Si7013EVB-UDP/Si7013EVB-UDP-F960 User's Guide

1. Introduction

This user's guide describes the hardware and software included with the Si7013EVB-UDP and Si7013EVB-UDP-F960 evaluation kits. The Si7013EB-UDP port header card contains an Si7013 sensor. The port header card plugs into a Unified Development Platform (UDP) C8051F960/Si1020 MCU card with Multiplexed LCD (UPMP-F960-MLCD).

This user's guide is included in two evaluation kits:

- Si7013EVB-UDP
- Si7013EVB-UDP-F960

Note: Si7013EB-UDP refers to the board. Si7013EVB-UDP refers to the kit.

The Si7013EVB-UDP-F960 kit contains a port header card and an MCU card. The Si7013EVB-UDP kit contains only the port header card. The Si7013EVB-UDP kit is for customers who already have the MCU card from the C8051F960-A-DK kit.

2. Hardware Description

The Si7013 temperature and humidity sensor uses standard I^2C protocol which is a two wire bidirectional communication protocol with control signals SCL and SDA. For more information about the Si7013, refer to the Si7013 data sheet. Temperature can be read from the internal temperature sensor or from an external thermistor.

Figure 1 shows the Si7013EB-UDP port card and Figure 2 shows the Si7013EB-UDP plugged into a C8051F960/Si1020 MCU card (UPMP-F960-MLCD). As will be explained in more detail below, the Si7013EB-UDP can be used with other Silicon Laboratories unified development platforms.

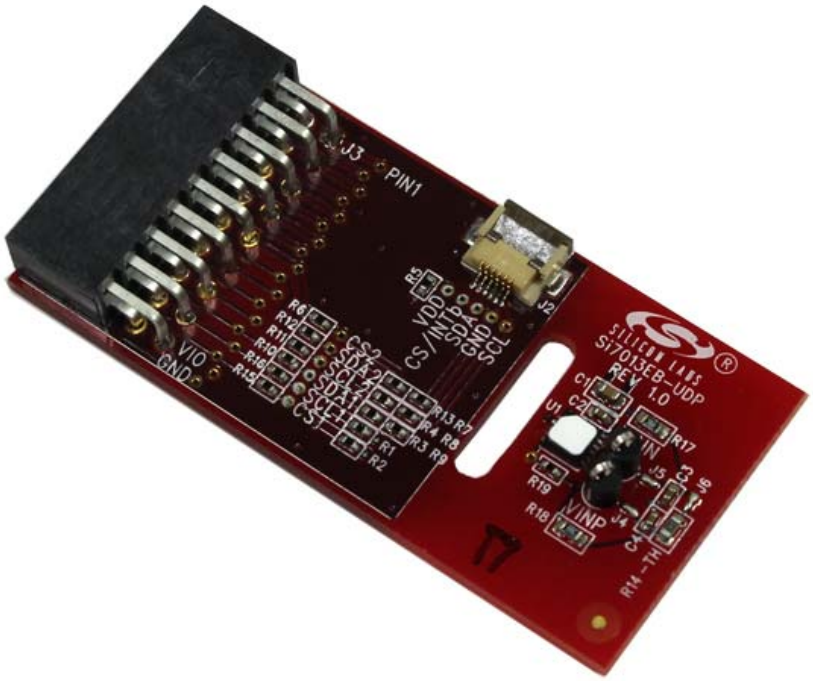


Figure 1. Si7013EB-UDP Port Header Card

Copyright © 2013 by Silicon Laboratories

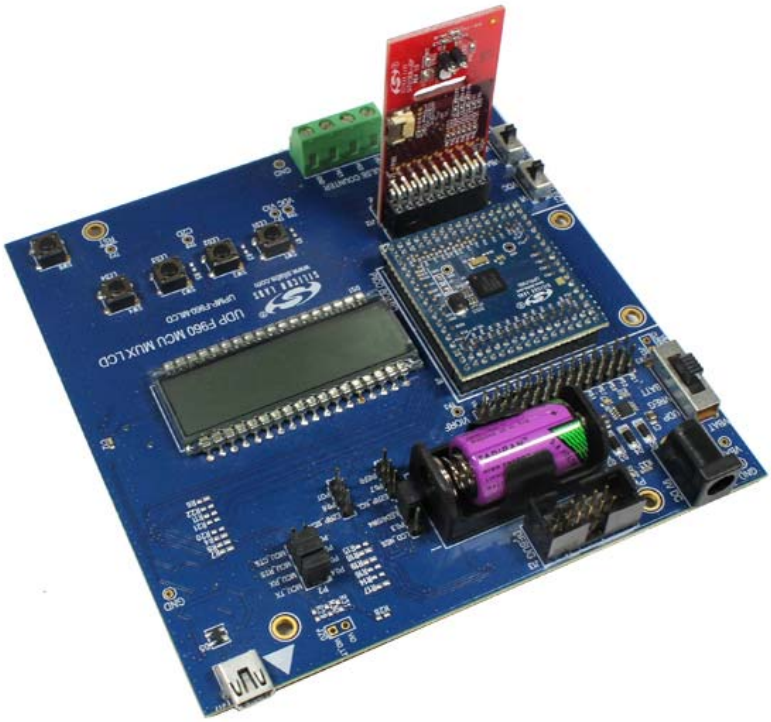


Figure 2. Si7013EB-UDP Port Header Card Plugged into a UDP C8051F960/Si1020 MCU Card with Multiplexed LCD

2.1. Si7013EB—UDP Schematics and BOM

Figure 3 shows the schematics of the Si7013EB-UDP port header card. VDD is connected to port pin 17 and GND is connected to port pin 16. This is the standardized location for power and ground on Silicon Laboratories UDP port headers. For port headers with only 8 port pins, the card can be offset with power and ground pins connecting properly and pins 1–8 of the Si7013EB-UDP overhanging.

By default, the on board Si7013 (U1) connects SDA to port pin P1.5, SCL to P1.6 and CS to P1.7. There is an optional 6 pin flat flexible header connector J2 that can connect to an external “postage stamp” size evaluation board for Si7013, Si7020, or Si7021. The “postage stamp” size evaluation boards and a 3-foot flat flexible cable are included with the Si7013USB-DONGLE evaluation kit (ordered separately). The optional Si7013 postage-stamp-size EVB would have SDA connect to port pin 1.3 SCL to P1.4 and CS to P1.2. The BOM for the Si7013EB-UDP is shown in Table 1.

Table 1. Si7013-UDP Bill of Materials

Qty	Ref	Value	Rating	Tol	Type	PCB Footprint	Mfr Part Number	Mfr
1	C1	4.7uF	6.3 V	±20%	X5R	C0603	C0603X5R6R3-475M	Venkel
3	C2, 3, 4	0.1uF	10 V	±20%	X7R	C0402	C0402X7R160-104M	Venkel
1	J2	FH12			FH12	CONN6N-FPC/ FFC-P0.5	FH12-6S-0.5SH(55)	Hirose
1	J3	CONN SOCKET 2x9			Socket		SSW-109-02-F-D-RA	Samtec
3	R1, R2, R13	10K	1/16 W	±5%	ThickFilm	R0402	CR0402-16W-1002J	Venkel
2	R3, R4	2K	1/16 W	±5%	ThickFilm	R0402	CR0402-16W-2001J	Venkel
10	R5, R6, R7, R8, R9, R10, R11, R12, R15, R16, R19	0	1 A		ThickFilm	R0402	CR0402-16W-000	Venkel
1	U1	Si7013			Humidity	QFN24 4x4P0.5	Si7013-A-GM1	Silicon Labs

