ARMY RESEARCH LABORATORY

Electrical and Software Design Report for the Data Fusion Testbed

Brian Mays

ARL-MR-536 July 2002

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Adelphi, MD 20783-1197

ARL-MR-536 **July 2002**

Electrical and Software Design Report for the Data Fusion Testbed

Brian Mays
Sensors and Electron Devices Directorate

Abstract

This report describes in detail both the electrical and software design of the Data Fusion Testbed (DFT). The DFT was developed by the Acoustic Signal Processing Branch of the U.S. Army Research Laboratory as a field research platform to conduct state of the art Unattended Ground Sensor (UGS) research. The DFT was developed to bridge the gap between early algorithm research and real-world scenario test and evaluation. The DFT is a custom assembly of signal acquisition hardware, communications equipment, and signal processing hardware packaged in an environmentally rugged housing. The hardware assets of the DFT combined with its custom software architecture forms a generic UGS for use as a unique engineering tool in this specialized field of UGS research.

Contents

Appendices

Figures

1. Introduction

The Acoustic Signal Processing Branch of the U.S. Army Research Laboratory (ARL) is carrying out research into Battlefield target tracking with Unattended Ground Sensors (UGS). In support of these efforts, both sensor technologies and signal processing algorithms are evaluated for performance and utility so that new capabilities can be added to UGS. Once a sensor or algorithm has been tested in the laboratory, the next phase of engineering evaluation is a field experiment with live targets and realistic scenarios. The step from the laboratory tests to field demonstration can be significant, since the new sensor or algorithm is typically a small part of the overall concept being demonstrated. For example, a researcher may want to add a magnetic detection capability to a sensor field to see if this improves the ability to discriminate and track multiple vehicles in a Battlefield scenario. The desire to quickly conduct such field evaluations motivated ARL to develop the Data Fusion Testbed (DFT). Figure 1 shows a typical remote installation of the DFT to provide an acoustic array sensor location with metrology.

This report describes the electrical design, software architecture, and theory of operation for the DFT. Appendices are include to document wiring harness specifications, interconnect drawings and schematics for all custom subassemblies manufactured. For detailed information on the mechanical design and fabrication drawings of the DFT, see Probst[1].

Figure 1. Typical Installation of DFT.

[1]Probst, M., "Mechanical Design Report for the Main Control Unit of a New Acoustic Sensor Unit" (ARL-MR-467), July 2000.

2. Data Fusion Testbed Electrical Design

The design goal of the DFT was to provide researchers in the field of UGS a surrogate platform for algorithm and sensor development. The DFT provides the common infrastructure required by a UGS system allowing newly developed sensors and algorithms to be efficiently integrated into field experimentation hardware. This ability greatly reduces the time between concept and validation. The DFT was also structured to allow high visibility of the sensor's performance during field experiments and to aid researchers in the evaluation process. The electrical design maximized the use of commercial off-the-shelf (COTS) components but still required several custom subassemblies to ensure robust field performance of the DFT.

2.1 Overall System Architecture

Figure 2 shows the DFT overall system configuration. The DFT is the core unit providing electrical connectivity to raw sensor inputs as well as communication assets to remote research tools. This section covers the hardware, which the DFT is comprised of and general specifications related to their performances.

2.2 DFT Hardware

The DFT was designed with several key requirements established from the outset. The Testbed must support a large number (56) of analog input channels, host local and remote processing algorithms, provide data recording capabilities, work reliably in adverse environmental conditions, and run from a single DC power source. All mechanical media must be removable and the units must be self-contained for shipping purposes and to ease setup on site.

The DFT consists of two types of sub-assemblies: the Main Control Unit (MCU) and Analog Signal Conditioning Boxes (ASCB). The complete system will consist of one MCU and up to seven ASCBs. Each ASCB groups the channels in two blocks of four for the purpose of programmable gain and cut-off frequency control. The ASCBs are designed to be fully controlled by the MCU in the system using a simple four-wire interface to each box. This command and control (C2) capability is a key feature in removing human error from the setup phase of operation. This programmability also allows for automatic gain control or adaptive bandwidth control if so desired. The MCU is a self-contained assembly with several internal components. The main unit

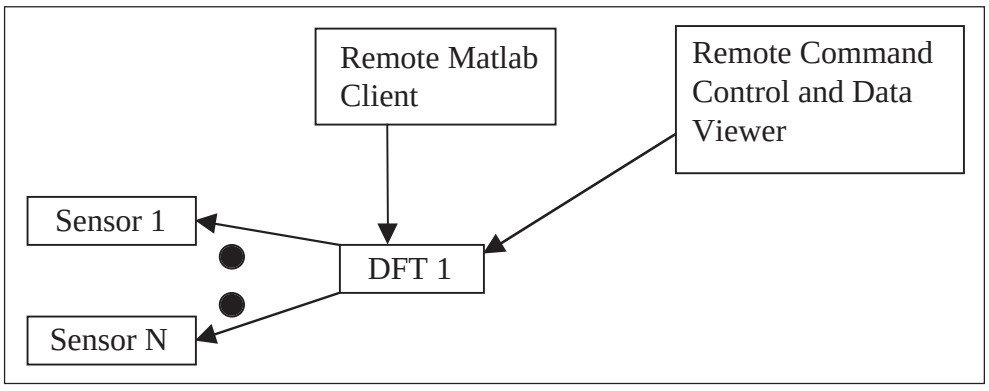


Figure 2. DFT Overall System Architecture.

provides a climate-controlled environment for the internal components and external connections to the ASCB, main system power, and diagnostic port for the unit. The internal components consist of a Compact Peripheral Computer Interface (CPCI) Cage, Packet Radio, wireless LAN radio, Removable Small Computer System Interface (SCSI) Drive, Precision Lightweight GPS Receiver (PLGR)/Global Positioning System (GPS) Receiver, Climate Control Circuit and DC/DC power Converter Unit. The system layout appears in the Figure 3.

In addition to the local DFT hardware, remote assets may also be attached to the system via the wireless LAN connection. These remote assets typically command and control the remote processing computers. More detail is provided section 3.2.

2.3 Analog Signal Conditioning Boxes

The system can handle up to seven ASCB components with each box providing eight channels of analog data. Each box groups the channels in two blocks of four for the purpose of programmable gain and cut-off frequency control. The ASCB is designed to be fully controlled by the MCU using a simple four-wire interface to each box. This C2 capability is a key feature in removing human error from the setup phase of operation. This programmability also allows for automatic gain control or adaptive bandwidth control if so desired. The units are powered by the MCU and contain no serviceable parts.

2.4 MCU

The MCU is a self-contained assembly with several internal components. The MCU provides a climate-controlled environment for the internal components and external connections to the ASCB, main system power and diagnostic port for the unit. In this section the internal components will be described and consist of: Compact Peripheral Computer Interface (CPCI) Cage, Packet Radio, Removable SCSI Drive, PLGR/GPS Receiver, Climate Control Circuit and DC/DC power Converter Unit. The internal configuration of the main unit is pictured in Figure 4.

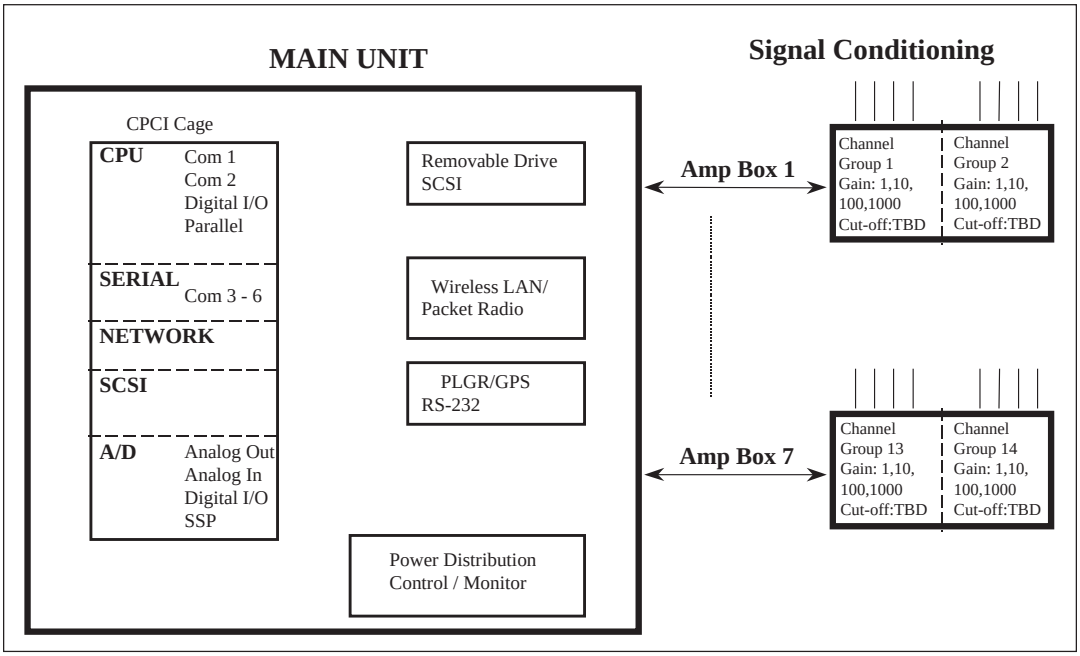


Figure 3. DFT Hardware System Architecture.

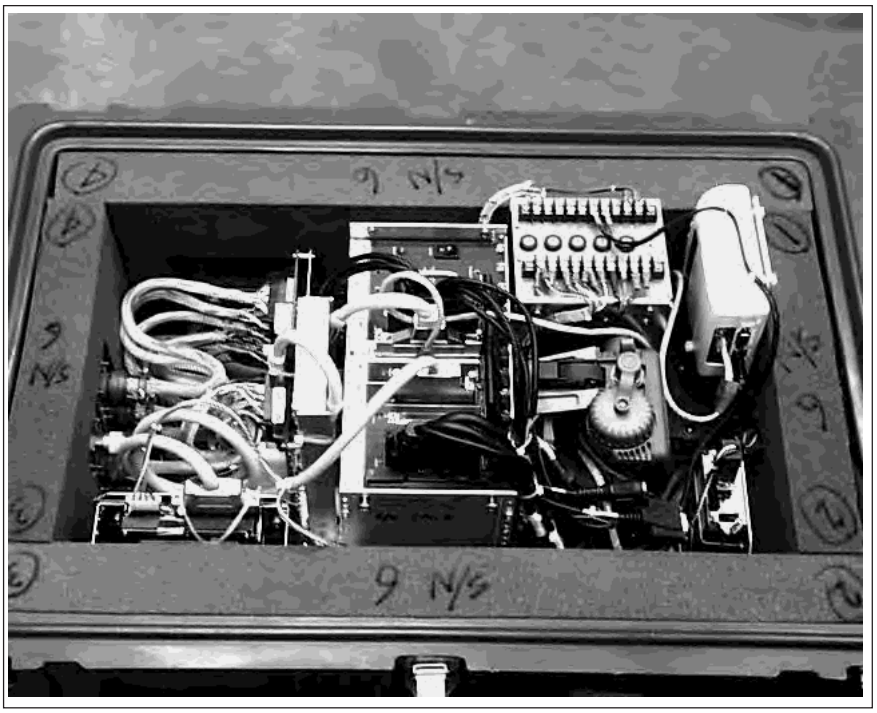


Figure 4. Top View of MCU.

2.4.1 CPCI Cage

The heart of the system is a Compact PCI Chassis. This architecture was chosen for two key reasons. One is that it can run all standard desktop software that is Intel compatible and the other is form factor. The chassis is a 3U high cage with a backplane structure and rugged mechanical frame. The backplane structure is important to allow expandability and component by component upgradability.

While Compact PCI is an open specification, all vendors have slightly different implementations of the overall package. The selected unit is manufactured by the Ziatech Corporation as Model ZT6080. This manufacturer was selected, at the time, for maturity of the product and the ability to remove the system's hard-drive easily. The hard-drive is mounted on a standalone card, which can be removed without disconnecting any other portions of the system.

The ZT6080 uses a Pentium-based system board with the following features:

- Compact PCI Bus Specification, Rev. 2.1 compliant.
- Supports Pentium processor CPUs to 200 MHz.
- Built-in numeric coprocessor support (Pentium).
- 16 kB of CPU cache (expandable with L2 cache).
- Optional floppy disk interface (local to ZT 6500).
- 8, 16, 32, or 48 Mbytes of DRAM memory.
- 2 Mbytes of Flash memory (expandable to 4 Mbytes).
- Standard AT peripherals include:

  - Two enhanced interrupt controllers (8259).
  - Three counter/timers (one 8254).
  - Real-time clock/CMOS RAM (146818).
  - Two enhanced DMA controllers (8237).
  - 8042 compatible keyboard controller.
- Centronics printer port (ECP/EPP compatible).
- Two 16C550 RS-232 serial ports.
- Single-stage watchdog timer.
- Speaker interface.
- On-board high-efficiency 3.3 V DC-DC converter.
- Push-button reset.
- Software programmable LED.
- DC power monitors (3.3 V and 5 V).
- Compatible with the following software: MS-DOS, OS/2, UNIX, QNX, VRTX32, Windows 3.1 and 3.11, Windows 95, and Windows NT.

The ZT6080 also contains a 10-Base T network card, SCSI card, IDE Hard Drive card and VGA card.

2.4.2 Analog to Digital Converter Card

The Analog to Digital Converter Card Selected is the ADC64 made by Innovative Integrations. The card is a 3U CPCI unit containing a TMS320C32 processor, and eight parallel analog to digital converter channels each with an eight to one multiplexer. This produces a total channel count of sixty-four channels for use by the sensor. The card has sixteen bits of digital input/ output and two digital-to-analog converter channels.

