UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER II
SESSION 2010/2011

COURSE	:	MICROCONTROLLER APPLICATION
COURSE CODE	:	BER4223
PROGRAMME	:	BEE
EXAMINATION DATE	:	APRIL / MEI 2011
DURATION	:	3 HOURS
INSTRUCTION	:	ANSWER ALL QUESTIONS

THIS PAPER CONSISTS OF EIGHT (8) PAGES

All questions are based on a differential wheeled mobile robotics system. The system architecture and a part of programming code are given in Figure 1 and Appendix 1 respectively.

Q1 The mobile robot is controlled by PIC18F4550 with Fosc = 10MHz. This mobile robot needs two 9VDC motors to drive left and right wheels that are controlled by using pulse width modulation (PWM) at microcontroller's CCP pins. Its navigation depends on two right analog IR sensors to keep at certain distance to the wall and another IR sensor at front is used to identify the obstacles. These sensors use the same power supply as microcontroller as well as their voltage references. An alphanumeric LCD is connected to an input/output (I/O) expander of MCP23x17. Two buttons have been assigned and connected to RB2 and RB3. The related components involved are given in Figure Q1.

(a) Design a schematic of circuit for all related components to microcontroller. Choose either MCP23S17 or MCP23017 model as for I/O expander.

(6 marks)

(b) Based on your design, explain the proposed input/output interface signal between microcontroller and other devices.

(6 marks)

(c) Referring to Appendix 1.

(i) Develop a flow chart for program given.

(ii) Explain the operation of ***void main(void)***.

(8 marks)

Q2 The IR Range Finder sensor has capability to measure distance from 4cm to 30cm with scaling factor 11.56cm/V.

(a) Based on your design in Q1(a), determine the fastest conversion (T_{AD}) and acquisition (T_{ACQ}) times for each analog channel.

(4 marks)

(b) Complete a subroutine of ***void ADC_setup(void)*** to configure the ADC peripheral.

(3 marks)

(c) Write a subroutine of ***unsigned int Data_Distance(unsigned char position, unsigned char channel)*** by following sequence:

[1] Set channel
[2] Get result into dimension unit cm.
[3] Convert to decimal and ASCII
[4] Display to LCD by referring to unsigned char position.
[5] return result to main application

(13 marks)

Q3 Both motors are controlled with the same period.

(a) Determine the proper setting to create PWM signal with frequency 5KHz and write a code for ***void Motor_setup(void).***

(10 marks)

(b) Complete the following subroutine for motor control:

(i) ***void Robot_Forward(unsigned int DutyCycle_Motor_L,unsigned int DutyCycle_Motor_R)***

(ii) ***void Robot_Stop(void)***

(10 marks)

Q4 (a) Describe the write operation in your serial communication protocol interface for I/O expander.

(2 marks)

(b) Determine proper setting and complete a code for ***void IO_Expander_setup(void)***.

(6 marks)

(c) Write a subroutine to handle the data writing as for ***void IO_Expander_write(unsigned char opcode, unsigned char reg, unsigned char data)***

(12 marks)

Q5 (a) Referring to ***void msDelay(unsigned int iTime)*** subroutine, it will produce 1ms, if iTime = 1.

(i) Determine proper time delay setting based on delays.h

(ii) Write a code for this subroutine.

(4 marks)

(b) Write three subroutines that related to LCD operation to

(i) initialize/configure in ***void LCD_setup(void).***

(ii) write command in ***void lcd_cmd(unsigned char value).***

(iii) write data in ***void lcd_data(unsigned char value).***

(16 marks)

