Motor control with H bridges

Gunther Zielosko

1. Introduction

Controlling rather small DC motors using micro controllers as e.g. BASIC-Tiger® are one of the more common applications of those useful helpers. Even in our series about BASIC-Tiger® applications they play an important role. Just remember, for example, our experiments with DC motors (application note No. 019), in which we switched motors on and off, changed their direction of rotation and even set their revs per minute via pulse width modulation. So why another version? The explanation is simple: Former solutions simply kept the processor too busy and controlling several motors (e.g. for robot projects) would be hard to implement. You will learn about a clever solution in the following application note. Here up to 62 motors can be controlled independently, at a moderate CPU overhead and a minimal I/O pin demand. Motors can be run forward and reverse at 127 speed levels.

2. Some basics of motor control using H bridges

Of course we know that motors are usually turned on and off, its direction of rotation are changed by reversing polarities and their revs per minute are set via the on/off-forward/reverse-ratio via switches. Here switches can be mechanical contacts, relays or transistors. Today integrated circuits which contain complete logics and all elements for controlling motors are the state of the art. At the motor end of these ICs we will always find a so-called H bridge, which can implement the above mentioned controlling functions. It is called H bridge because of its shape which looks like the letter H. Usually an H bridge consists of power MOSFETs or bipolar transistors with different conductivities (PNP and NPN) which are often designed as Darlington transistors. The switchable currents are different from type to type, as well as the outer dimensions and features (heat sink, thermic deactivation, integrated free wheeling diodes etc.). If you neglect the differences such an H bridge looks as depicted schematically in figure 1. The red line shows the current flow for one direction of rotation, the blue line for the other direction, accordingly the logical signals "0" for a locked and "1" for a switched-through transistor. In the depicted case both are NPN transistors. Vcc2 operating voltage is the voltage for the motor, which is usually independent from logics and can be set higher than e.g. 5 V. Our circuit diagram does not consider that top and bottom transistors must be controlled at a different level – of course a really useful motor controlling IC does that internally.

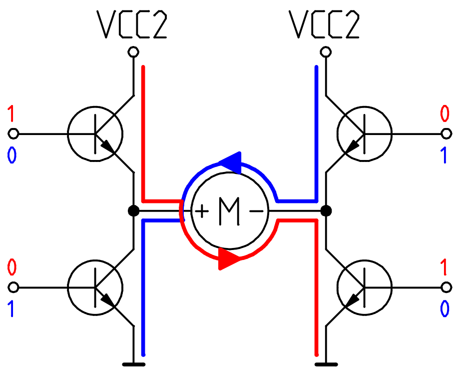

Fig. 1 Diagram of an H bridge

As you can see, current can flow through the motor in both directions. Things get dangerous when both transistors of a vertical branch are switched through at he same time. This has to be avoided by logics. Usually inputs are switched the right way by force either by the application or internally using inverters. There is a wide variety of different H bridge ICs, this is just a small choice:

L293, L298, LMD182XX, ZHB67XX, L620X, BA6XXX, TLE5XXX, LB1836M, SN754410

As an example for the functionality of such elements we chose SN754410 for further information.

3. Texas Instruments SN754410

The SN754410 is a rather old-fashioned element, but it is suitable for our purposes – it has a traditional 16 pole plastic DIP case (figure 2).

