

# SensorWatch™

A C/C++ Programmable 16-bit Controller  
for  
Data Acquisition, Test, or Control

## *Technical Manual*



1950 5<sup>th</sup> Street, Davis, CA 95616, USA

Tel: 530-758-0180

Fax: 530-758-0181

Email: [sales@tern.com](mailto:sales@tern.com)

<http://www.tern.com>

## COPYRIGHT

V25-Engine, BirdBox, PowerDrive, TERN\_PDC, Flash, and SensorWatch are trademarks of TERN, INC.

V25 is a Trademark of NEC Electronics Inc.

Turbo Debugger, Turbo C, and Borland C++ are trademarks of Borland International.

Microsoft, MS-DOS and Windows are trademarks of Microsoft Corporation.

IBM is a trademark of International Business Machines Corporation.

Paradigm LOCATE, DEBUG/RT-V25 and PDREM are trademarks of Paradigm Systems.

Version 2.00

October 29, 2010

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of TERN.



© 1994-2010

1950 5<sup>th</sup> Street, Davis, CA 95616, USA

Tel: 530-758-0180 Fax: 530-758-0181

Email: [sales@tern.com](mailto:sales@tern.com)

<http://www.tern.com>

### Important Notice

***TERN*** is developing complex, high technology integration systems. These systems are integrated with software and hardware that are not 100% defect free. ***TERN products are not designed, intended, authorized, or warranted to be suitable for use in life-support applications, devices, or systems, or in other critical applications.*** ***TERN*** and the Buyer agree that ***TERN*** will not be liable for incidental or consequential damages arising from the use of ***TERN*** products. It is the Buyer's responsibility to protect life and property against incidental failure.

***TERN*** reserves the right to make changes and improvements to its products without providing notice.

Temperature readings for controllers are based on the results of limited sample tests; they are provided for design reference use only.

---

# Chapter 1

## Introduction

### 1.1 Functional Description

The SensorWatch™ is designed for data-acquisition and control applications. Measuring 4.3x5.7x1 inches, the SensorWatch™ features analog signal conditioning circuit, eight channels of 12-bit ADC, seven channels of comparator inputs, three serial ports (RS-232/RS-485), one channel of 10-bit DAC, seven solenoid drivers, three 16-bit high speed (upto 10MHz) counters, seven digital inputs, eight digital outputs, one power relay, a PDC (Portable Data Carrier) interface, 4x4 keypads, character or graphic LCD interfaces, beeper, LEDs, and on-board power supplies. The SensorWatch™ is an I/O board driven by a V25-Engine™ or a C-Engine™. A functional block diagram is shown in Fig. 1.1.

The analog signal conditioning circuit provides configurable gain and high input impedance to the input analog signals, allowing direct interface to low-voltage output sensors, such as EKG electrodes, thermocouples, or strain gauges. Six Schmitt-trigger inverters are provided for high speed counter/timer inputs and digital inputs, to increase noise immunity and transform slowly-changing input signals to fast-changing and jitter-free signals. A high-performance programmable counter chip ( $\mu$ PD71054) includes three 16-bit counters (up to 10 MHz) and 6 software programmable modes. With a multiple latch command, the content of the status register and 3 counter reading can be latched simultaneously without interfering the high speed input counting. A serial 12-bit ADC with internal track-and-hold is on-board. Three LCD interface headers allow direct interface to a character LCD (16 x 2, or 20 x 4, or 40 x 4) or to a graphics LCD (240 x 64 dots). An RS-485 port for the UART (SCC2691) supports multiprocessor networking. The low power version of the SensorWatch™ consumes only 65 mA at full speed, 35 mA standby and 5 $\mu$ A in the power\_off mode. An optional instrumentation Op circuit provides a configurable high gain (upto 1000) control and a high input impedance ( $>10^{12} \Omega$ ) for channel 1 and channel2, allowing directly interface with low voltage output sensors, such as ECG stress test electrodes, thermocouples, or strain gauges. By default, all analog signal input voltage levels are 0 to 5 V. The six high speed Schmitt-trigger inverters are provided for the 3 high speed counter/timer inputs and 3 digital inputs (SIN0, SIN1, and SIN2 at the terminal block T2, Fig. 1.2).

You program the SensorWatch™ from your PC via serial link. You can use your favorite Borland or Microsoft C/C++ compilers. With Paradigm LOCATE support, the C/C++ program can be remotely debugged on the SensorWatch™ with Paradigm DEBUG over the serial link, at 115,000 baud rate. TERN provides I/O driver libraries, sample programs, target EPROMs, batch files, and all the hardware necessary for you to quickly develop your application software. For more details please see the V25-Engine™ Technical Manual.

