Solar Pi Platter
Operating Instructions for Firmware 1.0, Board Rev 03
Ver 6 - 11/8/2016

Assembly Notes

1. Attach the plastic stand-offs to the Raspberry Pi Zero board and tighten before installing on the Pi Platter. This is because of the fit on the Pi against the HDMI and Power USB connectors and proximity to other Pi Zero components. It's easier to tighten the stand-off on the bottom of the board while holding the stand-off on the top of the board motionless.
2. Attach the battery first then any other power source.

Hardware Notes

The Pi Platter is designed for applications where the Pi will be powered down a significant amount of time while the battery is recharged. It can operate in a continuously powered situation for low load systems as described below. In this case the Pi would probably be required to do some active power management of USB peripherals. In general, it is assumed that the Pi uses the ability to power on and off the USB connectors as a way to save power.

Charging sources include low impedance USB power supplies connected to the Micro-B USB connector CON2 or high impedance supplies like a solar panel. The Pi Platter is designed for 6V panels (which typically have a maximum open circuit voltage of 7V) connected to the 5.5x2.1mm jack, CON1. Six- or 9-W panels are a good match (6-watt for areas with a lot of sunlight). The absolute maximum input voltage is 20V, however the charger will only operate with input voltages in the range of 3.5 - 7.0V.

Additional functionality is available with the USB power supply. The Pi Platter uses that source to directly power the system and charge the battery (current limits may apply based on the maximum current output of the USB power supply) automatically switching to powering the system from the battery when USB power is removed.

A single-cell, 3.7V, Li-Ion battery must be connected to either CON3 with a JST connector (compatible with batteries from both Sparkfun and Adafruit) or directly wired to CON4. The Pi Platter is designed for Li-Ion batteries with capacities between about 1000 mAh to 5000 mAh. Sparkfun claims that their batteries using the JST connector shouldn't have a load current in excess of 1A so CON4 should be used for systems that will have a higher load current as described below. CON4 also provides a connection for an external 103AT-4 thermistor that can be attached to the battery to limit battery charging in extreme temperature situations. Remove J2 to use an external thermistor.

All power inputs (charger and battery) are reverse polarity protected. The Pi Platter also protects the battery against any internal shorts or massive over current conditions.

Maximum current output is designed to be 2A at 5V. The boost converter that converts the battery voltage to 5V should run with an efficiency of 92-95% for output currents of a 1A or less. Its efficiency falls off as the battery voltage falls and as current rises above 1A. The following calculations assume 85% efficiency for an output current of 2A with a battery voltage of 3.1V and 93% efficiency for an output current of 2A with a battery voltage of 4.2V. Current consumption from the battery goes up as the battery voltage goes down.

- Maximum high-battery current draw (2A 5V load): 5V/4.2V/0.93 * 2 = 2.6A
- Maximum low-battery current draw (2A 5V load): 5V/3.1V/0.85 * 2 = 3.8A

The maximum battery charge current is 750 mA. This was chosen to support a variety of battery capacities and input power sources that match the intended functionality of this product. This charge current will be available from a low impedance source such as a 5V 1- or 2A USB charger when the current required by the rest of the system can also be supplied by the low impedance source. Charge current will be reduced if the low impedance voltage levels droop due to excessive current draw. It may not always be available from a solar panel depending on the amount of light and the maximum power output of the panel. The Pi Platter will try to optimize the power it can get out of the solar panel by backing down on the charge current to keep the charge voltage above the battery voltage. Any load being taken by the system will count against the charge current going into the battery.

The maximum continuous recharge time when charging from a high-impedance power source such as a solar panel is 10 hours. The charger enforces these timeouts (10 hours with a high-impedance source and 20 hours with a low-impedance source) to protect the battery. The charger can start another charge sequence if the battery voltage drops approximately 100 mV or if the charge supply stops (e.g. the solar panel sees dark). This should present no problems where the panel is sized correctly and there is sufficient sun to fully charge the battery in less than 10 hours. It may be necessary to limit the time the Pi is powered up to ensure these conditions.

Base current should be assumed to be about 320 mA at 5V (400 - 560 mA battery draw). This includes the Pi Zero board and the initialized USB Subsystem on the Pi Platter, but no USB peripherals. Ethernet adds about 75-80 mA from the 5V supply.

USB Connector CON 5 is identified as USB devices 2 and 3. Power to these two USB devices may be switched on and off (on by default). USB Connector CON7 is identified as USB device 1. There is a 1A current limit imposed by the USB power switch for CON5. CON6 has no current limiting. Over-current may cause the boost converter to shut down when running on batteries or the USB power source to current limit.

PWM1 and PWM2 expansion ports are identified as PWM devices 1 and 2 respectively.

Analog inputs A1 and A2 are identified as Analog devices 1 and 2 respectively. The Analog inputs are converted to a value with a configurable voltage reference. The voltage reference may be set to one of three fixed, precision, values (1.024, 2.048, 4.096 volts) for making absolute measurements or it may be set to the PMIC/RTC power supply rail (VD) for making relative measurements. Input voltages on A1 and A2 should never exceed VD or damage to the PMIC/RTC may occur.

