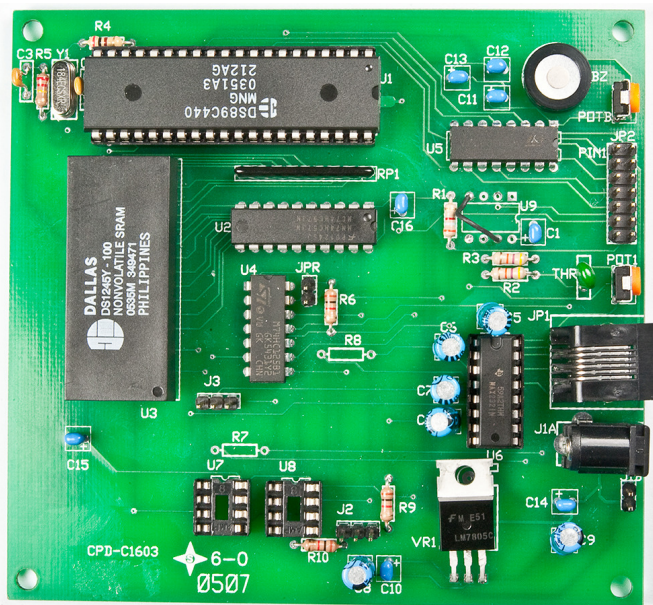


# IS-C1603H Intelligent Controller Users Manual

Revision A

IS-C1603H

40 LCD 64x32 Controller Version 1.2

**NKK SWITCHES**7850 E. Gelding Drive  
Scottsdale, AZ 85260

Toll Free 1-877-2BUYNKK (877-228-9655)

Phone 480-991-0942

Fax 480-998-1435

e-mail &lt;engineering@nkkswitches.com&gt;

**All Rights Reserved Worldwide**

NKK Switches makes no warranty for the use of these products and assumes no responsibility for any errors, which may appear in this document, nor does it make a commitment to update the information contained herein. Smart Switch is trademark of NKK Switches.

## TABLE OF CONTENTS

Table of Contents .....	2
1. General Controller Features .....	3
2. Compatible Logic Boards.....	4
3. Operational Detail.....	5
4. Communication Protocol.....	7
5. Hardware .....	32
6. Key Terms & Definitions.....	42
7. Programming the Firmware.....	43
8. Firmware problems .....	43
Appendix.....	44

## 1. General Controller Features

The IS-C1603H controls up to 40 LCD 64x32 switches/ displays. The IS-C1603H is designed to be used in many different applications. There is total flexibility with user-defined features that allow the controllers to be programmed for specific applications. The user defined data and set-up are stored in a non volatile memory and specify the way the system behaves. The firmware can be customized based on customer requirements. Below are current features:

### **General features:**

- Control up to 40 LCD 64x32 switches/displays.
- Serial communication via RS232 (9.6K, 19.2K, 57.6K default, 1 start bit, 8 bit , 1 stop bit).
- RS422 optional.
- User programmable for images, attributes and set-ups.
- Look up tables for fonts 5x7 and 7x10.
- Additional 160 user defined look up table for each of font 5x7 and font 7x10.
- LED Brightness settings.
- Memory for 255 images and attributes.
- Reports switch activities via serial port.
- User defined activity reports based on switch or picture presses and releases. Up to 15 bytes.
- Can be programmed to mimic an existing keyboard that transmits over RS232 or RS422.
- Stand alone operation or real time control by host.

### **Switch-action report from the controller to the host:**

User selectable for any of the reports below:

- Default switch press event (1 byte in the form of 80H + Switch #).
- Default switch release event (1 byte in the form of C0H + Switch #).
- Default image press event (2 byte indicating location).
- Default image release event (2 byte indicating location).
- User-defined switch press event report (up to 15 bytes).
- User-defined switch release event report (up to 15 bytes).
- User-defined picture press event report (up to 15 bytes).
- User-defined picture release event report (up to 15 bytes).

### **Set-up options:**

- Adjustable audio feedback while switch is pressed.
- User-defined font 5x7 (256 character).
- User-defined font 7x10(256 character).
- User-defined backlighting blinking OFF duration.
- User-defined backlighting on-to-off ratio.

**Real-time operation features:**

- Download a graphic image data to any address.
- Select image display for each LCD module.
- Download one row or one characters and controller will create graphic data using 5x7 font for the specified position in the display at any address.
- Download one row or one characters and controller will create graphic data using 7x10 font for the specific position in the display at any address.
- Control any single pixel on any of 255 images.
- Change backlight color on any LCD module.

## 2. Compatible Logic Boards

Switch Logic Boards: PCB's with mounted switches or displays that are used for this controller are called Logic Boards. The available Logic Boards are listed below. Based on components on the Logic Boards and the length of the interconnect cables, the number of switches that can be connected to the controller may be much less than 40 switches.

All the Logic Boards have two 7x2 connectors and can be daisy chained via 14 pin ribbon cables. JP2 of the IS-C1603H gets connected to J1 of the first Logic Board, J2 of the first Logic Board gets connected to J1 of the second Logic Board and so on. The red wire of the ribbon cable, indicating pin 1, must be connected to pin 1 for all connections.

The following is a list of compatible Logic Boards and switches.

**RGB, Two Modules:**

- A. IS-L02A1-C Two LCD 64x32 RGB switch
- B. IS-L02F1-C Two LCD 64x32 RGB Display
- C. IS-L02H2-C Two LCD 64x32 RGB Compact switch

Note: Maximum of 16 switches can be connected using these Logic Boards. Last Logic board in daisy chain needs terminating 200 Ohm resistor between J2 pin 3 and RJ pin 2.

**RGB, Six Modules: Side by side stackable.**

- A. IS-L06A1-C Six LCD 64x32 RGB, 2x3 (CxR).

Note: Maximum of 40 switches can be connected using this Logic Boards.

**New Logic Boards can be designed based on customer requirements.**

**Note: Make sure the power is off when connecting or disconnecting the Logic Boards to or from the controller or to each other.**

**Note: Improper installation of the Logic Boards could damage the Logic Boards, LCD modules and the controller.**

### 3. Operational Detail

**Power-up Sequence:**

Each switch has a Pointer. There are 255 addresses in the memory (01H to FFH). Each Address holds a Page of data. Each Page has an image data, Attribute block, and user-defined look-up tables.

Upon power-up the data from Page 1 is displayed on switch 1, the data from Page 2 is displayed on switch 2 and so on for maximum of 40 switches.

The image data for modules are sent to module once and the modules handle the refreshing. The LED codes are transferred to an intermediate buffer for refresh. Therefore any command from host that make changes to the storage data will go to effect immediately only if it also transfers the code to intermediate buffer or modules.

After power-up or reset, the controller transmits 11H. This lets the host know that the controller is on line and ready for commands.

The controller then waits for a switch action or a command from the host.

**Main Operational Mode:**

The controller takes the following actions upon a switch press/release.

**1. Switch Release:**

- A. If bit 0 of FLAG-A is enabled (default: enabled), transmit (Switch # + C0H) to host in HEX format.
- B. If bit 2 of FLAG-C is enabled (default: disabled), transmit to host 0 to 15 characters according to user-defined switch release look-up table associated with the switch number.
- C. If bit 0 of FLAG-C is enabled (default: disabled), transmit to host in ASCII HEX format the Pointer value of the image being displayed.
- D. If bit 1 of FLAG-C is enabled (default: disabled), transmit to host up to a maximum of 8 characters according to user-defined image release look-up table associated with the Pointer.

**2. Switch Closure:**

- A. If bit 1 of FLAG-A is enabled (default: enabled), transmit (Switch # + 80H) to host in HEX format.
- B. If bit 2 of FLAG-A is enabled (default: disabled), transmit to host 0 to 15 characters according to user-defined switch press look-up table associated with the switch number.
- C. If bit 3 of FLAG-A is enabled (default: disabled), transmit to host in ASCII HEX format the Pointer value of the image being displayed.
- D. If the bit 4 of FLAG-A is enabled (default: disabled), transmit to host up to a maximum of 15 characters according to user-defined image press look-up table associated with the Pointer.
- E. If bit 5 of FLAG-A is enabled (default: enabled) then do the following:

1. Update the Pointers with the Addresses from the Attribute Block in the Page of the displaying Image. If any of the Addresses in the attribute block is equal to zero do not change that Pointer.
2. If bit 0 of FLAG-B is enabled (default: enabled), update the LCD/LED codes for the switches according to updated Pointers.

## LED Backlighting for 64x32 Resolution Switches:

The LED code has two bytes, one byte for “on cycle” and one byte for “off cycle”. The format for both bytes is the same. Following is the byte format:

B7	B6	B5	B4	B3	B2	B1	B0
R	R	G	G	B	B	1	1

For each of red, green and blue color:

**00=off**  
**01= ¼ brightness**  
**10 = ½ brightness**  
**11= full brightness**

## Attribute Block for 64x32 Resolution Switches:

The number of bytes in the Attribute block is equal number of switches plus 2.  
The last two bytes are the LED backlight codes.  
Below is the attribute block for a 16 switch system.

Byte	Description
1	Address for LCD Module #1 when switch is pressed.
2	Address for LCD Module #2 when switch is pressed
•	
•	
15	Address for LCD Module #15 when switch is pressed.
16	Address for LCD Module #16 when switch is pressed.
17	LED ON cycle
18	LED Off cycle

## Image Block for 64x32 Resolution Switches:

A block of 256 bytes.

Byte	Description
1-8	First line of picture
9-16	Second line of picture
•	
•	
•	
241-248	31st line of picture
249-256	32nd line of picture

## Bit map format for the Legend block

Byte 8	Byte 7	...	Byte2	Byte 1
D7D6D5D4D3D2D1D0	D7D6...D1D0	...	D7D6...D1D0	D7D6D5D4D3D2D1D0
Byte 16		...		Byte 9
D7D6D5D4D3D2D1D0		...		D7D6D5D4D3D2D1D0
.	.	...	.	.
.	.	...	.	.
.	.	...	.	.
.	.	...	.	.
Byte 256		...		Byte 249
D7D6D5D4D3D2D1D0	D7D6...D1D0	...	D7D6...D1D0	D7D6D5D4D3D2D1D0

Please note this bit map is different than the switch bitmap.

