MSI-P412

ANALOG INPUT CARD

USER MANUAL

PC/104 Embedded
Industrial Analog I/O Series

Microcomputer Systems, Inc.
1814 Ryder Drive ¨ Baton Rouge, LA 70808
Ph (225)769-2154 ¨ Fax (225) 769-2155
Email: staff@microcomputersystems.com
http://www.microcomputersystems.com

CONTENTS

I. INTRODUCTION

The MSI-P412 is a low cost, high performance 12-bit analog input card designed for use with all PC/104 embedded systems. Four models provide input capacities of 8, 16, 24 or 32 channels which operate from a single +5V supply. Software programmable input ranges are 0-5V, 0-10V, ±5V and ±10V with a linearity of 1/2 LSB. In addition, the inputs are overvoltage tolerant to ±16.5V and a fault condition on any channel will not effect the conversion result on the selected channel. A block diagram of the card is shown below.

The card employs up to four MAX197 eight-channel A/D converters that incorporate a precision 2.5V reference source with buffer amp, an internal 1.56 MHz clock, and successive approximation and internal input track/hold circuitry to convert the analog signal of each channel into a 12-bit digital signal. Low span and offset errors result in no adjustments being required for these functions. Typical total conversion times of 12 us gives a sample rate of 83 ksps for each group of eight channels yielding rates up to 332 ksps for 32 input channels.

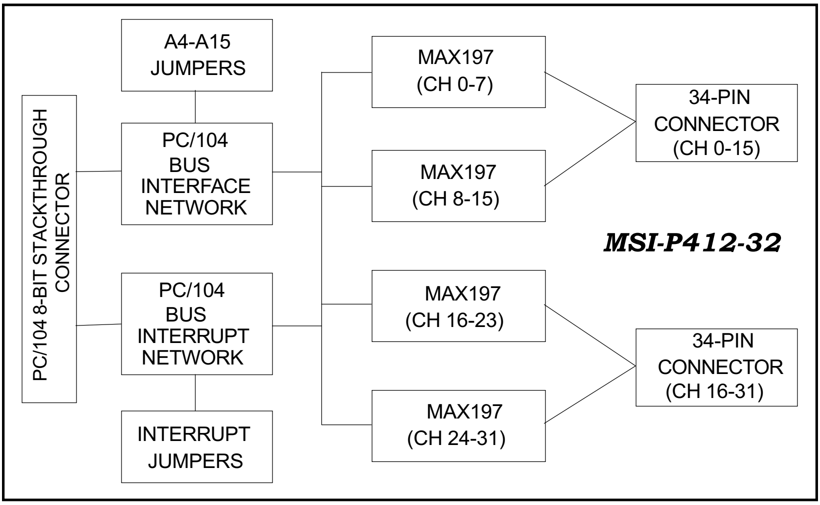
A4-A15 JUMPERS
PC/104 8-BIT STACKTHROUGH CONNECTOR
PC/104 BUS INTERFACE NETWORK
PC/104 BUS INTERRUPT NETWORK
INTERRUPT JUMPERS
MAX197 (CH 0-7)
MAX197 (CH 8-15)
MAX197 (CH 16-23)
MAX197 (CH 24-31)
34-PIN CONNECTOR (CH 0-15)
34-PIN CONNECTOR (CH 16-31)
MSI-P412-32

Figure 1. Block Diagram of the MSI-P412-32.

The card is I/O mapped using 16-bit addressing to select the input channels and device status. Option jumpers are provided by JP1 for specifying the card address (A4 - A15) and interrupt processing is provided for IRQ4 thru IRQ7 and IRQ9 using options jumpers, as described in the next section.

The MSI-P910 terminal card can be used to provide up to 16 analog inputs via terminal strips. This card contains resistor sites for accommodating current inputs of 0-20 mA or 4-20 mA and includes surge protection for protecting against spurious voltages prevalent in harse or industrial environments.

II. HARDWARE DESCRIPTION

A. Card Configuration

The MSI-C412 card is a CMOS design using through-hole and surface-mounted devices. The card configuration is shown in Figure 2 and a circuit diagram of the network is given in the Appendix. The input signals for channels 0 thru 15 are applied to connector P1 and channels 16 thru 31 to P2. These signals are directed to the input terminals of A/D converters U6 (Ch. 0-7), U7 (Ch. 8-15), U8 (Ch. 16-23) and U9 (Ch. 24-31).

