Contents

Title **Page**

Content (Continued)

Title **Page**

ET- REMOTE GLCD12864 V1.0

1. Specifications of Board ET-REMOTE GLCD12864 V1.0

1.1 128x64 Dot DISPLAY with Display Board Controller and send ASCII Command

1.2 Be able to send command to control operation through Port RS232, RS422 and 4-Line RS485

1.3 Baud Rate for communication can be set by DIP.SW: 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200

1.4 Be able to set ID into Board by DIP.SW in the range of '00-FF' to use network communication by RS232

1.5 Be able to create Notes and control ON/OFF of Back Light

1.6 VR to adjust the contrast of GLCD and including LED to display states of the command

1.7 Internal Self-Test and Demo system

1.8 AC/DC 7-12V Power Supply

1.9 Be able invert DISPLAY to show the contrasting result or be different from the current result

1.10 There are 3 Modes: Text Mode, Graphic Mode and Picture Mode that can be used together.

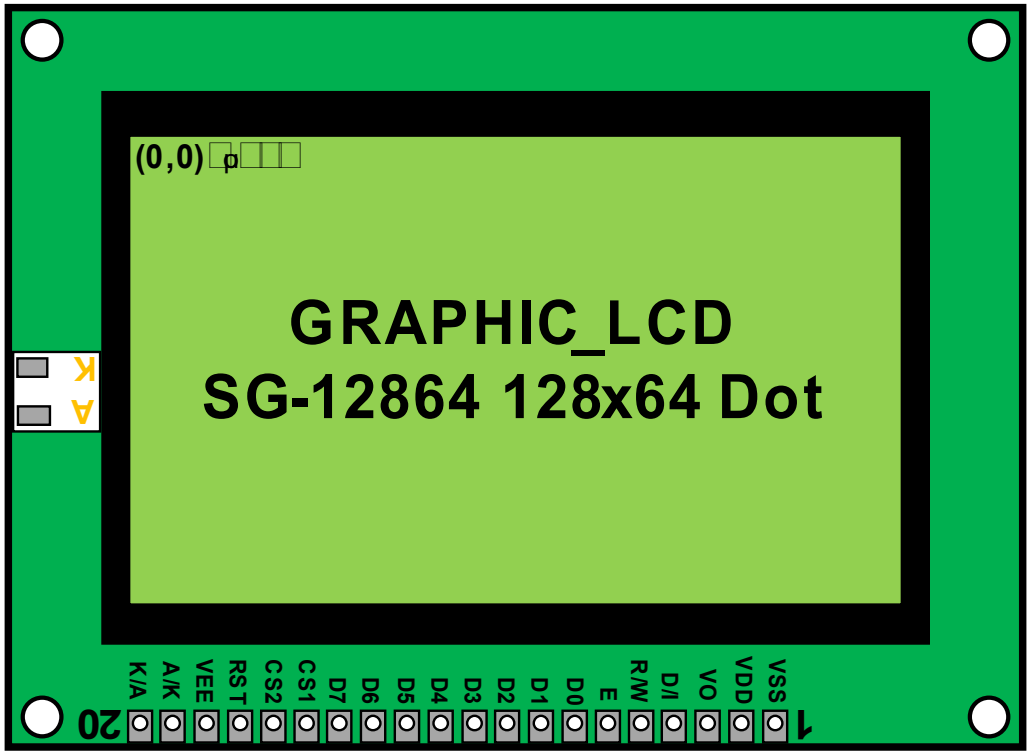
1.11 Text Mode

- Be able to display both Thai and English alphabets, so it does not waste time to create Font by self.
- Size of English alphabet is 5x7 Dot (width x height) and size of Thai alphabet is 7 Dot but its width depends on the alphabet type to make it beautiful font. This font that is displayed is the same as MS Sans Serif of Windows.
- Be able to set any position on monitor to display alphabet; normally, one monitor can be displayed messages about 4 lines or more. If we want to know that how many alphabets can be displayed within one line, it is not certain data because it depends on font width that is received and each alphabet has different width.
- Be able to delete alphabet in any position on monitor.
- Be able to send Text about 200 alphabets in each time (alphabet + vowel + pitch (tone mark in Thai writing)
- Be able to align the received message as right justification, left justification or centre justification of the line (the exceeding message will be abandoned)
- Be able to start the new line automatically when message is complete in a line.
- When messages are full of a monitor, the exceeding message will be abandoned and we can command to clear monitor.

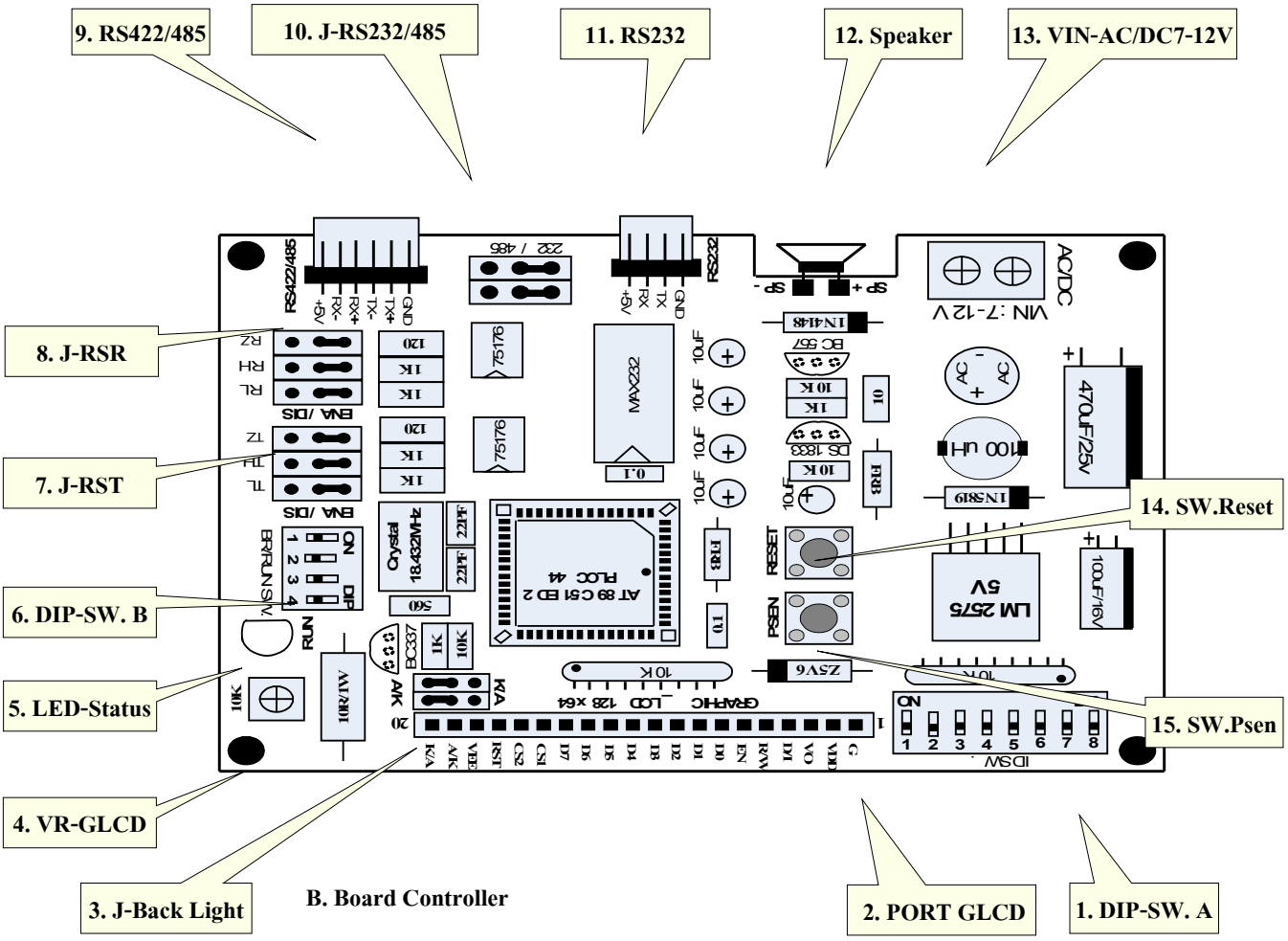
1.12 Graphic Mode and Picture Mode

- Be able to create any triangle, quadrilateral, circle, ellipse and straight line in any size and any position on monitor as desired
- Be able to fill whole picture with black color or configure it to display as line
- Be able to use and convert external Photo File that is 128x64 Dot BIT MAP File into Code first and then send that Code to Board and it makes that picture display on monitor.

2. Structure of Board ET-REMOTE GLCD12864 V1.0



A. Graphic LCD 128x64 Dot



B. Board Controller

Figure 2.1 displays structure of Board ET-REMOTE GLCD12864 V1.0.

For structure of Board ET-REMOTE GLCD12864 V1.0 is divided into 2 parts; the first part is 128x64 Dot Graphic LCD. There are 20 pins and Back Light for usage as show in the figure A. The second part is Board Controller to receive command from user through RS232/RS422 and 4-Line RS485 as shown in the figure B. Functions in Board Controller can be summarized as follows;

No.1 DIP-SW.A: It is DIP SW. to set ID into board in the range of ID '00'-'FF'. This ID value is useful when using many boards simultaneously as Network for RS485 Communication. We can set ID as shown in the table below;

Table 2.1 DIP SW.A for SET ID.BOARD ('00'-'FF')

Number of DIP-SW.A								ID CODE (ASCII- 2 byte)
S8	S7	S6	S5	S4	S3	S2	S1	
OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	' 00 '
OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON	' 01 '
OFF	OFF	OFF	OFF	OFF	OFF	ON	OFF	' 02 '
OFF	OFF	OFF	OFF	OFF	OFF	ON	ON	' 03 '
OFF	OFF	OFF	OFF	OFF	ON	OFF	OFF	' 04 '
OFF	OFF	OFF	OFF	OFF	ON	OFF	ON	' 05 '
OFF	OFF	OFF	OFF	OFF	ON	ON	OFF	' 06 '
OFF	OFF	OFF	OFF	OFF	ON	ON	ON	' 07 '
OFF	OFF	OFF	OFF	ON	OFF	OFF	OFF	' 08 '
OFF	OFF	OFF	OFF	ON	OFF	OFF	ON	' 09 '
OFF	OFF	OFF	OFF	ON	OFF	ON	OFF	' 0A '
OFF	OFF	OFF	OFF	ON	OFF	ON	ON	' 0B '
OFF	OFF	OFF	OFF	ON	ON	OFF	OFF	' 0C '
OFF	OFF	OFF	OFF	ON	ON	OFF	ON	' 0D '
OFF	OFF	OFF	OFF	ON	ON	ON	OFF	' 0E '
OFF	OFF	OFF	OFF	ON	ON	ON	ON	' 0F '
OFF	OFF	OFF	ON	OFF	OFF	OFF	OFF	' 10 '
↓			↓	↓			↓	↓
ON	ON	ON	ON	ON	ON	OFF	OFF	' FC '
ON	ON	ON	ON	ON	ON	ON	OFF	' FE '
ON	ON	ON	ON	ON	ON	ON	ON	' FF '

The method to read ID value from this DIP.SW is to read it as Code BCD8421. We will set S8-S5 to be 4 Bit High and S4-S1 to be 4 Bit Low; when we can read this value, take the Code to join together and we will get ID as desired. If we want to use this value, look at this ID in the format of ASCII CODE that is divided into 2 Byte. For example, ID '00'; when it is used in Command, it can be replaced by ASCII '00" or hexadecimal number that is 0x30,0x30 or ID 'FC' = 0x46,0x43. User can compare ASCII CODE with hexadecimal number from the table at the end of this handbook. If user can not read ID value from DIP SW.A, user can see ID value from GLCD monitor by shifting DIP-SW.B No.4 to ON position and it will display ID value of board to user instantly.

NOTE: Every time when ID of board is changed, the changed ID will be affected when Reset Board or shifting DIP-SW.B No.4 to ON position and wait for ID value on GLCD monitor is updated to be the new value.

No.2 PORT GLCD: It is 20 Pin Connector to interface with Graphic LCD and its pin arrangement is shown as in the picture above.

No.3 J-Back Light: It is 2 pairs of Jumper to switch Back Light Connector of GLCD at Pin 19 and Pin 20. Due to pins of some GLCD version are altered, so it is different from the pin arrangement that is shown in this figure. We must set pin as follows; if Pin 19 is Connector A and Pin 20 is Connector K, set both Jumper to A/K side but if Pin 19 is Connector K and Pin 20 is Connector A, set both Jumper to K/A side instead.

No.4 VR-GLCD: It is VR to adjust the contrast of GLCD monitor.

No.5 LED Status: There are 2 functions for this LED. Firstly, it will be ON when running in Self-Test Mode, and then be OFF when the operation is finishes; moreover, it will be ON again when restarting this mode again. Secondly, in Run Mode if commands are sent to board correctly and board starts running follow the command; it makes LED ON and this LED will be OFF when operation of the command finishes. If LED is not ON or remains ON not OFF, it means that the format of command that is sent is not correct or command is done incompletely, so we should send 0x0D or Enter to Clear Status LED before sending the new command.

No.6 DIP-SW.B: It is DIP.SW to set Baud Rate for receiving/sending data into board and select modes between Self-Test Mode and Run Mode. SW No.1 - No.3 are used to set 8 values of Baud Rate as shown in the table 2.3; and SW No.4 is used to select modes between Self-Test Mode and Run Mode as shown in the table 2.

Table 2.2 DIP-SW.B to select modes

Number of DIP-SW.B	Function
S4	
OFF	Operation of Run Mode starts running to wait for receiving command from user.
ON	Self-Test Mode starts running to test operation of board, GLCD monitor and display ID value and Baud Rate that is set into board

Table 2.3 DIP-SW.B to set Baud Rate (1200-115200 bit/s)

Number of DIP-SW.B			Baud Rate (bit/s)
S3	S2	S1	
OFF	OFF	OFF	1200
OFF	OFF	ON	2400
OFF	ON	OFF	4800
OFF	ON	ON	9600
ON	OFF	OFF	19200
ON	OFF	ON	38400
ON	ON	OFF	57600
ON	ON	ON	115200

In this case, there are 2 methods to see Baud Rate. Firstly, we can compare DIP-SW position with the table; and secondly, we can see Baud Rate that is set into board through GLCD monitor by setting DIP SW.B No.4 to ON position to enter Self-Test Mode and it will display Baud Rate on monitor instantly.

NOTE: Every time when Baud Rate is changed, the changed value will be affected when Reset Board or shifting DIP-SW.B No.4 to ON position and wait for Baud Rate on GLCD monitor is updated to be the new value.

No.7, 8 J-RST, J-RSR: It is Jumper to increase distance of RS422/485 Communication longer by setting Jumper RL, RH, RZ and TL, TH, TZ to Enable side.

No.9 RS422/485: It is 6 PIN Connector to interface RS422/485 Cable if communicating or sending external Command to control board and its pin arrangement is shown as in the figure 2.2 below.

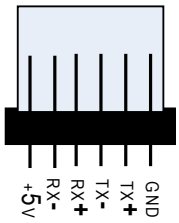


Figure 2.2 displays pin arrangement of RS422/485 Connector.

No.10 J-RS232/485: IT is 2 pairs of Jumper to alter communication types between RS232 and RS422/485. If using RS232 Communication, set both Jumpers to RS232 side but if using RS422/485 Communication, set both Jumpers to RS422/485 side instead.

