Vector 300

User Manual

Prepared by Prolificx
Date Thursday, 2 November 2006
Version V300_UserManual_Rev2.2.doc

Table of Contents

V300

Important
This guide is intended to provide advice and guidelines for OEM developers. Responsibility for how the information is used lies entirely with the developer. It is the developer's sole responsibility to install and use the instrument in a manner that will not cause accidents, personal injury or property damage.

The Global Positioning System (GPS) is operated by the US Government which is solely responsible for its operation, accuracy and maintenance. The GPS system is subject to changes which could affect the accuracy and performance of all GPS equipment anywhere in the world. To reduce the risk of misusing or misinterpreting this product, the developer must read and understand all aspects of this manual.

Communications coverage, availability, or grade of service provided, whether through a service provider or otherwise, are not provided by Prolificx and therefore the responsibility of the developer.

Any changes or modifications to the V300 not authorized by Prolificx may void the users authority to operate it under the rules of the FCC.

Prolificx disclaims all liability for any use of this product in a way that may cause accidents, direct or indirect damage or violate any laws. This manual represents the product as at the time of printing. Prolificx reserves the right to make changes to specifications without notice.

This device complies with Part 15, Part 22, Part 24 and Part 25 of the FCC rules. Operation is subject to the following conditions: (1) this device must not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Changes made to the V300 without the authority from Prolificx may void the users right to operate the equipment.

Introduction

Prolificx's Vector family of Windows® CE hardware platforms are versatile, cost-effective solutions for a broad range of telematics and M2M applications, including:

- Data Logging
- Security
- Vehicle Tracking
- Fleet Management

- Asset Management

With variety of options, the Vector family offers solution developers, system integrators, VARs, ASPs and fleet managers a range of products that can easily be configured to provide the optimal hardware solution. Moreover the Windows CE operating system simplifies application software development and provides access to the many extended features of an industry standard operating system.

Additional assistance and some hardware applications information is available in the Vector 300 Installation Manual

Vector 300FM

The Vector 300FM is a low cost in-vehicle device developed specifically for fleet management and vehicle location applications. It has been designed from the ground up for extremely low power consumption and can be woken from standby modes by various external events - including incoming calls, ignition, auxiliary inputs and device movement.

Despite its small size and low power consumption, the Vector 300FM exposes a rich set of hardware and protocol implementations including Bluetooth, MMC/SD Card, J1708 or RS232, protected digital and analog inputs, protected driver outputs, AC'97 audio capabilities, Global Positioning System (GPS) and a GPRS cellular modem. An optional rechargeable internal backup battery enables some degree of operation when main power is unavailable. With a wide range of flexible options, a Vector solution will fulfill your hardware requirements for almost any telematics or M2M project.

Vector 300DM

The Vector 300DM has all the functionality of a 300FM, but also includes a modem module for the OrbComm satellite network.

The Vector 300DM utilizes cellular data protocols when the GSM network is available, and has a customizable, configurable satellite interface as a backup option to the cellular data medium. The satellite modem communicates via OrbComm Low Earth Orbit satellites. This feature makes the Vector 300DM suitable for rural and maritime telematics applications, extending its range effectively to anywhere in the world with line of sight to the sky. The Vector 300DM can be configured with priority queues to maximize data availability and minimize financial cost of data transmissions, depending on data priority.

An overview of the OrbComm Satellite system can be downloaded from here

External Ports

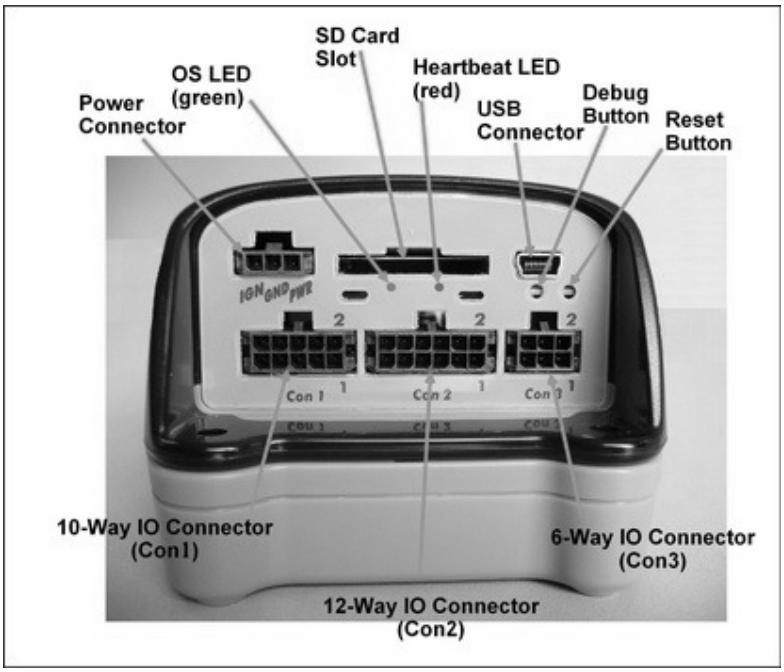
The Vector 300 has several connectors as illustrated below. The following restrictions/considerations apply when choosing installation locations and logistics for the Vector 300:.

