SBC-R9 ARM9 RISC Single Board Computer

User Manual | Relio R91001-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 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 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 ARM® Processor
- Up to 256MB SDRAM and 256MB NAND Flash Memory
- Dual SD/MMC Expansion Card Slots
- LCD and Backlight Controller
- Resistive Touchscreen Controller
- 10/100 BaseT Ethernet
- Two USB 2.0 Ports; USB Device Port
- CAN Bus Interface
- On-board Serial, Digital, and Analog I/O
- Compatible with Windows Embedded CE 6.0 and Linux
- Low Power Requirements
- Power and Status LED Indicators

Before You Get Started

What's Included

The SBC-R9 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**
- **SD Card with CE runtime image, Talos .NET Framework, application samples, and documentation**
- **Microsoft® Windows® CE 6.0 Core license**

Advisory Conventions

W **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.

i **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.

N **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.

Quickstart Kit

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

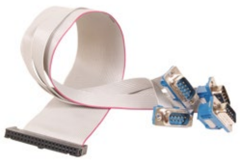
The SBC-R9-KT includes the following items:

- SBC-R9 ARM9 Embedded RISC Single Board Computer
- SD Card with CE runtime image, Talos .NET Framework, application samples, and documentation
- Microsoft Windows CE 6.0 Core License
- TR134 – 100-240VAC to 12VDC @ 2.5A, wall mount power supply
- CA179 – USB Type A to USB Type B, device cable
- CA429 – R9 serial debug cable
- CA246 – CAT5 patch cable, 6' length
- CA273 – 40-pin IDC ribbon cable to (4) DB9M connectors

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 (864) 843-4343.

Cables

USB Type A to USB Type B, 72" in Length - Device Cable (Item# CA179)	
The CA179 is a 72" standard USB device cable that connects USB peripherals with a 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).	
40-Pin IDC Ribbon Cable to (4) DB9 Male Connectors, 14" in Length (Item# CA273)	
40-Pin IDC Ribbon Cable terminates to (4) DB9 Male Connectors, 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.	

Power Supply

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.	

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 (64 MB Standard)

256MB NAND Flash

Two SD Memory Card Sockets

LCD CONTROLLER

Supports Passive or Active Displays

16-bit Color in TFT/STN Modes

Resolution Up to 2048 x 2048

TOUCHSCREEN CONTROLLER

Supports 5-wire Resistive Touchscreens

BUS INTERFACES

10/100 BaseT Ethernet

USB Device Port

Two USB 2.0 Ports

CAN Bus

Dedicated RS-485 Expansion

INDUSTRIAL I/O

Four Software Configurable RS-232/422/485 Ports

Eight Optically Isolated Inputs (5 – 24V)

Eight Open-Collector Outputs (5 – 30V; 3 with PWM)

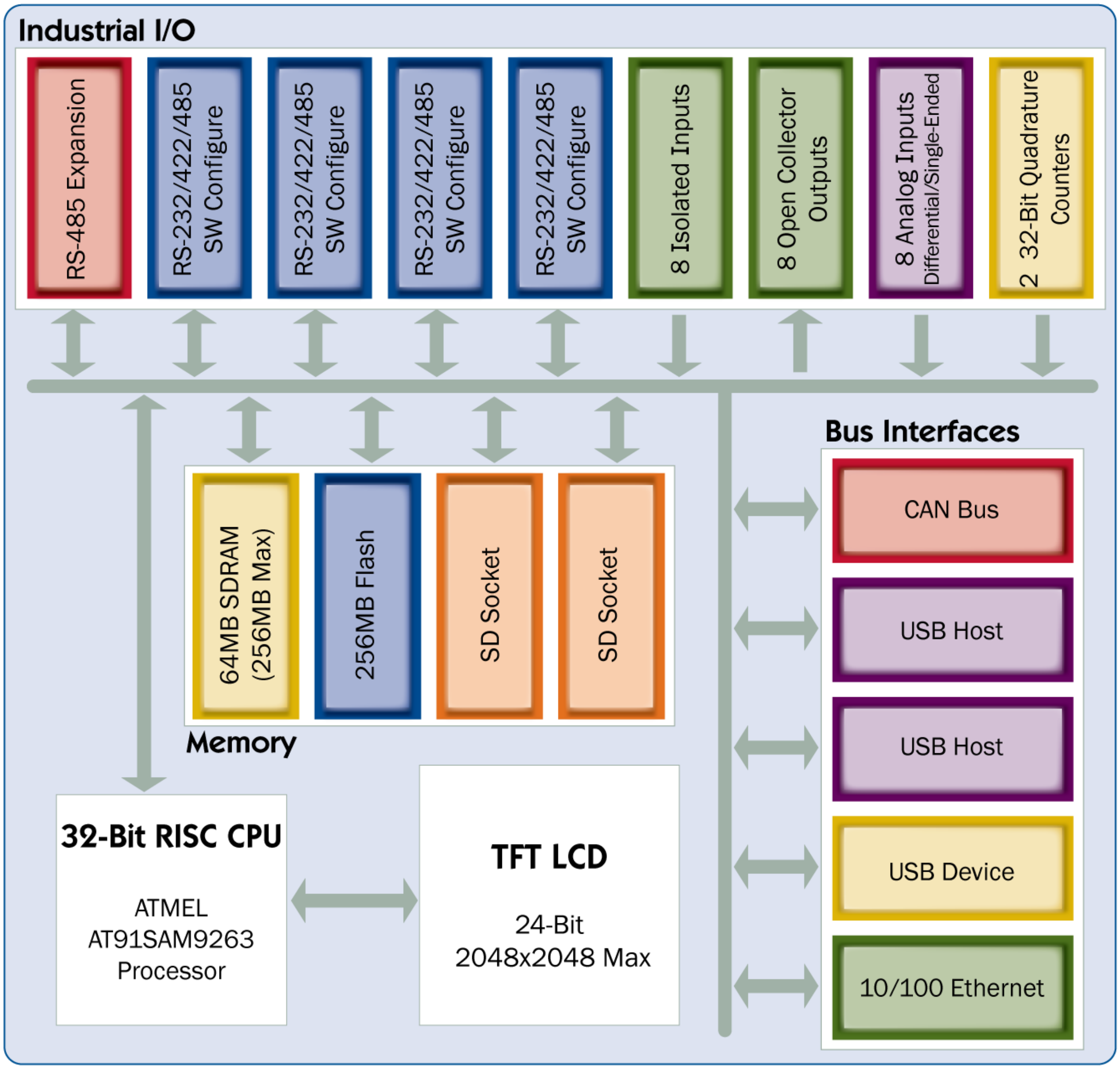
Eight Analog Inputs (12-bit or 16-bit)

Two 32-bit Quadrature Counters

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.

Technical Description

Memory

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

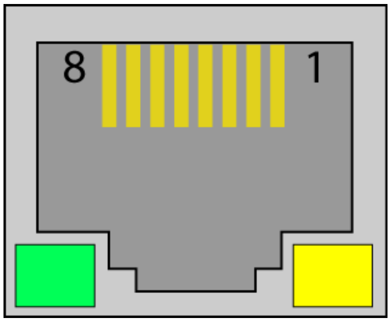
Ethernet

The SBC-R9 includes a 10/100 BaseT Ethernet interface accessed via the RJ45 connector located at (J14).

W

The RJ45 port on the left side of the SBC-R9 is a RS-485 Expansion Port (labeled J5) and is NOT an Ethernet port. Damage to Ethernet networking equipment can result if connected to the RS-485 RJ45 connector.

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

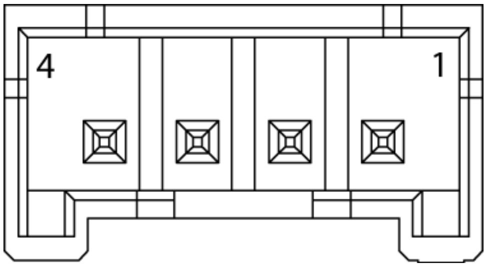


USB

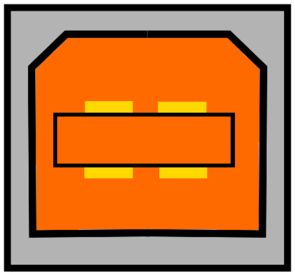
The SBC-R9 provides two USB 2.0 host ports, and one device port. The host USB ports are located at (J7) and (J8). The device USB port is located at (J13).

Connector:	J7, J8
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



Connector:	J13
Manufacturer:	Samtec
Description:	High Retention USB Type B



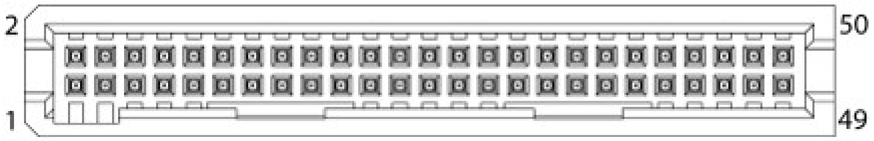
LCD and Touchscreen Controllers

A variety of LCDs can be directly controlled by the SBC-R9's on-board LCD controller. All LCD power and control signals are available on header connector P2.

Connector:	P2
Manufacturer:	Samtec
Part Number:	TFML-125-02-S-D
Description:	Locking terminal strip, 50 pos, 0.050" pitch
Mates with:	Samtec SFML-125-T2-S-D or Samtec TFMDL-25-T-03.00

Position	Signal
1	GND
2	DCLK
3	HSYNC
4	VSYNC
5	GND
6	R0
7	R1
8	R2
9	R3
10	R4
11	R5
12	GND
13	G0
14	G1
15	G2
16	G3
17	G4
18	G5
19	GND
20	B0
21	B1
22	B2
23	B3
24	B4
25	B5

Position	Signal
26	GND
27	LCDEN
28	3.3V
29	3.3V
30	HDMODE
31	VDMODE
32	NC
33	NC
34	NC
35	NC
36	Touch UL
37	Touch LL
38	Touch UR
39	Touch LR
40	Touch Wiper
41	NC
42	NC
43	NC
44	NC
45	LCDLED3
46	GND
47	LCDLED2
48	GND
49	LCDLED1
50	GND

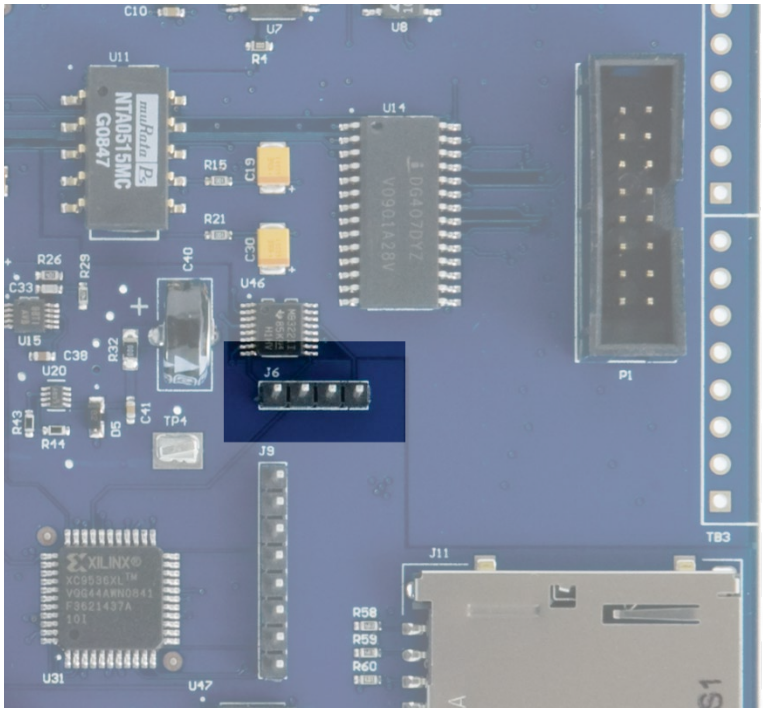


Serial Debugging

Debug the R9 through the RS-232 debug port.

Connector:	J6
Manufacturer:	Amp/Tyco
Part Number:	9-146278-0-04
Description:	Header, 0.100” Polarized 4 pos, pin 3 Removed

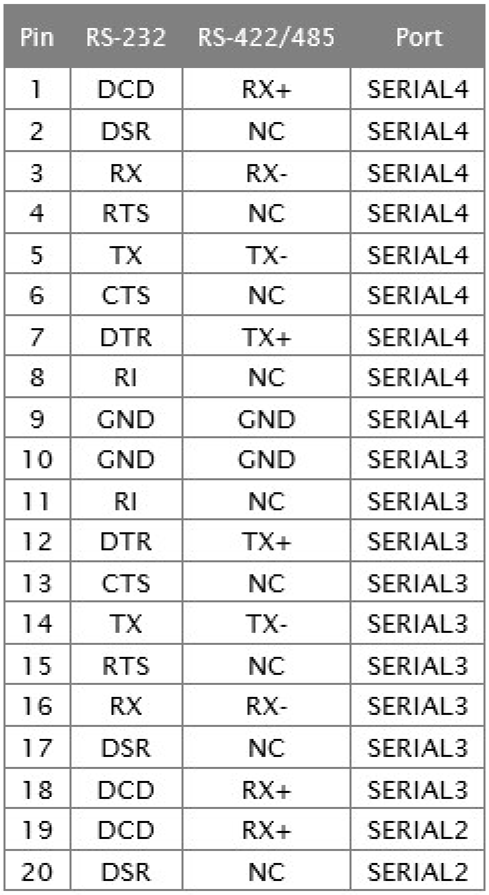
Pin	RS-232
1	RX
2	GND
3	Key
4	TX



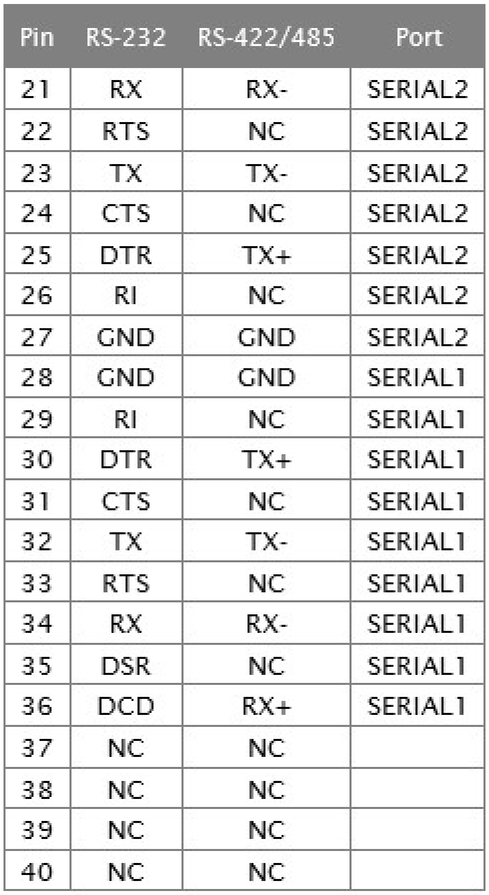
Serial Communications

Connect to a variety of serial peripherals via the SBC-R9’s software configurable RS-232/422/485 ports. Each port is located on connector (P4).

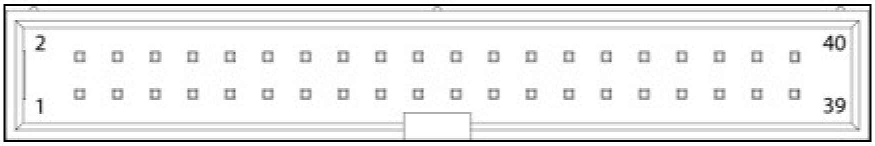
Connector:	P4
Manufacturer:	Sullins
Part Number:	SBH11-PBPC-D20-ST-BK
Description:	Box Header, 0.100” Polarized 40 pos (2x20)
Mates with:	SFH213-PPPC-D20-ID-BK-M181 or equivalent



Pin	RS-232	RS-422/485	Port
1	DCD	RX+	SERIAL4
2	DSR	NC	SERIAL4
3	RX	RX-	SERIAL4
4	RTS	NC	SERIAL4
5	TX	TX-	SERIAL4
6	CTS	NC	SERIAL4
7	DTR	TX+	SERIAL4
8	RI	NC	SERIAL4
9	GND	GND	SERIAL4
10	GND	GND	SERIAL3
11	RI	NC	SERIAL3
12	DTR	TX+	SERIAL3
13	CTS	NC	SERIAL3
14	TX	TX-	SERIAL3
15	RTS	NC	SERIAL3
16	RX	RX-	SERIAL3
17	DSR	NC	SERIAL3
18	DCD	RX+	SERIAL3
19	DCD	RX+	SERIAL2
20	DSR	NC	SERIAL2



Pin	RS-232	RS-422/485	Port
21	RX	RX-	SERIAL2
22	RTS	NC	SERIAL2
23	TX	TX-	SERIAL2
24	CTS	NC	SERIAL2
25	DTR	TX+	SERIAL2
26	RI	NC	SERIAL2
27	GND	GND	SERIAL2
28	GND	GND	SERIAL1
29	RI	NC	SERIAL1
30	DTR	TX+	SERIAL1
31	CTS	NC	SERIAL1
32	TX	TX-	SERIAL1
33	RTS	NC	SERIAL1
34	RX	RX-	SERIAL1
35	DSR	NC	SERIAL1
36	DCD	RX+	SERIAL1
37	NC	NC	
38	NC	NC	
39	NC	NC	
40	NC	NC	

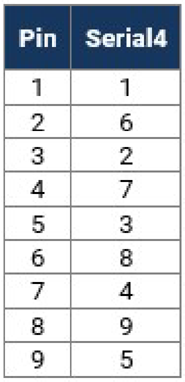


COM Port Assignments

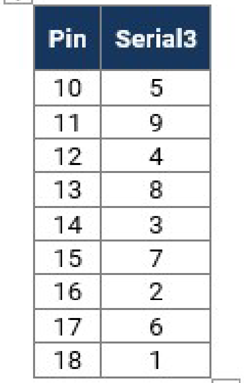
Serial Port	Assignment
RS232 Debug Port	COM0
RS485 Expansion Port	COM1
SERIAL1	COM2
SERIAL2	COM3
SERIAL3	COM4
SERIAL4	COM5

CA273 Accessory Cable

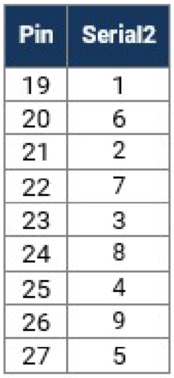
The DB9 pin out is achieved using the CA273 accessory cable. The 40-pin connector is in the first column and corresponding DB9 connectors are in the second column.



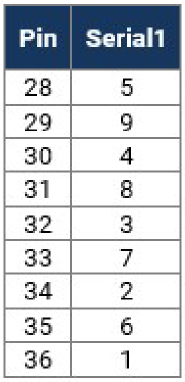
Pin	Serial4
1	1
2	6
3	2
4	7
5	3
6	8
7	4
8	9
9	5



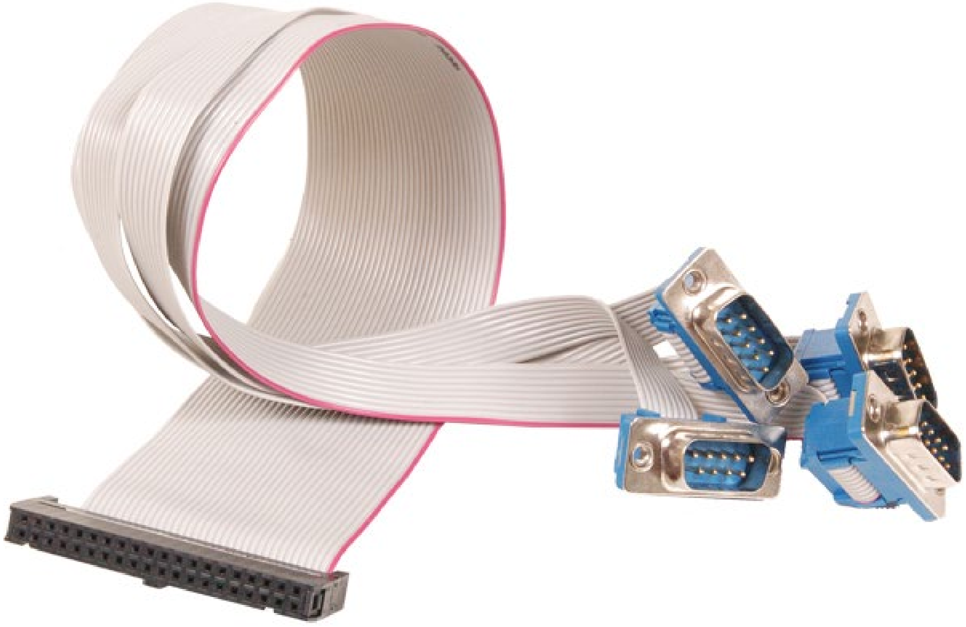
Pin	Serial3
10	5
11	9
12	4
13	8
14	3
15	7
16	2
17	6
18	1



Pin	Serial2
19	1
20	6
21	2
22	7
23	3
24	8
25	4
26	9
27	5



Pin	Serial1
28	5
29	9
30	4
31	8
32	3
33	7
34	2
35	6
36	1

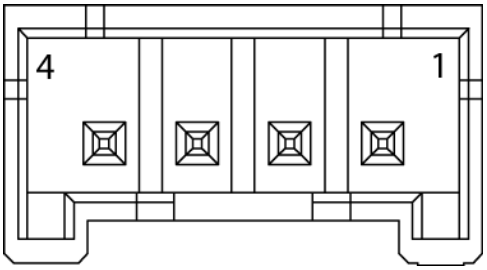


CAN Bus

Connect directly to a Control Area Network (CAN) bus via connector (J3). A Molex 4-pin vertical 2mm locking header is used for the connection.

