Modtronix Engineering

Modular Electronic Solutions

Modtronix Engineering
PO Box 59
Castle Hill NSW 1765
Australia
Tel: +61 (0)405 353466
Fax: +61 (0)2 82094824
ABN: 48 663 118 043

SBC66ECL Revision 2

Ethernet Enabled Single Board Computer

Table of Contents

1 Introduction

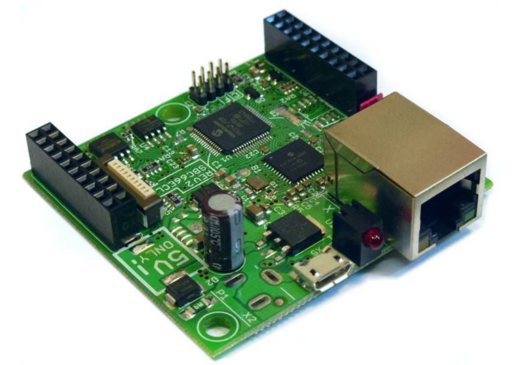
The SBC66ECL is a single board computer with a 10/100 Mbit/s Ethernet port and USB. In addition it has 25 free I/O port pins that can be configured for various input, output, analog and serial functions. It ships with a built in web server that allows it to be configured, monitored and controlled via the internet. It can also be accessed via Ethernet, USB, SPI, I2C or a Serial port.

Firmware can be updated in a secure way via USB, Internet or an Ethernet Network. For details, and latest firmware, see netcruzer.com/nzfirmware.

Web pages are written in standard HTML and JavaScript.

All source code (written in C) is available for free. It is compiled with the Microchip XC16 C compiler. A free student version of this compiler is also available.

Typical applications include HTTP Web servers, Mail clients, Ethernet to RS232 interface converter, Ethernet to RS485 interface converters, Remote control via Web Server, Protocol Bridge applications - Ethernet to UART, I2C, SPI etc.



Picture 1 - SBC66ECL Board

2 Features

- Assembled with the Microchip PIC24FJ128GB106 Microcontroller
- 10/100 MBit/sec Ethernet port, with Power Over Ethernet
- USB port via Micro B USB connector, can also power board
- 128KBytes internal FLASH memory for program and constant data
- 16KBytes Bytes of internal SRAM
- 16 MIP Microcontroller execution speed
- External 32MBit (4Mbytes) SPI FLASH
- External 8Kbyte SPI EEPROM
- External RTC with CR2032 Battery holder, can provide backup power for over 20 years (assuming SBC board is powered most of the time)
- 4 UARTs (RS-232, RS-485, LIN Bus...), that can be routed (via Microchip's Peripheral Pin Select feature) to any of 19 *Relocatable Ports*
- 2 SPI Ports, that can be routed (via Microchip's Peripheral Pin Select feature) to any of 19 *Relocatable Ports*
- 9 PWM Modules, that can be routed (via Microchip's Peripheral Pin Select feature) to any of 19 *Relocatable Ports*
- 1 I2C port
- 25 Digital 3.3V User I/O ports, of which 8 have on board 5V pull-up resistors to allow them to be used with 5V logic
- 11 of the I/O ports can be configured as 10-bit Analog Inputs (0 to 3.3V).
- 2.5V high precision external voltage reference for ADC converter
- Very High quality assembly, **with brand name, quality components**. No cheap, “no name brand” components are used!
- 3.3V Linear Regulator (Texas Instruments, ST or ON Semiconductors brand), with very high quality, brand name capacitors!
- Filtered analog supply to CPU for accurate analog measurements
- Daughter board connector with 40 pins for adding prototype, I/O or user expansion boards
- 10 pin FFC (Flat Flexible Cable) connectors with power and 8 I/O port. Uses standard 10 pin, 1.0mm spacing FFC cable
- Micro Match connector for connecting a serial I2C device (5V or 3.3V configurable), like the LCD2S serial LCD display with keypad decoder.
- Red System LED (will be changed to green in next revision. All SBC66 board will have green system LEDs)
- Shielded RJ45 network connector with link and activity LEDs
- Pin header with jumper that can be mounted in 3 positions: Park, "C" or "F". Software can determine what position jumper is in, and use "C" position to restore default configuration, and "F" position to update firmware via bootloader.
- Space for 2.1mm power connector to be assembled (for standard DC transformer). Is **not** assembled by default!
- Has an ICSP (In Circuit Serial Programming) connector – board can be programmed and debugged in circuit. For details see www.netcruzer.com/nzdoc-program.
- Project can be built with free fully functional Microchip XC16 compiler. Available from www.microchip.com.
- Wide operating temperature of -40 to 85°C.
- RoHS Compliant, Lead Free

3 Getting Started

3.1 Precautions

The SBC66ECL contains sensitive electronic circuitry, proper ESD precautions should be taken at all times when handling the board! Additionally, the Microcontroller can easily be damaged by improper use of the I/O ports. It is a good idea always connecting a resistor (1k or higher is recommended) in series with any I/O port. Never place the SBC66ECL on a conducting surface, ensure it is either mounted on a main board, in and enclosure, or placed on a non-conducting ESD safe surface!

3.2 Applying power

This board must be powered with a stable 5V DC supply! A higher voltage will damage the board!

At delivery, the board is configured to be powered via it's daughter board connector. This will be the case if using it with a main board. The main board will supply a regulated 5V supply to the SBC66ECL.

To power via the USB connector, solder jumper S16 has to be made. This allows any USB charger with a Micro USB connector to be used. These chargers have become very popular, and are currently the standard for mobile phones and other portable devices. Once doing this it is **very important** not to power the board from both USB and the daughter board connector (or 2.1mm power connector if assembled) at the same time! See Power Supply section of this document for details.

Additionally a 2.1mm power connector can be assembled to power the board via a DC transformer. See Power Supply section of this document for details.

4 Built in Web Server

At delivery, the SBC66ECL is programmed with the *Netcruzer Webserver*, and a *USB Bootloader*.

The Webserver enables the board to be controlled and monitored via it's web interface, UDP or USB commands. The Bootloader is used to update the firmware via USB or Ethernet. For details on the Webserver, see www.netcruzer.com/nz/doc/proj_webserver.

4.1 Updating Web Pages

Web pages are written in standard HTML, and can contain JavaScript and other client side scripts. The default web pages at delivery are located in the ".../src/common_prj/webserver/WebPages/Default" folder of the Netcruzer Download. They are shown here www.netcruzer.com/nz/doc/proj_webserver_webpages_default. To upload web pages to the SBC board, they are first compiled into a single "MPFS Image" file, and this file is then uploaded to the SBC board. There are a couple of ways to do this (for details see www.netcruzer.com/nz/doc/proj_webserver_webpages_update), they are:

- Use a web browser. Simply access the board's web interface (http://mchpboard), and go to *Upload Firmware and Webpages* section. Use the browse button to locate the web pages MPFS Image file, and upload it.
- In a web browser, enter the following address:
  http://mchpboard/mpfsupload
  Where "mchpboard" is the target board's NetBIOS or IP address (default is mchpboard). A web page will now appear in the browser asking for the MPFS Image file. This method is useful if the current web pages on the board are corrupt, and the method above is not available.
- Use the Microchip MPFS2 application to create the MPFS Image and upload it to the target board.

5 Updating Firmware

At delivery, the SBC66ECL is programmed with the *Netcruzer Webserver*, and a *USB Bootloader*. Firmware can be updated via the web interface (provided by the Netcruzer Webserver), or the USB bootloader. The firmware update is done in a very secure way, even a power failure will not corrupt it. New firmware is first copied to a location in Flash memory, an only after it has successfully been uploaded, it is used to update the firmware.

5.1 Updating Firmware via the Web Interface

To update the firmware via the Web interface, enter the board's address in a web browser (default is http://mchpboard), and go to the *Upload Firmware and Webpages* section. Use the browse button to locate the firmware (a *.hex file), and upload it. For details, see www.netcruzer.com/nz/doc/update_firmware_web.

5.2 Updating Firmware via USB

To update the firmware via USB, the board must be connected to a PC running the *Netcruzer USB Bootloader* application. It must then be placed into bootloader mode (hold button in at power-up), after which the new firmware (*.hex file) can be uploaded to the board. For details, see www.netcruzer.com/nz/doc/update_firmware_usb.

