



Modular OSMC Brain v1.3

Supporting documentation for the OSMC project
Manual revision 1.3

Written in part by:

Robert L. Doerr – rdoerr@home.com – <http://www.robotworkshop.com>

Chris Baron - chris@robot-power.com - <http://www.robot-power.com>

and from many helpful additions from the
crew at the site listed below

Development archive & ongoing support can be found at:

<http://groups.yahoo.com/group/osmc>

This documentation is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See page 2 for more details.

Open source promotes software reliability and quality by supporting independent peer review and rapid evolution of source code.

Table of Contents

GNU GENERAL PUBLIC LICENSE	2
INTRODUCTION	3
ASSEMBLY INSTRUCTIONS.....	3
PROGRAMMING	6
INSTALLATION	7
INPUTS	8
JUMPER SETTINGS.....	9
LED INDICATORS	10
OPERATION	10
MOB SOFTWARE OPTIONS.....	11
SETTING SOFTWARE OPTIONS.....	11
HARDWARE SPECIFICATIONS	12
IN CASE OF DIFFICULTY	13
CIRCUIT DESCRIPTION.....	13
REPLACEMENT PARTS LIST	14
REVISION HISTORY.....	14
SPECIAL THANKS	14

MOB – Modular OSMC Brain

GNU General Public License

The full license can be found in the README.txt file that is included in the OSMC package. It can also be found at: <http://www.opensource.org/licenses/gpl-license.html>

Please note that the GNU General Public License covers the entire OSMC project excluding the following (which has its own different license):

- AVRx Kernel
- avrx_tasking.s
- MOB & uMOB software

For licensing information on these topics please email Larry Barello at larry@barello.net

Please note that joining the OSMC discussion forum requires you to sign-in using your own yahoo user ID and password. Yahoo has its own Privacy Policy & Terms of Service regarding this issue which can be found from links on the group's home page: <http://groups.yahoo.com/>

MOB – Modular OSMC Brain

INTRODUCTION

The Modular OSMC Brain (MOB) is a full-featured controller for two Open Source Motor Controller (OSMC) power H-bridge boards. The MOB takes the signals from a standard hobby Radio Control receiver and generates the proper control signals to drive a pair of OSMC boards. It is very flexible and can directly control each channel for two-stick operation or it can mix two channels for single stick control. It also has a third auxiliary channel that controls a pair of opto-isolators for switching relays or other devices. It supports 2, 4, 16, and 32 khz PWM selection, an adjustable deadband, I2C interface, serial interface, and has spare Analog I/O channels for expansion such as current or velocity sensing. It will support locked-antiphase or sign-magnitude PWM by user selection. The user interface is primarily through a 20x4 LCD display and 4 command buttons.

At the core of the controller is the powerful Atmel AVR ATmega163 processor running at 8 MHz. It is programmed in GNU C for the AVR and uses Larry Barello's (www.barello.net) real-time operating system AvrX. It can be programmed in circuit so new features, software updates and enhancements can easily be installed. All software for the MOB is freely available under the GNU Public License or other free license. The MOB is designed to be flexible and useful for a large number of applications.

This manual is meant to be your guide to help with the assembly of this great board.

NOTE: Although the software is currently very functional it is changing often to add more features. The most recent version should be posted in the files section of the OSMC group on yahoo. Once you set the options the way you like you can save the settings and never have to worry about them again.

If the MOB is already assembled, please proceed to the Programming and Installation section.

ASSEMBLY Instructions

This board should be built using standard construction methods. The following items are required to build the kit: diagonal cutter, needle nose pliers, tweezers, fine tip soldering iron (pencil type), 60/40 Rosin core solder, liquid flux, solder wick, and some patience. During the Assembly process, keep in mind that the Component side of the board has the silkscreen (white lettering). All the components will be mounted on the Component side of the board. For the following steps, the LCD connector (J7) should be up and the 20-pin header for the two OSMC boards (J8) should be on the left. After installing each component at the specified location solder it in place before proceeding to the next.

