

LCD36x24 Engineering Kits User Manual

Revision B

Firmware Version 1.X



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. All Engineering Kits are tested 100% and programmed with the default images and attributes. As the firmware for these kits can be modified by the customer, any damage caused by a customer mistake is not warranted.

Table of Contents

1.General Features	3
2.Available LCD36x24 SmartSwitches	4
3.Possible Engineering Kits Using the S0205 Controller	4
4.Firmware Overview	5
5.Operational Overview	8
6.Communication Protocol	9
7.Commands to the Controller	10
8.Hardware Considerations	14
9.Programming Setup	14
10.Modifying the Firmware	15
11.Board Dimensions.....	16
12.Key Terms & Definitions.....	17
13.Schematics	17

1. General Features

The Engineering Kits (IS-ENG-KIT-x) are designed to accelerate the incorporation of the SmartSwitch family of products into real world applications by, not only familiarizing design engineers to the SmartSwitch products, but also as a development platform for experimentation and design viability testing.

The controller used for the LCD36x24 SmartSwitches is the S0205. The schematic is at the end of this document. The C language source codes and a Windows based communication software, Engineering Kits Communicator, can be downloaded from www.nkswitches.com. The software is designed to download images, attributes, and manual commands as well as display responses from the kit.

Features of the S0205:

- Controls up to two LCD36x24 SmartSwitches.
- 1M on-board NOR flash memory to store images, attributes, or any other information.
- Serial communication via USB (115.2K, 1 start bit, 8 bit, 1 stop bit).
- Auxiliary port with 7 I/O pins of microcontroller to control/sense other user-defined applications.
- Stand-alone or real-time operation.
- Preprogrammed for basic operation.
- Reports via USB the switch closures and releases, timer expirations, and addresses of the images displayed.
- Programmable for changing the displayed images based on switch actuations and user defined timers.
- Schematics and firmware are provided for user firmware customization.
- User can program the Engineering Kits using a PICKIT3 debugger and pin adaptor.
- The kits come with a 6-foot USB 2.0 A to Mini-B cable (IS-USB1).
- Power Specs: 4.5V to 5.5V.
- Maximum current: 100mA@5VDC.
- Can be powered either by USB or by optional 5V power connection (.100" spacing).
- Window based software is provided for communication.
 - Accepts bitmap files, then extracts the images and downloads them to the controller.
 - Allows typing of commands and downloading to the controller.
 - Extracts HEX or ASCII data from Excel files and downloads to the controller.
 - Messages to and from the controller are displayed in different colors.

2. Available LCD36x24 SmartSwitches

There are 8 products in LCD36x24 family, two SmartDisplays, several compact size SmartSwitches and several standard size SmartSwitches. The backlighting options are RGB, Red/Green, and Yellow/Green.

Part Number	Description	Socket
IS01BBFRGB	Display RGB backlight	5x2; 2mmx2mm
IS01BBFEF	Display Yellow/Green backlight	5x2; 2mmx2mm
IS15BSBFP4RGB	Compact size switch RGB backlight	AT970465F
IS15BSAFP4CF	Compact size switch Red/Green backlight	AT970465F
IS15BSBFP4EF	Compact size switch Yellow/Green backlight	AT970465F
IS15BBFP4RGB	Standard size switch RGB backlight	AT970465E
IS15BAFP4CF	Standard size switch Red/Green backlight	AT970465E

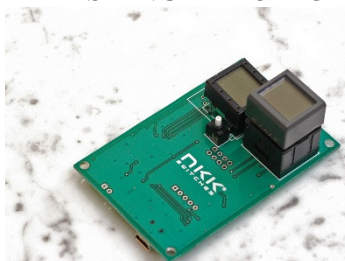
3. Possible Engineering Kits Using the S0205 Controller

Many Engineering Kits can be made with the S0205 controller simply by soldering on various sockets (Table below). Only the Engineering Kit options marked in green are marketed and kept in stock. However, all of them can be ordered.

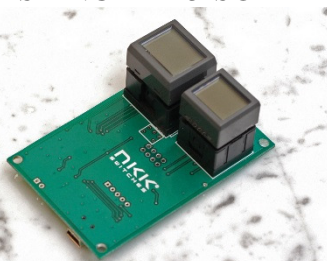
All the Engineering Kits come with switches/display on the sockets and a USB cable. Controlling all the LCD36x24 kits is the same.

