Table of Contents

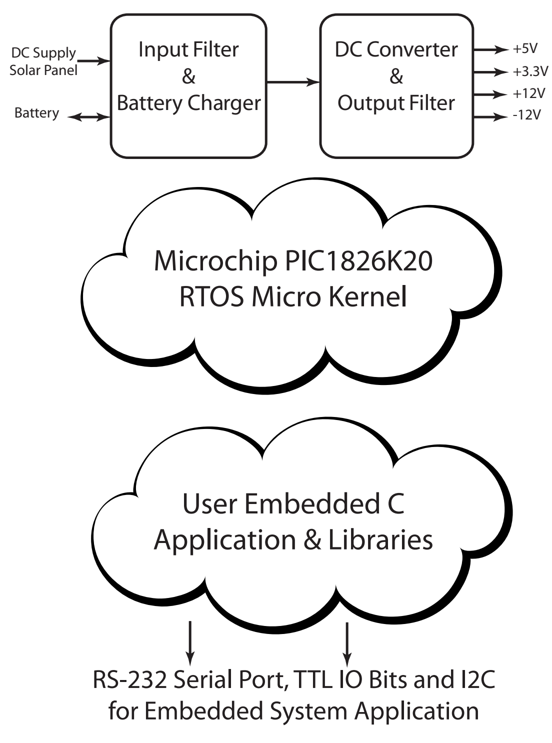
Introducing the SunDuino

1

Overview

General Description

The SunDuino is a complete micro-controller board with an integrated battery system designed for instrumentation, data collection and portable equipment support. Both solar panels and DC power sources are used to charge external batteries and supply power to regulated outputs. User written applications can be loaded into 48-KB of included FLASH memory for execution in parallel with SunDuino battery



charger and control software. A micro kernel provides cooperative multitasking between user-loaded and native-battery functions.

The software libraries supplied with a SunDuino board provide a means for user applications to interface with SunDuino power functions. By using the SLEEP mode support supplied in these libraries, a user application can easily execute operations while using as little as 200 ua. Small rechargeable batteries provide the means for the SunDuino board to operate at this current for a year or more.

All connections to the SunDuino are made through a single 25 pin D male connector. Various connector styles such as solder cup and mass termination connectors can be used to simplify physical interface with user system components.

A SunDuino can manage batteries of various chemistries and sizes. Charging and battery control is based on a downloadable EEPROM configuration table. Configuration tables are provided for numerous types and sizes of batteries. New battery support is easily downloaded.

SunDuino interfaces can be divided into four sections.

- Primary Power Input
- Battery Connection
- Regulated Power Output
- Control Logic

Primary Power Input

Unregulated Direct Current power is supplied to the SunDuino through board's DC Input. This input provides power for target operation, battery charging, and internal operation. A low-cost wall mount type supply is the most common source for primary power. Solar panels are supported along with provisions for setting the Maximum Power Point (MPP) of the connected solar panel. Charging logic will attempt to maintain the output of each panel at its specified MPP to maximize power delivered.

A major requirement for the primary power source is the ability to maintain proper input voltages during both target operation and battery charging. To charge the battery and run loads properly, the DC input voltage must be maintained at a level greater than the receiving battery's peak voltage. For an 12V lead acid battery pack, the corresponding minimum DC input voltage must exceed 16 volts (15.0v charging voltage and 1.0v 'overhead'). Note that this 16 volt figure includes leeway for any ripple voltages. If the input voltage is ever less than 16 volts, battery charging will halt during final charging. If the DC input voltage continues to drop, the battery will start to discharge as the output converter is switched to battery power.

A major source of failure in microprocessor based systems is the intermittent application of power. Rapid application and removal of power to a system will at best hang the system. At worst such behavior can damage both the power supplies and the processing equipment. Hardware circuits and software logic within the SunDuino isolate the target from severe input variations and supply power to the target in a controlled manner.

Battery Operation

Battery operating characteristics are maintained within a table held in processor EEPROM. This table of ~43 parameters is configured to control operation of a specific battery type and size. These EEPROM constants can be overwritten by the simple download of an alternate .hex configuration file. A checksum is included in this table and SunDuino software will verify this checksum when power is applied.

Optimization of battery life is a primary goal in the SunDuino software design. Major logic includes functions that limit over-discharge during battery usage and over-charging during charge cycles. Charge limit is a particularly difficult task given the desire for 100% charge at a maximum rate. Internal software logic adjusts the charge cutoff point to account for changes in total charge, cell voltages, and temperature. Additional logic attends to peak charge maintenance and eliminates the overcharge associated with constant current trickle charging.

While the SunDuino controls all phases of battery charging - design of a robust system that takes full advantage of the SunDuino requires an understanding of battery basics. A broader understanding of battery operation will help with the selection of the appropriate battery pack for a specific load - a selection that will significantly enhance system operation.

