

IS-Dev Kit-8 User Manual

Revision E

Version 3.0

IS18WWC1W Development Kit.



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 <engineering@nkkswitches.com>

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. Thank you for purchasing NKK Switches' IS-Dev Kit-8..... | 4 |
| 3. Preface..... | 5 |
| 4. Operational Overview | 6 |
| 5. Programming the IS-Dev Kit-8..... | 8 |
| 6. Communication Protocol | 16 |
| 7. Commands | 17 |
| 8. Character Commands..... | 24 |
| 9. Hardware..... | 27 |
| 10. Key Terms & Definitions..... | 31 |
| 11. Firmware problems | 32 |
| A. Appendix | 33 |

1. General Features

The IS-Dev Kit-8 Version 3.0 controls one monochromatic OLED Rocker with a resolution of 96x64 pixels. The CL04 controller is used in IS-DEV KIT-8 as well as IS-DEV KIT-7/7D. For complete features of the CL04 controller refer to IS-CL04R user manual.

Below are current features:

- Controls one OLED Rocker (IS18WWC1W).
 1. Three button rocker; top, middle, and bottom.
 2. White monochrome OLED display with a resolution of 96x64 pixels.
 3. IP64 dust protected & splashproof.
 4. Snap-in mount.

- Power from USB or 9V power supply. (Power supply not included).
- On board MicroSD Flash that can hold up to 65,536 images.
- Adjustable audio feedback when a switch is pressed.
- Software control for 16 levels of brightness setting to extend OLED life.
- User programmable for images, attributes and set-ups.
- Real-time control by host.
 - Commands to direct the controller to display any of 65536 images
 - Command to send characters so controller create images for the displays
 - Commands to check various statuses of the controller
 - Report switch activities to host
 - Report the address of any new image displayed to host
- Stand-alone operation
 - Set up the controller to function based on switch activities.
 - Set up to the controller to function based on timer expire.
 - Report switch activities to host
 - Report the address of any new image displayed to host

- Communication via USB (57.6K, 1 start bit, 8 bit, 1 stop bit).
- Set the type of activity reports from the controller to host.
- Power requirement: 5V to 12V at 120 mA Max
- Controller board firmware can be customized based on customer requirements.

2. Thank you for purchasing NKK Switches' IS-Dev Kit-8

By purchasing this kit a new horizon of design is expanding before your eyes. The OLED Rocker can be the distinguishing feature within your application that sets your product apart from the competition. This kit contains all the necessary components to get you started using, testing and ultimately incorporating NKK Switches' OLED Rocker into your designs.

The development kits come preprogrammed for demonstration. They can be reprogrammed by the user for demonstration or real time application. NKK provides a free PC software (Universal Communicator 2015) for programming the development kits.

There are many documents such as “How to” in the web page below that can be useful.

<http://www.nkkswitches.com/SmartSwitch-Engineering-Documentation.aspx>

The part numbers that come with each kit are listed below.

IS-DEV KIT-8

| | Part Number | QTY | Description |
|---|-------------|-----|---------------------------------|
| 1 | IS-CL04R | 1 | Controller |
| 3 | IS18WWC1W | 2 | OLED monochrome Rocker (96x64) |
| 4 | IS-USB1 | 1 | Cable, USB 2.0, A to MINI-B, 6' |

3. Preface

The IS-Dev Kit-8 is a development kit designed to demonstrate the features of the SmartSwitch OLED rocker as well as facilitate the incorporation of the rocker into new application designs. The OLED Rocker is best used for displaying and navigating through a menu so operator can select an action and/or displaying statuses for operator to monitor.

This user manual will go through general features and rudimentary commands, like how to download images and attributes, and then go into the more technical details of how the Dev Kit-8 operates.

Section 4 and 5 briefly describe the basic Dev Kit operations and provide a starter guide for programming the Dev Kit with the most popular commands using the Universal Communicator program. These sections were designed for those who wish to simply start using the Dev Kit without getting into the technical details.

Section 6, 7, and 8 covers communication and commands between the host computer and the Dev Kit controller. Section 9 is list of newly added commands. Section 10 is the hardware and the schematic and section 11 and 12 are definitions and firmware issues. There is also an appendix for understanding hexadecimal.

Universal Communicator 2015

Universal Communicator 2015 can download the images and attributes to the development kit. The 64 x 48 images can be created by any graphic software that can save as 24 color bitmap. The attribute values that determine how the development kit works can be created in Microsoft Excel or Notes.

The development kit can communicate with any communication software that can perform the communication protocol. Universal Communicator 2015 is provided as demonstration software.

4. Operational Overview

Power-up and Brightness Adjustment:

When the power is turned on, the OLED displays the image below and allows for brightness adjustment. The brightness level is displayed in the middle.



There are 16 level of brightness 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F where F is the maximum brightness. The level of brightness is displayed in the middle of image. After power-up, the display is at the maximum brightness level (F). Pressing the bottom switch will make the display dimmer. Pressing the top switch will make the display brighter. The OLED rated life of 52,000 hours is based on the maximum brightness. The dimmer the level is set the longer the life of the OLED. Any switch activity will keep the unit in this mode.

The brightness changes stay in effect for duration of the session.

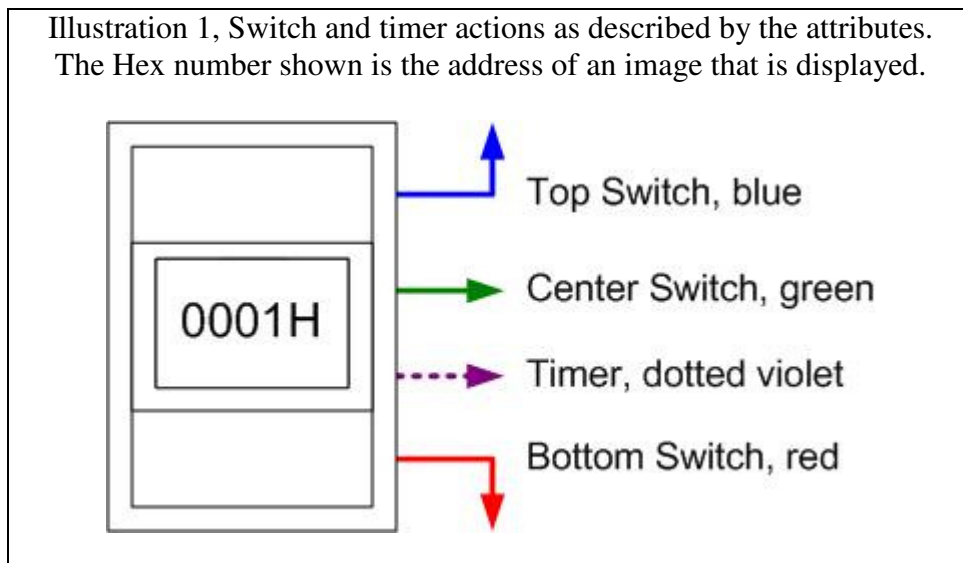
After 2.6 seconds of user inactivity the controller will move into the main operations mode.

Main Operational Mode

The IS-DEV KIT-8 is programmed with factory default images. The user can change the images and programming through the use of attributes. The images are pictures to be displayed. The attributes are user defined information on what image to display based on the switch activity and when the timer expires. After the brightness adjustment mode, the OLED Rocker displays the image at address 1 and uses the attribute information of address 1. There is room for 65,535 images and attributes (address 1 to 65535) in memory.

