# APPENDIX 1

## PMSM DATASHEET

### BRUSHLESS SERVOMOTORS

#### SERIES

**TETRA 85SR2.2**

**TORQUE**

![Torque Graph]

**Torque**

### SINEWAVE FORM

<table>
<thead>
<tr>
<th>SYMBOLS</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vn</td>
<td>[V ac]</td>
</tr>
<tr>
<td>Ih</td>
<td>[A]</td>
</tr>
<tr>
<td>Pr</td>
<td>[W]</td>
</tr>
<tr>
<td>Iq</td>
<td>[A]</td>
</tr>
<tr>
<td>PWM</td>
<td>[Hz]</td>
</tr>
</tbody>
</table>

### WINDING DATA

<table>
<thead>
<tr>
<th>Motor rpm</th>
<th>P</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vn drive 3-phase 45 V ac</td>
<td>0000</td>
<td>5000</td>
<td>10000</td>
<td>2000</td>
<td>0000</td>
<td>5000</td>
<td>10000</td>
<td>2000</td>
<td>0000</td>
<td>5000</td>
<td>10000</td>
<td>2000</td>
<td>0000</td>
<td>5000</td>
<td>10000</td>
<td>2000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vn drive 3-phase 145 V ac</td>
<td>0000</td>
<td>5000</td>
<td>10000</td>
<td>2000</td>
<td>0000</td>
<td>5000</td>
<td>10000</td>
<td>2000</td>
<td>0000</td>
<td>5000</td>
<td>10000</td>
<td>2000</td>
<td>0000</td>
<td>5000</td>
<td>10000</td>
<td>2000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vn drive 3-phase 220 V ac</td>
<td>0000</td>
<td>5000</td>
<td>10000</td>
<td>2000</td>
<td>0000</td>
<td>5000</td>
<td>10000</td>
<td>2000</td>
<td>0000</td>
<td>5000</td>
<td>10000</td>
<td>2000</td>
<td>0000</td>
<td>5000</td>
<td>10000</td>
<td>2000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SERVOMOTOR

- **Continuous output torque (Nm)**: 2.0 Nm
- **Voltage constant ± 5% (Nm/V)**: 0.23 Nm/V
- **Torque constant ± 5% (Nm/W)**: 0.23 Nm/W
- **Hall current (A)**: 0.11 A
- **Peak current (A)**: 1.16 A
- **Maximum current (A)**: 2.8 A
- **Phase resistance at 25°C (Ohm)**: 1.2 Ohm
- **Thermal time constant (s)**: 20 s
- **Protection degree (IP)**: 40 IP
- **Weight (Kg)**: 1.5 Kg

### THERMAL P

- **Type of thermal cut-off**: N C: normally closed
- **Rated current (A)**: 2.5 A
- **Operative temperature (°C)**: 140 °C ± 5°C
- **Reseting temperature (°C)**: 100 °C + 15°C
- **Insulation classes (F)**: F

### BRAKE

- **Type**: STD 4.5
- **Static torque (Nm)**: 4.5 Nm
- **Rated torque (Nm)**: 0.5 Nm
- **Rated current (A)**: 0.5 A
- **Engaging time (ms)**: 7 ms
- **Release time (ms)**: 35 ms

(*) with oil seal mounted on the flanges

DATA SHEET N° 18/1004/100006

Motor Power Company s.r.l. - Reggio Emilia - Italy Tel. +39 0522 682719 - Fax 0522 683552
APPENDIX 2

SPARTAN-3A DSP FPGA CODING

-- Module Name : Behavioral
-- Project Name : TRM using FLC with SVPWM Based FOC
-- Target Devices: Spartan3A-DSP
-- Tool versions : ISE 13.2
-- Dependencies:
-------------------------------------------------------------------------------------------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_unsigned.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity pmsm_vector_control is
port(clk: in std_logic;
cap1: in std_logic;
cap2: in std_logic;
oe1: out std_logic;
oe6: out std_logic;
dir1: out std_logic;
dir6: out std_logic;
--

