AN746

Interfacing Microchip's MCP41XXX and MCP42XXX Digital Potentiometers to a PICmicro® Microcontroller

Author: *Craig L. King & Ezana Haile*
Microchip Technology Inc.

OVERVIEW

The MCP41XXX and MCP42XXX family of digital potentiometers communicate using a standard 3-wire SPI™ compatible interface. This application note will discuss communications between these devices and a PIC16F876 microcontroller. The code supplied with this application note will include both absolute and relocatable assembly code, written for both hardware SPI and firmware SPI implementations.

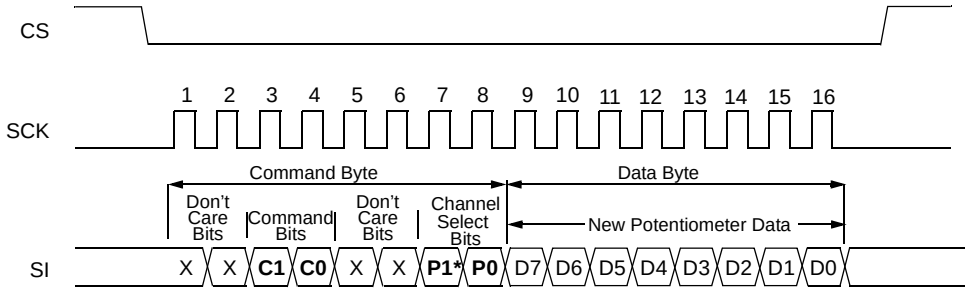
COMMUNICATION

Instructions for the MCP41XXX and MCP42XXX devices consist of 16 clock cycles or two bytes. Figure 1 shows the format of these two bytes using a standard 3-wire SPI interface. The first byte is the command byte which must contain four bits to program the state of the digital potentiometer. The command byte determines the operation that is performed as well as identifies which potentiometer will execute the command (MCP42XXX devices contain two potentiometers). The second byte is the data byte. The MCP41XXX and MCP42XXX potentiometers are 8-bit or 256 tap potentiometers. All 8 bits in the data byte are wiper data bits. Depending on the state of P0 and P1 in the command byte, the data byte sets the wiper's position or positions.

The four command bits to consider are bits 4:5 (C0:C1) and bits 0:1 (P0:P1). C0 and C1 determine which command is being issued. For the MCP41XXX and MCP42XXX devices, there are three possible commands:

- Write new data to potentiometer(s)
- Shutdown potentiometer(s)
- NOP (No Operation).

The MCP42XXX devices contains two potentiometers, P0 and P1. P0 uses pins 5, 6 and 7. P1 uses pins 8, 9 and 10. Using these two bits, the user can select either, both or neither potentiometer. A '1' for either P1 or P0 will cause the data to be written to the respective data register and a '0' for P1 or P0 will cause no change. The MCP41XXX devices contain only one potentiometer. For these devices, P1 is a don't care.



C1	C0	Command	Command Summary
0	0	None	No Command will be executed
0	1	Write Data	Write the data contained in Data Byte to the potentiometer(s) determined by the potentiometer selection bits
1	0	Shutdown	Potentiometer(s) determined by potentiometer selection bits will enter Shutdown Mode. Data bits for this command are don't cares
1	1	None	No Command will be executed

P1*	P0	Potentiometer Selections
0	0	Dummy Code: Neither Potentiometer affected
0	1	Command executed on Potentiometer 0
1	0	Command executed on Potentiometer 1
1	1	Command executed on both Potentiometers

*P1 is a don't care bit for the MCP41XXX

FIGURE 1: *Instruction sequence and command byte summary for MCP41XXX AND MCP42XXX DIGITAL potentiometers*

SPI™ is a trademark of Motorola Inc.

IMPLEMENTATION

Appendix A has the absolute assembly code using hardware SPI implementation. The Synchronous Serial Port (SSP) is first initialized to clock data out on the falling edge, drive the clock high when idle and clock with a frequency of FOSC/4. This is done by setting the SSPCON register to 0x30. Communication to the potentiometer is initiated by pulling the chip select line low. A command byte of 13h is then loaded into the SSPBUF of the PIC16F876. This command byte value will instruct a write command to potentiometer P0 and P1. The BF bit in the SSPSTAT register is then monitored. When this bit is high, the 8-bit transfer is complete. Once this transfer is complete, the data byte is then loaded into the SSPBUF register with the resistor value to be programmed into the digital potentiometer. This example shows the potentiometer being set to code 8Ch (140d). Again, the BF bit of the SSPSTAT register is monitored. Once this byte is transferred, chip select is raised and the instruction is complete. At the rising edge of chip select, the MCP41XXX or MCP42XXX will change the wiper position.

