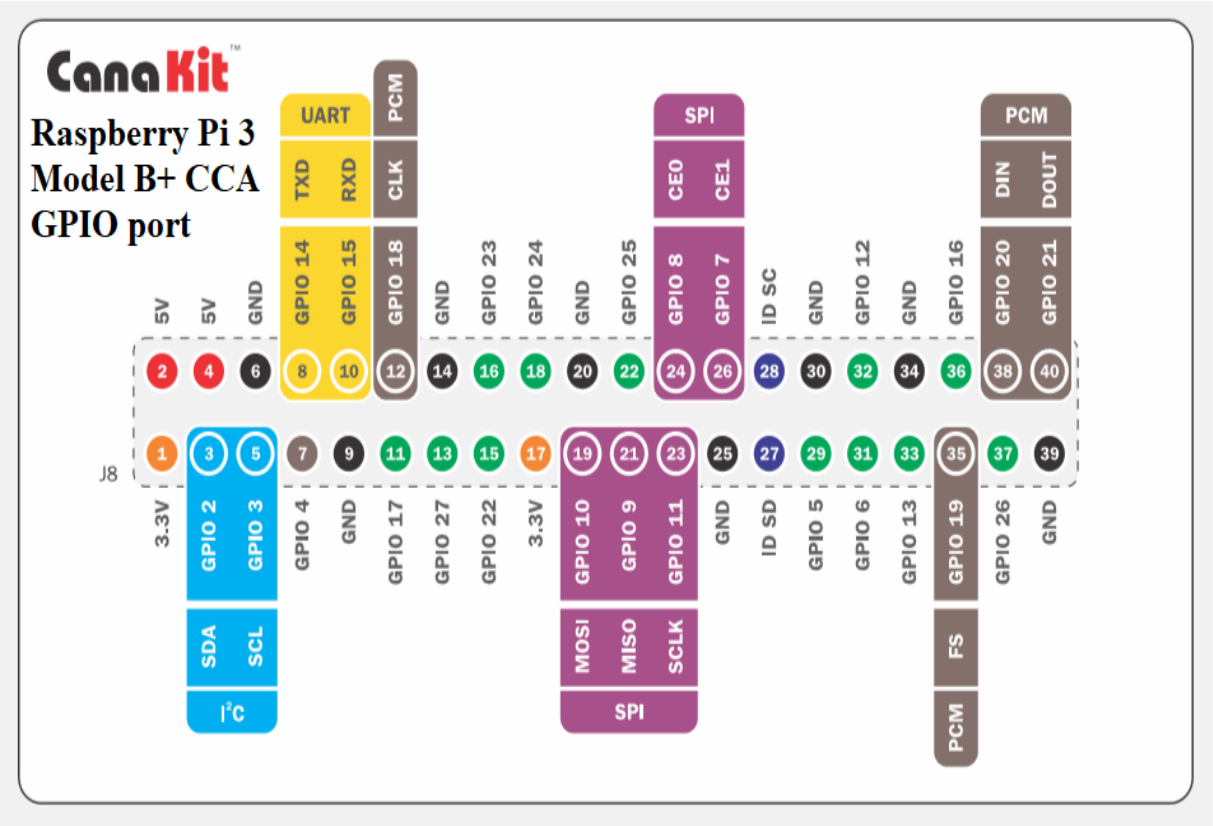


Figure 2. Raspberry Pi CCA GPIO port

9.4) You can but you don't have to use Pin 12 (GPIO 18/CLK/PCM) for driving the LED. You can use some other GPIO pin and figure out how to turn on and turn off the LED with it by running your software.

9.5) Write Python code (as it is shown in the manual example) or some other language code you prefer if you can figure out how to compile it and run it on your Raspberry Pi CCA that will periodically turn on the LED for 3 seconds and then turn off the LED for 3 seconds.

9.6) Write Laboratory report where you will explain the following:

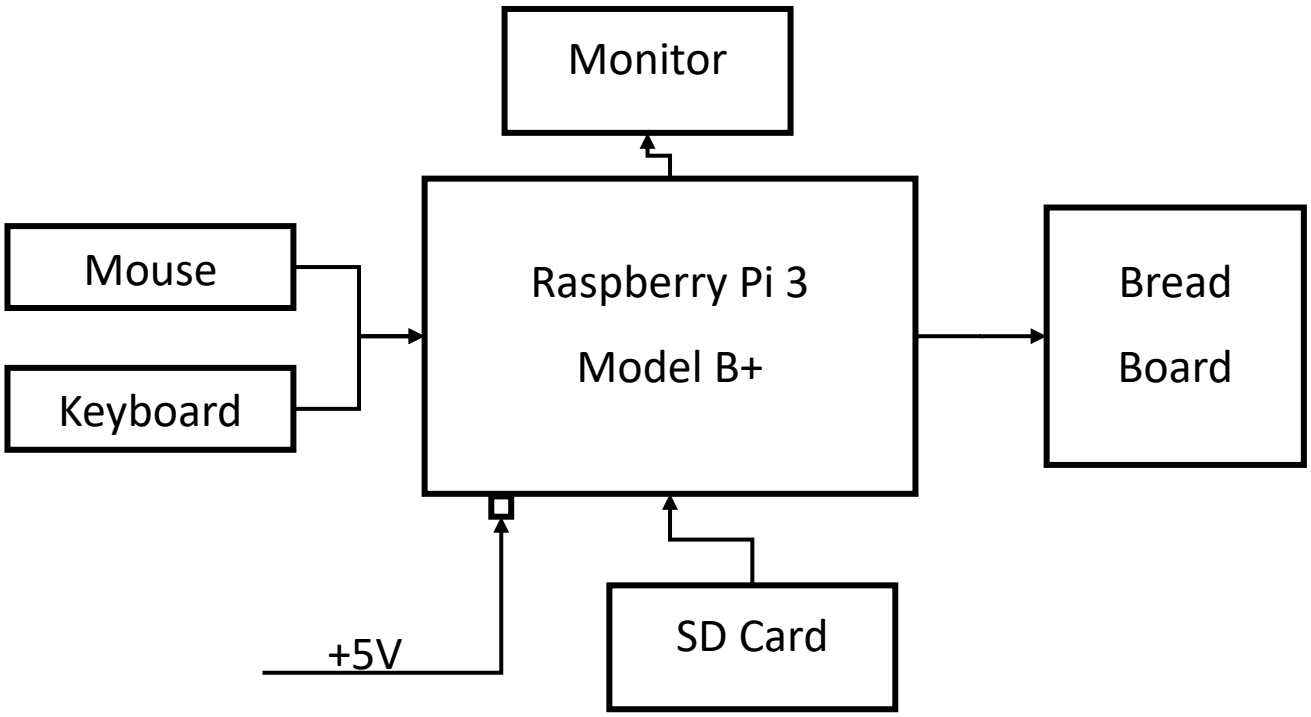
(a) In your hardware setup of the lab report include your embedded system block diagram and show some photos of your setup. You should also explain your entire hardware design integration process.

(b) In your software part of the lab report, include your code and explain entire process of compilation and running of the code on your Raspberry Pi CCA.

(c) Submit short video (up to 30 seconds) where you will show your LED blinking experiment and how you turn on and turn off the LED every 3 seconds.

