

# **PRINTED CIRCUIT BOARD DESIGN FOR FREQUENCY DISTURBANCE RECORDER**

**By**

**Lei Wang**

**Thesis submitted to the Faculty of the  
Virginia Polytechnic Institute and State University  
In partial fulfillment of the requirements for the degree of**

**MASTER OF SCIENCE  
In  
Electrical Engineering**

**APPROVED:**

---

Dr. Yilu Liu, Chairman

---

Dr. Richard Connors

---

Dr. Virgilio Centeno

December 12, 2005  
Blacksburg, Virginia

Keywords: Frequency Disturbance Recorder (FDR)  
Frequency Monitoring Network (FNET)  
Power system monitoring  
Printed Circuit Board (PCB)  
Hardware interface  
MPC555

# **Printed Circuit Board Design for Frequency Disturbance Recorder**

Lei Wang

## **ABSTRACT**

The FDR (Frequency Disturbance Recorder) is a data acquisition device for the power system. The device is portable and can be used with any residential wall outlet for frequency data collection. Furthermore, the FDR transmits calculated frequency data to the web for access by authorized users via Ethernet connection. As a result, Virginia Tech implemented Frequency Monitoring Network (FNET) with these FDR devices. FNET is a collection of identical FDRs placed in different measurement sites to allow for data integration and comparison. Frequency is an important factor for power system control and stabilization. With funding and support provided by ABB, TVA and NSF the FDRs are placed strategically all over the United States for frequency analysis, power system protection and monitoring.

The purpose of this study is to refine the current FDR hardware design and establish a new design that will physically fit all the components on one Printed Circuit Board (PCB). At the same time, the software that is to be implemented on the new board is to be kept similar if not the same as that of the current design. The current FDR uses the Axiom CME555 development board and it is interfaced to the external devices through its communication ports. Even though the CME555 board is able to meet the demands of the basic FDR operations, there are still several problems associated with this design. This paper will address some of those hardware problems, as well as propose a new board design that is specifically aimed for operations of FDR.

## **ACKNOWLEDGEMENTS**

I would like to express my deepest gratitude to my academic and research advisor Dr. Yilu Liu for her guidance and constant support in helping me to conduct and complete this work. I would also like to thank Dr. Richard Connors for his constant support and expertise in many different areas of the research. In addition, I want to thank Dr. Vigilio Centeno for serving on my advisory committee, as well as for his kind advice and help in the classroom and lab. Also, I want to thank Ryan (Jian) Zuo, Emily (Chunchun) Xu, Kevin (Zhian) Zhong and Bruce Billian for their generous advices and helpful comments regarding the development of Frequency Disturbance Recorder and FNET. Finally, with deepest love and appreciation, I would like to thank my father Yuwen Wang, my mother Zengjuan Li and my lovely sister Julie Wang for their consistent support and encouragements.

# TABLE OF CONTENTS

Chapter 1: Introduction.....	1
1.1 Current FDR Hardware Architecture.....	1
1.2 Problem Statement and Proposed Work.....	2
1.3 Thesis Organization.....	3
Chapter 2: Axiom CME555.....	5
2.1 MPC555 Processor.....	8
2.2 External Memory.....	11
2.3 Jumpers and Switches.....	11
2.4 Ports and Connectors.....	13
Chapter 3: External Peripherals.....	14
3.1 Signal Conditioning Unit.....	14
3.2 ADC Unit.....	14
3.3 Motorola M12+ Receiver.....	15
3.4 Serial to Ethernet Converter.....	17
Chapter 4: FDR Software.....	18
Chapter 5: Past and Related Efforts.....	21
5.1 Protel.....	21
5.2 MPC555 Support Pins.....	21
Chapter 6: Printed Circuit Board Design – PB555 Prototype.....	27
6.1 Prototype Proposal.....	27
6.2 PB555 board.....	28
6.3 Prototype Development.....	30
6.4 LCD Interface.....	32
6.5 LCD Test.....	40
6.6 GPS Interface.....	48
6.7 Signal Conditioning Unit and ADC Interface.....	52
Chapter 7: Conclusions and Future Work.....	54
7.1 Conclusions.....	54
7.2 Future Work.....	54

References.....	55
Appendix A Initialization and configuration code for PB555.....	60
Appendix B LCD driver code.....	68
Appendix C Schematics.....	77
Appendix D CME555 jumper and switch setting for FDR.....	86
Appendix E Netlisting for FDR PCB design.....	88

## LIST OF FIGURES

Figure 1 Block diagram of current FDR architecture.....	1
Figure 2 Picture of FDR.....	2
Figure 3 CME555 board layout.....	6
Figure 4 CME555 memory map.....	7
Figure 5 MPC555 memory map.....	10
Figure 6 Block diagram of MPC555.....	11
Figure 7 Functional diagram of AD976A.....	15
Figure 8 Bottom view of M12+ receiver.....	16
Figure 9 Functional diagram of M12+ receiver.....	17
Figure 10 Flow Chart of Algorithm for FDR Initialization.....	19
Figure 11 Flow Chart of Algorithm for main program.....	20
Figure 12 Preliminary MPC555 pin configuration for FDR board design.....	26
Figure 13 Block Diagram of proposed PCB design.....	28
Figure 14 Picture of PB555 board.....	28
Figure 15 PB555 Memory map.....	30
Figure 16 Wire wrapping – View from the top of terminal.....	31
Figure 17 Wire wrapping – view from the side of terminal.....	32
Figure 18 Schematic of Address decoding circuitry for LCD on CME555.....	33
Figure 19 Schematic of LCD port connection and contrast circuitry on CME555.....	33
Figure 20 Write Operation Timing for DMC20261NY-LY-AXE.....	36
Figure 21 Read Operation Timing for DMC20261NY-LY-AXE.....	36
Figure 22 Schematic of LCD interface for PB555 board.....	37
Figure 23 Example power supply circuit for LCD module.....	39

Figure 24 Wire wrapped PB555 prototype for LCD interface.....	40
Figure 25 Functional diagram of DMC20261NY-LY-AXE.....	41
Figure 26 Null modem connection without RTS/CTS handshaking.....	49
Figure 27 Power supply circuit for QS3861.....	50
Figure 28 Schematic of GPS interface for PB555 board.....	52
Figure 29 AD976A circuitry.....	53

## **LIST OF TABLES**

Table 1 Summary of MPC555 Support Signal Connections.....	25
Table 2 Mapping of LCD port pins to MPC555 signals.....	34
Table 3 DMC20261NY-LY-AXE timing constraints.....	35
Table 4 Absolute maximum ratings for DMC20261NY-LY-AXE power supply.....	39
Table 5 HCD44780 controller register selection.....	42
Table 6 Operation code for HCD44780 instructions.....	42
Table 7 Memory initialization modifications.....	44
Table 8 M12+ connector pin information.....	51

## **LIST OF ABBREVIATIONS**

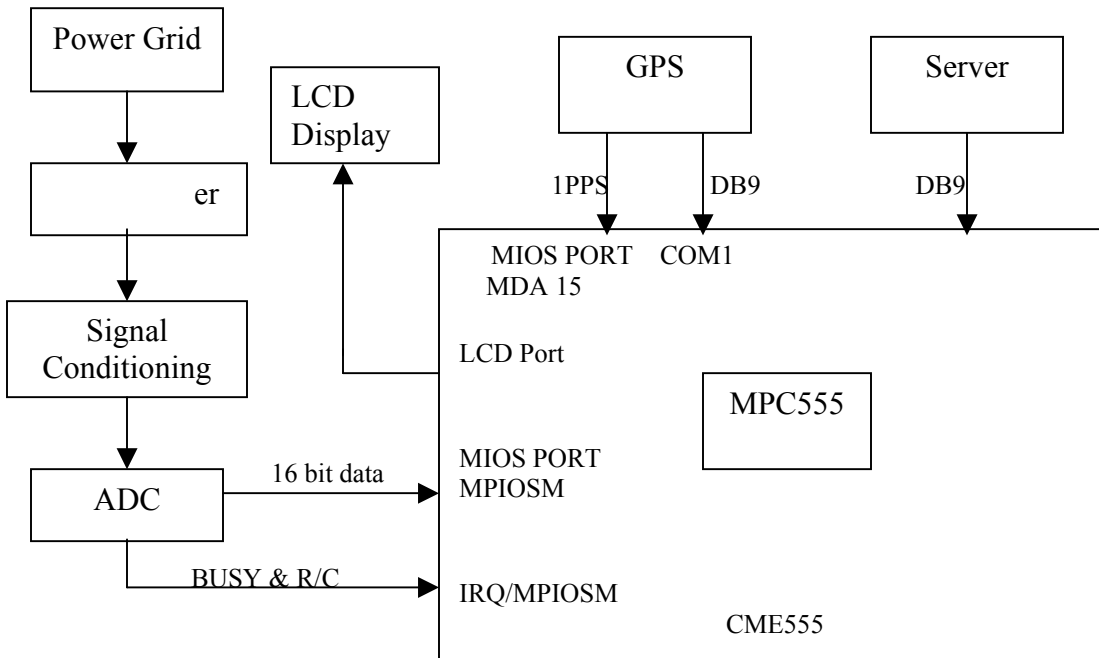
PCB: Printed Circuit Board  
FDR: Frequency Disturbance Recorder  
FNET: Frequency Monitoring Network  
LCD: Liquid Crystal Display  
ADC: Analog to Digital Converter  
GPS: Global Positioning System  
EPLD: Electrically Programmable Logic Device  
LED: Light Emitting Diode  
JTAG: Joint Test Action Group  
BDM: Background Debug Monitor  
PWM: Pulse Width Modulation  
TPU: Time Processor Unit (MPC555 subsystem)

MIOS: Modular Input Output Subsystem (MPC555 subsystem)  
QSMCM: Queued Serial Multi-Channel Module (MPC555 subsystem)  
QADC: Queued Analog to Digital Converter (MPC555 subsystem)  
TouCAN: Two 2.0B Controller Area Network (MPC555 subsystem)  
DASM: Double Action Submodules (MPC555 subsystem)  
PWMSM: Pulse Width Modulation Submodules (MPC555 subsystem)  
MCSM: Modulus Counter Submodules (MPC555 subsystem)  
PIOSM: Parallel Port Input Output Submodules (MPC555 subsystem)  
GPIO: General Purpose Input Output  
EEPROM: Electrically Erasable Programmable Read Only Memory  
EPROM: Electrically Programmable Read Only Memory  
RTS: Request to Send (Protocol signal)  
CTS: Clear to Send (Protocol signal)  
EMI: ElectroMagnetic Interference  
TTFF: Time To First Fix  
CPU: Central Processing Unit  
SRAM: Static Random Access Memory  
SCI: Serial Communication Interface  
SPI: Serial to Parallel Interface  
CMOS: Complementary Metal Oxide Semiconductor  
TTL: Transistor Transistor Logic  
DGPS: Differential Global Positioning System  
TCP/IP: Transmission Control Protocol/Internet Protocol  
UTC: Coordinated Universal Time  
PPS: Pulse Per Second

# CHAPTER 1 INTRODUCTION

## 1.1 Current FDR Hardware Architecture

The current FDR hardware is composed of the Axiom CME-555 board, an Analog to Digital Converter unit (ADC), a Global Positioning System unit (GPS), a signal-conditioning unit, a Serial to Ethernet Converter and a power supply [14]. See Figure 1 [14] for a block diagram of the overall system. The main algorithm is implemented in a microprocessor-based unit, where the frequency computation, timing synchronization and communication tasks are executed. The overall operation of the FDR can be described as the following: the transformer first lowers the power grid voltage to a reasonable level, then the signal conditioning and ADC unit process the signal for sending it to the microprocessor, and finally the microprocessor communicates to the GPS for timing synchronization and outputs the frequency estimation results to the server



**Figure 1 - Block diagram of current FDR architecture**

[14]. Figure 2 shows the outside view of a FDR unit. There is a Serial to Ethernet converter sitting on top of the unit. The embedded Serial to Ethernet converter makes the device networked, which allows for remote control and access to the device. Also, there



is a liquid crystal display (LCD) in front of the unit panel that shows real time data including current time, frequency, voltage and the number of satellites acquired. Furthermore, this particular design allows plenty of space for development and the frequency estimation can be used in applications that involve power system monitoring, protection and control.



**Figure 2 - Picture of FDR**

## **1.2 Problem Statement and Proposed Work**

The current FDR hardware design is not optimized for efficiency or cost. In addition, most of the hardware that is implemented within the device is unused. The CME555 board is not specifically aimed for FDR applications and the cost of the board is relatively high for the some of the unnecessary functionalities it provides. Also, the device is not so compact for installation in some circumstances. For every installation, the CME555 requires some on-board hardware configurations as well as establish connection to the GPS, ADC and the signal-conditioning unit. As a result, the need for optimizing FDR hardware has triggered the initiative for the design of a new FDR board. The main objective of the new FDR board is to integrate the GPS, ADC and signal

conditioning unit into one board, as well as to reduce the cost for manufacturing and installation of FDR. Also, the new FDR board needs to be able to support some of the basic characteristics of FDR. The board has to be able to accommodate a LCD of some kind that displays real time data and serial ports for GPS and server communication.

### **1.3 Thesis Organization**

Much work has been done in the field of circuit design for FDR printed circuit board (PCB). This thesis presents a proposal for the design of a new FDR PCB, as well as a detailed procedure for establishing a final board design. Finally, results are presented for a PCB prototype implemented for interfacing to the LCD.

Chapter 1 is an overview of the background information for the project, including the current hardware architecture of FDR and problems that are associated with the current design.

Chapter 2 presents a detailed description of the current board Axiom CME555 that is currently implemented in the FDR. Also, some of the board design requirements are briefly discussed.

Chapter 3 goes through the hardware that are interfaced to the CME555 board.

Chapter 4 briefly describes the current FDR software architecture.

Chapter 5 investigates the MPC555 subsystems and the external logic that is needed for supporting the chip. Some of the earlier research efforts are presented with emphasis on the actual circuit design for MPC555.

Chapter 6 proposes a practical way of approaching the PCB design. Specifically, a prototype based on the PB555 board needs to be implemented for each of the external devices. The chapter concludes by discussing some of the results obtained from the LCD interface.

Chapter 7 summarizes the entire thesis work and suggests the future work.

Appendix A has a set of software initialization code for PB555. These initialization files are needed to setup configuration for the PB555 board. The Linker Command File was modified from that of CME555 provided by Metrowerks Codewarrior and the file is used to specify the location of RAM and ROM and other memory intensive parameters. Metrowerks provided the Runtime initialization file for

single chip operation, in this case the MPC56X single chip Runtime initialization file is used. Note that MPC56X has the same in-chip configuration as MPC555, so it can be used to replace the Runtime initialization file for the Axiom CME555. Furthermore, the memory configuration file is also based on a single chip MPC56x, this file allows for the proper display of memory during debugging.

Appendix B contains the code for driving the LCD to display characters. The C++ driver code in Appendix B was obtained from part of the FDR software, it simply displays the initializing messages to the LCD. The LCD initializing steps are referenced from the HD 44780 controller datasheet [37]. Also the EPLD (Erasable Programmable Logic Device) code in the end of Appendix B was received from Axiom, which is the same code that is implemented on the EPLD chip in the CME555 and used in the PB555 prototype.

Appendix C contains schematics of CME555 obtained from Axiom, which is referred to frequently throughout the thesis. Appendix C also shows the schematic for PB555 interfacing to the external components of FDR, the PB555 schematic was obtained from Axiom and the interfacing schematic was developed based on the discussions in Chapter 6 of this thesis.

Appendix D shows the current CME555 jumper and switches settings for FDR, which is included in the thesis to show how the board connections and configurations are setup in the current design.

Finally, Appendix E contains the netlisting files that were generated from the Protel schematics. The netlisting files are in EDIF format and they are separated into three different schematic files including the MPC555 chip connections, LCD and GPS interface connections and ADC and signal conditioning unit connections. The MPC555 netlisting was generated from the schematic in Figure 12 of Chapter 5 and the external device netlistings were generated from the PB555 external interface schematics in Appendix C. Appendix E was incorporated into this thesis to summarize the overall thesis work.

## **CHAPTER 2**

### **AXIOM CME555**

The Axiom CME555 board has a fully configured development system for the Motorola MPC555 PowerPC microcontroller. The system allows for Plug and Play and it has a large amount of external memory, 128K x 32 bits (512 Kbytes) Static RAM and 128K x 32 bits (512 Kbytes) Flash EPROM. In addition, the board has two memory sockets for additional memory expansion, two built-in RS-232 connectors on board, two TouCAN ports with transceivers, a LCD connector with contrast, a 16 key keypad connector, a large number of port connectors for MPC555 signals and a large prototyping area and Debug/JTAG Development connectors [1].

Physically, the CME555 has a board size of 7" x 7.5" [1]. Figure 3 [1] shows the physical allocation of different devices on the board. Specifically, the MPC555 chip is located close to the center of the board. In addition, there are rows of headers along the sides of the chip. Located below the MPC555 chip, there are several external memory chips along with latches and an EPLD chip. At the very bottom of Figure 3, there is the LCD and keypad port along with the contrast potentiometer. To the right of MPC555, there is a large amount of breadboard and expansion space alongside with connectors, testing LEDs and switches. At the very top of Figure 3, there are two RS232 connectors, a RS232 transceiver chip, several jumpers and switches. On the left side of MPC555 there are jumpers, external memory configuration switches and various connectors. More significantly, the BDM port is located just left of MPC555.

The physical architecture of CME555 is not optimized for the general operation of FDR. It is obvious that the CME555 board has a large amount of external memory that is not needed for the operation of FDR. In addition, there is the CAN (Controller Area Network) communication interface, the keypad interface, the external memory configuration switches, the testing LEDs and switches not used for the operation of FDR. An optimized PCB for the FDR would eliminate all of this unnecessary hardware to allow for a cheaper, cleaner and simpler solution. A replacement of the CME555 board would also affect the software that is currently used. Since the current FDR software is specifically for the MPC555-based system, the new board design will have to be MPC555-based as well to accommodate the software. Furthermore, to allow for

reusability of the current FDR software, the new PCB needs to be compatible with the current software. As a result, a good starting point in the design of FDR board would be a review of the CME555 hardware. One of the most significant aspects of the CME555

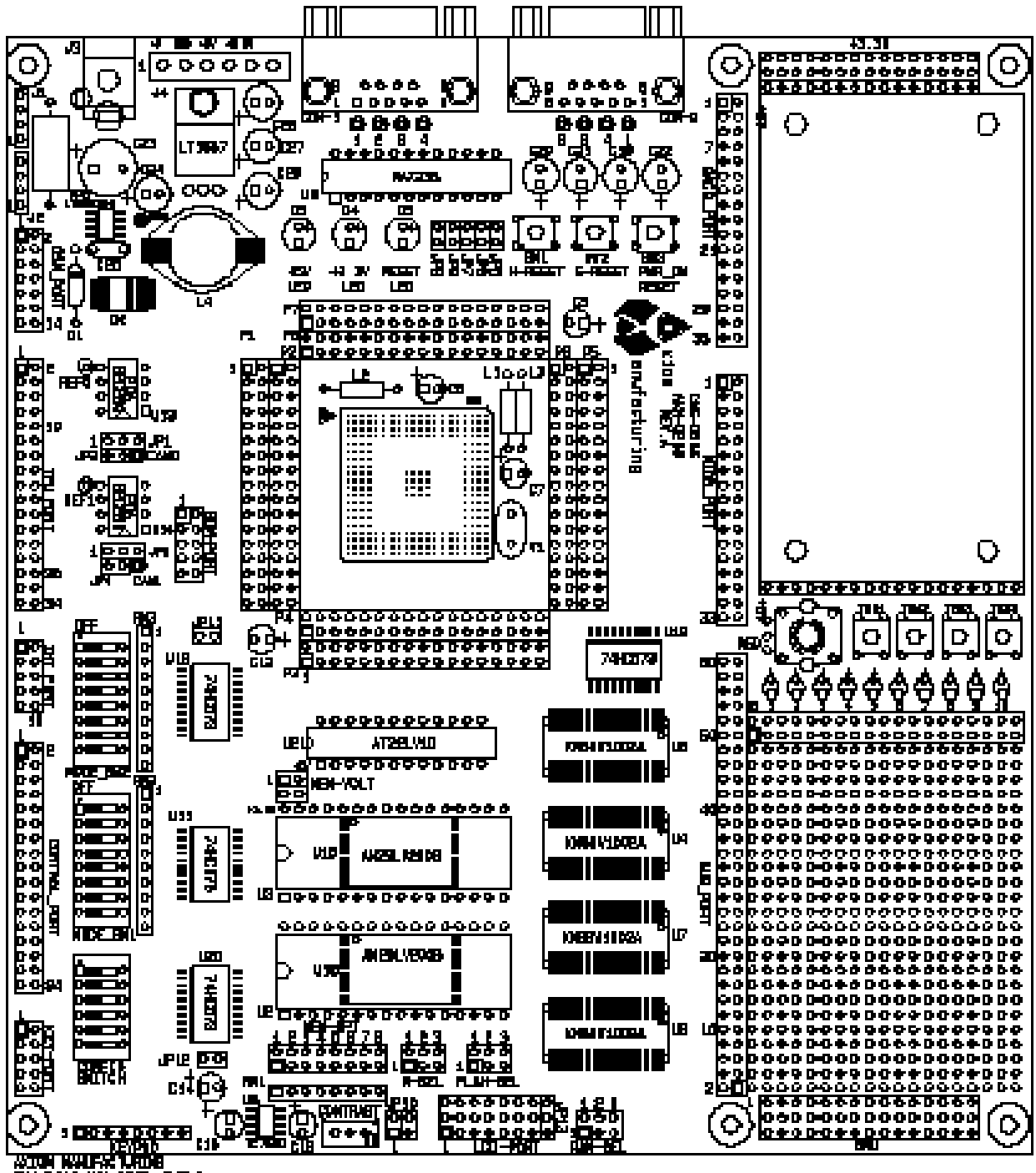


Figure 3 - CME555 board layout

board is the large amount of memory it has. To better understand the memory structure of CME555, a memory map of the board is shown in Figure 4 [1].

<b>Internal Memory</b>		
0000 0000 0006 FFFF	CMFI Internal Flash Memory Array	
0007 0000 002F BFFF	Reserved	
002F C000 003F 97FF	On-chip Control and Status Registers (see Figure 5)	
003F 9800 003F FFFF	Internal SRAM	
<b>External Memory</b>		
0040 0000 007F FFFF	40 0000 – 40 FFFF for 64 K device 40 0000 – 47 FFFF for 512 K device	CS 0 – External EPROM – 16 bit M-SEL devices U2-U3 (shipped with monitor program here)
0080 0000 00BF FFFF	<b>CS1</b> – External SRAM – 32 bit RAM-SEL devices U4-U7 512K bytes at 80 0000 – 87 FFFF on this board	
00C0 0000 00FF FFFF	<b>CS2</b> – External Flash Memory – 32 bit FLSH-SEL devices U15-U16 512K bytes at C0 0000 – C7 FFFF on this board (located on the board underneath the external EPROM)	
0100 0000 013F FFFF	<b>CS3</b> – External Peripheral Devices	
0140 0000 FFFF FFFF	Un-Mapped Address Space	

**Figure 4 - CME555 memory map**

Figure 4 shows the internal memory map of MPC555 as well as the external memory map of CME555 board. The external memory mapping is especially important since it would affect where the FDR software is placed as well as where the LCD is memory mapped. As seen in Figure 3, starting at memory location 0x00400000 there is the optional external memory expansion slot. Axiom ships the CME555 board with a monitor program preprogrammed within EPROM (Electrically Programmable Read Only Memory) units installed in the optional memory sockets. Memory block starting 0x00800000 to 0x00FFFFFFF is dedicated to the static RAM (Random Access Memory) and Flash memory, each taking up 512 Kbytes of memory. Finally, memory location starting 0x01000000 to 0x013FFFFFFF is for keypad and LCD interfacing. For the new PCB design, some of the external memory can be removed. However, for software

compatibility issues, the original LCD memory allocation needs to stay intact. The ideal PCB design would have the LCD at the same memory location as the CME555 board.

## **2.1 MPC555 Processor**

The MPC555 from Motorola is a 32 bit embedded microcontroller [24]. It is used most extensively in the automotive applications, such as engine control, transmission control, suspension and stability. It is also used in the robotics and avionics control industry. However, in the recent years, the MPC555 is rapidly being integrated into applications such as avionics, controls, analysis equipment, robotics and power management. In the case of frequency disturbance recorder, all of the frequency estimations are done within the MPC555 chip. The clock speed of MPC555 is able to run as high as 40MHz. And with 448 Kbytes of embedded FLASH memory and 26 Kbytes of Static RAM [24], the MPC555 allows for ease of modification or upgrading of the software implementation. Moreover, the MPC555 supports a wide range of on-board peripherals, these include third generation Time Processor Units (TPU3), two TouCAN Controller Area Network (CAN 2.0B) modules, and dual queued analog to digital converter (QADC) [24]. However, for the purpose of FDR operations, only the TPU3 is used. Each TPU includes its own RISC core and memory system and functions like another microcontroller within the MPC555. The TPU's sole purpose is to provide timing functions capable of processing up to 20 million instructions per second with a 40 MHz system clock. The Controller Area Network (CAN) is a broadcast-based communication protocol. It is often utilized in automotive and industrial networking. Finally, the QADC within the MPC555 has 32 channel internally multiplexed or 81 externally multiplexed with 10 bit resolution with 5 uS conversion time. It is fully capable of simultaneously sampling two analog inputs [24]. However, since the internal QADC only allows up to 10 bits conversion with internal sample/hold, an external ADC is used instead to tolerate more precise frequency estimates.

To allow for communication with external devices, the MPC555 has many different kinds of Input/Output interfaces. The 24 address and 32 data pins can be used for general purpose I/O (GPIO). Also, many peripheral pins can be used for general purpose I/O as well when not used for primary function. Out of the 18 channels Modular

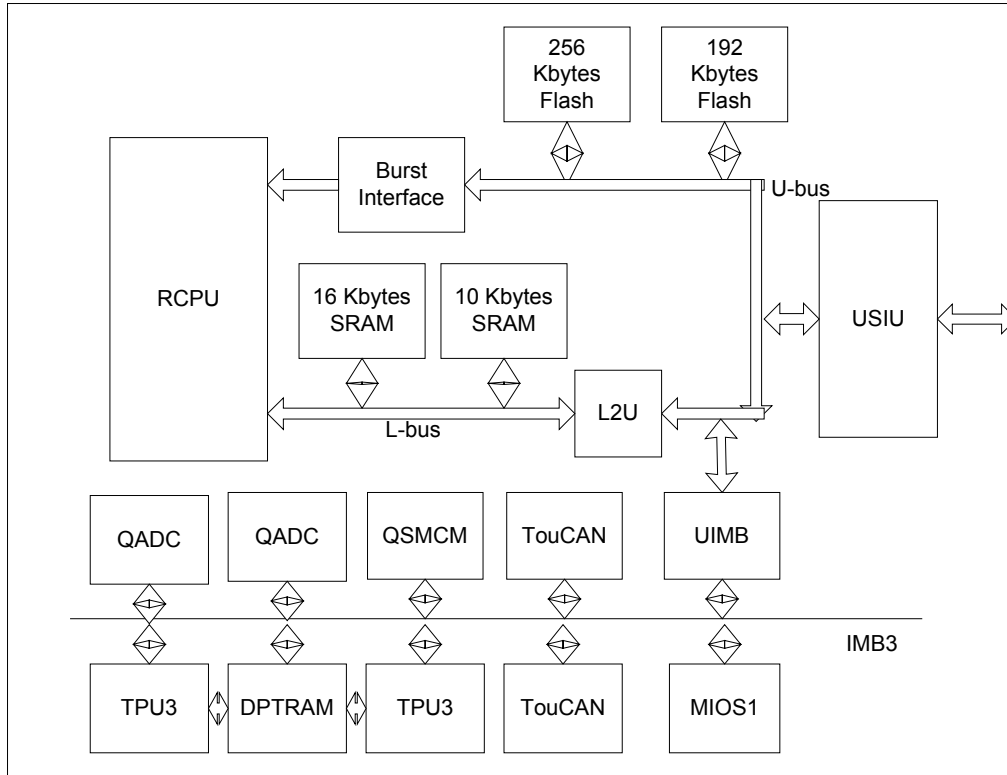
I/O System (MIOS) in the MPC555, 10 are used for double action submodules (DASM), 8 are dedicated to pulse width modulation (PWM) operations [24]. Also included in the MIOS are two 16 bit modulus counter submodules (MCSM) and two parallel port I/O submodules (PIOSM). For serial communication interface (SCI) and serial peripheral interface (SPI), the MPC555 introduces the Queued Serial Multi-Channel Module (QSMCM). The Queued Serial Peripheral Interface (QSPI) provides full-duplex communication port for peripheral expansion or inter-processor communication [24]. Also, the QSPI has four programmable peripheral-select pins that supports up to 16 devices. Each SCI provides 16 register receive buffer and 16 register transmit buffer, as well as separate transmitter and receiver enable bits and double buffering of data.

The memory map of MPC555 [24] microcontroller is shown in Figure 4. The available memory space for development includes 448 Kbytes of flash memory – address space 0x00000000 to 0x0006FFFF and 26 Kbytes of SRAM – address space 0x003F9800 to 0x003FFFFFF. Control registers and IMB2 (intermodule bus) modules take up 64 Kbytes. In addition, Figure 6 [24] shows a block diagram of MPC555 with all of the subsystems and their bus connections shown.



0x00 0000	CMF Flash A 256 Kbytes	
0x04 0000 0x06 FFFF	CMF Flash B 192 Kbytes	
0x07 0000 0x2F BFFF	Reserved for Flash (2.6 Mbytes – 16 Kbytes)	
0x2F C000 0x2F FFFF	USIU & Flash Control 16 Kbytes	
	0x2F C000	USIU Control Registers 1 Kbyte
	0x2F C800	FLASH Module A (64 bytes)
	0x2F C840	FLASH Module B (64 bytes)
	0x2F C880	Reserved for USIU
0x30 0000 0x30 7FFF	UIMB Interface & IMB3 Modules (32 Kbytes)	
	0x30 0000	DPTRAM Control (12 bytes)
		Reserved (8180 bytes)
	0x30 2000	DPTRAM (6 Kbytes)
		Reserved (2 Kbytes)
	0x30 4000	TPU3_A (1 Kbyte)
	0x30 4400	TPU3_B (1 Kbyte)
	0x30 4800	QADC_A (1 Kbyte)
	0x30 4C00	QADC_B (1 Kbyte)
	0x30 5000	QSMCM (4 Kbytes)
	0x30 6000	MIOS1 (4 Kbytes)
	0x30 7080	TouCAN_A (1 Kbyte)
	0x30 7480	TouCAN_B (1 Kbyte)
	0x30 7884	Reserved (1920 bytes)
0x30 7F80 0x30 7FFF	UIMB Registers (128 bytes)	
0x30 8000 0x37 FFFF	Reserved for IMB3 (480 Kbytes)	
0x38 0000	SRAM Control A (8 bytes)	
0x38 0008	SRAM Control B (8 bytes)	
0x38 0010	Reserved (485.98 Kbytes)	
0x3F 9800	SRAM A (10Kbytes)	
0x3F C000 0x3F FFFF	SRAM B (16 Kbytes)	

**Figure 5 - MPC555 memory map**



**Figure 6 - Block diagram of MPC555**

## 2.2 External Memory

The Axiom CME555 board features two AM29LV2008 Flash Memory chips from AMD, four KM681002AJ Static RAM chips from Samsung and two on-board Option Memory EPROM banks for 16 bit wide memory. The Option Memory EPROM banks come with two NM27C256 memory chips. Each of these memory banks can be configured individually to operate from the MPC555 chip selects CS0, CS1, or CS2.

## 2.3 Jumpers and Switches

CME555 board provides many switches and jumpers for external hard reset configuration, external memory device selection and configuration, LCD and keypad port configuration, MPC555 power supply options, serial port options, as well as LEDs, test switches and potentiometer for development purposes.

Mode Switch 1, Mode Switch 2 and Config Switch provide the external Hard Reset Configuration Word when enabled [1]. The switches provide a logic 0 when it is in off position and a logic 1 when it is in the on position. Config Switch 4 is used to enable

External Configuration, when it is turned on, a Hard Reset signal will enable the External Reset Configuration Word from Mode Switches 1 and 2 on to the Bus for input by the MPC555. Config Switch 5 enables the EPEE signal to program the on-chip CMF flash [1]. Config Switch 6 provides the Vpp supply connection for programming the internal Flash EPROM Memory module. Config switches 1 to 3 provide the clock selection for the board [1].

External memory device selection is done with RAM-SEL, FLSH-SEL and M-SEL Jumpers. The RAM-SEL jumper selects which chip select (CS) accesses the on-board Static RAM memory bank. This memory bank is often used for program debugging before programming into internal or external flash memory. The FLSH-SEL jumper selects which chip select accesses the on-board EPROM memory bank. The memory bank is available for user to provide additional program code or data space and optional Boot start memory space. Finally, the M-SEL jumper is used to select which chip select accesses the on-board Option Memory EPROM bank. Once the external memory selection jumpers are removed, the memory space specified by that particular jumper will be disabled and becomes idle. However, the default memory selection sets the Static RAM to CS1, Flash EPROM to CS2 and the Option Memory EPROM to CS0. The MEM\_OPT jumper selects the type of devices installed on the two optional memory sockets. Axiom has provided CME555 board with a simple Monitor program in the EPROM that is installed in these optional sockets. Finally, the MEM\_VOLT Option jumper is used for selection of 5V or 3.3V for the power supply of the optional memory devices installed.

The rest of CME555 jumpers include configuration for LCD port, keypad port, serial port and MPC555 power supply. Default configuration has jumpers 10, 11 and 12 installed. Jumper 10 allows for selection of display power pin polarity on the LCD port, jumper 11 enables CS3 to be used for LCD and keypad port and jumper 12 enables IRQ4 to be used as a key input interrupt service. MPC555 cut-away jumpers are used for power supply configuration and are rarely used. Jumpers 5 to 9 control serial port configurations. COM1 and COM2 provide serial connection to the MPC555 internal serial ports SCI1 and SCI2 respectfully. SCI1 is connected to the RS232 transceiver device and SCI2 can be configured to use that same transceiver. If RTS/CTS handshaking

is needed, jumpers 5 through 7 and jumper 9 allows TPU channel 14 or 15 to be used as CTS or RTS signal. Lastly, jumper 8 can be installed to allow SCI2 to be connected to COM2. Option header jumper 5 provides connections to certain I/O ports of MPC555 or access to user components by other signals. See Appendix C for connections of jumper 5. As a conclusion, the current FDR operates with the CME555 jumper and switch settings shown in Appendix D.

## **2.4 Ports and Connectors**

The LCD Port on the CME555 board provides a flexible connector to attach 80 character display modules, 160 character display modules and some graphics display modules with embedded drivers. For those specific applications of the FDR, an 80 characters display module is required to display all the data. The LCD and Keypad ports are memory mapped to chip select 3, or memory space 0x01000000 to 0x013FFFFFF. Serial communication is implemented on COM1 and COM2 serial ports. MPC555 provide SCI1 and SCI2 for transmitting and receiving data bits. The CME555 uses a transceiver to translate the TTL voltage levels to RS232 voltage levels for data transmission to peripheral devices. Software development on the CME555 is usually performed using a BDM tool connected to the on-board BDM port connector. This provides real time access to all hardware, peripherals and memory on the board. BDM software such as Metrowerks Codewarrior allows for high-level source code debugging. During the development of the current FDR, the BDM port is accessed using Wiggler from Macraigor Systems Inc. The new board design will also seek to use the BDM port as the interface for software development. Finally, the CME555 provides port connectors for all the MPC555 subsystems and signals. See Figure 3 for each of the ports available on board.

## **CHAPTER 3 EXTERNAL PERIPHERALS**

### **3.1 Signal Conditioning Unit**

As seen in Figure 1, the signal from the transformer is the input for the signal conditioning system. The signal conditioning system includes a voltage divider, a passive low-pass filter, and an output bias circuit. The system must first take the 12Vrms signal from the transformer and attenuate it down to 20V peak to peak. Then the filter takes the 20V peak to peak signal as the input and outputs the appropriate signal that is readable by the ADC. The filter provides anti-aliasing and gets rid of high frequency noise [9]. As a result, the ADC gets a clean analog signal for conversion. See Appendix C for circuit diagram of signal conditioning unit.

### **3.2 ADC Unit**

The AD976A is a high speed, low power, 16 bit sampling, analog to digital converter that can operate from a single +5 volt power supply. The AD976A takes standard +/-10 volt input range. With a 100/200kSPS throughput rate and a parallel interface [10], the AD976A is capable of connecting directly to microcontrollers. In the case of overall FDR system, (see Figure 1) the AD976A receives the voltage signal from the filter and samples 1440 times per second for the frequency algorithm [9]. Specifically, the AD976A uses the R/C' (Ready/Convert') line as a trigger for conversions and it is driven by a PWM signal from MIOS port of the MPC555. There may be inaccuracies or drift in the MPC555's oscillator output but this problem can be corrected by measuring the one pulse per second (PPS) signal from the GPS receiver using the MDASM subsystem [14]. The AD976A does require some extra circuitry to operate. Some resistors and capacitors are needed to connect to its input and reference pins. See Figure 7 [10] for a functional diagram and see Appendix C for schematic. The AD975A is parallel-interfaced to the MPC555 through the GPIO subsystem and there are two control signals, one is R/C' that trigger for conversion and the other is a inverted BUSY' signal that is generated from the MPC555 IRQ[1] pin.

