Keywords: passive keyless entry, PKE, remote keyless entry, RKE, signal conditioner, low power, proximity sensor, touch sensor, RF ID readers, meter tampering prevention

APPLICATION NOTE 5128

SPI Interface User Manual for the MAX1441 Automotive, Two-Channel Proximity and Touch Sensor

By: Youssof Fathi
Oct 19, 2011

Abstract: This user manual describes how to use interface firmware to facilitate evaluation of the MAX1441 automotive, two-channel proximity and touch sensor during product development. The document assumes that the user is familiar with operating the MAX1441 and MAX1441EVSYS. It also assumes the user is familiar with the MAX-IDE or similar tools to upload an application firmware into the MAX1441 flash memory. For more information and detailed specification of the MAX1441 proximity and touch sensor, refer to the MAX1441 data sheet. For instructions on how to operate the MAX1441EVSYS and upload application firmware into the MAX1441 flash memory using the MAX-IDE application, consult the MAX1441EVSYS data sheet.

The MAX1441 SPI interface firmware is a standalone firmware that includes an initialization section for programming startup values of the MAX1441 registers (including range, conversion rate, and excitation frequency). The firmware must be loaded into the MAX1441 through the USB/JTAG interface and is the only firmware residing in the flash memory that is executed by the embedded micro.

The CMAXQUSB+ command module controller board is used for the interface between the MAX1441 evaluation board (MAX1441EVSYS+) and your computer. The driver for the CMAXQUSB+ board can be downloaded from Maxim's website. Choose the USB driver that is appropriate for your system.

Figure 1 shows interconnections between different components of the setup. A second USB connection can be used instead of the power supply, to power the MAX1441 on the master board. The user interface program resides in the PC and facilitates communication between the user and the MAX1441 through a command-line window called MAX1441_Console. Communication rate between the MAX1441 and CMAXQUSB+ board is fixed at 31.25kHz.

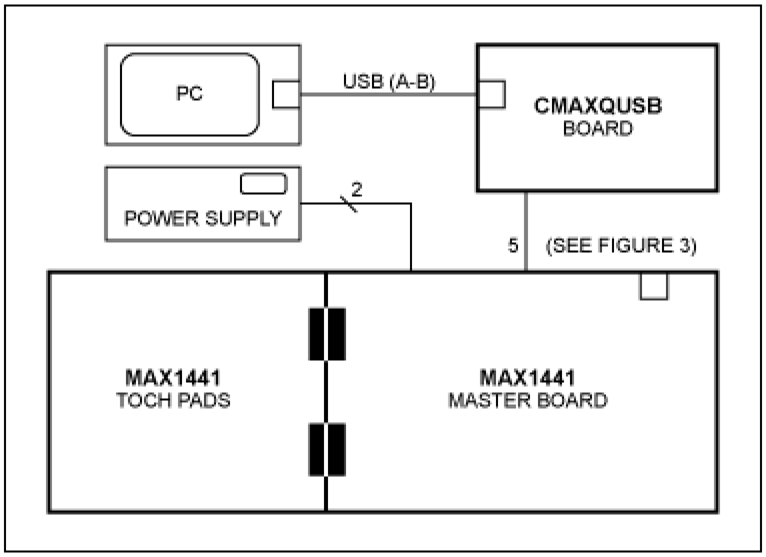


Figure 1. Block diagram of hardware setup.

Hardware Setup

1. Verify that JU105 on the MAX1441 evaluation board is in position 2–3.
2. Load the SPI *Interface_Rxx* firmware to the MAX1441 flash memory (using MAX-IDE or equivalent application).
3. Remove jumpers JU120 through JU124.
4. Connect the cable between the CMAXQUSB interface board and the MAX1441 evaluation board.
5. Power the MAX1441 evaluation board (if not already powered) through an external supply or a USB port:
    a. If powering through a USB port, two USB cables are needed: one to power and communicate with the CMAXQUSB board and one to power the MAX1441.

    b. If powering through an external supply, move jumper JU4 to position 2–3 and apply external power to 6V < VBAT < 28V terminals.
6. Connect a USB cable from the CMAXQUSB board to the PC.
7. Execute the *MAX1441_Console.exe* application.

See **Figure 3** for a drawing of the interface harness between the MAX1441EVSYS+ and CMAXQUSB+ interface boards.

Note: Jumpers JU120, JU121, JU122, JU123, and JU124 must be removed from the MAX1441 evaluation board before operating the SPI interface.

