Using the Intec SS555 Embedded Development Kit

Intec Automation Inc.
Victoria, British Columbia

version 1.3

WARRANTY

Intec Automation Inc. warrants the hardware components of this product to be free from defects in material and workmanship. This warranty extends for a period of 90 days from the date of purchase. Any component under warranty will be repaired or replaced at the sole discretion of Intec Automation Inc., without charge to the purchaser, providing that the return of the component or board is pre-authorized by Intec, that shipping is prepaid by the purchaser, and that Intec determines the defect is not a result of misuse.

The components of this product are provided "as is" without warranty. The entire risk for the results and performance of these components is assumed by the purchaser. Intec Automation Inc. does not warrant, guarantee or make any representation regarding the use of this product.

No other warranties are made, expressly or implied, including, but not limited to, the implied warranties of merchantability and suitability of products for a particular purpose. In no event will Intec Automation Inc. be held liable for additional damages, including lost profits, lost savings or other incidental or consequential damages arising from the use or inability to use Intec's products or the products resold by Intec.

DISCLAIMERS

Intec Automation Inc. reserves the right to make changes without notice, to any product, to improve reliability, performance, capabilities, design or ease of use, or to reduce size or cost.

COPYRIGHT

Intec Automation Inc. products may not be used as components in life support devices of any description.

The design of the Freedom16-mite is copyrighted, and copying part or all of the design is prohibited by law.

Table of Contents

TABLE OF CONTENTS I

LIST OF TABLES III

LIST OF FIGURES IV

1. SUMMARY 1

2. GLOSSARY 2

3. NOMENCLATURE 3

4. INTRODUCTION 4

5. SS555 OVERVIEW 5

6. VERIFYING OPERATION OF THE SS555 7

7. GETTING STARTED WITH INTEC PROJECT MANAGER 8

8. GETTING STARTED WITH ICDPPC 10

9. HEADERS 13

- General Information 13
- Header Location 13
- Pinout of the SS555 Connectors 15

10. POWER 16

- General Information 16
- Power Pin Location 16
- Power Setup 17
- Advanced Users 17

11. RESETS 18

- General Information 18
- Reset Pin Location 18
- *PORESET - Power-on Reset 18
- *HRESET - Hardware Reset 19
- *SRESET - Software Reset 19

12. BACKGROUND DEBUG MODE (BDM) PORT 20

- General Information 20
- Pin Location 20
- Advanced Users 21

13. MEMORY 22

- General Information 22
- Expansion Header Location 22
- *CS0 - External SRAM 23

*CS1 - *CS2 23
ADDITIONAL CHIP SELECTS 24
INTERNAL SRAM 24
FLASH EPROM 24

14. FLASH PROGRAMMING 25

15. INTERRUPT REQUEST (IRQ) 31

GENERAL INFORMATION 31
IRQ PIN LOCATION 31
SPECIAL NOTE 32
ADVANCED USERS 32

16. PULSE WIDTH MODULATION (PWM) 33

GENERAL DESCRIPTION 33
PWM PIN LOCATION 33
PERIOD AND DUTY CYCLE CALCULATION 33

17. QUEUED ANALOG TO DIGITAL CONTROLLER(QADC) 36

GENERAL INFORMATION 36
PIN LOCATION 36
SPECIAL NOTE 36
ADVANCED USERS 37
Specifying the QCLK 37
Multiplexing the QADC lines 37

18. TIMERS (PIT/RTC/DEC/TB/WDT) 39

19. CPLD 40

INTRODUCTION 40
HARDWARE 40
INTERFACING TO THE CPLD 40
EXTRA CS SIGNALS 41

APPENDIX A – SCHEMATIC DIAGRAMS 1

APPENDIX B – DAUGHTER BOARDS 1

CLOCK CALENDAR DAUGHTER BOARD 1
ANALOG TO DIGITAL MULTIPLEXER DAUGHTER BOARD 1
LOW POWER DAUGHTER BOARD 1
SWITCHING POWER SUPPLY 1

REFERENCES 2

List of Tables

Table 1 - Signal Nomenclature 3
Table 2 - SS555 Module Icons and Section Numbers 6
Table 3 - Header Descriptions 13
Table 4 - Fanout of the SS555 power supplies 16
Table 5 - Possible BDM pinouts available on the MPC555 21
Table 6 - SS555 Chip Selects 23
Table 7 - IMMR register 24
Table 8 - Build Type Characteristics 25
Table 9 - IRQ pinout versus GPIO pinout 32
Table 10 - *PORESET clock configuration (MODCK1:3) 32
Table 11 - Some Possible PWM frequencies and resolutions 35
Table 12 - RTL555 vs. MPC555 ADC naming convention 36
Table 13 - Common calculations for QCLK 37
Table 14 - RTL555 vs. MPC555 Naming Convention for ADC 38
Table 15 - Calculation used with the Generic sysclk_set() function 38
Table 16 - Timer Properties 39
Table 17 - CPLD functionality 40
Table 18 -Partial Address Decoding of CS3a/CS3b/CS3c 41

List of Figures

Figure 1 – Pins used by flash program (blinkall.prj) 7
Figure 2 - Header Configuration of the SS555 13
Figure 3– Power Scheme for the SS555 16
Figure 4 - Correct Placement of Power Jumpers 17
Figure 5 - Location of Reset Pins 18
Figure 6 - Schematic representation of the CPLD *PORESET code 18
Figure 7 - *HRESET design considerations 19
Figure 8 - BDM debug cable connecting to the SC555's BDM port 20
Figure 9 - Location of Peripheral Expansion Pins 22
Figure 10 - Little Endian vs. Big Endian Nomenclature 23
Figure 11 - Location of IRQ pins 31
Figure 12 - Location of PWM pins 33
Figure 13 - Origin of the PWM clock period 34
Figure 14 - Location of QADC pins 36
Figure 15 - Location of Additional Chip Select Pins 41
Figure 16 - Location of Clock Calendar Pins 42
Figure 17 - Location of Hardware Handshaking Pins 43
Figure 18 - Block Diagram of the Schematics 1

1. Summary

Intec Automation Inc. has created a very small development board based on the Motorola MPC555 embedded PowerPC© microcontroller. The MPC555 has many features built around its 40MHz PowerPC© core including: 448kB of internal flash EPROM, 26kB of internal SRAM, 32 ADC pins, 2 SCI ports, 1 SPI port, 8 PWM pins, 10 Double Action pins, 8 IRQ pins, 2 CAN 2.0b ports, 16 IO pins, and 2 x 16 channel TPU ports.

With all this functionality comes a steep learning curve, even for engineers experienced with other microcontrollers. Though there is a very thorough manual from Motorola, it can be somewhat overwhelming. The function of this document is to allow users to start to program the SS555 development board without an in-depth knowledge of the Motorola MPC555 User's Manual. This manual is meant as an introduction to the SS555 development board, as well as the capabilities of different modules on the MPC555.

2. Glossary

ADC — Analog to Digital Controller. Converts an analog signal from 0-5V0 to a numerical count.

BDM — Background Debug Mode. A non-intrusive method of taking control of the MPC555 through a debug program running on a host PC.

Big Endian — see Endian

CAN — Controller Area Network. An asynchronous communications protocol used in automotive and industrial control systems.

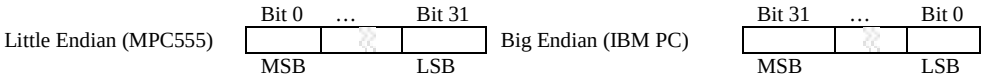
CPLD — Complex Programmable Logic Device. Provides the logic for many of the unique features on the SS555 board.

DA — Double Action Functions that capture or output a time stamped or timed double (on & off) sequence of digital events.

DEC — Decrementer. A down counter that can generate interrupts.

Duty Cycle — The % ON time of a PWM [pulse train] signal.

Endian — Big Endian / Little Endian. Describes the convention for naming the most significant bit in a word. The MSB is 0 in little endian and is 31 in big endian. Little endian is used on the MPC555. The endians are character's in Jonathan Swift's Gulliver's Travels, who fought wars over which side of a boiled egg should be cracked.



GCC — GNU C. An open license GNU C compiler. The compiler with the SS555 development system compiles under DOS/Windows for the PowerPC (MPC555) architecture.

GDB — GNU debugger. At the moment this is not supported by Intec Automation Inc.

GNU — GNU not UNIX. A recursive acronym. A C compiler governed by the GNU Public License(GPL).

GPIO — General purpose input/output. Pins that can either generate or capture logic 0's or 1's. Though these are 5V0 pins, the minimum hi/lo levels are such that 3V3 signals can also be used as inputs to the MPC555.

GPL — The GNU public license. The licensing agreement that binds the use and distribution of GCC. A copy of the GPL can be found on the SS555 CD-ROM distributed with the development system.

Header — Rows of equally spaced pins protruding above the circuit board to which connectors may be attached. All headers on the SS555 are double row, spaced 0.01", to interface to standard IDC (Insulation Displacement Connectors) ribbon cable connectors.

IPM – Intec Project Manager. Intec's own easy to use graphical interface for compiling under GCC.

IRQ – Interrupt Request. The ability of a module or external pin to request servicing of the interrupt controller. An IRQ stops the current execution of a program and jumps to a piece of code found in the interrupt vector table.

Little Endian see Endian.

LSB Least significant bit. Bit 31 on the MPC555.

MPC555 The Motorola PowerPC microcontroller that is at the heart of the Steroid Stomp.

MSB Most significant bit. Bit 0 on the MPC555.

Partial Address Decoding multiple addresses refer to the same memory location.

PIT Periodic Interrupt Timer. A down counter that can generate interrupts.

PWM Pulse Width Modulation, that is, the ON time of a digital pulse is varied to vary the Duty Cycle (% ON time) of the signal. When filtered, can be used to generate analog output voltages from 0V to 5V. Used frequently in motor control.

SS555 Steroid Stomp 555. This refers to the development board. The current revision of the board is version 1.0. The version number is marked on the lower left-hand side of the board.

TB Time Base. Up counter that can generate an interrupt at two user selected values of the TB.

TPU Time Processor Unit. Executes complex timing control functions independent of the PowerPC core. The TPU is currently not supported by Intec Automation Inc.

VisionCLICK see Vision PROBE.

VisionPROBE BDM debugging cable/hardware from Windriver Inc., that connects to the MPC555's BDM port and allows their VisionCLICK debug software to access the MPC555 chip through the PC's printer port.

WDT Watchdog Timer. A timer that resets the MPC555 if the running program gets caught in an infinite loop or branches incorrectly.

3. Nomenclature

*Signal	Active Low	0	Driven	Asserted
		1	Not Driven	Negated
Signal	Active High	0	Not Driven	Negated
		1	Driven	Asserted

Table 1 - Signal Nomenclature

4. Introduction

Intec Automation Inc. is a company dedicated to creating new and innovative products for the microcontroller industry. Intec Automation Inc. creates extremely small microcontrollers that can be used not only to speed the development process, but can affordably be embedded in small to medium product production runs. These very small development boards are packaged with a C compiler, debugging software, hardware specific runtime library, and a complete manual and schematic diagrams.

Intec Automation Inc. currently produces a series of development systems based on the Motorola HC16 microcontroller. The development system based on the Motorola MPC555 is the first in a series of planned development systems aimed at broadening Intec Automation Inc.'s line card.

This manual is the primary reference for users of the SS555. The SS555 is broken down into modules, and the same modular design is kept throughout the hardware, schematics, runtime libraries, and manual.

This manual has two sections. The first section discusses the hardware design of the SS555 and discusses the capabilities of each module in the MPC555. The second section discusses the modules in terms of the software capabilities of the runtime libraries. Each section is presented in a separate file.

For information on IPM (Intec Project Manager), GCC, and the several debugger options, the user is referred to the specific program documentation.

5. SS555 Overview

The "Steroid Stomp" is a completely integrated development system for the Motorola MPC555. The SS555-DK development kit contains a microcontroller development board, a GNU C compiler, P&E's ICD-PPC debugger, ICDPPC BDM debug cable and ProgPPC flash EPROM programmer, and a C library of useful functions (RTL555). These components are integrated with Intec's project management software (IPM).

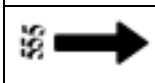
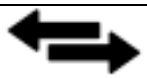
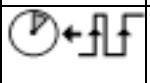
The MPC555 CPU is composed of a Power-PC core and internal modules. In keeping with the modular design of the CPU, the schematics, headers on the SS555, runtime libraries and example code are also broken down into the same modules. The basic modules are:

- Power system – 5V0, 3V3, GND, Analog 5V0, Analog GND
- Background Debug Mode (BDM) port
- Resets – Power-On , Hardware, Software
- Memory – Internal / External SRAM, External / Internal Flash EPROM
- Analog to Digital module (QADC)
- Modular Input / Output System (MIOS) – MPIO, MDA, MPWM, MIOS counters
- Serial module (QSMCM) – QSPI, QSCI
- Controller Area Network modules (TouCAN) – CAN 2.0B controller
- Time Processor Unit (TPU)
- Interrupt Request Subsystem (IRQ)

Icons in the top right hand corner of each page can be used to quickly identify the modules throughout this manual. Below is a list of SS555 modules, icons, section numbers in this manual, and .pdf file in the Motorola MPC555 User's Manual (available from www.maneacambs.freeserve.co.uk/555_0600).

Power	BDM	Reset	Memory	ADC	GPI
					
Power	Background Debug Mode (BDM)		Memory		
c8cloc.pdf	c22jtag.pdf	c7rese.pdf	c10mem.pdf	c13qadc.pdf	c13qadc.pdf c15mios.pdf c6sys.pdf

SCI	SPI	CAN	TPU	Clocks	Timing
					
					Timers
c14qsmc.pdf	c14qsmc.pdf	c16touc.pdf	c17tpu3.pdf	c8cloc.pdf	c6sys.pdf

GPO	GPIO	PWM	DA	IRQ	CPLD
					
		Pulse Width Modulation (PWM)		Interrupt Request (IRQ)	CPLD
c15mios.pdf c6sys.pdf	c15mios.pdf c6sys.pdf	c15mios.pdf	c15mios.pdf	c6sys.pdf	

FLASH	IPM	ICDPPC	PROGPPC		
					
Flash	Intec Project Manager	P&E Background Debug Software	P&E Flash Programming Software		
c19cmf.pdf c7rese.pdf					

Table 2 - SS555 Module Icons and Section Numbers

6. Verifying Operation of the SS555

The SS555 is shipped with the program blinkall.prj burned into the flash of the microcontroller to enable users to test the operation of the board without much knowledge of the microcontroller. This program toggles the GPIO and IRQ pins high and low, sets the PWM pins to output different pulse widths, samples the Analog to Digital Pins and writes out the values to the PC via COM1. This program allows the user to verify that the SS555 is operating correctly and that it is connected serially to the host to the expected port.

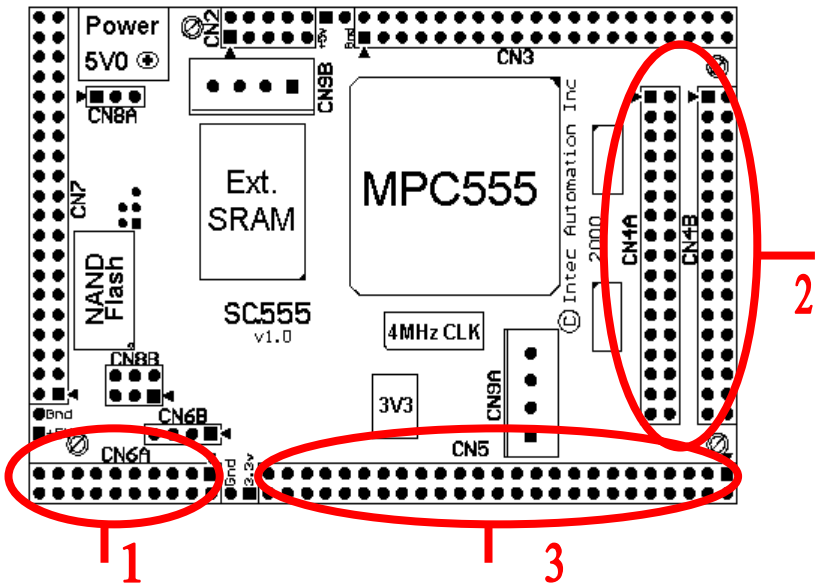


Figure 1 – Pins used by flash program (blinkall.prj)

1) Before powering the board, first attach the SS555 to COM1/2 port to the PC. This is done through a special 20-pin to 2xDB-9 connector. The pinout is given on page 5 of 8 of the Schematic Diagrams and can be purchased through Intec Automation Inc. The 20-pin connector is placed on CN6a. Care must be taken to ensure that the red wire on the connector matches up with the triangle that indicates Pin 1 on the SS555 board. The program in flash transmits at 9600 baud, 8 data bits, no parity, and 1 stop bit (9600-8-N-1). Setup a terminal program to receive data from COM1/2.

   Before connecting power, please review Power Setup to ensure that the jumpers on the SS555 are correct. A **regulated** 5V, **Center Positive** regulator should be connected to CN1. On power-up, the terminal program should begin to display information from the SS555's analog to digital pins.

2) To test the operation of the Analog to Digital section, short the odd numbered pins of CN4a and CN4b to the even numbered pins (exclude the last pin in each connector as this is an external trigger pin). Connecting a jumper across the connector results in an analog value being shorted to AGND and should display a value close to 0. Once the jumper is removed, the pin will start to float high and the value of the A/D pin will increase. Alternately, measure your galvanic skin resistance by placing your finger on some A/D pins and their associated ground pins (CN4A/B).

3) Connect either an oscilloscope or a logic probe to the pins of CN5. Pins 23-46 will all toggle high then low. The pins toggle very quickly and the change in voltage is too quick to be seen with a voltmeter. Logic probes should have both HI and LO LEDs on. Placing the logic probe to pulse mode will generate a visible toggling of the pulse LED.

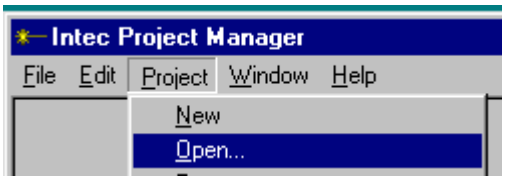
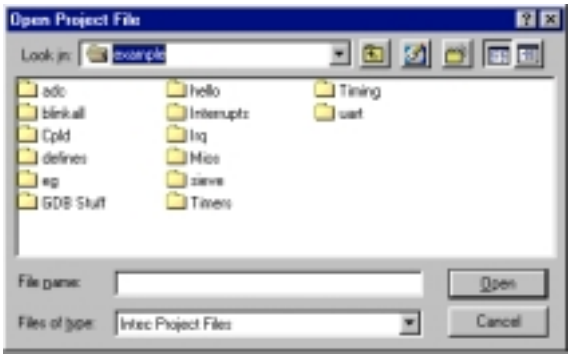
If all of these test work, the SS555 board is in good operation and the PC is properly configured for serial communication with the board.

