

Description

The EVK-SH3000 MicroBuddy™ evaluation kit is designed to help an engineer get up to speed quickly and simply on the functions and features of the SH3000 family of MicroBuddy devices. This board and the accompanying software allows the user to configure all the registers in the MicroBuddy, and demonstrate the features described in the MicroBuddy data sheet, including

- ◆ CPU Supervisor
- ◆ Clock Management System
- ◆ Real-Time Support
- ◆ Periodic timer

All of these blocks can be individually configured and updated separately from the other settings. There is also a provision for writing to one register at a time for maximum flexibility.

The communication with the MicroBuddy device is handled by the on-board MCU (a Cypress CY7C63001). This is configured to communicate PC via the USB port standard on all modern computers. The PCB is supplied with two rows of headers, duplicating the pins on the MicroBuddy, allowing the user to connect to a separate MCU to evaluate MicroBuddy in a real-world application. All of the settings are documented completely in the accompanying data sheet and the *SH3000 Reference Manual*.

Features

- ◆ **Highly integrated MicroBuddy IC**
 - 3 mm x 3 mm x 0.9 mm 16-lead MLP (QFN) package
- ◆ **MC68HC705C8A example source code supplied**
 - Complete Send and Receive blocks documented to help write code for any MCU.
- ◆ **Demonstrations**
 - Demonstrate Watchdog, Wakeup Timer, Real-Time Clock, High-Frequency Clock, Low-Frequency Clock, Spread Spectrum and Reset functions from the PC.
 - Programmable clock output from 32.768 kHz to 16.7 MHz in 2048 Hz steps
 - Directly demonstrate microcontroller STOP function
- ◆ **Software evaluation**
 - Windows software supplied to demonstrate MicroBuddy functions
 - All settings programmable in real-time, defaults restored from nonvolatile memory on reset
- ◆ **Powered by USB port, no external power needed**
- ◆ **MicroBuddy IDD <850 μ A/2 MHz, <3 mA/16 MHz, <10 μ A/standby**

Applications

- ◆ Home automation and security
- ◆ Consumer products
- ◆ Portable/handheld computers
- ◆ Industrial equipment
- ◆ Any microcontroller-based product



Semtech, the Semtech logo, MicroBuddy, μ Buddy, and μ B are marks of Semtech Corporation. All other marks belong to their respective owners.

Evaluation Board

The evaluation kit includes the evaluation board shown below. It consists of a PCB with 3 major parts. The first part is towards the middle and is the MicroBuddy DIP adapter. The second part is the Cypress CPU toward the right hand side and labeled U2. The crystal and resistor on the DIP adapter can be omitted in production to allow even smaller real-estate usage. See the *SH3000 Reference Manual* for more details. The final part is the support circuitry comprising the LED indicators D1, D2 and D3 on the left.

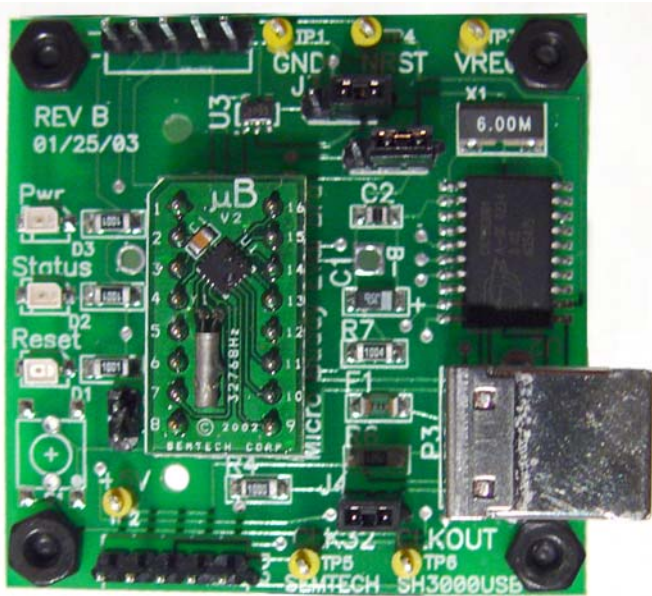


Figure 1. Evaluation board: top view

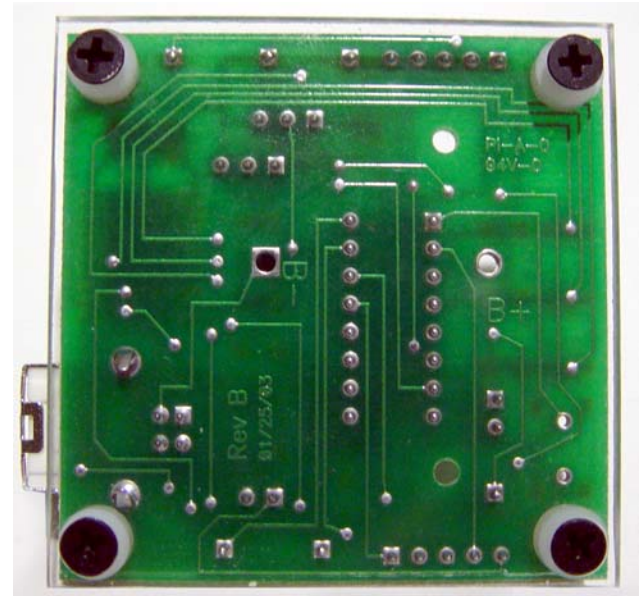


Figure 2. Evaluation board: bottom view

The function of the DIP adaptor is to supply a working, pre-programmed MicroBuddy with crystal and de-coupling already connected, and wired up to an industry standard DIP profile socket. This makes it easy to quickly prototype up a circuit to evaluate MicroBuddy without having to develop a PCB and re-flow the MicroBuddy's small package on the PCB. It can be removed from the EVK-SH3000USB.



Figure 3. DIP adapter board

Using the Evaluation Board

The evaluation board is designed to demonstrate the functions and features of the MicroBuddy product. This can be broken down into several stages, namely Watchdog, Low-Frequency Oscillator (LFO), High-Frequency Oscillator (HFO) and Wakeup Timer (WUT). These terms and functions are explained in more detail in the *SH3000 Reference Manual*, and only a brief introduction is given here on how to demonstrate and observe these functions.

Connecting the board

First, install the software from the CD-ROM in the evaluation kit by double-clicking on the file SETUP.EXE. Follow the directions and reboot the PC if instructed.

Next connect the MicroBuddy board to the system. Wait for the computer to find and identify the new hardware. When asked for the driver, press BROWSE and select the "DRIVER" directory on the CD-ROM enclosed with the kit.

Now find the MicroBuddy shortcut (which should be located in the Semtech folder of the start menu) and double click on it

The software Project Settings screen should now appear on the computer (see the Product Settings section).

Watchdog demonstration

- Press the **CPU Supervisor functions** tab to select the watchdog settings screen.
- Select a value for the **watchdog time** using the slider control.
- Press **Change Now**.

