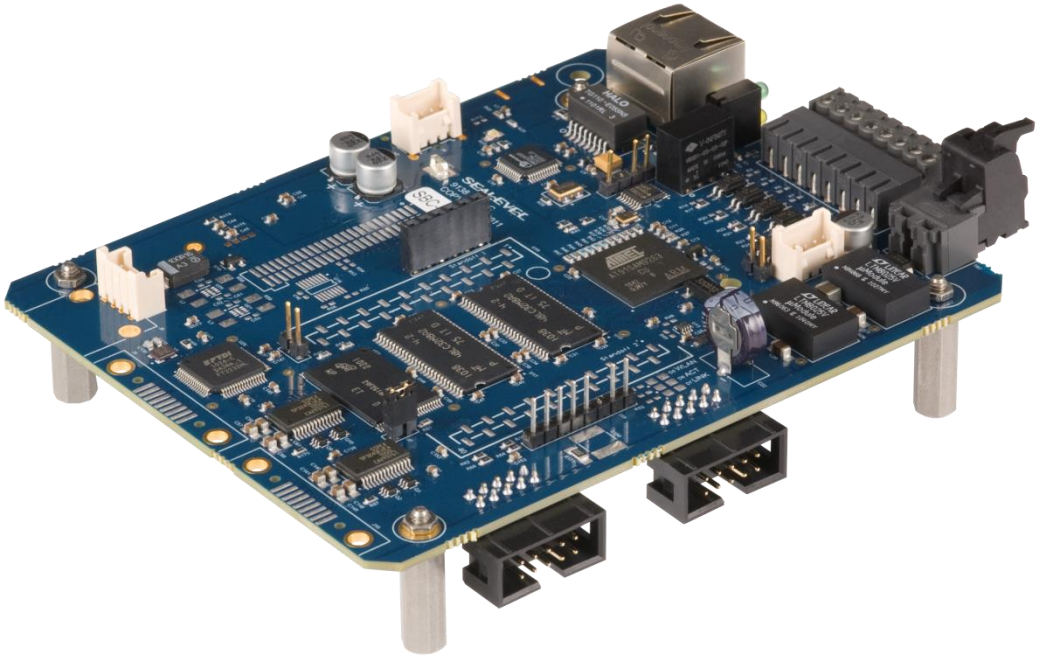
SBC-R9-2100 ARM9 RISC Single Board Computer

User Manual



Item#R92100-SBC

Contents

Safety Instructions

ESD Warnings

Electrostatic Discharges (ESD)

A sudden electrostatic discharge can destroy sensitive components. Proper packaging and earthing rules must therefore be observed. Always take the following precautions.

- Transport boards and cards in electrostatically secure containers or bags.
- Keep electrostatically sensitive components in their containers, until they arrive at an electrostatically protected workplace.
- Only touch electrostatically sensitive components when you are properly earthed.
- Store electrostatically sensitive components in protective packaging or on anti-static mats.

Grounding Methods

The following measures help to avoid electrostatic damages to the device:

- Cover workstations with approved antistatic material. Always wear a wrist strap connected to workplace as well as properly grounded tools and equipment.
- Use antistatic mats, heel straps, or air ionizers for more protection.
- Always handle electrostatically sensitive components by their edge or by their casing.
- Avoid contact with pins, leads, or circuitry.
- Turn off power and input signals before inserting and removing connectors or connecting test equipment.
- Keep work area free of non-conductive materials such as ordinary plastic assembly aids and Styrofoam.
- Use field service tools such as cutters, screwdrivers, and vacuum cleaners which are conductive.
- Always place drives and boards PCB-assembly-side down on the foam.

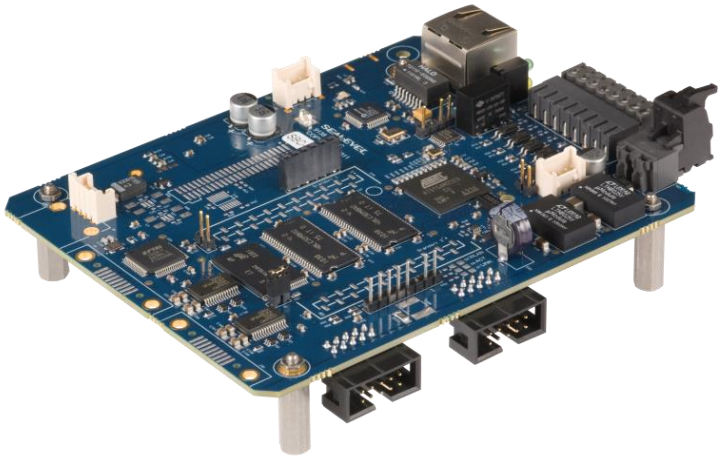
Introduction

The SBC-R9-2100 is an application-ready platform for your next product design. The system is based on the 200MHz Atmel AT91SAM9263 microcontroller boasting a 32-bit ARM® instruction set for maximum performance. With up to 256MB RAM and 256MB NAND Flash memory, the unmatched I/O features of the SBC-R9-2100 extend the possible uses beyond traditional ARM applications.

To provide the fastest time to market, the Windows CE 6.0 BSP binary and low-level drivers for system I/O are included. Additionally, the SBC-R9 software package is equipped with the Sealevel Talos I/O Framework, which offers a high-level object-oriented .NET Compact Framework (CF) device interface. This interface provides an I/O point abstraction layer with built-in support for the specific needs of analog and digital I/O such as gain control and debouncing.

Features

- Atmel AT91SAM9263 ARM9® Processor
- Up to 256MB SDRAM and 256MB NAND Flash Memory
- 10/100 BaseT Ethernet
- USB 2.0 Port and USB Device Port
- CAN Bus Interface
- On-board Serial and Digital I/O
- Stereo Audio Output
- Compatible with Windows Embedded CE 6.0 and Linux
- 7-30VDC Wide Input Range Power Supply
- Low Power Requirements
- Power and Status LED Indicators



Before You Get Started

Advisory Conventions

Warning - The highest level of importance used to stress a condition where damage could result to the product or the user could suffer serious injury.

Important – The middle level of importance used to highlight information that might not seem obvious or a situation that could cause the product to fail.

Note – The lowest level of importance used to provide background information, additional tips, or other non-critical facts that will not affect the use of the product.

What's Included

The SBC-R9-2100 is shipped with the following items. If any of these items are missing or damaged, please contact Sealevel for replacement.

- SBC-R9 ARM9 Embedded RISC Single Board Computer with standoffs
- CD with Setup files, CE runtime image, Talos .NET Framework, application samples and documentation
- Microsoft® Windows® CE 6.0 Core license

QuickStart Kit

The SBC-R9-2100 QuickStart Kit (Item# SBC-R9-2100-KT) is available, which includes the most common accessories. For applications with specialized hardware requirements, developers can use the SBC-R9-2100 as a platform for application development while Sealevel designs a customized target system specific to the user's application requirements.

The SBC-R9-2100-KT includes the following items:

- SBC-R9-2100 ARM9 Embedded RISC Single Board Computer
- CD with Setup files, CE runtime image, Talos .NET Framework, application samples and documentation
- Microsoft Windows CE 6.0 Core License
- TR123 – 100-240VAC to 12VDC @ 2.5A, wall mount power supply
- CA473 – USB Type A to USB Type Mini-B, device cable
- CA429 – R9 serial debug cable
- CA246 – CAT5 patch cable, 6' length
- Two CA152 – 10-pin IDC ribbon cable to DB9M connector
- CA464 – 3.5mm headphone jack audio cable

Optional Items

Depending upon your application, you are likely to find one or more of the following items useful with the SBC-R9. All items can be purchased from our website (www.sealevel.com) by calling our Sales team at +1 864-843-4343.

Cables

USB Type A to USB Type B, 72" in Length - Device Cable (Item# CA473)	
The CA473 is a 72" standard USB device cable that connects USB peripherals with a 5-pin Mini Type B connector to the Type A connector on a host computer. The CA179 is USB 2.0 compliant and is compatible with USB 1.1 and 1.0 devices.	
CAT5 Patch Cable, 7' in Length – Blue (Item# CA246)	
Standard 7' CAT5 UTP Patch Cable (RJ45).	
10-Pin IDC Ribbon Cable to DB9 Male Connector, 14" in Length (Item# CA152)	
10-Pin IDC Ribbon Cable terminates to DB9 Male Connector, 14" in Length.	
R9 Serial Debug Cable, 72" in Length (Item# CA429)	
The CA429 is a 72" serial debug cable with a 1x4 connector on one end and a standard DB9F connector on the other end. The DB9F connector is compatible with any standard RS-232 DB9M serial port.	
Stereo Audio Cable, 6" in Length (Item# CA464)	
The CA464 is a 6" stereo audio cable with a 4 pin Molex (Molex 35507-0400) connector on one end and a standard stereo 3.5mm barrel jack connector on the other end.	

Power Supplies

100-240VAC to 12VDC @ 2.5A, Wall Mount Power Supply (Item# TR123)	
The TR123 is a wall mount (wall wart style) power supply rated for 100-240VAC input and 12VDC output at 2.5 amps. The 72" cable has tinned leads for use with products that have screw terminals for input power. The white line or printing on the insulation indicates positive polarity.	
100-240VAC to 12VDC @ 4A, Desktop Power Supply w/ 5mm 2-Position Terminal Block (Item# TR135-US)	
Perfect for powering a Relio R9 embedded computer system or a stack of SeaI/O modules, the TR135 is a desktop style power supply rated for 100-240VAC input and 12VDC output at 4 amps. The power supply features over voltage, short circuit and over current protection. Refer to the appropriate product manual for information useful in calculating the proper power supply requirements. Choose a country specific IEC power cord under the Accessories tab. For the US version of this power supply, please order the TR135-US.	
100-240VAC to 12VDC @ 4A, Desktop Power Supply w/ 5mm 2-Position Terminal Block (Requires Power Cord) (Item# TR135)	
Perfect for powering a Relio R9 embedded computer system or a stack of SeaI/O modules, the TR135-US is a desktop style power supply rated for 100-240VAC input and 12VDC output at 4 amps. The power supply features over voltage, short circuit and over current protection. It includes a US power cord (Item# CA248). Refer to the appropriate product manual for information useful in calculating the proper power supply requirements. For the international version of this power supply, please order Item# TR135 and select the correct power cord from the list of accessories.	

Product Overview

Specifications

Processor

Atmel (AT91SAM9263) — 200MIPS RISC Processor
16KB Data Cache, 16KB Instruction Cache, Write Buffer
Integrated Memory Management Unit (MMU)

Memory

Up to 256MB SDRAM (128 MB Standard)
256MB NAND Flash

Audio Output

Stereo Output with Amplified Headphone Drive Capability

Bus Interfaces

10/100 BaseT Ethernet
USB Device Port
USB 2.0 Port
CAN Bus
Dedicated RS-485 Expansion

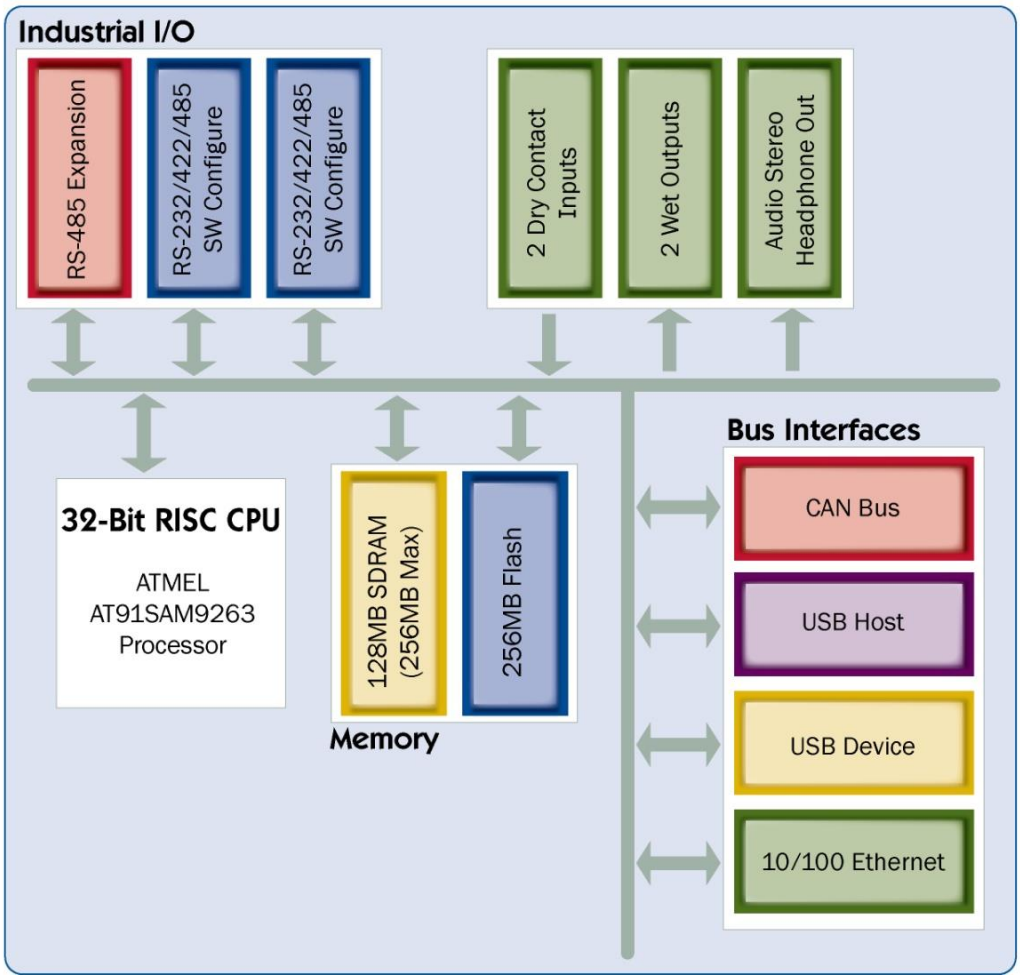
Industrial I/O

Two Software Configurable RS-232/422/485 Ports
Two Optically Dry Contact Inputs
Two Wet Outputs (5V @ 30mA Max.)

Indicators

Dual LED Indicators for Power and Status

Block Diagram



See Appendix B for the Connector Reference Table, which details the connectors, jumpers and test points located on the SBC-R9-2100.

Technical Description

Memory

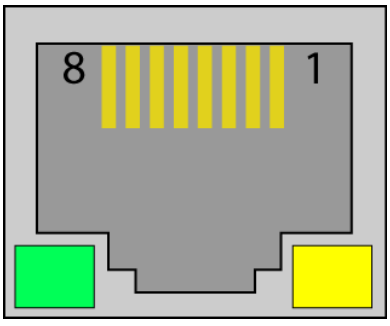
The SBC-R9-2100 base configuration includes 128MB SDRAM and 256MB NAND Flash. For memory intensive applications, the board can also be ordered preconfigured with 256MB SDRAM.

Ethernet

The SBC-R9-2100 includes a 10/100 BaseT Ethernet interface accessed via the RJ45 connector located at (J2).

Connector: J2
Manufacturer: Amphenol
Part Number: RJHSE-5384
Description: RJ-45 Through-Hole Jack, Tab up with LEDs
Mates with: RJ-45 Plug

Pin	Signal
1	TX+
2	TX-
3	RX+
4	NC
5	NC
6	RX-
7	NC
8	NC

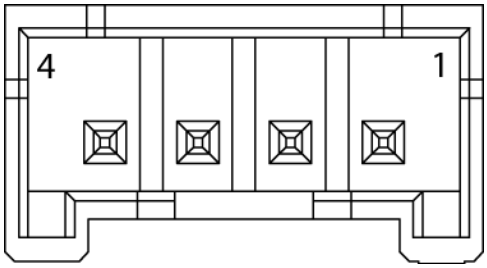


USB

The SBC-R9-2100 provides one USB 2.0 host port and one device port. The host USB port is located at (J18). The device USB port is located at (J1).

