

HamSČi



Personal Space Weather Station (PSWS)

# System Specification

*Preliminary Rev 0.5.1*

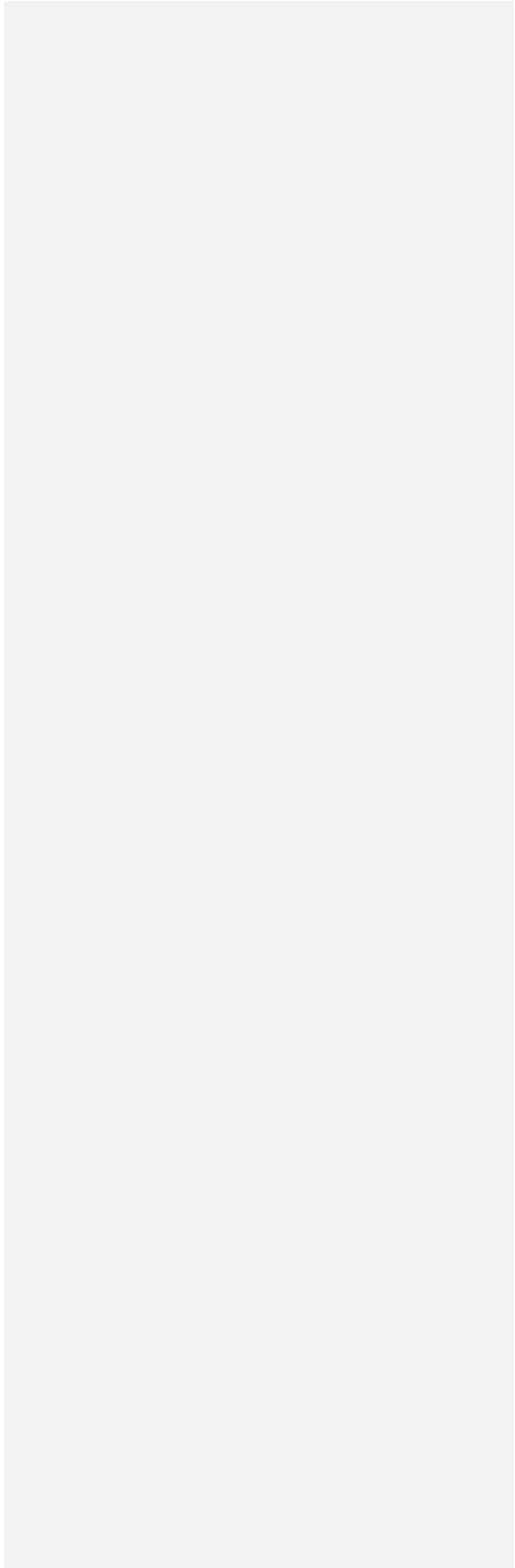
5 June 2019

1. INTRODUCTION.....	4
1.1. Cost Goals.....	4
1.2. System Components.....	4
1.3. Use Cases .....	5
1.4. Custom Design vs. COTS .....	5
1.5. Decision Needed.....	5
1.6. Parking Lot / Topics that need to be covered. ....	5
2. POWER SUPPLY .....	6
3. CLOCK MODULE.....	6
4. MAGNETOMETER MODULE.....	7
5. RF MODULE.....	7
6. DATA ENGINE FUNCTION.....	8
7. HOST COMPUTE FUNCTION.....	9
8. LOCAL COMMAND AND CONTROL PROTOCOL (LCC) .....	<del>11</del> 1140
9. REMOTE COMMAND AND CONTROL PROTOCOL (RCC).....	11
10. FIRMWARE REQUIREMENTS.....	12
11. SOFTWARE REQUIREMENTS.....	12
12. METADATA SPECIFICATION .....	<del>13</del> 1342
13. FILE FORMAT .....	13
14. IMPLEMENTATION SCENARIOS.....	<del>14</del> 1413

14.1. Figure 14.1 (Implementation A) ..... 14

14.2. Figure 14.2 (Implementation B) ..... 1544

14.3. Figure 14.3 (Implementation C) ..... 15



## 1. Introduction

The Personal Space Weather Station (PSWS) is a low-cost system intended to provide the following functionality:

- Two-channel fully coherent receiver covering 0 – 60 MHz.
- Up to 8 receive frequency slots simultaneously providing up to 192 ks/s each.
- User-pluggable RF filtering to reduce receiver overload in certain situations.
- Programmable 31-step 1 dB RF receive attenuator.
- Amplitude calibration of both receiver channels via an on-board noise generator including RF switching.
- GPS-based receiver providing both accurate timestamping of the received signals and well-controlled receive frequency.
- Integrated 3-axis magnetometer capable of 1 sample/second.
- Gigabit Ethernet interface.
- Storage of up to 24 hours of data in a circular buffer.
- Ability to be remotely polled and to trigger upload of a selected period of previously stored data from the past 24 hours.
- Capable of providing specified metadata.

