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

Doc# TRM-0922-01003, Rev 1.0.0

PulseRain M10 – ADC

Technical Reference Manual

Sep, 2017

This page is intentionally left blank.

Table of Contents

References

1. Intel MAX 10 Analog to Digital Converter User Guide, UG-M10ADC, 2017.07.06
2. The schematic of PulseRain M10 board, Doc# SH-0922-0039, Rev 1.0, 02/2017

1 Introduction

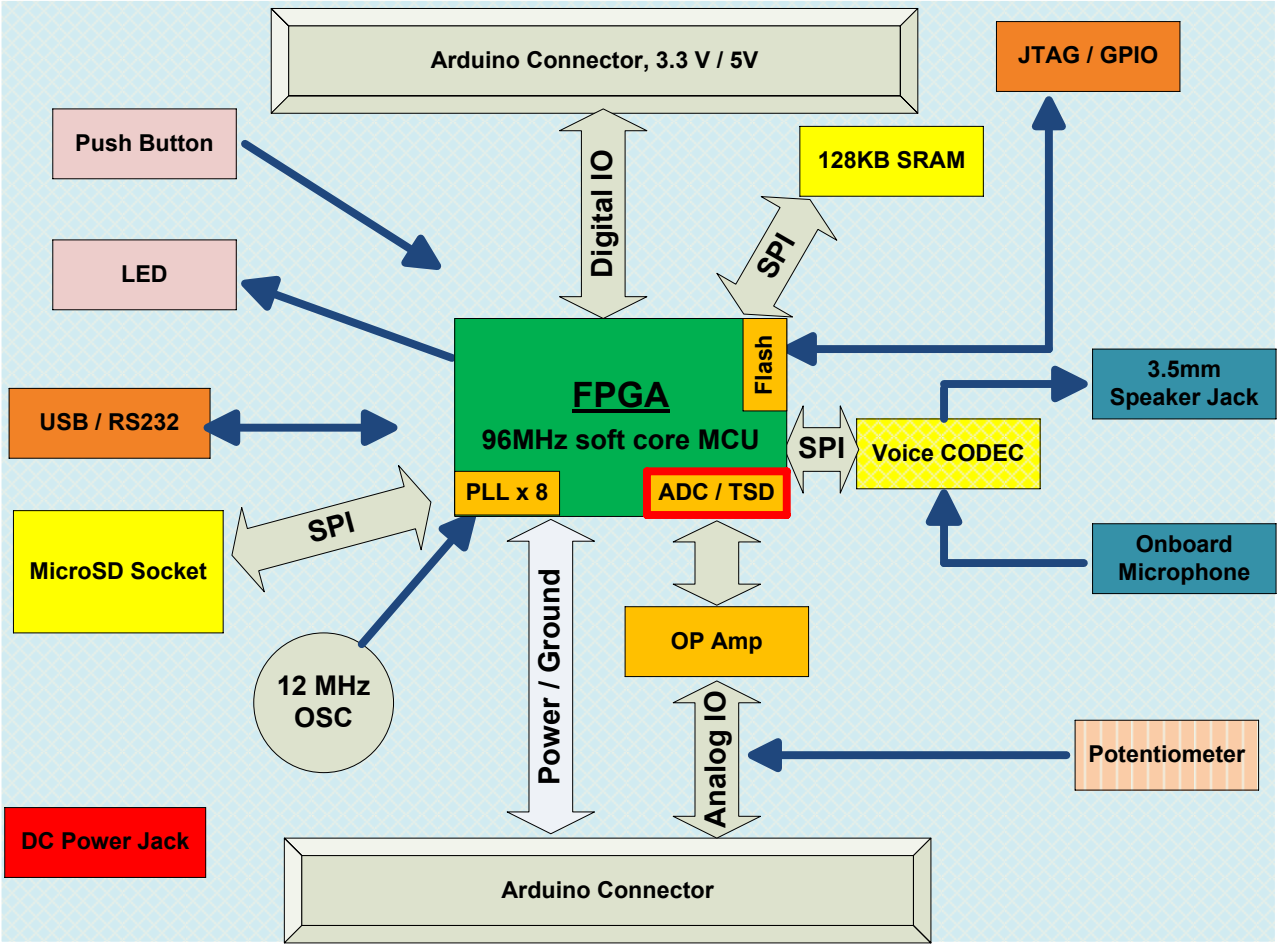

Figure 1-1 The Close View of M10

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. For the particular MAX10 FPGA that is chosen (10M08SAE144C8G), it carries an on-chip A/D Converter (Ref [1]) that supports up to 8 channels with 12-bit precision. And 6 out of the 8 channels are being used by the M10 board for Arduino compatibility.