## 4. Communication Protocol

The controller communicates with the host by serial communication via RS232. The default setting is 57.6K baud rate. The controller receives the data via an interrupt routine that places the data on the circular receive buffer. When the controller detects data in the circular receive buffer, it reads one byte and executes the following:

- If the byte is a command, the controller transmits a 61H and executes the subroutine for the command and upon completion of command the controller transmits 79H. One byte commands do not transmit 79H. All of the commands are explained in detail in this section.
- If the byte is not a command, it is ignored.

When the controller executes a subroutine and expects additional information:

- A timer is set. If the expected data byte is not received in 50ms, the controller transmits 6EH and terminates the routine.
- If the byte value is not acceptable (invalid range, option, etc.), the controller transmits 6EH and terminates the routine.

Commands are one byte in the range of 20H to 2FH and 01H and are transmitted in hex format. The controller transmits 61H upon receiving of any of the command byte stated above. When a command has more bytes associated as options or data then the controller transmit 79H upon completion of the command. If a command requires the controller to transmit information, the information will be transmitted after 61H and before 79H. The proper format for all command options and data is specified for each command.



## Commands to the controller

### 1. Commands that do not alter the data in the storage memory:

#### Command to reboot the controller

The command reboots the controller to power-up state.

command format:       **24H**  
transmit format:       (xxH)

After reset, controller transmits 11H indicating it is ready for operation. There is not enough time before reset for 61H to be transmitted for this command.

#### Command to check communication

The command is used to check if the controller is on-line.

command format:       **01H**  
transmit format:       (xxH)

The controller transmits back 61H to the host.

#### Command to update LED codes from storage.

The command cause the controller to put LED codes to the refresh buffer from storage according to the pointers.

command format:       **23H**  
transmit format:       (xxH)

The controller transmits back 61H to the host.

#### Command to refresh the images

This command refreshes the images from storage memory to the specified modules according the active pointers.

command format:   **25H**   [1 to 8]   [9 to 16]   [17 to 24]   [25 to 32]   [33 to 40]  
transmit format:   (xxH) (xxAH) (xxAH)   (xxAH)   (xxAH)   (xxAH)

[1 to 8] is one byte and must be sent in ASCII HEX format. Bit7 is for the module 1 and bit0 is for the module 8. A high bit causes the refresh.

[9 to 16] is one byte and must be sent in ASCII HEX format. Bit7 is for the module 9 and bit0 is for the module 16. A high bit causes the refresh.

[17 to 24] is one byte and must be sent in ASCII HEX format. Bit7 is for the module 17 and bit0 is for the module 24. A high bit causes the refresh.



[25 to 32] is one byte and must be sent in ASCII HEX format. Bit7 is for the module 25 and bit0 is for the module 32. A high bit causes the refresh.

[33 to 40] is one byte and must be sent in ASCII HEX format. Bit7 is for the module 33 and bit0 is for the module 40. A high bit causes the refresh.

**Command to upload a Legend Block**

This command uploads a Legend block from the EEPROM location to the host.

command format: **29H** [Address]

transmit format: (xxH) (xxAH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

The controller transmits back the 256 bytes of the image in ASCII HEX format.

**Command to upload the Pointer values of all the Modules**

This command uploads the active Pointer values for all the modules in ASCII HEX format. There are as many bytes as number of the switches. The bytes are transmitted in ASCII HEX format

command format: **2DH** 52H

transmit format: (xxH) (xxH)

**Command to upload Pointer value for a specified Module**

This command uploads the Pointer value for the specified module in ASCII HEX format.

command format: **2EH** 52H [Module #]

transmit format: (xxH) (xxH) (xxAH)

[Module #] is one byte with value of 01H to 10H and must be sent in ASCII hex format.

**Command to upload LED codes of all Modules**

This command uploads the active LED codes in ASCII HEX format. There are two codes (ON cycle, OFF cycle) for each module.

command format: **2CH** 52H

transmit format: (xxH) (xxH)

**Command to upload LED Code for a specified Module**

This command uploads the LED codes (ON cycle, OFF cycle) of the specified module in ASCII HEX format.

command format: **2FH** 52H [Module #]  
transmit format: (xxH) (xxH) (xxAH)

[Module #] is one byte with value of 01H to 10H and must be sent in ASCII HEX format.

**Command to upload an attribute block for an Address**

This command uploads an Attribute Block from the EEPROM location to the host.

command format: **2BH** [Address]  
transmit format: (xxH) (xxAH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

Upon receiving this command the controller transmits back the [Attribute Block] which can be up to 42 bytes transmitted in ASCII HEX format.

## **2. Operation Commands**

These commands are used in real time operation.

**Command to download Pointer values for all the Modules**

This command downloads the Pointer values for active pointer. If the Pointer value is zero then the corresponding Pointer is left unchanged. The image and LED for all the modules are put to effect immediately.

command format: **2DH** 30H [Pointer Block]  
transmit format: (xxH) (xxH) (xxAH)

[Pointer Block] is many bytes as the number of switches and must be sent in ASCII HEX format.

**Command to download one Pointer value to specified Module**

This command downloads the Pointer value for specified module. The module will display the image and the LED code of the new pointer immediately.

command format: **2EH** 30H [Module #] [Address]  
transmit format: (xxH) (xxH) (xxAH) (xxAH)

[Module #] is one byte with value of 01H to 28H and must be sent in ASCII HEX format.

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

**Command to set LED codes for all Modules**

This command sends the LED codes for all the modules. The new codes go to effect immediately.

command format: **2CH** 30H [LED code block]  
transmit format: (xxH) (xxH) (xxAH)

[LED code block] is two bytes (ON cycle, OFF cycle) for each Module, and must be sent in ASCII HEX format.

If the value of a LED code is zero, the corresponding code in the refresh buffer is left unchanged.

**Command to download one LED Code to specified Module**

This command sends an LED code for a specified module. It goes to effect immediately.

command format: **2FH** 30H [Module #] [LCD/LED code]  
transmit format: (xxH) (xxH) (xxAH) (xxAH)

[Module #] is one byte with value of 01H to 28H and must be sent in ASCII HEX format.

[LED code] is two bytes and must be sent in ASCII HEX format.

**3. Commands to download the pictures, LED codes and control codes.**

These commands change the data in the storage. If a location is being displayed on a module, these changes do not go to effect immediately. They go into effect upon update command, or any command or switch activity that causes an update.

**Command to download an Image block**

This command downloads a Legend Block from the host to a storage location.

command format: **28H** [Address] [Image Block]  
transmit format: (xxH) (xxAH) (xxAH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[Image Block] is 256 bytes and must be sent in ASCII HEX format.

**Command to clear/fill the Image data at a specified Address in**

This command sets all the data bytes of an image to the specified byte. To clear the image this byte should be 00H and to have all ON the byte should be FFH.

command format: **27H** [Address] [00H] [Fill byte]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[00H] is one byte and must be sent in ASCII HEX format (30 30).

[Fill byte] is one byte and must be sent in ASCII HEX format. All the image bytes will be equal this byte.

**Command to write a data byte in an image at a specified Address**

This command writes a byte within an image. The image is 8 byte x 32 Row.

command format: **27H** [Address] [01H] [column] [row] [Byte]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH) (xxAH) (xxAH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[01H] is one byte and must be sent in ASCII HEX format (30 31).

[column] is one byte with value of 01H to 08H and must be sent in ASCII HEX format.

[row] is one byte with value of 01H to 20H and must be sent in ASCII HEX format.

[Byte] is one byte and must be sent in ASCII HEX format.

**Command to write a row of data in an image at a specified Address**

This command write a byte for all the 8 bytes of an image row. To clear a row this byte should be 00H and to have all ON the byte should be FFH. The image is 32 Row.

command format: **27H** [Address] [06H] [row] [Byte]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH) (xxAH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[06H] is one byte and must be sent in ASCII HEX format (30 31).

[row] is one byte with value of 01H to 20H and must be sent in ASCII HEX format.

[Byte] is one byte and must be sent in ASCII HEX format.

**Command to download LED Code to specified Address**

This command sets the LED code for a specified Address.

command format: **27H** [Address] [09H] [LED code ON cycle] [LED code OFF cycle]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH) (xxAH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[09H] is two byte and must be sent in ASCII HEX format (30 39).

[LED code ON cycle] is one byte AND must be sent in ASCII HEX format.

[LED code OFF cycle] is one byte AND must be sent in ASCII HEX format.

The LED code is not transferred to LED code refresh block.

**Command to clear a pixel within an image**

This command sets a pixel within an image to zero (off) at a specified address. The image is 64 bit x 32 Row.

command format: **27H** [Address] [02H] [column] [row]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH) (xxAH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[02H] is two byte and must be sent in ASCII HEX format (30 32).

[column] is one byte with value of 01H to 40H and must be sent in ASCII HEX format.

[row] is one byte with value of 01H to 20H and must be sent in ASCII HEX format.

**Command to fill a pixel within an image**

This command sets a pixel within an image to one (on) at a specified address. The image is 64 bit x 32 Row.

command format: **27H** [Address] [03H] [column] [row]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH) (xxAH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[03H] is two byte and must be sent in ASCII HEX format (30 33).

[column] is one byte with value of 01H to 24H and must be sent in ASCII HEX format.

[row] is one byte with value of 00H to 17H and must be sent in ASCII HEX format.

**Command to download and insert data for the left column of an image**

This command inserts one column of pixel data to the left side of the image. All the columns shift to the right. The right most column pixel data is lost

command format: **27H** [Address] [07H] [pixel data]  
transmit format: (xxH) (xxAH) (xxH) (xxAH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[07H] is one byte and must be sent in ASCII HEX format (30 37).

[pixel data] is FOUR bytes representing the 32 pixels in the column. The top-most pixel is the most significant bit of first byte. The values must be sent in ASCII HEX format.

**Command to download Attribute data to a specified Address**

This command downloads the Attribute data to a specified Address.

command format: **2AH** [Address] [Attribute block] [LED code ON cycle] [LED code OFF cycle]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH) (xxAH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[Attribute block] is 16 bytes of Addresses and must be sent in ASCII HEX format.

[LED code ON cycle] is one byte AND must be sent in ASCII HEX format.

[LED code OFF cycle] is one byte AND must be sent in ASCII HEX format.