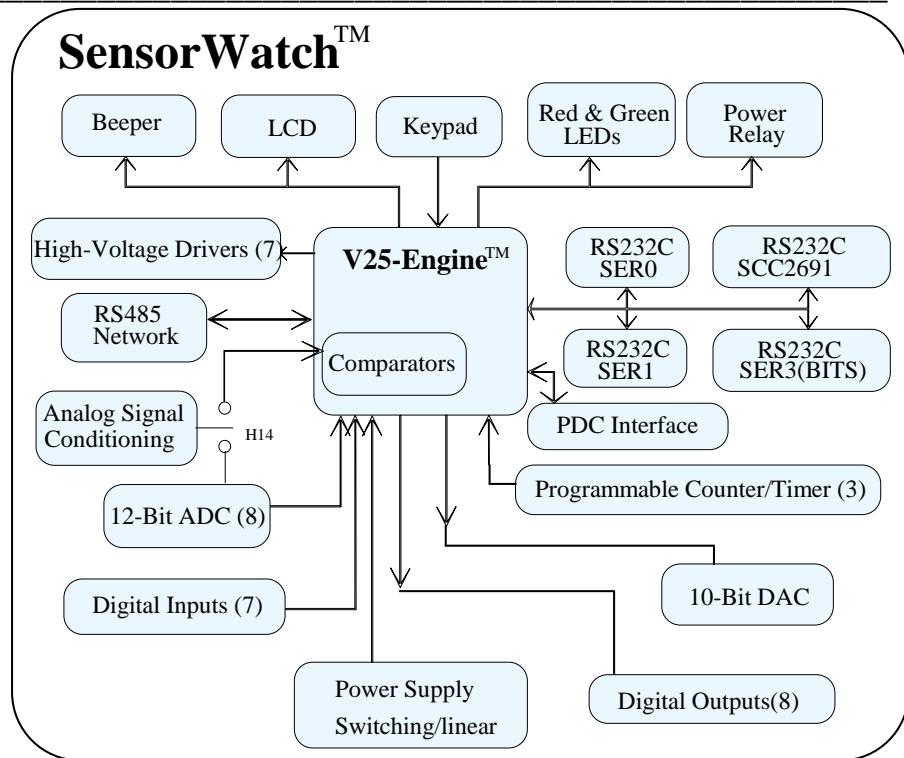


Fig. 1.1. Functional block diagram of the SensorWatch™

The comparator inputs of the V25-Engine™ can be used to measure either digital inputs or analog signals with a variable reference voltage. The high speed 10-bit DAC can provide 0-5V voltage output. A power relay provides a normal-open, normal-close, and common terminals which carries at least 5A (available at the terminal block T2, pins 1, 2, and 3). A PDC (Portable Data Carriers) interface is designed for accessing external EEPROMs which can be used as portable data carriers. Three LCD interface headers are on board, which allows directly connection with LCDs (16 character x 2 lines, 20 characters x 4 lines, 40 characters x 4 lines, or 240 x 64 dots graphics). A SCC2691 UART with an RS485 driver supports high speed multiprocessor networking. The on-board DC power supply circuits provides +5V, -5V, -9V, +7V, and +9V with a single unregulated DC input (9-12V, 500 mA).

### 1.2 Features

- Measures 5.7 x 4.3 x 1 inches
- Power consumption: 130 mA full speed. 3  $\mu$ A in the power-off mode with switching power supply
- Low power version: 65 mA full speed, 35 mA standby.
- 32K EPROM, 32K SRAM (up to 512K). 256 bytes built-in RAM, DMA, 5 external interrupts
- 24 digital I/O lines in the V25 CPU. Two 16-bit timers, 16-bit time base counter
- Seven channels of 10-bit analog inputs
- Two RS-232 channels for the V25 internal serial ports. One SCC2691 UART for RS-485 networking
- Real-time clock RTC72421, lithium coin battery. Serial EEPROM 512 bytes (up to 8KB)
- Supervisor chip (691) for power failure, watchdog timer
- Seven solenoid drivers
- Three 16-bit high speed (10MHz) timer/counters
- Seven digital inputs and eight digital outputs
- One channel of 10-bit DAC

- One power relay (5-10A), 44-position terminal block
- Eight channels of 12-bit ADC and □□ analog signal conditioning circuit.
- Character (16x2,20x4,40x4) or graphic (240x64) LCD
- Interrupt-driven 4 x 4 keypad, □beeper, reset□, and LEDs

