

# en/BL-Ctrl\_V1.2

20

LotharF  
MikroKopter.de

# Contents

<b><u>1 Differences to BL-Ctrl V1.1</u></b> .....	<b>1/11</b>
<b><u>2 Choosing the address</u></b> .....	<b>2/11</b>
<b><u>3 Software</u></b> .....	<b>5/11</b>
<b><u>4 Assignment of the terminal</u></b> .....	<b>6/11</b>
<b><u>5 Flying with 4S LiPo</u></b> .....	<b>7/11</b>
<b><u>6 Schematic</u></b> .....	<b>8/11</b>
<b><u>7 Troubleshooting</u></b> .....	<b>9/11</b>
<b><u>8 Controlling with own Hardware</u></b> .....	<b>10/11</b>
<u>8.1 AVR (plain)</u> .....	10/11
<u>8.2 Arduino</u> .....	10/11

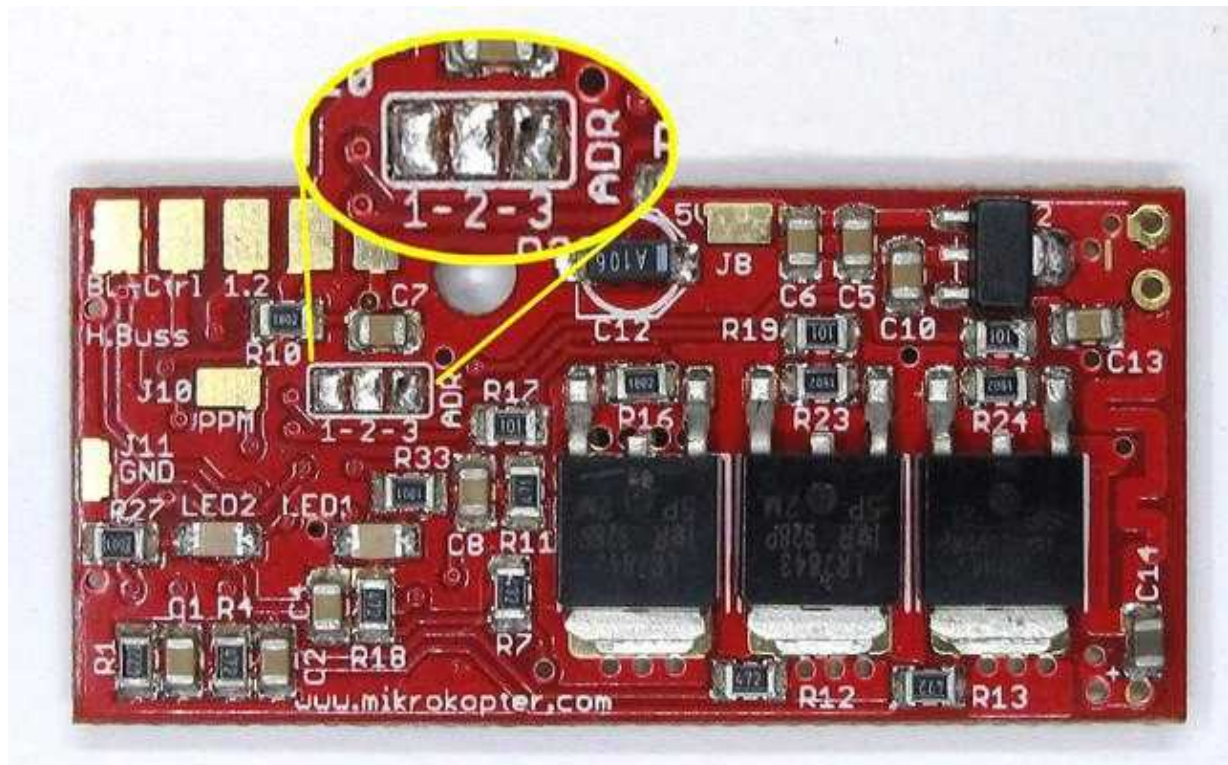
# 1 Differences to BL-Ctrl V1.1

- Visually no difference to BL-Ctrl 1.1
- PCB comes now with **4-layer**
- Better heat dissipation of the FET's and the Shunts
- Transistors T1, T2, T3 are replaced now by transistors with built-in resistors
- Less sensitive to moisture

See also the version history [Ctrl\\_History](#).