NOTE: When installing each of the surface mount capacitors and resistors put a small amount of solder on one pad. Next, warm the solder on the pad and position the part with a pair of tweezers. To make sure the part is seated you can gently press down on it while warming the solder pad with the soldering iron. If the part shifts or does not line up properly then you can warm it up again and reposition the part with the tweezers. Once the part is in place solder the other side. If needed re-solder the first pad again using some fresh solder. Only a very little solder is needed on these small parts.

- Install the .47uf capacitor (SMD) at location C1. Polarity does not matter.
- Install the 33uf Tantalum capacitor (SMD) at location C2. The band should point toward the right of the board toward the + sign. Some people have replaced this part with a standard through hole component by trimming and forming the leads so they can be soldered to the SMD pads.
- Install ten .1uf capacitors at locations C3 through C12. Polarity does not matter.
- Install ten 10K ohm resistors (SMD) at locations R11 through R18, R21, and R24. Polarity does not matter.

MOB – Modular OSMC Brain

- Install six 680 ohm resistors (SMD) at locations R19, R20, R22, and R25 through R27. Polarity does not matter. These are the current limiting resistors for the LED's and the opto-isolators.
- Install ten 1K ohm resistors (SMD) at locations R1 through R10. Polarity does not matter.

NOTE: Installing the CPU is probably the most difficult step in the assembly of this board. Carefully position the chip then solder two opposing leads to tack it down. If needed warm each connection and move the chip until it is perfectly aligned. Next solder each of the pins allowing enough time in between for cooling. Although the leads are extremely close don't worry if there are solder bridges. Once done put a dab of liquid flux on the leads around the chip. The last step is to slide de-soldering wick across the sides of the chip with the soldering iron. It will remove any solder bridges and excess solder. The result is a perfect solder connection that looks like it was done on a machine. You can use a similar process on the other two surface mount chips.

- Install the Atmega163 CPU at location U2. When positioned properly the writing on the CPU will be upside down in relation to all the writing silk-screened on the board. The small indentation on the chip indicates pin 1. This indentation should be at the same corner as the "cut corner" on the silkscreen outline (lower right in this case).

NOTE: On the through hole parts pin 1, or the positive lead for two pin devices, is usually designated by a square hole on the board. When installing DIP components U5, U6 and the Surface mount IC's at U3, U4 it works best if each of the two opposing corners is soldered first. Double check the orientation of each chip then, gently press on the center of the component while warming the solder on those corner pins with the soldering iron to make sure the component is properly seated. Finally solder the rest of the leads on the part.

- Install the 74ACT158 at location U3. Make sure pin 1 is toward the left. When positioned properly the writing on the chip will be in the same orientation as the silk-screen on the board. A small dip in the center of the silkscreen outline at one end shows the pin 1 side of the chip. All dual in-line IC silkscreen outlines will follow this convention. Also a small indentation or band on one end of the chip shows the pin one location.
- Install the 74HCT244 at location U4. Make sure pin 1 is toward the right. When positioned properly the writing on the chip will be upside down in relation to the silk-screen on the board. Observe the markings on the chip and the silkscreen outline.
- Install two 4N26 opto-isolators at locations U5 and U6. Make sure pin 1 is toward the top of the board. These are through-hole ICs but again pin one is indicated by an indentation in the package and a dip in the silkscreen outline for the part.

NOTE: After installing each through hole component on the board solder each lead and trim them flush with the back of the board.

HINT: You can spread the leads of through hole parts slightly to hold them in place while you flip the board over to solder the leads from the bottom side of the board.

- Install two UF1002 diodes at locations D5 and D6. On D5 the banded end should be up and for D6 the banded end should be on the right side. The silkscreen shows which side to put the banded end of the part on.
- Install the 1N5231B 5v zener at location D4. The banded end should be up.
- Install the 8 Mhz ceramic resonator at location X1. This can be inserted in either direction since polarity does not matter.

MOB – Modular OSMC Brain

- Install the 5v regulator at location U1. This does not require an insulator between the component and the board. You can optionally use a 6-32 screw and nut to securely fasten the U1 to the board.
- Install the 2K potentiometer at location R23. This is used to control the contrast on the LCD.
- Install five switches at the locations for LEFT, RIGHT, UP, DOWN, and RESET. Note, if you plan to use a remote LCD pendant with this board you don't have to install the UP, DOWN, LEFT and RIGHT switches on the board. The pendant will still work with them installed so they do no harm in either case.
- Install the RED LED at location marked POWER (D1). Make sure the shorter lead goes in the hole on the left. There is also usually a flat spot or notch on the LED case on the same side.
- Install a GREEN LED at location D2. Make sure the shorter lead goes in the hole on the top.
- Install the remaining two GREEN LED's at the locations marked Right Disable (D8) and Left Disable (D7). Make sure the shorter leads go in the hole on the right.