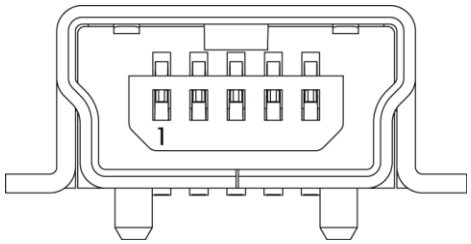
Port: USB-A Host Port
Connector: J18
Manufacturer: Molex
Part Number: 35362-0450
Description: 2.00mm (.079") Pitch Sherlock™ Wire-to-Board Header, Vertical, with Positive Lock, 4 Circuits
Mates with: Molex 35507-0400 Sherlock™ Wire-to-Board Housing with Molex 0502128100 2.00mm (.079") Pitch Crimp Terminals

Pin	Signal
1	5VDC
2	Data-
3	Data+
4	GND



Port: USB Mini-B Device Port
Connector: J1
Manufacturer: Hirose
Description: USB mini Type B

Pin	Signal
1	5VDC
2	Data-
3	Data+
4	NC
5	GND

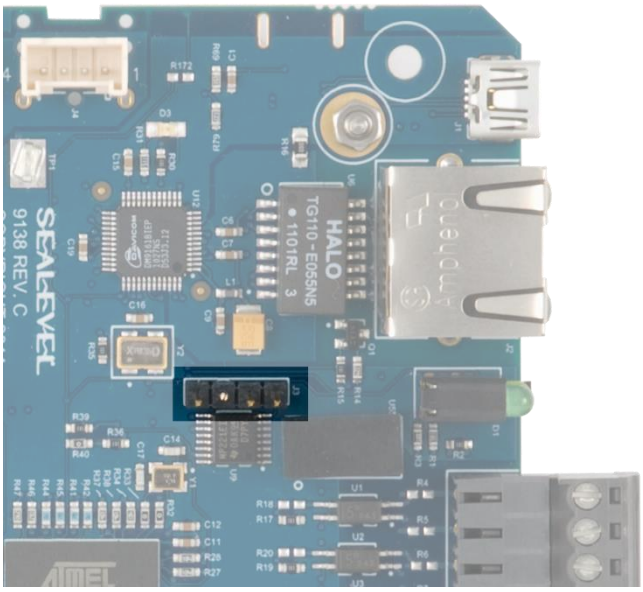


Serial Debugging

Debug the SBC-R9-2100 through the RS-232 debug port using the CA429 cable.

Connector: J3
Manufacturer: Amp/Tyco
Part Number: 9-146278-0-04
Description: Header, 0.100" Polarized 4 pos, pin 3 Removed

Pin	RS-232
1	RX
2	GND
3	Key
4	TX

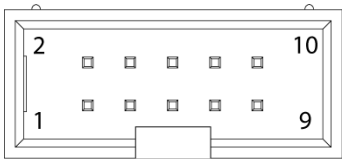


Serial Communications

Connect to a variety of serial peripherals via the SBC-R9's software configurable RS-232/422/485 ports. Ports are located at connector (P2 Bottom) and (P3 Bottom).

Connector: P2, P3
Manufacturer: Sullins
Part Number: SBH11-PBPC-D05-RA-BK
Description: Box Header, 0.100" Polarized 10 pos (2x5)
Mates with: SFH213-PPPC-D05-ID-BK-M181 or equivalent

Pin	RS-232	RS-422/485
1	DCD	RX+
2	DSR	NC
3	RX	RX-
4	RTS	NC
5	TX	TX-
6	CTS	NC
7	DTR	TX+
8	RI	NC
9	GND	GND
10	NC	NC



COM Port Assignments

Serial Port	Assignment
RS485 Expansion Port	COM1
SERIAL1	COM2
SERIAL2	COM3

CA152 Accessory Cable

The DB9 pin out is achieved using the CA152 accessory cable.

Pin	DB9
1	1
2	6
3	2
4	7
5	3
6	8
7	4
8	9
9	5
10	NC

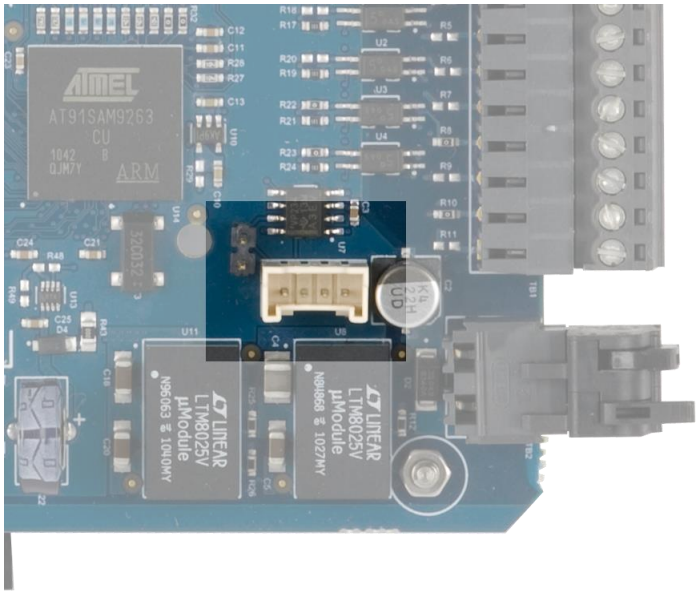
CAN Bus

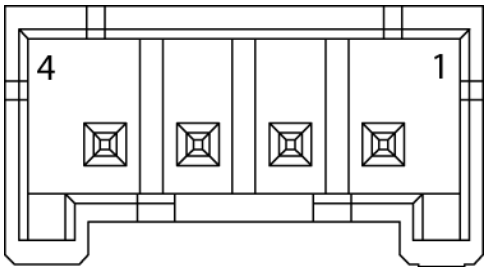
Connect directly to a Control Area Network (CAN) bus via connector (J12). A Molex 4-pin vertical 2mm locking header is used for the connection.

J12 connector designator is located on the bottom of the board.

Connector: J12
Manufacturer: Molex
Part Number: 35362-0450
Description: 2.00mm (.079") Pitch Sherlock™ Wire-to-Board Header, Vertical, with Positive Lock, 4 Circuits
Mates with: Molex 35507-0400 Sherlock™ Wire-to-Board Housing with Molex 0502128100 2.00mm (.079") Pitch Crimp Terminals

Pin	Signal
1	CAN High
2	GND
3	CAN Low
4	Shield



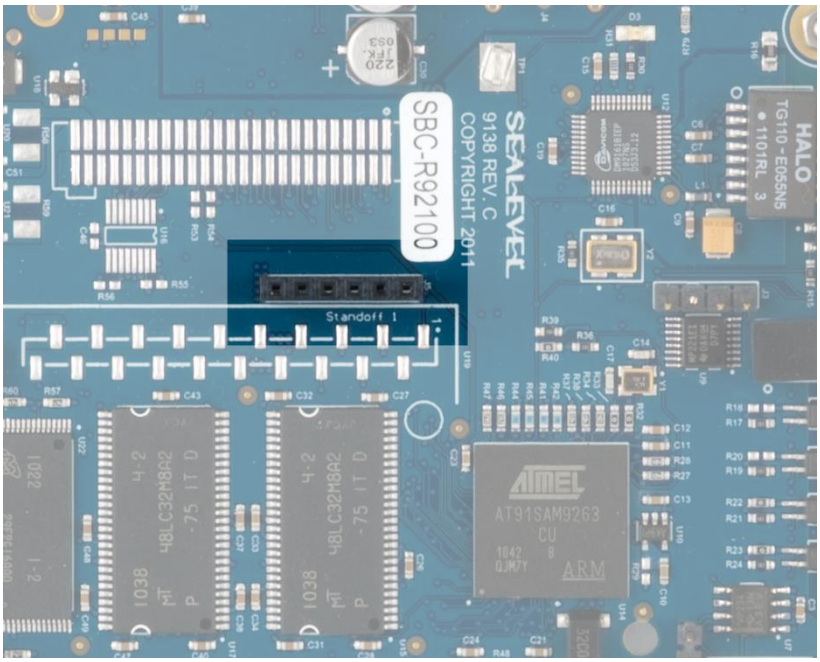


RS-485 Expansion

The SBC-R9-2100 provides a RS-485 Expansion Port. The port is available through the 6-pin vertical 100mil header (J5). This offers a convenient option for adding additional expansion modules from the Seal/O product line.

Connector: J5
Manufacturer: TE Connectivity
Description: 100mil (2.54mm) Pitch Single Row Pins, Vertical, 6 Circuits
Mates with: 100mil Single Row Socket, 6 Circuits

Pin	Signal
1	5VDC
2	NC
3	RS-485+
4	RS-485-
5	GND
6	NC

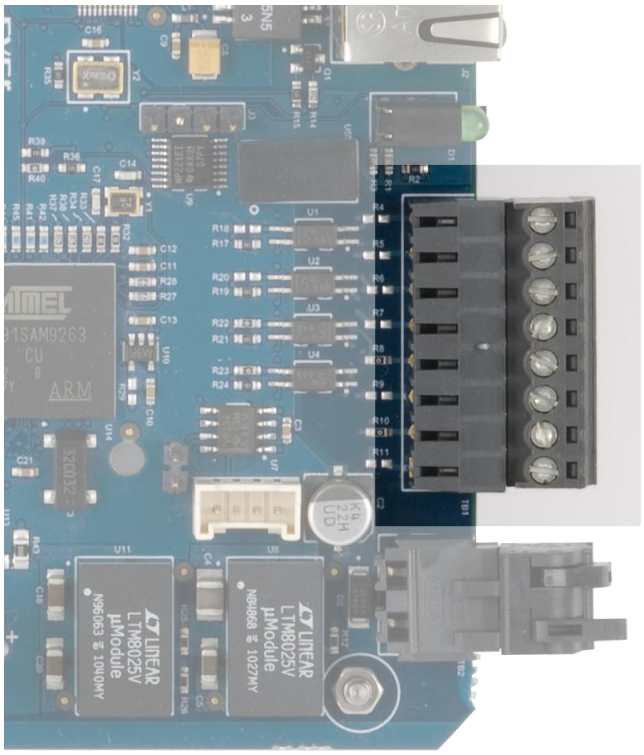


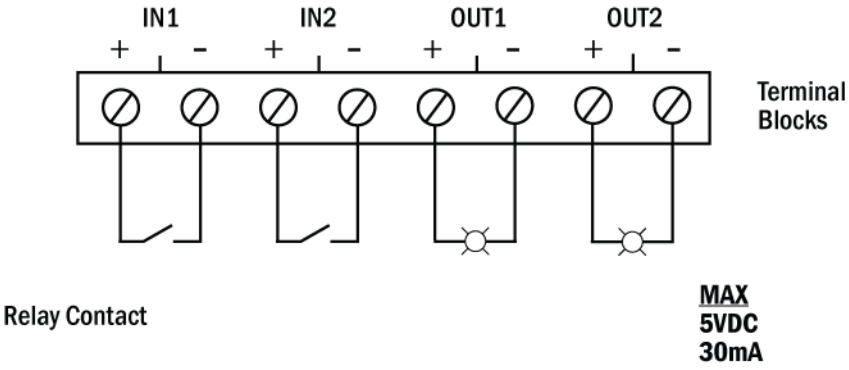
Input/Output connector

The R92100 has both dry contact inputs and isolated wet outputs, all found on connector (TB1). Both input and outputs are isolated with 1.0KV isolation.

Manufacturer: Weco
Part Number: 110-M-111/08
Description: Terminal Block 8 position 3.5mm spacing
Mates with: Weco 110-A-111/08 8 position screw-terminal plugs (provided)
Connector: TB1

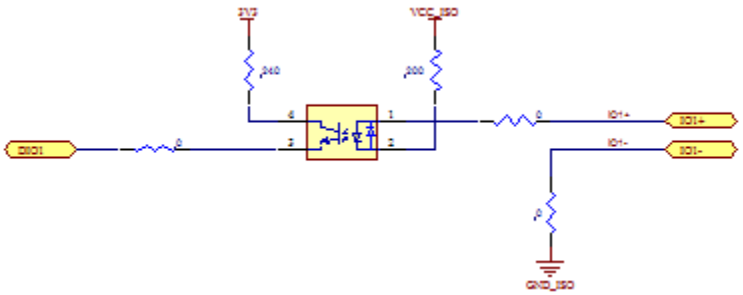
Pin	Signal
1	Input 1+
2	Input 1-
3	Input 2+
4	Input 2-
5	Output 1+
6	Output 1-
7	Output 2+
8	Output 2-

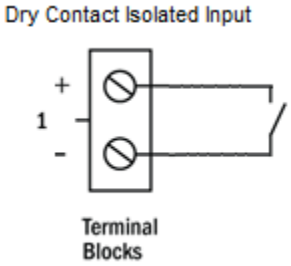



IN1
IN2
OUT1
OUT2
+
-
+
-
+
-
+
-
Terminal Blocks
Relay Contact
MAX
5VDC
30mA

Isolated Dry Contact Inputs

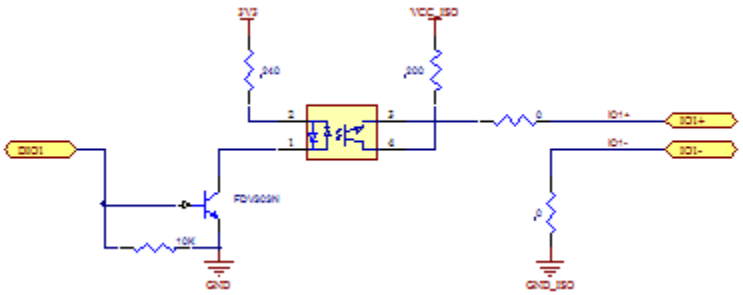
Dry contact inputs allow for reading the state of relays, switches, and dry contact sensors. The SBC-R9-2100 provides two inputs with a 5V source and ground that detect the closure of the contact with up to 10mA of current.

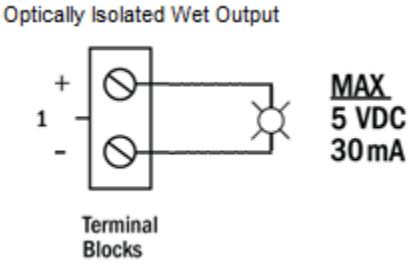




Isolated Wet Outputs

The isolated wet outputs on the SBC-R9-2100 allow for powering external indicators or relays while maintaining isolation. Two outputs provide 5VDC at 30mA to LEDs, buzzers, solid state relays, or other low power devices.



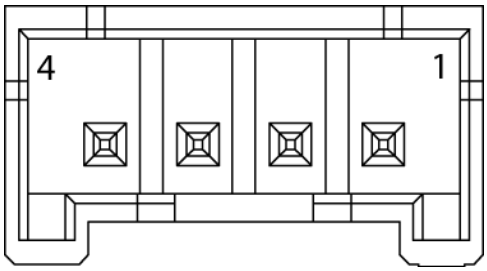


Audio Output

Output an alarm signal or play music through the R92100-SBC's stereo audio output port. Use the CA464 cable accessory to break out the (J4) connector to a standard 3.5mm headphone jack.

Connector: J4
Manufacturer: Molex
Part Number: 35362-0450
Description: 2.00mm (.079") Pitch Sherlock™ Wire-to-Board Header, Vertical, with Positive Lock, 4 Circuits
Mates with: Molex 35507-0400 Sherlock™ Wire-to-Board Housing with Molex 0502128100 2.00mm (.079") Pitch Crimp Terminals

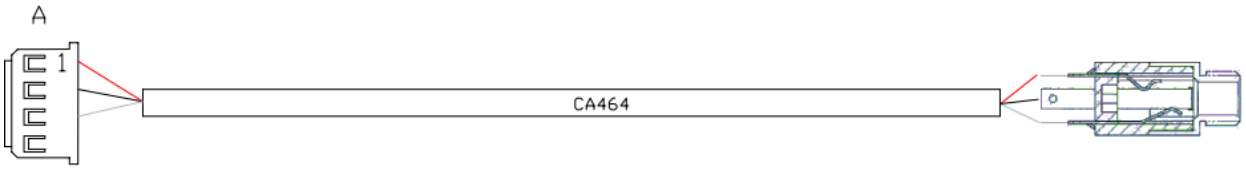
Pin	Signal
1	Left Audio
2	GND
3	Right Audio
4	NC



CA464 Accessory Cable

A 3.5mm stereo barrel jack connector can be achieved using the CA464 cable.

