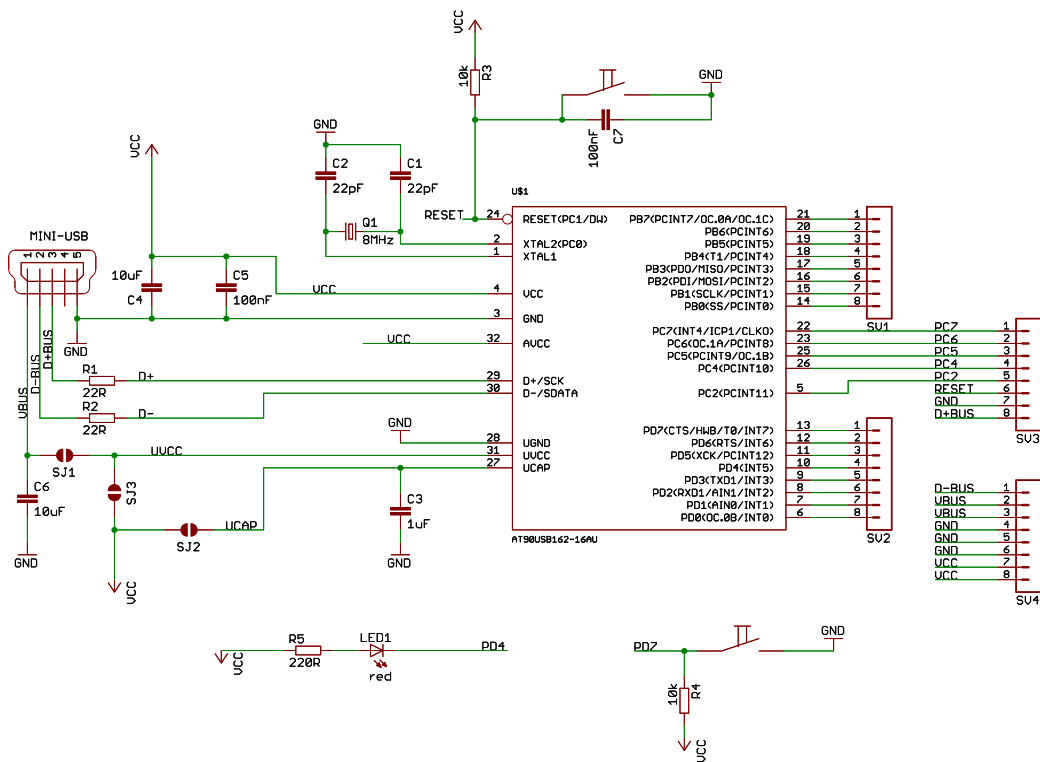


# 1 Overview

The USBfoo is a lightweight development board based on the Atmel AVR at90usb162 microcontroller. It can be configured for 5V and 3.3V operation, can be programmed via USB and features a LED as well as a reset- and a hardware-boot button, where the latter can be used as a regular push button as well.

# 2 Schematics



# 3 Power Configuration

Before using the board you need to configure the power supply for the board. It supports three basic modes, configured by closing two of the jumper pads on the bottom of the board.

- **5V USB:** The board is powered via the USB port. The logic level for the IO pins is 5V.
- **3.3V USB:** The board is powered via the USB port and the 3.3V voltage regulator embedded in the at90usb162. The logic level for the IO pins is

3.3V. Note that the amount of current available for the other peripherals is limited. Please refer to the at90usb162 datasheet for further information.

- 3.3V EXT: The board is powered via an external 3.3V power supply. Please note that you should disable the internal voltage regulator embedded in the at90usb162. Please refer to the at90usb162 datasheet for further information.

There is a legend to the jumper configuration on the bottom of the board.

## 4 Additional Hardware

The USBfoo features two buttons and a red LED. They are connected as follows:

- Reset-Button: connected to the Reset-pin of the at90usb162, also uses an external pull up resistor.
- HWB-Button: the hardware-boot button is connected to PD7, also using an external pull up resistor to ensure a stable reset behaviour. It can be used as a general purpose button by a custom firmware.
- Red LED: the LED is connected to PD4. To switch it on you need to configure PD4 as output and set it to 0.

## 5 Programming the USBfoo

The USBfoo can be programmed via USB. As programming software you can for example use the FLIP software from Atmel or the “dfu-programmer” command line utility. Note that for programming via USB with the default Bootloader as shipped by Atmel you need to first do a chip erase before flashing the new firmware.

After flashing a custom firmware the new firmware gets executed immediately. To enter the bootloader again do the following:

- push down the reset button
- push down the HWB button
- release the reset button
- release the HWB button

The USBfoo then enumerates as a DFU device again and can be reprogrammed using FLIP or dfu-programmer.