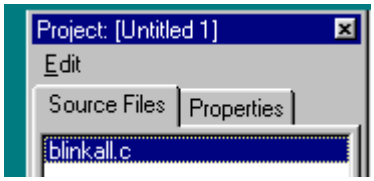
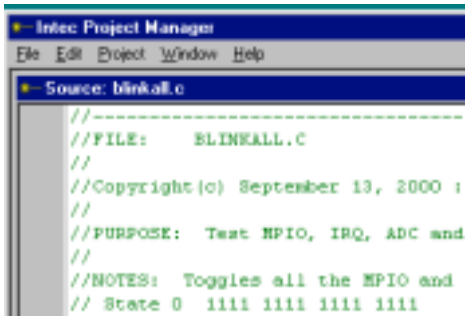
7. Getting Started with Intec Project Manager

Before reading this section, the Intec Project Manager should be installed on the PC. For installation instructions, please refer to the SS555-DK Installation Manual.

Intec Project Manager (IPM), is a graphical user's interface allowing users to easily setup project files and communicate with GNU C. The purpose of this section is to quickly get the user to open, view, and compile an existing GNU C project.

Open IPM	
Either double click on the IPM icon on the desktop or open IPM from the start menu. These should both have been created during the installation of IPM **Double click on IPM**	 desktop start menu

Open an Existing Project within IPM	
Open a new project within IPM. If IPM is exited with an open project, the next time IPM is started, the same project will open by default. **Click Project – Open**	
Find the example subdirectory. This will be included in the \..\gcc\example. Each subfolder contains a different project. It is strongly suggested that a new folder be started for every project as the number of related files can quickly become overwhelming. **Find the example subdirectory. The default path is c:\Intec\SS555\GCC\EXAMPLE.**	
Enter the blinkall subdirectory and open the blinkall project. **Double click on the blinkall folder and double click on blinkall.ipj.**	

View the C code	
Open the program and briefly inspect the C code. The purpose at the present should be to learn how to maneuver through IPM rather than to learn the details of the code. **To view the C code, double click on the blinkall.c**	
The c code can be modified in this window. Clicking on file-open can open other files not explicitly listed in the project window. **Though code can be changed in this window, do not make any modifications at this time.**	

Compile the project	
There are several ways to compile a project. Individual c files can be compiled by highlighting them in the project window and selecting "Compile". "Make" compiles and links any sections of a project whose time stamp is not up to date. "Rebuild all" forces a build of all sections, regardless of the time stamp. **Click Project – Rebuild All.**	
The build messages window echoes messages from the gcc compiler and other tools that are executed in the creation of the executables. Briefly look at the build messages. They will be discussed in detail later.	

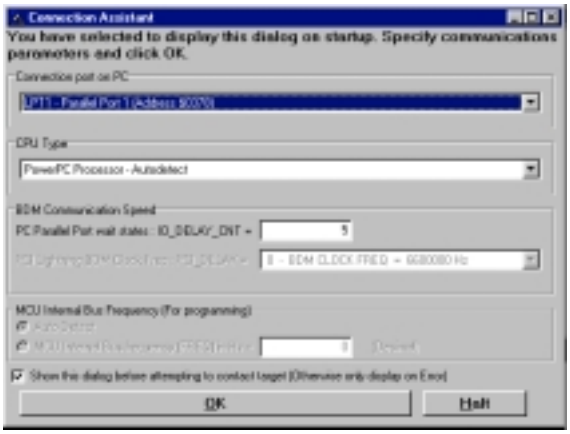
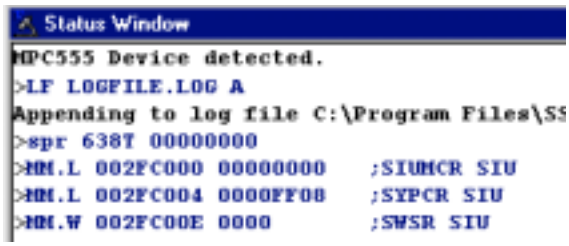
The project is now ready to be either loaded into SRAM, mainly as a result of the default settings. However, because the flash memory has a limited life, it should be changed sparingly. A program should first be thoroughly debugged and verified by stepping/executing it from SRAM before even thinking of burning [loading] it in flash.

The next section shows how to load the program into SRAM and step through the code using the P&E PowerPC debugger, and the following section shows the same steps for the Windriver VisionPROBE debugger.

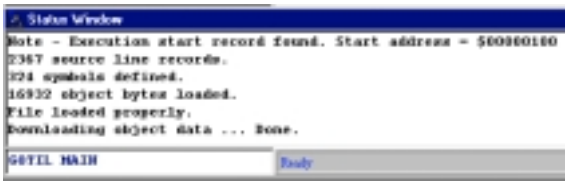
8. Getting Started with ICDPPC

ICDPPC is not part of the IPM suite. This is a separate program from P&E Microcomputers Inc., which must be installed separately. For installation instructions, please refer to the SS555-DK Installation Manual.

Open P&E's ICDPPC Debugger	
Double click on the ICDPPC icon from the start menu.	

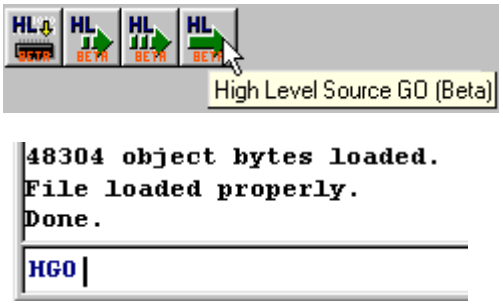
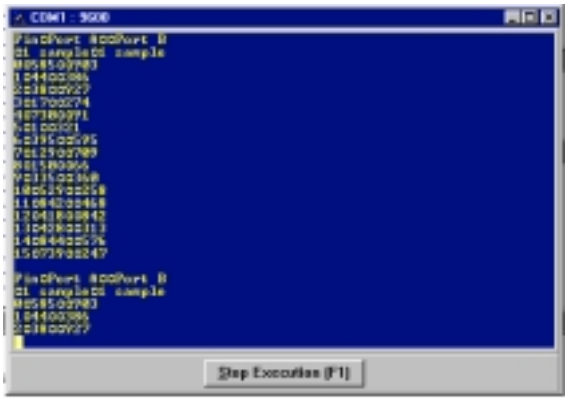
Establish communication with the SS555	
Ensure that the correct connection port has been selected and that the CPU type is PowerPC Processor – Autodetect. These settings are remembered on subsequent opens. The IO_DELAY_CNT can be increased if the communication with the SS555 is slightly unreliable. Ensure that the SS555 board is powered and that the power PC interface cable is attached. See Background Debug Mode (BDM) for setup instructions. Connect power to the SS555 board. Connect the P&E interface cable with attention to polarity. Choose the correct port and click OK.	
The Power PC device is detected and a script called startup.icd is run to properly configure the SS555. The default startup.icd file resides in the ..\ICDPPC\startup directory. If the MPC555 is not detected, either the message "Cannot enter BDM mode" or "PowerPC cable not detected" will be displayed. Ensure that the board is powered and the PowerPC interface cable is attached with the correct polarity.	

Load the program into SRAM on the SS555	
The GNU C compiler produces a variety of different file formats for different debuggers and programmers. The ICDPPC debugger uses the .s19 and the .695 format. The .695 format should always be used as it contains more debugging information than the .s19 file. **To load the .695 file either type HLOAD or click on the High level load button. Locate the file and click OK.**	

Locate function main( ) in the program	
To run the program until the start of any function, type GOTIL followed by the function name. The program will then execute until the function is reached. **Type "gotil main".**	

Toggle to see the c code if it is available	
All user written sections of code are visible in both the original c and in assembly. The display mode of either code window 1 or 2 can be changed by clicking with the right mouse button and then selecting "Show Disassembly" and then selecting either the source or the disassembly. **If the main program is in assembly right click the mouse when in the code window, select Show Disassembly-Show Source/Disassembly.**	

Step through the code	
The program can either be single stepped or multi-stepped. Single step will run the program until the next c instruction. This may consist of several assembly line instructions. Multi-step runs until the user presses a key. Both step functions update the register windows after each instruction. This enables the user to see the exact state of the MPC555. **To view registers and control program execution click either step (type HST) or multi (type HSTEPFOR)**	

Run the code in real time.	
To run the code at the same speed as it will run when programmed into flash, press the GO button. The debugger will not update any of its windows during this process. **To run in real time, click GO or type HGO at the prompt.**	
If the serial port is turned on (this is the default case), any serial communication to the PC will be displayed within the pop-up window. Execution can be stopped at any time by clicking the Stop Execution button or pressing F1. **Click the stop execution button to stop running the program on the SS555.**	

More detailed information about the P&E debugger can be found in their online help documentation or on their website at www.pemicro.com.

9. Headers

General Information

The CN1 Power adapter is a 2.5mm center positive coaxial adapter. The connections to the CAN_A and CAN_B are 0.014" terminal blocks. The rest of the connections to the SS555 are made through 0.10" male headers. Pin 1 on each header is marked by a white arrow. Pin 1 can also be ascertained by looking for a square via on the bottom side of the board. The pins on each header are grouped according to MPC555 function. The SS555 headers are grouped in the same manner as the modules in the MPC555 User's Manual.

Header Location

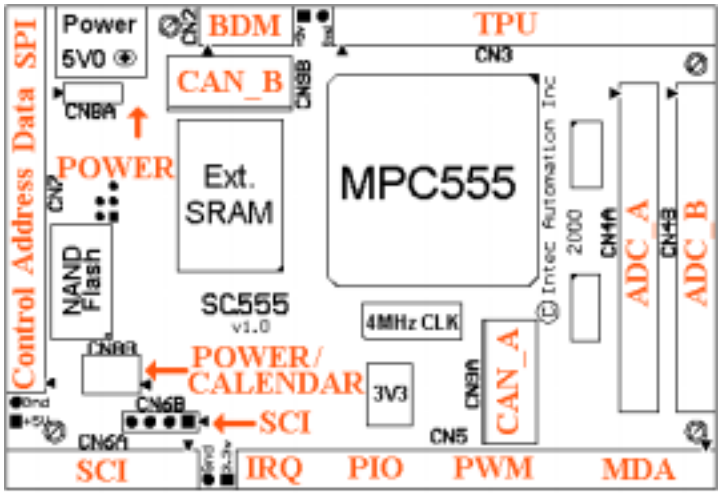


Figure 2 - Header Configuration of the SS555

Header	Type	Main Use	Description	Alternate
CN1	coax.	Power	5v regulated input, **Center Positive**	
CN2	2x5	BDM	Debug cable connected to PC	
CN3	2x20	TPUA/B	2 x 16 channel Time Processor Units	
CN4A	2x17	QADCA	Analog to Digital input pins	GPIO
CN4B	2x17	QADCB	Analog to Digital input pins	GPIO
CN5	2x25	MIOS/IRQ	IRQ / PWM / DA and MPIO pins	GPIO
CN6A	2x10	SCIA/B	2 Serial communication ports	GPIO
CN6B	1x4	SCIB	Connector for serial LCD/kpd (RS-232 level)	GPIO
CN7	2x20	CNTRL/SPI	Address/data/chip select/control lines, SPI	
CN8A	1x3	Low Power	Optional low power daughter board	
CN8B	2x3	Low Power	Optional low power/clock calendar daughter board	*PORESET
CN9A	1x4	CAN	Priority based controller area network 2.0B protocol	
CN9B	1x4	CAN	Priority based controller area network 2.0B protocol	

Table 3 - Header Descriptions

Pinout of the SS555 Connectors

The following provides a quick reference to the pinout of the headers on the SS555. The complete schematics are in Appendix A – Schematic Diagrams.

CN1A (Coax Connector)
Power Supply

5V(coax)	1

CN2 (2x5 header)
BDM connector

VFLS0	1	2	/RESET
GND	3	4	DSCK
GND	5	6	VFLS1
*RESET	7	8	DSDI
5V0	9	10	DSDO

CN3 (2x20 Header)
TPU Connector

BTPUCH0	1	2	BTPUCH1
BTPUCH2	3	4	BTPUCH3
BTPUCH4	5	6	BTPUCH5
BTPUC H6	7	8	BTPUCH7
GND	9	10	5V0
BTPUCH8	11	12	BTPUCH9
BTPUCH10	13	14	BTPUCH11
BTPUCH12	15	16	BTPUCH13
BTPUCH14	17	18	BTPUCH15
BT2CLK	19	20	AT2CLK
GND	21	22	5V0
ATPUCH0	23	24	ATPUCH1
ATPUCH2	25	26	ATPUCH3
ATPUCH4	27	28	ATPUCH5
ATPUCH6	29	30	ATPUCH7
GND	31	32	5V0
ATPUCH8	33	34	ATPUCH9
ATPUCH10	35	36	ATPUCH11
ATPUCH12	37	38	ATPUCH13
ATPUCH14	39	40	ATPUCH15

CN4A (2x17 Header)
QUEUED A/D PORT A

AN0_A	1	2	AGND
AN1_A	3	4	AGND
AN2_A	5	6	AGND
AN3_A	7	8	AGND
AN4_A	9	10	AGND
AN5_A	11	12	AGND
AN6_A	13	14	AGND
AN7_A	15	16	AGND
AN8_A	17	18	AGND
AN9_A	19	20	AGND
AN10_A	21	22	AGND
AN11_A	23	24	AGND
AN12_A	25	26	AGND
AN13_A	27	28	AGND
AN14_A	29	30	AGND
AN15_A	31	32	AGND
ETRIG1	33	34	5V0 (Digital)

CN4B (2x17 Header)
QUEUED A/D PORT B

AN0_B	1	2	AGND
AN1_B	3	4	AGND
AN2_B	5	6	AGND
AN3_B	7	8	AGND
AN4_B	9	10	AGND
AN5_B	11	12	AGND
AN6_B	13	14	AGND
AN7_B	15	16	AGND
AN8_B	17	18	AGND
AN9_B	19	20	AGND
AN10_B	21	22	AGND
AN11_B	23	24	AGND
AN12_B	25	26	AGND
AN13_B	27	28	AGND
AN14_B	29	30	AGND
AN15_B	31	32	AGND
ETRIG2	33	34	5V0 (Digita

CN5 (2X25 Header)
MIOS / IRQ

MDA0	1	2	MDA1
MDA2	3	4	MDA3
MDA4	5	6	MDA5
MDA6	7	8	MDA7
MDA8	9	10	MDA9
5V0	11	12	GND
sPWM0	13	14	MPWM1
sPWM2	15	16	MPWM3
sPWM4	17	18	MPWM5
sPWM6	19	20	MPWM7
5V0	21	22	GND
MPIO0	23	24	MPIO1
MPIO2	25	26	MPIO3
MPIO4	27	28	MPIO5
MPIO6	29	30	MPIO7
MPIO8	31	32	MPIO19
MPIO10	33	34	MPIO11
MPIO12	35	36	MPIO13
MPIO14	37	38	MPIO15
FRZ	39	40	/IRQOUT
/IRQ0	41	42	/IRQ1
/IRQ2	43	44	/IRQ3
/IRQ4	45	46	/IRQ5
/IRQ6	47	48	/IRQ7
3v3	49	50	GND

CN6A (2X10 Header)
QSCI

TX2	1	2	
PC_RX2	3	4	
PCTX2	5	6	
	7	8	
GND	9	10	RX2
TX1	11	12	
PC_RX1	13	14	PC_RTS
PC_TX1	15	16	PC_CTS
PC_DTR	17	18	
GND	19	20	RX1

CN6B (1X4 Header)
Serial LCD/KPD

5V0	1
PC_RX2	2
PC_TX2	3
GND	4

CN7 (2x20 Header)
Address/Data/Control/QSPI

GND	1	2	5V
MOSI	3	4	MISO
PCS0	5	5	PCS1
PCS2	7	8	PCS3
SCK	9	10	3V3
*HRESET	11	12	BCK/BUCK
D16	13	14	D17
D18	15	16	D19
D20	17	18	D21
D22	19	20	D23
D24	21	22	D25
D26	23	24	D27
D28	25	26	D29
D30	27	28	D31
A0	29	30	A1
A2	31	32	A3
A4	33	34	RD/*WR
*OE	35	36	*CS3C
*CS3B	37	38	*CS3A
*CS2	39	40	*CS1

CN8A (1X3 Header)
Daughter Board Power Supply

GND	1
5V0 (to board)	2
5V0 (from coax)	3

CN8B (2X3 Header)
Daughter Board Power Supply/Calendar chip

*CS3D	1	2	3V3
CLE	3	4	VSTBY
*CE	5	6	PWR_ON

CN9A (1x4 Header)
CAN Controller A

CANA_V-	1
A_CANL	2
A_CANH	3
CANA_V+	4

CN9B (1x4 Header)
CAN Controller B

CANB_V-	1
B_CANL	2
B_CANH	3
CANB_V+	4

10. Power

General Information

The MPC555 is a dual powered CPU requiring both 3V3 and 5V0 for correct operation. The user must supply the SS555 board with ***regulated*** 5V either through CN1 or through one of Intec's Low Power Daughter Boards.

Power Pin Location

There are pins providing 5V0 and GND on every header on the board. There are also 3V3 pins located on CN5 and CN7, as well as convenient test points for 3V3, 5V0, and GND. For location of the power pins, see Schematic 6/8 and 7/8.

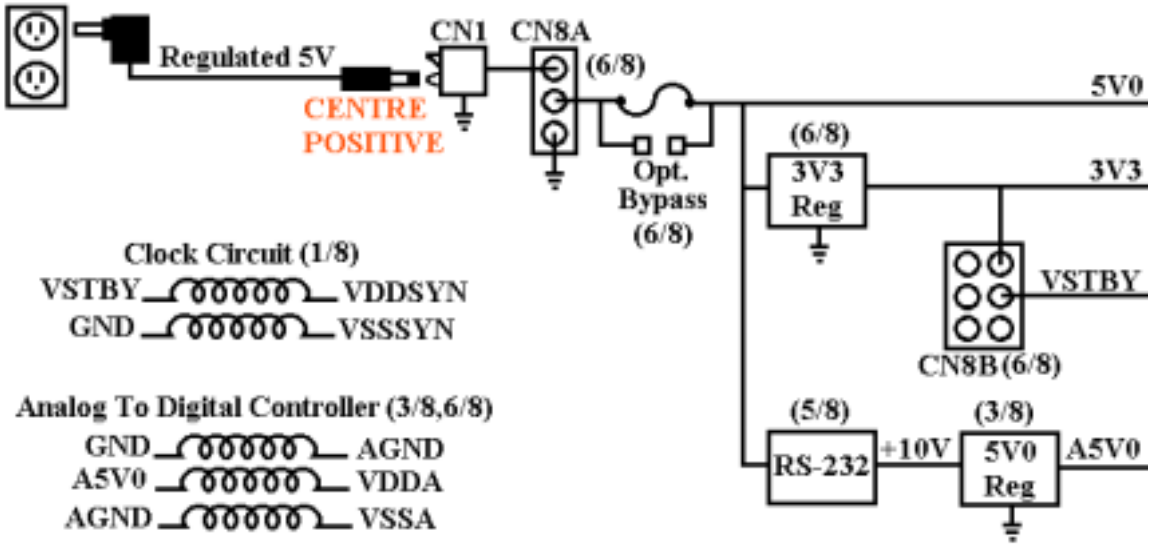


Figure 3– Power Scheme for the SS555

The table below shows the fanout of the different power levels. The double line indicates the break between MPC555 signals and peripheral components on the SS555.

