The following document contains information on Cypress products. The document has the series name, product name, and ordering part numbering with the prefix “MB”. However, Cypress will offer these products to new and existing customers with the series name, product name, and ordering part number with the prefix “CY”.

How to Check the Ordering Part Number

1. Go to www.cypress.com/pcn.
2. Enter the keyword (for example, ordering part number) in the **SEARCH PCNS** field and click **Apply**.
3. Click the corresponding title from the search results.
4. Download the Affected Parts List file, which has details of all changes

For More Information

Please contact your local sales office for additional information about Cypress products and solutions.

About Cypress

Cypress is the leader in advanced embedded system solutions for the world's most innovative automotive, industrial, smart home appliances, consumer electronics and medical products. Cypress' microcontrollers, analog ICs, wireless and USB-based connectivity solutions and reliable, high-performance memories help engineers design differentiated products and get them to market first. Cypress is committed to providing customers with the best support and development resources on the planet enabling them to disrupt markets by creating new product categories in record time. To learn more, go to www.cypress.com.

AN205429

FR, MB91460, SPI - Daisy Chain Communication

This application note describes how to communicate via SPI using the MB91467-USART with a peripheral which supports the Daisy Chain feature, and explains how DAC and Digital Potentiometer are interfaced to MB91467D USART.

Contents

1 Introduction

This application note describes how to communicate via SPI using the MB91467-USART with a peripheral which support Daisy Chain feature. In this note it explained how DAC and Digital Potentiometer are interfaced to MB91467D USART

Please note that this document only gives a rough overview about the communication. The described source codes were written for understanding not for code size or speed. Neither interrupts nor timers were used. Time critical coding is always done by simple flag polling or wait loop.

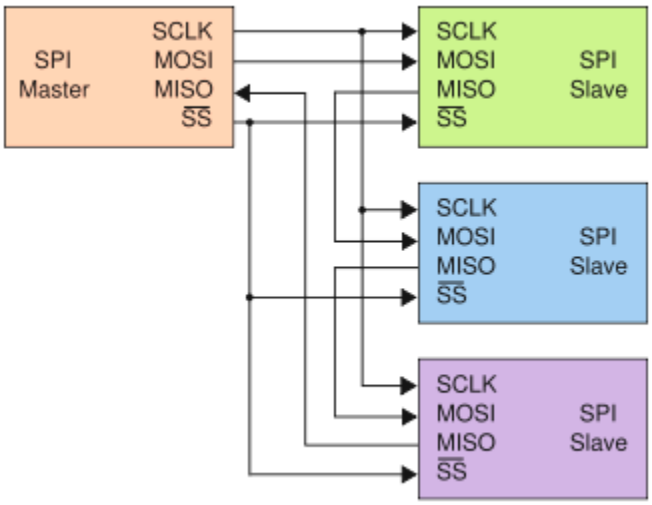
2 Daisy Chain Configuration

This Chapter Describes how Daisy Chain Configuration Works.

In Daisy chain SPI interface; command propagates from one device to another, connected in serial.

Figure below shows device connected in daisy chain configuration.