2.4.3 Packet Radio

The radio used in the DFT is a commercial packet radio made by the Pacific Crest Corporation. These radios are multichannel and have multiple access within a given channel. They employ a carrier sense collision detect network protocol with positive and negative acknowledgements which make them ideally suited for this application.

The model selected for the DFT is a two watt 9600 BAUD radio. These radios are simplex in their RF transmission so the 9600 BAUD represents the total data transmission rate in both transmit and receive directions. The radio's packet protocol allows individual unit addressing or broadcast mode for messages on a message-to-message basis. Typical use has shown the radios to maintain reliable communications over a 1 to 2 km range with quarter-wave dipoles at both ends of the links.

2.4.4 Removable SCSI Drive

The New Sensor Hardware includes a standard mounting location for a 3.5 in. SCSI drive. The primary function of this drive is for data storage of raw analog data, which is collected by the sensor unit. The specific SCSI drive selected for the sensor was a removable drive bay.

Removable media is important to the DFT so the sensor can be transported without any fragile rotating mechanical media. Another benefit is the data collection phase of a test can be extended by replacing the removable media with little or no impact to the sensor set-up.

2.4.5 PLGR/GPS Receiver

For applications requiring a mobile or self-locating sensor configuration, a mounting bracket and power connections have been provided for a standard PLGR/GPS receiver. The mounting bracket used is a standard vehicle mount with a quick release. This will allow easy crypto-keying of the unit and subsequent zeroizing prior to shipment.

2.4.6 Climate Control

A Climate Control system was designed for the DFT to extend the useful operating range of the sensor hardware. Because the sensor hardware is composed of COTS equipment, its temperature range of operation is quite limited. While all of the components of the system have different operating ranges, none of the equipment can tolerate condensing humidity or temperatures much below 0 °F.

The limiting piece of hardware for temperature rating currently is the hard drive, which has a range of 50–115 °F. While this could be extended by selection of another SCSI device, most commercial hard drives will have a similar operating temperature range.

The Climate Control System has two distinct control units. The first unit is a thermostat controlled heater and fan unit. The heater currently is a 30-W resistive unit whose primary role is for stand-by heat when the main unit electronics are turned off. The fan for the unit, shown in the following picture, is set up as a louvered door. This is to minimize any unwanted cooling due to wind penetrating the sensor. The fan and heater operation temperatures are currently set for “fan on” at 90 °F and “heater on” at 68 °F. The thermostat also has 7 °F of hysteresis.

The second portion of the climate control unit is out-of-range emergency shutdown. This portion of the electronics will remove all main system power from the sensor in the event of the equipment overheating or dropping below the minimum operating temperature. The current settings for out-of-range temperature are 51 °F for low and 116 °F for high. While this is not a gradual shutdown of the system, it is intended as overall protection of the hardware to prevent catastrophic failure of internal units.

Initial measurements show the current sensor configuration nominally draws sixty watts of power. With this power draw, the final thermal saturation point of the unit was 70° above ambient with no fan ventilation or additional heat from the resistive heater. With the inclusion of the resistive heater, this should allow for operation of the unit down to ambient temperatures of –50 °F. The airflow with the fan unit allows the internal temperature to be maintained at 5 °F above ambient. This delta will allow proper operation up to 110 °F.

The schematic for the Climate Control system is included in Appendix A of this report for reference.

Figure 5. Side View of MCU.

2.4.7 DC/DC Power Converter Unit

The DC-to-DC converter unit is a preassembled module built for the DFT. The module is comprised of four Vicor DC/DC converter units and two Vicor Ripple Attenuation Modules.

DC power for the sensor is divided into analog and digital supplies with separate grounds. The grounds are tied on the module via a 1-KW resister, but this is for protection purposes in the case where the main digital and analog ground becomes separated. The module generates four voltages: analog 7.5, analog –7.5, digital 12, and digital 5 V. The two analog supplies include the ripple attenuation modules for noise suppression. All of the supply module are capable of generating 75 W of power but are fused to 8 A for wiring protection.

The input power to the module is 24–36 V DC and is fused at 10 A. A schematic for the DC/DC power converter unit is included in Appendix A for reference.

2.5 Diagnostic Port

A diagnostic port is included on the unit and is accessible via the side panel. The port is enabled by a toggle switch also on the side panel. With the port enabled, the user can bypass the radio serial link to the computer and emulate a normal radio connection. This is useful in check out or diagnostic phases where an RF link is not possible.

2.6 Wiring Harnesses

All internal wiring for the sensor has been designed to allow complete cable harnesses to be prebuilt. This greatly eases the assembly process in that all internal connections are connector-to-connector or connector-to-screw lug on a terminal strip.

Cable Assembly Drawings are included in Appendix A of this report for reference.

2.7 Power Requirements

The unit is designed to be powered from a single 24–40 V DC source. The nominal power draw measured with 8 input channels of analog data was 60 W without the stand-by heater operating. The worst case estimates for power draw is 160 W for a unit operating with 64 analog input channels. The power consumption breakdown is shown in Figure 6.

3. DFT Software Architecture

The DFT software architecture design is based on a distributed multiprocessor, multi-tasked system. The design emphasizes a network of independent software processing modules to be interconnected by an interprocess communication link. With this architecture, different applications can be added or removed without adverse effects to the other areas of the Testbed. The design philosophy of the software architecture is that the system be modular, with independent software tasks and well-defined inputs and outputs. The software is designed to allow new algorithms to be quickly implemented as stand-alone processes. These additional algorithms simply attach to several DFT resources, which provide full integration into the DFT system.

The DFT can support 56 channels of analog input data. These channels are organized into groups of eight, clustered into individual ASCBs. The data groups can have different sample rates to allow multirate processing and minimize data throughput requirements. A group allocation file limits which algorithms can request data from specific groups to ensure algorithm isolation. The algorithms can be allocated multiple groups if more than eight channels of analog data are required. The current DFT can support up to 10 independent algorithms. The actual number of concurrent algorithms may be lower due to processor loading.

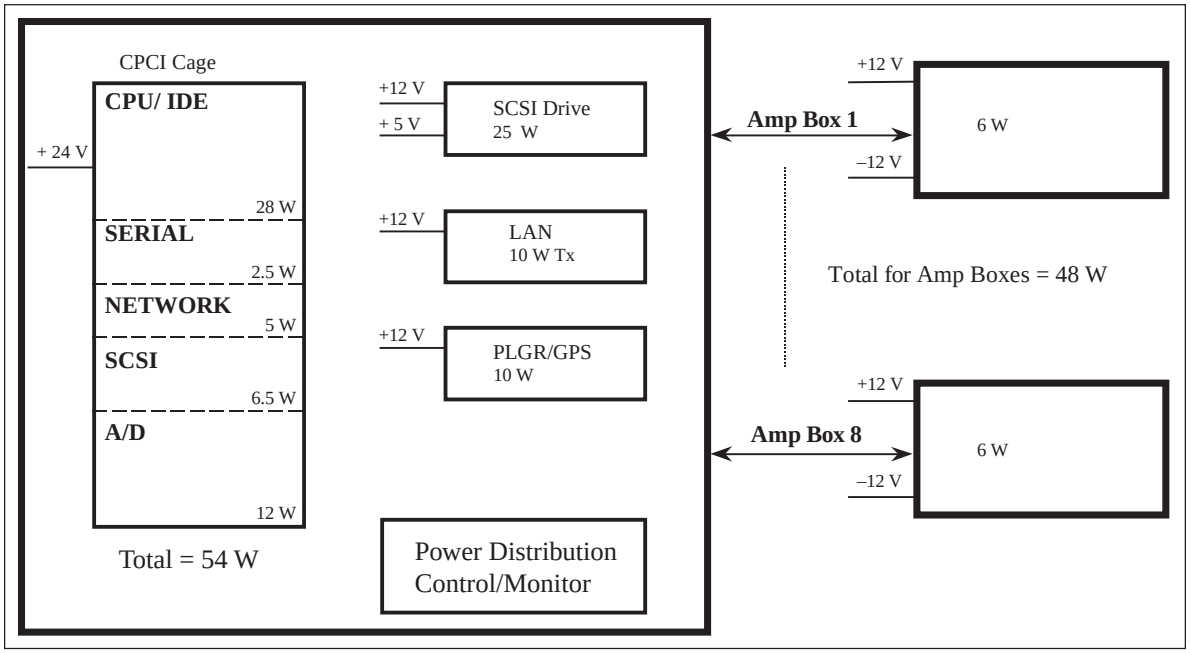


Figure 6. Total Power Consumption: 147 W + DC/DC Conversion Loss : Approx. 160 W.

3.1 Data Fusion Testbed Architecture

The DFT resources available to the algorithms include a data server, an inbound message dispatcher, an outbound message system and global DFT parameters. Figure 7 shows the main processes running on the DFT and their basic relationships. The main processes are the Data Server, Communication Manager, Dispatcher, Algorithm Server, Remote Monitor, and Recorder.

The Data Server manages the Analog-to-Digital converter asset and provides analog sensor data to the processing algorithms, while maintaining separation between the algorithms. The Data Server operates as a driver for the low-level interface to the ADC64. The specifics of the driver is covered by Vu[2]. The Data Server will process data requests and requests to change gains and cutoff values for the Analog Groups. The algorithm-based data requests are asynchronous to the data acquisition process so requesting algorithms receive the most recent data available. This approach allows algorithms that cannot keep up in real time to block process the data and still produce periodic answers. For example, an algorithm may request a block of data and take 5 s to process the information. The algorithms next request for data will be the most recent block of data acquired that is not 4 s old. When algorithms are keeping up in real time, requests made before the receipt of a new block of data are held until the data request can be fulfilled.

The data available to algorithms come in two forms: raw and Fast Fourier Transform (FFT) data. Algorithms request data by groups from 1 to 8 with each group containing eight channels of analog data. When multiple groups are requested by a single algorithm, the data server attempts to provide data from the same time window across the groups. Because the data acquisition and data requests are asynchronous processes, it is possible that all of the blocks are not from the same time window. In the rare case that this occurs, the requesting algorithm should re-request the multiple groups to ensure the times coincide. The data are passed to the requesting process with each group in a separate array. Each array of data contains a header followed by a data block.

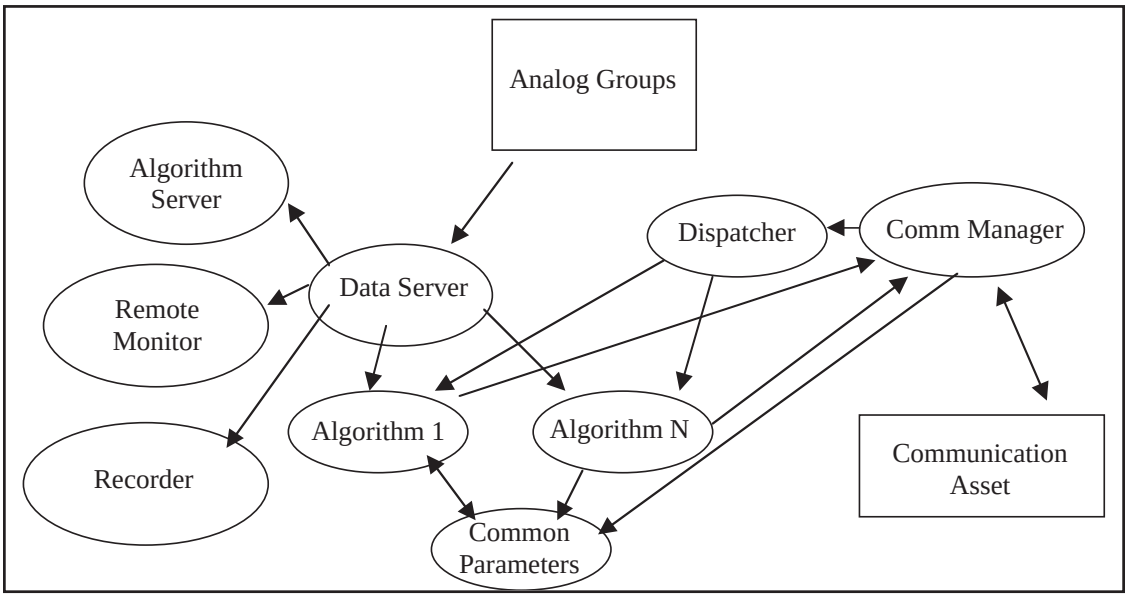


Figure 7. DFT Software System Architecture.

[2]Vu, H., "Data Server Architecture" (Internal ARL Report not published), 2001.

The Communication Manager is the process that maintains an external communications link to the Data Fusion Gateway Node (DFGN). The DFGN is an external system similar to the DFT which processes the results collected from many DFT nodes. The DFGN is physically based on the same hardware as the DFT. Its software and operation are described in Mays and Vu[3]. The manager has a simple message queue interface with the algorithms that allow arbitrary messages to be passed to the DFGN. The only requirement on the message format is that the top nibble of the first byte represents the task identification (ID) number of the process sending the message; this allows proper interpretation by the DFGN. The final role of the Communications Manager is to provide three different physical link options to the DFGN, which are transparent to the hosted algorithms on the DFT.

The Dispatcher process is closely tied to the communications manager and routes inbound messages to the appropriate algorithms. The inbound messages again can be arbitrary as long as the top nibble of the first byte represents the task id of the destination algorithms. Algorithms need to attach to the dispatcher and provide a child parser process to handle routed messages.