### 1.1. Cost Goals

The goal of the project is to produce the receiver portion at a manufacturing cost of \$300. The Magnetometer, receive antennas, data storage, computer control, power supplies, housings, and Internet connectivity functions are not part of this cost goal.

### 1.2. System Components

The PSWS system is comprised of the following components:

- Data Engine (DE) function.
- Dual-channel receiver module.
  - The DE and RF modules might or might not be combined – TBD.
- 3-Axis Magnetometer module.
- GPS Receiver plus low-jitter and low-phase-noise clock synthesizer module.
- Two receive antennas plus connecting cables, suitably sited. It is recommended that field strength calibration be conducted per site.
- Local host compute function. This includes TBD TB of non-volatile storage and a TBD Mb/s connection to the Central Server (Locally, or via the Internet). Initially this is anticipated to be a Single Board Computer (SBC) for example a Raspberry PI 3B+ or an Odroid N2. But other implementations are possible.
- Power supply, power distribution, and equipment enclosure.

### **1.3. Use Cases**

The different use cases should be considered in the implementation of this specification. See the document:

PSWS Objectives, Use Cases, Ancillary Uses

Section 14 describes some implementation diagrams that may align with the various use cases.

### **1.4. Custom Design vs. COTS**

From a specification and operation point of view it is easiest to limit all the components to single specific units. This means that the performance can be well controlled and known, that FPGA firmware has a single variant, and that the software has a single variant. Multiple sources of supply for each component introduces variation in performance (perhaps unknown at the time), and the need for multiple variants of firmware and software.

On the other hand, having multiple sources for each component opens the possibility to enhancements, improved performance, reduction in cost, and alleviation of supply constraints. Additionally the use of pre-existing commercial-off-the-shelf (COTS) hardware and software may reduce the time needed to design, integrate, and test the system solution.

If multiple module realizations are permitted for each functional block, then the number of FW and SW variants needed could grow exponentially. This has the likelihood of making the project unmanageable in terms of on-going development effort and uncontrolled measurement uncertainty.

At the time this specification is written it is too early to tell which approach is required. It is hoped that finalization of the equipment list of materials can be reached and incorporated into this document over the course of the project.

### **1.5. Decision Needed**

Key open decisions are annotated in this document with **\*\*Decision Needed** in the text. These should be decided and documented during the development process.

### **1.6. Parking Lot / Topics that need to be covered.**

Add additional topics that need to be covered here. Move these to appropriate sections as defined and cross-off this list. There is a 'Parking Lot' document on the Hamsci google drive to capture issues from all project documents in one place.

## 2. Power Supply

The PSWS system shall operate from a single external power supply input of +11.0VDC to +15.0VDC (nominally +12VDC). This is desirable in order to allow remotely-powered and battery-backup deployments. All modules and subsystems will need to condition power from this input. If COTS modules are required that cannot utilize +12 VDC as input, then the project will need to design an appropriate voltage step-down and distribution unit. This unit shall provide > 80% power conversion efficiency in order to prevent heat accumulation when deployed in a remote enclosure and to minimize battery backup power consumption (thus extending the run-time hours available per unit of battery capacity).

**\*\* Decision Needed** The unit may be separate, or may be incorporated as a part of another subsystem (such as the data engine). The power consumption from the +12 VDC power input shall be less than TBD amps.

## 3. Clock Module

The clock module provides up to 4 clock outputs that are derived from GPS. It has a DC-power-sourcing short-protected +5 VDC SMA antenna connector that powers the GPS antenna/preamp, an on-board low-jitter and low-phase-noise multi-channel synthesizer, and up to 4 outputs. The outputs are:

1. FPGA clock.
2. ADC clock to each RF module(s). This shall come from one synthesizer output, and may utilize any stable distribution method (clock driver per module, or other approach the synthesizer may use to guarantee stable phase offset between the outputs over time and DC power cycles).
3. High-speed reference clock same as #2, but available externally.
4. One Pulse-per-second (PPS) timing output to the FPGA.
5. 10.0 MHz fixed reference output.

