Tutorial:

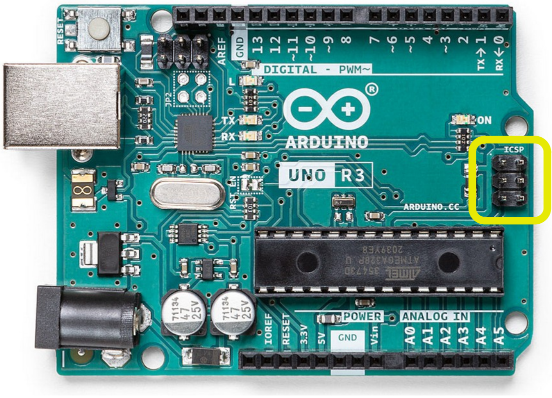
In-Circuit Serial Programming (ICSP) of the Commander X16 System Management Controller (SMC) with an Arduino UNO as programmer

WARNING: This is a draft that may contain errors or omissions that damage your hardware.

1. Introduction

This tutorial shows how to use an Arduino UNO to program the SMC of the Commander X16.

The ICSP header on the Arduino board is the six pins at the middle of the right edge. Pin 1 is the top left one.



2. Install Arduino IDE

It is not strictly necessary to install the Arduino IDE, but it makes programming the SMC easier. A bonus is that you can build the SMC firmware yourself with the IDE.

The Arduino IDE can be downloaded from https://www.arduino.cc/en/software. This tutorial is based on version 1.8.19.

3. Configure your Arduino UNO as an ICSP programmer

The Arduino UNO does not by default work as an ICSP programmer. You need upload some code to the board before continuing.

In the Arduino IDE, click File/Examples/11.ArduinoISP/ArduinoISP. This will open a project containing all code that we need to install.

Attach a USB cable between your computer and the Arduino UNO.

In the IDE, select Tools/Board/Arduino AVR boards/Arduino UNO.

Verify under Tools/Port that the port you attached the Arduino UNO to is selected.

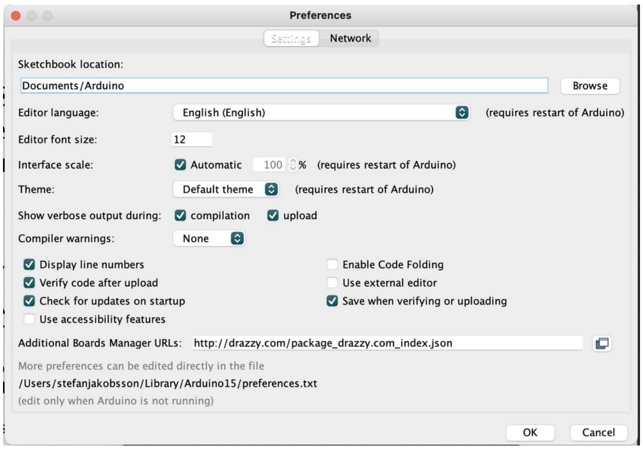
Click on Sketch/Upload to upload the ArduinoISP code to your board.

4. Add support for ATtiny boards

The SMC is an ATtiny861, which is not supported in the IDE by default.

To add support for ATtiny boards, open the Preferences dialog, and add the following to the Additional Boards Manager URLs: http://drazzy.com/package_drazzy.com_index.json.

While you are there, also tick the option Show verbose output during upload.



Sketchbook location: Documents/Arduino

Editor language: English (English) (requires restart of Arduino)

Editor font size: 12

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: ☑ compilation ☑ upload

Compiler warnings: None

☑ Display line numbers
☑ Verify code after upload
☑ Check for updates on startup
□ Use accessibility features
□ Enable Code Folding
□ Use external editor
☑ Save when verifying or uploading

Additional Boards Manager URLs: http://drazzy.com/package_drazzy.com_index.json

More preferences can be edited directly in the file
/Users/stefanjakobsson/Library/Arduino15/preferences.txt
(edit only when Arduino is not running)

5. Make a test upload

Connect the Arduino UNO to the computer via a USB cable. Verify that nothing is attached to the Arduino ICSP header.

