Encoder The Newsletter of the Seattle Robotics Society

'Three 68332 Designs Presented'

By: Kenneth Maxon

Additional designs contributed by: Jeff Bronk, Mark Castelluccio

The CRC332 (C-Robot-Controller) board has been well received with a surprisingly high volume of interest. Further more, with the MRM332 (Mini-Robo-Mind) also available 32 bit processing power really is bringing a large following of amateur roboticists to that proverbial next level. The 68332AAB e-groups list server has been buzzing with lots of shared code, several people working on vision systems and other specialized projects. Amongst all of this excitement one item that has been lacking is the presentation of the board design and layout itself. In previous articles the author has explored programming the onboard CPLD's CPLD Encoder Article and Interfacing the TPU 68332 Servo Control Encoder Article to encoders and PWM motor drivers. Since the author no longer offers these boards for sale having shipped 62 of them in a few short months, the schematic and board artwork for the CRC332 are presented here in the hopes that putting this design into the public domain, schematics, artwork and all, it will encourage others who follow to make use of it. At the very end of this article the reader can see artwork showing how three of us have already put this technology to good use.

When I started this article I had but my design upon which to base it and as such, much of the article focuses on the CRC332. Since then, Jeff Bronk and Mark Castelluccio have come foreword to share their schematics and design information with the author. These two Key members of the 332SIG drove much of the design process during our respective board developments and provided a qualitative sounding board and expert level technical input.

Now having the full resources of all three designs to draw upon this article has grown significantly in length. Each applicable section (reference the article overview following) will be further broken down into three sub-sections to address each of the three designs in turn.

It is the authors intent to present all three designs fully and impartially. A hard task, given the authors personal bias :-) If one feature appears favored over another, please excuse this lack of editing skill and understand that the designs evolved under differing sets of criteria, each toward a different end. Each design is unique and offers special consideration as to functionality and implementation.

Article Format:

In the first twelve sections the author presents the overall 68332 based board designs. Each section is further broken into three sub sections highlighting and contrasting differences and features between the three designs.

Section	content		
1:	1:1 - CRC332 - General Overview Files and Board Layout and component stuffing	1:2 - Jeff's Board - General Overview Files and Board Layout and component stuffing	1:3 - MRM332 - General Overview Files and Board Layout and component stuffing
2:	2:1 - CRC332 - CPU, Reset, Clock & LCD Connection	2:2 - Jeff's board - Reset CPU & clock oscillator circuitry	2:3 - MRM332 - Reset, CPU & clock oscillator circuitry
3:	3:1 - CRC332 - SPI interface ports & Real Time Clock	3:2 - Jeff's board - SPI ports	3:3 - MRM332 - SPI ports
4:	4:1 - CRC332 - Battery Backup, RAM, ROM & FLASH	4:2 - Jeff's board - Battery Backup, RAM & ROM	4:3 - MRM332 - Battery Backup, RAM & FLASH
5:	5:1 - CRC332 - Power Supply & Onboard Inverter	5:2 - Jeff's board - Power Supply	5:3 - MRM332 - Power Supply
6:	6:1 - CRC332 - Analog To Digital Converter	6:2 - Jeff's board - Analog To Digital Converter	6:3 - MRM332 - High Speed Analog to Digital Converter

7:	7:1 - CRC332 - CPLD-1 PIA function.		
8:	8:1 - CRC332 - RS-232 Level Drivers.	8:2 - Jeff's board - In cable 232 converter	8:3 - MRM332 - In cable 232 converter
	8:4 - General RS232 Drivers link & schematic		
9:	9:1 - CRC332 - Vertical Stacking Bus & TPU breakout.	9:2 - Jeff's board - IO expansion peripherals & TPU breakout	9:3 - MRM332 - Expansion port & TPU breakout
10:	10:1 - CRC332 - CPLD-2 built in BDM Driver.	10:2 - Jeff's board - BDM cable & reset driving	10:3 - MRM332 - BDM cable & reset driving, including Mark's FLASH modified BDM32
	10:4 - General BDM interfaces		
11:	11:1 - CRC332 - Special LCD cable wiring.	11:2 - Jeff's board - Software Driven LCD connection	11:3 - MRM332 - ECLK LCD connection
12:	12:1 - CRC332 - Artwork	12:2 - Jeff's Artwork	
	12:4 - General APCircuits Information & GCPrevue		
13:	13:1 - CRC332 - IO Expansion Board & schematics		
14:	14:1 - CRC332 - Memory Expansion Board & schematics		
15:	15:1 - Jeff's I2C Implementation		
16:	16:1 - General - Using the board software & misc.	16:2 - General - Jeff's 8 to 16 bit switch over boot loader	16:3 - General - Link to Karl Lunt's home page for the S-Basic programming language
17:	17:1 - General - Documentation.		
18:	18:1 - General - Parting Words.	18:1.1 - CRC332 photo	18:1.2 - Jeff's board photo
	18:1.3 - MRM332 photo		

Section 1: General Overview

1:1

The underlying design theme used in the development of the CRC332 is to completely encapsulate enough power on the board to implement a personal robot. Some tradeoffs in PCB real estate were made to add features required to accomplish this task. Throughout this article the reader will observer that every attempt has been made to configure headers and provide feature connectors to attain this goal.

In undertaking the writing of this article it is the author's sincere hope that enough information is provided for the reader to build his or her own 68332-based variant. Following is a complete list of features found on the CRC332:

● 1M-Byte RAM ● 512K-Bytes Flash ● 256K-Bytes EPROM ● 8 channels AtoD ● 24 firmware programmable digital IO's ● 4 onboard SPI ports ● Onboard power regulation & reset ● Built in BDM interface circuitry, all that's in the cable is wire. ● Fully buffered RS232 w/industry standard DB-9 connector ● LCD interface with Custom circuitry to run display while the processor executes at full speed ● Onboard inverter to supply negative rail voltages to sensors, amplifiers & LCD contrast adjustment ● RAM battery back up, onboard & off ● Real time clock calendar w/battery backup ● Vertical Stacking Expansion bus w/several modules available. ● Jumper Configurable boot source, RAM, EPROM or Flash ● Jumper Configurable bus width ● Silk Screened & Solder Masked Production Quality PCB ● Size: 3.375" x 3.9" ●

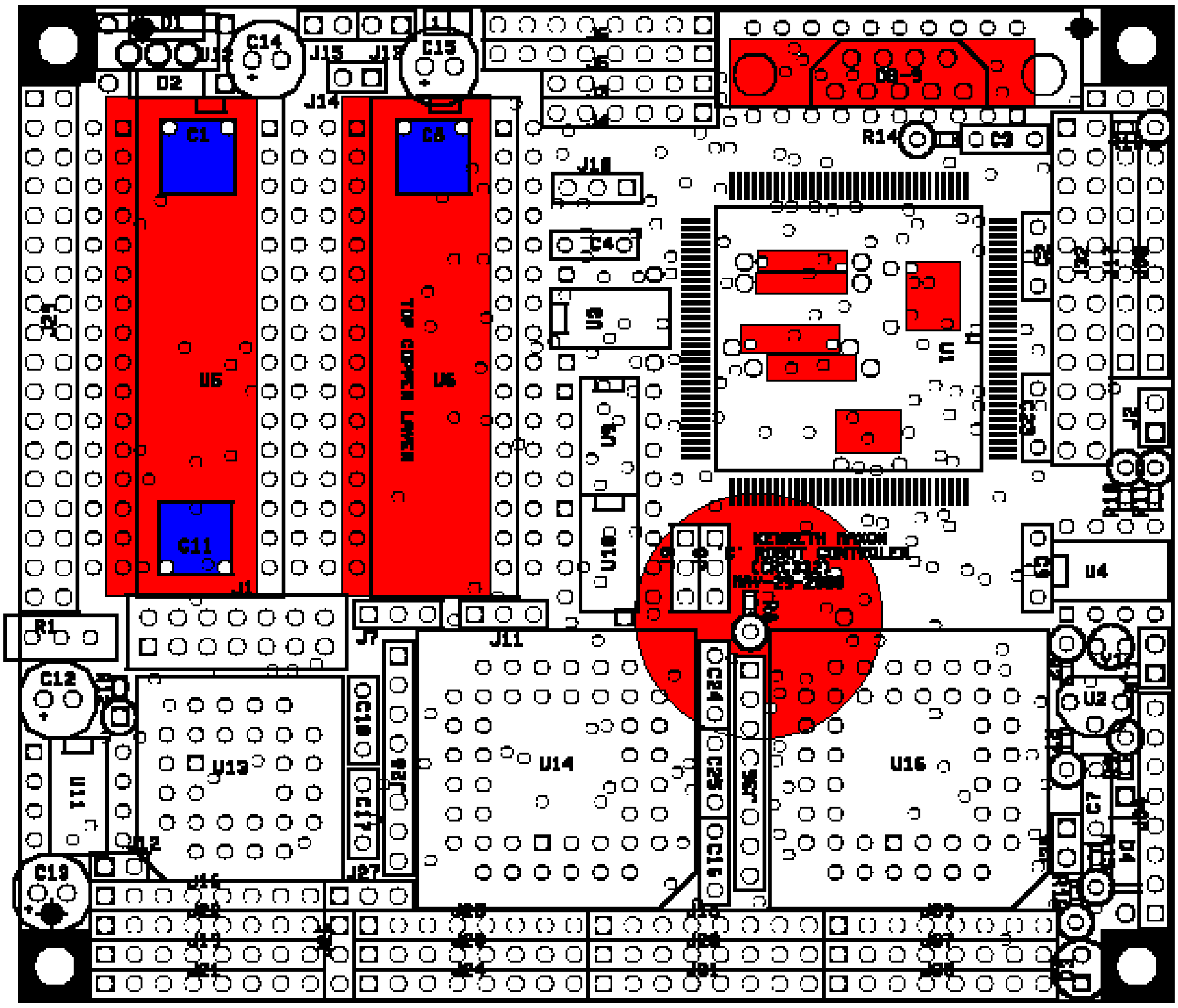
Bill Of Materials (6K)

Included (left) is the Bill Of Materials including all of the components required to fully stuff the users own version of the CRC332 assuming that he/she makes no additions/modifications to the design. The pricing information shown is that available for full retail mail order centers like Digi-Key & Jameco. Looking to save quite a bit of money, many of these components can be purchased through clearing houses / surplus centers at greatly discounted rates in single or 1-2 quantities.

The board overview image (below) depicts all of the major functionality. Not visible from this top down view are the ROM, FLASH, BATTERY & RS232 Driver IC's as they are mounted to the bottom of the board. In the second image (below) these components are identified in red.

C1, C8 & C11 showing through the bottom mounted FLASH & EPROM (Blue, in the diagram below) are actually mounted to the top side of the board and are bent over to lay down flat inside of the IC's respective sockets. The battery used has built in stand off tabs that keep it from shorting out on the leads protruding from the bottom of the board. Other mounting discussions for the bottom mount components are discussed in their respective sections of this article.

Board Top & Bottom Stuffing View



One of the nice features that benefited the design in two ways is the choice of through-hole components. Most novice designers assume that since surface mount components are used in miniaturized layouts throughout the industry, that the only way to pack large amounts of silicon into their own designs is to follow suite. Unfortunately, this only holds true when using multi-layer boards where additional routing space is not at such a premium. Through hole parts alleviated the routing problem on the CRC332. Additionally, through hole parts permitted the use of sockets on every IC except the processor which is not made in a through hole package. Over the years, the author has taken notice of the volume of e-mails that go back and forth on the SRS list server every year. In these e-mails the magic of the blue smoke that powers all IC's seems to be a favorite subject and through hole *socketed* parts allow the end user to replace the one part they've smoked. All this without damaging the board during solder re-work.

One of the more often asked questions is why I chose to use a non-standard BDM implementation. The design philosophy here was to give the end user everything they need up front. Other boards out there offer the BDM PCB / Cable assembly and serial PCB / Cable assembly as options to buy at a later date. I chose to build both of these functions into the board. All of the IC functionality required to drive BDM has been written as logic inside U16. (The second CPLD on the board) All that's needed to connect the BDM interface to the parallel port on the users computer is a simple cable that has nothing but wire inside. These cables were shipped with the CRC332 as well. The same concept applied to the serial implementation, which required only a simple 9 pin D-sub cable to connect to a PC's serial port.

Another important feature comes by way of the systems design. The design is portable and modular. Hardware-wise, functionality may be added or removed in discrete increments, removing concerns due to interference with the rest of the system. This design criteria was adopted towards the goal of selling the partial stuffed CRC332 option. By this means, users of the board could purchase a bare minimum system and later add functionality at their discretion. This pays off a second time for you the reader in that you may remove or add functionality when redesigning your own design based on the CRC332 framework.

Author's CRC332 schematic

CRC332_Schematic1(89K) Artwork_File(38K) Zipped_Software_File(15K)

The schemata for the CRC332 have been laid out to fit the aspect ratio of a "B" sized plot (11"x17") They will be best viewed when plotted on this size of paper. Viewing and plotting will require the Adobe Acrobat viewer. When you open this file you should find 10 pages if the download was successful.

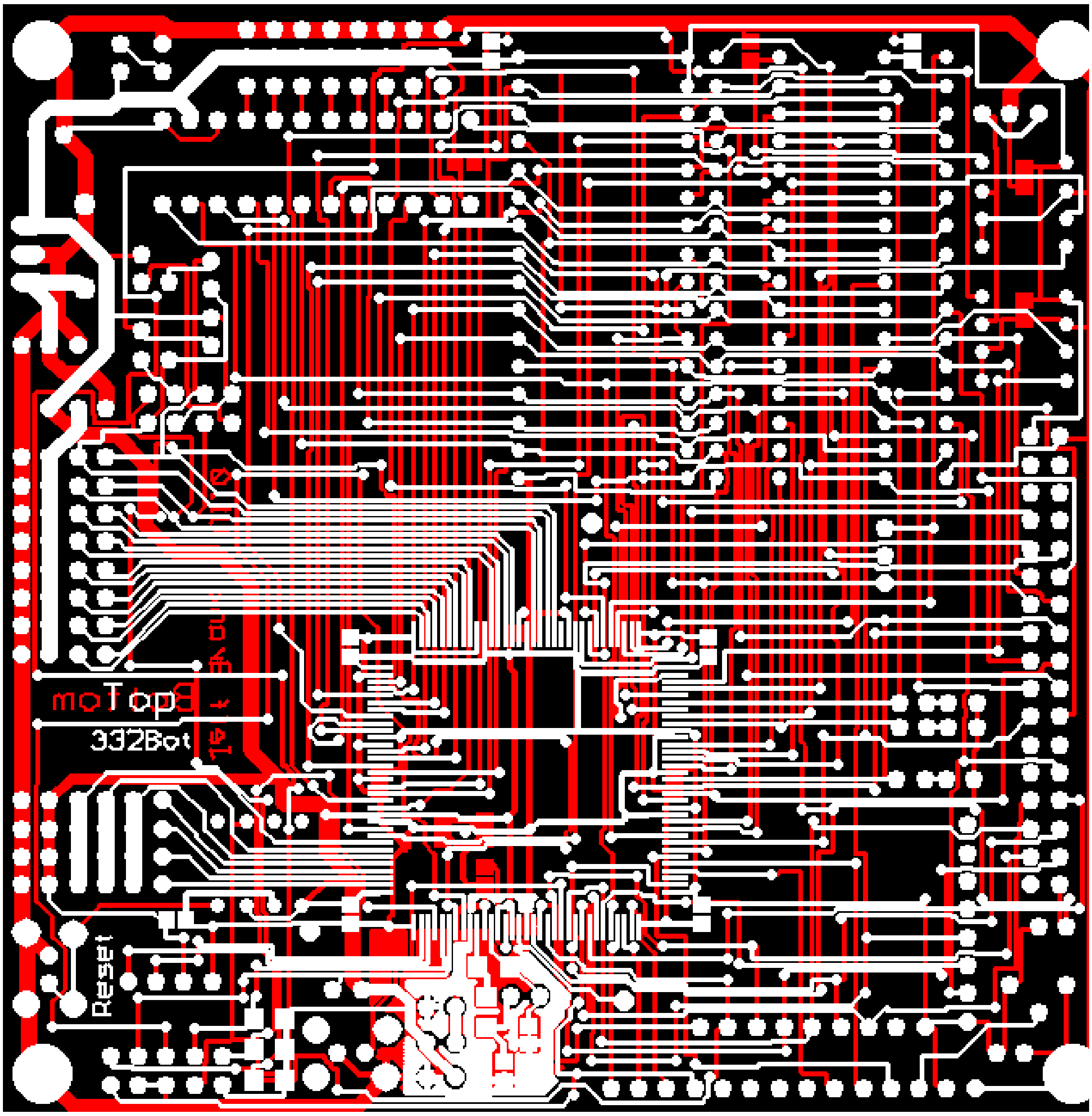
Additional Schematics and support files are sprinkled through out the article below including those that describe the IO expansion board and memory expansion board.

● 1:2 ●

Jeff Bronk's board lies between the authors CRC332 and Mark's MRM332. The design is clean and balances extra features against the requirement for **NO** extra onboard glue logic.

