LCD Module User's Manual
A125 Ver 0.3

Author: *Davis Wang*
ICP Electronics Inc.

INTRODUCTION

A125 is an LCD Module designed for system message display. The maximum text allowed is 16x2 characters, which are separated in 2 lines. The device is easy to install because it uses only a 2- wire RS232 interface to communicate with users' system, and 2 wires for +5V power supply and ground connectivity. Based on ICP Peripheral Communication Protocol in Appendix A, A125 also provide two readable buttons for system to access external information.

Implementation

The device comprises two major components:

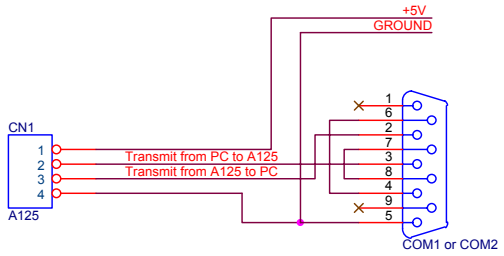
a. One A125 board

b. One UART Cable

Pin Definition

CN1	1	VCC (+5V)	Power	5V power in
	2	Rx	In	RS232 Data Input
	3	Tx	Out	RS232 Data Out
	4	GND	Power	Ground

**The RS232 uses fixed 1200 Baud Rate, 8 bit, 1 stop bit, no parity check, default ±12V Signal.
Note: If A125 want to be installed on a 5V signal transaction (UART), please refer to Appendix B for modification.



How to Displaying Messages on LCD

A125 is able to display most of the characters you can find on the ASCII code. Here is an example to display text messages on the LCD.

Send to A125: 0x4D 0x0D

Where:
0x4D is prefix code.
0x0D is Clear LCD.

Send to A125: 0x4D 0x0C 0x00 0x03 0x49 0x43 0x50

Where:
0x4D The Prefix Code.
0x0C Display Character on LCD
0x00 Characters displayed on line0 (first line on LCD)
0x03 3 characters will be displayed
0x49 0x43 0x50 ASCII codes for 'ICP'

The LCD first clear the screen and then the text 'ICP' is display on the upper-left corner of LCD like

ICP

How to Turn Off the Back Light of LCD

Sometime, if you find the back light of LCD is too bright or not necessary, you can turn off it by this way:

Send to A125 : 0x4D 0x5E 0x00

Where 0x4D the same and 0x5E is the command to turn on/off the LCD back light, and 0x00 instruct to turn off.

Also similarly

Send to A125 : 0x4D 0x5E 0x01

Will turn on the back light of LCD.

Other commands

The ICP Peripheral Communication Protocol in Appendix A comprises 12 commands and can be separated in 2 groups.

Group A: from system to A125

Get_ID (0x00)
Get_Switches_Status (0x06)
Get_Protocol_Version (0x07)
Display_Character_On_LCD (0x0C)
Clear_LCD (0x0D)
Reset (0xFF)

Group B: From System to A125

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

For more details, please refer to examples in

Appendix A

The ICP Peripheral Communication Protocol Version 0.3

History

2001/10/29	Version 0.3

Get ID : 0x00

Direction	PC → A125
Content	Get ID
Command	0x4D 0x00
Emphasis	0x4D=Prefix Code; 0x00=Get ID
Response To	None
Example	0x4D 0x00

Report ID : 0x01

Direction	A125 → PC
Content	Report ID
Command	0x53 0x01 0xXX 0xYY
Emphasis	0x53=Prefix Code; 0x04=Report ID; 0xXX, 0xYY=ID;
Response To	Get ID
Example	0x53 0x01 0x00 0x7D (Board ID= 0x007D ---A125)

Report Button Status : 0x05

Direction	A125 →PC
Content	Report Button Status (**Auto Report** when button is pressed or be polling by **GetButtonStatus** Command)
Command	0x53 0x05 0xXX 0xYY
Emphasis	0x53=Prefix Code; 0x05=Report Button status; 0xXXYY=Buttons on/off XXYY<15:0>=Button<15:0>, 1=Pressed, 0=Release
Response To	Get Button Status
Example	0x53 0x05 0x00 0x01 (Sw0 is On)

