

# LCD Interfacing on Firebird V Robot

e-Yantra Team

Embedded Real-Time Systems Lab  
Indian Institute of Technology-Bombay

IIT Bombay  
June 30, 2015



# Agenda for Discussion

- 1 Introduction
  - LCD-Definition
- 2 Understanding LCD
  - Pin-Configuration
  - Control Pins
  - Data Pins
- 3 LCD Programming
  - LCD Interfacing
  - Some Important commands
  - LCD Initialization
  - Programming



# Liquid Crystal Display



# Liquid Crystal Display

- ① A liquid crystal display (LCD) is a thin, flat panel used for electronically displaying information such as text, images, and moving pictures



# Liquid Crystal Display

- ① A liquid crystal display (LCD) is a thin, flat panel used for electronically displaying information such as text, images, and moving pictures
- ② LCDs are economical and easy to use device. These are most commonly used display devices in an embedded system. Commonly available display are set up as 16 to 20 characters by 1 to 4 lines



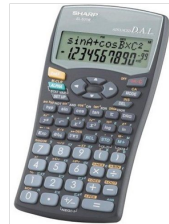
# Liquid Crystal Display

- ① A liquid crystal display (LCD) is a thin, flat panel used for electronically displaying information such as text, images, and moving pictures
- ② LCDs are economical and easy to use device. These are most commonly used display devices in an embedded system. Commonly available display are set up as 16 to 20 characters by 1 to 4 lines



# Liquid Crystal Display

- 1 A liquid crystal display (LCD) is a thin, flat panel used for electronically displaying information such as text, images, and moving pictures
- 2 LCDs are economical and easy to use device. These are most commonly used display devices in an embedded system. Commonly available display are set up as 16 to 20 characters by 1 to 4 lines



# Dot Matrix Liquid Crystal Display





# Dot Matrix Liquid Crystal Display

- ① LCD used here has HD44780 dot matrix lcd controller. It is also called 16x2 Alpha Numeric LCD



# Dot Matrix Liquid Crystal Display

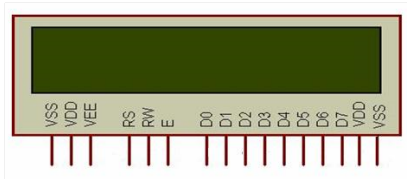
- ① LCD used here has HD44780 dot matrix lcd controller. It is also called 16x2 Alpha Numeric LCD
- ② It can be configured to drive a dot-matrix liquid crystal display under the control of a 4 or 8-bit microprocessor



