Aardvark I²C/SPI Control Center

Summary

The Control Center is a graphical application for use with the Aardvark I²C/SPI Host Adapter. It provides access to all I²C , SPI and GPIO functions of the Aardvark adapter in an easy to use graphical interface. Batch scripting capability has been added since version 3.00.

TOTAL PHASE

Aardvark I²C/SPI
Embedded Systems Interface

Control Center User Manual v3.55
September 17, 2013

Contents

List of Figures

List of Tables

1 Overview

The Aardvark I²C/SPI Embedded Systems Interface is a multi-functional host adapter. The Control Center software interacts directly with the Aardvark adapter.

The Control Center provides basic access to all the functionality of the Aardvark adapter. It is built upon the freely available Aardvark I²C/SPI Software API as detailed in the Aardvark I²C/SPI Embedded Systems Interface user manual.

1.1 Changes in version 3.55

- Fixed bug with connecting to some Aardvark adapters.

1.2 Changes in version 3.54

- Added a message when connecting to Aardvark OEM adapters.

1.3 Changes in version 3.53

Bug Fixes

- Fixed an issue with XML i2c_free_bus batch commands causing the GUI to hang.

1.4 Changes in version 3.52

New Features

- XML examples are now included within the GUI package rather than as a separate package.

1.5 Changes in version 3.51

Bug Fixes

- Fixed interoperability issue with FW >= v3.40 when doing 10-bit I2C reads.

New Features

- Added support for 64-bit Windows. Note that Control Center will run on 64-bit systems as a 32-bit application.

1.6 Aardvark I²C/SPI Host Adapter

The Aardvark I²C/SPI Adapter has 4 core functional modules: I²C , SPI, GPIO, and Batch Scripting.

I²C and SPI are serial protocols for communicating between devices. Information about their specifications can be found in the Aardvark I²C/SPI Embedded Systems Interface datasheet and on the Total Phase website: http://www.totalphase.com/

GPIO, General Purpose IO, allows the users to take the six pins that are normally used for I²C and SPI and use them to send and receive general signals. These six pins are SCL, SDA, MOSI, SCLK, MISO, and SS. GPIO functionality can be used concurrently with either I²C or SPI, or can be used by itself.

Batch Scripting allows the user to write scripts in an XML based language to automate tasks. The batch scripting functionality in the Control Center software encompasses a subset of the I²C , SPI and GPIO functionality of the Aardvark adapter.

1.7 Aardvark I²C/SPI Host Adapter Functional Modes

The Aardvark I²C/SPI Host Adapter has six functional modes:

1. I2C + SPI

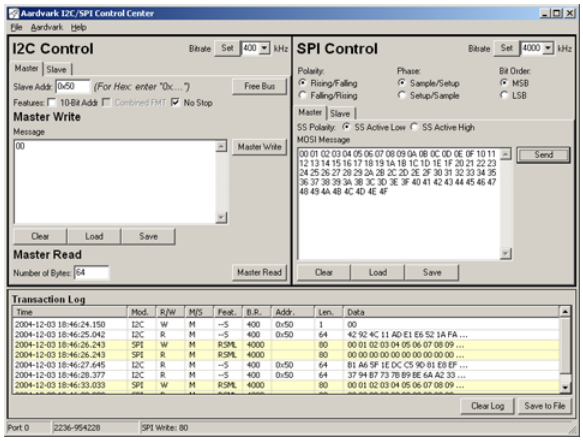

Figure 1: *I²C and SPI*

2. I2C + GPIO

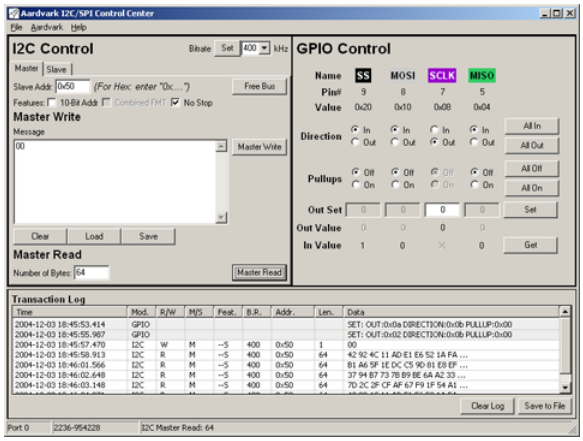

Figure 2: *I²C and GPIO using the 4 available SPI pins: MOSI, SCLK, MISO, SS*

3. SPI + GPIO

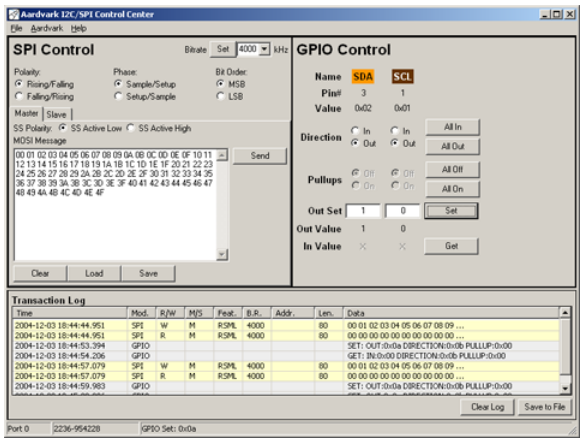

Figure 3: *SPI and GPIO using the 2 available I²C pins: SCL, SDA*

4. GPIO Only

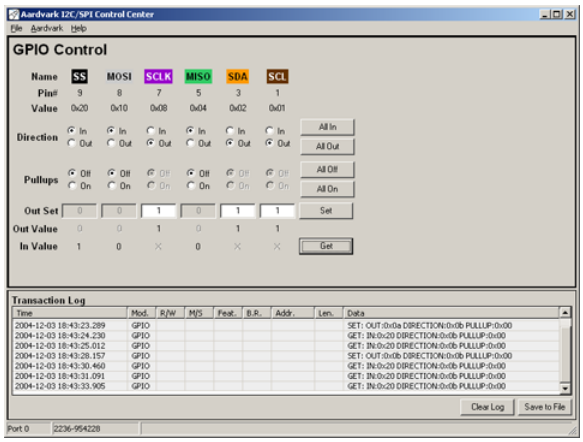

Figure 4: *GPIO using the 6 available I²C and SPI pins: SCL, SDA, MOSI, SCLK, MISO, SS*

5. Batch Scripting

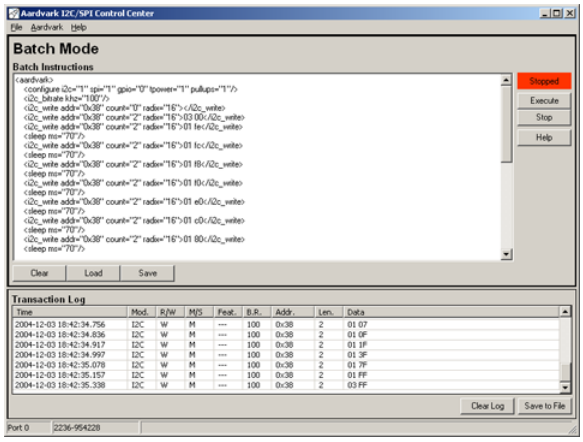

Figure 5: *Batch Scripting Module*

2 Getting Started

2.1 Requirements

Overview

The Control Center software is offered as a 32-bit binary and it is compatible with 64-bit operating systems. The following sections describe the requirements to run Control Center. Be sure the device driver has been installed before plugging in the Aardvark adapter. Refer to the Software section of the Aardvark user manual for additional information regarding the driver and compatibility.

