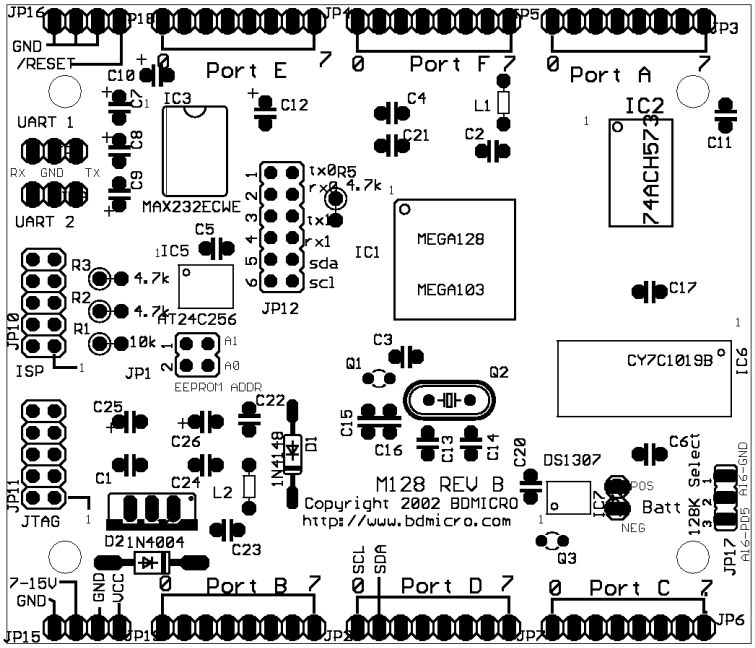


M128 REV B

Technical Manual

BDMICRO
http://www.bdmicro.com/

December 29, 2003

DISCLAIMER

BDMICRO products are designed and assembled with care and all assembled products must pass functional testing to ensure quality before they are shipped. However, our products are not rated for use in medical, life support, or systems where failure could result in the loss of life or have serious life threatening or economic impact.

BDMICRO does not warrant merchantability for any purpose and shall not be liable for any direct, indirect, incidental, special, exemplary, or consequential damages of any kind, however incurred, through the use of our products.

Contents

1 Introduction

This document describes features of the BDMICRO M128 REV B board. The M128 is an ATmega128 based microcontroller board. The main features of the board are:

- Atmel ATmega128 microcontroller - this MCU is packed with features. Some of the highlights are:
  - 128K Flash memory (program space)
  - 4K internal static RAM
  - 4K EEPROM
  - optional 64K external static RAM
  - 8 channel analog to digital converter
  - dual UARTs
  - I^2C interface
  - up to 48 digital I/O pins
  - JTAG interface for programming and debugging
  - 2 8-bit timers, 1 16-bit timer
  - 6 PWM channels
  - Watchdog timer

  These are just the highlights of the chip. To find out more about the ATmega128, visit Atmel's web site (http://www.atmel.com/) where they have an excellent datasheet describing the chip in detail.
- Up to 128K Static RAM [upper 64K is bank selectable using PORTD5]
- Real Time Clock crystal connections are routed for TOSC1 and TOSC2 pins on the ATmega128. This allows you to connect a 32.768 KHz watch crystal at location Q1 which will clock the ATmega128's TIMER0 asynchronously from the main system clock, and keep real time, even in power-down modes.
- I^2C based RTC (Real Time Clock) using the Dallas 1307 with battery backup connector
- 24CXXX I^2C serial EEPROM
- 2 RS-232 ports (3 pin headers)
- On board voltage regulator for flexibile power supply input
- Standard 10-pin JTAG debugging and programming header
- Standard 10-pin ISP programming header
- All ports (A, B, C, D, E, and F) in addition to power, ground, and reset are brought out to the board edges like a giant DIP so that it can be bread-boarded and used with prototyping board such as Vector board. Additionally, the pin headers provide spacing for organization as well as to provide the ability to use 0.1 inch screw terminals for connections.
- Generously labeled for ease of use and parts placement
- Footprint for 32KHz crystal and capacitors to connect to TOSC1,2 provided
- Compact board size: 2.95 x 3.45 inches
- 4 mounting holes at the corners
- Unused features can be left off the board if desired. The only part absolutely required for a working board is the processor itself. Even a crystal is not required because the ATmega128 incorporates a 1 MHz internal oscillator.