Connector:	J3
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

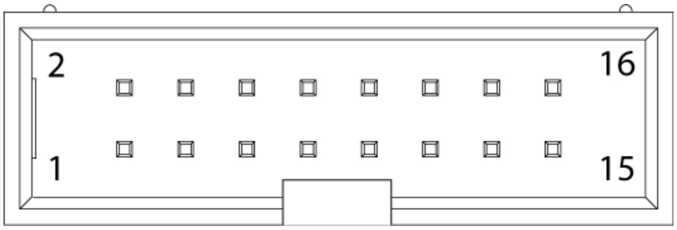


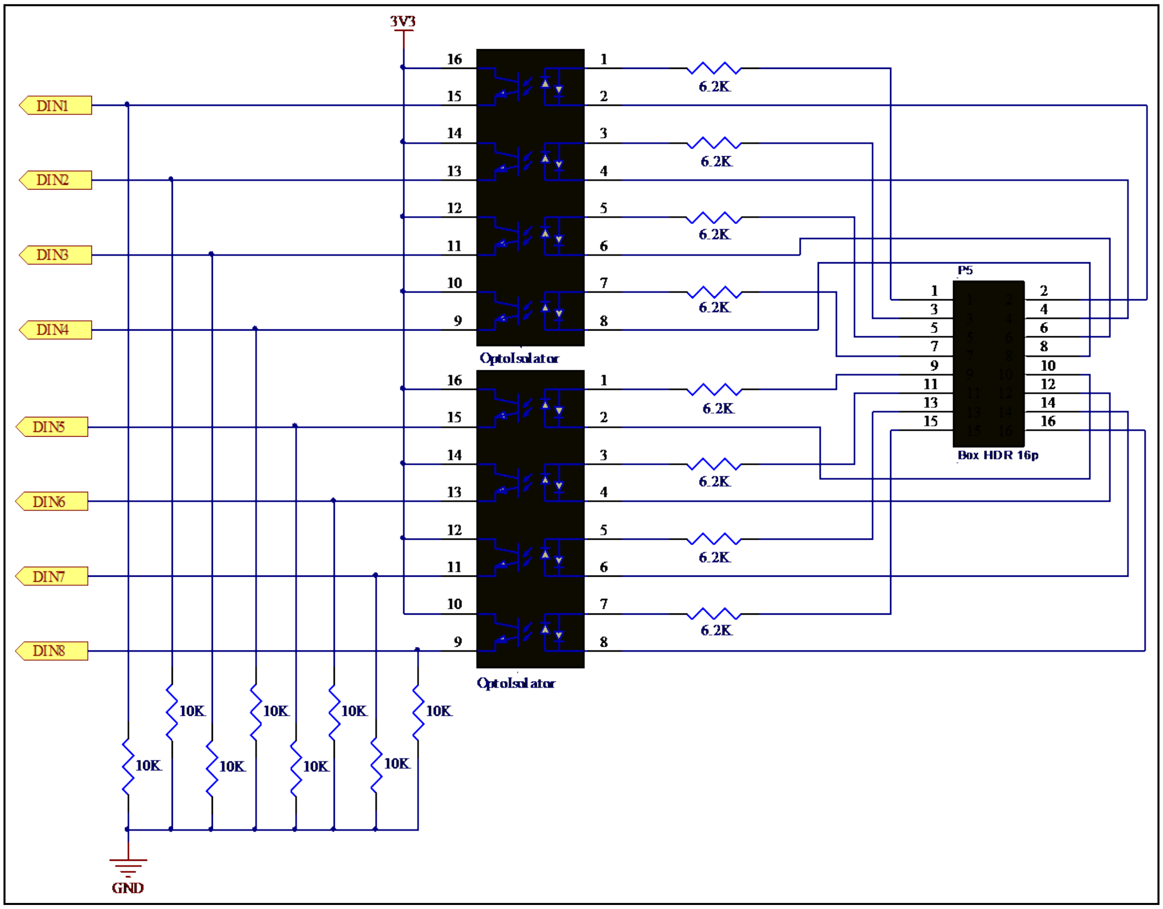
Optically Isolated Inputs

Directly monitor 8 optically isolated inputs, which are found on connector (P5). The non-polarized inputs can range from 5-24VDC and provide 300V external isolation.

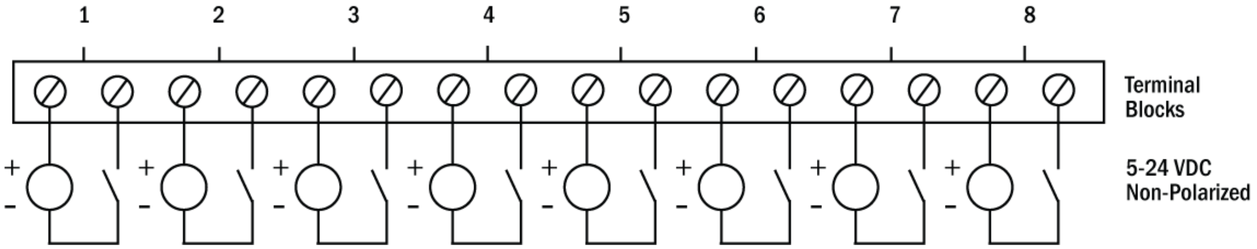
Connector:	P5
Manufacturer:	Sullins
Part Number:	SBH11-PBPC-D08-ST-BK
Description:	Box Header, 0.100" Polarized 16 pos (2x8)
Mates with:	Sullins SFH213-PPPC-D08-ID-BK-M181 or equivalent

Pin	Signal
1	Input 1A
2	Input 1B
3	Input 2A
4	Input 2B
5	Input 3A
6	Input 3B
7	Input 4A
8	Input 4B
9	Input 5A
10	Input 5B
11	Input 6A
12	Input 6B
13	Input 7A
14	Input 7B
15	Input 8A
16	Input 8B





8 Optically Isolated Inputs

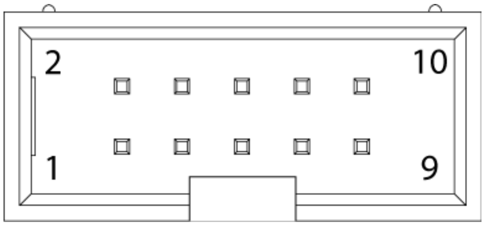


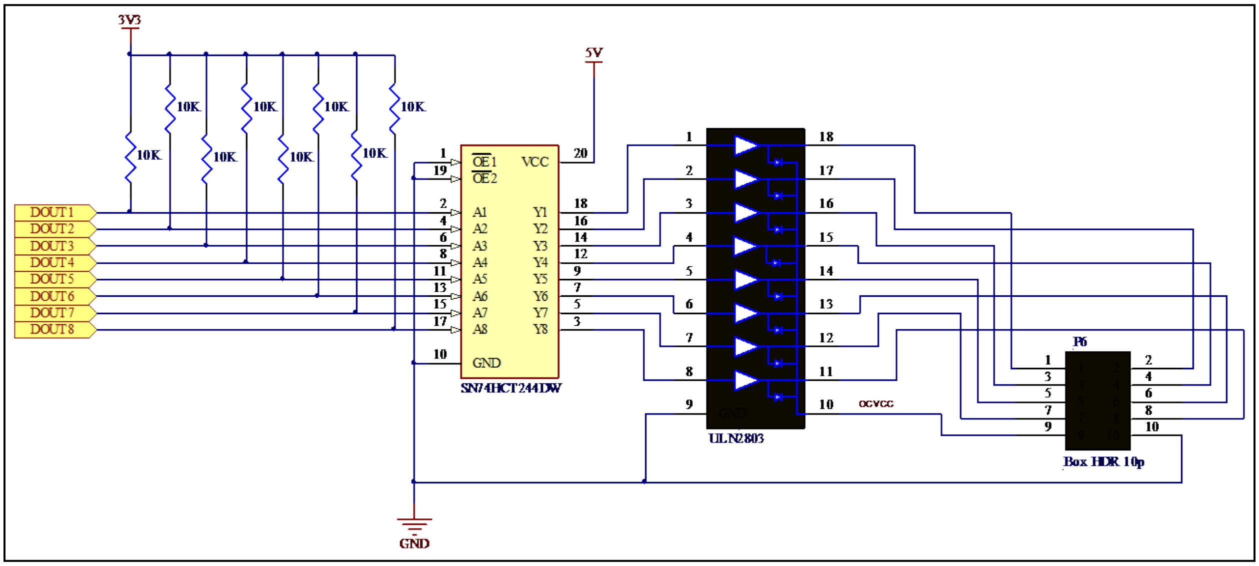
Open Collector Outputs

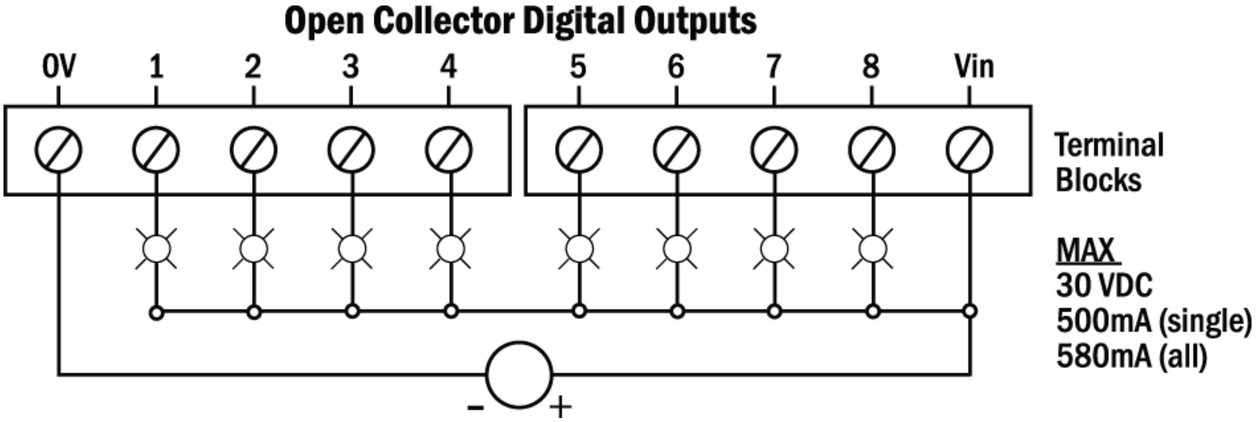
Directly control 8 outputs via the SBC-R9's open-collector outputs found on connector (P6). The open collector outputs have a range of 5 – 30V with a maximum sink current of 500mA on a single output with a combined maximum sink current of 580mA on all outputs.

Connector:	P6
Manufacturer:	Sullins
Part Number:	SBH11-PBPC-D05-ST-BK
Description:	Box Header, 0.100" Polarized 10 pos (2x5)
Mates with:	Sullins SFH213-PPPC-D05-ID-BK-M181 or equivalent

Pin	Signal
1	Output 1/ PWM1
2	Output 2 / PWM2
3	Output 3 / PWM3
4	Output 4
5	Output 5
6	Output 6
7	Output 7
8	Output 8
9	OCVCC
10	GND




3V3
10K
5V
DOUT1
DOUT2
DOUT3
DOUT4
DOUT5
DOUT6
DOUT7
DOUT8
OE1
OE2
VCC
A1
A2
A3
A4
A5
A6
A7
A8
Y1
Y2
Y3
Y4
Y5
Y6
Y7
Y8
GND
SN74HCT244DW
ULN2803
OCVCC
P6
Box HDR 10p
GND

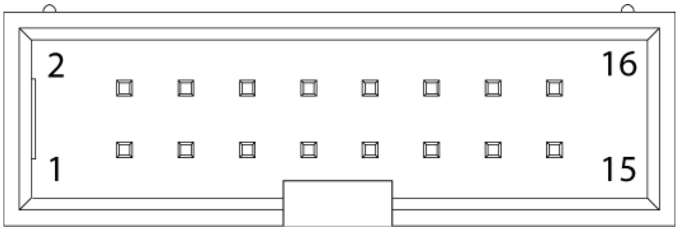

Open Collector Digital Outputs
0V
1
2
3
4
5
6
7
8
Vin
Terminal Blocks
MAX
30 VDC
500mA (single)
580mA (all)
-
+

Analog Inputs

The SBC-R9 base configuration includes a 12-bit ADC. Software programmable input ranges are 0V to 5V, 0V to 10V, ±5V or ±10V. Interface a variety of transducers and other analog signals via eight 12-bit analog inputs located on connector (P1). For applications requiring higher resolution, the board can be ordered preconfigured with a 16-bit A/D converter.

Connector:	P1
Manufacturer:	Sullins
Part Number:	SBH11-PBPC-D08-ST-BK
Description:	Box Header, 0.100” Polarized 16 pos (2x8)
Mates with:	Sullins SFH213-PPPC-D08-ID-BK-M181 or equivalent

Pin	Signal
1	AIN1+
2	AIN1-
3	AIN2+
4	AIN2-
5	AIN3+
6	AIN3-
7	AIN4+
8	AIN4-
9	AIN5+
10	AIN5-
11	AIN6+
12	AIN6-
13	AIN7+
14	AIN7-
15	AIN8+
16	AIN8-



Quadrature Counters

High-speed input monitoring is accomplished with minimal software overhead using the two onboard 32-bit quadrature counters. Both counters are available on a single connector (P7). Input levels are LVTTL (0 – 3.6VDC).

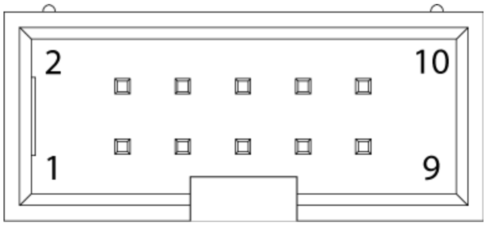
Connector:	P7
Manufacturer:	Sullins
Part Number:	SBH11-PBPC-D05-ST-BK
Description:	Box Header, 0.100” Polarized 10 pos (2x5)
Mates with:	Sullins SFH213-PPPC-D05-ID-BK-M181 or equivalent

QUADRATURE COUNTER 1

Pin	Signal
1	A
3	B
5	#INDEX
7	GND
9	3.3VDC

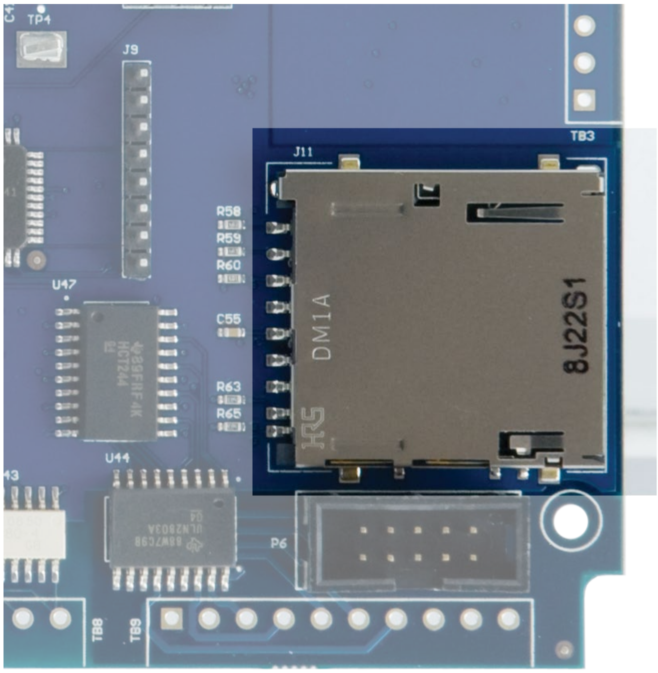
QUADRATURE COUNTER 2

Pin	Signal
2	A
4	B
6	#INDEX
8	GND
10	3.3VDC



SD/MMC Cards

The SBC-R9 provides two SD/MMC Card slots, Slot A (bottom of board) and Slot B (top of board), located on the right side of the board. Each slot will accept standard-capacity SD/MMC Cards up to 2GB. SD/MMC Card slot A may be used for booting.



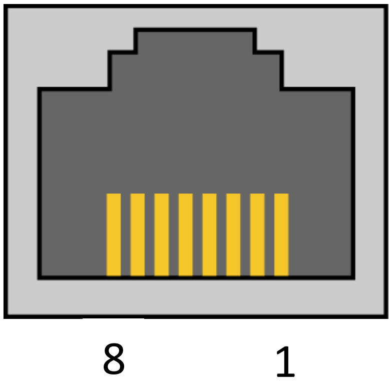
RS-485 Expansion

The SBC-R9 provides a RS-485 Expansion Port. The port is available via a RJ-45 connector (J5), as well as via a Molex 4-pin vertical 2mm locking header (J10). This offers two convenient options for adding additional expansion modules from the SealO product line.

W The RJ45 port (J5) is a RS-485 Expansion Port and is NOT an Ethernet port. Damage to Ethernet networking equipment can result if connected to the RS-485 RJ45 connector.

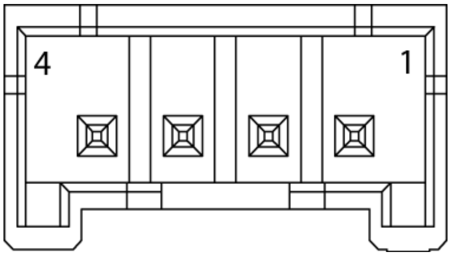
Connector:	J5
Manufacturer:	Xmultiple
Part Number:	XRJM-S-01-8-8-F2 or XRJM-S-01-8-8-0
Description:	RJ45 Socket, W/O LEDs, Shielded
Mates with:	Standard RJ45 Plug

Pin	Signal
1	9-30VDC Source
2	9-30VDC Source
3	Not connected
4	485+
5	485-
6	Not Connected
7	Common (GND)
8	Common (GND)



Connector:	J10
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	485-
2	485+
3	Common (GND)
4	Shield (GND)



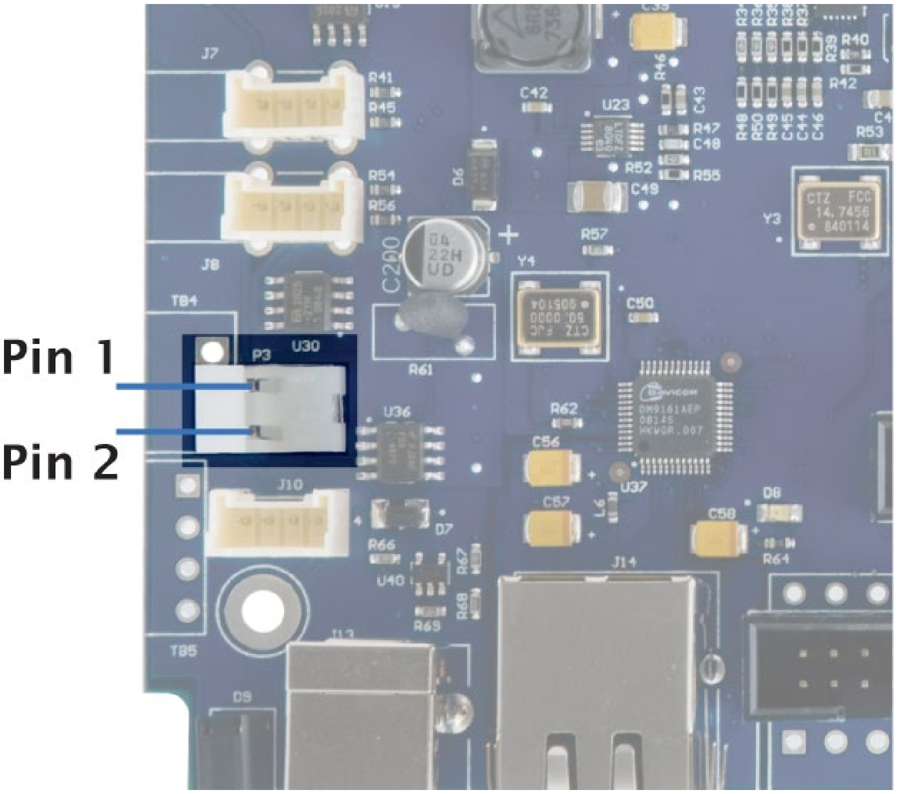
Power

The SBC-R9 can be powered with the Sealevel TR134.