**Command to set the Images displayed at power up.**

By default upon power up, module #1 display information at address 01H, module #2 display information at address 02H and so on. This command downloads the Attribute data to a specified Address.

command format: **21H** [CCH] [power up Addresses]  
transmit format: (xxH) (xxAH) (xxAH)

[CCH] is one byte and must be sent in ASCII HEX format (43 43).

[power up Addresses] is up to 40 bytes. One byte for each module. The byte values are from 01H to FFH and must be sent in ASCII HEX format. The default values are 01H, 02H, 03H, ..., ..., 26H, 27H, 28H.

**4. Commands to create image from characters look-up table.**

The controller can generate an image using the font look-up tables. The default look-up tables are ASCII character values 20H to 7FH in fonts 5x7 and 7x10. The user can modify the default look-up tables as well as create a user-defined look-up table for remainder of the values from 01H to FFH.

The font 7x10 is treated as 8x10. The default look up table has B0 as off. The font 5x7 is treated as 6x7. The default look up table has B3 as off. The commands provided do not add spacing so it is possible to create user defined characters for graphic block.

**Command to create a line of character in font 5x7**

This command creates a bitmap for one line of 5x7 character font and places it from starting row of the image data at a specified address. The ten characters are created from the 3rd pixels to 62<sup>nd</sup> pixels of each row. The pixels 1, 2, 63 and 64 are not changed.

command format: **27H** [Address] [08H] [Start row] [ASCII string]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH) (xxH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[08H] is two byte and must be sent in ASCII HEX format (30 38).

[Start row] is one byte with value of 01H to 19H and must be sent in ASCII HEX format.

[ASCII string] is 10 bytes and is a string of font 5x7 look-up table indexes(01H to FFH). It must be sent in HEX format.

### **Command to create a line of character in font 7x10**

This command create bitmap for one line of 7x10 character font and place it from starting row of the image data at a specified address. All the pixels in the rows are used to create the 8 characters.

command format: **27H** [Address] [04H] [Start row] [ASCII string]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH) (xxH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[04H] is one byte and must be sent in ASCII HEX format (30 34).

[Start row] is one byte with value of 01H to 16H and must be sent in ASCII HEX format.

[ASCII string] is 8 bytes and is a string of font 7x10 look-up table indexes. It must be sent in HEX format.

### **Command to create a 7x10 font character in an image at a specified Address**

This command writes a character within an image. The starting byte for the character is in column and row data. The image is 8 byte x 32 Row. All the bits of starting byte are used for the character.

command format: **27H** [Address] [05H] [column] [row] [Character]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH) (xxAH) (xxH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[05H] is one byte and must be sent in ASCII HEX format (30 35).

[column] is one byte with value of 01H to 08H and must be sent in ASCII HEX format.

[row] is one byte with value of 01H to 16H and must be sent in ASCII HEX format.

[Character] is one byte and must be sent in HEX format (01H to FFH). This the ASCII code for the character

### **Command to create a 7x10 font character in an image at a specified Address (split between two bytes)**

This command writes a character within an image. The starting byte for the character is in column and row data. The image is 8 byte x 32 Row.

The character is created on B3 to B0 of starting byte and B7 to B4 of the next bytes.

command format: **27H** [Address] [0CH] [column] [row] [Character]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH) (xxAH) (xxH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[0CH] is one byte and must be sent in ASCII HEX format (30 43).

[column] is one byte with value of 01H to 08H and must be sent in ASCII HEX format.

[row] is one byte with value of 01H to 16H and must be sent in ASCII HEX format.

[Character] is one byte and must be sent in HEX format. This the ASCII code for the character



**Command to create a 5x7 font character in an image at a specified Address.**

This command writes a character within an image. The starting byte for the character is in column and row data. The image is 8 byte x 32 Row.

The character is created on B7 to B2 of starting byte. Other bits in the bytes are not affected.

command format: **27H** [Address] [0DH] [column] [row] [Character]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH) (xxAH) (xxH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[0DH] is one byte and must be sent in ASCII HEX format (30 44).

[column] is one byte with value of 01H to 08H and must be sent in ASCII HEX format.

[row] is one byte with value of 01H to 16H and must be sent in ASCII HEX format.

[Character] is one byte and must be sent in HEX format. This the ASCII code for the character

**Command to write a 5x7 font character in an image at a specified Address.**

This command writes a character within an image. The starting byte for the character is in column and row data. The image is 8 byte x 32 Row.

The character is created on B5 to B0 of starting byte. Other bits in the bytes are not affected.

command format: **27H** [Address] [0EH] [column] [row] [Character]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH) (xxAH) (xxH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[0EH] is one byte and must be sent in ASCII HEX format (30 45).

[column] is one byte with value of 01H to 08H and must be sent in ASCII HEX format.

[row] is one byte with value of 01H to 16H and must be sent in ASCII HEX format.

[Character] is one byte and must be sent in HEX format. This the ASCII code for the character

**Command to write a 5x7 font character in an image at a specified Address (split between two bytes)**

This command writes a character within an image. The starting byte for the character is in column and row data. The image is 8 byte x 32 Row.

The character is created on B3 to B0 of starting byte and B7 to B6 of the next bytes. Other bits in the bytes are not affected.

command format: **27H** [Address] [0FH] [column] [row] [Character]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH) (xxAH) (xxH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[0FH] is one byte and must be sent in ASCII HEX format (30 46).

[column] is one byte with value of 01H to 08H and must be sent in ASCII HEX format.

[row] is one byte with value of 01H to 16H and must be sent in ASCII HEX format.

[Character] is one byte and must be sent in HEX format. This the ASCII code for the character

**Command to write a 5x7 font character in an image at a specified Address (split between two bytes)**

This command writes a character within an image. The starting byte for the character is in column and row data. The image is 8 byte x 32 Row.

The character is created on B1 to B0 of starting byte and B7 to B4 of the next bytes. Other bits in the bytes are not affected.

command format: **27H** [Address] [10H] [column] [row] [Character]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH) (xxAH) (xxH)

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[10H] is one byte and must be sent in ASCII HEX format (31 40).

[column] is one byte with value of 01H to 08H and must be sent in ASCII HEX format.

[row] is one byte with value of 01H to 16H and must be sent in ASCII HEX format.

[Character] is one byte and must be sent in HEX format. This the ASCII code for the character

**5. Commands to download to Look-up tables.**

For look up table report based on switch activity to work the appropriate flag bit must be set in addition to look up table data.

**a. Switch Report Look-up Table:****Command to download user-defined Switch Press event for a switch.**

This command sets user-defined actions taken when the specified switch is pressed using the look-up table.

command format: **21H** [AAH] [Module #] [number] [string]  
transmit format: (xxH) (xxAH) (xxAH) (xxH) (xxH)

[AAH] is one byte and must be sent in ASCII HEX format (41 41).

[Module #] is one byte with value of 01H to 28H and must be sent in ASCII HEX format.

[number] is one byte (00H to 0FH) which indicates the number of bytes. Only the lower nibble is used.

[string] may consist of up to a maximum of 15 bytes according to the lower nibble of the [number].

All default values for this section are 00H.

**Command to download user-defined Switch Release event for a switch.**

This command sets user-defined actions taken when the specified switch is released using the look-up table.

command format: **21H** [ABH] [Module #] [number] [string]  
transmit format: (xxH) (xxAH) (xxAH) (xxH) (xxH)

[ABH] is two byte and must be sent in ASCII HEX format (41 42).

[Module #] is one byte with value of 01H to 10H and must be sent in ASCII HEX format.

[number] is one byte which indicates the number of bytes. Only the lower nibble is used.

[string] may consist of up to a maximum of 15 bytes according to the lower nibble of the [number].  
All default values for this section are 00H.

**Command to download user-defined Image Press event for an address.**

This command sets user-defined actions taken when any switch with the Image associated with the specified address is pressed using the look-up table.

command format: **21H** [BBH] [Address] [number] [string]  
transmit format: (xxH) (xxAH) (xxAH) (xxH) (xxH)

[BBH] is two byte and must be sent in ASCII HEX format (42 42).

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[number] is one byte which indicates the number of bytes. Only the lower nibble is used. If B4=1 then the string will be repeatedly send while the Image is pressed.

[string] may consist of up to a maximum of 15 bytes according to the lower nibble of the [number].  
All default values for this section are 00H.

**Command to download user-defined Image Release event for an address.**

This command sets user-defined actions taken when any switch with the Image associated with the specified address is released using the look-up table.

command format: **21H** [BCH] [Address] [number] [string]  
transmit format: (xxH) (xxAH) (xxAH) (xxH) (xxH)

[BCH] is two byte and must be sent in ASCII HEX format (42 43).

[Address] is one byte with value of 01H to FFH and must be sent in ASCII HEX format.

[number] is one byte which indicates the number of bytes. Only the lower nibble is used. If B4=1 then the string will be repeatedly send while the Image is pressed.

[string] may consist of up to a maximum of 15 bytes according to the lower nibble of the [number].  
All default values for this section are 00H.

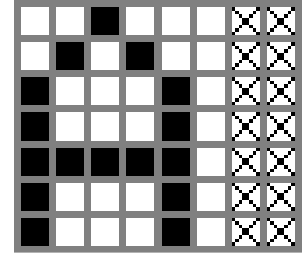
**b. Font-Set Look-up Table:**

There are 255 fonts (01H to FFH) in each font set. The default bit map look-up tables are created for English ASCII characters (20H to 7FH). The default values for rest of the fonts are 00H. User can change the default as well as create fonts for unused locations.

**Command to download a font to the look-up table for font 5x7**

This command downloads the look up bit map for one font of font 5x7.

command format: **21H** [BEH] [font Index] [font bitmap]  
 transmit format: (xxH) (xxAH) (xxAH) (xxH)



[BEH] is one byte and must be sent in ASCII HEX format (42 45).

[font Index] is one byte with value of 01H to FFH and must be sent in ASCII HEX format. It is the look-up table index.

[font bitmap] is 7 bytes making up a character graphic. The first byte is the top row of the character graphic and the 7th byte is the bottom row of the character graphic.

To allow for graphic block, 6 bits of each byte are used (B7 to B2). In the default font B2 of each byte is used as space.

