MMME3085 example solutions January Exam 2021-22

Section A

See mark sheet for mark scheme

```
// Example solution for Q1
//

#include <stdio.h>
#include <stdlib.h>

struct TempData
{
    int time;
    float temperature;
};

int main()
{
   FILE *finput;
    char FileName[50];
    struct TempData *TemperatureData;

    // User input for filename
    printf ("Please enter the name of the file to open ");
    gets(FileName);

    // Check file opened successfully
    if ( ( finput = fopen(FileName,"r")) == NULL )
    {
        printf ("ERROR: File cannot be opened, code will now terminate");
        exit (0);
    }

    // Read first item from file giving number of time/temp pairs
    int numEntries;
    fscanf( finput, "%d", &numEntries);
```

```
    // Allocate memory for data, terminate if unsuccessful
    if ( (TemperatureData = calloc( numEntries, sizeof( struct TempData))) == NULL )
    {
        printf("ERROR: Could not allocate memory for data, code will now terminate");
        fclose(finput);
        exit(0);
    }

    // Output header
    printf("Time/s  Temperature\n");

    // Iterate through data, reading from file and sending formatted output to console
    int i;
    for( i = 0; i < numEntries; i++)
    {
        fscanf( finput, "%d", &TemperatureData[i].time);
        fscanf( finput, "%f", &TemperatureData[i].temperature);
        printf( "%-8d%-8.2f\n", TemperatureData[i].time, TemperatureData[i].temperature);
    }

    // Tidy up
    free(TemperatureData);
    fclose(finput);
    return 0;
}
```

Output:

```
Please enter the name of the file to open TemperatureData.txt

Time/s  Temperature
0       12.34
1       13.57
2       14.36
3       14.37
4       15.69
5       17.49
6       19.49
7       20.43
8       21.76
9       20.40
10      19.44
11      18.74
```

```
// Solution for Q2
//


#include <stdio.h>

#include <stdlib.h>


int CalcKineticEnergy( float mass, float vel, float *KEnergy );


int main()

{

    float mass = 0, velocity = 0;

    float KineticE = 0;


    // User input of mass and velocity

    printf( "Input values for mass and velocity: ");

    scanf("%f %f", &mass, &velocity);

    // Check if valid return value and print message accordingly

    if ( CalcKineticEnergy( mass, velocity, &KineticE ))

        printf( "The kinetic energy of a car of mass %.2fkg, velocity %.2fm/s is %.2fkJ", mass,
velocity, KineticE/1000 );

    else

        printf( "ERROR: invalid values entered");

    return 0;

}


int CalcKineticEnergy( float mass, float vel, float *KEnergy )

{

    // Return false if invalid mass

    if ( mass <= 0 )

        return 0;

    // Calculate kinetic energy

    *KEnergy = 0.5 * mass * vel * vel;

    return 1;

}
```

Sample output:

Input values for mass and velocity: 1000 25

The kinetic energy of a car of mass 1000.00kg, velocity 25.00m/s is 312.50kJ

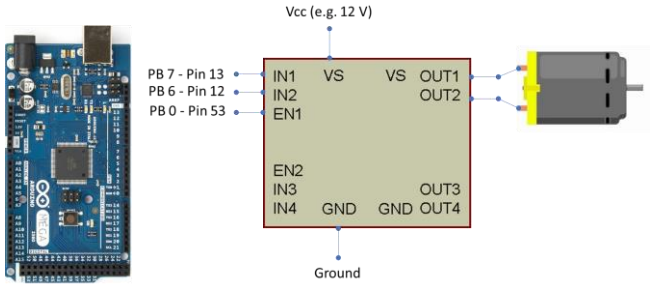
Section B

Solution for Q3

(a) The minimum change in the measured angle that can be detected by the system is $360/(2^8)=1.4^o$ [2 Marks]. As the output will be in the range of 0-360 with integer output only, three 7-segemnt displays will be enough to show the system output without the decimal point [1 Mark].

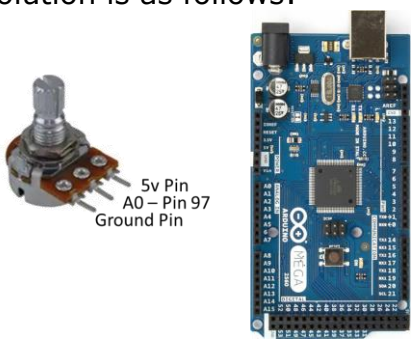
(b)

i. One possible solution is as follows:



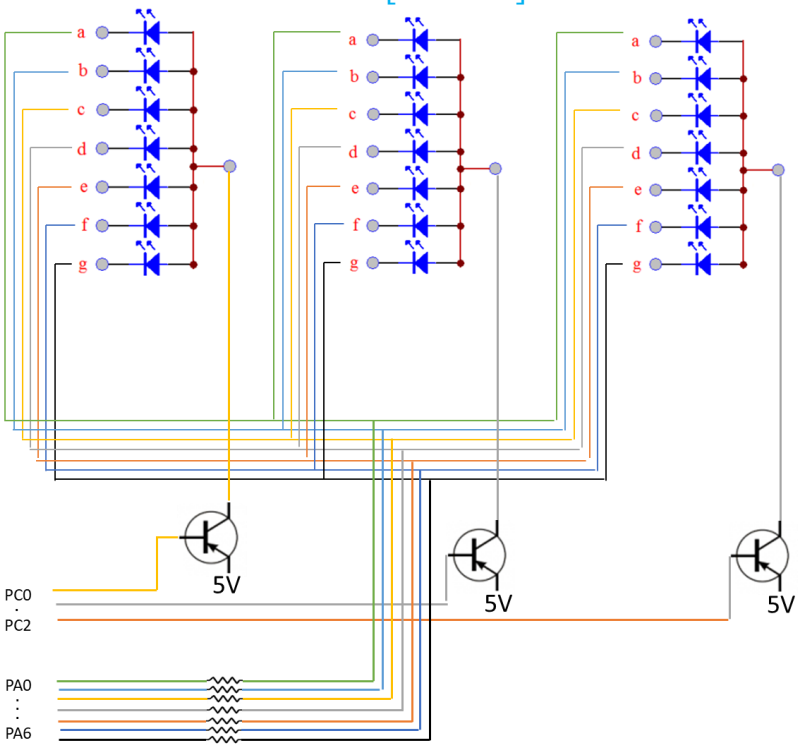
The key point here is to select the PWM pins for Input-1 and Input-2. [3 Marks]

ii. A possible solution is as follows:

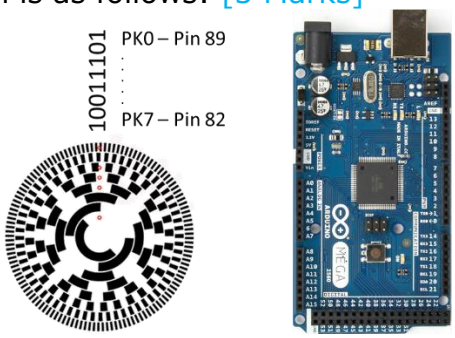


The key point here is to connect the output to a an analoge input pin. [2 Marks]

iii. A possible solution is as follows: [5 Marks]



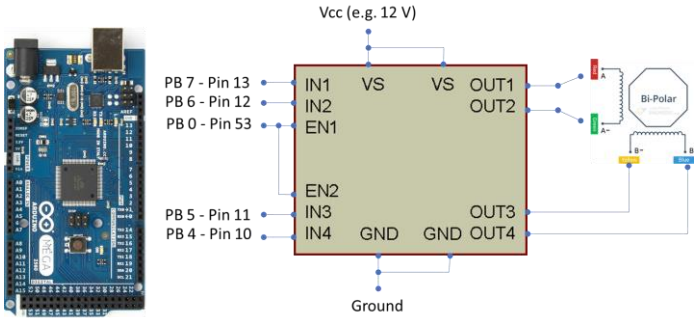
iv. A possible solution is as follows: [3 Marks]



(c) After connecting the items to the Arduino as shown above, the following steps need to be executed:

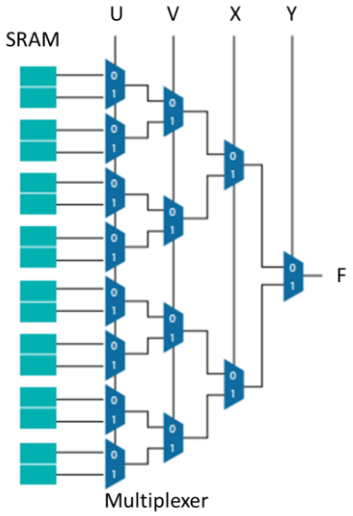
- Define the digital input pins/ports: Port K. [0.5 Marks]
- Define the digital output pins/ports: All the other used pins/ports are outputs. [0.5 Marks]
- Read the inputs from the Port K. [1 Marks]
- Use a look-up table to convert Port K data from gray code to decimal (the rang is 0-255). [1 Mark]
- Read the analoge input from the potentiometer, the input range (0-1023). [1 Mark]
- Compare the reading from the encoder with the input from the potentiometer (after scaling it down). [1 Mark]
- If the difference is positive, drive the motor (e.g., forward by sending High to enable and High/PWM to Input-1) and vice versa. If the difference is 0, stop the motor. [1 Mark]
- While doing that, scale the reading from the encoder (0-255) to (0-360) and show them on the 7-segment displays as follows:
  - Enable the first 7-segment display by setting PC0 to high while disabling the others by setting PC1 and PC2 to low. [1 Mark]
  - Divide the scaled-up output from the encoder (i.e., 0-360) by 100 to get the most significant digit and show it (e.g., to show 256°, 126/100 as integer returns 2, then send to Port A 0x24 (PORTA=0x24;) and Port C 0x01(PORTC=0x01;)) to all the displays. [1 Mark]
  - Wait for very short time (few milliseconds) [1 Mark]
  - Enable the second 7-segment display by setting PC1 to high while disabling the others. [1 Mark]
  - To calculate the second digit, subtract 2*100 from 256 and divide the result by 10 as integer. [1 Mark]
  - Send the corresponding digital combination to show the second digit to all the displays. [1 Mark]
  - Wait for very short time (few milliseconds)
  - Repeat the above sequence for all displays and start from beginning again and again. [1 Mark]

(d) The connection is as follows: [5 Marks]



Solution for Q4

(a) The schematic is as follows: [8 Marks]



(b) The values for the SRAM are in Yellow in the following table: [2 Mark each]

Inputs				Output
U	V	X	Y	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	0
1	1	1	1	0

END