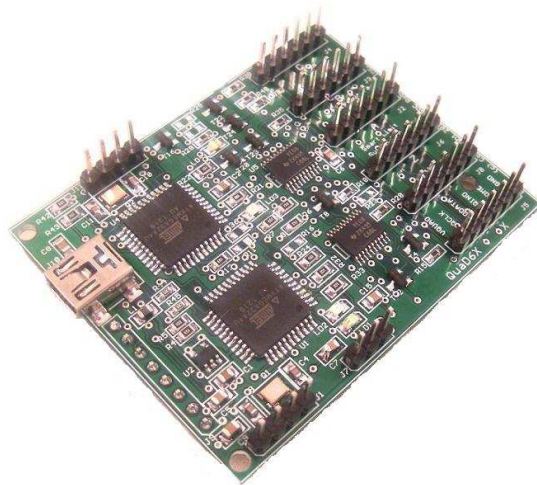


Quad6X 6 Axis Quadrature Decoder

Linear and Rotary Sensors
Technical Reference Manual
PCB Rev 1.0



www.soc-robotics.inc

Warranty Statement

SOC Robotics warrants that the Product delivered hereunder shall conform to the applicable SOC Robotics Data Sheet or mutually agreed upon specifications and shall be free from defects in material and workmanship under normal use and service for a period of 30 days from the applicable date of invoice. Products that are “samples”, “design verification units”, and/or “prototypes” are sold “AS IS,” “WITH ALL FAULTS,” and without a warranty. If, during such warranty period, (i) SOC Robotics is notified promptly in writing upon discovery of any defect in the goods, including a detailed description of such defect; (ii) such goods are returned to SOC Robotics, DDP SOC Robotics facility accompanied by SOC Robotics Returned Material Authorization form; and (iii) SOC Robotics examination of such goods discloses to SOC Robotics satisfaction that such goods are defective and such defects are not caused by accident, abuse, misuse, neglect, alteration, improper installation, repair, improper testing, or use contrary to any instructions issued by SOC Robotics. SOC Robotics shall (at its sole option) either repair, replace, or credit Buyer the purchase price of such goods. No goods may be returned to SOC Robotics without SOC Robotics Returned Material Authorization form. Prior to any return of goods by Buyer pursuant to this Section, Buyer shall afford SOC Robotics the opportunity to inspect such goods at Buyer’s location, and any such goods so inspected shall not be returned to SOC Robotics without its prior written consent. SOC Robotics shall return any goods repaired or replaced under this warranty to Buyer transportation prepaid, and reimburse Buyer for the transportation charges paid by Buyer for such goods. The performance of this warranty does not extend the warranty period for any goods beyond that period applicable to the goods originally delivered.

THE FOREGOING WARRANTY CONSTITUTES SOC ROBOTICS EXCLUSIVE LIABILITY, AND THE EXCLUSIVE REMEDY OF BUYER, FOR ANY BREACH OF ANY WARRANTY OR OTHER NONCONFORMITY OF THE GOODS COVERED BY THIS AGREEMENT. THIS WARRANTY IS EXCLUSIVE, AND IN LIEU OF ALL OTHER WARRANTIES. SOC ROBOTICS MAKES NO OTHER WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING WITHOUT LIMITATION ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THE SOLE AND EXCLUSIVE REMEDY FOR ANY BREACH OF THIS WARRANTY SHALL BE AS EXPRESSLY PROVIDED HEREIN.

Limitation on Liability

Notwithstanding anything to the contrary contained herein, SOC Robotics shall not, under any circumstances, be liable to Buyer or any third parties for consequential, incidental, indirect, exemplary, special, or other damages. SOC Robotics total liability shall not exceed the total amount paid by Buyer or SOC Robotics hereunder. SOC Robotics shall not under any circumstances be liable for excess costs of re-procurement.

© Copyright 2014. SOC Robotics, Inc. All rights reserved.

SOC Robotics, Inc. makes no warranty for the use of its products, other than those expressly contained in the Company’s standard warranty that is detailed in SOC Robotics Terms and Conditions located on the Company’s web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. The Company in connection with the sale of SOC Robotics products grants no licenses to patents or other intellectual property of SOC Robotics, expressly or by implication. SOC Robotics products are not authorized for use as critical components in life support devices or systems.

Pentium is a registered trademark of Intel Corporation.

Windows, Windows NT and Windows XP are registered trademarks of Microsoft Corporation.

Marks bearing ® and/or ™ are trademarks of SOC Robotics, Inc.