Jumper block JP1 is used for address selection (Pins 1 thru 24) and interrupt configuration (Pins 25 thru 34), as described below.

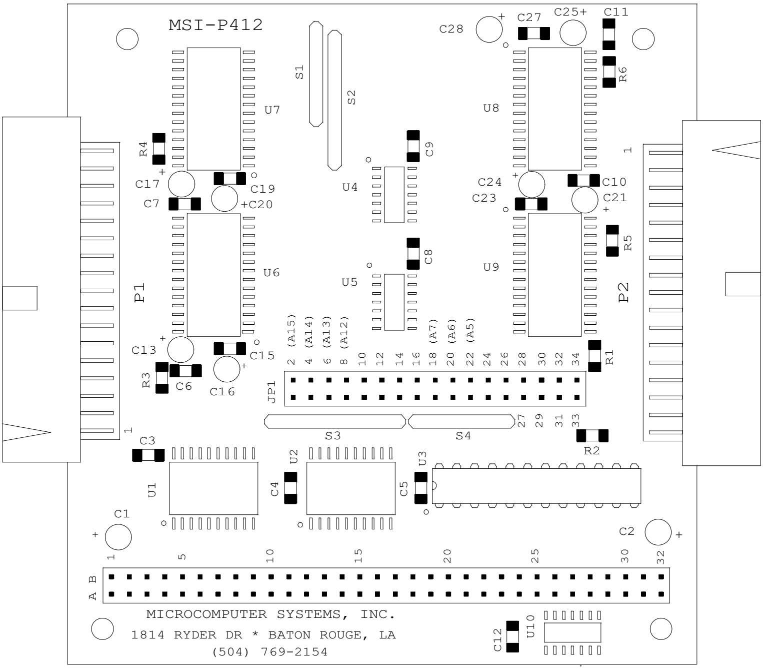

Figure 2. MSI-P412 card outline.

B. Card Addressing

The card address is set by installing appropriate jumpers on JP1, pins 1 thru 24, as shown in Fig. 3. An uninstalled jumper for a given address bit sets the bit to 1 (true) and an installed jumper sets the bit to 0 (false). Addresses A4 thru A15 are jumper selectable for defining the *base address* of the card from 0000H to FFF0H on integral 10H boundaries, where H denotes a hexadecimal number. To assign a base address of 3040H, for example, install jumpers JP1-A4, JP1-A5 and JP1-A7 thru JP1-A13. Pins 25 thru 34 are used to configure the interrupt connections, if interrupts are used, as described in the Section II.C.

The MAX197 converters each have two registers for performing data conversions, a control output register (C) and a input data register (I). A third register implemented on the card for denoting interrupt status is called the status register. The addresses of the control, input data (C/I) and status for each channel is given in Table 1. The functions of the control, the input data (hi and lo bytes), and status registers are described in the Section III.

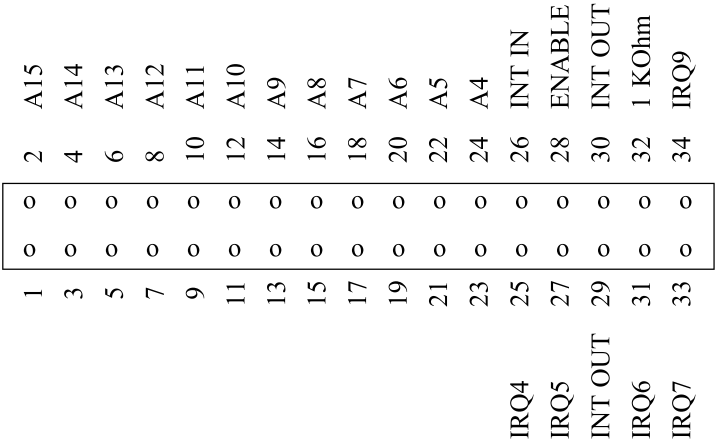

Figure 3. Jumper block JP1 configuration.

