USER GUIDE

How to update ATmega328P Xplained Mini for use with the Arduino IDE

ATmega328P Xplained Mini

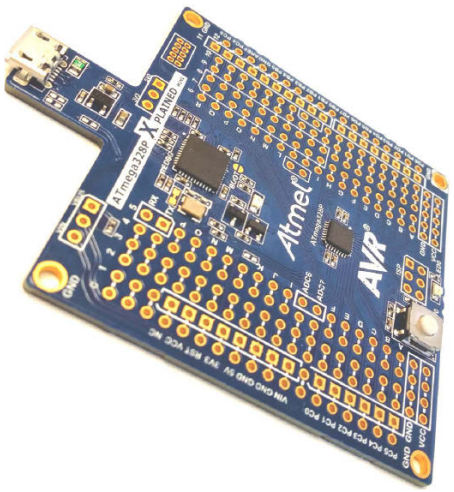
Introduction

This document explains how to update the firmware on the ATmega328P Xplained Mini to be able to use the kit directly with the Arduino IDE.

After updating the ATmega328P Xplained Mini kit, it will still be recognized as normal in Atmel® Studio 6.x, and one will still have full programming and debugging support in Atmel Studio.

Features

- One-click installer for making the ATmega328P compatible with the Arduino IDE without having to install Atmel Studio 6.x
- Installs Atmel Command Line Tools for command line programming and tool firmware update (can be uninstalled after the firmware update has been completed)
- Updates the mEDBG (embedded debugger) with the latest firmware (version 1.1) to allow for larger data packets on the virtual COM-port (mandatory for the Arduino bootloader to work)
- Writes a modified Arduino compatible bootloader to the ATmega328P



1 Downloading the Installer

The installer file is available from the project folder on Atmel Spaces:

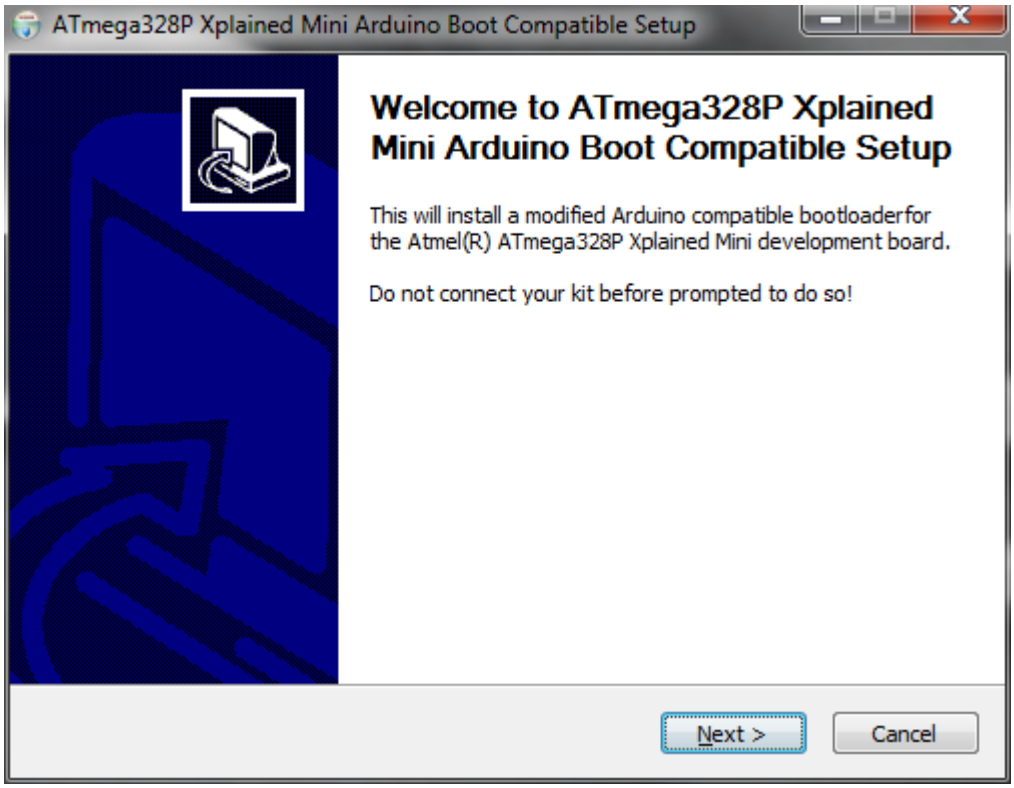
http://spaces.atmel.com/gf/project/avr_xp_mini/frs/

Download the file “Burn ATmega328P Xplained Arduino Bootloader.exe” and save it to an easily found location on the hard disk, e.g. your Desktop.

