

Overview:

This application note describes how to control the Motor Mind C (MMC) with a Parallax BASIC Stamp 2 (BS2) using the MMC's SetDC command as well as how to detect and clear fault conditions. Commands are packets consisting of numerical bytes of information sent serially to the MMC using the BS2 Serin and Serout instructions. These numerical bytes can be in hexadecimal, decimal or binary form.

Command Syntax:

*Note that to the BS2, D0 hex is exactly the same as 208 decimal or 11010000 binary. Thus, the command:

```
SEROUT FM,BAUD,[$55,$03,$00,$0A,$62]
```

is equivalent to:

```
SEROUT FM,BAUD,[85,3,0,10,98]
```

1. SetDC command:

The syntax of the packet required to implement the SetDC command is as follows:

Command Packet Byte	Tells MMC which command is being sent. For the SetDC command, this value will be 208 (decimal) or D0 (hex)
Address of Destination Byte	Address of the MMC [1 (decimal or hex) (or 2 if ADDR pin is tied to ground)]
Length of Data Byte	This is the number of bytes of data following the length byte, but preceding the Checksum byte. For the SetDC command, this value will be 4 (decimal or hex).
Value to load into PWM1_HI reg	0x00 to 0x03 hexadecimal
Value to load into PWM1_LO reg	0x00 to 0xFF hexadecimal
Value to load into PWM2_HI reg	0x00 to 0x03 hexadecimal
Value to load into PWM2_LO reg	0x00 to 0xFF hexadecimal

The values loaded into the PWM registers will control both the speed and the direction of the motors. Positive values correspond to forward rotation, while negative values correspond to reverse rotation. Higher magnitude numbers (i.e. Decimal 1000 or -1000) correspond to higher speed, while lower magnitude numbers (i.e. Decimal 100 or -100) correspond to slower speeds.

2. Detecting a Fault Condition

A fault condition can occur if too high of temperature or current is detected in either H-bridge. This causes a bit to be set in the MMC status register. The MMC status register can be read using the MMC Read command. The syntax is as follows:

Command Packet Byte	Tells MMC which command is being sent. For the Read command, this value will be 209 (decimal) or D1 (hex)
Address of Destination Byte	Address of the MMC [1 (decimal or hex) (or 2 if ADDR pin is tied to ground)]
Length of Data Byte	This is the number of bytes of data following the length byte, but preceding the Checksum byte. For reading the MMC status register, this value will be 1 (decimal or hex).
Register number	This is the number of the register to be read from. For the MMC status register, this would be 0.

