

USB-900 Control Center User's Guide

APPLICABLE DEVICES

This Application Note applies to the following Keterex devices: USB-910H Embedded Systems Interface.

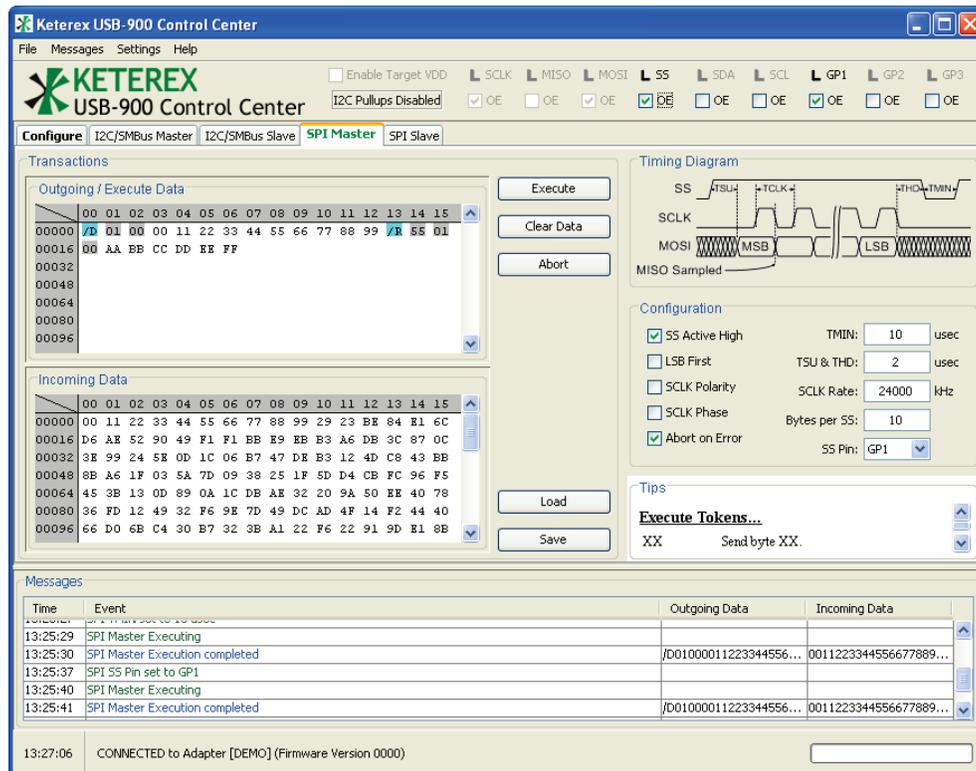
The USB-910H Adapter provides configurable I/O voltage levels from 1.65V to 5V. In addition to setting the drive and sense logic levels, this voltage can be connected to the target hardware, providing up to 400mA to the target device.

INTRODUCTION

The USB-910H Embedded Systems Interface provides connectivity between a host PC and an I2C, SMBus, or SPI interface. Other interfaces can be supported by "bit-banging" up to 9 general-purpose I/O. USB transactions are generated on the Host PC using either the Keterex USB-900 Control Center application or calls to a provided API library. The USB-910H Adapter converts these transactions to I2C, SMBus, SPI, or general-purpose I/O operations. Built-in scripting accommodates complex bus protocols.

KEY POINTS

- SPI, I2C, SMBus Master or Slave
- Full I2C/SMBus Multi-Master Support
- Provides up to 9 General-Purpose I/O
- Configurable I/O Levels from 1.65V to 5V
- SPI up to 24MHz, I2C/SMBus to 1.5MHz
- Can Power the Target from the Adapter
- Supports up to 16 Slave Addresses
- Powerful Scripting Feature



1. The Keterex USB-910H Embedded Systems Interface

The Keterex USB-910H Embedded Systems Interface includes the following components:

- The Keterex USB-910H Adapter.
- The Keterex USB-900 Control Center Application.
- The Keterex USB-910H API Library.

This application note describes the USB-900 Control Center. See the USB-910H Datasheet and Application Note AN2101 for additional information.

2. The Keterex USB-900 Control Center

The Keterex USB-900 Control Center provides access to all features of the Keterex USB-910H Adapter, including:

- Selecting an adapter for communications.
- Setting the voltage of the I/O interface from 1.65V to 5V.
- Connecting the configured supply voltage to the target system.
- Controlling the direction and state of up to 9 general-purpose I/O pins.
- Activating an adapter as an I2C, SMBus, or SPI master or slave device.
- Selecting a wide range of bit-rates and other timing controls.
- Enabling or disabling built-in I2C pull-up resistors.
- Initiating simple or complex master or slave transactions.
- Responding to multiple slave addresses or slave-select pins.

The Control Center provides a powerful but intuitive interface to the USB-910H Adapter. As shown below, common control, status, and messaging information are always visible on the screen. Information associated with specific features, such as the I2C Master or SPI Slave, are grouped onto four separate feature panes accessed through tabs on the main Control Center window. An additional configuration pane provides access to common adapter controls, such as searching for available adapters and setting I/O voltage levels.

The screenshot shows the Keterex USB-900 Control Center application window. The interface includes a menu bar (File, Messages, Settings, Help), a toolbar with various feature icons (I2C, SMBus, SPI, etc.), and a main content area with tabs for 'Configure', 'I2C/SMBus Master', 'I2C/SMBus Slave', 'SPI Master', and 'SPI Slave'. The 'Configure' tab is active, showing options for selecting an adapter, setting I/O voltage (3.3V), and enabling features like I2C Master/Slave and SPI Master/Slave. A 'Pinout Bottom View' shows a grid of pins (GP2 to GP9) and their signals (SCL, GND, SDA, GP3/VDD, MISO, SCLK, MOSI, SS). A 'Messages' pane at the bottom displays a log of events with time-stamped entries. Callouts point to specific elements: 'Feature Panes' (the tabs), 'Time-stamped Messages' (the log), 'Connection Status' (the bottom bar), 'Adapter Pins Control/Status' (the pinout grid), and 'Adapter Status' (the bottom right corner).

2.1. Menus

2.1.1. File → Exit

This menu item exits the Control Center application.

2.1.2. Messages → Save Messages

This menu item allows the user to save all messages to a selected file. These messages can be restored at a later time to allow the user to continue working with previous incoming and outgoing data sets and configurations (e.g. by double-clicking on restored messages). The Message Log is stored as a text XML file. See Section 2.5 for more information on messages.

2.1.3. Messages → Load Messages

This menu item allows the user to restore a previously saved Message Log. A restored message log has all the capabilities of a live log, including double-clicking on events to restore data sets and configurations. Since a long Message Log can take a while to load, the load operation occurs in the background, allowing the user to continue using the Control Center. See Section 2.5 for more information on messages.

2.1.4. Messages → Clear All Messages

This menu item deletes all messages after prompting for confirmation by the user.

2.1.5. Settings → Display HEX Labels in Editors

Enabling this setting causes the data editors in the various feature panes to label their rows and columns using hexadecimal numbers. Otherwise, the rows and columns are labeled using decimal numbers.

2.1.6. Settings → Enable Color Highlighting in Editors

This setting controls color highlighting in the data editors in the feature panes. When enabled, the editors color-code the outgoing data to identify tokens, parameters, etc. When disabled, only errors and slave-response addresses are highlighted.

2.1.7. Settings → Auto Load/Save Messages Log

Enabling this setting causes the Control Center to automatically save and reload the Message Log from a default location each time the application terminates and starts up.

2.1.8. Settings → Beep on Error Message

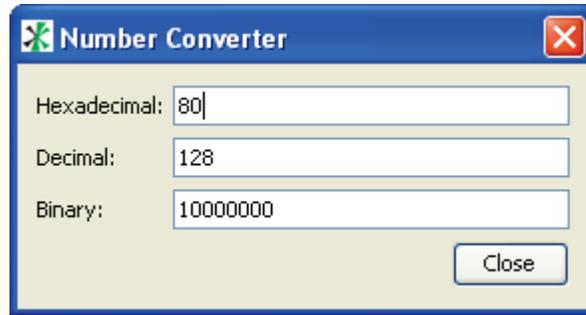
Enabling this setting causes the Control Center to beep each time an error message is generated.