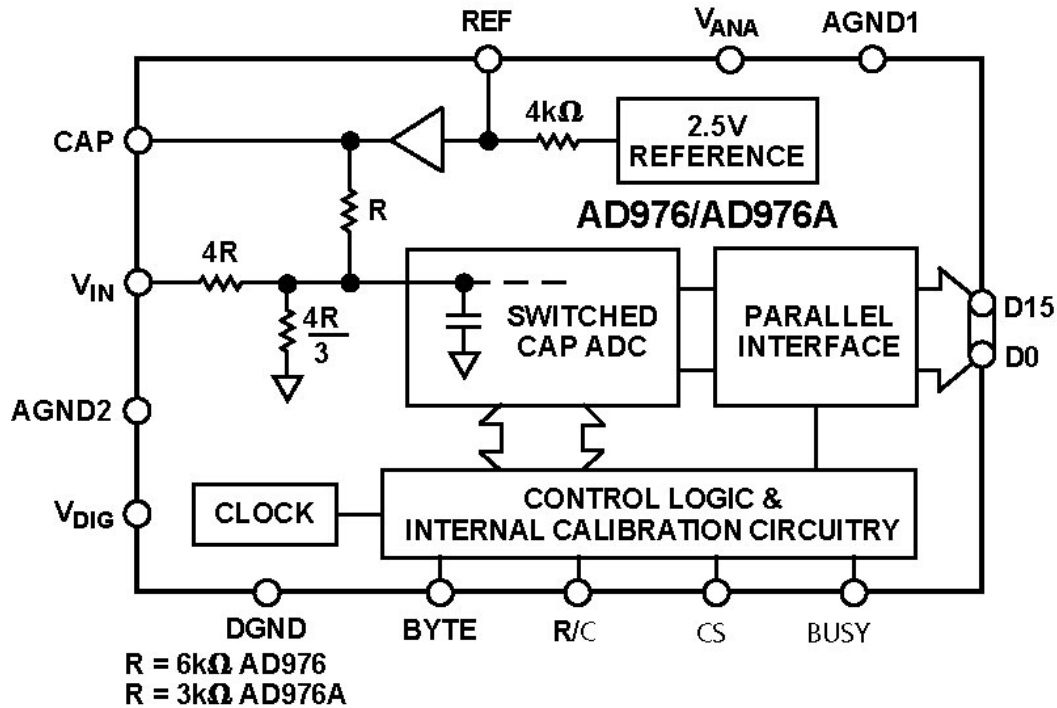


Figure 7 - Functional diagram of AD976A

### 3.3 Motorola M12+ Receiver

Time synchronization for FDRs is provided by the Motorola M12+ GPS receiver. The synchronized signal of GPS is the 1PPS pulse train where the rising edge of the pulse indicates the second change of UTC (Coordinated Universal Time) time and the pulses that triggers analog to digital conversion should be in phase with the 1PPS. The M12+ does not only provide the 1PPS but also a timestamp for each second of time. The GPS software will format each timestamp to be the current UTC, in seconds referenced to midnight on January 1, 1970. The timestamp is then provided to the TCP/IP communications interface where it is attached to each frequency estimate.

The M12+ GPS receiver uses a 12 channel parallel receiver design and operates on a +3Vdc power supply. See Figure 9 [39] for a functional diagram. In addition, the receiver uses 3V CMOS/TTL serial interface for communication to host equipment. Timing accuracy for M12+ [4] is list below:

200s TTF (Time To First Fix)-cold (with current almanac, position, time and ephemeris

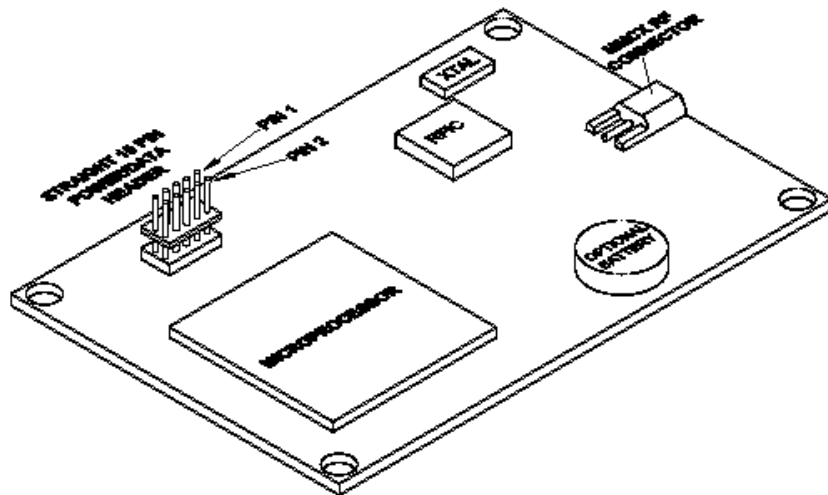
50s TTFW-warm (with current almanac, position and time)

25s TTFW-hot (no stored information)

< 1.0s internal reacquisition

Positioning accuracy is within 25 meters with Selective Availability (SA) disabled [4].

The M12+ also features straight 10-pin power/data header for low-profile flat mounting against host circuit board [41]. Finally, the optional on-board Lithium battery allows for saving setup information and increasing the speed of satellite acquisition when the receiver is powered up after a period of inactivity. Finally, Figure 8 [38] shows the bottom view of M12+ GPS receiver.



**Figure 8 - Bottom view of M12+ receiver**

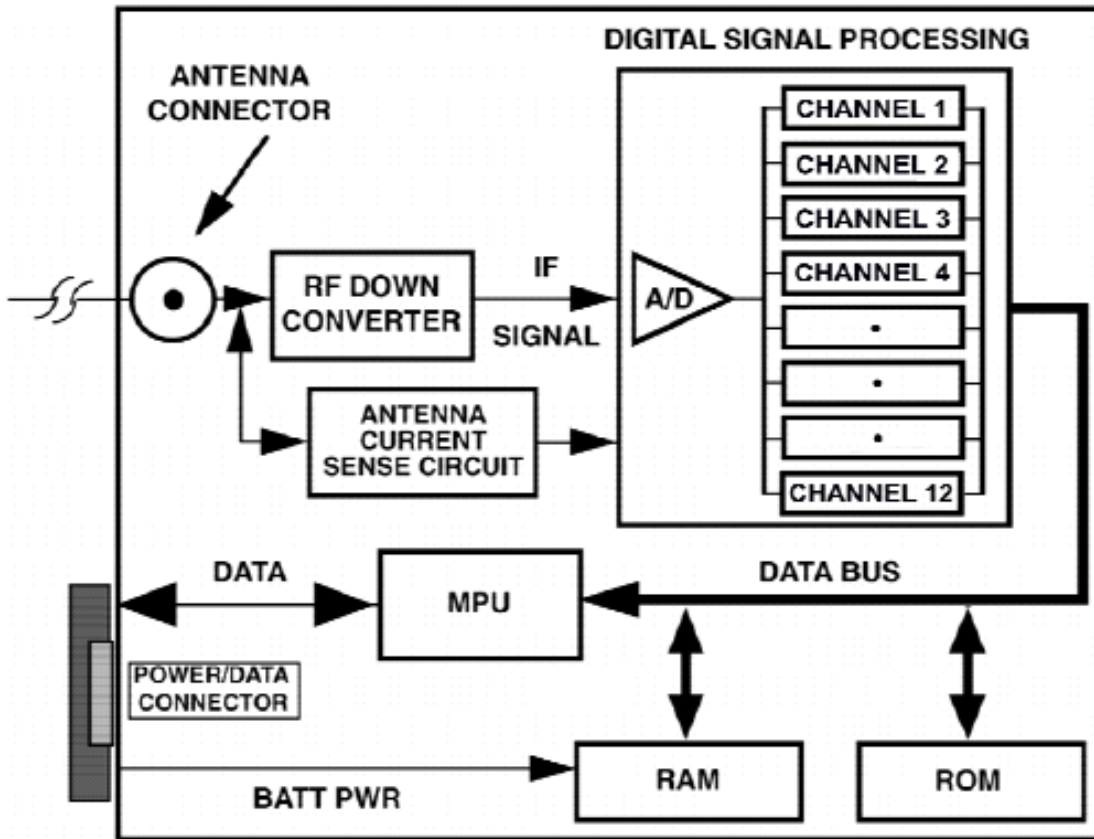


Figure 9 - Functional diagram of M12+ receiver

### 3.4 Serial to Ethernet Converter

In order to transmit data to the central server, there is a need for a TCP/IP device. Since there is no Ethernet port on the CME555, there needs to be a serial-to-Ethernet converter. The device that is currently used for serial-to-Ethernet conversion is the Moxa DE-311. The DE-311 provides a DB9 serial interface to the CME555 board and an RJ-45 Ethernet interface to a TCP/IP network. Furthermore, DE-311 uses CTS/RTS handshaking to accommodate for the binary data sent to the server.



## **CHAPTER 4**

### **FDR SOFTWARE**

The FDR has 3 states of operation: acquisition, initialization, and collection. In the acquisition phase, the GPS receiver gets a hold of the GPS satellites and receipt of UTC time. Since all the calculations are synchronized to the UTC time, the FDR unit will not proceed until a good GPS signal is established. The threshold for determining if a GPS signal is established is the receipt of 4 one-second time pulses from the GPS receiver [14]. An internal timer within the MPC555 is used to verify the accuracy of the pulses. In the initialization phase, the GPS signal has already been received and is providing accurate UTC time. The FDR begins collecting signal measurements to seed frequency algorithm. This initialization process takes up 1 second so that synchronized frequency measurements can be made at the start of every second afterwards [14]. If the GPS signal is lost during this stage, the FDR will go back to the acquisition state and discard all measurements collected. This makes certain that the FDR is using only current data with known times. Once the frequency algorithm has started, the FDR is able to generate frequency estimates in a continuous fashion until there is a loss of power or GPS signal. If GPS signal is lost, the FDR will go back to the acquisition phase and discard the current phasor and frequency calculations. A flow chart of FDR initialization procedures as well as the main program is displayed in Figure 10 [14] and Figure 11 [14] respectively.

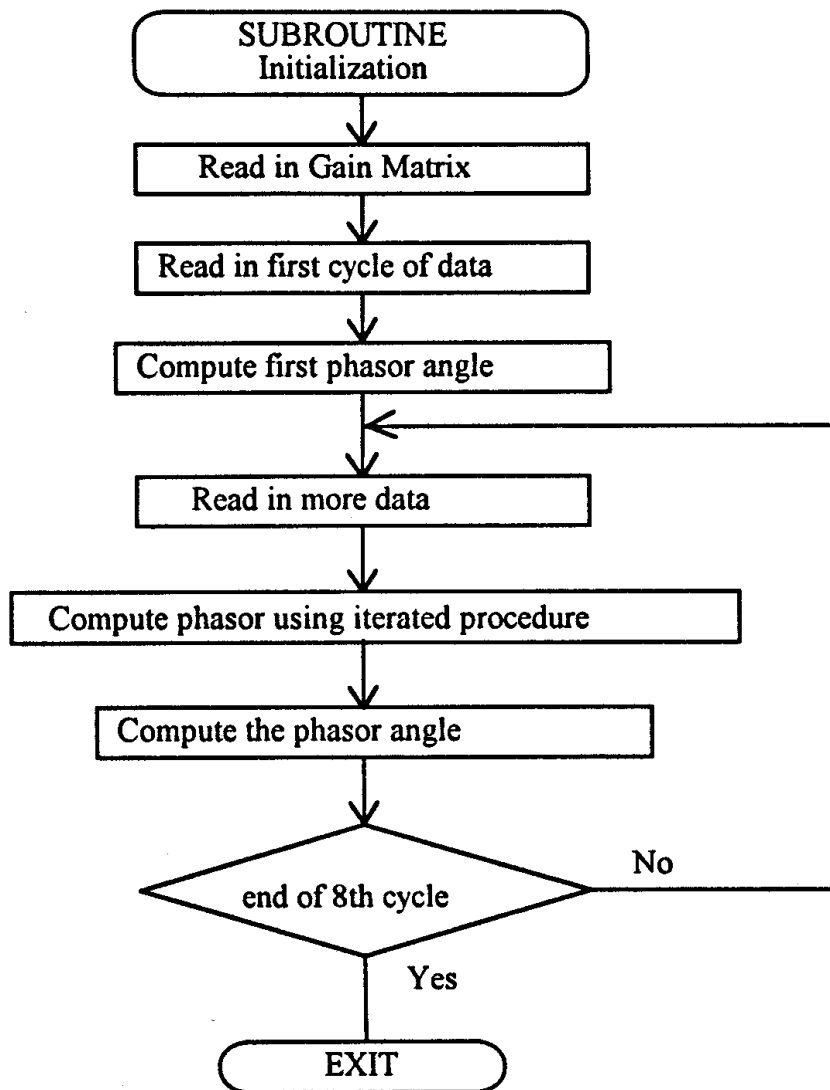
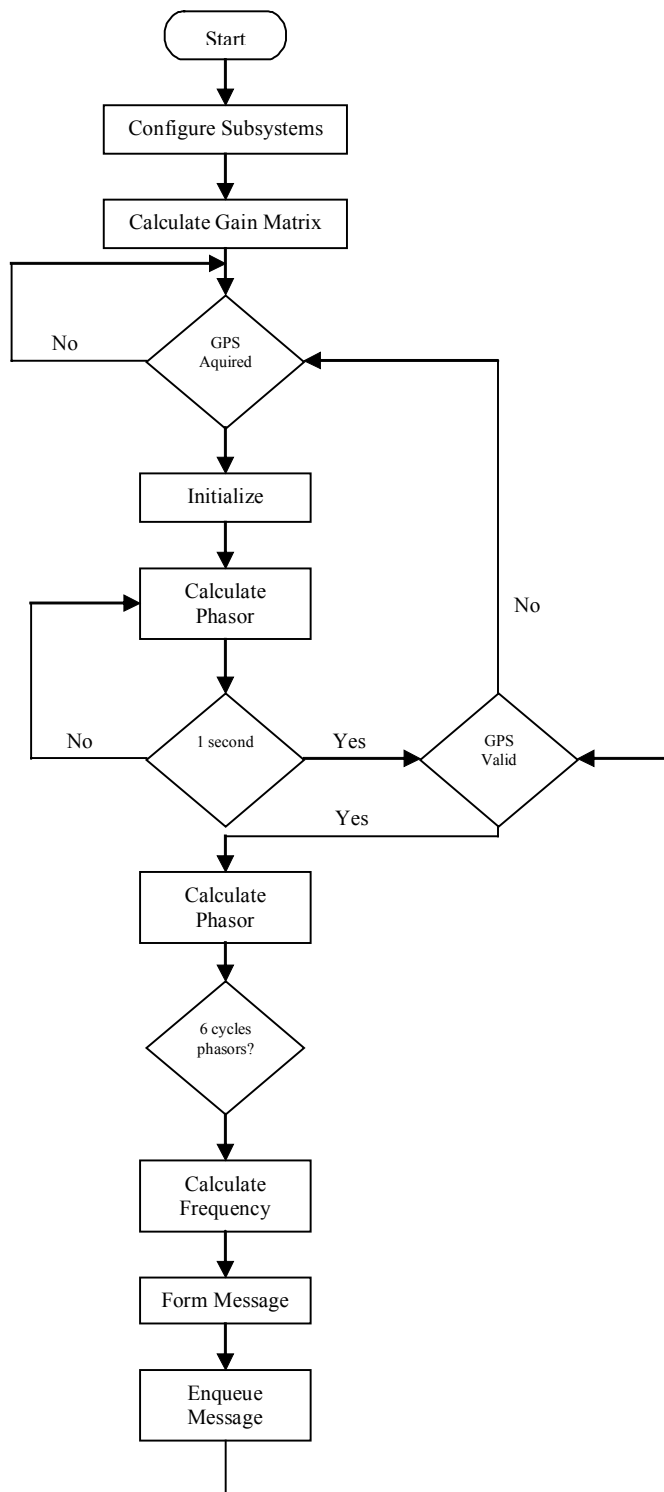


Figure 10 - Flow chart of algorithm for FDR initialization



**Figure 11 - Flow chart of algorithm for main program**

## **CHAPTER 5 PAST AND RELATED EFFORTS**

### **5.1 Protel**

To start the PCB design, it is necessary to create the circuit schematic. Once the schematic is developed, the PCB layout can be established. With the aid of modern day design software, circuit design becomes trivial once the concept is recognized. There are many circuit design software in the market today and the one that is used very prominently for PCB design is Protel DXP. As a result, a brief overview of the Protel software is needed before starting the FDR PCB design.

Protel DXP from Altium is a single application that provides all the functionalities needed to take a design concept to a board level design. Therefore, Protel is used to generate all of the design schematics shown in this thesis. Many of the functionalities of Protel are provided specially for PCB design. However, for the purpose of this research, the schematic editor was used most extensively. At the preface of PCB design, the circuit schematic is the first step to the final PCB design. Included with Protel DXP is a comprehensive set of integrated component libraries that combine each schematic symbol with their respective PCB footprints [16]. So once the schematic of the PCB has been established, the board layout can be completed fairly quickly. Protel's PCB editor allows for specifying the board requirements and a hierarchical design rules and DRC system allows the control of all aspects of board design [16], verifying that the design is correct from start to finish.

### **5.2 MPC555 Support Pins**

Design for FDR PCB started with a preliminary study of the 272 pins on the MPC555 chip. In order to establish a circuitry for the PCB, it is crucial to know which signals from the MPC555 are needed for FDR operations. Once the essential signals are known, those signals that are not essential to the operation of FDR will also become obvious. In general, the beginning efforts were concentrated on separating the MPC555 signals that are needed and not needed. The essential MPC555 signals include those that are needed to support the operation of MPC555 chip as well as the FDR. However, it is obvious that the internal QADC and TouCAN subsystem are not needed for the FDR.

The rest of the MPC555 subsystems are needed for operation of FDR in some manner. Before delving into the external interface for the MPC555, it is important to go through a detailed description of the MPC555 support signals and how they should be setup for proper operation of the chip. Some of the pins may have to be pulled high or low with resistors to keep electromagnetic interference from interfering with operation of the chip.

The MPC555 support signals include the clock, RESET, RESET configuration word, power supply, JTAG/BDM(background debug monitor) and bus control. The RESET pins consist of PORESET, HRESET and SRESET. Power on Reset (PORESET) initializes everything and is often used as a reset input when power is lost to the chip [25]. PORESET pin must be driven by an external source and cannot be left floating [13]. Hard Reset (HRESET) initializes registers other than those kept alive by keep alive power [25] and is used by Macraigor Wiggler to reset the processor via the background debug monitor connector. HRESET pin must be pulled up to 3.3V with a 10K pull up resistor [13]. Soft Reset (SRESET) can be used as a CPU output to memory or/and peripherals reset input pin [25]. It is recommended for SRESET to be pulled up to 3.3V with a 10K pull up resistor [13].

There are three options to use RESET configuration words (RSTCONF) [29]. But the CME555 board is setup to read the reset configuration from the internal flash shadow array [1]. The new design will keep this characteristic and the RSTCONF pin is pulled up through a 10K resistor to 3.3V (CMFCFIG register bit HC is programmed to 0) [29].

There are 5 support signals for MPC555 clock. External clock input (EXTCLK) [26] pin should be connected directly to the ground since there is no need for external clock input [13]. Engineering clock output/Backup clock output (ENGCLCK/BUCLK) pin is the engineering clock output, and the backup clock is operated when the chip is in limp mode, which enables a less precise on-chip oscillator to allow the system to continue minimum functionality until the system clock is fixed [26]. ENGCLCK/BUCLK pin should be pulled low with a 10K resistor to ground to reduce electromagnetic interference (EMI) [13].

XTAL and EXTAL pins are used to connect to either a 20 MHz or 4 MHz external crystal for the internal oscillator circuitry [26]. The FDR circuitry will use a 4

MHz external crystal connected to XTAL and EXTAL pins to reduce electromagnetic interference emission from the oscillator [13].

The external filter capacitance (XFC) pin is an input line for an external capacitor filter for the PLL circuitry [26]. One terminal of the capacitor is to be connected to XFC and the other terminal is to be connected to VDDSYN. The capacitor value is calculated as [26]:

$$0 < MF + 1 < 4 \quad (680 \times (MF + 1) - 120) \text{ pF} \quad \text{Equation 1}$$

$$MF + 1 \geq 4 \quad 1100 \times (MF + 1) \text{ pF} \quad \text{Equation 2}$$

Where MF is one less than the desired frequency multiplication factor

For a 4 MHz XTAL the multiplication factor would be 10 (40 MHz operation) and one less of multiplication factor MF is 9. According to Equation 2, capacitance yields 11000 pF or 11 nF, which results in a 10nF capacitor to be implemented. Finally, the Clock out (CLKOUT) pin represents the external bus clock [29], and it is used to provide clock pulses to the EPLD on the CME555 board.

Joint Test Action Group (JTAG) [31]/ Background Debug Monitor (BDM) is a functional group within MPC555 that allows for uploading and debugging of FDR software. To permit compatibility of the firmware for FDR and compiler, the JTAG/BDM subsystem has to remain intact in the FDR board. MPC555 allows for three possible pin outs of the BDM connector to meet the various needs for different applications. Option 1 allows for maximum debug capability, which is recommended by Motorola and used in the CME555 board [13]. Option 2 gives the maximum external bus capability, which is the default power on reset condition [13]. Option 3 maximizes the input/output configuration [13]. In the case of FDR development, option 1 would yield the most efficient solution since it allows for compatibility to the current debugger hardware that is used, or the Wiggler.

Bus arbitration on the MPC555 is handled by 3 pins, specifically Bus Grant (BG), Bus Request (BR) and Bus Busy (BB) [28]. Bus Grant indicates external data bus status; it is low when the bus is granted to the MPC555 [28]. BG pin should be pulled low if the board is to have only a single chip system [13]. Bus Request indicates that the data bus

has been requested for external cycle [28]. Bus Busy indicates that the master is using the bus [28]. BR and BB pin should be pulled up to 3.3V if the board is a single chip system and pulled up with 10K resistor to 3.3V otherwise [13].

Power supply pins makes up the most pin counts on the MPC555. Global power supplies provide various voltage levels to the MPC555 for internal logic operation. Also there is the CDR MoneT flash (CMF) [32] power supply for the flash memory operation [27]. Since these power supplies are crucial to the internal operation of MPC555 and they do not affect interfacing with other components, the power supply circuit should be kept similar to that of CME555. However, special attention should be focused on the CMF power supply since CME555 has a large number of switches for configuring the flash memory. The FDR board should keep number of configuration switches to the minimum. Specifically, the FDR board needs just one jumper for providing power supply to the internal CMFI Flash memory. Once installed, the jumper enables +5V DC for programming internal CMFI flash memory and it should be removed when not programming to protect the internal flash memory during power cycles [32].

At this point, the internal support signals of MPC555 have been covered. The rest of the signals deal with external interfacing to other devices. There are a total of 8 external interrupt service pins on the MPC555 [3]. The interrupt service pins can request service routine by internal interrupt controller. However, the FDR only uses one interrupt service signal for triggering the analog to digital converter [14]. All of the interrupt service pins should be pulled up to +5V with 10K resistors to allow for programming as IRQs [13].

For the rest of the external interfacing signals, the unused bi-directional IO pins for MDA, PWM, GPIO and TPU should be pulled up with a 10K resistor to 5V [13]. But if electromagnetic interference becomes a major consideration, then pulling the unused pins low with a 10K resistor will reduce emissions [13]. In the FDR design (Figure 1), there is only one pin needed from MDA, one pin from PWM, one pin from IRQ and 16 GPIO pins from MIOS subsystem.

Finally, to summarize the preliminary study of MPC555 support pins, a schematic was drawn in Protel to show the MPC555 connections based on the study of the support pins (see Figure 12). As seen in Figure 12, some of the MPC555 pins are not used for the

operation of FDR and thus were left floating. The preliminary study has suggested connections for some of these pins to reduce electromagnetic interference [13] or provide

**Table 1 - Summary of MPC555 Support Signal Connections**

<b>Pin</b>	<b>Connection</b>
PORESET	Cannot left floating, needs to be driven by external source
HRESET, SRESET, RSTCONF	Pulled up to 3.3V with 10K resistor
EXTCLK	Connect to ground with pull down 4.7K resistor
ENGCLK/BUCLK	Pulled low with 10K resistor to ground
XTAL/EXTAL	4MHz crystal, circuit setup same as CME555 (see Appendix C)
XFC	Place 10 nF capacitor between XFC and VDDSYN 4700 pF capacitor for XFC? (larger capacitor result in stable clock, time to lock will be longer)
BDM/JTAG	SRESET, HRESET, VFLS0, DSCK, VFLS1, DSDI, DSDO connection according to CME555 schematic (see Appendix C)
IRQ[1:7]	Pulled up to +5V with 10K resistor
Unused MDA, PWM, GPIO and TPU	Unused bidirectional I/O pulled up with 10K resistor to 5V.
BG'	Pulled low
BR', BB'	Pulled up to 3.3V without resistor to allow for single chip system
EPEE	Pulled up to 5V
VDDH[1:10]	Connect to 5V supply
VDDL[1:8], VDDI[1:4]	Connect to 3.3V supply
VPP	5V applied when programming flash, use jumper connection
VDDSRAM	Connect to 3.3V with diode
VDDSYN VSSSYN	Filter circuit to keep noise low, uncertain Use CME555 connection (see Appendix C)?
ETRIG[1:2]	Connect to ground
TOUCAN pins	Uncertain, floating?
VRH	+5V OR +3V supply
VRL, VSSA	Connect to ground
ADC pins AAN[0:59], BAN[0:59]	Uncertain, left floating?
Unused WE[0:3], CS[0:3]	Uncertain, left floating?
Bus control pins *	Uncertain, left floating?
VDDA	Uncertain, left floating?
ECK	Pulled low with 10K resistor
GND	Connect to ground

\*Bus control pins include BURST', TSIZ[0:1], AT[0:3], RSV, PTR, RETRY', BDIP', TS', STS', CR, KR', TA', TEA', BI'



the support logic needed by MPC555. To conclude this study, Table 1 [13] is compiled for the possible connections of MPC555 support signals. A detailed schematic for MPC555 connection is shown in Figure 12. As seen in Figure 12, most of the unused pins are pulled up to 5V supply with 10K resistors. These include the unused data and address pins, unused MDA, PWM, QGPI and TPU pins. However, some unused pins are still left floating due to uncertainty. These include the pins that support the QADC subsystem, TouCAN subsystem, and most of the bus control pins. Since all of these pins are used for input/output operations, they are susceptible to EMI and it is uncertain whether they should be pulled high or low. Nevertheless, to speed up the process for developing a new PCB, the research work has turned to a new direction. Instead of dwelling on the study of MPC555 signals, it would be more effective to take a step forward in the PCB design and start the implementation.

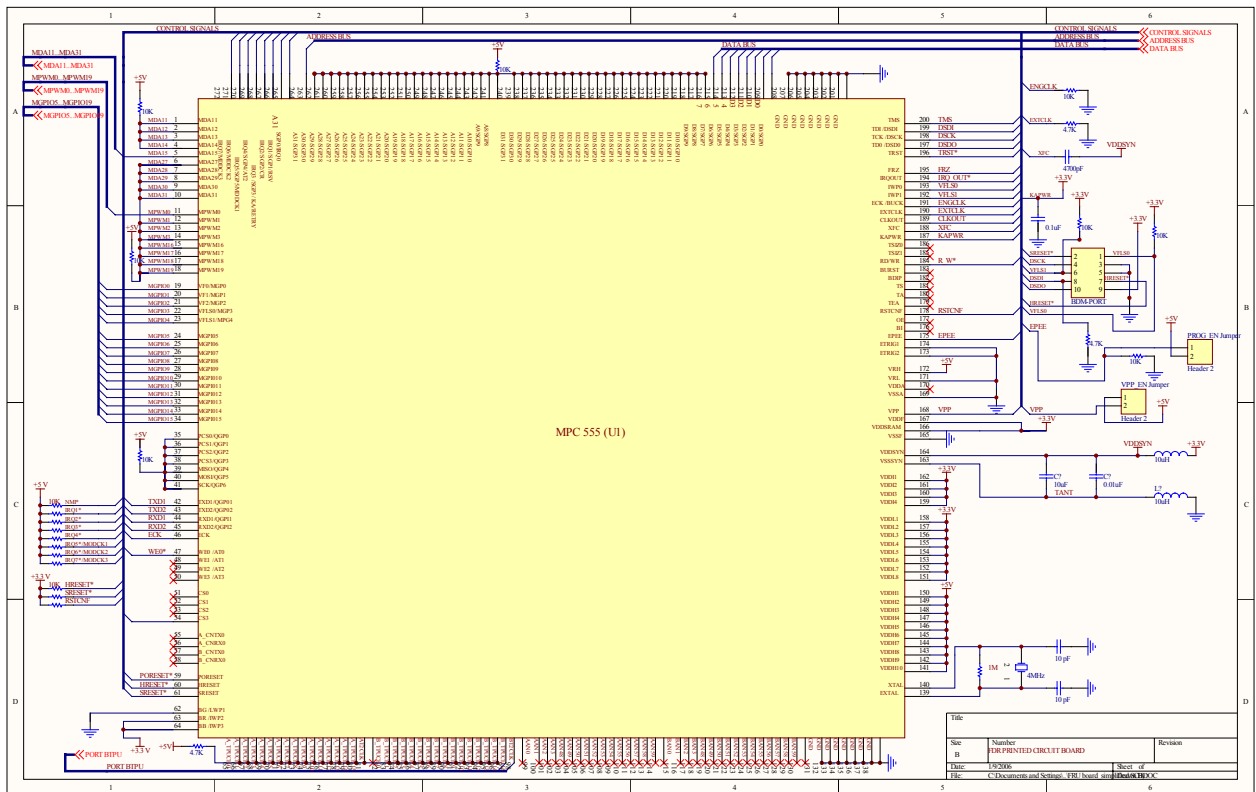


Figure 12 - Preliminary MPC555 pin configuration for FDR board design

## **CHAPTER 6**

### **PRINTED CIRCUIT BOARD DESIGN – PB555 PROTOTYPE**

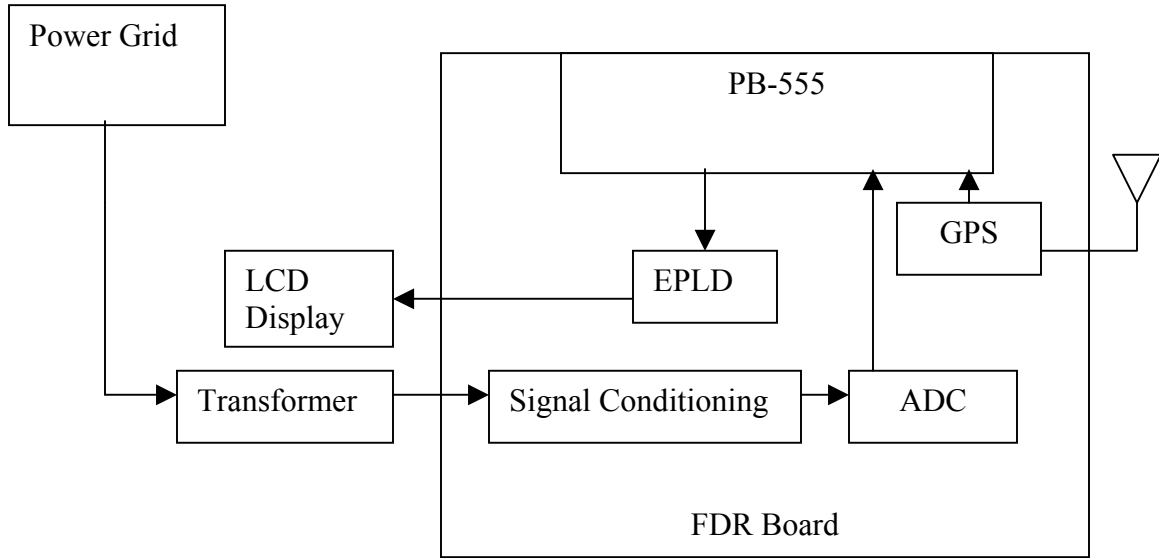
#### **6.1 Prototype Proposal**

Initial PCB implementation started with a simple development board from Axiom. This approach to PCB prototyping was aroused from the uncertainty with MPC555 pin connections. Instead of concentrating on the details of MPC555 pins and the external circuitry that is needed for operation, the research effort has been shifted to the use of a simpler development board for a PCB prototype. The idea for using a simpler development board is driven by the need for a prototype strictly for PCB development. The development board to be used for prototyping needs to have the MPC555 chip mounted and be simple in circuitry. With a simple MPC555-based development board, there would be no need for going through all the support pins for the chip and the design process would move forward to interfacing of FDR components.

One ideal candidate for this prototype development is the PB555 from Axiom. With the introduction of PB555, the concept for PCB design became clear. The proposed final PCB design has the functional diagram shown in Figure 13. Since the MP555 needs to be interfaced to several external devices, there will be some tasks to be completed in order to establish the final version of FDR PCB. The tasks are listed as follow but not necessarily in this order:

- Interface the PB555 board to LCD
- Interface the PB555 board to GPS
- Interface the PB555 board to ADC and signal conditioning unit

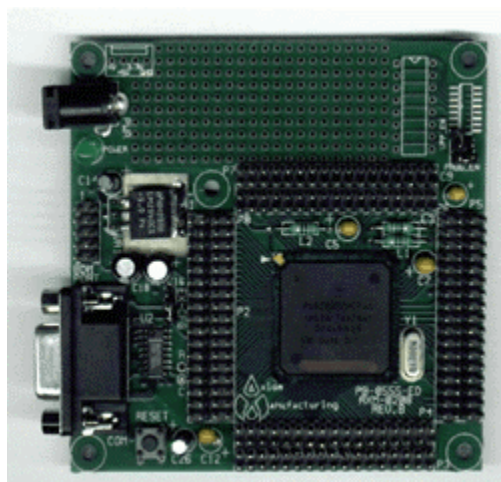
The important aspect of this design approach is by breaking down the problem into parts, the problem can be approached in a more efficient manner. Each of the tasks listed above needs to be done separately to avoid wiring complexity and when all of the tasks have been successfully implemented, a final PCB design can be established.



**Figure 13 - Block diagram of proposed PCB design**

## 6.2 PB555 board

The Axiom PB555 is a Motorola MPC555 module with I/O pin headers, oscillator circuit, power supply and one RS232 port [2]. In addition, the PB555 has the same BDM real-time debugger as the CME555. The PB555 was chosen for the prototype design of PCB because of its similarities to the CME555 board.



**Figure 14 - Picture of PB555 board**

Since one of the requirements of the new board design is to have software compatibility with the CME555, it is important for the new PCB to be as similar to the CME555 as possible. However, the PB555 has some drawbacks in terms of communication interfaces. The PB555 has only one RS232 port and no LCD port [2], which indicates that the GPS and the LCD interfaces need to be created in some manner. The ADC and device server can be interfaced to the PB555 board just like the CME555 since there is no need for additional external circuit. As a result, most of the research work is concentrated on interfacing the LCD and GPS to the PB555 board.

At the same time, it is also important to take into account the software compatibility issues. Specifically, the memory map of PB555 board is a little different from the CME555 board. Figure 15 [2] shows the memory map for PB555 board, it is important to note that there is no external memory involved and the board monitor utilities are located in memory space 0x00000000 to 0x0000FFFF [2]. Therefore, care should be taken when writing to the on-chip Flash memory so that the monitor utilities would not be accidentally erased.

The most critical aspect of this design is moving the code from external memory to the valid memory space within the MPC555 chip. Consequently, the software that was used for the CME555 board needs to be modified so that it can operate within the on-chip memory.

0000 0000	<b>Monitor Utilities</b> – CMFI flash memory Debug monitor and CMFI programming tools	64K
0001 0000	<b>User Program Memory</b> – CMFI flash memory Can be used for non-volatile program and data storage  The utilities also re-map interrupt vectors here starting at 0x10100 for the RESET vector and continue to 0x12000	380K
0007 0000	Unused	
002F C000	Control and Status Registers	
0030 2000	<b>User RAM</b> –DPTRAM Array	6K
0030 3800	Control and Status Registers	
003F 9800	<b>Monitor Utilities RAM</b> – SRAM Array Used by monitor and CMFI programming tools	2K
003F A000	<b>User RAM</b> – SRAM Array	24K
0040 0000	Unused	

**Figure 15 - PB555 Memory map**

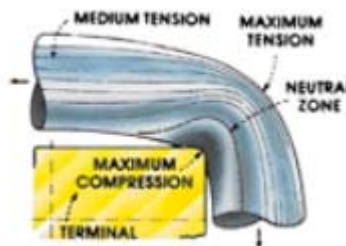
### 6.3 Prototype Development

Once the development board is specified, the next step is to select a method for connecting the board to external devices. Since the PB555 board does not have any connectors for the MPC555 signals, connecting the board to external devices becomes a challenge. The most frequently used method for electrical interconnection is soldering, which does require a certain degree of aptitude and a large amount of time to do well, thus the idea was abandoned quickly. Another solution was proposed instead and it was accepted as the best way to approach the prototype connections. Wire wrapping has been in existence for over fifty years [11] and it is used most significantly for interconnection on PCBs.

Wire wrapping was originally developed in the late 1940's by Bell Labs in the United States. Up until that time, soldering was the commonly used method for cable interconnection. Most importantly, soldering presented many problems in electrical interconnections [11]. Solder joints can easily fail if the soldering process is not just right and it is quite messy to attempt rework of a defective one. Mechanically it is brittle, weakens with age and varies widely in strength from connection to connection [11].

With the introduction of wire wrapping into the electronics world, the problems presented by soldering were quickly diminished. Wire wrapping is a process for connecting stripped copper wire to a terminal to form an electrical circuit. The wires are tightly coiled around the terminal (see Figure 16 [11] and Figure 17 [11]) through the motion of a rotating equipment, called a bit. The bit is produced from solid steel rods and has a large central hole along its axis for slipping the tool over the terminal. The terminal will remain inside this cavity as the bit rotates. The top surface of the bit has a wire slot, which accepts the length of stripped wire that is to be wrapped. While the bit is in operation, it rotates and pulls the wire across the sharp corners of the terminal.

