

SS555-DK

Installation and Quick Start Manual



26/09/00
Intec Automation Inc.

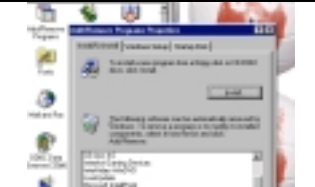

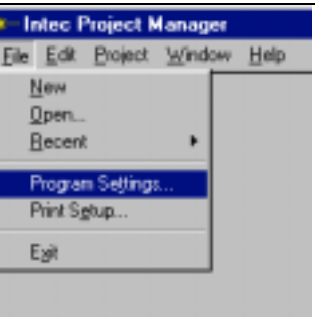
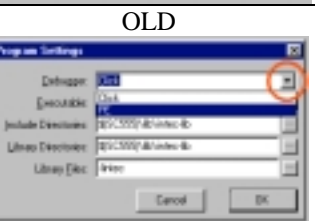
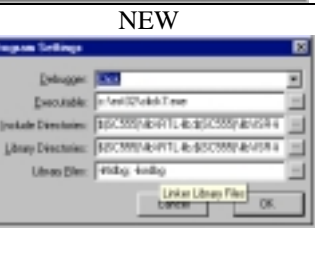
Table of Contents

- 1 SPECIAL NOTE: UPGRADING TO THE SS555-SDK 1.3 1**
- 2 INTRODUCTION..... 2**
- 3 WARRANTY 2**
- 4 DISCLAIMER 2**
- 5 COPYRIGHT..... 3**
- 6 PACKING LIST..... 4**
- 7 INSTALLATION OF IPM, GNU C AND DOCUMENTATION..... 5**
- 8 INSTALLATION OF ICDPPC 7**
- 9 INSTALLATION OF PROGPPC 9**
- 10 COPYING OF .PCP FILES..... 10**
- 11 SS555-SDK DEMONSTRATION PROGRAMS 11**
 - 11.1 ADC.IPJ - ANALOG TO DIGITAL READINGS..... 11
 - 11.2 BLINKALL.IPJ – SIMPLE PROGRAM TO TEST BOARD OPERATION..... 11
 - 11.3 HRESET.IPJ - HARDWARE RESET THROUGH SOFTWARE..... 12
 - 11.4 FLASH.IPJ – ENABLE/DISABLE FLASH PROGRAMMING 12
 - 11.5 SER_ACK.IPJ/SER_INT.IPJ/SER_RST.IPJ - HARDWARE HANDSHAKING FOR SCI 1 12
 - 11.6 IRQ_GPIO.IPJ - IRQ PINS ARE SET FOR I/O OPERATION..... 12
 - 11.7 IRQ_ISR_ALL.IPJ - SETS ALL IRQ PINS FOR INTERRUPT 12
 - 11.8 MDA.IPJ – MIOS DOUBLE ACTIO PINS 12
 - 11.9 MPIO_DEMO.IPJ - MIOS PARALLEL IO PINS ARE SET FOR I/O OPERATION..... 12
 - 11.10 MPIO_PORT.IPJ - MIOS PARALLEL IO PINS ARE SET FOR OUTPUT OPERATION 12
 - 11.11 PWM.IPJ - PWM PINS SET FOR PWM OPERATION..... 12
 - 11.12 PWMI.IPJ - PWM PINS ARE SET FOR INPUT OPERATION 12
 - 11.13 PWMO.IPJ - PWM PINS ARE SET FOR OUTPUT OPERATION 12
 - 11.14 TB.IPJ - TIME BASE TIMER (NON-INTERRUPTING) 13
 - 11.15 PIT.IPJ - PERIODIC INTERRUPT TIMER (NON-INTERRUPTING)..... 13
 - 11.16 RTC.IPJ - REAL TIME CLOCK (NON-INTERRUPTING)..... 13
 - 11.17 SCI1.IPJ - SCI 1 COMMUNICATION AND THE CHANGING OF THE MPC555 CLOCK FREQUENCY . 13
 - 11.18 SCI1-2.IPJ – SCI 1 BI-DIRECTIONAL COMMUNICATION..... 13
 - 11.19 SCI2.IPJ - SCI 2 COMMUNICATION AND THE CHANGING OF THE MPC555 CLOCK FREQUENCY . 13
- 12 ISR555 DEMONSTRATION PROGRAMS..... 13**
 - 12.1 MCS.IPJ – MIOS COUNTER SUBMODULE..... 14
 - 12.2 MDAPWM_ISR.IPJ – MIOS DOUBLE ACTION SUBMODULE AS PWM..... 14
 - 12.3 PWM_ISR.IPJ – MIOS PULSE WIDTH MODULATION SUBMODULE..... 14
 - 12.4 COL.IPJ – CHANGE OF LOCK 14
 - 12.5 DEC.IPJ - DECREMENTER TIMER GENERATES INTERRUPTS 14
 - 12.6 PITINT.IPJ - PERIODIC INTERRUPT TIMER GENERATES INTERRUPTS..... 14
 - 12.7 PITRTC.IPJ - PERIODIC INTERRUPT TIMER AND REAL TIME CLOCK BOTH GENERATE INTERRUPTS
14
 - 12.8 ALARMS.IPJ – REAL TIME CLOCK TO GENERATE AN INTERRUPT WHEN A PARTICULAR VALUE IS
REACHED 14

12.9	COUNTSECS.IPJ - REAL TIME CLOCK GENERATES AN INTERRUPT EVERY SECOND	14
12.10	TB_ISR.IPJ - TIME BASE TIMER GENERATES INTERRUPTS.....	14
12.11	WDT.IPJ - WATCHDOG TIMER SETUP AND SERVICED	15
13	SETTING UP ICDPPCZ WITHIN IPM	16
14	COMPILING A PROGRAM IN IPM.....	17
15	LOADING/DEBUGGING WITH ICDPPC	19