Terms and product names in this document may be trademarks of others.

1935A-08/00/5M

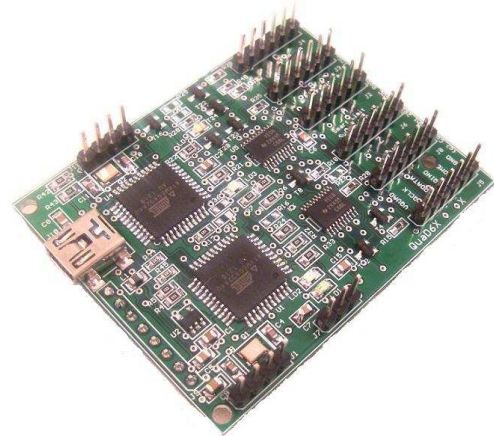
Table of Contents

Warranty Statement	2
1.0 Introduction	4
2.0 Detailed Description	5
2.1 Introduction.....	5
2.2 Hardware Configuration.....	5
2.3 Software Operation.....	7
2.4.1 DFU Configuration.....	8
2.4.2 Reprogramming the Processors.....	8
3.0 Electrical and Mechanical Description	9
3.1 Component Layout.....	9
3.2 Electrical Specifications.....	9
3.3 Mechanical Dimensions	9
4.0 Circuit Schematics	12

1.0 Introduction

Features:

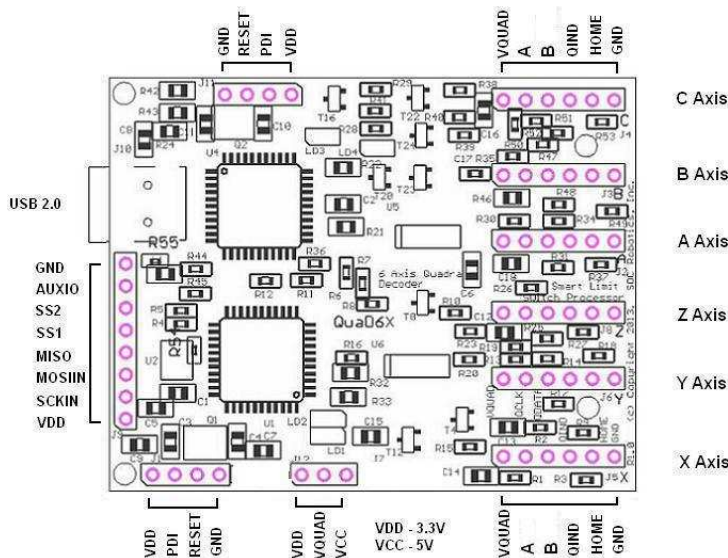
- 6 Axis linear/rotary quadrature decoder
- 5V or 3.3V sensor operation
- USB 2.0 serial interface
- SPI interface for each processor
- Green/Red Status LEDs for each processor
- Two ATxmega64A4U processors
- Small form factor (1.75x2.40 in)
- 5VDC @ 35ma Power input for logic
- Board logic runs at 3.3V
- Level shifter converts 5V sensor data to 3.3V
- Jumper sets sensor voltage – 3.3V or 5V



Hardware

The Quad6X is a compact six Axis quadrature decoding device with USB interface, smart limit switch inputs on each axis and selection of sensor voltage (3.3V-5V).

Quad6X Connector Pin Assignment



Configure Hardware

Sensors require either 5V or 3.3V input power. Installing a jumper on connector header J7 sets the correct voltage. VDD is 3.3V and VCC is 5V. All sensors are set to the same selected voltage. A voltage level shifter converts the input sensor voltage to 3.3V.

