Flashing your own USB K+Dcan cable with enhanced firmware

Why?

Unmodified USB K+Dcan cables which are sold online have a firmware bug which corrupt certain BMW ECUs and do not operate at optimum speed. Flashing your stock cable with enhanced firmware will allow you to safely flash and code your vehicle without the danger of corrupting your ECU, its also faster.

Credit:

First of all, thanks and all credit goes to Ulrich Holeschak who developed this firmware and documented the procedure for use with his open source Ediabaslib. Without his foundational work with EDIABAS and associated hardware adapters, this would not be possible. Please visit and support him here: https://github.com/uholeschak

Enhanced firmware benefits:

Able to flash the following BMW ECUs without corruption:

- M3/M5/M6 DME (MSS6X)
- E6x M5/M6 SMG (GDSMG3)
- E9x GM Auto (GM1912)

Additional benefits include:

- Faster CAN communication (block size 0 and no separation time).
- Allows configuration of CAN mode all the time, not only after power on.
- Allows configuration of the CAN parameters (block size and separation time)
- Correctly displays the ignition status (only available for K-line mode).
- Contains a bootstrap loader that allows firmware update without opening the device.
- Reduced power consumption due to use of sleep mode.

Requirements:

HARDWARE:

In addition to one of the compatible cables listed later in this document, you will need the items below:

1. Arduino Uno (R3, doesn't matter) or other model Arduino - $3.65 on ebay
2. Dupont or Arduino jumper wires (male to female) - $1.33 on ebay for 40ct
3. 2.54mm breakout pins (6 pins total) - $1.13 on ebay
4. Soldering iron/solder (if flashing a generic USB cable with switch that has empty ISP pinouts). If you have a steady hand you may be able to use male to male jumper wires and hold in place during the flash process.

SOFTWARE:

To flash the required software to the Arduino and cable, you will need two programs and the latest firmware file.

Software:

- ARDUINO IDE: https://www.arduino.cc/en/main/software
- AVRDUDESS: http://blog.zakkemble.net/avrdudess-a-gui-for-avrdude/

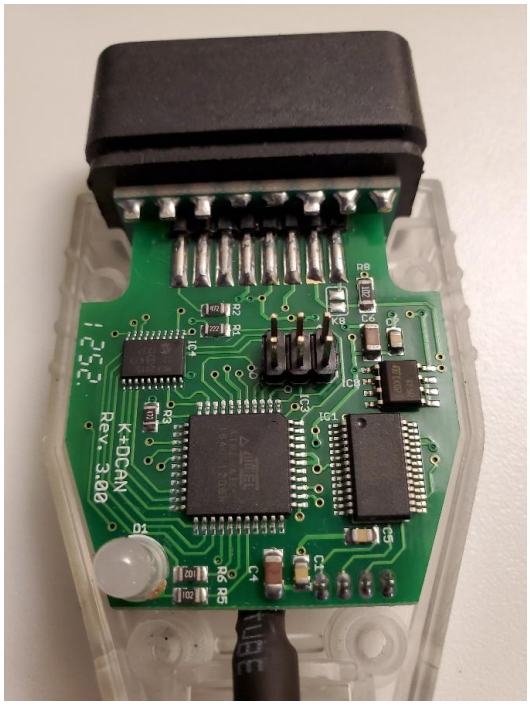
Firmware:

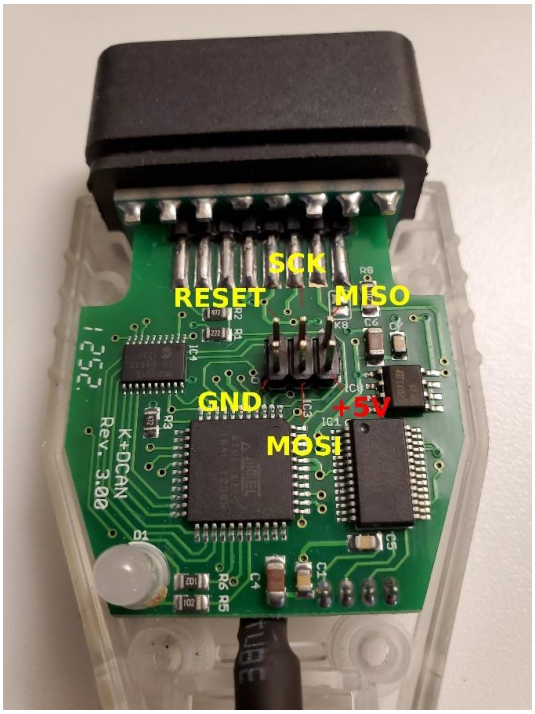
- Latest Ediabaslib Binary cable Firmware: https://github.com/uholeschak/ediabaslib/releases

COMPATIBLE CABLES:

It seems that any ATMega162 based board can be flashed with this updated firmware. However, we have only tested it on the cables listed in this document. As such, please be aware that we cannot be responsible if a cable not listed in this document fails to flash.