Part Number	Switch 1 / Display 1		Switch 2	
IS-ENG-KIT-5-DC	IS01BBFRGB	Display LCD 36x24 RGB	IS15BSBFP4RGB	Compact LCD 36x24 RGB
IS-ENG-KIT-5-SC	IS15BBFP4RGB	Standard LCD 36x24 RGB	IS15BSBFP4RGB	Compact LCD 36x24 RGB
IS-ENG-KIT-5-BC	IS15BAFP4CF	Standard LCD 64x32 Red/Green	IS15BSAFP4CF	Compact LCD 36x24 Red/Green
IS-ENG-KIT-5-EF	IS01BBFCFEF	DISPLAY LCD36X24 Yellow/Green	IS15BSBFP4EF	Compact LCD36x24 Yellow/Green

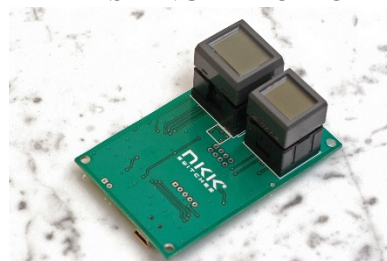
IS-ENG-KIT-5-DC



IS-ENG-KIT-5-SC



IS-ENG-KIT-5-BC



4. Firmware Overview

The firmware is written such that it is easy to modify or add features. Only basic features are implemented in the factory default to keep the firmware easy to follow. As such it is not optimized.

There is no difference in controlling the various LCD36x24 switches/displays. The displayed LCD36x24 image must be refreshed. If the LCD is not refreshed while the LCD voltage is present, the LCD can get damaged. The images are refreshed from RAM in the timer interrupt. One row of pixels is sent to the switches in each interrupt.

The firmware communicates to a host computer via RS232 through USB. Any data sent from the host are received via interrupt and placed in a receive buffer. The data get processed in the main program. Any transmit datum is put in a transmit buffer. Each timer interrupt transmits a byte from the transmit buffer if it is not empty. The firmware also handles read and write operations from and to the external flash memory, as well as controls the two LCD switches. The communication protocol is outlined in following sections.

The 1M flash memory is a NOR flash and as such, when writing to the memory, it can only turn an ON cell to the OFF state. The erase command turns all the cells to the ON state. If images or other data on the flash memory need to be changed then it is required that the erase command be sent first. The flash memory has partial erase commands available, however, as of the first firmware release date only the complete erase is implemented in the firmware.

The 7 I/O pins in the auxiliary port are initialized as inputs with weak pull-ups. These pins can be used to sense or control in user-defined custom applications.

The flash memory is set up to save up to 2047 images and attributes. Two bytes are needed to address an image or attribute block.

Address	LCD64x32	
0001	Image	Attribute Block
0002	Image	Attribute Block
---	Image	Attribute Block
---	Image	Attribute Block
07FE	Image	Attribute Block
07FF	Image	Attribute Block

Each image is 120 bytes and each Attribute Block is 16 bytes long.

Attribute Block										
Name	Current address	End address for the loop	Timer1 for the loop	Timer2 for the loop	Next address for SW1 upon sw press	Next address for SW2 upon sw press	Jump address for SW1 at the end of the loop	Jump address for SW2 at the end of the loop	LED color code for ON-cycle	LED color code for OFF-cycle
Range	0000 to 07FF	0000 to 07FF	00 to FF	00 to FF	0000 to 07FF	0000 to 07FF	0000 to 07FF	0000 to 07FF	LED code	LED code
# of bytes	2	2	1	1	2	2	2	2	1	1

Each switch has an associated active Attribute Block in the RAM. Additionally, each switch is associated to an active display address and active backlight codes. If the system detects an image needs changing, the image from the flash memory address of the active address is transferred to the RAM. The RAM data automatically get refreshed to the display via timer interrupt. If the system detects that the backlight timer has expired, the active LED codes are used to turn on/off the LED backlights according to the ON/OFF cycle.

An address is assigned to a switch via switch press, timer expiration, or upon power-up. Before an address is assigned to a switch, the controller checks the Attribute Block in the flash at the assigned address. If the first two bytes are equal to the assigned address, the Attribute Block from flash is loaded to the active Attribute Block. After loading, the active LED code and active display are loaded based on the attributes. If Timer1 does not equal zero, the loop timer starts running. If the first two bytes of the attribute block are not equal to the assigned address, no action is taken.

If Timer1 does not equal zero, a loop timer is started. The loop timer is calculated by multiplying Timer1 and Timer2. When the loop timer expires the displayed address is compared to the End Address. If they are equal, and the jump addresses for SW1/SW2 do not equal zero, then the jump addresses become the new active addresses for switch1 and switch2. Then the attributes and images are loaded accordingly. If the jump addresses of SW1/SW2 are equal to zero, there is no change to the active addresses and the loop restarts for the associated switch. If the displayed address and the End Address are not equal, then the displayed address + 1 is loaded as the displayed address and the image is sent to the switch. The loop timer then restarts. When a new displayed address is loaded, the backlight LED codes for that address are sent to the switch.