Batteries are complex chemical systems which convert energy between electrical and chemical forms. Power, or the rate of energy transfer, requires a corresponding chemical reaction rate. Batteries do not maintain a constant ability to react agents as their chemical composition changes. A fully charged battery contains a large amount of active material, providing delivery of energy at a high rate. Since the fixed voltages on SunDuino outputs deliver constant power to the load, the DC converter requires greater battery current as the battery voltage drops. A battery with a low charge does not contain the same supply of active chemicals. A partially discharged battery cannot provide the high peak currents possible with a fully charged battery. Low temperatures will further reduce the ability of a battery to supply power. Significant battery de-rating is required for low temperature operation with partially discharged batteries.

When selecting a battery it is important to consider peak load currents in addition to the total mahr capacity. A good example is a portable instrument using a thermal print head. Print heads draw more than 2 amps during brief printing cycles while the remaining system might draw only 100mA. Selecting a 1 Amp-Hour battery for 10 hours of operation will result in failure after only a couple hours of operation. Failures take place when the print head energizes, battery voltage drops, and the system loses power resetting the processor and jamming the printer. The key to a successful system is knowing the remaining battery capacity and disabling the printing process. The SunDuino solves these problems by maintaining this information and making it available to the user applications for the control of system operation.

Software logic within SunDuino and its EEPROM tables prevent operating the battery down to the point of cell reversal. Detection of a minimum battery voltage will abort the discharge process and shut off regulated power. At this point the battery is considered EMPTY, and battery usage will not resume. Restoring primary power will not clear the EMPTY condition. If primary power again fails while the battery is EMPTY the target is turned off and the battery is not used for power. Only after the battery is partially recharged will the EMPTY condition be cleared and the battery made available for target operation.

DC Outputs

Output Converter

A 100 KHz flyback converter accepts energy from the DC input or battery and delivers this energy to 4 output windings. Since one of the flyback outputs (either the +5 or +3.3 outputs) controls flyback feedback, and given the nature of the coupled secondary flybacks, all 4 outputs remain at constant output voltage while the primary input voltage changes (Line Regulation). SunDuino line regulation is typically < 2%. As the target load changes, the remaining 3 none-monitored outputs do change voltage slightly, typically less than 5%. The single monitored output is very stable with load variations affecting output voltage measuring at less than 2%. The level of regulation is dependent on the actual load current, while the minimal load current (about 10mA) can greatly improve flyback coupled regulation.

Digital systems typically require the greatest regulation on the main logic supply voltage, with less regulation required for analog processing supplies. The SunDuino provides two digital outputs: +5 and +3.3 volts. Either of these outputs can be directly regulated. Some systems require both +5 and +3.3 supplies. Strapping options on the SunDuino connector allow selection of the most variable load, or the most heavily loaded output for direct regulation. Construction techniques used in manufacturing the main flyback inductor provide a high degree of regulation between the +5 and +3.3 sections.

Voltage for analog processes consist of two outputs: +15 and -15 volts. These outputs are not directly regulated but are filtered to reduce noise. Flyback operation maintains the load regulation of these outputs to within 10%. All output voltages are adjustable via a potentiometer mounted on the SunDuino PCB. All voltages are adjusted proportionally. The BB25E unit includes 12V, 100mA linear regulators with strapping options to allow selection of the 15V raw outputs or 12V regulated outputs. The BB10B is hard wired with 13V flyback regulated outputs.

BB25E units include additional common mode chokes for EMI control. Multiple winding common mode chokes provide a high degree of isolation for high frequency signals between the BB25E and the regulated outputs. This high level of filter isolation make the BB25E an ideal power source for sensitive analog processing and measurement instruments. High frequency noise on digital power lines is also blocked by these chokes, simplifying system EMI filtering and regulatory compliance.

The BB10B is a smaller, lower power-lower cost unit. It does not contain the additional common mode chokes. However, both BB10B and BB25E units incorporate multilayer PCBs which help to minimize system noise.

Control Logic

A single On/Off control signal along with internal SunDuino software provides the user with a number of possible power control options. These options include:

- Push On, Push Off (‘Toggle’) Operation
- Push On, Release Off Operation
- Continuous Operation (UPS Mode)
- Push On - Target Software (Hold and Release)

Control of the Battery Boss ON/OFF status is through a signal on the 25 pin connector. A 3 volt signal applied to this ON/OFF circuit (> 10K ohm input impedance) will cause the switching regulator to start if power conditions permit. Software logic will not permit the inverter to start if the primary voltage is less than permitted or if pervious operation has left the battery in an EMPTY condition. When the converter starts, hardware circuits limit inrush current and temporarily disable primary input voltage monitors.

