LCD Interfacing on Firebird V Robot

e-Yantra Team

Embedded Real-Time Systems (ERTS) Lab
Indian Institute of Technology, Bombay

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

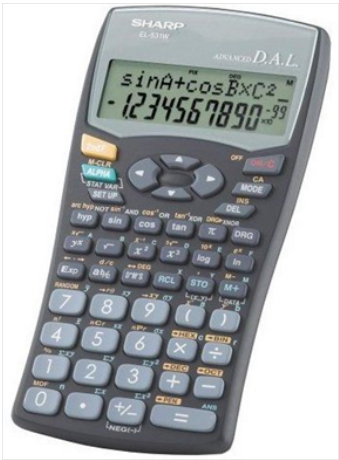
1. A liquid crystal display (LCD) is a thin, flat panel used for electronically displaying information such as text, images, and moving pictures

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

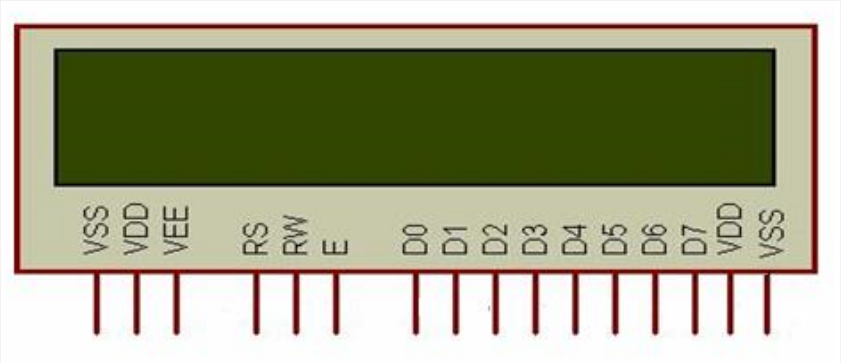
Dot Matrix Liquid Crystal Display

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

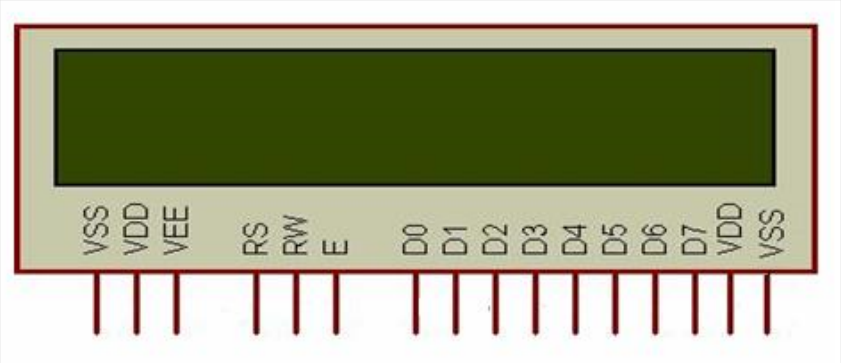
2. It can be configured to drive a dot-matrix liquid crystal display

Pin-Configuration

Pin-Configuration



Pin-Configuration



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

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

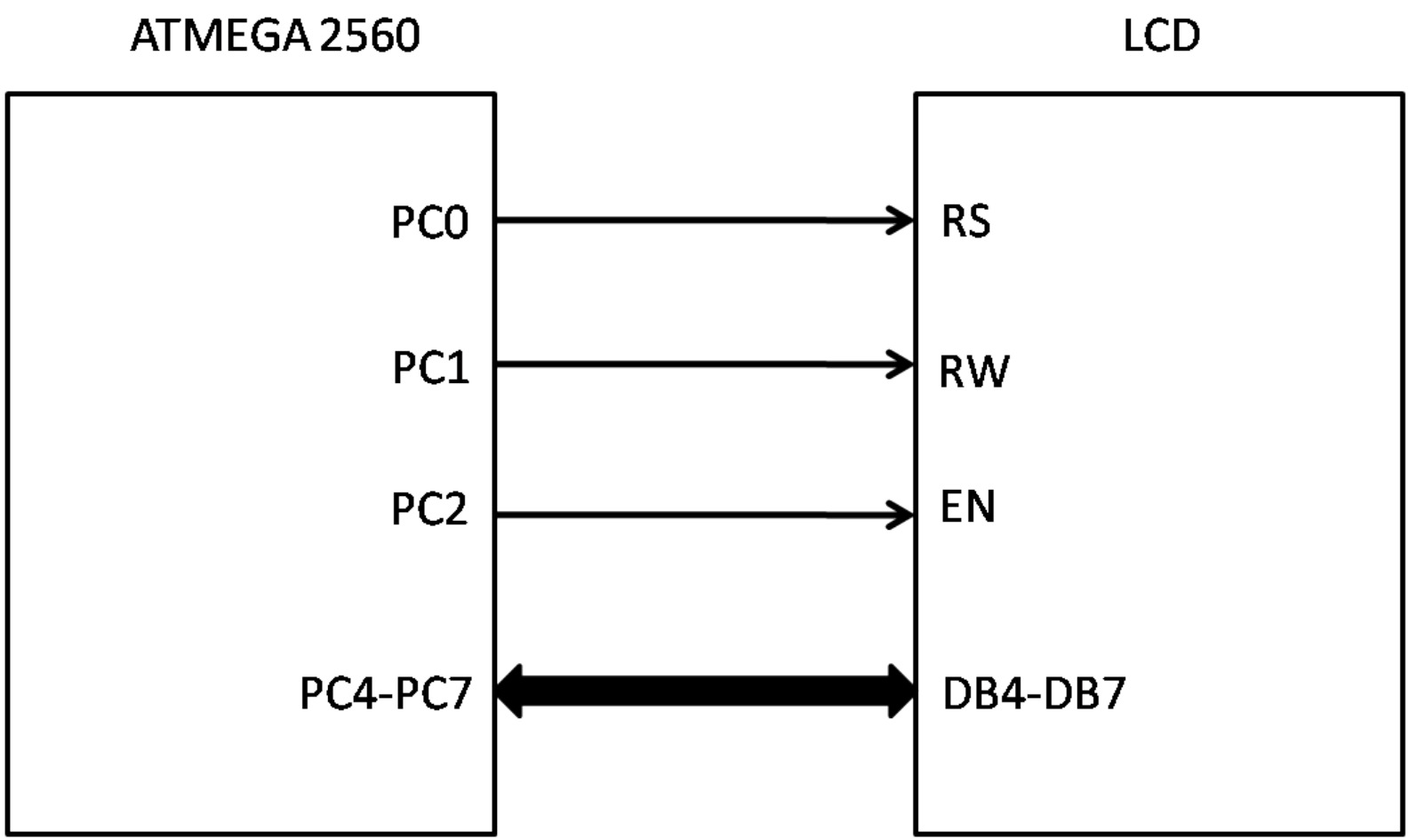
- Data Lines
  - There are 8 data pins from D0 to D7
  - Bidirectional Data / Command Pins
  - Alpha Numeric Character are sent in ASCII format
  - We can use LCD either 8 bit mode or 4 bit mode