Software

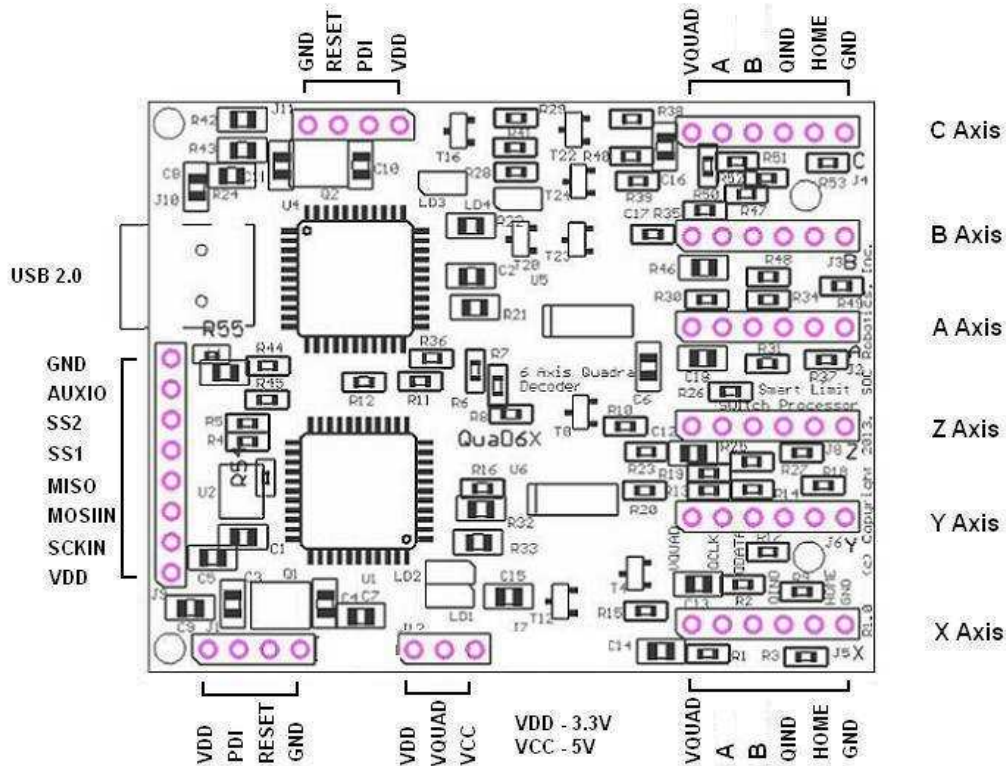
The Quad6X is configured by setting a number of parameters for linear or rotary operation, data format, stream vs query/response and limit switch operation. See the software section below for more detail.

2.0 Quad6X Detailed Description

2.1 Introduction

The Quad6X is a compact 6 Axis Quadrature decoding device with USB interface and smart limit switch input on each axis.

Quad6X Connector Pin Assignment



2.2 Hardware Configuration

There is a three-pin header that sets the voltage level on the quadrature input port to either 5V or 3.3V. Install a shorting jumper between VDD and VQUAD for 3.3V and VCC and VQUAD for 5V operation. Glass linear sensors require 5V while SOC Robotics magnetic sensors use 3.3V.

2.3 Software Operation

The USB interface runs as a serial CDC device. An ASCII command set allows a host to configure each axis for resolution and set access method. Both continuous stream and query/response are supported. In stream mode the current position of each axis is output in engineering units or raw binary (converted to ASCII) at a user defined rate. In query response mode selected axis position data is output for each query. In stream mode data is output continuously based on the axis enabled. Configuration parameters set basic operational modes such as linear verse rotary decoding, resolution conversion and output in Imperial, metric or raw format.

The quadrature capture subsystem is hardware based and is capable of tracking quadrature signals are very high rates. Returned converted data to the desktop via the USB interface is slower than the capture

rate so the desktop software must consider delays. The hardware capture subsystem, however, does not lose increments.

Position is output in metric, Imperial or raw binary units. A correction factor can be applied to fine tune the resolution of each quadrature binary conversion. The index signal is tracked for correct roll over.

Linear sensors are typically set to an incremental length of 65535 (maximum size of the dedicated hardware capture register counters) with index reset disabled. Software keeps track of transitions (both plus and minus) ensuring the correct position is output by the decoder. Setting the axis position to zero also zeros the quadrature counter to zero.

Rotary sensors have two settings – pole increment length and number of poles. The two numbers are used to set total increments per 360 degrees of rotation. This number is tracked by dedicated hardware on the board. Although it is possible to have the index pulse reset the increment count on each transition it is best to allow the hardware to keep track of the position on the circle. Setting the position to zero sets the increment count to zero which also sets angle position 0.00. This allows control software to set the initial position of the rotary sensor.

The Quad6X also has a SPI interface for more direct embedded processor control such as a Raspberry PI, BeagleBone or Arduino. The command protocol is similar to the USB serial protocol.

