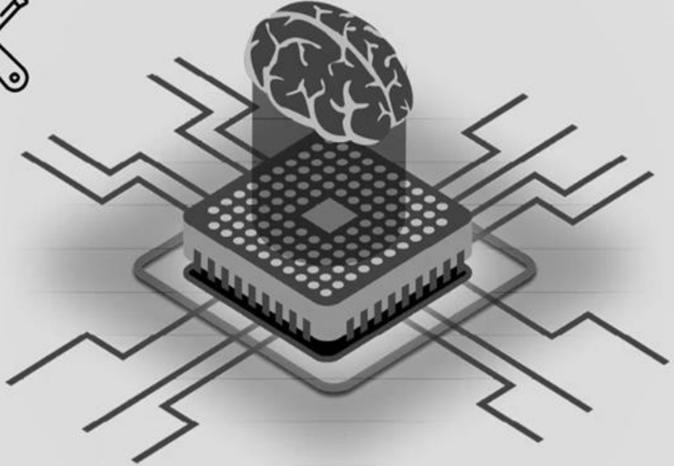
EE6352 - Embedded System Design



Avanthi Jayasundara
B.Sc.Eng.(Hons)(RUH, SL)
Lecturer (probationary)
Computer Engineering | Dept. of Electrical and Information Eng. | Faculty of Engineering

Email address:avanthi@eie.ruh.ac.lk

Learning Outcome

LO4 : Interfacing I/O : On successful completion of this session, you will be able to,

- o Apply knowledge to connect LCD to Microcontroller
- o Decide what communication mechanisms should use according to the different applications

Labs :

- Lab 05 : USART / ADC
- **Lab 06 : Interfacing LCD**

Instructions to Complete the Workshop and Labs

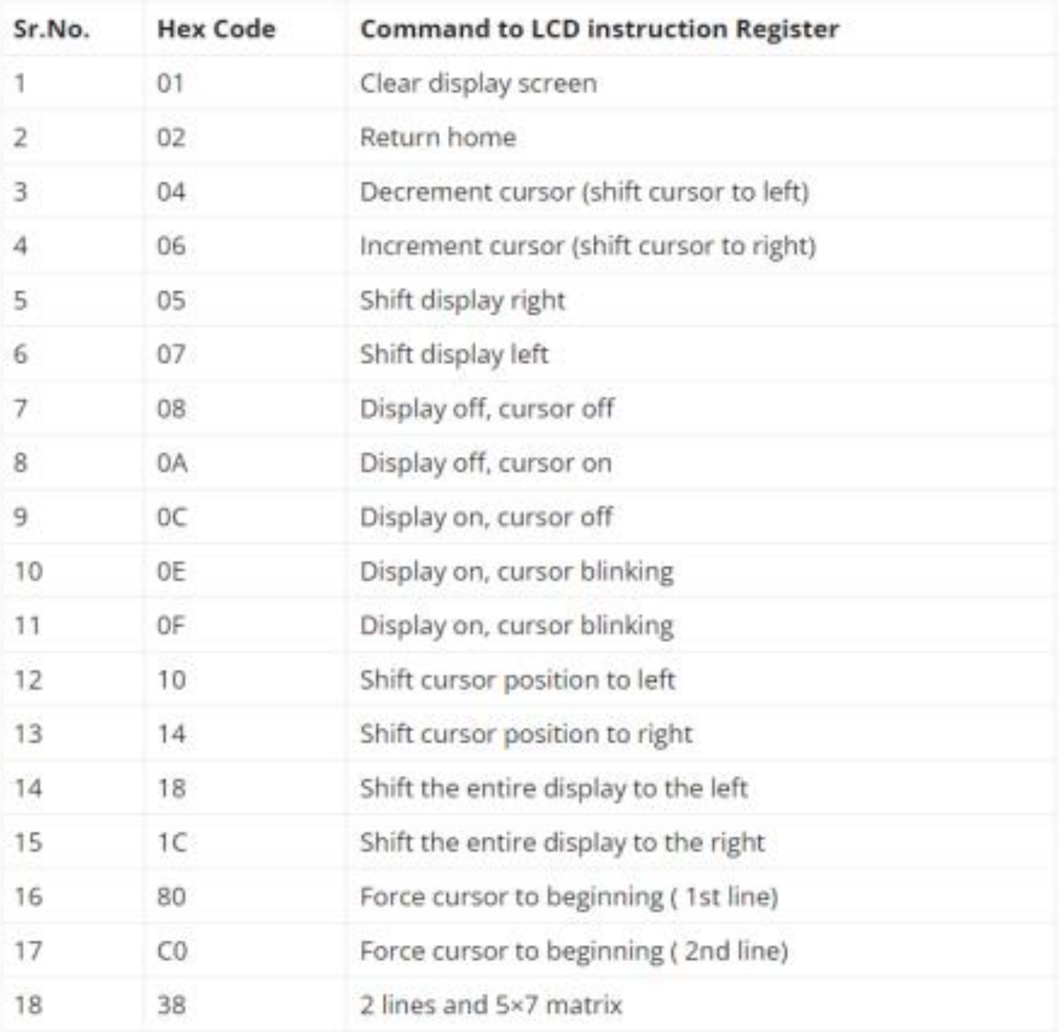
Workshops are group activities with lab group members for each group.