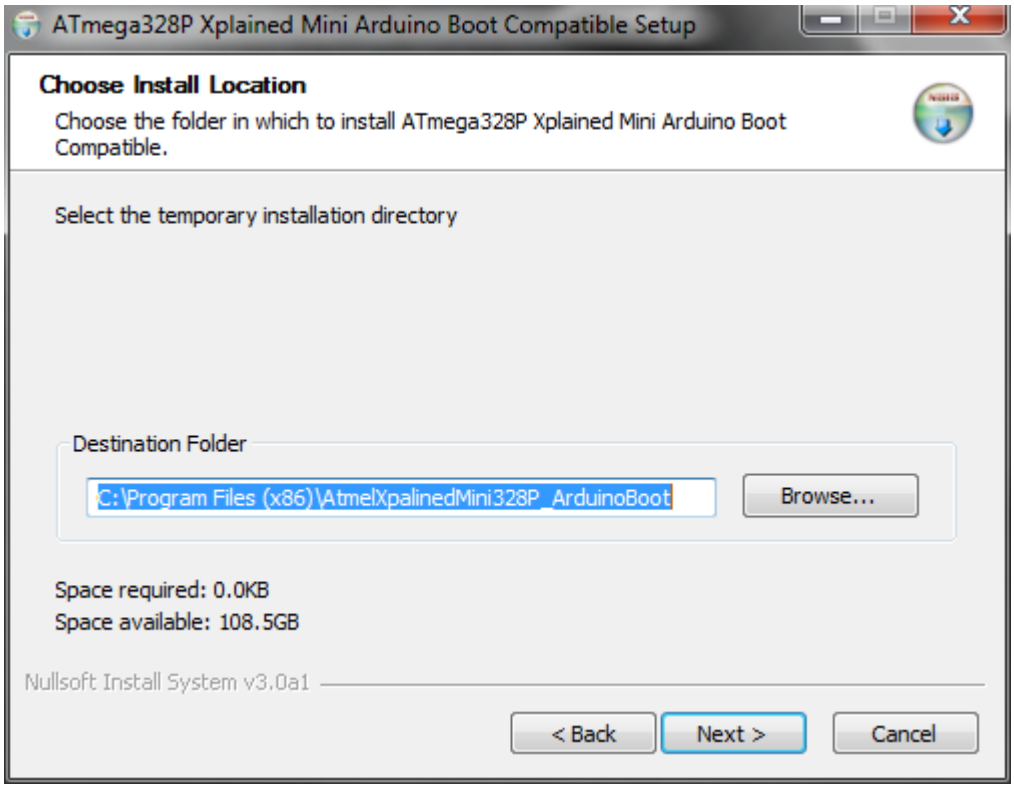
2 Running the Installer

Note: Do not connect your kit to the computer before prompted to do so by the installer.