● 1M-Byte RAM ● 256K-Bytes EPROM ● 8 channels AtoD ● 4 onboard SPI ports ● Onboard power regulation & reset ● External BDM interface ● External RS-232 serial expansion ● LCD interface with software interface ● RAM battery back up ● Break out expansion IO connector ● 16 bit wide RAM bus ● Hardware to support TPU based I2C functions ● Jumper selectable TPU power sources ●

Below the footprint layout for Jeff's board can be seen. This board is completely laid out by hand. Note how clean all of the routes are. A superior job, this design will function well and be fairly immune to noise even on a two layer board without ground plane:



Jeff also used a large number of through hole components in his design, including RAM, EPROM, AtoD, discreets and others... Again the through hole functionality eased board layout efforts. Using a copper size of 0.010" on a 0.010" grid with a minimum spacing of 0.008" seems to work well allowing two traces between two adjacent through hole pins on 0.100" centers. Pad sizes must be shrunk slightly from 0.055" down to 0.050" for this spacing scheme to work correctly. The author used similar spacing on the CRC332 board. Note: all power traces are substantially larger (0.022" on the CRC332).

One of the features that makes Jeff's board truly stand out is the inclusion of the I2C bus driver hardware and functionality. We'll delve into this in section 15 below.

In choosing to use an in the cable BDM interface along with an in the cable serial interface, Jeff has reduced power draw, board size and weight all at the same time.

Jeff Bronk's Schematic

Jeff's_Schematic(236K) Jeff's_Zipped_Boot_Code(3K) Artwork_File(xxK)

Thinking quite a bit ahead, Jeff has taken the time to fit the entire schematic onto a single 'B' sized page. This page is also viewable with the acrobat viewer.

● 1:3 ●

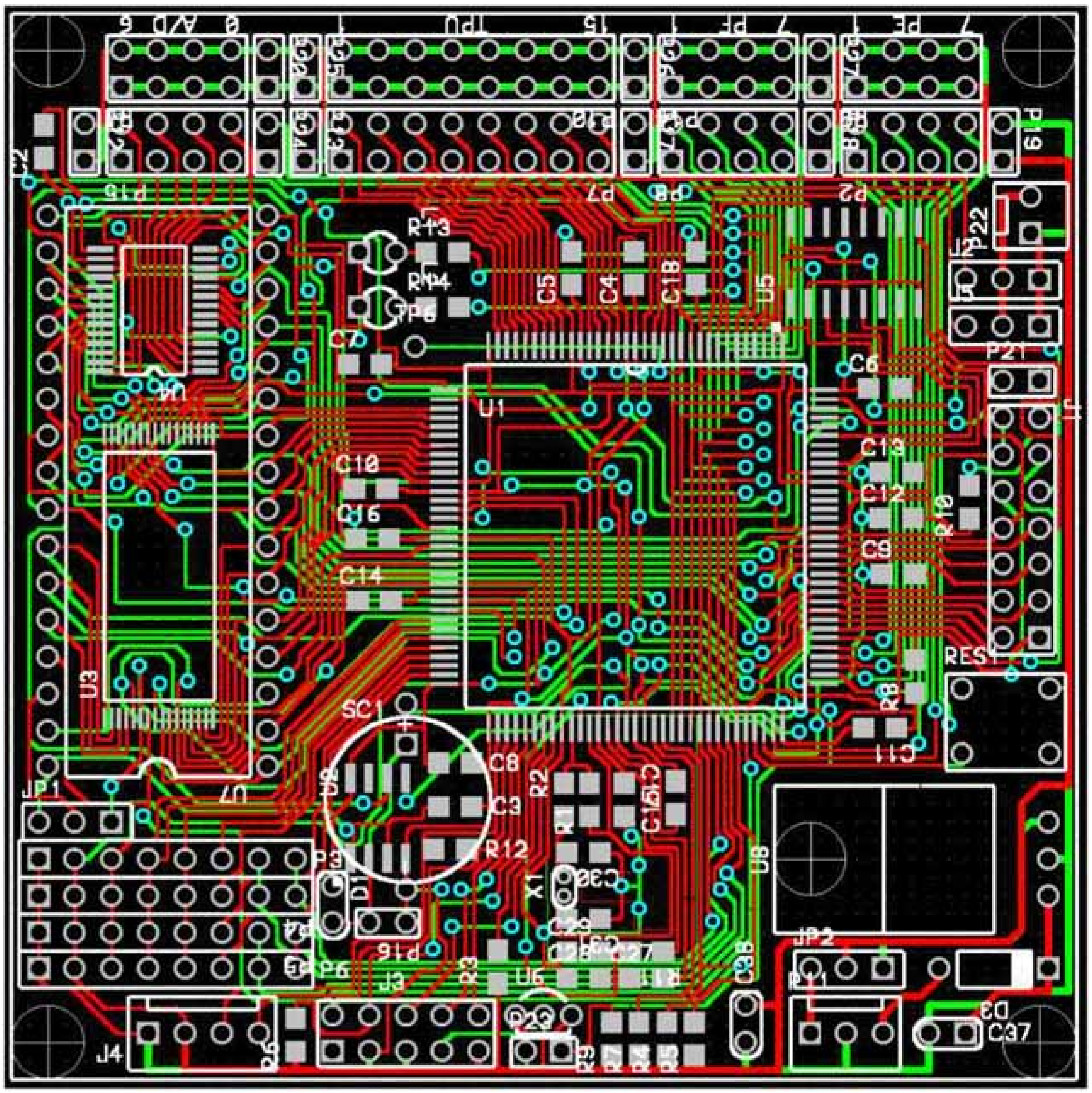
Mark Castelluccio's 68332 board design, the MRM332 (Mini-Robo-Mind) sits at the far end of the spectrum from the CRC332. While still packing the raw processing power of the 68332 Motorola processor, Mark's design focused on small, smaller, smallest implementation while still being manufactureable. Mark has taken this design to the extreme with respect to power conservation and size. This being Mark's second time designing systems with the 68332 processor his experience and technical expertise proved a significant benefit to the 332-SIG.

● 512K-Byte RAM ● 512K-Bytes Flash ● 7 channels AtoD 1 MHz acquisition speed ● 4 onboard SPI ports ● Onboard power regulation & reset ● Off board BDM cable interface ● Optional onboard RS232 level shifters ● E clock LCD interface ● RAM battery back up ● Expansion header ● 8 bit fixed bus width ● Small, Small, Small !!! - Size: 2.9" x 2.9" ●

One feature that really stands out on Mark's board is the high speed Analog to Digital conversion capabilities. With 7 channels capable of an excess of 1MHz data acquisition rate, the MRM332 lends itself directly to working with video. This board has been the center of a lot of the work being done with the game boy cameras. Much of this work is archived through the 68332AAB e-groups list server archive functionality.

Mark's board uses much more surface mount technology than either Jeff's or the author's. Much of the advantage provided around these component centers around the layout and use of the discreets, resistors, and capacitors. A second significant advantage in surface mount components came through the use of surface mount FLASH and AtoD converters. We'll look at this further in the article and especially how the combination of surface mount and through hole components greatly benefited Mark's design.

The PCB footprint layout below shows the level of integration to which Mark was able to achieve.



Mark Castelluccio's Schematic

Mark's_Schematic(171K) Mark's_Zipped_BDM_w/FLASH_Support_Code(3K)

Mark has also taken the time to fit his entire schematic design to one file rather than break it out by functionality. This file is viewable with the Adobe Acrobat reader as well. When viewing, note that there are five pages contained within, merely scroll down to see the rest of the design.

Section 2: CPU, Reset, Clock & LCD Connection

2:1

Jumping right in with page one of the CRC332 schematic, the reader can see the basic processor implementation along with the reset circuitry, clock and LCD interface. The processor chosen for this task is the Motorola MC68332GCFC-25. This part comes in a 132-pin PLCC package. Don't let this frighten you away from working with the 332 as the power of the processor is well worth the extra bit of hassle. Reference the author's previous article on oven soldering. Oven Soldering Encoder Article

Clearly evident in this first page are the address bus, data bus, TPU bus and chip select bus. All of the TPU signals are routed to header pins in combination with power pins (Ref section 9). The address, data and chip select lines are all distributed through these respective busses to the memory and IO components on the board that require them.

The Motorola manuals spend a lot of time explaining the use of the crystal oscillator and nearly ten discrete components required to buffer and properly filter the clock circuitry for the 68332 processor. The author opted to take a much more direct solution in his board design. As depicted in the schematic, the design proceeded with a canned crystal oscillator. The early prototypes built used the crystal and filter system, but these proved to have reliability issues during startup, and also showed to be extremely susceptible to solder flux, grit and stray capacitance on the board. When using a canned oscillator one must drive the /MODCLK line low coming out of reset, much like one of the data pins. If no other use is required of that pin, it can just be grounded. Using a canned oscillator assures positive startup each time without the delay associated with the PLL locking onto the crystal frequency. Also reduction of parts and eliminating the need for the high stability filter on the XCF pins as well as relaxed routing of the board, more than pay for the canned solution.

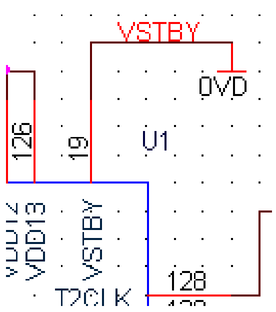
The design considerations involved in choosing a canned oscillator's speed centers on the mechanisms inside the processor that are based directly on a sub-multiple of the oscillator speed. Specifically, the required serial port speed, as well as many of the TPU timing functions, are based directly on the oscillator speed, so the Motorola documentation should be consulted rather than blindly choosing one out of thin air.

The reset part chosen was the DS1233D-10 part by Dallas Semiconductor. Much of the reasoning behind the choice of this part stemmed from availability issues. Again, the prototypes used an alternate surface mount part by Maxim that proved to be unavailable in quantities under 2,500. The reset signal is also brought to the edge of the board on a two-pin header so that the user may attach an external momentary pushbutton switch. The 820-ohm resistor shown on the schematic is not required when using the Dallas reset part. The /RESET signal is also run off of the board and to the BDM CPLD to aid in driving the data bus lines while the CPU comes out of reset.

The LCD connector shown here has the alternate wiring depicted, so one must exercise care when hooking up a generic LCD module. The wiring diagram is presented, below, in section 11. The chip select chosen to drive the LCD was /CS2. Note that the timing for the average LCD module is such that it needs 1-MHz or under address and data lines. Although the internally configurable chip select registers are indeed very flexible in that they allow for the insertion of up to 13 wait states on a 25MHz, this is not enough. To address this, /CS2 is run through the BDM CPLD to generate a much slower version /MCS2. The BDM CPLD is then also responsible for generating /DSACK0 to terminate the bus cycle.

If the user intends to not use the switched capacitor inverter discussed in the power supply section of this article, then the VEE signal present on PIN 1 of R1 on this page will not be available. To still be able to control LCD contrast, the user must then jumper VEE to VSS. On many LCD modules, the text will be just readable. For further design revisions, VCC should be replaced with VSS on Pin #3 of R1. This will allow the contrast to be varied between VEE and VSS, which is desirable. Additionally, if VEE is not used on the board, then the connection is made automatically.

The only three usable signals not brought out from the processor were left due to routing / real-estate considerations on the board and the other features offered in the extremely small space more than make up for this.



A last note on this page before moving on. The Battery backup for the internal RAM was not used. As such, VSTBY must be grounded. The excessively high current drain of the internal RAM precludes the use of a small battery to supply this current. The schematic snippet to the left defines this connection.

There are several discrete components on the PCB directly under the CPU. These components, visible in the bottom-stuffing picture earlier in this article, should be installed from the bottom side of the board after the CPU has been installed. Pre-bending and cutting the leads to just fit the mounting holes and soldering from the same (bottom) side as the component stuffing addresses this issue.

2:2

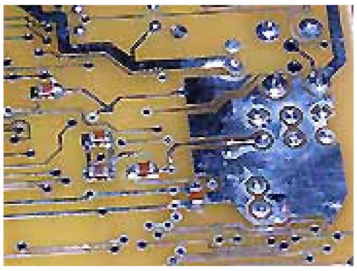
Jeff's board highlights one common thread amongst the three boards presented here, the processor used is the MC68332GCFC. Specifically, the G after the 332 in the part number indicates the internal mask set of the die used within this part. This specifies a specific set of pre-programmed TPU functions. These functions vary from mask set to mask set. The other common mask set is the 'A' set. It contains a set of functions geared towards the automotive industry where the 'G' set functions are geared towards the motion control industry.

Now lets take a few minutes and look at the CPU configuration on Jeff's board.

The first big feature that is evident compared to those discussed above is the clock circuitry. This design focuses around the crystal tuning fork, taking full advantage of the internal phase locked loop with internal divider to create the processor clock. There are some advantages to using this type of arrangement including cost and a large reduction in board power requirements.

Note that this implementation although costing less monetarily does cost more in PCB real estate using 11 discrete components. Jeff took the time to add the extra components for the high stability filter for the XCF pin. This involves a few additional parts above and beyond that recommended in the Motorola manuals. He has reported that after taking the time to wash the solder flux carefully from his board, the clock circuitry has been working well and starting correctly and repeatedly. The clock circuitry is made of up (C15,C14,Q1,R5 & R4) while the filter is made up of (C16,C17 & C18) and finally, the high stability components consist of: (C25,L1 & C24)

The photo (left) shows the careful hand layout work Jeff did in the area around the crystal circuitry including the small area of ground plane (both on the top and bottom of the PCB). Yes, this is still only a two-layer board so the clock oscillator circuitry must have top priority. Hats off for the attention to detail in this area. The photo, left, depicts this section of Jeff's board:



The next photo, right, shows the ground plane laid in underneath all of the oscillator circuitry. Along with this, careful decoupling has been done. Take a moment to appreciate the nice thick ground traces leading to and from this area!

Jeff brought all of the TPU lines to headers with easy access as well. This seemed to be a key theme, as we'll revisit this several times through out the article.

One way that Jeff really kept the layout of his PCB clean came through the use of SIP pull-up packages for many of the discrete resistors. The author started with this design and abandoned it when routing problems appeared. However, this design feature is vastly superior when it comes time to stuff 62 boards with fifteen resistors each vs. 1 pair of SIP's. Hands down, using SIP resistor packages is the better way to go.

IC1 in Jeff's schematic is the reset chip. A Maxim Semiconductor MAX705CPA part provides the reset functionality required by the processor. Jeff took the time to make PCB board space available to mount a reset switch directly to the board so that it can never be removed and lost at a competition. Further, a pair of header pins allows an external reset source or expansion board to be connected to the board. (Again, hats off for the good idea) The Maxim reset part does require the 820 ohm resistor discussed in the Motorola Doc's. These documents are available further below in the article, Section 17:1

Refer to the section on BDM and CPLD 2 below for a discussion on driving the data lines out of reset on all three boards.

Another common thread found amongst the three designs presented herein is the choice not to support the battery backup of the RAM internal to the processor. Here too, the reader can observe that Pin #19 on the 68332 processor has been grounded.

2:3

The Mini-Robo-Mind uses the 'G' mask set processor with the motion control TPU set. Mark used the output match function in the 'G' mask set for generation of PWM signals rather than the standard PWM or MCPWM functions provided in the TPU. For further information on this function refer to the authors previous encoder article on servo control with the 68332 processor.

The MRM332 uses the crystal tuning fork oscillator circuitry as well. Here, mostly surface mount components have been used focusing on 0805 packaging where available. Again, a large number of discrete components are required at the sacrifice of board space. One distinct advantage of this scheme is the availability of the clock pre-scaler functionality. The scaling functionality comes into play should the reader have the need to trim the processors actual operating speed to some value not dictated by a readily available canned oscillator speed. The Motorola documentation (Section 17 below) covers the setup and configuration of the registers associated with the internal PLL setup.

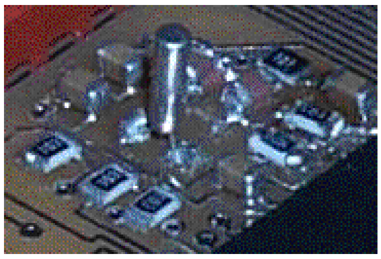
The reset part that Mark originally used is the incredibly small MAX___. This part comes in a SOT23-4 surface mount package and is produced by Maxim Semiconductor. The author used this same part in his prototype but discontinued its use due to availability issues. Mark switched over to the DS1233-D for his final shipping version of the MRM332. The board contains an onboard push button reset. This reset functionality is nearly a must during the software debugging cycle. Without it, the user will be cycling the power quite often. In the final version of the board the 820 ohm strong pull up recommended by Motorola on the /RESET line was not required just as with the CRC332 above. Header pins P23 bring a second set of the reset signals to provide access to either external reset circuitry or for the connection of an external push button reset switch that can be mounted in an easily accessible place on the robot.

Mark used yet a third way to interface to a standardized LCD module. While Jeff's board used software techniques through Port - F, and the author's board used the data bus and external DSACK generation, Mark's board, the MRM332, uses the E-CLOCK functionality of the 68332. E-CLOCK sources a synchronous clock cycle compatible with many standard peripherals still in use from the M6800 - 6809 days. This connection can be seen in the LCD connection section or on page one of Mark's schematic above.

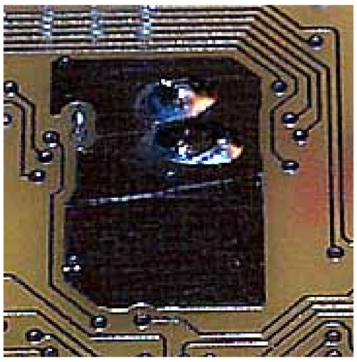
The MRM332's contrast connection is tied to the ground through R9 a 1K resistor. This does not offer any adjustment to the user. This will work just fine with most LCD modules used around room temperature. Again, the idea here is that the addition of this extra component (pot) takes up board space, adds weight and consumes power for a function that can operate without it.