Connector:	P3
Manufacturer:	Molex
Part Number:	09-65-2028
Description:	3.96mm Pitch Friction Lock Header
Mates with:	Molex 09-50-1021

W Be sure that you connect the power lead to the proper pin. Reversing the polarity of the power input will damage your SBC-R9.

Pin	Signal
1	9-30VDC
2	GND

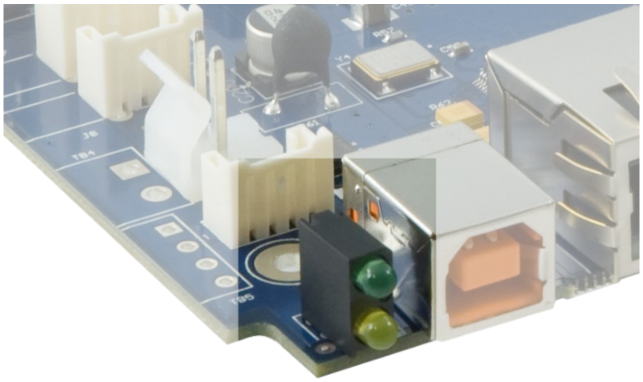


LED Indicators

The SBC-R9 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



Installation

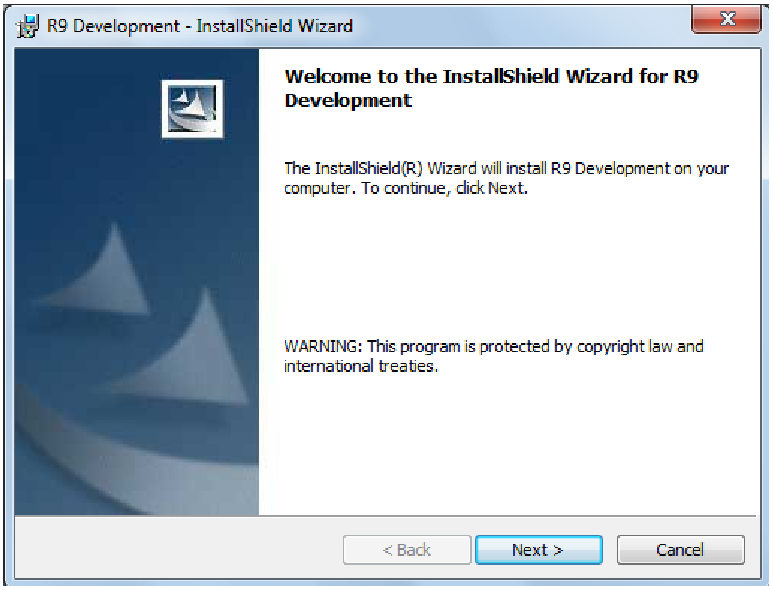
Software Installation

Windows Installation

Do not install the hardware until the software has been fully installed.

Only users running Windows 7 or newer should utilize these instructions for accessing and installing the appropriate driver via Sealevel's website. If you are utilizing an operating system prior to Windows 7, please contact Sealevel by calling 864.843.4343 or emailing support@sealevel.com to receive access to the proper driver download and installation instructions.

1. Begin by removing the contents from the box.
2. Download the correct software from the website - Software: R9 Development SDK - Windows - Sealevel. This will install the R9 Development SDK, which includes the Talos.NET Framework, documentation, and examples, on to your PC (See Figure 1.)"



R9 Development - InstallShield Wizard

Welcome to the InstallShield Wizard for R9 Development

The InstallShield(R) Wizard will install R9 Development on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

Figure 1 *- Installation Wizard*

3. The setup files will automatically detect the operating environment and install the proper components. Follow the information presented on the screens that follow.

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

5. 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.

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

W To avoid accidental damage be sure to observe proper power connector polarity. See Power Pin-out section.

Apply power to the SBC-R9 by connecting the TR134 Molex connector to the SBC-R9 (P3) connector, noting proper polarity. Attach the other end of the TR134 into a 120VAC wall outlet. *(See Figure 2)*

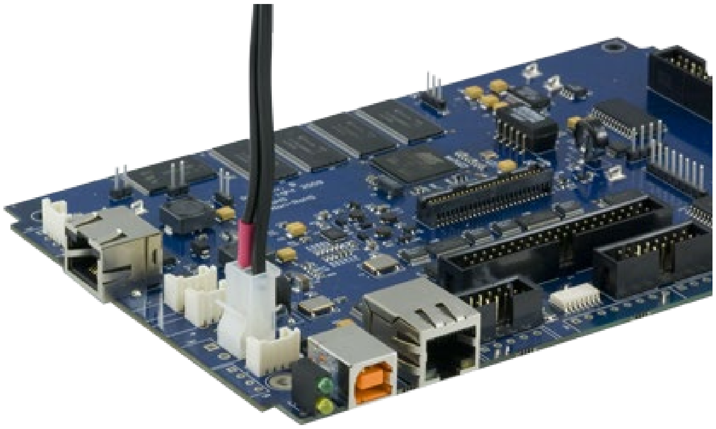


***Figure 2* – Connect the TR134 Molex connector to the SBC-R9 (P3) connector**

Use a standard USB device cable and connect the Type B connector to the SBC-R9. (See Figure 3.) Connect Type A connector into the host PC.

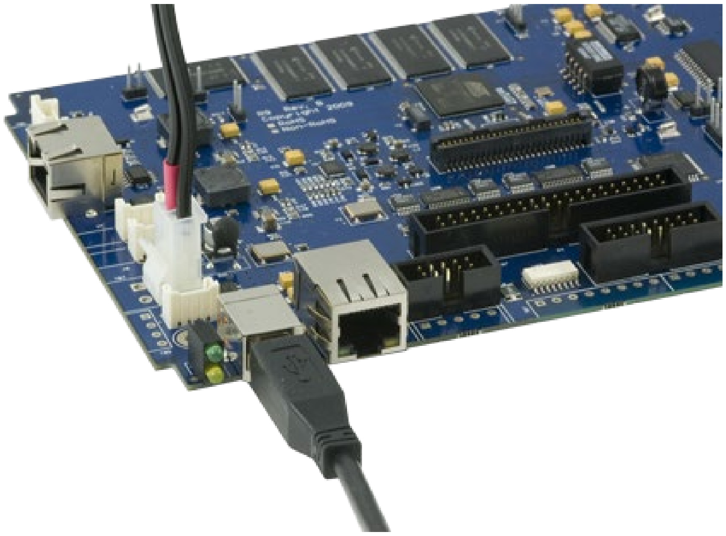


Figure 3 – Connect the Type B connector to the SBC-R9

You are now ready to set up a USB communication interface between the host PC and the SBC-R9 board. 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 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 board and the device center connection screen will appear. (See Figure 4.)

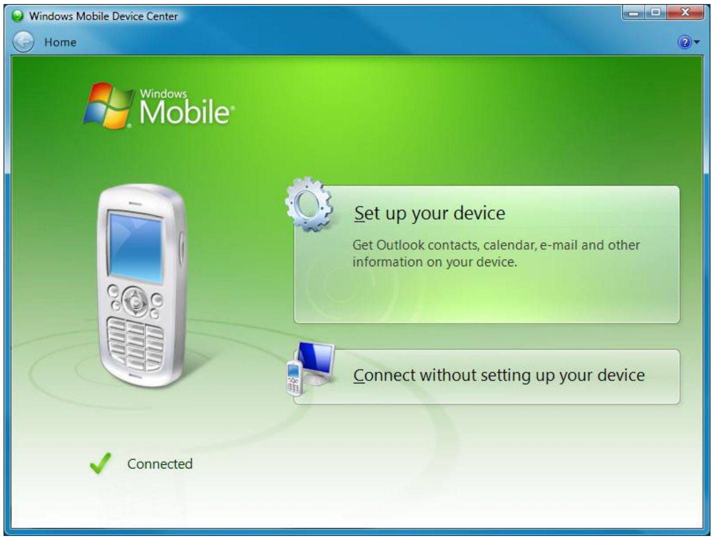

Windows Mobile Device Center
Home
Windows Mobile
Set up your device
Get Outlook contacts, calendar, e-mail and other information on your device.
Connect without setting up your device
Connected

***Figure 4**- Device Center connected screen*

Using your mouse, click “Connect without setting up your device”. The idea is to explore the file system on the SBC-R9 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 5.)

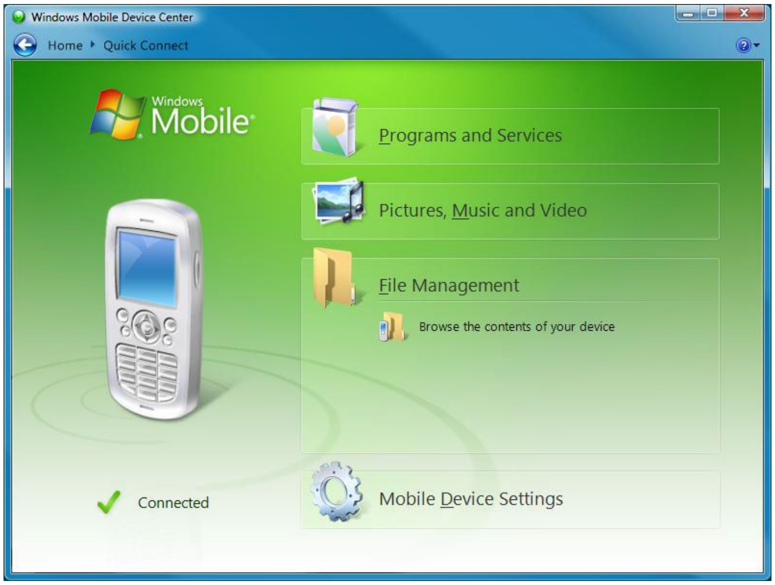


***Figure 5** - Device Center File Management*

This action opens a standard Windows Explorer where the default file contents of the SBC-R9 can be read or written to. (See Figure 6.)

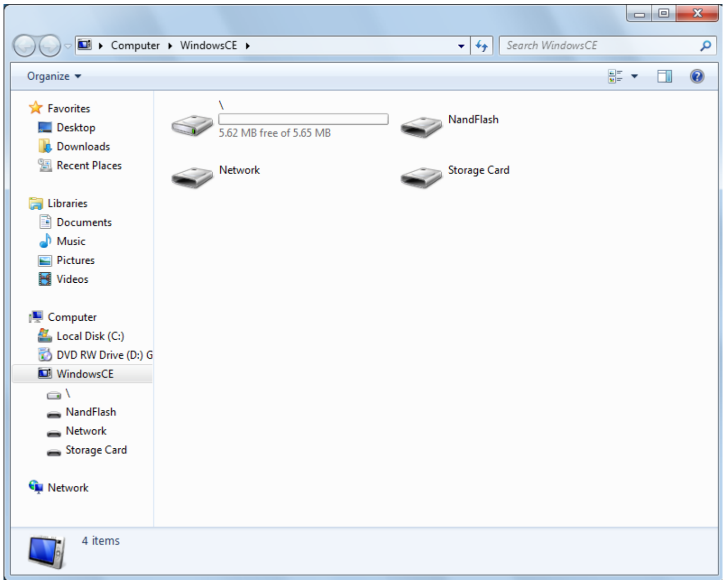
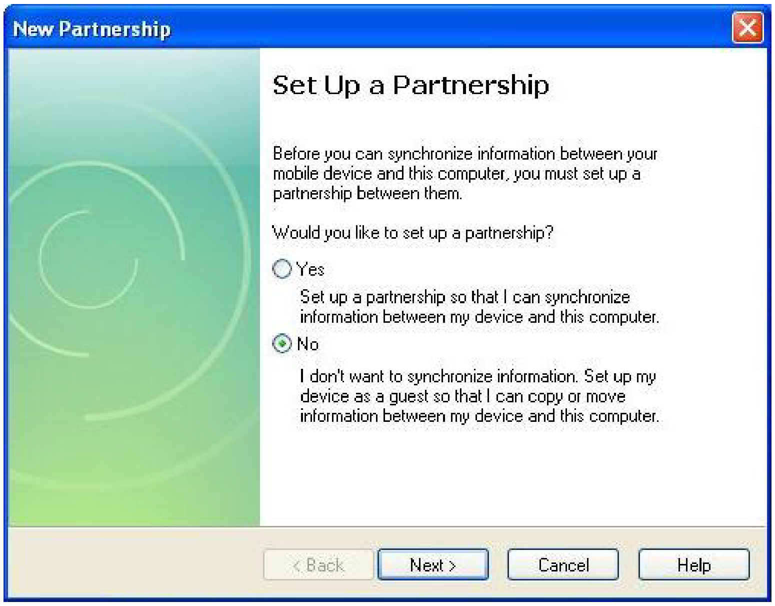


Figure 6 – *Contents of SBC-R9*

Windows ActiveSync for XP

If your host PC is running Windows XP, ActiveSync is required to establish connection to the SBC-R9. ActiveSync differs from 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 board, and the "New Partnership" dialog will appear. (See Figure 7.)



New Partnership

Set Up a Partnership

Before you can synchronize information between your mobile device and this computer, you must set up a partnership between them.

Would you like to set up a partnership?

○ Yes

Set up a partnership so that I can synchronize information between my device and this computer.

● No

I don't want to synchronize information. Set up my device as a guest so that I can copy or move information between my device and this computer.

Figure 7 - *ActiveSync New Partnership screen*

Using your mouse, select “No” and then select “Next”. The ActiveSync main dialog will appear. Click the “Explore” icon. This action opens a standard Windows Explorer where the default file contents of the SBC-R9 can be read or written. (See Figure 8.)

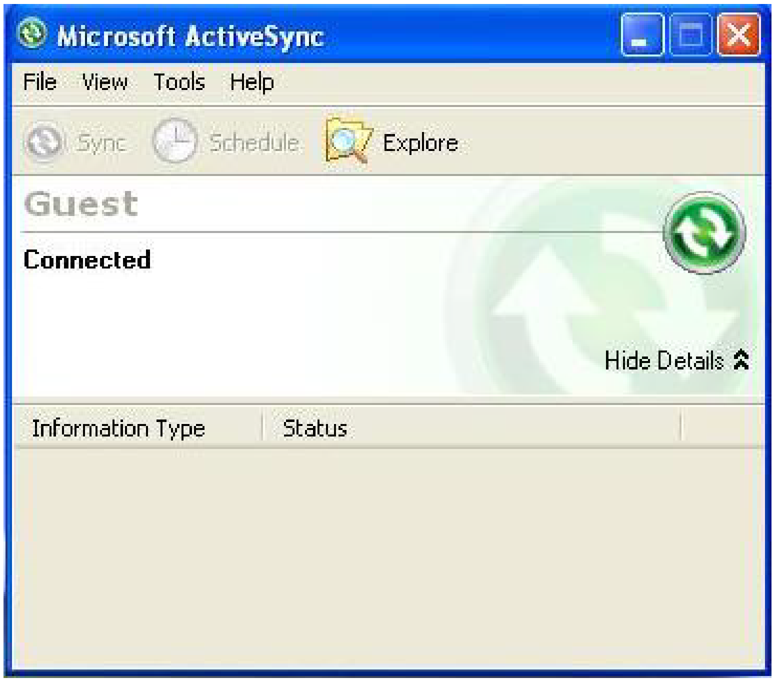


***Figure 8** - ActiveSync Main Dialog screen*

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.

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 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.

Our Talos Framework allows access to the more specific I/O sections of the SBC-R9 development board such as analog and digital I/O points, CAN bus, quadrature counter inputs, and the multi-electrical interface 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.

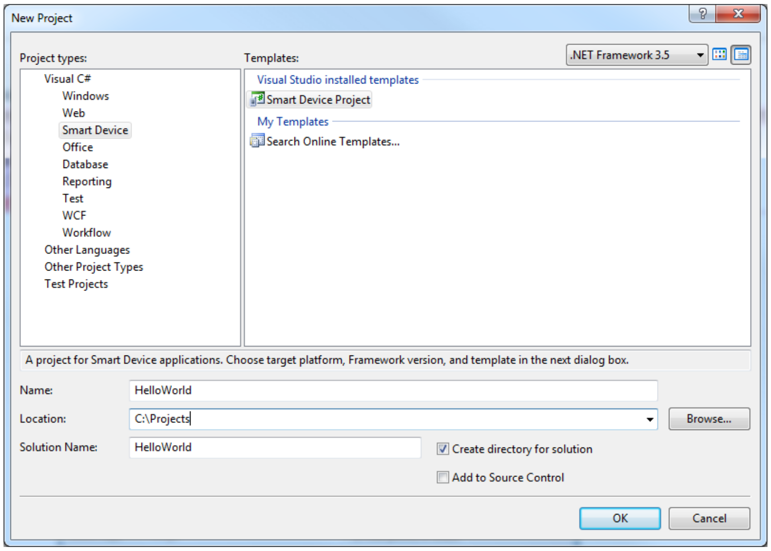
Writing .NET applications for the SBC-R9 is very similar to writing desktop or console applications for Windows XP, Vista, and 7. The only difference is the amount of resources available. Because the memory footprint is smaller compared to a desktop computer, care should be taken where allocation of 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 9.)



New Project

Project types: Visual C# → Smart Device

Templates: Smart Device Project

.NET Framework 3.5

A project for Smart Device applications. Choose target platform, Framework version, and template in the next dialog box.

Name: HelloWorld

Location: C:\Projects

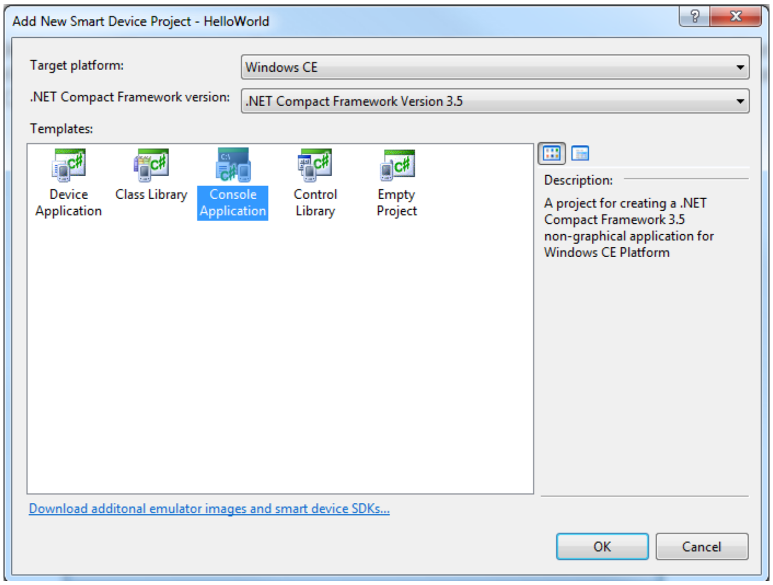
Solution Name: HelloWorld

☑ Create directory for solution

☐ Add to Source Control

Figure 9 *- 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 10.)