Get Button Status : 0x06

Direction	PC → A125
Content	Get Button Status
Command	0x4D 0x06
Emphasis	0x4D=Prefix Code; 0x06=Get Button status
Response To	None
Example	0x4D 0x06

Get Protocol Version : 0x07

Direction	PC → A125
Content	Get Protocol Version
Command	0x4D 0x07
Emphasis	0x4D=Prefix Code; 0x07=Get Protocol Version
Response To	None
Example	0x4D 0x07

Report Protocol Version : 0x08

Direction	A125 → PC
Content	Report Protocol Version
Command	0x53 0x08 0xXX 0xYY
Emphasis	0x53=Prefix Code; 0x08=Report Protocol Version; 0xXX=Class; 0xYY=version (00~FF)
Response To	Get Protocol Version
Example	0x53 0x08 0x00 0x02 (Version 02)

Display Character on LCD : 0x0C

Direction	PC → A125
Content	Display Character on LCD
Command	0x4D 0x0C 0x0L 0x0N $0xCC_1$ ~ $0xCC_{15}$
Emphasis	0x4D=Prefix Code; 0x0C=Display Character On LCD; 0x0L=0x00 (Line 0), 0x0L=0x01 (Line 1); 0x0N=N Character (1~15), no more than 15 characters; 0xCCn=ASCII Code of Characters,
Response To	None
Example	0x4D 0x0C 0x01 0x03 0x49 0x43 0x50 (Line 1, 3 Characters, 'ICP')

Clear LCD : 0x0D

Direction	PC → A125
Content	Clear LCD
Command	0x4D 0x0D
Emphasis	0x4D=Prefix Code; 0x0D=Clear LCD
Response To	None
Example	0x4D 0x0D

Set Back Light On/Off : 0x5E

Direction	PC → A125
Content	Set Back Light On/Off
Command	0x4D 0x5E 0xDD
Emphasis	0x4D=Prefix Code; 0x5E=Set Back Light On/Off, 0xDD=0x00 Back Light off, 0xDD=0x01 Back Light On
Response To	None
Example	0x4D 0x5E 0x01 (Back Light On)
Acknowledge	None

Negative Ack : 0xFB

Direction	A125 → PC
Content	Negative Acknowledge
Command	0x53 0xFB 0xXX
Emphasis	0x53=Prefix Code; 0xFB=Negative Ack; 0xXX Command from PC;
Response To	Ack Not Support Command 0xXX
Example	0x53 0xFB 0xF0 (NAK 0xF0 Command)

Reset : 0xFF

Direction	PC → A125
Content	Reset
Command	0x4D 0xFF
Emphasis	0x4D=Prefix Code; 0xFF=Reset
Response To	None

Example	0x4D 0xFF

Reset OK : 0xAA

Direction	A125 → PC
Content	Reset OK
Command	0x53 0xAA
Emphasis	0x53=Prefix Code; 0xAA=Reset OK;
Response To	Reset
Example	0x53 0xAA

Appendix B

Modifying A125 to UART (+5V Signal) or RS232 (±12V Signal)

The internal signal from the micro processor of A125 is UART 5V. To switch between UART +5V and RS232 ±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
RS232	RS232 Transceiver like LT1381CS	0.1uF	N/A

Appendix C

LCD-Module-supported ASCII codes

	!	“	#	$	%	&	‘	(	)	*	+	,	-	.	/
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

```
/*
 *
 *    Title              : A125 Demo program
 *    Editor             : Davis Wang in ICP Electronic
 *    Compilier          : TC 2.0
 *    OS                 : DOS 6.22 or Above
 *    Execute            : C:\A125 String0 String1
 *                         where String0 will display on LCD line 0
 *                               String1 will display on LCD line 1
 *                         e.g. C:\A125 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_LCD(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_LCD();
     switch(argc){
       case 2:
              SendString(0, argv[1]);
              break;
       case 3:
              SendString(0, argv[1]);
              SendString(1, argv[2]);
              break;
     }
}
```

Appendix E

Function Block