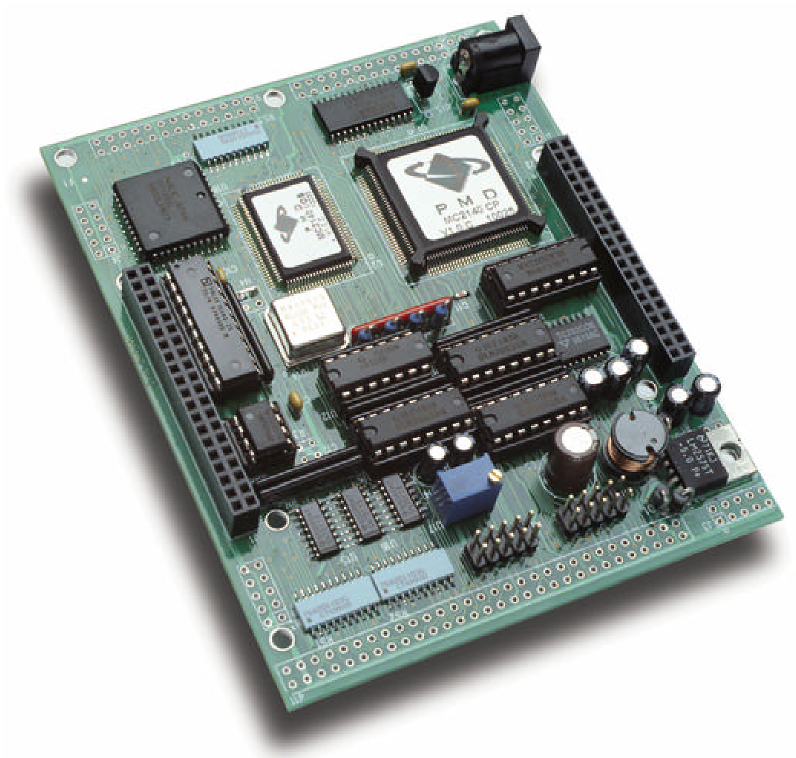
*MotionC*2140™

C/C++ Programmable, 40 MHz
Standalone DSP 2- or 4-axis Servo Motion Controller



Technical Manual

TERN INC.
1724 Picasso Avenue, Davis, CA 95616-0547, USA
Tel: 530-758-0180 Fax: 530-758-0181
Email: sales@tern.com *http://www.tern.com*

Version 2.00

February 21, 2003

1724 Picasso Avenue, Davis, CA 95616-0547, USA
Tel: 530-758-0180   Fax: 530-758-0181
Email: sales@tern.com   ***http://www.tern.com***

Important Notice

TERN is developing complex, high technology integration systems. These systems are integrated with software and hardware that are not 100% defect free. ***TERN products are not designed, intended, authorized, or warranted to be suitable for use in life-support applications, devices, or systems, or in other critical applications. TERN*** and the Buyer agree that ***TERN*** will not be liable for incidental or consequential damages arising from the use of ***TERN*** products. It is the Buyer's responsibility to protect life and property against incidental failure.

TERN reserves the right to make changes and improvements to its products without providing notice.

Table of Contents

Chapter 1: Introduction

1.1 Functional Description

The *MotionC*2140 is a low-cost, reliable, compact, high performance, C/C++ programmable industrial motion controller. It includes a DSP chipset (MC2140/2120, PMD) and it is driven by a host (x-*Engine*)*: A-Engine, A-Engine86, i386-Engine, or 586-Engine***.** The *MotionC*2140 is a complete, ready to run, motion controller with built in sophisticated field proven control firmware. User only needs to define parameters for PID algorithm and trajectory profile. The DSP calculates velocity, position and stabilizes the motor output. At the same time, the host *x-Engine* interfaces with a PC, monitors I/Os, and computes or pre-loads a new set of parameters.

The *x-Engine* interfaces to the DSP chipset via high-speed data bus. User can easily develop, download, and debug application programs via serial link to a PC. The host writes pre-defined motion commands to the DSP. The DSP can interrupt the host at any time.

The *MotionC*2140 provides protected inputs for home switches, limit switches, and fault switches via Darlington arrays which are capable of inputs up to +30V. Seven solenoid drivers are available and can sink up to 350mA at 50V. A PPI (82C55) provides 24 user-programmable bi-directional I/O lines. Two RS-232 and one RS-485 drivers can be installed.

The *MotionC*2140 supports up to 4-axis closed-loop digital servo controls. The digital servo control signals use incremental quadrature encoders for position inputs. The DAC outputs ±10V servo control signals. Each axis contains sophisticated trajectory profile and digital servo capabilities. The *MotionC*2140 provides electronic gearing, PID/PI control, a choice of S-curve, trapezoidal, or contoured velocity profile modes, automatic motor error shutdown, and monitoring of switches**.** The *MotionC*2140 supports a 16-bit parallel-word input mechanism, such as ADC. Expansion headers are available for 8 channels of 10-bit ADC inputs, as well as PWM output and trace memory expansion.

*MotionC*2140 provides:

- Electronic gearing
- PID or PI control
- Choice of S-curve, trapezoidal, or contoured velocity profile modes
- 1/T counter for stable low velocity motion
- Automatic motor error shutdown
- Monitoring travel limit switches
- Protected home, capture, and fault switches

The host x-*Engine* provides many options. See x-*Engine* manual for details.