2.1.9. Settings → Highlight New Messages

Enabling this setting causes the Control Center to highlight new messages in the Message Log. If enabled, a new message is highlighted to help differentiate it from older messages. After a few seconds, the highlighting will fade away.

2.1.10. Tools → Number Converter

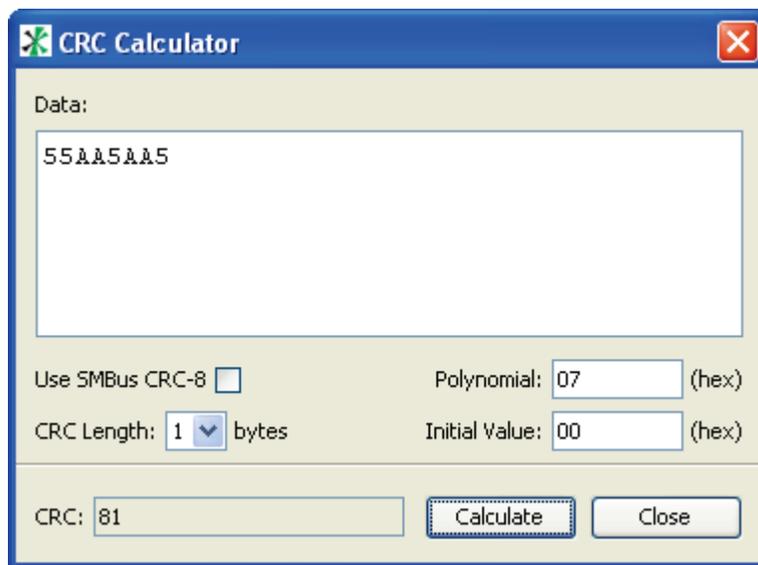
This menu item opens a simple number converter which allows the user to convert between hexadecimal, decimal, and binary formats, as shown below.



A number typed into any field causes the values in the other fields to automatically update to match the entered value. The Number Converter can remain open and available while using the Control Center.

2.1.11. Tools → CRC Calculator

The menu item opens the general-purpose CRC Calculator shown below.



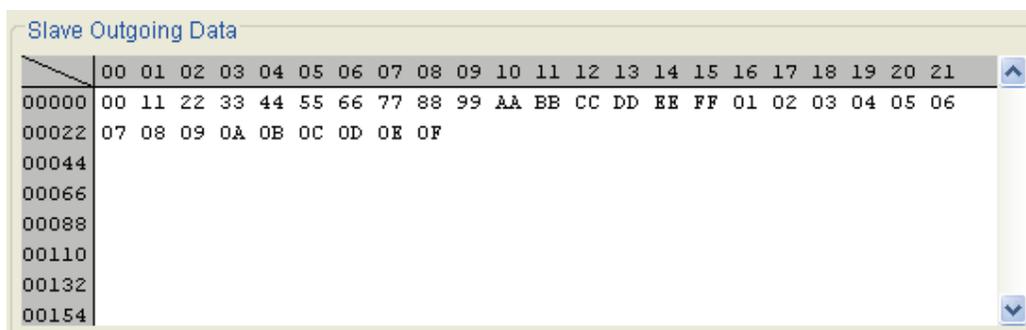
To calculate the CRC for a data set, enter the hexadecimal data into the Data box and click the **Calculate** button. The resulting CRC is shown in the field at the bottom left. By default, the calculator uses the SMBus CRC-8 algorithm to generate the CRC. To use a different algorithm, uncheck the **Use SMBus CRC-8** box and enter the desired CRC polynomial in the **CRC Polynomial** box in hexadecimal. The length of the polynomial (and the resulting CRC) is selected using the **CRC Length** field. The initial value of the CRC can be set using the **Initial Value** field.

2.1.12. Help

The Help menu provides direct access to all documents and tutorial videos. You can also go to the Keterex website or send an email to Keterex support from Help.

2.2. Data Editors

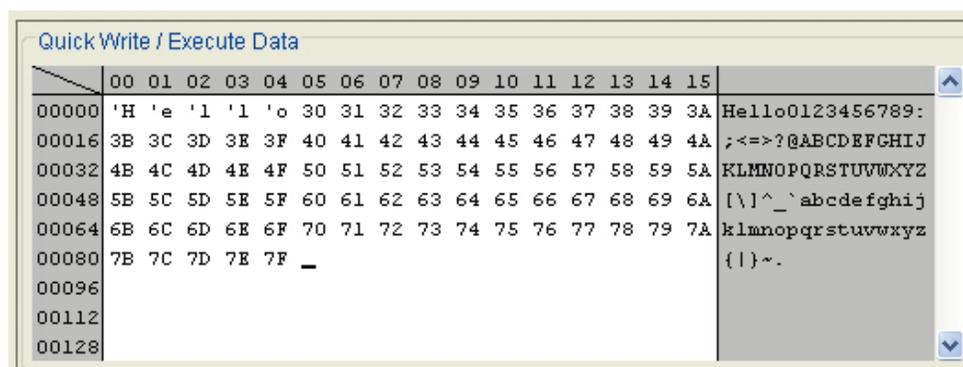
The various feature panes in the Control Center use editors to allow the user to enter outgoing data and scripts and view incoming data. An example editor is shown below.



The data is displayed left-to-right starting in the upper left corner. Each row is labeled with the index of the first byte in that row (shown in the grey box). Each column is labeled with the offset relative to this index. For instance, in the example above the byte 0x09 is at index $22+02 = 24$. The row and column labels in this example are displayed as decimal numbers. The user can force the Control Center to display the row and column labels as hexadecimal values by changing the **Settings** → **Display HEX Labels in Editors** menu item (see Section 2.1.5). However, the data is always shown in a hexadecimal format (or by using two-character tokens as explained later).

Data equivalent to the ASCII code for a character can be entered by preceding each character with an apostrophe ('). For example, 'A' equates to the hex value 0x41 (see the example editor display below). The apostrophe must be the first character of the character pair. The editor also supports string-entry mode in which each character typed is automatically preceded by an apostrophe. String entry mode is toggled by typing **Ctrl-'** (typing an apostrophe while Ctrl is held down) or right-clicking the mouse and clicking **String Entry Mode**. The cursor color changes to red when in string entry mode.

Each data editor can also display the ASCII equivalent characters for HEX data, as shown below. Display of this ASCII text can be enabled or disabled for each editor pane by right-clicking over the editor and selecting the **Show ASCII** option.



The data editors have features common to most text editors, including copy, cut, paste, find, undo, redo, goto, etc. Available editor commands (and their associated short-but keystrokes) are shown below. Editor commands can also be performed by right-clicking in the editor pane.

Editor Command	Short-Cut
Undo	Ctrl-Z
Redo	Ctrl-Y
Copy	Ctrl-C
Cut	Ctrl-X
Paste	Ctrl-V
Delete	Delete

Select All	Ctrl-A
Find	Ctrl-F
Find Next	F3
Find Next Error	F7
Goto	Ctrl-G

Regions of data can be selected using either keystrokes or the mouse. A selected region of data can be copied, deleted, etc. Copied or cut data is moved to the system clipboard. This allows data to be moved between the Control Center and other applications on the host computer. Data in an editor can also be printed by right-clicking over the editor and selecting **Print Data...**

2.3. I/O Controls and Status

The real-time status of all pins of the connected USB-910H Adapter is shown at the upper right of the Control Center (see the example below).



The top row of this region shows the actual state of each pin of the adapter, regardless of whether the pin is an input, an output, or whether the pin is presently controlled by an enabled feature. Pins whose state is logic-high are identified by the symbol **H**. Pins whose state is logic-low are identified by the symbol **L**. If a pin is presently configured as a general-purpose output, its output state can be toggled by clicking on this state symbol.

The check box below the pin state shows the present direction of the adapter pin, regardless of whether the pin is used by a feature or as general-purpose I/O. If an input, the **OE** check box will be clear. If an output, the **OE** check box will be checked. If available as a general-purpose I/O (i.e. not controlled by an enabled feature), the direction of a pin can be toggled by clicking the **OE** check box.

2.4. Adapter Status

The present status of the connected adapter is shown in the lower right corner of the Control Center window (see the example below).



When a requested operation is pending (e.g. a master or slave bus transaction), the sweeper on the right will be active). In addition, zero or more of the following messages below may appear depending on the state of the adapter.

