LCM Module
for Puzzle Series

User Manual

Rev. 1.00 – September 19, 2024

Revision

Date	Version	Changes
September 19, 2024	1.00	Initial release

Copyright

COPYRIGHT NOTICE

The information in this document is subject to change without prior notice in order to improve reliability, design and function and does not represent a commitment on the part of the manufacturer.

In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.

TRADEMARKS

All registered trademarks and product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective owners.

Table of Contents

Chapter

1

Introduction

1.1 Overview

The LCM Module is an advanced character display module designed for easy message display from a PC via the RS-232 (COM) port. It supports a maximum display of 2 lines with 16 characters each. The LCM Module follows the ICP Peripheral Communication Protocol (see Appendix A) for seamless communication with the PC. Additionally, it features two buttons that allow users to make selections or control functions directly.

The LCM Module, upon power-up, automatically shows an initial display page labeled "SYSTEM BOOTING." This message remains visible until the PC begins to send messages to the LCM or until the LCM is cleared. This functionality is particularly useful for operating systems that have slow boot times and cannot control the COM port immediately upon powering on.

1.2 Features

The LCM Module features are listed below:

- 16x2 characters LCM
- 2 general-purpose buttons
- Backlight deactivation via software commands
- RS-232 interface (1200, N, 8, 1)
- Automatic display of a "SYSTEM BOOTING" page upon power-on, prior to message transmission from the PC.

1.3 Pin Definition

The pinouts of the CN1 connector are shown below.

PIN NO.	DESCRIPTION		
1	VCC (+5V)	Power	5V power in
2	Rx	In	RS-232 Data Input
3	Tx	Out	RS-232 Data Output
4	GND	Power	Ground
The RS232 interface operates at a fixed baud rate of 1200, with 8 data bits, 1 stop bit, and no parity check, using a default ±12V signal. *Note: For installation of the LCM Module on a 5V signal interface (UART), please refer to Appendix B for modification instructions.*			

Table 1-1: RS-232 Connector (CN1) Pinouts

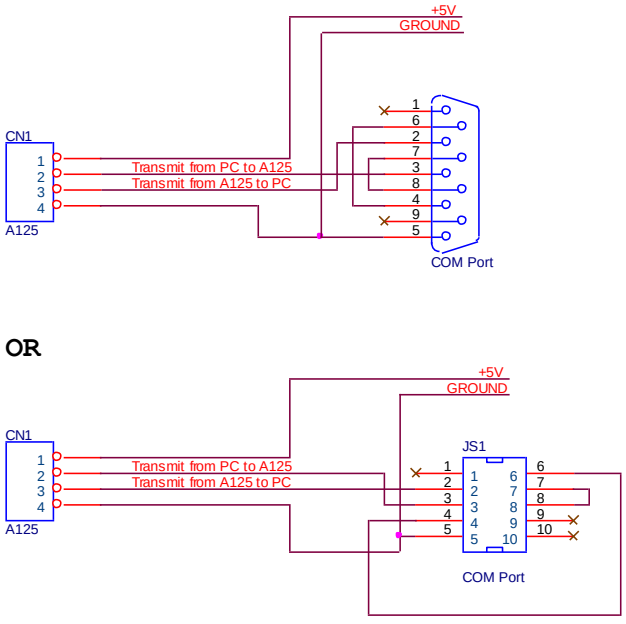


Figure 1-1: RS-232 Connector (CN1) Pinout Diagram

Chapter

2

Programming

2.1 How to Display Messages on the LCM

The LCM Module can display a wide range of characters from the ASCII code table. Below is an example of how to display text messages on the LCM.

Send to LCM Module: 0x4D 0x0D

Where:
0x4D is prefix code.
0x0D is to clear LCM.

Send to LCM MODULE: 0x4D 0x0C 0x00 0x03 0x49 0x43 0x50

Where:

0x4D	The leading code of the PC.
0x0C	Display character on LCM
0x00	Characters displayed on line0 (first line on LCM)
0x03	3 characters will be displayed
0x49 0x43 0x50	ASCII codes for ICP

The LCM first clears the screen, and then the text "ICP" is displayed in the upper-left corner, as shown below.