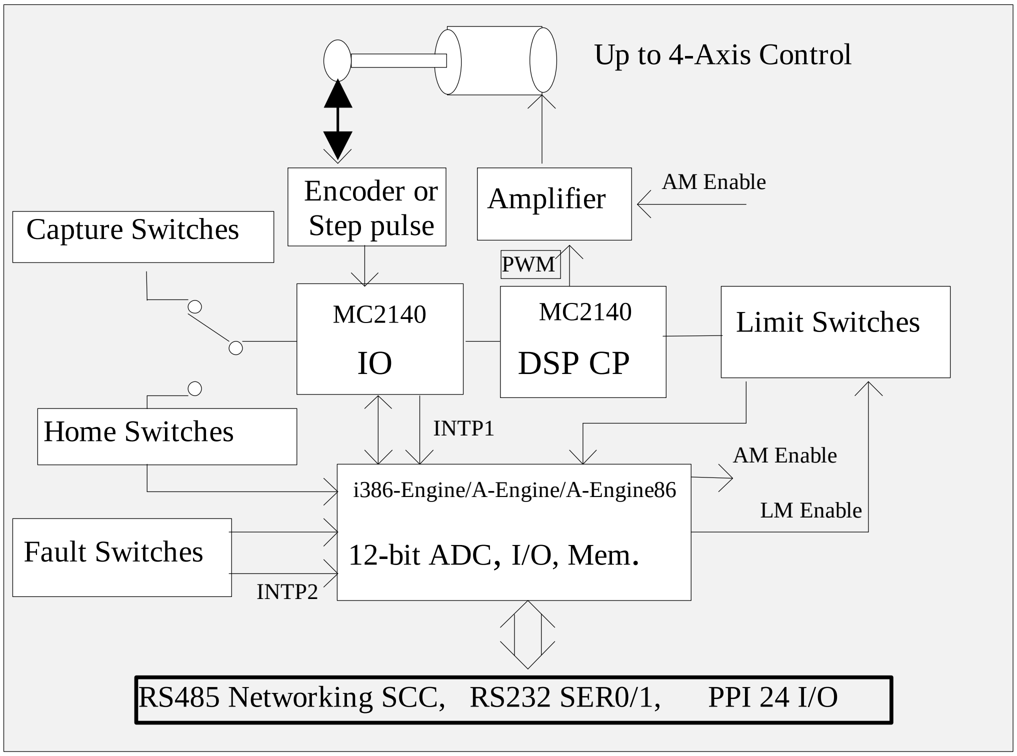


Figure 1.1 Functional block diagram of the MotionC 2140

Standard Features:

Dimensions: 4.65 x 3.75 inches (MC2140, 40 MHz)
Driven by an i386-Engine/A-Engine/A-Engine86/586-Engine (C/C++ programmable)
Power consumption: 200 mA at 12V
Temperature range: -40°C to +80°C
7 solenoid drivers, 24+ TTL I/Os
2 RS-232 drivers, 1 RS-485 driver (optional)
Protected switches for position, velocity, acceleration and jerk
32-bit registers for position, velocity, acceleration and jerk
S-curve, trapezoidal, or contoured velocity profile modes
Electronic gearing for multi-axis
1/T counter for stable low velocity motion
PID or PI control, Programmable loop rate to 100 microseconds

1.2 Physical Description

The physical layout of the *MotionC*2140 is shown in Figure 1.2. Dimensions are available in Appendix A.

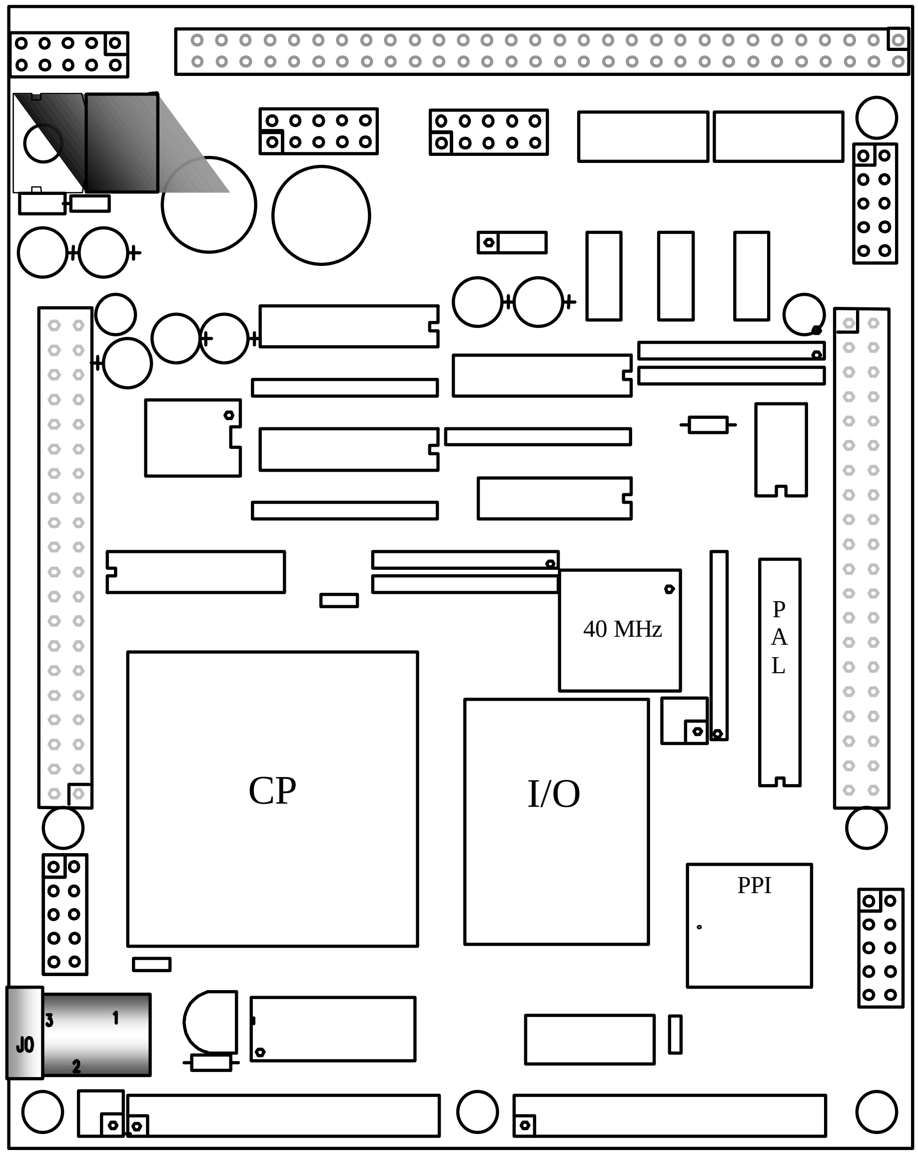


Figure 1.2 Physical layout of the *MotionC*2140

1.3 MotionC 2140 Programming Overview

Development of application software for the MotionC2140 consists of three easy steps, as shown in the block diagram below.

STEP 1 Serial link PC and *MotionC* + x-*Engine*.
Generate source code in C/C++ using samples on TERN CD.
Compile, Link, Locate, and Remote Debug source code with Paradigm C/C++ Environment.

STEP 2 Test *MotionC* + x-*Engine* in the field, away from PC.
Application program resides in the battery-backed SRAM.
Return to STEP 1 for source code adjustments, if necessary.