Power Control

2

Operating Modes

The SunDuino operates in three configurations depending on application requirements. Variations include:

- Operation with both Battery and User Application
- Operation without User Application
- Operation without Battery

Operation with Battery and User Application

The most common application configuration of the SunDuino PCB uses a battery for backup power while providing DC power conversion and battery charge control in the background. The ‘On/Off’ input on the 25 Pin D connects directly to either a momentary switch, a toggle switch, or an external control signal. Logic in lower level SunDuino code allows a momentary switch to toggle power ‘On’ then ‘Off’. A user application executing in the SunDuino can access these signals as well as exert direct control over power outputs and can cycle of the user load ‘On’ and ‘Off’. A user application can also lock power into an ‘On’ state during critical operations such as a data storage write operation.

User applications are downloaded from a PC using the SunDuino Boot-Loader program. The boot-loader operates using the Microchip AN-1310 protocol. Communications between the PC and SunDuino through this boot-loader are initiated via a push-button interface, after which a 60-second timeout is provided within the SunDuino firmware to prevent inadvertent lockout of normal operation. The SunDuino status LED toggles with the button push and then stays on throughout the duration of boot-load operation. Additional features such as download of battery configuration files and controlling operation modes are also supported by the SunDuino Boot-Loader application.

Operation without User Application

Many applications exist where power requirements do not exceed simple Uninterruptible Power Supply (UPS) operation. While the

SunDuino is operating in this mode, no user application is downloaded into SunDuino FLASH and only the internal control logic is executed.

Operation of the target power is controlled by the On/Off signal (25 pin D connector) while UPS and battery charger functions operate normally. If SunDuino EEPROM has been configured to operate with a solar panel, the firmware will attempt to maintain operation at the Maximum Power Point (MPP) of the panel. The SunDuino Boot-Loader program is used to customize operating modes and set up solar panel operation.

Operation without Battery

Operation of the SunDuino does not require that a battery be connected. The SunDuino detects when a system battery is not connected and will automatically change operating modes to compensate for the change. When power is applied to the DC input and no battery is connected, the SunDuino will clear the Gas Gauge, start as normal, and then start execution of the user application. When power is removed, the SunDuino will shut down as the input voltage fails. During SLEEP MODE operation, the low level of operating current can result in the SunDuino running off of internal filter caps for 10s of seconds.

Whenever the SunDuino detects a missing battery, the Gas Gauge is reset to zero. This zero value is also written to the EEPROM locations where knowledge of the remaining capacity is stored. If a battery is disconnected and remains disconnected for 30 seconds or more during standard operation, the SunDuino will declare the battery as missing and the Gas Gauge values will be reset. If a battery is disconnected for service it must be reconnected quickly. If a new battery is connected the user should wait 2 minutes to ensure the Gas Gauge has been cleared.

Sleep Mode Operation

Active vs. Sleep Currents

When operating any system from a battery, it is important to limit execution time as much as possible to minimize power requirements, resulting in an increased battery life and run time. Turning off processor clocks and the ability to quickly power down the proper circuits is critical, but also critical is the SunDuino's ability to wake rapidly. A standard PC might take 10s of seconds to wake, whereas an embedded processor could have the need to wake up in a matter of microseconds.

The handling of how a system moves into and out of sleep is often a complex section of embedded design and is prone to various unseen design issues. The SunDuino provides this function within its lower level RTOS kernel.

Internally, the SunDuino maintains 4 major states: ‘Idle’, ‘Running on Battery’, ‘Full’, and ‘Charging’. ‘Idle’ is a key state, as it indicates that the DC power converter is off and NO input voltage is present from a DC supply or solar panel. It is in the ‘Idle’ state that the SunDuino can enter sleep to reduce power consumption. During normal operation, the basic SunDuino processor circuits draw approximately 10mA, in sleep, the current is reduced to 10s of microamperes. The key parameter associated with average power is the percentage of time the processor is operating. This is about 1% in the SunDuino, including the execution of a minimal user application. At a minimum the SunDuino exits from sleep at 8Hz (every 125ms) and executes for about 1ms. The resulting usage is approximately 150 ìA of average battery load. If the user application does significant processing (as in floating point operations) during the wake time, this average draw increases.

Entry into sleep must be coordinated with any user application running on the SunDuino processor. Since only the user application knows when it is safe to sleep the, SunDuino library includes a function by which the processor can invoke sleep mode. The function returns to the user application when one of several interrupt events are triggered.

Sleep Wake Up Events

There are several events that cause the processor to exit sleep and return to the user application.

- Real Time Clock tick at 8 Hz.
- UART Receive character.
- External ON/OFF signal change.

If the processor is in sleep the above events will bring it out of sleep. There are also a number of processes that will prevent the processor from going into sleep.

- Battery system not in ‘Idle’ state.
- Critical analog processing running.
- Various Errors. (Missing battery or power errors.)

