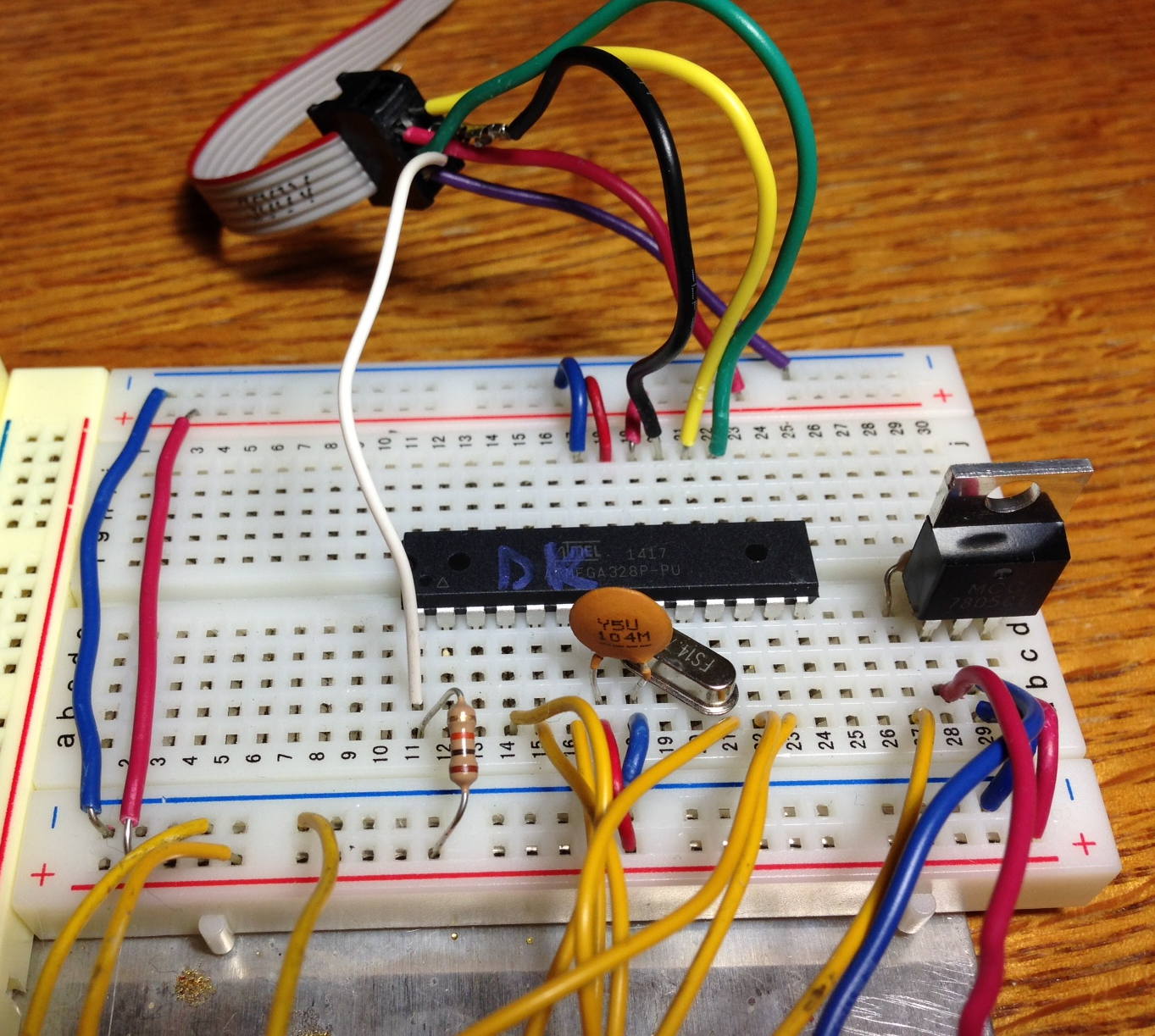
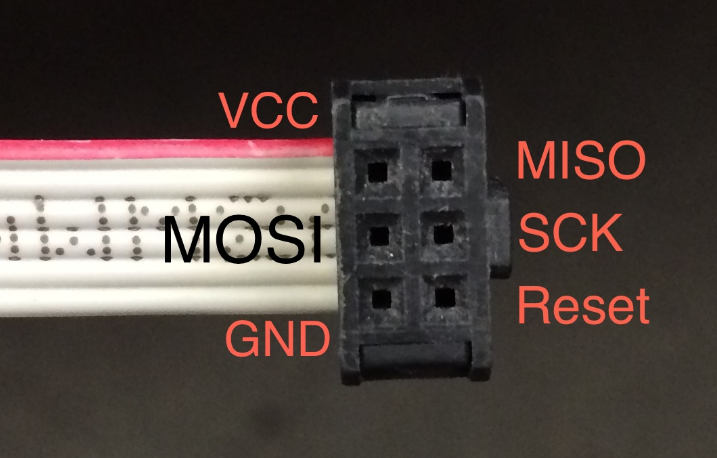
**Loading the Nerdkit Bootloader on ATMega328P-PU**

