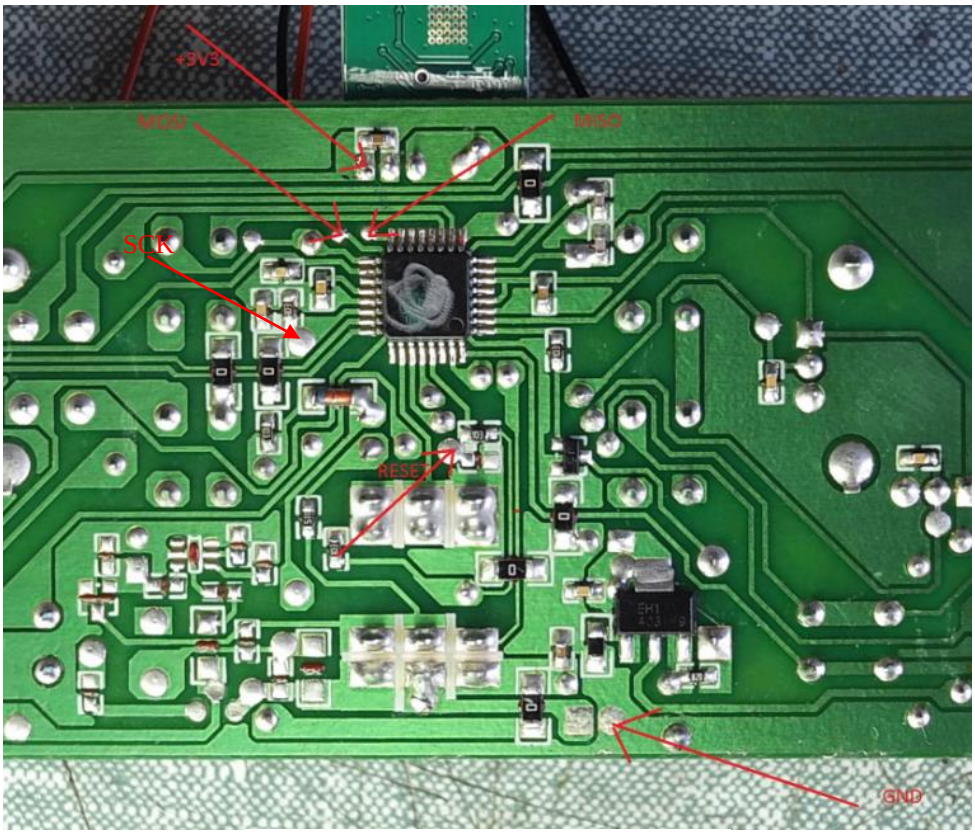
# MLP4DSM

Flashing the alternative Firmware




# Foreword

This Firmware was mainly made to fly Blade Nano CP X.

The Nano CP X is so beautiful small that it fits well into travel luggage.
Unfortunately there is no small transmitter available that can be used to fly the Nano CP X.
The smallest transmitter that works for that is the DX4e and that one only in “none computer mode”.
The smallest TX for computer mode is the DX6i.
Both transmitters are to large to fit in travel luggage.

The MLP4DSM would be more the right size. But with the MLP4DSM you can’t get the Nano into “nono computer mode” and in computer mode the pitch isn’t working.
Only the manufactory knows why, it is working with DX4e.

For a long time I saw no chance to change this and instead took the  mSR X with me when on travel.

Someday on Google I found a project with a custom firmware for the DX4e.
That may be an option, I thought.
A Google search for an alternative Firmware for  MLP4DSM did not find annyting.
But, at least I found a (nearly complete) Schematic of the MLP4DSM.
Also mal nach einer Firmware für die MLP4DSM gesucht, leider Fehl
http://www.baronerosso.it/forum/attachments/elimodellismo-micromodelli/220463d1327353167-blade-mqx-ultra-micro-quad-copter-mlp4dsm.pdf

So it has an ATMEGA. That is a Start.
A short test with AVRDUDE „auto detect“ tells me that it is an ATmega 88p
As known for the arduino project there is a free C compiler for the AVR microcontroller available.

To be sure I scanned and analyzed the layout of the PCB.
The schematic from WEB is accurate and the 2 missing Signals (PY &PP) where found fast as well.