The ECLK signal shares the /CS10 or Address 23 pin on the processor. In order to use the ECLK functionality on this pin the appropriate data lines must be driven low coming out of reset or software reconfiguration must take place after reset. (More on this under the BDM & Reset section below)

Mark's board also shares the non-support of battery back up for the RAM internal to the 68332 processor mostly because of the high current draw of that internal RAM which would quickly drain a long-term backup battery.

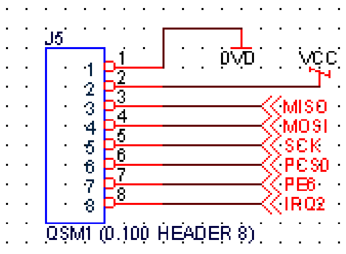


The image to the right is a close up of the surface mount components that make up the crystal oscillator and filter circuitry on the MRM332. Note the compact and careful layout practices used. This photo comes from the prototype MRM332. Mark has progressed to further minimize the PCB real estate required by the oscillator circuitry while managing to find room to lay down the crystal and wire loop it over a ground plane as well. The picture that follows demonstrates the ground plane and wire connections underneath the crystal section.



Section 3: SPI interface Ports & Real Time Clock.

3:1



The next section discussed is the connection to the SPI interface and the real time clock. The 332 special interest group that the author was a part of during the development of this board worked out a standardized pin out for the SPI ports. This standardization was done to allow the interchange of SPI peripherals. Note that two of the four ports have in addition to the standard SPI pin interfaces, an extra port 'E' pin and port 'F' pin. The port 'F' pins share functionality with the IRQ lines. If the port 'F' pins are used as generic IO pins then they may be used as is. If however the designer chooses to use them as interrupt sources, then an off board pull up resistor should be added. One of those output port organizations is shown: (diagram above left)

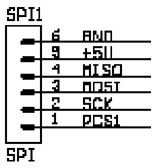
The real time clock circuitry on the CRC332 is provided via a Dallas Semiconductor 1302 part. This part is battery backed, and is interfaced to the processor through port 'F' pins in a simple software bit-banged protocol. The pull up resistors keep false data off the lines as the processor comes out of reset. Coming out of reset, both port 'E' & 'F" are set to IO functionality as inputs. This keeps any false triggering due to interrupts, invalid DSACK's, etc... from occurring.

A few notes about using the 32.768KHz tuning fork crystals with Dallas time keeping products. If the reader takes the requisite time to read and digest the Dallas Semiconductor data sheet for this part they may notice the small detail concerning a 6-pF capacitance for the tuning fork crystal. Only the good quality tuning fork crystals actually have 6pF specs. Most of the cheaper brands are out around 12pF or higher. Just as a note of caution, this two-fold miss-match can cause problems including failure to start the oscillator. Since the capacitance of the housing of the oscillator is involved with the actual power required to drive the oscillator, spend the extra $0.79 to buy a good quality crystal. Also, since the housing is part of the tuning circuitry itself, do not let the crystal come into contact with any other components.

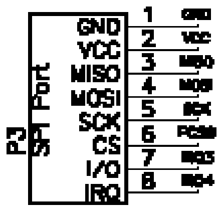
There are a great number of interface IC's that use the SPI interface including the 68HC68 line by Harris, which includes PWM's, AtoD's, IO ports, etc. The Vector 2X compass module by Precision Navigation also uses this interface.

3:2

Since the boards have this section in common the reader will note that SPI0, SPI1, SPI2 & SPI3 on Jeff's board have the same signals present in the same order. This affords the interchangeability of modules and support PCB's between the boards.



3:3



On the MRM332 you will see the same configuration of the SPI ports used in the author's and Jeff's board above. P3, P4, P5 & P6 are the SPI interface ports on Mark's board. They are grouped together to facilitate cabling the board.

These SPI connectors can be found on page four in Mark's MRM332 schematic found at the beginning of the article.

Section 4: Battery Backup, RAM, ROM & FLASH

4:1

Next to the processor page this, the third schematic page, is the most important in the design. This page outlines the RAM, ROM, FLASH, Battery Backup and all the associated chip select jumpers used on the author's 'C' Robot Controler (CRC332) board.

The two RAM chips are U5 & U6. These parts are 512K each. When only one RAM chip is used in the minimal configuration, it is stuffed into the socket for U5. This is done because the 68332 processor expects eight bit peripherals to send and receive data on the DATA[15..8] lines. Also, when using only one RAM chip with the board, J10 is set to position 1-2. This is done because both odd and even addresses must gain access to the same chip. In contrast when using two RAM chips, the jumper on J10 is moved to the 2-3 position. In this way, one RAM chip supplies the odd address and the other the even address. As such only even addresses need to be supplied to the RAM and A0 is not needed.

Battery backup functionality is provided via the DS1210's. These two parts serve a dual role in this design. First, when the system power falls below a preset limit they switch the power of the RAM chips over to the battery. Secondly, these chips switch the chip select from a 'pass through' state, to an output that is pulled high to the battery voltage. This is important, as the RAM chips can not get into low power mode unless their respective chip select inputs are held high.

A note of importance regarding RAM power and battery life: When purchasing RAM chips, insist on the '-L' suffix as a bare minimum. This insures a regulated maximum of current draw while the part is in low power, battery backup mode. To further increase the battery life, use '-LL' suffix parts. The '-LL' stands for Low-Low Power Backup, providing for additional savings in battery life. Also in the interest of extending battery life, use only certified ceramic chip capacitors for the decoupling caps on the VBAT power trace. Most designers overlook the fact that capacitors have significant leakage when compared to the current drain of a battery backup designed to last many years. The wrong choice of capacitors will take a five / seven-year battery backup design down to a few months.

Although 16 bit wide mode is present for the RAM, and /CSBOOT is available as a chip select source for both RAM chips, the CRC332 can not perform a 16 bit wide boot from ram operation as 16 bit wide access requires two chip selects. A small boot loader in either FLASH or EPROM must enable the 16 bit wide chip selects and then jump to code within these parts. Jeff Bronk has already developed this boot loader and it is presented under the software section later in this article.

U7 is the onboard FLASH part. It is 512K-bytes in size and is implemented for use as an 8-bit wide memory device. A jumper placed between pins 1-2 or 2-3 on J11 selects the chip select source for this device. The system can boot from U7 or it can be configured to use /CS5. All programming voltages and algorithms are present within the device for in-system programming from user supplied code. Although several companies make substitutes for this AMD part only the AMD part has been tested. The data sheet for the ST-Micro part appears to show that it can not drive the bus capacitance at speed. Also take special note that the setup and hold times for these parts differ from those for the RAM chips and the chip select configuration in the SIM registers needs to reflect this point.

U8 is the onboard EPROM. Here two access times must be observed when configuring the SIM chip select registers due to access speeds of the parts. In the original CRC332 systems this device was set up to access /CSBOOT since the first 20 bytes were zeroed out. This permitted the 68332 to enter BDM mode where the user could modify SIM registers to enable the RAM chips, and download code to run. This may seem a bit obscure, but the processor must see valid boot addresses and stack pointers in the first eight bytes of memory (under 1 meg) when it boots or it will hang with a double bus fault. If battery backed RAM, a volatile memory source, had been used for /CSBOOT it is more likely than not that many of the boards would have arrived in non-functional states. J7 allows the user to change the EPROM from /CSBOOT to /CS4.

Both U7 & U8 are mounted to the bottom side of the board. To facilitate this while still using through-hole components, sip sockets were installed in an interleaved fashion on 0.100" x 0.100" centers.

4:2

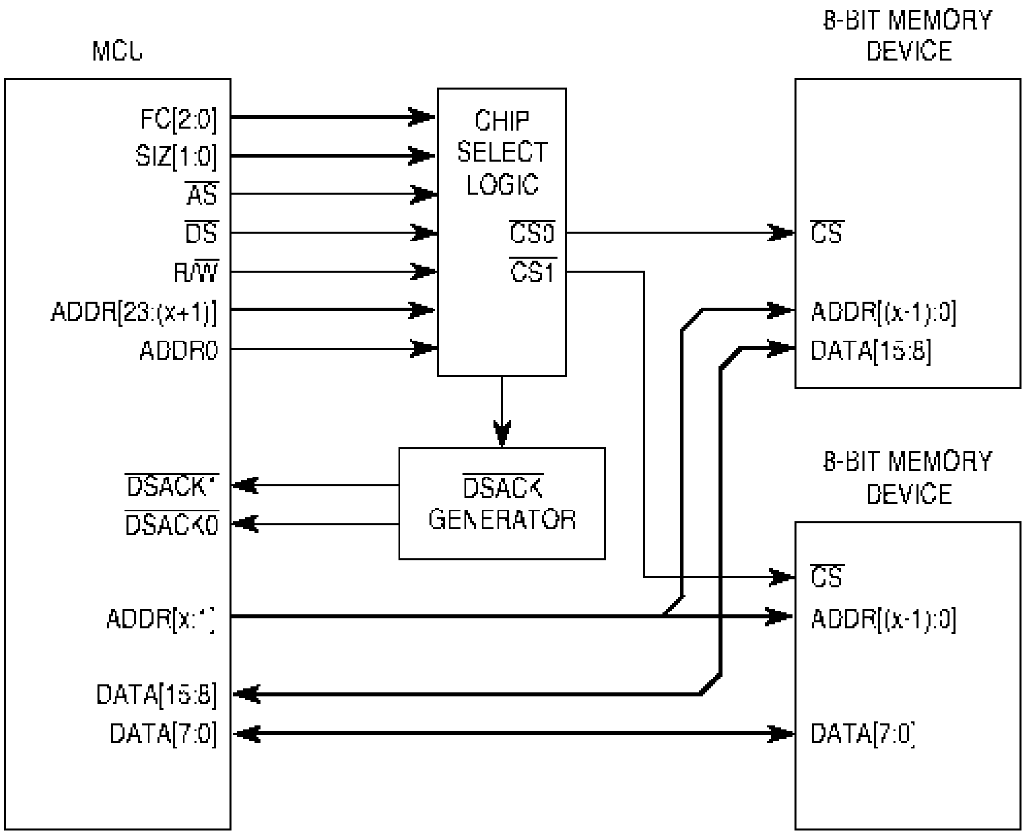
Careful inspection of Jeff's schematic shows U2 & U3 hooked up to provide 16 bit wide RAM accesses as well. Where the design significantly departs from the CRC332 is in the use of CS2 and CS3 (both active low) to drive the output enables of both RAM chips. Through the configuration of the System Integration Module (SIM) chip select signals can be configured to system reads, system writes, interrupts or combinations of each. In this manner /WRH & /WRL signals can be generated.

Jeff to took advantage of bottom mounted components to maximize PCB real estate. The second RAM chip on his board is bottom mounted, using sip sockets to allow access to the staggered pins of the top mounted RAM's solder pads.

U4 is an EPROM hard wired to /CSBOOT. This allows dedicated boot code to set up the RAM chips and then jump to them in 16 bit wide mode.

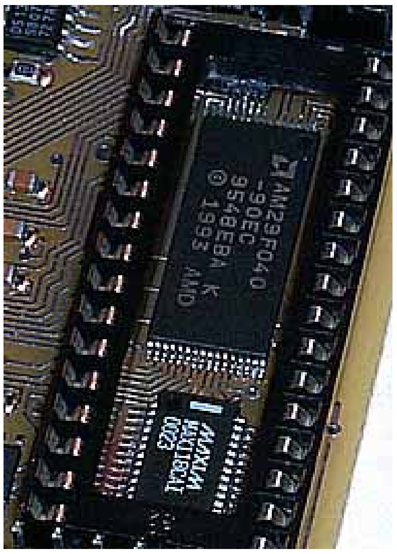
U5 & U6 are the Maxim Semiconductor battery backup parts equivalent to the Dallas parts discussed above. These parts switch battery power between VCC and VBAT and also hold the appropriate chip select high during battery backup mode. The Dallas and Maxim parts are drop in replacements for each other. During the design phase of this project the members of the 332SIG discussed using only one battery backup chip and small MOSFET's based on it's output state to switch battery backup and chip selects for several RAM chips. We decided against that solution when we compared the current draw of these dedicated integrated IC's with the current draw we could achieve with our discrete solution.

The schematic depiction (below) comes from the Motorola CPU32 manual. It shows the connection of two 8 devices to the 68332 processor in parallel. This can be contrasted against the connection of a single 8-bit device presented in section 4:3 (below). Note the exclusion of Address-0 and that Data [15..0]are used in a linear fashion.



In the diagram above note the generation of two separate chip selects. This is important such that the processor can write an 8-bit value to one or the other device without clobbering the memory stored in the odd/even byte sharing the same address.

4:3

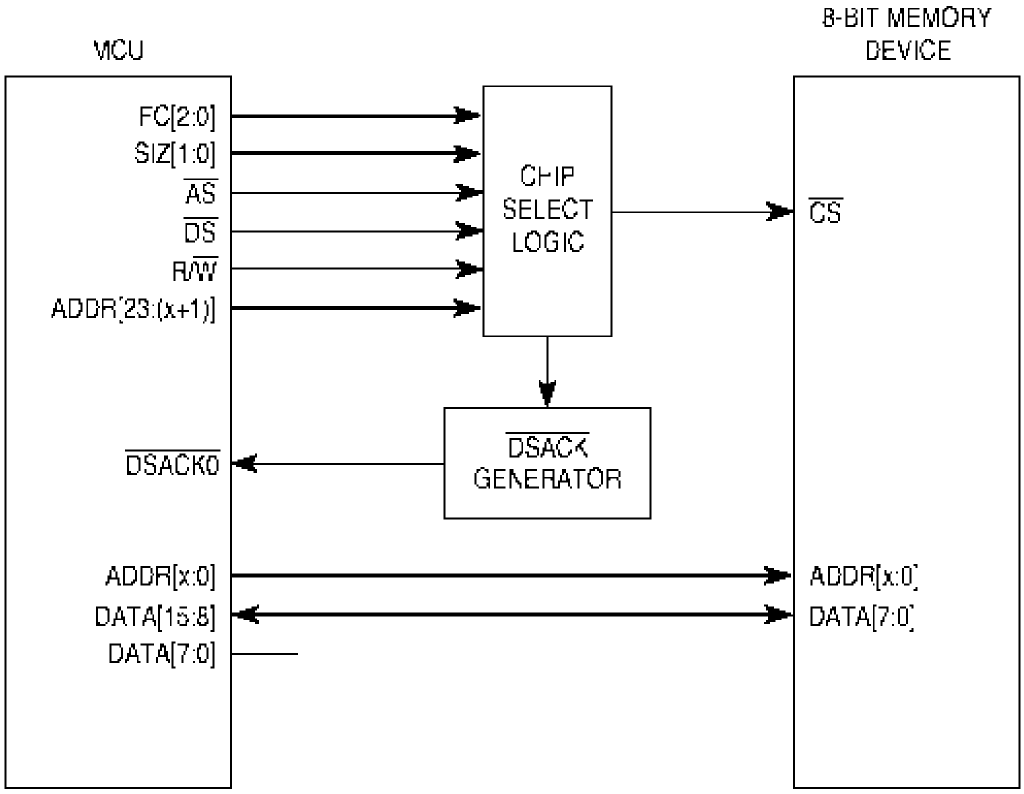


The MRM332 uses the 8 bit data path size to enhance it's small foot print and economy of real estate. U7 is the 512K RAM on the board. Mark uses a processor chip select to generate the CS0/BAT signal for this IC.

Battery backup is afforded via U2. This 8 pin surface mount IC by Maxim (MXD1210 in the S0-8 package) allows for clean disconnect of the /CE signal and switching between VCC and V_BAT.

U3 is the 8 bit flash chip used on the MRM332 board. Mark chose a surface mount part in the TSOP-32 package. One step finer than the processor itself, the pin spacing on this IC requires a steady hand when clearing solder bridges. The packaging on this IC is worthy of extra note due to Mark's creative mounting of this part (and the AtoD which we'll investigate further below) underneath the socket for the through hole RAM chip. This is evident in the layout image (left):

The eight-bit memory interface used on the MRM332 saves lots of valuable board space that would otherwise be lost to signal routing. The depiction below comes from the Motorola CPU32 manual and illustrates connection between a memory device eight bits wide and the 68332 processor. Note the use of Address-0 and the fact the D[7..0] on the memory device are connected to Data[15..8] on the microprocessor. Contrast this with the 16-bit implementation above.



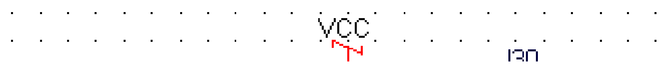
Section 5: Power Supply & Onboard Inverter

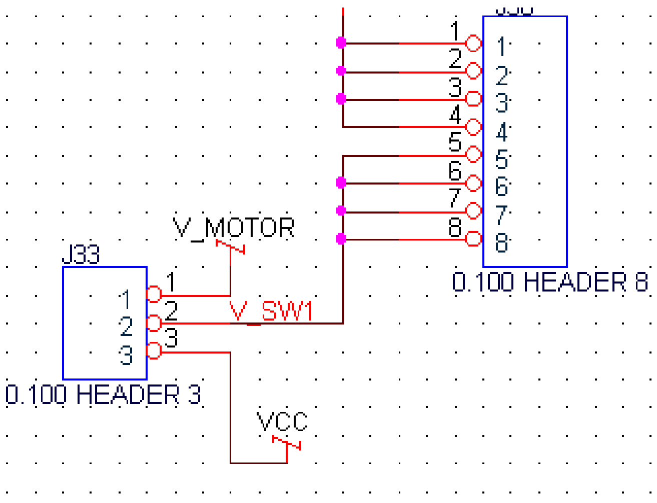
5:1

Page 4, of the 'C' Robot Controller, the power supply depicts the linear regulator and the inverter. The single biggest upgrade any user can make to the CRC332 system is to drop in a TI (formerly Power Trends) switching high efficiency 3 terminal regulator. This simple change can more than double the life span of a 9v battery.