The ADC may supply a source-synchronous data clock back to the FPGA. The clock module may also have a 10.0 MHz reference input when GPS input is not possible. Individual clock outputs shall be able to be deactivated if not used.

**\*\*Decision Needed** The electrical format (differential/voltage swing, etc.) and FPGA / ADC clock distribution method is TBD. One possibility is differential LVDS because single-ended clocks are likely to be too noisy across connectors.

The FPGA clock provides the timing to the FPGA, and may provide the ADC clock via clock distribution, or may receive the clock from the ADC via clock distribution. This decision is TBD. Typically this signal will be in the 122.88 MHz range.

The ADC clock may need to be separately provided or may be provided via clock distribution from the FPGA. The quality of the ADC clock is critical to the performance of the entire PSWS. Excess phase noise decreases the dynamic range of the receiver and increases baseband jitter, while excess jitter degrades the noise level of the ADC at higher frequencies.

- The ADC clock phase noise shall be less than -60 dBc at 1 Hz offset, declining as 1/f to less than -120 dBc at 1 kHz offset.
- The ADC clock shall exhibit less than  $\pm 1$  picoseconds jitter, peak.

The PPS timing pulse shall provide better than  $\pm 50$  nanosecond peak timing compared to GPS time. This requires the use of a holdover oscillator, which ideally should be within the GPS receiver module itself.

#### 4. Magnetometer Module

The magnetometer shall be a 3-axis type, capable of being interrogated at least once-per-second. The magnetometer module shall be interconnected to the system using a TBD interface. (Candidates include but are not limited to: I2C, SPI, RS-422, serial, other).

It may be necessary to physically isolate the magnetometer from the system by a certain distance in order to avoid disturbing the magnetic field in the vicinity of the measuring instrument. The interface shall be capable of both powering and interrogating the module.

The magnetometer shall have less than TBD nanoTesla resolution, and greater than  $\pm$ TBD nanoTesla overload.

Cost of the module is a likely a key consideration, although outside the scope of this specification.

**\*\*Decision Needed** The actual magnetometer needs to be selected. The electrical interconnection interface, power requirements, and the range and resolution of the magnetometer need to be defined.

#### 5. RF Module

The RF module provides HF receive capability to the system. It amplifies and filters the received signals, and digitizes them. The performance of the RF module is key to the operational capabilities of the entire PSWS. The RF module or modules shall provide:

- Two separate RF channels. The channels shall have better than -30 dB isolation.
- These two channels shall be synchronously and coherently sampled without accumulating or varying phase offset. This requires that the same ADC clock be used for the two converters and that the same NCO signal be used for any downconversion. The use of two NCOs almost always results in uncontrolled difference in the phase of the downconverted signal and is a key problem in many COTS units. If the RF module is separate from the Data Engine, then the NCO requirement is imposed on the DE, not the RF module.
- The RF module shall accommodate a filter module that is able to accept various filters that may be needed for different receiver locations. The RF module shall also function with no filter. The filters might include:
  - AM Broadcast band rejection
  - SW station rejection.
  - FM Broadcast rejection.
- The RF module shall have programmable attenuation in the front end to reduce the possibility of ADC overload that can occur in some situations.
- The RF module shall provide at least 14 bits ADC resolution. The actual ENOB of the converter is TBD.
- The RF module shall provide less than 10 dB noise figure when no attenuation is selected.
- The RF module shall provide greater than 88 dB dynamic range for directly-sampled (unconverted) signals. This is defined as the ratio of the maximum allowable signal to the spurious and/or 3<sup>rd</sup> order signal(s) which are likely below the ADC noise floor and thus might only be uncovered after averaging or equivalent.
- A noise generator that can be switched into each RF antenna path for generating a known amplitude signal level to be used for calibration.

**\*\*Decision Needed** Whether the RF module is separate or integrated with the Date Engine. This is primarily a decision of purpose-built vs. COTS implementation.

## 6. Data Engine Function

The Data Engine function (DE) provides the electrical signals to control the receiver, and possibly the Magnetometer (although that could also be controlled by the Host computer instead). It converts the ADC samples to packetized information. It provides a mechanism to mark low-level metadata within the packets. It provides a control interface to the host computer for setting frequencies and downconversion parameters. The DE contains a programmable FPGA due to the high sample rates and precision timing needed. It provides data processing for 2 RF channels.