5.2.1 Details

The *Netcruzer USB Bootloader* PC application is written with QT, a cross-platform application and UI framework. Full source code is available for free from www.netcruzer.com. All development tools for modifying and compiling source code is available for free from http://qt.nokia.com It is very easy, for example, modifying this application to display your companies name on logo.

Currently Modtronix Engineering only supports the Windows build of this application. It can however also be built for Linux or Mac.

5.3 Updating Firmware Technical Details

The firmware updating process is very secure. It will not get corrupted even if a power failure occurs during the process. Space is reserved in the external Flash chip for two copies of the firmware (Firmware and contents of external EEPROM for configuration data). One will always be marked as "C*urrent"*. When new firmware is downloaded, it is copied to the second area in external Flash, and marked as *"New"* when successful. On a reboot, the bootloader will check for new firmware, and if available, update the firmware with it. The The following is a simplified description of how it works:

1. New firmware is downloaded, and copied to the external Flash chip. The external flash will now have two copies of Firmware, one with *"Current"*, and one with *"New"* firmware.
2. On completion, downloaded firmware is verified. If OK, it is marked as *"New"*. If an error occurred before marking it as *"New"*, it will not be used for updating the firmware.
3. Device is restarted. On power up, device will check for new firmware in external Flash. If found, will start updating firmware. When done, firmware is verified, and if OK, old firmware is marked as *"Previous"*, and new firmware marked as *"Current"*, in this order! If an error occurs during the firmware updating, it will be attempted again at next power up, seeing that firmware in external Flash will still be marked as *"New"*. Only after successfully updating firmware it is changed from *"New"* to *"Current"*. This ensures firmware is successfully updated even if a power failure or other error occurs during update process.

5.4 Reverting back to previous version of Firmware

Seeing that the old Firmware is still present in the external Flash (marked as *"Previous"*), it is possible reverting back to the previous version of Firmware if the new Firmware does not work. This feature will be implemented in a future version of the *Netcruzer Webserver*.

6 User Inputs and Outputs

6.1 Netcruzer Ports

The user input and output ports of the SBC66ECL are referred to as *Netcruzer Ports*, and are listed in Table 1 below. They are listed for all Netcruzer Products at www.netcruzer.com/nzdoc-sbc66-ports. All ports listed can be used as 0 to 3.3V digital input or output ports. They can additionally be configured for functions as listed in Table 1 below. The **RP, A**, **I2C** and **5V** columns headers in Table 1 have the following meaning:

- **Netcruzer Port**: This is the Netcruzer Port name. All Prototype boards use these labels. They are also used in the firmware code to address ports.
- **CPU Port**: This is the PIC Microcontroller port used for this Netcruzer Port, as listed in the PIC24FJ128GB106 datasheet
- **RP**: Relocatable Port, can be assigned different functions as listed in 6.2 Relocatable Ports section below (PWM, UART, SPI, Interrupt input...).
- **A**: Analog Port, can be connected to the 10-bit ADC or Voltage Comparator inputs of the Microcontroller. When marked **A**, the port can be configured as a 10-bit ADC input, when marked **C** it can be configured as a Comparator input. When marked **AC** it can be both.
- **I2C**: I2C port, the number will indicate what I2C port of the Microcontroller it is connected to. A **C** will indicate I2C Clock, and **D** I2C Data.
- **5V**: 5V tolerant input ports. When marked **5V**, it **i**ndicates this port can tolerate 0 to 5V inputs. When marked **5C**, it indicates this port can tolerate 0 to 5V inputs, and can be configured as an Open Collector output with 5V pull-up resistors. The pull-up resistors are enabled via solder jumpers at the bottom of the board. They are not made by default.

Table 1: Netcruzer Ports

Netcruzer Port	CPU Port	RP	A	I2C	5V	Comment	Netcruzer Port	CPU Port	RP	A	I2C	5V	Comment
S0						No Connection!	X6	G6	RP	C			General purpose IO.
S1						No Connection!	X7	G7	RP	C			General purpose IO.
SC	D10			C1		I2C 1 Clock, connected to on board RTC and red Micro Match connector (for LCD)	X8	G8	RP	C			General purpose IO.
SD	D9			D1		I2C 1 Data, connected to on board RTC and red Micro Match connector (for LCD)	X9	G9	RP	C			General purpose IO. Can be connected to RTC Square wave output via solder jumper S15.
T0	B6					Used for ICSP (In Circuit Serial Programming), and System LED. Do not use!	X10	G8	RP	C			**Note:** This pin is connected to X8!
T1	B7					Used for ICSP (In Circuit Serial Programming), and to detect Bootloader mode at startup (when high). Do not use!	X11	G9	RP	C			**Note:** This pin is connected to X9!
T2	RST					Connected to MCLR pin of Microcontroller. When 0V, board will reset	Y0	B8	RP	A			General purpose IO.
T3						No Connection!	Y1	B9	RP	A			General purpose IO.
T4	F1				5C	General purpose IO.	Y2	B10		A			General purpose IO.
T5	F3	RP			5C	General purpose IO.	Y3	B11		A			General purpose IO.
T6	F4	RP			5C	General purpose IO.	Y4	B12	RP	A			General purpose IO.
T7	D8	RP			5C	General purpose IO.	Y5	B13	RP	A			General purpose IO.
X0	B0					**Note:** Connected to 2.5V external voltage reference, and **not** available.	Y6	D0	RP			5C	General purpose IO.
X1	B1	RP	A			General purpose IO.	Y7	D1	RP			5C	General purpose IO.
X2	B2	RP	A C			General purpose IO.	Y8	D2	RP			5C	General purpose IO.
X3	B3		A			General purpose IO.	Y9	D3	RP			5C	General purpose IO.
X4	B4	RP	A C			General purpose IO.	Y10	D6					Power Sense input. Can still use as output, seeing that sense is via 5K resistors.
X5	B5	RP	A			General purpose IO.	Y11	D7					General purpose IO.

6.2 Relocatable Ports

The SBC66ECL has 19 RP (Relocatable Ports), as listed in Table 1 above. A relocatable port can be assigned various input and output functions. The relocatable ports are: T5, T6, T7, X0, X1, X2, X4, X5, X6, X7, X8, X9, X10, X11, Y0, Y1, Y4, Y5, Y6, Y7, Y8 and Y9.

They can all be used as general purpose I/O ports, but in addition can be configured for the following functions:

- **Comparator Output**: Three comparator modules are available. The inputs to the comparator can be configured to use any one of four external analog inputs, and the output can be configured to be any RP port.
- **External Interrupt Input**: Up to 5 external interrupts sources are available, with many configuration options available.
- **Input Capture**: Up to 9 independent input capture modules are available. Each offers a wide range of configuration and operating options for capturing external pulse events and generating interrupts.
- **PWM Output**: Up to 9 PWM channels are available. Each PWM has it's own timer, and can independently be configured .
- **SPI**: Up to 3 SPI modules are available. By default, SPI 1 is used by the Webserver software provided with the SBC66ECL to communicate with the on-board external Flash and EEPROM. This leaves 2 SPI modules that can be used by the user.
- **Timer External Input**: Four Microcontroller 16-bit Timers (Timer 2 to 5) can be configured for external timer inputs.
- **UART**: Up to 4 UART (Universal Asynchronous Receiver Transmitter) ports are available. The UART is a full-duplex asynchronous system that can communicate with peripheral devices, such as personal computers, LIN, RS-232 and RS-485 interfaces. The module also supports a hardware flow control option with the UxCTS and UxRTS pins and also includes an IrDA® encoder and decoder.

6.3 Analog Inputs

The SBC66ECL has 11 analog inputs, as listed in Table 1 in chapter 6.1. They are *Netcruzer Ports* X1 to X5, and Y0 to Y5. All analog inputs have a 10-bit resolution. They are configured to use the external 2.5V voltage reference via port X0.

The ADC unit can be configured to use the precision 2.5V external voltage reference, or the internal 3.3V supply as a reference voltage. Using 2.5V will result in more precise measurements, seeing that the precision external reference is used.

The inputs can always tolerate voltages from 0 to 3.3V. When configured to use the 2.5V reference, only voltages from 0 to 2.5V can be measured. When configured to use the 3.3V reference, voltages from 0 to 3.3V can be measured. To measure higher voltages, a simple 2 resistor divider can be connected between the input and voltage source.

