# AVR Programming Methods

Dean Camera

September 2, 2012

**********



For more tutorials, project information, donation information and author contact information, please visit www.fourwalledcubicle.com.

# Contents

# Chapter 1

# AVR Programming Methods

There are many ways to program AVR microcontrollers. Since many people ask about different ones at one time or another, I thought I'd outline them here so that their questions can be answered quickly and efficiently. Please forgive me if I miss a method or make an error.

## 1.1 In System Programming (ISP)

| | |
|---|---|
| **Supported By:** | Most MEGA and TINY devices |
| **Supported Programmers:** | AVRISP MKI/II, JTAG MKII, STK500, STK600, Dragon, AVRISP clones, AVR910 Programmers, AVRONE |

In System Programming is perhaps the most common method of programming the flash, EEPROM, fuse and lockbytes of the entire AVR line. ISP can program AVRs at extremely high clock rates (assuming the target AVR is running at a high frequency and the programmer supports it) and is the method of choice for almost all AVR hobbyists. There are many, many AVRISP clones and AVR910 programmers on the market in addition to simple do-it-yourself dongles which connect to your computer's parallel port.

Recent new dongle designs may use the computer's serial port, however anecdotal evidence has said that this method is *extremely* slow due to technical limitations.

ISP requires that the target AVR is running at a clock rate of at least four times that of the ISP clock. This is a common pitfall and a source of confusion to many new to AVRs.

## 1.2 JTAG

| | |
|---|---|
| **Supported By:** | AVR32 and most large pin count MEGA and XMEGA devices |
| **Supported Programmers:** | JTAG-ICE, JTAG-ICE MKII, JTAG-ICE3, Dragon, JTAG-ICE clones, AVRONE, STK600 (programming only) |

Technically JTAG is a debugging system, not a programming method. Still, the JTAG interface allows for the programming of an AVR which supports it.

JTAG is an in-system debugging tool which allows you to manipulate and examine the status of a supported AVR while it is running in a circuit. JTAG allows the user to stop execution at any time, the manipulation of the AVR's internal registers and much more.

The official JTAG-ICE units from ATMEL have been superseded by the JTAG-ICE MKII, which supports the newer and more widely supported across the AVR range DebugWire debugging protocol as well as programming via the ISP method (see above).

JTAG-ICE clones are available for low prices, however their limited compatibility with only a handful of AVRs limits their usefulness. Regardless of this, if your AVR supports the JTAG interface the JTAG-ICE remains a very nice and effective debugging method and programmer.

## 1.3 DebugWire (dW)

**Supported By:** Most low pin count MEGA and most TINY devices
**Supported Programmers:** JTAG-ICE MKII, JTAG-ICE3, Dragon, AVRONE

Again DebugWire is a debugging rather than a programming interface, but can be used to load in programs into supported AVRs. The dW interface uses a single AVR pin (the /RESET line) for all communications, making it ideal for the low-pin count AVR devices.

## 1.4 Pre-programmed Bootloader

**Supported By:** Most AVR devices
**Supported Programmers:** N/A

Again technically not a direct programming method. A bootloader is a small AVR program which sits in a user-settable reserved section of the regular flash. Bootloaders make use of the flash self-modification features available in the newer AVRs to allow the AVR to program itself via program data loaded from an external source. Bootloaders may source their data from any location (eg external dataflash or SD card) however by far the most common type of Bootloader communicates with a PC via the AVR's RS-232 (serial) port.

Bootloaders are limited in that they do consume flash space (limiting the size of the flash available to the AVR's application) and they are unable to change the AVRs fusebits.

Bootloaders are widely available on the internet for download, but they suffer from a "chicken and egg" problem; you need another type of programmer listed here to program in the bootloader in the first place. This is usually solved by the construction of a simple parallel port dongle (See ISP section) or by the purchase of an AVR already preloaded with a bootloader (eg the AVRButterfly board).

## 1.5 High Voltage Parallel Programming (HVPP)

**Supported By:** Most MEGA devices
**Supported Programmers:** STK500, STK600, Dragon, Homebrew Dongles, AVRONE

High Voltage Parallel Programming is a method of programming which is rarely used, because of the hassle it requires to set up. Despite this, HVPP programming is commonly used to "resurrect" AVRs whose fusebits have been mis configured via another programming method.

Both the STK500 and the Dragon supports HVPP. During HVPP, the target's /RESET pin is raised to the unusually high value of 12V which engages the internal parallel programming circuitry. The /RESET pin is the only pin of the AVR (on HVPP supported AVRs) which can be safely raised to this level.