The on-chip ADC also contains a TSD (Temperature Sensor Diode) as one additional ADC channel, for which the M10ADC library offers APIs to convert the input into units of Celsius.

2 Hardware

2.1 OpAmp and Potentiometer

The 10M08SAE144C8G contains one ADC block that can measure up to 3.3V. To scale it up to 5V, external OpAmps have been placed in front of the ADC input pins. And for ADC channel 1, a potentiometer is also installed to make the input gain adjustable, as illustrated in Figure 2-1.

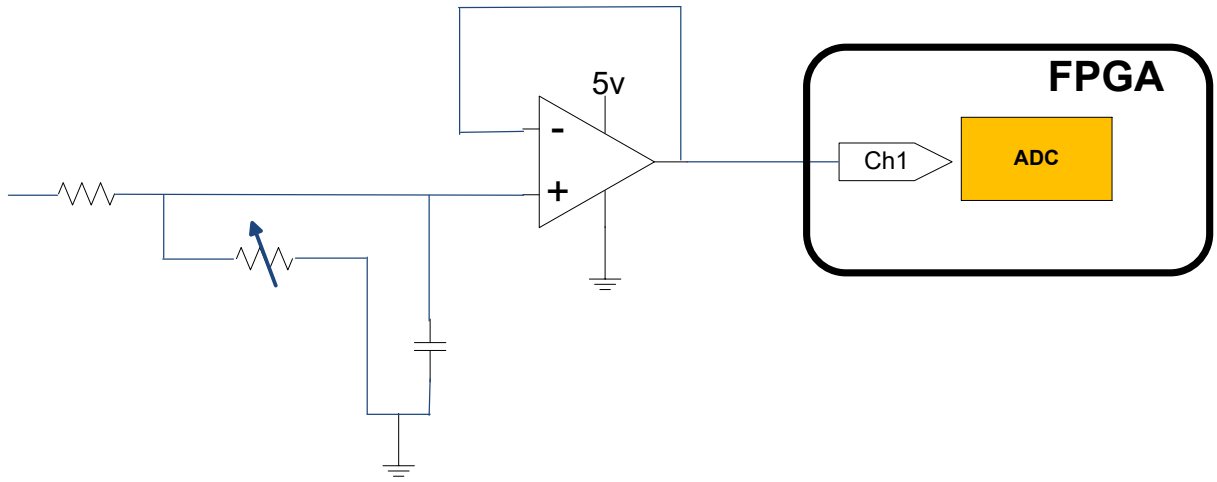

Figure 2-1 ADC Channel #1 with Adjustable Gain

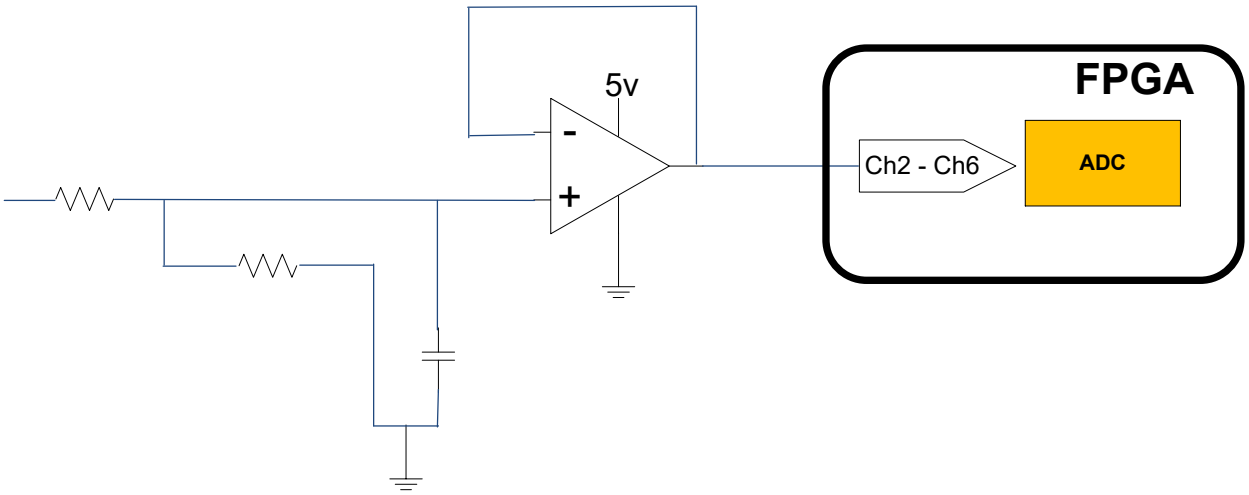

Figure 2-2 ADC Channel #2 - #6 with Fixed Gain

2.2 Pin Assignment

The pin assignment of those 6 ADC channels are as following:

ADC Channel Index	FPGA Pin Assignment (10M08SAE144C8G)	Arduino Uno Channel Index
1	6	A0
2	7	A1
3	8	A2
4	10	A3
5	11	A4
6	12	A5

Table 2-1 FPGA Pin Assignment

