SOLUTIONS CUBED

Motor Mind C Carrier Board
Data Sheet

Revision 1
March 8th, 2002

Table of Contents – Electrical / Mechanical Specifications

List of Figures

1. Revision Log – Electrical / Mechanical Specifications

Date	Rev	Description	By
03-08-02	1	Original Implementation	L. Glazner

2. Introduction

Motor Mind C Carrier Board

Features

- Screw terminal for motor connections
- BASIC Stamp 2, 2SX, 2P24, 2E socket and programming port
- MMC_12VAC fan kit mounting holes
- TM, FM indicator and Power LEDs
- Analog and R/C inputs compatible with R/C receivers
- Motor Mind C mode select DIP switch
- 5V 100mA regulator

2.1 Description

The Motor Mind C Carrier Board was designed to simplify connectivity to, and control of, the Motor Mind C. It can also be used to implement application notes involving the BASIC Stamp 2 series of programmable controllers sold by Parallax Inc.

3. Engineering Specifications

3.1 Absolute Maximum Ratings

These are stress ratings only. Stresses above those listed below may cause permanent damage and/or affect device reliability.

Storage Temperature	-55°C to +150°C
Operating Temperature	-20°C to +85°C
Motor Voltage (V_{MOT})	-0.3V to 30.0V
Voltage on control pins	-0.3V to +5.5V
Voltage on V_{MOT}, Mx+, Mx-	30V
Motor Current Load	5A peak / 4.0A continuous

3.2 DC Electrical Characteristics

At T_A = 25°C, V_{MOTOR} = 12V, I_{LOAD} = 0.5A V_{5VDC} = 5V

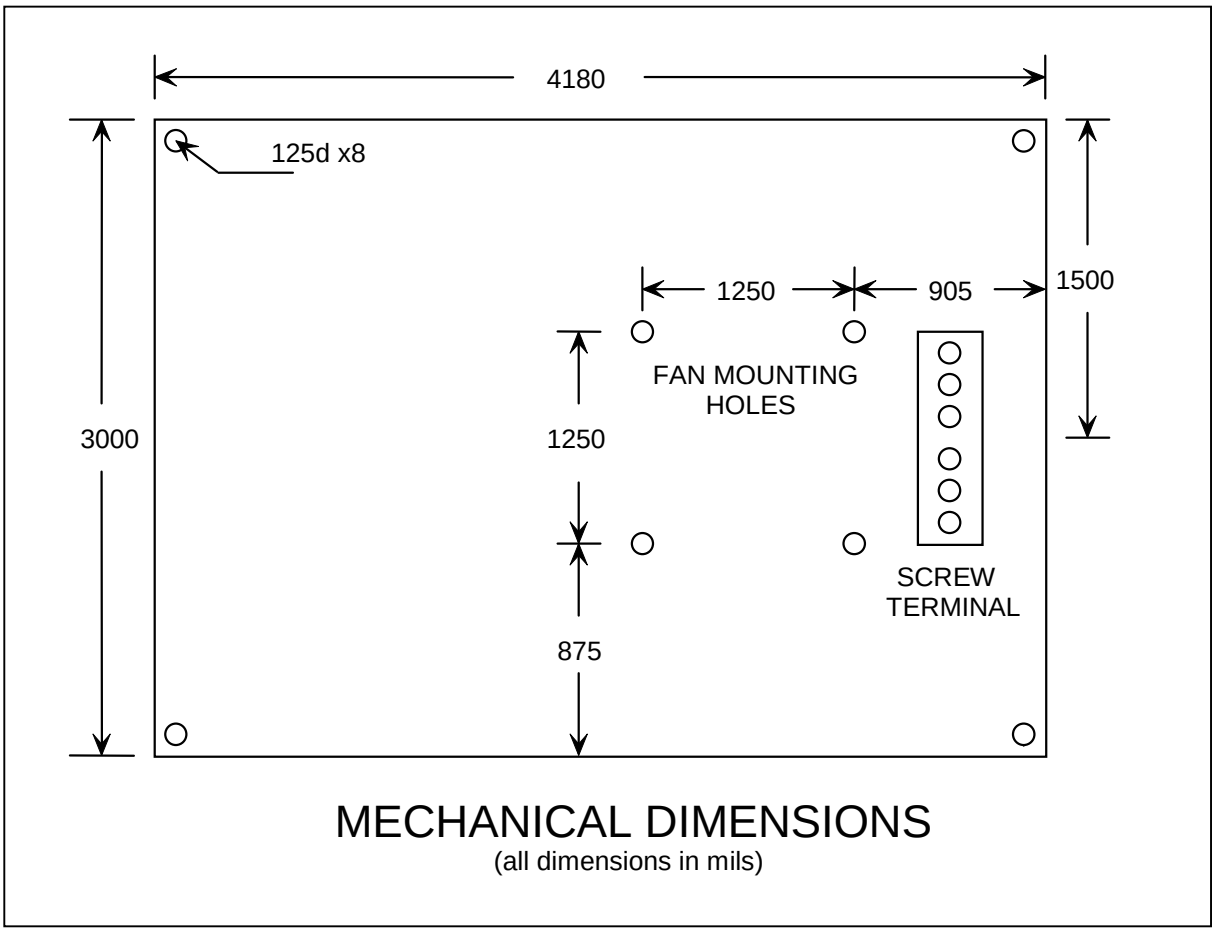
Characteristic	Symbol	Min	Typ	Max	Unit	Notes
Motor Supply Voltage	V_{MOT}	10		24	V	
ANx/RCx input voltage range	V_{AN}	0		5	V	5V is the full-scale input for the 8-bit ADC
Peak load current	I_{PK}			5	A	Transient <500ns
Max continuous current	I_{CONT}		4.0		A	

