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

Doc# TRM-0922-01008, Rev 1.0.0

PulseRain M10 – JTAG

Technical Reference Manual

Sep, 2017

This page is intentionally left blank.

Table of Contents

References

1. Embedded Peripheral IP User Guide, UG-01085, Altera Corporation, 05/08/2017
2. The schematic of PulseRain M10 board, Doc# SH-0922-0039, Rev 1.0, 02/2017
3. Intel FPGA Download Cables, https://www.altera.com/products/boards_and_kits/download-cables.html

1 Introduction

1.1 The Whole Picture

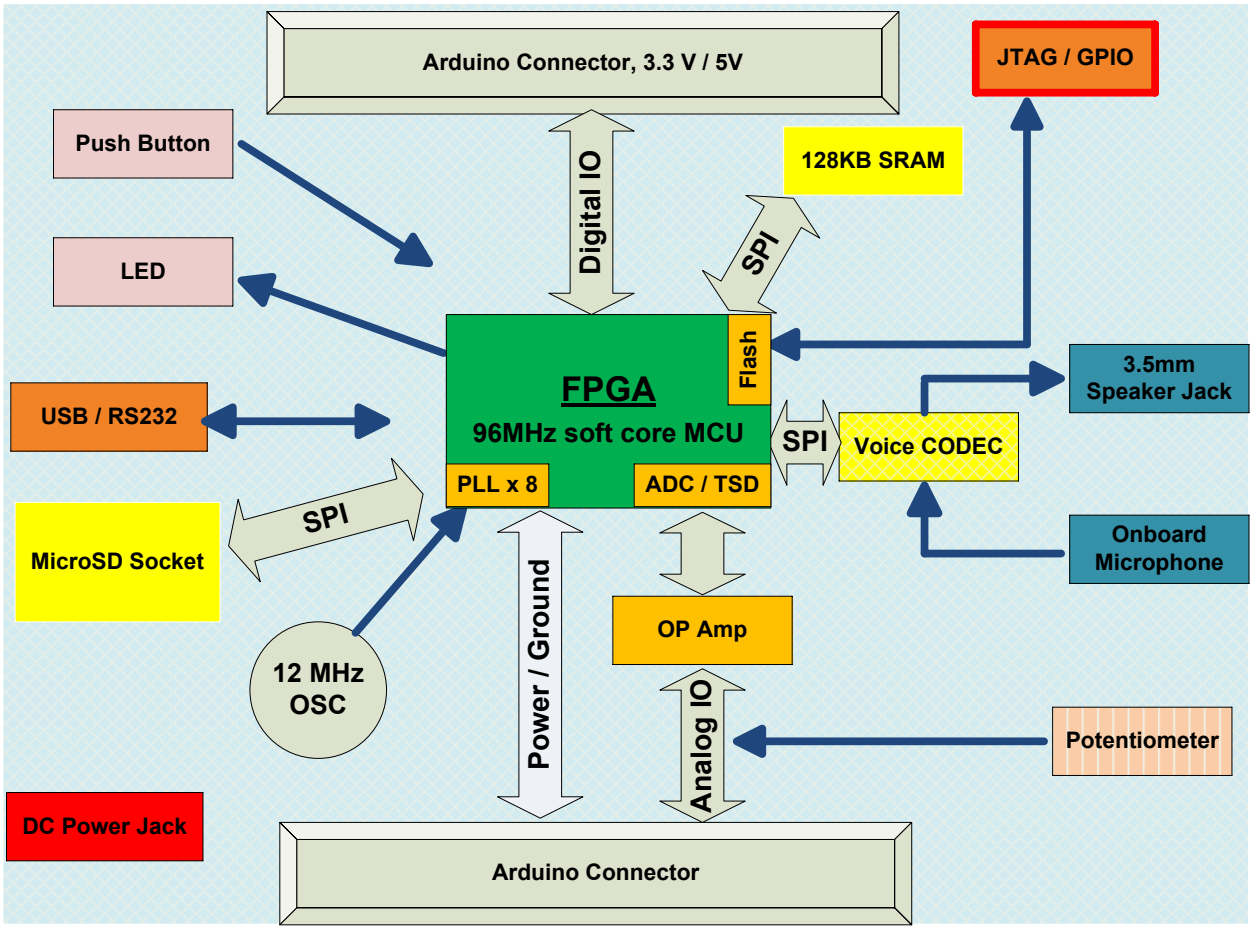

Figure 1-1 The Whole Picture

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 connectors, there is a standard JTAG header that can be used to program the FPGA device. In addition, this JTAG connection can also be used as a UART port with the help of Altera JTAG UART IP core (Ref [1]). And this document serves as a technical reference manual for the JTAG port.

