American International University- Bangladesh (AIUB)
Faculty of Engineering

Course Name:	Microprocessor and Embedded Systems	**Course Code:**	EEE 4103
Semester:		**Term:**	Mid
Total Marks:	20	**Submission Date:**	Exam Day
Instructor Name:	Md Sajid Hossain	**Assignment:**	02

Course Outcome Mapping with Questions

Item	COs	POIs	K	P	A	Marks	Obtained Marks
Q1	**CO1**	**P.a.4.C3**	**K4**	**P1, P3, P7**		**5**	
Q2	**CO1**	**P.a.4.C3**	**K4**	**P1, P3, P7**		**5**	
Q3	**CO1**	**P.a.4.C3**	**K4**	**P1, P3, P7**		**5**	
Q3	**CO1**	**P.a.4.C3**	**K4**	**P1, P3, P7**		**5**	
					Total:	**20**	

Student Information:

Student Name:	**Kowshik Halder**	**Student ID:**	**21-45408-3**
Section:	**F**	**Department:**	**CSE**

Submission link is in the notice section of the portal.			
Submission Date:		**Due Date:**	**Exam Day**

Marking Rubrics (to be filled by Faculty):

Problem #	Excellent [5]	Proficient [4]	Good [3]	Acceptable [2]	Unacceptable [1]	No Response [0]	Secured Marks
	Detailed unique response explaining the concept properly and answer is correct with all works clearly shown.	Response with no apparent errors and the answer is correct, but explanation is not adequate/unique.	Response shows understanding of the problem, but the final answer may not be correct	Partial problem is solved; response indicates part of the problem was not understood clearly.	Unable to clarify the understanding of the problem and method of the problem solving was not correct	No Response/(Copied/identical submissions will be graded as 0 for all parties concerned)	
1							
2							
3							
4							
Comments						Total marks (20)	

1. In an alternate universe, where live forms with supernatural abilities are common, a top secret research lab is trying to implement a very robust security system against both physical and cyber threats. They have decided to build their physical defenses with an **ATMega328P MCU** at the center. As part of the security measures, the MCU is supposed to generate **two pulses of 80µs and 2s**. The team of engineers cleverly realized, they can easily generate these pulses by configuring the timer modules on the MCU. The MCU is running at 16 MHz and there are 3 timer modules (Timer0, Timer1 and Timer2) available of which Timer0 and Timer2 are 8-bit timers and Timer1 is a 16-bit timer. **The available prescalers are 8, 64, 256 and 1024**. It is apparent that 2 different timers need to be used for this purpose. **Select** 2 from the 3 available timers to be used for this system. Then **compute** the necessary timer counts to generate the stated pulse signals using the selected timers. To leave room for future changes, also **compute** the maximum pulses which can be generated using the selected timers if the MCU clock frequency remains unchanged at 16 MHz. [5]

2. To meet the conditions specified in Q1, at least 1 of the 8-bit timers needs to be used. **Prepare** the necessary registers associated with an 8-bit timer to achieve the time count computed in Q1. Consider the timer to be running in normal mode. [5]

[**Hint:** TCCRxA and TCCRxB have to be initialized. The useful bits of TCCRxA are bit 0 and bit 1 which represent WGMx0 and WGMx1 respectively. Bits 0 to 3 are the useful bits from TCCRxB. Bits 3 represent WGMx2. Bits 0 to 2 represent the clock select functions CSx0, CSx1, CSx2 respectively. Consider WGM = 0 for normal mode of operation. The bits not mentioned here can be ignored/considered as 0]