The PC ("Power Control") expansion in port is connected in parallel to the power button. External circuitry (e.g. a relay or N-Channel MOSFET) can pull this signal low to simulate a button press. Assert the signal low for more than 2 seconds to power on the board. Assert the signal low for more than 6 seconds to power off the board.

The VD, VB, 5 and 3.3 expansion connections provide access to the various power rails in the system. VD is the PMIC/RTC VCC signal and might be useful as the reference voltage for devices using the Analog inputs and setting Vref to VCC. VB is the battery voltage. Current should be limited to 500 mA. 5 is the 5V output. 3.3 is a 3.3V output (that is shared with the USB and Ethernet subsystem). Output current on the 5V line is limited to a system total of 2A. Output current on the 3.3V line is limited to 100 mA.

G expansion connections are connected to ground. There are several to simplify connecting to various external devices.

The ethernet expansion port is designed for use with the Sparkfun BOB-13021 RJ45 MagJack breakout board (Adafruit has no similar product).

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

Operational Notes

The Pi Platter starts up in sleep mode when power is first applied and the real-time clock initialized to 0 seconds. Press the button for 2 seconds to turn power on. Press the button for 6 seconds to turn 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 (default 3.8 volts) 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.

The Pi Platter shows up as an ACM serial port to the Raspberry Pi. Depending if any other ACM serial devices are connected to one of the USB ports, it will show up as /dev/ttyACM0 (or /dev/ttyACM1 if there is a USB ACM device, etc). The PMIC/RTC micro-controller is connected to USB port 4 on the board's hub so will enumerate after other USB ACM devices. The baud rate does not matter since it is a USB ACM device.

Warnings are enabled by default. They are simple text strings with the form "WARN <N><CR><LF>" where <N> is a number

identifying the warning type, <CR> is the carriage return character, and <LF> is the linefeed character. A serial connection must be open in order to receive any warning messages. Individual warning messages may be disabled.

- "WARN 0": Low Battery Warning Message
- "WARN 1": Critical Battery Warning Message
- "WARN 2": USB Fault Detected Message

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 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.

A watchdog function can be enabled by writing a watchdog timeout period (in seconds 1-65535) to a control register. Software running on the Pi must write a new watchdog timeout to this register (or the value 0 to disable the watchdog timeout function) before the specified number of seconds passes or the Pi Platter will turn the 5V power off (Pi and USB) for 5 seconds. Reading the register indicates how many seconds remain before the watchdog reset.

Command Interface

Commands are simple ASCII sequences with two forms terminated by a Carriage Return character (<CR>), shown below, sent to or received from the 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 a 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 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.

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<CR> - Returns the battery voltage (B=<VOLTS><CR><LF> - for example "B=4.05")

Power **O**ff - "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.

O=30<CR> - Sets a 30 second timeout until power down
O<CR> - Returns the number of seconds until power down (O=<NUM><CR><LF>)

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.

U1=1<CR> - Turns USB Port 2 on
U2=0<CR> - Turns USB Port 3 off
U1<CR> - Returns the power control state for USB Port 2 (U1=<N><CR><LF> where <N> is 0 or 1)

Real Time Clock

Current **T**ime - “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> - Sets the RTC to 1000 seconds
T<CR> - Returns the current RTC value (T=<SECONDS><CR><LF>)

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> - Sets the Wakeup Time to 1100 seconds
W<CR> - Returns the current Wakeup Time (W=<SECONDS><CR><LF>)

Wakeup **R**epeat 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> - Sets the Wakeup Repeat Time to 86400 seconds (1 day)
R<CR> - Returns the current Wakeup Repeat Time (R=<SECONDS><CR><LF>)

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> - Returns the 10-bit unsigned value from Channel 1 (A1=<VALUE><CR><LF> where <VALUE> = 0 - 1023)
A2<CR> - Returns the 10-bit unsigned value from Channel 2

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.

P1=40<CR> - Sets PWM 1 to 40
P2<CR> - Returns the 8-bit value from PWM 2

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

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.

C0=1<CR> - Sets Control Register 0 to 1
C1<CR> - Returns the value of Control Register 1

The 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> - Returns the 8-bit status value.

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 to power-up the board)
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).

V<CR> - Returns the version (V=<VERSION><CR><LF>, Version 1.0 returns "V=65792")

System Configuration

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.

E9=0<CR> - Sets Configuration Index 9 (USB 1 Default Enable) to 0 (off).
E14<CR> - Returns the value of Index 14 (PWM Mode)

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 (see below).
1: Battery Critical Threshold. A 16-bit unsigned integer (0-1023) that sets the Critical Battery Threshold (see below).
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).

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).

Threshold (0-1023) = ((Vthresh * 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).

Threshold (0-1023) = (1.024 / (Vthresh - 0.15)) * 1023

Special Note for EEPROM access
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).

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 set setup 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.

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.