Atmel Startup 1: Atmel Studio and Programmer

M. A. Parker, Angstrom Logic, LLC, Copyright 2015-07-28

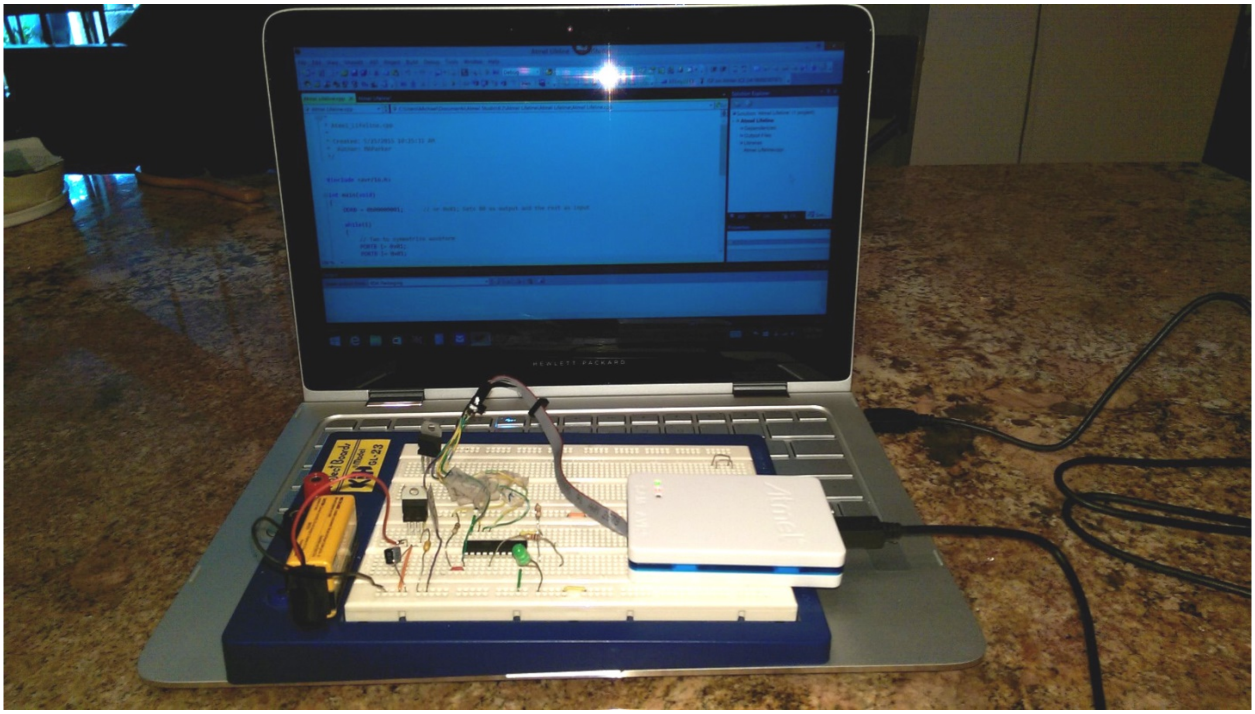


Figure 1: Laptop, programmer and experimenter's breadboard

Installing Atmel Studio (AS) and a programmer should be the first steps [0] on the path to using an individual Atmel AVR Microcontroller unit MCU [1] as opposed to the MCU-on-a-board such as Arduino [2]. Here starts the journey, albeit rather short, through a series of 'Startup' Instructables [0] focusing on 'getting started' with the Atmel MCU. Along the way, the series (Startup 5) shows how to construct the 'Lifeline' to fix wrongly set fuse bits – a simplest of projects but useful to the expert and makes a nice gift. This initial Startup describes the setup of the Atmel Studio and Programmer and the construction of an adapter cable for connecting the programmer to an experimenter's board. Startup 2 describes the construction of a 'platform' circuit on an experimenter's breadboard used to set the internal fuse bits of the MCU. The circuit will be used for the remaining three Startup Instructables. Startups 3 and 4 use the platform to demonstrate 'Blinky', the blinking LED, and to describe the 'secret' life of the Atmel PORT, PIN and DDR registers and pull-up resistors.