1 Special Note: Upgrading to the SS555-SDK 1.3

To upgrade to the current version of the SS555-SDK, please do the following:

<p>Save all of your current applications and move them out of the SS555 directory.</p>	
<p>Remove your current version of IPM through the add/remove utility in the control panel.</p>	
<p>Open your c:\autoexec.bat file in a text viewer such as notepad and ensure that the following entries has been removed</p> <p>SET SC555=C:\PROGRA~1\SS555\GCC</p> <p>Also ensure that C:\PROGRA~1\SS555\GCC\BIN has been removed from you path.</p>	
<p>Install the new version of IPM as outlined in this document. Please note that the directory path cannot contain spaces as this was causing hidden errors in previous versions.</p>	
<p>Under the IPM window Click on File-Program Settings.</p> <p>Open File-Program Settings.</p>	
<p>Please note that the Include Directory, Library directory, and Library files should have changed. The new values are listed below and should be typed in manually if they have not been updated:</p> <p>Include Directories: \$(SC555)\LIB\RTL-lib;\$(SC555)\LIB\ISR-lib Library Directories: \$(SC555)\LIB\RTL-lib;\$(SC555)\LIB\ISR-lib Library Files: -rtldbg; -isrdbg</p>	<p>OLD</p>  <p>NEW</p> 

2 Introduction

The SS555 development kit includes everything a user needs to start writing, compiling, debugging, and flash programming a program for the Motorola MPC555. Included in the kit is the SS555 board, SS555-SDK CD-ROM, GNU C compiler for the PowerPC, Intec Project Manager (IPM), SS555 Runtime Libraries (RTL-555), SS555 User's Manual, P&E PowerPC BDM debug cable, P&E PowerPC debugger software (ICDPPC), P&E PowerPC flash programming software (PROGPPC), 5V0 centre positive wall wart, serial cable and serial adaptor.

The SS555 board, a 2.5" x 3.7" development board, is well suited for both prototyping and embedding into finished products. The board is equipped with external SRAM, CAN, and RS-232 transceivers, 3V3 power supply, analog power supply, low pass filters on A/D pins, and a CPLD for extra board functions.

The GNU C compiler included in this package is fast becoming the compiler of choice for its robustness, and ability to span multiple operating systems and targets. The version included on this CD-ROM runs under DOS and compiles for the MPC5xx family of microcontroller. The largest drawback with GNU C is a very steep learning curve associated with the makefiles and scripts that must be written in order to compile programs under GNU. The Intec Project Manager (IPM) is a Windows application that hides the complexities of GNU. IPM is a graphical interface for writing, building, linking and compiling projects.

The P&E PowerPC debugger software (ICDPPC) is a program that, in conjunction with the P&E debug cable, allows the user to step, run, set breakpoints, change and inspect both registers and memory, and test code in SRAM. Unlike many conventional debugging tools that operate via the serial port, the SS555-DK uses Background Debug Mode (BDM) to control execution and read registers without interfering with serial communications. The user gets more debugging information without tying up the serial port.

The P&E PowerPC flash programming software (PROGPPC) enables the user to load debugged code into the internal Flash of the MPC555 for stand-alone program execution. Once this step is completed, the MPC555 runs independently of the PC.

PLEASE NOTE: This is not the SS555 User's Manual. The User's Manual is installed during the SS555-SDK installation. The User's Manual can be found in:

`..\SS555\Documentation\SS555 User's Manual\SS555um.pdf`

3 Warranty

Intec Inoventures 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 Inoventures 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 Inoventures 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 Inoventures 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.

4 Disclaimer

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.


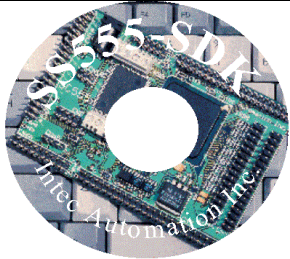






5 Copyright

Intec Inoventures Inc. products may not be used as components in life support devices of any description. The design of the SS555 and the Intec Project Manager software is copyrighted, and copying part or all of the design is prohibited by law.

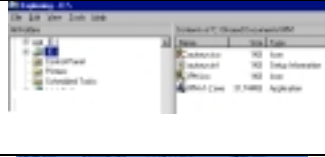

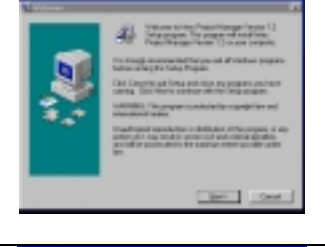

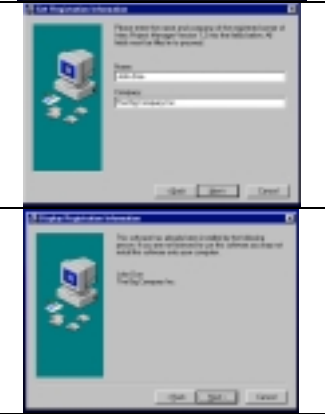
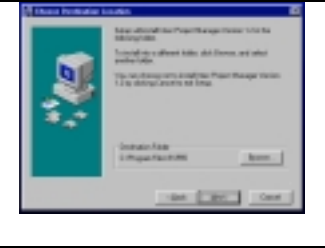
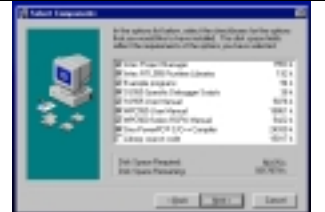
6 Packing List