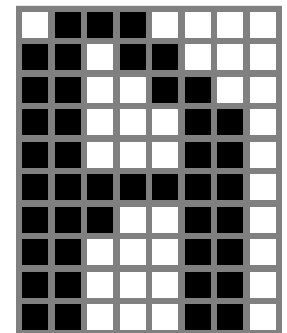
The B7 of each byte of the character graphic data is the leftmost column of the character graphic. B1 and B0 are not used and should be set to zero.

Example: (21 4245 3030 20 50 88 88 F8 88 88) sends the above illustration to look-up table index 00.

**Command to download a font to the look-up table for font 7x10**

This command downloads the look up bit map for one font of font 7x10.

command format: **21H** [BDH] [font value] [font bitmap]  
 transmit format: (xxH) (xxAH) (xxAH) (xxH)



[BDH] is two bytes and must be sent in ASCII HEX format (42 44).

[font value] is one byte with value of 00H to FFH and must be sent in ASCII HEX format. It is the look-up table index.

[font bitmap] is 10 bytes making up a character graphic. The first byte is the top row of the character graphic and the 10th byte is the bottom row of the character graphic.

To allow for graphic block, all bits of each byte are used. In the default font B0 of each byte is used as space.

The B7 of each byte of the character graphic data is the leftmost column of the character graphic.

Example: (21 4244 3030 70 D8 CC C6 C6 FE E6 C6 C6 C6) sends the above illustration to look-up table index 00.

## 6. Download Set-up Values

**Command to set the time for LED's Off Cycle duration.**

This command sets the timer value for the duration of the LED's Off Cycle.

command format: **21H** [E1H] [Timer value] [Timer value]  
 transmit format: (xxH) (xxAH) (xxAH) (xxAH)

[E1H] is two byte and must be sent in ASCII HEX format (45 31).

[Timer value] is one byte with value of 01H to FFH sent twice and must be sent in ASCII HEX format. The default value is 40H.

Off Cycle Duration = 7 (Timer value) ms.

**Command to set the On Cycle/Off Cycle ratio.**

This command sets the timer value for the duration of the LED's Off Cycle.

command format: **21H** [E2H] [Ratio] [Ratio]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH)

[E2H] is two byte and must be sent in ASCII HEX format (45 32).

[Ratio] is one byte with value of 01H to FFH sent twice and must be sent in ASCII HEX format. The default value is 03H. The ratio value determines the On Cycle duration for LED's.

On Cycle Duration = (Ratio) (Off Cycle duration) ms.

**Command to set the number of Modules attached to the controller.**

This command sets the number of Modules attached to the controller.

command format: **21H** [E9H] [# of Modules] [# of Modules]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH)

[E9H] is one byte and must be sent in ASCII HEX format (45 39).

[# of Modules] is one byte with value of 01H to 28H and must be sent twice in ASCII HEX format. This command goes into effect after the following power up / reboot.

## 7. Setting the Flags

The controller features can be customized based on customer requirements by manipulating the settings of the Flags. There are three Flag bytes and a command for setting each Flag byte. The changes go into effect after the following power up / reboot. All the RESERVED bits should be set to zero.

**Command to set a Flag Byte A.**

This command sets the flag byte A which determines the features of the controller.

command format: **21H** [EEH] [Flag Byte] [Flag Byte]  
transmit format: (xxH) (xxAH) (xxAH) (xxAH)

[Flag Byte] is one byte and must be sent twice in ASCII HEX format. This command writes the downloaded flag data to storage memory. The changes go into effect after the following power up / reboot.

## Flag Byte A Default Value=63H

Bit	Enable Value	Flag	Controller action when flag is set
B0	1	Default Switch release report	Reports switch release events (Module # + C0H) in XXH format.
B1	1	Default Switch press event report	Reports switch press events (Module # + 80H) in XXH format.
B2	1	User-defined Switch press event report	Transmits user-defined data for that switch upon switch press event in XXH format.
B3	1	Default Image press event report	Transmits Pointer value upon switch press event in XXAH format.
B4	1	User-defined Image press event report	Transmits user-defined data for that Image upon Image press event in XXH format.
B5	1	Execute Attribute for Switch press enable	updates Pointer values if non-zero according to the pointer block.
B6	1	Switch press event beep enable	Provides audio feedback through the audio connector upon any switch press event.
B7		RESERVED	

### Flag Byte A bit settings:

B0=0 ---> No transmission.

B0=1 ---> Upon switch release, transmit one byte (C0H+switch number).

B1=0 ---> No transmission.

B1=1 ---> Upon switch press, transmit one byte (80H+switch number).

B2=0 ---> No transmission.

B2=1 ---> Upon switch press, transmit from the user-defined switch press look up table (0 to 15 bytes).

B3=0 ---> No transmission.

B3=1 ---> Upon switch press, transmit the Pointer value of the Module in ASCII HEX format (01H to FFH).

B4=0 ---> No transmission.

B4=1 ---> Upon switch press, transmit from the user-defined Image press look up table (0 to 15 bytes).

B5=0 ---> Do not execute the Attributes.

B5=1 ---> Execute Attributes upon switch press.

B6=0 ---> Disable buzzer for switch presses.

B6=1 ---> Enable buzzer beep for when any switch is pressed.

## Command to set a Flag Byte B.

This command sets the flag byte B which determines the features of the controller.

command format: **21H** [EFH] [Flag Byte] [Flag Byte]  
 transmit format: (xxH) (xxAH) (xxAH) (xxAH)

[Flag Byte] is one byte and must be sent twice in ASCII HEX format. This command writes the downloaded flag data to storage memory. The changes go into effect after the following power up / reboot.

### Flag Byte B Default Value=60H

Bit	Enable Value	Flag	Controller action when flag is set
B0	0	LED code from storage	Uses the LED Code from storage. Default is enabled.
B1	0	RESERVED	
B2	1	RESERVED	
B3	1	RESERVED	
B4	1	XON/XOFF enable	Uses XON/XOFF protocol to communicate.
B5	1	79H flag	79H is transmitted upon completion of command.
B6	1	61H flag	61H is transmitted in response to command.
B7	1	RESERVED	

### Flag Byte B bit settings:

B0=0 ---> When executing attribute pointers, the Controller pull the LCD/LED code from Pages to refresh Buffer.

B0=1 ---> Do not use LCD/LED codes from pages.

B1=0 ---> Upon power up/ reboot, use the pointers that were in effect before power down/ reboot.

B1=1 ---> Upon power up/ reboot, use the pointers from Power-up look up table.

B4=0 ---> Disable

B4=1 ---> Enable

Use XON(11H) and XOFF(13H) for communication. The controller stops transmitting upon receiving 13H and resumes transmitting upon receiving 11H. The transmit buffer is 256 bytes. The data will be lost if more than 256 bytes is to be transmitted. Upon power up the controller will be in XOFF mode.

B6=0 ---> No transmission.

B6=1 ---> Upon receipt of a command byte (20H to 2FH) from host, the controller transmits 61H in XXH format. Regardless of B6 setting, the controllers transmit 61H in respond to Command 01H from host.



## Command to set a Flag Byte C.

This command sets the flag byte C which determines the features of the controller.

command format: **21H** [EDH] [Flag Byte] [Flag Byte]  
 transmit format: (xxH) (xxAH) (xxAH) (xxAH)

[Flag Byte] is one byte and must be sent twice in ASCII HEX format. This command writes the downloaded flag data to storage memory. The changes go into effect after the following power up / reboot.

### Flag Byte C Default Value=00H

Bit	Enable Value	Flag	Controller action when flag is set
B0	1	Default Image release report	Transmits Pointer value upon Image release event in XXAH format.
B1	1	User-defined Image release report	Transmits user-defined data for that switch upon switch release event in XXH format.
B2	1	User-defined switch release report	Transmits user-defined data for that switch upon switch release event in XXH format.
B3	1	User-defined Image press event repeat	Repeatedly transmits user-defined Image press event data upon Image press event in XXH format.
B4	1	Multi-Drop	Multi-Drop.
B5	1	RESERVED	
B6	1	Timer for command 26H	Controller waits for additional optional bytes until user-defined timer expires.
B7	1	RESERVED	

### Flag Byte C bit settings:

B0=0 ---> No transmission.

B0=1 ---> Upon switch release, transmit Pointer value of the switch in ASCII HEX format.

B1=0 ---> No transmission.

B1=1 ---> Upon switch release, transmit from the user-defined image release look up table (0 to 15 bytes).

B2=0 ---> No transmission.

B2=1 ---> Upon switch release, transmit from the user-defined switch release look up table (0 to 15 bytes).

B3=0 ---> Disable user-defined Image press repeat.

B3=1 ---> Upon switch press, transmit from the user-defined image press look up table (0 to 15 bytes).

B4=0 ---> Normal operation

B4=1 ---> Multi-drop operation for RS422 option.

This option should be selected when many of the controllers are connected to RS422 communication network.

To communicate to any of the controller FFH followed by the unit ID of that controller must be transmitted. The controller will be on line until FEH is received. When the controller is online all the communication is as normal operation. In multi-drop operation XONXOFF option should be activated so the transmit from controllers are managed by the master host.

B6=0 ---> Disable timer for command 26H.

B6=1 ---> Enable timer for command 26H. Controller waits for additional optional bytes until user-defined timer expires. (50ms)

## 8. Controller Set-up, Command 26H

This command is for set-up/testing of the controller. The controller is put into the set-up mode, where any number of options may be transmitted. This is the only command where the timer between two consecutive received bytes is not set. The user must exit this mode by sending X (58H) or ESC (1BH). It is recommended that the B6 of Flag-C to be set during the operation.

### Command 26H.

command format: **26H** [Option(s)]  
 transmit format: (xxH) (xxH)

Option **Read Unit ID**  
 41H

Controller transmits the unit ID to the host. (Default ID = 41H)

Option **Set Unit ID**  
 42H [Unit ID (1 byte)]

Controller sets the unit ID to the next byte received.

Option **Red LED Backlighting on**  
 43H

Controller sends the red color code to the modules directly. It does not affect the refresh buffer.

Option **Green LED Backlighting on**  
44H

Controller sends the green color code to the modules directly. It does not affect the refresh buffer.

.....

Option **White LED Backlighting on**  
45H

Controller sends the white color code to the modules directly. It does not affect the refresh buffer.

.....

Option **LED Backlighting Off**  
46H

Controller sends the OFF color code to the modules directly. It does not affect the refresh buffer.

.....

Option **Turn all the module pixels ON**  
47H

Turn all the pixels ON for all the modules.