No.11 RS232: It is 4 PIN Connector to interface RS232 Cable when communicating or sending external Command to control board and its pin arrangement is shown as in the figure 2.3 below.

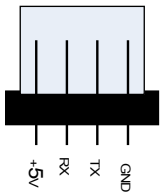


Figure 2.3 displays pin arrangement of RS232 Connector.

No.12 Speaker: This speaker produces sound when we start supplying power into board and entering to Self-Test Mode. Moreover, we can send command to control sound at the speaker corresponding with musical notes.

No.13 VIN-AC/DC 7-12V: It is Connector of Power Supply that can support both AC and DC 7-12V (No connector).

No.14 SW.Reset: It is Switch for Reset MCU to restart operation when board hangs up; moreover, it can be used with SW.PSEN to download program to update new Firmware.

No.15 SW.PSEN: This Switch is used with SW.Reset to download program to update new Firmware. We must follow these instructions; press and hold SW.PSEN, press SW.Reset, then remove SW.Reset and finally remove SW.PSEN. Next, we can start downloading new Firmware; normally, this process will be done by ETT.

3. Operation of Board ET-REMOTE GLCD12864

When we start supplying power into board, musical sound will be loud; it means that board is opened. Next, board will set ID value and Baud Rate according to values that are set by user from DIP-SW.A and DIP-SW.B respectively. After the end of the musical sound, board will send Respond Command Sync Byte through RS232/422/485 as the format below to notify user to know that board is ready to receive command. If user does not check Sync Byte before opening board, user maybe use time delay for board that sends command, otherwise the command that is sent out has not any effect on the operation if board is not ready to receive command. The method to check Sync Byte can be used in case of using only one Board GLCD but if interfacing many boards together, it makes a lot of Respond Sync Byte be sent simultaneously and board that receives data can not read data correctly.

∗ xx=OK **< ---------- Respond Sync Byte Command 6 Byte**

When xx=ID of board is displayed as 2 digits of ASCII ('00'-'FF').

After board has already sent Command Sync Byte, it will enter mode that user has already set from DIP-SW.B No4 instantly. There are 2 modes as follows;

3.1 Self-Test Mode: When DIP-SW.B No.4 is ON position, program will enter this mode. The purpose of this mode is to test operation of Board Controller and GLCD; moreover, it displays ID value and Baud Rate that is configured by user. So, it is quite convenient for user because it saves time and user can see these values instantly without comparing with position of DIP SW. from the table above.

When program starts running in this mode, the musical sound will be loud, LED will be ON and GLCD monitor will display ID value and Baud Rate for a while. When LED is OFF, it means that this mode finishes and this program will be configured to restart operation as long as DIP-SW.B No.4 still is at ON position. While

it is being in this mode, we can set ID and Baud Rate. If value that is set on GLCD monitor is changed, it means that ID value and Baud Rate have already updated completely and it can return to RUN Mode to receive command from user by using the new ID value and Baud Rate instantly.

3.2 RUN Mode: When DIP-SW.B No.4 is in OFF position, program will enter this mode and its function is to receive command from user through RS232, RS422 and 4-Line RS485. When entering to this mode, GLCD monitor will be cleared as blank, so we can send command to control operation of GLCD instantly. User can see additional format of command and example applications from section 4. When we send command into board; if command is correct and board starts running follows the command, LED will be ON and if the command is complete, LED will be OFF. On the other hand, if command has already sent out but LED is still OFF, it means that command is not correct; or LED holds in ON status, it means that operation of that command is error, so we must send value 0x0D or press key Enter to Reset command or Reset board if LED still holds.

Every time when command is done by program completely, it always sends Respond End Command to user through RS232/422 or 485 and the format of Respond End Command is shown below;

∗ xx#yy=OK **< --------- Respond End Command 9 Byte**

When xx=ID of board is displayed as 2 digits of ASCII ('00'-'FF'), yy=Command number that is sent out is displayed as 2 digits of ASCII ('00'-'34').

The Command that is sent out is useful to user because it will be notify user to send the next command. *When we have already sent the first command, we always checks End Command before sending the next command to board.* Otherwise, it makes the second command error and it makes program do the first command incorrectly because of the different time of each command. So, it should not be use delay before sending the next command, but it is better if checking End Command instead. Please see example programs in the section of example programs.

NOTE: *While it is being in **RUN Mode** or **Self-Test Mode**, we can switch DIP-SW.B No.4 from one mode to another mode. For **RUN Mode**, if user switches mode while program is in the processing or doing command incompletely, it does not any effect on the operation because the program continues to do the command until it is complete and then it will switch mode from the current mode to the new mode that is selected by user.*

4. Format and Application of commands that are supported by Board ET-REMOTE GLCD12864

Format of Command and Data will be in all ASCII Code and it can be replaced by ASCII symbol such as '*' or ASCII Code that equals 0x2A, '0'=0x30, 'A'=0x41.

In the part of Byte Start must be replaced by 0x1B (Key Esc) and Byte End must be replaced by 0x0D (Key Enter) because these 2 bytes have not any ASCII symbol, so it must be replaced by only ASCII CODE. In the case of ID that is 'A-F' alphabet, it must be used only capital letter.

There are 21 commands to communicate with Board ET-REMOTE GLCD12864; we will divide it into 4 groups follow its function to understand better.

4.1 Commands Set regarding Control Display and devices on board

COMMAND '00' (Clear Screen)

It is command to clear GLCD monitor as blank. If using this command in the normal status, the monitor will be cleared as white as nothing (blank). On the other hand, if Invert value is set by command '01' to be '1' and this command is used, the monitor will be cleared as black instead. The format of command is shown as in the table below.

Start	ID	Mark1	Command	Mark2	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	00	=	Enter (0x0D)
Respond Command from Board					
*	00-FF	#	00	=	OK

Ex1. Set ID = '9A'

```
main()
 {
   char  esc = 0x1B , enter = 0x0D  ;

   printf("%c9A#00=%c",esc,enter)   ;// Clear Screen
 }
```

COMMAND '01' (Invert Screen)

It is command to invert data before displaying on the monitor and its result will be different from the current status. We must set value in the Invert blank of the command; if setting it as '1', it will invert (black screen), but if setting it as '0', it will be the normal status (white screen (default)). If using this command and setting value in the Invert blank to be '1', it makes data on the monitor unchanged until data is sent out to plot on the monitor; in this case, we will see the result of inverted value. If monitor has already displayed data and we want to invert data that is displayed on monitor, we must send this command first and then follow by Display Screen command (Command No.'02'), it makes us be able to see result of the inverted value without sending the same data again. If we want to cancel Invert Display proceeding, we must send this command again and set value in the Invert blank to be '0'. The format of command is shown as in the table below.

Start	ID	Mark1	Command	Mark2	Invert	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	01	=	0-1	Enter (0x0D)
Respond Command from Board						
*	00-FF	#	01	=	OK	

Invert blank = '0' : Normal (default) , '1' : Invert

Ex2. Set ID = '5F'

```
main()
{
   char  esc = 0x1B , enter = 0x0D  ;

   printf("%c5F#01=1%c",esc,enter)  ; // Invert Display

   printf("%c5F#01=0%c",esc,enter)  ; // Cancel Invert Display

}
```

COMMAND '02' (Display Screen)

It is command to display data through GLCD monitor. The purpose of this command is to use with command Invert together; it means that when Invert command is sent out, the old data that is displayed on monitor is not updated or changed, if we want to update or change this data, we must send the same data again. This command will solve this problem; after Invert command is sent out,

we must send this command instantly, it makes the monitor be updated instantly without sending the same data again. The format of command is shown as shown in the table below.

Start	ID	Mark1	Command	Mark2	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	02	=	Enter (0x0D)
Respond Command from Board					
*	00-FF	#	02	=	OK

Ex3. Set ID = 'AA'

```
main()
 {
    char  esc = 0x1B , enter = 0x0D  ;

    printf("%cAA#02=%c",esc,enter)  ; //Display Screen
 }
```

COMMAND '03' (On/Off Back Light)

It is command to ON or OFF Back Light of GLCD. We must set value in the Back Light blank; value '1'= ON (default) and '0'=OFF. Before using this command, we must check whether Jumper No.3 is set to Back Light position of GLCD correctly or not. The format of command is shown as shown in the table below.

Start	ID	Mark1	Command	Mark2	Back Light	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	03	=	0-1	Enter (0x0D)
Respond Command from Board						
*	00-FF	#	03	=	OK	

Back Light blank = '0' : OFF Back Light , '1' : ON Back Light (default)

Ex4. Set ID = '55'

```
main()
 {
    char  esc = 0x1B , enter = 0x0D  ;

    printf("%c55#03=1%c",esc,enter)  ; //Back Light ON
    printf("%c55#03=0%c",esc,enter)  ; //Back Light OFF
 }
```

COMMAND '04' (Sound)

It is command to produce sound through speaker as musical notes; Do, Re, Mi, Fa, Sol, La and Ti. In this case, we can select sound of musical notes and configure the length of its sound. We can configure musical notes in the Note blank in the range of '0-7' and configure the musical length in the Delay blank in the range of '0-9'. The format of command is shown as shown in the table below.

Start	ID	Mark1	Command	Mark2	Note	Mark3	Delay	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte	1byte	1 byte
Esc (0x1B)	00-FF	#	04	=	0-7	,	0-9	Enter (0x0D)
Respond Command from Board								
*	00-FF	#	04	=	OK			

Note blank = '0': No Sound, '1': Do (524Hz), '2': Re (587Hz), '3': Mi (659Hz), '4': Fa (698Hz), '5': Sol (784Hz), '6': La (880Hz), '7': Ti (988Hz)

Delay blank = '0': The shortest of sound length and its sound length will be increased follow the number and '9': The longest of sound length

Ex5. Set ID = '00'

```
main()
 {
    char  esc = 0x1B , enter = 0x0D    ;

    printf("%c00#04=1,4%c",esc,enter)  ; //Produce Do note with 4 sound
                                           length
 }
```

4.2 Command Set regarding Graphic Mode

Command in this group is to draw a picture as triangle, rectangle or quadrilateral, polygon (several corners not over than 9 corners), circle, ellipse, dot, straight line, horizontal line, vertical line, and diagonal line. These lines can be drawn to display in any position on the monitor by specifying position x,y of picture into the command. Value that is configured to x position must be 3 Byte ASCII that is number 0-9 ('000-127') and y position must be 2 Byte ASCII that is number 0-9 ('00-31'). We can see position of points on the monitor from the table in the appendix; it is quite convenient to configure position easier. In

this case, the left-top corner of the GLCD monitor is the beginning position x,y = (0,0).

The characteristic of picture that is drawn is the line and can be filled with black color into whole picture, except several corners but we can configure its thick line. It takes a few times to fill black color into the whole picture, especially triangle, so it will waste the time to send the next command.

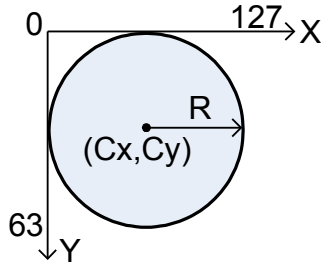
The method to plot picture to display on the monitor is to overlap the old data that is displayed on the monitor. If any dot on the monitor is not yet plotted by the new picture, it is not deleted but it is still displayed on the monitor. It is good because we can write message in the picture that has already drawn completely; in this case, we must configure position of picture or the message not overlap the old position that is used to display result.

COMMAND '10' (Plot Circle)

It is command to draw the circle as line and we must configure parameter value into this command as follows;

1. Configure Center position in the Cx blank (3 Byte: '000-127') and Cy blank (2 Byte: '00-63').
2. Radius of circle in Radius blank (2 Byte: '00-31').
3. Fill value in Fill blank; when '1' = Fill (to display whole picture with black color); '0' = Not Fill (to display only black line). If we have already set Invert command (command '01') and then follow by this command, the characteristic of picture will be opposite of the description mentioned above.

Start	ID	Mk1	Command	Mk2	Cx	Mk3	Cy	Mk4	Radius	Mk5	Fill	END
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	10	=	000-127	,	00-63	,	00-31	,	0-1	Enter (0x0D)
Respond Command from Board												
*	00-FF	#	10	=	OK							



Ex5. Set ID = '01' Plot circle that has centre position at Cx = the 60th dot, Cy = the 30th dot, the length of radius is 20 dot, and Not Fill

```
main()
 {
    char  esc = 0x1B , enter = 0x0D    ;
    printf("%c01#10=060,30,20,0%c",esc,enter)  ; // Plot circle
 }
```

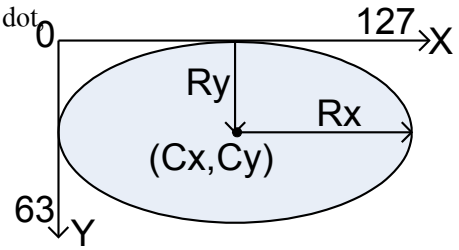
COMMAND '11' (Plot Ellipse)

It is command to draw the ellipse as line. If we want to draw this ellipse in the horizontal line, we must configure value as Rx > Ry but if we want to draw this ellipse in the vertical line, we must configure value as Ry > Rx. In this case, we must configure parameter value into this command as follows;

1. Configure Centre position in Cx blank (3 Byte: '000-127') and Cy blank (2 Byte: '00-63').
2. Radius of ellipse on the X axis in the Rx blank (2 Byte: '00-64').
3. Radius of ellipse on the Y axis in the Ry blank (2 Byte: '00-32').
4. Fill value in Fill blank; when '1' = Fill (to display whole picture with black color); '0' = Not Fill (to display only black line). If we have already set Invert command (command '01') and then follow by this command, the characteristic of picture will be opposite of the description mentioned above.

Start	ID	Mk1	Command	Mk2	Cx	Mk3	Cy	Mk4	Rx	Mk5	Ry	Mk6	Fill	END
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2byte	1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	11	=	000-127	,	00-63	,	00-64	,	00-32	,	0-1	Enter (0x0D)
Respond Command from Board														
*	00-FF	#	11	=	OK									

Ex6. Set ID = '02' Plot ellipse that has centre position at Cx = the 63rd dot, Cy = the 31st dot, the length of radius on the X axis: Rx = 64 dot and the length of radius on Y axis: Ry = 32 dot and Not Fill




```
main()
 {
    char  esc = 0x1B , enter = 0x0D    ;
    printf("%c02#11=063,31,64,32,0%c",esc,enter)  ; // Plot ellipse
 }
```

COMMAND '12' (Plot Dot)

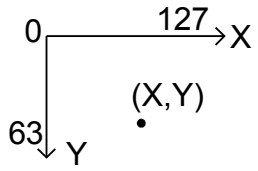
It is command to plot 1 dot on the monitor in the specified position. If sending command once time, it equals plotting 1 dot. We must configure parameter value into this command as follows;

1. Configure position on X axis in X blank (3 Byte: '000-127').
2. Configure position on Y axis in Y blank (2 Byte: '00-63').

If Invert command (command '01') is set and then follow by this command, the characteristic of dot will be displayed as white dot.