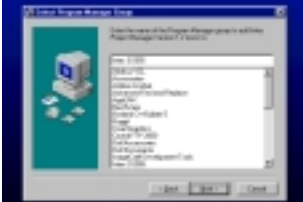
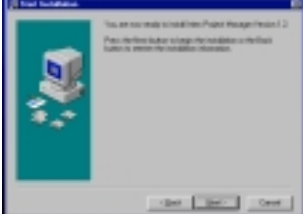
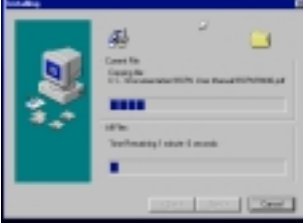
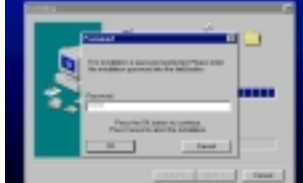
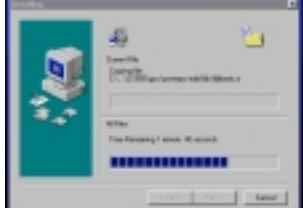
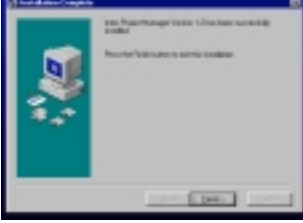
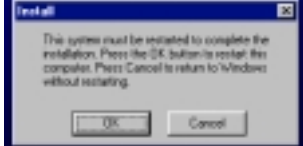
Each SS555 development kit includes:

- 1 SS555 board.
- 1 SS555-SDK CD-ROM
 - 1 GNU C compiler for the PowerPC
 - 1 Intec Project Manager (IPM) – Windows GUI for the GNU C compiler
 - 1 SS555 Runtime Libraries (RTL-555) – useful library functions and demo programs
 - 1 SS555 User's Manual – introduction to the board and runtime libraries
- 1 P&E PowerPC BDM debug cable
- 1 P&E PowerPC debugger software (ICDPPC)
- 1 P&E PowerPC flash programming software (PROGPPC)
- 1 5V0 centre positive wall wart
- 1 Serial adaptor
- 1 Serial cable

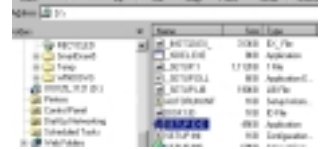

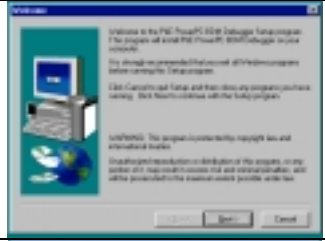


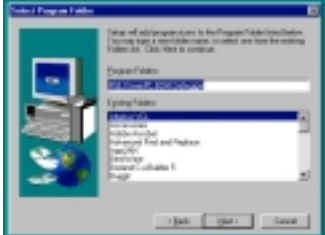
		
SS555 Board	GNU C compiler Intec Project Manager (IPM) SS555 Runtime Libraries SS555 User's Manual	P&E PowerPC BDM debug cable
		
P&E Debugger – ICDPPC	P & E Programmer – PROGPPC	5V0 Centre Positive Wall Wart
		Actual items may not be exactly as shown. Intec Automation Inc. reserves the right to alter, modify or substitute hardware and software without notice.
Serial Cable Adaptor	Serial Cable	

7 Installation of IPM, GNU C and Documentation

<p>Insert the SS555-SDK CD-ROM into the drive. The CD should automatically start the installation program. If it doesn't run the IPMVx.exe file located on the SS555-SDK CD.</p> <p>Insert CD. Run the IPMVx.exe file if necessary.</p>	
<p>The installer will start.</p>	
<p>Click Next</p>	
<p>Read the warranty information. A copy will be installed to your hard drive. Make sure that you understand the terms and conditions of both hardware and software associated with this product before continuing. When you have read the warranty, click next.</p> <p>Click Next</p>	
<p>Enter your personal information into the form. The Company field cannot be left blank. Click Next when complete.</p> <p>If IPM has already been installed on the PC previously, the owner information will be displayed. Click Next.</p> <p>Click Next</p>	
<p>Choose a download location. The default is: C:\Intec\SS555\</p> <p>Selecting a path leads to subtle errors with the GNU compiler. Do not include spaces in the path name. Reinstalling IPM to different directories may lead to problems. If problems occur see the troubleshooting FAQ.</p> <p>Choose the path that does not include spaces and click Next.</p>	
<p>Select the components to install. Unless you are familiar with IPM it is recommended to choose the default configuration (automatically highlighted). If you would like the entire GNU C source, it is available by clicking on library source code dialog box.</p> <p>Choose components and click Next.</p>	

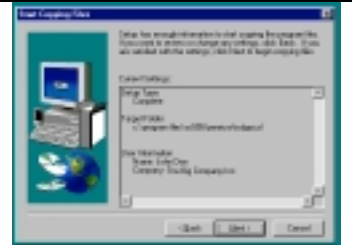
<p>Select a group to add the IPM shortcut icons. The default is: Intec SS555.</p> <p>Click Next.</p>	
<p>You are ready to start installing the SS555-SDK.</p> <p>Click Next.</p>	
<p>The installation will begin.</p>	
<p>A dialog will appear asking for the user password before the installation of IPM. Enter the CD Key found on the inside cover of the SS555-SDK jewel case. When finished click OK.</p> <p>Enter CD-Key and press OK.</p>	
<p>The installation will complete.</p>	
<p>A message will appear telling that IPM has successfully installed on your PC.</p> <p>Click Finish.</p>	
<p>The PC must be restarted before IPM will work. IPM will not function properly until the PC is restarted.</p> <p>Click OK.</p>	

8 Installation of ICDPPC

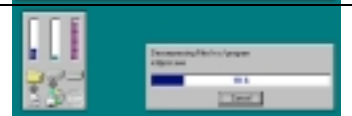
<p>Insert the ICDPPC CD-ROM into the drive. The CD should automatically start the installation program. If it doesn't run the setup.exe file located on the CD_ROM.</p> <p>Insert the ICDPPC CD. Run SETUP.EXE file if necessary.</p>	
<p>The installer will start.</p>	
<p>Click Next.</p>	
<p>If you are using NT make sure that you are logged in as administrator. Ignore this window for all other versions of Windows.</p> <p>Click Next.</p>	
<p>Enter the base directory to install the ICDPPCZ debugger. The default directory suggested by P&E Micros is: C:\pemicro\ICDPPCZ\</p> <p>Intec Automation Inc. suggests the following directory in order to keep all of the SS555 software under one directory tree. C:\Intec\SS555\pemicro\ICDPPCZ.</p> <p>Whichever you decide will not affect program execution. Make sure that you make a note of where you installed the program for reference later in the installation.</p> <p>Choose the path and click Next.</p>	
<p>Select a group to add the ICDPPC shortcut icons. The default is: P&E PowerPC BDM Debugger.</p> <p>Click Next.</p>	

You are ready to start installing the P&E PowerPC Debugger.

