PM9603AP

SINGLE PHASE POWER/ENERGY METERING MODULE SPI INTERFACE

FEATURES

- Performs both power and energy measurement
- Meets the accuracy requirements for Class 1 AC Watt hour meters
- Protected against ESD
- Total power consumption rating below 500mW (excluding current sensing)
- Uses a shunt resistor for current sensing
- Operates over a wide temperature range
- Isolated SPI interface.

DESCRIPTION

The SAMES single phase power/energy metering module, the PM9603AP, provides energy data via a isolated SPI interface.

Energy consumption is determined by the power measurement being integrated over time.

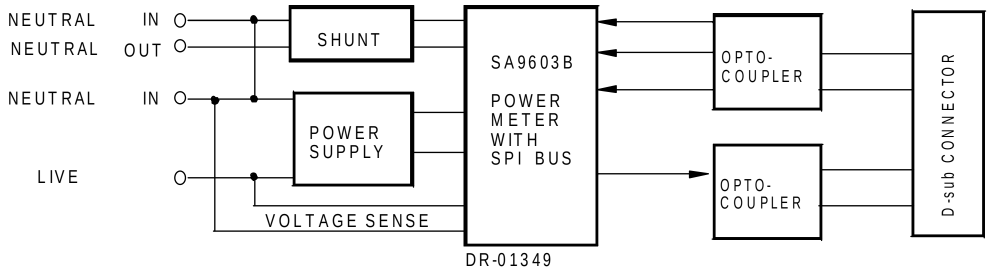
The method of calculation takes the power factor into account.

The output of this innovative universal power/energy meter is ideally suited for energy calculations in applications using a micro-controller.

The application utilises the SAMES SA9603B power metering integrated circuit for power measurement.

As a safety measure, this application shows the current sensor connected to the neutral line. In practice, the live line may be used for current sensing, provided that the supply connections (MAINS) are reversed on the module.

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS*

Parameter	Symbol	Min	Max	Unit
Supply Voltage (Note 1)	V_{AC}		300	V
Current Sense Input (Note 1)	V_{IV}	-2.5	+2.5	V
Storage Temperature	T_{STG}	-25	+125	°C
Operating Temperature	T_O	-10	+70 (Note 2)	°C
Max Current	I_{MAX}		800 (Note 3)	A
through Sensor	I_{MAX}		2000 (Note 4)	A

Note 1: Voltages are specified with reference to Live.
Note 2: The SA9603B integrated circuit is specified to operate over the temperature range -10°C to +70°C. The module functionality will however depend upon the external components used.
Note 3: t = 500ms
Note 4: t = 1ms

*Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only. Functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification, is not implied. Exposure to Absolute Maximum Ratings for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS

(Over the temperature range -10°C to +70°C, unless otherwise specified. Power consumption figures are applicable to the PM9603APE only.)

Parameter	Symbol	Min	Typ	Max	Unit	Condition
Supply Voltage (Continuous)	V_{AC}	180 90	230 115	265 135	V V	PM9603APE PM9603APA
Power Measurement range	P_{RNG}	-18400		18400	W	Specified accuracy
Power Consumption[1]				800	mW	V_{AC} = 230V Supply direct from mains
Isolation Voltage[2]	V_{IS}			2500	V	Continuous
Opto-coupler Output Current	I_O			10	mA	V_{OL} = 1V
Opto-coupler Input Current	I_I			10	mA	

Note 1: Power consumption specifications exclude power consumed by the current sensor.

Note 2: Isolation voltage may be specified, depending on customer requirements.

CONNECTION DESCRIPTION

Designation	Description
MAINS	Voltage supply connection to Neutral line
	Voltage supply connection to Live line
NEUTRAL IN	Connection to positive side of current sensor
NEUTRAL OUT	Connection to negative side of current sensor
SK1 25-Way female (D-type)	P1 Supply (+ve) to opto-couplers
	P2 Input SA9603B - SCK
	P8 Input SA9603B - DI
	P9 Input SA9603B - CS
	P12 Output SA9603B - DO
	P18, 20-25 Common emitters and cathodes opto-couplers

FUNCTIONAL DESCRIPTION

1. Power Calculation

In the Application Circuit (see Figure 2), the output current from the current sensor will be between 0 and 16µA_{RMS} (0 to 80A through a shunt resistor of 625µΩ). The current input stage of the module, saturates at input currents greater than 18µA_{RMS}. The mains voltage (Voltage + 15% - 20%) is used to supply the circuitry with power. A SA9603B utilize current information from the current sensor (shunt resistor), together with the mains voltage to perform the power calculation.

