

MoBoStamp-pe BS2pe Motherboard (#28300)

General Description

The MoBoStamp-pe provides a compact, professional-grade platform for BASIC Stamp applications. With the MoBoStamp-pe and the assortment of available daughterboards, you will be able to integrate and package one-off or multiple application systems with ease. The onboard AVR coprocessors permit the offloading of compute-intensive and background tasks from the BASIC Stamp, yielding a high performance level, while retaining the BASIC Stamp's ease of programming. They come preprogrammed for digital I/O, analog input, pulse-width-modulated output, and frequency counting. Each coprocessor interfaces to both the BASIC Stamp and one daughterboard each and may be reprogrammed by the user. Potential uses include background serial I/O, floating point processing, and background servo pulsing.

Before using your MoBoStamp-pe, please read and understand this entire document.

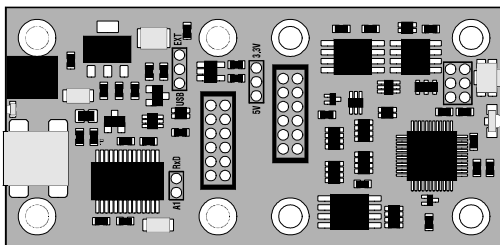
Features

- Compact size: designed to fit available packaging.
- BS2pe BASIC Stamp chip for high performance with low current consumption.
- 32K x 8 EEPROM for program and data storage.
- Built-in USB interface, capable of powering the entire board.
- Two sockets for plug-compatible daughterboards, enabling the easy integration of sensors and interface options.
- Two user-programmable Atmel AVR coprocessors, preprogrammed for digital I/O, PWM output, analog input, and frequency measurement.
- Multi-mode power sourcing: board can run from USB or external power.
- Multiple Vdd levels: 3.3V and 5V, jumper selectable.
- User-programmable multi-color LED for status indication.
- Programming header for future interface to an SX-KEY. This will allow programming the SX chip directly.

Application Ideas

- Robotics
- Remote Sensing
- Data Acquisition
- Industrial Control
- Desktop Appliances

What's Included



1 ea. MoBoStamp-pe Motherboard, preconfigured with:



4 ea. 3/16" dia. x 5/16" Threaded Standoffs

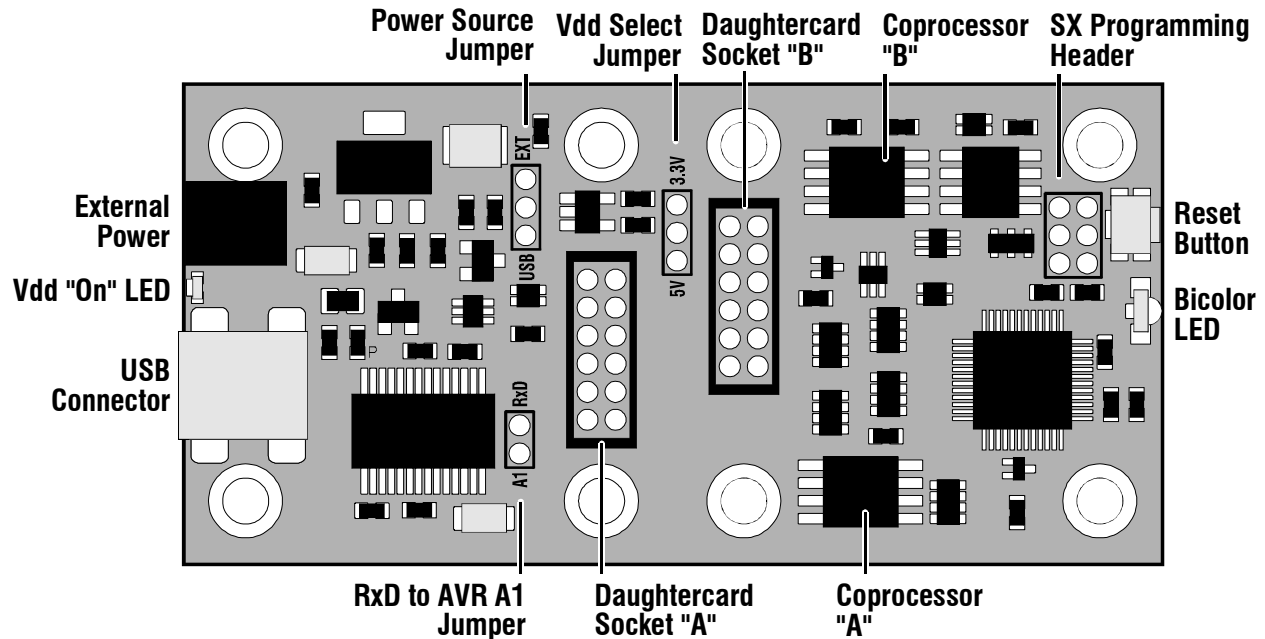


8 ea. 4-40 x 3/16" Panhead Machine Screws



3 ea. 2mm Jumpers

Interface Connections and Jumpers



External Power Connector

The external power connector enables the connection of a filtered, unregulated 6-9VDC power supply to power the motherboard. The plug and cable for this connector may be obtained separately from DigiKey, using the part numbers given under "Specifications" near the end of this document. Power applied here supplies both the onboard 5V regulator and the Vin connection to Daughterboard Socket "A".

Vdd "On" LED

This LED comes on when the board's Vdd is active. When the board is powered externally, this will occur whenever external power is applied. When the board is powered from the USB port, the LED will come on once the board has been connected to a USB port and the onboard USB chip enumerated by the PC.

For applications (such as light-sensing) where such a light source is undesirable, or for situations requiring minimum current draw, there is a "bow-tie" trace on the bottom of the board (under the external power connector) which can be cut to disable the LED.

USB Connector

The USB connector is the mini-B type. Compatible "A-to-mini-B" cables may be obtained at most local computer stores or from Parallax as part number 805-0006.

Power Source Jumper

The power source jumper can be placed to select either external or USB power. When external power is selected (upper position, marked **EXT**), the onboard regulator provides 5V to the rest of the board from Vin. Vin can come from either the external power connector or a daughterboard plugged into Daughterboard Socket "A". The USB chip, however, will continue to be powered by the host PC through the USB cable.

When USB power is selected (lower position, marked **USB**), the host PC provides 5V to the rest of the board from the USB cable. **Note:** *The USB "5V" supply can range anywhere from 4V to 5V.* When an accurate supply voltage is critical (e.g. for analog applications), it's best either to use an external supply or, if all circuitry can run from 3.3V, to select a Vdd of 3.3V.

Vdd Select Jumper

The Vdd select jumper allows the BASIC Stamp, the EEPROM, and both AVR coprocessors to run from either 3.3V or 5V. By selecting 5V (lower position, marked **5V**), Vdd is provided by the power source jumper, without further regulation.

By selecting 3.3V (upper position, marked **3.3V**), the onboard 3.3V regulator supplies these chips, as well as the "Vdd" pin on each daughterboard connector. In this position, the "5V" receptacle on each daughterboard connector continues to receive 5 volts.

RxD to AVR A1 Jumper

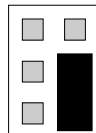
This jumper allows the incoming serial data from the USB interface to be connected to one of the AVR coprocessor pins. This will permit a properly programmed AVR to receive serial data in the background, relieving the BASIC Stamp of the task.

SX Programming Header

This six-pin header includes all the signals (plus RST) required to reprogram the SX with something other than the BS2pe interpreter. This function will be available via an adapter that the Parallax SX Key plugs into.

Important: Overwriting the BS2pe firmware is permanent. It is not possible to revert to BASIC Stamp operation once this has been done.

For normal operation, this header should always be configured with a jumper in the lower right-hand corner, thus:



Reset Button

The tiny reset button, when pressed, will reset the BASIC Stamp chip and both AVR coprocessors. It's designed more to be used in conjunction with a paperclip and a pinhole enclosure opening than with a finger; hence, its diminutive size. During debugging, resetting the board via the debug screen's DTR button will likely be easier.

Bicolor LED

This red/green LED is illuminated by pulling BASIC Stamp port P13 low for red or P14 low for green. Pulling both low simultaneously will show, at a distance, as amber.