The controller associates one address to each image and the display shows the image at the associated address. Each address has a group of seven attributes which together is called an Attribute Block. The function of the Attribute Block is to tell the controller what image to display next either when the switch associated with that particular address is pressed or when the timer expires. These are called actions.

For visual illustration, through out the document blue is used to show a top switch action, green is a middle switch action, a red is the bottom switch action, and a dotted violet line is the timer action.



Attribute Information (Attribute Block)

There are seven attributes per image.

1. Top Switch Address. The address of the image to be displayed if the top switch is pressed.
 When the top switch is pressed:
 - A. If the address is set to 0x0000 there will be no change.
 - B. If the address is set to 0x0001 to 0xFFFF the image at the address is displayed and the attributes at the address become active.
2. Middle Switch Address. The address of the image to be displayed if the middle switch is pressed.
 When the middle switch is pressed:
 - A. If the address is set to 0x0000 there will be no change.
 - B. If the address is set to 0x0001 to 0xFFFF the image at the address is displayed and the attributes at the address become active.
3. Bottom Switch Address. The address of the image to be displayed if the bottom switch is pressed.
 When the bottom switch is pressed:
 - A. If the address is set to 0x0000 there will be no change.
 - B. If the address is set to 0x0001 to 0xFFFF the image at the address is displayed and the attributes at the address become active.
4. End Address. The address of the last image to be displayed in an image sequence starting at the current address and ending at the end address where the speed of the sequence is based on the timer value.
 - A. If the timer is set to zero this is attribute is ignored.

5. Timer. The timer controls how fast to cycle through displaying the images sequentially. The timer can be up to 65 seconds. The images change sequentially from the beginning address (the active attribute address) to the end address (4)
 - A. If the timer is set to zero, there will be no changes based on the timer.
 - B. If the timer is set to a positive number, the timer starts running. If any of the switches are pressed, the timer gets reset. If a switch press causes a new image to be displayed the attributes associated with the new image become active.

When the timer expires:

For each timer expire:

- If the address of the image displayed is not equal to the End address (4):
 - i. Then the next image is displayed. (The Attribute Block of the new image is ignored.)
 - ii. And the timer is reset and starts running again.
- If the address of the image being displayed is equal to the End address (4) then the Middle address (2) of the End address (4) attribute block is checked.
 - i. If the Middle address (2) is to zero, the image from the beginning address is displayed. This will create a loop of the images from beginning address to end address to be displayed until a switch press causes a change
 - ii. If the Middle address (2) is a positive number, the image at the address is displayed and the attributes at the address become active.

6 & 7 Reserved for future use and must be set to zero.

5. Programming the IS-Dev Kit-8

The IS-DEV KIT-8 communicates via USB with any communication software. The commands and protocol will be explained in sections 6 and 7.

This section covers programming the Dev Kit using the Universal Communicator 2015 without getting into technical details.

Creating Images

The OLED Rocker display is 96x64 pixel monochrome. Images can be created graphically or by cropping from a picture using graphic software such as Paint, Photoshop, etc. The images must be saved as 96x64 monochromatic bitmap files. The image files should be saved in a single folder. Universal Communicator imports the files from the folder in alphanumeric order according to the file names. The file names should be selected so the order of images will be as desired.

There is documentation on creating images and videos in the “How to” section of this webpage:

<http://www.nkkswitches.com/SmartSwitch-Engineering-Documentation.aspx>

Naming Images

The Universal Communicator 2015 loads the images in alphabetic order according to the image file names and displays the addresses that each image will have in the controller.

Be sure to name images so the video images or loop images are listed sequentially. Avoid using symbols in the names as some symbols interfere with alphabetic ordering.

It is not necessary to have all images in one folder as Universal Communicator 2015 allows for selecting a starting address for download. However, for making attributes, it is easier if all the images are in one folder so the image addresses in the controller can be checked in the Universal Controller.

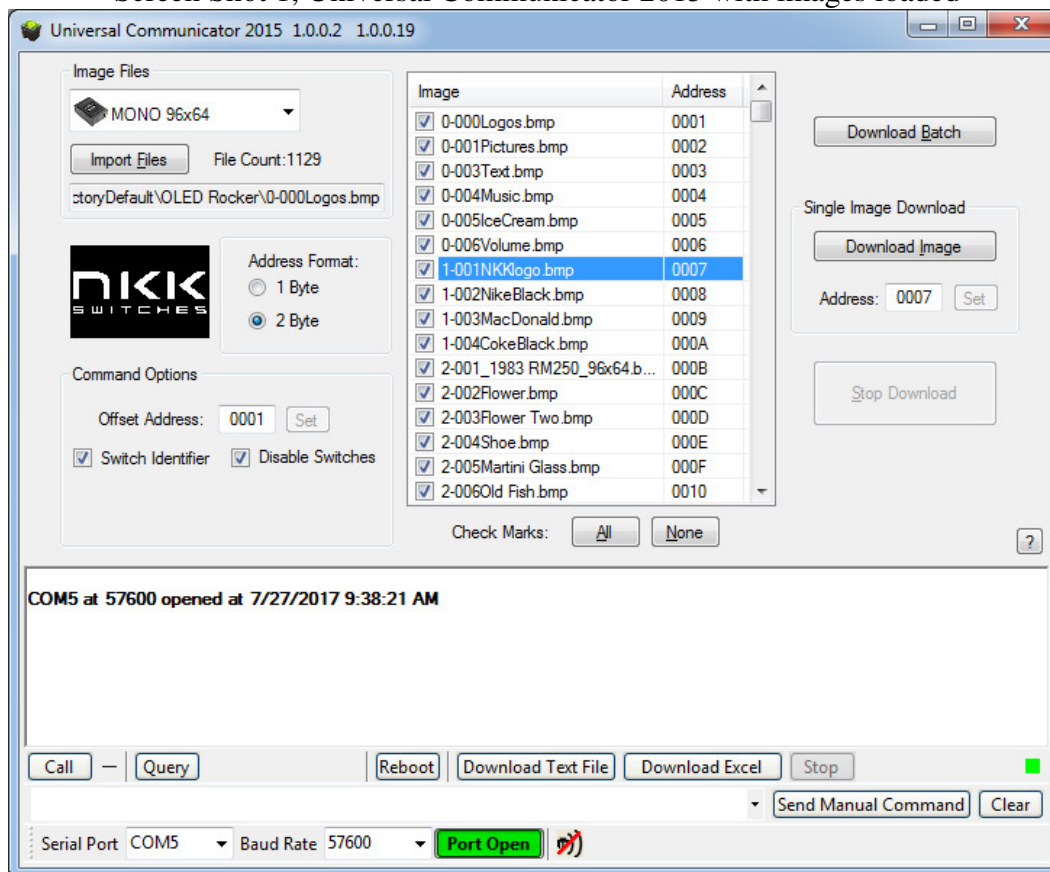
Downloading the Images to the Controller

Start the Universal Communication 2015 software and follow the following steps:

1. Click “Import Files” and then navigate to the image folder and click “OPEN”. The images will load to the Universal Communicator. Notice the addresses next to the image names. These will be the image addresses in the controller. Clicking on any image name will display the image on the left of the screen.
2. Make sure the development kit is connected to a computer USB connection.
3. Select the serial communication port number to which the development kit is connected.
4. Select Baud Rate 57600.
5. Click the “Port Closed” button to open the communication. If the development kit has the factory default images and program, the controller will transmit information to the computer which will be displayed in green font. For example: “83” and “84” will be displayed. The addresses for the images are also displayed.