J15 is the main power input connector for the board. An inline diode protects from accidental backward connection to a power supply. J13 is a secondary power connector used to provide power to servomotors and other noisier or comparatively high power devices. In amongst the header pins that supply power on the board (for example next to the TPU lines) are jumpers that allow the user to select between supplying VCC and VMOTOR. This means that a second battery source or power supply can supply power independently to these header pins through the J13 connector. Simply put, a second battery pack for operating servomotors could be connected to J13 and all the power pins at the TPU connectors would come out in the right order. Moreover this removes the likelihood of damaging the processor from inductive spikes or current inrush when dealing with servos / steppers / etc...

J14 provides a pseudo second supply to VMOTOR.





When this jumper is inserted, then no external power supply should be connected to J13. This jumper connects the raw unfiltered power input from the battery pack, present at J15, directly to VMOTOR. In this configuration, one battery pack can supply power to the processor through the onboard regulation system and additionally drive servos in a safe manner. If the user opts to use this configuration it is recommended that C14 be paralleled with a 12pf cap or that a fast acting battery pack such as Ni-CAD's are used.

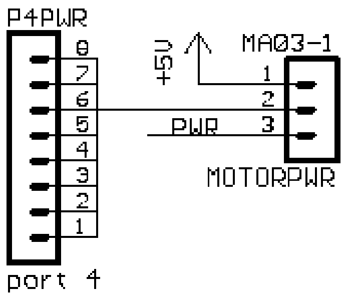
The schematic snippet above echoes the design configuration idea of switchable header power used throughout the board. Much of this functionality is afforded by the careful consideration for placement during the actual layout process. The schematic snipped above comes from the vertical stacking and TPU interface page. It is shown here as it has bearing on the V_MOTOR power supply issue. Switchable power connectors placed along side signal header pins offer ultimate flexibility.

U11, C12 & C13 make up an onboard-switched capacitor inverter. This system provides two functions. First the negative bias to drive the LCD's contrast pot is derived here. The -5v supplied to the contrast adjust allows for crisp dark text on most one and two line LCD displays. Secondly, the -5v generated by this inverter is brought out onto the eight header pins associated with the analog to digital converter. This is useful for powering dual rail operational amplifiers, sample and hold registers, CCD's etc... A surprising number of analog sensor implementations become open with this VEE supply available. Note, most of the IC's on the board can not tolerate to be shorted to a negative voltage for even the smallest amount of time. If the user does not intend to use this part simply pull it from the socket. In designing one's own board, I highly recommend including the space for it, as down the road it will save you from building a separate power supply.

Lastly, a power out connection (J12) consisting of VEE and VSS is shown on this page. This is intended to provide a convenient access point to the VEE signal and is used in conjunction with the IO expansion board (below). This connector is physically located at the low left corner of the AtoD converter. Keeping all of the traces that contain negative voltages close together and isolated in the corner of the board was a key design issue. This was done to help reduce the chances of accidentally shorting out a negative voltage to any other component on the board.

For my uses I'm using an external switching regulator capable of souring 3&1/2 amps at 96% efficiency with a lot of extra front end interference filtering. In this application, the linear regulator and it's associated diodes (U12,D1 & D2) are not required and a simple wire jumper on the bottom of the board connects the power input header pins to the regulator output terminal. (U12 Pin #3)

5:2



REG1 is the component that provides voltage regulation for Jeff's board. Using the same linear three terminal regulators a simple but effective approach to the design power requirements is met. C12 gives a large amount of input filtering and C13 the requisite output filtering. An LED - resistor combination provide clear evidence that the board is indeed powered up. A three pin power connection with the outer two pins both carrying the ground (GND) supply keeps the user from accidentally connecting the power pins backward. I.E. foreword or backward, the three pin connector works both directions.

Jeff too used power distribution jumpers to allow the user to select which power supply was available at the header pins along with the TPU lines. In the image (right) the three-pin header allows the user to jumper in the boards internal power supply or PWR a tap from the units incoming power before the linear regulator. This functionality will go a long way towards keeping the board running correctly when using it with Servo's and other electrically noisy end effectors.

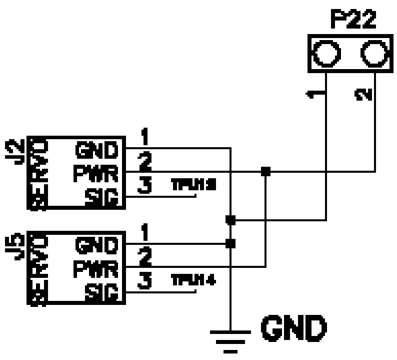
5:3

Yes, another linear regulator. Drawing only 90mA this is quite an acceptable solution for the MRM332. Still, to increase battery life, a high efficiency-switching regulator could take the design from 60% to 96% efficient. This means less heat and longer battery life. The trade off, of course, is size as the switching regulators are larger and cost. Many of the commercially available units are in the $19 range.

The MRM332 uses U8 to provide the regulator functionality. Going a step further than either Jeff or the author, Mark managed to make space to lay the regulator down on the board, even with his super small board foot print. This makes it easier for any potential robot builders to drop his board in and go without having a large warm regulator sticking up.

The Mini-Robo-Mind also uses diode protection via D3. This feature is an absolute must when supplying boards to end users who will eventually hook up the power backwards. An ounce of prevention.... In early prototypes, the regulator was mounted vertically. Laying it down is a dramatic design improvement. Way to go Mark!

The power input P11 in conjunction with the power selection header JP2 offer the reader access to the onboard regulator or the option to supply a pre-filtered +5VDC supply.

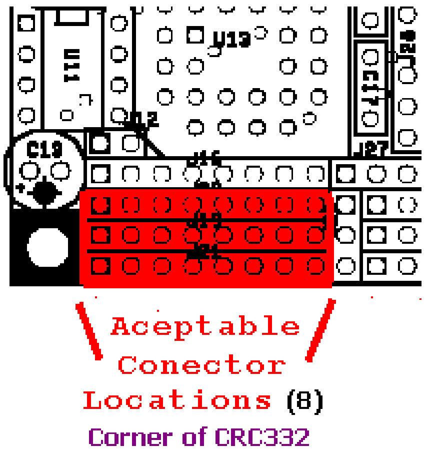


The image to the left shows Mark's commitment to use of the MRM332 as an embedded robot controller. J2 & J5 are both set up to provide special access to an alternate power supply for use with servo motors on these two special servo motor connectors. P22 is a port provided to bring in the extra power connection for the servo ports. Features like these keeps the processor power supply clean from the noise fed back down the power lines from large inductive loads and once again highlights Mark's forethought in setting up this PCB.

Section 6: Analog To Digital Converter

6:1

The analog to digital conversion section of the schematic is depicted in the 'C' Robot Contorllers fifth schematic page presented above. A to D conversion is facilitated by U13 a National Semiconductor ADC0848. This 8 channel single ended converter was chosen for its decent response time and ease of processor bus implementation. The sample code that shipped with all versions of the CRC332 continuously displayed the values present on all eight channels. Note the pull up resistor on /IRQ1.



Probably the most important features on this page are J16, J17, J18 & J19, specifically, the grouping of these header-pins on the board. The outer pins of course being ground (J17) to protect against the accidentally dropped screw driver, etc. Working inward from the board edge, the reader would find a signal pin from J19 followed by a VCC pin (J18) and finally VEE supplied by (J16). This arrangement provides easy access and connection for sensors, joysticks and other analog data sources. Connections to these pins can be seen in the photo that follows:

One note, when using external dual rail operational amplifiers with VEE. The analog to digital converter is single ended. I.E. its inputs must see values between 0 & 5 volts. When working with negative voltages, use an inverting amplifier. When working with AC voltages, a DC offset must be added to keep the most negative point above 0 volts. This can later be corrected in software. These considerations do not affect positive voltage sources or ratiometric devices such as sharp distance sensors, joysticks etc...

When the reader lays out his/her own board, if you choose not to use the supplied artwork later in the article, make sure to decouple the analog to digital converter correctly for best results. Since this is a single ended converter it is easy enough to simply run the digital VCC directly into the part. Careful decoupling and a separate VAA line directly from the regulator can go a long way to increase the accuracy returned in the least significant bits of a reading. Parallel decoupling of 0.1uF along with 12pF caps will increase performance.

6:2

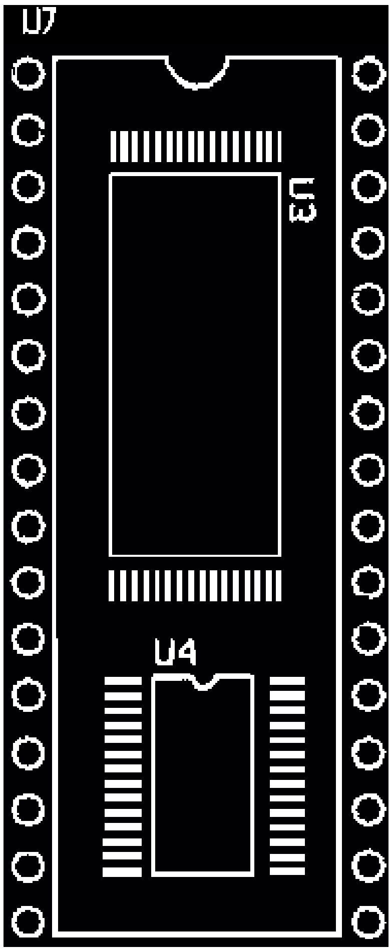
U8 is an ADC0848 analog to digital converter used on Jeff's board. The author based the AtoD functionality on the CRC332 on this implementation by Jeff (Thanks Jeff) Note the power decoupling right at the part. Also note the arrangement of the power and ground header pins made available with the ADC inputs.

Analog to Digital implementation is another one of those places where Jeff's board departs vastly. His implementations of the Chip selects to drive the /CS input as well as the /WR and /RD lines of the AtoD converter show a differing design philosophy. In so doing, Jeff did not require extra glue logic to generate the write signal from the read (/RD) signal presented by the processor. Again, another good move.

Jeff chose to use the DIP package in his board layout, which facilitated connections to the DATA lines. It also saved over $1.78 for the PLCC adapter socket required for the CRC332.

6:3

U4 the Analog to digital converter on the Mini-Robo-Mind (MRM332) is a significant departure from the two designs above. Mark choose a MAX1185 by Maxim Semiconductor to provide 7 channels of high speed (1M-S/s+) analog to digital conversion. U9 provides latched address buffering to keep the address inputs to the AtoD converter stable throughout the entire conversion process.



Analog to digital conversion speed is application specific. When listening to a microphone for audio or tracking a potentiometer, high conversion rates are not required. The MRM332 however has found a specific niche with its high speed conversion and is quickly becoming a favorite amongst those working with the game boy camera for digital imaging. The speed of data conversion is high enough to bring frames into the board at a rate of about 30/second.

The reader can follow the link below to the 68332ABB e-groups web based list server. If the reader then takes the requisite time to join they will be granted access to the files area where source code used on the MRM332 to interface to the Game-Boy camera is stored. Additionally, the message archives will become available when the reader joins this group. The message archives have a large number of game boy camera related threads.

68332ABB e-Groups site

Another great use of space showing the hybrid use of surface mount and through hole technology is the placement of the analog to digital converter next to the flash underneath the socketed RAM in the diagram (right).

Looking clear back to the general overview section where the top down view of Mark's board is presented the reader will appreciate the proximity of the AtoD converter (U4) to the header pins (P15) that bring in the signal to be digitized. The short direct path again points out the careful layout planning present in the MRM332.

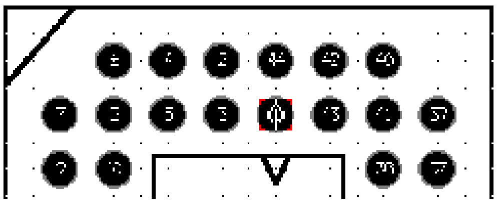
Section 7: CPLD-1 PIA Function

7:1

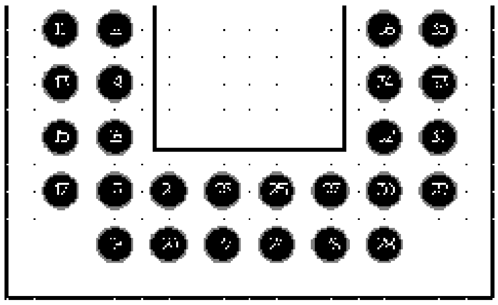
The sixth schematic page of the 'C' Robot Controller presents the first CPLD implementation. When the CRC332 shipped, this IC was configured to implement a simple PIA functionality. This part offers some of the real power available on the CRC332 platform. With the software tools available for download from the Lattice web site, this part can be reconfigured to implement just about any IO function imaginable. (PWM, I2C, RS232 etc...)

Earlier the author presented an article to working with CPLD devices that may be found at the following link CPLD Encoder Article. Armed with the information in this article and a download cable (presented in the article as well) the user may use J28 to reprogram the part directly from their PC's parallel port. The entire source code for the PIA implemented on the CRC332 is present in the article in the preceding link.

Much like the section above on A to D conversion, the order and placement of J20 - J25 plays a significant role in the ability of the CRC332 to interface easily to external devices, either as complex semiconductor functions or simple switches. J26 provides an external clock input and J27 switches between this clock source or an internally generated clock source via TPU channel 1. This may seem like a trivial connection arrangement, but with the ability to switch clock sources combined with the CPLD's re-programmability many implementations will be realized.



A note about the choice of PLCC adapter sockets. These sockets give one the competitive edge when trying to cram an extra RAM, RS232 driver and two more 44-pin packages onto ones board than comparable boards in close to the same form factor while still sticking to two layers. The pin configuration is shown in the diagram (left).



Also note that the depiction to the left is a top down view. An often-overlooked point when working on the bottom of the PCB is that the pin numbers go clockwise (opposite those shown).

In presenting the information fully, it is the author's duty to drive this one home one more time. The PLCC packages are keyed. When using these adapter sockets it is easy to defeat this keying by simply applying an additional pound of thumb pressure when installing the IC. Please take extreme care when installing the IC to align pin #1 correctly in the socket. Furthermore, extreme care must be observed when stuffing the socket into the board, as there is no keying mechanism between the board and the socket. The artwork has pin number one built with a square pad and it is the reader's obligation to make sure they line up before soldering the socket in place.

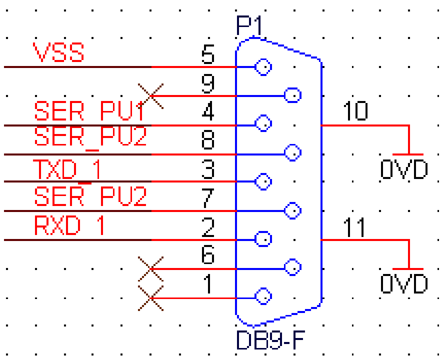
Neither Jeff's board nor the Mini_Robo_Mind offer this type of IO expansion due to the space constraints placed on the boards by their respective designers.

Section 8: RS-232 Level Drivers

8:1

U15 is a simple RS232 level shifter. Many other boards leave the level shifter off the PCB in order to save real estate and require a special cable with this part and a few discreets on a small PCB inline within the cable. Although this does make the PCB a bit smaller and less costly, I chose to include it rather than make the user buy a separate cable and PCB assembly later. This allows the user to use any old, 'off the shelf' nine pin serial cable with the CRC332. The DB-9 hood also takes up a substantial amount of board real estate.

To help offset this down side, I've mounted U15 to the bottom side of the board underneath this hood. The only trick to doing this, is that the socket to U15 **MUST** be installed before the D-sub hood. Additionally, the choice of socket for U15 becomes important. A regular DIP socket will not work, as it would obscure access to the solder pads for the D-Sub connector after installation. To skirt this problem a pair of 10 pin sip sockets are soldered down leaving access to the 9-pin shell between.



Quite commonly the pin-out of a package changes depending on the foot print of that physical package. The MAX233 is one of these devices. Specifically, take note that the pin numbers change between package types with this IC. The 233 was chosen due to the fact that it has internal charge pump capacitors which greatly reduce board real-estate and that it is second sourced by many companies. (Analog Devices, Sipex, Maxim to name a few)

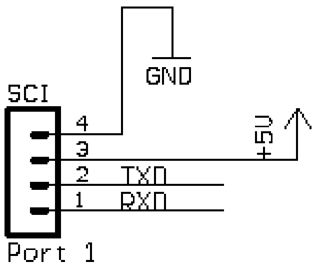
The diagram left depicts the connector pinning:

8:2

The SCI connector labeled as Port 1 on Jeff's schematic brings out TTL level RS-232 signals from his board. These pins are arranged with power pins to line up with a cable kit sold by Kevin Ross (SRS member) to convert the TTL level signals to RS232 buffered level signals. Here again, this is a different design philosophy to get the hardware off of the board.

