Using the Aardvark™ I2C/SPI Host Adapter with the ZL3026x and ZL4025x

1. Introduction

This application note explains how to use the Aardvark I2C/SPI Host Adapter to access devices in the ZL3026x and ZL4025x families using the Python language. It is often useful to be able to read and write device registers from a scriptable environment running on a PC, rather than using an on-board processor or micro controller. For example, a user may wish to program an initial configuration in the synthesizer or buffer on a prototype board for which the local firmware is not yet ready. Similarly, the user may want to burn a custom configuration into EEPROM on an in-system device to test it before committing to a final configuration that will be factory-programmed.

The goal of this application note is to guide the user in creating the software and hardware environment needed to enable the Aardvark adapter to access the device via the SPI interface.

The Aardvark™ Adapter

The Aardvark adapter is a module available from Total Phase that bridges a device's I2C/SPI interface to a USB port on a PC. It translates instructions generated on the PC into SPI or I2C instructions driven to the device. The adapter can be ordered from the Total Phase website:

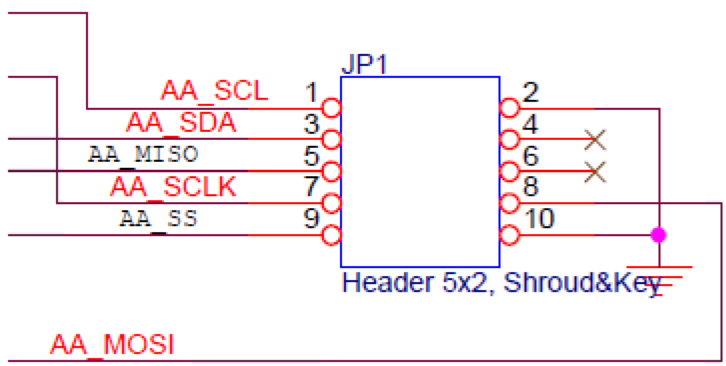
https://www.totalphase.com/products/aardvark-i2cspi/

Full documentation is also available at the website. The adapter is pictured below.

This application note explains how to use the Aardvark to generate SPI commands to the device. I2C communication is also possible, but different scripts are required. The Aardvark ribbon cable ends in a 5x2 header. Six pins – four signals and two gnds – are used for SPI communication.

Pin	Function	Needed for SPI communication
1	SCL	No
2	GND	Yes
3	SDA	No
4	NC	No
5	MISO	Yes
6	NC	No
7	SCLK	Yes
8	MOSI	Yes
9	SS	Yes
10	GND	Yes

The following schematic segment shows how connector pins map to the physical connector:



2. System Requirements

The target board – either a customer board or a Microsemi evaluation board – must provide access to the SPI interface of the device via a header that the Aardvark adapter can connect to. The common way to do this is to provide a 5x2 header on the board, but a smaller connector could be used on the board, with an adapter cable to connect between the header and the Aardvark 5x2 pin connector.

To avoid potential contention, the board needs to provide a mechanism to isolate SPI signals on the 5x2 header from any signals that might be driven by an on-board microprocessor or micro controller. Jumpers or 0 ohm resistors can be used to select between the SPI masters.

A Personal Computer with a 64-bit Windows operating system (Windows 7 or later) is also required. A 32-bit operating system is also possible, but this app note explains the steps necessary for obtaining software component for a 64-bit system.

Evaluation board considerations

If a ZLE30267 evaluation board is being used as the target board, the dip switches on SW2 should be set as follows:

- sw1: set to 1
- sw2: set to 1
- sw3: set to 0
- sw4: set to 0

These settings will connect the Aardvark SPI control signals to the DUT. Other switch settings should be as indicated in the Evaluation Board Quick Start Guide which is available from the device GUI Documentation menu.

Note that the USB connector on the board should be left un-connected.

A typical example of the setup is shown below:

3. Software Installation

Three software components need to be downloaded and installed:

- USB Drivers for the Aardvark adapter
- Software APIs for the Aardvark adapter
- Python 2.7

The software API supports additional programming environments including C, C-sharp, and Labview. However this application note assumes that user scripts will be written in Python.

Note that the following instructions make reference to the Total Phase website, links, and filenames as they existed at the time of this writing. Total Phase may make changes to the website without informing Microsemi. Note also that Total Phase may ask the user to register before allowing file downloads.

Create a working directory for the Aardvark files (for example in My Documents).

First, download the USB drivers to your Aardvark directory. These are available on a link on the Total Phase webpage cited in Section 2. The file name is totalphaseusb-v2.15.zip. Ensure that the Aardvark adapter is not plugged in. Unzip the files and run the executable to install the drivers.

Second, download version v5.15 of Windows 64-bit APIs for Aardvark to the Aardvark directory. It is available at: https://www.totalphase.com/products/aardvark-software-api/

The filename is aardvark-api-windows-x86_64-v5.15. Extract all files to a directory

Third, download the 64-bit version of Python 2.7.12. It is available at:

https://www.python.org/downloads/release/python-2712/. Choose Windows x86-64 MSI installer.