-- led: out std_logic_vector(16 downto 1);
swt1,swt5: in std_logic;
swt2,swt3: in std_logic;
swt4,swt6,swt7: in std_logic;
cs : out std_logic;
diow : out std_logic;
rs : out std_logic;
data : out std_logic_vector(7 downto 0);
pulse1: out std_logic;
pulse2: out std_logic;
pulse3: out std_logic;
pulse4: out std_logic;
pulse5: out std_logic;
pulse6: out std_logic;
int Kp, xk, DT;
float kp = 0.1, ki = 0.1, kpq = 0.1, kpd = 0.1, kiq = 0.1, kid = 0.1;
int Const1, Const2;
//extern int setspeed, speed;
extern int Error, T1, T2, Prop, Integral, Speed1, counterr, Filt_iq_m1;
extern int Var1, Var2, Var3, Var4, Var5, VD1, VD2, VD3, VD4;
extern int Iq, Iq_m1, Filt_iq, omega_mr, Rotor_Speed, Slip_Speed, VD, VQ, Degree;
extern int Vq, Vd, Vqref, Vdref, Va, Vb, Vc, w1, Vq1, Vd1, V_alfa, V_beta;
unsigned int Index;
extern int Sin_Theta, Cos_Theta, Te1, Te2;
extern float VA, VB, VC; // ia, ib, ic, i_alfa, i_beta, id, iq;
extern int V_ALFA, V_BETA;
extern int COMPARE1, COMPARE2, COMPARE3;
int setspeed = 300, setcurrentq, setcurrentd = 0, speed, lcd, error, feedback, duty, prop,
 integral, lcd1, error1, feedback1, dutyq, prop1, integral1, lcd2, error2, feedback2, dutyd, prop2, integral2;
extern long int temp;
int duty1, duty2;
extern float iq, id;
extern float torque;
------------
ADC1
busy : in std_logic;
douta : in std_logic; -- serial data out
doutb : in std_logic; -- serial data out
adccs : out std_logic; -- chip select
adcsclk : out std_logic;
addr : out std_logic;
range0 : out std_logic;
cnvst : out std_logic;
led : out std_logic_vector(11 downto 0);
------------
ADC2
busy1 : in std_logic;
douta1 : in std_logic; -- serial data out
doutb1 : in std_logic; -- serial data out
adcs1 : out std_logic; -- chip select
adcsclk1 : out std_logic;
addr1 : out std_logic;
cnvst1 : out std_logic);
end pmsm_vector_control;
architecture Behavioral of pmsm_vector_control is
component PI_CONTROLE1 is
port(clk : in std_logic;
speed : in integer;
fd_bk : in integer;
output : out integer);
end component;
component QEP is
  port(clk : in std_logic;
       sen_a : in std_logic;
       rpm : out integer);
end component;
component lcd_sr is
  Port (clock : in std_logic; -- 20 mega hz;
        swtch1 : in std_logic;
        swtch5 : in std_logic;
        swtch2 : in std_logic;
        swtch6 : in std_logic;
        swtch3 : in std_logic;
        swtch7 : in std_logic;
        d1 : in integer;
        d2 : in integer;
        -- Lcd Control Pins
        cs : out std_logic;
        diow : out std_logic;
        rs : out std_logic;
        data : out std_logic_vector(7 downto 0);
        sp_out : out integer;
        act_rpm : in integer;
        kp_out : in integer);
  --adc_in : in std_logic_vector(11 downto 0));
end component;
component fuzzy is
  port(clk: in std_logic;
      set: in integer;
      act: in integer;
      output: out integer);
end component;