This process gives the wire a large amount of tension and compression, typically 100,000 PSI at the terminal's edge [11]. The wrapping action causes both the wire and terminal to diffuse. This results in a connection of high mechanical strength with very low electrical resistance. It also allows a gas-tight connection with a design life of 40 years [11], yet the wires can be easily unwrapped to correct any wiring errors or to replace the equipment to which the wire is terminated. This high level of reliability and environmental resistance makes wire wrapping a preferred method for building the PB555 prototype.



**Figure 16 - Wire wrapping - View from the top of terminal**



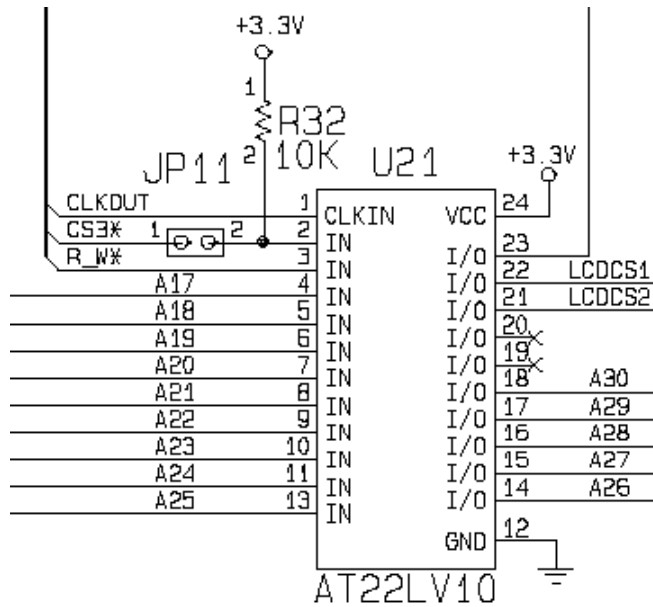
Regular

**Figure 17 - Wire wrapping - View from the side of terminal**

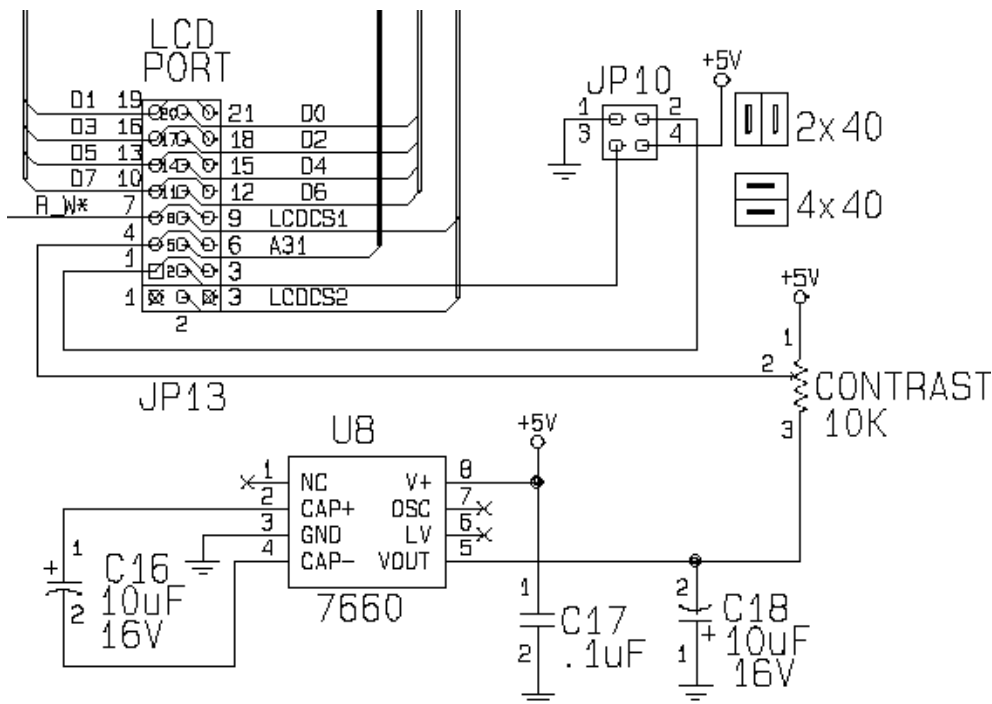
#### **6.4 LCD Interface**

To determine how the LCD should be interfaced to the PB555, a study was conducted to see how this device was interfaced to the CME555. The study indicates that the LCD is interfaced by memory mapped I/O with the use of an EPLD (Electrically Programmable Logic Device) chip for address decoding. See Figure 18 [22] and Figure 19 [22] for LCD interfacing in CME555.

In memory-mapped I/O the LCD device has an address just like a memory location. The MPC555 has its lower 14 address lines going into the AT22LV10 for address decoding. When the processor needs to read or write to the LCD, it can select the memory location specifically for the LCD. According to Figure 18 [22] the EPLD AT22LV10 generates a LCD chip select signal that makes sure the LCD chip is selected when it is intended. Since the LCD chip will always be memory mapped to CS3 [1], CS3 should always be connected to AT22LV10. The CLKOUT line from MPC555 generates the clock pulses and the R/W\* line provides the read/write control signals for the EPLD. As a result, the AT22LV10 generates two chip select signals for the LCD and one chip select for Keypad ports [1].



**Figure 18 - Schematic of address decoding circuitry for LCD on CME555**



**Figure 19 - Schematic of LCD port connection and contrast circuit on CME555**

For the LCD display connection, there is a LCD port on the CME555 board, which is connected according to Figure 19 [22]. The LCD port on CME555 can be connected via either to the left or the right two columns of pins. Actual connection is only made for 14 pins since there is connection between signals on either side of the port. The



current FDR uses a ribbon cable for making the connection between the port and the display. To clarify the connections, a mapping of signals between the LCD port and the display is given in Table 2. Table 2 shows where the signals are generated for each

**Table 2 - Mapping of LCD port pins to MPC555 signals**

CME555 LCD port (Pin number)	MPC555 signal	DMC20261NY-LY-AXE display (Pin number)	DMC20261NY-LY-AXE display (pin name)
1	+5V	1	V <sub>SS</sub>
2	Ground	2	V <sub>DD</sub>
4	**	3	V <sub>EE</sub>
5	A31	4	RS (Register Select)
7	R/W'	5	R/W (Read/Write)
8	LCDCS1*	6	E (Enable)
10	D7	7	DB0
11	D6	8	DB1
13	D5	9	DB2
14	D4	10	DB3
16	D3	11	DB4
17	D2	12	DB5
19	D1	13	DB6
20	D0	14	DB7

\* Generated from EPLD

\*\* Generated from power supply circuit

pin the LCD. Note that the register select line is generated from the least significant address line of the processor, R/W signal is connected to the R/W\* from MPC555 and Enable is generated from LCD chip select 1.

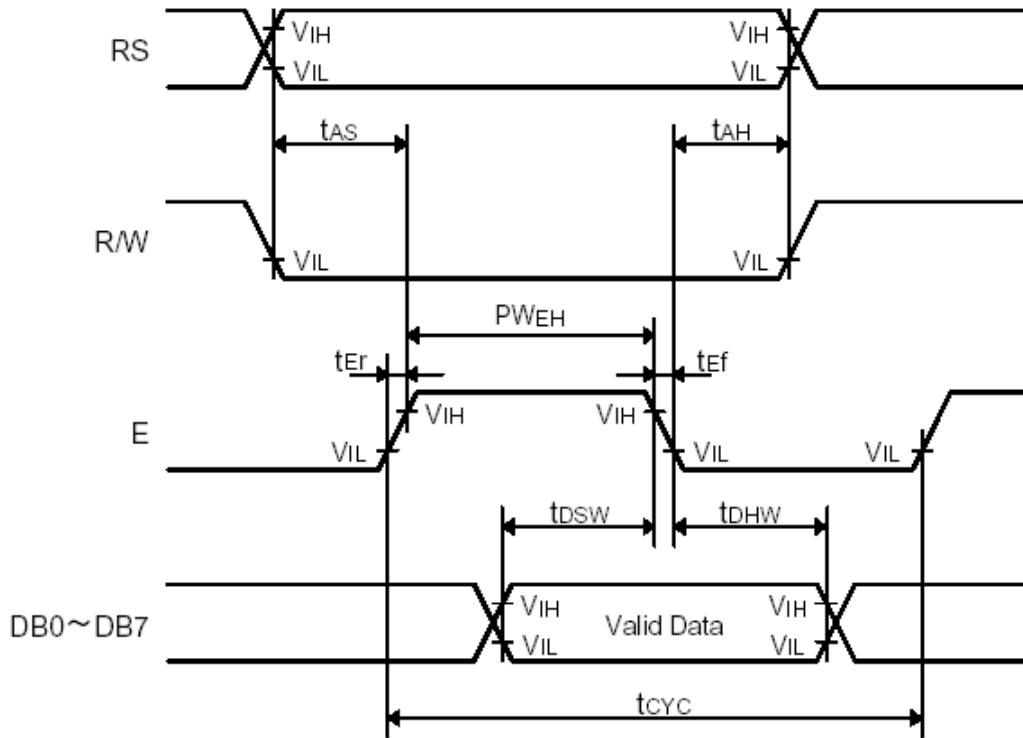
To better understand how the LCD operates, it is important to go through an analysis of the timing diagrams specified by the LCD. Figure 20 [5], Figure 21 [5] and Table 3 [5] shows the timing diagram and specific timing constraints for each signal entering the LCD. Figure 20 illustrates signal timing when there is a write operation on the LCD and Figure 21 shows what signal timing when there is a read operation on the LCD. In order to read or write to the LCD, the processor must provide stable data within the time limits shown relative to the clock.

Table 3 indicates that for the write data operation, minimum setup time is 80 ns. So the time period for when there is valid data on the data bus to when Enable signal starts to fall has to be at least 80ns. Likewise, Table 3 also indicates that the minimum time period for after the fall of Enable signal to the end of valid data has to be 10 ns.

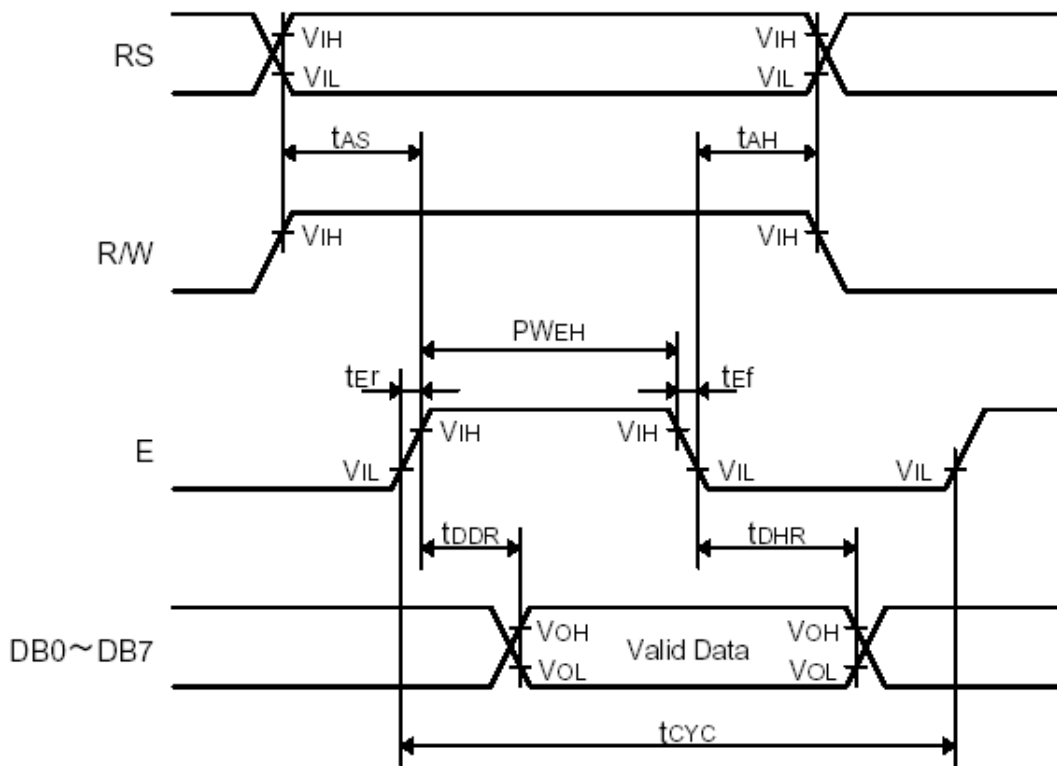
Consequently, these two criteria need to be fulfilled when writing to the LCD. When reading data from the LCD, the maximum delay time or the maximum time it takes from the rise of Enable signal to when there is valid data on the data bus has to be 160 ns. Again, the minimum hold time for reading data is 5 ns. As a conclusion, there are a total of four timing criteria to be fulfilled when writing or reading to the LCD.

**Table 3 - DMC20261NY-LY-AXE timing constraints**

Parameter	Symbol	Minimum	Maximum	Units
Enable Cycle Time	$t_{CYC}$	500	-	ns
Enable Pulse Width	$PW_{EH}$	230	-	ns
Enable Rise/Fall Time	$t_{ER}, t_{EF}$	-	20	ns
Address Setup Time	$t_{AS}$	40	-	ns
Address Hold Time	$t_{AH}$	10	-	ns
Write Data Setup Time	$t_{DSW}$	80	-	ns
Write Data Hold Time	$t_{DHW}$	10	-	ns
Read Data Delay Time	$t_{DDR}$	-	160	ns
Read Data Hold Time	$t_{DHR}$	5	-	ns



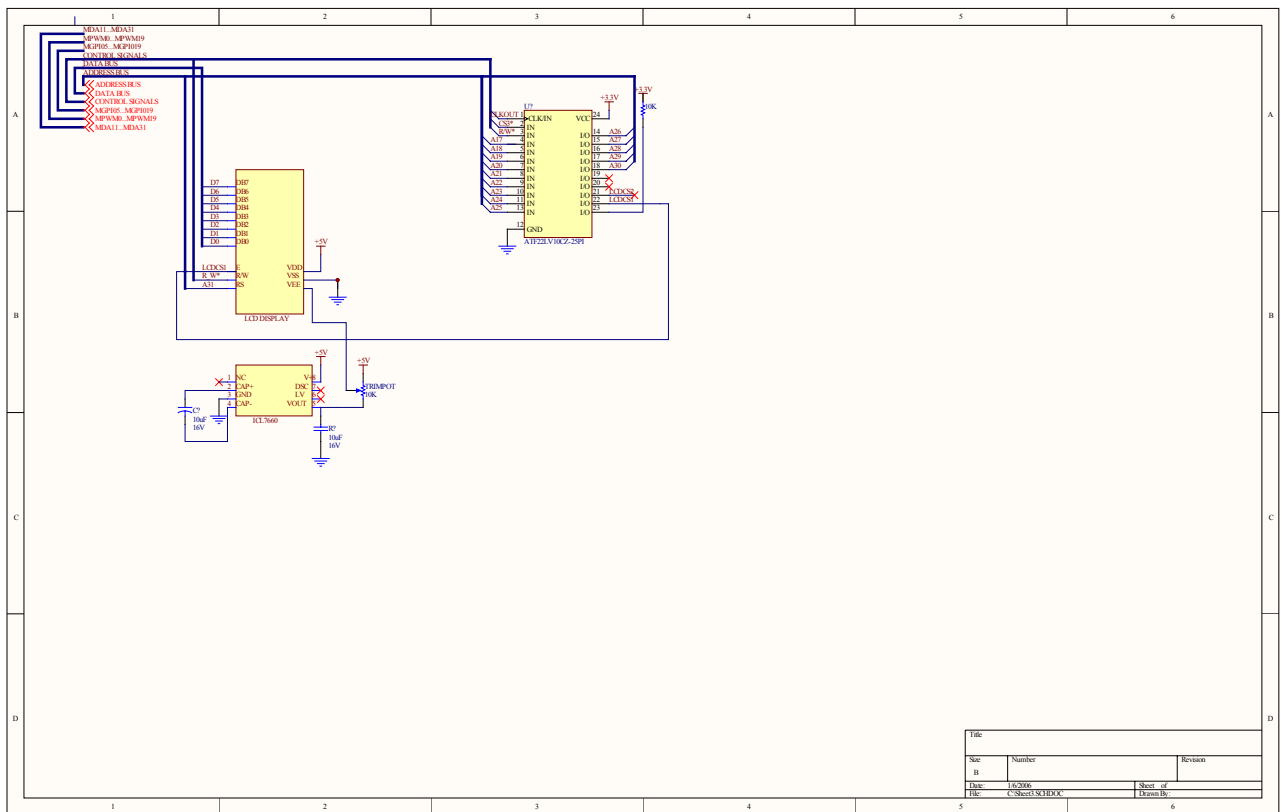
**Figure 20 - Write Operation Timing for DMC20261NY-LY-AXE**



**Figure 21 - Read Operation Timing for DMC20261NY-LY-AXE**

After the analysis of the timing criteria for reading/writing operation to LCD, it is necessary to isolate the AT22LV10 EPLD and study how it functions as an address decoder. Since the AT22LV10 was implemented on the Axiom CME555 board, the EPLD software is available to the public for development. See Appendix B for the EPLD code in PLD format. The EPLD codes basically setup the pins for input/output and implements the decoding logic using the R/W\* strobe, address, and chip select signals.

The next step was to take action in the implementation of the PB555 prototype. As indicated earlier, the FDR software compatibility issue comes into play if the design is differentiated from that of CME555. To speed up the process of building the prototype and be able to reuse the same FDR software, it was decided the LCD interface that is to be implemented on the PB555 board to be the same as on the CME555 board. Moreover, the address-decoding scheme that was used in CME555 is implemented the same way in



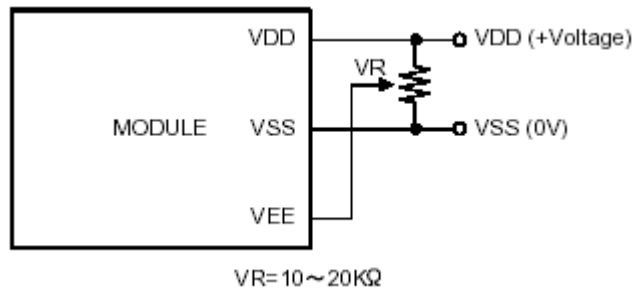
**Figure 22 - Schematic of LCD interface for PB555 board**

the PB555 prototype. See Figure 22 for a schematic of LCD connections of the PB555 prototype. Figure 22 shows that the address decoding is implemented with Atmel ATF22LV10CZ, which is a new version of the AT22LV10 chip. The AT22LV10 chip

that was used on the CME555 board has become old and is not supported by most of the modern EPLD programmers. Therefore, this particular EPLD is to be replaced by the ATF22LV10CZ, which is fusemap and pin-to-pin compatible with the AT22LV10 [6]. According to Atmel, the ATF22LV10CZ has pin-keeper circuits on all of its input and I/O pins while the AT22LV10 does not, so as long as the outputs are not considered when they are tri-stated, the pin-keeper should not affect the decoding logic. In effect, by using the same AT22LV10 software from Axiom (See Appendix B), the address decoding should work in the same way as it was on the CME555 board.

Since the proposed decoding algorithm is the same as that of CME555 board, the code in Appendix B is implemented in the ATF22LV10CZ chip. The implementation was done using an EPLD programmer, the EMP-11 from Needhams Electronics Inc. EMP-11 universal device programmer supports devices that include EPROM's, EEPROM's, FLASH, 8051 microcontrollers, Microchip PIC, PLDs and other specialty devices [20]. The main advantage of the EMP-11 is its compatibility with Windows operating system, which makes it user-friendly compared to other programmers. Also, the programmer communicates with the PC through a standard parallel port cable [20]. The programmer also comes with EMPWin software that is specifically for programming different devices from a PC. Within the EMPWin software, there is a main menu where the basic functions of a programmer can be accessed. The programmer includes functions such as erase, blank check, program, verify and read. To initiate connection to the device of interest, simply select the device in the device selection menu and all the functions will become accessible. The code in PLD format was uploaded to the ATF22LV10CZ chip successfully.

One of the concerns during the PCB design was whether there is a need for the use of a contrast circuit for the LCD. The CME555 board has a potentiometer for adjusting voltage levels supplied to the LCD for contrast adjustment. The DMC20261NY-LY-AXE manual recommends a potentiometer for the contrast adjust due to the tolerance of the driving voltage and its temperature dependence [5]. An example power supply circuit is shown in Figure 23 [5]. The PB555 prototype power supply is designed according this example circuit and the electrical specifications indicated in Table 4 [5]. The power supply for logic  $V_{DD}$  can be generated by the +5V supply from



**Figure 23 - Example power supply circuit for LCD module**

**Table 4 - Absolute maximum ratings for DMC20261NY-LY-AXE power supply**

Parameter	Symbol	Minimum	Maximum
Supply Voltage (Logic)	$V_{DD}-V_{SS}$	-0.3V	7.0V
Supply Voltage (LCD Drive)	$V_{DD}-V_{EE}$	0V	13.0V
Input Voltage	$V_t$	-0.3V	$V_{DD}+0.3V$

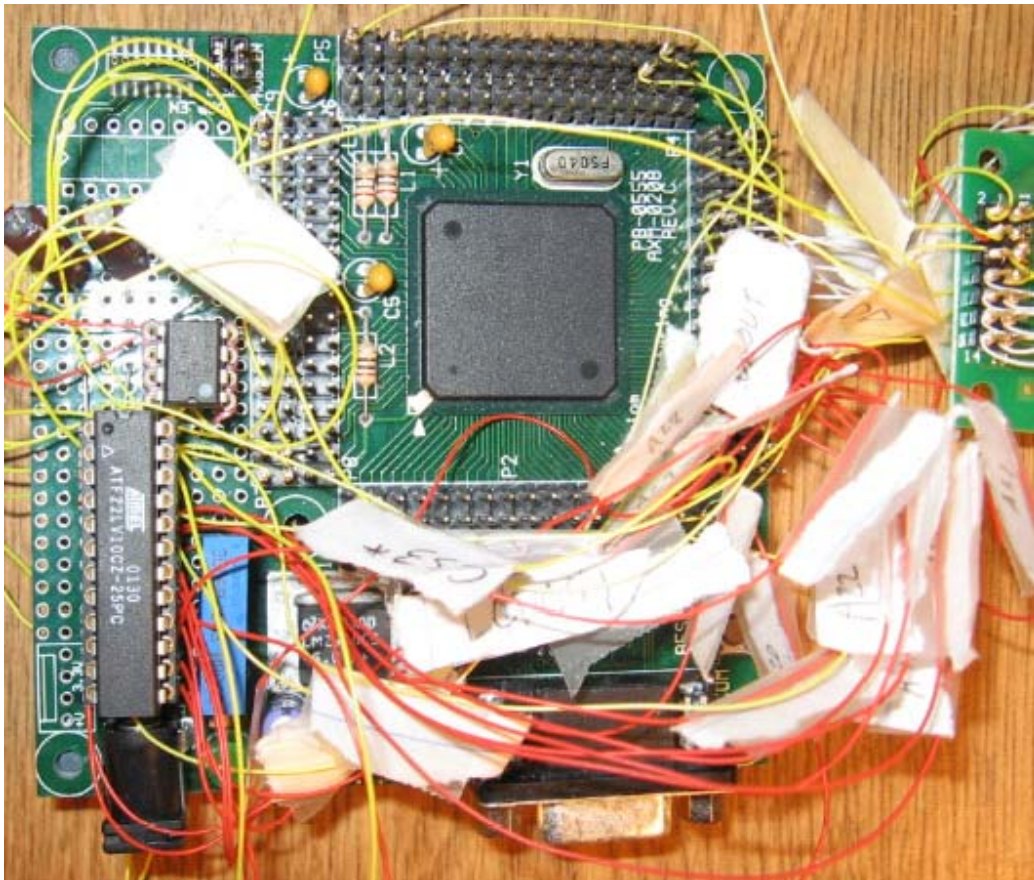
MPC555 but the power supply for LCD drive  $V_{EE}$  needs to be negative as a bias source [5]. This bias voltage cannot be generated straight from MPC555 and extra circuitry is needed to generate this voltage. Two different approaches are available to resolve this problem and each one has its own trade offs in terms of implementation or efficiency.

One approach would be switch to another LCD that has a voltage converter built inside to generate the bias voltage and the other approach is to get a separate voltage converter chip. By switching to another LCD, there wouldn't be the need to have extra circuitry for power supply. But the LCD dimensions are constrained so that it can be able to fit inside the FDR case.

To save time and speed up the process for building the prototype, it was decided to get a separate voltage converter chip for generating the bias voltage. The voltage converter implemented in the prototype is ICL7660 from Intersil. This is a CMOS voltage converter that performs supply voltage conversions from positive to negative for an input range of +1.5V to +10.0V resulting in complementary output voltages of -1.5V to -10.0V [23]. The operation of the voltage converter requires that two external capacitors used, one for a charge pump function and one for a charge reservoir function [23] (See Figure 22 for schematic). With the ICL7660, a +5V power supply from

MPC555 can be translated to  $-5V$  for bias voltage of LCD and the power supply circuit for LCD is complete.

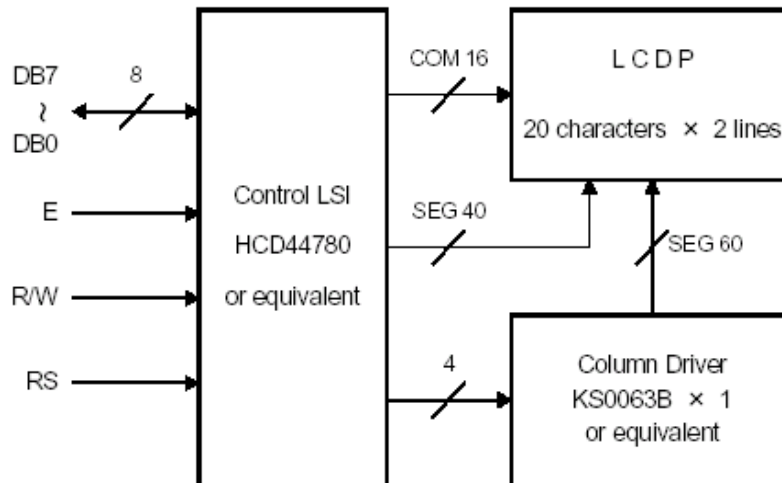
As it was discussed earlier, the prototype is implemented with wire wrapping. The idea is to get wire wrap sockets for the EPLD and the voltage converter and wire wrap the sockets to the PB555 board. The wire wrap socket is just like the regular DIP socket with special wire wrapping pins for connection. Once the sockets are wire wrapped to the PB555 board, the EPLD and voltage converter chips can be plugged into the sockets for testing. During the development phase, the sockets allows for easy removal of the chip for debugging or replacement. Likewise, the LCD DMC20261NY-LY-AXE has header connectors on the bottom, which allows for wire wrapping to be implemented. Once the LCD interfacing has been established physically, it is ready for a software driver to test it.



**Figure 24 - Wire wrapped PB555 prototype for LCD interface**

## 6.5 LCD Test

The DMC20261NY-LY-AXE display uses Hitachi HCD44780 controller. See Figure 25 [5] for a functional diagram of the controller. HCD44780 has two 8 bit registers, an instruction register (IR) and a data register (DR) [35]. The instruction register stores instruction codes such as display clear and cursor shift, and address



**Figure 25 - Functional diagram of DMC20261NY-LY-AXE**

information for display data RAM (DD RAM) and character generator RAM (CG RAM) [8]. DD RAM stores display data represented in 8-bit character codes, its capacity is 80x8 bits, or 80 characters [8]. With displays using fewer than 80 characters, DD RAM can be used as a general data RAM. CG RAM is RAM with which the user can redefine character patterns in software [8]. The instruction register can be written from the processor but cannot read by the processor. The data register temporarily stores data to be written into the DD RAM or the CG RAM and data to be read out from the DD RAM or the CG RAM [8]. When address information is written into the instruction register, data is read into the data register from the DD RAM or the CG RAM by internal operation. Data transfer to the processor is then completed by reading data register. After the processor reads the data register, data in the DD RAM or CG RAM at the next address is sent to the data register for next read from the processor [8]. Register select (RS) signal indicates register selection between data and address. Table 5 [35] shows the operation of HCD44780 using register select, read/write and Enable line.



**Table 5 - HCD44780 controller register selection**

RS	R/W	Enable	Operation
0	0	High, then High to Low	Write to Instruction Register
0	1	High	Read busy flag indicated by DB7
1	0	High, then High to Low	Write data from processor to HD44780
1	1	High	Read data from HD44780 to processor

Writing the LCD driver software requires referencing to the instruction set that is available for the controller. See Table 6 [34] for instruction set and its respective operation code. To make sure the LCD displays characters correctly, there are several initialization steps it must go through after power-on (see Appendix B). The initialization must consist of at least a Function Set command, followed by Display Control, Clear Display and Entry Mode Set [36]. Issuing each of these commands ensures that the programmer knows

**Table 6 - Operation code for HCD44780 instructions**

Instruction	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Clear Display	0	0	0	0	0	0	0	0	0	1
Return Home	0	0	0	0	0	0	0	0	1	*
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S
Display ON/OFF	0	0	0	0	0	0	1	DOF	C	B
Cursor and Display Shift	0	0	0	0	0	1	S/C	R/L	*	*
Function set	0	0	0	0	1	DL	N	F	*	*
Set CG RAM address	0	0	0	1	CG RAM address					
Set DD RAM address	0	0	1	DD RAM address						
Read bus flag and address	0	1	BF	CG RAM/DD RAM address						
Write data to CG or DD RAM	1	0	Write data							
Read data from CG or DD RAM	1	1	Read data							

\* - Don't cares (either 1 or 0)  
B - Blink

C - Cursor display  
DOF - Display On/Off  
I/D – Increment/decrement DD RAM address  
BF - Busy Flag  
DL - Data Length  
N - Number of display lines  
F - Character font  
S/C and R/L - Cursor and display shift

the state of the display. After the initialization is finished, the HCD44780 is ready for read and write operation. The pseudo-code for read/write to LCD is shown below:

#### Reading from a register

1. Set Register Select line high or low to indicate which register to access
2. Set Read/Write line high to indicate a read
3. Set Data Port for the processor to input
4. Set Enable line high to begin read cycle
5. Set delay to allow LCD to fetch data
6. Read data from Data Port of the processor
7. Set Enable line low to finish read cycle

#### Writing to a register

1. Set Register Select line high or low to designate the register to access
2. Set Read/Write line low to indicate a write
3. Set Data Port for the processor to output
4. Write data to Data Port of processor
5. Set Enable line high to begin write cycle
6. Set delay to allow LCD to accept the data
7. Set Enable line low to finish read cycle

The actual LCD driver for testing is listed in Appendix B. This driver program only displays the initialization messages for the FDR. Since FDR firmware development was all done in Metrowerks Codewarrior and implemented using C++, the LCD test driver was also implemented in the same programming environment. However, ever since the beginning of FDR development, there have been numerous upgrades for

Metrowerks Codewarrior. In order to keep up with the latest programming environment, the LCD test driver was implemented in the newest version of Codewarrior Development Studio for MPC5xx - version 8.1. All the code development and debugging were done in Codewarrior Integrated Development Environment (IDE).

The approach to testing of the PB555 prototype begins by implementing the LCD test driver on the CME555 board. Once it is verified that the test driver is able to drive the LCD on CME555, the test driver can then be transferred to the PB555 prototype with some modifications in memory allocation and initialization settings. In Codewarrior IDE, memory allocation can be setup within EPPC Linker and linker command file [18]. The initialization settings for the board involves several files such as debug initialization and memory configuration files [19]. When it was verified that the code in Appendix B is able to drive the LCD successfully on the CME555 board, the same code can be used on PB555 prototype to test LCD interface.

Early research efforts were focused on the study of PB555 memory mapping. Since the software driver was verified to be bug-free, the rest of the work would be placing the code within the appropriate memory space in the PB555 board. However, Metrowerks Codewarrior does not support PB555 board, the board initialization and configuration files will have to be derived from that of CME555. See Appendix A for initialization and configuration files for the PB555 board modified from that of CME555. Generally, the goal was to move the code from CME555 external memory to the internal memory of MPC555. Table 6 shows the modifications that were implemented in Codewarrior settings, see Appendix B for details on coding.

**Table 7 - Memory initialization modifications**

	CME555 (beginning address)	PB555 (beginning address)
Flash source	0x00c10000	0x00010100
RAM	0x00c02000	0x003fa000
ROM	0x00000000	0x00010100
Stack	0x00c7c000	0x003fff0
Code (Debug target)	0x00c10000	0x003fa000
Code (ROM target)	0x00000000	0x00010100
Code (Flash target)	0x00000000	0x00010100
RAM buffer (Flash)	0x00000000	0x00010100
ROM image (Flash)	0x00000000	0x00010100

From the configurations listed in Table 7, the project was compiled using debug target within Codewarrior. The following shows part of the resulting mapping file that was generated from the debug target:

Memory map:

	Starting address	Size	File Offset
.compress.init	003fa000	00000000	00000180
.init	003fa000	00000378	00000180
.compress.text	003fa378	00000000	000004f8
.text	003fa378	000004bc	000004f8
.compress.fini	003fa834	00000000	000009b4
.fini	003fa834	00000000	000009b4
.rodata	003fa834	00000000	000009b8
.ctors	003fa834	00000008	000009b4
.dtors	003fa83c	00000008	000009bc
.extab	003fa844	00000040	000009c4
.extabindex	003fa884	00000080	00000a04
.BINARY	003fa904	00000000	00000a84
.data	003fa908	0000002c	00000a88
.got	003fa934	00000000	00000ab4
.sdata	003fa938	00000004	00000ab8
.sbss	003fa940	00000010	00000ac0
.sdata2	003fa950	00000000	00000ac0
.sbss2	003fa950	00000000	00000ac0
.PPC.EMB.sdata0	fff8000	00000000	00000ac0
.PPC.EMB.sbss0	fff8000	00000000	00000ac0
.bss	003fa950	00000110	00000ac0
.debug_abbrev	0000012b	00000ac0	
.debug_info	000014a5	00000beb	
.debug_line	00001f71	00002090	
.debug_loc	0000068c	00004001	
.debug_pubnames	0000034a	0000468d	

Linker generated symbols:

_f_compress_init	003fa000
_f_compress_init_rom	003fa000
_e_compress_init	003fa000
_f_init	003fa000
_f_init_rom	003fa000
_e_init	003fa378
_f_compress_text	003fa378
_f_compress_text_rom	003fa378
_e_compress_text	003fa378
_f_text	003fa378

_f_text_rom	003fa378
_e_text	003fa834
_f_compress_fini	003fa834
_f_compress_fini_rom	003fa834
_e_compress_fini	003fa834
_f_fini	003fa834
_f_fini_rom	003fa834
_e_fini	003fa834
_f_rodata	003fa834
_f_rodata_rom	003fa838
_e_rodata	003fa834
_f_ctors	003fa834
_f_ctors_rom	003fa834
_e_ctors	003fa83c
_f_dtors	003fa83c
_f_dtors_rom	003fa83c
_e_dtors	003fa844
_fextab	003fa844
_fextab_rom	003fa844
_eextab	003fa884
_fextabindex	003fa884
_fextabindex_rom	003fa884
_eextabindex	003fa904
_f_BINARY	003fa904
_f_BINARY_rom	003fa904
_e_BINARY	003fa904
_f_data	003fa908
_f_data_rom	003fa908
_e_data	003fa934
_f_got	003fa934
_f_got_rom	003fa934
_e_got	003fa934
_f_sdata	003fa938
_f_sdata_rom	003fa938
_e_sdata	003fa93c
_f_sbss	003fa940
_e_sbss	003fa950
_f_sdata2	003fa950
_f_sdata2_rom	003fa950
_e_sdata2	003fa950
_f_sbss2	003fa950
_e_sbss2	003fa950
_f_PPC_EMB_sdata0	ffff8000
_f_PPC_EMB_sdata0_rom	ffff8000
_e_PPC_EMB_sdata0	ffff8000
_f_PPC_EMB_sbss0	ffff8000

```

_e_PPC_EMB_sbss0  ffff8000
  _f_bss          003fa950
  _e_bss          003faa60
  _stack_addr     003ffff0
  _stack_end      003fbff0
  _heap_addr      003faff0
  _heap_end       003fbff0
  _nbfunctions    0000001c
SIZEOF_HEADERS    00000180
  _SDA_BASE_      00402938
  _SDA2_BASE_     00402950
  _ABS_SDA_BASE_  00402938
  _ABS_SDA2_BASE_ 00402950

```

The mapping file verifies that the driver program has been uploaded to the MPC555 chip in the correct locations. However, some important observations can be made from the mapping file. The stack memory location indicates that there are 16Kbytes resorted to stack use and 4Kbytes resorted to heap use. The default configuration in Codewarrior allocates 16Kbytes for stack and heap, but this is not possible on the PB555 board since there is only 24 Kbytes of accessible RAM space. Therefore, care should be taken when uploading to RAM space since there is restricted amount of space in comparison to CME555. Future developments in FDR firmware may run out of RAM space on the PB555 board since the ROM and Flash target does require more on-board RAM space to operate. In particular, the ROM and AutoFlash target requires a RAM buffer space allocated for flashing the program to ROM.

AutoFlash target also requires some code that programs the flash memory, which takes up the RAM buffer space [18][19]. If RAM does become insufficient in the future for AutoFlash target, another option would be using the Codewarrior built-in Flash programmer. The built-in Flash programmer does not need the extra RAM space but it does take more time to program the flash memory than AutoFlash target [19]. Depending on the specific need for development, more RAM chips may be needed to interface to the board in the future.