## 1.6 High Voltage Serial Programming (HVSP)

**Supported By:** Many TINY devices
**Supported Programmers:** STK500, STK600, Dragon, Homebrew Dongles, AVRONE

HVSP is similar to HVPP, except the data transfer is performed serially rather than in parallel. This is the alternate programming method used on many TINY series AVRs who lack enough pins for HVPP.

## 1.7 Program and Debug Interface (PDI)

**Supported By:** Most XMEGA devices
**Supported Programmers:** STK600, JTAG-ICE3, AVRONE

PDI is the new programming interface based on the debugWire protocol, for the XMEGA line of AVRs. It's not currently used on any other 8-bit AVR microcontrollers. PDI provides a low pincount method of both programming and debugging XMEGA devices in-system.

## 1.8 Tiny Programming Interface (TPI)

**Supported By:** 6-Pin TINY AVRs (ATTINY10, etc.)
**Supported Programmers:** STK600, JTAG-ICE3, Dragon, AVRISP MKII

TPI is a very tiny programming interface for the newer TINY line of AVRs with limited pins, like the 6 pin ATTINY10. Like dW, TPI uses the device's /RESET line as part of the communication interface, but there the similarity ends. Since the pint-sized TINY AVRs lack a on-chip debugging circuit, the TPI protocol uses a new programming interface of three pins, in a half-duplex protocol. Because the /RESET line needs to be raised to +12V for programming when the device's RSTDSB pin is set, this is currently only supported by the newer STK600 programming board.

## 1.9 aWire (aW)

**Supported By:** Most AVR32 devices
**Supported Programmers:** STK600, JTAG-ICE3, AVRONE

aWire is a three wire programming protocol, similar to DebugWire, that is designed to program AVR32 devices in-system without conflicting with the resources on the target board.

# Chapter 2

# AVR Programming FAQ

## 2.1 Which is the best method?

There is no universal "best" method. ISP programming is simple and extremely popular, however all the above methods will work. The two high voltage programming modes (whichever is applicable to your device) are the most feature rich, as they allow for the repair of an AVR which has had its fuses misconfigured. However, those methods are a pain to set up, hence the reason most users go with with ISP.

## 2.2 I've made a parallel port dongle. Can I use it with AVRStudio?

I'm afraid not. AVRStudio cannot interface with any "dumb" dongles - it requires a smart programming device - containing a microcontroller itself - to decipher the communication protocol it sends. Simple dongles without a microcontroller must be "bit-banged" (ie. the appropriate signals simulated through the dongle via the computer) itself.

## 2.3 So my dongle's useless then?

No. You can still program through a home made dongle with a third party programming software tool. AVRDude is a good, known, free command line utility - and it comes included with the WinAVR package.

## 2.4 What are my options if I want my programmer to work with AVRStudio?

Choose a programmer that uses an AVRStudio-supported protocol. This can be the simple "AVR910" protocol (deprecated) or a custom implementation of the protocol used by the STK500/AVRISP. Note that these programmers require a micro controller in them, leading to a catch-22 situation. This may be solved by having the programmer's AVR pre-programmed at time of purchase with the appropriate firmware, or by having the AVR pre-programmed with a bootloader.

## 2.5 Ok, I want to use a bootloader. How do I get it in there in the first place?

To use a bootloader in an AVR, you first have to have the bootloader programmed in. If you do not have an existing programmer (even a simple dumb dongle will suffice for the initial programming), you can alternatively purchase AVRs pre-programmed with a bootloader from several suppliers.

Atmel also manufactures the Butterfly demo board, whose MEGA169 AVR comes pre-loaded with an AVR-Studio compatible bootloader.

## 2.6 Help! I've messed with the fuses and knackered my AVR while using ISP!

The most common mistake is changing the clock selection fuses to an invalid setting. Try putting an external clock on the AVR's XTAL1 pin and see if that helps.

Failing that, if possible use one of the high-voltage methods. These will fix any misconfiguration, including ones involving the clock source as the high-voltage methods provides its own clock to the AVR for programming.

## 2.7 How do I interface with my programmer?

Which software you use to interface with your programmer depends on the type of programmer you are using.

Simple "dumb" dongles require third party software, such as PonyProg or AVRDude. These may be command line or GUI tools — look around on the web and you will fine one to fit your needs.

Programmers and bootloaders based on the AVR910 protocol can be used within AVRStudio. From the Tools menu, select the "AVRProg" option to open up a GUI screen to interface with your programmer. As an alternative, third party tools such as AVRDude are also AVR910 compatible.

Official tools are tightly integrated into AVRStudio, especially in the case of the debugging variants (JTAG/Dragon/etc). From the AVRStudio Tools menu, select the "Program AVR..." submenu and click the "Connect" item. From the new window, select your tool and its connection interface and click ok.

As is the case with the dumb dongles and AVR910 programmers, the official tools may also be used with third party programming software.