WARNING: The antenna used for these transmitters must be installed to provide a separation distance of at least 89.23 cm from all persons and must not be co-located or operating in conjunction with any other antenna or transmitter.

- Access to the USB socket and the SD card slot
- Visibility of LEDs for status
- The access cover for the SIM card is deliberately designed to be difficult to remove as it is not intended for user access
- Allow at least 60 mm between the V300 and an obstacle for free cable access.
- Mount with 4 screws or bolts to a stable surface. Size is 140 x 89 x 45 mm.
- Orientation - if required to be mounted vertically then orient so the antenna cable / SIM card slot are located on the top.

Front View

The front view of the Vector 300 contains a 3 prong power connector (from left, ignition [orange], ground [black] and power [red]), as well as the Secure Digital card slot, status LEDs (see below), USB type B connector (for Microsoft ActiveSync), and several I/O connectors whose function and pinouts are specified below.



NOTE: The Vector 300DM is shown. The Vector 300FM looks identical to the picture shown here, without the larger grey base holding the satellite modem module.

IO Connectors

The Vector 300 has three different IO connectors which serve a variety of purposes from serial to USB to audio, as illustrated in the following table:..

	10-way				
Alternative				J1708+	
Standard	CAN+	V_{USB}	USBH-	RS232 RxD	V_{BATT}
Upper	10	8	6	4	2
Lower	9	7	5	3	1
Standard	CAN-	GND	USBH+	RS232 TxD	GND
Alternative				J1708-	

	12-way					
Alternative	MIC IN			LINE IN R		
Standard	Analog In	GND	HS Input	Input 4	Input 2	V_{BATT}
Upper	12	10	8	6	4	2
Lower	11	9	7	5	3	1
Standard	LINE OUT L	GND	Input 5	Input 3	Input 1	GND
Alternative			LINE OUT R	LINE IN L		

	6-way		
Alternative			
Standard	LS4	LS2	V_{BATT}
Upper	6	4	2
Lower	5	3	1
Standard	LS3	LS1	GND
Alternative			

Standard Functions

Inputs	Digital Input with rating to 60V, switches on at 4V, switches off at 2V
Analog In	Analogue Input with protection to 60V, clamps at 36V, 8-bit 0-36V
HS Input	Odometer input (use Inputx for direction)
LSx	Low-side switch, 40V protected, rated for 24V, 250mA continuous load
RS232 RxD	DCE RS232 level serial port data signals (DATA from V300 - connect your Receiver to this line)
RS232 TxD	DCE RS232 level serial port data signals (DATA to V300 - connect your Transmitter to this line)
CAN+, CAN-	Bosch CAN bus physical interface
USBH+	USB Host Data +
USBH-	USB Host Data -
VBUS	USB +5V supply
LINE OUT L	Audio Line Out Mono 0-1Vrms
V_{BATT}	Protected vehicle battery supply thermal fused to 400mA (combined for all V_{BATT} pins)

Alternative functions as manufacturing option

J1708 +/-	J1708 serial lines
MIC IN	Microphone Input
LINE IN L	Audio Line In Left 0-1Vrms
LINE IN R	Audio Line In Right 0-1Vrms
LINE OUT R	Audio Line Out Right 0-1Vrms

O/S LED

The O/S LED (**GREEN** LED) serves as an indicator as to the status of the Windows CE boot process. When the device is first powered on, both the status LED and the heartbeat LED flash simultaneously. The second time the O/S LED flashes, the Vector 300 has loaded Windows CE from the specified medium. The timing for this depends on a combination of factors such as what state the device was the last time it was powered off and if the user has indicated via the Debug button whether a new O/S image is to be loaded from the SD card. The second O/S LED flash can take up to 5-10 minutes if the user has specified a firmware upgrade, or could be almost immediate if the O/S is loading from memory. The 3rd flash of the O/S LED indicates that the CE kernel has taken control of the processor and system resources and is now actively processing data. The O/S LED continues to flash ~3 times a second continuously thereafter.

