Electronics

There are countless electrical and electronic options to use in a combat robot. This subject by itself could result in an entire book. Because of that, in this chapter we try to summarize and limit the discussion to the most used components in combots, with effectiveness verified in practice, in the arena.

Combat robot operation demands a great number of electronic components, among them: radio transmitter, receiver, RC interface, speed controllers, relays/solenoids and on-off power switches, connected by plugs, terminals and wires. These components are described next.

Radio Transmitter and Receiver

Transmitters

A radio transmitter allows the driver to send commands to a receiver inside the combat robot. There are several radio manufacturers, such as Futaba, Airtronics, JR, Hitec, GWS and Spektrum. There are also other cheaper solutions, such as radios adapted from toys, wireless gamepads, and transceiver circuits, however you must guarantee that these low-cost systems will have enough power to avoid signal loss when the robot is inside the arena, as well as implement failsafe features in all channels, as explained later.

Radio systems are named for their number of channels, which is the number of outputs that a transmitter-receiver set has. For instance, a four-channel set can control four different devices. Most combat robots only use three channels: two for the drive system (forward/backward and left/right) and one to control the weapon, if any. Three and four-channel radios are cheaper and, in general, enough for a combot. However, radios with 6, 7 or more channels usually have more functions, being programmable and including internal memory, such as the Futaba 7CAP 75MHz radio pictured to the right.

Most radios use the 72MHz frequency band, which is reserved for air models only, which makes them prohibited in almost all combot competitions. The usual ground band is 75MHz, but several others are also used: 27MHz, 50MHz, 433MHz (only in Europe, Africa and Middle East), 900MHz (only in the Americas), 2.4GHz and 5.8GHz. All these frequencies are part of the Industrial, Scientific and Medical (ISM) radio bands, originally reserved by an international treaty for the use of radiofrequency in the cited fields other than communications.

The 27 and 50MHz bands are normally employed in radio-controlled toys and old cordless phones, 433MHz is found in European transceiver pairs, while 900MHz and 2.4GHz are a commonplace in modern life, being present in wireless LANs, Bluetooth, cordless phones, and even microwave ovens. The 5.8GHz band is not very common, but it is also present in wireless network systems and phones.

Each band is divided into channels. There are only 30 channels for the 75MHz band, from channel 61 (75.410MHz) to 90 (75.990MHz). The 72MHz band, which is not allowed in combot competitions, has 50 channels, from 11 (72.010MHz) to 60 (72.990MHz). And the 27MHz band has only 6 channels, while the 50MHz has 10.

In a competition, it is forbidden to have two radios using the same channel at the same time. This is a safety measure, since the radio from one team could accidentally activate the robot from another. With this in mind, in the events it is mandatory to only turn your radio on if it has an appropriate frequency clip.

However, if a distracted builder forgets to return a clip after using it, the event might suffer delays. As a result, a few events require the use of single bind systems, which only allow a receiver to follow commands from a single radio, normally using the 900MHz or 2.4GHz frequencies.

Those systems rely on Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS), which use a unique identifier code for the receiver and transmitter. They dynamically change the paired transmitter and receiver channels upon signal quality loss, caused normally by interferences (noise). This bi-directional system is only possible due to the larger bandwidth and higher frequency they use. A few 900MHz and 2.4GHz systems can use the available bandwidth to perform robot telemetry as well, such as in the IFI FRC or Spektrum Telemetry packs.

In 72 and 75MHz systems, channels are defined by a pair of crystals (pictured to the right), usually sold together. The Tx crystal must be installed into the transmitter, while its pair Rx is placed into the receiver. Always have a spare crystal pair from a different channel, in case you have to face an opponent that uses the same channel. Only buy crystals for the same band as your radio - 72MHz crystals do not work on 75MHz radios and vice-versa. It is possible to convert a radio from 72MHz to 75MHz, a procedure that costs between US$20 and US$50 if done by professionals.

Besides frequency bands and channels, there is also modulation, which is the way that information is encoded into the radio wave. The most commonly used modulations among the countless existing standards are AM, FM, PCM and DSM, described next.

AM stands for Amplitude Modulation, in which information is transmitted varying the radio signal amplitude.

FM means Frequency Modulation, it is a standard less prone to noise than AM. Control information is transmitted adding a variable frequency wave into a carrier wave. The receiver then extracts the information from the carrier and sends the appropriate commands to the devices attached to it.

The third modulation type is PCM, which means Pulse Code Modulation. Technically, this transmission is also FM, the difference is that in PCM the information is digitally transmitted. Instead of an analog transmission, signals are send in digital form, coded, becoming a lot more reliable. PCM provides an even greater noise immunity.

DSM, which stands for Digital Spectrum Modulation, is Spektrum's proprietary modulation for 2.4GHz systems. It divides the 2.4GHz band into 80 channels (slots), using some DSSS and FHSS features, with a unique identifier that only allows communication between a single transmitter and its bound receiver. In the unlikely event of all channels becoming occupied, the link between the Tx and the Rx won't happen. This standard has been updated to DSM2; unfortunately, older radios such as Spektrum DX6 only accept DSM receivers, although DSM2 radios work with DSM receivers.

Besides Spektrum and JR (which also uses DSM), other radio brands have their own 2.4GHz modulation, such as Futaba's FASST, Airtronics' FHSS-2, and Hitec's AFHSS. Their differences are minimal, so choose them keeping in mind your budget and favorite brand.

Receivers

A receiver is the component responsible to demodulate the radio-transmitted signals and direct the commands to servos and other electronic circuits. A typical receiver is pictured to the right, a Futaba 75MHz They come in several sizes and weights.

For insect robots, such as fairyweights (150g), antweights (1lb, equivalent to 454g) or beetleweights (3lb, equivalent to 1361g), it is a good choice to use the GWS micro receiver (left picture to the right) or the Nano receiver (right picture). They are really small and weigh between 2 and 8 grams without the crystal. Be careful, they do not work with regular crystals, they need special ones.

It is extremely important that the Rx crystal (if needed) is well attached to the receiver. It is usually a good idea to use an adhesive tape around the receiver to prevent the crystal from becoming loose. Many fights have been lost because of a knocked off crystal due to the vibrations from a major hit. Tape as well the connectors, to avoid the cables from becoming loose. It is also a good idea to use some adhesive tape to cover the receiver inputs that are not used, to prevent metal debris from causing a short-circuit during a round. Also, it is fundamental to shock-mount the receiver, using for instance foam, EVA or rubber.

All modern receivers pick up the modulated signal and then decode it, resulting in a Pulsed Position Modulation (PPM) signal that contains the information about all output channels. This information is framed in an envelope of 20ms (or some other fixed value between 18 and 25ms), consisting of a train pulse with several 5V pulses, one for each output. These 5V pulses have periods that vary between 1ms and 2ms, depending on the command sent by the driver. If a radio stick is completely to the left or down, then the period of the associated pulse from this channel is usually 1.0ms. If the stick is centered in neutral position, then it is 1.5ms. And if it is to the right or up, then the pulse lasts 2ms. For switches, as those that are usually placed on the top portion of a radio transmitter, 1ms would be associated to the off position and 2ms to the on position. Most high-end systems allow these configurations to be changed on the transmitter unit. For instance, a channel can be reversed, or it can be mixed with others. PPM is further explained in section 7.8.2.

Among the radio features, *failsafe* is one that is required by all combot events. A few vendors call it *failproof* or *smartsafe*. It consists of a subsystem that allows to pre-program each receiver channel output in case of a signal loss. Program the failsafe to, in the event of signal loss, send a 1.5ms pulse signal to the drive system channels (associated with a centered radio stick, which would stop the robot's translational movement), as well as send a 1ms pulse signal to a channel from a solenoid or relay from the weapon system (if the 1ms pulse is associated with the off condition). This is an obligatory feature, checked in the safety inspection from any combot event. For hobbyweights (12lb) or heavier robots, all used channels must have a failsafe. Usually, robots in insect classes only need to have failsafe on the weapon channel.

AM, FM and a few other radio systems do not have built-in failsafe. Failsafe must then be implemented between the receiver and the commanded devices, which can be accomplished using an appropriate RC (radio-control) interface board, or through a dedicated module such as the Micro Failsafe Dynamite (pictured to the right), which can be bought in R/C hobby stores.

Listed below is a summary of the main features from radio-receiver systems:

- number of output channels: in general from 4 to 9 in air radios, or 3 in pistol-style ground radios; for a robot with an active weapon, at least three channels are needed;
- reversion: ability to program channel output inversion;
- ATV / EPA: adjustable maximum and minimum values that an output can have;
- dual rate / exponential: output sensibility and linearity adjustment;
- mixing: ability to mix channels (very useful), which sometimes can be programmable;
- multiple models: allows the storage of several distinct programs, one for each robot;
- failsafe: allows to program the channel outputs in case of a signal loss; only digital radios have this feature, but a few do not have it in all channels; analog radios must use an external module;
- frequency channel reassignment: ability to automatically switch the transmitter and receiver.

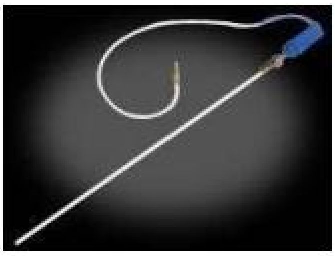
Antennas

A huge problem that strikes combat robots is signal loss. Combots are made out of metal, therefore antennas placed inside them may suffer from the Faraday's Cage effect, where the robot chassis blocks the signal or at least considerably reduces its strength.

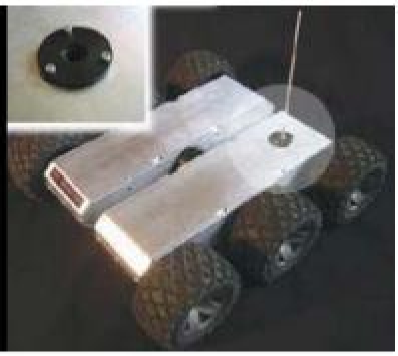
Robots with polycarbonate covers don't suffer from this problem, since this material is radio transparent. Sometimes, if a robot is completely shielded by metal, then the solution is to place the antenna outside it. There is a risk that the antenna will get damaged, impairing signal reception, however it is a better choice than to not have a control signal at all.

The good news is that the higher the frequency, the smaller is the wavelength. Therefore, modern systems that use 2.4GHz have waves that can go through the breaches and gaps on the robot covers (such as the holes around the wheels). This allows you to place the antenna inside the robot.

All our bots have internal antennas, since we’ve changed the radio system from 75MHz to 2.4GHz. Antennas are usually a conductive wire with one fourth of the signal wavelength. Therefore, a 75MHz signal traveling at light speed (about 300,000 km/s) has a wavelength of $(3 \times 10^8$ m/s$) / (75 \times 10^6$ Hz$)$ = 4 meters (about 13 feet). Since a 4 meter long wire would be too long, the 75MHz systems use one with 1/4 of this length, 1 meter (a little over 3 feet). In practice, it is a good idea to place the antenna wire in a zig-zag pattern inside the robot. Often, the longer the antenna, the higher is its gain, so if your robot needs a signal boost then try longer antenna lengths.



You can replace the 3-foot wire from the receiver antenna by a miniantenna with less than 8 inches long. But you need an amplified antenna for a good result, such as the Deans Base-Loaded Whip (pictured to the right). Before switching from 75MHz to 2.4GHz, we used this antenna in our middleweights Touro and Titan without any problems, however it had to be placed outside the robot due to Fadaray's Cage Effect.



Avoid placing the antenna directly over metal. Ideally, it is good to have at least 6mm (1/4 inch) between any part of the antenna and a metal part. This can be achieved with a large rubber grommet or with some Lexan or Delrin spacer, such as the Delrin Antenna Mount pictured to the right, sold at The Robot MarketPlace.

Since the wavelength in 2.4GHz systems is only (3 × 10 8m/8s) / $(2.4 \times 10^9$ Hz$)$ = 0.125m (less than 5 inches), their antennas are really tiny, between 31mm and 62.5mm in length. These receivers (such as the Spektrum 2.4GHz BR6000 model pictured to the right) normally have two antennas, due to wave polarization. It is recommended to place the antennas forming a 90° angle to maximize signal reception.

Gyroscopes

To guarantee that a robot can follow a straight line, you can use gyroscopes (a.k.a. gyros), which are sensors that measure the robot orientation angle. Easy to use, they can be directly connected to most receivers. Almost all R/C hobby stores sell gyros, which are usually used in radio-controlled model helicopters. One of the cheaper models is the GWS PG-03 Micro Gyro (pictured to the right), which costs US$38 at The Robot MarketPlace, which has a great cost-benefit.

The secret to use gyros in combat robots is to adjust the feedback gain to a maximum of 20% of its full scale. This worked very well with our hobbyweight wedge Puminha using a GWS PG-03. Higher gains can make the system become unstable, since the gyroscope will pick up motor vibrations and try to compensate them.

It's not recommended to use gyros on invertible robots, because when the robot is upside down the control gain is inverted, giving positive feedback and making the robot spin out of control (a.k.a. the "Death Spin"). To solve this problem, it would be necessary to have an electronic system to turn the gyro off when the robot is flipped over, or even better, to invert the gain in this case.

Note that gyros must be very well shock-mounted inside the robot not to break. And they must be well secured because, if they get loose or shift their position inside the robot, it will go crazy or go into "Death Spin" mode.

Battery Elimination Circuit

In several model airplanes, there is a small battery pack used exclusively to power the receiver. If your electric motors generate a lot of noise, it can be a good idea to have separate packs for the motors and for the receiver, to avoid interference or signal loss. However, if the motors have noise suppression capacitors or shielded armature, then it is much simpler to use the robot's main batteries to power as well the receiver, without using a separate pack.

This small receiver pack is troublesome: it can become loose inside the robot, it increases the robot weight, and it is another battery pack that needs to be charged, hogging your chargers. In addition, there's a chance you might forget to charge them during a hectic event.

To eliminate the receiver pack, you need a Battery Elimination Circuit (BEC). The BEC is nothing more than a voltage regulator that guarantees a constant supply of, usually, 5V (or other value between 4.8V and 6V).

A few speed controllers, discussed later, already have built-in BECs. But most of them use linear regulators, such as the LM7805, which can overheat if the voltage drop is too high. Our middleweight Touro needs a BEC because the Victor speed controllers that it uses do not have this feature. We once were out of BECs, so we've experimented using BaneBots BB-12-45 speed controllers just to work as BECs. These speed controllers have a built-in BEC, however due to the voltage drop from 27.2V

(20 NiCd cells in series fully charged) to 5V they overheated, malfunctioned, and sometimes even got burnt. This speed controller is rated for 24V at most, so it is not recommended to power it beyond this voltage, unless some heatsink is attached to it to dissipate the heat from the linear regulator.

If a speed controller has a built-in BEC but you don't need to use it (because you have a separate stand-alone BEC, for instance), then it is a good idea to remove the red wire (the middle one) from the crimp connector that goes into the receiver. This will avoid it from overheating, especially if it uses a linear regulator. This tip is valid for both brushed and brushless speed controllers. There are commercially available stand-alone Battery Elimination Circuits, named UBEC (U stands for Universal), such as the S-BEC Super BEC 5V model pictured to the right. They are usually cheap, found in R/C hobby stores. A few UBECs use switching regulators, such as the LM2575, which switches the voltage on and off in order to drop its value, instead of dissipating the power from the excess voltage as heat. In this way, they are less prone to overheat or malfunction.

You can also develop your own BEC. We've used our own BECs for years with great results, as discussed in section 7.8. If you decide to build your own BEC, choose switching regulators, they don't overheat as easily as linear regulators.

