IS-C32G1 Intelligent Controller User Manual

Revision A

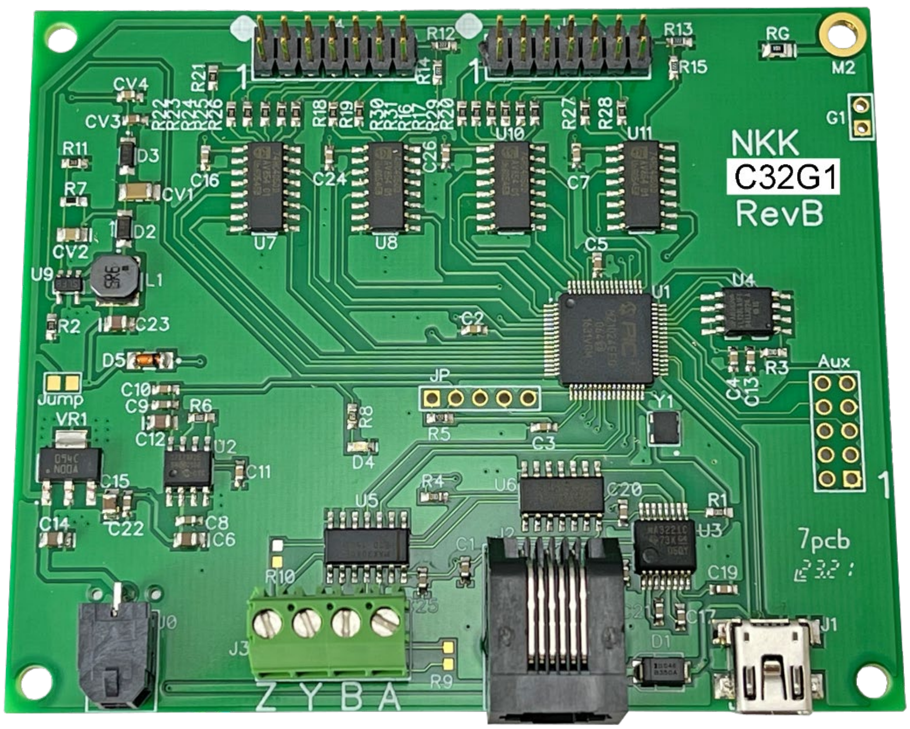


TABLE OF CONTENTS

Contents

1. General Controller Features

The IS-C32G1 controls up to 32 LCD 64x32 switches/ displays. The IS-C32G1 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 32 LCD 64x32 switches/displays. Two banks of 16.
- USB, RS232, RS422/RS485 communication.
- User downloadable images and backlight colors
- Look up tables for fonts 5x7 and 7x10.
- 8 LED Brightness settings.
- Memory for 16,000 images, backlights, and Attributes.
- Reports switch activities via serial port.
- Stand-alone operation or real time control by host.
- A 10 pins Auxiliary port with 7 MC pins for control or sense other devices

Switch-action report from the controller to the host:
- On switch press/release status change, the new statuses are sent over the last communication interface.

Set-up options:
- User defined blinking durations.
- Firmware upgradable via USB.

Real-time operation features:
- Download a graphic image data to any switch.
- Select any image from flash memory to display on any of switches.
- Create 6x8 font string for a specified row on a specified switch.
- Create 9x12 font string for a specified row on a specified switch.
- Change backlight color on any switch.

2. Compatible Logic Boards

2.1 Logic Boards

Switch Logic Boards: PCBs with mounted switches or displays that are used for this controller are called Logic Boards. The available Logic Boards are listed below.
Based on the number of Logic Boards and the length of the interconnect cables, a signal booster (IS-LBUF01) may be required.
All the Logic Boards have two 7x2 connectors and can be daisy chained via 14 pin ribbon cables. J2 of board A gets connected to J1 board B, 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 standard compatible Logic Boards; there are other Logic Boards available that are not listed as standard. Additionally new Logic Boards can be designed based on customer requirements.

Item	Part# with Socket and switch	Part# with switch	Description
1	IS-L02A1-CS	IS-L02A1-S	Logic Board, LCD 64x32 RGB, IS15EBFP4RGB, 2SW. Side by side stackable
2	IS-L02G1-CS	IS-L02G1-S	LOGIC BOARD, 1x2, LCD 64x32 RGB, IS15EBFP4RGB-09YN, 2SW. Side by side stackable
3	IS-L02H2-CS	IS-L02H2-S	LOGIC BOARD, 1x2, LCD 64x32 RGB, IS15ESBFP4RGB, 2SW. Side by side stackable
4	IS-L04G1-CS	IS-L04G1-S	LOGIC BOARD, 2x2, LCD 64x32 RGB, IS15EBFP4RGB-09YN, 4SW. Side by side stackable
5	IS-L16G2-CS	IS-L04G2-S	LOGIC BOARD, 4x4, LCD 64x32 RGB, IS15EBFP4RGB-09YN, 16SW. Side by side stackable

Note: All the logic board for IS15EBFP4RGB-09YN can be populated with the compact switch IS15ESBFP4RGB.