note: “Typ” values are for design guidance only and are not guaranteed

3.3 Mechanical Dimensions

The following diagram may be used to develop enclosures for use with the Motor Mind C Carrier Board
.

Figure 1: Mechanical Dimensions



3.4 Connectivity

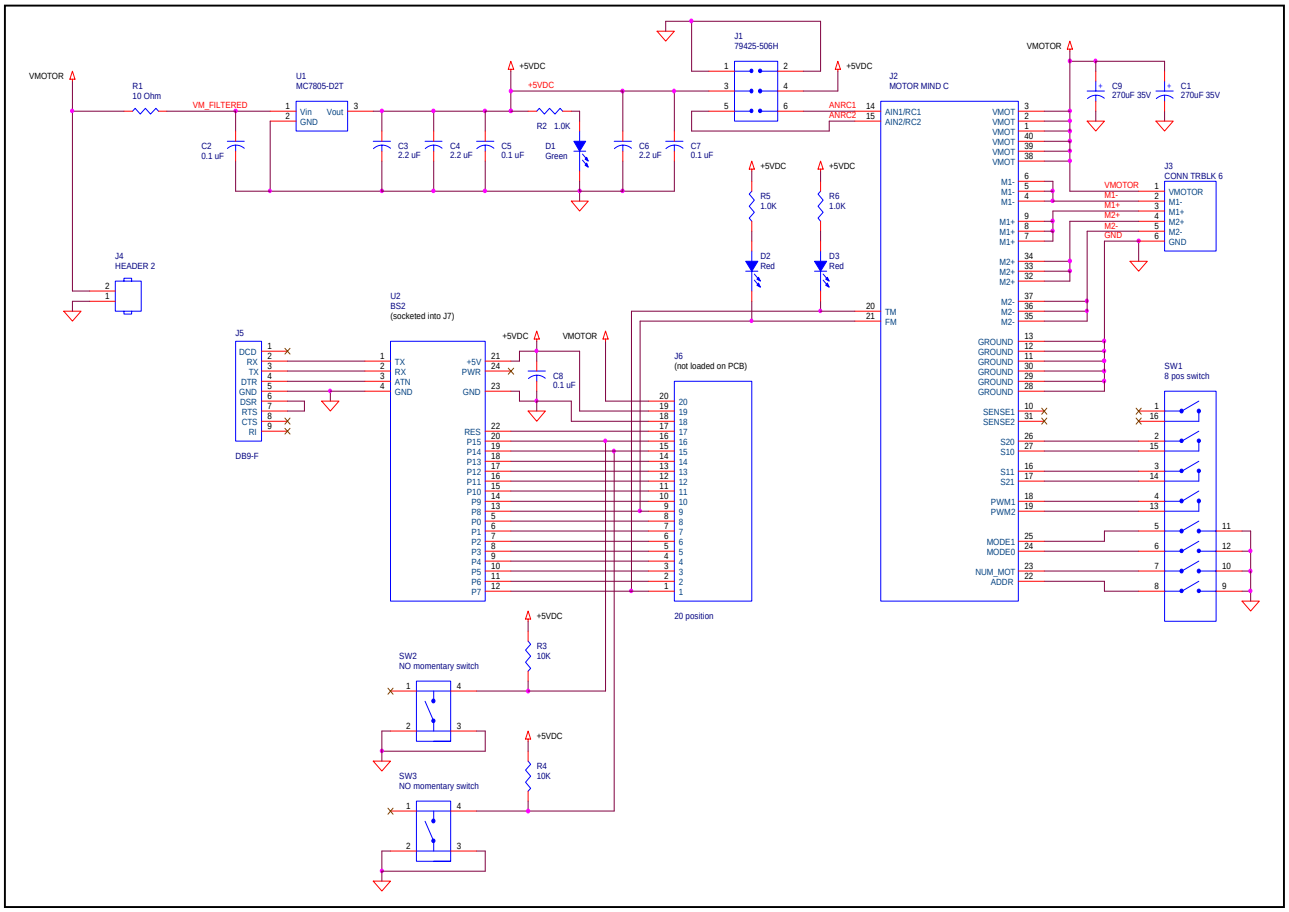
The Motor Mind C Carrier Board comes equipped with a socket for the Motor Mind C and a socket compatible with the Parallax BASIC Stamp 2, or BS2 (including the BS2SX, BS2P24, and BS2E). Support for the BS2 includes a programming port (J5), and a strip of 0.1" plated holes that connect directly to the BS2 I/O pins. Two momentary push buttons are connected to BS2 pins P15 and P14 to help facilitate a rudimentary user interface like that used in AN700. **Please note that the silk screen labels for these buttons are reversed on REV2 of the Motor Mind C Carrier Board.**

