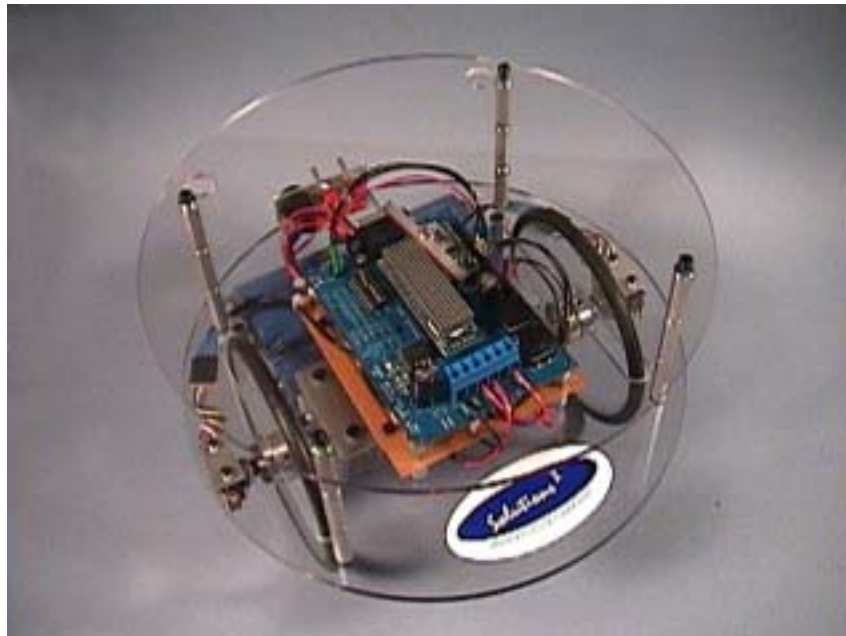


**Overview:**

Early in the development of the Motor Mind C Solutions Cubed asked Paul Garcia, a member of the Robotics Society of Southern California (RSSC), to do some beta testing of the product. What follows are a brief biography for Paul, and the results of his tests. Paul used the Motor Mind C Rev3 and the Motor Mind C Carrier Board to add versatility to the club robot for RSSC. The document following this overview was written by Paul, and Solutions Cubed would like to extend our thanks to him for evaluating and documenting his work with our Motor Mind C product.

Paul Garcia has been building models and electromechanical gadgets for the last 30 years, but has only taken up hobby robotics since September 2001 when he joined the Robotics Society of Southern California (RSSC). As a Boeing Project Engineer, he is primarily responsible for directing the Engineering efforts on large commercial aircraft modifications, such as DC-10/MD-11 passenger to freighter modifications. Having to work out the design and installation issues on each robot has given him a greater appreciation for the efforts of the "detail guys".

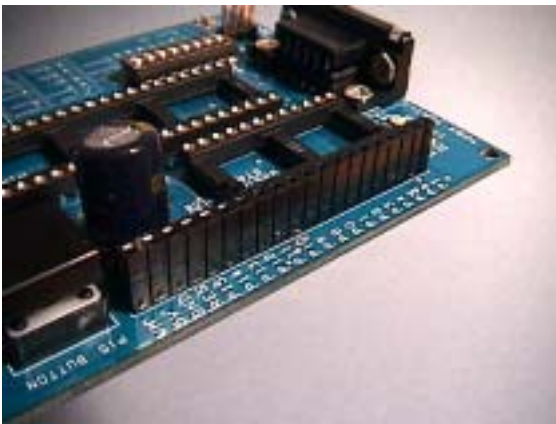
**Using the MMC Carrier Board with the BasicMicro ATOM – by Paul Garcia**

Some time after completing the Ebot II robot described in App note AN701, I decided I wanted a more "integrated solution". The original build utilized a Parallax Board of Education, which necessitated a separate carrier board for the MMC. After considering different options, I decided to upgrade my installation with a Motor Mind C carrier board. The conversion was quick, and the end results speak for themselves.

