UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 6 Examination in Engineering: November 2022

Module Number: EE6304 **Module Name: Embedded Systems Design – N/C**

[Three Hours]

[Answer **all questions**, each question carries 10 marks]

Q1 a) i) Storage is an important part of an embedded system. State the types of storage used in a microcontroller to store program code, program data, and runtime state, respectively.

[1 Mark]

ii) State in point form the difference between compiler and linker.

[1 Mark]

iii) Draw a block diagram of a typical embedded system.

[1 Mark]

b) i) Briefly explain in point form how an IDE is beneficial in embedded systems programming than using a text editor with a separate compiler.

[1 Mark]

ii) List two advantages of using a microcontroller in an embedded system than using discrete logic-based circuits.

[1 Mark]

iii) Briefly explain in point form how an interrupt vector is used in a microcontroller.

[1 Mark]

c) Suppose that you are requested to design a device to record the environmental temperature against the GPS location and time upon a press of a button. Then, the device is supposed to transmit these data to a central server.

i) State two functional requirements and two quality requirements for this device.

[2 Marks]

ii) State two challenges you may come across in designing the device.

[1 Mark]

iii) Suppose that the data collection, data storing, and data transmission are carried out using three tasks in the given order. If the real-time operating system running in the device supports contest switching and mutli-tasking, briefly explain in point form how would you guarantee that the given three tasks are executed in the given order.

[1 Mark]

Q2 a) Figure Q2a on page 6 shows a part of a C code written to implement a portable temperature logger device implemented using an ATmega 2560 or PIC16F877A microcontroller. It is supposed to read and record a data point upon press of a button attached to PORTA pin 1. Each data point consists of three parameters, namely, temperature, GPS location, and time. After recording a predefined number of data points, the device transmits a set of data points to a central server.

The header file lcd.h contains the functions required to work with the LCD display attached to the device. The displayTemperature() function makes use of the library functions defined in lcd.h. Functionality of the functions given in the function prototypes are given in Table Q2.

Table Q2

Function	Description
`readGPS`	Read the current GPS location into a predefined variable passed in to it.
`readTime`	Read the current time into a predefined variable passed in to it.
`readTemperature`	Read the current temperature and return as a double
`storeData`	Store the data GPS location, time and the temperature passed into the function (store a data point passed into the function)
`configureIOPorts`	Set the data directions of the I/O ports
`transmitData`	Transmit the past N data points to the to a central server where N is passed into the function as a parameter
`displayCharacter`	Displays the character corresponding the ASCII value passed in as a parameter
`getOne`	Obtain the value of ones position of a number passed in
`getTens`	Obtain the value of tens position of a number passed in

i) Complete declaration of the variable "currentTime" in line 25.
[0.5 Marks]

ii) Complete the condition in line 29 to check if the button in pressed if the push button is connected as "Active high".
[1 Mark]

iii) Complete the function calls readGPS in line 30.
[0.5 Marks]

iv) Complete the function prototype of storeData in line 17.
[1 Mark]

v) Complete the condition in line 36 and the transmitData function call in line 37.
[2 Marks]

b) Briefly explain the BCF and BSF instructions given in Figure Q2b in page 7.
[2 Mark]

c) Explain in point form, the functionality or write a simple pseudocode equivalent to the following assembly code. Refer to the list of assembly instructions given in Figure Q2b in page 7 if required.

```
START   MOVLW  0x1B
        MOVWF  0x21
REPEAT  INCF   0x21
        INCFSZ 0x21, 0x01
        GOTO   REPEAT
        END
```

[3 Mark]

Q3 Consider a portable temperature logger device implemented using a hypothetical microcontroller which is supposed to read and record data points upon press of a push button. Suppose that it is required to carry out a functional upgrade to log data points every 5 minutes. The microcontroller consists of a 16-bit and 32-bit timers attached to a prescalar circuit which can provide prescaling of magnitude 1, 8, 64, and 256. The timer circuit is similar to that of PIC16F877A or ATmega 2560, other than the differences mentioned above.

a) i) State the two most common types of interrupts associated with a timer.

[1 Mark]

ii) Identify the timer and the prescalar that can be used for this purpose. Show the calculations and give reasons in point form.

[2 Marks]

iii) Calculate the value that should be set at timer compare register

[1 Mark]

b) Suppose that you have to use a counter which counts down from 5 minutes to 0 and the counter value should be displayed at each second. Reading and recording of data points are supposed to be done when the counter value reaches 0. After the counter reaches 0, it should start over, counting down from 5 minutes. Briefly explain how you would achieve this using only one timer. Answer should contain timer interrupt duration and the mechanism to be used. You may use explanations in point form, code segments, and/or flow charts when necessary.
Note: You do not need to explain the timer identification process or anything about reading and recording of data points.

[2 Marks]

c) Suppose that the counting down process should stop and reset to 5 minutes when the push button is pressed. Explain how this is achieved using a pseudo code, C code or a flow chart along with necessary explanations.