Click Next.



The P&E PowerPC Debugger will now install.

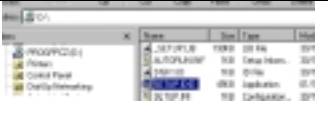

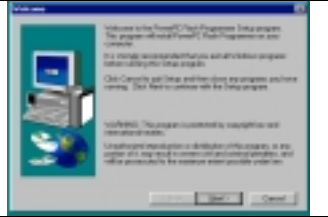
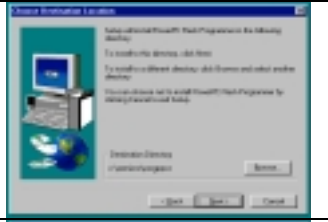
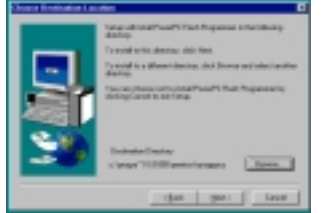

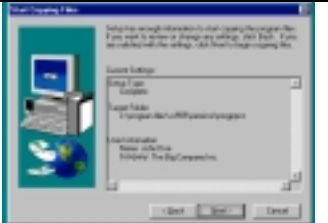

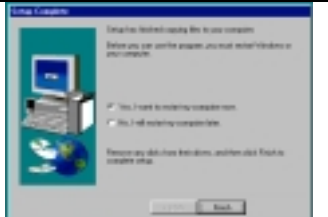


The PC must be restarted before the P&E PowerPC Debugger will work. ICDPPC will not function properly until the PC is restarted.

Click FINISH.









9 Installation of PROGPPC

<p>Insert the PROGPPC CD-ROM into the drive. The CD should automatically start the installation program. If it doesn't run the setup.exe file located on the CD_ROM.</p> <p>Insert the PROGPPC CD. Run SETUP.EXE file if necessary.</p>	
<p>The installer will start.</p>	
<p>Click Next.</p>	
<p>Enter the base directory to install the PROGPPC Programmer. The default directory suggested by P&E Micros is: C:\pemicro\PROGPPCZ \</p> <p>Intec Automation Inc. suggests the following directory in order to keep all of the SS555 software under one directory tree. C:\Intec\SS555\pemicro\PROGPPCZ.</p> <p>Whichever you decide will not affect program execution. Make sure that you make a note of where you installed the program for reference later in the installation.</p> <p>Choose the path and click Next.</p>	 
<p>Select a group to add the PROGPPC shortcut icons. The default is: P&E PowerPC Flash Programmer.</p> <p>Click Next.</p>	
<p>You are ready to start installing the P&E PowerPC Flash Programmer.</p> <p>Click Next.</p>	
<p>The P&E PowerPC Flash Programmer will now install.</p>	
<p>The PC must be restarted before the P&E PowerPC Flash Programmer will work. PROGPPC will not function properly until the PC is restarted.</p> <p>Click FINISH.</p>	

10 Copying of .PCP files

The .PCP files that are included on the P&E PROGPPC CD-ROM do not contain SS555 specific code that is needed to program the internal Flash. The correct files must be copied into this directory and the old .PCP files deleted.

<p>First determine which version of MPC555 chip is mounted on your board. To do this read the five-digit revision code on the top of the MPC555 chip.</p> <p>Rev G: 3J76N Rev K3: 01K83H</p> <p>If your version differs from those listed above, contact Intec Automation Inc. for updated programming algorithms for your particular revision.</p> <p>Do not use incorrect algorithms to try to program the MPC555 as damage may result.</p>	   <p>Rev K3 Rev G</p>
<p>Go to the scripts\PROGPPCZ directory, located in the base directory that the SS555-SDK was installed into. The default location is: C:\Intec\SS555\Scripts\PROGPPCZ\</p> <p>Go into the correct subdirectory for your revision of CPU and copy the four .PCP files located in the directory.</p> <p>Copy ..\SS555\Scripts\PROGPPCZ\ *.PCP</p>	
<p>Go to the directory that PROGPPCZ was installed into. Delete the 555_*.PCP files. These files do not contain the modified code needed to operate with the SS555 board. Delete the following files: 555_192c.PCP 555_192k.PCP 555_256c.PCP 555_256k.PCP</p> <p>Delete the all 555_*.PCP files in the PROGPPCZ program directory</p>	
<p>Paste the files into the directory that PROGPPCZ was installed into. The recommended default is: C:\Intec\SS555\pemicro\PROGPPCZ</p> <p>Paste startup.icd into ..\pemicro\PROGPPCZ\</p>	

11 SS555-SDK Demonstration Programs

This is the directory tree of example programs that come with the SS555-DK. There is also a small précis on the operation of the program and the purpose of the demo. More demo programs are added on a periodic basis. Check the Intec Automation Inc. website at www.steroidmicros.com for the most up to date version of IPM.