Servos

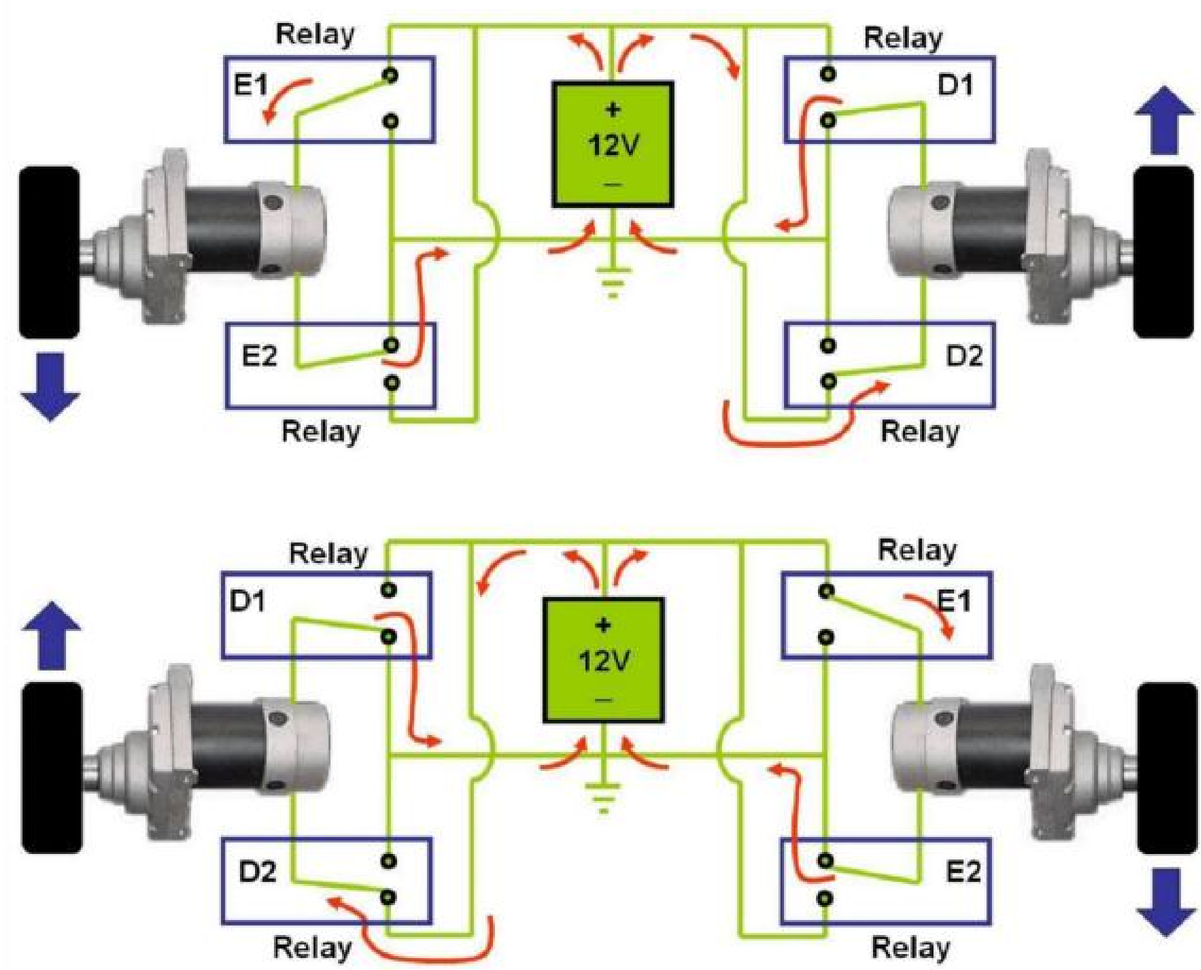
Servos, a.k.a. servo-motors, are motors with embedded position control, such as the Hitec standard model pictured to the right. In model aircrafts, small low-power servos are directly connected to the receiver, and powered by it. Servos are very practical and cheap, a few combat robots use them in the throttle system of internal combustion engines, or to mechanically control some electric switch. The problem with this approach is the great risk of servo failure after an impact, which can break them or let them become loose. In combots, always implement your control system electronically (in solid state), avoiding moving parts or servos.

Servos have been used in insect robots, actuating lifter or clamper/grabber mechanisms that need position control to properly function. With a simple modification, servos can become DC motors with continuous rotation, with an embedded electronic control to convert PPM signals into movement. The modification consists of disassembling the servo, exchanging the internal potentiometer by a tworesistor ladder, removing the plastic stop from the largest gear, and reassembling the unit. A good tutorial on this servo modification can be found at http://www.acroname.com/robotics/info/ideas/continuous/continuous.html. The modified servos can be used in the drivetrain of small robots such as insect combots or light sumo bots, however only the spin direction can be controlled, not its speed, as in the bang-bang control discussed later.

Note that servos included in radio transmitter-receiver packages do not have enough power to actuate most systems in larger bots. High torque servos should be preferred, however most of them will need to be modified if you need continuous rotation. If you need continuous rotation, including speed control, then i t is better to stick to DC motors with speed controllers, discussed next.

Controlling Brushed DC Motors Bang-Bang Control

The most primitive way to drive a DC motor is through a bang-bang controller. Is consists of using relays to create a basic H-Bridge scheme, in such a way that the motor can spin at full power forward, full power backward, or stop. The figures below show how it is possible to use 4 relays to create a bang-bang controller for two opposite wheel motors.



As shown in the upper figure, when the E1 and E2 relays are closed and the D1 and D2 are open, the robot turns to the left. On the other hand, closing relays D1 and D2 while opening E1 and E2 would make the robot turn to the right, as shown in the lower picture. To move forward, you only need to open all four relays. To move backward, you should close all relays. Finally, to stop the robot, you can choose for instance to close E2 and D2, while opening E1 and D1. In the above example, if a 12V b attery was used, then each motor would only receive either +12V, 0V or −12V.

Avoid using this type of controller, because it always provides the maximum voltage (in absolute value) to your motors, abruptly reversing their direction. This sudden reversion of the movement causes premature brush wear and can lead to broken gears because of the associated impacts (which are the origin of the “bang-bang” expression). In addition, the high inductance of the motors will create voltaic arches on the relay terminals, shortening their useful life. This kind of controller is only acceptable in low speed motors, which are not very common in a combat robot, except perhaps for a slow lifting or clamping/grabbing mechanism.

A slightly improved version of a bang-bang control would use the 4 relays in series with a single adjustable linear voltage regulator. As seen in chapter 5, the speed of a brushed DC motor is proportional to the applied voltage, so the linear regulator would control speed, while the relays would control direction. Our first combot, the middleweight Lacrainha, used such bang-bang implementation to control the speed of its wheels. The electronic board that we developed also featured an automatic system that would briefly lower the applied voltage down to zero using the linear regulator during the reversion of the relays, to avoid voltaic arches on the relay terminals.

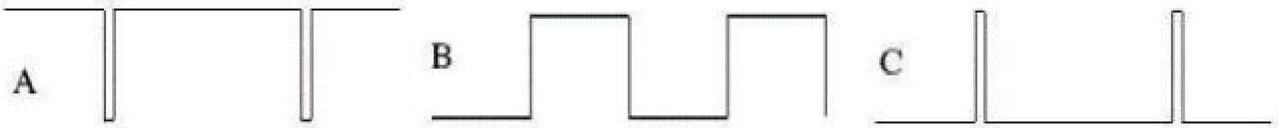
Although very simple, this version of bang-bang control has serious issues, mainly due to its low efficiency at low speeds, since all the energy from the excess voltage that is not utilized by the motor is dissipated as heat on the linear regulator. In addition, the resulting electronic system may still be unreliable since it depends on mechanical moving parts from the relays. To achieve a higher motor efficiency and more compact and reliable circuits, it is necessary to vary the motor speed in a different way, known as PWM, described next.

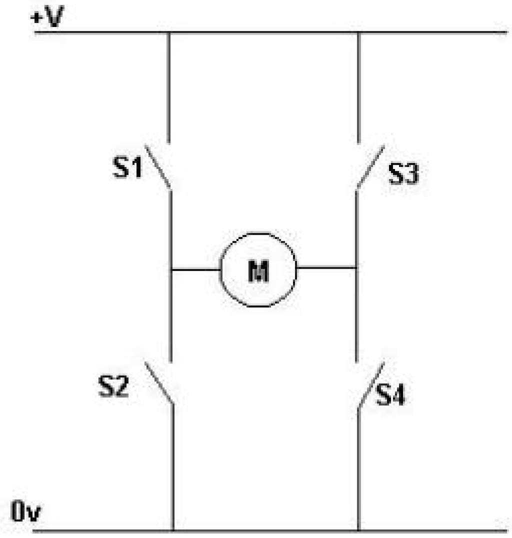
Pulse Width Modulation

The Pulse Width Modulation (PWM) method consists of turning the motor voltage on and off, on a fixed frequency basis, through an electronic switch, usually some kind of transistor. The transistor can be, for instance, a Bipolar Junction Transistor (BJT) or a Metal-Oxide Semiconductor Field Effect Transistor (MOSFET, a.k.a. FET).

The motor speed is then proportional to the ratio between the time interval T_{on} during which the motor is on and the pulse period T. This relation is named Duty Cycle (D) and, if multiplied by the peak voltage supplied to the motor, it results in an averag e voltage that can be controlled.

The figures below show three PWM signals that use the same frequency. Figure A shows the PWM signal with almost 100% D, because the time T_{on} is almost equal to the period T (the signal is high almost 100% of the time). Therefore, a motor subject to the pulse A would behave as if it was powered by almost the nominal voltage.





The pulse in figure B, on the other hand, is high half of the time, therefore T_{on} is equal to T/2, and the motor would receive about half of the nominal voltage, spinning at half-speed (50% D). Finally, the pulse in figure C would make a motor spin very slowly, since the time T_{off} during which it is off, equal to $T - T_{on}$, is almost equal to T (almost 0% D).

The efficiency of a PWM circuit is, in an ideal case, 100%, since the switching process that would power the motor wouldn't have any losses. But in practice there are losses in the used transistors, due to their resistance. Despite this, the efficiency of a well-designed PWM controller is usually above 90%.

Be careful not to confuse PWM with PPM (or even PCM), several people mix these concepts, saying that a receiver output signal is PWM. Although these signals are similar, this is not true. Both PWM and PPM are analog pulse trains, yet their functionalities are completely distinct. One of the differences is that the PPM needs to use a train pulse with a precisely defined period. The PWM, on the other hand, can have any pulse period T, what really matters is only the Duty Cycle, the ratio between T_{on} and T. The pulse frequency 1/T is an arbitrary value, but it should be large enough to avoid undesired oscillations in the motor. Typical values of 1/T are above 4kHz, sometimes higher than 16kHz. Using 20kHz or higher is a good idea to avoid buzz sounds, since it is above the range of human hearing.

H-Bridge

PWM by itself is only able to control the absolute value of the motor speed. There are a few possible ways to control the motor direction. You can, for instance, physically invert the terminals using mechanical switches such as relays and solenoids, however this would result in a bulky and impact-sensitive system, with lower reliability, as discussed before. Another option is to generate a negative voltage, however this can result in a very complex system since the main power supply is a battery, which provides direct current.

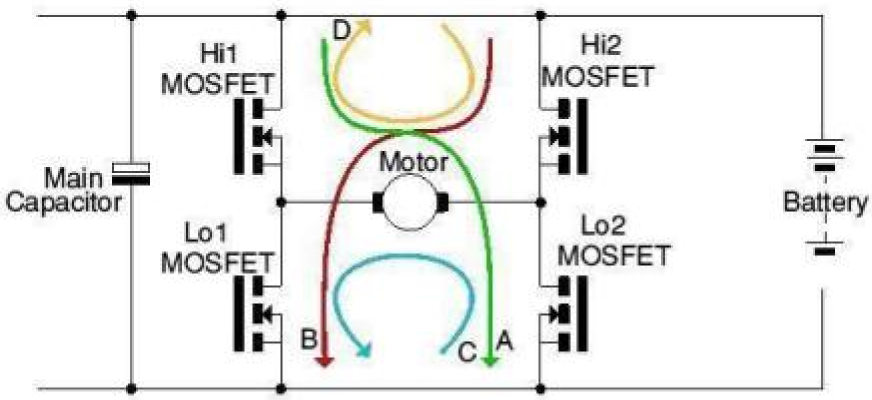
The best option is to use an H-Bridge, as mentioned before, named after the disposition of the switches in the circuit. It does not need to generate negative voltages, or to mechanically disconnect the terminals. The H -Bridge can use transistors, which easily stand high currents.

The picture to the right shows a basic H-Bridge, with a motor M and 4 transistors S1, S2, S3 and S4 that work as solid-state switches.

To make the motor spin forward with a voltage V, you just need to activate S1 and S4 and deactivate (open) S2 and S3. To spin backward with voltage V, activate S2 and S3 and deactivate S1 and S4.

To brake the motor, you can either activate only S1 and S3, or activate only S2 and S4, shorting the motor terminals. This braking effect is called motor brake and happens due to the entire energy being dissipated by the motor internal resistance, which is usually very low. This results in quick energy dissipation, which happens until the motor stops. Note that motors with very small internal resistance can generate a lot of heat in this process, so be careful!

In the figure to the right, the switches are implemented using MOSFETs, resulting in a simple H-Bridge circuit.



To spin the motor forward, the current from the battery needs to go through the FETs Hi1 and Lo2, following the path A shown in the figure. To spin backward, Hi1 and Lo2 must be deactivated, and immediately after this Hi2 and Lo1 need to be activated, making the current follow path B. Clearly, to brake the motor you can either activate Lo1 and Lo2 (for the current to follow path C, in either sense), or activate Hi1 and Hi2 (to follow path D, in either sense). Note that paths C and D are only possible because these FETs feature integrated freewheeling diodes, which allow the current to go from source to drain (the upward direction in the figure). Make sure these diodes are present, otherwise the motor inductance will fry the FETs.

PWM is very easy to be implemented into the H-Bridge. When spinning forward, it is enough to keep Hi2 and Lo1 always deactivated and Hi1 always activated, while Lo2 is only active during the T_{on} interval from the PWM, making the current follow path A.

Due to the motor's inductance, the current tends to keep a constant flow even during the T_{off} period. Therefore, during T_{off} from a forward movement, the current will follow the opposite sense of path D, flowing through the motor in the same sense as it did in path A. Note that Hi2 can remain deactivated during T_{off}, because its free-wheeling diode always allows conduction from source to drain, necessary for the opposite sense of path D (Hi2 only controls the flow from drain to source).

This path during T_{off} shorts out both motor's terminals, however with minimal energy losses. It indeed brakes the motor, but this is not a problem, because it only happens during T_{off}. It is the combination of motor acceleration (during T_{on}) and braking (during T_{off}) that allows the resulting average speed to achieve any value between zero and the top speed. As mentioned before, the period T must be small enough so that the acceleration and braking effect cannot be noticed, avoiding oscillations in the motor.

Analogously, to spin backward, you just need to keep Hi1 and Lo2 always deactivated and Hi2 always activated, while Lo1 is only active during the T_{on} interval from the PWM. During T_{on}, the current will follow path B, while during T_{off} the motor inductance will make the current flow through the motor in the same sense as it did in path B, resulting in path D in the sense described in the figure (not in its opposite sense). Note that Hi1 can remain deactivated during Toff, because its free wheeling diode always allows conduction from source to drain, necessary for path D.

A major issue with H-Bridges is an effect called shoot-through, which occurs if two switches from the same side are active at the same time, for instance Hi1 and Lo1. If this happens, the battery is shorted out, generating a very large current that usually destroys the MOSFETs.

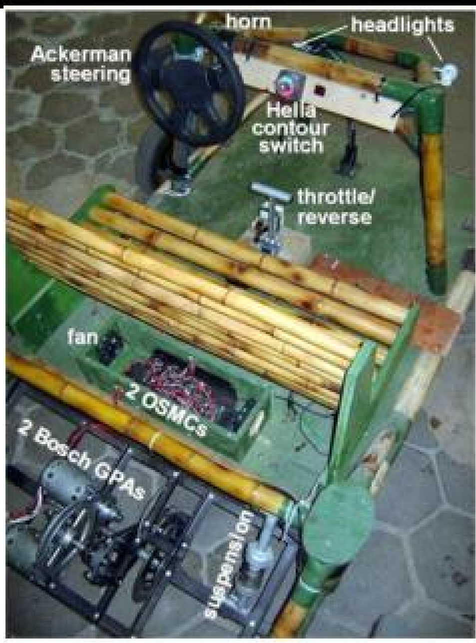
Electronic Speed Controllers

Electronic Speed Controllers, or simply ESC, are electronic power systems that implement an HBridge with PWM, to control both spin direction and speed of a motor. There are several ESCs in the market, so we'll focus on the ones that we have already tested in combat: OSMC, Victor, Scorpion and BaneBots, which are all brushed Permanent Magnet DC (PMDC) motor controllers. All these systems can be purchased, for instance, at The Robot MarketPlace, IFI Robotics, BaneBots, Trossen Robotics or Robot Power.

To control Brushless DC (BLDC) motors, a Brushless Electronic Speed Controller (BESC) is required, explained in section 7.3.6.

OSMC – Open Source Motor Controller

OSMC is a speed controller board capable of powering a single DC motor with nominal voltages between 13V and 50V, handling continuous currents of up to 160A, with 400A peaks. They are very robust, using 16 MOSFETs on an H-Bridge (4 transistors per leg), cooled by a fan. The pictures to the right show the OSMC board with and without the fan.



We have used OSMCs in a few of our combots, such as the middleweights Anubis, Ciclone and Titan. We also use 2 OSMCs to power the 2 Bosch GPA motors that drive our bamboo electric vehicle, pictured to the right. This vehicle, built by the students, is useful to carry two people and a middleweight from our lab to the "weapon testing field" from our University (a.k.a. the soccer field).