Once the code has been placed in the appropriate RAM memory space, the LCD interface is ready to be tested. Codewarrior debugger indicates that during the initialization procedures for LCD, the instruction register of the LCD keeps returning a Busy Flag. In particular, the processor reads the instruction register and returns the value

'0xFF' repeatedly. As a result, the debugger suspends in checking the Busy Flag. To make sure that it was not the hardware that caused the problem, the same program (see Appendix 2) was tested on the CME555 board using the PB555 memory configuration. Debugger shows that the instruction register returns the same value '0xFF' and suspends during code execution. Clearly, the processor is not reading the correct value from the instruction register. In view of the fact that the only change made was where the program is placed in memory, it is difficult to find the source of the problem by debugger. With the time limit imposed in this research effort, the LCD interface for PB555 prototype is currently still undergoing testing.

To conclude the LCD interface discussion, the root of the problem may lie in one of the following:

- To access the external LCD memory space, the code has to be placed within a certain memory space – CME555 has the code allocated in external Flash for debug target
- Problem with board initialization or configuration. Since Codewarrior cannot provide these overhead files for PB555, there is not enough information given to determine the source of the problem.

## **6.6 GPS Interface**

As it was indicated earlier, the current FDR implements timing synchronization using the M12+ GPS receiver. However, Motorola will discontinue the production of M12+ at the end of 2005 [17]. Therefore new FDRs need to have a different GPS receiver to replace the M12+. At this point, several GPS have been explored and the most promising one being the M12M from Motorola. The M12M is a direct replacement to the current M12+ receiver. In addition the M12M has the same mechanical dimensions, same options, same communication commands and it has improved Time to First Fix by 50 seconds for a Cold Start [17]. At the same time, power consumption on the M12M has reduced to 150 mW whereas the M12+ consumes 195mW. Finally, the M12M will be launched in the December of 2005 [17].

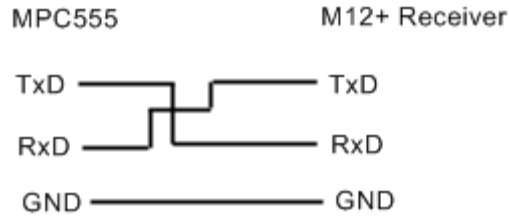
Since the M12M seems to be a good replacement for M12+ and it is backwards compatible with M12+, the design for a new FDR PCB continued with M12+ as the timing synchronizer. M12+ receiver is interfaced to the main board via serial interface. As mentioned earlier, there is a serial port on M12+ for DB-9 cable connection to the CME555 board. The same implementation would not be possible for the PB555 prototype. PB555 has only one serial port available, and it must be used for connection to the TCP/IP device since the software driver requires RTS/CTS handshaking [14], where RTS is the request to send signal and CTS is the clear to send signal.

The GPS interface will be restricted to a connection setup without hardware handshaking. This interface can be implemented by wire wrapping the M12+ to the PB555 using null modem connection without hardware handshaking. Note that this would not be possible if the FDR software implements GPS interface using RTS/CTS handshaking. See Figure 26 for data communication connections. Flow control for this particular interface is done in software only; hardware flow control is not implemented.

However, there is one flaw with transmitting and receiving data via TxD and RxD, the M12+ serial ports are not 5V logic compliant [42]. MPC555 processor SCI transmits data using 5V CMOS logic [3], which may cause damage to GPS serial port even through their logic levels are compatible. The 3V low voltage TTL devices like the M12+ cannot withstand more than approximately 3.3V on their I/O pins. If an I/O pin of a 3V TTL device is driven above its 3.3V limit, the P Channel device in the output driver will conduct, causing current flow from the bus to the 3.3V input through the device. The resulting high current flow can cause destruction of device by latchup effects [15].

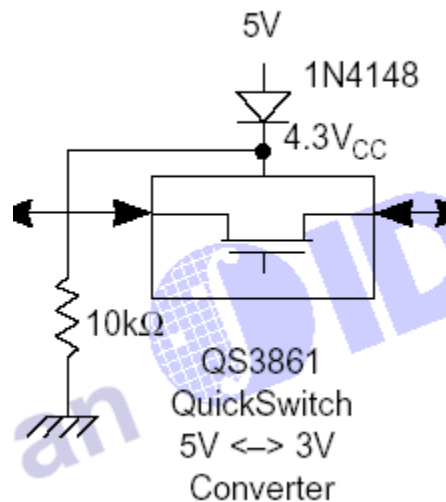
One possible solution to this is implementing another serial port by using a RS-232 driver/receiver IC such as Maxim's MAX3232 [33]. But this solution would overcome the purpose of using the PB555 for prototyping since the proposed solution would require only one serial port for RTS/CTS handshaking. The implementation of another serial port would not be best choice for the PCB design.





**Figure 26 - Null modem connection without RTS/CTS handshaking**

Another solution involves getting a TTL converter chip that drops the 5V TTL to 3V TTL. One TTL converter that is used prominently in the industries is the QuickSwitch QS3861 5V to 3V converter chip. The QS3861 provides a set of ten high speed CMOS TTL compatible bus switches [15]. The 5Ω ON resistance of the QS3861 allows inputs to be connected without adding propagation delay [15]. Figure 27 [15] shows a power supply circuit for the QS3861. By supplying 4.3V to the Vcc pin of QS3861, the driven output will be limited to 3.3V maximum [15] to avoid damaging the GPS serial port. A 4.3V power supply can be created by adding a diode between the 5V supply and the device (see Figure 27). The diode will provide approximately 0.70V drop, thereby creating the 4.3V supply [15]. The 10kΩ resistor is added between the diode's cathode and ground to provide a current path for the diode [15].



**Figure 27 - Power supply circuit for QS3861**

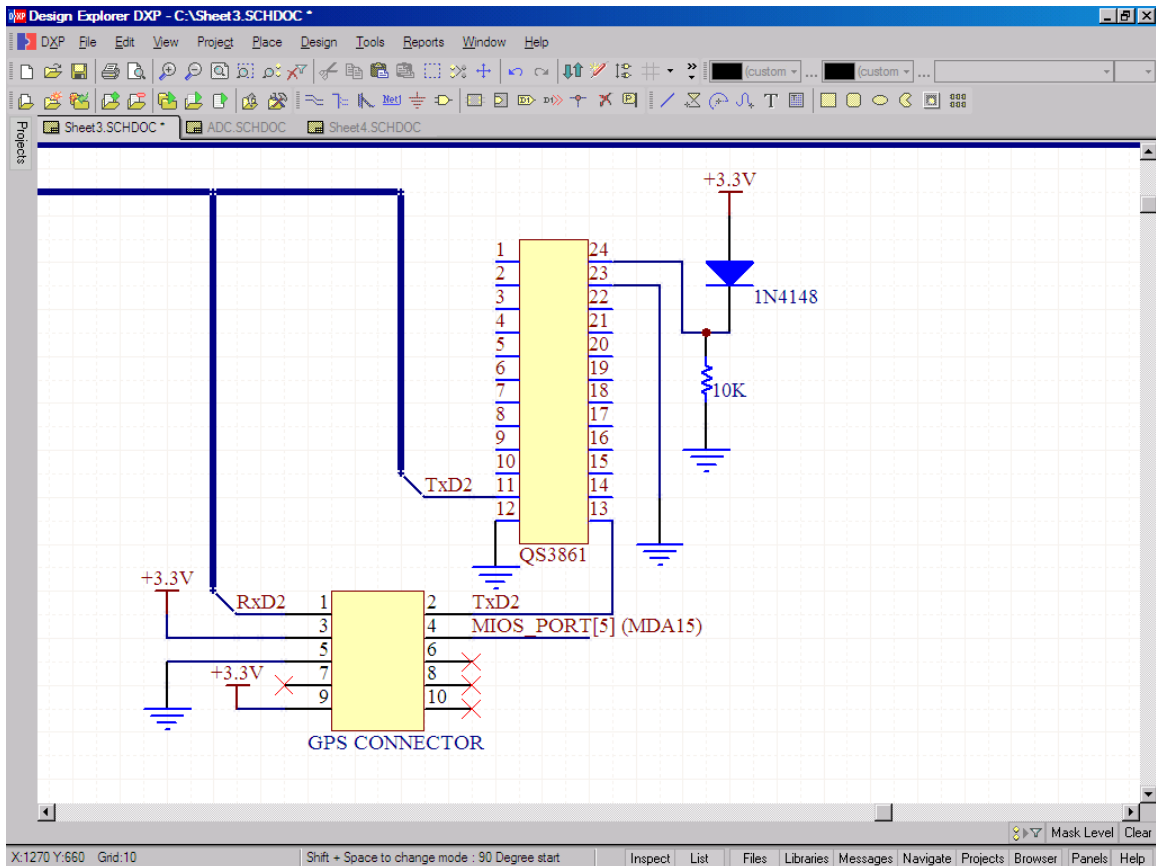
Once the data transmission interface is established for the M12+ to the MPC555, the next step would be to establish connection for the rest of the M12+ signals. There are

10 pins on the M12+ receiver and the best way to make connections to these pins is having a port on the PCB to hold the GPS receiver. However, during the initial testing stages, the GPS can be wire wrapped to the PB555 board.

Before making connections to the GPS pins, a review of the pin functions is necessary. See Table 8 [41] for M12+ pin functions. Pins 1 through 5 are connected to the MPC555 like shown in Figure 28. Pin 6 uses optional backup battery, which is not a mandatory connection and can still have the same effect by applying an external backup voltage source. If without any backup power, the receiver must perform a “Cold Start”, where none of the setup information is available [40]. The Time to First Fix (TTFF) will be longer than if the information had been available [4]. For testing purposes, the optional backup battery will not be implemented in the prototype. However, it can be implemented in the final PCB design by simply adding an external voltage supply [40]. Pin 7 and pin 10 can be ignored since they are reserved and have no functionalities [41]. Pin 8, or the RTCM (Radio Technical Commission Marine) pin takes in SC-104 format for the reception of differential corrections [41]. RTCM input pin allows for Differential GPS (DGPS) correction message [43], which is transmitted through another serial port on the M12+. Since the current FDR does not use any DGPS correction messages, the pin can be left floating. Finally, the antenna bias voltage is applied to pin 9 and this pin can accept any voltage from +2.5 to 5.5 Vdc [44]. So either a +5V or +3.3V supply from the MPC555 can generate the voltage for this pin.

**Table 8 - M12+ connector pin information**

<b>Pin #</b>	<b>Signal Name</b>	<b>Description</b>
1	TxD1	Transmit Data (3V logic)
2	RxD1	Receive Commands (3V logic)
3	+3V PWR	Regulated 3Vdc Input
4	1PPS	1 pulse-per-second output
5	Ground	Signal and Power common
6	Battery	Optional External Backup
7	Reserved	Not currently used
8	RTCM In	RTCM correction input
9	Antenna Bias	3V-5V antenna bias input
10	Reserved	Not currently used



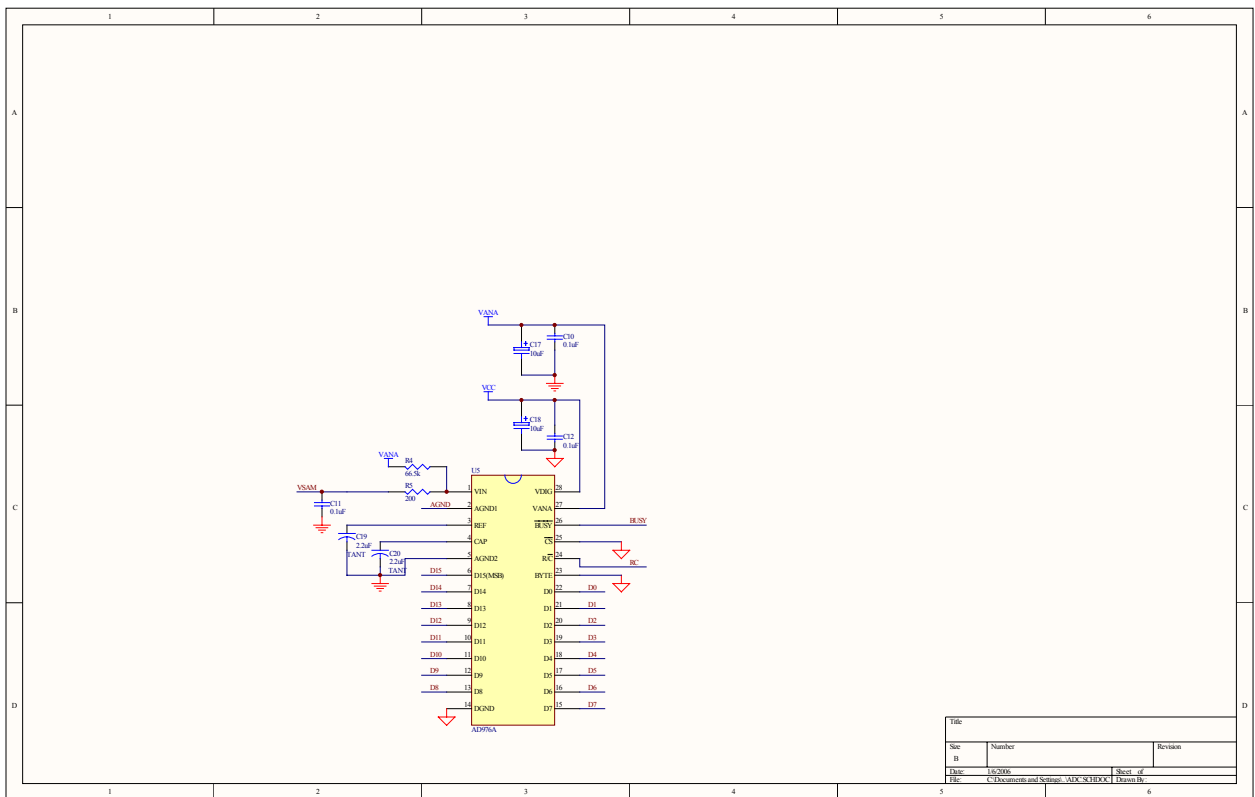
**Figure 28 - Schematic of GPS interface for PB555 board**

As a conclusion, a PB555 prototype for GPS interface can be implemented using wire wrapping. But due to time constraints on this project, a prototype for GPS interface was not implemented. Nevertheless, once the proposed GPS interface is implemented on the PB555 and confirmed to be able to provide the 1PPS and timestamps, the final PCB can use the same circuit and have a 10 pins port to mount the M12+ or the M12M receiver.

### 6.7 Signal Conditioning Unit and ADC Interface

With the GPS and LCD interface established for the PB555 prototype, the next step would be integrating the signal conditioning circuit and ADC into the FDR board. Since the signal conditioning unit is only consist of resistors and capacitors (see Appendix C), it can be easily implemented in the prototype area [2] of the PB555 board for initial testing phase. However, when the final version of the PCB has been established, the circuit needs to be soldered to the board. On the other hand, the AD976A

analog to digital converter has a more complex circuitry. But for initial testing purposes, the ADC circuitry can be simply implemented with wire wrapping since the current ADC unit is on a PCB with the necessary circuits provided. See Appendix C for ADC board circuitry and Figure 29 for AD976A circuitry. Also, the ADC board has header pins on the bottom that allows for wire wrapping. Once again, when the ADC goes through the testing phase, the final FDR board will have a port specifically for mounting of the ADC board.



**Figure 29 - AD976A circuitry**

As a conclusion, the ADC and signal conditioning circuit can be interfaced to the PB555 in the same way as the current CME555 interface(see Appendix C). However, once the wire wrapped prototypes have gone through the testing phase, the final PCB will need to have an ADC port and the signal conditioning circuit soldered on board.

## **CHAPTER 7**

### **CONCLUSIONS AND FUTURE WORK**

#### **7.1 Conclusions**

In this thesis, the need for a new FDR hardware design is discussed. Furthermore, the requirements for the new design are detailed and the approach to a final design has been proposed as well. The thesis goes through the early stages of PCB design starting from scratch and eventually proposes a more practical method of PCB design. The proposed method involves the use of a simple MPC555-based development board, the Axiom PB555 for establishing the required circuitry for MPC555 operation. With the existence of PB555, there is no need to start the PCB design from the bottom level. Instead, the research work can be advanced to the interfacing of external devices. Specifically, the LCD and GPS interface needs to be established with additional circuitry but the ADC and signal conditioning unit can be interfaced to the board using the current design. In addition, the interface between the devices can be tested by wire wrapping a PB555 prototype. However, due to time constraints only the LCD interface has been implemented on the PB555 board and it is still going through testing stages. The idea for establishing the final PCB will be based on how the PB555 prototype is implemented. Once the interface has been verified in implementation, a final board design can be established. Appendix C shows schematics for potential final board design based on the PB555 board and Appendix E has the netlistings for the final board design based on the overall thesis work.

#### **7.2 Future Work**

This thesis has proposed a practical approach to the FDR PCB design. However, due to time constraints some of the interfaces did not get implemented in hardware. In order to confirm to a final PCB design, it is necessary to verify each of the interfaces by implementing a wire wrapped prototype. Currently, there is a prototype built for LCD interface and it is still undergoing testing stages. Once the LCD interface is confirmed, a prototype needs to be implemented for each of the interfacing involved, this include the GPS and the ADC and signal conditioning unit. A final design of the PCB will be determined by the results from the prototype implementations.

## REFERENCES

- [1] Axiom Manufacturing, Garland, Texas. “CME555 Single Board Computer for the Motorola MPC555 Microcontroller Users Manual”, <http://www.axman.com/files/MPC5XX/CME-555/doc/Manuals/555man.PDF>, 555man June 8, 2001, pp. 12-19.
- [2] Axiom Manufacturing, Garland, Texas. “PB-0555 Freescale MPC555 PowerPC Microcontroller Board”, <http://www.axman.com/support/PB-0555/PB-555man.pdf>. PB-555man February 11, 2005.
- [3] Freescale Semiconductor Inc, Austin, Texas. “MPC555/MPC556 User’s Manual”, [http://www.freescale.com/files/microcontrollers/doc/user\\_guide/MPC555UM.pdf](http://www.freescale.com/files/microcontrollers/doc/user_guide/MPC555UM.pdf). MPC555UM October 15, 2000, pp. 2-12 to 2-28.
- [4] Synergy Systems LLC, San Diego, California. “M12+ GPS Receiver User’s Guide”, <http://www.synergy-gps.com/STRMM12+%20User’s%20Guide%20C.pdf> STRMM12+ User’s Guide C 2004, pp. 21
- [5] Optrex Corporation, Tokyo, Japan. “LCD Module Specification”, [http://www.optrex.co.jp/us/product/catalog/pdf\\_dmc/20261cce.pdf](http://www.optrex.co.jp/us/product/catalog/pdf_dmc/20261cce.pdf) 20261cce July 12, 1999.
- [6] Atmel, San Jose, California. “Low-Voltage UV Erasable Programmable Logic Device”, <http://pdf.alldatasheet.com/datasheet-pdf/view/56060/ATMEL/AT22LV10.html> AT22LV10 May, 1998.
- [7] Atmel, San Jose, California. “High-performance EE PLD”, <http://www.ortodoxism.ro/datasheets/atmel/DOC0779.PDF> DOC0779 April, 2001.
- [8] Hitachi, Tokyo, Japan. “HD44780U Dot Matrix Liquid Crystal Display Controller/Driver”, <http://www.electronic-engineering.ch/microchip/datasheets/lcd/hd44780.pdf> hd44780 pp. 11-14
- [9] Zhian Zhong, Chunchun Xu, Bruce J. Billian, Li Zhang, Shu-Jen Steven Tsai, Richard W. Conners, Virgilio A. Centeno, Arun G. Phadke, and Yilu Liu, “ Power System Frequency Monitoring Network”, IEEE Transactions on Power Systems, Vol. 20, pp 1914-1921, 2005.

- [10] Analog Devices Inc, Norwood, Massachusetts. “16 bit, 100kSPS/200kSPS BiCMOS A/D Converters”,  
[http://www.analog.com/UploadedFiles/Data\\_Sheets/61476620AD976\\_A\\_c.pdf](http://www.analog.com/UploadedFiles/Data_Sheets/61476620AD976_A_c.pdf)  
61476620AD976\_A\_c rev. C 1999
- [11] Michael Frank, “Wire Wrapping: The Solderless Connection”,  
<http://www.tessco.com/yts/customerservice/techsupport/whitepapers/pdf/WireWrap.pdf>  
Report of OK Industries.
- [12] Bin Qiu, “Next Generation Information Communication Infrastructure and Case Studies for Future Power Systems”, PhD Dissertation 2002, Dept. of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University, Blacksburg.
- [13] Thomas Walter, “Hardware Checklist for designing with the MPC555”,  
<http://www.nabble.com/MPC555-HARDWARE-DESIGN-CHECKLIST-t531476.html>  
Report of Motorola August 28, 2000.
- [14] FNET team “Frequency Record Unit Generation 2 Description and Operation of FRU for FNET”, Report of Bradley Department of Electrical and Computer Engineering.
- [15] Integrated Device Technology Incorporated, San Jose, California. “High-Speed CMOS 10-bit Bus Switch with Flow-Through Pinout”,  
<http://chipcatalog.com/Datasheet/F8B7C2C896160E42EE694AF61510A907.htm>  
F8B7C2C896160E42EE694AF61510A907 December, 2000.
- [16] Altium, San Diego, California. “Introducing Protel DXP Design Explorer”, 2003, pp.67-72
- [17] Synergy Systems LLC, San Diego, California. “New Motorola Oncore M12M GPS Receiver to Replace M12+ Receiver”, [http://www.synergy-gps.com/End\\_of\\_Life\\_M12\\_2\\_10\\_05.pdf](http://www.synergy-gps.com/End_of_Life_M12_2_10_05.pdf) End\_of\_Life\_M12\_2\_10\_05 February 21, 2005.
- [18] Freescale Semiconductor Inc, Austin, Texas. “Codewarrior Development Studio MPC5xx Edition version 8.1 Targeting Manual”,  
[ftp://ftp.metrowerks.com/pub/docs/targets/MPC5xx\\_8.1/Targeting\\_MPC5xx.pdf](ftp://ftp.metrowerks.com/pub/docs/targets/MPC5xx_8.1/Targeting_MPC5xx.pdf)  
Targeting\_MPC5xx March 17, 2004.

- [19] Freescale Semiconductor Inc, Austin, Texas. “Codewarrior Development Studio IDE 5.5 User’s Guide”,  
[ftp://ftp.metrowerks.com/pub/docs/targets/MPC5xx\\_8.1/IDE\\_5.5\\_Users\\_Guide.pdf](ftp://ftp.metrowerks.com/pub/docs/targets/MPC5xx_8.1/IDE_5.5_Users_Guide.pdf)  
IDE\_5.5\_Users\_Guide June 10, 2003.
- [20] Needham’s Electronics, Sacramento, California. “EMP Series Programmer User’s Manual”, [http://www.needhams.com/downloads/emp\\_win.pdf](http://www.needhams.com/downloads/emp_win.pdf) emp\_win (v4.0).
- [21] Moxa, Brea, California. “Nport DE-311/DE-211”,  
[http://www.moxa.com/product/download/NPort\\_DE-311\\_DE-211.pdf](http://www.moxa.com/product/download/NPort_DE-311_DE-211.pdf) NPort\_DE-311\_DE-211.
- [22] Axiom Manufacturing, Garland, Texas. “AXM-0258 CME-0555”,  
<http://www.axman.com/files/MPC5XX/CME-555/doc/Schematics/CE555bsh3.pdf>  
CE555bsh3 rev. C August 9, 2001.
- [23] Intersil, Milpitas, California. “ICL7660, ICL7660A Data Sheet”,  
<http://www.intersil.com/data/fn/fn3072.pdf> fn3072 April, 1999.
- [24] Freescale Semiconductor Inc, Austin, Texas. “MPC555 Product Brief”,  
[http://www.freescale.com/files/microcontrollers/doc/prod\\_brief/MPC555PB.pdf](http://www.freescale.com/files/microcontrollers/doc/prod_brief/MPC555PB.pdf)  
MPC555PB rev. 3 February 2003.
- [25] Freescale Semiconductor Inc, Austin, Texas. “MPC555/MPC556 User’s Manual”,  
[http://www.freescale.com/files/microcontrollers/doc/user\\_guide/MPC555UM.pdf](http://www.freescale.com/files/microcontrollers/doc/user_guide/MPC555UM.pdf)  
MPC555UM October 15, 2000, pp. 7-1 to 7-4.
- [26] Freescale Semiconductor Inc, Austin, Texas. “MPC555/MPC556 User’s Manual”,  
[http://www.freescale.com/files/microcontrollers/doc/user\\_guide/MPC555UM.pdf](http://www.freescale.com/files/microcontrollers/doc/user_guide/MPC555UM.pdf)  
MPC555UM October 15, 2000, pp. 8-1 to 8-6.
- [27] Freescale Semiconductor Inc, Austin, Texas. “MPC555/MPC556 User’s Manual”,  
[http://www.freescale.com/files/microcontrollers/doc/user\\_guide/MPC555UM.pdf](http://www.freescale.com/files/microcontrollers/doc/user_guide/MPC555UM.pdf)  
MPC555UM October 15, 2000, pp. 8-20 to 8-22.
- [28] Freescale Semiconductor Inc, Austin, Texas. “MPC555/MPC556 User’s Manual”,  
[http://www.freescale.com/files/microcontrollers/doc/user\\_guide/MPC555UM.pdf](http://www.freescale.com/files/microcontrollers/doc/user_guide/MPC555UM.pdf)  
MPC555UM October 15, 2000, pp. 9-4 to 9-7.



- [29] Freescale Semiconductor Inc, Austin, Texas. “MPC555/MPC556 User’s Manual”, [http://www.freescale.com/files/microcontrollers/doc/user\\_guide/MPC555UM.pdf](http://www.freescale.com/files/microcontrollers/doc/user_guide/MPC555UM.pdf). MPC555UM October 15, 2000, pp. 7-7 to 7-8.
- [30] Freescale Semiconductor Inc, Austin, Texas. “MPC555/MPC556 User’s Manual”, [http://www.freescale.com/files/microcontrollers/doc/user\\_guide/MPC555UM.pdf](http://www.freescale.com/files/microcontrollers/doc/user_guide/MPC555UM.pdf). MPC555UM October 15, 2000, pp. 8-12.
- [31] Freescale Semiconductor Inc, Austin, Texas. “MPC555/MPC556 User’s Manual”, [http://www.freescale.com/files/microcontrollers/doc/user\\_guide/MPC555UM.pdf](http://www.freescale.com/files/microcontrollers/doc/user_guide/MPC555UM.pdf). MPC555UM October 15, 2000, pp. 22-1 to 22-3.
- [32] Freescale Semiconductor Inc, Austin, Texas. “MPC555/MPC556 User’s Manual”, [http://www.freescale.com/files/microcontrollers/doc/user\\_guide/MPC555UM.pdf](http://www.freescale.com/files/microcontrollers/doc/user_guide/MPC555UM.pdf). MPC555UM October 15, 2000, pp. 19-1 to 19-2.
- [33] Maxim Integrated Products, Sunnyvale, California. “3.0V to 5.5V, Low-Power, up to 1Mbps, True RS-232 Transceivers Using Four 0.1uF External Capacitors”, <http://pdfserv.maxim-ic.com/en/ds/MAX3222-MAX3241.pdf>. MAX3222-MAX3241 October, 2003.
- [34] Hitachi, Tokyo, Japan. “HD44780U Dot Matrix Liquid Crystal Display Controller/Driver”, <http://www.electronic-engineering.ch/microchip/datasheets/lcd/hd44780.pdf> hd44780, pp. 25.
- [35] Hitachi, Tokyo, Japan. “HD44780U Dot Matrix Liquid Crystal Display Controller/Driver”, <http://www.electronic-engineering.ch/microchip/datasheets/lcd/hd44780.pdf> hd44780, pp. 10.
- [36] Hitachi, Tokyo, Japan. “HD44780U Dot Matrix Liquid Crystal Display Controller/Driver”, <http://www.electronic-engineering.ch/microchip/datasheets/lcd/hd44780.pdf> hd44780, pp. 44.
- [37] Hitachi, Tokyo, Japan. “HD44780U Dot Matrix Liquid Crystal Display Controller/Driver”, <http://www.electronic-engineering.ch/microchip/datasheets/lcd/hd44780.pdf> hd44780, pp. 46.
- [38] Synergy Systems LLC, San Diego, California. “M12+ GPS Receiver User’s Guide”, <http://www.synergy-gps.com/STRMM12+%20User's%20Guide%20C.pdf> STRMM12+ User’s Guide C 2004, pp. 17

- [39] Synergy Systems LLC, San Diego, California. “M12+ GPS Receiver User’s Guide”, <http://www.synergy-gps.com/STRMM12+%20User’s%20Guide%20C.pdf> STRMM12+ User’s Guide C 2004, pp.12
- [40] Synergy Systems LLC, San Diego, California. “M12+ GPS Receiver User’s Guide”, <http://www.synergy-gps.com/STRMM12+%20User’s%20Guide%20C.pdf> STRMM12+ User’s Guide C 2004, pp.13 – pp.14
- [41] Synergy Systems LLC, San Diego, California. “M12+ GPS Receiver User’s Guide”, <http://www.synergy-gps.com/STRMM12+%20User’s%20Guide%20C.pdf> STRMM12+ User’s Guide C 2004, pp. 18
- [42] Synergy Systems LLC, San Diego, California. “M12+ GPS Receiver User’s Guide”, <http://www.synergy-gps.com/STRMM12+%20User’s%20Guide%20C.pdf> STRMM12+ User’s Guide C 2004, pp. 20
- [43] Synergy Systems LLC, San Diego, California. “M12+ GPS Receiver User’s Guide”, <http://www.synergy-gps.com/STRMM12+%20User’s%20Guide%20C.pdf> STRMM12+ User’s Guide C 2004, pp.38
- [44] Synergy Systems LLC, San Diego, California. “M12+ GPS Receiver User’s Guide”, <http://www.synergy-gps.com/STRMM12+%20User’s%20Guide%20C.pdf> STRMM12+ User’s Guide C 2004, pp.16

## APPENDIX A

### Initialization and configuration code for PB555

#### 1. Linker Command File for CME555 (modified for PB555)

```

/* original declaration
_flash_source = 0x00c10000;           // **NOTE: MUST match RAM buffer
                                       // setting in linker preference panel
*/
_flash_source = 0x00010100;           //modification for PB made by Lei
MEMORY {
/*original declaration
    ram : org = 0x00c02000
    rom : org = 0x00000000 // desired ROM address (boot address for 555)
*/
    ram : org = 0x003FA000             //modification for PB made by Lei
    rom : org = 0x00010100
}
/* FORCEFILES are used so that the linker will not deadstrip the file reset.s. The
function reset would be deadstripped since it is not ever called by anything */
FORCEACTIVE { gInterruptVectorTable, __reset }
SECTIONS {
    .reset : {} > rom
    .init : {} > rom
    GROUP : {
        .text (TEXT) ALIGN(0x1000) : {}
        .rodata (CONST) : {
            *(.rdata)
            *(.rodata)
        }
        .ctors : {}
        .dtors : {}
        extab : {}
        extabindex : {}
    } > rom           // for ROM images, this can be 'rom' if you want to execute in
ROM
                       // or 'code' if you want to execute in RAM
    GROUP : {
        .data : {}
        .sdata : {}
        .sbss : {}
        .sdata2 : {}
        .sbss2 : {}
        .bss : {}
        .PPC.EMB.sdata0 : {}
        .PPC.EMB.sbss0 : {}
    } > ram

```

```

// The dummy section is just a placeholder. The linker automatically
// generates an address for it in the ROM image, which tells us
// where the end of the ROM image is.
.dummy ALIGN(64): {}
_flash_dest = _f_reset; // true flash address starts w/.init
section
_flash_size = _f_dummy_rom - _flash_dest;
// The .fcopy section contains a small piece of code that copies the
// ROM image to flash. We don't copy the .fcopy section itself to flash
// because it could erase the flash if it were accidentally executed
// at a later time.
//
// Bind it to the address it will occupy in the RAM buffer so we can
// execute it directly from the RAM buffer.
.fcopy BIND(_flash_source + _flash_size) ALIGN(64) : {
    *(.fcopy)
}
.fcopy_data : {}
// The internal flash algorithms provided by Motorola are
// packaged in a binary file. The linker includes the contents
// in the .BINARY section.
.BINARY : {}
}

```

## 2. Runtime initialization file (single chip operation)

```

/*
    56X_Chip_init.c

    This is the chip initialization file for the MPC56X series
*/
//file used for single chip operation – added by Lei

#pragma section code_type ".init"

#define SIUMCR                0x2fc000
#define SYPCR                 0x2fc004
#define SIPEND                0x2fc010
#define SIMASK                0x2fc014
#define SIEL                  0x2fc018
#define SIVEC                 0x2fc01c
#define PLPRCR                0x2fc284

//SPR defines
#define SPR_SRR1              27

```

```

#define SPR_ICTRL                158
#define SPR_IMMR                 638
#define SPR_BBCMCR              560

#ifdef __cplusplus
extern "C" {
#endif

asm void __reset(void);
asm void usr_init();
extern void __start();

#ifdef __cplusplus
}
#endif

asm void __reset(void)
{
    //
    // Enable machine check exceptions
    //
        lis        r3, 0
        ori        r3, r3, 0x1002
        mtmsr r3

    //
    // ROM is not relocated, so nothing left to do here.
    //

        b         __start
}

asm void usr_init()
{
        nofralloc

// Adr 0x002fc384 = 0x55ccaa33 PLPRCRK: open key
        lis        r5, 0x55cc
        ori        r5, r5, 0xaa33
        lis        r4, 0x002fc384@ha
        stw        r5, 0x002fc384@l(r4)

// Adr 0x002fc284 = 0x00900000 PLPRCR
        lis        r5, 0x0090
        ori        r5, r5, 0x0000
        lis        r4, 0x002fc284@ha

```

```

        stw                r5, 0x002fc284@l(r4)

// SPR MSR = 0x00003002 MSR
        lis                r5, 0x0000
        ori                r5, r5, 0x3002
        mtmsr             r5

// SPR 27 = 0x00003002 SSR1
        lis                r5, 0x0000
        ori                r5, r5, 0x3002
        mtspr             27, r5

// SPR 638 = 0xffff0080 IMMR
        lis                r5, 0xffff
        ori                r5, r5, 0x0080
        mtspr             638, r5

// Adr 0x002fc000 = 0x00000000 SIUMCR
        lis                r5, 0x0000
        ori                r5, r5, 0x0000
        lis                r4, 0x002fc000@ha
        stw                r5, 0x002fc000@l(r4)

// Adr 0x002fc004 = 0x0000ff88 SYPCR
        lis                r5, 0x0000
        ori                r5, r5, 0xff88
        lis                r4, 0x002fc004@ha
        stw                r5, 0x002fc004@l(r4)

// SPR 158 = (SPR 158 & 0xfffffff0) | 0x00000007 ICTRL: switch serialised mode off
        lis                r5, 0xffff
        ori                r5, r5, 0xffff0
        mfspr             r4, 158
        and                r4, r4, r5
        lis                r5, 0x0000
        ori                r5, r5, 0x0007
        or                 r4, r4, r5
        mtspr             158, r4

// Adr 0x002fc380 = 0x55ccaa33 SCCRK: open key
        lis                r5, 0x55cc
        ori                r5, r5, 0xaa33
        lis                r4, 0x002fc380@ha
        stw                r5, 0x002fc380@l(r4)

// Adr 0x002fc280 = 0x00010000 SCCR

```

```

lis          r5, 0x0001
ori          r5, r5, 0x0000
lis          r4, 0x002fc280@ha
stw          r5, 0x002fc280@l(r4)

// Adr 0x00305014 = 0x0000 PORTQS
li          r5, 0x0000
lis          r4, 0x00305014@ha
sth          r5, 0x00305014@l(r4)

// Adr 0x00305016 = 0x0000 PQSPAR/DDRQS
li          r5, 0x0000
lis          r4, 0x00305016@ha
sth          r5, 0x00305016@l(r4)

// Adr 0x00307f80 = 0x00000000 UMCr
lis          r5, 0x0000
ori          r5, r5, 0x0000
lis          r4, 0x00307f80@ha
stw          r5, 0x00307f80@l(r4)

```

### 3. Memory configuration file (single chip operation)

```

// 56x_chip.mem - memory config file for MPC56x
// The CodeWarrior debugger uses this file to allow the proper display of memory.
// This is for internal memory only. If using external memory,
// add the appropriate range command.
reservedchar 0xBA
// Flash area (UC3F flash array)
range 0x00000000 0x000FFFFFF 4 Read
reserved 0x00100000 0x002F7FFF
// DECRAM SRAM
range 0x002F8000 0x002F8FFF 4 ReadWrite
reserved 0x002F9000 0x002FA003
// BBC
range 0x002FA004 0x002FA00F 4 ReadWrite
range 0x002FA010 0x002FA013 2 ReadWrite
range 0x002FA014 0x002FA03F 4 ReadWrite
reserved 0x002FA040 0x002FBFFF
// USIU
range 0x002FC000 0x002FC007 4 ReadWrite
reserved 0x002FC008 0x002FC00D
range 0x002FC00E 0x002FC00F 2 Write
range 0x002FC010 0x002FC01B 4 ReadWrite
range 0x002FC01C 0x002FC01F 4 Read
range 0x002FC020 0x002FC033 4 ReadWrite