Every member of the group should have a copy of the activity and should be able to demonstrate the results.

Evaluation deadlines will be given after each workshop session. Every student/group has to demonstrate their work and submit the required documents/simulation files to the LMS depending on the instructions give during the workshop.

It is your responsibility to get your work demonstrations within one week from the date of workshop delivery

LCD Display : Commands



Sr.No.	Hex Code	Command to LCD instruction Register
1	01	Clear display screen
2	02	Return home
3	04	Decrement cursor (shift cursor to left)
4	06	Increment cursor (shift cursor to right)
5	05	Shift display right
6	07	Shift display left
7	08	Display off, cursor off
8	0A	Display off, cursor on
9	0C	Display on, cursor off
10	0E	Display on, cursor blinking
11	0F	Display on, cursor blinking
12	10	Shift cursor position to left
13	14	Shift cursor position to right
14	18	Shift the entire display to the left
15	1C	Shift the entire display to the right
16	80	Force cursor to beginning (1st line)
17	C0	Force cursor to beginning (2nd line)
18	38	2 lines and 5×7 matrix

Task 1 : Select suitable hardware peripheral interfaces to microcontrollers

- Use an LCD display with ATMEGA 328P
- At the end of this activity, you will be able to implement different applications on ATMEGA 328P microcontroller displaying data using an LCD display
- You need to have the data sheet for ATMEGA 328P microcontroller in order to complete this exercise

- You are required to write a code that Displays the string ‘Embedded Systems’
- Shift the display to the left till the display gets cleared
- Shift the display to the right till the display gets cleared
- Clear the display and display ‘DEIE’ at the second line Wait for a while and repeat the operation

LCD Library

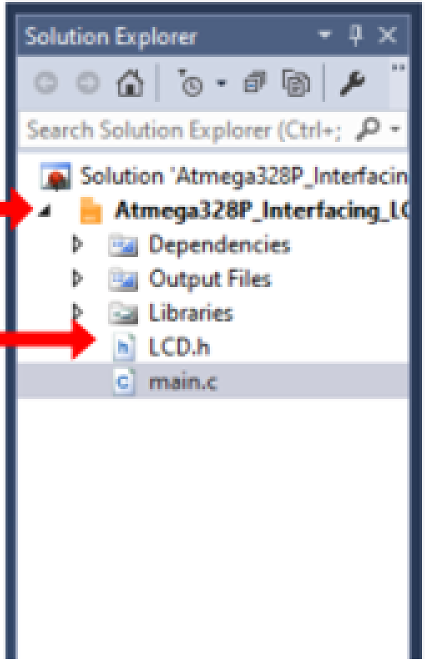
Each LCD display has its own data sheet which explains how to use You can write your own library to use the LCD display or can use freely available library form the web.