1.3 Physical Description

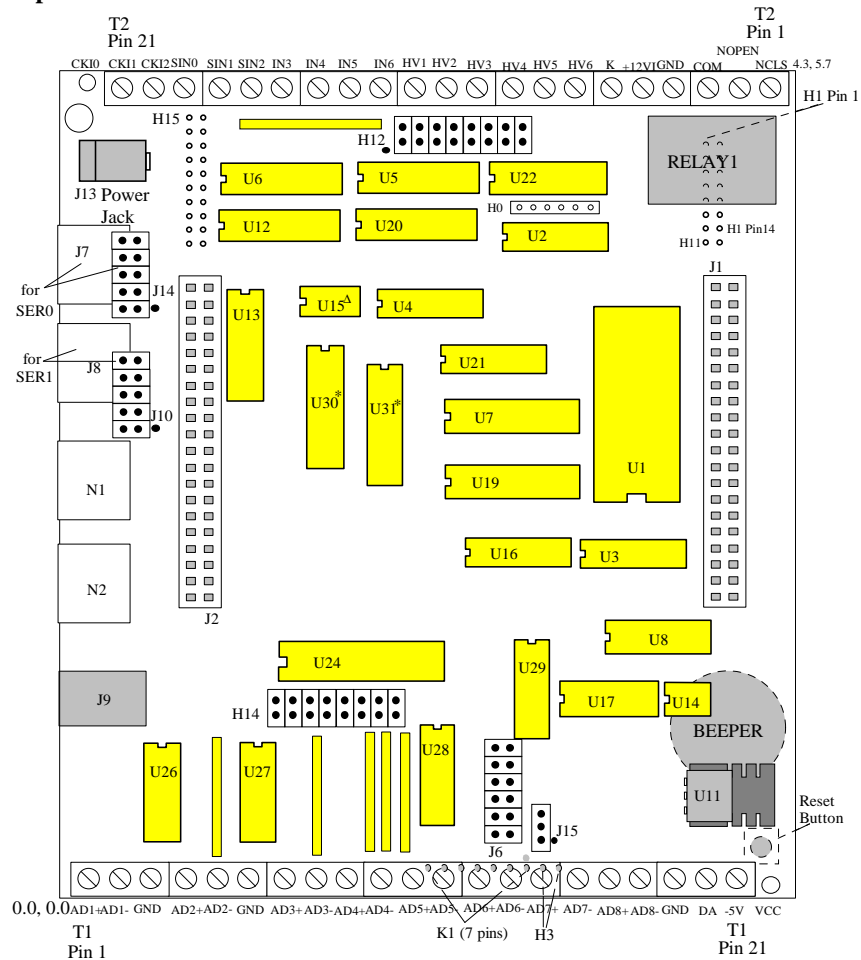


Fig. 1.2 shows the physical layout of the SensorWatch™ before a V25-Engine™ is installed.

List 1.1. Components used in the SensorWatch™

Part Name	Description
U1	P8254-2 or μPD71054, 16-bit programmable timer/counter.
U2, U3, U4, U5	74HC259, 8-bit addressable latches
U6	74HC14, Schmitt-trigger
U7	PAL16V8
U8	74HC05, open-collector inverter
U9, U10	TK112xx, regulators with on/off switches
U11	LM2575, switching regulator
U12, U31	1488 or 75C188, RS232C transmitters
U13, U30	1489, RS232C receivers
U14	ICL7662, negative voltage generator
U15	75176/LTC485, RS485 driver
U16	DAC08, 8-bit DAC, element to constitute 10-bit DAC

U17	LM723, +7V precise voltage regulator
U18	LM79L05, -5V regulator
U19	74HC273, 8-bit latch
U20	74HC257, input multiplexer
U21	74HC138, decoder
U22	ULN2003, high voltage/current driver
U24	MAX186/LTC1294, 12-bit serial ADC
U26, U27, U28, U29	LM324A, Operational Amplifiers

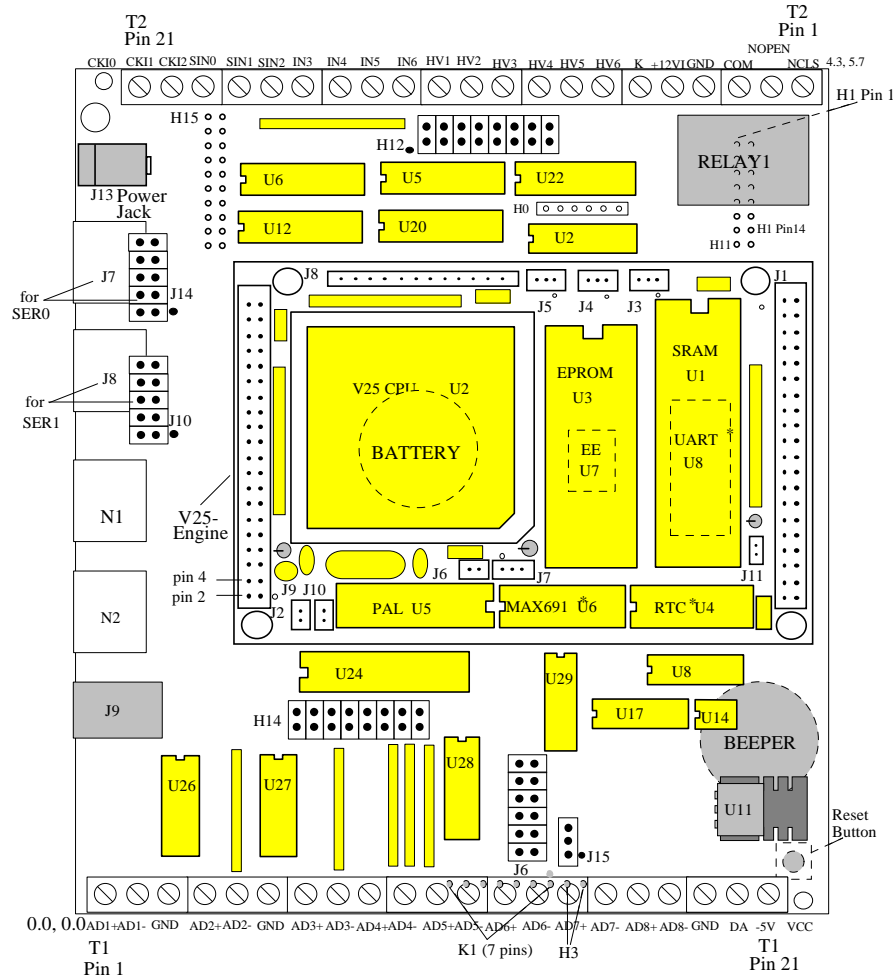


Fig. 1.3 Physical layout of the SensorWatch™, after a V25-Engine™ is installed

**1.4 Minimum Hardware and Software Requirements**

- A SensorWatch™ plus a V25-Engine™ with TERN EPROM (V25-Engine-0-xxx-E);
- A serial cable with a DB9 connector and an 5x2 IDC connector;
- A center negative wall transformer (+9 V, 500 mA);
- A PC or PC compatible computer;
  
- Microsoft C/C++ 8.0/7.0/6.0/5.1 or Borland C/C++ 3.1/3.0/2.0, or Turbo C/C++ 3.0;
- Paradigm DEBUG/RT-V25 and Paradigm LOCATE.
- TERN System Disk.