2 Soldering the M128 REV B Board

If you purchased a bare board or a kit with a bare board, you will be soldering the components on the board. For the surface mount components, and especially for the ATmega128 itself, I highly recommend you get some 0.015 inch solder to do the soldering. A short tutorial with some photos is available on the web site at `http://www.bdmicro.com/smt/`. The main points to remember are:

- Use your solder sparingly, it takes only a tiny amount to make a good connection.
- Don't use your soldering iron directly to melt the solder. Doing so causes the solder to bead up due to surface tension on the iron and then release in a big drop across the pins. This results in way too much solder and causes bridging, shorting adjacent pins together.

  Instead, use your iron to heat the pin, which then melts the pad underneath the pin. At this point, touch your 0.015 inch solder to the base of the pin. It should melt very quickly, consuming only one or two millimeters of solder, and flow onto the pin and around the pad. Remove the solder quickly.
- Make sure your iron is hot — about 630 degrees F seems to be about right. It needs to be hot enough to melt the solder indirectly, but no so hot that it damages the part.
- While not absolutely required, a good temperature-controlled soldering iron can make the process go much easier than with the standard soldering pencil type irons. If you plan on soldering smaller parts much, it would be well worth your while to aquire one. Good deals on these can usually be had on E-Bay.

The order in which you solder the parts can make the soldering easier. You don't want to install the tall parts first, because they would then tend to block access to shorter parts. I solder the surface mount parts first in the following order:

- ATmega128 (IC1)
- MAX232 (IC3)
- AT24C256 (IC5)
- DS1307 (IC7)
- CY7C1019B (IC6)
- 74AHC573 (IC2)

Then begin soldering the through-hole parts in the following order:

- bypass-capacitors (C1, C2, C3, C4, C5, C6, C11, C17, C20, C21, C22, C23, C24)
- 1 uF capacitors around the MAX232 (C7, C8, C9, C10, C12)
- the two diodes (D1 and D2)
- resistors (R1, R2, R3) [there is no R4]

  NOTE 1: ***DO NOT INSTALL R5*** — install a shorting jumper or short piece of wire instead (see note below).

  NOTE 2: while the silkcreen indicates that R2 and R3 (I^2C bus pull-up resistors) should be 4.7K Ohms, it was later found that values of 1.8K Ohms allow for higher I^2C bus clock rates and/or longer I^2C bus lengths.
- the jumper blocks and pin headers (JP1, JP10, JP11, JP12, Batt, UART1, UART2, JP17)
- port headers around the edge of the board (ports A-F + power, ground, and /RESET headers)

- two inductors (L1, L2)
- the 47 uF power supply filter capacitors (C25, C26)
- the crystals (Q1, Q2, Q3)
- finally the LM7805

Note: capacitors C15 and C16 are not required and should not be installed.

Note: R5 should not be installed. The intent of R5 was to allow the MAX232 to be active at the same time as the ISP programmer is programming the board. However, it was later found that R5 inhibits serial reception on UART0. Shorting (bypassing) R5 restores serial reception. If, however, you find that you have trouble programming the M128 when R5 is shorted, just remove the RX0 jumper on J12 (second from the top) while programming.

3 Fuse Bit Settings

Atmel AVR processors incorporate *fuse bits* which control various functions of the chip and persist even across a chip erase. By default, Atmel ships the ATmega128 with several fuse bits already programmed by default. Notably, the M103C fuse bit (ATmega103 compatibility mode) is enabled, as well as JTAGEN which enables the JTAG debugging lines of PORT F. Also, by default, the internal clock source is selected to run the processor at 1 MHz.

If you ordered a Kit, your ATmega128 processor's fuse bits are all at their default values and will need to be changed to run on the M128 REV B board at 16 MHz. For 16 MHz operation, you will need to program the CKOPT fuse bit (set to '0'). The clock selection lines will need to be modified as well: unprogram CKSEL3, CKSEL2, CKSEL1, and CKSEL0 (set to '1'). Also recommended when using the external crystal is to set SUT1 and SUT0 to '1' (unprogrammed). This setting is for slow rising power which may be necessary depending on your power supply. Additionally, you will need to unprogram the M103C fuse bit (set to '1') if you are compiling your code to run on an ATmega128 (as opposed to an ATmega103).

If you orded an assembled and tested board, these fuse bits are already set appropriately to run at 16 MHz, and the M103C fuse bit has been unprogrammed.