Add New Smart Device Project - HelloWorld

Target platform: Windows CE

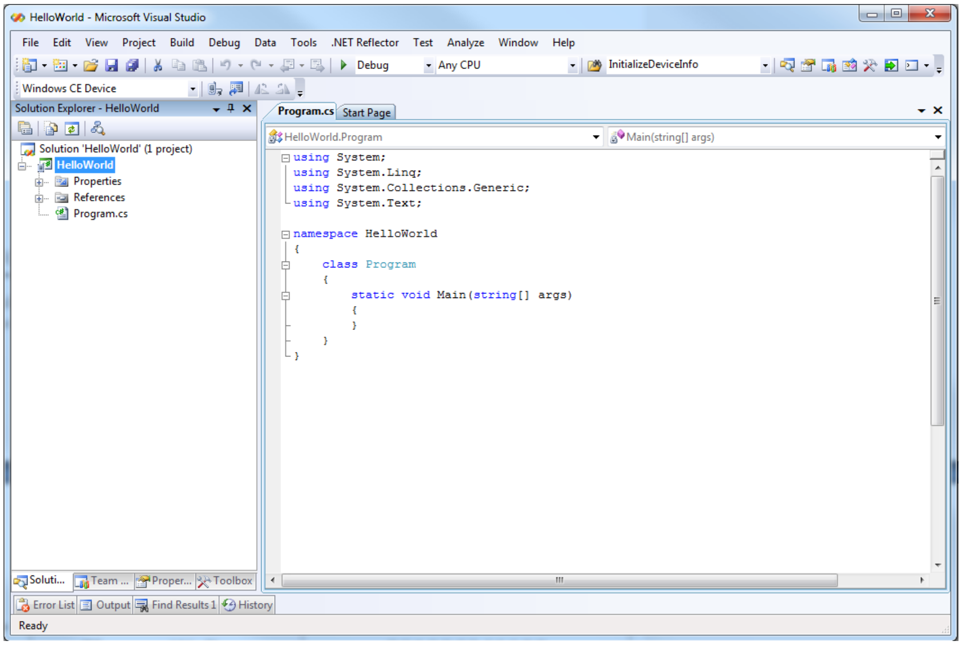
.NET Compact Framework version: .NET Compact Framework Version 3.5

Templates: Console Application

Description:
A project for creating a .NET Compact Framework 3.5 non-graphical application for Windows CE Platform

Figure 10 *- 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 11.)



```
using System;
using System.Linq;
using System.Collections.Generic;
using System.Text;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

Figure11 *- 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 12.)

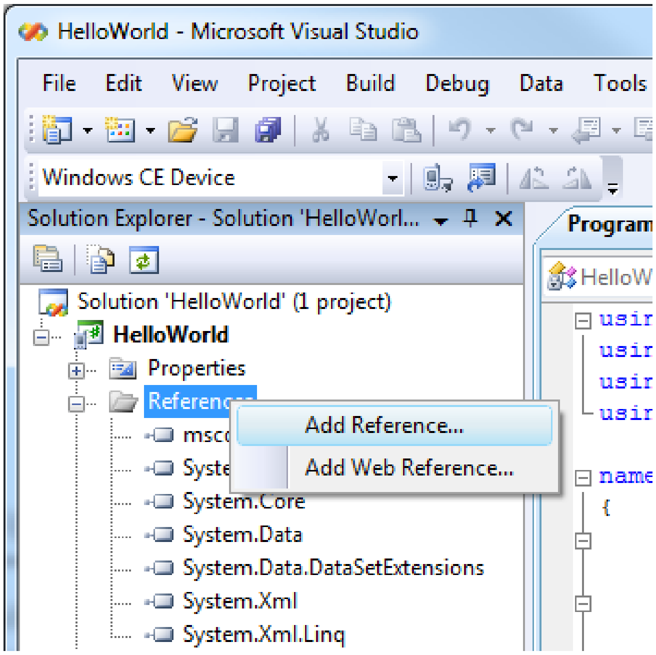
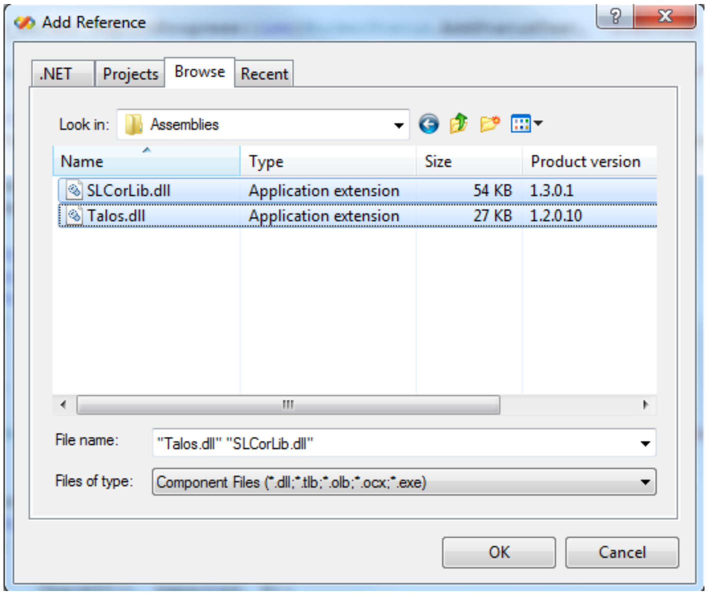

HelloWorld - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools
Windows CE Device
Solution Explorer - Solution 'HelloWorl...
Solution 'HelloWorld' (1 project)
HelloWorld
Properties
References
Add Reference...
Add Web Reference...
msco
Syste
System.Core
System.Data
System.Data.DataSetExtensions
System.Xml
System.Xml.Linq
Program
HelloW
usir
usir
usir
usir
name
{

Figure 12 *- 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 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 13.)



Look in: Assemblies

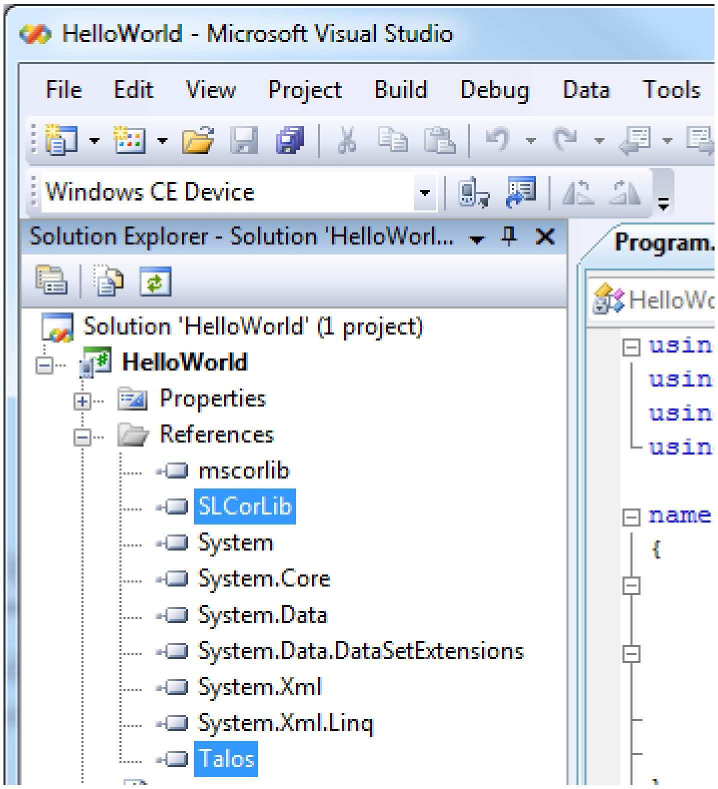
Name	Type	Size	Product version
SLCorLib.dll	Application extension	54 KB	1.3.0.1
Talos.dll	Application extension	27 KB	1.2.0.10

File name: "Talos.dll" "SLCorLib.dll"

Files of type: Component Files (*.dll;*.tlb;*.olb;*.ocx;*.exe)

Figure 13 *- Core library reference*

Both DLLs should appear in your “References” list. (See Figure 14.)



Solution 'HelloWorld' (1 project)
- **HelloWorld**
  - Properties
  - References
    - mscorlib
    - SLCorLib
    - System
    - System.Core
    - System.Data
    - System.Data.DataSetExtensions
    - System.Xml
    - System.Xml.Linq
    - Talos

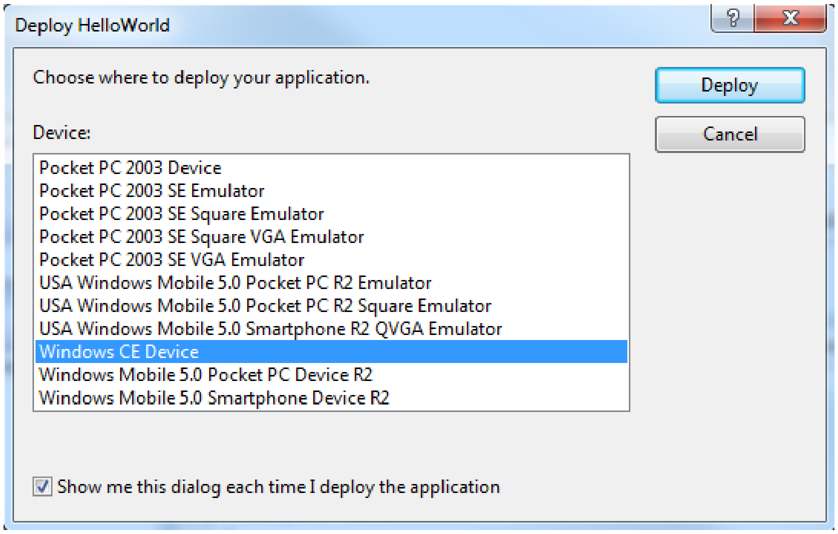
Figure 14 - Verification of added library references

Now that the Talos Framework has been referenced, you have access to all the I/O points exposed on the SBC-R9 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 15.)



Deploy HelloWorld

Choose where to deploy your application.

Device:

Pocket PC 2003 Device
Pocket PC 2003 SE Emulator
Pocket PC 2003 SE Square Emulator
Pocket PC 2003 SE Square VGA Emulator
Pocket PC 2003 SE VGA Emulator
USA Windows Mobile 5.0 Pocket PC R2 Emulator
USA Windows Mobile 5.0 Pocket PC R2 Square Emulator
USA Windows Mobile 5.0 Smartphone R2 QVGA Emulator
Windows CE Device
Windows Mobile 5.0 Pocket PC Device R2
Windows Mobile 5.0 Smartphone Device R2

☑ Show me this dialog each time I deploy the application

Figure 15 *- Choose Windows CE Device and Deploy*

After the deployment phase the "Hello World" message will appear on the Debug Serial console output.

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 embedded IO system. The SBC-R9 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 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 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 Quick Start section.

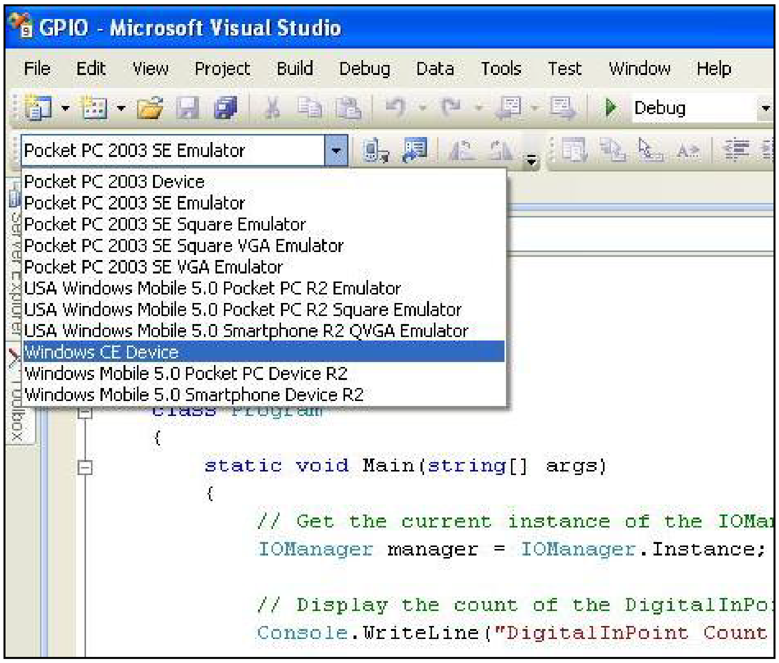
Debugging an Application

Once the SBC-R9 has been successfully attached to your PC, it is easy to begin debugging an application on the SBC-R9. This section will demonstrate how to attach the Microsoft Visual Studio debugger to the SBC-R9, 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.

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 16.)

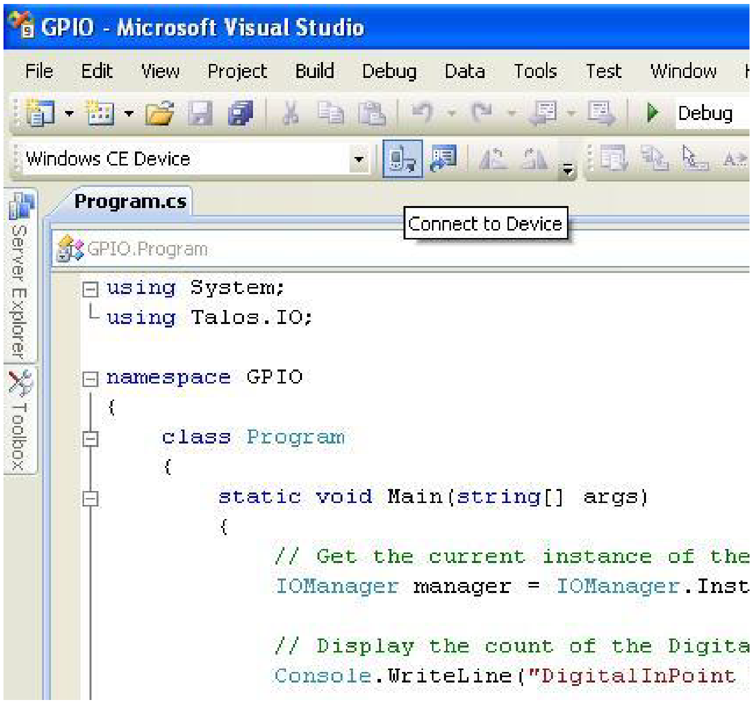

GPIO - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Debug
Pocket PC 2003 SE Emulator
Pocket PC 2003 Device
Pocket PC 2003 SE Emulator
Pocket PC 2003 SE Square Emulator
Pocket PC 2003 SE Square VGA Emulator
Pocket PC 2003 SE VGA Emulator
USA Windows Mobile 5.0 Pocket PC R2 Emulator
USA Windows Mobile 5.0 Pocket PC R2 Square Emulator
USA Windows Mobile 5.0 Smartphone R2 QVGA Emulator
Windows CE Device
Windows Mobile 5.0 Pocket PC Device R2
Windows Mobile 5.0 Smartphone Device R2
Server Explorer
Toolbox

```
    class Program
    {
        static void Main(string[] args)
        {
            // Get the current instance of the IOMa
            IOManager manager = IOManager.Instance;

            // Display the count of the DigitalInPo
            Console.WriteLine("DigitalInPoint Count
```

Figure 16 *- Device Target Selection*

If you would like to Use the faster Ethernet connection for debugging instead of the USB connection, refer to Appendix C – Application Debugging over Ethernet.

Now select the “Connect to Device” icon to initiate synchronization between Visual Studio and the SBC-R9 device. (See Figure 17.)

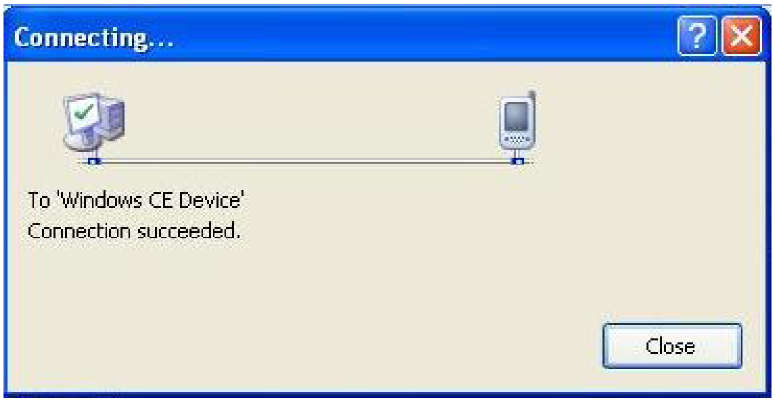

GPIO - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window
Debug
Windows CE Device
Program.cs
Connect to Device
GPIO.Program
Server Explorer
Toolbox

```
using System;
using Talos.IO;

namespace GPIO
{
    class Program
    {
        static void Main(string[] args)
        {
            // Get the current instance of the
            IOManager manager = IOManager.Inst

            // Display the count of the Digita
            Console.WriteLine("DigitalInPoint
```

***Figure 17** - Connect to Device icon*

.

You should now see a connection dialog appear. (See Figure 18.)

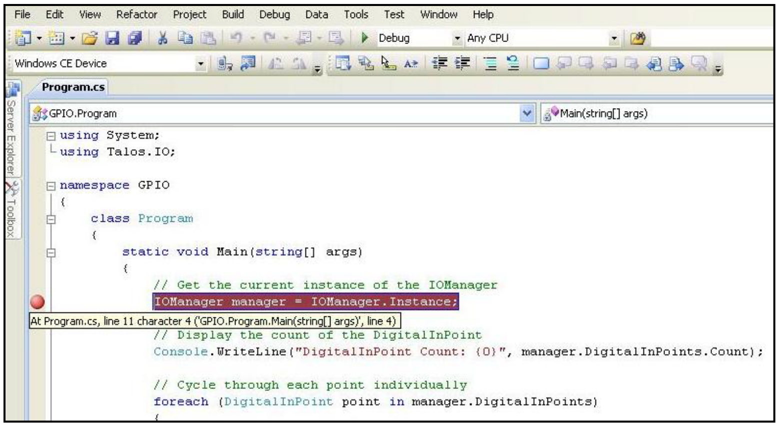


To 'Windows CE Device'
Connection succeeded.

Figure 18 - *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 19.)



```
using System;
using Talos.IO;

namespace GPIO
{
    class Program
    {
        static void Main(string[] args)
        {
            // Get the current instance of the IOManager
            IOManager manager = IOManager.Instance;
At Program.cs, line 11 character 4 ('GPIO.Program.Main(string[] args)', line 4)
            // Display the count of the DigitalInPoint
            Console.WriteLine("DigitalInPoint Count: {0}", manager.DigitalInPoints.Count);

            // Cycle through each point individually
            foreach (DigitalInPoint point in manager.DigitalInPoints)
            {
```

Figure 19 - Breakpoint selection

To begin debugging the application, click the "Start Debugging" button. (See Figure 20.)

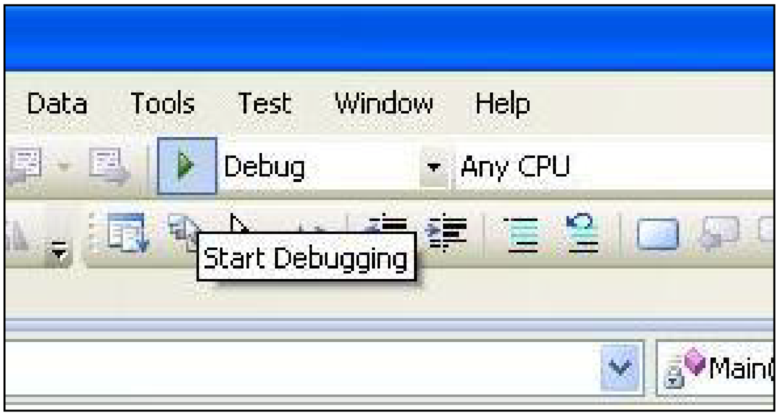
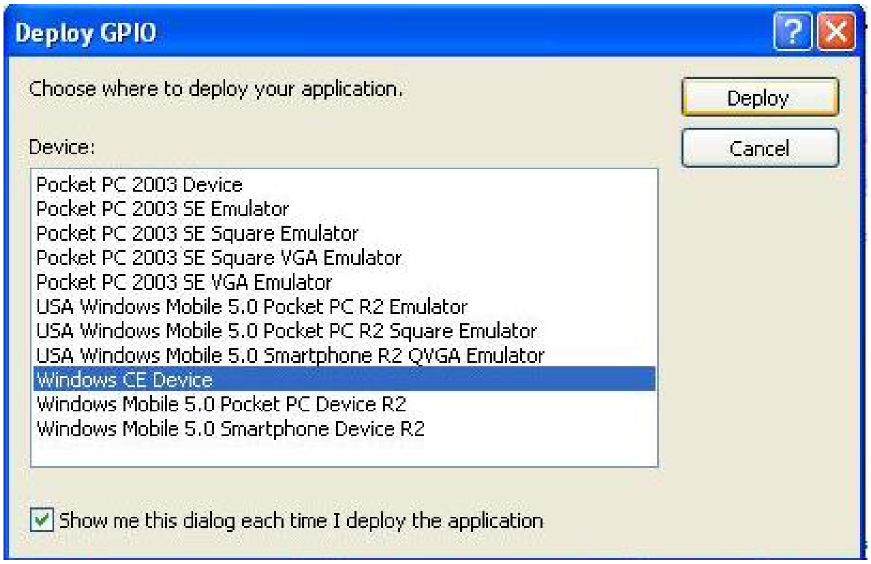


Figure 20 - Run Debugger icon

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 21.)



Deploy GPIO

Choose where to deploy your application.

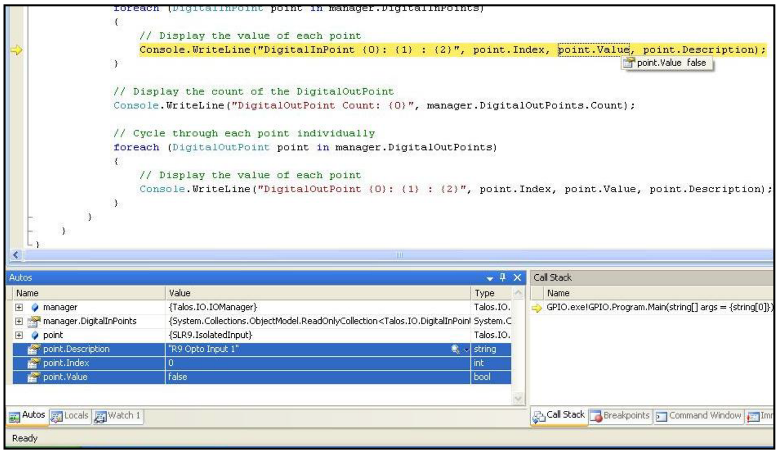
Device:

- Pocket PC 2003 Device
- Pocket PC 2003 SE Emulator
- Pocket PC 2003 SE Square Emulator
- Pocket PC 2003 SE Square VGA Emulator
- Pocket PC 2003 SE VGA Emulator
- USA Windows Mobile 5.0 Pocket PC R2 Emulator
- USA Windows Mobile 5.0 Pocket PC R2 Square Emulator
- USA Windows Mobile 5.0 Smartphone R2 QVGA Emulator
- Windows CE Device
- Windows Mobile 5.0 Pocket PC Device R2
- Windows Mobile 5.0 Smartphone Device R2

☑ Show me this dialog each time I deploy the application

Figure 21. Target Deployment dialog

Once the application is deployed to the SBC-R9, 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 22.)