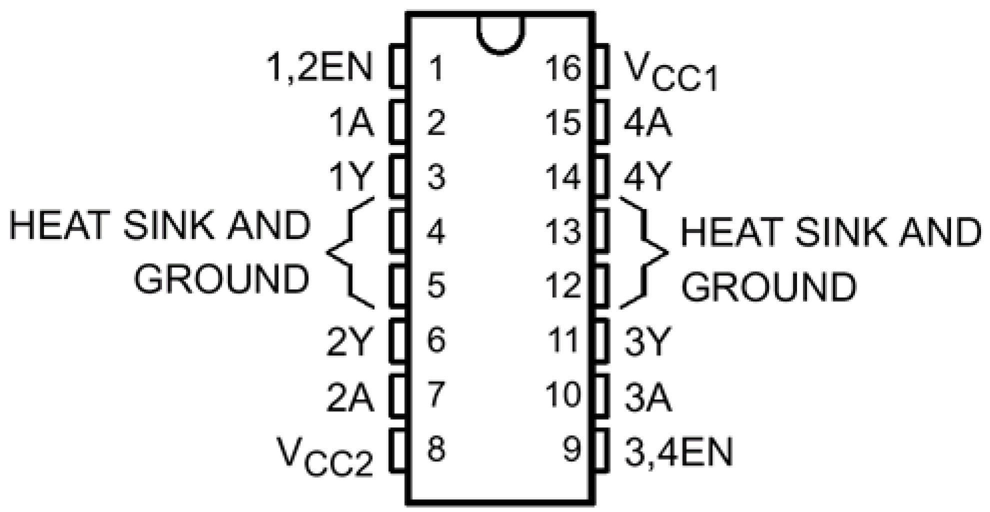

Fig. 2 Module SN754410 can control two DC motors independently

Its datasheet is available on

http://www-s.ti.com/sc/psheets/slrs007b/slrs007b.pdf

Connections designated as “A” are inputs, connections designated as “Y” the corresponding outputs. For a DC motor you require two inputs respectively two outputs. Inputs denominated as “EN” can switch the corresponding outputs to high-impedance and with it the corresponding motor off. On the input side the SN754410 is supplied with common logic levels (TTL or CMOS). Vcc1 supplies internal logics and can be between 4.5 V and 5.5 V. Vcc2 only supplies the output stages and, of course, the motors. Voltage can be chosen in between 4.5 V and 36 V. Each output can provide up to 1 A current for the motor.

Looking at the truth table (Table 1) you notice that the output of each driver (H = possibly high voltage, which means no TTL signal!) directly follows the input signal (logic H/L, i.e. TTL or CMOS!), provided that the corresponding ENABLE input is high (H). If it is low (L), the output will be high-impedance. The EN input always controls two drivers.

Inputs		**Output**
1A	**1,2EN**	**1Y**
H	H	H
L	H	L
X	L	Z

Tab. 1 SN754410 truth table (driver 1)

No problem so far - but what about the inputs? The subtask “power driver” including thermo protection, free wheeling diodes etc. is solved by the SN754410, but controlling several motors by PWM still is not as the SN754410 so can only be used as an on/off-switch and as a polarity reversing switch. A direct PWM control by BASIC-Tiger® is possible, but the BASIC-Tiger® would, however, almost only be busy with this problem, which is not our objective. We will show you a small but fine solution: A complete motor controller with a serial control for two motors.

4. Pololu Dual Serial Motor Controller

US company Pololu provides small modules for robotics and similar purposes, which independently control two DC motors each via a serial interface.

http://www.pololu.com/products/pololu/0101/ (Dual Serial Motor Controller)
http://www.pololu.com/products/pololu/0401/ (Micro Serial Motor Controller)

The ‘heart’ of these modules is a small PIC controller which controls a SN754410 (or similar IC, the LB1836M) on its part. Several of these modules (programmed accordingly by the producer) are controllable by one serial interface – the cascade can control up to 62 motors.

While the first module (figure 3) is available as a construction kit for 20 $, you can purchase the second version as a finished "micro" version (figure 4) for 23 $. Both modules are used for choosing the motor, controlling revs per minute and changing the direction of rotation by software completely independently. The parameters set remain conserved until RESET or another control command is entered, i.e. the motor keeps rotating in the same setting until a new command arrives. With this idea controlling projects with several motors via BASIC-Tiger® becomes a walkover.