Using the MAX1441_Console Program

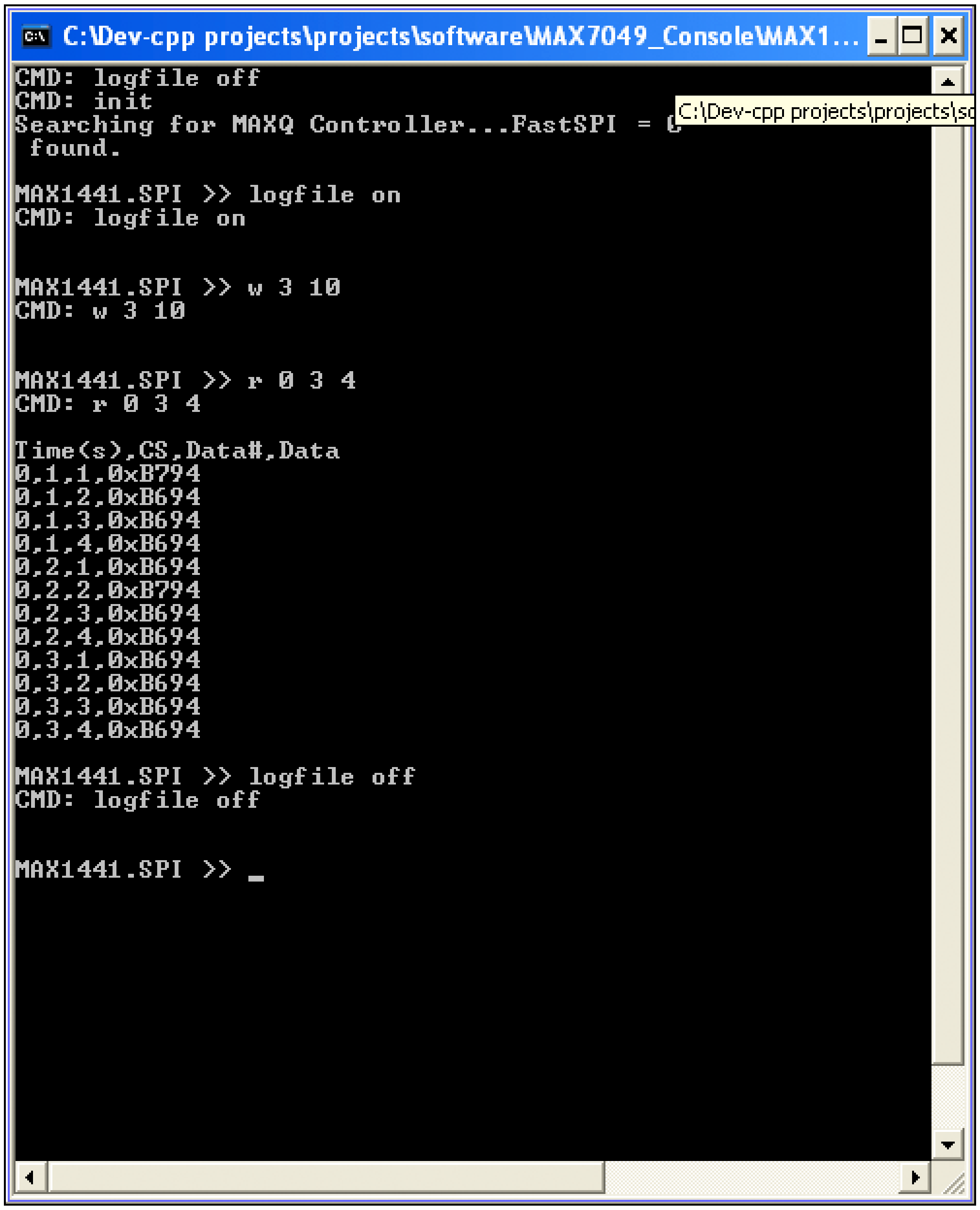
Once the SPI Interface firmware is written to the MAX1441 flash memory, and hardware is set up to communicate SPI, launch the *MAX1441_Console.exe* application. All modifications and inquiries of the

MAX1441 internal registers status is achieved through a Read or a Write command. For a list of commands acceptable by the *MAX1441_Console*, see **Table 1**.

Table 1. MAX1441_Console Commands

Command	Description	Example	Return
Init	System will search the USB ports with CMAXQUSB board connected	Init	CMD: init Found or Not Found message
logfile on	Will create and opens a log file "MAX1441log.txt" (If file already exists, it will be cleared)	logfile on	CMD: logfile on
logfile off	Will close the MAX1441log.txt file	logfile off	CMD: logfile off
delay value	Will cause a delay of **value** between each CS-cycle (**value** is in seconds)	Delay 3	CMD: delay 3
w address value	Write **value** to the register specified by **address**. **Value** is always in hexadecimal format	w 3 22	CMD: w 3 22
r address #ofData #ofCS-cycles	Read **#ofData** times for each **#ofCS-cycles** from register at **address**	r 0 2 3	CMD: r 0 2 3 Time(s),CS,Data#,Data 0,1,1,0xXXXX 0,1,2,0xXXXX 0,1,3,0xXXXX 0,2,1,0xXXXX 0,2,2,0xXXXX 0,2,3,0xXXXX

