“*Mega128-DEV*elopment Board”

Progressive Resources LLC
4105 Vincennes Road
Indianapolis, IN 46268
(317) 471-1577
(317) 471-1580 FAX
http://www.prllc.com

GENERAL

The ***Mega128-Dev***elopment board is designed for prototyping and laboratory use. The board hosts a ATMega128 processor along with two RS232 ports, a Real-Time-Clock, 64K of battery-backed SRAM, two Optrex® compatible LCD connectors, JTAG and ISP ports, an MMC/SD slot, and a fully decoded expansion bus connection.

Mega128 AVR® Boot Loader

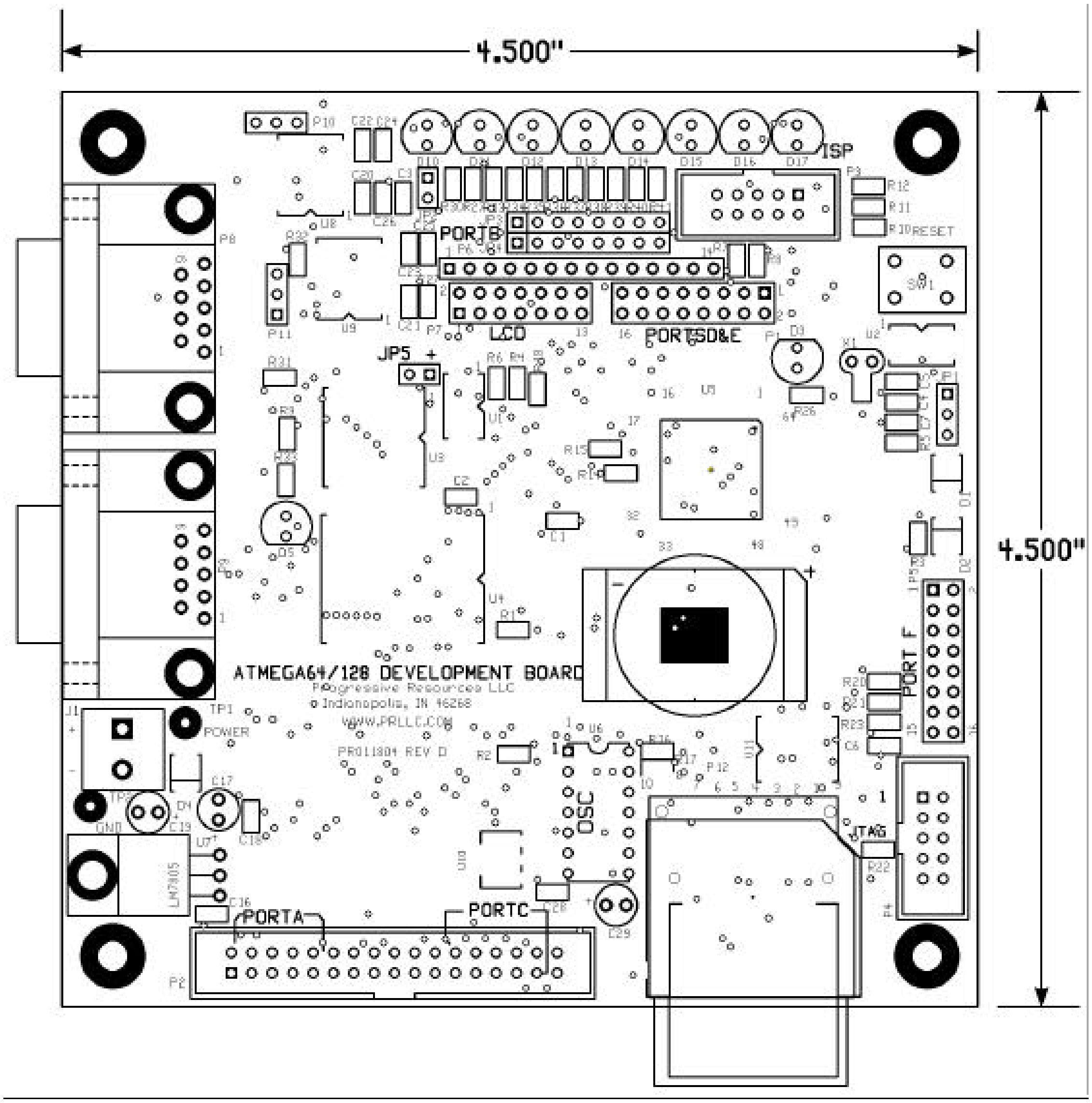
The ***Mega128-Dev***elopment board now comes programmed with the AVRBL-128, Mega128 AVR Boot Loader. The AVR Boot Loader (AVRBL) is a bootstrap loader that, once programmed into the AVR boot-block memory area, allows reprogramming of AVR microcontrollers without need for a chip programmer. The AVRBL makes use of the selfprogramming features of the AVR microcontrollers to allow in-circuit reprogramming. Once the AVRBL is programmed into the microcontroller, it remains resident until the chip is erased. Application programs require only a very minimum software interface to use the AVRBL. The PC application to interface to the boot loader and more documentation about interfacing to the boot loader are available free at http://www.prllc.com. (Some notes are provided on the following pages.)