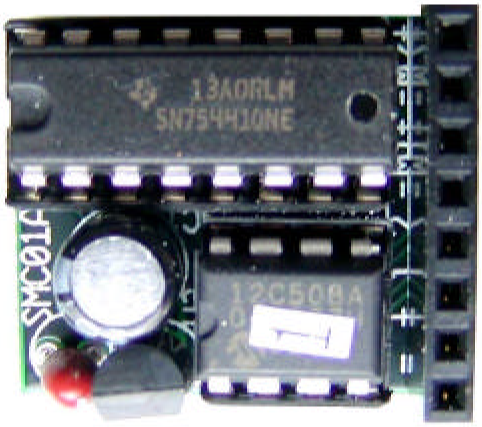

Fig. 3 Construction kit Pololu Dual Serial Motor Controller with SN754410 (here programmed as motor controller #1)

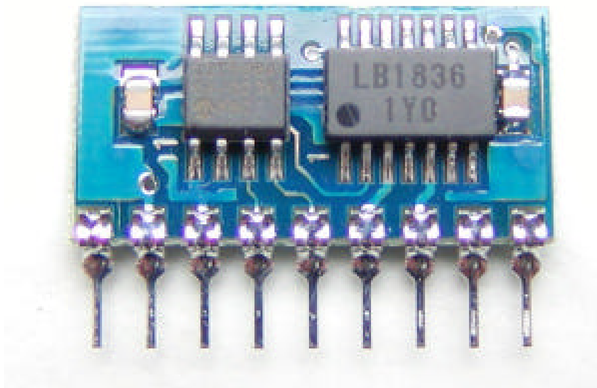

Fig. 4 Finished miniature version, the Pololu Micro Dual Serial Motor Controller with LB1836

Let's take a closer look at the hardware and software of these motor controlling modules.

5. Connecting BASIC-Tiger® and the motor controller

Both modules are connected similarly, although one has 8 and the other one has 9 connections. This is because the bigger module has an own 5 V regulator, which supplies both ICs directly with Vcc1 from the (high) motor voltage, while the smaller module has an own pin for the motor and logic supply. This is advantageous, if the motor voltage is supposed to be between 1.8 V and 9 V independent from the logic voltage and the separate 5 V voltage from the BASIC-Tiger® system should be used. In the case of the big module a supply voltage has to be over 5.6 V (in order to enable the regulators to produce 5 V), but can go up to 25 V which is the motor voltage at the same time. The remaining part is simple. Both motors are connected to the accordingly depicted module pins, the logics input "reset" is connected to a Tiger-I/O pin and the input "serial control input" is connected to a serial BASIC-Tiger® output (TxD0 or better TxD1). Since a bidirectional operation is not necessary, further motor controllers can be connected to this cable.

Here you must consider that both Pololu Motor Controller versions expect TTL levels at the serial input. This is no problem for many Tiger modules. If you absolutely want to use a BASIC-Tiger® with integrated RS232 level converters or the Plug and Play Lab, a further external RS232 level converter or something similar is required (e.g. the one from the Motor Controller datasheet)!

The finished circuitry for bigger modules looks as follows (figure 5).

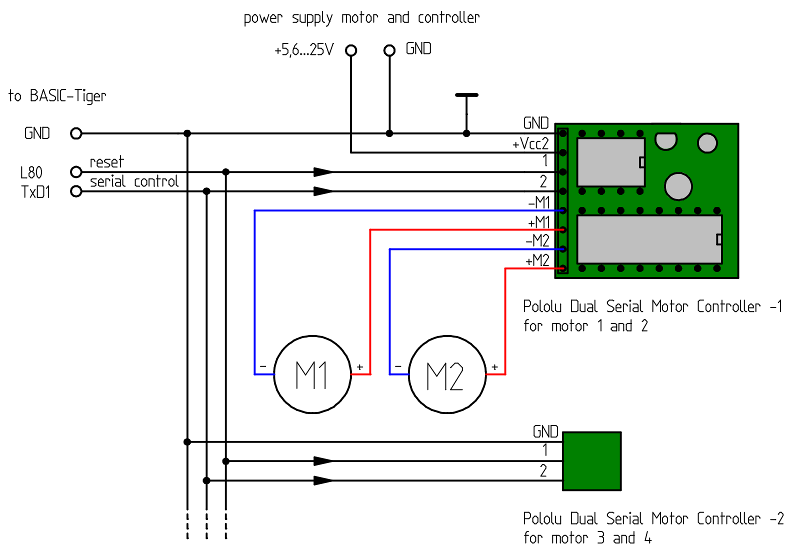

Fig. 5 Connection of the original Pololu Dual Serial Motor Controller with BASIC-Tiger®. Power supply of further modules as well as the connection of the motors is not depicted!