## 2 Choosing the address

On the V1.2 the motor address is set by a solder jumper on the board.



It is as follows:

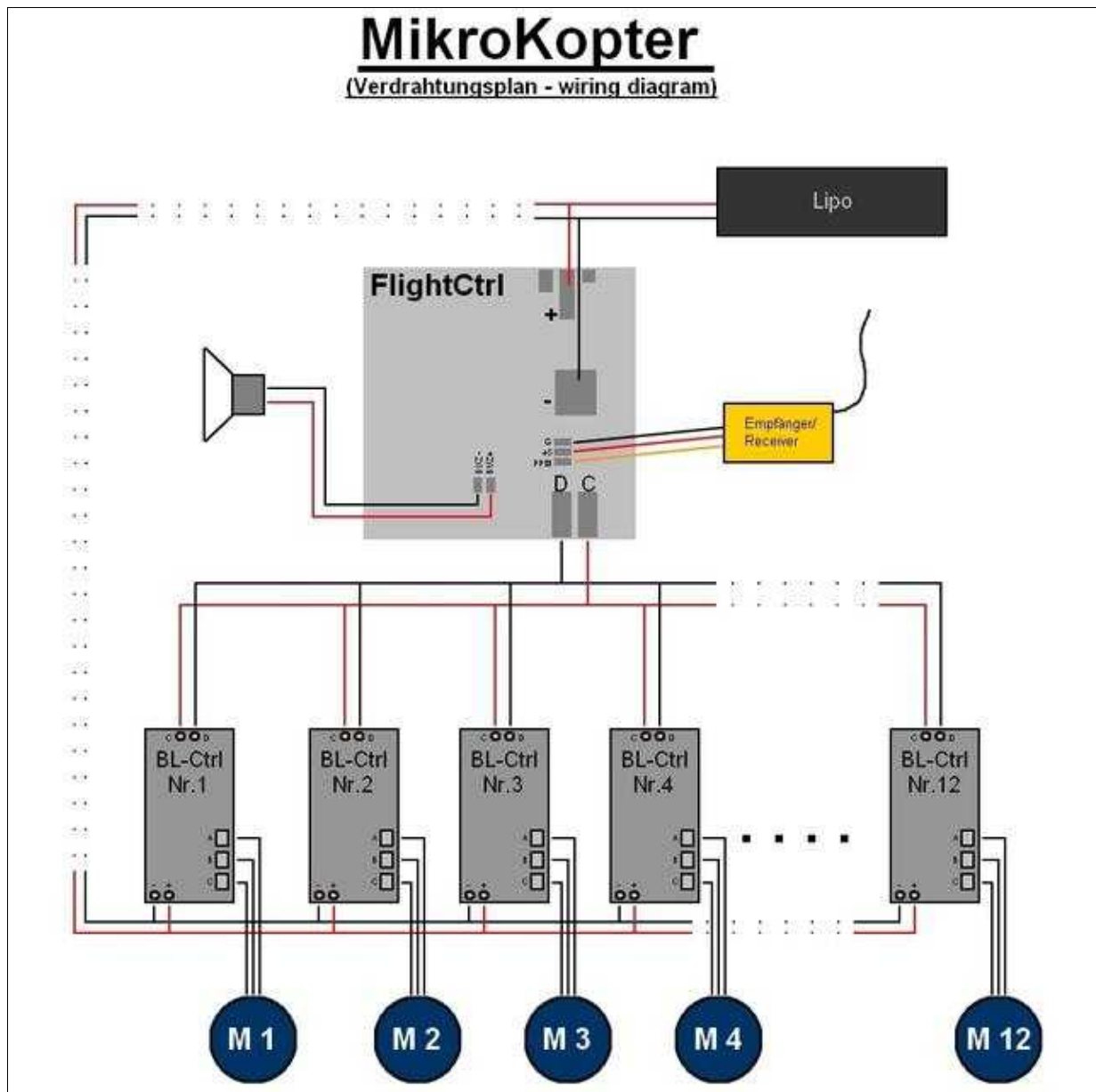
Adr.	1-2	2-3	Required Software:
1	open	open	BL-Ctrl V1.2 - Adr. 1-4
2	open	closed	
3	closed	open	
4	closed	closed	
5	open	open	BL-Ctrl V1.2 – Adr. 5-8
6	open	closed	
7	closed	open	
8	closed	closed	
9	open	open	BL-Ctrl V1.2 - Adr. 9-12
10	open	closed	
11	closed	open	
12	closed	closed	