The OSMCs are a little bulky if compared with other ESCs available in the market, but they are very reliable. The middleweight wedge Max Wedge, honorable mention in the Robot Hall of Fame, used OSMCs to control the speeds of its power hungry D-Pack motors.

The Open Source Motor Controller is the collaborative result of several combot and electric vehicle builders. The OSMC diagrams can be found at http://robotpower.com/osmc_info. In this very site it is possible to buy an assembled and tested OSMC board for US$219, or a full kit including a bare-board and all components needed for the assembly at US$169.

To save some money, our team bought several bare-boards for US$29 each at Robot Power, and looked for information about the needed components and their suppliers in the OSMC Discussion Group (http://groups.yahoo.com/group/osmc) to buy them individually. The savings were not significant if compared to the Robot Power full kit, our cost per board was slightly below US$159. But at least we've learned a lot about each individual component and its vendors.

One disadvantage of the OSMC board for use in combots is the need for a separate electronic RC interface between the OSMC and the receiver, called MOB (Modular OSMC Brain), or µMOB for a smaller version. Unfortunately, the MOB/µMOB interfaces have been discontinued, so we've developed our own RC interface board, which is able to control 2 OSMCs.

IFI Victor

Victor is a family of speed controllers from IFI. These ESCs are extremely compact, almost as robust as the OSMC. We use Victors in our middleweight Touro to save precious space for other components (Touro ended up so compact that OSMCs wouldn't fit inside it). Each Victor only controls one motor, so you'll need at least two in the robot drivetrain.

There are several Victor models, however externally they have the same look, as pictured to the right. A fan is used to cool the MOSFETs from the H-Bridge. There are several different models and prices, listed next.

- Victor 884 – US$ 114.95: single channel, forward and reverse, from 6 to 15V, handling up to 40A continuously; it uses 12 MOSFETs, 6 for each direction;
- Victor 883 – US$ 149.95: single channel, forward and reverse, from 6 to 30V, up to 60A continuously, with a surge current capability of 100A for less than 2s, and 200A for less than 1s; it uses 12 MOSFETs, 6 for each direction;
- Victor 885 – US$ 199.95: single channel, forward and reverse, from 6 to 30V, up to 120A continuously, surge currents of 200A for less than 2s, and 300A for less than 1s; it uses 12 MOSFETs, 6 for each direction;
- Victor HV-36 (the model shown in the picture above) – US$199.95: single channel, forward and reverse, from 12 to 42V, up to 120A continuously, surge currents of 250A for less than 2s, and 275A for less than 1s; it uses 16 MOSFETs, 8 for each direction;
- Victor HV-48 – US$199.95: single channel, forward and reverse, from 12 to 60V, up to 90A continuously, surge currents of 200A for less than 2s, and 225A for less than 1s; it uses 16 MOSFETs, 8 for each direction;
- Victor 883SC – US$169.95: single channel, forward only (therefore it is usually used to power a spinning weapon), from 6 to 30V, up to 90A continuously, surge current capability of 100A for

less than 2s, and 200A for less than 1s; it uses 12 MOSFETs, all of them for a single direction.

- Thor 885SC – US$219.95: single channel, forward only, from 6 to 30V, up to 150A continuously, surge current capability of 200A for less than 2s, and 300A for less than 1s.

Victors can be connected directly to the receiver, without the need for an external RC interface (such as the MOB needed by the OSMC). However, a signal booster cable (US$ 15, pictured to the right) is highly recommended between the receiver and the Victor, improving the quality of the signal. The booster cable is also interesting for robots that suffer from radio signal noise or loss, especially in combots with internal combustion engines.

IFI also sells speed controllers specifically developed to drive spinning weapons, denoted by the SC (Spin Controller) suffix, such as the Victor 883SC and Thor 885SC discussed above. They only work in one direction, but they feature softer acceleration and deceleration ramps that minimize current peaks, saving battery capacity. With them it is possible to control the speed of a spinning weapon, which can be especially useful in three situations:

1. batteries not lasting the entire match: you could spin the weapon at an intermediate speed, saving the batteries, and accelerate to full speed a few moments before striking your opponent;
2. damaged or cracked weapon: if, after a tough match, some structural damage or crack is found on a weapon component that cannot be replaced, then it is possible to avoid a broken weapon in the following fights adopting a lower speed;
3. robot vibration when the weapon is at full speed: slightly slowing down the weapon can minimize this problem, allowing you to control the weapon speed to avoid natural frequencies.

Despite the acceleration and deceleration ramp features from the SC types, most builders prefer to use in their weapon systems ESCs that are capable of forward and reverse spin, resulting in a fully reversible weapon system (which is very useful, for instance, for invertible drumbots). If you don't need a fully reversible weapon, such as in most horizontal bar spinners, and if you don't care about controlling the weapon speed, then it is a good idea to use solenoids, due their lower price tag and higher current limits. Solenoids are studied in section 7.4.

We have been using Victors HV-36 to control the drive system of all our combots from 30lb to 120lb. Our 4-wheel drive hobbyweight wedge Puminha also uses Victors HV-36, each one is used to power both drive motors from each side of the robot. But, since Victors in hobbyweights are usually overkill, we've removed their fans in Puminha without needing to worry about overheating problems, saving precious space.

The Victor HV-36 is a better option than the Victor 885 even for 24V combots, because it uses 16 MOSFETs instead of only 12, better handling high peak currents from high power motors. And the use of the Victor HV-36 also allows future robot upgrades up to 36V.

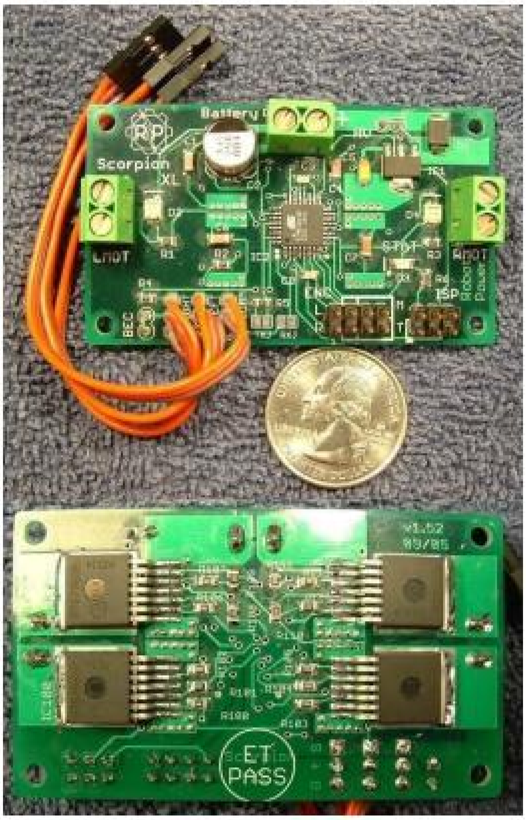
The Victor HV-36 is also a good option for the weapon system. After RoboGames 2008, we've replaced the weapon solenoids from our middleweights Touro and Titan, lightweight Touro Light, and hobbyweight Tourinho, with Victors HV-36, resulting in fully reversible weapons with speed control. One Victor HV-36 is needed for each Magmotor S28-150 or DeWalt 18V motor from the weapon systems of Touro Light and Tourinho. But we use 2 Victors HV-36 for each Magmotor S28-400 from the weapon system of our middleweights, as explained in section 7.7.3, for redundancy and to avoid burning the ESCs due to the high stall current of this motor.

Since the fans are really important to cool down the MOSFETs, it is recommended to protect them. Debris or even loose wires from the electric system can touch the fan blades, making it stop. To avoid this, you can use a fan grill (pictured to the right), which can be made out of steel wire or from an aluminum sheet. Another good idea is to use the IFI Stainless Steel PWM Clip, which locks the signal cables onto the Victors, preventing the connectors from popping loose.

Robot Power Scorpion

Victors and OSMCs are suited for larger robots, typically featherweights (30lb) or heavier. They are expensive and relatively large to be used in robots that weigh 15lb or less. Robot Power sells four ESCs that are a good option for those lighter combots.

Scorpion XL and XXL



The most famous product from the Scorpion line is the Scorpion XL (US$119.99, pictured to the right showing its front and back sides), a controller that offers two PWM output channels, normally used to control both drivetrain sides. It can handle from 4.8 to 28V, up to 12.5A continuous, and 45A peaks per output, which can be combined into a single channel to provide higher current limits.

In 2008, a beefier version was introduced, the Scorpion XXL (US$159.99). It uses the same board from the XL version but, instead of one BTS7960B power IC per Hbridge leg, this version has a pair of them, one on top of the other. This allows a higher continuous current of 20A. The channels can also be combined, resulting in a single PWM output that could handle continuous 40A. To withstand the higher currents, the battery screw terminals were replaced with wire pig-tails.

Since both XL and XXL versions use the same board, most features are the same, such as current and temperature limiting, flip input to allow a radio channel to invert the robot drive system commands (to be used if an invertible robot is flipped over), five signal mixing options, exponential output, built-in BEC, and failsafe.

These controllers are a good choice for combots up to 15lb, especially for their size and built-in features. As a matter of fact, until 2007 our hobbyweights Tourinho and Puminha used Scorpions XL. However, in 2008 we've reduced the size of Tourinho's chassis to save weight, which required us to switch the Scorpion XL board to two BaneBots BB-12-45 ESCs, which are smaller in size. In addition, in 2008 we've replaced Puminha's drive motors to higher power models that would greatly exceed the 45A peaks, so its Scorpion XL was switched to a pair of Victors.

Scorpion XL is a very good board, however there are a few tips to make it bulletproof, as we've learned after two years using and abusing its 2006 version in our hobbyweights:

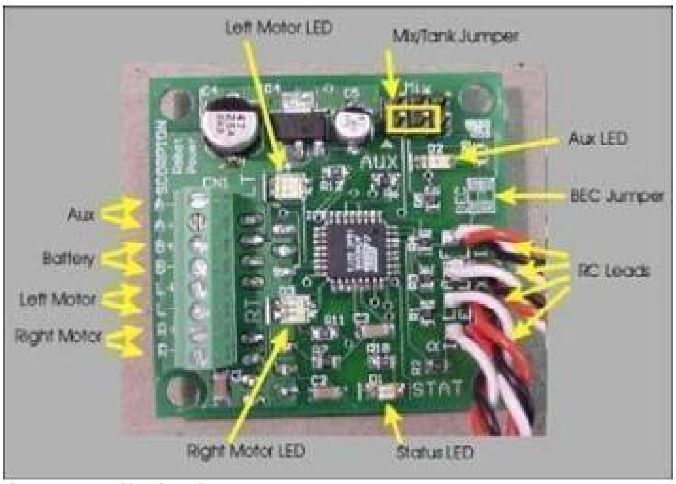
- the large SMD capacitor (silver cylinder with black notch under the battery writing in the top picture) can become loose or even break during an impact, unless the board is very well shockmounted; to avoid this, you can use epoxy resin without metal additives, hot glue, or even tape, to better secure the capacitor; if the capacitor gets knocked off, the board will present an inconsistent behavior; you can replace the capacitor by carefully soldering it onto the surface

contacts, bending it towards the board over the Scorpion writing, and gluing it to the PCB; • the green screw terminals are prone to become unfastened during combat, enabling the wires to fall off; we've removed these terminals and soldered the wires directly to the board;

- the flip feature doesn't work in the 2006 versions that we've bought; the solution, if available, is to implement this function using mixes in the radio transmitter;
- to improve heat dissipation and stretch the current limits a little bit, it is possible to attach heatsinks onto the BTS chips; old CPUs are a good source for small aluminum heatsinks.

Scorpion HX

The Scorpion HX (pictured to the right) is a great option for lighter combots such as beetleweights and antweights. The board weighs only 0.78oz (22g), with a compact 1.6" × 1.6" × 0.5" size.



Unlike the XL/XXL controllers, this board features 3 channels. Two of them are PWM outputs, usually used to control drive motors, and one on-off switch (Aux/Weapon channel) that can be used to operate a brushed DC motor in a single direction, usually in the weapon system.

The PWM channels accept motors between 4.8V and 22V, delivering each 2.5A continuously with 6A peaks. If you need a few extra amps, it is possible to attach a heatsink onto the drive chips. Regarding the Aux/Weapon channel, it can handle constant 12A and as much as 35A peaks.

Other characteristics include flip (which works very well), current and temperature protection, channel mixing, BEC (which must be deactivated for input voltages above 12V), and safe weapon channel start (where the robot won't start if the weapon switch is activated during power-up), besides failsafe in all channels.

Mini-Touro used a Scorpion HX ESC during RoboGames 2006, with great results. However, in 2007 we've decided to use a brushless motor to power its drum, instead of a brushed one, so we had to replace the HX board with two BaneBots BB-3-9 ESCs (for the drive system), to give room for the BESC needed by the weapon motor.

Scorpion Mini

The Scorpion Mini (pictured to the right) is the smallest ESC from Robot Power, weighing only 0.21oz (6g) and measuring 0.625” × 1.6” × 0.4”. It is a single channel PWM H-Bridge, which takes input voltages from 4.8 to 18V, or up to 34V if an external 5V supply is used. Current limits are 2.5A continuous without a heatsink, up to 4A with one, and 6A peaks.

Its version 2.4 includes BEC, which must be disabled if using voltages above 16V, and failsafe. It also features limit switch inputs that stop motion when pressed, which makes it an interesting choice to actuate clamper or lifter weapon systems in insect combots.

BaneBots

BaneBots is another manufacturer with ESCs aimed for light robot classes, from fairyweights to hobbyweights. There are three versions, all featuring a single fully reversible PWM output that can handle inputs from 6 to 24V: the BB-12-45, the BB-5-18, and the BB-3-9, pictured below from left to right, respectively.

The larger unit, the BB-12-45 (US$57), can withstand 12A continuously and 45A peaks (thus its name 12-45), presenting current limiting and thermal protection. It also counts with an integrated

BEC, failsafe, and neutral start. It also includes receiver cables and leads to the motor and power supply. It weighs 0.98oz (26.3g) and measures 1.7” × 1.1” × 0.5”, including the orange shrink wrap. It is a good option for most hobbyweights.

The BB-5-18 (US$ 46.50) is the intermediate model, it can withstand 5A continuously and 18A surges (thus its name 5-18). Similarly to the BB-12-45, it includes all wires, it has an integrated BEC, failsafe, and neutral start. It measures 1.45” × 0.825” × 0.375” and weighs only 0.60oz (17g).

The tiniest model is the BB-3-9 (US$28.75), it only weighs 0.33oz (9.4g). Its dimensions are 1.2” × 0.52” × 0.29”, allowing it to fit in small spaces inside the robot. As its 3-9 name suggests, it can handle 3A continuously and 9A peaks.

One great feature that is common to all BB speed controllers is their modularity. Single channel units are easier to fit inside the robot, because several different orientation combinations can be experimented. In addition, if for instance a wheel locks up and damages your ESC, there is no need to replace the entire electronic system, only replacing the ESC associated with that channel.

We use BaneBots speed controllers in our lighter combots: the hobbyweight Tourinho uses a pair of BB-12-45 for its drive system, while the beetleweight Mini-Touro, the antweight MicroTouro, and the fairyweight Pocket, use each a pair of BB-3-9 ESCs.

Despite their advantages, BaneBots ESCs aren’t fail-proof. The BEC circuit heats up very easily, leading to unexpected failures when used at 24V. One of our BB-12-45 caught fire while tested at 24V drawing only 500mA, due to the BEC issue. It’s probably a good idea to disable their BEC if you’re using input voltages near 24V, which might require you to install a UBEC, discussed in section 7.1.5. Another issue is that the names from all of its electronic components have been sanded out by the manufacturer, making any homemade repair impossible.

Other Brushed Motor Speed Controllers

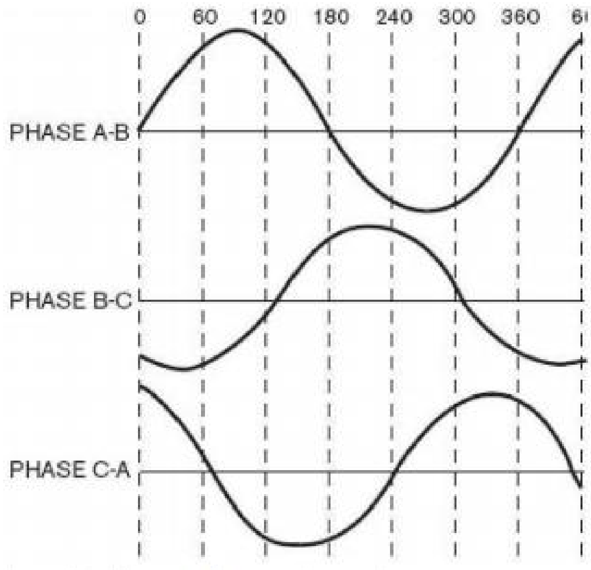
There are several other speed controllers in the market. A few of them are very sophisticated, including several programmable features, input voltages of up to 48V with peak currents exceeding 100A, tachometer and potentiometer inputs for closed-loop speed and position control, regenerative braking, temperature sensors, RC and microcomputer interfaces, and much more.