The SA9603B integrated circuits may be adjusted to accommodate any voltage or current values. The method for calculating external component values is described in paragraph 6 (Circuit Description).

SAMES offers two evaluation module options, namely 230V/80A and 115V/80A.

The on chip registers are accessed via the isolated SPI bus.

2. Electrostatic Discharge (ESD) Protection

The device's inputs/outputs are protected against ESD according to the Mil-Std 883C, method 3015. The modules resistance to transients will be dependant upon the protection components used.

3. Power Consumption

The overall power consumption rating for this power metering application (Figure 2), is under 500mW, excluding the current sensor, when the supply is taken directly from the mains.

4. Isolation

The reference of the module is connected to neutral.

5. Isolated Input/Output Interface

The isolated interface is provided to allow the user to access the registers of the SA9603B.

A 25-Way D type connector (female) is provided on the PM9603AP module. The connector SK1 connects via a one to one connected cable to a Personal Computer parallel port.

6. Circuit Description

In the Application Circuits, (Figures 1), the components required for power metering applications are shown.

In Figure 1, a shunt resistor is used for current sensing. In this application, the circuitry requires a +2.5V, 0V, -2.5V DC supply.

The current sense input requires a differential approach to cater for precision across the dynamic range. It is therefore important that the PC board layout of the branches to the sensing element, are as symmetrical as possible and the loop area is kept to a minimum.

The most important external components for the SA9603B integrated circuit are: R_2, R_1 and RSH are the resistors defining the current level into the current sense input. The values should be selected for an input current of 16µA_{RMS} into the SA9603B at rated line current.

Values for RSH of less than 200µΩ should be avoided.

$R_1 = R_2 = (I_L/16\mu A_{RMS}) * RSH/2$

Where I_L = Line current

RSH = Shunt resistor/termination resistor

R_3, R_6 and R_4 set the current for the voltage sense input. The values should be selected so that the input current into the voltage sense input (virtual ground) is set to 14µA_{RMS}.

R_7 defines all on-chip bias and reference currents. With R_7 = 24kΩ, optimum conditions are set.

XTAL is a colour burst TV crystal (f = 3.5795MHz) for the oscillator. The oscillator frequency is divided down to 1.7897MHz on-chip and supplies the A/D converters and the digital circuitry.

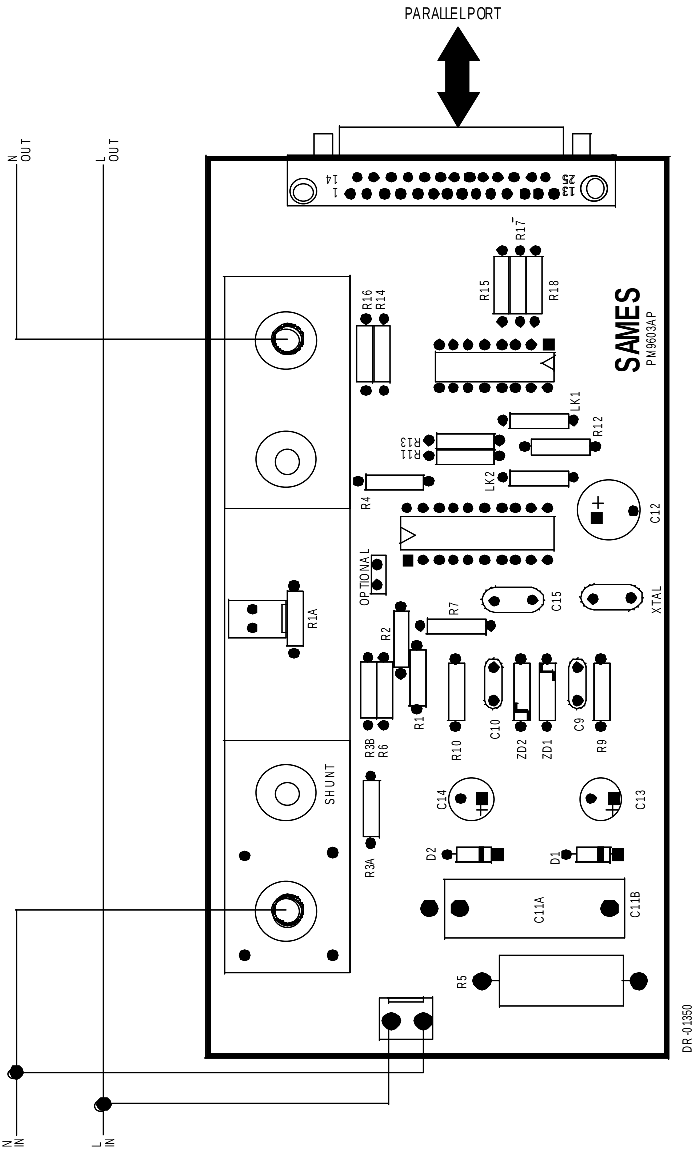
7. Demonstration Software

Software which runs under Windows 3.1 and Windows 95 is provided with each evaluation module. See README.TXT on the diskette supplied for the installation instructions.

