

TOUCH CONTROLLER

# CUWIN

User Manual



**COMFILE**  
TECHNOLOGY

550 Pilgrim Dr. Suite K  
Foster City, CA 94404  
sales@cubloc.com  
(650)-638-1350

## 1.0 Introduction

---

The CUWIN combines a graphic display and touch interface with a high efficiency industrial controller.

It is equipped with Microsoft Windows CE 5.0 Embedded, a stable operating system used in many industrial touchscreen applications. Windows CE supports applications developed in Microsoft Visual Studio 2005 and other development tools able to create Windows CE ARM executables. It can also run NSBasic scripts.

Developing for the CUWIN is similar to normal PC program development; anyone with previous PC programming experience can develop CUWIN applications without difficulty. The CUWIN supports the .NET Compact Framework (1.0, 2.0) which reduces development and testing time by allowing the use of cross-platform .NET functions.

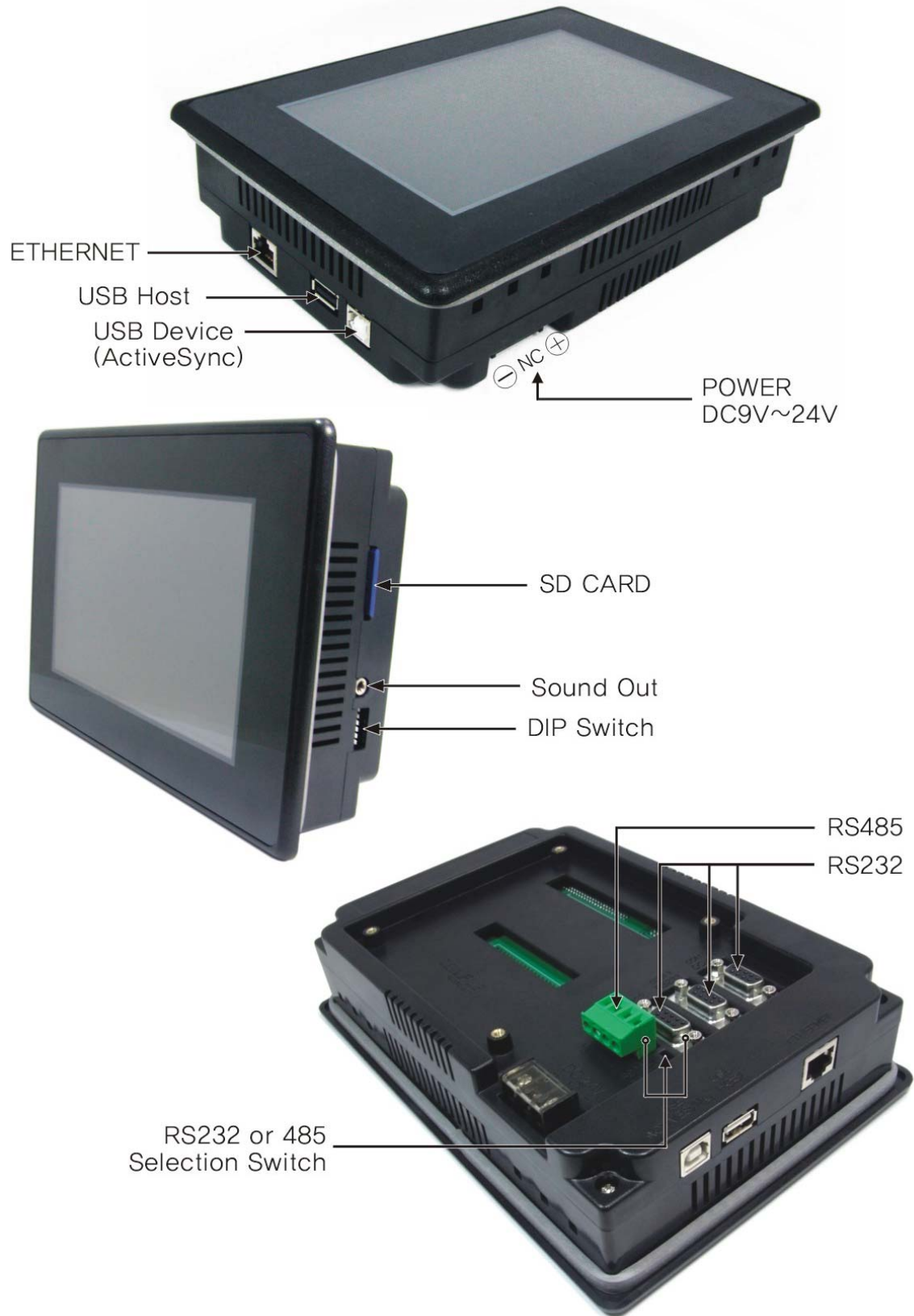
The CUWIN can also operate in IntelliLCD Mode, which allows the CUWIN to emulate an IntelliLCD as a drop-in replacement. Existing IntelliLCD applications can upgrade to a CUWIN today, and not lose functionality while the CUWIN native applications are developed.

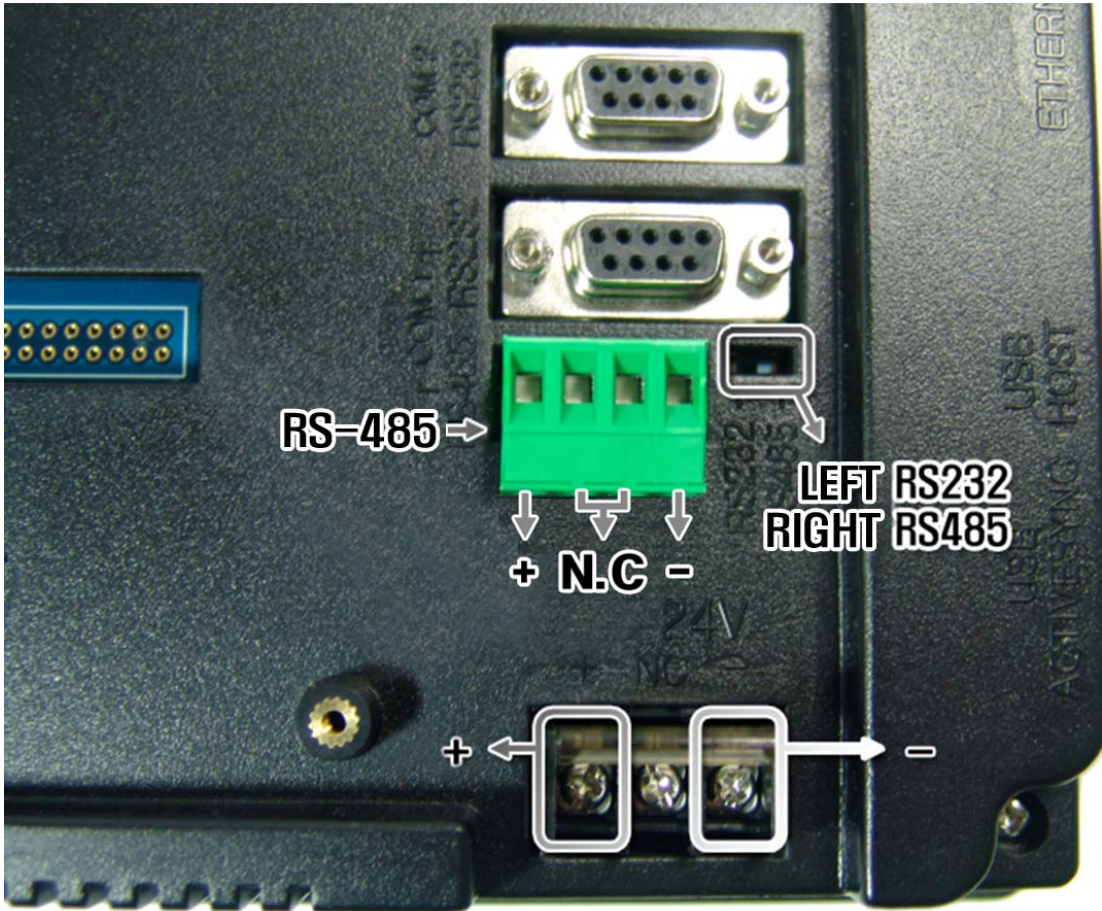
The CUWIN offers RS232, RS485, and Ethernet for external communication. It can even browse web sites using the internal browser, which is useful for displaying diagnostic data or even remote control panels through a web interface.

Depending on which CUWIN features you choose to access, it can be used for anything from a user interface panel to the main controller for your automation equipment.