component PI_CONTROLE2 is
  port(clk : in std_logic;
       speed : in integer;
       fd_bk : in integer;
       output : out integer);
end component;
component PI_CONTROLE3 is
port(clk : in std_logic;
speed : in integer;
fd_bk : in integer;
output : out integer);
end component;
component adc_testing is
port(clk : in std_logic;
--------------
ADC1
busy : in std_logic;  -- serial data out
douta : in std_logic;  -- serial data out
doutb : in std_logic;  -- serial data out
adccs : out std_logic;  -- chip select
adcsclk : out std_logic;
addr : out std_logic;
range0 : out std_logic;
cnvst : out std_logic;
led : out std_logic_vector(11 downto 0);
--------------
ADC2
busy1 : in std_logic;
douta1 : in std_logic;  -- serial data out
doutb1 : in std_logic;  -- serial data out
adccs1 : out std_logic;  -- chip select
adcsclk1 : out std_logic;
addr1 : out std_logic;
cnvst1 : out std_logic;
CH1X : OUT INTEGER;
CH2X : OUT INTEGER;
CH3X : OUT INTEGER;
CH4X : OUT INTEGER;
CH5X : OUT INTEGER;
CH6X : OUT INTEGER;
CH7X : OUT INTEGER;
CH8X : OUT INTEGER);
end component;
signal flag1: std_logic:= '0';
signal flag2: std_logic:= '0';
signal max, min, tri, car1, car2, car3: integer:= 0;
signal f: std_logic:= '0';
type trig is array (1 to 1000)of integer;
--signal i: integer:= 1;
signal del: integer:= 0;
signal sinetheta: integer:= 0;
signal sinetheta_120: integer:= 0;
signal sinetheta_240: integer:=0;
signal cosetheta: integer:=0;
SIGNAL ADC1,ADC2,adc3,adc4,adc5,adc6,adc7,adc8: INTEGER:=0;
SIGNAL IA,IB: INTEGER:=0;
SIGNAL IQ, ID: INTEGER:=0;
SIGNAL ph_a: INTEGER:=0;
SIGNAL ph_b: INTEGER:=0;
SIGNAL SINT: INTEGER:=0;
SIGNAL COST: INTEGER:=0;
--signal i2: integer:=0;
signal va,vb,vc: integer:=0;
SIGNAL IALPHA,IBETA: INTEGER:=0;
signal iq_ref,iq_out: integer:=0;
signal id_ref,id_out: integer:=0;
signal w, w_ref, w_out: integer:=0;
component getspeed_disp is
Port ( clock  : in std_logic; -- 20 mega hz;
swtch1 : in std_logic;
swtch5 : in std_logic;
-- Lcd Control Pins
cs       : out std_logic;
diow  : out std_logic;
rs       : out std_logic;
data   : out std_logic_vector(7 downto 0);
Rpm_get : out integer; -- getspeed & disply that value
pv  : in integer);
end component;
signal led_temp: std_logic_vector(16 downto 1);
signal car : integer:=1000;
signal count : integer:=0;
signal setrpm,rpm: integer:=0;
signal adc: integer:=0;
--signal i: integer:=1;
--signal i1: integer:=0;
begn
--new
--fuz : fuzzy port map (clk,Setrpm,rpm,w_out);
--pi_iq : pi_controle1 port map(clk,w_out,iq,iq_out);
--pi_id : pi_controle2 port map(clk,0,id,id_out);
--map4: qep port map (clk,cap2,rpm);
--map1: getspeed Disp
--port map (clk,swt1,swt5, cs,diow,rs,data,Setrpm,ADc1);
map0: lcd_sr
port map (clk,swt1,swt5,swt2,swt3,swt6,swt7,ADC1,ADC2,cs,diow,rs,data, Setrpm,rpm,iq,id);
mapx : adc_testing
port map (clk,busy,douta,doubt,adccs,adcsclk,addr,range0,cnvst,led,busy1,douta1,doubt1,adccs1,adcsclk1,addr1,cnvst1,adc1,adc2,adc3,adc4,adc5,adc6,adc7,adc8);