CSx2	CSx1	CSx0	Prescaler
0	1	0	8
0	1	1	64
1	0	0	256
1	1	1	1024

3. A vending machine at a university campus will offer three different options for cold drinks. This machine will be able to dispense a maximum of 3 drinks at once. The machine will have a total of three buttons (button 1 for 1 drink, button 2 for 2 drinks, and button 3 for 3 drinks) to select the number of drinks to be dispensed. After this initial selection, the vending machine will wait for 3s and then display the options for available drinks. The consumer can use the 3 buttons (button 1 for A drink, button 2 for B drink, and button 3 for C drink) to select their desired drink(s). If only 1 drink is to be dispensed, the consumer will press the relevant button and the vending machine will dispense the chosen drink after a 3 s wait. If more than 1 drink is to be dispensed, the relevant buttons need to be pressed one after another and the machine will dispense the drinks one at a time with a 3 s gap between the consecutive drinks. This vending machine is to be built around an Arduino Uno platform. Now, **prepare** a program to execute the mentioned tasks if the Arduino Uno system is operating at 16 MHz. [For this program, digital I/O pins D0 to D9 can be used whose pin numbers are 0 to 9 correspondingly.] [5]

4. Prepare a flowchart for the program in Q3 to show the flow of logic. [5]

Q1

frequency = 16×10^6 Hz

Prescaler = 1024

8 bit Timer,

$$\text{Maximum possible Delay} = (\text{TimerCount} + 1) \times \frac{\text{Prescaler}}{\text{frequency}}$$

$$= (255 + 1) \times \frac{1024}{16 \times 10^6}$$

$$= 16 \text{ millisecond}$$

16 bit Timer,

$$\text{Maximum possible Delay} = (65535 + 1) \times \frac{1024}{16 \times 10^6}$$

$$= 4.194 \text{ sec.}$$

To generate 80μs Delay, Timer0 is selected.

prescaler = 256

$$\text{TimerCount} = 80 \times 10^{-6} \times \frac{16 \times 10^6}{256} - 1$$

$$= 4$$

To generate 2s Delay, timer1 is selected.

Prescaler = 1024

$$\text{Timer count} = 2 \times \frac{16 \times 10^6}{1024} - 1$$

$$= 31249$$

Maximum pulse possible with timer0 = $2^8 = 256$

Maximum pulse possible with timer1 = $2^{16} = 65536$

Q2

Prescaler = 256.

8 bit timer (timer0),

						WGM01	WGM00
0	0	0	0	0	0	0	0

TCCR0A

				WGM02	CS02	CS01	CS00
0	0	0	0	0	1	0	0

TCCR0B

TCCR0A = 0b00000000.

TCCR0B = 0b00000100.

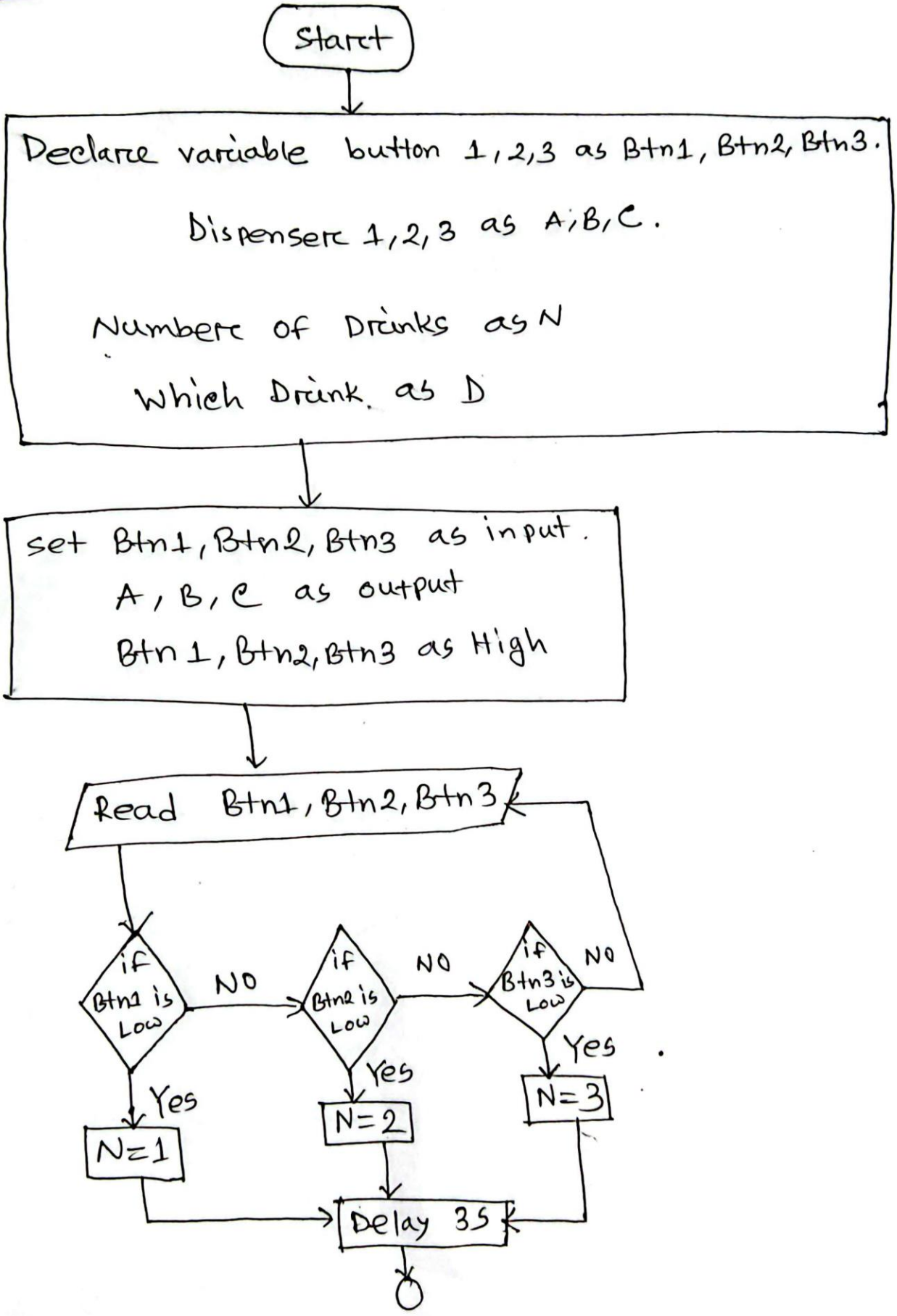
Q3:

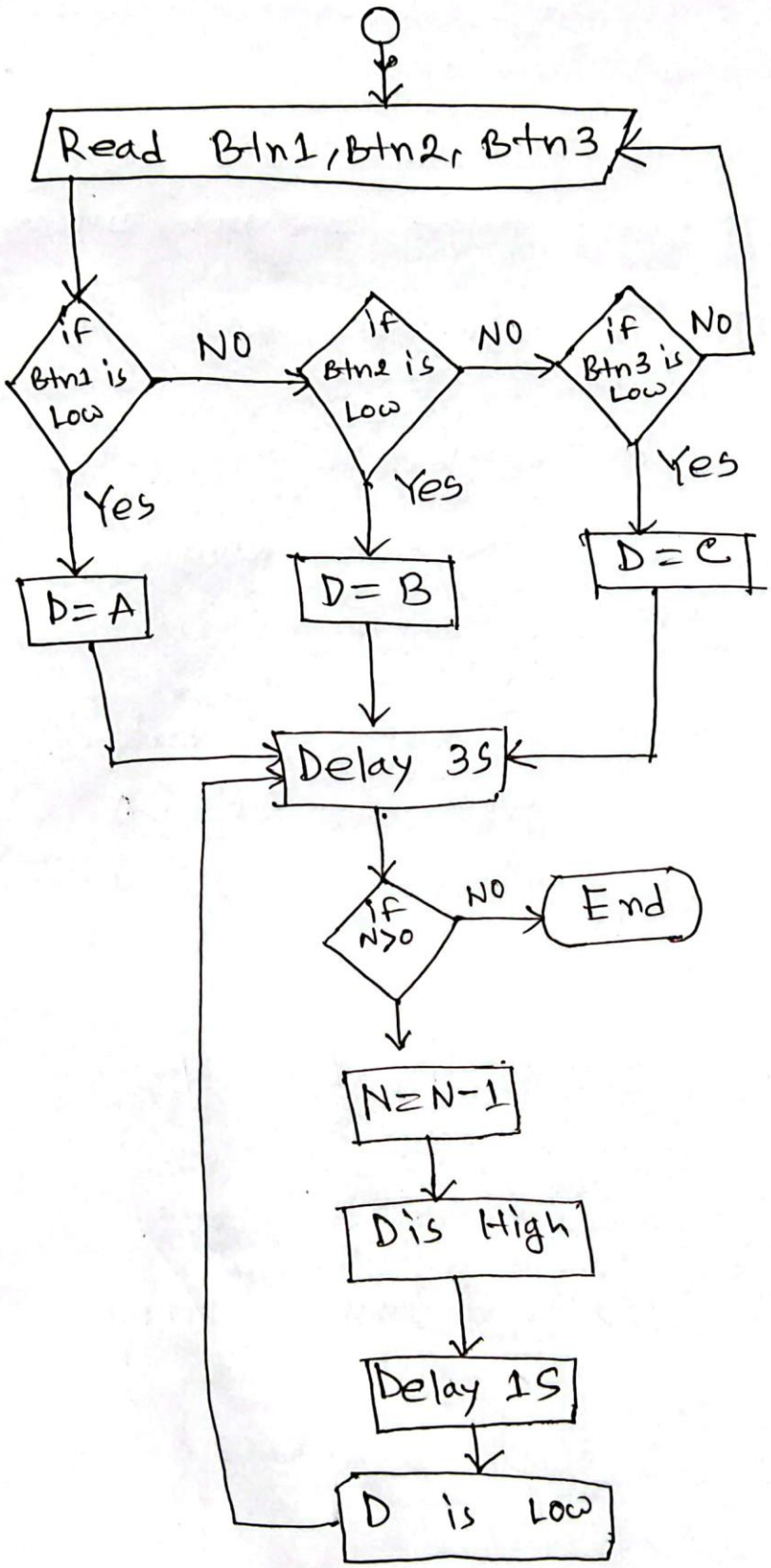
```
#define btn1 2;
#define btn2 3;
#define btn3 4;
#define drinkA 5;
#define drinkB 6;
#define drinkC 7;

void setup() {
  pinMode(btn1, INPUT);
  pinMode(btn2, INPUT);
  pinMode(btn3, INPUT);
  pinMode(drinkA, OUTPUT);
  pinMode(drinkB, OUTPUT);
  pinMode(drinkC, OUTPUT);
  digitalWrite(btn1, HIGH);
  digitalWrite(btn2, HIGH);
  digitalWrite(btn3, HIGH);
}
void loop() {
  //code for selection number of drink
  int N = 0;
  int drinkNum = 0;
  while (N == 0) {
    if (digitalRead(btn1) == LOW) {
        N = 1;
      }
    else if (digitalRead(btn2) == LOW) {
        N = 2;
      }
    else if (digitalRead(btn3) == LOW) {
        N = 3;
      }
  }
  // code for selection of the desired drinks
  While(drinkNum == 0) {
    if (digitalRead(btn1) == LOW) {
      drinkNum = 5;
    } else if (digitalRead(btn2) == LOW) {
      drinkNum = 6;
    } else if (digitalRead(btn2) == LOW) {
      drinkNum = 7;
    }
  }
  for (int i = 0; i < N; i++) {
    delay(3000);
    dispenseDrink(drinkNum);
```

```
  }
}
void dispenseDrink(int drinkPin) {
  digitalWrite(drinkPin, HIGH);
  delay(1000);
  digitalWrite(drinkPin, LOW);
}
```

Q4


Start
Declare variable button 1,2,3 as Btn1, Btn2, Btn3.
Dispenser 1,2,3 as A;B,C.
Number of Drinks as N
Which Drink as D
set Btn1, Btn2, Btn3 as input.
A, B, C as output
Btn 1, Btn2, Btn3 as High
Read Btn1, Btn2, Btn3
if Btn1 is Low
NO
if Btn2 is Low
NO
if Btn3 is Low
NO
Yes
N=1
Yes
N=2
Yes
N=3
Delay 35


Read Btn1, Btn2, Btn3
if Btn1 is Low
NO
if Btn2 is Low
NO
if Btn3 is Low
No
Yes
Yes
Yes
D=A
D=B
D=C
Delay 35
if N>0
NO
End
N=N-1
Dis High
Delay 15
D is Low