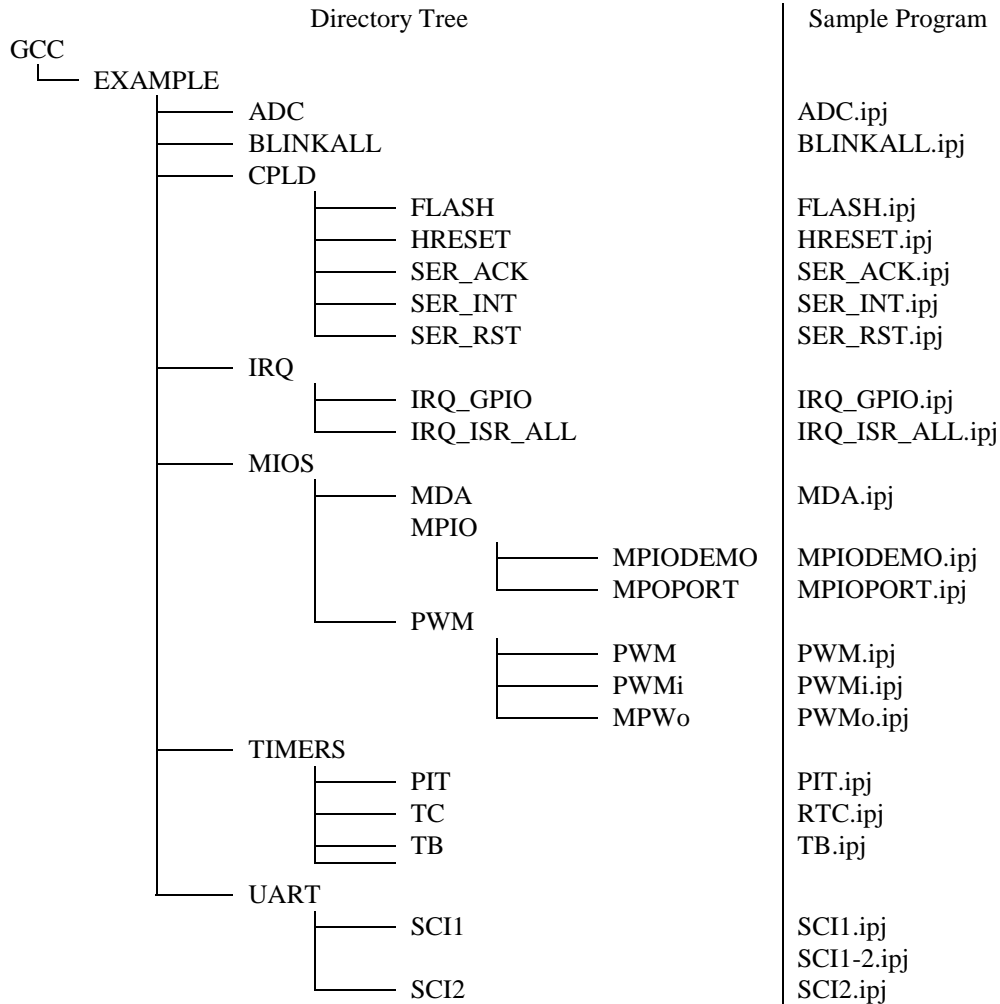


Figure 1 - Directory Tree of SS555-DK Demo Programs and their IPM Project Files

11.1 ADC.ipj - Analog to Digital readings

Port A and Port B have three modes of operation. The port can either sample all 16 pins once or sample a single pin 64 times. The 64 samples can either be displayed as a sum of 64 readings or an average of 64. This program demonstrates all three modes of operation. Each port can also be setup independently of one another.

11.2 Blinkall.ipj – Simple program to test board operation

Demonstration program for the various modules of the SS555 Created for the Embedded Systems Conference in San Francisco, April 2001. Test of all external pins supported by the SS555-SDK.

11.3 HRESET.ipj - Hardware reset through software

The MPC555 cannot assert an *HRESET through software. The SS555 allows the user to assert *HRESET through software by using an external CPLD.

11.4 FLASH.ipj – Enable/Disable Flash Programming

This is necessary to program the onboard FLASH at runtime.

11.5 SER_ACK.ipj/SER_INT.ipj/SER_RST.ipj - Hardware handshaking for SCI 1

At present, there is no software support in stdio.h for this functionality, but these functions could be used to design a basic handshaking protocol.

11.6 IRQ_GPIO.ipj - IRQ pins are set for I/O operation

Setup the IRQ pins for general purpose output functionality. Connect the SCI1 port to the PC serial port. The pins will toggle once the user has hit the ENTER key on the keyboard. Notice that on version 1.1 of the MPC555 that *IRQ0 cannot be used as a general-purpose output.

11.7 IRQ_ISR_ALL.ipj - Sets all IRQ pins for interrupt

The IRQ pins are setup for interrupt functionality. The currently interrupting IRQ will transmit its number out COM1.

11.8 MDA.ipj – MIOS Double Actio pins

Show the operation of the MDA in output PWM (OPWM) mode. PWM pulses are generated on MDA0:9.

11.9 MPIO_DEMO.ipj - MIOS Parallel IO pins are set for I/O operation

Odd and even pins are connected together. Even pins are setup as outputs and odd pins are setup as inputs. The values of the odd pins are read and the results are sent to the PC via the COM port.

Pin0-Pin1	Pin4-Pin5	Pin8-Pin9	Pin12-Pin13
Pin2-Pin3	Pin6-Pin7	Pin10-Pin11	Pin14-Pin15

Odd pins are toggled high then low indefinitely. Even pins are read indefinitely.

11.10 MPIO_PORT.ipj - MIOS Parallel IO pins are set for output operation

All MPIO pins are setup as outputs. The value of the MPIO pins toggles indefinitely.

11.11 PWM.ipj - PWM pins set for PWM operation

Sets up different duty cycles on each PWM pin. This sets up a fast PWM duty cycle. For a slower cycle and more detail on changing period and duty cycle see the blinkall.ipj project.