Serial Commands (all commands are terminated with a CR character):

- i - Return sign-on message
- d - Dump configuration parameters
- u - Update system configuration parameters
- p6 - Get current position of all six axis once
- ph - Get current position on axis h where h is x, y, z, a, b or c
- z6 - Zero all axis
- zh - Zero axis h where h is x, y, z, a, b or c
- gr6 - Get all six axis resolutions
- ga - Get resolution of axis a where a is x, y, z, a, b or c
- sr6x.xxxxxxx - Set all six axis to resolution x.xxxxxxx (x.key format supported)
- srax.xxxxxxx - Set axis a to resolution x.xxxxxxx for axis a where a is x, y, z, a, b or c
- ss - Enable stream mode – any sent character after stream mode enabled disables stream mode
- sce - Set coordinate system to English units
- scm - Set coordinate system to metric units

Commands can also be sent with a preceding ESC character

- ESCi - Return device name and version

System Configuration Parameters

The Quad6X is capable of tracking either all linear or all rotary sensors. System parameters set linear or rotary mode on an individual basis. The index signal can be used to reset each axis's counter. There are ten parameters that can be used to configure settings for each axis. A configuration command consists of a number (0-9) an optional axis identifier (such as x or a) followed by the = character followed by either a single parameter or all six parameters. If numbers are entered separate using a space character otherwise single character settings are entered with no white space.

- 0h=x.xxxx-e-y - Set resolution for axis h where h is x, y, z, a, b or c
- 1=hhhhhh - Set index state for all six axis [o-on,f-off] default is all off
- 2=hhhhhh - Set sensor mode for all six axis [l-linear, r-rotary] default is linear
- 3=ddd ddd ddd ddd ddd ddd - Set six increment lengths as integer

- 4=ddd ddd ddd ddd ddd ddd – Set six number of poles (for rotary) as integer
- 5=hhhhhh – Set data output format [r-raw, i-imperial, m-metric] default imperial
- 6=hhhhhh – Set roll over mode [n-not active, a-active] default not active
- 7=hhhhhh – Set limit switch mode [n-not active, a-active] not implemented at this time
- 8=hhhhhh – Set Smart limit switch mode [n-not active, a-active] not implemented at this time
- 9=hhhhhh – Set stream mode [n-not active, a-active] default not active

Some functional limitations apply with the V1.00 software (current version is output with the “i” command). Note that data output in streaming mode is set on an individual axis basis so it is possible to stream a single axis.

Limitations

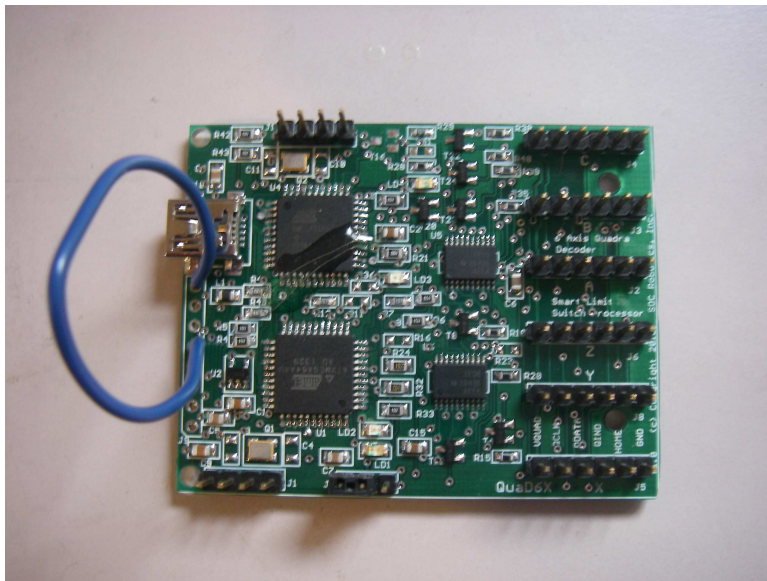
- Either all linear or all rotary sensors must be selected.
- If index mode is enabled for an axis then the dipole or optical sensor roll over increment number must be used in the increment setting.
- Limit switch handling is not implemented at this time
- Roll over mode not implemented

Most of these limitations will be removed in the next release of the software.

2.3.1 DFU Programming

There are two Atmel ATxmega64A4U processors on the Quad6X – each handles three quadrature axis. Software running on each processor is field upgradeable by using Atmel’s Flip programming utility (free download from Atmel’s web site at www.atmel.com). Only one processor can be programmed at a time. Although both processors have the ability to communicate to the desktop via USB only one can do so for correct operation. In order to program both processors through the USB interface using the DFU bootloader each must be configured for programming so the other does not interfere.

The first step in re-programming the board is to download Atmel’s Flip programming utility and install it. To re-program both processors each must be connected one at a time to the USB line and the DFU bootloader started.