The Algorithm Server allows remote clients to attach to the DFT system and operate as if they were integrated into the DFT via the wireless TCP/IP connection. The Algorithm Server has facilities that allow the remote clients to request data from the DFT, control the ASCBs attached to the DFT, and send reports back to the DFGN via the DFT radio. This allows the results of the algorithms to propagate to the rest of the system as if the algorithm were hosted locally on the DFT. The Algorithm Server allows any TCP/IP capable client to attach to the DFT. The client is intended to allow higher level languages such as MATLAB and LABVIEW to be used as a processing engine in the DFT system. The remoting of the client also provides two additional benefits to system performance. First, it allows much deeper visibility into the remote algorithm because the remote application will have all of its display capabilities available on its hosting platform. Second, the remote algorithm can be integrated without reducing overall system reliability because no additional processor loading occurs and software crashes are isolated to the client. A final purpose of the remote client is to allow state of the art computer power to be applied to a problem, if required, without requiring the base DFT to be upgraded. As mentioned above, one key desire was to host native MATLAB m-files in the system. At the time of development, MATLAB did not provide native TCP/IP support. To solve this problem, ARL also developed a library of MEX files along with m-file rappers to allow direct access to the DFT's Algorithm Server from within standard m-files.

The Remote Monitor is also TCP/IP-based and operates over the wireless TCP/IP connection. It has a companion client, which is used for data monitoring during the DFT operations. This task is similar to that performed by the algorithm server except that data can only be viewed and no control of the ASCBs is possible. This limitation ensures that the data are not changed. The Remote Monitor allows complete data visibility into the DFT for data integrity monitoring or real-time sensor analysis. The monitor client was developed under LABVIEW to provide real-time data plot of specific analog channels or spectral waterfall plots of a specific analog channel.

The Recorder task supports various commands that allow the remote-controlling application to record specific data and monitor items such as disk space, channels being recorded, and to fully reconfigure the unit for changing system parameters such as sample rate, etc. ARL developed a

[3]Mays B., H. Vu, "Data Fusion and Tracking System Testbed." Proceedings of the MSS Specialty Group on Battlefield Acoustics, October 2000.

LABVIEW C2 application to operate as a main system console for multiple DFTs[4]. The recorder can be used in stand-alone data collection exercises or for background data achieved during full up-field experiments.

The final asset available to the hosted algorithms is a common parameter table in the form of shared memory. These common parameters are updated directly by the communications manager and contain system parameters that all algorithms should adhere to. Examples of these common parameters are report rates, report enable/disable status, etc.

4. DFT Adc64cc Software

4.1 Overview

The DFT uses the ADC64cc A/D card with onboard C30 processor for data acquisition and preprocessing. The software on the ADC64 provides multirate data acquisition for up to 64 channels, data preprocessing, and control of external signal conditioning boxes.

The hardware configuration groups the 64 analog channels into eight groups of eight channels. From a data standpoint, each group is treated as an entity and any parameters associated within a group are common to those eight channels. The channel groups are organized such that all channels within a group are sampled simultaneously. There are delays between groups due to multiplexers on the ADC64cc. These delays will depend on specific board setup and will be explained in a later section.

The current software allows the host to enable preprocessing on any number of the active groups. The preprocessing consists of windowing the data (Hanning), performing an FFT, and converting the data to IEEE floating point values. This processing is performed on every channel within the group when enabled.

The final responsibility of the software is to control external signal conditioning boxes, which have adjustable gain and cutoff frequencies. The software aligns gain change requests from the host to data block edges to minimize the transient effect on subsequent processing.

4.2 Host Requirements

The ADC64cc exchanges data with the host through two mechanisms. The first is simple register access referred to as mailboxes and the second is via DMA transfer to host memory. Basic command and status information is sent via the mailboxes and all data transfers are performed via DMA. General configuration data is sent to mailbox 0 and commands are sent to mailbox 3. To enable the ADC64cc card to perform the DMA transfers, the host must allocate contiguous physical memory from the host operating system. The host must also enable the ADCC64cc card to request the bus via the PCI configuration registers.

The first phase of the software on the ADC64cc is a configuration phase in which all basic card setup information is passed from the host to the ADC64cc. This phase is entered when the code is first run or when the host sends a reset command to the ADC64cc. The host must send the configuration information in strict order: DMA Pointers, Group Divisors, Max Sample Rate, and Update Rate.

[4]Goto, J., Cox, A. “User’s Guide for DFT Multi-Recorder.” Internal ARL Report not published, 2001.

17 total DMA Pointers are required for initialization. The first 16 Pointers configure the eight groups and the last Pointer is used for the gain/cutoff table. Each group is associated with a pair of Pointers where the first Pointer is for raw data and the second is for FFT'd data. Sending a "NULL" signifies that data or FFT results are not requested for that group. Note that the host can request raw data only, FFT data only, or both by sending appropriate Pointer values. The DMA Pointers must reserve sufficient space in host memory for the data blocks. The size calculation for the data blocks will be covered in the data organization section. The software will only enable groups that have at least one valid pointer to optimize performance and memory requirements on the ADC64cc.

The next piece of information sent by the host is a divisor list. These divisors provide the subsampling rate between groups and enable the multirate operation. The host must send eight values regardless of how many groups are enabled. Values sent for disabled groups will be ignored.

Next, the host sends the Maximum Sample Rate. This value and the group's divisor set the actual data rate for that group. The final parameter sent by the host is the update rate. This is used to specify how often data are sent to the host in Hertz. More frequent transfers to the host reduce memory requirements on the ADC64cc and will allow for increased sample rates if desired. The update rate also equals the FFT resolution.

All data transferred to the host are done through a double buffer system. The current software configures the buffer sizes so that all enabled channels fill their respective buffers at the same time. This allows the ADC64cc to transfer all data blocks at the same point in time. The ADC64 transfers one set of blocks of data to host memory that is half the total host buffer size and interrupts the host with a message, indicating which of the two buffers is ready. The host must release the buffer back to the ADC64cc prior to it needing the storage space, or an overrun condition will occur. The current software signals the host on overrun conditions and terminates.

4.3 Channel Group Configuration

The multirate data operation is achieved via data subsampling. This allows for efficient data acquisition but places some restriction on rates available to individual groups. This section describes how the data are captured and guidelines for selecting group rates, which are compatible. The current version of the software doesn't check for consistent divisor settings.

When the host specifies the Maximum Sample Rate and the Divisor for a group, the effective sample rate for that group is simply

$$\text{Group Sample Rate} = \text{Maximum Sample Rate} / \text{Group Divisor}$$

The ADC64cc must scan through the groups at a higher rate to produce the effective Group Sample Rates required. The Board Sample Rate used on the ADC64cc is then

$$\text{Board Sample Rate} = \text{Maximum Sample Rate} * \text{Number of Active Groups}$$

The algorithm operates by scanning all active groups sequentially at the Board Sample rate but only recording data at the Group Sample Rate to produce multirate data.

The sampled data are grouped into blocks and all block are sent to the host at a common time. To ensure that the data maintains alignment between the groups, the Maximum Sample Rate and the group divisors should all be integerly related. This ensures that all groups start with the first

sample aligned and contain a consistent number of samples per block. If this condition is not meet, the algorithm will still produce data, but care must be taken in its interpretation. The following two examples show how groups are related in the multirate case.

First example:

MSR = 6 Hz Group1 Divisor = 1 Group 2 Divisor = 3 Group 5 Divisor =6

Group 1 samples : S S S S S S S S S S S S ….

Group 2 samples : S S S S S…

Group 5 samples: S S S

|—BLOCK 1—|—BLOCK2—|….

Second example:

MSR = 6 Hz Group1 Divisor = 1 Group 2 Divisor = 3 Group 5 Divisor =5

Group 1 samples : S S S S S S S S S S S S ….

Group 2 samples : S S S S S …

Group 5 samples: S S S

|—BLOCK 1—|—BLOCK2—|….

Note that in the second example that Group 5 achieves is an effective sample rate even if the number of samples in a block are not consistent. While the host could handle this data, it should be avoided. In example one, when the groups are sampled they are always aligned to preserve multigroup multirate phase information.

The other value to extract from the examples is the mux delay between groups. The ADC64cc is running at the Board Sample Rate (producing a base mux delay = 1/Board Sample Rate), and the groups are scanned in order from 1 to 8. Only active groups affect the mux delay (i.e., inactive groups are skipped). Therefore the delay between Group 1 and 2 and Group 2 and 5 is 1-mux delay and between Group 1 and 5 is 2-mux delays.

The final issue in sample rate selection is that if any FFTs are enabled, the Group Sample Rates and update rate must be a power of 2. This ensures that the block sizes are powers of 2.

4.4 Data Organization And Host Buffer Size Requirements

Three types of data are sent to the host: RAW data, FFT'd data and Gain/Cutoff values.

RAW DATA:

The raw data sent to the host is in individual blocks for each group. Within the block, the eight channels are stored as signed 16-bit integers (Little Indian) in an interleaved fashion. The inter-leave pattern is [(ch1,ch2,ch3,ch4,ch5,ch6,ch7,ch8)(ch1,ch2....)]. Note that within a sub-block [ch1–ch8], the samples are simultaneous and each sub-block represents one sample event.

FFT:

The FFT's data sent to the host are in individual blocks for each group. Within the block where the FFT results of all eight channels are stored, a 4-byte IEEE Floats in Little Indian format and have the following pattern. The FFT results are stored sequentially for each group [(FFTch1),

(FFTch2),…..(FFTch8)]. The individual channel results (FFTchX) are complex in nature and have the following pattern (note: R signifies Real and I signifies Imaginary). [R(DC),R(1),R(2), R(3)...R(FFT_SIZE/2)I(FFT_SIZE/2-1)...I(2),I(1)] The indices are FFT bin numbers and must be calculated from sample rate and update rate.

Gain/Cutoff Table:

The current Gain and Cutoff values for the corresponding data blocks are presented in this table. The table lists a value for each group whether active or not. Each entry is a 4-byte integer representing the value programmed into the box. The table is stacked first with the eight gain settings then the eight cutoff settings. A value of 0xFF signifies the box has not been programmed.

The host buffer size requirements are calculated in bytes:

For Raw Data:

2*16*max_sample_rate/(group_divisor[i]*update_rate) =

(2Blocks)*(16 bytes each time group sampled)*(samples per update)

For FFT Results:

2*32*max_sample_rate/(group_divisor[i]*update_rate) =

(2 Blocks)*(32 bytes each time group sampled)*(samples per update)

For Gain/Cutoff Table:

2*4*Maximum Number of Groups *2 = 128

(2 Blocks)*(4 bytes each value)*(8 Groups)*(2 due to gain and cutoff)

4.5 Commands

All commands sent by the host are 4-byte quantities with bytes 1 and 2 specifying the command and bytes 3 and 4 providing command specific data as required.

Start: Issued by host after configuration to reset all data buffers and begin data capture and transfer.

Stop: Issued by host to halt data capture and transfer.

Reset: Issued by host to restart program. Host must resend configuration information.

Terminat: Issued by host to cause ADC64cc code to halt and terminate.

Set Group Gain: Issued by host to specify new Gain for a specific Group.

Byte 4 specifies group number.

Byte 3 contains Gain setting.

Set Group Cutoff: Issued by host to specify new cutoff for a specific group.

Byte 4 specifies group number.

Byte 3 contains cutoff setting.

Buffer Zero Free: Issued by host to indicate buffer available to ADC64cc.

Buffer One Free: Issued by host to indicate buffer available to ADC64cc.

Error Host Overrun: Issued by ADC64cc to indicate Host to slow in releasing buffers.

Error Target Overrun: Issued by ADC64cc to indicate ADC64cc cannot run at sample rate specified and process data.

Error Out of Heap: Issued by ADC64cc to indicate out of memory for local data storage.

4.6 Limitations

The maximum FFT that can be performed is 8192 Point FFT. If FFTs are enabled then no block size can exceed this value.

The local space used for data buffers is 10-K samples per update rate in size. The sum of effective group sample rates cannot exceed this 10-K limit over the update period. For example, if the update rate is 1 Hz, then 2 groups at 5 KHz sample rate is acceptable, but if 4 groups at 5 KHz are desired, then an update rate of 2 Hz must be used.

5. Algorithm Integration Guide

5.1 Background

To minimize the effort of testing new algorithms, the DFT contains all of the necessary software infrastructure to support the core-processing algorithm. These basic modules include a data server, which provides specific data to algorithms and allows control of signal conditioning hardware associated with these data streams. The DFT also provides a communications manager, which maintains a connection to ARL's Gateway system. This communication system allows bidirectional message passing, with the Gateway, to present algorithms results or fuse results at a higher level. The structure of the DFT is such that all algorithms are isolated from one another, which allows follow-on integration efforts to focus solely on the new algorithms.

The DFT can support 56 channels of analog data. These channels are organized into groups of eight, which are clustered into individual signal conditioning boxes. The data groups can have different sample rates to allow multirate processing and minimize data throughput requirements. A group allocation file limits which algorithms can request data from specific groups to ensure algorithm isolation. The algorithms can be allocated multiple groups if more than eight channels of analog data are required.

The current DFT can support up to 10 independent algorithms. The actual number of concurrent algorithms may be lower due to processor loading.

5.2 Configuration And Co-Existence Issues

While the design of the DFT allows the hosted algorithms to operate independently, the system configuration cannot be solely based on unique requirements of each algorithm. The need for compromise arises in the selection of sample rates for each algorithm. The DFT has a single analog-to-digital converter asset and must be configured in a fashion to allow proper operation.

The DFT design allows multirate data acquisition across groups of data, which is transparent to the hosted algorithms. To achieve this, the sample rates between the groups must be integerly related. A second consideration in selecting the sample rates is whether any algorithm will be requesting preprocessed FFT data. If this is the case, the sample rates must also be a power of 2. To select the different samples rates between groups, a group divisor is specified for each group in the system. The group divisor can be a value between 1 and 128. If the relationships previously mentioned are not meet, the system will attempt to run but may eventually fail due to buffering problems.