I made this firmware from scratch. Actually that worked out quite fast.
Only the flash memory size of only 8K was quite challenging.
At the end, I spend more time on optimizing and rearranging the source to fit into 8K than for programming the functionality.
Currently 95.7% of Flash is used. But during development it happens 3 times that I ran out of flash while still not having all functionality I wanted to implement.
Airplane mode is not tested on air, I have no airplanes.

# Table of Contents

# Image Index

# 1 Introduction

Du to it is required here some Notes.

In any case you shall that chapter.

Do not update your firmware before you have read this chapter.

## 1.1 Legal notice

By installing this firmware you are the creator of the new transmitter (new functionality)

So you are the manufacture of that new transmitter.

If you sell or give away that Transmitter you are also the distributor.

That has some legal consequences

- You can't hold the original manufacturer, distributor or sales man reliable for anything.
- All that the manufacturer, distributor or sales man is liable for are now on you.

## 1.2 Liable exclusion

I will and can't guarantee anything.

I do not accept liability for anything. Neither can I guarantee the functionality, nor will I accept liability for any direct or indirect damage that may be caused by using this firmware.

## 1.3 Confirmation

To proceed you have to confirm that I refuse any liability.

Do not install this firmware if you do not accept the points 1.1 or 1.2.

## 1.4 Risks

- You can't hold the original manufacturer, distributor or sales man reliable for anything.
- Even if very seldom, there is a chance to kill the Atmel when flashing. Most likely due to wrong Programming Voltage or bad protection against electrostatic.
- The Firmware for the RF module stays unchanged, so the RF properties of the transmitter should be unchanged. But without expansive Lab testing you can't guarantee that.
  There is a theoretical chance that the transmitter does not met the local RF regulation after the firmware update.
- If you are going to use a USB to Serial converter to monitor the serial data stream to the RF module, make sure your converter das TTL or 3.3V signal level. Regular RS232 signal levels (+/-12V) will kill the microcontroller.

# 2 Before starting

## 2.1 Check / verify PCB layout

Please make sure that the PCB of your MLP4DSM equal to the pictures below.

If your PCB does not match the pictures below, you must make sure that the Schematic of your transmitter is unchanged. In doubt do not install this firmware.

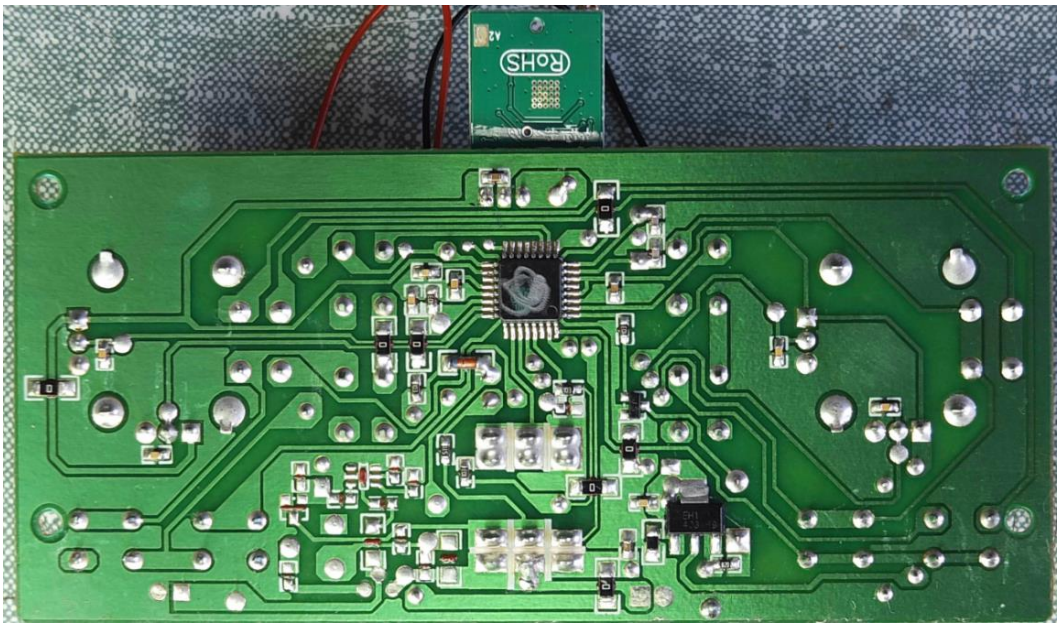


Image 1 Layout side of MLP4DSM PCB.