6. Click the “Call” button. The computer will send “01” in blue and the controller responds with “61” in green. This may not be seen if the controller is sending the timer expirations and addresses. However, if the controller does not send anything, there is likely a problem with the serial connection or set up. It should be fixed before proceeding.
7. Click “Download Batch”. The controller will respond with “6179” for each image downloaded. Every image takes 2.5 to 3.5 seconds to download. Universal Communication 2015 will display an error message if anything is wrong or notify when the download is complete.
8. Leave Universal Communication 2015 open in order to look up the image addresses for creating the attributes block which determine how the various images are displayed on the OLED switches.

Screen Shot 1, Universal Communicator 2015 with images loaded



Timer

The timer controls the number of milliseconds to display an image before switching to the next in the sequence. The timer consists of two numbers called Timer2 and Timer1. Each timer is a two-digit hex number between 00H and FFH (0 to 255 decimal). If Timer1 is set to zero, the timer does not run. If Timer2 is set to 0 it will count as 256.

The amount of time in millisecond = (Timer2) x (Timer1)

The maximum timer is when the Timer1 = FFH and Timer2 = 00H:

The maximum timer value = 255x256= 65280 ms = 65.28 Seconds. The time should not be set for less than 13 ms.

The timer for 25 frame/s video is 40 ms so some possible Timer1 and Timer2 values can be as followed:

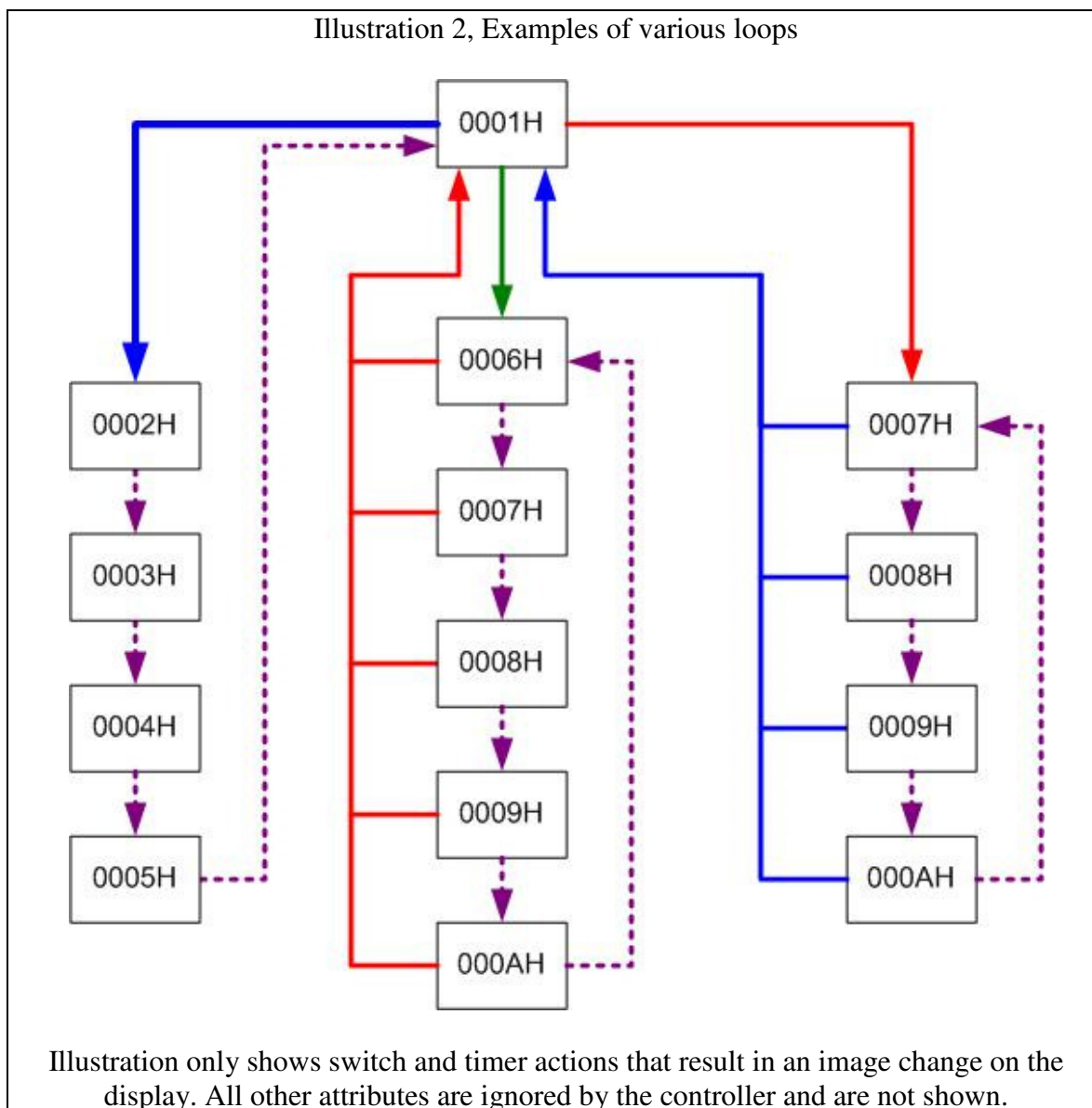
Timer1 = 01H and Timer2= 28H or Timer1=05H and Timer2=08H.

For HEX numbering and conversion between HEX and decimal please refer to the appendix.

An Example of How Attributes Work

The following demonstrates the way attributes work. Illustration 2 (below) visually demonstrates the way these attributes work.

The blue is the top switch action, the green is the middle switch action, the red is the bottom switch action, and the dotted violet line is the timer action.



When the Dev Kit enters the Main Operational Mode the image at the address 0001H is displayed. The Attribute Block at address 0001H becomes active.

Attribute Block at address 0001H

| | Top | Middle | Bottom | | | |
|---------|---------|---------|---------|---------|--------|--------|
| Current | Switch | Switch | Switch | End | Timer2 | Timer1 |
| Address | Address | Address | Address | Address | | |
| 0001H | 0002H | 0006H | 0007H | 0000H | 00H | 00H |

Since the Timer1 is zero the timer does not run so there will be no changes based on the timer. The image at the address 0001H is displayed until any of the switches are pressed:

Pressing the Top Switch

Attribute Block at address 0001H

| | Top | Middle | Bottom | | | |
|---------|---------|---------|---------|---------|--------|--------|
| Current | Switch | Switch | Switch | End | Timer2 | Timer1 |
| Address | Address | Address | Address | Address | | |
| 0001H | 0002H | 0006H | 0007H | 0000H | 00H | 00H |

If the top switch is pressed, the image at address 0002H is displayed and the Attribute Block of address 0002H becomes active.

Attribute Block at address 0002H

| | Top | Middle | Bottom | | | |
|---------|---------|---------|---------|---------|--------|--------|
| Current | Switch | Switch | Switch | End | Timer2 | Timer1 |
| Address | Address | Address | Address | Address | | |
| 0002H | 0000H | 0000H | 0000H | 0005H | FFH | 01H |

Since all the switch addresses are zero there will be no changes if a switch gets pressed. The timer is 255ms. If there is no switch activity for 255ms the timer expires causing the next image, 0003H, to be displayed. This continues until images 0004H and 0005H are also displayed. When the timer expires at address 0005H the address is equal to the End Address. The Middle Switch Address from the Attribute Block of 0005H is then checked. Please note the attribute information of addresses 0003H and 0004H are not used.