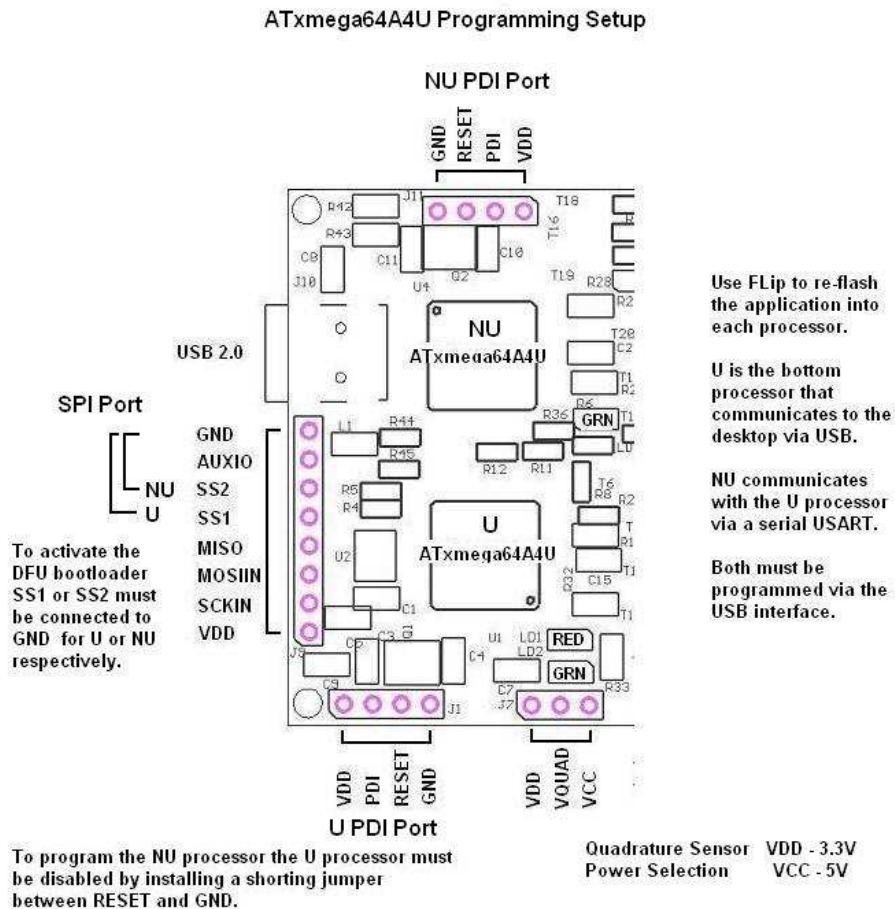


To program the processor near the bottom of the board (USB processor-U) install a jumper on connector J9 as shown above and re-power the board. This will start the DFU bootloader. Windows will ask you to install a driver which is located in Flip’s USB directory. Next follow the programming instructions below.

To program the other processor shift the jumper on J9 one position closer to the top, next install a jumper on the PDI programming port shorting RST to GND, (this disables one of the other processor) and re-power the board. Follow the programming procedure as before.

Note that each processor must be loaded with the correct hex file for proper operation. The processor closest to the USB connector is identified as processor NU and the other processor U. Each hex file is identified either as Quad6X_NU_Controller_V1.xx.hex or Quad6X_U_Controller_V1.xx.hex where xx is the current version number.

The diagram below shows the location of each processor, DFU activation pins and PDI programming ports.