There are several implementation possibilities:

1. The DE is physically separate from the Host Compute function and is interconnected to it via 1 GE Ethernet. The Host Computer might be a Single



Board Computer (SBC) or it might be a high performance Computer merged with the Central Server and connected to the DE via 1 GE.

2. The DE and Host Compute functions are merged into a single module. In this case there does not need to be a 1 GE interconnecting them. However the Local Command and Control protocol would need to be virtually implemented.

The DE connects to the clock module to receive FPGA clock, ADC clock, and PPS timing mark.

The DE shall provide downconversion, filtering, and decimating up to 8 RF bands of 192 kHz bandwidth per RF channel.

The DE shall provide low-level metadata marking in the data packets consisting of:

1. Identify which RF channels the samples are from.
2. Identify which RF band (if downconversion is enabled) that the samples are from.
3. Identify the sample rate of the RF band.
4. Identify and timestamp the received data. There are several approaches possible:
  - a. Put a timestamp in each packet identifying the time of the first data sample within the packet.
    - i. The format should include both time in seconds and time in fractional seconds.
    - ii. It is recommended to use the 64-bit format described in NTP.
    - iii. If the Clock module only supplies GPS time, then the DE should use GPS time. If the clock module can provide UTC time, then that is preferable. The Host or Central Server would need to change GPS time to UTC time if the packet is marked with GPS time.
  - b. Place a PPS mark in the packet, and indicate which sample number it applies to. Use sample zero if the packet does not contain a PPS edge.
- 4-5. Identify whether the data consists of received signal samples or amplitude calibration samples.

The host computer is responsible for higher-level metadata marking, and potentially the file encoding format (TBD).

If the DE and Host are physically separate devices, then the DE may need to stream data directly to the central server if the Host cannot process fast enough.

## 7. Host Compute Function

The host compute function is responsible for:

- Controlling the DE, i.e. setting the RF bands and attenuator.
- Processing received packets from the DE
  - Either over 1 GE or via a virtual interface.

- Applying high-level metadata to the packets.
  - Metadata version marking
  - Frequency, time and date of the data.
  - SiteID or other data (such as GPS coordinates) of the receive antennas.
- Storing the Packets in a 24-hour circular buffer.
- Receiving commands via the Ethernet interface.
  - Remotely via the Internet, and
  - Locally via a Web interface.
- Transmitting the data to a central server upon request.

**\*\* Decision Needed**

- Still to be determined is whether the Host computer will provide file formatting such as Digital RF / HDF5 format, or whether the central server will perform that conversion. The issue is that the Host may not be able to keep up in real time with HDF5 encoding due to the very high processing load.

The Host compute function shall provide a 1 GE Ethernet interface, either to handle both DE $\leftrightarrow$ Host communication and/or Host $\leftrightarrow$ Central Server communication.

**\*\* Decision Needed**

- Still to be determined is whether the DE or the Host will interface to, collect, and format the data from the Magnetometer module.

The Host computer shall provide sufficient local non-volatile storage to hold 24 hours of received samples. The exact amount of storage needed is not completely defined at this point, but can be estimated by:

$$\begin{array}{ll}
 86400 \text{ seconds} & (24 * 60 * 60) \\
 6.144 \text{ Ms/s} & (2 \text{ RF chan} * 8 \text{ RF band} * 192000 \text{ samples/sec} * 2 \text{ (I+Q)}) \\
 4 \text{ bytes/s} & (\text{single precision floating point})
 \end{array}$$

$$= 2.13 \text{ Terabytes}$$

The non-volatile memory interface data sample rate can be estimated as:

$$\begin{array}{ll}
 6.144 \text{ Ms/s} & \text{sample rate} \\
 4 & \text{bytes / sample} \\
 \text{few bytes} & \text{metadata per packet}
 \end{array}$$

$$= \text{Roughly } 25 \text{ Megabytes / sec.}$$

This requires SATA-2 or equivalent higher disk throughput (the interface could be SATA2/STAT3, USB3, etc.). It is unlikely that a USB2 interface to disk could handle this data rate on a sustained basis.

## 8. Local Command and Control Protocol (LCC)

The protocol between the DE and Host is called the Local Command and Control protocol. This is used by the host to set the DE mode of operation, and control the subtended modules (i.e. the RF receiver, Magnetometer, and others). It also communicates the status of the controlled units back to the host.

At this point the protocol is not defined, and will await further development, and probably will be dependent on any COTS modules selected for the PSWS.

