

Bootloader and Bridge Solutions Low-cost interface and Control for the QF1Da512

1) Introduction

An important attribute of both the QF1D512 and QF1Da512 is their capability to be rapidly and dynamically reconfigured in-circuit. Built on a foundation of RAM memory technology, they can be dynamically modified quickly and without the write cycle limitations inherent in non-volatile memory technologies such as Flash and EEPROM. In gaining this advantage, they are required to be initialized after a power cycle, requiring them to be controlled by a processing element of some kind, such as a microcontroller.

Quickfilter provides a variety of low-cost, ready-made microcontroller solutions for bootloading a filter configuration at initialization. These devices then release the SPI interface, which is native to the QF1D family of parts, and make it accessible to another SPI master, should one be available.

While the QF1Da512 is designed to interface directly with the serial peripheral interface (SPI) bus of microcontrollers, it is possible for it to interface to and be controlled by an Inter-IC (I2C) master. There are a number of hurdles that must be overcome in order to implement I2C control, requiring time and resources to resolve. Quickfilter's I2C-to-SPI Bridge implementations provide a ready-made solution at very low-cost. They act as bootloaders at power-up and, after initialization, as bridge devices, allowing an I2C master to communicate with and control a QF1D family part.

2) Functional Description

The most significant difference between SPI and I2C lies in their addressing schemes. Both interface protocols use a Master/Slave hierarchy and synchronous data transmission, with the SPI interface using two separate data lines (one input and one output) and the I2C interface using a single bi-directional data line.

On the SPI bus, each SPI device has its own chip select (CS) input. The SPI master selects any given SPI slave device by activating its chip select (for the QF1Da512, pulling it low). At any given time, only one SPI slave will be selected, so any commands sent are responded to by that device.

The I2C interface does not use chip selects. All I2C slave devices receive every command and the addressing scheme, imbedded in the software protocol, determines which slave is to respond to the command (see figure 5).

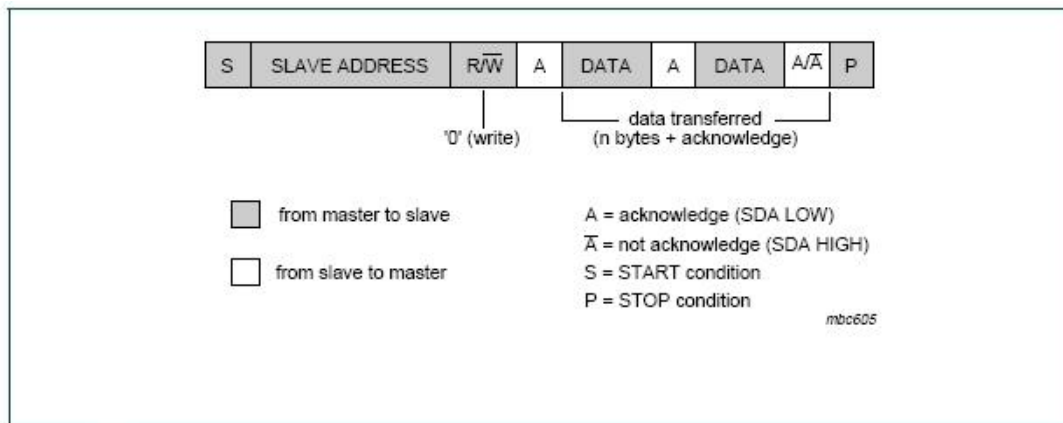


Figure 1: I2C Master/Slave Interaction

Each slave must detect its own address and respond only to those commands intended for it. For this reason, the I2C protocol includes arbitration and collision detection. While an SPI command is a single stream of bits from the master to the slave (and, for a READ command, a stream of bits back to the master), an I2C command is a succession of bit-string transfers and Acknowledge/Not-Acknowledge signals, interspersed throughout the command (see figure 6).

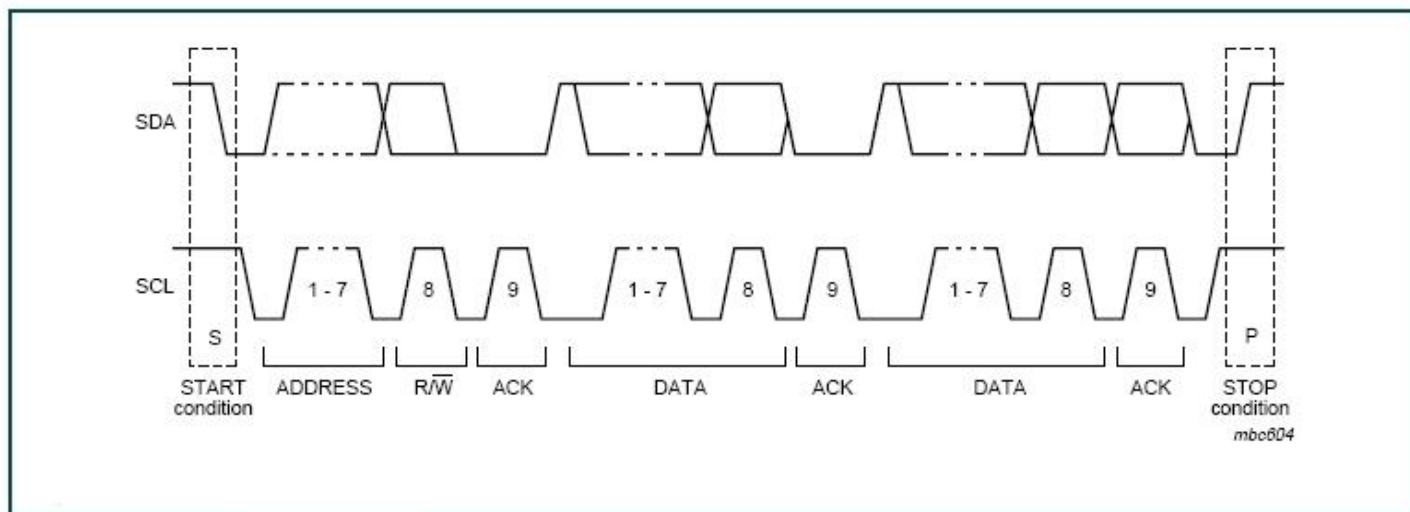


Figure 2: I2C Master/Slave Command/Response Timing

In interfacing the QF1Da512 to an I2C bus, the presence of an ACK/NAK bit approximately every 8th location in the I2C data stream, must be properly handled with respect to both the I2C master and the QF1Da512. This requires a response from the QF1Da512 (or at least, a simulated response from the interface to the QF1Da512). From the perspective of the QF1Da512, the presence of the ACK/NAK bits interferes with the QF1Da512's recognition of the address and data fields within the command and must be removed from the command stream.

Successfully interfacing the QF1Da512 to an I2C bus requires the use of a bridge circuit (see figure 3). The bridge can be implemented entirely in hardware (e.g., logic gates in a PLD or PAL) or through a combination of hardware and software (e.g., a low-cost microcontroller with embedded code). The implementations presented in this document use one of Freescale's MC9RS08KA1 or MC9RS08KA2 microcontrollers. In every case, the circuit must recognize and respond to both the start and stop conditions; reply automatically back to the I2C master with ACK's (signifying that all is well and ensuring that the master continues with the transmission); and, strip the ACK's from the stream of data sent on to the 1D (the data received by the 1D must look like a normal SPI command).

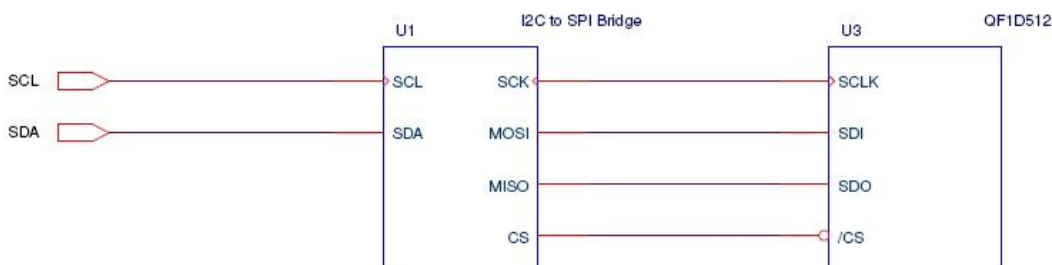


Figure 3: I2C to SPI Bridge Circuit

The I2C-to-SPI Bridge looks to the I2C master like an I2C slave and to the QF1Da512 like an SPI master. It intercepts the I2C commands, responding appropriately as applicable, strips out the embedded QuickFilter commands, and sends on the equivalent SPI command to the QF1Da512.

3) I2C Data Format

On reception of an I2C transmission, the I2C bridge circuit reads the I2C address field. When a message is intended for the Bridge, the proper I2C slave protocol is executed (reception of the address field is acknowledged as are the subsequent data fields, etc.) and the data fields are received and stored in memory. The data fields contain the SPI command which is to be sent on to the filter IC.

Figure 4 shows a typical format for I2C communication.

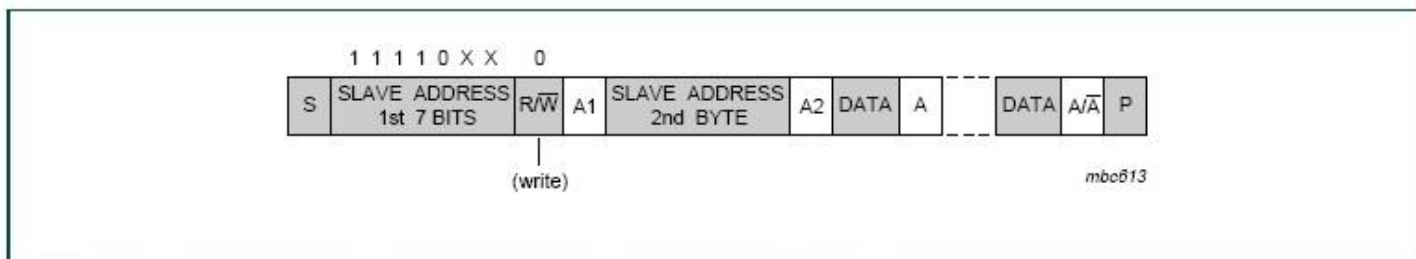


Figure 4: I2C Message Protocol

Figure 5 delineates how the fields are used by the Bridge.

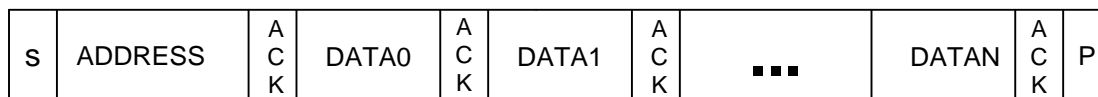


Figure 5: SPI Use of I2C Addressing Protocol

Where:

- S = Start Condition
- ADDRESS = I2C Slave Address
- ACK = Acknowledge/Not Acknowledge bit
- DATA0 = Filter IC designator (0, 1, 2, 3, etc. – as appropriate).
- ACK = Acknowledge/Not Acknowledge bit
- DATA1 = Quickfilter Command (see Table 1)
- ACK = Acknowledge/Not Acknowledge bit
- ...
- DATAN = Final data byte, if present
- ACK = Acknowledge/Not Acknowledge bit
- P = Stop Condition

Command Byte	Command	Additional Fields	Description
82	Write Configuration Register	Data2 ... DataN = Register Data Byte(s)	Minimum data sent is one byte (Data2).
86	Write Coefficient	Data2 ... DataN = Coefficient(s)	Minimum data sent is two data bytes (Data2 and Data3) which is one coefficient.
88	Volume Down	Data2 = Amount of volume decrease (0x00 = mute)	Not implemented in this release.
89	Volume Up	Data2 = Amount of volume increase (0xFF = max volume)	Not implemented in this release.
90	Reset Filter	None	Resets the Filter IC. Not implemented in this release.
91	Load Filter	None	Loads the filter indicated by the Filter IC Designator in DATA0

Table 1: Quickfilter Commands for I2C-to-SPI Bridge

4) Additional Functionality

Quickfilter provides multiple versions of the I2C-to-SPI Bridge implementation. Each implementation provides a different set of characteristics and additional functions to enable the end user to find a solution tailored to his/her particular requirements.

The alternatives available in each case are these:

1) **Bootloader Capability.**

The bootloader stores one or more filter configurations in its non-volatile memory and, upon power-up, initializes the filter(s) with one of these configurations.

2) **Number of Stored Configurations.**

A bridge-only configuration holds no stored filter configurations and acts solely as an I2C-to-SPI bridge, conveying messages from an I2C master to the filter IC. If the bootloader option is enabled, then the device will store one or more filter configurations – limited only by the size of the non-volatile memory available.

3) **Coefficient Size.**

The maximum size of coefficients is 32 bits. In most cases, the coefficients can be reduced to 16 values with no loss of audio quality. The use of the smaller coefficients allows the implementation to run in a smaller and lower cost microcontroller.

4) **Zero Insertion.**

For implementations that process two channels of data with a single filter, the coefficient field must interlace valid coefficients with zeros. This can be accomplished by building the filter configuration with the zeros in place or by allowing the bootloader to insert zeros as the filter is loaded into the filter IC. By allowing for zero insertion, the amount of memory required to store the filter coefficients is reduced by half.

5) **Target Device.**

Quickfilter provides two devices that can be used in conjunction with a Bridge/Bootloader, the QF1D512 and the QF1Da512. Each implementation is targeted specifically at one of the devices.

6) Volume Control.

This allows the I2C master to directly control the Gain registers in the QF1Da512 via I2C commands. It is not available for the QF1D512.

The functionality available in each of our software configurations is called out in the naming convention for the software (see Figure 6).

SW1Da	.	B#	.	F#	.	C#	.	Z#	.	V#	.	Rev#
												— —
										— —		— —
								— —		— —		— —
						— —		— —		— —		— —
				— —		— —		— —		— —		— —
		— —		— —		— —		— —		— —		— —
—		— —		— —		— —		— —		— —		— —

Revision Number
 Volume Control: 0 = Not enabled, 1 = Volume Control
 Zero Insertion: 0 = Not enabled, 1 = Coefficient zero insertion enabled
 Coefficient Size: 16 = 16 bits, 32 = 32 bits
 Number of Filters: 1 = One filter, 2 = Two filters, n = N filters
 Functionality: 0 = I2C Bridge only, 1 = Bootloader only, 2 = Bootloader with I2C-to-SPI Bridge
 Target Device: SW1D = QF1D512, SW1Da = QF1Da512

Figure 6: Software Nomenclature

5) Available Solutions

The solutions available from Quickfilter, at present, are listed here:

Boot_Only_1 (SW1Da.B1.F1.C16.Z0.V0.Rev1.0)

This is a bootloader-only device for the QF1Da512 supporting a single filter using 16-bit coefficients. Zero insertion is not implemented – it can be achieved by including it in the filter configuration. Dynamic control of the filter is possible via an external SPI master device.

I2C_Boot_1 (SW1Da.B2.F1.C16.Z0.V0.Rev1.0)

This is a Bootloader and I2C Bridge device for the QF1Da512 supporting a single filter using 16-bit coefficients. Zero insertion is not implemented – it can be achieved by including it in the filter configuration. Dynamic control of the filter is possible via an external I2C master device.

I2C_Boot_2 (SW1Da.B2.F1.C16.Z1.V0.Rev1.0)

This is a Bootloader and I2C Bridge device for the QF1Da512 supporting a single filter using 16-bit coefficients. Zero insertion is implemented allowing the resident filter configuration to consume less memory. Dynamic control of the filter is possible via an external I2C master device.



I2C_Boot_3 (SW1Da.B2.F2.C16.Z0.V1.Rev1.0)

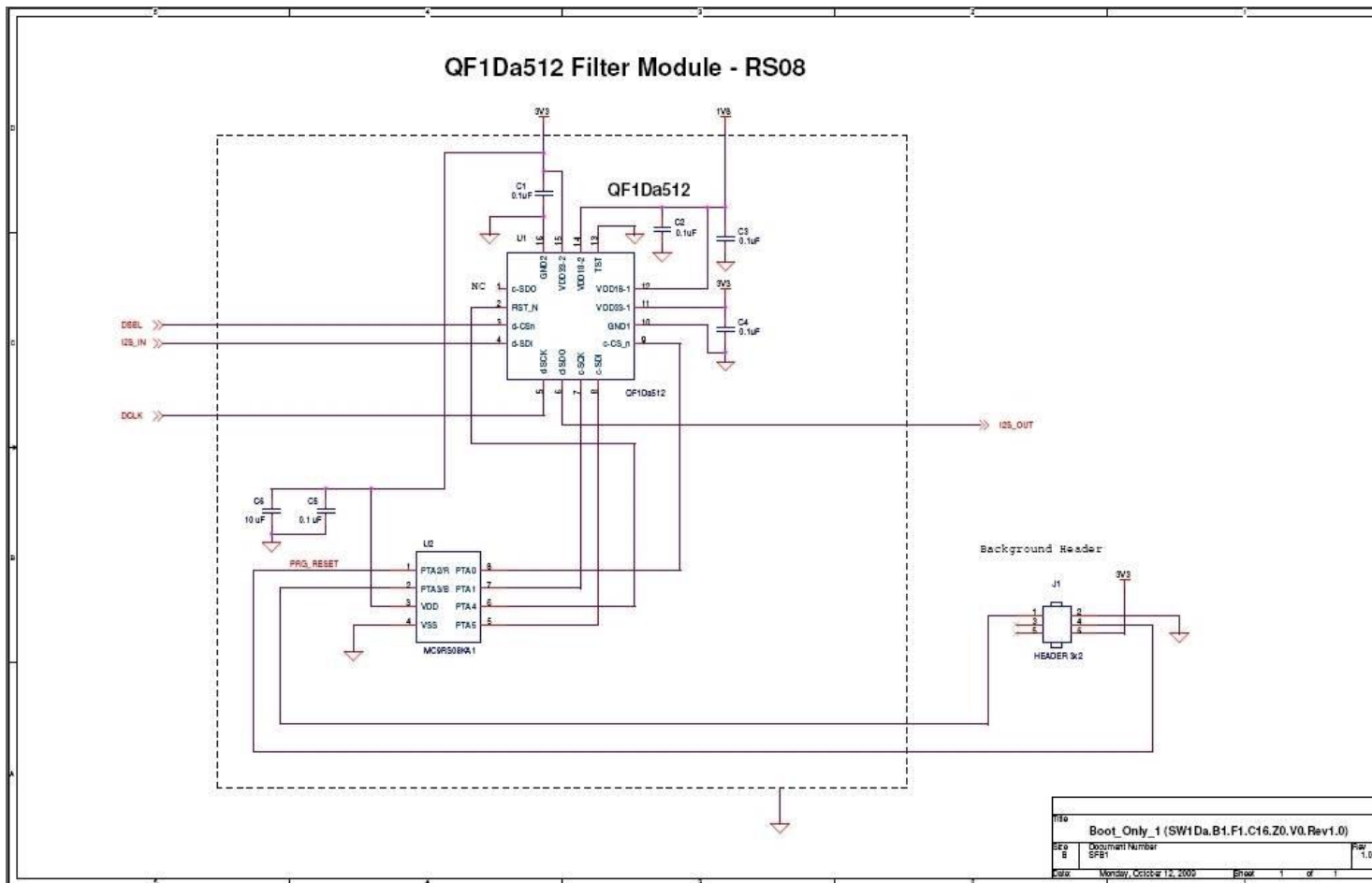
This is a Bootloader and I2C Bridge device for the QF1Da512 supporting two filters using 16-bit coefficients. Zero insertion is not implemented – it can be achieved by including it in the filter configuration. Dynamic control of the filter is possible via an external I2C master device.

I2C_Boot_4 (SW1Da.B2.F2.C16.Z1.V0.Rev1.0)

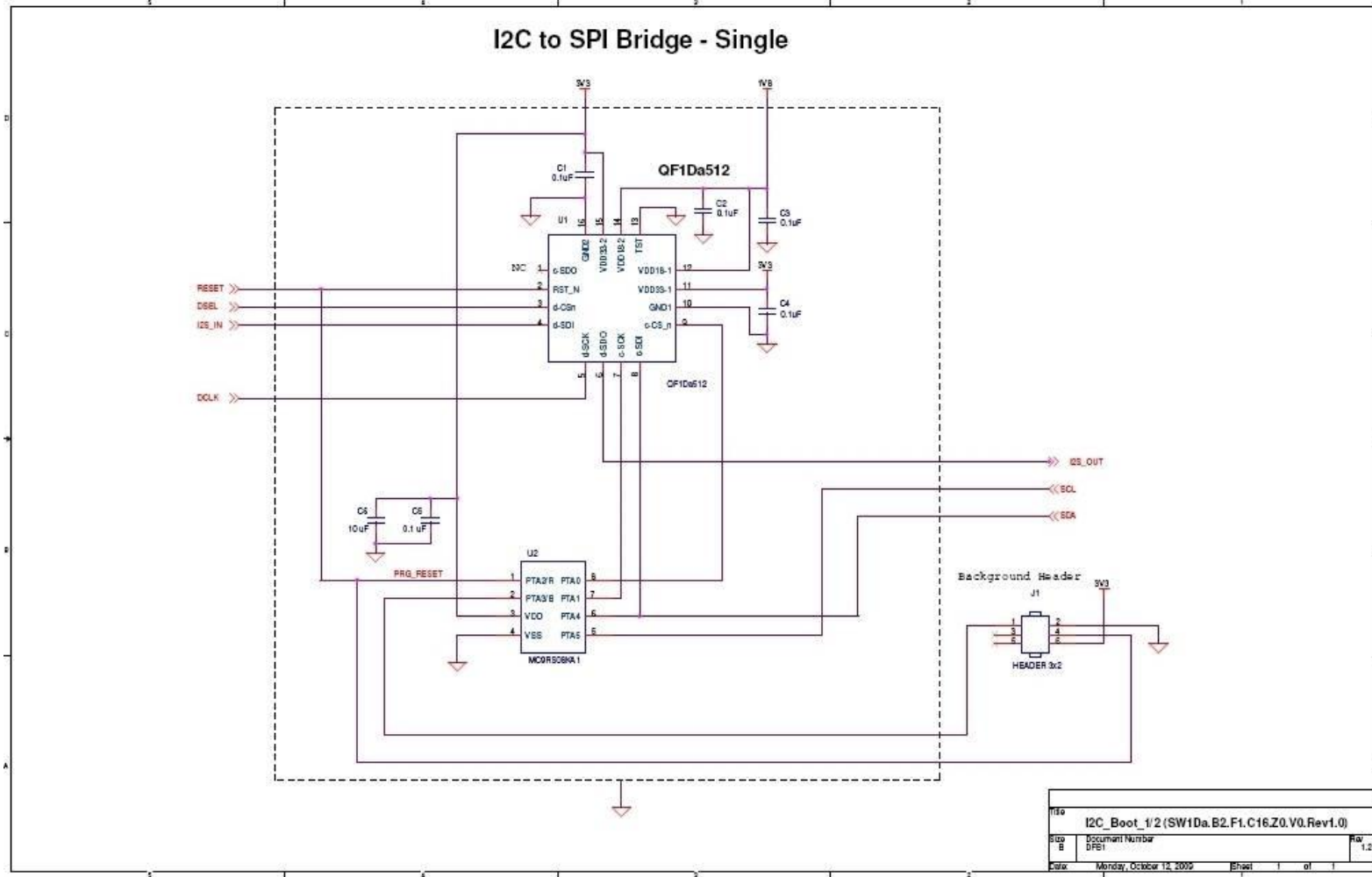
This is a Bootloader and I2C Bridge device for the QF1Da512 supporting two filters using 16-bit coefficients. Zero insertion is implemented allowing the resident filter configuration to consume less memory. Dynamic control of the filter is possible via an external I2C master device.