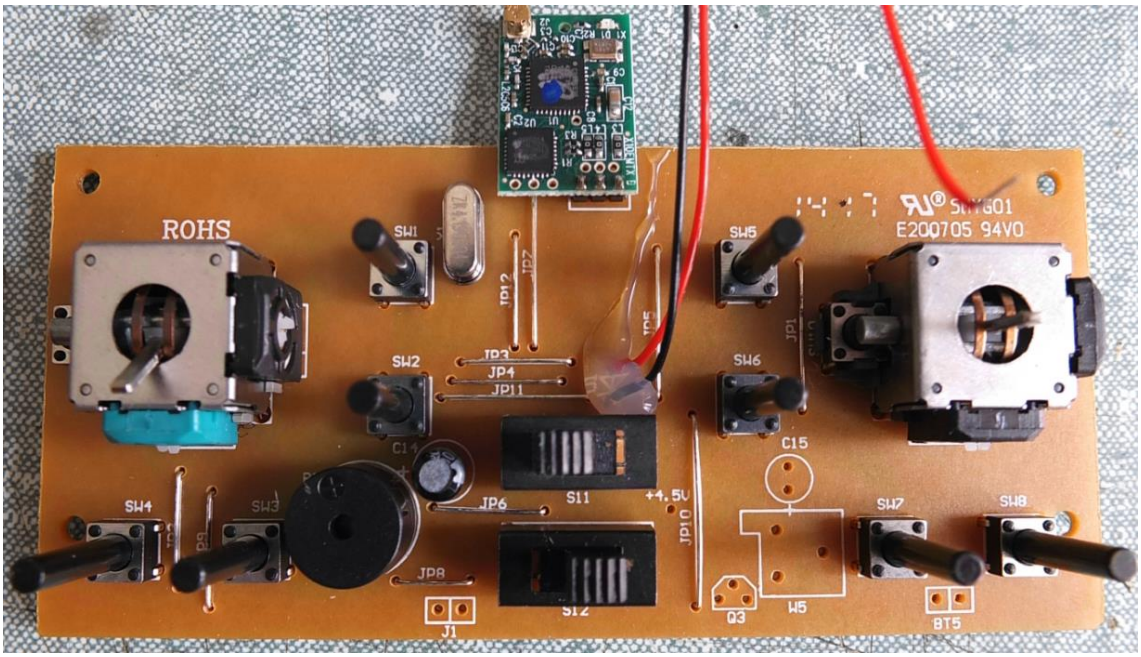



Image 2 Component side of MLP4DSM PCB.

Especially make sure that the crystal has a frequency of 4.194304Mhz.

If that is not the case, you must change and compile the source code first.

There are, at least, two versions of the PCB around. The other version of the PCB not only has a different crystal it is also using a different baud rate to the RF module.
The version with the 4.194304Mhz crystal is using 131.000 baud, the other version is using 135000 baud.

## 2.2 Requirements

To perform the firmware updates some prerequisites must be true.
This section will list these prerequisites.
You need:

- An AVR programmer.
  If you already have a programmer you can use that one.
  If you have a arduino you can convert that into an ISP programmer (arduino isp).
  Or you buy yourself a cheap programmer: I for example use a cheap USBASP Clone.
- A AVR programming Software.
  I use AVRDUDE
- 3-4 Wires to hook up your PCB to the programmer.
- Soldering equipment
- PC to run the Programming Software.

You should:

- Be able to solder.
  If you are uncomfortable with solder please train soldering on an old PCB first.
- Not having two left hands.
- Have a solid understanding of protection against electrostatic.
  Electrostatics can damage your circuits.

