Naval Research Laboratory

Washington, DC 20375-5320

NRL/MR/5320--09-9159

Analysis of Trimble Mini-T GPS Receiver

Richard M. White
John P. McConnell
Daniel S. Newton

Advanced Radar Systems Branch
Radar Division

March 31, 2009

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
31-03-2009	NRL Memorandum Report	FY07 – FY08

4. TITLE AND SUBTITLE

Analysis of Trimble Mini-T GPS Receiver

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Richard M. White, John P. McConnell, and Daniel S. Newton

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/5320--09-9159

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
875 North Randolph Street
Suite 1425
Arlington, VA 22203-1995

10. SPONSOR / MONITOR'S ACRONYM(S)

ONR

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report summarizes the function, capability, and I/O of the Trimble Mini-T GPS receiver (officially known as the Trimble Mini-T GPS Disciplined Clock Module; P/N 57303-05). The primary objective of this report is to document the intricacies of the Mini-T beyond the scope of its user manual, for the purpose of future integration of the Mini-T into larger, GPS-based systems. A detailed analysis on the input and output of the Mini-T was performed, down to the bit-level, and the basic Mini-T communication packets were dissected. Details and suggestions on how to write custom software or design hardware to communicate with the Mini-T are also provided, as well as a discussion on using an FPGA to communicate with the receiver.

15. SUBJECT TERMS

Trimble Mini-T
GPS receiver

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Richard White
Unclassified	Unclassified	Unclassified	UL	13	**19b. TELEPHONE NUMBER** *(include area code)* (202) 404-5413

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

CONTENTS

Analysis of Trimble Mini-T GPS Receiver

Introduction

This memo summarizes the function, capability, and I/O of the Trimble Mini-T GPS receiver (officially known as the Trimble Mini-T GPS Disciplined Clock Module; P/N 57303-05). The following analysis was performed primarily to develop an understanding of the I/O of the Mini-T so that the Mini-T could be integrated into a custom designed system. Details on how to write custom software or design hardware to communicate with the receiver are also provided, as well as a discussion on using an FPGA to communicate with the receiver. A list of all of the available commands used to transmit to the Mini-T, and their respective reports from the Mini-T, will not be given here since the Mini-T user manual [1] already contains these details.

1.0 Background on the Mini-T

The Mini-T is a GPS receiver capable of delivering information about the location of the receiver as well as the satellite signals. The user manual [1], as well as Trimble's demo software to control the board, can be found at www.trimble.com. The demo software, DSP_Mon, does not come with any support and Trimble will not provide the source code.

The Mini-T uses a precision ovenized oscillator for "near-atomic" clock precision timing. The Mini-T compares a 1 pulse-per-second (PPS) derived from this oscillator to the 1 PPS from an onboard GPS receiver, and uses the GPS measurements to steer the oscillator. The oscillator will warm up within a few minutes after power-up. Once at least one satellite signal has been accessed (usually within a few seconds after power-up), the Mini-T will then begin to output a 1 PPS signal. The Mini-T then begins a self-survey mode using all available satellites to eventually acquire an overdetermined time reference.

The Mini-T uses a serial communication link to send and receive data. Trimble's demo software (DSP_Mon) is a PC executable file that runs on any modern version of Microsoft Windows. The demo software provides a simple GUI that allows the user to issue any of the commands listed in the user manual, including firmware changes, specific information requests, and changes to the default reporting practices of the Mini-T. The demo software GUI also allows the user to easily interpret the reporting from the Mini-T and notifies the user of any errors or alarm conditions present. More specific information, such as signal levels from all available satellites, can also be requested from the GUI and viewed in a pop-up window.

Manuscript approved March 22, 2009.

2.0 Setting up and running the Mini-T

Details on setting up and running the Mini-T can be found in the Mini-T user manual [1] beginning on p. 13. Generally, this involves installing the demo software on a PC, connecting the serial port of the PC to the Mini-T, connecting the antenna to the Mini-T, and then just plugging in the Mini-T power cord. Running DSP_Mon now should produce the GUI shown in Fig.1. Within a few seconds, the GUI should show the Mini-T looking for satellites as it warms up the oscillator. Once the oscillator is warmed, the self-survey progress will be displayed in the "GPS Status" window.