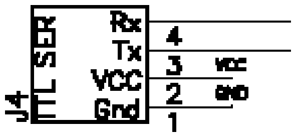
One advantage provided by this solution is that once removed from a robot, by disconnecting the cable, the charge pump in the RS232 level shifter does not have to be powered by the robot's batteries while the serial port is not in use. This extends the life of the batteries and lightens the board.

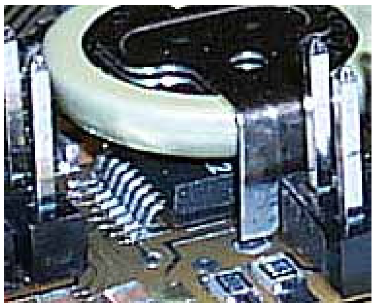
The schematic snippet with Port 1 visible (right) shows the correct pin-out for the TTL interface of the cable assembly from Jeff's schematic.



For information on cable kits available refer to section 8:4 (below) for a link to Kevin Ross' web site where these units are available.

8:3





In the first prototype stages of the MRM332 Mark used an off the board in the cable serial RS-232 level shifter. The production version of his product now offers an on board option for RS-232 drivers. This part is so small it's hidden most of the way under the battery, which can be seen in the photo to the right. Mark sure has his work cut out for him when soldering these boards together. Much like the CRC332 assembly must take place in a specific order.

The connector shown above (J4) and discussed above brings RS232 level signals out of the board. Mark went one step further by allowing the inclusion of R16 & R17, a pair of zero ohm resistors. These two resistors can be used to bypass the RS232 transceiver if it is not installed for use with Kevin Ross's in the cable RS232 solution which you can see below.

8:4

The link below will take the reader to Kevin Ross's web page where they may locate the RS232/TTL kit that provides the 4 pin TTL to RS232 9 Pin Serial connection.

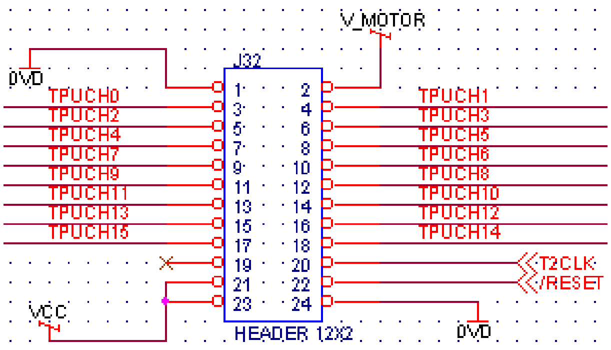
Kevin's Product Page

Kevin Ross also provides this incredibly informative PDF information file that describes building the level shifter, assembly, testing and at the very end includes the schematic. I highly suggest that reader reviews it carefully and then buy it from the link above as it is a sound deal at a good price. Thanks Kevin!

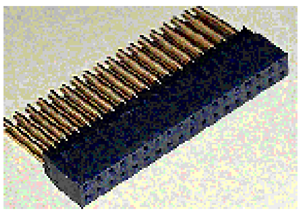
Kevin's RS232 level shifter document

Section 9: Vertical Stacking Bus & TPU breakout

9:1



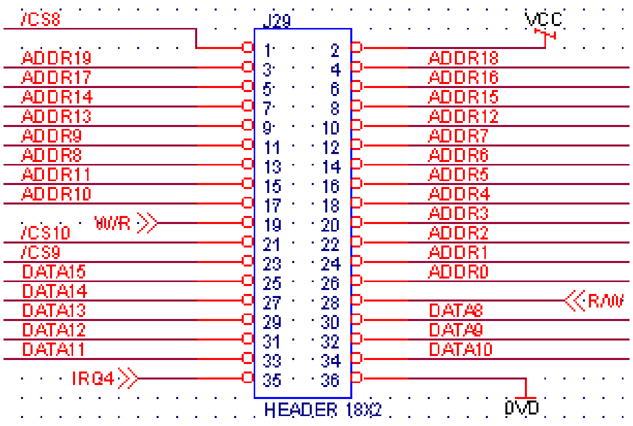
Schematic page 8, of the 'C' Robot Controller, depicts the breakout of the TPU lines and the vertical stacking expansion connector. The vertical stacking connector used a special 36-pin version of the gold plated PC 104 connector. The use of this rather expensive (~$5.00) connector seems to be a bit of over kill on my part until the reader considers what type of signals and the speed of those signals that are traversing this connector. Then consider the vibration and mechanical stresses present in the target environment and the choice becomes clear. Specifically, this connector has three sets of spring loaded contact leaves pressing on all four sides of each pin for high vibrational loss resistance.

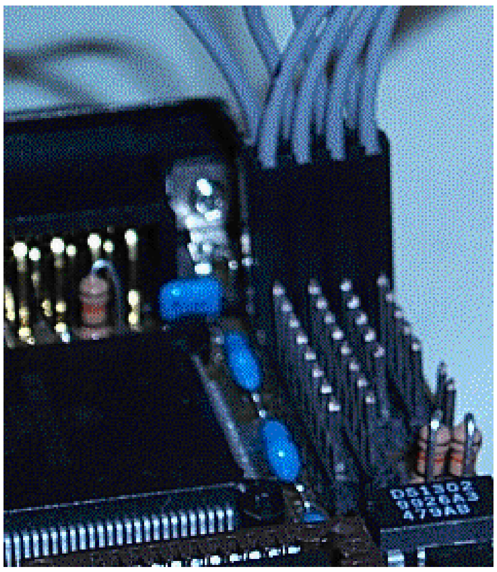


The photo (right) shows the specialized 36-conductor PC104 style connector with deep gold plating and four-sided contact. This connector is made by Samtec and can be found in their catalog under the board inter-connect section. This connector carries address, data and bus control signals for external expansion.

In a later section we'll look at some of the peripherals designed to work with this expansion connector. It may seem limiting to only have three chip selects available on the connector, but with the simple inclusion of a small (~$1.79) PLD on the peripherals PCB, these chip selects can be further broken down based on the memory size required by each device. This is the scheme the author used with his peripherals expansion boards.

One advantage of using the vertical stacking buss in the way it's been designed come through the length of pins selected. In choosing the correct length, the buss may stack both upwards and downward from the CRC332.





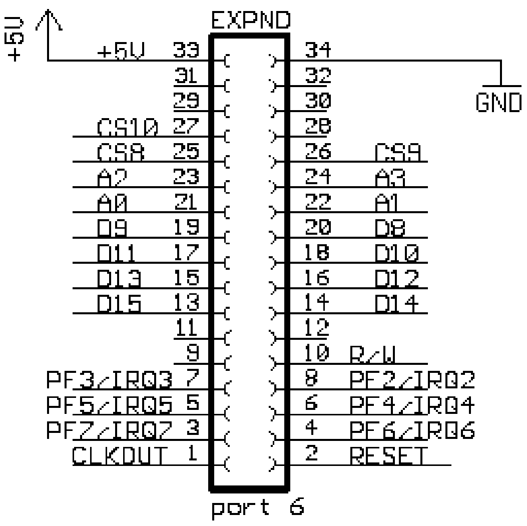
J32 shows the breakout of the TPU lines from the processor. Here, too, the placement of J30 and J31 complement the functionality of the TPU. With each pair of TPU pins having access to both power and ground on the same connector, driving peripherals couldn't be easier. For example, the connection of an encoder requires the user to simply hook up the wires to a single four-pin connector. J33 offers some of the functionality discussed previously in the power supply section. Shorting pins 1-2 on this header can provide the VMOTOR supply to half of the pins on J30 instead of VCC. A useful feature when the user wants to drive servo's directly from the board. Again, the grounded pins of J31 are on the outside of the board to protect against accidental screwdriver drops.

The photo below shows the connection of a DC PWM amplifier for brush style DC motors and also an encoder. Both of these cables supply power and ground to the target device.

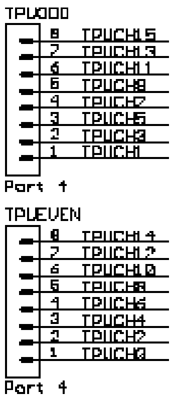
For ideas on interfacing motors using PWM or encoders to the TPU lines, refer to the author's previous encoder article. 68332 Servo Control Encoder Article

9:2

The expansion port and TPU headers on Jeff's board were implemented as follows:

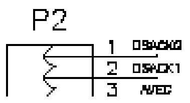


EXPND - Port 6 (Left) brings several address lines, eight data lines all of the port F pins, the system clock, reset, R/W, and three chip select lines to an expansion header. This is a powerful and well thought out mix of signals to allow for IO and peripherals expansion. Coming from a different design philosophy again, Jeff chose not to pack all of the extra Address lines that the author did onto this expansion port and rather made sure all of the CPU's Port-F was available. This leads the user directly towards IO and peripherals expansion rather than memory expansion. The reader must weigh whether 1.5 megabytes is really enough. If it is, then this type of expansion may be for you.

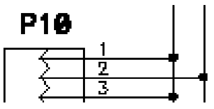


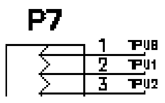
Port 4 consists of connectors TPUODD, TPUEVEN, P4PWR & P4GND. These header pins grouped together show the attention paid to board layout and the thought put into the types of devices that will later be plugged into these header pins. Here, too, Jeff has provided the ability to switch the power supply used to supply these pins allowing for servo power to come from sources other than the regulated digital supply.

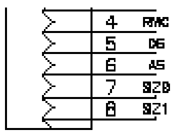
9:3

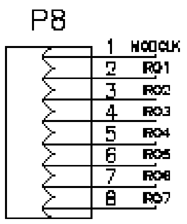


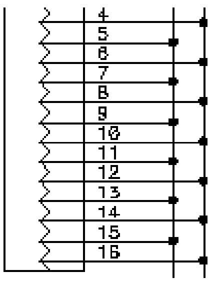
The MRM332 brings all of the TPU lines out to expansion headers with power in much the same fashion as Jeff's board and the author's. A significant amount of PCB real

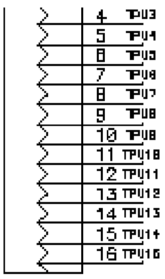












estate is dedicated to the placement and routing of these pins on all the boards. The functionality gained however is well worth it as these TPU lines offer some of the real power on the 68332.

Due to the extreme of space constraint on the MRM332 there are not as many configurable jumpers for setting switched power options. This is offset by the ease of use not having to sort through all of the documentation and extra fore-sight put into the design by Mark up front in determining exactly what would be needed by the average user.

The MRM332 also brings the all of the port 'E' & port 'F' signals to header pins through ports P2 & P8 respectively. These pins offer the user access to the interrupt pins and other bus control signals or when configured, direct software IO access to 16 bits of dedicated IO. These can be seen in the graphic to the left. A handful of other system critical signals are available on headers including several chip selects and low order address bits.

Section 10: CPLD-2 BDM Driver & Reset functionality

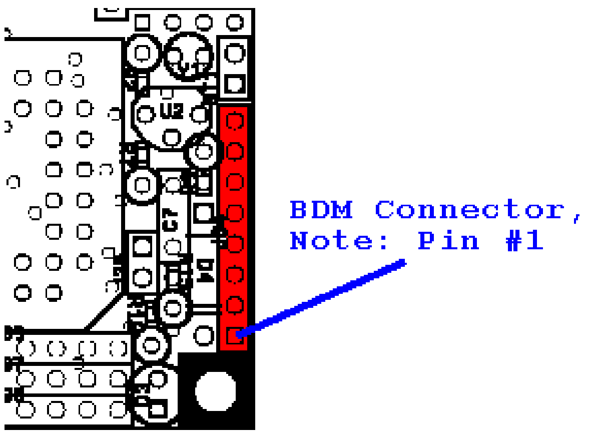
10:1

Refering now to schematic page #9 of the 'C' Robot Controller: This page depicts the implementation of the second CPLD on the board. Another 44-pin part and yes it still fits the small foot print. This part provides several sets of functionality. When the board ships, this part is responsible for generating the delayed /MCS2 for the LCD screen and terminating the bus cycle.

The single largest function that this part provides is the onboard BDM function. Just like the serial driver functionality, this onboard BDM function keeps the user from having to buy a separate PCB assembly and IC's to interface the PC's parallel port to the 68332's BDM pins at a later date. The 25 pin D-sub on this page is merely for reference to aid in construction of the wire only version of the BDM cable.

The other members in the 332SIG opted to use Motorola's standard BDM implementation for two reasons. 1) Size. Since the components are on a separate PCB at the other end of a cable, they do not take up any board real estate. 2) Design time. The author spent a considerable amount of time reverse engineering the BDM functionality and ended up working through several sets of documentation that are slightly lacking. Much of this development time would have been better spent building robots... Both of these alternatives have strong merits and make Mark and Jeff's boards good operating platforms. However they did not fall into the design criteria of the CRC332 to have all required hardware onboard. The added real estate takes just over 1 square inch of PCB, a rather high 'cost' on a 3.375" x 3.9" board.

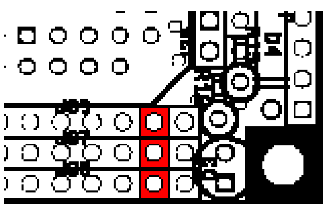
In the photo below the reader can see the connection used for BDM.



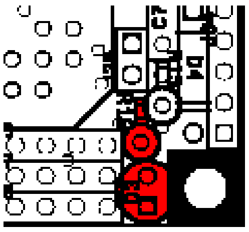
Corner of CRC332

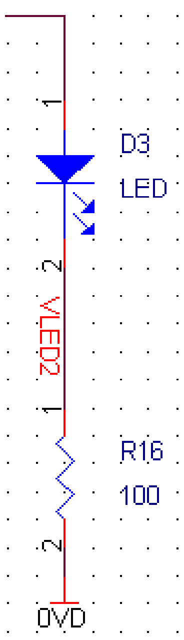
This part is also responsible for the generation of the W/R signal. This is simply an inverted version of the R/W signal from the processor used to drive the /OE signal on the RAMs, EPROM and FLASH parts. It is worth mentioning here due to the settling time and speed grade of ISPLSI1016E chosen. All of the original prototype and development work was done using 125MHz and 100MHz parts. If running the processor near 25MHz using fast RAM, the settling time of this signal should be analyzed if other than a fast part is used to save on cost.

The 'unused' pins on this part are brought out to headers J37 along with power and ground (J39 & J38). The functionality of these pins can be redefined by the user to implement logic, clocking, interface, etc... One could easily jumper a chip select from the vertical stacking bus to one of these 'spare logic' pins and use the existing data pins to interface this part to the processor bus. These pins could well server to implement the master / slave xor function required by MCPWM or any of a thousand other functions.



Some time after I'd begun shipping these boards it became evident that there was a problem when trying to reboot without the BDM cable connected. It turns out that when a real BDM cable is disconnected, the driving logic is removed. My driving logic is built into the board and thus never disconnected. As such I commandeered use of Pin #7 on J37 to indicate, when shorted to ground, use of BDM or when shorted to VCC, not. As booting from 16-bit mode has been ruled out by the necessity of an extra chip select, jumper J35 could be used for this purpose. It would appear that Jumper J35 can serve no usable purpose as an input due to the missing pull up resistor. In fact, the resistor is not missing at all. The resistor is internal to the ISPLSI1016E and is enabled via the programming equations.





Although not populated on the units shipped the LED and associated resistor light to indicate when the unit is in BDM mode and the processor is halted. This is great design feature as it was extremely helpful in debugging.

The equations in the file below implement all of the functions described in this section. If you need a refresher to understand what's going on in these equations please refer to the author's previous article on introductory CPLD devices. The user will have to excuse the authors hurriedly implemented state machine to generate the DSACK0 signal. Correct usage (syntactically) would be to build a counter that generates DSACK on overflow and doesn't count until certain conditions are met. If the user observes the compiler output they would recognize that this is exactly what they received in an easier to code method. Here again, the idea of "good enough" engineering comes into play.

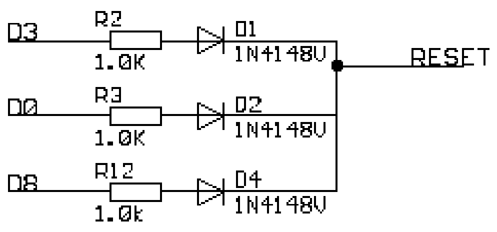
BDM CPLD Equations

Okay, so I've mentioned it before, but if the user is unfamiliar with the use of CPLD's and the equations in the above file, then please take the time to read the author's previous article on CPLD devices and their use on the CRC332 board. CPLD Encoder Article

A note for input. I've done all of my testing with a six-foot length of cable for BDM. Due to driving connections and noise, I do however recommend keeping the users cable run short 2&1/2-3 feet if working in an electronically noisy environment.

10:2

R2, R3 & R12 along with D1, D2 & D4 are deceivingly simple on Jeff's board. One might wonder why some data lines would be connected to the reset lines with diodes and resistors? In fact this connection happens on the CRC332 as well. On the CRC332 that function takes place inside the CPLD. So what is this connection? It turns out that Motorola made the 68332-microprocessor hardware configurable. This means that you can configure the hardware that the microprocessor uses to interface to the rest of the system.



How to you configure the system hardware while it's running? The answer is that at the hardware level you can't. But there is a time when everything is powered up and functional but nothing is running. You may have already guessed, but this time is while the processor is in it's /RESET state. While in reset the data bus pins are configured as inputs and during the last 12 clock cycles the processor stays in reset, it reads the values on the data bus pins to see if there is an input.