NOTE: When installing the headers insert the shorter leads into the PCB. The longer lead is where the connectors and configuration jumpers will go.

- Install the 14-pin (1x14) header at location LCD (J7)
- Install two 4-pin (1x4) headers at the locations SERIAL and AUX OUT (J10).
- Install four 3-pin (1x3) headers at locations I2C (J6), THRO (J1), DIR (J2), and AUX (J3).
- Install two 2-pin (1x2) headers at locations JP1 and JP2. These are for standard jumpers.
- Install the 10-pin (2x5) connector at location ISP (J5). Pin 1 will be on the top right. If installing a protected header then trim some material off the lower right corner to clear R12. This will ensure the connector is level with the board.

NOTE: Before installing the 20-pin connector for the OSMC boards make sure the two locking tabs are installed. It is much easier to press in the pins for the locking tabs ahead of time.

- Install the 20-pin (2x10) connector at location J8. Pin 1 will be on the top left of the board. The polarity slot in the center of the shroud should be toward the center of the board not the outside edge.
- OPTIONAL:* Fill in the via's with solder. These are the holes that connect top traces to traces on the other side of the board. Over time and from expansion/contraction of the board material (Not to mention flexing of the board) small cracks can develop in the via's. In order to prevent this from happening it is best to fill them in with solder. It might be necessary to heat both sides of the via in order to get a clean connection.
- OPTIONAL:* De-flux and clean the board. This step is highly recommend. When properly cleaned the board's appearance is better and it is easier to spot cold solder joints and solder bridges. Depending upon the type of flux used, this can be done economically using common rubbing alcohol and an old toothbrush.

MOB – Modular OSMC Brain

This concludes the Assembly procedures for the MOB. Congratulations! Before proceeding look over the board and verify the correct location and orientation of all parts. Also check to make sure there are no solder bridges or cold solder joints. Some of the traces are very close together, and it is easy to accidentally create a solder bridge across a trace or two. Also it is not uncommon to have an open connection or two on the Atmega163. Use a meter between the pin and the trace on the PCB to verify that each pin is connected.

PROGRAMMING

The CPU chip used in the MOB is the Atmel ATmega163. This is a high-performance 8-bit RISC processor running at 8 MHz. The RISC architecture of the Mega163 allows it to achieve nearly 8 MIPS of processing speed at very low power consumption. The Mega163 includes 16 KB of FLASH, 1 KB of RAM, and 512 Bytes of EEPROM. One of the most attractive features of the Mega163 is that it can be programmed in-circuit using simple low-cost hardware. The MOB comes with a standard 10-pin 2x5 programming header for programming the Mega163. This header is labeled J5 and is located near the LCD connector on the board. The format of this header matches the STK200/STK300 standard supported by most programmers and software that support the ATmega family.

There are many different hardware and software programmers available for the ATmega family. One of the most popular and best supported is PonyProg published by Claudio Lanconelli. The latest version of the software may be downloaded for free from <http://www.lancos.com/prog.html>. PonyProg works with Windows 95 or higher and Linux. See the PonyProg Web site for details on the system requirements.

...Add PonyProg step by step here...

- Setup/calibration
- setting device type
- connecting hardware
- setting fuses
- programming FLASH
- programming EEPROM

The Atmel Atmega163 parts come pre-programmed to use the internal RC clock which runs about 1 MHz. Since the MOB board supplies an external 8 MHz ceramic oscillator you need to set the Mega163's internal configuration fuses for an external clock. For the MOB select the external ceramic resonator with slow rising power.