The protocol shall be covered in a separate specification. The LCC Protocol specification should cover at a minimum:

- Protocol revision control, revision marking and reporting.
  - Including GitHub or other revision control
- Description of all HW modules utilized and controlled.
- Reference to the Low level metadata specification.
- Command and Control interface.
- How it will handle precision time marking. [See section 6 for discussion.](#)
- Status interface.
- Reference to the packetization definition.
- **DECISION NEEDED on whether the I & Q samples are 24-bit 2's complement binary, or 32-bit single precision floating point in the packets from DE to Host.**

Formatted: Font: Bold

The LCC will exist on a physical interface if the DE and Host are separated, or may exist on a virtual interface if not.

[The DE may implement a discovery function on the local LAN, or may delay this to a later phase. Parts of the discovery function would need to be substituted if not implemented in order to allow the DE MAC address to be discovered, the DE IP address to be assigned, and potentially give the DE knowledge of the gateway MAC address. This could happen via provisioning into DE non-volatile memory for example.](#)

**DECISION NEEDED: The implementation of DE discovery in Phase 1 needs to be decided and documented.**

Formatted: Font: Bold

## 9. Remote Command and Control Protocol (RCC)

The protocol between the Host and the Central Server is called the Remote Command and Control Protocol. Because this protocol will be carried over the public internet, it must be capable of being secured against unauthorized use. What type of security and authentication is necessary is TBD. The purpose of the protocol is for the central server to be able to:

- Establish the operating parameters of the PSWS, such as bands tuned, sample rate, attenuator setting, etc.

- Trigger the retrieval of data covering a time in the past that is within the PSWS buffer on non-volatile storage.
- Negotiate the file format (whether the host or the server does file encoding, etc.)
- Identify the remote station and confirm.
- This protocol is likely subject to frequent revision as the project needs evolve.

The protocol shall be covered in a separate specification. The RCC Protocol specification should cover at a minimum:

- Protocol revision control, revision marking and reporting.
  - Including GitHub or other revision control
- Reference to the High level metadata specification.
- Reference to the File format specification if the Host does file encoding (such as Digital RF / HDF5).
- Command and Control interface.
- Status interface.

The RCC will exist on a physical interface if the Host and Central Server re separated, or on a virtual interface if not (for example via localhost 127.0.0.1 if the Host and Central Server functions reside on the same high-performance computer).

## 10. Firmware Requirements

Firmware is defined to be the FPGA code that is deployed on the DE module. This shall be covered in a separate specification due to the size and complexity anticipated. The FW spec should cover at a minimum:

- FW Revision control, revision marking and reporting.
  - Including GitHub or other revision control
- Hardware interface definition for the FPGA.
- Block diagram.
- Low level metadata applied.
- Command and control interface.
- Status interface.
- Packetization process.
- External modules used (Protocol stack, GE interface, filters, etc.).
- Toolchain utilized, including versioning.

## 11. Software Requirements

Software is defined to be the Host source code. This shall be covered in a separate specification due to the size and complexity anticipated. The SW spec should cover at a minimum:

- SW Revision control.

- Including GitHub or other revision control method.
- SW Architecture document.
- High level metadata applied.
- Command and control interfaces:
  - To the DE,
  - To the Internet,
  - To a local Web client.
- Status interface.
- Toolchain utilized, including versioning.
- Non-volatile memory storage.
- Interface to the Internet or modem/switch/gateway (Layers 2 and 3).
  - Including DHCP, Firewall, VPN, etc. as needed.

## 12. Metadata Specification

The metadata is split into two portions:

- Low-level metadata (applied by the DE).
- High-level metadata (applied by the host).

The metadata shall be defined by a separate specification. Critical to this specification is revision control, revision identification, and revision marking of the data. It is anticipated that the metadata will change on a research-by-research project basis. As each new research activity needs to identify some differing (and at the present unknown) attribute of the data, the metadata will likely change.

Additionally, as new hardware is added or changed out to accommodate new research requirements, the metadata formats applied will also likely change. This includes the identification of the hardware version producing the data, as the resolution, accuracy, and range of the new hardware may differ significantly from older hardware.

Each data packet sent to a central server needs to be annotated with metadata, and with metadata version identification if it is to be utilized in the future. It defines the ability to accurately curate the archived data long-term. If key attributes of the data are unknown 5 or 10 years hence, the archived data may be of little use.

