Arduino bootloader with watchdog timer support http://www.microfarad.de

15 commits 1 branch 0 releases 1 contributor GPL-2.0

Branch: master New pull request Find file Clone or download

microfarad-de Update comments Latest commit d0a29fc on 18 Jun

.gitattributes	Initial commit	5 months ago
.gitignore	Initial commit	5 months ago
ATmegaBOOT_168.c	Update comments	4 months ago
ATmegaBOOT_168_atmega328.hex	Added bootloader HEX files	4 months ago
ATmegaBOOT_168_atmega328_pro_8MHz.hex	Added bootloader HEX files	4 months ago
LICENSE	Initial commit	5 months ago
README.md	Update README.md	4 months ago

README.md

Arduino Bootloader with Watchdog Timer Support

This repository contains a customized version of the Arduino bootloader firmware with watchdog timer (WDT) support.

The stock Arduino bootloader introduces a couple of seconds of delay during boot in order to provide the Arduino IDE with enough time to start uploading a new sketch. This has the following disadvantages:

- The Arduino waits for a couple of seconds before starting to execute the main sketch upon initial power on.
- The Arduino gets stuck in an endless loop following a watchdog timer expiry. The delay causes the WDT to expire before the main sketch is able to load and perform a WDT reset.

With this custom bootloader, the aforementioned delay is introduced only if the reboot reason is due to pulling the external reset pin to ground. A delay is no longer introduced during initial power on or a reboot due to a watchdog timer expiry, which brings the following benefits:

- The Arduino boots-up instantly after initial power on.
- The Arduino is no longer stuck in an endless loop following a WDT expiry.

The bootloader modification consists of defining `WATCHDOG_MODS` macro which activates the required functionality that has been already implemented within the stock bootloader.

This modification has been tested on an ATmega328P based Arduino Pro Mini and would supposedly work on any ATmega328 based arduino board.

Burning the Bootloader

Following are the instructions for burning the bootloader.

The ATmega bootloader files are stored in the following locations:

- On MAC OS X: `/Applications/Arduino.app/Contents/Java/hardware/arduino/avr/bootloaders/atmega`
- On Windows: `C:\Program Files (x86)\Arduino\hardware\arduino\avr\bootloaders\atmega`

Replace the following files in the above location with the versions from this repository, creating a backup copy of the original files:

- For ATmega328P at 16MHz: `ATmegaBOOT_168_atmega328.hex`
- For ATmega328P at 8MHz: `ATmegaBOOT_168_atmega328_pro_8MHz.hex`

Follow these steps to upload the bootloader to the Arduino board:

- Connect your Arduino board with your PC via an in-system programmer (ISP)
- Start your Arduino IDE and select your ISP model inside: Tools > Programmer
- Select your board model inside: Tools > Board
- Select your processor model inside: Tools > Processor
- Select the serial port name inside: Tools > Port
- Burn the bootloader by selecting: Tools > Burn Bootloader

Compiling the Bootloader

Alternatively, you may compile the bootloader `.hex` files from the souce code provided in this repository. Only instructions for MAC OS X are currently provided as this happens to be the platform used by the author. This procedure has been tested with Arduino IDE version 1.8.8.

Please follow these steps:

- Install the Arduino IDE
- Install XCode from the App Store
- Open Terminal and go to the following path: `/Applications/Arduino.app/Contents/Java/hardware/arduino/avr/bootloaders/atmega`
- Make a backup copy of and replace the following file with the version from this repository: `ATmegaBOOT_168.c`

For ATmega328P at 16MHz:

- Make a backup copy and remove the following file: `ATmegaBOOT_168_atmega328.hex`
- Execute: `make atmega328`

For ATmega328P at 8MHz:

- Make a backup copy and remove the following file: `ATmegaBOOT_168_atmega328_pro_8MHz.hex`
- Execute: `make atmega328_pro8`

For other platforms:

- Within the `Makefile` you can find further targets for compiling bootloaders for other platform types.