

Directory: MSP430F2012PlaybackWAV:

- MSP430F2012PlaybackWAV.c
Source file of the eZ430-speech application, based on MSP430F2012 and an external SPI Flash memory.

Directory: MSP430F2012WriteWAV

- MSP430F2012WriteWAV.c
Source file to read, write and erase the external SPI Flash memory. The software is a full duplex UART to SPI converter. The UART is running 38400 baud, 8 data bits, 1 stop bit and not parity, translating all received data to SPI and sending all received SPI data through the UART. To use this software, the protocol of the SPI Flash memory must be understood. This protocol is explained in the data sheet of the SPI memory circuit.

Directory: MSP430F2012WriteWAV\SPI-FlashData

- Several files with the extension '.19'
These files can be used to read, write and erase the external SPI Flash memory. The files are sent from the PC (host) to the SPI Flash memory with the software MSP430F2012WriteWAV.c. A terminal emulation software is necessary for the PC (host). Because no UART handshake is implemented a ~50ms delay for line synchronization should be used. A simple freeware terminal emulator which has been used successfully for SPI Flash programming is occonsole.exe and this can be downloaded from the Internet.
- ReadWavFile.xls
The Macro ReadWavFile inside this Excel-File can read a 8ksps, 8-bit, mono WAV-file and convert it to SPI Flash protocol for usage with the software MSP430F2012WriteWAV.c. The user is responsible for the correct data format, because the Marco doesn't check it. Microsoft SoundRecorder can be used to check and convert WAV file formats.

To start the Macro press 'Alt'-F8 and double-click on 'ReadWavFile '. Now you get a file-dialog and can start conversion with a double-click on the file name of the WAV file. After conversion the result is in the first column of the Excel worksheet. To copy it to a text file, simply select cell A1 with 'Ctrl'-'Home' and then press 'Ctrl'-Shift-'↓' to select the converted data. Now you can copy the data to the Clipboard with 'Ctrl'-C and paste it into your text file with 'Ctrl'-V. Example text files with data generated by this Macro can be found in the directory 'SPI-FlashData'.

Directory: SW-FullDuplexUART38400Baud-16MHzSMCLK

- SW-FullDuplexUART38400Baud-16MHzSMCLK.c
Source file of a full duplex software UART running 38400 baud, 8 data bits, 1 stop bit and not parity for MSP430F2012. This software implements a simple echo and can be used to test the UART communication between the PC (host) and the MSP430F2012. Testing the USART communication is helpful before using the software MSP430F2012WriteWAV.c to write to the external SPI Flash memory.

Directory MSP430FG4618PlaybackWAV:

- MSP430FG4618PlaybackWAV.c
Audio1.h

Audio2.h

Source files for PWM audio playback with MSP430FG4618 based on the board MSP-EXP430FG4618.

- ReadWavFileXCPU.xls

The Macro ReadWavFileXCPU inside this Excel-File can read a 8ksps, 8-bit, mono WAV-file and convert it to C-language source code for usage with the software MSP430FG4618PlaybackWAV.c. The user is responsible for the correct data format, because the Marco doesn't check it. Microsoft SoundRecorder can be used to check and convert WAV file formats.

To start the Macro press 'Alt'-F8 and double-click on 'ReadWavFileXCPU'. Now you get a file-dialog and can start conversion with a double-click on the file name of the WAV file. After conversion the result is in the first column of the Excel worksheet. To copy it to the C-Source file, simply select cell A1 with 'Ctrl'-'Home' and then press 'Ctrl'-Shift-'↓' to select the converted data. Now you can copy the data to the Clipboard with 'Ctrl'-C and paste it into your source file with 'Ctrl'-V.

- MSP430FG4618PlaybackWAV.hex

Intel HEX file with the software and data for the MSP-EXP430FG4618 PWM demo. Can be programmed into the Flash with a programmer.

- MSP430FG4618PlaybackWAV.txt

TI-TEXT file with the software and data for the MSP-EXP430FG4618 PWM demo. Can be programmed into the Flash with a programmer.

Connections for the PWM demo on the board MSP-EXP430FG4618:

- The audio output is on header 4, pin 1. Surrounding pins on header 4 are programmed to a logic low level and can be used as GND.