Start	ID	Mark1	Command	Mark2	X	Mark3	Y	END
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2byte	1 byte
Esc (0x1B)	00-FF	#	12	=	000-127	,	00-63	Enter (0x0D)
Respond Command from Board								
*	00-FF	#	12	=	OK			

Ex7. Set ID = '03' Plot dot that has position at X= the 30th dot, Y=the 15th dot




```
main()
 {
    char  esc = 0x1B , enter = 0x0D    ;
    printf("%c03#12=030,15%c",esc,enter)  ; // Plot  dot 1 dot
 }
```

COMMAND '13' (Plot Triangle)

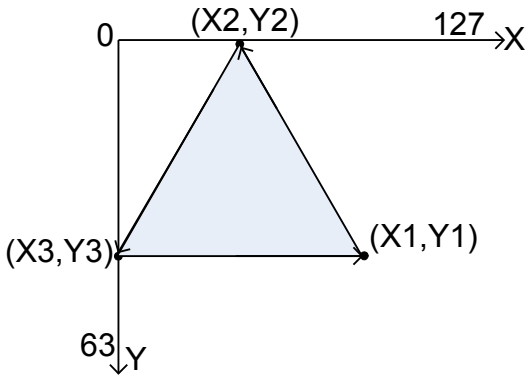
It is command to draw triangle as line. We must configure 3 positions (x,y) of triangle. The process of program is to draw a line from the first position to the second position, from the second position to the third position and from the third position to the first position and finally we will get the triangle. So, if we want to get the triangle in any type, we must configure position to draw picture of program as mentioned above. We must configure parameter value into this command as follows;

1. Configure the first position in the X1 blank (3 Byte: '000-127') and in the Y1 blank (2 Byte: '00-63').
2. Configure the second position in the X2 blank (3 Byte: '000-127') and in the Y2 blank (2 Byte: '00-63').
3. Configure the third position in the X3 blank (3 Byte: '000-127') and in the Y3 blank (2 Byte: '00-63').
4. Fill value in Fill blank; when '1' = Fill (to display whole picture with black color); '0' = Not Fill (to display only black line). If Invert command (command '01') is set and then follow by this command, the characteristic of picture will be opposite of the description mentioned above. It is quite slower if filling triangle with color and it maybe fills a partial picture because there is some white dot.

Start	ID	Mk1	Command	Mk2	X1	Mk3	Y1	Mk4	X2	Mk5	Y2	Mk6
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2byte	1 byte	3 byte	1 byte	2 byte	1 byte
Esc (0x1B)	00-FF	#	13	=	000-127	,	00-63	,	000-127	,	00-63	,
Respond Command from Board												
*	00-FF	#	13	=	OK							

(Continue)

X3	Mk7	Y3	Mk8	Fill	END
3 byte	1 byte	2byte	1 byte	1 byte	1 byte
000-127	,	00-63	,	0-1	Enter (0x0D)



Ex8. Set ID = '04' Plot triangle that has position as (X1 = the 100th dot, Y1 = the 60th dot), (X2 = the 50th dot, Y2 = the 0 dot), (X3 = the 0 dot, Y3 = the 60th dot) and Not Fill

```
main()
 {
    char  esc = 0x1B , enter = 0x0D    ;
    printf("%c04#13=100,60,050,00,000,60,0%c",esc,enter) ; // Plot
triangle
 }
```

COMMAND '14' (Plot Rectangle)

It is command to draw rectangle or square as line. We must configure 2 positions (x,y) of rectangle; the first position is the left-top corner of the picture and the second position is the right-bottom corner of the picture.

Moreover, this command can be used to delete message or picture in any position on monitor by creating rectangle or quadrilateral override in the desired area to erase. In this case, we must configure value in Fill blank to be '1' if setting it as black screen saver or '2' if setting it as white screen saver. We must configure parameter value into this command as follows;

1. Configure the first position in X1 blank (3 Byte: '000-127') and Y1 blank (2 Byte: '00-63').
2. Configure the second position in X2 blank (3 Byte: '000-127') and Y2 blank (2 Byte: '00-63').
3. Fill value in Fill blank; when '0' = Not Fill (to display only black line). If user has already set Invert command (command '01'); '1' = Fill Black (to display whole picture with black color); '2' = Fill White (to display whole picture with white color without any line). If Invert command is set and then follow by this command, the characteristic of picture will be opposite when value in the Fill blank is '0' only. If value in Fill blank is '1' or '2', Invert command is not affected on the picture.

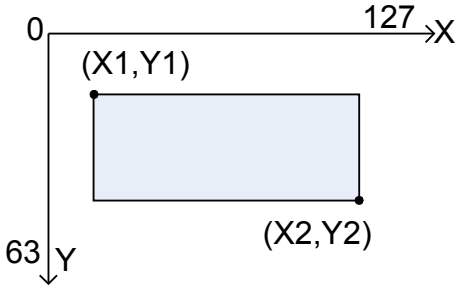
Start	ID	Mk1	Command	Mk2	X1	Mk3	Y1	Mk4	X2	Mk5	Y2	Mk6	Fill	END
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2byte	1 byte	3 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	14	=	000-127	,	00-63	,	000-127	,	00-63	,	0-2	Enter (0x0D)
Respond Command from Board														
*	00-FF	#	14	=	OK									

Fill blank '0': Line, '1': Whole black picture, '2': Whole white picture without any line

Ex9. Set ID = '05' Plot rectangle that has position as

(X1= the 10th dot, Y1 = the 10th dot) and

(X2 = the 100th dot, Y2 = the 30th dot), Not Fill




```
main()
{
  char  esc = 0x1B , enter = 0x0D    ;
  printf("%c05#14=010,10,100,30,0%c",esc,enter)  ; // Plot rectangle
}
```

COMMAND '15' (Plot Line)

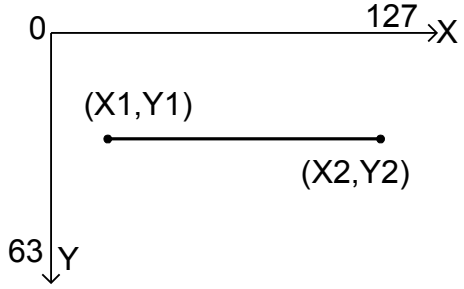
It is command to draw a straight line as horizontal line, vertical line or diagonal line depends on position that is configured in the command. In this case, we must configure 2 positions (x,y) that are the beginning and the end of the line into this command. We must configure parameter value into this command as follows;

1. Configure the first position (the beginning) in the X1 blank (3 Byte: '000-127') and Y1 blank (2 Byte: '00-63').
2. Configure the second position (the end) in the X2 blank (3 Byte: '000-127') and Y2 blank (2 Byte: '00-63').

If Invert command (command '01') is se and then follow by this command, the characteristic of line will be changed to white color.

Start	ID	Mk1	Command	Mk2	X1	Mk3	Y1	Mk4	X2	Mk5	Y2	END
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2byte	1 byte	3 byte	1 byte	2 byte	1 byte
Esc (0x1B)	00-FF	#	15	=	000-127	,	00-63	,	000-127	,	00-63	Enter (0x0D)
Respond Command from Board												
*	00-FF	#	15	=	OK							

Ex10. Set ID = '06' Plot the horizontal straight line that has position as (X1 = the 5th dot, Y1 = the 30th dot) and (X2 = the 120th dot, Y2 = the 30th dot)




```
main()
{
  char  esc = 0x1B , enter = 0x0D    ;
  printf("%c06#15=005,30,120,30%c",esc,enter)  ; // Plot straight line
}
```

COMMAND '16' (Plot Line Thick)

It is command to draw a straight line as horizontal line, vertical line or diagonal line depends on position that is configured in the command; moreover, there are 5 levels to configure the thick of line. In this case, we must configure 2 positions (x,y) that are the beginning and the end of the line into this command. We must configure parameter value into this command as follows;

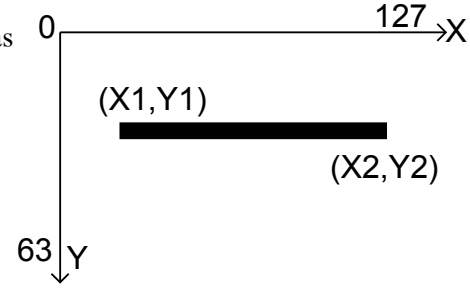
1. Configure the first position (the beginning) in the X1 blank (3 Byte: '000-127') and Y1 blank (2 Byte: '00-63').
2. Configure the second position (the end) in the X2 blank (3 Byte: '000-127') and Y2 blank (2 Byte: '00-63').
3. Configure value of thick line in the Thick blank as 1 Byte; in this case, we can configure the thick value in the range of '0'-'4'.

If Invert command (command '01') is set and then follow by this command, the characteristic of line will be changed to white color.

Start	ID	Mk1	Command	Mk2	X1	Mk3	Y1	Mk4	X2	Mk5	Y2	Mk6	Thick	END
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2byte	1 byte	3 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	16	=	000-127	,	00-63	,	000-127	,	00-63	,	0-4	Enter (0x0D)
Respond Command from Board														
*	00-FF	#	16	=	OK									

Remember: If drawing the vertical straight line with any thick level, the width of line is always 1 dot.

Ex11. Set ID = '06' Plot the horizontal straight line that has position as (X1 = the 5th dot, Y1 = the 30th dot) and (X2 = the 120th dot, Y2 = the 30th dot), and then Configure the level 2 for the thick of line




```
main()
{
  char  esc = 0x1B , enter = 0x0D    ;
  printf("%c06#16=005,30,120,30,2%c",esc,enter)  ; // Plot thick and
                                                     straight line
}
```

COMMAND '17' (Plot Polygon)

It is command to draw polygon (several corners) from 3 corners to 9 corners; moreover, there are 5 levels from '0' to '4' to configure the thick of line in the Thick blank. Remember, if using this command to draw picture, be not able to fill color. The method to configure the position (x,y) of the picture is to configure amount of point corresponding with amount of side of picture; for example, if drawing hexagon, we must configure 6 points. Program will start plotting from the first point to the second point, from the second point to the third point… respectively depend on the amount of side. We must configure parameter value into this command as follows;

1. Configure amount of point or side of picture in NumPoint blank (1 Byte: '3-9') corresponding with amount of point

in Position (x,y) blank. For example, if it is triangle, must be put number '3' and in the Position (x,y) blank must be (x1,y1,x2,y2,x3,y3).

2. We must configure position value in Position (x,y) blank by setting amount side of the preferred picture in this blank but it is not over than 9 sides.
3. The value of thick line in Thick (1 Byte) blank can be configured in the range of '0'-'4'.

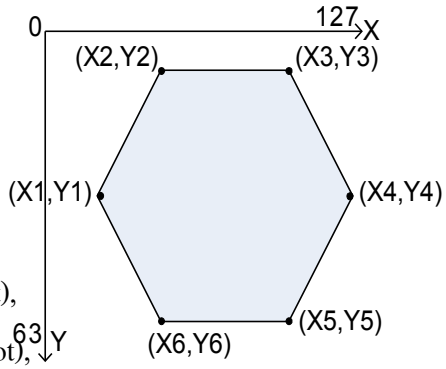
If Invert command (command '01') is set and then follow by this command, characteristic of the drawing line will be changed to white color.

Start	ID	Mk1	Command	Mk2	NumPoint	Mk3	Position (x,y)	Mk6	Thick	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte	X1,Y1, … ,X9,Y9	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	17	=	3-9	,	000-127,00-63, ... ,000-127,00-63	,	0-4	Enter (0x0D)
Respond Command from Board										
*	00-FF	#	17	=	OK					

NumPoint blank = Amount of point or side of picture from '3' to '9'

Position (x,y) blank = Position value x: 3 Byte ('000-127'), y: 2 Byte ('00-63')

Thick blank = Thick of line from '0' – '4'



Ex12. Set ID = '07' Plot hexagon that has position as follows;

(X1 = the 20th dot, Y1 = the 30th dot), (X2 = the 40th dot, Y2 = the 10th dot),

(X3 = the 87th dot, Y3 = the 10th dot), (X4 = the 107th dot, Y4 = the 30th dot),

(X5 = the 87th dot, Y5 = the 50th dot), (X6 = the 40th dot, Y6 = the 50th dot),

Configure the level 3 for the thick of line

```
main()
  {
    char  esc = 0x1B , enter = 0x0D    ;

    printf("%c07#17=6,020,30,040,10,087,10,107,30,087,50,040,50,3%c",esc,enter)
;
  }
```

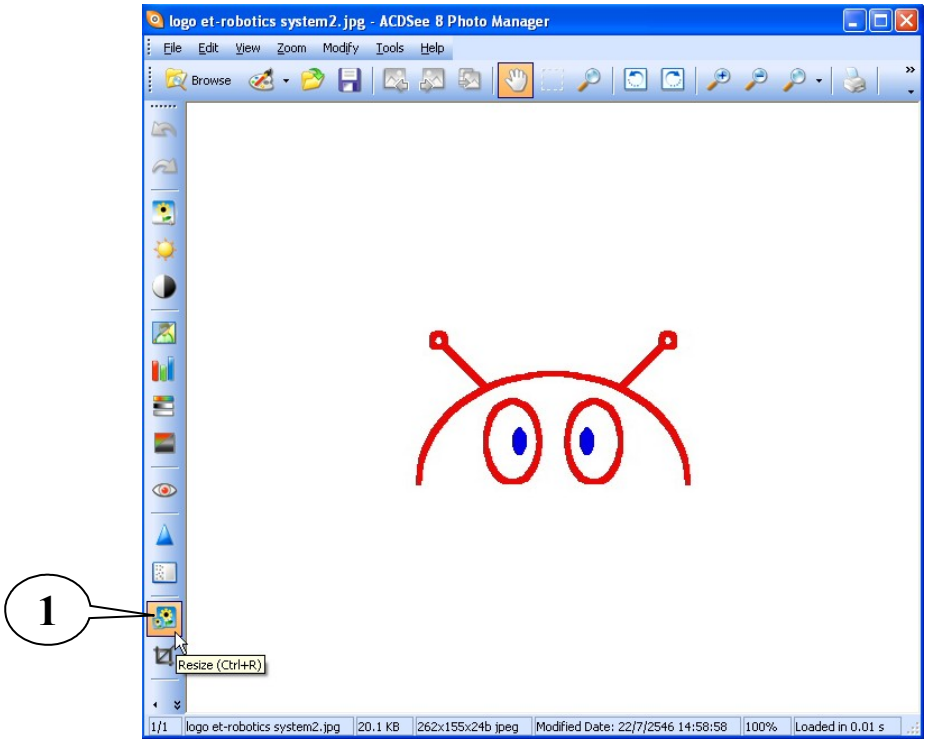
NOTE: *If we want to configure position of picture or line, we must consider that it is in the area of monitor throughout the plotting picture or not, otherwise it makes the drawing incorrect or program hang-up. For example, if drawing circle with 10 Dot radius, Centre position (x,y) of circle must be configured in the range of plotting and all sides of circle is still in the monitor.*

4.3 Command Set regarding Picture Mode

It is the command group that displays external picture on Board GLCD including picture of Back Ground that is provided with board. The method to send picture to display on GLCD monitor is to convert picture to be Code first, save the Code into the memory of board that is used to send Command, send Command and Code of picture to Board GLCD and finally picture will be displayed on GLCD monitor. Normally, the Code that is converted from picture is 1024 Byte and it is fit for GLCD monitor perfectly.