process(clk)
variable x: std_logic:='0';
variable y: std_logic:='0';
variable it: integer:=1;
variable i1t: integer:=1;
variable i2t: integer:=1;
variable i: integer:=1;
variable i1: integer:=1;
variable i2: integer:=1;
begin
iq_ref<=0;
--id_ref<=-250;
if rising_edge(clk) then
    case x is
    when '0' =>
        if cap1='0' then
            x:='1';
        end if;
    when '1' =>
        if cap1='1' then
            x:='0';
            flag1<=1';
        end if;
    when others=>
    end case;
sinetheta<=sine(i);
sinetheta_120<=sine(i1);
sinetheta_240<=sine(i2);
cosetheta<=cose(i);
SINT<=sine(i);
COST<=cose(i);
ialpha<=sinetheta/256; -- 5
ibeta<=148*(ialpha+2*ib)/256; --1/root3;
iq  <= (ialpha*COST - ibeta*SINT)/256;----------------- park transformation
id   <= (ialpha*SINT + ibeta*COST)/256;----------------- clarke transformation
\[\begin{align*}
\text{ph}_a &\leq \frac{(\text{COST} \cdot \text{iq}_\text{out} + \text{sint} \cdot \text{id}_\text{out})}{256}; \quad \text{inverse park transformation} \\
\text{ph}_b &\leq \frac{(-\text{sint} \cdot \text{iq}_\text{out} + \text{COST} \cdot \text{id}_\text{out})}{256}; \\
\text{va} &\leq \text{ph}_a; \quad \text{inverse clarke transformation} \\
\text{vb} &\leq \frac{(-\text{ph}_a - (\frac{443 \cdot \text{ph}_b}{256}))}{2}; \\
\text{vc} &\leq \frac{(-\text{ph}_a + (\frac{443 \cdot \text{ph}_b}{256}))}{2}; \\
\text{max} &\leq \text{va}; \quad \text{space vector pwm} \\
\text{min} &\leq \text{va}; \\
\text{tri} &\leq \frac{-(\text{max} + \text{min})}{2}; \\
\text{car}_1 &\leq (\text{tri} + \text{va}) \cdot 10; \\
\text{car}_2 &\leq (\text{tri} + \text{vb}) \cdot 10; \\
\text{car}_3 &\leq (\text{tri} + \text{vc}) \cdot 10; \\
\text{flag}_2 &\leq '0'; \\
\text{count} &\leq \text{count} + 1; \\
\text{if} \quad \text{count} > 0 \quad \text{and} \quad \text{count} \leq 1000 \quad \text{then} \\
\text{car} &\leq \text{car} - 2; \\
\text{else if} \quad \text{count} > 1000 \quad \text{and} \quad \text{count} < 2000 \quad \text{then} \\
\text{car} &\leq \text{car} + 2; \\
\text{else if} \quad \text{count} = 2000 \quad \text{then} \\
\text{count} &\leq 0; \\
\text{car} &\leq \text{car} + 2; \\
\text{end if}; \\
\text{if} \quad \text{car} > \text{va} \quad \text{then} \\
\text{pulse}_1 &\leq '1'; \\
\text{else} \\
\text{pulse}_1 &\leq '0'; \\
\text{end if}; \\
\text{if} \quad \text{car} > \text{va} - 60 \quad \text{then} \\
\text{pulse}_2 &\leq '0'; \\
\text{else} \\
\text{pulse}_2 &\leq '1';
\end{align*}\]
end if;
if car>vb then
pulse3<='1';
else
pulse3<='0';
end if;
if car>vb-60 then
pulse4<='0';
else
pulse4<='1';
end if;
if car>vc then
pulse5<='1';
else
pulse5<='0';
end if;
if car>vc-60 then
pulse6<='0';
else
pulse6<='1';
end if;
end if;
end if;
end if;
e1<='0';
e6<='0';
dir1<='1';
dir6<='0';
torque=(iq*0.6)/1515;