5V0 – GND		3V3 – GND		A5V0 – AGND		VSTBY – GND	
Component	Description	Component	Description	Component	Description	Component	Description
VDDH	5V0 supply	VDDL	3V3 supply	VDDA	Analog 5V0	VDDSRAM	internal SRAM
		VDDI	Internal logic			KAPWR	oscillator/keep alive registers
		VDDF	Flash core			VDDSYN	oscillator power
VPP	Flash Write	VPP	Flash Read				
4053T	MODCK/IRQ switch	Ext. SRAM	Debug/Storage			CPLD	glue logic
Si920ey	CAN Transceiver	NAND Flash	Mass Storage				
ADM208	RS-232	573	Data latches				

Table 4 - Fanout of the SS555 power supplies

Power Setup

The SS555 requires a regulated 5v, 400 ma. Power supply with a CENTER POSITIVE plug connected to CN1. The SS555 is shipped with two jumpers to correctly setup the power of the SS555. The first jumper connects the 5V0 to CN1. The second jumper connects VSTBY to run directly from 3V3. PLEASE ENSURE THAT THE SETUP OF THESE JUMPERS ARE CORRECT BEFORE CONNECTING THE SS555 5v REGULATOR. These jumpers are required to take the place of the optional Low Power Daughter Boards (See SS555 Daughter Boards and Accessories) that connect to CN8A and CN8B.

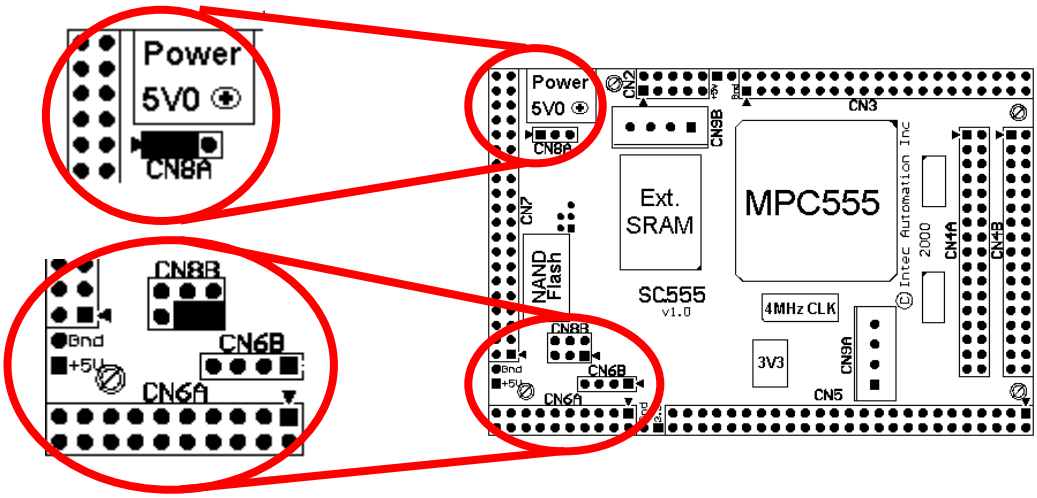


Figure 4 - Correct Placement of Power Jumpers

WARNING: To avoid damage to the SS555, only attach a coaxial center positive (2.5 mm), regulated 5v supply.

Advanced Users

The ferrite beads in the clock and analog to digital circuits shown in Figure 3 are to reduce digital noise on the power rails and to create quiet analog ground planes. The end result is more accurate readings from the analog to digital controller and a cleaner, more stable clock source for the MPC555.

There are also two power monitors onboard the SS555. If either 3V3 or VSTBY drop below 3.15 volts, the CPLD asserts *PORESET and effectively turns off the MCP555. Once the 3V3 or VSTBY returns to above 3.15 volts, *PORESET is negated and the MPC555 behaves as if power had just been connected.

Though there are several 3V3 and 5V0 pins available to the user on the SS555 board, the available current is quite small. Do not try to drive devices with large current requirements (> 100 ma) directly from the SS555 board.

11. Resets

General Information

There are 3 reset signals on the SS555, power-on reset (*PORESET), hardware reset (*HRESET), and software reset (*SRESET). All resets are 3V3 signals. Users have unlimited access to the *HRESET pin, but only advanced users should consider directly controlling the *PORESET and *SRESET signals.

Reset Pin Location

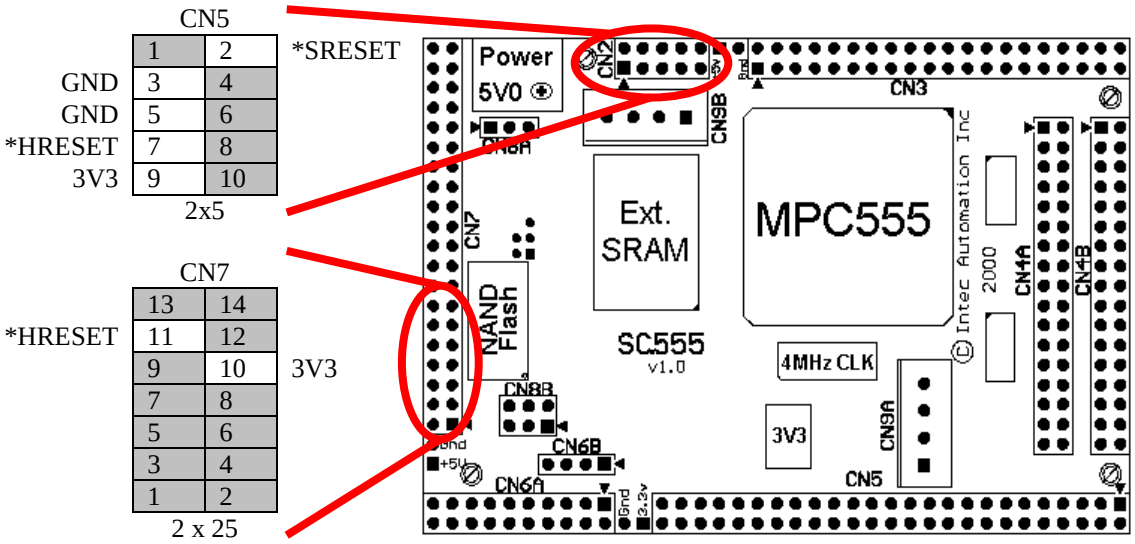


Figure 5 - Location of Reset Pins

*PORESET - Power-on Reset

During power-up, the CPLD circuit asserts *PORESET until both the 3V3 and VSTBY power monitors negate their reset signals. If, while the MPC555 is running, the voltage drops below 3.15V on either VSTBY or on 3V3, the appropriate reset line is asserted and the CPLD issues a power-on reset. If the voltage monitors assert *PORESET, it is likely that data in the external SRAM will be corrupt.

There is the possibility of creating a pin on the SS555 that can assert *PORESET. If you need this functionality, please contact Intec Automation Inc., as this will require special CPLD code, specific to your application.

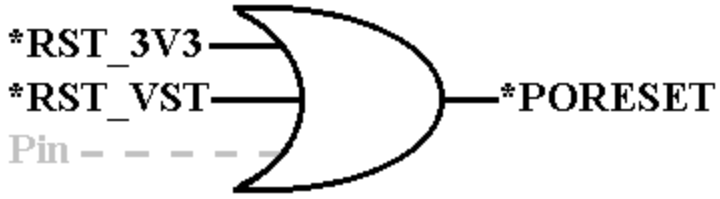


Figure 6 - Schematic representation of the CPLD *PORESET code

*HRESET - Hardware Reset

Assertion of *HRESET resets the software and the SS555 internal registers. All registers return to their default value and the hard reset configuration word is read. The source of the hard reset configuration word is either the default value (0x0000 0000) if flash memory is erased, or it is set by a register in flash EPROM. The SS555 does not support reading the external data bus for reset configuration information.

The user has access to the *HRESET signal through CN7, the Address/Data header.

The *HRESET pin on CN7 may be used as either an input or output. If the *HRESET pin is being used as an output to sense the current *HRESET level, the user can connect peripheral circuitry directly to the *HRESET pin. If the pin is to be used as an input, the user must ensure that the device toggling the pin is an open-collector configuration. If this condition is not met then the user is responsible for connecting a low forward voltage diode to the *HRESET pin. This diode ensures that the peripheral circuitry can pull the device low, but cannot drive the pin high, to avoid contention on the signal. Likewise, any debugger used with the SS555 must not drive the *HRESET pin high. EST's VisionPROBE, P&E Micros' Power PC debugger, and Macraigor's Wiggler all avoid driving the *HRESET pin high.

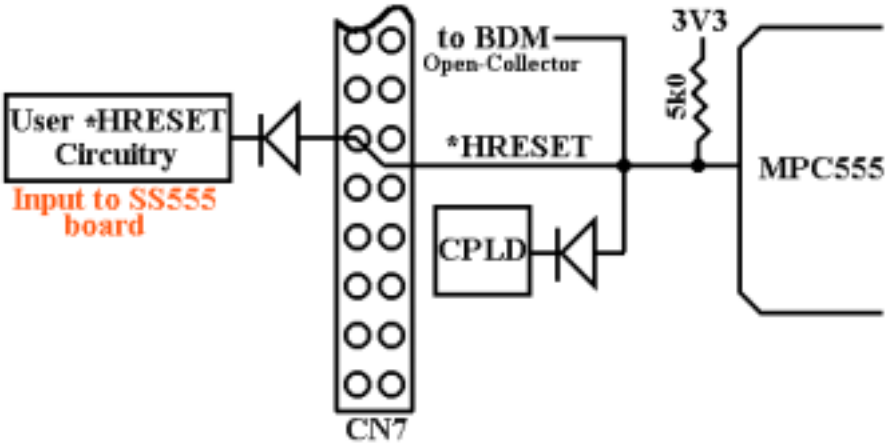


Figure 7 - *HRESET design considerations

*SRESET - Software Reset

The only access the user has to *SRESET is on the BDM port. The pin is inaccessible during debugging without some form of breakout daughter board. *SRESET resets the software without modifying the MPC555 internal register.

12. Background Debug Mode (BDM) Port

General Information

The background debug port is used to load, inspect and debug programs in the external SRAM, and finally load the working programs into Flash EPROM. BDM/JTAG ports are becoming the norm, replacing the traditional approach in which a monitor/bootstrap-loader program in EPROM is used to load, debug and execute code. BDM offers several advantages:

- The flash EPROM is dedicated entirely to user code and data. Consequently, it can be erased and programmed with impunity. No monitor code needs to be preserved.
- Jump tables associated with a monitor can be avoided, thereby simplifying and speeding programs.
- A high degree of in-circuit emulation is possible.
- Because neither UART on the SS555 is involved with debugging, they can both be used for program serial I/O. There is no monitor program obstruction, and there is no need to avoid monitor commands in serial communications.
- Control is clearly defined: When *FRZ signal is asserted (0) by the host PC, the debug program in the host PC is in control. When *FRZ is negated (1), the SS555 is in control.

There are several versions of BDM debugger that work with the SS555. The choice of which depends on the debugging software that will be used. Windriver's VisionPROBE BDM debugger is paired with their VisionCLICK debugging software. P&E's PowerPC cable is paired with their ICD-PPC software. Macraigor Systems Wiggler is paired with Software Development Systems (SDS) SingleStep debugger. For each debugger the pinout is the same, although each communicates differently between debugging software and the MPC555. Contact Intec Automation Inc. for pricing of debug cables and debuggers.

Pin Location

Pin 1 on the SS555 board is identified by a square pad and a white arrowhead on the top side of the board. Position 1 on the debug cable is identified by a colored strip (usually red), on one side of the cable. No damage will be caused to either the debugger or to the SS555 if the debug cable is attached either backwards, only on one set of pins, or is skewed. The debugger will simply not work.

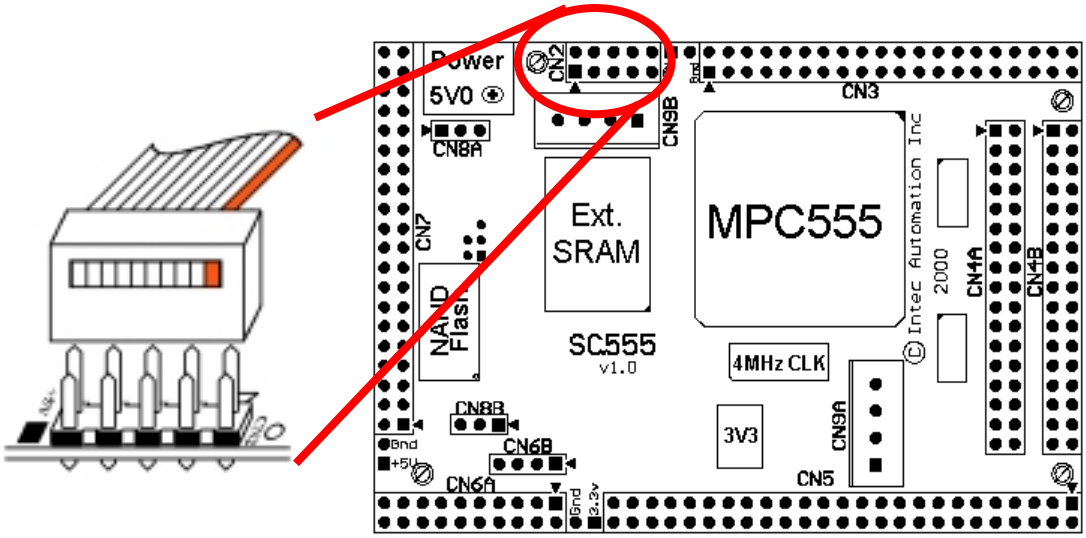


Figure 8 - BDM debug cable connecting to the SC555's BDM port

Advanced Users

There are several flavors of BDM pinouts. The SS555 is configured for maximum external bus capability. If one of the other modes is needed, the user is responsible for the creation of a suitable daughter board as well as the necessary changes to the hardware and software initialization ..neither of which is recommended or supported by Intec Automation Inc.

Maximum Debug				Maximum External Bus				Maximum I/O			
MPIO3	1	2	*SRESET	VFLS0	1	2	*SRESET	FRZ	1	2	*SRESET
GND	3	4	DSCK	GND	3	4	DSCK	GND	3	4	DSCK
GND	5	6	MPIO4	GND	5	6	VFLS1	GND	5	6	FRZ
*HRESET	7	8	DSDI	*HRESET	7	8	DSDI	*HRESET	7	8	DSDI
V+	9	10	DSDO	V+	9	10	DSDO	V+	9	10	DSDO

Table 5 - Possible BDM pinouts available on the MPC555

V+ is a sense voltage that many debuggers use to determine whether to drive the BDM pins with 3v or 5v. This should be kept at 3V for the MPC555 but, as the 555 pins are 5v tolerant, either voltage can be used. Care must be taken that the BDM debugger can also handle the extra [5v] voltage.

13. Memory

General Information

The MPC555 is a 32-bit microcontroller with contiguous memory addressing (32 address lines and 32 data lines). The address range is a continuous block of memory from 0x0000 0000 to 0xFFFF FFFF. The MPC555 provides four programmable chip select lines that can be mapped to address blocks within this address space. One of these (*CS0) is used to access the external SRAM, one (*CS3) is multiplexed into additional chip select signals (*CS3A..C), and the remaining two are brought out to header CN5. Though many different memory configurations are possible, this manual only addresses two different memory maps (one for debugging code in SRAM and one for running programs out of the flash EPROM).

Expansion Header Location

All 32 data bits are brought out to the external SRAM and the lower [most significant] 16 data bits are brought out to a peripheral expansion header. Five address lines allow access of registers in memory mapped peripheral devices. The user is able to map five 32 byte blocks of memory. All necessary control signals are provided.

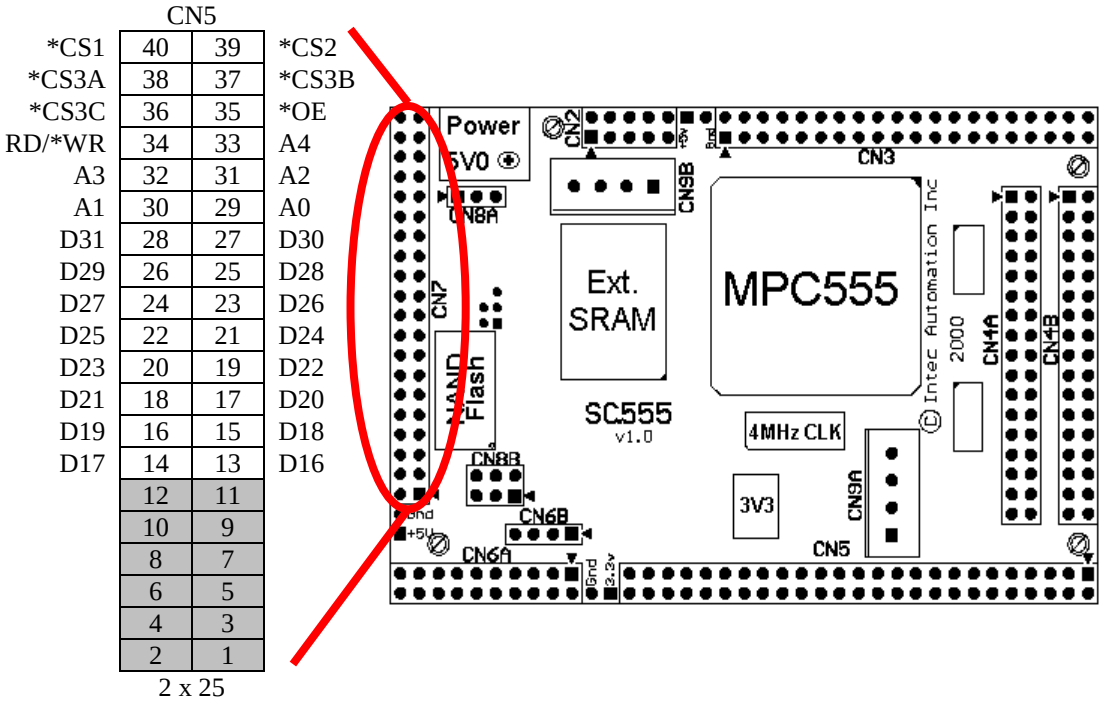


Figure 9 - Location of Peripheral Expansion Pins

Chip Select Number	Use	Max Size	Location
*CS0	External SRAM	up to 1 MB	0x0000 0000 – 0x007F FFFF (debugging) 0x0040 0000 – 0x00BF FFFF (exec flash)
*CS1	User	32 bytes	0x0300 0000 – 0xFEFF FFFF
*CS2	User	32 bytes	0x0300 0000 – 0xFEFF FFFF
*CS3	CPLD	32 bytes	0xFF00 0000 – 0xFFFF FFFF
*CS3a	User	32 bytes	0xFF00 0000 – 0xFF1F FFFF
*CS3b	User	32 bytes	0xFF20 0000 – 0xFF3F FFFF
*CS3c	User	32 bytes	0xFF40 0000 – 0xFF5F FFFF
*CS3d	RTC daughter board	32 bytes	0xFF60 0000 – 0xFF7F FFFF

Table 6 - SS555 Chip Selects

The MPC555 memory uses the little endian nomenclature. Bit 31 is the least significant bit, and bit 0 is the most significant bit.

