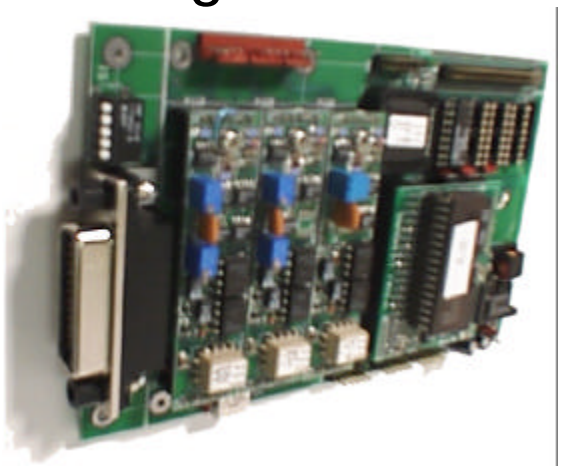
Technical Manual

SBC488A Single Board Computer



and

ADAIO I^2C Analog/Digital IO Modules

TIDAL ENGINEERING
2 EMERY AVENUE
RANDOLPH, NJ 07869
Tel(973)328-1181 Fax(973)328-2302

Date: 11 May 1998 Rev. C

P/N TE1122

CHANGES

Original 18 July 1997
Rev A 29 July 1997
1. Corrected drawing showing polarity of address dipswitch.
2. Corrected Quick basic examples.
Rev B 4 January 1998
1. Updated Address.
2. Changed Digital I/O to 32 bit from 16 bits.
Rev C 11 May 1998
1. Added Forward and Reference Sections.
Rev D 14 June 1999
1. Corrected Figure on Page 10.

FORWARD

This technical manual describes the operation, programming and maintenance of Tidal Engineering's SBC488A Single board computer and ADAIO Analog Input and Output modules.

Additional technical manuals concerning the SBC488A are:

SBC488A Development Kit Manual (P/N TE1123)
SBC488A-SCPI Technical Manual (P/N TE1120)
SBC488A-SCPI Development Kit manual (P/N TE1121)

These manuals are provided with the Development Kits for users who are developing custom applications for the SBC488A.

The MT488A manual describes the use of the SBC488A in controller applications.

REFERENCE

The following documents are referred to throughout this document:

Document	Title	Notes
IEEE Std 488.1-1987	IEEE Standard Digital Interface for Programmable Instrumentation	1
IEEE Std 488.2-1992	IEEE Standard Codes, Formats, Protocols, and Common Commands for use with IEEE Std 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation	
SCPI 1997	Standard Commands for Programmable Instruments	2,3

1.0 Overview

This manual outlines the specification for a power supply controller incorporating IEEE 488 communication capabilities. The design specifications are listed below for a 0 to 50°C operating temperature range.

Analog Inputs (2)

Digital I/O

- 32 Bits Input or Output (Bi-directional)
- TTL levels

Communications

- IEEE 488 or RS-232
- Address Range; 5 bits, 32 addresses (Dip switch and EEPROM programmable)

Power Supply

- 12-36 VDC, 3.5 VA

Size

- 4.00" X 6.55" max.

