

---

# PropIO V2

---

Wayne Warthen  
N8VEM Project

---

March 17, 2014

---

**Contents**

Summary ..... 2

Architecture ..... 3

Compatibility..... 5

Construction..... 5

Configuration ..... 8

Connectors ..... 8

Testing..... 8

Usage..... 9

References ..... 10

# ECB PropIO V2

A Propeller-based peripheral support board for the N8VEM ECB processors.

Wayne Warthen  
[wwarthen@gmail.com](mailto:wwarthen@gmail.com)

17 March 2014

## Summary

The PropIO is a peripheral support board for the N8VEM projects based on the [Parallax Propeller P8X32A MCU](#) as an intelligent I/O controller. The Propeller MCU is ideally suited to this role as it allows for simple connection of multiple peripheral devices and utilizes a custom EEPROM based firmware that can be enhanced over time.

The PropIO V2 is intended to interface with a host processor/controller on the ECB bus such as the N8VEM SBC. The onboard EEPROM firmware of the board must be programmed with the correct firmware for the host processor/controller being used. The host software/firmware and the Propeller EEPROM firmware must act cooperatively using a prearranged protocol to communicate. At present, the pre-built PropIO V2 EEPROM firmware is intended to work with a host processor/controller running the RomWBW firmware.

Note that the PropIO V2 does support a standalone operation mode in which it can be powered by a +5V DC Jack without being connected to an ECB bus. The idea is that it would be possible to create a Propeller firmware allowing the board to act independently of a host controller – perhaps acting as a simple single board computer or terminal emulator.

The PropIO V2 supports the following peripherals:

1. VGA Monitor

A DB-15 VGA monitor connector is provided and the board will drive any standard VGA monitor. The Propeller is capable of a variety of text and graphics modes. However, the current firmware implements standard 80X25 text.

2. PS/2 Keyboard

A standard PC compatible PS/2 keyboard connector is provided and supports buffered keyboard input.

### 3. SD Card

A Mini-SD Card connector is provided which allows block I/O with any standard SD, SDHC, or SDXC memory card. A host would typically utilize the SD Card to provide a solid-state disk drive.

### 4. RS-232 Serial Port

A 10 pin header is provided to support standard RS-232 serial connections. The header requires a pigtail to convert to a standard RS-232 connector. The serial port is a simple 3-wire interface (send, receive, ground). No flow control is supported, however the Propeller firmware does implement a small buffer. In the present firmware, the serial port is not addressable by the host controller, but can be used for in-situ programming of the Propeller EEPROM firmware as described below. A future firmware upgrade will allow the host processor/controller to utilize this serial port.

### 5. Speaker

A small, low fidelity speaker is provided to allow generation of simple sounds. This is mostly intended to allow for various “beep” codes as desired. The current firmware does not allow the host processor/controller to directly control the speaker, but this should be possible with a simple upgrade to the onboard EEPROM firmware.

Version V2 of the PropIO is a significant redesign of the original version and combines features of the original V1 with the ParPortProp. The following architecture changes were made in V2:

- Hardware latch to assert the ECB /WAIT signal
- Adds an onboard speaker
- Adds an onboard RS-232 port
- Switches the SD Card port from a Micro-SD socket to a Mini-SD socket
- Allows standalone operation powered directly by a +5V jack
- Removes the second Propeller chip
- Removes the breadboarding area

## Architecture

The core of the PropIO V2 design is the Parallax Propeller MCU. The Propeller is an 8 core parallel processor with 16 general purpose I/O signals. The PropIO V2 uses some of these I/O signals to interface with the ECB bus. The others are used to interface with the support peripheral connections. In this way, the Propeller acts as a powerful, intelligent I/O controller sitting between the ECB bus and the peripheral connectors.

The Propeller runs an interpreted language called Spin as well as providing an assembly language. The code is stored in an EEPROM attached directly to the Propeller. At startup, the Propeller loads the code from the EEPROM firmware into RAM and begins execution. The functionality of the Propeller is determined entirely by the pre-programmed code.

The PropIO V2 firmware is responsible for interfacing with the ECB bus as well as managing the attached peripherals. Most of the cores of the Propeller are dedicated to peripheral management. In general, one or more cores (Propeller “cogs”) are dedicated to each of the peripherals. One core is dedicated to the ECB bus interface. This core watches the dedicated ECB port addresses and dispatches work according to the data received.