2.2. Si7013EB-UDP Schematic

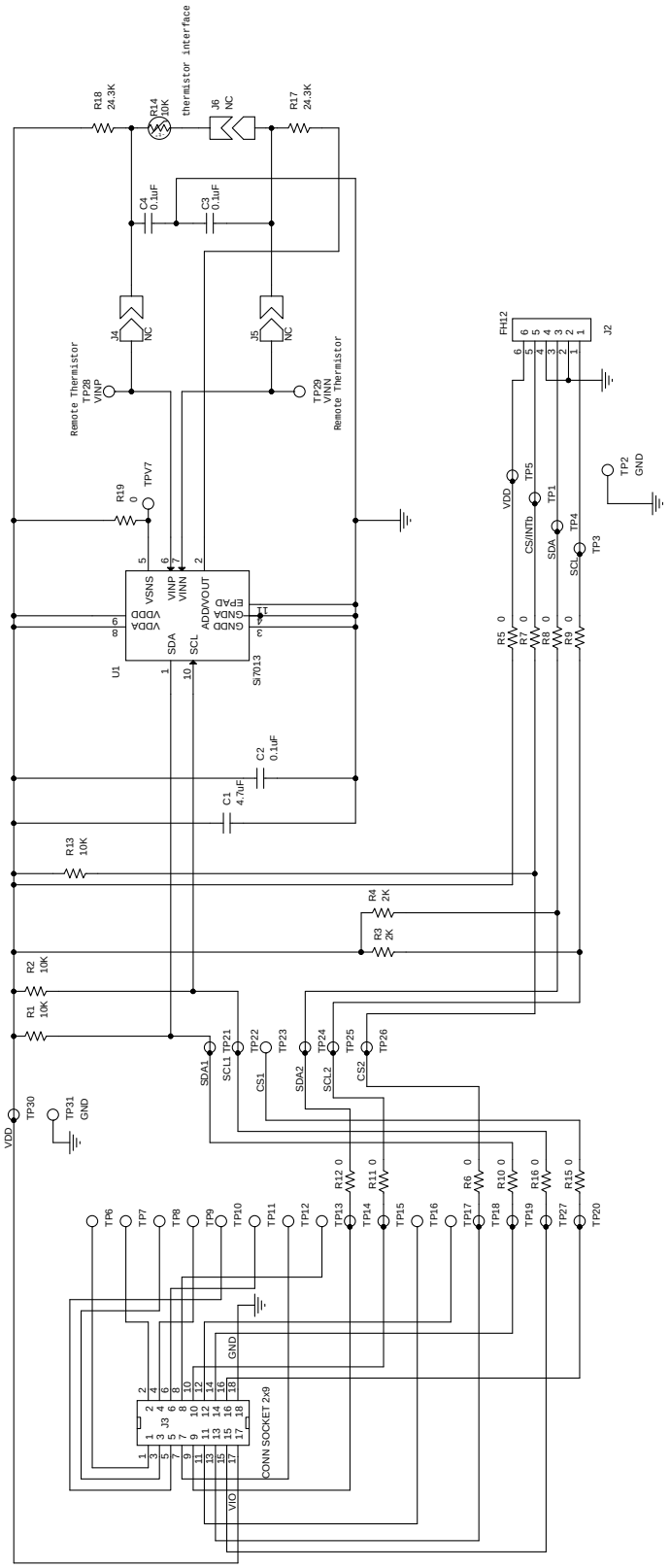


Figure 3. Si7013EB-UDP Schematic

2.3. Extending Battery Life

The data logger demo will work for several days on a fresh battery once disconnected from the PC. At the default sample rate of once per two minutes, the 8196 point storage buffer will fill at about 11.4 days; so, generally, the buffer will not fill before the battery runs out.

In a real application where it might be desirable to extend battery life and record data for longer time periods, the following steps are recommended:

1. Reduce the interval between samples so the buffer will not fill during the time period desired.
2. Although powered down, the USB interface and level translators (U4, U5, and U63 of the C8051F960/Si1020 MCU Card with Multiplexed LCD MCU) consume power and are not needed in operation. (See Figure 10 of the UPMP-F960-MLCD User's Guide.) These can be disconnected from the power supply except when the data log is being read or configured.

3. Software Description

The Si7013 evaluation kit (Si7013EVB-UDP or Si7013EVB-UDP-F960) contains two software projects:

- Demo Software
- Data Logger

The purpose of the Demo Software is to provide simple example code. The Data Logger project provides a more full featured demonstration of the Si7013's capability.

3.1. Demo Software

The Si7013 Unified Development Platform (UDP) Demo software runs on a UDP C8051F960/Si1020 MCU card (UPMP-F960-MLCD). The MCU card may be plugged into a UDP motherboard, but this is not necessary. A Si7013 port header card (Si7013EB-UDP) is plugged into the MCU card.

The demo reads the temperature and humidity from the Si7013 port header card and displays them on the MCU card's LCD. The user may press the SW1 button to toggle the displayed temperature between Celsius and Fahrenheit. The user may press the SW2 button to switch the displayed temperature from the internal temperature sensor and the external thermistor. If the thermistor temperature is displayed, then a dot is shown after the temperature.

3.1.1. Running the Demo Software

The "Si7013 UDP F960 Evaluation Board Documentation and Software" CD contains the source code for the Demo Software. However, you do not need to build the Demo Software, because the CD also contains a pre-built image file and a batch file to download the image file to the MCU card.

1. Plug the Si7013 port header card on to J11 of the MCU card. The Silicon Labs logo on the port header card faces the battery on the MCU card.
2. Connect the USB Debug Adapter to a USB port on the PC and connect the USB Debug Adapter to the debug connector (J13) on the MCU card.
3. Connect the power supply to the MCU board (P1).
4. Table 2 lists the correct MCU card switch settings:

Table 2. MCU Card Switch Settings

Switch Label	Number	Setting
VBAT SEL	SW5	VREG
VIORF SELECT	SW12	VBAT
VIO SELECT	SW7	VBAT

5. Open Windows Explorer and change to the project folder: \UDP_Software_vX.X\Demo\F960\ (where vX.X is the version number of the UDP software)

6. Double-click on the “Install_Si70xx.bat” file in the project folder. This opens a command window and downloads the pre-built image file (UDP_Demo_Si70xx.hex) to the MCU card.
7. If the command window says “Could not connect with the board. Unknown device.,” then the MCU may be in sleep mode. The flash utility cannot talk to the MCU while it is sleeping. You must pause the MCU before running the batch file. Turn off the MCU board (VBAT SEL to UDP), press and hold SW4, and then turn on the MCU board (VBAT SEL to VREG). The LCD displays “PAUSED”. While continuing to hold SW4, double-click on the “Install_Si70xx.bat” file. When the command window says “Press any key to continue”, then release SW4.
8. The download is complete when the command window says “Press any key to continue”. Press any key to close the command window.

The demo software displays temperature and humidity on the LCD. The LCD is updated twice a second with new temperature and humidity readings. If you prefer the temperature to be displayed in Fahrenheit, then press the SW1 button.

The demo software is now installed on the MCU card and the USB Debug Adapter is no longer needed. Disconnect the USB Debug Adapter from the debug connector on the MCU card.

3.1.2. Source Code for the Demo Software

The source code for the demo software consists of eight components:

- Main
- Sensor
- Power Management
- Real Time Clock
- Tick Counter
- Port Match
- LCD Driver
- I^2C Driver

3.1.2.1. Main

The Main component is the entry point for the demo software and it calls routines in the other components to implement the application.

The Main component is implemented in a single file:

Main.c

The demo software is written to have the longest battery life possible. The demo software uses the internal low-power oscillator and it puts the MCU into the lowest power mode (sleep) between sensor reads.

The main() routine begins by initializing the MCU and the other components. It then configures a real time clock (RTC) alarm to wake up the MCU twice a second. The Power Manager is configured to wake up when an RTC alarm occurs or when a port match occurs.

The main() routine then enters an infinite loop where it alternates between sleeping the MCU, and reading and displaying the temperature and humidity.

3.1.2.2. Sensor

The Sensor component provides routines for reading temperature and humidity from an Si7013 device and is implemented in a single file: Sensor_Si70xx.c

The Sensor component provides the following routines: Sensor_ReadTemperature(), Sensor_ReadThermistor(), and Sensor_ReadHumidity(). These routines call the Sensor_Measure() routine to take a measurement and return the value from the Si7013’s data register. The Sensor_ReadTemperature() and Sensor_ReadThermistor() routines convert the returned value to deci-degrees Celsius. The Sensor_ReadHumidity() routine converts the returned value to deci-percent relative humidity.

The Sensor component also provides the Sensor_Sleep() and Sensor_WakeUp() routines. These routines are not needed by the Si7013 because the Si7013 automatically sleeps between commands.

The demo software does not use floating point numbers because the floating point library is large and slow. The

demo software avoids floating point numbers by using deci-units.

A deci-unit is one tenth of a unit. The Sensor_ReadHumidity() routine returns the humidity in deci-percent RH and the Sensor_ReadTemperature() routine returns the temperature in deci-degrees Celsius. For example, 43.7 percent RH is 437 deci-percent RH. Likewise, 25.4 degrees Celsius is 254 deci-degrees Celsius.

The demo software uses equations that are derived from the equations in the Si7013 data sheet. The derived equations are modified to use deci-units.