Daughterboard Sockets

The daughterboard sockets receive the specially-designed daughterboards which, in conjunction with the onboard AVR coprocessors, perform sensing, interfacing, and other functions. They are labeled "A" and "B".

The daughterboards plug in parallel to the motherboard and will typically have connectors facing the ends of the motherboard. The daughterboard sockets are identical, with one exception: the "A" socket carries the Vin signal, while the "B" socket does not. This permits a daughterboard plugged into the "A" socket to power the whole system. That way, daughterboards can be designed, for example, with 24V-input DC-DC converters for industrial systems, or simple wall-transformer inputs for hobby systems. The Parallax PWR-I/O-DB card (part #28301) is an example of the latter.

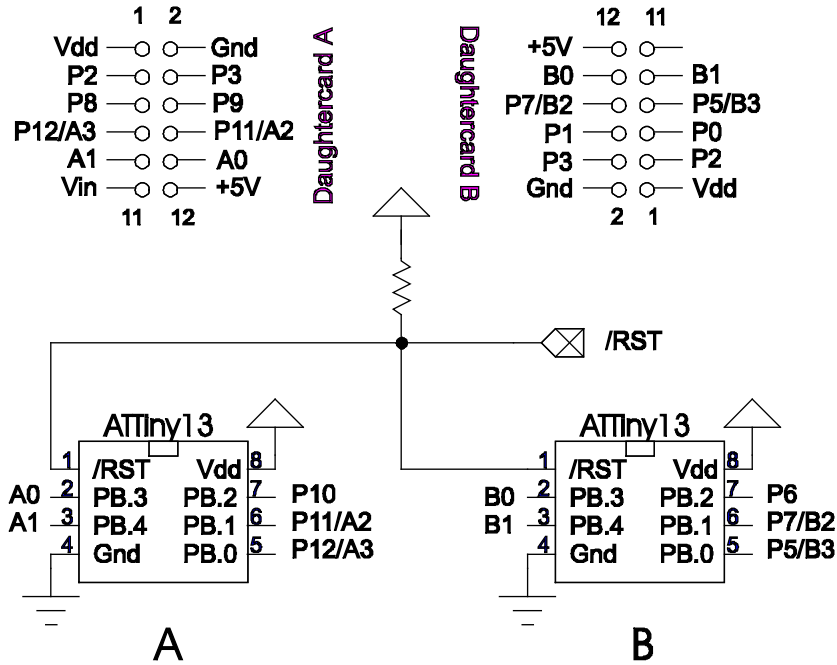
Important: Never insert jumper wires into the daughterboard sockets. Unlike 0.1" header sockets, these are not big enough to receive most jumper wires without damage.

The "A" daughterboard is sometimes referred to as the "Interface" daughterboard, since it is frequently used to interface to the outside world, with its connectors on the same end of the motherboard as the USB connector. Likewise, the "B" daughterboard is often called the "Sensor" daughterboard, since that's the socket where sensors are more likely to be plugged in. In reality, these names are arbitrary, since the capabilities of the two sockets are virtually identical.

In addition to the Vin line for socket "A", each daughterboard socket includes +5V, Vdd, Gnd, and eight signal lines. It is convenient to think of the signal lines in pairs. Their functions can be described as follows:

- **Common pulled-up pair:** These lines connect to both sockets and also to BASIC Stamp ports P2 and P3. They are pulled up to Vdd via 4.7K pull-ups. They can be used for daughterboard-to-daughterboard communication, without BASIC Stamp intervention. They can also be used with open collector drivers as poll inputs to the BASIC Stamp or as actual interrupts to a raw SX system.
- **Individual pulled-up pair:** These lines connect ports P8 and P9 to socket "A" and P0 and P1 to socket "B". They are pulled up to Vdd via 4.7K pull-ups. These ports can be used with the BS2pe firmware as an I2C interface. Daughterboards with I2C peripherals will use these lines as SDA and SCL.
- **AVR/BASIC Stamp shared pair:** Two lines to each daughterboard connect to both the BASIC Stamp and to the daughterboard's associated AVR coprocessor. These lines are not pulled up. Socket "A" receives P11/A2 and P12/A3. (A2 and A3 are pin designators for coprocessor "A", as shown below.) Socket "B" receives P7/B2 and P5/B3. A2, A3, B2, and B3, can be programmed as PWM outputs from the AVR chips.
- **AVR exclusive pair:** Two lines to each daughterboard connect exclusively to its AVR coprocessor. These are A0 and A1 for socket "A" and B0 and B1 for socket "B". These pins can be programmed as analog inputs.