The program pauses while communication takes place between the computer and the evaluation board. The Log window shows the results of the communication that took place and the title bar of the window should go green. If it remains red, there was a problem. Watchdog codes are now sent to the MicroBuddy to prevent the Watchdog from timeout, simulating what a production MCU would do. These codes can be observed in the output log window.

To demonstrate a watchdog timeout, press the **Force Timeout** button on the CPU supervisor functions screen. The causes the Cypress chip to stop updating the watchdog. MicroBuddy then enters reset because of a watchdog timeout. The Reset Status LED flashes, the Status window shows **Reset caused by Watchdog timeout**. The watchdog is now disabled, and can be re-enabled by setting a value on the slider and pressing **Change Now** again.

LFO demonstration

The LFO has several settings that can be demonstrated. First press the **LF Clock** tab to select the Low-Frequency Oscillator settings screen, and set the evaluation board jumper J1 to the XTAL position (towards the MicroBuddy dip-adaptor). The MicroBuddy is now using the 32.768 kHz crystal as a reference. Press the switch button next to pin 13 on the user interface to enable the 32.768 kHz out pin and press **Change Now**. Attaching a scope probe to the CLK32 pin on the evaluation board should show a clean 32.768 kHz signal.

Now change jumper J1 on the evaluation board to the RC position (towards the Cypress chip). The MicroBuddy is now using the internal 32.768 kHz oscillator as a reference. The scope should again show a 32.768 kHz signal. Note there may be some noise on the signal now due to the long thin wires used on the evaluation board. In practice a good solid ground plane would be present and this noise would disappear.

Note: For normal operation, jumper J4 should be connected, and jumper J3 should be in position towards the Cypress chip U2.

HFO demonstration

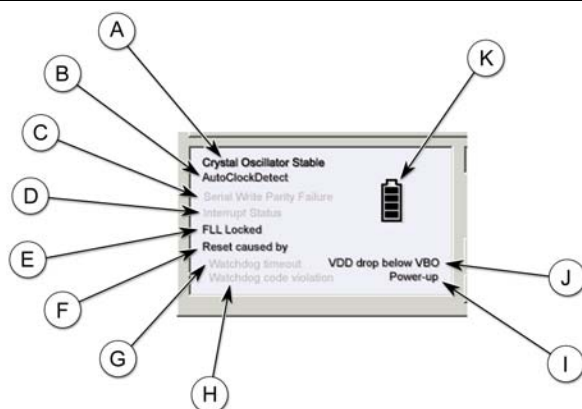
The HFO also has many functions that can be demonstrated. Refer to the *SH3000 Reference Manual* to for the best explanation of these functions. Connect the scope to the CLKout pin, and the clock output can be observed. Note that if the Simulate MCU Clock STOP button is pressed, the Clock disappears, simulating the operation of a HALT or STOP instruction.

To change the frequency, type a value in the **Autoset** window and press the **Help me set values** button. This configures the settings with the closest match for you. Pressing **Change Now** updates the frequency, which should be visible on the scope.

WUT demonstration

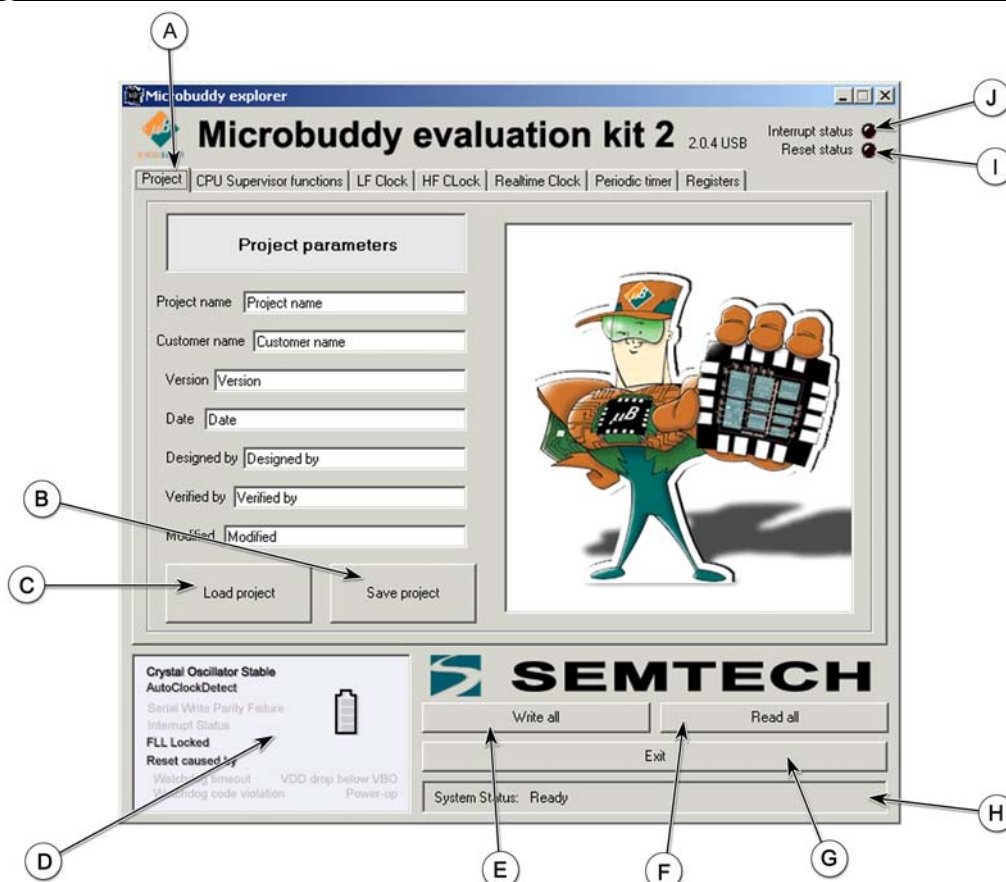
The wakeup timer is the last block to demonstrate. To do this, select the **Periodic Timer** tab and choose a value of time using the value box and hours, minutes, seconds, etc. settings to select a time to test. Press **Change Now** to set these values to the MicroBuddy.

The LED on the evaluation board flashes every time an interrupt is created. If a value of 500 mS or greater is selected, this LED flash is easily visible. The outputlog also shows the action of the Cypress MCU clearing the interrupt status register after each interrupt, simulating the action of a in-circuit MCU attached to MicroBuddy.

Status Window


This is the Status window. It appears when the program is started and stays on the screen as the user change tabs. It displays the status of the MicroBuddy at the point when the **Read all** button was pressed. It decodes the two status bytes **cause of reset** and **status** and puts them on the screen in an easy-to-understand format.