Figure 1. Micro controller with Daisy chain slaves

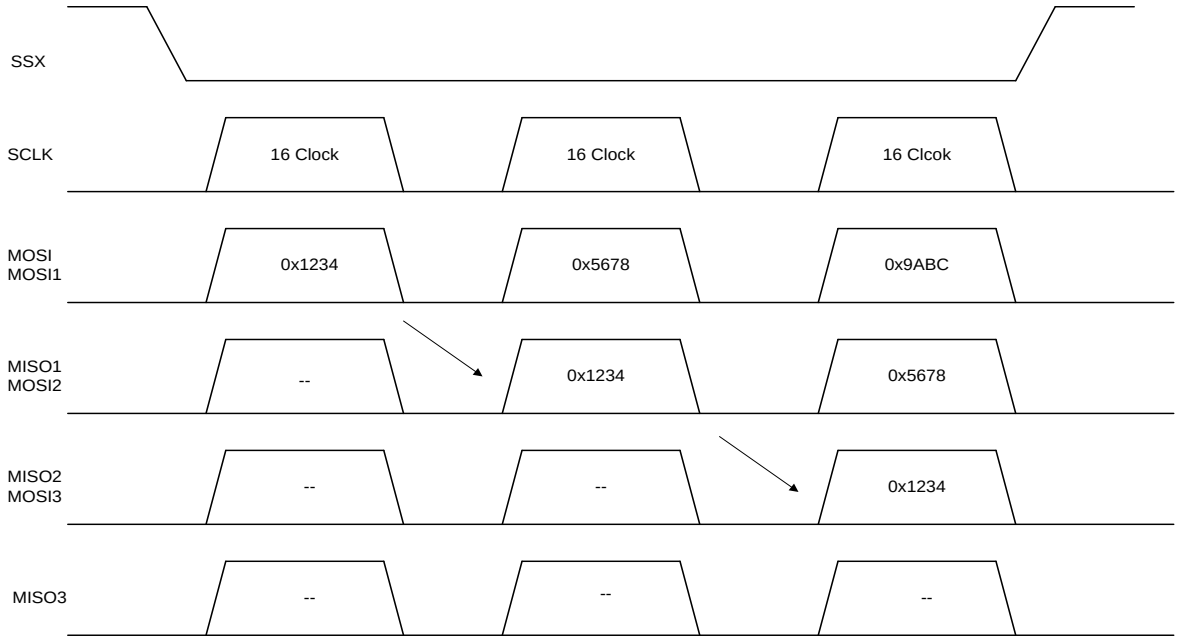


Active low slave select signal and SCLK are connected to all devices. Only the first slave in the chain receives the command data directly from the microcontroller. Every other slave in the network receives its MOSI data from the MISO output of the preceding slave in the chain.

The SPI port of each slave is designed to send out during the second group of clock pulses an exact copy of what it received during the first group of clock pulses. The whole chain acts as an SPI communication shift register.

Figure below shows command sequence if we want to send data 0x9ABC to Slave 1, data 0x5678 to Slave 2 and data 0x1234 to Slave 3.

Figure 2. Command Sequence



During the first command cycle (set of 16 SCLK pulses), 0x1234 gets loaded in the shift register of Slave 1. With active-low slave select remaining low, this data propagates through Slave1 and is output at MISO1 during the next command cycle. During this second command cycle, the data from MISO1 moves directly into MOSI2, and 0x1234 is loaded into Slave 2's shift register. Simultaneously, a new command, 0x5678, gets loaded into Slave 1’s shift register, thus overwriting its previous command.

In the third command cycle, the first command, 0x1234, is loaded into Slave 3's shift register. The second command, 0x5637, gets loaded into Slave 2, and Slave 1 receives a new command, 0x9ABC. All three Slaves now have a command, which they received through the daisy-chain in their shift registers. When active-low Slave select goes high, the loaded command is executed.

3 Daisy Chain peripheral

This chapter describes how to communicate with SPI – daisy chain Peripheral.

3.1 Digital to Analog Converter

MAX5232/33 is dual 10-bit voltage output Digital to Analog converter. It works on single 5V supply and has internal reference voltage at 2.465V and features full scale output of 4.092V.

The MAX5232/33 has the following pin-out:

Figure 3. Pin Diagram MAX5232/33

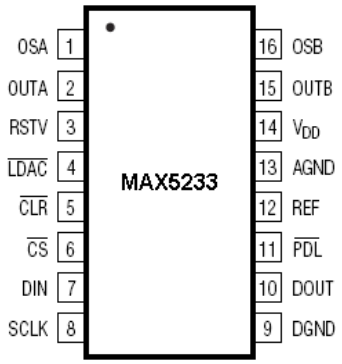


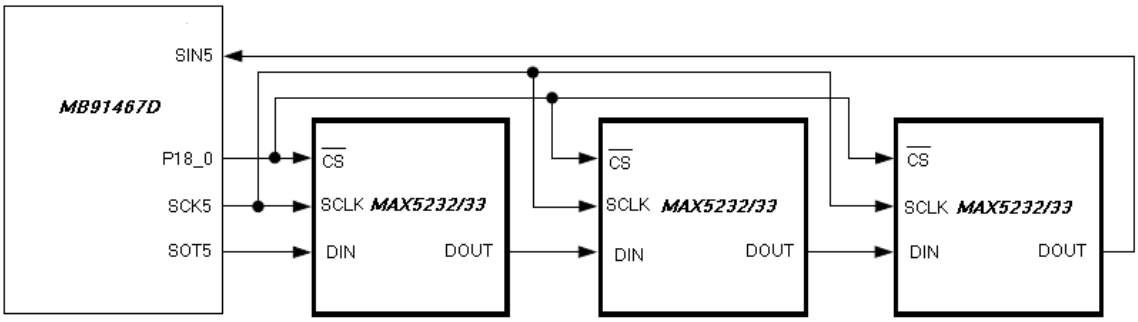
Table 1. Command Sequence

PIN	Name	Function
1	OSA	DAC A Offset Adjust
2	OUTA	DAC A Output
3	RSTV	Reset Value Input 1: Connect to VDD to select mid scale as the reset value. 0: Connect to DGND to select zero as the reset value.
4	LDACX	Load DACs A and B
5	CLRX	Clear Input. Both DAC outputs go to zero or mid scale. Clears both DAC internal registers (input register and DAC register) to its predetermined (RSTV) state.
6	CSX	Chip-Select Input
7	DIN	Serial Data Input. Data is clocked in on the rising edge of SCLK.
8	SCLK	Serial Clock Input
9	DGND	Digital Ground
10	DOUT	Serial Data Output
11	PDLX	Power-Down Lockout. Disables shutdown of both DACs when low.
12	REF	Reference Output. Reference provides a 2.465V (MAX5233) or 1.234V (MAX5232) nominal output.
13	AGND	Analog Ground
14	VDD	Positive Power Supply. Bypass VDD with a 0.1µF capacitor in parallel with a 4.7µF capacitor to AGND, and bypass VDD with a 0.1µF capacitor to DGND.
15	OUTB	DAC B Output
16	OSB	DAC B Offset Adjust

3.1.1 Connection to MB91467D

The DAC can be connected as shown in the following schematic. Please note that no power supply pins and other MCU-Pins are drawn than those for the connection to the DAC.

Figure 4. Connections between MAX5232/33 and MB91467D

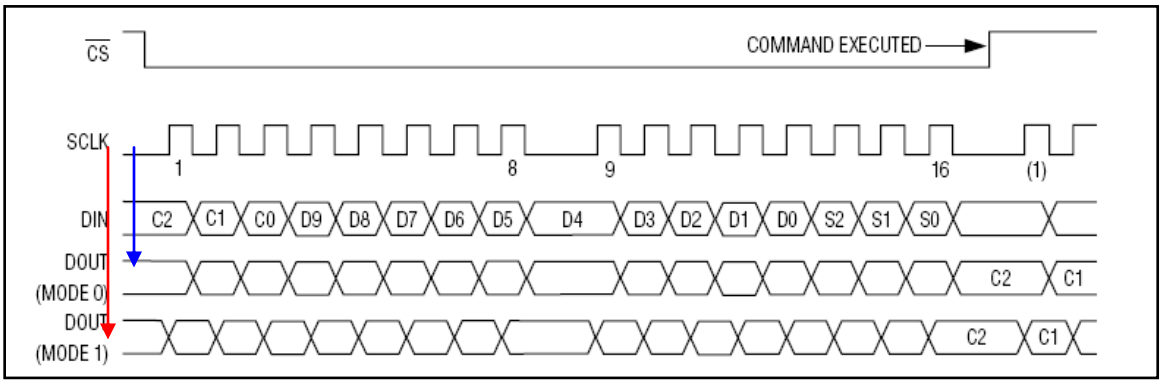


Apart from this, RSTV pin of all DACs are connected to ground to select zero as the reset value. PDLX pin is connected to ground to disables shutdown of both DACs when low.