Here you can see the addresses and positions of the motor controls and also the direction of rotation of the assigned propeller:

(Click image for high resolution)



The connection diagram:



## 3 Software

Depending how many motors you will use on your copter you need for the upper addresses of the BL-Ctrl's a different software. This can be downloaded here:

BL-Ctrl V1.2 - Adr. 1-4: [Download](#)

BL-Ctrl V1.2 - Adr. 5-8: [Download](#)


BL-Ctrl V1.2 - Adr. 9-12: [Download](#)

## 4 Assignment of the terminal

On the pre-assembled BL-Ctrl V1.2 you need to solder the enclosed Elko:

It is advisable to bend the capacitor by 90° so that it fit parallel to the outer side of the board.

Further you can see on the picture how to connect the I2C-Bus-Wires and the power supply.

 The controller is still sensitive to moisture and should be shrink wrapped with shrink tubing! For details see also [BrushlessCtrl](#) and water landing.

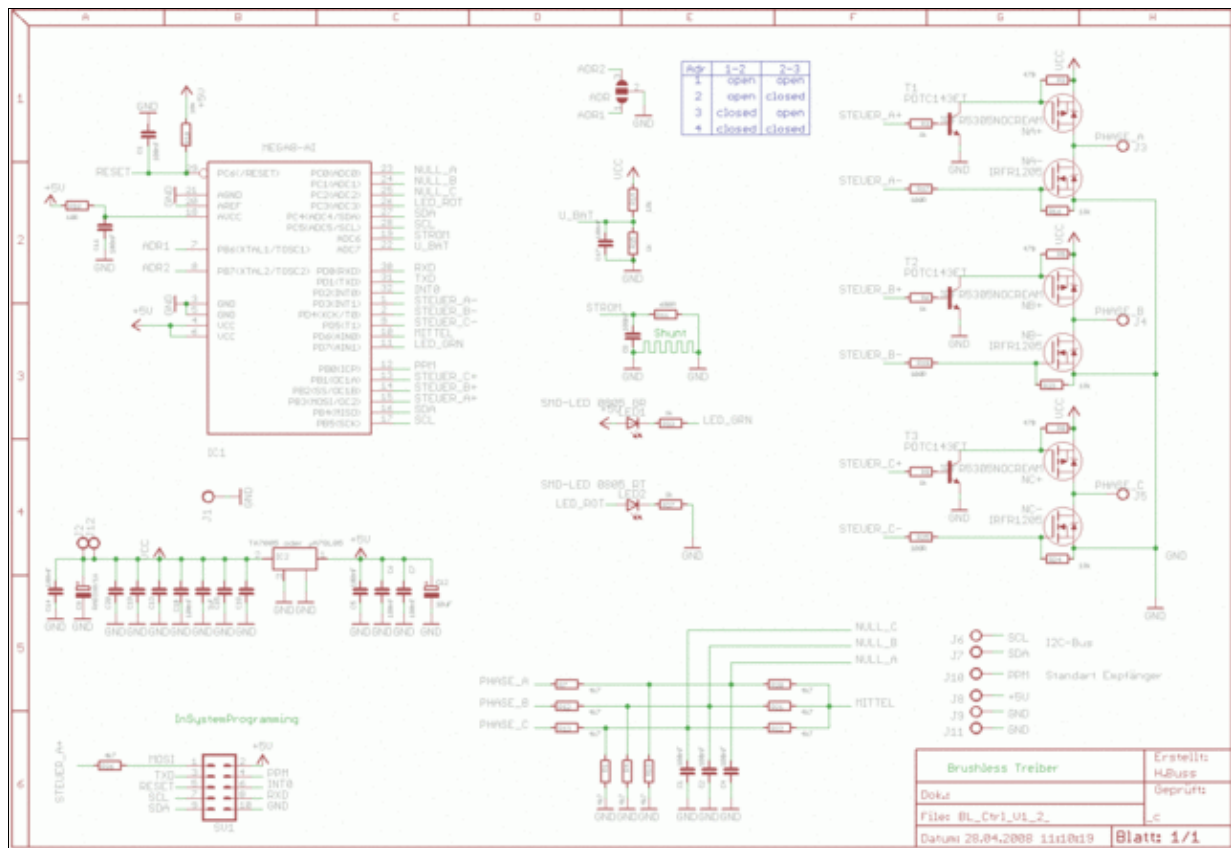


## 5 Flying with 4S LiPo

It is also possible and the same as with the BL-Ctrl V.1.1 to fly the MK with four-cell LiPo's. See also [4S-LiPos](#).

Here you need to adjust the wiring in that way that the BL-Ctrl are hooked up directly to the power supply and not over the switch of the Flight-Ctrl. That is necessary because the switch is not designed for those current and starting flashes coming with the high voltage.

# 6 Schematic



(click for high resolution)

Please note: In version 1.2 for T1-3 the part PTC143 with integrated base resistor is used. The base resistors R3, R6 and R9 need to be changed through a low-resistance resistor, i.e. 100 Ohm.

## 7 Troubleshooting

If the BL-Ctrl have a fault, you can see this e.g. on the flashing red LED on the BL-Ctrl.

How to find and fix a fault on this BL-Ctrl you can see here: [Troubleshooting](#)

## 8 Controlling with own Hardware

Controlling the BL-Ctrl with own hardware is easy.