LCD Library : Create a New Header File

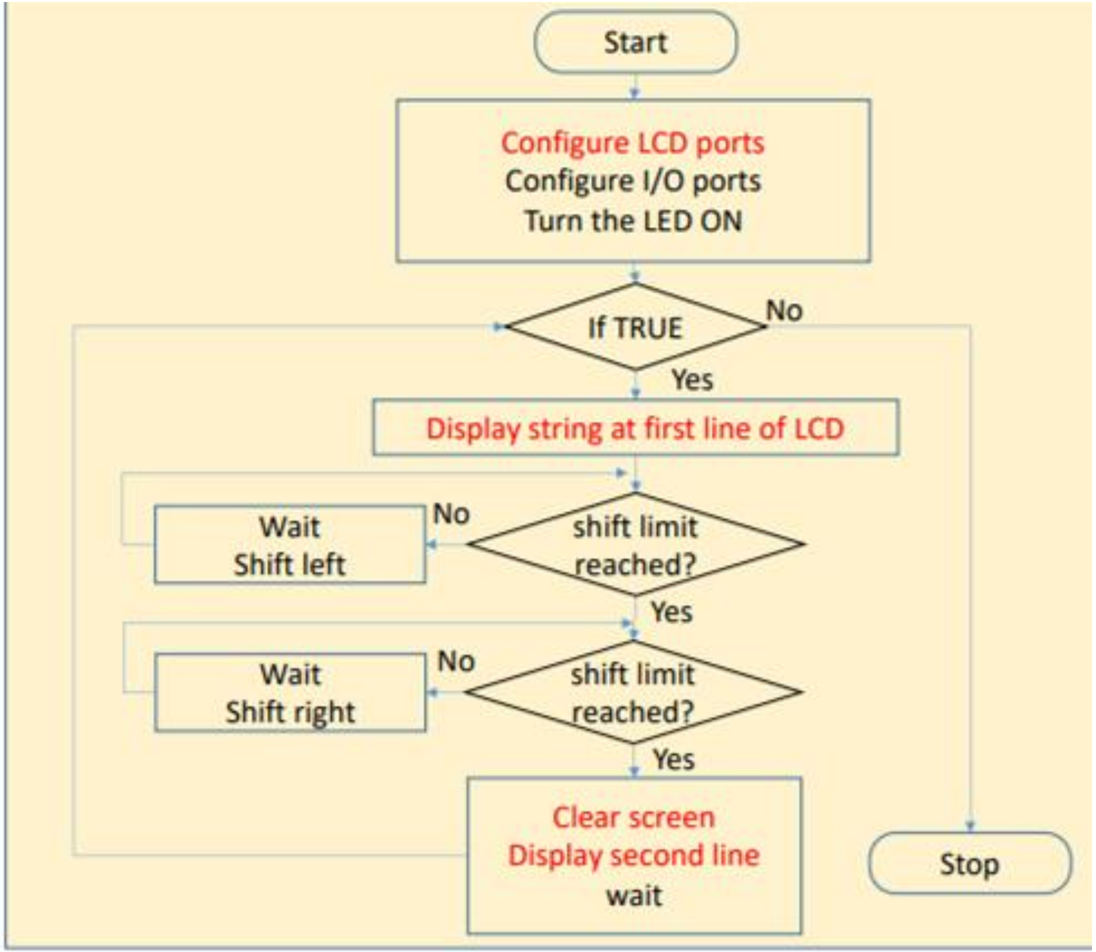
Right-click on project name > select ‘Add’> ‘New Item’

Select ‘Include File’ and name the file as ‘LCD.h’

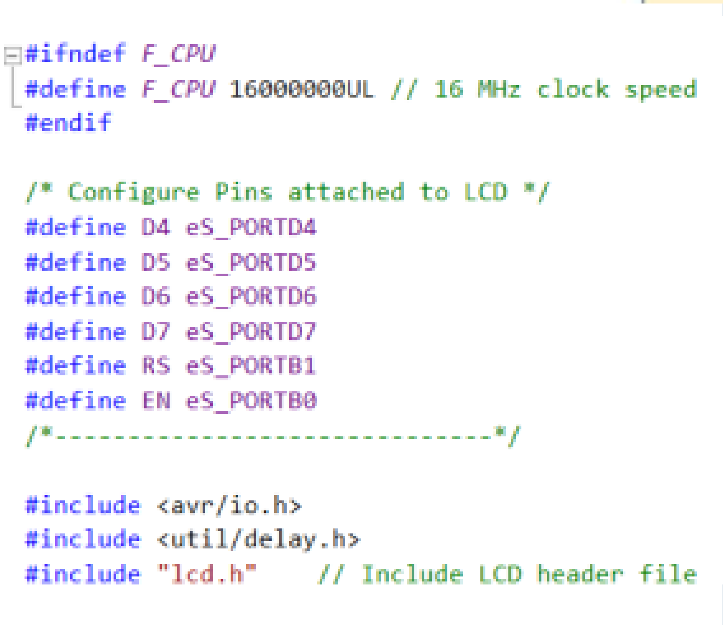
The new file will appear under your project name at the ‘Solution Explorer’