Windows

The Control Center software is compatible with 32-bit and 64-bit versions of Windows XP (SP2 or later), Windows Vista, Windows 7, and Windows 8/8.1. The software will run on 64-bit systems as a 32-bit application. Windows 2000 and legacy 16-bit Windows 95/98/ME operating systems are not supported.

Linux

The Control Center software has been designed for Red Hat Enterprise Linux 4 and 5 with integrated USB support. Kernel 2.6 or newer is required. Other distributions, including Ubuntu, Fedora, CentOS, SuSE, Debian, and Arch have also been known to work. The software will run on 64-bit systems as a 32-bit application, provided that 32-bit system libraries are available.

Mac OS X

The Control Center software is compatible with Intel versions of Mac OS X 10.5 Leopard, 10.6 Snow Leopard, 10.7 Lion, and 10.8 Mountain Lion. Installation of the latest available update is recommended.

Aardvark I²C/SPI Host Adapter

Version 3.50 of the Control Center software requires an Aardvark adapter that runs version 3.40 or greater of the firmware.

Aardvark adapters with version 2.1 or greater of the hardware will be able to use the target power and configure I²C pullup functionality.

2.2 Installing Control Center

The Control Center software is a self-contained application. Installing the software is as easy as unarchiving the software package. To install the Control Center:

1. Download the latest version of the software from the Total Phase website.
2. Unzip the zip archive to your desired location.

2.3 Launching the Control Center

Windows

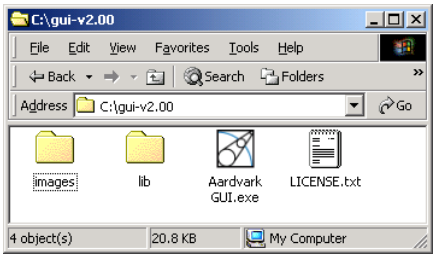

Figure 6: *Control Center Directory*
This is what the directory should look like after the software has been extracted.

1. Go to the folder where the software package was extracted.
2. Click on "**Aardvark GUI.exe**"

Linux

1. Go to the installation directory where the software package was unzipped.
2. Run >`./aardvark-gui`

Mac OS X

1. Go to the installation directory where the software package was unzipped.
2. Click on "**Aardvark GUI**"

2.4 Operating the Control Center

After launching the Control Center, you will see the screen in figure 7.

An Aardvark adapter must be configured for use before the Control Center software can be used to send and receive any messages. Configuring the Aardvark adapter binds the instance of the application with an available Aardvark unit until the adapter is disconnected or the application is terminated.

Configure the Aardvark Adapter

The configuration window (figure 8)is organized into two sections: the list of available Aardvark adapters and a list of operational modes.

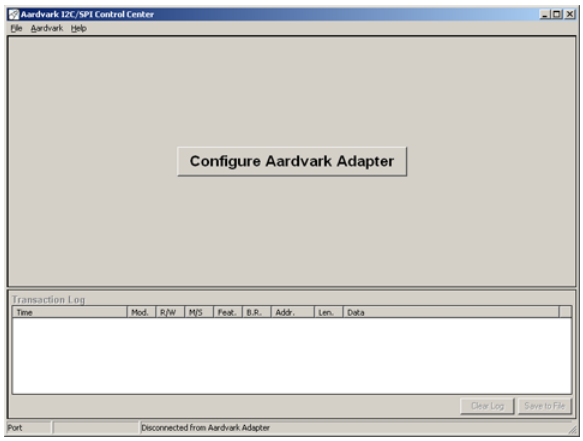

Figure 7: *Initial Screen*
After first launching the Control Center, the Aardvark Adapter must be configured.

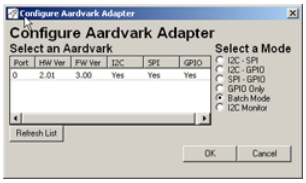

Figure 8: *Configure Aardvark Adapter Window*
The Configure window is organized in two major sections: the list of available Aardvark adapters and the list of operational modes.

List of Available Aardvark Adapters

In the configure window, there is a list of all the available Aardvark adapters that are connected to the computer. If no Aardvark adapters appear in the list, then there are no available units connected to your computer.

Refresh List

To see an updated list of Aardvark adapters attached to the computer, simply click on the "**Refresh List**" button to rescan the USB bus. Please note that Aardvark adapters that are in use by other applications are no longer available and consequently do not appear in the list of available units.

The list of Aardvark adapters provides the following information:

Port

The port that the Aardvark adapter occupies. The port number is a zero based number. For more information about USB port assignments, please consulting section *3.4: USB Port Assignment* of the Aardvark I²C/SPI Host Adapter user manual.

Hardware Version (HW Ver.) & Firmware Version (FW Ver.)

For more information about version numbers, please consult the Aardvark I²C/SPI Embedded Systems Interface user manual, *Sections 3.5 Aardvark Dynamically Linked Library – Versioning* and *3.6 API Integration into Custom Applications – Versioning*.

Serial Number

The serial number of the Aardvark adapter.

I2C, SPI, and GPIO

Supported modules. "**Yes**" indicates that a module is supported. "**No**" indicates that a module is not supported.

Operational Modes

On the right side of the window is a list of the six operational modes: "**I2C + SPI**", "**I2C + GPIO**", "**SPI + GPIO**", "**GPIO Only**", and "**Batch Mode**". Select the radio button next to the desired mode.

The mode of the Aardvark adapter can be changed after the unit has already been connected, see *Section 2.3 Reconfiguing the Aardvark Adapter*.

Once you have selected the Aardvark adapter and the desired mode, click on "**OK**" to continue.

The port and serial number of the Aardvark adapter will appear in the status bar at the bottom of the window to indicate which Aardvark adapter is bound to this instance of the application.

2.5 Reconfiguring the Aardvark Adapter

After the Control Center has been configured with a specific Aardvark adapter, it is possible to change the Aardvark adapter and/or the mode that it is operating in.

Change Mode

To change the mode but continue using the same Aardvark adapter go to the menu item:

Aardvark

Select the desired mode and the Control Center will be automatically reconfigured to reflect this selection.

2.6 Powering Downstream Devices

It is possible to power a downstream target, such as an I²C or SPI EEPROM with the Aardvark adapter's power (which is provided by the USB port). More information about powering downstream devices can be found in the Aardvark I²C/SPI Embedded Systems Interface user manual.

NC/+5V (Pin 4): I²C Power

NC/+5V (Pin 6): SPI Power

By default, these pins are left unconnected at the time of shipping. For Aardvark adapters with hardware versions 2.00 and greater, these pins can be enabled through the Control Center software. Simply go to the menu item: **Aardvark** | **Target Power**.

A checkmark indicates that power will be supplied to downstream devices on both pins.

2.7 Disconnecting the Aardvark Adapter

The Aardvark adapter can be disconnected from the current application. To do so, go to the menu item: **Aardvark** | **Disconnect**.

When disconnected, the application will return to the starting screen.

2.8 Exiting the Application

To exit the application, go to the menu item: **File** | **Exit Application**.

3 Application

3.1 General

The main application window is divided into two sections. The top section contains the modules for use with the Aardvark adapter. Depending on the selected mode, different modules will appear in the main display. The available modules are I²C , SPI and GPIO. Each one has different features and functionality which are explained in *Section 4. Modules*.

The bottom section of the application contains the Transaction Log. The log keeps track of all transactions that the Aardvark adapter sends or receives.