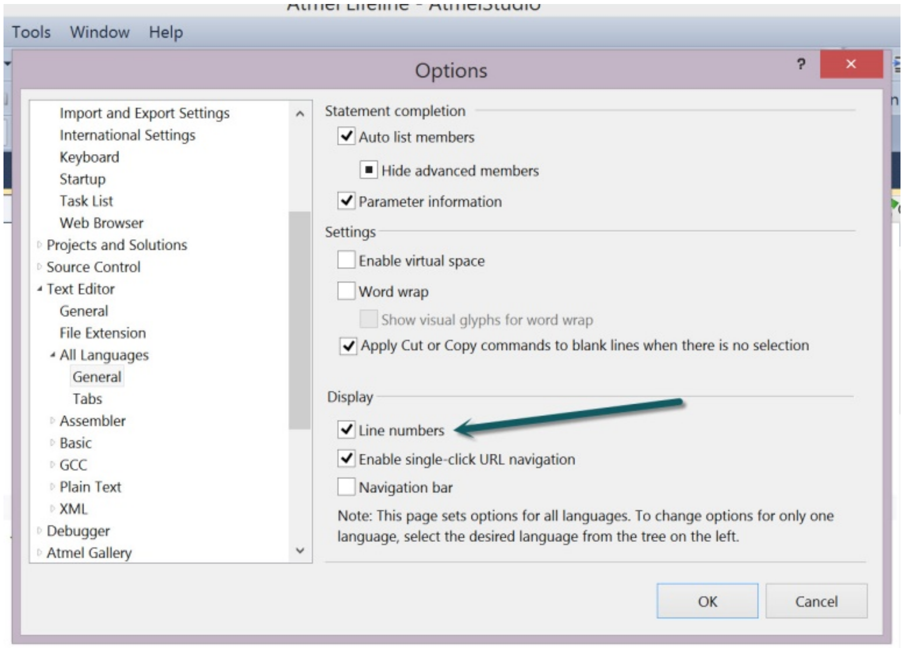
We began using the Atmel MCUs in the early 2000s in order to create Data Acquisition units for experiments with IC fabrication and for demonstrating a unique multichannel noise communicator. Similar to other people, we had to learn certain well published 'secrets' of the Atmel AVR

Microcontroller including, at the most basic level, the Data Direction Register DDR, PORT versus PIN and the USART data format (inverted). Fortunately, a consulting firm supplying our MCU needs, also provided assistance for 'getting started' and pointers on the 'secrets'. While planning to publish some MCU projects, it seemed reasonable to pass along the favor by providing some 'Getting Started' guides, although in considerably more detail, so that other interested people might benefit.

This Instructable provides an introduction to using a single Atmel MCU. Although we start with the ATTiny2313A [3], the discussion applies to the popular Atmel AVRs (8bit) such as the ATMega328p (of Arduino fame). With the single Atmel MCUs, the builder controls all the options and expends much less cash than for the larger and more expensive Arduinos [2]. The 2313A have a price of about $1 each and the budding experimenter will be much less traumatized if a couple should go up in smoke. Future articles will center on other Atmel MCUs such as the ATMega328p and the SAM line. On the other hand, the remarkable Raspberry PIs [4] run Linux with plenty of USB peripherals but uses an event driven architecture that limits real time data acquisition.

The Atmel Studio 6.2 (AS6) software provides an Integrated Development Environment (IDE) which, among other features, (i) allows the user to type programming statements, (ii) compiles the programming statements into proper MCU syntax (i.e., hex code), (iii) transfers the MCU code to a programmer usually attached to a USB port, and (iv)saves/retrieves programs from disk. AS6 will function with any of the Atmel MCUs including AVR (i.e., 8 bit) and SAM (i.e., 32bit). The programmer generates the necessary signals onto the MCU's Serial Peripheral Interface SPI pins that engage and synchronize the MCU's internal 'In-System Programmer ISP' [5]. In short, the programmer transfers the code to the MCU. Atmel Studio provides C and C++ compilers which convert programming statements into MCU syntax/Hex code.

This instructable includes notes on tools toward the end and some references for the programmer. However, the majority of the reference/tutorial links for electronic components and C/C++ will be found over the next several installations of the Startup series where they will be used.



Statement completion

☑ Auto list members

■ Hide advanced members

☑ Parameter information

Settings

☐ Enable virtual space

☐ Word wrap

☐ Show visual glyphs for word wrap

☑ Apply Cut or Copy commands to blank lines when there is no selection

Display

☑ Line numbers

☑ Enable single-click URL navigation

☐ Navigation bar

Note: This page sets options for all languages. To change options for only one language, select the desired language from the tree on the left.

Figure 2: Activate the Numbers option by using: Tools>Options>Text Editor> All Languages>General

Step 1: Atmel Studio: Download and Install

Use the following items to download, install and initialize AS6 on a computer with a USB port compatible with the programmer chosen in Step 2.

1. Download Atmel Studio (free of charge) from [6] http://www.atmel.com/tools/atmelstudio.aspx.
   You will be asked to register as a guest if you don't already have an account. Atmel has recently opened service offices around the world rather than the central office. Atmel has also initiated new (free) memberships with 'MyAtmel' that is worth the effort to register for downloads and assistance.
2. Install AS. USB drivers will also be installed. The process can be quite slow.
3. Activate the line numbers for the coding page as follows [Figure 2]. The line numbers will be listed on the left side of the coding page so that each program line will have its own number. When the program is compiled, AS will present any syntactic errors in an 'Error List' along with the line number. Unfortunately, the AS menu item to enable the line numbers is always well hidden. Perform the following:

   Select the Tools menu item in the AS6 top menu strip. Select Options at the bottom.
   In the left hand tree, scroll down until 'Text Editor' becomes visible.
   Left click the small "+" next to the title 'Text Editor'.
   Left click the small "+" next to "All Languages"
   Select 'General'
   On the right hand side, place a check mark in the 'Line numbers' under the 'Display heading'.
   Click OK.