When a switch is pressed, if the press addresses for SW1/SW2 of the active Attribute Block are not equal to zero they become the active addresses for switch1 and switch2, and are sent to the switches. If the press addresses of SW1/SW2 are equal to zero, there will not be any update or change to the active addresses.

The backlight code is designed for future expansion. Even though at this time each backlight has two options of ON/OFF, the firmware can be written to have 4 options for each color. The current firmware turns each LED

color ON if any of the two bits are ON. However, it is recommended to put both bits to 1 for future compatibility.

Backlight LED Code								
RGB switch/display	RED		GREEN		BLUE		Must be 1	
RGB Compact	RED		GREEN		BLUE		Must be 1	
Red/Green Switch	NA		RED		GREEN		Must be 1	
Red/Green Compact	GREEN		RED		NA		Must be 1	
Yellow/ Green Compact	YELLOW		GREEN		NA		Must be 1	
Yellow/ Green Display	YELLOW		GREEN		NA		Must be 1	
Bits	B7	B6	B5	B4	B3	B2	B1	B0
Possible values	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	0 or 1	1	1

The controller alternates between the ON and OFF codes every second.

The LCD voltage is generated using PWM. Eight options for adjustment are provided (0 to 7). The default for PWM is set for level 4. A lower PWM level will produce a lower voltage. This value can be changed permanently in the setup command or for the session via host command.

5. Operational Overview

Power-up Sequence:

Upon power-up the controller executes the following 8 steps and then enters the main program.

1. Initializes the active attribute blocks for both switches to zero.
2. Put 0001H for active address of switch#1, put 0002H for active address of switch#2, initialize all the active backlight codes to white, and initialize the contrast value to level 4.
3. Check the setup values from flash memory and if valid, change the active address for switch#2 and the contrast value.
4. Check the Attribute Blocks from the flash memory and if valid, load them to the active Attribute Blocks in the RAM.
5. Set the pulse width modulator according to the contrast level.
6. Set the timer-interrupt for every 0.5ms.
7. Set up the UART (RS232) communication.
8. Transmit 11H to host.

Main Program:

The main program continuously goes through the following steps:

1. Check and execute commands if there is data in the RS232 receive buffer.
2. If the time for switch scan is expired: Scan the switches for being pressed.
3. If switch #1 timer has expired: The timer is stopped, then report 83H and process according to the active attribute of switch#1.
4. If switch #2 timer has expired: The timer is stopped, then report 84H and process according to the active attribute of switch#2.
5. If switch#1 is pressed: Report 81H and process according to the active attribute of the switch#1.
6. If switch#2 is pressed: Report 82H and process according to the active attribute of the switch#2.
7. If switch#1 is released: Report C1H.
8. If switch#2 is released: Report C2H.
9. If switch#1 image changes: Report FFH followed by the switch#1 active address and update the memory for switch#1 with the active address image from the flash for switch#1. Start timer for switch#1 if Timer1 is non-zero.
10. If switch#2 image changes: Report FEH followed by the switch#2 active address and update the memory for switch#2 with the active address image from the flash for switch#2. Start timer for switch#2 if Timer1 is non-zero.
11. If the backlight timer has expired: The switch#1 backlight is updated with the active backlight code for switch#1 and according to the ON/OFF cycle and the switch#2 backlight is updated with the active backlight code for switch#2 and according to the ON/OFF cycle.
12. Go to step1.

Timer-Interrupt:

Timer-interrupt is set for every 0.5ms. It does the following functions:

1. If the line counter is 0, set FLM signal high.
2. Send one row of pixel data according to the line counter.
3. Set the LP signal high and then low.
4. Set the FLM signal low.
5. Increment the display line counter. If line counter > 23, set to 0.
6. Increment the master half millisecond timer that the main program uses for all timers. All timers use this for a starting time and to check if the stop time has passed.
7. If the RS232 transmit buffer is not empty, transmit a byte.
8. Exit the interrupt.

6. Communication Protocol

The controller communicates with the host via USB serial communication (115.2K, 1 start bit, 8 bit, 1 stop bit).

Communication Initiated by the Controller

The controller transmits the switch activities, timer expirations, and the addresses of images when they are displayed. The protocol for the code transmitted are explained in the Main Program in the Operational Overview.

Communication Initiated by the Host

The controller receives the data via a serial interrupt routine that places the data in the circular receive buffer. When the controller detects data in the circular receive buffer in the main program, the controller reads one byte and executes according to the following scenario.

- A. If the byte is 01H the controller responds by putting 61H in the transmit buffer and exits.
- B. If the byte is a 20H to 2FH the controller responds by putting 61H in the transmit buffer. The controller checks the command procedure and processes it accordingly. If the command procedure exists and all the data is proper, the controller puts 79H in the transmit buffer and exits. If the command procedure does not exist, the data is not acceptable, or consecutive bytes are not received within 100ms then the controller puts 6EH in the transmit buffer and exits.
- C. If the byte is not 01H, or 20H to 2FH, it is ignored.