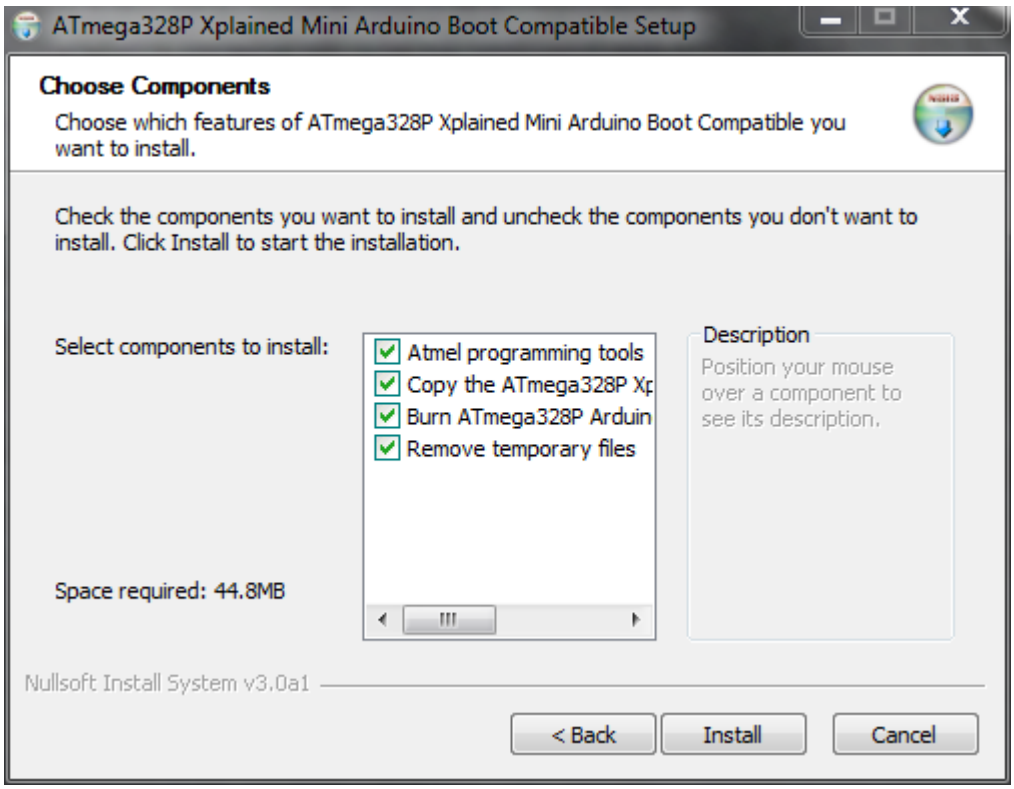
Double click the file “Burn ATmega328P Xplained Arduino Bootloader.exe” and follow the instructions give on the screen:



Select a location for where the (temporary) files are installed and click “Next”:



The installer will detect what components that may already be previously installed, as this installer relay on software which may or may not have been installed through other software packages and it will check the appropriate options in the list:



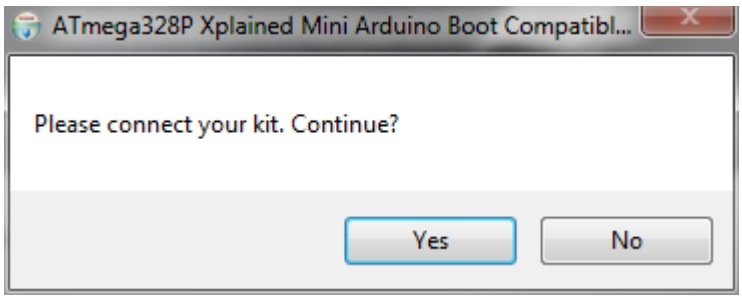
There are four different software packages/steps in the installer:

- “Atmel programming tools”
  - This will install command line tools for doing Atmel Tools firmware upgrades (e.g. JTAGICE3, mEDBG, EDBG, …) and device programming, and also contains the USB drivers
- “Copy the ATmega328P Xplained…”
  - This will copy the firmware file and bootloader hex file to the temporary location, and making this available for firmware updating and programming tools for the next step
- “Burn the ATmega328P Arduino…”
  - This will not install any file on the computer, but run a .bat script which will:
    1. Check the current firmware version of the mEDBG on the ATmega328P Xplained Mini and update it to version 1.1 if the current version is lower than this.
    2. Set the correct fuses for the ATmega328P to boot from the bootloader section and adjust the size of the section (same fuses as ATmega328P on the standard Arduino UNO board).
    3. Erase, program and verify the Arduino compatible bootloader on to the ATmega328P.

    Note: Existing firmware on the ATmega328P will be erased in this process.
- “Remove temporary files”
  - This will remove all the files used during the firmware update and programming of the device after the installation is completed, this will typically be used when you want to update a single board once
  - If you want to re-run the firmware update for several kits keep this option unchecked, this way files do not need to be reinstalled on your computer each time you want to update a kit

    Note: Atmel Command Line Tools will not be uninstalled, to uninstall these tools one has to use the Add Remove Programs function for your OS version.