In configuring the group sample rates, first a master sample rate must be selected. Then all of the group divisors can be selected to match individual algorithm needs. Again if FFT data are to be provided the sample rate must be a power of 2. The next option is the update rate. This determines the data block sizes passed around the system. The value represents how many blocks per second will be used. This parameter also affects the FFT data and represents the bin width in Hertz. For example, given you have an update rate of 4, the data blocks span 250 ms and the FFT resolution is 4 Hz.

The final configuration option is a group allocation table. This table is used to control which groups an algorithm can request and manipulate through the data server. This table serves as a protection device to keep rogue algorithms from affecting others in the system. Multiple algorithms can have common access to a common group of data but they must be aware of the other's actions. In this case, the algorithms are slightly integrated in the sense that they are aware of each other.

The two configuration files are included in Appendix C for reference.

5.3 Template Algorithm For Integration

The template algorithm has a parent algorithm, which is responsible for system initialization and data processing, and a child process, which is responsible for message parsing. New algorithm development requires writing the core processing algorithm and a message parser. The message parser is only required if task-specific commands will be sent by the Gateway system. The overall template algorithm is shown in the following:

Initialize Algorithm

Attach Name to Operating System (OS)

Open Message Queue for Report Transmissions

Open General Parameters Shared Memory.

Create Shared Memory

Create Shared Memory for Use Between Parent and Child Parser

Initialize Shared Memory Elements.

Spawn Child Parser Process

Child Attaches Name to OS

Child Determines Process Identification Number (PID) of Dispatcher

Child Creates Unique Proxy for Dispatcher to Use

Child Registers with Dispatcher with Task Identification Number and Proxy ID

Child waits on Dispatcher to Trigger Proxy Indicating a Message Received

Child Parsers Message and Performs Appropriate Action

Child Waits for Next Proxy Trigger.

Parent Continues

Parent Locates Data Server Task

Parent Determines which Groups are Available to this Task

Parent Allocates Resource to Store Future Data Request

Parent Sets Gain and Cutoff Values in Signal Conditioning Units

Parent Requests a Set of Data from the Data Server

Parent Runs Core-Processing Algorithm

Parent Sends Any Results to Communication Manager via Message Queue

Parent Continues to Request, Process Data and Control Signal-Conditioning Boxes as Required.

5.4 Algorithm Integration Steps

This section provides a checklist for integrating a new algorithm into the DFT.

1) Assign new Task ID to this algorithm.
2) Determine the list of Analog Groups that will be available to this Task ID.
3) Enter Proper Mask Value in the Group Allocation File under this Task ID.
4) Add new entries into IPC_INFO.h to define PROC_NAME_ALGR_X, PROC_NAME_CHILD_ALGR_X, and SOURCE_ALGR_X.
5) Modify template algorithm TASK_ID_ALGR_X entries to match assigned Task ID.
6) Modify template algorithm PROC_NAME_ALGR_X entries to match assigned PROC_NAME.
7) Select Sample Rate Divisors Required for Analog Groups to be used (Note: May need to adjust Maximum Sample Rate and Other Divisors to Accommodate all Algorithms Hosted).
8) Update Flags if Data and or FFT Results will be Available for Groups used by this Algorithm.
9) Merge New Algorithm and Child Parser into Template Algorithm.
10) Compile and Build New Algorithm Executable.
11) Update System Start and Stop Programs to include New Algorithm. Note that Stop Must Run as Root to Kill all DFT Processes.

6. Algorithm Server Interface Protocol

The DFT runs a process referred to as the Algorithm Server. The Algorithm Server is a TCP/IP-based server that is willing to accept a single client connection on port 3081. The server's main functions are to provide data from the DFT, allow remote C2 of Signal Conditioning Boxes used in the data acquisition process and route message reports.

The DFT is comprised of a 64-channel data acquisition system in which the channels are organized into eight groups. The samples within a group are simultaneous while data between groups will contain mux delays due to the specific A/D hardware used in the DFT.

The Algorithm Server operates in conjunction with other processes on the DFT. To allow for proper operation of the DFT, an integration process allocates groups to the Algorithm Server. Once this allocation is specified, the server will process only commands pertaining to enabled groups.

6.1 Command Definitions

The current version of the server supports four basic commands: Get Data, Set Gain, Set Cutoff, and Report LOBs. These will be explained in the following pages.

*Note all values are decimals unless prefaced with 0x for hex representation.

Get Data: A 4-byte command with byte definition is as follows:

Byte 1	Byte 2	Byte 3	Byte 4
40	0x00	Data Type(0 or 1)	Request Mask

Data Type 0 requests raw data and Type 1 requests FFT'd data.

The Request Mask signifies which groups are being requested from the server. A 1 in a bit position (0–7) will case data from the corresponding group (0–7) to be transferred. For example 0x24 will request data from Groups 2 and 5. A Request mask of 0x00 is invalid and will be ignored by the server.

Server Response: For each group requested by the Request Mask, the server transfers one Data Packet. A Data Packet consists of two parts: Header and Data Block. Applications must parse the Header to determine the byte size of the Data Block. Requests for groups which are not allocated to the Server will produce a response of a NULL Header (all fields 0x00) with no Data Block to indicate a configuration error.

The format for the Header and Data Block is defined in Appendix A.

Note: In multiple group data requests the server attempts to send data from groups with one mux delay between groups but this is not guaranteed. Applications should check the time stamp in the header to ensure this is the case when the data is to be processed coherently.

Set Gain: A 4-byte command with byte definition is as follows:

Byte 1	Byte 2	Byte 3	Byte 4
20	0x00	Gain Value	Group Number

Group Number is a value from 0–7 indicating which group to affect.

Gain Value selects the Gain used in the Signal Conditioning Boxes. It is a Hex Pair with the top nibble affecting the low four channels and the low nibble affecting the high four channels. The nibble values map to Gains is as follows:

Constant	Box Gain	Mic Gain
0	1	1
1	1	10
2	10	1
3	10	10
4	100	1
5	100	10
6	1000	1
7	1000	10
A	AGC ch0	AGC ch0
B	AGC ch4	AGC ch4

Note the A or B options enables AGC on the respective group of channels with the reference channel indicated above. Any other valid Gain sent to the group will disable AGC.

Server Response: NONE

Set Cutoff: A 4-byte command with byte definition is as follows:

Byte 1	Byte 2	Byte 3	Byte 4
30	0x00	Cutoff Value	Group Number

Group Number is a value from 0–7 indicating which group to affect.

Cutoff Value selects the cutoff frequency used in the Signal Conditioning Boxes. It is a Hex Pair with the top nibble affecting the low four channels and the low nibble affecting the high four channels. The nibble values map to frequencies is as follows:

0	No Change
1	20 KHz
2	10 KHz
3	5 KHz
4	2.5 KHz
5	1.25 KHz
6	625 Hz
7	312 Hz
8	156 Hz
9	78 Hz
A	39 Hz
B	19 Hz
C	10 Hz

For example cutoff value 0x34 sets low four channels to 5 KHz and top four channels to 2.5 KHz.

Server Response: NONE

REPORT LOBS: A 16–88 byte command with byte definition is as follows:

- All multibyte binary values are sent in Big Indian format (MSB first)
- Message length sent = Number of LOBs *9 +7
- Number of LOB range from 1 to 9
- Azimuth angle reported is measured clockwise from North.

Byte 1	**Byte 2**	**Byte 3**	**Bytes 4–7**
100	0x00	Number of LOBs	Seconds From Midnight Sunday
Bytes 8–16	Bytes 17–25	Bytes 26–79 (if required)	Bytes 80–88
LOB Report 1	LOB Report 2 (if required)	LOB Report 3–8 (if required)	LOB Report 9 (if required)

LOB Report Definition: x=8+LOB Report Number

Byte x->x+1	**Byte x+2->x+3**	**Byte x+4-x+5**
Track Number	Azimuth in Tenths of Degrees	Azimuth Error in Hundredths of Degrees
Byte x=6	**Byte x+7**	**Byte x+8**
0x00	0x00	0x00

7. DFT C2

7.1 Overview

The Current Sensor topology used in Warrior Enhanced Battlespace Sensor (WEBS) testing has several DFT Units communicating with a single Gateway Unit. This document describes the communication requirements and protocol used between these units. A DFT unit requires a single bidirectional communication channel to the Gateway node. The current DFT can operate in two modes: Direct and Routed. In the direct mode of operation, a unique and transparent connection between each DFT and the Gateway is required. This is referred to latter as Point to Point. In the Routed case, the DFT shares a common channel with respect to the Gateway. This case is not transparent, as the radio-net is required to provide additional information covered in the Point to Multipoint section.

The current single DFT/Gateway connection can operate with an effective channel rate of 300 BAUD and with a maximum message packet size of 36 bytes.

General Requirements

- The physical connection between the DFT or Gateway and the radio will be via an RS-232 compatible interface with connector specifications and settings to be determined.

- The maximum single packet size (data plus header) that the interface is willing to accept should be 4096 bytes.
- The interface must pass true binary data with no restrictions on control characters or sequences of them.
- Only Hardware Flow Control can be used at the serial port level.

7.2 Point-to-Point Operation

The DFT in this mode of operation requires a transparent bidirectional communications channel. This channel should provide a virtual RS-232 serial connection between the DFT and Gateway. Note that the data will still be packetized to allow for radio specific processing if required but can be passed blindly if so desired. In this mode of operation, the communications link cannot inject or require additional message traffic. The packet structure follows:

VALUE	DESCRIPTION
0X01	Start of Packet Character 1
0X00	Type of Packet (always 0x00 for data)
Source Address	DFT or Sensor ID of Sender
Number of Bytes (MSB)	MSByte of # of Bytes to Follow in Packet (1)
Number of Bytes (LSB)	LSB Byte of # of Bytes to Follow in Packet (1)
Byte 1	First Byte of Data
Byte 2	Second Byte of Data
……..	Remaining bytes of data (as required)
Checksum	One's Compliment of all bytes in packet excluding the start of packet character (0x01)

Note:

(1) The Number of Bytes to follow does not include the checksum byte

7.3 Point to MultiPoint Operation

The DFT and Gateway in this mode of operation requires a routed communications channel. This allows the DFT or Gateway specifies the address of the receiving party and the radio network delivers the information to the appropriate destination. The radio network is also required to inform the recipient of the source address of the unit, which generated the message. In addition, the radios must provide Acknowledgement/Negative Acknowledgement (ACK/NACK) information to the sending unit. This information will be in the form of control packets, which are distinguishable from data packets. If an ACK or NACK packet is not received after 30 s it will be handled as a NACK response. Note that the DFT or Gateway will not transmit additional data packets to the radio until an ACK or NACK state is reached. This can be used as flow control as long as the previously mentioned response timeout is not exceeded.

In this mode, three different packet structures exist: DATA, ACK, and NACK. The packet definitions follow:

DATA PACKETS

VALUE	DESCRIPTION
0X01	Start of Packet Character 1
0X00	Type of Packet (always 0x00 for data)
Source or Destination Address	If a transmit packet this field is the destination address. If a receive packet then it's the source address.
Number of Bytes (MSB)	MSByte of # of Bytes to Follow in Packet (1)
Number of Bytes (LSB)	LSB Byte of # of Bytes to Follow in Packet (1)
Byte 1	First Byte of Data
Byte2	Second Byte of Data
……..	Remaining bytes of data (as required)
Checksum	One's compliment of all bytes in packet excluding the start of packet character (0x01)

Note:

(1) The Number of Bytes to follow does not include the checksum byte.

ACK PACKET

0x01	Start of Packet Character 1
0x80	Type of Packet (0x80 for status)
0x00	ACK
0x7F	Checksum

NACK PACKET

0x01	Start of Packet Character 1
0x80	Type of Packet (0x80 for status)
0x01	NACK
0x7E	Checksum

7.4 Communication Related Messages: (Task 0)

*Note 1: All values are decimals unless prefaced with 0x for hex representation.

*Note 2: Byte 1 is Command in low nibble with Task ID in high nibble.

*Note 3: All Commands and Response definitions to follow form the data portion of the previously defined data packets.

Connection Request (login): A 2-byte command sent by sensor with Byte 2 representing sensor type.

Byte 1	Byte 2
0x04	0x0A

Connection Request Acknowledge: Sent by controlling unit to enable communications.

Byte 1
0x05

Connection Close (logout): Sent by controlling unit to close connection.

Byte 1
0x01

Connection Close ACK (logout ACK): Sent by DFT to ACK logout.

Byte 1
0x07

Re-Connect: Sent by controlling unit to force DFT to close then open connection.

Byte 1
0x06

Suspend LOB Reports: Sent by controlling unit to disable LOBs.

Byte 1
0x0A

Resume LOB Reports: Sent by controlling unit to enable LOBs.

Byte 1
0x0B

Connect: Sent by controlling unit to force login from DFT.

Byte 1
0x02

Reboot System: A 1-byte command with byte definition as follows:

Byte 1
0x03

Recorder Response: NONE

Reconfigure: A 14-byte command with byte definition is as follows:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x08	Sample Rate (LSB)	Sample Rate (MSB)	Update Rate	Active Group Mask	Active FFT Mask	GD0
Byte 8	**Byte 9**	**Byte 10**	**Byte 11**	**Byte 12**	**Byte 13**	**Byte 14**
GD 1	GD2	GD3	GD4	GD5	GD6	GD7

Recorder Response: Echo Configuration Parameters or 14-byte NULL message to indicate error in reconfiguration attempt.

GD0-Gd7 specify group divisor values

Active Group Mask: Bits 0–7 enable groups 0–7 for acquisition.

Active FFT Mask: Bits 0–7 enable FFT results for groups 0–7.