7. Commands to the Controller

Command to reboot the controller

This command reboots the controller to power-up state and upon completion reports 11H:

Command format: **24H**
 Transmit format: (xxH)

Command to check communication

This command is used to check if the controller is on-line:

Command format: **01H**
 Transmit format: (xxH)

The controller transmits 61H back to the host.

Command to download LCD contrast level

This command downloads the LCD contrast level for both switches:

Command format: **27H** 4FH [Contrast level]
 Transmit format: (xxH) (xxH) (xxAH)

[Contrast level] is one byte transmitted in ASCII HEX. The acceptable range is 00H to 07H, where 07H is the maximum contrast level.

Command to query the 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:

61	51	53 30 32 30 35	31 30	79
61H	[switch code]	[Controller name]	[Version]	79H

[switch code] is one byte. 51H indicates the LCD36x24 module.

[Controller name] is five bytes. 53H 30H 32H 30H 35H (S0205)

[Version] is two bytes. For example, 31h 30H would be version 1.0 (converted from ASCII hex)

Command to manually set images for switch#1 and switch#2

Command format: **2DH** [Address for switch#1] [Address for switch#2]
 Transmit format: (xxH) (xxAH) (xxAH)

[Address for switch#1] is two bytes transmitted in ASCII HEX. The acceptable range is 0001H to 07FFH.
 [Address for switch#2] is two bytes transmitted in ASCII HEX. The acceptable range is 0001H to 07FFH.

The controller will assign the desired addresses to the switches.

Command to download backlight codes for switch#1 and switch#2

Command format:	2CH	[ON cycle coded for switch#1]	[OFF cycle coded for switch#1]	[ON cycle coded for switch#2]	[OFF cycle coded for switch#2]
Transmit format:	(xxH)	(xxAH)	(xxAH)	(xxAH)	(xxAH)
Number of bytes	1	1	1	1	1

[ON cycle for switch#1] is one byte transmitted in ASCII HEX. The code according to the Firmware Overview section.

[OFF cycle for switch#1] is one byte transmitted in ASCII HEX. The code according to the Firmware Overview section.

[ON cycle for switch#2] is one byte transmitted in ASCII HEX. The code according to the Firmware Overview section.

[OFF cycle for switch#2] is one byte transmitted in ASCII HEX. The code according to the Firmware Overview section.

The controller puts the desired backlight codes as the active backlight codes.

Commands that Disable Switch and Timer Execution

Upon transmitting any of the following commands, the timers stop running and attributes for the switches activity do not execute. However, the switches are still scanned and reported. The attributes execution is enabled upon reboot or power-up.

The reason for disabling the attribute execution is to enable faster downloads of images and attributes.

Command to disable switch and timer execution

This command disable switch and timer execution

Command format: **26H** 51H 5AH
 Transmit format: (xxH) (xxH) (xxH)

Command to erase the flash memory

This command erases the memory by turning all the cells ON. Before proceeding, the host needs to wait for the 79H to be received from the controller, indicating the memory is erased. The erase time for this flash memory is less than a second.

Command format: **21H** 55H AAH 52H 52H
 Transmit format: (xxH) (xxH) (xxH) (xxH) (xxH)

Command to download setup data

This command downloads the setup data. The setup data is saved on the flash memory at address 0000H:

	2AH	51H	[0000H]	[Switch 2 start address]	[Contrast Level]	[Reserved]
Transmit format:	(xxH)	(xxH)	(xxAH)	(xxAH)	(xxAH)	(xxAH)
Number of bytes	1	1	2	2	1	11

[Contrast Level] is one byte sent in ASCII HEX format. Range 00H to 07H.

[Switch 2 start address] is two bytes with the value of 0001H to 07FFH sent in ASCII HEX format. This is the address for the image that switch#2 displays upon power-up or reboot.

[Reserved] is 11 bytes of 00H sent in ASCII HEX format. This is for future use. The customer can configure the last two bytes as needed for their application. The rest of the reserved bytes should be zero. The controller saves the setup to flash memory.

This setup data can be read using the attribute upload command.

On startup the controller checks if the setup attribute is valid, and if not, the setup is not used.

Command to download an image

This command downloads an image from the host to the flash memory address:

Command format: **28H** 51H [Address] [Image]
 Transmit format: (xxH) (xxH) (xxAH) (xxAH)

[Address] is two bytes with value of 0001H to 07FFH sent in ASCII HEX format.

[Image] is 120 bytes transmitted in ASCII HEX format.