11.12 PWMi.ipj - PWM pins are set for input operation

All PWM pins are setup as inputs and the value of the pins is sent out the data port. An external signal is expected to change the PWM pins from a LO to a HI.

11.13 PWMo.ipj - PWM pins are set for output operation

The value on the pins toggles indefinitely.

11.14 TB.ipj - Time Base timer (non-interrupting)

Demonstration of the basic Time Base functionality of the MPC55. The value of the TB is printed via SCI1 every time the ENTER button is pressed. Typing any character other than ENTER adds characters to the receive buffer, causing multiple reads of the TB once the ENTER key is finally pressed.

11.15 PIT.ipj - Periodic Interrupt Timer (non-interrupting)

Demonstration of the basic PIT functionality of the MPC55. The value of the PIT is printed via SCI1 every time the ENTER button is pressed. Typing any character other than ENTER adds characters to the receive buffer, causing multiple reads of the PIT once the ENTER key is finally pressed.

11.16 RTC.ipj - Real Time Clock (non-interrupting)

Demonstration of the basic RTC functionality of the MPC55. The value of the RTC is printed via SCI1 every time the ENTER button is pressed. Typing any character other than ENTER adds characters to the receive buffer, causing multiple reads of the RTC once the ENTER key is finally pressed.

11.17 SCI1.ipj - SCI 1 communication and the changing of the MPC555 clock frequency

The clock frequency of the MPC555 is changed and the SCI port is written to at different operating speeds.

11.18 SCI1-2.ipj – SCI 1 bi-directional communication

This program demonstrates both read and write capabilities of the serial port.

11.19 SCI2.ipj - SCI 2 communication and the changing of the MPC555 clock frequency

The clock frequency of the MPC555 is changed and the SCI port is written to at different operating speeds.

12 ISR555 Demonstration Programs

This is the directory tree of example interrupt programs that come with the ISR555 kit. There is also a small précis on the operation of the program and the purpose of the demo.

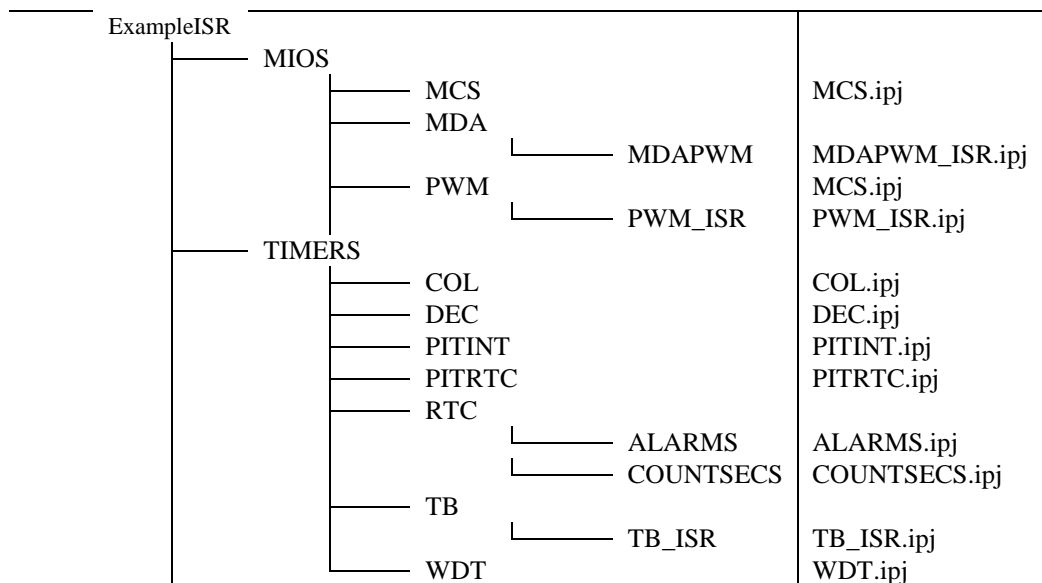


Figure 2 - Directory Tree of ISR555 Demo Programs and their IPM Project Files

12.1 MCS.ipj – MIOS Counter Submodule

Shows the operation of the MIOS counter interrupts. The CS are driven by the internal counters and an interrupt is generated every time the counters overflow.

12.2 MDAPWM_ISR.ipj – MIOS Double Action Submodule as PWM

Show the operation of the MDA in 0PWM mode. The DA pins interrupt on the PGT edge. The counters update the number of PWM periods and the number of CS overflows.

12.3 PWM_ISR.ipj – MIOS Pulse Width Modulation Submodule

Sets a long pulse train on each of the eight PWM pins. The PWMs all are set to generate an interrupt on their positive edge. The interrupt subroutines set a one-hot encoded bit in the interrupt_source variable, which is monitored by the mainline program. The current number of cycles for each PWM pin is then written out of the serial port.

12.4 COL.ipj – Change of Lock

Interrupt from a change of lock interrupt condition. This occurs when the PLL loses lock.

12.5 DEC.ipj - Decrementer timer generates interrupts

The decrementer is set to timeout once per second. When it times out the DEC is reset and the number of DEC tics is sent out SCI1 to be displayed on the PC.

12.6 PITINT.ipj - Periodic Interrupt Timer generates interrupts

Interrupt driven PIT counter. Uses the PIT to update a flag in the interrupt subroutine. The flag is then captured by the mainline program which updates a counter and transmits the counter out SCI1.

12.7 PITRTC.ipj - Periodic Interrupt Timer and Real Time Clock both generate interrupts

This is a good example of using two different sources to interrupt. The user should note that the semaphores are completely encapsulated and do not rely on anything from the library file.

12.8 ALARMS.ipj – Real Time Clock to generate an interrupt when a particular value is reached

Interrupt driven alarm counter for the MPC555. Uses the RTC to update a flag every time the value of the RTC matches the value set in an alarm register. The flag is then captured by the mainline program which updates the counter.