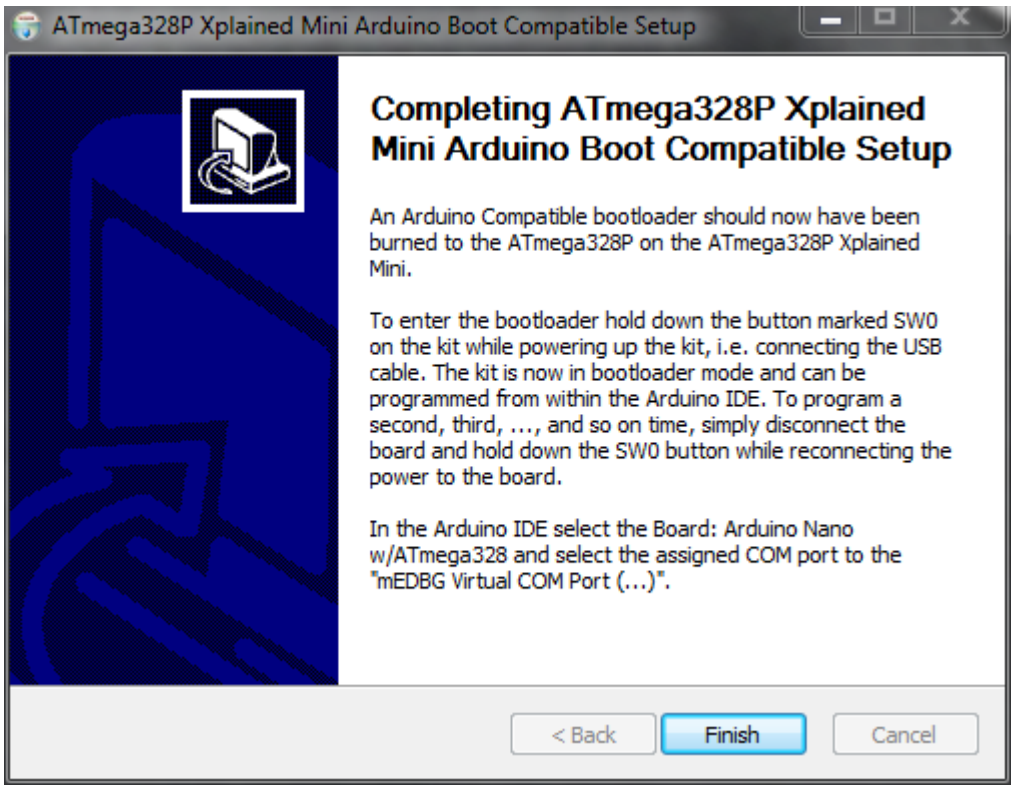
Once ready to proceed click “Install”. Depending on the options chosen it will install the different packages. At a point in the process you will be asked to connect the kit:



Connect the kit to the computer using a micro USB cable and wait for your OS to detect the kit and install the necessary drivers, once done click “Yes”. A command line window will now pop-up and program the firmware and bootloader to the kit, if everything goes ok you should see a similar result as shown below:

You might see an error regarding “sleep” not being a recognized command, this can in most cases be ignored, but might in some cases cause issues with programming the bootloader due to that Windows® may not have had enough time to enumerate the USB device before trying to contact the board, a simple solution is then to re-run the installer and the issue should be gone, as this should only be an issue when the mEDBG firmware is updated.

When the installer has completed the following screen comes up:



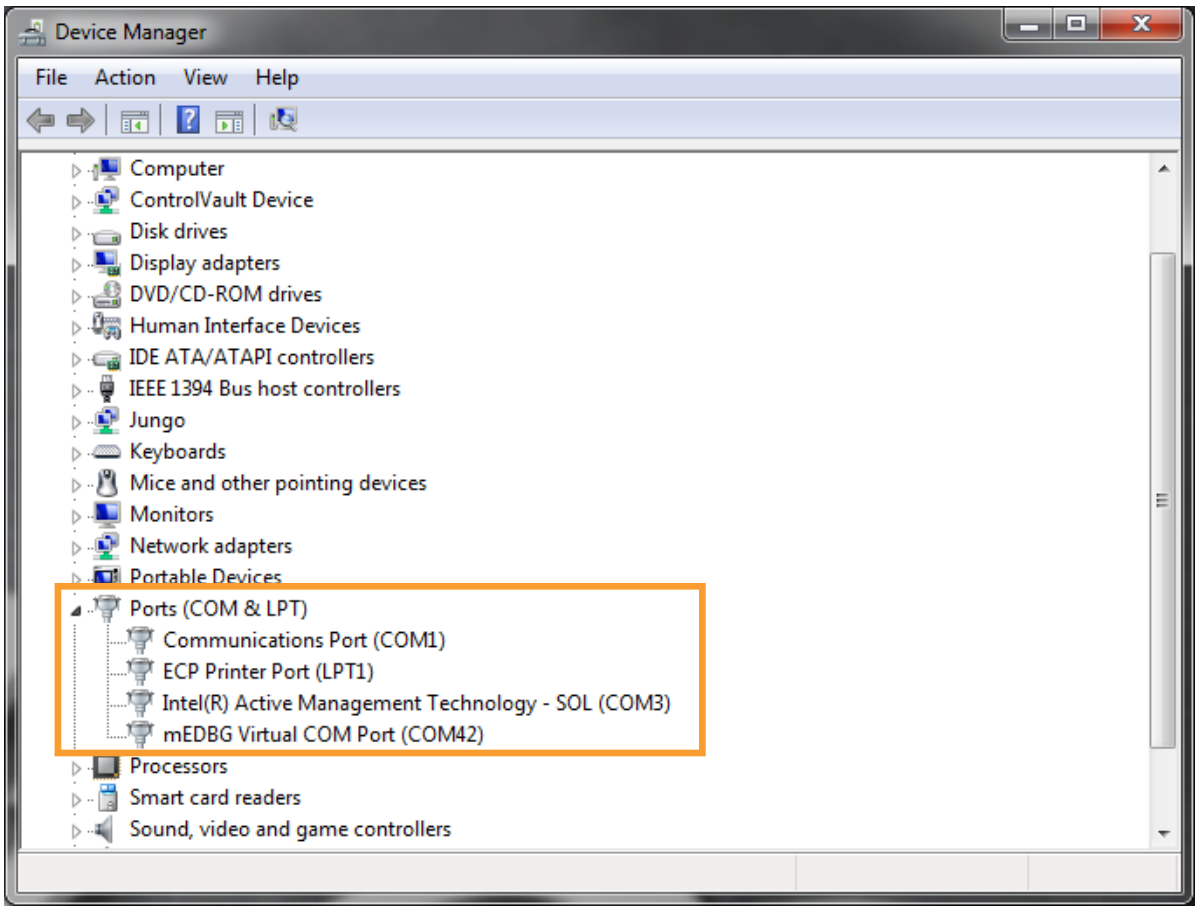
3 Programming the ATmega328P from the Arduino IDE

3.1 Detecting which COM-Port to Use

The kit will enumerate as a debugger and programmer for use in Atmel Studio and a “mEDBG Virtual COM Port (…) to communicate with the ATmega328P directly through a terminal program (e.g. putty (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html), or in this case the Arduino IDE).

Start the “Device Manager”, either by hitting the Windows button and writing “Device Manager”, if you are using Microsoft® Windows 7, or start the device manager via the Control Panel.

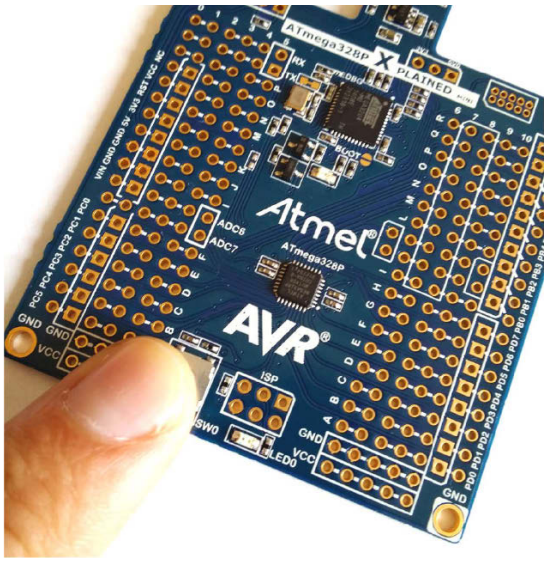
Locate the “Ports (COM & LPT)”-entry in the list:



Locate the entry “mEDBG Virtual COM Port (COMxx)” where “xx” represents the COM port number that has been assigned for the kit. This is the COM port number which will be used in the Arduino IDE.