.....

Option **Turn all the module pixels OFF**  
48H

Turn all the pixels OFF for all the modules.

.....

Option **Increases LED Brightness Level**  
49H

Controller reads the current LED brightness level, increments it and stores its value. The new LED brightness value goes into effect immediately and remains permanently, even after reboot. There are 8 level of brightness.

.....

Option **Decreases LED Brightness Level**  
4AH

Controller reads the current LED brightness level, decrements it and stores its value. The new LED brightness value goes into effect immediately and remains permanently, even after reboot. There are 8 level of brightness.

.....

Option **Toggle Beeper On/Off**  
4DH

Controller toggles the beeper On/Off. The effect lasts for only the duration of the session.

.....

Option **Toggle Switch Closure Beep**  
4EH

Controller toggles the flag for audible response to switch closures. The effect lasts for only the duration of the session.

.....

Option **Return Controller to Default State**  
4FH

Controller alters the virgin check key memory, so that the subsequent power-up or reset brings the controller to the default stage. All the attribute block, look up table are replaced with default. It only replaces the image one with "BLANK FONT MEMORY". All other images are unchanged.

.....

Option **Set Attribute Execution On**  
50H

Controller enables the flag for attribute execution. The effect lasts for only the duration of the session.

.....

Option **Set Attribute Execution Off**  
51H

Controller disables the flag for attribute execution. The effect lasts for only the duration of the session.

.....

Option **Return Firmware Version**  
52H

Controller transmits the firmware Version in ASCII Hex.

**Example:** (61 32 43 31 36 30 33 48 56 31 32 79)Hex translates to 2C1603HV12

.....

Option **Serial Set-up Mode**  
53H     *[serial setup option]*

Controller is put into serial set-up mode. Once in serial set-up mode, only options for serial set-up are available. The mode is not changed until power-up or reset.

31H	Select 9600 BAUD rate
32H	Select 19200 BAUD rate
33H	Select 57600 BAUD rate
49H	Write the selected option to the Ram, transmit 79H and exit command 26H.
53H	Transmit the baud rate to the host: 9600 (20H 39H 36H 30H 30H), 19200 (31H 39H 32H 30H 30H) and 57600 (35H 37H 36H 30H 30H).

**Example:** (26 53 33 49) sets the baud speed to 57600. It goes to effect upon next power-up or reset.

.....

Option **Reserved**  
54H

.....

Option **Reserved**  
55H

.....

Option **Reserved**  
56H

.....

Option **Reserved**  
57H

.....

Option **Enable Switch Release Reports**  
61H

Controller sets the flag which enables switch release reports. The effect lasts for only the duration of the session.

.....

Option **Disable Switch Release Reports**  
62H

Controller sets the flag which disables switch release reports. The effect lasts for only the duration of the session.

.....

Option **Disable Switch Closure Beep**  
63H

Controller sets the flag which disables the switch closure beep. The effect lasts for only the duration of the session.

.....

Option **Enable Switch Closure Beep**  
64H

Controller sets the flag which enables the switch closure beep. The effect lasts for only the duration of the session.

.....

Option **Turn Beeper Off**  
65H

Controller turns beeper off. The effect lasts for only the duration of the session.

.....

Option **Turn Beeper On**  
66H

Controller turns beeper on. The effect lasts for only the duration of the session.

.....

Option **Disable Switch Press Reports (Default value)**  
67H

Controller sets the flag which disables switch press reports (default value). The effect lasts for only the duration of the session.

.....

Option **Enable Switch Press Reports (Default value)**  
68H

Controller sets the flag which enables switch press reports (default value). The effect lasts for only the duration of the session.

.....

Option **Disable Switch Press Reports (User-Defined)**  
69H

Controller sets the flag which disables switch press reports (User-Defined). The effect lasts for only the duration of the session.

.....

Option **Enable Switch Press Reports (User-Defined)**  
6AH

Controller sets the flag which enables switch press reports (User-Defined). The effect lasts for only the duration of the session.

.....



Option **Disable Image Press Reports (Default value)**  
6BH

Controller sets the flag which disables Image press reports (Default value). The effect lasts for only the duration of the session.

.....

Option **Enable Image Press Reports (Default value)**  
6CH

Controller sets the flag which enables Image press reports (Default value). The effect lasts for only the duration of the session.

.....

Option **Disable Image Press Reports (User-Defined)**  
6DH

Controller sets the flag which disables Image press reports (User-Defined). The effect lasts for only the duration of the session.

.....

Option **Enable Image Press Reports (User-Defined)**  
6EH

Controller sets the flag which enables Image press reports (User-Defined). The effect lasts for only the duration of the session.

.....

Option **Exit command 26H**  
1BH

Controller transmits 79H and exits command 26H.

.....

Option **Exit command 26H**  
58H

Controller transmits 79H and exits command 26H.

.....

Option **Any non-option byte**  
XXH

Controller transmits 6EH and exits command 26H.

## 5. Hardware

**WARNING:** These products are ESD sensitive. The ESD handling procedure must be followed.

Power Requirement: 6.5 V to 12 V. Maximum current is 200 mA plus number of switches times 50.

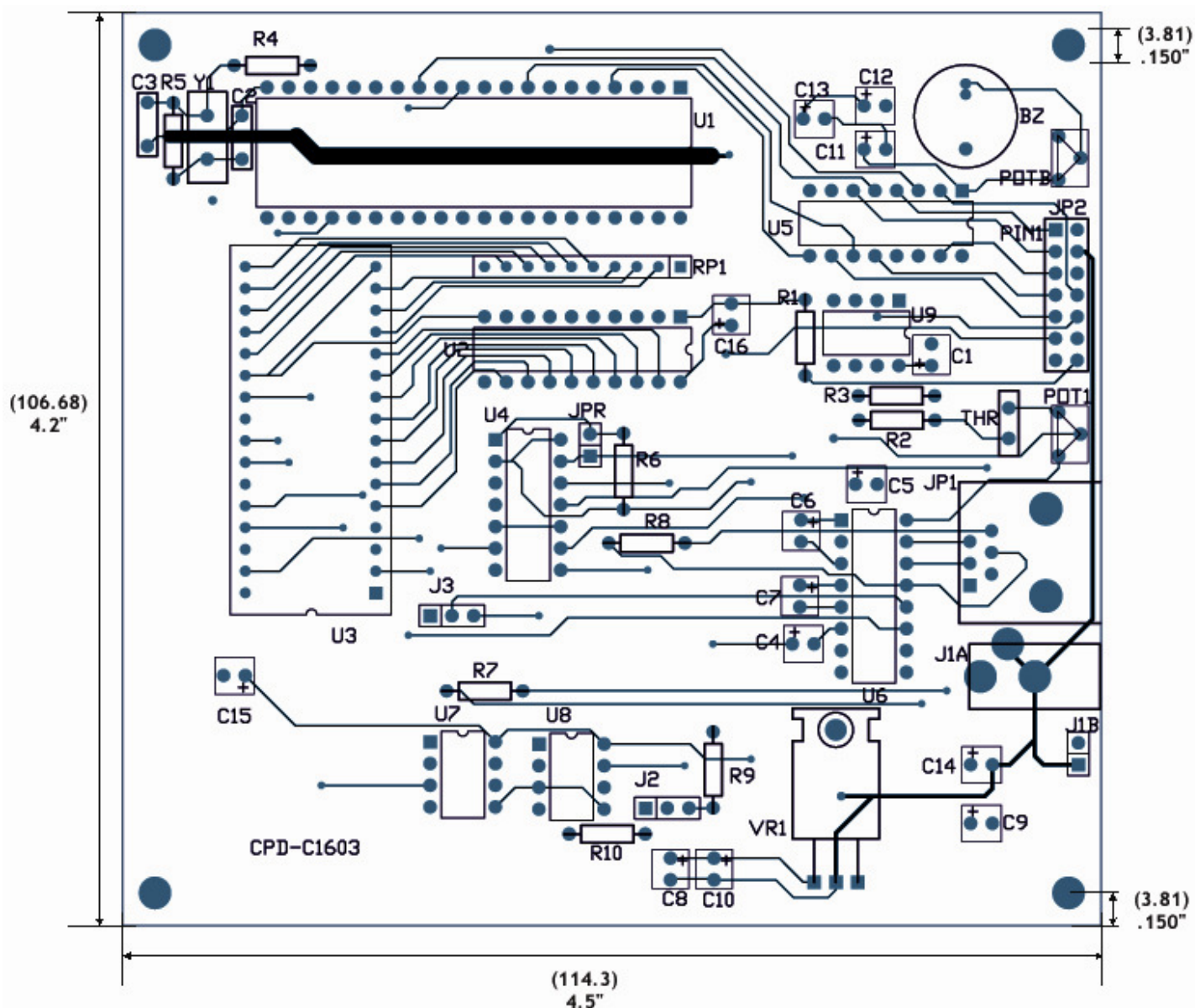
### Dimensions

Dimension: Board size: 4.500" x 4.200" (114.3 x 106.68 mm)

Mounting holes: Four

Hole size: .150" (3.81mm)

Distance from edges of the board: .150" (3.81mm)



## Controls Overview

**POTB: Beeper Volume** adjusts the volume of the beeper.

**JP2: Logic Board Header** is the connector for the ribbon cable to the logic boards. Pin numbers much match up from connector to connector using Pin 1 as a guide. 7x2 male header .1"x.1" spacing.

**Note: Improper installation of the Logic Boards could damage either/both the Logic Boards and controller.**

**POT1: not used.**

**JP1: RS232/ RS422/RS485 Connector** links the controller to the host.

	1	2	3	4	5	6
RS232	.....	GND	TX	RX	GND	.....