Byte	Description
1	Bit 7 is top left pixel of the display
5	Bit 4 is top right pixel of the display
•	
•	
•	
116	Bit 7 is bottom left pixel of the display
120	Bit 4 is bottom right pixel of the display

Command to upload an image

This command uploads an image from the flash memory location to the host:

Command format: **29H** 51H [Address]

Transmit format: (xxH) (xxH) (xxAH)

[Address] is two bytes with value of 0001H to 07FFH sent in ASCII HEX format.

The controller transmits the image, 120 bytes, back in ASCII HEX format.

Command to download Attribute Block

This command downloads an Attribute Block from the host to the flash memory location:

Command format: **2AH** 51H [Attribute Block]

Transmit format: (xxH) (xxH) (xxAH)

[Attribute Block] is 16 bytes transmitted in ASCII HEX format.

The first two bytes of the Attribute Block are the desired address of the block. The Attribute Block is stored in flash memory according to these first two bytes.

When the controller reads the Attribute Block, it compares the desired address to the saved address. If they do not match, all attributes are deemed invalid and do not get loaded.

Command to upload an Attribute Block

This command uploads an Attribute Block from the flash memory location to the host:

Command format: **2BH** 51H [Address]

Transmit format: (xxH) (xxH) (xxAH)

[Address] is two bytes with the value of 0000H to 07FFH sent in ASCII HEX format.

The controller transmits the 16 bytes Attribute Block via RS232 in ASCII HEX format.

8. Hardware Considerations

The power can be supplied to the Engineering Kits via USB port or the J2 header. The voltage range can be 4.5V to 5.5V. Maximum current consumption is 100mA. A fuse and a Zener diode are used for high voltage input protection.

The PIC18F46K40 microcontroller is used in the Engineering Kits. The VDD for the microcontroller is 3.3V. The LED backlights are controlled by a constant-current controller. The current for each LED is less than 10mA. The bicolor switch LED forward current recommendation is 15mA. To achieve 15mA, the R5 value needs to be changed from 2k to 1.35k.

The LCD voltage is designed so it can be adjusted via PWM. For normal operation, the VLC does not have to be changed. Only an application with an extreme operating temperature variation will need to adjust the LCD voltage.

The auxiliary ports I/O are directly connected to microcontroller pins without any safety circuits. Care must be taken not to exceed the VDD voltage.

9. Programming Setup

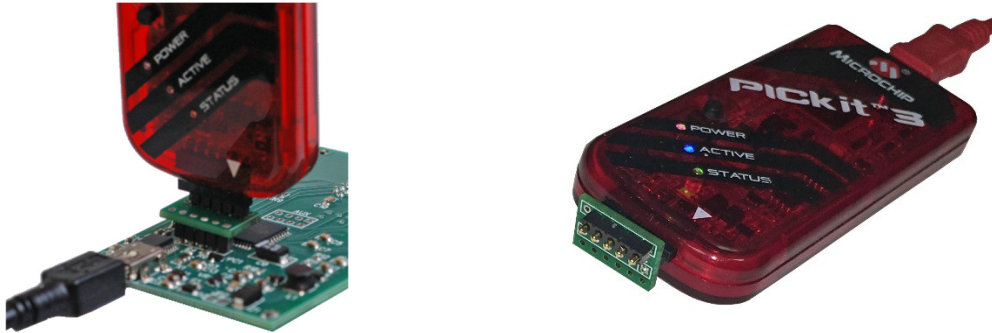
The Engineering Kits are designed to be easily programmed and have a 5-position SIP footprint for this purpose (JP1). They can be programmed by a variety of different programmers. Our recommendation is to use the PICKit 3 with the IS-PA2 adaptor and a USB power source. The IS-PA2 is an adaptor with a 5-position SIP header on one side and a 5-position, spring-loaded header on the other side.

Note: Position 1 of the PICit3 must be at position 1 on the PCB or damage could occur. The 5V power must be supplied; usually through the USB cable although there are through-hole pads on the PCB for an alternative 5V power connection.

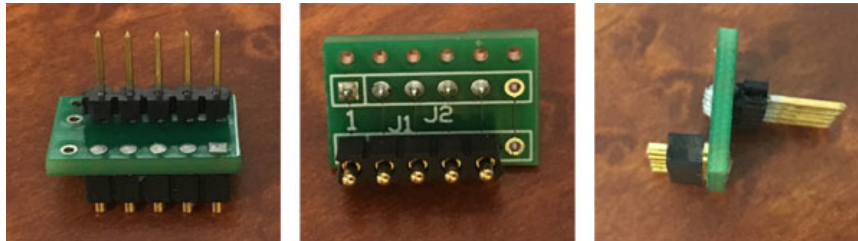
Connector callout of the IS-PA2 and controller JP1.