Pin	Barrel Jack
1	Tip
2	Sleeve
3	Ring
4	NC



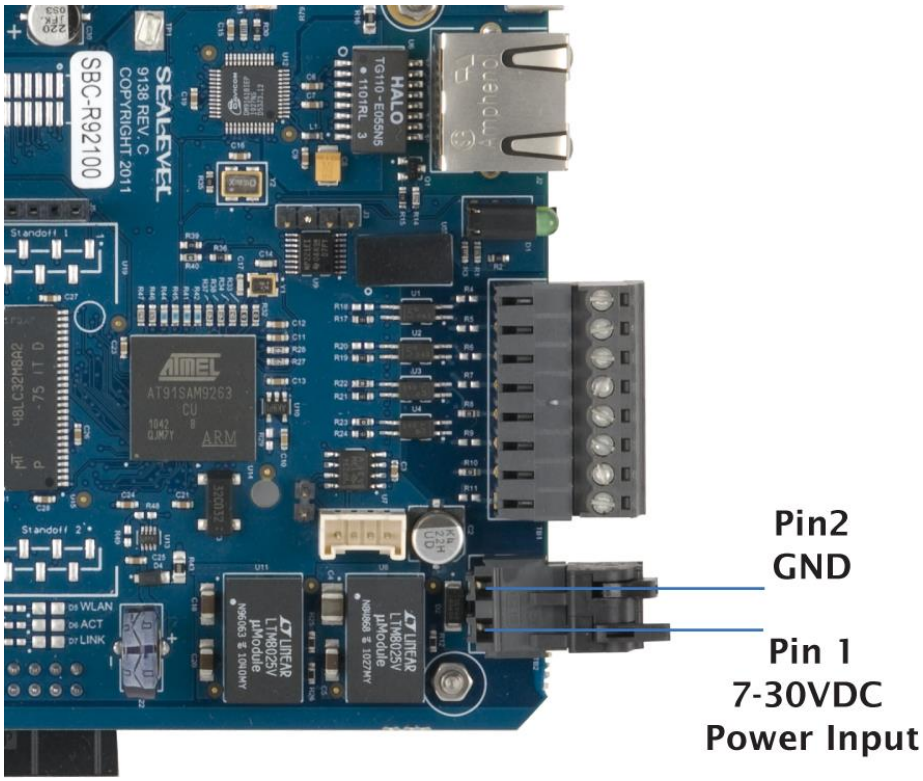
Power

The SBC-R9-2100 can be powered with the Sealevel TR123.

Connector: TB2
Manufacturer: Weco
Part Number: 121-M-111/02
Description: 5.08mm Pitch Friction Lock Header
Mates with: Weco 121-A-111/02

The SBC-R9-2100 integrated reverse polarity protection. User should still be sure that the power lead is connected to the proper pin. Reversing the polarity of the power input may damage your SBC-R9-2100.

Pin	Signal
1	7-30VDC
2	GND

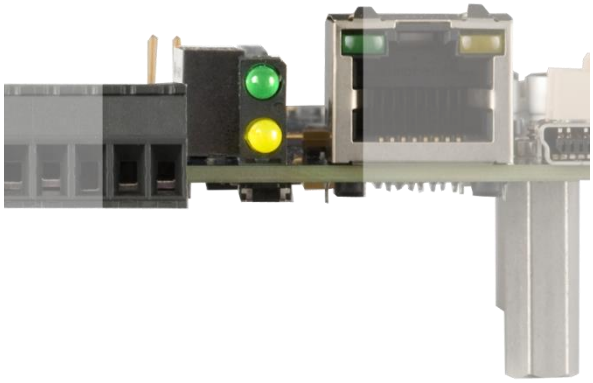


LED Indicators

The SBC-R9-2100 features two LED indicators for power and status. The Green LED (Top) is illuminated when power is applied to the board. The Yellow LED (Bottom) is a GPIO controllable indicator accessible through the TALOS API.

Designator: D9
Description: Dual Stacked LED Indicators

LED	Color	Signal
Top	Green	Power
Bottom	Yellow	Status



Software

SBC-R9-2100 Quick Start

Remove the contents from the box.

Insert the accompanying CD into your PC and run the installation program. This will install Talos Framework binaries, documentation, and examples on your PC (See Figure 1.)

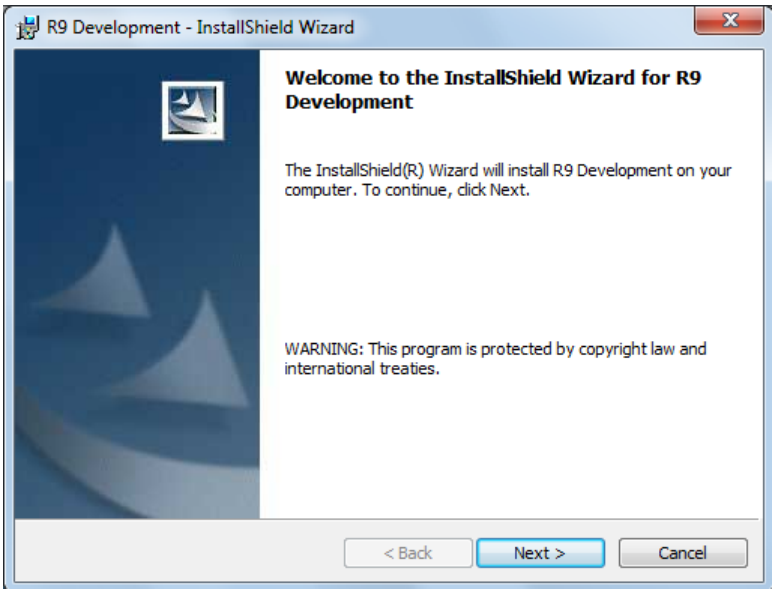


Figure 1. Installation Wizard

After installation, the package can be found in Windows by clicking Start → All Programs → Sealevel Systems → R9 Development.

The contents of the factory provided NAND Flash build will allow the SBC-R9-2100 to run Windows CE 6.0 OS when power is applied to the board.

To avoid accidental damage, be sure to follow proper ESD procedures by grounding yourself and the board.

Apply power to the SBC-R9-2100 by connecting the TR123 tinned leads to the SBC-R9-2100 input power connector screw terminals, noting proper polarity. Attach the other end of the TR123 into a 120VAC wall outlet.

Use a CA473 USB Mini Type B device cable and connect the USB Mini Type B connector to the SBC-R9-2100. Connect Type A connector into the host PC. *(See Figure 2.)*

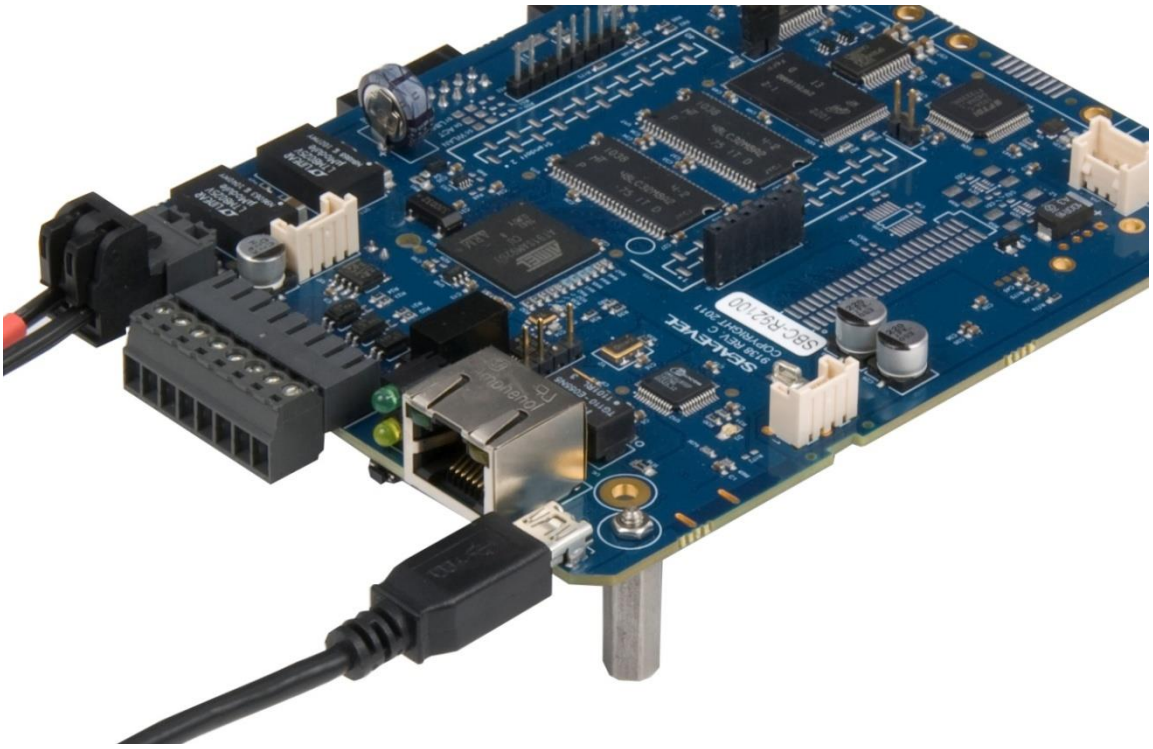


Figure 2. TR135 Tinned Leads and Type B USB Connector.

You are now ready to set up a USB communication interface between the host PC and the SBC-R9-2100. Depending on which operating system you are using – Windows 7, Vista or XP – the setup experience will vary.

Windows Device Center

If your host PC is running Windows Vista or later and you are connected to the Internet, then Windows Mobile Device Center software will install automatically. If you are not connected to the Internet but have obtained the Windows Mobile Device Center software manually, then running their setup will achieve the same result. (See Appendix A.)

After installation, a negotiation will begin between the PC and the SBC-R9-2100 board and the device center connection screen will appear. (See Figure 3.)

Figure 3. Device Center Connected

Using your mouse, select "Connect without setting up your device". The idea is to explore the file system on the SBC-R9-2100 without setting up synchronization with contacts, calendar, or e-mail. Now choose "File Management → Browse the contents of your device" from the screen. (See Figure 4.)

Figure 4. Device Center File Management

This action opens a standard Windows Explorer where the default file contents of the SBC-R9-2100 can be read or written to. (See Figure 5.)

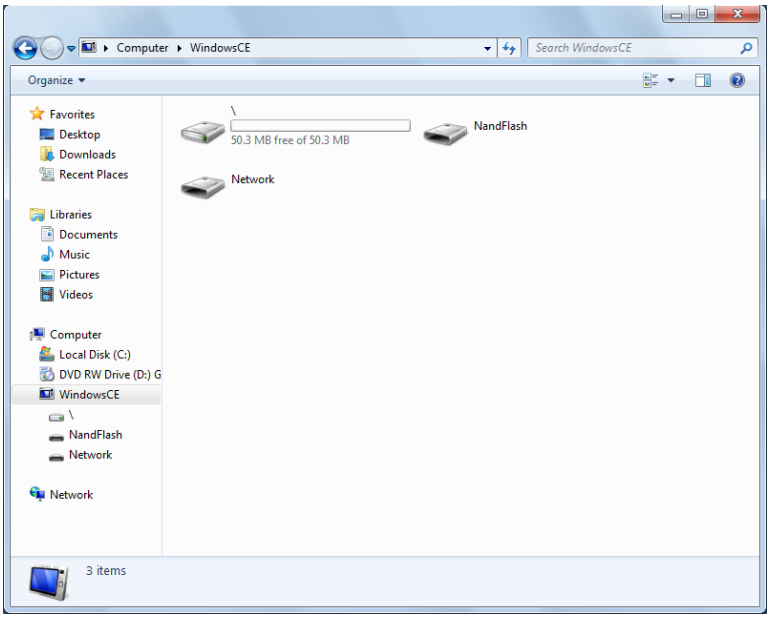


Figure 5. Contents of SBC-R9-2100

Windows ActiveSync for XP

If your host PC is running Windows XP, ActiveSync is required to establish connection to the SBC-R9-2100. ActiveSync differs from Windows Mobile Device Center in that having an internet connection will not establish an automatic download and installation. For installation procedures, refer to Microsoft's website. (See Appendix A). After installation, a negotiation will begin between the PC and the SBC-R9-2100 board, and the "New Partnership" dialog will appear. (See Figure 6.)

Figure 6. ActiveSync New Partnership Dialog

Using your mouse, select “No” and then select “Next”. The ActiveSync main dialog will appear. Select the “Explore” icon. This action opens a standard Windows Explorer where the default file contents of the SBC-R9-2100 can be read or written. (See Figure 7.)

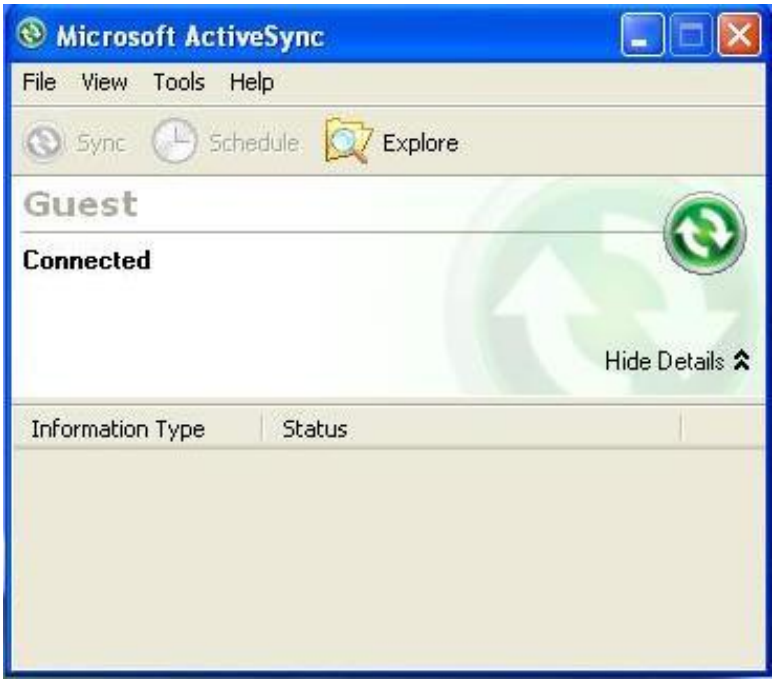


Figure 7. ActiveSync Main Dialog

Connection Complete

You are now ready to set up a complete development environment for building and debugging smart device applications and libraries. The next section guides you by example using Microsoft Visual Studio.

Programming Using the .NET Compact Framework

Application Development

INTRODUCTION

With .NET Compact Framework coupled with our Talos .NET Framework, C# and VB.NET programmers can develop powerful embedded applications on the SBC-R9-2100 such as mobile, robotics, home automation, industrial, and a broad range of other embedded applications. The low cost of licensing for Windows 6.0 CE has created an ideal environment to develop a new generation of embedded products around the SBC-R9-2100.

Our Talos Framework allows access to the more specific I/O sections of the SBC-R9-2100 development board such as digital output points, Audio Output, and the serial ports. A complete list of the API documentation can be found either in Windows by clicking Start → All Programs → Sealevel Systems → R9 Development → Talos Documentation.html.

Writing .NET applications for the SBC-R9-2100 is very similar to writing desktop or console applications for XP and Vista. The only difference is the amount of resources available. Because the memory footprint is smaller compared to a desktop computer, care should be taken where allocation of memory is concerned, such as large object creation.

REQUIREMENTS

- Visual Studio Professional 2005 or 2008
- .NET Compact Framework 3.5

GETTING STARTED

For this demonstration, we will construct a smart device console application using Visual C#. Start Visual Studio and select File → New → Project. A 'New Project' dialog will appear. Select a project type of Visual C# → Smart Device. Select 'Smart Device Project' as the Template. Make sure the combo box has .NET Framework 3.5 selected. Type the name of the project. In this case, call it *HelloWorld*. (See Figure 8.)