# 1.1 Overview

---

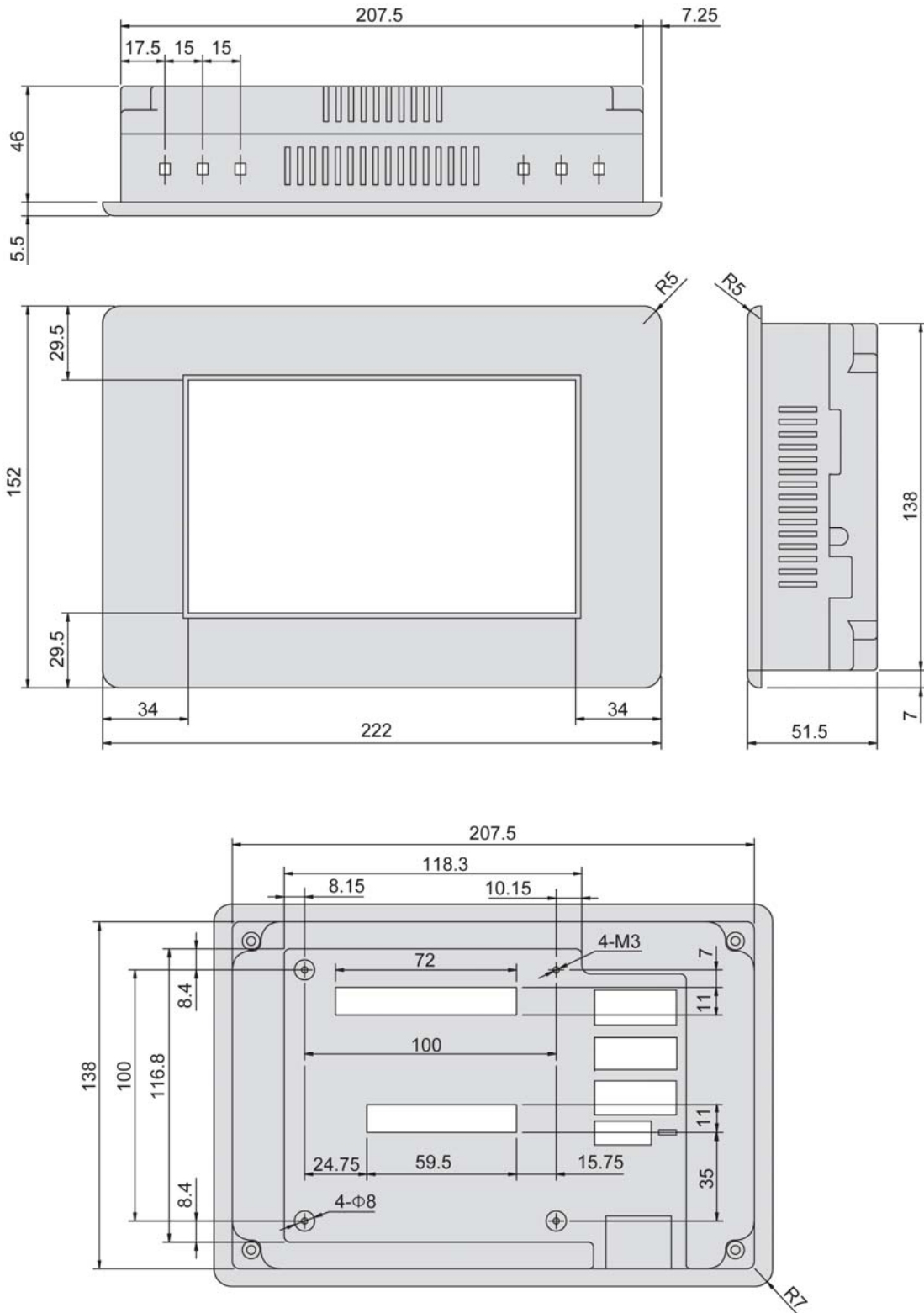




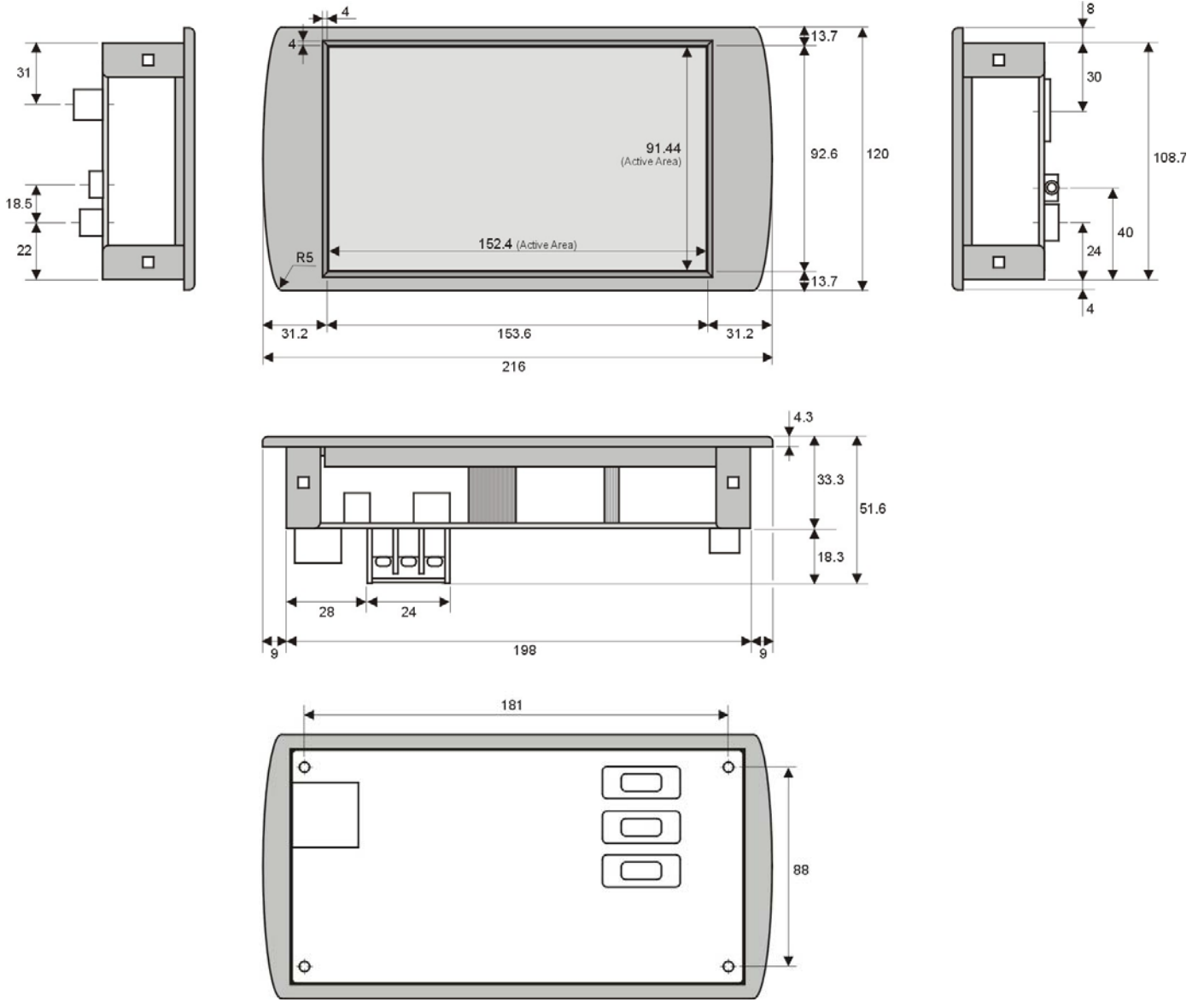
The CUWIN has 3 RS-232 Ports (COM1~COM3) and 1 RS-485 port(COM1), and COM1 is can be switched to RS-485 or RS-232. There are 4 terminals for RS-485 Port. The first one if for plus, and the last one is for minus. The second and third one is not used.

# 1.2 Dimensions

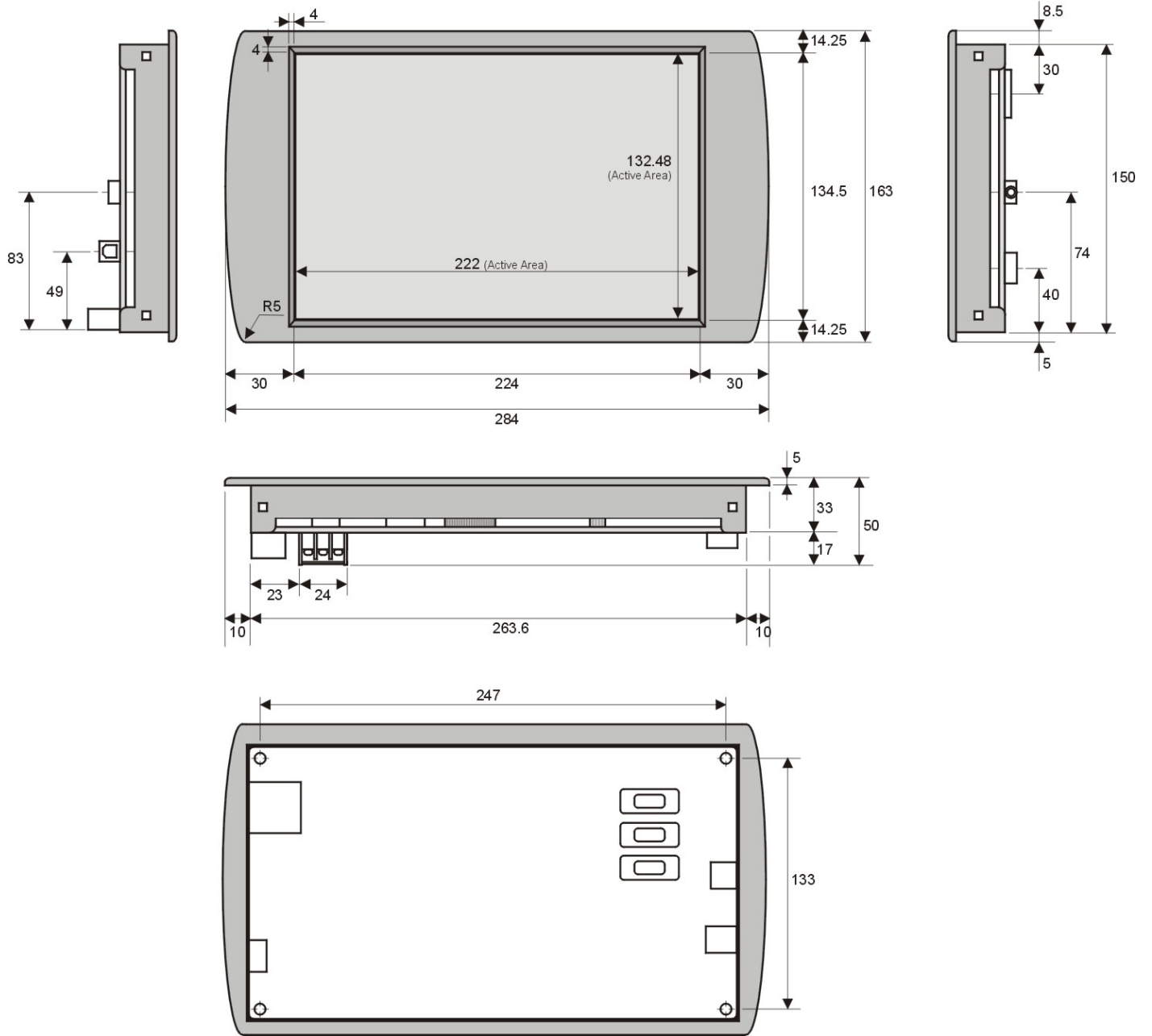
[CUWIN3500]



[CUWIN3200]



[CUWIN4300]



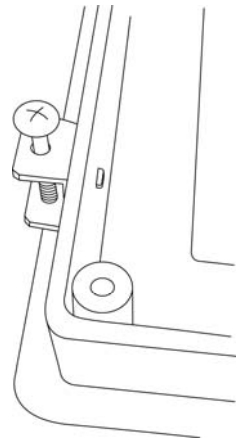
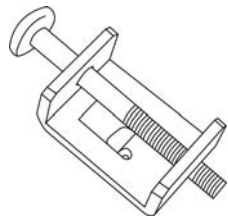
## 1.4 Panel Mounting

---

Mounting screws and brackets are provided with each CUWIN. They are designed to clamp the CUWIN bezel against a front panel, with the rear portion of the device extending through a panel cutout.

After assembling the brackets as shown below, insert the side hooks into the appropriate mounting holes on the top and bottom of the device.

Then tighten the screws enough to hold the CUWIN securely against the front panel. Do not over tighten, or you will risk damaging the CUWIN.





## 1.5 Hardware Specifications

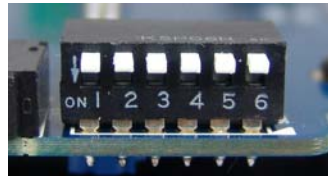
---

Processor	32-bit RISC (ARM920T Core) - 266MHz
Memory	RAM: 64MB, NAND Flash: 64MB, NOR Flash: 1MB
LCD	CUWIN3200/3500 : 7" TFT, CUWIN4300 : 10.2" TFT 18-bit / 260k colors, 800 x 480 pixels, LED backlight
Touch	Four wire resistive panel
Serial	(2) RS232-only, (1) RS232/RS485 switchable
USB	(1) host interface, (1) device interface
Storage	SD card Flash memory slot (up to 2GB)
Sound	Internal mono speaker, external 3.5mm stereo jack
Clock	Internal realtime clock
Power	DC 9V to 24V, 500mA

The Start Kit for each model includes a manual, CD, cables, and 1 GB SD Flash card. We recommend a Start Kit for your initial development purchase, and the bare devices for later production quantities.

## 2.0 Boot Modes

The CUWIN supports various boot modes which are selected with DIP switch settings.



Position	Setting			
1	Reserved			
2	Boot Mode	IntelliLCD Mode	2	ON
			3	ON
Auto-Run Mode		2	ON	
		3	OFF	
3		Desktop Mode	2	OFF
			3	OFF
4	Reserved			
5	RS232 Speed	9600	5	ON
			6	ON
19200		5	ON	
		6	OFF	
6		38400	5	OFF
			6	ON
115200	5	OFF		
	6	OFF		

## ***IntelliLCD Mode***

Graphics and text can be displayed on the LCD in response to commands received over the RS232 port, and touchscreen events can be output through the RS232 port. In this mode, the CUWIN precisely emulates an IntelliLCD. For more details, please refer to the IntelliLCD product manual.

## ***Auto-Run Mode***

The CUWIN will boot and then automatically run a specific program immediately on startup. The Windows CE desktop will not be displayed. This mode would be used in production, when your application should be the only program executed by the device. If no auto-run program is found in auto-run mode, the CUWIN will start in development mode.

### ***2.1 Loading an Auto-run Application***

1. Connect the CUWIN USB Device port (square B-style plug) to the development PC (with ActiveSync installed), or insert an SD card into a reader on your development PC.
2. Copy the desired executable file to one of the locations below:

<b>Flash Disk</b>	<b>SD Card</b>
\\Flash Disk\\AutoRun\\	\\Storage Card\\AutoRun\\
Copy the executable to the folder using ActiveSync.  There should be only one *.EXE in the folder.  The Flash storage area is restricted to a total of 30MB, but actual available space is under 3MB due to included fonts and other files. If you require extra space, you may delete the font folder if those fonts will never be used. Rather than delete the font folder, we encourage the use of an SD memory card for application storage.	If using ActiveSync, copy the executable to the folder. If using an SD card reader on the PC, copy to the AutoRun folder on the SD Card.  If the AutoRun folder does not exist on the SD Card, create it.  There should be only one *.EXE in the folder.  The SD Card should be FAT formatted, up to 2GB storage (no SDHC cards).
If there are executable files in both locations, only the program in the Flash Disk is executed.	

3. Make sure the boot mode DIP switches are set correctly to enable auto-run mode.
4. Cycle the CUWIN power supply. After the device has booted, the program in the auto-run folder will be executed.

## 3.0 Development Environment

---

The CUWIN can execute Windows CE applications developed with Microsoft Visual Studio 2005. Developing for the CUWIN is similar to PC development.

### ***Managed Code (.NET Compact Framework)***

The CUWIN supports code developed for the .NET Compact Framework. Managed code can be generated in Visual Basic .NET, Visual C#, etc.

### ***Native Code***

Application performance can be increased by developing native code instead of managed code. However, the .NET Compact Framework supplies a wide range of useful functions, so native code may require more time to develop. Developing native code requires installing the correct SDK for the target hardware, in this case the CUWIN. Please refer to chapter 4-4 for details regarding the installation of the correct SDK for developing native code. Native code can be developed with tools such as Embedded Visual C++ 4.0, Visual Studio 2005 C++ and Basic, etc.

## 3.1 Setting Up Visual Studio 2005

---

The following instructions are a guide to setting up Visual Studio 2005 and other necessary files for CUWIN development.

### *Getting Visual Studio*

Visual Studio 2005 is a software development platform provided by Microsoft. If you wish to use Visual Studio to develop native and managed code applications for the CUWIN, you will need to purchase Visual Studio from Microsoft.

Files you will need
<b>Visual Studio 2005</b> If you do not have Visual Studio 2005, an evaluation copy of Visual Studio 2005 Professional Edition is available for download from Microsoft. This version can be installed and used for a period of 90 days, which should provide enough time to test several application concepts on the CUWIN and determine whether to purchase a full Visual Studio license.
<b>Visual Studio 2005 Service Pack 1</b> This should be installed before developing applications for the CUWIN. If your version of Visual Studio 2005 does not have Service Pack 1 installed, locate the Service Pack on Microsoft's web site and install.
<b>.NET Compact Framework 2.0 Service Pack 2</b> If you intend to develop managed code (.NET) applications for the CUWIN, you should also locate and install the .NET Compact Framework 2.0 Service Pack 2.
<b>CUWIN SDK for Visual Studio 2005 and EVC++ 4.0</b> The CUWIN SDK provides target platform settings in order to compile programs for the CUWIN. This is available from the Comfile website at <a href="http://www.cubloc.com/downloads/cuwin_sdk.msi">http://www.cubloc.com/downloads/cuwin_sdk.msi</a>

If you have any difficulty locating files on Microsoft's website, you can search for them using the URL below:

<http://www.microsoft.com/downloads/Search.aspx?displaylang=en>

## ***Setting up the Development Environment***

If you have not purchased Visual Studio, you can install and test the 90-day trial version described above.

- 1 Locate and run the Visual Studio 2005 setup program, following all installation prompts. Default options should be adequate, unless your system has special requirements.
- 2 Locate and run the Visual Studio 2005 Service Pack 1 setup program, following all installation prompts.
- 3 Locate and run the .NET Compact Framework 2.0 Service Pack 2 installation file, following all installation prompts.
- 4 Locate and run the CUWIN SDK installation file, following all installation prompts.

If the above installations completed successfully, you should now have a working development environment for the CUWIN.

To easily access the CUWIN during development for program testing and debugging, you should also install ActiveSync, though it is not strictly required. If you do not wish to install ActiveSync, your application can be placed in the AutoRun folder on an SD card using an external reader, and it will be executed on startup if the DIP switch settings are correct. If you wish to install ActiveSync, please refer to section 5.0 for installation instructions.