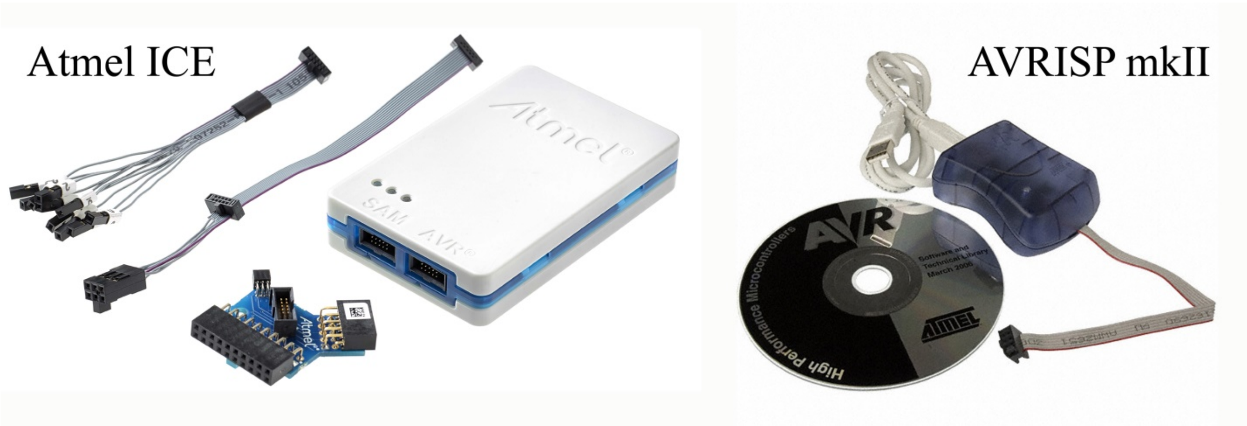
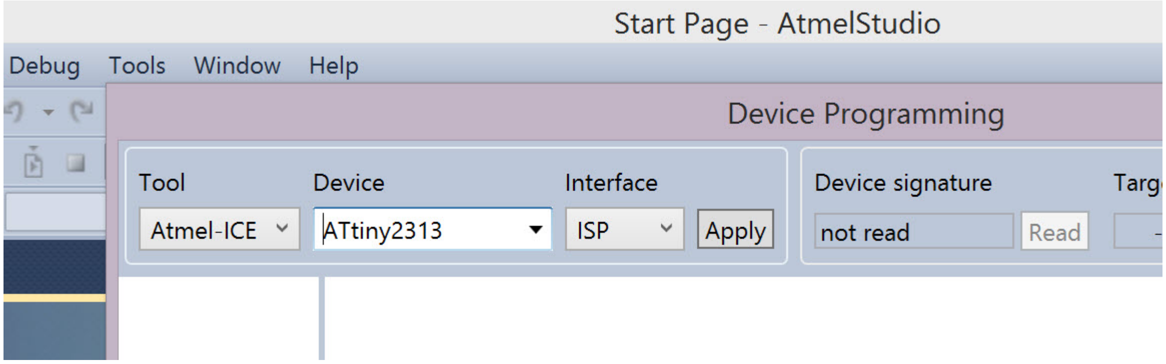


Figure 3: Atmel ICE (left) and AVRISP mkII (right) programmers. The Basic version for ICE includes the USB cable and the cable 2nd from the top left.

Step 2: Obtain a Programmer

Procure a programmer already known to be compatible with AS6 such as the Atmel ICE [7]. Digikey.com as an Atmel Distributor and Mouser.com offer the ATATMEL-ICE-BASIC (~ $55) and Atmel ATATMEL-ICE (~ $96) shown in Figure 3. The ‘basic’ version does not include all the adapters of the full version [7] but it does include the 6-pin adapter needed for the present project and most other AVR projects. The Atmel-ICE can be used with either the Atmel AVR MCUs (8bit) or with the Atmel SAM (32bit) MCUs that have the ARM cores. Unfortunately, the programming cable accompanying the ICE is rather flimsy and will require care not to break the wires. Perhaps some method of securing the wires could be developed. The older Atmel ‘AVRISP mkII’ (a.k.a. AVR USB MKII) was the workhorse programmer for Atmel AVR (8bit MCUs); however, neither Digikey nor Atmel sell it although eBay might have some (Figure 3). It’s probably not a good idea to buy a look-alike unit [8-10] without being certain of its compatibility with AS. Digikey sells an Olimex unit [8] with stock number AVR-ISP-mk2 (~$30) that Olimex claims to work with AS6. We have not tried the Olimex unit. The ICE unit works great.

The following two steps show how to install the Atmel-ICE programmer and the AVR-ISP mkII programmer. The ICE installed easier than the AVRISP which should probably be avoided.



Tool: Atmel-ICE
Device: ATtiny2313
Interface: ISP
Device signature: not read

Figure 4: Look for the 'Apply' button: Tools>Device Programming.

Step 3: Atmel ATATMEL-ICE-BASIC Installation

The Atmel ICE debugger/programmer uses native Windows drivers and those loaded by AS. After installing AS6 as in Step 1 above, install the Atmel ICE as follows.