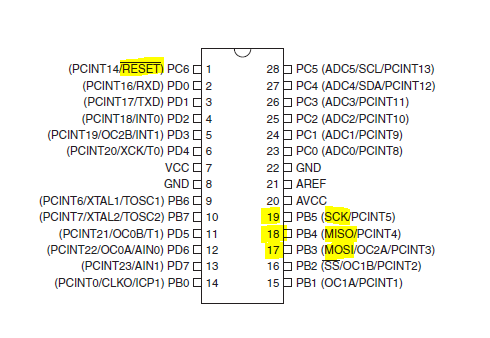
1. Obtain an AVRISP mkII (make sure it’s genuine)
2. Set up a breadboard similar to Nerdkit instructions with the ATMega328p-PU including:
   1. ATMega328p-PU
   2. 14.7 MHz Crystal oscillator
   3. .1uF capacitor
   4. 5V Regulator
   5. LCD setup not needed (gold wires shown on bottom of below photo and contrast resistor behind voltage regulator)
   6. Additionally, instead of wire from MCU pin 1 to +5V (as called for by Nerdkit Guide), use 10k resistor



1. Connect the AVRISP mkII connector to the breadboard pin assignments and apply power to the board (when connected properly, the light on the AVRISP mkII is green):



* 1. MOSI goes to MCU pin 17
  2. MISO goes to MCU pin 18
  3. SCK goes to MCU pin 19
  4. RESET goes to MCU pin 1
  5. GND and VCC go to ground and +5V respectively



1. Install the proper USB driver for the mkII that works with the avrdude command-line program. The article below gives instructions and link to download the drivers. (It took me two tries with a reboot in between to get the driver installation to work on Windows 7).

# [AvrIsp MkII (and others) Usb Driver for Arduino](http://www.visualmicro.com/post/2014/01/17/AvrIsp-MkII-Usb-Driver-for-Arduino.aspx)

By [Visual Micro](http://www.visualmicro.com/author/Admin.aspx)17. January 2014 09:21

**NOTE: Oct 2014 - Recent versions of Visual Micro install with a programmer option called "Tools>Visual Micro>Programmers>Atmel Studio AVRISP". Tool names that begin with the word "Atmel" use native Atmel Studio usb drivers and do NOT require the instructions mentioned below. The Atmel Jungo drivers will work well with Visual Micro and Atmel Studio functions.**

This document solves usb driver errors for the AVRMKII programmer. The error will be:- avrdude: usbdev\_open()

Atmel changed the usb driver they install to Jungo which doesn't work with Arduino or Visual Micro

This document explains how to switch the driver to one that the Arduino tool chain will work with.