Table 1. Analog Converter Control Register Addresses

Channels	Control Output (C)	Data Input (I)	Status / Bit
0-7	base+0	base+0 (lo) base+1 (hi)	base+8 / 0
8-15	base+2	base+2 (lo) base+3 (hi)	base+8 / 1
16-23	base+4	base+4 (lo) base+5 (hi)	base+8 / 2
24-31	base+6	base+6 (lo) base+7 (hi)	base+8 / 3
0-31			base+8 / 4

B. Interrupt Connections

Interrupt connections are implemented by pins 25 thru 34 of JP1. The steps in the procedure are as follows.

1) Pin 30 (INT OUT) is the composite interrupt signal from the analog converters (see circuit diagram in Appendix). This is jumpered to a single desired interrupt, IRQ4 thru IRQ9, of JP1 shown in Fig. 3. This connection causes the interrupt selected to be activated when a conversion is performed.

2) If no other cards in the system are using the interrupt line chosen in step 1, proceed to step 3.

Pin 26 of JP1 (INT IN) is used to control the tri-state buffer of INT OUT when other cards are using the same interrupt line chosen in step 1. In this case, connect pins 26 and 28 of JP1. This results in the tri-state buffer of INT OUT being enabled only when an interrupt request is active. The status is then checked to determine which interrupt is active, as described in Section III.

3) Pin 32 of JP1 (1 KOhm pull-down resistor) is used to properly terminate the interrupt line selected in step 1. This pin should be jumpered to the interrupt line on only one card in the system.

D. Connecting Inputs to P1 and P2.

Inputs are interconnected to the card via P1 and P2 using 34-pin flat cable connectors. Pin assignments are given in Table 2.

Table 2. Pin assignments for Input Connectors P1 and P2.

Input	Pin	Input	Pin	Input	Pin	Input	Pin
Ch 0+	P1-15	Ch 8+	P1-33	Ch 16+	P2-15	Ch 24+	P2-33
Ch 0-	P1-16	Ch 8-	P1-34	Ch 16-	P2-16	Ch 24-	P2-34
Ch 1+	P1-13	Ch 9+	P1-31	Ch 17+	P2-13	Ch 25+	P2-31
Ch 1-	P1-14	Ch 9-	P1-32	Ch 17-	P2-14	Ch 25-	P2-32
Ch 2+	P1-11	Ch 10+	P1-29	Ch 18+	P2-11	Ch 26+	P2-29
Ch 2-	P1-12	Ch 10-	P1-30	Ch 18-	P2-12	Ch 26-	P2-30
Ch 3+	P1-9	Ch 11+	P1-27	Ch 19+	P2-9	Ch 27+	P2-27
Ch 3-	P1-10	Ch 11-	P1-28	Ch 19-	P2-10	Ch 27-	P2-28
Ch 4+	P1-7	Ch 12+	P1-25	Ch 20+	P2-7	Ch 28+	P2-25
Ch 4-	P1-8	Ch 12-	P1-26	Ch 20-	P2-8	Ch 28-	P2-26
Ch 5+	P1-5	Ch 13+	P1-23	Ch 21+	P2-5	Ch 29+	P2-23
Ch 5-	P1-6	Ch 13-	P1-24	Ch 21-	P2-6	Ch 29-	P2-24
Ch 6+	P1-3	Ch 14+	P1-21	Ch 22+	P2-3	Ch 30+	P2-21
Ch 6-	P1-4	Ch 14-	P1-22	Ch 22-	P2-4	Ch 30-	P2-22
Ch 7+	P1-1	Ch 15+	P1-19	Ch 23+	P2-1	Ch 31+	P2-19
Ch 7-	P1-2	Ch 15-	P1-20	Ch 23-	P2-2	Ch 31-	P2-20

Note: Pins P1-17, P1-18, P2-17 & P2-18 are connected to +5V.

E. Using the MSI-P910 and Current Inputs.

A schematic of the MSI-P910 terminal card is given in the Appendix. The card provides terminal strips for connecting field wiring for 16 channels (8 channels for the MSI-P910-8) using a 34-pin flat cable. Pads are also included for adding 250 Ohm precision resistors for 0-20 mA or 4-20 mA inputs. <u>The 0-5V input range is used in these cases.</u>

III. PROGRAMMING

Performing data conversions involves a write operation to the control register of the appropriate MAX197, which selects the mux channel and configures the input mode. The data is then read, lo byte and hi byte, when the conversion has been completed.