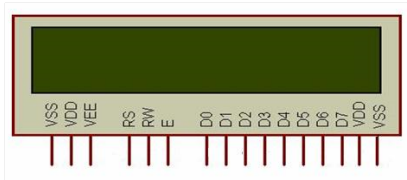
# Pin-Configuration



# Pin-Configuration



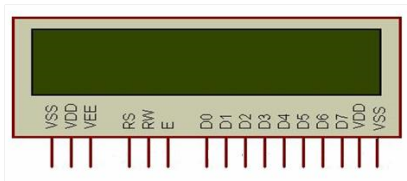
# Pin-Configuration



Pin	Description
-----	-------------



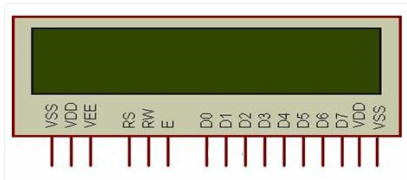
# Pin-Configuration



Pin	Description
Vss	Ground



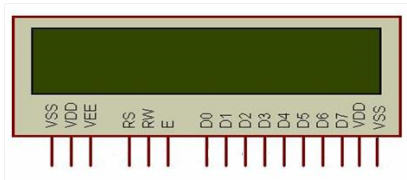
# Pin-Configuration



Pin	Description
Vss	Ground
Vdd	Supply Voltage



# Pin-Configuration

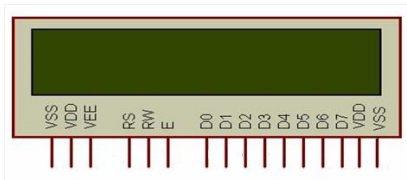


Pin	Description
Vss	Ground
Vdd	Supply Voltage
Vee	Contrast Voltage





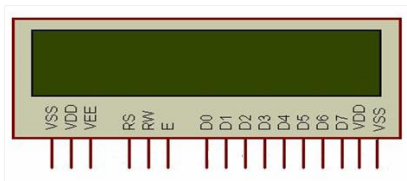
# Pin-Configuration



Pin	Description
Vss	Ground
Vdd	Supply Voltage
Vee	Contrast Voltage
RS	Register Select



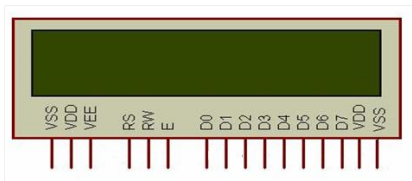
# Pin-Configuration



Pin	Description
Vss	Ground
Vdd	Supply Voltage
Vee	Contrast Voltage
RS	Register Select
RW	Read/Write



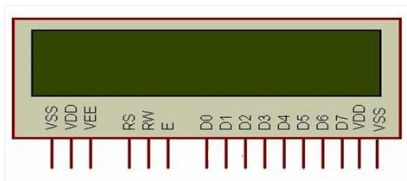
# Pin-Configuration



Pin	Description
Vss	Ground
Vdd	Supply Voltage
Vee	Contrast Voltage
RS	Register Select
RW	Read/Write
E	Enable



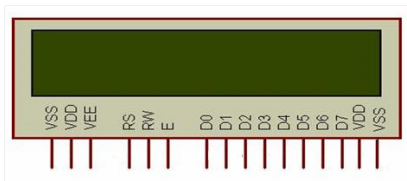
# Pin-Configuration



Pin	Description
Vss	Ground
Vdd	Supply Voltage
Vee	Contrast Voltage
RS	Register Select
RW	Read/Write
E	Enable
D0-D7	Bidirectional Data Bus



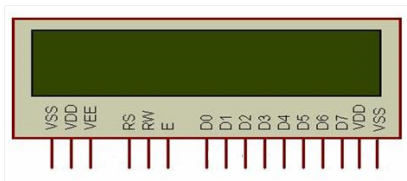
# Pin-Configuration



Pin	Description
Vss	Ground
Vdd	Supply Voltage
Vee	Contrast Voltage
RS	Register Select
RW	Read/Write
E	Enable
D0-D7	Bidirectional Data Bus
Vdd,Vss	Back Light Supply



# Pin-Configuration



Pin	Description
Vss	Ground
Vdd	Supply Voltage
Vee	Contrast Voltage
RS	Register Select
RW	Read/Write
E	Enable
D0-D7	Bidirectional Data Bus
Vdd,Vss	Back Light Supply



# Control Pins



# Control Pins

## ① Register Select





# Control Pins

- ① Register Select
  - If  $RS=0$ ; Command Register



# Control Pins

- ① Register Select
  - If  $RS=0$ ; Command Register
  - If  $RS=1$ ; Data Register



# Control Pins

- ① Register Select
  - If  $RS=0$ ; Command Register
  - If  $RS=1$ ; Data Register
- ② Read/Write Select



# Control Pins

- ① Register Select
  - If  $RS=0$ ; Command Register
  - If  $RS=1$ ; Data Register
- ② Read/Write Select
  - If  $RW=0$ ; Write Mode



# Control Pins

- ① Register Select
  - If  $RS=0$ ; Command Register
  - If  $RS=1$ ; Data Register
- ② Read/Write Select
  - If  $RW=0$ ; Write Mode
  - If  $RW=1$ ; Read Mode