## 3.2 Example Visual Basic project

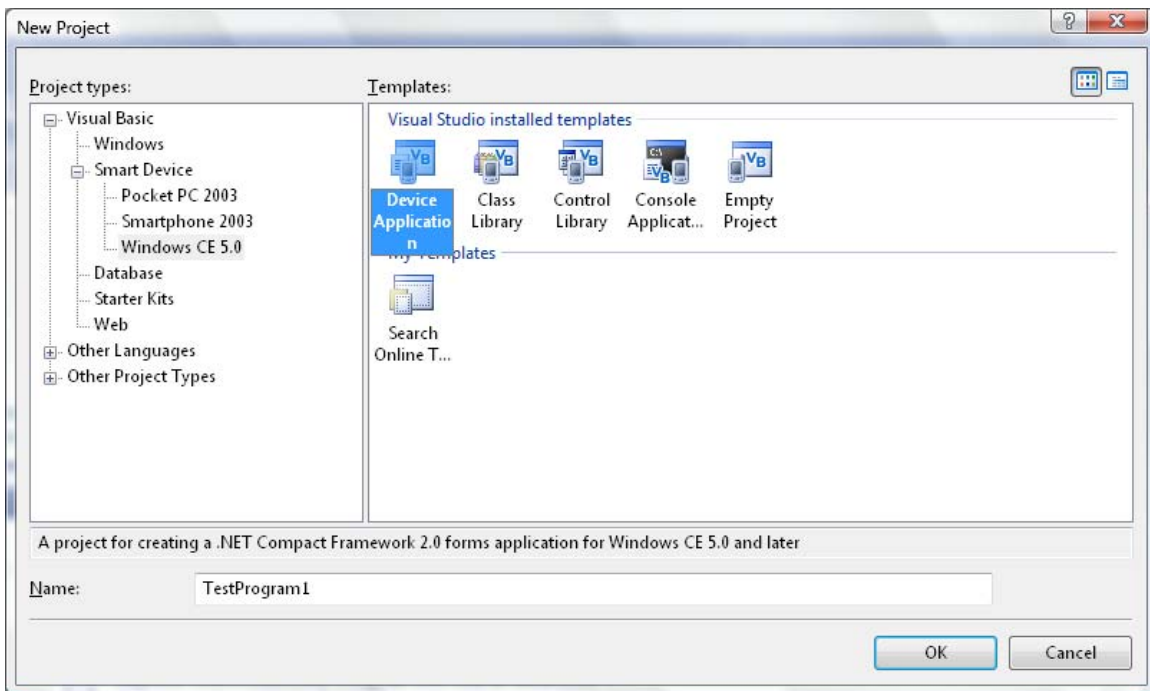
---

This example uses Visual Studio's automatic download and run feature over ActiveSync. If you do not wish to use ActiveSync, the resulting executable files can be transferred to the CUWIN using an SD card.

1. Run 'Microsoft Visual Studio 2005'.

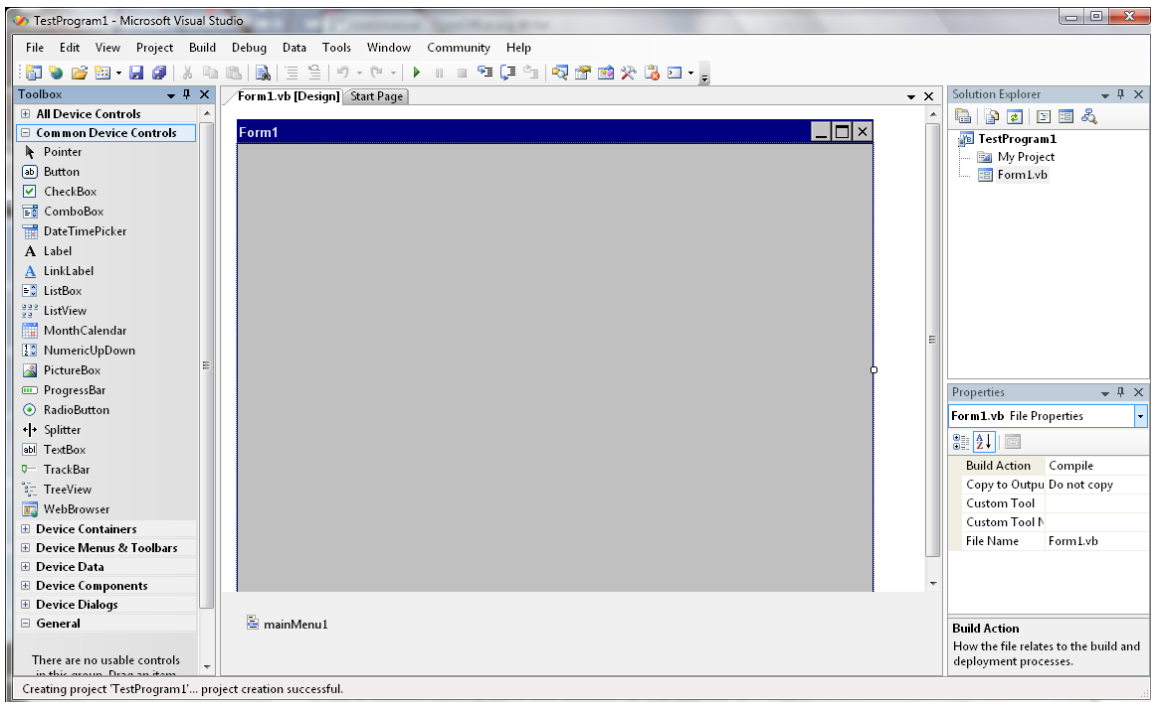


2. Choose "New Project" from the File menu.

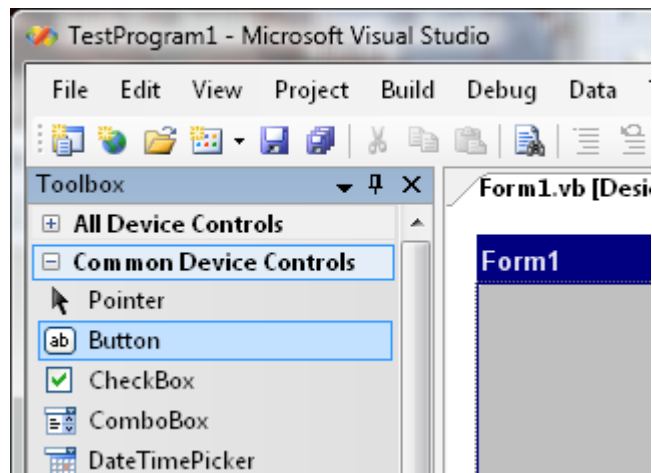


3. Select "Windows CE 5.0" and "Windows Application". Type the name of project ("TestProgram1" in this example) and click "OK".

4. A new blank project is created as shown below.

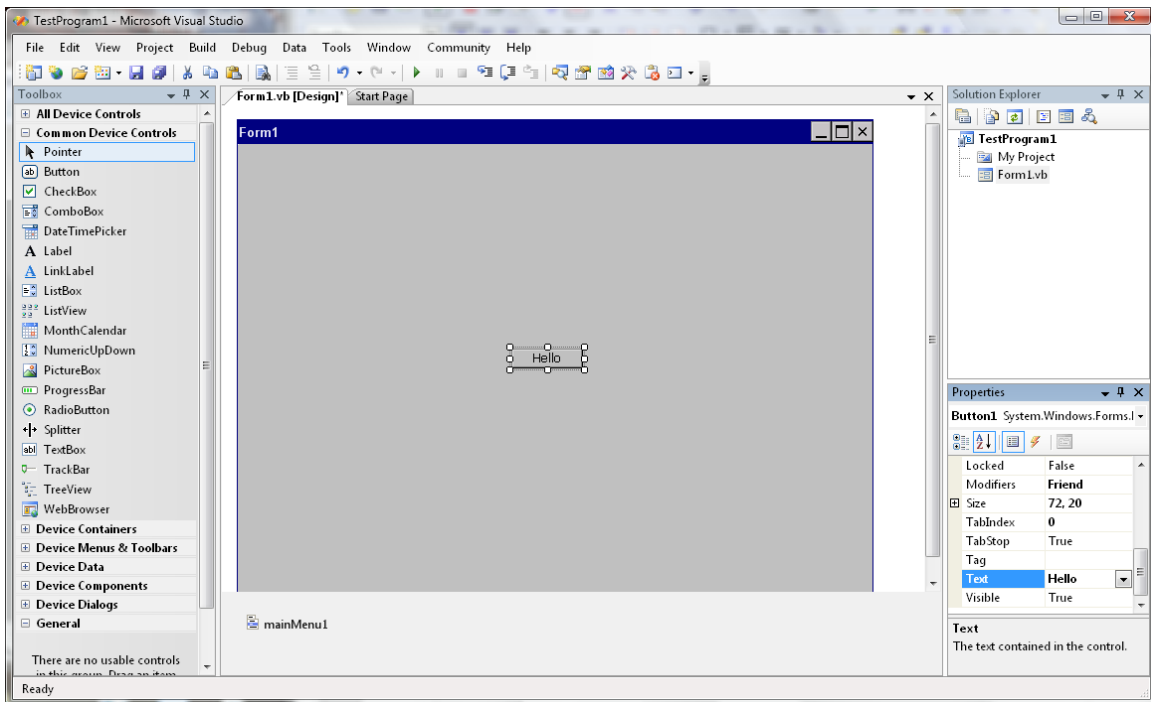


5. Drag a Button control from the toolbox to the middle of Form1.

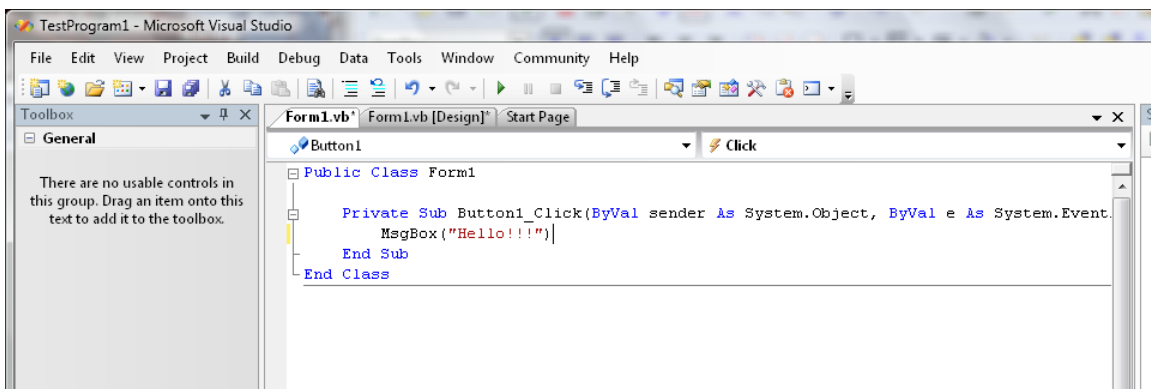




6. Select the new button. Change the Text attribute from “Button1” to “Hello”.



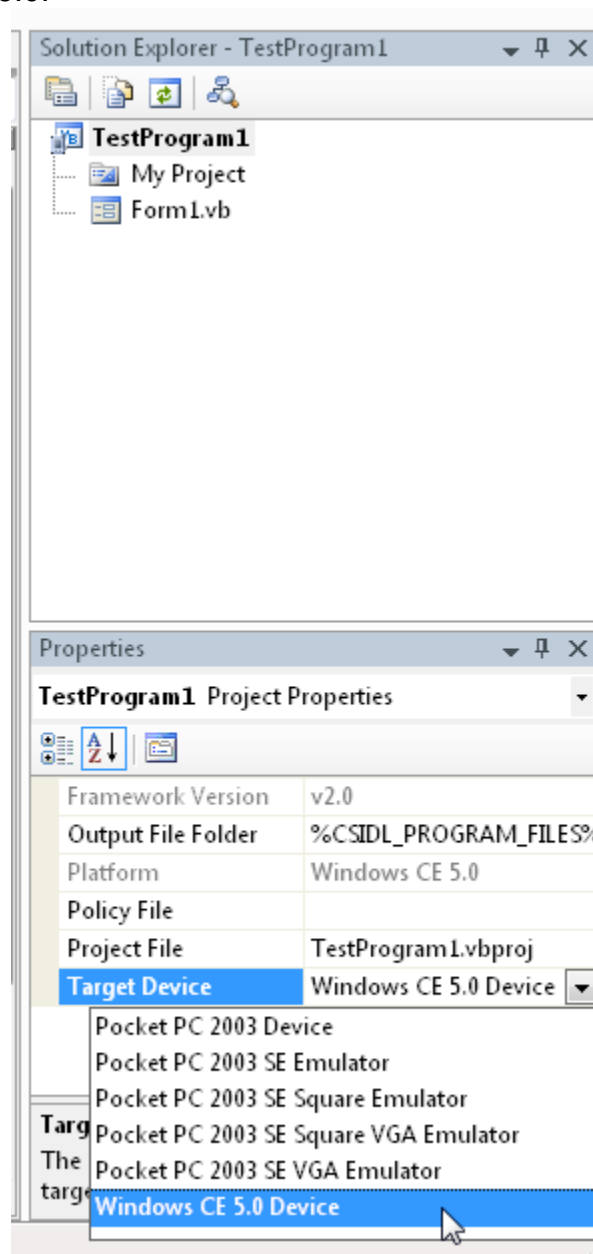
7. Double-click the “Hello” button. Visual Studio will open a window to edit the code that handles “Hello” button click events. Type 'MsgBox("Hello!!!")'