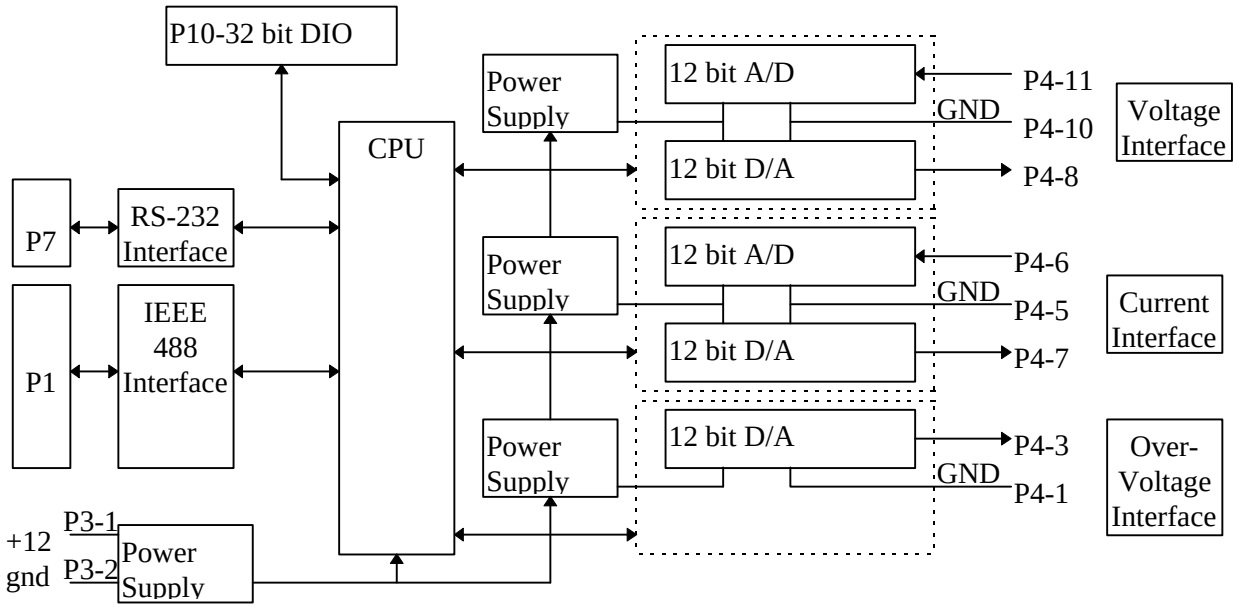


Figure 1 SBC488A with optional D/A's shown

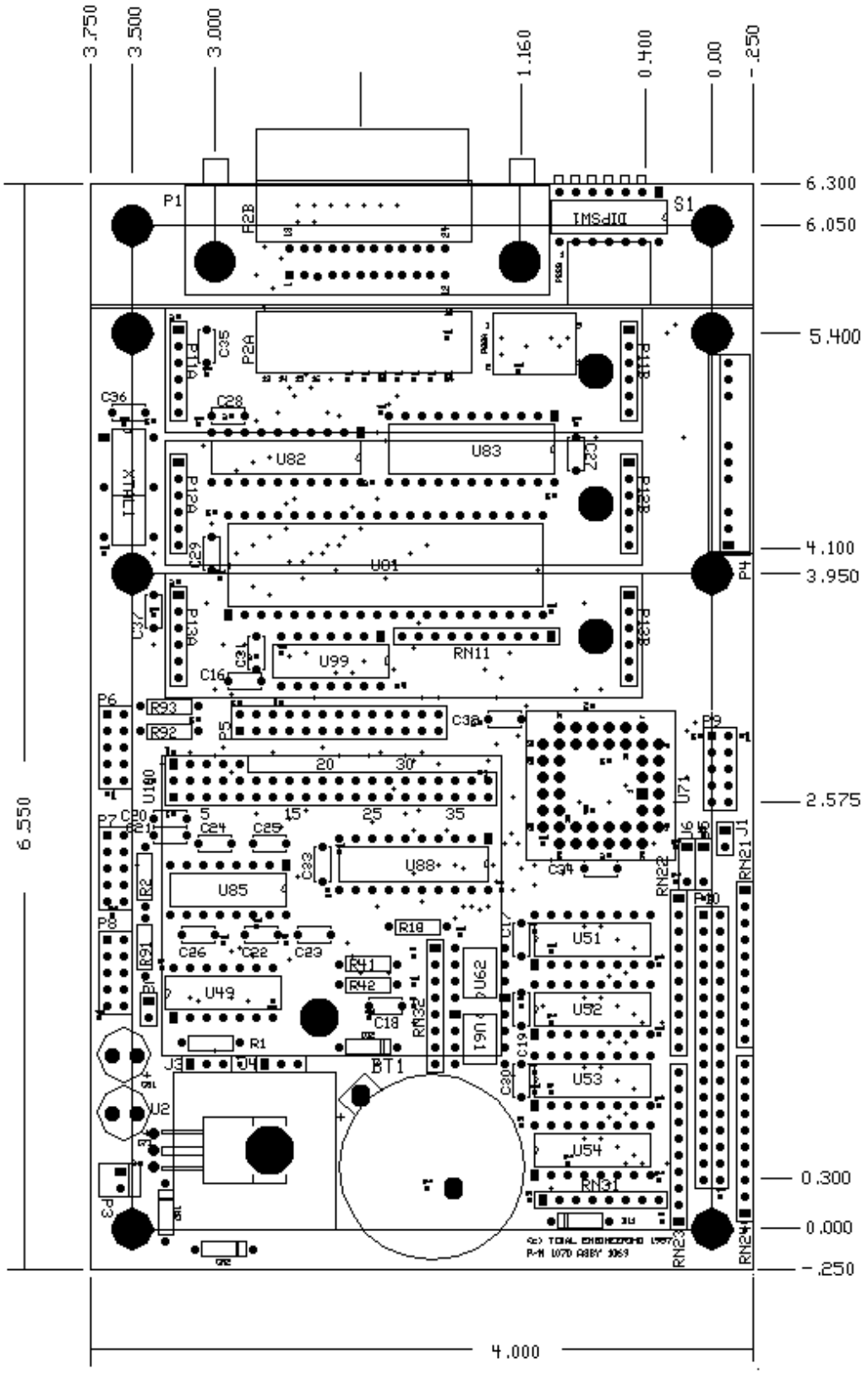


Figure 2 SBC488A Outline Drawing

2.0 Pin Descriptions

a. P1 IEEE-488 Connector

Pin	Name/Description	Pin	Name/Description
1	DIO1/Data line	13	DIO5/Data line
2	DIO2/Data line	14	DIO6/Data line
3	DIO3/Data line	15	DIO7/Data line
4	DIO4/Data line	16	DIO8/Data line
5	EOI/End or Identify	17	REN/Remote Enable
6	DAV/Data Valid	18	DAV Gnd
7	NRFD/Not Ready For Data	19	NRFD Gnd
8	NDAC/Not Data Accepted	20	NDAC Gnd
9	IFC/Interface Clear	21	IFC Gnd
10	SRQ/Service Request	22	SRQ Gnd
11	ATN/Attention	23	ATN Gnd
12	Shield	24	Ground

g. P10 Digital I/O Connector

Pin	I/O	Name/Description	Voltage Level
1	O	+5 VDC Power	+5 VDC
2	I/O	D1-LSB Data 0 I/O	+5 VDC
3	I/O	D1 Data 1 I/O	+5 VDC
•	•	•	•
•	•	•	•
10	I/O	D2-LSB Data 0 I/O	+5 VDC
11	I/O	D2- Data 1 I/O	+5 VDC
•	•	•	•
•	•	•	•
18	I/O	D3-LSB Data 0 I/O	+5 VDC
19	I/O	D3 Data 1 I/O	+5 VDC
•	•	•	•
•	•	•	•
26	I/O	D4-LSB Data 0 I/O	+5 VDC