```

```

reserved 0x002FC034 0x002FC03B
range 0x002FC038 0x002FC057 4 ReadWrite
reserved 0x002FC058 0x002FC0FF
range 0x002FC100 0x002FC11F 4 ReadWrite
reserved 0x002FC120 0x002FC13F
range 0x002FC140 0x002FC147 4 ReadWrite
reserved 0x002FC148 0x002FC177
range 0x002FC178 0x002FC179 2 ReadWrite
reserved 0x002FC17A 0x002FC1FF
range 0x002FC200 0x002FC201 2 ReadWrite
reserved 0x002FC202 0x002FC203
range 0x002FC204 0x002FC20B 4 ReadWrite
reserved 0x002FC20C 0x002FC21F
range 0x002FC220 0x002FC221 2 ReadWrite
reserved 0x002FC222 0x002FC223
range 0x002FC224 0x002FC22F 4 ReadWrite
reserved 0x002FC230 0x002FC23F
range 0x002FC240 0x002FC241 2 ReadWrite
reserved 0x002FC242 0x002FC243
range 0x002FC244 0x002FC247 4 ReadWrite
range 0x002FC248 0x002FC24B 4 Read
reserved 0x002FC24C 0x002FC27F
range 0x002FC280 0x002FC287 4 ReadWrite
range 0x002FC288 0x002FC289 2 ReadWrite
reserved 0x002FC28A 0x002FC28B
range 0x002FC28C 0x002FC28D 2 ReadWrite
reserved 0x002FC28E 0x002FC28F
range 0x002FC290 0x002FC291 2 ReadWrite
reserved 0x002FC292 0x002FC2FF
range 0x002FC300 0x002FC30F 4 ReadWrite
reserved 0x002FC310 0x002FC31F
range 0x002FC320 0x002FC32F 4 ReadWrite
reserved 0x002FC330 0x002FC33F
range 0x002FC340 0x002FC347 4 ReadWrite
reserved 0x002FC348 0x002FC37F
range 0x002FC380 0x002FC38B 4 ReadWrite
reserved 0x002FC38C 0x002FC7FF
// Flash Control Registers (UC3F_A)
range 0x002FC800 0x002FC80B 4 ReadWrite
reserved 0x002FC80C 0x002FC83F
range 0x002FC840 0x002FC84B 4 ReadWrite
reserved 0x002FC84C 0x002FFFFFF
// DPT RAM Control Registers
range 0x00300000 0x0030000B 2 ReadWrite // AB
reserved 0x0030000C 0x0030003F
range 0x00300040 0x0030004B 2 ReadWrite // C

```



```

reserved 0x0030004C 0x0030007F
// DLCM
range 0x00300080 0x00300081 2 ReadWrite
reserved 0x00300082 0x00300083
range 0x00300080 0x0030008F 2 ReadWrite
reserved 0x00300090 0x003000FF
// DPT RAM Array
// NOTE: Access to DPT RAM Array is disabled once TPUMCR[EMU]=1
range 0x00301000 0x00303FFF 2 ReadWrite
// TPU3
// NOTE: Most TPU3 registers also allow 32 bit access
range 0x00304000 0x0030402F 2 ReadWrite
reserved 0x00304030 0x003040FF
range 0x00304100 0x003041FF 2 ReadWrite
reserved 0x00304200 0x003043FF
range 0x00304400 0x0030442F 2 ReadWrite
reserved 0x00304430 0x003044FF
range 0x00304500 0x003045FF 2 ReadWrite
reserved 0x00304600 0x003047FF

// QADC64
range 0x00304800 0x00304813 2 ReadWrite
reserved 0x00304813 0x003049FF
range 0x00304A00 0x00304C13 2 ReadWrite
reserved 0x00304C14 0x00304DFF
range 0x00304E00 0x0030500F 2 ReadWrite
reserved 0x00305010 0x00305013
range 0x00305014 0x0030501D 2 ReadWrite
range 0x0030501E 0x0030501F 1 ReadWrite
range 0x00305020 0x0030506B 2 ReadWrite
reserved 0x0030506C 0x0030513F
range 0x00305140 0x003051DF 2 ReadWrite
reserved 0x003051F0 0x003053FF
// QSMCM_B
range 0x00305400 0x0030540F 2 ReadWrite
reserved 0x00305410 0x00305413
range 0x00305414 0x0030541D 2 ReadWrite
range 0x0030541E 0x0030541F 1 ReadWrite
range 0x00305420 0x0030546B 2 ReadWrite
reserved 0x0030546C 0x0030553F
range 0x00305540 0x003055DF 2 ReadWrite
reserved 0x003055F0 0x00305BFF
// TPU3
range 0x00305C00 0x00305C2F 2 ReadWrite
reserved 0x00305C30 0x00305CFF
range 0x00305D00 0x00305DFF 2 ReadWrite

```

```
reserved 0x00305E00 0x00305FFF
// MIOS14
range 0x00306000 0x00306047 2 ReadWrite
reserved 0x00306048 0x0030604F
range 0x00306050 0x003060C7 2 ReadWrite
reserved 0x003060C8 0x003060D7
range 0x003060D8 0x00306103 2 ReadWrite
reserved 0x00306104 0x003067FF
range 0x00306800 0x00306807 2 ReadWrite
reserved 0x00306808 0x00306815
range 0x00306816 0x00306817 2 ReadWrite
reserved 0x00306818 0x00306BFF
range 0x00306C00 0x00306C01 2 ReadWrite
reserved 0x00306C02 0x00306C03
range 0x00306C04 0x00306C07 2 ReadWrite
reserved 0x00306C08 0x00306C2F
range 0x00306C30 0x00306C31 2 ReadWrite
reserved 0x00306C32 0x00306C3F
range 0x00306C40 0x00306C41 2 ReadWrite
reserved 0x00306C42 0x00306C43
range 0x00306C44 0x00306C47 2 ReadWrite
reserved 0x00306C48 0x00306C6F
range 0x00306C70 0x00306C71 2 ReadWrite
reserved 0x00306C72 0x0030707F
// TOUCAN
range 0x00307080 0x0030708B 2 ReadWrite
reserved 0x0030708C 0x0030708F
range 0x00307090 0x0030709B 2 ReadWrite
reserved 0x0030709C 0x0030709F
range 0x003070A0 0x003070A7 2 ReadWrite
reserved 0x003070A8 0x003070FF
range 0x00307100 0x003071FF 4 ReadWrite
reserved 0x00307200 0x0030747F
range 0x00307480 0x0030748B 2 ReadWrite
reserved 0x0030748C 0x0030748F
range 0x00307490 0x0030749B 2 ReadWrite
reserved 0x0030749C 0x0030749F
range 0x003074A0 0x003074A7 2 ReadWrite
reserved 0x003074A8 0x003074FF
range 0x00307500 0x003075FF 4 ReadWrite
reserved 0x00307600 0x0030787F
// TouCAN_C
range 0x00307880 0x0030788B 2 ReadWrite
reserved 0x0030788C 0x0030788F
range 0x00307890 0x0030789B 2 ReadWrite
reserved 0x0030789C 0x0030789F
```

```

range 0x003078A0 0x003078A7 2 ReadWrite
reserved 0x003078A8 0x003078FF
range 0x00307900 0x003079FF 2 ReadWrite
reserved 0x00307A00 0x00307F7F
// UIMB
range 0x00307F80 0x00307F83 4 ReadWrite
reserved 0x00307F84 0x00307F8F
range 0x00307F90 0x00307F93 4 ReadWrite
reserved 0x00307F94 0x00307F9F
range 0x00307FA0 0x00307FA3 4 ReadWrite
reserved 0x00307FA4 0x00307FFF
// CALRAM
range 0x00380000 0x0038002F 4 ReadWrite
reserved 0x00380030 0x0038003F
range 0x00380040 0x0038006F 4 ReadWrite
reserved 0x00380070 0x003800FF
range 0x003F7000 0x003FFFFFF 4 ReadWrite

```

## APPENDIX B LCD DRIVER CODE

### 1. LCD initialization steps

[Power ON]

[ Wait more than 15ms ]  
[after Vdd rises to 4.5v]

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Can't check BF before this instruction
0	0	0	0	1	1	*	*	*	*	Function set (8-bit interface)

[Wait more than 4.1ms]

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Can't check BF before this instruction
0	0	0	0	1	1	*	*	*	*	Function set (8-bit interface)

[Wait more than 100us]

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Can't check BF before this instruction
0	0	0	0	1	1	*	*	*	*	Function set (8-bit interface)

BF can be checked after the following instructions. When BF is not checked, the waiting time between instructions is longer than the execution time.  
(See Instruction set)

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
----	-----	-----	-----	-----	-----	-----	-----	-----	-----

0	0	0	0	1	1	N	F	*	*	Function set	[8-bit Interface ]
											[Specify display lines]
RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		[and character font ]
0	0	0	0	0	0	1	0	0	0	Display OFF	These cannot be changed afterwards
RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
0	0	0	0	0	0	0	0	0	1	Display clear	
RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
0	0	0	0	0	0	0	1	I/D	S	entry mode set	

## 2. Main program for LCD driver

```

#include <iostream>
#include "FNET_LCD.h"

using namespace std;

void main()
{
    USIU.BR3.R = 0x01000403;
    LCD_InitDriver();
    LCD_WriteControl(OnOffCtrl);
    LCD_WriteControl(DisplayClear);
    LCD_CursorOff();
    LCD_WriteControl(E_M_RightIncrm);
    LCD_Clear();
    delay1ms(100);
    LCD_DisplayString(2,1,"Initializing...");
    LCD_DisplayString(2,16,"done!");

    LCD_Clear();
    LCD_DisplayString(1, 1, " Collecting Data...");
}

// Busy wait loop to generate a delay multiple of 1ms
void delay1ms(unsigned int msec) {
    unsigned int i,j;

    if (msec > 0) {
        for (j=0; j<=msec; j++)
            for (i=0; i<DELAY1MS; i++)
                ;
    }
    return;
}

```

```

void LCD_DisplayCharacter (unsigned char a_char)
{
    delay1ms(10); //jcp
    LCD_WriteData (a_char);
}

/*
** LCD_DisplayString: Display a string at the specified row and column.
*/
// Note: this fn uses D0 as the start address of column 2 because
// using C0 doesn't work

void LCD_DisplayStringMoving (char row, char column, const char *string)
{
    switch (row) {

        case 1:

            LCD_WriteControl (0x80 + column - 1);

            while (*string)
                {
                    delay1ms(3000);
                    LCD_DisplayCharacter (*string++);
                }

            break;

        case 2:

            LCD_WriteControl (0xD0 + column - 1);
            while (*string)
                {
                    delay1ms(3000);
                    LCD_DisplayCharacter (*string++);
                }

            default: break;
    }
}

void LCD_DisplayString (char row, char column, const char *string)
{
    switch (row) {

        case 1:

```

```

        LCD_WriteControl (0x80 + column - 1);
        while (*string)
        LCD_DisplayCharacter (*string++);
        break;

    case 2:
        LCD_WriteControl (0xC0 + column - 1);
        while (*string)
        LCD_DisplayCharacter (*string++);
    }
}

/*
** LCD_WipeOffLR: "Wipe" screen left-to-right.
*/
void LCD_WipeOffLR (void)
{
    // "wipe" off old screen (left to right)
    char i;
    for (i=0; i<20; i++) {
        LCD_WriteControl (0x80 + i - 1);
        LCD_DisplayCharacter(BLOCK);
        LCD_WriteControl (0xC0 + i - 1);
        LCD_DisplayCharacter(BLOCK);
        delay1ms(800);
    }
}

/*
** LCD_DisplayRow: Display a string at the specified row.
*/
void LCD_DisplayRow (int row, unsigned char *string)
{
    char i;
    LCD_Cursor (row, 1);
    for (i=0; i<20; i++)
        LCD_DisplayCharacter (*string++);
}

/*
** LCD_CursorLeft: Move the cursor left by one character.
*/
void LCD_ShiftLeft (void)
{

```

```

    LCD_WriteControl (0x1c);
}

/*
** LCD_CursorRight: Move the cursor right by one character.
*/
void LCD_ShiftRight (void)
{
    LCD_WriteControl (0x18);
}

/*
** LCD_CursorOn: Turn the cursor on.
*/
void LCD_CursorOn (void)
{
    LCD_WriteControl (0x0e);
}

/*
** LCD_CursorOff: Turn the cursor off.
*/
void LCD_CursorOff (void)
{
    LCD_WriteControl (0x0c);
}

/*
** LCD_InitDriver: Initialize the LCD driver.
*/
void LCD_InitDriver (void)
{
    delay1ms(500);           // 10ms delay
    LCD_WriteControl(InitWord1);           // 38h:8 bits,2 lines, 5*7 dots
    delay1ms(50);           // 10ms delay
    LCD_WriteControl(InitWord1);
    delay1ms(50);           // 10ms delay
    LCD_WriteControl(InitWord1);
    delay1ms(50);
    LCD_WriteControl(InitWord1);           // 10ms delay
}

```

```

void LCDWaitForBusy(){
    unsigned char lcdval;
    do{
        lcdval = get8(LCD_CMD);
    }while((lcdval & 0x80) != 0);// wait for busy bit to go low
}

/*
** LCD_WriteControl: Write a control instruction to the LCD
*/
void LCD_WriteControl (unsigned char cmdval)
{
    LCDWaitForBusy();
    put8(LCD_CMD,cmdval);
}

/*
** LCD_WriteData: Write one byte of data to the LCD
*/
void LCD_WriteData (unsigned char datval)
{
    LCDWaitForBusy();
    put8(LCD_DAT,datval);
}

/*
** LCD_Clear: Clear the LCD screen (also homes cursor).
*/
void LCD_Clear (void)
{
    LCD_WriteControl(0x01);
}

/*
** LCD_Home: Position the LCD cursor at row 1, col 1.
*/
void LCD_Home (void)
{
    LCD_WriteControl(0x02);
}

```

### 3. LCD driver header file



```

#ifndef FNET_LCD_H
#define FNET_LCD_H

#include "mpc555.h"

/*-----
CONSTANTS
-----*/

typedef unsigned char UNS8;
typedef unsigned short UNS16;
typedef unsigned long UNS32;

typedef unsigned char BOOL;          /* unsigned 8 bits */
typedef int SINT32;                  /* signed int 32 bits */
typedef unsigned short UINT16;      /* unsigned int 16 bits */
typedef short SINT16;                /* signed int 16 bits */
typedef unsigned char UINT8;        /* unsigned int 8 bits */
typedef char SINT8;                  /* signed int 8 bits */

typedef volatile unsigned char VUNS8;
typedef volatile unsigned short VUNS16;
typedef volatile unsigned long VUNS32;

typedef unsigned char* UNS8P;

#define TRUE 1
#define FALSE 0

#define put8(p, x)    ((*((volatile UNS8*)(p))) = (x))
#define put16(p, x)  ((*((volatile UNS16*)(p))) = (x))
#define put32(p, x)  ((*((volatile UNS32*)(p))) = (x))

#define get8(p)       ((*((volatile UNS8*)(p))))
#define get16(p)      ((*((volatile UNS16*)(p))))
#define get32(p)      ((*((volatile UNS32*)(p))))

// begin lcd stuff

#define _IO_BASE      0x01000000
#define LCD_CMD       0x01000000
#define LCD_DAT       0x01000001
#define KEY_DAT       0x01000004

#define LINE_1        0x80          // beginning position of LCD line 1

```

```

#define LINE_2    0xC0           // beginning position of LCD line 2
#define LINE_3    0x14           // beginning position of LCD line 3
#define LINE_4    0x54           // beginning position of LCD line 4

#define WaitTime  15
#define DELAY1MS  73             // number of loops = 1ms

// I/O Port Addresses

#define CS3_Base    0x0100
#define LCD_offset  0x0000
#define Keypad_offset 0x0004

#define InitWord1   0x38         //8 bits,2 lines, 5*7 dots
#define InitWord2   0x30         //8 bits 1 line and 5*7 dots
#define InitWord3   0x3c         //8 bits,2 lines, 5*10 dots

#define OnOffCtrl   0x0e
#define DisplayClear 01
#define E_M_ShiftLeft 07
#define E_M_RightIncrm 06
#define NormalWrite 0x14
#define Init_ScrClear 01        // Clear Display, Cursor to Home 1.65ms
#define CrstoHome    02         // Cursor to Home $02 1.65ms
                                // Entry Mode
#define CrsrDcrmnt_Shftoff 04   //Cursor decrement and shift off 40us
#define CrsrDcrmnt_Shfton  05   //Curson Increment and shift on 40 ns
#define CrsrIncrmnt_Shftoff 06 //Cursor Increment and shift off 40us
#define CrsrIncrmnt_Shfton  07 //Cursor Increment and shift on 40us
#define LineLength 20

                                // Display Control:
#define BLOCK 0xFF
#define Resolution 8 // significant number of the converting double,

void DisplayScreen (unsigned char *ptr);
void DisplayScreenWipeOld (unsigned char *ptr);

/*
** Local (Module-Level) Function Prototypes
*/
//-----
// INITIALIZE HARDWARE REGISTERS
// This routine is called once following RESET
void inithw(void);
void InitLCD(void);
void LCDWaitForBusy();

```

```

void LCD_InitDriver (void);
void LCD_WriteControl (unsigned char cmdval);
void LCD_WriteData (unsigned char datval);
void LCDWaitForBusy();

void LCD_Init (void);
void LCD_DefineChar (unsigned char address, const unsigned char *pattern);
void LCD_DisplayOff (void);
void LCD_DisplayOn (void);
void LCD_Clear (void);
void LCD_Home (void);
void LCD_Cursor (char row, char column);
void LCD_ShiftLeft (void);
void LCD_ShiftRight (void);
void LCD_CursorOn (void);
void LCD_CursorOff (void);
void LCD_DisplayCharacter (unsigned char a_char);
void LCD_DisplayString (char row, char column, const char *string);
void LCD_DisplayStringMoving (char row, char column, const char *string);
void LCD_DisplayStringCentered (char row, char *string);
void LCD_DisplayScreen (unsigned char *ptr);
void LCD_DisplayRow (int row, unsigned char *string);
void LCD_DisplayRowShift(int row, unsigned char *string);
void LCD_WipeOnLR ( char *ptr);
void LCD_WipeOnRL ( char *ptr);
void LCD_WipeOffLR (void);
void LCD_WipeOffRL (void);
char *_gcvt(double value, int digits, char *buffer);

```

#### 4. EPLD code

```

Name      AXM0054;
Partno    CME0555;
Revision  01;
Date      01/04/99;
Designer  Forrester;
Company   Axiom Manufacturing;
Assembly  CME0555 lcd decode;
Location  U21;
Device    g22V10;
/*****
*/
/* This device generates the LCD and Key chip select signals */
/* outputs = LCDCS1/2, KEYEN */
/*****
*/

```

```

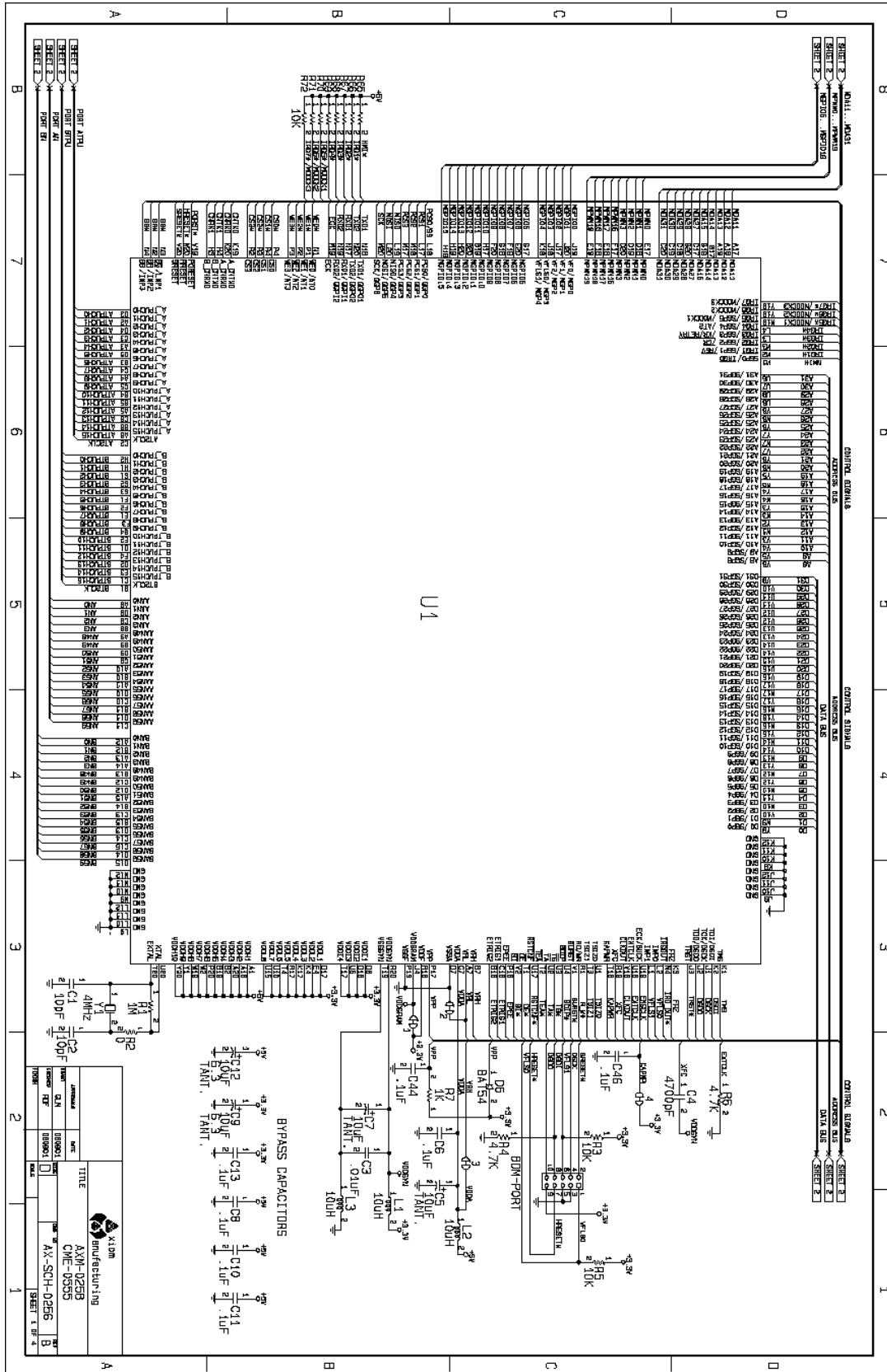
/** Inputs **/
pin 1    = Clk ;           /* CPU clock input      */
pin 2    = !CS_in ;       /* Chip select input    */
pin 3    = R_W ;         /* Read / Write strobe in */
pin [4..11,13..18] = [A14..A7,A6..A1] ; /* system addresses a2 - a15 */
/** Outputs **/
pin [19..20] = [Q0..Q1] ; /* Clock counter        */
pin 21    = L2CS ;       /* LCD 2 CS             */
pin 22    = L1CS ;       /* LCD 1 CS             */
pin 23    = KEY ;        /* Key CS               */
/** Declarations and Intermediate Variable Definitions **/
key_eqn   = [A14..A1] : 0004 ; /* keypad address */
lcd1_eqn  = [A14..A1] : [0000..0001] ; /* LCD 1 address */
lcd2_eqn  = [A14..A1] : [0002..0003] ; /* LCD 2 address */
field count = [Q1..0] ; /* declare counter bit field */
$define S0 'b'00 /* define counter states */
$define S1 'b'01
$define S2 'b'10
$define S3 'b'11
clear = !CS_in ; /* define count clear mode */
up    = CS_in ; /* define count direction */
/** Logic Equations **/
Sequenced count { /* LCD wait state generator */
present S0 if up next S1;
           if clear next S0;
present S1 if up next S2;
           if clear next S0;
present S2 if up next S3;
           if clear next S0;
present S3 if up next S0;
           if clear next S0;
}
L1CS.d = lcd1_eqn & ((Q0 & Q1) # L1CS) ; /* define lcd 1 enable */
L1CS.ar = !CS_in ; /* define async reset */
L2CS.d = lcd2_eqn & ((Q0 & Q1) # L2CS) ; /* define lcd 2 enable */
L2CS.ar = !CS_in ; /* define async resets */
Q0.ar = !CS_in ;
Q1.ar = !CS_in ;
!KEY = key_eqn & R_W & CS_in ; /* Keypad select */