1.2 The JTAG Connector

In order to use JTAG, the user needs to hook up a JTAG cable (Ref [3]) to the connector, as illustrated in Figure 1-2. The JTAG connector is a 10-pin header, whose pin map can be found in Table 1-1.

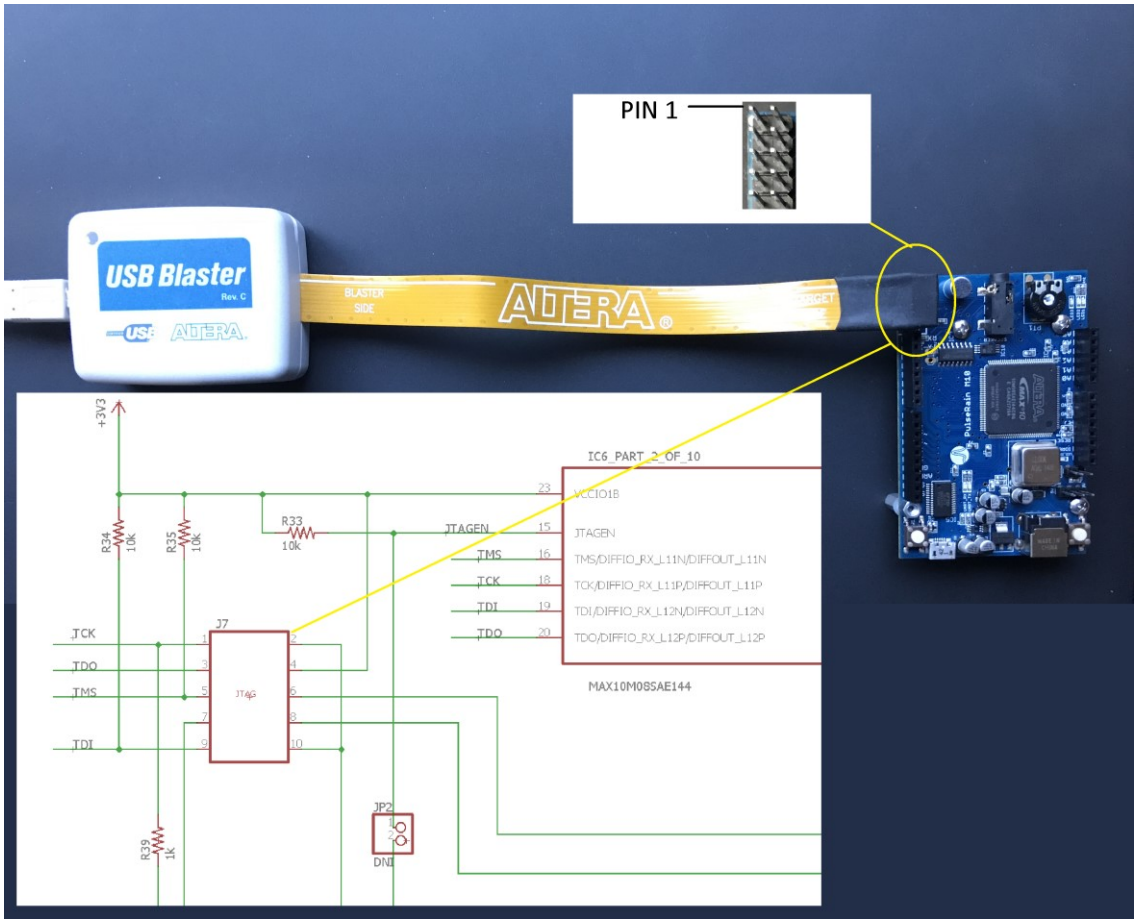

Figure 1-2 The JTAG Connector and JTAG Cable

PIN Index	Name	Description
1	TCK	JTAG Test Clock
2	GND	Ground
3	TDO	JTAG Test Data Out
4	POWER	3.3V
5	TMS	JTAG Test Mode Select
6	GPIO	General Purpose IO
7	GPIO	
8	GPIO	
9	TDI	JTAG Test Data In
10	GND	Ground

Table 1-1 Pin Map for JTAG Connector

Among the 10 pins on the JTAG header, 3 of them can be used for GPIO. If more GPIOs are needed, the user can solder a 2-pin header (2.54mm pitch) to JP2 (not populated by default). The JP2 is connected to the FPGA's JTAGEN pin (Ref [2]). Putting a jumper on it will disable the JTAG signals and turn them into GPIOs.