--- Fuzzy logic controller with Membership function -----------------------------

entity fuzzy is
port(clk: in std_logic;
    set: in integer:=0;
    act: in integer:=0;
    output: out integer:=0 );
end fuzzy;

architecture Behavioral of fuzzy is
type st_flow is (n_b,n_s, p_b,p_s,z_e,N_H,P_H);
type st_flow1 is (cn_b,cn_s, cp_b,cp_s,cz_e);
signal state : st_flow:=n_s;
signal statec : st_flow1:=cn_s;
signal vref: integer:=0;
signal SAMP: integer:=0;
begin
process(clk)
variable change,error,prev_error,iq: integer:=0;
begin
if rising_edge(clk) then
SAMP<= SAMP+1;
if SAMP=400000 THEN
error:= set-act;
if error>100 then
error:=100;
elsierror<-100 then
error:=-100;
else
error:=error;
end if;
end if;
change:=error-prev_error;
if error = -100 THEN STATE<= N_H;
ELSif error > -100 and error<=80 then
state<= n_b;
elsierror>-80 and error<=-5 then
state<= n_s;
elsierror>-5 and error<=5 then
state<= z_e;
elsierror>5 and error<=80 then
state<= p_s;
elsierror>80 and error<100 then
state<= p_b;
elsierror=100 THEN
state<= P_H ;
end if;
if change > -50 and change<=-40 then
statec<= cn_b;
elsiif change>-40 and change<=-3 then
statec<= cn_s;
elsiif change>-3 and change<=3 then
statec<= cz_e;
elsiif change>3 and change<=40 then
statec<= cp_s;
elsiif change>40 and change<=50 then
statec<= cp_b;
end if;
IF STATE = N_H THEN VREF<=-10;
ELSIF STATE=P_H THEN VREF<=10;
ELSif state = n_b and statec=cn_b then
vref<=-9;
elsiif state = n_s and statec=cn_b then
vref<=-5;
elsif state = z_e and statec=cn_b then
  vref<=-3;
elsif state = p_s and statec=cn_b then
  vref<=-5;
elsif state = p_b and statec=cn_b then
  vref<=-9;
elsif state = n_b and statec=cn_s then
  vref<=-5;
elsif state = n_s and statec=cn_s then
  vref<=-3;
elsif state = z_e and statec=cn_s then
  vref<=-1;
elsif state = p_s and statec=cn_s then
  vref<=-3;
elsif state = p_b and statec=cn_s then
  vref<=-5;
elsif state = n_b and statec=cp_s then
  vref<=5;
elsif state = n_s and statec=cp_s then
  vref<=3;
elsif state = z_e and statec=cp_s then
  vref<=1;
elsif state = p_s and statec=cp_s then
  vref<=3;
elsif state = p_b and statec=cp_s then
  vref<=5;
elsif state = n_b and statec=cp_b then
  vref<=9;
elsif state = n_s and statec=cp_b then
  vref<=5;
elsif state = z_e and statec=cp_b then
  vref<=1;
elsif state = p_s and statec=cp_b then
  vref<=5;
elsif state = p_b and statec=cp_b then
  vref<=9;
elsif state = n_b and statec = cz_e then
  vref<=-3;
elsif state = n_s and statec=cz_e then
  vref<=-1;
elsif state = z_e and statec=cz_e then
  vref<=0;
elsif state = p_s and statec=cz_e then
  vref<=1;
elsif state = p_b and statec=cz_e then
vref<=3;
end if;
prev_error:= error;
if iq>400 then
iq:=400;
elself iq<2 then
iq:=2;
else
IQ:=IQ;
end if;
output<=iq;
SAMP<=0;
end if;

----PI Controller-Iq
error1 = dutyq - iq;
prop1 = error1 * (float)kpq;
lcd1 = iq - feedback1;
integral1 = lcd1 * (float)kiq;
dutyq = dutyq + prop1 + integral1;
feedback1 = iq;
duty1=duty+50;
duty2=duty-50;
if(dutyq>duty1) dutyq = duty1;
if(dutyq<duty2) dutyq = duty2;

----PI Controller-Id
error2 = setcurrentd - id;
prop2 = error2 * (float)kpd;
lcd2 = id- feedback2;
integral2 = lcd2 * (float)kid;
dutyd =0;// dutyd + prop2 + integral2;
feedback2 = id;
if(dutyd>10) dutyd = 1;
if(dutyd<-10) dutyd = -1;
end process;
end Behavioral;
APPENDIX 3

INTELLIGENT POWER MODULE USER MANUAL

INTELLIGENT POWER MODULE
(Model No: PEC16DSMO1)

User Manual

Version 2.0

Technical Clarification / Suggestion:

Technical Support Division,
Vi Microsystems Pvt. Ltd.,
Plot No:75, Electronics Estate,
Perungudi, Chennai - 600 096, INDIA.
Ph: 91-44-4204 0142, 91-44-2496 3142
Mail: service@vimicrosystems.com,
Web: www.vimicrosystems.com
A3.1 INTRODUCTION

IPM based power module work as DC-DC Converter (Chopper) or DC-AC Converter (Inverter). It works using a IGBT based IPM and works on basis of software from DSP Processor. Intelligent Power Modules (IPMs) are advanced hybrid power devices that combine high speed, low loss IGBTs with optimized gate drive and protection circuitry. Highly effective over-current and short-circuit protection is realized through the use of advanced current sense IGBT chips that allow continuous monitoring of power device current. System reliability is further enhanced by the IPM's integrated over temperature and under voltage lock out protection. IPM has been optimized for minimum switching losses in order to meet industry demands for acoustically noiseless inverters with carrier frequencies up to 20 KHz.

A3.1.1 About Our Trainer

The following Figure A3.1 shows the front panel of "IPM BASED POWER MODULE PEC16DSMOI".
Front panel description

R,Y,B = Applied to 3 phase AC input supply.
BR1 & BR2 = Breaking Rheostat (470S - 2A).
+HV = Rectifier with filter DC output voltage (DC link voltage).
Voltmeter = Read the DC link voltage.
V/2 = Voltage across V/2 is half of the DC link voltage.