If you wish to use the BASIC-Tiger® system's Vcc for the motor controller (construction kit version) and/or motors with either low or very high nominal voltages are used, we can abstain from its regulator and its primarily sided electrolytic capacitor. Like this we cancel the connection of both current supplies. At the output of the (now not integrated) regulator we apply Vcc (5 V) from the BASIC-Tiger® system. Now the input voltage Vcc2 is only motor voltage, which is allowed to be between 4.5 V and 36 V. Figure 6 shows a schematic of this version and figure 7 shows a view of both module versions. The author simply soldered an additional socket into the "original hole" of the provided voltage regulator so that we now have to apply two voltages (Vcc1 to the additional socket and Vcc2 to the former combination socket) again.

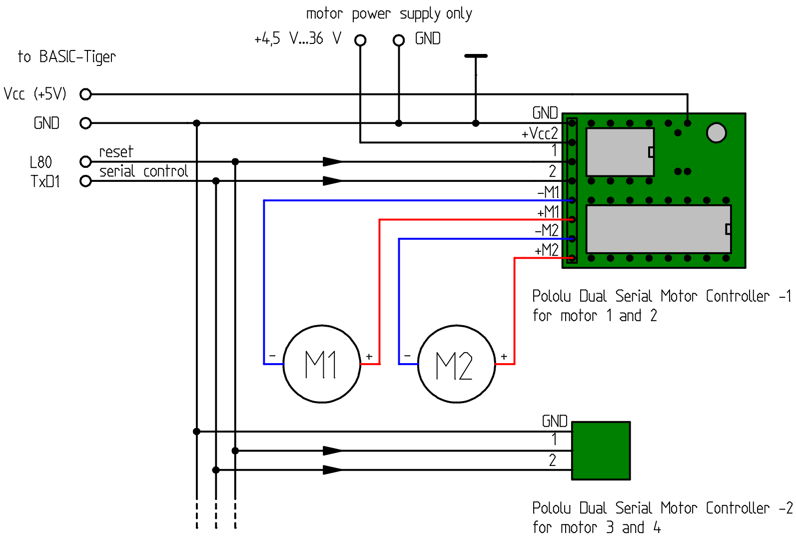

Fig. 6 *the independent power supply for the motors and the additional supply of the controllers from the Vcc of BASIC-Tiger®. The regulator and its primarily sided electrolytic capacitor are simply left out.*

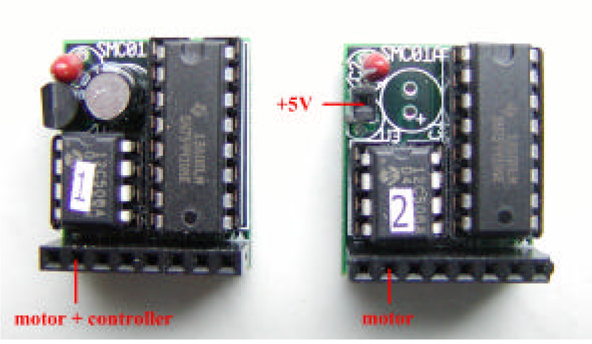

Fig. 7 *Both construction versions, on the left the original with combined voltage supply, on the right with separated voltage supply for motor and controller*