27	I/O	D4- Data 1 I/O	+5 VDC
•	•	•	•
•	•	•	•
34	O	Power Return	0 VDC

The P3 Connector pinout is as follows:

PIN	Power
1	Positive Voltage Supply (8-20VDC)
2	Ground

RS-232 Connections: P6 for software developmentand P7 for communications

Connector Pinout is as follows:

PIN	RS-232
1	
2	TX
3	RX
4	
5	GND
6	
7	CTS
8	RTS
9	

The P8 RS-485/RS-422 Connector Pinout is as follows:

PIN	RS-422	RS-485
1	TD(-)	TD(-)
2	RD(+)	RD(+)
3	RD(-)	RD(-)
4		
5	GND	GND
6		
7	CTS	CTS
8	RTS	RTS
9	TD(+)	TD(+)

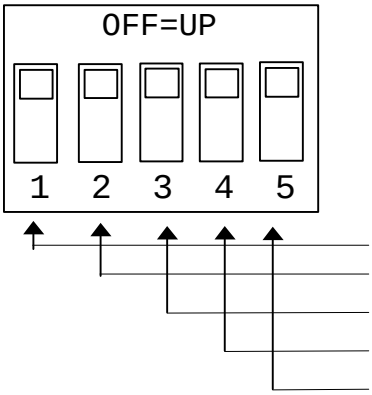
The P4 Analog I/O Connector Pinout is as follows:

PIN	Signal	Voltage
12	DAC 0	0-10VDC
11	ADC 0	0-10VDC
10	Chan 0	GND
7	DAC 1	0-10VDC
6	ADC 1	0-1VDC
5	Chan 1	GND
3	DAC 2	0-10VDC
2	ADC 2	0-10VDC
1	Chan 2	GND

3.0 Indicators and Controls

a. Address switch

The IEEE-488 Address can be programmed via the 5 position dip switch on the rear of the SBC488A. The address is configured as follows:



RS-232 Mode

The SBC488A is in RS-232 mode when all the switches are UP (off).