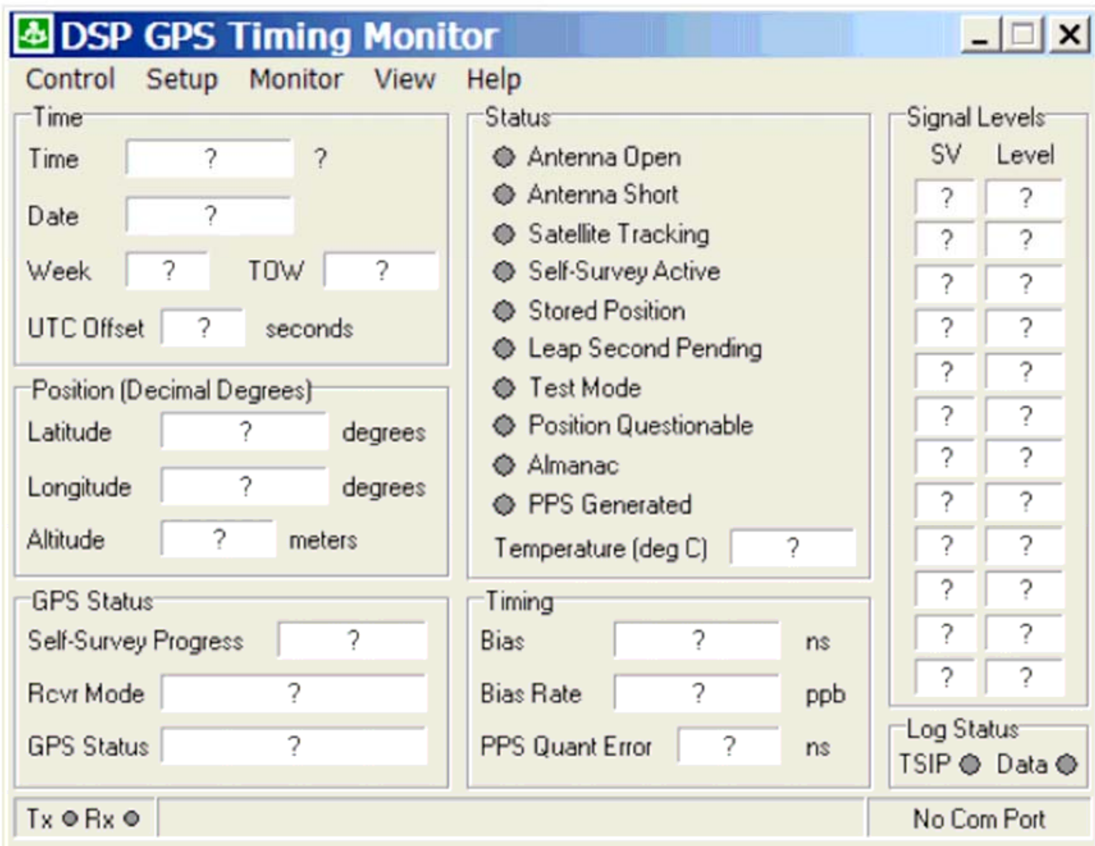


Figure 1. The initial DSP_Mon GUI (v. 1.53) on first start-up

Details about the receiver position and GPS time and date should be filled in shortly. All of the fields shown here in the main GUI (Fig. 1) are for display only and are not user adjustable, although the "Position" section and the "Temperature" field units can be changed by left-clicking on them.

2.1 DSP_Mon GUI inputs

All of the user inputs are accessed from the drop-down menus at the top left of the GUI (Fig. 1). Various types of resets can be issued from the "Control" menu, including a firmware reset to set all the options back to factory defaults. All user options such as "Signal Level Mask" (to ignore satellites with a signal level below a specified threshold) can be changed through the “Setup” menu. The “Monitor” menu only provides an option to enable or disable auto polling. Additional displays and output data, such as the position and status of each satellite being tracked, can be accessed through the “View” menu. Packet details and raw measurements can also be found under the “View” menu. The “Help” menu only contains an “About” screen. Trimble may provide some basic assistance if any further help is needed using the GUI, but it is unlikely Trimble will provide any detailed help since DSP_Mon is unsupported software.