8. Sample C source code

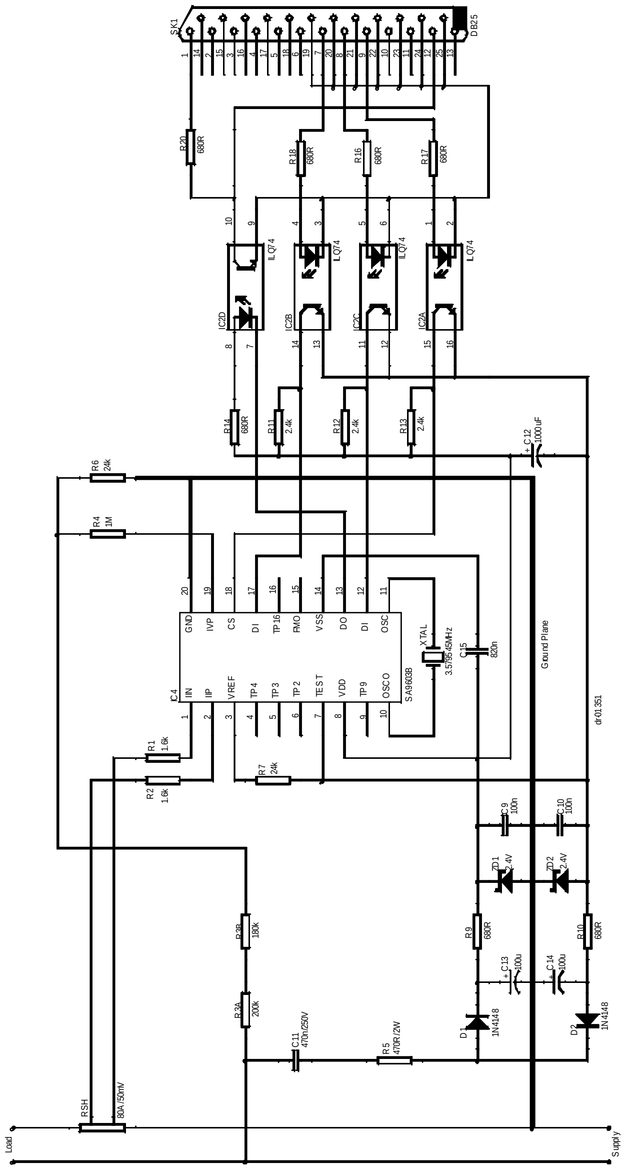
The following software demonstrates how to synchronize the reading of the registers to the SA9603B's internal offset cancellation scheme. The software is also available on the SAMES Internet web pages.

Figure 1: Connection Diagram



APPLICATION CIRCUIT

Figure 2: Application using a Shunt Resistor for Current Sensing, having a PC (Personal Computer) Interface.



Parts List For Application Circuit: Figure 2

Item	Symbol	Description	Detail
1	IC-1	SA9603B	DIP-20
2	IC-2	Opto Coupler, ILQ74	DIP-16
3	D1	Diode, Silicon, 1N4148	
4	D2	Diode, Silicon, 1N4148	
5	ZD1	Diode, Zener, 2.4V, 200mW	
6	ZD2	Diode, Zener, 2.4V, 200mW	
7	XTAL	Crystal, 3.5795MHz	Colour burst TV
8	R1	Resistor, 1.6kΩ, 1%, metal	Note 1
9	R2	Resistor, 1.6kΩ, 1%, metal	Note 1
10	R3A	Resistor, 1%, metal	Note 2
11	R3B	Resistor, 1%, metal	Note 2
12	R4	Resistor, 1M, ¼W	
13	R5	Resistor, 470Ω, 2W, 5%, carbon	
14	R6	Resistor, 24k, ¼W, metal	
15	R7	Resistor, 24k, ¼W, metal	
16	R9	Resistor, 680Ω, ¼W, 5%	
17	R10	Resistor, 680Ω, ¼W, 5%	
18	R11	Resistor, 2.4k, ¼W, 5%	
19	R12	Resistor, 2.4k, ¼W, 5%	
20	R13	Resistor, 2.4k, ¼W, 5%	
21	R14	Resistor, 680R, ¼W, 5%	
22	R15	Resistor, 680R, ¼W, 5%	
23	R16	Resistor, 2.4k, ¼W, 5%	
24	R17	Resistor, 680R, ¼W, 5%	
25	R18	Resistor, 680R, ¼W, 5%	
26	C9	Capacitor, 100nF	
27	C10	Capacitor, 100nF	
28	C11	Capacitor, polyester	Note 2
29	C12	Capacitor, 1000µF, 16V	
30	C13	Capacitor, 100µF, 16V	
31	C14	Capacitor, 100µF, 16V	
32	C15	Capacitor, 820nF, 16V	
33	RSH	Shunt Resistor, 80A, 50mV (625µΩ)	Note 1