STEP 3 Create application BIN or Intel HEX file based on source code and Paradigm C/C++ DV-P Kit.
Create application ROM with BIN file or download Intel HEX file into ACTF Flash.
Project is complete.

There are three steps in the development of a TERN controller C/C++ application program. These steps are explained thoroughly in the Technical Manuals for the A-Engine/A-Engine86/i386-Engine/586-Engine. The EV-P Kit supports Step 1 and Step 2, but does not support STEP 3. Step 3 allows you to generate an Intel HEX or BIN file to produce your own ROM/Flash chip. The full Development version (DV-P) is required for STEP 3.

This technical manual is intended primarily to provide hardware support for your MC2140. The respective technical manuals for your host engine can provide additional details on the development of your application. See the tern_docs\manuals directory on your TERN CD. In addition, the tern_docs\parts directory contains complete technical specifications for the PMD MC2100 series chip sets.

1.3.1 Minimum Hardware Requirements

PC or PC-compatible computer with serial COMx port that supports 115,200 baud
MotionC 2140 controller (*MC*2140)
x-Engine host controller:

586-Engine with debug kernel 5860_115.hex loaded into on-board flash,
i386-Engine with DEBUG ROM ie8_0_115,
A-Engine with DEBUG ROM ae_0_115,
or, A-Engine86 with debug kernel ae86_115.hex loaded into on-board flash

Debug serial cable (RS-232; DB9 connector for PC COM port and IDE 2x5 connector for controller)
Center negative wall transformer (+9V 500 mA)

1.3.2 Minimum Software Requirements

TERN EV-P or DV-P software kit

PC software environment: Windows95/98/2000/NT/ME/XP

Chapter 2: Installation

2.1 Software Installation

Please refer to the Technical manual for the "C/C++ Development Kit and Evaluation Kit for TERN Embedded Microcontrollers" for installing software.

The README.TXT file on the root directory of the TERN EV-P/DV-P CD contains important information about the installation and evaluation of TERN controllers.

2.2 Hardware Installation

Hardware installation for the MotionC 2140 consists primarily of connecting the microcontroller to your PC and to power.

Overview

Install x-Engine controller to the MotionC 2140:
586-Engine, A-Engine, A-Engine86, or i386-Engine

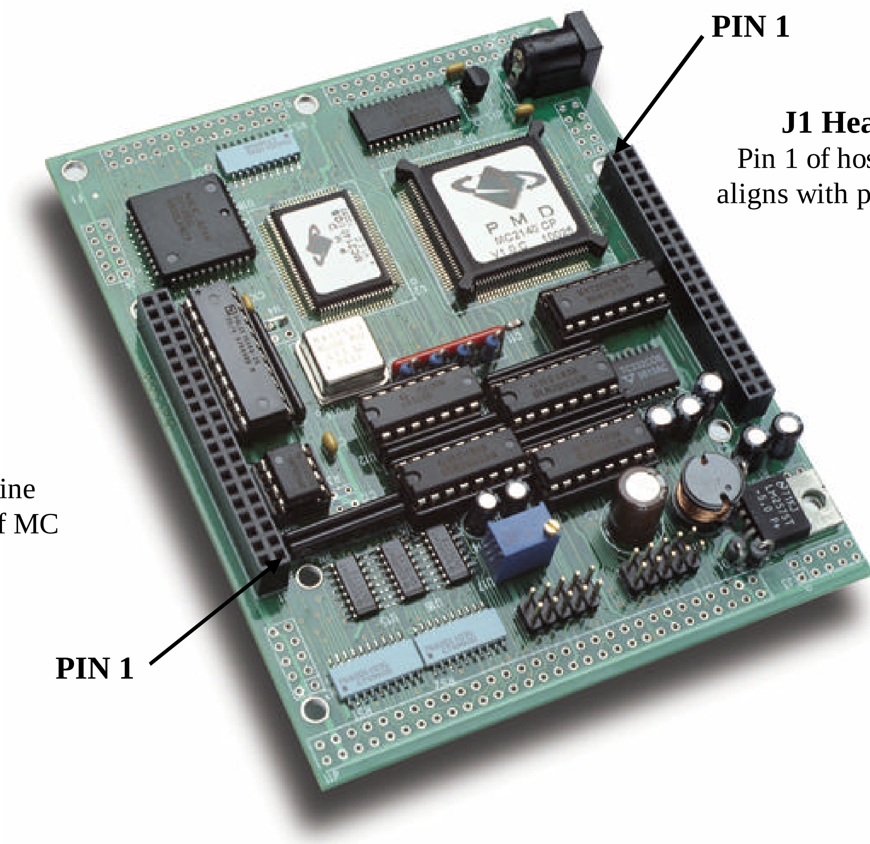
Connect PC to the MotionC 2140 using the debug serial cable provided in the EV-P/DV-P Kit.