Tested on Xp, Win7 and Win8.1

1) We downloaded the latest release (1.2.6.0) of libusb-win32 from <http://sourceforge.net/projects/libusb-win32/files/libusb-win32-releases/>  
  
2) Unpack the libusb zip and run the inf-wizard.exe in the bin folder AS ADMINISTRATOR  
  
3) Select the avrisp mkII from the list that appears when you run inf-wizard then click next through the wizard.  
  
4) When prompted select a place on your pc to save the .inf file that the wizard will create. (If you loose the file in the future you can simply repeat the task)  
  
5) When prompted click OK to install the driver.  
  
If required you can run the Atmel Studio 6.1 Jungo driver install again and revert to Atmel Studio drivers with the avrispmkII.   
  
It is probably also possible to switch back to arduino/avrdude mode by reapplying the .inf created in step 6) via device manager

Note: On my Windows 10 machine, I had to disable the Windows security check which only allows signed drivers. I had to click on the Restart button while holding the shift, which restarts with a special menu. Then, I had to choose Troubleshoot-> Advanced -> Startup Settings, choose Restart, then choose to disable driver signature enforcement.

Note: On my Windows 7 machine, there is no driver signature enforcement, however, I had to first uninstall the Jungo drivers provided by Atmel which were loaded with Atmel Studio using these instructions. The instructions say that the drivers will be in the Windows Device Manager under USB Controllers, but they were under a special Jungo branch. Once I uninstalled them, I had to find the .inf files with windrvr6.sys and delete them, too. For the Toshiba running Windows 10, I did not have to do this.

## Uninstall USB Drivers on Windows 7

Tuesday, February 14, 2012 posted by CSch

In the time you used your current Windows 7 system you undoubtly have plugged in a lot of different USB devices into your computer, most of them never to be used on it again. You will have notices that for every different device, Windows comes up with a message telling you that it automatically installs all drivers needed to use it.

That is perfectly alright, but what if you won't use the device ever again? The drivers will remain on your hard disk in case you still need it. This is not optimal for two reasons - first: in no time, you will have a massive amount of drivers for different USB devices installed; second: drivers may be out of date the time you use your device again, even if you use it frequently.

Therefore, there is an option to uninstall or update your USB device drivers manually. Open a command prompt by searching for cmd and enter:

*set devmgr\_show\_nonpresent\_devices=1*

This will set an environmental variable which can also be seen in Control Panel > System and Security > System > Advanced system settings > Environment Variables.... Afterwards, enter

*devmgmt.msc*

to open the device manager. This can also be done by right clicking Computer and choosing Manage, the device manager will be in the left column.

In the device manager menu, click View and enable Show hidden devices. If you now expand Universal Serial Bus controllers (USB) you will most likely have a bunch of transparent entries, which are all non present and/or hidden devices.

1. Try to erase your chip by executing avrdude on the command line.

NOTE: Use the version of avrdude that comes with WinAVR-20090313. I tried a newer version and it ran the “lock” command, but not the “efuse” command (said it couldn’t open the USB device).

* avrdude -c avrispmkii -P usb -p m328p –e

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.02s

avrdude: Device signature = 0x1e950f

avrdude: erasing chip (this message doesn’t always come)

avrdude: safemode: Fuses OK

avrdude done. Thank you.

1. Using a command prompt, go to the nerdit code folder, “.\bootloader328P”
2. Adjust the AVRDUDEFLAG variable in your Makefile so it now looks like this:

AVRDUDEFLAGS=-c avrispmkii -P usb -p m328p

all: fuses install

fuses:

avrdude ${AVRDUDEFLAGS} -U lock:w:0x2f:m

avrdude ${AVRDUDEFLAGS} -U efuse:w:0x05:m

avrdude ${AVRDUDEFLAGS} -U hfuse:w:0xd2:m

avrdude ${AVRDUDEFLAGS} -U lfuse:w:0xf7:m

install:

avrdude ${AVRDUDEFLAGS} -U flash:w:foodloader.hex:a

1. Execute the “fuses” target of the Makefile:

C:\Dev\NerdKit\code\bootloader328P>make fuses

avrdude -c avrispmkii -P usb -p m328p -U lock:w:0x2f:m

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.03s

avrdude: Device signature = 0x1e950f

avrdude: reading input file "0x2f"

avrdude: writing lock (1 bytes):