Note 1: Resistor (R1 and R2) values are dependant upon the selected value of RSH. See paragraph 6 (Circuit Description) when selecting the value for RSH.

Note 2: See the table below, detailing the component values for the selected voltage standard.

Item	Symbol	Description		Detail
		PM9603APA 115V	PM9603APE 230V	
12	R3A	120kΩ	200kΩ	
13	R3B	82kΩ	180kΩ	
31	C11	1µF	0.47µF	

ORDERING INFORMATION

Part Number	Description
PM9603APA	115V, 80A Module
PM9603APE	230V, 80A Module

```
//  This program will read the registers from a SA9603B device
and
 // display the content on the screen.
 // The program work on a PM9603AP demonstration module.

 #include <stdio.h>
 #include <math.h>

 #define   PI       3.141593
 #define   win_size 20

 // Definitions for the parallel port
 #define CLK_b           0x20 //D5 on port  pin 7
 #define DO_b            0x40 //D6 on port  pin 8
 #define DI_b      0x20 //Paper out   pin 12
 #define CS_b      0x80 //D7 on port  pin 9
 #define PCTrig       0x02 //D1 on port  pin 3
 //#define freq_bits               0x00FFFF
 //#define inv_bits                0x0C0000

 #define freq_bits  0x01FFFF       // D16..D0
 #define inv_bits   0x0E0000
 #define bits_50hz  0x080000

 // Mask definitions for the parallele port
 #define CLK_m         ~CLK_b //D5 on port  pin 7
 #define DO_m          ~DO_b  //D6 on port  pin 8
 #define DI_m       ~DI_b  //Paper out   pin 12
 #define CS_m       ~CS_b  //D7 on port  pin 9
 #define byte_mask   0x80  //1000 0000

 #define CLOCKFREQ  3579545
 #define LPT1             0x378
 #define LPT2             0x278

 // Glogal variables
 unsigned char   portval;

 double
arr_energy[win_size],arr_volt[win_size],arr_reactive[win_size];
 long     delay_time;
```

```
 int prt, buffindex = 0;


 double convert_24bits(double value)
 // This function will sort out the 24 bits of the register
values
 {
     if (fabs(value) >  0x7FFFFF)
     {
       if (value > 0)
         value = (16777216-value) * (-1);
        else
         value = (16777216+value)* (1);
      }
   return(value);
 }


 void spi_wait(void)
 // This function will wait for the specified period, very
short time
        {
     int  ti;
         for (ti = 1; ti < 1 ; ti++);
        }

 void spi_out(unsigned char value)
 // Put value on the parallel port
        {
        int n;
        portval = value;
        outport(prt,(~portval));
        // Remove the ~ if you dont use opto couplers
        for(n=0 ; n< delay_time ; n++); // Specify the pulse
width
        }

 void spi_clock(void)
 // This function will pulse CLK pin of spi port
        {
        spi_wait();
        spi_out(portval | CLK_b);  // or
        spi_wait();
```

```
    spi_out(portval ^ CLK_b);  // xor
        }

 void spi_PCTrig(void)
 // This function will set the PC trigger pin
        {
        spi_out(portval & (~PCTrig));  // and
        spi_wait();
        }

 void spi_ClearPCTrig(void)
 // This function will clear the PC trigger pin
        {
        spi_out(portval | PCTrig);  // or
        }


 void spi_cs(void)
 // This function will set CSs pin of spi port
        {
        spi_out(portval | CS_b);  // or
        }


 void spi_reset(void)
 // all bits of the port are made 0
        {
        spi_out(portval & CLK_m);
        spi_out(portval & DO_m);
        spi_out(portval & CS_m);
        }

 void spi_start_read(unsigned char value)
 // All bits of the port are made 0
        {
        int counter;
        unsigned char and_mask,mask;

       mask = 0x80;
       value = (value | 0xC0 );   // Put the header for the
adress 110—
```

```
        for ( counter = 0; counter <8; counter++)
           {
            and_mask = (mask & value);
            if (and_mask == mask)
             {
             spi_out(portval | DO_b);  // 1
             spi_clock();
             }
            else
            {
            spi_out(portval & DO_m);   // 0
            spi_clock();
            }

             value = value << 1;
            }
            spi_clock();  // sort out the extra clock cycle
between command & response
        }

 unsigned long spi_read_register(void)
 // This function will read 24 bits of a register
 // the register to be red would be indicated by the