3.1.2.3. Power Management

The Power Management component is implemented in a single file called PowerManagement.c and contains the routines, LPM_Enable_Wakeup() and LPM_Disable_Wakeup(), for entering and exiting low-power modes. These routines configure which events will wake up the MCU from low-power mode (LPM).

The LPM() routine puts the MCU into the specified low-power mode. The MCU remains in the low-power mode until one of the configured wake up events occurs. When the MCU wakes up, then the LPM() routine examines the MCU registers to determine which event caused the wake up and sets the appropriate WakeUp global variable. The LPM() routine then returns to the caller. The caller may examine the WakeUp global variables to find out which event caused the wake up.

3.1.2.4. Real Time Clock

The Real Time Clock (RTC) component uses the SmaRTClock peripheral to wake up the MCU from low-power modes.

The Real Time Clock component is implemented in a single file called SmaRTClock.c.

The Real Time Clock component contains the RTC_Read() and RTC_Write() routines for reading and writing RTC registers. RTC registers are not MCU Special Function Registers (SFRs), but must be accessed indirectly using the SFRs. The RTC_Read() and RTC_Write() routines perform the indirect reads and writes of the RTC registers.

The RTC timer can be read and written with the RTC_CapureTimer() routine and the RTC_SetTimer() routine. The RTC timer is a 32-bit counter, which increments at a 32.768 kHz rate. The RTC timer continues to increment when the MCU is in low-power mode.

The RTC_WriteAlarm() and RTC_ReadAlarm() routines allows RTC alarms to the written and read. When the RTC timer value equals an alarm value, then an RTC alarm event occurs.

3.1.2.5. Tick Counter

The Tick Counter component provides delay and timeout capabilities.

The Tick Counter component is implemented in a single file called Tick.c.

Timer 2 is configured to interrupt once a millisecond. The timer 2 interrupt handler increments an unsigned 16-bit variable named TickCounter. The tick counter wraps around about every 65 seconds.

The current value of the tick counter is obtained by calling TickCount(). Timeouts are implemented by calling TickCount() to get a starting time and then repeatedly calling ElapsedTime() to get the number of milliseconds that have elapsed since the starting time. The Delay() routine does not return until the specified number of milliseconds have elapsed.

3.1.2.6. Port Match

The Port Match component changes the temperature scale between Celsius and Fahrenheit when the user presses the SW1 button. In addition, the SW2 button switches the temperature readings between the internal sensor and the external thermistor.

The Port Match component is implemented in a single file called PortMatch.c.

The SW1 button on the MCU board is connected to bit 0 of Port 0 (P0.0). This bit is high when the button is not pressed, but when the button is pressed the bit goes low. The SW2 button is connected to bit1 of Port0 (P0.1).

The MCU is configured to generate a port match interrupt when P0.0 or P0.1 goes low. If the MCU is sleeping when a button is pressed, then the port match event wakes up the MCU and the port match interrupt handler is called.

When the interrupt handler detects that a button has been pressed, then the interrupt handler reconfigures the port match to detect when the button is released. This prevents a large number of useless port match interrupts while the button is held down.

3.1.2.7. LCD Driver

The LCD driver contains routines for displaying characters, strings, and numbers on the LCD.

The LCD driver is implemented in four files:

- lcdConfig.c
- lcdPutChar.c
- lcdPutString.c
- lcdPutNumber.c

The LCD driver provides the routines lcdPutChar(), lcdPutString() and lcdPutNumber() for displaying characters, strings and numbers on the LCD. The lcdPutNumber.c file also contains the routines lcdPutTemperature() and lcdPutHumidity() for displaying temperature and humidity on the LCD.

The demo software does not use the printf() routine, because the printf() routine is very large and slow. Instead, the lcdPutHumidity() and lcdPutTemperature() routines use the lcdPutNumber() routine, which is much smaller and quicker. The lcdPutTemperature() routine formats the temperature such that it is displayed with the greatest possible resolution.

3.1.2.8. I^2C Driver

The I^2C driver provides routines for reading and writing registers on an I^2C slave device.

The I^2C driver is implemented in a single file called I2C.c.

The I2C_ReadByte() routine and the I2C_WriteByte() routine read and write 8-bit registers on an I^2C slave device. The demo software uses these routines to access most registers on the Si7013.

The I2C_ReadData() routine and I2C_WriteData() routine perform multi-byte reads and writes from an I^2C slave device. The demo software uses the I2C_ReadByte() routine to read the 16-bit Data register on the Si7013.

The Si7013 daughter card has two I^2C buses. The first bus is connected to the onboard Si7013 device. The second bus is connected to the flat flexible cable (FFC) connector (J2) on the side of the board. A second Si7013 device can be attached to the FFC connector. The “Bus” parameter in each of the I^2C routines selects which bus to use (I2C_BUS_1 or I2C_BUS_2).

3.2. Data Logger

The Si70xx Data Logger demonstrates how the C8051F960 microcontroller unit (MCU) and the Si7013 sensor can be used to record temperature and humidity over an extended period of time. This project includes Data Logger firmware that runs on a C8051F960/Si1020 MCU card (UPMP-F960-MLCD) and a Data Logger application that runs on a PC. The Data Logger firmware reads the temperature and humidity from an Si7013 port header card (Si7013EB-UDP), which is plugged into the MCU card, and stores the samples in the MCU's flash memory. The flash memory can store up to 8192 samples. The Data Logger application displays the samples on a graph.

3.2.1. Installing the Data Logger

There are three steps to installing the Data Logger:

1. Install the Data Logger firmware on the MCU card.
2. Install CP210x drivers on the PC.
3. Install the Data Logger application on the PC.

3.2.1.1. Installing the Data Logger Firmware

The “Si7013 UDP F960 Evaluation Board Documentation and Software” CD contains the source code for the Data Logger firmware. However, you do not need to build the Data Logger firmware, because the CD also contains a pre-built image file and a batch file to download the image file to the MCU card.

1. Plug the Si7013 port header card on to J11 of the MCU card. The Silicon Labs logo on the port header card faces the battery on the MCU card.
2. Connect the USB Debug Adapter to a USB port on the PC and connect the USB Debug Adapter to the debug connector (J13) on the MCU card.
3. Connect the power supply to the MCU board (P1).
4. Table 3 lists the correct MCU card switch settings:

Table 3. MCU Card Switch Settings

Switch Label	Number	Setting
VBAT SEL	SW5	VREG
VIORF SELECT	SW12	VBAT
VIO SELECT	SW7	VBAT

5. Open Windows Explorer and change to the project folder: \UDP_Software_vX.X\DataLogger\FW\F960\ (where vX.X is the version number of the UDP software)
6. Double-click on the “Install_Si70xx.bat” file in the project folder. This opens a command window and downloads the pre-built image file (UDP_DataLogger_Si70xx.hex) to the MCU card.
7. If the command window says “Could not connect with the board. Unknown device.”, then the MCU may be in sleep mode. The flash utility cannot talk to the MCU while it is sleeping. You must pause the MCU before running the batch file. Turn off the MCU board (VBAT SEL to UDP), press and hold SW4, and then turn on the MCU board (VBAT SEL to VREG). The LCD displays “PAUSED”. While continuing to hold SW4, double-click on the “Install_Si70xx.bat” file. When the command window says “Press any key to continue”, then release SW4.
8. The download is complete when the command window says “Press any key to continue”. Press any key to close the command window.

The Data Logger firmware displays temperature and humidity on the LCD. The LCD is updated twice a second with new temperature and humidity readings. If you prefer the temperature to be displayed in Fahrenheit, then press the SW2 button.

The Data Logger firmware is now installed on the MCU card and the USB Debug Adapter is no longer needed. Disconnect the USB Debug Adapter from the debug connector on the MCU card.

3.2.1.2. Installing CP210x Drivers

The MCU card contains a CP2102 USB-to-UART bridge chip. The Data Logger application uses a virtual COM port (VCP) on the PC to send serial packets to the CP2102 over USB. The CP2102 then uses its UART to forward the packets to the Data Logger firmware.

This section installs two drivers on the PC:

- CP210x VCP Driver
- CP210x USB-to-UART Bridge Driver

Perform the following steps to install these drivers:

1. Run the following program from the “UDP_Software_vX.X” folder of the “Si7013 UDP Evaluation Board Documentation and Software” CD: CP210x_VCP_Win_XP_S2K3_Vista_7.exe
2. Use the InstallShield Wizard to install the VCP driver.