	1	2	3	4	5	6
RS422/485	T-	NC	T+	R-	NC	R-

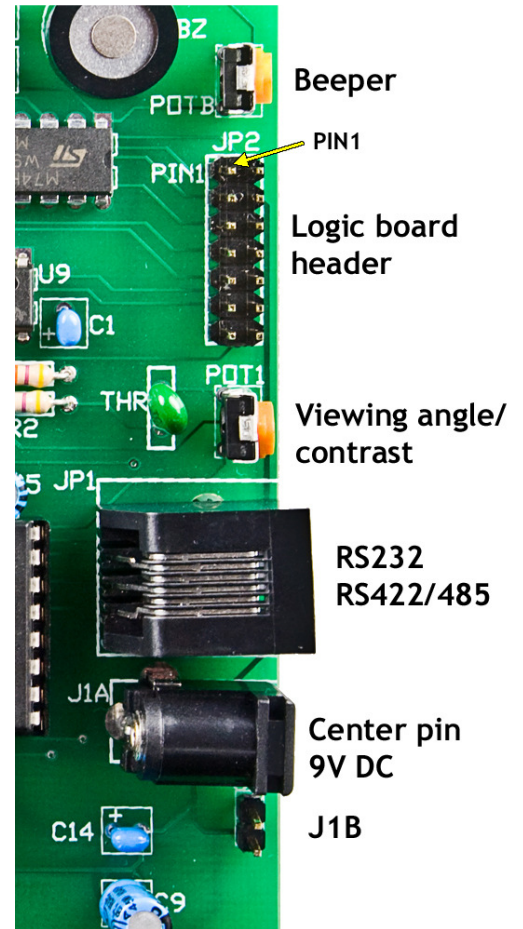
For RS232 U6 is populated with MAX232 and U7 and U8 are not populated. For RS422 U7 and U8 are populated with SN75176 and U6 is not populated. For RS485 only U7 is populated.

**J1A: 9V DC Power** jack mates with a 2.5mm cylinder power connector. Center positive.

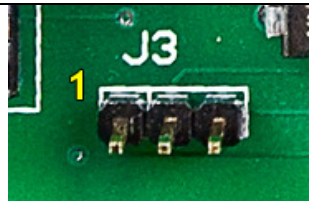
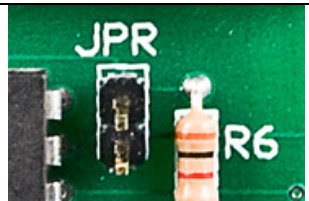
**J1B:** Alternative power connector: 2 pin header .1" spacing. Pin1 GND, pin2 +9V.

## Logic Board Header (JP2) pin call outs:

Pin	Function
1	Data for logic board shift register
2	GND
3	Clock for Logic board shift registers
4	GND
5	Not used
6	+9V, VCC
7	Latch pulse for Logic Board latches
8	Out put enable for Logic Boards latches
9	Serial clock for LCD modules
10	Not used
11	Data for LCD modules
12	+9V, VCC
13	Not used
14	SWREAD

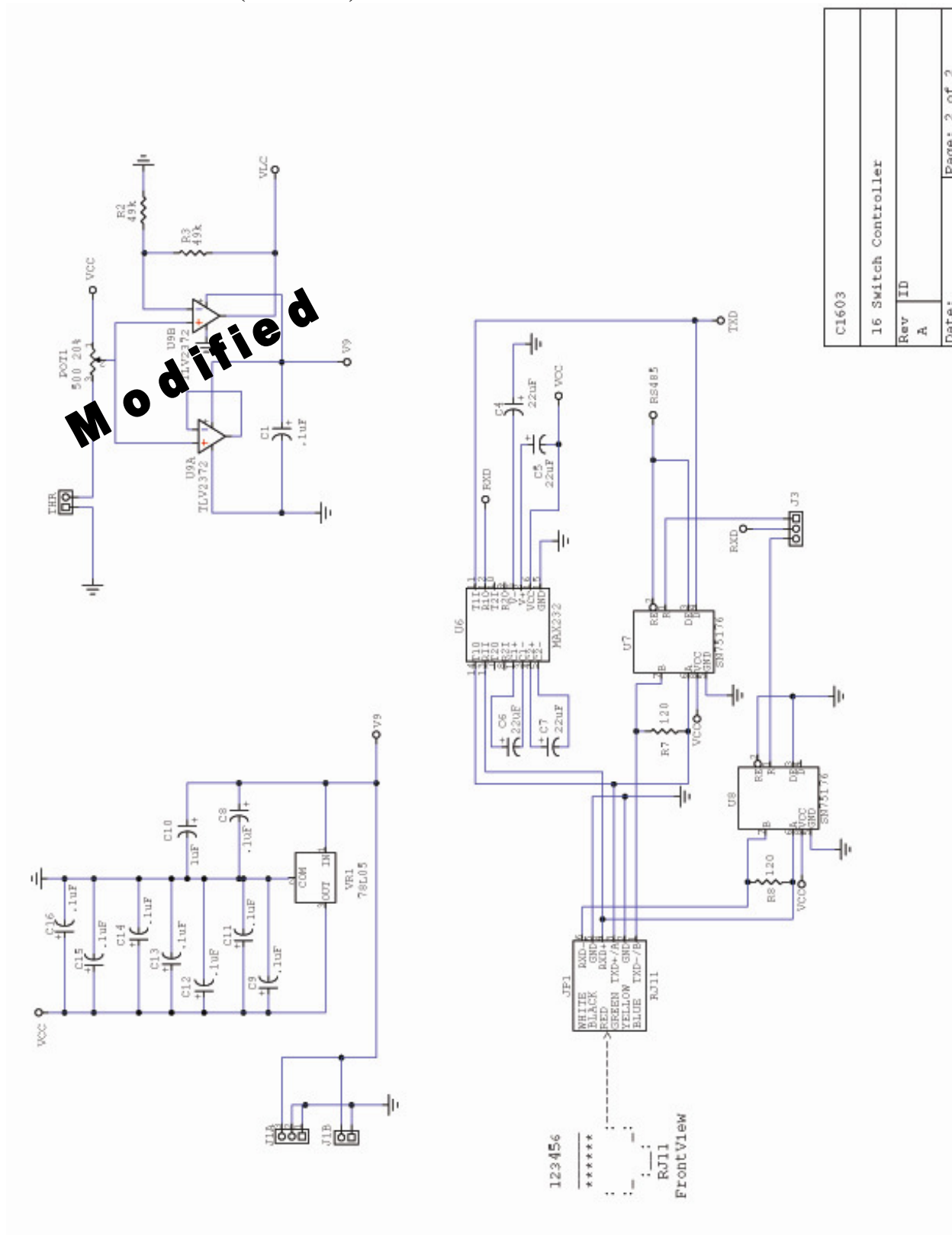


## Jumper Overview

<p>J3:      RS232: No jumper                      RS485: Jumper on pin 1 and pin 2                      RS422: Jumper on pin 2 and pin 3                      Unnecessary for programming.</p>	
<p>JPR:      Used for programming the DS89C450 microcontroller. It should be placed for the duration of programming.</p>	