1. Close AS6 if it's open.
2. Attach the Atmel ICE to the computer USB port using the supplied USB cable. The computer shows a status message that the Atmel ICE drivers are installing.
3. Attach the ribbon cable to the AVR side of the programmer.
4. Attach the programmer to the PC USB port.
5. Start AS6.
6. Run the Windows Device Manager on the PC by using one of the following methods. Choose the method easiest for your Windows Installation. We have tested both Windows 7 and 8.

   6.A Device Manager A:

   Win7: Click Start (lower left), open Control Panel, double click Device Manager

   Win8: Right click Windows symbol (lower left), click Control Panel, Double click Device Manager

   6.B Device Manager B:

   Win7: Click 'Start' on the PC, right click 'Computer', left click 'manage'.

   Win8: Right click Windows symbol (lower left), click Device Manager.

   6.C Device Manager C:

   Win7: Click 'Start' on the PC, enter the words Device Manager into the text search box.

   Win8: In Start Screen (with active tiles), Click magnifying glass (upper right) or just start typing Device Manager, and then select Device Manager.

   6.D Device Manager D:

   Win7 or Win8: Open AS6, then click the top menu sequence: Tools > Command Prompt; then type Device Manager.

   6.E Device Manager E:

   Win7 or Win8: Open windows explorer and navigate to C:\Windows\System32. Click 'cmd.run' then type Device Manager.

7. Verify programmer installation

7.A In Device Manager, click the + next to 'Human Interface Devices' and verify the presence of the item 'USB Input Device'. Unplugging the programmer should eliminate the relevant 'USB Input Device' item. Make sure to reattach the programmer.

7.B In AS, open the top menu sequence View > Available Atmel Tools and verify that the ICE programmer is listed.

8. Select the menu item 'Tools' and then 'Device Programming'. Select the 'Tool' drop down box and make sure that the 'Atmel ICE' option appears. Select the ICE programmer and ISP interface. Proceed to the next step without closing the Device Programming dialog.

9. Click the 'Apply' button if able to do so (Figure 4) – the 'apply' button might not be available until loading a Solution (i.e., program). Most likely AS will want to upgrade the programmer firmware prior to its first use. Go ahead and update. Sometimes an error occurs and the programmer looks dead and the programmer no longer appears in the Tools > Device Programming drop down box and under the View > Available Atmel Tools. Don't panic, proceed to Step 5 below. If the 'Apply' button was not enabled, remember to return to this step if/when the update occurs.

Step 4: Obsolete: Installing Atmel AVRISP mkII

As previously mentioned, the AVRISP mkII will work with AS6.2 but it is not recommended for use here especially given that (i) it will not work with 32bit Atmel SAM MCUs, (ii) it's price approaches that of the ICE, and (iii) the installation can be more problematic.

1. Install the drivers for the AVRISP MKII which can be found on the CD/DVD accompanying the programmer.
2. Attach the AVR USB MKII programmer to the computer USB port if not already attached. Allow any drivers to fully load.
3. Open Windows Device Manager by using one of the following methods easiest for your Windows Installation.

3.A Device Manage A:

Win7: Click Start (lower left), open Control Panel, double click Device Manager

Win8: Right click Windows symbol (lower left), click Control Panel, Double click Device Manager

3.B Device Manager B:

Win7: Click 'Start' on the PC, right click 'Computer', left click 'manage'.

Win8: Right click Windows symbol (lower left), click Device Manager.

3.C Device Manager C:

Win7: Click 'Start' on the PC, enter the words Device Manager into the text search box.

Win8: In Start Screen (with active tiles), Click magnifying glass (upper right) or just start typing Device Manager, and then select Device Manager.

3.D Win7 or Win8: Open AS6, then click the top menu sequence: Tools > Command Prompt then type Device Manager.

3.E Win7 or Win8: Open windows explorer and navigate to C:\Windows\System32. Click 'cmd.run' then type Device Manager.

4. The tree of items in Device Manager should have an entry called Jungo with sub-entries ‘AVRISP mkII’ and ‘WinDriver’. If you skipped item 1, then you might see the error ‘No driver found’.
5. In our case, we had to download Ria Services from Microsoft and then the computer would not load the installation package for DotfuscatorCE5.0.2500.msi. An internet search on that file name provided a solution at social.msdn.mincrosoft.com.
6. Start AS6.
7. Verify the AVRISP appears under top menu sequence View > Available Atmel Tools.
8. Select the menu item ‘Tools’ and then ‘Device Programming’. Select the ‘Tool’ drop down box and make sure that the ‘AVRISP’ option appears. Select the AVRISP programmer and ISP interface. Proceed to the next item without closing the Device Programming dialog.
9. Click the ‘Apply’ button if able to do so (Figure 4) – the ‘apply’ button might not be available until loading a Solution (i.e., program). Go ahead and update. Sometimes an error occurs and the programmer looks dead and the programmer no longer appears in the Tools > Device Programming drop down box and under the View > Available Atmel Tools. In such a case, search the internet for the error to find a solution (most likely found on AVRfreaks). You can also try using the procedure outlined in Step 5 below. We have not tried it for the AVRISP mkII. If the ‘Apply’ button was not enabled, remember to return to this step if/when the update occurs.