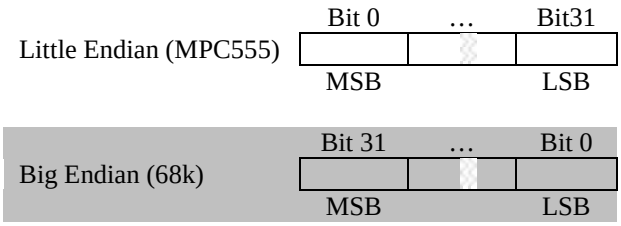


Figure 10 - Little Endian vs. Big Endian Nomenclature

*CS0 - External SRAM

*CS0 selects up to 1MB of external SRAM. The external SRAM provides two functions on the SS555. For debugging, the SRAM is mapped to the same physical address space as the MPC555 internal flash EPROM (0x0000 0000 – 0x007F FFFF). User code is debugged in the same location as when it is burned into flash. When the program is running out of flash, the external SRAM is mapped above the MPC555 internal registers and is used as a heap.

The least significant bits (31:30) on the MPC555 are not connected to the external SRAM. Byte and short word access from external SRAM is accomplished with the MPC555's Byte_Enable signals. Note that the pins from the MPC555 are reversed on the big endian SRAM.

*CS0 is set up for synchronous burst reading, which is a feature supported by the MPC555 in program space.

WARNING: To avoid damage to the SS555 External SRAM, do not set the Address/Data lines for GPIO operation.

*CS1 - *CS2

*CS1 and *CS2 have the most flexibility of the CS lines. The user has complete control over timing, port size, and position in the memory map of these two *CS lines. Setup of *CS1-*****CS2 is beyond the scope of this manual. The user must program the BRx and ORx registers to properly communicate with each

particular external device. Detailed information can be found in section 10.3 Chip-Select Timing and in Appendix G.11 Generic Timing of the MPC555 user's manual.

Additional Chip Selects

The MPC555's *CS3 is divided into 4 additional chip select lines: *CS3a, *CS3b, *CS3c by the SS555's CPLD. *CS3d is available to select the clock calendar daughter board, but can also be used as another chip select if needed.

The *CS3x chip selects are much more limited than *CS1 and *CS2. The location of each chip select within the memory map is predetermined, the chip selects can only be used as an 8-bit port, and the speed of the chip selects is relatively slow. For complete timing specifications see section CPLD of this manual.

Internal SRAM

The MPC555 has 26kB of internal SRAM. The main use for the internal SRAM is as stack space. Using the internal SRAM for data space is preferable over the External SRAM as the access time is faster. Placing the heap and stack in the internal SRAM, instead of the larger, slower external SRAM, is detailed in the IPM documentation.

Flash EPROM

When programs are fully tested in external SRAM and are error free, they can be burnt into flash EPROM. The program is then a part of the MPC555 and will be retained when power is removed. Flash has a limited life: The program can be erased and rewritten in the order of 1,000 times. Data in flash is retained for 10 years providing it is not stored above 85°C.

The flash EPROM is divided into two modules: a 256-Kbyte array and a 192-Kbyte array. Details on how to compile the program for flash EPROM and how to load the code into flash EPROM are outlined in the Flash Programming section of this manual.

In order to debug a program loaded in SRAM, the flash must first be disabled (to allow the SRAM to take its place) by clearing the FLEN (Flash Enable) bit in the IMMR (Internal Memory Mapping Register):

0 7	8 15	16 19	20	21 22	23	24 27	28 30	31
PARTNUM	MASKNUM	RSVD	FLEN	RSVD	CLES	RSVD	ISB	0
			1					

Table 7 - IMMR register

The script (i.e., startup.icd) run by the debugger on startup must clear the FLEN bit in the IMMR.

To run a program out of flash EPROM, the CMFGFG (CM Flash Configuration) register (reset configuration word in flash) must be cleared. The SS555 is set up for the cpu to check this register on negation of *HRESET. If it is erased ($FFFF FFFF), the CPU will use the default reset configuration word of $0000 0000, which has FLEN cleared, indicating that flash is disabled and the SS555 will not run the program in flash. If CMFGFG is cleared, then the cleared *HC (has configuration) bit results in the FLEN bit being set in the IMMR, allowing the cpu to read and execute the program in flash.

14. Flash Programming

Running a program automatically out of flash, when the SS555 is powered on is a three step process:

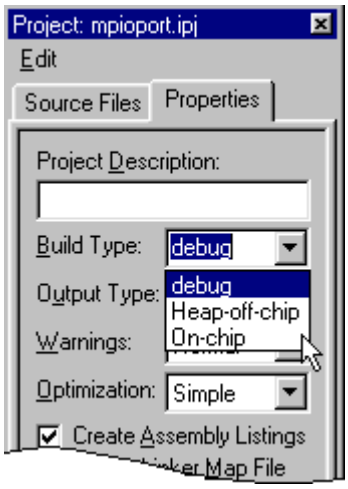
1. The program is first compiled for the desired memory map: Heap on_chip or heap off_chip (in external SRAM).
2. The program is burned into flash.
3. Finally, the internal reset configuration word (CMFCFG) must be programmed into the start of the shadow flash.

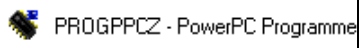
Once code has been debugged and tested, the finished program can be loaded into flash. In the Intec Project Manager the user must choose the desired build type. The current choices are debug (in SRAM), [heap] on-chip, and [heap] off-chip, with the former only being used in very special cases (speed or quietness).

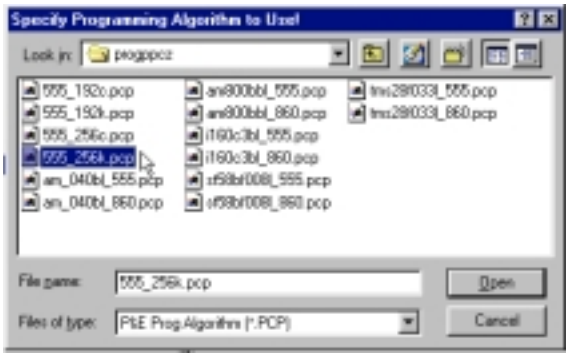
	Flash	Start of Stack	Stack Size	Start of Data	Start of Heap	End of Heap
Debug	Disabled	0x0040 0000	4k	follows program		
On-Chip	Enabled	0x0040 0000	4k	0x3f9800		
Heap Off-Chip	Enabled	0x0040 0000	4k	0x3f9800	0x0040 0000	0x0044 0000

Table 8 - Build Type Characteristics

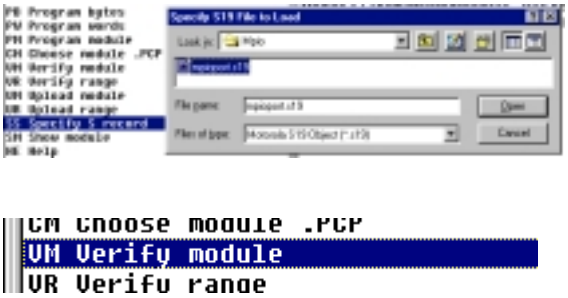
The Stack grows downwards numerically with 0x0040 0000 being the top of the stack (the first valid stack address is 0x3F FFFC). The Heap grows upwards with 0x3f9800 (on-chip) or 0x0040 0000 (off_chip) being the bottom of the heap.

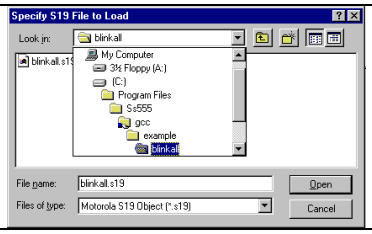
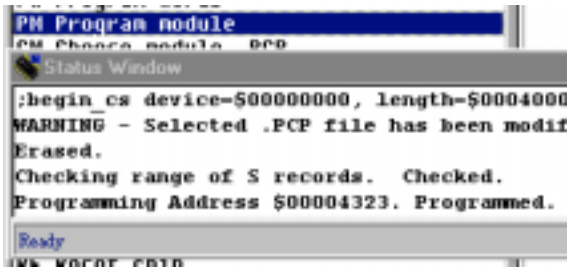
Select the proper build type	
To select the desired build type click on the properties tab under the open project. Click on the pull-down menu for Build Type and select either Heap off-chip or On-chip configuration. On-chip is the faster of the two choices. The Heap off-chip should be used is if more than 26k of data and stack space is required. Once this is done recompile the project. The .bin file is generated for systems using VisionCLICK and the .695 file is generated for systems using PROGPPC. **Choose Heap off-chip or on-chip and press shift+F9 (rebuild all).**	

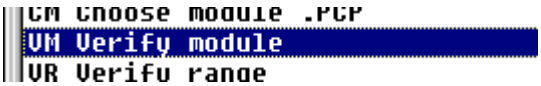
Open the Flash Programmer (P&E Micros Suite)	
Open the flash programmer - PROGPPC.	
Connect the debugger to the SS555 board. Make sure that the LPT port is correct and the CPU type and the MCU internal bus frequency are set for auto-detect. The parallel port delay can be increased if communication is erratic. If the delay needs to be set to above 7, there may be a problem with the parallel cable. Try connecting the P&E debug cable directly to the LPT port on the PC to check if there is a problem with the extension cable. **Click OK.**	

Select the correct programming algorithm for programming the 256k flash	
The Flash is divided into two blocks. IPM currently only sets the program counter to 0x100. This means that the program must be loaded into the first 256k block of flash. The 555_xxxk.pcp files allow the programming of the flash module. This is where the program and initialized data resides. **Highlight the 555_256k.pcp file and click OK.** A "Warning: Selected .PCP device…" is normal.	
Choose the correct base address. The correct value is the begin cs_device number. In the case of 555_256k.pcp the block starts at 0x0000 0000. Please note that although the program starts at 0x0100 the base address is still set at 0. **Type 0 and press OK.**	

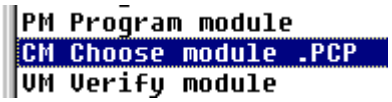
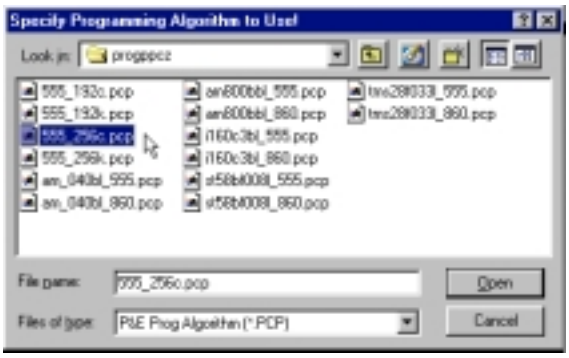
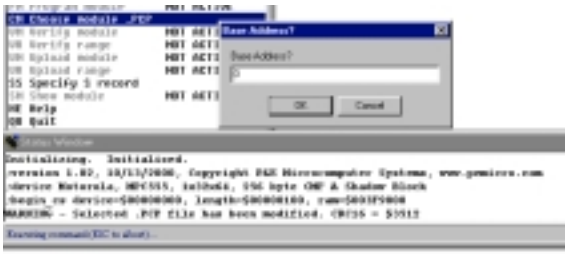
Once the .pcp file is selected the SS555 is initialized. During the initialization, the EPEE (programming enable) pin is set high by the CPLD. This in turn steps the programming voltage to VPP to 5v. Flash programming is enabled until either the power is turned off or the user manually negates EPEE.

Verify current contents of the Flash	
Check to see if the module is erased. If the module is not erased the following message will appear: Module word not erased at $xxxx xxxx word is xxxx xxxx. If the module is erased the message “Erased” will appear. Skip the rest of the steps in this panel following steps and go directly to programming the CMFCFG. **double click on blank check module**	
If the module is already programmed the next step is to verify that the correct program is residing in flash. Specify S record and set the path to the file that was generated in Intec Project Manager in step 1. If the two files are the same the message “Verified.” will appear. If the files differ, the address of the first inconsistent byte is displayed. **Double click on Specify S record and find the appropriate .s19 file and click ok. Double click on Verify Module.**	
If the files do not verify, erase the module. **Double click Erase module.**	

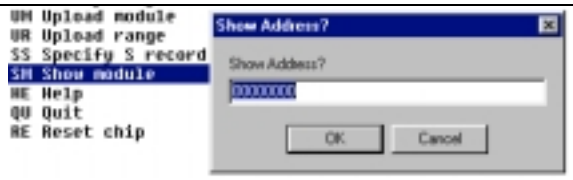
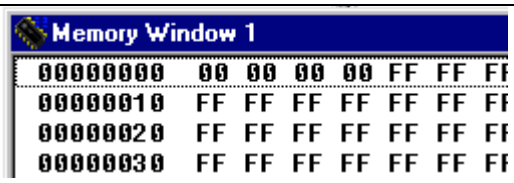
Burn the program into flash	
Choose the program to be burned into flash with the **Specify S Record** command.	
Burn the .s19 file into the flash. Again it is imperative that the flash be erased (all registers=FF) before attempting to write to flash. This process should take less than 15 sec. on rev G MPC555s and less than 80 sec. on higher revisions. If the Programming Address doesn’t change every second or so, cancel the programming by hitting ESCAPE. **Double click on program module.**	

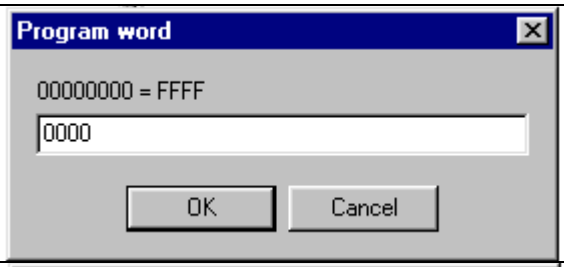
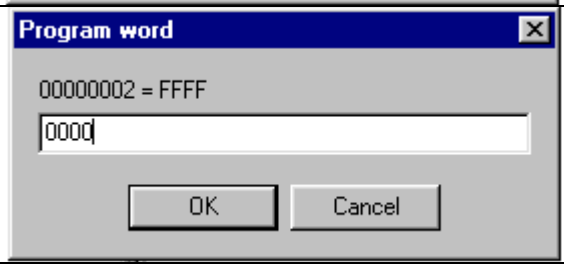
Verify that the module programmed correctly. This verification may seem unnecessary but is a good habit to develop. **Double click on verify module.**	

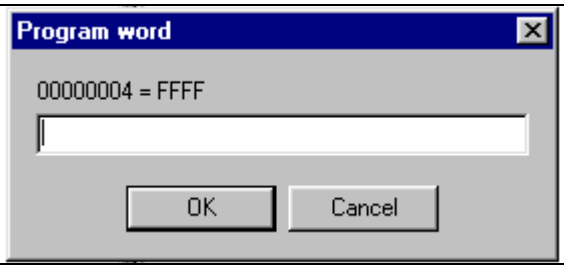
The program is now burned into flash. The final step is to configure the internal flash EPROM hardware reset configuration word. In the absence of a reset configuration word programmed in shadow flash, the SS555 will not will start program execution out of flash on power-up.

Select the correct programming algorithm for programming the shadow information	
Change the programming algorithm by double clicking on Choose Module .PCP.	
The 555_xxxc.pcp files allow the programming of the shadow information. This is where the internal reset configuration word resides. **Highlight the 555_256c.pcp file and click OK.**	
Choose the correct base address. In the case of 555_256c.pcp the block starts at 0x0000 0000. **Type 0 and press OK.**	

Verify current contents of the Shadow Information	
Check to see if the module is erased. If the module is not erased the following message will appear: Module word not erased at $xxxx xxxx word is xxxx xxxx. If the module is erased the message “Erased.” will appear: Skip the erasing steps in this panel and go directly to programming the CMFCFG. **double click on blank check module**	

If the module word is not erased and the address is less than or equal to 0x0000 0008, check the value of the internal reset configuration word (CMFCFG). **Double click on show module, type 0000 0000 and click OK.**	
Ensure that the first 4 bytes of the CMFCFG are zero. Bit 20 is the *HC bit and must be set to zero in order to boot out of flash. **Make sure that 000000000 = 00 00 00 00. Close the window.**	

Programming the CMFCFG	
If the CMFCFG is not properly programmed, erase the module. This not only erases the shadow information but also the program in flash. This is not true in reverse. Erasing the program does not erase the shadow information. **Double click Erase module.**	
Verify that the module correctly erased by clicking on blank check module again. The message "Erased." will appear. **Double click Blank check module.**	
If the module is properly erased, write the internal hard reset configuration word automatically. Click on either program words or program bytes and enter the starting address of 0x0000 0000. This corresponds to the location of the CMFCFG register. **Double click on Program words, type 0 and hit OK.**	
Determine the desired value for the reset configuration word and enter the high byte first. **Type in the desired value and press OK**	
Determine the desired value for the reset configuration word and enter the low byte. Bit 20 must be zero to enable the flash EPROM and allow for the free running of programs. **Type in the desired value and press OK**	

Click cancel to stop programming the shadow information. **Click Cancel.**	

The shadow information and the CMFCFG register should now be correctly programmed. This can be verified by showing the module and inspecting that the first 4 bytes are what were programmed.
The flash is now properly programmed. The debug cable and power can now be removed. Once the power is reconnected the program that was loaded into flash will run without the need to download code from the PC.

15. Interrupt Request (IRQ)

General Information

10 pin – 3 Volt interrupt request port or 5V0 GPIO port. *IRQ0 is the non-maskable interrupt and also is the highest priority. *IRQ1 to *IRQ7 are individually maskable with *IRQ7 having the lowest priority. Each interrupt corresponds to it's own interrupt level in the Interrupt Vector Table. To indicate that an interrupt has been requested, the MPC555 asserts *IRQOUT. The FRZ pin is asserted whenever the MPC555 is under the control of the debugger. For more information see section 6.4 Interrupt Controller in the MPC555 user's manual.