---

# Chapter 2

## Installation

The installation procedure presented here assumes that the user already established Borland C/C++, Microsoft C/C++, or Turbo C/C++ environment on a PC.

### 2.1 Software Installation

Refer to the Chapter 2 of the V25-Engine™ manual for the details of Software Installation.

### 2.2 Hardware Installation

Connect the 5x2 IDC connector of the RS232 serial cable to the J14(or J7) for using SER0 debugging. Pay attention to the red side of the cable corresponding to pin 1 of J14 (a small circle close to J14 indicates pin number 1), and connect the DB9 connector of the serial cable to COM1 or COM2, the PC serial port, as shown in Fig. 2.1. Connect the output of the center negative wall transformer (+9V DC) to the SensorWatch™ DC power jack J13.

### 2.3 Simple Test

You can run a simple sample program "**led.c**" under your default directory to test the software and hardware installation. The LED should blink if everything is fine. The procedure for the simple test program "**led.c**" is described in the V25-Engine™ Technical Manual (Chapter 3).

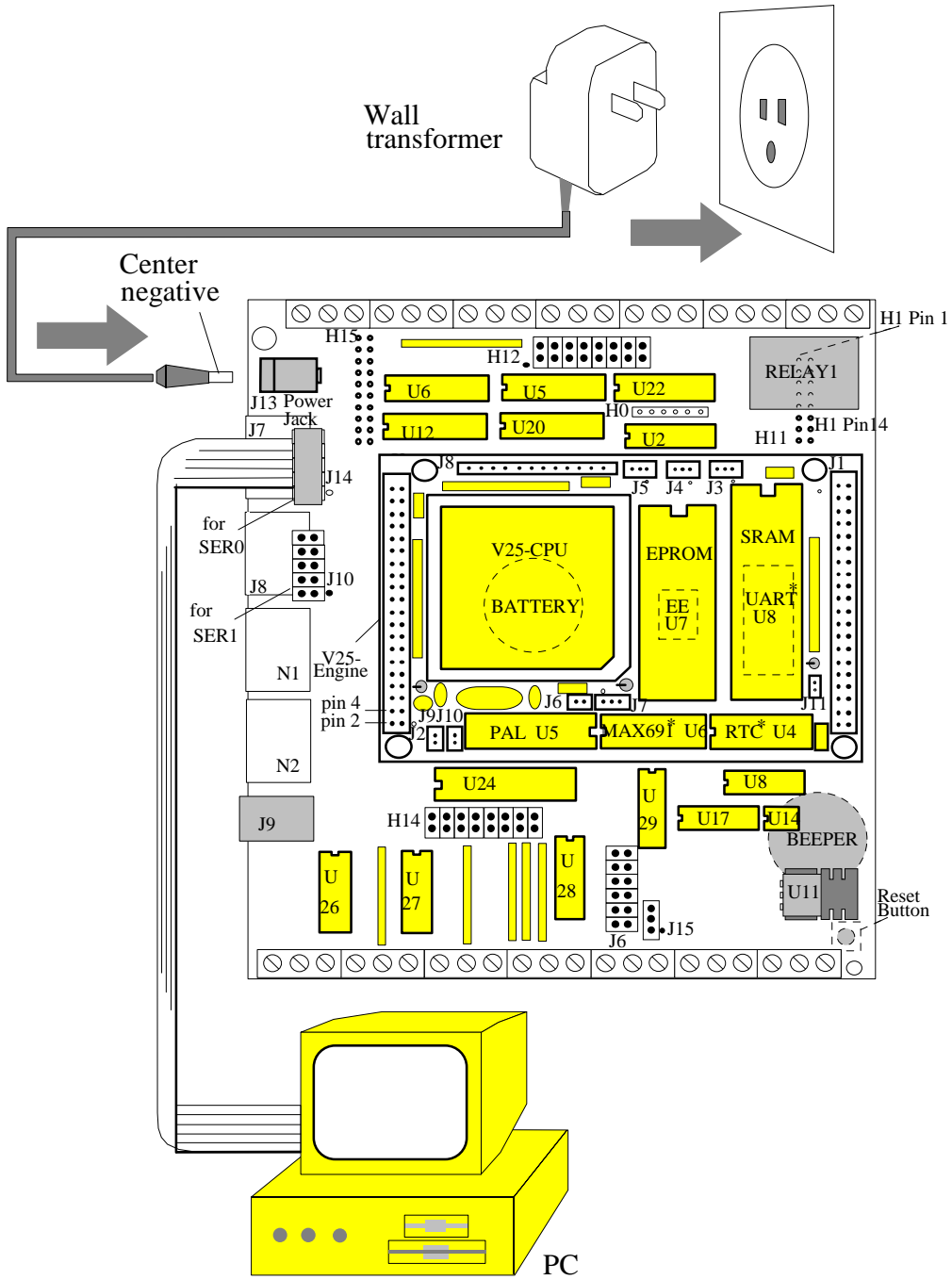


Fig. 2.1 Hardware Installation for the SensorWatch™



## Chapter 3 Hardware

### 3.1 V25-Engine™

The SensorWatch™ is an I/O board for the V25-Engine™ or C-Engine™. The SensorWatch™ must be driven by a V25-Engine™ or a C-Engine™. Please refer to the V25-Engine™ Technical Manual for details.