3.2 Transaction Log

Elements of the Transaction Log

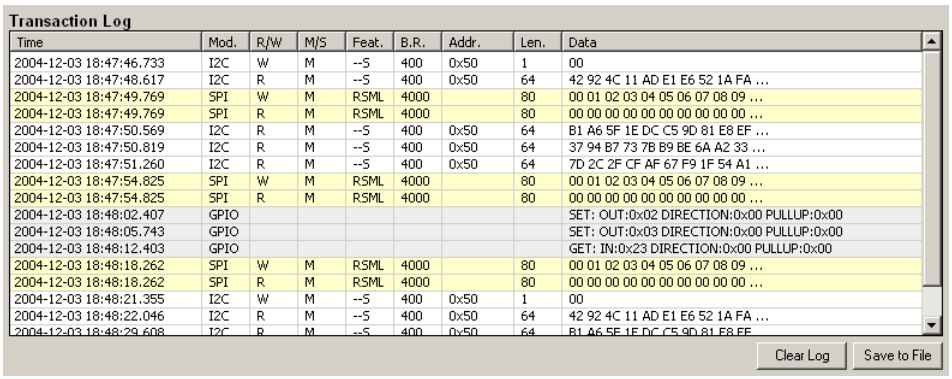

Figure 9: *The Transaction Log*
The Transaction Log records all transactions that are performed by the Aardvark adapter. This log can be saved as a file for future reference.

The Transaction log is a scrolling list of all the transactions that the Aardvark adapter sends or receives. The information is arranged in the following columns.

Time

Time of the transaction. This information is displayed in the format:

```
YYYY-MM-DD hh:mm:ss.xxx
```

Module (Mod.)

The module that logged the transaction. This can be either “**I2C**”, “**SPI**”, or “**GPIO**”. Log entries from for different modules have different background colors to make them easier to identify.

Table 1: *Transaction Background Colors*

I²C	White
SPI	Yellow
GPIO	Gray

Read/Write (R/W)

This column is only used by I²C and SPI

Whether the transaction was a read transaction ("**R**") or a write transaction ("**W**").

Master/Slave (M/S)

This column is only used by I²C and SPI

Indicates the mode that the attached Aardvark adapter is functioning in. If the Aardvark unit is operating as a master, then an "**M**" appears. In the Aardvark unit is operating as a slave, then an, "**S**" appears.

Features (Feat.)

This column is only used by I²C and SPI

This is a string that indicates the features that were active during the transaction.

I²C

I²C features are encoded in the string: "TCS"

```
T - 10-bit Addressing
C - Combined FMT
S - No Stop
```

If the feature is not being used, then a "**-**" appears in place of the character. More information about the specific features can be found in *Section 4.1 I²C* .

For example, in a transaction that uses 10-bit addressing and the no stop feature, the feature string would be "**T-S**".

SPI

SPI data exchange parameters are string encoded. The order of the parameters is:

Polarity ("R" or "F")
Rising ("**R**") or Falling ("**F**")

Phase ("S" or "s")
Sample/Setup ("**S**") or Setup/Sample ("**s**")

Bit Order (“M” or “L”)
Most Significant Bit First - MSB (“**M**”)
Least Significant Bit First - LSB (“**L**”)

SS Polarity (“L” or “H”)
SS Active Low (“**L**”)
SS Active High (“**H**”)

If a feature is not being used, then a “**-**” appears in the place of the character. More information about the SPI Bit Protocol can be found in *Section 4.2 SPI*

For example, in a transaction that uses Falling polarity, Setup/Sample Phase and Most Significant Bit First, the feature string would be: “**FsM-**”

Bitrate (B.R.)

This column is only used by I²C and SPI

Indicates the speed of the transaction in kbps. The value logged in this column many differ from the bitrate specified. The value reported here is the actual bit rate returned by the Aardvark adapter.

Address (Addr.)

This column is only used by I²C

The address that was the target of the transaction.

In the case of I²C master, the address is the slave address that the master targeted for communication.

In the case of I²C slave, the address will be the address of the Aardvark adapter unless the Aardvark slave is answering a general call in which case 0x80 will appear in the address column.

Length (Len.)

This column is only used by I²C and SPI

The number of bytes in the transaction. In the case where bytes are dropped from the transmission, “DB” will appear in this column to indicate that the message was truncated and bytes were dropped from the end of the message.

Data

I²C and SPI

The message in hexadecimal.

GPIO

The GPIO values in the log is the OR-ed values of the pins involved. The pins have the following values:

```
SCL  = 0x01
SDA  = 0x02
MISO = 0x04
SCK  = 0x08
MOSI = 0x10
SS   = 0x20
```

Set

The data is in the format:

`OUT:0xXX DIRECTION:0xXX PULLUPS:0xXX`

Get

The data is in the format:

`IN:0xXX DIRECTION:0xXX PULLUPS:0xXX`

OUT and IN

1 indicates that the pin has been set logic high and 0 indicated logic low. For example, if **SCL**, **MOSI** and **SS** where set to logic high, then the value of OUT would be: 0x31.

```
0x01 | 0x10 | 0x20 = 0x31
```

DIRECTION

1 indicates that the direction is out and 0 indicates that the direction is in. For example, if **SDA**, **MISO** and **SS** were set to output and the rest of the pins to input, the value of DIRECTION would be: 0x26.

```
0x02 | 0x04 | 0x20 = 0x26
```

PULLUPS

1 indicates that the pullup is on and 0 indicates the pullup is off. For example, if **MISO**, **SCK**, **MOSI** and **SS** had their pullups turned on, the value of PULLUP would be: 0x3c.

```
0x04 | 0x08 | 0x10 | 0x20 = 0x3c
```

More information about GPIO can be found in *Section 4.3 GPIO.*

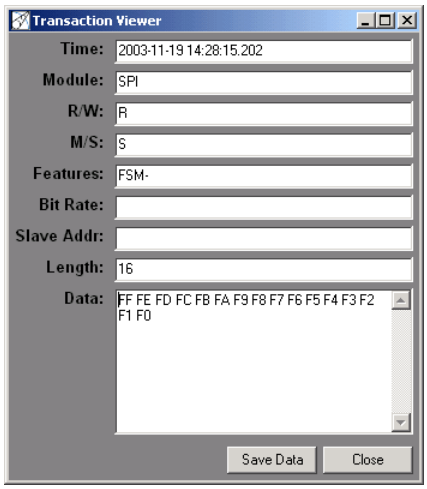

Figure 10: *The Transaction Viewer*

Transaction Viewer

The transaction viewer is a convenient way to view the full details of a transaction. To see an transaction in the Transaction Viewer, double-click on an entry in the transaction log.

Save Data

The transaction data can be saved in a binary file. Click on the "**Save Data**" button to bring up the save file dialog. This binary file can later be loaded as a message in I²C or SPI.

Clear Log

Deletes all entries from the Transaction Log.

Note that all transactions are cleared immediately when the button is pressed.

Save To File

The data in the log can be exported in a comma separated values (CSV) format by clicking on the "**Save To File**" button. You will be prompted for a filename to save the data.

The log file has a header with the following information:

```
Export Time: [time of export]
Port [port number]
Aardvark HW_Version: [hw version]
FW_Version: [fw version]
```

After the header, all transactions are appended, one per line. The column order in the exported file is the same as the order columns in the transaction log.

4 Modules

4.1 I²C

Inter-IC bus, or I²C , was developed by Philips in the 1980s. I²C is a low-bandwidth, short distance protocol for on board communications. All devices are connected through two wires: serial data (SDA) and serial clock (SCL).