The photographs below show the steps involved:



1.) The MMC carrier board has provisions for accessing the individual micro controller pins via solder connections at J6. Since the board would be used as a robot controller, a .100 mil 20 pin single row receptacle was added to J6 to make for easy connections.



2.) Receptacle installation complete.



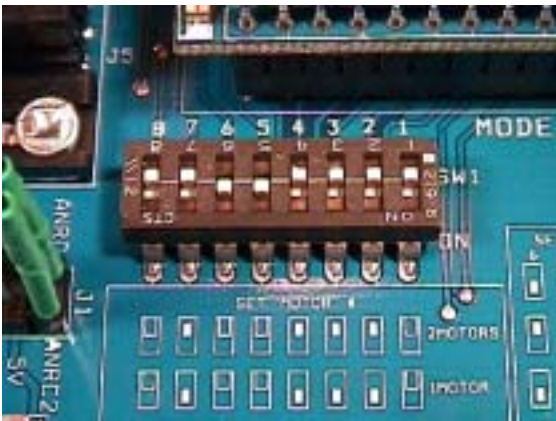
3.) The MMC is installed in the J2 DIP socket, with the LEDs closest to the serial connector. Place the carrier board on a firm surface, and align the MMC pins with the socket before pressing firmly on the heat sink. Be sure the MMC is seated all the way around.



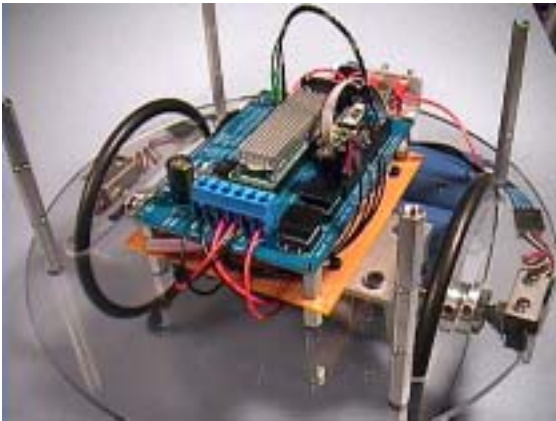
4.) The ATOM microprocessor is installed in J4 DIP socket. Exercise care when seating the ATOM's pins. The gray ribbon cable coming from underneath the ATOM in this photo was added prior to installation. The four conductors are soldered to the A0–A3 solder pads on the underside of the ATOM, which are used for A to D input.



5.) In order to use the ATOM's PWM output, it is necessary to use RC filters to smooth the pulses into an analog output. The filters are connected between pin 9 & AIN1 and pin 10 & AIN2. This photograph shows a nifty little freeform RC filter I came up with.

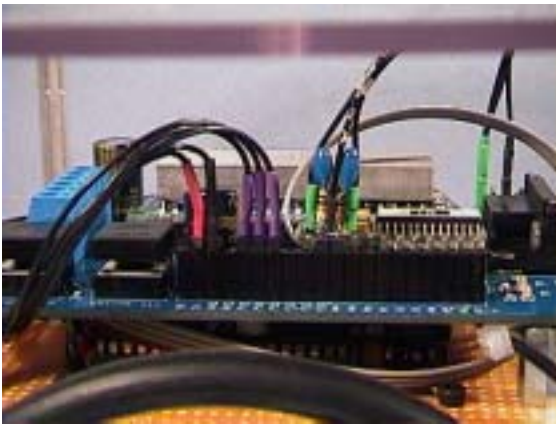


6.) The DIP switches were set for analog control of dual motors per section 4.2, figure 3 of the MMC carrier board datasheet.

