

LCD Module User's Manual

A78 Ver 0.3

*Author: Davis Wang
ICP Electronics Inc.*

INTRODUCTION

A78 is an LCD Module designed for system message display. The maximum text allowed is 40 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, A78 has versatile functions and provide users a battery-backup real time clock on board with default display on the right corner. The clock can be adjusted either manually by the two buttons on the right, or by

system via software. A78 provides 4 programmable LEDs and two readable buttons for system to access external information.

Implementation

The device comprises two major components:

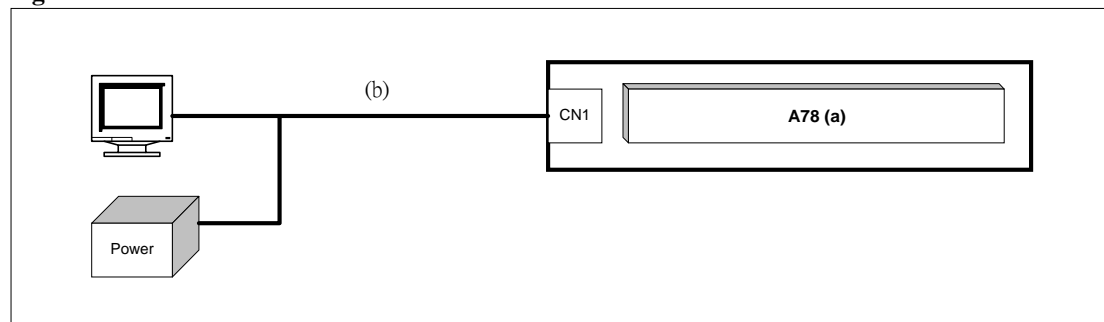
- One A78 board
- One UART Cable

For connection schematic, refer to Fig. 1.

Complete the connection and turn on the power, the LCD defaulted display the date and time on the right, for example:

12/18
17:07

Fig 1

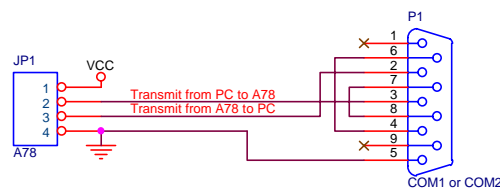


Pin Definition

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 $\pm 12V$ Signal.

Note: If A78 want to be installed on a 5V signal transaction (UART), please refer to Appendix B for modification.



How to Displaying Messages on LCD

A78 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 A78: *0x4D 0x28 (optional)*

Where

0x28 Stop LCD Auto Clock Display.

This optional command stops the LCD from displaying the clock on the right.

Send to A78: *0x4D 0x0D (optional)*

Where 0x0D Clear LCD.

Send to A78: *0x4D 0x0C 0x00 0x03 0x49 0x43 0x50*

Where:

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 stops and clear the clock display,

then the text 'ICP' is display on the upper-left corner of LCD like



The same times you can use the command 0x29 to recover the clock display on LCD.

Send to A78: *0x4D 0x29*

Where

0x29 Start LCD Auto Clock Display

The default display then re-appears on LCD.

How to Adjust The Clock on LCD

A78 has implemented a real time clock on board which provides information of the year, month, day, hour, min, and second. It is also equipped a backup battery to keep the clock running independently for over 3 years. There are 2 ways to adjust the clock; one by two buttons on the board and the other via software.

Adjusting by Buttons

Step 1: Press the upper button for over 3 seconds until a blinking cursor appears on the LCD screen.

Step 2: Press the lower button to adjust time.

Step 3: Press the upper button again to advance the blinking cursor to the next position.

Step 4: repeat Step 2 and Step 3.

Step 5: When time is set, press the upper button again for over 3 seconds to leave Edit Mode, until the cursor disappears and the clock starts running.

Adjusting via Software

Here is an example of how to adjust the clock on A78 via software.

Send to A78: 0x4D 0x09 0x00 0x0C 0x07 0x0D 0x28 0x05

While 0x4D means this command comes from System, 0x09 is the command of 'Setting Clock Time'.

