

# Software Manual

## Advantech SOM-A200 series Software Manual

Released Version : V1.00

Released Date : May, 19, 2004

---

---

**ABSTRACT**

This software manual details the Windows® CE.NET operating system on the SOM-A series platform. Sections include Introduction、Windows® CE Startup Procedure、Upgrade Procedure、Utilities、Network、M-System Persistent Storage Manger、Application Program Development and Windows® CE.NET 4.2 Require Components.

---

---

**Copyright**

This document is copyrighted, © 2003. All rights are reserved. The original manufacturer reserves the right to make improvements to the products described in this document at any time without notice.

No part of this document may be reproduced, copied, translated or transmitted in any form or by any means without the prior written permission of the original manufacturer. Information provided in this document is intended to be accurate and reliable. However, the original manufacturer assumes no responsibility for its use, nor for any infringements upon the rights of third parties that may result from such use.

**Acknowledgements**

IBM, PC/AT, PS/2 and VGA are trademarks of International Business Machines Corporation.

Intel® is trademark of Intel Corporation.

Microsoft® Windows® CE.NET is a registered trademark of Microsoft Corp.

All other product names or trademarks are properties of their respective owners.

For more information on this and other Advantech products please visit our website at :

<http://www.advantech.com>

<http://www.advantech.com/risc>

For technical support and service for please visit our support website at :

<http://eservice.advantech.com.tw/eservice/>

Or directly mail to Advantech RISC platform application engineer :

[AE.RISC@advantech.com.tw](mailto:AE.RISC@advantech.com.tw)

Advantech RISC SOM design-in member can login-in the Advantech SOM Design-in Zone for professional & real-time technical support & service :

<http://risc-designin.advantech.com.tw>

PS. The RISC design-in zone web portal is only for SOM design-in member only.



### Revision History

Version	Date	Reason
V1.00	2004.05.19	Formal released.

# 1

## Software Functionality

This chapter details the Windows®

CE.NET operating system on the

SOM-A series platform.

Sections include:

- Introduction
- Windows® CE Startup Procedure
- Upgrade Procedure
- Utilities
- Network
- M-System Persistent Storage Manger
- Application Program Development
- Windows® CE.NET 4.2 Require Components

## **1.1 Introduction**

The SOM-A series is a modular board, which integrated main component chip including CPU, VGA, Audio, Flash and DRAM chip. The operating system is Windows® CE.NET embedded. The Windows® CE.NET is a compact OS that occupies less storage space or system resources compared with other operating systems such as Windows® NT or Windows® XP. By its modular nature, it is possible to choose those functions that are useful for specific application. Not only reducing the system resources required, but also reduces start-up time. In the field of embedded applications, this is an appealing feature because the impact of downtime would be minimized.

Furthermore, the small storage space it needs makes OS on solid-state disk possible, which implies higher robustness to harsh environments.

SOM series products are flexible for customer design. It is due to public for customer to design their own CARRIER board, they can design their necessary peripherals. Thus, customer also need take more care to make their specific Windows CE image, the unused driver could be removed by Platform Builder.

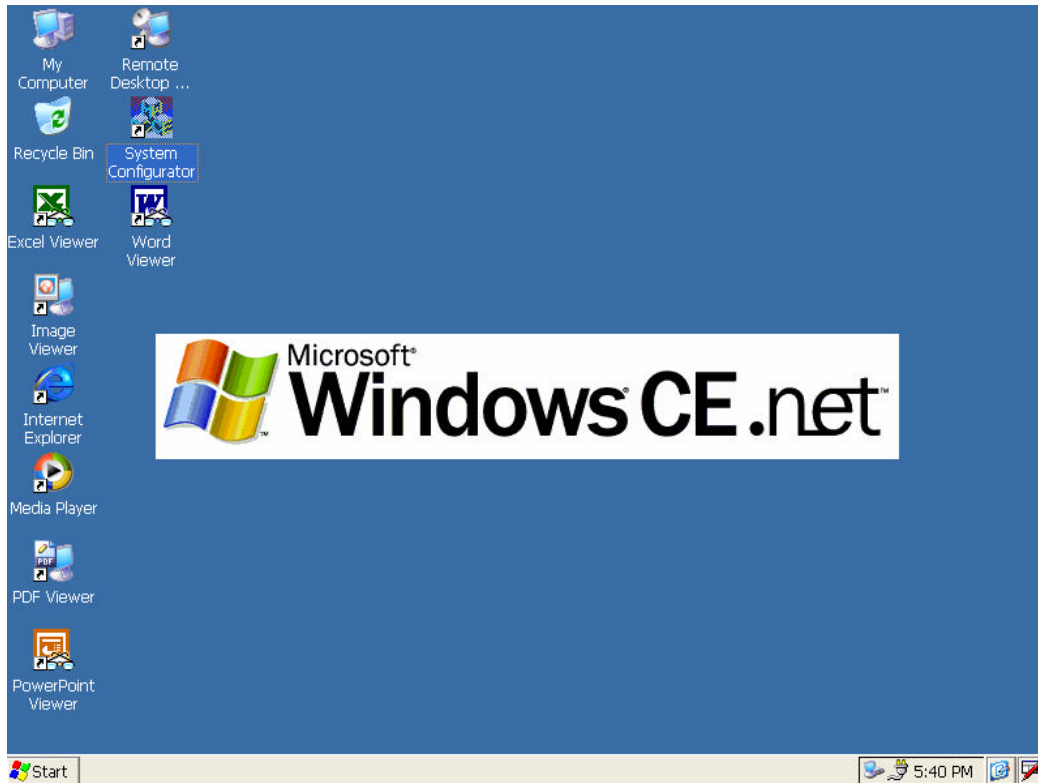


Figure 1.1 Windows® CE.NET on the SOM-A Series

## **1.2 Windows CE Startup Procedure**

Windows CE can be loaded by two methods, first is by Compact Flash card, and second is by system flash chip. Compact Flash card is high boot priority than system flash chip. In BOOTLOADER criteria, it will first read the Windows CE image in Compact Flash card. If Compact Flash card is unavailable or no Windows CE image inside, BOOTLOADER will load the Windows CE image from system flash chip. BOOTLOADER copy Windows CE image to DRAM and launch WinCE from DRAM, whenever loaded by Compact Flash card or system flash chip.

Therefore, there are some advantages,

- 1, Easy Windows CE image installation: Customer can copy Windows CE image they want to Compact Flash card, and then load this image by BOOTLOADER.
- 2, Flexible Hardware design : Flash on board design is unnecessary. The Windows CE can be loaded by Compact Flash card.

## **1.3 Upgrade Procedure**

After the OS image was built, we may want to burn it to the onboard flash ROM. Advantech provides the upgrade utility “Upgrade” to upgrade bootloader image, WinCE image or boot logo to onboard flash ROM. The upgrade procedure is described as following :

Step1. Copy “Upgrade” utility and image files you want ( for example : nk.nb0, eboot.nb0, and window sce.bmp ) to CF storage card.

Step2. Insert CF storage card to SOM-A platform, and then launch Upgrade.exe.

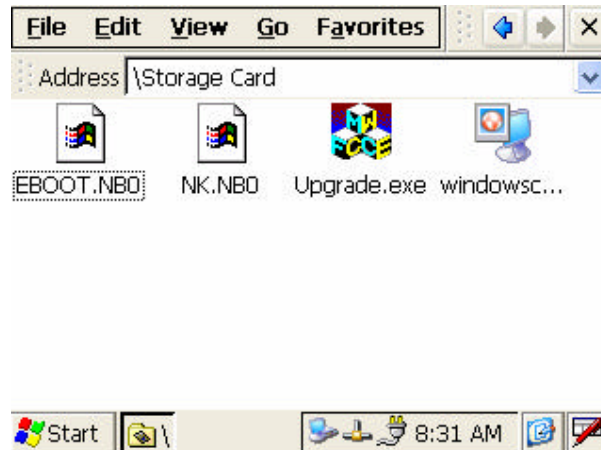


Figure 1.2 Image files and upgrade utility in CF storage card

Step3. Check the items you want to upgrade as the figure shown below. If you want to upgrade boot logo, you can key in the path of the bitmap file in the edit box or click 'Browse' button to select the file.