Pony Programmer 2000 has all the fuse bits for the Atmega163 (look under edit.modify security bits and Command.{read, write} security bits) You will need to consult a data sheet to set the bits properly. The bits I would use are:

Boot sector = 128 words (BOOTSZ = 11)
Boot reset vector enabled (BOOTRST = 1)
BOD enabled (BODEN = 0)
BOD level = 4.0v (BOLEVEL = 0)
CKSEL = Ceramic resonator, BOD (CKSEL = 1111)

The MOB software binaries and are called "mob.hex" and "mob.eep".

NOTE: If using the STK500 development board from Atmel you must make sure to check the VTARGET jumper on the STK500 before connecting the programming cable to the MOB board. If you are powering the MOB from an external source or from an OSMC board you must remove the VTARGET jumper on the STK500. If you have no external power attached to the MOB you need to install VTARGET to supply power to the MOB. Also, you must remove any chips installed in the STK500 sockets.

MOB – Modular OSMC Brain

INSTALLATION

Connecting MOB to two OSMC boards: The MOB is connected to up to two OSMC boards by using a 20-conductor ribbon cable and IDC connectors.

1. To start carefully crimp the 20-conductor cable to the 20-pin MOB IDC connector. On gray cable with a red strip, the strip should connect to pin 1 of the connector. If you don't have an IDC crimper tool you can use a vice. Simply place the connectors in the vice and squeeze them together stopping when they are fully mated while holding the ribbon cable steady. We don't recommend using pliers as they usually have teeth that can mar the sides of the connector and interfere with mating of the connector with the pins. If you are using a strain-relief bar be sure the cable exits away from the MOB with the strain relief installed. Make sure the cable is long enough to reach the location of the OSMC if you are mounting them away from the MOB. Note, these are not shielded cables and should not be used for long runs. It is better to mount the OSMCs as close to the MOB as possible and use longer motor or battery wires.
2. Once the cable is connected to the MOB connector carefully split the cable into two 10-conductor halves. This can be done by carefully cutting the insulation between the halves of the cable at the end. Once a short cut has been made the cables can be pulled apart gently by hand and they will split along the insulation between the conductors.
3. When the cable is split install the 10-pin IDC connectors using the same method you used for the 20-pin connector. Double check that conductor 1 is in the same location on the 20-pin and on the first 10-pin connector. Then orient the second 10-pin connector the same way and crimp it.
4. To connect the MOB to the OSMC boards simply insert the appropriate connectors and raise the locking tabs. To eject the cables press the locking tabs down all the way to eject the cable. Using locking tabs is highly recommended as they both secure the cable and save wear and tear on the ribbon cable when installing and removing the cable from the boards. Additional security may be added by securing the cables with a dab of hot-glue. Since these are low current cables they won't heat up and soften the glue like the high-current wires might.

Connecting LCD: The LCD display is connected to the MOB using a 14-conductor cable connected to the MOB at J7. The signal lines for the four command buttons are also in this cable. The MOB uses the LCD's 4-bit mode so the additional 4 lines on the 14-conductor cable are available for the command buttons. If you are constructing your own LCD interface these lines should not be connected to the LCD module pins but broken out and connected to the buttons. You can also simply not connect these lines at all and use the buttons mounted on the MOB PCB.

1. The most common connector for the LCD is a single-row female connector with crimp pin connections. Either discrete wires or a ribbon cable may be used for this. Special ribbon cable with double spacing is available but is not as common as the standard ribbon cable with .05" spacing. To use the standard ribbon you will need to separate each conductor and spread them all out to match the connector. The outer wires will have farther to go to reach the connector so you may need to clip the center wires to keep an even distance once the ribbon is split apart.
2. To connect the individual conductors you must strip a short length of wire and then carefully crimp the wire into the female pin. When all these are done carefully insert the pins into the connector housing until they click. Push them in by the wire as you shouldn't need to use anything to force them in. Give them each a gentle tug to make sure they are seated. Be sure that wire #1 is connected to pin one on the connector.
3. Repeat this procedure for the LCD side of the interface. Remember to split off the lines from the four command buttons and don't connect those to the actual LCD module.
4. Route the 4 button wires to the buttons on the pendant if needed.