Install Python by double-clicking the MSI file. This will add a Python27 to your C: directory. Later versions of the 2.7 series will likely work, but have not been tested. Please note that the Aardvark APIs do not work with Python 3.

Next, create a working directory for your Python 2.7 code, for example:

My Documents/python27_64

Now copy the files “aardvark.dll” and “aardvark_py.py” from the Aardvark Windows API directory to the python27_64 directory. Together with the USB drivers, these files form the interface from the Python environment to the Aardvark adapter.

Finally, copy the files that were included in the zip file with this application note into your Python code working directory (for example My Documents/python27_64). The files included are:

- microsemi_aardvark_utils_dd.py
- microsemi_dump_regs.py
- microsemi_load_mfg.py
- microsemi_program_eeprom.py
- microsemi_read_write_interactive.py
- ZL30267_DUT_Config.mfg
- ZL30267_EEPROM_Image.txt

4. Examples

Four examples are provided:

- microsemi_dump_regs.py
- microsemi_load_mfg.py
- microsemi_program_eeprom.py
- microsemi_read_write_interactive.py

Start by powering the target board, then plugging the 5x2 connector of the Aardvark adapter into the header on the target board. Connect the USB cable from the PC to the Aardvark adapter.

Example 1 Dump Registers

The first example dumps a number of registers in the device. Run the 'microsemi_dump_regs.py' script by right-clicking and selecting "Edit with Idle". Idle is a Python editor and shell program. Launch the program by selecting Run->Run Module (or press F5).

You should see the following results or something similar:

```
Reg 0x0030: 0x1f
Reg 0x0031: 0x92
Reg 0x0032: 0x00
Reg 0x0033: 0x00
Reg 0x0034: 0x00
Reg 0x0035: 0x00
Reg 0x0036: 0x00
Reg 0x0037: 0x00
Reg 0x0038: 0x00
Reg 0x0039: 0x00
Reg 0x003a: 0x00
Reg 0x003b: 0x00
Reg 0x003c: 0x00
Reg 0x003d: 0x00
Reg 0x003e: 0x00
Reg 0x003f: 0x00
```

Example 2 Load an MFG File

The second example loads a .mfg file that was generated by the ZL30267 GUI. Reset the device on the board, then open microsemi_load_mfg.py by right-clicking and selecting Open with Idle. Launch the program by selecting Run->Run Module. The file to be loaded is named "ZL30267_DUT_Config.mfg". The program parses this file and applies all the register writes via the Aardvark adapter. The result should be a 156.25 MHz differential signal on OC3. The .mfg file assumes the ZL30267 is driven from a 125 MHz XO, and that OC3 drives into a DC-coupled 100 ohm load. If the device under test has a different clock, the output will be off-frequency, but should still be visible. If the termination is different, the signal may not be visible. Note that the script assumes the device has just come out of reset. If that is not the case, the results may be indeterminate.

Example 3 Program Device EEPROM

The third example programs a non-volatile configuration into config 0 of the device EEPROM. The Python script that writes the EEPROM image is named “microsemi_program_eeprom.py”. The EEPROM image file was generated by the GUI after loading the ZL30267_DUT_Config.mfg configuration from Example 2. Run the script by right-clicking, opening with Idle, and selecting Run->Run Module. After the script completes, the image is burned into EEPROM. Ensure that the AC[2:0] pins on the device are set to ‘000’, then reset the device. The device will generate a 156.25 MHz signal on OC3 after reset and power up. Note that the same assumptions from Example 2 apply here as well.

Example 4 Interactive Read and Write

The fourth example, microsemi_read_write_interactive.py, is used to provide a simple interface for interactive reading and writing of registers on the device. Run the program by right-clicking, opening with Idle, and selecting Run->Run Module.

From the Idle command line, read and write commands can be entered.

For example, to check the device ID registers, use the following commands:

```
>>> read(0x0030)
0x0030:  0x1f
>>> read(0x0031)
0x0031:  0x92
```

The write command is similar. To disable OC3, use the following command:

```
>>> write(0x0221,0x00)
>>>
```

To enable the output in the programmable diff format, use the following command:

```
>>> write(0x0221,0x02)
>>>
```

Note that addresses and values can be entered in decimal or hex. Hex numbers should be prefixed with a “0x” as in the examples above.

When the interactive session is complete, release the Aardvark resource by using the following command:

```
>>> ms.close(handle)
```

5. Troubleshooting

If a script is started, but ends with an exception before the program completes, the Aardvark resource may not be properly released. You may see the following error message on the next attempt to run the program:

```
Unable to open Aardvark device on port 0
Error code = -7
```

This state can usually be cleared from the Idle command line by issuing the following command:

```
>>> ms.close(0)
```

This will close the Aardvark resource, and make it available for the next program.

In rare cases, it may be necessary to quit the python environment and restart by right-clicking a script and using the ‘Edit with Idle’ option

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high–performance and radiation–hardened analog mixed–signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power–over–Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 4,800 employees globally. Learn more at **www.microsemi.com**.

ZLAN-650/06.17