```

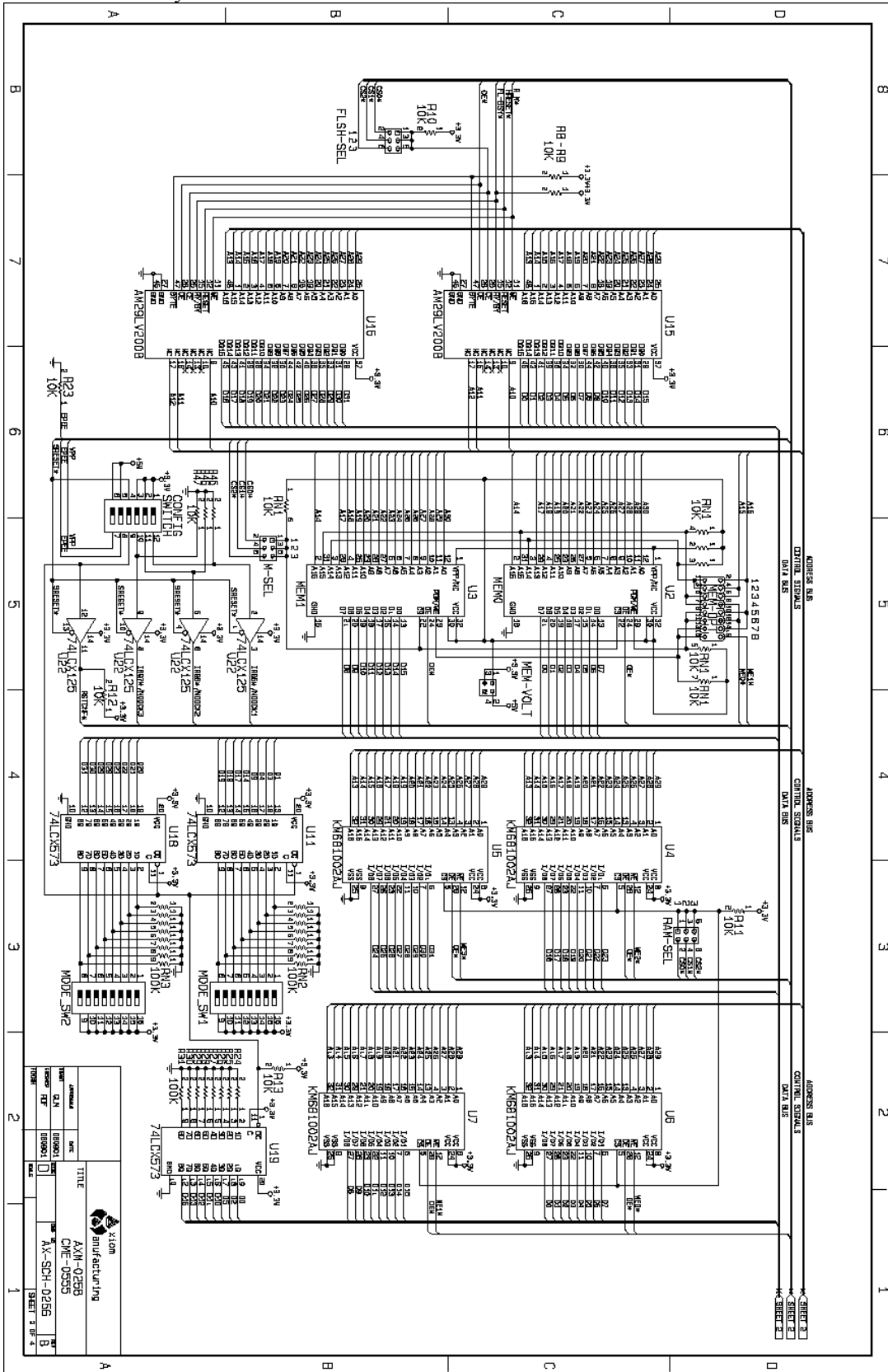
## APPENDIX C SCHEMATICS

### 1. CME555 schematic

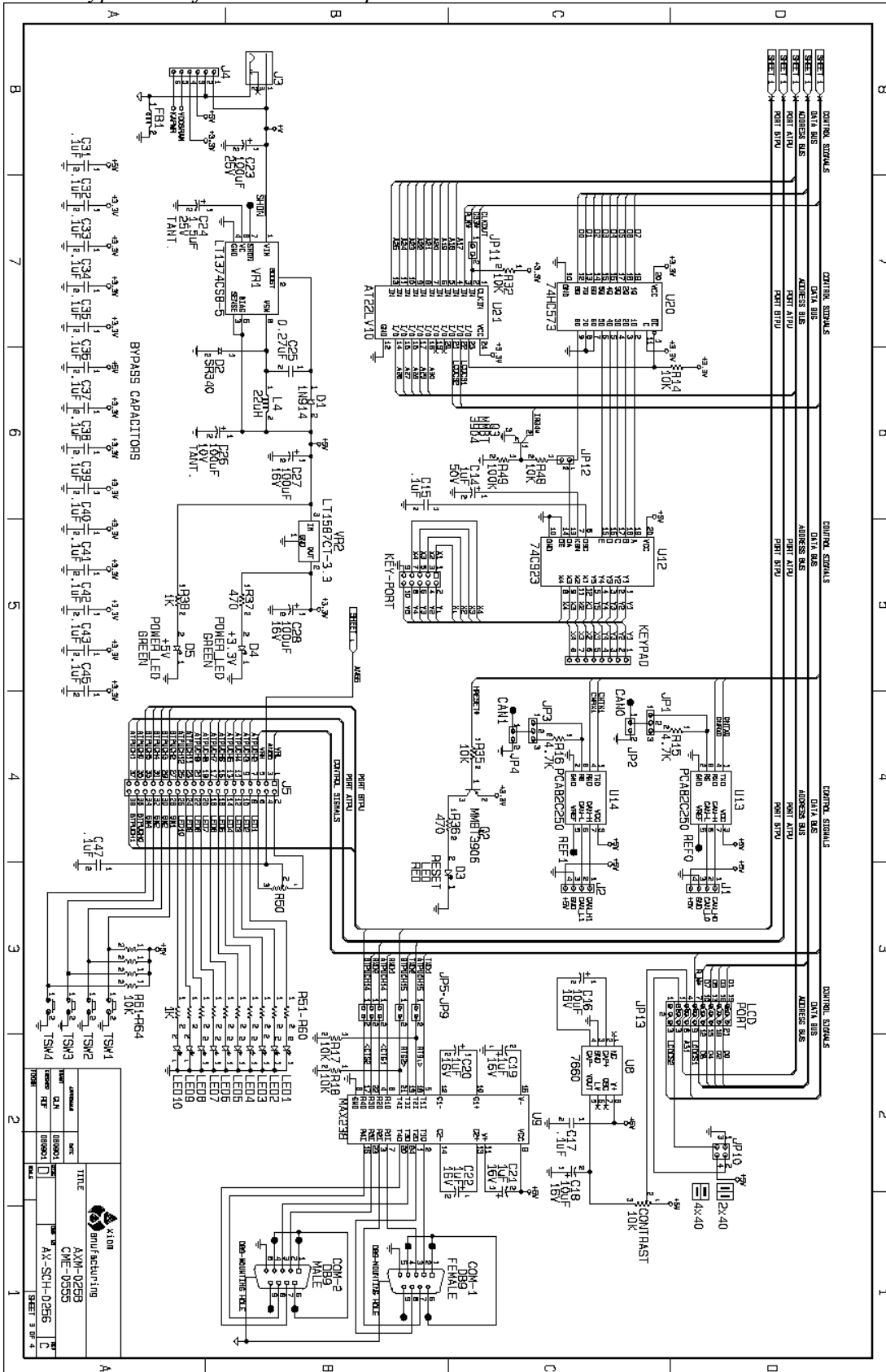
MPC555 connections

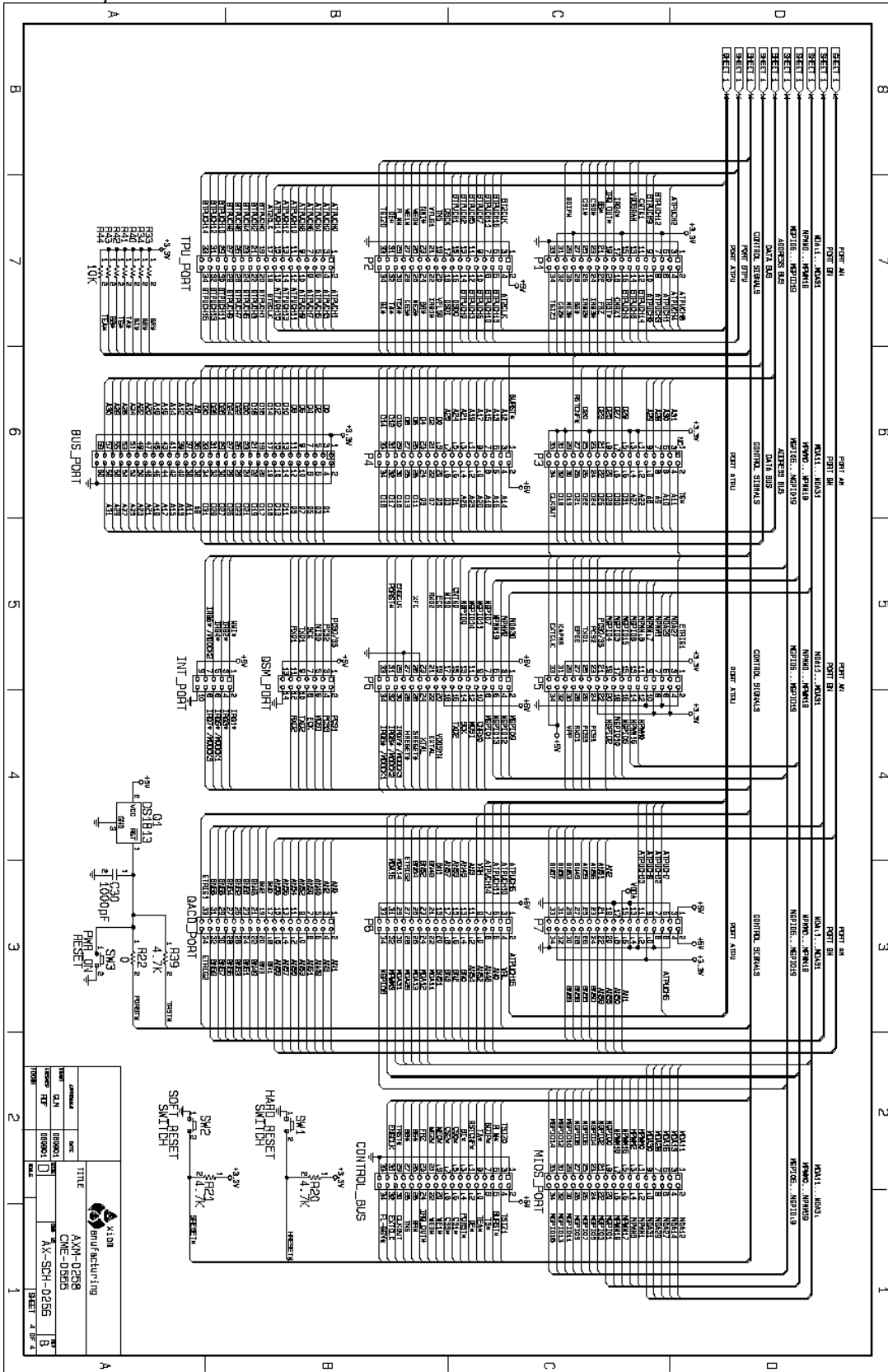


# External Memory Units



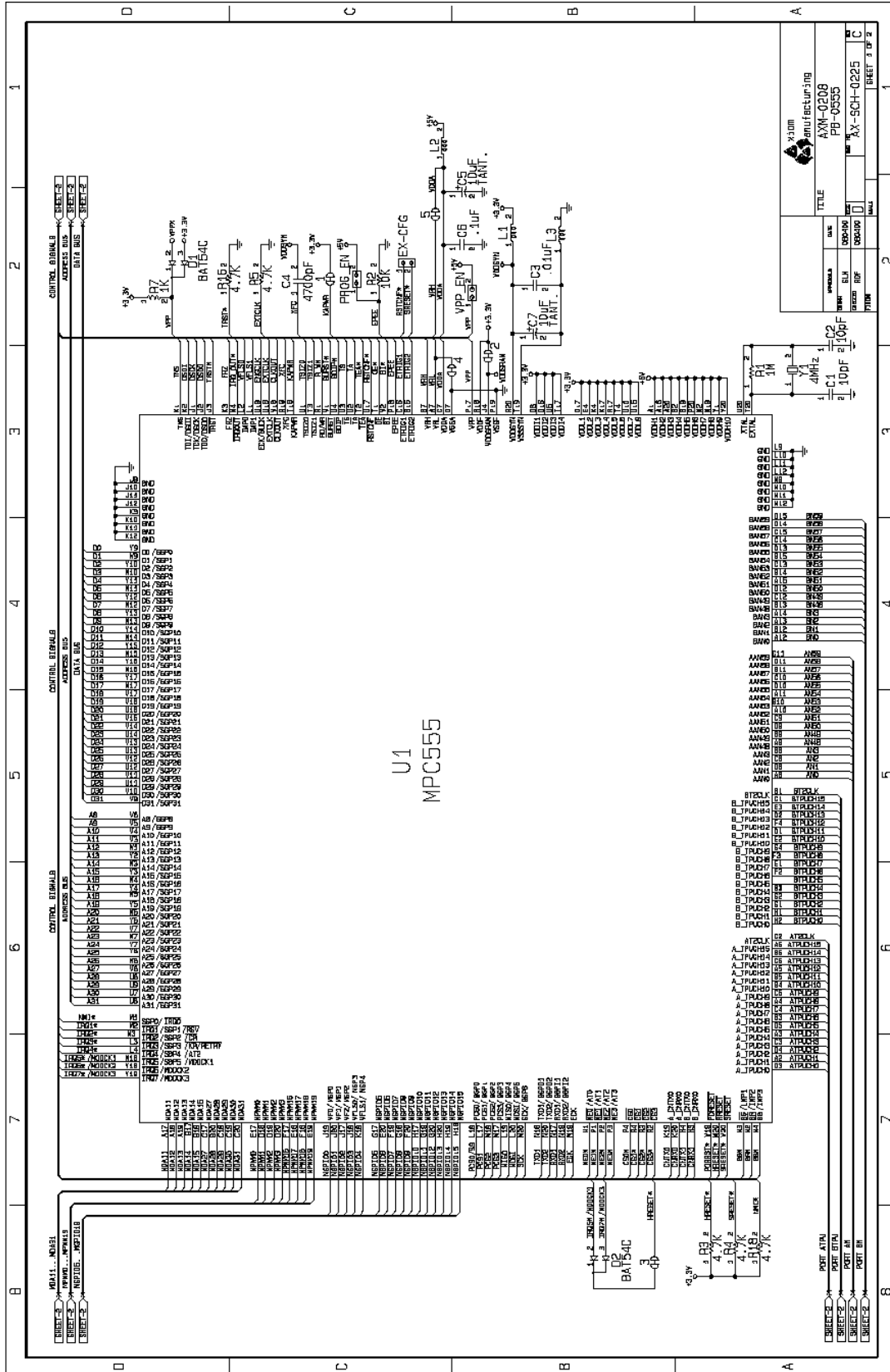
LCD/Keypad interface and RS232 port



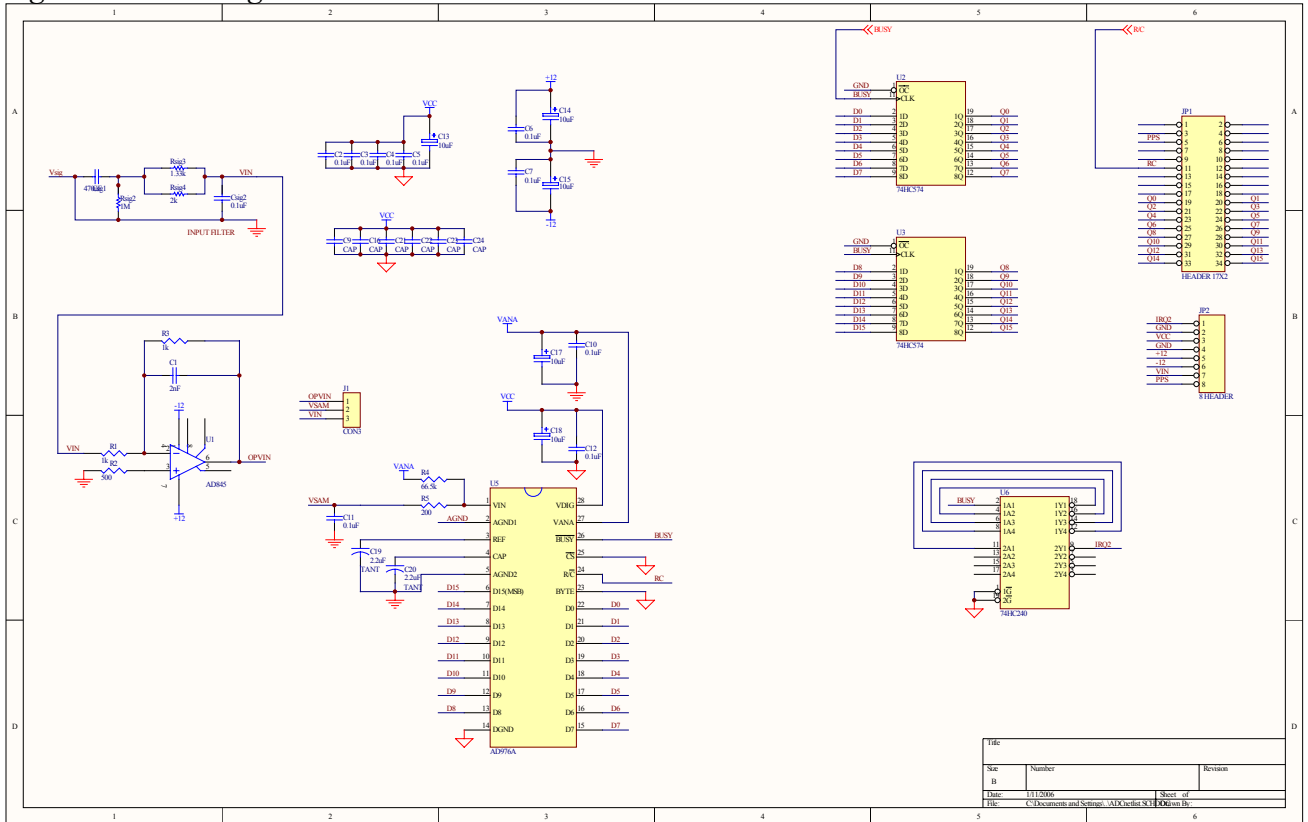




## 2. PB555 schematic with external interface MPC555

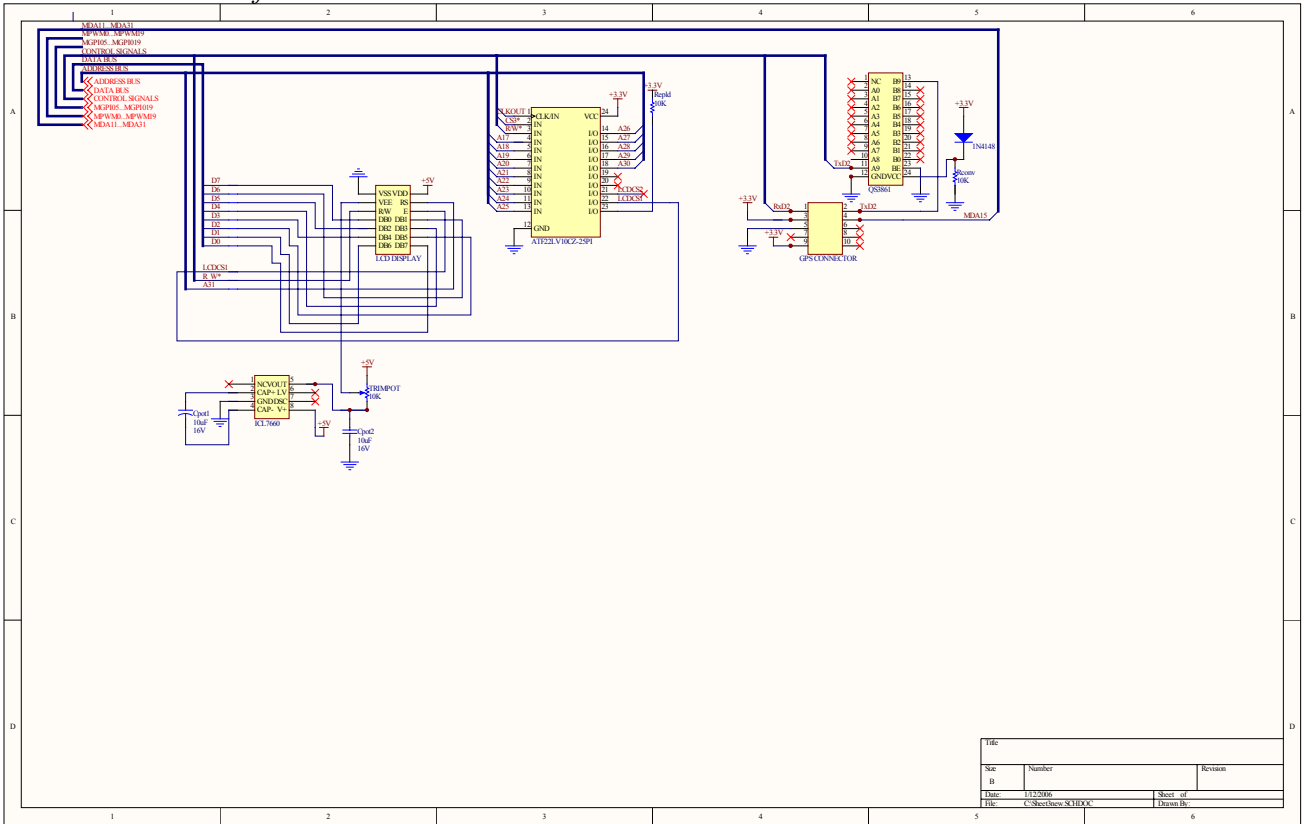


# Signal Conditioning Unit and ADC



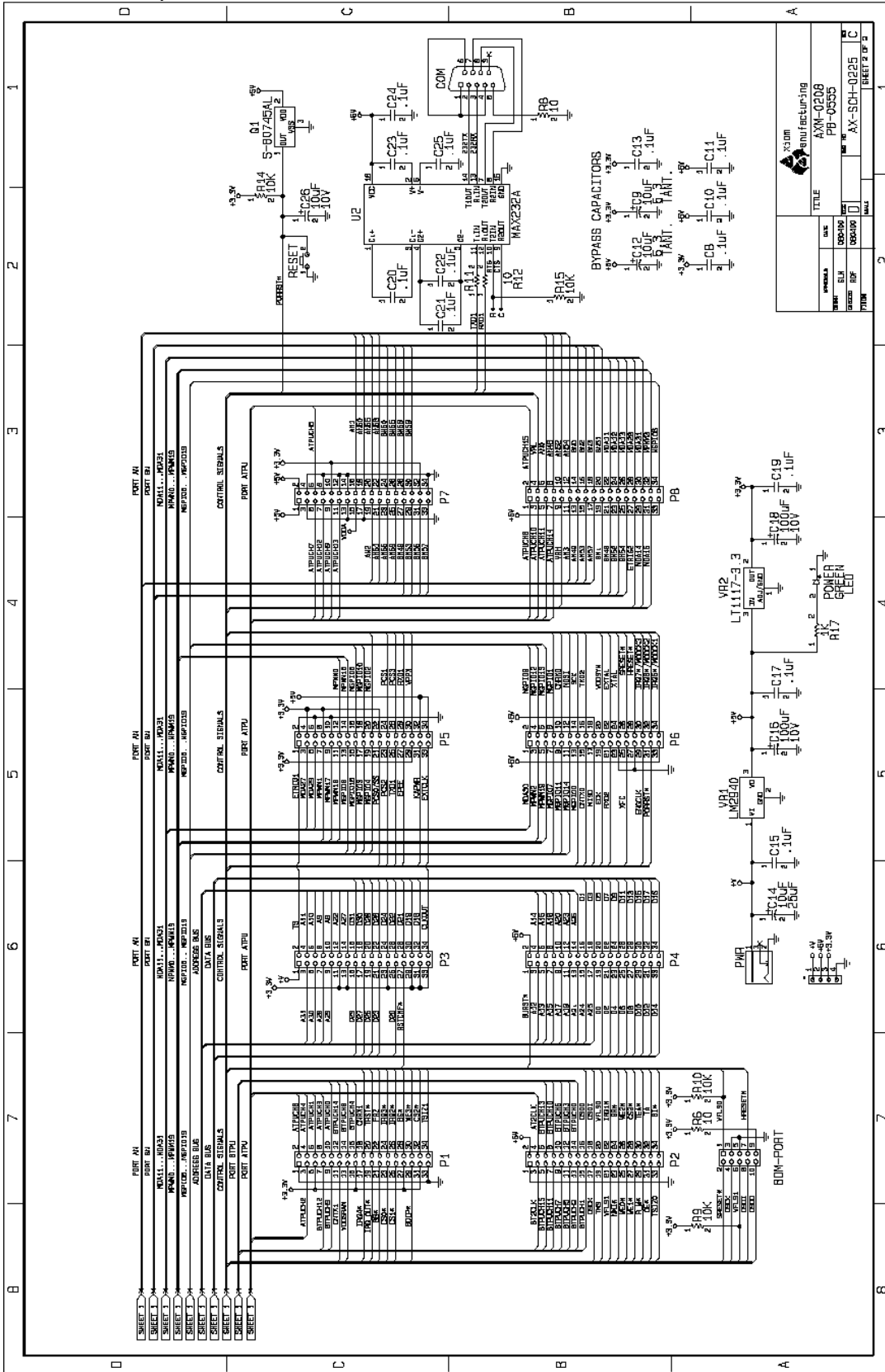
File	Number	Revision
SB		
B		
Date:	11/1/2006	Sheet of
File:	C:\Documents and Settings\ADK\local S... ..	Revision No:

# GPS and LCD interface



Title		
Rev	Number	Revision
B		
Date:	01/2/2006	Sheet of
File:	C:\Sheet\Sheet SCH00	Drawn By:

PB555 header pins



## APPENDIX D

### CME555 JUMPER AND SWITCH SETTING FOR FDR

#### CONFIG SWITCH

Position	Reset Function	Default Position
1	MODCK1 – clock select	Off
2	MODCK2 – clock select = 4MHz Oscillator	On
3	MODCK3 – clock select	Off
4	External Hard Reset Word enable	On
5	EPEE signal High – on-chip program enable	On
6	VPP power enable	On

#### MODE SWITCH 1

Position	Reset Function	Default Position
1	Initial interrupt prefix (IP), On = 1	Off
2	External Boot (BDIS), Disabled = 1	Off
3	Boot Port size (BPS0), on = 16 bit boot port, off = 32 bit	Off
4	Debug pins configuration (DBGCO)	Off
5	External Bus division factor (EBDF1), on = clock/2	Off
6*	Single Chip Mode Select (SC0:1)	Off
7*	Single Chip Mode Select (SC0:1)	Off
8	Exception table relocation enable (ETRE), on = relocated	Off

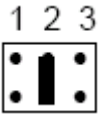
\* Note:

Position 6	Position 7	
Off	Off	Extended 32 bit data bus
Off	On	Extended 16 bit data bus
On	Off	Single chip with show cycles
On	On	Single chip with no external bus

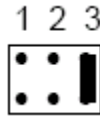
#### MODE SWITCH 2

Position	Reset Function	Default Position
1	Internal Flash enable (FLEN), Off = internal flash removed from memory map	On
2	Enable Compression (EN_COMP)	Off
3	Exception Compression Enable (EN_COMP)	Off
4	Reserved, do not enable	Off
5	Internal register space select (ISB)	Off
6*	off:off:off = 0:0:0, register space at 0x00000000	Off
7*		Off
8	Dual mapping enable (DME), Off = dual mapping disabled	Off

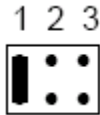
RAM-SEL JUMPER



FLSH-SEL JUMPER



M-SEL JUMPER



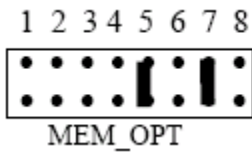
Note: Above jumpers are set to CME555 default settings for FDR

Position 1 = CS0

Position 2 = CS1

Position 3 = CS2

MEM\_OPT JUMPERS

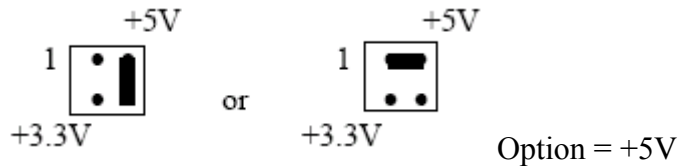


Note:

Position 5 = A16 to U2 pin 29

Position 7 = A16 to U3 pin 29

MEM\_VOLT OPTION JUMPER



JUMPER 10, JUMPER 11 and JUMPER 12 are installed

JUMPER 6, 8, and 9 are installed

## APPENDIX E

### NETLISTING FOR FDR PCB DESIGN

#### 1. Netlist for MPC555 connection

```
(edif Design_FRU_board_simplifiednetlist_SCHDOC
  (edifVersion 2 0 0)
  (edifLevel 0)
  (keywordMap
    (keywordLevel 0)
  )
  (status
    (written
      (timeStamp 2006 1 11 20 18 22)
      (program "Design Explorer DXP - EDIF For PCB"
        (version "1.0.0"))
      )
    (author "EDIF For PCB")
  )
)

(library COMPONENT_LIB
  (edifLevel 0)
  (technology
    (numberDefinition
      (scale 1 1 (unit distance))
    )
  )
  (cell Cap
    (cellType GENERIC)
    (view netListView
      (viewType NETLIST)
      (interface
        (port (rename &1 "1") (direction INOUT))
        (port (rename &2 "2") (direction INOUT))
      )
    )
  )
  (cell GND
    (cellType GENERIC)
    (view netListView
      (viewType NETLIST)
      (interface
        (port Y (direction OUTPUT))
      )
    )
  )
)
```

```

)
(cell (rename Header_2 "Header 2")
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INOUT))
      (port (rename &2 "2") (direction INOUT))
    )
  )
)
)
)
(cell (rename Header_5X2 "Header 5X2")
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INOUT))
      (port (rename &2 "2") (direction INOUT))
      (port (rename &3 "3") (direction INOUT))
      (port (rename &4 "4") (direction INOUT))
      (port (rename &5 "5") (direction INOUT))
      (port (rename &6 "6") (direction INOUT))
      (port (rename &7 "7") (direction INOUT))
      (port (rename &8 "8") (direction INOUT))
      (port (rename &9 "9") (direction INOUT))
      (port (rename &10 "10") (direction INOUT))
    )
  )
)
)
)
)
(cell Inductor
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INOUT))
      (port (rename &2 "2") (direction INOUT))
    )
  )
)
)
)
)
(cell MPC555
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INOUT))
      (port (rename &2 "2") (direction INOUT))
    )
  )
)
)

```



(port (rename &3 "3") (direction INOUT))  
(port (rename &4 "4") (direction INOUT))  
(port (rename &5 "5") (direction INOUT))  
(port (rename &6 "6") (direction INOUT))  
(port (rename &7 "7") (direction INOUT))  
(port (rename &8 "8") (direction INOUT))  
(port (rename &9 "9") (direction INOUT))  
(port (rename &10 "10") (direction INOUT))  
(port (rename &11 "11") (direction INOUT))  
(port (rename &12 "12") (direction INOUT))  
(port (rename &13 "13") (direction INOUT))  
(port (rename &14 "14") (direction INOUT))  
(port (rename &15 "15") (direction INOUT))  
(port (rename &16 "16") (direction INOUT))  
(port (rename &17 "17") (direction INOUT))  
(port (rename &18 "18") (direction INOUT))  
(port (rename &19 "19") (direction INOUT))  
(port (rename &20 "20") (direction INOUT))  
(port (rename &21 "21") (direction INOUT))  
(port (rename &22 "22") (direction INOUT))  
(port (rename &23 "23") (direction INOUT))  
(port (rename &24 "24") (direction INOUT))  
(port (rename &25 "25") (direction INOUT))  
(port (rename &26 "26") (direction INOUT))  
(port (rename &27 "27") (direction INOUT))  
(port (rename &28 "28") (direction INOUT))  
(port (rename &29 "29") (direction INOUT))  
(port (rename &30 "30") (direction INOUT))  
(port (rename &31 "31") (direction INOUT))  
(port (rename &32 "32") (direction INOUT))  
(port (rename &33 "33") (direction INOUT))  
(port (rename &34 "34") (direction INOUT))  
(port (rename &35 "35") (direction INOUT))  
(port (rename &36 "36") (direction INOUT))  
(port (rename &37 "37") (direction INOUT))  
(port (rename &38 "38") (direction INOUT))  
(port (rename &39 "39") (direction INOUT))  
(port (rename &40 "40") (direction INOUT))  
(port (rename &41 "41") (direction INOUT))  
(port (rename &42 "42") (direction INOUT))  
(port (rename &43 "43") (direction INOUT))  
(port (rename &44 "44") (direction INOUT))  
(port (rename &45 "45") (direction INOUT))  
(port (rename &46 "46") (direction INOUT))  
(port (rename &47 "47") (direction INOUT))  
(port (rename &48 "48") (direction INOUT))

(port (rename &49 "49") (direction INOUT))  
(port (rename &50 "50") (direction INOUT))  
(port (rename &51 "51") (direction INOUT))  
(port (rename &52 "52") (direction INOUT))  
(port (rename &53 "53") (direction INOUT))  
(port (rename &54 "54") (direction INOUT))  
(port (rename &55 "55") (direction INOUT))  
(port (rename &56 "56") (direction INOUT))  
(port (rename &57 "57") (direction INOUT))  
(port (rename &58 "58") (direction INOUT))  
(port (rename &59 "59") (direction INOUT))  
(port (rename &60 "60") (direction INOUT))  
(port (rename &61 "61") (direction INOUT))  
(port (rename &62 "62") (direction INOUT))  
(port (rename &63 "63") (direction INOUT))  
(port (rename &64 "64") (direction INOUT))  
(port (rename &65 "65") (direction INOUT))  
(port (rename &66 "66") (direction INOUT))  
(port (rename &67 "67") (direction INOUT))  
(port (rename &68 "68") (direction INOUT))  
(port (rename &69 "69") (direction INOUT))  
(port (rename &70 "70") (direction INOUT))  
(port (rename &71 "71") (direction INOUT))  
(port (rename &72 "72") (direction INOUT))  
(port (rename &73 "73") (direction INOUT))  
(port (rename &74 "74") (direction INOUT))  
(port (rename &75 "75") (direction INOUT))  
(port (rename &76 "76") (direction INOUT))  
(port (rename &77 "77") (direction INOUT))  
(port (rename &78 "78") (direction INOUT))  
(port (rename &79 "79") (direction INOUT))  
(port (rename &80 "80") (direction INOUT))  
(port (rename &81 "81") (direction INOUT))  
(port (rename &82 "82") (direction INOUT))  
(port (rename &83 "83") (direction INOUT))  
(port (rename &84 "84") (direction INOUT))  
(port (rename &85 "85") (direction INOUT))  
(port (rename &86 "86") (direction INOUT))  
(port (rename &87 "87") (direction INOUT))  
(port (rename &88 "88") (direction INOUT))  
(port (rename &89 "89") (direction INOUT))  
(port (rename &90 "90") (direction INOUT))  
(port (rename &91 "91") (direction INOUT))  
(port (rename &92 "92") (direction INOUT))  
(port (rename &93 "93") (direction INOUT))  
(port (rename &94 "94") (direction INOUT))

(port (rename &95 "95") (direction INOUT))  
(port (rename &96 "96") (direction INOUT))  
(port (rename &97 "97") (direction INOUT))  
(port (rename &98 "98") (direction INOUT))  
(port (rename &99 "99") (direction INOUT))  
(port (rename &100 "100") (direction INOUT))  
(port (rename &101 "101") (direction INOUT))  
(port (rename &102 "102") (direction INOUT))  
(port (rename &103 "103") (direction INOUT))  
(port (rename &104 "104") (direction INOUT))  
(port (rename &105 "105") (direction INOUT))  
(port (rename &106 "106") (direction INOUT))  
(port (rename &107 "107") (direction INOUT))  
(port (rename &108 "108") (direction INOUT))  
(port (rename &109 "109") (direction INOUT))  
(port (rename &110 "110") (direction INOUT))  
(port (rename &111 "111") (direction INOUT))  
(port (rename &112 "112") (direction INOUT))  
(port (rename &113 "113") (direction INOUT))  
(port (rename &114 "114") (direction INOUT))  
(port (rename &115 "115") (direction INOUT))  
(port (rename &116 "116") (direction INOUT))  
(port (rename &117 "117") (direction INOUT))  
(port (rename &118 "118") (direction INOUT))  
(port (rename &119 "119") (direction INOUT))  
(port (rename &120 "120") (direction INOUT))  
(port (rename &121 "121") (direction INOUT))  
(port (rename &122 "122") (direction INOUT))  
(port (rename &123 "123") (direction INOUT))  
(port (rename &124 "124") (direction INOUT))  
(port (rename &125 "125") (direction INOUT))  
(port (rename &126 "126") (direction INOUT))  
(port (rename &127 "127") (direction INOUT))  
(port (rename &128 "128") (direction INOUT))  
(port (rename &129 "129") (direction INOUT))  
(port (rename &130 "130") (direction INOUT))  
(port (rename &131 "131") (direction INOUT))  
(port (rename &132 "132") (direction INOUT))  
(port (rename &133 "133") (direction INOUT))  
(port (rename &134 "134") (direction INOUT))  
(port (rename &135 "135") (direction INOUT))  
(port (rename &136 "136") (direction INOUT))  
(port (rename &137 "137") (direction INOUT))  
(port (rename &138 "138") (direction INOUT))  
(port (rename &139 "139") (direction INOUT))  
(port (rename &140 "140") (direction INOUT))

(port (rename &141 "141") (direction INOUT))  
(port (rename &142 "142") (direction INOUT))  
(port (rename &143 "143") (direction INOUT))  
(port (rename &144 "144") (direction INOUT))  
(port (rename &145 "145") (direction INOUT))  
(port (rename &146 "146") (direction INOUT))  
(port (rename &147 "147") (direction INOUT))  
(port (rename &148 "148") (direction INOUT))  
(port (rename &149 "149") (direction INOUT))  
(port (rename &150 "150") (direction INOUT))  
(port (rename &151 "151") (direction INOUT))  
(port (rename &152 "152") (direction INOUT))  
(port (rename &153 "153") (direction INOUT))  
(port (rename &154 "154") (direction INOUT))  
(port (rename &155 "155") (direction INOUT))  
(port (rename &156 "156") (direction INOUT))  
(port (rename &157 "157") (direction INOUT))  
(port (rename &158 "158") (direction INOUT))  
(port (rename &159 "159") (direction INOUT))  
(port (rename &160 "160") (direction INOUT))  
(port (rename &161 "161") (direction INOUT))  
(port (rename &162 "162") (direction INOUT))  
(port (rename &163 "163") (direction INOUT))  
(port (rename &164 "164") (direction INOUT))  
(port (rename &165 "165") (direction INOUT))  
(port (rename &166 "166") (direction INOUT))  
(port (rename &167 "167") (direction INOUT))  
(port (rename &168 "168") (direction INOUT))  
(port (rename &169 "169") (direction INOUT))  
(port (rename &170 "170") (direction INOUT))  
(port (rename &171 "171") (direction INOUT))  
(port (rename &172 "172") (direction INOUT))  
(port (rename &173 "173") (direction INOUT))  
(port (rename &174 "174") (direction INOUT))  
(port (rename &175 "175") (direction INOUT))  
(port (rename &176 "176") (direction INOUT))  
(port (rename &177 "177") (direction INOUT))  
(port (rename &178 "178") (direction INOUT))  
(port (rename &179 "179") (direction INOUT))  
(port (rename &180 "180") (direction INOUT))  
(port (rename &181 "181") (direction INOUT))  
(port (rename &182 "182") (direction INOUT))  
(port (rename &183 "183") (direction INOUT))  
(port (rename &184 "184") (direction INOUT))  
(port (rename &185 "185") (direction INOUT))  
(port (rename &186 "186") (direction INOUT))

(port (rename &187 "187") (direction INOUT))  
(port (rename &188 "188") (direction INOUT))  
(port (rename &189 "189") (direction INOUT))  
(port (rename &190 "190") (direction INOUT))  
(port (rename &191 "191") (direction INOUT))  
(port (rename &192 "192") (direction INOUT))  
(port (rename &193 "193") (direction INOUT))  
(port (rename &194 "194") (direction INOUT))  
(port (rename &195 "195") (direction INOUT))  
(port (rename &196 "196") (direction INOUT))  
(port (rename &197 "197") (direction INOUT))  
(port (rename &198 "198") (direction INOUT))  
(port (rename &199 "199") (direction INOUT))  
(port (rename &200 "200") (direction INOUT))  
(port (rename &201 "201") (direction INOUT))  
(port (rename &202 "202") (direction INOUT))  
(port (rename &203 "203") (direction INOUT))  
(port (rename &204 "204") (direction INOUT))  
(port (rename &205 "205") (direction INOUT))  
(port (rename &206 "206") (direction INOUT))  
(port (rename &207 "207") (direction INOUT))  
(port (rename &208 "208") (direction INOUT))  
(port (rename &209 "209") (direction INOUT))  
(port (rename &210 "210") (direction INOUT))  
(port (rename &211 "211") (direction INOUT))  
(port (rename &212 "212") (direction INOUT))  
(port (rename &213 "213") (direction INOUT))  
(port (rename &214 "214") (direction INOUT))  
(port (rename &215 "215") (direction INOUT))  
(port (rename &216 "216") (direction INOUT))  
(port (rename &217 "217") (direction INOUT))  
(port (rename &218 "218") (direction INOUT))  
(port (rename &219 "219") (direction INOUT))  
(port (rename &220 "220") (direction INOUT))  
(port (rename &221 "221") (direction INOUT))  
(port (rename &222 "222") (direction INOUT))  
(port (rename &223 "223") (direction INOUT))  
(port (rename &224 "224") (direction INOUT))  
(port (rename &225 "225") (direction INOUT))  
(port (rename &226 "226") (direction INOUT))  
(port (rename &227 "227") (direction INOUT))  
(port (rename &228 "228") (direction INOUT))  
(port (rename &229 "229") (direction INOUT))  
(port (rename &230 "230") (direction INOUT))  
(port (rename &231 "231") (direction INOUT))  
(port (rename &232 "232") (direction INOUT))

```
(port (rename &233 "233") (direction INOUT))
(port (rename &234 "234") (direction INOUT))
(port (rename &235 "235") (direction INOUT))
(port (rename &236 "236") (direction INOUT))
(port (rename &237 "237") (direction INOUT))
(port (rename &238 "238") (direction INOUT))
(port (rename &239 "239") (direction INOUT))
(port (rename &240 "240") (direction INOUT))
(port (rename &241 "241") (direction INOUT))
(port (rename &242 "242") (direction INOUT))
(port (rename &243 "243") (direction INOUT))
(port (rename &244 "244") (direction INOUT))
(port (rename &245 "245") (direction INOUT))
(port (rename &246 "246") (direction INOUT))
(port (rename &247 "247") (direction INOUT))
(port (rename &248 "248") (direction INOUT))
(port (rename &249 "249") (direction INOUT))
(port (rename &250 "250") (direction INOUT))
(port (rename &251 "251") (direction INOUT))
(port (rename &252 "252") (direction INOUT))
(port (rename &253 "253") (direction INOUT))
(port (rename &254 "254") (direction INOUT))
(port (rename &255 "255") (direction INOUT))
(port (rename &256 "256") (direction INOUT))
(port (rename &257 "257") (direction INOUT))
(port (rename &258 "258") (direction INOUT))
(port (rename &259 "259") (direction INOUT))
(port (rename &260 "260") (direction INOUT))
(port (rename &261 "261") (direction INOUT))
(port (rename &262 "262") (direction INOUT))
(port (rename &263 "263") (direction INOUT))
(port (rename &264 "264") (direction INOUT))
(port (rename &265 "265") (direction INOUT))
(port (rename &266 "266") (direction INOUT))
(port (rename &267 "267") (direction INOUT))
(port (rename &268 "268") (direction INOUT))
(port (rename &269 "269") (direction INOUT))
(port (rename &270 "270") (direction INOUT))
(port (rename &271 "271") (direction INOUT))
(port (rename &272 "272") (direction INOUT))
)
)
)
(cell Res1
  (cellType GENERIC)
  (view netListView
```









(Property (rename Component\_Kind "Component Kind") (String "Standard" ))  
(Property (rename Component\_Kind "Component Kind") (String "Standard" ))  
(Property (rename Component\_Kind "Component Kind") (String "Standard" ))  
(Property (rename Component\_Kind "Component Kind") (String "Standard" ))  
(Property (rename Component\_Kind "Component Kind") (String "Standard" ))  
(Property (rename Component\_Kind "Component Kind") (String "Standard" ))  
(Property (rename Component\_Kind "Component Kind") (String "Standard" ))  
(Property Description (String "Resistor" ))  
(Property Description (String "Resistor" ))  
(Property Description (String "Resistor" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Resistor" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Crystal Oscillator" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Inductor" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Resistor" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Capacitor" ))  
(Property Description (String "Resistor" ))  
(Property Description (String "Capacitor" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Resistor" ))  
(Property Description (String "Resistor" ))  
(Property Description (String "Resistor" ))  
(Property Description (String "Resistor" ))  
(Property Description (String "Resistor" ))  
(Property Description (String "Resistor" ))  
(Property Description (String "Resistor" ))  
(Property Description (String "Resistor" ))  
(Property Description (String "Resistor" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Ground" ))  
(Property FPGAVendor (String "Actel" ))  
(Property FPGAVendor (String "Actel" ))  
(Property FPGAVendor (String "Actel" ))  
(Property FPGAVendor (String "Actel" ))  
(Property FPGAVendor (String "Actel" ))  
(Property FPGAVendor (String "Actel" ))

(Property FPGAVendor (String "Actel" ))  
 (Property FPGAVendor (String "Actel" ))  
 (Property FPGAVendor (String "Actel" ))  
 (Property FPGAVendor (String "Actel" ))  
 (Property FPGAVendor (String "Actel" ))  
 (Property FPGAVendor (String "Actel" ))  
 (Property FPGAVendor (String "Actel" ))  
 (Property FPGAVendor (String "Actel" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property Footprint (String "BCY-W2/D3.1" ))  
 (Property Footprint (String "C1005-0402" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property Footprint (String "RAD-0.3" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property Footprint (String "RAD-0.3" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property Footprint (String "AXIAL-0.3" ))  
 (Property (rename Library\_Name "Library Name") (String "Miscellaneous  
 Devices.IntLib" ))  
 (Property (rename Library\_Name "Library Name") (String "Miscellaneous  
 Devices.IntLib" ))  
 (Property (rename Library\_Name "Library Name") (String "Miscellaneous  
 Devices.IntLib" ))  
 (Property (rename Library\_Name "Library Name") (String "Actel 3200DX  
 FPGA.IntLib" ))  
 (Property (rename Library\_Name "Library Name") (String "Miscellaneous  
 Devices.IntLib" ))  
 (Property (rename Library\_Name "Library Name") (String "Actel 3200DX  
 FPGA.IntLib" ))  
 (Property (rename Library\_Name "Library Name") (String "Actel 3200DX  
 FPGA.IntLib" ))  
 (Property (rename Library\_Name "Library Name") (String "Actel 3200DX  
 FPGA.IntLib" ))  
 (Property (rename Library\_Name "Library Name") (String "Actel 3200DX  
 FPGA.IntLib" ))





(Property Manufacturer (String "Generic Components" ))  
(Property Manufacturer (String "Generic Components" ))  
(Property Manufacturer (String "Generic Components" ))  
(Property Manufacturer (String "Generic Components" ))  
(Property Manufacturer (String "Generic Components" ))  
(Property Manufacturer (String "Generic Components" ))  
(Property Manufacturer (String "Generic Components" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))  
(Property Publisher (String "Altium Limited" ))  
(Property Publisher (String "Altium Limited" ))  
(Property Publisher (String "Altium Limited" ))  
(Property Publisher (String "Altium Hobart Technology Centre" ))  
(Property Publisher (String "Altium Limited" ))  
(Property Publisher (String "Altium Hobart Technology Centre" ))



```

(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Inductor" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Cap" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Cap" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property Simulation (String "RESISTOR" ))
(Property Simulation (String "RESISTOR" ))
(Property Simulation (String "RESISTOR" ))
(Property Simulation (String "RESISTOR" ))
(Property Simulation (String "XTAL" ))
(Property Simulation (String "INDUCTOR" ))
(Property Simulation (String "RESISTOR" ))
(Property Simulation (String "CAP" ))
(Property Simulation (String "RESISTOR" ))
(Property Simulation (String "CAP" ))
(Property Simulation (String "RESISTOR" ))
(Property Simulation (String "RESISTOR" ))
(Property Simulation (String "RESISTOR" ))
(Property Simulation (String "RESISTOR" ))
(Property Simulation (String "RESISTOR" ))
(Property Simulation (String "RESISTOR" ))
(Property Simulation (String "RESISTOR" ))
(Property Simulation (String "RESISTOR" ))
(Property Simulation (String "RESISTOR" ))
(Property Value (String "10K" ))
(Property Value (String "10uH" ))
(Property Value (String "10K" ))
(Property Value (String "0.1uF" ))
(Property Value (String "10K" ))
(Property Value (String "4700pF" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property Description (String "Ground" ))
(Property UniqueId (String "$$$\DWKVTYMS" ))
)
(Instance (rename BDM_PORT "BDM-PORT")
(viewRef NetlistView

```



```

    (cellRef Header_5X2
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property (rename Class_I "Class I") (String "Connector" ))
  (Property Comment (String "BDM-PORT" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "Header, 5-Pin, Dual row" ))
  (Property Footprint (String "HDR2X5" ))
  (Property (rename Library_Name "Library Name") (String "Miscellaneous
Connectors.IntLib" ))
  (Property (rename Library_Reference "Library Reference") (String "Header 5X2"
))
  (Property Manufacturer (String "Generic Components" ))
  (Property Publisher (String "Altium Limited" ))
  (Property Revision (String "July-2002: Re-released for DXP Platform." ))
  (Property (rename Signal_Integrity "Signal Integrity") (String "Connector" ))
  (Property Footprint (String "HDR2X5" ))
  (Property Description (String "Header, 5-Pin, Dual row" ))
  (Property UniqueId (String "$$$\SGDCJUAM" ))
)
(Instance Cmpc1
  (viewRef NetlistView
    (cellRef Cap
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property (rename Class_I "Class I") (String "Passive" ))
  (Property (rename Class_II "Class II") (String "Capacitor" ))
  (Property Comment (String "Cmpc1" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "Capacitor" ))
  (Property Footprint (String "RAD-0.3" ))
  (Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
  (Property (rename Library_Reference "Library Reference") (String "Cap" ))
  (Property Manufacturer (String "Generic Components" ))
  (Property Published (String "8-Jun-2000" ))
  (Property Publisher (String "Altium Limited" ))
  (Property Revision (String "July-2002: Re-released for DXP Platform." ))
  (Property (rename Signal_Integrity "Signal Integrity") (String "Cap" ))
  (Property Simulation (String "CAP" ))
  (Property Value (String "10 pF" ))
  (Property Footprint (String "RAD-0.3" ))
  (Property Description (String "Capacitor" ))
  (Property UniqueId (String "$$$\YFUUHWBQ" ))
)

```

```

)
(Instance Cmpc2
  (viewRef NetlistView
    (cellRef Cap
      (LibraryRef COMPONENT_LIB)
    )
  )
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Capacitor" ))
(Property Comment (String "Cmpc2" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "RAD-0.3" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Cap" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Cap" ))
(Property Simulation (String "CAP" ))
(Property Value (String "10 pF" ))
(Property Footprint (String "RAD-0.3" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$JFJOSIFA" ))
)
(Instance Cmpc3
  (viewRef NetlistView
    (cellRef Cap
      (LibraryRef COMPONENT_LIB)
    )
  )
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Capacitor" ))
(Property Comment (String "Cmpc3" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "RAD-0.3" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Cap" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))

