Switchboard: Preparations for installing the Switchboard firmware on an Arduino Micro

Author: Valentin Py, Djen Kühnel
Date: 03.09.2020
This applies to the Arduino Micro board

1. Flashing bootloader:

Arduino micro uses a bootloader which runs during boot and handles flashing the board over USB. The default bootloader causes problems because it flashes two LEDs which are connected to some of the same pins the Switchboard uses to control the high voltage output. This could result in high voltage being applied to the output unintentionally during flashing of the firmware or startup of the Arduino.

How to flash bootloader:

A modified bootloader is provided based on Katiana (https://github.com/aweatherguy/Katiana).

The pre-compiled bootloader is the file "Katiana.hex". It has to be flashed using a different procedure as flashing a normal application on an Arduino. Instructions are below (for more information, see here: https://www.arduino.cc/en/Tutorial/ArduinoISP).

Required Hardware:

- Arduino Uno (programmer)
- Arduino Micro (to be programmed)
- USB cable for the Arduino Uno
- 6 Jumper cables (male-to-female or female-to-female)

Flashing Arduino Uno

- Using Arduino IDE, open example "ArduinoISP" (File=>Examples=>11.ArduinoISP)
- make sure that the following defines are correct and not commented out:

```
#define RESET     10 // Use pin 10 to reset the target rather than SS
#ifdef USE_OLD_STYLE_WIRING
#define PIN_MOSI   11
#define PIN_MISO   12
#define PIN_SCK        13
```

- Select board (**Arduino Uno**! We are not using the Arduino micro yet!!)
- Select correct COM Port
- Compile and Launch the program
- Check that it compiled and uploaded correctly, then close Arduino IDE

Connect Arduino Uno to Arduino Micro

Connect the following pins on the Arduino Uno and the Arduino Micro (6 jumper cables required):

Arduino Uno	Arduino Micro	(recommended color)
5V	5V	RED
GND	GND	BLACK
Pin 10	RST	YELLOW
Pin 11	MOSI	BLUE
Pin 12	MISO	GRAY
Pin 13	SCK	GREEN

Compilation of bootloaders

This step is only required is you don't find the file `Katiana.hex` in the folder: `Switchboard/tools/Katiana-bootloader/Katiana-Bootloader pre-compiled binary` or if you want to do some modifications

- Use a Linux PC or find a solution on Windows)
- Install required programs:
- `sudo apt update`
- `sudo apt install arduino arduino-mk`
- Using a terminal, go to folder `Switchboard/tools/Katiana-bootloader/lufa-LUFA-170418/lufa-LUFA-170418/Projects/Katiana`
- Launch following commands:
  `-make clean`
  `-make`
- You should now have a file named `Katiana.hex` in the current folder, copy it to your PC

Burning bootloader:

- Connect the Arduino Uno using USB (Arduino Micro is not USB-connected, it will be powered by the Arduino Uno)
- Open a Windows Terminal (CMD or Git bash)
- Navigate to the folder where `Katiana.hex` is located.
- Launch the 2 following commands (replace `$USERNAME$` with your user name and make sure that `avrdude.exe` exists in the specified folder)
- `"C:\Users\$USERNAME$\AppData\Local\Arduino15\packages\arduino\tools\avrdude\6.3.0-arduino17/bin/avrdude" -C"C:\Users\$USERNAME$\AppData\Local\Arduino15\packages\arduino\tools\avrdude\6.3.0-arduino17/etc/avrdude.conf" -v -patmega32u4 -cstk500v1 -PCOM8 -b19200 -e -Ulock:w:0x3F:m -Uefuse:w:0xcb:m -Uhfuse:w:0xd8:m -Ulfuse:w:0xff:m`
- `"C:\Users\$USERNAME$\AppData\Local\Arduino15\packages\arduino\tools\avrdude\6.3.0-arduino17/bin/avrdude" -C"C:\Users\$USERNAME$\AppData\Local\Arduino15\packages\arduino\tools\avrdude\6.3.0-arduino17/etc/avrdude.conf" -v -patmega32u4 -cstk500v1 -PCOM8 -b19200 -Uflash:w:Katiana.hex:i -Ulock:w:0x2F:m`

Testing

- Unplug the boards
- Connect Arduino Micro using USB
- Make sure it's detected by Windows and that you can flash it using the Arduino IDE

2. Updating pins_arduino.h

On a standard Arduino board, the RX_LED and TX_LED flash during boot and whenever data is transferred via USB. Because some components of the Switchboard are connected to the same pins corresponding to those LEDs, the flashing needs to be disabled to ensure that no high voltage is generated unintentionally (same problem as with the default bootloader).

This can be done by modifying the file "pins_arduino.h" in the Arduino library used for the Arduino Micro board (be careful: a different file exists for each board).

The file should be located here (replace $USERNAME$ with your user name and adjust the version number `1.8.2` as required):
`C:/Users/$USERNAME$/AppData/Local/Arduino15/packages/arduino/hardware/avr/1.8.2/variants/micro/pins_arduino.h`

Edit the file and change the macros for **`TXLED0/TXLED1`** and **`RXLED0/RXLED1`** to **`void(0)`**.

This is what the contents of the modified file should look like:
~~~~
```
/*
  pins_arduino.h - Pin definition functions for Arduino
  Part of Arduino - http://www.arduino.cc/

  Copyright (c) 2007 David A. Mellis

  This library is free software; you can redistribute it and/or
  modify it under the terms of the GNU Lesser General Public
  License as published by the Free Software Foundation; either
  version 2.1 of the License, or (at your option) any later version.

  This library is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
  Lesser General Public License for more details.

  You should have received a copy of the GNU Lesser General
  Public License along with this library; if not, write to the
  Free Software Foundation, Inc., 59 Temple Place, Suite 330,
  Boston, MA  02111-1307  USA
*/

#include "../leonardo/pins_arduino.h"

#undef TXLED0
#undef TXLED1
#undef RXLED0
#undef RXLED1
#undef TX_RX_LED_INIT

#define TXLED0 void(0)
#define TXLED1   void(0)
#define RXLED0   void(0)
#define RXLED1   void(0)
#define TX_RX_LED_INIT void(0)
```
~~~~

This modification disables the macros used by the USB core (and others) and prevents the flashing of the LEDs.

Be careful: this constitutes a modification of the Arduino library and it will affect all Arduino Micro boards flashed with the computer where the modification was applied!

3. Install the Switchboard firmware

Simply connect the Arduino Micro via USB and flash the firmware using the Arduino IDE, same as any other Arduino program. This can now be done safely even when the Arduino is mounted on the Switchboard.