# Control Pins

- ① Register Select
  - If  $RS=0$ ; Command Register
  - If  $RS=1$ ; Data Register
- ② Read/Write Select
  - If  $RW=0$ ; Write Mode
  - If  $RW=1$ ; Read Mode
- ③ Enable



# Control Pins

## 1 Register Select

- If  $RS=0$ ; Command Register
- If  $RS=1$ ; Data Register

## 2 Read/Write Select

- If  $RW=0$ ; Write Mode
- If  $RW=1$ ; Read Mode

## 3 Enable

- Used to latch the data present on the data pins



# Control Pins

## 1 Register Select

- If  $RS=0$ ; Command Register
- If  $RS=1$ ; Data Register

## 2 Read/Write Select

- If  $RW=0$ ; Write Mode
- If  $RW=1$ ; Read Mode

## 3 Enable

- Used to latch the data present on the data pins
- A high-to-low edge is needed to latch the data





# Data Pins



# Data Pins

✓ Data Lines



# Data Pins

- ✓ Data Lines
  - There are 8 data pins from D0 to D7



# Data Pins

- ✓ Data Lines
  - There are 8 data pins from D0 to D7
  - Bidirectional Data / Command Pins



# Data Pins

## ✓ Data Lines

- There are 8 data pins from D0 to D7
- Bidirectional Data / Command Pins
- Alpha Numeric Character are sent in ASCII format



# Data Pins

## ✓ Data Lines

- There are 8 data pins from D0 to D7
- Bidirectional Data / Command Pins
- Alpha Numeric Character are sent in ASCII format
- We can use LCD either 8 bit mode or 4 bit mode



# Data Pins

## ✓ Data Lines

- There are 8 data pins from D0 to D7
- Bidirectional Data / Command Pins
- Alpha Numeric Character are sent in ASCII format
- We can use LCD either 8 bit mode or 4 bit mode
- We use 4 bit mode: only D4 to D7 data pins are used

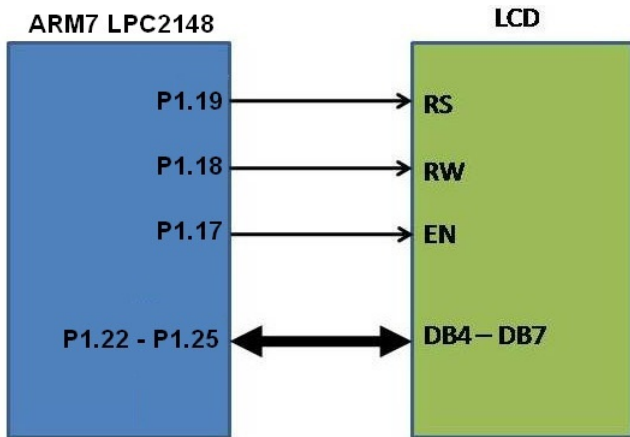


# LCD Interfacing





# LCD Interfacing



# Some Important Commands



# Some Important Commands

Description	Hex
-------------	-----



# Some Important Commands

Description	Hex
Function set (8-bit interface, 2 lines, 5*7 Pixels)	38



# Some Important Commands

Description	Hex
Function set (8-bit interface, 2 lines, 5*7 Pixels)	38
Function set (4-bit interface, 2 lines, 5*7 Pixels)	28



# Some Important Commands

Description	Hex
Function set (8-bit interface, 2 lines, 5*7 Pixels)	38
Function set (4-bit interface, 2 lines, 5*7 Pixels)	28
Clear display screen	01



# Some Important Commands

Description	Hex
Function set (8-bit interface, 2 lines, 5*7 Pixels)	38
Function set (4-bit interface, 2 lines, 5*7 Pixels)	28
Clear display screen	01
Return Home (First line first block)	02



# Some Important Commands

Description	Hex
Function set (8-bit interface, 2 lines, 5*7 Pixels)	38
Function set (4-bit interface, 2 lines, 5*7 Pixels)	28
Clear display screen	01
Return Home (First line first block)	02
Display ON cursor Blinking	0F