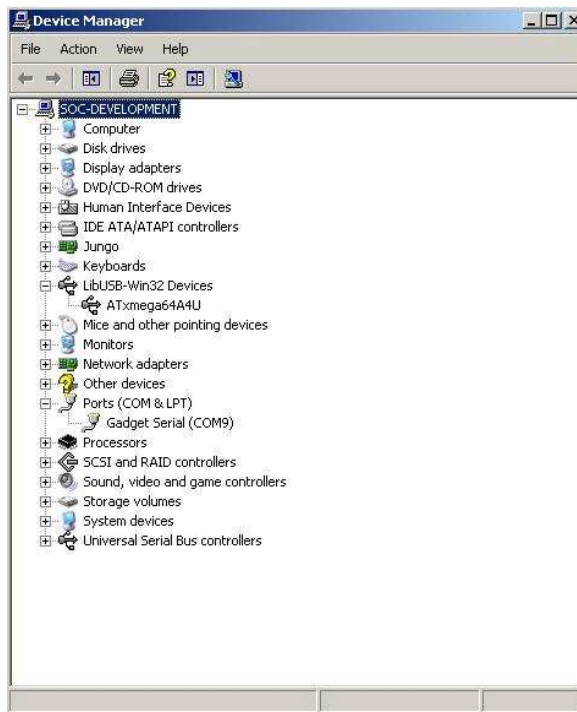


2.3.2 Programming the ATxmega64A4U

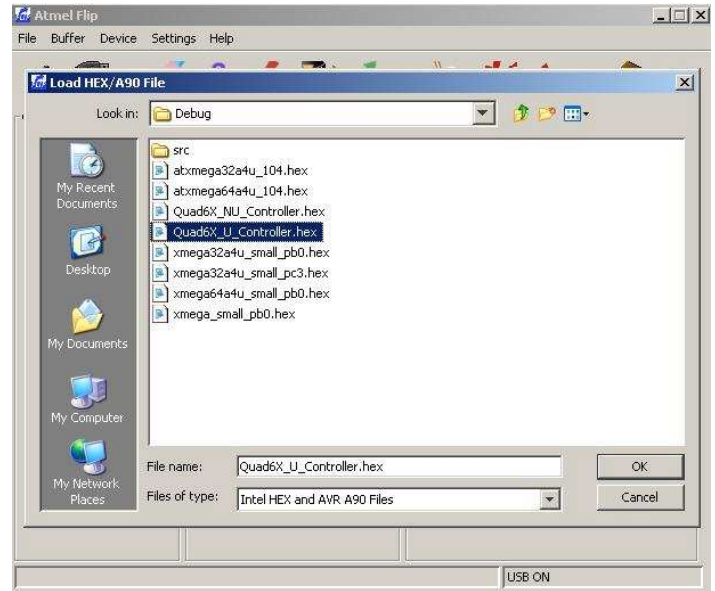
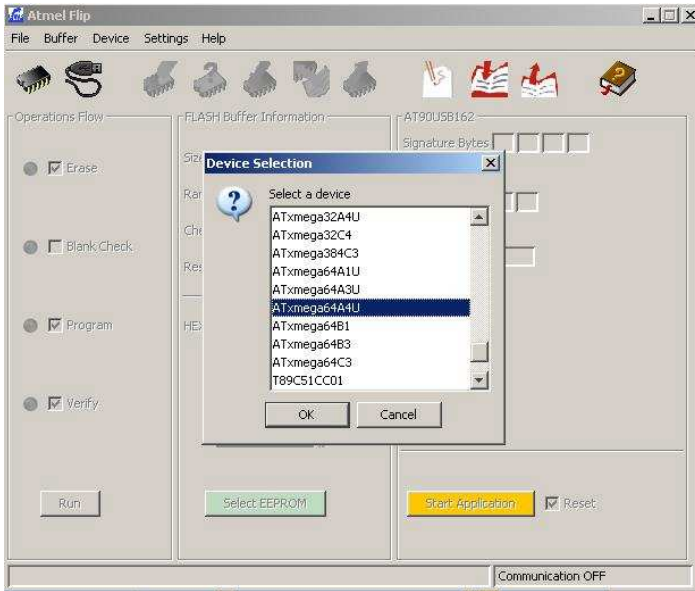
To program the ATxmega64A4U using the DFU bootloader follow the steps below. The Quadrature Decoding application running on the Quad6X is updated on a regular basis. Current version is 1.00. The

procedure below walks you through the process involved in re-flashing the latest version of the Quadrature Decoder Application.

- Download the Flip 3.4.7 programming utility from the Atmel web site
- Download the latest version of the Quad6X Setup zip folder from the SOC Robotics web site (www.soc-robotics.com). This setup folder contains the latest Quadrature Decoding application hex files, DFU inf and a serial CDC inf file.
- If you haven't powered the board yet do the following quick check to make sure the board is communicating with the Windows desktop:
 - o Attach a USB cable from the board to a Windows PC and load the serial driver called "atmel_devices_cdc.inf" file located in the Quad6X Prog folder if this hasn't been done yet to install the CDC Virtual COM driver
 - o Start a terminal app such as HyperTerminal and make sure Quad6X communicates with the desktop
- There are two processors on the Quad6X. To program the processor furthest from the USB connector (bottom processor in the picture above) install a shorting jumper between SS2 and Ground on SPI port J9 and re-power the board - this will start the DFU bootloader. If this is the first time the DFU bootloader has been started Windows will ask you to install a *.inf driver called atmel_usb_dfu.inf.
- The *.inf file for the DFU bootloader is located in the Flip utility subfolder called usb located under Atmel in Windows Program Files. If the DFU .inf file is properly installed it will show up in Device Manager as shown below.