Examples of using SLEEP MODE can be found in the various sample application provided with SunDuino toolkits. In general a “Main Loop” in a SunDuino application will look something like:

```
    while (1) {
        do
        {
            SleepRes = bb_func_DoSleep ();
            RX_Process ();      //Check for incoming characters
            Sec = bb_time_getSeconds();
        } while ( Sec == j);
// Drop through means 1 second has passed, do something
         do_something();

        j = Sec;                 //Setup to detect next second.
```

In the above loop code at “do_something()” will execute once a second. The SleepRes variable can be used to detect if or if not sleep mode was blocked and why. User application can use SleepRes to control other software threads if required. In this sample the processor will exit from sleep and use a change in the Real Time Clock Seconds counter to trigger execution of the do_something() routine.

SunDuino Software

3

XC8 Compiler/Linker

SunDuino PCBs use the XC8 Microchip C compiler/linker for the PIC18F26K20 processor. Various XC8 versions are available, including a free version that requires minimal code optimization. Even when using the free compiler to compile a user application, floating point libraries and extensive ‘printf’ functions, only about 30% of the available 48KB of program FLASH is occupied. User applications without ‘printf’ statements use significantly less FLASH memory, typically only a small percentage of available program FLASH. Advanced optimizing compilers are available for a fee.

Processor Memory Model

SunDuino firmware for low level battery control and charging is stored in FLASH locations 0x0000-0x3FFF. These locations are protected from external reading and writing. The remaining memory space (0x4000-0xFFFF) is free for user applications, typically written in C. SunDuino low level code also uses 256 bytes of RAM from locations 0x100-0x1FF.

Battery configurations and operating constants are stored in the low 128 bytes of EEPROM. These locations can be programmed with one of various battery files, allowing users to easily switch between battery types and sizes. Example source code and .hex files for various battery types and sizes are provided. A Boot Loader is provided for downloading FLASH and EEPROM from .hex files using a PC with COM port.

When a project is setup in the MPLAB tool, both a code offset and RAM exclusions are required by the XC8 Linker:

Memory Model Tab: (In hex.)
RAM ranges   default,-100-1ff

Additional Options Tab:
Code Offset 4000

Both of these settings are required to ensure that users' compiled C applications do not use SunDuino's reserved RAM locations and that executable code is pushed into where the write-safe areas of FLASH memory begins (0x4000).

When the SunDuino powers on, the battery charging and control firmware is initialized. If location 0x4000 is anything but erased (erased bytes have a value of 0xff), a jump is made to location 0x4000. XC8 will place the run time startup code at 0x4000 (with the code offset defined in the linker), and C code executions takes place. The C application should execute within an indefinite 'while' loop and never return.

C Library Functions

The following is a partial list of SunDuino library functions. See SunDuino_Vxxx.h for details.

Command Name	Cmd Letter
bb_battery_getGas()	Returns remaining capacity as mahr.
bb_battery_getPercentage()	Returns remaining capacity as 0 - 100.
bb_battery_getSize()	Returns full capacity of battery.
bb_getTemperature()	Returns battery temperature from -40C to 80C.
bb_volts_getVRun()(	Returns DC input voltage. 0V to 40V
bb_volts_getVBattery()	Returns battery voltage. 0v to 40V
bb_volts_getVCharger()	Returns voltage available to charge battery 0V to 40V
bb_amps_getAverageCurrent()	Returns current used by output regulator. 0 Amp to 6.5 Amp
bb_func_DoSleep()	Used to place processor in sleep if permitted.
bb_func_PwrOnOff(Sec)	Used to initiate an application WDT.
bb_func_PowerIsOn	Returns 1 or 0 if output regulator is On or Off.
bb_time_getxxxx()	Functions to read time from Real Time Clock.
bb_date_getxxxx()	Functions to read date from Real Time Clock.

Microchip provides a large library of C functions for numerous types of IO. The most important of these functions are 'I2C' and 'printf'. I2C allows the user application to control a vast array of hardware functions, while printf sends characters to external devices using true RS-232 levels. Various power control states and analog readings are available from the SunDuino libraries, for a more detailed description of these, see the SunDuino_Vxxx.h header file. Sample C files are provided.

Once a user application is compiled into a .hex file, the SunDuino Boot Loader can be used to download this file into SunDuino FLASH or EEPROM. The Boot Loader runs in a PC and uses a COM port to download code.

A button on the SunDuino PCB must be pushed to start the Boot Load process which also aborts all other SunDuino processes. Both the battery charger and output DC regulator are disabled during Boot Loader operation. When using the Boot Loader, it is impossible to overwrite low level code in FLASH. Note that (FLASH is write protected, but EEPROM is writable. Various battery configuration .hex files are provided for programming the lower 128 bytes of EEPROM. It is these locations that control operation for a specific battery type and size. Writing to any other locations in the lower 128 bytes of EEPROM is not recommended.

