

SPI Shortcut Datasheet

2009.07.07

Overview

The SparkFun SPI Shortcut is a simple and easy way to communicate with devices over SPI. Everything needed to start connecting with new devices is set up, with full user control over SPI options, all in a simple USB user interface.

Features

- User interface firmware that provides easy SPI communication out-of-the-box
- USB connection
- Four wire SPI interface breakouts with 3.3V input/output levels
- 3.3V and 5V sources on board with breakouts
- Optional TWI breakout

Use

To communicate with the SPI Shortcut through its serial user interface, start by downloading the virtual com port driver from <http://www.ftdichip.com/Drivers/VCP.htm> and install it using the instructions found at <http://www.ftdichip.com/Documents/InstallGuides.htm>. Then simply connect the device to a USB port and open a terminal. The the FT245 USB communication chip on the SPI Shortcut has an autobaud feature, so the device can communicate at any standard baud rate. Connect to the com port to which the device is connected, and press the reset button on the device. The first thing you should see is the SPI Shortcut main menu.

-----SparkFun SPI Shortcut-----

MAIN MENU:

- (1) Actions
- (2) Settings

If you haven't already, connect your SPI device to the appropriate pins on the SPI Shortcut. There are vias for 5V and 3.3V power supplies, and a via for ground.

The SPI Shortcut main menu has two options, to select an option in any of the menus, press the number corresponding to the menu item. The Actions Menu contains all of the functions for interacting with an SPI device. The Settings Menu is where you can adjust the SPI settings of the SPI Shortcut to suit you particular device.

Option 1: Actions Menu

The Actions menu looks like this:

ACTIONS MENU:

- (1) Send command string
- (2) Send ASCII characters
- (3) Continuous receive
- (4) Return to main menu

The send command string option allows you to send a string of up to 256 bytes that contains hex characters to send mixed with command options that give better control of signal line and command flow. When this option is chosen, this appears in the terminal:

Enter hex string of 256 values or less. Press return when finished.

RR = Receive, CH = Chip Select High, CL = Chip Select Low, DY = 10ms Delay

SPI Shortcut Datasheet

2009.07.07

At this point you can copy/paste or hand type your command string. For example, if you have a device that requires you to write the values 0x0A and 0x40, delay for 20ms and then read the response from the device, the command string would be as follows:

```
CL 40 0A DY DY RR CH
```

After pressing return, the device will perform the specified actions and then print a full report:

```
CS Low  
Sent 0x40  
Sent 0x0A  
Delay 10ms  
Delay 10ms  
R1 = 0xXX  
CS High  
String sent!
```

0xXX represents the data that would be received from the device. Subsequent receives would be named R2, R3, and so on.

Note: The device will automatically drive the Chip Select line high at the end of each command string. It will not, however, automatically drive the Chip Select line low at the beginning of each transaction. This must be done by including the CL command.

The send ascii characters function sends ASCII characters one at a time. When a character is entered, it automatically drives the CS line low, sends the character, and then drives the CS line high.

The continuous receive function simply clocks the SCLK line at the data rate specified by the settings options. It prints whatever is residing in the SPI data register and prints the data in a 16-bytes-per line format. It will do this continuously until CTRL+C is pressed.

Option 2: Settings Menu

The Settings Menu looks like this:

SETTINGS MENU:

- (1) Set clock polarity and phase
- (2) Set frequency
- (3) Set data order
- (4) Show current settings
- (5) Return to main menu

Using the options in this menu, you can manipulate all of the SPI communication options on the SPI Shortcut. Namely, clock polarity, clock phase, data order, and SPI clock frequency. The options are volatile, so they are only saved until power-down. The default settings for the SPI Shortcut are 500kHz SPI clock, 8 data bits, MSB sent first, and 0/0 CPOL CPHA. To change these options, enter the number for the particular option in the menu, and then follow the guides for changing modes. For example, to change the SPI clock frequency, hit 2 to select the menu item, and then choose the new frequency from the list:

SPI Shortcut Datasheet

2009.07.07

Frequency Options:

MODE	EFFECTIVE FREQUENCY
(1)	4MHz
(2)	2MHz
(3)	1MHz
(4)	500kHz
(5)	250kHz
(6)	125kHz
(7)	62.5kHz

To change the frequency to, say, 4MHz, hit 1. The SPI frequency will be changed and saved until power down. To view the current settings, choose option 4 from the Settings Menu.

Application Example 1: Communicating with a VS1053 audio decoder breakout board

The SparkFun VS1053 audio decoder breakout board is capable of decoding MP3 files and playing them through a 30 Ohm speaker. To easily test the functionality, the chip includes a “sine test” feature that is activated through a series of SPI commands. After the commands are entered, the chip plays a sine wave through the speaker. To use the SPI Shortcut to test this particular board, all we will need is the SPI Shortcut itself, the VS1053 breakout board, and a speaker. We connect the speaker to the speaker outs on the VS1053 board, and then connect the appropriate SPI lines. The VS1053 board power and ground is connected to the 5V out and ground on the SPI Shortcut.

The VS1053 spec states that the max SPI clock frequency is 5MHz, the data order is MSB, and the data is valid on the rising edge of the clock, so the default settings of the SPI Shortcut (500kHz, MSB, rising edge) are applicable.

The datasheet also specifies what bytes must be sent to the VS1053 in order to start the test. The chip must be configured into “test mode” and then the sine test must be initiated. To configure the chip into test mode, the following bytes must be sent:

0x02	Read command
0x00	Address
0x0C	Data MSB
0x20	Data LSB

After this sequence, the test must be initiated. This is done by sending the following bytes (page 35 in VS1053 datasheet):

```
0x53
0xEF
0x6E
0xAA
0x00
0x00
0x00
0x00
```

Now that we know the test sequence, all that has to be included now is the chip select commands. According to the datasheet, once the chip is put into test mode, the CS line becomes active high, so after configuring the mode, the CS line must be high for the second sequence. The chip must be selected only once for the first command sequence. Putting it all together, this gives us a final sequence of

```
CL 02 00 0C 20 CH DY 53 EF 6E AA 00 00 00 00
```

SPI Shortcut Datasheet

2009.07.07

Now that we have our sequence, we can fire up the SPI Shortcut in a terminal. In the menus, we choose Action Menu->Send command byte. Then type in the above sequence (no spaces) of sends/commands. Hit enter, and the command is sent. The speaker begins to play the sine tone and an action report is printed by the SPI Shortcut:

```
CS Low
Sent 0x02
Sent 0x00
Sent 0x0C
Sent 0x20
CS High
Delay 10ms
Sent 0x53
Sent 0xEF
Sent 0x6E
Sent 0xAA
Sent 0x00
Sent 0x00
Sent 0x00
Sent 0x00
String sent!
```

Application Example 2: Communicating with the SCP1000 pressure sensor

The SPC1000 pressure sensor reads temperature and pressure data from its environment. It is available on a breakout board from SparkFun (SEN-08161). The sensor is entirely configured and read via SPI communication. Now we will use the SPI Shortcut to quickly configure the device and test the temperature of the room. The steps to do this are as follows:

- 1)Configure the device in low noise mode
- 2)Delay 100ms
- 3)Configure the device in high resolution mode
- 4)Read the temperature data

From the data sheet, we determine that the commands for operation are as follows:

```
Write 0x2D to 0x02
Write 0x03 to 0x01
Write 0x02 to 0x03
Delay 100ms
Write 0x0A to 0x03
Write 0x21 with read configuration
Read next two bytes (MSB and LSB temperature data)
```

The SCP1000 communicates with CPHA = 1 and CPOL = 1, so we must change this in the settings menu of the SPI Shortcut. We connect the SCP1000 to the SPI Shortcut, start a terminal, and press reset. We then select Settings Menu->Set Clock Polarity/Phase->Mode 4. Now our SPI Shortcut is ready to communicate. The SCP1000 communication protocol states that register addresses be sent shifted up two bytes with the first two LSB containing either 00 for a register read, or 10 for a register write, so the register values above must be translated accordingly.

This gives us the SPI Shortcut command string of

```
CL 0A 2D CH // Configure device in low noise mode
```

SPI Shortcut Datasheet

2009.07.07

```
CL 06 03 CH
CL 0E 02 CH
DY DY DY DY DY DY DY DY DY DY // Delay 100ms
CL 0E 0A CH // Configure the device in high resolution mode
CL 84 RR RR CH // Read Temperature data
```

From the SPI Shortcut Main Menu, we select Actions Menu->Send command string. And then copy-paste the above commands to the terminal omitting the comments and spaces:

```
CL0A2DCHCL0603CHCL0E02CHDYDYDYDYDYDYDYDYDYDYDYDYDYCL0E0ACHCL84RRRRCH
```

After pressing enter, the device performs the commands and prints a report:

```
CS Low
Sent 0x0A
Sent 0x2D
CS High
CS Low
Sent 0x06
Sent 0x03
CS High
CS Low
Sent 0x0E
Sent 0x02
CS High
Delay 10ms
Delay 10ms
Delay 10ms
Delay 10ms
Delay 10ms
Delay 10ms
Delay 10ms
Delay 10ms
Delay 10ms
Delay 10ms
Delay 10ms
CS Low
Sent 0x0E
Sent 0x0A
CS High
CS Low
Sent 0x84
R1 = 0x01
R2 = 0xE1
CS High
String sent!
```

Our two received bytes are 0x01 and 0xE1, or as a 16-bit value, 0x01E1. According to the datasheet, this is the value in hex of 20 times the read temperature. Converting to Fahrenheit:

$0x01E1 = 481$ decimal $\Rightarrow 481/20 = 24.05$ degrees Celsius = 75.3 degrees Fahrenheit

Therefore, 75.3 degrees Fahrenheit is the temperature of the office testing environment and we have successfully tested the SCP1000 without using a dedicated microcontroller.