Copyright © 2017
PulseRain Technology, LLC.

10555 Scripps Trl, San Diego, CA 92131
858-877-3485 858-408-9550
http://www.pulserain.com

Doc# TRM-0922-01002, Rev 1.0.0

PulseRain M10 – DTMF

Technical Reference Manual

Aug, 2017

This page is intentionally left blank.

Table of Contents

References

1. PulseRain M10 – Voice CODEC Technical Reference Manual, DOC# TRM-0922-01001, Rev 1.0.1

1 Introduction

1.1 CODEC

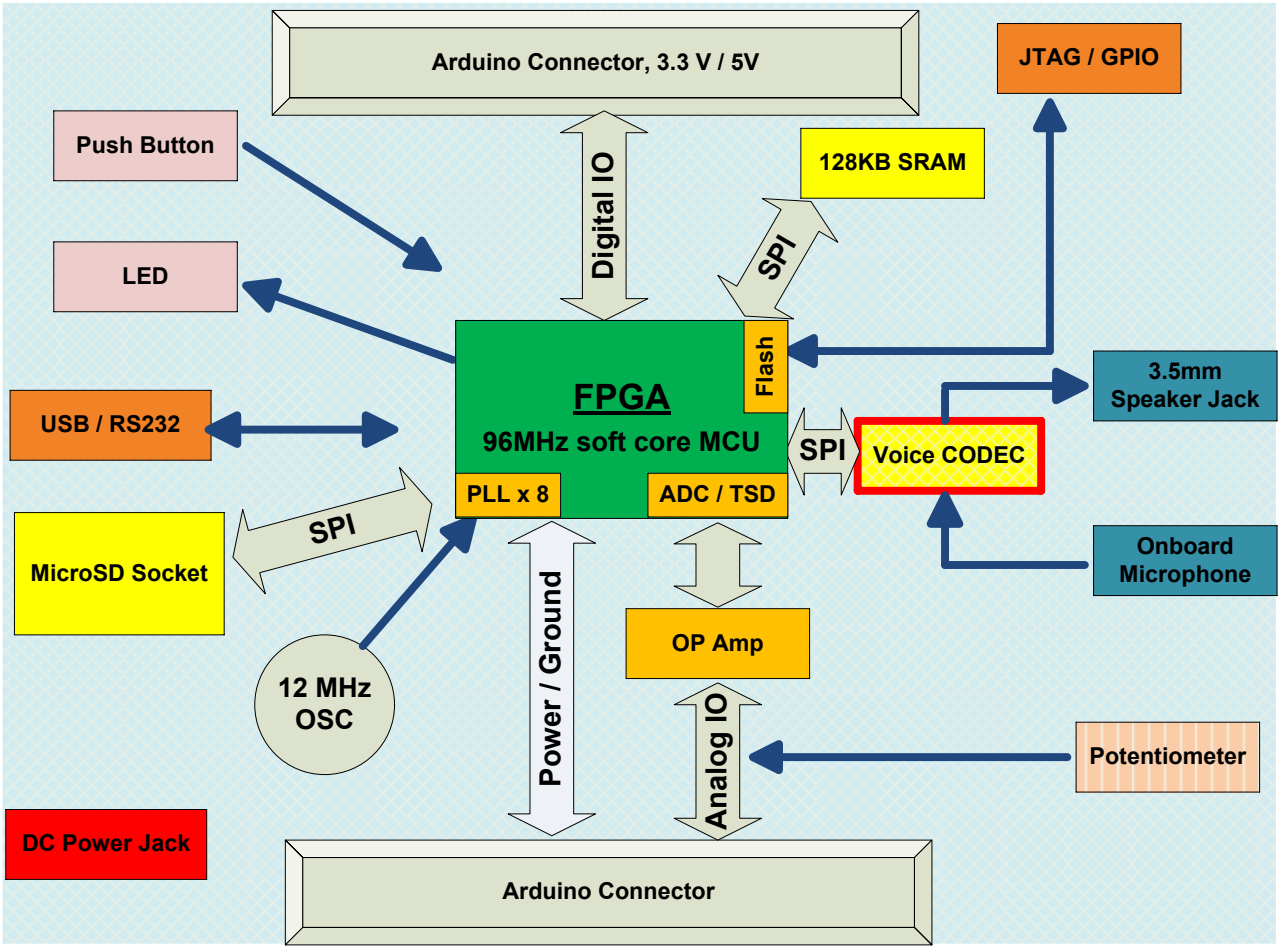

Figure 1-1 The Close View of M10

As shown Figure 1-1, the M10 board takes a distinctive technical approach by embedding an open source soft MCU core (96MHz) into an Intel MAX10 FPGA, while offering an Arduino compatible software interface and form factors. Among all the onboard peripherals, there is a voice CODEC to interface with the speaker and microphone. DTMF (Dual Tone Multi Frequency) tones can be received through the onboard microphone to the CODEC. And PulseRain Technology has designed an open source controller to interact with the CODEC from the FPGA side, whose technical detailed can be found in Ref [1]. On top of that, PulseRain Technology has also provided an additional library to help the decoding of DTMF tones, for which this document serves as a TRM (Technical Reference Manual).

1.2 DTMF

DTMF stands for Dual Tone Multi Frequency. It is a way of signaling key press by assigning two distinctive frequencies to each key, as shown below in Table 1-1. The bold numbers with shades are all frequencies in Hz, while the symbols without shades are the correspondent keys.

	1209	**1336**	**1477**	**1633**
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

Table 1-1 DTMF Lookup Table

As Table 1-1 shows, the 2nd harmonic of those frequencies are not used by any of the keys, thus those 2nd harmonic frequencies can be utilized as a way to prevent false detection.

