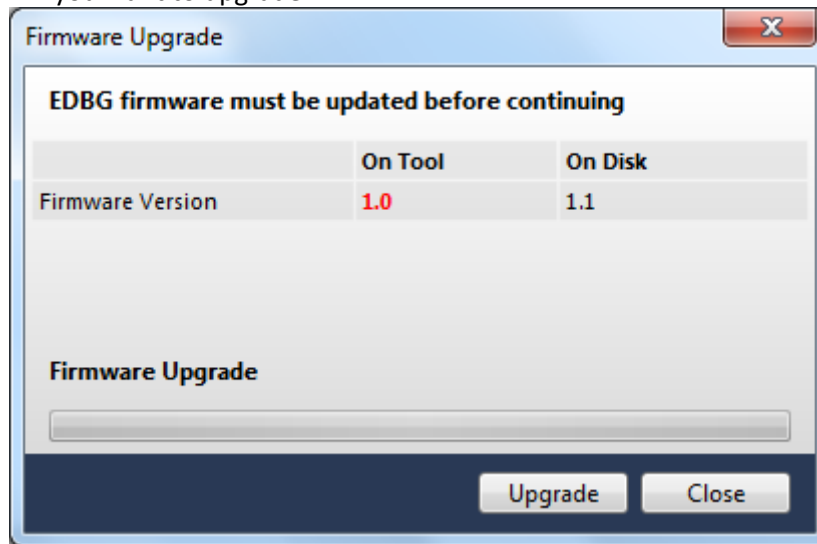


How to connect the AVR Xplained Mini to the Arduino IDE

- 1** Download the Arduino IDE from [Arduino.cc](http://arduino.cc)
- 2** Upgrade the mEDBG firmware on the AVR Xplained Mini to latest revision.

1. If you have an Atmel Studio release 6.2.1153 or later
Download the “medbgdebugger” package from Atmel spaces and follow the instructions
http://spaces.atmel.com/gf/project/avr_xp_mini/frs/
2. Start Atmel Studio
3. Connect the AVR Xplained Mini to the computer.
4. In Atmel Studio, select Tools – Device programming (Ctrl – Shift – P)
5. In the Device Programming window, select Tool to mEDBG and click Apply, Studio will now ask you if you want to upgrade.



3a Set the bootloader fuses in ATmega328P

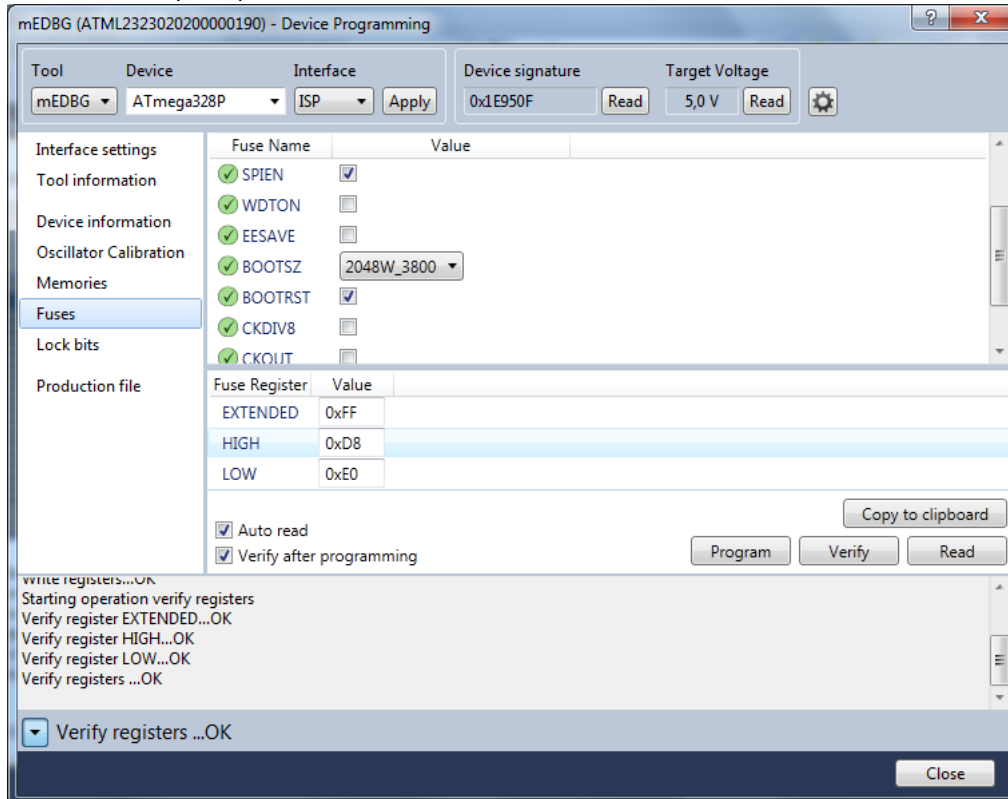
1. Start Atmel Studio
2. Connect the AVR Xplained Mini to the computer.
3. In Atmel Studio, select Tools – Device programming (Ctrl – Shift – P)
4. In the Device Programming window, select Tool to mEDBG and click Apply.
5. Select Fuses
6. Change value on EXTENDED, HIGH, and LOW and click Program