The foregoing description is summarized in the following schematic fragment:



Coprocessors

The two AVR coprocessors interface to the daughterboards and to the BASIC Stamp as shown above. P6 and P10 are reserved for BASIC Stamp/AVR communication. These lines are pulled up to Vdd, so that open-collector comms (e.g. OWIN and OWOUT) may be utilized. The actual interface details will depend on the AVR firmware. See the GPIO-3 document for the firmware that comes preinstalled on the motherboard.

Getting Started

Before connecting your MoBoStamp to a PC, you should download and install the FTDI drivers necessary to operate the USB interface. These can be obtained, along with installation instructions, here:

http://www.parallax.com/html_pages/downloads/software/ftdi_drivers.asp

For programming the BASIC Stamp, you will need the Parallax BASIC Stamp Editor software, available for download here:

http://www.parallax.com/html_pages/downloads/software/software_basic_stamp.asp

Finally, to use the general-purpose I/O firmware pre-installed in the AVR coprocessors, download the GPIO user's guide here:

http://www.parallax.com/detail.asp?product_id=28300

Once these tasks have been completed, make sure your MoBoStamp is jumpered for USB power, and plug it into your PC. You should hear a "boo-beep" signal from the PC, indicating that the device has been recognized, and the green Vdd "on" LED should light. You are now ready to start programming.

Open the BASIC Stamp Editor, and key in the following program:

```
' {$STAMP BS2pe}
' {$PBASIC 2.5}

DO
  HIGH 14
  LOW 13
  PAUSE 500
  HIGH 13
  LOW 14
  PAUSE 500
LOOP
```

Hit Ctrl-R to load and run the program. You should see the bicolor LED alternately flash red and green. You've just run your first MoBoStamp program!

AVR Coprocessors

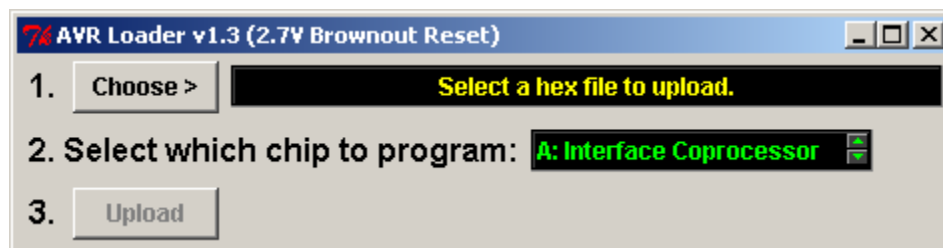
The coprocessors on the MoBoStamp board are Atmel ATtiny13s. They run on an internal 9.6 MHz RC clock and approach 9 MIPS (million instructions per second), depending on the actual firmware used. These controllers include 10-bit ADC (analog-to-digital conversion) capability, as well as built-in timers for PWM (pulse-width modulation) output, among other timing functions.

Each coprocessor connects to both the BASIC Stamp and one each of the daughterboards. This permits the BASIC Stamp's interaction with a daughterboard to be intermediated by the coprocessor, possibly in the background, thus relieving the BASIC Stamp from much of the work it would otherwise handle by itself. In addition, the built-in ADC and PWM functions permit the coprocessors to serve as analog peripherals for motherboards that require analog interfacing.