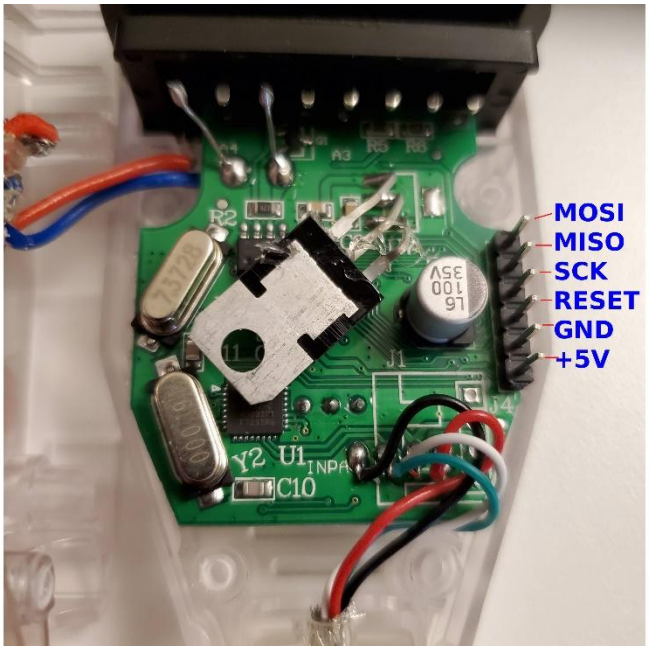
ONESTOP Cable with ISP header: No soldering required



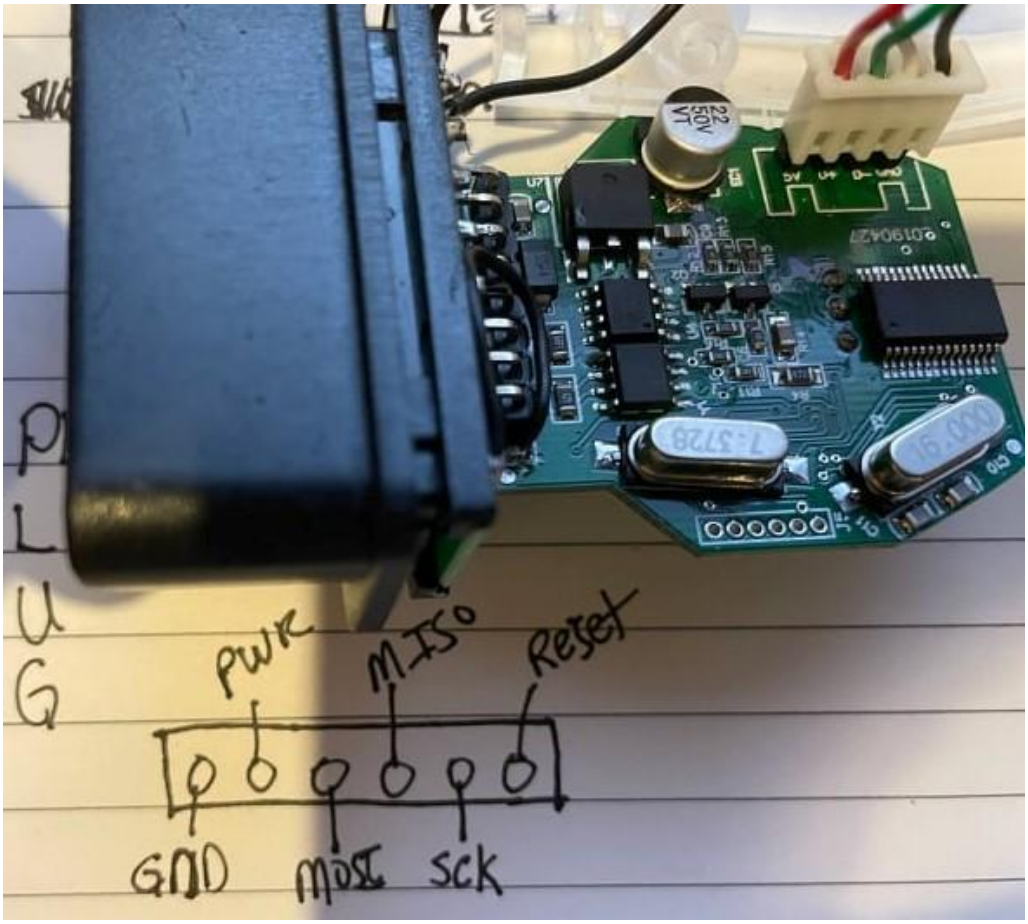


Generic AtMega162 based USB INPA cable with switch and ISP headers on circuit board: Soldering of header pins required

There are many sellers on ebay and amazon which sell a cable with integrated switch. However, not all come with a circuit board with ISP pin headers. Verify with the seller that the cable does have the ISP header prior to purchasing. In the images below, we show the circuit board with pin header that we have already soldered on. The stock cable will not include the male pin headers, but rather empty holes which will allow you to solder on your own pins.



Another generic AtMega162 based USB INPA cable with different pin header layout (thanks go to Joe):
Soldering of header pins required



SOFTWARE INSTALLATION

STEP 1: Install software programs and download latest firmware

After downloading the files listed earlier in this document, install all of them using the ***default options***.

Notes:

Arduino IDE:

- When the Arduino IDE asks if you want to install any drivers from ADAfruit industries, select yes or OK. It will prompt multiple times, but continue to agree to the driver installations.

AVRDUDESS:

- Allow AVRDudess to put an icon on your desktop

Binary firmware:

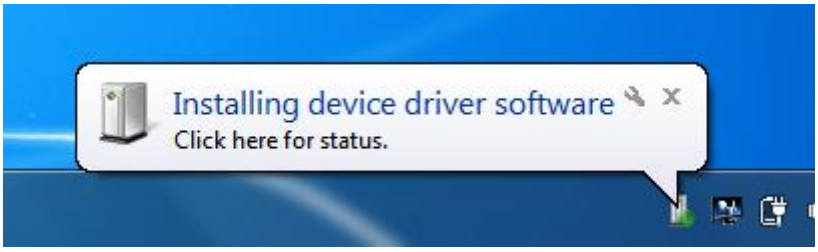
- Extract file with your favorite .zip archive extracting utility and save it to a folder on your desktop.