```
            foreach (DigitalInPoint point in manager.DigitalInPoints)
            {
                // Display the value of each point
                Console.WriteLine("DigitalInPoint {0}: {1} : {2}", point.Index, point.Value, point.Description);
            }

            // Display the count of the DigitalOutPoint
            Console.WriteLine("DigitalOutPoint Count: {0}", manager.DigitalOutPoints.Count);

            // Cycle through each point individually
            foreach (DigitalOutPoint point in manager.DigitalOutPoints)
            {
                // Display the value of each point
                Console.WriteLine("DigitalOutPoint {0}: {1} : {2}", point.Index, point.Value, point.Description);
            }
        }
    }
}
```

point.Value false

Autos

Name	Value	Type
manager	{Talos.IO.IOManager}	Talos.IO.
manager.DigitalInPoints	{System.Collections.ObjectModel.ReadOnlyCollection<Talos.IO.DigitalInPoin	System.C
point	{SLR9.IsolatedInput}	Talos.IO.
point.Description	"R9 Opto Input 1"	string
point.Index	0	int
point.Value	false	bool

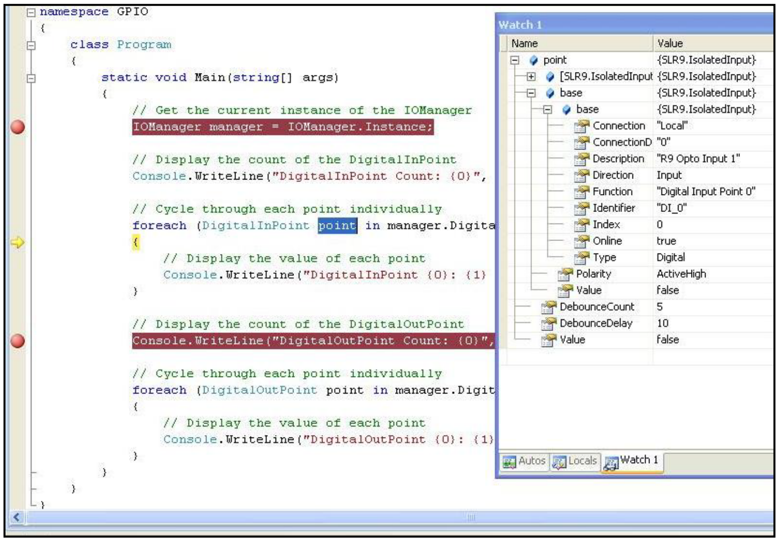
Call Stack

Name
GPIO.exe!GPIO.Program.Main(string[] args = {string[0]})

Figure 22 - *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 23.) Each watch window provides a convenient tree type structure for viewing hierarchical class variables.



```
namespace GPIO
{
    class Program
    {
        static void Main(string[] args)
        {
            // Get the current instance of the IOManager
            IOManager manager = IOManager.Instance;

            // Display the count of the DigitalInPoint
            Console.WriteLine("DigitalInPoint Count: {0}",

            // Cycle through each point individually
            foreach (DigitalInPoint point in manager.Digita
            {
                // Display the value of each point
                Console.WriteLine("DigitalInPoint {0}: {1}
            }

            // Display the count of the DigitalOutPoint
            Console.WriteLine("DigitalOutPoint Count: {0}",

            // Cycle through each point individually
            foreach (DigitalOutPoint point in manager.Digit
            {
                // Display the value of each point
                Console.WriteLine("DigitalOutPoint {0}: {1}
            }
        }
    }
}
```

Watch 1

Name	Value
point	{SLR9.IsolatedInput}
[SLR9.IsolatedInput	{SLR9.IsolatedInput}
base	{SLR9.IsolatedInput}
base	{SLR9.IsolatedInput}
Connection	"Local"
ConnectionD	"0"
Description	"R9 Opto Input 1"
Direction	Input
Function	"Digital Input Point 0"
Identifier	"DI_0"
Index	0
Online	true
Type	Digital
Polarity	ActiveHigh
Value	false
DebounceCount	5
DebounceDelay	10
Value	false

Figure 23 - *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 onto the SDCARD or 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 on the SDCARD or on-board NAND Flash. To accomplish this you will need to establish connectivity via Mobile Device Center or ActiveSync as outlined in the SBC-R9 Quick Start section above.

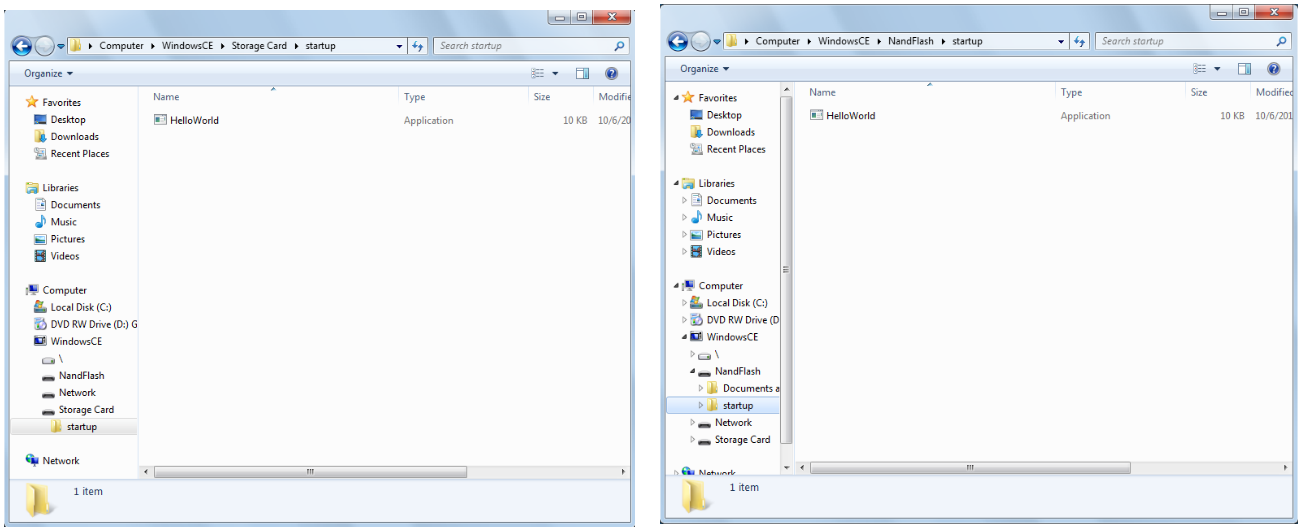
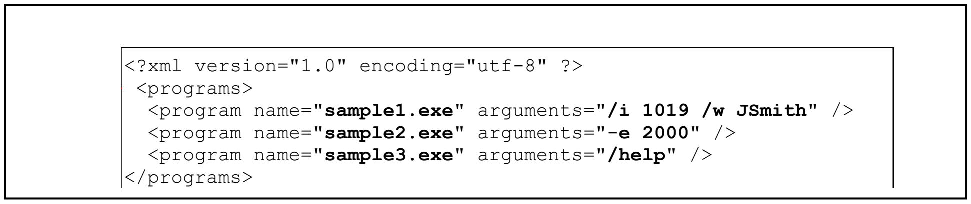


Figure 24- *Application Placement*

The SBC-R9 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 '/Storage Card/startup/' or `/nandflash/startup/'. After Windows CE runs, SpringBoard automatically starts applications located in the NAND Flash followed by applications in the Storage Card.

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: '/storage card/startup/startup.xml' and/or '/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 included for 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 25 - *startup.xml*

SDCARD Boot Sequence

Upon power up, the SBC-R9 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 locations include SDCARD and NAND Flash. Ultimately, eboot attempts to load and execute the OS runtime image based on those configuration settings.

The SBC-R9 development board checks the root directory of the bottom SDCARD for a valid Eboot boot loader (boot.bin). The file must be named boot.bin and the SDCARD must be formatted as FAT 12/16/32. If no boot image is found, the device will next check the raw data in the NAND Flash.

Only the bottom SDCARD slot (Slot A) or NAND Flash can be used for booting to an OS runtime image. .

The SBC-R9 ships with an SDCARD loaded with the OS files listed below:

- Boot.bin
- Eboot.bin
- NK.bin

OS File Restoration

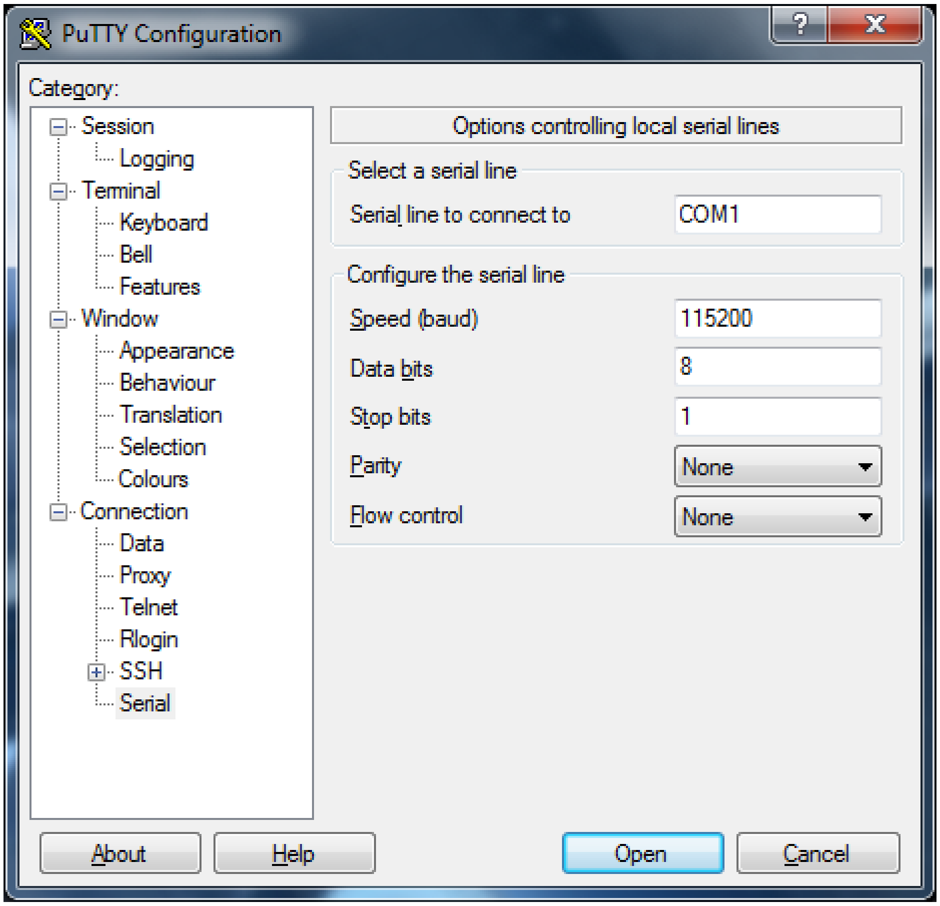
In the event that Sealevel produces updated OS file versions, or a backup is desired, the OS files will need to be copied to the root directory of an SDCARD or programmed to the NAND Flash. There are a variety of ways to copy files to the SDCARD; please see the section labeled “Upgrading the OS runtime image on SDCARD” below for more detail. Please see the section labeled “Upgrading the OS runtime image on NAND Flash” for further detail into that process. The NAND Flash cannot be programmed until any existing OS runtime image has been removed and the SDCARD is removed or the OS image on it is removed.

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 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 RS-232 cable into the SBC-R9 connector (J6). Connect the DB9 end of the SBC-R9 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 26.)



Options controlling local serial lines

Select a serial line

Serial line to connect to: COM1

Configure the serial line

Speed (baud): 115200

Data bits: 8

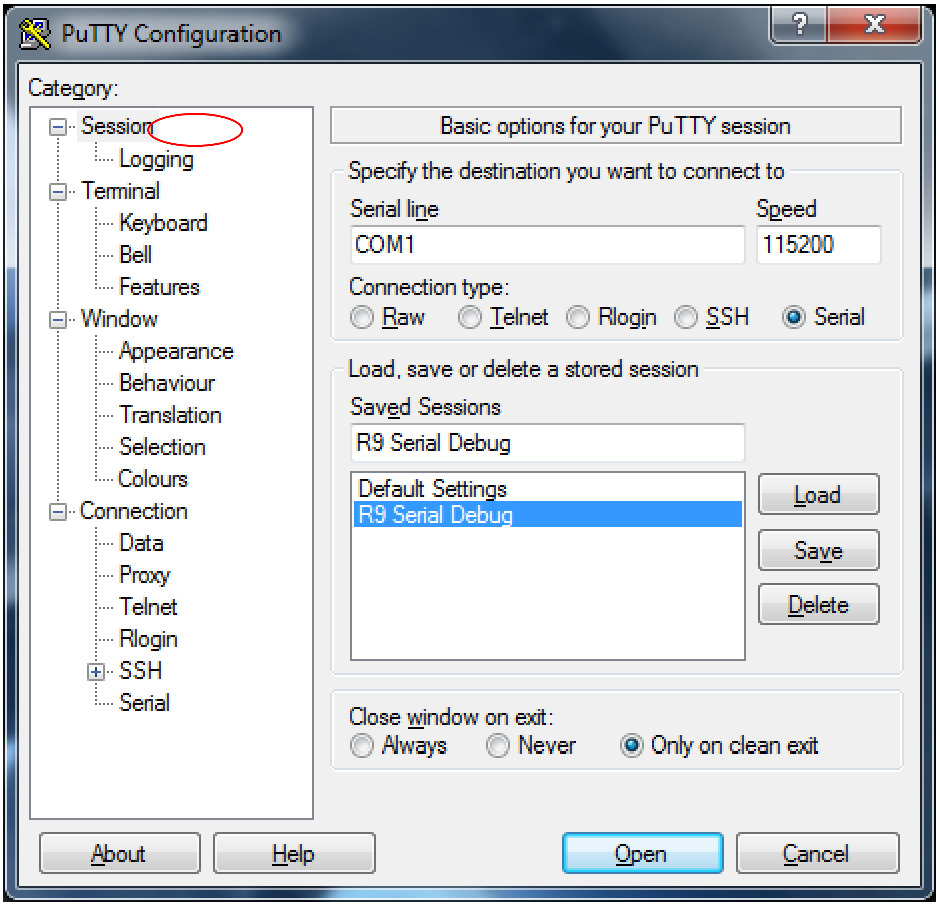
Stop bits: 1

Parity: None

Flow control: None

Figure 26- 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. Next select Serial for the connection type. Type a name for this session under “Saved Sessions”, then press the “Save” button. (See Figure 27.)



Specify the destination you want to connect to

Serial line: COM1

Speed: 115200

Connection type: ○ Raw ○ Telnet ○ Rlogin ○ SSH ● Serial

Load, save or delete a stored session

Saved Sessions: R9 Serial Debug

Default Settings
R9 Serial Debug

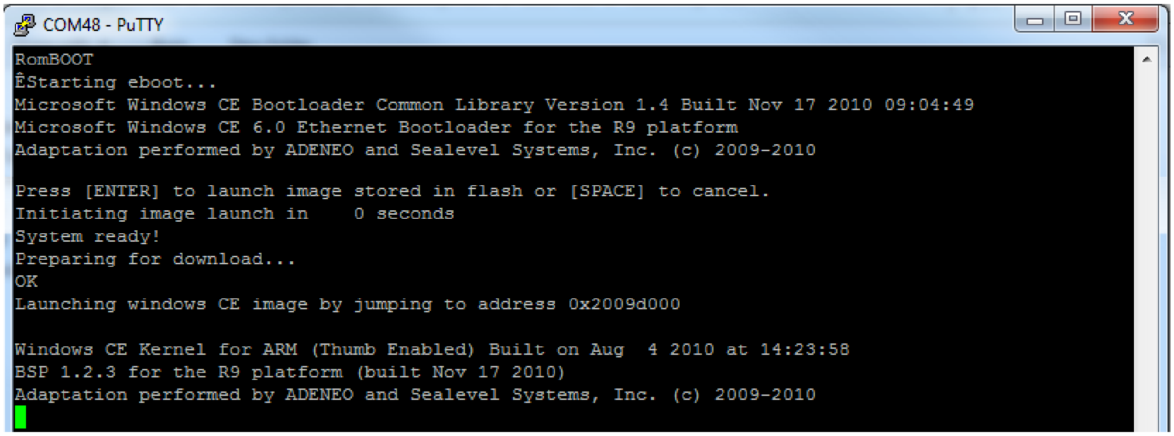
Close window on exit: ○ Always ○ Never ● Only on clean exit

***Figure 27-** 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 board. Press the reset button on the SBC-R9 to display the Ethernet boot loader configuration screen. (See Figure 28.) When the unit boots, the following menu 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"
```

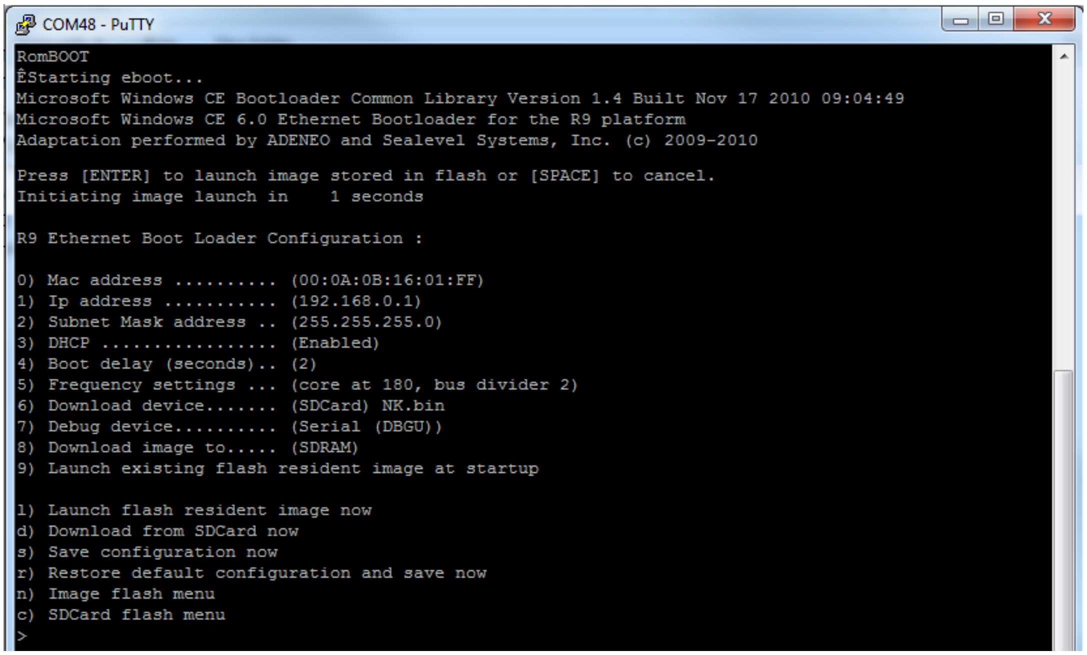
Once the prompt period expires, the OS runtime will be loaded from SDCARD or NAND Flash (depending on boot sequence and boot files available) into RAM and executed. At this point, the OS is running, and all console output is redirected to the debug serial port. (See Figure 28.)



```
RomBOOT
ÊStarting eboot...
Microsoft Windows CE Bootloader Common Library Version 1.4 Built Nov 17 2010 09:04:49
Microsoft Windows CE 6.0 Ethernet Bootloader for the R9 platform
Adaptation performed by ADENEO and Sealevel Systems, Inc. (c) 2009-2010

Press [ENTER] to launch image stored in flash or [SPACE] to cancel.
Initiating image launch in    0 seconds
System ready!
Preparing for download...
OK
Launching windows CE image by jumping to address 0x2009d000

Windows CE Kernel for ARM (Thumb Enabled) Built on Aug  4 2010 at 14:23:58
BSP 1.2.3 for the R9 platform (built Nov 17 2010)
Adaptation performed by ADENEO and Sealevel Systems, Inc. (c) 2009-2010
```

Figure 28- 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 29.)