A few examples of sophisticated controllers are the RoboteQ, AmpFlow and Sidewinder, pictured below. They can independently control 2 brushed DC motors at the same time, which is perfect for the drivetrain of most combat robots. They also have mounting brackets for easy installation. They are good options for middleweights, heavyweights, and super heavyweights.

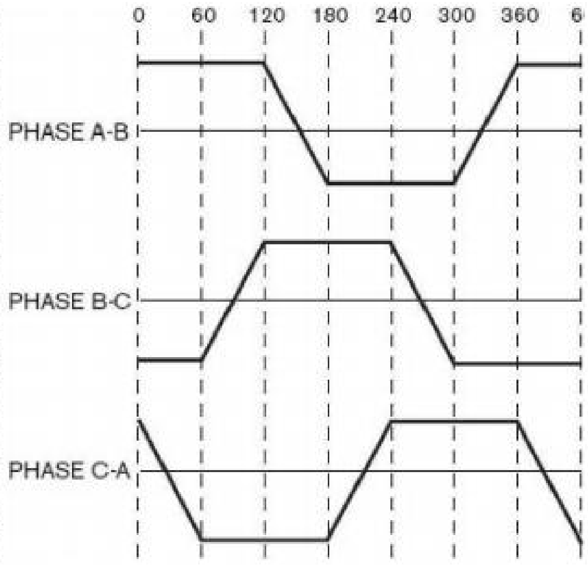


Brushless Electronic Speed Controllers

As seen in chapter 5, brushless DC (BLDC) motors demand a specific controller to be powered. This kind of motor resembles an alternate current (AC) synchronous three-phase motor, in which three waves with 120° shifted phases and same frequency actuate the rotation. Similarly to AC motors, brushless motors also spin due to three 120° phase-shifted waves but, instead of sinusoidal, these waves are three trapezoidal PWM signals, as pictured below. Each PWM signal acts over each one of the three motor windings from the brushless motor.

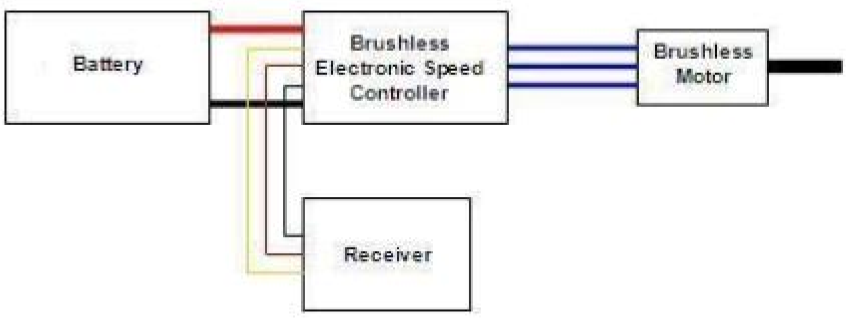


sinusoidal inputs for AC three-phase motors



trapezoidal PWM inputs for brushless motors

The wiring scheme to control brushless motors using a BESC is a little different from the one from brushed DC motors controlled by an ESC, as pictured to the right. Never connect the BESC to the battery with reverse polarity. Note that the BESC can be connected to the brushless motor using any combination of the motor's 3 wires. However, if the brushless motor spins in the wrong direction, then simply switch connection between any 2 wires. If, when the motor is unloaded, it takes too long to start spinning, you could try other wiring combinations between the BESC and the motor, this might solve the problem.



In order to start the motor, the controller must know which of the three windings should be first triggered. There are two methods to do that. The first is to have position sensors on the motor to measure the angle of the rotor, however brushless motors with sensors usually have high cost and complexity.

The other method is sensorless, where the controller is able to figure out the angle of the rotor by measuring the differences between the windings inductances. This is why you should never shorten brushless motor wires, this would change their inductances and confuse a sensorless system. Sometimes the motor wires are the coil wires extended out, so if you don't cut them at exactly the same length, getting them cleaned up nicely, removing all the varnish, and evenly tinning them, then your motor will single phase and not work, but just chatter. So, leave the motor wires alone.

On the other hand, long wires between the battery and the BESC should be avoided. This is why most BESCs (and ESCs as well, for brushed DC motors) have a capacitor between their input power lines: its role is to suppress voltage spikes that could fry the speed controller. So, it is a good idea to shorten the wires between the battery and the BESC, placing these components close together inside the robot, and leaving the brushless motor wires with their original length. Long motor wires only lose a bit of power, they don't damage the BESC. There are several sensorless Brushless Electronic Speed Controllers (BESC) in the market, from various manufacturers, such as Castle Creations, Hextronik, and Dynamite. The picture to the right shows the Castle Creations Phoenix 25 BESC, used in the weapon system of our beetleweight Mini-Touro. The downside of most BESCs is that they can only spin the motor in one direction. To reverse rotation during combat, you

can commute any two of the motor wires, which can be done, for instance, using traditional or solidstate relays. Or you can use a reversible BESC, usually developed for R/C model cars, which can reverse the spin direction of brushless motor s using a three phase H-Bridge circuit.

A few important aspects when choosing a BESC are its maximum voltage capacity, maximum continuous and peak currents, and motor reversion ability. Brushless motor reversion is still uncommon in combots, however it is very useful to reverse the direction of a weapon, allowing for instance an invertible drumbot that was flipped over to reverse its drum to continue launching the opponent. Brushless motor reversion has also potential applications in the robot drivetrain, resulting in a very efficient drive system with a high power-to-weight ratio.

A usual feature found in most BESCs is a BEC to power the receiver. But most BECs found in BESCs use linear regulators, which may overheat at high voltages. You may need to disable the BEC if the battery voltage is high. For instance, the HXT120 BESC from Hextronik (pictured to the right) can handle 120A continuously if properly ventilated, at input voltages of up to 24V. The BEC can handle up to 2A, but only if the input voltage is below 12V, otherwise it will overheat. For voltages higher than 12V, the BEC needs to be disabled, which is done by removing the middle red wire from the receiver connector. We use this BESC to power the weapon motors from our featherweight Touro Feather and hobbyweight Touro Jr.

High quality BESCs usually feature soft start, to prevent damages to fragile gearboxes, along with several programmable features such as:

- low-voltage cut-off: if the battery voltage per cell drops below a certain threshold, the motor speed is reduced or even brought to zero, to prevent damaging lithium batteries; the problem with this feature is that combat robot weapons usually require high current bursts for short periods, which can momentarily lower the battery voltage below the threshold, turning your weapon off; the default threshold settings are usually too conservative to use in combat, so program this feature to the lowest allowed value;
- brake type: option to brake the motor when the radio transmitter stick is in neutral position; this can be useful to stop spinning weapons in less than 60 seconds, as required in most competitions; note that the entire energy is dissipated in the motor, which might overheat;
- throttle range: option to limit the maximum power output, which is an interesting option in case the motor is overheating and/or drawing too much current from the batteries during combat; • timing advance: ability to advance motor timing, resulting in a higher top speed in one direction;
- reverse delay: if a BESC is capable of reversing the motor by itself, without the aid of any external system, then it will probably feature this option; it defines a time delay between the moment when the radio control stick is moved to a reverse position and the moment when the spin reversion is fully commanded; this feature is used to prevent damages to gearboxes from the impact caused by a sudden reversion; if you need a very fast reversion, then set this delay to the lowest possible value, however keep in mind that the entire braking energy will need to be dissipated by the motor, which can end up overheating.

BESC programming methods vary among manufacturers and models. High-end models can be programmed via USB (using an appropriate connector, such as the one pictured to the right) in a computer. This is a really nice feature, making it possible to tweak the performance on the fly using computer software with intuitive user interfaces.

Cheaper models are usually programmed using the radio transmitter throttle stick, with the aid of feedback beep sounds emitted by the BESC, which can be quite confusing (and very annoying).

Other BESCs allow the use of a special programming card, such as the one pictured to the right, which significantly eases the task. It is plugged between the receiver and the BESC, allowing you to program it according to a series of LED indicators. The card usually has buttons at the bottom to navigate through the menu, select the desired feature, and then effectively change it in the BESC.

Solenoids

High power motors will probably require expensive speed controllers to be activated. A few motors, such as the D-Pack, are so powerful that they can even blow tough ESCs such as OSMCs, if care is not taken. Using your ESC in combat near its operational limit is risky, not to mention that its life will be significantly shortened.

So, if you don't really need speed or direction control, such as in the spinning weapon motor of most combots that have self-righting mechanisms, then solenoids might be a good option. Solenoids, a.k.a. contactors, are basically relays on steroids, capable of handling high currents to operate powerful weapons. Two of the most famous solenoids used in combots are the WhiteRodgers 586 SPDT, and Team Whyachi's TW-C1, discussed next.

White-Rodgers 586 SPDT

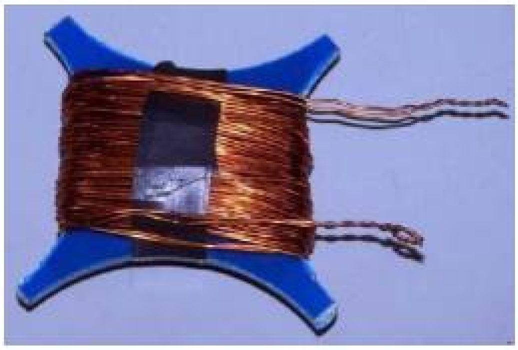
The White-Rodgers 586 SPDT (pictured to the right, sold at The Robot MarketPlace for US$96) handles 200A continuous and withstands peaks higher than 600A, being, therefore, appropriate for almost any combot weapon even in super heavyweights. It is the solenoid used to power the Etek motor from our middleweight horizontal bar spinner Ciclone.

The SPDT in its name stands for Single Pull Double Throw, in other words, a single signal can switch between the Normally Open (NO) and Normally Closed (NC) terminals, necessary to activate or brake the weapon.

Most competitions require a robot's weapon to stop in less than 1 minute, therefore spinners with large weapon inertia might need some sort of braking system - it is not enough to just turn the motor off. A very simple braking system can be implemented in a brushed DC motor by shorting its leads. A spinning motor that is shorted out will become a generator, the same principle used in power plants, producing a current that is dissipated by the internal resistance of the motor, converting its kinetic energy into heat. This will effectively brake the system, as it can be easily verified by turning by hand the shaft of a motor with or without its leads shorted out. It will be much harder to turn the shaft when the leads are shorted.

Therefore, to implement a braking system, it is enough to connect the motor to the solenoid NO terminals, while shorting the terminals from the NC pair passing through the motor. When the solenoid is activated, current will flow through the NO terminals and the motor will accelerate. When deactivated, the motor leads will be shorted out at the NC terminals, braking the motor.

However, the kinetic energy of a few weapons, such as spinning bars or disks, is so high that shorting out the motor would simply fry it, due to the entire energy being dissipated as heat by the motor's usually small internal resistance. To prevent this, a power resistor must be placed between the NC leads, instead of just shorting them. Its value can be, for instance, 9 times the internal resistance of the motor, which would result in 90% of the weapon's energy being dissipated by this resistor, and only 10% by the motor. Note, however, that higher resistance values mean a longer braking time, because the dissipated power is inversely proportional to the resistance of the circuit. Therefore, choose a power resistor with a high enough resistance to avoid frying the motor, but not too high to guarantee that the weapon will stop in significantly less than 60 seconds.



Instead of a power resistor, it is possible to use a long coated copper wire that is wound up on some heat resistant material (as pictured to the right, wound up around a blue piece of garolite). Knowing the wire resistivity, you can calculate the needed length for the desired resistance. Then, wind it up as a reel to keep its size very compact. The advantages of this braking system are its low cost, its low weight, and the efficiency of the heat dissipation provided by the long wire length. It is also easy to change its resistance: if the weapon is taking too long to stop, then just cut a few feet of wire from the reel to lower its resistance. It is important to use a heat resistant material such as garolite, because the temperatures while braking a powerful weapon can reach very high values. The blue garolite from our braking system usually comes out with a burntblack color after a competition.

We have used this wire braking solution in our middleweight horizontal bar spinners Ciclone and Titan. It works really well. Touro’s drum doesn’t need a brake, despite the weapon’s high kinetic energy, because the moment of inertia of the drum is not as high as the one from the spinning bars. Even if two spinning weapons have the same kinetic energy, the one with lower moment of inertia will most likely stop faster, because it will have a much higher spin speed (to result in the same energy) that will result in larger bearing friction. Touro’s drum stops in a few seconds, without the need for a braking system. Drumbots in general do not need a braking system to stop under 60 seconds. But horizontal and vertical spinners, which have weapons with very large moments of inertia, might need it. A few powerful spinners hit the arena walls after the end of a match to slow down and stop their weapon in much less than 60 seconds, to avoid delaying the event - but you must check if this is allowed, especially with the arena’s owner!

Team Whyachi TW-C1

Team Whyachi’s TW-C1 contactor (pictured to the right) is also SPDT, however it is smaller, lighter and cheaper than White-Rodger's 586. It tolerates, at 48V, currents of 80A continuous, 240A for 3 minutes, and 500A for 25 seconds.

Until 2008, both Touro and Titan had TW-C1s powering their weapons. The only issue with both presented solenoids is their plastic casing, which can break due to the high impact accelerations. Therefore, always shock-mount solenoids inside the robot structure.

Wiring

To connect the previously presented components, high quality wires and connectors are needed. The wires must bear high currents, while the connectors can't become loose during impacts. These components are presented next.

Wires

Wires must be very flexible, making it easy to route them through the robot's inside without rupturing the solders during impacts. Therefore, never use cables with a solid metallic core, use instead cables with multiple wires. A good example is the Deans Wet Noodle (pictured to the right), formed by over a thousand extremely thin wires.

Also, it is important to leave a little slack in all wires, in order to avoid them from getting stretched, ruptured or disconnected during combat, especially if the robot suffers structural deformations or if internal parts slightly move.

It is important to keep in mind the current ratings, which depends on the wire diameter (gauge, usually measured in AWG, which stands for American Wire Gauge) and the isolation material. The higher the AWG, the smaller the wire diameter is. When it comes to isolation, there are two usual types: PVC, which withstands temperatures up to 221°F (105°C), and silicone, withstanding up to 392°F (200°C). The highest current ratings for typical wire gauges are the following:

8 AWG: 70A to 80A continuous (PVC); 100 to 110A continuous and 500A peaks (silicone);

10 AWG: 50 to 60A continuous (PVC); 75 to 85A continuous and 350A peaks (silicone);

12 AWG: 35 to 45A continuous (PVC); 55 to 65A continuous and 200A peaks (silicone);

14 AWG: 30 to 35A continuous (PVC); 45 to 50A continuous (silicone).

The picture to the right shows a typical device used to measure wire gauges, ranging from the 0.3249" (8.252mm) diameter 0 AWG to the 0.005" (0.127mm) diameter 36 AWG.

Note that most wires can withstand without problems very brief and sporadic current peaks that are 4 times higher than the continuous limits

A good tip is to use zip-ties to organize the wiring inside the robot. This can save a lot of time during a pitstop.

Also, always use rubber grommets (pictured to the right) to protect wires that need to go through metallic plates: if not protected, the friction between the wires and a hole in a metallic part can cut the isolation layer and cause short-circuits. Smoothing out with a metal file is also a good idea.

Terminals, Plugs and Connectors

Terminals, plugs and connectors are critical in combat, because they must withstand impact accelerations and high currents. Avoid using fork, slide or quick connection terminals, since they are prone to disconnect upon impacts. Always use ring terminals (pictured below to the left). Fasten tightly the connectors with nuts, along with a pressure washer. Never place a washer between the contacts, because it usually has a large electric resistance. Also, apply liquid electrical tape (as pictured in white to the right, applied to a WhiteRodgers 586 SPDT solenoid) to avoid shorted contacts due to metal debris.

It is a good idea to stick several pieces of double face adhesive tape inside the robot, especially on the inner part of the bottom cover, near the electronics. If the tape is very sticky, such as the VHB4910 tape, it works as a "flypaper" to pick up any debris that enters the robot, which could cause problems such as shorting out the electronics or getting stuck in the clearances between the wheels and the structure.