- There are two hex files in the Quad6X Prog folder – one for the non-USB processor called Quad6X_NU_Controller_V1.xx.hex and the other for the USB processor called Quad6X_U_Controller_V1.xx.hex. Double click on the Flip icon to start the programming utility.
- Contact SOC for the latest version of hex files for the Quad6X or go to the download page at www.soc-robotics.com for the latest version.
- Once the Flip utility starts select the correct processor from the Device menu – ATxmega64A4U.



- Next select the USB processor hex file called Quad6X_NU_Controller_V1.xx.hex as shown above. Click the USB Icon to connect to the Quad6X bootloader and then click Run – this will start the programming process.



- To program the other processor connect SS2 to Ground on the SPI Port J9, connect RST and GND on the PDI programming connector of the bottom processor and repeat the programming process above but this time select hex file Quad6X_NU_Controller_V1.xx.hex.
-
- Now remove all the jumpers and repower the board – both quadrature apps will now start.

Re-Programming the ATxmega64A4U DFU Bootloader

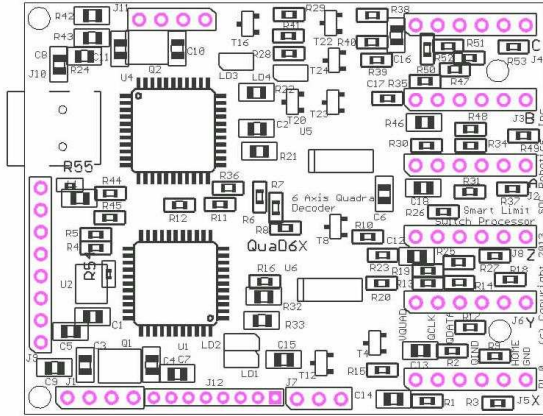
There are a few steps required to reload the DFU bootloader on Quad6X ATxmega64A4U processors. Special programming hardware is required to reload the bootloader – it cannot be done via the USB interface. A DRAGON programmer (amongst others) must be used in PDI mode along with AVR Studio 6.2 to re-program the bootloader. Additional technical information is available on the Atmel web site.

There is a four pin PDI programming port for each processor on the board identified as J1 and J11. These ports use an Atmel proprietary PDI programming method to load the DFU bootloader into the bootloader memory of the processor. The DFU bootloader file is called atxmega64a4u_dfu_pc6.hex and the same file is used for both processors. When processor port pin PC4 is pulled low at power up the bootloader starts. The PDI interfaces use VDD, RESET, TDI and GROUND.

3.0 Electrical and Mechanical Description

3.1 Component Layout

Components are mounted on the top side of the board. Not all components may be mounted.