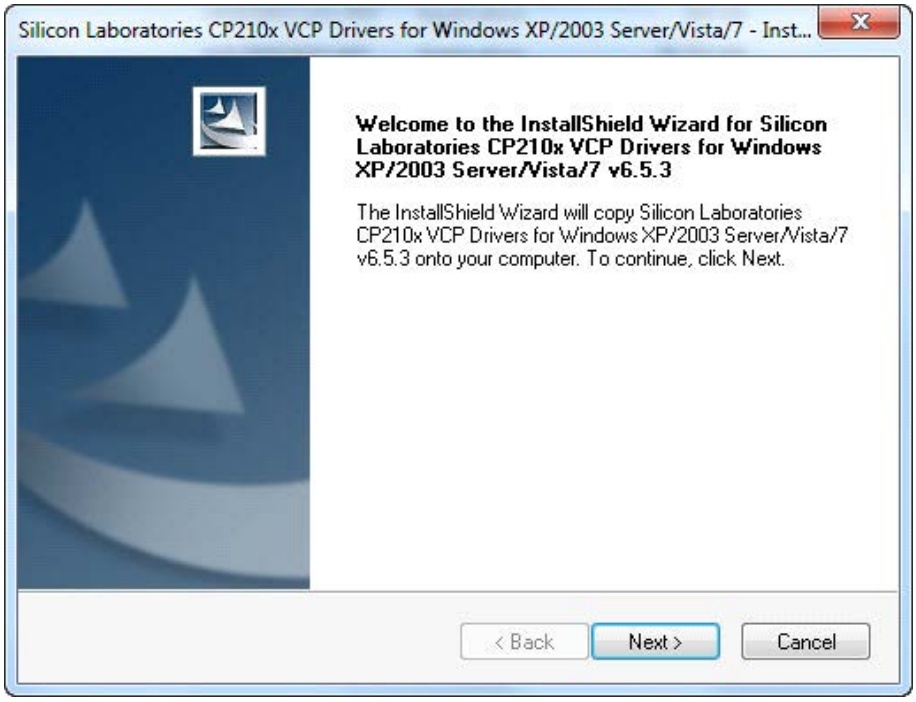


Figure 4. InstallShield Wizard

3. At the “InstallShield Wizard Complete” window, make sure that “Launch the CP210x VCP Driver Installer” is checked; then click on the “Finish” button. This begins the installation of the USB-to-UART bridge driver.

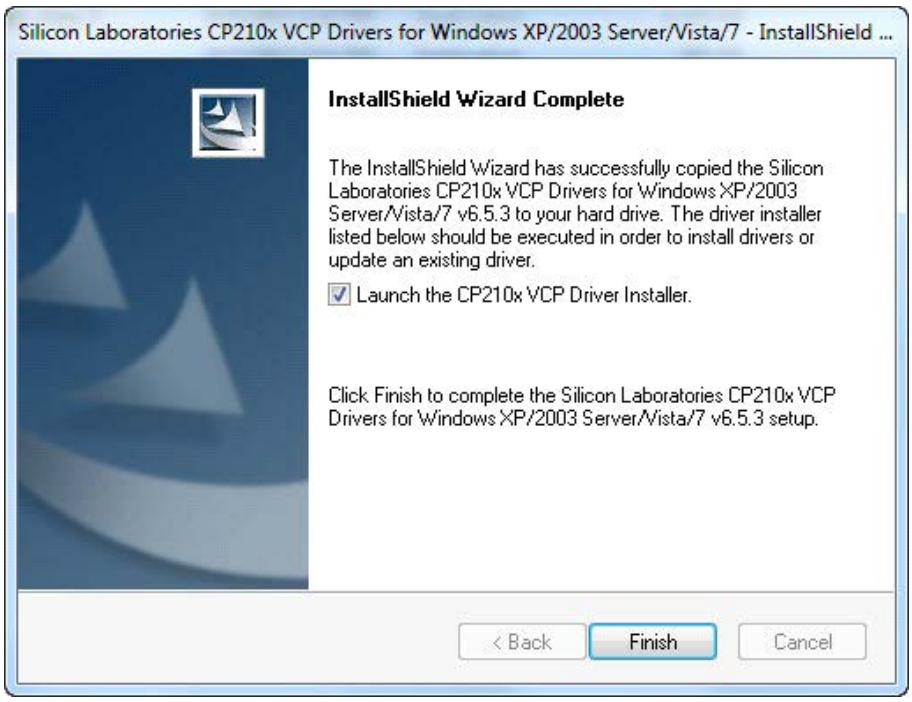


Figure 5. InstallShield Wizard Complete Window

4. When the installation of the bridge driver is complete, finalize the installation by connecting the MCU card to the PC with a USB cable.

3.2.1.3. Installing the Data Logger Application

Perform the following steps to install the Data Logger application from the "UDP_Software_vX.X" folder of the "Si7013 UDP Evaluation Board Documentation and Software" CD.

1. Run the following program from the "Si7013 UDP Evaluation Board" CD:
   DataLoggerSetup_vX.X.exe (where X.X is the latest version number)
2. Accept the Software License Agreement by clicking on the "Yes" button.

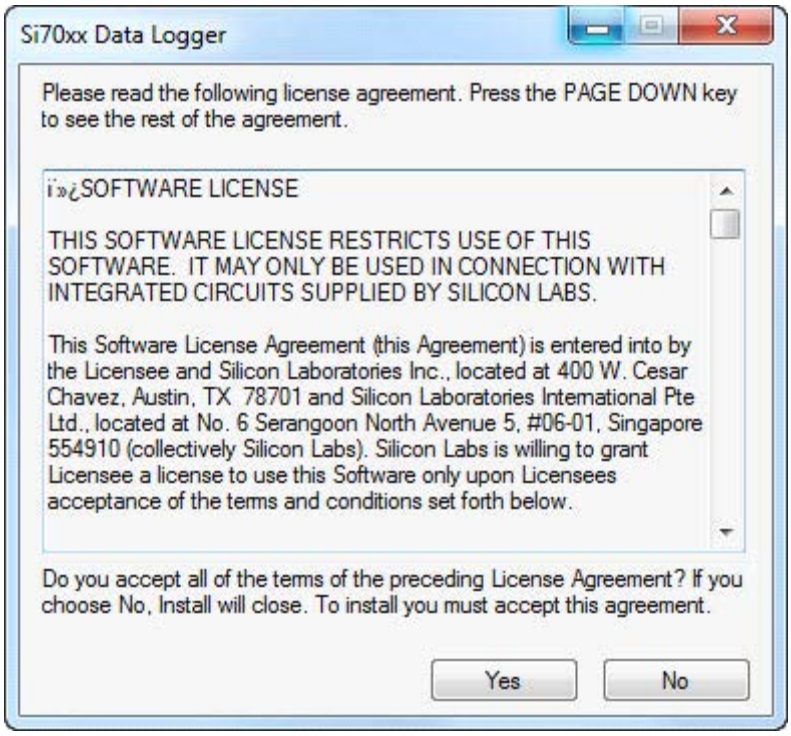


Figure 6. Software License Agreement

3. Use the Si70xx Data Logger Setup Wizard to install the application.

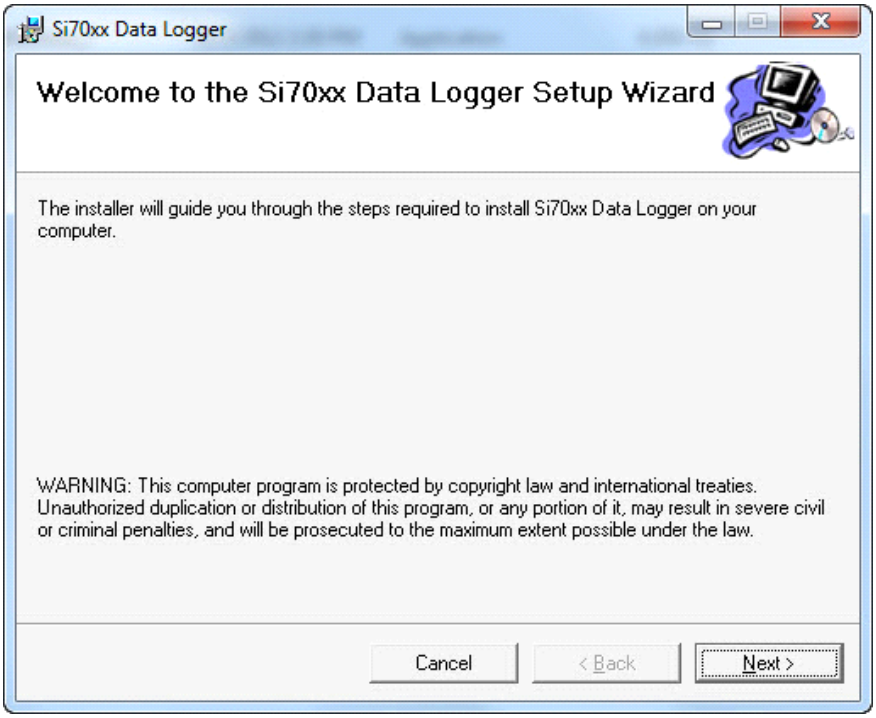


Figure 7. Si70xx Data Logger Setup Wizard

3.2.2. Using the Data Logger

When running the Data Logger firmware for the first time, the firmware takes a sample every two minutes but does not store the samples in the log. To start storing samples, press the SW3 button. Each sample includes temperature, humidity, and a time stamp. The firmware's clock has not yet been set; so, time stamps begin at midnight January first 2000. You may use the Data Logger application to display the stored samples. The application can also set the firmware's clock and adjust the time between samples.