Feedback signals (Isolated current/voltage/speed sensor output)

I1, I2, I3 = 3 phase R,Y,B current transducer output currents are I1, I2, I3 respectively and measure this current across the terminals U, V and W.
VDC = DC link voltage (voltage transducer output).
IDC = DC link current (current transducer output).
N (Speed) = Analog voltage (0 - 5V).
F = Fault output signal comes from IPM, when over temperature/current occurs.
MCB = Power ON/OFF the 3 phase AC supply.
Power = Power ON/OFF the control circuits.

IGBT - PWM Inputs (from controller)

PWM1,..., PWM6 = PWM pulses are coming from controller.
PWM output

High - 5V = IGBT ON.
Low - 0V = IGBT OFF.
CAP1,...,CAP6 = Capture input to processor.

Protection Circuit

RST = Reset the protection circuit, then 'SD' LED will be off.
SD = Shut down LED will glow, when over voltage/current occurs in power circuit.

A3.2 SPECIFICATIONS

Input

* Three - phase AC supply (415V±10%)
* Single - phase AC supply (230V±10%)

Output

* DC Link voltage - 750V DC.
* Max. Current - 8A.
* Three- phase variable voltage & variable frequency.
* Fixed and variable DC voltage
* Single - phase variable voltage & variable frequency.

A3.3 DEVICE SPECIFICATIONS

Bridge Rectifier = 3N diode bridge Rectifier (60A,1200V)
IGBT Intelligent power module (25A, 1200V)

* Switching frequency = 20KHz (Max), 10KHz Nominal.

* Braking of IGBT = 10A, 1200V (Max). 6A Nominal.

* Fault output current (IFO) = 20mA (Max)

* Fault output voltage (VFO) = 20V (Max)

Voltage Transducer (LV25-P)

* Primary nominal r.m.s current (IPN) = 10mA

* Primary nominal r.m.s voltage (VPN) = 10 to 500V

* Secondary nominal r.m.s current (ISN) = 25mA

Current Transducer (LTS 25-NP)

* Primary nominal r.m.s current (IPN) = 8-12-25A

* R.M.S rated voltage (Vb) = 525V

* Analog output voltage (Vout) = 2.5V (Ip = 0)

IPM Power Supply

* Output = Four +15V supply
Max power=3W

* Primary to Secondary isolation = 2500 VRms, one minute

* Secondary to Secondary isolation voltage = 1500 VRms, one minute
High Voltage Input DC-DC Converter

* Input voltage (Vin) = 113V to 400V DC
* Output voltage (Vout) = 18 to 22 V DC
* Load Current = 220mA

A3.2 HARDWARE DESCRIPTION

The block diagram of IPM Based Power Module (PEC16DSMO1) is shown in Figure A3.2.

Figure A3.2 Block diagram for IPM based power module

It consists of:

1. Intelligent Power Module
2. Voltage and Current Sensor
3. Signal Conditioner
4. Protection Circuit
5. Opto Coupler
6. 3 N diode bridge Rectifier
7. Speed Sensor
8. Frequency to Voltage converter

A3.2.1 Intelligent Power Module

Mitsubishi Intelligent power Modules utilize many of the same field proven module packaging technologies used in Mitsubishi IGBT modules. Cost effective implementation of the built in gate drive and protection circuits over a wide range of current ratings was achieved using two different packaging techniques. Low power devices use a multilayer epoxy isolation system while medium and high power devices use ceramic isolation.

A3.2.2 IPM Protection

The following protection schemes available for Intelligent power Module.

i. Self Protection
ii. Under-Voltage Lock-Out Protection
iii. Over-Temperature Protection
iv. Over-Current Protection
v. Short Circuit Protection
A3.2.3 Handling Precautions for IPM

i. Electrical Considerations

* Apply proper control voltages and input signals before static testing.

* Carefully check wiring of control voltage sources and input signals. Miswiring may destroy the integrated gate control circuit.

* When measuring leakage current always ramp the curve tracer voltage up from zero. Ramp voltage back down before disconnecting the device. Never apply a voltage greater than the VCES rating of the device.

* When measuring saturation voltage low inductance test fixtures must be used. Inductive surge voltages can exceed device ratings.

ii. Mechanical Considerations

* Avoid mechanical shock. The module uses ceramic isolation that can be cracked if the module is dropped.