Attribute Block at address 0005H

| | Top | Middle | Bottom | | | |
|---------|---------|---------|---------|---------|--------|--------|
| Current | Switch | Switch | Switch | End | Timer2 | Timer1 |
| Address | Address | Address | Address | Address | | |
| 0005H | 0002H | 0001H | 0014H | 0010H | 24H | 35H |

The image at address 0001H is displayed and the Attribute Block of address 0001H becomes active. Notice that the other attributes within the Attribute Block of address 0005H are ignored.

Pressing the Middle Switch

Attribute Block at address 0001H

| | Top | Middle | Bottom | | | |
|---------|---------|---------|---------|---------|--------|--------|
| Current | Switch | Switch | Switch | End | Timer2 | Timer1 |
| Address | Address | Address | Address | Address | | |
| 0001H | 0002H | 0006H | 0007H | 0000H | 00H | 00H |

If the middle switch, the image at the address 0006H is displayed and the Attribute Block of the address 0006H becomes active.

Attribute Block at address 0006H

| | Top | Middle | Bottom | | | |
|---------|---------|---------|---------|---------|--------|--------|
| Current | Switch | Switch | Switch | End | Timer2 | Timer1 |
| Address | Address | Address | Address | Address | | |
| 0006H | 0000H | 0000H | 0001H | 000AH | FFH | 02H |

The timer is 510ms. If there is no switch activity for 510ms the timer expires causing the next image, 0007H, to be displayed. This continues until images 0008H, 0009H, and 000AH are also displayed. When the timer expires at address 000AH the address is equal to the End Address. The Middle Switch Address from the Attribute Block of 000AH is then checked.

Attribute Block at address 000AH

| | Top | Middle | Bottom | | | |
|---------|---------|---------|---------|---------|--------|--------|
| Current | Switch | Switch | Switch | End | Timer2 | Timer1 |
| Address | Address | Address | Address | Address | | |
| 000AH | 000CH | 0000H | 000AH | 0000H | 00H | 00H |

Since the Middle Switch Address is zero then the image at address 0006H is displayed and the Attribute Block of address 0006H stay active.

If the bottom switch is pressed in any address from 0006H to 000AH then the image at address 0001H is displayed and the Attribute Block of address 0001H become active.

Pressing the Bottom Switch

Attribute Block at address 0001H

| | Top | Middle | Bottom | | | |
|---------|---------|---------|---------|---------|--------|--------|
| Current | Switch | Switch | Switch | End | Timer2 | Timer1 |
| Address | Address | Address | Address | Address | | |
| 0001H | 0002H | 0006H | 0007H | 0000H | 00H | 00H |

If the bottom switch is pressed, the image at the address 0007H is displayed and the Attribute Block of the address 0007H becomes active.

Attribute Block at address 0007H

| | Top | Middle | Bottom | | | |
|---------|---------|---------|---------|---------|--------|--------|
| Current | Switch | Switch | Switch | End | | |
| Address | Address | Address | Address | Address | Timer2 | Timer1 |
| 0007H | 0001H | 0000H | 0000H | 000AH | 80H | 01H |

The timer is 128ms. If there is no switch activity for 128ms the timer expires causing the next image, 0008H, to be displayed. This continues until images 0009H and 000AH are also displayed. When the timer expires at address 000AH the address is equal to the End Address. The Middle Switch Address from the Attribute Block of 000AH is then checked.

Attribute Block at address 000AH

| | Top | Middle | Bottom | | | |
|---------|---------|---------|---------|---------|--------|--------|
| Current | Switch | Switch | Switch | End | | |
| Address | Address | Address | Address | Address | Timer2 | Timer1 |
| 000AH | 000CH | 0000H | 000AH | 0000H | 00H | 00H |

Since the Middle Switch Address is zero then the image at address 0007H is displayed and the Attribute Block of address 0007H becomes active.

If the top switch is pressed in any address from 0007H to 000AH then the image at address 0001H is displayed and the Attribute Block of address 0001H become active. Notice that the other attributes within the Attribute Block are ignored. Notice that the loop starting at address 0007H overlaps the loop starting at address 0006H and both end at the same End Address without interfering with each other.

Please note when the timer causes a new image to be displayed the Attribute Block of the beginning image stays active. All other Attribute Blocks within the loop are ignored except the Middle Switch Address of the End Address.

6. Communication Protocol

Communication initiated by Host

The controller communicates with the host via USB serial communication (57.6K, 1 start bit, 8 bit, 1 stop bit). 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, the controller reads one byte and executes the following:

- A. 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.
- B. If the byte is not a command, it is ignored.

When the controller executes a subroutine, and expects additional information:

- A. A timer is set. If the expected data byte is not received in 64ms, the controller transmits 6CH 6EH and terminates the routine.
- B. 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 proper format for all command options and data is specified for each command. See Section 7.

Communication initiated by the IS-Dev Kit-8

The Dev Kit transmits codes to host via USB based on activities stated below. Many of these reporting can be enable/disable by the flag setup.

- A. Top switch press is reported as 91H
- B. Top switch Release is reported as C1H
- C. Middle switch press is reported as 92H
- D. Middle switch Release is reported as C2H
- E. Bottom switch press is reported as 93H
- F. Bottom switch Release is reported as C3H
- G. Timer Expire is reported as 94H
- H. Upon changing the image, the image address is reported as FCH followed by the address in HEX so total of 3 bytes transmitted for each image change.
- I. If there is a communication problem between microcontroller and the microSD flash, the controller will transmit 6FH to host for each communication failure.

7. Commands

Command to reboot the controller

The command reboots the controller to power-up state.

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

The controller resets itself and transmits 11H when it is ready for operation. Please note there may be other bytes transmitted before 11H, depending on the state of the controller at the reset time.

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. This command does not affect the controller operation.

Command to display image at the requested address

This command displays an image at the requested address on the OLED display. The timer and operations are stopped.

command format: **2DH** 55H [address]
transmit format: (xxH) (xxH) (xxAH)

[address] is two bytes with value of 0001H to FFFFH sent as ASCII HEX format.

Example: 2D 51 30303445 displays the image stored at address 004EH. The attribute block of the address 004EH will be used.

Command to update the display

This command resends the picture from current address to the OLED Rocker. The controller transmits back 61H to the host.

command format: **2CH**
transmit format: (xxH)

Command to query controller for mode, controller, and firmware version

This command queries the controller for the mode as set by the Mode Select Switch, the controller name, and the firmware version installed.

command format: **26H** 52H 58H
transmit format: (xxH) (xxH) (xxH)

Example: The command is sent. The controller responds with the following:

| | | | | |
|-----|--------|-------------------|-----------|-----|
| 61H | [mode] | [controller name] | [version] | 79H |
| 61 | 33 | 43 4C 30 31 | 12 | 79 |

[mode] one byte. 33H indicating OLED switch mode or 34H indicating OLED display mode.

[controller name] 4 bytes. 43H 4CH 30H 31H (CL04)

[version] one byte. 11H (version 1.1)

Command to query controller for firmware version

This command queries the controller for the firmware version installed.

command format: **26H** 15H
 transmit format: (xxH) (xxH)

Example: The command is sent. The controller responds with the following:

| | | |
|-----|-----------|-----|
| 61H | [version] | 79H |
| 61 | 11 | 79 |

[version] one byte. 11H (version 1.1)

Command to disable the timer and switch execution

This command disables the timer and switch execution.

command format: **26H** 51H 5AH
 transmit format: (xxH) (xxH) (xxH)

The controller disables the timer and switch execution upon receiving this command. However, the switches are still scanned and reported. They are enabled upon reboot/power up or by command from host.