The O/S LED indicates to the user that the device is active and processing data by flashing once every 300 milliseconds. This cadence is set by the Windows CE kernel, ie. if it is flashing, the Windows CE kernel is in

control of the V300.

Heartbeat LED

This LED interface has been designed to be programmable, but this capability has not yet been exposed to a user API.

Debug Button

The debug button presently only has one functionality: firmware upgrades. See the section on Firmware Upgrades for details on how to use this button.

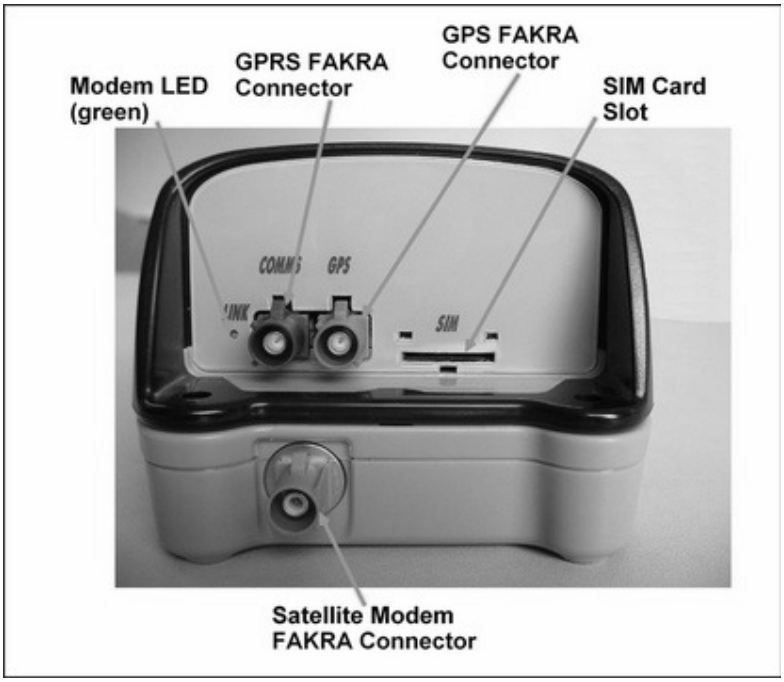
Reset Button

The reset button provides hard reset functionality to the device so that the device may be reset without unplugging the power *and* backup battery. To hard reset the device, hold this button down for eight (8) seconds with the power on. When the bootloader takes over, the green OS LED will stop winking, which indicates the reset has taken effect.

The O/S LED will indicate the steps the device is taking as it proceeds through the boot process.

Back View

The back view of the Vector 300 contains two FAKRA antenna connectors for the Vector 300FM, and three for the Vector 300DM. It also contains a GSM Subscriber Identity Module (SIM) card slot. The Modem LED provides limited status information on the GPRS modem.



NOTE: The Vector 300DM model is pictured above. Without the satellite modem module, the Vector 300FM model is not so tall, and has one less FAKRA antenna connector.

SIM Card Orientation

The SIM card must be properly oriented in order to ensure correct operation. The gold connectors should be inserted facing upwards and closest to you, as shown:

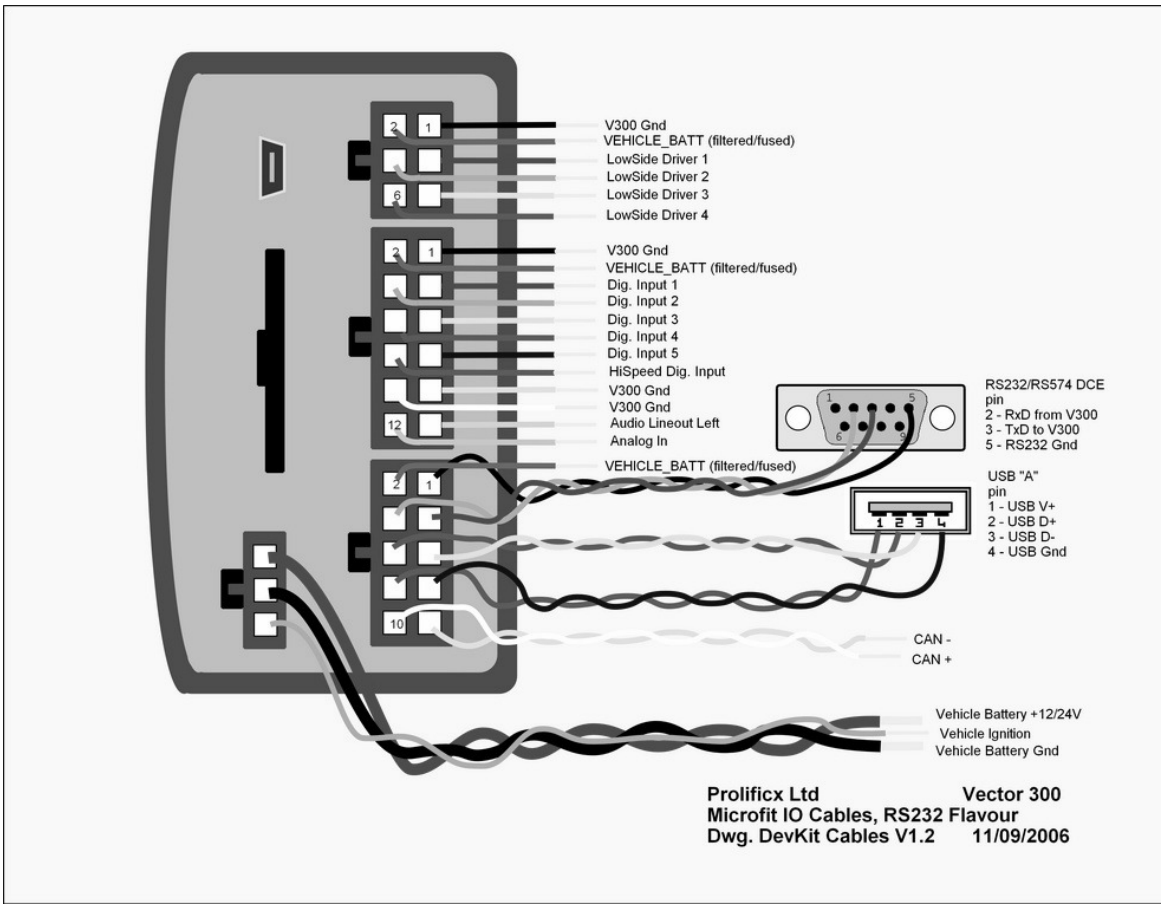
Modem LED

The modem LED currently does not have any extended functionality beyond indicating that the modem is working.

Development Kit

The Vector 300 Development Kit is designed to make testing and integrating the capabilities of the Vector 300 as easy as possible. The following sections outline required and included components as well as optional components.

For a full resolution copy of the following diagram, click on the image. Details of what is provided in your Vector 300 Development kit are also available in the Vector 300 Installation Manual



Required Components

- 12V DC Bench Power Supply (included)
- USB Type B Cable (included)
- Visual Studio .NET 2005 (CF2.0)
- .NET Compact Framework 2.0 and Service Packs
- Prolificx Libraries for Framework 2.0
- ActiveSync

Optional Components

- Secure Digital Card (included)
- GPS Antenna (included)
- GSM Antenna (included)
- Satellite Antenna (included for Vector 300DM kits only)
- Data-enabled SIM card This is specific to each telco/network

Operating System

The Vector series devices runs the Microsoft Windows CE 5.0 operating system. Windows CE is like traditional Windows operating systems in terms of look and feel; it has a Start button and exposes a graphical user interface like Windows XP, but in a compact fashion. Windows CE manages memory,

processor, file system, etc. in scaled down ways as well.

Unlike Windows XP, Windows CE is built via customized operating system builds that Prolificx provides -- think of these as firmware. It has a few other important differences that will be important to you as a consumer of the end product; they are outlined below.

For more information on Windows CE, see the Windows CE Homepage. Specific requirements for developing for Vector 300 are outlined below

Firmware

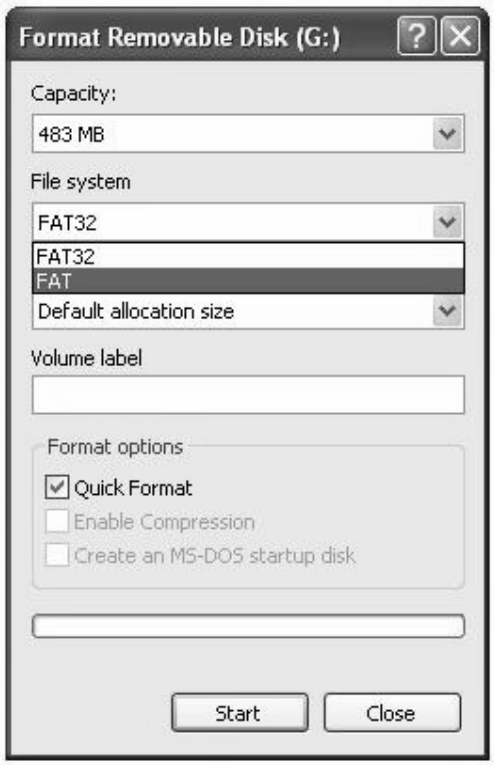
New firmware images are available from Prolificx.