A. Control Register Format

The control register is an 8-bit (write-only) register that selects the mux channel and mode of the converter. The format is

D7(MSB)	D6	D5	D4	D3	D2	D1	D0(LSB)
PD1	PD0	ACQMOD	RNG	BIP	A2	A1	A0

where

PD1, PD0 select the clock and power down mode (Table 3).

ACQMOD - 0 = internally controlled acquisition,
1 = externally controlled acquisition.

RNG selects the full-scale voltage range (Table 4).

BIP selects unipolar or bipolar conversion mode (Table 4).

A2, A1, A0 select the desired input channel 0-7 of the MAX197.

Table 3. Clock and Power Down Selection

PD1	PD0	DEVICE MODE
0	0	Normal Operation/External Clock Mode
0	1	Normal Operation/Internal Clock Mode
1	0	Standby Power-down (STBYPD); clock unaffected
1	1	Full Power-down (FULLPD); clock unaffected

Table 4. Range and Polarity Selection

BIP	RNG	INPUT RANGE (V)
0	0	0 to 5
0	1	0 to 10
1	0	±5
1	1	±10

The card is designed to operate using the internal clock with PD1 = 0 and PD0 = 1 in normal operation. The internally controlled acquisition (ACQMOD = 0) is normally used .

B. Performing a Conversion

Conversions are initiated with a write operation to the control register (Table 1), which selects the mux channel of the desired MAX197 (U6, Ch 0-7; U7, Ch 8-15; U8, Ch 16-23; or U9, Ch 24-31) and configures the device mode. Selecting ACQMOD = 0 in the control register selects the internal acquisition mode. This causes the write to the control register to initiate the acquisition interval whose duration is internally timed. Conversion starts when this six-clock-cycle acquisition interval ends. Writing a new control byte during the conversion cycle will abort the conversion and start a new acquisition interval.

C. Reading the Data Register

Conversions are complete when the appropriate status bit (Table 1) becomes 0 following a write to the control register. Channels 0-7 (U6), for example, are ready when bit 0 at address *base+8* is 0. The data can now be read at the appropriate lo and hi byte addresses for the lo and hi bytes of the channel selected (A2, A1, A0 of the control register). In the case of channels 0-7, addresses *base+0* and *base+1*, respectively, are read.

D. Input Data Format

Unipolar Mode - the output data format is binary. In this case, 0 volts input yields 000H, where H denotes a hexadecimal number. The value increases linearly from 000H to FFFH with increasing input voltage. A +FS input (full-scale +5V or +10V ranges) gives FFFH.

Bipolar Mode - the output data format is twos-complement binary. In this case, a -FS input (±5V or ±10V ranges) yields 800H. The value increases linearly from 800H toward FFFH as the input voltage changes from -FS toward 0 volts.

At 0 volts input (in the ideal case), the value is 000H. Again, the value increases linearly from 000H toward 7FFH as the input voltage changes from 0 toward +FS volts.

The lo byte read, bits D0 thru D7, of the input data is the low byte B0 thru B7 of the conversion result. The hi byte read of the input data contains bits B8 thru B11 of the conversion result in bits D0 thru D3. Bits D4 thru D7 contain all 0's and all 1's, respectively, for the unipolar and bipolar modes.

E. Power-Down Modes

To save power, the converters can be placed into a low-power shutdown mode between conversions. Two programmable power-down modes are available. Select STDBYPD or FULLPD by programming PD0 and PD1 in the control register. When software power-down is asserted, it becomes effective only after the end of conversion. In both power-down modes, the interface remains active and conversion results may be read. Input overvoltage protection is active. The converter returns to normal operation on the first write to the control register. In STDBYPD each device typically consumes 700uA and in FULLPD 120 uA maximum.

The converter voltage reference remains active in STDBYPD. This is a DC power state that does not degrade after power-down of any duration and any sampling rate can be used without regard to start-up delays. In FULLPD, however, start-up delays will effect the conversion. It is recommended when using this mode that a STDBYPD power-down cycle be performed prior to starting conversions to allow the reference voltage to stabilize.