The Boot Loader allows .hex files and .ini files to be passed as command line (or desktop shortcut) options. A complete .ini file is written to a default file name each time the Boot Loader closes. If this .ini file is copied to another directory and renamed, it will completely define a programming session. Passing this renamed .ini file as a command line operation will completely control a programming session and, causing the default .ini file to be ignored. When the Boot Loader uses a command line option, it will automatically close, allowing support with simple batch control files.

The default .ini file location is:

C:\ProgramData\IndexDesigns\SunD_Loader.ini

When a .hex file is passed on a command line, the .hex file will be downloaded using programming options from the default .ini file. Again, the Boot Loader will automatically close once the file is downloaded.

LED Blink Patterns

A small LED on the SunDuino PCB is used to provide feedback to the user about current operation.

- Fully Charged — Solid On
- Charging — 1 Second On and Off
- Run On Battery — 1/2 Second On 1.5 Off
- Sleep Mode — Short Blink Every 5 Second

- Boot Load Mode   Solid On

The Sleep Mode short blink is VERY short, only milliseconds in duration. Users must watch closely as this short blink only happens once every 5 seconds. This short blink was chosen to minimize battery current drain during SLEEP MODE operation.

SunDuino Hardware

4

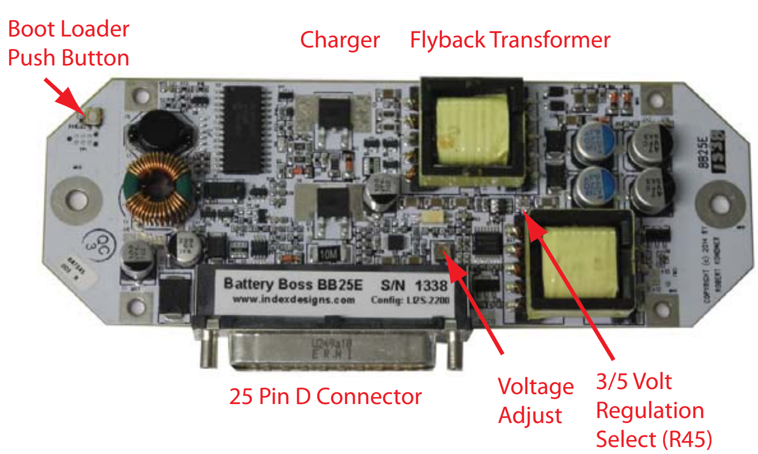
Overview

SunDuino hardware exists in two versions, the BB25E and BB10B. Both are similar in function, and both run the same firmware. The BB10 is the smaller, lower power, and less expensive of the two boards. The BB25E is built using an aluminum backed PCB which supports operation at higher power through improved thermal management.

Each version provides a general purpose user analog input. A gain set resistor at location R20 determines the full scale input voltage range. A default value of 10.2K ohms at R20 provides a full scale reading of 10.0 Volts. Adding an external 15K 1% resistor in series increases this value to 40V . See schematic for details.

BB25E

The most distinguishing feature of the BB25E is the merging of a thin, thermally conductive, multi-layer PCB and an aluminum base plate.



The metal base spreads heat away from various power components, allowing operation at higher power. This technique eliminates the need for heat sinks and simplifies assembly of the final user system. Two common mode chokes are included, one on the DC input (left side), and one on the four DC outputs (on the lower right). Two large mounting holes (one on each side), as well as four smaller mounting holes are provided for mounting simple user hardware. The back side of the PCB is bare aluminum and is electrically isolated from other circuits. The top side of the mounting holes are connected to the CHASSIS GND signal. Inserting a metallic screw into these holes will bond the CHASSIS GND to the aluminum base. Small ceramic bypass caps on both the input and output circuits form AC connections to SIGNAL GND. Care must be taken that the CHASSIS GND to input and the output GND voltage do not exceed the 50V rating of these bypass caps. See the schematics.

While the EFD20-size flyback transformer will operate at 25 watts, the true limits are affected by input voltage, available battery current, and thermal design. Attempting to supply 25 watts from a 6 volt battery would require 4-5 amps which is greater than what most small batteries can provide. Output diodes would be stressed both electrically and thermally if that level of power was delivered to the 3.3V or 5V outputs. OEM implementations with output voltages higher than 5V can easily reach 25 watts or more with custom magnetic components and synchronous rectifiers.

The BB25E includes an opto-isolator and drive circuit which together make the 5, 3.3, and +/- 12 volt outputs optically isolated from the input power side. Care must be taken that any common mode voltage differences do not exceed 50V as the circuits contain small AC bypass capacitors between the various GND signals.