The Propeller firmware implements fairly robust peripheral management. The most complex portion of this is the VGA display. A complete VGA display controller is implemented and includes simple VT-100 emulation. The PS/2 keyboard interface code implements complete, buffered keyboard input including management of the keyboard LED’s. The SD Card interface implements a direct block-level I/O with any standard SD Card via the SPI interface.

The ECB bus interface utilizes 4 sequential I/O port addresses. The base address is configurable, but is normally set at 0x40. The port functions are summarized below. The host processor/controller can read or write to each port as shown. Please refer to the PropIO V2 firmware source code included in the distribution archive for greater detail.

Port	Direction	Function
Base + 0	Read	Terminal Status: returns number of keyboard characters in buffer
	Write	Terminal Command: not implemented
Base + 1	Read	Terminal Read: read a character from the keyboard
	Write	Terminal Write: write a character to the display w/ ANSI emulation
Base + 2	Read	Disk Status: Return status of latest SD Card I/O operation
	Write	Disk Command: Initiate an SD Card I/O operation
Base + 3	Read	Disk Read: Return next byte of disk buffer to host
	Write	Disk Write: Send next byte to disk buffer

When the host processor/controller initiates a read or write to the PropIO, the PropIO asserts the ECB bus /WAIT signal to force the host to pause while the PropIO performs the requested function. The original PropIO V1 used a Propeller GPIO pin to assert /WAIT. This worked well, but the Propeller was not capable of asserting /WAIT fast enough if the host CPU was operating at higher clock speeds (over 8MHz for Z80). PropIO V2 implements the /WAIT assertion using a latch IC so that operation of much faster host clock speeds is possible. The latch asserts /WAIT immediately upon any host read/write to a PropIO port. The /WAIT signal remains asserted until cleared by the PropIO which it does upon completion of the requested host operation.

## Compatibility

The PropIO V2 is compliant with the standard N8VEM ECB bus architecture. In theory, any of the N8VEM ECB-based host CPU boards should be capable of interacting with the PropIO V2. However, as indicated above, it is mandatory that the host OS software and the onboard Propeller EEPROM firmware act cooperatively. The following table documents the combinations that are supported with existing host and Propeller firmware combinations:

Host Controller	Status	Host Software	PropIO Firmware
SBC V1/2	Supported	RomWBW v2.5.4	PropIO V2 Firmware v0.94
Mark IV	Experimental	RomWBW v2.6 Alpha	PropIO V2 Firmware v0.94
N8	Untested	RomWBW v2.5.4	PropIO V2 Firmware v0.94

The PropIO V2 firmware is included in the RomWBW distribution archives as a pre-built binary. It is found in the .zip file in the “Support\PropIO2” sub-directory.

At present, the only pre-built host firmware is for the SBC V1/2 – this host firmware is called N8VEM\_propio.rom. This pre-built firmware assumes that the only card connected to your SBC is the PropIO and therefore does not include floppy disk support. If you have additional boards installed, you can perform a custom RomWBW build to include support for your specific system configuration. Support for the N8 and Mark IV is untested, but should be possible via a custom build of RomWBW.

The RomWBW host software is compatible with either the original PropIO V1 or the new PropIO V2. However, the onboard Propeller firmware is slightly different for PropIO V1/V2, so you must ensure you are using the correct PropIO firmware for your board.

The latch based operation of PropIO V2 has been tested at host clock speeds of 16MHz (Z80) with no problems. It is likely that operation will be possible at significantly faster speeds, but this has not yet been tested.

## Construction

Board construction is similar to other N8VEM ECB boards. A Bill of Materials is posted on the Wiki in the [PropIO V2 Folder](#). For general PCB building instructions, there is a folder on the N8VEM Wiki called [Building the N8VEM for the Total Beginner](#) that contains some excellent advice on building boards. The documents in this folder generally refer to the N8VEM SBC board, but most of the advice applies perfectly to the PropIO V2.

The following construction options exist for the PropIO V2:

- **Optional Power Connector**

Connector P8 is only required if you wish to operate your PropIO V2 in standalone mode (meaning operation when **not** connected to an ECB bus). You may want to install this connector to enable initial testing and EEPROM programming before you install the card in the ECB bus. Installing the connector itself does **not** preclude you from using the board in an ECB bus, but you

should **not** apply power to the board via both P8 and the ECB bus at the same time. It is entirely possible to test and program the board when powered via the ECB bus, so this connector is optional.