12.9 COUNTSECS.ipj - Real Time Clock generates an interrupt every second

Every time the RTC increments, an interrupt is generated. The handler clears the interrupt and sets a semaphore. The mainline program then updates a counter, clears the semaphore and transmits the counter out of SCI1.


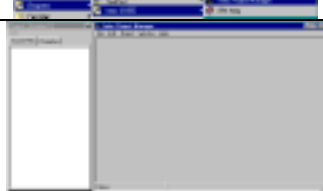
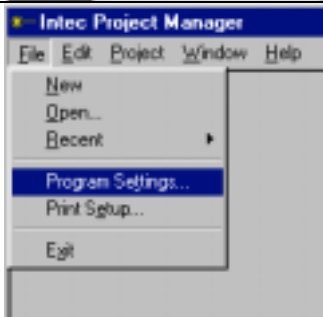
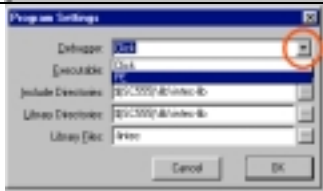

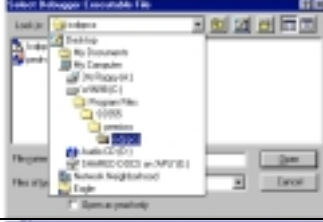
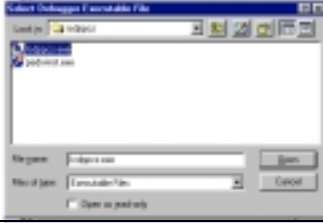
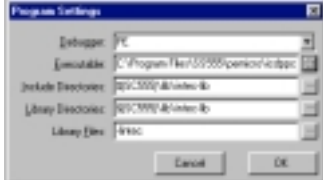
12.10 TB_isr.ipj - Time Base timer generates interrupts

The Time base has two registers that can be set for an interrupt. The program leap frogs between the two interrupting registers. Each time a register interrupts, the interrupt handler sets a semaphore. In the mainline program the interrupting reference counter is incremented, the register is set to interrupt at a later time and the semaphore is cleared. This program also demonstrates setting up the TB clock.


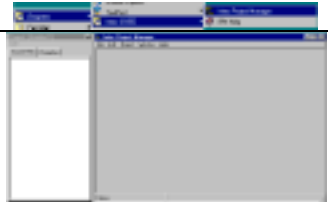
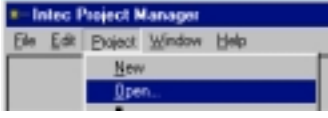

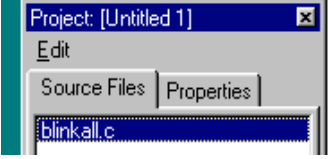
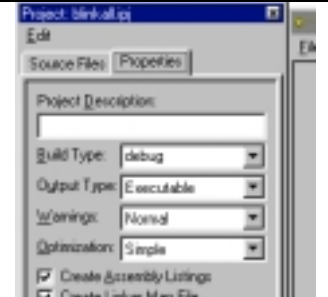
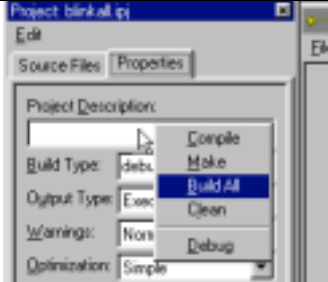
12.11 WDT.ipj - Watchdog Timer setup and serviced

The mainline of this program services the watchdog timer. The setup of the watchdog timer takes place in the special hwhooks.h that is located in this project.

13 Setting up ICDPPCZ within IPM

<p>Open IPM. This can either be done through double clicking on the Intec Project manager icon on the desktop or finding the shortcut in:Start-Programs-SS555-Intec Project Manager.</p> <p>Open IPM</p>	
<p>IPM will start. Note that there are two windows visible, the project window and the IPM main window. Note that there are two windows, the Project window and the Intec Project Manager (main) window.</p>	
<p>Under the IPM window Click on File-Program Settings.</p> <p>Open File-Program Settings.</p>	
<p>Click on the down arrow to view the pull down menu for the Debugger field. Change the Debugger to PE.</p> <p>Change the Debugger field to PE.</p>	
<p>Click on the .. icon to the right of the Executable field to set the path to the ICDPPC program.</p> <p>Click on the .. beside the Executable field.</p>	
<p>Change to the directory that ICDPPC was installed to. The P&E default is: C:\pemicro\ICDPPCZ\ The Intec Automation Inc. recommended default is: C:\Intec\SS555\pemicro\ICDPPCZ\ Change to the ..\pemicro\ICDPPCZ\ directory.</p>	
<p>Click on icdppcz.exe then click open.</p>	
<p>Notice that this is also the location where include directories, library directories and library files can be globally set. Any settings in this window will be applied to all projects.</p> <p>Click OK to save the changes.</p>	