Command to enable the timer and switch execution

This command enables the timer and switch execution.

command format: **26H** 51H 65H
 transmit format: (xxH) (xxH) (xxH)

The controller enables the timer and switch execution upon receiving this command.

Command to turn the switch display off

This command turns off the OLED display.

command format: **26H** 51H 69H
transmit format: (xxH) (xxH) (xxH)

Command to turn the switch display on

This command turns on the OLED display.

command format: **26H** 51H 66H
transmit format: (xxH) (xxH) (xxH)

Command to turn the OLED voltage off

This command turns off the 16V supply to the OLED display.

command format: **26H** 51H 74H
transmit format: (xxH) (xxH) (xxH)

Command to turn the OLED voltage on

This command turns on the 16V supply to the OLED display.

command format: **26H** 51H 75H
transmit format: (xxH) (xxH) (xxH)

Note: This command should only be used when the OLED displays are in the off mode.

Commands that disable switch and timer execution

Upon transmitting any of the following commands, the timer stops running and attributes for the switches activity do not execute. The switches are still scanned and reported. The attributes execution is enabled upon reboot/power up or by command from host.

The reason for disabling attribute execution is for faster download of images and attribute as well as command to check the image at any address.

Command to download setup data

The command downloads the flags and other setup values. This information is stored on the MicroSD flash.

command format: **2AH** [00H] [00H] [55H] [flag] [0001H] [0001H] [reserved]
transmit format: (xxH) (xxAH) (xxAH) (xxAH) (xxAH) (xxAH) (xxAH) (xxAH)

[flag] is one byte sent in ASCII HEX format. (Detail below)

[reserved] is two bytes sent in ASCII HEX format. These bytes should be 00H. This is for future use. Customer could use the last byte for storing the data version.

All other byte must be transmitted with the stated values.

Flag Byte Default value=80H

| Bit | Enable Value | Flag | Controller action when flag is set |
|-----|--------------|----------------------------|------------------------------------------------------------------------------------------------------|
| B0 | 0 | Buzzer | Beeper sound for switch press |
| B1 | 0 | 61H flag | 61H is transmitted in response to command. |
| B2 | 0 | 79H flag | 79H is transmitted upon completion of command. |
| B3 | 0 | Switch release report flag | Switch release are reported |
| B4 | 0 | 6EH flag | 6CH/6EH is transmitted if there is any error during communication to host. |
| B5 | 0 | Timer expire report flag | Timer expiration is reported |
| B6 | 0 | Switch press report flag | Switch presses are reported |
| B7 | 0 | Address change report flag | Whenever a new image is displayed, the two bytes address, preceding by FCH, are transmitted via USB. |

Example: **2A30303030 3535 3030 30303031 30303031 30303130** sets all the flags ON and set the data version to 10. The flag takes affect when the controller is reset.

Command to upload the setup data

This command uploads the flags and other setup values to the host.

command format: **2BH** 0000H

transmit format: (xxH) (xxAH)

The controller transmits 10 bytes in HEX. The first two bytes are always 00H.

Example: The command is sent. The controller responds with the following:

```

61H [address]      [setup data]      79H
61    0000    55 00 00 01 00 01 00 10    79

```

[address] is two bytes with value of 0001H to FFFFH sent as ASCII HEX format.

[setup data] is ten bytes in HEX.

Command to download an image

This command downloads an image from the host to the EEPROM location.

command format: **28H** 55H [address] [image data]

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

[address] is two bytes with value of 0001H to FFFFH sent as ASCII HEX format.

[image] is 768 bytes transmitted in ASCII HEX format.

An image is handled as a block of 768 bytes. Each pixel requires one bit with eight pixels to a byte.

| Byte | Description |
|---------|----------------------|
| 1-12 | First line of image |
| 13-24 | Second line of image |
| • | |
| • | |
| • | |
| 745-756 | 63th line of image |
| 757-768 | 64th line of image |

Command to upload an image

This command uploads an image from the microSD location to the host.

command format: **29H** 55H [address]

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

[address] is two bytes with value of 0001H to FFFFH sent as ASCII HEX format.

The controller transmits back the image (768 bytes) in HEX format.

Command to download attribute block

This command downloads an attribute block from the host to the EEPROM location.

command format: **2AH** 55H [address] [attribute block]

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

[address] is two bytes with value of 0001H to FFFFH sent as ASCII HEX format.

[attribute block] is fourteen bytes transmit in ASCII HEX format.

The address and attribute block are saved at this address. When the controller reads the attribute block it compares the requested address to the saved address and if they do not match controller knows that the attribute block is not programmed and ignores it.

Image and Attribute block

Attribute Block:

An attribute is handled as a block of 14 bytes. Table below show the transmit order of the data.

| Byte | Description |
|------|------------------------------------------------|
| 1 | High byte of action address for Top Switch. |
| 2 | Low byte of action address for Top Switch. |
| 3 | High byte of action address for Middle Switch. |
| 4 | Low byte of action address for Middle Switch. |

| | |
|----|----------------------------------------------------------------------|
| 5 | High byte of action address for Bottom Switch. |
| 6 | Low byte of action address for Bottom Switch. |
| 7 | High byte of end address. |
| 8 | Low byte of end address. |
| 9 | Timer 1 |
| 10 | Timer 2 |
| 11 | High byte of action address for Left Switch. Reserved. Must be (00) |
| 12 | Low byte of action address for Left Switch. Reserved. Must be (00) |
| 13 | High byte of action address for Right Switch. Reserved. Must be (00) |
| 14 | Low byte of action address for Right Switch. Reserved. Must be (00) |

Command to upload an attribute block

This command uploads an attribute block from the EEPROM location to the host.

command format: **2BH** 55H [address]
 transmit format: (xxH) (xxH) (xxAH)

[address] is two bytes with value of 0001H to FFFFH sent as ASCII HEX format.

The controller transmits back the saved address and [Attribute block] in hex format (16 bytes).

Command to display image at the requested address

This command displays an image at the requested address on the OLED display. Stops timer and operation.

command format: **2EH** 51H [address]
 transmit format: (xxH) (xxH) (xxAH)

[address] is two bytes with value of 0001H to FFFFH sent as ASCII HEX format.

Example: 2E 51 30303445 displays the image stored at address 004EH.

Command to increment address and display it

This command increments the pointer to the next address and displays the image. Stops timer and operation.

command format: **2EH** 53H
 transmit format: (xxH) (xxH)

Example: Start at address 00A1H. Send 2E 53 and the OLED display displays the image stored at address 00A2H. Send 2E 53 again and the OLED display displays the image stored at address 00A3H.

Command to decrement address and display it

This command decrements the pointer to the next address and displays the image. Stops timer and operation.

command format: **2EH** 52H
transmit format: (xxH) (xxH)

Example: Start at address 1004H. Send 2E 52 and the OLED display displays the image stored at address 1003H. Send 2E 52 again and the OLED display displays the image stored at address 1002H.

Commands to download information directly to OLED Rocker

Note: Care must be taken using these commands. The OLED Rocker could get damaged if it is initialized with a value exceeding the specifications. Please check the OLED Rocker data sheet and application notes before using this command.