- **RS-232 Level Shifter Options**

IC U10 is the RS-232 level shifter. There are a variety of ICs which are compatible with the MAX232A indicated in the schematic. Some of these require 1uF capacitors, while others require 0.1uF capacitors. You should install the correct capacitor value for C12-C16 based on the level shifter IC that you choose. The bill of materials specifies a TRS202ECN for IC U10 and this variant specifies 0.1uF capacitors, so you will see that the bill of materials also specifies 0.1uF capacitors for C12-C16. If you are using something other than a TRS202ECN, you should consult the datasheet and use the correct capacitor value.

In order to support either a ceramic or electrolytic capacitor for C12-C16, you will notice that a special footprint is used on the PCB. The footprint has three holes. You will use pins 1 & 2 to install an electrolytic capacitor being sure to match the positive lead of the capacitor with the pin 1 (square hole marked with a '+' sign). For ceramic capacitors (as specified in the BOM), you would use pins 1 & 3. Since ceramic capacitors do not have polarity, it does not matter which lead goes in each hole.

- **Programming onboard EEPROM**

As indicated above, the Propeller is an MCU which requires firmware in order to operate. The pre-built firmware compatible with RomWBW is found in the RomWBW .zip file in the "Support\PropIO2" subdirectory. The file is called "PropIO2.eeprom". This file must be programmed into U2 (24C512 EEPROM).

There are 3 options for programming the EEPROM as indicated below. Options 2 & 3 allow you to program the EEPROM in-situ (installed on the board). For these options, you must complete the construction of your board and install all components before attempting to program the EEPROM (in-situ programming relies on the Propeller).

1. Offboard programming using a PROM programmer

Many standalone PROM programmers have the ability to program a 24C512 using dedicated hardware connected directly to your PC. For this option, you will need to follow the directions for your PROM programmer.

2. Onboard programming using the Parallax PropPlug

- Procure the Parallax PropPlug and connect it to an available USB port on your PC.
- Apply power to the PropIO V2 (via either the ECB bus or the power jack) and confirm that both the 3V3 and 5V LEDs are lit.
- Connect the PropPlug to P7 using the silkscreen markings to ensure correct orientation.

- Start the Parallax Propeller Tool software on your PC to load and program the EEPROM:
    - ✓ Select Run -> Identify Hardware and confirm that the Propeller Chip is located.
    - ✓ Select File -> Open and locate and load the PropIO2.eeprom file
    - ✓ From the resulting Object Info dialog box, choose Load EEPROM
    - ✓ Wait for the programming operation to complete and make sure the verification was performed without error
  - Remove power from the PropIO V2.
  - Remove the PropPlug from P7.
3. Onboard programming using the RS-232 port
- Connect a standard N8VEM serial port pigtail to the P2 connector on your PropIO V2. Instructions for building a pigtail can be found in [Building the N8VEM for the Total Beginner](#) on the N8VEM Wiki.
  - Connect the pigtail to your PC via an RS-232 null modem cable.
  - Move the shorting plug at JP1 from it's default setting of 1-2 to position 2-3.
  - Apply power to the PropIO V2 (either via ECB bus or power jack) and confirm that both the 3V3 and 5V LEDs are lit.
  - Start the Parallax Propeller Tool software on your PC to load and program the EEPROM:
    - ✓ Select Run -> Identify Hardware and confirm that the Propeller Chip is located.
    - ✓ Select File -> Open and locate and load the PropIO2.eeprom file
    - ✓ From the resulting Object Info dialog box, choose Load EEPROM
    - ✓ Wait for the programming operation to complete and make sure the verification was performed without error
  - Remove power from the PropIO V2.
  - Return the shorting plug at JP1 back to it's default setting of 1-2.
  - Remove the pigtail.



## Configuration

The following configuration settings are possible on the PropIO V2:

JP1	1-2: Default setting allowing use of the serial port for host controlled I/O (default) 2-3: Allows programming the Propeller EEPROM via the serial port
JP2	1-2: ECB bus /RESET follows traditional N8VEM standard (default) 2-3: ECB bus /RESET conforms to Kontron standard
P6	Short pin pairs to control port I/O address that PropIO V2 will respond to. Note that you <b>omit</b> a jumper to set the corresponding bit on. The hex numbers next to each pair of pins indicates the value of the corresponding bit. The default port is 0x40, so you would short all pairs of pins <b>except</b> the pair marked '40' to achieve the default port address of 0x40.
SW1	Momentary contact pushbutton switch that allows you to reset the onboard Propeller without resetting other devices connected via the ECB bus.