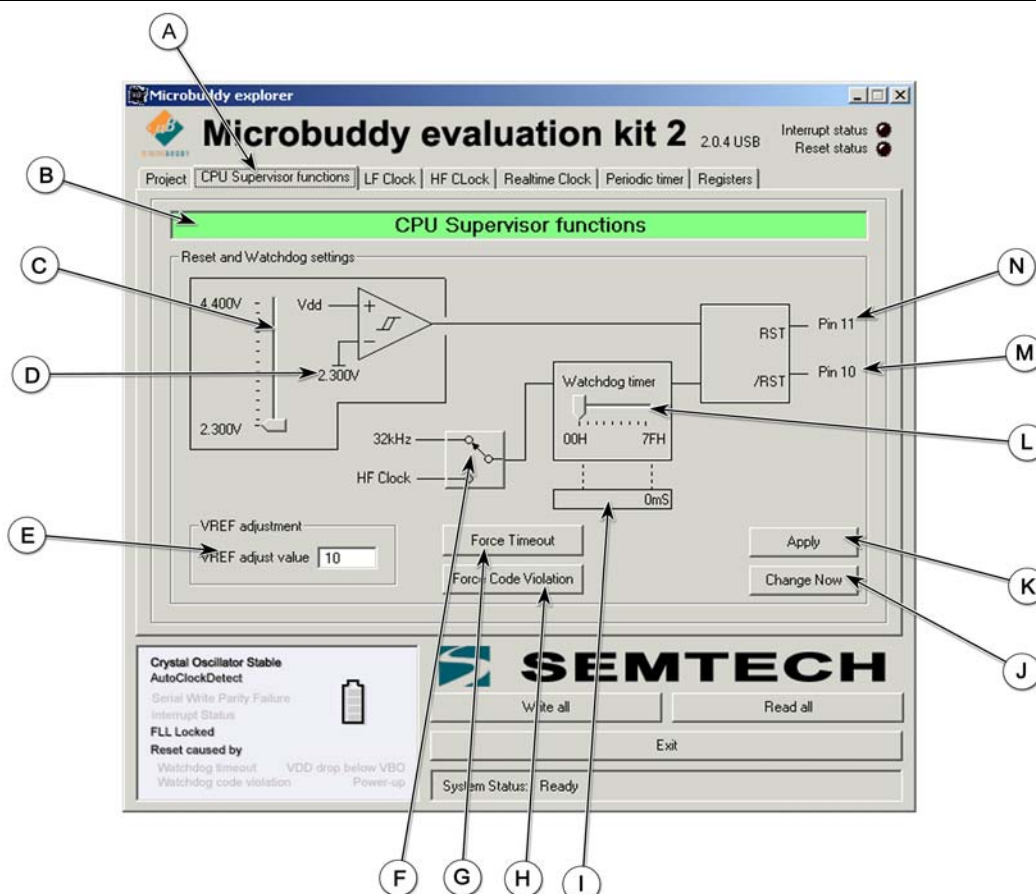
- A. **Crystal Oscillator stable** – If this label is highlighted, it indicates that the crystal oscillator is stable. If the MicroBuddy is configured to use the internal oscillator with or without the external reference resistor, this label is not highlighted.
- B. **AutoClockDetect** – This label indicates that the MicroBuddy has detected a clock on the CLKIN pin and is primed to detect the loss of the MCU clock and turn the HF clock output off. For a more detailed description of this function, please refer to the *SH3000 Reference Manual*.
- C. **Serial Write Parity Failure** – This label indicates that the communication between the MCU and MicroBuddy has a problem, and a parity failure occurred. As the MCU is communicating directly with the MicroBuddy and derives its clock source from the MicroBuddy, this is extremely unlikely ever to occur, and the user should never see this label highlighted.
- D. **Interrupt Status** – This label indicates that an interrupt has been generated by the MicroBuddy and has not yet been cleared by the MCU. As the MCU on the evaluation board uses the interrupt settings in its normal mode this label may also never be seen highlighted.
- E. **FLL Locked** – This label indicates that the FLL block internal to the HF oscillator is enabled, running and locked to the frequency that the user programmed. By default, the evaluation board operates at 2 MHz.
- F. **Reset caused by** – This label is the title for the labels underneath (G and H).
- G. **Watchdog timeout** – If this label is highlighted, then the MicroBuddy was most recently reset by the watchdog timeout. See “Using the evaluation board” elsewhere in this document.
- H. **Watchdog code violation** - If this label is highlighted, the MicroBuddy was most recently reset by a violation in the watchdog code sequence. See the *SH3000 Reference Manual* for a more detailed explanation.

Project Settings


This is the project settings window. It is the first window to be seen when the program starts up, and allows the user to record details of the settings used for a particular session. These settings could reflect a particular product or customer, or can be ignored if used to explore the functions of the MicroBuddy.

- A. **Project** – The tab that displays this window.
- B. **save project** – Saves all the MicroBuddy register settings to a text file for later retrieval.
- C. **Load Project** – Loads all the MicroBuddy register settings from a text file saved earlier.
- D. This is a graphical representation of the status of MicroBuddy, showing the various values in the Status and Cause of reset registers – see the *SH3000 Reference Manual* for a register list. This status view is detailed more clearly on the next page.
- E. **write all** – Transfers all the register values from this program to the attached evaluation board. Care must be taken to ensure that these values are valid. When the program is first started, press **Read all** first to read the current register values from the MicroBuddy.
- F. **Read all** – Transfers all the register values currently stored in the MicroBuddy to this program. All the functional settings in the following pages are updated at this time.
- G. **Exit** – Closes and exits the program.
- H. **Status bar** – Shows the current status of the software
- I. **Reset status** – This LED indicates the RESET state of the MicroBuddy. When this LED is RED it indicates that the MicroBuddy reset line is active. This LED typically flashes on then off during a normal RESET event.

- J. **Interrupt status** – This LED indicates the Interrupt status. If this LED is RED it indicates that the MicroBuddy has signaled an interrupt. This LED turns off when the interrupt event is cleared.