Weights		**Switch**
Off	*On*	*No.*
16	0	1
8	0	2
4	0	3
2	0	4
1	0	5

Figure 3, IEEE-488 Address Switch

Notes:
1. Cycle SBC488A power after changing dip switch setting.
2. The dip switch settings can be over ridden by software commands which write to the EEPROM. (See the command descriptions that follow in section 5.0).

4.0 SBC488A Part Number Breakdown

The part number for the SBC488A determines the installed options. The part number is created as follows:

SBC488A	-xDA	x.xx	-xAD	-xDIG	-RSxxx
Model number	no. of DAC channels	DAC voltage reference	no. of ADC channels	No. of Digital Ports	Serial Port Option
SBC488A	blank	blank	blank	blank	blank
4″ x 5″	-1DA	2.50	-1AD	-1DIG	-RS232
	-2DA	4.10	-2AD	-2DIG	-RS485
	-3DA	5.00	-3AD		
		10.0			

Part number example:

SBC488A-4DA2.50-3AD-2DIG

This part number is for an SBC488A single board computer with the OEM firmware and the following hardware options:

- Four DACs with a 0-2.5 VDC full scale range.
- Three channels of ADC.
- Four digital I/O ports, 8 bits each for a total of 32 I/O.

5.0 Programming

5.1 Device Specific Commands

The SBC488 is addressable via the IEEE-488 Interface and the RS232 Interface. The SBC488 assumes IEEE-488 communications if the address switch is set in the range of 0 to 30 (Valid IEEE-488 address) and RS232 communication for a setting of 31. The Device specific commands are implemented for both interfaces. The IEEE-488 commands are only available over the IEEE-488 interface. Example programs use IOTech's DRIVER488/DRV driver syntax but should be useful for other drivers.

The following are the device specific commands:

ADC?x Read A/D converter, x is channel number, 0,1 or 2

Example: ADC?0 Measure Channel 0

'QuickBasic Example using a/d read commands

```
'QuickBasic Example using a/d read commands
OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
OPEN "\DEV\IEEEIN" FOR OUTPUT AS #2
'ASSUME SBC488 IS AT ADDRESS 9
FS_VOLTS=16      'FULL SCALE VOLTAGE IS 16 VDC
FS_AMPS=320      'FULL SCALE CURRENT IS 320 ADC
PRINT#1, "OUTPUT9;ADC?0"
INPUT #2, R$
PRINT "OUTPUT VOLTAGE IS "+R$
PRINT#1, "OUTPUT9;ADC?1"
PRINT "OUTPUT CURRENT IS "+R$
```

DAC?x Read DAC converter setting, x is channel number, 0,1 or 2

Example: DAC?0 Read Channel 0 DAC setting

DAC!x,nn.nn Set DAC converter, x is channel number, 0,1 or 2

nn.nn is setpoint

Example: DAC!0,112.3 Set Channel 0 DAC setting

QuickBasic Example using D/A set commands

```
'QuickBasic Example using D/A set commands
OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1  'IOTech command
OPEN "\DEV\IEEEIN" FOR OUTPUT AS #2   'IOTech command
'ASSUME SBC488 IS AT ADDRESS 9
FS_VOLTS=16      'FULL SCALE VOLTAGE IS 16 VDC
FS_AMPS=320      'FULL SCALE CURRENT IS 320 ADC
V_SET=5.5            'VOLTAGE SETTING IS 5.5 VDC
I_SET=100            'CURRENT SETTING IS 100 AMPS
OV_SET=6.5           'SET OVERVOLTAGE TO 6.5 VOLTS
PRINT#1, "OUTPUT9;DAC!0,"+ STR$(V_SET)
PRINT#1, "OUTPUT9;DAC!1,"+ STR$(I_SET)
PRINT#1, "OUTPUT9;DAC!2,"+ STR$(OV_SET)
```

DIG? Read Digital Ports

Example: DIG?

