599 Menlo Drive, Suite 100
Rocklin, California 95765, USA
Office: (916) 624-8333
Fax: (916) 624-8003

General: info@parallax.com
Sales:sales@parallax.com
Technical: support@parallax.com
Web Site: www.parallax.com

PropScope Firmware Specifications

Synopsis

The PropScope USB Oscilloscope is controlled by a Propeller microcontroller, clocked at 100 MHz, using a 6.25 MHz crystal and the 16x multiplier. An FT232RL USB to serial converter translates serial communications over the USB port, and a AT24C512~~24WC128~~ serial EEPROM has 32 KB for program storage and 32 KB for calibration and data logging. The Propeller is connected to an LTC2286 two-channel, 10-bit ADC with a top sample rate of 25 Msps. The input divider and ADC range can be set to measure 20, 10, ~~4, and 2~~2, and 1 volt peak to peak readings with AC or DC coupling. The LTC2286 is connected to the Propeller through two 10-bit data buses, which can be read simultaneously, or multiplexed onto a single bus. A PCA9538 I2C I/O expander controls AC/DC coupling, input division, and ADC input range independently for each channel. The I/O expander also controls a -2 V supply rail for the analog front end, and switches the ADC between single and dual data buses. The expansion port accesses all power rails, the Propeller's reset pin, six dedicated Propeller I/O pins, 10 Propeller I/O pins that are shared with the the ADC, and a pin signifying the state of the shared I/O pins.

Propeller I/O connections

- **P0** through **P9** connect directly to the primary data bus on the ADC. These pins should always be set as inputs. When **P27** is driven high, **P0** through **P9** will contain the value of the last sample on channel 1; when **P27** is driven low, **P0** through **P9** will contain the value of the last sample on channel 2. When **P27** is floating, the pins will be in an undetermined state. Samples are taken on the falling edge of the 'clock' net.
- **P10** through **P19** connect to the secondary data bus on the ADC, as well as the expansion port. When the 'card' net is driven high, the ADC will leave the bus in a high-impedance state, otherwise it will drive the bus. When the ADC is driving **P10** through **P19**, and the 'mux' net is high, **P10** through **P19** contain the value of the last sample on channel 2; if the 'mux' net is low, **P10** through **P19** contain the value of the last sample on channel 1. When the 'mux' net is floating, the bus will be in an undetermined state. Samples are taken on the falling edge of P26. Before setting any of **P10** through **P19** on the Propeller as outputs, first pull the 'card' net high.
- **P20** through **P25** connect directly to the expansion port. Their function is dictated by the expansion card.
- **P26**, connected to the 'clock' net, clocks the ADC. Samples are taken on the falling edge of the clock pulse. When using a counter module to generate the clock, use NCO mode to ensure that the clock pulses stay aligned with the instruction timing.
- **P27**, connected to the 'mux' net, multiplexes channel 1 and channel 2 onto **P0** through **P9** and **P10** through **P19**. When 'mux 'is high, **P0** through **P9** contain the value of the last sample on channel 1 and **P10** through **P19** contain the value of the last sample on channel 2; when mux is low, P0 through P9 contain the value of the last sample on channel 2 and and **P10** through **P19** contain the value of the last sample on channel 1. When the 'mux' net is floating, the **P0** through **P19** are in an undetermined state.
- **P28** and **P29** connect to the SCL and SDA nets, respectively, of the I2C bus. There is an AT24C512 EEPROM at address **$50** and a PCA9538 I/O expander at address **$70**. The I2C bus is also connected to the expansion port.

• Pins **P30** and **P31**, respectively transmit and receive serial data over the USB port, using an FT232RL USB to serial converter. The reset pin also also coupled to DTR, for programming over USB.

PCA9538 I/O expander functions

I/O pins **0** through **3** are connected to relays. To reduce inrush currents, avoid simultaneous low to high transitions. I/O pins **0** through **3** should only be transitioned low to high one at a time, with at least 50 ms between each transition.

PCA9538 Address 112 (decimal), 70 (hexadecimal) [Times 2 as direction is bit 0]
Address,command,value for write operation

Commands:
00 Read input port
01 Write output port [default 11111111]
02 Reverse polarity of inputs (1 = invert) [default 00000000]
03 Set direction (0 = output, 1 = input) [default 11111111]

• I/O pins **0** and **1** select the coupling for channels 1 and 2, respectively. Both are externally pulled low. Let the pin float or pull it low for ~~AC~~ DC coupling; drive the pin high for ~~DC~~ AC coupling.

• I/O pins **2** and **3** select the input division ratio for channels 1 and 2, respectively. Both are externally pulled low. Let the pin float or pull it low to divide the channel's input by ~~ten~~twenty; drive the pin high to divide the channel's input by two.

• I/O pins **4** and **5** select the ADC input range for channels 1 and 2, respectively. Drive the pin low to read 2 volts peak-to-peak; drive it high to read 1 volt peak-to-peak. When floating, the channel will be in an undetermined state.

• I/O pin **6** sets the state of the -2 V supply, used by the analog input circuitry. The pin is externally pulled low. When floating or pulled low, the -2 V supply will be disabled; when pulled high, the -2 V supply will be enabled.

• I/O pin **7**, connected to the net 'card', sets the state of the secondary data bus on the ADC. The pin is externally pulled low. When not driven, the bus is enabled and the ADC will drive Propeller pins **P0** through **P19**; when pulled high, the ADC will only drive **P0** through **P9**, leaving **P10** through **P19** available for expansion port communications. This pin should never be driven low, and should be driven high while any of the Propeller pins from **P10** through **P19** are set as outputs.

For more information about controlling the PCA9538, refer to the data sheet, available from NXP's web site, http://www.nxp.com

Reading from the LTC2286 ADC

To ready the ADC, first enable the -2 V supply by driving I/O pin 6 of the PCA9538 high. Then clock the ADC by setting **P26** and **P27** to outputs, with **P26** low, and set **frqa** to the result of the following equation, with a maximum frequency of 25 MHz, or a maximum **frqa** value of **$40_00_00_00**:

$$\texttt{frqa} = 2^{32} \times \frac{frequency}{clkfreq}$$

To read both channels from the ADC simultaneously, using both data buses, let the 'card' net float or drive it low, drive **P27** high, and set **ctra** to **$20_00_00_1A** Read **P0** through **P19**, after the

Formatted: Indent: First line: 0 cm

Formatted: Indent: First line: 0 cm

falling edge of each clock pulse. **P0** through **P9** will read Channel 1, and **P10** through **P19** will read channel 2.

To read a single channel from the ADC, using only the primary data bus, let the 'card' net float or drive it low, and set **ctra** to **$20_00_00_1A** Read **P0** through **P19**, using ina, after the falling edge of each clock pulse. If set **P27** is high, **P0** through **P9** will read channel 1; if **P27** is low, **P0** through **P9** will read channel. **P10** through **P19** will be available for access on the expansion port.

To read each channel from the ADC alternately, using only the primary data bus, drive the 'card' net high, set **P27** high, set **ctra** to **$24_00_1B_1A**. Read **P0** through **P9**, using **ina**, after the falling edge of each clock pulse for channel 2, and after the rising edge of each clock pulse for channel 1. Both channels will be sampled on the falling edge of the clock pulse, but the first channel's result will be stored until the rising edge of the clock pulse. **P10** through **P19** will be available for access on the expansion port.

When using multiple cogs to sample the ADC, keep in mind that hub access instructions will align each cogs **ina** sample period with the hub access window, and each consecutive cog has a hub access window two system clock cycles after the previous cog. Each instruction requires 4 system clock cycles, so the second cog would be able to sample 2 system clock cycles after the first, plus 4 clock cycles for each instruction executed in between, resulting in a 2, 6, 10, etc. system clock cycle gap. If sampling at 25 Msps, each sample would have to take place 4 system clock cycles after the previous, but this is not possible with adjoining cogs, so the sampling cogs need to either be interleaved throughout the Propeller, e.g. cogs 1, 3, 5, and 7, or the sampling cogs need to avoid hub instructions during sampling and align themselves using **waitcnt** instructions immediately before sampling.

ADC resolution

5V	Relay off / 20	2 Volt range	+/- 10 volts	ADC step 19.5 mV
2V	Relay off / 20	1 Volt range	+/- 5 volts	ADC step 9.8 mV
1V	Relay off / 20	1 Volt range	+/- 5 volts	ADC step 9.8 mV
0.5V	Relay on / 2	2 Volt range	+/- 1 volts	ADC step 2.0 mV
0.2V	Relay on / 2	2 Volt range	+/- 1 volts	ADC step 2.0 mV
100mV	Relay on / 2	1 Volt range	+/- .5 volts	ADC step 1.0 mV

Formatted: Font: (Default) Courier New, 8 pt

For more information about the LTC2286 ADC, refer to the data sheet, available from Linear Technology's web site, http://www.linear.com,

The internal EEPROM

- Addresses **$0000** through **$7FFF** are for program and data storage. When the Propeller programs the EEPROM through the boot loader, it will automatically overwrite **$0000** through **$7FFF**. Any space not used for program storage can be used for data storage, but will be lost when the Propeller is reprogrammed. Under most circumstances, the EEPROM will not contain a program in EEPROM, but will receive a program from the PC connected to the PropScope. When an application or expansion card requires a program on the EEPROM, the program should not transmit data on **P30** until it has received data on **P31**.
- Addresses **$8000** through **$FF7E** can be used for any type of data storage and will persist when the Propeller boot loader programs the EEPROM.
- Addresses **$FF7F** through **$FF8E** contain an identifier string, currently **"PropScope Rev A "**, or **$50726F7053636F70652052657620412 0** in hexadecimal.
- Address **$FF8F** and **$FF90** contain the hardware version in a two-byte word, least significant byte first.
- Address **$FF91** and **$FF92** contain the test version in a two-byte word, least significant byte first.

• Address $FF93 through $FF5F FFEE contain four-byte calibration words, least significant bit first, following the contents outlined in table 1.

Table 1: EEPROM calibration data

Address	Contents
$FF93	Ideal value for positive scale with 1/10 20 divider and 1 Vp-p ADC range
$FF97	Actual reading for channel 1 2 positive scale with 1/10 20 divider and 1 Vp-p ADC range
$FF9B	Actual reading for channel 2 1 positive scale with 1/10 20 divider and 1 Vp-p ADC range
$FF9F	Ideal value for negative scale with 1/10 20 divider and 1 Vp-p ADC range
$FF13FFA3	Actual reading for channel 1 2 negative scale with 1/10 20 divider and 1 Vp-p ADC range
$FF17FFA7	Actual reading for channel 2 1 negative scale with 1/10 20 divider and 1 Vp-p ADC range
$FF1BFFAB	Ideal value for positive scale with 1/10 20 divider and 2 Vp-p ADC range
$FF1FFFAF	Actual reading for channel 1 2 positive scale with 1/10 20 divider and 2 Vp-p ADC range
$FF23FFB3	Actual reading for channel 2 1 positive scale with 1/10 20 divider and 2 Vp-p ADC range
$FF27FFB7	Ideal value for negative scale with 1/10 20 divider and 2 Vp-p ADC range
$FF2BFFBB	Actual reading for channel 1 2 negative scale with 1/10 20 divider and 2 Vp-p ADC range
$FF2FFFBF	Actual reading for channel 2 1 negative scale with 1/10 20 divider and 2 Vp-p ADC range
$FF33FFC3	Ideal value for positive scale with 1/10 2 divider and 2 Vp-p ADC range
$FF37FFC7	Actual reading for channel 1 2 positive scale with 1/2 divider and 2 Vp-p ADC range
$FF3BFFCB	Actual reading for channel 2 1 positive scale with 1/2 divider and 2 Vp-p ADC range
$FF3FFFCF	Ideal value for negative scale with 1/2 divider and 2 Vp-p ADC range
$FF43FFD3	Actual reading for channel 1 2 negative scale with 1/2 divider and 2 Vp-p ADC range
$FF47FFD7	Actual reading for channel 2 1 negative scale with 1/2 divider and 2 Vp-p ADC range
$FF4BFFDB	Ideal value for ground voltage for all ranges
$FF4FFFDF	Actual channel 1 2 reading for ground voltage with 2 Vp-p ADC input range
$FF53FFE3	Actual channel 2 1 reading for ground voltage with 2 Vp-p ADC input range
$FF57FFE7	Actual channel 1 2 reading for ground voltage with 1 Vp-p ADC input range
$FF5BFFEB	Actual channel 2 1 reading for ground voltage with 1 Vp-p ADC input range

Formatted Table

The calibration information provides enough information to characterize all input dividers and ADC ranges for both channels.