Communication status with the PC is depicted in the lower right of the GUI, which displays the COM port number, baud rate, and bit format if a COM port link is set up. If “No Com Port” is displayed here, right-click to verify that the Mini-T’s COM port settings match the PC COM port settings. Either change the Mini-T port settings in the GUI or change the PC port settings so that the GUI and the PC are using identical COM port settings. The communication protocol can also be changed in the COM port settings window to either TSIP or NMEA . TSIP, which is Trimble’s proprietary protocol, is recommended since NMEA does not allow full use of the Mini-T’s features. This is noted and detailed in the Mini-T user manual.

If there is still no data shown in the GUI and the COM port is correctly set, it may be necessary to issue a hot or warm reset to the Mini-T (found under the “Control” menu). If no data appears after the reset, there is most likely a hardware problem, so contact Trimble for assistance at www.trimble.com/support.shtml.

As noted above, the Mini-T will not produce output on its 1 PPS port unless it has acquired at least one satellite signal. The Mini-T will most likely not be able to acquire any satellite signals while the antenna is indoors, even if the antenna is placed next to a large window. Therefore make sure that the antenna is properly mounted outdoors before even attempting to use the Mini-T.

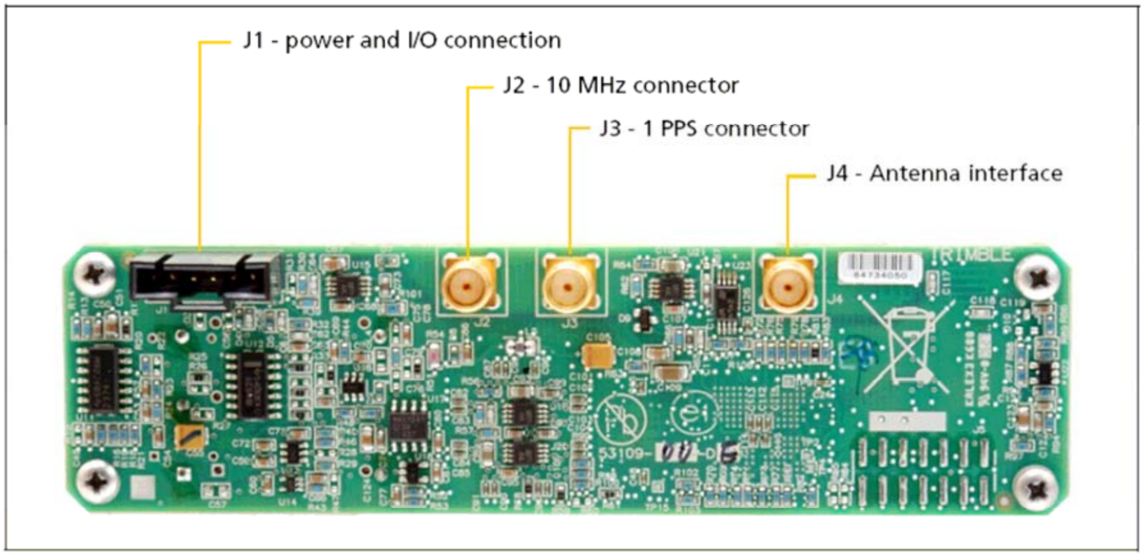


Figure 2. The I/O ports on the Mini-T board

3.0 Mini-T I/O

Information on the I/O ports of the Mini-T and hardware integration can be found in the Mini-T manual [1] beginning on p. 21. Details of the TSIP communication protocol can be found in the Mini-T manual beginning on p. 31. Figure 2 shows Mini-T with all of the I/O ports labeled.

Port J1 (6-pin Molex power connector P/N 70543-0005) provides the power and serial communication pins. The pinout for J1 is shown in Figure 3.

Pin	Description
1	NC
2	GPS TXD 3.3 V CMOS level
3	GPS RXD 3.3 V CMOS level (5 V tolerant) Pulled to 3.3 V using a 10 kΩ resistor
4	Ground
5	+V antenna. Supports 3.3 V ±10% at up to 100 mA.
6	+5 V ±0.25 V at <750 mA (cold) and <350 mA (warm)

Figure 3. Port J1 pinout

The optional starter kit that comes with the Mini-T includes a connector for J1 that connects to an AC power adapter and also has a breakout RS-232 connector that converts the onboard 3.3V CMOS data lines to RS-232 logic levels (+/- 5 V).