Response:DIG?255,112,122,212

Quick Basic Example

```
'QuickBasic Example using D/A set commands
OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1  'IOTech command
```

```
OPEN "\DEV\IEEEIN" FOR OUTPUT AS #2   'IOTech command
'ASSUME SBC488 IS AT ADDRESS 9

`Set all Digital I/O as inputs (high)
PRINT#1, "OUTPUT9;DIG!0,"+ STR$(255)
PRINT#1, "OUTPUT9;DIG!1,"+ STR$(255)
PRINT#1, "OUTPUT9;DIG!2,"+ STR$(255)
PRINT#1, "OUTPUT9;DIG!3,"+ STR$(255)

' ASSUME DIGITAL I/O IS CONFIGURED AS FOLLOWS
current mode=1
volt mode=2
soft-start mode=4
thermal fault=8
phase balance fault=16
program line fault=32
over-voltage fault=64
standby mode=128

PRINT#1, "OUTPUT9;DIG?"
INPUT #2, R$
PORT=MID(R$,INSTR(R$,'?'), INSTR(R$,',')-INSTR(R$,'?'))
IF (PORT AND currentmode) THEN
   IN_CURRENT_M()
END IF
IF (PORT AND voltmode) THEN
   IN_VOLT_M()
END IF
.
.
```

DIG!x,nnn Set Digital Ports
Example: DIG!0,255 Set all bits on port 0 to +5VDC

CAL:IDNsssssssssssssssssssssssss Set IDN Message
Example: CAL:IDN SBC488A,July 23, 1997, Ver E

ADCG!x,nn.nn Set ADC converter gain, x is channel number, 0,1 or 2
nn.nn is full scale range
Example: ADCG!0,112.3 Set Channel 0 ADCG full scale setting

ADCO!x,nnn Set ADC converter offset, x is channel number, 0,1 or 2
nnn is offset in parts per 4095
Example: ADCO!0,1 Set Channel 0 ADCG offset

DACG!x,nn.nn Set DAC converter gain, x is channel number, 0,1 or 2
nn.nn is full scale range
Example: DAG!0,112.3 Set Channel 0 DACG full scale setting

DACO! Set DACO converter offset, x is channel number, 0,1 or 2
nnn is offset in parts per 4095
Example: DACO!0,1 Set Channel 0 ADCG offset

GPIB!nn Set GPIB Address where nn is 0 to 30
Example: GPIB!,13 GPIB address to 13
When the unit is reset, and the address on the dip switch is not 31, then GPIB Address will be 13.

GPIBEXT Set GPIB Address to dip switch
Example: GPIBEXT
When the unit is reset, and the address on the dip switch will be GPIB Address. (Address 31 is RS-232 Interface)

GPIB? Read the GPIB setting
ExampleL GPIB?
Response: "0" thru "30" or "dip switch".

QuickBasic program using Digital I/O

```
‘QuickBasic Example using digital I/O

OPEN “\DEV\IEEEOUT” FOR OUTPUT AS #1
OPEN “\DEV\IEEEIN” FOR OUTPUT AS #2
‘ASSUME SBC488 IS AT ADDRESS 9
‘Dig? COMMAND POSITIONS
START=1
STOP=2
REM_SHUTDOWN=4
OV_RESET=8
‘D?0 COMMAND BIT POSITIONS
C_MODE=1              ‘current mode
V_MODE=2              ‘volt mode
SS_MODE=4             ‘soft-start mode
OT=8                  ‘thermal fault
PH_BAL=16             ‘phase balance fault
PGM_LINE=32           ‘ program line fault = 32
OV=64                 ‘ over-voltage fault
STDBY=128             ‘ standby mode

PRINT#1, “OUTPUT9;D1,”+STR$(START) ‘HIT START
PRINT#1, “OUTPUT9;D1,0”       ‘RELEASE START
PRINT#1, “OUTPUT9;D1,”+STR$(STOP)  ‘HIT START
PRINT#1, “OUTPUT9;D1,0”       ‘RELEASE STOP
PRINT#1, “OUTPUT9,D?,0”
INPUT #2, R$
IF VAL(R$) AND C_MODE THEN PRINT “CURRENT MODE”
IF VAL(R$) AND V_MODE THEN PRINT “VOLTAGE MODE”
IF VAL(R$) AND SS_MODE THEN PRINT “SOFT START MODE MODE”
IF VAL(R$) AND OT THEN PRINT “OVER TEMPERATURE FAULT”
IF VAL(R$) AND PH_BAL THEN PRINT “PHASE BALANCE FAULT”
IF VAL(R$) AND PGM_LINE THEN PRINT “PROGRAM LINE FAULT”
IF VAL(R$) AND OV THEN PRINT “OVER VOLTAGE FAULT”
IF VAL(R$) AND STDBY THEN PRINT “STANDBY MODE”
```