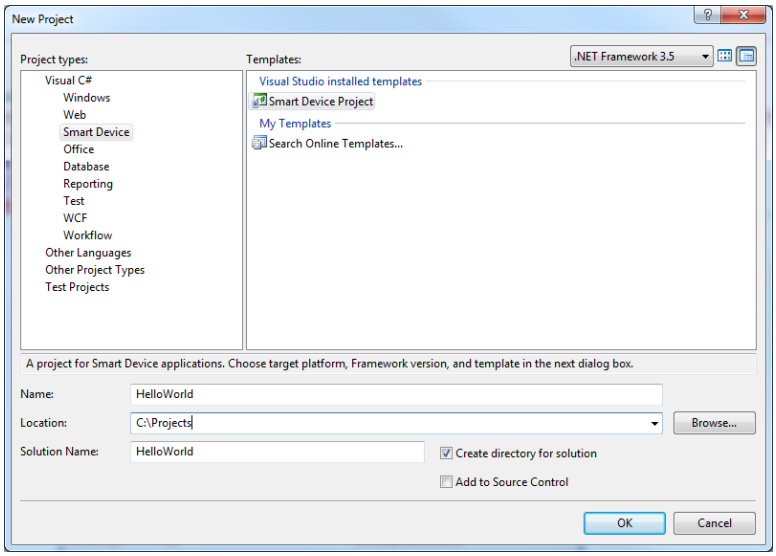


Figure 8. Visual Studio New Project Dialog

Click the "OK" button. The next configuration screen allows you to select the type of project you are creating. Select "Windows CE" for the target platform, .NET Compact Framework version 3.5 and select the "Console Application" icon for the template. (See Figure 9.)

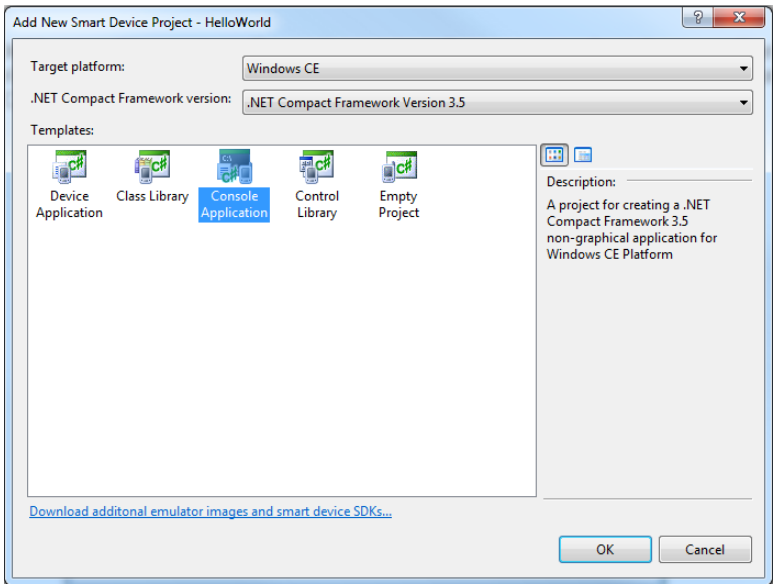


Figure 9. Visual Studio Add Smart Device Dialog

Once you have selected all of the configuration options, click the "OK" button. You will now see a console application template called *HelloWorld* in Visual Studio. (See Figure 10.)

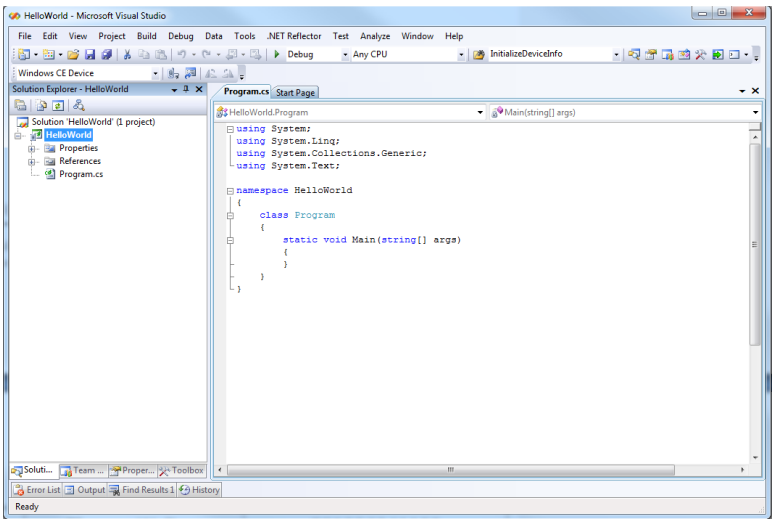


Figure 10. Visual Studio Main Window

We can now add the references to the Talos Framework. Right click on the “References” and select the "Add Reference…" selection. (See Figure 11.)

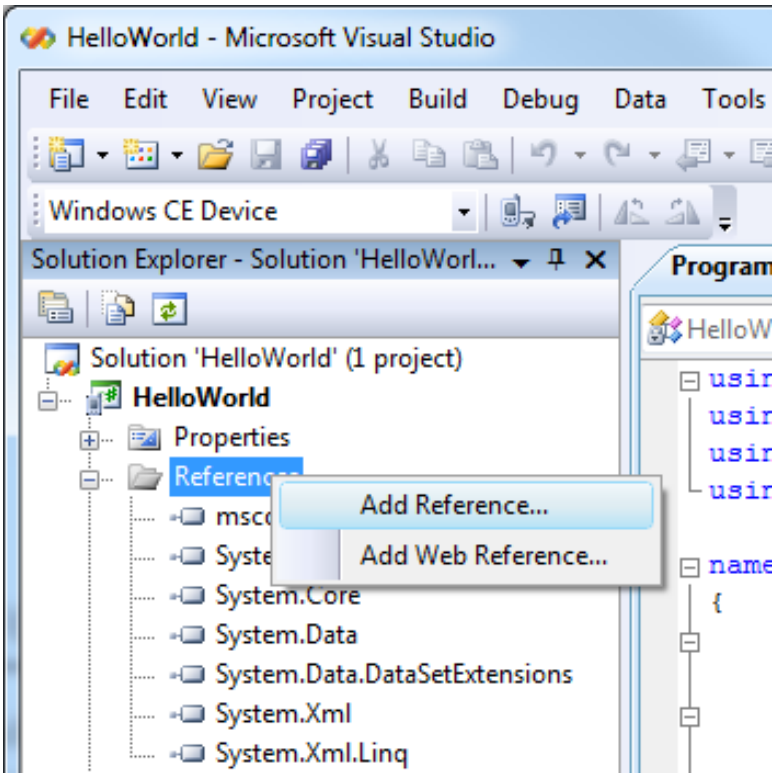


Figure 11. Adding References to Project

An ‘Add Reference’ dialog will appear. Click on the ‘Browse’ tab then search for the installed library path “C:\Program Files\Sealevel Systems\R9 Development\Assemblies”. If you don’t see a list of the

R9 libraries as shown in Figure 12, then refer to the SBC-R9-2100 QuickStart section for software installation details. While holding down the CTRL key, click on both "SLCorLib.dll" and "Talos.dll". Click the “OK” button. (See Figure 12.)

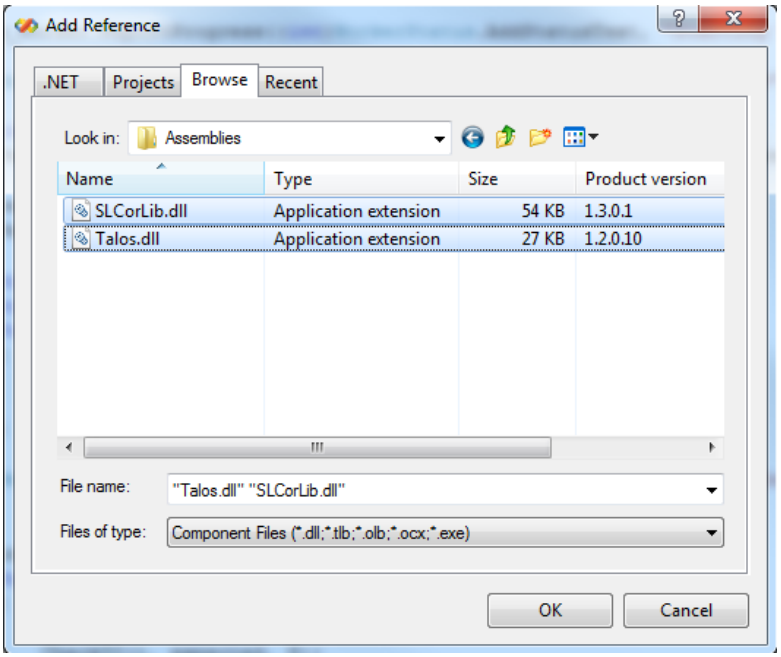


Figure 12. Core Library References

Both DLLs should appear in your “References” list. (See Figure 13.)

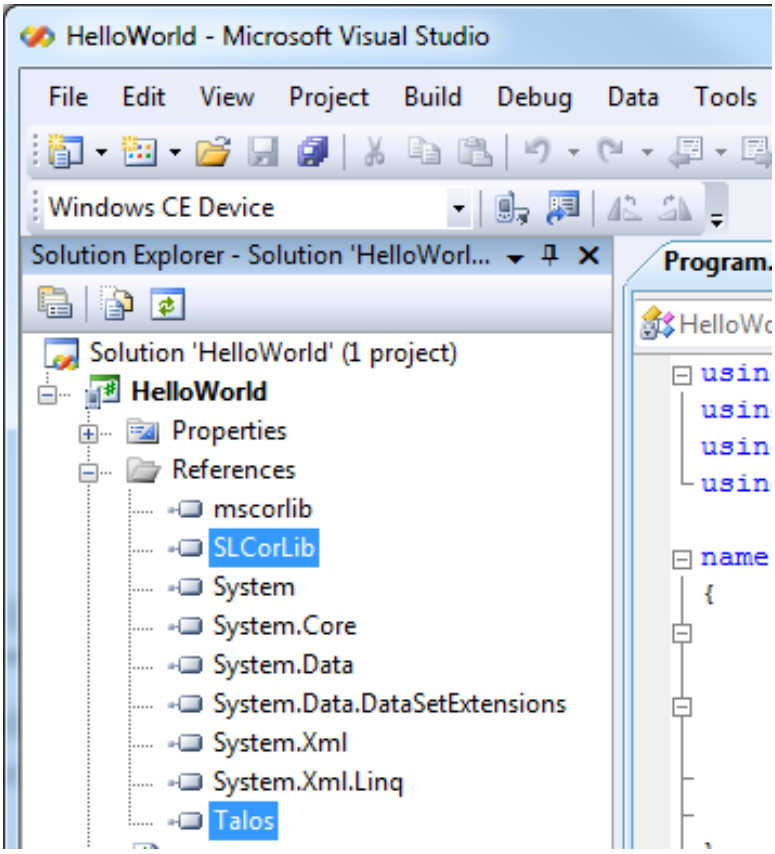


Figure 13. Verification of Added Library References

Now that the Talos Framework has been referenced, you have access to all the I/O points exposed on the SBC-R9-2100 device.

For this simple HelloWorld application, we will just echo the string “Hello World” in the console window. This can be accomplished by adding the following code to the automatically created Program::Main() method. This code will echo “Hello World” and then pause for 5 seconds.

```
static void Main(string[] args)
{
      Console.WriteLine("Hello World");
      System.Threading.Thread.Sleep(5000);
}
```

From Visual Studio’s menu bar, select “Build → Build HelloWorld”. After the build process has completed select from the same menu bar, “Build → Deploy HelloWorld”. A “Deploy HelloWorld” dialog will appear for you to choose the appropriate target. Choose “Windows CE Device” then press the ‘Deploy’ button. (See Figure 14.)

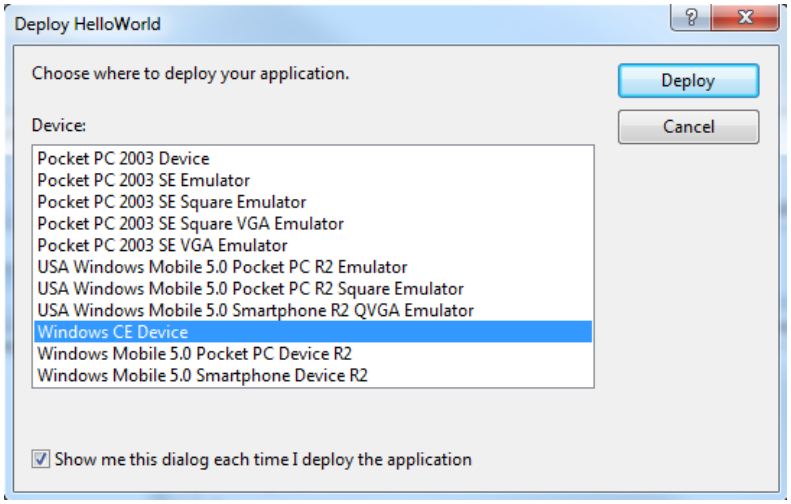


Figure 14. Choose Windows CE Device and Deploy

After the deployment phase, select "Debug->Start Without Debugging" from the Visual Studio menu bar. A console will appear to display the "Hello World" message. After 5 seconds, the window will automatically close.

Examples can be found from the installation directory under '..\R9 Development\Samples\C#' and '..\R9 Development\Samples\VB.NET'.

Application Debugging

Introduction

This guide details the process of debugging an application developed for the SBC-R9-2100 embedded IO system. The SBC-R9-2100 development platform easily integrates into standard Microsoft development tools to make the debugging process extremely easy. The following sections detail the requirements to begin debugging an application on Microsoft Windows 7, Vista, or XP.

Requirements

- Microsoft Windows Mobile Device Center using Vista or ActiveSync using XP
- Microsoft Visual Studio Professional 2005 or 2008
- USB Cable or Ethernet connection

Debugging your SBC-R9-2100 applications is a simple process that requires a USB cable or Ethernet connection, Microsoft device synchronization software, and Visual Studio. Depending on your version of Windows, you will need to follow a different process to install the device synchronization software as outlined in the SBC-R9-2100 Quick Start section.

Debugging an Application

Once the SBC-R9-2100 has been successfully attached to your PC, it is easy to begin debugging an application on the SBC-R9-2100. This section will demonstrate how to attach the Microsoft Visual Studio debugger to the SBC-R9-2100, show the use of breakpoints in the debugger, and show how to access useful information while debugging an application.

We will be using the GPIO example application found in the "samples" directory of the Talos Framework installation. The same methods will apply to any application you wish to debug on the SBC-R9-2100.

Attach the Debugger

Once your solution is opened, it is necessary to specify the device target that you would like to use in conjunction with the debugger. The default option is an emulator. Select "Windows CE Device" from the target device drop down. (See Figure 15.)

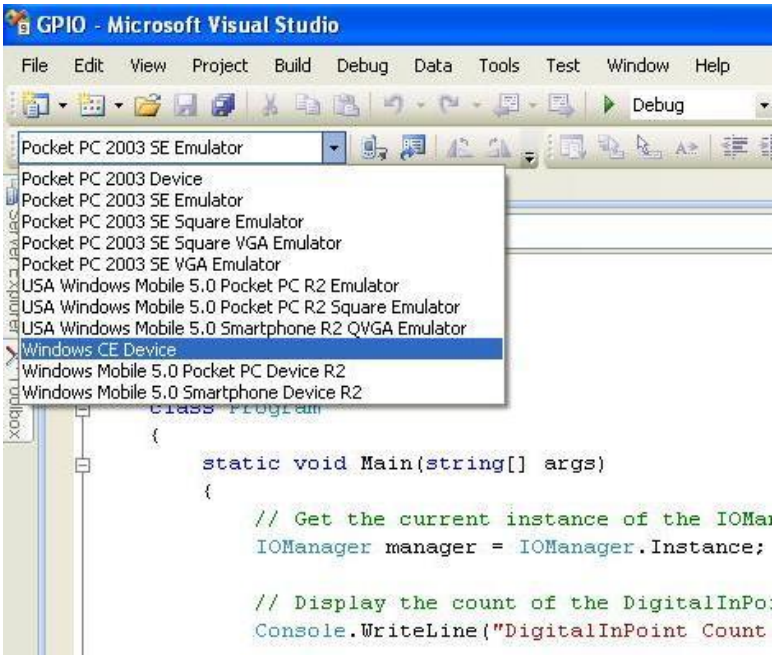


Figure 15. Device Target Selection

If you would like to use the faster Ethernet connection for debugging instead of the USB connection, refer to Appendix C.

Now select the "Connect to Device" icon to initiate synchronization between Visual Studio and the SBC-R9-2100 device. (See Figure 16.)