Byeongchan Gwak, Student ID : 501026

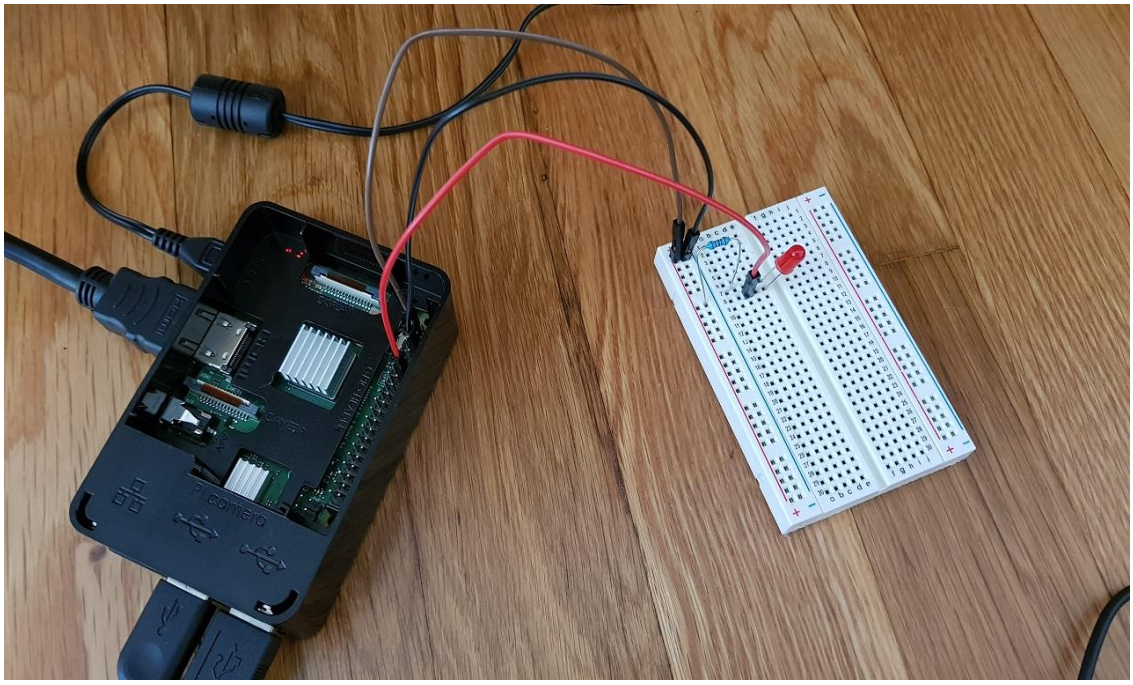
System Block Diagram



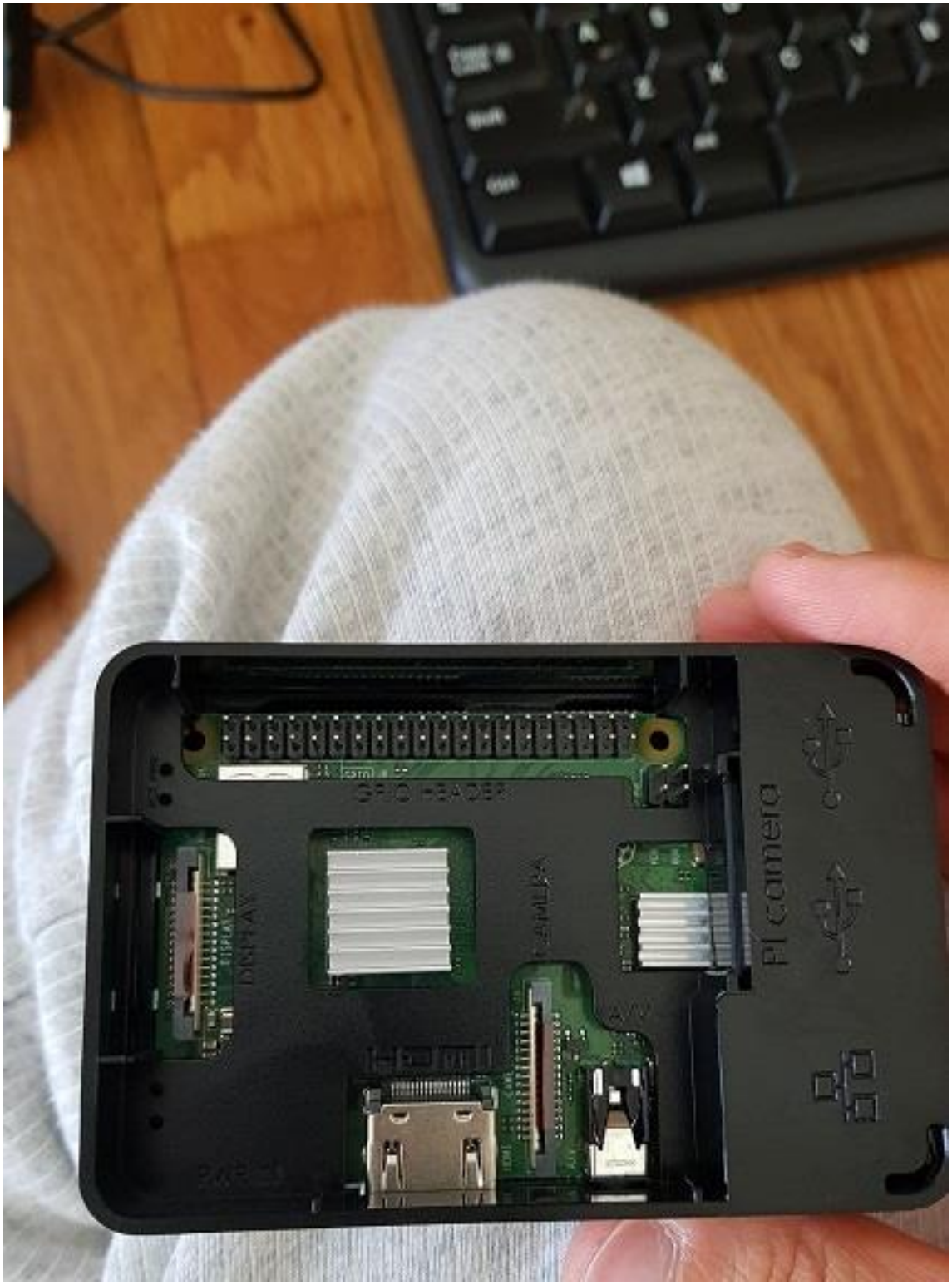
(a) In your hardware setup of the lab report include your embedded system block diagram and show some photos of your setup. You should also explain your entire hardware design integration process.

First, I put my Raspberry Pi into the case and put a SD card into it. And then connect the Pi with a monitor, keyboard, mouse, and power. After connecting the power line and then it started automatically and set up Raspberry OS(which a spinoff of Linux Debian). After checking OS booted successfully, turn off the Pi and connected it with the breadboard. And I put a red LED and a 220 resistor and connect them to a power and a GND. As a guide paper I also connected a LED anode with GPIO 18 pin. It's all about setting up the Raspberry with a breadboard.
Next, I opened a Python IDE on Raspberry OS and I implement some code which sends on and off signal every 3 seconds. I ran the code and it worked.