5.2 IEEE-488 Event Processing

The IEEE-488 standard defines a standard method for status reporting. This method uses the IEEE-488.1 status byte (See Figure 13) Three bits of this byte are defined:

RQS =>Service Request bit
ESB =>Event Status bit
MAV =>Message Available bit

The RQS bit is set when the SBC488 has requested service via the SRQ control line.

The ESB bit is set when one of the events defined in the standard Event Status Register (ESR) has occurred. The ESR is masked by the Event Status Enable register (ESE) so that the user can mask specific or all events from causing an ESB and subsequent RQS. (See Figures 12 and 13)

The MAV bit is set when a message is available in the SBC488 output buffer.

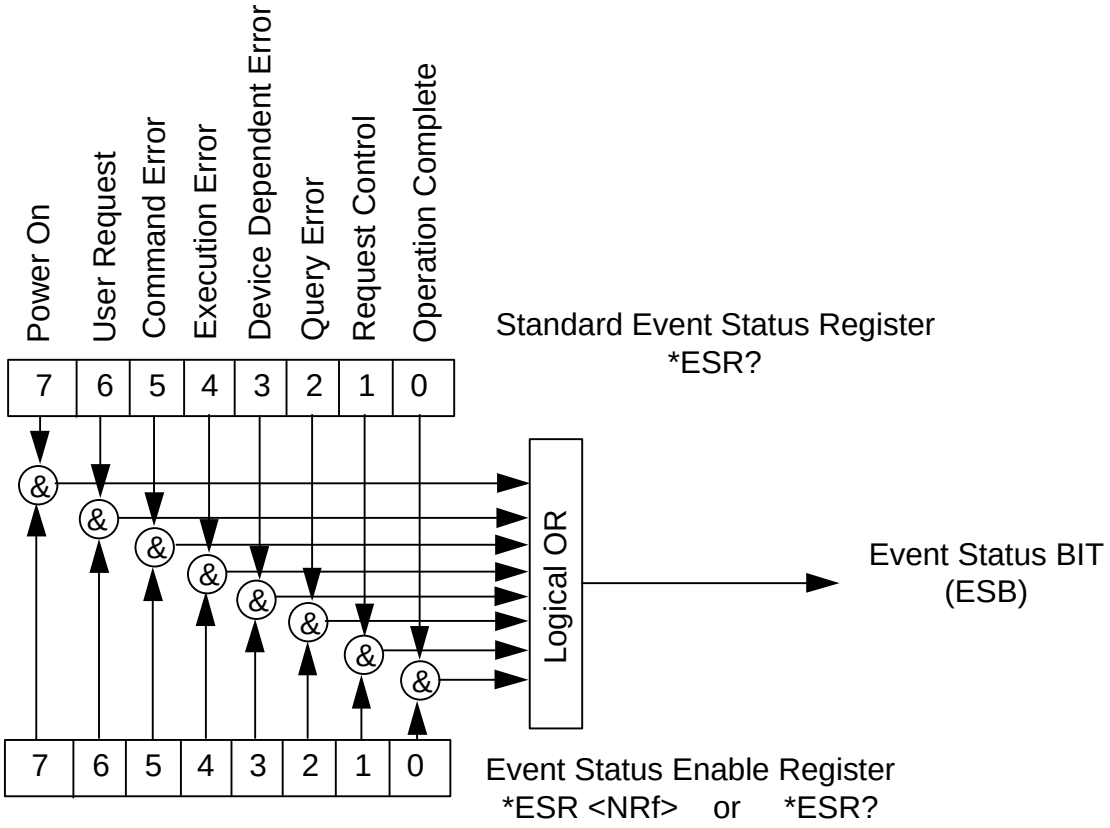


Figure 4, ESE and ESR Generation

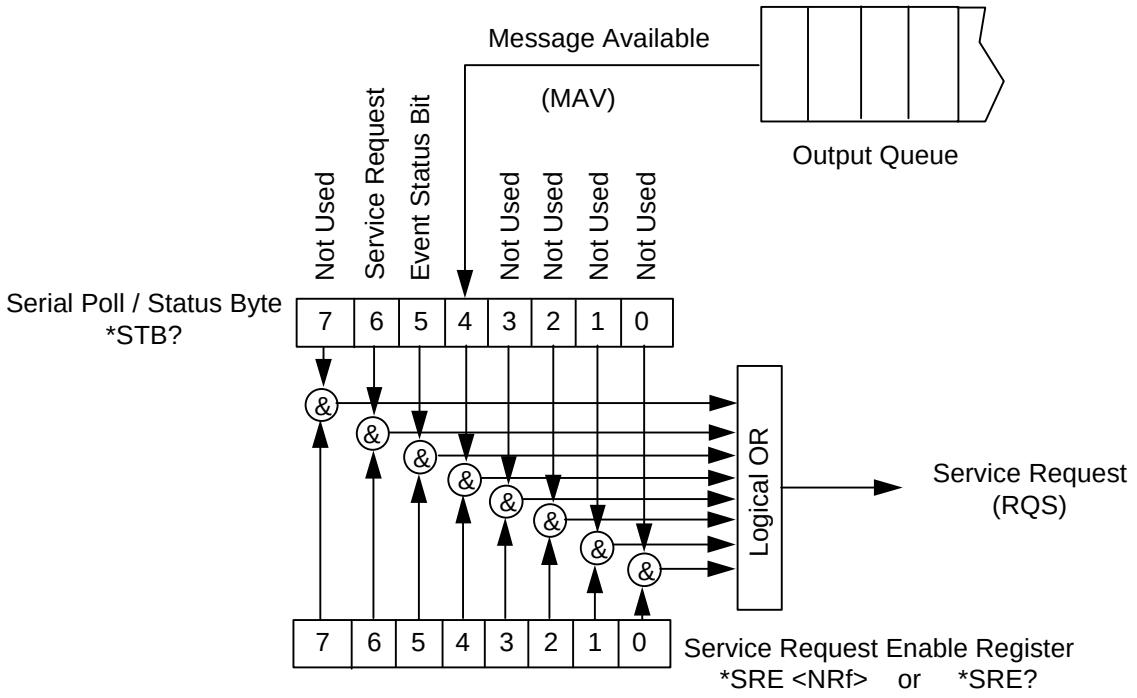


Figure 5, Status Byte Generation

5.3 IEEE-488 Standard Commands

a. Read Event Status Register (See Figure 12)

*ESR?
Response: Decimal number weighted as follows:

1 => Operation Complete
2 => not used (Request Control)
4 => not used (Query Error)
8 => Device Dependent Error
16.=> Execution Error
32 => Command Error
64 => not used (User Request)