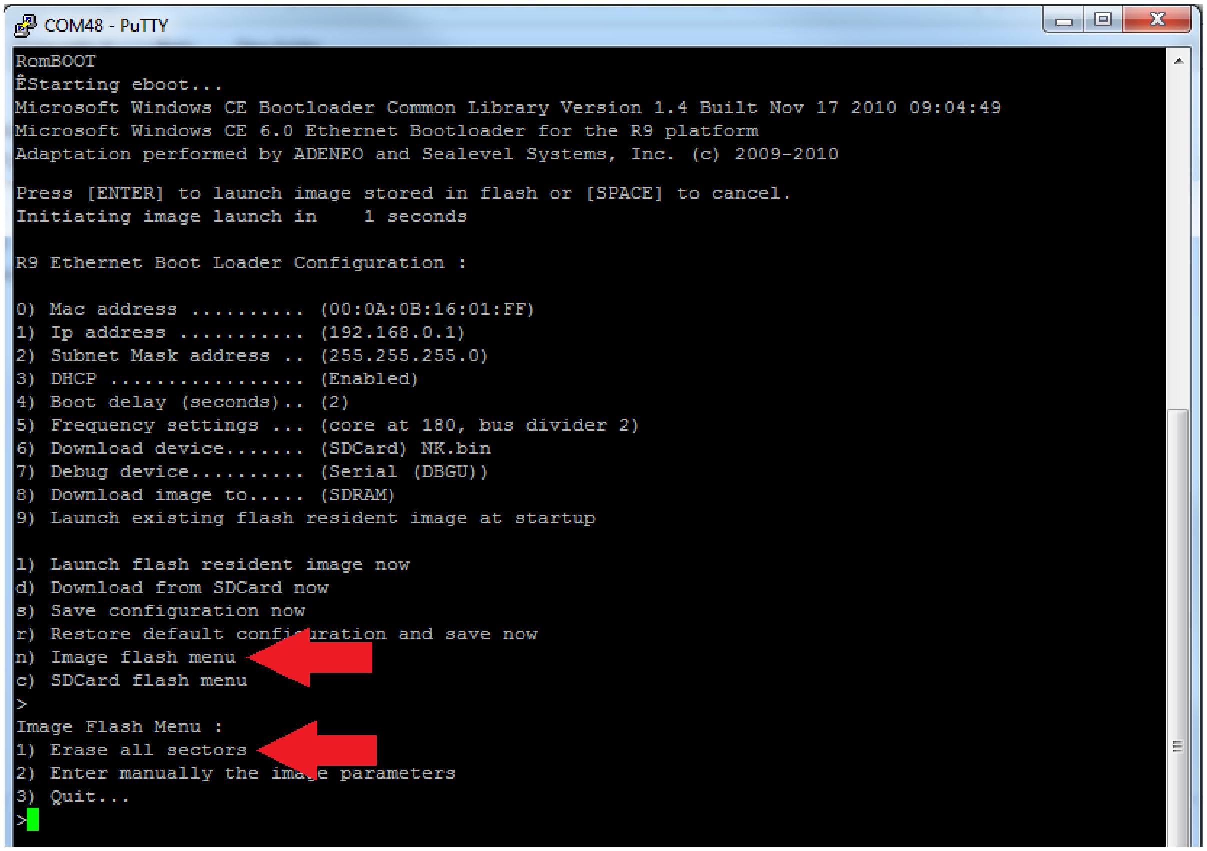


```
RomBOOT
ÊStarting eboot...
Microsoft Windows CE Bootloader Common Library Version 1.4 Built Nov 17 2010 09:04:49
Microsoft Windows CE 6.0 Ethernet Bootloader for the R9 platform
Adaptation performed by ADENEO and Sealevel Systems, Inc. (c) 2009-2010

Press [ENTER] to launch image stored in flash or [SPACE] to cancel.
Initiating image launch in    1 seconds

R9 Ethernet Boot Loader Configuration :

0) Mac address .......... (00:0A:0B:16:01:FF)
1) Ip address ........... (192.168.0.1)
2) Subnet Mask address .. (255.255.255.0)
3) DHCP ................. (Enabled)
4) Boot delay (seconds).. (2)
5) Frequency settings ... (core at 180, bus divider 2)
6) Download device....... (SDCard) NK.bin
7) Debug device.......... (Serial (DBGU))
8) Download image to..... (SDRAM)
9) Launch existing flash resident image at startup

l) Launch flash resident image now
d) Download from SDCard now
s) Save configuration now
r) Restore default configuration and save now
n) Image flash menu
c) SDCard flash menu
>
```

Figure 29. Eboot configuration output

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

When upgrading an existing OS runtime stored in the NAND Flash, 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 30.)



```
RomBOOT
ÊStarting eboot...
Microsoft Windows CE Bootloader Common Library Version 1.4 Built Nov 17 2010 09:04:49
Microsoft Windows CE 6.0 Ethernet Bootloader for the R9 platform
Adaptation performed by ADENEO and Sealevel Systems, Inc. (c) 2009-2010

Press [ENTER] to launch image stored in flash or [SPACE] to cancel.
Initiating image launch in    1 seconds

R9 Ethernet Boot Loader Configuration :

0) Mac address .......... (00:0A:0B:16:01:FF)
1) Ip address ........... (192.168.0.1)
2) Subnet Mask address .. (255.255.255.0)
3) DHCP ................. (Enabled)
4) Boot delay (seconds).. (2)
5) Frequency settings ... (core at 180, bus divider 2)
6) Download device....... (SDCard) NK.bin
7) Debug device.......... (Serial (DBGU))
8) Download image to..... (SDRAM)
9) Launch existing flash resident image at startup

l) Launch flash resident image now
d) Download from SDCard now
s) Save configuration now
r) Restore default configuration and save now
n) Image flash menu
c) SDCard flash menu
>
Image Flash Menu :
1) Erase all sectors
2) Enter manually the image parameters
3) Quit...
>
```

***Figure 30**-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 SDCARD

Factory OS runtime images are stored in the “Boot Files” directory of the R9 Development installation (see Quick start guide). There is a few ways to upgrade the OS runtime image (*.bin) located on your bootable SDCARD:

- a memory card reader (preferred method)
- USB connection with Windows Mobile Device Center or ActiveSync
- FTP connection.

Memory Card Reader

When inserting the SDCARD into your memory card reader, you may be prompted with an “AutoPlay” option. Choose “Open folder to view files”. If the “AutoPlay” feature has been disabled, navigate to the memory card reader manually. (See Figure 31).

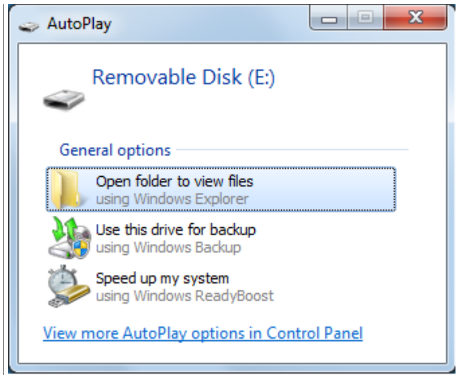
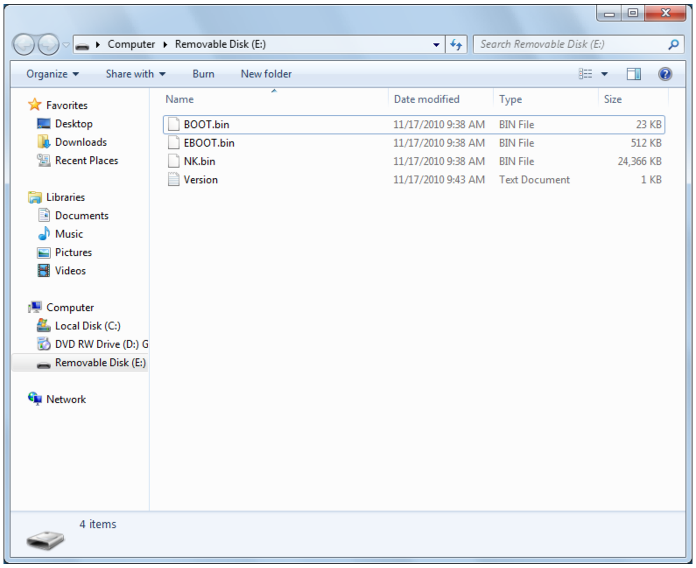


Figure 31-*AutoPlay screen*

The OS runtime image consists of three binary (.bin) files as demonstrated below. (See Figure 32).

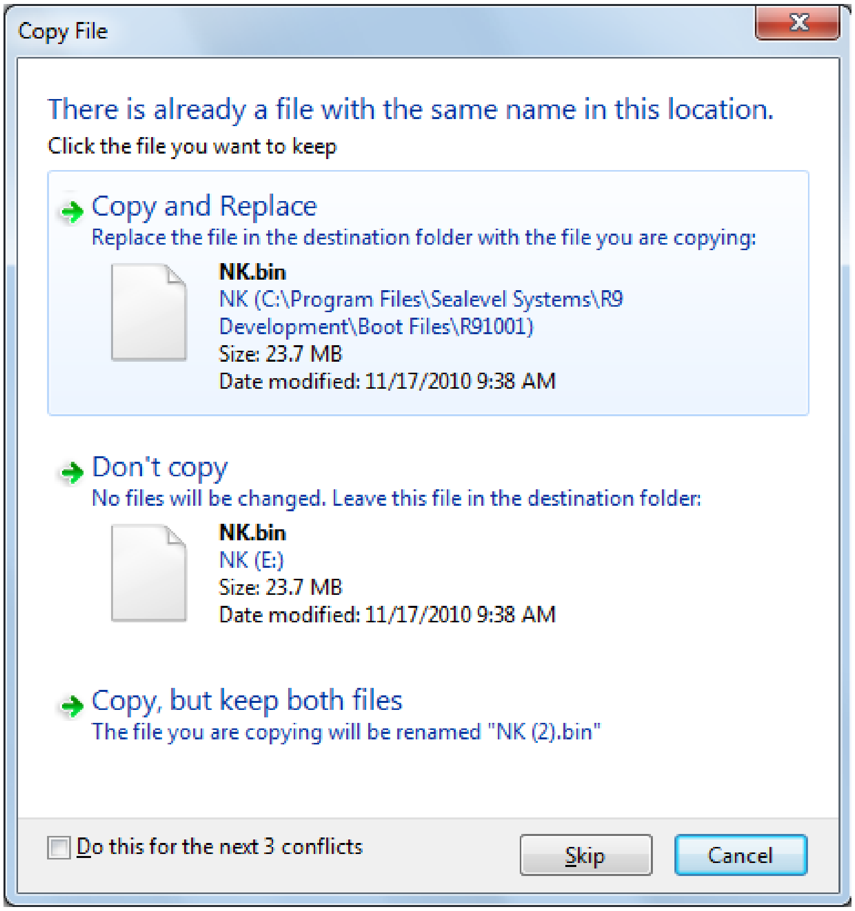


Name	Date modified	Type	Size
BOOT.bin	11/17/2010 9:38 AM	BIN File	23 KB
EBOOT.bin	11/17/2010 9:38 AM	BIN File	512 KB
NK.bin	11/17/2010 9:38 AM	BIN File	24,366 KB
Version	11/17/2010 9:43 AM	Text Document	1 KB

Figure 32-SDCARD File Contents

To save the existing OS runtime Image, backup the files previously stored on the SDCARD. .

Copy the new OS runtime image to the SDCARD. A popup will appear asking you to override your current files. Select the "Copy and Replace" option to over-write the existing OS runtime image. The new OS runtime image will be loaded the next time the device is booted with the SDCARD. (See Figure 33.)



Copy File

There is already a file with the same name in this location.

Click the file you want to keep

Copy and Replace

Replace the file in the destination folder with the file you are copying:

NK.bin
NK (C:\Program Files\Sealevel Systems\R9 Development\Boot Files\R91001)
Size: 23.7 MB
Date modified: 11/17/2010 9:38 AM

Don't copy

No files will be changed. Leave this file in the destination folder:

NK.bin
NK (E:)
Size: 23.7 MB
Date modified: 11/17/2010 9:38 AM

Copy, but keep both files

The file you are copying will be renamed "NK (2).bin"

□ Do this for the next 3 conflicts

Figure 33*-Copy and Replace*

USB Connection: Using Windows Mobile Device Center or ActiveSync

Another way to upgrade the OS runtime image is to connect via Windows Mobile Device Center or Active Sync; for instructions on installing Windows Mobile Device Center or ActiveSync refer to the SBC-R9 Quick Start section above.

Using the device file explorer, navigate to the "Storage Card" folder to view the SDCARD contents. (See Figures 34/35.)

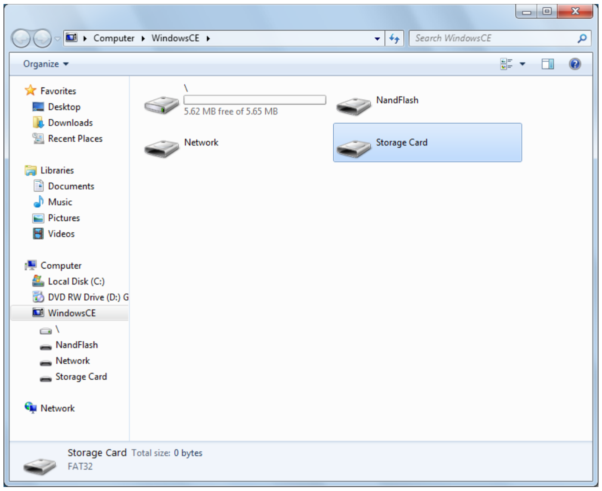
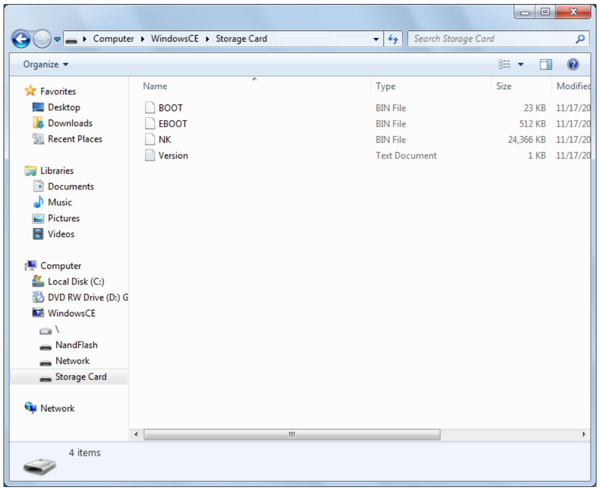


Figure 34- *WindowsCE Device Explore*

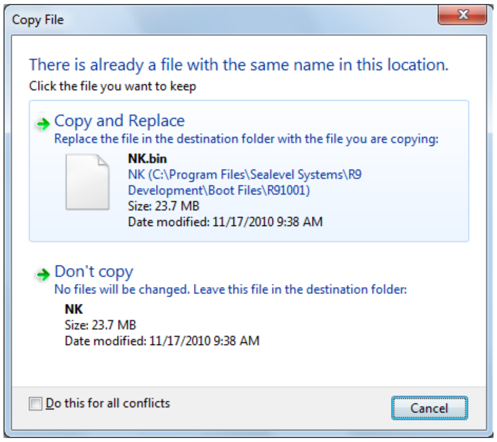


Name	Type	Size	Modified
BOOT	BIN File	23 KB	11/17/20
EBOOT	BIN File	512 KB	11/17/20
NK	BIN File	24,366 KB	11/17/20
Version	Text Document	1 KB	11/17/20

Figure 35-*Storage Card contents*

N To save the existing OS runtime image, backup the SD Card contents.

Copy the new OS runtime image to the SDCARD. A popup will appear asking you to over-write your current files. (See Figure 36.)



There is already a file with the same name in this location.

Click the file you want to keep

Copy and Replace

Replace the file in the destination folder with the file you are copying:

NK.bin
NK (C:\Program Files\Sealevel Systems\R9 Development\Boot Files\R91001)
Size: 23.7 MB
Date modified: 11/17/2010 9:38 AM

Don't copy

No files will be changed. Leave this file in the destination folder:

NK
Size: 23.7 MB
Date modified: 11/17/2010 9:38 AM

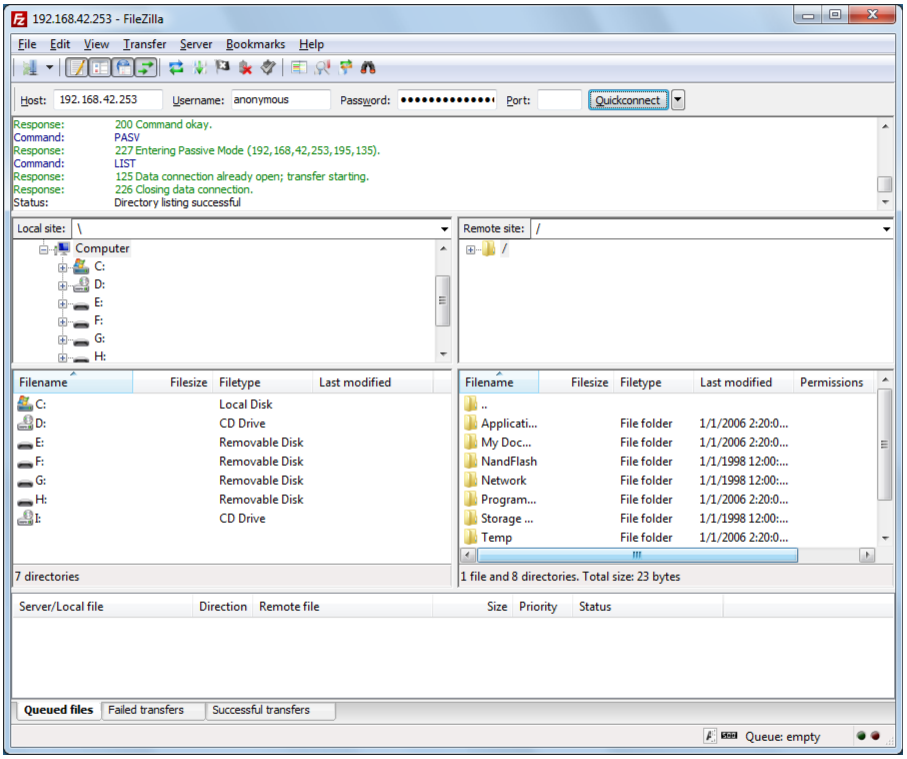
□ Do this for all conflicts

Figure 36-*Copy and Replace*

Select the "Copy and Replace" option to over-write the existing OS runtime image. Reboot the SBC-R9 once the file has been copied. The new OS runtime image will be loaded on bootup.

FTP Connection

Use an FTP program to connect to the SBC-R9 and upload the new OS runtime image to the SDCARD. FileZilla (See Appendix A), an open-source FTP client, is used in the example below. By default, FTP is open to anonymous access with no password needed. (See Figure 37.)