One other note with regard to fuse bits — the JTAG interface uses the upper nibble of PORT F, and the JTAGEN fuse bit is programmed by default. Since the JTAG interface supercedes all other functions of these PORT F lines, their other functions are not available while the JTAGEN fuse bit is programmed. Thus, if you are not using a JTAG programmer/debugger, you may wish to unprogram the JTAGEN fuse bit.

For a description of all the fuse bits, see pages 286 and 287 of the ATmega128 data sheet available from Atmel's web site (`http://www.atmel.com/`).

4 Initial Jumper Settings

For the purposes of verifying operation of your board, you should install all jumpers on JP12 (the 2x6 jumper block) and install the `A16-GND` jumper on JP17. These jumper settings enable all features and tie the A16 address line of the 128K RAM to ground.

5 I^2C Addresses

Up to 128 devices can coexist on the I^2C bus. The M128 REV B board incorporates 2 such devices, the Dallas DS1307 Real Time Clock and the Atmel AT24C256 Serial EEPROM. The DS1307 resides at a fixed I^2C address that cannot be changed, while the AT24C256 can reside at one of four possible addresses, depending on the jumper settings of JP1. The device addresses are:

- DS1307 ..... 0x68
- AT24C256
  JP1 - empty 0x50
  JP1-2 installed 0x51
  JP1-1 installed 0x52
  JP1-1 & 2 installed 0x53

6 Utilizing the full 128K of RAM

The ATmega128 can directly access up to 64K of external memory. However, the M128 REV B incorporates a 128K RAM chip. How is this accessible? On the M128 REV B, the A16 address line of the RAM chip is optionally tied to PORTD5 of the ATmega128 via the JP17 jumper. When this jumper is connected to PORTD5, then the A16 address line follows the state of PORTD5, and thus the extra 64K can be accessed under program control.

Only one 64K bank of RAM can be seen by the CPU at a time. Thus you need to be careful when switching to the other memory bank so that variables and data that are being referenced in the other bank aren't accidentally referenced while that memory bank is switched out.

7 Special Considerations for PORT A and PORT C

The design of the ATmega128 MCU precludes the use of ports A and C while at the same time utilizing external memory. Therefore, if your RAM chip is installed, you should avoid using ports A and C for general purpose I/O. Using these ports for I/O while the RAM chip is installed can damage either the RAM chip, the ATmega128, or both.

If you purchased a fully assembled board and you find that you will not be utilizing the external RAM that the RAM chip provides, but you need to utilize PORT A and/or PORT C, I recommend that you desolder the RAM chip and latch. This way, there's no way you can cause damage to the ATmega128 due to a programming error.