# Some Important Commands

Description	Hex
Function set (8-bit interface, 2 lines, 5*7 Pixels)	38
Function set (4-bit interface, 2 lines, 5*7 Pixels)	28
Clear display screen	01
Return Home (First line first block)	02
Display ON cursor Blinking	0F
Address for Line 1	80



# Some Important Commands

Description	Hex
Function set (8-bit interface, 2 lines, 5*7 Pixels)	38
Function set (4-bit interface, 2 lines, 5*7 Pixels)	28
Clear display screen	01
Return Home (First line first block)	02
Display ON cursor Blinking	0F
Address for Line 1	80
Address for Line 2	C0



# Some Important Commands

Description	Hex
Function set (8-bit interface, 2 lines, 5*7 Pixels)	38
Function set (4-bit interface, 2 lines, 5*7 Pixels)	28
Clear display screen	01
Return Home (First line first block)	02
Display ON cursor Blinking	0F
Address for Line 1	80
Address for Line 2	C0
Display ON cursor OFF	0C



# Some Important Commands

Description	Hex
Function set (8-bit interface, 2 lines, 5*7 Pixels)	38
Function set (4-bit interface, 2 lines, 5*7 Pixels)	28
Clear display screen	01
Return Home (First line first block)	02
Display ON cursor Blinking	0F
Address for Line 1	80
Address for Line 2	C0
Display ON cursor OFF	0C



# Steps for LCD Initialization



# Steps for LCD Initialization

## ① Set Port1 as GPIO Port



# Steps for LCD Initialization

- 1 Set Port1 as GPIO Port
- 2 Initialize Port1 as Output Port



# Steps for LCD Initialization

- 1 Set Port1 as GPIO Port
- 2 Initialize Port1 as Output Port
- 3 Set Control Lines i.e.  $RS=0$  and  $RW=0$





# Steps for LCD Initialization

- 1 Set Port1 as GPIO Port
- 2 Initialize Port1 as Output Port
- 3 Set Control Lines i.e. RS=0 and RW=0
- 4 Send LCD init value i.e. 0x38 for 8-bit mode OR 0x28 for 4-bit mode



# Steps for LCD Initialization

- ➊ Set Port1 as GPIO Port
- ➋ Initialize Port1 as Output Port
- ➌ Set Control Lines i.e. RS=0 and RW=0
- ➍ Send LCD init value i.e. 0x38 for 8-bit mode OR 0x28 for 4-bit mode
- ➎ Generate High-Low Pulse on Enable Pin of LCD



# Steps for LCD Initialization

- 1 Set Port1 as GPIO Port
- 2 Initialize Port1 as Output Port
- 3 Set Control Lines i.e. RS=0 and RW=0
- 4 Send LCD init value i.e. 0x38 for 8-bit mode OR 0x28 for 4-bit mode
- 5 Generate High-Low Pulse on Enable Pin of LCD
- 6 Send LCD Clear value i.e. 0x01



# Steps for LCD Initialization

- ➊ Set Port1 as GPIO Port
- ➋ Initialize Port1 as Output Port
- ➌ Set Control Lines i.e. RS=0 and RW=0
- ➍ Send LCD init value i.e. 0x38 for 8-bit mode OR 0x28 for 4-bit mode
- ➎ Generate High-Low Pulse on Enable Pin of LCD
- ➏ Send LCD Clear value i.e. 0x01
- ➐ Send LCD Display On value i.e. 0x0F



# Steps for LCD Initialization

- 1 Set Port1 as GPIO Port
- 2 Initialize Port1 as Output Port
- 3 Set Control Lines i.e. RS=0 and RW=0
- 4 Send LCD init value i.e. 0x38 for 8-bit mode OR 0x28 for 4-bit mode
- 5 Generate High-Low Pulse on Enable Pin of LCD
- 6 Send LCD Clear value i.e. 0x01
- 7 Send LCD Display On value i.e. 0x0F
- 8 Send LCD Cursor Home i.e. 0x02