CONFIGURING THE ARDUINO

STEP 2: Connect the Arduino to PC using supplied USB cable

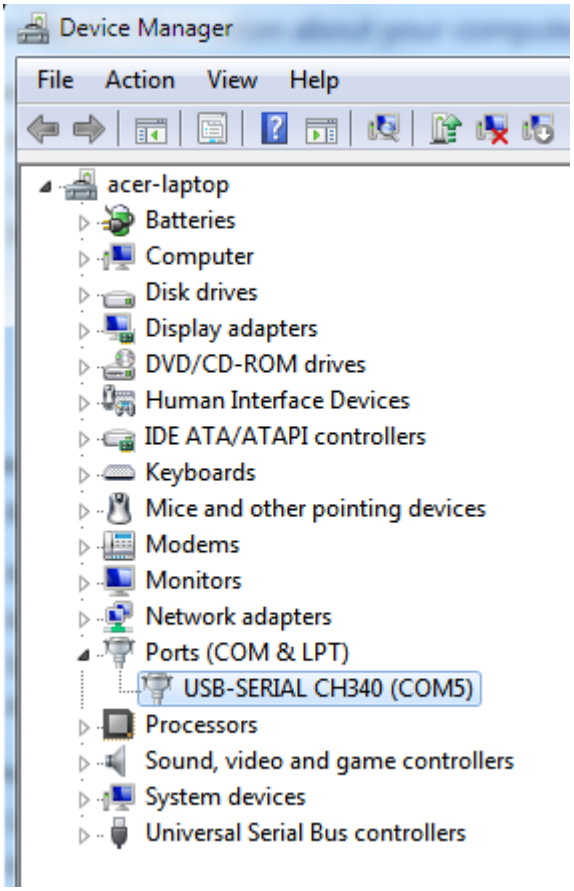
1. After installing the software above, plug in your Arduino to the computer using the supplied USB cable.

   Windows will detect and install the driver for the device.



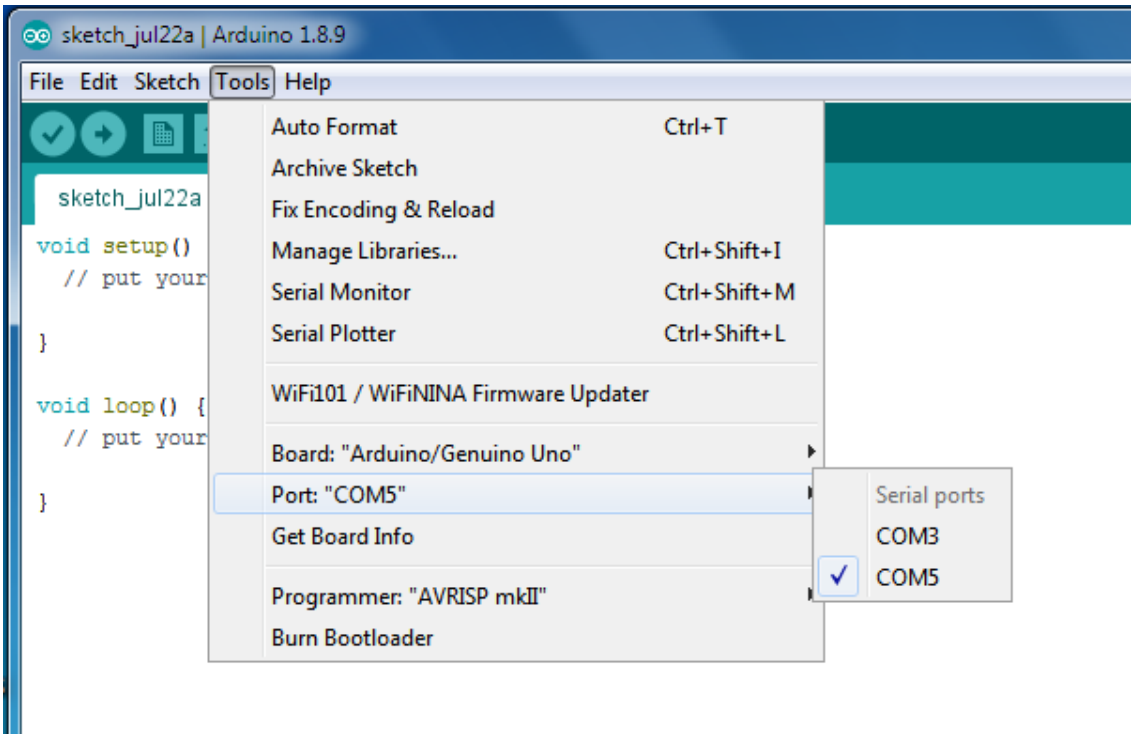
   However, if it does not and it shows that windows has failed to install a driver, download and install the driver from the following link (CH341 based Arduino boards):
   https://github.com/himalayanelixir/Arduino_USB_Drivers/tree/master/Windows

2. Verify the com port of the Arduino:
    a. Open up windows device manager and expand the ports tree to view the Arduino com port designation (COM5). Make note of this for the next step.