FINAL EXAMINATION

SEMESTER/SESSION : SEMESTER II / 2010/2011
COURSE : MICROCONTROLLER APPLICATION

PROGRAMME : 4 BEE
COURSE CODE : BER4223

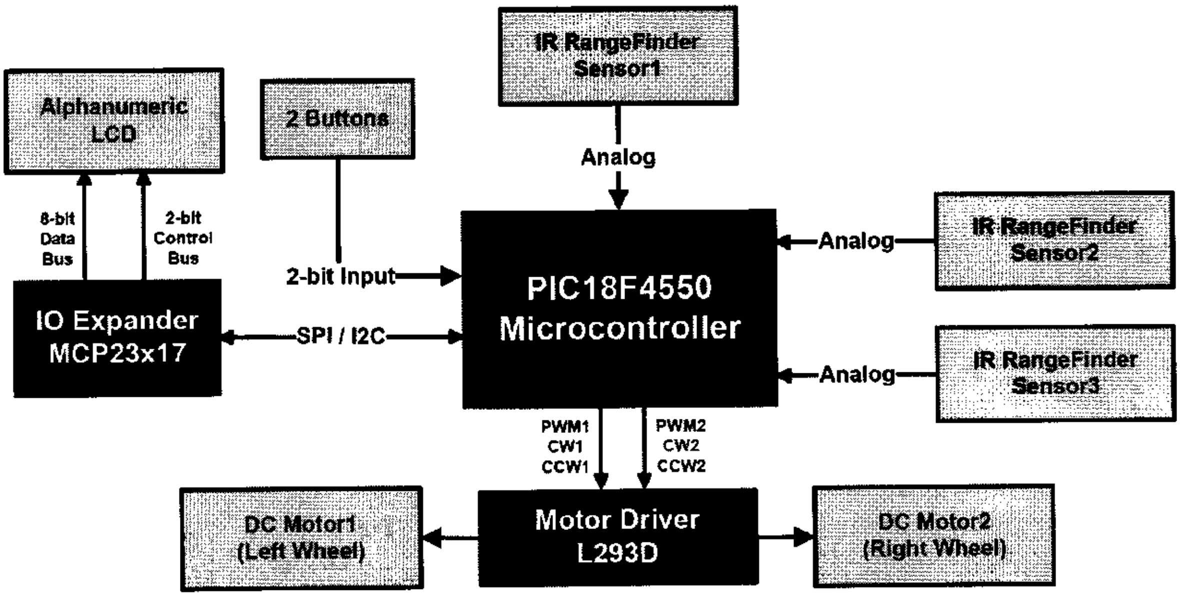
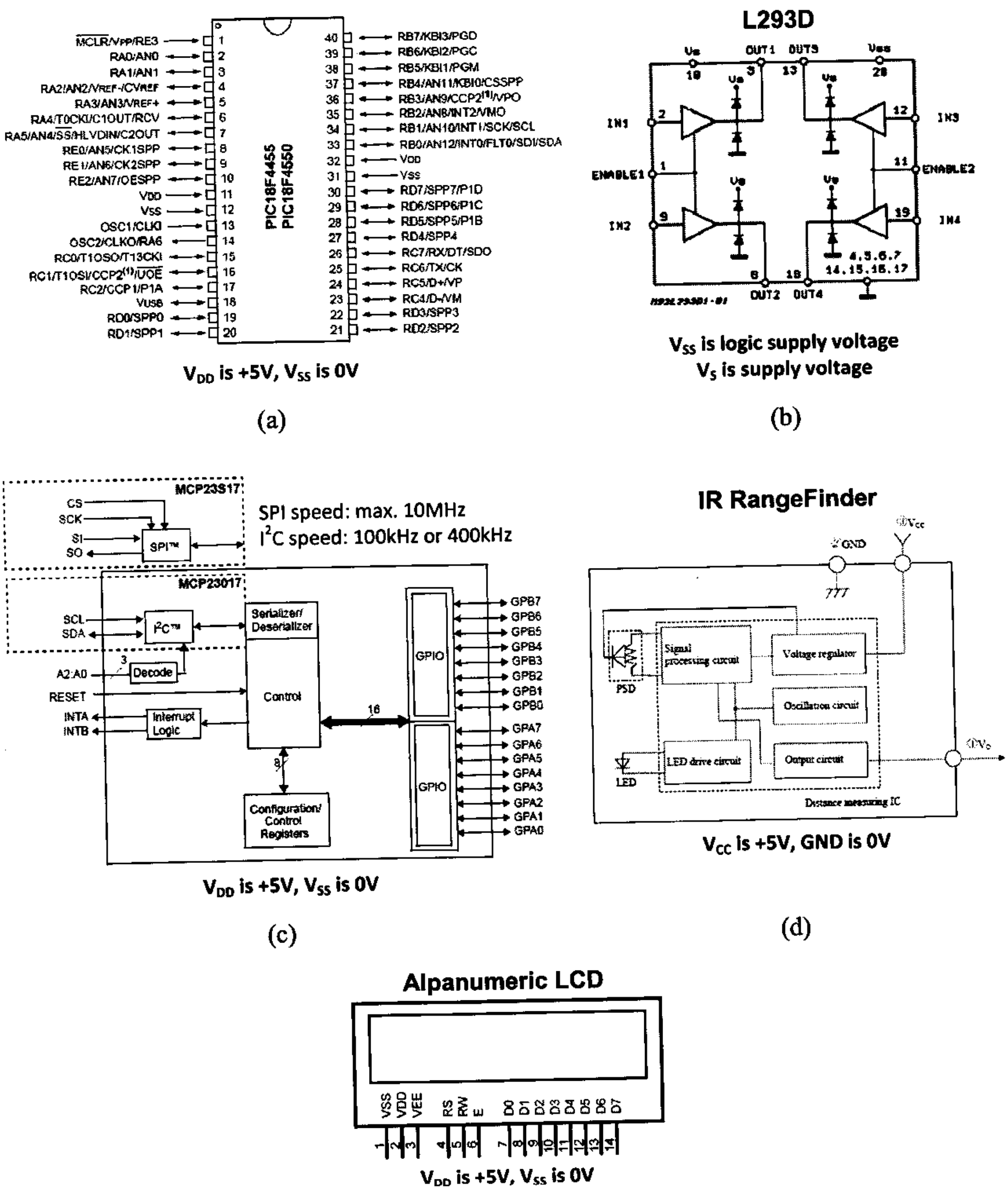