6) Schematics

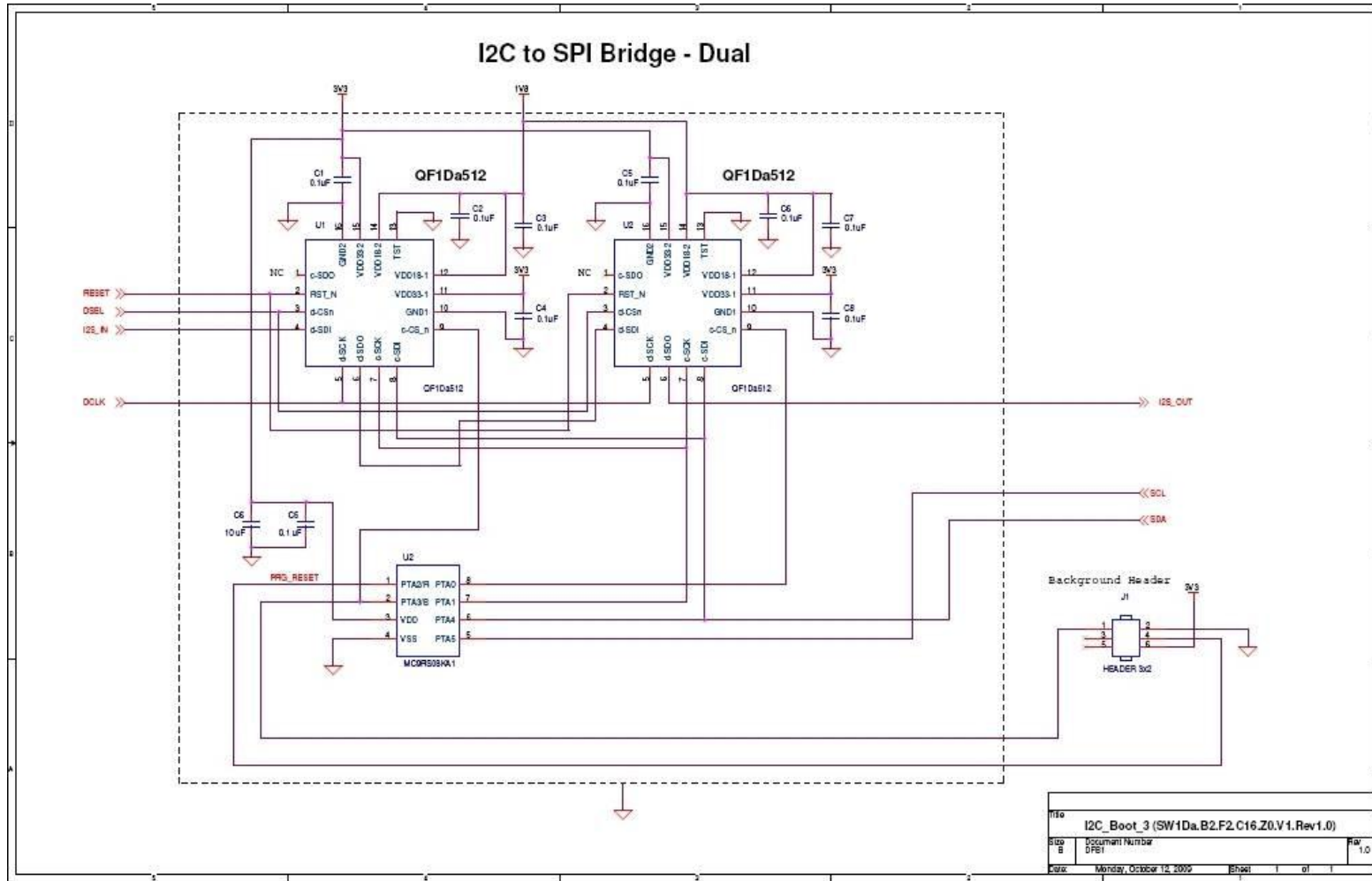
Boot_Only_1 (SW1Da.B1.F1.C16.Z0.V0.Rev1.0)



I2C_Boot_1 (SW1Da.B2.F1.C16.Z0.V0.Rev1.0) and I2C_Boot_2 (SW1Da.B2.F1.C16.Z1.V0.Rev1.0)



I2C_Boot_3 (SW1Da.B2.F2.C16.Z0.V1.Rev1.0) and I2C_Boot_4 (SW1Da.B2.F2.C16.Z1.V0.Rev1.0)





Contact Information:

Quickfilter Technologies, Inc.
1024 S. Greenville Avenue, Suite 100
Allen, TX 75002-3324

General:info@quickfilter.net
Applications:apps@quickfilter.net
Sales:sales@quickfilter.net
Phone:214-547-0460
Fax:214-547-0481

The contents of this document are provided in connection with Quickfilter Technologies, Inc. products. Quickfilter makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this publication. Except as set forth in Quickfilter's Standard Terms and Conditions of Sale, Quickfilter assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right.

Quickfilter's products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of Quickfilter's product could create a situation where personal injury, death, or severe property or environmental damage may occur. Quickfilter reserves the right to discontinue or make changes to its products at any time without notice.

© 2007 Quickfilter Technologies, Inc.
All rights reserved.

Quickfilter, the Quickfilter logo and combinations thereof, are trademarks of Quickfilter Technologies, Inc.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.