Step 5: Optional: Resolve programmer problems

The solutions to problems/errors can most often be found by typing the wording of an error into an internet search engine which quite often points to AVRfreaks.net. Obviously one benefit to exclusively employing Atmel Technology is that Atmel should be able to correct problems when they occur.

A. Firmware Upgrade Issue

One example problem concerns programmer firmware that does not properly upgrade. The AVR experts did find a solution. The following paraphrases the solutions found in the references [11,12].

1. Copy File:

   Copy the file: C:\Program Files (x86)\Atmel\Atmel Studio 6.2\tools\Atmel-ICE\atmelice_fw.zip
   To the folder: C:\Program Files (x86)\Atmel\Atmel Studio 6.2\atbackend
   Use the Windows Explorer for this purpose.

2. Run the following command (by following Items 3-4 below):

```
C:\Program Files (x86)\Atmel\Atmel Studio 6.2\atbackend> atfw -t atmelice -a atmelice_fw.zip
```

3. Open a command window:

   A. Win7 or Win8: Open Windows Explorer and navigate to C:\Windows\System32. Click ‘cmd.exe’. Note that depending on the Explorer settings, the extension ‘exe’ might not be displayed as part of the file name.

B. Win7: Click ‘Start’ (lower left) on the PC, enter ‘cmd.exe’ in the search box. Do not include quotes. Win8: In Start screen with the live tiles, either select the magnifying lens (upper right) or just start typing, type ‘cmd.exe’ without the quotes.
C. Win7: Click ‘Start’ (lower left) on the PC, select Run. Enter ‘cmd.exe’. Do not include quotes. Win8: Right click the windows symbol (lower left), select Run. Enter ‘cmd.exe’ wo quotes.
D. Win7 or Win8: In AS6 select the menu item sequence: Tools > Command Prompt.

4. Change the current directory (i.e., folder) using the Command Window

In the Command Window, use CD which means ‘change directory’. It’s often easiest to do the CD in stages in case of a typing mistake. If a mistake occurs, use the up-arrow key to get the previous typed command then use the left/right arrows to make corrections. Enter the following sequences (press return after each one):

A. cd c:\program files (x86)\atmel
B. cd atmel studio 6.2\atbackend

5. Enter the command

Finally, make sure the Atmel Ice is connected to the PC USB port and type the following command into the command window:

```
atfw -t atmelice -a atmelice_fw.zip
```

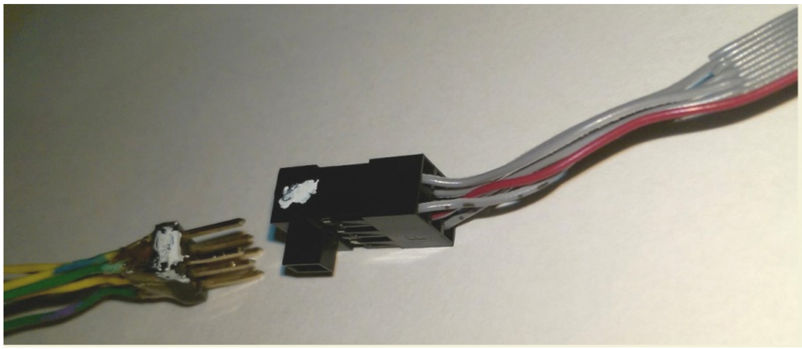
Be sure to press Return after entering this command. The script will require a few minutes but the Atmel ICE should be updated at the end.

6. Rumor:

One person mentioned in references [11,12] that his atfw.exe didn’t properly work. He did the copy as stated in Item 1 above and then ran the upgrade through Atmel Studio 6.2 again. We have not tested this approach.

B. Programmer Busy

For later info: Sometimes the compiled software/firmware won’t load into the MCU and the dialog for ‘Tools > Device Programming’ shows the programmer as *BUSY*. In such a case, close Atmel Studio and restart it. Check the connections on the board. Make sure that the two ISP rates are the same. One under Project > Properties and the other under Tools > Device Programming.



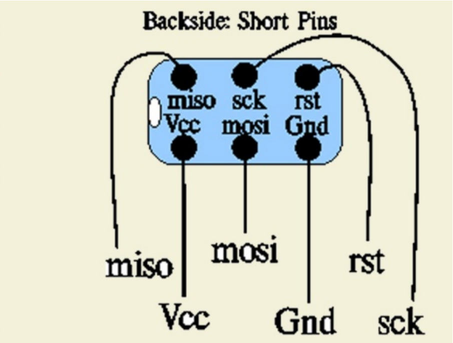


Figure 5: The 6-pin connectors with white dots. Wires have been soldered to the short portion of the pins for the adapter connector M2x3 (left). **Figure 6:** The 6-pin adapter M2x3 as viewed from the short pin side. Note the white dot on the side next to the Vcc and Miso pins.

Step 6: Construct an adapter for the programmer

As will be seen, the use of Experimenter's Breadboards make quick and easy work of prototyping MCU circuits. However, an adapter cable must be constructed to connect the 6-pin female header connector on the programmer to the experimenter's board. The adapter cable is constructed by soldering 24Ga solid wire to the pins in a male header consisting of 2 rows of 3 pins each. Figures 5 and 6 show the adapter with one end having the 6-pin male header, denoted M2x3, to connect with the programmer. It might be a good idea to make several adapters since any work with MCUs on experimenter's boards will require one. Refer to the Parts List for sources.