3.1.2 Communication Timing

Detailed description of the timing and timing parameters can be found in the corresponding datasheet of the MAX5232/33 DAC.

Figure 5. Write Cycle



Above figure shows the timing for the serial interface. The serial word consists of 3 control bits followed by 10 data bits (MSB first) and 1 sub-bit as described in the command table shown below. When the three control bits are all zeros or all 1, D9–D6 are used as additional control bits, allowing for greater DAC functionality.

The digital inputs allow any of the following:

- Loading the input register(s) without updating the DAC register(s),
- Updating the DAC register(s) from the input register(s),
- Updating the input and DAC register(s) simultaneously.

The control bits and D9–D6 allow the DACs to operate independently.

The control bits and D9–D6 determine which registers update and the state of the registers when exiting shutdown.

The 3-bit control and D9–D6 determine the following

- Registers to be updated
- Selection of the power-down and shutdown modes

The general timing diagram of Figure 2.3 illustrates data acquisition. Driving CSX low enables the device to receive data. Otherwise the interface control circuitry is disabled. With CS low, data at DIN is clocked into the register on the rising edge of SCLK. As CSX goes high, data is latched into the input and/or DAC registers, depending on the control bits and D9–D6. The maximum clock frequency guaranteed for proper operation is 13.5MHz.

3.1.3 Used DAC Commands

The following commands are used in the code example below:

Table 2. DL1806 Commands

16 Bit SERIAL WORD					FUNCTION
C2	C1	C0	D9 to D0	S2-S0	
0	0	1	10-bit DAC data	000	Load input register A; DAC registers are unchanged
0	1	0	10-bit DAC data	000	Load input register A; all DAC registers are updated.
0	1	1	10-bit DAC data	000	Load all DAC registers from the shift register (start up both DACs with new data, and load the input registers).
1	0	0	X X X X X X X X X X	000	Update both DAC registers from their respective input registers (start up both DACs with data previously stored in the input registers).
1	0	1	10-bit DAC data	000	Load input registers B; DAC registers are unchanged.
1	1	0	10-bit DAC data	000	Load input registers B; all DAC registers are updated.
1	1	1	P1A P1B X X X X X X X X	000	Shut down both DACs, respectively; according to bits P1A and P1B (see Table 2.6). Internal bias and reference remain active.
0	0	0	0 0 1 X X X X X X X	000	Update DAC registers A from input register A (start up DAC A with data previously stored in input register A).
0	0	0	0 1 1 P1A P1B X X X X X	000	Full Power-Down. Power down the main bias generator and shut down both DACs, respectively, according to bits P1A and P1B (see Table 2.6).
0	0	0	1 0 1 X X X X X X X	000	Update DAC registers B from input register B (start up DAC B with data previously stored in input register B).
0	0	0	1 1 0 P1A X X X X X X	000	Shut down DAC A according to bit P1A (see Table 2.6).
0	0	0	1 1 1 P1B X X X X X X	000	Shut down DAC B according to bit P1B (see Table 2.6).
0	0	0	1 0 0 0 X X X X X X	000	Mode 0. DOUT clocked out on SCLK falling edge (default).
0	0	0	1 0 0 1 X X X X X X	000	Mode 1. DOUT clocked out on SCLK rising edge.