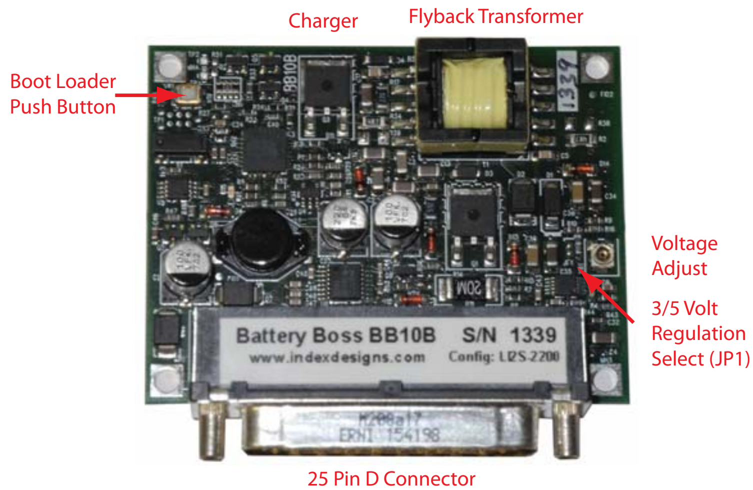
Two linear regulators are included for +/- 12V regulation from the flyback 15V outputs. Maximum output current from these regulators is 100mA. Flyback windings are designed to provide the +/- 15V which drives these linear regulators. Jumper pads are included on the lower and right PCB edges which can short out the regulators and allow delivery of +/- 15V to the SunDuino output. Solder blobs are easily added.

The BB25E also includes a location where .050 inch-spaced posts can be mounted for the attachment of a Microchip In-Circuit-Development (ICD) connector. While small probe points are available for factory programming, engineering development is simplified with a fixed ICD connection.

System testing and development is best done using a BB25E. If development is accomplished through the ICD connector care must be taken to NOT ERASE FLASH. Doing so will delete the low level SunDuino operating code and RTOS. Erasing this flash will require factory re-flashing as the low level FLASH code is not released as a hex file for user programming.

BB10B

The BB10B is a simple four layer PCB built using 0.062 thick FR4 material. While internal power planes help conduct heat away from power components, overall thermal characteristics limit total available power. A smaller flyback transformer along with lower charger currents are used as a consequence of this simplified thermal design.



Mounting can be accomplished using thin layers of mounting adhesive tape or by using the 4 corner mounting holes with #4 hardware. The bottom side is void of components, which allows simple mounting to the flat surface. The bottom side of the BB10B does contain circuit traces, so mounting to conductive surfaces requires insulation.

CHASSIS GND on the PCB is connected to the 4 mounting holes. Small 50V ceramic bypass caps are included between electrical GND and CHASSIS GND, so any common mode voltage difference must be limited to 50V.

5V / 3.3V Regulation Selection

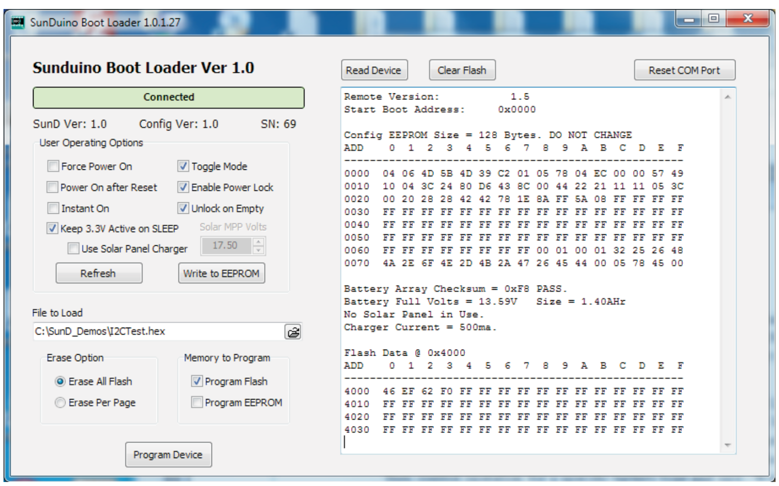
Each SunDuino version includes a 0603 sized set of mounting pads which can be used to control the source for output voltage regulation. The BB25E part is labeled as R45 while the BB10B part is labeled as JP1. When these locations are open (default), the 5V output is directly regulated. Installing a jumper (or solder blob) at this location connects the 3.3V output into the regulation circuit which causes the 3.3V output to be directly regulated. In place of a short, a small resistor can be used to force regulations to a midpoint of both 5 and 3.3 volt outputs.

Boot Loader

5

Overview

The SunDuino Boot Loader is a Windows PC application with the main form shown below. The left hand column shows from top to bottom: the app version, the green or red connection status, user options, down-loaded file, erase options, and memory areas to program. The right hand side shows Read Device, Clear Flash, and COM Port Select but-tons. Also on the right hand side is an area where device content and programming status is displayed.