ICP

2.2 How to Turn Off the Backlight of the LCM

If you find the LCM backlight too bright or unnecessary, you can turn it off using the following command:

Send to LCM Module: 0x4D 0x5E 0x00
Where:
0x4D is prefix code.
0x5E controls the backlight
0x00 instructs it to turn off.

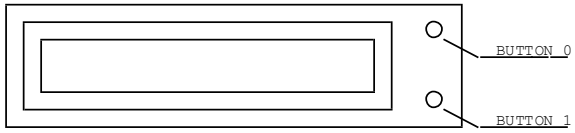
Similarly, to turn the backlight back on, send:

Send to the LCM Module: 0x4D 0x5E 0x01

This will activate the backlight again.

2.3 How to Use Buttons

When the buttons on the LCM Module are pressed or released, the module automatically transmits an electrical signal (RS232 code) to the PC immediately.



When button 0 is pressed :

LCM MODULE send to PC : 0x53 0x05 0x00 0x01;
Where
0x53 The leading code of the LCM Module.
0x05 Report button status.
0x00 Button data byte 0.
0x01 Button data byte 1, the bit 0 of byte 1 respect the button 0 is pressed.

When button 0 is released after pressed :

LCM MODULE send to PC : 0x53 0x05 0x00 0x00;

Where

0x53 The leading code of the LCM Module.
0x05 Report button status.
0x00 Button data byte 0.
0x00 Button data byte 1, the bit 0 of byte 1 respect the button 0 is released

When button 1 is pressed :

LCM MODULE send to PC : 0x53 0x05 0x00 0x02;

Where

0x53 The leading code of the LCM Module.
0x05 Report button status.
0x00 Button data byte 0.
0x02 Button data byte 1, the bit 1 of byte 1 respect the button 1 is pressed.

When button 1 is released after pressed :

LCM MODULE send to PC : 0x53 0x05 0x00 0x00;

Where

0x53 The leading code of the LCM Module.
0x05 Report button status.
0x00 Button data byte 0.
0x00 Button data byte 1, the bit 1 of byte 1 respect the button 1 is released

2.4 Initial Display Page

The LCM Module automatically displays an initial page upon powering on the PC, and **this page remains visible until the PC begins sending messages to the LCM or until the LCM is cleared.**

SYSTEM BOOTING **>>>>>>>>>>**

This feature is supported only in firmware version 1.3 and later.

2.5 Other Commands

The ICP Peripheral Communication Protocol in Appendix A consists of 12 commands, which can be categorized into two groups.

- Group A: from system to LCM module

  Get_ID (0x00)
  Get_Switches_Status (0x06)
  Get_Protocol_Version (0x07)
  Display_Character_On_LCM (0x0C)
  Clear_LCM (0x0D)
  Reset (0xFF)

- Group B: from LCM module to system

  Report_ID (0x01)
  Report_Switches_Status (0x05)
  Report_Protocol_Version (0x08)
  Ack (0xFA)
  Nack (0xFB)
  Reset_OK (0xAA)

For more details, please refer to the examples listed in Appendix A.

Appendix

A

ICP Peripheral Communication Protocol

Get ID : 0x00 and **Report ID** : 0x01

Direction	PC → LCM MODULE
Command	Get ID
Code	0x4D 0x00
Explain	0x4D=Leading Code of PC; 0x00=**Get ID** Command
Example	0x4D 0x00
Response	LCM MODULE → PC
Command	Report ID
Code	0x53 0x01 0xXX 0xYY
Emphasis	0x53=Leading Code of LCM MODULE; 0x01=Report ID; 0xXXYY=ID;
Example	0x53 0x01 0x00 0x7D (Board ID= 0x007D ---LCM MODULE)