14 Compiling a Program in IPM

<p>Open IPM. This can either be done through double clicking on the Intec Project manager icon on the desktop or finding the shortcut in:Start-Programs-SS555-Intec Project Manager.</p> <p>Open IPM</p>																			
<p>IPM will start. Note that there are two windows visible, the Project window and the Intec Project Manager (main) window.</p>																			
<p>In the IPM main window Click on Project->Open.</p> <p>Locate directory C:\Intec\ss555\gcc\example\blinkall.</p>																			
<p>Double-click on blinkall.ipj.</p> <p>Load project blinkall.ipj.</p>																			
<p>If the Project window was not open, it will now open and show all the files contained in the project. In this case, the only file is "blinkall.c".</p> <p>Double-click on "blinkall.c" in the Project window to open this file in the editor..</p>																			
<p>Select the Properties tab on the Project window and set the properties:</p> <table border="0" style="width: 100%;"> <tr> <td style="width: 30%;">Build-Type: debug</td> <td>To load in SRAM for easiest debugging.</td> </tr> <tr> <td style="padding-left: 20px;">Heap-off-chip</td> <td>To load into flash EPROM – final build.</td> </tr> <tr> <td style="padding-left: 20px;">Heap-on-chip</td> <td>To load into flash EPROM – use internal RAM</td> </tr> <tr> <td>Output Type: Executable</td> <td>Compile to an executable program.</td> </tr> <tr> <td style="padding-left: 20px;">Library</td> <td>Compile into a library (*.a) file.</td> </tr> <tr> <td>Optimization: None</td> <td>Recommended for debugging.</td> </tr> <tr> <td style="padding-left: 20px;">Full</td> <td>Code will be compact, fast and unrecognizable in the debugger.</td> </tr> <tr> <td>Create Assembly Listings</td> <td>They can be informative.</td> </tr> <tr> <td>Create Linker Map File</td> <td>Required for debugging. Not required for flash.</td> </tr> </table> <p>Select debug Executable Normal None</p>	Build-Type: debug	To load in SRAM for easiest debugging.	Heap-off-chip	To load into flash EPROM – final build.	Heap-on-chip	To load into flash EPROM – use internal RAM	Output Type: Executable	Compile to an executable program.	Library	Compile into a library (*.a) file.	Optimization: None	Recommended for debugging.	Full	Code will be compact, fast and unrecognizable in the debugger.	Create Assembly Listings	They can be informative.	Create Linker Map File	Required for debugging. Not required for flash.	
Build-Type: debug	To load in SRAM for easiest debugging.																		
Heap-off-chip	To load into flash EPROM – final build.																		
Heap-on-chip	To load into flash EPROM – use internal RAM																		
Output Type: Executable	Compile to an executable program.																		
Library	Compile into a library (*.a) file.																		
Optimization: None	Recommended for debugging.																		
Full	Code will be compact, fast and unrecognizable in the debugger.																		
Create Assembly Listings	They can be informative.																		
Create Linker Map File	Required for debugging. Not required for flash.																		
<p>Right-click anywhere on the Project window and click on Make or Build All:</p> <p>Make: Only recompiles files that have changed since last build.</p> <p>Build All: Recompile and re-links all files in a project, regardless of date.</p> <p>Press F1 with cursor over field of interest, for context sensitive help.</p> <p>Right-click on the Project window and click on Build All.</p>																			

A new window will open under the Project and Main windows to display messages resulting from the build operation. These may be status messages and error messages.

Click on error messages to display the offending lines of code in the Main window editor.


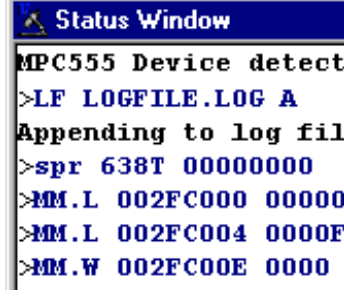




The debugger (selected in File->Program Settings) can be launched by right clicking on the Project window and clicking Debug on the pop-up menu. The debugger can also be launched by selecting Project->Debug from the menu bar in the Main window, or, by simply pressing the <F5> key.

Launch the debugger through any of these methods.



15 Loading/Debugging with ICDPPC

<p>Start the ICDPPC debugger as described earlier, or by clicking on its shortcut, or, from the start menu.</p> <p>Start ICDPPC</p>							
<p>Before any code can be loaded into SRAM, the SRAM must be set up. This and a whole lot of other things are set up by a script file called STARTUP.ICD which is executed by the debugger when it first starts up ..provided it can find this file.</p> <p>It is useful to have a copy of STARTUP.ICD in every development directory containing files that will be loaded onto the SS555 with the debugger.</p> <p>Run STARTUP.ICD (File->Play Macro)</p>	 <pre> Status Window MPC555 Device detect >LF LOGFILE.LOG A Appending to log fil >spr 638T 00000000 >MM.L 002FC000 00000 >MM.L 002FC004 0000F >MM.W 002FC00E 0000 </pre>						
<p>Load a *.695 file by clicking the HL Load button (right of the “?” on the toolbar) and selecting the corresponding file.</p> <p>Select and load “blinkall.695”.</p>							
<p>A number of messages will appear in the status window that describe the progress of the load operation.</p> <p>Type “gotil main<Enter>”. It will appear at the bottom of the Status Window.</p>							
<p>The program will execute until it reaches the instruction corresponding to the symbol “main”, corresponding to function main(), and the C source code will become visible in the Source window.</p>							
<p>It is possible to step to the next line of C code by typing “ss <Enter>”. This may correspond to many assembler instructions.</p> <p>Right-clicking on a subsequent line opens a pop-up menu with a number of choices:</p> <table border="0" data-bbox="235 1270 1015 1365"> <tr> <td>Toggle Breakpoint</td> <td>Writes or removes a hardware breakpoint.</td> </tr> <tr> <td>Set PC at Cursor</td> <td>Changes the Program Counter to this address.</td> </tr> <tr> <td>Gotil Address at Cursor</td> <td>Handy for stepping over functions.</td> </tr> </table> <p>Type “g<Enter> to start execution of the program. Press <F1> to stop.</p>	Toggle Breakpoint	Writes or removes a hardware breakpoint.	Set PC at Cursor	Changes the Program Counter to this address.	Gotil Address at Cursor	Handy for stepping over functions.	
Toggle Breakpoint	Writes or removes a hardware breakpoint.						
Set PC at Cursor	Changes the Program Counter to this address.						
Gotil Address at Cursor	Handy for stepping over functions.						

Please refer to the SS555 User’s Manual for information about using the hardware and software included with the SS555-DK. The location of the SS555 User’s manual is:

..\SS555\Documentation\SS555 User’s Manual\SS555UM.pdf.