Set Date and Time: A 17-byte command with byte definition is as follows (Note: All date and time values are sent as ACSII values):

Byte 1	**Bytes 2–5**	**Byte 6–7**	**Byte 8–9**	**Byte 10–11**	**Byte 12–13**	**Byte 14**
0x0C	Year YYYY	Month MM	Day DD	Hour HH	Minute mm	0x2E (dot)
Byte 15–16	**Byte 17**					
Seconds SS	0x20 (space)					

Recorder Response: Current date and time. Note if a Set Date and Time Command is issued with all date and time values set to 0, the current time setting will be reported without altering the setting.

Current Date and Time: A 17-byte response with byte definition is as follows. Note all date and time values are sent as ACSII values.

Byte 1	**Bytes 2–5**	**Byte 6–7**	**Byte 8–9**	**Byte 10–11**	**Byte 12–13**	**Byte 14**
0x0C	Year YYYY	Month MM	Day DD	Hour HH	Minute mm	0x2E (dot)
Byte 15–16	**Byte 17**					
Seconds SS	0x20 (space)					

Global Algorithm Delay: A 2-byte command to specify delay for all hosted algorithms. This delay is the min standoff time between processing intervals.

Byte 1	**Byte 2**
0x09	Delay value (s)

Bad Command Received: Sent by DFT to indicate a bad command received by task 0.

Byte 1
0x0F

7.5 Recorder Command Definitions: (Task 15)

The DFT runs a process referred to as Recorder. The Recorder's role in the DFT is to allow basic system C2 and data archiving. The Recorder is intended for use in both stand-alone operations when the DFT is only a recording asset and when the DFT is operating in a networked sensor environment. The current data link layer used is a DDR-96 radio modem made by Pacific Crest Corporation. This protocol document specifies the message protocol as passed to a protocol handler for the radio. That is, all commands and responses specified here are data fields in the radio protocol.

Because the Recorder process may be operating in stand-alone, it has full capability to access all groups of data in the DFT. Care must be taken when operating in conjunction with algorithm testing as not to interfere with settings that an algorithm may have imposed on a specific data

group. In this supportive role during algorithm evaluation, the Recorder provides data archiving that is transparent to the test algorithm.

The command structure of the Recorder is designed to allow the remote-controlling program to reconfigure the data acquisition parameters, change settings for Signal Conditioning Boxes, and retrieve disk statistics from the DFT.

As part of the DFT, the Recorder process operates in several phases. The first phase is an Initialization Phase. This step allows a specific DFT to become active on the sensor field. The DFT will repeatedly attempt to Connect and will be unable to transmit any information back to the Controlling Unit until it has received an acknowledgement to the connection request. This phase is entered on power-up and after receiving a Reboot command. Note that prior to connecting to the controlling unit, the DFT will still receive and process commands—only the responses are disabled. After receiving a connection acknowledgement, the DFT will receive from and respond to the controlling unit as described in this document.

Data, which are recorded by the Recorder process, is stored in one file per group. Each time the recording process is started and stopped, a new set of files will be created. The data are recorded as sequential data packets consisting of a header and data block. All data packets have an identical form and are described in Appendix A. Note that if a request is made to record a group, which is not enabled, the Recorder response will indicate this and proceed.

The Recorder is designed to degrade gracefully when the system loads do not allow for continuous data streaming to disk. In this event, all data packets are guaranteed to be blocked contiguously but not all packets may be recorded. This will cause momentary dropouts in the data but the data file still maintains its integrity, and the dropout is fully detectable.

Record to Jazz: A 2-byte command with byte definition is as follows:

Byte 1	Byte 2
0xF0	Request Status

The Request Mask signifies which groups are being requested for recording. A “1” in a bit position (0–7) will cause data from the corresponding group (0–7) to be recorded. For example, 0x24 will request that data from Groups 2 and 5 be recorded.

Recorder Response: One of the following (See Response Section for Specifics)

Recording to Jazz

Error Invalid Data

Error Already Recording

Error Out of Resources

Error Drive Not Responding

Record to Hard Drive: A 2-byte command with byte definition is as follows:

Byte 1	Byte 2
0xF1	Recording Status

The Request Mask signifies which groups are being requested for recording. A “1” in a bit position (0–7) will cause data from the corresponding group (0–7) to be recorded. For example, 0x24 will request data from Groups 2 and 5 be recorded.

Recorder Response: One of the following (See Response Section for Specifics)

Recording to Hard Drive

Error Invalid Data

Error Already Recording

Error Out of Resources

Error Drive Not Responding

Set Gain: A 3-byte command with byte definition is as follows:

Byte 1	**Byte 2**	**Byte 3**
0xF5	Group Number	Gain Value

Group Number is a value from 0–7 indicating which group to affect.

Gain Value selects the Gain used in the Signal Conditioning Boxes. It is a Hex Pair with the top nibble affecting the low four channels and the low nibble affecting the high four channels. The nibble values map to Gains is as follows:

Constant	Box Gain	Mic Gain
0	1	1
1	1	10
2	10	1
3	10	10
4	100	1
5	100	10
6	1000	1
7	1000	10
A	AGC ch0	AGC ch0
B	AGC ch4	AGC ch4

Note: The A or B options enables AGC on respective group of channels with the reference indicated above. Any other valid Gain sent to the group will disable AGC.

Recorder Response: Gain Changed

Set Cutoff: A 3-byte command with byte definition is as follows:

Byte 1	**Byte 2**	**Byte 3**
0xF6	Group Number	Cutoff Value

Group Number is a value from 0–7 indicating which group to affect.

Cutoff Value selects the Cutoff frequency used in the Signal Conditioning Boxes. It is a Hex Pair with the top nibble affecting the low four channels and the low nibble affecting the high four channels. The nibble values map to frequencies is as follows:

0	No Change
1	20 kHz
2	10 kHz
3	5 kHz
4	2.5 kHz
5	1.25 kHz
6	625 Hz
7	312 Hz
8	156 Hz
9	78 Hz
A	39 Hz
B	19 Hz
C	10 Hz

For example, Cutoff value 0x34 sets low four channels to 5 KHz and top four channels to 2.5 KHz.

Recorder Response: Cutoff Changed

Eject Jazz Drive: A 1-byte command with byte definition is as follows:

***Not Supported in Rev 1.0

Byte 1
0xF4

Recorder Response: Error Drive Not Responding (Rev 1.0)

Jazz Ejected in latter version

Stop Recording: A 1-byte command with byte definition is as follows:

Byte 1
0xF7

Recorder Response: One of the following

Recorder Stopped

Error Already Stopped

Report Available Hard Drive Space: A 1-byte command with byte definition is as follows:

Byte 1
0xF3

Recorder Response: Report Hard Drive Space

Report Available Jazz Drive Space: A 1-byte command with byte definition is as follows:

Byte 1
0xF2

Recorder Response: Report Jazz Drive Space

Report Skip Count: A 1-byte command with byte definition is as follows:

Byte 1
0xF8

Recorder Response: Report Skip Count

RESPONSE DEFINITIONS:

*Note 1: All values are decimals unless prefaced with 0x for hex representation.

*Note 2: Byte 1 is command in low nibble with Task ID in high nibble.

Recording to Jazz: A 2-byte response with byte definition is as follows:

Byte 1	Byte 2
0xF0	Request Status

Recording Status: Each bit set to “1” indicates that its corresponding group is recording. This may differ from the recording request if a group is disabled in configuration

Recording to Hard Drive: A 2-byte response with byte definition is as follows:

Byte 1	Byte 2
0xF1	Recording Status

Recording Status: Each bit set to “1” indicates that its corresponding group is recording. This may differ from recording request if a group is disabled in configuration.

Report Disk Space Hard Drive: A 5-byte response with byte definition is as follows:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0xF3	Disk Space (LSB)	—	—	Disk Space (MSB)

*Note bytes 2–5 represent disk space available in Mbytes as an unsigned integer in Little Indian Format.

Report Disk Space Jazz Drive: A 5-byte response with byte definition is as follows

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0xF2	Disk Space (LSB)	—	—	Disk Space (MSB)

*Note bytes 2–5 represent disk space available in MB as an unsigned integer in Little Indian Format.

Recorder Stopped: A 1-byte response with byte definition is as follows:

Byte 1
0xFA

Gain Changed: A 1-byte response with byte definition is as follows:

Byte 1
0xFB

Cutoff Changed: A 1-byte response with byte definition is as follows:

Byte 1
0xFC

Report Skip Count: A 5-byte response with byte definition as follows:

*Note 1: Bytes 2–5 represent Skip Count as an unsigned integer in Little Indian Format.

*Note 2: Skip count should be decremented by 1 on C2 unit.

Error Responses: A 2-byte command with byte definition is as follows:

Byte 1	Byte 2
0xFF	Error Code

Error Codes:

Configuration Error	0
Drive Not Responding	1
Out of Resources	2
File Write Error	3
Already Recording	4
Already Stopped	5
Unknown Command	6
Invalid Data	7

Jazz Ejected: A 1-byte response with byte definition is as follows:

Byte 1
0xF4

Navigation and Status Commands: (Task 13)

Status Request: A 1-byte command with byte definition is as follows:

Byte 1
0xD0

NAV_Status Response: Status Message

Status Report: A 4-byte response with byte definition is as follows:

Byte 1	Byte 1	Byte 3	Byte 4
0xD0	Battery Voltage	Internal Temp	Fan|Heat Status

Internal temperature is in degrees Celsius.

Byte 4 has top nibble indicating fan status and low nibble indicating heat status with 0xF = ON and 0x0 = OFF.

Position Request: A 1-byte command with byte definition is as follows:

Byte 1
0xD1

NAV_Status Response: Position Message

Position Report: A 13-byte response with byte definition is as follows:

Byte 1	Byte 2–5	Byte 6–9	Byte 10–13
0xD1	Latitude	Longitude	Altitude

Notes: Position reported in WGS-84 coordinates with ellipsoidal height.

Lat, Lon, and Alt represented as 4-byte IEEE Floats (MSB).

Lat and Lon in decimal degrees, alt in meters

Panic Shutdown: A 2-byte message with byte definition is as follows:

Byte 1	Byte 2
0xDF	Time Until Shutdown (sec)

Note: This message indicates that the Unit's system power has dropped to 23 V. The shutdown process is 60 s with a message every second.

7.6 Algorithm Server LOB Report: (Task 14)

REPORT LOBS: A 16–88 byte command with byte definition is as follows:

- All multibyte binary values are sent in Big Indian Format (MSB first)
- Message length sent = Number of LOBs *9 +7
- Number of LOB range from 1-9
- Azimuth angle reported is measured clockwise from North.

Byte 1	Byte 2	Byte 3	Bytes 4–7
0xE4	Sensor ID	Number of LOBs	Seconds From Midnight Sunday
Bytes 8-16	Bytes 17-25	…	Bytes 80-88
LOB Report 1	LOB Report 2 (if required)	…	LOB Report 9 (if required)

LOB Report Definition: x=8+LOB Report Number.

Byte x->x+1	Byte x+2->x+3	Byte x+4-x+5
Track Number	Azimuth in Tenths of Degrees	Azimuth Error in Hundredths of Degrees
Byte x=6	Byte x+7	Byte x+8
0x00	0x00	0x00

7.7 ARL Acoustic Algorithm LOB Report: (Task 1)

REPORT LOBS: A 8–29 byte command with byte definition is as follows:

- All multibyte binary values are sent in Big Indian Format (MSB first)
- Message length sent = Number of LOBs *3 +5
- Number of LOB range from 1–8
- Azimuth angle reported is measured clockwise from North.

Byte 1	Byte 2–5	Byte 6–8	Bytes 9–11
0x1B	Number of Lob's and Report Time	LOB Report 1	LOB Report 2 (if required)
Bytes12-14	Bytes 15-17	…	Bytes 27-29
LOB Report 3 (if required)	LOB Report 4 (if required)	…	LOB Report 8 (if required)

LOB Report Definition:

Byte x	Byte x+1	Byte x+2
Azimuth in (360/256) degrees per unit	Track Number Top Nibble Azimuth Error Low Nibble	Target ID Top 5 Bits Target Confidence Low 3 Bits

Notes:

- Byte 2 has the Number of (LOBs –1) packed in the top 3 bits. This value ranges from 0–7 but maps to 1–8.
- Byte 2 has Report Time packed in the low 29 bits. This value is an unsigned integer (MSB) and units are in 100ths of seconds since midnight Sunday. To get milliseconds from Sunday multiply by 10.
- Azimuth error is table look up TBD.

Pan Tilt Enable: A 1-byte command with byte definition is as follows:

Byte 1
0x12

Pan Tilt Disable: A 1-byte command with byte definition is as follows:

Byte 1
0x13

Pan Tilt Cue: A 3-byte command with byte definition is as follows:

Byte 1	Byte 2–3
0x14	Cue Angle Clockwise from North (0–360) (LSB)

7.8 ARL Magnetic Report and Commands: (Task 3)

Magnetic Detection Report: A 6-byte report with byte definition is as follows:

Byte 1	Byte 2–5	Byte 6
0x31	Time	Type (1 or 2)

Notes: Time in Seconds from Midnight Sunday (MSB Integer)

Type 1 detection is single sided; waveform Type 2 is N-wave.

Set Magnetic Threshold : A 5-byte command with byte definition is as follows:

Byte 1	Byte 2–5
0x33	Threshold

Notes: Threshold represented as 4-byte IEEE Floats (MSB) and in units of A/D counts (0 – 2^15).

Set Magnetic Filter Gate : A 5-byte command with byte definition is as follows:

Byte 1	Byte 2–5
0x33	Gate Value

Notes: Gate is a low pass filter coefficient represented as 4-byte IEEE Float (MSB).

7.9 ARL Sniper Report and Commands: (Task 4)

REPORT LOBS: A 7- or 9-byte command with byte definition is as follows:

- All multibyte binary values are sent in Big Indian Format (MSB first)
- Number of LOB range from 1 to 2
- Azimuth angle reported is measured clockwise from North.