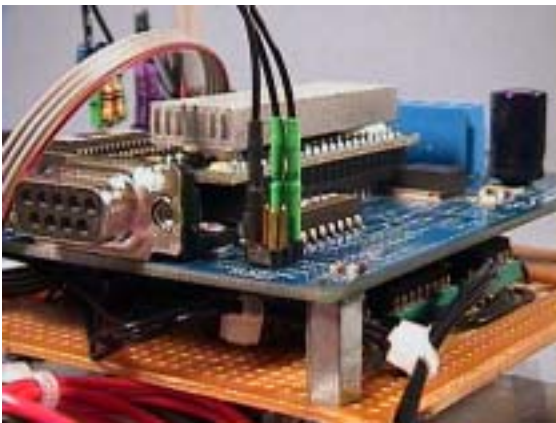


7.) The MMC carrier board was installed on the robot using  $\frac{1}{2}$ " long aluminum hex standoffs and R/C steering servo tape. This tape is available in the R/C car section of the hobby shop, and uses thinner foam than normal servo tape. The motor (Left motor to M1, and right motor to M2) and battery leads have been connected, which completes the conversion.

(The board under the MMC is an interface board used to read the wheel encoders.)



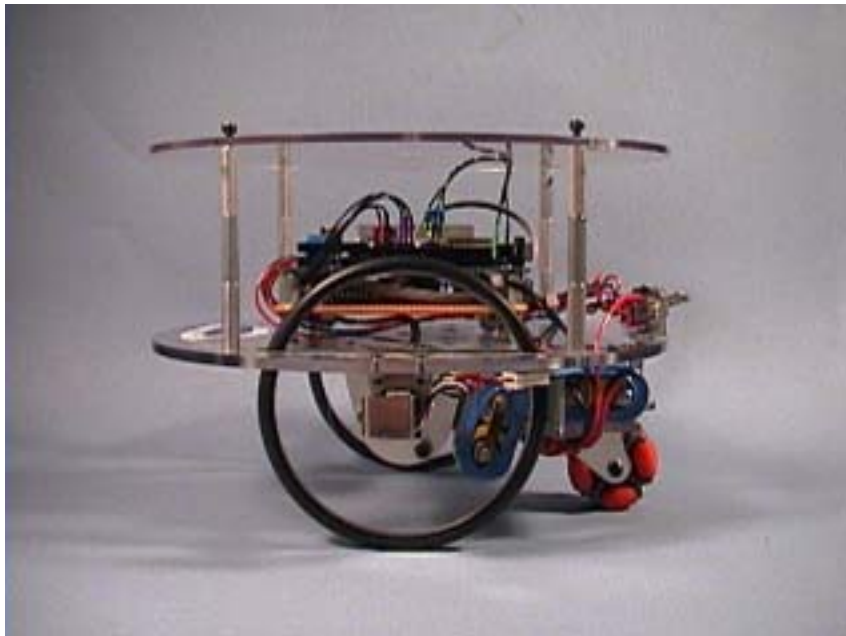
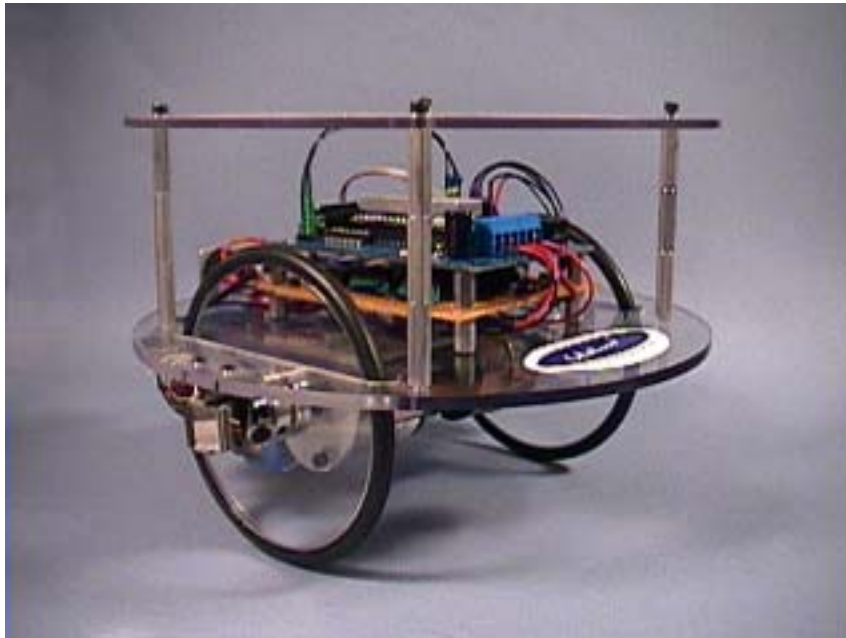
A close-up view showing the wheel encoder interface and RC filter installed on the receptacle strip.