Optional:
During development and for Stick calibration you may find it useful to analyse the serial data stream the transmitter sends to the RF module. If you have a FTDI USB to TTL Serial converter you can use the „DSM_Serial_Analyse“ program.
Currently this program only supports FTDI based USB to serial converters, on the other hand, a FT232RL based converter can be source for about 3€

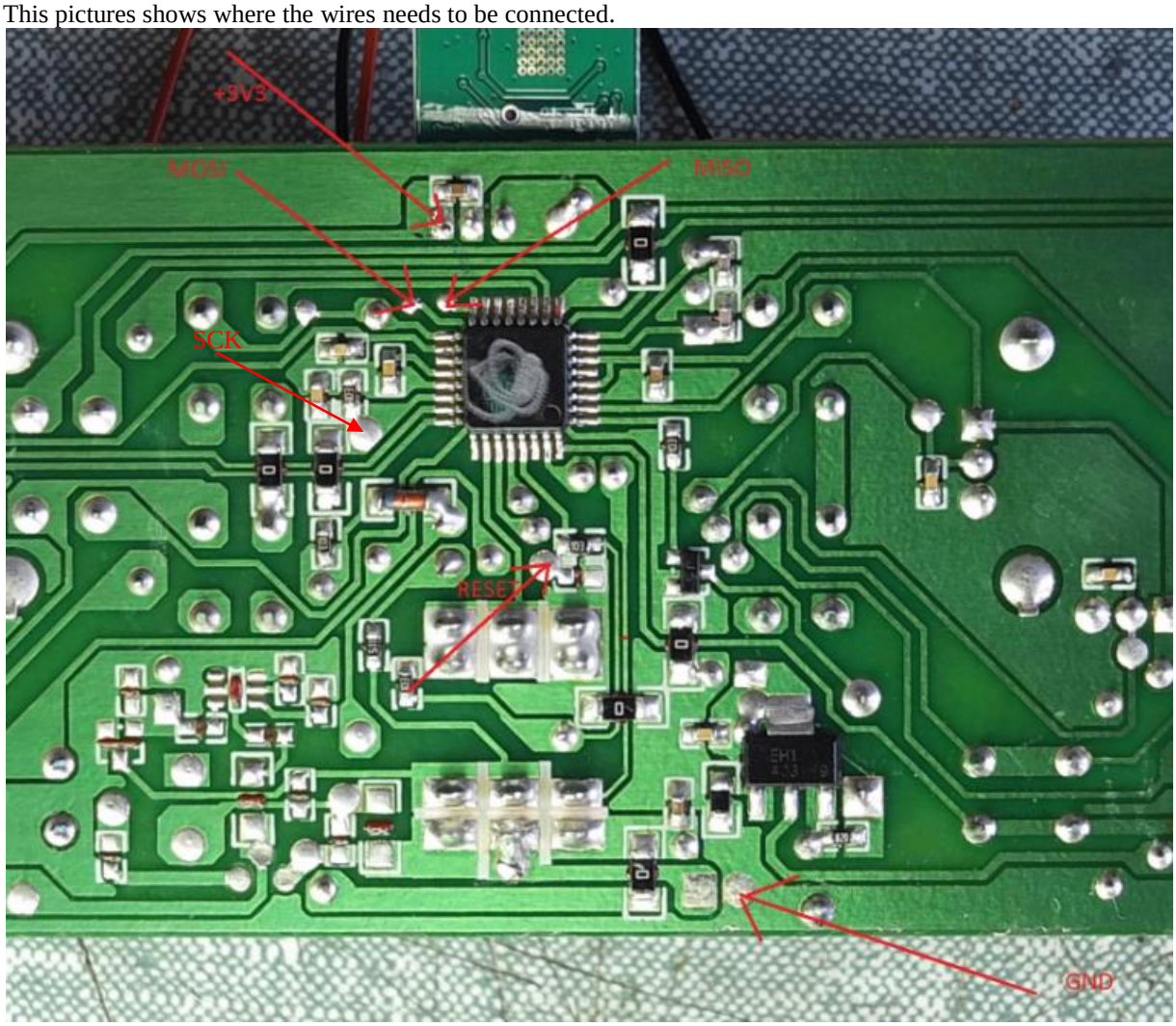
## 2.3 Preparation

To flash the Atmega you need to solder the programming wires to the PCB. And connect these wires to your programmer.
You need at least:
MOSI
MISO
SCK
GND

You can solder a plus wirer as well and power the circuit from your programmer.