Different daughterboards will require, and be provided with, different AVR firmware. This firmware will come in the form of pre-assembled hex files, which can be uploaded to the coprocessors' onboard flash memory. This is done with the AVR Loader program, LoadAVR.exe, available here:

http://www.parallax.com/detail.asp?product_id=28300

To use the loader program, copy it to any folder on your hard drive, and double-click on it to start it up. You will first be asked to locate the BASIC Stamp Editor program, Stampw.exe. Once you've found and selected it, the following window will appear:



In step 1, click the "Choose" button to bring up a file dialog. Locate the desired AVR .hex file, and click "Open". In step 2, select which coprocessor to load: "A", "B", or both. In step 3, just click the button, and Stampw.exe will be invoked. Once the AVR loader program is uploaded to the BASIC Stamp, a debug window will pop up showing the AVR programming progress. If everything goes okay, the AVR chip(s) will be programmed with the new code.

Important: Once the AVR loading process has begun, do not interrupt it: let it run to completion.

In the event that you install an updated BASIC Stamp Editor after using LoadAVR at least once, you will want to make sure that the loader uses the new editor. This can be accomplished in two ways:

1. Make sure to uninstall the older BASIC Stamp Editor, or
2. Delete the file LoadAVR.ini that appears in the same directory as LoadAVR.exe.

Either action will force LoadAVR.exe to request the location of the new editor software the next time it is run.

It is possible, and even encouraged, for users to write their own firmware for the AVR controller. You can download a complete development system for the AVR processors free of charge from the Atmel website:

http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725

Coprocessor programs written for the ATTiny13, should use the internal 9.6Mhz RC clock. The other fuse settings configured automatically by the loader program are:

- Self-programming: Disabled
- Debug wire: Disabled
- Brown-out reset: Enabled, 2.7V
- Serial programming: Enabled
- Preserve EEPROM during chip erase: Enabled
- Watchdog timer always on: Disabled
- Divide clock by 8: Disabled
- Startup time: 4mSec after reset

These settings cannot be changed.

Startup Time

Due to the 4mSec startup time for the AVR coprocessors, a BASIC Stamp program may well begin before the AVRs come out of reset. For this reason, you should always put a `PAUSE 5` at the beginning of each program that uses one or more coprocessors. This will give the coprocessors time to start before being accessed.

Serial Data Echo

When the BASIC Stamp starts up, the Editor requires that it echo any characters sent to it by the PC. Hence, BASIC Stamps are configured in hardware to do just that. But this is not always a desirable thing during the execution of an application program. Therefore, a provision has been made to disable the echo. Port P4 is reserved for this purpose and, when pulled low (e.g. `LOW 4`), will disable the echo from RxD to TxD.

Port Summary

Port	Description and Typical Use	Daughterboard	4.7K pullup?
P0	General-purpose pin or SDA.	B	Yes
P1	General-purpose pin or SCL.	B	Yes
P2	General-purpose pin or poll input.	A & B	Yes
P3	General-purpose pin or poll input.	A & B	Yes
P4	Receive data echo enable. Pull low to disable echo.	(none)	Yes
P5	General-purpose I/O. Shared with AVR "B", port B3.	B	No
P6	Communication port for coprocessor "B".	(none)	Yes
P7	General-purpose I/O. Shared with AVR "B", port B2.	B	No
P8	General-purpose pin or SDA.	A	Yes
P9	General-purpose pin or SCL.	A	Yes
P10	Communication port for coprocessor "A".	(none)	Yes
P11	General-purpose I/O. Shared with AVR "A", port A2.	A	No
P12	General-purpose I/O. Shared with AVR "A", port A3.	A	No
P13	Red LED. Pull low to turn on.	(none)	No
P14	Green LED. Pull low to turn on.	(none)	No
P15	Reserved for AVR programming. DO NOT USE!	(none)	No