Task : Flow Chart



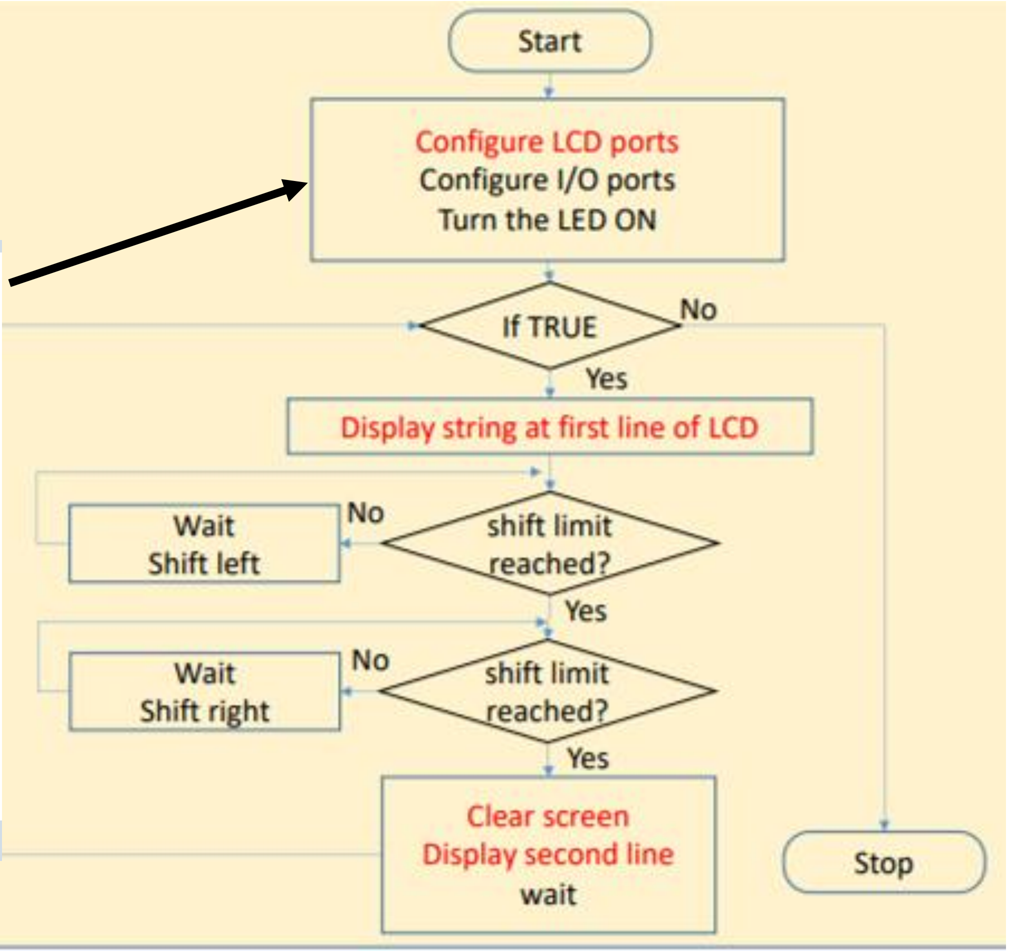
Task : Flow Chart



```
#ifndef F_CPU
#define F_CPU 16000000UL // 16 MHz clock speed
#endif

/* Configure Pins attached to LCD */
#define D4 eS_PORTD4
#define D5 eS_PORTD5
#define D6 eS_PORTD6
#define D7 eS_PORTD7
#define RS eS_PORTB1
#define EN eS_PORTB0
/*-----------------------------*/

#include <avr/io.h>
#include <util/delay.h>
#include "lcd.h"    // Include LCD header file
```



Start

Configure LCD ports
Configure I/O ports
Turn the LED ON

If TRUE

No

Yes

Display string at first line of LCD

shift limit reached?

No

Wait
Shift left

Yes

shift limit reached?

No

Wait
Shift right

Yes

Clear screen
Display second line
wait

Stop

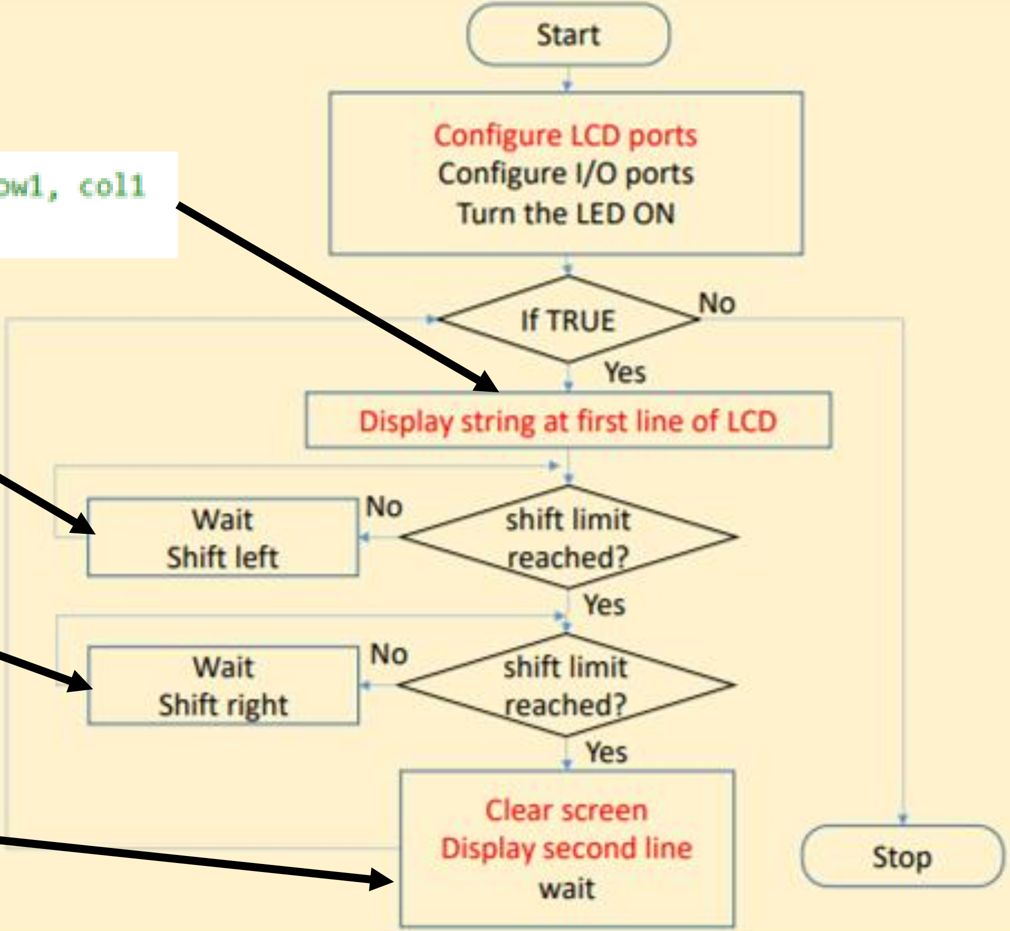
Task : Flow Chart

```
Lcd4_Set_Cursor(1,1); // Set cursor to row1, col1
Lcd4_Write_String("Embedded Systems");
```