Connect 9V wall transformer to DC power jack on MotionC 2140

User is ready to begin development

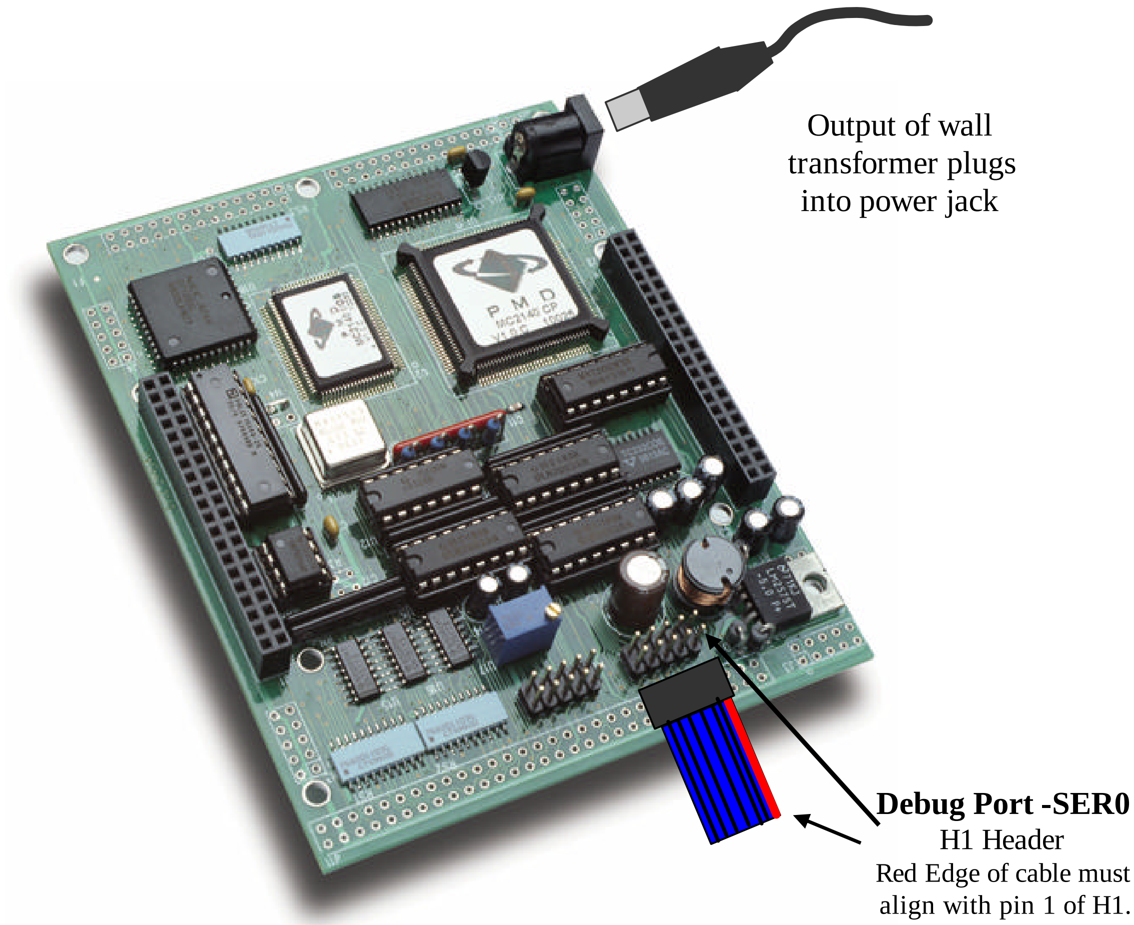
*2.2.1 Connecting the MotionC*2140 *to the host engine (AE, AE86, IE, 5E)*

The host engine controller installs onto the MotionC 2140 via two 20x2 pin headers. These pin headers are named J1 and J2. They are labeled on both your host engine and the MotionC 2140.Confirm the correct orientation by aligning J1 pin 1 on your host engine with J1 pin 1 on your MotionC 2140.This will align J2 pin 1of your host engine with J2 pin 1 of your MotionC 2140. Pin 1 of any header can be identified as the pin closest to the header label. For example, the pin closest to the "J1" marking on your host engine is J1 pin 1.



MotionC shown without a host

2.2.2 Connecting the MotionC2140 to the PC and power



Chapter 3: Hardware

3.1 i386-Engine/A-Engine/A-Engine86/586-Engine

The MotionC2140 uses an i386-Engine, A-Engine, A-Engine86, or 586-Engine as its host microprocessor core module. Please refer to the corresponding x-Engine Technical Manual for more information.

3.2 Interface with PMD MC2140 DSP chipset

The *MC*2140 DSP chipset is packaged in two surface mount chips, “CP” and “I/O”. The chipset is driven by a host x-*Engine* via an 8-bit, bi-directional port. Communication to and from the chipset consist of packet-oriented messages. An interrupt line /HINT, from the “CP” pin 98, is routed to J2 pin 6 of the i386-Engine (/INT6), A-Engine (/INT1), or A-Engine86(/INT1), so the chipset can signal the host when special conditions occur, such as receiving an encoder index pulse.

The i386-Engine / A-Engine / A-Engine86 / 586-Engine write commands to the *MC*2140 and reads data from the *MC*2140 chipset. Each command consists of a 16-bit word, with a command code value defined in the *MC*2140 manual. Data is transmitted to and from the chipset in 16-bit words.
“C” Functions are available in **mc21.lib** and prototypes are listed in mc21.h file. Many sample programs are also available in the EV-P/DV-P kits in the **samples\mc21** directory of the respective host. See Chapter 4 for a list of appropriate sample code.

void mc21_host_dat_wr(unsigned int dat); // host Engine writes 16-bit dat to MC
void mc21_host_cmd_wr(unsigned char cmd); // host Engine writes 16-bit cmd to MC
unsigned int mc21_host_dat_rd(void); // host Engine reads 16-bit dat from MC
char mc21_host_rdy(void); //return 0 for “I/O” pin 8, HRDY low, indicating busy

3.3 *MotionC* 2140 I/O Map

The following tables list the I/O address of the *MC*2140, together with their Data Bits, Chip-Select Symbol and Functions.

Base I/O Address	Data Bits	Select Symbol	Function
0x??b0	D0-D7	/MC	Read/Write D0-7 from/to *MC*2140 “I/O” chip (U2)
0x??c0	D0-D7	/PPI	Read/Write D0-7 from/to PPI 82c55 I/O chip (U18)
0x??e	D0	/RST1	Hardware reset “CP”.

3.4 Quadrature Encoder Inputs (*MC*2140)

The *MC*2140 supports up to 4 channels of Incremental Encoder inputs for motor position information. Each quadrature encoder channel consists of a square wave, offset 90-degree from the other, with the leading phase indicating direction. For every channel, four position inputs and control signals are supported:

A channel pulses (QDAx)
B channel pulses (QDBx)
Index pulse

Home switches signal

The quadrature inputs (QDAx, QDBa, and IDx) support differential and single-ended inputs. If differential inputs are required, differential line drivers (75173) can convert the differential inputs into single-ended signals, which ultimately input to the DSP chipset. The differential line drivers buffer all four axes and are located at U15, U16, and U17. Refer to the 75173 data sheet for additional specifications (75173.pdf in the tern_docs\parts directory). If only single-ended inputs are needed, then the quadrature inputs should be connected to the (+) signal only. The (-) signal should be left open (or floating). For example, to connect single-ended quadrature input to axis 1 of the MC21, use the signals +QA1, +QB1, and +ID1 only, leave –QA1, -QB1, and –ID1 unconnected. In addition, if the index pulse input is not needed, leave those signals open also. The quadrature encoder inputs are not optically isolated from digital ground (GND).

For a better understanding of how signals travel throughout the MotionC2140, examine one example. Consider Axis 1. Refer to the MotionC2140 schematic.