Other features of the development board include:

- Two RS232 ports with 9-Pin D-Shell as well as a jumper header for convenient connection.
- Up to 128K of In-System Programmable FLASH memory with 4K of EEPROM and 4K of Internal RAM.
- The optional on-board SD/MMC memory module connector can provide tremendous storage and can be accessed with MSDOS compatibility using the *FlashFile-SD* source code package (sold separtely).
- Eight, 10 bit, Analog Inputs, using either internal or user supplied reference.
- Nine I/O controlled LEDs, 8 of which are jumper selectable.
- A PCF8563 Real-Time-Clock IC with a 32KHz "watch" crystal for on-board Real-Time operations. Battery backed with a 3V lithium coin cell.
- 64K of external SRAM, battery backed with a 3V lithium coin cell.
- A universal clock socket allows for "canned oscillators", as well as a variety of crystals, ceramic resonators, and passive terminations. A 14.7456Mhz oscillator comes standard with the unit, providing accurate baud rates across a wide range.
- Two Optrex® compatible LCD connectors, one single-in-line, one dual.
- 0.1" centered headers provide for simple connection to the processor special function pins and I/O.
- 10-pin, polarized, ISP and JTAG connections are provided for in-system programming and debugging. Mega128s can also be programmed through either RS232 port when using an appropriate boot loader application.
- On-board regulation allows for power inputs from 8-38VDC with an LED power indicator.
- Termination is provided for 5VDC output at 250ma

SPECIFICATIONS

-	Voltage range	8 V to 38 VDC
-	Power consumption	250 mW (nominal)
-	Dimensions:	W 4.5 inches x H 4.5 inches
-	Mounting	0.125" holes and Rubber feet, 4 places
-	Weight	~3 OZ
-	Operating temperature	0 deg. C to +60 deg. C
-	Storage temperature	0 deg. C to +85 deg. C
-	Humidity	0% to 95% at +50 deg. C (non-condensing)



APPLICATION NOTES:

Power

J1 (screw terminal connector) is the power input point. Acceptable voltages are 8 – 38 VDC (J1-1 is +, J1-2 is ground).

JP2 is an output of the regulated 5 VDC that may be used to power other devices. Note, however, that the LM7805 does not have a heat sink and so the actual available power output is somewhat limited, depending on the input voltage and power being consumed. Check the LM7805 regulator specification for details.

Serial Connections

P8 and P9 are standard DB-9 connectors usually used to connect to a PC. The TX signal is on P8-2 (P9-2) and the RX signal on P8-3 (P9-3). These are RS-232 level signals.

P10 and P11 also provide connections for the RS-232 level serial signals. On each connector, pin 1 is the TX signal, pin 3 is the Rx signal and pin 2 is ground.

ISP Connection

P3 is provided for In-System-Programming of the Mega128 using appropriate ISP programming hardware. This connector is compatible with TheCableAVR ISP programmer available from Progressive Resources LLC, **http://www.prllc.com/.**

JTAG Connection

P4 is provided for In-System-Programming and Debugging of the Mega128 using appropriate JTAG hardware. This connector is compatible with Atmel's JTAG-ICE.

For boards released after January 28, 2004, the JTAG and OCD Fuse bits are enabled, however the installed bootloader sets the JTD bit (disabling the JTAG and OCD connections) so all analog channels can be used. To use the JTAG-ICE, check "Connect on Reset" when prompted by AVR Studio, so the JTD bit does not get set before the JTAG-ICE can communicate with the AVR processor.

MMC/SD Memory Connection

P12 is provided for Multi-Media and Secure-Data Flash Cards. These memory cards utilize the SPI interface of the ATMega128 allowing for large amounts of memory to be accessed without consuming many I/O pins.

Parallel Ports

Parallel ports A, B, C, D, E and F are connected to various headers and labeled clearly on the board. Refer to the schematics to locate the connection points of these ports. When possible, the entire port (i.e. PORTB) is brought out to the connector in sequence. Ports A and C can be found on the memory expansion connection and are available as I/O when the Mega128 is not in external memory mode. These are normal TTL-level signals with or without pull-ups depending on the port initialization set up in the software.