As a note, these Atmel Startup Instructables will at least require soldering the adapter and then the remainder can be accomplished with Experimenter's breadboards. However, truly functional devices cannot be created without a basic set of tools. For a discussion of soldering equipment and tools refer to Step 8. See reference [13] for a soldering tutorial.

1. Before starting, place a dot of white paint on the programmer connector near pin 1 to make it easy to identify. Actually, the product called White-Out, available at office supply stores, dries much faster. The Atmel Programmer has a red stripe on the ribbon cable to indicate Strand 1 for the cable. Place the white dot on the side of the programmer connector nearest the red line (see Figures 5 and 6). Refer to the user guide [7].
2. Generally, the as-received double row header is too long and needs to be cut to the 2x3 size using wire cutters.
3. Cut some solid conductor wires 6-8 inches long, strip the insulation from both ends (approximately ¼"), and solder to the short ends of the M2x3 pins. Stranded wires will not work for this application since the wires must be used with the Experimenter's Breadboard. The builder might find it easiest to hold M2x3 in the jaws of a 'third hand' and the stripped wire in the other jaw and position the two for soldering. Generally I coat the pieces to be joined with

solder prior to positioning. Be sure to solder to the shorter ends of the pins and do not apply the iron for more than a second or two since the plastic retainer must not be melted.

4. Once all the wires have been soldered then either of two procedures can be used to finish. The first procedure covers the solder joints with heat shrink tubing followed by a coat of quick-cure epoxy to hold the wires stationary. However, I have noticed that the pins can move if glue is not also applied to the short ends of the pins. The second procedure covers the full back side with epoxy and leaves off the heat shrink tubing. The epoxy adds strength to the joints as well as electrical insulation (see Figure 5).
5. Add a drop of White-Out to the side of the adapter that will match the side of the programmer socket with White-Out (see Figure 5, 6).
6. Finally and perhaps most importantly, label each wire with the Atmel names MOSI, MISO, SCK, RST, Vcc, and Gnd (Figure 6).   No one wants to spend time looking up the pin arrangement on the programmer prior to attaching the wires at the MCU.   I usually cut small pieces of paper and write the name on them and then take tape and tape to the wire (the paper hangs off the wire like a small flag on a pole). Table 1 provides the definitions for the acronyms.
7. The Atmel Studio, programmer and adapter will be used in the Atmel Startup 2 article.

Step 7: Comments on the Programmer Connections

The wires from the adapter connect directly to an Experimenter’s board containing the MCU to be programmed. A couple of comments are in order: (1) The ‘Reset’ line on the MCU often uses a 10k pull-up resistor (although it’s not necessary except for very noisy environments). (2) The SPI programming pins on the MCU can be used for both programming (using MOSI, MISO, etc) and for normal port operations since the programming functions won’t initiate without pulling the MCU reset pin to Logic ground. Furthermore, the programmer will be removed from the circuit once the MCU has been programmed and so the programmer could not possibly interfere with the MCU operation at that point.

Table 1: Programmer acronyms

MOSI	Master Out, Slave In	The PC supplies the master signals to the slave MCU
MISO	Master In, Slave Out	The slave MCU passes signals to the Master PC using this line
SCK	Serial Clock	The Master PC supplies this clock signal to the MCU to synchronize data transfer. Also known as Clk or SCLK or USCK.
RST	Reset	Connects to MCU ‘Reset’ pin which the programmer pulls to ground. Often, a 10k resistor connects the MCU Reset pin to Vcc. The MCU reset pin can also be used as a hardware reset to restart program execution by momentarily bringing the Reset pin to ground.
Vcc	Positive Voltage	The positive supply voltage for the MCU to be programmed and generally ranges 2.7 to 5V.
Gnd	Ground	The logic ground usually set at 0V

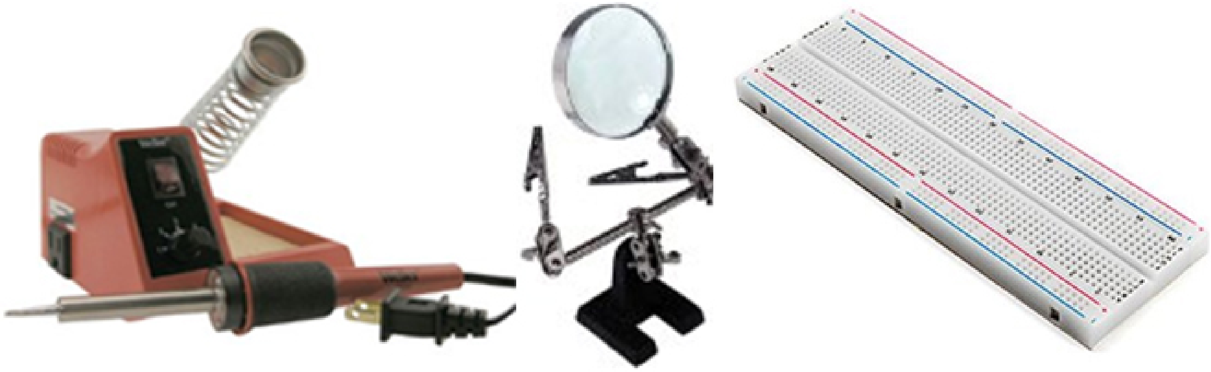


Figure 7: Left: Weller soldering station; middle: Helping Hand; right: breadboard