Differential Quadrature Encoded Inputs can be tied to J11 pins 25 (+QA1) and 27 (-QA1). To start, these signals are pulled high via the resistor network at **RN5**. These signals then lead to the differential line driver located at **U17** which are converted to a single-ended signal, **QDA1**. **QDA1** is routed to the DSP “IO” (U2) at pin 47.

This same process applies to +QB1 (J11.21) and –QB1 (J11.23) which are pulled high by the resistor network at **RS1** and converted into **QDB1** by the differential line driver at **U17**. **QDB1** is then routed to the DSP “IO” at U2.25.

Similarly, +ID1 (J11.17) and –ID1 (J11.19) are pulled high by **RS1**, and converted into a single-ended signal, **ID1**, by the differential line driver at **U16**, and then routed to the DSP “IO” at U2.49.

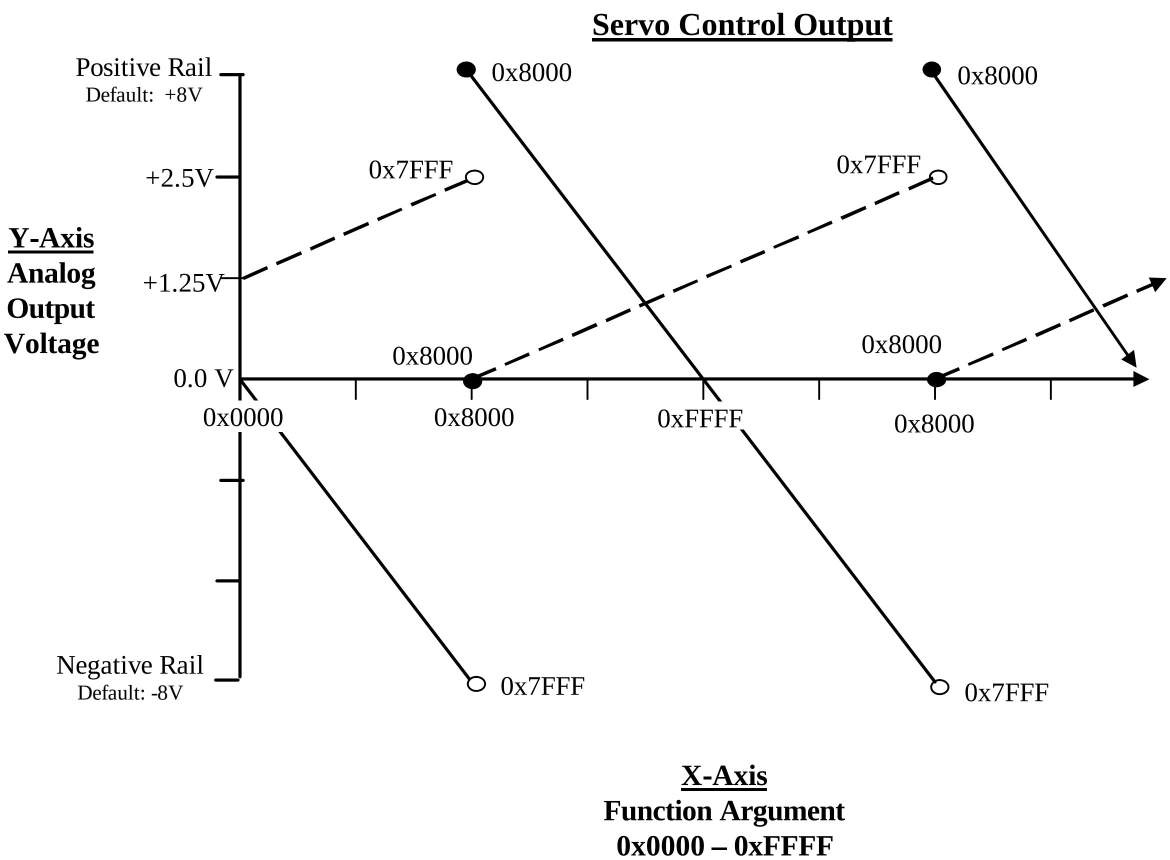
3.5 *MC*2140 DAC Servo Analog Outputs

The *MC*2140 supports both DAC and PWM output mode for external servo motor amplifier. The *MC*2140 uses a quad 12-bit voltage output DAC converter (DAC7625, U7). While the 16-bit DAC output mode is used, the 12-bit DAC uses the higher order 12 bits from the DSP “CP” chip.

The DAC7625 contains four precision output buffer amplifiers, providing full 12-bit performance at 1LSB total unadjusted error without adjustments. The DAC7625 has a typical 3µs output setting time and outputs 0 to 2.5V with an external 2.5V precision reference. A quad amplifier buffer (LM324A, U12) with adjustable gain and offset supports 4 channels of default ±10V analog servo control signals at header J11 (DA1-4), supporting a variety of motor amplifier interfaces. The J11 pin header is detailed later in this chapter.

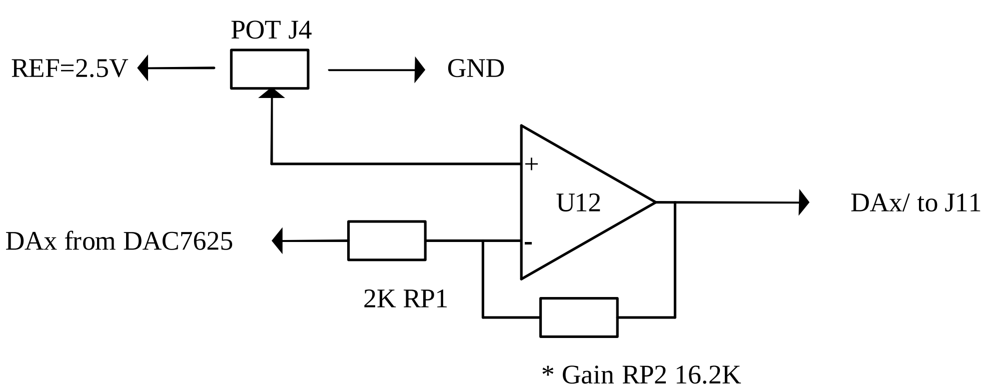
A resistor potentiometer is installed in J6 providing the offset voltage. The default gain is 8, as derived from the value of RP2 / RP1 = 16K / 2K. The user may replace default 16K gain resistors in RP2 to setup necessary gains.