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. Initialize PortC as Output Port

Steps for LCD Initialization

1. Initialize PortC as Output Port
2. Set Control Lines i.e. RS=0 and RW=0

Steps for LCD Initialization

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

Steps for LCD Initialization

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

Steps for LCD Initialization

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

Steps for LCD Initialization

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

Steps for LCD Initialization

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

lcd.h - The header file

lcd.h - The header file

1. This file must be copied into Project Folder

lcd.h - The header file

1. This file must be copied into Project Folder

lcd.h - The header file

1. This file must be copied into Project Folder

```
// Configure the Port where LCD is connected
void lcd_port_config();
```

lcd.h - The header file

1. This file must be copied into Project Folder

```
// Configure the Port where LCD is connected
void lcd_port_config();

// To initialize LCD
void lcd_init();
```

lcd.h - The header file

1. This file must be copied into Project Folder

```
// Configure the Port where LCD is connected
void lcd_port_config();

// To initialize LCD
void lcd_init();

// To send command
void lcd_wr_command(unsigned char cmd);
```

lcd.h - The header file

1. This file must be copied into Project Folder

```
// Configure the Port where LCD is connected
void lcd_port_config();

// To initialize LCD
void lcd_init();

// To send command
void lcd_wr_command(unsigned char cmd);

// To write single character
void lcd_wr_char(char row, char column, char alpha_num_char);
```

lcd.h - The header file

1. This file must be copied into Project Folder

```
// Configure the Port where LCD is connected
void lcd_port_config();

// To initialize LCD
void lcd_init();

// To send command
void lcd_wr_command(unsigned char cmd);

// To write single character
void lcd_wr_char(char row, char column, char alpha_num_char);


// To print string of characters
void lcd_string(char row, char column, char* str);
```

lcd.h - The header file

1. This file must be copied into Project Folder

```
// Configure the Port where LCD is connected
void lcd_port_config();

// To initialize LCD
void lcd_init();

// To send command
void lcd_wr_command(unsigned char cmd);

// To write single character
void lcd_wr_char(char row, char column, char alpha_num_char);


// To print string of characters
void lcd_string(char row, char column, char* str);


// To place cursor at a desired location
void lcd_cursor(char row, char column);
```

lcd.h - The header file

1. This file must be copied into Project Folder

```
// Configure the Port where LCD is connected
void lcd_port_config();

// To initialize LCD
void lcd_init();

// To send command
void lcd_wr_command(unsigned char cmd);

// To write single character
void lcd_wr_char(char row, char column, char alpha_num_char);


// To print string of characters
void lcd_string(char row, char column, char* str);


// To place cursor at a desired location
void lcd_cursor(char row, char column);

// To print numeric values
void lcd_numeric_value(char row,char column, int value, int digits);
```

lcd.h - The header file

1. This file must be copied into Project Folder

```
// Configure the Port where LCD is connected
void lcd_port_config();

// To initialize LCD
void lcd_init();

// To send command
void lcd_wr_command(unsigned char cmd);

// To write single character
void lcd_wr_char(char row, char column, char alpha_num_char);


// To print string of characters
void lcd_string(char row, char column, char* str);


// To place cursor at a desired location
void lcd_cursor(char row, char column);

// To print numeric values
void lcd_numeric_value(char row,char column, int value, int digits);
```

Syntax for C-Program

Syntax for C-Program

#include

Syntax for C-Program

#include

```
#include <avr/io.h>
#include <util/delay.h>
#include "lcd.h"                    // User-defined header file
```

Syntax for C-Program

#include

```
#include <avr/io.h>
#include <util/delay.h>
#include "lcd.h"                    // User-defined header file
```

Main Program

Syntax for C-Program

#include

```
#include <avr/io.h>
#include <util/delay.h>
#include "lcd.h"                    // User-defined header file
```

Main Program

```
int main (void)
{
     lcd_port_config();
     lcd_init();
     lcd_string(1, 3, "e-Yantra");
     lcd_string(2, 3, "IIT Bombay");
     _delay_ms(5000);
     lcd_clear();
     lcd_numeric_value(2, 6, 458, 4);
     while(1);
  }
```

Thank You!

Post your queries on: support@e-yantra.org