Appendix B shows absolute assembly code using firmware SPI implementation. The same pins are used to generate the clock and data signals as the hardware SPI example. Port initialization occurs, setting the $\overline{CS}$, CLK and SDO port pins to outputs. The `TRANSMIT` routine handles the firmware SPI implementation, excluding the toggling of $\overline{CS}$. Communication is again initiated by pulling chip select low. The Working Resister (W) is pre-loaded with the command byte and a call to the `TRANSMIT` routine is made. This routine generates 8 clock cycles and also sends out the W register on the data line. Upon completion of this routine, the W register is then pre-loaded with the data byte, or the resistor value to be programmed into the digital potentiometer. A second call to the `TRANSMIT` routine follows and communication is completed by raising the chip select line. At the rising edge of chip select, the MCP41XXX or MCP42XXX device will execute the command and the wiper position will change.

Appendix C is a relocatable version of the hardware SPI code in Appendix A. Appendix D is the relocatable version of the software SPI code in Appendix B. The linker script file (16F876.lkr) is shown in Appendix E. This file controls where the relocatable segments are placed in the PIC16F876 program memory and defines the processors available RAM space for the linker. Please consult the MPASM™ User's Guide for more details on how to write and assemble relocatable code.

SCHEMATIC

The code for this application note was developed on the MXDEV™ Analog Evaluation Driver Board along with the MCP42XXX evaluation board. An equivalent circuit of the board used in this application note is shown in Appendix F. A full schematic of the MXDEV driver board and the MCP42XXX evaluation board can be found in the MXDEV Driver Board Users Manual (DS51221) and the MCP42XXX Evaluation Board Users Guide (DS51229). The SPI communication lines CLK and DOUT use pins RC3 and RC5, respectively. The chip select signal is generated using port pin RA4. The PIC16F876 uses crystal oscillator, X1. An MCP130 is used as the power on reset device. An MCP42010 is used as the digital potentiometer. A volt meter, V1, is used to measure the voltage at the output of the wiper which was used to determine the correct operation of the code.

CONCLUSION

The example code given in this application note shows how to interface either an MCP41XXX or an MCP42XXX device to a PICmicro. Multiple styles of implementation were given to allow the developer to use this code in almost any end-user application.

MEMORY USAGE

In the Digital Potentiometer, the following memory was used:

Program Memory:	24 bytes
Data Memory:	0 bytes
EEPROM Memory:	0 bytes

REFERENCES

MCP41XXX/MCP42XXX, Single/Dual Digital Potentiometer with SPI Interface, Microchip Technology, Document DS11195, 2000.

KEYWORDS

1	Potentiometer
2	Digital Potentiometers
3	MCP4XXXX
4	MCP41XXX
5	MCP42XXX
6	Interfacing PICmicro microcontroller
7	SPI
8	PIC16F876