- **I2C Master Pending** – Indicates that a requested I2C/SMBus Master Transaction is either pending or still in progress.
- **I2C Slave Pending** – Indicates that a requested I2C/SMBus Slave Transaction is either pending or still in progress.
- **I2C Master Lost Arbitration** – Indicates that the I2C/SMBus Master lost arbitration during a requested transaction. Depending on the feature configuration, the master will either retry the transaction or abort.
- **SPI Master Pending** – Indicates that a requested SPI Master Transaction is either pending or still in progress.
- **SPI Slave Pending** – Indicates that a requested SPI Slave Transaction is either pending or still in progress.

2.5. Messages

The user is notified of major events by lines generated in the Message Log at the bottom of the Control Center (see the example below).

Messages			
Time	Event	Outgoing Data	Incoming Data
10:13:30	SPI Master Executing		
10:13:58	SPI Master Execution completed	0102030405060708090A0B...	0102030405060708090A0B...
10:14:06	SPI Master Executing		
10:14:07	SPI Master Execution completed	0102030405060708090A0B...	0102030405060708090A0B...
10:14:17	SPI Master Executing		
10:14:18	SPI Master Execution completed	0102030405060708090A0B...	0102030405060708090A0B...

Each message contains a time stamp (with a format of hours, minutes, and seconds), an event summary, and a summary of any outgoing or incoming data associated with the event. Messages are color-coded: informational messages are colored green, events associated with feature transactions (i.e. those with incoming and/or outgoing data) are colored blue, and events associated with errors are colored red. New messages are highlighted with a yellow background for a few seconds before fading to white to distinguish them from older messages.

Messages can be deleted by selecting one or more messages in the Message Log and pressing the Delete key. Note that the Control Center only keeps the last 500 messages. Once 500 messages are reached, the oldest message is automatically deleted. The entire Message Log can be saved, restored, and cleared using commands found under the **Messages** menu.

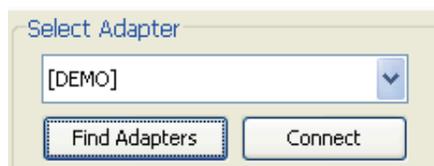
In addition to providing feedback to the user, messages can be used to save the results of a previous operation or even repeat a previous operation. The user can double-click on any blue message to restore the feature pane that generated that message back to its state just after the message was generated. This includes all configuration settings, outgoing data, and incoming data. When a message is restored, all configuration settings (e.g. LSB First, Free Bus Detection, etc.) are automatically restored to the connected adapter. Although the Message Log shows only a summary of the data actually associated with the reported event, double-clicking the message restores all data available at the time to the pane. For example, the outgoing editor contents will be restored even if the message was associated with a read transaction.

This feature allows a transaction to be repeated by double-clicking on a message and then re-executing the operation by clicking on the appropriate button on the feature pane. The data and configuration associated with the message can also be saved independently by clicking the **Save** button on the appropriate feature pane. However, double-clicking on the message does not change which features are presently enabled for the connected adapter.

2.6. Configuring an Adapter

2.6.1. Finding Adapters

The first step in using the USB-900 Control Center is identifying what adapters are physically connected to the host computer. This is accomplished by clicking the **Find Adapters** button near the upper left of the **Configure** tab of the main window.



This button will populate the drop-down menu above the button with all available adapters (plus a demo adapter). An adapter is considered available if it is not already allocated to a Control Center utility or another USB-910H application.

The available adapters are identified by six-digit numbers (e.g. “000100”) which match numbers found on barcode labels on the bottom of each adapter. In addition, an adapter identified as “[DEMO]” is always available which can be used to experiment with the Control Center utility without having an actual adapter connected to the host computer.

2.6.2. Connecting to an Adapter

To connect the Control Center to an Adapter, a specific adapter must be selected from the drop-down list and opened for communications by clicking the **Connect** button (shown above). This adapter will no longer appear as an available adapter in other Control Center utilities running on the same host computer. An adapter must be connected before other operations can be performed, such as setting its I/O voltage. Connecting to an adapter will force the Control Center to revert all configurations back to the defaults of the adapter.

2.6.3. Setting the I/O Voltage

Generally, the next step in setting up an adapter is to set its I/O voltage level. This is the voltage level at which the adapter will drive its logic-high outputs. This level also controls the sense thresholds of incoming logic levels. In addition, the I/O voltage sets the VDD voltage connected to the target system when enabled. The I/O voltage is setup using the controls shown below (found on the **Configure** pane).



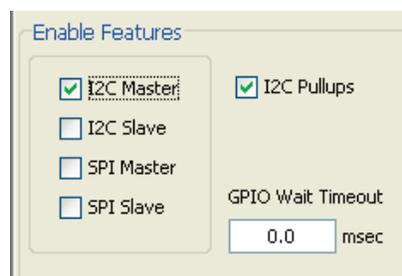
Use the following steps to set the I/O voltage to the desired value:

1. If the desired I/O voltage is between 1.65V and 3.6V, enter the voltage in the left box. If the desired I/O voltage is 5V, check the **+5 Volts** box. No other voltage values are available.
2. Click the **Set Voltage** button to set the actual adapter I/O voltage. *The I/O voltage is not changed until the **Set Voltage** button is clicked.*

The configured I/O voltage can also be supplied to the target system (through the GP3 pin) by checking the **Enable Target VDD** check box at the top of the Control Center. However, to prevent this feature from being used un-intentionally, it is inhibited by default. To allow this feature, uncheck the **Inhibit VDD Enable** check box on the **Configure** pane. At this point, the **Enable Target VDD** check box can be toggled at any time to enable or disable power to the target system.

2.6.4. Enabling a Feature

The Keterex USB-910H Adapter can operate as an I2C/SMBus Master, SPI Master, I2C/SMBus Slave, SPI Slave, or none of the above. Each of these modes is referred to as a feature. At any time, pins not claimed by an enabled feature operate as general-purpose I/O. To enable a feature for the connected adapter, click the appropriate check-box, as shown below. Once a feature is enabled, its respective tab will be bold and green.



At any given time, only one of the four features on the left can be enabled. However, the I2C Pullups feature can be enabled at any time, regardless of whether the adapter is serving as an I2C Master or Slave device. Depending on the requirements of the target system, it might be desirable to configure the feature (see the sections below) before enabling it on the **Configure** pane.

2.6.5. GPIO Wait Timeout

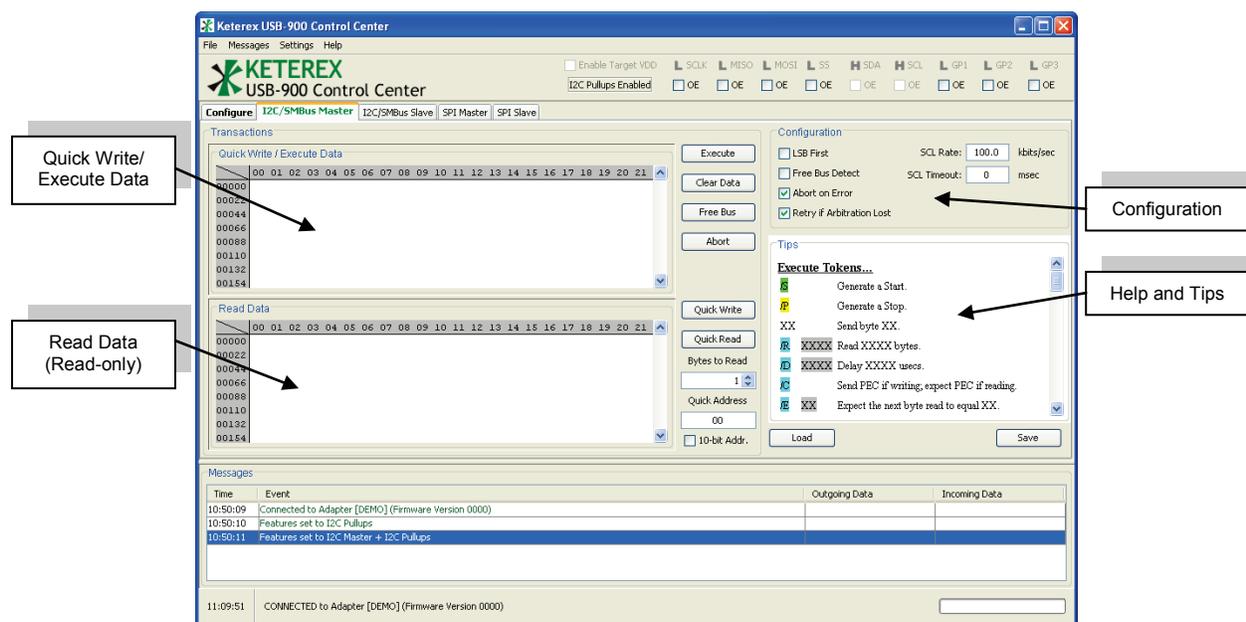
The scripting feature available in the USB-910H Adapter allows execution of a bus transaction to stall until a specified adapter pin is sensed either high or low. The **GPIO Wait Timeout** field controls how long the adapter is allowed to wait before aborting the operation. This value can be set between 0 and 8388msec. If set to 0, the timeout is disabled, i.e. the adapter will wait indefinitely for the specified pin to achieve the required logic level (or until the operation is manually aborted).