MOB – Modular OSMC Brain

Connecting to Radio Control: There are several ways to connect the MOB to the radio receiver. Basically the radio receiver uses a 3-pin connector with 0.1” spacing. However, many radio manufacturers have added tabs and other shapes to the connector to aid in orientation and insertion/extraction. The easiest way to connect the MOB to the radio receiver is as follows:

1. Purchase 2 or 3 servo plug/cable replacements for your particular radio system. Several aftermarket sources exist for these parts. Generally these have a 12” pigtail cable connected to a female plug. A more expensive alternative is a servo extension cable compatible with your radio system. Often generic plugs or extensions will work with several radios and cost less than “name brand” extensions. Servo extensions come in various lengths from 6” to 24” or more.
2. One end of the cable will go into the receiver. This is a female end. The other end will either be bare wires or a male connector designed to plug into the female end on the actual servo. This male end will be shrouded to protect the servo’s female connector in normal operation. What we need to do is connect the male/bare side of the cable to the MOB board. There are basically 4 options.
 - a. Cut off the male end (if any) and solder the wires into the MOB at locations J1, J2 and J3.
 - b. Break off the shroud around the male pins to expose them and solder the male connector pins into the board.
 - c. Install a female connector on the MOB to accept the male pins. A standard 3-pin female header with 0.1” spacing will work.
 - d. Install regular male header pins on the MOB and replace the male servo extension connector with a female 3-pin header. A friction lock header should be used with an appropriate header for a secure connection.

Since the servo connections will be subject to a lot of plugging in and out during development and vibration during use, a permanent connection to the MOB is recommended i.e. option a. or b. above. An alternative is to use a friction locking male header and replace the bare or male servo end with a mating female connector. This last option provides the most flexibility with good reliability but at the most cost.

To construct a completely custom cable for the MOB to radio connection you’ll need to build female-female cables with 3-pin connectors. You should use friction lock headers on the MOB side or solder the wires into the board for a secure connection. The radio-side connector should be a plain 3-pin female Molex-style connector without locking ridge or polarizing tabs. Amp/TYCO 3-pin plain housings will work for most radio receiver plugs. You will probably have to trim the radio-side connector to match the shape of the opening on the radio receiver. These are often tapered or beveled to force proper orientation. Using Molex/Amp connectors is the lowest cost option and allows the most flexibility in cable length etc. Be sure to match the signals from the receiver to those on the MOB. Generally the red and black wires are +V and GND respectively and the yellow or white wire is the signal wire. Looking at the parts that came with the radio will give you a clue as to what wire does what. You can often look at the battery connector (it often has only two wires) for your radio to be sure which are the power leads.

INPUTS

The following RC inputs are available on the MOB board:

Label	Name	Mix Mode function	Normal Mode function
THRO	Throttle	Speed forward or reverse	Motor A speed
DIR	Direction	Steering right or left	Motor B speed
AUX	Auxiliary	Switch AUX 1&2 outputs on or off as stick passes center point (see below)	

MOB – Modular OSMC Brain

There are two main operational modes to the MOB software - Mixed Mode and Normal Mode a.k.a Tank Mode.

In Mixed Mode the throttle input controls the speed of both motors equally while the direction input causes a differential to be added or subtracted from the throttle speed setting for each motor. Thus a 50% throttle setting causes both motors to spin at 50% power in the same direction. Moving the stick to the left causes the right motor to spin faster and the left motor to spin slower proportional to the stick position. This allows a single transmitter stick to be used for speed and direction in an intuitive way. The stick is pushed forward or back from the center point to control the overall forward or reverse speed of the robot and the left or right position of the stick controls the relative speed of the motors to achieve turning and steering. For high speed maneuvering and driving this mode is generally best.

In Normal Mode two control inputs are used to control the speed of the two motors independently. This is similar to a tank or tracked construction vehicle where the speed of each side of the vehicle is controlled independently by a single control. For slow speed driving or where spinning in place is needed this mode may be best. Normal mode may also be used where mixing is done on the transmitter or in an external mixing unit.