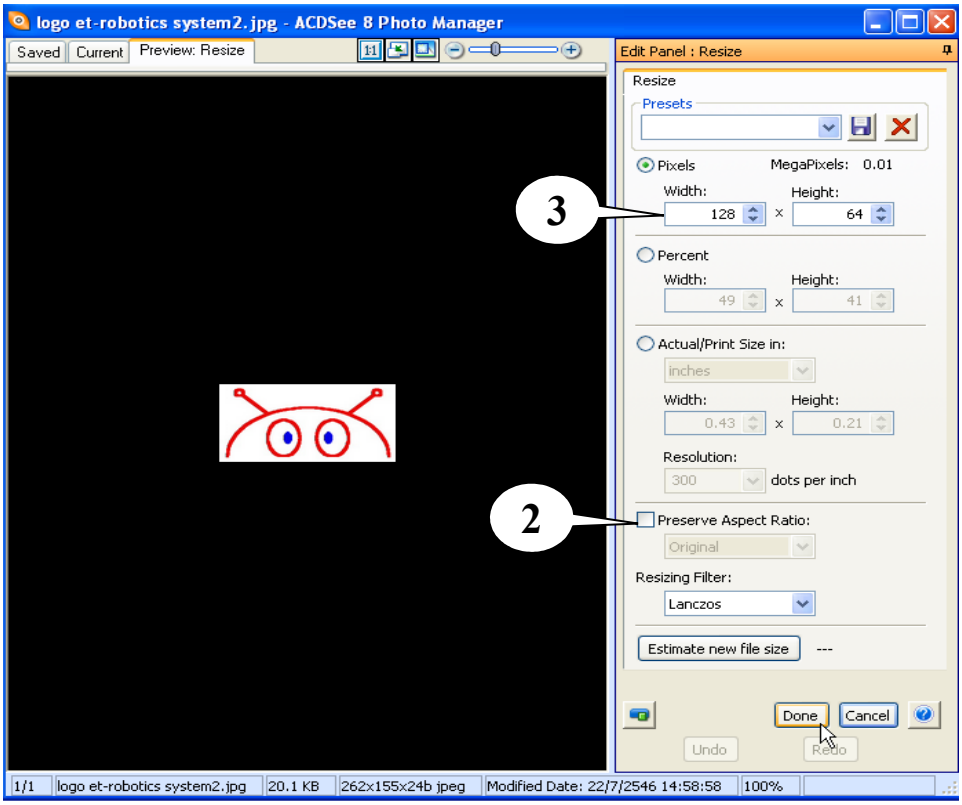
How to convert picture to Hex Code

1) Reduce picture size to be 128x64 Pixel (width x height), open picture with Program ACDSee, and select Icon **Resize** (No.1) as shown in the picture 4.3.1.



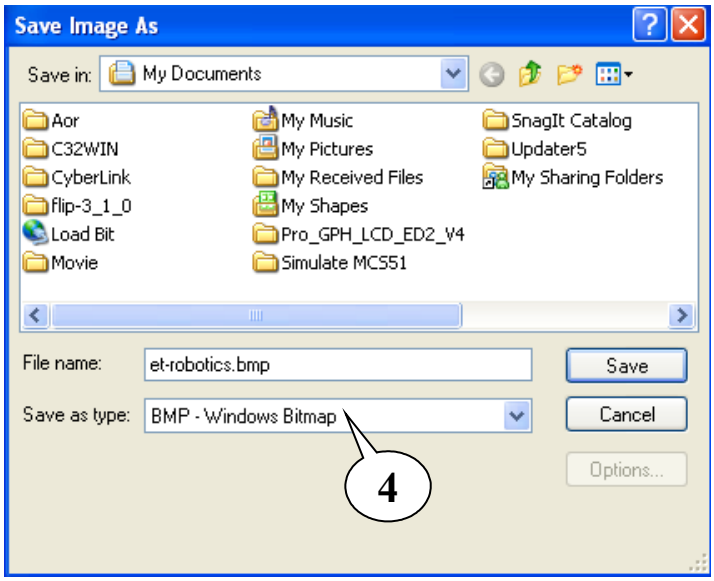
Picture 4.3.1

2) We will get window as shown in the picture 4.3.2; remove tick sign in the in the blank of **Preserve Aspect Ratio:** (No.2); select **Pixels** and configure the picture size in the blank of **width: 128 x height: 64** (No.3) and finally, click **Done.**



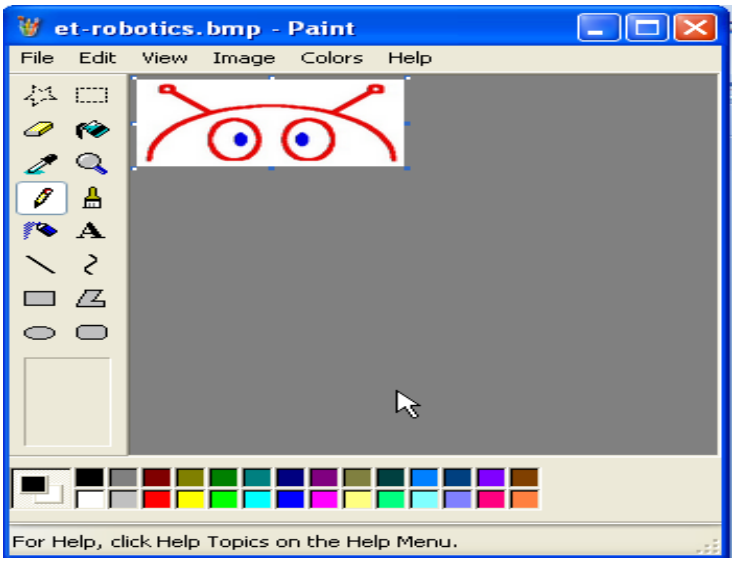
Picture 4.2.3

3) After click **Done;** it will come back to the window as same as in the picture 4.3.1. Open Menu **File,** select **Save as..** and specify filename and only save its file surname as **.BMP** as shown in the picture 4.3.3. After specified filename successfully, click **Save** and finally close Program **ACDSee.**



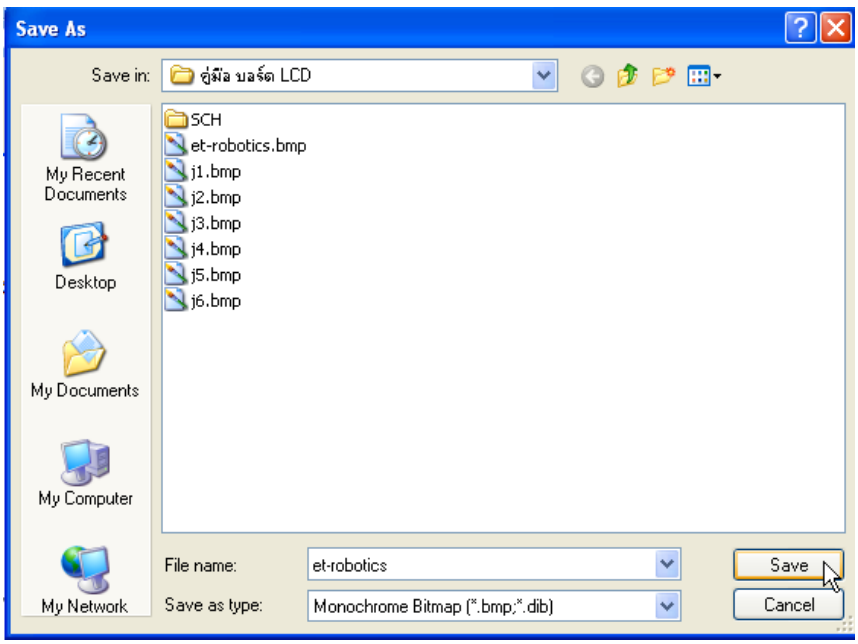
Picture 4.3.3

4) Open Program **Paint** ( ) of Windows and open Menu **File,** select **Open**… to open picture file that is saved in step 3 and we will get window as shown in the picture 4.3.4.



Picture 4.3.4

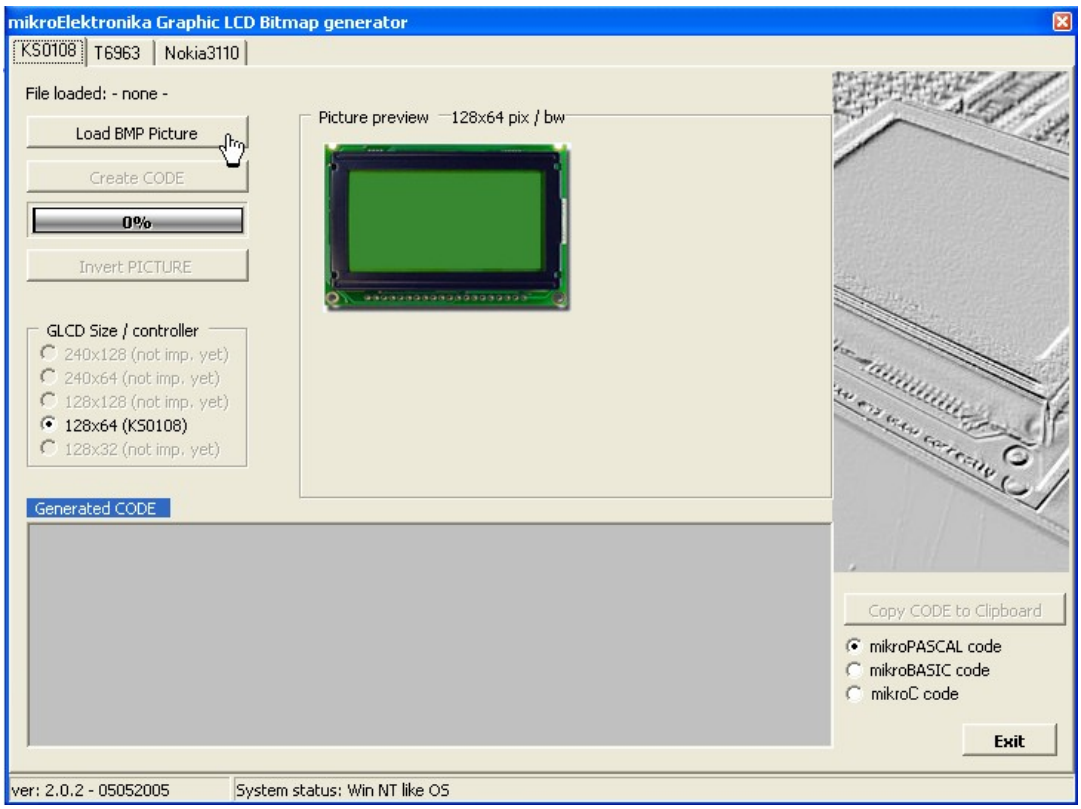
5) Go to Menu **File,** select **Save as**… in the Filename blank to specify filename in the blank of **Save as type:**, select **Monochrome Bitmap (*.bmp;*.db)** as shown in the picture 4.3.5; and then click **Save.** It will display window as same as picture 4.3.4 again but it will be changed to be black and white picture, and then close Program **Paint.**



Picture 4.3.5

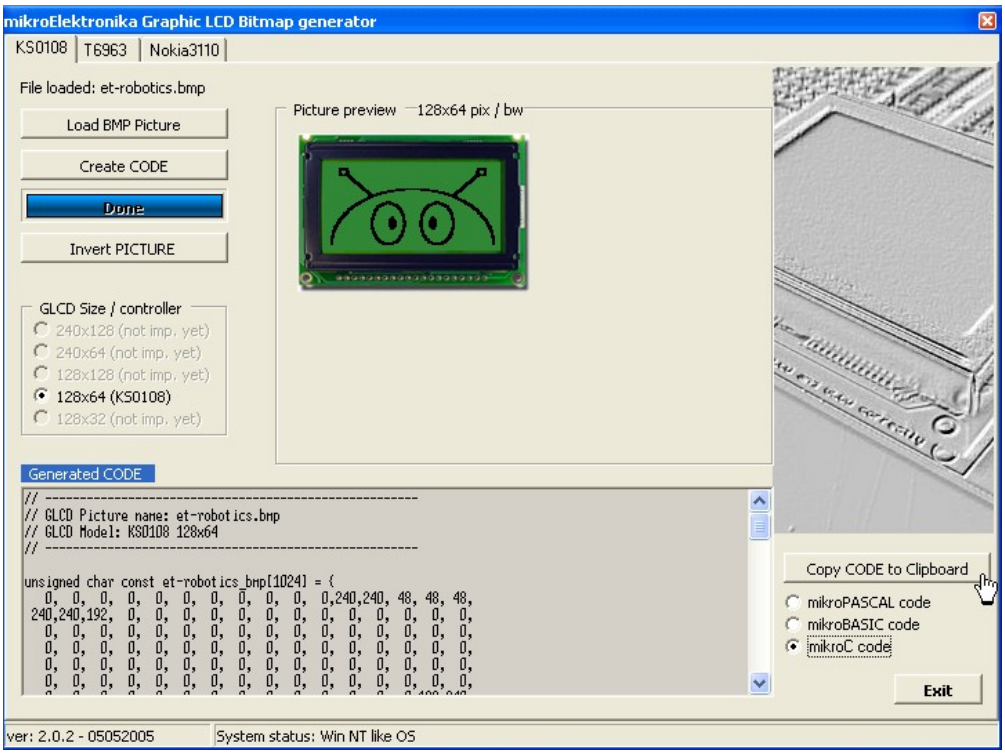
6). Now, we get picture file that can be converted into Hex code; next, open Program **glcd_editor.exe** that is provided with CD-

ROM (pleases copy into computer before opening) and we will get the window as shown in the picture 4.3.6. Next, select Tab **KS0108,** click Button **Load BMP Picture** to load picture file that is saved by Program Paint in the step 5 to this program.



Picture 4.3.6

7). When we have already loaded file into GLCD monitor of this program successfully, it will display example of the loaded picture and in the blank of **Generated Code** will display Code as shown in the picture 4.3.7. In this case, we can click Button **Invert PICTURE** to invert the desired picture and Code will be changed when Button **Invert PICTURE** is clicked.



Picture 4.3.7

8) From the picture 4.3.7 above, after we have already gotten Code, click Button **Copy CODE to Clipboard**. In this case, we can select the desired Code type to copy but it depends on what language that is written program into board to send Command. After Code is copied completely, paste it on the window of program that is used to write program; moreover, this partial Code of variable declaration maybe modified to adjust it according to the format of the written program. Then send this Code to Board GLCD follow the format of command for sending picture.

COMMAND '20' (Write Picture)

It is command to display picture file through LCD monitor but this picture file must be converted into Code first as the steps mentioned above. We must configure parameter value into this command as follows;

1) Data that is Code of 1024 Byte picture file must be configured in the blank of Data File BMP.

Start	ID	Mark1	Command	Mark2	Data File BMP	END
1 byte	2 byte	1 byte	2 byte	1 byte	1024 byte	1 byte
Esc (0x1B)	00-FF	#	20	=	Code	Enter (0x0D)
Respond Command from Board						
*	00-FF	#	20	=	OK	

Code blank = Data of 1024 byte picture file that is converted.