The typical procedure for using the Data Logger is as follows:

1. Connect the MCU card to the PC with a USB cable.
2. Use the Data Logger application to stop logging and erase the log.
3. Set the time and the desired sample interval.
4. Ensure the MCU card is set up to use the onboard battery.
5. Disconnect the MCU card from the PC and take it to the location where you want to take samples.
6. Press the SW3 button on the MCU card to start logging.
7. Wait while samples are logged.
8. Press the SW3 button on the MCU card to stop logging.
9. Take the MCU card back to the PC and reconnect it to the PC.
10. Use the Data Logger application to display the log data.
11. Save the log data to a file.

The Data Logger firmware can generate alarms when the temperature or humidity gets too high or too low. When an alarm occurs, the Data Logger firmware displays a flashing message on the LCD and pulses an output pin. If a buzzer is connected to the output pin, an audible alarm is produced. The Data Logger application can configure the thresholds for the alarms.

3.2.2.1. Using the Data Logger Application

The Data Logger application runs on a PC and is used to configure the Data Logger firmware. The Data Logger application can also display the log data.

The Data Logger application uses the RTS signal on the MCU card to wake up the Data Logger firmware. The MCU card must be configured such that P0.6 is used for RTS. To do this, move the shorting block from J18 to P2 as shown in Figure 8.

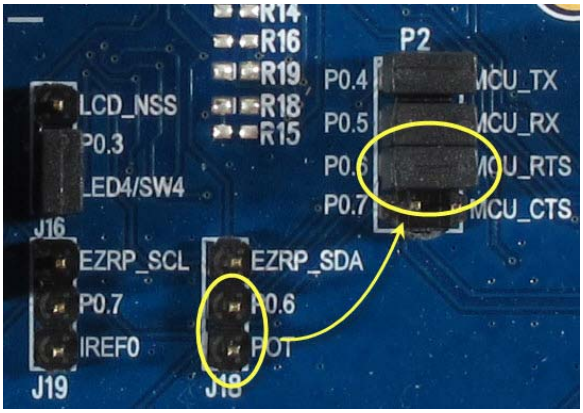


Figure 8. Configuring the MCU Card

To run the Data Logger application from the Start menu, select Start → Silicon Laboratories → Si70xx Data Logger. This opens up the Data Logger application shown in Figure 9.

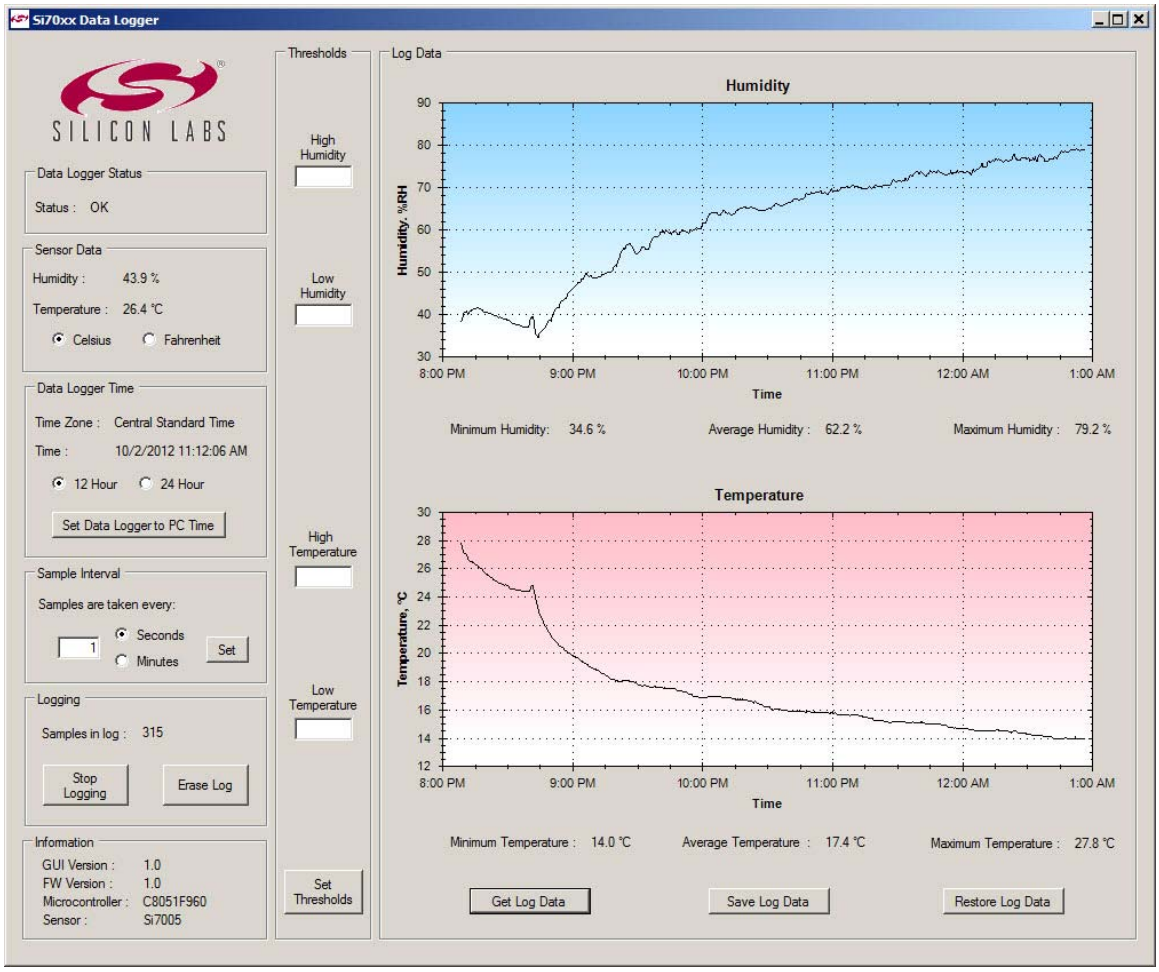


Figure 9. Data Logger Application

The left side of the application window is used to configure the Data Logger firmware. The right side of the application window is used to display log data.

3.2.2.1.1. Data Logger Status

The Data Logger Status box shown in Figure 10 displays the status of the Data Logger firmware.



Figure 10. Data Logger Status Box

If there are no alarms or errors, this box displays “OK”. If there is an alarm, this box displays an alarm message. If the status is “Communications Error”, this means the MCU card cannot communicate with the Si7013 port header card.

3.2.2.1.2. Sensor Data

The Sensor Data box shown in Figure 11 displays the current humidity and temperature.

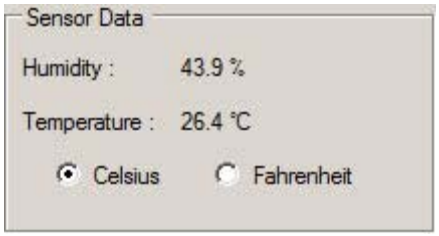


Figure 11. Sensor Data Box

This box displays the temperature and humidity from the most recent sample.

The Celsius and Fahrenheit radio buttons select whether the temperatures displayed in this box, the Thresholds box, and the Log Data box are in Celsius or Fahrenheit. These radio buttons do not affect how temperatures are displayed in the LCD.