Step 8: Optional: Comments on Tools for Electronics

The successful completion of the Atmel Startup Instructables generally requires soldering equipment, hand tools, knowledge to use an experimenter's prototype breadboard and relevant electronic components, and knowledge to use Atmel Studio 6 (which is free) and program in C/C++. This and other Instructables can provide some suggestions (and will do so) but the knowledge base is large and is outside the scope of the present Instructable. See Refence [13] for some links regarding tools for electronics.

For any work involving electronics, the workshop should have a good set of tools including a multimeter, soldering equipment, hand tools, and parts. Radio Shack has a good selection of tools for Electronics work not to mention most of the parts required for the projects.

1. For the multimeter, it is generally better to find one having auto shutoff and capable of measuring Ohms, DC and AC voltage, DC mA, transistor hfe (gain). Some can measure capacitance and inductance. Check Radio Shack for some good ones. Inexpensive on Amazon: Etekcity Digital Multimeter and Velleman DVM850BL cost near $10 on Amazon.com but see the comments there. Generally the Fluke brand offers very good performance but at higher price. Check Sears and Home Depot too.
2. The soldering equipment includes a soldering station with temperature control such as Weller WLC100 (Amazon for $50) or a soldering iron such as the Weller SP25NUS (Amazon for about $15) works ok. However, soldering stations with temperature controls allow the temperature to be set and regulated. A soldering station with digital temperature read/set is really best such as the Elenco SL30A. The temperature readout is important since the temperature must be above 225C for 60-40 solder to melt but not so hot as to smoke components [15]. I generally run around 330C. Weller is a good brand. You can also check for Hakko soldering stations. It is probably a good idea to check for interchangeable tips and also find some with sharp tips on the order of 0.3mm.

3. As for solder, either the 60-40 rosin core or newer lead-free solder can be used. Generally the solder diameter should be in the range of 0.6mm to 1mm diameter. Do not use acid core solder such as might be used for plumbing. Consult for example reference [7] for soldering information. The solder can be found at Radio Shack or by searching Amazon.com for one of the diameters of 0.6mm, 0.8mm and 1.0mm to find some proper choices (for example search: "1mm solder"). The newer electronic silver solder should not be confused with the silver solders for jewelry which have much higher melting temperatures.
4. Tools including helping hands, wire strippers, wire cutters, pliers, long nose pliers, files, power drills and bits can be found at a number of stores including Sears, Harbor Freight Tools, Home Depot, and Lowes among others. A good slew of hand tools will be indispensable for work someday.

   The Helping Hand has two alligator clips on a bar with a magnifying lens; these are very useful for positioning and soldering wires without scorching the one's fingers. Many versions are available such as the SE MZ101B for $6 at Amazon.com.

   Except for the power drill, and screw cutter, the tools should be SMALL since the wire and the parts associated with the MCU are small. For example, the wire cutter Hakko CHP-170 is more suitable (Amazon.com) than Stanley 84-105 (Amazon.com). The long nose pliers should have tips about 1mm but should not be easily bent. Also check for needle nose pliers.

   A screw cutter can be found at Home Depot such as Urrea 298P or 298.

   Some needle files can be found at HarborFreight.com and HomeDepot.com among others.

   A power drill can be found at many locations and it should be able to accept up to at least 3/8 inch drill bits. Cordless power drills are OK and easier to use. Also pick up a set of drill bits that include about 20 bits with sizes from about 1/16" to about 1/2". The 1/2" will need to have a 3/8" shank to fit the 3/8" drill chuck.
5. Someday, the hobbyist will probably want to add electronic and optical equipment such as an oscilloscope, function generator, power supplies, RF and Optical Spectrum Analyzers, and a major stock of ICs, resistors, capacitors, switches, connectors, cables, and not to mention the hardware including wide range of screw sizes, washers, plastic boxes, stock metal … Eventually you will want a small machine shop (hint: LittleMachineShop.com and HarborFreight.com) with perhaps the capability to produce Printed Circuit Boards (PCBs) using Eagle Cad (free). Ebay.com often has inexpensive used, but guaranteed working test equipment.
6. The Experimenter's prototype board (often called a breadboard) can be found at many sources including Amazon (search for "Solderless Breadboard"). They start at about $5. Find one about 6 inches long such as BB830. Some come with wires. Others come mounted on a metal backing with connectors, and some have built-in power supplies (search powered breadboards). The powered boards are more expensive of course but can really be indispensable at times.
7. While Amazon and EBay have quite a few electronic components, it is an extremely good idea to become familiar with Digiky.com, Mouser.com, Allied electronics, Jameco.com, and Newark.com (Newark Electronics). The future of Radio Shack at this time is not known but it was always a great place to find just the right part or tool.