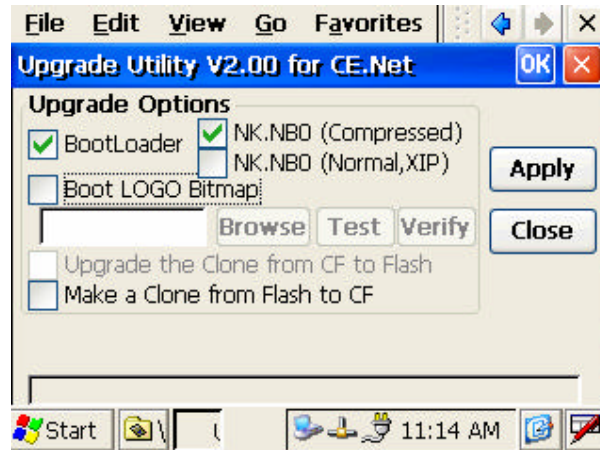


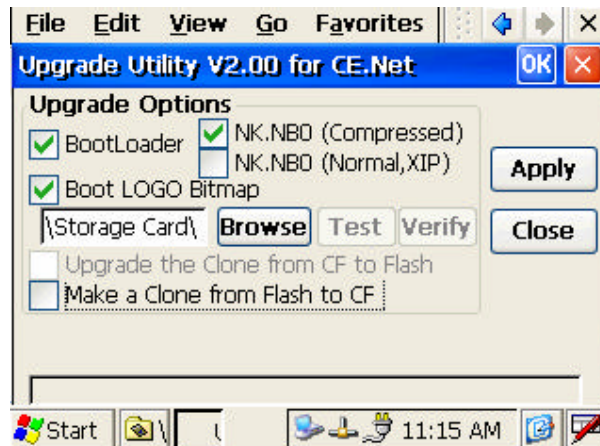
Figure 1.3 Upgrade utility for burn image to flash ROM

*Note:* The difference between NK.NB0 (Compressed) and NK.NB0 (Normal, XIP) :

The option “NK.NB0 (Normal, XIP)” means that the nk.nb0 will be upgraded directly to the flash ROM, and “NK.NB0 (Compressed)” means that we compress nk.nb0 first, and then write the compressed data to the flash ROM. As a result of the different upgrade type, there are two differences when we use SOM-A :

- (1) Bootup time : compressed image take more time in system bootup.
- (2) DiskOnChip size : compressed OS image would result in larger DiskOnChip size.

Step4. Press 'Apply' button on the dialog. Then the items you select will be upgraded to the flash ROM. See Figure 1-4.



#### Figure 1.4 Press Apply button in order to upgrade onboard flash ROM

After the upgrade process is finished, please power off platform and power on it. On the first time bootup, it will take about 1 minute to format the DiskOnChip if your OS image includes DiskOnChip driver.

### **1.4 Utilities**

There are several useful utilities added in the standard Windows® CE.NET OS:

#### 1.4.1 Test Utility

The utility “Auto4\_V1.17.exe” is one integrated test tool, which includes the function validation for peripheral. You can use this tool to verify whether the peripheral function work or not. You just copy this utility to CF and then launch from WinCE by double click it.

#### **Main Function :**

Item	Function Description
COMx (LOOPBACK)	Lookback test for COM1 ~ COM9
COMx <=> COMy	Two COM ports RS232 transmission
TOUCH SCREEN	Touch Screen function validation
PS2 Keyboard	PS2 Keyboard function validation
PS2 Mouse	PS2 Mouse function validation
Audio Play	Audio play function validation
Audio Record	Audio record function validation
USB Keyboard	USB Keyboard function validation
USB Mouse	USB Mouse function validation
WATCHDOG Timer	Watchdog timer validation
Backlight Test	Backlight function validation
A-Sync	Launch ActiveSync
WRITE EVA SOC-M	Write MAC ID
Power Properties	Get power information
DIO Control	DIO function validation
HARDDISK R/W	CF Flash read and write test.
PAUSE	Pause test process
REPEAT TEST	Repeat test process
Etc .....	



**Test Procedure :**

You should see a lot of test items in left zone after launch this utility. You can insert enough test items you want to right zone by pressing "Add" button. The test items in right zone will be executed. Also, you can remove test items from right zone by pressing "Remove" button.

After you add test items done, you can go function test by pressing "EXE ALL" button. To press "View Rpt" button, you will see the test result.

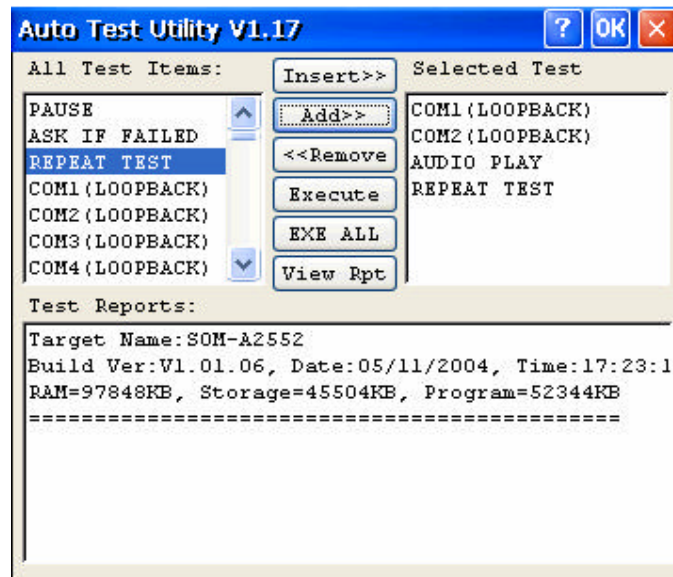


Figure 1.5 Test Utility

**1.4.2 Regflash**

The utility "Regflash" is a convenient tool to save, overwrite or delete registry data, as well as erase the content of DiskOnChip folder. From the Windows® CE.NET status bar, tap "Start/Run". Use the soft-keyboard to type "regflash" command in the command text box and press "OK". There are four selections here:

Save to Flash, Delete from Flash, Save to CF Disk, Erase IPSM.

Choose the options you want and press "Save" button to proceed. "Save to Flash" option was used to save the registry setting to on board flash ROM. In contrast, "Delete from Flash" option was intended to erase the on board registry data. "Save to CF Disk" option would save the registry data to CompactFlash card as a file "wince.reg". "Erase IPSM" option erase the DiskOnChip region of the on board flash.

*Note: Please be careful using this utility "Regflash". This utility is able to overwrite all registry data.*

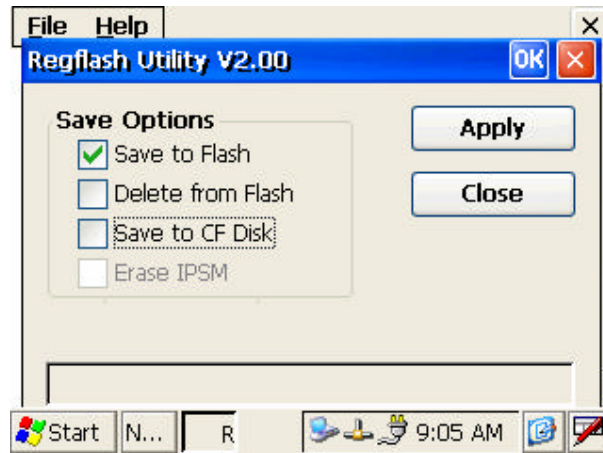


Figure 1.6 Regflash utility

It is important to keep the power normal during "Save to Flash" process. If the power were broken down during the registry saving process, then the registry would be lost and corrupted. On the next time you turn on platform, the system would load the default registry setting rather than the previously customized registry setting.

#### 1.4.3 Reboot

The utility "Reboot" is a convenient tool to reset the system. From the Windows® CE.NET status bar, tap "Start/Run". Use the soft-keyboard to type "reboot" command in the command text box and press "OK". The other ways to reboot are clicking the "Reboot" button on the Watchdog page of the built-in utility System Configurator and hardware reboot button.

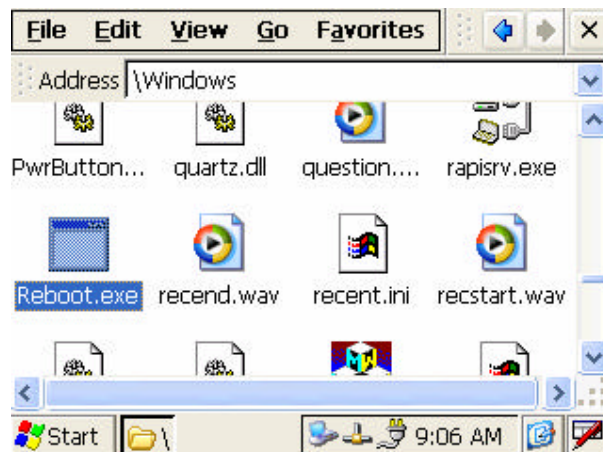


Figure 1.7 Reboot the system

#### 1.4.4 Startup execution

The SOM-A has a useful function call "Startup execution". After the system boot up, the startup execution function would automatically perform. This function is useful for control system to do the initialization processes or some other procedures. In SOM-A, there are two ways to perform "Starup" function.

##### Method 1:

Step1: Create "startup" directory in CompactFlash card or in "\\DiskOnChip\\".

Step2: Copy executable files to "startup" directory that is created by Step 1.