### 3.2 V25 I/O mapping

The I/O pins of the V25 CPU used in the V25-Engine™ are listed below:

P00	OUTPUT	V25-Engine™ EEPROM (U7) clock SCL
P01	I/O	V25-Engine™ EEPROM (U7) data SDA
P02	INPUT	V25-Engine™ J2 pin4. If low, V25 runs code starting at 0800:0000
P03	OUTPUT	HWD (Hit Watch Dog)
P04	INPUT	WDO (Read watchdog output), if low, watchdog time-out reset.
P05	OUTPUT	V25-Engine™ on-board LED control
P06	INPUT	
P07	OUTPUT	CLKOUT, 8 MHz, V25-Engine™ J1 pin4
P10	NMI	Connected to PFO of MAX691, if J10 jumper is on.
P11	INTP0	External Interrupt input, falling edge active, VE J2.8 and SW J9.5.
P12	INTP1	External Interrupt input, falling edge active, VE J2.6 and SW H0.3.
P13	INTP2	External Interrupt input, falling edge active, VE J2.33 and SW U7.19.
P14	OUTPUT	RTS1 for SER1
P15	INPUT	SW H0.5
P16	OUTPUT	RTS0 for SER0
P17	RDY	V25-Engine™ ready signal, used for more wait states
P20	DR0	V25-Engine™ DMA channel 0 request, active high, VE J2 pin30
P21	DA0	V25-Engine™ DMA channel 0 Ack, active low, if VE J9 jumper is on
P22	INPUT	V25-Engine™ J2 pin38. If low, V25 runs code starting at 0400:0000
P23	OUTPUT	EN485 for SCC RS485 driver, if low, receiving
P24	INPUT	Keypad ROW 2
P25	INPUT	Keypad ROW 3
P26	INPUT	Keypad ROW 1
P27	INPUT	Keypad ROW 4
PT0	INPUT	SW H14 pin 1.
PT1	INPUT	SW H14 pin 2.
PT2	INPUT	SW H14 pin 3.
PT3	INPUT	SW H14 pin 4.
PT4	INPUT	SW H14 pin 5.
PT5	INPUT	SW H14 pin 6.
PT6	INPUT	SW H14 pin 7.
PT7	INPUT	V25-Engine™ PFI (Power Failure Input, if PFI<1.3V, reset)

### 3.3 The SensorWatch™ I/O Map

**Table 3.1 Writing External I/O Registers**

Address	Data bits	Signal	Function
---------	-----------	--------	----------

0xc0	D0-D7	/DAC	Write D0-D7 to 8-bit DAC. Power on or reset DA=0. If outputb(0xc0,0); DA=0V. If outputb(0xc0,0xff); DA=5V
0xc8	D0	/W08	Write D0 to digital output O7=H12 pin1. Power on or reset O7=0. If outputb(0xc8,0); O7=0. If outputb(0xc8,0); O7= +5V.
0xc9	D0	/W08	Write D0 to digital output O6=H12 pin3. Power on or reset O6=0. If outputb(0xc9,0), O6=0. If outputb(0xc9,1), O6= +5V.
0xca	D0	/W08	Write D0 to digital output O5=H12 pin5. Power on or reset O5=0. If outputb(0xca,0), O5=0. If outputb(0xca,1), O5= +5V.
0xcb	D0	/W08	Write D0 to digital output O4=H12 pin7. Power on or reset O4 = 0. If outputb(0xcb,0), O4=0. If outputb(0xcb,1), O4= +5V.
0xcc	D0	/W08	Write D0 to digital output O3=H12 pin9. Power on or reset O3 = 0. If outputb(0xcc,0), O3=0. If outputb(0xcc,1), O3= +5V.
0xcd	D0	/W08	Write D0 to digital output O2=H12 pin11. Power on or reset O2 = 0. If outputb(0xcd,0), O2=0. If outputb(0xcd,1), O2= +5V.
0xce	D0	/W08	Write D0 to digital output O1=H12 pin13. Power on or reset O1 = 0. If outputb(0xce,0), O1=0. If outputb(0xce,1), O1= +5V.
0xcf	D0	/W08	Write D0 to digital output O0=H12 pin15. Power on or reset O0 = 0. If outputb(0xcf,0), O0=0. If outputb(0xcf,1), O0= +5V.
0xd0	D0	/W10	Write D0 to U2 pin 4 P0. P0= U24 pin17 ADC serial data in DIN. If outputb(0xd0,0), P0=0. If outputb(0xd0,1), P0= 1. Power on or reset low.
0xd1	D0	/W10	Write D0 to P1 and HV1. Power on or reset HV1 open. If outputb(0xd1,0), P1=0, HV1=high. If outputb(0xd1,1), P1= 1 HV1=low.
0xd2	D0	/W10	Write D0 to P2 and HV2. Power on or reset HV2 open. If outputb(0xd2,0), P2=0, HV2=high. If outputb(0xd2,1), P2= 1 HV2=low.
0xd3	D0	/W10	Write D0 to P3 and HV3. Power on or reset HV3 high. If outputb(0xd3,0), P3=0, HV3=high. If outputb(0xd3,1), P3= 1 HV3=low.
0xd4	D0	/W10	Write D0 to P4 and HV4. Power on or reset HV4 high. If outputb(0xd4,0), P4=0, HV4=high. If outputb(0xd4,1), P4= 1 HV4=low.
0xd5	D0	/W10	Write D0 to P5 and HV5. Power on or reset HV5 high. If outputb(0xd5,0), P5=0, HV5=high. If outputb(0xd5,1), P5= 1 HV5=low.
0xd6	D0	/W10	Write D0 to P6 and HV6. Power on or reset HV6 high. If outputb(0xd6,0), P6=0, HV6=high. If outputb(0xd6,1), P6= 1 HV6=low.