Ex13. Set ID = '08' To display picture through GLCD monitor

```
#include <stdio.h>
#include <ez8.h>

            // ------------------------------------------------------
           // GLCD Picture name: et-robotics.bmp such as Code File
           // GLCD Model: KS0108 128x64
          // ------------------------------------------------------

     rom char robotics_bmp[1024]= {                          //Save data of 1024
byte picture in Flash
         0,0,0,0,0,0,0,0,0,0,0,240,240,48,48,48,
          240,240,192,0,0,0,0,0,0,0,0,0,0,0,0,0,
                      . . .
                      . . .
                      . . .
                0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
               0,0,0,0,0,0,0,0,0,0,15,15,0,0,0,0

         };


      //------ Transmit data ---------
        void Tx(unsigned char data)      //Function Sent data for Z8Encore
         {
           while(!(U0STAT0 & 0x04)){;}
           U0D = data;
         }

        main()
        {
          int k;
          char  esc = 0x1B , enter = 0x0D    ;
          printf("%c08#20=",esc)             ; //Send command '20'

          for(k=0;k<1024;k++)
            Tx(robotics_bmp[k])              ; //Send data 1024 byte

          printf("%c",enter)                 ; //End byte of Command
        }
```

COMMAND '21' (Demo Test GLCD)

It is command to test operation of GLCD monitor and the format of this command is shown below;

Start	ID	Mark1	Command	Mark2	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	21	=	Enter (0x0D)
Respond Command from Board					
*	00-FF	#	21	=	OK

Ex14. Set ID = '09' Demo Test

```
main()
{
  char  esc = 0x1B , enter = 0x0D    ;
  printf("%c09#21=%c",esc,enter)     ; // Demo Test
}
```

COMMAND '22' (Display Wall Paper)

It is command to display Wallpaper. There are 7 pictures that are provided with Board GLCD and we can call these pictures to display through GLCD monitor or it maybe used to be screen saver. We must configure Parameter value in Picture blank in the range of '1-7' to select the preferable picture.

Start	ID	Mark1	Command	Mark2	Picture	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	22	=	'1-7'	Enter (0x0D)
Respond Command from Board						
*	00-FF	#	22	=	**OK**	

Picture blank = Select picture to display through GLCD monitor (from '1' – '7')

Ex15. Set ID = '0A' To display Wall Paper No.4

```
main()
{
  char  esc = 0x1B , enter = 0x0D     ;
  printf("%c0A#22=4%c",esc,enter)     ; // Display Wallpaper
}
```

4.4 Command Set regarding Text Mode

It is command to plot message to display through GLCD monitor and the format of command is shown below.

COMMAND '30' (Set Cursor Position)

This command is used to configure position of Cursor or the beginning position to write message or letter. Normally, if booting computer up, its beginning position will be set at position (0,0) and after we have already written message or letter, position of Cursor will be moved automatically.

Understanding of setting position to write message or alphabet on GLCD monitor

First of all, we must understand that alphabet size that is used in this board is 7 dot heights but the width of each alphabet is different. The height of one alphabet of both Thai and English alphabet is 16 Dot and area of each alphabet is divided into the area of upper vowel-lower vowel and alphabet. From the picture 4.4.1, if we set the beginning position to write message at position x,y = (0,0), we will start counting it down until it reaches 16 dot and it is the height of upper vowel + alphabet + lower vowel. So, the height of the real alphabet is in the position of the 6th – the 12th dot. If it is English message or English alphabet that has not any upper and lower vowel, so these areas are free and we can configure the second line closer to the alphabet in the first line. In this case, it makes us can write message more than 4 lines in only one monitor. We must configure parameter value into this command as follows; the beginning position of message in X blank (3 Byte: '000-127') and Y blank (2 Byte: '00-63').

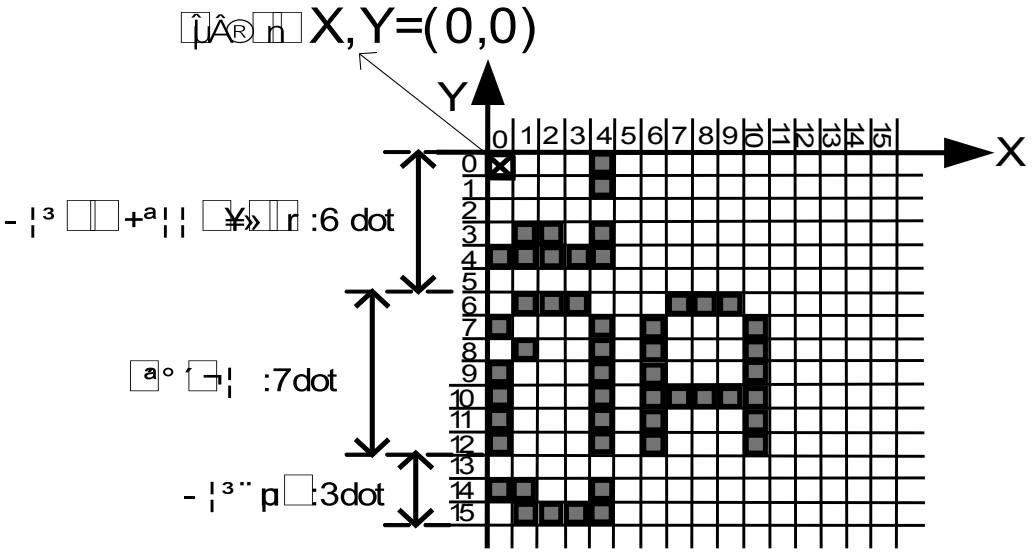
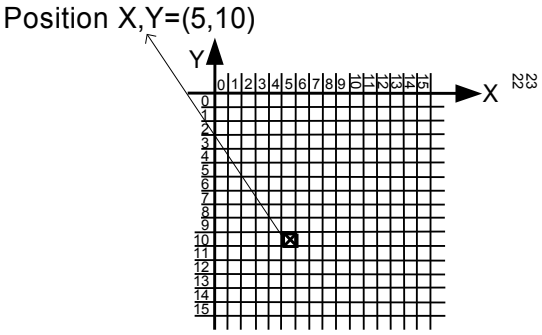


Figure 4.4.1 displays the method to divide area of alphabet on GLCD monitor.

Start	ID	Mark1	Command	Mark2	X	Mark3	Y	END
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2 byte	1 byte
Esc (0x1B)	00-FF	#	30	=	000-127	,	00-63	Enter (0x0D)
Respond Command from Board								
*	00-FF	#	30	=	OK			

Ex16. Set ID = '08' To configure the beginning position to write message at the position X = the 5th dot, Y = the 10th dot




```
main()
{
  char  esc = 0x1B , enter = 0x0D     ;

 printf("%c0B#30=005,10%c",esc,enter) ; // Set x,y
}
```

COMMAND '31' (Write Message1)

It is command to write message or alphabet to display on GLCD monitor. This command will update only message or data that is displayed through GLCD monitor only. There is mode to select color of alphabet to be black or white without mention Invert command (command '01'). Although screen of GLCD is black or white, we can configure color of alphabet to be black or white color as desired but if we configure the color of alphabet the same as the color of screen, it make us can not see any alphabet. When we write message full of the line, the exceeding message will be appeared in the new line automatically but if message is full of the monitor, the exceeding message in the last line of the monitor will be abandoned. We must configure parameter value into this command as follows;

1. Configure color of alphabet in Mode (1 Byte) blank; '0'= White alphabet and '1' = Black alphabet.
2. Configure message to display through Monitor in Data Text blank and amount of alphabet is not over than 200 bytes (vowel + pitch (tone mark in Thai writing) + alphabet).

Start	ID	Mark1	Command	Mark2	Mode	Mark3	Data Text	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte	1-200 byte	1 byte
Esc (0x1B)	00-FF	#	31	=	0-1	:	Message	Enter (0x0D)
Respond Command from Board								
*	00-FF	#	31	=	OK			

Mode blank = '0': White alphabet, '1': Black alphabet

Data Text blank = Data ASCII Thai-Eng: 1-200 byte

Ex17. Set ID = '0C' To begin message at the position X,Y = (10,20), write message as "สวัสดีครับ" to display through monitor with black alphabet.

```
main()
{
  char  esc = 0x1B , enter = 0x0D     ;

      printf("%c0C#30=010,20%c",esc,enter) ; // Set x,y(10,20)

 printf("%c0C#31=1:สวัสดีครับ",esc,enter) ; // Write Message "สวัสดีครับ"
}
```

This example does not check Respond Command; normally, we must check its respond to protect the application from error. In the proceeding of the second command if program does not support Thai Font; when we type Thai message, it displays strange alphabet that can not be read but there is no any problem.

COMMAND '32' (Write Message2)

It is command to write message or alphabet to display on GLCD monitor as same as command '31' but this command will update all data on monitor when sending a command in each time. In this case, we can select 4 types to justify message in Mode blank. The method to configure color of alphabet in this command depends on Invert command (command '01'); if Invert = '1', alphabet will be written by white color (black screen); on the other hand, if Invert = '0', alphabet will be written by black color (white screen). We must configure parameter value into this command as follows;

1). Select format of message alignment in Mode (1 Byte) blank in the range of '0'-'3' when

'0' = Be able to display message until it is full of monitor for sending a command in each time. When it reaches the end of the line, the exceeding message will be written in

the new line automatically, but if it is the last line of monitor, the exceeding message will be abandoned.

'1' = Be able to display message only one line for sending a command in each time and its message is left justification. The exceeding message will be abandoned.

'2' = Be able to display message only one line for sending a command in each time and its message is centre justification. The exceeding message will be abandoned.

'3' = Be able to display message only one line for sending a command in each time and its message is right justification. The exceeding message will be abandoned.

2) Configure message to display through monitor in Data Text blank, amount of alphabet is not over than 200 bytes (vowel + pitch (tone mark in Thai writing) + alphabet).

Start	ID	Mark1	Command	Mark2	Mode	Mark3	Data Text	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte	1-200 byte	1 byte
Esc (0x1B)	00-FF	#	32	=	0-3	:	Message	Enter (0x0D)
Respond Command from Board								
*	00-FF	#	32	=	OK			

Ex18. Set ID = '0D' To begin message at position X,Y = (0,0), write message as "บริษัทอีทีที" to display through monitor and configure this message as centre justification

```
main()
{
  char  esc = 0x1B , enter = 0x0D     ;

      printf("%c0D#30=000,00%c",esc,enter) ; // Set x,y(0,0)

 printf("%c0D#32=2:บริษัทอีทีที",esc,enter) ; // Write Message as "บริษัทอีทีที"
}
```

This example does not check Respond Command; normally, we must check its respond to protect the application from error. In the proceeding of the second command if program does not support Thai Font; when we type Thai message, it displays strange alphabet that can not be read but there is no any problem.

Restrictions regarding writing command with command '30' and '31'

- Before using this command, we must configure the beginning position for writing with command '30'
- To send a message in each time for a command, it can send message not over than 200 byte (vowel + alphabet + pitch (tone mark in Thai writing)).

- When the message that is received is displayed longer than one monitor, the exceeding message will be abandoned.
- Alphabets can be typed from Key Board instantly or if sending it as Code, please see ASCII table in the Appendix.
- The principle of sending Thai alphabets into Board GLCD is the same as the general typing Thai word such as "น้ำ" = น - ้ - ำ and "สู้" = ส - ู - ้
- Use Space Bar or 0x20 to leave a space; in this case, 1 Space Bar = 5 dot space. If leaving space less than 5 dots, we can use 0x01 to replace the alphabet. Value 0x01 is sent out in each time will replace 1 dot space.
- If writing data in each time in the same position and we do not delete the previous data, the new data will overlay the old data and it makes us can not read the data.
- When each alphabet is sent out in each time, program will leave a space between alphabets about 1 dot automatically.

COMMAND '33' (Del Text 7x5 dot)

It is a command to delete the alphabet size (height x width) that is 7 x 5 dot only; for example, number from 1 to 9. It will delete alphabet at the Cursor position appear or delete alphabet in front of Cursor position (can configure Cursor position by command '30'). If we want to delete the different size of alphabet or message such as Thai alphabets or Thai vowel, we must use command '14' instead. In this case, we must configure value in Fill blank to be '1' or '2'; so, it makes us be able to delete alphabet in any position on monitor. We must configure parameter value for this command in Mode blank if it is;

'0' = To delete black alphabet at the Cursor position appears,
'1' = To delete white alphabet at the Cursor position appears,
'2' = To delete black alphabet in front of Cursor position,
'3' = To delete white alphabet in front of Cursor position

Start	ID	Mark1	Command	Mark2	Mode	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	33	=	0-3	Enter (0x0D)
Respond Command from Board						
*	00-FF	#	33	=	OK	

Mode blank = '0': To delete black alphabet at the Cursor position appears,

'1': To delete white alphabet at the Cursor position appears,

'2': To delete black alphabet in front of Cursor position,

'3': To delete white alphabet in front of Cursor position

Ex19. Set ID = '0E' To delete the black alphabet at the position X,Y = (0,0)

```
main()
{
  char  esc = 0x1B , enter = 0x0D     ;

      printf("%c0E#30=000,00%c",esc,enter) ; // Set position to delete alphabet
                                             at x,y(0,0)
 delay(200)                                ;   //delay command
 printf("%c0E#33=0",esc,enter)             ; // Delete alphabet
}
```

The method to configure position to delete alphabet is to use command '30' as described above. After command '33' have already done, it will delete alphabet in the specified area that is a part of alphabet only but the upper vowel and lower vowel still is appeared.

COMMAND '34' (ON/OFF Cursor)

It is a command to ON/OFF Cursor when using command ON Cursor; we will see blinking cursor in the position next to the last written alphabet. When we write the alphabet completely, Cursor will be shifted to the right side automatically. Moreover, we can configure Cursor position to display position of the next alphabet to write by using command '30' because it is clear to see the position of the next alphabet that will be written on the monitor easier. We must configure parameter value in OFF/ON blank for this command; if it is '0' = Cursor OFF, '1' = Cursor ON.

Start	ID	Mark1	Command	Mark2	OFF/ON	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	34	=	0-1	Enter (0x0D)
Respond Command from Board						
*	00-FF	#	34	=	OK	

OFF/ON blank = '0' : Cursor OFF

'1' : Cursor ON

Ex20. Set ID = '0F' To configure Cursor ON at the position X,Y = (30,30)

```
main()
{
  char  esc = 0x1B , enter = 0x0D     ;
```

```
        printf("%c0F#30=030,30%c",esc,enter) ; // Set position to display Cursor:
                                                x,y(30,30)

        printf("%c0F#34=1",esc,enter)        ; // Set Cursor ON
      }
```

***Remember**: The method to count position on GLCD monitor to replace it into the command that must configure parameter value to be position is described as followed; on the X axis, it will count the dot from 0 to 127 and on the Y axis, it will count the dot from 0 to 63 that refers to the table of GLCD monitor in the Appendix. After command has already sent to GLCD Board, we must always check Respond of command before start sending the next command. In this case, we maybe only check "OK" in the last 2 Byte; otherwise if we send command while Board GLCD has not send "OK" yet, it makes nothing happen.*

5. Application of RS232/422 and 4-Line 485

5.1 Sending Command through RS232

The method to send command by RS232 Communication is suitable for controlling only one Board GLCD and we must put IC Line Driver Max232 into the specified Socket on Board GLCD, set both Jumper NO.10 on 232 side and then connect communication cable as shown in the picture below.

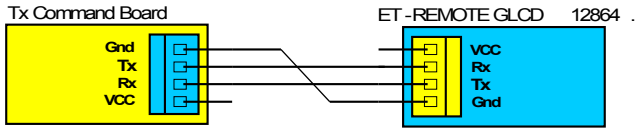


Figure 5.1 displays the method to connect RS232 cable between sender and receiver.

