Keywords: function, ADC, Smart micro, MCU, Microcontroller, Keil, 8051, uvision, analog-to-digital conversion, c compiler, micro, C program, ANSI C, C tutorial, pointers

APPLICATION NOTE 3083

Performing ADC Conversions Using the MAX7651

Mar 26, 2004

Abstract: This article includes the source code and function calls for performing analog-to-digital conversions using the MAX7651 EV kit. It is the first of a 3-part application example for writing, compiling, and downloading a simple program to the MAX7651 EV kit target board.

Also See:

- Configuring Keil µVision IDE for the MAX7651 EV Kit
- Downloading a Program to Flash Using the MAX7651 EV Kit Serial Downloader

The source code in this article demonstrates and is an example of how to perform analog-to-digital conversions using the MAX7651 8051-compatible microcontroller. It includes the key steps of writing, compiling, and downloading code to the MAX7651EVKIT necessary to perform the conversions and read the results. The program uses the on-chip 12-bit integrated ADC contained in the MAX7651.

The example source code was created using Keil DK-51 IDE tools. You should download and install the Keil DK-51 demo software at www.keil.com to easily use the example source code. This Keil software includes the basic library for the MAX7651. Next, you should download the µVision2® project files and source code: AN3083_uVision2_Project_Files.zip.

This example uses the MAX7651EVKIT. This source code can, however, be demonstrated without the MAX7651EVKIT by using the Keil µVision® IDE Simulator. Refer to the MAX7651EVKIT quick start manual for more details on the installation of the Keil software.

Note: The example source code provided by Maxim is public domain. Feel free to copy, modify, or use as required.

Description of C Code

The effective use of this example source code assumes some familiarity with the C programming language. It consists of 3 parts, the main() program, called "ADC Test Function", and 2 function calls: convert_all_channels() and convert_channel(). The main program begins by initializing the program with the register locations of the MAX7651 and including the standard library (stdio.h) function. Next, it configures the MAX7651 to communicate with a PC using a standard serial port.

The main() program calls 2 functions. The convert_all_channels() function performs a conversion of all 8 ADC input channels sequentially. The convert_channel() function allows the user to select the conversion channel and returns a result upon completion of the ADC conversion.