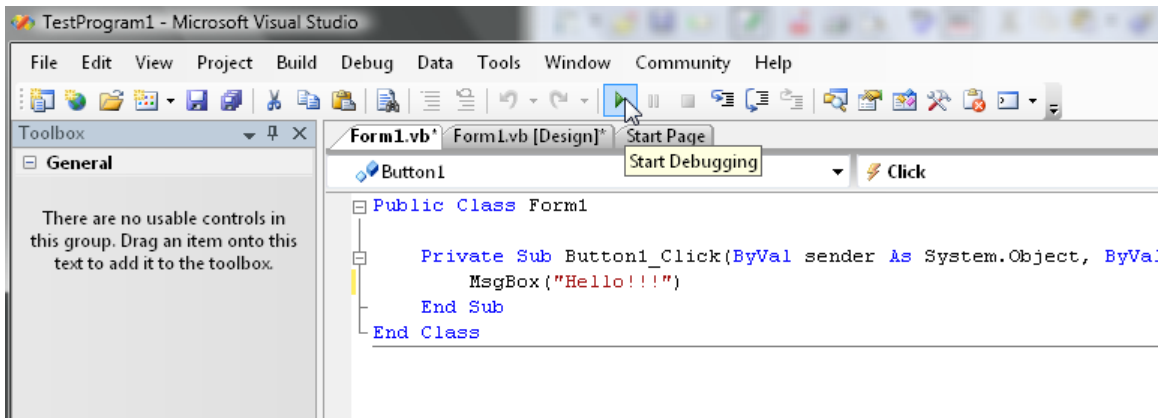
8. Set the target device.

(1) In the Solution Explorer, select the top-level item.

(2) Select “Target Device” and a list of possible devices will be displayed. Choose Windows CE 5.0.



9. Click the Start Debugging icon as indicated below. This will build and publish the program in debugging mode.



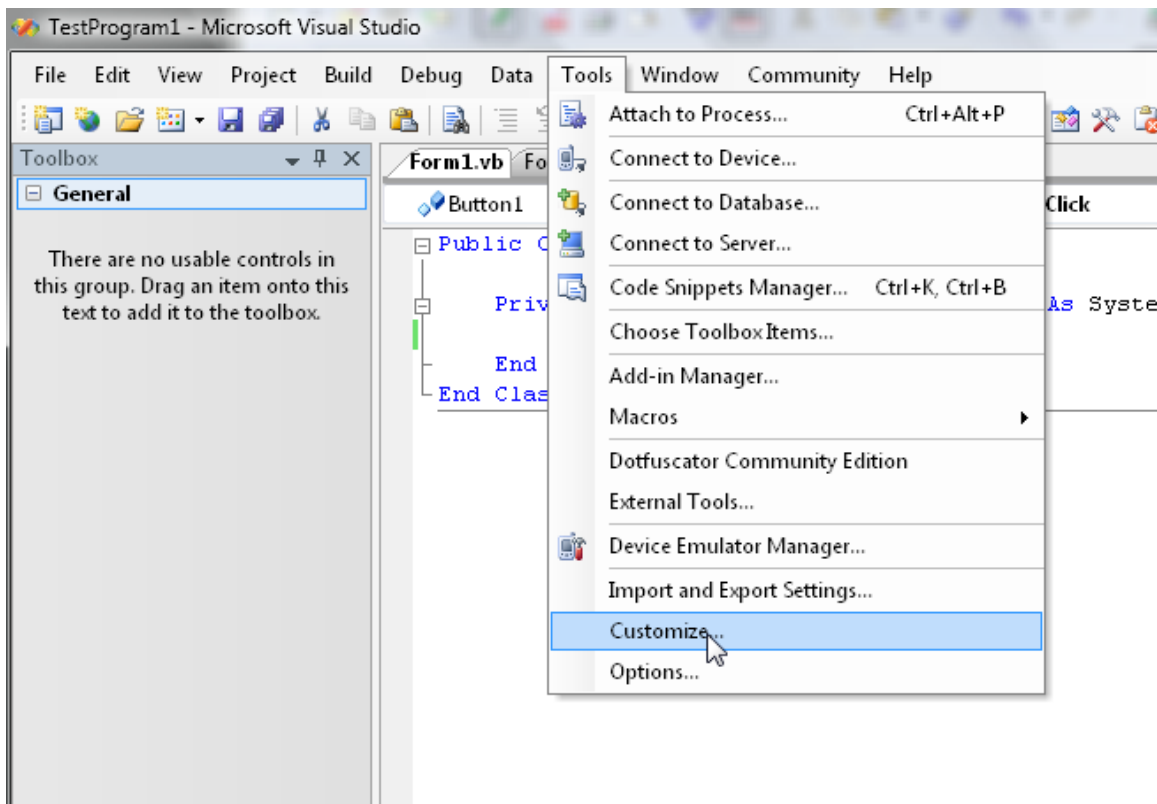
10. After a short delay, the program is transferred to the CUWIN and executed (assuming a working ActiveSync connection).

## ***Start Without Debugging***

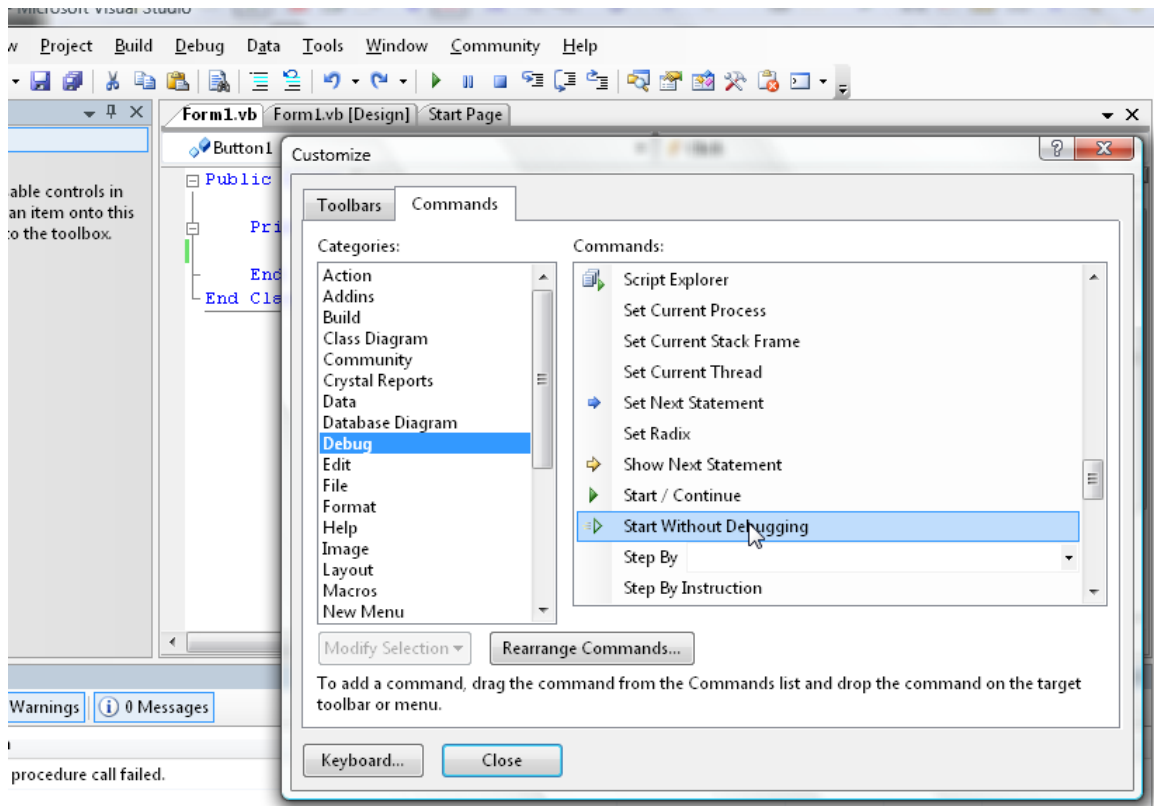
If you click the “Start Debugging” button, you can set breakpoints in the code and step through the program. This will start up the debugging interface in Visual Studio, which adds some overhead to the execution of the program. If you simply want to compile and execute the program on the CUWIN for testing without starting the debugger, it is possible to configure Visual Studio to do so with one click.

A “Start without debugging” button is not in the toolbar by default. You can add the button as shown below:

1. Click Tools > Customize... on the menu bar.

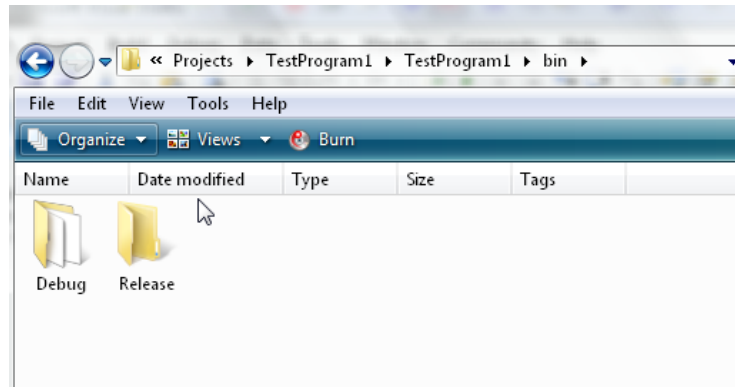


2. Click the Commands tab, then the Debug category. Find “Start without debugging” and drag it to a convenient location on the toolbar.



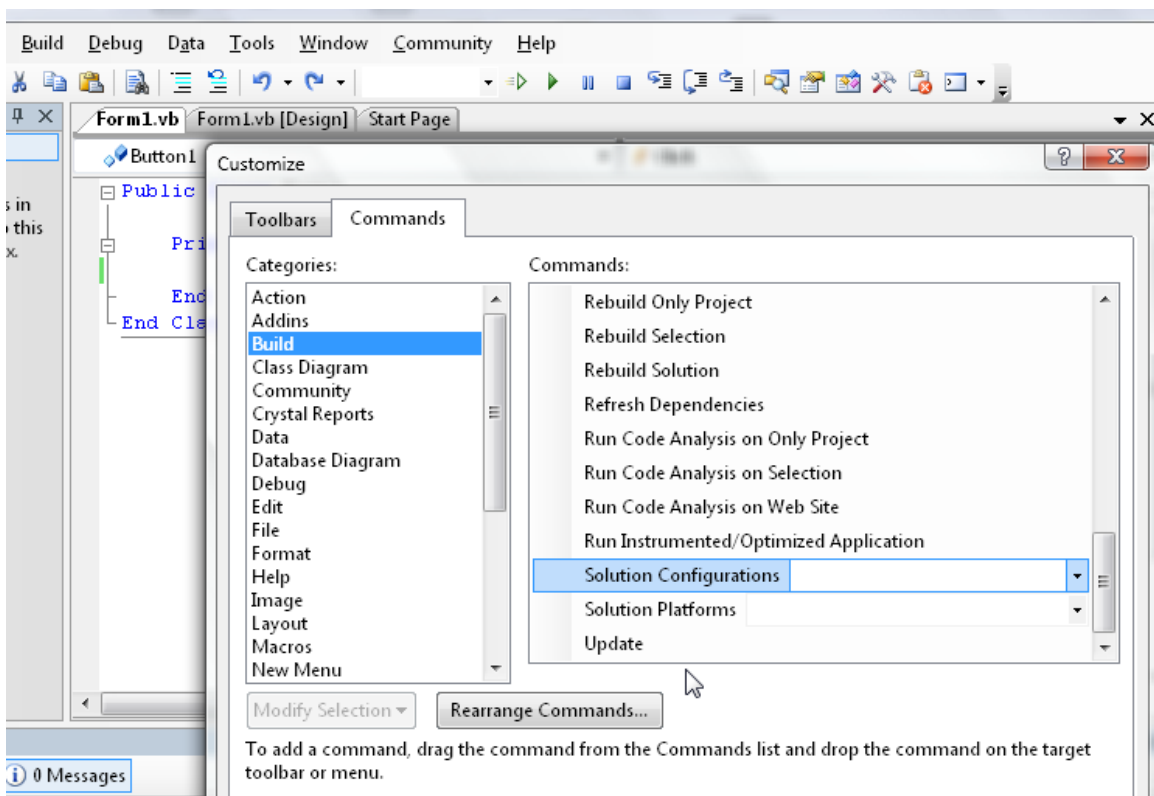
## ***Debug and Release Mode***

Visual Studio projects are compiled with debugging routines by default. This adds to the final program size and reduces performance. Once your program has been fully tested, it should be compiled without debugging routines.

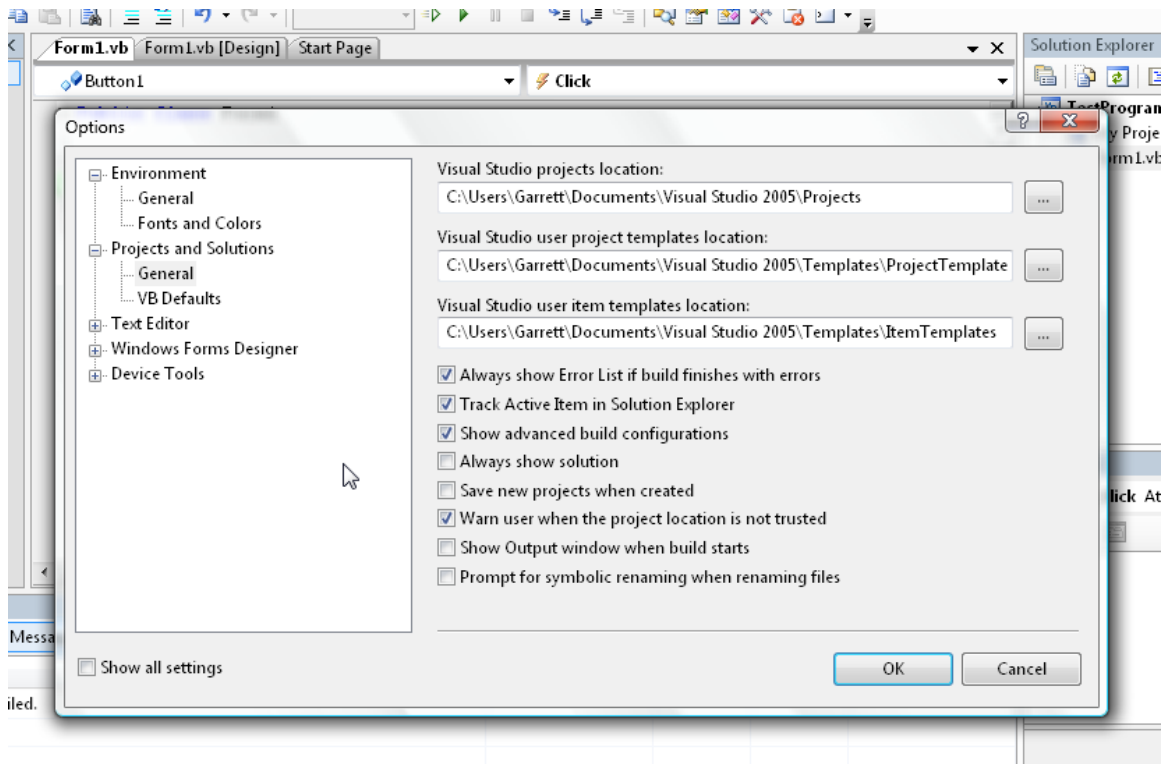


It is possible to switch between Release and Debug modes by right-clicking on the project name in the Solution Explorer. However, it is more convenient to add a button to the toolbar.