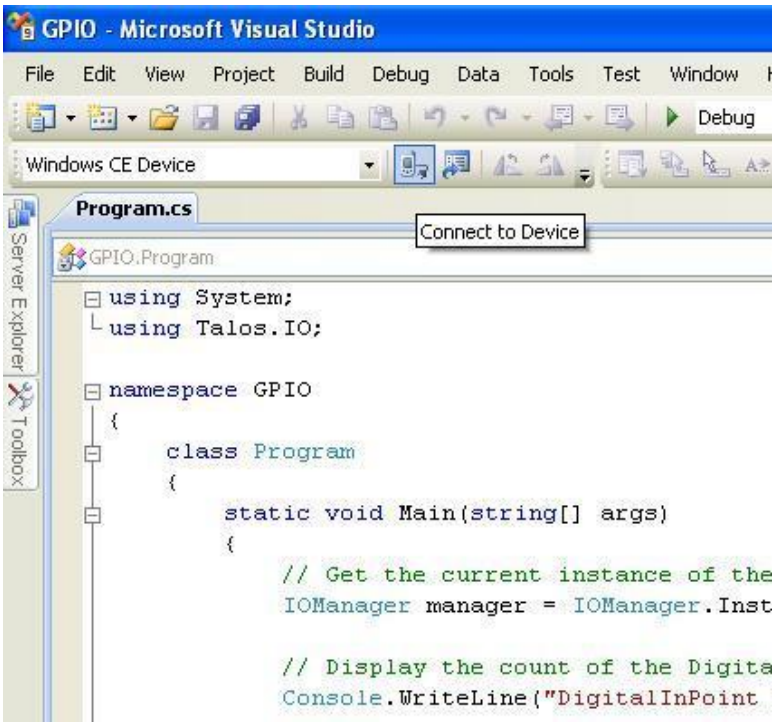


Figure 16. Connect to Device Icon

You should now see a connection dialog appear. (See Figure 17.)

Figure 17. Connection Status Dialog

Breakpoints

Setting breakpoints allows you to stop execution of your application at any point and examine the state of the application. A breakpoint may be set by selecting a line and pressing the "F9" hotkey. (See Figure 18.)

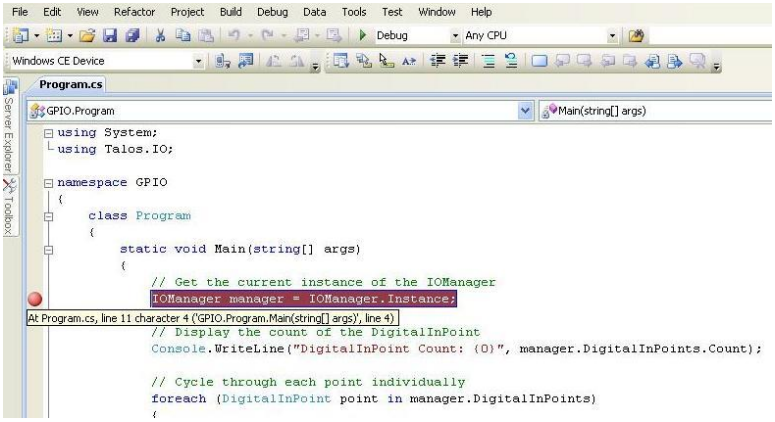


Figure 18. Breakpoint Selection

To begin debugging the application, click the "Start Debugging" button. (See Figure 19.)

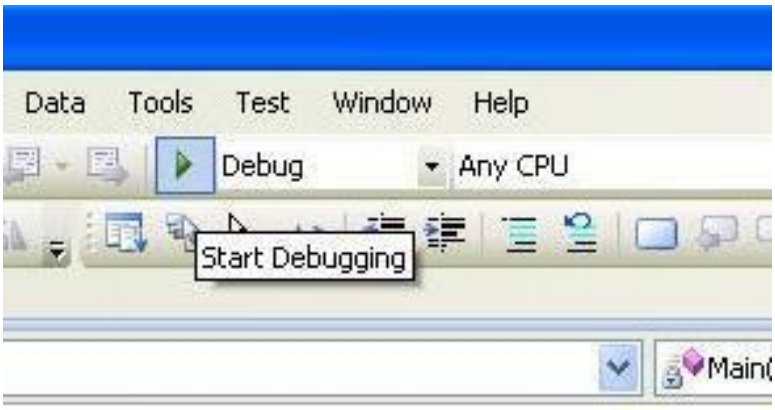


Figure 19. Run Debugger Icon

Although you previously set up the target device, upon starting the first debug session, you will be prompted to select the device to deploy the application to. Select the "Windows CE Device" as was done earlier when selecting the target. (See Figure 20.)

Figure 20. Target Deployment Dialog

Once the application is deployed to the SBC-R9-2100, it will begin execution. As soon as the first breakpoint is reached, execution will cease and you will gain full control over the running application. You may use the debugging options to continue execution, execute a single line, or execute multiple lines. You may view the status of each variable by either hovering over it with the cursor or by examining the windows at the bottom of Visual Studio just as you would with a desktop application. (See Figure 21.)

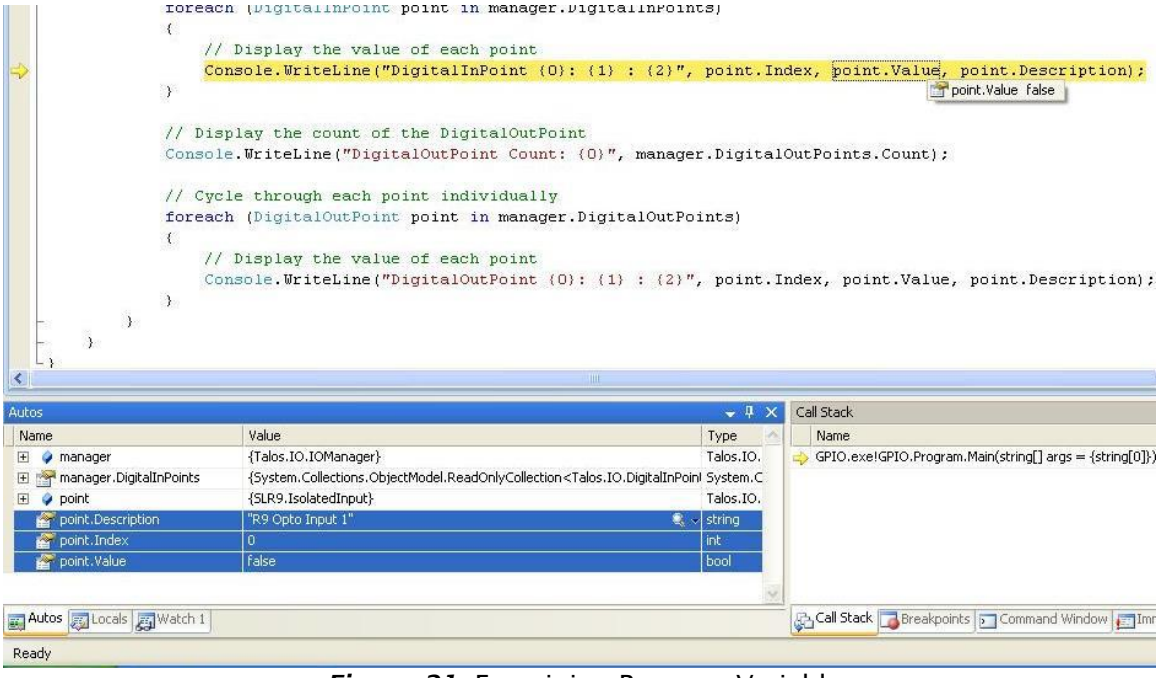


Figure 21. Examining Program Variables

Watching Variables

When program execution is halted due to a break point condition being met, the debugger will display the state of all local variables. In addition to those variables, class specific variables can be grouped together as a view to aid in debugging your application. This is accomplished by right clicking on a variable and selecting "Add Watch". Each addition appends a tab to the “Watch n” window where n is incremented for each variable added. (See Figure 22.) Each watch window provides a convenient tree type structure for viewing hierarchical class variables.

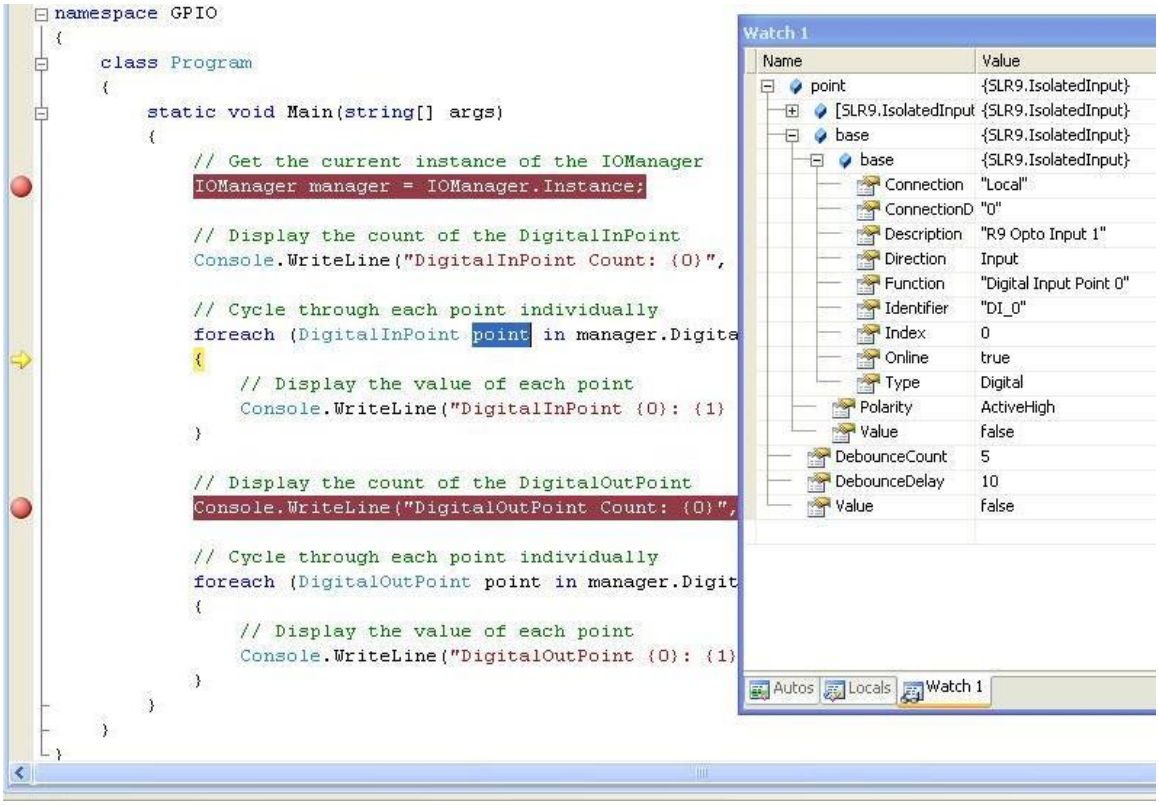


Figure 22. Watch View

Target Deployment and Execution

After your application is built using Visual Studio, either a debug or release executable, it may be desirable to copy it into NAND Flash. This would provide a means to store and execute your application without the need for connectivity to a host computer. The first step is transferring your application to a suitable directory in the on-board NAND Flash. To accomplish this you will need to establish connectivity via Windows Mobile Device Center or ActiveSync as outlined in the SBC-R9-2100 Quick Start section above.

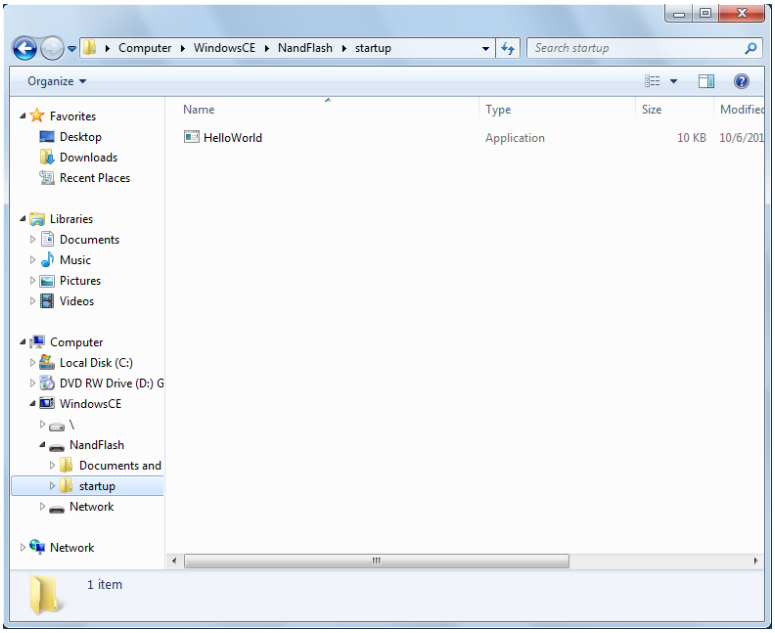


Figure 23. Application Placement

The SBC-R9-2100 Runtime image comes pre-loaded with a utility program called "SpringBoard". This utility provides a solution for automatically running your applications at startup. Rather than copying your application files to '/Windows/Startup/' – which is in volatile memory – the executables should be copied to `/nandflash/startup/'. After Windows CE runs, SpringBoard automatically starts applications in this startup directory.

SpringBoard also provides a way to specify program arguments by supplying an XML configuration file. You will need to create a simple XML file called "startup.xml". This XML file should consist of an element list each with an application name and the desired arguments for that application. (See Figure 24.) This file must reside in the following location '/nandflash/startup/startup.xml'.

If the startup.xml file is not found or is not desired, SpringBoard will still automatically run all the applications placed in the aforementioned directory structure, only no arguments will be passed to those applications.

```
<?xml version="1.0" encoding="utf-8" ?>
 <programs>
  <program name="sample1.exe" arguments="/i 1019 /w JSmith" />
  <program name="sample2.exe" arguments="-e 2000" />
  <program name="sample3.exe" arguments="/help" />
</programs>
```

Figure 24. startup.xml

Boot Sequence

Upon power-up, the SBC-R9-2100 follows a specific boot sequence. The initial sequence is “firstboot”. The firstboot process initializes the low level hardware and is responsible for loading the next sequence called “eboot”. Eboot provides a configuration menu for setting connection types and start up memory locations. Connection types include Ethernet and USB. Memory location is NAND Flash. Ultimately, eboot attempts to load and execute the OS runtime image based on the configuration settings found here.

The SBC-R9-2100 development board checks the raw data in the NAND Flash for a valid Eboot boot loader (eboot.nb0).

The SBC-R9-2100 ships with a NAND Flash programmed with the OS binaries listed below:

- FIRSTBOOT.nb0
- Eboot.nb0
- NK.nb0

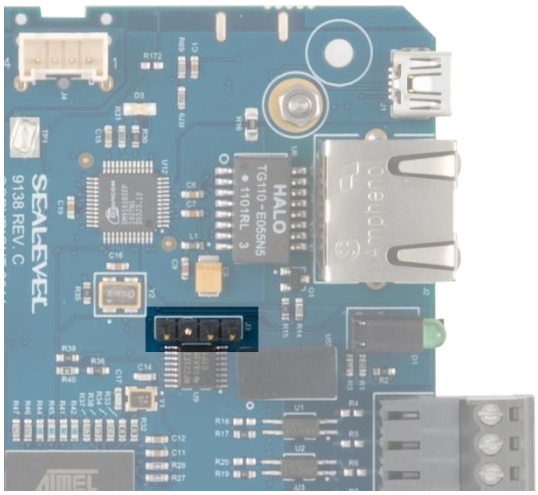
OS File Restoration

In the event that Sealevel produces updated OS file versions or a restore is desired, the OS files will need to be programmed to the NAND Flash. Please see the section labeled “Upgrading the OS runtime image on NAND Flash” below for more detail. The NAND Flash cannot be programmed until the existing OS runtime image has been removed. This can be accomplished through the debug port as described in the following section.

Using the Debug Port

This procedure requires an available RS-232 COM port or USB to RS-232 serial port adapter attached to a host PC, a SBC-R9-2100 Serial Debug cable (Item# CA429), and any telnet terminal client application such as PuTTY (See Appendix A). For this procedure, we will demonstrate the use of PuTTY.

Connect the 4-pin keyed female end of the SBC-R9-2100 RS-232 cable into the SBC-R9-2100 connector (J3). Connect the DB9 end of the SBC-R9-2100 RS-232 cable into an available serial port on the host PC.