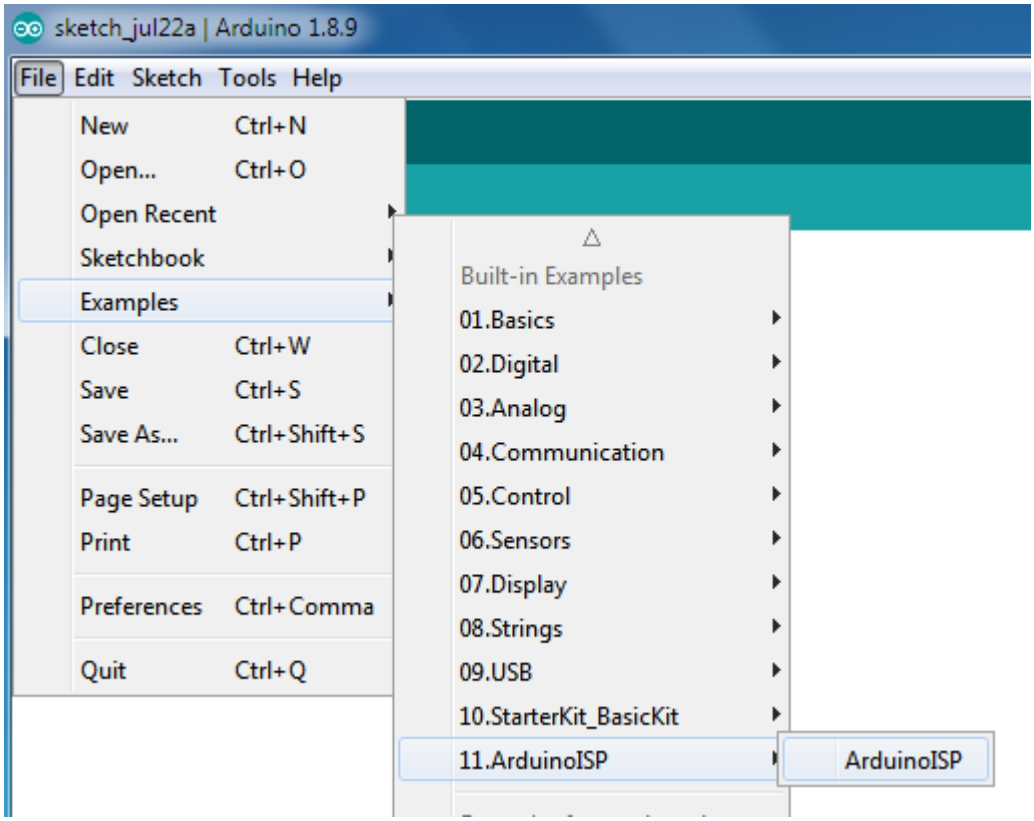


STEP 3: Flash Arduino as programmer

1. Run the Arduino IDE application and click OK to any prompts that show (java/firewall).
2. Click the TOOLS menu > BOARD > and select the Arduino board you have (be it Arduino Uno, mega, etc).
3. Click the TOOLS menu > PORT > and select the COM port you made note of in the previous step (In this example its COM5).

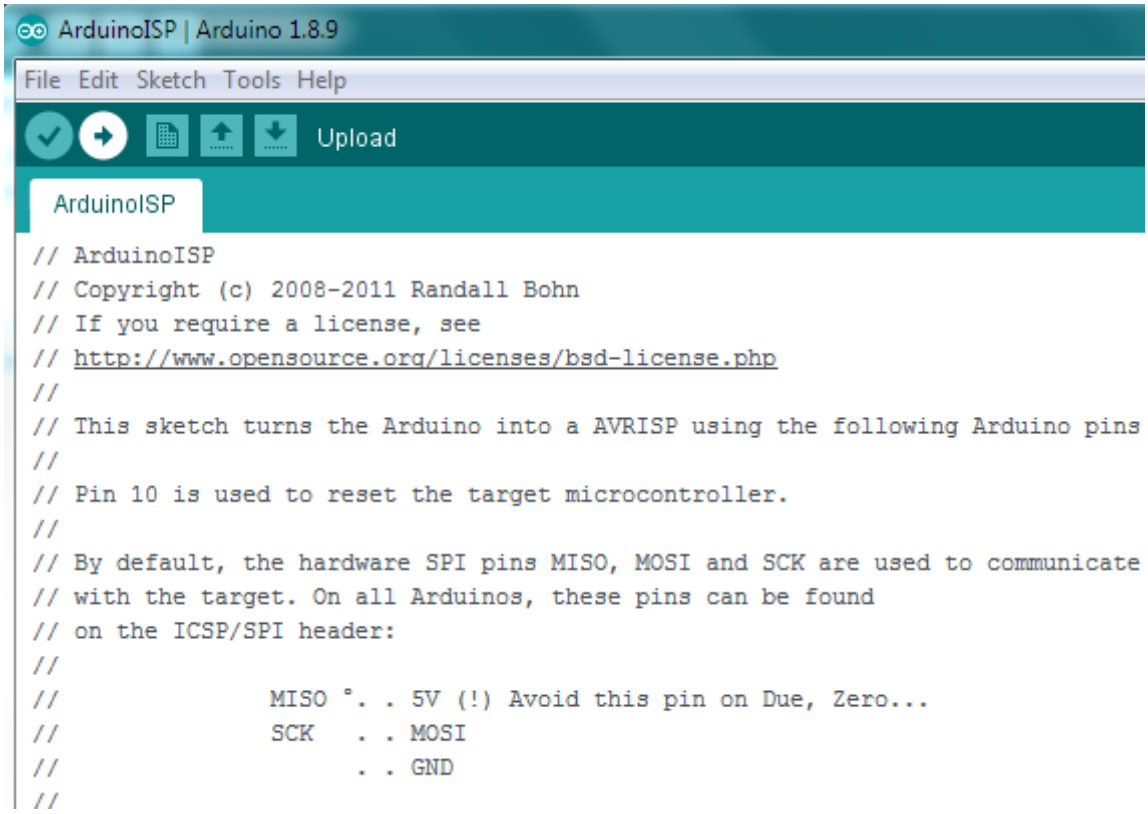


Next, click on the FILE menu > Examples > 11.ArduinoISP > ArduinoISP



It will open up a new IDE window containing the code which we will upload to the Arduino.