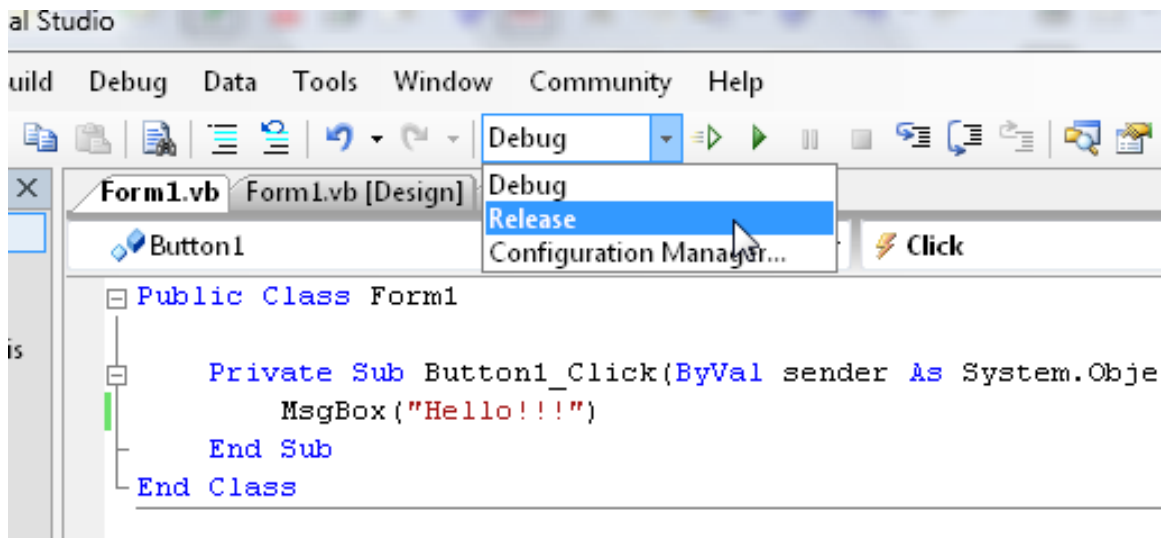
1. Click Tools > Customize... on the menu bar.
2. In the Build section, find “Solution Configurations” and drag to the toolbar.



3. Though the Solution Configurations item should now be on a toolbar, it is probably disabled. To enable, click Tools > Options in the menu bar
4. Choose the "General" section, and check the box for "Show advanced build configurations".



5. The Solution Configurations listbox should now be enabled. Debug and Release modes can now be selected easily.



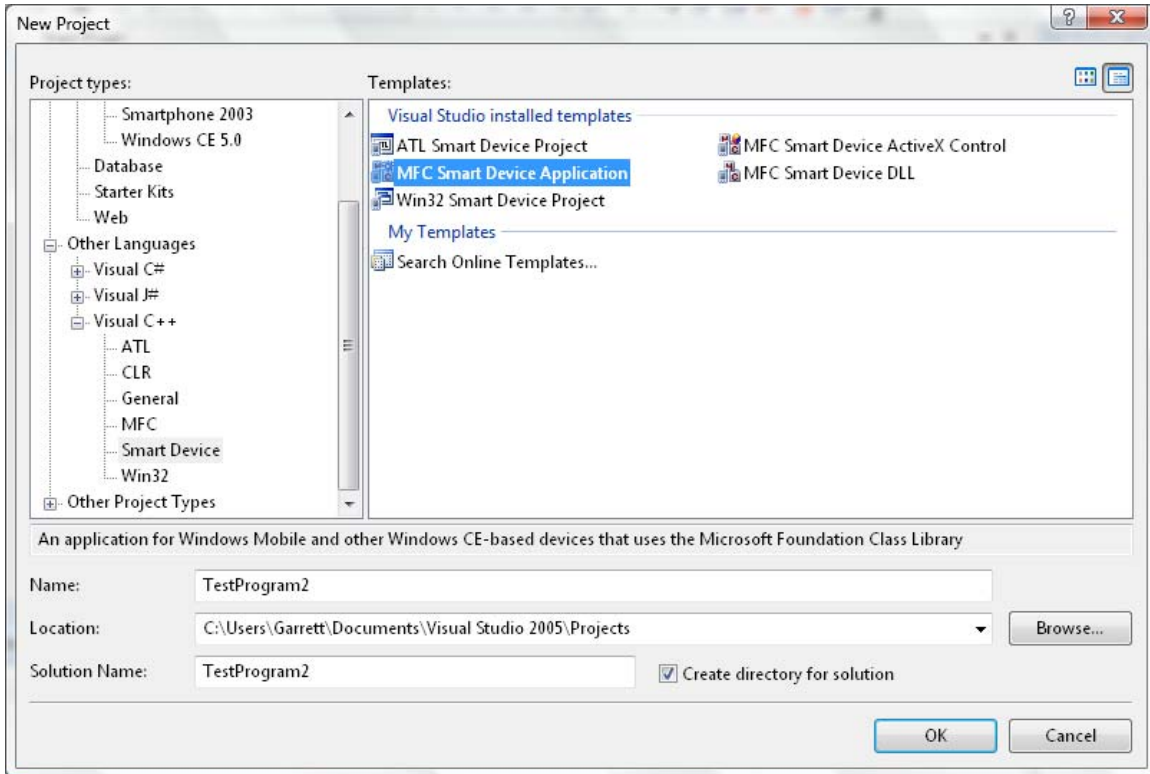


### 3.3 Visual C++ Development Notes

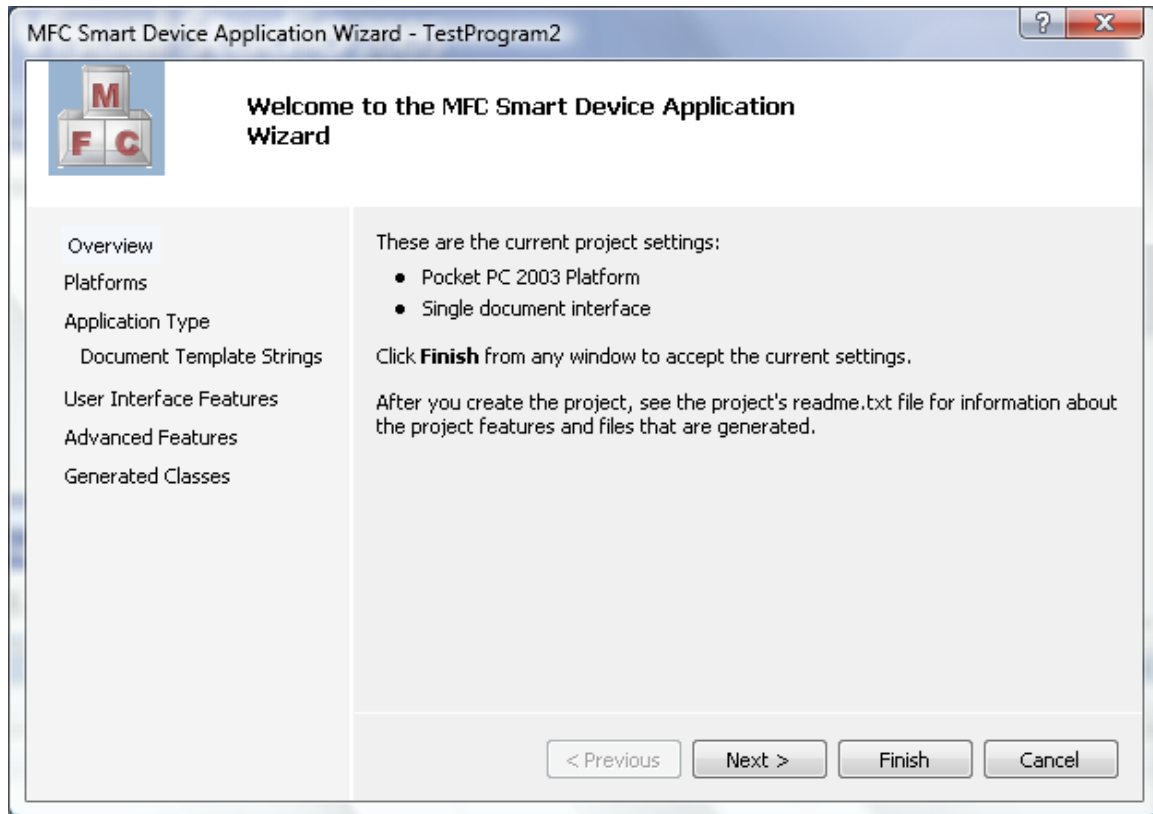
---

Visual C++ development may be preferred by advanced developers. The following is a quick startup guide to allow developers to ensure the CUWIN environment is working correctly.

1. Click File > New Project on the menu bar.
2. Choose “Smart Device” in the “Project types” box. Then choose “MFC Smart Device Application” from the “Templates” box. Type the desired name in the “Name” box.



3. Click “OK”. The Application Wizard will be displayed as below, and will guide the process of selecting desired overall program behavior.



After the Application Wizard completes, a framework application will be created for MFC application development.

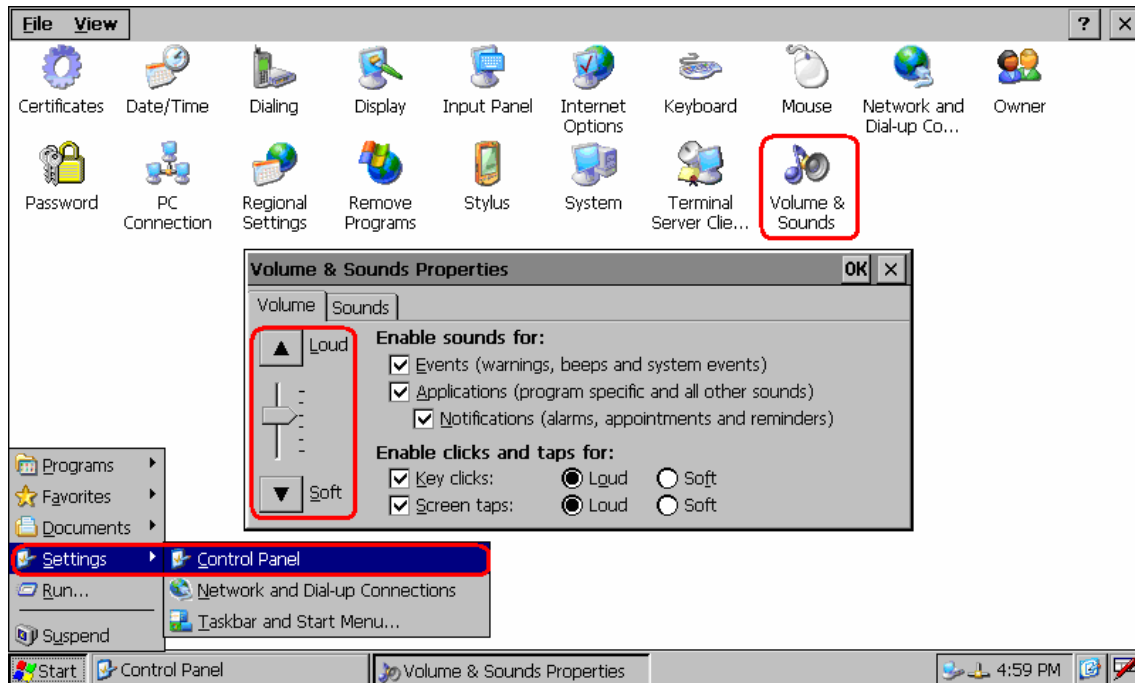
The MFC Library included in Windows CE is contains a subset of the functions available on a Windows PC environment. For additional information regarding the targeting of applications to a Windows CE 5.0 device, and for application developments in general, please refer to the MSDN library and other available program development literature.

## 4.0 Device Features and Settings

### 4.1 Sound

The CUWIN contains a built-in 0.8W speaker for mono sound. For stereo sound, headphones or external speakers can be connected to the CUWIN using the 3.5mm audio jack.

Volume can be controlled by double-clicking on the “Volume & Sounds” icon in the Control Panel on the CUWIN.



Note: if Control Panel settings are changed, the registry must be saved in order to allow the changes to persist over a power loss. Please refer to Section 4.9, Saving the Registry.

### 4.2 USB Ports

The CUWIN has one USB Host port and one USB Device port, both capable of USB Low and Full speed operation.

The USB Host port (rectangular USB A-style connector) can be used to connect USB mice, keyboards, and Flash drives. Using more than one USB device at a time will require the use of an external USB hub.

The USB Device port (square USB B-style connector) is used to connect to a desktop computer for device access over ActiveSync. Please refer to the ActiveSync section 5.0.

### ***4.3 SD Flash Memory***

The CUWIN SD memory slot supports cards up to 2 Gigabytes in capacity.