The Control Center I²C module consists of 2 tabs, master and slave.

I²C Pull-ups

There is a 2.2K resistor on each I²C line (SCL, SDA). The lines are effectively pulled up to 3.3V, so that results in approximately 1.5 mA of pull-up current. For more information about the pull-up resistors, please consult the Aardvark I²C/SPI Host Adapter User Manual.

Aardvark adapters with a hardware versions 2.00 and greater have the ability to enable the pull-up resistors through the Control Center application. To toggle the pull-up resistors, go to the menu item:

Aardvark -> I2C Pull-ups

A checkmark indicates that the pull-up resistors have been enabled on the I²C lines.

I²C Master

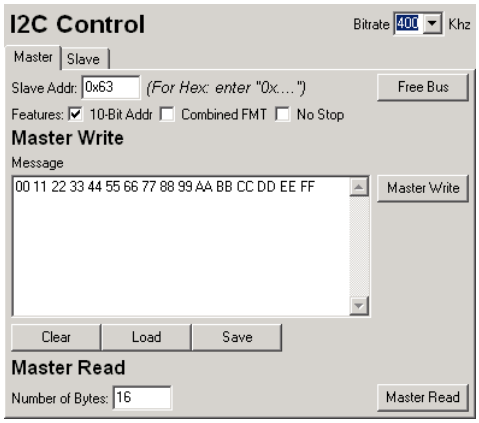

Figure 11: *I²C Master tab of the I2C Module*

As a master device, there are two actions, write or read. For these actions, there are a number of parameters that can or must be specified: bitrate, slave Address, and other I²C features.

Bitrate

The bitrate is the speed of communications between the master and the slave. The maximum master I²C bitrate is 800 kbps and a minimum of 1 kbps. The Aardvark I²C module supports

many intermediate bit rates between these values. More information about the bit rate can be found in the Aardvark I²C/SPI Host Adapter User Manual.

Changing the Bitrate

To change the bitrate, simply select a bitrate from the pull-down menu or alternatively, enter your own bitrate and press **<Enter>**.

Figure 12: *Bitrate Confirmation Dialog*

If the bitrate you entered is not available, the application will display a message indicating the close matching bitrate. Click on "**Yes**" to accept this alternative bit rate or click on "**No**" to continue using the existing bitrate.

Slave Address

The slave address is the address of the target I²C slave device. This address can be entered in either decimal or hexadecimal notation. If using hexadecimal notation, preface the number with "0x". For 7-bit and 10-bit addressing, the 7 and 10 least significant bits should be used to specify the address, respectively.

Features

The Aardvark adapter supports many of the additional I²C features.

10-Bit Slave Address (10-Bit Addr.)

When 10-bit slave address is selected, the slave address will be treated as a 10-bit address. The appropriate actions as described in the I²C specification will be performed to address the 10-bit slave on the bus.

Note: The Aardvark slave is always a 7-bit addressed device.

Combined Format (Combined FMT)

When Combined Format is selected, the"combined" format will be used for **Master Read** commands. This feature is only enabled when 10-bit addressing is active because it is only useful when used in conjunction with 10-bit slave Addressing

This flag indicates to the Aardvark adapter that the address is a 10-bit address but that it is not necessary to send the entire address using a master write before executing the read.

For specific information about the "combined" format, consult *Section 14.2 "Formats with 10-bit address"* in the Philips I²C Specification. A link to the specification can be found on the Total

Phase website.

No Stop (No Stop)

When No Stop is selected, the master device will explicitly not signal the stop command after the last byte in a transaction. The bus will be held and the subsequent master read or master write events will issue the repeated start on the bus.

Free Bus

The "**Free Bus**" button will explicitly issue the stop command on the I²C bus.

If the Aardvark I²C subsystem had executed a master transaction and is holding the bus due to a previous "No Stop" transaction, the Free Bus command will issue the stop command and free the bus.

I²C Master Write

Message to Send

Enter the message to be sent in hexadecimal in this field. Spaces will be automatically added for better legibility but these spaces will not be sent as part of the message. The maximum message size is 64k (65535) bytes when using 7-bit addressing and 64k - 1 (65534) bytes when using 10-bit addressing.

The message can be loaded from a binary file by clicking on the "**Load**" . Conversely, the message can also be saved to a binary file by clicking on the "**Save**" button.

Once a message has been set, click on the "**Master Write**" button to initiate the action. The results of the action will appear in the transaction log.

I²C Master Read

Number of Bytes

This value is the maximum number of bytes the master will accept in a single transaction. The master may receive fewer bytes than are specified in this field, but not more. In the case that a slave does not have the requested number of bytes available, the remainder of the bytes will simply default to `0xff` due to the pullup resistors on the bus.

I²C Slave

Slave Enable

An I²C slave can send messages to and receive messages from a master device after the master has initiated a transaction. An Aardvark adapter will not respond as an I²C slave device unless it has been enabled as a slave device.

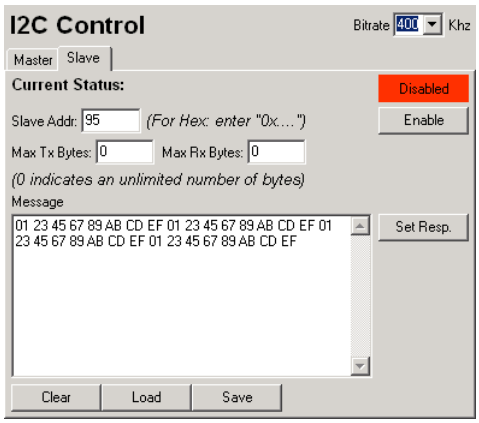

Figure 13: *I²C Slave tab of the I2C Module*

When enabling an Aardvark adapter as a slave device, three parameters must be provided: the "**Slave Address**", the maximum number of bytes to send ("**Max Tx Bytes**") and the maximum number of bytes to receive ("**Max Rx Bytes**").

Slave Address

This is the I²C address that the Aardvark adapter will use as an I²C slave device. The Aardvark adapter always uses a 7-bit slave address. The address is specified in the 7 least significant bits. The most significant bit is ignored.

Max Tx Bytes & Max Rx Bytes

The Max Tx Bytes and Max Rx Bytes indicates the maximum number of bytes the Aardvark device will send and receive respectively. The Aardvark adapter will not exceed the maximum number of bytes that have been specified.

An exception to this rule is "0" which indicates that the number of bytes is unlimited.

Slave Response

A slave response message can be set in the Aardvark adapter as a response to a write request. The message entry field operates in the same manner as the I²C master message to send field. The maximum message size is 64 bytes due to buffer limits.

The message can be loaded from a binary file by clicking on the "**Load**" button. Conversely, the message can also be saved to a binary file by clicking on the "**Save**" button.

If more bytes are requested in a transaction than have been specified in the slave response, the response string will be wrapped as many times as necessary to complete the transaction. For example if the slave response has been set to:

```
00 01 02 03 04
```

and 12 bytes have been requested, the response that is sent to the master will be:

```
00 01 02 03 04 00 01 02 03 04 00 01
```

To set the response in the slave, click on the "**Set Resp**" button. It is advisable to set the slave response before enabling the slave. If a response is not set before the slave is enabled, it is possible that a slave response be requested before the slave device has one to return.

Note: Slave Message Can Be Overwritten

All I²C messages share memory in the Aardvark adapter. Therefore it is possible that the slave response may be overwritten in the Aardvark unit.