Figure 1

FINAL EXAMINATION

SEMESTER/SESSION : SEMESTER II / 2010/2011
COURSE : MICROCONTROLLER APPLICATION

PROGRAMME : 4 BEE
COURSE CODE : BER4223



V_{DD} is +5V, V_{SS} is 0V

(a)

V_{SS} is logic supply voltage
V_S is supply voltage

(b)

SPI speed: max. 10MHz
I^2C speed: 100kHz or 400kHz

V_{DD} is +5V, V_{SS} is 0V

(c)

V_{CC} is +5V, GND is 0V

(d)

V_{DD} is +5V, V_{SS} is 0V

(e)

Figure Q1

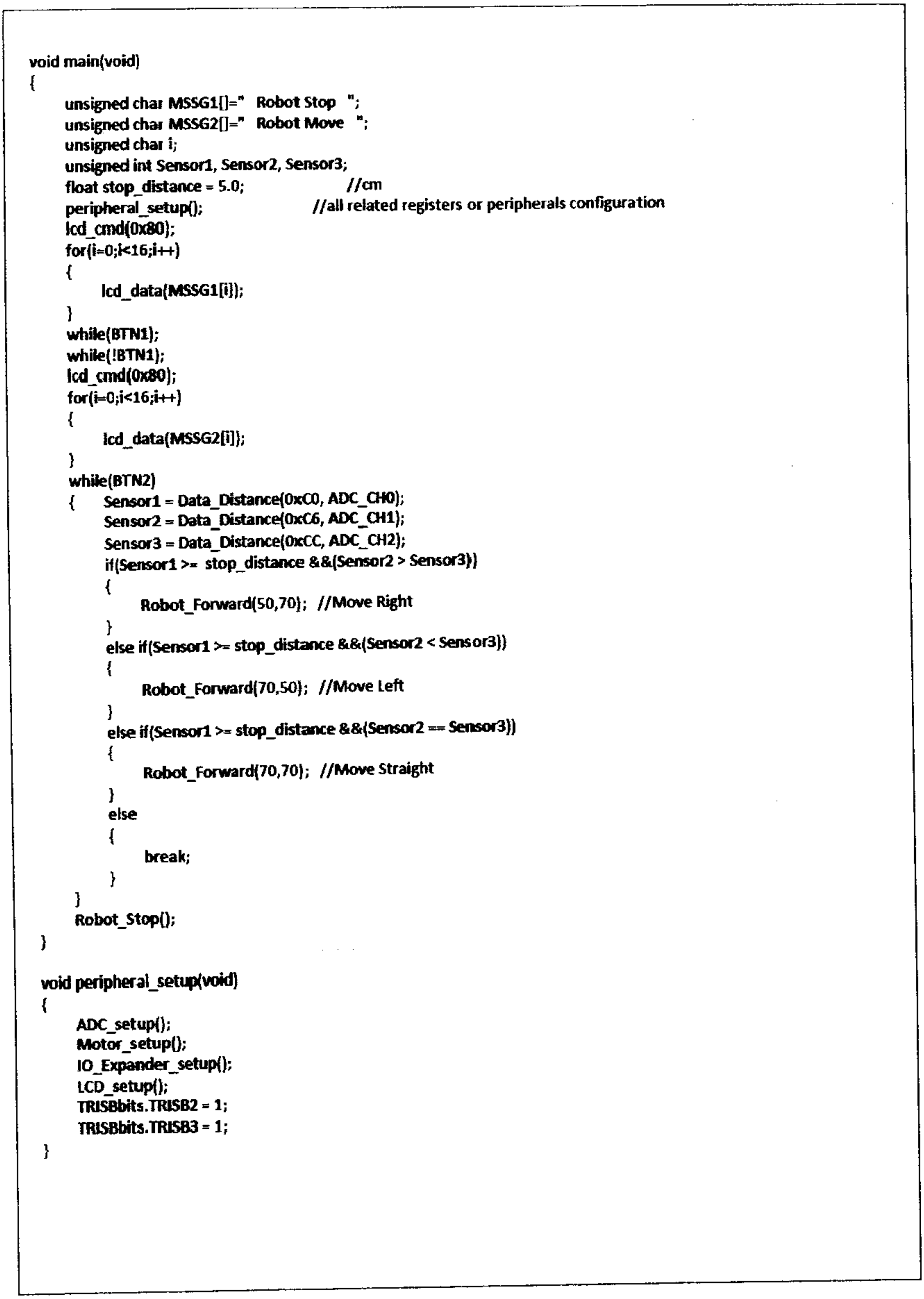
Appendix 1

```
#include <p18f4550.h>
#include <delays.h>
#include <spi.h>
#include <adc.h>
#include <pwm.h>
#include <timers.h>

//REGISTERS DEFINITION
#define  MCP23x17          0x40   //opcode and write mode
#define  IODIRA            0x00   //Control Bus
#define  IODIRB            0x01   //Data Bus
#define  IOCON             0x0A   //IOCON.BANK = 0
#define  GPIOA             0x12
#define  GPIOB             0x13
#define  LCD_Function_Set  0x38   //Function_Set
#define  LCD_Display       0x0B   //Display ON/OFF etc
#define  LCD_clear         0x01   //clear Display

#define  BTN1              PORTBbits.RB2
#define  BTN2              PORTBbits.RB3

//FUNCTION PROTOTYPES
void peripheral_setup(void);
void ADC_setup(void);
unsigned int Data_Distance(unsigned char position, unsigned char channel);
void Motor_setup(void);
void Robot_Forward(unsigned int DutyCycle_Motor_L,unsigned int DutyCycle_Motor_R);
void Robot_Stop(void);
void IO_Expander_setup(void);
void IO_Expander_write(unsigned char opcode, unsigned char reg, unsigned char data);
void msDelay(unsigned int iTime);
void LCD_setup(void);
void lcd_cmd(unsigned char value);
void lcd_data(unsigned char value);
```



```
void main(void)
{
    unsigned char MSSG1[]="  Robot Stop  ";
    unsigned char MSSG2[]="  Robot Move  ";
    unsigned char i;
    unsigned int Sensor1, Sensor2, Sensor3;