APPENDIX A: ASSEMBLY CODE USING HARDWARE SPI IMPLEMENTATION

```
;*******************************************************************************************
;                                                                                          *
;         Interfacing Microchip’s MCP42xxx digital potentiometer to the PICmicro MCU       *
;                                                                                          *
;                  - THIS PROGRAM IS ABSOLUTE ASSMEBLY USING THE HARDWARE                  *
;                  SPI MODULE TO PROGRAM THE DIGITAL POTENTIOMETER                         *
;                                                                                          *
;                                                                                          *
;*******************************************************************************************
;                                                                                          *
;                                                                                          *
;         Filename:          POTSPI1.ASM                                                   *
;         Date:              11.07.2000                                                    *
;         File Version:      1.00                                                          *
;                                                                                          *
;         Assembler:         MPASM     VERSION 2.50                                        *
;                                                                                          *
;         PROGRAMER:         PRO MATE DEVICE PROGRAMMER, VERSION 5.20.00                   *
;                                                                                          *
;         File Required:     PIC16F876.inc                                                 *
;                                                                                          *
;         Author:            Ezana Haile                                                   *
;         Company:           Microchip Technology Incorporated                             *
;                                                                                          *
;                                                                                          *
;*******************************************************************************************
;                                                                                          *
;                                                                                          *
;         This code demonstrates how Microchip’s MCP42xxx Digital Potentiometer            *
;         (Pot) is interfaced to the PICmicro MCU (PIC16F876). The Potentiometer            *
;         requires a serial communication to program the command byte and the data         *
;         byte. This MCU has a built-in Serial Peripheral Interface (SPI) which can        *
;         be used to program the pot effectively. The following program illustrates        *
;         how to interface the digital pot using the MCU’s SPI.                            *
;                                                                                          *
;         To change the command byte or the Pot wiper position the user must change        *
;         the COMMAND and R_VALUE variables properly and reprogram the MCU.                *
;                                                                                          *
;                                                                                          *
;*******************************************************************************************


                  #include <p16f876.inc>
         ERRORLEVEL -302
                  __CONFIG  _BODEN_OFF & _PWRTE_OFF & _CP_OFF & _WDT_OFF & _XT_OSC
```

```
;*****************************************************************************************
;********************  EQUATES  ******************************************************
;*****************************************************************************************

CS                  EQU      H'00'              ;CHIP SELECT

COMMAND             EQU      H'13'              ;VARIABLE FOR THE COMMAND BYTE
R_VALUE             EQU      D'140'             ;VARIABLE FOR THE RESISTANCE VALUE


;*****************************************************************************************
;********************  PROGRAM ORIGIN  ***********************************************
;*****************************************************************************************


                    ORG      0X00

;-----------------------------------------------------------------------------------------
;--------------------  PORTB AND SPI SETTING  --------------------------------------------
;-----------------------------------------------------------------------------------------


                    BSF      STATUS, RP0        ;SPECIFY BANK 1
                    MOVLW    H'00'
                    MOVWF    TRISA              ;SET PORTA AS AN OUTPUT
                    MOVWF    TRISC              ;SET PORTB AS AN OUTPUT
                    BCF      STATUS, RP0        ;SPECIFY BANK 0

                    CLRF     PCLATH             ;ENSURE PCLATH BIT 3 IS CLEARED
                    CLRF     INTCON             ;ENSURE ALL INTERRUPTS ARE DISABLED
                    MOVLW    0x30               ;
                    MOVWF    SSPCON             ;SET SYNC SERIAL PORT CONTROL REGISTER

;-----------------------------------------------------------------------------------------
;--------------------  PROGRAM ROUTINE  --------------------------------------------------
;-----------------------------------------------------------------------------------------


                    BCF      PORTA, CS          ;SELECT THE POT

                    MOVLW    COMMAND            ;LOAD THE COMMAND BYTE IN THE ACCUMULATOR
                    CALL     TRANSMIT           ;TRANSMIT THE COMMAND BYTE

                    MOVLW    R_VALUE            ;LOAD THE RESISTANCE VALUE IN THE ACCUMULATOR
                    CALL     TRANSMIT           ;TRANSMIT THE RESISTANCE VALUE

                    BSF      PORTA, CS          ;UNSELECT THE POT

                    GOTO     FINISH             ;FINISH

;-----------------------------------------------------------------------------------------
;--------------------  TRANSMISSION SUBROUTINE  ------------------------------------------
;-----------------------------------------------------------------------------------------


TRANSMIT            BCF      STATUS, RP0        ;SPECIFY BANK 0
                    MOVWF    SSPBUF             ;PLACE DATA IN BUFFER TO SEND

                    BSF      STATUS, RP0        ;SPECIFY BANK 1
LOOP                BTFSS    SSPSTAT, BF        ;CHECK IF TRANSMISSION IS COMPLETE
                    GOTO     LOOP               ;
                    BCF      STATUS, RP0        ;SPECIFY BANK 0

                    RETURN                      ;RETURN FROM SUBROUTINE
```

```
;*****************************************************************************************

FINISH              GOTO      FINISH

                    END

;********************  END OF PROGRAM  ***************************************************
```