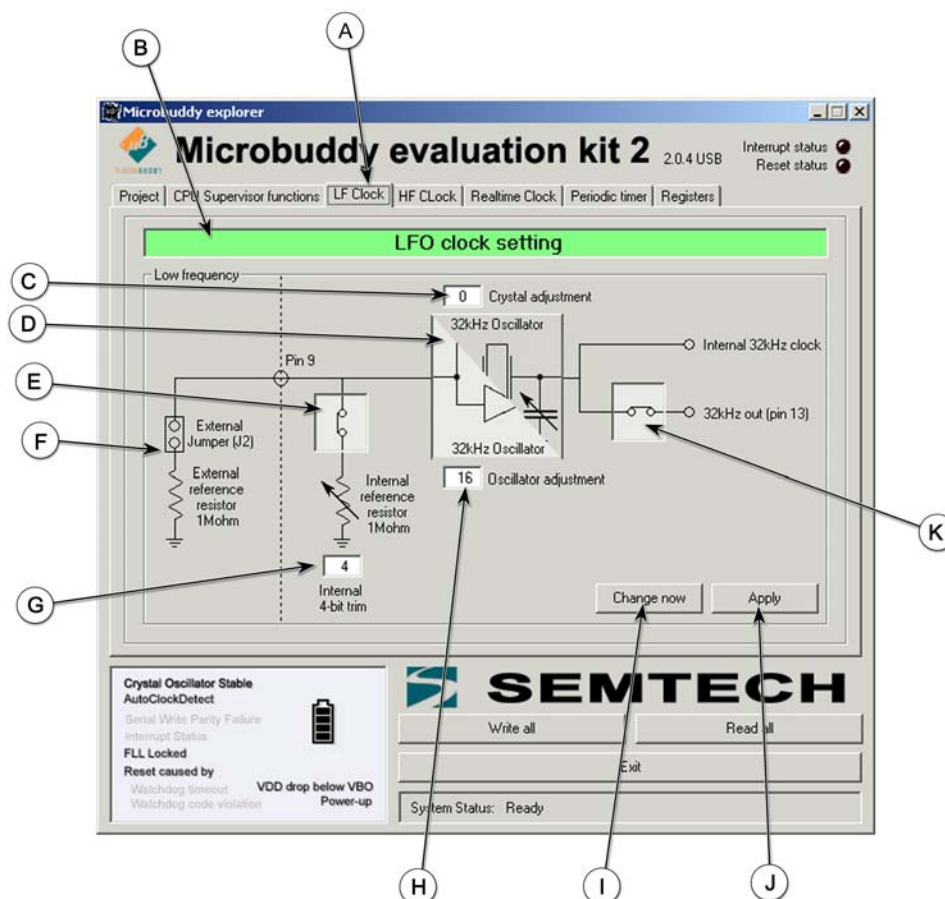
CPU Supervisor Functions


This is the CPU Supervisor function window. This window allows the user to configure the low-voltage reset and watchdog timer blocks of the MicroBuddy.

- A. **CPU supervisor functions** – The tab that displays this window.
- B. The title bar changes color to indicate the status of the settings. A green bar indicates that the settings shown are the settings described in the registers table. When the user changes a setting on this page, the bar turns red, indicating that the settings are different from the register table. Pressing **Apply** or **Change now** updates the register table with the new settings and changes the bar back to green.
- C. **Reset voltage setting** – This slider configures the reset voltage of the reset circuit. The limits are set to 2.3 V and 4.4 V. The user can either drag the slider, or click on the bar above/below to increment/decrement by one step (approximately 34 mV).
- D. **Reset voltage** – Actual value set by the slider (C).
- E. **Bandgap adjustment** – Used in the calibration of the internal reference voltage for the reset circuit. It should not be changed by the user. See the *SH3000 Reference Manual* for a more detailed explanation of this value.
- F. **Watchdog source select** – Allows the user to select 32.768 kHz or HF clock as the clock source to the watchdog circuit. If the HF clock is selected, the watchdog runs at a faster rate, but is also suspended during clock stop operations, preventing false re-starts.
- G. **Force Timeout** – When the Watchdog timer is enabled, this button forces the program to stop sending codes to the MicroBuddy to prevent Watchdog timer timeout. RESET then occurs and the status windows shows “Reset caused by Watchdog Timeout.”

SYSTEM MANAGEMENT**Preliminary**

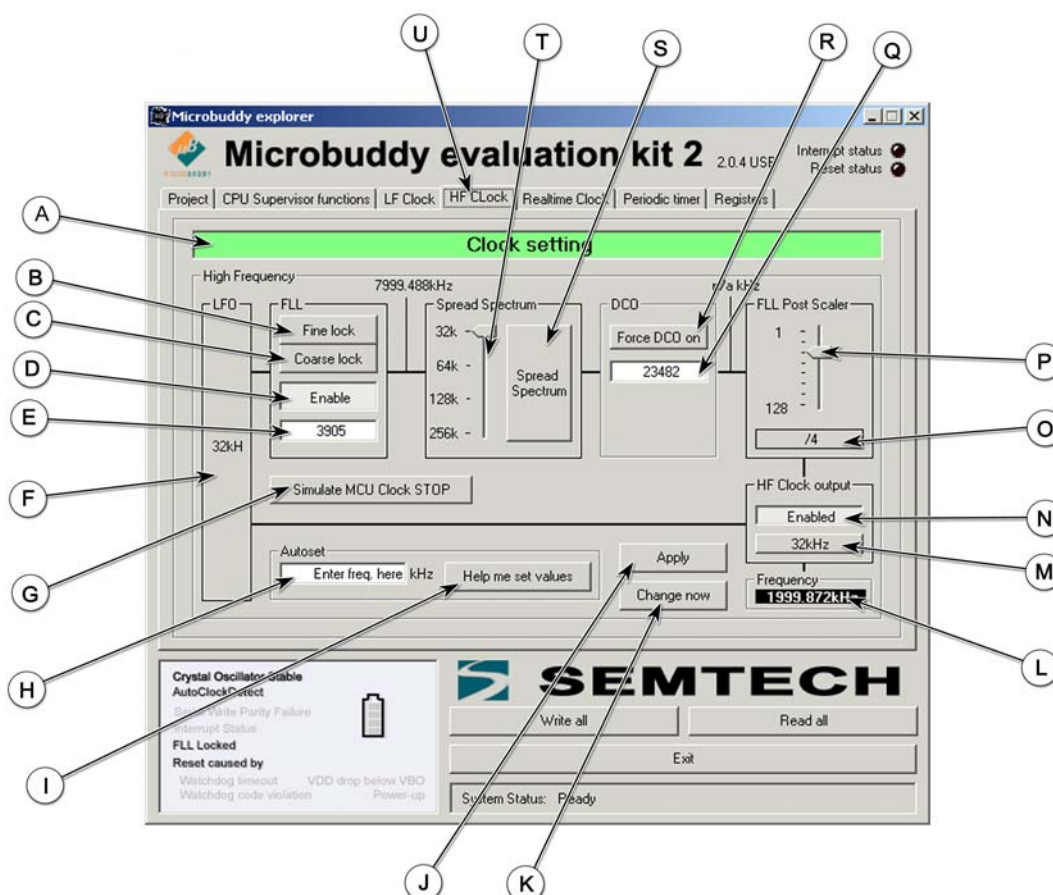
- H. **Force Code Violation** – When the Watchdog timer is enabled, this button forces the program to send a wrong codes to the MicroBuddy. RESET then occurs and the status window shows “Reset caused by Watchdog code violation.”
- I. **Watchdog value** – Effective watchdog time period within which the MCU must send a valid code to prevent reset occurring.
- J. **Change Now** – Updates the register table with the settings set in this window, and updates the MicroBuddy immediately with these values.
- K. **Apply** – Updates the register table only with the settings set in this window. By pressing the **Registers** tab, the user can see which registers are affected by this block.
- L. **Watchdog Timer** – This slider allows the user to set the watchdog timeout time.
- M. This is a diagrammatic representation of pin 10 of the MicroBuddy. Pin 10 creates an Output_Low reset signal.
- N. This is a diagrammatic representation of pin 11 of the MicroBuddy. Pin 11 creates an Output_High reset signal.

LFO Clock Setting


This is the LFO clock settings window. This window allows the user to configure the Low-Frequency Clock module, which is used internally to create 32.768 kHz as a reference for all the internal functions of the MicroBuddy. The accuracy of the MicroBuddy system clock crucially depends on these settings.