Port J2 is the 10 MHz output from the on-board oscillator. The phase noise of the 10 MHz output was measured against a low phase noise signal generator using an Agilent E5500 measurement set, and the result is plotted in Figure 4. Port J3 is the 1 PPS output locked to GPS time. This 1 PPS is CMOS level high for 10 μs. A timing data packet is transmitted to the port J1 TXD line shortly after the PPS pulse to which it refers. The 1 PPS output is asynchronous to the timing data packet sent from the board, and furthermore the data packet is sent at an inconsistent time after the PPS goes high. This time lag was observed to be anywhere from 10 ms to 150 ms, but is typically between 50-80 ms. Port J4 is the antenna interface. The Mini-T and the Bullet antenna can use 50 or 75 Ω cable. The input impedance of the Mini-T RF input and the Bullet antenna is 50 Ω, but 75 Ω cable can be used since (as the manual states) the impedance mismatch will only reflect 0.5 dB of attenuation at 1.5 GHz. RG-59 cable (75 Ω) comes with the starter kit and is recommended because it provides a better transmission for the 1.5 GHz GPS signal.

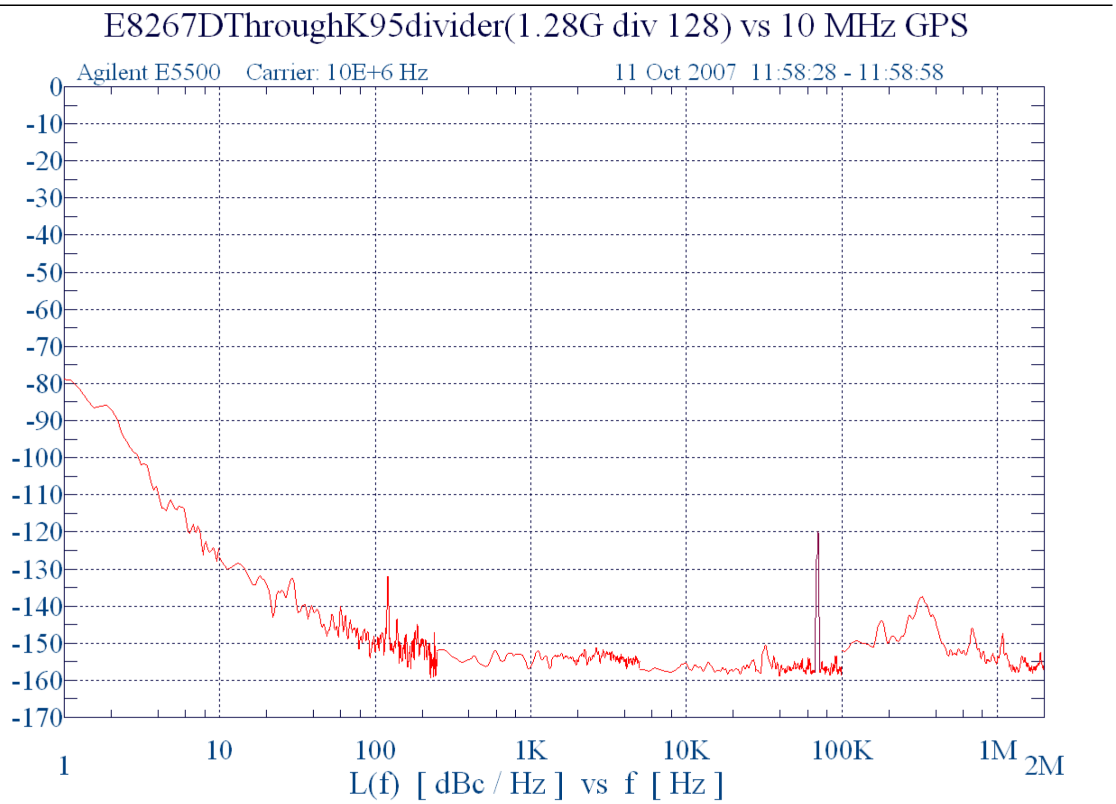


Figure 4. Phase noise on 10 MHz output

4.0 Communicating with the Mini-T

The details of the Mini-T serial protocol (TSIP) begin on p. 31 of the manual [1]. The basic structure is shown on p. 37 of the manual and is:

<DLE><ID><data string bytes><DLE><ETX>

where <DLE> is the byte 0x10, <ETX> is the byte 0x03, and <ID> is the packet identifier byte.