3.2 Put the Kit in Bootloader Mode

Each time a sketch should be uploaded from the Arduino IDE the kit needs to manually be put into bootloader mode. This is done by holding down the button marked SW0 while powering the kit (typically when connecting the USB cable), (or using the suggested tip in Chapter 4 Tips & Tricks).

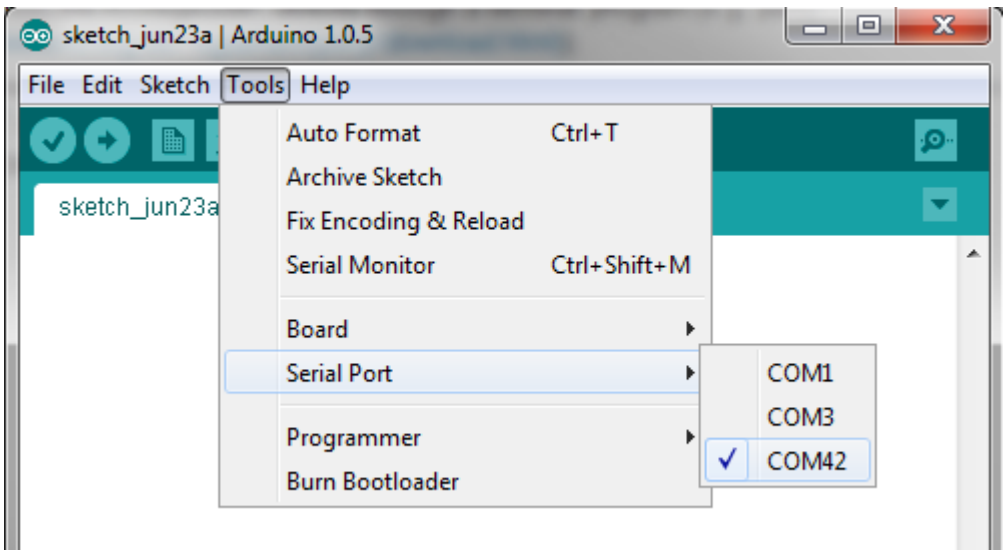


The kit is now in bootloader mode and can be used from within the Arduino IDE.