* Do not bend the power terminals. Lifting or twisting the power terminals may cause stress cracks in the copper.

* Do not over torque terminal or mounting screws. Maximum torque specifications are provided device data sheets.

* Avoid uneven mounting stress. A heatsink with a flatsink with a flatness of 0.001"/1" or better is recommended. Avoid one sided tightening stress. Uneven mounting can cause the modules ceramic isolation to crack.
iii. **Thermal consideration**

* Do not put the module on a hot plate. Externally heating the module’s base plate at a rate greater than 15°C/min. will cause thermal stress that may damage the module.

* When soldering to the signal pins and fast on terminals avoid excessive heat. The soldering time and Temperature should not exceed 230°C for 5 seconds.

* Maximize base plate to heatsink contact area for good heat transfer. Use a thermal interface compound such as white silicon grease. The heatsink should have a surface finish of 64 micro inches or less.

### A3.2.4 High Voltage Input DC-DC Converter

M57120L is a non-isolated DC-to-DC converter with a built in transformer wide range of input voltage (DC 113V-400V) enables direct connection to rectified 120V and 240V AC. This device is best suited for use as a pre regulator for standard DC-to-DC converters. The schematic diagram for High voltage input DC-DC Converter as shown in Figure A3.3.

![Figure A3.3 High Voltage input DC-DC Converter](image-url)
A3.2.5 IPM Power Supply

M57140-01 is an isolated DC-to-DC converter designed to drive IPMs (Intelligent Power Modules) with an input of DC 20V. The module supplies four 15V outputs. Isolation is provided from primary to Secondary and also between the secondaries. Interwinding isolation is designed for driving the IPM. IPM power supply schematic diagram is as shown in Figure A3.4.

IPM power supply output is connected to the power pin of IPM PM25RSB120. The M57410-01 is used under excessive Load condition, output side rectifying diodes will be destroyed. Care should be exercised so as not to operate the device above the rated maximum Load current.

![Figure A3.4 IPM Power Supply](image)

Coating Materials should not be applied on this device because the application of coating materials for water proofing could cause a stress and destroy a device.
A3.3 VOLTAGE AND CURRENT SENSOR

Intelligent power module output voltage and current is not directly feed to control (Protection) circuits. Intelligent power module output voltage is very high but control circuit operated in minimum voltage, So necessary for IPM output high voltage is convert into very low Voltage and current transducer sense from high voltage and output of transducer is low voltage (max 5V). The block diagram for voltage and current transducer output is as shown in Figure A3.5.

The sensor used for sensing current and voltage are works on the principle of hall effect, Hence these sensors are called hall effect transducer. Hall effect transducer output voltage and currents depends upon transducer primary and secondary winding ratio. The turns ratio represents the ratio of the number of primary turns to the number of secondary turns for a typical value of 1 : 1000, a primary current of 1A results in a secondary current of 1 mA. Voltage and current transducer principles of both are same, but one difference in primary winding of voltage transducer. The resistance connected in series with the primary winding of voltage transducer. This resistance can be external or integrated into the transducer construction.

Figure A3.5 Voltage and Current Transducer
A Hall effect current transducers senses the current IDC, I1(U), I2(V) I3(W) and one hall effect voltage transducer senses the DC link voltage (VDC).

### A3.3.1 Unipolar Power Supply

Most of the LEM transducers can also be supplied by an unipolar voltage for measurement of unidirectional currents. In this case the following must be taken into consideration:

1. The supply voltage is the sum of the positive and negative voltages indicated in the data sheet.

2. The load resistance shall be calculated separately, in order not to exceed the acceptable dissipated power of the amplifier’s final stage. As a first approximation this calculation is not necessary if one does not exceed half of the nominal primary current. In other cases please consult us.