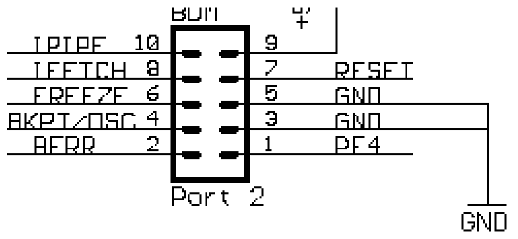
This is where the resistors and diodes on Jeff's board come into play. The processor data lines have weak pull up's associated with them and all it takes is a small diode to ground to over power the weak pull up. Since we only want to drive the data pins when the processor is in /RESET the diodes are connected to /RESET instead of ground (GND). The particulars of what each Data line does can be read in the Motorola Documentation below. In Jeff's case (and the CRC332 as well) the Processor is booted in 8 bit mode. Also, Port - F is instructed to come out of reset as discrete IO instead of with its interrupts in tack. This is important to keep an interrupt from occurring before the software is set up to handle it.

Again, Jeff has used six small inexpensive discrete components to implement quite a bit of functionality. This is contrasted to the CPLD #2 on the CRC332 discussed above although the CPLD affords many other features.

Jeff used the standard Motorola 10 pin BDM interface as discussed in the Motorola documentation below. Many 8 and 16 bit processor users may be thinking in the back of their minds, wait a minute, BDM is only 8 pins. That was true, but when interfacing to 32 bit processor, Motorola added two more pins for added functionality onto the connector. The original eight still have the same functionality.

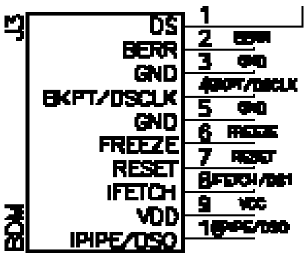
Use of the standard BDM implementation does require the use of

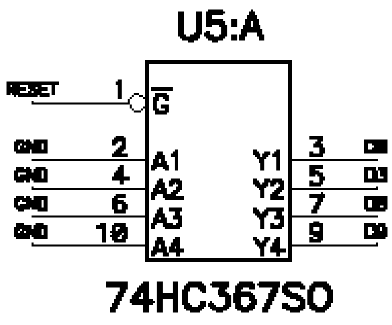
a special BDM cable / PCB assembly. A quick scan of the Internet will turn up a small handful of designs for use with the ten-pin interface. Below is a schematic snippet that shows the pin-out of this ten-pin connector as implemented on Jeff's board.



10:3

The Mini-Robo-Mind also uses the standard Motorola BDM implementation. This fits well with the design concepts of keeping the board small and only powering the absolutely necessary parts from the system battery. The extra PCB required to implement BDM sits at the PC end of a cable that plugs directly into a PC's parallel port. It conditions and drives the signals allowing for longer cable lengths than those used on the author's board.





Component U5 drives the processor data pins out of reset on the MRM332 using a different manner than either Jeff's board above, or the author's CRC332 board. Take a minute to review this section in the Motorola documentation then refer back to the MRM332 schematic, Jeff's schematic and the author's CRC332 schematic to see how each of these three solutions present a viable implementation of the reset driving solution. Note that even though interrupts or some other functionality is enabled or disabled through this hardware driving means as the processor comes out of /RESET, software can later re-enable/disable the feature when other conditions have been met. One such functionality that is specific to the MRM332 is configuring /CS10(also Address 23) to use as it's alternate ECLK functionality. This can be done via hardware coming out of reset or later via software. Since the signal does not have to be used immediately upon exiting reset this task could be equally handled by either means.

Mark took the time to modify the standard BD32 source code to support uploading source directly into flash. This extremely powerful tool facilitates board usage through BDM and relaxes the Battery Backed RAM requirement. This code is provided below as:

Mark's modified BDM_32

Note: for the above code to work correctly on the reader's 68332 based board, the chip selects for the FLASH part must be set up previous to using the FLOAD functionality.

The PCB / cable combination presented in section 10:4 (below) is compatible with the Mini-Robo-Mind as well.

10:4

Presented below are some links and designs for BDM interfaces and cables. The information presented here did not come from the 332SIG, but is presented here to offer a convenient all in one design resource. The product Charles Tidbury offers through the link below is well worth it.

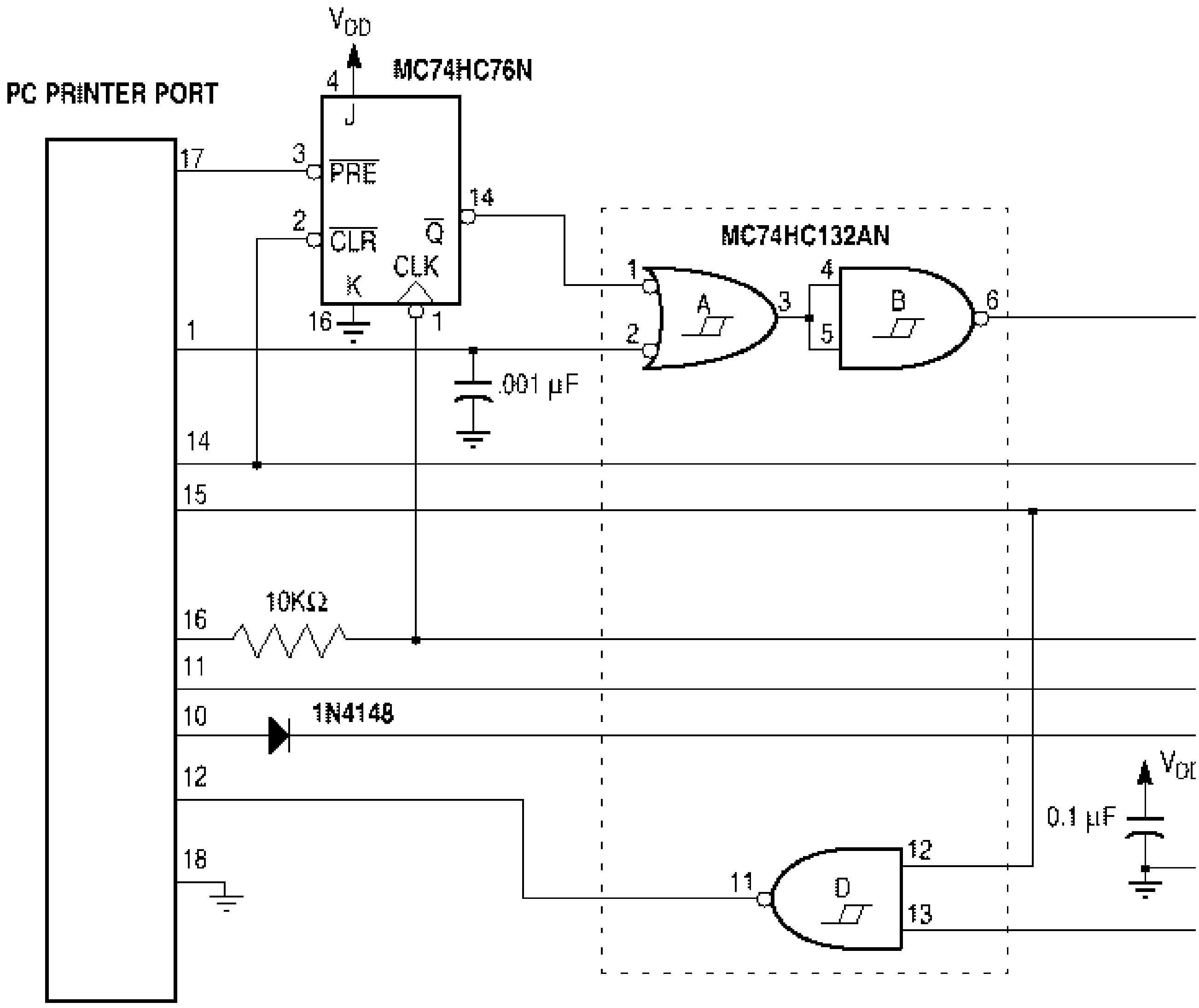
Introl BDM Schematic"

Introl Link

The PDF file below was used as the schematic basis for the hardware cable used to interface between the PC parallel port and the BDM cable connector on Jeff's board. It was located through a quick search of the web and is provided here for reference:

Gunder Magin's BDM interface schematic(74K)

Following is the Motorola suggested BDM interface from the Motorola specific application note: (below) This diagram was the basis for the version of the equations that shipped with the CRC332. So why are the equations not exact copies of the discrete functions presented here? Simply put, there are distinct problems using both /PRESET and /CLEAR on a flip-flop at the same time. In the world of programmable logic it is strictly forbidden. So one of the sources has to be written into the input equations for J or K and also worked into the /CLK equations. Once this transformation has been completed and the design has been modified for implementation with 'D' type flip-flops everything else is pretty straight foreword.

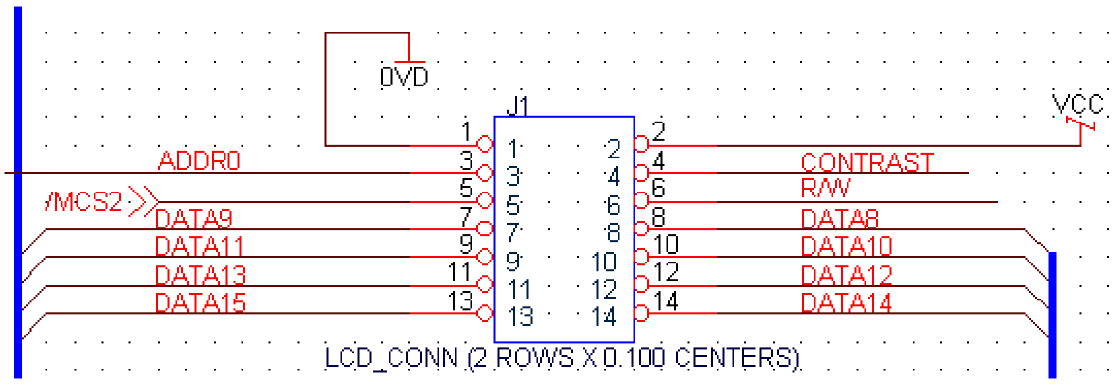


Section 11: LCD Cable Wiring.

11:1

Page 10 of the schematic simply shows the wiring of the LCD cable to interface to the "standard" two-line module. Note: When the reader builds their board, check the wiring of the particular module you have. The pin-out used on the CRC332 matched the particular module in use during the prototype development which it turns out is not widely available to the general public.

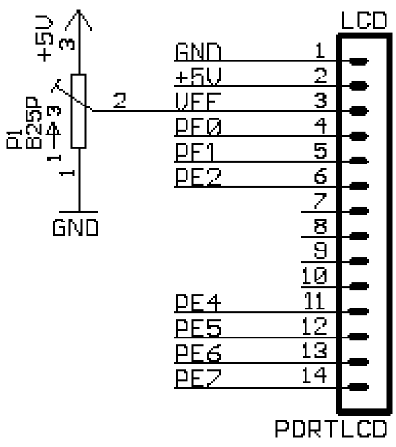
From the schematic side, the snipped below shows the connection used on the CRC332 end of the cable. This snipped comes from page one above:



The documentation form Sharp suggests using cable lengths fewer than six inches do to capacitive loading and cross coupled noise. This would then have to include the length of the copper traces on the PCB since this is connected to the processor's system bus. I've used (flawlessly) cable lengths in excess of 1 & 1/2 feet. In fact, the LCD and cable combination used to test all the production units shipped was over one foot in length. I'm definitely not recommending it, just passing this along for informational purposes.

If the user chooses to modify the board themselves, read the note under the CPU page about the suggested changes for the contrast (R1) power supply connections.

11:2



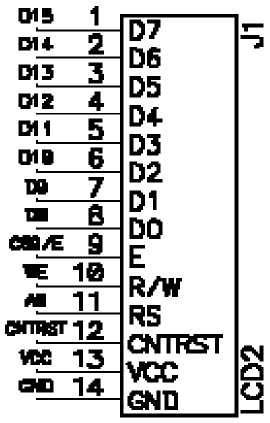
Jeff's board uses Port 'E' & 'F' to provide connection to a standard LCD device. Instead of having to tweak the clock, slow down the processor or generate special bus cycles and termination, as the CRC332 does. He's used the discrete Port - 'E' pins to software bit bang the LCD module in 4 bit access mode. A perfectly valid solution this removes the need for external circuitry and leaves it up to software to proved the data and clocking signals.

P1 is a pot for adjusting contrast on the LCD between VCC and VSS.

11:3

Discussed in the processor and LCD section above, the Mini-Robo-Mind uses the E-CLOCK signal to drive timing for the LCD. The cable wiring is one for one (straight through) from the fourteen header pins available.

Note: on the PCB this connection is a side by side header arrangement, two rows of seven pins on 0.100" centers.



Section 12: Artwork.

12:1

The file (right) contains artwork compatible with the GC-PREVIEW package. This artwork contains all the necessary information for the production of the CRC332 board by Alberta Printed Circuits (APCircuits), which includes TOP Copper artwork, Bottom Copper artwork & drill file information. With this file the user can have a pair of PCB's made for about ~$70.00 US Dollars.

CRC332 - Artwork_File(38K)

Since APCircuits does not offer solder masking and silk screening special care will need to be taken when stuffing the boards. First component orientation is critical, especially since the PLCC adapter sockets can be oriented 90, 180 or 270 deg out of alignment. Secondly, when soldering components to the board, care must be taken not to form any solder bridges between pads and close traces. The production version of the board was solder masked, which prevented this from happening. Since the parts are all through hole, (except the processor) one does not have to be a soldering expert to populate this board.

For instructions on soldering down the fine pitch surface mount processor refer to the authors previous article on oven soldering techniques. Oven Soldering Encoder Article This article has relevance to all three boards presented in this article. Yes, that's Mark and Jeff in the last photo. Thanks Guys!

If there is one thing to change with the artwork it would be to make the power traces from the power supply to the headers just a little bit heavier. At 0.020" in width, they can handle normal operation of most devices, but a large, stalled servo could draw enough current to damage the traces on the board. The simple answer is to widen the traces (a significant investment in time to change the layout). The easier solution would be to simply parallel a second 0.020" trace on the other side of the board. The author's design process is never finished as once a milestone is reached there are always points to be polished. Again, the author must suggest that we apply the concepts of "good enough" engineering and move on as time permits. I'll leave this feature change for the next incarnation of the board.

For those who want to make production quality boards, go ahead and contact me as I've got artwork for silk screen and solder mask layers that was used in the production of the CRC332's actually shipped. The full production mode of circuit board fabrication using solder mask and silk-screens is not recommended for the amateur roboticist. However, due to the high up front costs involved in making just a few of them.

12:2

The file (right) contains the board layout artwork produced by Jeff. It is worth taking time to review this artwork carefully as it is laid out completely by hand. Take special note of the parts placement and the clock circuitry.

These files also are Gerber formatted PCB artwork compatible with the processes used by AP Circuits. Refer to section 12:4 (below) for more information on viewing and working with these files.

Jeff's Artwork(XXK)

12:3

Currently the artwork file for the MRM332 is not available as the board itself is still in full production.

At the current board pricing, Mark can sell you a full board assembled and tested for just a few dollars more than it would cost to have a fab house make a small quantity of PCBs.

Mark's got a good product and it is well worth the cost. The link at the end of this article will take you to his web site for ordering information.

12:4

Alberta Printed Circuits company can be reached via the following web address. The file presented above can be e-mailed directly to them for production along with a filled out order form that is available from their web site below.

Alberta Printed Circuits

For the reader to get an idea of the product to expect from APCircuits, refer to Section 18:1.1 - 1.3. These three-proto type PCB's were fabricated at APCircuits.

AP Circuits also provides the GC-Preview program so that you may open up and view the artwork file before sending it. To download that software click on the link provided (below). Note: this software is DOS based and runs as a protected mode application. It is not too... user friendly.

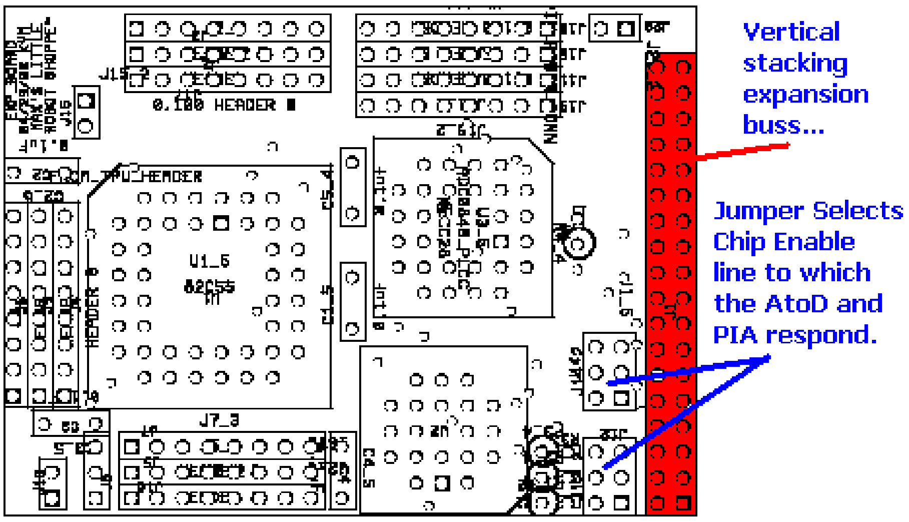
APC's GCPreview Program

Clear at the bottom of this article are three pictures that show off the layout of the boards developed by the 332SIG. These are presented to give the reader some perspective on board layout options other than those used on the CRC332.