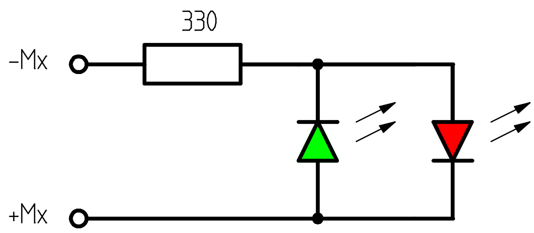

Fig. 8 *Indicator circuit, which displays both the set direction of rotation and revs per minute*

Besides, motor controllers can be used for controlling totally different things despite their designation, e.g. LEDs or lamps. Like this you can easily dim sources of light in models or other applications. A circuit according to figure 8 with two anti-parallel LEDs of different colour can be used for measured values (negative or positive).

For those of you who would like to learn more about the mysteries of the Pololu Dual Serial Motor Controller or would like to make further modifications, we depicted the internal circuitry in figure 9.

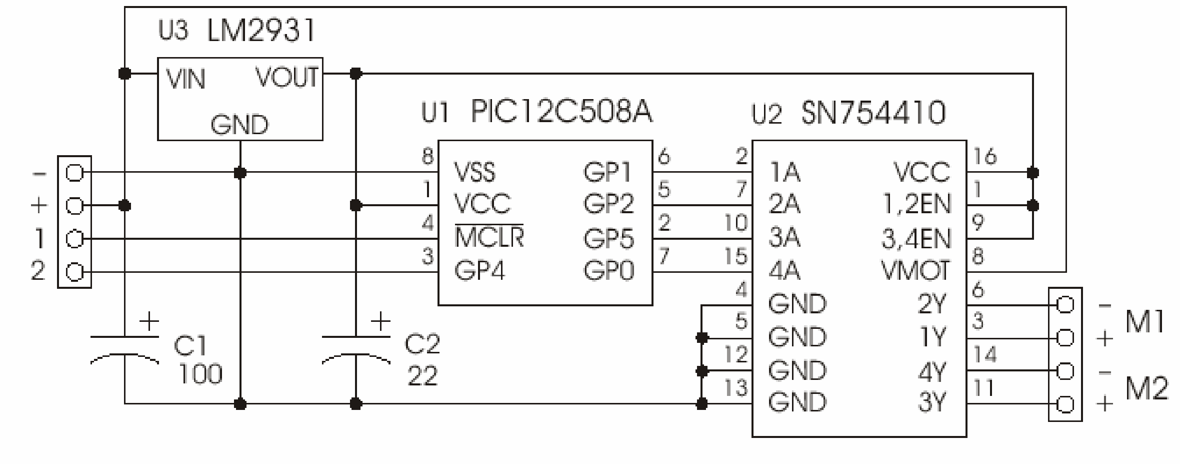

Fig. 9 Internal circuitry of Pololu Dual Serial Motor Controller with SN754410 (taken from Pololu fact-sheet and manual)

6. Here we go – the motor controlling software

Serial control takes place asynchronously according to the RS232 protocol (8 bits, no parity, and 1 stop bit with 1200 to 19,200 baud). The module automatically detects the baud rate. But how are the data sent from BASIC-Tiger® to the module? We remember that this only functions in one direction!

Basically all commands directed at the motor controller consist of 4 bytes with the following structure:

start byte	device type	motor # and direction	motor speed
80h	00h	0 x x x x x x x	0 x x x x x x x
		motor # (bits 6–1)	speed (bits 6–0)
		direction (bit 0)	

Tab. 2 structure of control command directed at the motor controller

Only the first start byte "80h" (or decimal 128) has a "1" at the highest bit position. The second byte, designated as "device type", has no importance for us in the first instance; it can only distinguish the motor controller from other devices. More important is the third byte which always has a "0" at its highest bit, the motor number at the following 6 bits and the direction of rotation in its lowest bit. Finally the fourth byte contains speed information at its seven lowest bits, 00h ... 7Fh, 00h meaning motor standstill. Bit 7 of byte four is always "0", too.