2 Hardware

2.1 Program the FPGA with JTAG cable

Image can be downloaded to the FPGA through JTAG cable. To do that, the user needs the following:

1. A JTAG Cable (Ref [3]). The rest of this document assumes Altera USB Blaster is used.
2. Quartus Prime Programmer Software, which can be downloaded from Altera Website.

Install the cable with its correspondent driver, and connect it to the M10 board as illustrated in Figure 1-2. Power on the board and launch the programmer software. Click AutoDetect to detect the FPGA, and select the 10M08SA device, as illustrated in Figure 2-1. If everything goes right, the user should see the JTAG chain with 10M08SA device displayed, as shown in Figure 2-2. Right click the row under "File", and chose "Change File" to select the image. The image file can be either a .sof file (cannot survive power cycle) or a .pof file (program the on-chip flash, and it can survive power cycle.). Click "start" to kick off the programming process, as shown in Figure 2-3.

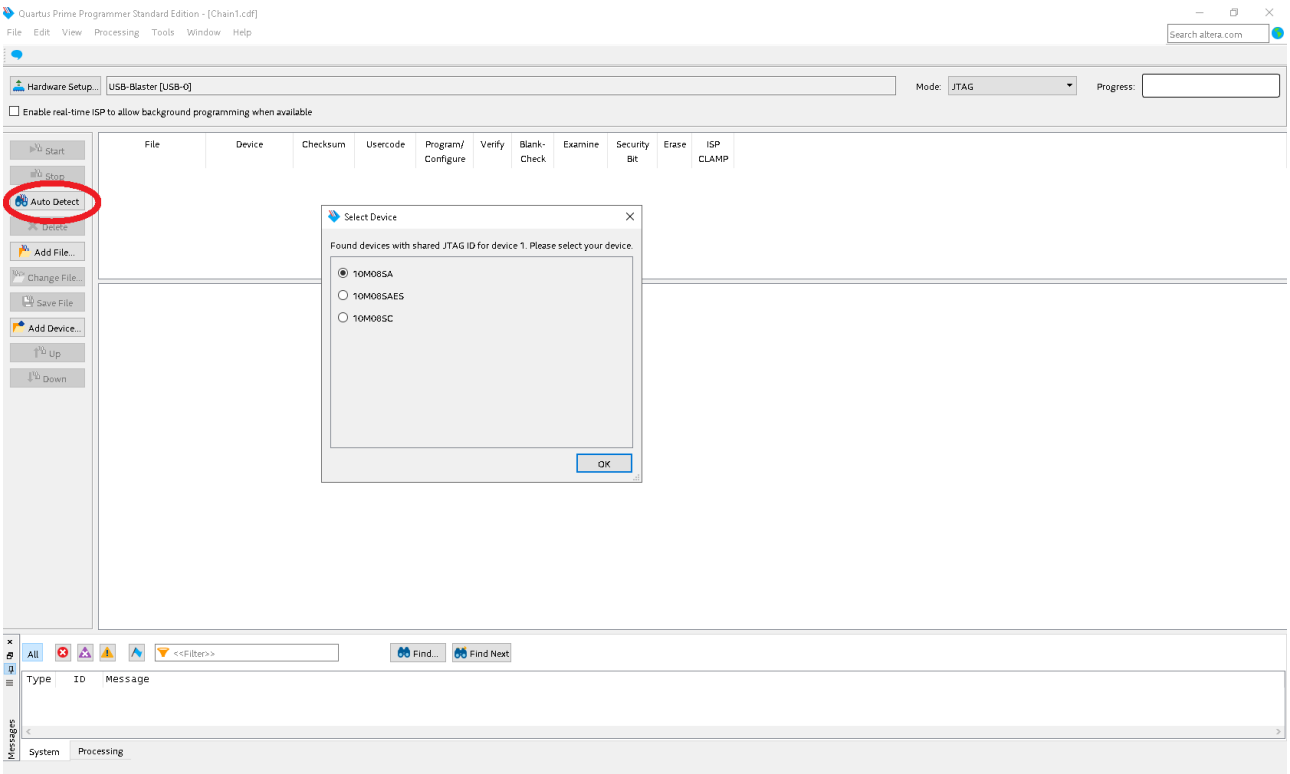

Figure 2-1 Detect the FPGA device with JTAG

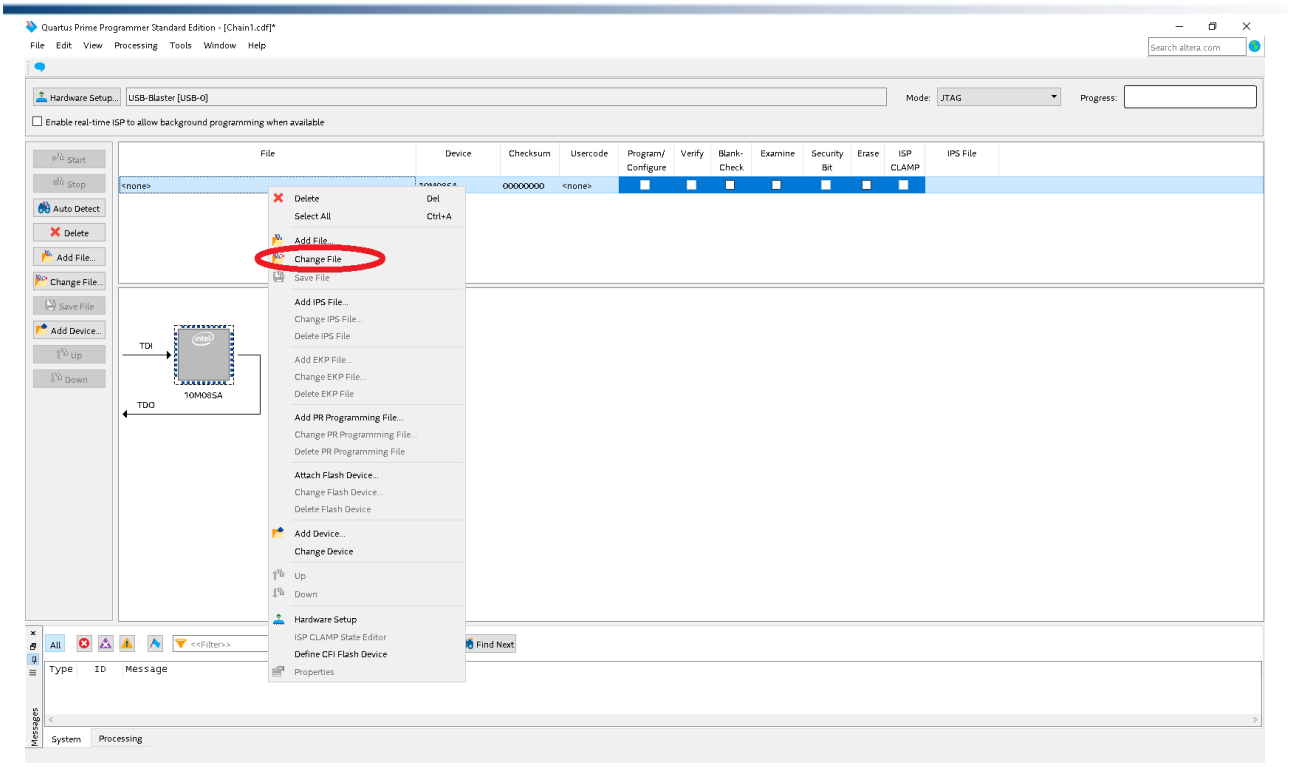