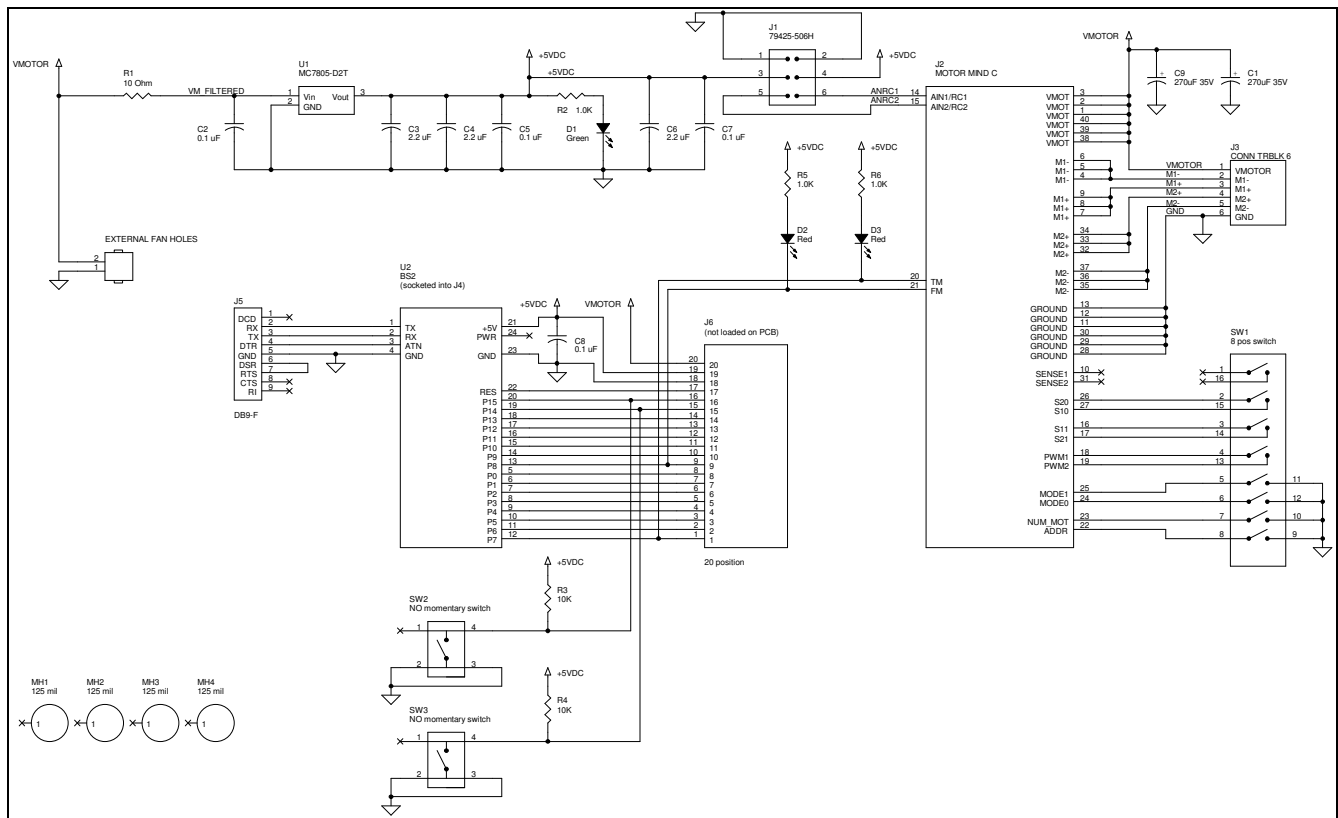
3. Clearing a Fault Condition

If a fault occurs, it will need to be cleared before normal function can resume. The syntax of the Clear Fault command is as follows:

Command Packet Byte	Tells MMC which command is being sent. For the Clear Fault command, this value will be 211 (decimal) or D3 (hex)
Address of Destination Byte	Address of the MMC [1 (decimal or hex) (or 2 if ADDR pin is tied to ground)]
Length of Data Byte	This is the number of bytes of data following the length byte, but preceding the Checksum byte. There will be no data sent, so the this will be 0.

Schematic:

This application note was written while using the MMC_BS2 carrier board for the hardware. Here is the schematic for that part.



Source Code:

'Communication string variables

CMMD	VAR	BYTE	'Command byte storage
ADDR	VAR	BYTE	'Address byte storage
LENG	VAR	BYTE	'Length byte storage
REG	VAR	BYTE	'Register byte storage
CKSUM	VAR	BYTE	'Checksum byte storage
DAT1	VAR	BYTE	'Data storage
DAT2	VAR	BYTE	'Data storage
STAT	VAR	BYTE	'Status register byte storage

'PWM storage registers

PWM_REG1	VAR	WORD	
P1HI	VAR	PWM_REG1.HIGHBYTE	
P1LO	VAR	PWM_REG1.LOWBYTE	
PWM_REG2	VAR	WORD	
P2HI	VAR	PWM_REG2.HIGHBYTE	
P2LO	VAR	PWM_REG2.LOWBYTE	

'Program constants

'	BAUD	CON	240	'Use BAUD = 240 for 9600 and BS2SX
'	BAUD	CON	84	'Use BAUD = 84 for 9600 and BS2
'	BAUD	CON	45	'Use BAUD = 45 for 38400 and BS2SX

'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 pin 8 as output, all others as inputs
OUTS	=%1111111111111111	'Set all outputs high
PAUSE	1250	'Wait 1250ms for MMC to power up.
DEBUG	CLS	'Clear debug screen
DAT2	=2	
GOTO	RAMP_MODE	

```

*****
'RAMP_MODE: This mode of operation continuously ramps the speed of the motors up
'           to approximately 99% of full speed, then ramps back down to a full stop.
'           **NOTE: To test for a fault condition, uncomment the CHECK_STAT statements
*****

```

RAMP_MODE:

Ramp_Up:

```

GOSUB CHECK_STAT
PWM_REG1 = PWM_REG1 + 8
PWM_REG2 = PWM_REG2 - 8
GOSUB SETDC
PAUSE 50
IF PWM_REG1 < 1024 THEN Ramp_Up

```

Ramp_Down:

```

GOSUB CHECK_STAT
PWM_REG1 = PWM_REG1 - 8
PWM_REG2 = PWM_REG2 + 8
GOSUB SETDC
PAUSE 50
IF PWM_REG1 > 0 THEN Ramp_Down
GOTO RAMP_MODE

```

END_RAMP_MODE

*****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 MMC's SetDC command.
*****

```

SETDC:

'The debug statements below can be used to better understand how the decimal numbers correspond to the speed and direction of the motors.

```

'   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

```

```
*****
'CHECK_STAT:This routine reads the contents of the status register.  If one or both of the
'          motor fault bits are set to 1, the FAULT subroutine is called in an attempt to
'          clear the fault.
*****
```

```
CHECK_STAT:
  CMMD      = $D1
  ADDR      = $01
  LENG      = $01
  REG       = $00
  CKSUM     = CMMD+ADDR+LENG+REG
  SEROUT    FM,BAUD,[CMMD,ADDR,LENG,REG,CKSUM]
  SERIN     TM,BAUD,150,CKST_ERR,[ADDR,LENG,STAT,CKSUM]
  IF       STAT & $30 = 0 THEN END_CKS
  GOSUB     FAULT
```

```
CKST_ERR:
  DEBUG     "CHECK STAT ERROR",CR
END_CKS:
  DEBUG     "Returning Stat:",ISBIN8 STAT,CR
  RETURN
```

```
*****
'FAULT:      This routine sends the Clear Fault command to the MMC, which turns the motor
'            associated with the fault condition off by setting the associated PMW to 0x0000
'            and attempts to clear the fault condition.
*****
```

```
FAULT:
  CMMD      = $D3
  ADDR      = $01
  LENG      = $00
  CKSUM     = CMMD+ADDR+LENG
  SEROUT    FM,BAUD,[CMMD,ADDR,LENG,CKSUM]
  DEBUG     "FAULT CONDITON",CR
  PAUSE    3000
RETURN
```