The port can also be configured for 5V0 GPIO in two distinct blocks. IRQ0:IRQ5 can be set for GPIO0:5, and FRZ:*IRQOUT can be separately set for GPIO6:7. Reads and writes to the GPIO port are simultaneous.

IRQ Pin location

CN5

	1	2	
	3	4	
	5	6	
	7	8	
	9	10	
	11	12	
	13	14	
	15	16	
	17	18	
	19	20	
	21	22	
	23	24	
	25	26	
	27	28	
	29	30	
	31	32	
	33	34	
	35	36	
	37	38	
FRZ	39	40	*IRQOUT
*IRQ0	41	42	*IRQ1
*IRQ2	43	44	*IRQ3
*IRQ4	45	46	*IRQ5
*IRQ6	47	48	*IRQ7
3V3	49	50	GND

2 x 25

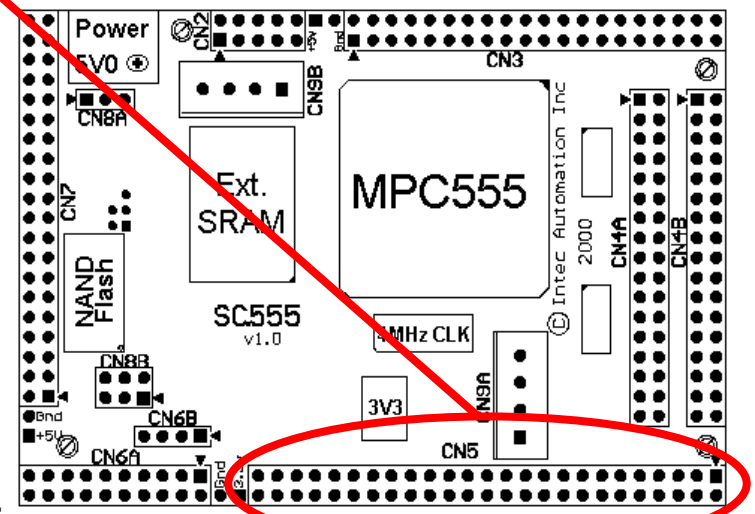


Figure 11 - Location of IRQ pins

Special Note

GPIO pins are not continuous. Be careful that you do not mistake *IRQ6 for GPIO6 and *IRQ7 for GPIO7.

FRZ	39	40	*IRQOUT	GPIO6	39	40	GPIO7
*IRQ0	41	42	*IRQ1	GPIO0	41	42	GPIO1
*IRQ2	43	44	*IRQ3	GPIO2	43	44	GPIO3
*IRQ4	45	46	*IRQ5	GPIO4	45	46	GPIO5
*IRQ6	47	48	*IRQ7	*IRQ6	47	48	*IRQ7

Table 9 - IRQ pinout versus GPIO pinout

Advanced Users

*IRQ5 - *IRQ7 are used to properly configure the MPC555 clock when the board exits power on reset. This is currently handled by a multiplexer that can be seen on schematic 1/1. *IRQ5 to *IRQ7 correspond to MODCK1 to MODCK3. The multiplexer disconnects the *IRQ pins during power up. This means that no externally generated IRQs will reach the MPC555 while *PORESET is asserted. For more information see section 8.6 MPC555 Internal Clock Signals.

MODCK[1:3]	LME	MF + 1	PITCLK Division	TMBCLK Division	Division
000	0	513	4	4	Used for testing purposes.
001	0	1	256	16	Normal operation, PLL enabled. Main timing reference is freq(OSCM) = 20 MHz. Limp mode disabled.
010	1	5	256	4	Normal operation, PLL enabled. Main timing reference is freq(OSCM) = 4 MHz. Limp mode enabled.
011	1	1	256	16	Normal operation, PLL enabled. Main timing reference is freq(OSCM) = 20 MHz. Limp mode enabled.
100	0	1	256	16	Normal operation, PLL enabled. 1:1 Mode. freqclkout(max) = freq(EXTCLK). Limp mode disabled.
101	0	5	256	4	Normal operation, PLL enabled. Main timing reference is freq(EXTCLK) = 3-5 MHz. Limp mode disabled.
110	1	1	256	16	Normal operation, PLL enabled. 1:1 Mode. freqclkout(max) = freq(EXTCLK). Limp mode enabled.

Table 10 - *PORESET clock configuration (MODCK1:3)

16. Pulse Width Modulation (PWM)

General Description

There are eight PWM pins on the MPC555, each independently configurable. These 5V0 pins are capable of either PWM output or general purpose input/output operation. The user has control over period, high time (duty cycle) of the output waveform. The period can be adjusted from 50ns to 3.2µs using a 40MHz f_{sys}. Decreasing the system frequency can further extend the period of the PWM. The high time can vary from 0% to 100%. The PWM pins are capable of a maximum of 16 bits of resolution. The resolution decreases as the period shortens.

PWM Pin location

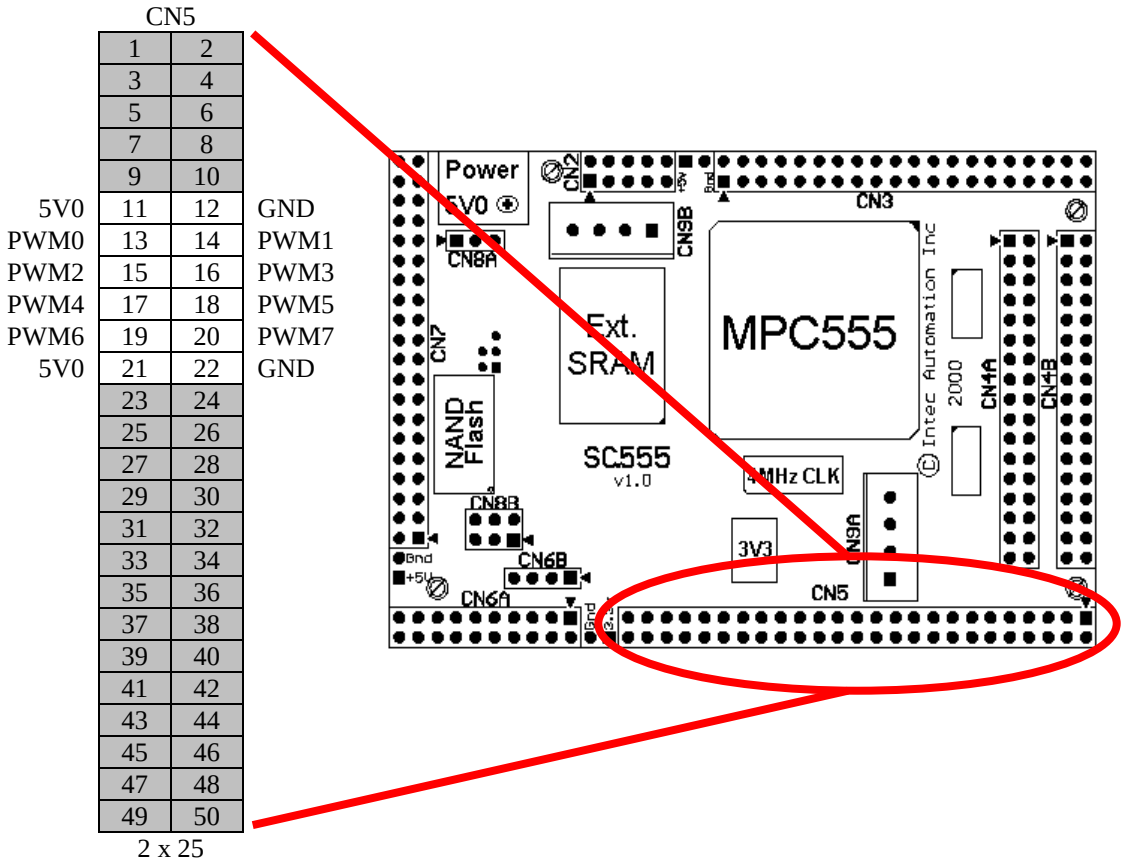


Figure 12 - Location of PWM pins

Period and Duty Cycle Calculation

The period for the PWM originates with the system frequency (f_{sys}). f_{sys} is first divided for the entire MIOS module. The PWM, DA, and the MIOS counters share this division factor (OFF, ÷2 to ÷16) . Each PWM pin then further divides this frequency (÷1 to ÷256). Finally, the number of tics multiplied by the PWM clock period to produce the PWM period (2 to 65536).

$$PWMx \text{ period} = \left[\frac{(IMB \text{ division factor}) \cdot (MIOS \text{ clock prescaler}) \cdot (\text{PWMx clock prescaler})}{f_{sys}} \right] \cdot (\#\text{ of tics})$$

Equation 1 - PWM period

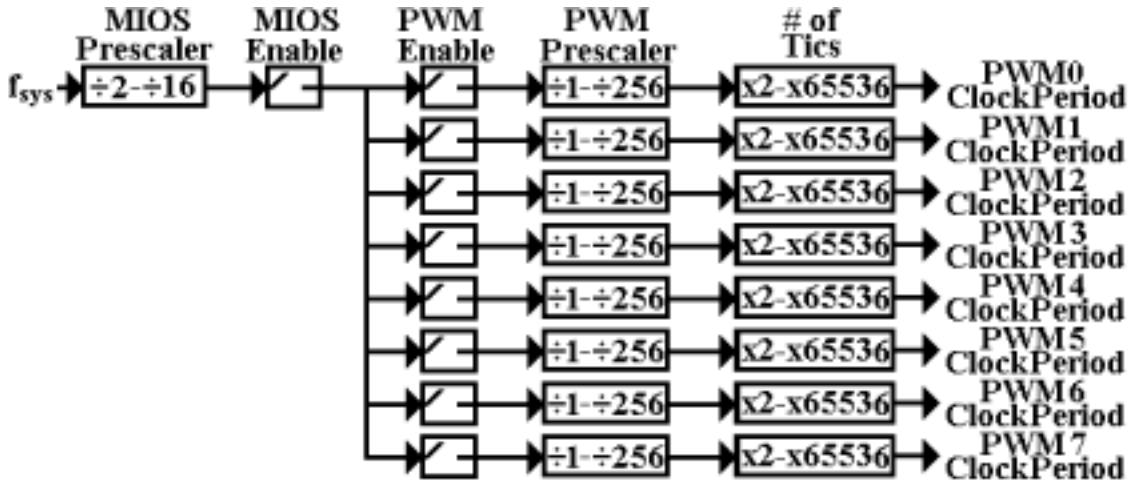


Figure 13 - Origin of the PWM clock period

The duty cycle is calculated as the number of tics in the high time of the PWM divided by the number of tics in the period of the PWM. If the number of tics high is greater than or equal to the number of tics in the PWM, the PWM is always high.

$$PWMx \text{ duty cycle} = \frac{(\#\text{ of tics high})}{(\#\text{ of tics})}$$

Equation 2 - PWM Duty Cycle

Example 1- Determining the correct values for the PWM functions

$$\left.\begin{array}{l} \text{desired PWMx period} = 10\mu s \\ \textit{desired} \text{ duty cycle} = 40\% \\ f_{sys} = \textbf{40MHz} \\ \text{MIOS Clock Prescaler} = \textbf{2} \end{array}\right\} \begin{array}{c} f_{sys} \cdot \text{PWMx period} = 40\text{MHz} * 10\mu\text{s} = 400 \\ 400 = (MIOS \text{ clock prescaler}) \cdot (\text{PWMx Clock Prescaler}) \cdot (\#\text{ of tics}) \\ (\text{PWMx Clock Prescaler}) \cdot (\#\text{ of tics}) = \dfrac{400}{(\text{MIOS Clock Prescaler})} \\ (\text{PWMx Clock Prescaler}) \cdot (\#\text{ of tics}) = \dfrac{400}{2} = 200 \end{array}$$

$200 < (\text{Max } \#\text{ of tics}) \ \therefore \text{PWMx Clock Prescaler} = \textbf{1}$

$\#\text{ of tics} = \textbf{200}$

$\#\text{ of tics high} = (\#\text{ of tics}) \cdot (PWMx \text{ duty cycle}) = 200 \cdot 40\% = \textbf{80}$

Set PWM 0 with the values found in the above example

```
sysclk_set( MHz_40);        // fsys = 40MHz (if fsys is the default, this can be removed)
MIOS_init( 2 );             // MIOS Clock Prescaler = 2
PWM_init( 0 , PWM );        // Initialize PWM 0 for PWM functionality (versus GPIO)
PWM_period( 0 , 1 , 200 );  // PWM 0's clock prescaler = 1 and the # of tics = 200
PWM_pulse( 0 , 80 );        // PWM 0's # of tics high = 80
```

The DA and MIOS counters are dependent on the setup of the MIOS clock prescaler. If the MIOS Clock Prescaler is changed, the calculations for other PWM pins, DA and MIOS counters must be recalculated. If the PWMx Clock Prescaler value is kept as small as possible and the number of tics in the period is increased, the resolution of the clock is increased.

Below is given a table of possible PWM frequencies at values of f_{sys} values supported by the sysclk_set() function.

	PWM Frequencies					
	2 bits of resolution		4 bits of resolution		8 bits of resolution	
	2 tics		16 tics		256 tics	
fsys	max	min	max	min	max	min
40 MHz	10.0 MHz	2.44 kHz	1.25MHz	305 Hz	78.1 kHz	19.1 Hz
32 MHz	8.00 MHz	1.95 kHz	1.00 MHz	244 Hz	62.5 kHz	15.3Hz
20 MHz	5.00 MHz	1.22 kHz	625 kHz	153 Hz	39.1 kHz	9.53 Hz
16MHz	4.00 MHz	976 Hz	500 kHz	122 Hz	31.3 kHz	7.62 Hz
1 MHz	250 kHz	61.0 Hz	31.3 kHz	7.63 Hz	1.95 kHz	0.477 Hz

	PWM Frequencies					
	10 bits of resolution		12 bits of resolution		16 bits of resolution	
	1024 tics		4096 tics		65536 tics	
fsys	max	min	max	min	max	min
40 MHz	19.5 kHz	4.77 Hz	4.88 kHz	1.19 Hz	305 Hz	0.0745 Hz
32 MHz	15.6 kHz	3.81 Hz	3.91 kHz	0.954 Hz	244 Hz	0.059 Hz
20 MHz	9.77 kHz	2.38 Hz	2.44 kHz	0.596 Hz	153 Hz	0.037 Hz
16MHz	7.81 kHz	1.91 Hz	1.95 kHz	0.477 Hz	122 Hz	0.030 Hz
1 MHz	488 Hz	0.119 Hz	122 Hz	0.0298 Hz	7.63 Hz	0.00186 Hz

Table 11 - Some Possible PWM frequencies and resolutions

max – (IMB division factor) * (MIOS clock prescaler)*(PWMx clock prescaler)=2*16*256=8192
min – (IMB division factor) * (MIOS clock prescaler)*(PWMx clock prescaler)=1*2*1=2

17. Queued Analog to Digital Controller(QADC)

General Information

There are two QADC modules on the MPC555 (Port A/Port B), each containing 16 analog to digital input pins and one external trigger pin. The QADC samples once every 7µs (QCLK= 2MHz) and has a 10 bit resolution. The QADC also supports GPIO operation on Port A, and GPI operation on Port B.

Pin Location

CN4A

AN0_A	1	2
AN1_A	3	4
AN2_A	5	6
AN3_A	7	8
AN4_A	9	10
AN5_A	11	12
AN6_A	13	14
AN7_A	15	16
AN8_A	17	18
AN9_A	19	20
AN10_A	21	22
AN11_A	23	24
AN12_A	25	26
AN13_A	27	28
AN14_A	29	30
AN15_A	31	32
	33	34

2x17

CN4B

AN0_B	1	2
AN1_B	3	4
AN2_B	5	6
AN3_B	7	8
AN4_B	9	10
AN5_B	11	12
AN6_B	13	14
AN7_B	15	16
AN8_B	17	18
AN9_B	19	20
AN10_B	21	22
AN11_B	23	24
AN12_B	25	26
AN13_B	27	28
AN14_B	29	30
AN15_B	31	32
	33	34

2x17

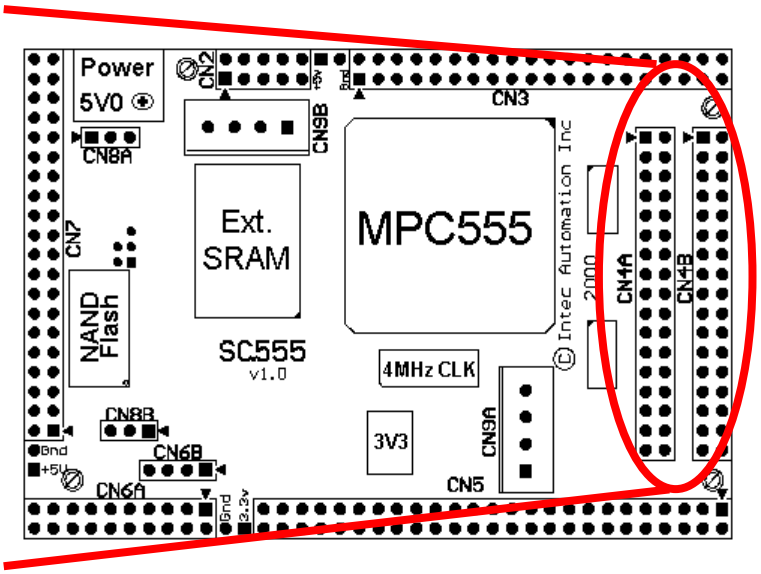


Figure 14 - Location of QADC pins

Special Note

The RTL555 considers the analog section in terms of pins, whereas the Motorola manual considers the analog section in terms of channels. A table is given below listing the channel number for each pin. When using the above functions the pin number should be used.

Pin #	Channel	Pin #	Channel	Pin #	Channel	Pin #	Channel
0	0	4	48	8	52	12	56
1	1	5	49	9	53	13	57
2	2	6	50	10	54	14	58
3	3	7	51	11	55	15	59

Table 12 - RTL555 vs. MPC555 ADC naming convention

Advanced Users

Specifying the QCLK

The QCLK is the clock that runs the QADC. The clock specific to the QADC section is called the QCLK and can be specified independently for each QADC port. The QCLK has an operating range from 0.5MHz to 3MHz. The RTL555 runs the QCLK at 2MHz if using the sysclk_set(MHz_x) function. QCLK is make up of the clock high time and the clock low time. The QCLK should have as close to a 50% duty cycle as possible.

If the user either wants to run the QADC at a different frequency, or wants to run fsys at a frequency not supported by the sysclk_set() function, the user must explicitly set QCLK. A partial list of values to be written to ADC_clk_set() is given below.