##### Example:

We copy two executable files "REGFLASH.exe" and "Notepad.exe" in "\\DiskOnChip\\Startup", and then reboot the system. After the system boot up, the two executable file would automatically execute.

##### Method 2:

Step1: The same as Step1 in Method 1.

Step2: Create a file called "startup.ini" in "startup" directory. Type in the commands you want to execute after boot up in that file.

##### Example:

Create "Startup.ini" in "\\DiskOnChip\\Startup" directory and reboot the system. The content of startup.ini was listed below:

\\windows\\tty.exe

\\windows\\registry.exe

After the system reboot, "\\windows\\ tty.exe" and "\\windows\\ registry.exe" would automatically execute. Be sure that the two methods are independent. It means they can be used simultaneously.

#### 1.4.5 Safemode

SOM-A allow user to alter registry setting, and save it by either "regflash.exe" or the registry frame of the "Misc" page of the System Configurator. But sometimes user may make some non-appropriate registry setting, and cause SOM-A fail to boot. In the circumstance, the easiest way to boot up SOM-A is to use the default registry setting from the Windows® CE.NET image. When the SOM-A is booted up with the default registry setting, we say that it is working in "safemode". To enter "safemode", user must perform several steps as described below:

Step 1: Create a file whose filename is "safemode" or a folder whose name is "safemode" in the CompactFlash card.

Step 2: Insert the CompactFlash card into the SOM-A series platform.

Step 3: Turn on the power switch of SOM-A series platform.

#### 1.4.6 System Configurator

System Configurator is an outstanding utility designed by Advantech Windows® CE.NET software team. It is an integrated environment where user can get useful system information as well as configure favorite system settings and apply system control function on demand. Double click the icon of System Configurator on the desktop to open the SOM-A utility. Following sections illustrate the functions of System Configurator.

##### *1.4.6.1 General*

The memory information including DRAM, CF Disk and DiskOnChip FLASH are displayed in the General page. And the versions of each part of the installed embedded OS, including Windows® CE.NET, Registry, Bootloader and System Configurator respectively.

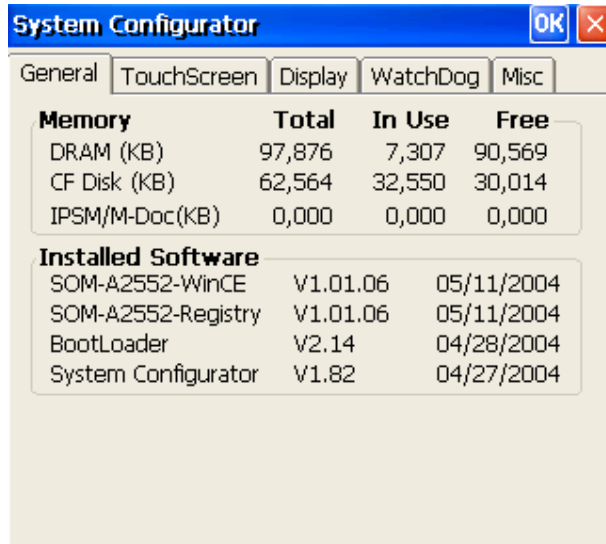


Figure 1.8 General information

#### 1.4.6.2 Touch-screen

The Touch-screen page provides the calibration function. Click the "calibration" button, the "Stylus Properties" windows would appear. Then click "calibrate" button in the Stylus Properties window to enter calibration process. In the calibration process, user taps on the center of the target on the screen then the target will move to the next position. After calibration, press "OK" to leave Stylus Properties window, and then the Regflash utility process would automatically start to save the registry setting.



Figure 1.9 Touch-screen calibration

### 1.4.6.3 Display

From time to time it is unnecessary to turn on the display attached to the SOM-A all the day. The Display page provides several frequently used functions such as turning off the LCD and backlight to elongate the display repair period, adjusting brightness or contrast. For example, if user wants the backlight turn-off setting function, he can press "setting" button. Then the backlight page of Display Properties of Control Panel will appear on the screen. Besides, user can click the "Off Now" button to turn off the backlight of the display panel immediately without waiting. Once the backlight was turned off, there were three inputs to turn it on: (1) mouse; (2) keyboard; (3) touch-screen; user can use any one of them to turn on the display. The lower "Brightness" and "Contrast" blocks have scroll bars by which users can tune brightness level of TFT LCD or the contrast level of passive matrix LCD.

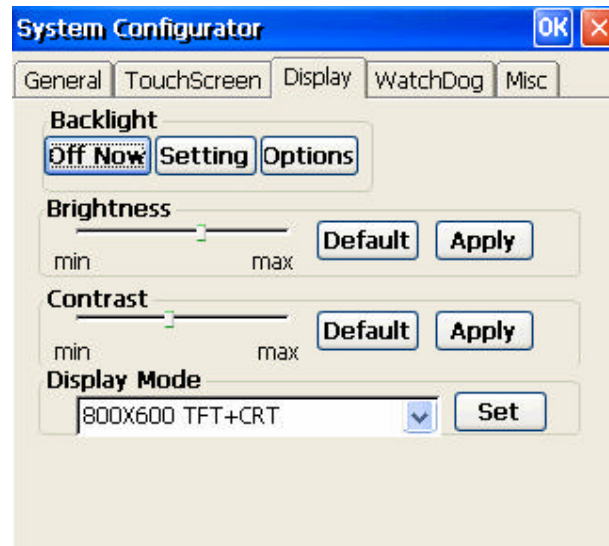


Figure 1.10 Display configuration

### 1.4.6.4 WatchDog timer

It is important in industrial applications that the control systems are rarely crashed, or are capable of self-reset if they are halted somehow. Watchdog function of automatic resetting system is therefore provided in SOM-A. There is a timer inside the watchdog function. User's AP could invoke the associated APIs in Watchdog function to start the timer, then Watchdog function would repeat the countdown of the specified period of time to reboot the system if the user's AP does not clear the timer in time periodically. The Watchdog function in the SOM-A provides eight different time

intervals: 2 seconds, 5 seconds, 10 seconds, 30 seconds, 60 seconds, 2 minutes, 5 minutes and 10 minutes. The “Enable” button is used to simulate the Watchdog function. Detail programming guide is illustrated at section 1.7.5. The “Sleep” button could make the system enter suspend mode as “Suspend” of “Start” could. Press the “SoftReset” button will cause system warm boot that clears DRAM, reloads all drivers and refresh the newest registry settings. Press the “REBOOT” button will cause the system cold boot.

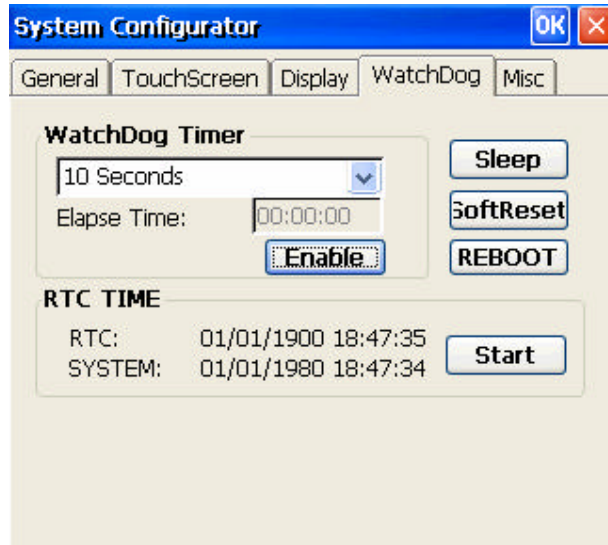


Figure 1.11 Watchdog timer

#### 1.4.6.5 Miscellaneous

The Misc page provides several functions as described below. The "Registry" block provides registry save and registry view function. The “A. Sync” button invokes ActiveSync to the host computer. The "HTTP Server Root" block was used to specify the root directory of http server. The default directory is "\windows\www\wwwpub", user can specify another directory by type the directory in the edit box and press "Set" button. The new setting would become effective after the system reboot. The “CF Disk Folder Name” block specifies the folder name of the storage card inserted. The default name is “Storage Card”. User can specify another

directory by type the directory in the edit box and press "Set" button. The new setting would become effective after the system reboot. The "MAC ID" block shows the network MAC address. The "COMM" block provides the communication functions, including IPConfig and Pinging Yahoo.

