NAG8-720

MARSHALL GRANT

IN-61-CR

VOL. 1

287451

19P.

(NASA-CR-185880) NETWORK, SYSTEM, AND STATUS SOFTWARE ENHANCEMENTS FOR THE AUTONOMOUSLY MANAGED ELECTRICAL POWER SYSTEM BREADBOARD. VOLUME 1: PROJECT SUMMARY Research Report, Jun. 1988 - May 1990

N91-10618

526857

Unclas

G3/61 0287451

19P

The University of Alabama in Huntsville

JRC Research Report No. 90-18
June 1988 through May 1990

Network, System, and Status Software Enhancements for the Autonomously Managed Electrical Power System Breadboard

Project Summary

Grant NAG8-720

Volume 1 of 4 Volumes

Prepared by

James W. McKee
University of Alabama, Huntsville
Huntsville, Alabama 35899
(205) 895-6257

Prepared for

Norma Whitehead
EB12
NASA/MSFC
Huntsville, Alabama

Table of Contents

1. Management overview

This is the final report on the **Network, Systems, and Status Software Enhancements for the Autonomously Managed Electrical Power System Breadboard** grant number NAG8-720 from NASA/MSFC. The final report will be in four volumes. This volume gives a summary of the original AMPS software system configuration, points out some of the problem areas in the original software design that this project is to address, and in the appendix collects all the bi-monthly status reports. Volume 2 presents the specification, structured flow charts, and code listing for the new protocol. Volume 3 presents the specifications for the new operation system and the new high level AMPS commands. Volume 4 presents the description, structured flow charts, prints of the graphical displays, and source code to generate the displays for the AMPS graphical status system.

2. Introduction

The purpose of AMPS is to provide a self reliant system to control the generation and distribution of power in the space station. A discussion of the design philosophy and considerations of AMPS is presented in the three volumes of the TRW report <u>Space Power Distribution System Technology</u> (report number 34579-6001-UT-00). The AMPS breadboard is being used to develop and evaluate the software recommendations of the TRW report. The present AMPS breadboard hardware and software are documented the five volume TRW report <u>Space Power Distribution System Technology</u> (report number 39243-6001-UT-00).

The software in the AMPS breadboard can be divided into three levels; the operating environment software, the protocol software, and the station specific software. The operating environment software is the general software that creates the environment for the protocol software and the station specific software. It consists of functions such as station initialization, invoking station specific functions, executing tasks, and passing commands between the station specific software and the protocol. (A command is a message to another station.) The station specific functions do the actual setting and monitoring of the electrical power hardware. The station specific tasks include any algorithm software that is run in the station, e.g., load balancing. The protocol software handles the sending and receiving of commands between stations.

This project deals only with the operating environment software and the protocol software. The present station specific software will not changed except as necessary to conform to new data formats. New station specific functions may needed to be added as necessary to support the station requirements. All stations will have the same operating environment software and protocol software.

At the present time there are three types of stations in AMPS: the electrical power system controller, EPSC, the load center controller, LCC, and the power source controller, PSC. The EPSC is the central controller and can set and clear load connections and monitor the status of the batteries, loads and the solar array simulator. The LCC can connect, disconnect, and monitor the load simulators. The PSC can monitor the status of the batteries, load currents, and line voltage and control the charging of the batteries.

3. Present Communication Network

The EPSC, PSC, and LCC have identical networking capability. This consists of software in two location: on the respective CMC ENP-30 cards and on the respective controller cards. The CMC ENP-30 card is set to perform the Ethernet protocol and has software written by TRW to move the data to/from the ENP-30 from/to the controller card. On the controller card the message is analyzed to determine if the message is a valid message. A message is considered valid if station can execute the instruction in the message. If it is not, the message is discarded. If message is valid, the instruction in the message is executed. All messages are broadcast, there is no receive node address in the message.

The overall protocol in the AMPs is essentially a datagram, i.e., the message is sent by a node, but the node does not know if the message was received; there is no form of hand shaking.

Because there is no receive node address in a message, a message is considered valid if the instruction in the message is meaningful to the node that receives the message. This has serious implications if more than one of any of the types of AMPS stations is on the network. There is no way to control individual AMPS units. For example, if a EPSC broadcast a message to a LCC, every LCC on the network will receive the instruction and execute the instruction.

4. Description of the AMPS Hardware

The purpose of this hardware description is to support the references to hardware in the discussion to follow in the individual design specification portions. Therefore, this description will not cover all the aspects of the hardware, only those portions that are germane to the requirements specifications.