Byte 1	Byte 2–5	Byte 6–7	Bytes 8–9
0x4A	Number of LOBs and Report Time	LOB Report 1	LOB Report 2 (if required)

LOB Report Definition:

Byte x	Byte x+1
Azimuth in (360/256) degrees per unit	Azimuth Error Low In degrees

Notes:

- Byte 2 has the Number of (LOBs –1) packed in the top 3 bits. This value ranges from 0 to 1 but maps from 1 to 2.
- Byte 2 has Report Time packed in the low 29 bits. This value is an unsigned integer (MSB) and units are in 100ths of seconds since midnight Sunday. To get milliseconds from Sunday multiply by 10.

7.10 Frame Grabber Reports and Commands: (Task 5)

- All multibyte binary values are sent in Big Indian Format (MSB first)

Grab Frame: A 1-byte command with byte definition is as follows:

Byte 1
0x51

Response: Frame Report

Frame Report: A sequence of three types of packets:

1. START: Packet specifying the size and number of data packets to follow.
2. DATA: Sequence of data packets.
3. LAST: Terminating Packet of size different than previous stream.

START Grab Frame: A 5-byte packet with byte definition is as follows:

Byte 1	Byte 2 to 3	Byte 3 to 4
0x51	Number of Packets	Data Packet size in bytes

DATA Grab Frame: A 5-byte packet with byte definition is as follows:

Byte 1	Byte 2 to 3	Byte 3 to 4	Byte 5-N
0x52	Packet number	Packet Size(N)	Data

LAST Grab Frame: A 5-byte packet with byte definition is as follows:

Byte 1	Byte 2 to 3	Byte 3 to 4	Byte 5-N
0x53	Packet number	Packet Size(N)	Data

Enable Grab Frame: A 1-byte command with byte definition is as follows:

Byte 1
0x52

Disable Grab Frame: A 1-byte command with byte definition is as follows:

Byte 1
0x53

---

8. TCP/IP Monitor Interface Protocol

The DFT runs a process referred to as the TCP/IP Monitor. The Monitor is a TCP/IP-based server that is willing to accept a single client connection on port 3080. The monitor's sole mission is to provide a detailed look at the data, which the DFT is processing on a noninterfering basis. This is designed for monitoring of data during data collection or quick-look analysis of data being processed by a resident algorithm.

The monitor doesn't have the ability to affect any actions relative to recording or processing but has unlimited access to all channels being acquired by the DFT.

The monitor operates by receiving single commands and replying with appropriate information.

8.1 Command Definitions

The current version of the monitor supports a single command, Get Data.

*Note all values are decimal unless prefaced with 0x for hex representation.

Get Data: A 3-byte ASCII string with byte definition is as follows:

Byte 1	**Byte 2**	**Byte 3**
Group Number	(Space)	Data Type(0 or 1)

Data Type 0 requests raw data and Type 1 requests FFT'd data.

Group Number specifies which group to report on.

Monitor Response: The Monitor transfers one Data Packet. A Data Packet consists of two parts: Header and Data Block. Applications must parse the Header to determine the byte size of the Data Block. Requests for groups which are not enabled will produce a response of a NULL Header (all fields 0x00) with no Data Block to indicate a configuration error.

The format for the Header and Data Block is defined in Appendix A.

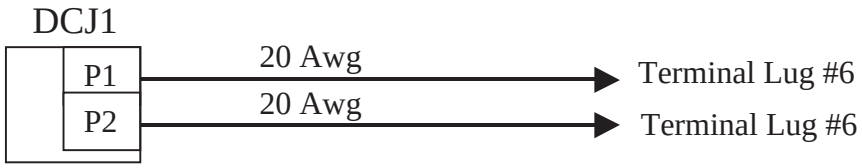
For example, the command "4 0" sent to the Monitor requests Raw Data from group 4.

Appendix A. DFT Drawings and Schematics

Items Included:
- Internal Cable Drawings
- External Cable Drawings
- Side Panel Assembly Drawing
- Base Plate Terminal Strip Detail
- Climate Control Schematic
- DC/DC Converter Schematic
- Unit Interconnect Drawing

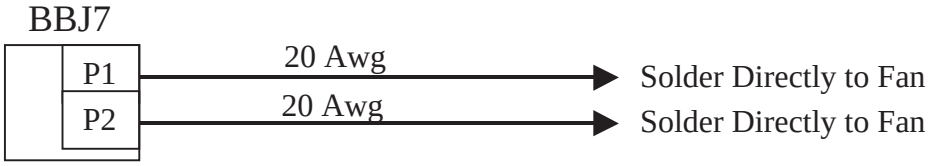
Internal Cable Drawings

Radio Power Cable:



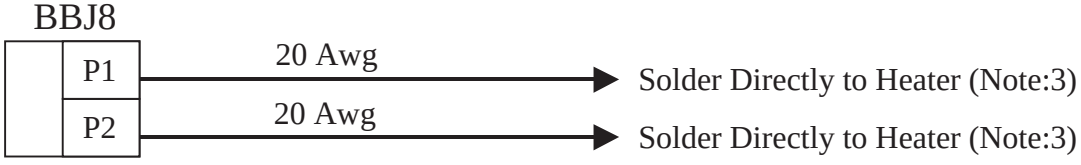
*Note 1: DCJ1 is supplied with Pacific Crest Radio.
*Note 2: Overall cable length is 30 in.

Fan Power Cable:



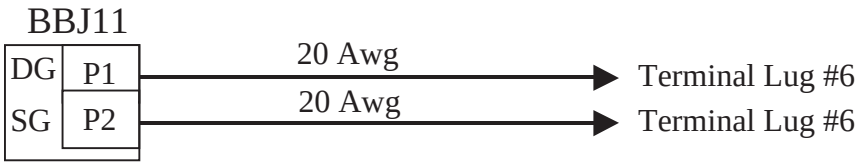
*Note 1: BBJ7 is Molex Connector Part Number 50-84-1020.
*Note 2: Overall cable length is 24 in.

Heater Power Cable:



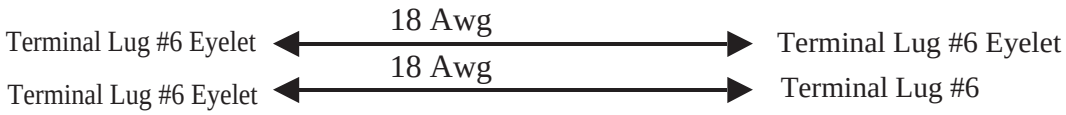
*Note 1: BBJ8 is Molex Connector Part Number 50-84-1020.
*Note 2: Overall Cable length is 24 in.
*Note 3: On Installation trim cable length to restrain heater from rotating.

Digital and Shield Ground Cable:



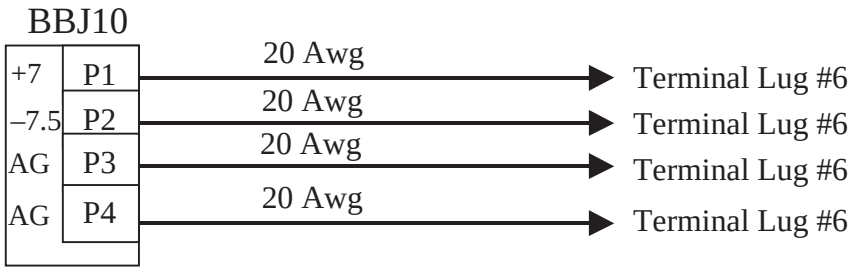
*Note 1: BBJ11 is Molex Connector Part Number 50-84-1020.
*Note 2: Overall cable length is 12 in.

DC/DC Converter Assembly Power Cable:



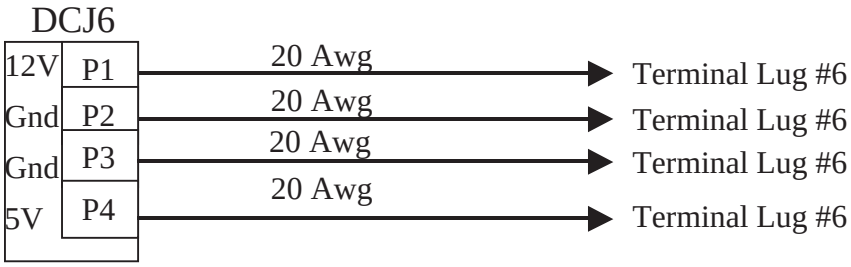
*Note 1: Overall cable length is 24 in.

Analog Power Cable:



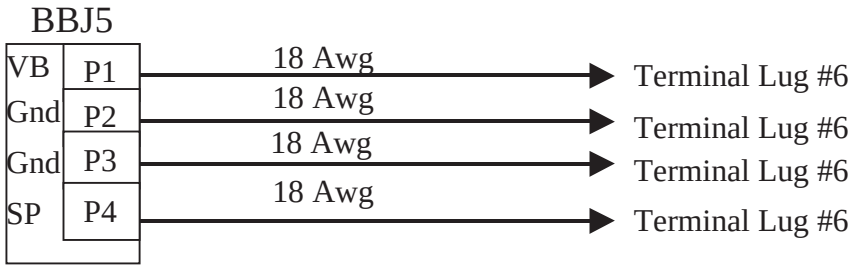
*Note 1: BBJ10 is Molex Connector Part Number 50-84-1040.
*Note 2: Overall cable length is 30 in.

Jazz Power Cable:



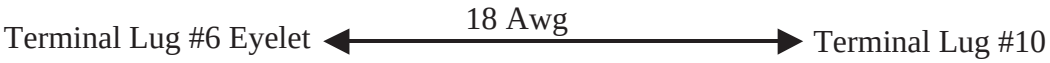
*Note 1: DCJ6 is Molex Connector Part Number 1-480424-0.
*Note 2: Overall cable length is 30 in.

Main Power and Thermostat Cable:



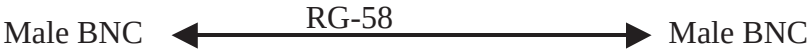
*Note 1: BBJ5 is Molex Connector Part Number 50-84-1040.
*Note 2: Overall cable length is 12 in.

Chassis/Plate Ground Cable:



*Note 1: Overall cable length is 3 in.

Radio Antenna Cable:



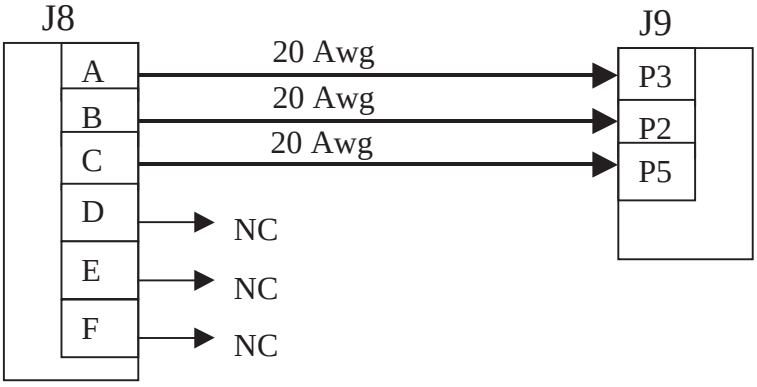
*Note 1: Overall cable length is 12 in.

Jazz Ribbon Cable:

*Note 1: Overall cable length is 7 in.

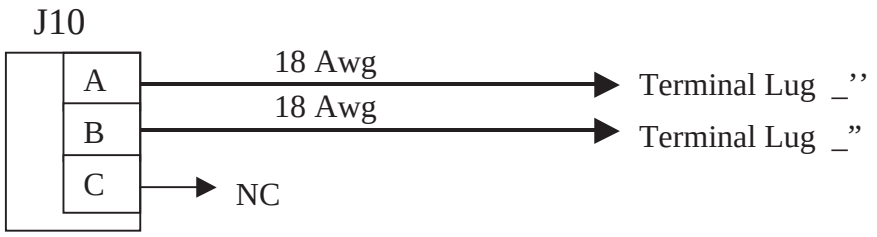
External Cable Drawings

Diagnostic Cable:



*Note 1: J8 is Mil Connector AMPHENOL GB62GB56TG10-6P.
*Note 2: J9 is DB-9 Female Connector.
*Note 3: Overall cable length is 6 ft of standard computer cable.

Power Cable:



*Note 1: J10 is Mil Connector Part Number MS3476W-12-3S.
*Note 2: Overall cable length is 6 ft and twisted.