Table 3. P1 Shutdown Modes

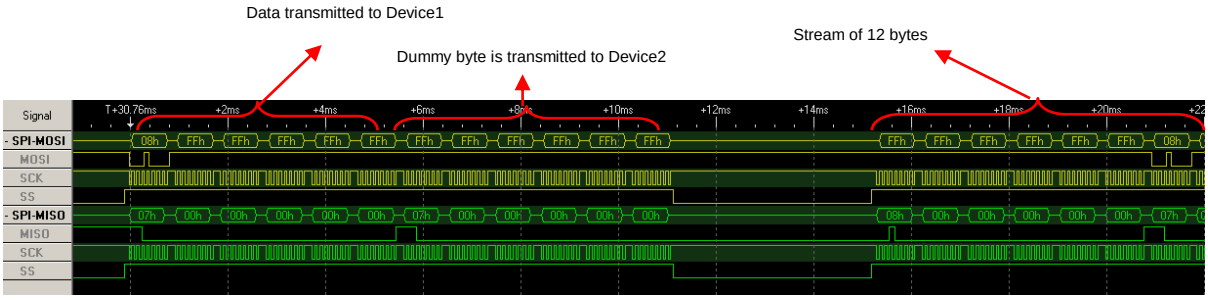
P1(A/B)	SHUTDOWN MODE
0	Shut down with internal 1kΩ load to GND
1	Shut down with internal 200kΩ load to GND

3.1.4 Example Code

The following code shows how to establish a communication with DAC.

For MAX5232/33 we need to transmit 2 bytes, hence if we connect 2 devices as per SPI Daisy chain connection than in total we need to transmit 4 bytes.

Figure 6. MAX5232/33 communication example




```
#define MAX_BYTE 4 // n * 2 -> n = number of device in series

void max5232_write_mode(unsigned char start, unsigned char mode)
{
       unsigned char count;

       if (device> MAX_BYTE)
       {
              printf("MAX5232 Param Error!\r\n");
              return;
       }

        spi_cs_enable() ;

       for(count=1;count<=MAX_BYTE;count++)
       {
              if (count == start)
              {
       // mode : b'0 0 0 1 0 0 0 X  -> DOUT clocked out on SCLK falling edge
       // mode : b'0 0 0 1 0 0 1 X  -> DOUT clocked out on SCLK rising edge
       // mode : b'0 0 0 1 1 0 P1A X -> Shut down DAC A according to bit P1A
       // mode : b'0 0 0 1 1 1 P1B X -> Shut down DAC B according to bit P1B
       // mode : b'1 1 1 P1A P1B X X -> Shut down both DACs, respectively
                     spi_tx_8Bit(mode);
                     spi_tx_8Bit(0x00);
                     count++;
              }
              else
              {
                     spi_tx_8Bit(0x00);
              }
       }
              spi_cs_disable() ;
}
```

In function, parameter `start` refers to the position of first byte to be inserted in the stream of MAX_BYTE (12 bytes) stream. There are 6 potentiometer inside DS1806, parameter `pot_no` refers to the potentiometer for which we want to set wiper setting. Byte `0xFF` is transmitted for wiper setting of other potentiometers with first two MSB bits set, wiper setting of other potentiometers of the same device or those of other device is not changed.

4 Appendix A

This chapter names the related documents

4.1 Related Documents

- AN205101 - F²MC-8L/8FX/16LX16FX , MB89XXX / MB95XXX / MB90xxx / MB96Xxx, Performing SPI
- MAX5232/33 Maxim data sheet for MAX5232/33
- DS1806 Dallas data sheet for DS1806

5 Additional Information

Information about Cypress Microcontrollers can be found on the following Internet page:

http://www.cypress.com/cypress-microcontrollers

The software example related to this application note is:

91460_uart_spi_master_DaisyChain_max5233

91460_uart_spi_master_DaisyChain_DS1806

It can be found on the following Internet page:

http://www.cypress.com/cypress-mcu-product-softwareexamples

Document History

Document Title: AN205429 - FR, MB91460, SPI - Daisy chain Communication

Document Number: 002-05429

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	NOFL	07/13/2008	V1.0; HPi, initial version
*A	5135942	NOFL	02/12/2016	Converted Spansion Application Note “MCU-AN-300101-E-V10” to Cypress format
*B	5872490	AESATMP9	09/04/2017	Updated logo and copyright.
*C	6054547	NOFL	02/12/2018	Updated links Updated template

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709