And

0x00	year 2000
0x0C	December
0x07	7 th Day
0x0D	13 o'clock
0x28	40 minutes
0x05	5 seconds

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 A78 : 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 A78 : 0x4D 0x5E 0x01

Will turn on the back light of LCD.

Other commands

The ICP Peripheral Communication Protocol in Appendix A comprises 26 commands and can be separated in 2 groups.

Group A: from system to A78

Get_ID (0x00)
Set_LED_On/Off (0x02)
Get_LED_Status (0x03)
Get_Switches_Status (0x06)
Get_Protocol_Version (0x07)

Set_Clock_Time (0x09)
Get_Clock_Time (0x0B)
Display_Character_On_LCD (0x0C)
Clear_LCD (0x0D)
Stop_LCD_Self_Regular_Display (0x28)
Start_LCD_self_Regular_Display (0x29)
Set_LCD_Special_Flags (0x2B)
Get_LCD_Special_Flags (0x2C)
Set_Clock_Adjustment_Mode (0x35)
Get_Clock_Adjustment_Mode (0x36)
Reset (0xFF)

Group B: From System to A78

Report_ID (0x01)
Report_LED_Status (0x04)
Report_Switches_Status (0x05)
Report_Protocol_Version (0x08)
Report_Clock_Time (0x0A)
Report_LCD_Special_Flags (0x2D)
Report_Clock_Adjustment_Mode (0x37)
Ack (0xFA)
Nack (0xFB)
Reset_OK (0xAA)

For more details, please refer to examples in Appendix A

Note:

Due to the limited resource of MCU, the UART buffer equipped in A78 is only 16 bytes. If the commands from the system exceed 16 bytes, the data traffic is better to be separated into smaller packages, each containing no more than 16 bytes, and little delays in intervals are also needed to avoid data overflow.

CAUTION : Danger of explosion if battery is incorrectly replaced.
Replace only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instruction.

Appendix A

The ICP Peripheral Communication Protocol Version 0.02

History

2001/03/20	Version 0.01
2001/10/29	Version 0.02 Addition Command Turn On/Off Back Light (0x5E)

Get ID : 0x00

Direction	Device → LCD Board
Content	Get ID
Command	0x4D 0x00
Emphasis	0x4D='M'; 0x00=Get ID
Response To	None
Example	0x4D 0x00

Report ID : 0x01

Direction	LCD Board → Device
Content	Report ID
Command	0x53 0x01 0xXX 0xYY
Emphasis	0x53='S'; 0x04=Report ID; 0xXX, 0xYY=ID;
Response To	Get ID
Example	0x53 0x01 0x00 0x4E (Board ID= 0x004E ---A078)

Set LED On/Off : 0x02

Direction	Device → LCD Board
Content	Set LED On/Off
Command	0x4D 0x02 0xXX 0xYY
Emphasis	0x4D='M'; 0x02=Set LED On/Off; 0xXXYY=LED on/off, XXYY<15:0>=LED<15:0>, 1=On, 0=off
Response To	None
Example	0x4D 0x02 0x00 0x03 (Set LED0 & LED1 On)

Get LED Status : 0x03

Direction	Device → LCD Board
Content	Get LED Status
Command	0x4D 0x03
Emphasis	0x4D='M'; 0x03=Get LED Status
Response To	None
Example	0x4D 0x03

Report LED Status : 0x04

Direction	LCD Board → Device
Content	Report LED Status
Command	0x53 0x04 0xXX 0xYY
Emphasis	0x53='S'; 0x04=Report LED status; 0xXXYY=LED on/off XXYY<15:0>=LED<15:0>, 1=On, 0=Off
Response To	Get LED Status
Example	0x53 0x04 0x00 0x0F (LED<3:0> is On)

Report Button Status : 0x05

Direction	LCD Board → Device
Content	Report Button Status
Command	0x53 0x05 0xXX 0xYY
Emphasis	0x53='S'; 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 0x80 (Sw7 is On)