J1	J2	Controller JP1	
1	1	1	MCLR
2	2	2	VDD
3	3	3	GROUND
4	4	4	PGD (data)
5	5	5	PGC (clock)

IS-PA2 on a control board:



IS-PA2 Board photos:

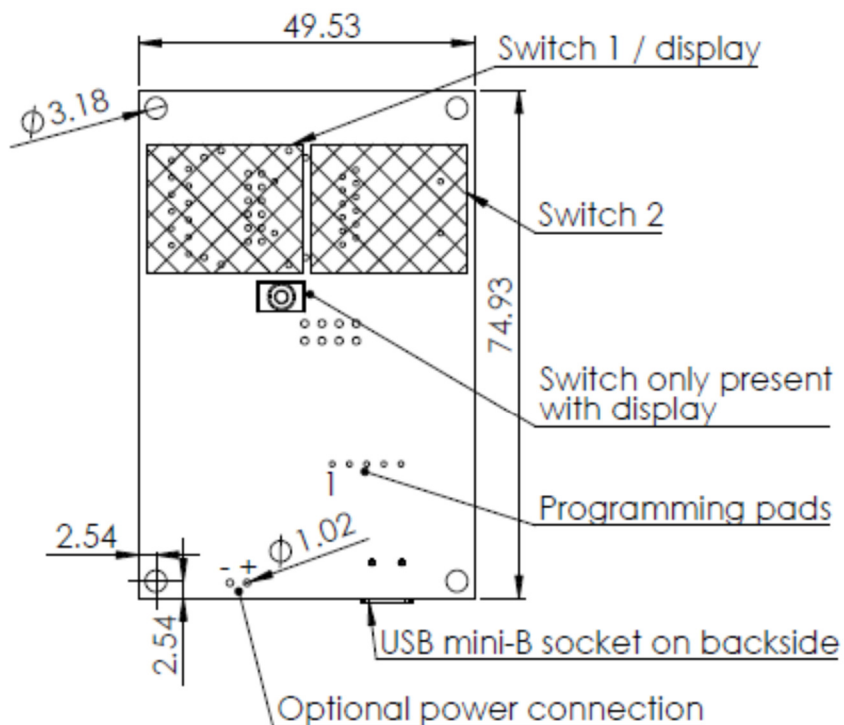


10. Modifying the Firmware

1. Download MPLAB-X from www.microchip.com.
2. Install MPLAB-X to the default location with the default options.
3. Download the MPLAB XC8 compiler from www.microchip.com.
4. Install the MPLAB XC8 compiler to the default location with the default options.
5. Extract the firmware source code to a location of your choice.
6. Open MPLAB-X IDE (Integrated Development Environment).
7. Press the open project button and navigate to the directory the source code was extracted to, then click the directory and click “Open Project”.
8. Click the “Production” menu at the top, and select “Clean and Build Project”. Wait for the build to complete.
9. Attach a PICKIT3 (or equivalent) to the PCB board. This can be done using a IS-PA2 adapter (available for purchase on the NKK website) or by soldering a header into JP1 and attaching the PICKIT3.
10. Press the debug project button and wait for the output window to say “Running”.
11. Observe the program is now running on the board and images appear on the SmartSwitch.
12. Press the stop button. You are now ready to start developing for the SmartSwitches.

11. Board Dimensions

Typical dimensions. Sockets and components vary with kit type.



Dimensions in mm.