128 => Power on , 1 after power on

Note
1. Power on bit is set every time the computer is powered up or after a power failure.
2. All bits are cleared after the *ESR? command is executed.

b. Read Event Status Enable mask (See Figure 12)

*ESE?
Response: Reads *ESE mask. Mask bitwise enables ESR generation. Same format as *ESR? above.

c. Set Event Status Enable mask (See Figure 12)

*ESE,xxx
Response: Sets *ESE mask. Mask bitwise enables ESR generation. Same format as *ESR? above.

d. Read Status byte (See Figure 12)

*STB?
Response: Decimal number weighted as follows:

1 => N/A
2 => N/A
4 => N/A
8 => N/A
16.=> MAV, Message Available
32 => ESB, Event Status Bit
64 => RQS, Request Service
128 => N/A

e. Read Status Register Enable mask (See Figure 12)

*SRE?
Response: xxx Mask bitwise enables ESR generation. Decimal number weighted as follows

1 => N/A
2 => N/A
4 => N/A
8 => N/A
16.=> MAV, Message Available
32 => ESB, Event Status Bit
64 => N/A
128 => N/A

f. Set Status Register enable mask (See Figure 12)

*SRE,xxx
Mask same as *SRE? above

g. Read SBC Part and serial number

*IDN?
Response: SBC488,REVxx where xx is the revision of the firmware.