Run PuTTY and select “Serial” from the Category section of the dialog. Identify the proper COM port number and always assign the speed (baud) equal to 115200. Set Data bits to 8, Stop bits to 1, Parity to None, and Flow control to None. (See Figure 25.)

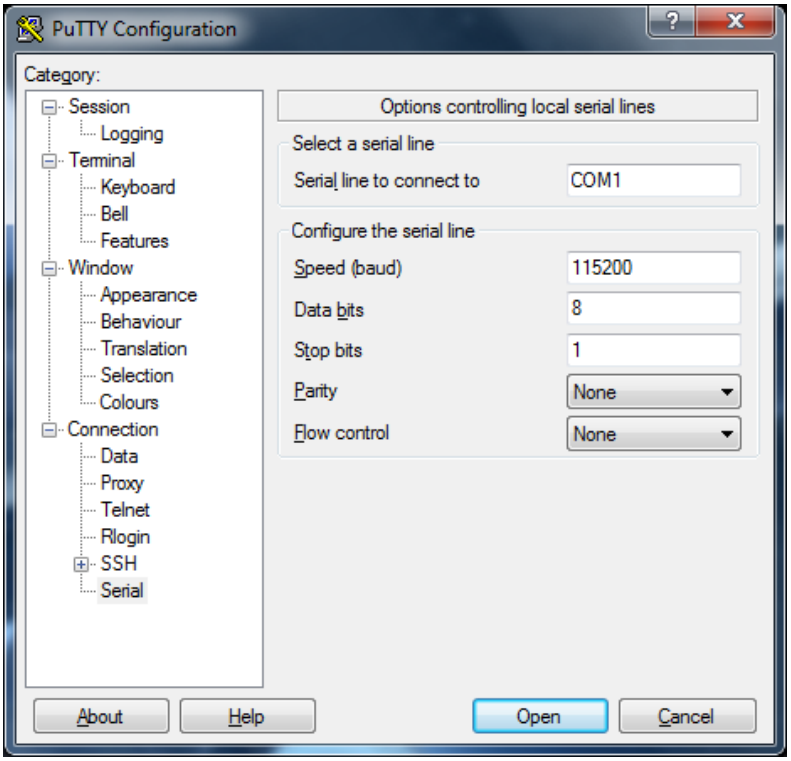


Figure 25. PuTTY Serial Configuration

Select “Session” from the Category section of the dialog. A saved session of this configuration can be performed to avoid reconfiguration in the future. Type a name for this session under “Saved Sessions”, then press the “Save” button. (See Figure 26.)

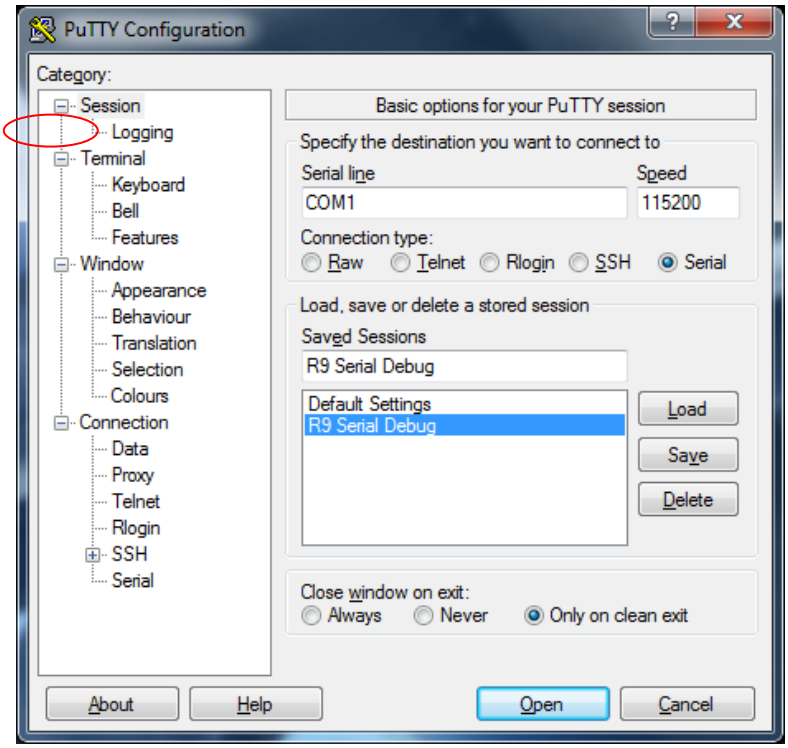


Figure 26. PuTTY Session Configuration

Press “Open” to start a new terminal session. A blank terminal window will appear. Debug messages may not appear until power is applied to the SBC-R9-2100. Press the reset button on the SBC-R9-2100 to display the Ethernet boot loader configuration screen. (See Figure 27.) When the unit boots, the following prompt on the debug port terminal will appear (no user input is required for booting):

```
“Press [ENTER] to download now or [SPACE] to cancel.
Initiating image download in 2 seconds"
```

Once the prompt period expires, the OS runtime will be loaded from NAND Flash into RAM and executed. At this point, the OS is running. (See Figure 27.)

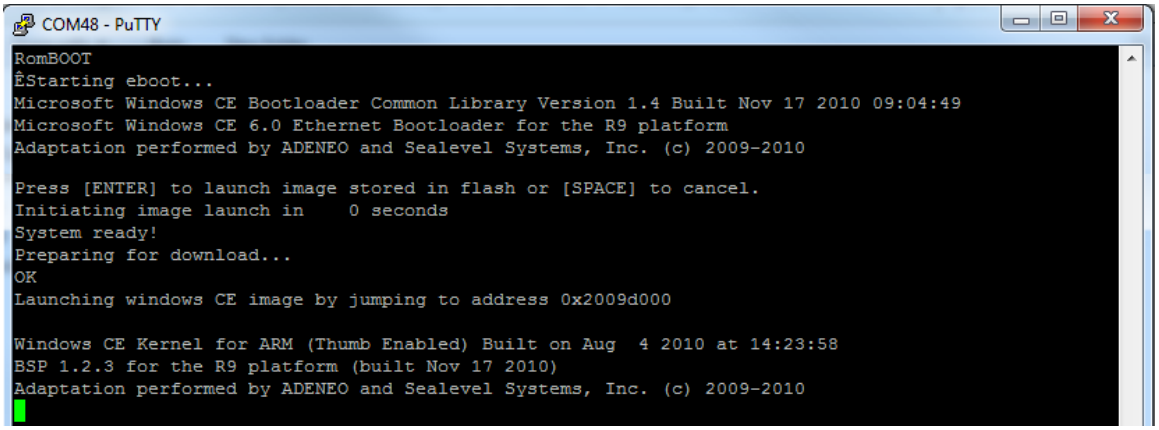


Figure 27. Application Debug Text Output

Eboot configuration settings can be modified by hitting the “space” key during the 2 second boot prompt period. When modifying the configuration, a menu such as the one below is displayed. (See Figure 28.)

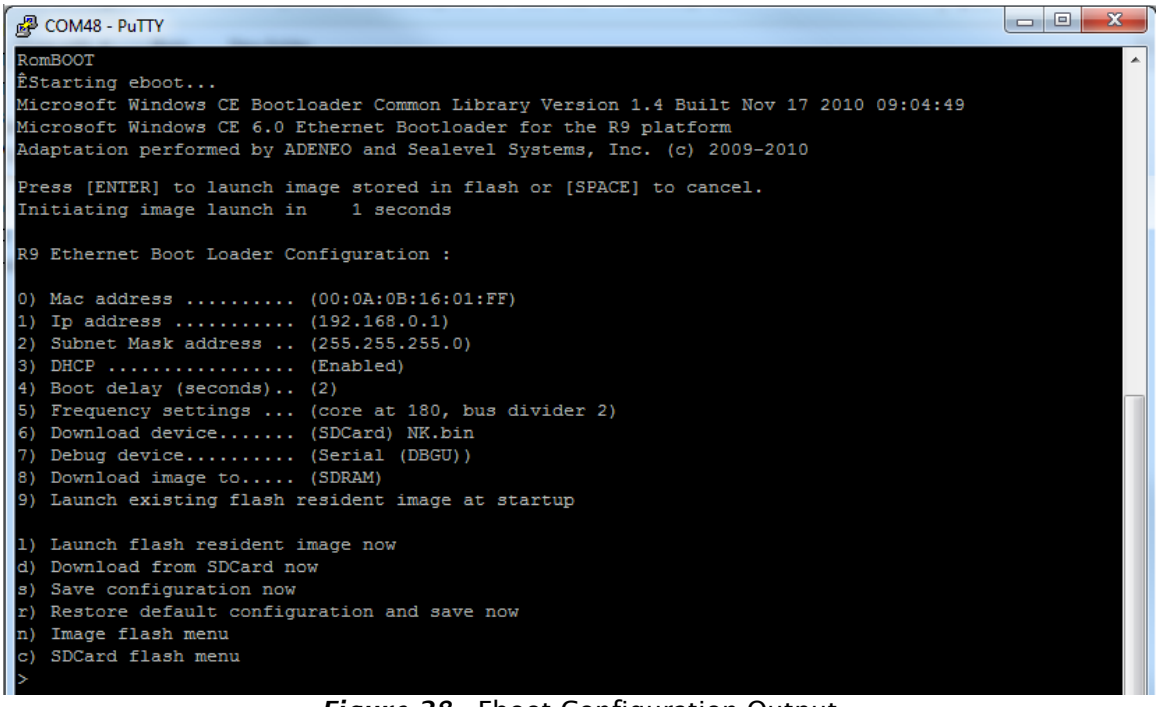


Figure 28. Eboot Configuration Output

ERASING THE NAND FLASH

Modifying any of these settings may render your SBC-R9-2100 unbootable.

When upgrading an existing OS runtime it is necessary to first erase the NAND Flash of a pre-programmed unit. This is accomplished through the "Image flash menu" ('n' key) in Eboot. The flash menu has an option to "Erase all sectors" of the NAND Flash ('1' key). (See Figure 29.)

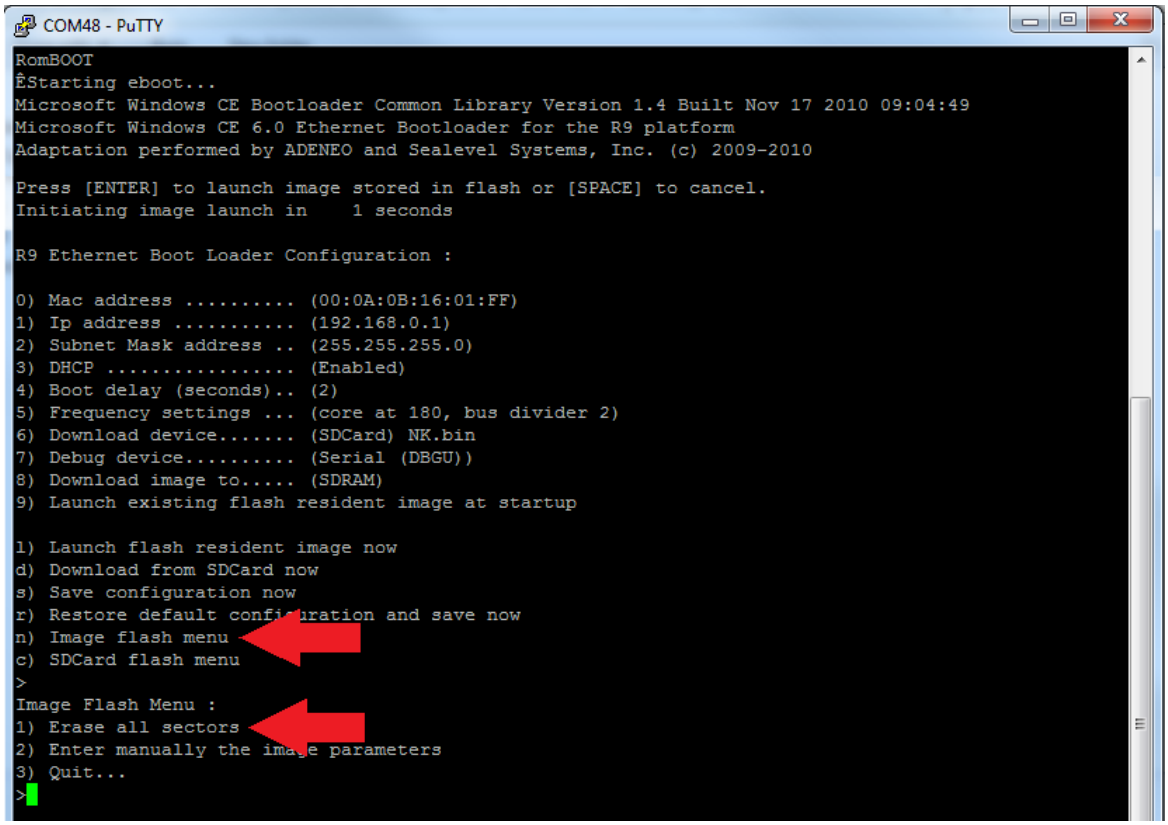


Figure 29. Eboot Image Flash Menu

The "Erase all sectors" option in Eboot will erase the entire NAND Flash, so be sure to back up any data you wish to save before attempting to erase the device.

Upgrading the OS Runtime Image on NAND Flash

Factory OS runtime images are stored in the "Boot Files" directory of the R9 Development installation (see Quick start guide). The OS runtime image present in the NAND Flash is programmed through the USB device port connection. Prior to programming an OS runtime, the existing image must be erased. The procedure to erase the NAND Flash is documented in the ERASING THE NAND FLASH portion of the Debug Port section above..

Once the NAND Flash has been erased, use a CA473 USB Mini Type B device cable and connect the USB Mini Type B connector to the SBC-R9-2100. Connect the Type A connector into the host PC. *(See Figure 30.)*

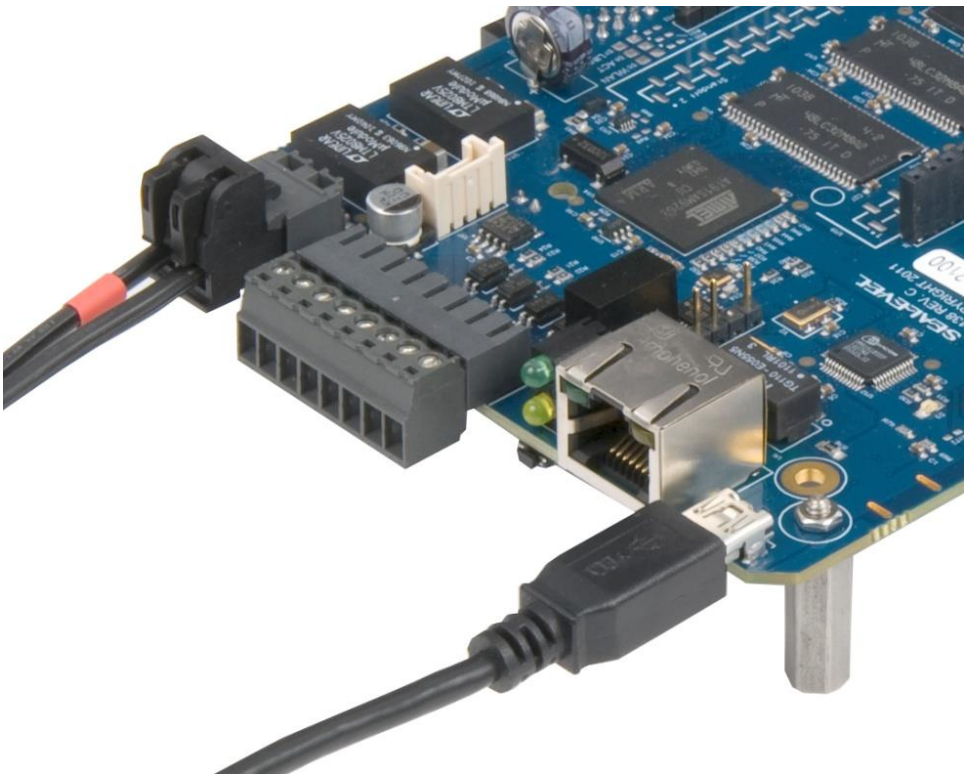


Figure 30. TR135 Tinned Leads and Type B USB Connector

In Microsoft Windows 7, the device is recognized as a GPS camera and will typically enumerate as a COM port. Verify the numeric assignment of the COM port in the device manager to determine the COM# associated with the device. If prompted with the Found New Hardware Wizard, install the driver using the following steps (Microsoft Windows XP dialog boxes are shown, other Microsoft Windows operating systems are similar). If your operating system prompts you to search Windows Update, choose “No, not this time”. Then, in the Found New Hardware Wizard, choose "Install from a list or specific location" and click Next. (See Figure 31.)