Finally, Click the Arrow button (next to the checkmark circle) highlighted (in white) in the image below.



Clicking the Upload button will then begin to compile the sketch, and when done will tell you it is “Done uploading”.

You may now close the Arduino IDE. The Arduino is now an Atmel programmer!

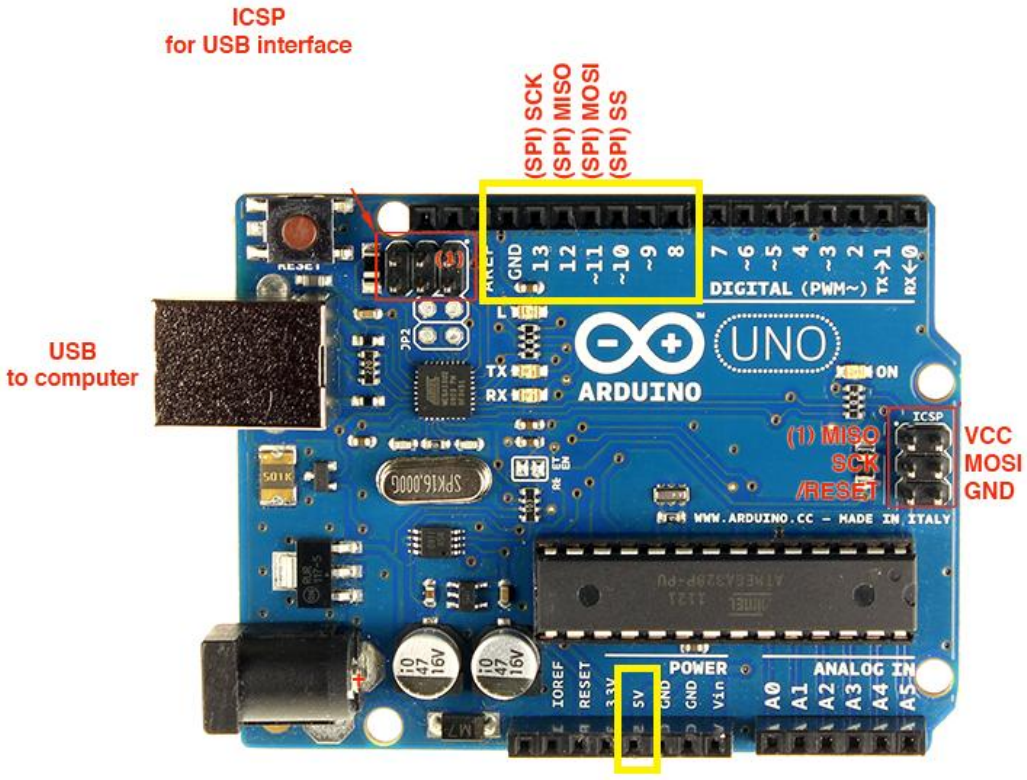
CONNECTING THE ARDUINO TO THE USB CABLE ISP PIN HEADER

STEP 4: Connect Arduino to USB cable using jumper cables

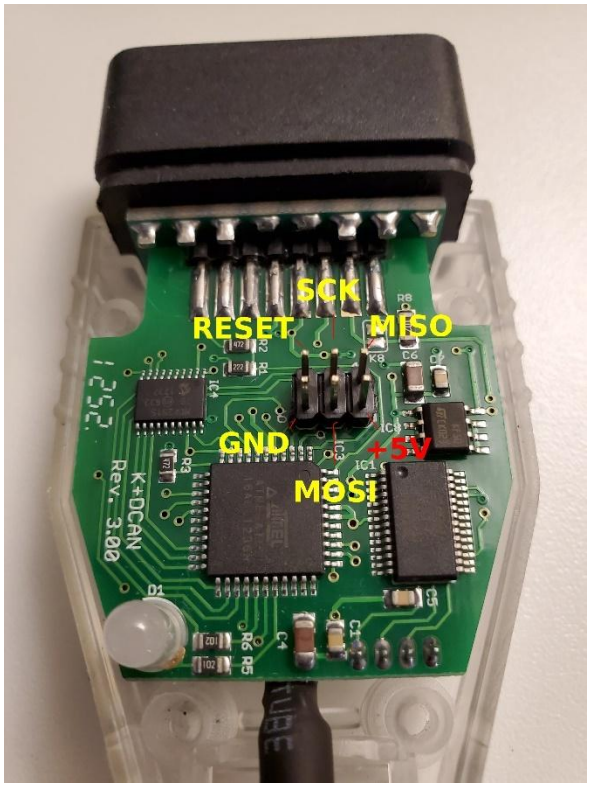
For this step you will reference the images in this document and table below to see the correct pinout designations for the type of cable you are trying to flash. It's imperative you connect the correct pins on the Arduino to the correct pins on the cable you are trying to flash. Failure to do so will prevent the flash from completing successfully. Bear in mind it makes no difference the color of the wires, as long as the pin number on the Arduino matches the Pin designation on the ISP header on the USB cable you are trying to flash.