5.2 Sending Command through RS422

The method to send command by RS422 Communication is suitable for controlling only one Board GLCD as same as RS232 Communication but it can communication in longer distance. We must put 2 IC Line Driver 75176 into the specified Socket on Board GLCD, set both Jumper No.10 on 485 side and then connect the communication cable as shown in the picture below.

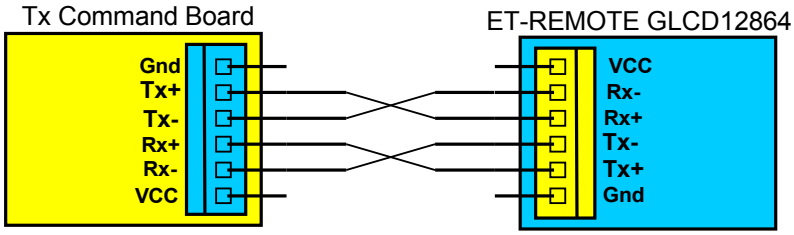


Figure 5.2 displays the method to connect RS232 Cable between sender and receiver.

5.3 Sending Command through 4-Line RS485

The method to send command by 4-Line RS485 Communication is to connect RS422 Cable as shown in the picture below. This communication type is suitable for controlling many Board GLCDs simultaneously. Each Board GLCD can be set ID value from '00'-'FF'; in this case, user can see more information from the table of Set Jumper in the beginning of manual.

We must put 2 IC Line Driver 75176 into the specified Socket on Board GLCD, set Jumper No.10 on 485 side and then connect communication cable as shown in the picture below.

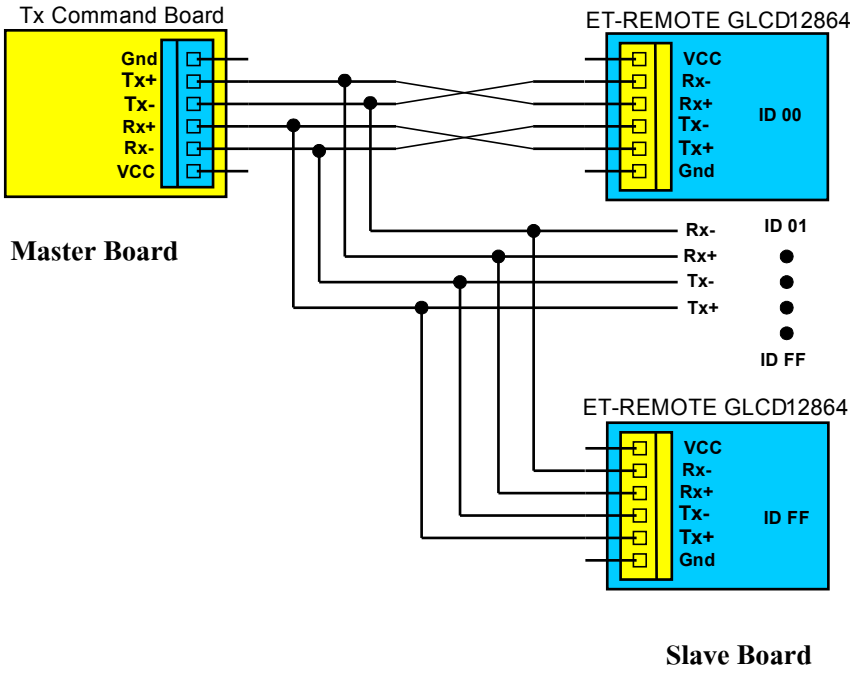


Figure 5.3 displays the method to connect 4-Line RS485 Cable between Master and Slave.

For this RS485 Communication, Board CLCD will receive Command when the Command has ID corresponding with ID of board that is set only. When board has already done Command completely, it will send Respond Command; on the other hand, other board that has different ID, it does not send anything. Be carefully, if Respond Command is sent from more 2 Slave Boards simultaneously, it makes Master Board receive Respond Command from Slave Board incorrectly.

6. Setup Board ET-REMOTE GLCD12864

1) Set Jumper No.10 on 232 side or 485 side depend on the communication type.
2) Set Jumper No.3 to configure Connector Back Light of GLCD correctly.
3) Configure Baud Rate to communicate for Board GLCD, set DIP-SW.B No.1, 2 and 3; in this case, user can see the method to set this value from the table 2.3.
4) Configure ID value for Board GLCD, set DIP-SW.A No.1-8; in this case, user can see the method to set this value from the table 2.1.
5) Connect DISPLAY GLCD with Board Control in the correct position.
6) Adjust VR on board to balance the contrast of GLCD monitor.
7) Connect cable at Connector RS232 or RS485 from Board GLCD to Connector of Board that sends Command.
8) Supply AC/DC 7-12 V into Board GLCD and it makes musical sound loud.
9) Check DIP-SW.B No.4; if it is in ON position, it means that it is in Self-Test Mode and board will display Baud Rate and ID value that is set by user and it will check whether it is correct or not. If we want to modify both values, we must reset DIP-SW.A and B and wait for the both values that are set are changed, it means that the new value has already set completely. If DIP-SW.B No.4 is in OFF position, it means that it is in RUN Mode, GLCD monitor will be blank and Board GLCD is ready to receive command from user.

7. Example program

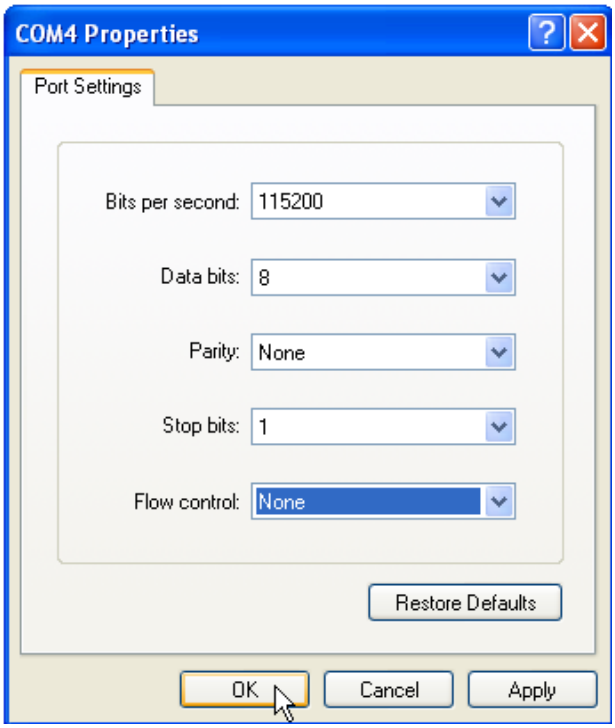
7.1 Example Program to test sending command through Hyper Terminal

1) Set Jumper No.10 on 232 side.

2) Configure Baud Rate to be 115200 bit/s for communication; set DIP-SW.B No.1-3 to ON position.
3) Configure ID value of Board to be '00'; set DIP-SW.A No.1-8 to OFF position.
4) Connect cable for communication at Connector RS232 of board to COM Port of PC.
5) Shift DIP-SW.B No.4 to OFF position to enter RUN Mode and be ready to receive the command.
6) Open Program Hyper Terminal and then set values as shown in the picture below;

(1) Identify name

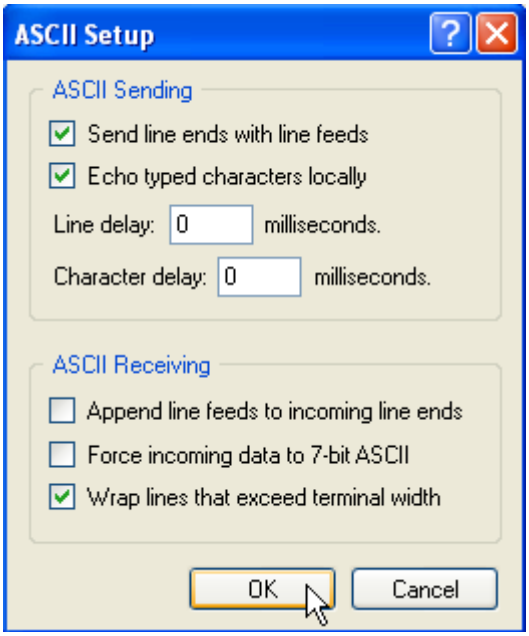
(2) Select COM PORT for communication



(3) Set Configuration of Com Port as shown in the picture above

7) Click Button **Properties** ( ) and it will display window. Select Tab **Settings,** click Button **ASCII Setup**… and it will

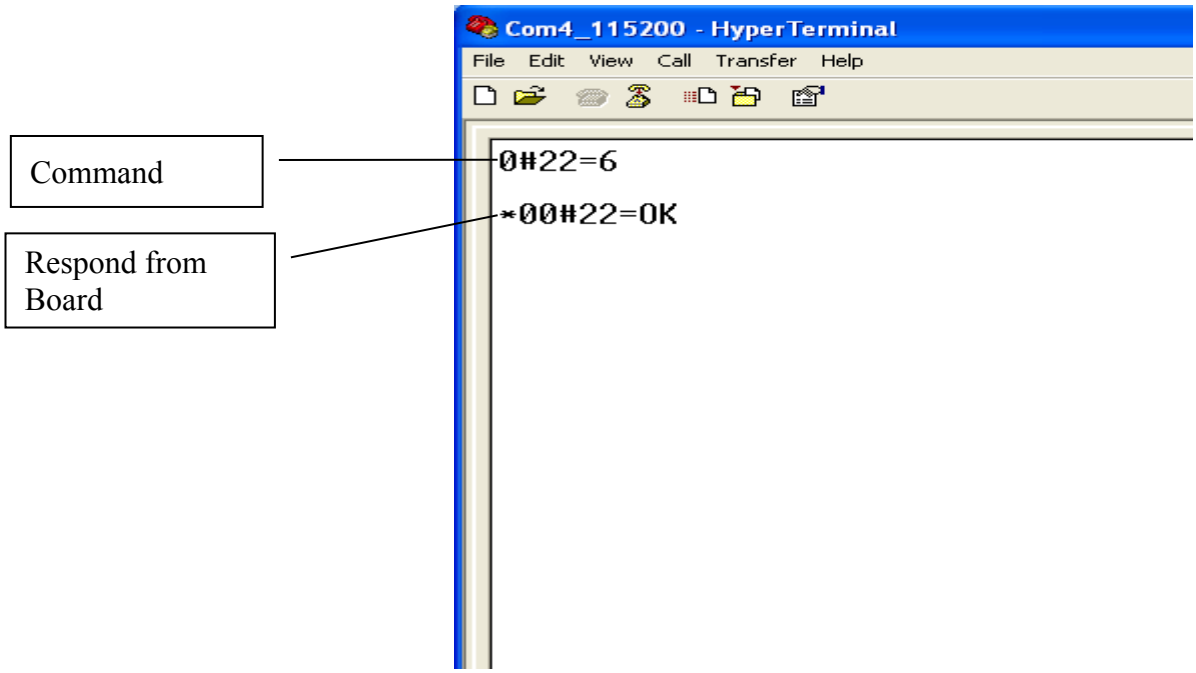
display window as shown below. Select values as shown the picture below and then click **OK**.



8) Try to type command and send it to board; in this case, we will test with command '22' and it will display Logo ETT through LCD monitor. Let's type this command;

Esc 00 # 22 = 6 Enter

Press Key ESC, follow by 00, and #, and 22, and =, and 6 and then Enter without any leaving space. On the monitor of Hyper Terminal will display picture as shown in the picture below.



Command that is sent out does not display number '0' on the monitor of Hyper Terminal because we have already pressed Key Esc. There is no any error because Code has already sent completely. When we have already typed command and pressed Key Enter, the command will be done. After the command has already done completely, Board GLCD will send Respond as shown in the format above.

7.2 Example Program to send command with MCU

Example 1: It is an example of sending command with MCU Z8 Encore (Z8F6422), which is written by C Language on Compile ZDS II, send command through Uart0(RS232:CH0), Baud Rate = 115200 bit/s and configure Board GLCD Set ID = '00'. We must write program for sending command to draw table as in the picture below to display on GLCD monitor.

บริษัท อีทีที จำกัด (2007)

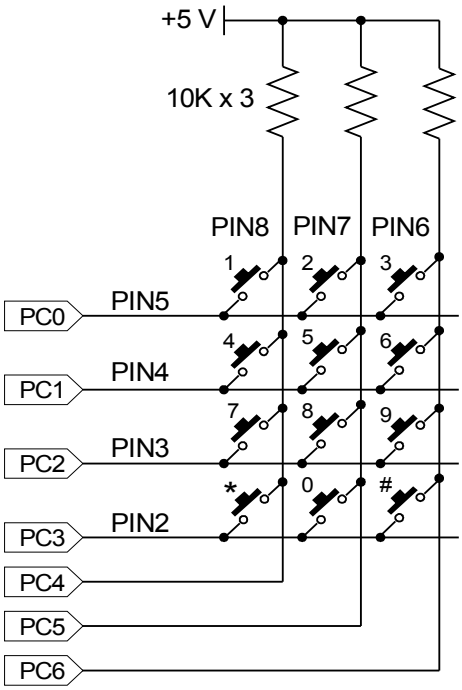
ET-ROBOTICS	ราคา(บาท)
R-MOTOR	210.-
R-TRACKER3	350.-

```
/*******************************************
**                                        **
**   Ex1.Buile Table On Display GLCD       **
**                                        **
*******************************************/

#include <stdio.h>
#include <ez8.h>

  //---------- Receive Data -----------

 unsigned char  Rx()
   {
     while(!(U0STAT0 & 0x80)){;}
     return U0D                ;
   }

 //------ Check Respond Command -------

 void acho()
  {
   unsigned char P,K;

   do{
      P = Rx()         ; //Read Respond 'O'
     }while(P != 'O') ; //ถ้าไม่ใช่ 'O'กลับไปอ่านต่อ

   do{
      K = Rx()         ; //Read Respond 'K'
     }while(K !='K')   ; //ถ้าไม่ใช่ 'K'กลับไปอ่านต่อ

  }


//------------ Main -------------

 void main(void)
 {
  unsigned char esc=0x1B,enter=0x0D,sp=0x01;

  U0BRH  = 0x00  ;
  U0BRL  = 10    ;   //Set Baud Rate 11520
  PAAF   = 0x30  ;   //Set Alternate Pin PA4-5 for Urat 0
  U0CTL0 = 0xC0  ;   //Control Register Uart0 Function
```

```
    //-------- Buile Table On Display GLCD -----------

     printf("%c00#00=%c",esc,enter)       ;  //Cmm Clear Screen
     acho();

     printf("%c00#30=000,00%c",esc,enter);  //Cmm Set Cursor Start Text(0,0)
     acho();

    printf("%c00#32=1:บริษัท  อีทีที  จำกัด (2007)%c",esc,enter);//Cmm write Text close Left
    acho();

    printf("%c00#14=000,16,127,63,0%c",esc,enter)  ;//Cmm Plot Rectang
    acho();

    printf("%c00#30=003,13%c",esc,enter)           ;//Cmm Set Cursor Start Text(3,13)
    acho();

    printf("%c00#31=1:ET-ROBOTICS  ราคา (บาท) %c",esc,enter); //Cmm write Text
    acho();

    printf("%c00#16=000,28,126,28,1%c",esc,enter)  ; //Cmm Plot Line Thick
    acho();

    printf("%c00#15=070,16,070,63%c",esc,enter)    ; //Cmm Plot Line แนวตั้ง
    acho();

    printf("%c00#30=003,28%c",esc,enter)           ;//Cmm Set Cursor Start Text(3,28)
    acho();

    printf("%c00#31=1:R-MOTOR           210.-%c",esc,enter); //Cmm write Text
    acho();

    printf("%c00#15=000,45,126,45%c",esc,enter)    ;//Cmm Plot Line  แนวนอน
    acho();

    printf("%c00#30=003,45%c",esc,enter)           ;//Cmm Set Cursor Start Text(3,45)
    acho();

    printf("%c00#31=1:R-TRACKER3      %c%c350.-%c",esc,sp,sp,enter);//Cmm write Text
    acho();

}
```

From this program, we only check the last 2 alphabets of Respond that are O and K. This program will draw table and write message into that table as shown in the table above. For this variable sp=0x01 is used to send data to replace the desired alphabet and it becomes 1 dot space on LGCD monitor. Moreover, it aligns alphabet position in a straight line.