The auxiliary channel operates the same in both Mixed and Normal modes. The Aux1 and Aux2 outputs are optically isolated from the rest of the MOB system and are designed to safely activate relays or other high-power devices without risk of electrical feedback affecting the MOB. The auxiliary outputs work in a complementary fashion so only Aux1 or Aux2 will ever be on at one time. When the RC signal connected to the AUX input is “above” the center value output Aux1 will be turned on and Aux2 will be off. Conversely when the signal is “below” the center value Aux2 will be on and Aux1 will be off. This allows a number of options for controlling on-off devices with the AUX channel. For example only Aux1 could be used to control a weapon. To turn the weapon on the control stick (or switch) is set to a value above the center point. To turn it off the stick is centered or moved below the center point. To control a bi-directional weapon, say an arm that extends and retracts, one Aux output is used to extend the arm while the other is used to retract it. Since the Aux outputs are never both on at the same time, the weapon will respond as expected to all command inputs.

NOTE: the current output of the Aux driver is very low and additional circuitry is required to switch a physical device such as a relay..

Some experimentation will be needed to determine which RC channel from the receiver should be connected to which input. Generally receiver channel 1 (aileron) is the DIR input while channel 2 (elevator) is used as the throttle input. The AUX input may be driven by either a stick such as the non-spring loaded throttle stick or by a switch such as a landing gear switch.

JUMPER SETTINGS

JP1: Battery Eliminator Circuit (BEC) for receiver On/Off (default Off). This jumper connects the power lead on the RC input connectors to the 5V supply of the MOB. It can be used to power the RC receiver so an external battery is not needed. The 5V supply on the MOB is not sufficient to power servos or other high-current devices so remove this jumper and use an external battery if other devices such as servos are connected to the receiver besides the MOB.

JP2: SPI Slave Select (SS): On/Off (default Off). This jumper connects the Slave-Select pin of the SPI bus to the 10-pin SPI/ISP programming connector. Since in-circuit programming does not use this line JP2 should be used to disable it during programming. It can be re-installed to allow operation of external SPI devices during normal operation. If you don't have any external SPI devices connected, this jumper can be left off permanently.

MOB – Modular OSMC Brain

LED INDICATORS

The MOB board has 4 LED indicators that provide a quick visual indication of the status of the board. These are intended for use when the LCD is not connected.

The following LED indicators are available on the MOB board:

Label	Name	Function
POWER	Power	LED lights when 5V power is available to the board
D2	Status	Blinks in various patterns to show status of the board and RC inputs (see below)
Right Disable	Motor A disable	LED lights when Motor A output is disabled
Left Disable	Motor B disable	LED lights when Motor B output is disabled

Note, the MOB has the capability to independently disable Motor A and Motor B but the current software release always disables or enables them together. You should always see Right Disable and Left Disable either on or off together.

The status indicator D2 is the primary means of monitoring the overall status of the MOB when the LCD is not connected. It blinks in the following patterns to indicate various states of the system:

- Slow Flash (50% on 50% off - 1/sec): Ok
- Fast Flash (50:50 - 2/sec): Corrupt EEPROM
- Blink (13% on, 87% off): No input signal
- Wink (87% on, 13% off): Output disabled
- Double Blink (12% on, 12% off, 12% on, 64% off): Output disabled AND no signal

OPERATION

For normal operation all that is required are power to the MOB board (usually supplied by the OSMCs) and the RC signal inputs. The MOB will automatically start when power is applied and begin operation without user input. Upon startup software immediately disables all OSMC outputs and begins monitoring the RC inputs for valid signals. The OSMC boards are disabled until a valid pulse stream is detected. If the RC signal on either of the two main inputs is lost the MOB immediately disables all outputs. These features are designed to meet BattleBots™ safety regulations.

If the LCD display is attached, the MOB displays a summary information screen during normal operation. The LCD is not required for operation and does not affect the operation of the MOB software whether it is connected or not.

A CPU reset button labeled RESET is available on the MOB board near the command buttons. Pressing this button will force a hard reset of the system CPU chip. This should not be needed in normal operation. It is available for unusual situations or to test the system response.

It is unlikely but possible that the internal EEPROM of the MOB CPU may become corrupted. This most commonly happens during programming when the MOB software is updated and the EEPROM is erased. This condition is indicated by the status LED D2 flashing rapidly and no response from the motors. To return the MOB to a safe operating condition the software disables all outputs and waits for any of the command buttons to be pressed. Once a button is pressed the software loads the default options into the EEPROM and continues. If you have set the software options differently than the default you will have to re-configure the software via the LCD.