SunDuino Boot Loader Main Screen

When the app connects or the Device Read button is clicked, various key memory areas are read and their content is summarized. The upper section shows EEPROM content for the first 128 bytes, with the lower section showing the start of FLASH. Users should never write to the lower 128 bytes of EEPROM. Only approved battery control files should be loaded into the lower 128 bytes of EEPROM. Notice that lo-cation 0x4000 contains compiled code which indicates that a user ap-plication is present. This screen provides a quick reference to battery type, status, and user application content. The green “Connected” area is red if the PC cannot detect a connected SunDuino that is in Boot Load Mode. A user must press the small button on a powered

SunDuino to initiate Boot Load Mode. Assuming the COM Port is connected and working, the pressing of the SunDuino PCB push button will allow the Boot Loader to connect as shown above. If a connection is not valid, the Boot Loader application will display ‘Not Connected’ over a red background.

This connection scheme allows the user to verify the quality of the COM Port connection. Every 500ms the Boot Loader PC application verifies the SunDuino is still connected by sending a small message to the SunDuino PCB. If this connection fails an internal ‘FAILED CONNECTION’ counter increments. If this counter ever reaches a value greater than 0, notice is displayed in the green connection field. By leaving a SunDuino thus connected for several hours (e.g. overnight), any connection loss will be captured and counted, however, a correctly operating COM connection should never fail.

The SunDuino firmware includes a 60 second connection time-out timer. Should the push button on the PCB ever be pressed WITHOUT a connection to the Boot Loader application, the SunDuino firmware will automatically reset after 60 seconds.

Operating Options

SunDuino firmware provides several options for controlling power operations. A number of check boxes are provided along with a WRITE button which allows the user to configure these options without any file downloads. When the user writes to EEPROM, a checksum is calculated from the resulting values and written alongside the user’s write in EEPROM. SunDuino firmware requires that a valid checksum be provided. An invalid checksum will cause the SunDuino PCB LED to rapidly blink while the SunDuino disables all other operations.

Control Option	Description
Force Pwr On	Forces Output DC Regulator ON
Pwr On after Reset	Force Output DC ON after Reset. (With 2 Second Delay)
Instant On	Force Output DC ON after Reset. (No Delay)
Toggle Mode	Input ON/OFF signal Toggles Output DC On/Off
Enable Pwr Lock	Prevents On/Off switch from turning DC Output Off
Unlock On EMPTY	If Battery hits EMPTY then clear Power Lock.
Keep 3.3V Active	Keep Trickle regulator ON during SLEEP Mode
Use Solar Panel	Enable use of Solar Panel Charging Logic
Solar MPP Voltage	When using Solar Panel this filed defines panel MPP
Write EEPROM	Click this button to WRITE all Options to EEPROM
Refresh	Click the button to refresh controls from EEPROM

Advanced Usage

The Boot Loader allows .hex files and .ini files to be passed as command line (or .lnk file) options. A complete .ini file is written to a default file name each time the Boot Loader closes. If this .ini file is copied to another directory and renamed, it will completely define a programming session. Passing this renamed .ini file as a command line option will completely control a programming session. When the Boot Loader uses a command line option the Boot Loader will automatically close. This means that the SunDuino integrates well with simple batch control files.

The default .ini file location is:

C:\ProgramData\IndexDesigns/SunD_Loader.ini

When a .hex file is passed on a command line, the .hex file will be downloaded using control options from the default .ini file. Again, the Boot Loader will automatically close once the file is downloaded.

Included in the .ini files is a provision for starting a new application when the Boot Loader exists. Parameter "ExitFile" defines a file to be opened by the Windows application associated with the indicated file type. The following sample line shows an ExitFile with a .ht file extension. This will result in a Hyper Terminal session being started using the connection profile of the TTY_9600.ht session file.

ExitFile="C:\ProgramData\IndexDesigns\TTY_9600.ht"

Sample Wiring

6

25 Pin D Signals

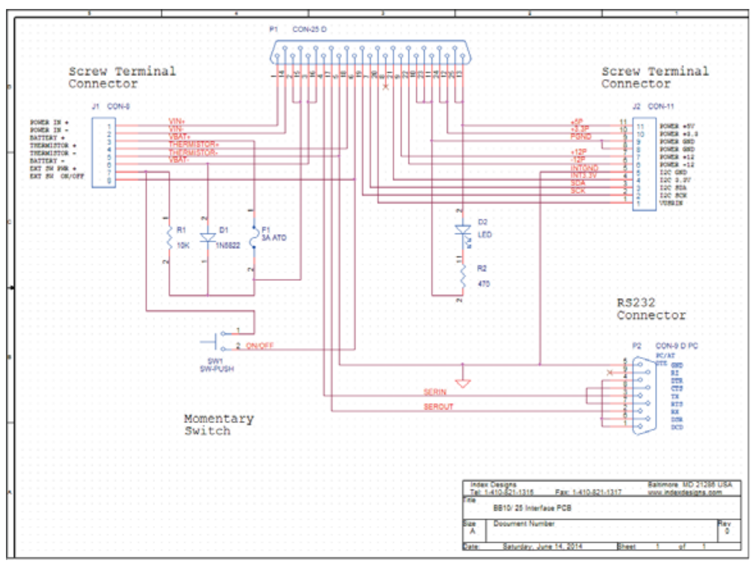
Connector Signals

The following table shows the pinout for the SunDuino 25 pin D Connector.