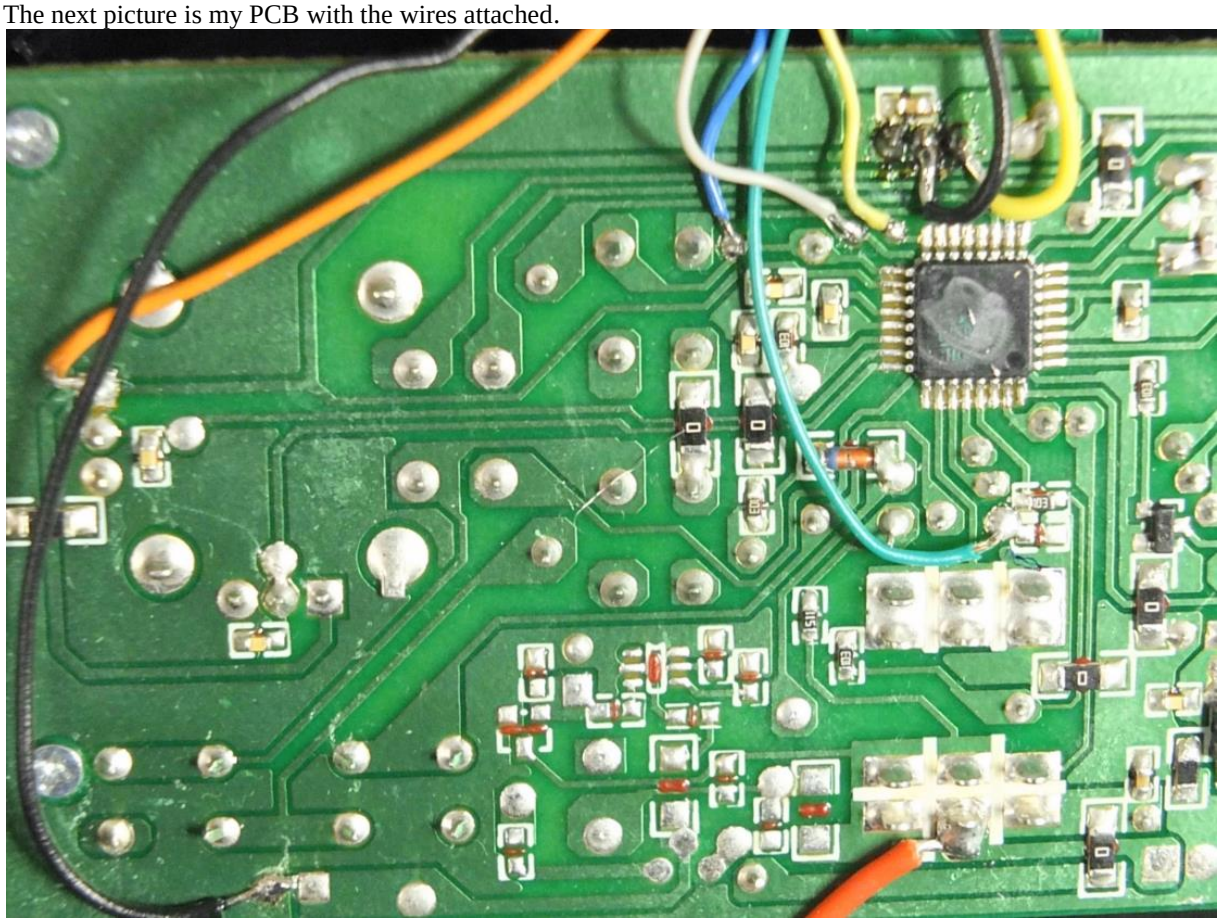
Otherwise you need to have batteries installed and have your transmitter powered on when programming.

This pictures shows where the wires needs to be connected.




Picture 3 Connection points for programming

The next picture is my PCB with the wires attached.



Picture 4 my PLC with wires on it.

### 2.4 Pre Test

After you have the wiring in place you are ready to test the connection to the atmel.

- Verify that your wiring is right.
- Connect your programmer to your PC.
- Start your programming Software
- Select your programmer
- Start „Auto Detect“ an ATMEGA 88P shall be reported.