Host: 192.168.42.253 Username: anonymous Password: •••••••••••••• Port:

```
Response:     200 Command okay.
Command:      PASV
Response:     227 Entering Passive Mode (192,168,42,253,195,135).
Command:      LIST
Response:     125 Data connection already open; transfer starting.
Response:     226 Closing data connection.
Status:       Directory listing successful
```

Local site: \

Computer
- C:
- D:
- E:
- F:
- G:
- H:

Filename	Filesize	Filetype	Last modified
C:		Local Disk	
D:		CD Drive	
E:		Removable Disk	
F:		Removable Disk	
G:		Removable Disk	
H:		Removable Disk	
I:		CD Drive	

7 directories

Remote site: /

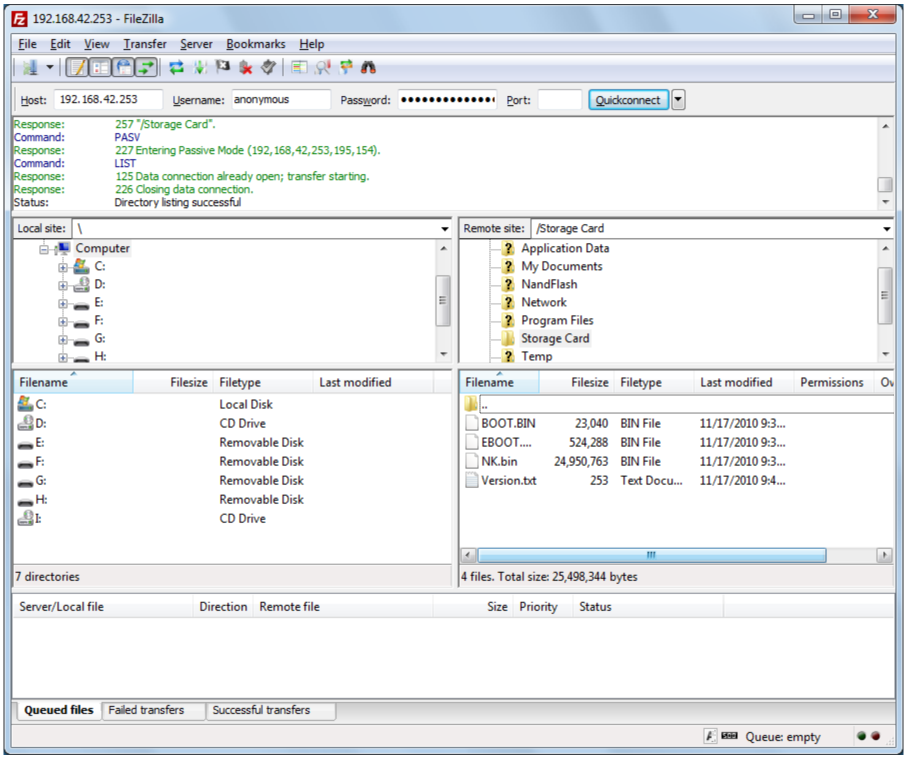
Filename	Filesize	Filetype	Last modified	Permissions
..				
Applicati...		File folder	1/1/2006 2:20:0...	
My Doc...		File folder	1/1/2006 2:20:0...	
NandFlash		File folder	1/1/1998 12:00:...	
Network		File folder	1/1/1998 12:00:...	
Program...		File folder	1/1/2006 2:20:0...	
Storage ...		File folder	1/1/1998 12:00:...	
Temp		File folder	1/1/2006 2:20:0...	

1 file and 8 directories. Total size: 23 bytes

Server/Local file	Direction	Remote file	Size	Priority	Status

Figure 37 - Connect to device through FTP

Select the "Storage Card" folder for the remote site. (See Figure 38.)



192.168.42.253 - FileZilla
File Edit View Transfer Server Bookmarks Help

Host: 192.168.42.253 Username: anonymous Password: •••••••••••••• Port:

```
Response:    257 "/Storage Card".
Command:     PASV
Response:    227 Entering Passive Mode (192,168,42,253,195,154).
Command:     LIST
Response:    125 Data connection already open; transfer starting.
Response:    226 Closing data connection.
Status:      Directory listing successful
```

Local site: \

- Computer
  - C:
  - D:
  - E:
  - F:
  - G:
  - H:

Remote site: /Storage Card

- Application Data
- My Documents
- NandFlash
- Network
- Program Files
- Storage Card
- Temp

Filename	Filesize	Filetype	Last modified
C:		Local Disk	
D:		CD Drive	
E:		Removable Disk	
F:		Removable Disk	
G:		Removable Disk	
H:		Removable Disk	
I:		CD Drive	

7 directories

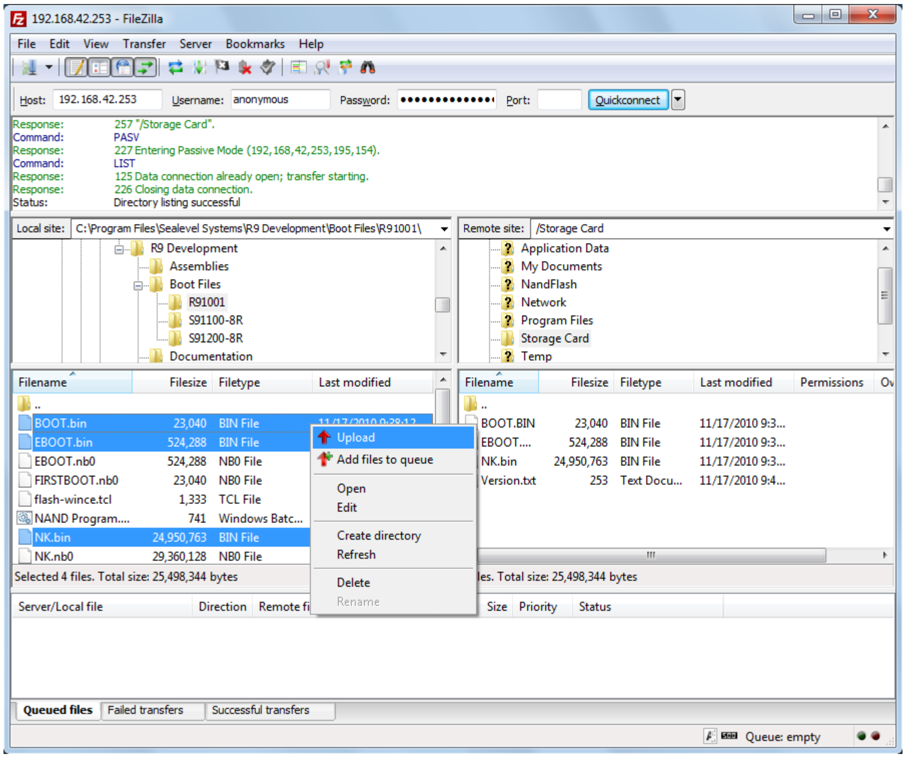
Filename	Filesize	Filetype	Last modified	Permissions	Ov
..					
BOOT.BIN	23,040	BIN File	11/17/2010 9:3...		
EBOOT....	524,288	BIN File	11/17/2010 9:3...		
NK.bin	24,950,763	BIN File	11/17/2010 9:3...		
Version.txt	253	Text Docu...	11/17/2010 9:4...		

4 files. Total size: 25,498,344 bytes

Server/Local file	Direction	Remote file	Size	Priority	Status

Figure 38-Select Storage Card

Navigate to the “Boot Files” directory of the R9 Development installation. Select the OS Runtime files to copy (*.bin). Right-click and select “Upload” to begin the file transfer. (See Figure 39.)



Host: 192.168.42.253 Username: anonymous Password: •••••••••••••• Port:

```
Response:    257 "/Storage Card".
Command:     PASV
Response:    227 Entering Passive Mode (192,168,42,253,195,154).
Command:     LIST
Response:    125 Data connection already open; transfer starting.
Response:    226 Closing data connection.
Status:      Directory listing successful
```

Local site: C:\Program Files\Sealevel Systems\R9 Development\Boot Files\R91001\

```
R9 Development
    Assemblies
    Boot Files
        R91001
        S91100-8R
        S91200-8R
    Documentation
```

Remote site: /Storage Card

```
Application Data
My Documents
NandFlash
Network
Program Files
Storage Card
Temp
```

Filename	Filesize	Filetype	Last modified
..			
BOOT.bin	23,040	BIN File	11/17/2010 9:38:12
EBOOT.bin	524,288	BIN File	
EBOOT.nb0	524,288	NB0 File	
FIRSTBOOT.nb0	23,040	NB0 File	
flash-wince.tcl	1,333	TCL File	
NAND Program....	741	Windows Batc...	
NK.bin	24,950,763	BIN File	
NK.nb0	29,360,128	NB0 File	

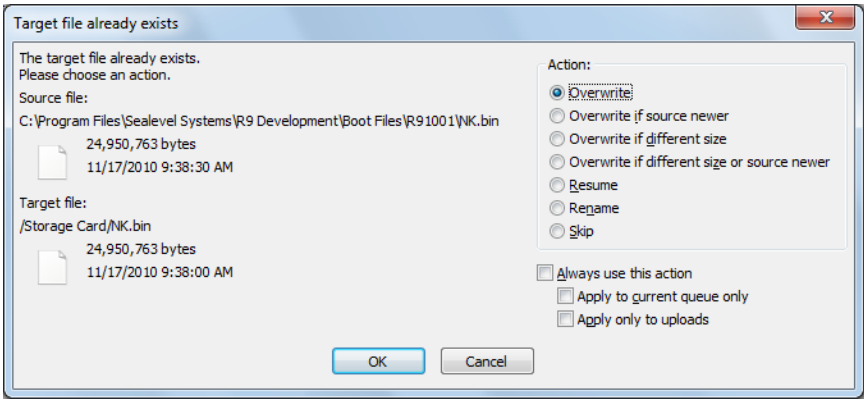
Selected 4 files. Total size: 25,498,344 bytes

Filename	Filesize	Filetype	Last modified	Permissions	Ov
..					
BOOT.BIN	23,040	BIN File	11/17/2010 9:3...		
EBOOT....	524,288	BIN File	11/17/2010 9:3...		
NK.bin	24,950,763	BIN File	11/17/2010 9:3...		
Version.txt	253	Text Docu...	11/17/2010 9:4...		

les. Total size: 25,498,344 bytes

Figure 39-Select boot files to update

You may be asked how to proceed when replacing existing files. Select the “Overwrite” radio button and click “OK”. (See Figure 40.)



The target file already exists.
Please choose an action.

Source file:
C:\Program Files\Sealevel Systems\R9 Development\Boot Files\R91001\NK.bin
24,950,763 bytes
11/17/2010 9:38:30 AM

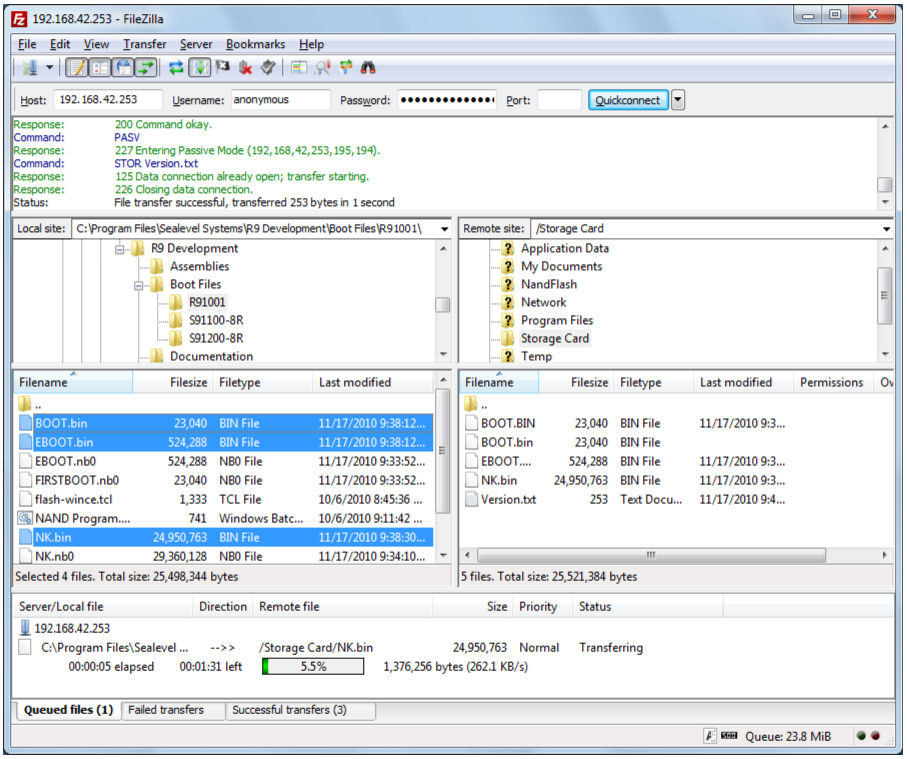
Target file:
/Storage Card/NK.bin
24,950,763 bytes
11/17/2010 9:38:00 AM

Action:
- ● Overwrite
- ○ Overwrite if source newer
- ○ Overwrite if different size
- ○ Overwrite if different size or source newer
- ○ Resume
- ○ Rename
- ○ Skip

□ Always use this action
□ Apply to current queue only
□ Apply only to uploads

Figure 40-*Over-write files*

Once the files have been uploaded, (See Figure 41.) reboot the device. The new OS runtime image will be loaded on bootup.



Host: 192.168.42.253 Username: anonymous Password: •••••••••••••• Port:

```
Response:    200 Command okay.
Command:     PASV
Response:    227 Entering Passive Mode (192,168,42,253,195,194).
Command:     STOR Version.txt
Response:    125 Data connection already open; transfer starting.
Response:    226 Closing data connection.
Status:      File transfer successful, transferred 253 bytes in 1 second
```

Local site: C:\Program Files\Sealevel Systems\R9 Development\Boot Files\R91001\

- R9 Development
  - Assemblies
  - Boot Files
    - R91001
    - S91100-8R
    - S91200-8R
  - Documentation

Remote site: /Storage Card

- Application Data
- My Documents
- NandFlash
- Network
- Program Files
- Storage Card
- Temp

Filename	Filesize	Filetype	Last modified
..			
BOOT.bin	23,040	BIN File	11/17/2010 9:38:12...
EBOOT.bin	524,288	BIN File	11/17/2010 9:38:12...
EBOOT.nb0	524,288	NB0 File	11/17/2010 9:33:52...
FIRSTBOOT.nb0	23,040	NB0 File	11/17/2010 9:33:52...
flash-wince.tcl	1,333	TCL File	10/6/2010 8:45:36 ...
NAND Program....	741	Windows Batc...	10/6/2010 9:11:42 ...
NK.bin	24,950,763	BIN File	11/17/2010 9:38:30...
NK.nb0	29,360,128	NB0 File	11/17/2010 9:34:10...

Selected 4 files. Total size: 25,498,344 bytes

Filename	Filesize	Filetype	Last modified	Permissions	Ov
..					
BOOT.BIN	23,040	BIN File	11/17/2010 9:3...		
BOOT.bin	23,040	BIN File			
EBOOT....	524,288	BIN File	11/17/2010 9:3...		
NK.bin	24,950,763	BIN File	11/17/2010 9:3...		
Version.txt	253	Text Docu...	11/17/2010 9:4...		

5 files. Total size: 25,521,384 bytes

Server/Local file	Direction	Remote file	Size	Priority	Status
192.168.42.253					
C:\Program Files\Sealevel ...	-->>	/Storage Card/NK.bin	24,950,763	Normal	Transferring
00:00:05 elapsed	00:01:31 left	5.5%	1,376,256 bytes (262.1 KB/s)		

Figure 41-*Uploading boot files*

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 Debug Port section.

Once the NAND Flash has been erased, use a standard USB device cable, and connect the Type B connector to the SBC-R9. Connect Type A connector into the host PC. *(See Figure 42.)*

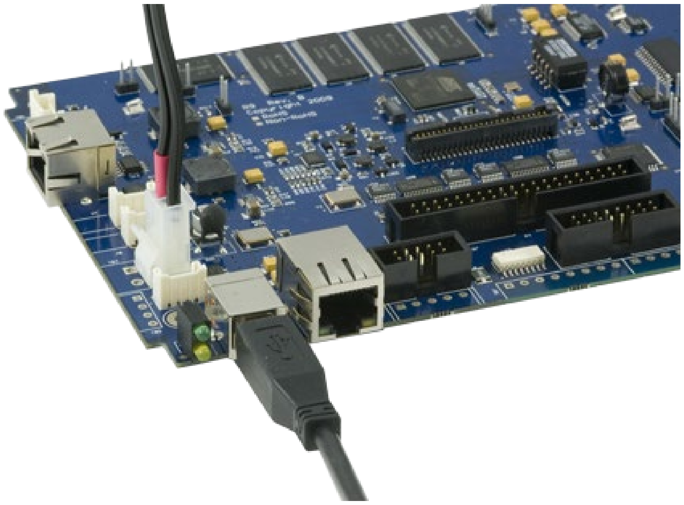
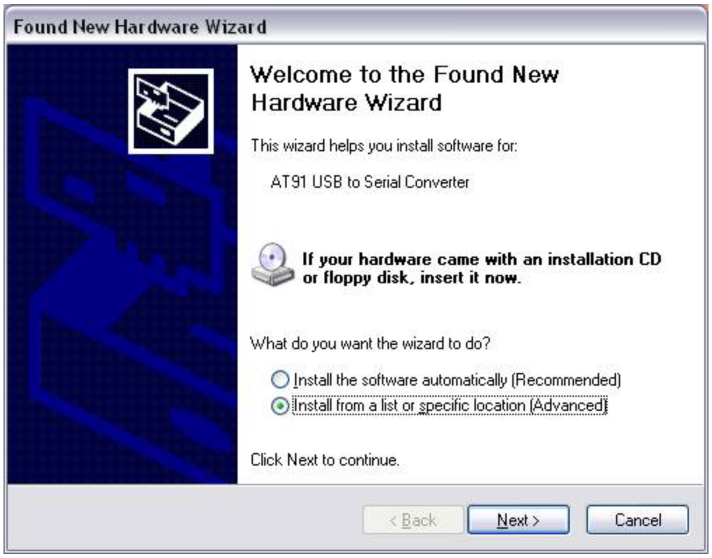


Figure 42-*TR123 tinned leads and Type B USB connector*

In Microsoft Windows 7, the device is recognized as a GPS camera and will typically enumerate as a COM port. Check the device manager to determine the COM# associated with the device. If prompted with the New Hardware wizard or the device is not recognized, then install the driver using the following steps (XP menus shown, but Vista is similar). In the Found New Hardware Wizard, specify "Install from a list or specific location" and click Next. (See Figure 43.)



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install software for:

AT91 USB to Serial Converter

If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

○ Install the software automatically (Recommended)

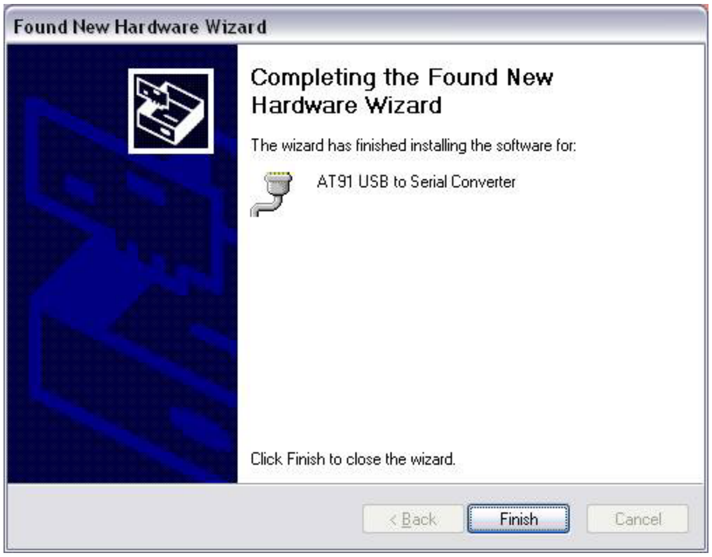
● Install from a list or specific location (Advanced)

Click Next to continue.

Figure 43 - 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 browse 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 44.)



Completing the Found New Hardware Wizard

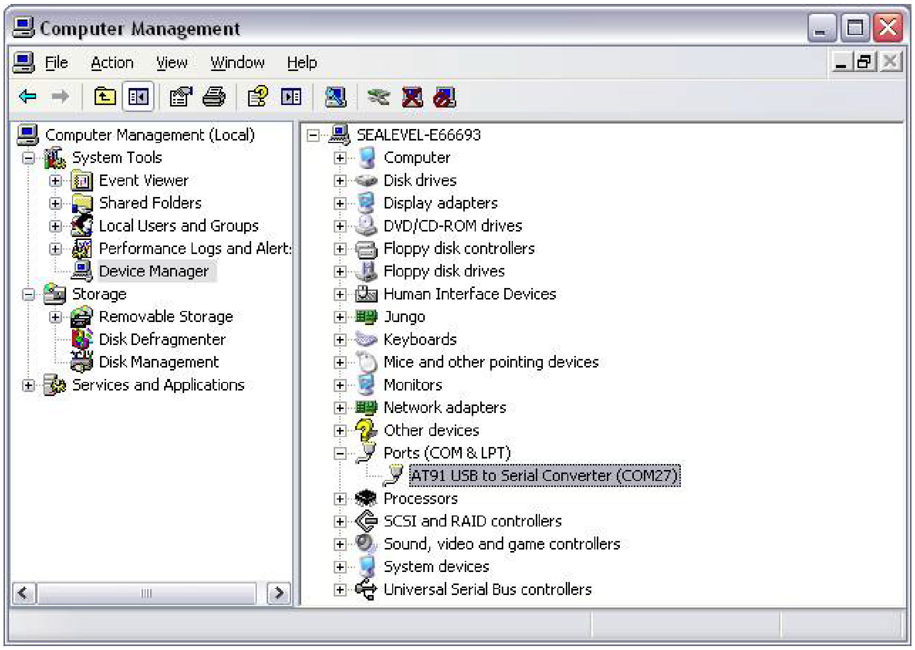
The wizard has finished installing the software for:

AT91 USB to Serial Converter

Click Finish to close the wizard.

Figure 44- *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 45.)