For example, an I²C slave response is set and then Aardvark adapter is enabled as an I²C slave, but then without disabling the slave, an I²C master transaction is executed. The Aardvark adapter will implicitly deactivate the I²C slave and because of the shared memory in the Aardvark adapter the I²C master operation will almost always overwrite the I²C slave response.

The safest course of action is to set the slave response each time before enabling or re-enabling the I²C slave.

Enabling the Slave

To enable the Aardvark adapter as an I²C slave, simply click on the "**Enable**" button.

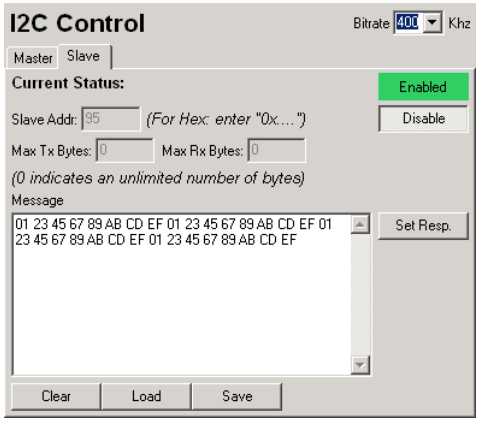

Figure 14: *Enabled I²C Slave tab of the I2C Module*

Once the slave is enabled, the status indicator at the top of the panel will change from "**Disabled**" in red to "**Enabled**" in green.

As request arrive for the slave, the transaction log will be updated with the read and write actions that the slave performed.

Disabling the Slave

To disable the Aardvark adapter as an I²C slave device, simply click on the "**Disable**" button. Once disabled the status indicator at the top of the panel will change from "**Enabled**" in Green to "**Disabled**" in Red.

Implicit Slave Disabling

Executing an I²C master write or I²C master read will implicitly disable the Aardvark device as an I²C slave device. After a master write or read event, it is recommended that the I²C slave response be resent to the Aardvark adapter and then it can be re-enabled as an I²C slave device.

Switching the I²C pins to GPIO will also implicitly disable the I²C slave.

4.2 SPI

SPI is a serial communication bus developed by Motorola. It is a full-duplex protocol which functions on a master-slave paradigm that is ideally suited to data streaming applications.

The SPI tab consists of two tabs: master and slave.

SPI Data Exchange Parameters

The SPI master and slave need to agree about the data frame for the transaction. The data frame is described by three parameters: clock polarity, clock phase and bit order.

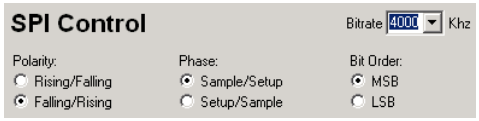

Figure 15: *SPI Parameters in the SPI Module*

These parameters must be the same for both the master and slave modes. More information about these parameters can be found in the Aardvark I²C/SPI Host Adapter User Manual.

Bitrate

The bitrate is the speed of communications between the master and the slave. The Aardvark SPI master can operate at bitrates between 125 kHz and 8 MHz, including, but not limited to the following frequencies: 125 kHz, 250 kHz, 500 kHz, 1 MHz, 2 MHz, 4 MHz, and 8 MHz. The quoted bitrates are only achievable within each individual byte and does not extend across bytes. More information about the bit rate can be found in the Aardvark I²C/SPI Host Adapter User Manual.

Changing the Bitrate

To change the bitrate, simply select a bitrate from the pull-down menu or alternatively, enter your own bitrate and press **<Enter>**.

If the bitrate you entered is not available, the application will display a message indicating the close matching bitrate. Click on "**Yes**" to accept this alternative bit rate or click on "**No**" to continue using the existing bitrate.

Figure 16: *Bitrate Confirmation Dialog*

SPI Master

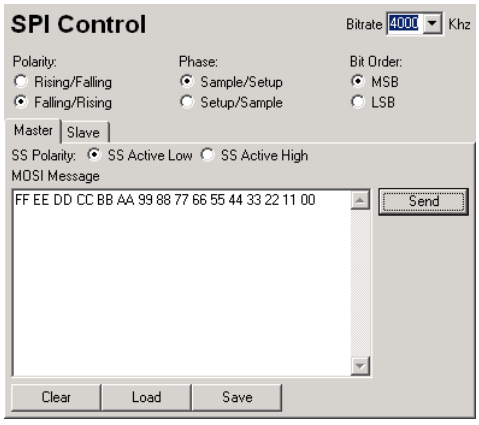

Figure 17: *SPI Master tab in the SPI Module*

The SPI master has an additional parameter that can be set, the polarity of the SS line.

SS Polarity

The SS Polarity indicates whether the Aardvark device will pull the SS pin high or low to activate the SPI slave device.

Note: When configured as an SPI slave, the Aardvark will always be setup with SS as active low.

MOSI Message

MOSI (Master Out, Slave In) message is entered here in hexadecimal format. Spaces are automatically inserted for legibility. The maximum message size is 4 KiB due to operating system buffer limits.

The message can be loaded from a binary file by clicking on the "**Load**" button. Conversely, the message can also be saved to a binary file by clicking on the "**Save**" button.

SPI is a full duplex protocol. When the MOSI message is sent a MISO message is received. The transaction log will log MOSI and MISO as two separate transactions that occur at the same time. The length of the two messages will be the same due to the duplex nature of the protocol.

SPI Slave

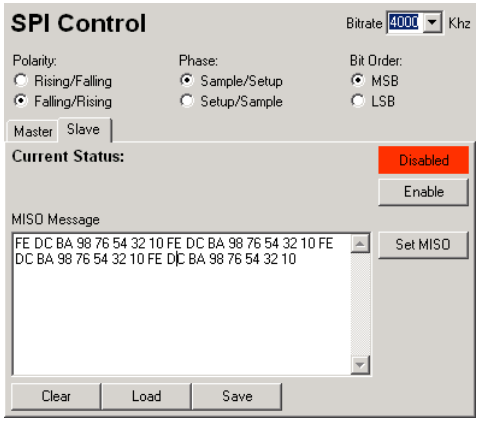

Figure 18: *SPI Slave tab in the SPI Module*

MISO Message

The MISO (Master In, Slave Out) message is the message that the Aardvark adapter will return as its response to a SPI transaction. Like the MOSI message, this message is entered in hexadecimal format. Spaces are automatically inserted for legibility, but are not sent in the transaction. The maximum message size is 64 bytes due to buffer limits. If more bytes are requested in a transaction than have been specified in the response, the response will be wrapped as many times as necessary to complete the transaction.

It is advisable to set the MISO message before enabling the slave. If a MISO message is not set before the slave is enabled, it is possible that the message may be requested before the slave device has one to return.

The message can be loaded from a binary file by clicking on the "**Load**" button. Conversely, the message can also be saved to a binary file by clicking on the "**Save**" button.

Note: MISO Message Can Be Overwritten

All SPI messages share memory in the Aardvark adapter. Therefore it is possible that the MISO message may be overwritten in the Aardvark unit.

For example, a MISO message is set and then Aardvark adapter is enabled as an SPI slave, but then without disabling the slave, an SPI master transaction is executed. The Aardvark adapter will implicitly deactivate the SPI slave and because of the shared memory in the Aardvark adapter the SPI master operation will almost always overwrite the MISO message.

The safest course of action is to set the MISO message each time before enabling or re-enabling the SPI slave.

Slave Enable

An Aardvark adapter will not respond as an SPI slave device until it has been enabled. It is advisable that the MISO message be set in the slave device before it is enabled to ensure valid data to all requests.