Note: Make sure the power is off when connecting or disconnecting the Logic Boards to or from the controller or each other.
Note: Connecting the Logic Boards improperly could damage either/both the Logic Boards and controller.

2.2 Signal Booster

This Logic Board signal booster may be needed when more than 10 logic boards are daisy chained or for long cable length.

Item	Part #	Description
1	IS-LBUF01	Signal Booster for Logic Boards

2.3 Ribbon Cables

These cables are used for connecting Logic Boards and the controller

Item	Part#	Length	Description
1	ISDCB81.2	1.2”	RIBBON CABLE, 14 CONDUCTORS, 28AWG, .050"
	ISDCB83	3”	RIBBON CABLE, 14 CONDUCTORS, 28AWG, .050"
	ISDCB88	8”	RIBBON CABLE, 14 CONDUCTORS, 28AWG, .050"
	ISDCB812	12”	RIBBON CABLE, 14 CONDUCTORS, 28AWG, .050"
	ISDCB824	24”	RIBBON CABLE, 14 CONDUCTORS, 28AWG, .050"
	ISDCB836	36”	RIBBON CABLE, 14 CONDUCTORS, 28AWG, .050"

These cables are used for connecting Logic Boards and the controller
Custom length cable can be made to order

3. Operational Detail

Power-up Sequence:
The system loads an image, backlight, and attribute from flash memory for 32 switches (even if the switches are not physically present). Switches 0-15 are connected to J4, and switches 16-31 are connected to J5. Images, attributes, and backlight addresses 1-32 are loaded to switches 0-31, respectively. The system then waits for a button press or host command while scanning for switch presses.

Main Operational Mode:
The controller waits for input or a host command. It will send the following bytes for a switch activity:
50 AA AA AA AA AA AA AA AA

Where AAs are the 4-byte switch number mask encoded in ASCII hex. The number is (1 << switch_press). A complete table is:

00000000	No switch pressed
00000001	Switch 0 pressed
00000002	Switch 1 pressed
00000004	Switch 2 pressed
00000008	Switch 3 pressed
00000010	Switch 4 pressed
00000020	Switch 5 pressed
00000040	Switch 6 pressed
00000080	Switch 7 pressed
00000100	Switch 8 pressed
00000200	Switch 9 pressed
00000400	Switch 10 pressed
00000800	Switch 11 pressed
00001000	Switch 12 pressed
00002000	Switch 13 pressed
00004000	Switch 14 pressed

00008000	Switch 15 pressed
00010000	Switch 16 pressed
00020000	Switch 17 pressed
00040000	Switch 18 pressed
00080000	Switch 19 pressed
00100000	Switch 20 pressed
00200000	Switch 21 pressed
00400000	Switch 22 pressed
00800000	Switch 23 pressed
01000000	Switch 24 pressed
02000000	Switch 25 pressed
04000000	Switch 26 pressed
08000000	Switch 27 pressed
10000000	Switch 28 pressed
20000000	Switch 29 pressed
40000000	Switch 30 pressed
80000000	Switch 31 pressed

LED Backlighting:

The LED code is one byte, Table below shows the format. A bit=0 denote OFF and a bit =1 denote ON. Bits 0 and 1 must be 1. There are 64 different color available

	LED code							
BIT	7	6	5	4	3	2	1	0
Color	R	R	G	G	B	B	1	1

The picture below is available backlight colors of LCD64x32. The top 2 digits on each color are the code for back light color.

03 00 00 00 Off	07 00 00 01 Midnight Blue	0B 00 00 10 Royal Blue	0F 00 00 11 Blue	43 01 00 00 Cherry	47 01 00 01 Fandango	4B 01 00 10 Purple	4F 01 00 11 Persian Purple	83 10 00 00 Brick	87 10 00 01 Rose	8B 10 00 10 Iris	8F 10 00 11 Red Violet	C3 11 00 00 Coral Red	C7 11 00 01 Carmine Pink	CB 11 00 10 French Rose	CF 11 00 11 Fuchsia
13 00 01 00 Astro Turf	17 00 01 01 Bluegrass	1B 00 01 10 Cerulean	1F 00 01 11 Electric Blue	53 01 01 00 Olive	57 01 01 01 Ashen	5B 01 01 10 Periwinkle	5F 01 01 11 Ceil	93 10 01 00 Gamboge	97 10 01 01 Salmon	9B 10 01 10 Puce	9F 10 01 11 Orchid	D3 11 01 00 Pumpkin	D7 11 01 01 Pink Orange	DB 11 01 10 Tea Rose	DF 11 01 11 Fuchsia Pink
23 00 10 00 Green	27 00 10 01 Jade	2B 00 10 10 Teal	2F 00 10 11 Sky Blue	63 01 10 00 Sap Green	67 01 10 01 Emerald	6B 01 10 10 Viridian	6F 01 10 11 Carolina Blue	A3 10 10 00 Citrine	A7 10 10 01 Maize	AB 10 10 10 Freezer Burn	AF 10 10 11 Thistle	E3 11 10 00 Golden Rod	E7 11 10 01 Amber	EB 11 10 10 Apricot	EF 11 10 11 Pink
33 00 11 00 Bright Green	37 00 11 01 India Green	3B 00 11 10 Pigment Green	3F 00 11 11 Juniper	73 01 11 00 Electric Lime	77 01 11 01 Light Lime	7B 01 11 10 Celery	7F 01 11 11 Turquoise	B3 10 11 00 Yellow Green	B7 10 11 01 June Bug	BB 10 11 10 Moss Green	BF 10 11 11 Baby Blue	F3 11 11 00 Canary Yellow	F7 11 11 01 Yellow	FB 11 11 10 Eggshell	FF 11 11 11 White

