

## Quick Start Guide to Building with iso7816\_vcc.c January 27, 2005

**Web:** <http://www.smartcache.net/>  
**E-mail:** [info@smartcache.net](mailto:info@smartcache.net)

### Introduction

iso7816\_vcc.c implements an interface between an RS-232 serial port and an ISO 7816 smart card, and is targeted for the Atmel AT90S2313 microcontroller in a development terminal. The source code is written entirely in C using the WinAVR tool suite. WinAVR is a suite of executable, open source software development tools for the Atmel AVR series of RISC microcontrollers hosted on the Windows platform. It includes the GNU GCC compiler (avr-gcc) for C and C++. WinAVR is provided free of charge.

WinAVR includes many popular UNIX utilities ported to Windows. The build environment in WinAVR uses the `make` utility and makefiles, just as in the UNIX environment. Although the `make` utility can be run from the command line, WinAVR includes Programmers Notepad, a GUI application that is used to edit source files and invoke tools such as `make`.

The `iso7816_vcc.c` source code has been analyzed with PC-lint, and has 0 warnings at warning level 4 (using an appropriate PC-lint options file that inhibits warnings of no concern). Programmers Notepad can be configured to run PC-lint, and PC-lint can be configured so that clicking on a warning or error message in the Output pane of Programmers Notepad takes you right to the source line. PC-lint options files for `avr-gcc` and a configuration guide for Programmers Notepad will be available in a separate download.

If you are serious at all about developing C or C++ code, or if you just want to improve your programming skills, I highly recommend purchasing PC-lint. PC-lint can be configured to run on just about any C or C++ source code, regardless of the compiler or target processor. Options files for many popular compilers are included. I have used PC-lint to analyze C and C++ source code for many compilers, including VC++ for Windows, Borland for Windows, `gcc` for Linux, Watcom for QNX 4.25, `gcc` for QNX Neutrino RTOS 6.x, CVI for Windows, Metrowerks for Macintosh, and now `avr-gcc` for Atmel microcontrollers. For more information on PC-lint, visit the Gimpel web site at:

<http://www.gimpel.com/>

After downloading `iso7816_vcc_c.zip`, perform the following steps to get started with building and using `iso7816_vcc`.

### Requirements

`iso7816_vcc` requires the following:

- Windows OS
- WinAVR
- A development terminal (such as the SD7816U – ULTRA USCT available from SDLOGIC at <http://www.sdlogic.us/>)
- An application such as SmartCache or SoftCard

### Set up directories

1. Create a directory where you will extract `iso7816_vcc_c.zip`. The directory `C:\Projects\Atmel` will be used in this example.

2. UnZIP `iso7816_vcc_c.zip` into `C:\Projects\Atmel`. This will create the directory `C:\Projects\Atmel\iso7816_vcc_c` containing the following files:

```
EULA.txt
iso7816_vcc.c
iso7816_vcc.pnproj
Makefile
```

3. If you will be using XpAtmel to program your development terminal, create a directory named `XpAtmel` in the `Atmel` directory (i.e., create `C:\Projects\Atmel\XpAtmel`). Copy `XpAtmel.exe` and `atmeldrv.sys` (provided with the SmartCache installation) to the `C:\Projects\Atmel\XpAtmel` directory.
4. At this point, your directory structure should look like this:

```
Projects
  Atmel
    iso7816_vcc_c
    XpAtmel
```

### Install WinAVR

If you do not already have WinAVR installed, download it from the link on the WinAVR web site and install it.

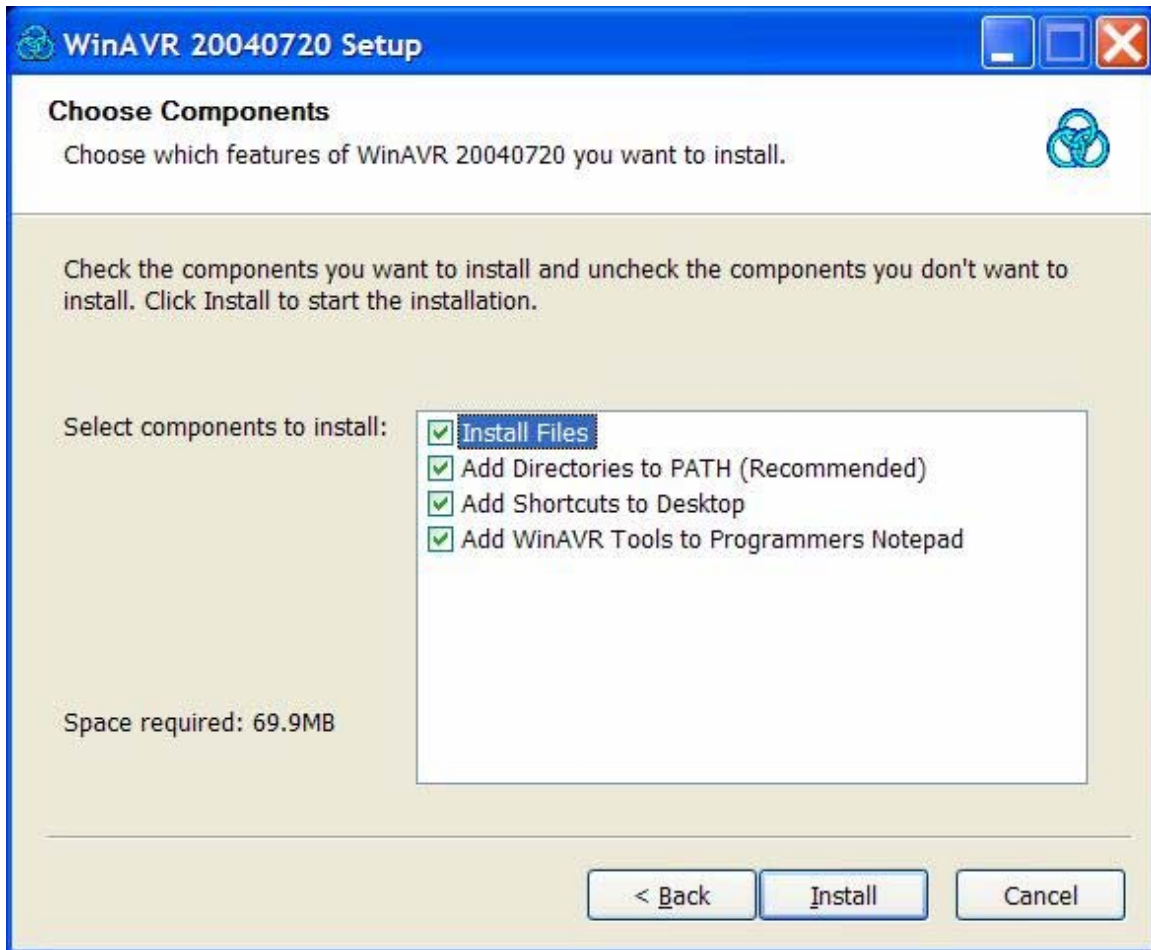
The current home page for WinAVR is:

<http://winavr.sourceforge.net/>

The current download URL for WinAVR is:

<http://sourceforge.net/projects/winavr/>

**NOTE:** In the **Choose Components** dialog of the WinAVR install, shown below, the only component you should consider optional is "Add Shortcuts to Desktop"; all other components should be checked. The WinAVR install opens the Programmers Notepad application. After pressing the **Finish** button in the WinAVR install, it is a good idea to close Programmers Notepad and relaunch it before building `iso7816_vcc` or any other projects, to insure that it picks up the updated PATH environment variable.



### Build iso7816\_vcc\_c

1. Open Programmers Notepad.
2. Select **Open Project(s)...** from the **File** Menu.
3. In the dialog that opens, navigate to the C:\Projects\Atmel\iso7816\_vcc\_c directory and open iso7816\_vcc.pnproj.
4. Double-click on iso7816\_vcc.c in the Projects pane to open the file for editing.
5. Select **[WinAVR] Make All** from the **Tools** Menu. This is equivalent to executing make in the iso7816\_vcc\_c directory at a command prompt. This will create the files iso7816\_vcc.hex and iso7816\_vcc.eep, among others, in the iso7816\_vcc\_c project directory. It will also create the files iso7816\_vcc\_c.exe.program.hex and iso7816\_vcc\_c.exe.eeprom.hex in the C:\Projects\Atmel\XPAtmel directory. The last lines of text in the Output pane should like this:

```
Errors: none
----- end -----

> Process Exit Code: 0
```

### Program with XPAtmel

1. If you will be using XPAtmel to program your development terminal, the XPAtmel files for iso7816\_vcc are automatically built and placed in the C:\Projects\Atmel\XpAtmel directory by the makefile (i.e., when you select **[WinAVR] Make All** from the **Tools** Menu in Programmers Notepad).

2. Open `XpAtmel.exe` in the `C:\Projects\Atmel\XpAtmel` directory. In XPATmel, select `iso7816_vcc_c` in the list and select the COM port that your development terminal is connected to. Press the **Program** button. Exit XPATmel.
3. Cycle power on your development terminal.
4. Now your development terminal is programmed to work with SmartCache or SmartCard.

### Program with STK500

1. If you will be using the Atmel STK500 board to program the Atmel microcontroller in your development terminal, perform the following steps.
2. Remove power to the development terminal.
3. Remove the Atmel microcontroller from the development terminal.
4. Place the Atmel microcontroller in the 20-pin red DIP socket on the STK500.
5. Open Atmel's AVR Studio and open the STK500 programming dialog.
6. When you open this dialog, AVR Studio will attempt to detect the STK500. If detection is successful, move to the next step. Otherwise, close the dialog, check serial port and power connections to the STK500, and try again.
7. Select AT90S2313 in the Device combo box.
8. Check the checkbox labeled **Erase Device Before Programming**.
9. In the Flash group box, select `iso7816_vcc.hex`, located in the `iso7816_vcc_c` project directory.
10. In the Flash group box, press the **Program** button and wait for programming to complete.
11. Uncheck the checkbox labeled **Erase Device Before Programming**.
12. In the EEPROM group box, select `iso7816_vcc.eep`, located in the `iso7816_vcc_c` project directory.
13. In the EEPROM group box, press the **Program** button and wait for programming to complete.
14. Remove the Atmel microcontroller from the STK500 and put it in the development terminal. **All pins must be seated.**
15. Apply power to the development terminal.
16. Now your development terminal is programmed to work with SmartCache or SmartCard.

### Changing the PWM Pulse Width

The PWM TOP value is set using the `top` member in the `EE_PWM` struct constant in `iso7816_vcc.c`. You can change this value to anything between `0x00` and `0xFF`. The pulse width in PWM, and consequently the DC voltage level applied to the ISO VCC and CLK contacts, is directly proportional to the TOP value.

### Notes

1. Read the HTML documentation that is installed with WinAVR.
2. I suggest downloading the AT90S2313 datasheet from Atmel, especially if you are a programmer.
3. Comments, suggestions and bug reports are always welcome.