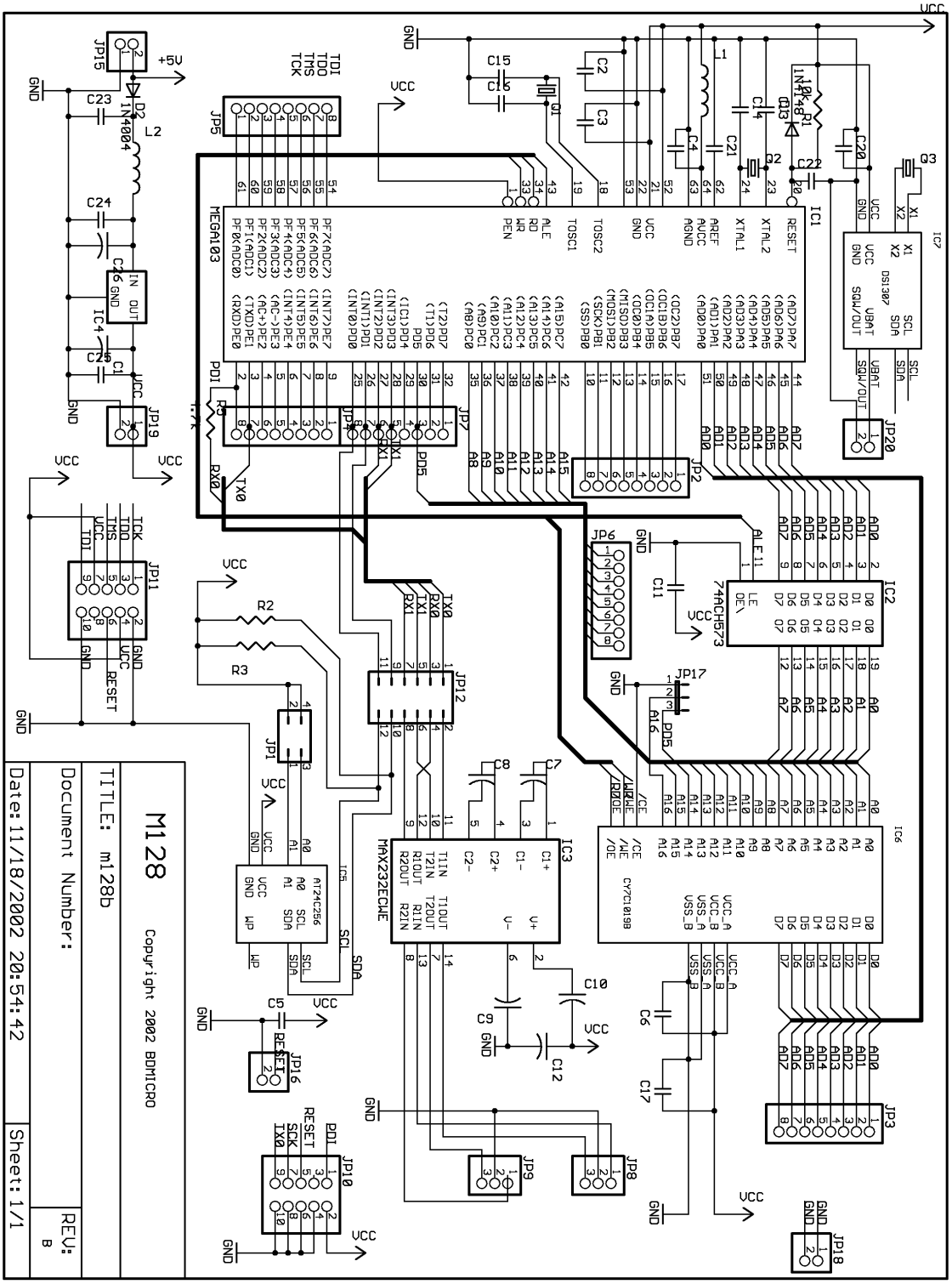
If you decide not to desolder the RAM chip, but find that you still must use PORT A or PORT C for general purpose I/O, you must take special precautions in your initialization code. Specifically, you must configure PORTG0, PORTG1, and PORTG2 as outputs, and set PORTG0 low, PORTG1 high, and PORTG2 low as follows (in the C programming language):

```
DDRG   = 0x07;  /* enable PG0, PG1, & PG2 as outputs */
PORTG &= 0xf8;  /* set PG0, PG1, & PG2 low */
PORTG |= 0x02;  /* set PG1 high */
```