Select Bootloader Address as Reset Vector and select largest boot size

EXTENDED = 0xFF (valid)

HIGH = 0xD8 (valid)

LOW = 0xE0 (valid)



3b Set the bootloader fuses in ATmega168PB,

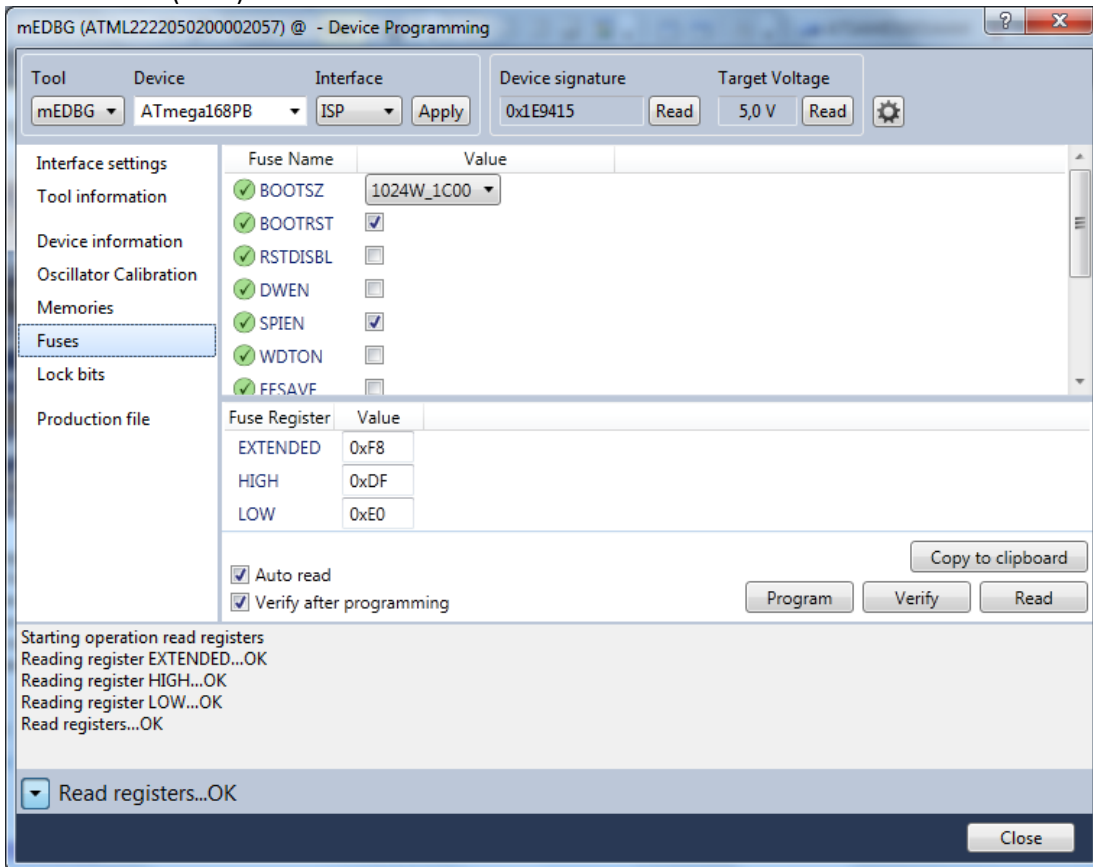
1. Start Atmel Studio
2. Connect the ATmega168PB-XMINI to the computer.
3. In Atmel Studio, select Tools – Device programming (Ctrl – Shift – P)
4. In the Device Programming window, select Tool to mEDBG and click Apply.
5. Select Fuses
6. Change value on EXTENDED, HIGH, and LOW and click Program

Select Bootloader Address as Reset Vector and select largest boot size

EXTENDED = 0xF8 (valid)

HIGH = 0xDF (valid)

LOW = 0xE0 (valid)