Selecting STDBYPD on every conversion automatically shuts the MAX197 down after each conversion without requiring any start-up time on the next conversion.

F. Example BASIC Program

A simple BASIC program that inputs continually inputs channels 0 thru 7 for the 5V unipolar mode and lists the results to the console is given below.

```
10 BASEADDR=&H300 'insert jumpers A4 thru A7, A10 thru A15
20 CBYTE = &H40 ‘Control Byte for 5V Unipolar Mode
30 FOR I = 0 TO 7
40 OUT BASEADDR,CBYTE+I ‘Write Control Byte
50 WHILE (INP(BASEADDR+8) AND 1)=1:WEND ‘Test Status
60 X=INP(BASEADDR) ‘Read LO Byte
70 Y=INP(BASEADDR+1) ‘Read HI Byte
80 X=X+256*(Y AND &HF) 'Mask off 4 MSB's of HI Byte
90 ‘Print CH0-CH4, Linefeed, CH4-CH7
100 IF I=3 OR I=7 THEN PRINT HEX$(X) ELSE PRINT HEX$(X),
110 NEXT
120 PRINT” “
130 GOTO 30 'Go Again
```

G. Example 'C' Program Sequence

For a simple 'C' program illustration using software polling of the device status, consider a case with the following parameters and events.

1) A base address for the card of 7FF0H (insert jumper A15).
2) Read A/D channel 9 (input to U7) in the +5V unipolar mode and store the result in CH_9_INPUT.
3) Read A/D channel 27 (input to U9) in the ±10V bipolar mode and store the result in CH_27_INPUT.

A simple program sequence for this operation is

```
/* Constant declarations */

     #define base_address       0x7ff0     /* card base address */
     #define control_byte_5     0x40       /* control byte for +5V range */
     #define control_byte_5B    0x48       /* control byte for ±5V range */
     #define control_byte_10    0x50       /* control byte for +10V range */
     #define control_byte_10B   0x58       /* control byte for ±10V range */
     #define delay_count        1000       /* delay count for converter
                                              time-out */
```

```
/* Memory assignments */

      int A_D_value, CH_9_INPUT, CH_27_INPUT;

/* Routine to input A/D channel CHAN(0-31) for control byte C_BYTE and
   returns 0 on a converter time-out error. Stores converted value in
   A_D_value */
int  input_A_D( int CHAN, int C_BYTE )
{
      int converter_error, a, i, ch_group;

      if( 0 <= CHAN && CHAN < 8 ) ch_group =0;
      else if( 7 < CHAN && CHAN < 16 )
            {ch_group = 2; CHAN = CHAN - 8;}
      else if ( 15 < CHAN && CHAN < 24 )
            {ch_group = 4; CHAN = CHAN - 16;}
      else if ( 23 < CHAN && CHAN < 32 )
            {ch_group = 6; CHAN = CHAN - 24;}
      outp( base_address+ch_group, C_BYTE + CHAN );//write control byte
      if( ch_group < 1 ) a = 1;
      else if( ch_group <  3 ) a = 2;
      else if( ch_group <  5 ) a = 4;
      else a =  8;
      i=0;
      do ++i;
      while ( (inp(base_address + 8) & a) && i < delay_count );
      if( i == delay_count ) converter_error = 1;/* converter time-out error */
      else converter_error = 0;
      A_D_value = inp( base_address + ch_group ); /* get converter value */
      A_D_value = A_D_value + ((inp(base_address + ch_group + 1) &
            0xf)<<8);
      return(  converter_error  );
}

void  main( void)
{
            .
            .
      /* Input channel 9 for +5V range and store if no time_out error */
      if( !input_A_D( 9, control_byte_5 ) ) CH_9_INPUT = A_D_value;;

      /* Input channel 27 for ±10V range and store if no time_out error */
      if( !input_A_D( 27, control_byte_10B) ) CH_27_INPUT = A_D_value;;
            .
            .
}
```

The function *input_A_D(int CHAN, int C_BYTE)* above is written in general terms to permit calls from the main routine or from other user defined functions by simply using the appropriate CHAN and C_BYTE values for the input channel desired and the desired input range.

APPENDIX

Circuit Diagrams

MSI-P412