To start a conversion, simply call either convert_all_channels() or convert_channel() functions. The functions perform the conversion(s) by writing the desired channel to the ADCON register in the MAX7651 and then, polling the ADCON register for the end of conversion. After the conversion finishes, it returns the result to the calling function. The main() program then displays the results to the serial port using the printf() function included in stdio.h.

```
/*---------------------------------------------------------------------------
ADC Test Function (main.c)
Copyright: Maxim Integrated Products
Target: MAX7651
Date: Feb 26, 2004
Author: Maxim Integrated Products

Description: This program will convert 8 channels using the MAX7651
and send the results to Serial Port 0
---------------------------------------------------------------------------*/
#include <reg51.h>                                          //include MAX7651
register definitions
#include <stdio.h>                                          //Standard I/O,
Print() function.

#define NUMBER_OF_CHANNELS 8                        //Convert 8 channels

void convert_all_channels(int* buffer);                 //Function declaration
int convert_channel(int adc_ch);                        //Function declaration

void main(void)                                      //Begin Main()
{
int Adc_Results[NUMBER_OF_CHANNELS];    //Array to store conversion result
int i;                                   //for loop counter

        convert_all_channels(Adc_Results);              //Convert all
channels, Store results in Adc_Results

        #ifndef MONITOR51                           //Setup Serial Port if not
using Keil Monitor
SCON  = 0x50;                            //9600 Baud, 8N1, Xon, Xoff
        TMOD |= 0x20;
        TH1   = 0xFA;
        TR1   = 1;
        TI    = 1;
        PCON |= 0x80;
        #endif

        for (i=0; i < NUMBER_OF_CHANNELS; i++)  //Display contents to
Adc_Results
        {
        printf ("CH %d:%x ",i,Adc_Results[i]);  //print the hex
        }

        printf("
 channel 0 %x", convert_channel(0));            //Convert a single channel and
display
        printf("
 channel 1 %x", convert_channel(1));
        printf("
 channel 2 %x", convert_channel(2));
        printf("
 channel 3 %x", convert_channel(3));
        printf("
 channel 4 %x", convert_channel(4));
        printf("
 channel 5 %x", convert_channel(5));
        printf("
 channel 6 %x", convert_channel(6));
        printf("
 channel 7 %x", convert_channel(7));
```

```
        while(1);                                               //End Program, Start
infinite loop since there is no OS
}
 /*---------------------------------------------------------------------------
Function: convert_all_channels
Copyright: Maxim Integrated Products
Target: MAX7651
Date: Feb 26, 2004
Author: Maxim Integrated Products

Usage: The function will return 8 conversion results to an array.

Parameters: p_buffer, pointer to an 8 location array stores the conversion
results

Return: Values are returned to the calling function using the function
parameters

/*---------------------------------------------------------------------------
Setup ADC in the MAX7651
---------------------------------------------------------------------------*/

sfr ADCON  = 0xC5;                                  //Define address of  MAX7651
ADCON
sfr ADDAT1 = 0xC3;                                  //Define address of  MAX7651
ADDAT1 (8MSBs)
sfr ADDAT0 = 0xC2;                                  //Define address of MAX7651
ADDAT0 (4LSBs)

void convert_all_channels(int* buffer);
---------------------------------------------------------------------------*/

#define NUMBER_OF_CHANNELS 8

void convert_all_channels(int* p_buffer)
//pointer Buffer to return
{
        int adc_ch;
        int conv_val;

/*-----------------------------------------------
Convert all ADC channels
-----------------------------------------------*/

        for (adc_ch = 0; adc_ch < NUMBER_OF_CHANNELS; adc_ch++)      //for ADC
channels 1 to 7
{

/*-----------------------------------------------
Start a conversion and wait for it to complete.
-----------------------------------------------*/

        ADCON = adc_ch;                                      //Start conversion
and select channel
while ((ADCON & 0x80) == 0);                        //wait for conversion to
complete
conv_val = (ADDAT0 >> 4) |  (ADDAT1 << 4);                  //Format the data in
12 bit format

          *(p_buffer+adc_ch) = conv_val;                    //Write result back to
calling function
}                                           //End For
}                                                   //End function
convert_all_channels()

 /*---------------------------------------------------------------------------
Function: convert_channel
Copyright: Maxim Integrated Products
Target: MAX7651
```

```
Date: Feb 26, 2004
Author: Maxim Integrated Products

Usage: The function will convert and return the result of a Channel.
Parameters: adc_ch, Select ADC channel to be converted.

Channels 0-7 = single ended channel 0-7.
        Channels 8-11 = differential channel pairs {CH0,1}, {CH2,3}, {CH4,5},
{CH6,7}
        Channel 12 = differential reference measurement {REF+,REF-}

Return: Function returns Integer with the conversion result

Function Declaration: int convert_channel(int adc_ch);
------------------------------------------------------------------------------*/

/*------------------------------------------------------------------------------
Setup ADC in the MAX7651
------------------------------------------------------------------------------*/

sfr ADCON  = 0xC5;                                  //Define address of  MAX7651
ADCON
sfr ADDAT1 = 0xC3;                                  //Define address of  MAX7651
ADDAT1 (8MSBs)
sfr ADDAT0 = 0xC2;                                  //Define address of MAX7651
ADDAT0 (4LSBs)

int convert_channel(int adc_ch)
{
        int conv_val;

if (ADCON <0 || ADCON >12){             //Check for valid channel
return (-2048);                         //Using -FS for the error code
}

        ADCON = adc_ch;                             //Select channel and Start
conversion
while ((ADCON & 0x80) == 0);            //Wait for the conversion to complete

        conv_val = (ADDAT0 >> 4) |  (ADDAT1 << 4);              //Format the
data in 12 bit format

        return (conv_val);                                      //Return
result back to calling function
}                                                       //End function
convert_chan
```

µVision is a registered trademark of ARM, Inc.

Related Parts	
MAX7651	Flash Programmable 12-Bit Integrated Data-Acquisition Systems
MAX7652	Flash Programmable 12-Bit Integrated Data-Acquisition Systems

More Information
For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

---