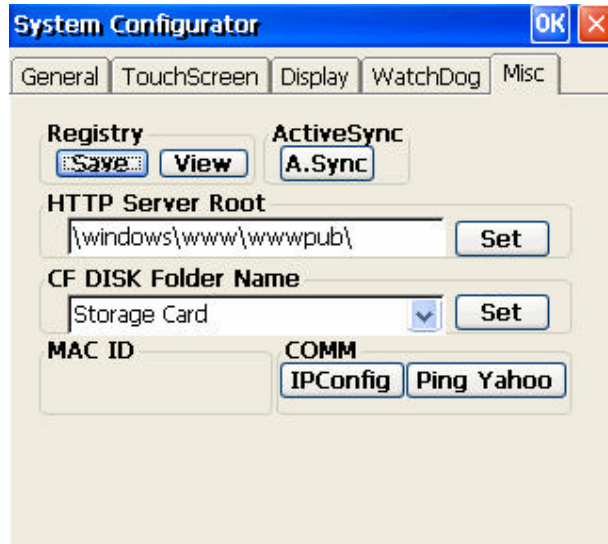


Figure 1.12 Miscellaneous settings



## **1.5 Network**

### **1.5.1 Networking via Ethernet**

SOM-A2558 and SOM255F both build in one 100Base-T Ethernet controller. It appears at “Control Panel/Network and Dial-up Connections” via “FMAC1”. User can configure its Ethernet support as follows:

1. Click "Start/Settings/Control Panel"
2. Double click "Network and Dial-up Connections"
3. This window will display all available connections. Pressing the connection icon, its pop-up menu appears and users could disable, rename or modify properties from there.
4. If the SOM-A2558 & F is a node of the LAN with DHCP servers, it is now available.
5. If the SOM-A2558 & F is a node of the LAN with fixed IP, the user has to consult with MIS to get specific IP addresses. Then fill them into the associated fields of the Properties Dialog that could be popped up by the properties item of the step 3 above. Then use the "Regflash" utility to save this changed registry. Reboot the system, the Ethernet functions would be available as previous configuration.

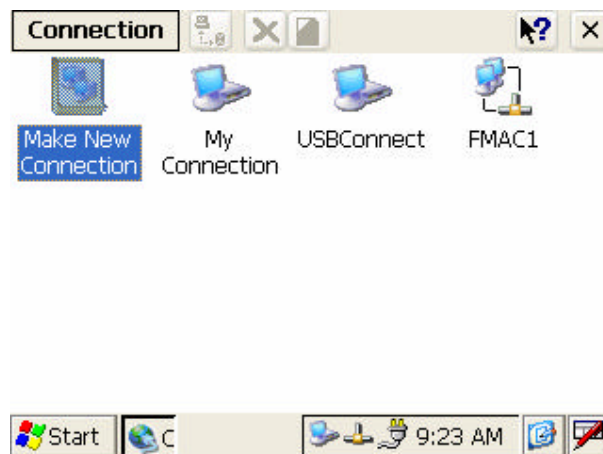


Figure 1.13 Networking via Ethernet

### **1.5.2 Networking via USB port**

The SOM-A supports USB port direct-connections to host computer. The host

computer must install the Microsoft ActiveSync service offered by Microsoft. Use the USB cable to connect the USB ports of them. Then activate ActiveSync service on the host computer. The host will automatically scan the USB ports to make a connection.

1. Make sure the Microsoft ActiveSync service and the Microsoft embedded Visual Tools are properly installed in the host PC.
2. Connect the USB port of the host PC and the SOM-A by a USB ActiveSync cable.
3. If users are using the Microsoft eMbedded Visual Tools to develop Windows® CE.NET application runtimes, make sure the SOM-A SDK provided in the SOM-A support CD is also properly installed in the host PC.
4. Click "Start/Settings/Network and Dial-up Connections"
5. Make a new connection. As the dialogue box pops out, choose the default "Direct Connection" radial button. Click "Next".
6. Select "USB Cable" from the combo box and click "Finish" to complete making new connection. It is recommended to keep the default settings of the ports connection.
7. Click "PC Connection" icon in the Control Panel. As the "PC Connection Properties" dialogue box pops up, change the connection to the newly made connection by clicking the "Change..." button.
8. If the ActiveSync service on the host PC has been activated, the above seven steps will make the SOM-A automatically try to connect the host; otherwise you can invoke "\\windows\\repllog.exe" to do the activesync connection.

*NOTE: Users should properly install the associated USB driver on the host computer while plugging in the SOM-A as a USB client device at the first time.*

*NOTE: The USB driver--wceusbsh.inf and wceusbsh.sys--are included in SOM-A support CD.*

*NOTE: Users may also use COM ports to do ActiveSync function thru RS-232 cable.*

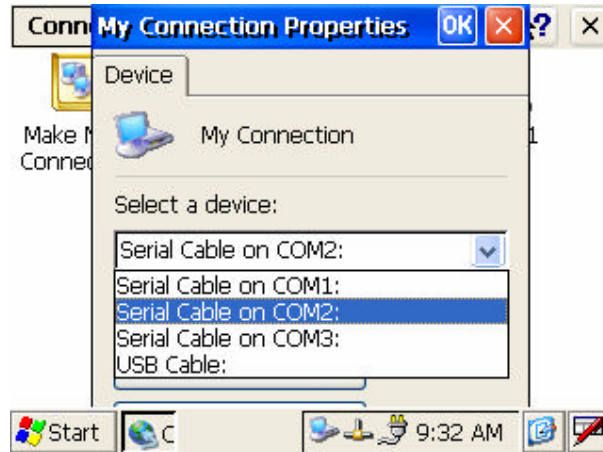


Figure 1.14 Networking via USB port

### 1.5.3 Networking via PPP

The SOM-A supports PPP protocol. To setup and utilize it, follow the steps below:

1. Click "Start/Settings/Network and Dial-up Connections"
2. Make a new connection. As the dialogue box pops out, choose the "Dial-Up Connection". Click "Next".
3. Click "Configure" to setup the device according to the specification of your modem, and then click "OK" on the top-right corner of the window.
4. Click "Next". Input the telephone number in the "Phone Number" window. Press "Finish" to complete the setup process.
5. Turn on your modem and use RS-232 cable to connect modem and COM1 of SOM-A series.
6. Double click the connection you have made in Step 4. Key in the user name, password and domain for the dial-up connection and press "Connect".

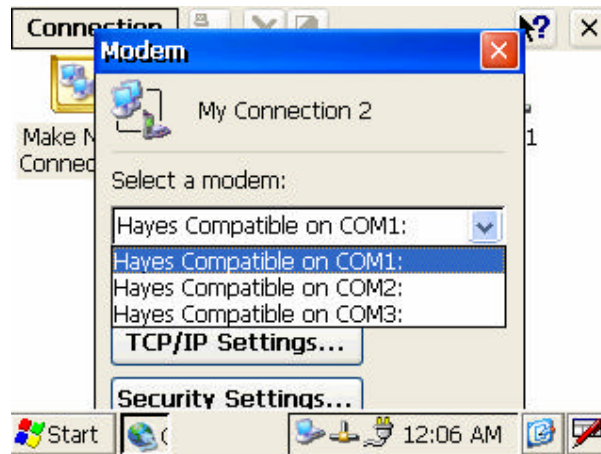


Figure 1.15 Networking via PPP

#### 1.5.4 Web browser

The SOM-A builds-in Windows CE OS includes IE Browser. It can be used to browse web pages on World Wide Web via LAN or PPP.

## **1.6 M-System Persistent Storage Manger (DiskOnChip)**

### **1.6.1 Introduction to M-System Persistent Storage Manger**

M-System Persistent Storage Manager was designed and developed specifically as an enhancement to Microsoft Windows CE operating systems. DiskOnChip eliminates extra disk-like storage such as storage cards, redundant RAM and ROM.

### **1.6.2 DiskOnChip folder in SOM-A Series**

SOM-A uses M-System Persistent Storage Manager to utilize the free space of flash ROM for persistent storage. The DiskOnChip region in the system is located in "\\DiskOnChip" directory. Any file or directory stored in "\\DiskOnChip" directory would be keep persistently, even if the power of SOM-A were turned off. The user can store software or data in \\DiskOnChip rather in CompactFlash card to avoid inconvenience.

## **1.7 Application Program Development**

The SOM-A is bundled with built-in Windows<sup>®</sup> CE.NET operating system. In real applications users need to execute various application programs on it. However, unlike its other family, the Windows<sup>®</sup> CE.NET is a hardware-dependent operating system. That is to say, Windows<sup>®</sup> CE.NET application programs are only portable in the source code level. Users must rebuild the runtime file for a different Windows<sup>®</sup> CE.NET platform even though the source code may not be changed at all.

### **1.7.1 System requirements**

- Intel<sup>®</sup> Pentium-90 CPU or more advanced
- Microsoft<sup>®</sup> Windows<sup>®</sup> 2000 Professional or Windows<sup>®</sup> XP
- Microsoft<sup>®</sup> eMbedded Visual Tools 4.0
- Platform SDK for SOM-A
- 64MB DRAM
- CD-ROM drive
- Monitor with VGA resolution at least
- Mouse
- 200MB free hard disk space at least
- SOM-A series platform
- Let the host PC and SOM-A connect on the same LAN to do kernel debugging if necessary
- USB cable (bundled in the standard SOM-A series)

### **1.7.2 Building Windows CE program**

By the platform SDK bundled with the standard SOM-A, users can build the Windows CE runtime application program by the eMbedded Visual Tools.