```

```

(Property (rename Signal_Integrity "Signal Integrity") (String "Cap" ))
(Property Simulation (String "CAP" ))
(Property Value (String "10uF" ))
(Property Footprint (String "RAD-0.3" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\VEUQFLOU" ))
)
(Instance Cmpc4
  (viewRef NetlistView
    (cellRef Cap
      (LibraryRef COMPONENT_LIB)
    )
  )
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Capacitor" ))
(Property Comment (String "Cmpc4" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "RAD-0.3" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Cap" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Cap" ))
(Property Simulation (String "CAP" ))
(Property Value (String "0.01uF" ))
(Property Footprint (String "RAD-0.3" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\JTAMHQGF" ))
)
(Instance (rename L_ "L?")
  (viewRef NetlistView
    (cellRef Inductor
      (LibraryRef COMPONENT_LIB)
    )
  )
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Inductor" ))
(Property Comment (String "" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Inductor" ))
(Property Footprint (String "C1005-0402" ))

```

```

(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Inductor" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Inductor" ))
(Property Simulation (String "INDUCTOR" ))
(Property Value (String "10uH" ))
(Property Footprint (String "C1005-0402" ))
(Property Description (String "Inductor" ))
(Property UniqueId (String "$$$\TDQRLSQC" ))
)
(Instance MPC555
(viewRef NetlistView
(cellRef MPC555
(LibraryRef COMPONENT_LIB)
)
)
(Property Comment (String "MPC555" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "" ))
(Property Footprint (String "None Available" ))
(Property (rename Library_Name "Library Name") (String "MPC555.SCHLIB" ))
(Property (rename Library_Reference "Library Reference") (String "MPC555" ))
(Property Footprint (String "None Available" ))
(Property UniqueId (String "$$$\IUXFQDYV" ))
)
(Instance MPC556
(viewRef NetlistView
(cellRef MPC555
(LibraryRef COMPONENT_LIB)
)
)
(Property Comment (String "MPC555" ))
(Property Comment (String "MPC555" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "" ))
(Property Description (String "" ))
(Property Footprint (String "None Available" ))
(Property Footprint (String "None Available" ))
(Property (rename Library_Name "Library Name") (String "MPC555.SCHLIB" ))
(Property (rename Library_Name "Library Name") (String "MPC555.SCHLIB" ))
(Property (rename Library_Reference "Library Reference") (String "MPC555" ))
)

```

```

(Property (rename Library_Reference "Library Reference") (String "MPC555" ))
(Property Footprint (String "None Available" ))
(Property UniqueId (String "$$$\MFUBTJQP" ))
)
(Instance MPC557
(viewRef NetlistView
(cellRef MPC555
(LibraryRef COMPONENT_LIB)
)
)
)
(Property Comment (String "MPC555" ))
(Property Comment (String "MPC555" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "" ))
(Property Description (String "" ))
(Property Footprint (String "None Available" ))
(Property Footprint (String "None Available" ))
(Property (rename Library_Name "Library Name") (String "MPC555.SCHLIB" ))
(Property (rename Library_Name "Library Name") (String "MPC555.SCHLIB" ))
(Property (rename Library_Reference "Library Reference") (String "MPC555" ))
(Property (rename Library_Reference "Library Reference") (String "MPC555" ))
(Property Footprint (String "None Available" ))
(Property UniqueId (String "$$$\QJUPEGTH" ))
)
(Instance MPC558
(viewRef NetlistView
(cellRef MPC555
(LibraryRef COMPONENT_LIB)
)
)
)
(Property Comment (String "MPC555" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "" ))
(Property Footprint (String "None Available" ))
(Property (rename Library_Name "Library Name") (String "MPC555.SCHLIB" ))
(Property (rename Library_Reference "Library Reference") (String "MPC555" ))
(Property Footprint (String "None Available" ))
(Property UniqueId (String "$$$\LGBPAMQO" ))
)
)
(Instance (rename PROG_EN_Jumper "PROG_EN Jumper")
(viewRef NetlistView
(cellRef Header_2
(LibraryRef COMPONENT_LIB)
)
)
)
)

```

```

(Property (rename Class_I "Class I") (String "Connector" ))
(Property Comment (String "PROG_EN Jumper" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Header, 2-Pin" ))
(Property Footprint (String "HDR1X2" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Connectors.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Header 2" ))
(Property Manufacturer (String "Generic Components" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Connector" ))
(Property Footprint (String "HDR1X2" ))
(Property Description (String "Header, 2-Pin" ))
(Property UniqueId (String "$$$\MXTHXKOB" ))
)
(Instance Rmpc1
(viewRef NetlistView
(cellRef Res1
(LibraryRef COMPONENT_LIB)
)
)
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Resistor" ))
(Property Comment (String "Rmpc1" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Resistor" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Res1" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property Simulation (String "RESISTOR" ))
(Property Value (String "10K" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property Description (String "Resistor" ))
(Property UniqueId (String "$$$\LLNLHDAN" ))
)
(Instance Rmpc2
(viewRef NetlistView
(cellRef Res1
(LibraryRef COMPONENT_LIB)
)
)
)

```

```

)
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Resistor" ))
(Property Comment (String "Rmpc2" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Resistor" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Res1" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property Simulation (String "RESISTOR" ))
(Property Value (String "10K" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property Description (String "Resistor" ))
(Property UniqueId (String "$$$\OCYFPJKP" ))
)
)
(Instance Rmpc4
(viewRef NetlistView
(cellRef Res1
(LibraryRef COMPONENT_LIB)
)
)
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Resistor" ))
(Property Comment (String "Rmpc4" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Resistor" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Res1" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property Simulation (String "RESISTOR" ))
(Property Value (String "4.7K" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property Description (String "Resistor" ))

```

```

(Property UniqueId (String "$$$\GMYDBEQU" ))
)
(Instance Rmpc5
(viewRef NetlistView
(cellRef Res1
(LibraryRef COMPONENT_LIB)
)
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Resistor" ))
(Property Comment (String "Rmpc5" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Resistor" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Res1" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property Simulation (String "RESISTOR" ))
(Property Value (String "10K" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property Description (String "Resistor" ))
(Property UniqueId (String "$$$\LGPHYTFA" ))
)
(Instance Rmpc6
(viewRef NetlistView
(cellRef Res1
(LibraryRef COMPONENT_LIB)
)
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Resistor" ))
(Property Comment (String "Rmpc6" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Resistor" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Res1" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))

```



```

(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property Simulation (String "RESISTOR" ))
(Property Value (String "4.7K" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property Description (String "Resistor" ))
(Property UniqueId (String "$$$\LPLIIFSA" ))
)
(Instance Rmpc7
(viewRef NetlistView
(cellRef Res1
(LibraryRef COMPONENT_LIB)
)
)
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Resistor" ))
(Property Comment (String "Rmpc7" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Resistor" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Res1" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property Simulation (String "RESISTOR" ))
(Property Value (String "4.7K" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property Description (String "Resistor" ))
(Property UniqueId (String "$$$\RPSOGLVS" ))
)
(Instance Rmpc8
(viewRef NetlistView
(cellRef Res1
(LibraryRef COMPONENT_LIB)
)
)
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Resistor" ))
(Property Comment (String "Rmpc8" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Resistor" ))
(Property Footprint (String "AXIAL-0.3" ))

```

```

(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Res1" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property Simulation (String "RESISTOR" ))
(Property Value (String "10K" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property Description (String "Resistor" ))
(Property UniqueId (String "$$$\HNXARVLD" ))
)
(Instance Rmpc9
(viewRef NetlistView
(cellRef Res1
(LibraryRef COMPONENT_LIB)
)
)
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Resistor" ))
(Property (rename Class_II "Class II") (String "Resistor" ))
(Property Comment (String "Rmpc9" ))
(Property Comment (String "Rmpc9" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Resistor" ))
(Property Description (String "Resistor" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Res1" ))
(Property (rename Library_Reference "Library Reference") (String "Res1" ))
(Property Manufacturer (String "Generic Components" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))

```

```

(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property Simulation (String "RESISTOR" ))
(Property Simulation (String "RESISTOR" ))
(Property Value (String "10K" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property Description (String "Resistor" ))
(Property UniqueId (String "$$$\YNDVQOOM" ))
)
(Instance (rename VPP_EN_Jumper "VPP_EN Jumper")
(viewRef NetlistView
(cellRef Header_2
(LibraryRef COMPONENT_LIB)
)
)
)
(Property (rename Class_I "Class I") (String "Connector" ))
(Property Comment (String "VPP_EN Jumper" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Header, 2-Pin" ))
(Property Footprint (String "HDR1X2" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Connectors.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Header 2" ))
(Property Manufacturer (String "Generic Components" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Connector" ))
(Property Footprint (String "HDR1X2" ))
(Property Description (String "Header, 2-Pin" ))
(Property UniqueId (String "$$$\AHHLJKGV" ))
)
)
(Net (rename PLUS3_3_V "+3.3 V")
(Joined
(PortRef &1)
(PortRef &1)
(PortRef &1)
)
)
)
(Net (rename PLUS5_V "+5 V")
(Joined
(PortRef &1)
(PortRef &1)
(PortRef &1)
(PortRef &1)
(PortRef &1)
(PortRef &1)
)
)

```

```

    (PortRef &1)
    (PortRef &1)
  )
)
(Net (rename PLUS5V "+5V")
  (Joined
    (PortRef &2 (InstanceRef PROG_EN Jumper))
    (PortRef &2 (InstanceRef Rmpc1))
    (PortRef &2 (InstanceRef Rmpc2))
    (PortRef &1 (InstanceRef Rmpc4))
    (PortRef &1 (InstanceRef VPP_EN Jumper))
  )
)
(Net (rename &10K "10K")
  (Joined
  )
)
(Net (rename ADDRESS_BUS "ADDRESS BUS")
  (Joined
  )
)
(Net (rename CONTROL_SIGNALS "CONTROL SIGNALS")
  (Joined
  )
)
(Net D0
  (Joined
    (PortRef &209 (InstanceRef MPC555))
  )
)
(Net D1
  (Joined
    (PortRef &210 (InstanceRef MPC555))
  )
)
(Net D2
  (Joined
    (PortRef &211 (InstanceRef MPC555))
  )
)
(Net D3
  (Joined
    (PortRef &212 (InstanceRef MPC555))
  )
)
(Net D4

```

```

(Joined
  (PortRef &213 (InstanceRef MPC555))
)
)
(Net D5
  (Joined
    (PortRef &214 (InstanceRef MPC555))
  )
)
(Net D6
  (Joined
    (PortRef &215 (InstanceRef MPC555))
  )
)
(Net D7
  (Joined
    (PortRef &216 (InstanceRef MPC555))
  )
)
(Net (rename DATA_BUS "DATA BUS")
  (Joined
  )
)
(Net DSCK
  (Joined
    (PortRef &4 (InstanceRef BDM-PORT))
  )
)
(Net DSDI
  (Joined
    (PortRef &8 (InstanceRef BDM-PORT))
    (PortRef &2 (InstanceRef Rmpc6))
  )
)
(Net DSDO
  (Joined
    (PortRef &10 (InstanceRef BDM-PORT))
  )
)
(Net ENGCLK
  (Joined
  )
)
(Net EPEE
  (Joined
    (PortRef &1 (InstanceRef PROG_EN Jumper))
  )
)

```

```

    (PortRef &1 (InstanceRef Rmpc5))
  )
)
(Net EXTCLK
  (Joined
    (PortRef &1 (InstanceRef Rmpc7))
  )
)
(Net (rename HRESET_ "HRESET*")
  (Joined
    (PortRef &7 (InstanceRef BDM-PORT))
  )
)
(Net KAPWR
  (Joined
    (PortRef &2)
    (PortRef &2)
    (PortRef &2)
    (PortRef &2)
    (PortRef &9 (InstanceRef BDM-PORT))
  )
)
(Net MDA11
  (Joined
    (PortRef &1 (InstanceRef Rmpc1))
  )
)
(Net (rename MDA11 ___ MDA31 "MDA11...MDA31")
  (Joined
  )
)
(Net MDA27
  (Joined
    (PortRef &1 (InstanceRef Rmpc2))
  )
)
(Net (rename MGPIO5 ___ MGPIO19 "MGPIO5...MGPIO19")
  (Joined
  )
)
(Net (rename MPWM0 ___ MPWM19 "MPWM0...MPWM19")
  (Joined
  )
)
(Net Net_1
  (Joined

```

```

    (PortRef &1)
    (PortRef &1 (InstanceRef Cmpc1))
    (PortRef &1 (InstanceRef Rmpc9))
  )
)
(Net Net_2
  (Joined
    (PortRef &2)
    (PortRef &1 (InstanceRef Cmpc2))
    (PortRef &2 (InstanceRef Rmpc9))
  )
)
(Net Net_Y
  (Joined
    (PortRef Y)
    (PortRef &2 (InstanceRef L?))
  )
)
(Net Net_Y
  (Joined
    (PortRef Y)
    (PortRef &2 (InstanceRef Rmpc5))
  )
)
(Net Net_Y
  (Joined
    (PortRef Y)
    (PortRef &3 (InstanceRef BDM-PORT))
    (PortRef &5 (InstanceRef BDM-PORT))
  )
)
(Net Net_Y
  (Joined
    (PortRef Y)
    (PortRef &2 (InstanceRef Rmpc8))
  )
)
(Net Net_Y
  (Joined
    (PortRef Y)
    (PortRef &2 (InstanceRef Rmpc7))
  )
)
(Net Net_Y
  (Joined
    (PortRef Y)

```

```

    (PortRef &2 (InstanceRef Cmpc1))
  )
)
(Net Net_Y
  (Joined
    (PortRef Y)
    (PortRef &2 (InstanceRef Cmpc2))
  )
)
(Net RSTCNF
  (Joined
  )
)
(Net (rename SRESET_ "SRESET*")
  (Joined
    (PortRef &2 (InstanceRef BDM-PORT))
  )
)
(Net TANT
  (Joined
    (PortRef &1 (InstanceRef Cmpc3))
    (PortRef &1 (InstanceRef Cmpc4))
    (PortRef &1 (InstanceRef L?))
  )
)
(Net VDDSYN
  (Joined
    (PortRef &1)
    (PortRef &2)
    (PortRef &2 (InstanceRef Cmpc3))
  )
)
(Net VFLS0
  (Joined
    (PortRef &1)
    (PortRef &1 (InstanceRef BDM-PORT))
  )
)
(Net VFLS1
  (Joined
    (PortRef &1)
    (PortRef &6 (InstanceRef BDM-PORT))
  )
)
(Net VPP
  (Joined

```





```

(viewType NETLIST)
(interface
  (port (rename &1 "1") (direction INOUT))
  (port (rename &2 "2") (direction INOUT))
  (port (rename &3 "3") (direction INOUT))
  (port (rename &4 "4") (direction INOUT))
  (port (rename &5 "5") (direction INOUT))
  (port (rename &6 "6") (direction INOUT))
  (port (rename &7 "7") (direction INOUT))
  (port (rename &8 "8") (direction INOUT))
)
)
)
)
(cell (rename &74HC240 "74HC240")
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INPUT))
      (port (rename &2 "2") (direction INPUT))
      (port (rename &3 "3") (direction INOUT))
      (port (rename &4 "4") (direction INPUT))
      (port (rename &5 "5") (direction INOUT))
      (port (rename &6 "6") (direction INPUT))
      (port (rename &7 "7") (direction INOUT))
      (port (rename &8 "8") (direction INPUT))
      (port (rename &9 "9") (direction INOUT))
      (port (rename &10 "10") (direction INOUT))
      (port (rename &11 "11") (direction INPUT))
      (port (rename &12 "12") (direction INOUT))
      (port (rename &13 "13") (direction INPUT))
      (port (rename &14 "14") (direction INOUT))
      (port (rename &15 "15") (direction INPUT))
      (port (rename &16 "16") (direction INOUT))
      (port (rename &17 "17") (direction INPUT))
      (port (rename &18 "18") (direction INOUT))
      (port (rename &19 "19") (direction INPUT))
      (port (rename &20 "20") (direction INOUT))
    )
  )
)
)
)
(cell (rename &74HC574 "74HC574")
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface

```

```

(port (rename &1 "1") (direction INPUT))
(port (rename &2 "2") (direction INPUT))
(port (rename &3 "3") (direction INPUT))
(port (rename &4 "4") (direction INPUT))
(port (rename &5 "5") (direction INPUT))
(port (rename &6 "6") (direction INPUT))
(port (rename &7 "7") (direction INPUT))
(port (rename &8 "8") (direction INPUT))
(port (rename &9 "9") (direction INPUT))
(port (rename &10 "10") (direction INOUT))
(port (rename &11 "11") (direction INPUT))
(port (rename &12 "12") (direction OUTPUT))
(port (rename &13 "13") (direction OUTPUT))
(port (rename &14 "14") (direction OUTPUT))
(port (rename &15 "15") (direction OUTPUT))
(port (rename &16 "16") (direction OUTPUT))
(port (rename &17 "17") (direction OUTPUT))
(port (rename &18 "18") (direction OUTPUT))
(port (rename &19 "19") (direction OUTPUT))
(port (rename &20 "20") (direction INOUT))
)
)
)
(cell AD976A
(cellType GENERIC)
(view netListView
(viewType NETLIST)
(interface
(port (rename &1 "1") (direction INOUT))
(port (rename &2 "2") (direction INOUT))
(port (rename &3 "3") (direction INOUT))
(port (rename &4 "4") (direction INOUT))
(port (rename &5 "5") (direction INOUT))
(port (rename &6 "6") (direction INOUT))
(port (rename &7 "7") (direction INOUT))
(port (rename &8 "8") (direction INOUT))
(port (rename &9 "9") (direction INOUT))
(port (rename &10 "10") (direction INOUT))
(port (rename &11 "11") (direction INOUT))
(port (rename &12 "12") (direction INOUT))
(port (rename &13 "13") (direction INOUT))
(port (rename &14 "14") (direction INOUT))
(port (rename &15 "15") (direction INOUT))
(port (rename &16 "16") (direction INOUT))
(port (rename &17 "17") (direction INOUT))
(port (rename &18 "18") (direction INOUT))
)
)
)
)

```

```

(port (rename &19 "19") (direction INOUT))
(port (rename &20 "20") (direction INOUT))
(port (rename &21 "21") (direction INOUT))
(port (rename &22 "22") (direction INOUT))
(port (rename &23 "23") (direction INOUT))
(port (rename &24 "24") (direction INOUT))
(port (rename &25 "25") (direction INOUT))
(port (rename &26 "26") (direction INOUT))
(port (rename &27 "27") (direction INOUT))
(port (rename &28 "28") (direction INOUT))
)
)
)
(cell CAP
(cellType GENERIC)
(view netListView
(viewType NETLIST)
(interface
(port (rename &1 "1") (direction INOUT))
(port (rename &2 "2") (direction INOUT))
)
)
)
(cell CAPACITOR
(cellType GENERIC)
(view netListView
(viewType NETLIST)
(interface
(port (rename &1 "1") (direction INOUT))
(port (rename &2 "2") (direction INOUT))
)
)
)
)
(cell CON3
(cellType GENERIC)
(view netListView
(viewType NETLIST)
(interface
(port (rename &1 "1") (direction INOUT))
(port (rename &2 "2") (direction INOUT))
(port (rename &3 "3") (direction INOUT))
)
)
)
)
)
(cell Cap
(cellType GENERIC)

```

```

(view netListView
  (viewType NETLIST)
  (interface
    (port (rename &1 "1") (direction INOUT))
    (port (rename &2 "2") (direction INOUT))
  )
)
)
)
(cell ELECTRO1
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INOUT))
      (port (rename &2 "2") (direction INOUT))
    )
  )
)
)
(cell (rename HEADER_17X2 "HEADER 17X2")
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INOUT))
      (port (rename &2 "2") (direction INOUT))
      (port (rename &3 "3") (direction INOUT))
      (port (rename &4 "4") (direction INOUT))
      (port (rename &5 "5") (direction INOUT))
      (port (rename &6 "6") (direction INOUT))
      (port (rename &7 "7") (direction INOUT))
      (port (rename &8 "8") (direction INOUT))
      (port (rename &9 "9") (direction INOUT))
      (port (rename &10 "10") (direction INOUT))
      (port (rename &11 "11") (direction INOUT))
      (port (rename &12 "12") (direction INOUT))
      (port (rename &13 "13") (direction INOUT))
      (port (rename &14 "14") (direction INOUT))
      (port (rename &15 "15") (direction INOUT))
      (port (rename &16 "16") (direction INOUT))
      (port (rename &17 "17") (direction INOUT))
      (port (rename &18 "18") (direction INOUT))
      (port (rename &19 "19") (direction INOUT))
      (port (rename &20 "20") (direction INOUT))
      (port (rename &21 "21") (direction INOUT))
      (port (rename &22 "22") (direction INOUT))
      (port (rename &23 "23") (direction INOUT))
    )
  )
)

```

```

(port (rename &24 "24") (direction INOUT))
(port (rename &25 "25") (direction INOUT))
(port (rename &26 "26") (direction INOUT))
(port (rename &27 "27") (direction INOUT))
(port (rename &28 "28") (direction INOUT))
(port (rename &29 "29") (direction INOUT))
(port (rename &30 "30") (direction INOUT))
(port (rename &31 "31") (direction INOUT))
(port (rename &32 "32") (direction INOUT))
(port (rename &33 "33") (direction INOUT))
(port (rename &34 "34") (direction INOUT))
)
)
)
(cell MC1439
(cellType GENERIC)
(view netListView
(viewType NETLIST)
(interface
(port (rename &1 "1") (direction INOUT))
(port (rename &2 "2") (direction INOUT))
(port (rename &3 "3") (direction INOUT))
(port (rename &4 "4") (direction INOUT))
(port (rename &5 "5") (direction INOUT))
(port (rename &6 "6") (direction INOUT))
(port (rename &7 "7") (direction INOUT))
(port (rename &8 "8") (direction INOUT))
)
)
)
)
(cell RES1
(cellType GENERIC)
(view netListView
(viewType NETLIST)
(interface
(port (rename &1 "1") (direction INOUT))
(port (rename &2 "2") (direction INOUT))
)
)
)
)
)
(cell Res1
(cellType GENERIC)
(view netListView
(viewType NETLIST)
(interface
(port (rename &1 "1") (direction INOUT))

```

```

        (port (rename &2 "2") (direction INOUT))
    )
)
)
)
(library SHEET_LIB
(edifLevel 0)
(technology
(numberDefinition
(scale 1 1 (unit distance))
)
)
)
(cell ADCnetlist_SCHDOC
(cellType generic)
(view netListView
(viewType netlist)
(interface
)
)
(contents
(Instance C1
(viewRef NetlistView
(cellRef CAP
(LibraryRef COMPONENT_LIB)
)
)
)
(Property Comment (String "2nF" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "RAD0.1" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "CAP" ))
(Property Footprint (String "RAD0.1" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\VBDOIS200392679671599276924QKDVERY" ))
))
)
(Instance C2
(viewRef NetlistView
(cellRef CAP
(LibraryRef COMPONENT_LIB)
)
)
)
)
(Property Comment (String "0.1uF" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "RAD0.1" ))
)

```

```

(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "CAP" ))
(Property Footprint (String "RAD0.1" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$\7902D3200392751720655237591@8T7EU" ))
)
(Instance C3
(viewRef NetlistView
(cellRef CAP
(LibraryRef COMPONENT_LIB)
)
)
(Property Comment (String "0.1uF" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "RAD0.1" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "CAP" ))
(Property Footprint (String "RAD0.1" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$\I$VA2B200392751720655237591HHRA9X" ))
)
(Instance C4
(viewRef NetlistView
(cellRef CAP
(LibraryRef COMPONENT_LIB)
)
)
(Property Comment (String "0.1uF" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "RAD0.1" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "CAP" ))
(Property Footprint (String "RAD0.1" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$\HC9A1G200392751720655237591ET8JD6" ))
)
(Instance C5
(viewRef NetlistView
(cellRef CAP
(LibraryRef COMPONENT_LIB)
)
)
(Property Comment (String "0.1uF" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))

```



```

(Property Description (String "Capacitor" ))
(Property Footprint (String "RAD0.1" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "CAP" ))
(Property Footprint (String "RAD0.1" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\OVX9PG200392751720655237591LM#MW3"
))
)
(Instance C6
(viewRef NetlistView
(cellRef CAP
(LibraryRef COMPONENT_LIB)
)
)
(Property Comment (String "0.1uF" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "RAD0.1" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "CAP" ))
(Property Footprint (String "RAD0.1" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\U7NA#J200392751720655237591#C81MF" ))
)
(Instance C7
(viewRef NetlistView
(cellRef CAP
(LibraryRef COMPONENT_LIB)
)
)
(Property Comment (String "0.1uF" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "RAD0.1" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "CAP" ))
(Property Footprint (String "RAD0.1" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\SEHX7U200392751720655237591DSFO02" ))
)
(Instance C9
(viewRef NetlistView
(cellRef CAP
(LibraryRef COMPONENT_LIB)
)
)

```

```

)
(Property Comment (String "CAP" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "1206" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "CAP" ))
(Property Footprint (String "1206" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\5OI34W2003101854237347965430X@J" ))
)
(Instance C10
(viewRef NetlistView
(cellRef CAP
(LibraryRef COMPONENT_LIB)
)
)
(Property Comment (String "0.1uF" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "RAD0.1" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "CAP" ))
(Property Footprint (String "RAD0.1" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\0HY$NF200392751720655237591NK9A#0" ))
)
(Instance C11
(viewRef NetlistView
(cellRef CAP
(LibraryRef COMPONENT_LIB)
)
)
(Property Comment (String "0.1uF" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "RAD0.1" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "CAP" ))
(Property Footprint (String "RAD0.1" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\OKCJMS200392751720655237591QHXPYP"
))
)
(Instance C12
(viewRef NetlistView

```

```

    (cellRef CAP
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property Comment (String "0.1uF" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "Capacitor" ))
  (Property Footprint (String "RAD0.1" ))
  (Property (rename Library_Name "Library Name") (String "" ))
  (Property (rename Library_Reference "Library Reference") (String "CAP" ))
  (Property Footprint (String "RAD0.1" ))
  (Property Description (String "Capacitor" ))
  (Property UniqueId (String "$$$\LXZ6FQ200392751720655237591R02NO8" ))
)
(Instance C13
  (viewRef NetlistView
    (cellRef ELECTRO1
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property Comment (String "10uF" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "Electrolytic Capacitor" ))
  (Property Footprint (String "CRB-.2/.4" ))
  (Property (rename Library_Name "Library Name") (String "" ))
  (Property (rename Library_Reference "Library Reference") (String "ELECTRO1"
))
  (Property Footprint (String "CRB-.2/.4" ))
  (Property Description (String "Electrolytic Capacitor" ))
  (Property UniqueId (String "$$$\9XWI20200392751720655237591S%ZFID" ))
)
(Instance C14
  (viewRef NetlistView
    (cellRef ELECTRO1
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property Comment (String "10uF" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "Electrolytic Capacitor" ))
  (Property Footprint (String "CRB-.2/.4" ))
  (Property (rename Library_Name "Library Name") (String "" ))
  (Property (rename Library_Reference "Library Reference") (String "ELECTRO1"
))
  (Property Footprint (String "CRB-.2/.4" ))
  (Property Description (String "Electrolytic Capacitor" ))

```

```

    (Property UniqueId (String "$$$\BBX@NC200392751720655237591QQ5AFJ"
))
)
(Instance C15
  (viewRef NetlistView
    (cellRef ELECTRO1
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property Comment (String "10uF" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "Electrolytic Capacitor" ))
  (Property Footprint (String "CRB-.2/.4" ))
  (Property (rename Library_Name "Library Name") (String "" ))
  (Property (rename Library_Reference "Library Reference") (String "ELECTRO1"
))
  (Property Footprint (String "CRB-.2/.4" ))
  (Property Description (String "Electrolytic Capacitor" ))
  (Property UniqueId (String "$$$\T609@Z200392751720655237591YO3M7J" ))
)
(Instance C16
  (viewRef NetlistView
    (cellRef CAP
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property Comment (String "CAP" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "Capacitor" ))
  (Property Footprint (String "1206" ))
  (Property (rename Library_Name "Library Name") (String "" ))
  (Property (rename Library_Reference "Library Reference") (String "CAP" ))
  (Property Footprint (String "1206" ))
  (Property Description (String "Capacitor" ))
  (Property UniqueId (String "$$$\3FZ$$Q2003101854237347965R@@@TV1" ))
)
(Instance C17
  (viewRef NetlistView
    (cellRef ELECTRO1
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property Comment (String "10uF" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "Electrolytic Capacitor" ))
  (Property Footprint (String "CRB-.2/.4" ))

```

```

(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "ELECTRO1"
))
(Property Footprint (String "CRB-.2/.4" ))
(Property Description (String "Electrolytic Capacitor" ))
(Property UniqueId (String "$$$\ANZND4200392751720655237591LR@40%"
))
)
(Instance C18
(viewRef NetlistView
(cellRef ELECTRO1
(LibraryRef COMPONENT_LIB)
)
)
(Property Comment (String "10uF" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Electrolytic Capacitor" ))
(Property Footprint (String "CRB-.2/.4" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "ELECTRO1"
))
(Property Footprint (String "CRB-.2/.4" ))
(Property Description (String "Electrolytic Capacitor" ))
(Property UniqueId (String "$$$\4T8TDU200392751720655237591ALM3CO" ))
)
(Instance C19
(viewRef NetlistView
(cellRef CAPACITOR
(LibraryRef COMPONENT_LIB)
)
)
(Property Comment (String "2.2uF" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "CRB-.2/.4" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String
"CAPACITOR" ))
(Property Footprint (String "CRB-.2/.4" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\1U11DF200392751720655237591BMB5DU" ))
)
(Instance C20
(viewRef NetlistView
(cellRef CAPACITOR
(LibraryRef COMPONENT_LIB)

```

```

)
)
(Property Comment (String "2.2uF" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "CRB-.2/.4" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String
"CAPACITOR" ))
(Property Footprint (String "CRB-.2/.4" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\U4WO8R200392751720655237591L6AEJ1" ))
)
(Instance C21
(viewRef NetlistView
(cellRef CAP
(LibraryRef COMPONENT_LIB)
)
)
)
(Property Comment (String "CAP" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "1206" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "CAP" ))
(Property Footprint (String "1206" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\PZQ0XE20031018542373479650Q8TXX" ))
)
(Instance C22
(viewRef NetlistView
(cellRef CAP
(LibraryRef COMPONENT_LIB)
)
)
)
(Property Comment (String "CAP" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "1206" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "CAP" ))
(Property Footprint (String "1206" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\02URE@2003101854237347965MGFFU$" ))
)
(Instance C23

```

```

(viewRef NetlistView
  (cellRef CAP
    (LibraryRef COMPONENT_LIB)
  )
)
(Property Comment (String "CAP" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "1206" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "CAP" ))
(Property Footprint (String "1206" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\SPR26V2003101854237347965ICEG5I" ))
)
(Instance C24
  (viewRef NetlistView
    (cellRef CAP
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property Comment (String "CAP" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "Capacitor" ))
  (Property Footprint (String "1206" ))
  (Property (rename Library_Name "Library Name") (String "" ))
  (Property (rename Library_Reference "Library Reference") (String "CAP" ))
  (Property Footprint (String "1206" ))
  (Property Description (String "Capacitor" ))
  (Property UniqueId (String "$$$\DM81YN200310185423734796582$FYE" ))
)
(Instance Csg1
  (viewRef NetlistView
    (cellRef Cap
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property (rename Class_I "Class I") (String "Passive" ))
  (Property (rename Class_II "Class II") (String "Capacitor" ))
  (Property Comment (String "Csg1" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "Capacitor" ))
  (Property Footprint (String "RAD-0.3" ))
  (Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
  (Property (rename Library_Reference "Library Reference") (String "Cap" ))
)

```

```

(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Cap" ))
(Property Simulation (String "CAP" ))
(Property Value (String "470uF" ))
(Property Footprint (String "RAD-0.3" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\BJQVWSFS" ))
)
(Instance Csig2
  (viewRef NetlistView
    (cellRef CAP
      (LibraryRef COMPONENT_LIB)
    )
  )
)
(Property Comment (String "0.1uF" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "RAD0.1" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "CAP" ))
(Property Footprint (String "RAD0.1" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\WXVJNPXL" ))
)
(Instance J1
  (viewRef NetlistView
    (cellRef CON3
      (LibraryRef COMPONENT_LIB)
    )
  )
)
(Property Comment (String "CON3" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Connector" ))
(Property Footprint (String "SIP-3" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "CON3" ))
(Property Footprint (String "SIP-3" ))
(Property Description (String "Connector" ))
(Property UniqueId (String "$$$\1XC8692003927517206552375917ZMOZE" ))
)
(Instance JP1
  (viewRef NetlistView
    (cellRef HEADER_17X2

```



```

    (LibraryRef COMPONENT_LIB)
  )
)
(Property Comment (String "HEADER 17X2" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "" ))
(Property Footprint (String "IDC-34" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "HEADER
17X2" ))
(Property Footprint (String "IDC-34" ))
(Property UniqueId (String "$$\EWN1B5200392679671599387393LU50PT" ))
)
(Instance JP2
  (viewRef NetlistView
    (cellRef &8_HEADER
      (LibraryRef COMPONENT_LIB)
    )
  )
)
(Property Comment (String "8 HEADER" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "8 Pin Header" ))
(Property Footprint (String "MODU8IV" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "8 HEADER"
))
(Property Footprint (String "MODU8IV" ))
(Property Description (String "8 Pin Header" ))
(Property UniqueId (String "$$\JHU2ZK2003926796715993873939E%29F" ))
)
(Instance R1
  (viewRef NetlistView
    (cellRef RES1
      (LibraryRef COMPONENT_LIB)
    )
  )
)
(Property Comment (String "1k" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "" ))
(Property Footprint (String "AXIAL-0.4" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "RES1" ))
(Property Footprint (String "AXIAL-0.4" ))
(Property UniqueId (String "$$\OGE7#C200392679671599387393TR48UQ" ))
)
(Instance R2

```

```

(viewRef NetlistView
  (cellRef RES1
    (LibraryRef COMPONENT_LIB)
  )
)
(Property Comment (String "500" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "" ))
(Property Footprint (String "AXIAL-0.4" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "RES1" ))
(Property Footprint (String "AXIAL-0.4" ))
(Property UniqueId (String "$$$\ZL6$T200392679671599387393AOAZ86" ))
)
(Instance R3
  (viewRef NetlistView
    (cellRef RES1
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property Comment (String "1k" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "" ))
  (Property Footprint (String "AXIAL-0.4" ))
  (Property (rename Library_Name "Library Name") (String "" ))
  (Property (rename Library_Reference "Library Reference") (String "RES1" ))
  (Property Footprint (String "AXIAL-0.4" ))
  (Property UniqueId (String "$$$\4T%KW@200392679671599276924TE$#@5"
))
)
(Instance R4
  (viewRef NetlistView
    (cellRef RES1
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property Comment (String "66.5k" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "" ))
  (Property Footprint (String "AXIAL-0.4" ))
  (Property (rename Library_Name "Library Name") (String "" ))
  (Property (rename Library_Reference "Library Reference") (String "RES1" ))
  (Property Footprint (String "AXIAL-0.4" ))
  (Property UniqueId (String "$$$\8C50VF200392751720655237591Z3DUSD" ))
)
(Instance R5

```

```

(viewRef NetlistView
  (cellRef RES1
    (LibraryRef COMPONENT_LIB)
  )
)
(Property Comment (String "200" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "" ))
(Property Footprint (String "AXIAL-0.4" ))
(Property (rename Library_Name "Library Name") (String "" ))
(Property (rename Library_Reference "Library Reference") (String "RES1" ))
(Property Footprint (String "AXIAL-0.4" ))
(Property UniqueId (String "$$$K2G$@Y200392751720655237591325Z1G" ))
)
(Instance Rsig2
  (viewRef NetlistView
    (cellRef Res1
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property (rename Class_I "Class I") (String "Passive" ))
  (Property (rename Class_II "Class II") (String "Resistor" ))
  (Property Comment (String "" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "Resistor" ))
  (Property Footprint (String "AXIAL-0.3" ))
  (Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
  (Property (rename Library_Reference "Library Reference") (String "Res1" ))
  (Property Manufacturer (String "Generic Components" ))
  (Property Published (String "8-Jun-2000" ))
  (Property Publisher (String "Altium Limited" ))
  (Property Revision (String "July-2002: Re-released for DXP Platform." ))
  (Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
  (Property Simulation (String "RESISTOR" ))
  (Property Value (String "1M" ))
  (Property Footprint (String "AXIAL-0.3" ))
  (Property Description (String "Resistor" ))
  (Property UniqueId (String "$$$RBKKVYTC" ))
)
(Instance Rsig3
  (viewRef NetlistView
    (cellRef Res1
      (LibraryRef COMPONENT_LIB)
    )
  )
)

```

```

(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Resistor" ))
(Property Comment (String "" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Resistor" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Res1" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property Simulation (String "RESISTOR" ))
(Property Value (String "1.33k" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property Description (String "Resistor" ))
(Property UniqueId (String "$$$\KJQLJTMH" ))
)
(Instance Rsig4
(viewRef NetlistView
(cellRef Res1
(LibraryRef COMPONENT_LIB)
)
)
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Resistor" ))
(Property Comment (String "" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Resistor" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Res1" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property Simulation (String "RESISTOR" ))
(Property Value (String "2k" ))
(Property Footprint (String "AXIAL-0.3" ))
(Property Description (String "Resistor" ))
(Property UniqueId (String "$$$\PYALVSTX" ))
)
)