## 13. File Format

The information archived by the central server should be in a readily-accessible standard format. One example is the Haystack Digital RF / HDF5 container set of standards.

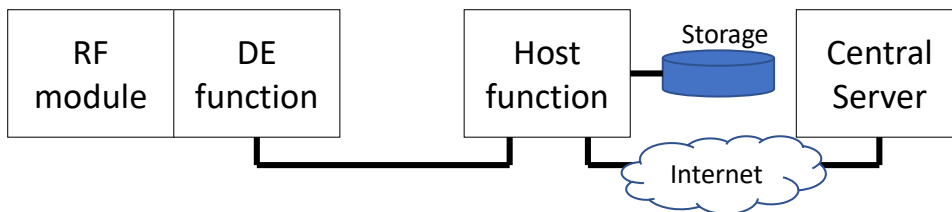
As previously described, it is still undecided whether that file format conversion happens in the Host computer or in the Central Server. If the later, then the Host computer needs

to provide sufficient information to the central server to accurately translate the information to the desired file storage format.

It is anticipated that the file format is adequately specified elsewhere, and shall be included by reference in this document.

## 14. Implementation Scenarios

The actual implementation of this specification is not standardized. There are three ways that the functional might be implemented or used depending on technology available and the use case needed.



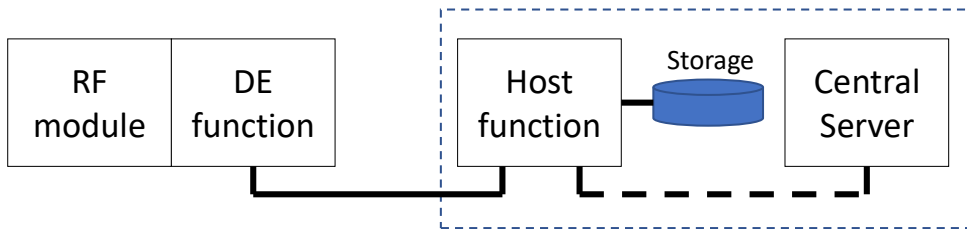
Implementation A: Separated DE, Host, Server.  
(Remote Node)

### 14.1. Figure 14.1 (Implementation A)

Illustrates an implementation where the DE, Host, and Central Server functions are separate. And the remote node is reachable via the Internet. The modules might consist for example of a TangerineSDR RF / Data Engine, a Single-Board-Computer (SBC) with disk storage, and an Internet connection to the Central Server.

Tucson Amateur Packet Radio

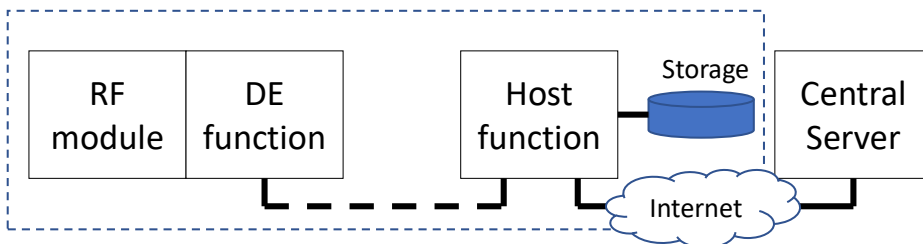
Personal Space Weather Station



Implementation B: Host Integrated with Central Server.  
High bandwidth LAN from Data Engine to Processing.

#### 14.2. Figure 14.2 (Implementation B)

Illustrates an implementation where the RF/DE function is located on the same high bandwidth LAN as the central server. For example, a TangerineSDR implements the RF/DE function, which is connected via Ethernet to the combined Host + Central Server function. The connection between the Host function and the Server function in this case is entirely internal to the server and could be implemented virtually. The Disk storage similarly could be physically resident on the server. This eliminates the single-board-computer hardware, and likely significantly increases the data rate from the DE to the Server function.



Implementation C: Data Engine Integrated with  
Host (e.g. FPGA with embedded CPU).

#### 14.3. Figure 14.3 (Implementation C)

Illustrates an implementation where the Host function is integrated with the RF/DE implementation. Such units exist today (for example the Red Pitaya) but the performance of today's technology is lacking for a practical implementation. The CPU

core as implemented on an FPGA may have insufficient processing capability. Known existing units do not have fast enough or large enough non-volatile storage to hold 24 hours of data. Thus this implementation might be practical in the future with improvements in silicon and storage technology, but may not be cost-effective today.