6.4 PWM Outputs

The SBC66ECL has 9 Output Compare modules, that can be used as PWM outputs. Any of the 19 *Relocatable Ports* (listed in Table 1 of chapter 6.1) can be configured as PWM outputs. Each of these modules offers a wide range of configuration and operating options for generating pulse trains on internal device events, and can produce pulse-width modulated waveforms for driving power applications. Key features of the output compare module include:

- Hardware-configurable for 32-bit operation in all modes by cascading two adjacent modules
- Synchronous and Trigger modes of output compare operation, with up to 30 user-selectable trigger/sync sources available
- Two separate period registers (a main register, OCxR, and a secondary register, OCxRS) for greater flexibility in generating pulses of varying widths
- Configurable for single-pulse or continuous pulse generation on an output event, or continuous PWM waveform generation
- Up to 6 clock sources available for each module, driving a separate internal 16-bit counter

7 Serial Ports

7.1 UARTs

The SBC66ECL has 4 UART (Universal Asynchronous Receiver Transmitter) modules. Any of the 19 *Relocatable Ports* (listed in Table 1 of chapter 6.1) can be configured as a UART. The UART is a full-duplex asynchronous system that can communicate with peripheral devices, such as personal computers, LIN, RS-232 and RS-485 interfaces. The module also supports a hardware flow control option with the UxCTS and UxRTS pins and also includes an IrDA® encoder and decoder.

- Full-Duplex, 8 or 9-Bit Data Transmission through the UxTX and UxRX Pins
- Even, Odd or No Parity Options (for 8-bit data)
- One or two Stop bits
- Hardware Flow Control Option with UxCTS and UxRTS Pins
- Fully Integrated Baud Rate Generator with 16-Bit Prescaler
- Baud Rates Ranging from 1 Mbps to 15 bps at 16 MIPS
- 4-Deep, First-In-First-Out (FIFO) Transmit Data Buffer
- 4-Deep FIFO Receive Data Buffer
- Parity, Framing and Buffer Overrun Error Detection
- Support for 9-bit mode with Address Detect(9th bit = 1)
- Transmit and Receive Interrupts
- Loopback mode for Diagnostic Support
- Support for Sync and Break Characters
- Supports Automatic Baud Rate Detection
- IrDA Encoder and Decoder Logic
- 16x Baud Clock Output for IrDA® Support

7.2 SPI

The SBC66ECL has 3 SPI (Serial Peripheral Interface) modules. Any of the 19 *Relocatable Ports* (listed in Table 1 of chapter 6.1) can be configured as a SPI Port. By default, SPI 1 is used by the Webserver software provided with the SBC66ECL to communicate with the on-board external Flash and EEPROM. This leaves 2 SPI modules that can be used by the user. They can be configured for a maximum frequency of 8 mbits/second. The Serial Peripheral Interface (SPI) module is a synchronous serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D Converters, etc. The SPI module is compatible with Motorola's SPI and SIOP interfaces. The module supports operation in two buffer modes. In Standard mode, data is shifted through a single serial buffer. In Enhanced Buffer mode, data is shifted through an 8-level FIFO buffer.

7.3 I2C

The SBC66ECL has 1 I2C Port.

By default, **I2C 1** is used by the Webserver software provided with the SBC66ECL to communicate with the on-board external RTC Chip, and is also connected to the red Micro Match connector, and the SC and SD pins of the Daughter Board connector. Seeing that the RTC has a maximum clock rate of 400kbits/sec, I2C 1 can not be configured for a higher bit rate than 400kbits/sec. I2C 1 is also connected to the Micro Match connector, which can be used to communicate with a LCD2S serial LCD display (available at www.netcruzer.com). I2C 1 can be configured via solder jumpers S1 and S2 for 3.3V or 5V operation. By default it is configured for 5V, meaning it can communicate with other 5V I2C devices, like the LCD2S display for example. It is **important** to note that **both S1 and S2 should be** configured for **the same** voltage, either both 3.3V, or both 5V!

The Inter-Integrated Circuit (I2C) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, display drivers, A/D Converters, etc.

8 Connectors

8.1 Daughter Board Connectors

The SBC66ECL has two 2x10 pin, 2.54mm female connectors. They contains all *Netcruzer Port* pins, as listed in Table 1 of chapter 6.1. The location of the pins are shown in Figure 1 on the right. The yellow text gives the *Netcruzer Port* name, and the gray text (in brackets) the Microcontroller Port pin.

For the location of the Daughter Board connectors, see the Dimensions chapter at the end of this document.

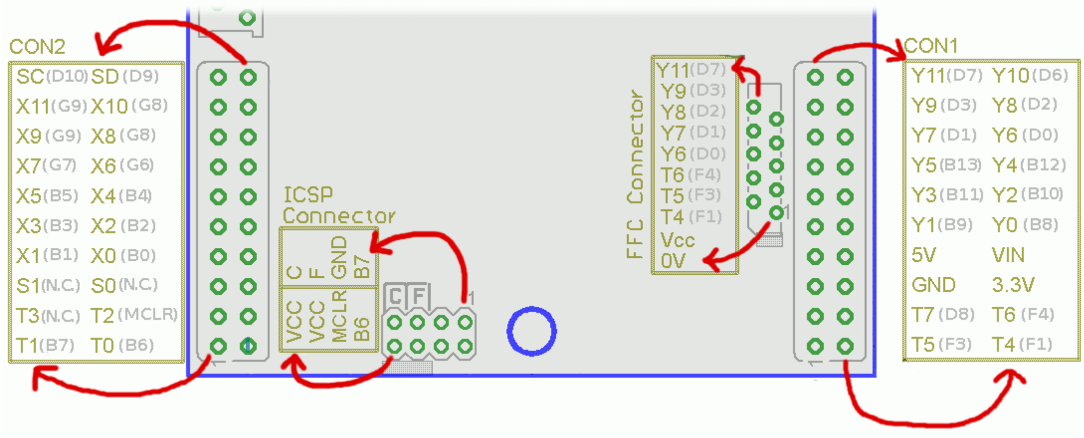
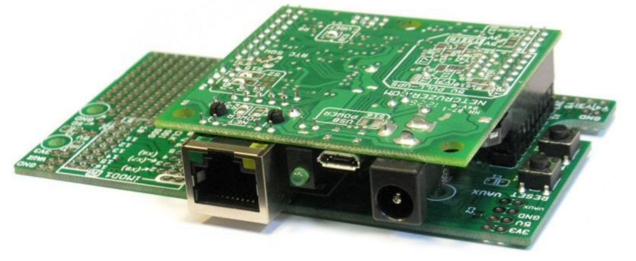


Figure 1 - Daughter Board, Programming and FFC Connectors

8.1.1 As a Daughter Board

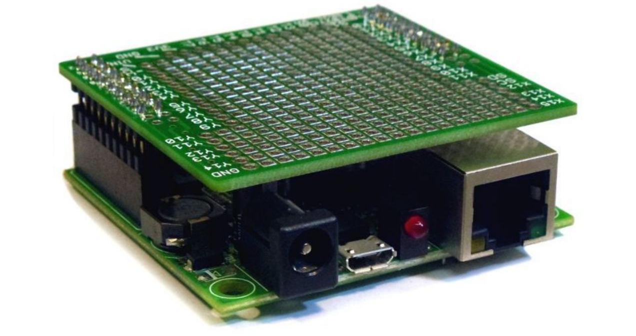


Picture 2 - SBC66ECL used as a daughter board on PT66ECI board

The SBC66ECL can be used to add Ethernet and USB capabilities to any board by using it as a daughter board. The board that is to take the SBC66ECL as a daughter board needs to provide two 2x10 pin, 2.54mm pin headers for the SBC66ECL to plug into. Additionally, 1 PCB supports can also be provided if additional stability is required. The 2x10 pin header connectors available from netcruzer.com are specially made so that when mated with the connectors on the SBC66ECL, the main board and SBC66ECL will be 14.3mm (9/16") apart. This is a standard height for PCB supports. All connectors and PCB standoffs can be purchased from www.netcruzer.com, or other electronic components suppliers.

The Picture to the right shows the SBC66ECL with a PT66ECI prototype daughter board. This and other prototype boards are available from our site at www.netcruzer.com/products-sbc66-prototype. Additionally, the Eagle PCB CAD files can also be downloaded on the product page, and can be used for creating custom boards.