Connectors must have a very low resistance and lock in well. An excellent connector is the Deans Ultra (pictured to the right), which withstands continuous 80A. Their maximum peak current is much higher: Touro's weapon motor draws almost 300A for a couple of seconds in the beginning of the drum acceleration, as we've measured, which goes through a single Deans Ultra without problems. We use these connectors on the batteries, motors and ESCs from all our hobby, feather, light and middleweight combots. An extra protection is, after connecting them, to duct tape them together to make sure they won't get disconnected. Be careful with knock-offs with cheap plastic (not nylon) housings that easily melt during soldering, letting their contacts come loose.

Another high power connector we've used is the Anderson PowerPole. The most common are the 45A version (pictured to the right) and the 75A. The 75A tolerates higher current peaks than the Deans Ultra, however it is a little bulky.

For lower currents, we use the Deans Micro Plug (pictured to the right),which is much smaller but even so tolerate currents higher than 20A. We use them as fan connectors in the bigger bots, as well as to connect the brushed DC motors from our beetleweight Mini-Touro.

Power Switches

Power switches, or other on/off mechanisms, are mandatory in every combot (except, in a few cases, in insect class robots without active weapons). The switch must interrupt any current flux inside the robot, even low power ones. It is not enough to be able to switch off the RC interface or the receiver, even if this is enough to stop your robot, because any malfunction could still activate your weapon or drive system.

For this reason, such switches must handle the entire amperage that goes through the robot, including the current required by the drivetrain and weapon systems. In the case of our middleweight Touro, current peaks can reach up to 400A. There are automotive switches that can handle this level of current, however they are usually bulky and heavy. Two small and light weight switches, popular with combot builders, are the Hella Master Power Switch and the Team Whyachi MS-2. The Hella switch (US$18, pictured to the right) can take continuous 100A, 500A for 10 seconds, and 1,000A current peaks. It is turned on or off by a red key that is not very convenient to use in combat. Most builders cut off the head of the red key, and file a notch on the remaining stub (as pictured to the right), allowing it to be switched on or off using a flathead screwdriver. We have used this switch in our middleweights Ciclone and Titan. Team Whyachi's MS-2 switch (pictured to the right) is better and smaller than the Hella switch, however it is more expensive (US$65). It withstands continuous 175A, 500A for 3 minutes, and 1,000A for 25 seconds. To turn the robot on or off, you need to insert an Allen wrench (also pictured to the right) and turn it four times in the appropriate direction. These four turns make it almost impossi ble to have the switch disconnected due to vibrations during combat.

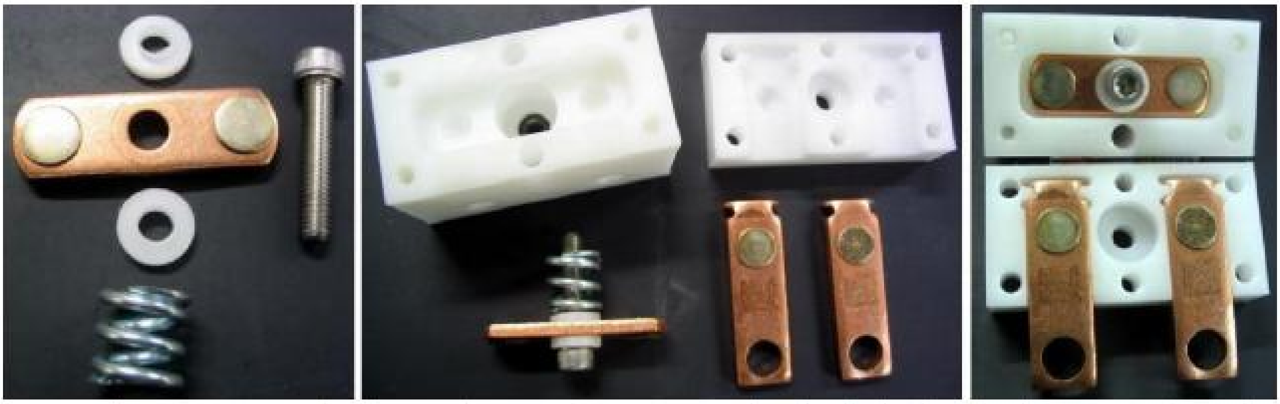
There is a smaller version that is rated for a lower current, the MS-05, good for hobbyweights (12lb), however it isn't much cheaper, at US$45. It takes continuous 40A, 140A for 3 minutes, and 250A for 25 seconds. In high power featherweights or heavier combots, it is better to use the MS-2, due its higher current rating. Our featherweight Touro Feather, which has a PolyQuest lithiumpolymer battery capable of delivering up to 225A, had a few problems with the MS-05; since we've replaced it with the MS-2 switch, we haven't had any more problems.

The MS-2 and MS-05 switches seem a little pricey, but they have an excellent cost-benefit relation considering that they are a vital part of the robot. And their manufacturing is not so simple, it involves the milling of two Delrin blocks, one for the body of the switch and the other for its cover, as pictured below.

milled Delrin body of the switch (inner and outer views)	milled Delrin cover of the switch (inner and outer views)

The Delrin body has an embedded threaded nut (in black at its center), to which a long Allen screw is attached, along with a copper bar with gold contacts, two plastic non-conductive washers, and a spring, pictured below to the left. After the switch is assembled, the long Allen screw will have the function to position the copper bar to close or open the circuit. As it is screwed, it will drive the copper bar away from the two copper terminals (pictured below in the center), opening the circuit and turning the robot off. When the long Allen screw is unscrewed, the spring-return will make the copper bar touch the two copper terminals, turning the robot on.

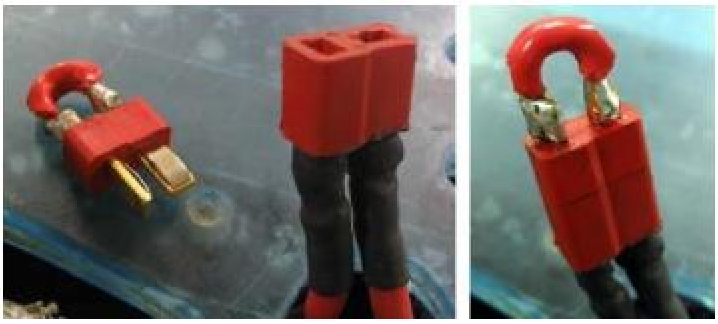
The picture below to the right shows the assembled body and cover, before they are attached together using 4 small Allen screws. The switch itself is attached to the robot with the aid of two threaded holes in its Delrin body.



Note that the non-conductive washers are important to electrically isolate the long screw from the copper bar, besides reducing friction between them.

Note also that the spring is important to guarantee contact between the copper bar and the terminals. It also works as a spring-lock to avoid the screw from turning due to vibration. And, if the switch breaks due to an impact, it is likely that it will remain in the “on” position due to the spring, keeping the robot alive during a violent match. This feature has saved us during RoboGames 2006: the violent impacts during the match against the undercutter The Mortician managed to detach the long Allen screw from the threaded nut in the Delrin body of Touro’s MS-2 switch; however, Touro continued to fully function because the spring was able to guarantee contact between the terminals.

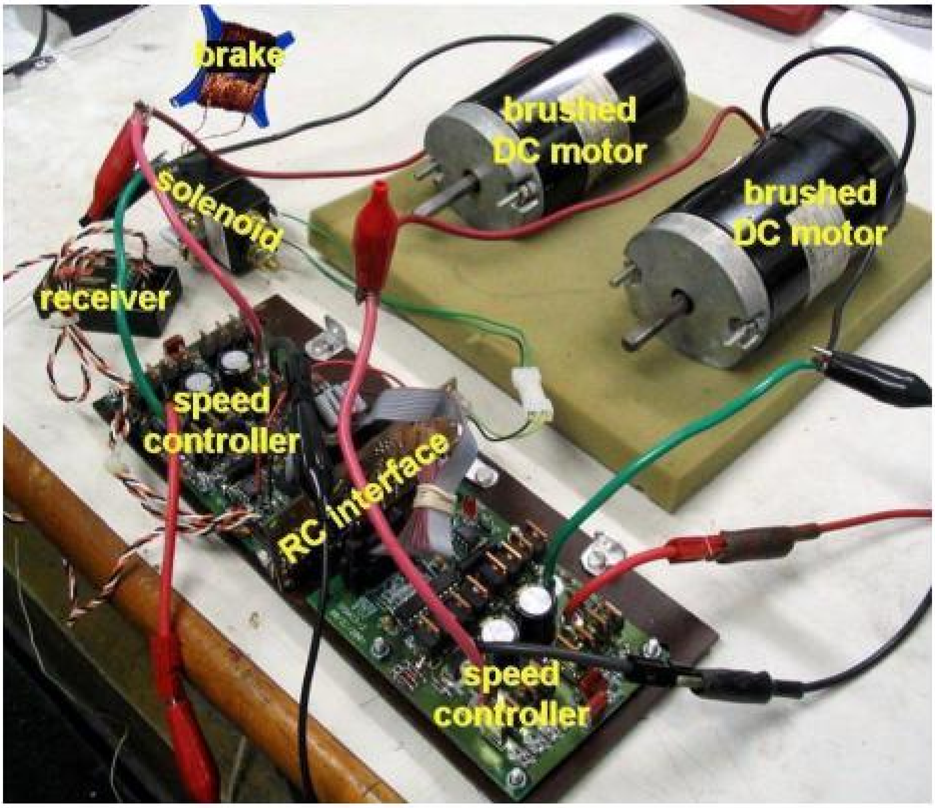
There are also even simpler and cheaper switches that can be made. One of them is the one used in the drive system of our middleweight Ciclone. A Hella switch controls the weapon, at 24V, but the drive motors use an 18V cordless drill battery pack, which needs a second switch. The adopted solution costs only a few bucks: a pair of Deans Ultra connectors wired as a jumper, as shown to the right while turned off (left picture) or on (right picture). The wire that is connected to the battery positive was cut in two, opening the circuit. Then, the two cut pieces were soldered to a female Deans Ultra plug. Next a short wire was soldered to both terminals of a male Deans Ultra plug. Don’t forget to isolate well both plugs (the Deans jumper in the pictures is shown without its isolation tape). Connecting the male into the female plug closes the circuit and turns the robot on. It’s important to insert the connected plugs well inside the robot, to avoid them from being knocked-off by an opponent. The bar spinner The Mortician was able to win a RoboGames 2006 match by knock out after knocking off a jumper switch from the launcher Sub Zero (as pictured to the right), even though the jumper was almost completely inserted into the robot.



An even simpler solution is to not use a switch. The robot must have an opening that allows the driver to directly connect or disconnect the battery (or batteries). This is the solution used in our hobbyweights (as pictured to the right) and smaller robots. A Deans Ultra female connector is soldered to the battery leads (never solder the male connector to a battery, to avoid accidental shorts). The robot electronics uses a male Deans Ultra. To turn the robot on, just connect the plugs insert them into the robot opening, and cover it to protect against debris. Make sure that the cover can be easily attached to the bot.

Connection Schemes

In this section, it is shown how to connect the presented components in a combat robot. A typical configuration is pictured to the right.



In this picture, the receiver converts the radio signals into a PPM signal, which is then interpreted by the RC interface (which is described in details in section 7.8.2).

This RC interface then generates low power PWM signals that are sent to the speed controllers, such as OSMCs (as shown in the picture), which amplify them to a high power PWM output to control the speed of the brushed DC motors used in the robot drivetrain.

The RC interface is also able to activate the solenoid that triggers the weapon motor (not shown in this picture), using a copper wire reel connected to the Normally Closed (NC) terminals to brake the weapon.

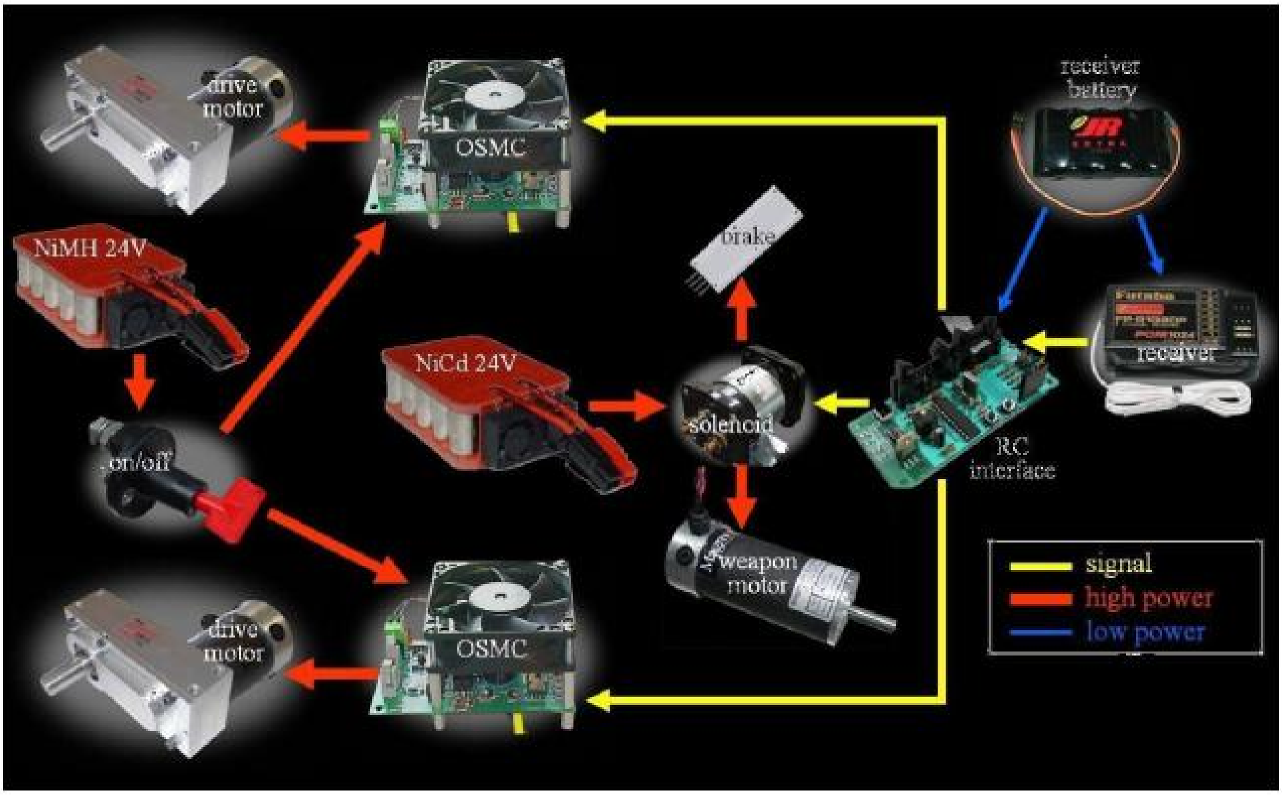
Three connection schemes are presented next. The first is a classic scheme, often used by beginners, which will surely not pass safety inspection. The second one is an improved version, which addresses all the issues from the first one. Finally, a third connection scheme is shown, better than the second one if you need a fully reversible weapon with speed control.

Classic Connection Scheme

The figure in the next page shows a classic connection scheme, including components that are sized to a middleweight combot.

It uses one Nickel-Cadmium (NiCd) battery pack to power the weapon, which is a good option due to its ability to provide very high peak currents (as it will be studied in chapter 8).

And it uses one Nickel-Metal Hydride (NiMH) battery pack for the drivetrain, which is also a good option because it has more capacity than NiCd, lasting longer (the current peaks from the drivetrain are usually much lower than the ones from the weapon system, because wheel slip acts as a torque limiter).



In the configuration shown in the scheme, the NiMH pack is connected to a Hella switch, which powers the OSMC speed controllers. The NiCd pack is connected to a White-Rodgers 586 SPDT solenoid that activates the weapon, while a power resistor is used for braking. A small battery pack powers both the receiver and the RC interface. The RC interface interprets the PPM signal from the receiver, sending a low power PWM signal to each OSMC. Each OSMC amplifies the received signal, sending a high power PWM output to the drive motor it is connected to. The RC interface is used as well to trigger the solenoid that powers the weapon motor.