- A. **LF Clock** – The tab that displays this window.
- B. The title bar changes color to indicate the status of the settings. A green bar indicates that the settings shown are the settings described in the registers table. When the user changes a setting in this window, the bar turns red, indicating that the settings are different from the register table. Pressing **Apply** or **Change now** updates the register table with the new settings and changes the bar back to green.
- C. **Crystal adjustment** – Allows the user to adjust the padding capacitance presented to the crystal. This is useful to adjust for temperature compensation for ambient operating temperatures outside of the usual operating temperature. See the *SH3000 Reference Manual* for a more detailed explanation.
- D. **32.768 kHz oscillator** – Displays the current operating configuration of the LF oscillator. The RC oscillator mode is shown, with the crystal oscillator mode half covering it. This status is determined by the MicroBuddy hardware. There is no setting for this in software.
- E. **Internal reference resistor disconnect** – This switch is used to switch out the internal 1 MΩ reference resistor when using the external reference resistor (F).
- F. **Jumper J2** – This is a graphic representation of jumper J2 on the evaluation board used to connect the external reference resistor.

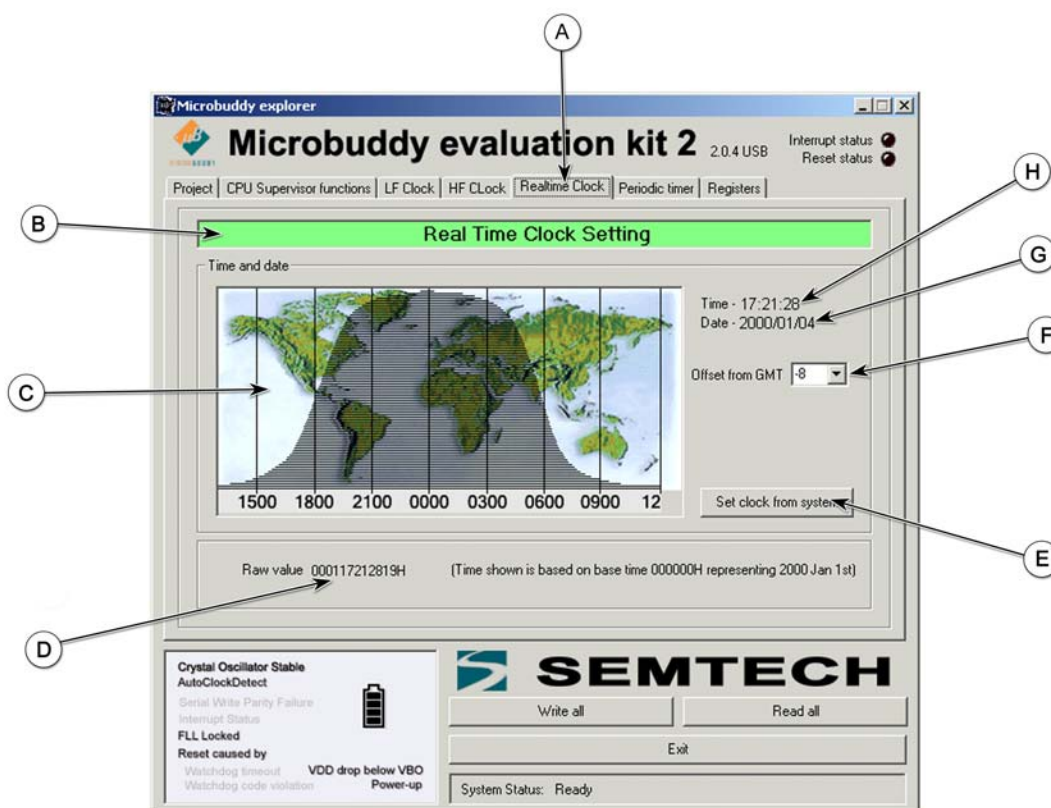
- G. This is the 4-bit internal reference resistor trim. It allows the user to calibrate the RC oscillator to exactly 32.768 kHz based on the internal reference resistor. This is a default value set at the factory and should not be changed, unless the user wants to re-calibrate the RC oscillator because of some difference in operating conditions from the factory conditions.
- H. **Oscillator adjustment** – This adjustment is the 6 bit internal oscillator adjust value. See *SH3000 Reference Manual* for more information.
- I. **Change Now** – This button updates the register table with the settings set in this window, and updates the MicroBuddy immediately with these values.
- J. **Apply** – This button updates the register table only with the settings set in this window. By pressing the **Registers** tab, the user can see which registers are affected by this block.
- K. This switch enables the internal 32.768 kHz reference signal to be output to an external pin on the MicroBuddy (J). This signal can be used for other low-power devices, but takes power from the system as the clock runs even when power has been removed by using VBAK as a power source. See the *SH3000 Reference Manual* for more details.

HF Clock Settings


This is the HF clock settings window. This window allows the user to configure the High-Frequency Clock module. The high-frequency clock is used to generate the output clock frequency that is sent directly to the attached CPU.

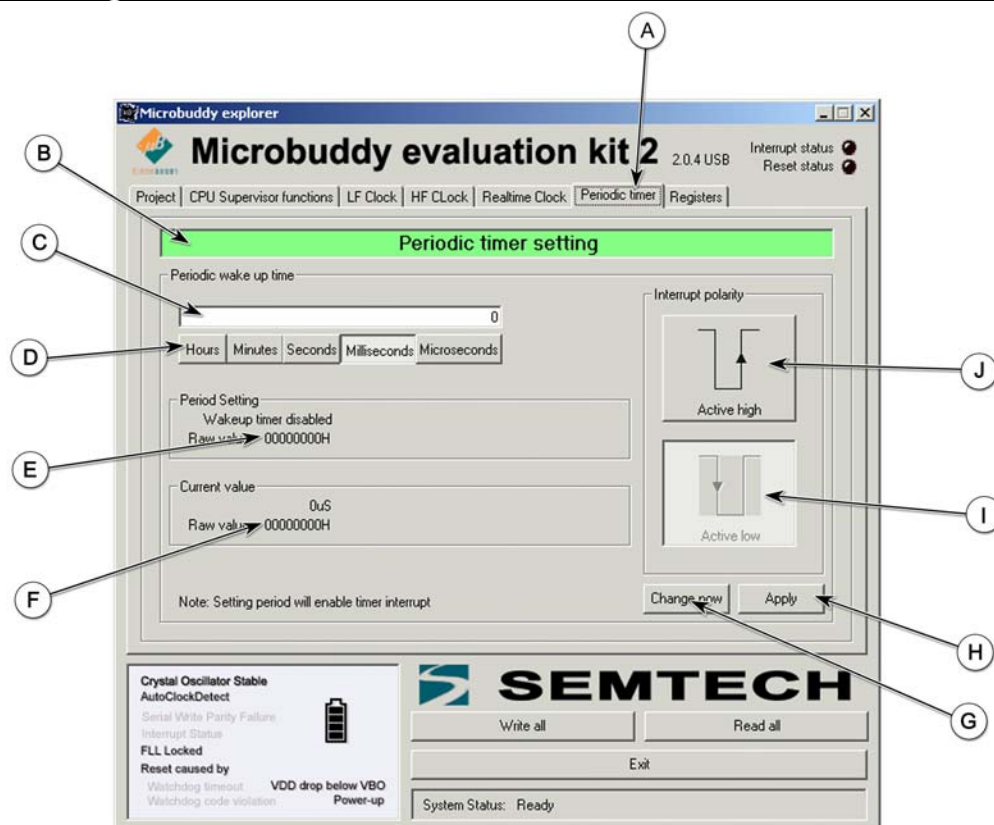
- A. The title bar changes color to indicate the status of the settings. A green bar indicates that the settings shown are the settings described in the registers table. When the user changes a setting on this page, the bar turns red, indicating that the settings are different from the register table. Pressing **Apply** or **Change now** updates the register table with the new settings and changes the bar back to green.
- B. **FLL Block, Fine lock** - Selects the Fine-Lock setting. If the frequency step is small (less than 256 kHz at the undivided output of the HF oscillator), the host may set this bit. The MicroBuddy then performs a successive approximation algorithm on the 7 least significant bits of the 18-bit DCO Code value, and finds a locked setting in approximately 5 ms. The clock may experience the maximum frequency fluctuations of only 3.2%.
- C. **FLL Block, Coarse lock** - Selects the Coarse-Lock setting. When the frequency of the FLL is changed, the MicroBuddy performs a successive approximation algorithm on the 18-bit DCO (digitally controlled oscillator) Code value, and finds a locked setting in approximately 25 ms. The clock may experience frequency fluctuations of up to 2:1. If neither the fine-freq-lock nor the coarse-freq-lock have been set, then the FLL locks slowly to the target frequency.
- D. **FLL Block, Enable** - Enables the FLL block. This allows the output clock frequency of the MicroBuddy to be under positive control, meaning that the absolute accuracy of the output clock is directly related to the absolute accuracy of the FLL source, which is the internal 32.768 kHz clock from the LFO block.