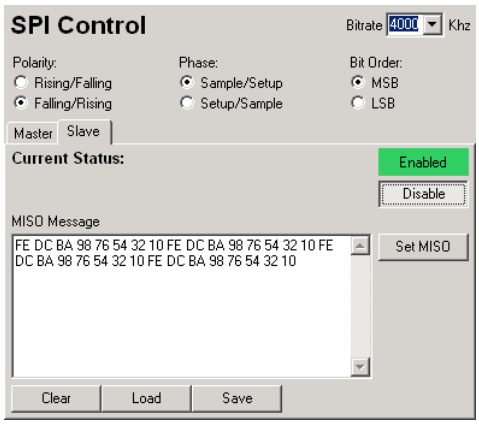

Figure 19: *Enabled SPI Slave tab in the SPI Module*

Once the slave is enabled, the status indicator at the top of the panel will change from "**Disabled**" in red to "**Enabled**" in green. As request arrive for the slave, the transaction log will be updated with the read and write actions that the slave performed.

When the MOSI message is received a MISO message is sent to the master. The transaction log will log MOSI and MISO as two separate transactions that occur at the same time.

Implicit Slave Disabling

Executing a SPI master write will implicitly disable the Aardvark device as an SPI slave device. After a master write event, it is recommended that the MISO message be resent to the Aardvark adapter and then it can be re-enabled as an SPI slave device.

Switching the SPI pins to GPIO will also implicitly disable the SPI slave.

4.3 General Purpose IO

General Purpose IO, GPIO, allows the users to use the six pins that are normally used for I²C and SPI and use them to send and receive signals. These six pins are SCL, SDA, MOSI, SCLK, MISO, and SS. GPIO functionality can be combined with either I²C or SPI or can be used by itself.

GPIO Configurations

When GPIO is combined with either I²C or SPI, only the pins of the unused module are available for GPIO. Therefore when using "**I2C + GPIO**" only the SPI pins are available for GPIO and when using "**SPI + GPIO**", only the I²C pins are available for GPIO.

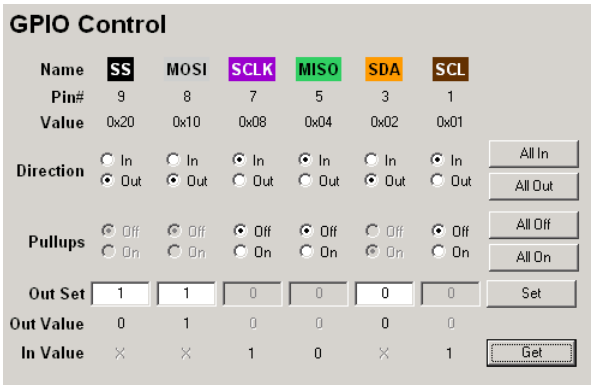

Figure 20: *GPIO Module*

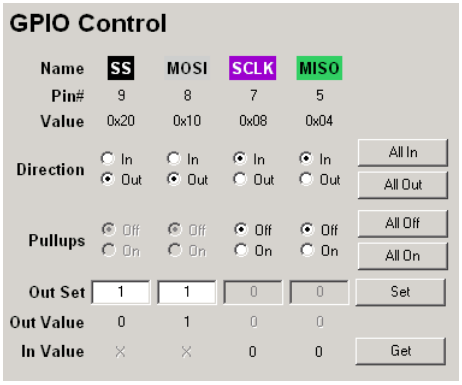

Figure 21: *GPIO module when using "**I2C + GPIO**"*

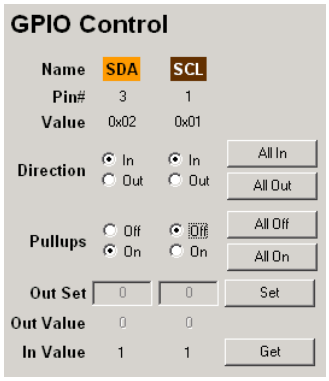

Figure 22: *GPIO module when using "**SPI + GPIO**"*

GPIO Parameters

When GPIO module is selected, only the available pins are displayed in the window. Each pin is labeled and has parameters that can be set by the user.

Name

The name of each pin is color coordinated to match the color of the lead in of the 10-pin split cable adapter. The 10-pin split cable adapter is not included with the Aardvark adapter, but is

available for purchase separately on the Total Phase website.

Pin #

The position of the pin in the 10-pin socket connector. Information about the pin arrangement can be found the Aardvark I²C/SPI Host Adapter User Manual and on the Total Phase website.

Value

Each pin has a different value which can be OR-ed together to produce a single number that represents the state of all the pins.

The pins have the following values:

```
SCL  = 0x01
SDA  = 0x02
MISO = 0x04
SCK  = 0x08
MOSI = 0x10
SS   = 0x20
```

For example, if SCL, MOSI and SS where set to 1, then the value of all pins would be:

```
0x01 | 0x10 | 0x20 = 0x31
```

Direction (Dir.)

The direction of the pin, Input or Output.

If a pin is configured as an input pin, then the pullup selector and In Value row are enabled and the Out Set and Out Value rows are disabled.

If a pin is configured as an output pin, then the pullup selector and the In Value are disabled and the Out Set and Out Value rows are disabled.

All In and All Out

The "**All In**" and "**All Out**" buttons are convenience buttons to set all pins to input or output respectively.

Pull Ups (P.U.)

Indicates whether the pullup is active or inactive on a pin. The pullup selector is only enabled when the pin is set to the in direction because the pullup are only turned on on pins that have been configured as input. If a line is configured as output, the pullup mask is cached and the pullup configuration for that line will only take effect if the line is later configured as an input.

All On and All Off

The "**All On**" and "**All Off**" buttons are convenience buttons to turn on the pullups on or off respectively on all pins.

Note: Only pins that have been configured to be input will be affected by these buttons.

Note: I²C Pin Pullups

It is not possible to disable the pullups for SCL and SDA input on Hardware Version 1.02.

Out Set & Out Value

The "**Out Set**" boxes are a staging area for setting the levels of the output pins. Only "**0**" and "**1**" are accepted in these text boxes.

The values in the "**Out Set**" boxes are only applied once the "**Set**" button is clicked. The "**Out Value**" indicates the last known values of the output pins. After clicking the Set button the "**Out Set**" and the "**Out Value**" for all pins configured as output should match. The parameters of the transaction are added to the transaction log.

If a pin is switched from output to input, the values in "**Out Set**" and "**Out Value**" are disabled but are conserved. When a pin is switched back to output, these values will be restored.

In Value

The "**In Value**" is the last known values of the input pins. Initially the values for the In Value will be "**X**" indicating that the value is not known on the pin.

The "**In Value**" of the input pins are updated when the "**Get**" button is clicked. The parameters of the transaction are added to the transaction log.

The "**In Value**" for an output pin will always display an "**X**" because the input value of this pin is not known.

4.4 Batch Mode

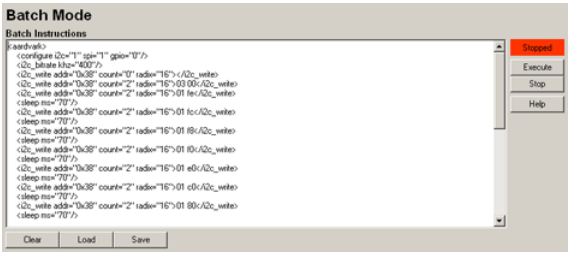

Figure 23: *Batch Scripting Module*

In Batch mode, the user can specify an arbitrary set of instructions for the Aardvark adapter to execute in sequence. This scripting language is based on XML.