APPENDIX B: ASSEMBLY CODE USING FIRMWARE SPI IMPLEMENTATION

```
;*******************************************************************************************
;                                                                                          *
;       Interfacing Microchip's MCP42xxx digital potentiometer to the PICmicro MCU         *
;                                                                                          *
;                - THIS PROGRAM IS ABSOLUTE ASSMEBLY USING PORTB TO ACT AS SOFTWARE        *
;                SPI AND PROGRAM THE DIGITAL POTENTIOMETER                                 *
;                                                                                          *
;*******************************************************************************************
;                                                                                          *
;                                                                                          *
;       Filename:          POT_PRG.ASM                                                     *
;       Date:              11.02.2000                                                      *
;       File Version:      1.00                                                            *
;                                                                                          *
;       Assembler:         MPASM    VERSION 2.50                                           *
;                                                                                          *
;       PROGRAMER:         PRO MATE DEVICE PROGRAMMER, VERSION 5.20.00                     *
;                                                                                          *
;       File Required:     PIC16F876.inc                                                   *
;                                                                                          *
;       Author:            Ezana Haile                                                     *
;       Company:           Microchip Technology Incorporated                               *
;                                                                                          *
;                                                                                          *
;*******************************************************************************************
;                                                                                          *
;                                                                                          *
;       This code demonstrates how Microchip's MCP42xxx Digital Potentiometer              *
;       (Pot) is interfaced to the PICmicro MCU (PIC16F876). The Potentiometer             *
;       requires a serial communication to program the command byte and the data           *
;       byte. This MCU has a built-in serial communication system which can be             *
;       used to program the pot effectively. However, for this application three           *
;       lines from PORTC are dedicated to serially program the Pot. This method            *
;       is selected to clearly demonstrate the Digital pot programing sequence.            *
;                                                                                          *
;       PORTC:<3,5> are connected to SCK, and SI pins of the Pot, respectively.            *
;       Chip Select is connected to PORTA: <4>.                                            *
;                                                                                          *
;       To change the command byte or the Pot wiper position the user must change         *
;       the COMMAND and R_VALUE variables properly and reprogram the MCU.                  *
;                                                                                          *
;                                                                                          *
;*******************************************************************************************


                #include <p16f876.inc>
                ERRORLEVEL -302
                __CONFIG _BODEN_OFF & _PWRTE_OFF & _CP_OFF & _WDT_OFF & _XT_OSC
```

```
;*****************************************************************************************
;********************  VARIABLES  ********************************************************
;*****************************************************************************************


                    CBLOCK   H'20'
                    OUT, COUNT                  ;VARIABLES USED TO TRANSMIT SERIAL DATA
                    ENDC

;*****************************************************************************************
;********************  EQUATES  **********************************************************
;*****************************************************************************************


CS                  EQU      H'00'              ;PORTC:   <7> CHIP SELECT
SCLK                EQU      H'03'              ;         <6> SERIAL CLOCK
SI                  EQU      H'05'              ;         <5> SERIAL DATA

COMMAND             EQU      H'13'              ;VARIABLE FOR THE COMMAND BYTE
R_VALUE             EQU      D'140'             ;VARIABLE FOR THE RESISTANCE VALUE

;*****************************************************************************************
;********************  PROGRAM ORIGIN  ***************************************************
;*****************************************************************************************


                    ORG      0x00

;-----------------------------------------------------------------------------------------
;--------------------  PORT SETTING  -----------------------------------------------------
;-----------------------------------------------------------------------------------------


                    BCF      STATUS, RP0        ;MAKE SURE TO BE IN BANK 0
                    BCF      STATUS, RP1        ;

                    BSF      STATUS, RP0        ;GO TO BANK 1
                    MOVLW    0x00
                    MOVWF    TRISA              ;MAKE PORTA AN OUTPUT
                    MOVWF    TRISC              ;MAKE PORTC AN OUTPUT
                    BCF      STATUS, RP0        ;RETURN TO BANK 0

                    CLRF     PORTC              ;CLEAR PORTB

;-----------------------------------------------------------------------------------------
;--------------------  PROGRAM ROUTINE  --------------------------------------------------
;-----------------------------------------------------------------------------------------


                    BCF      PORTA, CS          ;SELECT THE POT

                    MOVLW    COMMAND                    ;LOAD THE COMMAND BYTE IN THE ACCUMULATOR
                    CALL     TRANSMIT           ;TRANSMIT THE COMMAND BYTE

                    MOVLW    R_VALUE                    ;LOAD THE RESISTANCE VALUE IN THE ACCUMU-
LATOR
                    CALL     TRANSMIT           ;TRANSMIT THE RESISTANCE VALUE

                    BSF      PORTA, CS          ;UNSELECT THE POT

                    GOTO     FINISH                     ;FINISH
```

```
;-------------------------------------------------------------------------------------------
;--------------------  TRANSMISSION SUBROUTINE  ---------------------------------------------
;-------------------------------------------------------------------------------------------


TRANSMIT  MOVWF     OUT                         ;MOVE W TO 'OUT' VARIABLE

                    MOVLW     0X08              ;LOAD A COUNTER TO 'COUNT' THE BIT
                    MOVWF     COUNT             ;TRANSMISSION

L_SHIFT             BTFSC     OUT, 7            ;MONITOR THE 7TH BIT
                    GOTO      HI
                    BCF       PORTC, SI         ;IF LOW: CLEAR SERIAL-IN LINE
                    GOTO      PASS
HI                  BSF       PORTC, SI         ;IF HI: SET SERIAL-IN LINE

PASS                BSF       PORTC, SCLK       ;SET SERIAL CLOCK: HI
                    RLF       OUT, F            ;ROTATE OUT LEFT
                    BCF       PORTC, SCLK       ;SET SERIAL CLOCK: LOW

                    DECFSZ    COUNT, F          ;DECREMENT COUNTER UNTIL ITS ZERO
                    GOTO      L_SHIFT

                    CLRF      PORTC             ;WHEN COUNTER IS ZERO IT'S END OF
                                                ;TRANSMISSION
                    RETURN                      ;RETURN FROM SUBROUTINE

;*******************************************************************************************


FINISH              GOTO      FINISH

                    END


;********************  END OF PROGRAM  ***************************************************
```