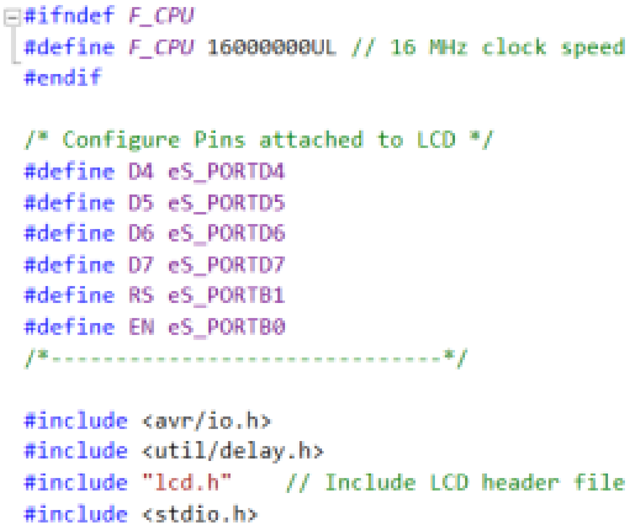
```
for(i=0;i<15;i++){
    _delay_ms(250);
    Lcd4_Shift_Left(); // Shift display left
}
```

```
for(i=0;i<14;i++){
    _delay_ms(250);
    Lcd4_Shift_Right(); // Shift display right
}
```

```
Lcd4_Clear(); // Clear the display
Lcd4_Set_Cursor(2,1); // Set cursor to row2, col1
Lcd4_Write_Char('D');
Lcd4_Write_Char('E');
Lcd4_Write_Char('I');
Lcd4_Write_Char('E');
_delay_ms(2000);
```