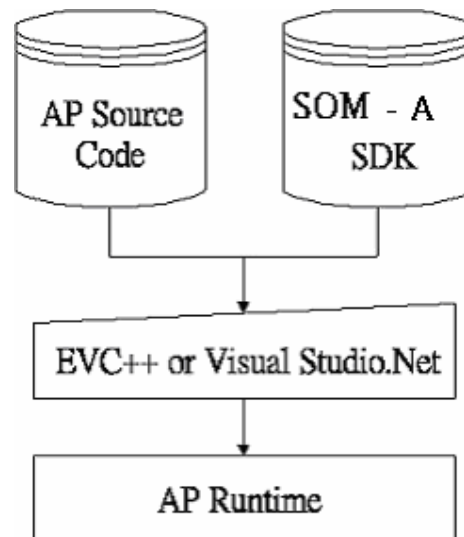


Figure 1.16 Flow-chart of Building Windows® CE.NET runtime

### 1.7.3 How to install SDK

Copy SOM-A SDK file “SOM\_A\_SDK.msi” to your PC, and launch it. You can install SDK by steps.

Step 1, Launch SOM-A SDK file, and then tap Next button.



Figure 1.17

Step 2, Accept License Agreement and go next.

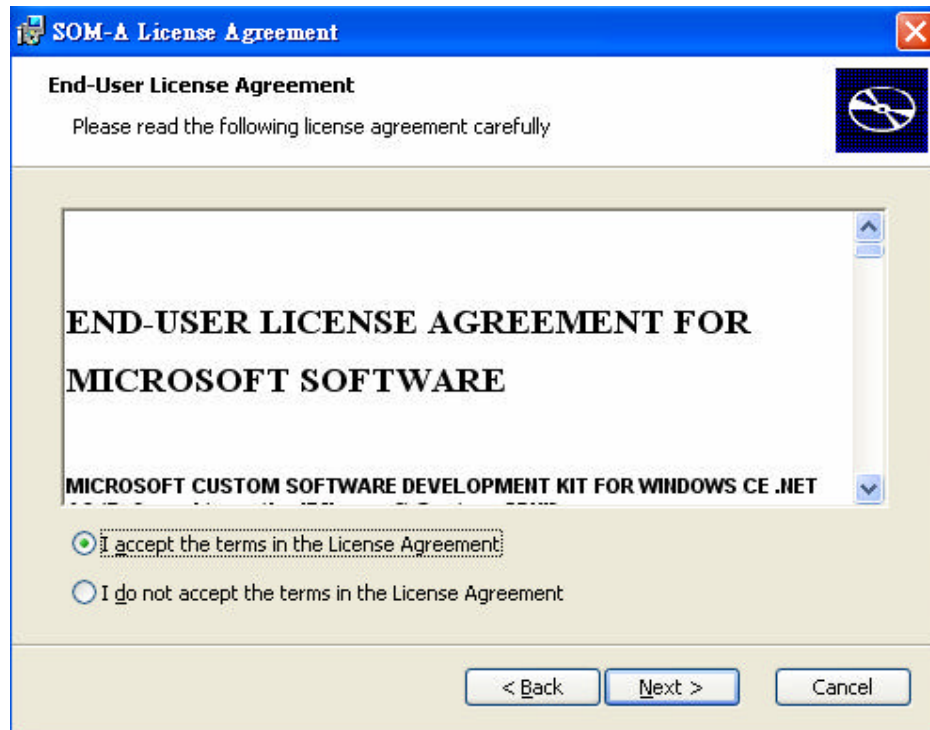
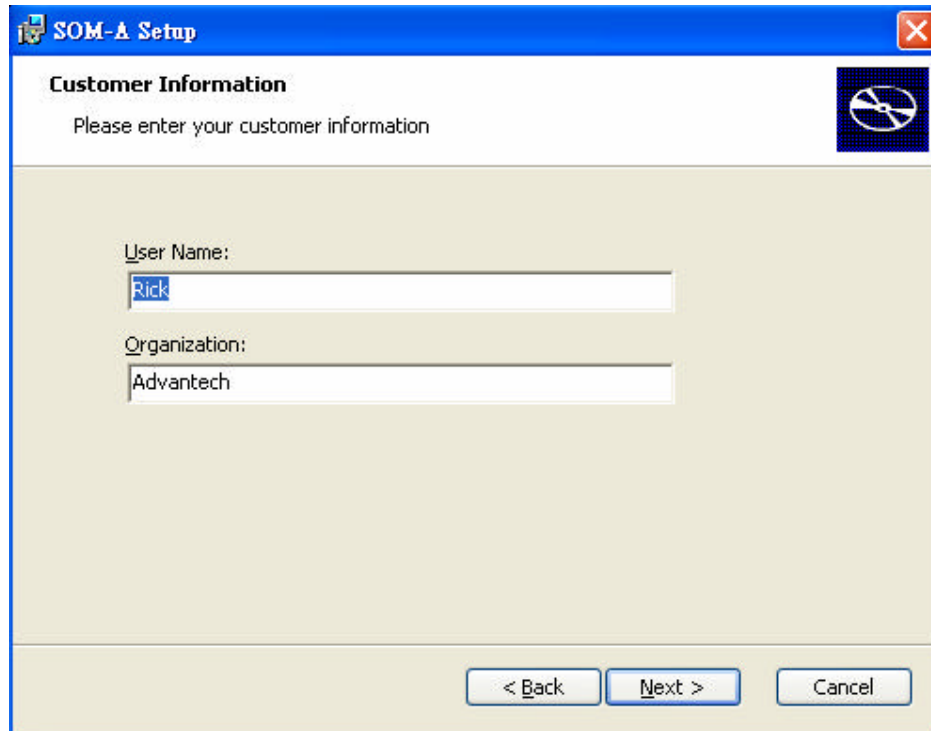


Figure 1.18

Step 3, Key in your information and go next.





The image shows a Windows-style dialog box titled "SOM-A Setup". Inside the dialog, there is a section titled "Customer Information" with the instruction "Please enter your customer information". Below this, there are two text input fields. The first field is labeled "User Name:" and contains the text "Rick". The second field is labeled "Organization:" and contains the text "Advantech". At the bottom of the dialog, there are three buttons: "< Back", "Next >", and "Cancel". A small icon of a crossed-out square is visible in the top right corner of the dialog box.

Figure 1.19

Step 4, Choose setup type.

There are 3 options “Embedded Visual C++”, “Microsoft .NET Compact Framework”, and “Documentation” in Custom Setup.