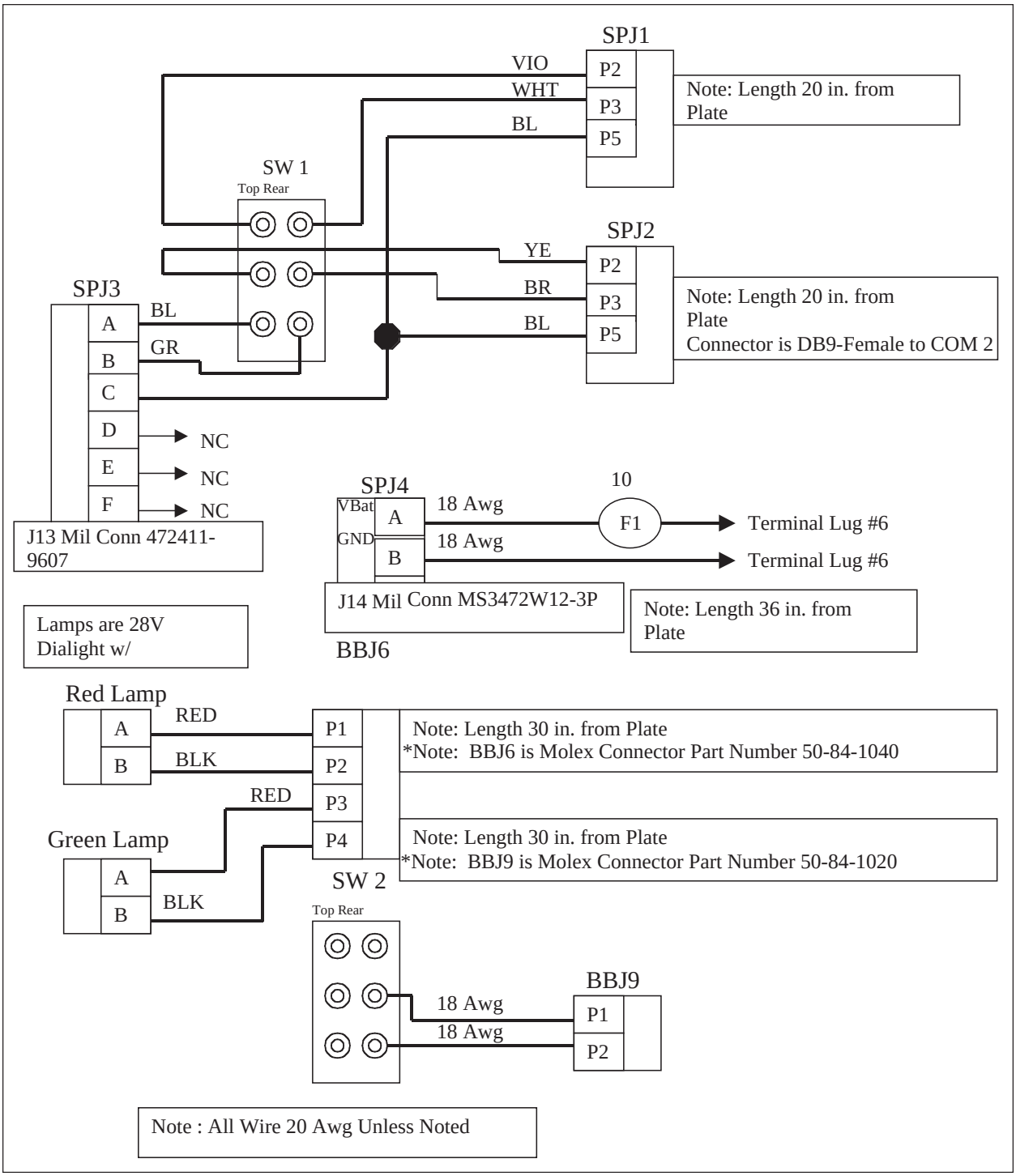


Figure A-1. Side Panel Assembly Drawing

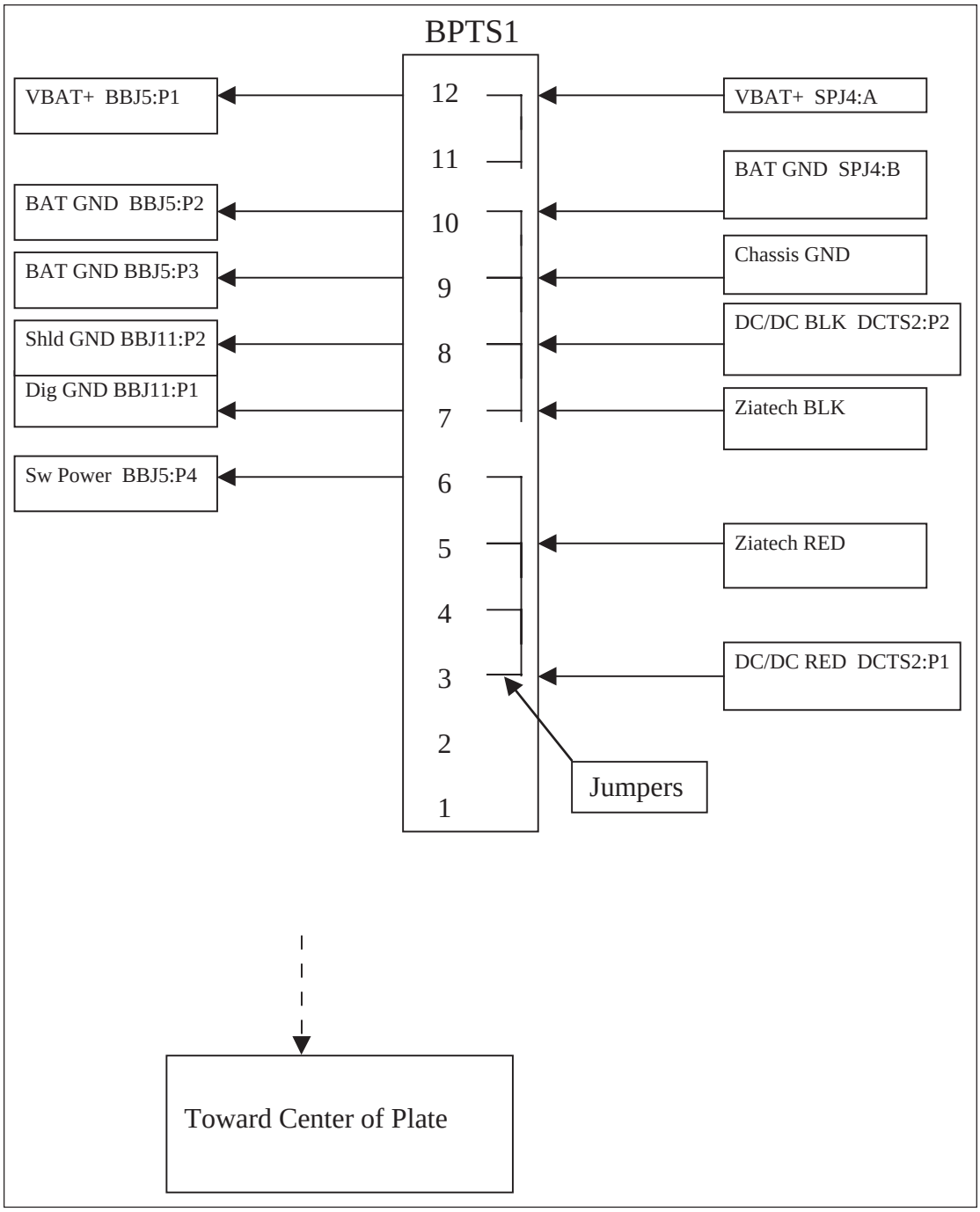


Figure A-2. Base Plate Terminal Strip Detail

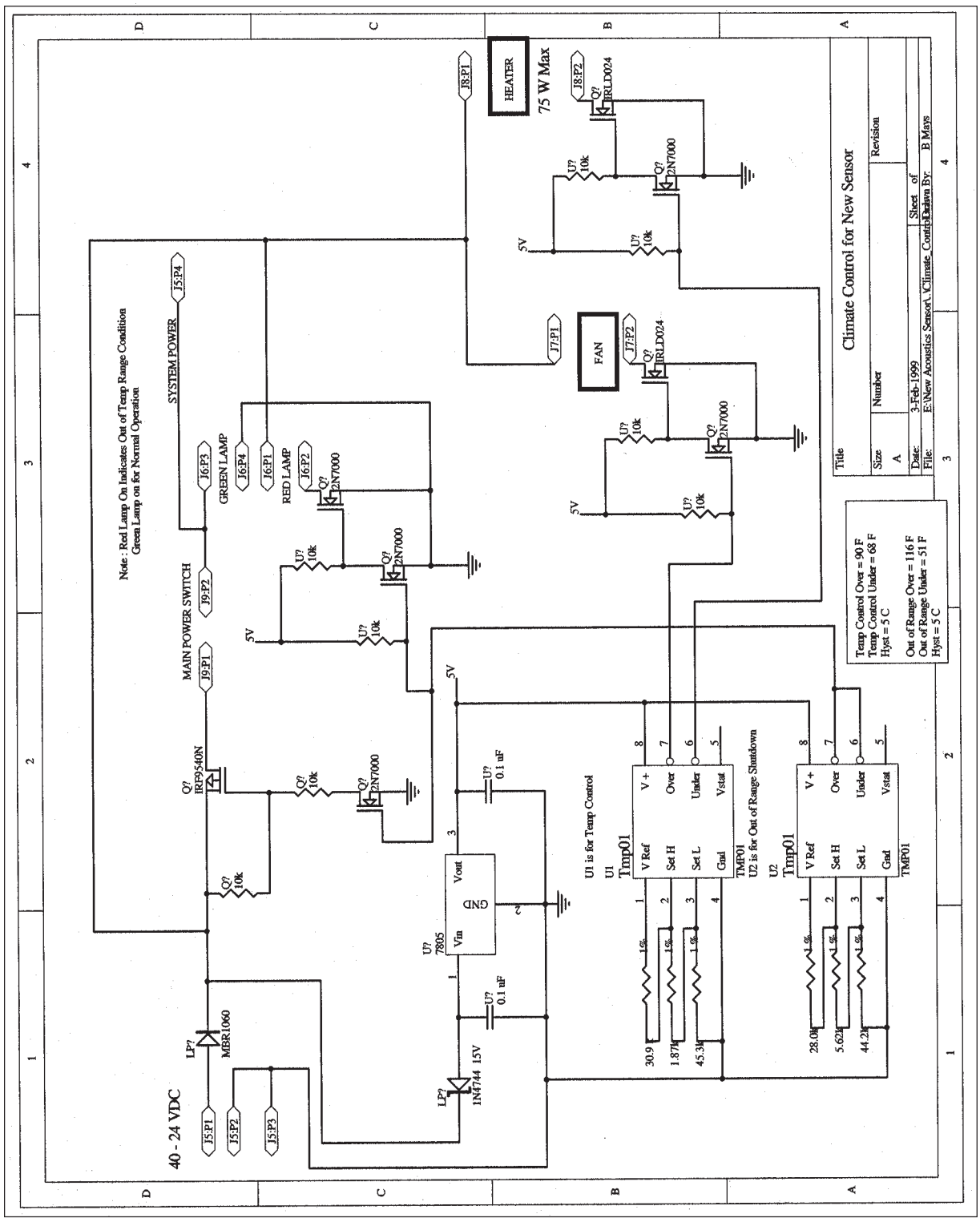


Figure A-3. Climate Control Schematic.

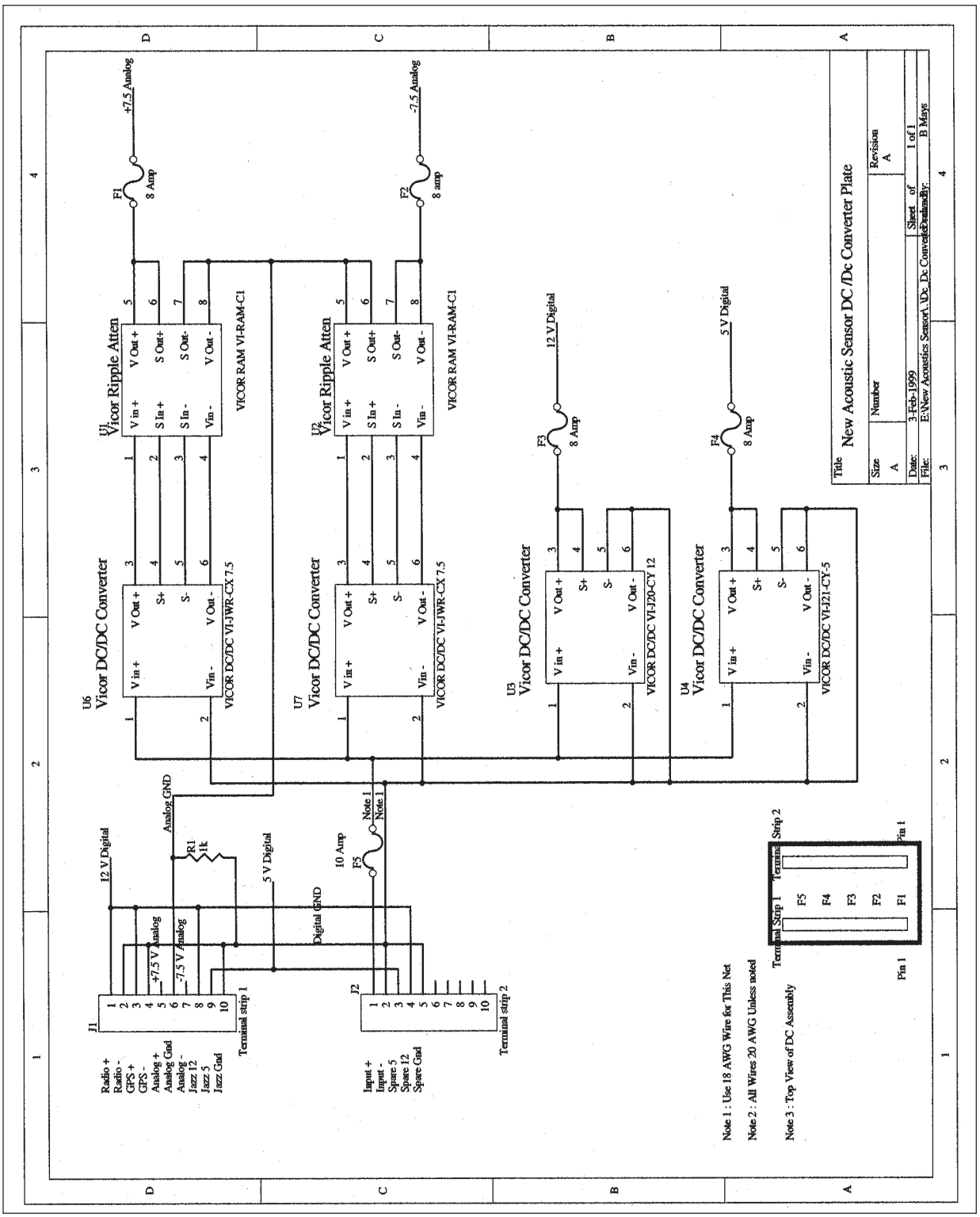


Figure A-4. DC to DC Converter Schematic.