	40MHz			32MHz			20MHz		
fqclk	PSH+PSL+2	PSH+1	PSL+1	PSH+PSL+2	PSH+1	PSL+1	PSH+PSL+2	PSH+1	PSL+1
.5MHz	80	NOT POSSIBLE		64	NOT POSSIBLE		40	32	8
1MHz	40	32	8	32	24	8	20	12	8
1.5MHz	26.666	18.666	8	21.333	13.333	8	13.333	7.333	6
2MHz	**20**	**12**	**8**	**16**	**8**	**8**	**10**	**5**	**5**
2.5MHz	16	8	8	12.8	7	6	8	4	4
3MHz	13.333	7.333	6	10.666	5.666	5	6.666	3.666	3

	16MHz			1MHz			<1MHz		
fqclk	PSH+PSL+2	PSH+1	PSL+1	PSH+PSL+2	PSH+1	PSL+1	PSH+PSL+2	PSH+1	PSL+1
.5MHz	32	23	7	2	1	1	2	NOT POSSIBLE	
1MHz	16	7	7	1	NOT POSSIBLE		1	NOT POSSIBLE	
1.5MHz	10.666	4.666	4	1.333	NOT POSSIBLE		1.333	NOT POSSIBLE	
2MHz	**8**	**3**	**3**	**0.5**	NOT POSSIBLE		**0.5**	NOT POSSIBLE	
2.5MHz	6.4	2.4	2	0.4	NOT POSSIBLE		0.4	NOT POSSIBLE	
3MHz	5.333	2	1.333	0.333	NOT POSSIBLE		0.333	NOT POSSIBLE	

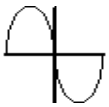
Table 13 - Common calculations for QCLK

$$PSH + PSL = \left(\frac{f_{sys}}{f_{qclk}} \right) - 2 \text{ where} \begin{cases} 1 < \text{PSH} < 33 \\ 1 < \text{PSL} < 8 \\ \text{PSH} \approx \text{PSL} \end{cases}$$

For more information see the 13.10.4 QADC Clock Generation in the Motorola MPC555 user's manual.

Multiplexing the QADC lines

The QADC can be externally multiplexed, allowing 41 separately addressable ADC lines per port. This gives a maximum of 82 possible ADC inputs on the MPC555. Though this type of multiplexing is possible with the SS555, it is not explicitly supported. The user is responsible for the hardware multiplexing of the ADC pins, and the writing of appropriate software functions to access the multiplexed pins. The RTL555 cannot be used to communicate with multiplexed analog inputs. For more information on multiplexing the ADC, see section 13.7 in the MPC555 user's manual.



There is one fundamental difference between the Motorola manual and the RTL555. The RTL555 considers the analog section in terms of pins, whereas the Motorola manual considers the analog section in terms of channels. A table is given below listing the channel number for each pin. The reason for the gap in channel numbers is due to the multiplexing ability of the QADC. The SS555 doesn't explicitly support the multiplexing of the QADC. See section 13.7 in the MPC555 user's manual for multiplexing details.

Pin	Channel	Pin	Channel	Pin	Channel	Pin	Channel
0	0	4	48	8	52	12	56
1	1	5	49	9	53	13	57
2	2	6	50	10	54	14	58
3	3	7	51	11	55	15	59

Table 14 - RTL555 vs. MPC555 Naming Convention for ADC

	40MHz			32MHz			20MHz			16MHz			1MHz*			<1MHz
fqclk	PSH +PSL+2	PSH+ 1	PSL +1	PSH +PSL+2	PSH +1	PSL +1	PSH+ PSL+2	PSH +1	PSL +1	PSH+ PSL+2	PSH +1	PSL +1	PSH+ PSL+2	PSH +1	PSL+ 1	NOT POSSIBLE
2MHz	20	12	8	16	8	8	10	5	5	8	4	4	**2**	**1**	**1**	

Table 15 - Calculation used with the Generic sysclk_set() function

*At 1MHz, fqclk has a maximum possible frequency of 0.5MHz. It does NOT run at 2MHz. Lower than 1MHz, the fqclk falls outside the minimum frequency given in Appendix G – Electrical Specifications of the MPC555 User's Manual.

18. Timers (PIT/RTC/DEC/TB/WDT)

There are 5 timers on the MPC555. These timers are the periodic interrupt timer (PIT), real time clock (RTC), decrementer (DEC), the time base clock (TB), and the watchdog timer (WDT). There are a few dependencies between clocks that are handled by combined PIT/RTC functions, and combined DEC/TB functions. Though there are many applications of the timers, a typical example for each is given below.

The PIT is normally used to schedule tasks. The PIT is loaded with an initial value, generates an interrupt, performs a task and then continues with the mainline code. Once the PIT interrupt is run, the PIT is once again loaded with its initial count value and the process repeats.

The RTC is used to keep track of seconds elapsed. This doesn't keep track of date and time, but rather seconds since the RTC was enabled. In order to keep track of date and time, please refer to Appendix B – Daughter Boards.

The RTC can interrupt once every second, or once a day/week/year. The RTC and PIT share the same clock source and care must be taken when using both the PIT and RTC.

The DEC is usually used to wake up the processor from low power modes. The DEC is loaded with a sleep time and is started, the MPC555 is put into sleep mode and the chip consumes less power. When the decrementer times out, the DEC interrupt wakes up the processor. This can also be used in conjunction with Intec's low power daughter board (see Appendix B – Daughter Boards). The DEC generates an interrupt whenever the MSB changes from a zero to a 1. This usually occurs when the DEC rolls over from 0x0000 0000 to 0xFFFF FFFF. Care must be taken that an unwanted interrupt doesn't occur during the setting of the DEC register. If the DEC equals 0x7FFF FFFF and user sets DEC=0x8000 0000 the MSB has changed from a zero to a 1 and an interrupt will occur.

0x7FFF FFFF **0**111 1111 1111 1111 1111 1111 1111 1111
0x8000 0000 **1**000 0000 0000 0000 0000 0000 0000 0000 (this will generate an interrupt)

The TB is normally used to wakeup the MPC555 for a short period of time. The TB has two reference registers (REF0 and REF1) that can interrupt the SS555. When the first interrupt (REF0) is generated, the SS555 is put to sleep. The TB continues to increment until the second interrupt (REF1) is generated, at which point the SS555 is set to run at full speed. This can also be used in conjunction with Intec's low power daughter board (see Appendix B – Daughter Boards).

The WDT is used to make sure that the processor doesn't enter an endless loop. The WDT is programmed to timeout after a given period of time. The user, usually in the mainline program, gives the WDT a kick periodically to make sure that the watchdog doesn't timeout. Though this function is very useful, it should not be used frivolously. Special care must be taken to ensure that the WDT doesn't mistakenly timeout during long functions or during the worst case timing of interrupts.

	PIT	RTC	DEC	TB	WDT
Counter Direction	down	up	down	up	down
Interrupt	YES	YES	YES	YES	YES
Resolution	µs	s	µ s	µ s	µ s
Min Time-(@40MHz)	1µs / 64 µs	1s	1 µs	1 µ s	25ns
Max Time-(@40MHz)	4.19 seconds	133 years	13 hours	13 hours	3.35s

Table 16 - Timer Properties

19. CPLD

Introduction

The in-situ programmable CPLD provides the glue logic and control for several sections of the SS555 board. The CPLD performs the following tasks:

- controls power-on reset
- generates additional chip select signals
- interfaces to the external NAND flash
- controls the hardware handshaking of COM1
- controls the optional real time clock daughter board
- enables/disables writes to the internal MPC555 FLASH EPROM
- controls the optional low power daughter board

The purpose of this section is to familiarize the user with the hardware, programming of the CPLD, and describe the default CPLD code. Intec Automation Inc. can either create specialty software or provide a quickstart CPLD package. Please contact us for details and pricing.

Hardware

The CPLD hardware is located on page 8/8 of the schematic diagrams. The CPLD is a 3V device powered by VSTBY so that it can operate in low power modes. The CPLD can be programmed in either low power mode (VSTBY=ON 3V3=OFF 5V0=OFF), or when the MPC555 is running in full power mode (VSTBY=ON 3V3=ON 5V0=ON).

Interfacing to the CPLD

The CPLD is divided into manual and automatic functions. The power-on reset and low power control are automatic functions, being run without any user interaction. The following sections will describe the manual functions that users are able to control. Control is gained by reading and writing certain registers in the MPC555 memory map. The CPLD is selected by *CS3 from the MPC555. By default, *CS3 is mapped from 0xFF00 0000 to 0xFFFF FFFF.

Address Rangee	Functionality
FFFF FFFF	Nand Flash control signals.
FFE0 0000	
FFDF FFFF	Nand Flash control signals.
FFC0 0000	
FFBF FFFF	Write Protection for the Internal Flash EPROM and the external NAND Flash.
FFA0 0000	
FF9F FFFF	Hardware Handshaking.
FF80 0000	
FF7F FFFF	External clock calendar chip. This functionality is currently
FF60 0000	
FF5F FFFF	Chip select 3c. 32 bytes of 8/16 bit addressable memory.
FF40 0000	
FF3F FFFF	Chip select 3b. 32 bytes of 8/16 bit addressable memory.
FF20 0000	
FF1F FFFF	Chip select 3a. 32 bytes of 8/16 bit addressable memory.
FF00 0000	

Table 17 - CPLD functionality

Extra CS signals

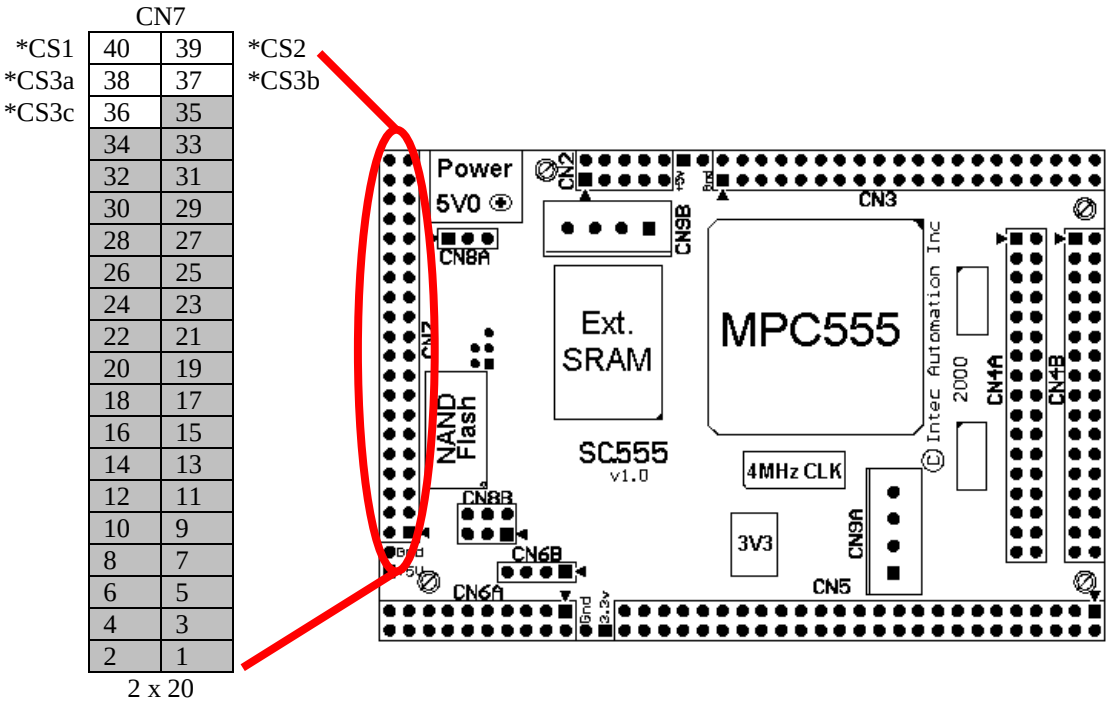


Figure 15 - Location of Additional Chip Select Pins

One of the weaknesses of the MPC555 is its lack of chip select lines. CS0 is being used to select the external SRAM and CS3 is used to select the CPLD. This leaves only 2 remaining CS lines for the user. To remedy this shortfall, the SS555 decodes CS3 into three extra chip select lines for the user, CS3a, CS3b, and CS3c. On the SS555, the user has access to 16 data bits and five address lines. This means that the user has five 32 byte blocks of individually selectable 8 or 16 bit data space. The user should treat the CS3a/b/c signals as they would any other chip select. A read or write to the appropriate memory location will assert the corresponding chip select signal.

CS3a	CS3b	CS3c	
FF1F FFFF	FF1F FFFF	FF1F FFFF	32 bytes are repeated due to partial address decoding. Partial address decoding has multiple addresses referring to the same memory location. i.e. CS3a - FFFF FFE0=FF00 0040=FF00 0020=FF00 0000
..	..	..	
FF00 0020	FF00 0020	FF00 0020	
FF00 001F	FF00 001F	FF00 001F	32 bytes of byte/word addressable memory
FF00 001E	FF00 001E	FF00 001E	
…	…	…	
FF00 0001	FF00 0001	FF00 0001	
FF00 0000	FF00 0000	FF00 0000	

Table 18 -Partial Address Decoding of CS3a/CS3b/CS3c

External Real Time Clock Interface

CN8B

6	5	*CE
4	3	CLE
2	1	*CS3d

2 x 3

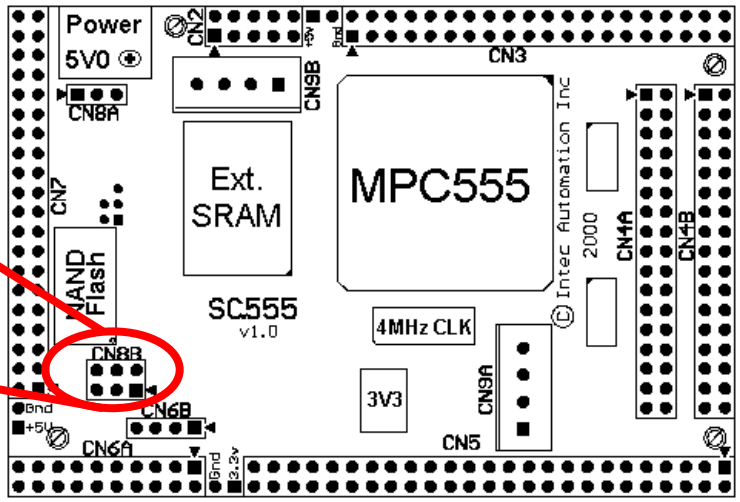


Figure 16 - Location of Clock Calendar Pins

The external real time clock calendar daughter board is still under development. Details regarding this section are forthcoming.

Hardware Handshaking (COM1)

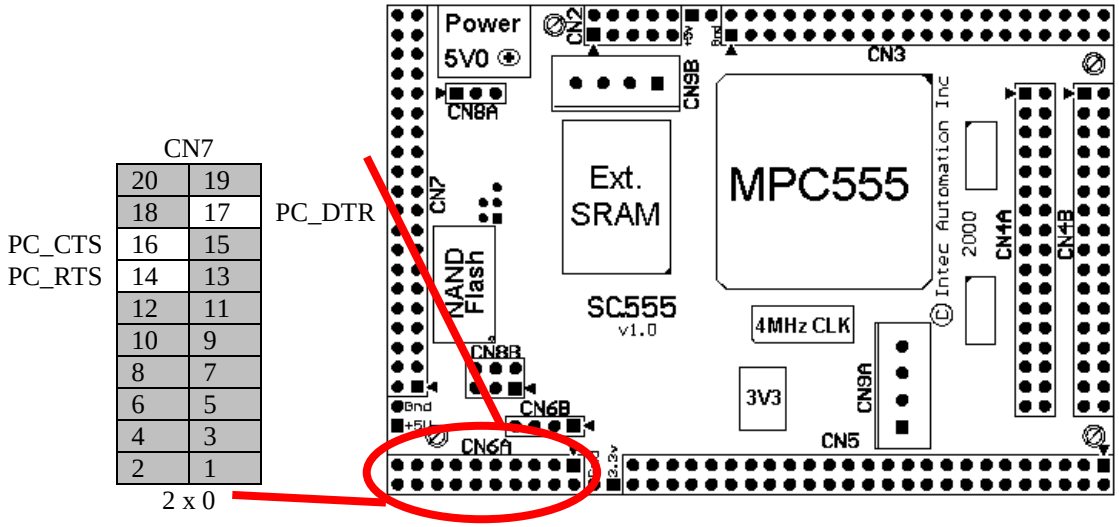


Figure 17 - Location of Hardware Handshaking Pins

PC_DTR becomes *SER_RST SER_ACK becomes PC_CTS PC_RTS becomes SER_INT

Hardware handshaking allows the MPC555 to acknowledge a received character and allows a serial device conntected to COM1 to either interrupt the MPC555 on IRQ0 or issue a hardware reset on the MPC555. The user can enable or disable the reset/interrupt functionality of COM1 and can assert/negate the SER_ACK line. Upon a hard reset, the serial interrupt and reset are disabled and SER_ACK is low.

FF9F FFFF … FF80 0003	Interrupt, reset, acknowledge are repeated due to partial address decoding.
FF80 0002	Acknowledge serial transfers.
FF80 0001	Enable/Disable serial resets.
FF80 0000	Enable/Disable serial interrupts

The functions ser_rst_en(enable) , ser_int_en(enable) and ser_ack(value) are provided to allow the user to interface to these CPLD functions.

Write Protects

Both the internal flash EPROM and the external NAND flash have signals that prevent unwanted writes of the flash. The MPC555 needs to have EPEE (EROM programming enable signal) set high and VPP (flash programming voltage) to be 5V0 to allow for writing to the internal Flash. On the SS555, writing to a single register on the CPLD controls these two signals.

If the EPEE and VPP are set for flash programming, the FLASH LED on the SS555 board will turn on. The board is set for flash programming automatically when using the P&E Flash programmer, but must be done manually if using EST's VisionCLICK.

FF9F FFFF	EPROM and NAND Flash enables and repeated due to partial address decoding.
…	
FF80 0002	
FF80 0001	Enable/Disable external NAND Flash writes.
FF80 0000	Enable/Disable internal Flash EPROM writes.

The NAND flash section of the SS555 is still under development. Details regarding this section are forthcoming.