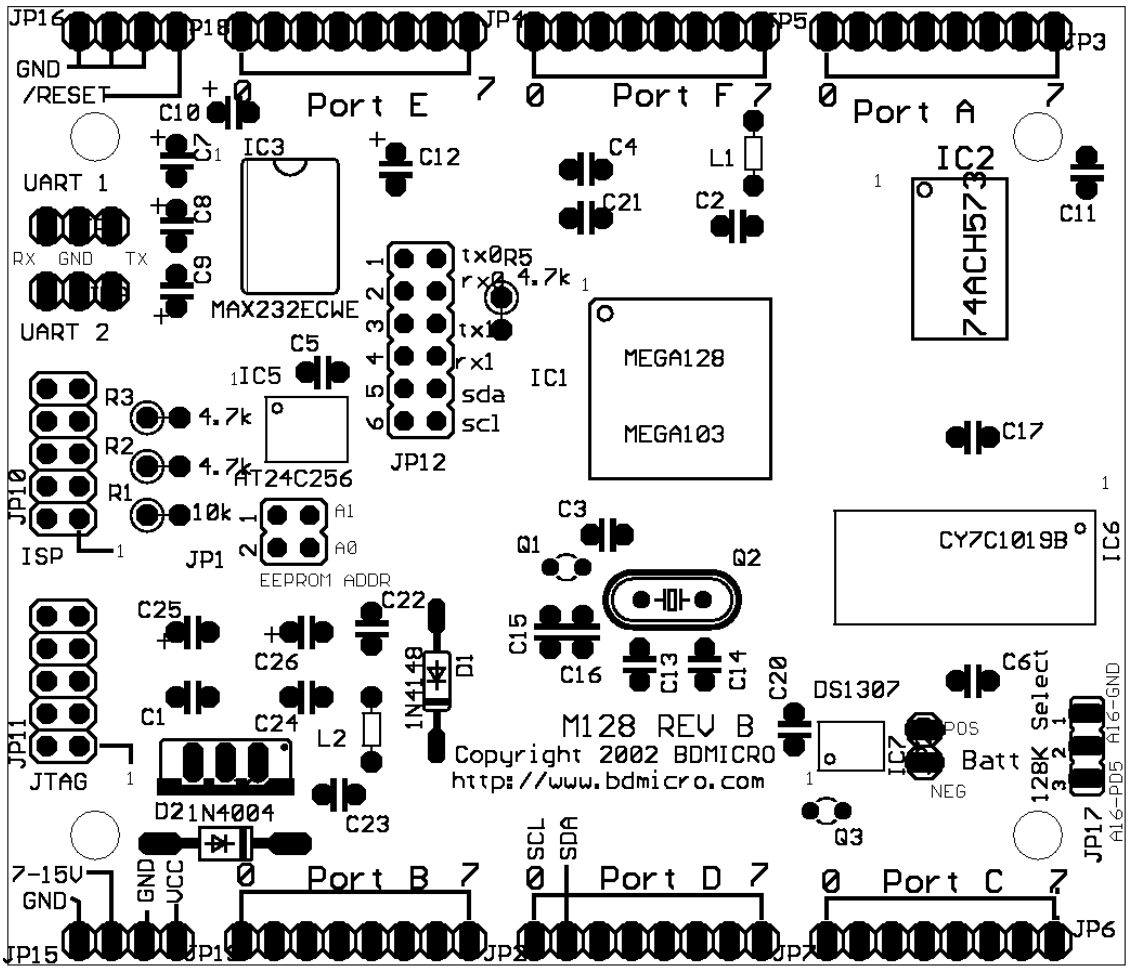
These settings disable the RAM chip output drivers, setting its /output-enable line high which causes its outputs go to a high impedance state. You must do this before you configure PORT A and/or PORT C as an output port, otherwise you risk damage to the RAM chip or the ATmega128 or both.

A future revision of the M128 board will offer a jumper or dip switch to disable the RAM chip and latch.

8 Schematic Diagram



9 Silkscreen



Note that the silkscreen indicates 4.7K Ohm resistors for R2 and R3 (I^2C pull-ups), it was later found that using 1.8K Ohm resistors allows for higher I^2C clock rates and/or longer I^2C bus lengths, while still remaining within I^2C specifications.

10 Header Descriptions

10.1 JP15+JP19

These two, 2-pin headers are physically arranged as a single 4-pin header in the lower left hand corner of the board. From left to right, the pins are:

- GROUND
- 7-15 VDC
- GROUND
- VCC (+5V regulated power)

One can supply power to the board via the 7-15 VDC connection. It is also possible to tap off of the VCC pin in this case to power off-board components as long as they don't draw too much current. Alternatively, one can supply regulated 5V power to the VCC pin and bypass the on-board power regulation circuit. This may be desirable if you already have an ample supply of 5V regulated power in your application. In this case, be sure and **do not** connect external power to the 7-15 VDC input.

10.2 JP16+JP18

These two, 2-pin headers are physically arranged as a single 4-pin header in the upper left hand corner of the board. From left to right, the pins are:

- GROUND
- GROUND
- GROUND
- /RESET

10.3 JP10 (ISP)

This is the Atmel standard 10-pin ISP programming header. Use this header to connect to the 10-pin ribbon cable from your STK-500 or other programmer.

10.4 JP11 (JTAG)

JP11 is the 10-pin JTAG debugging and programming header. If you have a JTAG-ICE device, you can connect the 10-pin ribbon cable to this header to debug and program the ATmega128 CPU.

10.5 UART1

UART1 provides RS232 level shifted lines to the first UART of the ATmega128 (RX0 and TX0).

10.6 UART2

UART2 provides RS232 level shifted lines to the second UART of the ATmega128 (RX1 and TX1).

10.7 JP1 (EEPROM ADDR)

JP1 is a 2x2 header for selecting the two least significant bits of the serial EEPROM's I^2C address. Leave the jumper off to set the bit to 0, install the jumper to set the bit to 1.