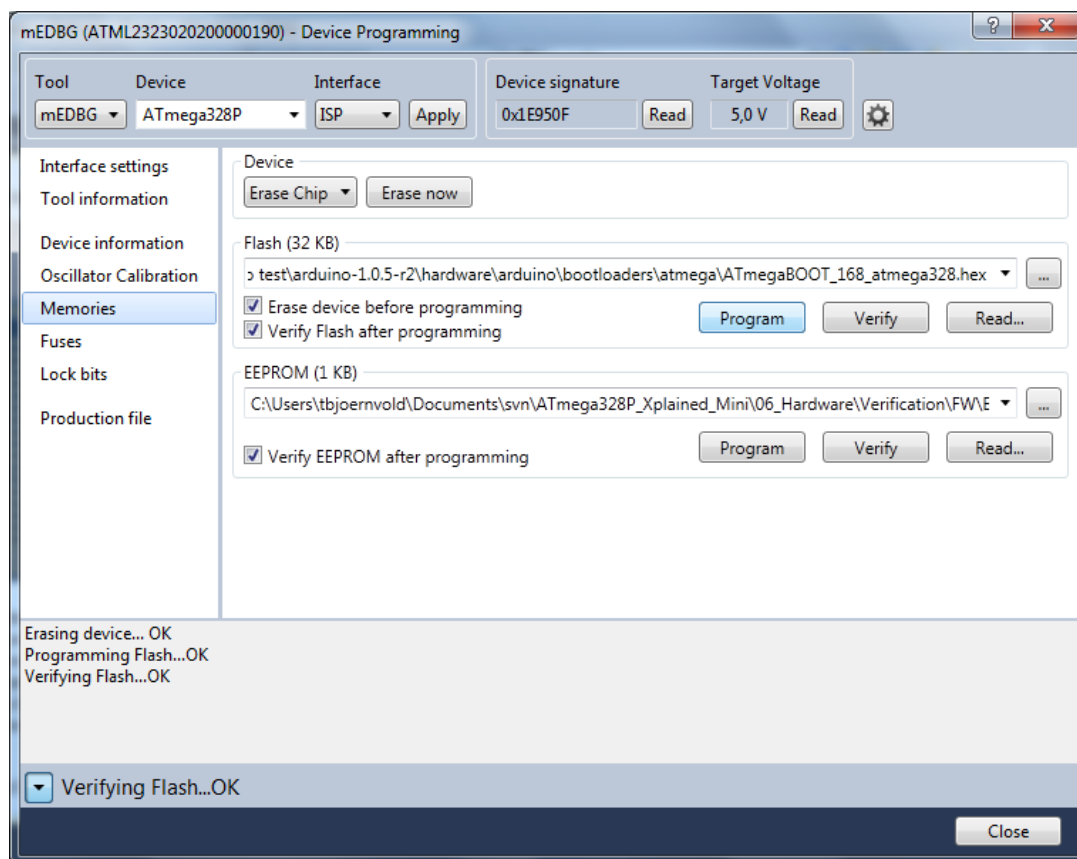
4 Program the ATmega328P with the Arduino bootloader

The bootloader hex file is located in the Arduino IDE folder:

/hardware/arduino/bootloaders/atmega/*.hex

Select bootloader according to your board configuration as listed in table below:

1. Start Atmel Studio
2. Connect the AVR Xplained Mini to the computer.
3. In Atmel Studio, select Tools – Device programming (Ctrl – Shift – P)
4. In the Device Programming window, select Tool to mEDBG and click Apply.
5. Select Memories
6. Browse for file
7. Click Program