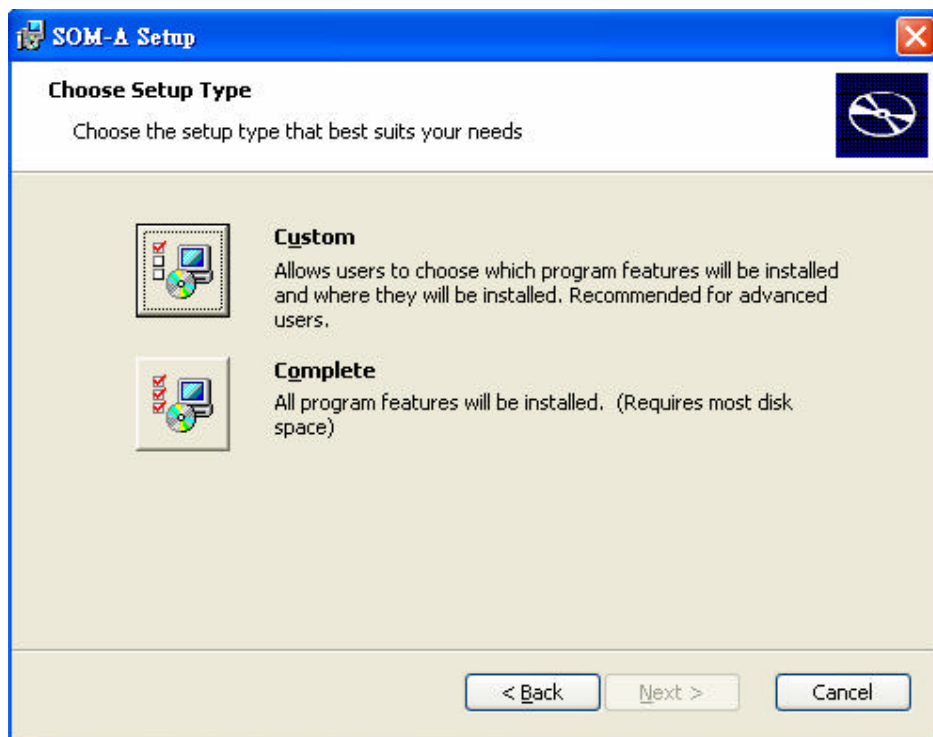


Figure 1.20

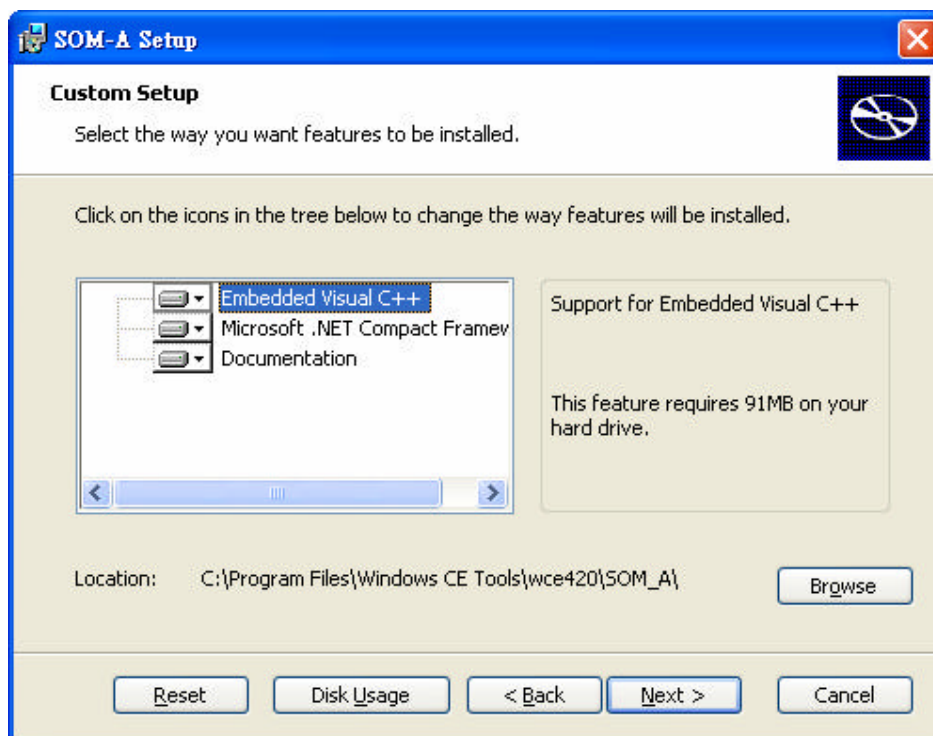


Figure 1.21

Step 5, Tap “Install” button to install SDK.

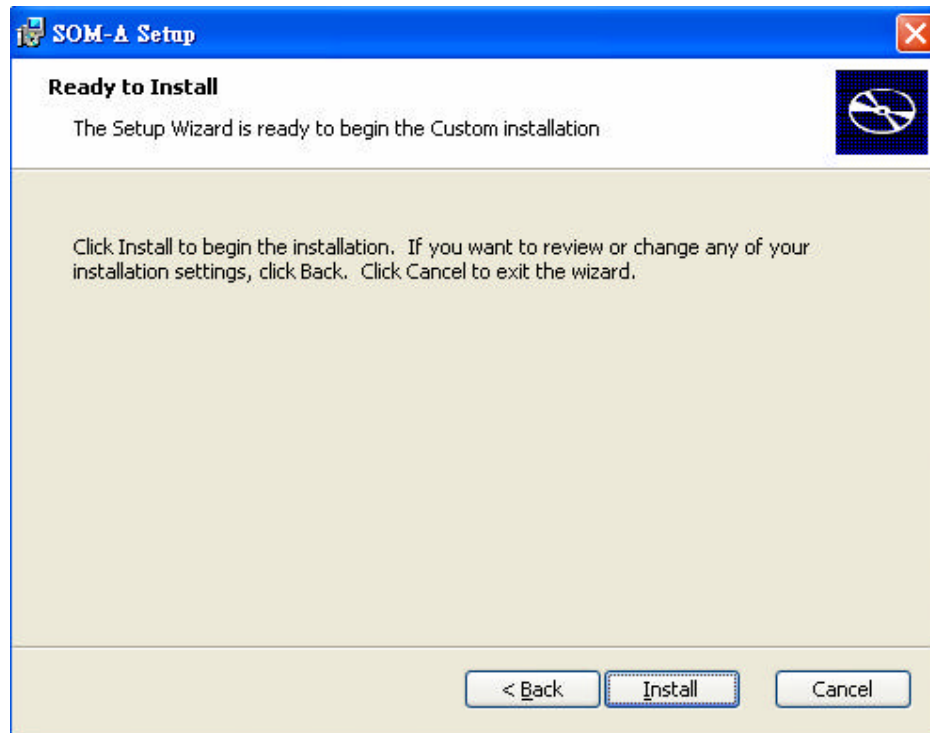


Figure 1.22

Install SDK.....

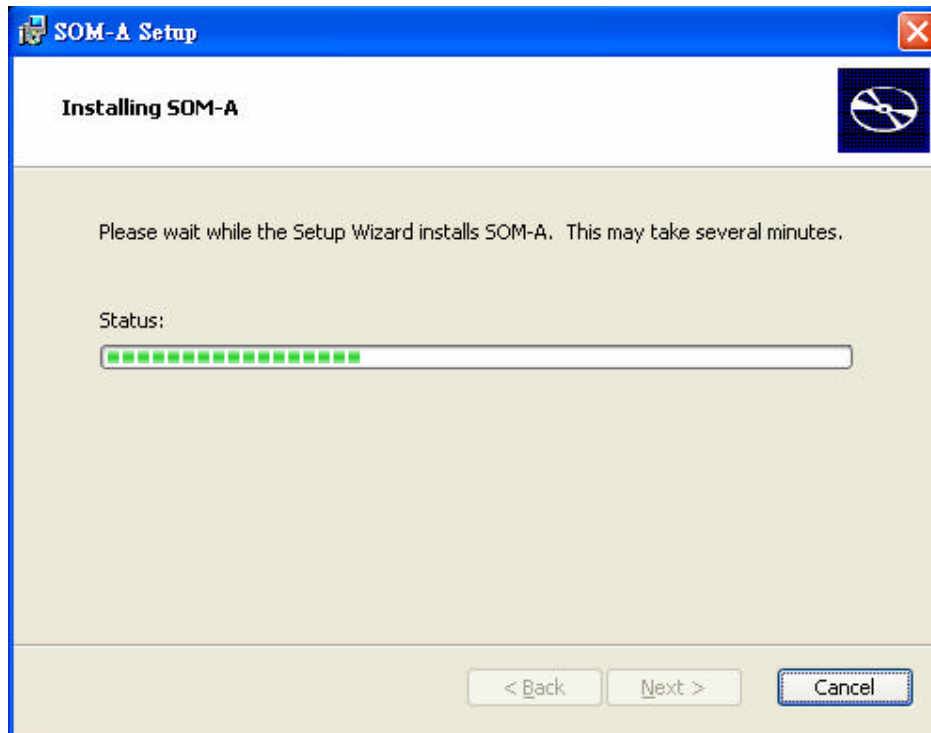


Figure 1.23

Step6, Finish installing.

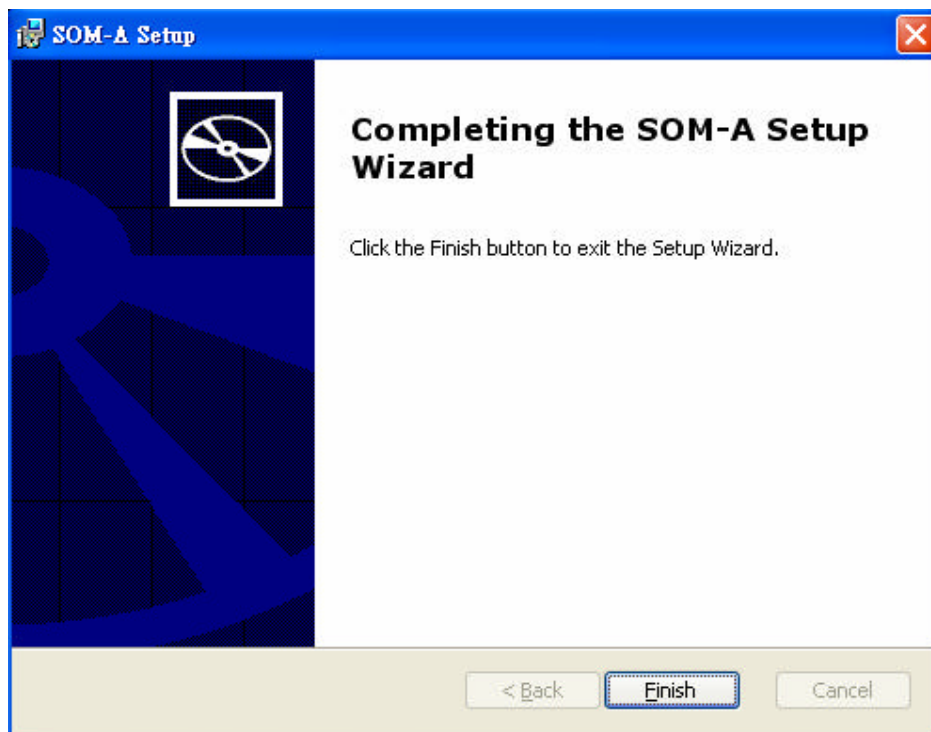


Figure 1.24

#### 1.7.4 Running your application programs

Activesync would automatically transfer the built application program to platform. Choose SDK type as SOM-A once compile your application program.