SD Flash memory can be used in the following ways:

- Application storage; images, sounds, log files etc., for use with native Windows CE programs or with the IntelliLCD-mode DrawImage, Sound, FCreate and other commands.
- Custom TrueType fonts. Additional fonts can be placed in a \Fonts directory on the storage card. On the next reboot, the fonts will be recognized automatically.
- Convenient storage for large programs. The Auto-Run feature can execute a program in an \AutoRun folder on the storage card. Please refer to section 2.0 for details on automatic execution of programs on startup.

Note: SD cards use FAT filestructure and may become fragmented with frequent use. This may reduce performance and increase the time required for the CUWIN to recognize the card. In this case, the SD card can be formatted on a PC with an SD slot.

### ***4.4 Managing TrueType Fonts***

TrueType fonts can be added and removed depending on user requirements.

Fonts can be placed in the following locations:

- \Storage Card\Fonts (or \Fonts folder on SD card)
- \Flash Disk\Fonts (must use ActiveSync)

Acceptable file extensions for fonts are .TTF or .TTC. After copying a font to one of the above locations, it will be detected automatically after the device is power cycled.

Built-in fonts: Tahoma, Courier New, Gulimche, Gulim, Dotumche, Dotum, Batangche, Batang

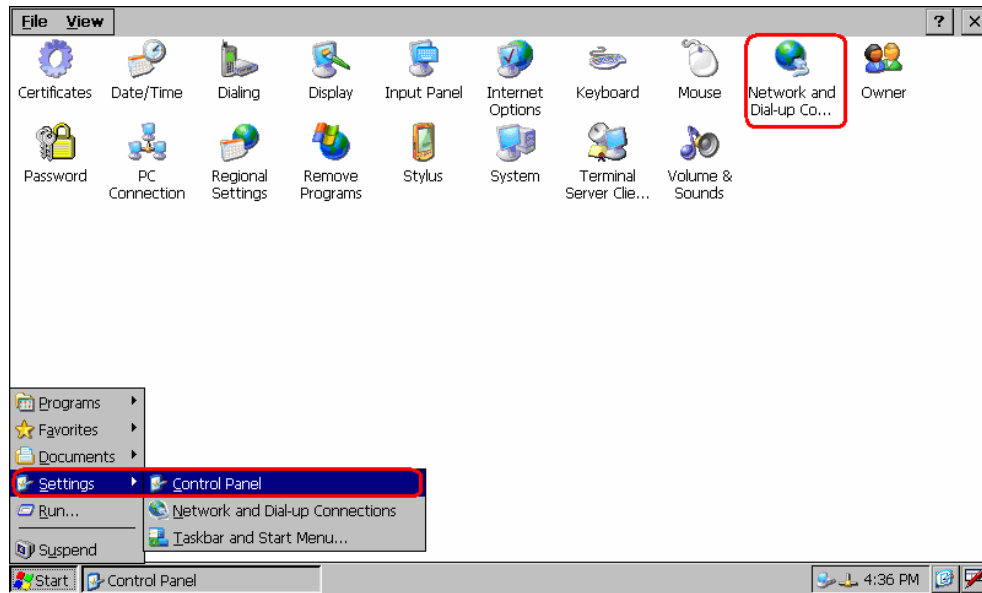
Note: The Batang and Gulim fonts are located in the \Flash Disk\Fonts folder. TrueType fonts can use a significant amount of space. Unneeded fonts may be deleted to free up space on the internal storage for user applications.

Active fonts in \Flash Disk\Fonts cannot be deleted unless the procedure below is followed:

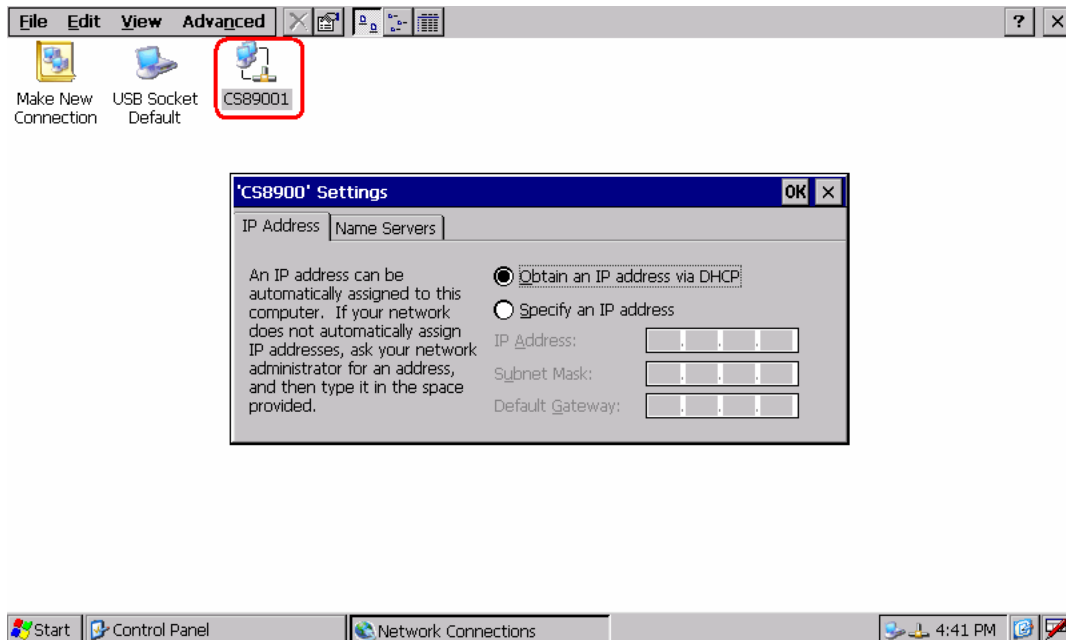
1. Change the name of the \Flash Disk\Fonts folder to a temporary name such as Fonts2
2. Power cycle the CUWIN
3. Delete the unneeded fonts in the renamed folder
4. Rename the folder back to Fonts
5. Power cycle the CUWIN

## 4.5 Internet Connection Settings

1. Double-click “Network and Dial-Up Connections” in the Control Panel.



2. Double-click the CS89001 icon. A window will appear that allows selecting either DHCP or manual entry of DNS and IP settings.

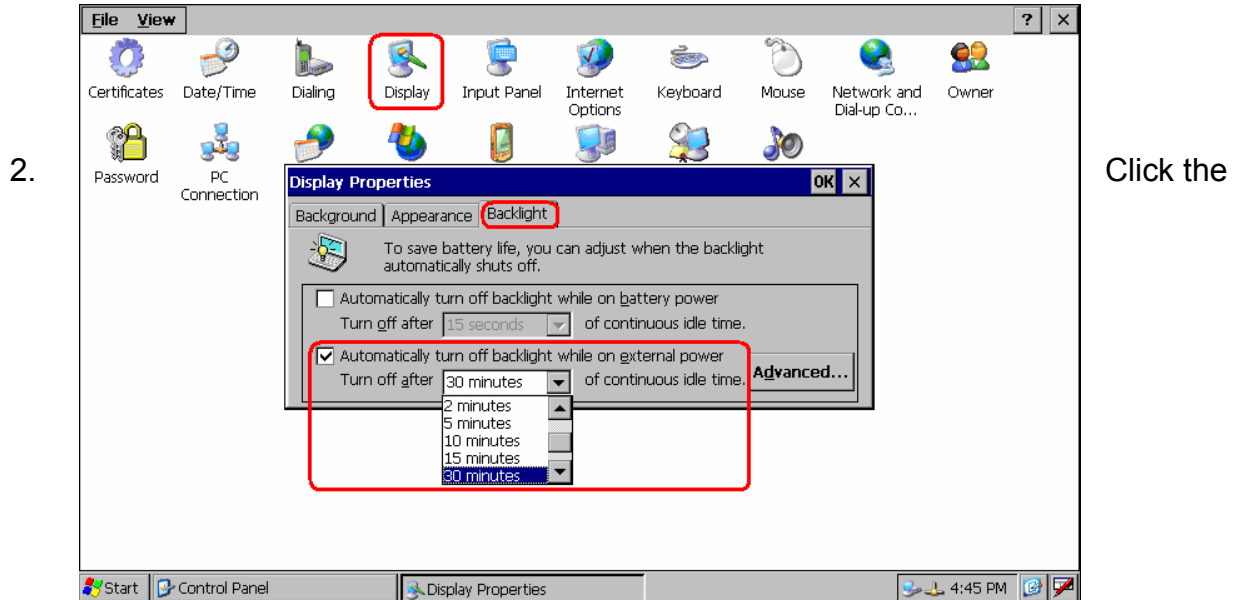


Note: if Control Panel settings are changed, the registry must be saved in order to allow the changes to persist over a power loss. Please refer to Section 4.9, Saving the Registry.

## 4.6 Screen Saver

Enabling a power-saving screensaver can reduce the power drain of the CUWIN significantly during idle periods. This can also improve the life of the backlight and LCD.

1. Double-click “Display” in the Control Panel.



“Backlight” tab. Since the CUWIN has no battery power option, check the box for “Automatically turn off backlight while on external power” and select the desired delay. Alternately, clear this check box if no display timeout is desired.

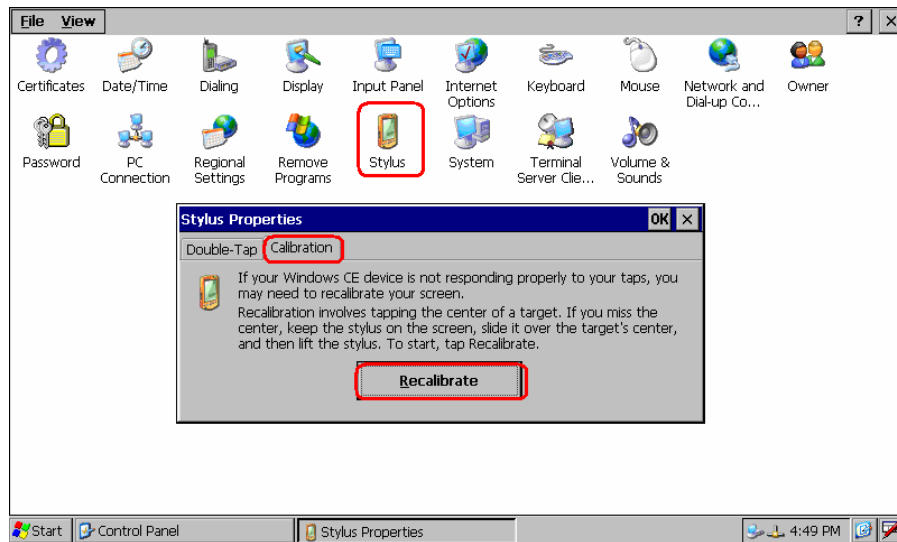
3. Click on the “Advanced” button and check the box that enables the backlight to be enabled while on external power. **If this option is not selected, the CUWIN will not return to an active display mode when the screen is touched.** The warning about reduced battery life does not apply to the CUWIN.

Note: if Control Panel settings are changed, the registry must be saved in order to allow the changes to persist over a power loss. Please refer to Section 4.9, Saving the Registry.

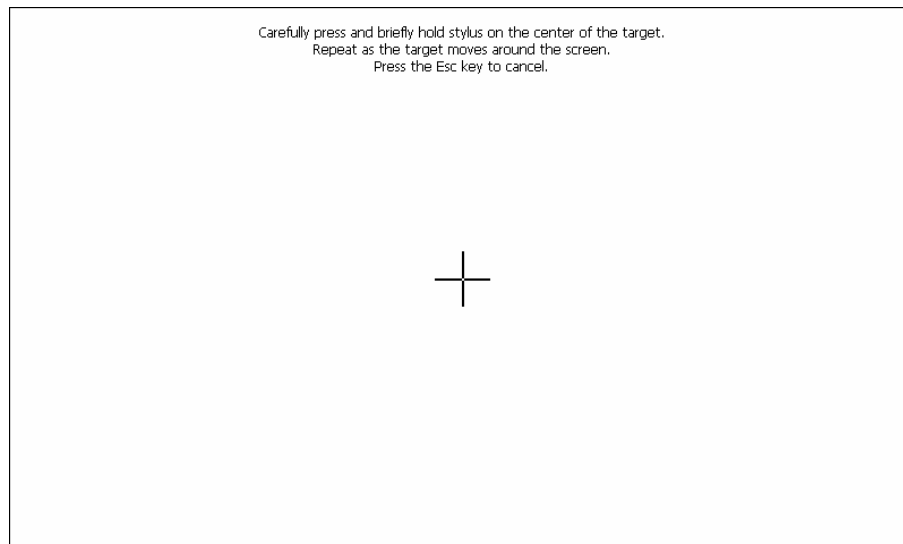
## 4.7 Touch panel calibration

Note: Initial calibration is performed at the factory. If the screen needs to be recalibrated due to wear, using a stylus instead of a finger will result in a more accurate calibration.

1. Double-click on the “Stylus” Control Panel icon.



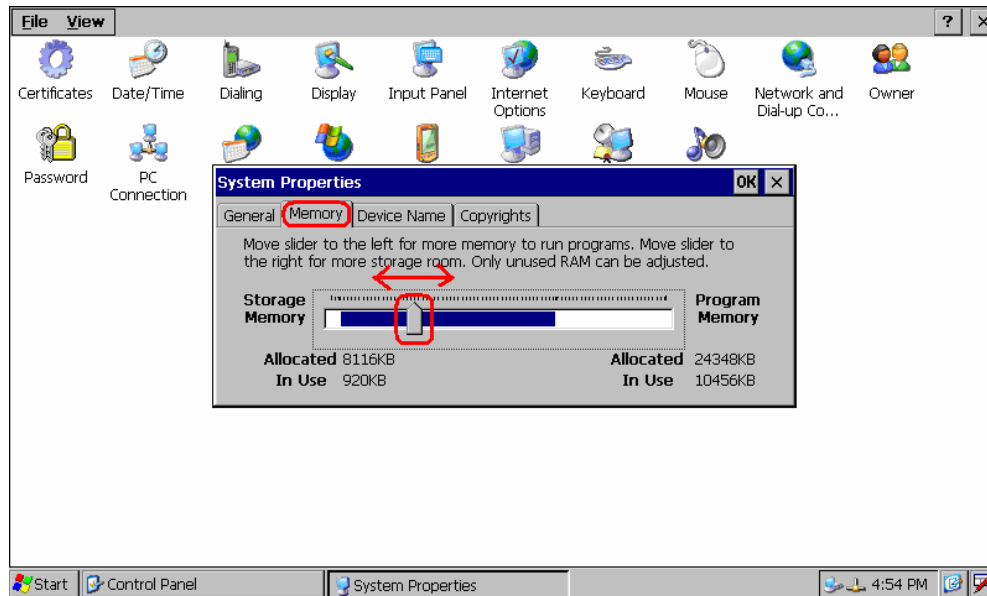
2. Click the "Calibrate" tab, then click the “Recalibrate” button. The utility will position crosshairs on different areas of the screen, which should be pressed with a stylus as accurately as possible. If the initial calibration fails, repeat carefully.