Arduino Digital Pins	ISP Header on USB cable
10	RESET
11	MOSI
12	MISO
13	SCK
GND	GND
5V	5V

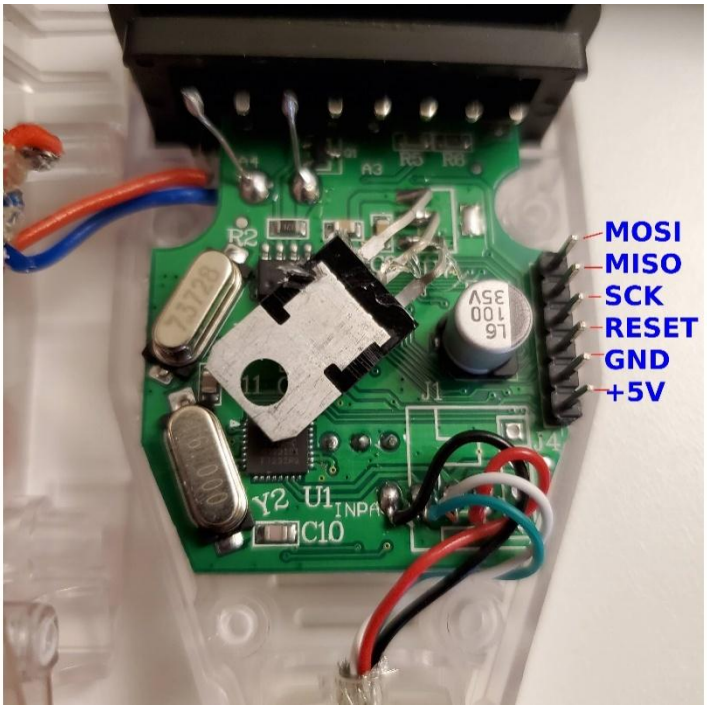
Arduino Pins referenced above



ONE STOP ELECTRONICS CABLE



GENERIC CABLE WITH SWITCH AND ISP BREAKOUT



STEP 5: Flash cable with enhanced firmware

Verify:

1. The Arduino and USB cable are correctly connected together via jumper cables
2. The Arduino is connected to the computer by USB cable
   *Optional: Plug the USB cable into the computer to supply it with power (it isn't necessary to install any drivers for it).

Once the above has been verified, run the AVRDUDESS application and configure the options *exactly* as shown below:

Programmer (-c): Atmel AVR ISP

Port (-P): COM PORT YOU NOTED ABOVE (COM 5 in this example)

Baud rate (-b): 19200

Bit clock (-B): blank

Presets: Default

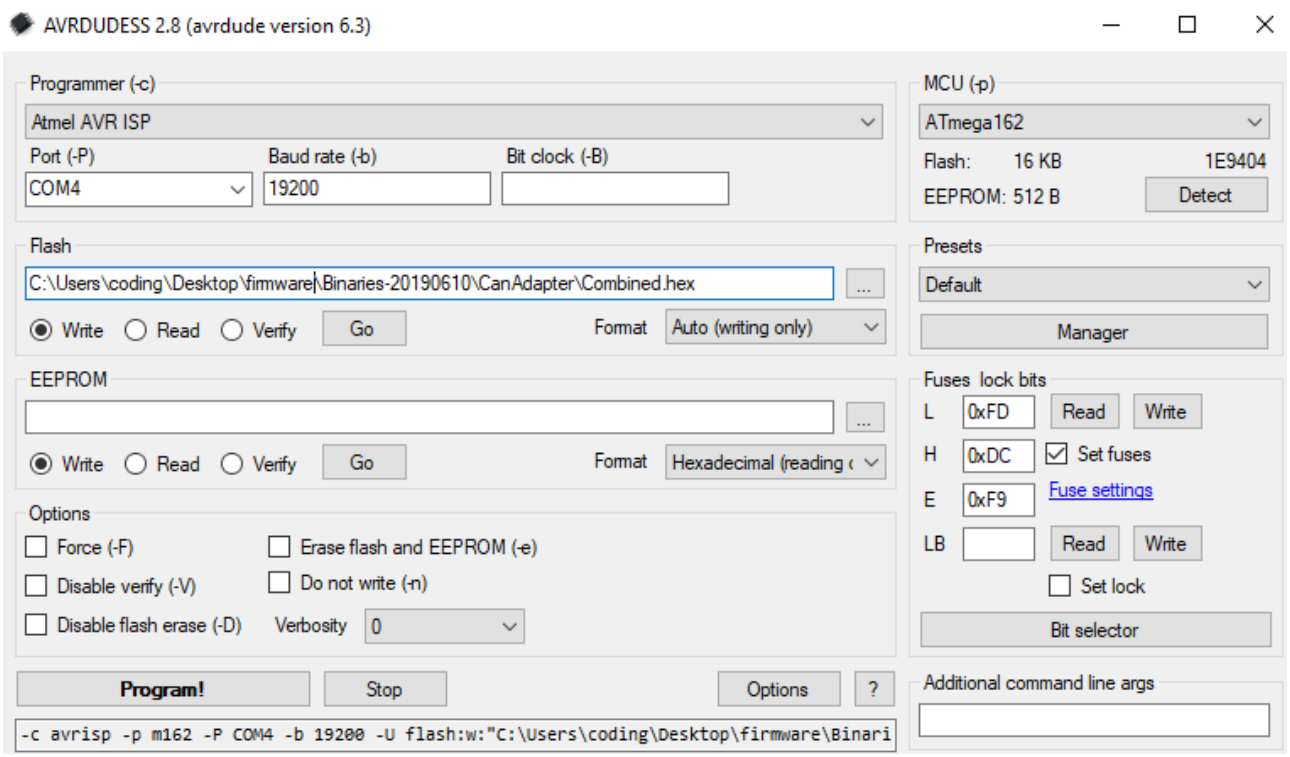
Options: Uncheck all (none selected)