The AMPS breadboard can be thought of as a collection of stations that perform specific functions and that are connected by an Ethernet cable. The stations communicate with each other by sending commands by way of the Ethernet. Each station is capable of controlling and monitoring the electrical hardware associated with the station. Each station has an interface to the Ethernet, i.e., the CMC ENP-30 card.

The function of the ENP-30 card is to buffer, package, and send commands to other stations and to receive, un-package, and buffer commands coming from other stations. The ENP-30 card has 128K bytes in on-board dual ported RAM and a 68000 CPU. The ENP-30 can be a bus master and, therefore, access the RAM on the main CPU board. There is also 32K of EPROM, but it is dedicated to the Kernel software.

On station power up the Kernel software takes control of the ENP-30 and goes through a power up sequence which among other tasks, initialize the Lance (the Ethernet interface) chip. Then the Kernel turns the operation of the ENP-30 over to the software down-loaded from the CPU board. (All user created software must be down loaded to the ENP-30 from some other source on power up.)

The software on the ENP-30 operates in a three level priority scheme. The highest priority level is the Lance which takes control of the local CPU by way of interrupts; it also has the highest priority on the local RAM and can monopolize the RAM in blocks of a minimum of greater than 9 us. Lance is the actual interface to the Ethernet signals and does the transmission and reception of packets. One important implication of this structure is that the data to be sent must be in the memory on the ENP-30 card before the transmission is started because the Lance must be able to get immediate access to the data when needed.

The next highest priority level software is the Kernel. The Kernel is the interface between the user and Lance. Communication with the Kernel is through subroutine calls from the user's software. All data or parameters are passed by way of control blocks. The user subroutine call passes a pointer of the control block to the Kernel. All parameters are in predefined locations in the control block. If data is being passed, the address of the data block is in the control block.

The user's software has the lowest priority. It has access to the CPU and the memory when the other two levels are not using them. But this is most of the time because the Lance only needs the memory to read or write data in the packet being sent or received. The Kernel only uses the CPU and memory to set up the Lance from the data in the control block transferred to it in the subroutine call.

Each station has a CPU board (OB68K/MSBCI-256K-00-12) with on-board 256K bytes of RAM and 32K bytes of EPROM. This 32K bytes of EPROM contains all the user software, a portion of which is down-loaded to the ENP-30. Each station also has function specific interface boards that are controlled by the CPU (volume 1 of TRW 39243). The interface boards control and monitor the electrical hardware.

The PSC can program its interface boards to work as two channels working in parallel. The first channel can measure any of the following: 12 battery temperatures, 1 line voltage, and 6 currents (battery current, channel 1 current, channel 2 current, channel 3 current, load current, and solar array simulator current). The measurements can be made in any order, and each measurement requires 25 us. [can the board set a flag when complete?] The second channel can measure the voltage across any of the battery cells and the voltage across groups of 6 cells for a total of 196 measurements. The measurements can be made in any order, and each measurement requires a minimum of 78 us. [The I/O board can be setup to set a flag or give an interrupt when the conversion is complete. This has not been done. A wire must be added to the back plane.]

The LCC can operate its interface boards as two parallel channels. The first channel can set any of the 21 load switches (2 on each of 9 RPC simulator units and 3 on 1 RPC simulator unit) and monitor the setting of the 21 load switches in any order and in about 4 to 5 instruction times. The other channel can monitor 31 diode temperature, 10 load voltages and 10 load currents in any order at 25 us per conversion.

The EPSC has no I/O, A-to-D, or D-to-A hardware associated with it. Its function is to be the central controller of the power system and to monitor the settings of the various voltages, currents, temperatures and switch settings.

Appendix A Bi-monthly status reports

Network, System, and Status Software Enhancements for the Autonomously Managed Electrical Power System Breadboard

Bimonthly Status Report
For the Period June 1, 1988 through July 31, 1988

Submitted to: N. Whitehead Submitted by: J. McKee

1. I presented the schedule for the project in a meeting with Norma on July 1, 1988.

2. I have hired Dennis Wingo to work on the system software and the networking software. He is an undergraduate EE major. He has worked in industry for over nine years as an digital electronics technician. He is presently learning FORTH. Norma was given the paper work for Dennis to get his badge on July 20, 1988.

3. I have hired Shu Wang to develop the graphics. She is working on her masters in CS. She is a permanent resident. The paper work has been started to get her a badge. She is presently learning UNIX.

4. I have acquired a figFORTH and a polyFORTH interpreter. The figFORTH interpreter is adequate for learning the basics of FORTH. The polyFORTH interpreter lacks some documentation which I am attempting to acquire.