The packet identifier byte can have any value except <ETX> and <DLE>, but the board will not respond to unknown IDs (see the manual for all defined IDs). This is the format for every packet except in the case where the data byte <DLE> occurs in the data string, in which case another <DLE> is immediately added to avoid confusion. The data type can be 8-, 16-, or 32-bit signed or unsigned integer or single or double precision floats. The data type for each byte is defined in the manual under the detail about its respective packet ID, starting on p.38 of the manual.

Figure 5 shows a graphic representation of the serial transmission/reception used by the Mini-T. More information about the serial transmission format can be found at http://en.wikipedia.org/wiki/Asynchronous_serial_communication. The Mini-T serial port settings can be customized for various baud rates, numbers of data bits, and numbers of parity bits. The inclusion of start bit and stop bit cannot be changed, so each 8 bits of data will actually take 10 bits (plus any parity bits) to communicate. The example shown in Figure 4 is known as 8N1, which means eight character bits, no parity bits, and 1 stop bit.

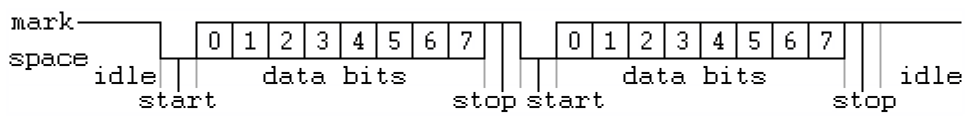
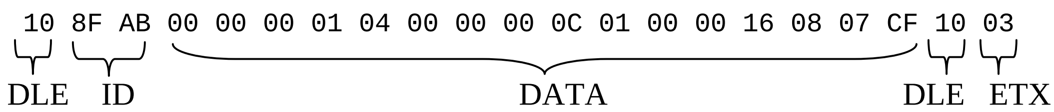


Figure 5. Asynchronous serial communication protocol example (8N1 shown)

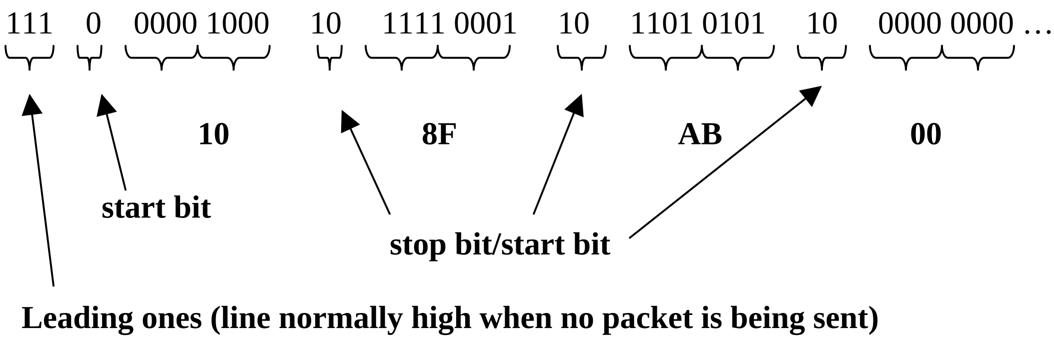
Trimble's demo software, DSP_Mon, includes a handy log writer that can write all the transmitted and received data to a text file. The user can choose to write raw hex data received (including DLE or ETX bytes which flag the beginning and end of data strings), or a more user friendly log that identifies the packet ID, data length, and hex data content of each transmit and receive packet. Neither option logs the start, stop, or parity bits since these are considered part of the serial protocol and not part of the message.

By default, the Mini-T automatically reports only the primary and secondary timing packet every second. The firmware can be set to automatically output other packets such as signal levels or raw measurements (or no packets at all). The primary and secondary timing packet are very useful though, as they contain most of the information a passive listener would be interested in, such as time, position, GPS status, and system health.