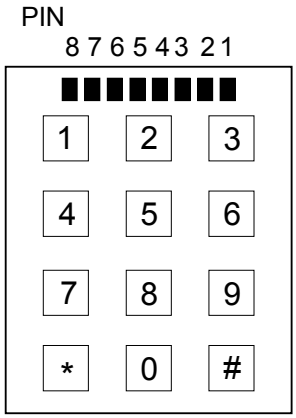
Example 2: It is an example of sending command with MCU Z8 Encore (Z8F6422), which is written by C language on Compile ZDS II, send command through Uart0 (RS232:CH0), Baud Rate = 115200 bit/s and configure Board GLCD set ID = '00'. We must write program to receive value from Key matrix 4x3 through Port C, and then display the value that is keyed through GLCD monitor. When we have already keyed all 10 numbers completely, it makes musical sound loud at Board GLCD. We set Key function as follow; Key * to clear all numbers that are keyed, Key # to delete a number in each time that is in front of Cursor position. User can see Source Code of this example in the provided CD in Example of EX2.CmmGled and see circuit to connect Key with MCU Z8F6422 in the picture 7.2.1.



Circuit

EE	DE	BE
ED	DD	BD
EB	DB	BB
E7	D7	B7

Key Code



Key Pad

Figure 7.2.1 displays the method to connect Key Switch 4x3 for the second example.

Example 3: It is an example of sending command with MCU AT89C51ED2, which is written by C Language on Keil µVISION 3 V3.33, send command through Uart(RS232), Baud Rate = 115200 bit/s and configure Board GLCD Set ID = '00'. We must write program to send command and it draws a circle to display on the monitor; moreover, there is hand inside circle that is rotating as same as clock. When it rotates completely in a round, it will display amount of round on the left top of the GLCD monitor. User can see Source Code of this example program in the provided CD in the example of Ex3.CmmGlcd.

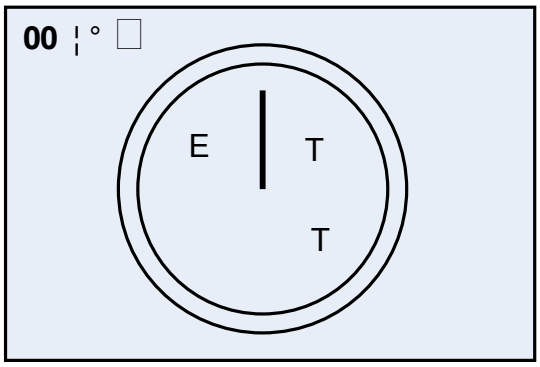


Figure 7.2.2 displays example of GLCD monitor when running the third example program.

Example 4: It is an example of sending command by MCU AT89C51ED2, which is written by C Language on Keil µVISION 3 V3.33, send command through Uart(RS232), Baud Rate = 115200 bit/s and configure Board GLCD Set ID = '00'. We must connect 5 SW. with Port P1.0 - P1.4 of MCU as shown in the picture 7.2.3 below. Next, write program to display switch button through GLCD monitor as shown in the picture 7.2.4 (A); when we press any switch from external, switch button on GLCD monitor will be displayed its status that is also pressed as same as the external switch button and it makes musical notes loud. User can see Source Code of this example from the provided CD in the example of EX4.CmmGLcd.

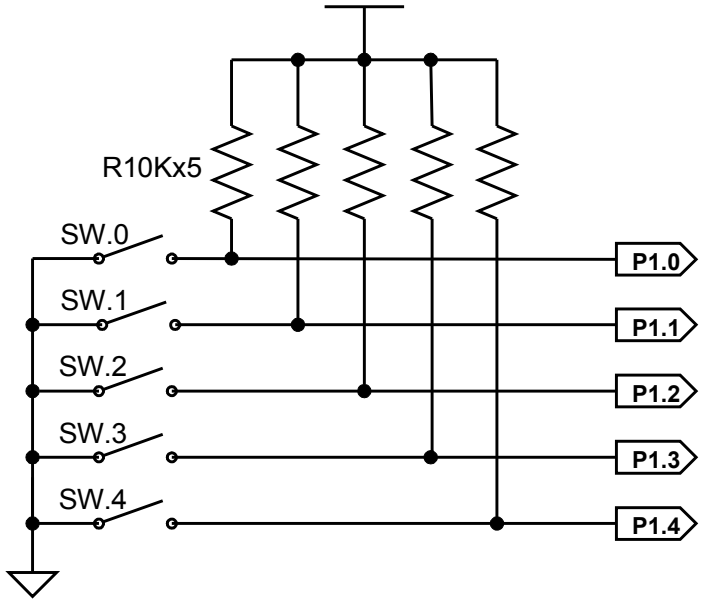
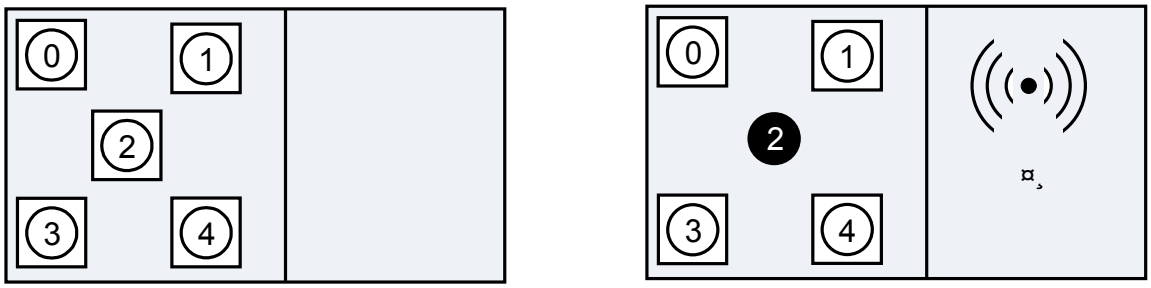


Figure 7.2.3 displays circuit to connect SW with Port P1.0-P1.4.



A) Monitor while not pressing any switch

B) Monitor while pressing SW.2

Figure 7.2.4 display the GLCD monitor when running the forth example program.

Remember: *While supplying power into board for sending command and Board GLCD and before sending command to board GLCD, we must ensure that Board GLCD is in the status of ready to receive command. The method to write program into the board for sending command is to delay value at the beginning of program or check Sync Byte that is sent out by Board GLCD first.*

Appendix A: To summarize the table of command

1) COMMAND '00' (Clear Screen)

Start	ID	Mark1	Command	Mark2	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	00	=	Enter (0x0D)

2) COMMAND '01' (Invert Screen)

Start	ID	Mark1	Command	Mark2	Invert	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	01	=	0-1	Enter (0x0D)

Invert = '0' : Normal (default) , '1' : Invert

3) COMMAND '02' (Display Screen)

Start	ID	Mark1	Command	Mark2	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	02	=	Enter (0x0D)

4) COMMAND '03' (On/Off Back Light)

Start	ID	Mark1	Command	Mark2	Back Light	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	03	=	0-1	Enter (0x0D)

Back Light = '0' : OFF , '1': ON

5) COMMAND '04' (Sound)

Start	ID	Mark1	Command	Mark2	Note	Mark3	Delay	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte	1byte	1 byte
Esc (0x1B)	00-FF	#	04	=	0-7	,	0-9	Enter (0x0D)

Note = '0': No Sound, '1': Do (524 Hz), '2': Re (587 Hz), '3': Mi (659 Hz), '4': Fa (698 Hz), '5': Sol (784 Hz), '6': La (880 Hz), '7': Ti (988 Hz)

Delay = '0': The shortest of sound length and its sound length will be increased follow the number and '9': The longest of sound length

6) COMMAND '10' (Plot Circle)

Start	ID	Mk1	Command	Mk2	Cx	Mk3	Cy	Mk4	Radius	Mk5	Fill	END
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2byte	1 byte	2 byte	1 byte	1 byte	1 byte

Esc (0x1B)	00-FF	#	10	=	000-127	,	00-63	,	00-31	,	0-1	Enter (0x0D)

7) COMMAND '11' (Plot Ellipse)

Start	ID	Mk1	Command	Mk2	Cx	Mk3	Cy	Mk4	Rx	Mk5	Ry	Mk6	Fill	END
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2byte	1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	11	=	000-127	,	00-63	,	00-64	,	00-32	,	0-1	Enter (0x0D)

8) COMMAND '12' (Plot Dot)

Start	ID	Mark1	Command	Mark2	X	Mark3	Y	END
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2byte	1 byte
Esc (0x1B)	00-FF	#	12	=	000-127	,	00-63	Enter (0x0D)

9) COMMAND '13' (Plot Triangle)

Start	ID	Mk1	Command	Mk2	X1	Mk3	Y1	Mk4	X2	Mk5	Y2	Mk6
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2byte	1 byte	3 byte	1 byte	2 byte	1 byte
Esc (0x1B)	00-FF	#	13	=	000-127	,	00-63	,	000-127	,	00-63	,

(Continued)

X3	Mk7	Y3	Mk8	Fill	END
3 byte	1 byte	2byte	1 byte	1 byte	1 byte
000-127	,	00-63	,	0-1	Enter (0x0D)

10) COMMAND '14' (Plot Rectangle)

Start	ID	Mk1	Command	Mk2	X1	Mk3	Y1	Mk4	X2	Mk5	Y2	Mk6	Fill	END
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2byte	1 byte	3 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	14	=	000-127	,	00-63	,	000-127	,	00-63	,	0-2	Enter (0x0D)

Fill: '0' = line, '1' = Whole black picture, '2' = Whole white picture without any line

11) COMMAND '15' (Plot Line)

Start	ID	Mk1	Command	Mk2	X1	Mk3	Y1	Mk4	X2	Mk5	Y2	END
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2byte	1 byte	3 byte	1 byte	2 byte	1 byte
Esc (0x1B)	00-FF	#	15	=	000-127	,	00-63	,	000-127	,	00-63	Enter (0x0D)

12) COMMAND '16' (Plot Line Thick)

Start	ID	Mk1	Command	Mk2	X1	Mk3	Y1	Mk4	X2	Mk5	Y2	Mk6	Thick	END
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2byte	1 byte	3 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	16	=	000-127	,	00-63	,	000-127	,	00-63	,	0-4	Enter (0x0D)

13) COMMAND '17' (Plot Polygon)

Start	ID	Mk1	Command	Mk2	NumPoint	Mk3	Position (x,y)	Mk6	Thick	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte	X1,Y1, … ,X9,Y9	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	17	=	3-9	,	000-127,00-63, ... ,000-127,00-63	,	0-4	Enter (0x0D)

NumPoint = Amount of point or side of picture that can be configured from '3'-'9'.

Position (x,y) = Position X: 3 byte ('000-127'), Y: 2 byte ('00-63')

Thick = Thick of line can be configured from '0'-'4'.

14) COMMAND '20' (Write Picture)

Start	ID	Mark1	Command	Mark2	Data File BMP	END
1 byte	2 byte	1 byte	2 byte	1 byte	1024 byte	1 byte
Esc (0x1B)	00-FF	#	20	=	Hex Code	Enter (0x0D)

Hex Code = Data of 1024 byte Picture file is converted

15) COMMAND '21' (Demo Test GLCD)

Start	ID	Mark1	Command	Mark2	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	21	=	Enter (0x0D)

16) COMMAND '22' (Display Wall paper)

Start	ID	Mark1	Command	Mark2	Picture	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	22	=	'1-7'	Enter (0x0D)

Picture = To select the desired picture to display through GLCD monitor (from '1-7')

17) COMMAND '30' (Set Cursor Position)

Start	ID	Mark1	Command	Mark2	X	Mark3	Y	END
1 byte	2 byte	1 byte	2 byte	1 byte	3 byte	1 byte	2 byte	1 byte
Esc (0x1B)	00-FF	#	30	=	000-127	,	00-63	Enter (0x0D)

18) COMMAND '31' (Write Message1)

Start	ID	Mark1	Command	Mark2	Mode	Mark3	Data Text	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte	1-200 byte	1 byte
Esc (0x1B)	00-FF	#	31	=	0-1	:	ข้อความ	Enter (0x0D)

Mode: '0': To display white alphabet, '1': To display black alphabet

Data Text = Data ASCII Thai-Eng: 1-200 byte

19) COMMAND '32' (Write Message2)

Start	ID	Mark1	Command	Mark2	Mode	Mark3	Data Text	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte	1-200 byte	1 byte
Esc (0x1B)	00-FF	#	32	=	0-3	:	ข้อความ	Enter (0x0D)

Mode = '0': To display alphabet is not over than one monitor in each time and go to the new line automatically,

'1': To display alphabet is not over than one line and it is in the left justification,

'2': To display alphabet is not over than one line and it is in the center justification,

'3': To display alphabet is not over than one line and it is in the right justification.

20) COMMAND '33' (Del Text ขนาด 7x5 dot)

Start	ID	Mark1	Command	Mark2	Mode	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	33	=	0-3	Enter (0x0D)

Mode = '0': To delete the black alphabet in the Cursor position appears,

'1': To delete the white alphabet in the Cursor position appears,

'2': To delete the black alphabet in front of the Cursor position,

'3': To delete the white alphabet in front of the Cursor position.