Report Button Status : 0x05 and **Get Button Status** : 0x06

Direction	LCM MODULE →PC
Command	Report Button Status (**Auto Report** when button is pressed or released or be polling by **Get Button Status** Command)
Code	0x53 0x05 0xXX 0xYY
Explain	0x53=Leading Code of LCM MODULE; 0x05=**Report Button status** Command; 0xXXYY=Buttons on/off, XXYY<15:0>=Button<15:0>, bit0= Button0, bit1=Button1, 1=Pressed, 0=Release.
Example	0x53 0x05 0x00 0x01 (Button0 is On) 0x53 0x05 0x00 0x00 (Button0 is Off)
Direction	PC → LCM MODULE
Command	Get Button Status
Code	0x4D 0x06
Explain	0x4D=Leading Code of PC; 0x06=**Get Button status** Code;
Example	0x4D 0x06
Response	LCM MODULE →PC
Command	Report Button Status (**Auto Report** when button is pressed or released or be polling by **Get Button Status** Command)
Code	0x53 0x05 0xXX 0xYY
Explain	0x53=Leading Code of LCM MODULE; 0x05=**Report Button status** Command; 0xXXYY=Buttons on/off, XXYY<15:0>=Button<15:0>, bit0= Button0,

	bit1=Button1, 1=Pressed, 0=Release.
Example	0x53 0x05 0x00 0x01 (Button0 is On) 0x53 0x05 0x00 0x00 (Button0 is Off)

Get Protocol Version : 0x07 and **Report Protocol Version** : 0x08

Direction	PC → LCM MODULE
Command	Get Protocol Version
Code	0x4D 0x07
Explain	0x4D=Leading Code of PC; 0x07=**Get Protocol Version** Command
Example	0x4D 0x07
Response	LCM MODULE → PC
Command	Report Protocol Version
Code	0x53 0x08 0xXX 0xYY
Explain	0x53=Leading Code of LCM MODULE; 0x08=**Report Protocol Version** Command; 0xXX=Class; 0xYY=version (00~FF)
Example	0x53 0x08 0x00 0x02 (Version 02)

Display Character on LCM : 0x0C

Direction	PC → LCM MODULE
Command	Display Character on LCM
Code	0x4D 0x0C 0x0L 0x0N $0xCC_1$ ~ $0xCC_{15}$
Explain	0x4D=Leading Code of PC; 0x0C=**Display Character On LCM** Command; 0x0L=0x00 (Line 0), 0x0L=0x01 (Line 1); 0x0N=N Character (1~16), do not more than 16 characters; 0xCCn=ASCII Code of Characters,
Example	0x4D 0x0C 0x01 0x03 0x49 0x43 0x50 (Line 1, 3 Characters, 'ICP')

Clear LCM : 0x0D

Direction	PC → LCM MODULE
Command	Clear LCM
Code	0x4D 0x0D
Explain	0x4D=Leading Code of PC; 0x0D=**Clear LCM** Command
Example	0x4D 0x0D

Set Back Light On/Off : 0x5E

Direction	PC → LCM MODULE
Command	Set Back Light On/Off
Code	0x4D 0x5E 0xXX
Explain	0x4D=Leading Code of PC; 0x5E=**Set Back Light On/Off** Command, 0xXX=0x00 Back Light off, 0xXX=0x01 Back Light On
Example	0x4D 0x5E 0x01 (Back Light On) 0x4D 0x5E 0x00 (Back Light Off)

Negative Ack : 0xFB

Direction	LCM MODULE → PC
Command	Negative Acknowledge the Command from PC, (means not support)
Code	0x53 0xFB 0xXX
Explain	0x53=Leading Code of LCM MODULE; 0xFB=**Negative Ack** Command; 0xXX Command from PC;
Example	0x53 0xFB 0xF0 (NAK 0xF0 Command)

Reset : 0xFF and **Reset OK** : 0xAA

Direction	PC → LCM MODULE
Command	Reset
Code	0x4D 0xFF
Explain	0x4D=Leading Code of PC; 0xFF=**Reset** Command
Example	0x4D 0xFF
Response	LCM MODULE → PC
Command	Reset OK
Code	0x53 0xAA
Explain	0x53=Leading Code of LCM MODULE; 0xAA=**Reset OK** Command;
Example	0x53 0xAA

Appendix

B

Modifying to UART or RS-232

Modifying LCM MODULE to UART (+5V Signal) or RS232 (±12V Signal)

The internal signal from the micro processor of the LCM Module is UART 5V. To switch between UART +5V and RS-232 ±12V signal interfaces, please refer to the table below.