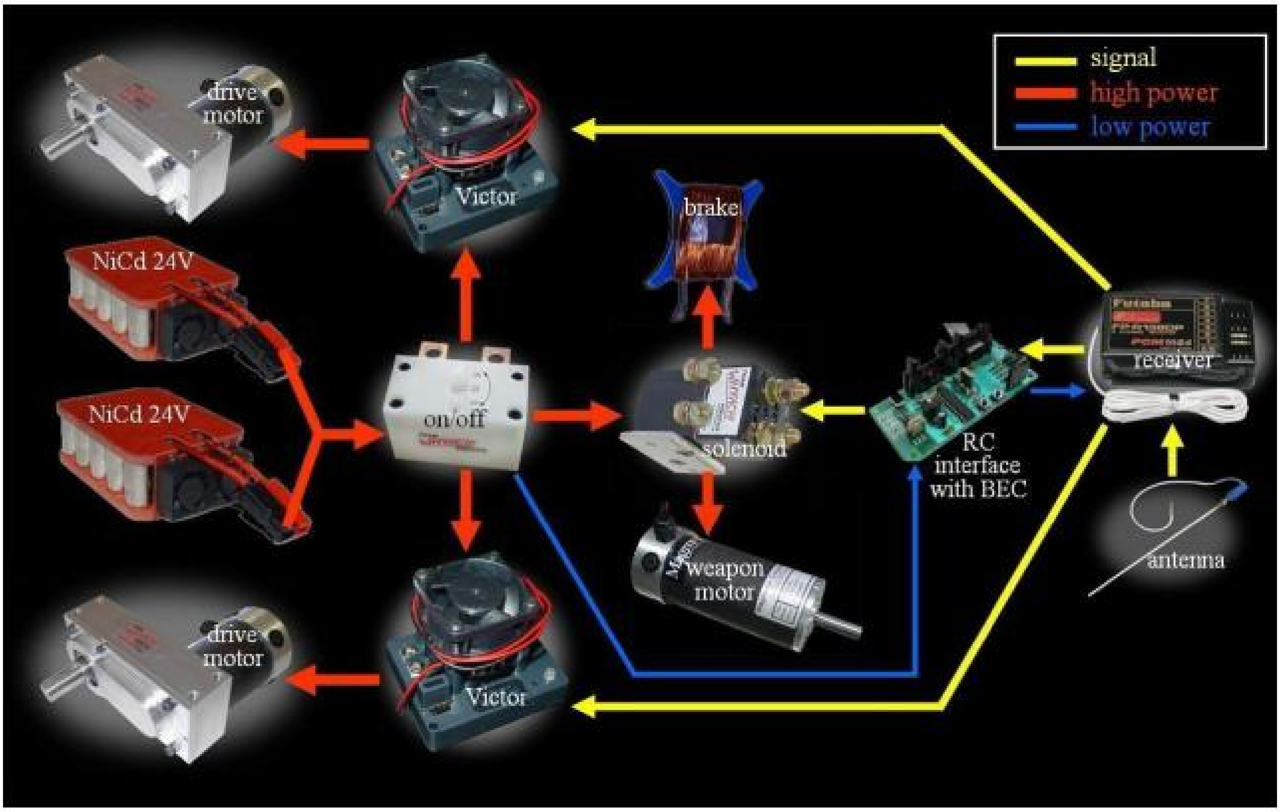
Note that this RC interface only works if powered by both 5V from the small battery pack and 12V from the OSMCs. Therefore, if the Hella power switch is off, the NiMH pack won't provide 24V to the OSMCs, which in turn won't provide 12V to the RC interface, which in turn won't be able to keep the solenoid active, turning the weapon off. In theory it would work, but it is unsafe.

The above scheme looks good, including the battery optimization feature: NiCd for the weapon, to deliver high peak currents, and NiMH for the drivetrain, for improved capacity. However, it has serious flaws:

- if the single NiMH pack breaks, the robot will stop working and, therefore, lose the match;
- if the small battery pack voltage is too low, the robot will become unresponsive;
- there isn't a power (on/off) switch between the NiCd pack and the weapon solenoid, thus if due to a surge current the solenoid terminals get soldered, the weapon won't stop, even with the main switch off – therefore this scheme won't pass safety inspection; you would need to include another switch between the NiCd pack and the solenoid;
- there isn't an on/off switch between the small pack and the receiver and RC interface, which is required to pass safety inspection; note that even the fans used in the robot must be turned off by the robot switch(es); therefore, the robot would need 3 on/off switches, because you can't connect the three battery packs in parallel due to their different types or voltages; the driver would need then to remember to turn on all three switches in the beginning of every match;
- the White-Rodgers 586 SPDT Solenoid is relatively large and heavy;
- OSMC speed controllers are not very compact, occupying a lot of the internal space;
- a resistor with both low-resistance and high-power, typically with less than 1Ω and more than 1kW for middleweights, needed to brake the weapon, isn't cheap and it can burn.

Improved Connection Scheme

To solve the problems presented above, you should use an improved scheme, such as the one pictured below. It includes 2 (or more) identical battery packs in parallel (NiCd in this example), connected to a single MS-2 power switch. A second power switch could be used in parallel to both packs, as a redundancy measure in case one of the switches breaks in the off position. Both packs need to be exactly the same, with same type, voltage and capacity, to be connected in parallel without any problems. This is why we use 2 identical NiCd packs. This switch powers the Victors, the TWC1 solenoid, and the RC interface. This interface, needed to activate the solenoid, has a built-in BEC to power the receiver. The Victors can be directly connected to the receiver without an RC interface. A copper wire reel is connected to the solenoid to act as a weapon brake. A Deans Base-Loaded Whip antenna is attached to the 75MHz receiver, enhancing reception quality.

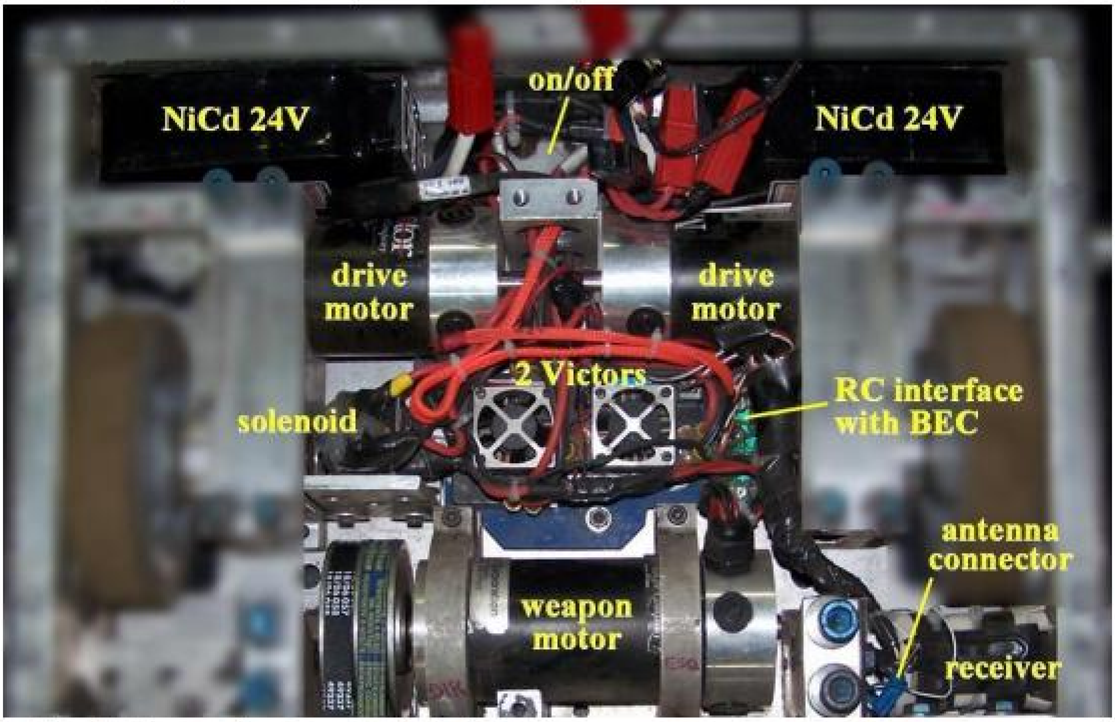


This improved scheme addresses all the issues from the classic scheme, because:

- if one of the batteries fails, due to a broken solder or connector malfunction, the robot will continue to fully function using the remaining pack(s), however with lower total capacity;
- the weapon will accelerate faster, since two packs in parallel can supply twice the current, assuming that the drivetrain isn't demanding too much from them during this acceleration; • there is no need to have a small battery pack for the receiver, due to the RC interface BEC; • a single on/off switch can power down the entire robot, including drivetrain and weapon motors, as well as the receiver and RC interface, which is required by safety inspections in the events; • the TW-C1 solenoid is smaller, lighter and cheaper than the White-Rodgers 586 SPDT;
- Victor speed controllers, besides smaller than the OSMC, can be directly connected to the receiver, without needing the RC interface (which is only used above to power the solenoid and to work as a BEC to power the receiver);
- Victors have a brake/coast jumper, used to set its action during a neutral condition from the radio; the brake setting sets the output to a short-circuit during neutral, while the coast setting sets an open circuit; when used in the drive system, the brake setting will stop your robot when you release the radio control stick, while the coast setting will let your robot continue moving due to its inertia; the brake setting is a good option for sharp turns in agile robots, while the coast setting is good to prevent the drive motor from overheating due to the short-circuits;
- the RC interface can become smaller, because it only needs to actuate the solenoid and to work as a BEC, without any need for PWM outputs for the drive system;

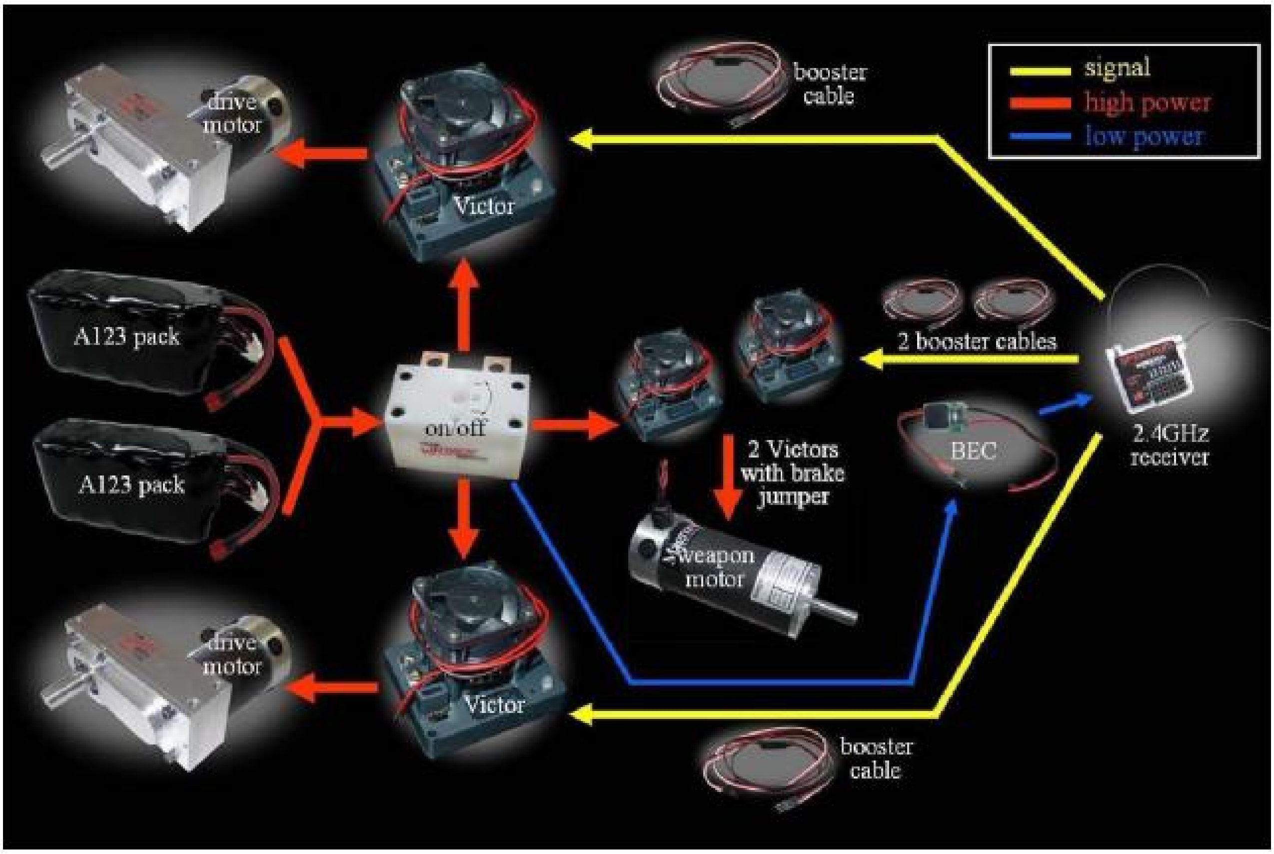
- the copper wire reel is cheaper and it dissipates heat better than the power resistor.

The presented scheme is also pictured below, showing a close up of the electronic system of the 2006 version of our middleweight Touro. Note that all components from this improved scheme are included below, except for the weapon brake, usually not needed in drumbots.



Connection Scheme for Reversible Weapons

After RoboGames 2008, we've decided to try a new connection configuration, one capable to reverse Touro's drum rotation and to solve minor issues from the previous schemes. To improve the battery capacity and voltage, the NiCd packs in parallel were replaced with A123 packs. Instead of using an RC interface and solenoid, two Victors are used to control the weapon motor, one for each pair of brushes. Using 2 ESCs to power the same DC motor is only possible if it has independent circuits for each pair of brushes, such as the Magmotors (which have 4 brushes).

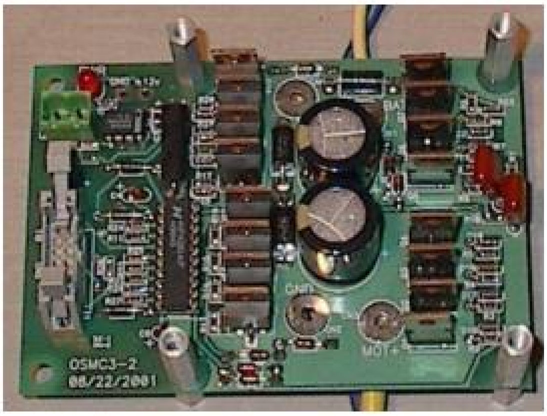


This scheme has improvements that are important for fully reversible spinners:

- the two Victors HV-36 powering the weapon allow its spin sense to be inverted if the robot is flipped, maximizing weapon effectiveness;
- one Victor for each pair of brushes from the weapon motor improves reliability, since if one of them fails the other will be able to spin the motor, although with less power;
- Victors weigh less than the TW-C1 solenoid, therefore their inertia is lower, reducing the risk of mechanical damage due to high impact accelerations;
- if the weapon is taking too long to brake, then simply set the jumpers from its Victors to the brake setting; otherwise, choose the coast setting to prevent the motor from overheating;
- Victors are less prone to lock-up than solenoids, therefore safety is also improved;
- a dedicated Universal BEC is used to power the receiver, instead of one integrated in an RC interface that also controls other devices, improving reliability; note that this BEC features a switching voltage regulator instead of a linear regulator, avoiding overheating problems.

Developing your Own Electronics

The speed controllers presented in this chapter and their radio-control interface (RC interface, a.k.a. logic interface, which interfaces the controllers with the receiver) are not cheap. But even so they are an off-the-shelf solution with very good cost-benefit, considering their complexity. Since most good quality electronic components needed by these systems are expensive, developing your own speed controller or assembling yourself an existing one doesn't save you too much money. For instance, a fully assembled and tested OSMC (pictured to the right) costs US$219 at www.robotpower.com, while its bare board (US$29) and components (about US$140) will set you back about US$169. You're basically paying US$50 for assembly and testing, which is quite reasonable, in special considering that you could burn out the entire controller if it's not carefully soldered.



But developing your own speed controller or RC interface, or even just assembling an OSMC (as pictured to the right), is a great learning experience. In addition, you'll be able to tailor the RC interface to your needs. For instance, Robot Power sells a very high-performance microcontroller-based closed-loop interface that controls two OSMCs, named Dalf. It is great for an autonomous robot, but it would be overkill for a radio-controlled combot, in special considering its US$250 price tag. The simpler MOB (Modular OSMC Brain) interface, or its more compact version μMOB, would be a much better option for robot combat, however both have been discontinued. This was the motivation for us to create a compact 3-channel RC interface to control two OSMCs (for the drivetrain motors) and one heavy duty relay or solenoid (such as the White-Rodgers 586 solenoid, for the weapon system), including a BEC to power the receiver, as explained in section 7.8.2.

Note, however, that bulletproof speed controllers and RC interfaces are not trivial to build. So, if you're planning to use your own system in combat, it is fundamental to perform several benchmark tests to avoid any surprises. There are a lot of things that can go wrong with these systems.

Speed Controller Development

Before we discuss the RC interface, it is important to understand the speed controller it will interface with. In this section, we'll introduce the main features of a typical high power speed controller, based on the OSMC design.