Port F (P5 odd pins) has a parallel row of ground pins next to it (P5 even pins) so that enabling the built-in pull-up resistors and then using two-pin jumpers to ground any pins that need a logic 0 applied for input purposes can effect simple input signals. This also provides a convenient ground reference when measuring analog voltages with the internal A/D converter (ADC).

Port B (JP4) has a parallel row of pins (JP3), each of which is connected to an LED through a 510-ohm series resistor to +5 VDC. Jumping any of the pins of JP4 to the corresponding pin of JP3 allows the use of the on-board LED's as an output. Because the LED's are connected to +5 VDC and the port is sinking the LED current, the LED will be on for any pin that outputs logic 0.

Memory Expansion Connection

P2 is a dual-in-line connector that can serve two purposes. If the Mega128 is setup in external memory mode, the fully decoded address, data, and access strobes (RD, WR, and ALE) are available to P2 (along with VCC and GND). If the Mega128 is in internal memory mode, then ports A and C are available at the connector as general, digital I/O.

P2-1 and P2-2 are used to enable or disable the on-board SRAM. If there is a jumper from P2-1 to P2-2, the on-board SRAM is disabled. This allows for entire external memory address space to be dedicated to whatever devices may be plugged in to the memory expansion connector. P2-1 (~RAMEN) can also be set to TTL high by off-board logic in order to disable the on-board SRAM for peripherals that may share the memory space.

System Clock

As supplied, the system clock is 14.7456 MHz. U6 contains the crystal and caps necessary for the oscillator. Replacing U6 with a TTL, crystal, ceramic resonator, or RC oscillator, or a different integrated oscillator unit allows changing the system clock if necessary.

Analog-to-Digital Converter Connections

JP1 provides the connections to control the V_{ref} for the ADC. A new V_{ref} may be connected to JP1, pin 2, if desired. Or the internal connections to V_{ref} may be controlled by software. 5V can be used as a reference by jumpering JP1-2 to JP1-3.

LCD Interface Connections

P6 and P7 are provided as convenient connections for Optrex compatible LCD displays. These connectors are wired in parallel and are driven from port B of the Mega128. The signaling definitions are as follows:

PORTB.0	RS
PORTB.1	RW
PORTB.2	E
PORTB.4-7	DB4-DB7

AVRBL-128, Mega128 AVR Boot Loader

There are two methods of running the boot loader, upon reset and by a direct call from the application code. The boot loader runs on UART0 at 115200 baud, XON, XOFF handshaking, 8 data bits and 1 stop bit.

The boot loader code is executed upon reset. If the boot loader does not receive instructions to load a new file with 5 seconds, then it jumps to the application code.

To call the boot loader from your application code consider using a condition and jump instruction.
For example:

```
if (UDR0 == '$')        // did I receive a $ on the USART?
{                       // if true, then jump to the AVRBL
        #asm
        JMP   0xFC00
        #endasm
}
```

This example code checks to see if the character received on USART is a ‘$’. If so, then the JMP instruction is executed, exiting the main application and entering the boot loader directly. If the boot loader does not receive any instructions within a selected amount of time it will jump to 0x0000, effectively resetting the processor and starting the main application code. Otherwise, if the boot loader does receive instructions, the opportunity is then provided to download new firmware. The PC application allows the user to establish what characters are used by the application to start the boot loader.

After the AVBRL is started (via a reset, a power-up, or a jump from the main application), the following protocol must be observed. The PC application provided handles the protocol for you with its default settings or you can create a custom application. In either case the protocol is:

1. Upon power-up, reset, or as a result of a jump from the main application, the AVRBL sends a ‘^’ (BOOTLOADER_ACTIVE_CHAR) at your selected baud rate.

2. The host is then required to send the three-character entry sequence. This is used to prevent an inadvertent attempt of reprogramming from taking place. If the AVRBL does not receive these characters within the timeout period, the AVRBL tests to see if there is code located in the main application area of flash. If there is, then the AVRBL jumps to it, otherwise, execution stays within the AVRBL indefinitely, waiting for the entry sequence.

3. Once the three-character entry sequence has been sent, the bootloader sends the version string (Vx.xx) followed by a ‘?’ (READY_CHAR).