Get Button Status : 0x06

Direction	Device → LCD Board
Content	Get Button Status
Command	0x4D 0x06
Emphasis	0x4D='M'; 0x06=Get Button status
Response To	None
Example	0x4D 0x06

Get Protocol Version : 0x07

Direction	Device → LCD Board
Content	Get Protocol Version
Command	0x4D 0x07
Emphasis	0x4D='M'; 0x07=Get Protocol Version
Response To	None
Example	0x4D 0x07

Report Protocol Version : 0x08

Direction	LCD Board → Device
Content	Report Protocol Version
Command	0x53 0x08 0xXX 0xYY
Emphasis	0x53='S'; 0x08=Report Protocol Version; 0xXX=Class; 0xYY=version (00~FF)
Response To	Get Protocol Version
Example	0x53 0x08 0x00 0x02 (Version 02)

Set Clock Time : 0x09

Direction	Device → LCD Board
Content	Set Clock Time
Command	0x4D 0x09 0xYY 0xMM 0xDD 0xHH 0xmm 0xSS
Emphasis	0x4D='M'; 0x09=Set Clock Time; 0xYY=Year (00~99) Map to (2000 ~ 2099) 0xMM=Month (01 ~ 12) 0xDD=Day (01 ~ 31) 0xHH=Hour 24Hr Mode (00~23) 0xmm=Minute (00~59) 0xSS=Second (00~59)
Response To	None
Example	0x4D 0x09 0x00 0x0C 0x07 0x0D 0x28 0x05 (Set Time on 2000/12/07 13:40:05)

Report Clock Time : 0x0A

Direction	LCD Board → Device
Content	Report Clock Time
Command	0x53 0x0A 0xYY 0xMM 0xDD 0xHH 0xmm 0xSS
Emphasis	0x53='S'; 0x0A=Report Clock Time; 0xYY=Year (00 ~ 99) Map to (2000 ~ 2099) 0xMM=Month (01 ~ 12) 0xDD=Day (01 ~ 31) 0xHH=Hour 24Hr Mode (00~23)

	0xmm=Minute (00~59) 0xSS=Second (00~59)
Response To	Get Clock Time
Example	0x53 0x0A 0x00 0x0C 0x07 0x0D 0x28 0x09 (Report Time 2000/12/07 13:40:09)

Get Clock Time : 0x0B

Direction	Device → LCD Board
Content	Get Clock Time
Command	0x4D 0x0B
Emphasis	0x4D='M'; 0x0B=Get Clock Time
Response To	None
Example	0x4D 0x0B

Display Character On : 0x0C

Direction	Device → LCD Board
Content	Display Character on LCD
Command	0x4D 0x0C 0x0L 0x0N 0xCC ₁ ~ 0xCC ₁₅
Emphasis	0x4D='M'; 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	Device → LCD Board
Content	Clear LCD
Command	0x4D 0x0D
Emphasis	0x4D='M'; 0x0D=Clear LCD
Response To	None
Example	0x4D 0x0D

Stop LCD Auto Clock Display : 0x28

Direction	Device → Alm/LCD Board
Content	Stop LCD Auto Clock Display
Command	0x4D 0x28
Emphasis	0x4D='M'; 0x28=Stop LCD Auto Clock Display;
Response To	None
Example	0x4D 0x28 (Stop LCD Auto Clock Display)

Start LCD Auto Clock Display : 0x29

Direction	Device → Alm/LCD Board
Content	Start LCD Auto Clock Display
Command	0x4D 0x29
Emphasis	0x4D='M'; 0x29=Start LCD Auto Clock Display
Response To	None
Example	0x4D 0x29 (Start LCD Auto Clock Display)

Set LCD Special Flags : 0x2B

Direction	Device → LCD Board
Content	Set LCD Special Flags
Command	0x4D 0x2B 0xXX
Emphasis	0x4D='M'; 0x2B=Set LCD Special Flags; 0xXX=Flags, X ₀ = 1:Turn On Watch_Dog_Flag Display on LCD 0:Turn Off Watch_Dog_Flag Display on LCD X _{1..7} Reserved
Response To	None