The DA1/ (J11.45), DA2/ (J11.46), DA3/ (J11.31), and DA4/ (J11.32) provides buffered output voltage from the operational amplifier (U12), capable of up to 20 mA. The *MC*2140 outputs a 16-bit data word as an unsigned 16-bit number with a range of 0-65535. The following graph shows the analog output of the servo control base upon the passed function argument. The argument is of type unsigned integer, yet the 12-bit DAC only uses the upper 12-bits of an integer value. The x-axis is the function argument, and the y-axis is analog voltage. The dashed lines represent the 0-2.5V output from the DAC itself, while the solid lines represent the output of the operational amplifiers, signals DA1/, DA2/, DA3/, and DA4/ routed to the J11 pin header. Note the discontinuities at 0x8000. As the function argument approaches 0x8000, the servo control will approach the negative rail of the operational amplifier, but at 0x8000 it will become the positive rail. This graph is not to scale.



An additional note about the supply voltages for the operational amplifiers: By default, the operational amplifiers use the (+) and (-) voltages generated by the on-board RS-232 Divers for the supply voltages. For this reason, the output of the amplifiers will not reach +10 and -10 volts. It is the responsibility of the user to route external supply voltages (+12V, -12V) for the amplifiers to the J11 pin headers (pins 59,60) if a full -10 to +10 volt scale is needed. See Section 3.8 for additional discussion about power supplies. The sample code "mc21_dac.c" is also available in the samples\mc21 directory that corresponds with the host engine.

Block diagram for the operational amplifiers, adjustable gain, and offset voltage is shown below. RP2 can be changed by the user to create a new output gain, where gain = RP2/ RP1, with the default setting 16K/2K = 8.



3.6 Limit, Home, and Fault Switches

There are 2 limit switch inputs for each axis: +L1 and –L1, yielding a total of 8 limit switch inputs (+L1-4, and –L1-4). There are also 4 home switch inputs: HOM1-4 and 4 fault switch inputs: FLT1-4. All switch inputs are routed to J11 and protected by Darlington transistor arrays (ULN2003A, see ULN2003A.pdf in the tern_docs\parts directory). Since all Limit, Home, and Fault switches are buffered by Darlington transistor arrays, the Home, Limit, and Fault signals at the DSP chipset will have inverted logic. More specifically, since the input on the J11 header are default low, the signals will be default high at the DSP chipset (In addition, the switch signals between the Darlington arrays and the DSP are pulled high to guarantee default high). Any voltage applied to the switch inputs at J11 must be between 3-30V to be considered a valid high input. This will then create an active low signal at the DSP chipset for the appropriate switch.

Consider Axis 1:

Home switch is routed to J11.55 (HOM1). HOM1 is routed to the Darlington array at U4. The corresponding output of the Darlington array is named HM1, routed to the DSP "IO", U2.82.

Fault switch is routed to J11.49 (FLT1). FLT1 inputs to the Darlington array at U4 and its corresponding output is IN1, routed to the DSP "CP", U1.72.

The Limit switches are routed to J11.51 (+L1) and J11.53 (-L1). These input to the Darlington array at U4. Their corresponding outputs are +LM1 and –LM1, routed to the DSP "CP", U1.63 and U1.64, respectively.

See sample programs: **mc21_h_i.c**, and **mc21_sta.c** for details.

3.7 Power Amplifier Control

Seven high voltage driver outputs (ULN2003) are designed to sink up to 350 mA at 50V. The high voltage driver is located at U11 and is driven by four signals from the DSP and three PIO's from the host controller. O1-O4 from the DSP drive the output lines EN1-EN4 on the J11 header. OU1-OU3 are driven by the host and are located at J2 pins 11, 18, and 20. They output signals HO1-HO3 on the J3 pin header. The data sheet ULN2003A.pdf gives additional information on the high voltage drivers. See sample code mc21_hv.c. Although four high voltage outputs are driven by the DSP chipset, they are not part of the PID/PI control. These lines can be used to enable external power amplifiers to drive DC motors, or drive solenoids. See sample program mc21_hv.c

3.8 Power Supplies, Digital and Analog Ground

The *MC*2140 can be powered by a single 12V DC via J11(30x2 header) or JDC power jack (J0). The on-board linear 5V or optional switching regulator (U9) can produce 5V for the DSP and the host engine. The RS232 driver (U8) is powered by 5V and produces negative voltage for both RS-232 and operational amplifiers for using a 12V wall transformer, as default. For this reason, as discussed earlier, the operational amplifiers will not output a full -10V - +10V range just by using the supply voltage from the RS-232 Most users do not find it necessary to be able to output a full -10V - +10V range. If this is required, the user may provide supply voltages on the J1 header to power the amplifiers to yield a full output range.

User may provide regulated field power supply +12V, -12V, VCC and GND via J11 header. The +12V, -12V and GND are used by analog output operational amplifiers. The +12VDI and GND will generate a regulated 5V with the on-board regulator installed. OEM product may use external 5V power supply with the on-board 5V regulator removed.

If you use the on-board +5V VCC regulator (7805) to power the external quadrature encoders, you will need to provide additional large heat sinks to the 7805 regulator, such as mounting on a large metal standoff. If you want to provide an external regulated +5V to the *MotionC*2140 via J11, you should not install the on-board 5V regulator.

3.9 24 I/O Lines of PPI

The MC21 is installed with an 82C55A PPI chip at location U18. The I/O lines are available at the J9 13x2 pin header. It provides 24 TTL user-programmable I/Os which can be used to interface LCDs, Keypad, or power relay drivers. A pull-up resistor pack is connected to 19 of the I/Os. The PPI has four 8-bit registers; one for each 8-bit I/O port and one control register. The PPI is mapped into the host controller's I/O space, yet the actual location varies with the host. See mc21_ppi.c for details.

3.10 RS-232 and RS-485

Two channels of RS-232 serial ports are available on the *MotionC*2140.

H1 SER0 for debugging
H2 SER1 for application.

An RS-485 driver and header H3 supports the optional SCC2691 UART.

3.11 DSP Ready Signal to the Host *i386-Engine/A-Engine/A-Engine86*

The DSP IO chip pin 8 (HRDY) is a hardware ready signal that indicates the DSP is busy while it is low.