3.2 Electrical Specifications

Electrical

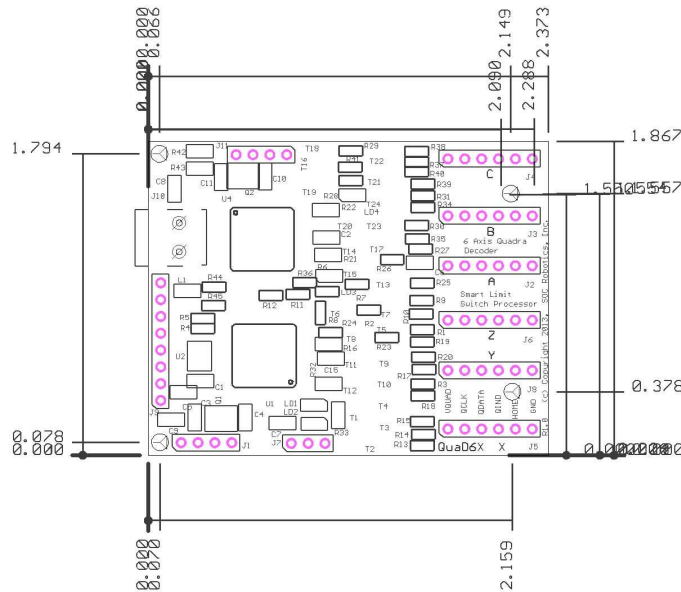
Input power: 5VDC @ 35ma

Mechanical

Dimensions: 2.40x1.90 in (one mounting hole)
 Weight: 12 grams

3.3 Mechanical Dimensions

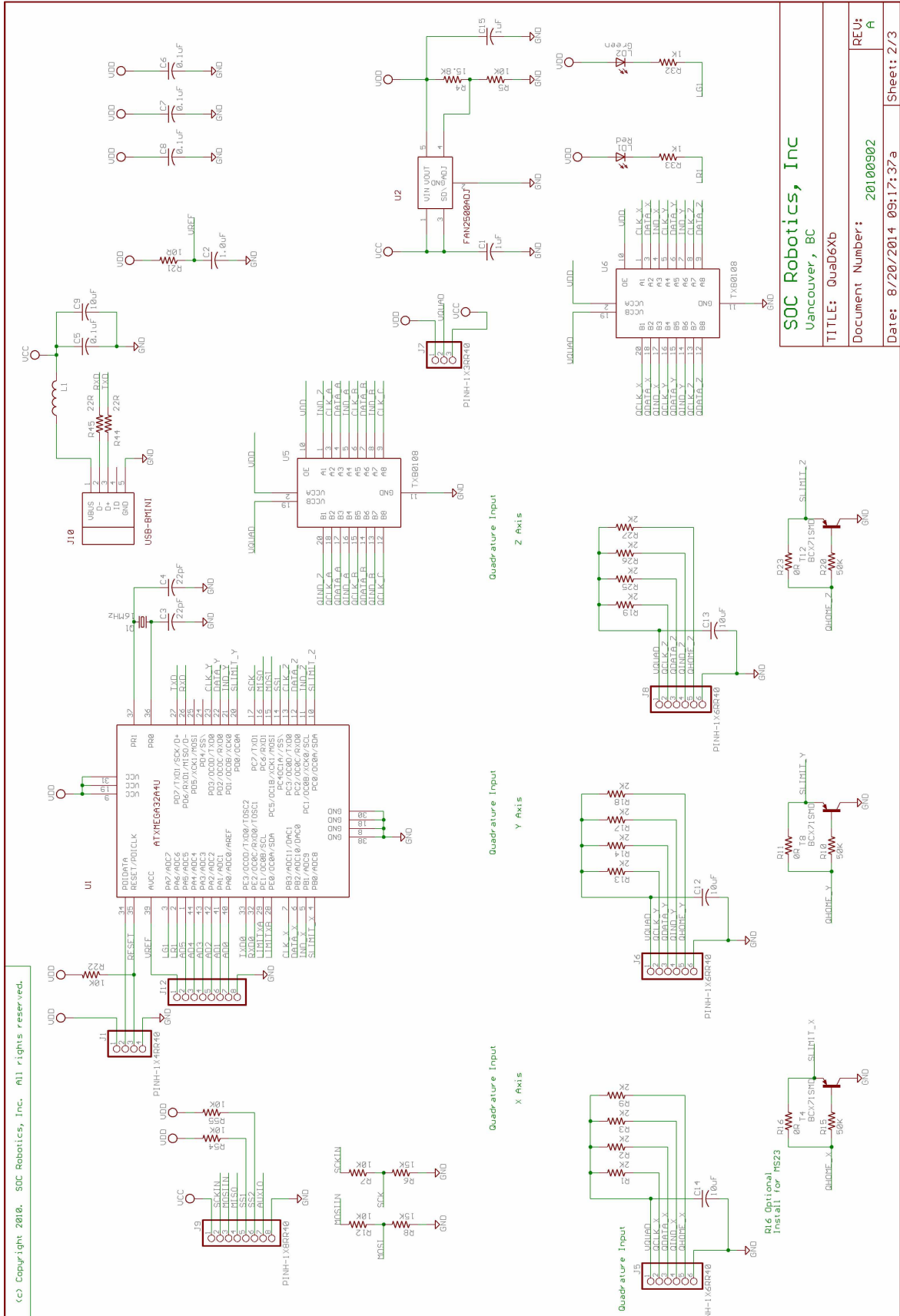
Board dimensions are stated in inches.



4.0 QUAD6X Circuit Schematics

<p style="font-size: 2em; text-align: center;">Quad6X</p> <p style="font-size: 3em; text-align: center;">6 Axis Quadrature Decoder</p> <p style="font-size: 1.5em; text-align: center;">PCB Rev 1.0</p>	SOC Robotics, Inc Vancouver, BC	
	TITLE: QuaD6Xb	
	Document Number: 20100902	REV: A
	Date: 8/20/2014 09:17:37a	Sheet: 1/3

(c) Copyright 2010. SOC Robotics, Inc. All rights reserved.



SOC Robotics, Inc
Vancouver, BC

TITLE: Quad6Xb

Document Number: 20100902

Date: 8/20/2014 09:17:37a

REU: A

Sheet: 2/3

Notes: