

Table of Contents

Solar Pi Platter	3
Description	3
Features	3
Applications	3
Version Information	4
Disclaimer	4
Contact	4
Electrical Specifications	5
Mechanical Specifications	7
Description	8
Block Diagram	8
Circuit Description	8
Connections	9
Battery Connection	10
Low Impedance (USB) Charge/Power source	12
High Impedance (Solar) Charge source	13
USB Ports	14
Ethernet Port Expansion	14
IO Expansion Header	15
Getting Started	16
Load Software	18
Communicate using screen	18
Communicate using talkpp	18
Command Interface	21
Command List	21
Power Management	22
Real Time Clock	23
User Peripherals	23
Control and Status	24
System Configuration	26

Operation	28
Default Configuration	28
Low Battery Handling	29
Battery Restart Operation	29
USB Overcurrent	30
Watchdog Operation	30
Low and Critical Battery Threshold Calculations	30
Start Under-voltage and Restart Threshold Calculations	30

Solar Pi Platter

Description

The Solar Pi Platter is a versatile expansion board for the Raspberry Pi Zero™ computer that provides power from a single-cell Rechargeable Lithium-Ion battery, additional peripherals including analog inputs, PWM outputs, USB ports and optional hardwired ethernet. A real-time clock allows for scheduled power cycling. Dual charging sources support both low-impedance devices like common USB chargers and high-impedance devices like Solar Panels. The Solar Pi Platter allows the Pi Zero to be used in a wide variety of applications ranging from solar-powered remote data acquisition systems to battery-backed file servers.

Features

- Single-cell Lithium Ion battery management system supplying up to 10W power at 5V
- Low impedance charging input with Un-interruptible Power Supply (UPS) functionality
- High impedance charging input for use with solar or inductive power sources
- Real Time Clock with Alarm Power Control
- Battery charged power-up control
- Automatic Low battery power-down control
- Three high-speed USB expansion ports with per-port Transaction Translator
- Power control for two USB expansion ports
- Expansion port for RJ45 Ethernet jack
- Two analog inputs with configurable reference
- Two PWM outputs with configurable period and support for Servo mode
- Simple command interface via a USB serial port
- Programmable power-on default operation
- User-accessible EEPROM configuration settings
- Watchdog Timer

Applications

- File server or router with built-in battery backup
- Solar powered web cam
- Remote control and sense applications
- Embedded system control with battery backup
- Portable computing applications

- Small robot control

Version Information

Datasheet Revision	Firmware Version	Comments
1.0	1.0	Initial Release
1.1	1.0	Fixed incorrect EEPROM max write specification.

Disclaimer

Copyright © danjuliodesigns, LLC, 2016, All rights reserved.

Neither the whole nor any part of the information contained in, or the product described in this manual, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder.

This product and its documentation are supplied on an as-is basis and no warranty as to their suitability for any particular purpose is either made or implied.

danjuliodesigns, LLC. will not accept any claim for damages howsoever arising as a result of use or failure of this product. Your statutory rights are not affected.

This document and the functionality of the product may be subject to change without notice.

Contact

Email: info@danjuliodesigns.com

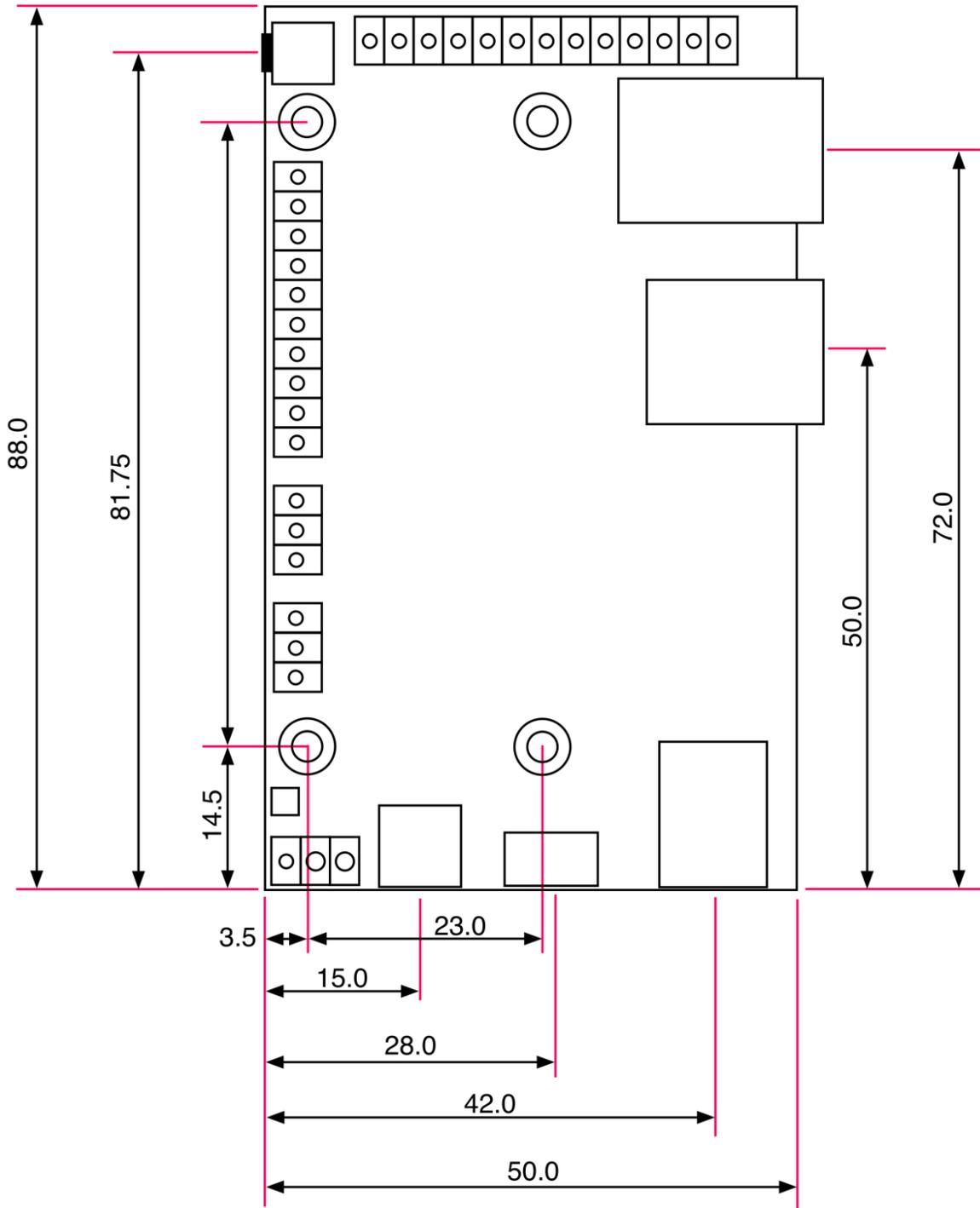
Website: http://www.danjuliodesigns.com/products/pi_platter/pi_platter.html

Electrical Specifications

Parameter	Min	Typ	Max	Unit	Conditions/Notes
Battery Supply Range	3.0		4.2	V	Single-cell Rechargeable Lithium Ion
Designed-for Battery Capacity	1000	2600	6000	mA/hr	Idea range for max charge current
Battery Charge Current			750	mA	
Battery Discharge Current			4	A	Maximum 5V load at low Vbatt
USB Power Input Voltage	4.75		5.5	V	USB voltages lower than Min will reduce charge current and may switch power source to battery
USB Power Input Current	1	2	2.75	A	USB Supply should be capable of power application load at 5V plus 750 mA Battery Charge current
Solar Power Input Operating Voltage Range	4.2	6	7	V	Designed for 6V nominal (7V open-circuit) solar panels
Solar Power Input Maximum Voltage			20	V	
Solar Power Input Current	0		750	mA	
Maximum 5V line current			2	A	Includes power taken by Solar Pi Platter, Raspberry Pi, any USB peripherals, external devices such as servos and Pi HATs
USB Ports 1 and 2 Current			1	A	Current Limited
Basic Solar Pi Platter with Raspberry Pi Zero current		320		mA	Current on 5V line for Solar Pi Platter and running Raspberry Pi Zero with USB Peripherals or Ethernet
Ethernet current		75		mA	Additional current on 5V line for an active ethernet connection
RTC current		20		uA	Average current consumption from battery with the RTC running and everything else powered down
Maximum 3.3V connection output current			100	mA	
Maximum VD connection output current			25	mA	
Maximum VB connection output current			500	mA	
PWM Digital Output Low Level			0.6	V	

Parameter	Min	Typ	Max	Unit	Conditions/Notes
PWM Digital Output High Level	VD - 0.7			V	
Maximum output current sunk by PWM signal			10	mA	
Maximum output current sourced by PWM signal			10	mA	
Analog input voltage range	0		VD	V	Exceeding this range will damage the PMIC/RTC
EEPROM write cycles	10K	100K		Write Cycles	$T_A \leq 85^\circ\text{C}$
EEPROM characteristic retention	40			Year	Provided no other specifications are violated
Operating Temperature Range	0	20	60	$^\circ\text{C}$	
Storage Temperature Range	-30	20	70	$^\circ\text{C}$	

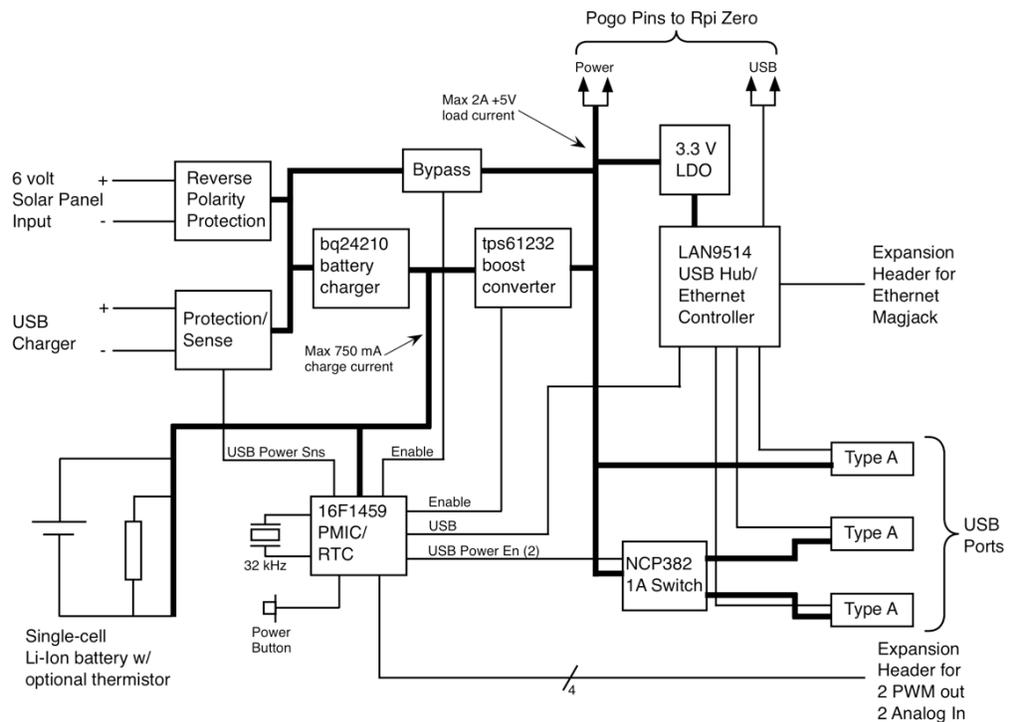
Mechanical Specifications



Board Dimensions (mm)

Description

Block Diagram



Solar Pi Platter Block Diagram

Circuit Description

The Solar Pi Platter is both a power controller and expansion peripheral for the Raspberry Pi Zero. It is comprised of two major subsystems controlled by a micro-controller (PMIC/RTC). The first subsystem is a power-path that manages charging the battery from one of two sources and providing +5 volts to the Pi and peripherals. It utilizes a TI bq24210 charge controller to provide up to 750 mA of charge current for the battery using a three-phase charging algorithm (conditioning, constant current and constant voltage). The charge controller dynamically adjusts charge current to prevent collapse of input voltages making it a good fit for high-impedance power sources like solar panels. A TI tps62232 boost converter converts the battery voltage to 5V at up to 2 A (10W) for the system. The boost converter is disabled and a bypass circuit activated to provide 5V directly from the low impedance USB power source. The PMIC manages switches between USB power and battery power turning the Solar Pi Platter into a Uninterruptible Power Supply when used with a USB power source. A NCP382 USB power switch provides the ability to control power on two

USB ports as well as limit their current to 1A. Power is delivered to the Raspberry Pi Zero using a pair of pogo pins that contact test points on the bottom of the Pi PCB.

The second subsystem provides a 4-port USB Hub for the Pi Zero's single USB host port. One port is used internally by the Solar Pi Platter to allow communication with the PMIC/RTC. The other three ports are available for the user. Two of these ports may have their power turned on or off by software running on the Pi as a way to manage power in a battery-powered application. The USB Hub is based on the same LAN9514 as found on higher-end Raspberry Pi models. It is a high-performance hub with multiple Transaction Translators to ensure that USB peripherals with different data rates can operate simultaneously. The Solar Pi Platter also brings out the ethernet expansion signals from the LAN9514 to an expansion port that matches a breakout board ([Sparkfun BOB-13021](#) RJ45 Magjack breakout board) for a 10/100 Mbit/sec wired network connection. Connection is made with the Pi's host USB port via a pair of pogo pins that contact test points on the bottom of the Pi PCB.

The PMIC/RTC micro-controller implements multiple functions. It is a real-time clock that runs as long as the battery is connected. An alarm with optional repeat can be used to power-up the Pi at a preset time. It also monitors the battery voltage and can provide warnings to software running on the Pi and automatically shutdown the system when the battery reaches a specified low-level. The PMIC can also power-up the Pi when the battery is charged to a specified voltage level. It provides two analog inputs to allow the Pi to measure voltages and two versatile PWM outputs supporting both traditional pulse-width modulation plus a mode designed to control hobby servos (e.g. for pan/tilt functionality). A power control input is made available to the user as well as connected to a push-button to allow manual control of power. The PMIC also has a watch-dog timer function that can be enabled which cuts power to the Pi for five seconds if software on the Pi fails to periodically update the timer.

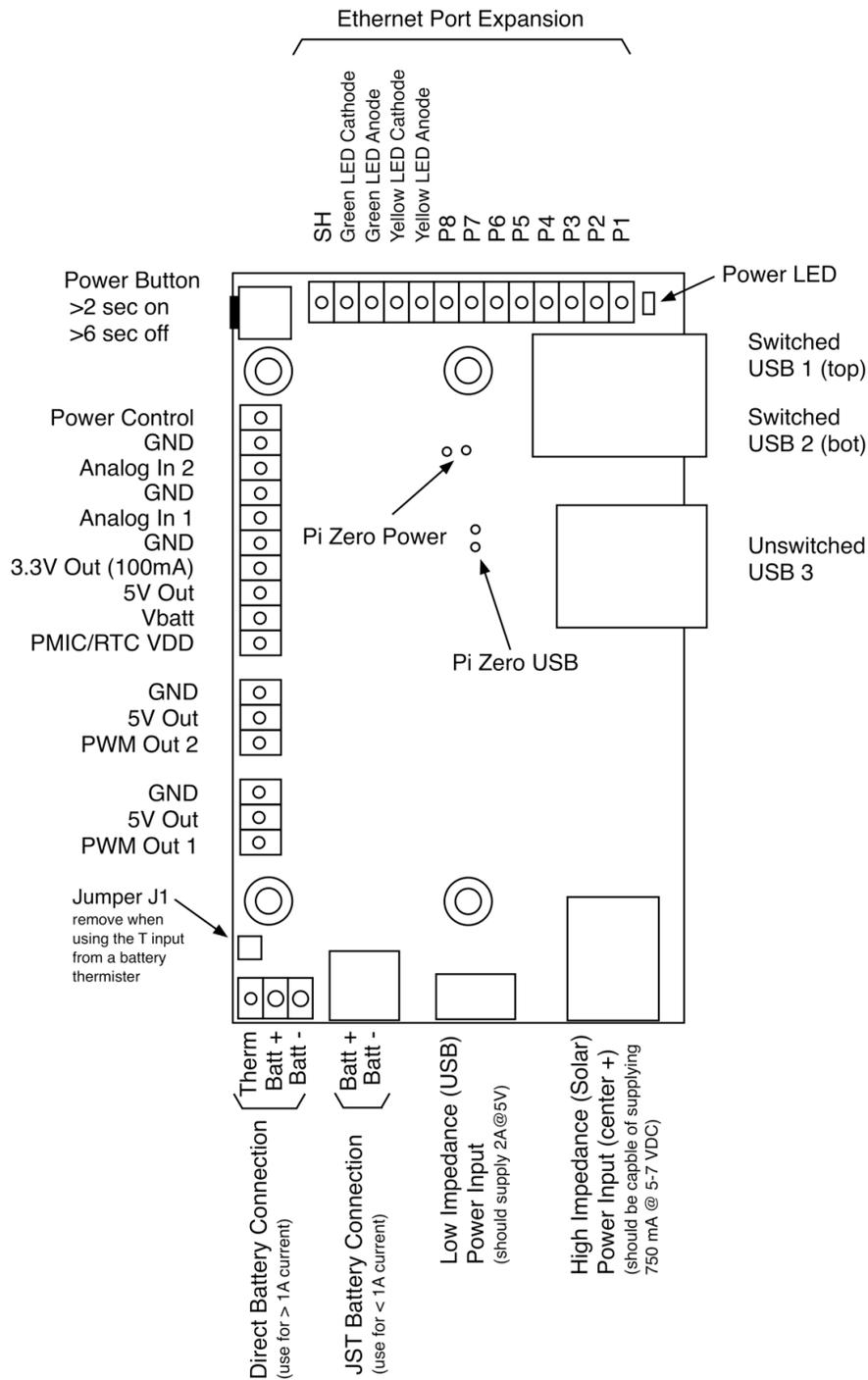
Connections

Connections to the Solar Pi Platter are made using a variety of standard connectors and with through-hole connectors on 0.1 inch (2.54 mm) centers or solder pads on the rear side of the circuit board. The core functionality may be accessed via the standard connectors. The through-hole connectors expose additional functionality and are designed to connect wires from the user's custom circuitry or an external circuit board containing an RJ45 Magjack for wired networking.

One jumper, described later, allows configuration with a Lithium Ion battery pack containing a built-in thermistor.

A button on the side of the Solar Pi Platter provides manual power control.

Note: Extreme care should be taken when soldering or de-soldering leads to the Solar Pi Platter solder pads or adding or removing a jumper. Too much heat or pressure may tear the pad or jumper location off the PCB. Use a fine-tip soldering iron and a minimum amount of solder.



Board Layout (Top View)

Battery Connection

The Solar Pi Platter provides two battery connections (only one should be used at a time). It is designed to charge and operate with single-cell 3.7volt Lithium Ion rechargeable batteries with capacities ranging from about 1000 mA to 6600

mA (e.g. 3 balanced 18650 cells in parallel). Cells smaller than this may be overcharged and not provide useful power for the system. Cells larger than this may not be able to be successfully charged with the Solar Pi Platter's 750 mA maximum charge current. Both battery connections are reverse-polarity protected.

A JST connector matches common batteries sold by Adafruit and Sparkfun making it easy to just plug in a battery. These batteries are ideal for systems that do not utilize a lot of peripherals besides the Pi and low-power USB devices. This is because the JST connector is typically specified for 1A current max although testing shows it can handle significantly more. Typical batteries with a compatible JST cable are shown below.



[*Adafruit Product ID 354*](#)



[*Sparkfun PRT-08483*](#)

The three terminal Direct Battery Connection is provided to directly solder the leads from the battery to the Solar Pi Platter to support higher currents (the Solar Pi Platter can require up to about 4 A from the battery when the battery is almost discharged and it is supplying the maximum power output).

The three terminal Direct Battery Connection also provides an optional terminal for a type 103AT NTC thermister integrated in the battery pack. This allows the Solar Pi Platter charge circuit to modify the charge current based on the temperature of the battery. By default this functionality is disabled and the Solar Pi Platter assumes the battery pack is at a temperature of between 10°C and 40°C.

To enable an external thermister the jumper J1 (above the 'T' terminal) must be removed by removing the solder bridge between the two terminals of the jumper. This may be done using a solder wick or by quickly dragging the tip between the terminals. The jumper must be in place if no thermister is used in order for charging to operate correctly. Only type 103AT NTC thermisters are supported.

!!!! Lithium Ion Battery Warning !!!!

Be very careful to never short the battery terminals. Lithium Ion batteries can deliver a lot of current which can be a fire hazard and they can damage themselves in a way that can cause them to burst into flame.

Used correctly they are an excellent power source but you should familiarize yourself with their operation and safe management of them. The Solar Pi Platter is designed to prevent over- and under-charging but it is still possible to damage a battery in a dangerous way.

Low Impedance (USB) Charge/Power source

The Low Impedance (USB) Charge/Power source connector is designed for a USB Micro B plug connected to a 5V regulated power supply such as a phone charger or a supply designed to work with a Raspberry Pi. The Low Impedance connector is reverse-polarity protected.

This supply provides power for both charging the battery and for the 5 Volts used by the Solar Pi Platter USB Hub, Raspberry Pi and all peripherals. It should be sized to supply the current required by the 5V devices as well as charge current for the battery. Typically well made 5 W units (1A output current) will operate a system consisting of just the Solar Pi Platter and Raspberry Pi Zero. Ten watt units (2 A output current) are necessary for systems with peripherals. A unit that can supply up to 2.75 A @ 5V is necessary for a fully loaded system. Special care must be taken that the USB cable can handle the current (a custom USB power cable may be necessary).

When the system is powered up and the voltage at the Low Impedance (USB) Charge/Power source input falls below about 4.6 volts, the Solar Pi Platter will automatically switch to the battery for power. The Low Impedance source may continue to supply some current for charging the battery down to about 4.5 volts.

Important Note about USB Chargers

Many inexpensive USB Chargers use fine gauge wire in their cables that have too high of resistance when supplying currents in excess of 500 mA - 1A. This manifests itself in a large voltage drop between the charger and the Solar Pi Platter which may cause it to fail to charge the battery to assume that the USB power has failed and only power the system from the battery. Also many inexpensive chargers cannot deliver their rated current and will either shut-down or have an output voltage significantly lower than required by the Solar Pi Platter.

It is best to use a known-brand USB charger, such as those designed to provide 2 A of charging current for Apple products or a PSU with known good output at high loads and high quality USB cabling (do not use long cabling).

High Impedance (Solar) Charge source

The High Impedance (Solar) Charge source connector is designed for a 4.5 - 7 volt DC high-impedance power source (such as a solar panel). 5.5mm x 2.1mm plug. Center pin is positive. The High Impedance connector is reverse-polarity protected.

The High Impedance (Solar) Charge source is routed through the battery charge circuit and thus can only supply up to the maximum charge current to the system (750 mA). Any current required by a running system is subtracted from the current available to charge the battery. It is designed for systems that spend a significant amount of time in the powered-down state (so the battery can charge), only powering up to do some work and communicate or store results and then powering down again.

The High Impedance (Solar) Charge source can handle input voltage up to 20 volts (to account for unloaded voltage levels of high impedance sources) but only operates with voltage levels above the current battery voltage and below 7 volts. It was designed primarily for 6 volt solar panels (that usually have a maximum, open-circuit, voltage of 7 volts) but can also be used with sources, such as inductive chargers, that put out 5 volts.

Sizing solar panels is an art that depends on the application load requirements, the battery capacity and the amount of available sun. A full analysis is beyond this manual, however for many applications, 6- to 9-watt 6V solar panels are a good fit. The 9-watt panels give some extra headroom to handle light levels less than full sun. Basically the user needs to calculate the amount of time the panel can deliver charge (and the amount of charge which will be limited to the Solar Pi Platter's 750 mA maximum charge current), how much energy can be stored in their battery and how much power their application takes. They may need to optimize the application's power requirements (for example by keeping the system in a powered-down state most of the time or carefully controlling the time various power-hungry USB devices are powered up).

Compatible solar panels may be obtained from companies like Adafruit or Sparkfun.



[*Adafruit Product ID 2747*](#)



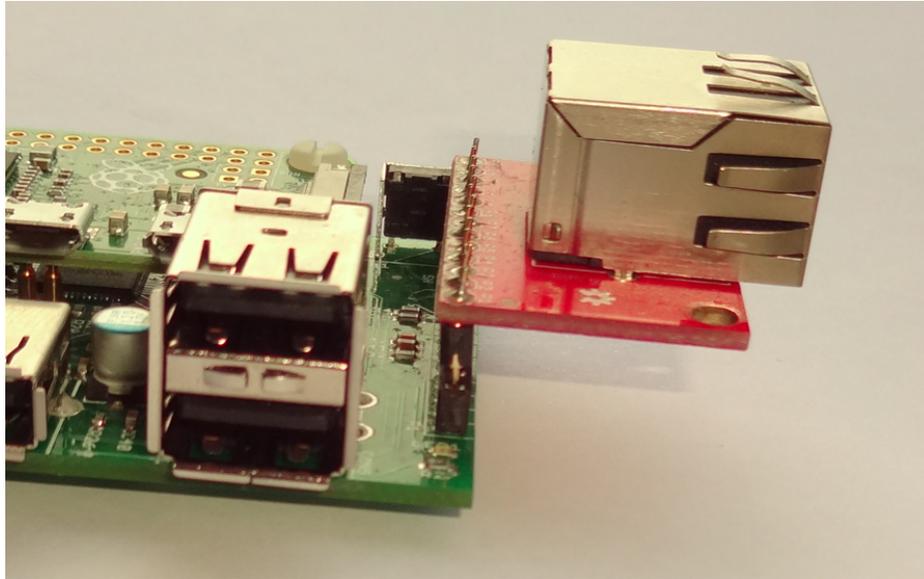
[*Sparkfun PRT-13784*](#)

USB Ports

The Solar Pi Platter provides three USB host ports. One port is powered whenever the system is powered up and is connected directly to the main 5V power bus. The other two ports' power is controlled by the Solar Pi Platter and may be switched on or off by software running on the Pi. Their default power may also be configured on or off when the Solar Pi Platter powers up. They are current limited to 1A and power is shut down when that limit is exceeded. A fault condition is also generated that may result in a message being communicated to the Pi via the Solar Pi Platter's serial link and a status bit being set that can be polled by software running on the Pi.

Ethernet Port Expansion

The Solar Pi Platter provides an expansion port designed to mate with the [Sparkfun BOB-13021 RJ45 MagJack Breakout](#) board. As shown below, the board is mounted component-side up and may be connected to the Solar Pi Platter in a variety of ways including a 13-pin machined pin socket and dual-pin header as shown below or directly by soldering a bare wire poked through both the MagJack Breakout board and the Solar Pi Platter. Discrete wiring should be avoided because of the high-frequency signaling that occurs with 100 MBit/sec ethernet.



Example MagJack Breakout Board mounting (removable)

IO Expansion Header

The Solar Pi Platter provides a set of expansion signals and auxiliary power outputs on a header that sits underneath the Raspberry Pi Zero. These are designed to be connected to wires interfacing to the user's circuits or connected to using a right-angle header assembly.

Power Control

The Power Control (PC) signal is an active-low control signal that is used to manually power-up or power-down system via the 5V power line. It is pulled high to the PMIC/RTC VCC power rail by a 10-kohm resistor. It is also connected to the power button. Pulling it to ground asserts the signal. Asserting the signal for more than 2 seconds when the Solar Pi Platter is powered down will cause the Solar Pi Platter to power-up (turn on) as long as there is power available on the Low Impedance (USB) Charge/Power source or the battery is above a low-voltage threshold (Battery Start Undervoltage-Threshold). Asserting the signal for more than 6 seconds when the Solar Pi Platter is powered up will cause it to power-down into sleep mode (turn off).

The PC signal is designed to be connected either to an external switch (or relay) or a low-impedance transistor switch (such as a N-channel MOSFET) with the other terminal connected to one of the Solar Pi Platter ground terminals.

It allows an external controller to manually power-up or power-down the Solar Pi Platter (and Pi). Various conditions (such as an Alarm on the Solar Pi Platter RTC or the battery voltage increasing beyond a threshold) can initiate power-up. A low battery condition or software running on the Pi can also initiate a power-down.

Analog Inputs

Two 10-bit analog inputs may be read by software running on the Pi. They are connected directly to inputs on the PMIC/RTC and their voltage range is 0 to VD (PMIC/RTC VCC). Exceeding these limits may damage the PMIC. Software can configure one of 4 independent voltage references for each analog input (1.024 volts, 2.048 volts, 4.096 volts and VD for ratio-metric readings). Signals connected to the analog inputs should have an impedance of 10 k-ohms or less. Add a small capacitor (e.g. 0.1 - 1 uF) for higher-impedance signals.

Auxiliary Power Outputs

The Solar Pi Platter makes several power sources available for external user circuitry on its expansion header.

- VB - Battery - limit current draw to 500 mA
- VD - PMIC/RTC VCC - limit current draw to 25 mA. This signal is available primarily for use as the high side for ratio-metric measurements on the analog inputs and as a reference for the PWM output high levels. It varies depending if the Solar Pi Platter is powered on or not. When the device is powered down then VD is about 0.15 volts less than VB. When the device is powered up then VD is about 4.75-4.8 volts.
- 5V - Power rail - available when the Solar Pi Platter is powered up. Current must be part of the 2A maximum limit.
- 3.3V - Power rail - available when the Solar Pi Platter is powered up. Current should be limited to about 100 mA. The 3.3V power is primarily used by the USB Hub.

PWM Outputs

Two 8-bit PWM outputs may be configured by software running on the Pi. The PWM outputs have two primary modes of operation: traditional PWM and Servo. In traditional PWM mode, they may be configured with one of three frequencies.

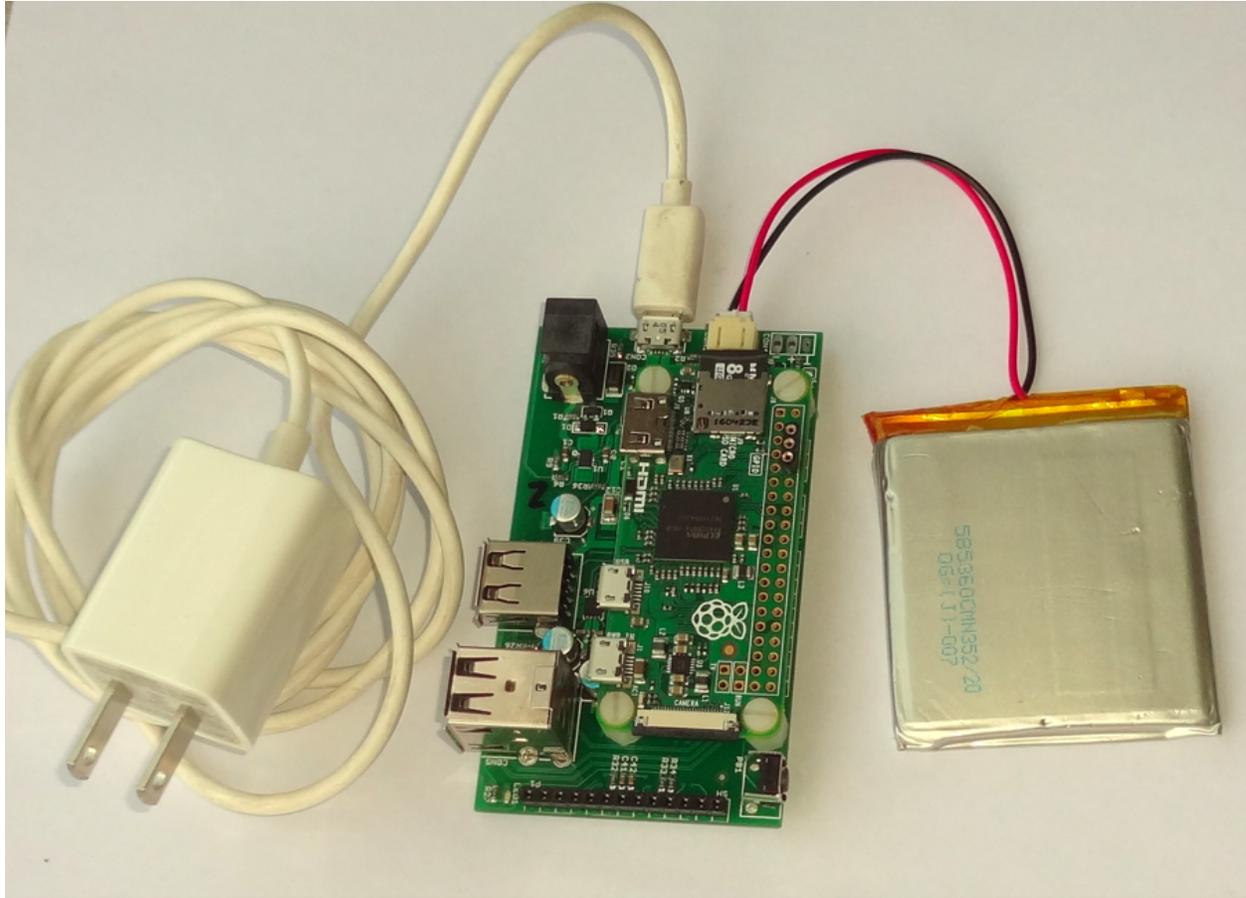
1. Slow (732 Hz) - Good for applications like LED dimming
2. Medium (2930 Hz)
3. Fast (46875 Hz) - Good for filtering to DC or for use with motor speed control

Servo Mode is designed for use with 3-wire hobby servos. It implements a 20 mSec (50 Hz) period with an active-high signal period between 0.8173 mSec and 2.1827 mSec with a center-point at 1.5 mSec. The PWM outputs are next to a pair of +5V and GND power terminals and designed to directly mate with most common servo cable pinouts (e.g Futaba).

Getting Started

This section assumes Raspbian has been loaded and configured on a Raspberry Pi Zero and some experience using Linux and a network connection.

Install the Raspberry Pi Zero on top of the Solar Pi Platter using the supplied nylon stand-offs and mounting hardware. Take care not to bend the pogo pins that make electrical contact between the two boards. Connect a battery and USB power source, as well as necessary peripherals such as WiFi USB module, keyboard/mouse and HDMI monitor.



Basic Solar Pi Platter Assembly

Plug in the USB power source and press and hold the Solar Pi Platter power button for more than two seconds. Green LEDs on the Solar Pi Platter and Raspberry Pi Zero should light indicating the board is powered up. The Pi should boot into Raspbian. The following instructions assume you have a shell prompt on the Raspberry Pi and the Pi has a network connection to the internet.

The Solar Pi Platter shows up as a USB serial device to the Pi. It is always the last USB serial device to enumerate on the USB Hub. If it is the only device then it shows up as `/dev/ttyACM0`. If there is another device connected to a USB connector (an Arduino Leonardo with USB Serial connection) then that device will show up as `/dev/ttyACM0` and the Solar Pi Platter will show up as `/dev/ttyACM1`.

There are many ways to communicate with the Solar Pi Platter.

Commands can be echoed directly to the device file. For example typing `echo -e "P1=190\r" > /dev/ttyACM0` sets PWM 1 to 190. We enable processing of backslash escapes so we can include a carriage return (<CR>) character to terminate the command.

A serial terminal emulator (for example the `screen` utility). The Solar Pi Platter interprets simple ASCII command sequences terminated with a <CR> character. It generates simple ASCII strings that can be displayed by the serial terminal emulator.

The talkpp utility allows sending commands to the Solar Pi Platter and configuring items such as the RTC. It automatically determines which serial device file is associated with the board. Instructions for loading and using it are given below.

A complete description of the Solar Pi Platter command set can be found in [Command Interface](#).

Load Software

Load `screen` using the following commands.

```
sudo apt-get update
sudo apt-get install screen
```

Download the `talkpp` binary from the Solar Pi Platter git repository: https://github.com/danjulio/rocketblue-automation/tree/master/pi_platter/unix_applets/talkpp and install it in `/usr/local/bin`.

Communicate using screen

Establish a connection to the Solar Pi Platter.

```
screen /dev/ttyACM0
```

Commands are terminated with the Enter/Return key (<CR>). The Solar Pi Platter does not echo back commands as they are entered.

Type `B<CR>`. You should see a response showing the battery voltage like

```
B=4.10
```

Type `S<CR>`. You should see a response showing the board status

```
S=20
```

Type `O=30<CR>`. This initiates a 30 second shutdown timeout. The Solar Pi Platter will turn off power in 30 seconds. Immediately type `CTRL-A \` to exit `screen` and then `sudo shutdown now` to gracefully shut down Raspbian before power is removed from the Pi (to prevent potential file corruption when killing power to a running system).

Communicate using talkpp

The `talkpp` utility is a scriptable command line interface to the Solar Pi Platter. It provides both a command interface and a set of commands for managing the RTC. Type `'talkpp -h'` for a list of commands and their arguments.

```
pi@raspberrypi0:~ $ talkpp -h
talkpp: [-c <command string>]
        [-s] [-t] [-a <alarm timespec>] [-d <delta seconds>]
        [-u | -h]

-c <command string> : send the command string. Command strings without
a '=' character cause the utility to echo back a response.
-s : Set the device RTC with the current system clock.
-t : Get the time from the Device RTC and display it in a form useful to
pass to "date" to set the system clock (mmddHHMMccyy.ss)
-f : Get the time from the Device RTC and display it in a readable form.
-a <alarm timespec> : Set the Device wakeup value (does not enable the alarm).
<alarm timespec> is the alarm time in "date" format (mmddHHMMccyy.ss)
```

```
-d <delta seconds> : Set the Device wakeup to <delta seconds> past the
                    current Device RTC time value (does not enable the alarm)
-w : Display the wakeup value in a readable form.
-u, -h : Usage (and optional help)
Version 0.4
```

Command Interface

Command strings are sent using the 'talkpp -c <command string>' command. Responses are printed.

To get the battery voltage:

```
pi@raspberrypi0:~ $ talkpp -c B
4.13
```

To set a PWM output value and then get the set value:

```
pi@raspberrypi0:~ $ talkpp -c P2=80
pi@raspberrypi0:~ $ talkpp -c P2
80
```

Time Interface

The Time Interface commands are used to set the Solar Pi Platter RTC and may be use to set the Raspberry Pi RTC when it boots in situations where it does not have access to network time.

To set the Solar Pi Platter RTC from the Pi's RTC:

```
pi@raspberrypi0:~ $ talkpp -s
```

To get the time from the Solar Pi Platter RTC in a format that is useful for setting the Pi's RTC:

```
pi@raspberrypi0:~ $ talkpp -t
121715332016.19
```

To set the Pi's clock from the Solar Pi Platter RTC:

```
pi@raspberrypi0:~ $ sudo date $(talkpp -t)
```

To display the Solar Pi Platter RTC in a human-readable form:

```
pi@raspberrypi0:~ $ talkpp -f
Sat Dec 17 15:39:10 2016
```

To set and the display a wakeup alarm in the Solar Pi Platter for Jan 1, 2017 at 08:00 (note that the alarm must be enabled by setting the Wakeup Enable control register to 1):

```
pi@raspberrypi0:~ $ talkpp -a 010108002017.00
pi@raspberrypi0:~ $ talkpp -w
Sun Jan 1 08:00:00 2017
```

To set an alarm repeat interval of one day (86400 seconds):

danjuliodesigns LLC

```
pi@raspberrypi0:~ $ talkpp -d 86400
```

Command Interface

Command List

Commands are simple ASCII sequences with two forms terminated by a Carriage Return character (<CR>), shown below, sent to or received from the Solar Pi Platter's ACM Serial interface (using the "screen" utility, for example). The first form is used to set a value. The second form is used to query a value. Each command is a single upper-case ASCII character, <CMD>, that is selected to mnemonically represent its function. A command may be followed by an optional numeric index, <INDEX>, required when it applies to more than one item (for example, USB port, PWM channel or Analog input). Values, <VALUE>, to be set are numeric values.

1. Form 1 (Set): <CMD>[<INDEX>]=<VALUE><CR>
2. Form 2 (Query): <CMD>[<INDEX>]<CR>

All parameters, <INDEX> or <VALUE>, are base-10 numeric values. All other characters besides the <CMD> character, numbers "0"-"9", the "=" character and the Carriage Return character are ignored. Illegal or malformed commands cause the Pi Platter to respond with an error message with the form "ERR <N><CR><LF>" where <N> is a number identifying the type of error.

"ERR 0": Unknown Command (The command character, <CMD>, did not match any of the supported commands).

"ERR 1": Illegal Index value (The index value, <INDEX>, did not apply or was out of range)

"ERR 2": Command does not support Form 1 (Set). It is a Query-only item.

The Solar Pi Platter does not echo back commands. It only generates responses to Queries, warning messages or command error messages. You can enable local echo if you are using a terminal emulator program. The Pi Platter command processor looks at the last characters before a <CR> entered to process commands. This means you can correct commands by just typing what you really meant before hitting <CR>. The Pi Platter also appends a Line Feed, <LF>, character to messages it sends to make them easier to see in a terminal emulator. It ignores <LF> characters sent to it.

The Solar Pi Platter is also capable of generating warning messages for various conditions at any time. Warning messages have the form "WARN <N><CR><LF>" where <N> is a number identifying the type of warning. Warning messages may be suppressed.

"WARN 0": Low Battery Warning Message

"WARN 1": Critical Battery Warning Message

“WARN 2”: USB Fault Detected Message

Commands are grouped into several categories.

1. Power Management - Monitor battery voltage, Set/Query the USB Port power, initiate a power-down timeout.
2. Real Time Clock - Set/Query the RTC, Wakeup Time and Wakeup Repeat Time.
3. User Peripherals - Query Analog Voltage readings, set PWM output values.
4. Control and Status - Set/Query operational control registers, Query the system Status.
5. System Configuration - Set/Query configuration values that are stored in EEPROM and are used to configure operation each time the board is powered up.

Power Management

Battery Voltage - “B”

Supports the Query form only. Returns the current battery voltage in volts with 10 mV resolution (B=<VOLTS><CR><LF> - for example “B=4.05”).

```
B<CR>
```

Power Off - “O”

Supports both forms. Set initiates a power-down in the specified number of seconds. Query returns the number of seconds until power is turned off. A value of 0 means the power down timer is not operating. The time value may range from 1-255 seconds (30 seconds in the example below).

```
O=30<CR>
```

Returns the number of seconds until power down (O=<NUM><CR><LF>).

```
O<CR>
```

USB Port Power Enable - “U”

Supports both forms. Set allows turning on or off an individual, indexed, USB port. Query returns the current power status of the indexed USB port. The index value may range from 1 - 2 for USB Ports 2 and 3. The power control value is 0 to turn a port off and 1 to turn a port on.

Turn USB Port 2 on and Port 3 off.

```
U1=1<CR>  
U2=0<CR>
```

Returns the power control state for USB Port 2 (U1=<N><CR><LF> where <N> is 1 or 2).

U1<CR>

Real Time Clock

Current Time - “T”

Supports both forms. Set allows setting the current RTC second count (32-bit unsigned number). Query returns the current RTC second count.

T=1000<CR>

Returns the current RTC value (T=<SECONDS><CR><LF>).

T<CR>

Wakeup Time - “W”

Supports both forms. Set allows setting the Wakeup Time (32-bit unsigned number). Query returns the current Wakeup Time. Note that Control Register Wakeup Enable (C0) must be set to 1 to enable wakeup to occur when the current time matches the wakeup time and the device is powered down.

W=1100<CR>

Returns the current Wakeup Time (W=<SECONDS><CR><LF>).

W<CR>

Wakeup Repeat Time - “R”

Supports both forms. Set allows setting the Repeat Time (32-bit unsigned number). This is the value added to the Wakeup Time when the RTC second count equals the Wakeup Time and Wakeup Enable is set to 1 (it is added when the board is both sleeping and when it is powered up). Query returns the current Repeat Time. A value of 0 means do not modify the Wakeup Time. Control Register Wakeup Enable (C0) is not cleared if the Wakeup Repeat Time is non-zero and the Wakeup Time is updated whenever Current Time matches it.