Pin	Function
1	VIN+ DC Input Power
2	VBAT + Battery Positive Terminal. Connect in Parallel with Pin 15
3	VBAT- Battery Negative Terminal. Connect in Parallel with Pin 16.
4	RS-232 Serial Input
5	INTGND Internal Ground.
6	ON/OFF Control Signal. Turns on at 3V and is 20V tolerant.
7	SDA signal for I2C. Port C bit 4 on processor.
8	Clock Out from processor. Port A bit 6.
9	+12V linear regulator output on BB25E. +13V on BB10B.
10	Output GND. Optoisolated on BB25E. INTGND on BB10B.
11	Output GND. Optoisolated on BB25E. INTGND on BB10B.
12	+3.3V Output. Connect in Parallel with Pin 24.
13	+5V Output. Connect in Parallel with Pin 25.
14	VIN- Primary Power Input -
15	VBAT + Battery Positive Terminal. Connect in Parallel with Pin 2
16	VBAT- Battery Negative Terminal. Connect in Parallel with Pin 3.
17	RS-232 Serial Out
18	Battery Thermistor. 10K Ohm 0.5% NTC Battery Thermistor.
19	SCK signal for I2C. Port C bit 3 on processor.
20	Input Select. Connect to Pin 21 to Directly Regulate the 3.3V Output.
21	3.3V linear trickle regulator directly from battery. Max. current 10ma.
22	-12V linear regulator output on BB25E. -13V on BB10B.
23	Output GND. Optoisolated on BB25E. INTGND on BB10B.
24	+3.3V Output. Connect in Parallel with Pin 12
25	+5V Output. Connect in Parallel with Pin 25

Interface Board

To simplify connection of external hardware to the SunDuino and aid in system integration, a SunDuino "Break Out Board" is available. Power connections are divided by functions and brought out to terminals with screw-down connections. The RS-232 serial connections are brought out to a 9 pin D connector with a pinout for direct connection to a PC (PC is DTE. Breakout Board is DCE). Additional features, such as a fuse, reverse battery diode, and On/Off switch are included. Note that the large diode will short any battery connected in reverse, and the resulting current will clear the board fuse. A push-button switch with external connections is connected to the SunDuino On/Off signal.



Also notice that the DC Output's common signals are separate from the INTGND signal. The BB25E DC outputs are optically isolated, while the low cost BB10B uses a common GND design. If I2C is to be used with circuits powered by the DC Outputs, the two GNDs must be connected.

I2C Voltage Levels

3.3V and 5V I2C Operation

The SunDuino process uses a PIC18F26K20 processor running at 3.3V. It is important to note that I2C lines must never be allowed to exceed 3.3V, doing so would back-drive the processor and alter the power supply's voltage. The internal 3.3V regulator is a precision device which allows the power rails to be used as a reference voltage for analog to digital conversion. Back driving the power supply would cause errors in analog readings or possibly even damage the processor.

I2C operation requires pull-up resistors on both the SCK and SDA lines. These resistors are typically located on user hardware. If the hardware uses pull-up resistors to +5 volts (or anything above 3.3V), some means of limiting SDA and SCK voltages is required. The simple solution is to add a pull-down resistor on the SDA and SCK lines such that the pull-up voltage never exceeds 3.3 volts. For example, a 5K ohm pull-up resistor with a 10K pull-down will only pull up voltages to reach 3.3 volts.

The 3.3V supply from the SunDuino internal 3.3V trickle regulator can be used as a pull-up source but that would increase SLEEP MODE current. If user I2C hardware is operated using the SunDuino regulated 3.3V regulated output, then using that regulated 3.3V output as a pull-up source is suggested. If user software turns off the regulator, the application software should also change the I2C into outputs driven with logic zeros. Driving a logic one into unpowered hardware will draw significant currents.

Examples of using pull-down resistors with I2C operation can be found in the 4-Digit 7-Segment display board design included with SunDuino files. When using red LEDs, this board will operate directly from 3.3V. When using other colors, 5V operation is sometimes required. The I2CTest.c source file (sample code directory) provides an example of where the I2C lines are controlled as outputs when power is turned off.