5. I have worked through two books on FORTH and have a basic understanding of the language.

Network, System, and Status Software Enhancements for the Autonomously Managed Electrical Power System Breadboard

Bimonthly Status Report
For the Period August 1, 1988 through September 30, 1988

Submitted to: N. Whitehead Submitted by: J. McKee

1. Dennis Wingo has spent most of his working time in the AMPS laboratory. He has aided DK in doing an inventory of the TRW drawing for AMPS. Their conclusion is that NASA has at least one copy of each drawing on the indentured drawing list.

2. Dennis also did an inventory on the hardware in the LCC, PSC, and EPSC. There appears to be only one floppy drive. The documentation indicates that there should be one drive for each system, for a total of three drives.

3. Dennis has been trying to compile, down load to the PROM burner, and burn a set of EPROMs. He has not been successful. There seems to be a problem in either the Data I/O or the cable. Dennis is working on the problem.

4. Dennis did an inventory of the AMPS documentation with the following results:

A. Documentation for AMPS hardware -- NASA has at least one copy of the documentation for each card in AMPS.

B. Documentation for the software development system -- NASA has copies four manuals of the five PolyFORTH manuals needed, one manual (the screen editor for PolyFORTH) is missing. We have contacted PolyFORTH and they are checking into what is involved in sending another copy to NASA.

C. We are also checking with PolyFORTH about the possibility of purchasing a screen editor for the development system.

5. I have read the three volumes of the TRW report 34579-6001-UT-00.

6. I have studying the five volumes of TRW report 39243-6001-UT-00 in conjunction with the manuals on the ENP-30 card. I have developed a more complete memory map and have collected in one document the description of all the buffers used in the protocol software. I have started commenting the TRW FORTH protocol software.

Network, System, and Status Software Enhancements for the Autonomously Managed Electrical Power System Breadboard

Bimonthly Status Report
For the Period October 1, 1988 through November 30, 1988

Submitted to: N. Whitehead   Submitted by: J. McKee

1. In a meeting with Norma, Louis, and Dr. Lee on October 24, it was decided to postpone the interface of the NCR Tower with the AMPS breadboard. Instead I am to concentrate on getting a protocol working that will allow Dr. Lee to have access to the data in AMPS.

2. Dennis has been trouble shooting the problem of the making PROMS with the Data I/O. He thinks that the Data I/O is not working. He has ask Norma to have the Data I/O sent off and fixed.

3. A preliminary version of the specification for the protocol and system software has been written. The first copy will be delivered to Norma in the first week of December.

4. I have been working with Silicon Graphics trying to get the proper commands to read a Tektronix's tape and convert the tape to the format the NCR. When we get this working we will be able to convert the Tektronix graphics software to a format that can be loaded onto the NCR.

Network, System, and Status Software Enhancements for the Autonomously Managed Electrical Power System Breadboard

Bimonthly Status Report
For the Period December 1, 1988 through January 31, 1989

Submitted to: N. Whitehead Submitted by: J. McKee

1. The first version of the protocol and system specification was submitted to Norma Whitehead during the first part of December. The specification was reviewed by Norma, Louis, and Bryen. A review meeting of the specification was held on January 4, 1989. A revised copy of the specification was delivered to Norma, Louis, Bryen on January 23, 1989. A copy of the revised specification was also sent to Dr. Lee.

2. In the meeting on January 4, 1989, Norma informed me that NASA was not going to use the NCR Tower computer to display the graphical status of AMPS. NASA will be evaluating other computers that will support the DI-3000 graphics package. I am to investigate how the protocol being designed can be implemented on each of the candidate computers.

3. I am to submit a revised scope of work for this project that will reflect the decision to use another computer for the status display and only use the NCR Tower as a monitor of the traffic on the network.

Network, System, and Status Software Enhancements for the Autonomously Managed Electrical Power System Breadboard

Bimonthly Status Report
For the Period February 1, 1989 through March 31, 1989

Submitted to: N. Whitehead   Submitted by: J. McKee

1. I submitted through the UAH contracts office a request for a one year extension of the grant. I also requested a change in the scope of the grant to be able to support NASA's decision to use a computer that can support the DI-3000 graphics package.

2. Dennis demonstrated to Norma that he could down load a program to the PROM programmer, and burn PROMs that would work in the system. The problem has been incorrect procedure instructions in the TRW documentation.

3. The structured flow diagrams have been completed and are being incorporated into the specification.

4. Dennis and Shu have started writing the new protocol in the AMPS laboratory.

Network, System, and Status Software Enhancements for the Autonomously Managed Electrical Power System Breadboard

Bimonthly Status Report
For the Period April 1, 1989 through May 31, 1989

Submitted to: N. Whitehead Submitted by: J. McKee

1. The request for a one year extension of the grant has been processed. The request for a change in the scope of the grant is also complete.