Addresses $FF5F FFEF through $FFFF are reserved for future use.

Using the expansion port

The expansion port connects to the +5, +3, and -2 volt power rails, reset, ground, the I2C bus, I/O pin 7 from the PCA9538 I/O expander, and P10 through P25 on the Propeller. Observe care to prevent excess voltage or current consumption on any I/O pin. The +3 and -2 volt power rails are generated from the USB supplied +5 volt rail, and total power consumption, including consumption from the PropScope itself, should not exceed 500 mA on the +5 volt power rail. Pin

The reset pin is pulled high and active low. When the reset pin is pulled low, the PropScope will stop running, and when it is released it will reset. The reset pin is intended for debugging firmware and for advanced expansion card functions, especially when the expansion card controls the PropScope.

Propeller pins **P10** through **P19** are shared with the secondary data bus on the ADC. PCA9538 I/O pin **7** indications the state of **P10** through **P19**. The expansion card should only drive **P10** through **P19** when either it or the Propeller have driven PCA9538 I/O pin **7** high. When developing firmware to accompany an expansion card, do not let the Propeller and the expansion card simultaneously drive any of **P10** through **P19**

Propeller pins P20 through P25 are connected directly to the Propeller and can be used in any way by the expansion card. When developing firmware to accompany an expansion card, do not let the Propeller and the expansion card simultaneously drive any of P20 through P25.

The 'card' net controls the state of the secondary data bus on the ADC. This pin is externally pulled low, and should not be driven low. When low, the ADC will drive Propeller pins P10 through P19, when high, the ADC will not drive P10 through P19.

The I2C bus, Propeller pins P28 for SDA and P29 for SCL, is also available from the expansion connector. Addresses $50 and $70 are reserved for the internal EEPROM and I/O expander. All expansion cards should contain an I2C EPROM, of any size, at address $51. The first 14 bytes of the EEPROM should contain a string identifying the card, followed by a one-byte hardware version and a one-byte test version.

The PropScope DAC card

The PropScope DAC card has four functions, a universal 8-bit Digital to Analog Converter (DAC), a 4-bit DAC designed specifically for base-band video, a 4-bit Logic State Analyzer (LSA), and an analog trigger. The analog trigger is always functional, but the LSA and DACs are only functional when the secondary data bus on the ADC is disabled. Also, the LSA and 8-bit DAC cannot be used simultaneously. The following connections are valid when the PropScope DAC card is connected.

Propeller pin P10 switches between the 8-bit DAC and the 4-bit LSA. Set both the 'card' net and Propeller pin P10 high to enable the DAC. Set Propeller pin P10 low to enable the LSA.

Propeller pin P11 sets the range of the DAC. Set P11 high for a 0 to +5 volt range, and set P11 low for a -1.5 to +1.5 volt range.

Propeller pins P12 through P15 control the 4-bit base-band video DAC. These pins are arranged identical to the Propeller Demo Board. D15 is the most significant bit, D13 the least, and D15 is the aural sub-carrier pin. The 'card' net must be high for the video output to function.

Propeller pins P16 through P23 control the 8-bit DAC. When both the 'card' net and Propeller pin P10 are high, the DAC will output the value of P23 through P16, with P23 the most significant bit and P16 the least significant bit. A value of $00 represents the lowest possible output voltage, and $FF the highest. The voltage range is 0 to +5 volt when P11 is high, -1.5 to 1.5 volts when P11 is low. The DAC output is enabled when both the 'card' net and Propeller pin P10 are high.

Propeller pin P24 sets the voltage threshold for the analog trigger. The threshold scales linearly with the duty-cycle, from -10 to +10 VDC. A high value sets the trigger threshold to +10 V, a low value sets the trigger threshold to -10 V, and 1 50% duty sets the trigger value sets the trigger threshold to 0 V.

Propeller pin P25 reflects the state of the trigger. When the trigger input is above the threshold set by P24, P25 is high, when it is below P25 is low.

The I2C bus contains a 24C02 EEPROM at address $51, currently containing the following ID "PS DAC Card A ", or $50532044414320436172642041 20 in hexadecimal.