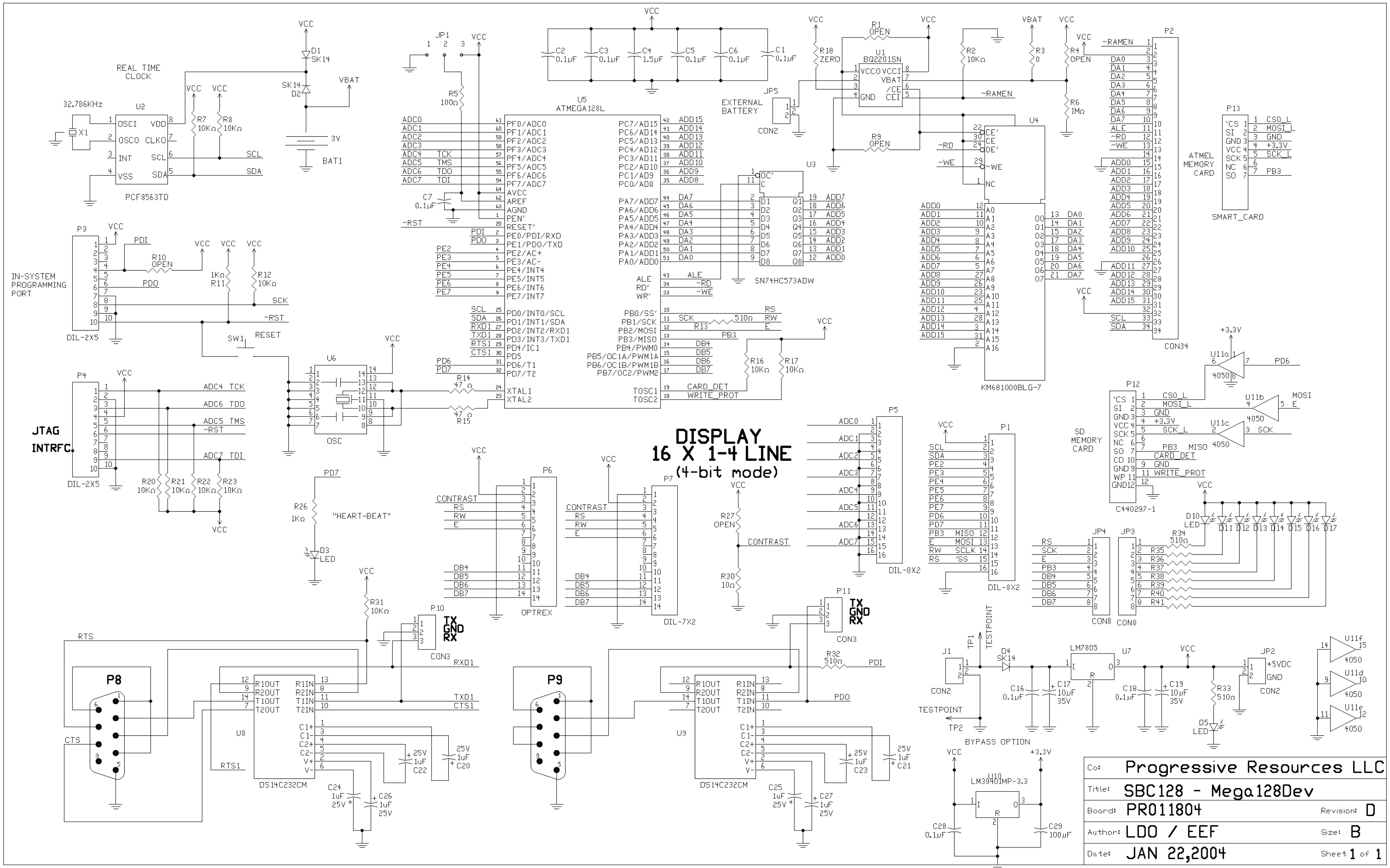
4. Upon receipt of the READY_CHAR, the host application should send the hex file for the new/updated application program observing an X-ON / X_OFF handshaking protocol to control data flow. The handshaking is very important as the flash memory area writes much more slowly than the serial port can send data. The programming software continues sending the hex file until it is all sent. After each line of “.hex” file is received by the bootloader, one of three characters is transmitted by the bootloader:
    - ‘~’ Line received with no errors.
    - ‘%’Line received with no error, but an error occurred while flashing.
    - ‘-‘ Checksum error detected while receiving the line.

5. After the programming is complete, the AVRBL sends either a ‘#’, meaning the programming is all right, or an ‘@’ indicating that an error has occurred and the program did not load successfully. In most cases an error during programming means that the main application program is corrupted and will need to be resent.

6. The AVRBL then starts the newly programmed application software. As stated in step 2, the AVRBL tests to see if there is code located in the main application area of flash. If there is, the AVRBL jumps to it, otherwise, execution stays within the AVRBL indefinitely, waiting for the entry sequence.

One final note about the **Mega128 AVR® Boot Loader**, the boot loader source code is available for purchase and as such provides the opportunity for customized boot loader solutions. The AVRBL-128 boot loader (as well as others) is available at http://www.prllc.com.

SCHEMATIC



Co:	Progressive Resources LLC		
Title:	SBC128 - Mega128Dev		
Board:	PR011804	Revision:	D
Author:	LDO / EEF	Size:	B
Date:	JAN 22,2004	Sheet	1 of 1