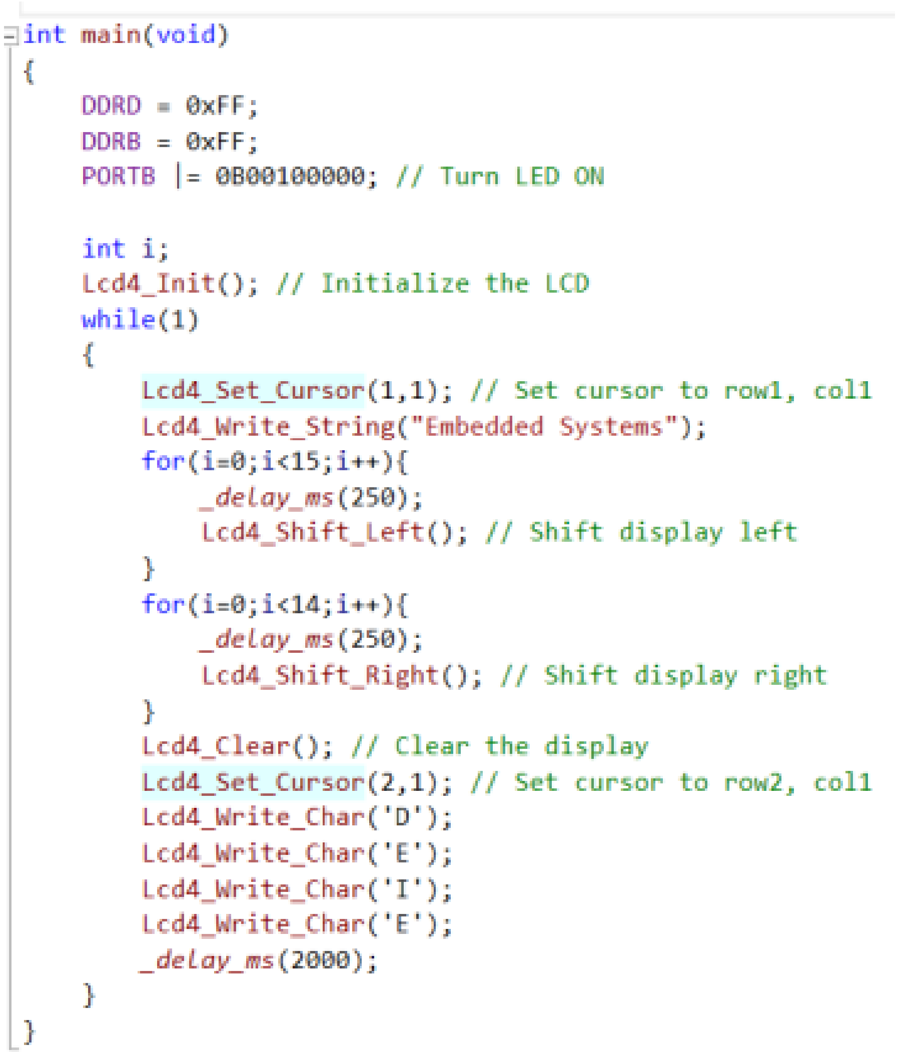
Task : Code



```
#ifndef F_CPU
#define F_CPU 16000000UL // 16 MHz clock speed
#endif

/* Configure Pins attached to LCD */
#define D4 eS_PORTD4
#define D5 eS_PORTD5
#define D6 eS_PORTD6
#define D7 eS_PORTD7
#define RS eS_PORTB1
#define EN eS_PORTB0
/*-----------------------------*/

#include <avr/io.h>
#include <util/delay.h>
#include "lcd.h"    // Include LCD header file
#include <stdio.h>
```



```
int main(void)
{
    DDRD = 0xFF;
    DDRB = 0xFF;
    PORTB |= 0B00100000; // Turn LED ON

    int i;
    Lcd4_Init(); // Initialize the LCD
    while(1)
    {
        Lcd4_Set_Cursor(1,1); // Set cursor to row1, col1
        Lcd4_Write_String("Embedded Systems");
        for(i=0;i<15;i++){
            _delay_ms(250);
            Lcd4_Shift_Left(); // Shift display left
        }
        for(i=0;i<14;i++){
            _delay_ms(250);
            Lcd4_Shift_Right(); // Shift display right
        }
        Lcd4_Clear(); // Clear the display
        Lcd4_Set_Cursor(2,1); // Set cursor to row2, col1
        Lcd4_Write_Char('D');
        Lcd4_Write_Char('E');
        Lcd4_Write_Char('I');
        Lcd4_Write_Char('E');
        _delay_ms(2000);
    }
}
```

Task : Simulation

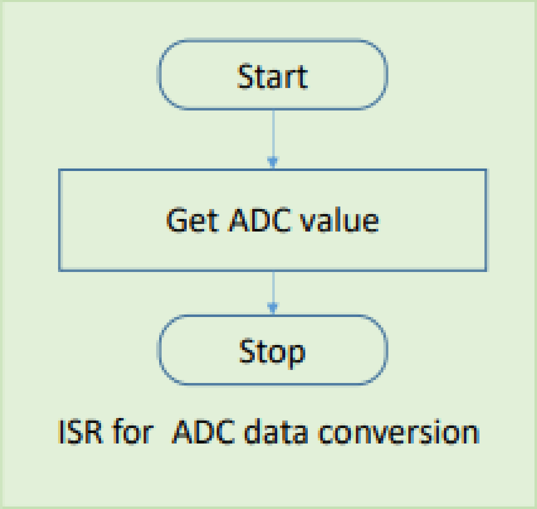
- Build the Project in Atmel Studio.
- Draw the simulation circuit in Proteus.
- Select the .hex file from just built Atmel Studio project to run the Arduino Uno development board. Run the simulation and observe output