Figure 2-2 Choose Program Image

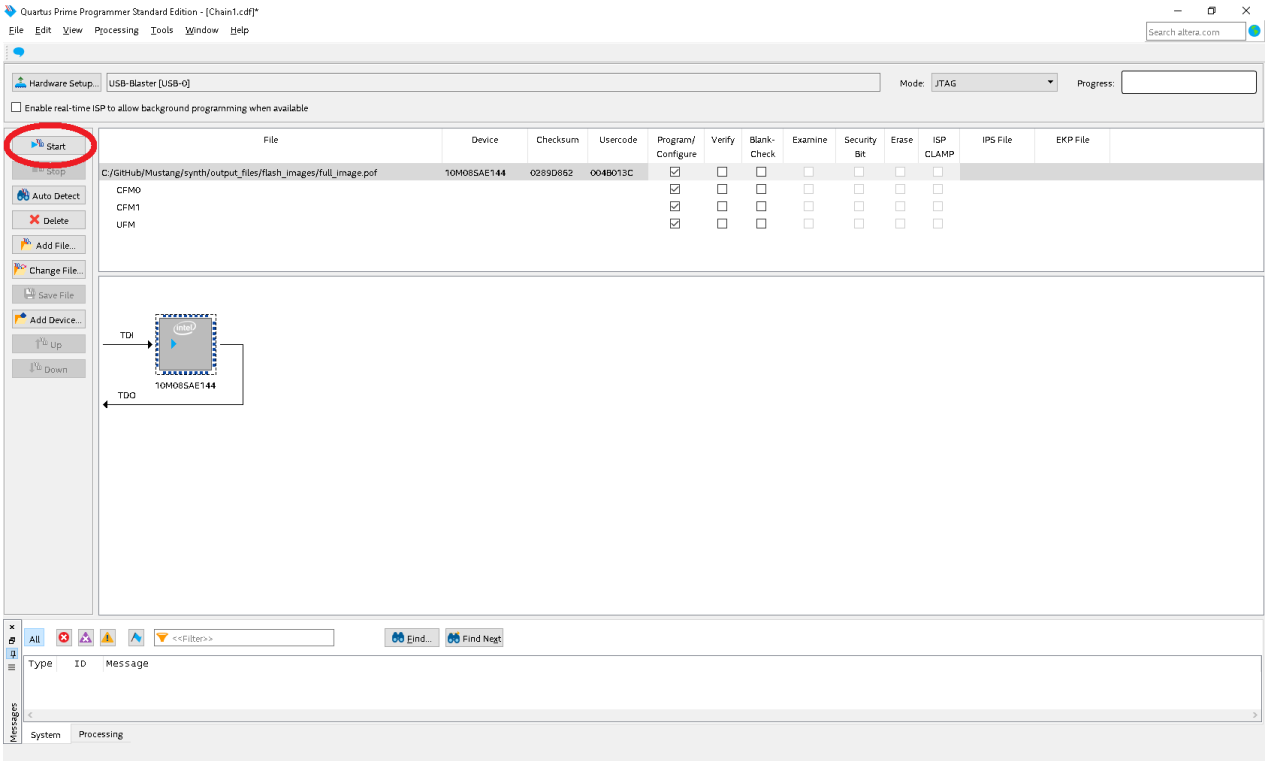

Figure 2-3 Program the Device

2.2 JTAG UART

The JTAG connection can also be used as a UART port, for which Altera has provided a JTAG UART IP core (Ref [1]). Accordingly, PulseRain Technology has designed a Wishbone wrapper to turn it into one of MCU's peripherals.

To interact with the JTAG UART, the user has to run "nios2-terminal.exe" on the PC side, as illustrated in Figure 2-4. The nios2-terminal can be launched by typing "nios2-terminal.exe" in Nios II Command Shell, as shown in Figure 2-5.