Appendix C: RELOCATABLE VERSION OF THE HARDWARE SPI CODE IN APPENDIX A

```
;*******************************************************************************************
;                                                                                          *
;       Interfacing Microchip's MCP42xxx digital potentiometer to the PICmicro MCU         *
;                                                                                          *
;                 - THIS PROGRAM IS ABSOLUTE ASSEMBLY USING THE HARDWARE                   *
;                 SPI MODULE TO PROGRAM THE DIGITAL POTENTIOMETER                          *
;                                                                                          *
;                                                                                          *
;*******************************************************************************************
;                                                                                          *
;                                                                                          *
;       Filename:          POTSPI1.ASM                                                     *
;       Date:              11.07.2000                                                      *
;       File Version:      1.00                                                            *
;                                                                                          *
;       Assembler:         MPASM     VERSION 2.50                                          *
;                                                                                          *
;       PROGRAMER:         PRO MATE DEVICE PROGRAMMER, VERSION 5.20.00                     *
;                                                                                          *
;       File Required:     PIC16F876.inc                                                   *
;                                                                                          *
;       Author:            Ezana Haile                                                     *
;       Company:           Microchip Technology Incorporated                               *
;                                                                                          *
;                                                                                          *
;*******************************************************************************************
;                                                                                          *
;                                                                                          *
;       This code demonstrates how Microchip's MCP42xxx Digital Potentiometer              *
;       (Pot) is interfaced to the PICmicro MCU (PIC16F876). The Potentiometer             *
;       requires a serial communication to program the command byte and the data           *
;       byte. This MCU has a built-in Serial Peripheral Interface (SPI) which can          *
;       be used to program the pot effectively. The following program illustrates          *
;       how to interface the digital pot using the MCU's SPI.                              *
;                                                                                          *
;       To change the command byte or the Pot wiper position the user must change          *
;       the COMMAND and R_VALUE variables properly and reprogram the MCU.                  *
;                                                                                          *
;                                                                                          *
;*******************************************************************************************


                #include <p16f876.inc>
                ERRORLEVEL -302
                __CONFIG _BODEN_OFF & _PWRTE_OFF & _CP_OFF & _WDT_OFF & _XT_OSC
```

```
;*****************************************************************************************
;********************  EQUATES  ******************************************************
;*****************************************************************************************

CS                  EQU      H'00'              ;CHIP SELECT

COMMAND             EQU      H'13'              ;VARIABLE FOR THE COMMAND BYTE
R_VALUE             EQU      D'140'             ;VARIABLE FOR THE RESISTANCE VALUE

;*****************************************************************************************
;********************  PROGRAM ORIGIN  ***********************************************
;*****************************************************************************************

prog1               code

;-----------------------------------------------------------------------------------------
;--------------------  PORTB AND SPI SETTING  --------------------------------------------
;-----------------------------------------------------------------------------------------

                    BSF      STATUS, RP0        ;SPECIFY BANK 1
                    MOVLW    H'00'
                    MOVWF    TRISA              ;SET PORTA AS AN OUTPUT
                    MOVWF    TRISC              ;SET PORTB AS AN OUTPUT
                    BCF      STATUS, RP0        ;SPECIFY BANK 0

                    CLRF     PCLATH             ;ENSURE PCLATH BIT 3 IS CLEARED
                    CLRF     INTCON             ;ENSURE ALL INTERRUPTS ARE DISABLED
                    MOVLW    0x30               ;
                    MOVWF    SSPCON             ;SET SYNC SERIAL PORT CONTROL REGISTER

;-----------------------------------------------------------------------------------------
;--------------------  PROGRAM ROUTINE  --------------------------------------------------
;-----------------------------------------------------------------------------------------

                    BCF      PORTA, CS          ;SELECT THE POT

                    MOVLW    COMMAND            ;LOAD THE COMMAND BYTE IN THE ACCUMULATOR
                    CALL     TRANSMIT           ;TRANSMIT THE COMMAND BYTE

                    MOVLW    R_VALUE            ;LOAD THE RESISTANCE VALUE IN THE ACCUMULATOR
                    CALL     TRANSMIT           ;TRANSMIT THE RESISTANCE VALUE

                    BSF      PORTA, CS          ;UNSELECT THE POT

                    GOTO     FINISH             ;FINISH

;-----------------------------------------------------------------------------------------
;--------------------  TRANSMISSION SUBROUTINE  ------------------------------------------
;-----------------------------------------------------------------------------------------

TRANSMIT            BCF      STATUS, RP0        ;SPECIFY BANK 0
                    MOVWF    SSPBUF             ;PLACE DATA IN BUFFER TO SEND

                    BSF      STATUS, RP0        ;SPECIFY BANK 1
LOOP                BTFSS    SSPSTAT, BF        ;CHECK IF TRANSMISSION IS COMPLETE
                    GOTO     LOOP               ;
                    BCF      STATUS, RP0        ;SPECIFY BANK 0

                    RETURN                      ;RETURN FROM SUBROUTINE
```

```
;*****************************************************************************************

FINISH              GOTO      FINISH

                    END

;********************  END OF PROGRAM  ***************************************************
```