Upgrade Instructions

The Windows CE operating system image ("firmware") can only be upgraded using a SD card. To begin, you will need a blank SD card and an SD card reader you can attach to your PC. The SD Card must be formatted with **FAT**, *not* **FAT32**, which is the default file system for the Windows SD card driver. The maximum SD card size supported by the operating system is 2 Giga byte.

In order to do this, perform the following steps:

1. Insert the SD card into your reader and connect it to your PC
2. Determine which drive letter in Windows Explorer (for example, G:\) corresponds to the SD card drive
3. Right click the drive in Windows Explorer, and select "Format"
4. Set the file system to FAT, as shown below, check Quick Format to make it go faster
5. Start the format



To upgrade the firmware, follow the steps below.

1. Download the latest firmware image from the link above
2. Extract NK.bin (O/S image) and EBOOT.bin (Bootloader) only onto the SD Card
3. Insert the SD Card into the Vector 300, gold connectors downward and away from you
4. Depress the debug button (see External Ports diagram), and hold it
5. Plug the power into the Vector 300
6. The red and green lights will flash simultaneously, continue holding the debug button
7. When the red LED begins flashing once every second (and the green LED is off), the process has begun, now you can release the debug button. This will continue to flash throughout the upgrade process. **If there is an error, the red LED will light continually, the error code will blink at 3 second intervals, indicated by the green LED.** For example, if there was an SD card read error, the green LED will blink twice, wait 3 seconds, blink twice, wait 3 seconds, etc.
8. When the upgrade is complete, the red LED will stop flashing and the unit will reboot. This will take about 10 minutes
9. Upon reboot, the red and green LEDs will flash simultaneously as is normal (ref. Operating System LED)
10. Once the boot process is complete, the red LED will intermittently flash as is normal and ActiveSync will indicate a connection if the USB cable is connected

If you receive an error indication, refer to the table below for its meaning.

Error Code	Meaning
0	No error detected; success
1	Error unknown
2	Error reading SD card
3	Incorrect SD card format, FAT (not FAT32) required
4	Unknown error with SD Card
5	EBOOT.bin file not found
6	NK.bin file not found
7	Error writing to FLASH

Bootloader

The bootloader handles the device operations from power on until Windows CE takes control. The boot process has different startup times and stages depending on the state of the device at last suspend / power off. The Operating System LED indicates the stage of the boot process.

Cold Boot

"Cold boot" means that the device is starting from a hardware reset, or is being started from a no-power (power on) state. This process typically takes 15-20 seconds, depending on how many device drivers are loaded.

Simply cycling power by removing/replacing the Vehicle Battery lead is not sufficient for most units, since the on-board battery will keep the Vector 300 alive for at least short periods of time.

Soft Boot / Resume

A "soft boot" means that the device is resuming from a suspended, low power state back into full operation. Conditions which take the device in and out of this state vary depending on software configuration (ref. Power States). A resume operation typically takes only 1 or 2 seconds.

Power States

The Vector 300 has four system power states. The system can enter Suspend via the GUI

but the states are really for field applications and should be set programatically. Use the SetSystemPowerState command (see Function reference for details) to do this.

- On - Everything is operating normally.
- Shutdown - processor powered down, minimum power consumption, can only be woken on ignition or movement
- Suspend - processor stopped, peripherals are powered on to enable wakeup
- Reset - this resets the V300 fully, i.e. does a "cold boot"

Backup Battery

The Vector 300 has two backup battery options, the standard 150mA rechargeable and an optional 1500mAhr rechargeable battery.

The battery backup is provided for two reasons - firstly to let the operating system do an orderly shutdown in the event of power loss or deep voltage dips during starter motor cranking, and secondly to provide sufficient time to send notifications of power failure or accident in the event of power being suddenly removed.

As the usage of the backup battery during normal operation is minimal, one of the design trade-offs made in Vector 300 is a long recharge time (to minimise power supply requirements). For the larger battery a full recharge can take 2-3 days, as the charging current is limited to around 50mA. Even the standard battery requires upwards of 3 hours to fully recharge a depleted cell. Note also that the charging is only active whilst the ignition is ON so as not to discharge the vehicle battery when ignition is off. Charging time is also dependent on temperature. For example, battery charging is disabled below -10C as LiPo batteries cannot be charged below this temperature or above 60C

Operating time on the optional 1500mAhr backup battery is up to 2 hours. This varies according to temperature. At -35C then the backup battery capacity is reduced to around 15 minutes. This should still be enough to make an emergency GPRS transmission.