2.7. The I2C/SMBus Master Feature

The USB-910H Adapter can operate as an I2C/SMBus Master device fully compatible with the I2C and SMBus standards, including multi-master requirements, automatically-generated packet error checks, repeated-starts, group operations, 10-bit addressing, etc. Master operations can be performed using “quick” read and write operations or by using a simple scripting language which supports a combination of bus operations, general-purpose I/O operations, and timing controls.

To use the USB-910H Adapter as an I2C/SMBus Master device, the **I2C Pullups** should be enabled if needed and then the **I2C Master** feature should be enabled (see the figure above). Click the respective tab to move to the I2C/SMBus Master feature pane.

The I2C/SMBus Master feature pane is shown below.



A Master operation is generally performed by entering either HEX data or a script into the top editor pane (the **Quick Write / Execute Data** editor) and clicking either the **Execute** button or the **Quick Write** button. A simple Master read operation can also be performed by setting the target address and the number of bytes to read and clicking the **Quick Read** button. Details on each type of Master operation are provided in the following sections.

2.7.1. The Editor Panes

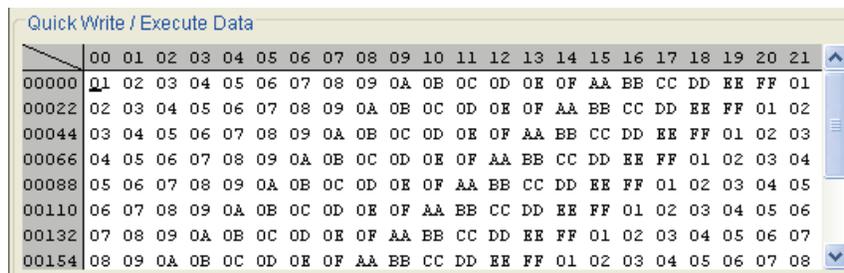
The I2C/SMBus Master pane has two data editors. The top editor is used to enter outgoing write data and scripts to be executed by the adapter. The lower pane holds incoming data resulting from read operations. The Read Data editor is read-only and supports only non-destructive operations, such as copy, find, etc.

2.7.2. Quick Writes

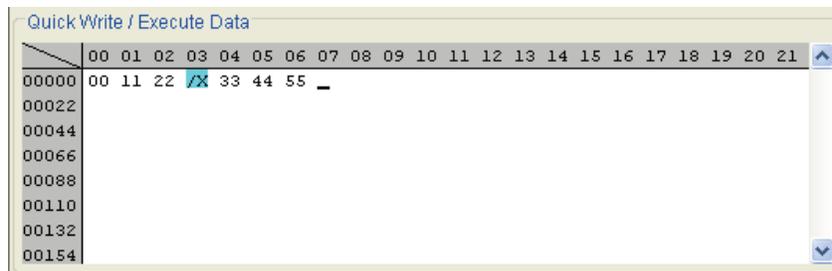
Quick Writes allow the user to quickly write data to a target slave device. To perform a quick write, enter the outgoing data in the top editor pane. You must also enter the address of the target slave device in the **Quick Address** box. If the slave is expecting a 10-bit address, check the **10-bit Addr.** box below the address. Clicking the **Quick Write** button will initiate the write operation.

A message will be generated reporting that the operation was initiated. When the operation is no longer pending, a second message will be generated reporting the results of the operation (e.g. whether the operation was successful, was aborted, whether an error occurred, etc.).

Generally, Quick Write data consist of simple hexadecimal data, such as that shown below.



However, the USB-910H adapter does support one special token in Quick Write data. The **/X** token will cause the Control Center to prompt for a file and insert the contents of that file into the write data of the device. This file can contain any valid hexadecimal data or other **/X** tokens, and any white-space is automatically omitted. However, the included file must have an even-number of characters (in order to create complete bytes). For example, consider the following data editor contents:



Upon encountering the **/X**, the Control Center will prompt for a file. If that file contains the text:

```
55 AA 5A A5
00 FF F0 0F
```

The complete write data for the device will be set to:

```
00 11 22 55 AA 5A A5 00 FF F0 0F 33 44 55
```

2.7.3. Quick Reads

Quick Reads allow the user to quickly read data from a target slave device. To perform a quick read, set the number of bytes to read using the **Bytes to Read** field and enter the target slave address in the **Quick Address** field. If the slave is expecting a 10-bit address, check the **10-bit Addr.** box below the address. Clicking the **Quick Read** button will initiate the read operation. The contents of the **Quick Write/Execute Data** editor are not used during a Quick Read operation.

A message will be generated reporting that the operation was initiated. When the operation is no longer pending, a second message will be generated reporting the results of the operation (e.g. whether the operation was successful, was aborted, whether an error occurred, etc.). When complete, the Control Center will automatically display the read data in the **Read Data** editor pane.

2.7.4. Execute

The USB-910H Adapter can execute simple scripts which allow the user to interleave bus, I/O, delay, and configuration operations. Scripts consist of a series of two-character tokens. A token can be a simple hexadecimal byte, a special command token, or a command parameter. The **Execute Data** editor color-codes the script to help identify the type of each token in the script. The following tokens are supported:

<i>I2C/SMBus Master Token Sequence</i>	<i>What the I2C/SMBus Master Does</i>
/S	Generates a START condition on the bus.
/P	Generates a STOP condition on the bus.
a HEX byte	Transmits the byte on the bus.
'X	Transmits the 8-bit ASCII code for character X.
/R XX XX	Reads XXXX number of bytes, where XXXX is a hexadecimal number (MSB first).
/D XX XX	Delay XXXX μ sec before continuing, where XXXX is a hexadecimal number (MSB first).
/C	If writing, write the correct PEC byte to the bus. If reading, read the expected PEC byte from the bus. If the read PEC is incorrect, assert an error. If Abort on Error is set, also abort.
/E XX	Compares the previously read byte to the value XX. If not equal, assert an error. If Abort on Error is set, also abort.
/F CF	Set the I2C Configuration to CF, where CF is an OR of the following values (see Section 2.7.6): 0x01 – Transmit data LSB first. 0x02 – Abort on Error. 0x04 – Retry if Arbitration Lost. 0x08 – Free Bus Enable. Changes to the configuration will be reflected in the Control Center.
/L PN	Set adapter pin PN as an input and wait until it is sensed as a Low, where PN is: SK = SCLK MI = MISO MO = MOSI SS = SS SD = SDA SL = SCL G1 = GP1 G2 = GP2 G3 = GP3 NO = NONE (the token is ignored)
/H PN	Set adapter pin PN as an input and wait until it is sensed as a High, where PN is from the table above.

<code>/0 PN</code>	Set adapter pin PN as an output and drive High, where PN is from the table above (excluding the SDA and SCL pins)*.
<code>/1 PN</code>	Set adapter pin PN as an output and drive High, where PN is from the table above (excluding the SDA and SCL pins)*.
<code>/Z PN</code>	Set adapter pin PN as High-Z (i.e. an input), where PN is from the table above (excluding the SDA and SCL pins)*.
<code>/X</code>	Prompt for a file to insert into the script. This file can contain any valid script data, including additional <code>/X</code> tokens, white space, and carriage returns.

*Note that the `/0`, `/1`, and `/Z` commands are not allowed to use the SDA and SCL pins because the I2C feature has control over these pins.