Serial data communication with devices other than the BS2 can be accomplished by connecting the *"data to"* the Motor Mind C at the point labeled P8. Serial *"data from"* the Motor Mind C will be present at the point labeled P7. Serial data at P8 will cause the FM LED to flash, while data present at P7 flashes the TM LED.

J1 may be used to connect potentiometers for use with analog control mode, or they may be connected to a radio control (R/C) receiver. The pin spacing at J1 is identical to the pin spacing used by R/C products and should be compatible with R/C jumpers.

The screw terminal J3 allows for quick connections to your motor(s) and the motor power supply.

Figure 2: Motor Mind C Carrier Board Schematic



4. Operating Information

4.1 Overview

The Motor Mind C Carrier Board may be used to connect to one or two brushed DC motors. The motor voltage should be limited to a range of 10-24VDC. Maximum current handling capability is 2.5A total (not per motor) when not cooled, in open air, at 25°C. Under the same conditions, but with the MMC_12VAC active cooling solution from Solutions Cubed, the board can handle 4.0A of continuous current (the PCB is equipped with mounting holes compatible with this fan kit).

A 5VDC linear regulator is provided on the Motor Mind C carrier board. This power supply can provide about 100mA to external circuitry including the BASIC Stamp 2 or an R/C receiver if used in R/C mode. Drawing more than 100mA may risk exceeding the power dissipation capability of the regulator. If extra current is necessary some method cooling the regulator should be included in your design.

4.2 Operating Mode Selection

The selection of the operating mode can be accomplished with the Mode Settings DIP switch (SW1) on the carrier board PCB. Mode selection should be done while the board is not powered. **Changes in the DIP switch settings should not occur while the carrier board is powered.** The Motor Mind C checks the settings defined by the DIP switch once shortly after power up and configures its I/O pins based on the DIP switch settings.

Some examples of different modes of operation are included here for informational purposes.