### 8.1 AVR (plain)

A sample code for nearly any AVR is given below. The code will spin up the selected motor with a given speed. This is just to give an idea on how to interface them.

```
.. 1 #include <avr/io.h>
.. 2 #include <stdint.h>
.. 3 #include <util/delay.h>
.. 4 // using I2C Master library from
.. 5 // Peter Fleury ( http://jump.to/fleury )
.. 6 #include "i2cmaster.h"
.. 7
.. 8 #define TWI_BLCTRL_BASEADDR 0x52
.. 9
..10 int main(void) {
..11     i2c_init();
..12     uint8_t motor = 0; // 0 -> Motor1, 1 -> Motor2 etc...
..13     uint8_t speed = 10;
..14     uint8_t ret;
..15     while (1) {
..16         ret = i2c_start(TWI_BLCTRL_BASEADDR + (motor << 1) +
I2C_WRITE);
..17         if (ret) {
..18             // release bus
..19             i2c_stop();
..20             // failed... maybe print error or
something
..21         } else {
..22             i2c_write(speed);
..23             i2c_stop();
..24         }
..25         _delay_ms(50);
..26     }
..27 }
```

### 8.2 Arduino

A sample code for Arduino compatible devices is given below. The code will spin up the selected motor with a given speed. This is just to give an idea on how to interface them.

```
.. 1 #include <Wire.h>
.. 2
.. 3 void setup() {
.. 4     Wire.begin();
.. 5 }
.. 6
```

```
. . 7 #define TWI_BLCTRL_BASEADDR 0x52
. . 8
. . 9 void loop() {
. 10
. 11 . . . . int motor = 0; // 0 -> Motor1, 1 -> Motor2 etc...
. 12 . . . . int speed = 10;
. 13
. 14 . . . . // The Wire library uses 7 bit addresses throughout. If you
have a
. 15 . . . . // datasheet or sample code that uses 8 bit address, you'll
want to
. 16 . . . . // drop the low bit (i.e. shift the value one bit to the
right),
. 17 . . . . // yielding an address between 0 and 127.
. 18 . . . . Wire.beginTransmission((TWI_BLCTRL_BASEADDR + (motor << 1)) >>
1);
. 19 . . . . Wire.write(speed);
. 20 . . . . Wire.endTransmission();
. 21
. 22 . . . . delay(50);
. 23 }

```

- 
- KategorieMK-Baugruppe/de