0xd7	D0	/W10	Write D0 to P7 and HV7. Power on or reset HV7 high. If <code>outportb(0xd7,0)</code> , P7=0, HV7=high. If <code>outportb(0xd7,1)</code> , P7= 1 HV7=low.
0xd8	D0	LED	Write D0 to LED (red LED) control line. If D0=0, red LED on. If D0=1, red LED off. When powered on or reset, the red LED is turned on.
0xd9	D0	LED1	Write D0 to LED1 (green LED) control line. If D0=0, green LED on. If D0=1, green LED off. When powered on or reset, the green LED is turned on.
0xda	D0	G0	Write D0 to G0 pin of the 8254 or 71054 (U1), a PTC (Programmable Timer/Counter). If D0 = 0, G0 is low. If D0 = 1, G0 is high. When powered on or reset, G0 is low.
0xdb	D0	G1	Write D0 to G1 pin of the 8254 or 71054 (U1), a PTC (Programmable Timer/Counter). If D0 = 0, G1 is low. If D0 = 1, G1 is high. When powered on or reset, G1 is low.
0xdc	D0	G2	Write D0 to G2 pin of the 8254 or 71054 (U1), a PTC (Programmable Timer/Counter). If D0 = 0, G2 is low. If D0 = 1, G2 is high. When powered on or reset, G2 is low.
0xdd	D0	DTR1	If D0=0, DTR1 is low. If D0=1, DTR1 is high. DTR1 is on J10 pin 16. Power on or reset low.
0xde	D0	B0	10-bit DAC, if D0=0, B0 is low. If D0=1, B0 is high. Power on or reset low.
0xdf	D0	B1	10-bit DAC, if D0=0, B1 is low. If D0=1, B1 is high. Power on or reset low.
0xe0	D0	/W20	Write D0 to HI1. Keypad <i>H1</i> high, if <code>outportb(0xe0,0)</code> . Keypad <i>H1</i> low, if <code>outportb(0xe0,1)</code>
0xe1	D0	/W20	Write D0 to HI2. Keypad <i>H2</i> high, if <code>outportb(0xe1,0)</code> . Keypad <i>H2</i> low, if <code>outportb(0xe1,1)</code>
0xe2	D0	/W20	Write D0 to HI3. Keypad <i>H3</i> high, if <code>outportb(0xe2,0)</code> . Keypad <i>H3</i> low, if <code>outportb(0xe2,1)</code>
0xe3	D0	BPL	Write D0 to the Beeper control line. Toggle D0 will sound the beeper. If D0=0, both beeper's pin high, less power. Power on or reset low.
0xe4	D0	KH	Write D0 to KH. Push-button P1-4 /KH high, if <code>outportb(0xe4,0)</code> . Pushbut P1-4 /KH low, if <code>outportb(0xe4,1)</code> . Power on or reset low.
0xe5	D0	/ACLK	Write D0 to /ACLK which is the MAX186 /CS. If D0=0, /ACLK is low. If D0=1, /ACLK is high. Power on or reset low. Set it high to disable ADC !
0xe6	D0	SDAB	Write D0 to the external EEPROM data line SDA. If D0=0, U8 pin 12 is open collector output, it can be pulled high or low. If D0=1, U8 pin12 is low.  Power on or reset SDAB low, U8 pin 12 open.

0xe7	D0	SCL	Write D0 to the external EEPROM clock line SCL. If D0=0, SCL is low. If D0=1, SCL is high. Power on or reset low.
0x40-0x43	D0-D7	/PTC	Write D0-D7 to PTC 8254 or 71054 (U1) registers PTC0, PTC1, PTC2, or PTCC.
0x80	D0-D7	LCD1	Write D0-D7 to H1 and H5 with A0 low, IOWR low, and LCD1 active high 500 ns.
0x81	D0-D7	LCD1	Write D0-D7 to H1 and H5 with A0 high, IOWR low, and LCD1 active high 500 ns.
0xa0	D0-D7	LCD2	Write D0-D7 to H1 and H5 with A0 low, IOWR low, and LCD2 active high 500 ns.
0xa1	D0-D7	LCD2	Write D0-D7 to H1 and H5 with A0 high, IOWR low, and LCD2 active high 500 ns.