Figure 3: Example of an Analog Control Mode

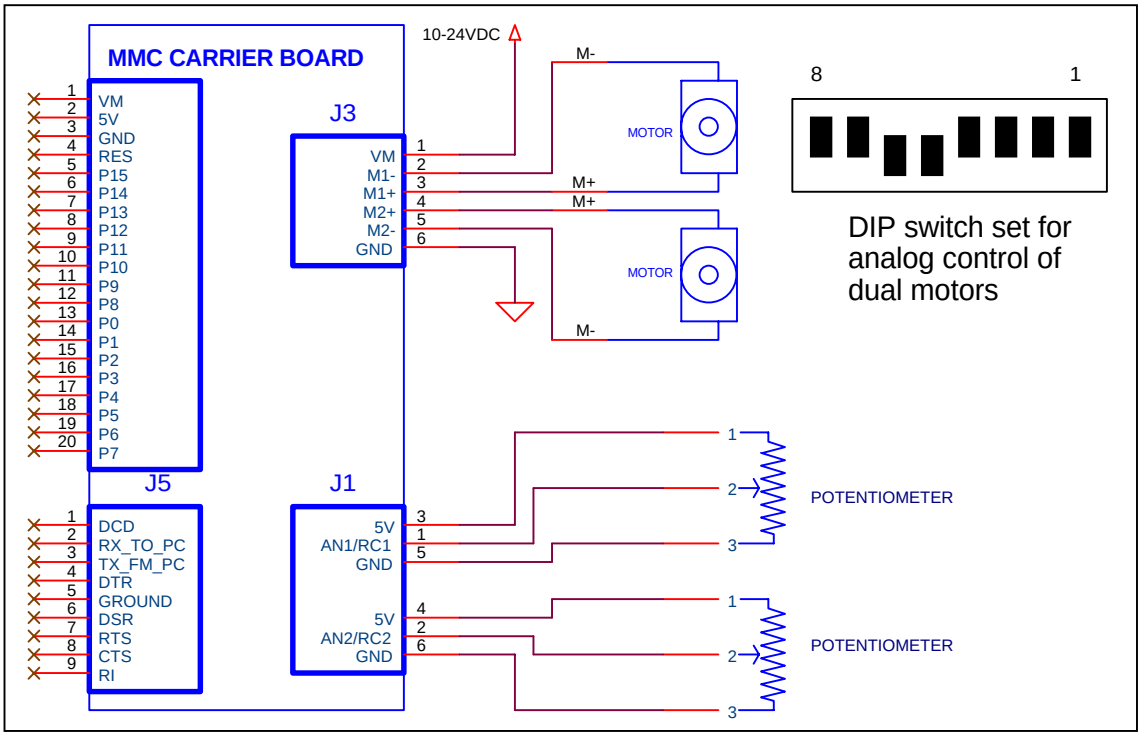
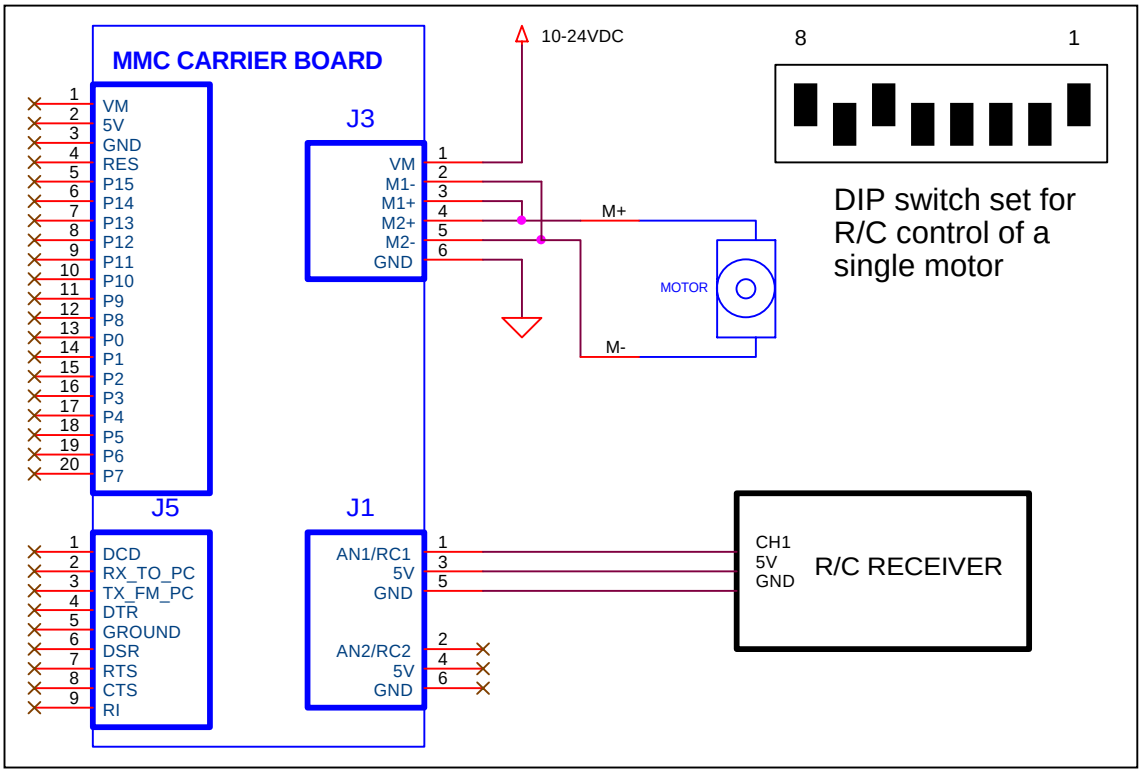