    float stop_distance = 5.0;                //cm
    peripheral_setup();                //all related registers or peripherals configuration
    lcd_cmd(0x80);
    for(i=0;i<16;i++)
    {
        lcd_data(MSSG1[i]);
    }
    while(BTN1);
    while(!BTN1);
    lcd_cmd(0x80);
    for(i=0;i<16;i++)
    {
        lcd_data(MSSG2[i]);
    }
    while(BTN2)
    {   Sensor1 = Data_Distance(0xC0, ADC_CH0);
        Sensor2 = Data_Distance(0xC6, ADC_CH1);
        Sensor3 = Data_Distance(0xCC, ADC_CH2);
        if(Sensor1 >=  stop_distance &&(Sensor2 > Sensor3))
        {
            Robot_Forward(50,70);  //Move Right
        }
        else if(Sensor1 >= stop_distance &&(Sensor2 < Sensor3))
        {
            Robot_Forward(70,50);  //Move Left
        }
        else if(Sensor1 >= stop_distance &&(Sensor2 == Sensor3))
        {
            Robot_Forward(70,70);  //Move Straight
        }
        else
        {
            break;
        }
    }
    Robot_Stop();
}

void peripheral_setup(void)
{
    ADC_setup();
    Motor_setup();
    IO_Expander_setup();
    LCD_setup();
    TRISBbits.TRISB2 = 1;
    TRISBbits.TRISB3 = 1;
}
```

```
void ADC_setup(void)
{
    //Q2(b)
}

unsigned int Data_Distance(unsigned char position, unsigned char channel)
{
    //Q2(c)
}

void Motor_setup(void)
{
    //Q3(a)
}

void Robot_Forward(unsigned int DutyCycle_Motor_L,unsigned int DutyCycle_Motor_R)
{
    //Q3(b)(i)
}

void Robot_Stop(void)
{
    //Q3(b)(ii)
}

void IO_Expander_setup(void)
{
    //Q4(b)
}

void IO_Expander_write(unsigned char opcode, unsigned char reg, unsigned char data)
{
    //Q4(c)
}

void msDelay(unsigned int iTime)
{
    //Q5(a)
}

void LCD_setup(void)
{
    //Q5(b)(i)
}

void lcd_cmd(unsigned char value)
{
    //Q5(b)(ii)
}

void lcd_data(unsigned char value)
{
    //Q5(b)(iii)
}
```