- E. **FLL Block, FLL Divide ratio** – Target frequency for the HF oscillator. Once enabled the FLL adjusts the DCO block to match this value. This frequency is determined by the equation $\text{Frequency} = (\text{FLL_Divide_ratio} + 1) \times (\text{LFO_value} / 16)$, which generally would be $f = (\text{FLL_Divide_Ratio} + 1) \times 2048$
- F. This is a diagrammatic representation of the LF-block-generated internal 32.768 kHz reference frequency, used as a reference for the HF clock.
- G. **simulate MCU Clock stop** – This button tells the Cypress MCU on the evaluation board to stop output of clock cycles to the CLKIN pin on the MicroBuddy. This causes the MicroBuddy to enter standby mode. This demonstrates the host MCU HALT or STOP functions.
- H. **Autoset block, frequency assist** – This value can be used by the user to select a target frequency for CLKOUT. This value along with the set button (H) allows the user who may be confused by a large array of options and settings to quickly set up a desired frequency and to see what settings would result in the closest match. This value represents kHz, and can vary from 32.768 kHz to 4 MHz.
- I. **Autoset block, Help me set values** – This button is used with the frequency assist value (G) in order to help users confused with the number of controls to easily set their desired frequency.
- J. **Apply** – Updates the register table only with the settings set on this page. By pressing the **Registers** tab, the user can see which registers this block affects.
- K. **Change Now** – Updates the register table with the settings set in this window, and updates the MicroBuddy immediately with these values.
- L. **Frequency** – Resultant frequency output from the MicroBuddy's CLKOUT pin.
- M. **HF Clock output block, 32.768 kHz** – This button allows the system to force the MicroBuddy's CLKOUT pin to 32.768 kHz. This can be used if the user wants to change the HF clock settings, but the system CPU cannot withstand the clock variations that occur when Coarse Lock (C) is enabled.
- N. **HF Clock output block, Enabled** – This button allows the system to manually enable or disable the CLKOUT pin of the MicroBuddy.
- O. **FLL Post Scaler block, value** – Value of the post scale divider block, which divides the internal HF frequency into the clock frequency that is output on the CLKOUT pin of the MicroBuddy.
- P. **FLL Post Scaler block, setting** – This setting adjusts the amount of division applied to the HF clock signal (N).
- Q. **DCO block, DCO value** – Current value of the DCO (Digitally controlled oscillator). When the FLL is not enabled, this is the source of the HF signal, its range is from 0x00000 to 0x1FFFF [again, this represents a frequency derived by DCO Value x 2048]. This value is written into nonvolatile memory at the factory to provide an “instant on” frequency of 2 MHz at initial power-up. This value can be changed, but is only useful if the FLL is disabled. The frequency derived from this block alone is in open-loop mode with no feedback and a consequently large tolerance in absolute accuracy. The FLL changes this value to achieve the target value in its block, and also locks this value to the accuracy of the internal 32.768 kHz oscillator.
- R. **DCO block, Force DCO on** – Forces the DCO to turn on immediately after power up.
- S. **Spread Spectrum block, Spread Spectrum enable** – Enables the spread spectrum circuit to reduce EMI. Please refer to the *SH3000 Reference Manual* for more details.
- T. **Spread Spectrum block, Spread Spectrum value** – This value is the amount of spread spectrum noise added to the HF frequency. The higher this value is set, the lower the EMI emission, but the greater the jitter on the clock frequency. The values labeled here represent the amount of frequency change to be applied to the HF clock, which is then divided down by the **FLL Post Scaler**.
- U. **HF Clock** – The tab that displays this window.

Real-Time Clock Settings


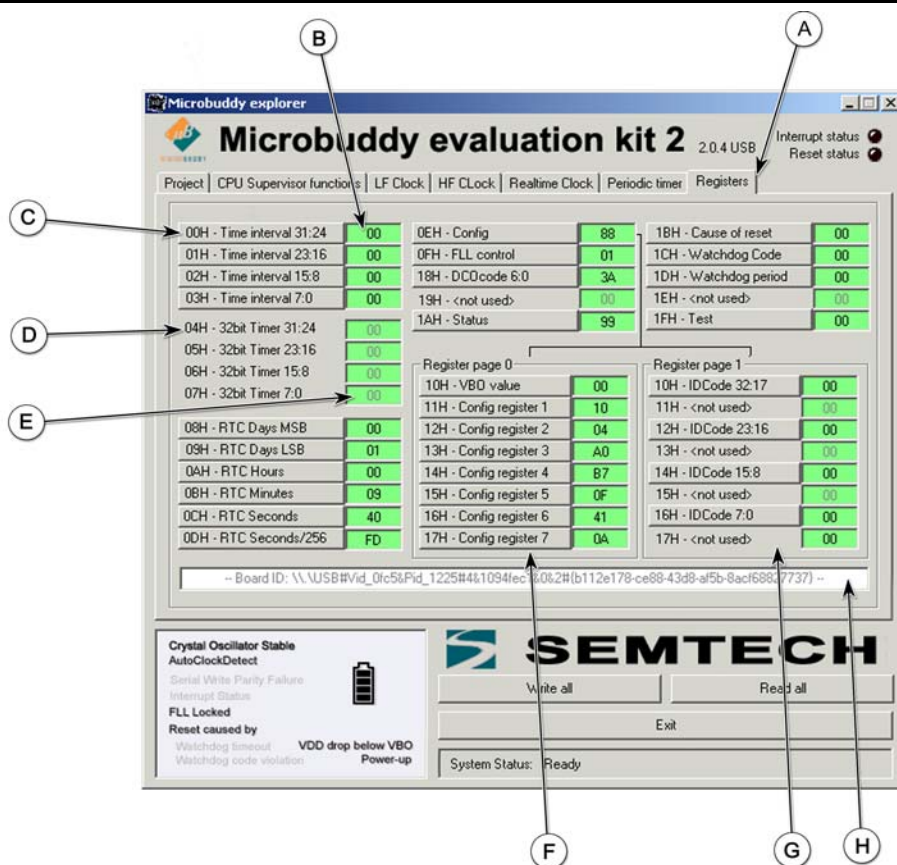
This is the real-time clock settings window. This window allows the user to configure the Real-Time Clock module.