For an example, the following primary timing packet was captured using the DSP_Mon raw data logger:



As noted above, the start, stop, and any parity bits used are not shown in the logger. A logic analyzer was also used to capture the serial data stream. The logic analyzer captured the following string showing more detail of the packet:



The above example shows the most detailed form of the protocol, taken directly from the logic analyzer. This is the beginning of the primary timing packet again, but here the start and stop bits are visible. In this example the Mini-T was using 8N1 port settings, so there is no parity bit. The line is high until a message is sent, and goes back to high immediately after the last stop bit is transmitted. The bytes are sent LSB first. Although not seen here, the secondary timing packet is sent immediately following the primary timing packet. So if a user is listening to the line using the default reporting of only the primary and secondary timing packet, then the above sequence starting 10 8F AB is transmitted every second.

The default start up sequence was observed using DSP_Mon to log the transmitted and received packets. Upon power-up, the Mini-T transmits packet ID 45 which is the firmware version information. DSP_Mon then transmits packet ID 3F, and the Mini-T replies with packet 5F, both of which are undocumented packets. The Mini-T then transmits packet 8F-AB and 8F-AC (primary and secondary timing packets) every second.

A logic analyzer was also used to capture the start up sequence. The logic analyzer was set for asynchronous sampling of the Mini-T transmit line at a sampling period of 26 μs (one quarter of the transmit period of the Mini-T while operating with a baud rate of 9600 bits/second). In this case, there were no packets sent to the Mini-T, so the analyzer was just passively listening to the Mini-T. The start up sequence was the same as above except the undocumented 5F packet was not present. Note that this is the default setting. In addition to the primary and secondary packets, the firmware can be programmed to automatically transmit some other packets of information.

It is possible to use Trimble's software, DSP_Mon, to send packets to the Mini-T. However, since source code for DSP_Mon is not provided, it will be necessary to write new interface software if DSP_Mon does not fit the desired application. This involves writing software that can read from and write to the serial port. A simple terminal program written in C# that can read from and write to the serial port under Windows was used for testing. The program only allows for keyboard input for transmit, but can be easily modified since the source code is available. The code could also be modified to decipher reports from the Mini-T and display them in a convenient manner for the user to view and/or send that data off for further processing. This code could also be ported to Linux (DSP_Mon operation was not tested under WINE, which is software that allows Windows programs to operate on Linux machines). To verify that the program was capable of communicating with the Mini-T, the program was used to view the Mini-T reports after powering on. The same start up sequence was observed as was captured by the logic analyzer (packet ID 45, then packet IDs 8F-AB and 8F-AC repeating). Furthermore, the program was used to transmit packet ID 3A to the Mini-T, which requests the last raw measurements of a specified satellite. The program transmitted the following string to the Mini-T:

```
 10   3A   14   10   03
DLE   ID        DLE  ETX
```

DATA (requesting measurements from satellite 20, which is 0x14)

The Mini-T then responded with the expected response packet ID 5A, which included the raw measurement of 13.2 AMU for satellite 20. The terminal program was then closed and DSP_Mon was launched to confirm this value (both programs cannot share the port at the same time). DSP_Mon showed a slowly fluctuating raw measurement of around 14 AMU for satellite 20, which verifies the results from the C# program. This program provides a good base for using the serial port of a PC to communicate with the Mini-T. Communicating with the Mini-T without the use of a PC (building a circuit to directly handle the serial I/O) is discussed in the next section.

5.0 Using an FPGA to communicate with the Mini-T

Trimble is not aware of any vendors or customers that have created VHDL to allow an FPGA to communicate with the Mini-T. Programming an FPGA (or building a circuit) to handle the full read/write capabilities of the Mini-T would require code that stores the entire set of possible TSIP packet IDs and encodes/decodes all the data packets. One approach would be to use an FPGA with a processor core and a RS-232 interface core. This would make the RS-232 communication relatively easy, but the packet decoding and encoding algorithms would still have to be written.