Specifications

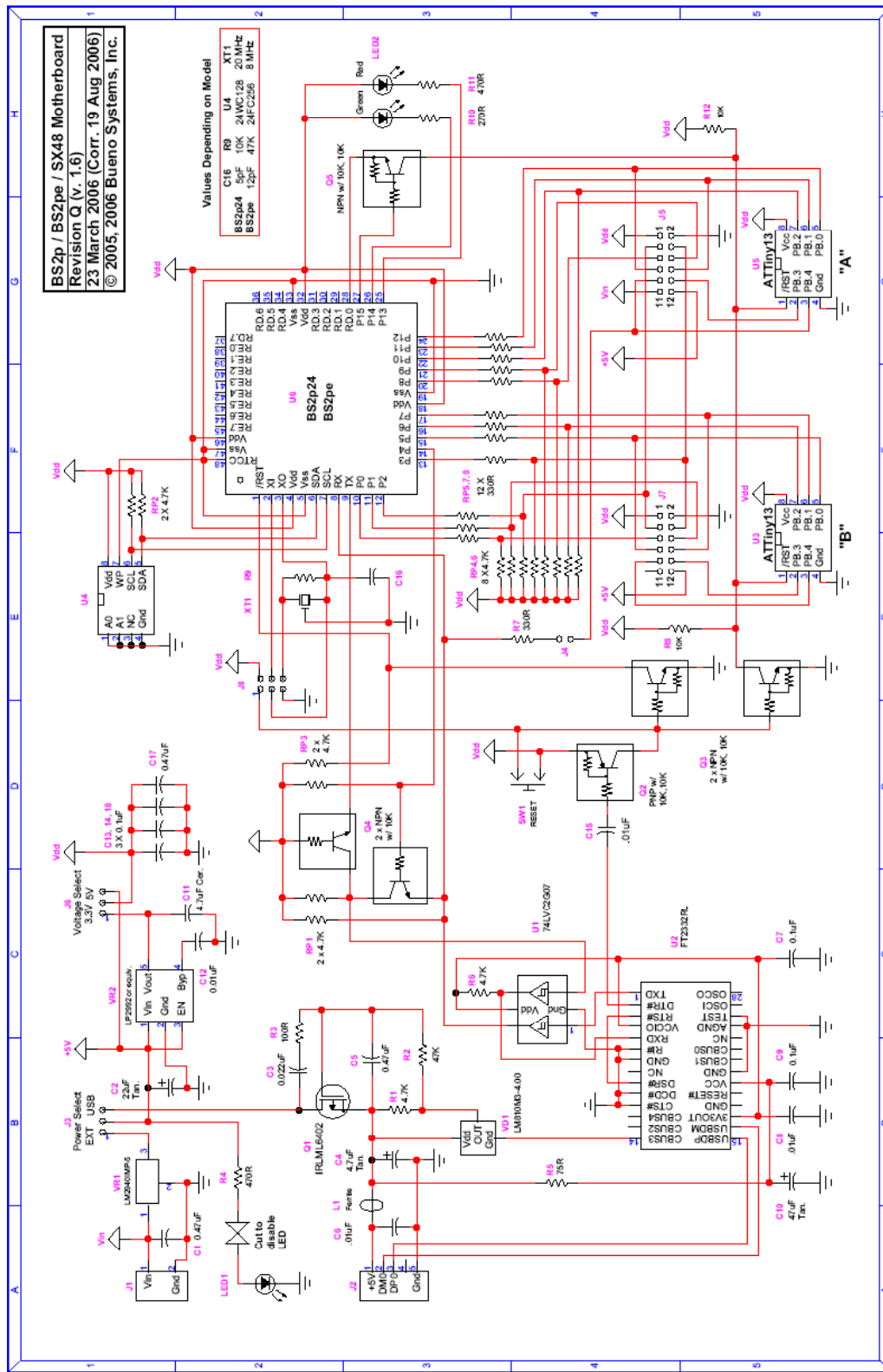
Circuit Board Size	2.75" x 1.35"
Nominal Daughterboard Size	1.35" x 1.35"
External Supply (optional)	6 – 9 VDC, 18mA minimum (3.3V, power LED disabled, no daughterboards)
USB Supply (from PC)	4 – 5 VDC
BASIC Stamp Clock	8MHz
Coprocessor Clocks	9.6MHz
USB Connector	Mini-B
Daughterboard Connectors (2mm)	Mating Header: Hirose DF11-12DP-2DSA(24)
External Power Connector (2mm)	Mating Receptacle: Shell: Hirose DF3-2EP-2C (DigiKey H4035-ND) Black Wire: DigiKey H2BXT-10112-B4-ND Red Wire: DigiKey H2BXT-10112-R4-ND

Sample Programs

Sample programs, written for various daughterboards, are available from the Parallax website:

http://www.parallax.com/detail.asp?product_id=28300

Schematic



Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Parallax:](#)

[28300](#)