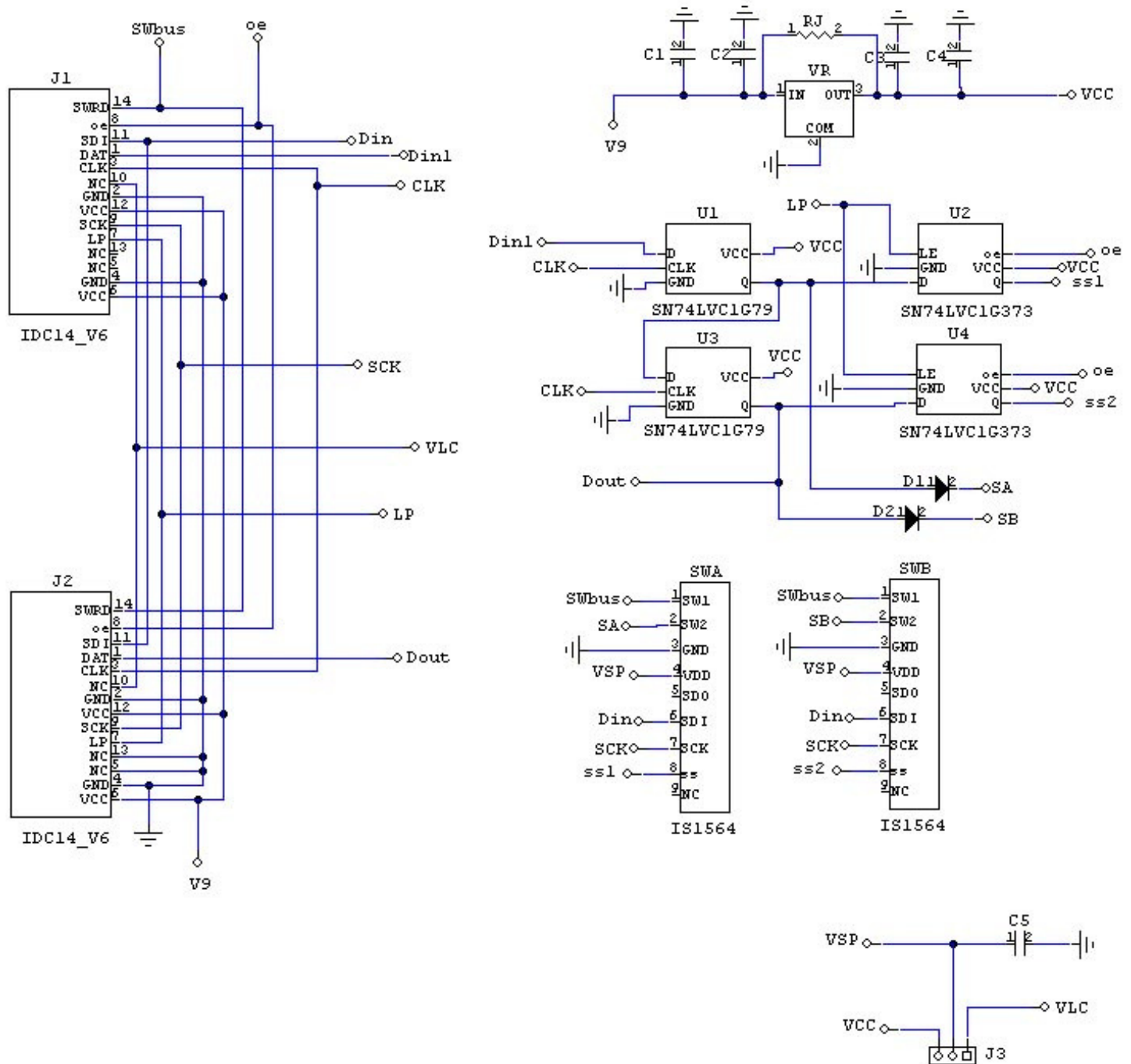




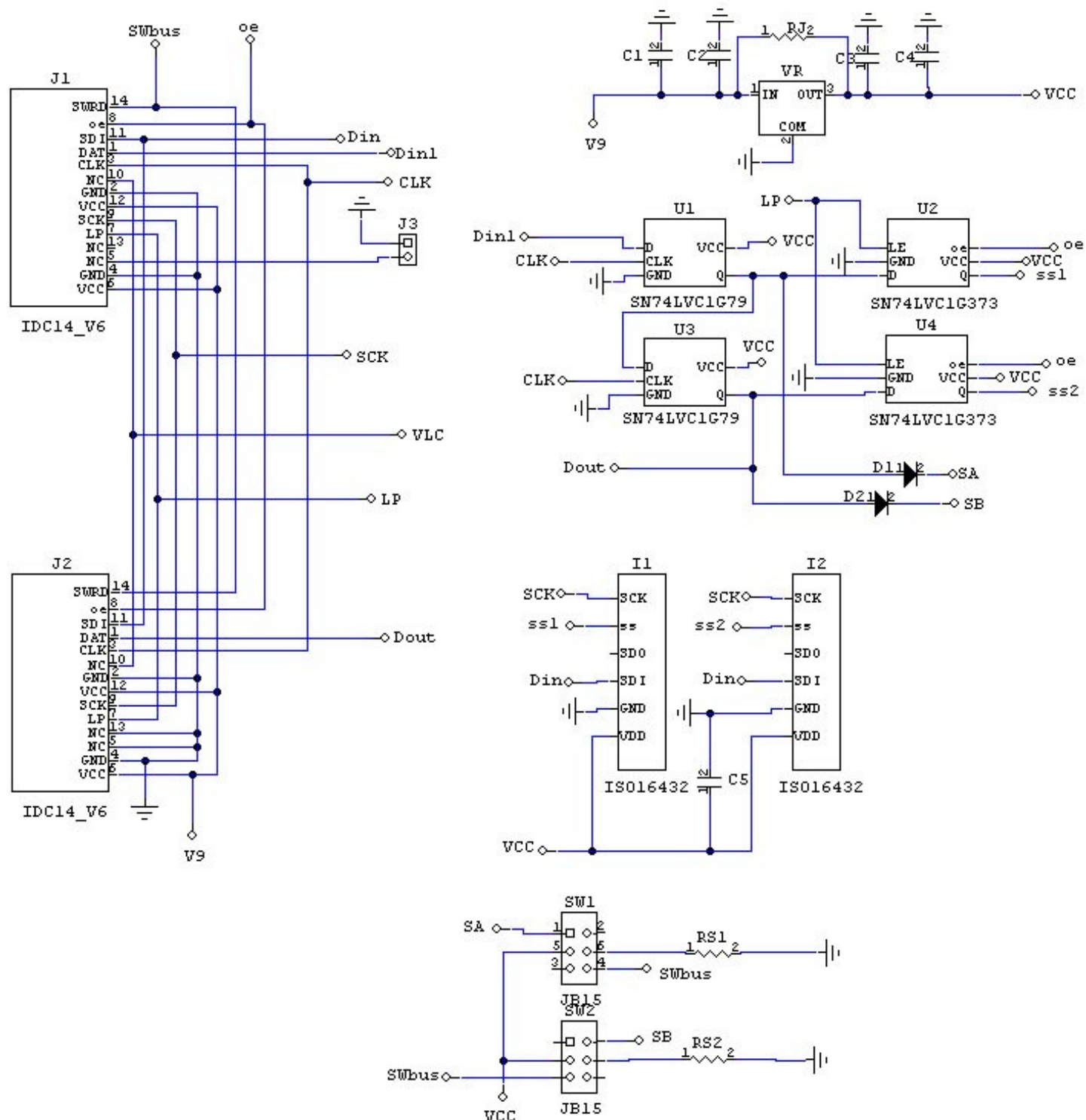




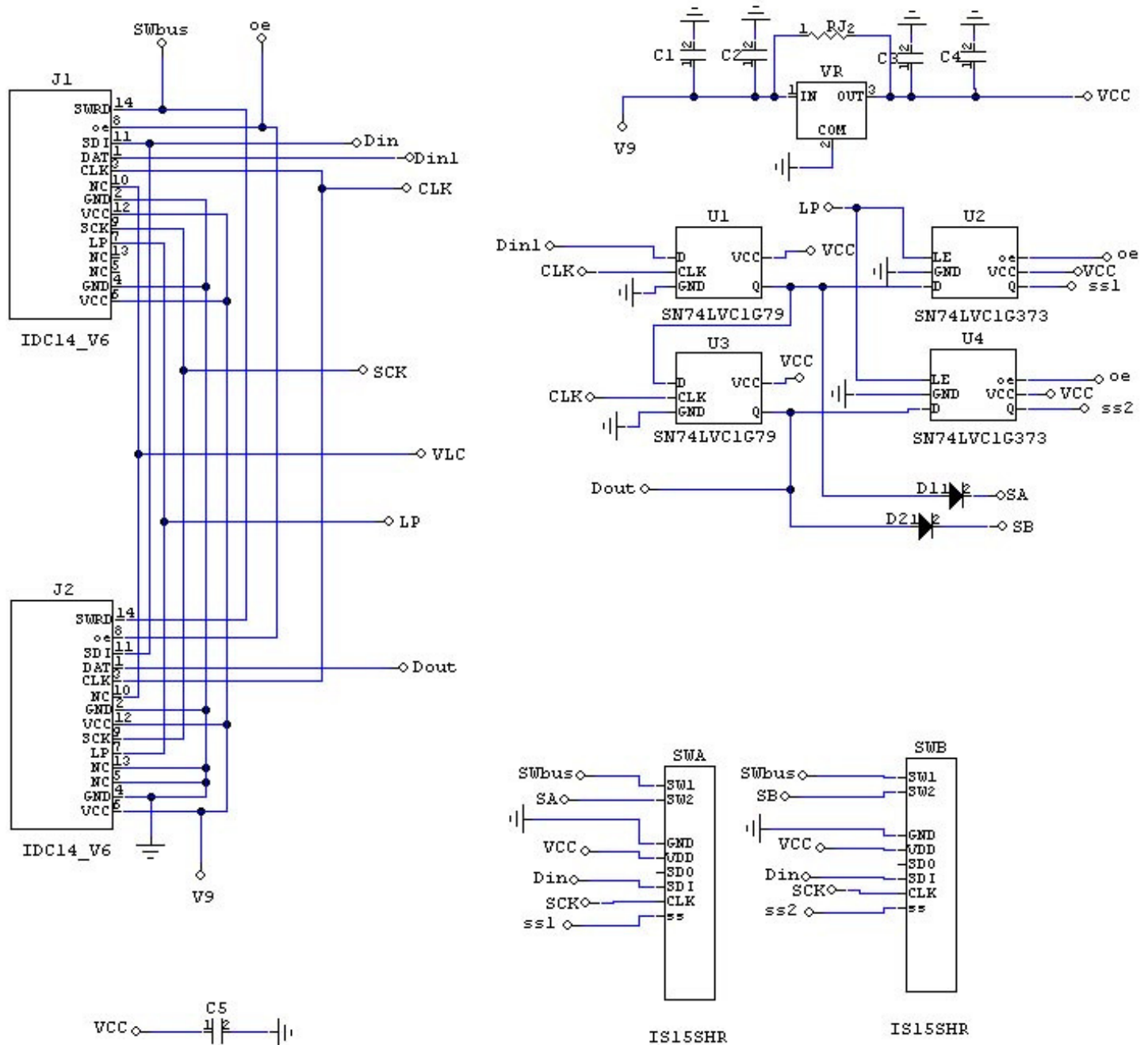
## IS-L02A1, logic board, two 64x32 RGB pushbuttons



## IS-L02F1, logic board, two 64x32 RGB Display



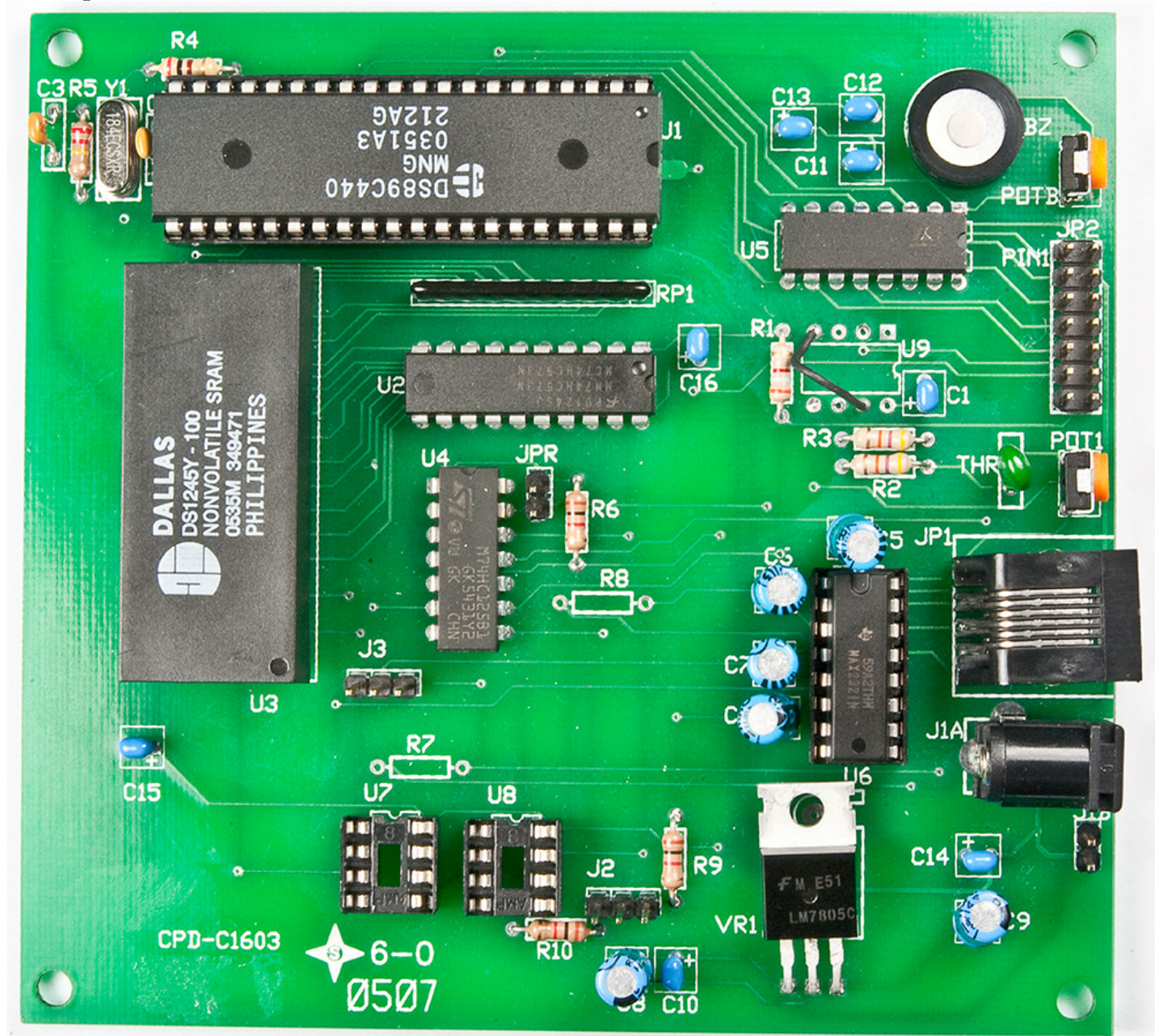
## IS-L02H2, logic board, two 64x32 RGB Compact







