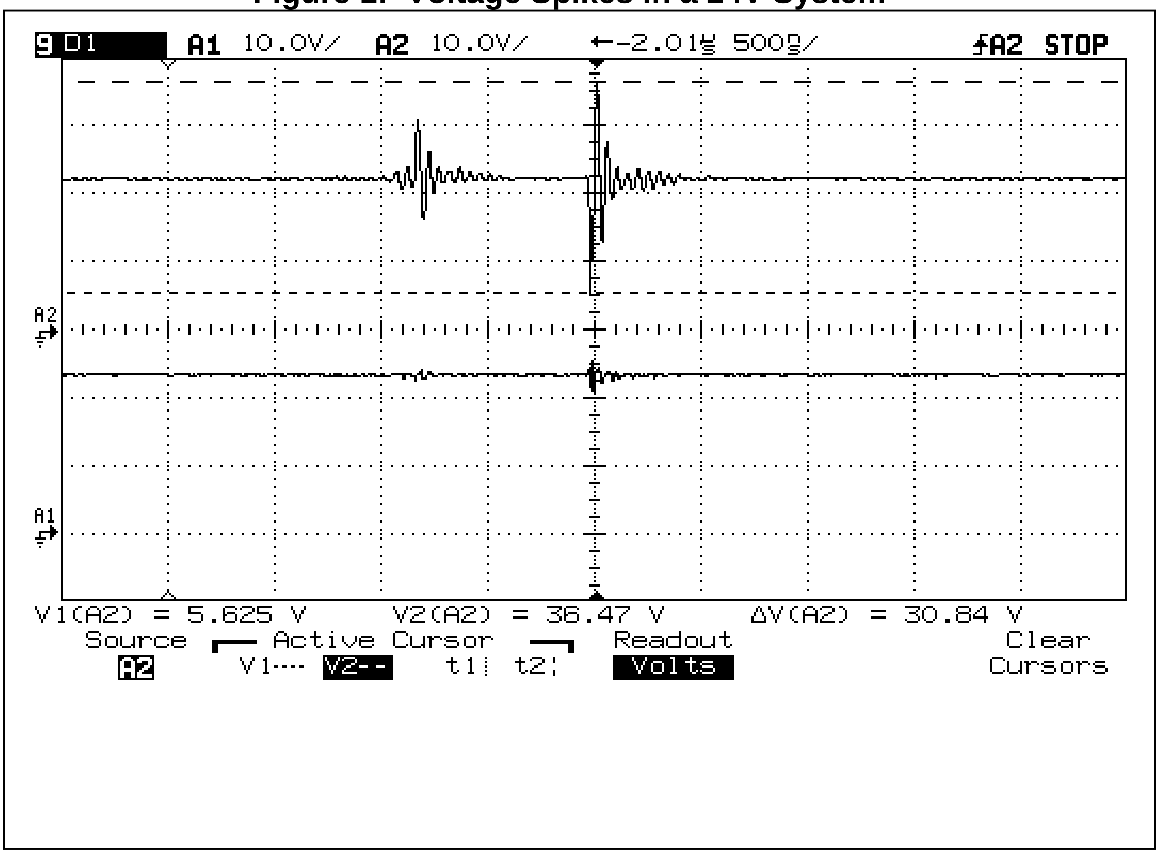
Overview:

The Motor Mind C (MMC) has is rated for a peak motor voltage of 30VDC. Operation in 24V systems can cause voltage spikes that easily exceed the 30V rating. The device was designed with a safety margin that should allow operation in 24V systems if some protective circuitry is included in any design utilizing the MMC.

Operating Voltage Determination:

The 5V linear regulator designed into the module limits the operating voltage of the MMC. This regulator is rated for 35V peak voltage; the H-bridges on the MMC are rated for 40V, and should present less of a problem for 24V systems. Therefore the primary area requiring protection is the linear regulator. The following oscilloscope measurement was taken with the MMC operating in analog mode, using one of our Easy Roller 12V motors. The motor voltage was supplied by a 24V 30A power supply, and the motor direction was repeatedly reversed using a potentiometer. Channel A2 was taken from the motor's positive lead during a reversal, while channel A1 shows the VM pin on the MMC (after the protection circuitry).