R=86400<CR>

Returns the current Wakeup Repeat Time (R=<SECONDS><CR><LF>)

R<CR>

User Peripherals

Analog Input - “A”

Supports the Query form only. Returns a 10-bit unsigned value from the indexed Analog Input. The number is referenced to the current Analog Channel reference setting. For example, if the Analog Channel reference is set to 1.024 volts then a 1.0 volt signal would return $(1.0/1.024) * 1023 = 999$. The index, <INDEX>, may have the value 1 or 2.

A1<CR>

```
A2<CR>
```

PWM Output - “P”

Supports both forms. Set allows setting a PWM value (8-bit unsigned value, 0-255) to the indexed PWM Output. Query returns the current value for the indexed PWM Output. The index, <INDEX>, may have the value 1 or 2. For example, to set PWM 1 to 40.

```
P1=40<CR>
```

Return the 8-bit value from PWM 2

```
P2<CR>
```

Note that the PWM subsystem has two different modes of operation. Normal PWM Mode is a traditional PWM function where the time the PWM output is set to 1 is directly proportional to the value set for that channel. This is the type of PWM operation that an Arduino provides with the “analogWrite” function on a PWM pin. The Pi Platter supports three different periods for Normal PWM Mode in order to support different uses of a PWM output. Fast mode is useful for generating a voltage level (simple DAC) using a RC network. Slow mode is useful for dimming a LED.

1. Fast: 21.33 uSec/period (46875 Hz)
2. Medium: 341.33 uSec/period (2930 Hz)
3. Slow: 1.364 mSec/period (732 Hz)

Servo PWM Mode is a special mode designed to support hobby Servo motor devices where the period is 20 mSec but the signal on time varies around a nominal servo mid-point on-setting of 1.5 mSec. Although individual servo devices vary wildly, the common understanding is that the servo is controlled over its entire mechanical range with settings ranging from 1 mSec (fully rotated to one side) to 2 mSec (fully rotated to the other side). The Pi Platter maps its 8-bit servo values to servo on settings from 0.8173 mSec to 2.1827 mSec with the mid-point on-setting of 1.5 mSec being a value of 128. The additional range supports the variation in individual servo mechanisms. A value of 0 sets a servo-on setting of 0.8173 mSec and a value of 255 sets a servo on-setting of 2.1827 mSec. The Pi Platter sets a value of 128 (mid-point) when PWM Mode is selected. Since each servo motor may vary, the user must determine their own limits. Exceeding a limit may damage the servo and will consume large amounts of current possibly causing the boost converter to shut down.

Control and Status

The Solar Pi Platter is controlled by values in a set of user-accessible 8-bit control registers.

Control Register - “C”

Supports both forms. Set allows setting an indexed control register to a value. Query returns the current value of an indexed control register. The index value selects a control register. The legal control register index values are shown below along with the legal values for each control register. For example, to set Control Register 0 to 1.

```
C0=1<CR>
```

Returns the value of Control Register 1.

C1<CR>

The Solar Pi Platter has the following control register index values. Default values may be set from EEPROM.

0: Wakeup Enable: 1 = Wakeup Enable, 0 = Wakeup Disable. Cleared automatically when a wakeup occurs unless the Repeat Time is non-zero.

1: Low Battery Warning Enable: 1 = Enable Message, 0 = Disable message

2: Critical Battery Warning Enable: 1 = Enable Message, 0 = Disable message

3: USB Fault Warning Enable: 1 = Enable Message, 0 = Disable message

4: Analog 1 Reference: (see below for Analog reference values)

5: Analog 2 Reference: (see below for Analog reference values)

6: PWM Mode: (see below for PWM Modes. Note the PWM Mode applies to both PWM channels)

7: Restart Enable: 1 = Enable restart when battery voltage exceeds Battery Restart Threshold, 0 = Disable restart

8: Watchdog Control: Set to a non-zero 16-bit second count (1-65535) to enable watchdog reset, 0 to disable

Analog Reference Values

0: Analog Channel Vref = PMIC/RTC VCC (PMIC/RTC power input - voltage on expansion port VD connection)

1: Analog Channel Vref = 1.024 Volts

2: Analog Channel Vref = 2.048 Volts

3: Analog Channel Vref = 4.096 Volts

PWM Modes

0: Normal PWM - Fast Period

1: Normal PWM - Medium Period

2: Normal PWM - Slow Period

3: Servo PWM (Initial servo value set to mid-point: 128)

Status - "S"

Supports the Query form only. Returns an 8-bit mask containing various status information.

S<CR>

The return value is a number 0 - 255 but can be broken down by converting to a hex value and looking at the following bit positions.

bit 7: USB Fault Detected: set when the USB Power Switch is indicating a USB Fault (over-current or short).

bit 6: Reserved, will return 0.

bit 5:4: Power Up Reason: (see below).

bit 3: Battery Charging: set if the Charger subsystem is supplying current to charge the battery and/or operate the system

bit 2: USB Supplying Power: set when power is being supplied by an external USB power supply, clear when it is being supplied by the battery.

bit 1: Battery Critical: set when the battery voltage is below the Battery Critical threshold, clear when the battery voltage is above the Battery Critical threshold.

bit 0: Battery Low: set when the battery voltage is below the Battery Low threshold, clear when the battery voltage is above the Battery Low threshold.

Power Up Reason (2-bits)

00: Alarm

01: Button

10: Battery Restart (battery charge exceeded Battery Restart Threshold causing power-up)

11: Watchdog Restart (Power was cycled for 5 seconds because of a watchdog timeout)

Version - "V"

Supports the Query form only. Returns the current board type and firmware revision as a 24-bit unsigned value (3 bytes). The lower byte (bits 7:0) contains the firmware minor number. The middle byte (bits 15:8) contains the firmware major number. The minor number is incremented for bug fixes and minor modifications to existing functionality. The major number is incremented for new functionality or user-visible modification of existing functionality. The minor number is the fractional value reported by the PMIC/RTC USB Version field. The major number is the integer value reported by the PMIC/RTC USB Version field. The upper byte (bits 24:16) contain the board ID (which is 0x01 for the Pi Platter). Version 1.0 returns "V=65792"

V<CR>

System Configuration

The Solar Pi Platter is initially configured by values stored in a non-volatile EEPROM memory. These values persist across applications of battery power.

EEPROM - "E"

Supports both forms. Set allows setting an indexed Configuration Value that is stored in the PMIC/RTC's EEPROM memory that persists across power-cycles (including battery removal). Query returns the value of an indexed Configuration Value. Configuration Values are read each time the board is powered-up and are used to set the default operating conditions. Setting a Configuration Value does not take effect until the Pi Platter has powered down and then powered up the Pi Zero. For example, to set Configuration Index 9 (USB 1 Default Enable) to 0 (off).

E9=0<CR>

Returns the value of Index 14 (PWM Mode)

E14<CR>

The Configuration Value indexes are shown below along with their legal values.