## 3 Flashing the Firmware

If the Pre Test was successful you are ready to program the Flash memory.

- Select the HEX file.
- Click Program.

That’s it.

## 4 Q&A

Q: Is there a way to restore the original blade firmware?

A: No.

Q: I like to have swash plate mixing what do I need to do?

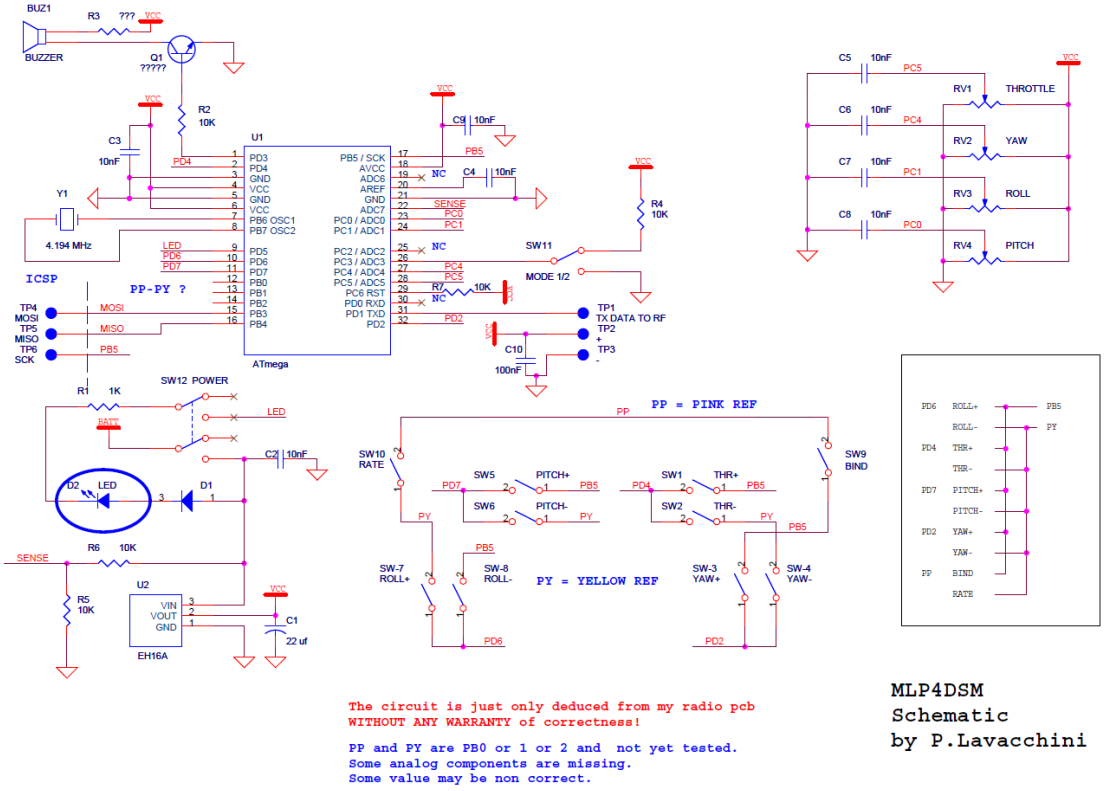
A: Sorry this is not supported, you need to change and recompile the source code.

# 5 Changing the source code

The flash memory is quite fill, smaller changes like a different crystal (`F_CPU in LP4DSM.h`), a different baud rate (`USART_BAUDRATE in SPECTRUM_TX.h`) or different default settings (`DEFAULT.h`) should not be a problem. For additional functionality you probably need to remove are optimize existing code.