MOB – Modular OSMC Brain

MOB Software Options

The MOB software provides several user configurable parameters to optimize its operation for a particular robot and usage scenario. The configurable options in this version of the MOB software are:

1. PWM Mode:
 - Sign Magnitude (default)
 - Locked Antiphase
2. PWM Rate
 - 2 kHz (default)
 - 4 kHz
 - 16 kHz
 - 32 kHz
3. Mixing Mode
 - Normal (default)
 - Mixed
4. Output Enable
 - PWM Outputs Enabled
 - PWM Outputs Disabled (default)
5. Deadband
 - Selectable 0-8. 8 gives ~10% stick throw deadband (2 default)
6. Store Settings?
 - Store for use even after power down or reset
 - Use only until power down or reset (default)

Setting Software Options

The MOB software options are changed by using the four buttons on the MOB PCB or on a remote LCD pendant. The user interface is very simple. Here are the functions of the four button switches:

Switch	Function	Notes
UP	Change value higher	Options wrap around in a circular fashion
DOWN	Change value lower	Same
LEFT	Select next configuration item	See option list above. Config Items wrap around
RIGHT	Select previous item	

The MOB software UI consists of seven “screens” or displays. They are listed below with the function of each button:

1. Status
 - UP/DOWN: toggle radio signal diagnostic display
 - RIGHT: go to PWM Mode
 - LEFT: go to Store Changes?
2. PWM Mode
 - UP/DOWN: toggle Sign Magnitude and Locked Antiphase mode
 - RIGHT: go to PWM Rate

MOB – Modular OSMC Brain

LEFT: go to Status

3. PWM Rate
 - UP/DOWN: cycle through PWM rate choices
 - RIGHT: go to Mix Mode
 - LEFT: go to PWM Mode
4. Mix Mode
 - UP/DOWN: toggle Mixed and Normal modes
 - RIGHT: go to Output Enable
 - LEFT: go to PWM Rate
5. Output Enable
 - UP/DOWN: toggle Enabled or Disabled
 - RIGHT: go to Deadband
 - LEFT: go to Mix Mode
6. Deadband
 - UP/DOWN: cycle through deadband values 0-8 with 8 being the largest deadband
 - RIGHT: go to Store Changes?
 - LEFT: go to Output Enable
7. Store Changes?
 - UP/DOWN: Store current settings to EEPROM
 - RIGHT: go to Status
 - LEFT: go to Deadband

CAUTION: It is important to realize that changes to the software options take place immediately and that RC pulse processing and PWM output continue during setup. This allows you to immediately see the effect of changing the various options but may also cause unexpected motion in your device. Ideally you should disconnect the PWM cable from the MOB when changing the software options. At a minimum you should always place your robot on blocks or otherwise prevent it from moving or actuating weapons while you are configuring software or reprogramming the MOB. You must also remember to store the changes into EEPROM for them to be permanently set. Otherwise the previous settings will be restored when the processor is reset or power is removed. In general it is best to always save the settings after changing anything.

HARDWARE SPECIFICATIONS

Power: 7.5-30V input, nominally 12V from OSMC board. 5V at 1A max is generated on board.

CPU: Atmel ATmega163-8AC

System clock speed: 8MHz

User Interface: 20x4 LCD compatible with Hitachi 44780 and four command buttons.

Display contrast: Adjustable using R23

PWM frequency: 2, 4, 16, and 32 khz user selectable

PWM mode: Locked-antiphase or Sign-magnitude PWM by user selection

Deadband: adjustable by user

I2C interface: at J6

Serial interface: One logic-level (compatible with RS-232 signals) via 4-pin header at J4

Analog I/O channels: 4 connected through J9. Two additional channels monitor battery voltage from the OSMC boards.

ISP: Standard 10-pin header for in circuit programming

SPI interface: using the ISP header J5 and the JP2 jumper.