3.2.2.1.3. Data Logger Time

The Data Logger Time box shown in Figure 12 is used to set the time in the Data Logger firmware.

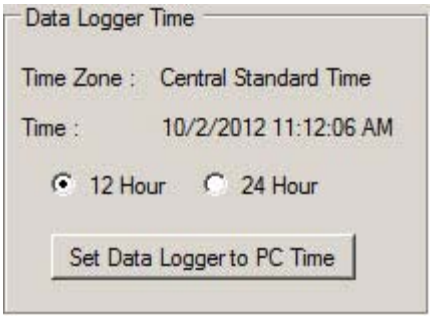


Figure 12. Data Logger Time Box

The upper part of this box displays the current time and current time zone of the Data Logger firmware.

The 12 Hour and 24 Hour radio buttons select the format of the time displayed in this box and in the Log Data box.

The “Set Data Logger to PC Time” button sets the time of the Data Logger firmware to the time of the PC. This button also sets the time zone in the Data Logger firmware to the time zone of the PC.

The Data Logger firmware does not use the time zone information; it is simply used to document the context of the time stamps in case the Data Logger travels to different time zones.

3.2.2.1.4. Sample Interval

The Sample Interval box shown in Figure 13 is used to configure how often samples are taken.

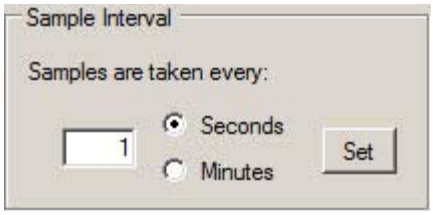


Figure 13. Data Logger Time Box

This text box displays the current setting of the sample interval. To change the sample interval, enter a number into the text box, select "Seconds" or "Minutes", and then click on the "Set" button.

The sample interval may be from 1 second to 1092 minutes (18 hours).

3.2.2.1.5. Logging

The Logging box shown in Figure 14 is used to control the log in the Data Logger firmware.

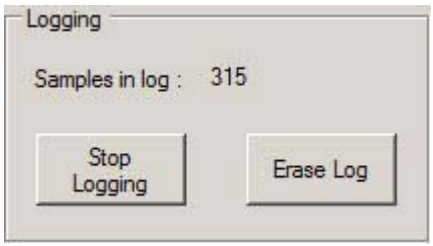


Figure 14. Logging Box

The upper part of this box displays the current number of samples in the log.

The Data Logger firmware constantly takes samples and displays them on the LCD. If logging is enabled, the samples are also saved in the log. Samples continue to be taken when logging is disabled.

If logging is enabled, the left button says “Stop Logging”, and logging is disabled if this button is clicked.

If logging is disabled, the left button says “Start Logging”, and logging is enabled if this button is clicked.

Note: Logging may also be enabled and disabled with the SW3 button on the MCU card.

The “Erase Log” button deletes all the samples in the log.

3.2.2.1.6. Information

The Information box displays version and identification information.

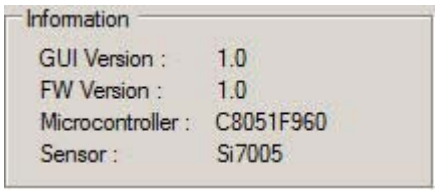


Figure 15. Information Box

3.2.2.1.7. Thresholds

The Thresholds box is used to configure alarm thresholds.

A humidity alarm occurs if the humidity is greater than the high humidity threshold or less than the low humidity threshold. A temperature alarm occurs if the temperature is greater than the high temperature threshold or less than the low temperature threshold.

The Thresholds box displays the current threshold settings. If a text box is empty, that threshold is not used. If all text boxes are empty, then no humidity or temperature alarms are generated by the Data Logger firmware.

To set the thresholds, enter numbers into the text boxes and then click on the “Set Thresholds” button. If you do not want to use a particular threshold, then make that text box empty, and click on the “Set Thresholds” button.

A humidity threshold may be from 0 to 100. A temperature threshold may be from –40 to 85 if the Celsius radio button is selected and from –40 to 185 if the Fahrenheit radio button is selected.

3.2.2.1.8. Log Data

The Log Data box shown in Figure 16 displays log data in humidity and temperature graphs.

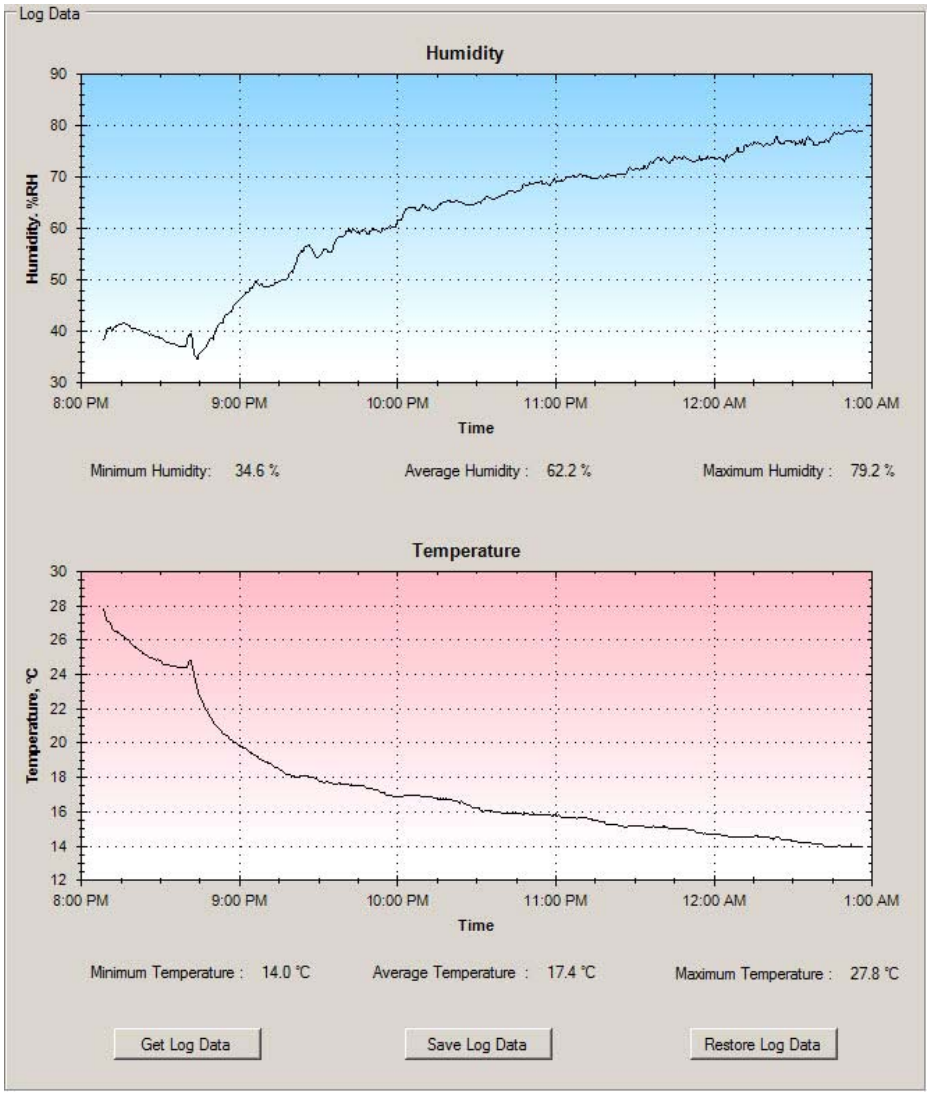


Figure 16. Log Data Box Temperature and Humidity Graphs

The “Get Log Data” button transfers all of the samples from the MCU card to the PC and displays them in the humidity and temperature graphs. The application also displays the minimum, maximum, and average values for temperature and humidity.

The “Save Log Data” button saves the log data to a file in comma-separated value (CSV) format. A dialog box allows you to specify the name and location of the file. The “Restore Log Data” button allows you to redisplay a CSV file that was previously saved with this application.

If you want to zoom into an area of a graph, then create a dashed box around the area by dragging the left mouse button across the graph as shown in Figure 17.