Example	0x4D 0x2B 0x01 : Turn On Watch_Dog_Flag Display
---------	---

Get LCD Special Flags : 0x2C

Direction	Device → LCD Board
Content	Get LCD Special Flags
Command	0x4D 0x2C
Emphasis	0x4D='M'; 0x2C=Get LCD Special Flags
Response To	None
Example	0x4D 0x2C

Report LCD Special Flags : 0x2D

Direction	LCD Board → Device
Content	Report LCD Special Flags
Command	0x53 0x2D 0xXX
Emphasis	0x53='S'; 0x2D=Report LCD Special Flags; 0xXX=Flags
Response To	Get LCD Special Flags
Example	0x53 0x2D 0x01 : Watch_Dog_Flag display turned on now

Set Clock Adjustment Mode : 0x35

Direction	Device → Alm/LCD Board
Content	Set Clock Adjustment Mode
Command	0x4D 0x35 0xNN
Emphasis	0x4D='M'; 0x35=Set Clock Adjustment Mode; 0xNN =0x00, Disable Manual Adjustment =0x01, Enable Manual Adjustment
Response To	None
Example	0x4D 0x35 0x01 Enable Clock Manual Adjustment

Get Clock Adjustment Mode : 0x36

Direction	Device → Alm/LCD Board
Content	Get Clock Adjustment Mode
Command	0x4D 0x36
Emphasis	0x4D='M'; 0x36=Get Clock Adjustment Mode
Response To	None
Example	0x4D 0x36

Report Clock Adjustment Mode : 0x37

Direction	Alm/LCD Board → Device
Content	Report Clock Adjustment Mode
Command	0x53 0x37 0xNN
Emphasis	0x53='S'; 0x37=Report Clock Adjustment Mode 0xNN =0x00, Disable Manual Adjustment =0x01, Enable Manual Adjustment
Response To	Get Clock Adjustment Mode
Example	0x53 0x37 0x00 (Manual Adjustment disabled)

Set Back Light On/Off : 0x5E (Only support on above version 0.02)

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

Acknowledge	None
-------------	------

0xFA

Direction	LCD/LED/Alarm Board → Device
Content	Ack
Command	0x53 0xFA
Emphasis	0x53='S'; 0xFA=Ack;
Response To	N/A
Example	0x53 0xFA

0xFB

Direction	LCD/LED/Alarm Board → Device
Content	Negative Ack
Command	0x53 0xFB 0XX
Emphasis	0x53='S'; 0xFB=Negative Ack; 0XX Command;
Response To	Ack Not Supported
Example	0x53 0xFB 0xF0 (NAK 0xF0 Command)

0xFF

Direction	Device → Alm Board
Content	Reset
Command	0x4D 0xFF
Emphasis	0x4D='M'; 0xFF=Reset Slave Device
Response To	None
Example	0x4D 0xFF

0xAA

Direction	LCD/LED/Alarm Board → Device
Content	Reset OK
Command	0x53 0xAA
Emphasis	0x53='S'; 0xAA=Reset OK;
Response To	Reset
Example	0x53 0xAA

Appendix B

Modifying A78 to UART (+5V Signal) or RS232 ($\pm 12V$ Signal)

The internal signal from the micro processor of A78 is UART 5V. To switch between UART +5V and RS232 $\pm 12V$ signal interfaces, please refer to the table below.

Components	U3	C8,9,10,11,12	R22,23
UART	N/A	N/A	0 ohm or jumper
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           : A78 Demo program
 *   Editor          : Davis Wang in ICP Electronic
 *   Compiler        : TCC Ver2.0
 *                   Use "TCC A78.c" to Compile A78.c
 *   OS              : DOS 6.22 or Above
 *   Execute         : Use "C:\A78 String0 String1"
 *                   where String0 will display on LCD line 0
 *                   String1 will display on LCD line 1
 *                   e.g. C:\A78 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;
    }
}
```