The *MC*2140 routes HRDY signal to:

i386-Engine - J2 pin 12 (P14)
A-Engine – J2 pin 12 (P10)

3.12 J11 pin header: Interface to the DSP

The following diagram is a complete summary of the signals routed to the J11 30x2 pin header on the MotionC2140. All signals for Axis 3 have been labeled as an example, which are identical to the signals for other axes. Fault, Limit, and Home input switches are buffered by Darlington transistor arrays (ULN2003A) and allow up to +30V input. They are low by default and require 3-30V input to be a valid high input. Quadrature encoder inputs are buffered by differential line drivers. If your application only requires single-ended quadrature inputs, use the (+) inputs only, and leave (-) floating.

	VCC	1	2	VCC	
	GND	3	4	GND	
Differential Quadrature Encoder, Index	+ID3	5	6	+ID4	
Differential Quadrature Encoder, Index	-ID3	7	8	-ID4	
Differential Quadrature Encoder, Channel B	+QB3	9	10	+QB4	
Differential Quadrature Encoder, Channel B	-QB3	11	12	-QB4	
Differential Quadrature Encoder, Channel A	+QA3	13	14	+QA4	
Differential Quadrature Encoder, Channel A	-QA3	15	16	-QA4	
	+ID1	17	18	+ID2	
	-ID1	19	20	-ID2	
	+QB1	21	22	+QB2	
	-QB1	23	24	-QB2	
	+QA1	25	26	+QA2	
	-QA1	27	28	-QA2	
		29	30		
Servo Control; Output, -10V to +10V	/DA3	31	32	/DA4	
Sinking Solenoid Driver; Output, Up to 350mA @ +50V	EN3	33	34	EN4	
Fault Switch; Input, 3-30V	FLT3	35	36	+ID4	
Limit Switch A; Input, 3-30V	+L3	37	38	+L4	
Limit Switch B; Input, 3-30V	-L3	39	40	-L4	
Home Switch; Input, 3-30V	HOM3	41	42	HOM4	
		43	44		
	/DA1	45	46	/DA2	
	EN1	47	48	EN2	
	FLT1	49	50	FLT2	
	+L1	51	52	+L2	
	-L1	53	54	-L2	
	HOM1	55	56	HOM2	
	K	57	58	GND	
Amplifier Supply Voltage, Input, Provided by User	+12VI	59	60	-12VI	Amplifier Supply Voltage

Chapter 4: Software

Please refer to the Technical Manual for the "C/C++ Development Kit for TERN 16-bit Embedded Microcontrollers" on debugging and programming tools. For software information related to the *A-Engine, A-Engine86, or i386-Engine* controller, please refer to the respective manual.

The sample code provided for the MotionC2140 is the best way to learn about programming the DSP. You will see examples on how the functions below are implemented, how to select axes, what to values to write, timing, interrupt routines, and more. Familiarize yourself with each sample code before attempting to create one huge application.

It is necessary to include mc21.h in your source code and link to mc21.lib in your project in Paradigm C/C++ environment.

4.1 Functions in MC21.LIB

void ***mc21_host_dat_wr***(unsigned int dat); // host A-Engine writes 16-bit dat to MC21

void ***mc21_host_cmd_wr***(unsigned int cmd); // host A-Engine writes 16-bit cmd to MC21

unsigned int ***mc21_host_dat_rd***(void); // host A-Engine reads 16-bit dat from MC21

char ***mc21_host_rdy***(void); // return 0 for MC IO pin 8 low, indicating host port busy

```
unsigned int mc21_host_status_rd(void); // return 16 bits of the
status register of DSP chipset.

void mc21_hard_reset(void); // Issue a hardware reset to the DSP chipset

unsigned int mc21_adc_rd(char ch); // reads 10-bit ADC from DSP
      // ch=0-7 for AN1-8 at header J8
```

4.2 Sample Programs

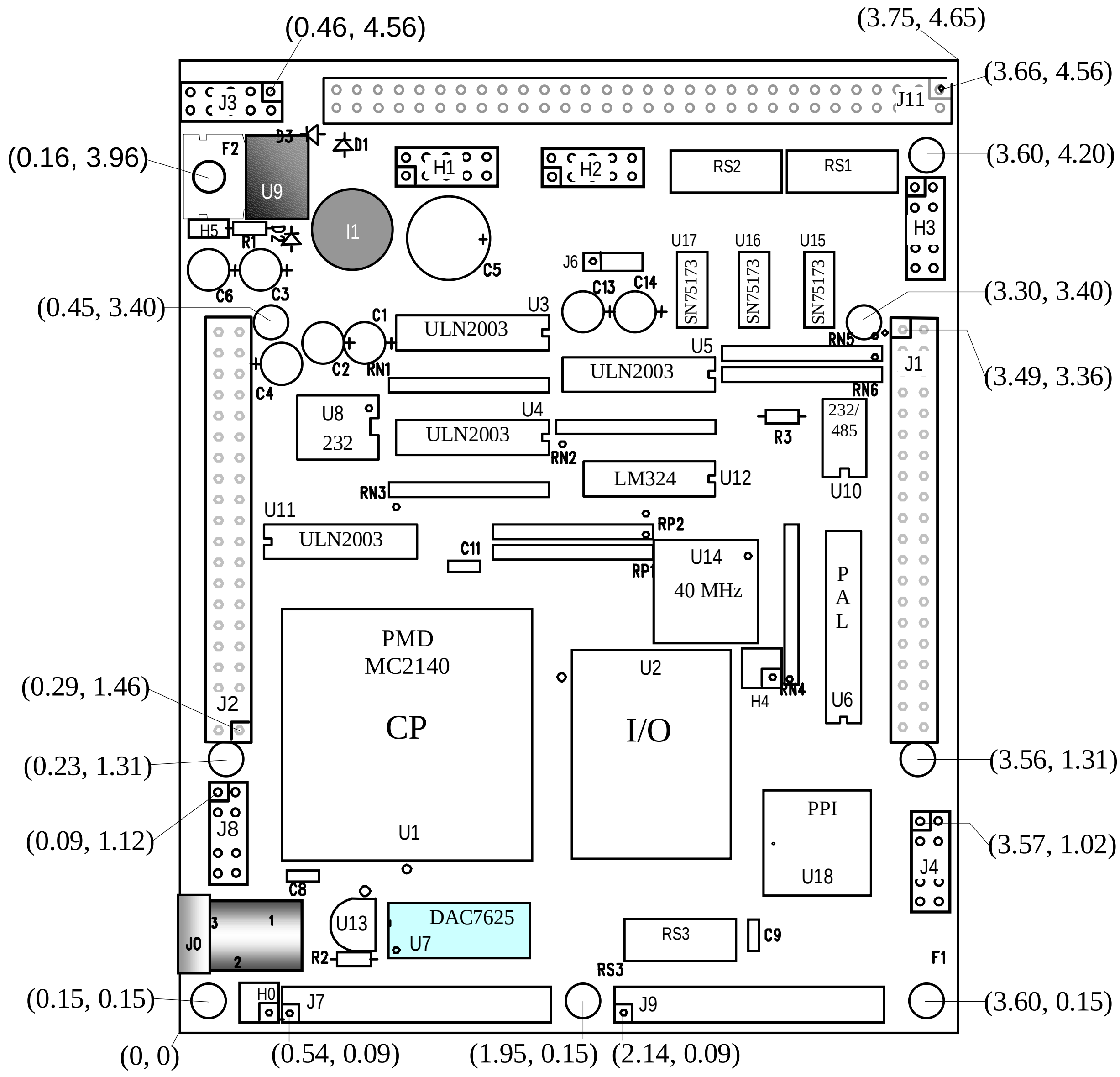
Sample programs for the *MotionC* 2140 are located in the samples directory at **`\samples\mc21`**.