Appendix D: RELOCATABLE VERSION OF THE FIRMWARE SPI CODE IN APPENDIX B

```
;*******************************************************************************************
;                                                                                          *
;       Interfacing Microchip's MCP42xxx digital potentiometer to the PICmicro MCU         *
;                                                                                          *
;               - THIS PROGRAM IS ABSOLUTE ASSMEBLY USING PORTB TO ACT AS SOFTWARE         *
;               SPI AND PROGRAM THE DIGITAL POTENTIOMETER                                  *
;                                                                                          *
;*******************************************************************************************
;                                                                                          *
;                                                                                          *
;       Filename:           POT_PRG.ASM                                                    *
;       Date:               11.02.2000                                                     *
;       File Version:       1.00                                                           *
;                                                                                          *
;       Assembler:          MPASM     VERSION 2.50                                         *
;                                                                                          *
;       PROGRAMER:          PRO MATE DEVICE PROGRAMMER, VERSION 5.20.00                    *
;                                                                                          *
;       File Required:      PIC16F876.inc                                                  *
;                                                                                          *
;       Author:             Ezana Haile                                                    *
;       Company:            Microchip Technology Incorporated                              *
;                                                                                          *
;                                                                                          *
;*******************************************************************************************
;                                                                                          *
;                                                                                          *
;       This code demonstrates how Microchip's MCP42xxx Digital Potentiometer              *
;       (Pot) is interfaced to the PICmicro MCU (PIC16F876). The Potentiometer             *
;       requires a serial communication to program the command byte and the data           *
;       byte. This MCU has a built-in serial communication system which can be             *
;       used to program the pot effectively. However, for this application three           *
;       lines from PORTC are dedicated to serially program the Pot. This method            *
;       is selected to clearly demonstrate the Digital pot programing sequence.            *
;                                                                                          *
;       PORTC:<3,5> are connected to SCK, and SI pins of the Pot, respectively.            *
;       Chip Select is connected to PORTA: <4>.                                            *
;                                                                                          *
;       To change the command byte or the Pot wiper position the user must change          *
;       the COMMAND and R_VALUE variables properly and reprogram the MCU.                  *
;                                                                                          *
;                                                                                          *
;*******************************************************************************************


                #include <p16f876.inc>
                ERRORLEVEL -302
                __CONFIG  _BODEN_OFF & _PWRTE_OFF & _CP_OFF & _WDT_OFF & _XT_OSC
```