8.1.2 Expansion boards



Picture 3 - SBC66ECL with a PT02TC daughter board

The SBC66ECL can be used as a full functional Single Board Computer. It's Daughter Board connector can be used as an expansion port to add additional functionality. It contains all *Netcruzer Ports* and power. For a list of available daughter boards see www.netcruzer.com. The Picture to the right shows the SBC66ECL with a PT02TC prototype daughter board plugged onto it.

Additionally, users can download PCB templates for creating their own Daughter Board from our site. Eagle PCB CAD files are provided for all our prototype board listed at http://netcruzer.com/products-sbc66-prototype/

8.2 ICSP connector

The SBC66ECL has an ICSP (In Circuit Serial Programming) connector. This enables the Microcontroller to be programmed and debugged in circuit if required. This will only be used for development, seeing that it is not required to update firmware (via USB or Network). For details on programming and debugging in circuit see www.netcruzer.com/nzdoc-program.

8.3 FFC Connector

The SBC66ECL has an 10-pin, 1.00mm Flat Flexable Cable connector. This connector has 8 *Netcruzer Ports*, and power. For details and pin outs see the Dimensions chapter at the end of this document. The available ports are T4, T5, T6, Y6, Y7, Y8, Y9 and Y11. The supply can be configured for 3.3V or 5V, and are selected via S17 solder jumper at the back of the board. **Note**, this solder jumper is not made by default!

The 8 port pins can be configured to be 3.3V digital inputs(5V tolerant) or outputs. Seven of them can be configured (via solder jumpers) as 5V outputs (open collector). Five of them are *Relocatable Ports*, and can be configured as UART, SPI, PWM, Interrupt, Timer inputs or Capture inputs (via PPS feature of CPU). Cables from many manufacturers can be used. The exposed end of the cable should be about 4.0mm. The width of the exposed end has to be 11.0mm. The following cables from Molex (available from Digikey) can be used:

Molex Part numbers 21039-0247, 21039-0249, 21039-0251, 21039-0253, 21039-0259, 21039-0263 and 21039-0265.

8.4 Micro Match Connector

A 6-pin Micro Match connector is provided for adding I2C slave devices, like a LCD2S serial LCD display. The Webserver firmware provided with the SBC66ECL supports this type of Display, and can write text to it, and read keys from it (if the LCD2S has a keyboard attached to it). The I2C voltage can be configured for 3.3V or 5V, see chapter *7.3 I2C* for details. Power is connected to pins 2 and 3, and is also configurable for 3.3V or 5V, as described in chapter *7.3*.

Pin 6 is an interrupt input, and can be used by the I2C slave device to signal it has data available. For example when using a LCD2S that has a keyboard attached to it, the LCD2S will pull this pin low (interrupt) when a key is pressed. This will signal the SBC66ECL that there is data available, and it will read the key data from the LCD2S via the I2C 1 bus.

A range of 16x2 and 20x4 character LCD2S serial LCD displays are available from netcruzer.com.

Pin	Signal	Description
1	SDA	I2C 1 Data
2	Vcc	Positive Power supply, configured via S1 and S2 for 3.3V or 5V.
3	0V	Negative Power supply
4	SCL	I2C 1 Clock
5	N.C.	No Connection
6	Int	Interrupt input (CPU pin RF0

Figure 2 - Micro Match Connector

9 Power Supply

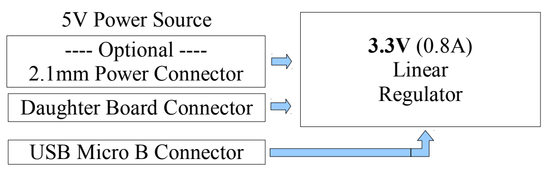


Figure 3 - Power Supply

The SBC66ECL needs an external 5V power supply. The 5V power can be supplied via the USB Micro B connector, or the daughter board connector. Optionally, a 2.1mm power connector can be added for supplying power using a 5V regulated power supply.

The SBC66ECL has a 3.3V (800mA) Linear Regulator to generate the 3.3V required for the Microcontroller and other components. The SBC66ECL only uses about 120mA. The rest is available for user applications. For example, if the SBC66ECL is used as a daughter board on a custom board, it can supply the custom board with 3.3V.

When using a 2.1mm power connector to supply the board with 5V, a protection circuit protects the board from over voltage and reverse polarity. A resettable fuse (PTC) is used for protection. If this fuse is triggered, it takes a couple of seconds (minutes) to reset (for PTC to cool), before the board will work again.

For details on powering the board via the USB port, see 9.2 Power via USB Connector section below for details.

9.1 Power Connector

The power connector is a 2.1mm type, with center positive.

9.2 Power via USB Connector

The SBC66ECL has a micro B USB connector, that can provide the board with power. To enable powering the board via USB, solder jumper S16 at the back of the board has to be made. Once jumper S16 is made, it is **very important** not to power the board from both USB and the 2.1mm power connector (if assembled)!

10 Hardware Configuration

The SBC66ECL board has various solder jumpers on the back of the board. They are used to configure the board for different mode, and are described in details in relevant sections of this document.

11 Hardware

11.1 Ethernet

The SBC66ECL has a 10/100 Mbs Ethernet port. The RJ45 connector meets IEEE 802.3 standards and FCC mechanical requirements. The RJ45 connector has two built in LEDs, a green LED for link indication, and a yellow LED for activity.

11.2 USB

The SBC66ECL has a Micro B USB connector. With the default software installed on the board at delivery, the USB connector is used for the following:

- Updating firmware via the USB bootloader. For details, see www.netcruzer.com/nz/doc/update_firmware_usb.
- Sending debug messages and information. To view this information, the *Netcruzer USB Terminal* application must be installed. For details, see netcruzer.com/usbterminal
- Sending commands, and requesting data from the SBC66ECL. This is done via the “Commands” tab of the *Netcruzer USB Terminal* application. See link given above for details.

11.3 RTC

The SBC66ECL has space for assembling an external RTC (real time clock) and battery holder for a CR2032 battery. They are **not assembled** by default! The RTC chip used is the Intersil ISL1208, which features extremely low battery supply current of only 400nA. It is connected to the I2C 1 bus of the Microcontroller, and has a maximum speed of 400 kbits/sec.

11.4 System LED

A red (will be green on next revision) status LED on the front of the SBC66ECL indicates it's current status.

- During normal operation, with USB **not** connected, the LED will have one long (600ms) and one short (100ms) flashe every 1.6 seconds.
- During normal operation, with USB connected, the LED will have one long (600ms) and two short (100ms) flashes every 1.6 seconds.
- When the board is in bootloader mode, the LED will have two long, followed by 1 or 2 short flashes. For details, see www.netcruzer.com/nz/doc/update_firmware_usb.

11.5 Microcontroller

The SBC66ECL has a PIC24FJ128GB106 16-bit Microcontroller.

11.6 Memory

11.6.1 External EEPROM

The SBC66ECL board has a 4 Kbyte external EEPROM for storing configuration data. The default data for the EEPROM is stored in the external FLASH, and is used to program the EEPROM when a firmware update is done.

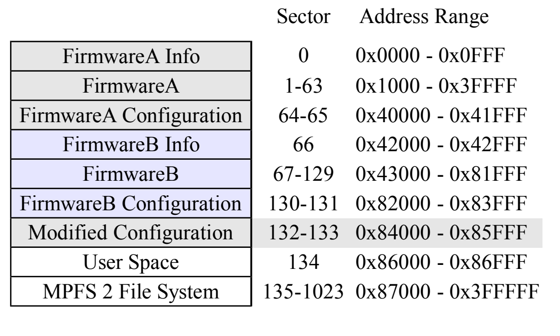
11.6.2 External FLASH

The SBC66ECL board has a 4 Mbyte external FLASH for storing web pages, firmware updates and user data, as shown in Figure 4. It consists of 1024 sectors of 4K Bytes each. Sectors are the smallest erasable areas.

Sectors 0 to 131 contain two copies of the Firmware and configuration data. The first copy (A) is located in sectors 0-65, and the second copy in sectors 66-131.

Each copy has 3 parts. *FirmwareA Info* and *FirmwareB Info* contains the current state of the Firmware, a check-sum and additional information. *FirmwareA* and *FirmwareB* contains the actual *User Program* that is written to the microcontroller's program memory. *FirmwareA Configuration* and *FirmwareB Configuration* contain the default Configuration Data used by the firmware, and is written to the external EEPROM when a firmware update takes place.