Appendix A – Schematic Diagrams

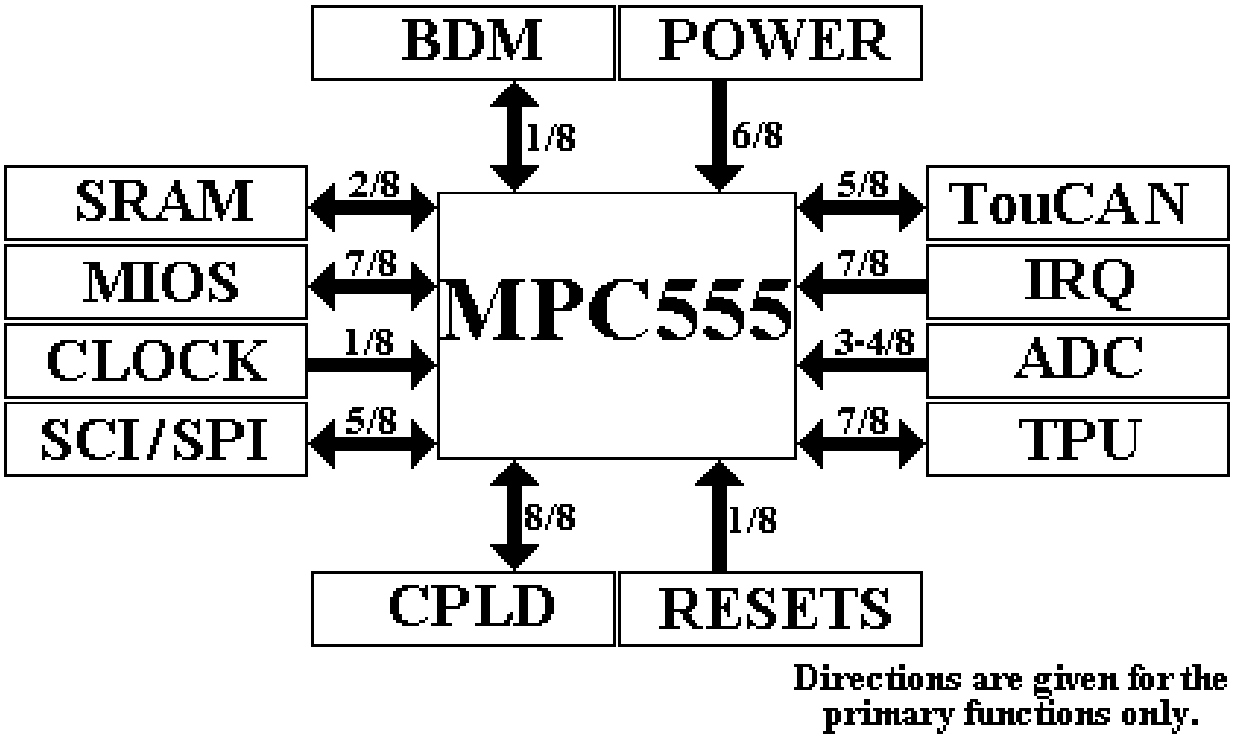
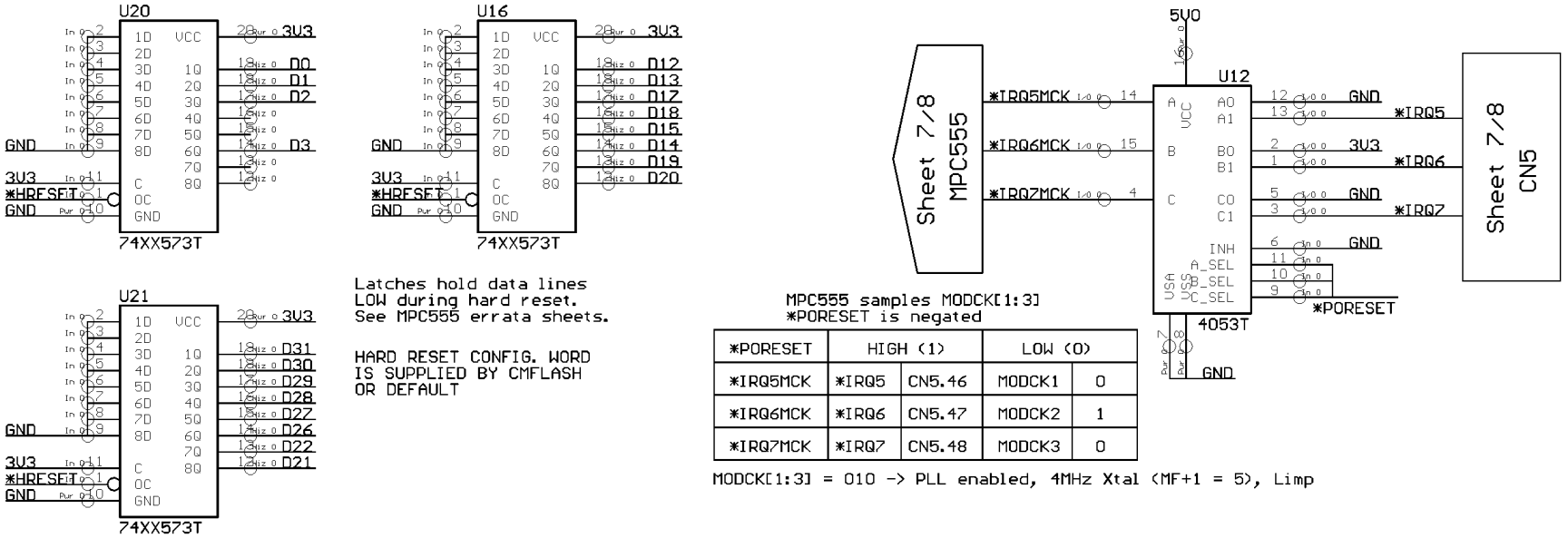


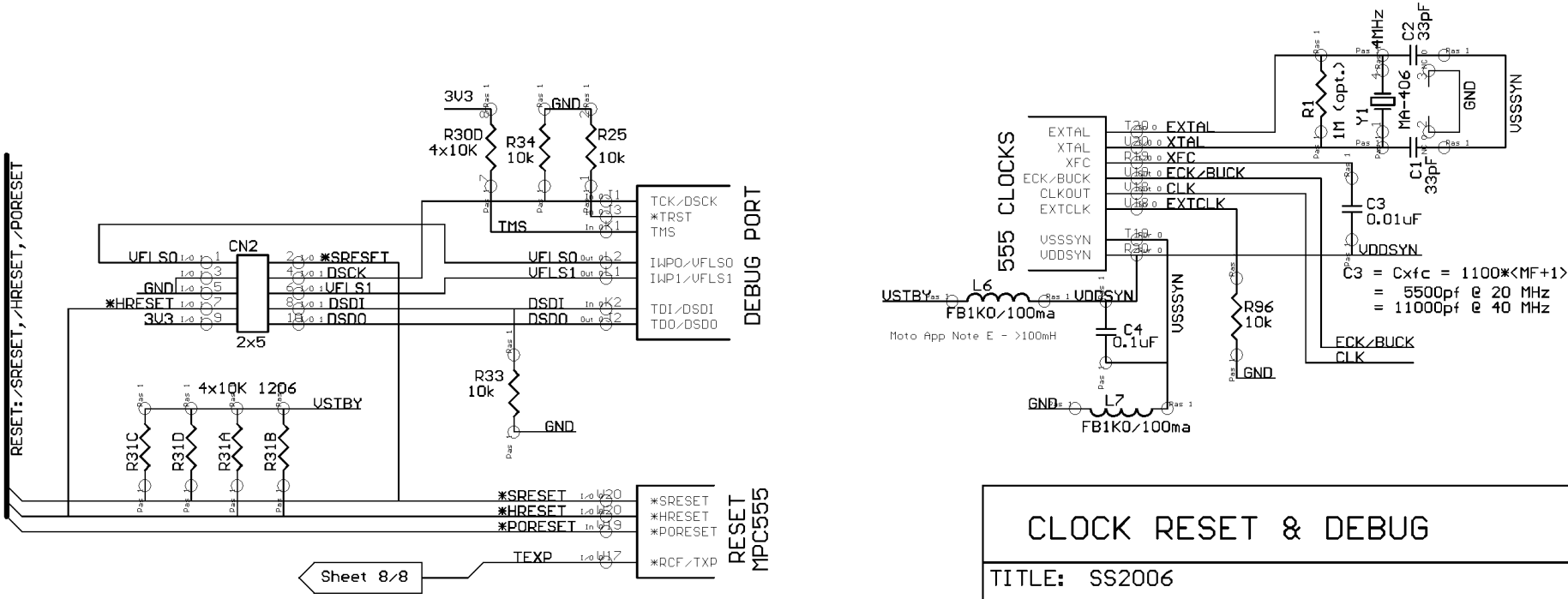
Figure 18 - Block Diagram of the Schematics