3.3 Configuring the Arduino IDE

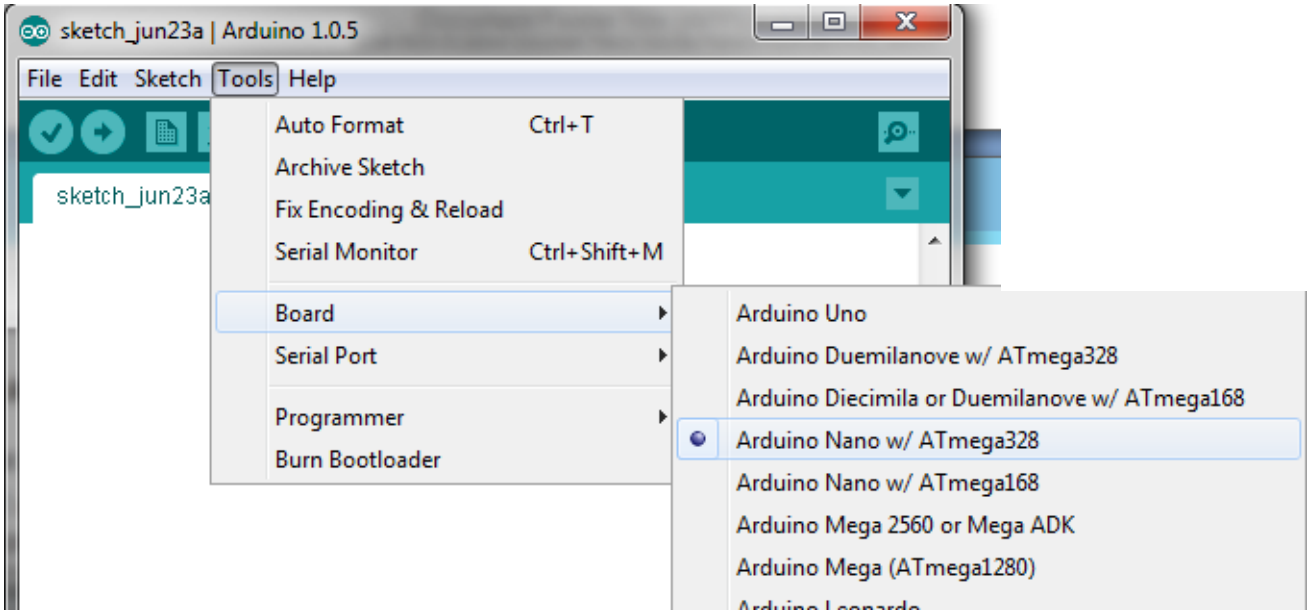
3.3.1 Selecting the Correct COM Port

Start the Arduino IDE and select File menu -> Tools -> Serial Port and select the COM port which is assigned to the connected ATmega328P Xplained Mini kit (found in the step above):



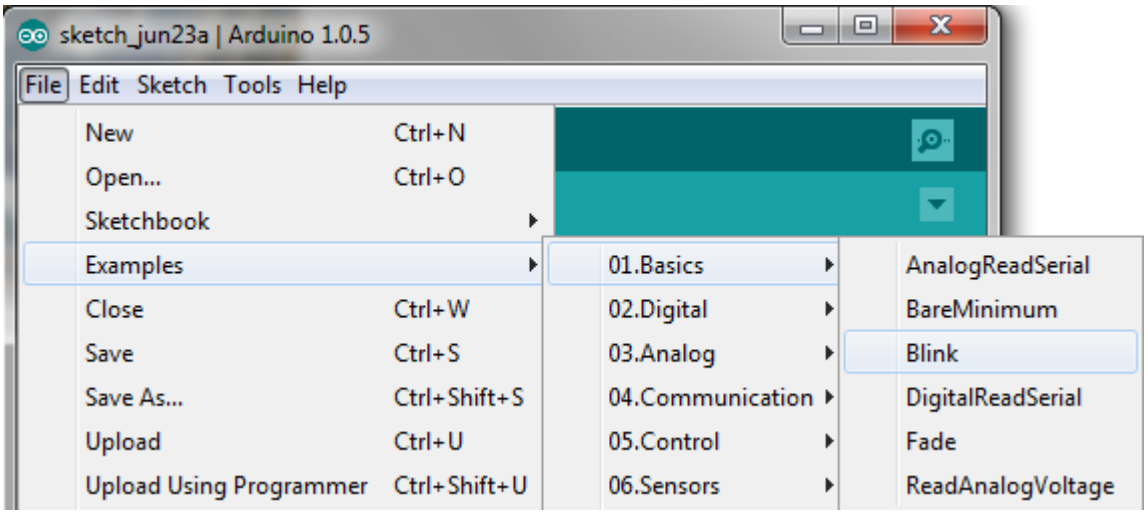
3.3.2 Selecting the Correct Board

In the File menu select Tools -> Board -> Arduino Nano w/ATmega328:

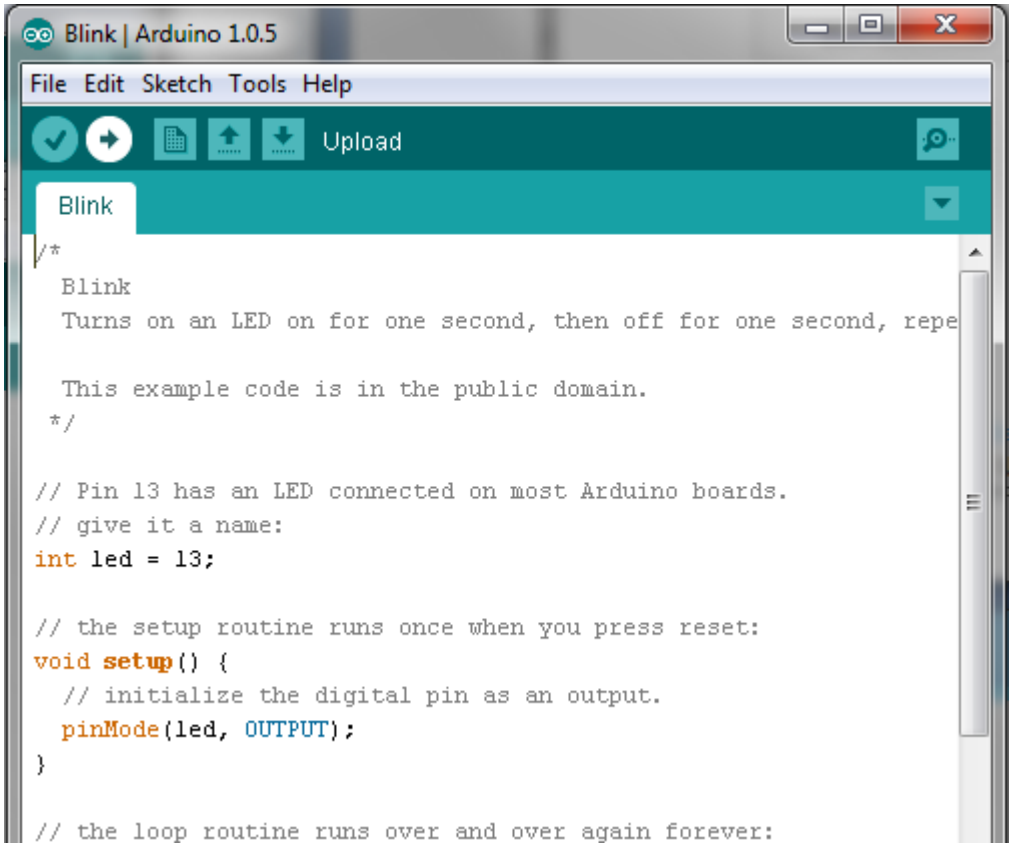


3.3.3 Testing that Everything Works

The kit can now be programmed from the Arduino IDE by pressing the Upload button:  . But before one do that one need to have some code, the easiest thing to test is using one of the examples included in the Arduino IDE. For this purpose the “Blink” example will do fine:



Once loaded hit the Upload button:  and the program will be uploaded to the kit:



If everything works as it should a led, marked LED0, on the ATmega328P Xplained Mini should start to blink with one second on and one second off. To re-upload the same sketch or a new sketch one has to put the board into bootloader mode again using the steps described in Section 3.2.

4 Tips & Tricks

4.1 Entering the Bootloader from Software

If you do not need the button, marked SW0, on the board to other things in your application you can use it to enter bootloader mode without powering off and on the board. This has been added to the standard Arduino Blink example below:

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

  This example code is in the public domain.
 */

/* Prototype for the bootloader function */
void (*bootloader)(void) = (void (*)(void))0x3800;

// 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

  if (!(PINB & (1 << 7))) {
    bootloader();
  }
}
```

4.2 Reprogramming the Bootloader with Atmel Studio

The bootloader can also be programmed using Atmel Studio, the hex file to use is located under: C:\Program Files (x86)\AtmelXpalinedMini328P_ArduinoBoot (if the default location has been used and the option “Remove temporary files” has not been used during installation). The file name is: “ATmegaBOOT_168_atmega328.hex” and can be programmed using the Atmel Studio programming interface, to see how to program a device from an existing hex file refer to the Atmel Studio help.

5 Revision History

Doc Rev.	Date	Comments
42331A	07/2014	Initial document release.

Atmel | Enabling Unlimited Possibilities®

Atmel Corporation 1600 Technology Drive, San Jose, CA 95110 USA **T:** (+1)(408) 441.0311 **F:** (+1)(408) 436.4200 | **www.atmel.com**