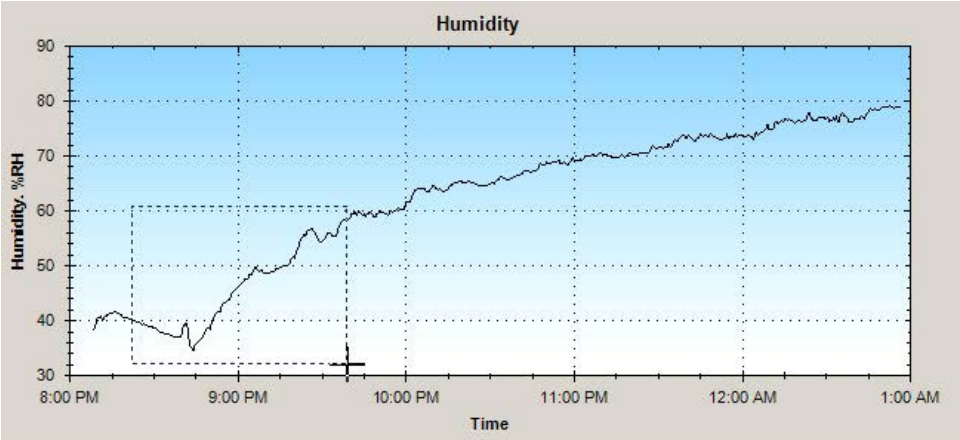


Figure 17. Zooming in on a Portion of the Graph

The graph will zoom in to the selected area. You may pan around the graph by first holding the Ctrl key, then dragging the graph with the left mouse button. You may also use the mouse wheel to zoom in and out.

If you click a graph with the right mouse button, a context menu is displayed as shown in Figure 18.

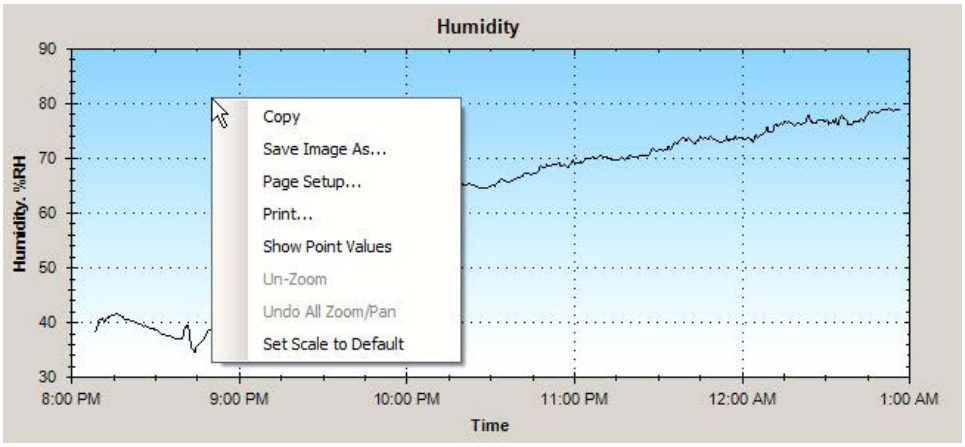


Figure 18. Context Menu

The context menu allows you to unzoom the graph. The context menu also lets you save and print images of the graph.

If you select “Show Point Values” from the context menu and hover the mouse over the graph line, a tool tip displays the time and value for that point on the graph as shown in Figure 19.

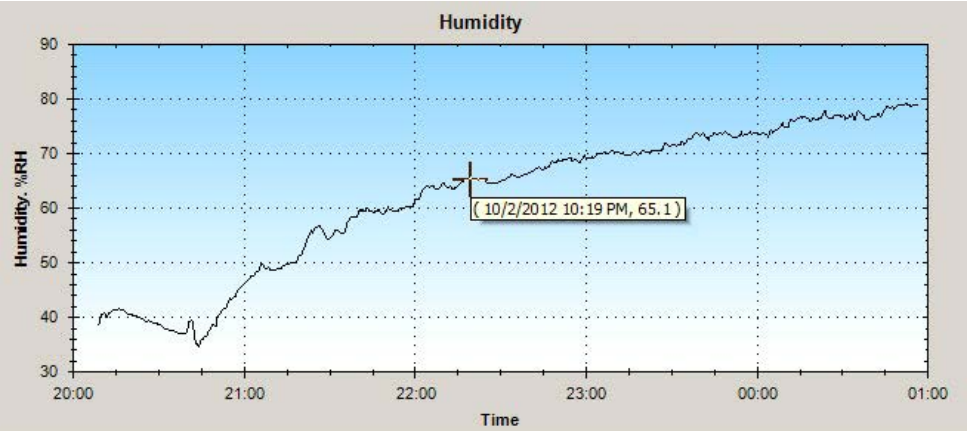


Figure 19. Sample Time and Value

3.2.2.2. Using the Data Logger Firmware

The Data Logger firmware runs on an MCU card where it logs temperature and humidity data. The Data Logger firmware also generates alarms if the temperature or humidity become too high or too low.

The user interface for the Data Logger firmware uses buttons, LCD displays, and an output pin to sound a buzzer.

3.2.2.2.1. Buttons

The Data Logger firmware shown in Figure 20 uses three buttons and an LED.

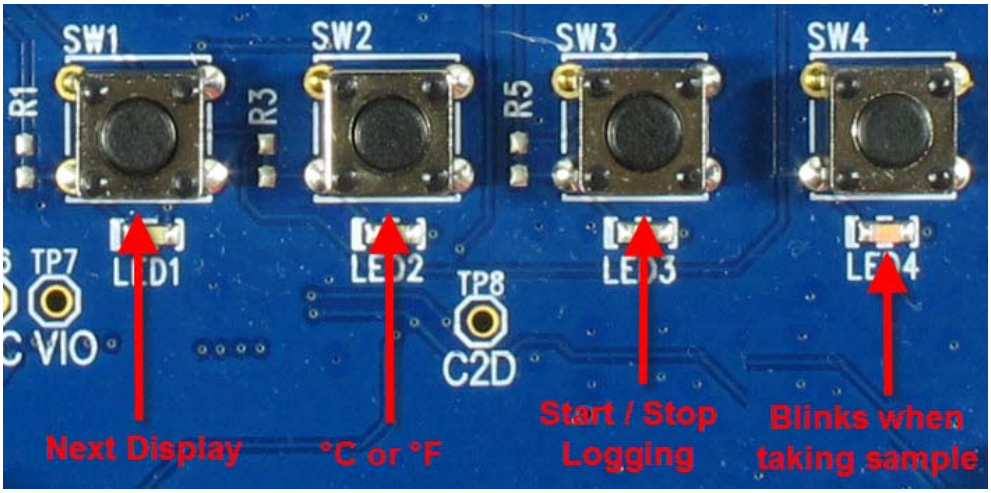


Figure 20. Data Logger Firmware Buttons

The SW1 button changes the information that is displayed on the LCD. The LCD rotates through the following displays:

- Temperature and Humidity
- Date
- Time
- Sample Count

If there is an alarm, the Data Logger firmware displays a flashing alarm message on the LCD. The alarm message can be dismissed by pressing the SW1 button. This allows the normal displays to be seen (until the next alarm occurs).

The SW2 button toggles the displayed temperature between Celsius and Fahrenheit.

The SW3 button starts and stops logging. If logging is disabled, pressing the SW3 button enables logging. If logging is enabled, pressing the SW3 button disables logging. If logging is enabled, samples are stored in the log.

The Data Logger firmware blinks LED4 when a sample is taken. This allows you to verify that samples are being taken at the expected rate.

If the Data Logger application is not connected to the MCU card, then LED4 is off (except for occasional blinks). If the Data Logger application is connected to the MCU card, then LED4 is on (except for occasional blinks). This allows you to verify that the Data Logger application is successfully communicating with the Data Logger firmware.

3.2.2.2.2. LCD Displays

There are five LCD displays for temperature and humidity, date, time, sample count, and alarm.

3.2.2.2.2.1. Temperature and Humidity Display

This display shows the temperature and humidity from the most recent sample.



Figure 21. Temperature and Humidity Display

You can press the SW2 button to switch the temperature between Celsius and Fahrenheit.

3.2.2.2.2.2. Date Display

This display show the current date.



Figure 22. Date Display

The date is formatted as month/day/year.

3.2.2.2.2.3. Time Display

This display shows the current time.

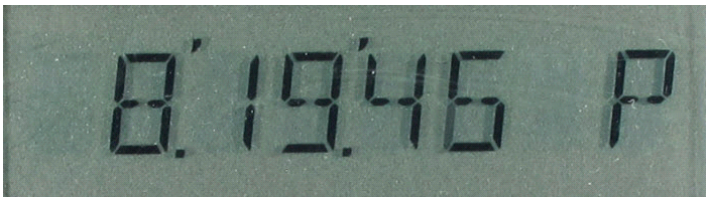


Figure 23. Time Display

The format of the time is hours : minutes : seconds A or P.