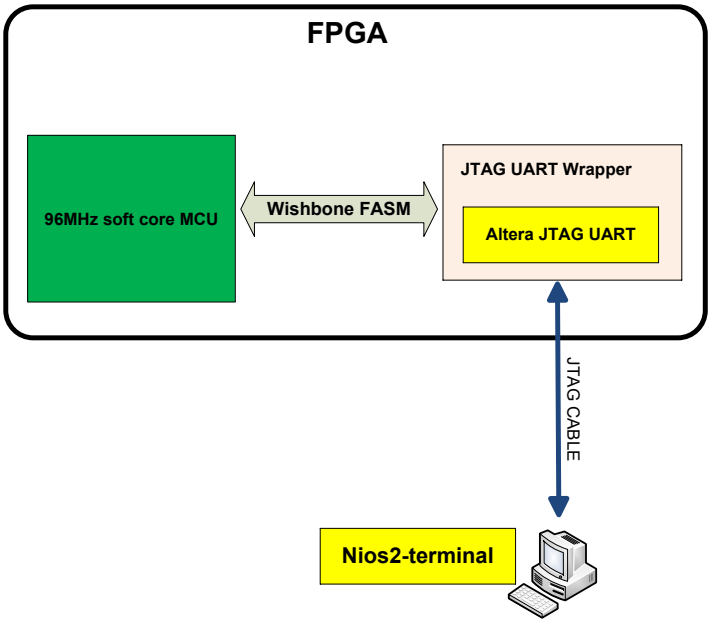

Figure 2-4 JTAG UART

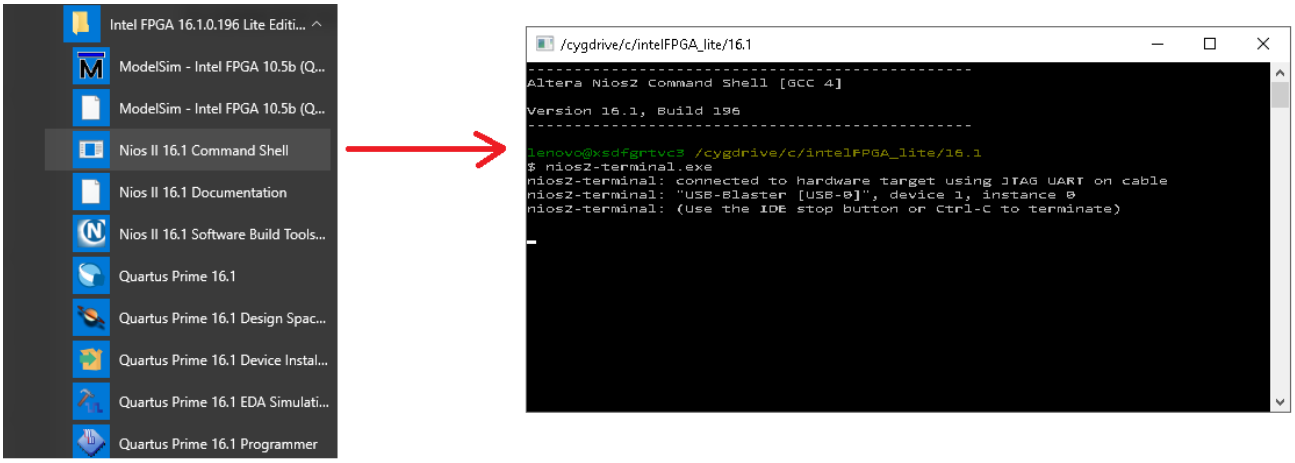

Figure 2-5 Launch nios2-terminal through Nios II Command Shell

2.3 Pin Assignment

The following pins are all related to the JTAG port of the onboard FPGA (10M08SAE144C8G):

Signal Name	FPGA Pin Assignment (10M08SAE144C8G)	Index on JTAG Header	Description
JTAGEN	15	N/A	Enable of JTAG function
TCK	18	1	JTAG Test Clock
TDO	20	3	JTAG Test Data Out
TMS	16	5	JTAG Test Mode Select
TDI	19	9	JTAG Test Data In
GPIO	46	6	General Purpose IO
GPIO	60	7	General Purpose IO
GPIO	47	8	General Purpose IO

Table 2-1 FPGA Pin Assignment

2.4 FIFO Depth of JTAG UART

By default, the JTAG UART was configured with

Write FIFO Depth = 16 bytes

2.5 Repository

The code for JTAG UART core can be found on GitHub, in the "cores" folder of

https://github.com/PulseRain/PulseRain_FP51_MCU

And its Wishbone wrapper can be found in the peripherals.sv under the "peripherals" folder.

3 Software

3.1 Register Definition

There is only one register for JTAG UART:

- JTAG_UART (8 bit, WO)
  A write to this register will send data to the write FIFO of the JTAG UART.

3.2 Address Map

The registers defined in Section 3.1 are mapped into MCU's address space, as shown in Table 3-1.

Address	Register Name
0xF8	JTAG_UART

Table 3-1 Address Definition for JTAG UART

3.3 Arduino Library

PulseRain Technology has provided the M10JTAG library to help users print data and string to the JTAG UART.

3.3.1 APIs

- *void print (int32_t num, uint8_t fmt = DEC)*
  Call this function to print out an integer in specified format. The valid formats are: DEC (Decimal), OCT(octal) and HEX(hexadecimal). And the default format is DEC.
- *void println (int32_t num, uint8_t fmt = DEC)*
  This function does the same as that of *print ()*, plus it will append a carriage return (new line '\n') at the end.

- *void write (uint8* buf)*
- *void write (uint8* buf, uint16_t length)*
  Call these two functions to print out a string to the JTAG UART. If *length* is not given, the *buf* must be a string that ends with a null (a byte of value zero).

3.3.2 Examples

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

- *JTAG_print*
  This example will print out a rolling counter at 1 second interval. To see the print out value, the user should launch the nios2-terminal on the PC side, as mentioned in Section 2.2.

3.3.3 Flow Control

Currently, the JTAG UART wrapper has no flow control in its hardware implementation. If the nios2-terminal running on the PC does not pull data fast enough, there could be a FIFO overflow. Thus, it is recommended to have enough margin of time in place between the writes to JTAG UART.