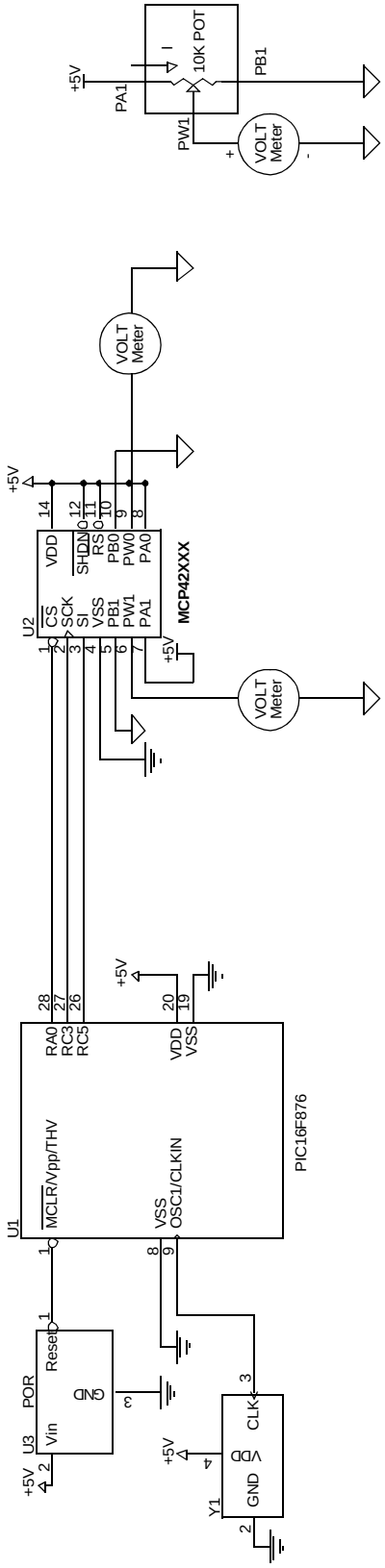
```
;*****************************************************************************************
;********************  VARIABLES  ********************************************************
;*****************************************************************************************

                udata

                OUT     res     1       ;VARIABLE USED TO TRANSMIT SERIAL DATA
                COUNT   res     1       ;VARIABLE USED TO TRANSMIT SERIAL DATA

;*****************************************************************************************
;********************  EQUATES  **********************************************************
;*****************************************************************************************


CS              EQU     H'00'           ;PORTC:  <7> CHIP SELECT
SCLK            EQU     H'03'           ;        <6> SERIAL CLOCK
SI              EQU     H'05'           ;        <5> SERIAL DATA

COMMAND         EQU     H'13'           ;VARIABLE FOR THE COMMAND BYTE
R_VALUE         EQU     D'140'          ;VARIABLE FOR THE RESISTANCE VALUE

;*****************************************************************************************
;********************  PROGRAM ORIGIN  ***************************************************
;*****************************************************************************************


prog1           code

;-----------------------------------------------------------------------------------------
;--------------------  PORT SETTING  -----------------------------------------------------
;-----------------------------------------------------------------------------------------


                BCF     STATUS, RP0     ;MAKE SURE TO BE IN BANK 0
                BCF     STATUS, RP1     ;

                BSF     STATUS, RP0     ;GO TO BANK 1
                MOVLW   0x00
                MOVWF   TRISA           ;MAKE PORTA AN OUTPUT
                MOVWF   TRISC           ;MAKE PORTC AN OUTPUT
                BCF     STATUS, RP0     ;RETURN TO BANK 0

                CLRF    PORTC           ;CLEAR PORTB

;-----------------------------------------------------------------------------------------
;--------------------  PROGRAM ROUTINE  --------------------------------------------------
;-----------------------------------------------------------------------------------------


                BCF     PORTA, CS       ;SELECT THE POT

                MOVLW   COMMAND         ;LOAD THE COMMAND BYTE IN THE ACCUMULATOR
                CALL    TRANSMIT        ;TRANSMIT THE COMMAND BYTE

                MOVLW   R_VALUE         ;LOAD THE RESISTANCE VALUE IN THE ACCUMULATOR
                CALL    TRANSMIT        ;TRANSMIT THE RESISTANCE VALUE

                BSF     PORTA, CS       ;UNSELECT THE POT

                GOTO    FINISH          ;FINISH
```

```
;*******************************************************************************************
;--------------------  TRANSMISSION SUBROUTINE  ------------------------------------------
;*******************************************************************************************


TRANSMIT  MOVWF     OUT                         ;MOVE W TO 'OUT' VARIABLE

                    MOVLW     0X08              ;LOAD A COUNTER TO 'COUNT' THE BIT
                    MOVWF     COUNT             ;TRANSMISSION

L_SHIFT             BTFSC     OUT, 7            ;MONITOR THE 7TH BIT
                    GOTO      HI
                    BCF       PORTC, SI         ;IF LOW: CLEAR SERIAL-IN LINE
                    GOTO      PASS
HI                  BSF       PORTC, SI         ;IF HI: SET SERIAL-IN LINE

PASS                BSF       PORTC, SCLK       ;SET SERIAL CLOCK: HI
                    RLF       OUT, F            ;ROTATE OUT LEFT
                    BCF       PORTC, SCLK       ;SET SERIAL CLOCK: LOW

                    DECFSZ    COUNT, F          ;DECREMENT COUNTER UNTIL ITS ZERO
                    GOTO      L_SHIFT

                    CLRF      PORTC             ;WHEN COUNTER IS ZERO IT'S END OF
                                                ;TRANSMISSION
                    RETURN                      ;RETURN FROM SUBROUTINE


;*******************************************************************************************


FINISH              GOTO      FINISH

                    END


;********************  END OF PROGRAM  ***************************************************
```