## Connectors

The following connectors exist on the PropIO V2:

P1	ECB bus connector. Signals follow N8VEM standards.
P2	Serial port connector. Pinout is N8VEM standard (also known on PC's as the DTK standard) and the appropriate pigtail is required. Port operates as DTE, so a null modem cable is required to connect to a standard PC serial port.
P3	Mini-SD card slot
P4	PS/2 Keyboard connector
P5	DB-15 VGA Monitor connector
P7	PropPlug connector
P8	Optional power connector jack for standalone operation. Use +5V regulated power with positive tip on connector. Do not apply power to this connector when the ECB bus is powered.

## Testing

As documented in [Building the N8VEM for the Total Beginner](#) you should perform a few simple tests before attempting to any functional testing, such as:

- Prior to installing any IC's, measure resistance across +5V and GND to ensure there is no short between the power connections.
- Also prior to installing any IC's, apply power to the board and check all sockets for power. The +5V and +3V3 LEDs should light even with no ICs installed.
- Install all IC's except for the Propeller, apply power, and make sure none of the ICs get hot.
- Install the Propeller, apply power, and make sure it does not get hot.

Following construction, it is useful to test the board by itself first. This allows you to ensure that all of the peripheral connections are working and that the EEPROM is properly programmed before trying to utilize the board in your N8VEM system.

To test the board independently, you should connect a known working VGA monitor and PS/2 keyboard. Apply power to the board by **either** connecting +5V to the power connector jack or by inserting it into a powered ECB bus.

When power is applied, you should observe the following things happening within 5 seconds:

1. The speaker should beep
2. The keyboard LED's should flash once
3. You should see text like the following displayed on the VGA monitor:

```
Initializing PropIO... OK
Initializing Keyboard... OK
Starting Timer... OK
Starting PortIO cog... OK
PropoIO Ready!
```

You should also look at the bottom line of the display and confirm that in the middle it reads “N8VEM PropIO V2 | RomWBW” and on the right it should show a version number of 0.94 or greater. Make sure that you see the “V2” after PropIO on the status line.

Once you have successfully completed independent testing of the board, you can connect it to an ECB bus along with your known working N8VEM SBC. For initial operation and testing, short jumper JP2 **on the SBC** to force initial console I/O to continue to go to your SBC's serial port as it always has.

## Usage

You will need to reprogram the PROM on your SBC with “N8VEM\_propio.rom” from the RomWBW distribution. Although it may be desirable to build a custom version of RomWBW for your system eventually, I strongly recommend using the pre-built ROM for initial testing. Install the new PROM in your SBC and start your SBC with the PropIO V2 installed on the ECB bus. At startup, you should see “PRP:” as one of the hardware discovery lines. At present, it is normal for this line to have nothing after the colon.

After pressing ‘R’ to perform a Boot from ROM operation, you will be at the CP/M prompt. You can use the command “B: PIP CRT:=CON:” to cause keys you type to be sent to the VGA monitor. You can use the command “B: PIP CON:=CRT:” to cause keys typed on the PS/2 keyboard to go to the CP/M console. In either case, type ^Z to return to normal operation on the CP/M console. Finally, you can type “B: STAT CON:=CRT:” to switch the CP/M console completely over to the PS/2 Keyboard and VGA Monitor. You can switch back by using the command “B: STAT CON:=TTY:”. If you want to have your system boot with the PS/2 Keyboard and VGA Monitor as the initial CP/M console, just remove the JP2 shorting plug on your SBC.

The RomWBW firmware for the PropIO will also allow you to access the SD Card for storage. For the pre-built version of RomWBW, this will be drive letters C:-F:. If the SD Card is not already formatted for N8VEM operation, you will need to switch to the C: drive and then enter the command “B:MULTIFMT” to format the logical units on the drive. You should refer to the RomWBW documentation for more information on using logical drives.

## References

- [ECB PropIO V2 Wiki Folder](#)
- [Parallax Propeller Manual and Datasheet](#)
- [Parallax Propeller Tool Software](#)
- [RomWBW Firmware](#)
- [N8VEM Project Wiki](#)