Figure 31. Found New Hardware Wizard

Select "Search for the best driver in these locations" and check "Include this location in the search". Use the Browse button to navigate to the “Utilities\SAM-BA\XP driver” directory of the R9 Development installation and click “Next”.

The driver should be installed, and will come in as "AT91 USB to Serial Converter." Click Finish to complete. (See Figure 32.)

Figure 32. Driver Installed

Determine COM port assignment using Device Manager > Ports. The USB function port should be listed. For Windows 7, it may be listed as a GPS camera, otherwise it should be "AT91 USB to Serial Converter." Take note of the COM port assignment, to modify the programming batch file used to program the new OS Runtime image. (See Figure 33.)

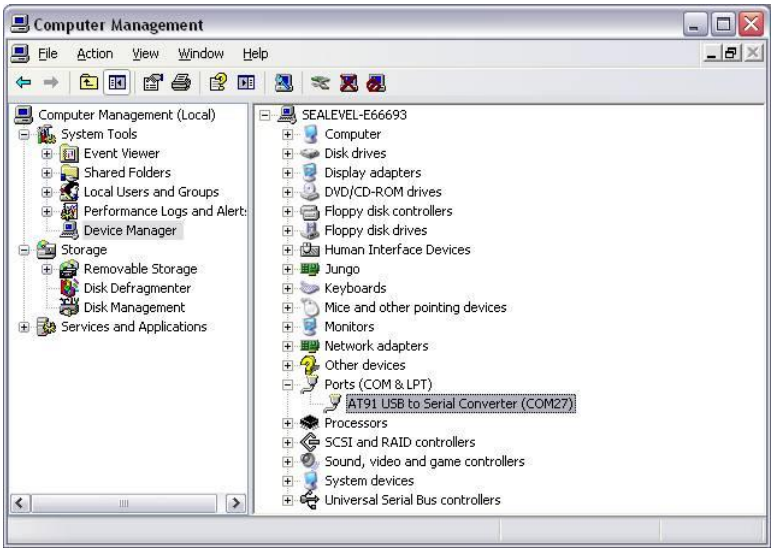


Figure 33. AT91 COM Port

Sample scripts have been provided in the R9 Development installation to automate the process of writing a complete OS runtime to the device. The script is configured to target a device attached to COM49 by default. This can be modified simply by editing the comport variable in the "NAND Program.bat" batch file. Once the batch file has been updated to reflect your system configuration, simply double-click the batch file

to begin the programming process. The process will take a few minutes. (See Figure 34).

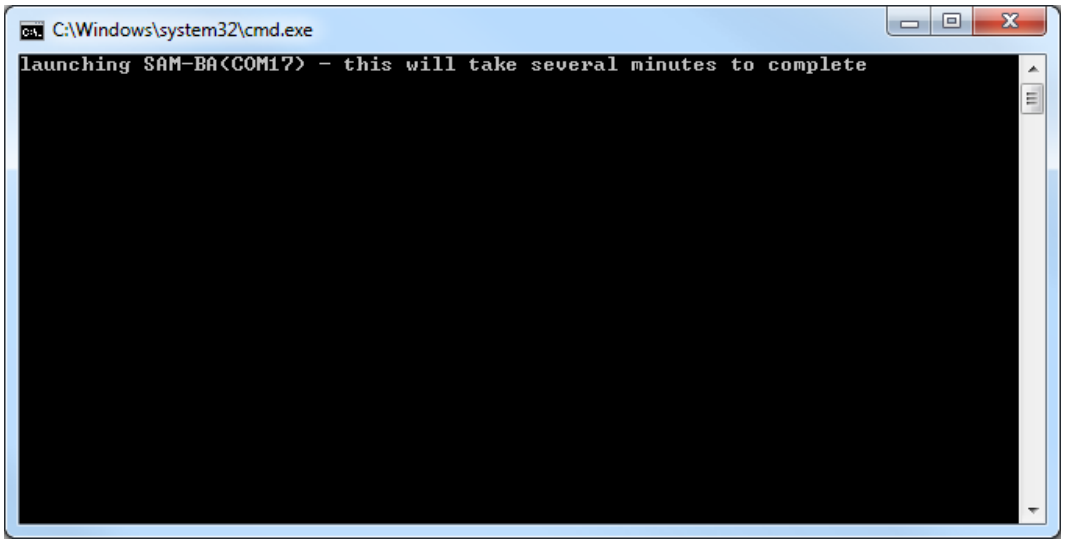


Figure 34. Programming NAND (COM17)

Once programming has completed, cycle device power and the OS runtime should boot. (See Figure 35.)

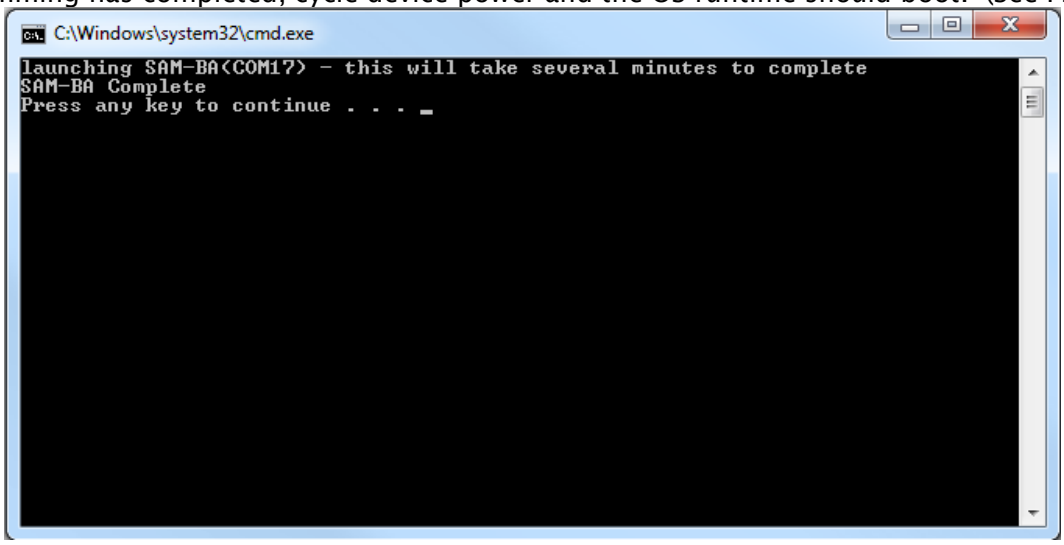


Figure 35. Programming Complete

As previously mentioned the process of programming the NAND Flash first erases all content from the NAND Flash. This includes the unique MAC address assigned to your device at the factory. The "finalize.exe" tool is provided in the "Boot Files" directory of the R9 Development installation. Finalize is a command line utility that accepts a MAC address in dashed notation (00-0A-0B-16-12-34). The application should be executed on the device – this can be accomplished with rapistart, telnet, or locally in the device's Command Prompt - after reprogramming the NAND Flash to reassign the MAC address. Once the application has been executed, the setting is applied upon device restart and persists.

Network Configuration

The Windows CE that runs on the SBC-R9-2100 is initially configured obtain its IP address via DHCP. Settings may be required for DNS or WINS server IP addresses or if you want to set up a static IP address. We have included an application in the OS that enables device configuration through a simple XML file format. The configuration is stored in a file that is kept up-to-date on the NAND Flash of the device. Likewise, edits to this file can be read as requests to modify the device's configuration. The configuration file can be accessed through ActiveSync using the USB device port connection or through an FTP client if you already know the IP address of the device. This section defines the XML configuration structure and corresponding values applicable for each element of the structure. Throughout this section the following definitions apply:

Term	Definition	Example
[int]	A number	123
[String]	Series of printable characters	This is a test string!234567609
[Multi-line String]	strings separated by \r\n	A\r\nNew\r\nMulti-liner
[Version]	A version number	1.2.3.4
[Boolean]	A binary state	True / False
[MACAddress]	A hardware identifier	00-0A-0B-16-11-1A
[IPAddress]	An IPv4 network address	192.168.0.100

The act of writing a new configuration file to the device will trigger a scan of that file (approximately every 5 seconds). If the file is invalid, it will be replaced with the current configuration. If a single element is invalid, that element and corresponding elements will be replaced with default values. To apply a new configuration, use the <Action> element with a value of "apply" as documented below.

```
<?xml version='1.0'?>
<Configuration>
    <System>
        <OS>WinCE</OS>
        <Version>6.0.0</Version>
        <Runtime>Proconex</Runtime>
        <RuntimeVersion>1.0.0.6</RuntimeVersion>
        <Processor>ATMEL, ARM926EJ-S-AT91SAM9263</Processor>
        <Name>WindowsCE</Name>
        <Description>WindowsCE Device</Description>
        <Owner></Owner>
        <Company></Company>
        <Address></Address>
        <Phone></Phone>
        <Extension></Extension>
    </System>
    <Ethernet>
        <Interface name="EMACB1">
            <DHCP>True</DHCP>
            <MAC>00-0A-0B-16-11-1A</MAC>
            <IPAddress>192.168.99.101</IPAddress>
            <Subnet>255.255.255.0</Subnet>
            <Gateway>192.168.99.1</Gateway>
            <Wifi enabled="True">
                <SSID>TESTNETWORK</SSID>
                <Mode>Infrastructure</Mode>
                <Channel>11</Channel>
                <Security>Wpa2Aes</Security>
                <Key encoding="Pass">********</Key>
            </Wifi>
        </Interface>
    </Ethernet>
</Configuration>
```

Sample configuration.xml read from device.

<Configuration> -Structure
The configuration element is the root XML element. This element must be present or the configuration file will not be considered valid. Invalid configurations will be replaced with a default configuration.

<System> -Structure
The system element contains all of the system information elements. This element must be present or the configuration file will not be considered valid.

<OS> - Readonly [string]
The OS element contains a string representation of the Operating System name. In the case of R9 products, this will be equivalent to "WinCE".

<Version> - Readonly [version]
The version element contains a dot-notation version string. This version is associated with the Operating System element.

<Runtime> - Readonly [string]
This element contains a string representation of the specific OS Runtime Image.

<RuntimeVersion> - Readonly [version]
This element contains a dot-notation version string. This version is associated with the OS Runtime Image.

<Processor> - Readonly [string]
This element contains a Processor Identification string.

<Name> - Read/Write [string]
This element may contain the device name string. This identifier is used as the WinCE host name.

<Description> - Read/Write [string]
This element may contain the device description string. This element can be used to further identify a device.

<Owner> - Read/Write [string]
This element may contain a string that can be used to identify a person or department responsible for maintaining a device.

<Company> - Read/Write [string]
This element may contain a string that can be used to identify the Company to which the device Owner is associated.
<Address> - Read/Write [multi-line string]
This element may contain a multi-line string (\r\n separated) to identify the location of the device Owner.

<Phone> - Read/Write [string]
This element may contain a string representation of a telephone contact number for the device Owner.

<Extension> - Read/Write [string]
This element may contain a string representation of a telephone extension for the device Owner.

<Ethernet> - Structure
The Ethernet element contains a list of Ethernet interfaces available to the device.

<Interface name=""> - Structure (Attribute Readonly [string])
The interface element is a container for the interface settings that are specific to the interface identifiable as "name". The name attribute is readonly and is used to uniquely distinguish Interface settings for the case where there are multiple Ethernet interfaces available.

<DHCP> - Read/Write [Boolean]
This element contains a Boolean value indicating whether DHCP Address resolution is enabled or disabled. Valid values are True or False.

<MAC> - Readonly [MACAddress]
This element contains a dash delimited string containing the unique MAC address of this interface. The first 3 octets identify the device as a Sealevel product (00-0A-0B). The fourth octet can be used to determine the product family (16). And the last two octets will be unique for each device (11-1A).

<IPAddress> - Read/Write [IPAddress]
This element may contain the current DHCP acquired IP Address or the current static IP address depending on the state of the DHCP element. Assigning a value to this element when DHCP is enabled has no effect.

<Subnet> - Read/Write [IPAddress]

This element may contain the current DHCP acquired Subnet Mask or the current static Subnet Mask depending on the state of the DHCP element. Assigning a value to this element when DHCP is enabled has no effect.

<*Gateway*> - Read/Write [IPAddress]
This element may contain the current DHCP acquired Gateway address or the current static Gateway address depending on the state of the DHCP element. Assigning a value to this element when DHCP is enabled has no effect.

<*Wifi enabled=""*> - Structure (Attribute Readonly)
The Wifi element is a container for wireless bridge settings if such a bridge is present. The "enabled" attribute will reflect whether the Interface is able to communicate with an approved wireless bridging module.

<*SSID*> - Read/Write [string]
This element contains the SSID string to be used when forming the wireless connection.

<*Mode*> - Read/Write [string: Adhoc, Infrastructure]
This element contains the overall Wireless configuration mode.

<*Channel*> - Read/Write [int: 1,11]
This element contains the wireless channel offset to use in Adhoc mode.

<*Security*> - Read/Write [string: None, WepOpen64, WepOpen128, WepShared64, WepShared128, WpaTkip, Wpa2Aes, Wpa2Tkip]
This element contains the security method for use in establishing the wireless connection.

<*Key encoding=""*> Writeonly [string] (Attribute [string: Hex, Ascii, Pass])
This key is used to set the wireless connection passphrase or value. Depending on the wireless configuration, the "encoding" attribute will need to be set accordingly. For security purposes this value cannot be read once it has been set.

<*Action*> - Writeonly [string]
This element may be used to trigger predetermined device behavior. For example, setting a value of "apply" to this element will result in the specified configuration being applied to the hardware and trigger a device restart so the settings will take effect.

Specifications

Dimensions

Length	Width	Height
4.9”	3.9”	0.75”

For CAD drawing with dimensions, see Appendix D – CAD Drawing.

Weight

0.24 lbs.

Power

Supply Line	7 – 30VDC Input
Rating	10 W Max (2.5W Nominal)

Connector: TB2
Manufacturer: Weco
Part Number: 121-M-211/02
Description: Locking Header, 2 pos, right angle, 5.08mm pitch
Mates with: Ria ASP0460202

Environmental Specifications

Specification	Operating	Storage
Temperature Range	-40° to 85° C	-60° to 150° C
Humidity Range	10 to 90% R.H. Non-Condensing	10 to 90% R.H. Non-Condensing

Manufacturing

All Sealevel printed circuit boards are built to UL 94V0 rating and are 100% electrically tested. These printed circuit boards are solder mask over bare copper or solder mask over tin nickel.

Appendix A – Resources

Books

Professional Microsoft Windows Embedded CE 6.0, Wrox, Phung.

Programming Windows Embedded CE 6.0 Developer Reference, Microsoft Press, Boling.
http://msdn.microsoft.com/en-us/library/cc526055.aspx

Web Sites

Atmel SAM-BA In-System Programmer (ISP)
http://www.atmel.com/dyn/products/tools_card.asp?tool_id=3883

FileZilla Open-Source FTP Client
http://www.filezilla-project.org

Microsoft Windows Embedded Home Page
www.microsoft.com/windows/embedded/default.mspx

Microsoft Windows Embedded CE Operating System Components
http://www.microsoft.com/windowsembedded/en-us/products/windowsce/component-library.mspx#type

Microsoft Windows Embedded CE 6.0 Evaluation Edition
http://www.microsoft.com/downloads/details.aspx?familyid=7E286847-6E06-4A0C-8CAC-CA7D4C09CB56&displaylang=en

Microsoft Windows Embedded CE 6.0 Online Documentation
http://msdn.microsoft.com/en-us/library/aa924073.aspx

Microsoft ActiveSync Download
http://www.microsoft.com/download/en/details.aspx?id=15

Microsoft Mobile Device Center 6.1
http://www.microsoft.com/windowsmobile/en-us/downloads/microsoft/device-center-download.mspx

Microsoft .NET Compact Framework
http://msdn.microsoft.com/en-us/netframework/aa497273.aspx

PuTTy Telnet/SSH Client Application
http://en.wikipedia.org/wiki/PuTTY
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Appendix B – SBC-R9-2100 Connector Reference

The following table details the connectors, jumpers, and test points located on the SBC-R9-2100. The connectors, jumpers, and test points are labeled by reference designator on the board silkscreen.