There are many ways to generate DTMF tones. The standard way is, of course, to dial in the numbers and letters at one end of the phone line, and hear it on the other end. Alternatively, DTMF can be generated by using Apps on a smartphone. For example, on iOS, the following Apps can be used:

- DTMF Pad, by IEIRISOFTWARE LAB
- Tone Dial, by River Rock Logic

2 Hardware

As mentioned early, the M10DTMF library, by itself, is only an additional software library that relies on the M10CODE library and the CODEC hardware to decode DTMF tones. And the details of the CODEC hardware can be found in Ref [1].

3 Software

3.1 Goertzel Algorithm

As mentioned in Section 1.2, the DTMF signal is nothing but a combination of two tones. Although FFT can always be used to detect those tones, the overhead that FTT carries might become a huge burden to MCUs. Instead, the M10DTMF library uses another approach called Goertzel Algorithm. Basically, the Goertzel algorithm is an IIR filter who has a pole on the frequency of interest. And the filtering can be done iteratively, like the following:

$$s(n) = x(n) + 2\cos(2 * \pi * f) * s(n-1) - s(n-2)$$

For M10DTMF library, the n is from 0 to 255. With a sample rate of 8kHz, it covers 256 * 1 / 8000 = 32 millisecond of duration. After 256 iterations, the power term is calculated as

$$s(n-1) * s(n-1) + s(n-2) * s(n-2) - 2 * \cos(2 * \pi * f) * s(n-1) * s(n-2)$$

In M10DTMF, the Goertzel algorithm is performed for 16 different frequencies, which covers the 8 frequencies in Table 1-1 and their 2nd harmonics.

3.2 Flow Chart

Prior to the Goertzel algorithm, a few other measures are taken to reduce the chance of misdetection, as illustrated in Figure 3-1. At the end, the frequency pair with the maximum power term will be compared against a threshold to determine the existence of key press. The 2nd harmonics also play a role in this part.