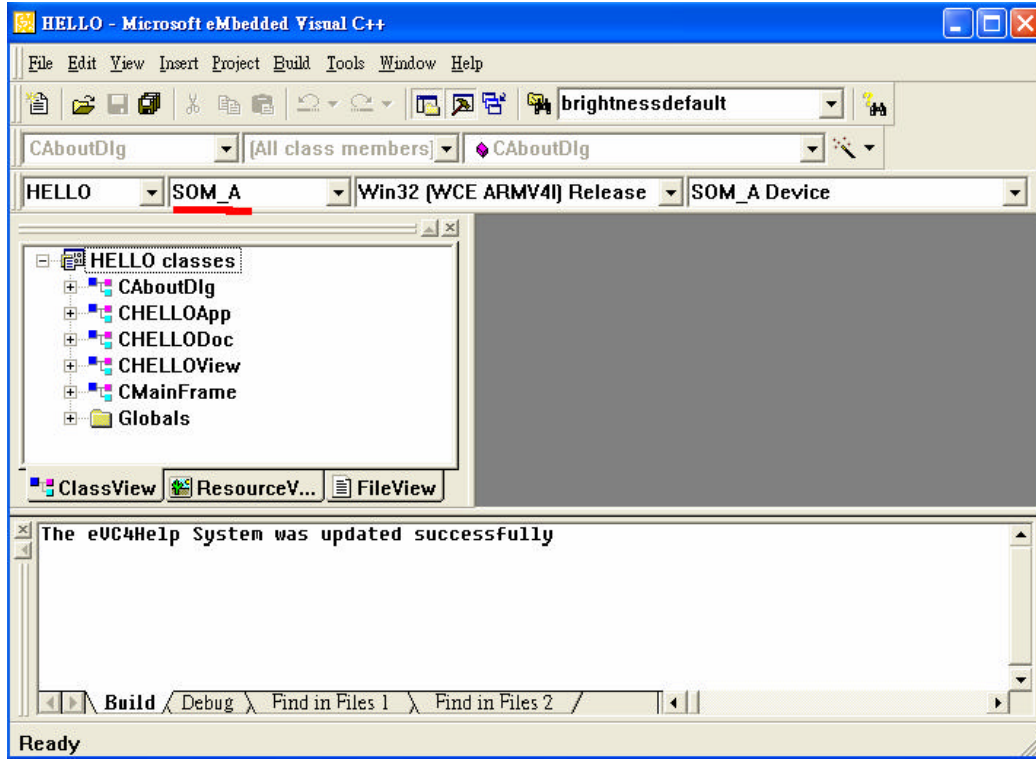


Figure 1.25

#### 1.7.5 WDT Modules

SOM-A is targeted to be the embedded device for web-enabled and data-acquisition systems. It is built-in with a useful dynamic link library, WATCHDOG.DLL, which has been designed to help AP developer easily implement his requirements of handling the system resources such as Watchdog timer and LCD brightness control etc. Programmers who are familiar with WIN32 API programming would feel very easy to use this DLL to create his functions. The DLL make its device services look as a file whose name is "WDT1:". The programmer could use this file name in "CreateFile()" to open it and get the file handler. Then the following controls of user required device services would be achieved by way of the file handler in one function call of "DeviceIOControl()". At most 5 applications can access watchdog timer simultaneously. The DeviceIOControl function is following:

## - DeviceIoControl

This function sends a control code directly to a specified device driver, causing the corresponding device to perform the specified operation.

```
BOOL DeviceIoControl(  
  
HANDLE hDevice,  
  
DWORD dwIoControlCode,  
  
LPVOID lpInBuffer,  
  
DWORD nInBufferSize,  
  
LPVOID lpOutBuffer,  
  
DWORD nOutBufferSize,  
  
LPDWORD lpBytesReturned,  
  
LPOVERLAPPED lpOverlapped);
```

## - Parameters

### *hDevice*

[in] Handle to the device that is to perform the operation. Call the Create- File function to obtain a device handle.

### *dwIoControlCode*

[in] Specifies the control code for the operation. This value identifies the specific operation to be performed and the type of device on which the operation is to be performed. No specific values are defined for the dwIo- ControlCode parameter. However, the writer of a custom device driver can define IOCTL\_XXXX control codes, per the CTL\_CODE macro.

These control codes can then be advertised, and an application can use these control codes with

DeviceIoControl to perform the driver-specific functions.

***lpInBuffer***

[in] Long pointer to a buffer that contains the data required to perform the operation.

This parameter can be NULL if the dwIoControlCode parameter specifies an operation that does not require input data.

***nInBufferSize***

[in] Size, in bytes, of the buffer pointed to by lpInBuffer.

***lpOutBuffer***

[out] Long pointer to a buffer that receives the operation's output data.

This parameter can be NULL if the dwIoControlCode parameter specifies an operation that does not produce output data.

***nOutBufferSize***

[in] Size, in bytes, of the buffer pointed to by lpOutBuffer.

***lpBytesReturned***

[out] Long pointer to a variable that receives the size, in bytes, of the data stored into the buffer pointed to by lpOutBuffer.

The lpBytesReturned parameter cannot be NULL. Even when an operation produces no output data, and lpOutBuffer can be NULL, the Device-IoControl function makes use of the variable pointed to by lpBytesReturned. After such an operation, the value of the variable is without meaning.

***lpOverlapped***

[in] Ignored; set to NULL.

**- Return Values**

Nonzero indicates success. Zero indicates failure. To get extended error information, call GetLastError.

## WDT Control Codes

There are 8 control codes for the operation codes in the WDT1 driver:

### 1. IOCTL\_ENABLE\_WDT (0x1001):

Enables the Watchdog timer on your application. Your application must trigger to Watchdog timer by IOCTL\_ACCESS\_WDT interface during specified period, otherwise the device will reboot automatically

*lpInBuffer* : unused.

*nInBufferSize*: unused.

*lpOutBuffer*: unused.

*nOutBufferSize*: unused.

### 2. IOCTL\_DISABLE\_WDT (0x1002):

Disable the Watchdog time on your application.

*lpInBuffer* : unused.

*nInBufferSize*: unused.

*lpOutBuffer*: unused.

*nOutBufferSize*: unused.

### 3. IOCTL\_GET\_WDTPERIOD (0x1003):

*lpInBuffer* :unused.

*nInBufferSize*: unused.



*lpOutBuffer*: the DWORD pointer to your Watchdog time setting. The unit is mini-second. Its value should be greater 1000. The default setting is 5000 mini-seconds.

*nOutBufferSize*: unused.

#### **4. IOCTL\_SET\_WDTPERIOD (0x1004):**

*lpInBuffer* : the DWORD pointer to your Watchdog time setting. Its value should be greater 1000. The unit is mini-second. If your application opens the WDT driver, the default Watchdog timer is set to 5000 mini-seconds.

*nInBufferSize*: unused.

*lpOutBuffer*: unused.

*nOutBufferSize*: unused.

#### **5. IOCTL\_ACCESS\_WDT (0x1005):**

Your application must trigger the Watchdog once during your Watchdog timer period. If your application has not trigger at the specified period, the device will reboot automatically.

*lpInBuffer* :unused.

*nInBufferSize*: unused.

*lpOutBuffer*: unused.

*nOutBufferSize*: unused.

#### **6. IOCTL\_GET\_SCREENOFFTIME (0x1006):**

*lpInBuffer* :unused.

*nInBufferSize*: unused.

*lpOutBuffer*: the DWORD pointer to your screen off time if user-interface idled. The unit is

mini-second. If the value is 0, screen-off function is disabled.

*nOutBufferSize*: unused.

## 7. IOCTL\_SET\_SCREENOFFTIME (0x1007):

*lpInBuffer* : the DWORD pointer to your screen off time if user-interface idled. The unit is mini-second.

If the value is 0, screen-off function is disabled.

*nInBufferSize*:unused.

*lpOutBuffer*: unused.

*nOutBufferSize*: unused.

## 8. IOCTL\_SET\_SCREENOFF (0x1010):

Set the LCD power off immediately.

*lpInBuffer* : unused.

*nInBufferSize*:.unused.

*lpOutBuffer*: unused.

*nOutBufferSize*: unused.