Figures for the standard 150mAhr battery are 10% of those for the optional larger battery. Under no circumstances should it be expected that enough power remains in the standard battery for any emergency GPRS transmission.

Wakeup Sources

The Vector 300 can be placed into a suspend state or shutdown state programatically as outlined in the Function Reference.

When in suspend, the following conditions will wake the Vector 300 back to the "On" state:

1. State change from low (0V) to high (>7V) on the ignition line
2. Accelerometer (if enabled, can be enabled via software, see Function Reference)
3. Incoming calls/message from modem (if enabled, see Function Reference)
4. Two of the Digital Inputs (if enabled, see Function Reference)
5. Real Time Clock (if enabled, see Function Reference)
6. GPS Antenna Fault

The following conditions can be programmed to wake the Vector 300 from a "Shutdown" state:

1. State change from low (0V) to high (>7V) on the ignition line
2. Accelerometer (if enabled, can be enabled via software, see Function Reference)
3. Real Time Clock (if enabled, see Function Reference)

File System

The Vector 300 does not have a traditional hard drive; this is to keep costs, power consumption and size down. The Vector 300 uses FLASH memory, which is persistent storage designed to retain data at power loss, but not meant for heavy amounts of I/O or large amounts of data transfer. The FLASH memory in the Vector 300 makes use of a file system manager specifically designed to minimize/distribute wear to the use-sensitive FLASH memory.

Although the underlying hardware implementation is different, the application developer will read and write to FLASH memory the same way they would on a traditional Windows machine with a hard drive. The .NET Compact Framework uses the same functions as the traditional framework to read and write from the file system.

READ FIRST: FLASH memory has a limited lifetime based on number of writes. The FLASH memory used in the Vector 300 has a minimum write threshold of 100,000 times; beyond that reliability and performance are not guaranteed. Applications should be designed/coded keeping this limitation in mind at all times.

For example, if planning for data logging on Vector 300, implement a simple buffer. Hold the data in memory for a period (eg. an hour or two), and then flush it to disk only on a schedule only as frequent as is required by your specifications. This practice will significantly extend the lifetime of the Vector 300's flash memory.

SDRAM

The Vector 300 uses SDRAM as its volatile working memory to run the operating system. SDRAM is inherently faster than FLASH memory and does not have write speed or lifetime limitations. The hardware implementation is again seamless to the developer; code to handle working memory can be written traditionally.

Development Environment

Prolificx developed its libraries and applications on the .NET Compact Framework 2.0, with Visual Studio 2005. **This is the official platform we *highly recommend* and support**. Where possible, we will make every attempt to accommodate customers' requests for support in eMbedded Visual C++ as well as the .NET Compact Framework 1.0 and Visual Studio .NET 2003, but support for these platforms and frameworks will not be as comprehensive or detailed.

If you haven't developed applications for Windows CE before, you must read the following sections first to understand the process by which applications are created, deployed, and executed on mobile devices.

ActiveSync

The first thing you need to do with your Vector 300 is establish an ActiveSync connection. ActiveSync acts as the communications conduit between your development PC and the Vector 300. It handles all the PC<->device communications so you don't have to. To set up an ActiveSync connection, follow these instructions.

1. Make sure you have downloaded and installed ActiveSync 4.2 or later
2. Plug the included USB B cable into the Vector 300's ActiveSync USB port and into your PC
3. You should see the following dialog. **It is best to select "No Partnership"** because the Vector 300 will attempt to create a new partnership each time you connect to it. Furthermore, you have no need to sync Outlook or files via ActiveSync -- the Vector 300 does not have Outlook; in fact if you do it may disrupt your host system's clock and Outlook.

4. You should now see a Window indicating you are connected. You can now open the remote display application to view the Windows CE GUI, copy files to/from the device in Windows Explorer, or deploy applications in Visual Studio.

Remote GUI Display

Now that you've set up the communications channel between your PC and the Vector 300 via ActiveSync, the next step is to open a remote display session to the actual Windows CE GUI. The Windows CE GUI looks a lot like traditional Windows -- it has a Start button, a desktop and program groups. Set up a Remote Display connection to the Vector 300 as follows:.

1. Make sure your ActiveSync window says "Connected" as outlined above.

2. Download the Remote Display Client and unzip **ALL** files to a directory of your choosing.
3. Run StartRemoteDisplay.exe
4. You should see your Windows CE GUI as pictured below!

