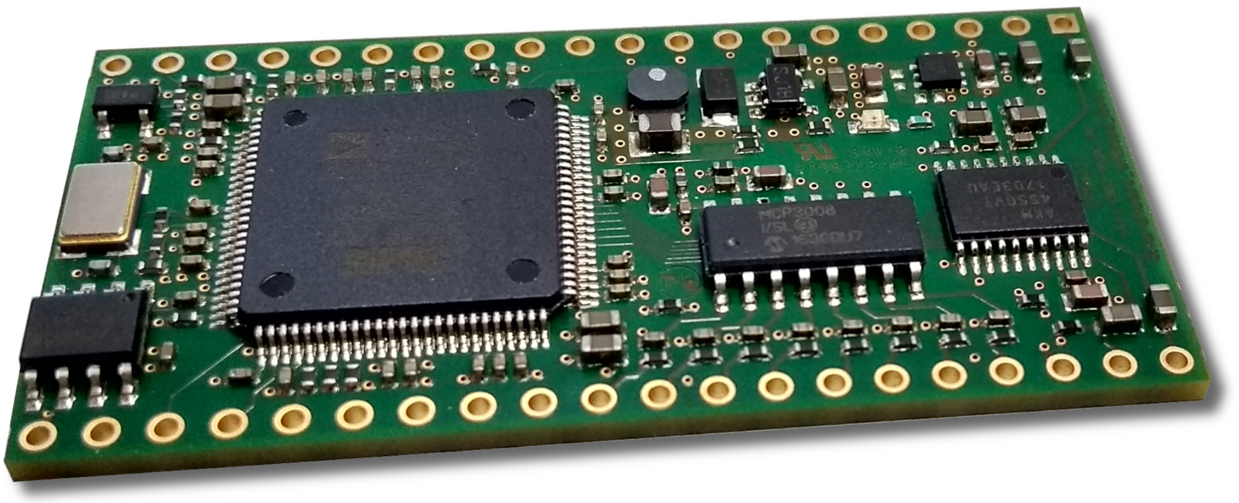
SHARCaudio Board Support Package



Hardware-Setup

To use the following examples in this manual, the SHARCaudio module is plugged into the motherboard.

The board can be powered via USB mini-B jack (JP200) connected to a 5V USB power supply, e.g. PC. Alternatively it is possible to power the board by the DC-jack (JP201).

Note that power supply has to be exactly 5.0 Volt DC.

Software Examples

The software in the board support package contains three projects with the following purpose:

- Audio-Loopback
- Reading ADC values
- Update flash with new program

All examples are written using VisualDSP++ Version 5.1.2.

Each project uses four sub-directories:

- Architecture:	Contains the linker description file.
- Debug:	Contains the output of VisualDSP++ (object files and executable).
- Include:	Contains Header files of the project.
- Sources:	Contains C source files of the project.

Note that a projects bootable from flash use the project type “Loader file” in the project settings. Projects of this type use the kernel file “479_spi.dxe”, which is located in the installation directory of VisualDSP++.

Audio-Loopback

The audio-loopback project collects input samples in an interrupt service routine. Samples are copied into the output buffer in the main routine and transmitted to the codec in another interrupt service routine. The copy routine in the main routine is the location to place useful routines for signal processing. The example project is located in directory “SharcAudioLoopback”.

The main routine in “main.c” first initializes the DSP with the routine “InitDSP()” located in “util.c”.

The routine “InitSRU()”, located in “configdai.c” initializes the DAI and DPI pins of the DSP.

The AKM AK4556 codec is initialized with the routine “InitAK4556Dma()” in “AK4556.c”. This routine initializes chaining DMA. The buffer “iAudioDmaInSamples[]” and “iAudioDmaOutSamples[]” are used as ping-pong buffers.

The routine “InitAK4556Mclk()” initializes a precision clock generator (PCG) to generate a master clock of 24.576 MHz for the codec. This masterclock is derived from the peripheral clock of the DSP running at 98.304 MHz.

With the current setting of codec setup pins (CKS0...CKS3) this master clock results in a sampling frequency of 96 kHz. The sampling rate is adjustable by CKSx pins or by changing the masterclock.

The interrupt service routine “Sport1RxIsr()” is called with each DMA receive interrupt of SPORT1. It copies samples from the recently filled by DMA buffer into the array “iAudioInSamples[]”. It additionally sets the flag “iSport1IsrOccured”.

The main loop idles until flag “iSport1IsrOccured” is set. When the flag is set, samples are copied from input buffer into the array “iAudioOutSamples[]”.

Samples from the output buffer are copied into the DMA buffer in the interrupt service routine “Sport3TxIsr()”. This interrupt is called with the DMA channel of SPORT3.

ADC Value Read

With the example project, located in directory “SharcAudioAdcGet”, it is possible to read the ADC values of the eight channel ADC MCP3008 from Microchip.

Similar to the audio loopback project, setup is done at the beginning of the main loop. ADC values are read using SPI-interface (channel A) of the DSP.

The forever loop in the main routine reads all ADC channels and stores the values in the array “iAdcChannels[]”. The values can be verified in the expression window of VisualDSP++.

Update Flash Program

The project in directory “SharcAudioUpdate” is used to update the serial flash of the SHARCaudio module with a new program bootable from flash

After intialization of DSP, SRU, SPI and the serial flash in the main routine, the routine “SoftwareDspUpdate()” is called to erase and to reprogram the serial flash. This routine is located in “update.c”.

The update routine opens a loader file (*.ldr) created by a VisualDSP++ project. In this example, it is the audio loopback routine. The order of the bits in a byte of the loader file has to be changed by the routine “BitRoll()”.

Note that execution of this project need about 30sec., caused by erasing time and slow file transfers in VisualDSP++.

After programming the serial flash, the module needs a power cycle to start booting the program.