```

```

(Instance U1
  (viewRef NetlistView
    (cellRef MC1439
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property Comment (String "AD845" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "" ))
  (Property Footprint (String "DIP-8" ))
  (Property (rename Library_Name "Library Name") (String "" ))
  (Property (rename Library_Reference "Library Reference") (String "MC1439" ))
  (Property Footprint (String "DIP-8" ))
  (Property UniqueId (String "$$\YCXOX02003926796715992769243NX2IR" ))
)
(Instance U2
  (viewRef NetlistView
    (cellRef &74HC574
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property Comment (String "74HC574" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "" ))
  (Property Footprint (String "DIP-20" ))
  (Property (rename Library_Name "Library Name") (String "" ))
  (Property (rename Library_Reference "Library Reference") (String "74HC574" ))
  (Property Footprint (String "DIP-20" ))
  (Property UniqueId (String "$$\7N$X%C200392751720655237591KJW%YQ"
))
)
(Instance U3
  (viewRef NetlistView
    (cellRef &74HC574
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property Comment (String "74HC574" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "" ))
  (Property Footprint (String "DIP-20" ))
  (Property (rename Library_Name "Library Name") (String "" ))
  (Property (rename Library_Reference "Library Reference") (String "74HC574" ))
  (Property Footprint (String "DIP-20" ))
  (Property UniqueId (String "$$\E@OIU72003927517206552375913US7X7" ))
)

```

```

(Instance U5
  (viewRef NetlistView
    (cellRef AD976A
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property Comment (String "AD976A" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "" ))
  (Property Footprint (String "CDIP28" ))
  (Property (rename Library_Name "Library Name") (String "" ))
  (Property (rename Library_Reference "Library Reference") (String "AD976A" ))
  (Property Footprint (String "CDIP28" ))
  (Property UniqueId (String "$$$\AWV1SZ200392751720655237591G0OON5"
))
)
(Instance U6
  (viewRef NetlistView
    (cellRef &74HC240
      (LibraryRef COMPONENT_LIB)
    )
  )
  (Property Comment (String "74HC240" ))
  (Property (rename Component_Kind "Component Kind") (String "Standard" ))
  (Property Description (String "" ))
  (Property Footprint (String "DIP-20" ))
  (Property (rename Library_Name "Library Name") (String "" ))
  (Property (rename Library_Reference "Library Reference") (String "74HC240" ))
  (Property Footprint (String "DIP-20" ))
  (Property UniqueId (String "$$$\81SMAN200392751720655237591BU4GX9" ))
)
(Net (rename PLUS12 "+12")
  (Joined
    (PortRef &1 (InstanceRef C6))
    (PortRef &1 (InstanceRef C14))
    (PortRef &5 (InstanceRef JP2))
    (PortRef &7 (InstanceRef U1))
  )
)
(Net (rename MINUS12 "-12")
  (Joined
    (PortRef &2 (InstanceRef C7))
    (PortRef &2 (InstanceRef C15))
    (PortRef &6 (InstanceRef JP2))
    (PortRef &4 (InstanceRef U1))
  )
)

```

```

)
(Net AGND
  (Joined
    (PortRef &2 (InstanceRef C6))
    (PortRef &1 (InstanceRef C7))
    (PortRef &2 (InstanceRef C10))
    (PortRef &2 (InstanceRef C11))
    (PortRef &2 (InstanceRef C14))
    (PortRef &1 (InstanceRef C15))
    (PortRef &2 (InstanceRef C17))
    (PortRef &2 (InstanceRef C19))
    (PortRef &2 (InstanceRef C20))
    (PortRef &1 (InstanceRef Csg1))
    (PortRef &2 (InstanceRef Csig2))
    (PortRef &2 (InstanceRef R2))
    (PortRef &1 (InstanceRef Rsig2))
    (PortRef &2 (InstanceRef U5))
    (PortRef &5 (InstanceRef U5))
  )
)
)
(Net BUSY
  (Joined
    (PortRef &11 (InstanceRef U2))
    (PortRef &11 (InstanceRef U3))
    (PortRef &26 (InstanceRef U5))
    (PortRef &2 (InstanceRef U6))
  )
)
)
(Net D0
  (Joined
    (PortRef &2 (InstanceRef U2))
    (PortRef &22 (InstanceRef U5))
  )
)
)
(Net D1
  (Joined
    (PortRef &3 (InstanceRef U2))
    (PortRef &21 (InstanceRef U5))
  )
)
)
(Net D2
  (Joined
    (PortRef &4 (InstanceRef U2))
    (PortRef &20 (InstanceRef U5))
  )
)
)

```

```
(Net D3
  (Joined
    (PortRef &5 (InstanceRef U2))
    (PortRef &19 (InstanceRef U5))
  )
)
(Net D4
  (Joined
    (PortRef &6 (InstanceRef U2))
    (PortRef &18 (InstanceRef U5))
  )
)
(Net D5
  (Joined
    (PortRef &7 (InstanceRef U2))
    (PortRef &17 (InstanceRef U5))
  )
)
(Net D6
  (Joined
    (PortRef &8 (InstanceRef U2))
    (PortRef &16 (InstanceRef U5))
  )
)
(Net D7
  (Joined
    (PortRef &9 (InstanceRef U2))
    (PortRef &15 (InstanceRef U5))
  )
)
(Net D8
  (Joined
    (PortRef &2 (InstanceRef U3))
    (PortRef &13 (InstanceRef U5))
  )
)
(Net D9
  (Joined
    (PortRef &3 (InstanceRef U3))
    (PortRef &12 (InstanceRef U5))
  )
)
(Net D10
  (Joined
    (PortRef &4 (InstanceRef U3))
    (PortRef &11 (InstanceRef U5))
  )
)
```



```

)
)
(Net D11
  (Joined
    (PortRef &5 (InstanceRef U3))
    (PortRef &10 (InstanceRef U5))
  )
)
(Net D12
  (Joined
    (PortRef &6 (InstanceRef U3))
    (PortRef &9 (InstanceRef U5))
  )
)
(Net D13
  (Joined
    (PortRef &7 (InstanceRef U3))
    (PortRef &8 (InstanceRef U5))
  )
)
(Net D14
  (Joined
    (PortRef &8 (InstanceRef U3))
    (PortRef &7 (InstanceRef U5))
  )
)
(Net D15
  (Joined
    (PortRef &9 (InstanceRef U3))
    (PortRef &6 (InstanceRef U5))
  )
)
(Net GND
  (Joined
    (PortRef &2 (InstanceRef C2))
    (PortRef &2 (InstanceRef C3))
    (PortRef &2 (InstanceRef C4))
    (PortRef &2 (InstanceRef C5))
    (PortRef &2 (InstanceRef C9))
    (PortRef &2 (InstanceRef C12))
    (PortRef &2 (InstanceRef C13))
    (PortRef &2 (InstanceRef C16))
    (PortRef &2 (InstanceRef C18))
    (PortRef &2 (InstanceRef C21))
    (PortRef &2 (InstanceRef C22))
    (PortRef &2 (InstanceRef C23))
  )
)

```

```

(PortRef &2 (InstanceRef C24))
(PortRef &2 (InstanceRef JP2))
(PortRef &4 (InstanceRef JP2))
(PortRef &1 (InstanceRef U2))
(PortRef &10 (InstanceRef U2))
(PortRef &1 (InstanceRef U3))
(PortRef &10 (InstanceRef U3))
(PortRef &14 (InstanceRef U5))
(PortRef &23 (InstanceRef U5))
(PortRef &25 (InstanceRef U5))
(PortRef &1 (InstanceRef U6))
(PortRef &10 (InstanceRef U6))
(PortRef &19 (InstanceRef U6))
)
)
(Net IRQ2
  (Joined
    (PortRef &1 (InstanceRef JP2))
    (PortRef &9 (InstanceRef U6))
  )
)
(Net NetC1_1
  (Joined
    (PortRef &1 (InstanceRef C1))
    (PortRef &1 (InstanceRef R1))
    (PortRef &2 (InstanceRef R3))
    (PortRef &2 (InstanceRef U1))
  )
)
(Net NetC19_1
  (Joined
    (PortRef &1 (InstanceRef C19))
    (PortRef &3 (InstanceRef U5))
  )
)
(Net NetC20_1
  (Joined
    (PortRef &1 (InstanceRef C20))
    (PortRef &4 (InstanceRef U5))
  )
)
(Net NetCsg1_2
  (Joined
    (PortRef &2 (InstanceRef Csg1))
    (PortRef &2 (InstanceRef Rsig2))
    (PortRef &1 (InstanceRef Rsig3))
  )
)

```

```

    (PortRef &1 (InstanceRef Rsig4))
  )
)
(Net NetR2_1
  (Joined
    (PortRef &1 (InstanceRef R2))
    (PortRef &3 (InstanceRef U1))
  )
)
(Net NetR4_1
  (Joined
    (PortRef &1 (InstanceRef R4))
    (PortRef &1 (InstanceRef R5))
    (PortRef &1 (InstanceRef U5))
  )
)
(Net NetU6_4
  (Joined
    (PortRef &4 (InstanceRef U6))
    (PortRef &18 (InstanceRef U6))
  )
)
(Net NetU6_6
  (Joined
    (PortRef &6 (InstanceRef U6))
    (PortRef &16 (InstanceRef U6))
  )
)
(Net NetU6_8
  (Joined
    (PortRef &8 (InstanceRef U6))
    (PortRef &14 (InstanceRef U6))
  )
)
(Net NetU6_11
  (Joined
    (PortRef &11 (InstanceRef U6))
    (PortRef &12 (InstanceRef U6))
  )
)
(Net OPVIN
  (Joined
    (PortRef &2 (InstanceRef C1))
    (PortRef &1 (InstanceRef J1))
    (PortRef &1 (InstanceRef R3))
    (PortRef &6 (InstanceRef U1))
  )
)

```

```
)
)
(Net PPS
  (Joined
    (PortRef &5 (InstanceRef JP1))
    (PortRef &8 (InstanceRef JP2))
  )
)
(Net Q0
  (Joined
    (PortRef &19 (InstanceRef JP1))
    (PortRef &19 (InstanceRef U2))
  )
)
(Net Q1
  (Joined
    (PortRef &20 (InstanceRef JP1))
    (PortRef &18 (InstanceRef U2))
  )
)
(Net Q2
  (Joined
    (PortRef &21 (InstanceRef JP1))
    (PortRef &17 (InstanceRef U2))
  )
)
(Net Q3
  (Joined
    (PortRef &22 (InstanceRef JP1))
    (PortRef &16 (InstanceRef U2))
  )
)
(Net Q4
  (Joined
    (PortRef &23 (InstanceRef JP1))
    (PortRef &15 (InstanceRef U2))
  )
)
(Net Q5
  (Joined
    (PortRef &24 (InstanceRef JP1))
    (PortRef &14 (InstanceRef U2))
  )
)
(Net Q6
  (Joined
```

```
(PortRef &25 (InstanceRef JP1))
(PortRef &13 (InstanceRef U2))
)
)
(Net Q7
(Joined
(PortRef &26 (InstanceRef JP1))
(PortRef &12 (InstanceRef U2))
)
)
(Net Q8
(Joined
(PortRef &27 (InstanceRef JP1))
(PortRef &19 (InstanceRef U3))
)
)
(Net Q9
(Joined
(PortRef &28 (InstanceRef JP1))
(PortRef &18 (InstanceRef U3))
)
)
(Net Q10
(Joined
(PortRef &29 (InstanceRef JP1))
(PortRef &17 (InstanceRef U3))
)
)
(Net Q11
(Joined
(PortRef &30 (InstanceRef JP1))
(PortRef &16 (InstanceRef U3))
)
)
(Net Q12
(Joined
(PortRef &31 (InstanceRef JP1))
(PortRef &15 (InstanceRef U3))
)
)
(Net Q13
(Joined
(PortRef &32 (InstanceRef JP1))
(PortRef &14 (InstanceRef U3))
)
)
```

```

(Net Q14
  (Joined
    (PortRef &33 (InstanceRef JP1))
    (PortRef &13 (InstanceRef U3))
  )
)
(Net Q15
  (Joined
    (PortRef &34 (InstanceRef JP1))
    (PortRef &12 (InstanceRef U3))
  )
)
(Net RC
  (Joined
    (PortRef &11 (InstanceRef JP1))
    (PortRef &24 (InstanceRef U5))
  )
)
(Net VANA
  (Joined
    (PortRef &1 (InstanceRef C10))
    (PortRef &1 (InstanceRef C17))
    (PortRef &2 (InstanceRef R4))
    (PortRef &27 (InstanceRef U5))
  )
)
(Net VCC
  (Joined
    (PortRef &1 (InstanceRef C2))
    (PortRef &1 (InstanceRef C3))
    (PortRef &1 (InstanceRef C4))
    (PortRef &1 (InstanceRef C5))
    (PortRef &1 (InstanceRef C9))
    (PortRef &1 (InstanceRef C12))
    (PortRef &1 (InstanceRef C13))
    (PortRef &1 (InstanceRef C16))
    (PortRef &1 (InstanceRef C18))
    (PortRef &1 (InstanceRef C21))
    (PortRef &1 (InstanceRef C22))
    (PortRef &1 (InstanceRef C23))
    (PortRef &1 (InstanceRef C24))
    (PortRef &3 (InstanceRef JP2))
    (PortRef &20 (InstanceRef U2))
    (PortRef &20 (InstanceRef U3))
    (PortRef &28 (InstanceRef U5))
    (PortRef &20 (InstanceRef U6))
  )
)

```

```

    )
  )
  (Net VIN
    (Joined
      (PortRef &3 (InstanceRef J1))
      (PortRef &7 (InstanceRef JP2))
      (PortRef &2 (InstanceRef R1))
      (PortRef &2 (InstanceRef Rsig3))
      (PortRef &2 (InstanceRef Rsig4))
    )
  )
  )
  (Net VSAM
    (Joined
      (PortRef &1 (InstanceRef C11))
      (PortRef &2 (InstanceRef J1))
      (PortRef &2 (InstanceRef R5))
    )
  )
  )
  )
  )
  )
  )
  )
  (design Design_ADCnetlist_SCHDOC
    (cellRef ADCnetlist_SCHDOC
      (libraryRef SHEET_LIB)
    )
  )
  )
  )
)

```

### **3. Netlist for LCD and GPS connection**

```

(edif Design_Sheet3new_SCHDOC
  (edifVersion 2 0 0)
  (edifLevel 0)
  (keywordMap
    (keywordLevel 0)
  )
  (status
    (written
      (timeStamp 2006 1 12 0 20 45)
      (program "Design Explorer DXP - EDIF For PCB"
        (version "1.0.0")
      )
    )
    (author "EDIF For PCB")
  )
)
)

```

```

(library COMPONENT_LIB
  (edifLevel 0)
  (technology
    (numberDefinition
      (scale 1 1 (unit distance))
    )
  )
)
(cell (rename ATF22LV10CZ_25PI "ATF22LV10CZ-25PI")
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INPUT))
      (port (rename &2 "2") (direction INPUT))
      (port (rename &3 "3") (direction INPUT))
      (port (rename &4 "4") (direction INPUT))
      (port (rename &5 "5") (direction INPUT))
      (port (rename &6 "6") (direction INPUT))
      (port (rename &7 "7") (direction INPUT))
      (port (rename &8 "8") (direction INPUT))
      (port (rename &9 "9") (direction INPUT))
      (port (rename &10 "10") (direction INPUT))
      (port (rename &11 "11") (direction INPUT))
      (port (rename &12 "12") (direction INOUT))
      (port (rename &13 "13") (direction INPUT))
      (port (rename &14 "14") (direction INOUT))
      (port (rename &15 "15") (direction INOUT))
      (port (rename &16 "16") (direction INOUT))
      (port (rename &17 "17") (direction INOUT))
      (port (rename &18 "18") (direction INOUT))
      (port (rename &19 "19") (direction INOUT))
      (port (rename &20 "20") (direction INOUT))
      (port (rename &21 "21") (direction INOUT))
      (port (rename &22 "22") (direction INOUT))
      (port (rename &23 "23") (direction INOUT))
      (port (rename &24 "24") (direction INOUT))
    )
  )
)
(cell Cap
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INOUT))
      (port (rename &2 "2") (direction INOUT))
    )
  )
)

```



```

)
)
)
(cell Cap2
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INOUT))
      (port (rename &2 "2") (direction INOUT))
    )
  )
)
)
)
(cell Component_1
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INOUT))
      (port (rename &2 "2") (direction INOUT))
      (port (rename &3 "3") (direction INOUT))
      (port (rename &4 "4") (direction INOUT))
      (port (rename &5 "5") (direction INOUT))
      (port (rename &6 "6") (direction INOUT))
      (port (rename &7 "7") (direction INOUT))
      (port (rename &8 "8") (direction INOUT))
    )
  )
)
)
)
)
(cell GND
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port Y (direction OUTPUT))
    )
  )
)
)
)
)
(cell (rename Header_5X2H "Header 5X2H")
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INOUT))
      (port (rename &2 "2") (direction INOUT))
      (port (rename &3 "3") (direction INOUT))
    )
  )
)

```

```

        (port (rename &4 "4") (direction INOUT))
        (port (rename &5 "5") (direction INOUT))
        (port (rename &6 "6") (direction INOUT))
        (port (rename &7 "7") (direction INOUT))
        (port (rename &8 "8") (direction INOUT))
        (port (rename &9 "9") (direction INOUT))
        (port (rename &10 "10") (direction INOUT))
    )
)
)
)
(cell (rename Header_7X2 "Header 7X2")
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INOUT))
      (port (rename &2 "2") (direction INOUT))
      (port (rename &3 "3") (direction INOUT))
      (port (rename &4 "4") (direction INOUT))
      (port (rename &5 "5") (direction INOUT))
      (port (rename &6 "6") (direction INOUT))
      (port (rename &7 "7") (direction INOUT))
      (port (rename &8 "8") (direction INOUT))
      (port (rename &9 "9") (direction INOUT))
      (port (rename &10 "10") (direction INOUT))
      (port (rename &11 "11") (direction INOUT))
      (port (rename &12 "12") (direction INOUT))
      (port (rename &13 "13") (direction INOUT))
      (port (rename &14 "14") (direction INOUT))
    )
  )
)
)
(cell (rename Header_12X2A "Header 12X2A")
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INOUT))
      (port (rename &2 "2") (direction INOUT))
      (port (rename &3 "3") (direction INOUT))
      (port (rename &4 "4") (direction INOUT))
      (port (rename &5 "5") (direction INOUT))
      (port (rename &6 "6") (direction INOUT))
      (port (rename &7 "7") (direction INOUT))
      (port (rename &8 "8") (direction INOUT))
      (port (rename &9 "9") (direction INOUT))
    )
  )
)

```

```

    (port (rename &10 "10") (direction INOUT))
    (port (rename &11 "11") (direction INOUT))
    (port (rename &12 "12") (direction INOUT))
    (port (rename &13 "13") (direction INOUT))
    (port (rename &14 "14") (direction INOUT))
    (port (rename &15 "15") (direction INOUT))
    (port (rename &16 "16") (direction INOUT))
    (port (rename &17 "17") (direction INOUT))
    (port (rename &18 "18") (direction INOUT))
    (port (rename &19 "19") (direction INOUT))
    (port (rename &20 "20") (direction INOUT))
    (port (rename &21 "21") (direction INOUT))
    (port (rename &22 "22") (direction INOUT))
    (port (rename &23 "23") (direction INOUT))
    (port (rename &24 "24") (direction INOUT))
  )
)
)
(cell RPot1
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INOUT))
      (port (rename &2 "2") (direction INOUT))
      (port (rename &3 "3") (direction INOUT))
    )
  )
)
)
)
(cell Res1
  (cellType GENERIC)
  (view netListView
    (viewType NETLIST)
    (interface
      (port (rename &1 "1") (direction INOUT))
      (port (rename &2 "2") (direction INOUT))
    )
  )
)
)
)
(library SHEET_LIB
  (edifLevel 0)
  (technology
    (numberDefinition
      (scale 1 1 (unit distance))
    )
  )
)

```

```

)
(cell Sheet3new_SCHDOC
  (cellType generic)
  (view netListView
    (viewType netlist)
    (interface
      )
    (contents
      (Instance (rename UNDEFINED ""))
      (viewRef NetlistView
        (cellRef GND
          (LibraryRef COMPONENT_LIB)
        )
      )
    )
    (Property Class (String "General" ))
    (Property Class (String "General" ))
    (Property Class (String "General" ))
    (Property Class (String "General" ))
    (Property Class (String "General" ))
    (Property Class (String "General" ))
    (Property Class (String "General" ))
    (Property Class (String "General" ))
    (Property (rename Class_I "Class I") (String "Discrete" ))
    (Property (rename Class_II "Class II") (String "Diode - Switching" ))
    (Property Comment (String "" ))
    (Property Comment (String "" ))
    (Property Comment (String "1N4148" ))
    (Property Comment (String "" ))
    (Property Comment (String "" ))
    (Property Comment (String "" ))
    (Property Comment (String "" ))
    (Property Comment (String "" ))
    (Property Comment (String "" ))
    (Property Comment (String "" ))
    (Property (rename Component_Kind "Component Kind") (String "Standard" ))
    (Property (rename Component_Kind "Component Kind") (String "Standard" ))
    (Property (rename Component_Kind "Component Kind") (String "Standard" ))
    (Property (rename Component_Kind "Component Kind") (String "Standard" ))
    (Property (rename Component_Kind "Component Kind") (String "Standard" ))
    (Property (rename Component_Kind "Component Kind") (String "Standard" ))
    (Property (rename Component_Kind "Component Kind") (String "Standard" ))
    (Property (rename Component_Kind "Component Kind") (String "Standard" ))
    (Property (rename Component_Kind "Component Kind") (String "Standard" ))
    (Property (rename Component_Kind "Component Kind") (String "Standard" ))
    (Property Description (String "Ground" ))
    (Property Description (String "Ground" ))
    (Property Description (String "Default Diode" ))
    (Property Description (String "Ground" ))
  )
)

```

(Property Description (String "Ground" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Ground" ))  
(Property Description (String "Ground" ))  
(Property FPGAVendor (String "Actel" ))  
(Property FPGAVendor (String "Actel" ))  
(Property FPGAVendor (String "Actel" ))  
(Property FPGAVendor (String "Actel" ))  
(Property FPGAVendor (String "Actel" ))  
(Property FPGAVendor (String "Actel" ))  
(Property FPGAVendor (String "Actel" ))  
(Property FPGAVendor (String "Actel" ))  
(Property FPGAVendor (String "Actel" ))  
(Property Footprint (String "DSO-C2/X3.3" ))  
(Property (rename Library\_Name "Library Name") (String "Actel 3200DX  
FPGA.IntLib" ))  
(Property (rename Library\_Name "Library Name") (String "Actel 3200DX  
FPGA.IntLib" ))  
(Property (rename Library\_Name "Library Name") (String "Miscellaneous  
Devices.IntLib" ))  
(Property (rename Library\_Name "Library Name") (String "Actel 3200DX  
FPGA.IntLib" ))  
(Property (rename Library\_Name "Library Name") (String "Actel 3200DX  
FPGA.IntLib" ))  
(Property (rename Library\_Name "Library Name") (String "Actel 3200DX  
FPGA.IntLib" ))  
(Property (rename Library\_Name "Library Name") (String "Actel 3200DX  
FPGA.IntLib" ))  
(Property (rename Library\_Name "Library Name") (String "Actel 3200DX  
FPGA.IntLib" ))  
(Property (rename Library\_Name "Library Name") (String "Actel 3200DX  
FPGA.IntLib" ))  
(Property (rename Library\_Reference "Library Reference") (String "GND" ))  
(Property (rename Library\_Reference "Library Reference") (String "GND" ))  
(Property (rename Library\_Reference "Library Reference") (String "Diode" ))  
(Property (rename Library\_Reference "Library Reference") (String "GND" ))  
(Property (rename Library\_Reference "Library Reference") (String "GND" ))  
(Property (rename Library\_Reference "Library Reference") (String "GND" ))  
(Property (rename Library\_Reference "Library Reference") (String "GND" ))  
(Property (rename Library\_Reference "Library Reference") (String "GND" ))  
(Property (rename Library\_Reference "Library Reference") (String "GND" ))  
(Property Manufacturer (String "Generic Components" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))  
(Property Published (String "8-Jun-2000" ))  
(Property Published (String "4/17/2002 3:26:26 PM" ))

```

(Property Published (String "4/17/2002 3:26:26 PM" ))
(Property Published (String "4/17/2002 3:26:26 PM" ))
(Property Published (String "4/17/2002 3:26:26 PM" ))
(Property Published (String "4/17/2002 3:26:26 PM" ))
(Property Published (String "4/17/2002 3:26:26 PM" ))
(Property Publisher (String "Altium Hobart Technology Centre" ))
(Property Publisher (String "Altium Hobart Technology Centre" ))
(Property Publisher (String "Altium Limited" ))
(Property Publisher (String "Altium Hobart Technology Centre" ))
(Property Publisher (String "Altium Hobart Technology Centre" ))
(Property Publisher (String "Altium Hobart Technology Centre" ))
(Property Publisher (String "Altium Hobart Technology Centre" ))
(Property Publisher (String "Altium Hobart Technology Centre" ))
(Property Publisher (String "Altium Hobart Technology Centre" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Diode" ))
(Property Simulation (String "DIODE" ))
(Property Footprint (String "DSO-C2/X3.3" ))
(Property Description (String "Ground" ))
(Property UniqueId (String "$$$\GQUSUFAJ" ))
)
(Instance (rename ATF22LV10CZ_25PI "ATF22LV10CZ-25PI")
  (viewRef NetlistView
    (cellRef ATF22LV10CZ_25PI
      (LibraryRef COMPONENT_LIB)
    )
  )
)
(Property (rename Class_I "Class I") (String "Programmable Logic Device" ))
(Property (rename Class_II "Class II") (String "Industry Standard SPLD" ))
(Property Comment (String "ATF22LV10CZ-25PI" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Datasheet (String "Latest Revision: Rev. 0779J?1/00" ))
(Property Description (String "High-Performance Electrically-erasable PLD" ))
(Property Footprint (String "DIP-24/X1.5" ))
(Property (rename Library_Name "Library Name") (String "Atmel Industry
Standard SPLD.IntLib" ))
  (Property (rename Library_Reference "Library Reference") (String
"ATF22LV10CZ-25PI" ))
    (Property Manufacturer (String "Atmel" ))
    (Property (rename Package_Reference "Package Reference") (String "24P3" ))
    (Property Published (String "15-Apr-2002" ))
    (Property Publisher (String "Altium Limited" ))
    (Property Revision (String "July-2002: Re-released for DXP Platform." ))
    (Property Footprint (String "DIP-24/X1.5" ))
    (Property Description (String "High-Performance Electrically-erasable PLD" ))
    (Property UniqueId (String "$$$\CXMWLFWF" ))
  )
)

```

```

)
(Instance Cpot1
  (viewRef NetlistView
    (cellRef Cap2
      (LibraryRef COMPONENT_LIB)
    )
  )
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Capacitor" ))
(Property Comment (String "10uF" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "CAPR5-4X5" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Cap2" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "23-Sep-2002" ))
(Property Publisher (String "Altium Limited" ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Cap" ))
(Property Simulation (String "CAP" ))
(Property Value (String "16V" ))
(Property Footprint (String "CAPR5-4X5" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\LRLWEVNV" ))
)
(Instance Cpot2
  (viewRef NetlistView
    (cellRef Cap
      (LibraryRef COMPONENT_LIB)
    )
  )
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Capacitor" ))
(Property Comment (String "10uF" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Capacitor" ))
(Property Footprint (String "VR5" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Cap" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Cap" ))

```

```

(Property Simulation (String "CAP" ))
(Property Value (String "16V" ))
(Property Footprint (String "VR5" ))
(Property Description (String "Capacitor" ))
(Property UniqueId (String "$$$\GCKDGWCS" ))
)
(Instance (rename GPS_CONNECTOR "GPS CONNECTOR")
(viewRef NetlistView
(cellRef Header_5X2H
(LibraryRef COMPONENT_LIB)
)
)
)
(Property (rename Class_I "Class I") (String "Connector" ))
(Property Comment (String "GPS CONNECTOR" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Header, 5-Pin, Dual row, Right Angle" ))
(Property Footprint (String "HDR2X5H" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Connectors.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Header
5X2H" ))
(Property Manufacturer (String "Generic Components" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Connector" ))
(Property Footprint (String "HDR2X5H" ))
(Property Description (String "Header, 5-Pin, Dual row, Right Angle" ))
(Property UniqueId (String "$$$\LBUYNVAH" ))
)
(Instance ICL7660
(viewRef NetlistView
(cellRef Component_1
(LibraryRef COMPONENT_LIB)
)
)
)
(Property Comment (String "ICL7660" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "" ))
(Property Footprint (String "None Available" ))
(Property (rename Library_Name "Library Name") (String "ICL7660.SchLib" ))
(Property (rename Library_Reference "Library Reference") (String
"Component_1" ))
(Property Footprint (String "None Available" ))
(Property UniqueId (String "$$$\JRJHFRUV" ))
)
(Instance (rename LCD_DISPLAY "LCD DISPLAY")

```



```

(viewRef NetlistView
  (cellRef Header_7X2
    (LibraryRef COMPONENT_LIB)
  )
)
(Property (rename Class_I "Class I") (String "Connector" ))
(Property Comment (String "LCD DISPLAY" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Header, 7-Pin, Dual row" ))
(Property Footprint (String "HDR2X7" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Connectors.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Header 7X2"
))
(Property Manufacturer (String "Generic Components" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Connector" ))
(Property Footprint (String "HDR2X7" ))
(Property Description (String "Header, 7-Pin, Dual row" ))
(Property UniqueId (String "$$$\GDQGBIJL" ))
)
(Instance QS3861
  (viewRef NetlistView
    (cellRef Header_12X2A
      (LibraryRef COMPONENT_LIB)
    )
  )
)
(Property (rename Class_I "Class I") (String "Connector" ))
(Property Comment (String "QS3861" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "" ))
(Property Footprint (String "HDR2X12_CEN" ))
(Property (rename Library_Name "Library Name") (String "QS3861.SchLib" ))
(Property (rename Library_Reference "Library Reference") (String "Header
12X2A" ))
(Property Manufacturer (String "Generic Components" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Connector" ))
(Property Footprint (String "HDR2X12_CEN" ))
(Property UniqueId (String "$$$\SEVNITND" ))
)
(Instance Rconv
  (viewRef NetlistView
    (cellRef Res1

```

```

    (LibraryRef COMPONENT_LIB)
  )
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Resistor" ))
(Property Comment (String "" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Resistor" ))
(Property Footprint (String "DSO-C2/X3.3" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Res1" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property Simulation (String "RESISTOR" ))
(Property Value (String "10K" ))
(Property Footprint (String "DSO-C2/X3.3" ))
(Property Description (String "Resistor" ))
(Property UniqueId (String "$$$\TEGNYTOE" ))
)
)
(Instance Repld
  (viewRef NetlistView
    (cellRef Res1
      (LibraryRef COMPONENT_LIB)
    )
  )
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Resistor" ))
(Property Comment (String "10K" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Resistor" ))
(Property Footprint (String "DIP-24/X1.5" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "Res1" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Signal_Integrity "Signal Integrity") (String "Res1" ))
(Property Simulation (String "RESISTOR" ))
(Property Value (String "" ))
(Property Footprint (String "DIP-24/X1.5" ))

```

```

(Property Description (String "Resistor" ))
(Property UniqueId (String "$$$\KIRLDUXY" ))
)
(Instance TRIMPOT
(viewRef NetlistView
(cellRef RPot1
(LibraryRef COMPONENT_LIB)
)
)
)
(Property (rename Class_I "Class I") (String "Passive" ))
(Property (rename Class_II "Class II") (String "Potentiometer" ))
(Property Comment (String "" ))
(Property (rename Component_Kind "Component Kind") (String "Standard" ))
(Property Description (String "Potentiometer" ))
(Property Footprint (String "DIP-24/X1.5" ))
(Property (rename Library_Name "Library Name") (String "Miscellaneous
Devices.IntLib" ))
(Property (rename Library_Reference "Library Reference") (String "RPot1" ))
(Property Manufacturer (String "Generic Components" ))
(Property Published (String "8-Jun-2000" ))
(Property Publisher (String "Altium Limited" ))
(Property Revision (String "July-2002: Re-released for DXP Platform." ))
(Property (rename Set_Position "Set Position") (String "0.5" ))
(Property Simulation (String "POT" ))
(Property Value (String "10K" ))
(Property Footprint (String "DIP-24/X1.5" ))
(Property Description (String "Potentiometer" ))
(Property UniqueId (String "$$$\SMMKDYDQ" ))
)
)
(Net (rename PLUS3_3V "+3.3V")
(Joined
(PortRef &1)
(PortRef &24 (InstanceRef ATF22LV10CZ-25PI))
(PortRef &3 (InstanceRef GPS CONNECTOR))
(PortRef &9 (InstanceRef GPS CONNECTOR))
)
)
)
(Net (rename PLUS5V "+5V")
(Joined
(PortRef &8 (InstanceRef ICL7660))
(PortRef &2 (InstanceRef LCD DISPLAY))
(PortRef &1 (InstanceRef TRIMPOT))
)
)
)
(Net A18
(Joined

```

```
    (PortRef &5 (InstanceRef ATF22LV10CZ-25PI))
  )
)
(Net A19
  (Joined
    (PortRef &6 (InstanceRef ATF22LV10CZ-25PI))
  )
)
(Net A20
  (Joined
    (PortRef &7 (InstanceRef ATF22LV10CZ-25PI))
  )
)
(Net A21
  (Joined
    (PortRef &8 (InstanceRef ATF22LV10CZ-25PI))
  )
)
(Net A22
  (Joined
    (PortRef &9 (InstanceRef ATF22LV10CZ-25PI))
  )
)
(Net A23
  (Joined
    (PortRef &10 (InstanceRef ATF22LV10CZ-25PI))
  )
)
(Net A24
  (Joined
    (PortRef &11 (InstanceRef ATF22LV10CZ-25PI))
  )
)
(Net A25
  (Joined
    (PortRef &13 (InstanceRef ATF22LV10CZ-25PI))
  )
)
(Net A26
  (Joined
    (PortRef &14 (InstanceRef ATF22LV10CZ-25PI))
  )
)
(Net A27
  (Joined
    (PortRef &15 (InstanceRef ATF22LV10CZ-25PI))
  )
)
```

```

)
)
(Net A28
  (Joined
    (PortRef &16 (InstanceRef ATF22LV10CZ-25PI))
  )
)
)
(Net A29
  (Joined
    (PortRef &17 (InstanceRef ATF22LV10CZ-25PI))
  )
)
)
(Net A30
  (Joined
    (PortRef &18 (InstanceRef ATF22LV10CZ-25PI))
  )
)
)
(Net (rename CS3_ "CS3*")
  (Joined
    (PortRef &2 (InstanceRef ATF22LV10CZ-25PI))
  )
)
)
)
(Net D7
  (Joined
    (PortRef &7 (InstanceRef LCD DISPLAY))
  )
)
)
(Net LCDCS1
  (Joined
    (PortRef &22 (InstanceRef ATF22LV10CZ-25PI))
    (PortRef &6 (InstanceRef LCD DISPLAY))
  )
)
)
(Net LCDCS2
  (Joined
    (PortRef &21 (InstanceRef ATF22LV10CZ-25PI))
  )
  (Property SuppressERC (String "True" ))
)
)
(Net (rename MDA11___MDA31 "MDA11...MDA31")
  (Joined
  )
)
)
(Net MDA15
  (Joined
    (PortRef &4 (InstanceRef GPS CONNECTOR))
  )
)

```

```

)
)
(Net (rename MGPI05___MGPI019 "MGPI05...MGPI019")
  (Joined
  )
)
(Net (rename MPWM0___MPWM19 "MPWM0...MPWM19")
  (Joined
  )
)
(Net (rename NetATF22LV10CZ_25PI_23 "NetATF22LV10CZ-25PI_23")
  (Joined
    (PortRef &23 (InstanceRef ATF22LV10CZ-25PI))
    (PortRef &1 (InstanceRef Repld))
  )
)
(Net NetCpot1_1
  (Joined
    (PortRef &1 (InstanceRef Cpot1))
    (PortRef &2 (InstanceRef ICL7660))
  )
)
(Net NetCpot1_2
  (Joined
    (PortRef &2 (InstanceRef Cpot1))
    (PortRef &4 (InstanceRef ICL7660))
  )
)
(Net NetCpot2_2
  (Joined
    (PortRef &2 (InstanceRef Cpot2))
    (PortRef &5 (InstanceRef ICL7660))
    (PortRef &2 (InstanceRef TRIMPOT))
  )
)
(Net (rename NetLCD_DISPLAY_3 "NetLCD DISPLAY_3")
  (Joined
    (PortRef &3 (InstanceRef LCD DISPLAY))
    (PortRef &3 (InstanceRef TRIMPOT))
  )
)
(Net Net_2
  (Joined
    (PortRef &2)
    (PortRef Y)
    (PortRef &24 (InstanceRef QS3861))
  )
)

```

```

    (PortRef &1 (InstanceRef Rconv))
  )
)
(Net Net_Y
  (Joined
    (PortRef Y)
    (PortRef &12 (InstanceRef QS3861))
  )
)
(Net Net_Y
  (Joined
    (PortRef Y)
    (PortRef &5 (InstanceRef GPS CONNECTOR))
  )
)
(Net Net_Y
  (Joined
    (PortRef Y)
    (PortRef &12 (InstanceRef ATF22LV10CZ-25PI))
  )
)
(Net Net_Y
  (Joined
    (PortRef Y)
    (PortRef &1 (InstanceRef LCD DISPLAY))
  )
)
(Net Net_Y
  (Joined
    (PortRef Y)
    (PortRef &1 (InstanceRef Cpot2))
  )
)
(Net Net_Y
  (Joined
    (PortRef Y)
    (PortRef &3 (InstanceRef ICL7660))
  )
)
(Net (rename R_W_ "R/W*")
  (Joined
    (PortRef &3 (InstanceRef ATF22LV10CZ-25PI))
  )
)
(Net RXD2
  (Joined

```