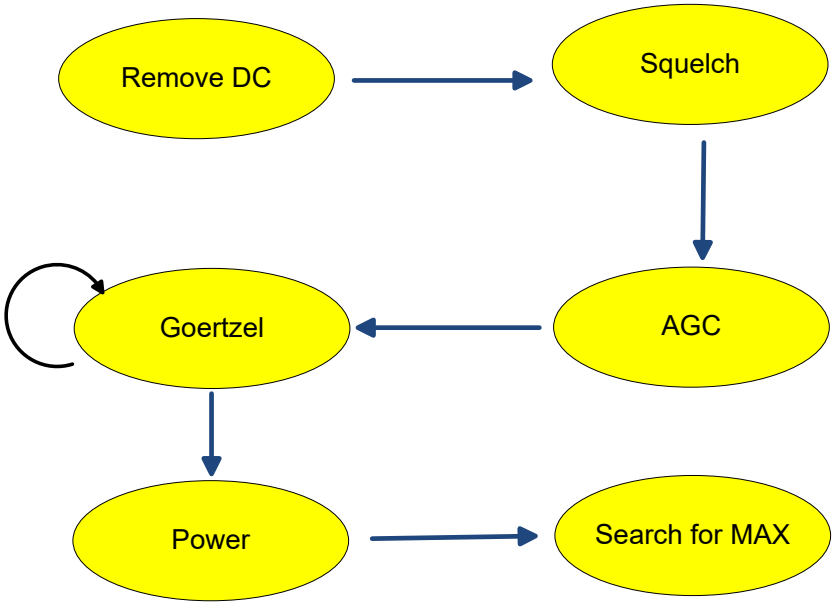

Figure 3-1 DTMF Detection

3.3 ISR

The M10DTMF library is built on top of M10CODEC library, and it attaches an ISR to the CODEC interrupt (IRQ #6) to collect audio samples. In other words, when M10DTMF is being used, the user can no longer hook up his own ISR to CODEC. However, the user can still play samples through CODEC without using ISR in this case (by calling CODEC.sampleWrite()). Examples of playing audio samples without ISR can be found in the following examples:

- The wav_record_playback example in M10CODEC library
- The DTMF Home Automation demo in

  https://github.com/PulseRain/DTMF_HOME_AUTOMATION

3.4 Work Flow

To use the M10DTMF library, do the following steps:

1. Call DTMF.begin() to initialize the library. ISR is also attached in this step.
2. Call DTMF.decode() to get the keys detected. A non-negative return value carries the key index in Table 3-1, and a negative return value means no key press is detected.

Return Value	Keys (Digit, Letter and Symbol)
0 (0x0)	Digit 0
1 (0x1)	Digit 1
2 (0x2)	Digit 2
3 (0x3)	Digit 3
4 (0x4)	Digit 4
5 (0x5)	Digit 5
6 (0x6)	Digit 6
7 (0x7)	Digit 7
8 (0x8)	Digit 8
9 (0x9)	Digit 9
10 (0xA)	Letter A
11 (0xB)	Letter B
12 (0xC)	Letter C
13 (0xD)	Letter D
14 (0xE)	The asterisk ('*')
15 (0xF)	The pound sign ('#')

Table 3-1 Return Value vs Keys

3.5 Arduino Library

3.5.1 APIs

- *void begin()*

  Call this function to initialize the library and setup the ISR for CODEC.

- *int8_t decode()*

  Call this function to get the latest key press detected. This is a non-block function, and it will return the key index shown in Table 3-1 if a key press is detected. Otherwise, it will return a negative value as the following:

  -1: 32 milliseconds of samples are collected, but no key press is found

  -2: There are not enough samples collected.

3.5.2 Examples

To further facilitate the software development, the following examples can be referenced:

- *dtmf_detector*
  This example comes with M10DTMF library. It will output the detected DTMF keys to the Serial port in 115200 bps. To use this example, run it in Arduino IDE and open the Serial Monitor and set it to 115200 bps. And use a smartphone to generate the key press with DTMF tones. On iOS, the following Apps can be used:
  - DTMF Pad, by IEIRISOFTWARE LAB
  - Tone Dial, by River Rock Logic

  Similar Apps can also be found for Android.

- *DTMF home automation*
  https://github.com/PulseRain/DTMF_HOME_AUTOMATION

  It is a big demo that can be used along with an answer machine. When the answer machine picks up the call, the caller can key in DTMF passcode and commands to start home automation. In this demo, it simply reads out temperature measured by onboard TSD.

  And this demo involves almost all the onboard peripherals:
  - The onboard microphone and speaker audio jack through Si3000 CODEC
  - The onboard SRAM (M23XX1024)
  - The SD card (audio files are saved on a microSD card)
  - PWM (If phone is picked up by using PWM, the Sparkfun Ardumoto - Motor Driver Shield is needed.

3.5.3 Scripts

To assist the development, the following scripts are also provided in the "extras" folder of M10DTMF library:

- dtmf_coef.py
  This script generates the fixed-point coefficient for 2*cos(w) used by in Goertzel Algorithm.