A close-up view of the AIN 1/AIN 2 connection.



The completed robot:



30 minutes later, and the robot is ready to run. Having the ATOM and MMC on a single board makes for a much neater installation.

**Considerations:**

The MMC carrier board Pins P7 and P8 (serial lines) are hardwired from the MMC to the MCU socket. Pin 8 (Connected to the MMC pin 21-FM) can be used as an input or output with the MMC configured for Analog Mode. P7 (connected to MMC pin 20-TM) should be left unconnected. As outputs, the MCU will have to sink about 4mA P8 is driven low (the LED D2 will light).

Although P14 & P15 of the MCU is hardwired to SW2 & SW3, these pins can also be used as inputs or outputs. As outputs, the MCU will have to sink about 5mA when P15 or P14 are driven low. In this case though you would not want to press SW2, SW3 when operating the pins as outputs. It should be noted that the pin designations (for the button connections) are wrong on the revision 2 of the PCB (see datasheet page 4). The schematic for these connections is correct.

Paul Garcia  
RSSC  
November, 2002

**Example Code:**

'INTERFACING AN ATOM TO THE MOTOR MIND C MODULE/BOARD - ANALOG MODE.

'This program is a modified version of the MMC AN700 software, intended to  
'Demonstrate the same functions in ANALOG mode, using the BasicMicro ATOM.

'GENERATING AN ANALOG OUTPUT:

'Using a PWM output to generate an analog voltage on any MCU requires the use  
'of an RC filter, which filters out the pulses and store the average voltage.

'The ATOM require two RC filters, one on P9, and another on P10:

```
'
' P9/P10 ---- 10K ohm --+----- AIN1/AIN2
'                   |
'                   .1 uf cap
'                   |
'                   Gnd
'
```

'DIR DIFFERENCE:

'To quote the ATOM users manual (Rev2.2):

"0 loaded into a particular DIRS bit makes the corresponding pin an output  
'and a 1 makes the corresponding pin an input. This is opposite for the  
'stamp, which is 0 for input and 1 for output. This reversal with the Atom  
'was done because this is the way the TRIS registers are handled on the  
'PICmicro® MCU the Atom is built around."

'ACCELERATIONS AND LIMITS:

'The Ebot II uses two 6 cell 7.2 volt nicad (R/C car) battery packs wired in series.

'When freshly charged, the voltage can be as high as 18 volts. Since the Jamco style  
'motors are rated at 12V, and utilize plastic gears, For/Next loops are used to  
'provide acceleration control to minimize strain/wear on the motors.

'The Fullspeed command limits the PWM to a 66% duty, which prevents commands that  
'result in more than 12 Volts being applied to the motors.

'BUTTON FUNCTION:

'The original AN700 uses the SW2 and SW3 buttons to demonstrate an acceleration  
'Since an acceleration is demonstrated in the full forward routine, and deceleration  
'in the creep forward routine, this program only uses button SW3 to initiate the  
'robot movement.