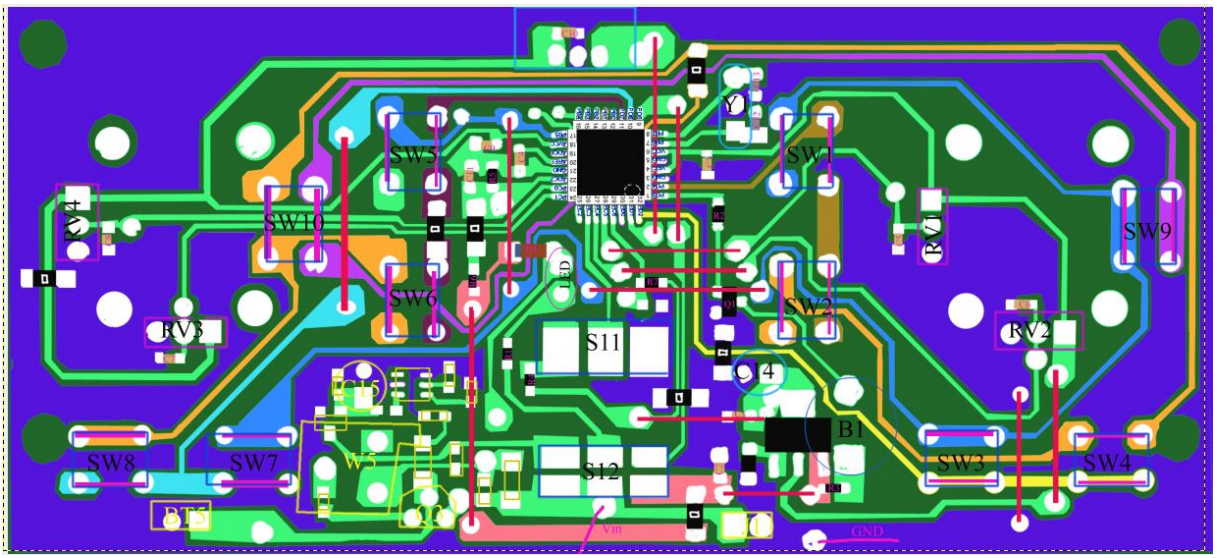
For compiling the source code you need Atmel Studio 7 or newer.

# Anlage A: The Schematics from internet

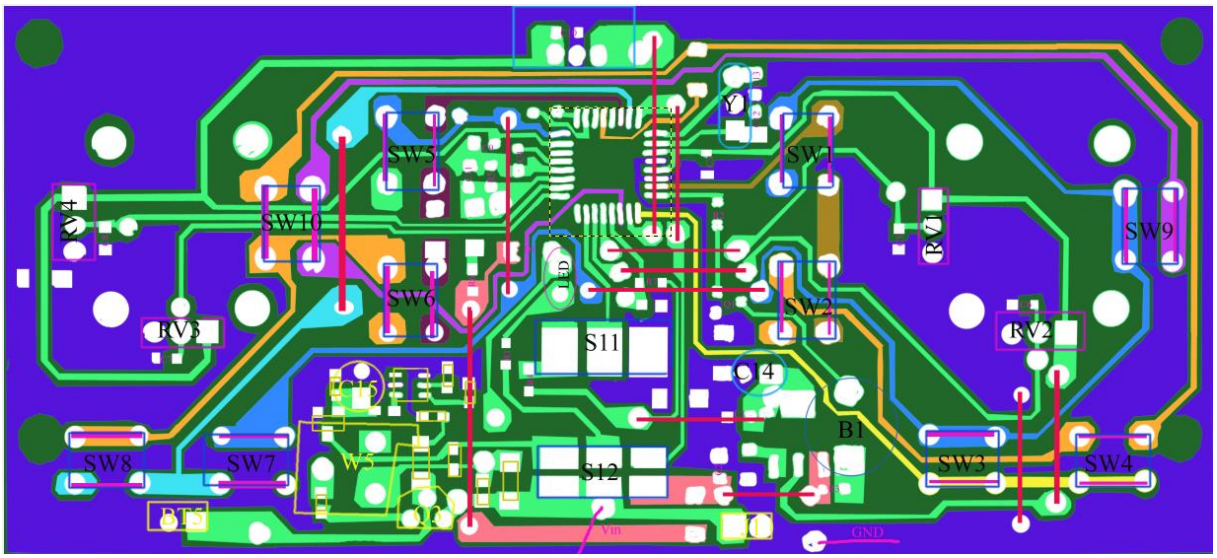



Picture 5 Schematics

# Anlage B: Scanned and reworked Layout




Picture 6 Layout with solder side components




Picture 7 Layout without solder side components