Table 3.2 Reading External I/O Registers

Address	Data bits	Chip-Select Symbol	Function
0xe0	D0-D3	/IOR	Read D0-D3 from U20 SIN0, SIN1, SIN2, and SIN3.
0xe1	D0-D3	/IOR	Read D0-D3 from U20 IN4-IN6.
0x40-0x43	D0-D7	/PTC	Read D0-D7 from 8254 (U1) registers PTC0, PTC1, PTC2, or PTCC.
0x80	D0-D7	LCD1	Read D0-D7 from H1 and H5 with A0 low, IOWR low, and LCD1 active high 500 ns.
0x81	D0-D7	LCD1	Read D0-D7 from H1 and H5 with A0 high, IOWR low, and LCD1 active high 500 ns.
0xa0	D0-D7	LCD2	Read D0-D7 from H1 and H5 with A0 low, IOWR low, and LCD2 active high 500 ns.
0xa1	D0-D7	LCD2	Read D0-D7 from H1 and H5 with A0 high, IOWR low, and LCD2 active high 500 ns.

### 3.4 Serial Ports

The V25-Engine™ has three serial ports: two are V25 (V25 is a CPU used in the V25-Engine™) internal ports: SER0 and SER1, and the other is on the V25-Engine™ board—the SCC2691 UART (U8, Figs. 1.2, 2.2, and 2.3). By default, the SER0 and SER1 are configured as RS232 channels, and the SCC2691 can either be configured as an RS232 or an RS485 channel. The SensorWatch™ provides RS232 and/or RS485 drivers. When the SCC2691 is configured as an RS232 serial channel, there are some spare RS232C drivers on the SensorWatch™, so a fourth RS232C serial port may be configured with general I/O pins and software. The drivers on the SensorWatch™ are: U12 (1488 or 75C188, for SER0 and SER1 serial ports), U13 (1489, for SER0 and SER1 serial ports), U15 (75176, for the RS485 serial port) or U30-U31 (1489, 1488 for the RS232C ports). When the SCC2691 is configured as an RS485 channel, U15 must be installed. When it is configured as an RS232 channel, U30 and U31 must be installed. Never install U15 and U30-U31 at same time, because physically the outputs of the SCC2691 are directly connected to both U15 and U30-U31 (Fig. 3.1). SER0 is routed to J7 (RJ11 phone jack) or J14 (10 pin header). SER1 is routed to J8 (RJ11 phone jack) or J10 (10 pin header). The SCC2691 serial channel is routed to N1 and N2 (RJ11). A functional diagram of all the serial ports is shown in Fig. 3.1. The components with "Δ" are not installed at the same time.

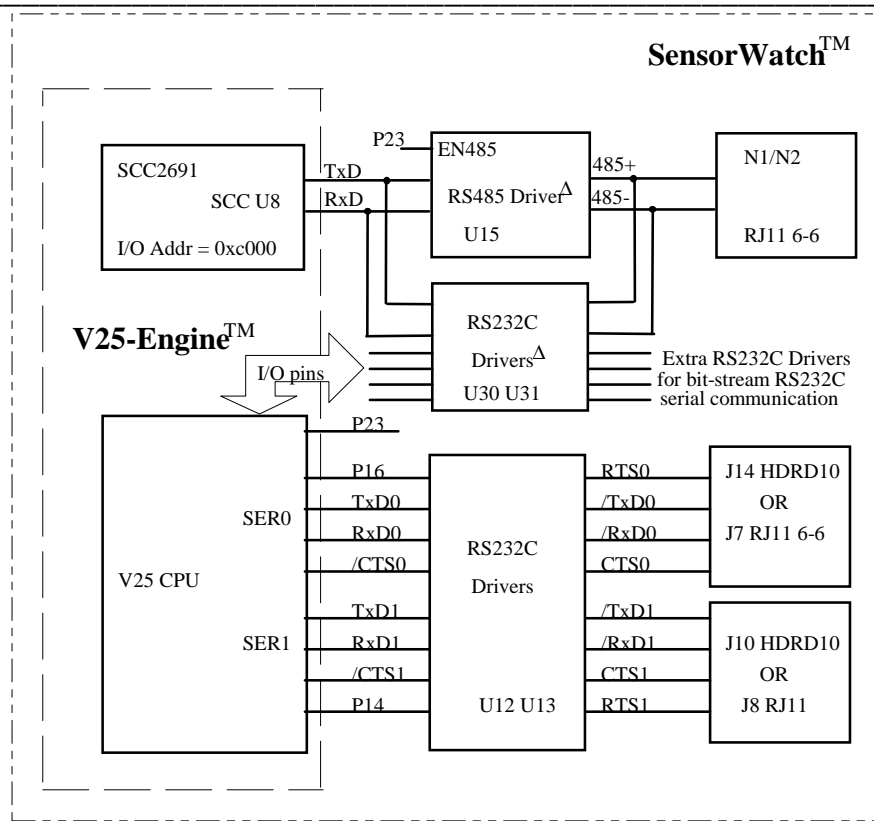


Fig. 3.1. Functional diagram of serial ports in the SensorWatch™, the components with "Δ" are mutually exclusive, not installed at the same time.

### 3.5 Digital Inputs and Outputs

There are eight digital output pins (O0 to O7) on the header H12 (Figs. 1.2 and 1.3). They can be set or reset at the I/O address 0xc8 (for O7) through 0xcf (for O0) (see Table 3.1). The digital output can source or sink 10-mA current for each pin.

Seven digital inputs are available at the terminal block T2, pins 19-13 (SIN0-2, IN3-6, see Figs. 1.2 and 1.3). These pins can be read at I/O address 0xe0 and address 0xe1 (see Table 3.2). Three digital inputs SIN0-2 are buffered by Schmitt-trigger inverters. The digital inputs IN3-IN6 have no Schmitt-trigger buffers. You may use `sw_di (i)` (see Chapter 4) to read the digital input channels ( $i = 0$  to 6).