This display allows you to verify that the time is set correctly.

3.2.2.2.2.4. Sample Count Display

This display shows the current number of samples in the log.

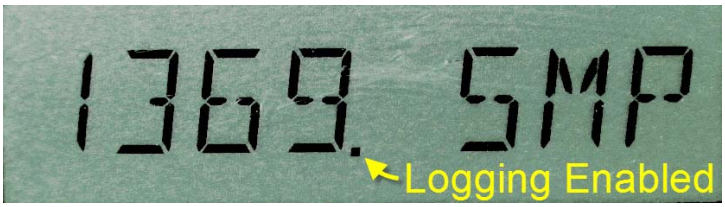


Figure 24. Sample Count Display

This display also indicates if logging is enabled. If logging is enabled, a dot is displayed to the right of the number.

3.2.2.2.2.5. Alarm Display

When an alarm occurs, the LCD displays a flashing alarm message as shown in Figure 25.



Figure 25. Alarm Display

An alarm message shows the offending temperature or humidity on the left and the word “HI” or “LOW” on the right. The above alarm message indicates that there is high temperature alarm and that the high temperature is 35.0°C.

If both a temperature alarm and a humidity alarm occur at the same time, the LCD alternates between the two alarm messages.

If the alarm message says “COMM ERR”, then the Data Logger firmware has failed to communicate with the Si7013 port header card. Please verify that the port header card is properly plugged into the MCU card.

3.2.2.2.3. Output Pin

When a sample is taken that causes an alarm, the Data Logger firmware pulses the output pin low. The output pin may be used to signal an external device or to sound a buzzer.

The output pin is P1.0, which is available at the PC0 screw terminal of the Pulse Counter connector (J14). The output pin is normally 3.3 V, but, when an alarm occurs, it goes to 0 V for 500 ms. The Data Logger firmware configures the output pin for high drive strength, so it can sink up to 80 mA.

Note: A buzzer is not included with the evaluation kit. If you purchase a buzzer, make sure the buzzer works with 3 V and not 12 V.

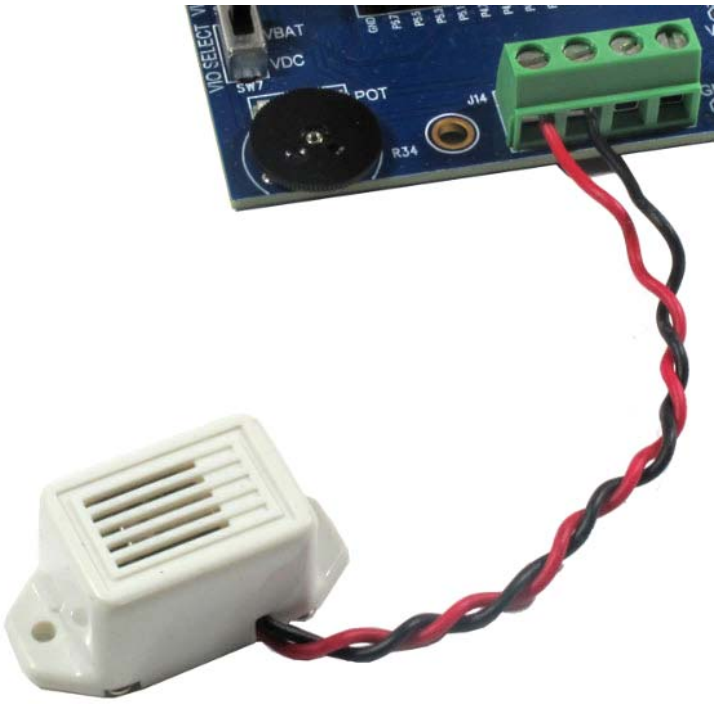
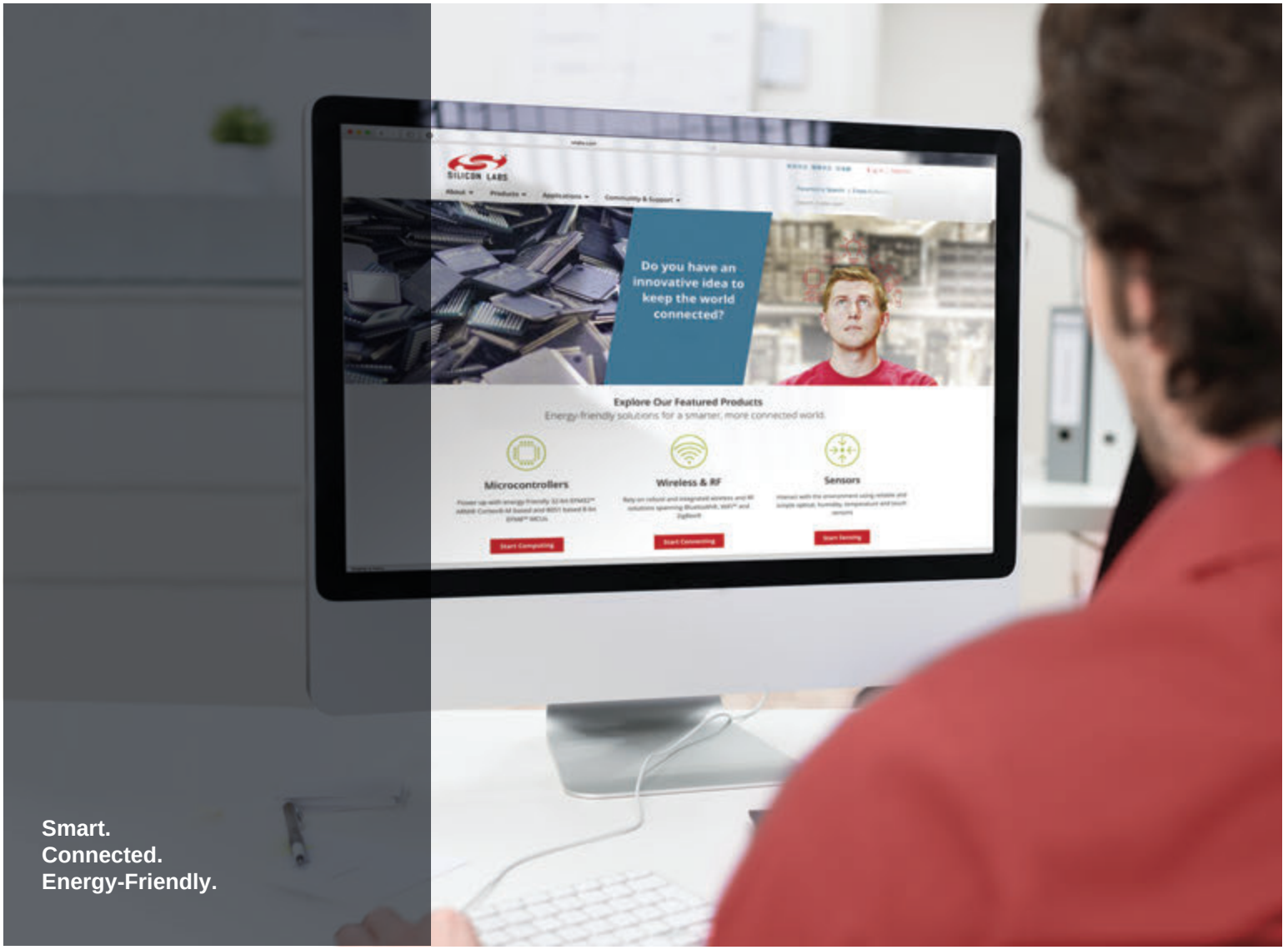


Figure 26. Optional 3 V Buzzer

4. Additional Reference Resources

- Si7013 data sheet
- AN607: Si70xx Humidity Sensor Designers Guide
- Si7013USB-dongle users guide
- C8051F96x data sheet
- UDP C8051F960/Si1020 MCU Card with Multiplexed LCD User's Guide



Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

Disclaimer
Silicon Laboratories intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Laboratories products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Laboratories reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Laboratories shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System without the specific written consent of Silicon Laboratories. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Laboratories products are not designed or authorized for military applications. Silicon Laboratories products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

Trademark Information
Silicon Laboratories Inc.® , Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, Clockbuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, ISOmodem®, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress® and others are trademarks or registered trademarks of Silicon Laboratories Inc. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com