# Syntax for C-Program



# Syntax for C-Program

```
#include
```



# Syntax for C-Program

```
#include
```

```
#include <lpc214x.h>
```

```
#include "LCD.h" // User-defined header file
```





# Syntax for C-Program

#include

```
#include <lpc214x.h>
```

```
#include "LCD.h" // User-defined header file
```

Main Program



# Syntax for C-Program

## #include

```
#include <lpc214x.h>
#include "LCD.h"           // User-defined header file
```

## Main Program

```
int main (void)
{
    Init_LCD_Pin();
    LCD_Init();
    LCD_Clear();
    LCD_Cursor(1,1);
    LCD_String("E-yantra");
    LCD_Cursor(2,3);
    LCD_String("IIT-Bombay");
    while(1);
}
```



# LCD.h- The Header File



# LCD.h- The Header File

❗ This file must be copied into Project Folder



# LCD.h- The Header File

- ❗ This file must be copied into Project Folder



# LCD.h- The Header File

❗ This file must be copied into Project Folder

```
//define port where LCD is connected  
void Init_LCD_Pin();
```



# LCD.h- The Header File

❗ This file must be copied into Project Folder

```
//define port where LCD is connected  
void Init_LCD_Pin();  
  
//LCD Initialization  
void LCD_Init();
```



# LCD.h- The Header File

❗ This file must be copied into Project Folder

```
//define port where LCD is connected  
void Init_LCD_Pin();  
  
//LCD Initialization  
void LCD_Init();  
  
//To Send Command  
void LCD_Command(unsigned int data);
```





# LCD.h- The Header File

❗ This file must be copied into Project Folder

```
//define port where LCD is connected  
void Init_LCD_Pin();  
  
//LCD Initialization  
void LCD_Init();  
  
//To Send Command  
void LCD_Command(unsigned int data);  
  
//To write single character  
void LCD_Data(unsigned int data);
```



# LCD.h- The Header File

❗ This file must be copied into Project Folder

```
//define port where LCD is connected  
void Init_LCD_Pin();  
  
//LCD Initialization  
void LCD_Init();  
  
//To Send Command  
void LCD_Command(unsigned int data);  
  
//To write single character  
void LCD_Data(unsigned int data);  
  
//To print string of characters  
void LCD_String(char*);
```



# LCD.h- The Header File

❗ This file must be copied into Project Folder

```
//define port where LCD is connected
void Init_LCD_Pin();

//LCD Initialization
void LCD_Init();

//To Send Command
void LCD_Command(unsigned int data);

//To write single character
void LCD_Data(unsigned int data);

//To print string of characters
void LCD_String(char*);

//To Place cursor at desired location
void LCD_Cursor(char row,char column);
```



# LCD.h- The Header File

❗ This file must be copied into Project Folder

```
//define port where LCD is connected
void Init_LCD_Pin();

//LCD Initialization
void LCD_Init();

//To Send Command
void LCD_Command(unsigned int data);

//To write single character
void LCD_Data(unsigned int data);

//To print string of characters
void LCD_String(char*);

//To Place cursor at desired location
void LCD_Cursor(char row,char column);

//To Print Numeric Value
void LCD_Print(char row,char coloumn,unsigned int value,int digits);
```



# LCD.h- The Header File

❗ This file must be copied into Project Folder

```
//define port where LCD is connected
void Init_LCD_Pin();

//LCD Initialization
void LCD_Init();

//To Send Command
void LCD_Command(unsigned int data);

//To write single character
void LCD_Data(unsigned int data);

//To print string of characters
void LCD_String(char*);

//To Place cursor at desired location
void LCD_Cursor(char row,char column);

//To Print Numeric Value
void LCD_Print(char row,char coloumn,unsigned int value,int digits);
```



# Thank You!

Post your queries on: <http://qa.e-yantra.org/>