Fuses lock bits:

L : 0xFD
H : 0xDC
E : 0xF9

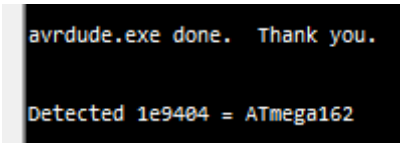
Set Fuses: CHECKED

LB: BLANK



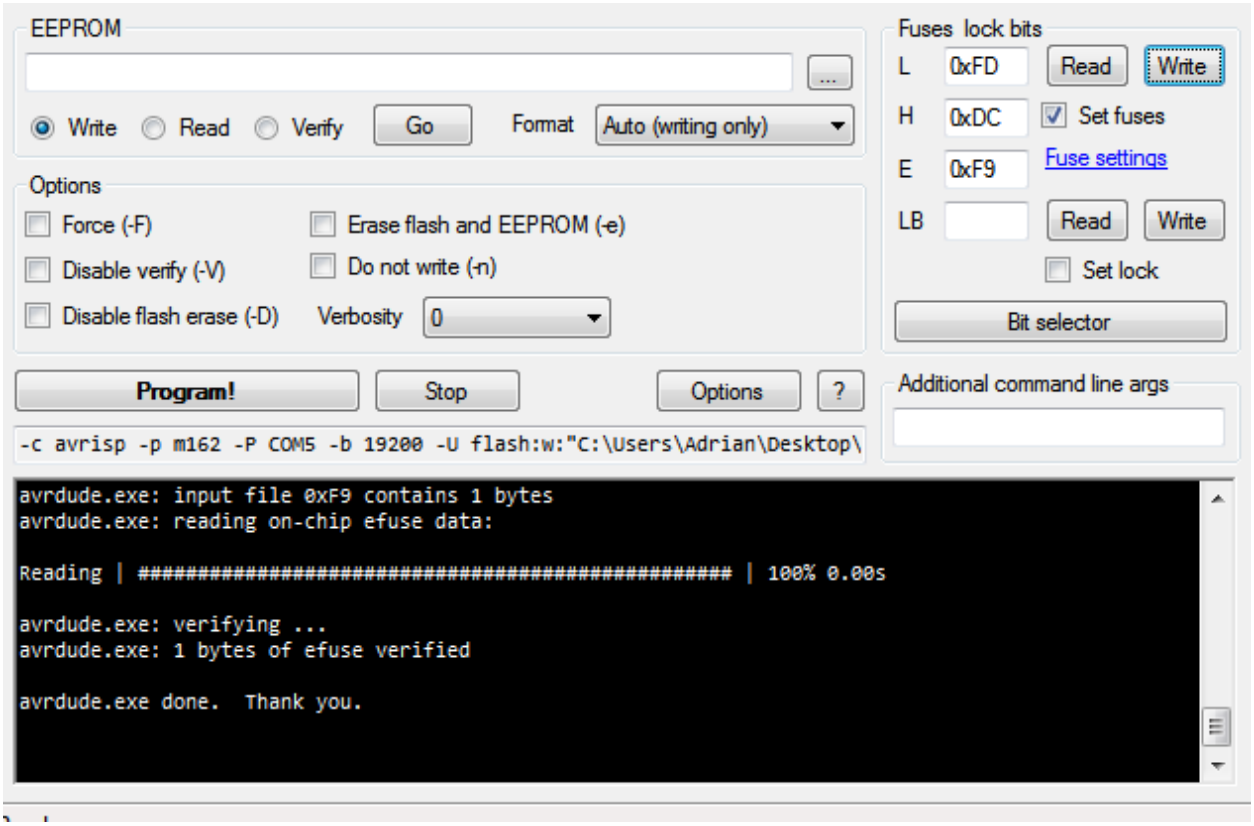
Once the AVRDUDESS options have been set:

1. Click the detect button at the top right corner to verify the Arduino can communicate with the USB Cable microcontroller. If everything is connected properly, it should detect and specify the chip in the terminal window and populate the MCU (-p) options with the correct microcontroller.



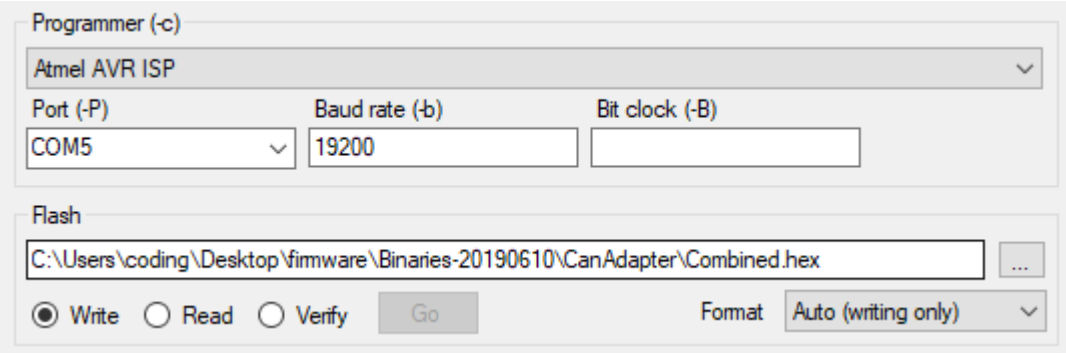
If AVRDUDESS cannot detect the correct chip, verify the Arduino to ISP header connections and double check soldering points.