For example, the following script writes the data 0x55 to address 0x01:

```
/S 02 55 /P
```

Note that since I2C left-justifies the 7-bit address on the bus (the LSB of the byte is the Read/Write bit), the 0x01 address appears as 0x02 in the script. Below are a set of other helpful examples:

<code>/S 03 /R 00 10 /P</code>	Read 16 bytes from address 0x01 (appears as 0x03 after left-justify and adding the read bit).
<code>/S 80 55 /S 81 /R 01 00 /C /P</code>	Write 0x55 to address 0x40, issue a repeat Start, then read 256 bytes from address 0x40 and expect a PEC.
<code>/S FE AA /D 00 64 /C /P</code>	Write 0xAA to 7-bit address 0x7F, delay 100µsec, send a PEC and stop.
<code>/H G1 /S 22 FF /P /1 G2</code>	Wait for GP1 to go High, Write 0xFF to address 0x11, then drive GP2 High.
<code>/S FE 80 55 /P</code>	Write 0x55 to 10-bit address 0x380.

2.7.5. Loading and Saving

The contents of the I2C/SMBus Master pane can be saved and restored (loaded) to/from a file. This includes the contents of both editors and all settings. The **Load** and **Save** buttons open a file browser to allow the user to select a file for the operation. The pane contents are stored as a text XML file. If the pane is restored from a file, the Control Center automatically attempts to store the new configuration to the connected adapter.

To save just the contents of one of the editors, click in the editor and type Ctrl-A, then Ctrl-C. This selects and copies the contents of the editor to the system's clipboard. You can then use Ctrl-V to paste the editor contents into any system application, such as a word processor or spread-sheet program.

2.7.6. Configuring the I2C Master Adapter

The USB-910H Adapter supports several modes when operating as an I2C/SMBus Master device. These modes are configured using the controls shown below. Changing these values also changes their corresponding value on the I2C/SMBus Slave tab.

When enabled, the **LSB First** control causes the adapter to transmit and expect all data bytes least-significant-bit first. However, this has no effect on the transmission of address bytes.

When enabled, the **Free Bus Detect** control causes the adapter to support the SMBus requirement that the bus be considered free if the SCL signal remains high for more than 50 μ sec. More specifically, the adapter will consider the bus free if both the SCL and SDA lines remain high for more than approximately 3.3 SCL clock periods, according to the present SCL Rate setting.

When enabled, the **Abort on Error** control causes the adapter to abort an operation if an expected value (e.g. a PEC or **E** value) is incorrect. Otherwise, the operation is continued. Regardless, an error flag is set, causing an error message to be logged.

When enabled, the **Retry if Arbitration Lost** control causes the adapter to continue to retry an operation if it loses arbitration while attempting the operation. In this case, a status flag is set to indicate that arbitration was lost. If disabled and arbitration is lost, the adapter will abort the operation and generate an error.

The **SCL Rate** field sets the approximate rate of the SCL clock. Supported values range from 31.25 kHz to 1.5 MHz. The actual maximum rate at which reliable communications can be maintained depends on the target system.

The **SCL Timeout** field sets the period of time the SCL signal is allowed to remain low before the adapter declares a bus error and resets its I2C interface. The SMBus standard requires that this time be between 25 and 35 msec. Supported timeout values range from 0 to 32.7 msec. A value of 0 disables the timeout feature, i.e. sets the timeout to infinity. When enabled, the adapter will assert an error and immediately re-initialize its I2C interface if the SCL signal remains low longer than the timeout value.

2.7.7. Abort and Free Bus

Any master bus operation can be aborted by clicking the **Abort** button. This instructs the adapter to attempt to cleanly terminate the requested operation. This causes the adapter to abort after the next byte and generate a stop.

The **Free Bus** button is similar to an abort but it causes the adapter to terminate an operation by immediately resetting the I2C interface without attempting to generate a stop condition.

2.8. The I2C/SMBus Slave Feature

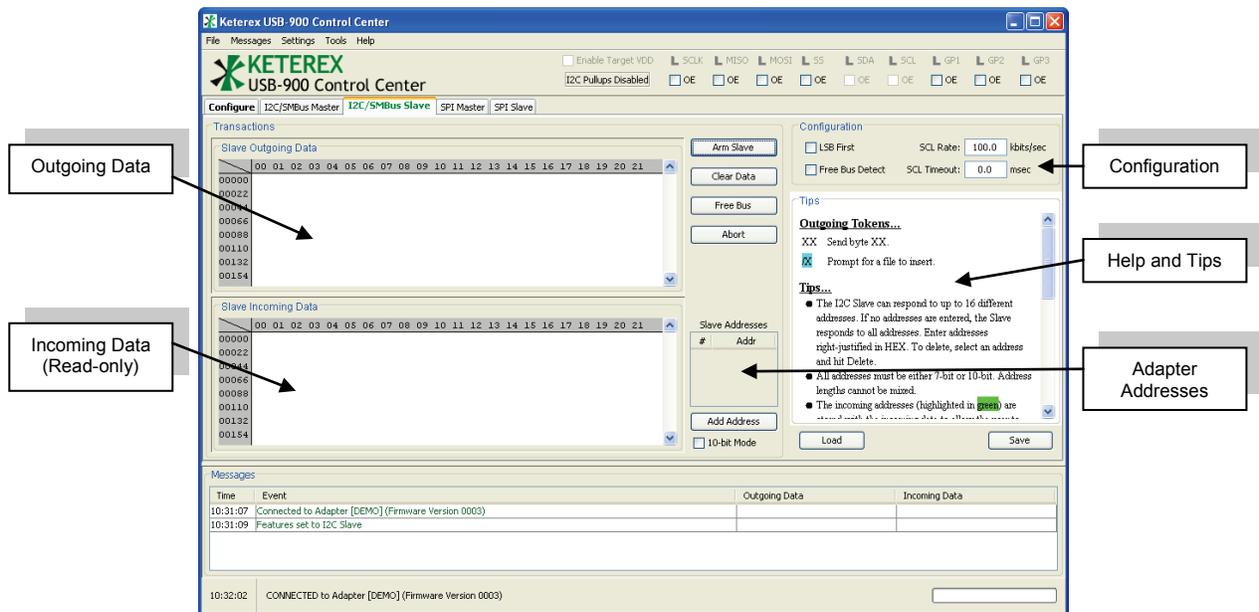
The USB-910H Adapter can operate as an I2C/SMBus Slave device fully compatible with the I2C and SMBus standards, including automatically-generated packet error checks, repeated-starts, group operations, 10-bit addressing, etc. A slave can respond to up-to 16 specific 7-bit or 10-bit addresses, or can respond to all 7-bit or 10-bit address.

To use the USB-910H Adapter as an I2C/SMBus slave device, the **I2C Pullups** should be enabled if needed and then the **I2C Slave** feature should be enabled, as shown below (found on the **Configure** pane).



At this point, the tab for the I2C/SMBus Slave feature pane should be bold and green. This indicates that this feature is active. Click the tab to move to the I2C/SMBus Slave feature pane.

The I2C/SMBus Slave feature pane is shown below.



A slave operation is generally performed by setting the addresses for the slave device, entering any desired outgoing (i.e. read) data in the top editor pane (the **Slave Outgoing Data** editor) and clicking the **Arm Slave** button. Details on slave operations are provided in the following sections.

2.8.1. The Outgoing and Incoming Data Editors

The I2C/SMBus Slave pane has two data editors. The top editor is used to enter outgoing data (i.e. read data) to be sent by the slave device during bus read operations. The lower editor pane shows incoming data sent to the slave device during bus write operations. This read-only editor pane also shows which addresses the adapter responded to. These addresses are included in the data stream in the same position as received from the bus (and with the same formatting and justification). Address bytes are color-coded green to differentiate them from data bytes.

2.8.2. Setting the Adapter Addresses

As an I2C/SMBus slave device, the USB-910H Adapter can respond to up-to 16 specific 7-bit or 10-bit address, or can respond to all 7-bit or all 10-bit addresses. To respond to specific addresses, add each address by clicking the **Add Address** button and enter the address value (in HEX) in the table (shown below). All addresses are entered right-justified. The adapter automatically left-justifies as needed.

#	Addr
4	20
5	22
6	40
7	44

Add Address

10-bit Mode

Addresses can be deleted by selecting one or more addresses in the table (hold the Ctrl key down to add/remove address from the selection, hold Shift to select a range) and typing the Delete key. You can edit an address by double-clicking the address and typing a new value.