### Examples:

```
#define IOCTL_ENABLE_WDT      0x1001
```

```
#define IOCTL_DISABLE_WDT    0x1002
```

```
#define IOCTL_GET_WDTPERIOD  0x1003
```

```
#define IOCTL_SET_WDTPERIOD  0x1004
```

```
#define IOCTL_ACCESS_WDT          0x1005

#define IOCTL_GET_SCREENOFFTIME   0x1006

#define IOCTL_SET_SCREENOFFTIME   0x1007

#define IOCTL_SET_SCREENON        0x100F

#define IOCTL_SET_SCREENOFF       0x1010
```

```
HANDLE m_hWDT=NULL;
```

```
TCHAR szClassName[60];
```

```
...
```

```
// assign the WDT driver name
```

```
wsprintf(szClassName, TEXT("WDT1:"));
```

```
// Open the WDT driver
```

```
m_hWDT = CreateFile(szClassName, GENERIC_READ | GENERIC_WRITE, 0,
```

```
NULL, OPEN_EXISTING, FILE_ATTRIBUTE_NORMAL, NULL);
```

```
if ( m_hWDT == INVALID_HANDLE_VALUE )
```

```
{
```

```
    DebugMsg(CString("WDT driver fail"));
```

```
    return;
```

```
}
```

```
...
```

```
DWORD dwTemp;
```

```
DWORD nPeriod=10000;
```

```
// Set the Watchdog Timer as 10 seconds (10000 mini-seconds)
```

```
DeviceIoControl(m_hWDT, IOCTL_SET_WDTPERIOD, &nPeriod, 4, NULL, 0, &dwTemp, NULL);
```

```
// Enable the Watchdog timer
```

```
DeviceIoControl(m_hWDT, IOCTL_ENABLE_WDT, NULL, NULL, NULL, 0, &dwTemp, NULL);
```

```
While (1)
```

```
{
```

```
    // do your job here...
```

```
    Sleep(8000);
```

```
    DeviceIoControl(m_hWDT, IOCTL_ACCESS_WDT, NULL, NULL, NULL, 0, &dwTemp, NULL);
```

```
}
```

```
DeviceIoControl(m_hWDT, IOCTL_DISABLE_WDT, NULL, NULL, NULL, 0, NULL, NULL);
```

```
CloseHandle(m_hWDT);
```

**1.8 Windows® CE.NET 4.2 Require Components***(Advantech Recommend)***Applications and Services Development** ( : with; : without)

<b>Feature</b>	<b>Default Selection</b>
Active Template Library (ATL)	
C Libraries & Runtimes	
Component Services (COM)	
Device Management	
Lightweight Directory Access Protocol (LDAP)	
Message Queuing (MSMQ)	
Microsoft Foundation Classes (MFC)	
Object Exchange Protocol (OBEX)	
Pocket Outlook Object Model (POOM) API	
Simple Object Access Protocol (SOAP) Toolkit	
Standard SDK for Windows CE .NET	
.NET Compact Framework	
XML	

**Applications – End User**

<b>Feature</b>	<b>Default Selection</b>
ActiveSync	
File Viewers	
Help	
Inbox	
Remote Desktop Connection	
Terminal Emulator	
Windows Messenger	
WordPad	

**Core OS Services**

<b>Feature</b>	<b>Default Selection</b>
Serial Port Support	

Parallel Port Support	
USB Host Support	
Debugging Tools	
Power Management	
Kernel Features	

### **Communication Services and Networking**

<b>Feature</b>	<b>Default Selection</b>
Networking Features	
Networking - Local Area Network (LAN)	
Networking - Personal Area Network (PAN)	
Networking - Wide Area Network (WAN)	
Servers (HTTPD)	

### File Systems and Data Store

<b>Feature</b>	<b>Default Selection</b>
Storage Manager	
File & Database Replication (Bit-based)	
File System – Internal (RAM & ROM File System)	
Registry Storage (RAM-based Registry)	

### Fonts

<b>Feature</b>	<b>Default Selection</b>
Arial	
Comic Sans MS	
Courier New	
Georgia	
Impact	
Kino	

MSLogo	
Symbol	
Tahoma	
Times New Roman	
Trebuchet MS	
Verdana	
Webdings	
Wingding	

## International

<b>Feature</b>	<b>Default Selection</b>
Locale Services	
Locale Specific Support (Input Method Selector Sample Application)	
Multilingual User Interface (MUI)	

## Internet Client Services

<b>Feature</b>	<b>Default Selection</b>
Browser Application (Internet Explorer 5.5 for Windows CE - Standard Components)	
Internet Explorer 5.5 for Windows CE Components	
- Internet Explorer Browser Control Host	
- Internet Explorer HTML/DHTML API	
- Internet Explorer Multiple-Language API	

- Internet Explorer TV-Style Navigation	
- URL Moniker Services	
- Windows Internet Services	
Pocket Internet Explorer HTML View (WEBVIEW)	
Sample IE 5.5 Internet Options Control Panel	
Scripting	

## Multimedia Technologies

<b>Feature</b>	<b>Default Selection</b>
Basic Multimedia	
Multimedia Components	
- Audio	
- DirectMusic	
- Digital Rights Management	
- Direct3D	
- DirectDraw	
- DirectShow	
- DVD-Video	
- Windows Media Player	
- Windows Media Technologies	

## Security

<b>Feature</b>	<b>Default Selection</b>
Authentication Services (SSPI)	
Cryptography Services (CryptoAPI 1.0) with High Encryption	



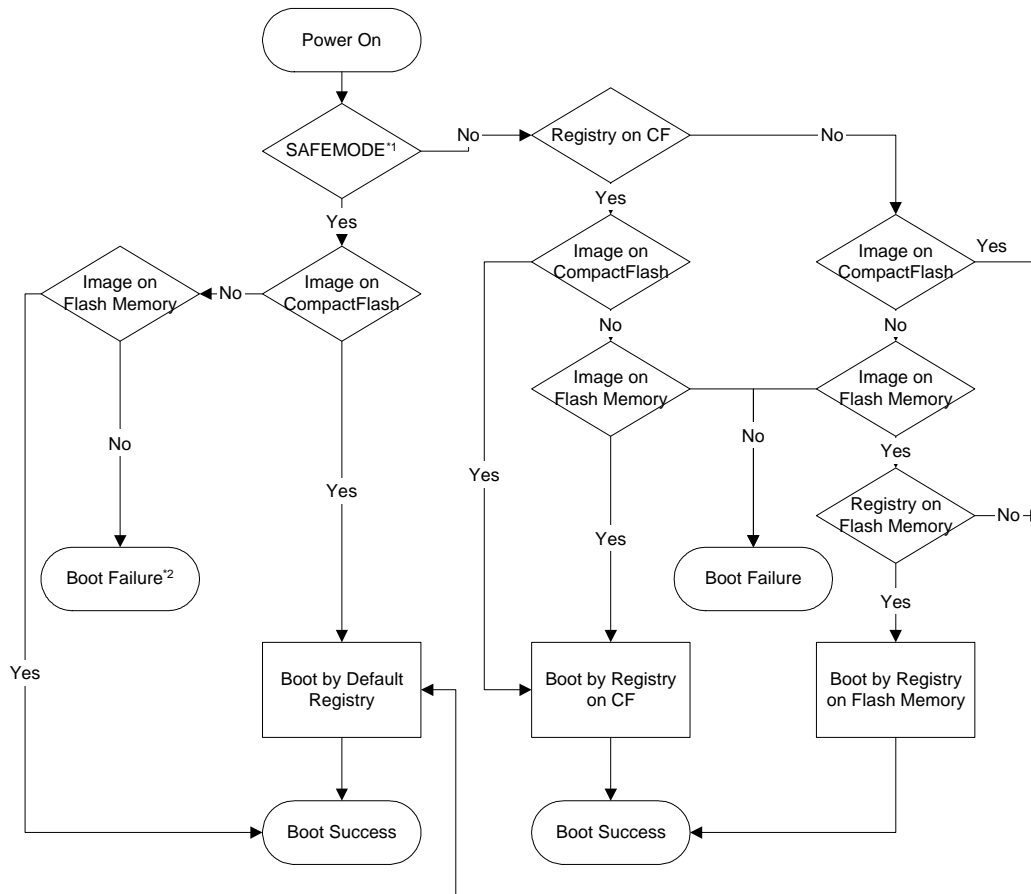
Provider	
----------	--

### **Shell and User Interface**

<b>Feature</b>	<b>Default Selection</b>
Shell	
User Interface	
- Accessibility	
- Customizable UI	
- Mouse	
- Touch Display (Stylus)	
- Network User Interface	
- Overlapping Menus	
- Software Input Panel	
- Speech Interface	

## Appendix A Boot & Registry Searching Sequence

The following diagram demonstrates the boot sequence and registry searching sequence when SOM-A boot-up.



**Figure A: Boot & Registry Searching Sequence**