APPENDIX E: LINKER SCRIPT FILE

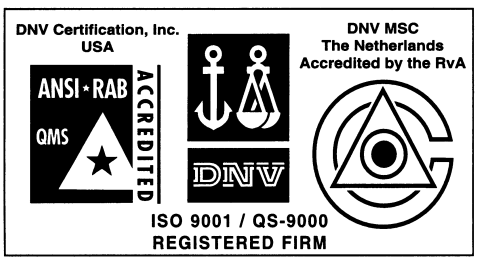
```
// File: 16f876.lkr
// Sample linker command file for 16F876

LIBPATH  .

CODEPAGE   NAME=vectors  START=0x0      END=0x4      PROTECTED
CODEPAGE   NAME=page0    START=0x5      END=0x7FF
CODEPAGE   NAME=page1    START=0x800    END=0xFFF
CODEPAGE   NAME=page2    START=0x1000   END=0x17FF
CODEPAGE   NAME=page3    START=0x1800   END=0x1FFF
CODEPAGE   NAME=.idlocs  START=0x2000   END=0x2003   PROTECTED
CODEPAGE   NAME=.config  START=0x2007   END=0x2007   PROTECTED

DATABANK   NAME=sfr0     START=0x0      END=0x1F     PROTECTED
DATABANK   NAME=sfr1     START=0x80     END=0x9F     PROTECTED
DATABANK   NAME=sfr2     START=0x100    END=0x10F    PROTECTED
DATABANK   NAME=sfr3     START=0x180    END=0x18F    PROTECTED

DATABANK   NAME=gpr0     START=0x20     END=0x6F
DATABANK   NAME=gpr1     START=0xA0     END=0xEF
DATABANK   NAME=gpr2     START=0x110    END=0x16F
DATABANK   NAME=gpr3     START=0x190    END=0x1EF

SHAREBANK  NAME=gprnobnk START=0x70     END=0x7F
SHAREBANK  NAME=gprnobnk START=0xF0     END=0xFF
SHAREBANK  NAME=gprnobnk START=0x170    END=0x17F
SHAREBANK  NAME=gprnobnk START=0x1F0    END=0x1FF

SECTION    NAME=STARTUP  ROM=vectors    // Reset and interrupt vectors
SECTION    NAME=PROG1    ROM=page0      // ROM code space - page0
SECTION    NAME=PROG2    ROM=page1      // ROM code space - page1
SECTION    NAME=PROG3    ROM=page2      // ROM code space - page2
SECTION    NAME=PROG4    ROM=page3      // ROM code space - page3
SECTION    NAME=IDLOCS   ROM=.idlocs    // ID locations
SECTION    NAME=CONFIG   ROM=.config    // Configuration bits location
```

APPENDIX F: SCHEMATIC



NOTES:

NOTES:



Microchip received QS-9000 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona in July 1999. The Company's quality system processes and procedures are QS-9000 compliant for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs and microperipheral products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001 certified.

WORLDWIDE SALES AND SERVICE

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200 Fax: 480-792-7277
Technical Support: 480-792-7627
Web Address: http://www.microchip.com

Rocky Mountain
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-7456

Atlanta
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Austin
Analog Product Sales
8303 MoPac Expressway North
Suite A-201
Austin, TX 78759
Tel: 512-345-2030 Fax: 512-345-6085

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848 Fax: 978-692-3821

Boston
Analog Product Sales
Unit A-8-1 Millbrook Tarry Condominium
97 Lowell Road
Concord, MA 01742
Tel: 978-371-6400 Fax: 978-371-0050

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423 Fax: 972-818-2924

Dayton
Two Prestige Place, Suite 130
Miamisburg, OH 45342
Tel: 937-291-1654 Fax: 937-291-9175

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

Mountain View
Analog Product Sales
1300 Terra Bella Avenue
Mountain View, CA 94043-1836
Tel: 650-968-9241 Fax: 650-967-1590

New York
150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

ASIA/PACIFIC

Australia
Microchip Technology Australia Pty Ltd
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

China - Beijing
Microchip Technology Beijing Office
Unit 915
New China Hong Kong Manhattan Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

China - Shanghai
Microchip Technology Shanghai Office
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

Hong Kong
Microchip Asia Pacific
RM 2101, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

India
Microchip Technology Inc.
India Liaison Office
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

Japan
Microchip Technology Intl. Inc.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

ASIA/PACIFIC (continued)

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea
Tel: 82-2-554-7200 Fax: 82-2-558-5934

Singapore
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-334-8870 Fax: 65-334-8850

Taiwan
Microchip Technology Taiwan
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Denmark
Microchip Technology Denmark ApS
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45 4420 9895 Fax: 45 4420 9910

France
Arizona Microchip Technology SARL
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany
Arizona Microchip Technology GmbH
Gustav-Heinemann Ring 125
D-81739 Munich, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Germany
Analog Product Sales
Lochhamer Strasse 13
D-82152 Martinsried, Germany
Tel: 49-89-895650-0 Fax: 49-89-895650-22

Italy
Arizona Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

United Kingdom
Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5869 Fax: 44-118 921-5820

01/30/01