Components	U2	C1, 2, 3, 4, 5	R3, 4
UART	N/A	N/A	0 ohm
RS-232	RS-232 Transceiver (such as LT1381CS)	0.1uF	N/A

Appendix

C

Supported ASCII Code

Here are the ASCII codes supported by the LCM module:

	!	“	#	$	%	&	‘	(	)	*	+	,	-	.	/
0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F

0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F

@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F

P	Q	R	S	T	U	V	W	X	Y	Z	[	¥	]	^	_
0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57	0x58	0x59	0x5A	0x5B	0x5C	0x5D	0x5E	0x5F

`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67	0x68	0x69	0x6A	0x6B	0x6C	0x6D	0x6E	0x6F

p	q	r	s	t	u	v	w	x	y	z	{	|	}	→	←
0x70	0x71	0x72	0x73	0x74	0x75	0x76	0x77	0x78	0x79	0x7A	0x7B	0x7C	0x7D	0x7E	0x7F

**ASCII codes over the 0x80 are reserved for special symbols, please contact your sales representatives for details.*

Appendix

D

Sample Code for Demo Program

```
/*
 *
 *    Title                : LCM MODULE Demo program
 *    Editor               : Davis Wang in ICP Electronic
 *    Compilier            : TC 2.0
 *    OS                   : DOS 6.22 or Above
 *    Execute              : C:\LCM MODULE String0 String1
 *                           where String0 will display on LCM line 0
 *                                 String1 will display on LCM line 1
 *                           e.g.  C:\LCM MODULE ICP Electronic
 *
 */

#include <dos.h>
#include <stdio.h>
#include <conio.h>
#define COM1    0x3f8
#define COM2    0x2f8
#define IOBASE COM1



void InitUART(void){

     outport(IOBASE+3, 0x80);     /* Line Control Register */
     outport(IOBASE+0, 0x60);     /* Divisor Latch Low     */
     outport(IOBASE+1, 0x00);     /* Divisor Latch High    */
     outport(IOBASE+3, 0x03);
}


void SendByte(char ch){

     while(!(inport(IOBASE+5) & 0x20));
     outport(IOBASE, ch);
}

char GetByte(void){

     while(!(inport(IOBASE+5) & 0x01));
     return inport(IOBASE);
}


void Clear_LCM(void){

     SendByte(0x4D);
     SendByte(0x0D);
}
```

```
void SendString(int line, char *s){
int i,j;

      i=strlen(s);
      SendByte(0x4D);
      SendByte(0x0C);
      SendByte(line);
      SendByte(i);
      for(j=0; j<i; j++)SendByte(*(s+j));
}

void main(int argc, char *argv[]){


      InitUART();
      Clear_LCM();
      switch(argc){
        case 2:
                SendString(0, argv[1]);
                break;
        case 3:
                SendString(0, argv[1]);
                SendString(1, argv[2]);
                break;
      }
}
```

```
/*
 *
 *    Title     : LCM MODULE.c Demo program
 *    Editor    : Davis Wang in ICP Electronic
 *    Compilier : TCC Ver2.01
 *                Use "TCC LCM MODULE.C" to Compile LCM MODULE.c
 *    OS        : DOS 6.22 or Above
 *    Execute   : USE "C:\LCM MODULE"
 *                e.g.  C:\LCM MODULE
 *    Result    : press any button on LCM MODULE, and to see button's status
 *                reflect on PC monitor
 *
 */

#include <dos.h>
#include <stdio.h>
#include <conio.h>
#define COM1    0x3f8
#define COM2    0x2f8
#define IOBASE COM1


void InitUART(void){

     outport(IOBASE+3, 0x80);    /* Line Control Register */