Another approach would be to build the circuit to only listen for the automatic primary and secondary timing packets, which contain most of the primary user information. The receive line on the Mini-T could be wired to a port on the enclosure which would provide a way for the user to plug in a PC and make changes to the Mini-T firmware or request additional information from the Mini-T. Such interactions would most likely be very infrequent, so for typical users, there may not be much usefulness in devoting the time into programming an FPGA to handle writing capabilities. A high-level block diagram of a proposed decoder is shown in Fig. 6. Programming the FPGA to only listen to the automatic primary and secondary timing packets would greatly simplify the packet decoding process inside of the FPGA since the same packets are sent each time, contained in one long string.

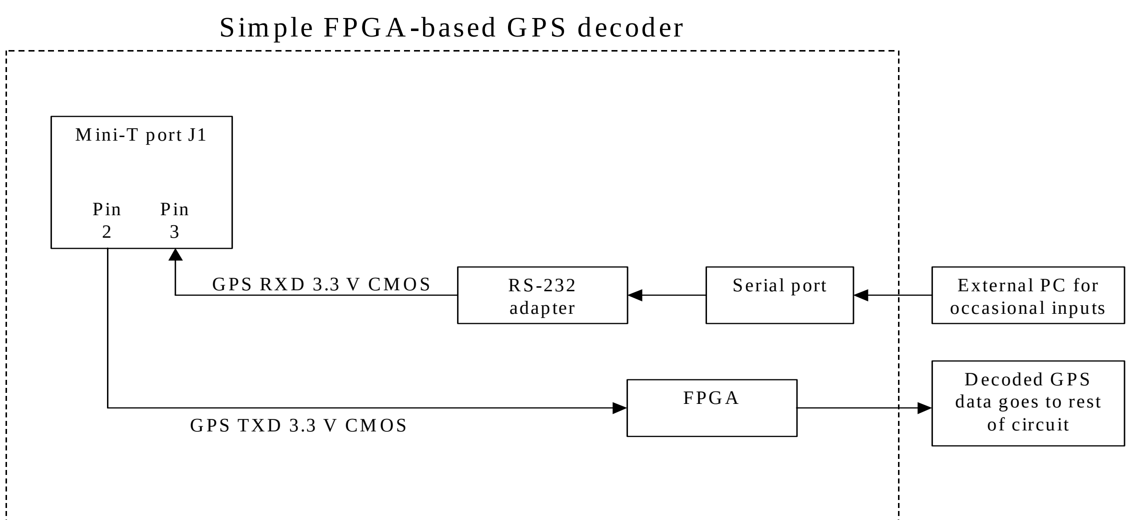


Figure 6. Proposed FPGA-based method for decoding Mini-T data packets

If no soft RS-232 interface core is used in the FPGA, a method to trigger the FPGA to start reading must be considered. The 1 PPS output from the Mini-T cannot easily be used as a trigger for the FPGA to start reading, since the time delay between the 1 PPS output and the start of the packet is inconsistent (more details on this in section 3.0). The FPGA would most likely have to be programmed to act like an asynchronous serial communication receiver, waiting for the line to drop low to signal the start of the message. The FPGA could then sample the line a fixed amount of times (since the primary and secondary packet always contain the same amount of bits) or until the <ETX> message is received. The speed of the FPGA sampling rate will not be an issue since any FPGA can easily handle the Mini-T's slowest baud rate of 9600 bits/second.

6.0 Conclusions

Trimble's Mini-T GPS receiver provides all of the expected GPS location information as well as some useful features such as a quality 10 MHz oscillator output and a 1 pulse-per-second output locked to GPS time. Trimble's proprietary communication protocol (TSIP) can be used to communicate with the Mini-T via a computer's serial port to provide extensive GPS data to the user. Trimble's demo software, DSP_Mon, allows a quick and easy method for viewing GPS data on a Windows-based PC. Experiments using a terminal program written in C# demonstrate how to write a more comprehensive program to access the Mini-T via the serial port and opens the possibility of data processing automation, as well as the possibility to communicate with the Mini-T in a Linux environment. Tests showed that it would not be overly complicated to hard-wire the Mini-T I/O into a larger system, such as using an FPGA to capture and process data from the Mini-T, if such a system were configured as read only. Another easy way to interface with the Mini-T would be through an FPGA that has a processor core and a RS-232 interface core.

7.0 References

1. Trimble Navigation Limited, "Trimble Mini-T GPS Disciplined Clock Module User Guide", Version 1.0, Revision A, March 2007.