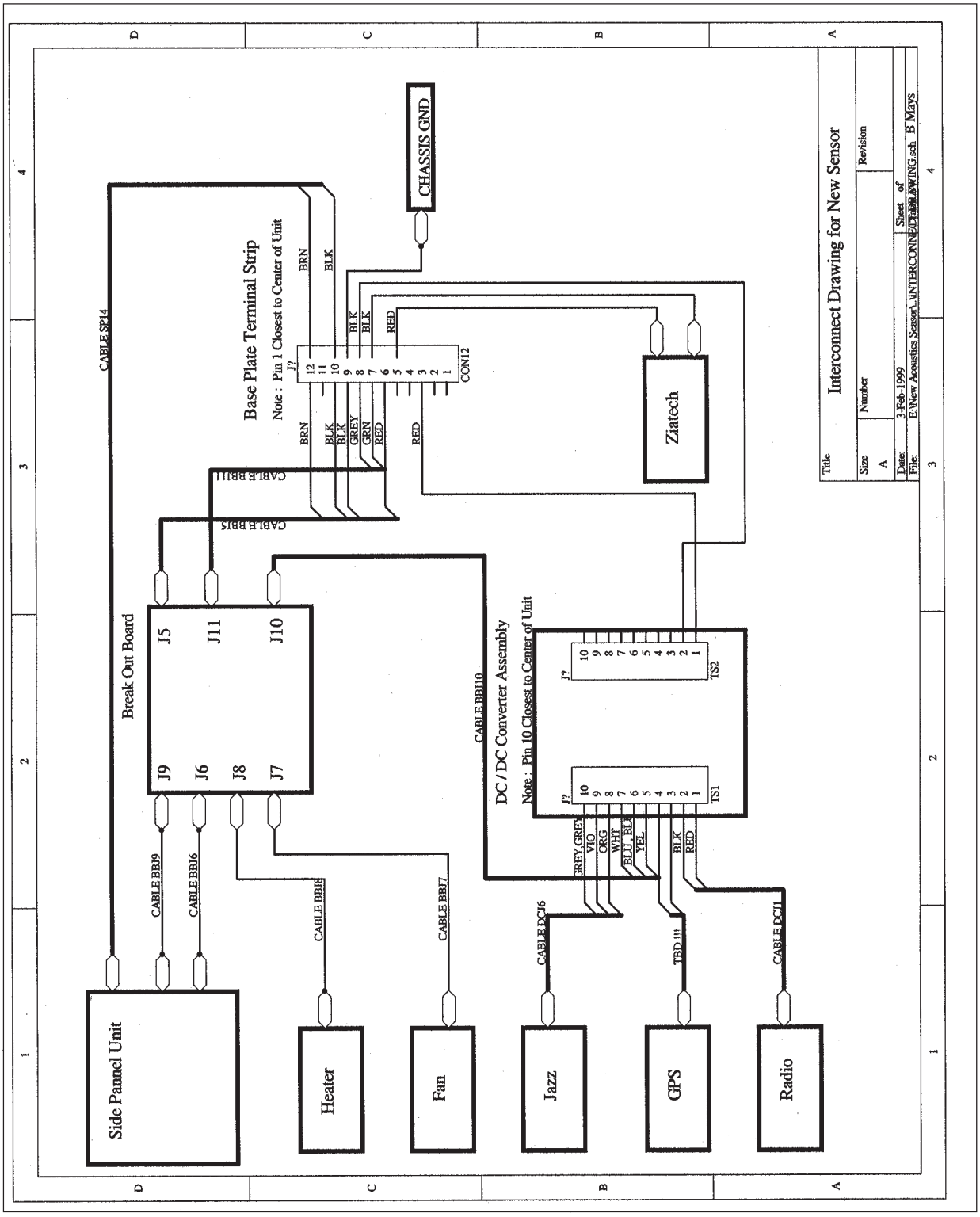


Figure A-5. DFT Internal Interconnect Drawing.

Appendix B. DFT Data Format Specifications.

Every data packet sent to an application will contain a Header section and a Data section. The Header section comes with a fixed length of 40 bytes long. Depending on the sampling rate, the length of the Data section is varied.

HEADER: Each field is a 16-bit unsigned integer in Little Indian Format (LSB first).

Fields	Name	Description	Value
1	Header ID	Distinguish different kind of data	2000 = acoustic data 0 = NULL data
2	Data type ID	Distinguish different type of data	0 = RAW data 1 = FFT data
3	Year	Years since 1900	99-
4	Month		1 – 12
5	Day		1 – 31
6	Hour	Hours after Midnight	0 – 23
7	Minute	Minutes after hour	0 – 59
8	Second	Seconds after Minute	0 – 59
9	Milli-Second	Milli seconds after second	0-999
10	Sensor/Group ID	Sensor ID / “Signal Condition” box ID	Sensor ID or w/ Group ID
11	Sampling Rate	Number of samples per second	2048 = Maximum sampling rate
12	Update Rate	How often data is being transfer to Host in Hertz	
13	Gain	Amplifier Gain setting for “Signal Condition” box	Refer to Appendix B for gain table
14	Cutoff	Cutoff frequency setting for “Signal Condition” box	Refer to Appendix B for cutoff table
15	Full scale voltage		+/- 5 volts
16	Number of channels Per group	A/D channels	64 A/D channels which split into 8 channels per group
17	Days Since Sunday	Number of Days From Sunday	0-6
18	Board Sampling Rate	Used for Calculating Mux delays across groups	
19	Active Group Mask	Bit 0-7 Set Indicate Corresponding Group Active	
20	Frame Counter	Rolling counter used to detect lost frames	1 - 65535

Note: The size of each data packet can be calculated as followed:

RAW data packet (in bytes):

(((Header.Sampling Rate * Header.Number of Channels) / Header.Update Rate) * 2) + 40

FFT data packet (in bytes):

(((Header.Sampling Rate * Header.Number of Channels) / Header.Update Rate) * 4) + 40

Mux Delay Calculation:

In applications using multi-group data; the mux delay between groups is calculated using Header Fields 18 and 19. The groups are scanned at the board sample rate making the base mux delay 1/ board sample rate. The total delay is the base delay * the number of active groups between groups of interest which is shown in Field 19. Note that only active groups add to delay calculations. For example, an Active group Mask of 0x0B, groups 0, 1, and 3 are active. The delay between 0 and 3 is 2*base mux delay; and between 0 and 1 and 1 and 3 the delay is 1*base mux delay.

Data Block Specifications:

RAW DATA:

The raw data sent to the host are in individual blocks for each group. Within the block the eight channels are stored as signed 16-bit integers (Little Indian Format) in an interleaved fashion. The interleave pattern is [[ch1,ch2,ch3,ch4,ch5,ch6,ch7,ch8][ch1,ch2....]]. Note that within a sub-block [ch1–ch8] the samples are simultaneous and each sub-block represents one sample event.

FFT'd:

The raw data are windowed with a Hanning Window and then an FFT with resolution equal to the data update rate is performed. The FFT's data sent to the host is in individual blocks for each group. Within the block, the FFT results of all eight channels are stored in a 4-byte IEEE Floats in Little Indian Format and have the following pattern. The FFT results are stored sequentially for each group [[FFTch1],[FFTch2],…..[FFTch8]]. The individual channel results [FFTchX] are complex in nature and have the following pattern, note R signifies Real and I Imaginary. [R(DC),R(1),R(2),R(3)...R(FFT_SIZE/2)I(FFT_SIZE/2-1)...I(2),I(1)] The indices are FFT bin numbers and must be calculated from sample rate and update rate.

Appendix C. Gain Table for Signal Conditioning Boxes

Gain Value selects the Gain used in the Signal Conditioning Boxes. It is a Hex Pair with the top nibble affecting the low four channels and the low nibble affecting the high four channels. The nibble values map to Gains as follows:

Constant	Box Gain	Mic Gain
0	1	1
1	1	10
2	10	1
3	10	10
4	100	1
5	100	10
6	1000	1
7	1000	10
A	AGC ch0	AGC ch0
B	AGC ch4	AGC ch4

Note: The A or B options enables AGC on respective group of Channels with the reference channel indicated above. Any other valid Gain sent to the group will disable AGC.

Cutoff Values for Signal Conditioning Boxes

Cutoff Value selects the Cutoff frequency used in the Signal Conditioning Boxes. It is a Hex Pair with the top nibble affecting the low four channels and the low nibble affecting the high four channels. The nibble values map to frequencies as follows:

0	No Change
1	20 KHz
2	10 KHz
3	5 KHz
4	2.5 KHz
5	1.25 KHz
6	625 Hz
7	312 Hz
8	156 Hz
9	78 Hz
A	39 Hz
B	19 Hz
C	10 Hz

Appendix D. Configuration File Specification

The first file is the Group allocation file a one in any bit position represents that the group is available to that task.

```
@GROUP_ALLOCATION
0x0F        TASK_ID_ALGR_SERVER
0x01        TASK_ID_ALGR_1
0x04        TASK_ID_ALGR_2
0x00        TASK_ID_ALGR_3
0x00        TASK_ID_ALGR_4
0x00        TASK_ID_ALGR_5
0x00        TASK_ID_ALGR_6
0x00        TASK_ID_ALGR_7
0x00        TASK_ID_ALGR_8
0x00        TASK_ID_ALGR_9
0x00        TASK_ID_ALGR_10
@
```

The second file is the configuration file. Note the sensor identification number is not used but is included for backward compatibility. To activate a data group or FFT group simply specify a “1” at that group entry.

```
@ADC64_PARAM
1.0             Version Number
SAMPLE_UPDATE_RATE
1024            Max_sampling_rate
1               Update_rate
2               Sensor_id
50              Full scale voltage in 10’s of volts
8               Number of channels
ACTIVE_DATA_GROUP
1               data group 0 flag
0               data group 1 flag
0               data group 2 flag
0               data group 3 flag
0               data group 4 flag
0               data group 5 flag
0               data group 6 flag
1               data group 7 flag
ACTIVE_FFT_GROUP
1               fft group 0 flag
0               fft group 1 flag
0               fft group 2 flag
0               fft group 3 flag
```

```
0           fft group 4 flag
0           fft group 5 flag
0           fft group 6 flag
0           fft group 7 flag
GROUP_DIVISOR
1           groupd 0 divisor
1           groupd 1 divisor
1           groupd 2 divisor
1           groupd 3 divisor
1           groupd 4 divisor
1           groupd 5 divisor
1           groupd 6 divisor
1           groupd 7 divisor

@
```

Appendix E. Sample MATLAB Program to Read DFT Data Files

```
% Set up look up vector for gain factors as read from packet header
% Gain values range from 0 to 7 and map as follows:
%   0=10,1=100,2=100,3=1000,4=1000,5=10000,6=10000,7=100000
gain_vector=[1;10;10;100;100;1000;1000;10000];
file_name = '2000_09_28_18_40_06_Sen2_Grp0.dat';
number_of_seconds = 7;

% All data is 16 bit unsigned int store in little indian format
input_fd = fopen(file_name,'r','l');

% Each data block is preceded by a packet header
[Header,count] = fread(input_fd,20,'ushort');

sample_rate = Header(11);
update_rate = Header(12);

% loop through file extracting the data and stripping of packet headers
data=[];
for j=1:number_of_seconds
     % Extract current gain values for packet The gains are stored in the low byte
   % with the top nibble containing the gain value for the low channel group (1-4)
   % and the low nibble used for the high channels.  The gain values stored are
   % converted to real values via the gain lookup vector

   gain_channel_1to4 = gain_vector(bitshift(bitand(Header(13),hex2dec('000000f0')),-4)+1);
   gain_channel_5to8 = gain_vector(bitand(Header(13),hex2dec('0000000f'))+1);

 % All of the data is read in as a two dimensional array with the row
% count set to the number of channels (8) and duration of 1 packet = sample rate/
% update rate
   clear temp
   [temp,count] = fread(input_fd,[8,sample_rate/update_rate],'short');
   temp(1:4,:) = temp(1:4,:)./gain_channel_1to4;
   temp(5:8,:) = temp(5:8,:)./gain_channel_5to8;
   data = [data,temp];
    % Grab next header
   [Header,count] = fread(input_fd,20,'ushort');
end

%convert to voltage full scall is +/- 5 volts
data = (data .*5.0)/2^15;
fclose(input_fd)
```

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 2002	Final, Oct. 2000 to Sept. 2001

4. TITLE AND SUBTITLE
Electrical and Software Design Report for the Data Fusion Testbed

5. FUNDING NUMBERS
DA PR: AH16
PE: 62120A

6. AUTHOR(S)
Brian Mays

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Attn: AMSRL- SE-SA email: bmays@arl.army.mil
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-MR-536

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
ARL PR: 1NE4T1
AMS code: 622120.H16

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report describes in detail both the electrical and software design of the Data Fusion Testbed (DFT). The DFT was developed by the Acoustic Signal Processing Branch of the U.S. Army Research Laboratory as a field research platform to conduct state of the art Unattended Ground Sensor (UGS) research. The DFT was developed to bridge the gap between early algorithm research and real-world scenario test and evaluation. The DFT is a custom assembly of signal acquisition hardware, communications equipment, and signal processing hardware packaged in an environmentally rugged housing. The hardware assets of the DFT combined with its custom software architecture forms a generic UGS for use as a unique engineering tool in this specialized field of UGS research.

14. SUBJECT TERMS
Acoustics, battlefield, testbed, design report

15. NUMBER OF PAGES
55

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102