2. Next, set the cable fuses by clicking the WRITE button directly under the 'Fuses Lock Bits' heading. This will write the three fuses we specified earlier to the cable. AVRDUDESS will also display a terminal message verifying the fuses as shown below.



3. Next, under the FLASH section of AVRDUDESS, click the '…' next to the empty space to browse and select the 'Combined.hex' firmware file located in the ***CanAdapter*** folder from the archive you extracted from the latest binary file located at

   https://github.com/uholeschak/ediabaslib/releases

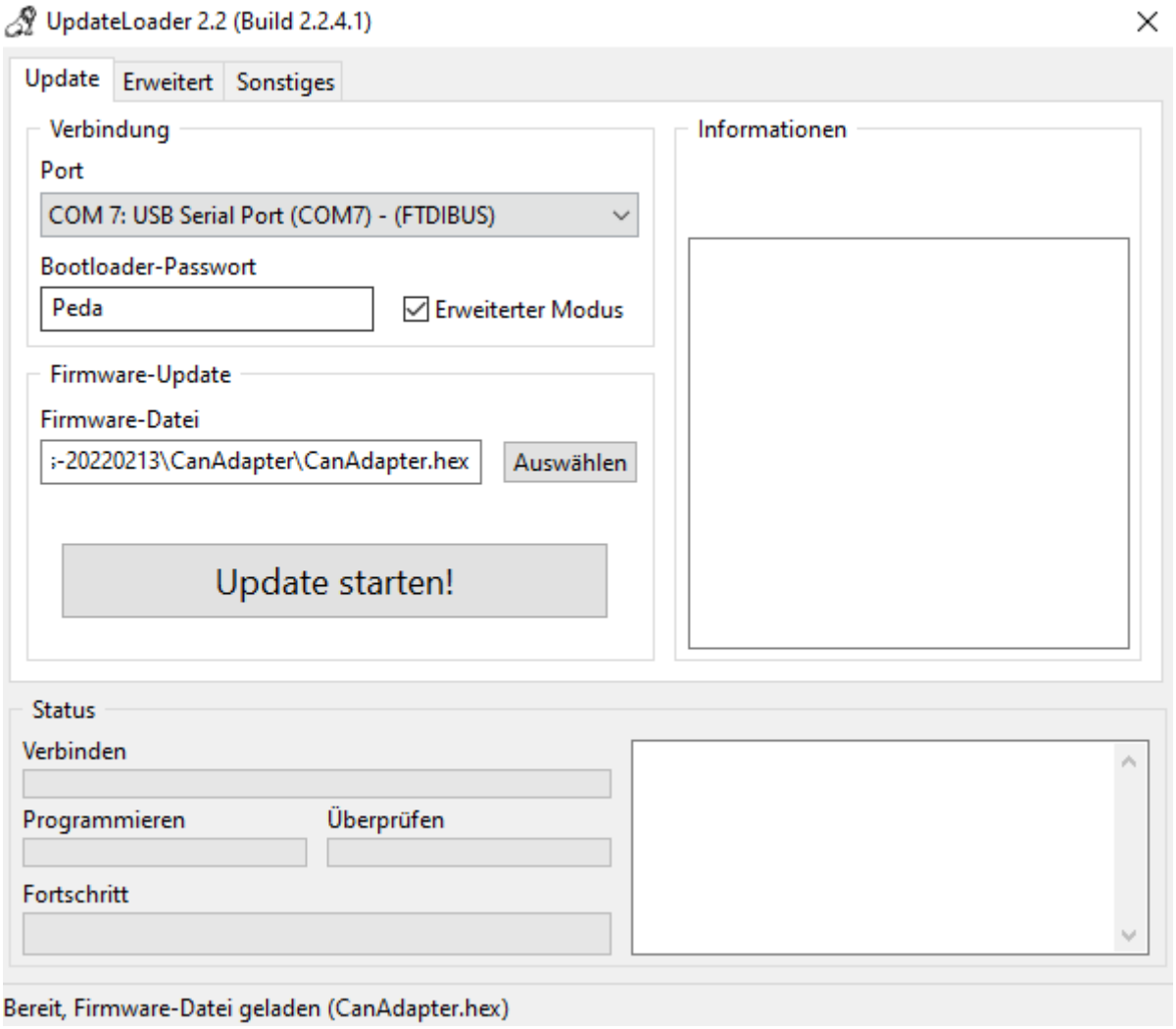


1. Verify that the Write radio button is selected, and Auto (writing only) is selected from the Format drop down selection.
2. Click 'Go' to begin the flashing process.

At the end of the flash, the LED may light up on the USB cable.

That's it, your cable is now updated with enhanced firmware which improves communication speed and guards against corrupt ECU flashes!

UPDATING WITH LATEST FIRMWARE via USB:



1. With `UpdateLoader-win32.exe` (application is only in the German language) it's possible to update the firmware without connecting a header.
2. Run UpdateLoader-win32.exe (located under the CanAdapter folder)
3. Connect the OBD plug to a vehicle OBD port (to supply 12v to the cable)
4. Select the COM port and the CanAdapter.hex file (not Combined.hex!) and click “Update starten!”

   NOTE: The USB update can only be done after the cable has been previously flashed with replacement firmware.