Batch Instructions

A set of batch instructions for an Aardvark adapter is scripted in an XML based language. A set of instructions must be contained within a set of <aardvark> tags. Each command is specified by an XML tag. These tags are described in section *5. Batch Instruction Commands*. Commands are executed in the order that they appear in the XML block.

Help for the Batch Commands is available in the Control Center software which explains all the available commands. Just click on the "**Help**" button.

Figure 24: *Batch Commands Help System*

Editing Batch Instructions

The Batch XML Instructions are entered in this text field. Batch instructions can be saved and loaded as XML files via the "**Save**" and "**Load**" buttons respectively. The "**Clear**" button will clear all contents out of the text field. The following dialog box will appear to confirm that the user wants to clear all data out of the text field.

Figure 25: *When clearing the batch command console, a dialog box will appear to confirm the user's action.*

Executing Batch Instruction

To execute a set of batch instructions, simply click on the "**Execute**" button. When a script is executing, the status indicator will turn green. The results from the commands will appear in the transaction log. While a script is running, it is not possible to edit the batch instructions.

To stop the execution of a script, simply click on the "**Stop**" button.

Once the script has completed, the status indicator will change back to red and the batch instructions will once again be editable.

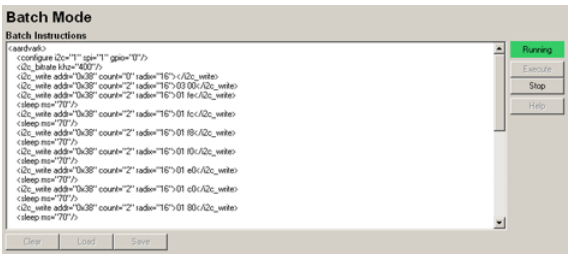

Figure 26: *Executing a batch command script.*

Batch Instruction Error

When the "**Execute**" button is clicked, the instruction set is parsed and validated. In the event of an error, a dialog box will open indicating the type of error and the command in which the error appears.

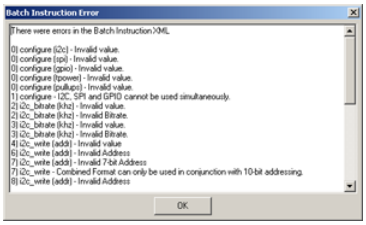

Figure 27: *Batch command error dialog.*

The format of the error message is:

```
n) command [attribute] - error message
```

where:

`n`
: zero-based index of the command

`command`
: command type

`attribute`
: attribute name where the error occurred. If an error is not associated with an attribute, this field will be omitted.

`error message`
: error message.

Click OK to close the dialog.

5 Batch Instruction Commands

5.1 Notes on Batch Instructions

Unless specified otherwise, all arguments to a batch instruction are “1” to enable/request the specified parameter, or “0” to disable/not request the specified parameter.

5.2 General Commands

Configure

```
<configure i2c="i2c"
           spi="spi"
           gpio="gpio"
           tpower="tpower"
           pullups="pullups"/>
```

Activate/Deactivate individual sub-systems and features

Parameters

`i2c`
: Enable I2C.

`spi`
: Enable SPI.

`gpio`
: Enable GPIO.

`tpower`
: Enable Target Power on pins 4 and 6.

`pullups`
: Enable pullup resistors on the I2C lines: SCL and SDA.

Details

It is not possible to activate I2C, SPI and GPIO all at the same time. The possible combinations are:

Please see the Aardvark I2C/SPI Embedded Systems Interface Datasheet for more information about supplying target power and I2C pullup resistors.

sleep

```
<sleep ms="ms"/>
```

Set the Aardvark adapter to sleep for a number of milliseconds

Table 2: *Allowed configuration combinations*

i2c	spi	gpio	Configuration
“1”	“1”	“0”	I2C and SPI enabled
“1”	“0”	“1”	I2C enabled and SPI pins used as GPIO
“0”	“1”	“1”	SPI enabled and I2C pins used as GPIO
“0”	“0”	“1”	I2C and SPI pins used as GPIO

Parameters

`ms`
Requested number of milliseconds to sleep.

5.3 I2C Commands

i2c_bitrate

```
<i2c_bitrate khz="khz"/>
```

Set the I2C bitrate in kilohertz.

Parameters

`khz`
The requested bitrate in kHz.

Details

The default power-on bitrate is 100khz

Only certain discrete bitrates are supported by the Aardvark I2C master interface. As such, this actual bitrate set will be less than or equal to the requested bitrate.

The actual bitrate is returned in the Transaction Log. Please see the Aardvark I2C/SPI Embedded Systems Interface Datasheet for more information.

i2c_write

```
<i2c_write addr="addr"
           count="count"
           nostop="nostop"
           ten_bit_addr="ten_bit_addr"
           combined_fmt="combined_fmt"
           radix="radix">
```

```
    message
</i2c_write>
```

Write a stream of bytes to the I2C slave device.

Parameters

`addr`
: The slave from which to read. The slave address can be specified in decimal or hexadecimal notation.

`count`
: The number of bytes to write (maximum 65535).

`nostop`
: Request that no stop condition is issued on the I2C bus after the transaction completes.

`ten_bit_addr`
: Request that the provided address is treated as a 10-bit address.

`combined_fmt`
: Request that the Philips combined format is followed during a I2C read operation. This only has an effect when used in conjunction with 10-bit addressing.

`radix`
: The base of the number system of the message, with the value being 10 for decimal, or 16 for hexadecimal.

`message`
: The message to transmit as a space separated list of numbers.

Details

For ordinary 7-bit addressing, the lower 7 bits of the addr should correspond to the slave address. The topmost bits are ignored. The Aardvark I2C subsystem will assemble the address along with the R/W bit after grabbing the bus. For 10- bit addressing, the lower 10 bits of addr should correspond to the slave address. The Aardvark adapter will then assemble the address into the proper format as described in the Philips specification. There is a limitation that a maximum of only 65534 bytes can be written in a single transaction if the 10-bit addressing mode is used.

The slave_addr 0x00 has been reserved in the I2C protocol specification for general call addressing. I2C slaves that are enabled to respond to a general call will acknowledge this address. The general call is not treated specially in the Aardvark I2C master. The user of this API can manually assemble the first data byte if the hardware address programming feature with general call is required.

Please see the Aardvark I2C/SPI Embedded Systems Interface Datasheet for more information.

i2c_read

```
<i2c_read addr="addr"
          count="count"
          nostop="nostop"
          ten_bit_addr="ten_bit_addr"
          combined_fmt="combined_fmt"/>
```

Read a stream of bytes from the I2C slave device.

Parameters

`addr`
: The slave from which to read. The slave address can be specified in decimal or hexadecimal notation.

`count`
: The number of bytes to read (maximum 65535).

`nostop`
: Request that no stop condition is issued on the I2C bus after the transaction completes.

`ten_bit_addr`
: Request that the provided address is treated as a 10-bit address.

`combined_fmt`
: Request that the Philips combined format is followed during a I2C read operation. This only has an effect when used in conjunction with 10-bit addressing.

Details

For ordinary 7-bit addressing, the lower 7 bits of the addr should correspond to the slave address. The topmost bits are ignored. The Aardvark I2C subsystem will assemble the address along with the R/W bit after grabbing the bus. For 10- bit addressing, the lower 10 bits of addr should correspond to the slave address. The Aardvark adapter will then assemble the address into the proper format as described in the Philips specification, namely by first issuing an write transaction on the bus to specify the 10-bit slave and then a read transaction to read the requested number of bytes. The initial write transaction can be skipped if the Combined Format feature is requested in conjunction with the 10-bit addressing functionality.