To write to an internal register, a Write command "w" must be issued. To read an internal register a Read command, "r" must be issued. Upper or lower case letters are valid for read and write commands. Any command other than the ones listed on the table above will result in an error message on the *MAX1441_Console* window and no action will be taken. For every Read or Write command from the *MAX1441_Console* to the CMAXQUSB board, a Chip Select (CS) line low/high cycle will be issued to the MAX1441EVSYS board. Duration of CS cycle is based on the number of data points requested.



C:\Dev-cpp projects\projects\software\MAX7049_Console\MAX1...
C:\Dev-cpp projects\projects\sc

```
CMD: logfile off
CMD: init
Searching for MAXQ Controller...FastSPI = 0
 found.

MAX1441.SPI >> logfile on
CMD: logfile on


MAX1441.SPI >> w 3 10
CMD: w 3 10


MAX1441.SPI >> r 0 3 4
CMD: r 0 3 4

Time(s),CS,Data#,Data
0,1,1,0xB794
0,1,2,0xB694
0,1,3,0xB694
0,1,4,0xB694
0,2,1,0xB694
0,2,2,0xB794
0,2,3,0xB694
0,2,4,0xB694
0,3,1,0xB694
0,3,2,0xB694
0,3,3,0xB694
0,3,4,0xB694

MAX1441.SPI >> logfile off
CMD: logfile off


MAX1441.SPI >> _
```

Figure 2. MAX1441_Console window.

Write Command

A Write command has a **w Address Value** format, where **Address** is the assigned register value given in Table 1 and **Value** is the value to be written to the register specified by **Address**. For example, command **w 5 2** will write "2" to CO1 register. **Value** is always in hexadecimal format.

Read Command

A Read command has the **r Address #ofData #ofCS-cycles** format, where, **Address** is address of the register to read; **#ofData** is number of readings from the register specified by **Address** during one CS cycle; and **#ofCS-cycles** is how many times CS line will cycle.

Maximum of 62 16-bit readings can be made from the MAX1441 during one CS- cycle. Each 16-bit reading will take approximate 600µs. A maximum of 1000 CS-cycle can be specified in a one command. The parameters **#ofData** and **#ofCS-cycles** apply to all registers; however, they are mostly relevant to CRSLT1H, CRSLT1L, CRSLT2H, and CRSLT2L registers. The **delay** command can extend monitoring of the MAX1441 performance for longer period of time. A delay of **value** seconds, specified in the delay command, will be applied after each CS-cycle. The format of response to a Read command is **Time**, **data#**, **cycle#**, **Data**, where **Time** is number of seconds elapsed since start of test, **data#** is the counter associated with **#ofData**, and **cycle#** is the counter associated with the **#ofCS-cycle**. **Table 2** lists registers and the associated addresses that can be written/read using the MAX1441_Console application. Consult the MAX1441 data sheet for detailed definition and function of each register.

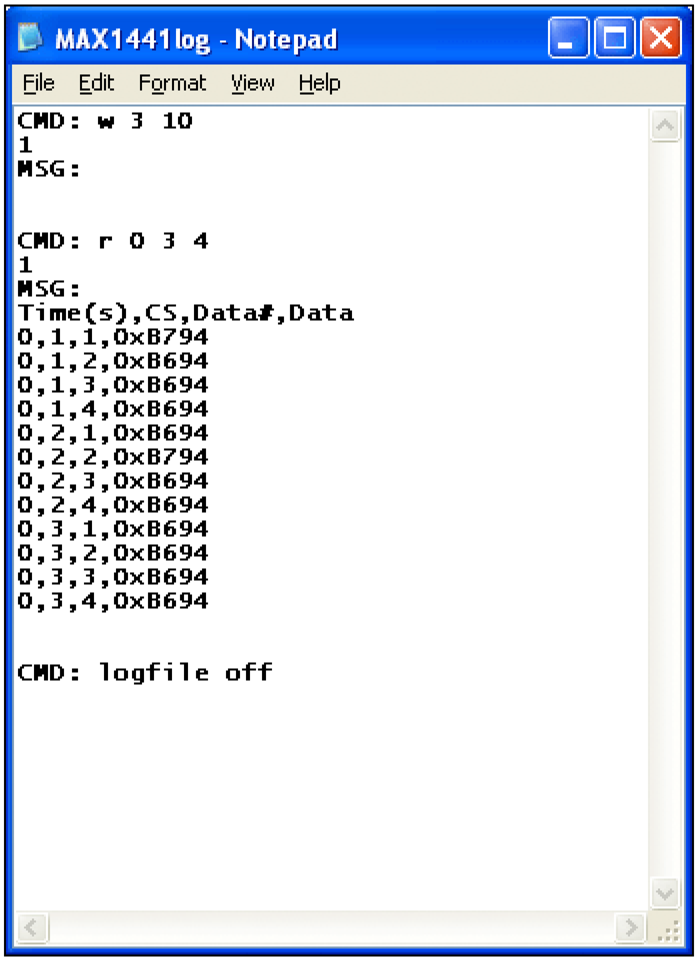
Table 2. Register Addresses in Read and Write Commands