Command to download data directly to OLED Rocker

This command downloads data directly from the host to the OLED Rocker.

command format: **2FH** 55H [number] [data]
transmit format: (xxH) (xxH) (xxAH) (xxAH)

[number] is one byte sent in ASCII HEX format (02H to 80H). This is the number of data bytes (2 to 128 bytes).

[data] is 2 to 128 bytes sent in ASCII HEX format. The controller sends the first byte as command and the rest as data to OLED Rocker.

This command is provided for checking all the features of the OLED Rocker. It is possible to change the set up, exceed specifications and cause damage to the OLED Rocker. Please follow the specification when using this command.

8. Character Commands for OLED Rocker

The following commands allow changing of the images being displayed. Host can download ASCII characters to a specified location and the controller creates the graphic image in normal or inverse, using a 7x10 font look up tables.

Host can command the controller to turn ON/OFF one or more row of pixels.

Host can specify the OLED's brightness.

None of these commands affect the storage memory.

Command to specify the OLED brightness (OLED Color & OLED Rocker)

This command specifies the brightness for both OLED color and OLED Rocker.

| | | | |
|------------------|----------------|------------------|------------------|
| command | | [BRIGHTNESS OLED | [BRIGHTNESS OLED |
| format: | 27H 4EH | COLOR] | ROCKER] |
| transmit format: | (xxH) (xxH) | (xxAH) | (xxAH) |

[BRIGHTNESS OLED COLOR] is one byte send as ASCII HEX format. The acceptable values are from 00H to 0FH where 0FH is the brightest. Default (0FH).

[BRIGHTNESS OLED ROCKER] is one byte send as ASCII HEX format. The acceptable values are from 00H to 0FH where 0FH is the brightest. Default (0FH).

The specified brightness levels go to effect immediately and remain in effect for duration of the session.

Command to generate characters for OLED Rocker positive mode

This command generates 1 to 16 characters in font 8x10 (include the space between the character) from the specified coordinate in the OLED Rocker memory. Please note the characters can be generated on the memory outside the viewing area. The character pixels are white.

| | | | | | |
|------------------|----------------|-------------------|--------|----------|--------------|
| command format: | 27H 54H | [# of characters] | [Row] | [column] | [Characters] |
| transmit format: | (xxH) (xxH) | (xxAH) | (xxAH) | (xxAH) | [xxH] |

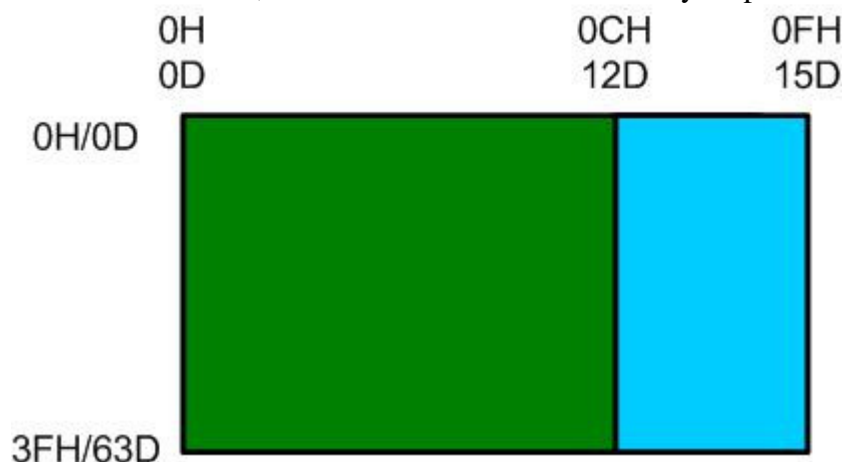
[# of characters] is one byte send as ASCII HEX format. The acceptable values are from 01H to 10H.

[Row] is one byte send as ASCII HEX format. It specifies the starting pixel row of the memory. The acceptable values are from 00H to 36H.

[Column] is one byte send as ASCII HEX format. It specifies the starting byte column in the memory. The acceptable values are from 00H to 0FH.

[Characters] 1 to 16 bytes of ASCII code for characters. The acceptable values for characters are from 20H to 7FH.

Illustration 2, OLED Rocker on-board memory map



Command to generator characters for OLED Rocker Negative mode

This command generates 1 to 16 characters in font 8x10(include the space between the character) from the specified coordinate in the OLED Rocker memory. Please note the characters can be generated on the memory outside the viewing area. The character pixels are black.

command format: **27H** 57H [# of characters] [Row] [column] [Characters]

transmit format: (xxH) (xxH) (xxAH) (xxAH) (xxAH) [xxH]

[# of characters] is one byte send as ASCII HEX format. The acceptable values are from 01H to 10H.

[Row] is one byte send as ASCII HEX format. It specifies the starting pixel row of the memory. The acceptable values are from 00H to 36H.

[Column] is one byte send as ASCII HEX format. It specifies the starting byte column in the memory. The acceptable values are from 00H to 0FH.

[Characters] 1 to 16 bytes of ASCII code for characters. The acceptable values for characters are from 20H to 7FH.

Command to set pixel row data to OFF for OLED Rocker

This command set all row pixel data to zero for 1 to 64 rows from the specified row in the OLED Rocker memory. Please note the entire row of the memory change including the outside the viewing area.

command format: **27H** 5CH [# of rows] [Row]

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

[# of Rows] is one byte send as ASCII HEX format. The acceptable values are from 01H to 40H.

[Row] is one byte send as ASCII HEX format. It specifies the starting pixel row of the memory. The acceptable values are from 00H to 3FH.

Command to set pixel row data to ON for OLED Rocker

This command set all row pixel data to one for 1 to 64 rows from the specified row in the OLED Rocker memory. Please note the entire row of the memory change including the outside the viewing area.

command format: **27H** 5DH [# of rows] [Row]