2. Dennis is in the process of determining how to compile the new protocol and down load the code to the ENP-30 card. There have been some problems trying to follow the logic in the FORTH compilers created by TRW. It appears that the FORTH system that NASA has is not being supported by either PolyFORTH or Omni-byte.

3. Shu is in the process of typing in the protocol. We will first debug the protocol in the MSBC-1 memory space. We will simulate the actions of the Kernel in the MSBC-1. This is because the ENP-30 environment does not have a terminal driver. Therefore, it is very difficult to "see" what is going on in the code in the ENP-30. The only way is to use the ENP-30 debugged PROMS.

Network, System, and Status Software Enhancements for the Autonomously Managed Electrical Power System Breadboard

Bimonthly Status Report
For the Period June 1, 1989 through July 31, 1989

Submitted to: N. Whitehead Submitted by: J. McKee

1. Shu has typed in the protocol depicted in the software flows. We have debugged the protocol in the MSBC-1 memory space. We have started the process of compiling the protocol and the application and debugging them together.

2. Dennis has determined how to compile the new protocol and down load the code to the ENP-30 card. There were some problems trying to follow the logic in the FORTH compilers created by TRW.

3. The protocol, as is stands now, is over 9 K bytes when compiled with zero length names. Some of the application software is close to 12 K bytes. Therefore, it will be necessary to use 128K PROMs (16K by 8).

4. We can not get the EPSC application software (supplied by TRW) to compile. This could be a major problem because we will need to be able to compile the application program to interface to the new protocol.

Network, System, and Status Software Enhancements for the Autonomously Managed Electrical Power System Breadboard

Bimonthly Status Report
For the Period August 1, 1989 through September 30, 1989

Submitted to: N. Whitehead   Submitted by: J. McKee

1. Dennis has modified the TRW compilers so that we can compile to 128K EPROMs. We are able to program 128K EPROMs on the Data I/O and run the EPROMs.

2. Dennis and Norma fixed the Multi-bus back plane which was causing intermittent system crashes.

3. Shu is writing a shell in which to run the protocol. The shell will allow ASCII messages to be typed in at one station and displayed on the receiving station.

4. Shu is writing the interrupt routines to interface the protocol with the Kernel.

Network, System, and Status Software Enhancements for the Autonomously Managed Electrical Power System Breadboard

Bimonthly Status Report
For the Period October 1, 1989 through November 30, 1989

Submitted to: N. Whitehead Submitted by: J. McKee

1. Shu has been working on the protocol for the period. She is still having trouble getting the software interface to the ENP-30 card working.

2. I have sent the letter to Precision Visuals informing them that I have removed any and all of their software form my Sun. I am ready to transfer the software and manuals to Norma as soon as she has her Sun configured.

3. I have hired another computer science graduate student to work on the creation of the menus for AMPS. Mr. Kwangsoo Yi has started the process of getting a NASA badge. He has been reading the DI-3000 manual and is ready to start creating simple graphics using the DI-3000 software.

Network, System, and Status Software Enhancements for the Autonomously Managed Electrical Power System Breadboard

Bimonthly Status Report
For the Period December 1, 1989 through January 31, 1990

Submitted to: N. Whitehead   Submitted by: J. McKee

1. Shu has been able to receive on one station packets broadcast from another station. She is now working on getting the protocol to initialize.

2. Kwangsoo has installed all of the Precision Visual software. He has been able to create graphics on the Tektronix display. He is in the process of designing a graphics representation of the AMPS circuit.

Network, System, and Status Software Enhancements for the Autonomously Managed Electrical Power System Breadboard

Grant No. NAG8-720

Bimonthly Status Report
For the Period February 1, 1990 through March 31, 1990

Submitted to: N. Whitehead Submitted by: J. McKee

1. Shu is continuing to debug the network software. She is having problems because she can not interactively run the network program on the ENP-30. (The protocol has to be compiled and down loaded to the ENP-30 card to run.) She only has the PROM debugger.

2. Kwangsoo has some of the graphics created. He is working with Norma on the design of the various levels of the graphics.

Network, System, and Status Software Enhancements for the Autonomously Managed Electrical Power System Breadboard

Grant No. NAG8-720

Bimonthly Status Report
For the Period April 1, 1990 through May 31, 1990

Submitted to: N. Whitehead

Submitted by: J. McKee

1. The project ended at the end of May. Therefore this will be the last status report on the project.

2. Yi has completed the implementation of the graphics display. He has worked with Norma in defining how the display should appear and function and has implemented that design.

3. Shu has not been able to debug the protocol. The debugging system on the ENP-30 card did not give her enough insight into the operation of the code on the card to locate the problems.