10.8 JP12

JP12 is used to disconnect the peripheral chips from the AVR so that you can regain the AVR lines at the port headers, even if your board is fully populated. Install the jumpers to use the features, remove the jumpers to disconnect them. This provides connections for UART0 (tx0, rx0), UART1 (tx1, rx1), and the I^2C lines (scl, sda).

10.9 JP17 (128K Select)

JP17 selects what to do with the upper 64K of RAM on the 128K RAM chip. Since the ATmega128 can only access 64K directly, and the installed chip is 128K, this jumper either ties the high A16 address line to ground, ignoring the upper 64K of RAM, or to PORTD5, allowing the upper 64K to be accessed by toggling the state of PORTD5.

10.10 JP20 (Batt)

JP20 (labeled `Batt` on the board) is the battery connector for the Dallas DS1307 real time clock. The side labeled `Pos` goes to the positive battery terminal and the side labeled `Neg` to the negative.

10.11 JP2 (PORT B)

The 8 pins on this header are wired directly to PORT B of the ATmega128. Pins 0 and 7 are labeled on the silkscreen of the board for easy idenfication.

10.12 JP3 (PORT A)

The 8 pins on this header are wired directly to PORT A of the ATmega128. Pins 0 and 7 are labeled on the silkscreen of the board for easy idenfication.

10.13 JP4 (PORT E)

The 8 pins on this header are wired directly to PORT E of the ATmega128. Pins 0 and 7 are labeled on the silkscreen of the board for easy idenfication.

10.14 JP5 (PORT F)

The 8 pins on this header are wired directly to PORT F of the ATmega128. Pins 0 and 7 are labeled on the silkscreen of the board for easy idenfication.

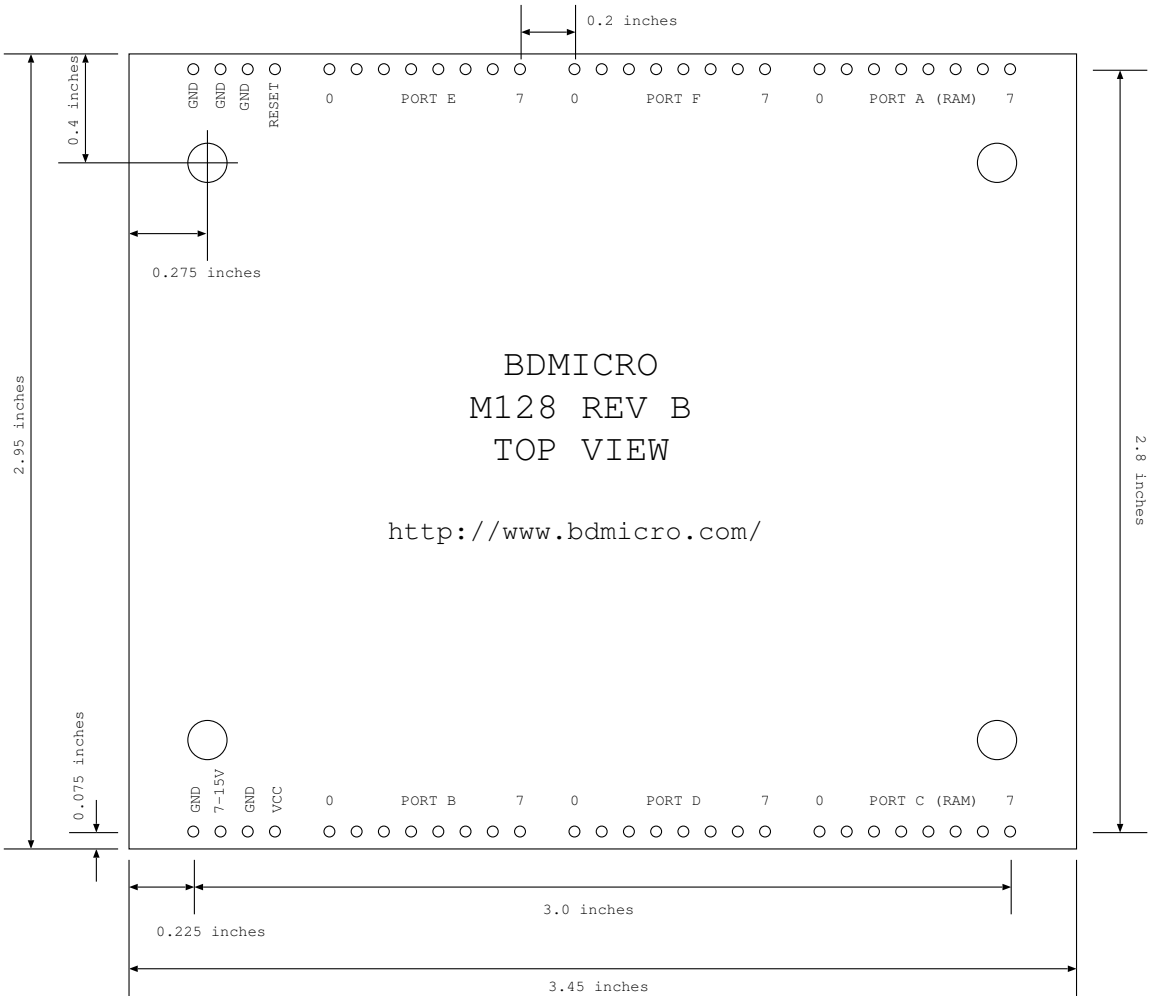
10.15 JP6 (PORT C)