Note: if Control Panel settings are changed, the registry must be saved in order to allow the changes to persist over a power loss. Please refer to Section 4.9, Saving the Registry.

## 4.8 Memory settings

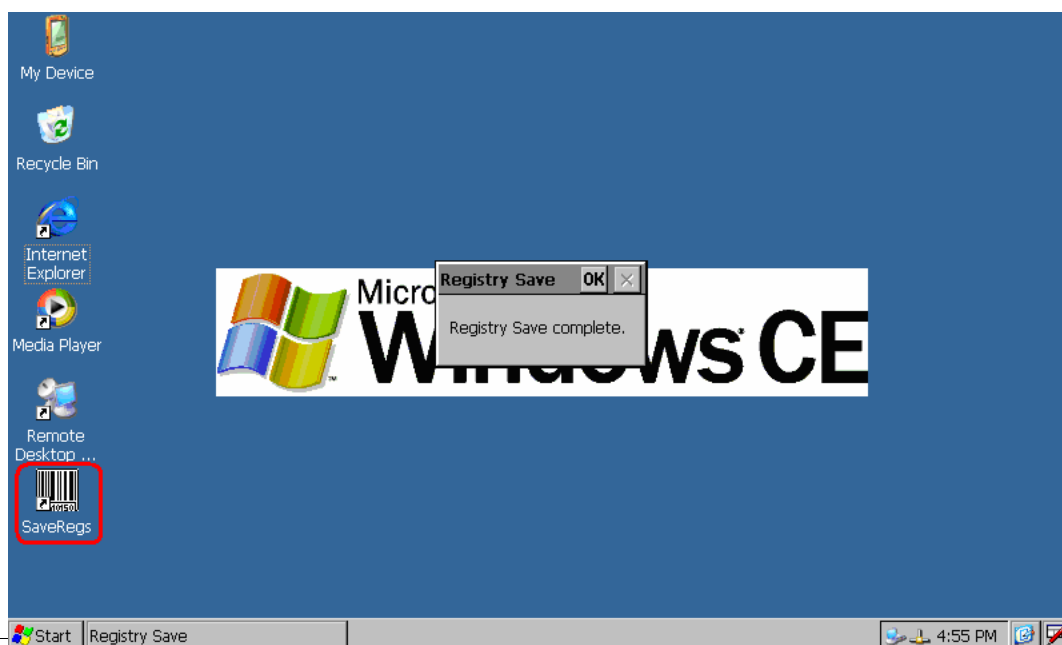
1. Double-click the “System” icon in the Control Panel and click the Memory tab.
2. Move the slider bar to adjust the memory available to programs.



Note: if Control Panel settings are changed, the registry must be saved in order to allow the changes to persist over a power loss. Please refer to Section 4.9, Saving the Registry.

## 4.9 Saving the Registry

If changes are made to the system registry (including most settings made in the Control Panel), they will disappear after the next power cycle. To prevent this, click the SaveRegs icon on the Desktop after a configuration change has been made.



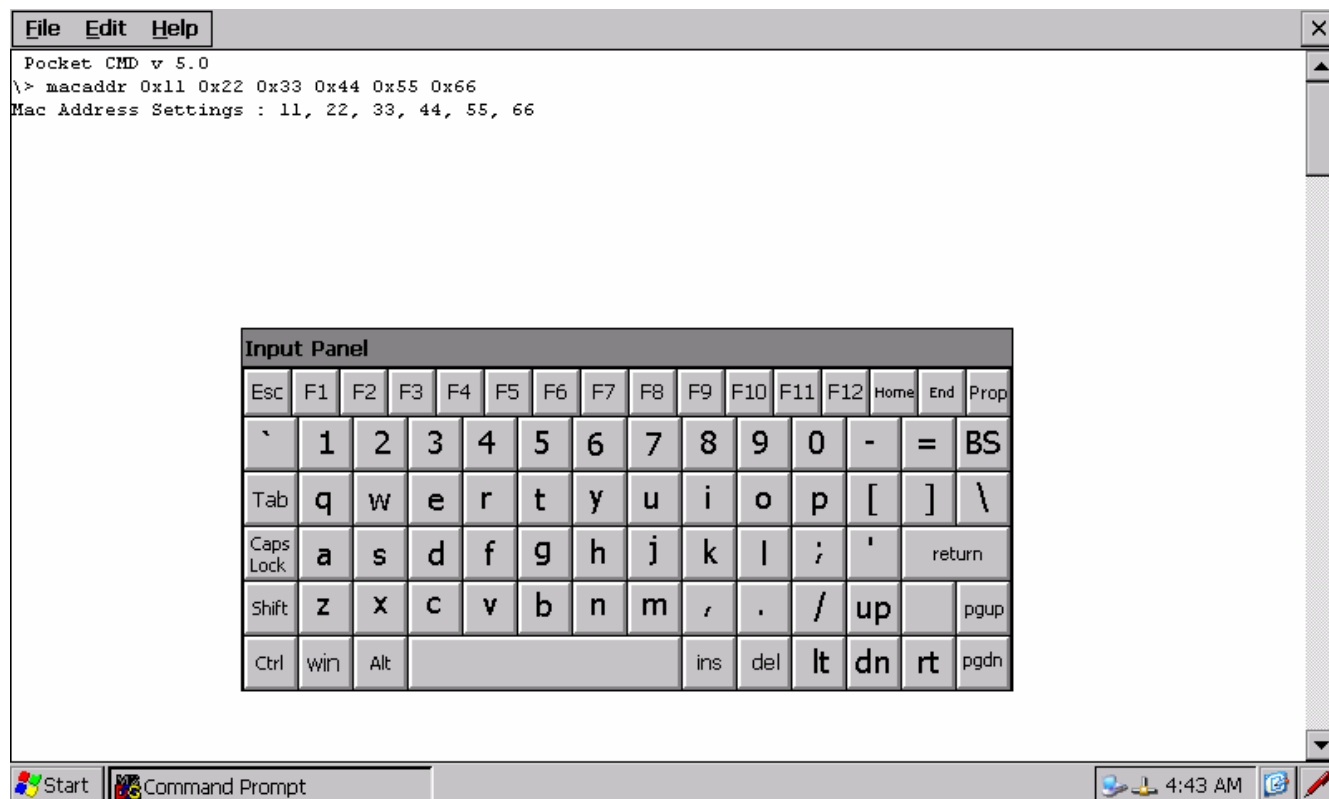


## 4.10 Changing the MAC address

The Cuwin has the default MAC address of factory setting, but it is desirable to modify it when you use many Cuwins under the same network.

You can configure MAC address following these steps:

1. Click [Start] -> [Programs] -> [Command Prompt]
2. Use MACADDR command with the address parameters.  
i.e) macaddr 0x11 0x22 0x33 0x44 0x55 0x66



3. Save the Registry.  
(See 4.9)

## 5.0 ActiveSync

---

ActiveSync uses the USB Device (square B-style plug) port to connect the CUWIN to a desktop computer for easy file transfer and application development/debugging. Visual Studio 2005, EVC++ 4.0, and NS Basic integrate with ActiveSync to provide one-click installation and execution of programs on the CUWIN, as well as remote configuration and debugging tools.

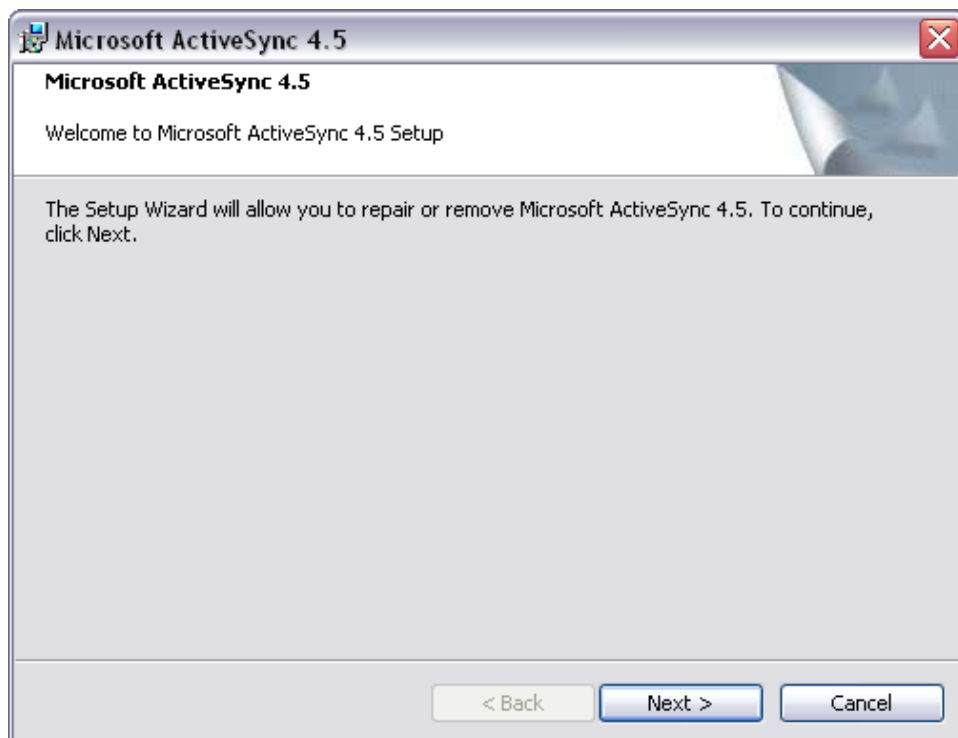
### 5.1 Installing ActiveSync

The version of ActiveSync described here is 4.5. It can be downloaded from Microsoft at the following location (URL is subject to change):

<http://www.microsoft.com/windowsmobile/activesync/activesync45.msp>

If the link above does not work, perform a search on Microsoft's web site for the download.

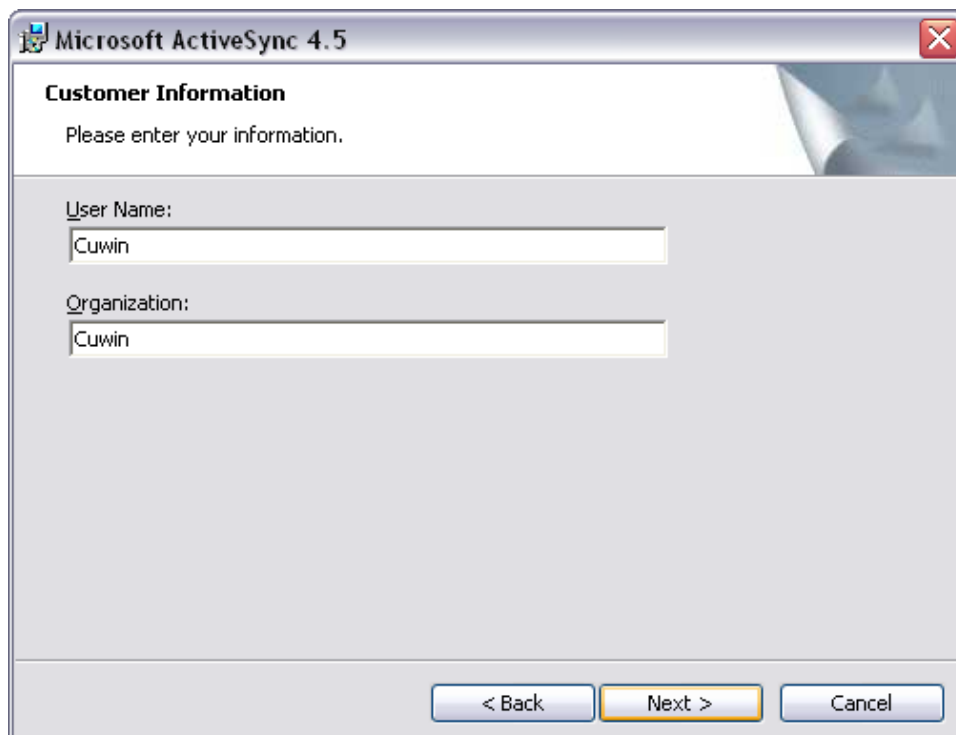
1. Do not connect the CUWIN to the PC before installing ActiveSync.
2. Locate and double-click the ActiveSync 4.5 installation file. Click "Next" when the screen below appears.