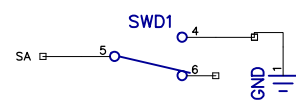
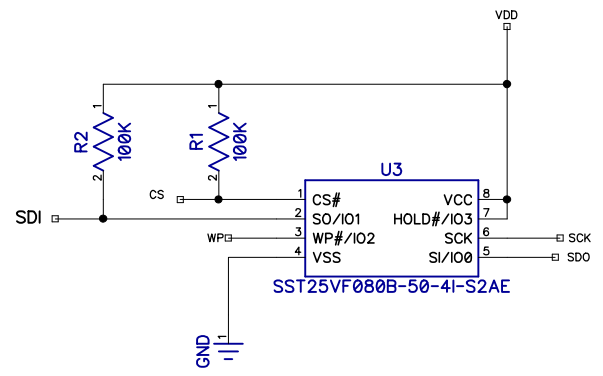
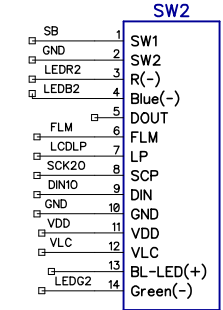
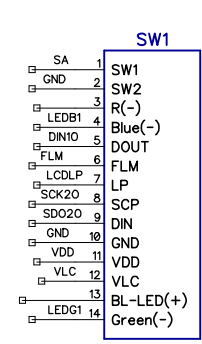
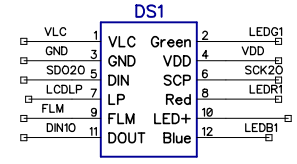
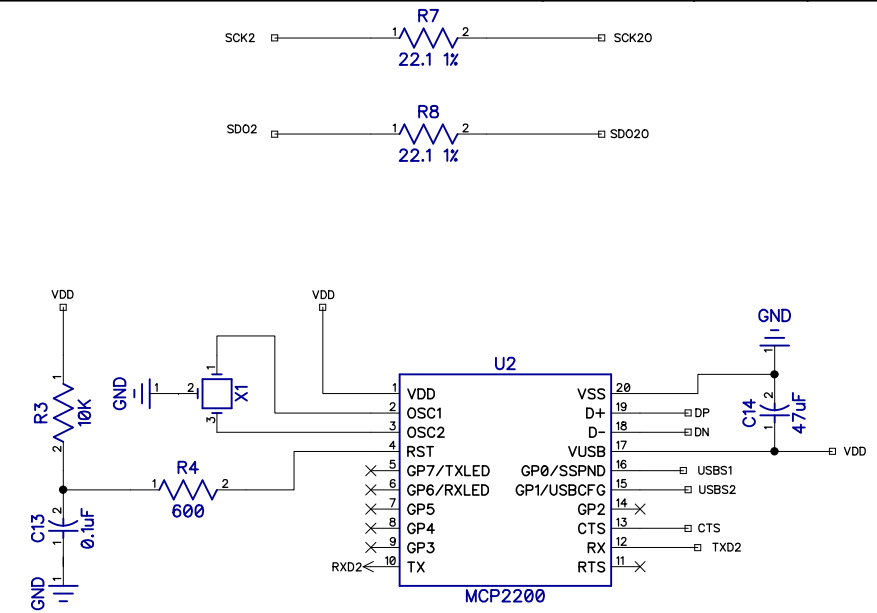
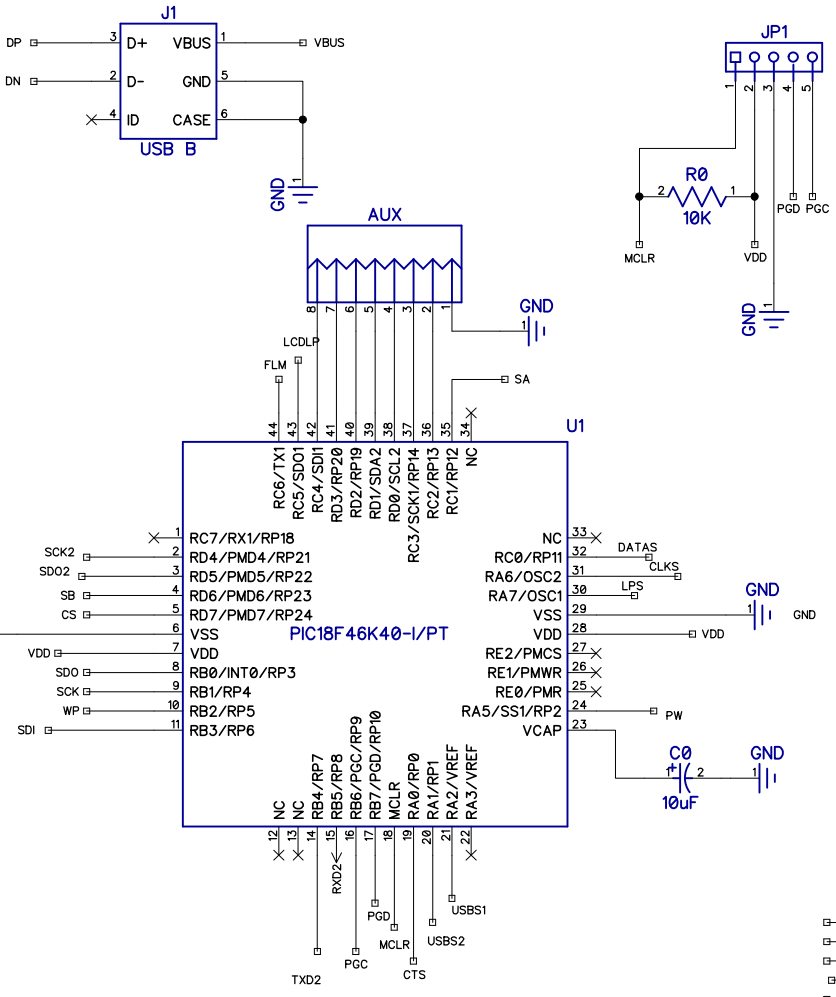
12. Key Terms & Definitions

Host	Any computer, terminal, or other device that can communicate over the USB line.
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. All commands and some data are sent by using this format.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. 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. All addresses and most data are sent using this format.
Address	A two-byte value ranging from 0001H to 07FFH representing the 2,047 memory locations for pictures and attributes on the flash memory.