The 8 pins on this header are wired directly to PORT C of the ATmega128. Pins 0 and 7 are labeled on the silkscreen of the board for easy idenfication.

10.16 JP7 (PORT D)

The 8 pins on this header are wired directly to PORT D of the ATmega128. Pins 0 and 7 are labeled on the silkscreen of the board for easy idenfication.

11 Mechanical Drawing



12 M128 REV B Parts List

1 ATMEGA128-16AC at IC1 (Digikey ATMEGA128-16AC-ND)

2 SN74AHC573DW at IC2 (Digikey 296-4612-5-ND)

1 MAX232DW at IC3 (Digikey 296-6937-5-ND)

1 UA7805CKC at IC4 (Digikey 296-1974-5-ND)

1 AT24C256W-10SI-2.7 at IC5 (Digikey AT24C256W-10SI2.7-ND)

1 CY7C1019B-15VC at IC6 (Digikey 428-1004-ND)

1 DS1307Z at IC7 (Digikey DS1307Z-ND)

5 1 uF 50V electrolytic radial capacitor at C7, C8, C9, C10, C12 (Digikey P993-ND)

13 0.1 uF 50V 20% ceramic radial capacitor at C1, C2, C3, C4, C5, C6, C11, C17, C20, C21, C22, C23, C24 (Digikey 399-2127-ND)

2 10.0 uH fixed inductor at L1, L2 (Digikey TK4224-ND)

1 1N4148 diode, DO-35 at D1 (Digikey 1N4148FS-ND)

1 1N4004 diode, DO-41 at D2 (Digikey 1N4004GICT-ND)

2 47 uF 35V electrolytic radial capacitor at C25, C26 (Digikey P11232-ND)

1 16.000 MHz 20 pF crystal, HC-49/US at Q2 (Digikey X176-ND)

1 32.768 KHz crystal at Q3 (Digikey X801-ND)

2 22 pF 100V ceramic disc capacitor at C13, C14 (Digikey 1305PH-ND)

1 10K ohm 1/4W 5% carbon film resistor at R1 (Digikey 10KQBK-ND)

2 1.8K ohm 1/4W 5% carbon film resistor at R2, R3 (there is no R4, do not install R5 - use a wire jumper instead) (Digikey 1.8KQBK-ND)

The following parts are 0.1 inch screw terminal connectors that can be used for the port headers. They add significant cost, but they make it easy to interface to your board. These are optional.

6 8 position terminal block 2.54 mm center at JP2, JP3, JP4, JP5, JP6, JP7 (Digikey 277-1279-ND)

2 4 position terminal block 2.54 mm center at JP15+JP19, JP16+JP18 (Digikey 277-1275-ND)

12.1 Headers

- .1 inch dual row headers: 1 2x2 (JP1), 2 2x5 (JP10, JP11), and 1 2x6 (JP12)
- .1 inch single row headers: 3 1x3 (UART 1, UART 2, JP17), 1 1x2 (Batt), 2 1x4 (JP15+JP19, JP16+JP18), and 6 1x8 (JP2, JP3, JP4, JP5, JP6, JP7)