Figure 1: Voltage Spikes in a 24V System



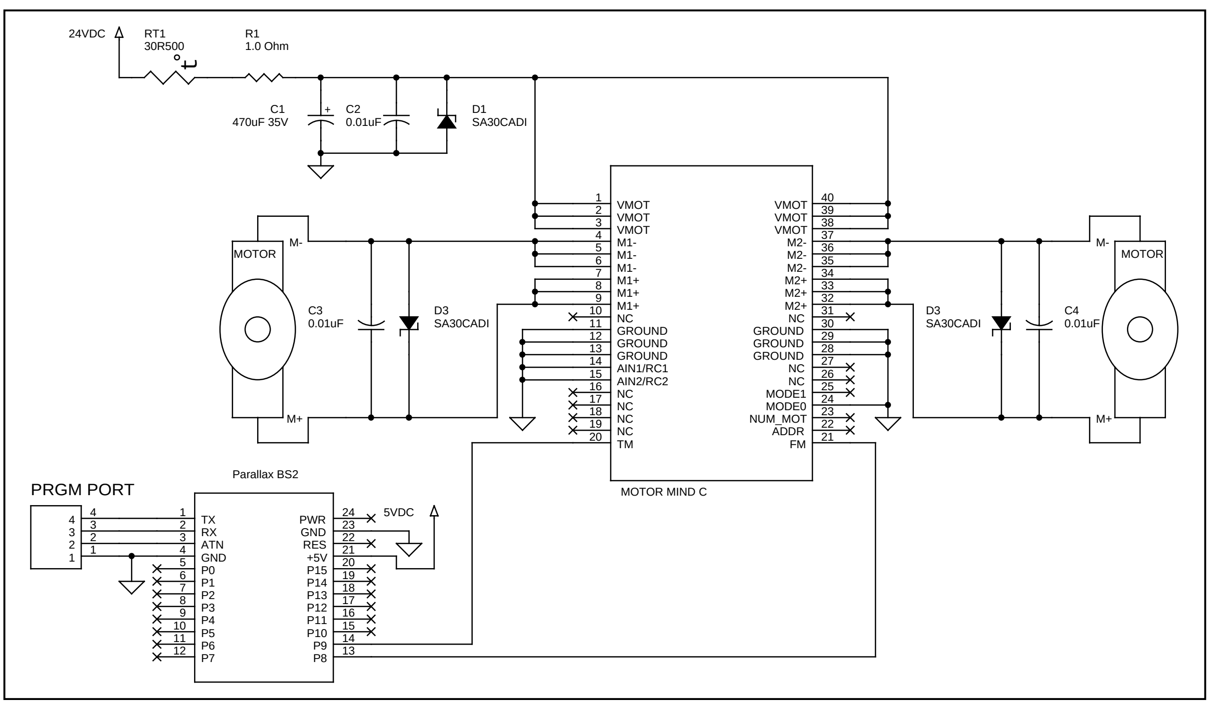
Protection Circuitry:

The circuitry described here provides protection from two sources of potential problems; over-current conditions, and voltage spikes. The components referenced in this section are those shown in the application note schematic below. Quick changes in motor speed and especially in direction are the primary cause of transients that can damage the MMC. The likelihood of dangerous transients occurring increases as the system approaches the maximum parameters defined in the MMC datasheet.

Over-Current Protection: RT1 is a resettable fuse rated for 5A and 30VDC (Raychem/Tyco is one manufacturer). These fuses are somewhat slow to act, but do not need to be replaced when tripped. As current through the fuse increases, the fuse heats up, and presents an increased resistance in the circuit. When the fuse “trip” current is reached it will begin starving the circuit for current and reducing the motor voltage and motor current. As the motor current decreases the fuse will cool and present a smaller resistance in the circuit. C1 provides operating current for brief periods, and is required with or without protection circuitry.

Voltage Spikes: D1, D2, and D3 are 30V bi-directional transient-voltage–suppressors (TVS). The TVS acts as a very fast zener diode, shorting to ground when it’s “trip” voltage is exceeded. RT1 and R1 act to damp voltage spikes at the MMC VM pin, and reduce currents through D1-3 should a TVS become active. C2, C3, and C4 act as short circuits to high frequency noise that may accompany transient voltage spikes.