\*\*\*\*\*

'Variables

BTN_VAR	VAR	BYTE	'Generic var for use with BUTTON cmd
Maxspeed	VAR	WORD	
x	VAR	WORD	'Loop counter
LmotorStop	VAR	WORD	
RmotorStop	VAR	WORD	
MotorStop	CON	8191	
StopCorL	CON	65	
stopCorR	CON	131	
correctR	CON	0	
correctL	CON	250	
FullSpeed	CON	5000	
HalfSpeed	CON	2500	
SlowSpeed	CON	750	

```

'Set ATOM i/o direction and initialization
  DIRS          =%1111100111111111 'Set P9 & P10 as output all others inputs
  OUTS          =%1111111111111111 'Set all outputs high
  LmotorStop = MotorStop + StopCorL
  RmotorStop = MotorStop + StopCorR

  HPWM 1,16383, LmotorStop          'Set AIN1 PWM output to 2.5 volts
  HPWM 0,16383, RmotorStop          'Set AIN2 PWM output to 2.5 volts

  PAUSE        1250                  'Wait 1250ms for MMC to power up

  BTN_VAR      = $00

START_ROBOT:
  BUTTON      15,0,5,1,BTN_VAR,0,START_SKID_STEERING 'Red FM LED flashes until button pressed
  LOW P8
  PAUSE 50
  HIGH P8
  PAUSE 500
  GOTO START_ROBOT

*****
'SKID_STEERING_MODE:      This mode of operation can be used when the Motor Mind C
'                          is in dual motor mode and is controlling a robot using skid steering.
'                          In skid steering direction changes are implemented by adjusting motor
'                          speed or direction. Two drive motors are used for skid steering.
'                          In this example it is assumed that motor 1 is located on the left-hand
'                          side of the chassis. It is also assumed that when given a positive value
'                          motor 1 turns forward, and that motor 2 requires a negative value to
'                          turn forward.
*****

START_SKID_STEERING:

'FULL-FORWARD
'drives both motors at top speed pulling the robot forward

  PAUSE 2000                          '2 second pause to move hand away
  FOR X = 1 to FullSpeed STEP 2        'Accelerate to fullspeed
    HPWM 1,16383, LMotorStop + X
    HPWM 0,16383, RMotorStop - X
  NEXT
  PAUSE 2000

'LEFT-FORWARD –
'drives motor1 at half speed and motor 2 at full speed causing chassis to veer left

  HPWM 1,16383, LMotorStop + HalfSpeed
  HPWM 0,16383, RMotorStop - FullSpeed 'It is not mandatory to change
'                                       'motor2 speed - it will remain
'                                       'at full speed

  PAUSE        2000

```



**'RIGHT-FORWARD**

'drives motor1 at full speed and motor 2 at half speed causing chassis to veer right

```
HPWM 1,16383, LMotorStop + FullSpeed
HPWM 0,16383, RMotorStop - HalfSpeed
PAUSE          2000
```

**'CREEP-FORWARD**

'drives both motors at slow speed allowing the chassis to creep forward

```
FOR X = FullSpeed to SlowSpeed STEP -2      'Decelerate to slow speed
      HPWM 1,16383, LMotorStop + X
      HPWM 0,16383, RMotorStop - X
NEXT
PAUSE 2000
```

**'CREEP-PIVOT**

'drives both motors slow with motor 2 reversed allowing the chassis pivot slowly

```
HPWM 1,16383, LMotorStop + SlowSpeed
HPWM 0,16383, RMotorStop + SlowSpeed
PAUSE          2000
```

**'FAST-PIVOT**

'drives both motors at full speed with motor 2 reversed allowing the chassis spin

```
HPWM 1,16383, LMotorStop + FullSpeed
HPWM 0,16383, RMotorStop + FullSpeed
PAUSE          2000
```

**'STOP**

'stops both motors

```
HPWM 1,16383, LMotorStop
HPWM 0,16383, RMotorStop
PAUSE          1000
```

**GOTO START\_SKID\_STEERING**