- 0: Battery Low Warn Threshold. A 16-bit unsigned integer (0-1023) that sets the [Low Battery Threshold](#).
- 1: Battery Critical Threshold. A 16-bit unsigned integer (0-1023) that sets the [Critical Battery Threshold](#).
- 2: Battery Start Under-voltage Threshold. A 16-bit unsigned integer (0-1023) that sets the [Under-voltage Threshold](#).
- 3: Battery Restart Threshold. A 16-bit unsigned integer (0-1023) that sets the [Restart-voltage Threshold](#).
- 4: Battery Critical to Turn-off Timeout. An 8-bit unsigned integer that sets the number of seconds (0-255).
- 5: Battery Warn Message Enable. Set to 0 to disable Low Battery Warning Messages, 1 to enable.
- 6: Battery Critical Message Enable. Set to 0 to disable Low Battery Warning Messages, 1 to enable.
- 7: Battery Restart Enable. Set to 0 to disable setting the Battery Restart Enable when powering up, 1 to enable.
- 8: USB Fault Message Enable. Set to 0 to disable Low Battery Warning Messages, 1 to enable.
- 9: USB Port 2 Default Power Enable. Set to 0 to disable power to USB Port 2 on power-up, 1 to enable.
- 10: USB Port 3 Default Power Enable. Set to 0 to disable power to USB Port 3 on power-up, 1 to enable.
- 11: Analog 1 Default Reference. One of the 4 Analog Reference Values for Analog 1 (0 - 3).
- 12: Analog 2 Default Reference. One of the 4 Analog Reference Values for Analog 2 (0 - 3).
- 13: PWM Mode. One of the 4 PWM Modes (0 - 3).

The PMIC/RTC EEPROM has a limited number of write cycles. Depending on the type of Configuration Value Index being written, the memory will only support a few tens-of-thousands of updates. While this should be sufficient for the purpose it was intended (making a custom system configuration) software should not make changes to the System Configuration unless absolutely necessary (do not treat EEPROM like a RAM variable).

Operation

The Pi Platter starts up in sleep mode when power is first applied (battery connected) and the real-time clock initialized to 0 seconds. Press the button for 2 seconds to turn 5V power on. Press the button for 6 seconds to turn 5V power off.

The Pi Platter will not turn power on while the battery voltage is less than the Battery Start Under-voltage Threshold.

Power can be turned on when the board is sleeping by the RTC. The RTC is simply a 32-bit second counter and can be associated with any time keeping scenario the user wants. It supports a single 32-bit Wakeup Time that can be enabled and is triggered when the RTC matches the Wakeup Time and the board is sleeping. There is also a 32-bit Wakeup Repeat time that is added to Wakeup Time every time an Wakeup condition is detected (Wakeup Time matches RTC and is enabled). This makes it possible to create a simple system that turns on the Pi periodically with software on the Pi turning the system off after it has completed its functions. The RTC runs as long the battery is connected.

Power can also be turned on when the battery voltage is charged above the Battery Restart Threshold. This value is stored in EEPROM and can be automatically enabled every time the Pi is powered by the EEPROM Battery Restart Enable or selectively for the next charge cycle by the Restart Enable Control register. This functionality is disabled by default.

By default all three USB connectors are enabled when the board is turned on. This is to allow the user to connect a keyboard/mouse in order to do initial configuration of their system. They can reconfigure the Pi Platter so that either of the USB ports on CON5 are disabled when the board turns on for power savings.

The PWM outputs are set to 0 (or Servo midpoint if PWM Mode is set to Servo) when the board is first turned on.

Default Configuration

Low Battery Threshold (configurable): 3.6V

Critical Battery Threshold (configurable): 3.1V

Battery Start Under-voltage threshold: 3.2V

Battery Restart voltage threshold: 3.8V

Battery Critical detection to turn-off: 30 seconds

Low Battery Warning Enabled

Critical Battery Warning Enabled

Battery Restart Disabled

USB Fault Warning Enabled

Power on USB Port 2: Enabled

Power on USB Port 3: Enabled

Analog 1 Reference: PMIC/RTC VCC (VD)

Analog 2 Reference: PMIC/RTC VCC (VD)

PWM Mode: Normal PWM - Slow

Low Battery Handling

The Solar Pi Platter detects two low-voltage conditions.

1. Low Battery: The battery is discharged but can continue operating the system for a while.
2. Critical Battery: The battery almost completely discharged and the system must be shut down to prevent damage to it.

Warning messages are generated if enabled and the associated bits in the Status are set.

Voltage levels must persist at Low or Critical levels for at least 5 seconds before a warning is issued. This prevents momentary voltage drops (from a servo, for example) from prematurely triggering a warning message (or power down).

Power will be turned-off a programmable time after the critical battery voltage level is detected. Power cannot be switched back on, either by the power button or a RTC Wakeup if the battery voltage is less than or equal to the Start-up under-voltage. The Wakeup Repeat interval is added to the Wakeup Time so that the board has a chance to power up at the next Wakeup Time if the battery has been sufficiently charged before then.

Software running on the Pi should initiate a controlled operating system shutdown when the Critical Battery condition is detected. This prevents corruption of the Pi's filesystem that may occur when power is removed from a running system.

Battery Restart Operation

The Battery Restart mechanism allows the board to automatically power-on the Pi when the battery is charged to a configurable voltage (Battery Restart Threshold). This functionality is disabled by default and may be enabled in one of two ways. The Battery Restart Enable EEPROM configuration entry can set the enable every time the Pi is powered. This may be useful in an application that should run whenever there is sufficient power in the battery and software running on the Pi does not have to do anything (other than execute an orderly shutdown if required to prevent SD card corruption).

If the Battery Restart Enable configuration entry is clear then system software running on the Pi can set the Restart Enable Control register to 1 to enable a restart on the next power down only. In this case the Restart Enable will be disabled on the next power up and the Pi must, again, enable it for the power cycle. This allows the Pi to set up a special condition if necessary.

The Battery Restart mechanism is only applied when power is shut down. It is checked once per second when the RTC is updated while the board is sleeping.

USB Overcurrent

Power-switched USB Ports 2 and 3 include a power-limiting function. Devices consuming more than 1 A initiate an automatic shut-down of the port while the over-current condition persists.

A warning message is generated if enabled and an associated bit in the Status is set.

Watchdog Operation

A watchdog function may be enabled that requires software on the Pi to write a control register within a specified period. Failure to write the control register causes the +5V power to be cycled to the Pi and IO subsystem for 5 seconds.

By default the watchdog function is disabled. Writing a non-zero value to the Watchdog Control control register starts a watchdog timeout. The current watchdog timeout value can be obtained by reading the register. Software on the Pi must either periodically reset the watchdog timer by rewriting the watchdog value or disable it by writing the value 0 to the control register. For example

```
C8=60<CR>
```

sets a 60 second watchdog timeout. Software must write C8=60<CR> again within 60 seconds or write C8=0<CR> to disable the timer.

Low and Critical Battery Threshold Calculations

The Low and Critical battery threshold configuration values may be calculated as follows. They are the ADC count threshold based on the Pi Platter battery sense circuitry and a Vref of 1.024 volts. Vthresh is the voltage level expressed in volts (e.g. 3.6).

$$\text{Threshold (0-1023)} = \text{round}(((V_{\text{thresh}} * 0.2362)/1.024) * 1023)$$

Note that the Pi Platter has been preset with reasonable values for Li-Ion batteries. These values should only be changed for specific reasons.

Start Under-voltage and Restart Threshold Calculations

The Start Under-voltage and Restart threshold configuration values may be calculated as follows. They are the ADC count threshold based on measuring a 1.024 volt reference against the PIC VDD power (through a diode with an estimated 0.15 volt Vf drop) when powered by the battery. Vthresh is the voltage level expressed in volts (e.g. 3.2). These are approximate values because the voltage is measured after a diode drop which may vary from board-to-board and based on temperature.

$$\text{Threshold (0-1023)} = \text{round}((1.024 / (V_{\text{thresh}} - 0.15)) * 1023)$$