mc21_pwm.c	// pulse-width modulation
mc21_ppi.c	// drive the PPI chip, 24 programmable I/Os
mc21_adc.c	// read the analog inputs on the DSP chipset
mc21_dac.c	// drive the analog outputs (servo control)
mc21_sta.c	// Status. Read the status of limit, home, and fault switches
mc21_h_int.c	// implement an interrupt-service-routine for the home switch

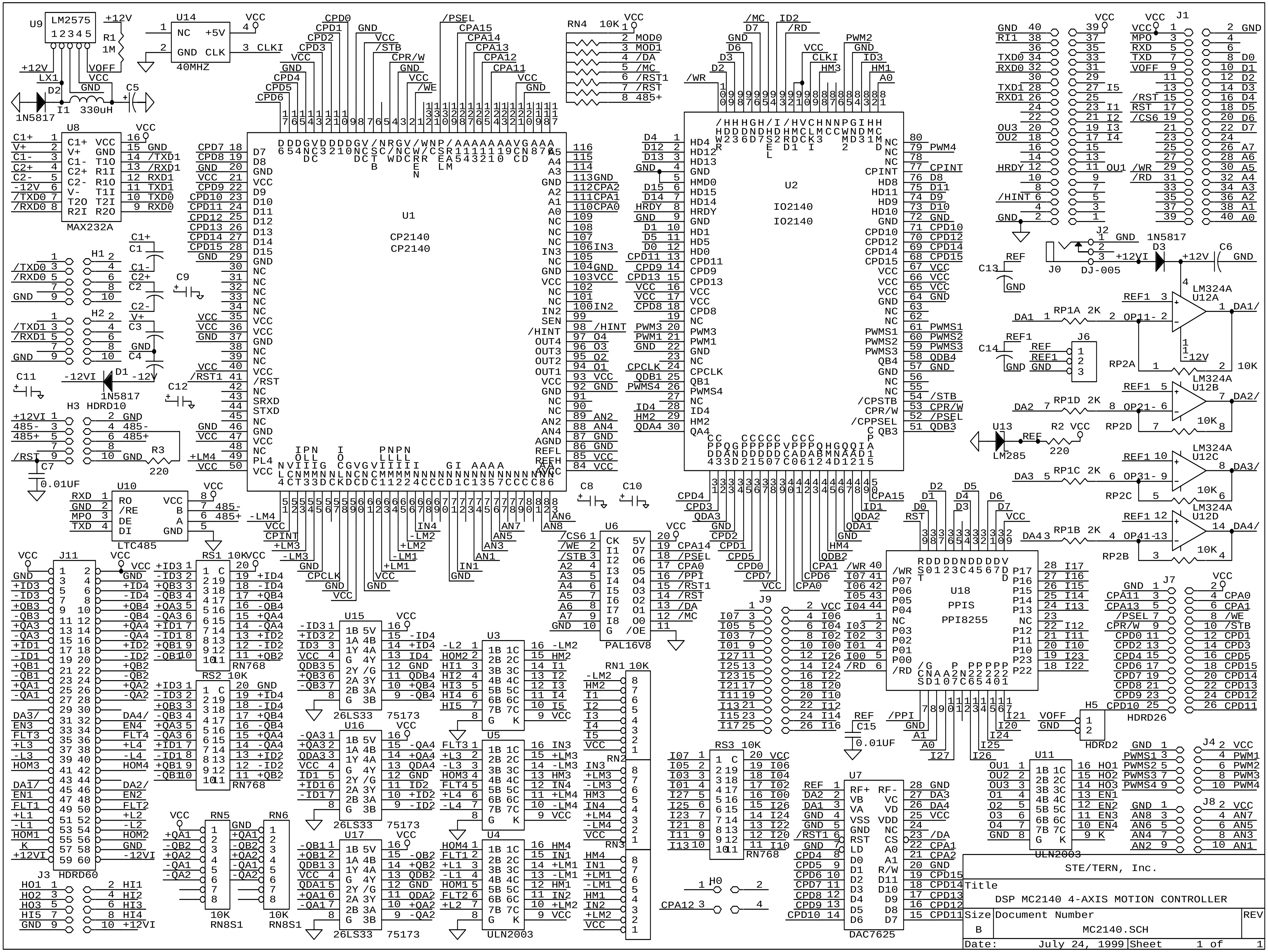
mc21_d3.c	// Demo application
mc21_pos.c	// read the quadrature encoder inputs
mc21_ver.c	// return DSP version number
mc21_hv.c	// drive the solenoid outputs

Appendix A: *MotionC*™ 2140 Layout

The ***MotionC*™ 2140** measures 4.65 x 3.75 inches. Layout is shown below.
All dimensions are in inches.



TERN INC.
1724 Picasso Avenue, Suite A, Davis, CA 95616-0547, USA
Tel: 530-758-0180 Fax: 530-758-0181
Internet Email: sales@tern.com *http://www.tern.com*


STE/TERN, Inc.
Title
DSP MC2140 4-AXIS MOTION CONTROLLER
Size B
Document Number MC2140.SCH
REV
Date: July 24, 1999
Sheet 1 of 1