Under normal conditions, one copy will be the current firmware, and the other the previous. It is always possible reverting back to the previous version of firmware. When new firmware is loaded via the bootloader, the *previous* copy gets overwritten by the new firmware. At the next power cycle, this new firmware will be used to upgrade the board, and the current firmware will be marked as the previous firmware. If the new firmware is faulty, the previous firmware can be restored.



	Sector	Address Range
FirmwareA Info	0	0x0000 - 0x0FFF
FirmwareA	1-63	0x1000 - 0x3FFFF
FirmwareA Configuration	64-65	0x40000 - 0x41FFF
FirmwareB Info	66	0x42000 - 0x42FFF
FirmwareB	67-129	0x43000 - 0x81FFF
FirmwareB Configuration	130-131	0x82000 - 0x83FFF
Modified Configuration	132-133	0x84000 - 0x85FFF
User Space	134	0x86000 - 0x86FFF
MPFS 2 File System	135-1023	0x87000 - 0x3FFFFF

Figure 4 - External Flash Memory Map, **Byte** addresses shown

The *Modified Configuration* block is used by the firmware if the default configuration is updated and saved. If the data contained in the external EEPROM becomes corrupt at any time, it is restored with the data contained in *FirmwareX Configurtion* or *Modified Configuration*.

Following this is the *User Space*, a variable size block of memory, that can be used by the user custom applications. By default it is only 4k Bytes (1 sector) in size. It can be changed by changing the value of the MPFS_RESERVE_BLOCK define in the source code.

Following this is the MPFS 2 *File System*, that currently contains the web pages.

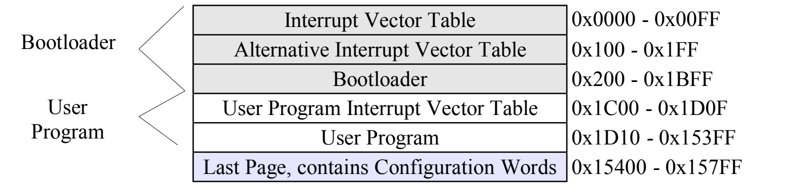
11.6.3 Microcontroller Program Memory

The Microcontroller has 128K Bytes of FLASH program memory, as shown in Figure 5. It is implemented as 16-bit Words, with only the lower 8 bits of each upper (odd address) Word implemented. It is organized into 1K Word (0x400) erasable blocks.

The lower 6K Words (0x1BFF x 1.5 = 9K Bytes) is reserved for the bootloader, and is write protected. It contains a USB bootloader, Flash Firmware upgrade manager, and Configuration Manager.

The last 1K Words (0x155FC - end) contains the Configuration Words, and is write protected.

The remaining space from 0x1C00 to 0x153FF (168K Words = 1.5 x 168K = 252K Bytes) contains the Firmware. The lower part of this (0x1C00 - 0x1D0F) contains the Interrupt Vector Table. The rest contains the Firmware.



Bootloader	Interrupt Vector Table	0x0000 - 0x00FF
	Alternative Interrupt Vector Table	0x100 - 0x1FF
	Bootloader	0x200 - 0x1BFF
User Program	User Program Interrupt Vector Table	0x1C00 - 0x1D0F
	User Program	0x1D10 - 0x153FF
	Last Page, contains Configuration Words	0x15400 - 0x157FF

Figure 5 - Microcontroller Program Memory Map, **Word** addresses shown.

12 Commands

The SBC66ECL currently only accepts a very limited number of commands via it's USB port. This will be expanded in future releases. Commands can be sent to the SBC66ECL via the following methods (now and **future releases**):

- USB Port. Commands can be send, and data requested via USB HID messages. The free *Netcruzer USB Terminal* app can be used. For details, see netcruzer.com/usbterminal. Full source code for the *Netcruzer USB Terminal* is also available, and can be modified if required.
- UDP or TCP message via a Ethernet Network - **Not available yet, in future version!**
- Web pages, via the HTTP GET command. Can be sent via various HTML elements, like button, checkboxes, radio-buttons and more - **Not available yet, in future version!**
- Serial ports. **Not available yet, in future version!** This enables commands to be sent via different protocols, like RS-232, RS-485, TTL Serial port and others.

All commands follow the syntax of name=value.

For example, to set port pin Y2, we can:

- Send the following HTTP GET command from a web page: `http://10.1.0.1/?py2=1` (**Not available yet, in future version!)**
- Send the following UDP message to the UDP Command Port: `py2=1` (**Not available yet, in future version!)**

In this example the name part is "py2" and the value part is “1”.

12.1 Defined Commands

For an up to date list of commands, see online documentation at netcruzer.com/nz/doc/proj_webserver_cmd

12.1.1 General Command

The General command has the following syntax: "`lx(o)=v`"

- The 'l' part is the *Command Group*, and indicates this is a *General Command*.
- The 'x' part is the *Command Code*, and specifies the command that is to be executed.
- The (o) part is an optional parameter that might be present. Not required, can be omitted.
  - (o,p) Indicates an optional parameter that must have a value as given by o or p. Not required, can be omitted
  - (o-p) Indicates an optional parameter that must have a value from o to p. Not required, can be omitted
- The 'v' part is the command operand, and is command specific.

The following General commands are implemented by the Netcruzer Webserver.

Command Syntax	*Description*

`lk(1-4)=str`	**Write Byte Array to LCD Display** - write given bytes to a LCD2S type serial display. The 'lk' command can include the LCD display address, which has to be a value from 1-4. If omitted, 1 is used. The address is set by a 2-position DIP switch on the back of the LCD2S display. A '_' escape character can be used to send hex values. The two bytes following the escape character gives the hex code. See LCD2S documentation for list of all commands, and their syntax. For example, the syntax for the LCD brightness command is *"0x81, value"*, where *value* is the new brightness value in hex from 00-FF (decimal 0-255). To set the brightness to a hex value of A0 (160 decimal), the following command can be send:
	`"lk=_81_A0"`
	To instruct the LCD display to remember the brightness setting after the power is removed, the command above must be preceded by a "Remember Command" (0x8D). Commands are separated by a '&' character. The syntax for sending a remember command, followed by the set brightness command shown above will be:
	`"lk=_8D&lk=_81_A0"`
	
	To include a '_' character, two '_' characters must be send after each other.

`ll(1-4)=str`	**Write String to LCD Display** - writes the given string to the LCD display. The 'll' command can include the LCD display address, which has to be a value from 1-4. If omitted, 1 is used. The address is set by a 2-position DIP switch on the back of the LCD2S display. A '_' escape character can be used to include one of the following LCD string escape sequences:
	_n = Go to beginning of next line.
	_f = Clear display and go to beginning of first line.
	_r = Go to beginning of first line
	_b = Cursor left
	_t = Cursor Right For example:
	
	Clear display, go to the beginning of the first line and write "Hello":
	`"ll=_fHello"`
	
	Clear display, write "Hello" on first line, and "world" on second line:
	`"ll=_fHello_nworld"`
	
	To include a '_' character, two '_' characters must be send after each other.

Examples:

The following example will write "Hello World" to the LCD2S display with address 1 (both positions of DIP switch set to off). First LCD is cleared(_f), then "Hello" is written to beginning of first line, then "World" is written to beginning of second line(_n):

```
ll=_fHello_nWorld
```

The following example will write "Hello World" to the LCD2S display with address 4 (both positions of DIP switch set to on):

```
l14=_fHello_nWorld
```

13 Specifications

13.1 Absolute Maximum Ratings

Item	Symbol	Min	Typ	Max	Unit
Operating Temperature:	Top	-40	85		°C

13.2 Electrical Characteristics

Item	Symbol	Condition	Min	Typ	Max	Unit
DC Supply Voltage:	Vdd	-	4.75	5.0	5.25	V
Typical Operating Current at 12V supply	Idd	Vdd = 12V		50		mA
Typical Operating Current at 24V supply	Idd	Vdd = 24V		25		mA

The RJ45 connector (Ethernet connector) meets IEEE 802.3 standards and FCC mechanical requirements.

13.3 D.C. Characteristics of user I/O pins on Daughter Board connector.

Item	Symbol	Condition	Min	Typ	Max	Unit
Input Low Voltage - TTL input:	VIL		0		0.5	V
Input Low Voltage - Schmitt Trigger input:	VIL		0		0.66	V
Input High Voltage - TTL input:	VIH		1.63		5.5	V
Input High Voltage - TTL input: with Analog Functions	VIH		1.63		3.3	V
Input High Voltage - Schmitt Trigger input:	VIH		2.64		5.5	V
Input High Voltage - Schmitt Trigger input with Analog Functions	VIH		2.64		3.3	V
I/O Port Output High Voltage:	VOH	IOH = 3.0mA	3.0			V
I/O Port Output High Voltage:	VOH	IOH = 6.0mA	2.4			V
I/O Port Output Low Voltage:	VOL	IOL = 6.6mA			0.4	V
Capacitive loading:	CIO			50		pF
Maximum output current sunk by any I/O pin				25		mA
Maximum output current sourced by any I/O pin				25		mA
Maximum current sunk by all ports				200		mA
Maximum current sourced by all ports				200		mA

14 Dimensions

The SBC66ECL dimensions are shown below.

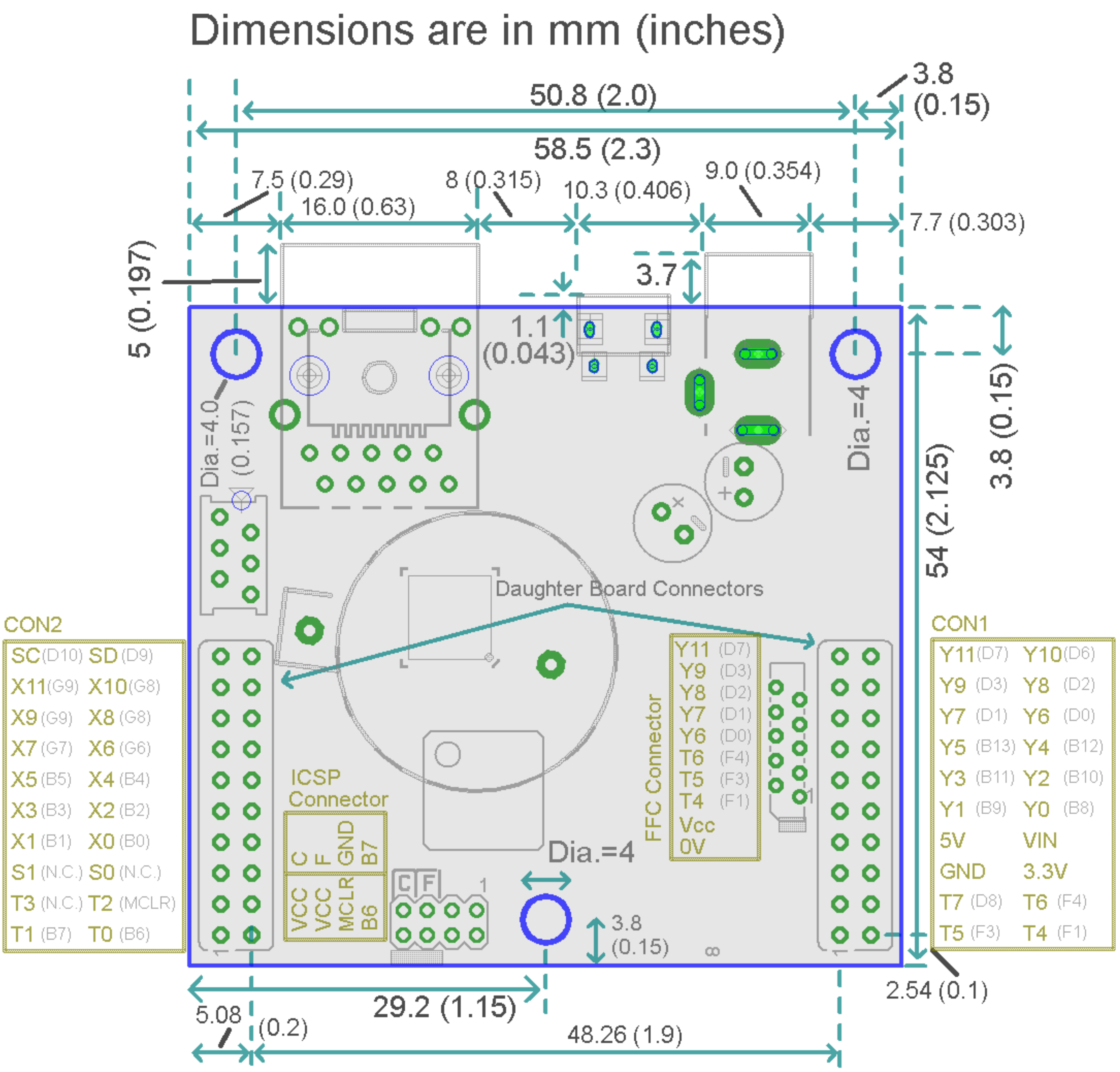
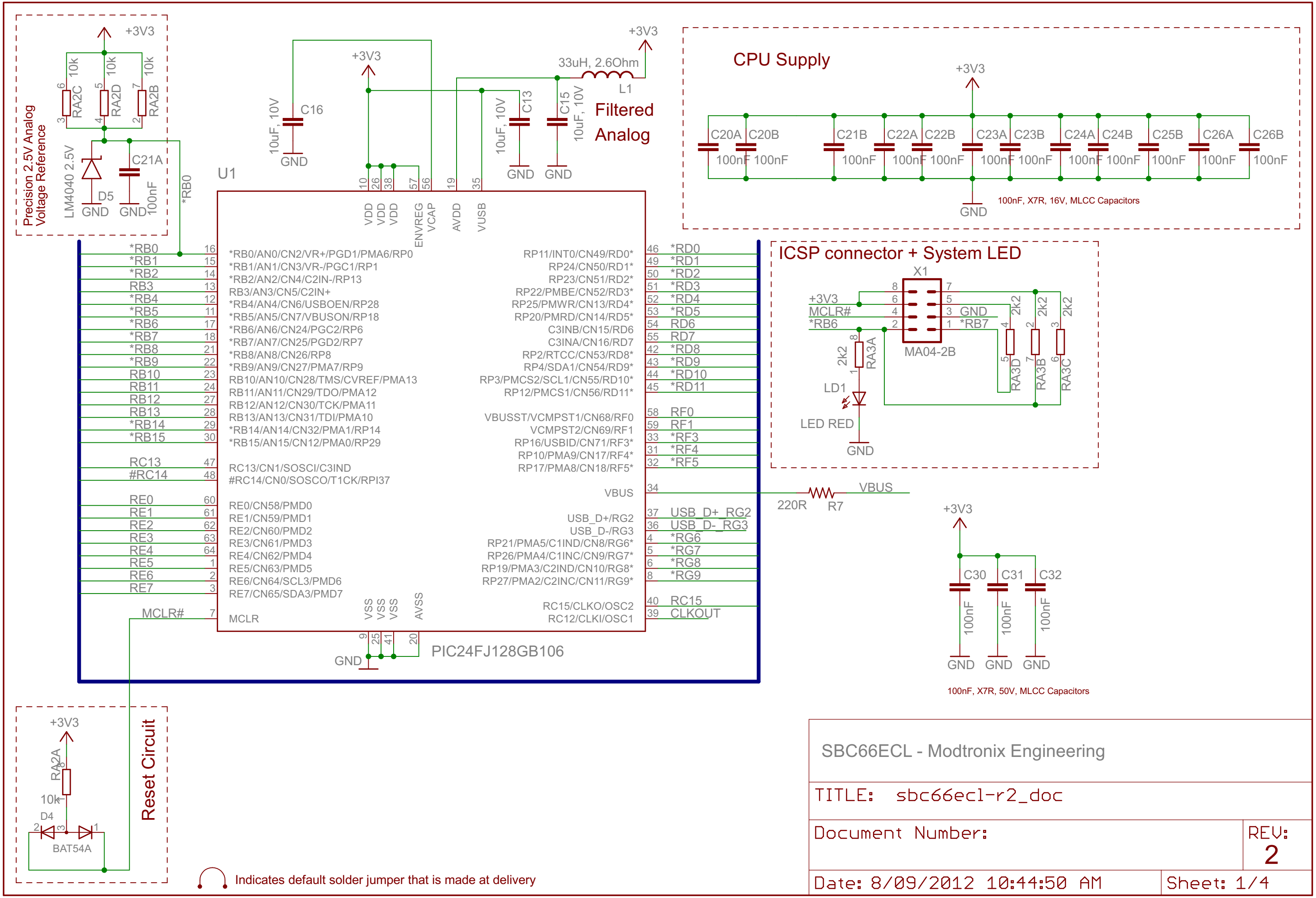


Figure 6

15 Schematics


Precision 2.5V Analog Voltage Reference
+3V3
10k
10k
10k
RA2C
RA2D
RA2B
LM4040 2.5V
D5
C21A
100nF
GND
GND
*RB0
U1
C16
10uF, 10V
GND
+3V3
C13
10uF, 10V
C15
10uF, 10V
GND
GND
33uH, 2.6Ohm
L1
+3V3
Filtered
Analog
VDD
VDD
VDD
ENVREG
VCAP
AVDD
VUSB
10
26
38
57
56
19
35
*RB0 16 *RB0/AN0/CN2/VR+/PGD1/PMA6/RP0
*RB1 15 *RB1/AN1/CN3/VR-/PGC1/RP1
*RB2 14 *RB2/AN2/CN4/C2IN-/RP13
RB3 13 RB3/AN3/CN5/C2IN+
*RB4 12 *RB4/AN4/CN6/USBOEN/RP28
*RB5 11 *RB5/AN5/CN7/VBUSON/RP18
*RB6 17 *RB6/AN6/CN24/PGC2/RP6
*RB7 18 *RB7/AN7/CN25/PGD2/RP7
*RB8 21 *RB8/AN8/CN26/RP8
*RB9 22 *RB9/AN9/CN27/PMA7/RP9
RB10 23 RB10/AN10/CN28/TMS/CVREF/PMA13
RB11 24 RB11/AN11/CN29/TDO/PMA12
RB12 27 RB12/AN12/CN30/TCK/PMA11
RB13 28 RB13/AN13/CN31/TDI/PMA10
*RB14 29 *RB14/AN14/CN32/PMA1/RP14
*RB15 30 *RB15/AN15/CN12/PMA0/RP29
RC13 47 RC13/CN1/SOSCI/C3IND
#RC14 48 #RC14/CN0/SOSCO/T1CK/RPI37
RE0 60 RE0/CN58/PMD0
RE1 61 RE1/CN59/PMD1
RE2 62 RE2/CN60/PMD2
RE3 63 RE3/CN61/PMD3
RE4 64 RE4/CN62/PMD4
RE5 1 RE5/CN63/PMD5
RE6 2 RE6/CN64/SCL3/PMD6
RE7 3 RE7/CN65/SDA3/PMD7
MCLR# 7 MCLR
RP11/INT0/CN49/RD0* 46 *RD0
RP24/CN50/RD1* 49 *RD1
RP23/CN51/RD2* 50 *RD2
RP22/PMBE/CN52/RD3* 51 *RD3
RP25/PMWR/CN13/RD4* 52 *RD4
RP20/PMRD/CN14/RD5* 53 *RD5
C3INB/CN15/RD6 54 RD6
C3INA/CN16/RD7 55 RD7
RP2/RTCC/CN53/RD8* 42 *RD8
RP4/SDA1/CN54/RD9* 43 *RD9
RP3/PMCS2/SCL1/CN55/RD10* 44 *RD10
RP12/PMCS1/CN56/RD11* 45 *RD11
VBUSST/VCMPST1/CN68/RF0 58 RF0
VCMPST2/CN69/RF1 59 RF1
RP16/USBID/CN71/RF3* 33 *RF3
RP10/PMA9/CN17/RF4* 31 *RF4
RP17/PMA8/CN18/RF5* 32 *RF5
VBUS 34
USB_D+/RG2 37 USB_D+_RG2
USB_D-/RG3 36 USB_D-_RG3
RP21/PMA5/C1IND/CN8/RG6* 4 *RG6
RP26/PMA4/C1INC/CN9/RG7* 5 *RG7
RP19/PMA3/C2IND/CN10/RG8* 6 *RG8
RP27/PMA2/C2INC/CN11/RG9* 8 *RG9
RC15/CLKO/OSC2 40 RC15
RC12/CLKI/OSC1 39 CLKOUT
VSS
VSS
VSS
AVSS
9
25
41
20
GND
PIC24FJ128GB106
220R
R7
VBUS
CPU Supply
+3V3
C20A
C20B
C21B
C22A
C22B
C23A
C23B
C24A
C24B
C25B
C26A
C26B
100nF
GND
100nF, X7R, 16V, MLCC Capacitors
ICSP connector + System LED
X1
+3V3
MCLR#
*RB6
8
6
4
2
7
5
3 GND
1 *RB7
MA04-2B
2k2
2k2
2k2
2k2
RA3A
RA3D
RA3B
RA3C
LD1
LED RED
GND
+3V3
C30
C31
C32
100nF
100nF
100nF
GND
GND
GND
100nF, X7R, 50V, MLCC Capacitors
+3V3
RA2A
10k
D4
BAT54A
Reset Circuit

Indicates default solder jumper that is made at delivery

SBC66ECL - Modtronix Engineering

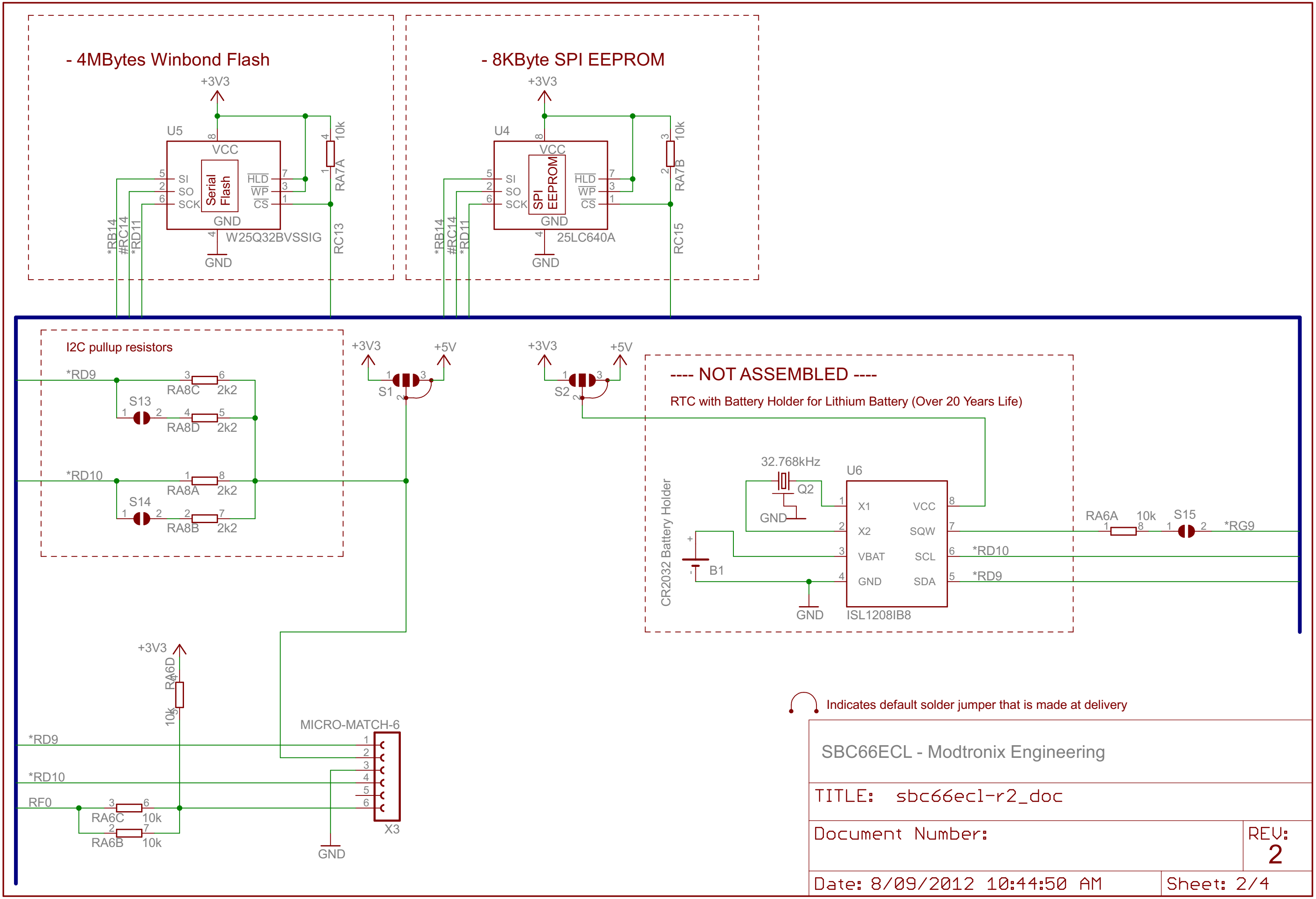
TITLE: sbc66ecl-r2_doc

Document Number:

REV: 2

Date: 8/09/2012 10:44:50 AM

Sheet: 1/4



SBC66ECL - Modtronix Engineering

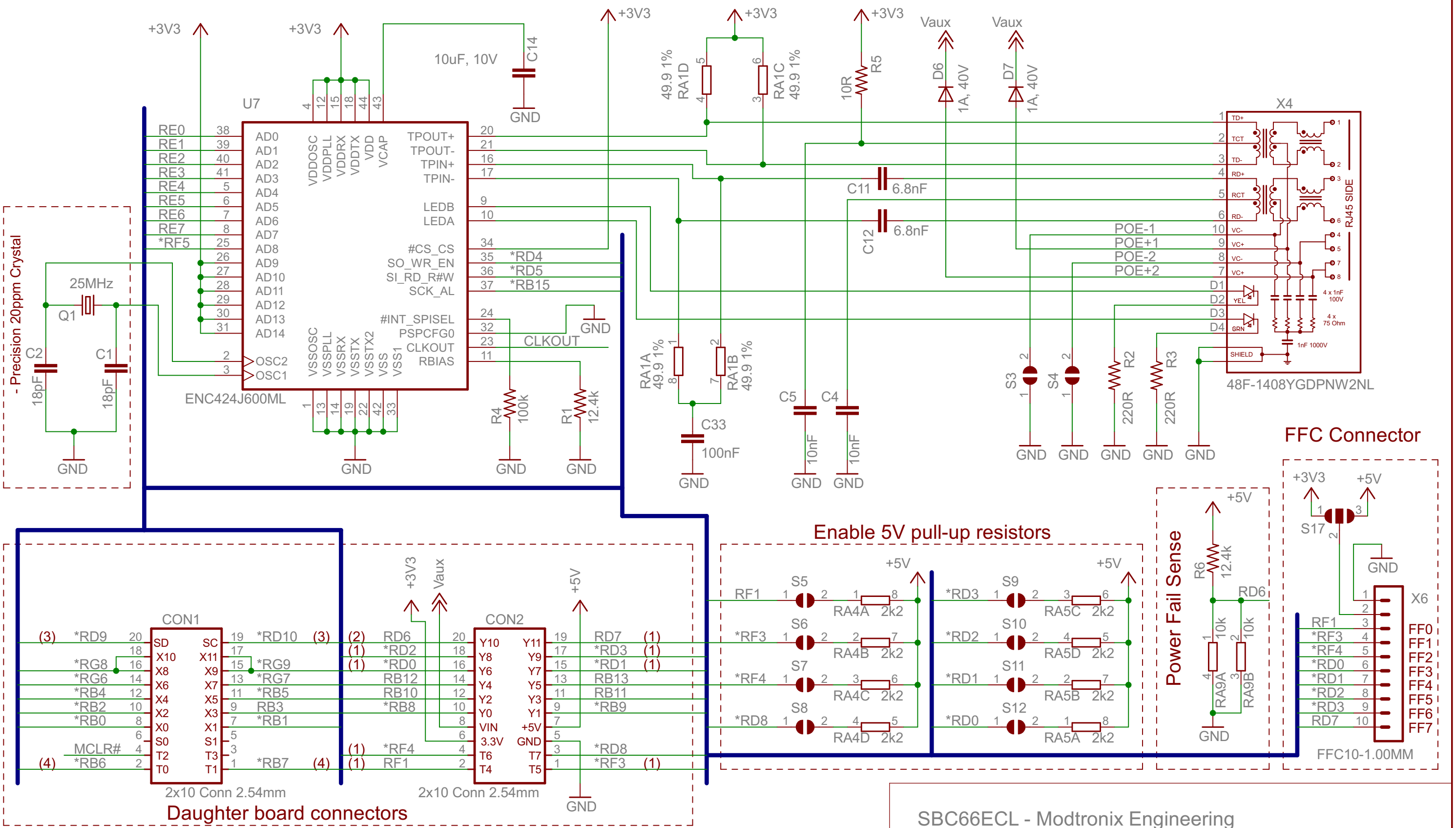
TITLE: sbc66ecl-r2_doc

Document Number:

REV: 2

Date: 8/09/2012 10:44:50 AM

Sheet: 2/4



(1) These pins are shared with the FFC Connector!
(2) RD6 Shared with Power Fail Sense!
(3) RD9 is I2C SDA, and RD10 is I2C SCK
(4) RB6 and RB7 are programming pins, do not use!

Indicates default solder jumper that is made at delivery

- 5V compatible = F1, F3, F4, D0-D3, D8

SBC66ECL - Modtronix Engineering

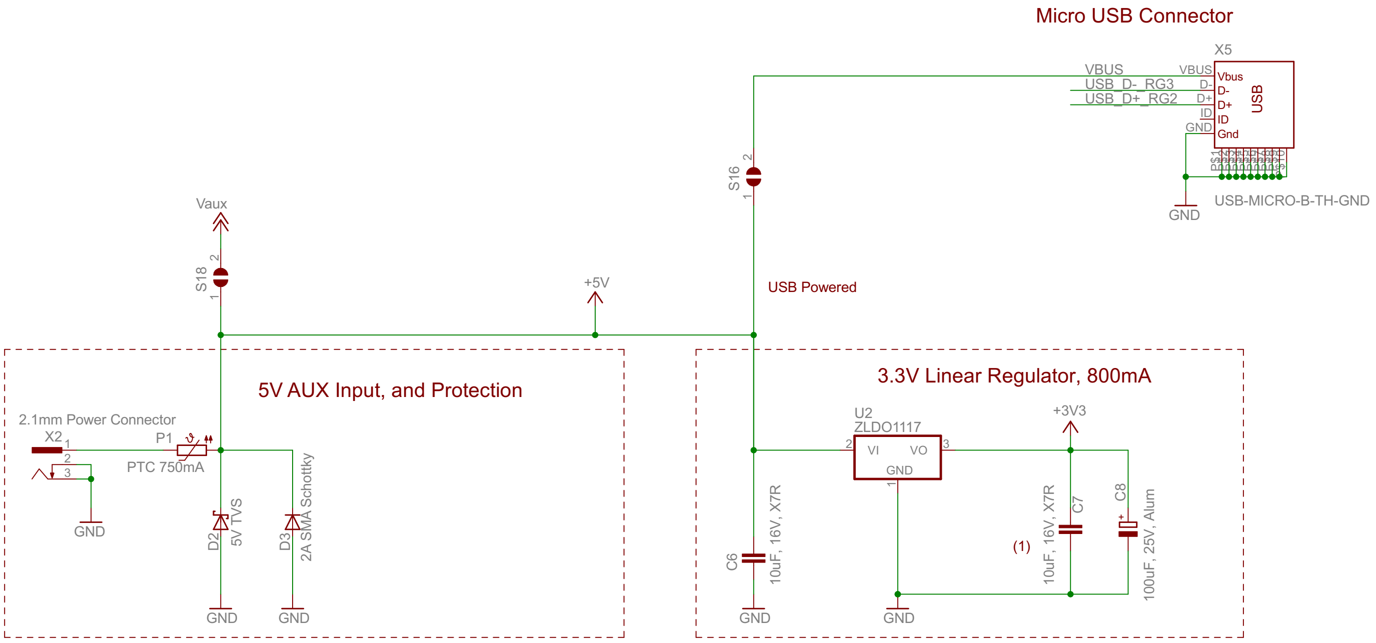
TITLE: sbc66ecl-r2_doc

Document Number:

REV: 2

Date: 8/09/2012 10:44:50 AM

Sheet: 3/4



SBC66ECL - Modtronix Engineering

TITLE: sbc66ecl-r2_doc

Document Number:

REV: 2

Date: 8/09/2012 10:44:50 AM

Sheet: 4/4