 // preceding spi_start_read function
         {
         int counter;
         unsigned char and_mask,mask;
         unsigned long fromport;

        fromport = 0 ;
        for ( counter = 0; counter < 24; counter++)
           {
                spi_clock();
                if (((inportb(prt+1)) & DI_b) == DI_b)
                fromport = fromport+ pow(2,(23-counter));
              }
              return(fromport);
        }
```

```
 /
***********************************************************************/
 /*
*/
 /*                   Main Function.
*/
 /*
*/
 /
***********************************************************************/

 void main (int argc,char *argv[])

 { FILE *out;

 char  key;

 int        port,del_time ;
 long    act , react , volt , freq, old_freq,
         act1, react1,volt1,prev_inv_bits;
 double  k, active, reactive, voltage,fm,vm;

 int first = 1;

              fm = 0;
              act=   0;
              react= 0;
              volt=  0;
              freq = 0;
              del_time = 0;

 if (argc == 3 ) {
   delay_time = atol(argv[2]);
   prt = atol(argv[1]);
   if(prt == 1) prt = LPT1;
   else prt = LPT2;

 } else {
   printf("\nusage : CHIP9603 <portnum> <SPI clock delay
lime>\n");
   printf("       <portnum>   =   1 or 2 (lpt port
```

```
number)\n");
   printf("       <SPI clock delay lime> = For loop delay\n");
   printf("       use 1000 for a 486dx266                \n");
   exit(0);
 }

       clrscr();

          while(key !='q')
         {
               if(bioskey(1)) {
                key = bioskey(0);
               }
          old_freq = freq;

          spi_reset();
          spi_cs();
           spi_start_read(3);
          freq  = spi_read_register();

          // Get the rising edge if the inversion bits
          if (((freq & bits_50hz) == bits_50hz )        //D19
is now set
              && ((old_freq & bits_50hz) != bits_50hz)) //D19
was not set previously
              {
              // The following is used to indicate that
              //registers is read every 8th mains cycle
              spi_PCTrig();
              delay(1);
              spi_ClearPCTrig();

               //del_time is the amount of inversion cycles to
wait before
               //reading the rest of the register values.
               del_time ++;
          }

          if(del_time == 10)    // 5 x 8 mains cycles time to
integrate registers
           {
            //  spi_PCTrig();
```

```
              spi_reset();
              spi_cs();
                spi_start_read(0);
              act   = spi_read_register();
              react = spi_read_register();
              volt  = spi_read_register();

 // Now calculate the incremental difference  and take care of
register
 // overflow as well as the sign of register values
              active = convert_24bits(act - act1);
              reactive = convert_24bits(react - react1);
              voltage = convert_24bits(volt1 - volt);

              k = (1/((double)(freq&freq_bits)*del_time)) * 80
* 230/(1.44*2);

              fm = (double)((CLOCKFREQ/2)/((double)(freq &
freq_bits)));

              vm = (double)voltage * ((14/17.5)*0.63)/
((double)(freq&freq_bits)*del_time) *230;

              if (first == 0){
                 printf("%08.2f \t",active*k);
                 printf("%08.2f \t",(reactive*k*PI/2));
                 printf("%08.2f \t",vm);
                 printf("%08.4f \t",fm);
                 printf("%5.2f  \n",(del_time*(8/fm)));
                 }

              first = 0;
              act1  = act;
              react1= react;
              volt1 = volt;
              del_time = 0;
             }
       }//while
 }
```

Notes:

Any Sales or technical questions may be posted to our e-mail address below:
energy@sames.co.za

For the latest updates on datasheets, please visit out web site:
http://www.sames.co.za

South African Micro-Electronic Systems (Pty) Ltd

P O Box 15888,
Lynn East,
0039
Republic of South Africa,

33 Eland Street,
Koedoespoort Industrial Area,
Pretoria,
Republic of South Africa

Tel: 012 333-6021
Fax: 012 333-8071

Tel: Int +27 12 333-6021
Fax: Int +27 12 333-8071