- A. **Realtime Clock** – This is the tab that displays this window.
- B. The title bar changes color to indicate the status of the settings. A green bar indicates that the settings shown are the settings described in the registers table. When the user changes a setting in this window, this bar turns red, indicating that the settings are different from the register table. Pressing **Apply** or **Change now** updates the register table with the new settings and changes the color back to green.
- C. **World map** – Shows the world map and the part of the world illuminated by the sun. The shaded region representing night scrolls across the world map according to the change in time when the Set clock from system (G) button is pressed.
- D. **Raw Value** – Raw hexadecimal value of the real-time clock registers. Note that as the RTC calendar counts days only, the date programmed into the MicroBuddy is based on a zero hexadecimal value representing January the first, 2000 AD.
- E. **Set clock from system** – Sets the MicroBuddy RTC values to be the same as the values on the host computer where this program is running.
- F. **Offset from GMT** – Signed difference in hours between the user's time (H) and Greenwich Mean Time (GMT). When this value is correct, the shadow in the world map window (C) is displayed correctly.
- G. **Date** – Date set in the RTC registers.
- H. **Time** – Time set in the RTC registers.

Periodic Timer Settings


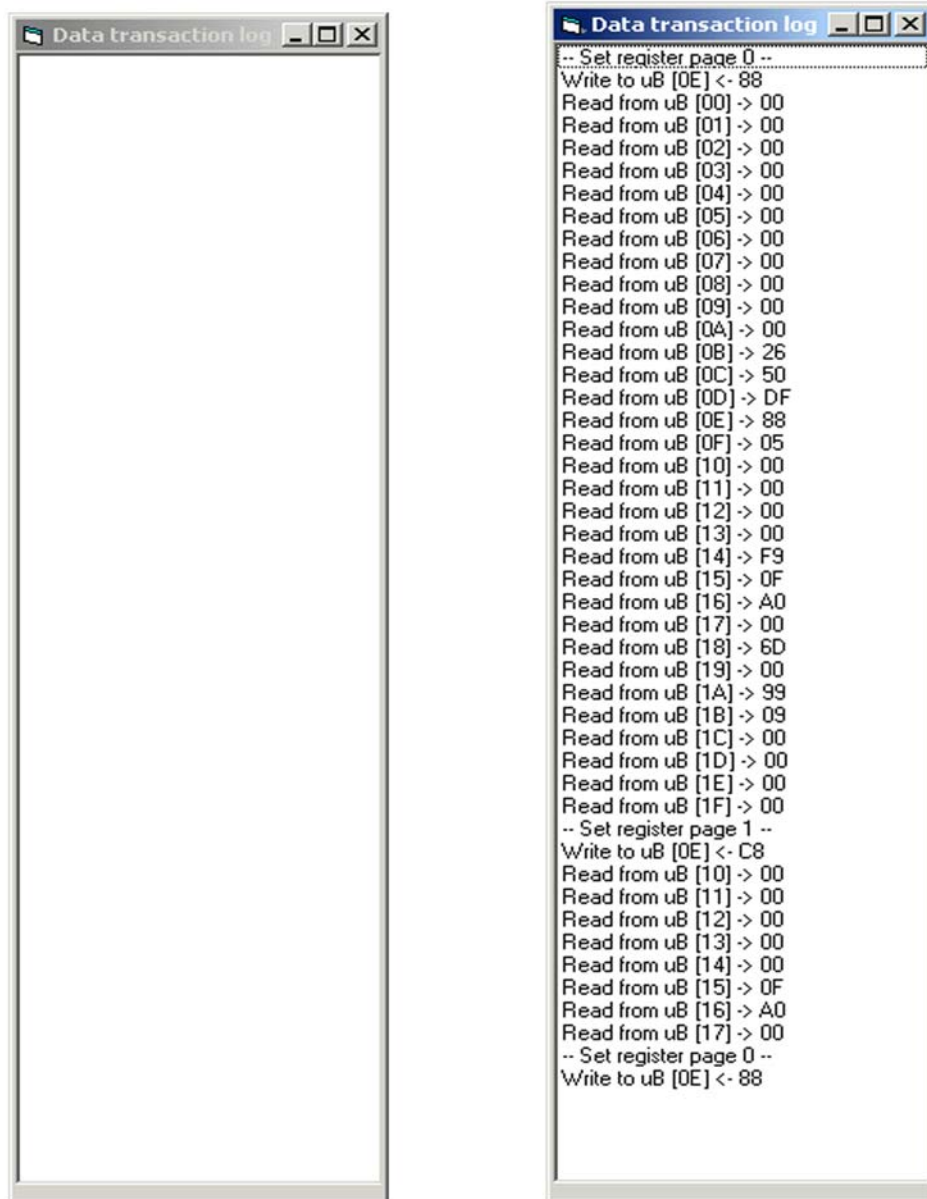
This is the periodic timer settings window.

- A. **Periodic timer** – The tab that displays this window.
- B. The title bar changes color to indicate the status of the settings. A green bar indicates that the settings shown are the settings described in the registers table. When the user changes a setting in this window, the bar turns red, indicating that the settings are different from the register table. Pressing **Apply** or **Change now** updates the register table with the new settings and changes the bar back to green.
- C. **Wakeup time value** – This is the value of time that the interval timer uses to generate an interrupt to the system.
- D. **Hours, Minutes, Seconds, Milliseconds, and Microseconds** – These buttons, when used with the wakeup time value (C), represent the time interval between interrupts.
- E. **Period setting, Raw value** – Raw hexadecimal value of the periodic time interval setting.
- F. **Current value, Raw value** – Raw hexadecimal value of the current periodic time interval as sampled from the MicroBuddy.
- G. **Change Now** – Updates the register table with the settings set on this window, and updates the MicroBuddy immediately with these values.
- H. **Apply** – Updates the register table only with the settings set on this window. By pressing the **Registers** tab, the user can see which registers are affected by this block.
- I. **Active low** – Sets the interrupt level from the MicroBuddy to be configured as active low.
- J. **Active high** – Sets the interrupt level from the MicroBuddy to be configured as active high.