Please see the Aardvark I2C/SPI Embedded Systems Interface Datasheet for more information.

i2c_free_bus

```
<i2c_free_bus/>
```

Free the Aardvark I2C subsystem from a held bus condition (e.g., no stop).

Parameters

None

Details

If the Aardvark I2C subsystem had executed a master transaction and is holding the bus due to a previous nostop flag, this function will issue the stop command and free the bus.

Please see the Aardvark I2C/SPI Embedded Systems Interface Datasheet for more information.

5.4 SPI Commands

spi_config

```
<spi_config polarity="polarity"
            phase="phase"
            bitorder="bitorder"
            ss="ss"/>
```

Configure the SPI master interface.

Parameters

`polarity`
: Set as either “rising/falling” or “falling/rising”.

`phase`
: Set as either “sample/setup” or “setup/sample”.

`bitorder`
: Set as either “msb” or “lsb”.

`ss`
: Set as either “active_low” or “active_high”.

Details

These configuration parameters specify how to clock the bits that are sent and received on the Aardvark SPI interface.

The polarity option specifies which transition constitutes the leading edge and which transition is the falling edge. For example, “rising/falling” would configure the SPI to idle the SCLK clock line low. The clock would then transition low-to-high on the leading edge and high-to-low on the trailing edge.

The phase option determines whether to sample or setup on the leading edge. For example, “sample/setup” would configure the SPI to sample on the leading edge and setup on the trailing edge. The bitorder option is used to indicate whether LSB or MSB is shifted first.

The ss option is used change the output polarity on the SS line. For example, “active_low” will pull the SS line low to active the slave device.

Please see the Aardvark I2C/SPI Embedded Systems Interface Datasheet for more information.

spi_bitrate

```
<spi_bitrate khz="khz"/>
```

Set the SPI bitrate in kilohertz.

Parameters

`khz`
: The requested bitrate in kHz.

Details

The power-on default bitrate is 1000 kHz. Only certain discrete bitrates are supported by the Aardvark adapter. As such, this actual bitrate set will be less than or equal to the requested bitrate unless the requested value is less than 125 kHz, in which case the Aardvark adapter will default to 125 kHz.

The actual bitrate is returned in the Transaction Log.

Please see the Aardvark I2C/SPI Embedded Systems Interface Datasheet for more information.

spi_write

```
<spi_write count="count"
           radix="radix">
    message
</spi_write>
```

Write a stream of bytes to the downstream SPI slave device and read back the full-duplex response.

Parameters

`count`
: The number of bytes to write (maximum 65535).

`radix`
: The base of the number system of the message, with the value being 10 for decimal, or 16 for hexadecimal.

`message`
: The message to transmit as a space separated list of numbers.

Details

If count is 0, no bytes will be written to the slave. However, the slave select line will be dropped for 5-10 microseconds. This can be useful in sending a signal to a downstream SPI slave without actually sending any bytes. For example, if an SPI slave has tied the slave select to an interrupt line and it sees the line is toggled without any bytes sent, it can interpret the action as a command to prepare its firmware for an subsequent reception of bytes.

Please see the Aardvark I2C/SPI Embedded Systems Interface Datasheet for more information.

5.5 GPIO Commands

The following table maps the named lines on the Aardvark I2C/SPI output cable to bit positions in the direction and pullups masks. All GPIO API functions will index these lines through a single 8-bit masked value. Thus, each bit position in the mask can be referred back its corresponding line through the mapping described below.

Table 3: *GPIO Bit Values*

Pin #	Bit Value	Description
Pin 1	0x01	I2C SCL line
Pin 3	0x02	I2C SDA line
Pin 5	0x04	SPI MISO line
Pin 7	0x08	SPI SCK line
Pin 8	0x10	SPI MOSI line
Pin 9	0x20	SPI SS line

gpio_config

```
<gpio_config direction="direction"
             pullups="pullups"/>
```

Configure the GPIO interface.

Parameters

`direction`
: A single byte value where each bit corresponds to the physical line as defined in Table 3. If a line's bit is 0, the line is configured as an input. Otherwise it will be an output.

`pullups`
: A single byte value where each bit corresponds to the physical line as defined in Table 3. If a line's bit is 1, the line's pullup is active whenever the line is configured as an input. Otherwise the pullup will be deactivated.

Details

Please see the Aardvark I2C/SPI Embedded Systems Interface Datasheet for more information.

gpio_get

```
<gpio_get/>
```

Get the value of current GPIO inputs.

Parameters

None

Details

A line's bit position in the mask will be 0 if it is configured as an output or if it corresponds to a subsystem that is still active.

Please see the Aardvark I2C/SPI Embedded Systems Interface Datasheet for more information.

gpio_set

```
<gpio_set value="value"/>
```

Set the value of current GPIO outputs.

Parameters

`value`
: A bitmask as defined in Table 3 specifying which outputs should be set to logic high and which should be set to logic low.

Details

If a line is configured as an input or not activated for GPIO, the output value will be cached. The next time the line is an output and activated for GPIO, the output value previously set will automatically take effect.

Please see the Aardvark I2C/SPI Embedded Systems Interface Datasheet for more information.

6 Notes

6.1 Multiple Units

It is possible to operate multiple Aardvark adapters simultaneously. Each window of the application is bound to a single Aardvark adapter. Additional Aardvark units can be accessed by opening additional windows.

To open a new Window simply go to the menu item: **File** | **New Window**.

A new window will open which will need to be configured to the additional Aardvark device.

6.2 Aardvark I²C/SPI Host Adapter Technical Specifications

Detailed information about the Technical Specifications of the Aardvark I²C/SPI Host Adapter can be found on the Total Phase website: http://www.totalphase.com/

7 Legal / Contact

7.1 Disclaimer

All of the software and documentation provided in this user manual, is copyright Total Phase, Inc. ("Total Phase"). License is granted to the user to freely use and distribute the software and documentation in complete and unaltered form, provided that the purpose is to use or evaluate Total Phase products. Distribution rights do not include public posting or mirroring on Internet websites. Only a link to the Total Phase download area can be provided on such public websites.

Total Phase shall in no event be liable to any party for direct, indirect, special, general, incidental, or consequential damages arising from the use of its site, the software or documentation downloaded from its site, or any derivative works thereof, even if Total Phase or distributors have been advised of the possibility of such damage. The software, its documentation, and any derivative works is provided on an "as-is" basis, and thus comes with absolutely no warranty, either express or implied. This disclaimer includes, but is not limited to, implied warranties of merchantability, fitness for any particular purpose, and non-infringement. Total Phase and distributors have no obligation to provide maintenance, support, or updates.

Information in this document is subject to change without notice and should not be construed as a commitment by Total Phase. While the information contained herein is believed to be accurate, Total Phase assumes no responsibility for any errors and/or omissions that may appear in this document.

7.2 Life Support Equipment Policy

Total Phase products are not authorized for use in life support devices or systems. Life support devices or systems include, but are not limited to, surgical implants, medical systems, and other safety-critical systems in which failure of a Total Phase product could cause personal injury or loss of life. Should a Total Phase product be used in such an unauthorized manner, Buyer agrees to indemnify and hold harmless Total Phase, its officers, employees, affiliates, and distributors from any and all claims arising from such use, even if such claim alleges that Total Phase was negligent in the design or manufacture of its product.

7.3 Contact Information

Total Phase can be found on the Internet at http://www.totalphase.com/. If you have support-related questions, please email the product engineers at support@totalphase.com. For sales inquiries, please contact sales@totalphase.com.