A typical speed controller is basically an H-Bridge (introduced in section 7.2.3) used to power DC motors with a controllable voltage. To do that, the H-Bridge uses one or more transistors in parallel at each of its four legs.

To activate the H-Bridge, do not use a Bipolar Junction Transistor (BJT), it is not efficient when dealing with the high electric currents needed in combat. Instead, use a MOSFET (Metal-Oxide Semiconductor Field-Effect Transistor, a.k.a. FET), such as the IRF1405 used in our OSMC boards. It has several advantages, despite its relatively high cost.

The first advantage of FETs is that they are voltage-activated (instead of current-activated such as in a BJT), making it easy to activate them. It is enough to guarantee that its input voltage (at the gate) is higher than its threshold voltage V_{th}, to allow the high currents to go through the drain and source.

When the FET is activated, in saturation mode, it behaves as a resistor, with resistance R_{on}. Very good quality FETs can have R_{on} as low as 5mΩ. To continuously supply, for instance, 160A to a motor, you will need more than one FET at each leg of the H-Bridge. If 4 FETs are used in parallel at each leg (totaling 4 × 4 = 16 FETs for all 4 legs from the H-Bridge), then the power dissipated by each FET is about

$$P = I^2 \cdot R_{on} = \left(\frac{160A}{4}\right)^2 \cdot 5m\Omega = 8W$$

which is an acceptable value for use with small heatsinks coupled with a fan to actively cool down the FETs. If the heatsinks were not used, then the maximum continuous current acceptable going through a system with 4 FETs in parallel would be approximately 100A. Another great advantage of the FETs is that they don't have any current limitation, as long as their maximum temperature is not exceeded. Therefore, FETs can easily take very high current peaks, as long as they are brief enough not to overheat them. And the FET commutation usually takes only a few dozen nanoseconds, keeping low the energy losses from this process.

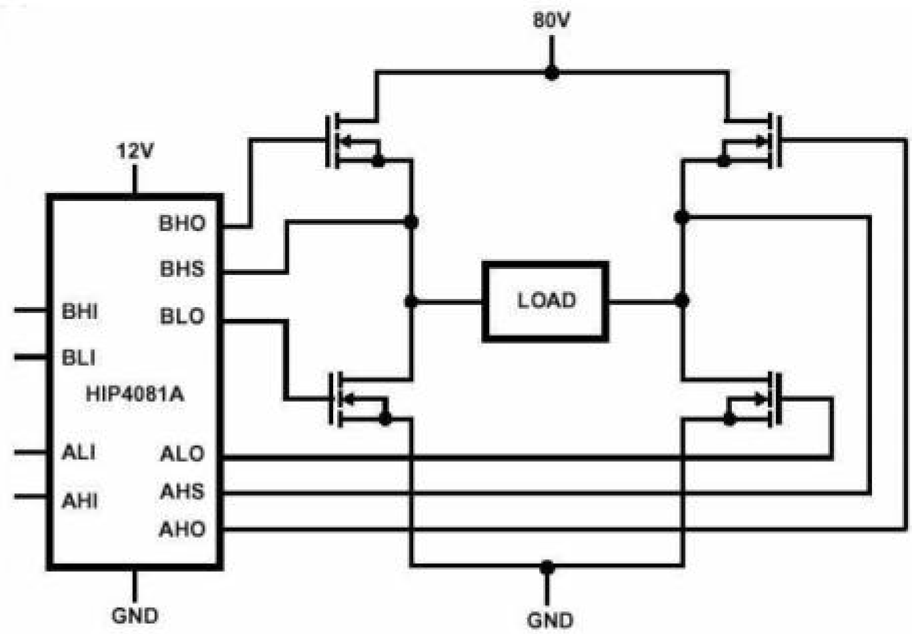
The fact that FETs are activated by voltage helps a lot in the development of an activation circuit. But there's a catch that can cause a few problems. For the FET to conduct (in saturation mode), an electric charge must be injected at the gate of the FET to make the voltage between the gate and the source reach approximately 10V. This 10V is, in general, the voltage required to completely enter saturation mode, minimizing the value of the resistance R_{on}. Such need to charge the FET, called parasitic capacitance effect, can be modeled as a capacitor in parallel with the gate of the FET.

To charge this large capacitance, the integrated circuit HIP4081A can be used. It is a high frequency H-Bridge driver, capable of supplying up to 2A for the four FETs connected in parallel to each output. To avoid shoot-through, which could happen for instance if an upper FET turns on while a lower FET from the same side of the H-Bridge is still conducting, a resistor is connected in series with the gate of each FET, limiting the total current and making the FETs take longer to be activated.

In this way, the resistors help to balance the T_{on} and T_{off} times from all FETs in parallel, by equalizing their resistor-capacitor constants.

Despite the presence of the resistors, there would still be a chance of happening a shoot-through. Two protection measures exist to avoid this condition. The first is a programmable time in the HIP4081A when both FETs get turned off (in cut-off mode). The second is the addition of extremely fast diodes in parallel with the resistors, so that during the T_{off} time of the PWM the entire current is drained by them, eliminating any chance of happening a shoot-through in the circuit.

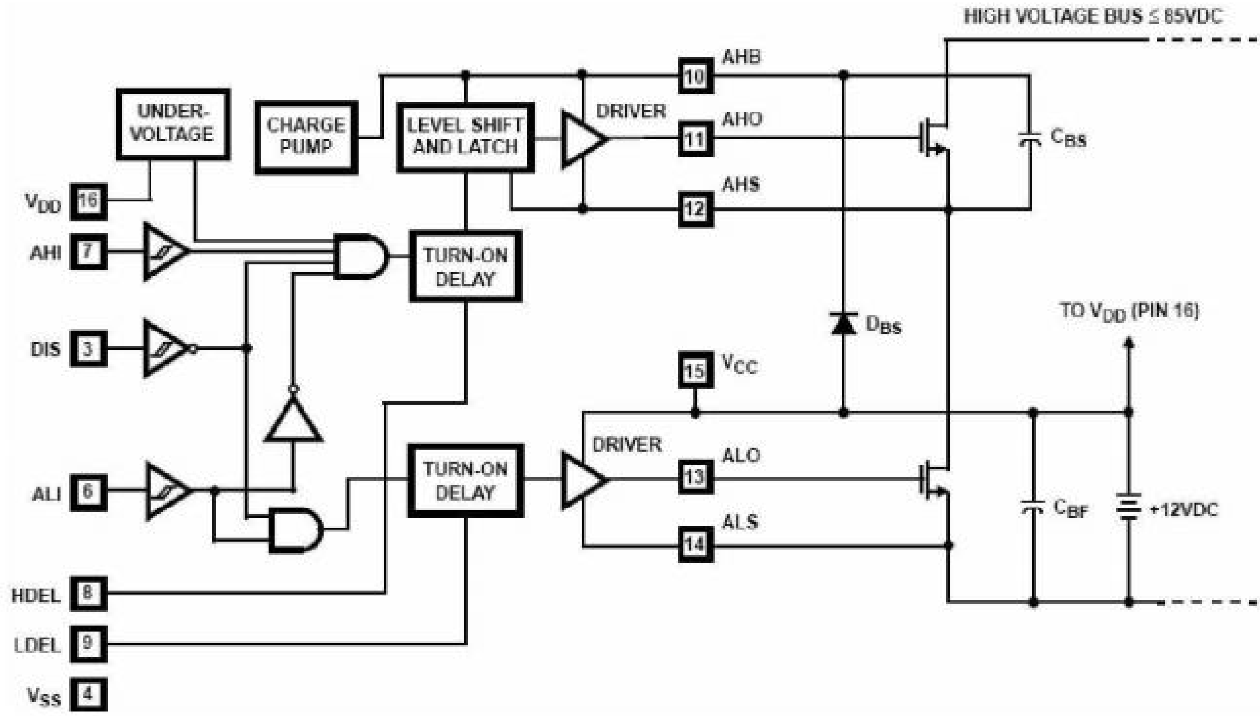
The HIP4081A has, therefore, the function to activate the upper and lower FETs, including a circuit to increase the voltage to the levels required by the FETs. In the application example pictured to the right, the HIP4081A is powered by 12V (allowable values are between 9.5V and 15V), while a battery voltage of 80V (or any other value between 12V and 80V) is applied to the load.



The load in this example can be, for instance, a brushed DC motor. If the voltage supplied to the HIP4081A is lower than 9.5V, then an internal protection turns off the upper FETs. On the other hand, if such voltage is higher than 16V, the HIP4081A can be damaged. In addition, to protect the FETs against voltage peaks, two Zener diodes are used to limit their voltage to 15V.

The HIP4081A has four digital inputs, AHI, ALI, BHI and BLI, each one corresponding to the outputs that power each group of FETs from a leg, respectively AHO, ALO, BHO and BLO, as pictured above. In other words, when an input is enabled, the FETs connected to the corresponding output are activated. The RC interface, which in the case of the OSMCs is an external electronic system (not an integrated one as in Victors), needs to send PWM and direction signals to the HIP4081A to control the H-Bridge. These digital signals are compatible with the TTL logic, but any input voltage above 3V, such as 5V or 12V, is recognized as a high (“1”) logic state.

Also, the HIP4081A has a protection in its internal logic against shoot-through, which shuts down the upper FETs connected to AHO (or BHO) when the lower FETs from the same side of the bridge, connected to ALO (or BLO), are activated, independently of the state of the upper inputs AHI and BHI. This protection is implemented using AND logic gates in all the HIP4081A inputs, as seen in the picture on the next page, which shows the functional diagram of half of a HIP4081A driver. The AND gates have as input the values of AHI, ALI, BHI and BLI, in addition to the complement from the DIS (Disable) pin, deactivating all FETs if DIS has a low logic level.



Each upper gate also features two other inputs. One of them is used for protection against low supply voltages, while the other is the complement of the lower gates, to guarantee that the upper gate output AHO (or BHO) will be turned off if the lower gate inputs ALI (or BLI), from the same side, are activated.

Resistors are also connected to the HIP4081A, between the input and ground pins. These resistors guarantee that all FETs will be turned off if no RC interface is connected to the power board. This is an additional protection to make sure that the motors will be turned off if the connection with the RC interface is lost.

Due to the nature of the used FETs, the gate voltage should be approximately 10V higher than the battery voltage to activate the upper FETs. To generate such higher voltage, the HIP4081A has a charge-pump system that, with the aid of a diode and an external capacitor, generates the necessary voltage at the outputs AHO and BHO, making it possible to activate the FETs connected to them.

To protect the circuit from voltage peaks caused by the DC motor brushes and commutators, a Transient Voltage Suppressor (TVS) is used. It works exactly as a Zener diode, in other words, when the voltage on the TVS is above a specified value, it starts conducting, “absorbing” the excess voltage. The TVS is optimized to tolerate voltage peaks with high currents. It is used to absorb the voltage peaks between the battery terminals, and to protect the FETs.

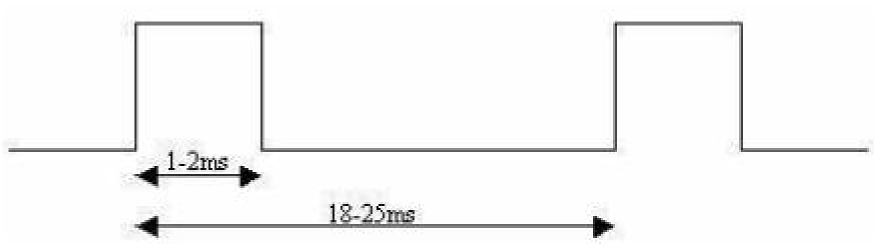
In addition to the TVS, resistor-capacitor circuits between the motor terminals provide an additional protection against high frequency peaks generated by the brushes. Also, large electrolytic capacitors are placed as close as possible to the H-Bridge to reduce the effects caused by the inductances of the wires that connect the battery to the circuit.

The last part of the circuit comprises the switched-mode power supply, which converts the battery voltage (between 12V and 80V) down to 12V, using a high efficiency regulator that does not need a heatsink. These 12V will also be used to power the RC interface, described next.

RC Interface Development

The power circuit discussed above cannot receive directly the signals from a radio-control (RC) receiver. These signals need to be treated first. This signal conditioning is made through a RC interface circuit, which is an interface between a receiver and a speed controller (or solenoid).

There are several off-the-shelf RC interfaces that you can buy, for instance, at www.robotmarketplace.com. But if you want to build one yourself, then you'll probably need to use a micro-controller, such as a PIC, dsPIC or AVR, capable of executing several million instructions per second. The RC interface that we've developed uses a PIC, capable of decoding the signals from up to four receiver channels, to use them to command power circuits, solenoids or relays. The input signal in the RC interface comes from the receiver, it is a pulse train following the PPM standard, as pictured to the right. This pulse train has a period that can vary between 18 an d 25ms, with each pulse lasting between 1ms (low) and 2ms (high).



As mentioned before, PPM and PWM are two completely different signals, even though they are both pulses. In our application, PPM is a low power pulse train from the receiver that carries the commands from several channels through a code that is based on the absolute width of each pulse, bringing not only the information about the desired speed of several motors, but also their direction. PWM, on the other hand, as explained in section 7.2.2, is a pulsed signal that carries, in our application, the information about the absolute speed (but not the direction) of a single motor, determined from the *ratio* between the periods T_{on} and T (and not from the absolute value of T_{on}). So, the job of our RC interface is to take the single PPM signal from the receiver, decode it, and send one PWM and one directionality signal to the power board of each motor.

When, for instance, a stick of the radio control is completely to the left, the PPM pulse from the associated channel has a 1ms width; if the stick is in the middle, then the width is 1.5ms; and if the stick is moved completely to the right, the pulse will take 2ms. All other stick positions will translate to a pulse width between 1ms and 2ms. The pulse width is either directly or exponentially proportional to the stick position, depending on the radio settings.

Therefore, to control a bi-directional motor, the 1ms pulse is usually associated with a command to move back at full speed, while a 2ms pulse would mean move forward at full speed, and a 1.5ms pulse means that the motor should stop. Also, for instance, a 1.9ms pulse would mean that we want to go forward with $2 \times (1.9 - 1.5) = 0.80 = 80\%$ of the top speed, while a 1.2ms pulse would mean that we want to go back at 60% of the top speed, because $2 \times (1.2 - 1.5) = -0.60 = -60\%$.

We've programmed our PIC to validate the PPM signal from the receiver, and then to count the width (time interval) of each pulse. Clearly, each pulse is associated with one receiver channel.

There are five output signals that need to be sent to the HIP4081A driver: AHI, ALI, BHI, BLI, and the DIS (Disable) signal. The DIS signal is only used in case you want to turn off the H-Bridge. Due to the protection against shoot-through in the HIP4081A, it is possible to simplify the involved logic, keeping both AHI and BHI signals in the high logic level, all the time. The suggested signals for the correct operation of the power circuit are shown in the table below.