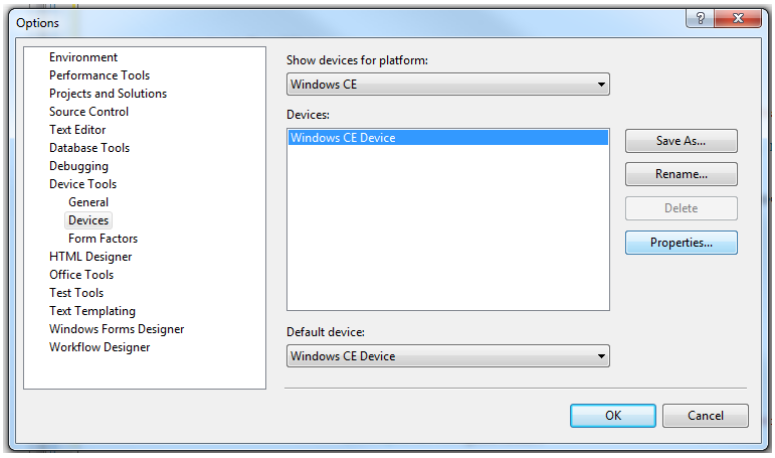
Reference Designator	Signal Description
J1	USB 2.0 device port (Type Mini-B)
J2	10/100 BaseT Ethernet interface
J3	RS-232 serial debug port
J4	Audio port
J5	RS-485 expansion port
J18	USB 2.0 host port (Molex connector)
J12	CAN 2.0b Bus interface
P2	RS-232, RS-422, RS-485 serial port
P3	RS-232, RS-422, RS-485 serial port
TB1	(4) Dry contact inputs/Optically isolated inputs/Wet outputs/Digital outputs
TB2	7-30VDC input power
Jumpers	
J6	Jumper - NAND Flash write protect
J7	Jumper - NAND Flash enable (Installed by default)
J14	Jumper - CAN Bus Termination (Installed by default)
Test Points	
TP1	Test Point - Ground

Appendix C – Application Debugging over Ethernet

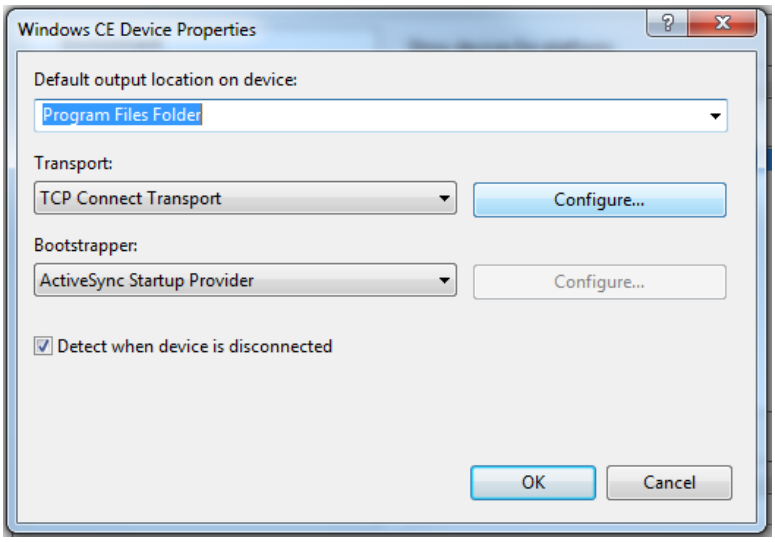
Applications can be debugged over an Ethernet connection in place of USB by configuring Visual Studio to directly connect to your device. For this method to work properly the Ethernet connection to the device must be properly configured to allow normal TCP/IP communications and you must know the IP address of the device you wish to execute the application on. For further information about configuring the Ethernet of the device see the Network Configuration section.

To configure Visual Studio to use your device for debugging over Ethernet, click the "Device Options" button on the Device toolbar. See below.

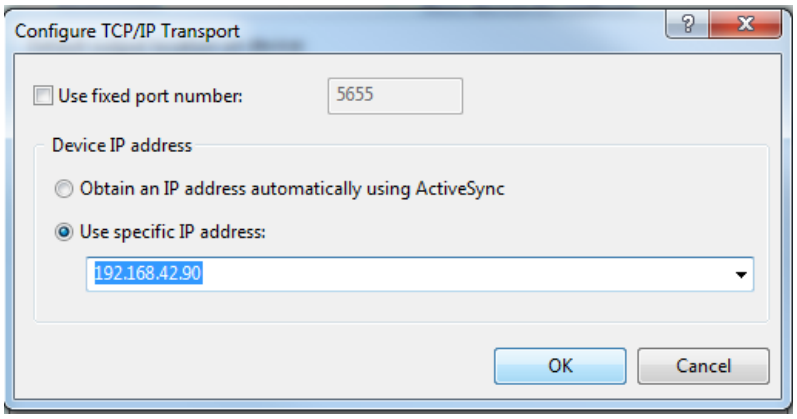
On the "Device Options" dialog, select the "Windows CE" platform and click the "Properties…" button. See below.



On the "Windows CE Device" properties dialog click the "Configure…" button. See below.



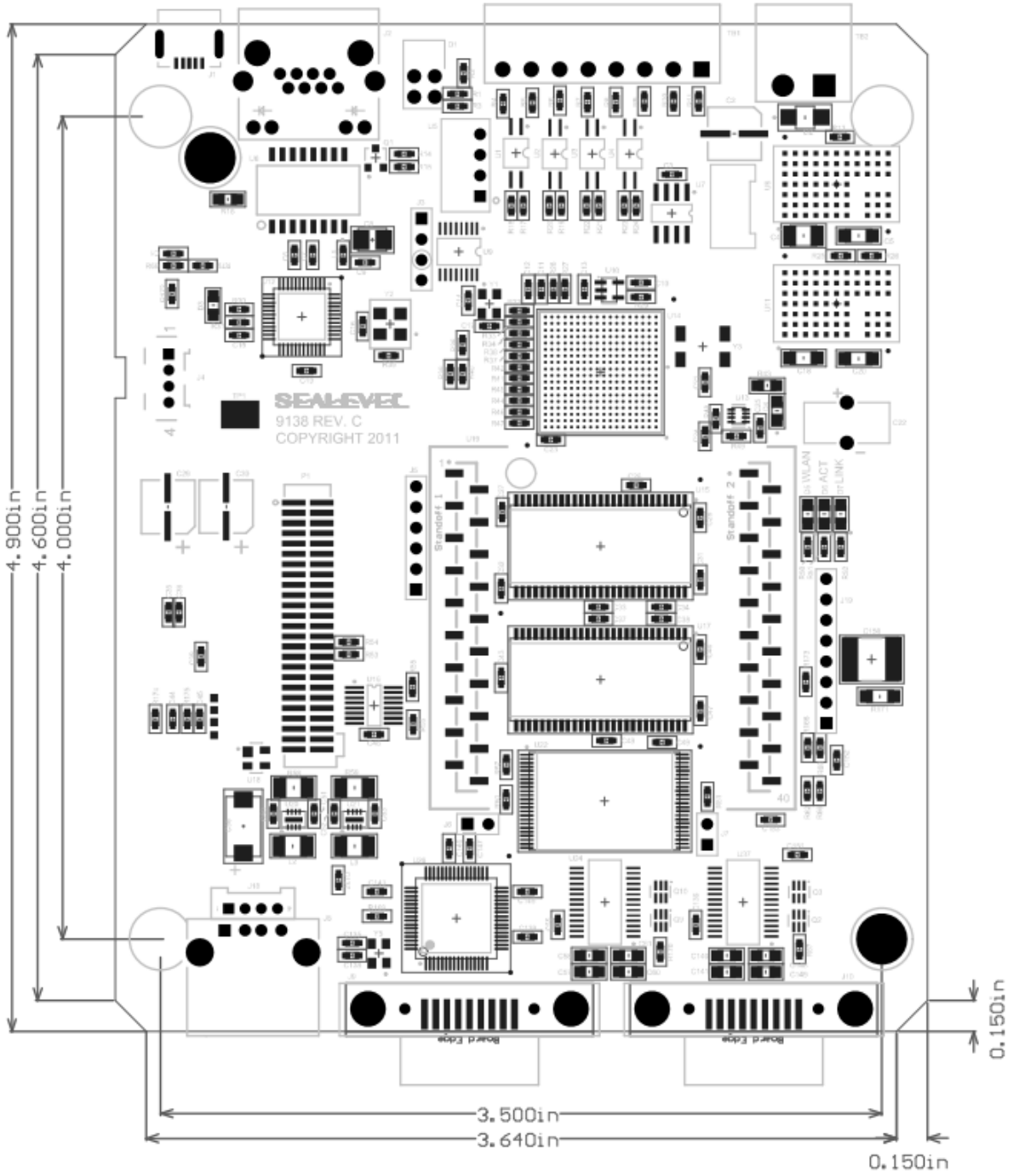
Now click the “Use specific IP address” radio button and type the IP address of the device in the text box. See below.



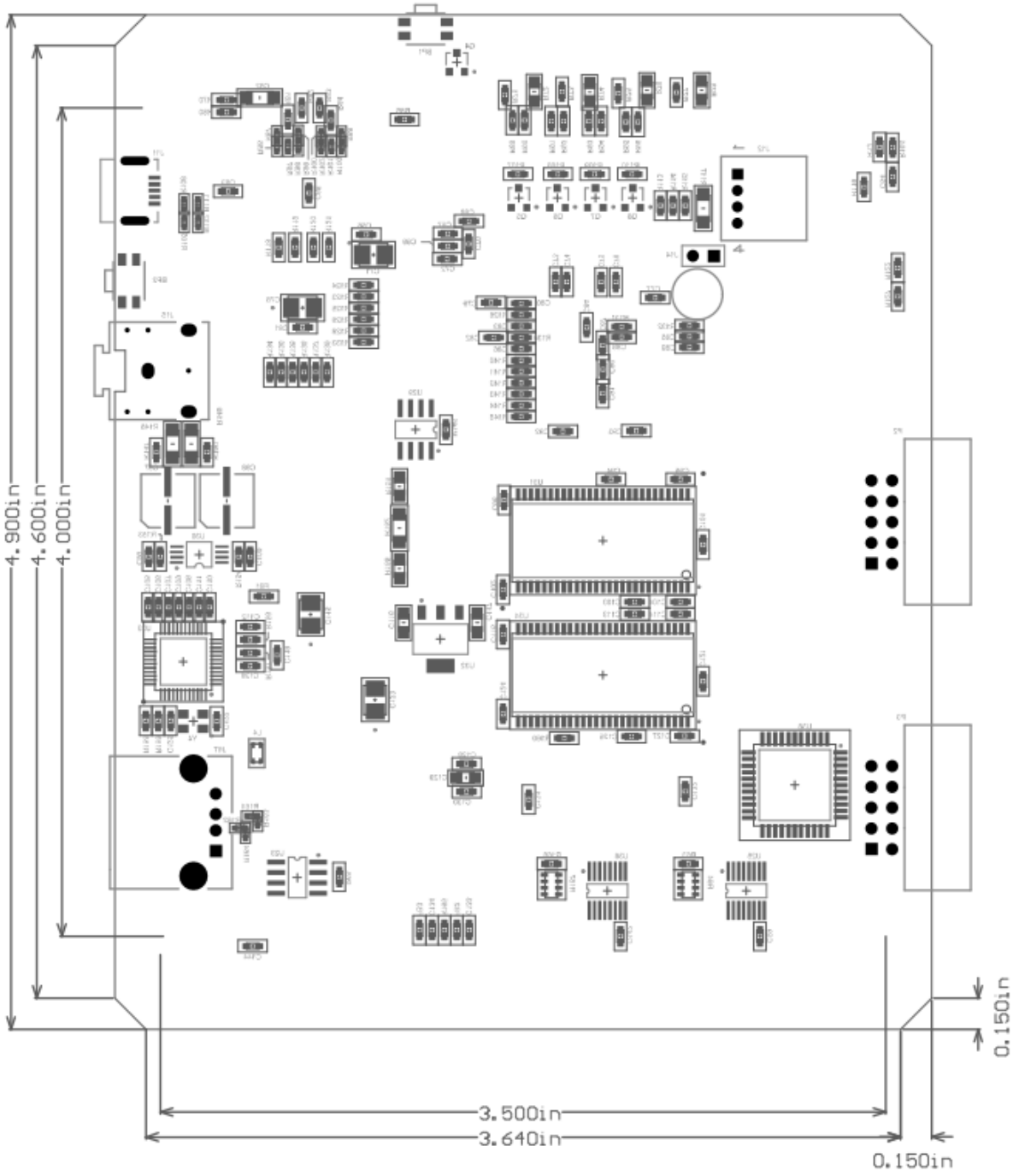
Click the “OK” button on all of the dialog windows and you should now be able to connect to the device through Ethernet for debugging. The application debugging guide can be continued as normal.

Appendix D – CAD Drawing

Top Side



Bottom Side



Appendix E – How to Get Assistance

When calling for technical assistance, please have the device installed and ready to run diagnostics. If possible, have your user manual and current settings ready.

The Sealevel website is an excellent resource located at www.sealevel.com. The most current software updates and user manuals are available via our homepage by clicking on the 'Drivers' or 'Manuals' links located under 'Technical Support.' Manuals and software can also be downloaded from the product page for your device.

The FAQ section of our website answers many common questions. Refer to this helpful resource by visiting www.sealevel.com/faq.asp.

Technical Support

Monday – Friday
8:00 am to 5:00 pm EST
Phone: +1 (864) 843-4343
Email: support@sealevel.com

RETURN AUTHORIZATION MUST BE OBTAINED FROM SEALEVEL SYSTEMS BEFORE RETURNED MERCHANDISE WILL BE ACCEPTED. AUTHORIZATION CAN BE OBTAINED BY CALLING SEALEVEL AND REQUESTING A RETURN MERCHANDISE AUTHORIZATION (RMA) NUMBER.

Warranty

Sealevel's commitment to providing the best I/O solutions is reflected in the Lifetime Warranty that is standard on all Sealevel manufactured I/O products. Relio™ industrial computers are warranted for a period of two years and the Relio™/SeaPAC™/SBC R9 family is warranted for a five year period from date of purchase. We are able to offer this warranty due to our control of manufacturing quality and the historically high reliability of our products in the field. Sealevel products are designed and manufactured at its Liberty, South Carolina facility, allowing direct control over product development, production, burn-in and testing. Sealevel achieved ISO-9001:2000 certification in 2002.

Warranty Policy

Sealevel Systems, Inc. (hereafter "Sealevel") warrants that the Product shall conform to and perform in accordance with published technical specifications and shall be free of defects in materials and workmanship for the warranty period. In the event of failure, Sealevel will repair or replace the product at Sealevel's sole discretion. Failures resulting from misapplication or misuse of the Product, failure to adhere to any specifications or instructions, or failure resulting from neglect, abuse, accidents, or acts of nature are not covered under this warranty.

Warranty service may be obtained by delivering the Product to Sealevel and providing proof of purchase. Customer agrees to insure the Product or assume the risk of loss or damage in transit, to prepay shipping charges to Sealevel, and to use the original shipping container or equivalent. Warranty is valid only for original purchaser and is not transferable.

This warranty applies to Sealevel manufactured Product. Product purchased through Sealevel but manufactured by a third party will retain the original manufacturer's warranty.

Non-Warranty Repair/Retest

Products returned due to damage or misuse and Products retested with no problem found are subject to repair/retest charges. A purchase order or credit card number and authorization must be provided in order to obtain an RMA (Return Merchandise Authorization) number prior to returning Product.

How to obtain an RMA (Return Merchandise Authorization)

If you need to return a product for warranty or non-warranty repair, you must first obtain an RMA number. Please contact Sealevel Systems, Inc. Technical Support for assistance:

Available	Monday – Friday, 8:00AM to 5:00PM EST
Phone	864-843-4343
Email	support@sealevel.com

Trademarks

Sealevel Systems, Incorporated acknowledges that all trademarks referenced in this manual are the service mark, trademark, or registered trademark of the respective company.