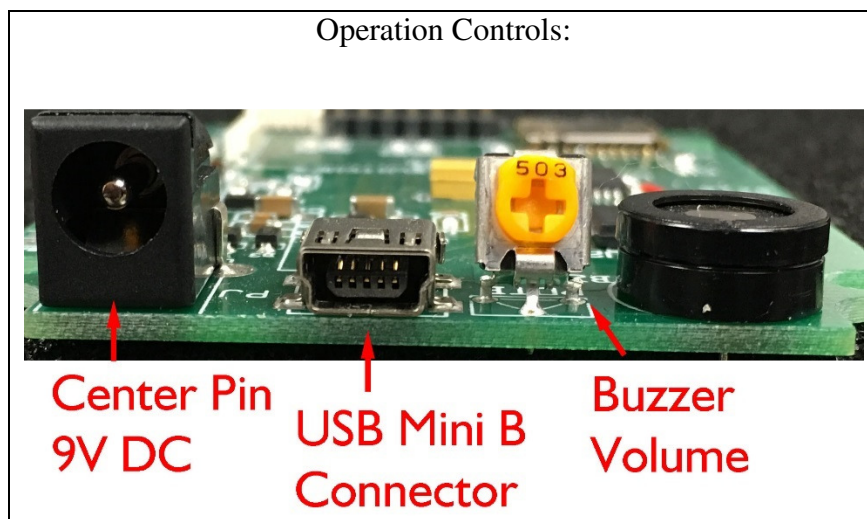
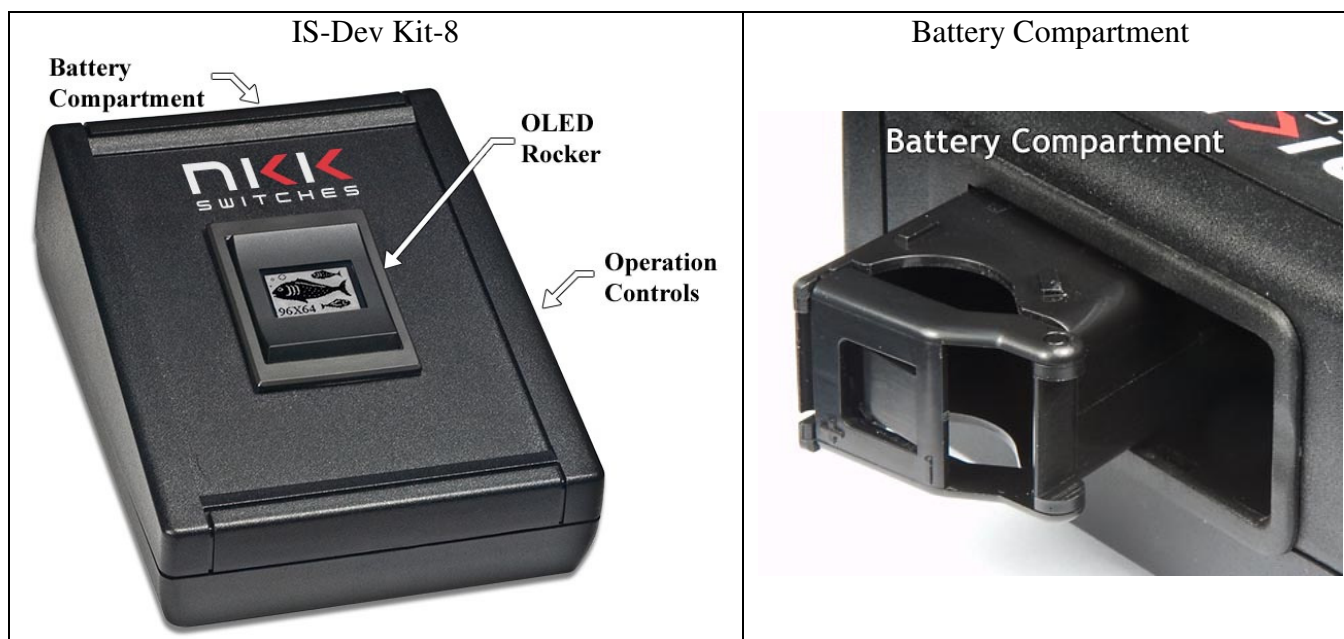
transmit format: (xxH) (xxH) (xxAH) (xxAH)

[# of Rows] is one byte send as ASCII HEX format. The acceptable values are from 01H to 40H.

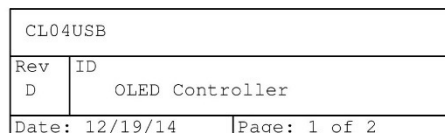
[Row] is one byte send as ASCII HEX format. It specifies the starting pixel row of the memory. The acceptable values are from 00H to 3FH.

9. Hardware

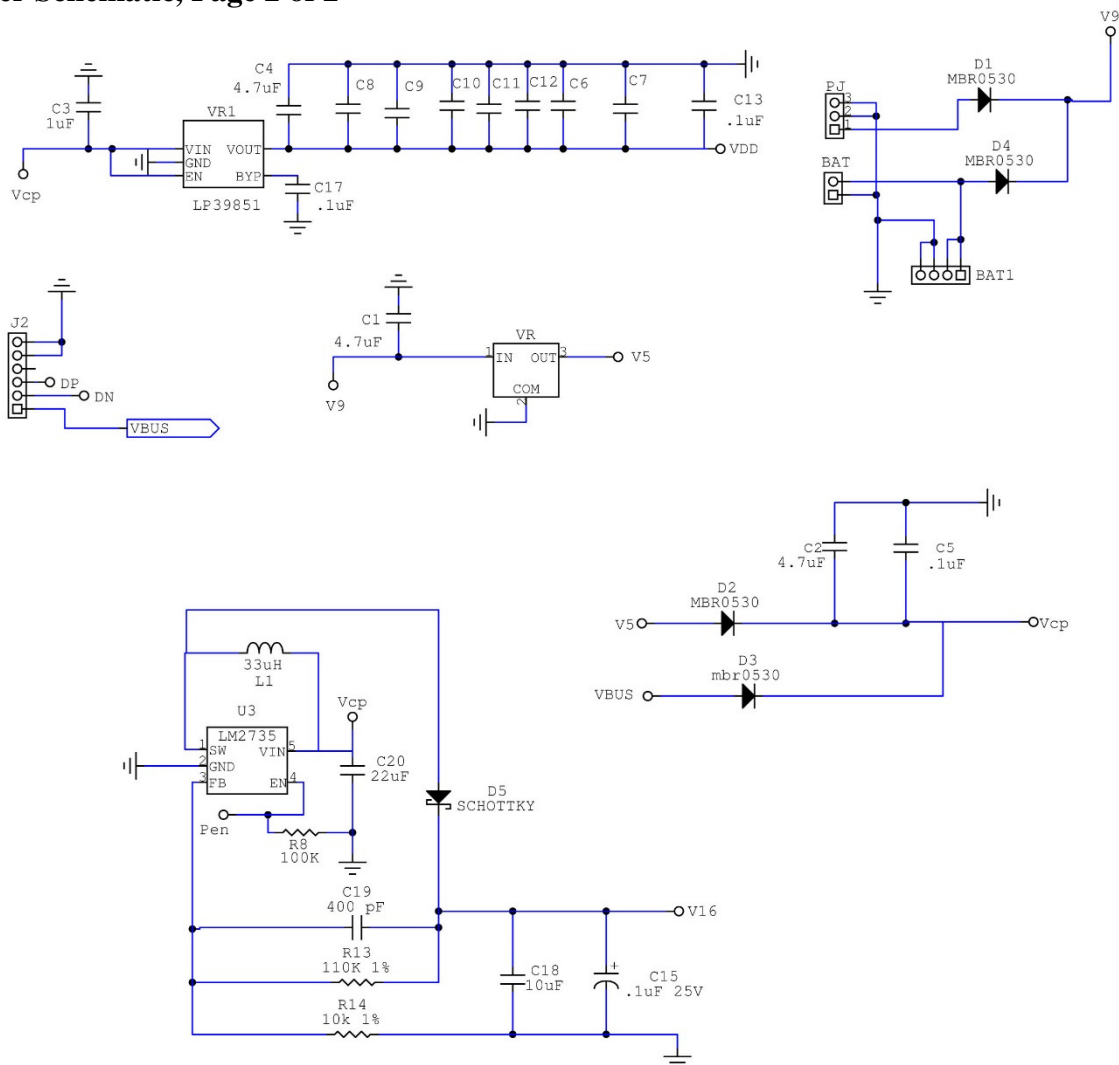
Controls Overview



The **OLED Rocker** has a monochromatic 96x64 OLED display and top, middle, and bottom buttons. The **Battery Compartment** is for an optional 9V battery. Battery must be removed to turn unit off. The **Operation Controls** are as follows:
Buzzer Volume adjusts the volume of the buzzer that activates when a button is pushed.
The **9V DC Power** jack mates with a 2.1mm cylinder power connector with a center positive.
The **USB Connector** links the controller to the host.