MPC555 samples MODCK[1:3]
*PORESET is negated

*PORESET	HIGH (1)		LOW (0)	
*IRQ5MCK	*IRQ5	CN5.46	MODCK1	0
*IRQ6MCK	*IRQ6	CN5.47	MODCK2	1
*IRQ7MCK	*IRQ7	CN5.48	MODCK3	0

MODCK[1:3] = 010 -> PLL enabled, 4MHz Xtal (MF+1 = 5), Limp

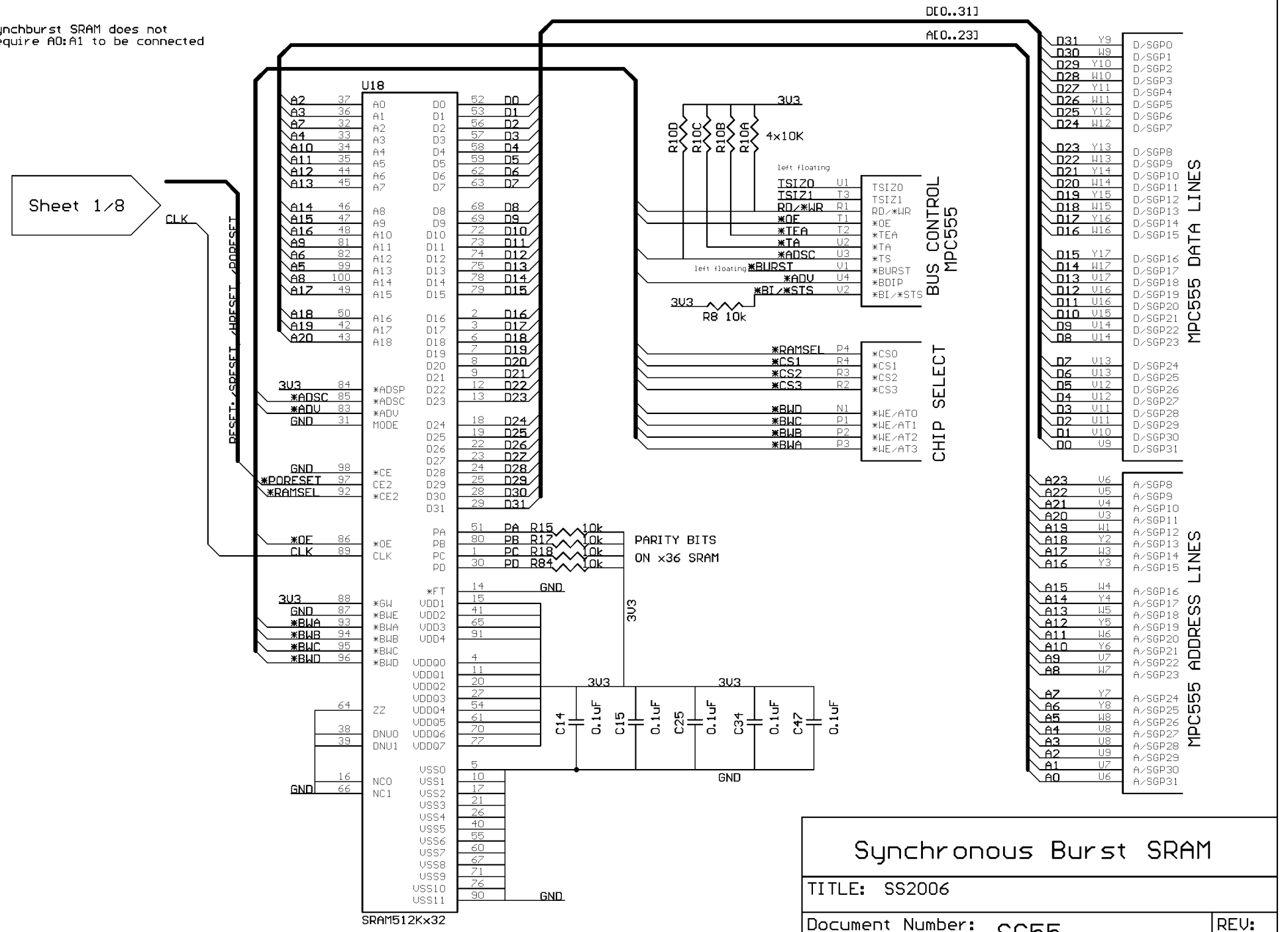


CLOCK RESET & DEBUG

TITLE: SS2006

Document Number: SC555 REV: B1

Date: 11-16-2000 09:42:44a Sheet: 1/8

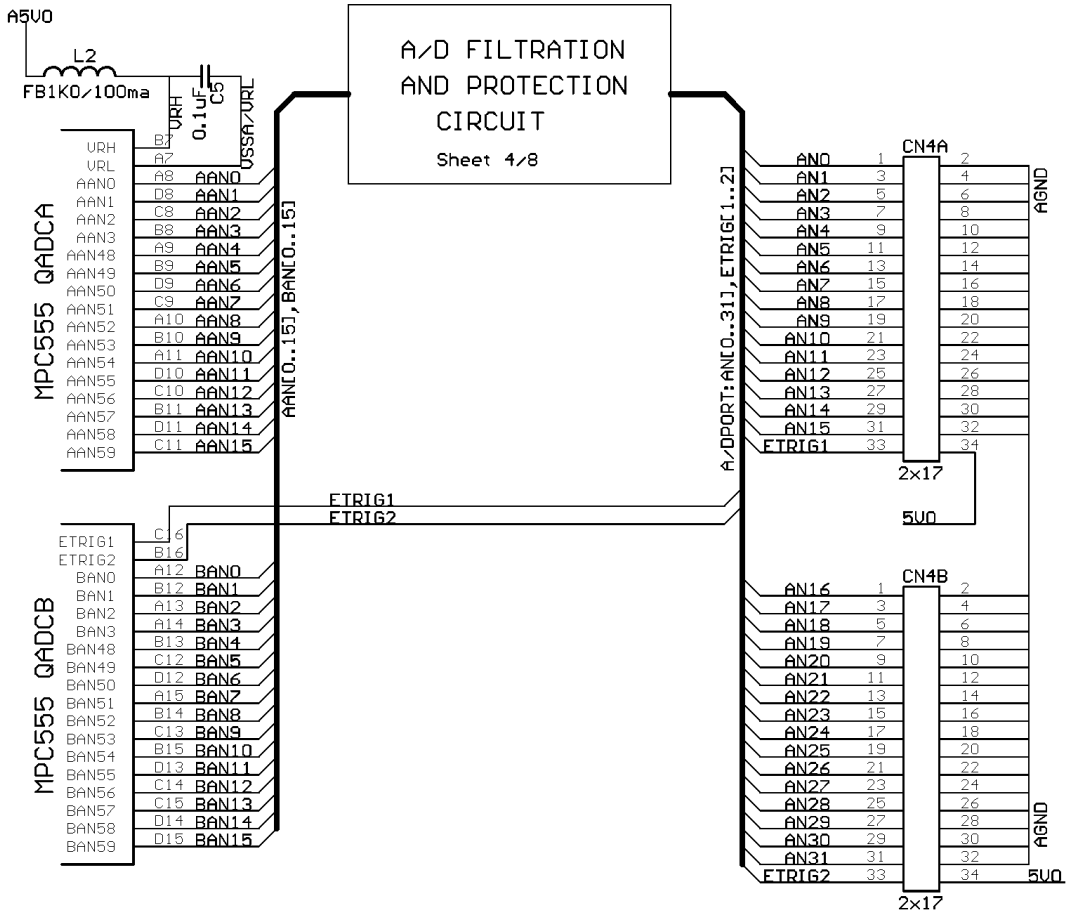


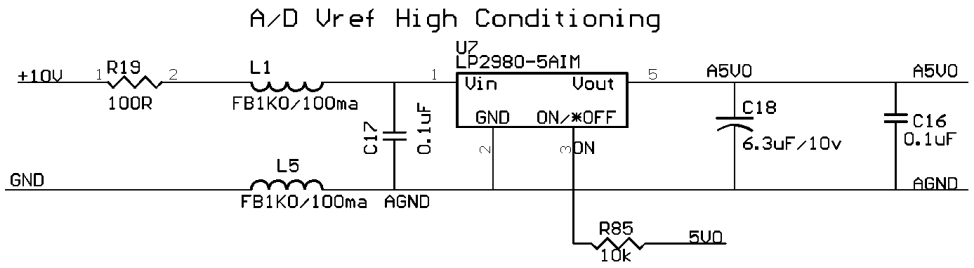
Synchronous Burst SRAM

TITLE: SS2006

Document Number: SC55 REV: B1

Date: 11-16-2000 09:42:44a Sheet: 2/8



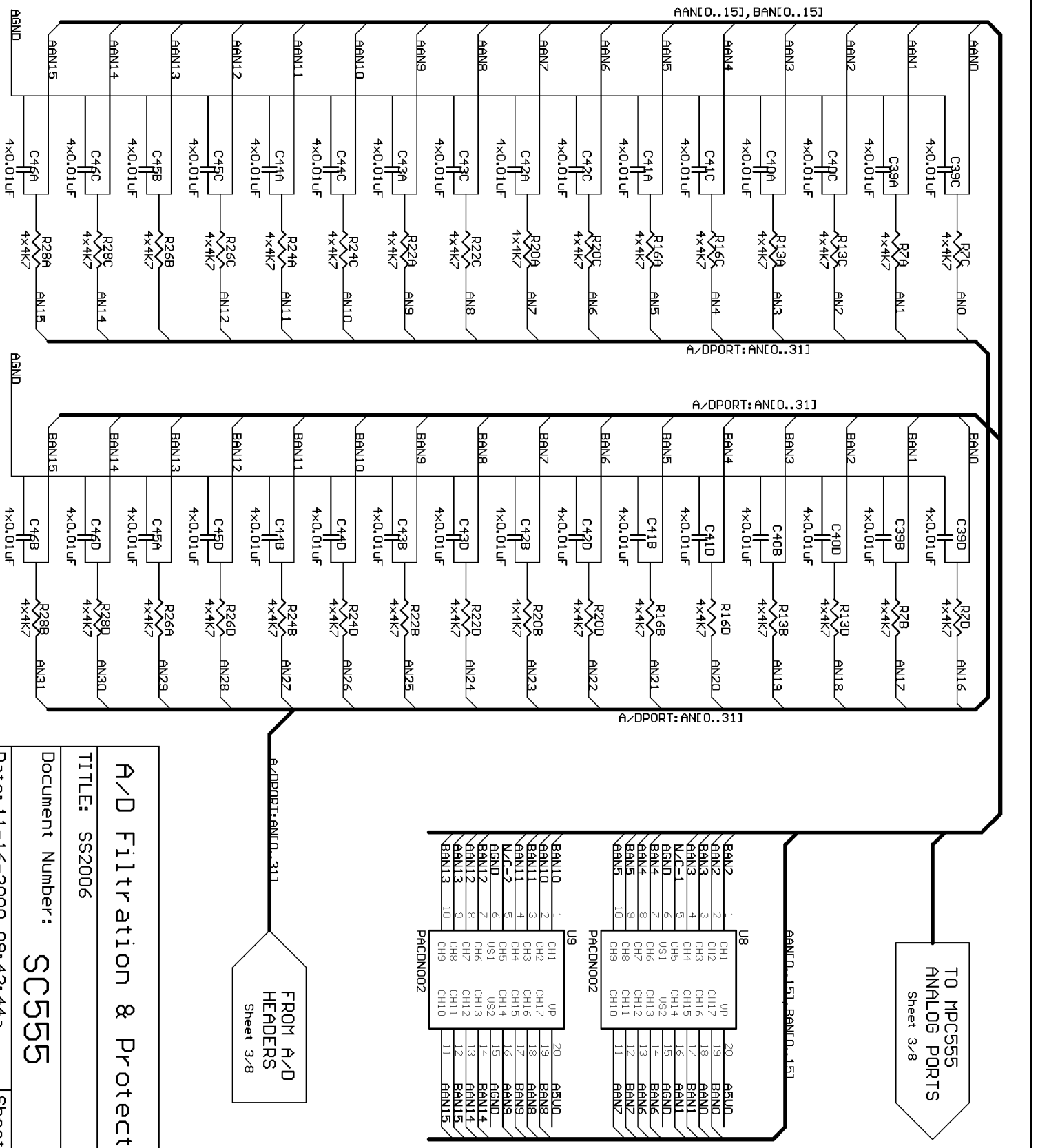


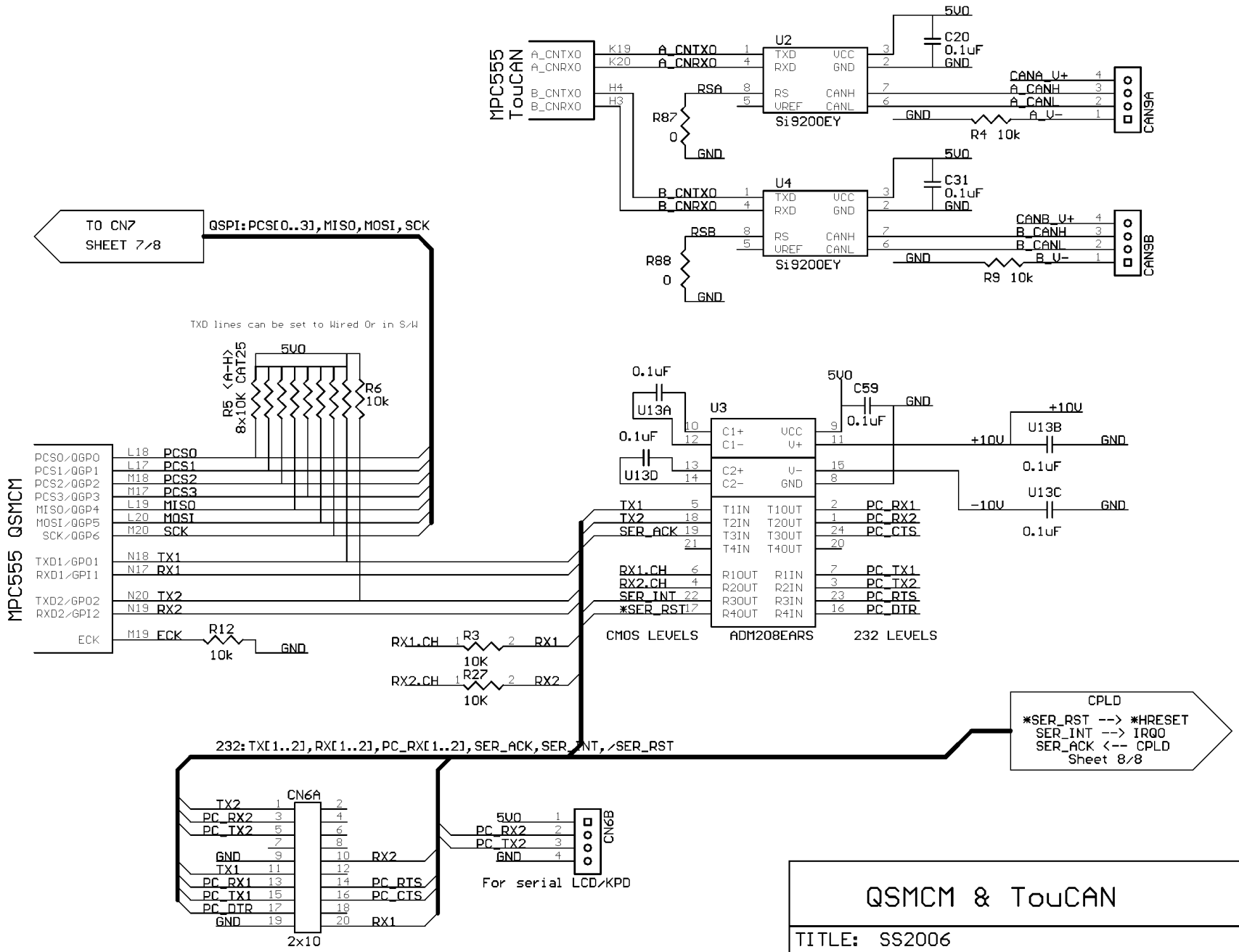
Queued A/D Inputs & Power

TITLE: SS2006

Document Number: SC555 REV: B1

Date: 11-16-2000 09:42:44a Sheet: 3/8



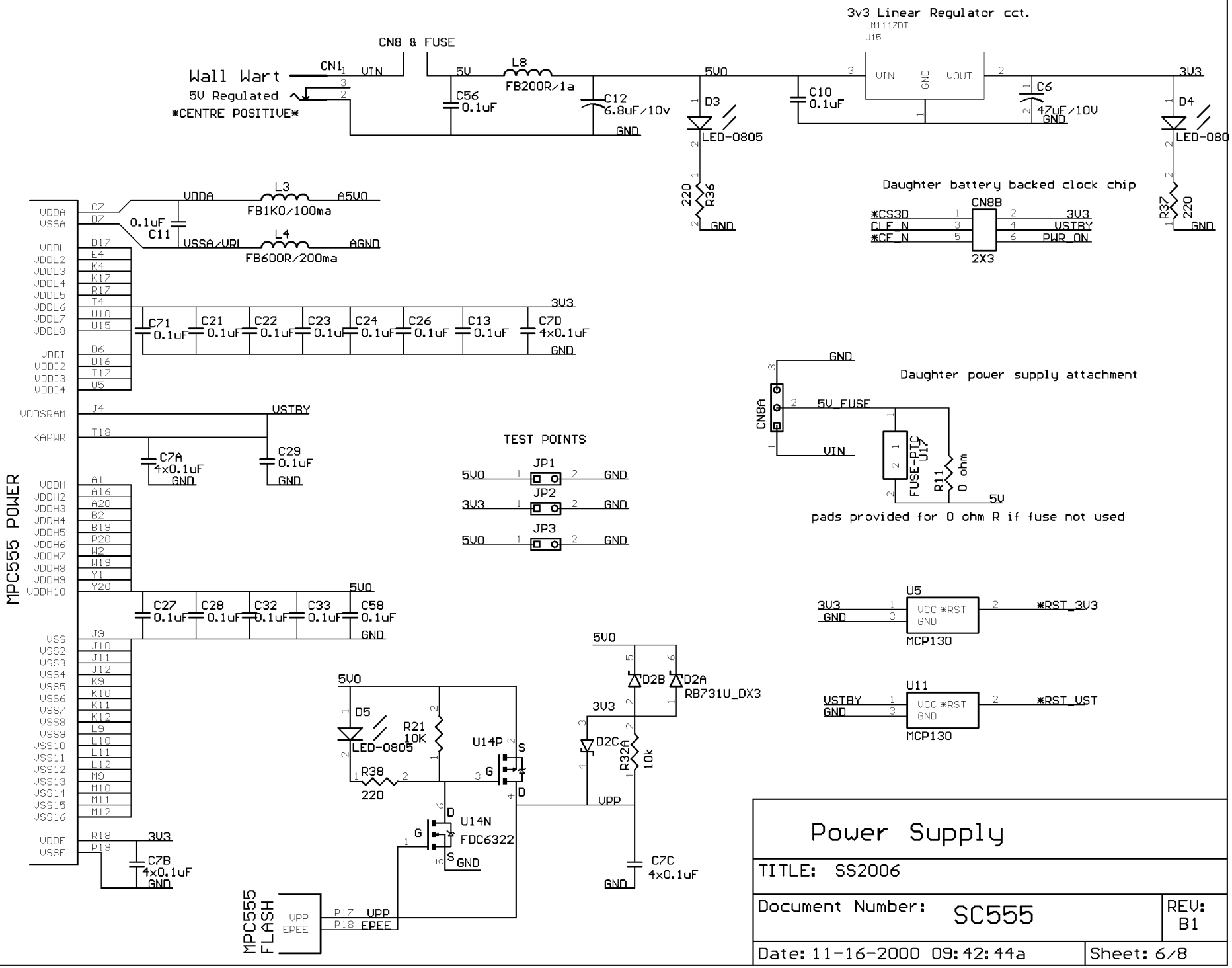


QSMCM & TouCAN

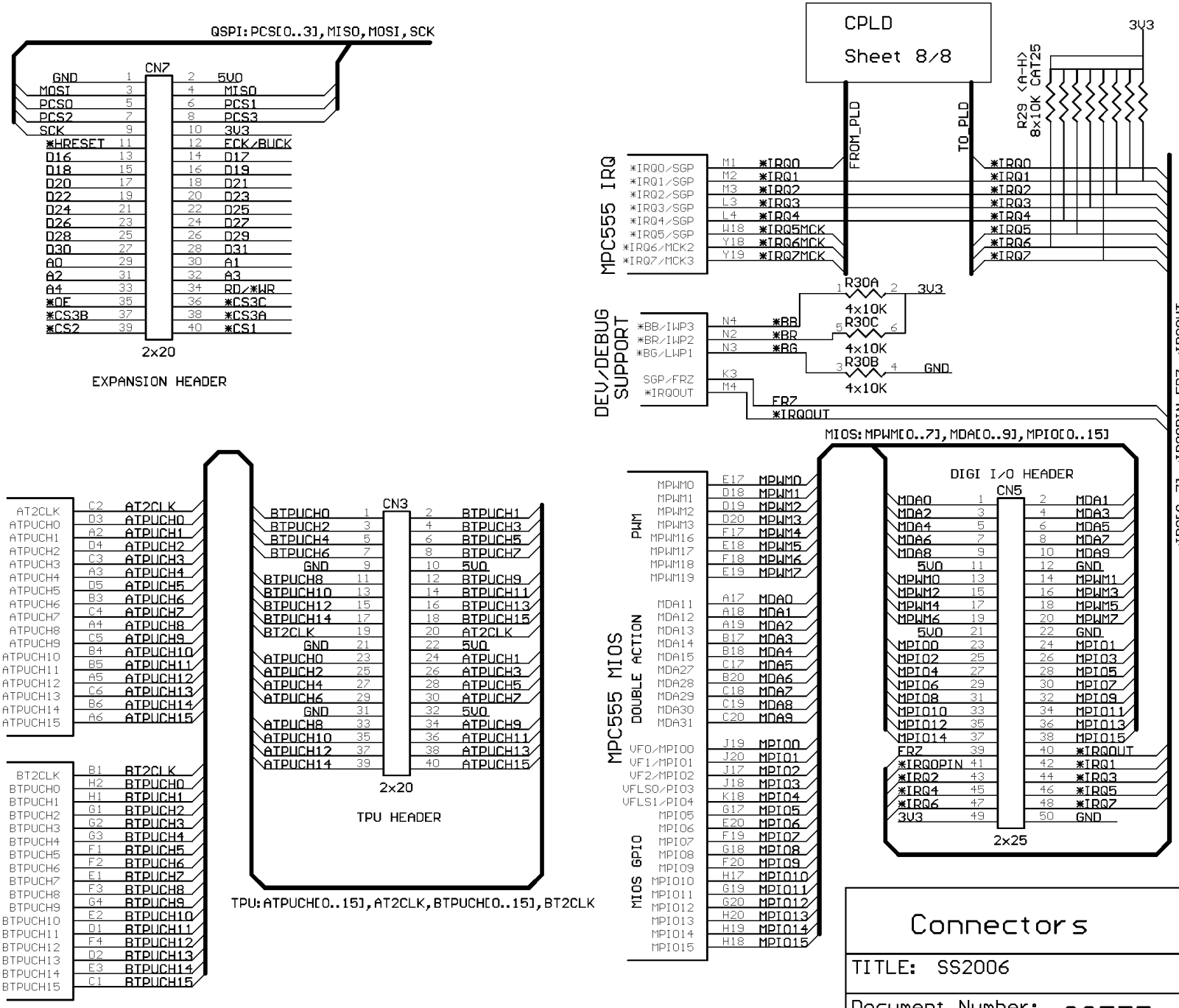
TITLE: SS2006

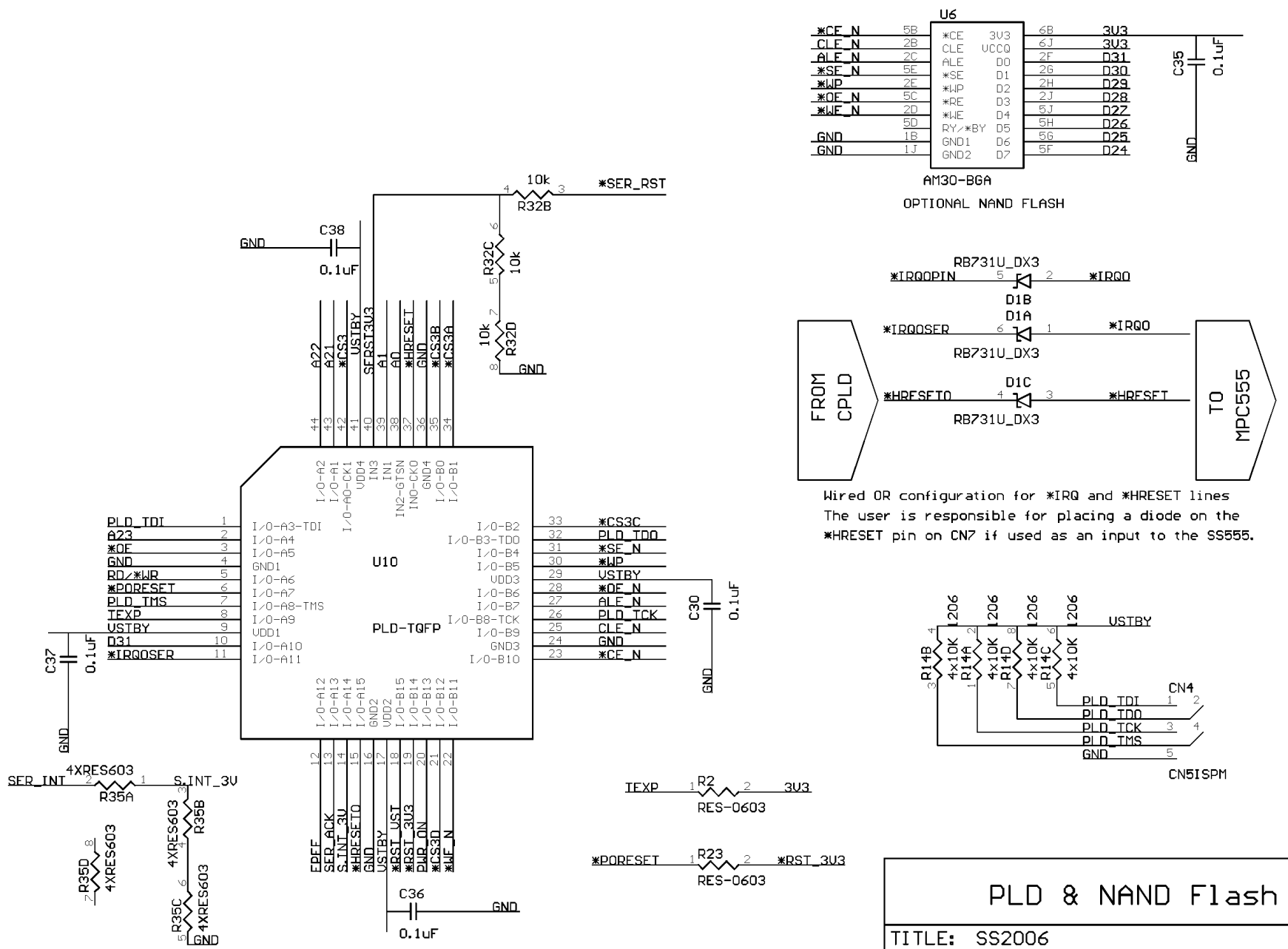
Document Number: SC555 REV: B1

Date: 11-16-2000 09:42:44a Sheet: 5/8



Power Supply	
TITLE: SS2006	
Document Number: SC555	REV: B1
Date: 11-16-2000 09:42:44a	Sheet: 6/8





PLD & NAND Flash	
TITLE: SS2006	
Document Number: SC555	REV: B1
Date: 11-16-2000 09:42:44a	Sheet: 8/8

Appendix B – Daughter Boards

Clock Calendar Daughter Board

Though the MPC555 is able to keep track of time elapsed since a program began, there is no way to keep track of time when the power is turned off, or to find out the date and time without reloading this information every time the SS555 is powered up. The solution is the clock calendar daughter board. This board keeps track of date and time and is battery backed so that it will retain it's data even if the SS555 power is disconnected.

Analog To Digital Multiplexer Daughter Board

This add-on board that multiplexes the existing 32 ADC channels to a total of 84 channels.

Low Power Daughter Board

This board allows the user to put the SS555 into the low power and sleep modes discussed in Section 6 of the MPC555 user's manual. The main power is turned off and VSTBY is provided by a battery source. The MPC555 can then be woken up either by the expiration of a MPC555 timer or asynchronously by the assertion of a wakeup pin on the daughter board itself.

Switching Power Supply

The switching power supply reduces 12V to 5V0. This is useful in automotive and industrial applications as it allows the SS555 to be run directly off of a car battery. Switching supplies are inherently noisy and using this supply causes a slight degradation in the accuracy of the analog to digital section. If analog to digital resolution is important, provide clean regulated 5V0 to the SS555.

References

"Motorola MPC555 User's Manual", Revised June 1 2000. www.mot-sps.com This document is currently only available in electronic format.

"The Zen of BDM", Craig Haller, MacRaigor Inc. www.macraigor.com. This document is currently only available in electronic format.

"Designing for Multiple SyncBurst SRAM vendors", Micron application note. www.micron.com. This document is currently only available in electronic format.

"IPM User's manual", Intec Automation Inc., ©2000. This is currently an internal document under development by Intec Automation Inc.

"Xilinx XCR3032C: 32 Macrocell CPLD with Enhanced Clocking", © 1999 Xilinx Semiconductors Inc.