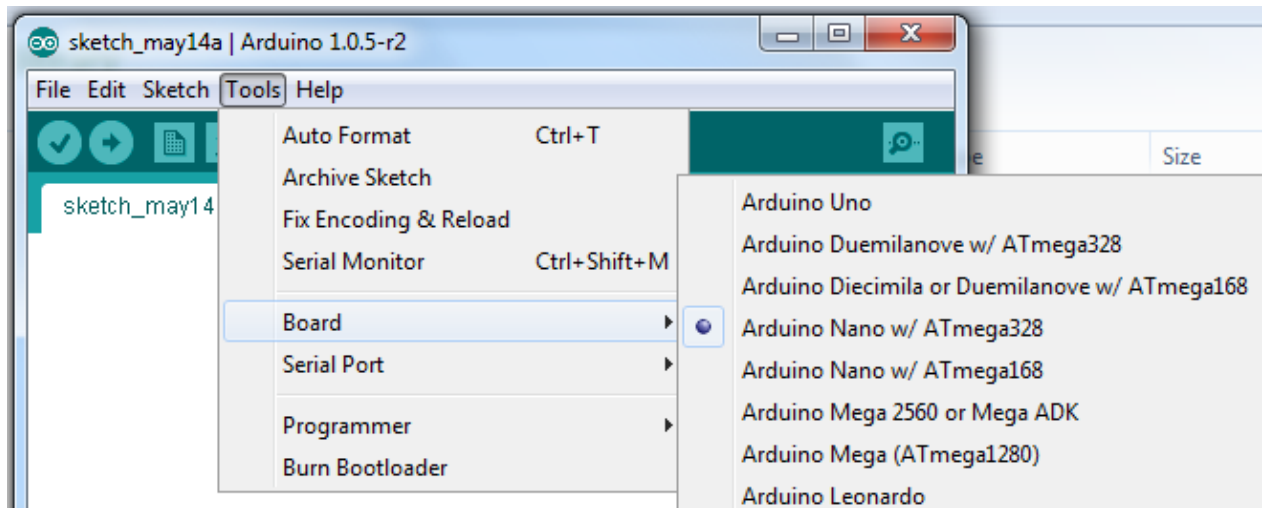
Xplained Mini

Bootloader

ATmega328P/5V/16MHz	ATmegaBOOT_168_atmega328.hex
ATmega328P/3.3V/8MHz	
ATmega168PB/5V/16MHz	ATmegaBOOT_168_ng.hex
ATmega168PB/3.3V/8MHz	ATmegaBOOT_168_pro_8MHz.hex

5 Configure the Arduino IDE

1. Start the Arduino IDE
2. select the mEDBG COM port
3. Select board according to table below



Xplained Mini

Board to select in Arduino IDE

ATmega328P-XMINI

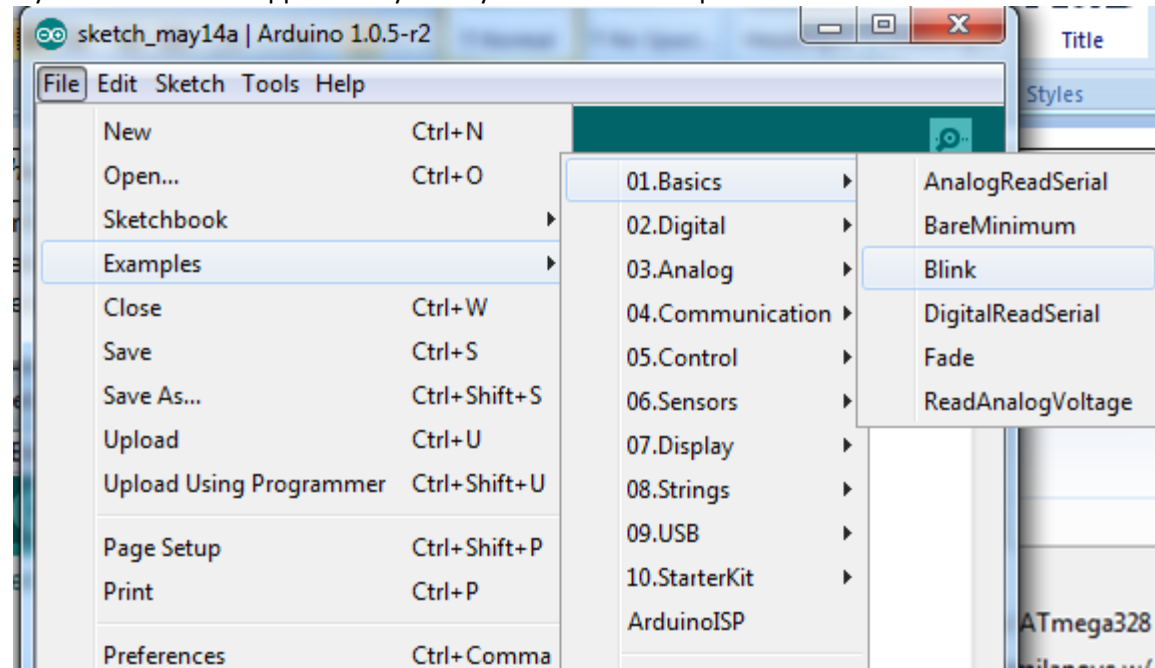
Arduino Nano w/ATmega328

ATmega168PB-XMINI

Arduino Nano w/ATmega168

6 Get your application

If you don't have an application yet – try the “Blink” example:



8 Use the Arduino IDE and upload your program to the AVR Xplained Mini

Upload the code:



The screenshot shows the Arduino IDE window titled "Blink | Arduino 1.0.5-r2". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for opening files, saving, uploading, and downloading. The main text area displays the Blink sketch code, which includes comments and C++ code for setting up and looping the LED. At the bottom, a status bar indicates "Done uploading." and "Binary sketch size: 1 084 bytes (of a 30 720 byte maximum)". The bottom status bar also shows "1" and "Arduino Nano w/ ATmega328 on COM87".

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
  */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

Done uploading.

Binary sketch size: 1 084 bytes (of a 30 720 byte maximum)

1 Arduino Nano w/ ATmega328 on COM87