MOB – Modular OSMC Brain

IN CASE OF DIFFICULTY

PROBLEM	POSSIBLE CAUSE
MOB does not power up.	<ol style="list-style-type: none">1. Check for solder bridges and cold or open solder joints.2. Make sure all components are installed in the proper locations.
Large current draw when powered.	<ol style="list-style-type: none">1. Check orientation of C2. Banded end is positive.2. Make sure all components are installed in the proper locations.3. Check for solder bridges.
Can't program Atmega163	<ol style="list-style-type: none">1. If connected to OSMC and using the STK500 then make sure VTARGET jumper is removed.2. If not connected to OSMC and using STK500 then make sure VTARGET jumper is installed.3. Insure all chips are removed from sockets in STK5004. Check for solder bridges and cold solder joints on Atmega163.5. Check clock FUSE settings in programming SW.
Erratic operation	<ol style="list-style-type: none">1. Check for solder bridges and cold solder joints.2. Check clock FUSE settings in programming SW.3. Check SW revision installed on MOB.
Very slow operation	<ol style="list-style-type: none">1. Make sure Mega163 fuses are set for external ceramic oscillator and slow-rising power.
Motors don't turn	<ol style="list-style-type: none">1. Verify that output is enabled in MOB software2. Check motor and battery connections3. Check radio transmitter power, receiver connections and JP1. If used in BEC mode JP1 must be connected4. Check PWM cable connections and orientation

CIRCUIT DESCRIPTION

The heart of this board is the Atmel AVR Atmega163. It is like an entire computer on a chip and already provides much of the I/O built into the chip.

MOB – Modular OSMC Brain

REPLACEMENT PARTS LIST

Most of the parts are listed below the Digikey part number <http://www.digikey.com> and second number below showing the same part from Mouser Electronics <http://www.mouser.com> as another source. In some cases there will only be the generic industry number.

Quantity	Part Number	Description	Location
1	PCC1901CT-ND	.47uf capacitor – 1206 SMD	C1
1	P11290CT-ND	33uf tantalum capacitor – 6032 SMD	C2
10	PCC1870CT-ND	.1uf capacitors – 1206 SMD	C3-C12
10	P10KECT-ND or 263-10K	10K ohm resistor – 1206 SMD	R11-R18, R21, R24
6	P680ECT-ND or 263-680	680 ohm resistor – 1206 SMD	R19-R20, R22, R25-R27
10	P1.0KECT-ND or 263-1K	1K ohm resistor – 1206 SMD	R1-R10
1	D4AA23-ND	2K .1W potentiometer	R23
1	160-1140-ND or 351-3102	3mm RED LED – T-1	D1
3	160-1143-ND or 351-3003	3mm GREEN LED – T-1	D2, D7, D8
1	1N5231BMSCT-ND or 583-1N5231B	1N5231B 5V Serial Zener – D0-35	D4
2	UF1002DICT-ND or 625-UF4002	UF1002 1A 100V Ultra fast diode – DO-41	D5, D6
1	LM2940CT-5.0-ND or 511-L4940V5	5V Regulator – TO-220	U1
1	ATMEGA163-8AC-ND	Atmel AVR Atmega163 CPU – TQFP 44 SMD	U2
1	296-4262-1-ND	74ACT158 – SO-16 SMD	U3
1	MM74HC244WM-ND	74HC244 – SO-20W SMD	U4
2	67-1563-5-ND or 512-4N26	4N26 – Opto-isolator 6-pin DIP	U5-U6
1	PX800MC-ND	8.0 Mhz Ceramic Resonator / cap	X1
5	P8009S-ND or 101-0261	NO momentary switch -6.5mm x 4mm	LEFT,RIGHT, UP, DOWN, RESET
1	MHB10K-ND	2x5 header - protected	ISP (J5)
1	CHS20G-ND	2x10 header – protected	J8
1 pair	CLKS01-ND	Locking tabs for above	J8 tabs
1	929834-02-36-ND	.1" single row header (break apart)	J4, J6-J7, J9-J10
1	MOB1.3	Custom Printed Circuit Board (through hole plated)	

Revision History

Breadboard - Initial prototype.

Version 1.1 – Did this exist. What changed?

Version 1.2 – Did this exist. What changed?

Version 1.3 – Current version of the MOB.

SPECIAL THANKS

A note of thanks to the following people who helped with the project: Dennis, David, Larry, Chris, Sonny, Jeff, Angela, Steven among many others who have done so much for this project. If you should be included on the above list let me know!!