![Figure A3.6 Disposition of diode(s) with an unipolar power supply](image)

3. As the amplifier circuit is designed for a bipolar power supply and is used here as unipolar, diodes must be inserted into the measuring circuit, as shown in Figure A3.6. This is in order to compensate the residual voltage across the unused output transistor which could generate a current comparable to an offset in the measuring circuit. Furthermore, variants specially adapted for unipolar operation are available as a standard device.
A3.4 PROTECTION CIRCUIT

The schematic diagram for protection circuit of IPM based power module is as shown in Figure A3.7. Protection circuit is used to prevent the over voltage, over current and under voltage. The current and voltage from signal conditioner's are given to input of master / slave JK Flip flops. Master / Slave JK flip flop output is connected to transistors Q1 and Q2. Transistor Q1 output C1 terminal is given to input of AND-7 & AND-(1-6) Gates. AND (1-6) Gates another input is feed from PWM output of DSP. These output of AND Gates depends upon transistor Q1 output, then AND gates output is given to input of optocoupler (1), then optocoupler (1) output signal is feed to input of IPM. IPM is generate to the fault output signals, when over current/ voltage occurs an IPM. This signal is feed to optocoupler (2) and optocoupler (2) output is ANDed with Q1 output (C1) signal. The AND7 gate output is given to input of PDPINT (DSP). Normal condition PDPINT - high, PDPINT is disable when over temperature and over current occurred in power circuit of joint to C1.
Many protection hardware requirement is build to this protection circuit, when over voltage or over current occurs in a power circuit, the DSP.CT.IN or DSP.VOLT.IN output is high '1', input of master/slave Jk Flip flop J = 1, output Q = 1, Q' = 0, then transistor Q1 and Q2 conduct and shut down LED "SD" will glow (LED Glow to indicate the power circuit affected by the over voltage / over current) at that same time transistor Q1 output (C1 terminal) is 0V. Then AND (1-6) & AND 7 gate output is low '0', then automatically cut the PWM signal to IPM and shutdown the IPM.

Voltage and current sensor, output is feed to ADC of DSP. When, find out the over voltage / current from the ADC inputs of DSP. Then cut the PWM signal to protection circuit and shutdown the IPM.

A3.5 THREE PHASE DIODE BRIDGE RECTIFIER

This rectifier provides the rectified DC voltage to the intelligent power module. 3 Phase Diode Bridge Rectifier circuit diagram is as shown in Figure A3.8.

![Figure A3.8 Three Phase Diode Bridge Rectifier](image)

Each Three phase line connects between a pair of diodes. One to route power to the positive (+) side of the load and the other to route power to
the negative (-) side of the load. Polyphase systems with more than three phase are easily accommodated into a bridge rectifier scheme. 3M AC supply is connected to input of 3M bridge rectifier module. 3M bridge rectifier convert the AC voltage into DC voltage with AC ripples. Capacitor is connected across the bridge rectifier. Capacitor is used to neglect the AC ripples. 3M diode bridge rectifier module output is connected to input of intelligent power module.

A3.6 FREQUENCY TO VOLTAGE CONVERTER

The square wave of speed sensor output is feed to frequency to voltage converter circuit. The XR4151 can be used as a frequency to voltage converter. The voltage applied to comparator input Pins6 and 7 should not be allowed to go below ground by more then 0.3V. The input frequency range is 0 to 10KHz and corresponding voltage output level is -10mV to -10V. In our module set the frequency to voltage converter with Max. output of 2.5V at rated motor speed of 1500 rpm.

A3.7 GENERAL INSTRUCTION

Precaution

1. You are working with IPM based power module, in case of over voltage/under voltage/over current/ over temperature occur; shut down LED 'SD' will glow. The Gating signals to the inverter switches will be switched off. Adjust the input voltage to minimum position. Then press the reset switch, shut down LED 'SD' will gets turned OFF. Then again start the experiments.
2. If CRO is used to the output voltage wave forms it is recommended to isolate the input AC.

**Measurement**

1. Measure the DC Link voltage from the voltmeter.
2. Check the pulse sequences PWM1 to PWM6 at the test points of IPM based module (PEC16DSMO1) by using CRO.
3. Observe the current and voltage wave forms from the test points for I1(R), I2(y), I3(B) and VDC with respect to ground.
4. Measure the frequency to voltage converter output across the test point ‘N’ and GND.
5. Shut down LED 'SD' indicates whether the fault has occurred (over voltage /under voltage/over current/ over temperature ) or not.

**Note**

1. Initially keep the MCB switch in 'OFF' position.
2. Connect the three-phase AC supply across the terminals R, Y & B.
3. The "FEED BACK SIGNALS" and "IGBT PWM INPUTS" connectors of IPM modules are connected to Controller by using 26 pin and 34 pin FRC connectors respectively.
4. Connect the external Rheostat (470Ω/2A) across the banana connectors BR1 and BR2.