21) COMMAND '34' (ON/OFF Cursor)

Start	ID	Mark1	Command	Mark2	OFF/ON	END
1 byte	2 byte	1 byte	2 byte	1 byte	1 byte	1 byte
Esc (0x1B)	00-FF	#	34	=	0-1	Enter (0x0D)

OFF/ON = '0' : Cursor OFF

'1' : Cursor ON

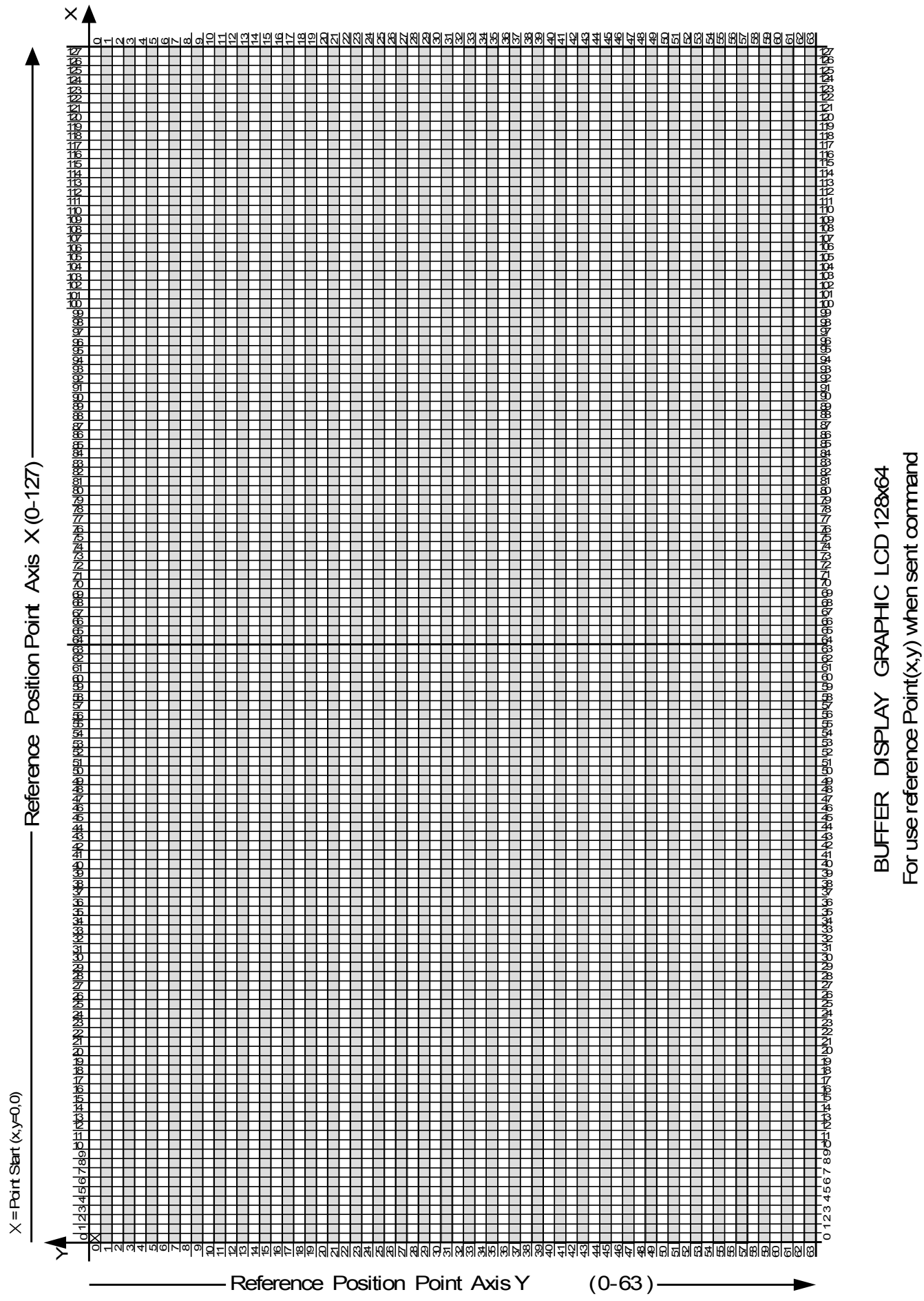
Appendix B: Table of ASCII Code สมอ.

Table Font ASCII Code - ¤° .

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	×	44	2C	,	88	58	X	132	84	×	176	B0	β	220	DC	⌐
1	1	×	45	2D	-	89	59	Y	133	85	…	177	B1	γ	221	DD	▼
2	2	×	46	2E	.	90	5A	Z	134	86	×	178	B2	δ	222	DE	▲
3	3	×	47	2F	/	91	5B	[	135	87	×	179	B3	ε	223	DF	ο
4	4	×	48	30	0	92	5C	\	136	88	×	180	B4	ζ	224	E0	□
5	5	×	49	31	1	93	5D	]	137	89	×	181	B5	η	225	E1	∩
6	6	×	50	32	2	94	5E	^	138	8A	×	182	B6	θ	226	E2	□
7	7	×	51	33	3	95	5F	_	139	8B	×	183	B7	ι	227	E3	Ш
8	8	×	52	34	4	96	60	`	140	8C	×	184	B8	κ	228	E4	ш
9	9	×	53	35	5	97	61	a	141	8D	×	185	B9	λ	229	E5	×
10	0A	×	54	36	6	98	62	b	142	8E	×	186	BA	μ	230	E6	ч
11	0B	×	55	37	7	99	63	c	143	8F	×	187	BB	ν	231	E7	Ъ
12	0C	×	56	38	8	100	64	d	144	90	×	188	BC	ξ	232	E8	Ы
13	0D	×	57	39	9	101	65	e	145	91	'	189	BD	ο	233	E9	Э
14	0E	×	58	3A	:	102	66	f	146	92	'	190	BE	π	234	EA	Ю
15	0F	×	59	3B	;	103	67	g	147	93	"	191	BF	ρ	235	EB	Я
16	10	×	60	3C	<	104	68	h	148	94	"	192	C0	ς	236	EC	∞
17	11	◀	61	3D	=	105	69	i	149	95	•	193	C1	σ	237	ED	∅
18	12	×	62	3E	>	106	6A	j	150	96	-	194	C2	τ	238	EE	∈
19	13	×	63	3F	?	107	6B	k	151	97	—	195	C3	υ	239	EF	∩
20	14	×	64	40	@	108	6C	l	152	98	×	196	C4	φ	240	F0	≡
21	15	×	65	41	A	109	6D	m	153	99	×	197	C5	χ	241	F1	±
22	16	×	66	42	B	110	6E	n	154	9A	×	198	C6	ψ	242	F2	≥
23	17	×	67	43	C	111	6F	o	155	9B	×	199	C7	ω	243	F3	≤
24	18	×	68	44	D	112	70	p	156	9C	×	200	C8	ΐ	244	F4	⌠
25	19	×	69	45	E	113	71	q	157	9D	×	201	C9	ΰ	245	F5	⌡
26	1A	×	70	46	F	114	72	r	158	9E	×	202	CA	ό	246	F6	÷
27	1B	×	71	47	G	115	73	s	159	9F	×	203	CB	ύ	247	F7	≈
28	1C	×	72	48	H	116	74	t	160	A0	×	204	CC	ώ	248	F8	°
29	1D	×	73	49	I	117	75	u	161	A1	Σ	205	CD	□	249	F9	∙
30	1E	×	74	4A	J	118	76	v	162	A2	Τ	206	CE	β	250	FA	·
31	1F	×	75	4B	K	119	77	w	163	A3	Υ	207	CF	γ	251	FB	√
32	20		76	4C	L	120	78	x	164	A4	Φ	208	D0	Υ	252	FC	×
33	21	!	77	4D	M	121	79	y	165	A5	Χ	209	D1	ΰ	253	FD	×
34	22	"	78	4E	N	122	7A	z	166	A6	Ψ	210	D2	ϋ	254	FE	×
35	23	#	79	4F	O	123	7B	{	167	A7	Ω	211	D3	φ	255	FF	×
36	24	$	80	50	P	124	7C	|	168	A8	ΐ	212	D4	β			
37	25	%	81	51	Q	125	7D	}	169	A9	Ύ	213	D5	□			
38	26	&	82	52	R	126	7E	~	170	AA	ά	214	D6	ζ			
39	27	'	83	53	S	127	7F	∎	171	AB	έ	215	D7	η			
40	28	(	84	54	T	128	80	×	172	AC	ή	216	D8	┌			
41	29	)	85	55	U	129	81	×	173	AD	ί	217	D9	┘			
42	2A	*	86	56	V	130	82	×	174	AE	ΰ	218	DA	┌			
43	2B	+	87	57	W	131	83	×	175	AF	α	219	DB	▶			

* × : Àø¤¸□° □¦

Appendix C : DISPLAY GRAPHIC LCD 128x64



Appendix D: Circuit Board

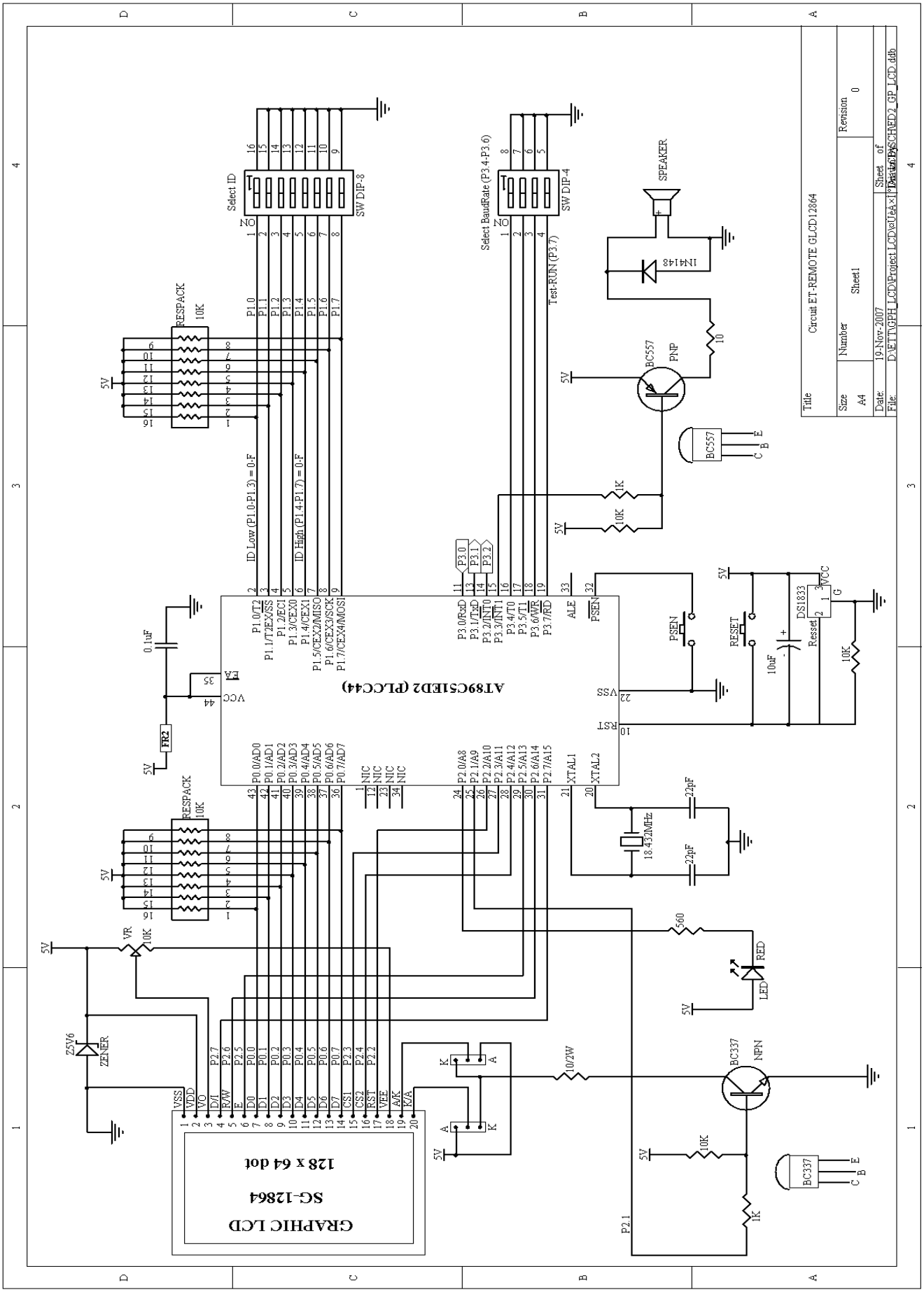


Figure of Circuit ET-REMOTE GLCD12864 Sheet 1

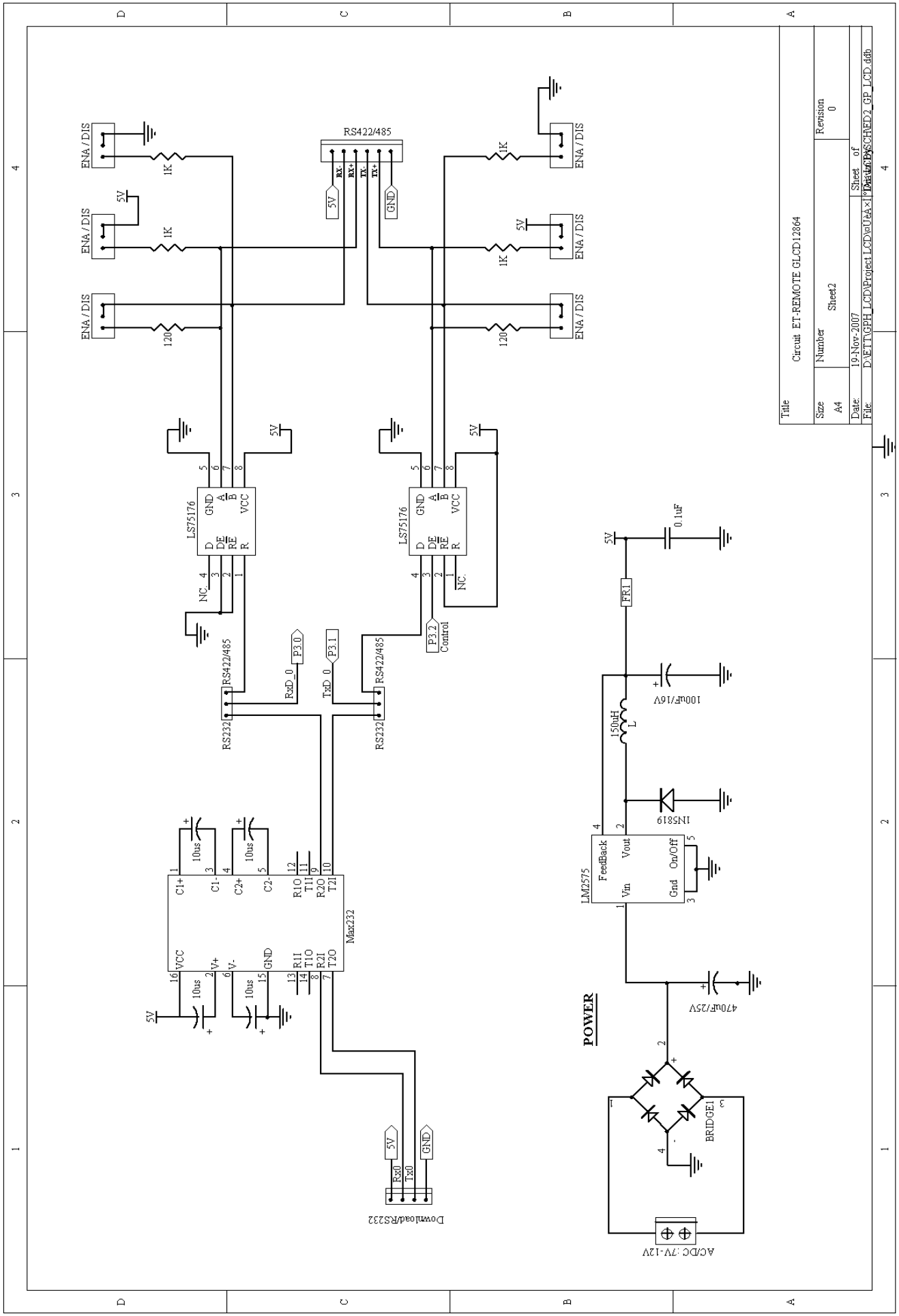


Figure of circuit ET-REMOTE GLCD12864 Sheet 2