     outport(IOBASE+0, 0x60);    /* Divisor Latch Low     */
     outport(IOBASE+1, 0x00);    /* Divisor Latch High    */
     outport(IOBASE+3, 0x03);
}


void SendByte(char ch){

     while(!(inport(IOBASE+5) & 0x20));
     outport(IOBASE, ch);
}


unsigned char Rs232Available(void){

     return inport(IOBASE+5)&0x01;
}


char GetByte(void){

     while(!(inport(IOBASE+5) & 0x01));
     return inport(IOBASE);
}
```

```
void SendString(int line, char *s){
int i,j;

      i=strlen(s);
      SendByte(0x4D);
      SendByte(0x0C);
      SendByte(line);
      SendByte(i);
      for(j=0; j<i; j++)SendByte(*(s+j));
}

void Read_LCM MODULE(void){
unsigned char Data1;
unsigned char Data2;

      if(GetByte()!=0x53)return;  /* LCM MODULE Leading Byte */
      switch(GetByte()){
       case 0x01: /* Report ID */
              Data1=GetByte();
              Data2=GetByte();
              printf("Board ID is 0x%04X\n", (Data1<<8)|Data2);
              break;
       case 0x05: /* Report Key Status */
              Data1=GetByte();   /* Data Byte 1 */
              Data2=GetByte();   /* Data Byte 2 */
              switch(Data2&0x03){
                  case 0x00:printf("Button is Released\n"); break;
                  case 0x01:printf("Button 0 is Pressed\n"); break;
                  case 0x02:printf("Button 1 is Pressed\n"); break;
              }
              break;
      }
}

void GetID(void){

      SendByte(0x4D);
      SendByte(0x00);
}

void main(void){

      InitUART();
      GetID();
      while(!kbhit()){
        if(Rs232Available())Read_LCM MODULE();
      }
}
```

```
/*
 *
 *   Title     : LCM MODULE.C Demo Program
 *   Editor    : Davis Wang in ICP Electronic
 *   Compilier : gcc
 *               use "gcc -o LCM MODULE LCM MODULE.c" to compile LCM MODULE.c
 *   OS        : Linux
 *   Execute   : USE "[..]$LCM MODULE String0 String1" as root
 *               Where String0 will display on LCM line0
 *                     String1 will display on LCM line1
 *               e.g. [root@localhost davis]$ ./LCM MODULE ICP Electronic
 *
 */

#include <stdio.h>
#include <sys/ioctl.h>
#include <fcntl.h>
#include <termios.h>
#include <stdlib.h>


struct termios tio;
int fd;

void InitUART(void){

    if((fd=open("/dev/ttyS0", O_RDWR|O_NDELAY|O_NOCTTY))<0){
       printf("Could not open Serial Port\n");
       exit(1);
    }
    tio.c_cflag       =B1200|CS8|CREAD|CLOCAL;
    tio.c_cc[VTIME]   =0;
    tio.c_cc[VMIN]    =0;
    tcflush(fd, TCIFLUSH);
    tcsetattr(fd, TCSANOW, &tio);
    fcntl(fd, F_SETFL, FNDELAY);
}

void Clear_LCM(void){
char s[]={0x4D, 0x0D};

    write(fd, s, 2);
}

void SendString(int line, char *s){
unsigned char c[]={0x4D, 0x0C, 0x00, 0x00};

     c[2]=line;
     c[3]=strlen(s);
     write(fd, c, 4);
     write(fd, s, strlen(s));
}
```

```
int main(int argc, char *argv[]){

    InitUART();
    Clear_LCM();
    switch(argc){
      case 2:
              SendString(0, argv[1]);
              break;
      case 3:
              SendString(0, argv[1]);
              SendString(1, argv[2]);
              break;
    }
    close(fd);
}
```

Appendix

E

Block Diagram

The function block diagram of the LCD Module is shown below.