Figure 4: Example of a R/C Control Mode



4.2 AN700 – Interfacing the Motor Mind C to the BASIC Stamp 2

Application note 700 provides a software listing and connection diagram for use with the Motor Mind C Carrier Board. This software may be used with the BASIC Stamp 2, or may be modified slightly to work with other BASIC Stamps of the 24-pin variety.

This software provides two examples for methods of controlling DC motors with a BASIC Stamp 2.

Holding down the push-button associated with P14 (labeled “P15 button” on the REV2 PCB and “P14 button” on all higher revisions) while the device is powered up enters the first example of control software. The Motor Mind C will begin executing “skid-steering” routines. Routines similar to these may be used to control a robot drive system, such as the kind that can be implemented with the Easy Roller product from Solutions Cubed (PN: ER_M12V200_WHL1).

If the push-button is not pressed while powering up, the software defaults to a piece of code that allows motor speed to be controlled via the buttons connected to P15 and P14. One button increases speed, while the other will decrease it.

Figure 5: Connections for AN700

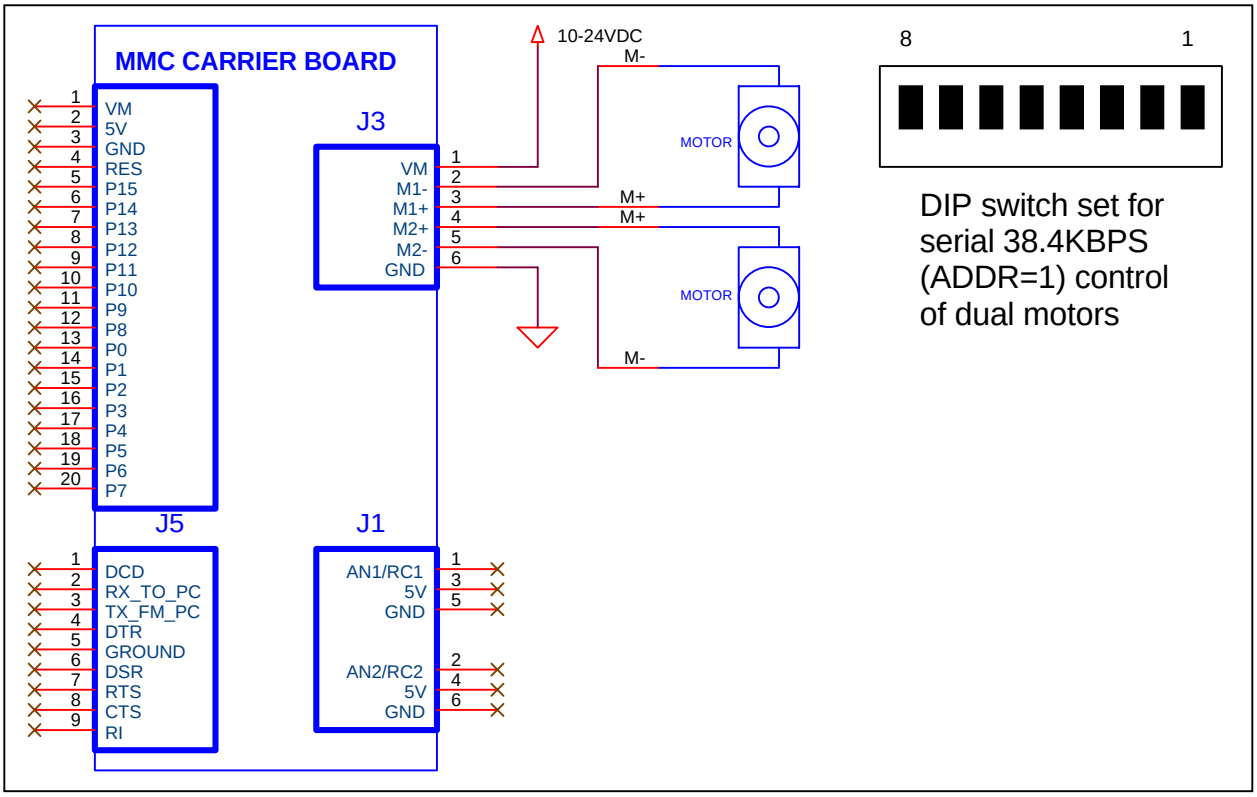


Figure 6: AN700 Software Listing

```
'AN700 Interfacing a BASIC Stamp 2 to the Motor Mind C Module

'Communication string variables
        CMMD            VAR     BYTE                    'Command byte storage
        ADDR            VAR     BYTE                    'Address byte storage
        LENG            VAR     BYTE                    'Length byte storage
        CKSUM           VAR     BYTE                    'Checksum byte storage
        DAT1            VAR     BYTE                    'Data byte registers
        DAT2            VAR     BYTE
        DAT3            VAR     BYTE
        DAT4            VAR     BYTE
        DAT5            VAR     BYTE
        DAT6            VAR     BYTE
        DAT7            VAR     BYTE
        DAT8            VAR     BYTE
        STAT            VAR     BYTE
        BTN_VAR VAR     BYTE                            'Generic variable for use with BUTTON cmd

'PWM storage registers
        PWM_REG1        VAR     WORD                    'PWM storage register for motor 1
                P1HI    VAR     PWM_REG1.HIGHBYTE
                P1LO    VAR     PWM_REG1.LOWBYTE
        PWM_REG2        VAR     WORD                    'PWM storage register for motor 2
                P2HI    VAR     PWM_REG2.HIGHBYTE
                P2LO    VAR     PWM_REG2.LOWBYTE

'PWM read registers
        RPWM_1          VAR     WORD                    'PWM storage register for motor 1
                RP1HI   VAR     RPWM_1.HIGHBYTE
                RP1LO   VAR     RPWM_1.LOWBYTE
        RPWM_2          VAR     WORD                    'PWM storage register for motor 2
                RP2HI   VAR     RPWM_2.HIGHBYTE
                RP2LO   VAR     RPWM_2.LOWBYTE

'Program constants
        BAUD            CON     45                      'Use BAUD = 45 for 38400 and BS2SX
'       BAUD            CON     6                       'Use BAUD = 6 for 38400BPS and BS2