13. Schematics

See schematics on next page(s).

A
B
C
D
E
F

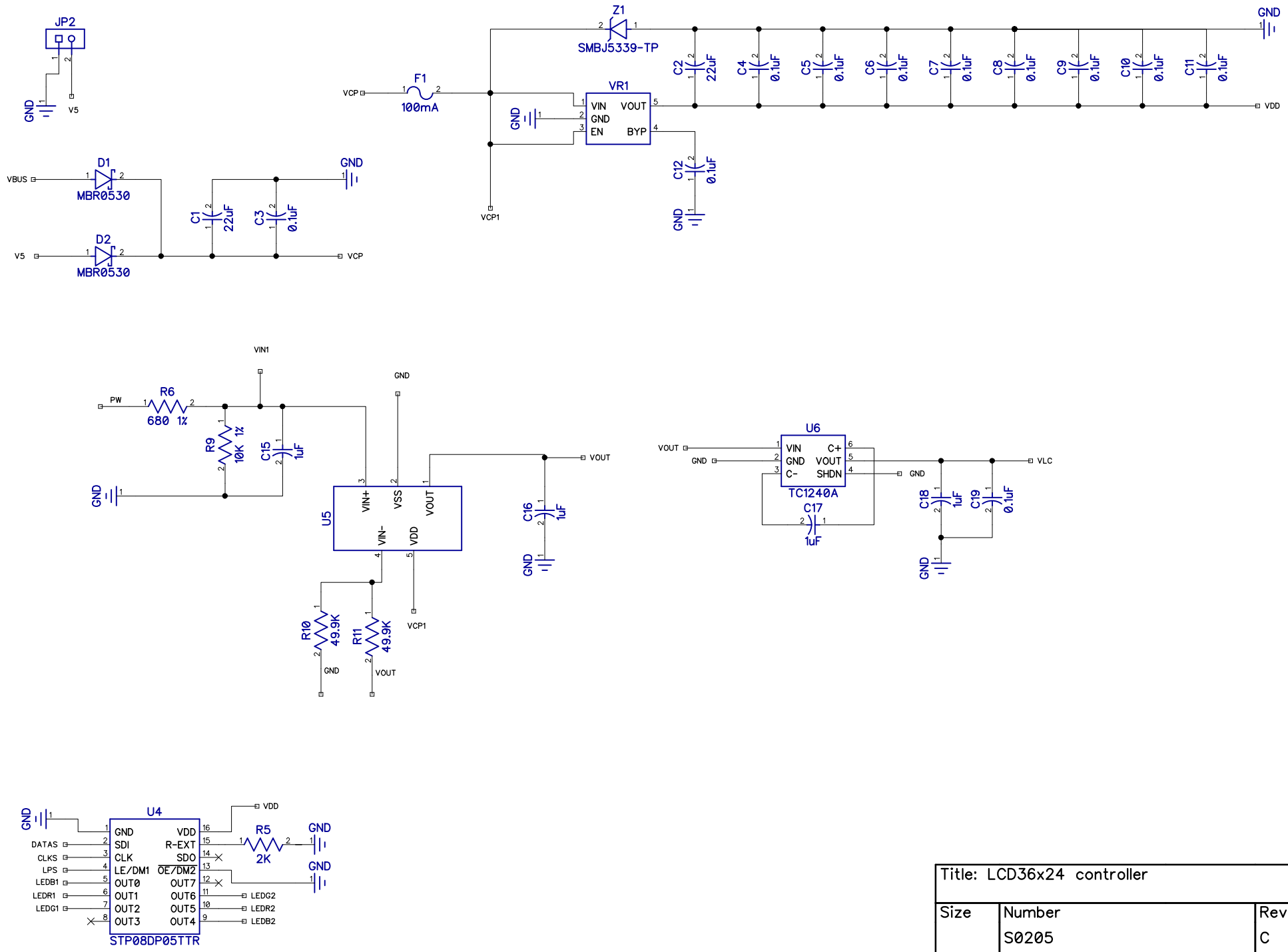


Title: LCD36x24 Controller		
Size	Number	Rev
	S0205	C
Date	8/25/2017	Drawn by: Hassan
Filename		Sheet 1/2

1 2 3 4 5 6 7 8

A
B
C
D
E
F

A
B
C
D
E
F



Title: LCD36x24 controller		
Size	Number	Rev
	S0205	C
Date	Drawn by	
Filename	Sheet 2/2	