Microsoft Tools Required

First, make sure you've downloaded, installed, and created your first ActiveSync connection - see above. Now

1. Download and install the .NET Compact Framework 2.0
2. Download and install .NET Compact Framework 2.0 SP1
3. Download and install .NET Compact Framework 2.0 SP1 patch

Note that due to incompatibilities, Compact Framework 1.0 is no longer supported.

Prolificx Libraries

First, make sure you've done the following:

- Downloaded, installed, and created your first ActiveSync connection
- Downloaded and installed the .NET Compact Framework, including Service Packs
- Bought and installed Visual Studio

In order to make use of the Prolificx namespace, the DLLs must be copied onto your device. To develop and deploy applications in C#, you need two files:

1. Prolificx.dll
2. V300.dll

Now download the Prolificx .NET CF 2.0 libraries. When you've downloaded those two files, keep them in a handy place so you can copy them into your development directory and reference them as outlined below in the "Creating a Project" section. Now that you've got the files, you'll need to place them in the Windows directory of your Vector 300 (you only need to do this once) so that you don't have to manually copy them over as you "deploy" your applications (more on this later). Visual Studio unfortunately does not do the

copying of the libraries for you, so the best way to make it easy on you is to copy the files into the Windows directory once you've decided on your development tools of choice. To do this, navigate in Windows explorer (with the Vector 300 connected) to the Mobile Device. You can explore it just as though it were a hard drive on your development PC, and move files back and forth as well. Take the two files you've downloaded from above (Prolificx.dll, V300.dll) and copy them into the Windows directory as shown.

We will be including them in the Windows directory in future firmware images, but for now you'll need to copy them over manually.

Now that have the files on your device, you are ready to create your first project.

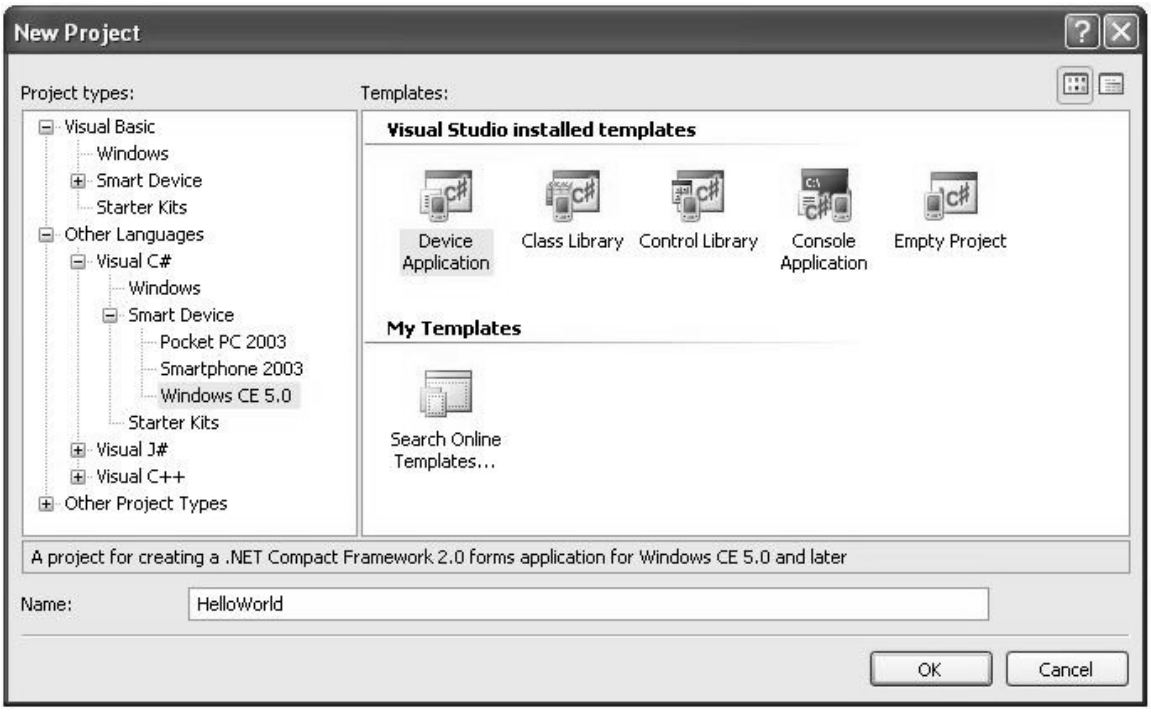
Creating a Project

Creating your first Windows CE application in Visual Studio is an easy process. First, make sure you've done the following:

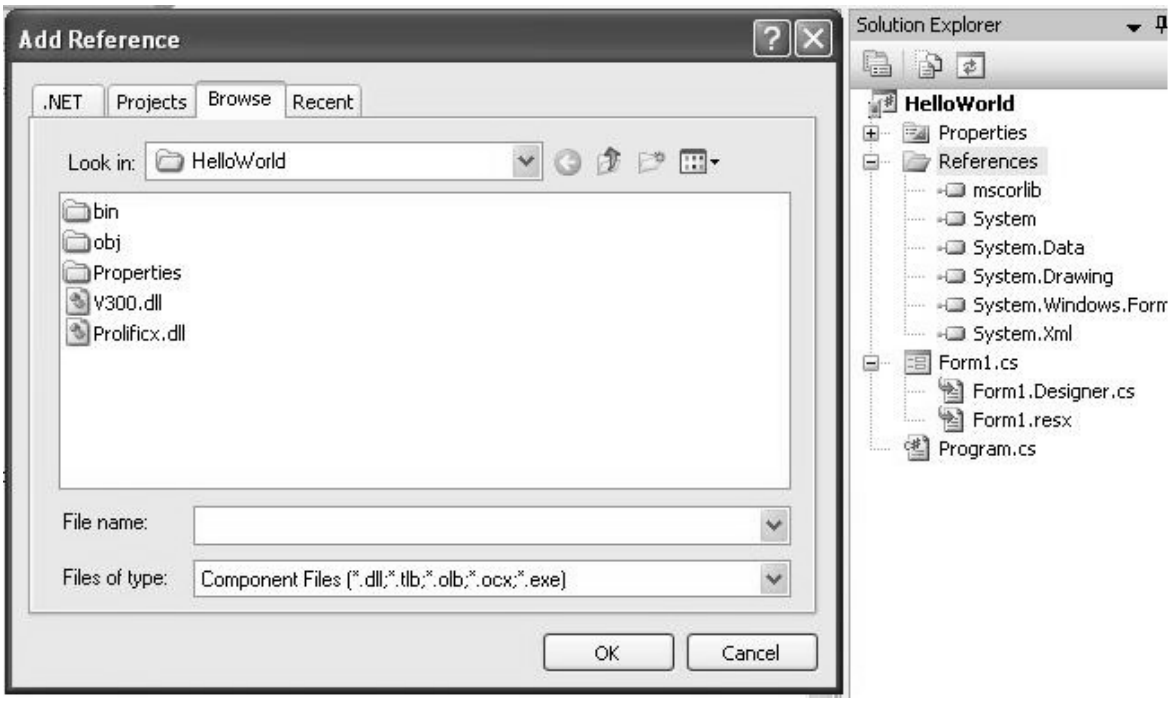
1. Downloaded, installed, and created your first ActiveSync connection
2. Downloaded and installed the .NET Compact Framework, including Service Packs
3. Bought and installed Visual Studio
4. Downloaded the Prolificx libraries of your choice, placed them in a handy place on your hard drive, and made copies of them in the Windows directory on your device

If and only if all these things are in order, you can proceed. For demonstration purposes, we'll use the .NET Compact Framework 2.0 and Visual Studio 2005, but the steps are essentially identical for Visual Studio 2003.

1. Open up Visual Studio, and select File > New Project in the menus or click Create: New Project on the start page
2. You'll be presented with the following dialog. Set the options as shown -- a new C# Windows CE 5.0 Device Application. Set a name for your application, and click OK.



3. Visual Studio will do its magic and create the project and a starting form with you. You can create and draw controls visually, which will cut down your design time significantly.
4. The next step is to set up references to the Prolificx namespace. Visual Studio hasn't actually saved the project to disk yet, so let's go ahead and do File > Save All. This will create the project on disk, and allow you to choose the location. Projects are typically created under the "Visual Studio Projects" folder in My Documents.
5. Remember when you were told to keep those DLLs in a handy place on your hard drive? Go find them, and copy them into the development directory you just specified and created (the one with files like Form1.designer.cs and so forth in it, not the parent directory with just the Solution Files).
6. Now you're ready to create references in your project to the DLLs. Right click the references folder under Solution Explorer, click Add reference, and navigate to the directory where you copied your DLLs into, and add Prolificx.dll as a reference as shown. **Don't add V300.dll as a reference, Prolificx.dll points appropriately to V300.dll**



7. You're now ready to begin writing code. If you want to do it yourself, create a button on your Form1 (in design view) and changes its name to btnHelloWorld. You can then copy the code below into your project (replace everything in Form1.cs code with this). Alternatively, you can just download the source

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace HelloWorld
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnHelloWorld_Click(object sender, EventArgs e)
        {
            MessageBox.Show("Hello World!");
        }
    }
}
```

8. Your application is now ready! Let's move on to getting it onto the device and executing it.