– Digital Microelectronics — SoundLoc –
Localication of Soundsources by cross-correlating three ΣΔ-microphon signals

Stefan Kull, Roy Seitz, Marco Zollinger

December 2, 2016

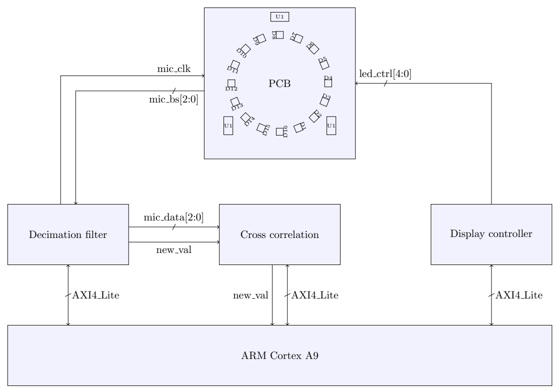

Figure 1: Top Level Block Diagramm

1 Overview

Sound travels at a speed of approximately $c \approx 340\,\mathrm{m\,s^{-1}}$. Using multiple microphones at well known locations allows calculating the direction from which the sound came. Three microphones are placed in an equilateral triangle of 42 mm of side length. A circle of 16 LEDs is placed on the same PCB to indicate the direction. Because only the direction in two dimensions is of interest, three microphones are sufficient. The microphone operate at approximately 3.2 MHz using ΣΔ-modulation. They output their bitstream directly without any digital filtering or decimation. Figure 1 shows a block diagram of the whole system.

The bitstream is filtered by a IP block, configurable by software via AXI4 Lite interface. This block contains a CIC filter of configurable order and decimation rate with a differential Delay of $M = 1$. An additional IIR filter can be enabled to remove the DC component of the signals. See Section 2 for further information.

From the decimated data, the cross correlation is calculated to find the signal delay between the microphones due to finite sound speed. This is done by another IP block, also configurable over AXI4 Lite. See section 3 for further information.

These delays are read by the CPU, which calculates the direction of the sound source. It basically consists of a base transform from microphone to Cartesian coordinates and calculating the inverse tangent of the coordinates. This is described in section 4.

The Angle is mapped to one of the 16 LED and then fed to another IP block that displays the direction. It consists of a simple 16 bit shift register that illuminates the corresponding LED. For this, see section 5.

2 Decimation Filter

The decimation filter generates the clock needed by the microphones and processes the received bitstreams. It is widely configurable by generic parameters at compile time as well as by software. It outputs filtered and decimated signed values and a signals to indicate when new values are available. Low pass filtering and decimation is done by a CIC filter of up to third order and programmable decimation rate. An additional high pass IIR filter with configurable pole location can be added to block potential DC components due to offsets. For more information, see the IP documentation.

3 Cross Correlation

This block calculates the cross correlation between the three microphone signals. To do this efficiently and in real time, the correlation is calculated iteratively, using fast block RAM and DSP slices. For more information, see the IP dokumentation.

4 Software

One ARM Cortex A9 core is used in this project. At first it configures the Filter block by setting order, decimation, IIR pole and post division factor. The filter delivers its values directly to the cross correlation. This informs the Cortex A9 each time recalculation of the cross correlation is finished. The Cortex A9 then reads the data and searches for the maximum. The cross correlation is maximal when the signals are equal. Thus the Tau at which the maximal value occurs is proportional to the signal delay caused by finite speed of sound.

If the distance between the microphones and the sound source is large, the incoming wave can be modeled as a plane wave. The delays are then proportional to the inner product between the wave vector and the distance between the correlated microphone. This is expressed by equation 1.

Since the microphones are arranged in a equilateral triangle, the two taus are not orthogonal. The vector directing to the source location can be expressed as per equation 2. To calculate the signal delay in cartesian coordinates, Bx needs to be left multiplied with B^{-1}.

$$Tau_{0n} \propto \overrightarrow{k} \cdot \overrightarrow{M_n M_0} \quad n \in \{1, 2\} \tag{1}$$

$$\begin{aligned} v &= E\tau = Bx && (2) \\ &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} Tau_{01} \\ Tau_{02} \end{pmatrix} = \begin{pmatrix} 1 & \cos(\frac{\pi}{3}) \\ 0 & \sin(\frac{\pi}{3}) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} && (3) \end{aligned}$$

$$\begin{aligned} x &= B^{-1}\tau && (4) \\ \begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} 1 & -\frac{1}{\sqrt{2}} \\ 0 & \sqrt{2} \end{pmatrix} \begin{pmatrix} Tau_{01} \\ Tau_{02} \end{pmatrix} && (5) \end{aligned}$$

From the Cartesian coordinates x, y, the four quadrant invers tangent can easily be calculated using the C-function *atan2(y,x)*. The resulting angle is finally mapped to the corresponding led and displayed.

5 LED Display

The 16 LEDs are controlled via 2 8-bit shift registers. The LED nearest to the calculated direction is switched on every cycle. Due to remaining noise and fast processing speed, interpolation seamingly happens by illuminating neighbouring LEDs alternatively.