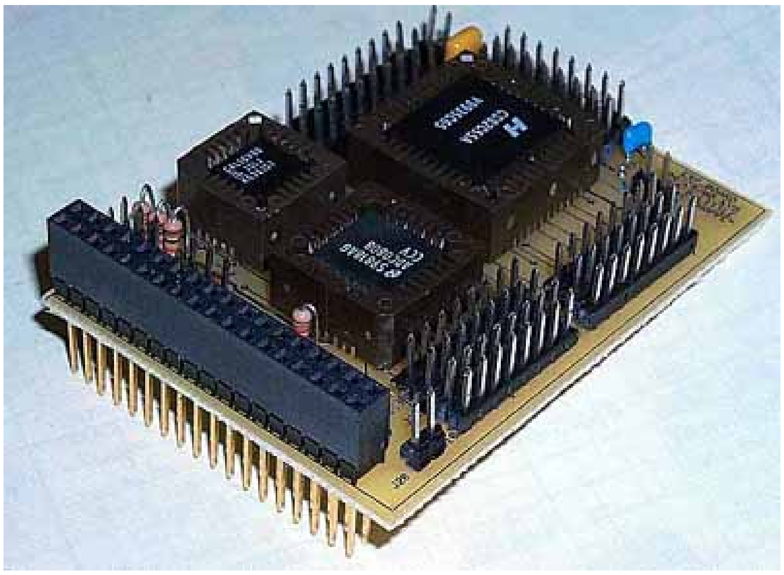
Section 13: IO Expansion Board.

13:1

Expansion_Schematic_1	Expansion_Schematic_2	Expansion_Schematic_3


Vertical stacking expansion buss...
Jumper Selects Chip Enable line to which the AtoD and PIA respond.
U1_6
82C55
0.100 HEADER
J7_3

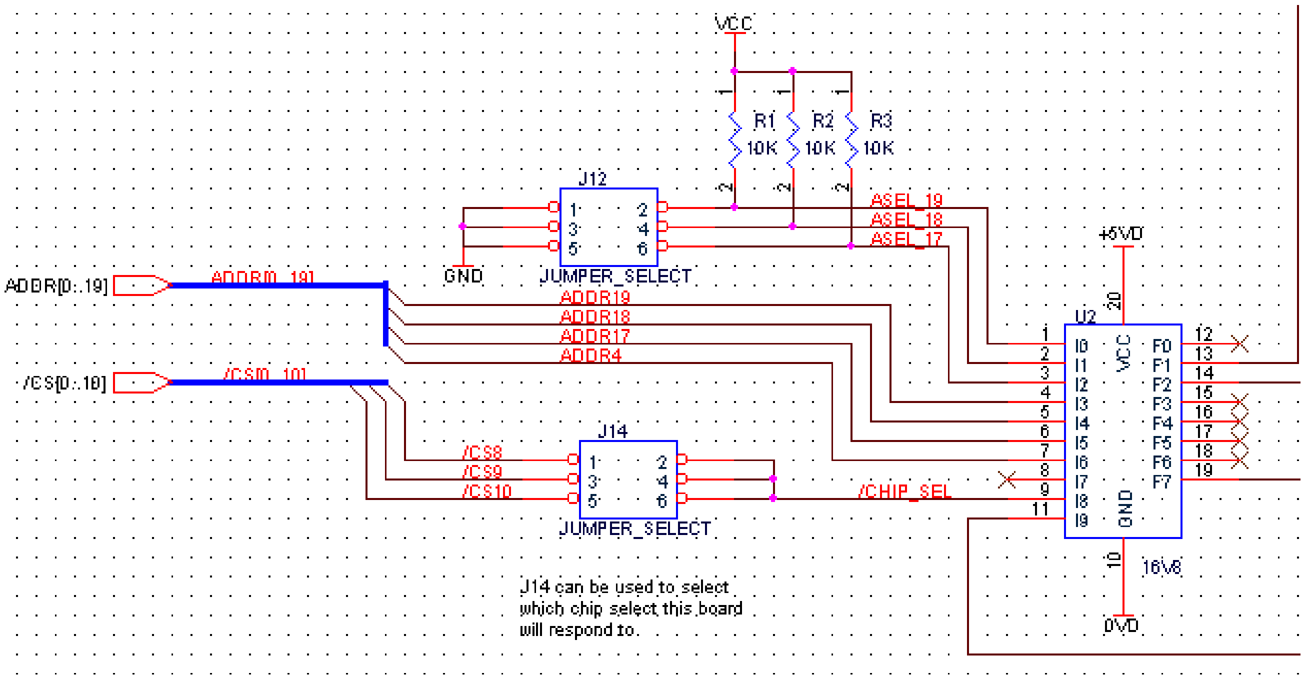
The IO expansion board schematics are provided (above) along with the simple equations for the PLD source code (below). The IO expansion board provides an additional 24 digital software configurable IO lines and an additional 8 channels of analog to digital conversion. The analog to digital conversion is provided via an ADC0848 just like the one on the CRC332 motherboard. The 24 additional channels of digital IO are facilitated via an 82C55- peripheral -interface-adapter. A photo of this board follows:



The PLD on this board, a 16V8 in a PLCC package for size, along with the associated jumpe,rs allow the user to configure which chip select the IO expansion board will respond to as well as which memory block within that range of the chip select the board will respond to. This allows for more than one expansion board to share a chip select (up to 8 with this configuration scheme). The only limitation here will be the current sourcing ability of the 68332 to drive that much parallel input and the speed at which this can be accomplished as the capacitance grows with added trace length and pin count.

The first of these three pages is the connection to the vertical stacking expansion bus. This page requires no additional comments, as it is identical to the mating vertical stacking connector on the CRC332 motherboard.

The second page contains the 82C55 PIA (peripherals interface adapter) and the associated header pins. Here, just like on the CRC332, the IO pins are arranged with power and ground connections to facilitate interface to other digital devices. Also, like the CRC332 connections, a jumper is provide to the power connection to allow an alternate power source to be used with the eight IO connections from port 'B'. The lower left of the page depicts the 16V8 PLD used to select which chip select the PIA and ADC will respond to. Additionally, jumpers determine which memory address range the devices will respond to within the range provided by the chip select. These chip select and address select jumpers are visible in the schematic snippet (below):



If you're unsure how to interpret the equation file (below) refer to the author's previous article on CPLD programming for an introduction.

CPLD equations for IO devices

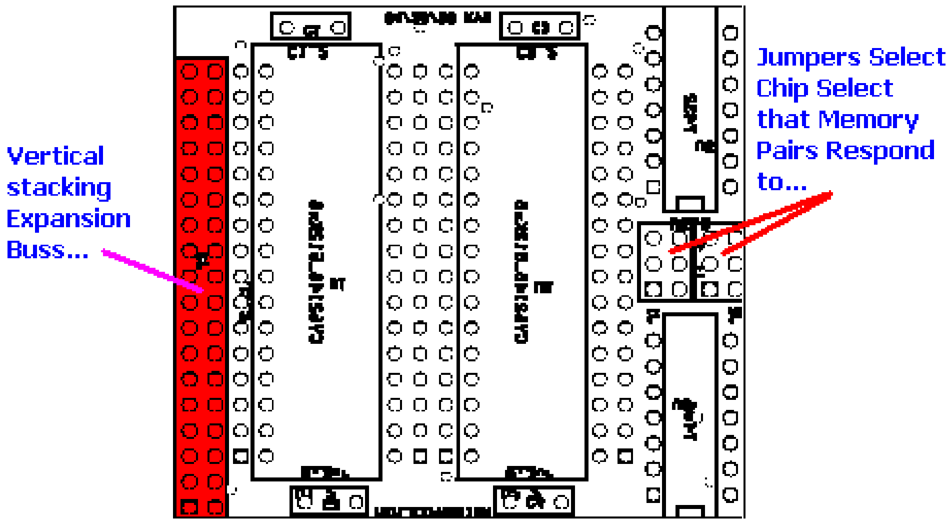
```
MODULE IOEXP4

TITLE 'IOEXP4'

ASEL_19 PIN 1;
ASEL_18 PIN 2;
ASEL_17 PIN 3;
ADDR19 PIN 4;
ADDR18 PIN 5;
ADDR17 PIN 6;
ADDR4 PIN 7;
L_CHIP_SEL PIN 9;
L_PIA_EN PIN 13 ISTYPE 'COM';
L_ADC_EN2 PIN 14 ISTYPE 'COM';
DA_NODE PIN 18 ISTYPE 'COM,KEEP';

EQUATIONS
DA_NODE = (ADDR19 == ASEL_19) & (ADDR18 == ASEL_18) & (ADDR17 == ASEL_17);
L_PIA_EN = !(!L_CHIP_SEL & DA_NODE & !ADDR4);
L_ADC_EN2 = !(!L_CHIP_SEL & DA_NODE & ADDR4);

END
```

The third and final page shows the implementation of the analog to digital converter used on the expansion board. The two-pin jumper can be used to bring VEE from the CRC332 power out connector to the header pins on this board for external op-amps etc as discussed in the power supply and AtoD sections above.

Section 14: The Memory Expansion Board.

14:1

Memory_Schematic_1	Memory_Schematic_2


Vertical stacking Expansion Buss...
Jumpers Select Chip Select that Memory Pairs Respond to...

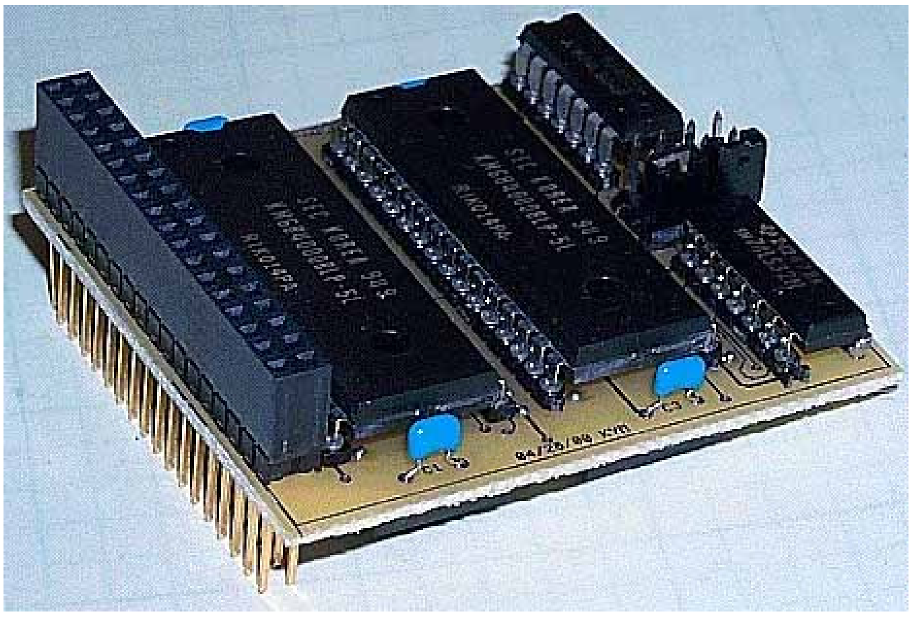
The schematic pages above delineate the design of the memory expansion board. The board shipped in three configurations. 1) 512K bytes, 2) 1 Meg-byte 3) 2 Meg-Bytes. Much like the IO expansion board above, jumpers select the Chip select signal to be used with these boards.

Page one of the memory expansion schematic contains the guts of the design, while page two should look familiar as it contains the vertical stacking expansion bus presented on the CRC332 board and the IO expansion board. If you are familiar with the author's previous work you may be surprised to see two discrete IC's populating this board; a 74HCT32 and a 74HCT04, rather than a GAL16V8. The decision to use the discreets rather than programmable logic was two fold. First, there just happened to be two rails full of these IC's left over from previous projects on the author's workbench. Secondly, the PLD, even at $1.78, costs more, and by the time one adds a socket that costs just more than the part at $1.80, the decision to use discreets was justified.

Again, to keep the routing simple just like on the CRC332 motherboard, through hole components were used. In addition components are mounted to both the top and bottom sides of the board. This is facilitated through the use of sip sockets soldered in a staggered fashion to allow insertion and removal after the solder work. (2Meg = 4 Memory IC's)

The two discrete IC's on the board allow the chip selects of the different IC's to be multiplexed based on the status of the A19 address line. The memory here is only accessed in 8 bit wide mode. Memory access of this type is best suited for application data that does not require instantaneous access at all times. I.E. Large volumes of mapping data. Mapping data needs small portions to be accessed real time with smart algorithms to determine what needs to be accessed and what does not. This then justifies the 8-bit access.

The two sets of jumpers on this board hint at one of the author's only pet peeves with the 68332 processor. In configuring the SIM Motorola allows a chip select to respond to an address range of only one-megabyte. This dictates the use of two chip selects (a valuable commodity) for this memory expansion board, hence the overlay capabilities (chip select sharing) of the IO expansion board above. This problem becomes extremely evident when trying to configure 16-bit wide memory. If the chip selects could be configured to span the entire address range then generating a /WRL & /WRH for 16 bit wide accesses would be easy. Additionally, in some of the authors other 68332 work, enabling 4 megabyte flash chips takes additional logic external to the processor and eats valuable PCB real-estate.



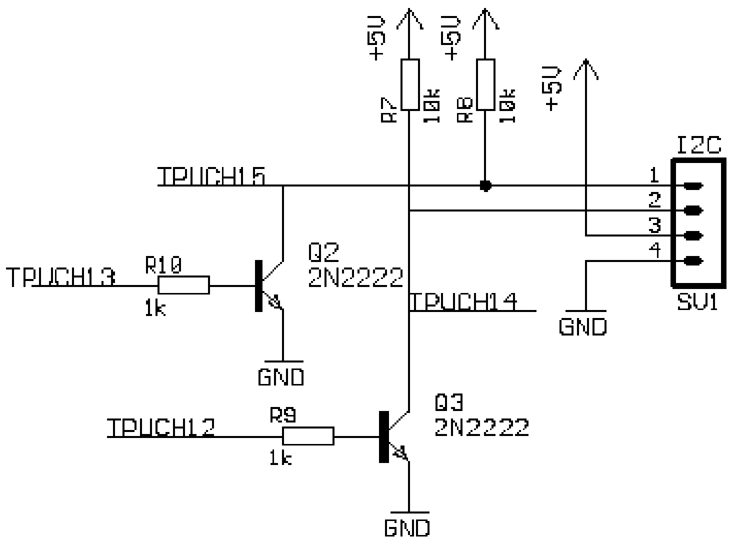
The RAM on the Memory expansion board is not battery backed. The principle behind usage of the CRC332 that allows for this functionality is the idea that the actual instruction code that may require battery backup would be stored in main memory on the CRC332 motherboard itself where battery backup is available. The volatile application data however will be stored on this expansion card and will be lost every time power is cycled. Two other ideas that drove this decision were the added cost of the memory expansion board and physical board real-estate requirements to add the battery. The only room currently available on the memory expansion board is on the bottom side under the discreets leaving the battery overhanging the edge of the PCB. Additionally, the author considered that the battery would hang down from this memory expansion board right into memory select jumpers and would have to be electronically isolated.

If one were monetarily challenged to the point of not being able to afford the extra memory expansion PCB, then a good hack would be to solder the expansion RAM chips directly on top of the existing RAM on the CRC332 leaving only the /CE pin bent outward. To this small wire jumpers could connect the requisite chip select circuitry. Instead of attaching the discreets with glue simply use the available Logic pins offered by CPLD #2 (above).

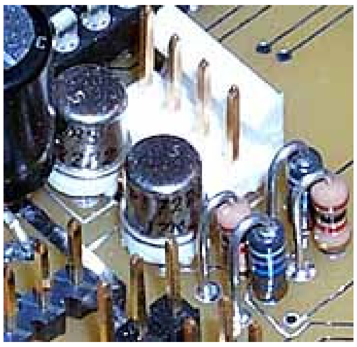
Section 15: I2C.

15:1

Built into the 68332 processor are four SPI ports (Ref the SPI section above). Many devices use Motorola's SPI bus to interface and communicate.



Back in 1985 Phillips introduced the I2C bus for interfacing multiple devices to a simple bi-directional serial bus with arbitration and synchronization. The 68332 processor itself does not directly implement I2C. The authors implementation used a pair of Port - E pins and an extremely large amount of software to 'bit bang' this protocol.



Jeff found and implemented a better solution. The file below was written by ***A. Morbach*** on 24/6/98. It uses four TPU pins and a few discrete components and a custom uploaded TPU function to implement I2C on the 68332 processor. The image (right) depicts Jeff's implementation of the hardware required to drive I2C from the processor.

To use this file you will need the Motorola TPU library compiler. It is available through the Motorola documentation link in section 17:1 (below). Once compiled the reader must enable the chip select for the internal 2k of RAM inside the processor and then use BDM32 to upload their newly compiled library to this RAM. A control register must also be modified to point to this internal RAM. This is one place that battery backup of the internal RAM would come in handy if it did not defeat the counter points discussed above under Section 2:1,2 & 3. Alternatively, these functions could be done by software during boot if BDM is unavailable.

I2C TPU Function Document (43K)

I2C Spec from Philips(280K)

If one function for the TPU is to be uploaded into the internal RAM in the processor, then all of the functions to be used with the TPU must be loaded into this internal RAM. I.E. the configuration register can only point the TPU to one place to retrieve its microcode, internal RAM or internal ROM masks. For this reason, Motorola makes the source for all of their TPU microcode available for the user to download and recompile. An added benefit to the user, this allows one to mix and match functions from both 'A' & 'G' processor mask sets at will. A good example of this was Mark's implementation of standard PWM on a 'G' masked CPU.