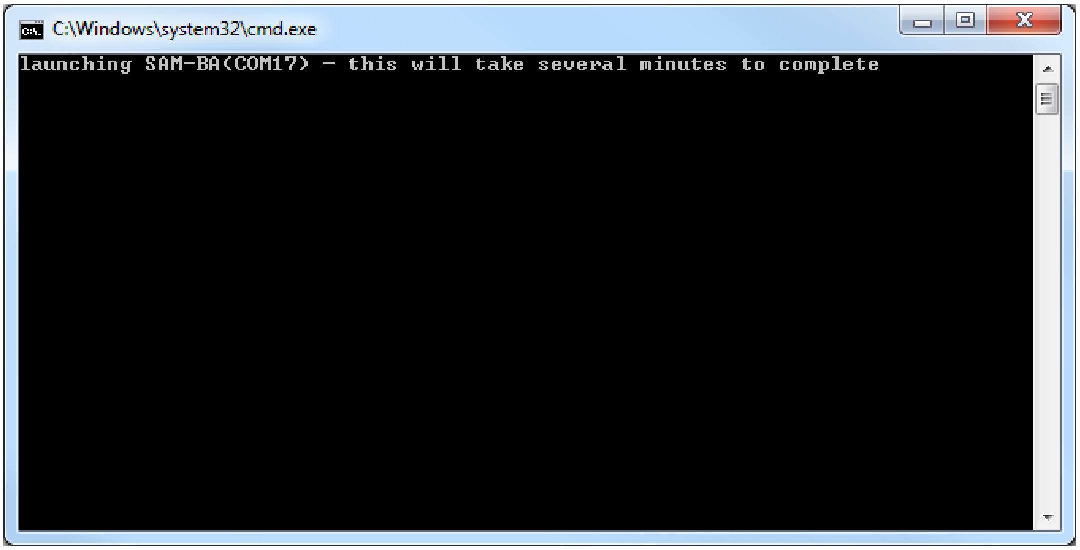


Computer Management (Local)
- System Tools
  - Event Viewer
  - Shared Folders
  - Local Users and Groups
  - Performance Logs and Alert:
  - Device Manager
- Storage
  - Removable Storage
  - Disk Defragmenter
  - Disk Management
- Services and Applications

SEALEVEL-E66693
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Floppy disk controllers
- Floppy disk drives
- Human Interface Devices
- Jungo
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Other devices
- Ports (COM & LPT)
  - AT91 USB to Serial Converter (COM27)
- Processors
- SCSI and RAID controllers
- Sound, video and game controllers
- System devices
- Universal Serial Bus controllers

Figure 45- AT91 COM Port

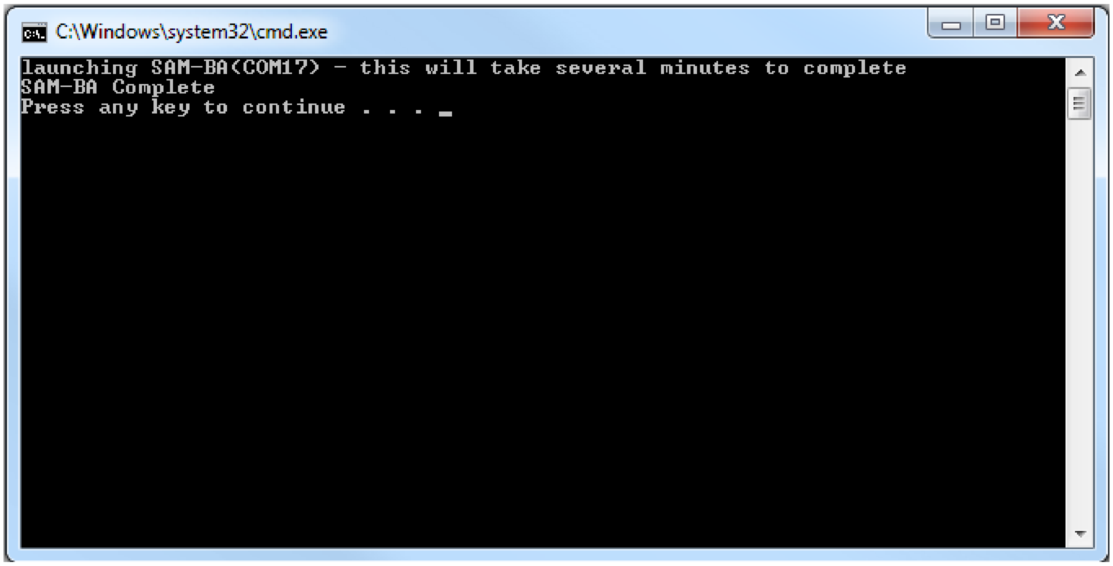
Sample scripts have been provided in the R9 Development installation to automate the process of writing a complete OS runtime to the device. The script is configured to target a device attached to COM49 by default. This can be modified simply by editing the comport variable in the "NAND Program.bat" 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 46).



```
launching SAM-BA(COM17) - this will take several minutes to complete
```

Figure 46 - Programming NAND (COM17)

Once programming has completed, cycle device power and the OS runtime should boot. (See Figure 47.)



```
launching SAM-BA(COM17) - this will take several minutes to complete
SAM-BA Complete
Press any key to continue . . . _
```

Figure 47*-Programming complete*

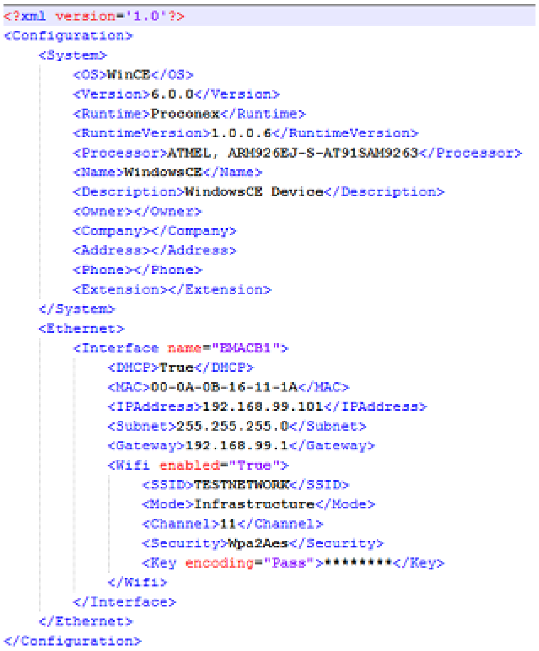
As previously mentioned, the process of programming the NAND Flash first erases all content from the NAND Flash. This includes the unique MAC address assigned to your device at the factory. The "finalize.exe" tool is provided in the "Boot Files" directory of the R9 Development installation. Finalize 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 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 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 10 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 a child of the Configuration element. 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. This element must be a child of the System element. 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 element must be a child of the System element. This version is associated with the Operating System element.

<RUNTIME> - READONLY [STRING]

This element contains a string representation of the specific OS Runtime Image. This element must be a child of the System element.

<RUNTIMEVERSION> - READONLY [VERSION]

This element contains a dot-notation version string. This element must be a child of the System element. This version is associated with the OS Runtime Image.

<PROCESSOR> - READONLY [STRING]

This element contains a Processor Identification string. This element must be a child of the System element.

<NAME> - READ/WRITE [STRING]

This element may contain the device name string. This element must be a child of the System element. This identifier is used as the WinCE host name.

<DESCRIPTION> - READ/WRITE [STRING]

This element may contain the device description string. This element must be a child of the System element. 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. This element must be a child of the System element.

<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. This element must be a child of the System element.

<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. This element must be a child of the System element.

<PHONE> - READ/WRITE [STRING]

This element may contain a string representation of a telephone contact number for the device Owner. This element must be a child of the System element.

<EXTENSION> - READ/WRITE [STRING]

This element may contain a string representation of a telephone extension for the device Owner. This element must be a child of the System element.

<ETHERNET> - STRUCTURE

The Ethernet element contains a list of Ethernet interfaces available to the device. This element must be a child of the Configuration element.

<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". This element must be a child of the Ethernet element. The name attribute is read-only 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. This element must be a child of the Ethernet element. Valid values are True or False.

<*MAC>* - READONLY [MACADDRESS]

This element contains a dash delimited string containing the unique MAC address of this interface. This element must be a child of the Ethernet element. 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. This element must be a child of the Ethernet 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. This element must be a child of the Ethernet 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. This element must be a child of the Ethernet 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. This element must be a child of the Ethernet element. 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. This element must be a child of the Wifi element.

<Mode> - Read/Write [string: Adhoc, Infrastructure]

This element contains the overall Wireless configuration mode. This element must be a child of the Wifi element.

<Channel> - Read/Write [int: 1,11]

This element contains the wireless channel offset to use in Adhoc mode. This element must be a child of the Wifi element.

<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. This element must be a child of the Wifi element.

<Key encoding=""> Writeonly [string] (Attribute [string: Hex, Ascii, Pass])

This key is used to set the wireless connection passphrase or value. This element must be a child of the Wifi element. 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.

<SEALEVEL> - STRUCTURE

The Sealevel element contains a list of Sealevel internal configuration parameters used for Sealevel supplied software plug-ins. This element must be a child of the Configuration element. The plug-in application should contain documentation for the configuration parameters used by that plug-in.

<USER> - STRUCTURE

The User element can be used to contain a list of user configurable parameters for use in custom software. This element must be a child of the Configuration element. Any elements stored under this element will be automatically persisted to the registry key HKLM/Software/User. They can be accessed through that key at any time by custom software.

<ACTION> - WRITEONLY [STRING]

This element may be used to trigger predetermined device behavior. This element must be a child of the Configuration element. 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
7.3”	4.9”	0.75”

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

.

Power Consumption

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

Connector	P3
Manufacturer	Molex
Part Number	09-65-2028
Description	Locking Header, 2 pos, vertical, 3.96mm pitch
Mates with	Molex 09-50-1021

Environmental Specifications

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

Manufacturing

All Sealevel Systems 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.
http://it-ebooks.info/book/1461/

Programming Windows Embedded CE 6.0 Developer Reference, Microsoft Press, Boling.
https://www.microsoft.com/learning/en-us/book.aspx?id=11064Web Sites

Websites

Atmel SAM-BA In-System Programmer (ISP)
http://www.atmel.com/tools/atmelsam-bain-systemprogrammer.aspx

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

Microsoft Windows Embedded Home Page
http://www.microsoft.com/windowsembedded/en-us/windows-embedded.aspx

Microsoft Windows Embedded CE 6.0 Online Documentation
https://msdn.microsoft.com/en-us/library/ee504812(v=winembedded.60).aspx

Microsoft ActiveSync Download
http://www.microsoft.com/windowsmobile/en-us/help/synchronize/ActiveSync-download.mspx

Microsoft Mobile Device Center 6.1
https://support.microsoft.com/en-us/kb/931937

Microsoft .NET Compact Framework
https://msdn.microsoft.com/en-us/library/ms376787.aspx

PuTTy Telnet/SSH Client Application
http://en.wikipedia.org/wiki/PuTTY

Appendix B – SBC-R9 Internal Connector Reference

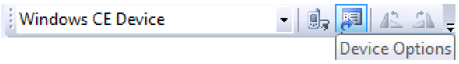
The following table details the connectors, jumpers, and test points located inside the SBC-R9. The connectors, jumpers, and test points are labeled by reference designator on the board silkscreen.

Reference Designator	Signal Description
P1	(8) 12-bit analog inputs
P2	LCD, Backlight, and touchscreen controller
P3	7-30VDC input power
P4	(4) RS-232, RS-422, RS-485 serial ports
P5	(8) Optically isolated inputs (5-24V)
P6	(8) Open-collector digital outputs
P7	(2) 32-bit Quadrature counters
J3	CAN 2.0b Bus interface
J5	RS-485 expansion port (RJ45 connector)
J6	RS-232 serial debug port
J7	USB 2.0 host port A (Type A)
J8	USB 2.0 host port B (Type A)
J10	RS-485 expansion port (Molex connector)
J11	SD/MMC expansion card slot B
J13	USB 2.0 device port (Type B)
J14	10/100 BaseT Ethernet interface
J15	SD/MMC expansion card slot A
Jumpers	
J1	Jumper - NAND Flash write protect
J2	Jumper - NAND Flash enable (Installed by default)
J4	Jumper - CAN Bus Termination (Installed by default)
J16	Jumper - Enable Push-button Reset (Installed by default)
Test Points	
TP1 TP2	Test Point - Analog ground Test Point - Received analog input
TP3	Test Point – Ground
TP4	Test Point – Ground

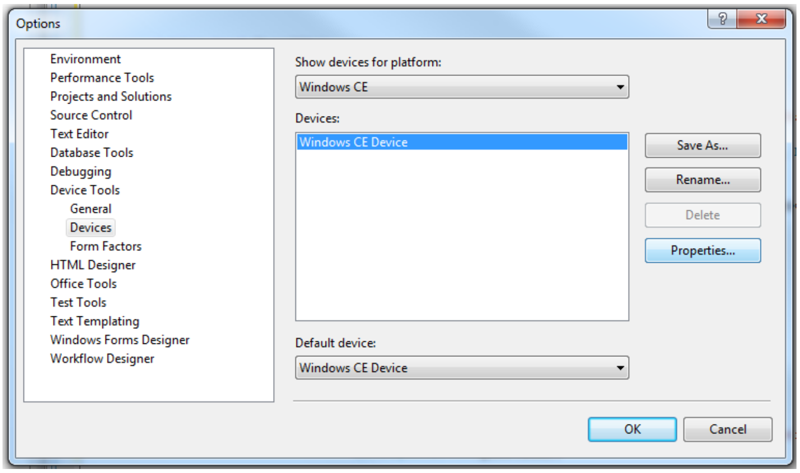
Appendix C – Application Debugging over Ethernet

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.

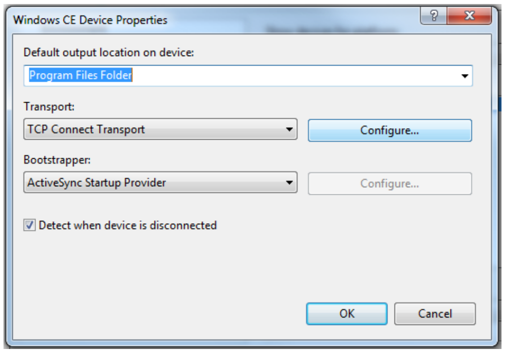


Show devices for platform: Windows CE

Devices: Windows CE Device

Default device: Windows CE Device

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



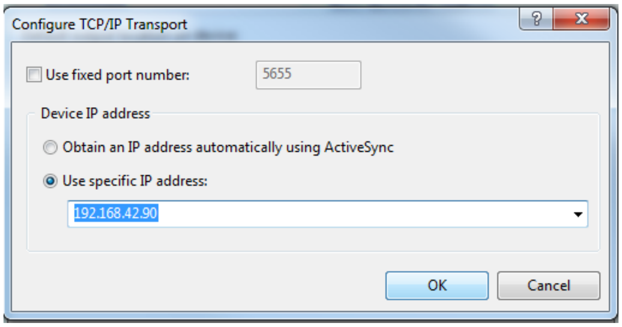
Default output location on device: Program Files Folder

Transport: TCP Connect Transport

Bootstrapper: ActiveSync Startup Provider

☑ Detect when device is disconnected

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


Configure TCP/IP Transport

☐ Use fixed port number: 5655

Device IP address

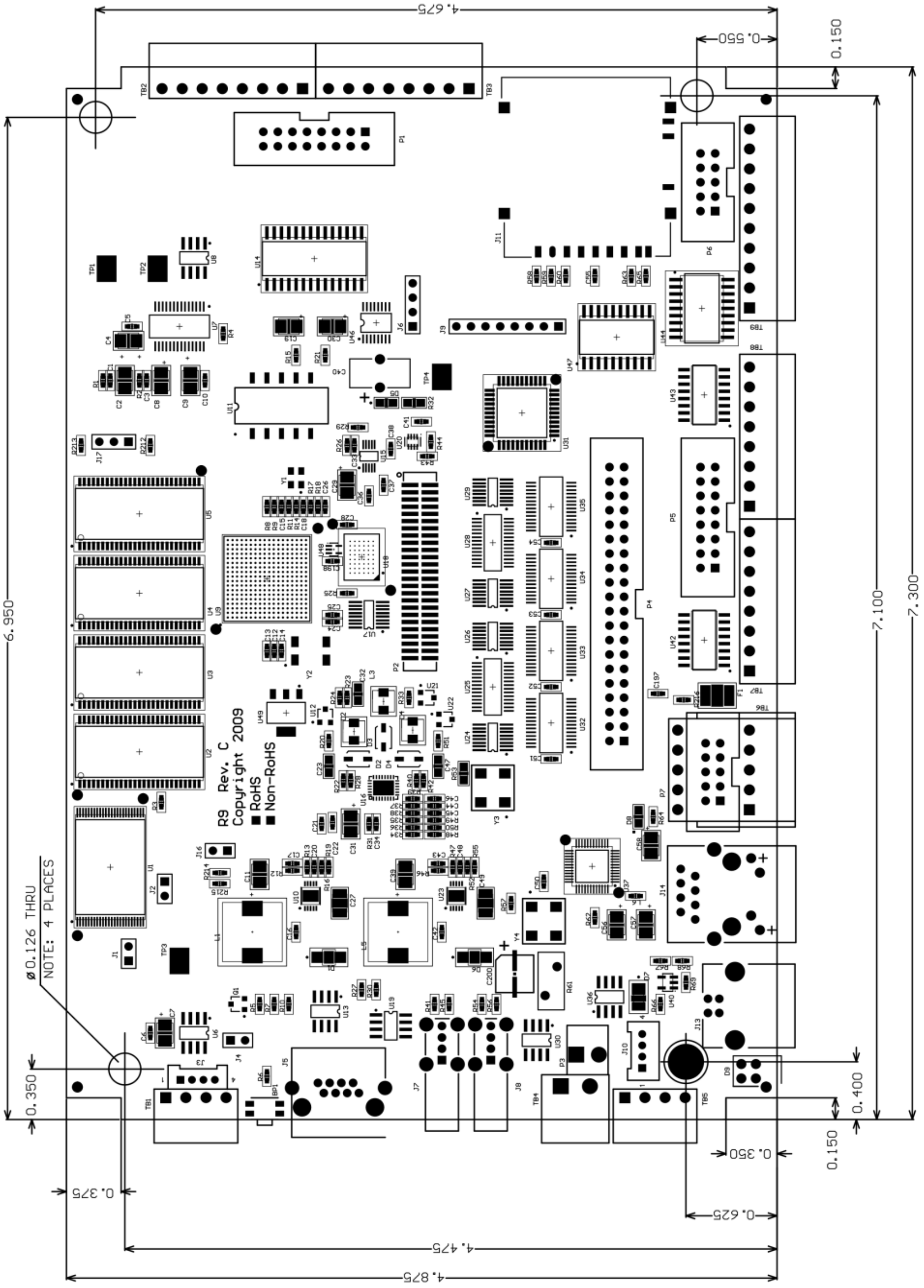
○ Obtain an IP address automatically using ActiveSync

● Use specific IP address:

192.168.42.90

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



Appendix E – How To Get Assistance

Please refer to Troubleshooting Guide prior to calling Technical Support.

1. Begin by reading through the Trouble Shooting Guide in Appendix F. If assistance is still needed, please see below.

2. When calling for technical assistance, please have your user manual and current adapter settings. If possible, please have the adapter installed in a computer ready to run diagnostics.

3. Sealevel Systems provides an FAQ section on its web site. Please refer to this to answer many common questions. This section can be found at http://www.sealevel.com/faq.htm .

4. Sealevel Systems maintains a Home page on the Internet. Our home page address is www.sealevel.com. The latest software updates, and newest manuals are available via our FTP site that can be accessed from our home page.

5. Technical support is available Monday to Friday from 8:00 A.M. to 5:00 P.M. Eastern Time. Technical support can be reached at (864) 843-4343. For email support contact support@sealevel.com.

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

Appendix F – Troubleshooting

The product should provide years of trouble-free service. However, in the event that device appears to not be functioning incorrectly, the following tips can eliminate most common problems without the need to call Technical Support.

1. Make sure the Sealevel Systems product is installed properly.

2. If you are utilizing an operating system prior to Windows 7, please contact Sealevel's Technical support as directed below to receive more information regarding the utility software which will determine if your product is functioning properly.

3. Always use the Sealevel Systems diagnostic software when troubleshooting a problem. This will help eliminate any software issues and identify any hardware conflicts.

If these steps do not solve your problem, please call Sealevel Systems' Technical Support, (864) 843-4343. Our technical support is free and available from 8:00 A.M.- 5:00 P.M. Eastern Time Monday through Friday. For email support contact support@sealevel.com.

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. W 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-9001:2015 certification in 2018.

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 ensure 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.