QuickBasic Program Example using IEEE-488 Standard Commands

```
‘QuickBasic Example using digital I/O

OPEN “\DEV\IEEEOUT” FOR OUTPUT AS #1
OPEN “\DEV\IEEEIN” FOR OUTPUT AS #2
‘ASSUME SBC488 IS AT ADDRESS 9
‘SRE BIT POSITIONS
MAV=16          ‘MAV, Message Available
ESB=32          ‘ESB, Event Status Bit
‘ESE BIT POSITIONS
DDE=8           ‘Device Dependent Error
EE=16           ‘Execution Error
COMERR=32       ‘ Command Error
PON=128         ‘ Power on        , 1 after power on

‘GET SBC488 VERSION NUMBER
PRINT#1, “OUTPUT9,*IDN?”
INPUT #2, R$
PRINT “FIRMWARE VERSION NUMBER ”+R$

‘ENABLE  MAV AND ESB IN SRE AND READ BACK
PRINT#1, “OUTPUT9,*SRE,”+STR$(MAV+ESB)
PRINT#1, “OUTPUT9,*SRE?”
INPUT #2, R$
PRINT “SERVICE REQUEST ENABLE REGISTER IS SET TO”+R$

‘ENABLE PON AND COMERR IN SRE AND READ BACK
‘POWER ON AND COMMAND ERRORS WILL PRODUCE SERVICE REQUESTS
PRINT#1, “OUTPUT9,*ESE,”+STR$(PON+COMER)
PRINT#1, “OUTPUT9,*ESE?”
INPUT #2, R$
PRINT “EVENT STATUS ENABLE REGISTER IS SET TO”+R$
```

6.0 Troubleshooting

FAULT CONDITION	PROBABLE CAUSE	CHECK, REMEDY
No IEEE-488 Communications	Address errors Note 3	SBC488 Dip switch. Valid addresses are 0 thru 30 Control program address
	Termination	Check control program termination, CR, LF and EOI
	IEEE-488 Cable	Replace Cable
	SBC488 Power Supplies	Check power supplies on SBC488 (U7).
	SmartCore Failure	Check data bus for activity
	7210 Chip failure	Check Interrupt line out of 7210 during transmission.
No RS-232 Communications	Address errors Note 3	SBC488 Dip switch. Valid address is 31.
	Format error	Terminal program set for 9600,N,8,1
	RS-232 Cable	Replace Cable
	SBC488 Power Supplies	Check power supplies on SBC488 (U7).
	SmartCore Failure	Check data bus for activity
	RS-232 transceiver Chip failure	Check U85 RX and TX lines during transmission
D/A out voltage error	D/A power supply	Check D/A power supply voltage (U1 and U2)
	Opto-Isolator failure	Check U41 and U42 outputs while commanding voltage change.
	Voltage reference	Check U21 for 2.50 VDC
	D/A IC failure	Replace U11
	Control software	Check data for errors
A/D Readback error note 1	A/D power supply	Check A/D power supply voltage (U5 and U3)
	Opto-Isolator failure	Check U43, U46 and U45 outputs while reading voltage over bus (AD?0).
	Voltage reference	Check U22 for 2.50 VDC
	A/D IC failure.	Replace U90
	Control software	Check data for errors

FAULT CONDITION	PROBABLE CAUSE	CHECK, REMEDY
Digital Readback error	Digital power supply	Check power supply voltage (U6)
	Opto-Isolator failure	Check U48 and U47 outputs while reading digital over bus (D?0).
	Control software	Check data for errors
	Port IC failure	Replace port IC U51, U52 or U53
	Voltage levels	Verify input voltage levels are valid (See section 2)
Digital set error	Digital power supply	Check power supply voltage (U6)
	Opto-Isolator failure	Check U48 and U47 outputs while reading digital over bus (D?0).
	Control software	Check data for errors
	Port IC failure	Replace port IC U51, U52 or U53
	load current note 2	Verify load current levels are valid (See section 2)