Under Tools/Board/ATTinyCore, select ATtiny261/461/861 (a). Do not select the one that ends with Optiboot.

Under Tools/Chip, select ATtiny861 (a).

Under Tools/Programmer, select Arduino as ISP.

Under Tools/Port, verify that the port you selected the Arduino UNO to is selected.

Open File/01.Basics/Blink. This is a simple code sample that blinks a LED. It does not matter, because we not really installing this code.

Click on Sketch/Upload.

The Arduino IDE uses the programming tool avrdude behind the scenes. If you enabled verbose output as mentioned above, you will see where the avrdude executable is stored in your system and what options are used when it is called. Copy that information from the output window.

The upload will of coarse fail, as we had not connected anything to the ICSP header.

6. Connection between the Arduino Uno and the Commander X16

Before you begin this step, ensure that both the Arduino UNO and the X16 are disconnected. I recommend that you not only turn off the X16, but also unplug the wall socket, as the SMC always is running when the board is connected to power.

The connection between the Arduino Uno to the Commander X16 rev 4 board is made through each board’s ICSP header.

On the Arduino Uno, you find a standard 6 pin ICSP header, and on the X16 board there is a standard 10 pin ICSP header.

One simple option is to use six female-to-female Dupont jump wires to make the connection as set out below:

Uno (pin/label)		*X16 (pin/label)*
1 MISO	↔	9 MISO
2 VCC	↔	2 5VSB
3 SCK	↔	7 SCL
4 MOSI	↔	1 SDA
5 RST	↔	5 RST
6 GND	↔	10 GND
		(Pins 3, 4, 6, 8 NC)

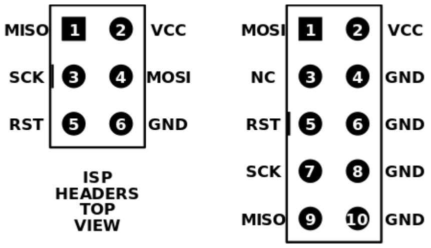


Figure 1: ICSP standard headers

7. Jumper settings on the Commander X16

Before you start updating the SMC, there is one more preparatory step. You need to remove the two jumpers where it says on the board “Remove Jumpers from J5 to Program Microcontroller”.

8. Download firmware

The SMC firmware is downloadable from https://github.com/X16Community/x16-smc.

Go to Releases, and fetch the file “x16-smc-r43+bootloader.hex”. This file contains both the firmware, and a bootloader that makes it possible to make further updates directly from the X16.

9. Programming the SMC

You do not need the Arduino IDE for this step. The actual programming of the SMC is done from the command line using the avrdude utility.

Keep the X16 disconnected from power.

You may now connect a USB cable between the computer and the Arduino UNO. The Arduino UNO ICSP header should already be connected to the X16 ICSP header, and the J5 jumpers should be removed.

Programming the SMC requires you to first set some board options that are called fuses in Arduino world. This is done by the following command:

```
avrdude -cstk500v1 -pattiny861 -P<yourport> -b19200 -Ulfuse:w:0xF1:m -
Uhfuse:w:0xD4:m -Uefuse:w:0xFE:m
```

After that you write the actual firmware to the X16 with the following command.

```
avrdude -cstk500v1 -pattiny861 -P<yourport> -b19200 -Uflash:w:x16-smc-
r43+bootloader.hex:i
```

You will probably need to write the absolute path to avrdude. You should have gotten that from the verbose output in step 5.

Also verify that the -c, -p and -b options are the same as you could see in the verbose output from the test run with the IDE. If not, change according to the options used in the IDE.

Do not change the values after -Ulfuse, -Uhfuse or -Uefuse as there is a risk of bricking the ATtiny.

10. Clean up

Unless there was an error message when running avrdude, you have now updated the SMC firmware.

Disconnect the USB cable between the computer and the Arduino UNO.

Disconnect the dupont jump wires between the Arduino UNO and the X16.

Reinstall the two J5 jumpers.

You may now power on and start the X16.