'       BAUD            CON     84                      'Use BAUD = 84 for 9600BPS and BS2

'Motor Mind C communication lines
        TM              CON     7                       'TTL serial data from Motor Mind C
        FM              CON     8                       'TTL serial data to Motor Mind C


'Set BS2 i/o direction and level
        DIRS            =%0000000100000000              'Set P8 as output all others inputs
        OUTS            =%1111111111111111              'Set all outputs high
        PAUSE           1250                            'Wait 1250ms for MMC to power up
        DEBUG           CLS                             'Clear debug screen
        BTN_VAR  = $00

        IF IN14 = 1     THEN START_BUTTON_MODE


'*****************************************************************************************
'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 (in relation to chassis movement).
'*****************************************************************************************

START_SKID_STEERING:

'FULL-FORWARD - drives both motors at top speed pulling the robot forward
        PWM_REG1        = 1023
        PWM_REG2        = -1023
        GOSUB           SETDC
        PAUSE           2000
```

```
'LEFT-FORWARD - drives motor1 at half speed and motor 2 at full speed causing chassis
'                       to veer left
            PWM_REG1        = 512
            PWM_REG2        = -1023
            GOSUB           SETDC
            PAUSE           2000

'RIGHT-FORWARD - drives motor1 at full speed and motor 2 at half speed causing chassis
'                       to veer right
            PWM_REG1        = 1023
            PWM_REG2        = -512
            GOSUB           SETDC
            PAUSE           2000

'CREEP-FORWARD - drives both motors at slow speed allowing the chassis to creep forward
            PWM_REG1        = 100
            PWM_REG2        = -100
            GOSUB           SETDC
            PAUSE           2000

'CREEP-PIVOT - drives both motors slow with motor 2 reversed allowing the chassis pivot slowly
            PWM_REG1        = 100
            PWM_REG2        = 100
            GOSUB           SETDC
            PAUSE           2000

'FAST-PIVOT - drives both motors at full speed with motor 2 reversed allowing the chassis spin
            PWM_REG1        = 1023
            PWM_REG2        = 1023
            GOSUB           SETDC
            PAUSE           2000

'STOP - stops both motors
            PWM_REG1        = 0
            PWM_REG2        = 0
            GOSUB           SETDC
            PAUSE           1000

            GOTO            START_SKID_STEERING


'*****************************************************************************************
'START_BUTTON_MODE:      This mode of operation monitors the momentary pushbuttons
'                       SW3 and SW2.  If SW2 is pressed then the motor speed is increased by 8
'                       to a maximum of +1016 (about 99% moving forward).  If SW3 is pressed then
'                       the motor speed is decreased by 8 to a minimum of -1016 (about 99% moving
'                       reversed).  I