Backlight Color Code Table

The backlight brightness can be adjusted using the "Change brightness" command. See the associated command list for details.

Attributes:

Attributes determine system behavior when a button is pressed. Each switch is assigned an address on startup (1-32). Each address has image, backlight, and attribute information. The Attribute determine what new address every switch in the system changes to (if any). When a switch is pressed, the attribute from the address of image being displayed is executed.

The attributes are user defined and downloaded either using the engineering kits communicator or manually using the command. See the associated "command list" for detailed examples.

4. Communication Protocol

The controller communicates with the host by USB, RS232, or R422. The commands and responses are the same regardless of protocol. One command should be transmitted at the time. If multiple commands sent at the same time only the first one gets executed and the rest are ignored. The command behavior is:

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. One-byte commands do not transmit 79H. See the associated command list.
B. If the byte is not a command, it is ignored.

When the controller expects additional information:

A. A timer is set. If the expected data bytes are not received in 50ms, the controller transmits 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 01H, 20H to 2FH 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 in the associated command list.

An ASCII Hex byte is a normal hex byte split in two halves and converted to their ASCII equivalent (www.asciitable.com). This is a safety measure so that all data sent is not accidentally interpreted as a command. Most data sent after commands and sub-commands are encoded in ASCII Hex. The only exception is when sending live images (images not saved in memory and sent directly to the switches)

Conversion **TO** ASCII Hex:	Conversion FROM ASCII Hex:
x = ((data & 0xF0) >> 4)	x = ASCII hex byte 1
y = ((data & 0x0F) >> 0)	y = ASCII hex byte 2 (conversion not shown)
if (0x0 <= x <= 0x9) x+= 0x30	z = converted byte
if (0xA <= x <= 0xF) x+= 0x37	if (0x30 <= x <= 0x39) x -= 0x30
if (0x0 <= y <= 0x9) y+= 0x30	if (0x41 <= x <= 0x46) x-= 0x37
if (0xA <= y <= 0xF) y+= 0x37	z = (x << 4) + y

- RS232/422: 115K, 8bit no parity, one stop bit.
- The USB communicates is over a virtual comm port with the same settings as the RS232.

Image Format
Images are 64x32 monochrome .bmp files. They can be created in Microsoft Paint. Images should be sent using the NKK Engineering Kits Communicator. The engineering kit extract the 256-byte Image and encode it as ASCII hex, so 512 bytes will be sent over USB/RS232/RS422 serial.
If you want to download the image with a different software the image format is described in the table below.

	Left of image							Right of image
Line #1	Byte# 1	Byte# 2	Byte# 3	Byte# 4	Byte# 5	Byte# 6	Byte# 7	Byte# 8
Line #2	Byte# 9	Byte# 10	Byte# 11	Byte# 12	Byte# 13	Byte# 14	Byte# 15	Byte# 16
Line #31	Byte# 241	Byte# 242	Byte# 243	Byte# 244	Byte# 245	Byte# 246	Byte# 247	Byte# 248
Line #32	Byte# 249	Byte# 250	Byte# 251	Byte# 252	Byte# 253	Byte# 254	Byte# 255	Byte# 256

Images, Attributes, and Backlight Numbering
All images, attributes, and backlight numberings are inherently linked together. E.G. Image 0001 corresponds with Backlight 0001 and Attribute 0001, Image 0002 corresponds with Backlight 0002 and Attribute 0002, and so on. Care should be taken when ordering images, attributes, and backlights to send so that the desired behaviors are grouped together.

Switch Numbering
Switches 0-15 (0x01 to 0x0F) are connected to the J4 connector. Switches 16-31(0x10 to 0x1F) are connected to the J5 connector. Using switch 16 (0x10) would be the first switch on the J5 connector.

Attributes
An example attribute is shown on a different tab of command list spreadsheet. For no switch change, address FFFF should be used.

Backlights
An example of backlight code and download command is shown on a different tab of command list spreadsheet.

Flash Memory
The flash memory holds up to 16k images backlights and attributes. The flash memory is NOR flash. It means that it needs to be erased before new data is written. Erasing flash memory set all the bits to high. Writing to flash memory can only change a high bit to low bit. Writing the same information does not require erasing. However, if the information is changed the flash should be erased. The images, backlight data and attribute are saved in different region of flash. There are commands to erase the entire flash or individual region.

5. Commands

See IS-C32G1 Command List.xlsx for commands and example.

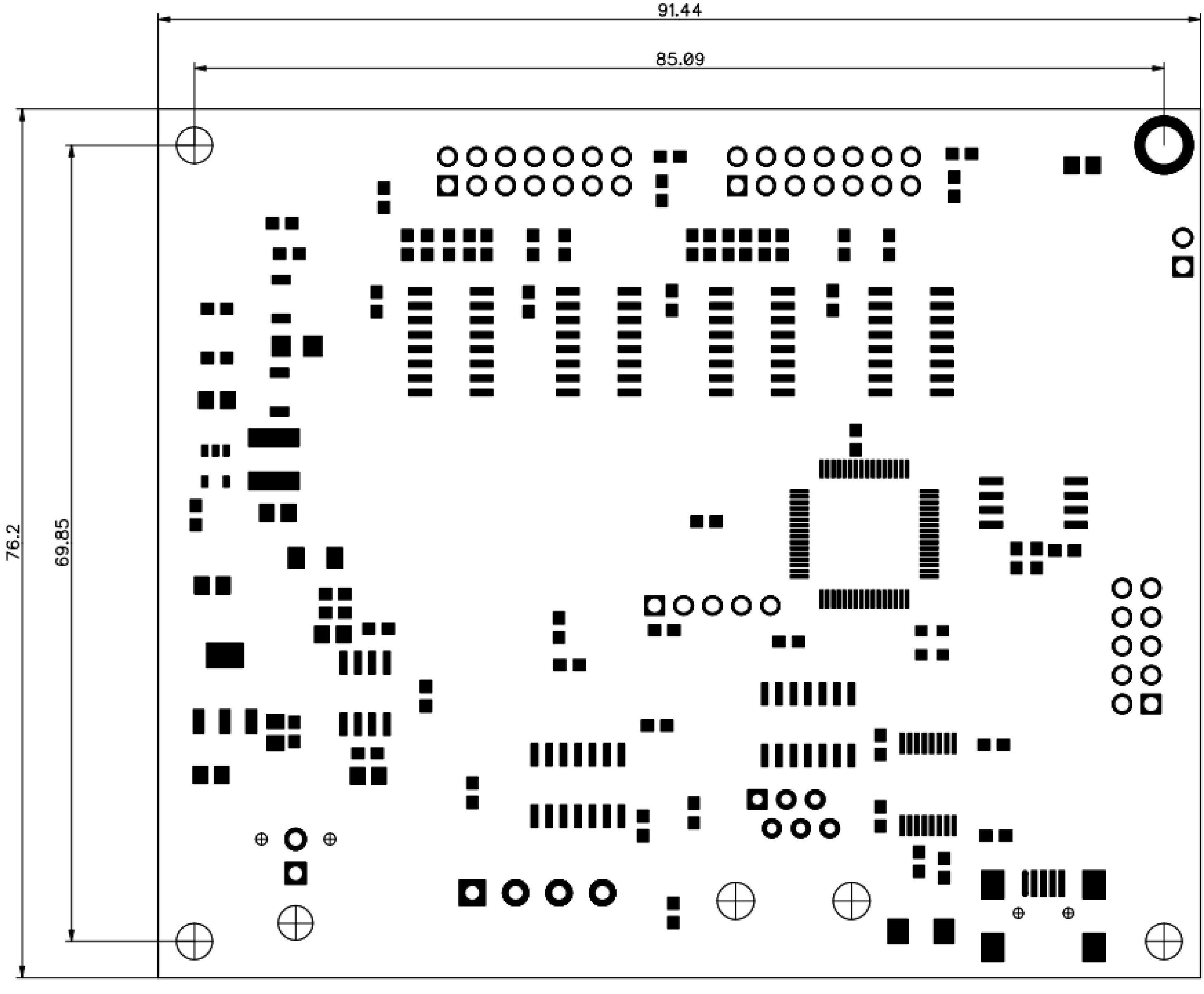
6. Hardware

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

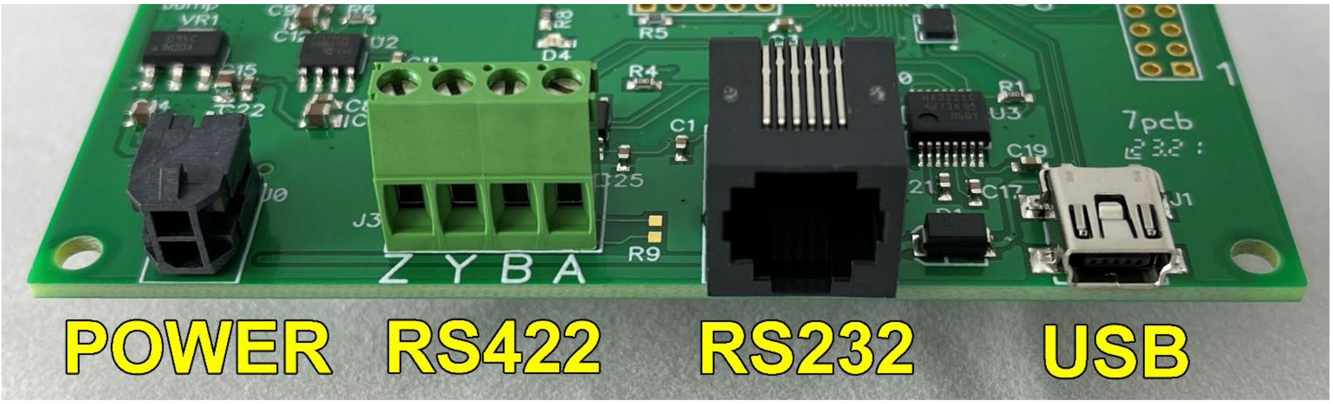
Power Requirement: 6.5V to 12V. Maximum current is 170mA without switches and 900mA with 32 switches.

Dimensions

Dimension: Board size: 3.6” x 3.0” (91.44 x 76.2 mm)
Mounting hole size: 0.125” (3.175 mm)



Connectors:



Power Connector: Molex 0430450200, top position power, bottom position ground.
Mating: Molex 0430250208

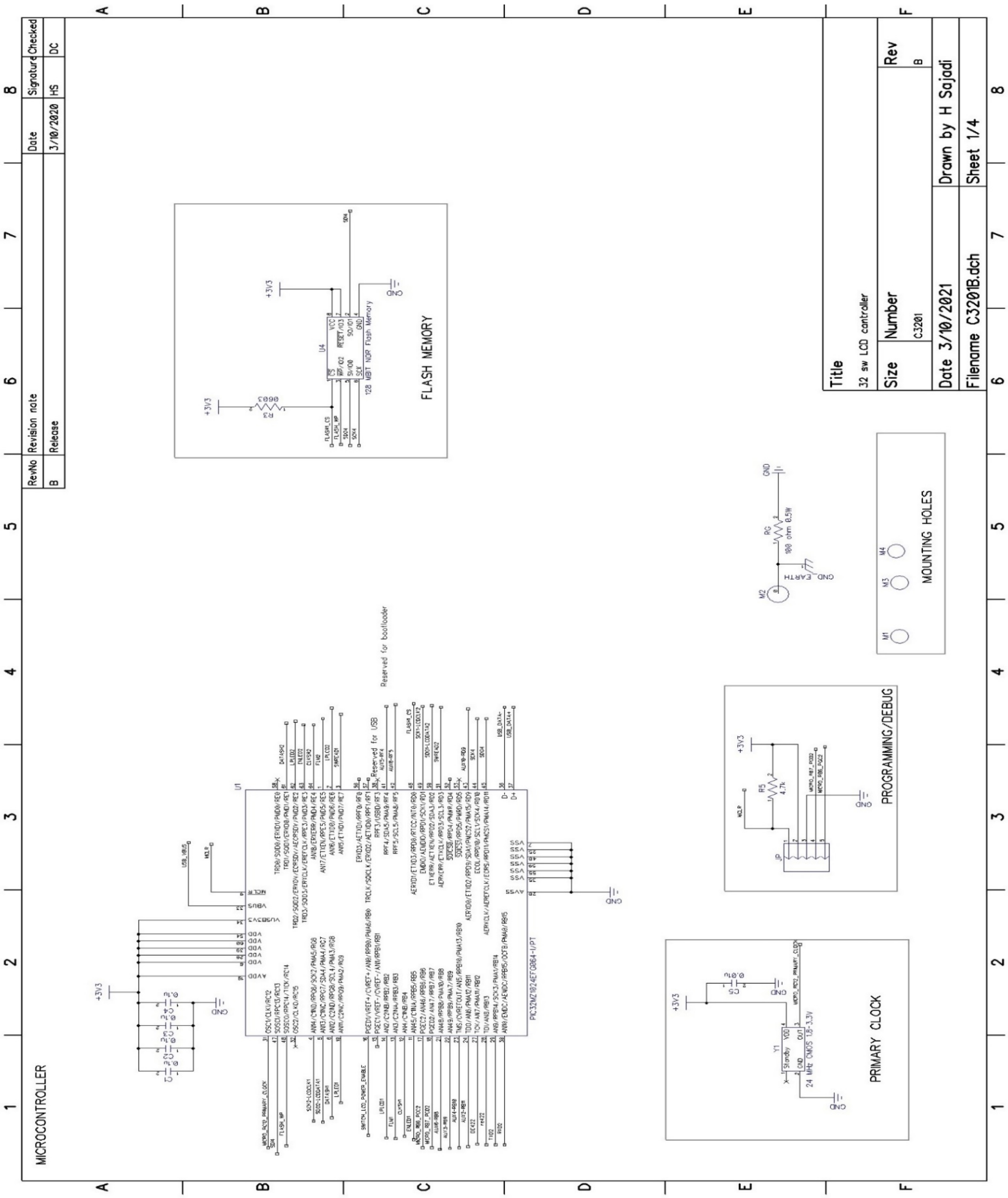
RS422 Connector: TE Connectivity 284392-4
Mating: Bare wire

RS485 Connector: TE Connectivity 284392-4 (R8&R9 have to be populated)
Mating: Bare wire

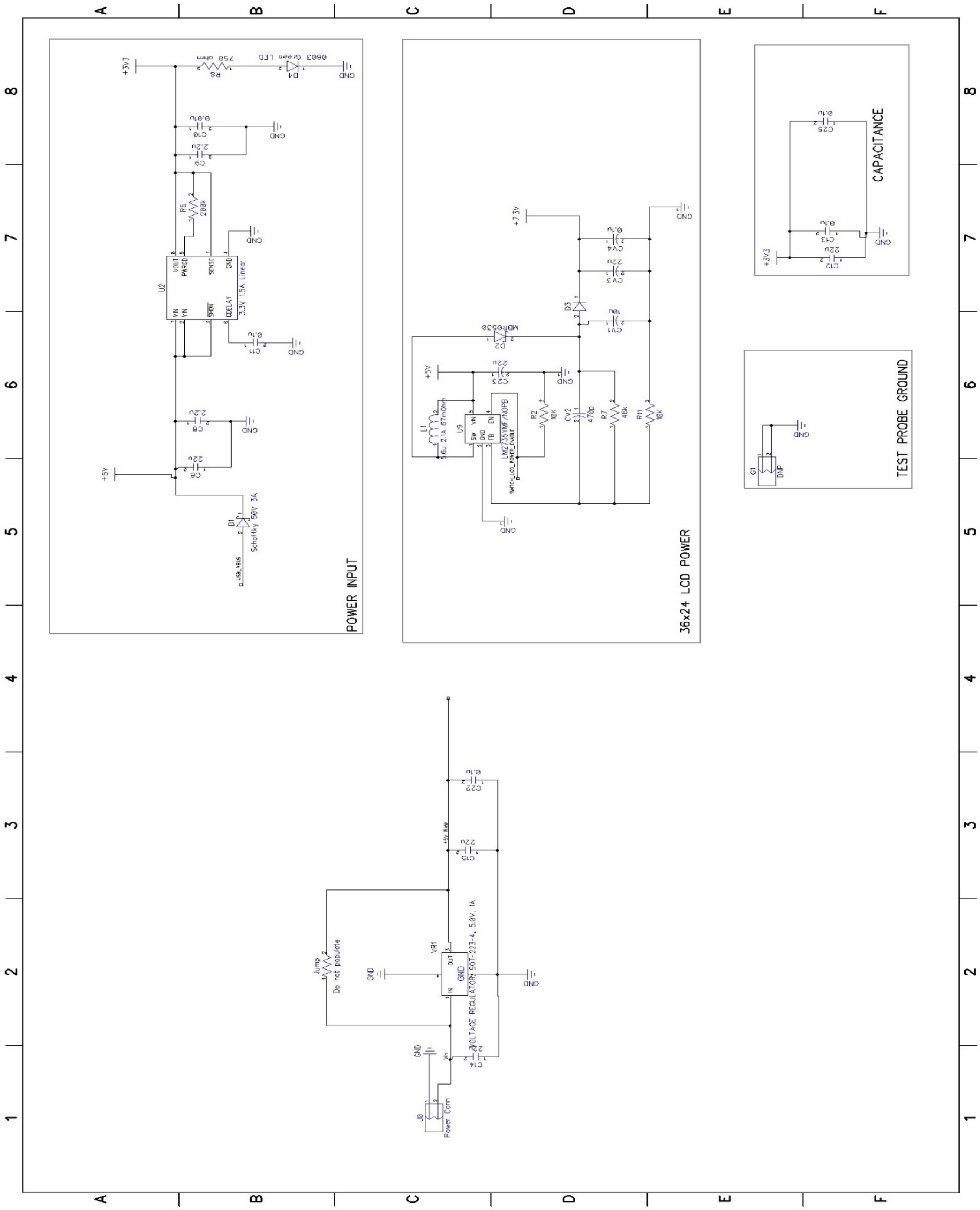
RS232 Connector: TE Connectivity 5555165-1
Mating: RJ25 connector

USB: TE Connectivity 1734035-2
Mating: USB Mini B

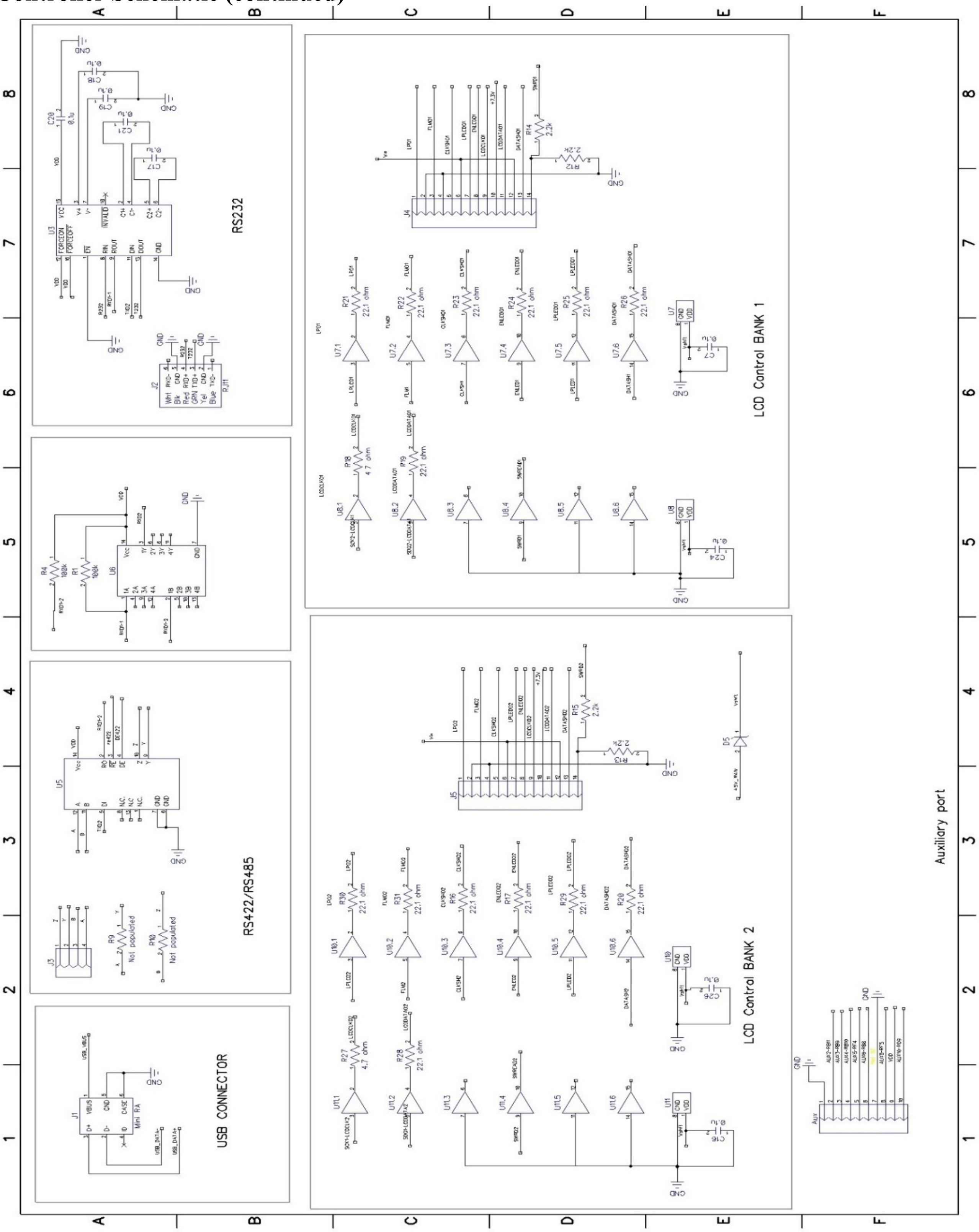
Controller Schematic



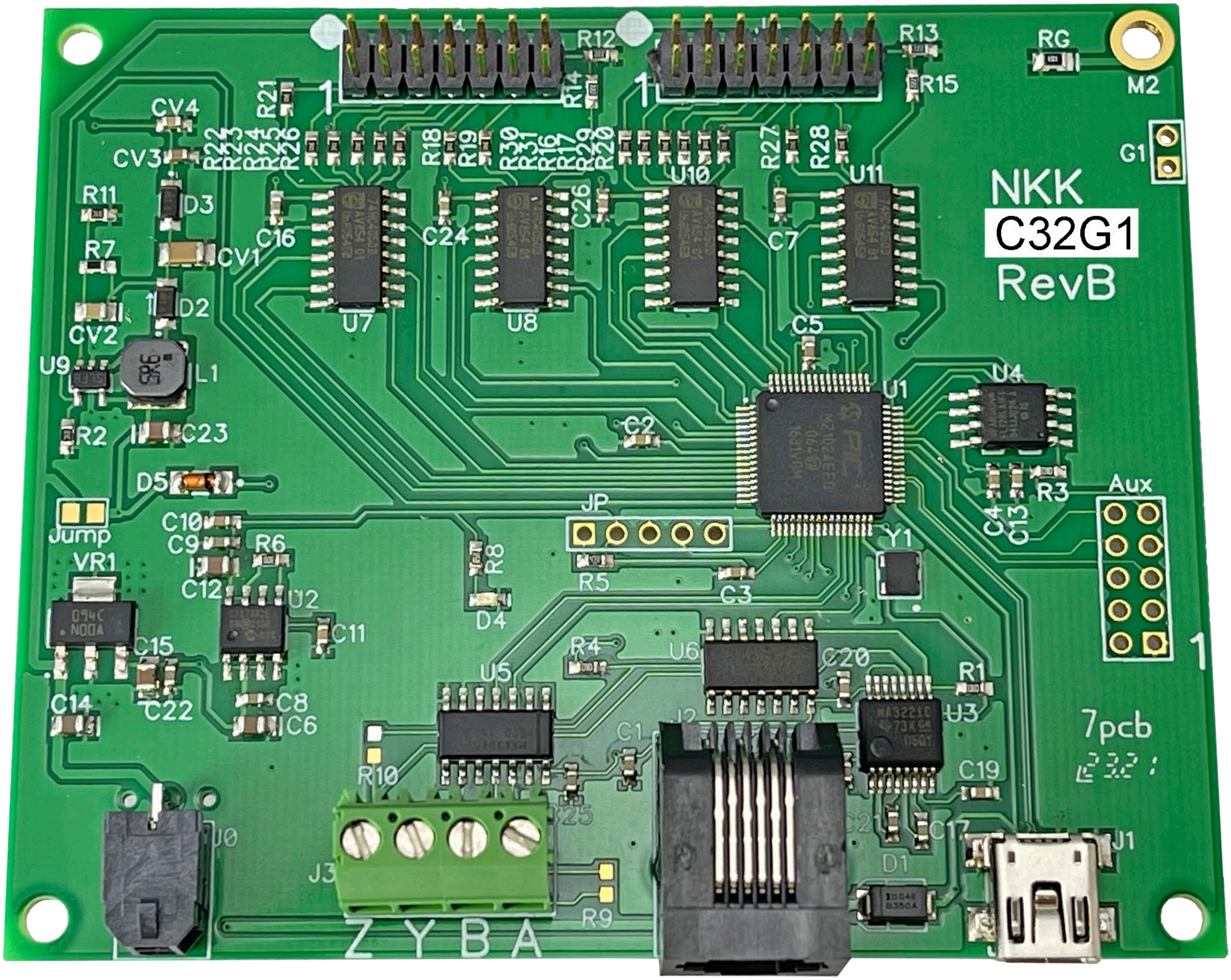
Controller Schematic (continued)



Controller Schematic (continued)



Board photo



7. Key Terms & Definitions

Host	Any computer, terminal, or other device that sends commands over USB, RS232, RS422
Controller	A PCB with a microcontroller that controls one or more logic boards
Logic board	A PCB with one or more SmartDisplays that can be daisy chained
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.
Communication format	There are two formats to transmit a byte: 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.

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 x 8) + (0 x 4) + (1 x 2) + (0 x 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

Digit 3: 2 x 256 = 512
Digit 2: (10) x 16 = 160
Digit 1: 7x1 = 7

679
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

Warranty

NKK SWITCHES LIMITED WARRANTY AND LIMITATION OF LIABILITY

The following limits our liability. Please read.

NKK Switches hereby warrants this product against any and all manufacturing defects for a period of one year from the date of sale of this product to the original end user. NKK Switches' liability in the event of such defect is limited to repair or replacement of the defective products. NKK Switches disclaims any liability or warranty obligation with respect to any product that is misused, damaged by any user, or not used in conformity with all applicable product specifications.

NKK SWITCHES HEREBY DISCLAIMS ANY WARRANTY, EXPRESS OR IMPLIED, OTHER THAN THAT CONTAINED HEREIN. NKK SWITCHES EXPRESSLY DISCLAIMS THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, AND SHALL HAVE NO LIABILITY BASED ON OR ARISING FROM ANY CLAIM OF SUCH WARRANTY.

NKK Switches shall have no liability to any person for any incidental, consequential, special, punitive, or other damages of any kind whatsoever relating to any use of this product.

USE OF THIS PRODUCT IN CONNECTION WITH ANY LIFE CRITICAL APPLICATION IS NOT RECOMMENDED.