Address	Register Affected By Write Command	Register(s) Content Reported in Response to Read Command		Description
		High Byte	Low Byte	
0	—	CRSLT2H	CRSLT1H	Channel 2 and Channel 1 conversion results (4 LSBs and Overflow bits not reported)
1	—	CRSLT1H	CRSLT1L	Channel 1 conversion results
2	—	CRSLT2H	CRSLT2L	Channel 2 conversion results
3	CRNG	0x00	CRNG	Input capacitance range
4	FEL	0x00	FEL	Excitation Frequency
5	FEB	0x00	FEB	Bandwidth of the spread spectrum modulation
6	DSB	0x00	DSB	Standby conversion rate
7	SSB2	0x00	SSB2	Channel 2 standby conversion rate subdivider
8	CO1	0x00	CO1	Channel 1 capacitance offset
9	CO2	0x00	CO2	Channel 2 capacitance offset
10	AT1H	0x00	AT1H	Channel 1 absolute threshold
11	AT2H	0x00	AT2H	Channel 2 absolute threshold
12	RT1H	0x00	RT1H	Channel 1 rate of change threshold
13	RT2H	0x00	RT2H	Channel 2 rate of change threshold
14	PD	0x00	PD	Power-Down Register
15	SCT	0x00	SCT	Single Conversion Register
16	AFEINRST	0x00	AFEINRST	AFE Interrupt Status Register
17	WU1	0x00	WU1	Channel 1 Wake-Up Control Register
18	WU2	0x00	WU2	Channel 2 Wake-Up Control Register
?? (Note 1) (Note 2)	—	0x??	0x??	Unrecognized address; address will be returned by the MAX1441

1. "??" denotes any invalid address. Valid addresses can be in decimal or hexadecimal format.
2. The command returns previous value in response to any invalid address, for example "ab". 0xab is a valid address and 0xABAB will be returned.

Storing Data to a File

All communications over the interface can be stored into a log file in .txt format. Storing of data into a file is started by issuing the **logfile on** command. Logging data can be stopped by issuing the **logfile off** command. The log file created is named "MAX1441.txt" and it will be cleared every time command **logfile on** is issued. As an example, content of a log file for command **r 0 3 4** is shown in Figure 3. To avoid overwriting a log file with useful data, rename the log file before new **logfile on** command is issued.



```
CMD: w 3 10
1
MSG:


CMD: r 0 3 4
1
MSG:
Time(s),CS,Data#,Data
0,1,1,0xB794
0,1,2,0xB694
0,1,3,0xB694
0,1,4,0xB694
0,2,1,0xB694
0,2,2,0xB794
0,2,3,0xB694
0,2,4,0xB694
0,3,1,0xB694
0,3,2,0xB694
0,3,3,0xB694
0,3,4,0xB694


CMD: logfile off
```

Figure 3. Example of log file content.

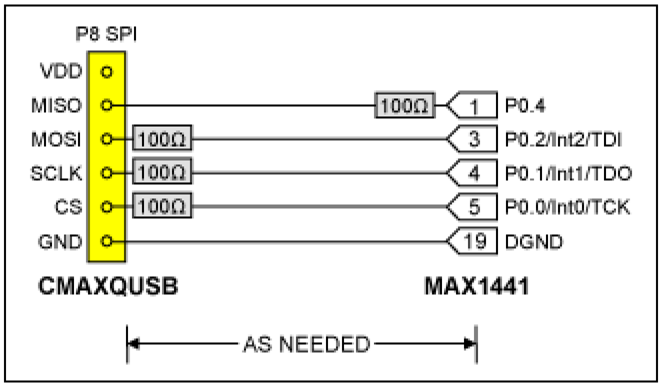


Figure 4. Interface harness between the MAX1441EVKIT and the CMAXQUSB board.

Related Parts	
CMAXQUSB	Evaluates: SPI and SMBus/I²C-Compatible Parts
MAX1441	Automotive, Two-Channel Proximity and Touch Sensor
MAX1441EVSYS	Evaluation System for the MAX1441