If all addresses are to be interpreted as 10-bit values, check the **10-bit Mode** box below the **Add Address** button. All addresses are either 7-bit or 10-bit – the adapter can not respond to a mix of 7-bit and 10-bit addresses.

To respond to all 7-bit or 10-bit addresses, delete all addresses from the table (type Ctrl-A in the table and type Delete). If the adapter is to respond to all 10-bit addresses, check the **10-bit Mode** box. Otherwise, clear the **10-bit Mode** box to respond to all 7-bit addresses.

To allow the user to determine which address the adapter responded to, the incoming addresses are stored along with the incoming data (and color-coded). See Section 2.8.1 for more information.

2.8.3. Outgoing Data

The outgoing data provided by the slave device during a bus read operation is entered in the top editor pane (the Slave Outgoing Data editor). This editor accepts either simple hexadecimal data and/or the **/X** token to insert a file containing hexadecimal data (see Section 2.7.2 for more information on the **/X** token). If a master device requests more data than is provided in the Outgoing Data editor, the adapter sends its PEC byte until the next STOP condition is received.

2.8.4. Arming the Slave

The adapter does not begin listening for addresses on the bus until it is “armed”. To arm the slave, simply click the **Arm Slave** button. At this point the slave is active, waiting to be addressed by a master device on the bus. The slave remains armed until either it is successfully addressed or until the operation is aborted. Once addressed, the slave disarms after the next stop condition is detected on the bus (or the bus is considered free due to the **Free Bus Detect** control – see Section 2.8.5), or after another device is addressed after a repeated START. The Control Center will automatically display any incoming addresses and data that resulted from the completed operation.

For example, consider the following incoming data example:

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21
00000	00	0E	AA	AD	CF	D5	B0	CA	27	FF	D8	9D	14	4D	F4	79	27	59	42	7C	9C	C1
00022	F8	CD	8C	87	20	23	64	B8	A6	87	95	4C	B0	5A	8D	4E	2D	99	E7	3D	B1	60
00044	DE	B1	80	AD	08	41	E9	01														
00066																						
00088																						
00110																						
00132																						
00154																						

The adapter responded to a write to address 0x00 (left-justified, color-coded green), received 50 data bytes (values 0x0E to 0xE9), then responded to a read of address 0x00 (color-coded green).

In the following example, the adapter was configured to respond to the 10-bit address 0x380:

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21
00000	FF	80	31	92	74	AF	8D	B5	A4	90	D5	5E	6A	40	FC	80	76	02	4B	17	6B	36
00022	E1	21	DB	7D	5A	EA	72	1E	82	8D	71	A8	8C	B8	5E	D9	4E	AF	FA	BF	B0	94
00044	74	1D	75	E5	DC	10	58	46	DA	F2	5B	81	A0	7F	5C	CB	1D	36	E9	49	74	02
00066	55	D2	AC	1A	0B	F7	A9	26	23	40	5B	A3	33	B9	35	88	68	AD	E1	2A	D5	B2
00088	32	5D	0A	E5	5A	DC	E9	77	5D	EB	B5	69	C5	3A	6C	93	98	0D	57	EB	87	9A
00110	DF	04	68	B2	A2	D5	E6	A4	C6	BC	77	5F	8D	C3	8F	D6	2A	21	14	A9	D4	04
00132	11	01	18	8D	AE	BB	73	1C	60	CA	20	CF	5D	D6	2F	45	53	29	D7	FF	80	
00154																						

Since 10-bit addresses occupy two bytes on the bus, two bytes are color-coded green for each address in the Incoming Data editor.

2.8.5. Configuring the I2C Slave Adapter

The USB-910H Adapter supports several modes when operating as an I2C/SMBus slave device. These modes are configured using the controls shown below. Changing these values changes the respective values on the I2C/SMBus Master tab.

When enabled, the **LSB First** control causes the adapter to transmit and expect all data bytes least-significant-bit first. However, this has no effect on the reception of address bytes.

When enabled, the **Free Bus Detect** control causes the adapter to support the SMBus requirement that the bus be considered free if the SCL signal remains high for more than 50µsec. More specifically, the adapter will consider the bus free if both the SCL and SDA lines remain high for more than approximately 3.3 SCL clock periods, according to the present SCL Rate setting.

The **SCL Rate** field sets the approximate rate of the SCL clock. Supported values range from 31.25 kHz to 1.5 MHz. For slave mode, this value is only used to determine the timing for the **Free Bus Detect** control. If **Free Bus Detect** is not enabled, the **SCL Rate** value is not used.

The **SCL Timeout** field sets the period of time the SCL signal is allowed to remain low before the adapter declares a bus error and resets its I2C interface. The SMBus standard requires that this time be between 25 and 35 msec. Supported timeout values range from 0 to 32.7 msec. A value of 0 disables the timeout feature, i.e. sets the timeout to infinity. When enabled, the adapter will assert an error and immediately reset its I2C interface if the SCL signal remains low longer than the timeout value.

2.8.6. Abort and Free Bus

A pending slave operation can be aborted by clicking the **Abort** or **Free Bus** buttons. The **Abort** button attempts to cleanly abort the operation by refusing (i.e. NACKing) all future bytes on the bus. The **Free Bus** button immediately resets the I2C interface on the adapter.

2.8.7. Loading and Saving

The contents of the I2C/SMBus Slave pane can be saved and restored (loaded) to/from a file. This includes the contents of both editors and all other settings. The **Load** and **Save** buttons open a file

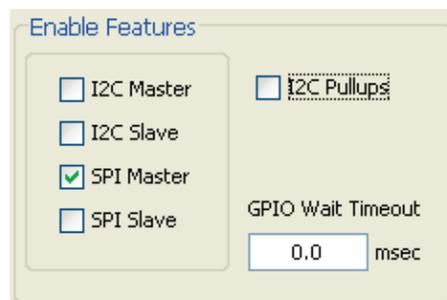
browser to allow the user to select a file for the operation. The pane contents are stored as a text XML file. If the pane is restored from a file, the Control Center automatically attempts to store the new configuration to the connected adapter.

If the user desires to save just the contents of one of the editors, click in the editor and type Ctrl-A, then Ctrl-C. This selects and copies the contents of the editor to the system's clipboard. You can then use Ctrl-V to paste the editor contents into any system application, such as a word processor or spread-sheet program.

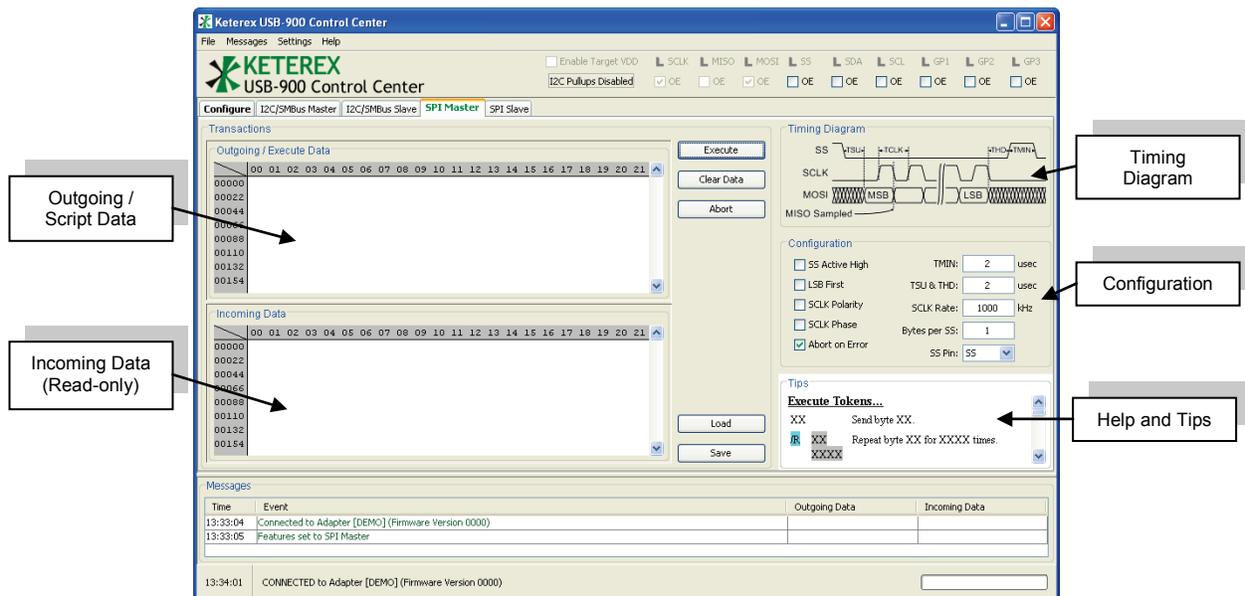
2.9. The SPI Master Feature

The USB-910H Adapter can operate as a SPI Master device fully compatible with SPI, Microwire, and many other three and four wire synchronous serial interfaces.