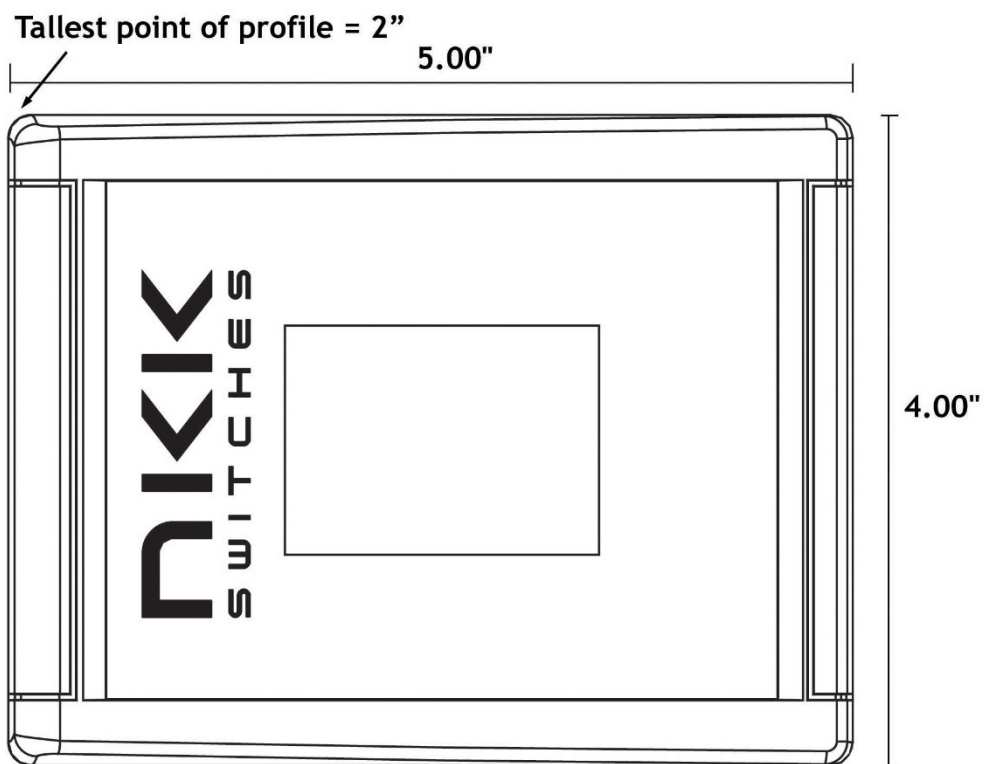


Controller Schematic, Page 2 of 2

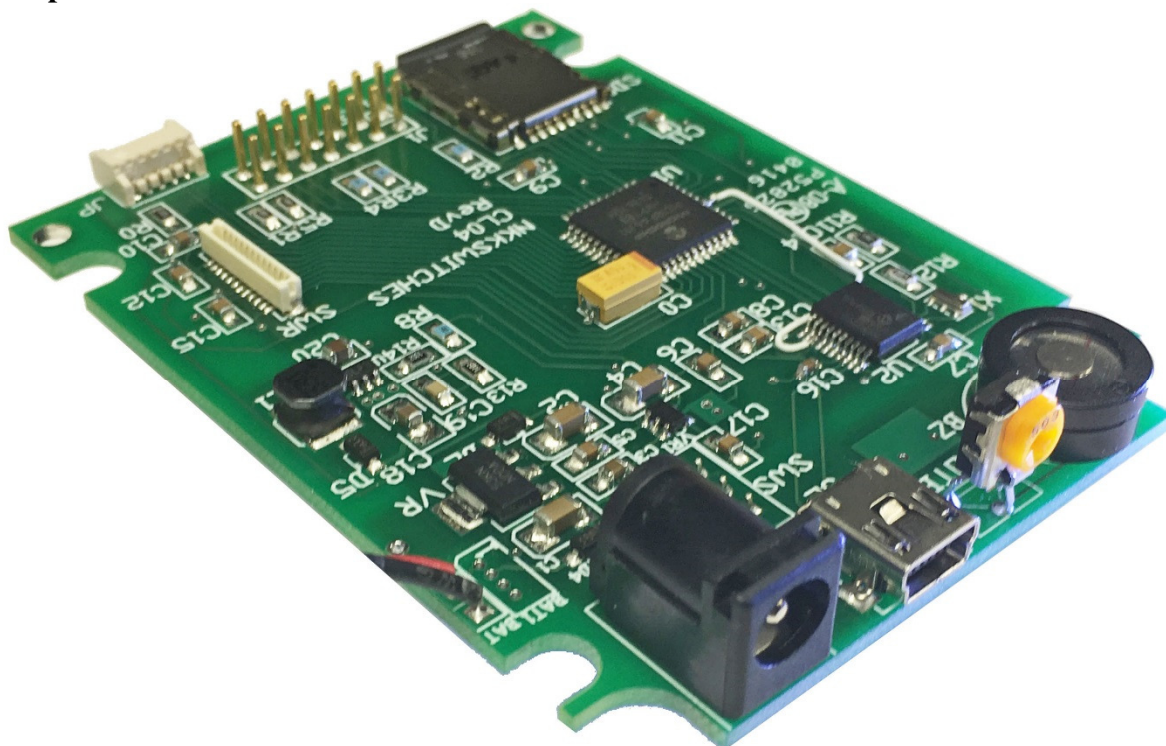


| | |
|----------------|-----------------|
| CL04USB | |
| Rev | ID |
| D | OLED Controller |
| Date: 12/19/14 | Page: 2 of 2 |

Container size



CL04 Board photo



10. Key Terms & Definitions

| | |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OLED module | NKK Switches' OLED SmartSwitches and SmartDisplays. |
| Host | Any computer, terminal, or other device that can communicate over the USB line. |
| Controller | A PCB assembly that controls one or more logic boards and the switches associated with them. It communicates with a host over the USB line. |
| Logic Board | A PCB assembly with “glue logic” for mounting switches. It is controlled by a controller. |
| Byte | 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. |
| Nibble/Hex digit | A four-bit value ranging from 0H to FH. A byte consists of two nibbles. |
| ASCII | A byte value representing a symbol. |
| Communication Format | <p>There are two formats to transmit a byte:</p> <ol style="list-style-type: none"> 1. Hex format - A hex byte is transmitted without any change to it. [xxH] will be used to denote this. <p>All commands and some data are sent by using this format.</p> <ol style="list-style-type: none"> 2. ASCII HEX format - Each nibble of the byte is converted to ASCII code and sent as a byte. [xxAH] will be used to denote this. <p>For example, the hex byte 5AH is transmitted in two bytes, 35H and 41H. The ASCII value for 5 is 35H and the ASCII value for A is 41H.</p> <p>All addresses and most data are sent using this format.</p> |
| Address | A two-byte value ranging from 0001H to FFFFH representing the 65536 memory locations for pictures and attributes on microSD flash. |

11. Firmware issues

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

Version V2.6

Xon/Xoff was implemented for receive communication. Controller transmits 0x11 after power up indicating it is ready for communication. The controller transmits 0x13 when the buffer is $\frac{3}{4}$ full and transit 0x11 when the buffer is $\frac{1}{4}$ full.

Character generation commands have been added.

Version V2.4

During batch downloads, due to the different response times of the MicroSD, sometimes the communication buffer gets filled/overwrite so some images would be corrupted.

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:

1. 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

2. 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 = 10101 base-2 or binary

The base-2 number 10101 reads as one zero one 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