'*****************************************************************************************

START_BUTTON_MODE:
            BUTTON          15,0,5,1,BTN_VAR,0,No_Press_SW2
            PWM_REG1        = PWM_REG1 + 8
            IF              PWM_REG1 <> 1024 THEN NO_Limit_PWM_Pos
            PWM_REG1        = 1016
NO_Limit_PWM_Pos
            PWM_REG2        = PWM_REG1
            GOSUB           SETDC

No_Press_SW2:
            BUTTON          14,0,5,1,BTN_VAR,0,No_Press_SW3
            PWM_REG1        = PWM_REG1 - 8
            IF              PWM_REG1 <> -1024 THEN NO_Limit_PWM_Neg
            PWM_REG1        = -1016
NO_Limit_PWM_Neg
            PWM_REG2        = PWM_REG1
            GOSUB           SETDC

No_Press_SW3:
            GOTO            START_BUTTON_MODE                   'Return to start of program
```

```
'************************************ Subroutines ***************************************

'****************************************************************************************
'SETDC:    This routine sends speed and direction data to the Motor Mind C.  The values in the PWM1_REG and PWM2_REG are sent
'          via the SetDC command.
'****************************************************************************************

SETDC:
' The debug statements below can be used to better understand how the decimal numbers
'
'         DEBUG           "PWM_REG1 =",ISHEX4 PWM_REG1,"; HI BYTE-",ISHEX2 P1HI," LO BYTE-",ISHEX2 P1LO,CR
'         DEBUG           "PWM_REG2 =",ISHEX4 PWM_REG2,"; HI BYTE-",ISHEX2 P2HI," LO BYTE-",ISHEX2 P2LO,CR
          CMMD            = $D0                                                   'SETDC command
          ADDR            = $01                                                   'MMC default address of "1"
          LENG            = $04                                                   'Length of SETDC is 4
          CKSUM           = CMMD+ADDR+LENG+P1HI+P1LO+P2HI+P2LO
          SEROUT          FM,BAUD,[CMMD,ADDR,LENG,P1HI,P1LO,P2HI,P2LO,CKSUM]
          SERIN           TM,BAUD,150,NA_SDC1,[DAT1]
          IF              DAT1 <> $6 THEN NA_SDC1
          RETURN
NA_SDC1:
          DEBUG           "SETDC ERROR",CR
          RETURN

'****************************************************************************************
'READ_REGS1: Reads and displays values stored in registers 0,2,3,4, and
'****************************************************************************************
READ_REGS1:
          CMMD            = $D1
          ADDR            = $01
          LENG            = $05
          DAT1            = $00
          DAT2            = $02
          DAT3            = $03
          DAT4            = $04
          DAT5            = $05

          CKSUM           = CMMD+ADDR+LENG+DAT1+DAT2+DAT3+DAT4+DAT5
          SEROUT          FM,BAUD,[CMMD,ADDR,LENG,DAT1,DAT2,DAT3,DAT4,DAT5,CKSUM]
          SERIN           TM,BAUD,150,NA_RDST1,[DAT1,DAT2,STAT,RP1HI,RP1LO,RP2HI,RP2LO,DAT8]
          DEBUG           CLS,"STATUS =",BIN8 STAT,CR
          DEBUG           "PWM1 =",ISHEX4 RPWM_1,CR
          DEBUG           "PWM2 =",ISHEX4 RPWM_2,CR
          GOTO            DONE_RDST1