To use the USB-910H Adapter as an SPI Master device, the **SPI Master** feature should be enabled, as shown below (found on the **Configure** pane).



At this point, the tab for SPI Master feature pane should be bold and green. This indicates that this feature is active. Click the tab to move to the SPI Master feature pane. The SPI Master feature pane is shown below.



A SPI Master operation is generally performed by entering either HEX data or a script into the top editor pane (the **Outgoing Data / Execute Data** editor) and clicking the **Execute** button. Details on Master operations are provided in the following sections.

2.9.1. The Editor Panes

The SPI Master pane has two editor panes. The top pane is used to enter outgoing data and scripts to be executed by the adapter. The lower pane holds read-only incoming data.

2.9.2. Execute

The USB-910H Adapter can execute simple scripts which allow the user to interleave bus, I/O, delay, and configuration operations. Scripts consist of a series of two-character tokens. A token can be a simple hexadecimal byte, a special command token, or a command parameter. The Execute Data editor color-codes the script to help identify the type of each token in the script. The following tokens are supported:

SPI Master Token Sequence	What the SPI Master Does
a HEX byte	Send the byte on the bus using the current configuration.
'X	Send the 8-bit ASCII code for character X.
/R BB XX XX	Send the byte BB XXXX number of times, where XXXX is a hexadecimal number (MSB first). This can be used to retrieve data from a slave where the data into the slave is not important. If XXXX is 0000, the byte is sent 0x10000 times.
/D XX XX	Delay XXXX μ sec before continuing, where XXXX is a hexadecimal value (MSB first).
/E XX	Expect the next incoming byte to equal XX. If not, assert and error. If Abort on Error is set, also abort.
/F CF XX XX	Set the SPI Configuration to CF, where CF is an OR of the following values: 0x01 – Transmit data LSB first. 0x02 – Abort on Error. 0x04 – Assert SS Active High. 0x08 – SCLK Phase. 0x10 – SCLK Polarity. The value XXXX sets the number of bytes to transfer per SS assertion cycle (where XXXX is a hexadecimal value provided MSB first). This configuration is used until changed and changes to the configuration will be reflected in the Control Center.
/L PN	Set adapter pin PN as an input and wait until it is sensed as a Low, where PN is: SK = SCLK MI = MISO MO = MOSI SS = SS SD = SDA SL = SCL G1 = GP1 G2 = GP2 G3 = GP3 NO = NONE (the token is ignored)
/H PN	Set adapter pin PN as an input and wait until it is sensed as a High, where PN is from the table above.
/O PN	Set adapter pin PN as an output and drive low, where PN is from the table above (excluding the SCLK, MISO, and MOSI pins)*.
/1 PN	Set adapter pin PN as an output and drive High, where PN is

	from the table above (excluding the SCLK, MISO, and MOSI pins)*.
/Z PN	Set adapter pin PN as High-Z (i.e. an input), where PN is from the table above (excluding the SCLK, MISO, and MOSI pins).
/X	Prompt for a file to insert into the script. This file can contain any valid script data, including additional /X tokens, white space, and carriage returns.

*Note that the **/0**, **/1**, and **/Z** commands are not allowed to use the SCLK, MISO, and MOSI pins because the SPI feature has control over these pins.

Below are a set of helpful examples:

11 22 33 44	Send the data 0x11, 0x22, 0x33, 0x44 to a slave.
/H G1 00 FF /1 G2	Wait for GP1 to go High, send bytes 0x00 and 0xFF, then drive GP2 High.
55 AA 5A A5 /E EE	Send four bytes, expect the last received byte to equal 0xEE.
55 /D 01 00 AA	Send 0x55, wait 256 μ sec, then send 0xAA.
/L G1 /X /0 G3	Wait for GP1 to go low, prompt for a file to send, then drive GP3 low.

2.9.3. Loading and Saving

The contents of the SPI Master pane can be saved and restored (loaded) to/from a file. This includes the contents of both editors and all other settings. The **Load** and **Save** buttons open a file browser to allow the user to select a file for the operation. The pane contents are stored as a text XML file. If the tab is restored from a file, the Control Center automatically attempts to store the new configuration to the connected adapter.

If the user desires to save just the contents of one of the editors, click in the editor and type Ctrl-A, then Ctrl-C. This selects and copies the contents of the editor to the system's clipboard. You can then use Ctrl-V to paste the editor contents into any system application, such as a word processor or spread-sheet program.

2.9.4. Configuring the SPI Master Adapter

The USB-910H Adapter supports several modes when operating as a SPI Master device. These modes are configured using the controls shown below. In general, the timing diagram above the Configuration pane updates to reflect the configuration settings. Changing these values also changes the respective values on the SPI Slave tab (except the SS Pin setting).

Configuration

<input type="checkbox"/> SS Active High	TMIN: <input type="text" value="2"/> usec
<input type="checkbox"/> LSB First	TSU & THD: <input type="text" value="2"/> usec
<input type="checkbox"/> SCLK Polarity	SCLK Rate: <input type="text" value="1000"/> kHz
<input type="checkbox"/> SCLK Phase	Bytes per SS: <input type="text" value="1"/>
<input checked="" type="checkbox"/> Abort on Error	SS Pin: <input type="text" value="SS"/>

When enabled, the **SS Active High** control causes the adapter to hold the selected Slave-Select pin low when de-asserted and high when asserted. Otherwise, the pin is driven high when de-asserted and low when asserted.

When enabled, the **LSB First** control causes the adapter to transmit and expect all bytes least-significant-bit first.

When enabled, the **SCLK Polarity** control causes the SCLK signal to idle high. Otherwise, the SCLK signal idles low.

When enabled, the **SCLK Phase** control causes the MOSI signal to transition on the leading edges of the SCLK signal (i.e. the first edge leaving the idle state) and the MISO signal is sampled starting on the second leading edge. When disabled, the MOSI signal transitions on the trailing edges of the SCLK signal and the MISO signal is sampled on the trailing edges of the SCLK signal.

When enabled, the **Abort on Error** control causes the adapter to abort an operation if an expected value (i.e. a **!E** value) is incorrect. Otherwise, the operation is continued. Regardless, an error flag is set causing an error message to be logged.

The **TMIN** period controls the minimum time the adapter will leave the Slave-Select pin de-asserted before continuing an operation. Supported values range from 2 μ sec to 55 μ sec.

The **TSU&THD** period controls the minimum time the adapter will delay between a transition in the Slave-Select pin and the beginning or ending of MOSI traffic. Supported values range from 2 μ sec to 55 μ sec.

The **SCLK Rate** field sets the approximate rate of the SCLK clock. Supported values range from 94 kHz to 24 MHz. The actual maximum rate at which reliable communications can be maintained depends on the target system.

The **Bytes per SS** field controls how many bytes the master will send on the bus each time the Slave-Select pin is asserted. After this many bytes are transferred, the adapter will delay the **TSU&THD** period and de-assert the **SS Pin**. If more data remains to be transferred, the adapter will delay the **TMIN** period and re-assert the **SS Pin**, then continue transferring data. This process continues until all data is transferred.

The **SS Pin** control selects the adapter pin to assert as a Slave-Select during the bus operation. Although one adapter pin is named SS, this is for convenience. Any available pin can serve as a Slave-Select pin. If no Slave-Select should be asserted, e.g. there is only one slave attached to the master, the **SS Pin** control can be set to **NONE**.

2.9.5. Abort

A pending master operation can be aborted by clicking the **Abort** button. This instructs the adapter to cleanly terminate the requested operation after the next byte.