Task 2 : Select suitable hardware peripheral interfaces to microcontrollers

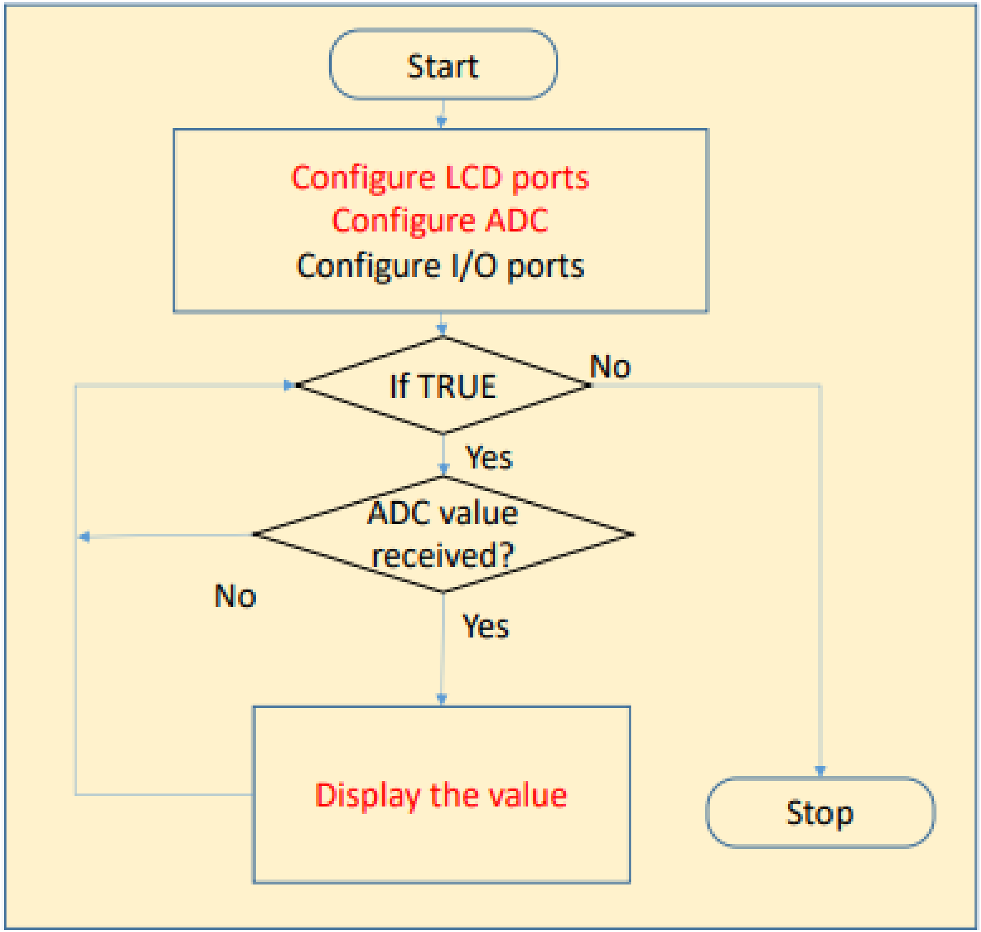
You are required to implement a Proteus simulation to display the digital value of an analog input in an LCD display using Atmega328P microcontroller.

Note that the A/D converter output is between 0 and 1024 and in the LCD display you may have to display digit by digit

Flowchart of Task 2



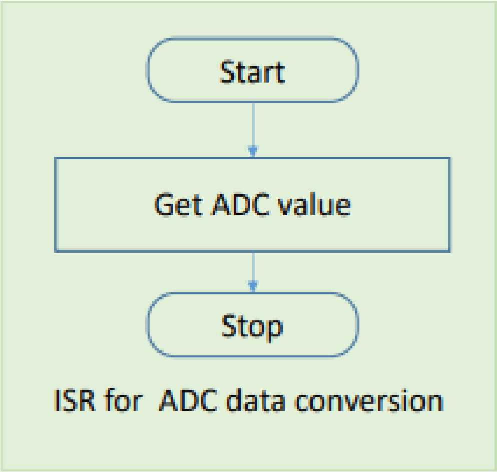
ISR for ADC data conversion



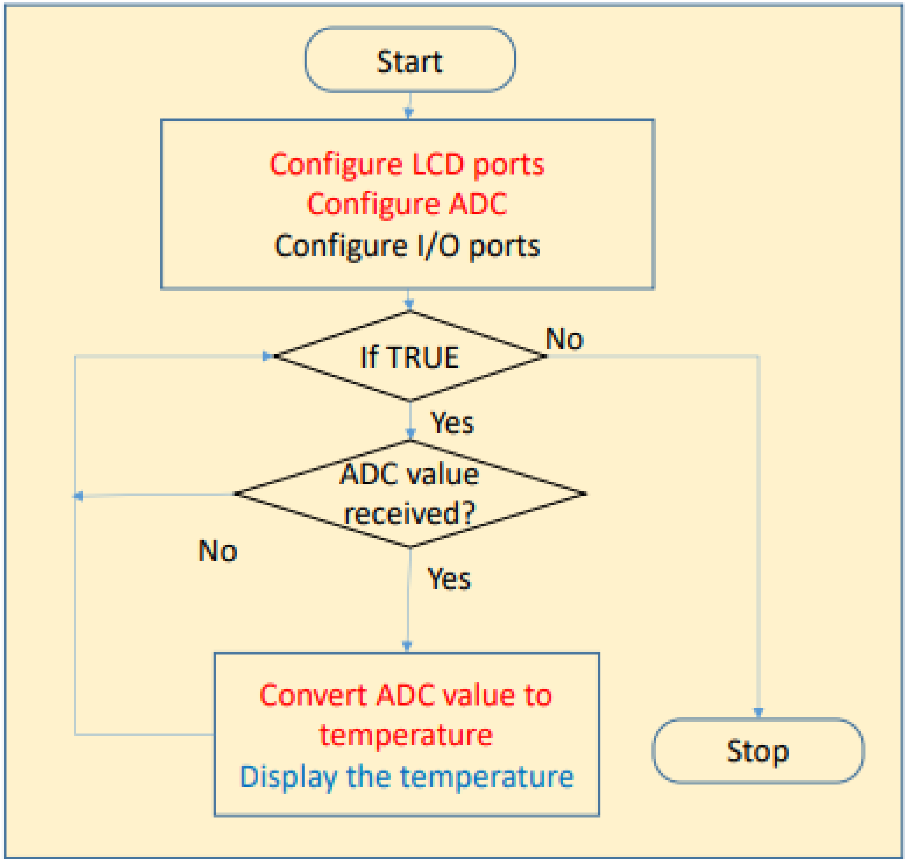
Task 4 : Select suitable hardware peripheral interfaces to microcontrollers