Writing | ################################################## | 100% 0.01s

avrdude: 1 bytes of lock written

avrdude: verifying lock memory against 0x2f:

avrdude: load data lock data from input file 0x2f:

avrdude: input file 0x2f contains 1 bytes

avrdude: reading on-chip lock data:

Reading | ################################################## | 100% 0.01s

avrdude: verifying ...

avrdude: 1 bytes of lock verified

avrdude: safemode: Fuses OK

avrdude done. Thank you.

avrdude -c avrispmkii -P usb -p m328p -U efuse:w:0x05:m

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.03s

avrdude: Device signature = 0x1e950f

avrdude: reading input file "0x05"

avrdude: writing efuse (1 bytes):

Writing | ################################################## | 100% 0.01s

avrdude: 1 bytes of efuse written

avrdude: verifying efuse memory against 0x05:

avrdude: load data efuse data from input file 0x05:

avrdude: input file 0x05 contains 1 bytes

avrdude: reading on-chip efuse data:

Reading | ################################################## | 100% 0.00s

avrdude: verifying ...

avrdude: 1 bytes of efuse verified

avrdude: safemode: Fuses OK

avrdude done. Thank you.

avrdude -c avrispmkii -P usb -p m328p -U hfuse:w:0xd2:m

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.02s

avrdude: Device signature = 0x1e950f

avrdude: reading input file "0xd2"

avrdude: writing hfuse (1 bytes):

Writing | ################################################## | 100% 0.01s

avrdude: 1 bytes of hfuse written

avrdude: verifying hfuse memory against 0xd2:

avrdude: load data hfuse data from input file 0xd2:

avrdude: input file 0xd2 contains 1 bytes

avrdude: reading on-chip hfuse data:

Reading | ################################################## | 100% 0.00s

avrdude: verifying ...

avrdude: 1 bytes of hfuse verified

avrdude: safemode: Fuses OK

avrdude done. Thank you.

avrdude -c avrispmkii -P usb -p m328p -U lfuse:w:0xf7:m

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.03s

avrdude: Device signature = 0x1e950f

avrdude: reading input file "0xf7"

avrdude: writing lfuse (1 bytes):

Writing | ################################################## | 100% 0.02s

avrdude: 1 bytes of lfuse written

avrdude: verifying lfuse memory against 0xf7:

avrdude: load data lfuse data from input file 0xf7:

avrdude: input file 0xf7 contains 1 bytes

avrdude: reading on-chip lfuse data:

Reading | ################################################## | 100% 0.01s

avrdude: verifying ...

avrdude: 1 bytes of lfuse verified

avrdude: safemode: Fuses OK

avrdude done. Thank you.

1. Execute the “install” target of the Makefile:

C:\Dev\NerdKit\code\bootloader328P>make install

avrdude -c avrispmkii -P usb -p m328p -U flash:w:foodloader.hex:a

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.02s

avrdude: Device signature = 0x1e950f

avrdude: NOTE: FLASH memory has been specified, an erase cycle will be performed

To disable this feature, specify the -D option.

avrdude: erasing chip

avrdude: reading input file "foodloader.hex"

avrdude: input file foodloader.hex auto detected as Intel Hex

avrdude: writing flash (31756 bytes):

Writing | ################################################## | 100% 0.60s

avrdude: 31756 bytes of flash written

avrdude: verifying flash memory against foodloader.hex:

avrdude: load data flash data from input file foodloader.hex:

avrdude: input file foodloader.hex auto detected as Intel Hex

avrdude: input file foodloader.hex contains 31756 bytes

avrdude: reading on-chip flash data:

Reading | ################################################## | 100% 9.84s

avrdude: verifying ...

avrdude: 31756 bytes of flash verified

avrdude: safemode: Fuses OK

avrdude done. Thank you.

DONE!