Whereas creating the parameters "direction" (logical 0 or 1) and "speed" (00h…7Fh or 0…127 decimal) is still comparatively easy, creating "motor number" takes some more effort.

Controllers have to be programmed by the producer, as mentioned above. You will find the following information in the order documents of Pololu:

Dual serial motor controller-1
Responds to motor numbers 0, 1, 2, and 3.

Dual serial motor controller-2
Responds to motor numbers 0, 1, 4, and 5.

Dual serial motor controller-3
Responds to motor numbers 0, 1, 6, and 7.

This is a bit long-winded, but simply means the following:

- If you need only one controller (for one or two motors), you can always choose motor numbers 0 or 1 for all controller versions.

- If several controllers are connected to a serial line (simply parallel!), the second pair of numbers becomes important. Then you can control motor numbers 2 and 3 with the first type *(Dual serial motor controller-1)*, motor numbers 4 and 5 with the second type (*Dual serial motor controller-2*) etc. Motor numbers 0 and 1 are not used in this case. If you need more than 6 motors, a special order from Pololu will be necessary, since higher numbers are not kept in stock. Order numbers result from the continuation of the list above. For programming keep in mind to always add 1 to the motor number, so the first motor gets number 2, the second 3, the third 4 etc.

In the underlying case up to 6 motors are simultaneously used with BASIC-Tiger®, which means we will need three controllers, to be exact *Dual serial motor controller-1, -2 and -3*.

Two examples:
1) We have 6 motors in total, so we use the extended motor number system, and we want to run the fourth motor forward at maximum revs per minute. Remember that the first controller administers motor numbers 2 and 3, the second controller numbers 4 and 5 etc. So the fourth motor has number 5, i.e. the command word looks as follows:

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1
80h								**00h**								**0Bh**								**7Fh**							

2) Again 6 motors in total, the sixth motor (number 7) is supposed to run slowly backwards:

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1
80h								**00h**								**0Eh**								**01h**							

Mentioning the “RESET” pin. During normal operation it should be connected to logic “1” or to logic Vcc1. If you set the “RESET” pin to “0” for at least 2 µs, all motors of the motor controller are switched off and the controller waits for the first serial command from BASIC-Tiger®. You do not have to use "RESET", but in this case you should connect the input directly to logical Vcc1. To create orderly starting conditions, however, "RESET" is appropriate at every start of the program.

The MULTIMOT.TIG program successively controls 6 motors in total, from standstill accelerating in one direction until maximum revs per minute and back. After a short break the procedure is repeated to the opposite direction. Then the second motor is started etc. This requires three controllers (numbers 1, 2, 3!) and their six motors to be connected to the serial line TxD1 (TTL level!). The BASIC-Tigers® display, if connected, shows the state of every single motor.

7. New options

The simple, but comfortable control of several motors via one single serial connection and their operation in only one direction offers us further options. You can easily transmit these serial commands via radio transmission or IrDA, i.e. wireless. Every model-maker will be glad about that. Although modern remote control systems are quite powerful, controlling 62 motors (or other functions such as lamps, LED etc.) remains a dream. Our solution is able to do much more, motors can be controlled locally, i.e. a BASIC-Tiger® (e.g. as a remote control) controls the transmitter and one receiver each controls a group of motors. This is perfect for home systems (blinds) or in the DIY area (vehicles, robots).
Not only the simple setting of direction of rotation and speed is interesting, using the combination of BASIC-Tiger® / motor controller model vehicles can accelerate and slow down like the originals. Applications which make a motor start slowly at the touch of a button and accelerate in a predefined way until maximum revs per minute are reached are also tempting. Relinquishing the button the procedure starts again from the beginning, e.g. in the other direction. Like this you can elegantly execute adjustments which require delicate movements (at the beginning) as well as fast alterations over a whole range (after pressing the button for a couple of seconds). Use the encoder or other feedback sensors, which signal position or speed of your drive! Since BASIC-Tiger® is hardly busy with motor controlling, there is plenty of processing and I/O capacities for further procedures. Figure 10 shows the author's experimental construction with 3 motor controllers for 6 motors in total.