- You are required to implement a Proteus simulation to display the temperature of an LM35 temperature sensor in Celsius in an LCD display using Atmega328P microcontroller.
- Note that the A/D converter output is between 0 and 1024.
- LM35 sensor has a minimum temperature output of -55C and maximum temperature output of 150C ADC output is 0 at 0C and output resolution is 10mV per Celsius.

Flowchart of Task 3



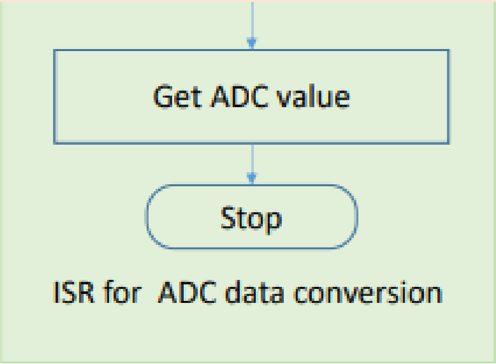
ISR for ADC data conversion



Flowchart of Task 3

Any multiplication of this **ADC value** may overflow 16 bits. Therefore, programmer should be careful in mathematical operations

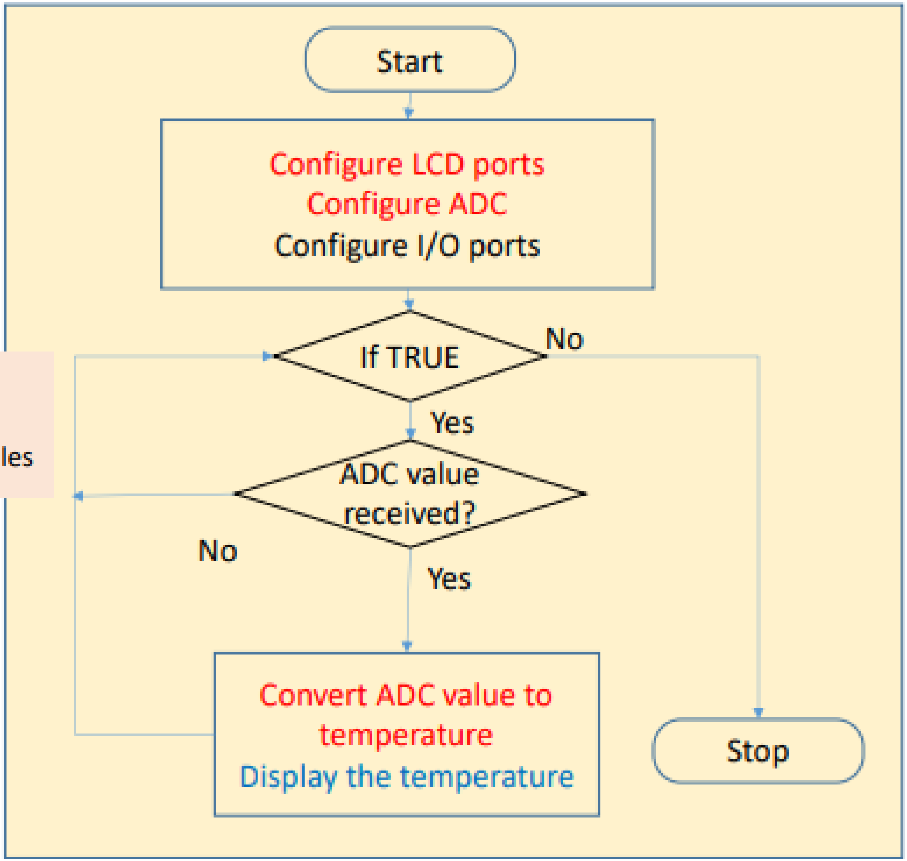
Eg. If ADC value = 512: 512 x 500 / 1023 = 256,000/1023
256,000 overflows a 16-bit register.
Therefore, this operation should be carried out in variables 32-bit variables



Get ADC value

Stop

ISR for ADC data conversion



Start

Configure LCD ports
Configure ADC
Configure I/O ports

If TRUE

No

Yes

ADC value received?

No

Yes

Convert ADC value to temperature
Display the temperature

Stop