MicroBuddy Registers


This is the MicroBuddy register display window. It allows the user to see all the contents of all the registers of the MicroBuddy at a glance.

- A. **Registers** – This is the tab that displays this window.
- B. Raw register value for (C). The background color is green when the value in this window corresponds with the value in the MicroBuddy, and red when a setting has changed in the previous windows and **Apply** has been pressed, but the MicroBuddy has not been updated by either a **Change now** operation, or a **Write all** operation. The user can change any value at any time by typing in a 2-digit hexadecimal value, and can update the register individually by pressing the adjacent button (C).
- C. Register label, the contents of which are shown in the adjacent box (B). By changing the value in (B) and pressing this button, the value is immediately updated to the MicroBuddy. If this update was successful, the value of (B) returns to green. If an error occurred, the value turns red. See (B) description.
- D. Read only register label, the contents of which are shown in the adjacent box (E). The value this time cannot be updated, and there is no button to update this value manually. It is updated by the **Read all** operation only.
- E. Raw register value for a read only register. See (D)
- F. Same as (C), but in Register page 0 – see the *SH3000 Reference Manual* for more details
- G. Same as (C), but in Register page 1 – see the *SH3000 Reference Manual* for more details
- H. USB ID – The Board ID as reported in the USB descriptor from the on-board Cypress MCU

Data Transaction Log


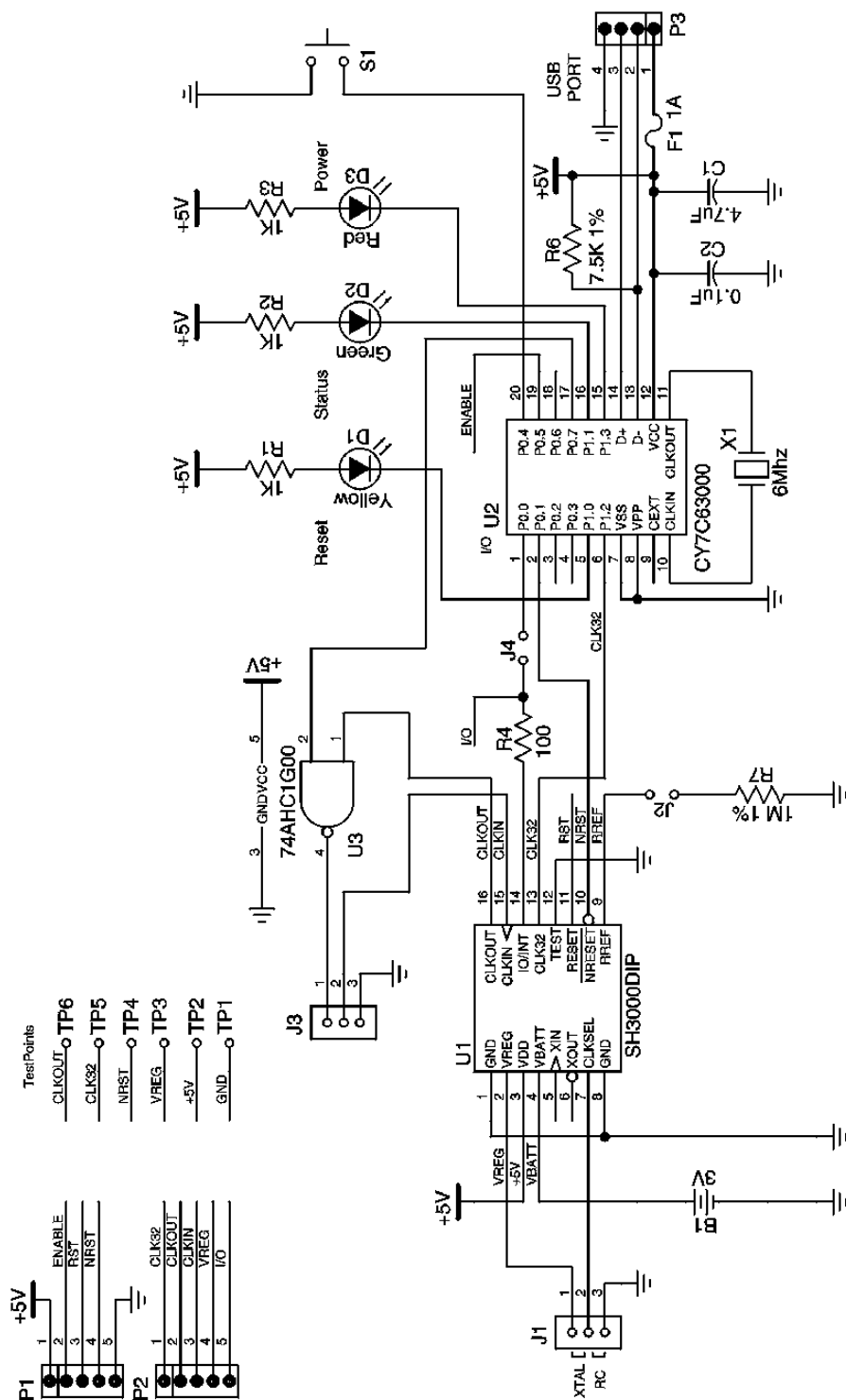
Any time that data is sent to the MicroBuddy, a record is kept in a log window. This can be one of three types of action. Write to μB , Read to μB , and <Ping>.

Write to μB

This action is recorded when the software writes a value to a MicroBuddy register. The format is *Write to μB [aa] \leftarrow bb* where aa is the register address, and bb is the data written.

Read from μB

In a similar way, this action records when the software reads a value from a MicroBuddy register. The format is *Read from μB [aa] \rightarrow bb* where aa is the register address, and bb is the data read.

Evaluation Board Schematic


**For sales information and product literature,
contact:**

Semtech Corporation
Human Interface Device (HID)
and System Management Division
200 Flynn Road
Camarillo, CA 93012-8790
sales@semtech.com
<http://www.semtech.com/>
(805)498-2111 Telephone
(805)498-3804 Fax

Copyright ©2002-2003 Semtech Corporation. All rights reserved. Semtech, the Semtech logo, MicroBuddy, µBuddy, and µB are marks of Semtech Corporation. All other marks belong to their respective owners.

LIMITED LICENSE GRANTED: NO WARRANTIES MADE

This specification is provided "as is" with no warranties whatsoever including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification or sample. Any suggestions or comments by Semtech concerning use of this product are opinion only, and Semtech makes no warranty as to results to be obtained in any specific application. A license is hereby granted to reproduce and distribute this specification for internal use only. No other license, expressed or implied to any other intellectual property rights is granted or intended hereby. Authors of this specification disclaim any liability, including liability for infringement of proprietary rights, relating to the implementation of information in this specification. Authors of this specification also do not warrant or represent that such implementation(s) will not infringe such rights.