The full I2C specification is available from Philips through the link above: As of the writing of this article, the most recent revision was in January of 2000.

Section 16: Using the Board, Software & Misc.

16:1

The key concepts in using any 68332 board out there is to digest the massive number of internal configuration registers. If figuring out a maze of jumpers is beyond the reader, then possibly the CRC332 or any 68332 board is not the platform level for you. However, if you enjoy jumping into complex configuration and digging through masses of documentation to end up with a hardware platform that is more powerful and capable of outperforming most others, then this is definitely the solution for you.

In the initial phases of working with the 68332 processor, learning to configure the registers in the SIM module should be the first task. The SIM controls (among other things) the chip select lines of the processor. The chip selects can be programmed to respond to different byte configurations (even - odd / 8 - 16 / read - write / interrupts) as well as auto-generation of bus termination of bus cycles. (DSACK's) The registers to look for here are the Chip Select Options Register (CSOR_X) the Chip Select Base Address Register (CSBAR_X) and the Chip Select Pin Assignment Register (CSPAR_X). Remember that using DSACK's requires modifying the Port E Data Direction Register and others. This of course can not be a one-stop shop for setup information. Motorola's SIM manual when printed is over 1/2" thick and, unfortunately, should be read cover to cover.

Often users of the CRC332 have asked, "Where is the memory map?", astounded that I did not furnish one with the board. Many piles of documentation later the end user will recognize that the memory map is completely dynamic and can be changed at will via software, a concept hard to accept at first.

Zipped Software File

The original software ship set is included (below). This was intended to be loaded into RAM via BDM and executed after launching a macro through BDM that sets up the first chip selects to allow the loading of RAM. (Odd, but that reads correctly) Anyway, the software initializes several other chip selects, start up a channel of MCPWM reads an encoder and prints it's current value to the LCD screen, continuously dumps all 8 AtoD channels to the screen as well as the PIA input port #1.

One advantage of using the TPU overlay structures the way I've set them up in the example software file (above) is the use of pointers to structures. GCC loves them and because of the addressing modes of the CPU32 core within the 68332 the output assembly code is pretty tight.

● 16:2 ●

Another Tricky issue is booting in 8 bit wide mode (I.E. From EPROM or FLASH) and switching to 16-bit wide memory to overcome the 40% performance hit described in the Motorola documentation. Jeff to the rescue! One of the other members of the 332 SIG developed some assembly language code to do just that. (Below) This code allows the processor to boot from address 0x000000 and jumps to the first boot loader. The first boot loader enables the RAM chip selects for the RAM inside the processor. It then copies the second boot loader from EPROM (or FLASH) to the RAM inside the processor and continues operation there. This second boot loader sets up the chip selects for the remaining RAM in the system at a higher address, say 1 Meg. (0x00100000) It then copies the rest of the program information from the EPROM (or FLASH) to this RAM at the 1 Meg address. Finally it disables the chip select for the boot device (/CSBOOT) either EPROM or FLASH, and changes the chip selects for the RAM to respond to memory location 0x00000000. Execution then resumes at the entry point just above the exception-processing table. Yup, a pretty tall order and he wrote it in assembly for maximum efficiency. This code is provided (below). Note the step where code is copied from the EPROM or FLASH to the external RAM can be skipped if using the battery backed RAM feature.

The collection of software that shipped with the CRC332 grew substantially with each passing week. In the last orders filled the CD's contained just over 200 Meg of compilers, assemblers, operating systems, documentation and design examples. Currently I have no way to make all of this available in one condensed location. (Imagine downloading 200 Meg?) Possibly, the most powerful of the software ship-set was the GNU C compiler setup to run in DOS and target the 68332. Just a few minutes to set up a couple of environment variables and away one goes. An exhaustive search of the net will locate most of the public domain software originally shipped. Mark C provides the GNU-C compiler with the Mini-Robo-Mind (MRM332) board that as of the writing of this article is still in production.

Jeff's Zipped Boot Code

● 16:3 ●

Finally, if the reader is looking for a programming language to get you started, how about basic? A language that almost everybody knows, Karl Lunt has adapted his S-BASIC to produce assembly language code compatible with Motorola's free assembler. What could be better? Karl provide his S-BASIC compiler for free as well:

Karl Lunt's Home Page

The assembler required to go along with the S-Basic package is provided as freeware from Motorola through the link below and is also included in the SBASIC software download from Karl's page (above).

● Section 17: General - Documentation.

17:1

From the selections below the reader can download the basic Motorola documentation. These documents are in PDF format and some of them are extremely large. I.E. the reader may want to start the download and go do something else unless they have an extremely fast connection.

Basic Motorola 68332 Manuals:

332_Tutorial(346K) 68332_Users_Manual(1,508K) CPU32_Manual(1,468K) SIMRM_Manual(1,169K) TPURM_Manual(769K) GPTRM_Manual(448K) QSMRM_Manual(1,299K)

Should any of the document links above have broken since the writing of this article, the link below will take you to the site that Motorola has been spending the most time updating as of late. Many of these documents can be found there.

Motorola Document Site

Section 18: Parting Words.

18:1

After announcing the discontinuance of the CRC332 I was deluged with e-mails conveying the same basic message. Most of them started out, I've been waiting just that additional month to save $$$... When will they be available again... Where... Why....

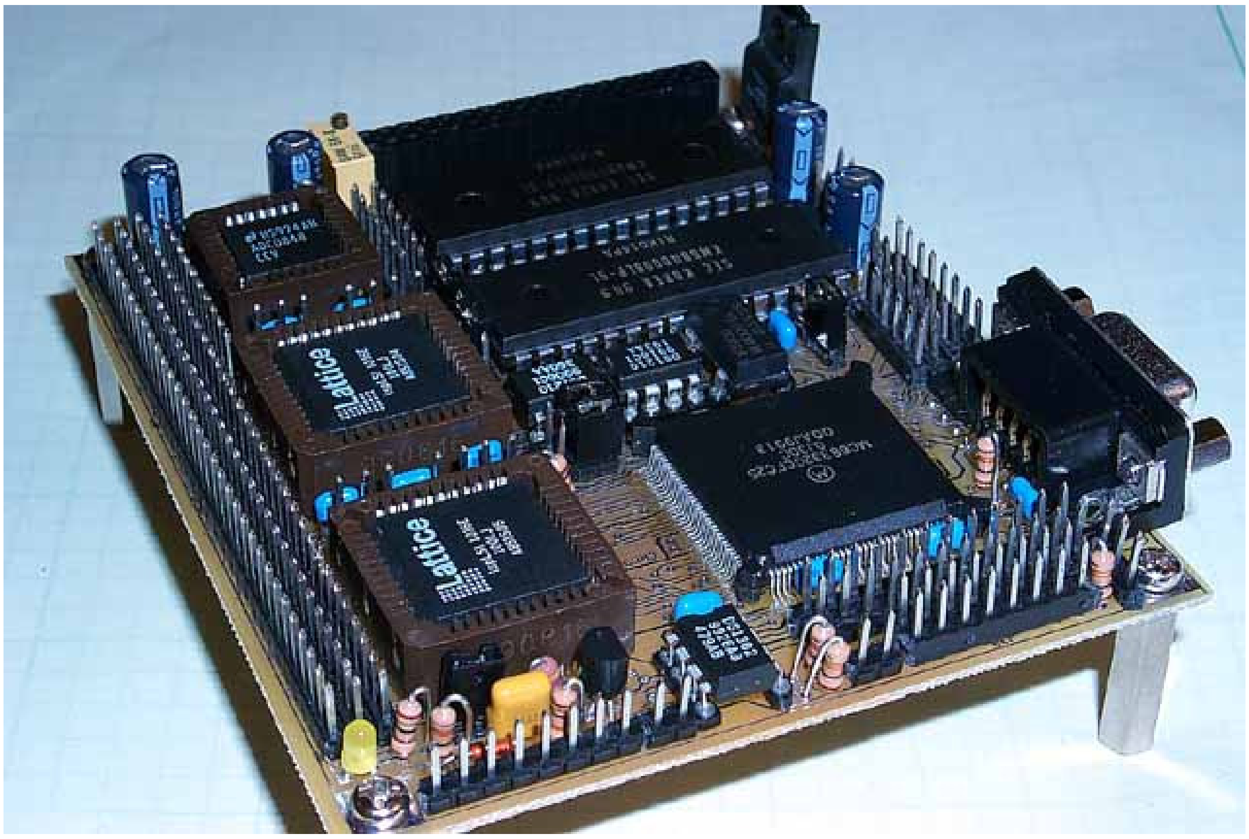
It is the authors sincere hope that this article will help provide the information required for anyone wanting to build their own CRC332 board. Additionally, this offers the ability of the end user to modify the design to include the users own functionality. If or more likely when this happens I hope that that user whoever and whenever will take the time to share with the rest of us what they've accomplished.

Mark still offers the Mini-Robo-Mind through the link at the end of this article. If this is your first attempt at working with a CPU32 core processor, it is the authors sincere belief that the reader should purchase a MRM332 and become familiar with its design and functionality before attempting to design / layout their own.

Following are photos of the three boards developed by the members of the 332SIG. These photos represent the culmination of over a year's worth of work. Each board has a slightly different set of features, different driving IC's and completely separate CAD systems and layout practices. This is a good showing of people utilizing the same basic technological platform to arrive at different design solutions.

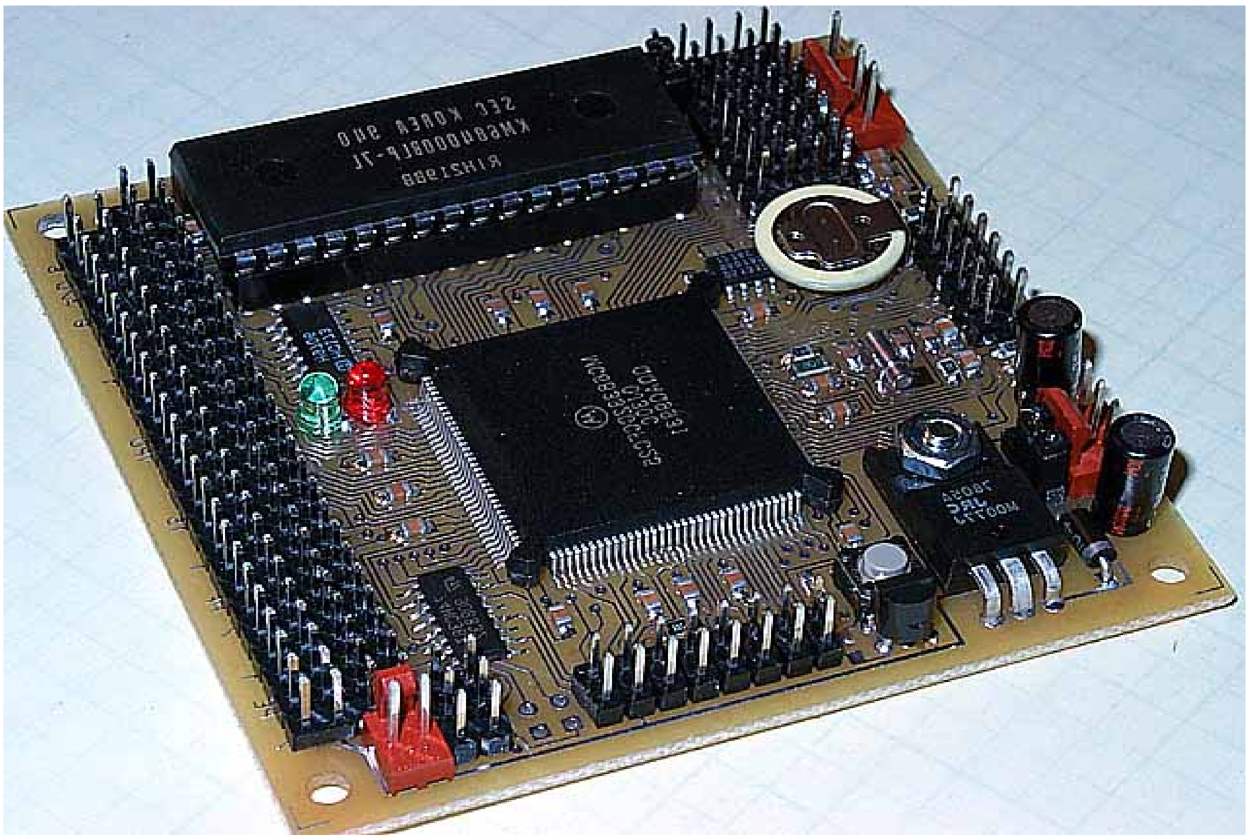
18:1.1

The Author's CRC332 Board



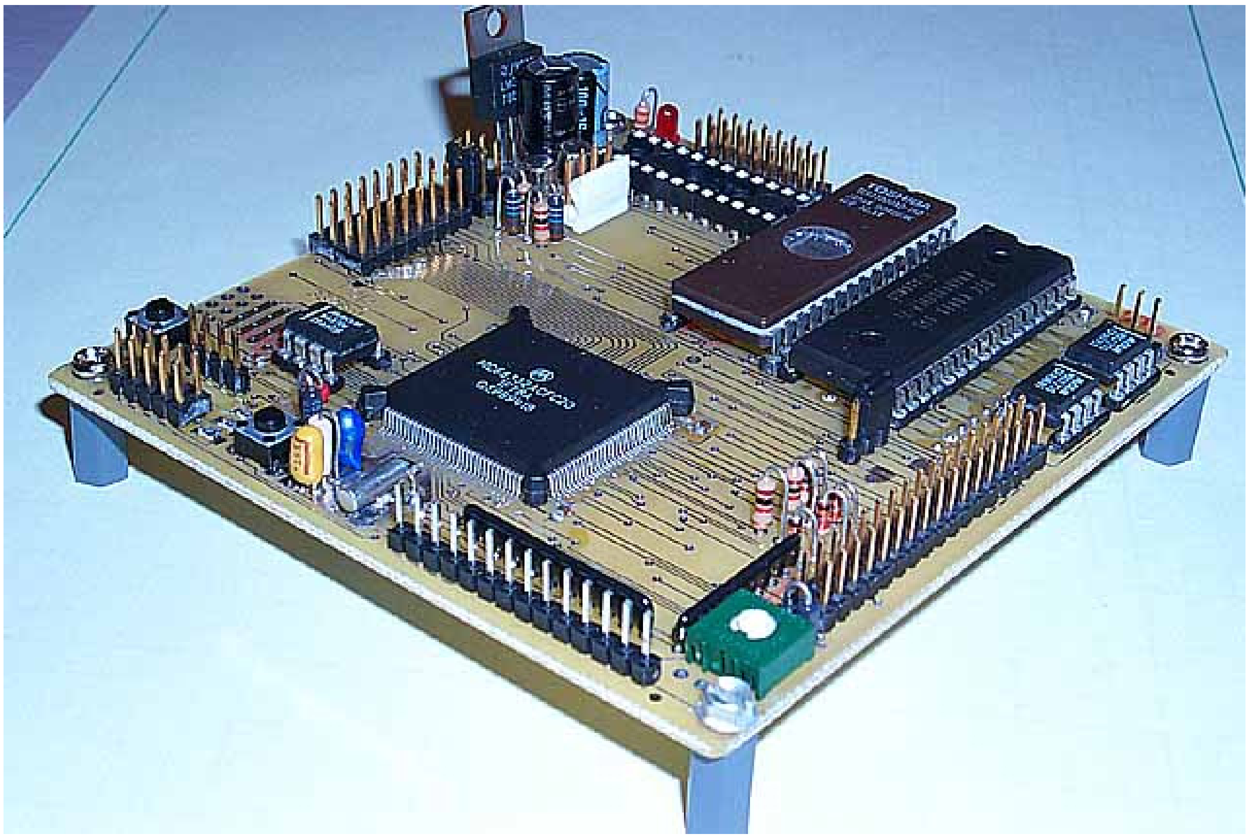
18:1.2

Mark's MRM332 Board



18:1.3

Jeff's Board



The design philosophies used on the boards above differed quite a bit. Mark went after the ultra small foot prints and added high speed AtoD and mixed surface mount parts (like FLASH and AtoD) underneath the through hole. Jeff went on to implement I2C drivers to work with the software loadable TPU I2C drivers. The author attacked this design from the philosophy of stuffing as much functionality as possible into a reasonable space for robotics. All in all very exciting differences that I hope will inspire new variations from the readers of this article.

Special thanks to Jeff for spending the time to develop the boot code that so many of us are now using. Additionally, extended thanks to Jeff for sharing his excellent design with all of the rest of us amateur roboticists.

Additionally, thanks to Mark for development of the FLASH-loading version of the BDM32 software. This also contributes greatly to the tools available to the Amateur roboticist.

If the reader is interested in buying a MRM332, Mark has them available through the following link:

Mark's 332 Page

For those of you curious why I needed to stop spending all my time building CRC332 boards and get back out into my shop, follow the link below to see some of my personal robotics stuff. There are a few new pictures there, and a whole new section on the latest injection mold machine and a second new section on controllers.

Kenneth's Home Page

kmaxon@uswest.net

So, how could one possibly follow up an embedded design like this? Coldfire anyone?....

Drinkin' Diet Coke & Gettin' it done... Kenneth