2.10. The SPI Slave Feature

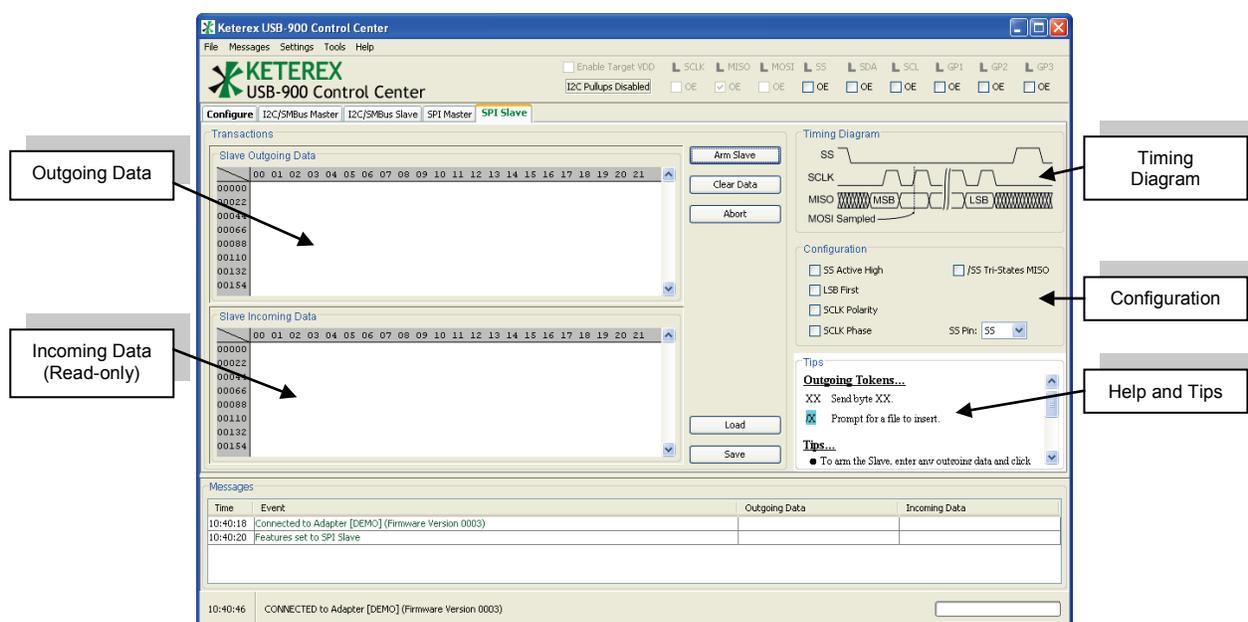
The USB-910H Adapter can operate as a SPI slave device fully compatible with SPI, Microwire, and many other three and four wire synchronous serial interfaces.

To use the USB-910H Adapter as a SPI slave device, the SPI slave feature should be enabled by checking the **SPI Slave** enable check box on the **Configure** pane shown below.



At this point, the tab for the SPI Slave feature pane should be bold and green. This indicates that this feature is active. Click the tab to move to the SPI Slave feature pane.

The SPI Slave feature pane is shown below.



A slave operation is generally performed by setting the Slave-Select pin for the slave device, entering the desired outgoing data in the top editor pane (the **Slave Outgoing Data** editor) and clicking the **Arm Slave** button. Details on slave operations are provided in the following sections.

2.10.1. The Outgoing and Incoming Data Editors

The SPI Slave pane has two data editors. The top editor is used to enter outgoing data to be sent by the slave device during bus operations. The lower read-only pane shows incoming data sent to the slave device during bus operations.

2.10.2. Outgoing Data

The outgoing data provided by the slave device during a bus operation is entered in the top editor pane (the Slave Outgoing Data editor). This editor accepts either simple hexadecimal data and/or the **XX** token to insert a file containing hexadecimal data (see Section 2.7.2 for more information on the **XX** token). If a master device requests more data than is provided in the Outgoing Data editor, the adapter sends the last received byte back to the master until its Slave-Select signal de-asserts.

2.10.3. Arming the Slave

The adapter does not begin listening for its Slave-Select pin until it is “armed”. To arm the slave, simply click the **Arm Slave** button. At this point the slave is active, waiting to be selected by a master device. The slave remains armed until it has transferred its outgoing data and its Slave-Select pin de-asserts or until the operation is aborted. The Control Center will automatically display any incoming data that resulted from the completed operation.

2.10.4. Configuring the SPI Slave Adapter

The USB-910H Adapter supports several modes when operating as an SPI slave device. These modes are configured using the controls shown below. Note that the timing diagram on the SPI Slave pane updates to reflect the present configuration settings. Changing these settings also changes the respective settings on the SPI Master pane (except for the SS Pin selection).



When enabled, the **SS Active High** control causes the SPI slave to begin a transfer when its Slave-Select pin is asserted high. Otherwise, the slave begins a transfer when its Slave-Select pin is asserted low.

When enabled, the **LSB First** control causes the adapter to transmit and expect all data bytes least-significant-bit first. Otherwise, all bytes are transferred most-significant-bit first.

When enabled, the **SCLK Polarity** control causes the slave to expect the SCLK signal to idle high. Otherwise, the SCLK signal is expected to idle low.

When enabled, the **SCLK Phase** control causes the MISO signal to transition on the leading edges of the SCLK signal (i.e. the first edge leaving the idle state) and the MOSI signal is sampled starting on the second leading edge of SCLK. When disabled, the MISO signal transitions on the trailing edges of the SCLK signal and the MOSI signal is sampled on the trailing edge of the SCLK signal.

When enabled, the **/SS Tri-States MISO** control causes the adapter to tri-state its MISO output when not selected by a master, i.e. when its assigned Slave-Select pin is de-asserted. If the adapter has no Slave-Select pin, the MISO will be tri-stated when the adapter is disarmed.

The **SS Pin** control sets the adapter pin to be used as the Slave-Select for the slave device. Although one adapter pin is named SS, this is for convenience. Any available pin can serve as a Slave-Select pin. If no Slave-Select should be used, the **SS Pin** control can be set to **NONE**. In this case, the SPI slave will begin its transfer as soon as SCLK activates, i.e. without waiting for a Slave-Select pin to assert, and it will disarm when all available outgoing data has been transferred.

2.10.5. Abort

A pending slave operation can be aborted by clicking the **Abort** button. The **Abort** button immediately disables the SPI slave device.

2.10.6. Loading and Saving

The contents of the SPI Slave pane can be saved and restored (loaded) to/from a file. This includes the contents of both editors and all other settings. The **Load** and **Save** buttons open a file browser to allow the user to select a file for the operation. The pane contents are stored as a text XML file. If the pane is

restored from a file, the Control Center automatically attempts to store the new configuration to the connected adapter.

If the user desires to save just the contents of one of the editors, click in the editor and type Ctrl-A, then Ctrl-C. This selects and copies the contents of the editor to the system's clipboard. You can then use Ctrl-V to paste the editor contents into any system application, such as a word processor or spread-sheet program.

3. Contact Information

Keterex, Inc.
7320 N. MoPac Expressway
Suite 201
Austin, Texas 78731
Tel: 512-346-8800
www.keterex.com
Email: support@keterex.com

Information furnished by Keterex, Inc. in this document is believed to be accurate and reliable. However, no responsibility is assumed for its use. Information in this document is subject to change without notice.

Trademarks: The Keterex name and logo are registered trademarks of Keterex Incorporated. Other products or brand names mentioned herein are trademarks or registered trademarks of their respective holders.

Use in life support systems: Keterex devices are not designed for use in life support and/or safety equipment where malfunction of the product can result in personal injury or death. Your use or sale of these devices for use in life support and/or safety applications is at your own risk. You agree to defend and hold us harmless from any and all damages, claims, suits or expenses resulting from such use.

Limitation of Liability: You acknowledge and agree that, in no event, shall Keterex be liable, whether in contract, warranty, tort (including negligence or breach of statutory duty), strict liability, indemnity, contribution, or otherwise, for any indirect, special, punitive, exemplary, incidental or consequential loss, damage, cost or expense of any kind whatsoever, howsoever caused, or for any loss of production, cost of procurement of substitute goods, loss of capital, loss of software, loss of profit, loss of revenues, contracts, business, cost of rework, loss of goodwill or anticipated savings, or wasted management time, even if Keterex has been advised of the possibility or they are foreseeable. The total liability of Keterex on all claims, whether in contract, warranty, tort (including negligence or breach of statutory duty), strict liability, indemnity, contribution, or otherwise, shall not exceed the purchase price of these devices.