[2 Marks]

d) Suppose that there are three modes of operation in this device based on how the reading and recording of data is triggered, namely, Automatic, Manual, and Mixed modes. Information of the three modes is given in Table Q3.

Mode selection is carried out using three switches attached to an input port. Write a short pseudo code or a C code segment that can be used to carry out the switching between the modes. State any assumptions.
Note: You can consider part d) to be independent from the rest of the question.

Table Q3

Mode	Description	Function Used
Automatic	Reading and recording of data will be triggered by the timer	automaticMode()
Manual	Reading and recording of data will be triggered by the push button	manualMode()
Mixed	Reading and recording of data will be triggered by both the timer and push button	mixedMode()

[2 Marks]

Q4 A portable RFID based scanner is designed for a supermarket where RFID tags attached to the goods are passive tags and the device consists of an RFID antenna to read the tags. Scanner device consists of separate units to read the RFID tag, process data, communication, and to display the required information. Functionality of the device is as follows.

I. The scanner starts its operation when the 'Scan' push button is pressed and if there is no RFID tag is read for 10 seconds, the reader unit turns itself off until the 'Scan' push button is pressed again.
II. When a RFID tag is read, the device processes the data to obtain the tag ID and sends it to the central server.
III. The server retrieves product name and the number of items of the same kind available at the stores and sends that information to the scanner device.
IV. The scanner device displays the product name and the remaining number of items at its display.

a) Identify the inputs to the device and outputs from the device.

[2 Marks]

b) i) Draw a block diagram for the device.

[1 Mark]

ii) Draw a flow chart for the operation of the device.
Note: you can omit the operation of the central server.

[1 Mark]

c) It is required to implement a controller using which a user can change the range of the RFID antenna. Explain in point form how you would implement this controller.

[2 Marks]

d) If the LCD display is said to be connected to the scanner device using a serial communication, name three possible interfaces for this connection. Using a wiring diagram show how the display and the main device is connected.

[2 Marks]

e) State two possible technologies to implement the communication between the device and the central server.

[1 Mark]

f) Suppose an RFID tag length is 32 bits and it is stored in a packet which is of the format given in Figure Q4. Tag data received from the reader to the processor in 5 bytes namely byte1, byte2, byte3, byte4, and byte 5, respectively where Header bits are in byte1 and stop bits are in byte 5.

Header (4-bit)	Tag ID (32bit)	Parity (1 bit)	Stop bits (3 bits)

Figure Q4

Explain how you would obtain the tag ID into to a 4-byte integer variable TagID. You may give the equations.

[1 Mark]

```
1. #define F_CPU 16000000UL
2. #include <stdio.h>
3. #include "lcd.h"
4. #define TX_DATA_POINTS 10
5. struct GPS{
6.   double Lat;
7.   double Lon;
8. }
9. struct Time{
10.   int Hour;
11.   int Minute;
12.   int Second;
13. }
14. void readGPS(struct GPS*);   //Function to read the GPS location
15. void readTime(struct Time*); //Function to read time
16. double readTemperature(void); //Function to read the temperature
17. void storeData(.....struct Time, .....); //Function to store data
18. void configureIOPorts(void); //Function to configure IO ports
19. void transmitData(int);      // Function to transmit last few data points
20. void displayTemperature(double); // Display the temperature in LCD
21. int main(void)
22. {
23.   int index = 0; // Defining variable to store the count number of data points
24.   struct GPS GPSlocation;     // Defining variable for GPS location
25.   ..... currentTime;   // Defining variable for current time
26.   double currentTemperature; // Defining variable for current temperature
27.   configureIOPorts();        // Configure the IO ports
28.   while (1){
29.     if (.....){          // Check if the button is pressed
30.       readGPS(.....);                    // read the GPS location
31.       currentTemperature = readTemperature();  // read the temperature
32.       readTime(&currentTime);                  // read the time
33.       index++;                                 //Increment the index
34.       storeData(GPSlocation, currentTime, currentTemperature)//Store data point
35.       displayTemperature(currentTemperature);  //Display the temperature
36.       if (.....){ // if # of data points to Tx reached limit
37.         transmitData(.....); //Transmit last few data points
38.       }
39.     }
40.   }
41.   return 0;
42. }
43. void displayTemperature(double temp){
44.    int onesTemp = getOnes(temp);  // Ones value of the temperature
45.    int tensTemp = getTens(temp);  // Tens value of the temperature
46.    clearDisplay();                // Clear the LCD display
47.    displayCharacter('T'); displayCharacter('e');
48.    displayCharacter('m');displayCharacter('p'); displayCharacter('=');
49.    displayCharacter(0x30+ tensTemp); displayCharacter(0x30+ onesTemp); //
Display the temperature as an integer
51.    displayCharacter(' '); displayCharacter('C');
52. }
```

Figure Q2a

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add Literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND Literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to Address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR Literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move Literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from Interrupt	2	00	0000	0000	1001		
RETLW	k	Return with Literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from Literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR Literal with W	1	11	1010	kkkk	kkkk	Z	

Figure Q2b