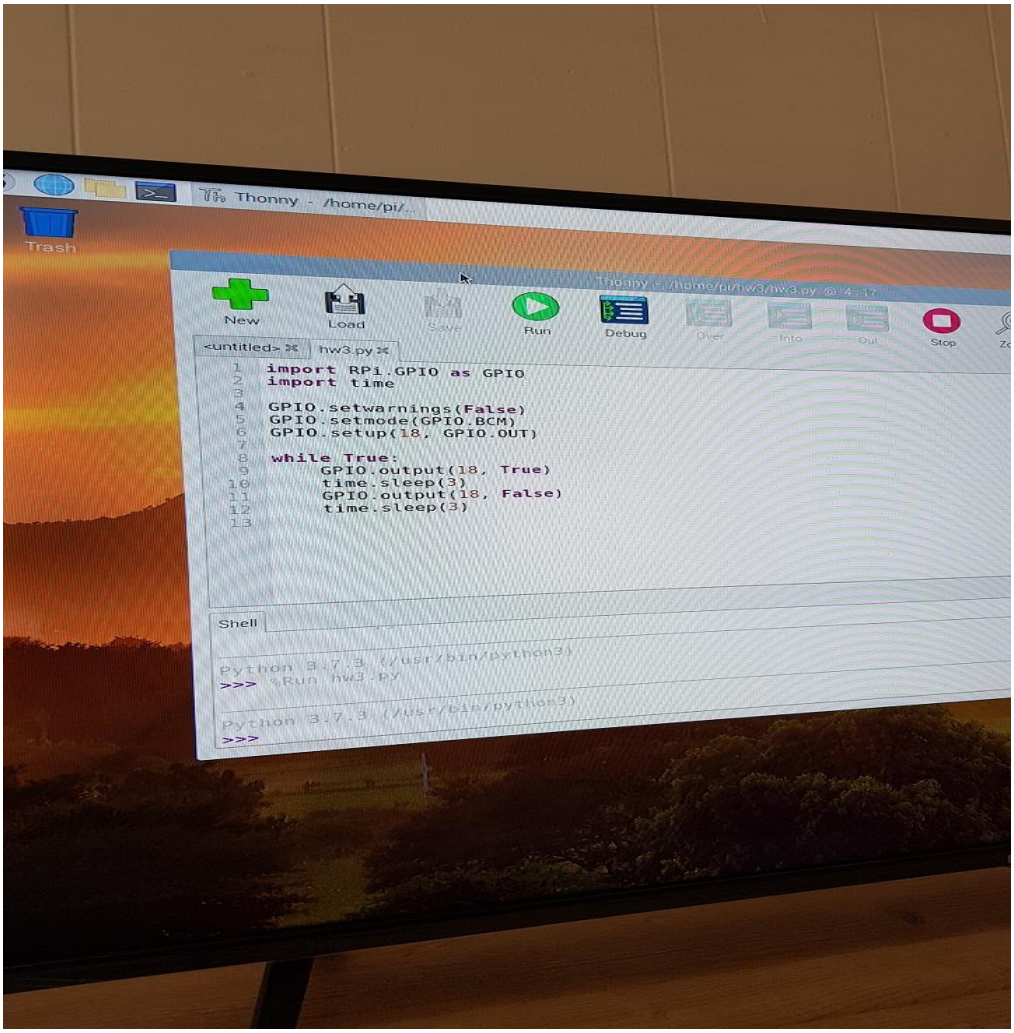
Byeongchan Gwak, Student ID : 501026



Byeongchan Gwak, Student ID : 501026



Byeongchan Gwak, Student ID : 501026



(b) In your software part of the lab report, include your code and explain entire process of compilation and running of the code on your Raspberry Pi CCA.

```
1   # Author: Byeongchan Gwak, 501026
2   import RPi.GPIO as GPIO     # import raspberry GPIO
3   import time                 # import time to call the sleep
4
5   GPIO.setwarnings(False)     # to disable warnings
6   GPIO.setmode(GPIO.BCM)      # set it to BCM mode to refef pin number
7   GPIO.setup(18, GPIO.OUT)    # set 18 pin for output signal
8
9   while True:                 # endless loop
10      GPIO.output(18, True)   # turn on the signal on the 18 pin
11      time.sleep(3)           # then take a sleep for 3 seconds
12      GPIO.output(18, False)  # turn off the signal on the 18 pin
13      time.sleep(3)           # then take a sleep for 3 seconds
14
```

(c) Submit short video (up to 30 seconds) where you will show your LED blinking experiment and how you turn on and turn off the LED every 3 seconds.

I added mp4 file with the same folder. Please see that clip.