**Mid Term Assignment**

**NAME: MINHAZ AHMED AYON**

**ID: 20-41859-1**

##### SECTION: A COURSE: MICROPROCESSOR AND EMBEDDED SYSTEMS FACULTY: TAHMIDA ISLAM

**1)** Biman Bangladesh operates a flight from Dhaka to London. Usually, the flight from London

starts at 8pm local time (after the sun has set). Even though the commercial aircrafts are

painted white, it doesn’t help when it comes to other aircrafts’ visibility. So, to prevent

accidents/collisions, aircrafts usually mount 1 lights at the bottom surface of the body, 2

lights on 2 wings (left and right) and 1 light on the tail and, these 4 lights flash together

periodically. Biman’s flight experienced an issue with the program controlling these lights

and upon investigating; it was found that the system is built with an Arduino Uno at the

heart. Interestingly, the Arduino was already set up to consume as little power as possible not

to strain the aircraft’s fuel efficiency. Now, **prepare** a program in Arduino Uno platform to

control the 4 lights mounted on the aircraft’s body so that all the lights flash together every

2s.

**ANS:**

#define Led1 3

#define Led2 4

#define Led3 5

#define Led4 6

void setup()

{

pinMode(Led1, OUTPUT);

pinMode(Led2, OUTPUT);

pinMode(Led3, OUTPUT);

pinMode(Led4, OUTPUT);

}

void loop()

{

digitalWrite(Led1, !digitalRead(Led1));

digitalWrite(Led2, !digitalRead(Led2));

digitalWrite(Led3, !digitalRead(Led3));

digitalWrite(Led4, !digitalRead(Led4));

delay(2000);

}

**2) Prepare** a flowchart to explain the flow of logic for the program in Q1 so that Biman

engineers can modify and perfect the program easily when the aircraft is taken to

maintenance after completing this flight.

START

Define Led1, Led2, Led3, Led4

DigitalWrite, digitalRead,  
delay (2000)

Display Output  
LED blinks

START

**3)** The steamer ‘Ostrich’ ferries people between Dhaka and Barisal. Ostrich usually starts from

Dhaka around 7.30pm (evening/night depending on the season). At night, the visibility is

usually very low. So, the steamers/launches usually have a flashing signal light on the roof.

On top of that, the master of the steamer also honks the horn when necessary. This signal

light and the horn are both controlled by an Arduino Uno. The signal light flashes every 4s

whereas the horn is sounded whenever the master presses the horn switch. The horn switch

already has a small RC circuit attached to it to counter bouncing. Now, **prepare** a program in

Arduino Uno to control the signal light and the horn considering the information mentioned

above.

**ANS:**

#define Led1 3

#define switch1 2

int LED\_blink = 4000;

int switch\_read;

int delay\_timer (int miliseconds)

{

int count = 0;

while(1)

{

if(TCNT0 >= 16)

{

TCNT0=0;

count++;

if (count == miliseconds)

{

count=0;

break;

}

}

}

return 0;

}

void setup()

{

pinMode(Led1, OUTPUT);

pinMode(switch1, INPUT);

TCCR0A = 0b00000000;

TCCR0B = 0b00000101;

TCNT0=0;

}

void loop() {

digitalWrite(Led1, !digitalRead(Led1));

delay\_timer(LED\_blink);

switch\_read=digitalRead(switch1);

if (switch\_read==LOW)

{

;play HORN

}

}

**4)** It’s the year 2000 and space shuttles are yet to be retired. NASA astronaut Anakin Skywalker

aboard the space shuttle wanted to spend some time solving short mathematical problems.

The problem he had on hand was:

**(AB + B2)2 – 4A2 + 9AB,** when A = 2 and B = 5

So, he turned to the computer system of the shuttle to solve the equation. The shuttle uses an

8086 system. Now, to help Anakin, prepare a program in 8086 assembly language to

**compute** the result of the given equation and store the final result in BX register.

**ANS:**

**(AB + B2)2 – 4A2 + 9AB,** when A = 2 and B = 5

MOV AX, 2

MOV BX, 5

MOV CX, AX

MOV DX, BX

ADD AX, BX || AX= AX+BX

MOV BX, AX

MUL BX || BX = AX \* BX

SUB BX, CX

MOV CX, 9

MUL CX || BX = CX\*BX

ADD BX, CX || BX = CX + BX

MOV AX, BX

MOV BX, AX

MUL BX || BX = AX\*BX

SUB BX, CX || BX = CX + BX

MOV CX, 4

MUL CX || BX = CX\*BX

ADD BX, CX || BX = CX + BX

MOV AX, BX

MOV BX, AX

MUL BX || BX = AX\*BX

SUB BX, CX || BX = CX - BX

MOV CX, 9

MUL CX || BX = CX\*BX

ADD BX, CX || BX = CX + BX

MOV AX, BX

MOV BX, AX

MUL BX || BX = AX\*BX

SUB BX, CX || BX = CX - BX

MOV CX, 4

MUL CX || BX = CX\*BX

ADD BX, CX

MOV AX, BX

MOV BX, AX

MUL BX || BX = AX\*BX

SUB BX, CX