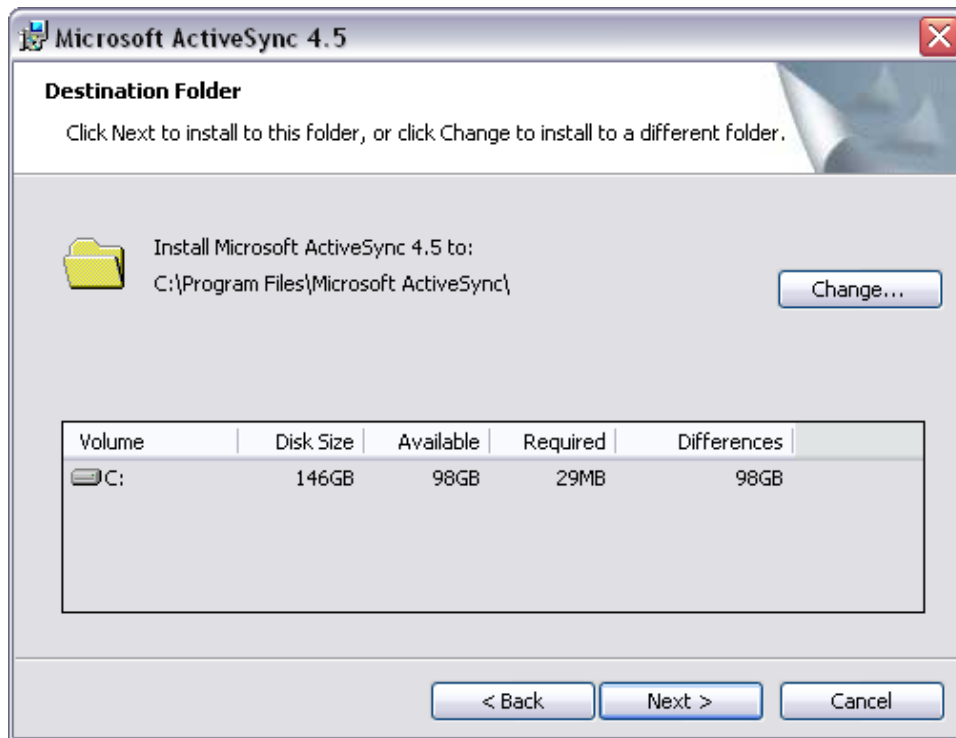
3. A license agreement screen will appear. Click the radio button to accept the terms and click “Next”.



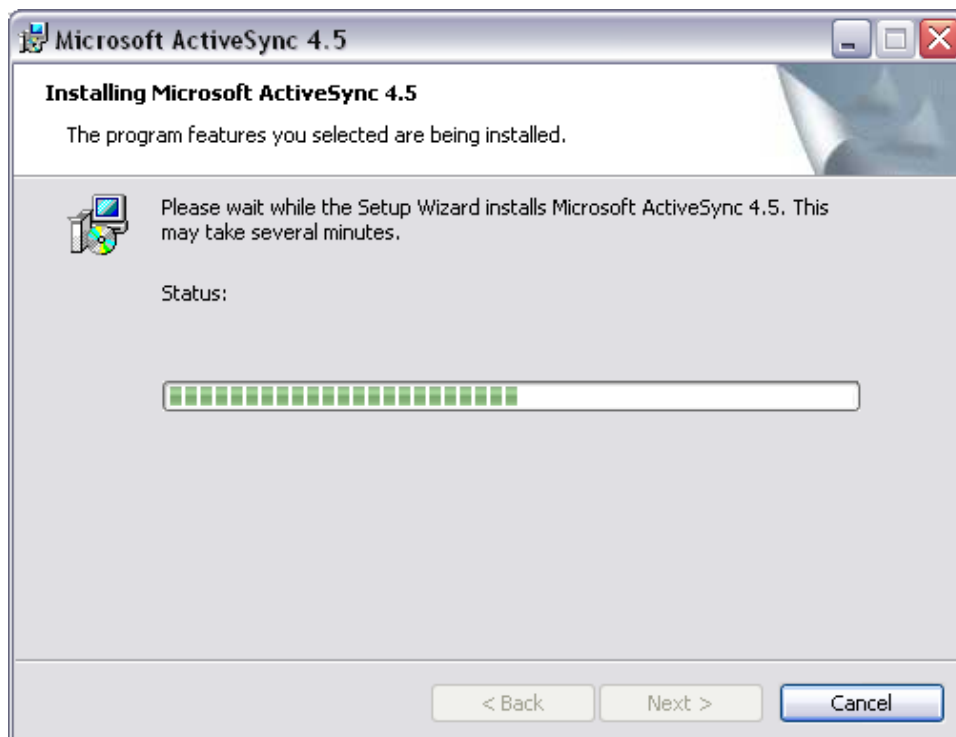
4. Type in some customer information. Click “Next”.



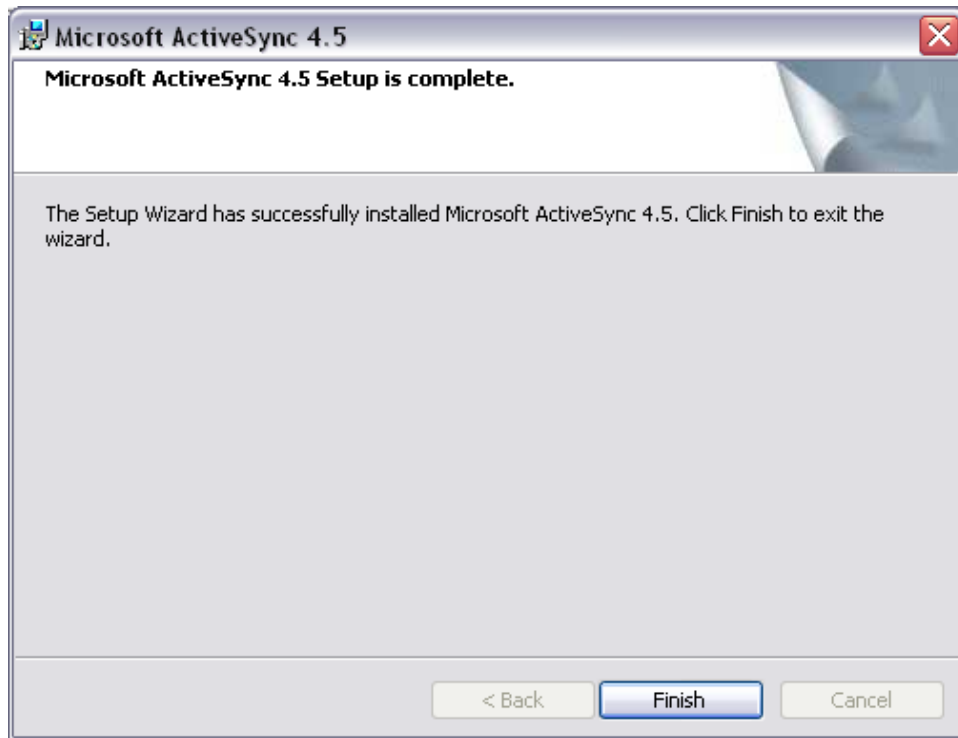
5. If a different install location is required, adjust as necessary; otherwise, click “Next.”



6. ActiveSync will install as shown below.

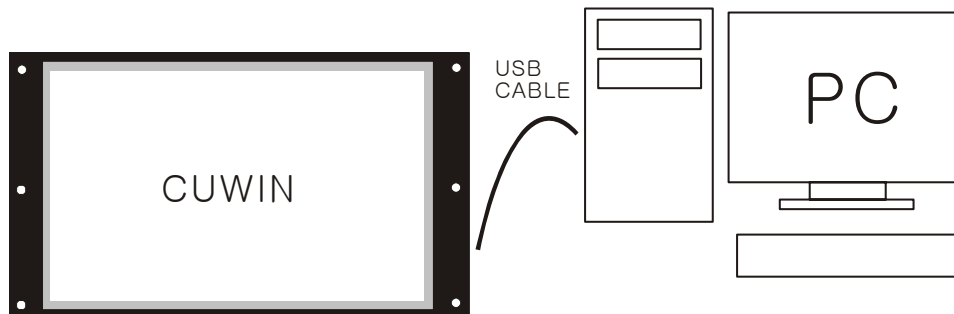


7. If the installation was successful, the following screen will be displayed. Click “Finish” to exit the installation program.



## 5.2 Using the CUWIN with ActiveSync

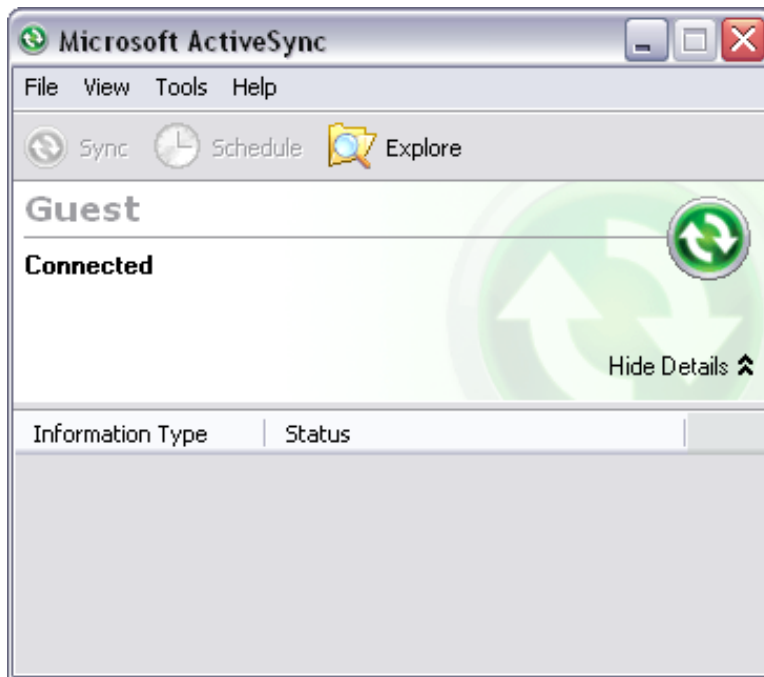
1. Connect the CUWIN Device port to a USB port on a desktop PC with ActiveSync installed.



2. The first time the CUWIN is connected, ActiveSync will ask to set up a partnership. This is typically intended for mobile phones and PDAs, so select “No” and click “Next.”

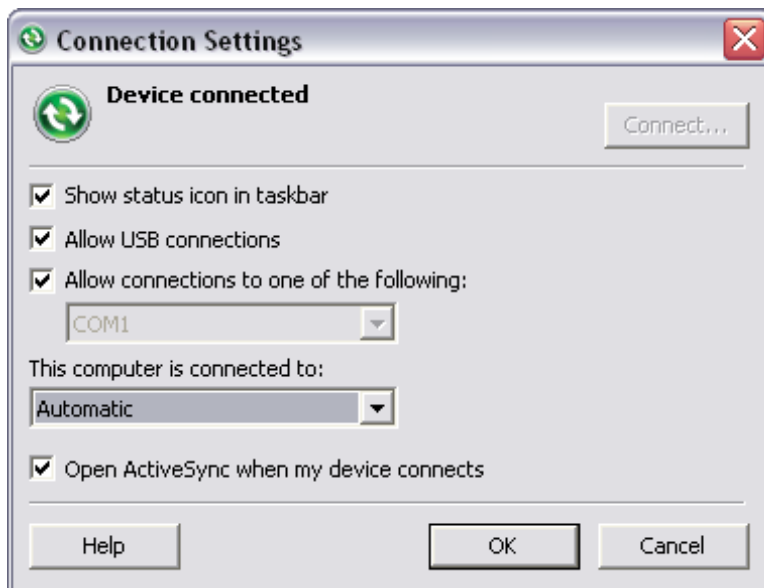


3. ActiveSync should connect to the CUWIN and display the following window:



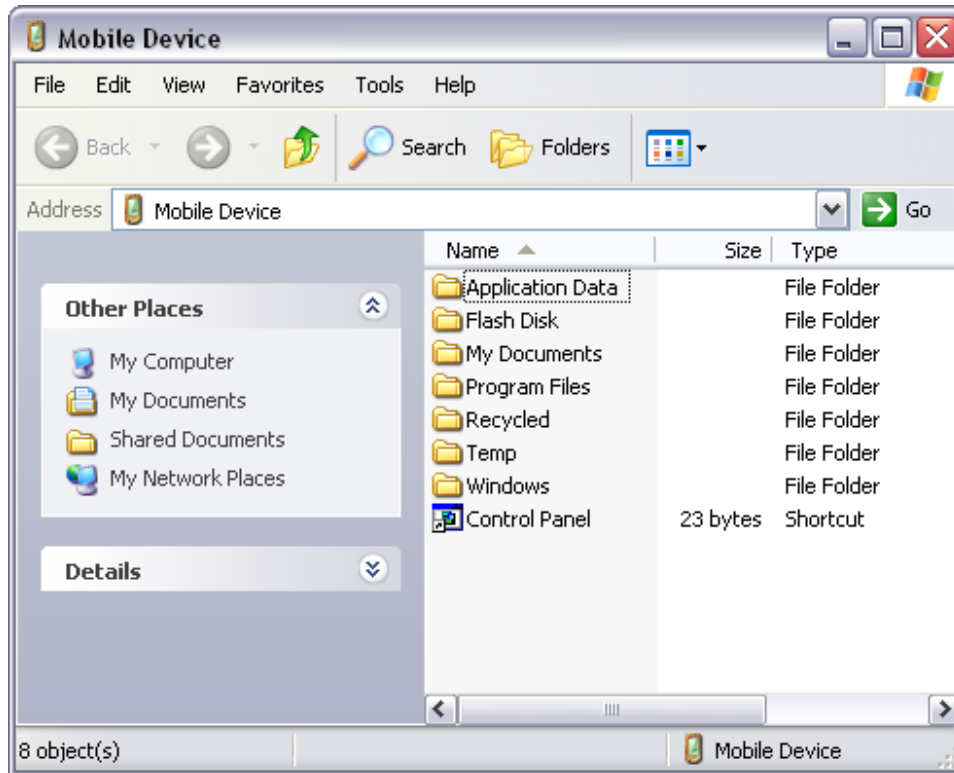
If ActiveSync does not automatically connect when the CUWIN is plugged into the PC, right-click on the ActiveSync system tray icon and click "Connection Settings."

Make sure the settings are as below:



### 5.3 Accessing CUWIN Storage with ActiveSync

With the CUWIN connected, click the “Explore” icon.



User-accessible folders:

- Flash Disk: Contains non-volatile device storage that will persist between reboots, can contain user applications and data while power is off. This memory area has an upper limit on write cycles as all Flash memory does; avoid unnecessary writes to this area.
- Storage Card: Contains the filesystem of an SD Flash card, if one is present. Can be used for applications and data storage up to 2 Gigabytes.
- Other folders: User-inaccessible or volatile folders stored in RAM. Any writes to this area will disappear on the next reboot.