NA_RDST1:
          DEBUG           "NO RESPONSE TO READ 1 ",CR
DONE_RDST1:
          RETURN

'****************************************************************************************
'READ_REGS2:  Reads and displays values stored in registers 1,6,7,8,9,and 10 (A in hexadecimal)
'****************************************************************************************
READ_REGS2:
          CMMD            = $D1
          ADDR            = $01
          LENG            = $06
          DAT1            = $01
          DAT2            = $06
          DAT3            = $07
          DAT4            = $08
          DAT5            = $09
          DAT6            = $0A

          CKSUM           = CMMD+ADDR+LENG+DAT1+DAT2+DAT3+DAT4+DAT5+DAT6
          SEROUT          FM,BAUD,[CMMD,ADDR,LENG,DAT1,DAT2,DAT3,DAT4,DAT5,DAT6,CKSUM]
          SERIN           TM,BAUD,150,NA_RDST2,[DAT1,DAT2,DAT3,DAT4,DAT5,DAT6,DAT7,DAT8,CKSUM]
          DEBUG           "FIRMWARE =",ISHEX2 DAT3,CR
          DEBUG           "AMPS1 =",ISHEX2 DAT4,CR
          DEBUG           "AMPS2 =",ISHEX2 DAT5,CR
          DEBUG           "DEAD_BAND =",ISHEX2 DAT6,CR
          DEBUG           "PWM_STEP =",ISHEX2 DAT7,CR
          DEBUG           "BRAKE_MODE =",ISHEX2 DAT8,CR
          GOTO            DONE_RDST2
NA_RDST2:
          DEBUG           "NO RESPONSE TO READ 2 ",CR
DONE_RDST2:
          RETURN
```

```
'*********************************************************************************
'WRITE_REGS1:  Writes data to registers 8,9, and 10 (A in hexadecimal)
'*********************************************************************************
WRITE_REGS1:
        CMMD            = $D2
        ADDR            = $01
        LENG            = $06
        DAT1            = $08
        DAT2            = $01
        DAT3            = $09
        DAT4            = $80
        DAT5            = $0A
        DAT6            = $00

        CKSUM           = CMMD+ADDR+LENG+DAT1+DAT2+DAT3+DAT4+DAT5+DAT6
        SEROUT          FM,BAUD,[CMMD,ADDR,LENG,DAT1,DAT2,DAT3,DAT4,DAT5,DAT6,CKSUM]
        SERIN           TM,BAUD,150,NA_WRST1,[DAT1]
        IF              DAT1 = $6 THEN DONE_WRST1
NA_WRST1:
        DEBUG           "NO RESPONSE TO WRITE1 ",CR
DONE_WRST1:
        RETURN


END:
```

Disclaimer of Liability and Accuracy: Information provided by Solutions Cubed is believed to be accurate and reliable. However, Solutions Cubed assumes no responsibility for inaccuracies or omissions. Solutions Cubed assumes no responsibility for the use of this information and all use of such information shall be entirely at the user's own risk.

Life Support Policy: Solutions Cubed does not authorize any Solutions Cubed product for use in life support devices and/or systems without express written approval from Solutions Cubed.

Warrantee: Solutions Cubed warrants all Motor Mind C Carrier Board Modules against defects in materials and workmanship for a period of 90 days. If you discover a defect, we will, at our option, repair or replace your product or refund your purchase price. This warrantee does not cover products that have been physically abused or misused in any way.