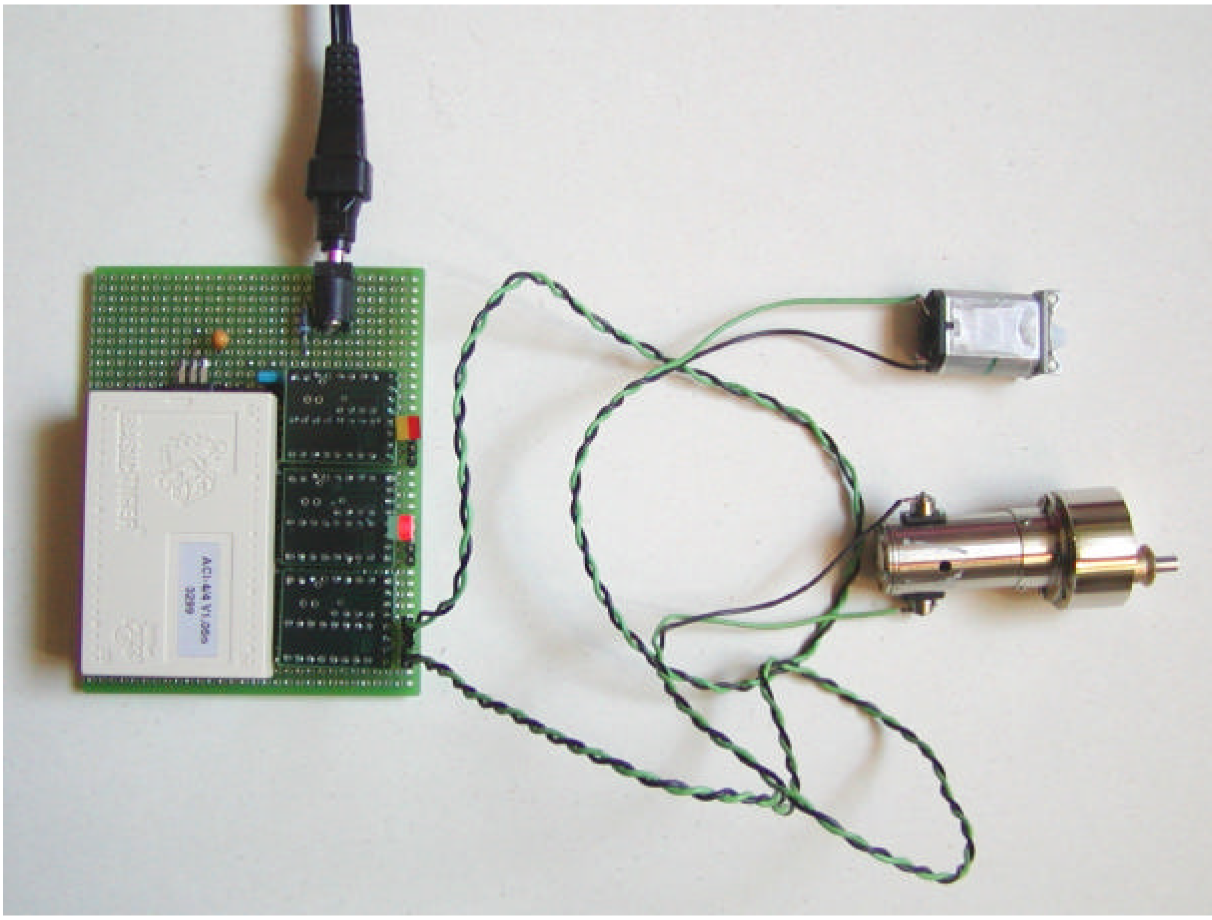

Fig. 10 Experimental construction with 3 motor controllers. Two different motors and two LED combinations are connected according to figure 9

Please note:
For projects with more than one motor the usage of an own power supply unit makes always sense.

Have fun with this really universal motor control!