AHI	BHI	ALI	BLI	DIS	Function
1	1	0	PWM	0	Forward
1	1	PWM	0	0	Back
1	1	0	0	0	Brake
1	1	1	1	0	Brake
×	×	×	×	1	Off

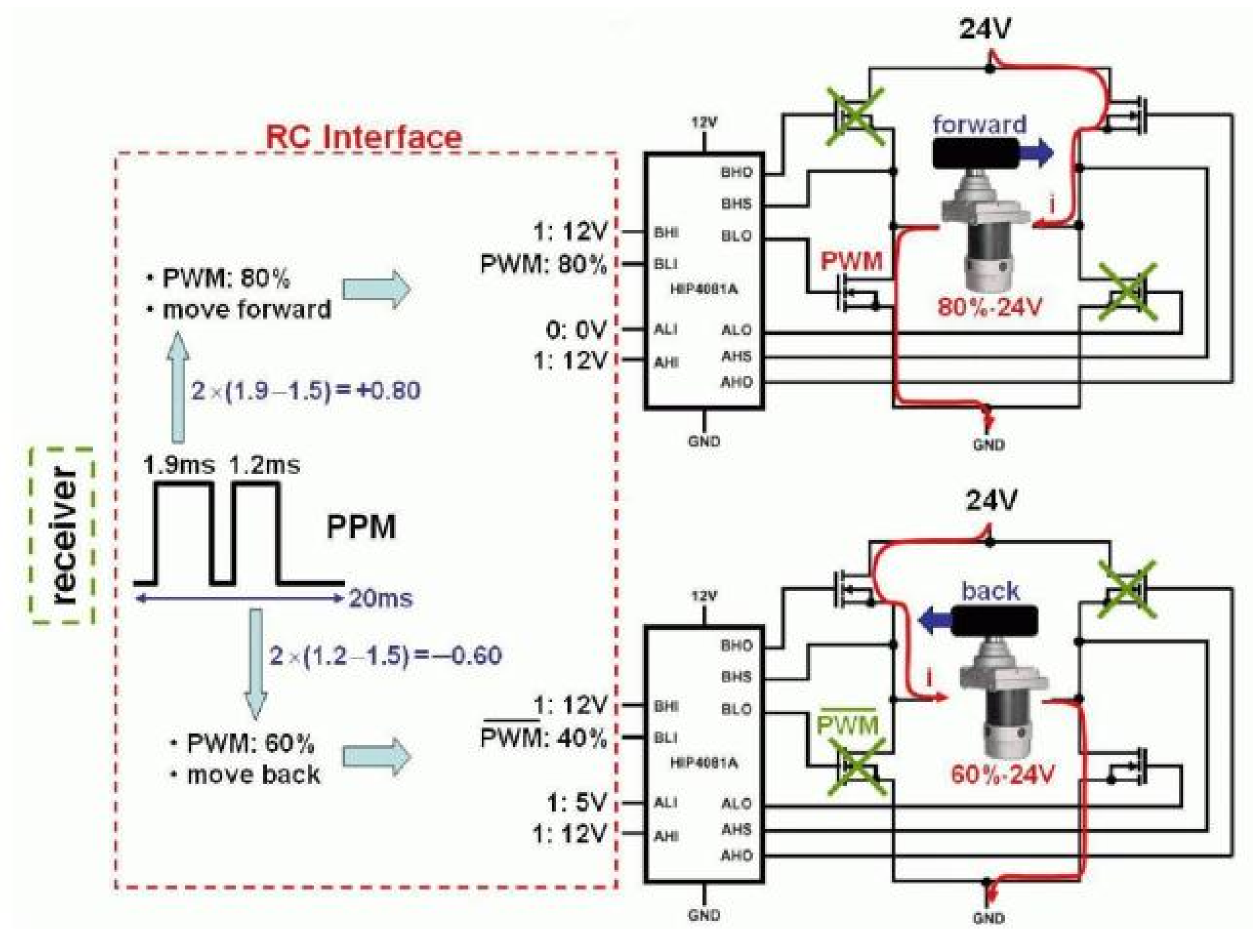
×: the state doesn't matter; 1: means 5V (or 12V); 0: means 0V;

So, we only need to deal with two signals, ALI and BLI, because AHI and BHI are always kept at the high ("1") logic level. But both ALI and BLI deal with the PWM signal, as seen in the table above, which is not good because you would need to use two PWM output pins from the PIC to control a single motor. We need to modify the table above such that only one signal takes care of the PWM (for instance, the BLI, which can be used to define the absolute speed of the controlled motor) while the other defines the direction of the movement (the ALI signal, in this example). In this way, only one PWM output pin from the PIC will need to be used per motor (to carry the BLI signal, in this example). The modification is shown in the table below.

AHI	BHI	ALI	BLI	DIS	Function
1	1	0	PWM	0	Forward
1	1	1	PWM	0	Back
1	1	0	0	0	Brake
1	1	1	1	0	Brake
×	×	×	×	1	Off

×: the state doesn't matter; 1: means 5V (or 12V); 0: means 0V; With this new table, the BLI will be the PWM signal, while the ALI will control the direction in such a way that a low ("0") logic level means forward, and a high ("1") logic level means backward. So, when moving forward, the current goes through the FETs connected to the HIP4081A outputs AHO and BLO, and when moving backward the current goes through the FETs at BHO and ALO. However, when moving back~~, the P~~WM that goes to BLI must be inverted, either

through software or hardware, resulting in the PWM signal as show n in the table.



The figure above shows an example of a PPM pulse train with 20ms period, generated by a receiver, used to control two 24V permanent magnet brushed DC motors. The RC interface uses its PIC controller to measure the pulse widths, resulting in 1.9ms for the first motor (the top motor in the figure) and 1.2ms for the second. Then, the RC interface figures out, from the calculations shown in the figure, that the first motor has been commanded to move forward with 80% of the provided 24V, while the second motor needs to move back with 60% of 24V.

For the first motor, the RC interface sets the ALI voltage to 0V (low level "0") to indicate that the motor should move forward, and the BLI receives a PWM signal that is high ("1") during 80% of the time. Note that the BHI and AHI are always at a high ("1") logic level, by default, which can be obtained using 12V hard-wired from the HIP4081A 12V pin (as discussed before, any voltage above 3V translates to "1" in the HIP4081A). With ALI at 0V, the FETs connected to ALO do not conduct.

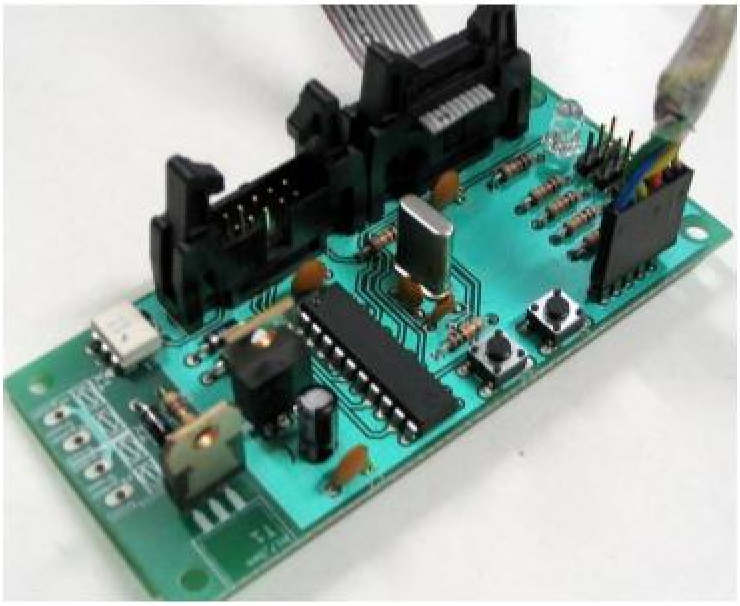
So, the current from the 24V battery has to flow to the first motor through AHO (which conducts because AHI is always set to "1") and BLO. But BLO only conducts 80% of the time, because of the 80% PWM signal at BLI, so the resulting motor voltage will be, in average, about 80% of the 24V input voltage, as desired.

For the second motor, ALI is set to 5V (at the high logic level "1") to indicate that the motor should move backward, whi le BHI and AHI are always at 12V (also at the high logic level "1"), by default. But the RC interface, instead of sending the desired 60% PWM signal to BLI, sends the inverse signal PWM , which is low (0V, instead of high) during 60% of the time. Therefore, this inverted signal is high (logic level "1") during only 40% of the time. With ALI set to the high logic level, the FETs at ALO will always conduct. Even though AHI is at the high logic level, the FETs at AHO do not conduct because of the shoot-through protection that prevents a short-circuit of the 24V battery through AHO and ALO.

During 60% of the time, when BLI is low ("0"), the FETs at BLO will not conduct, and the current from the 24V battery will flow to the second motor through BHO (which conducts because BHI is always set to "1") and ALO. But during the remaining 40% of the time, when BLI is high ("1") and hence the BLO FETs conduct, the BHO FETs will stop conducting due to the shootthrough protection that prevents a short-circuit through BHO and BLO. Without the BHO FETs to conduct, the second motor won't be powered during 40% of the time. So, the resulting motor voltage will be, in average, about 60% of the 24V input voltage, while moving back, as desired.

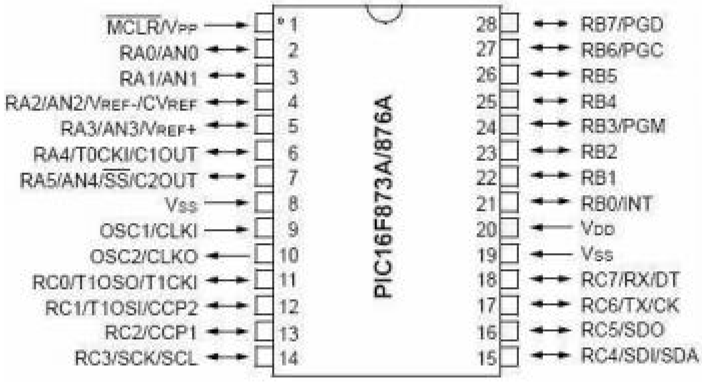
In summary, the PWM signal must be inverted at BLI to move back because we make use of the shoot-through protection, which only allows the motor to be powered when BLO is not conducting.

The hardware of the developed RC interface, pictured to the right, is relatively simple and compact in size, measuring 4" × 1.75". It includes a microcontroller PIC16F876A, and a buffer to isolate the signals generated by the PIC from the power board signals, to avoid any problems.



The interface board also features an independent circuit used to activate a high power relay or solenoid, which is completely isolated with the aid of an optocoupler.

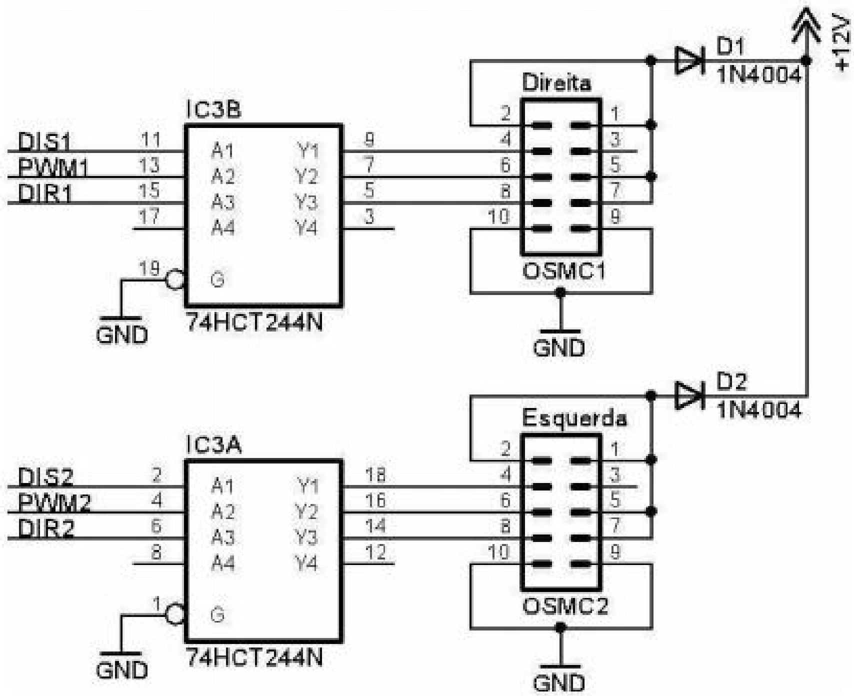
In addition, the developed board includes a BEC that takes the 12V from the power board (used in the HIP4081A driver) and converts it into 5V to power both the receiver and the RC interface itself, using a linear regulator. The developed board also includes two buttons, one to reset the PIC and the other to enter into calibration mode.



The micro-controller PIC16F876A (pictured to the right) features in-circuit serial programming, which allows it to be programmed without the need to remove it from the board. The input signals from the receiver are connected through resistors to the pins RB4 through RB7 from the PIC.

.

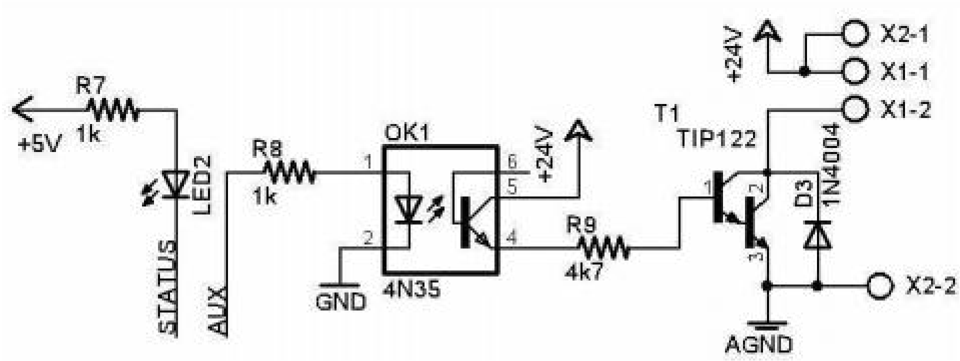
The buffer used in both PWM outputs from the RC interface is the chip 74HCT244, consisting of two sets of four buffers each. It is possible to use as well other chips equivalent to 74HCT244, such as the 74HC244, as long as their output voltages are high enough to be used with the HIP4081A from the power board. For instance, the 74LS244 chip is not recommended in this case, because it associates any voltage beyond 2V to a high logic level, while the HIP4081A requires a minimum value of 2.5V.



The figure to the right shows two connectors, named OSMC1 and OSMC2, which are used to connect the developed RC interface to two OSMC speed controllers. The output signals to AHI and BHI (pins 5 and 7, respectively) need to be permanently set at the high logic level "1" as discussed before. This is accomplished with any voltage higher than 3V, not necessarily 5V, so in our case we've connected these pins 5 and 7 to the pins 1 and 2, which provide the 12V supplied by the HIP4081A.

Note also in the figure above that there are two diodes D1 and D2 between the RC interface and both power boards. These diodes are a redundancy measure to ensure that the RC interface, which is powered by the 12V lines from both power boards, will still be functional even if one of the power boards burns out.

The developed RC interface is also able to activate a relay (or a solenoid), usually used in the weapon system, including as well a status LED, as pictured to the right. This circuit uses an optocoupler that, when enabled, makes the T1 transistor conduct, activating the relay. To do that, the relay terminals must be positioned at X1-1 and X1-2, while 24V should be applied to X2-1 and X2-2 (assuming a 24V relay). The transistor T1 can handle up to 3.5A with a heatsink, or 1.0A without one. The transistors TIP120, TIP121 or TIP122 can be used in this circuit. The figure also shows the status LED, which is used to inform the state of the program running in the PIC. In this case, the LED is used to indicate whether the circuit is under normal mode or calibration mode.

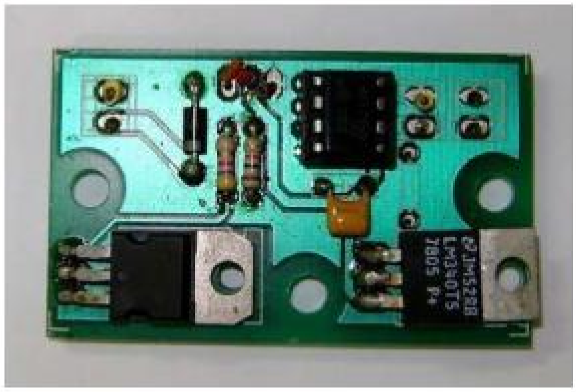


The software used by the PIC16F876A from the RC interface was written in the programming language C. The entire program, together with more details about our RC interface board, can be found in the undergraduate thesis (in Portuguese) from the RioBotz team member and former advisee *Felipe Maimon*, which can be downloaded at www.riobotz.com.br/en/tutorial.html. I've tried to summarize and translate to English the main points from this thesis here, in section 7.8. Note that the program, which is relatively lengthy, is very specific to the hardware of the developed RC interface. Our RC interface is nicknamed MOB, in honor of the discontinued "Modular OSMC Brain" interface, however here it stands for "*Maimon*'s OSMC Board."

Our RC interface board was successfully used in all our OSMC-powered middleweight combots: the overhead thwackbot Anubis (controlling the speed of both NPC T74 drive motors), as well as the horizontal bar spinners Ciclone (controlling the speed of two DeWalt gearmotors and activating its Etek weapon motor through a White-Rodgers 586 SPDT solenoid) and Titan (controlling 4 Magmotors S28-150, two of them for the drive system using PWM, and the other two for the weapon through a single TW-C1 solenoid). The board withstood well the rigors of combat.

We haven't been using our RC interface board described above since we migrated from OSMCs to Victors in all our middleweights. Victors don't handle as much current as the OSMCs do, but they're more compact and they can be connected directly to the receiver without an RC interface.

However, we still needed an RC interface to activate the solenoid from the weapon system of our middleweights. We've then designed another more compact RC interface, measuring 2" × 1.25" (pictured to the right), without the PWM outputs, featuring 1 output for a high power relay or solenoid and a BEC to power the receiver. It was used until 2008 by Touro and Titan to power the TW-C1 solenoid from their weapon systems.



Finally, we've considered designing a third RC interface board, featuring 2 solenoid outputs controlled by independent channels. Having two solenoid outputs, instead of one, would be useful to power a single weapon (such as a drum) in both directions using two SPDT solenoids arranged in the bang-bang configuration shown in section 7.2.1 (or it could be used in combots with more than one weapon). However, the added weight and volume of 2 solenoids, in addition to the risk of shorting out the battery if both of them are accidentally switched to a "shoot-through" configuration, made us choose instead to use Victors to power the weapon in both directions, as explained in section 7.7.3.