Step 9: Parts List

Generally, the electronic parts can be ordered from major distributors such as DigiKey.com, Mouser.com, and Newark.com to name a few. However, interestingly, Amazon.com also provides quite the range of electronic components while offering the possibility of 2 day shipping with Amazon Prime. Similarly, DigiKey can deliver within two days by choosing the USPS Priority Mail. Great deals can be found on Ebay but with longer shipping times in some cases.

1. Generally, the parts should be capable of 'through hole' mounting in order to be used with the experimenter's breadboard. If you have not done so already, consider purchasing several of each to not only have enough for both the enclosure and breadboard versions but also for building up your stock of parts. Atmel Programmer: ATATMEL-ICE-BASIC at Mouser.com or Digikey.com for about $55. The AVRISP mkII will work although it has been discontinued at Atmel. Digikey offers an Olimex unit AVR-ISP-mk2 for $28 claimed to be compatible with AS6.
2. Atmel Studio AS6 (free): http://www.atmel.com/tools/atmelstudio.aspx
3. Male Headers: Rows: 2, Pitch: 2.54mm (0.1"). Search Amazon.com for 'male header double row' 10 pieces for $7.
4. Heat shrink tubing (optional): Search amazon for 'heat shrink tube'. Kits with variaous sizes are available starting at about $5.
5. 5 min epoxy Amazon, HomeDepot, HarborFreight.com etc
6. 24 gauge solid core copper wire. I have been using old telephone wire and intercom wire that I have either found or purchased at Radio Shack. I especially like this wire because of the many colors, it can be cut to any length, and the diameter nicely fits the experimenter's boards. The 24ga has a diameter of approximately .51 mm but it's not critical. The 26ga should work ok even though the diameter is about 0.45mm. The outside plastic must be cut off to access the various colored inner wires.
   Amazon.com: search on '4 conductor 24 awg cable solid copper'
   either 60cents/foot or $20 for 100 feet.

   Ebay.com has nice 24 ga. solid core wire with multicolors. Search: 'solid 24 gauge wire kit'

   Home Depot: 100 ft. 6-conductor Indoor Phone wire for $20.
   I have not checked this for solid core copper.

7. Whiteout Pen: Search Amazon.com (or Staples or Office Depot …) for Whiteout Pen.

Step 10: References and Links

[0] The Atmel Startup articles, Instructables.com, 2015, M. A. Parker, Angstrom Logic:

A. Atmel Startup 1: Atmel Studio and Programmer
B. Atmel Startup 2: Microcontroller Circuits and Fuses
C. Atmel Startup 3: Blinky One – PORT, PIN, DDR and LED
D. Atmel Startup 4: Blinky Two – Switches, Pull-Up Resistor, and Bit Ops
E. Atmel Startup 5: Lifeline

[1] Atmel MicroController Units MCUs: www.atmel.com, see Products AVR

[2] Arduino products and learning: http://www.arduino.cc/

[3] The reader should have the complete Atmel manual for the ATTiny 2313A at
http://www.atmel.com/Images/doc8246.pdf.
Don't worry about its rather large size, a few simple 'tricks' will get you started. More information on the ATTiny2313A can be found at
http://www.atmel.com/devices/ATTiny2313A.aspx.

[4] Raspberry Pi products and learning: https://www.raspberrypi.org/

[5] Internal 'In-System Programmer ISP: http://www.atmel.com/images/doc0943.pdf

[6] Download Atmel Studio (free): http://www.atmel.com/tools/atmelstudio.aspx

[7] The Atmel ICE (Atatmel-ice-basic) can be found at either Digikey.com or Mouser.com starting for about $55. Atmel-ICE-basic can be used for both Atmel AVR and SAM. Here's the instruction document: http://www.atmel.com/Images/Atmel-42330-Atmel-ICE_UserGuide.pdf The Atatmel-ice (non-basic) sells for about $95 and includes the extra attachments.

[08] Olimex indicates that the AVR-ISP-mk2 can be used with AS6
https://www.olimex.com/Products/AVR/Programmers/AVR-ISP-MK2/resources/AVR-ISP-MK2.pdf

[9] A brief discussion of various programmers: http://www.ladyada.net/learn/avr/programmers.html

[10] The programmer and other valuable information and projects
http://startingelectronics.org/tutorials/AVR-8-microcontrollers/starting-AVR-development/

[11] Description of upgrade problem: http://www.avrfreaks.net/forum/new-atmel-ice-problem

[12] http://www.atmel.com/webdoc/atmelstudio/atmelstudio.AvrStudioUserGuide.FirmwareUpgrade.html

[13] Soldering tutorial: https://www.sparkfun.com/tutorials/106

[14] Examples of very good Instructables regarding tools for electronics

http://www.instructables.com/id/Tools-for-the-Electronics-Hobbyist/
http://www.instructables.com/id/Basic-Electronics-1/
http://www.instructables.com/id/Circuit-Building-101/

[15] Solder Temperature http://electronics.stackexchange.com/questions/1980/what-s-the-proper-soldering-iron-temperature-for-standard-031-60-40-solder