Figure 2: Protective Circuit Components



Protective Parts – Can Be Purchased From www.digi-key.com

Part	Description	Part Number	Manufacturer	Notes
C1	470uF 35V 1.2A	EEU-FC1V471	Panasonic	Electrolytic cap w/ high ripple current rating
C2, C3, C4	0.01uF 50V cap	Any	Any	Ceramic cap
D1, D2, D3	30V BI-DIR TVS	PKE30CA	Diodes Inc.	Breakdown voltage between 28-32VDC
R1	1 ohm 2W resistor	1.0W-2-ND	Yageo	Higher or lower wattage resistors may be necessary based on your system’s current draw
RT1	5A 30V PTC fuse	RUE500	Raychem	Lower current rated fuses should be used if the motors used are smaller

Software Solutions:

When operating in analog or R/C mode there are limitations to the speed changes that the MMC will execute. In these modes the rate of change in speed is limited by the time required to measure the control signal, and software routines limit the change in speed from the previous measurement. This is not the case with serial mode operation. In serial mode it is possible to command the MMC to go from full-speed forward to full-speed reverse within milliseconds.

Therefore, when operating in serial control mode some common sense software design can alleviate problems associated with transient signals. One of the easiest methods of reducing transient voltage signals is to ramp motor speed changes up and down. The following BASIC Stamp 2 software listing can be used with the schematic in figure 2 to ease a motor into higher or lower speeds (and through direction reversals). Similar methods of motor speed ramping could be used with more complex controllers.

Some important considerations regarding the Parallax BASIC Stamp2 ramping code should be addressed here. First, make sure that the step size used by the FOR…NEXT loop in your code can be divided evenly into the value you use to test for the end of the ramp (the IF…THEN statement). In this example the motor is ramped up in steps of 16 until a value of 1008 is reached. The step size of the ramp, and delay between commands, may all be adjusted to make the motor ramp up and down faster. The values used here should allow for safe motor speed changes in all but the most borderline of systems.

AN703.BS2 Listing – Ramping Motor Speeds

```
'AN703 Protecting the MMC with 24V Systems
'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

'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     84                      'Use BAUD = 84 for 9600BPS and BS2

'Motor Mind C communication lines
        TM              CON     9                       '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            =%0000000000000000              'Set all as inputs
        OUTS            =%1111111111111111              'Set all outputs high
        PAUSE           1250                            'Wait 1250ms for MMC to power up
        PWM_REG1        =0
        PWM_REG2        =0
```

```
START

Ramp_Up1:                                                         'Ramp up forward
            PAUSE           10                                    'Adjust or remove pause based on requirements
            PWM_REG1        = PWM_REG1 + 16                       'Adjust step size based on requirements
            PWM_REG2        =0                                    'Step must divide evenly with IF...THEN test value
            GOSUB           SETDC
            IF PWM_REG1 <> 1008 THEN Ramp_Up1                     '1008/16 is even value so OK
Ramp_Down1                                                        'Ramp from full forward to full reverse
            PAUSE           10
            PWM_REG1        = PWM_REG1 - 16
            PWM_REG2        =0
            GOSUB           SETDC
            IF PWM_REG1 <> -1008 THEN Ramp_Down1
Back_Up1
            PAUSE           10
            PWM_REG1        = PWM_REG1 + 16                       'Ramp from full reverse back to stop
            PWM_REG2        =0
            GOSUB           SETDC
            IF PWM_REG1 <> 0 THEN Back_Up1


Ramp_Up2:
            PAUSE           10                                    'Have motor 2 copy motor 1 movements
            PWM_REG2        = PWM_REG2 + 16
            PWM_REG1        =0
            GOSUB           SETDC
            IF PWM_REG2 <> 1008 THEN Ramp_Up2
Ramp_Down2
            PAUSE           10
            PWM_REG2        = PWM_REG2 - 16
            PWM_REG1        =0
            GOSUB           SETDC
            IF PWM_REG2 <> -1008 THEN Ramp_Down2
Back_Up2
            PAUSE           10
            PWM_REG2        = PWM_REG2 + 16
            PWM_REG1        =0
            GOSUB           SETDC
            IF PWM_REG2 <> 0 THEN Back_Up2


'********************************** 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:
            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                            'Expect ACK in return
            RETURN
NA_SDC1:
            RETURN

END:
```

Conclusion: With adequate circuitry, and well-designed software subroutines, transient signals in 24V systems can be reduced to a point where they do not pose a hazard to the MMC. Component selection accompanied with testing can determine whether your system requires additional components for protection.

Test your system with lower motor voltages and software speed control, where possible. Then move to higher motor voltages. This can help identify functionality errors and problematic conditions prior to the actual system test. When approaching the MMC's maximum specifications be prepared for transient signals that exceed those specifications, and plan ahead to alleviate them.