**Board photo**



## 6. Key Terms & Definitions

<b>Module</b>	NKK Switches' LCD SmartSwitches and SmartDisplays. Sometime referred to as switch.
<b>Host</b>	Any computer, terminal, or other device that can communicate over the RS232, RS422, RS485 line.
<b>Controller</b>	A PCB assembly that controls one or more logic boards and the switches associated with them. It communicates with a host over the RS232 line.
<b>Logic Board</b>	A PCB assembly with "glue logic" for mounting switches. It is controlled by a controller.
<b>Byte</b>	An eight bit hex value ranging from 00H to FFH (Decimal 0 to 255). The bit format of a byte is: (B7 B6 B5 B4 B3 B2 B1 B0) where B7 is most significant and bit B0 is least significant bit.
<b>Nibble/Hex digit</b>	A four bit value ranging from 0H to FH. A byte consists of two nibbles.
<b>ASCII</b>	A byte value representing a symbol.
<b>Communication Format</b>	<p>There are two formats to transmit a byte:</p> <ol style="list-style-type: none"> <li>1. <b>Hex format</b> - A hex byte is transmitted without any change to it. [xxH] will be used to denote this.</li> </ol> <p>All commands and some data are sent by using this format.</p> <ol style="list-style-type: none"> <li>2. <b>ASCII HEX format</b> - Each nibble of the byte is converted to ASCII code and sent as a byte. [xxAH] will be used to denote this.</li> </ol> <p>For example, the hex byte 5AH is transmitted in two bytes, <b>35H</b> and <b>41H</b>. The ASCII value for <b>5</b> is <b>35H</b> and the ASCII value for <b>A</b> is <b>41H</b>.</p> <p>All addresses and most data are sent using this format.</p>
<b>Module Number</b>	A one byte value ranging from 01H to 10H. One for each Module.
<b>Pointer</b>	Each module has a pointer. The pointer value is the current address of the module. It is a one byte value ranging from 01H to FFH.
<b>Address</b>	A one byte value ranging from 01H to FFH representing the 255 memory locations.

**Page** The data at each address. Each Page has image data, LCD/LED data, attribute block data, and user-define picture closure/release data.

## 7. Programming the Firmware

The controller firmware can be programmed via the RS232 connection.

The Dallas semiconductor MTK software can be downloaded from the site below.

<http://www.maxim-ic.com/products/microcontrollers/software/index.cfm#development>

After you install the MTK software follow the following instructions to program the microcontroller:

1. Disconnect the switches from the controller at JP2.
2. Install a jumper on JPR connector (top right side of the board). This will put the microcontroller into programming mode.
3. Run MTK software.
4. Select device DS89C450.
5. Under menu item “Options” select the “Configure Serial Port” and select relevant port and 19200 baud rate.
6. Under menu item “Target” select “Open COM X at 19200 baud”.
7. Plug in or power up the IS-C1603H controller.
8. Under menu item “Target” select “Connect to Loader”.
9. The microcontroller will respond with a version number and copyright notice in blue.
10. Under menu item “File” select “Load Flash”.
11. Specify the HEX file provided by NKK Switches (e.g. C1603V13.HEX)
12. Once loaded the display shall say “Load Complete”

**Note: The switches should not be connected while programming the controller since they are not being refreshed during programming.**

## 8. Firmware Issues

The problems found in the most current version are listed below and will be resolved in the following version.

### Version 1.3

1. No firmware issues at this time.



## Appendix

### Understanding Decimal, Hexadecimal and Binary Numbers

Decimal is the numbering system we use. It is called base-10. Every digit can be between 0 to 9. The value of each digit is equal to the digit with ((Digit order) -1) zero in front.

Table 1, Base-10								
Value in base-10 Digit x				10000	1000	100	10	1
Multiplication for converting to decimal	$10^{(N-1)}$	...	...	$10^4$	$10^3$	$10^2$	$10^1$	$10^0$
Digit order	Nth digit	....	....	Fifth digit	Fourth digit	Third digit	Second digit	First digit

**Example:** The value of each digit of the Base-10 number 7605 is as follow:

Fourth digit: 7 with (4-1) zero = 7000  
 Third digit: 6 with (3-1) zero = 600  
 Second digit: 0 with (2-1) zero = 0  
 First digit: 5 with (1-1) zero = 5

#### Numbering system

A numbering system can be based on any number (base-N). However it has to follow the rules:

- Each digit has to be between 0 to (Base minus one). For example:
  - Each digit for Base-2 numbering system can be 0 or 1
  - Each digit for Base-5 numbering system can be 0 to 4
  - Each digit for base-8 numbering system can be 0 to 7

- The value of each digit is equal to the digit with ((Digit order) -1) zeros in front.

All the operations that we use on base 10 numbering system such as addition, subtraction, multiplication, division... works the same for all the numbering systems. The difference is the carry over will be based on the base-number of the numbering system instead of 10.

#### Why do we need other base numbering systems?

Computers logic is based on two states:

Yes or No

False or True

High voltage or low voltage



## Base-2 numbering system (binary)

The numbering system to accommodate the computer logic is called binary or base 2. Each digit of binary can be 0 or 1.

Table 2, Base-2								
Value in base-2 Digit x				10000	1000	100	10	1
Multiplication for converting to decimal	$2^{(N-1)}$	...	...	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
Digit order	Nth digit	....	....	Fifth digit	Fourth digit	Third digit	Second digit	First digit

**Example:** The value of each digit of the Base-2 number 1010 is as follow:

Fourth digit: 1 with (4-1) zero = 1000  
 Third digit: 0 with (3-1) zero = 0  
 Second digit: 1 with (2-1) zero = 10  
 First digit: 0 with (1-1) zero = 0

To convert a base-2 number to decimal, multiply each digit by multiplier and add them together:

**Example:** converting base-2 number 1010 to decimal is as follow:  
 $(1 \times 8) + (0 \times 4) + (1 \times 2) + (0 \times 1) = 10$

To convert a decimal number to base-2 number

---divide the decimal number by 2, the remainder is the first digit of the base-2 number

--- continue dividing quotient by 2 and put the remainder as the next digit until the quotient is equal 0.

**Example:** Convert the decimal number 21 to base-2

21 divide by 2 = 10 with 1 remainder	First digit = 1
10 divide by 2 = 5 with 0 remainder	Second digit = 0
5 divide by 2 = 2 with 1 remainder	Third digit = 1
2 divide by 2 = 1 with 0 remainder	Fourth digit = 0
1 divide by 2 = 0 with 1 remainder	Fifth digit = 1

21 decimal = 101001 base-2 or binary

The base-2 number 101001 reads as one zero one zero zero one.

## Base-16 numbering system

Communicating base-2 numbers is difficult for human because of all zero's and one's. To make it easier to present computer data, hexadecimal or base-16 numbering system is used. Four digit of base-2 numbering system convert to one digit of base-16 numbering system. Since we did not have digits for 10, 11, 12, 13, 14 and 15, they were assigned letters A, B, C, D, E and F respectively.

Table 3, Number base conversion		
Hexadecimal base-16	Decimal base-10	Binary base-2
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111
10	16	10000
11	17	10001
etc	etc	etc

Table 4, Base-16								
Value in base-16 Digit x				10000	1000	100	10	1
Multiplication for converting to decimal	$16^{(N-1)}$	...	...	$16^4$ =65536	$16^3$ =4096	$16^2$ =256	$16^1$ =16	$16^0=1$
Digit order	Nth digit	....	....	Fifth digit	Fourth digit	Third digit	Second digit	First digit

**Example:** Convert 2A7 hex to decimal

$$\begin{array}{rcl}
 \text{Digit 3: } 2 \times 256 & = & 512 \\
 \text{Digit 2: } (10) \times 16 & = & 160 \\
 \text{Digit 1: } 7 \times 1 & = & 7 \\
 \hline
 & & 679
 \end{array}$$

2A7 hex = 679 decimal

**Example:** Convert 925 decimal to hex

925 divide by 16 = 57 with 13 remainder	First digit = D
57 divide by 16 = 3 with 9 remainder	Second digit = 9
3 divide by 16 = 0 with 3 remainder	Third digit = 3

925 decimal = 39D hex

Converting between hex and binary is as easy as replacing each digit of hex with equivalent 4 digit of binary.

**Example:** convert A5B hex to binary

A = 1010  
 5 = 0101  
 B = 1011

A5B hex = 1010 0101 1011 binary

Converting binary to hex is as easy as replacing each 4 digit of binary to equivalent digit of hex. If the binary digits are not multiple of 4 for grouping, add enough zero to the left to make them multiple of 4.

**Example:** Convert 0111 1010 1111 binary to hex

1111 = F  
 1010 = A  
 0111 = 7  
 0111 1010 1111 binary = 7AF hex

## Common terms:

Bit = binary digit  
 Nibble = 4 binary digits  
 Byte = 8 binary digits = 2 nibbles = 2 HEX digits