From the standpoint of FPGA, these pins are dual function and can be configured to be digital I/O if necessary. However, due to the presence of OpAmp, these pins are used for Analog Input only on the M10 board.

2.3 Sample Rate

The on-chip ADC is a SAR (successive approximation register) type converter that can run at a sample rate up to 1MHz. On the M10 board, its default sample rate is set to be 50kHz with a 2MHz clock input.

2.4 Controller

Intel/Altera has provided a Modular ADC IP core to interact with the on-chip ADC. Accordingly, PulseRain Technology, has offered a FPGA controller (a Wishbone wrapper around the core indeed), as illustrated in Figure 2-3. And interrupt is also supported by the Wishbone wrapper.

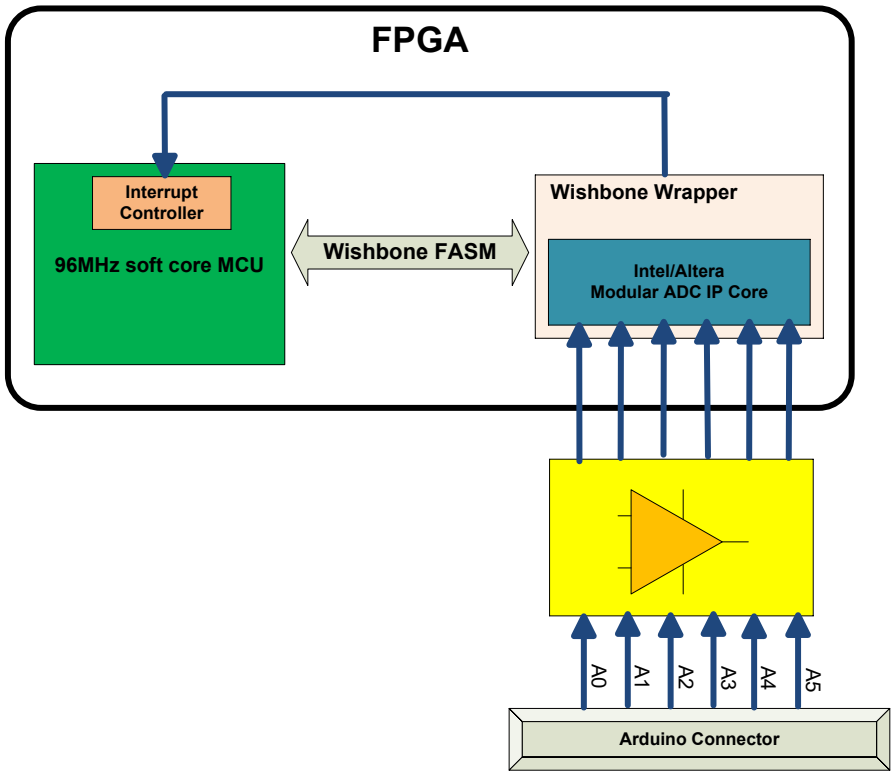

Figure 2-3 The FPGA Controller for ADC

2.5 Repository

The latest ADC IP core from Intel/Altera can be found in the "cores" folder of the FP51_MCU repository:

https://github.com/PulseRain/PulseRain_FP51_MCU/tree/master/cores/ADC

And the RTL code for ADC controller (Wishbone wrapper) is part of PulseRain Technology's RTL library, which can be found on GitHub:

https://github.com/PulseRain/PulseRain_rtl_lib

3 Software

3.1 Register Definition

The controller shown in Figure 2-3 contains all the registers to control the on-chip ADC. In a nutshell, the registers are defined as following:

- DATA_HIGH (8 bit) and DATA_LOW (8 bit)
  These two read-only registers contain the higher 4 bits and lower 8 bits of the ADC output on the active channel. To get the full 12 bits, read the DATA_HIGH register first, followed by another read on DATA_LOW.

- CSR (Control Status Register)
  The bits for CSR are defined in Table 3-1:

Bits	R/W	Default	Name	Description
0	WO	0	command_valid	Write 1 to enable the active channel designated in bit [7 : 3]
1	RO	0	adc_data_ready	When new data is latched into the DATA_HIGH and DATA_LOW register, this flag will be set high. And this flag can be cleared by a read on CSR
7:3	RO	0	command_channel	The index for the active channel. And the valid values are: 0 – 5: corresponds to Arduino UNO A0-A5 16: Channel for TSD (Temperature Sensor Diode) 30: Channel for Recalibration

Table 3-1 Bit Map for CSR (Control Status Register)

3.2 Address Map

The registers defined in Section 3.1 are mapped into MCU's address space, as shown in Table 3-2.

Address	Register Name
0xF5	ADC_DATA_HIGH
0xF6	ADC_DATA_LOW
0xF7	ADC_CSR

Table 3-2 Address Definition

3.3 Work Flow

To get the A/D converted value with an ISR, do the following:

1. Call ADC.begin() to initialize the library and calibrate the ADC.
2. Attach the user defined ISR to ADC_INT_INDEX (IRQ #5).
3. read the ADC_DATA_HIGH and ADC_DATA_LOW in the ISR to get the result

To get the A/D converted value without an ISR, do the following:

1. Call ADC.begin() to initialize the library and calibrate the ADC.
2. Call the analogRead() or ADC.read() to read result.

To get the temperature reading from on-chip TSD, do the following:

1. Call ADC.begin() to initialize the library, followed immediately by another call to ADC.getCelsius().
2. Call the ADC.getCelsius() periodically to read the temperature in Celsius degree.

To recalibrate, simply call ADC.begin()

3.4 Arduino Library

3.4.1 APIs

- *void begin()*

  Call this function to initialize the library and recalibrate the ADC.

- *uint16_t read (uint8_t index)*

  Call this function to get the reading from the designated channel.

- *int16_t getCelsius ()*

  Call this function to get the temperature reading in Celsius degree.

- *void end ()*
  Call this function to disable the ADC controller.

3.4.2 Examples

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

- *ADC_ISR*
  This example comes with M10ADC library. It will output the A/D converted value on pin A0 to Serial port. Run this example in Arduino IDE, and observe the converted value on Serial Monitor at 115200bps.

  To test this example, it is recommended to put a wire to connect the IO REF pin to A0, and set the jumper to 5V on JP1. Then use a screw driver to adjust the potentiometer and watch the value change on the Serial Monitor. (As mentioned early, the gain on A0 channel can be adjusted by the potentiometer).

- *TSD*

  This example also comes with M10ADC library. Run it in Arduino IDE, and this example will output the temperature reading (Celsius with a time stamp) to the Serial Port at 115200bps.