If the keypad is not used, four open-collector digital outputs (*H1-H3*, /KH) and five general purpose I/O pins (P11, P24-27) are available on the keypad header K1 and header H3.

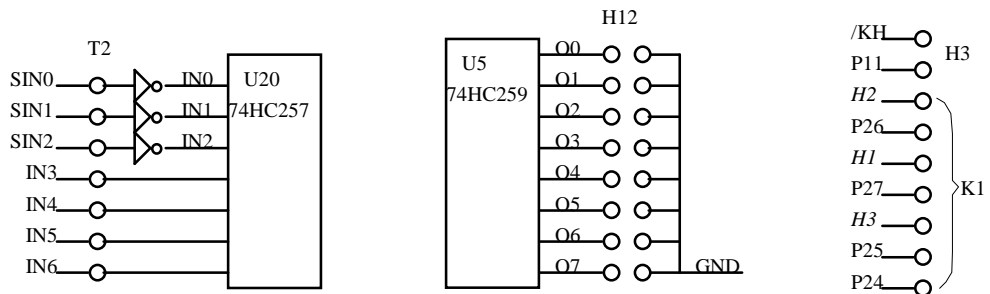


Fig. 3.2 Digital inputs and outputs

### 3.6 12-Bit ADC

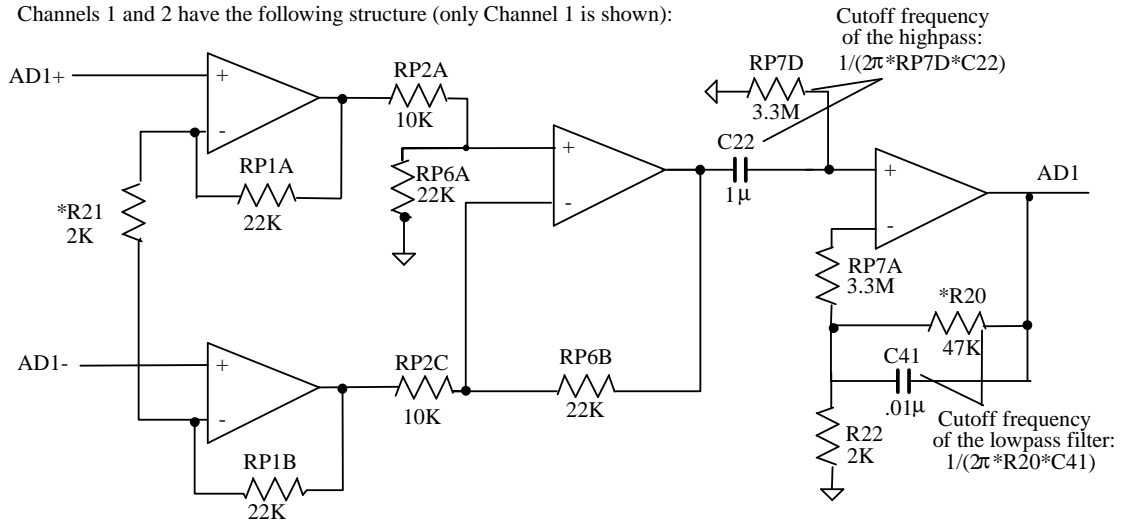
The ADC channels consist of the analog signal conditioning circuits and a 12-bit ADC, which may be a MAX186, LTC1294, or TLC2543. The type of ADC chip installed depends on the availability of the parts.

**3.6.1 Analog Signal Conditioning Circuits**

The analog signal conditioning circuits (Fig. 3.3) of the ADC channels, consisting of operational amplifiers (op-amps), filters, and configurable gain resistors, preprocess the analog input signals and allow the SensorWatch™ to interface with various sensors.

In channels 1 and 2 of the ADC channels, three-stage amplifiers are used. These amplifiers have differential inputs (in the first stage), high input impedance (upto  $10^{11}$ - $10^{12}\Omega$ ), high gain (upto 1000, see Fig. 3.3 for the gain of each stage), and bandpass filters (in the last stage, see Fig. 3.3). It is suitable to interface some sensors such as ECG electrodes, thermocouples, or strain gauges. A default gain of 1189 is provided for these two channels, but configurable by changing the resistors with "\*". In the rest of the channels a single differential op-amp is used to condition the analog signals. The default gain of these channels is set to 10, but configurable by changing the resistors with "\*".

Channels 1 and 2 have the following structure (only Channel 1 is shown):



The gain of the first stage:  
 $1 + 2 * RP1A / R21$

The gain of the second stage:  
 $RP6B / RP2C$

The gain of the third stage:  
 $R20 / R22$

Channels 3 through 8 have the following structure (only Channel 3 is shown):

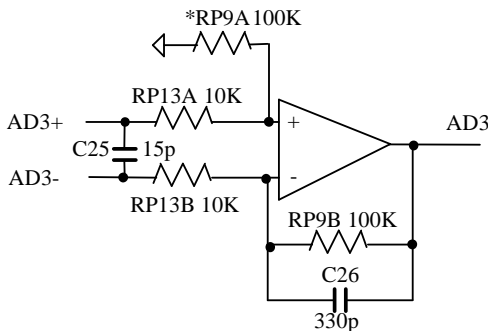


Fig. 3.3 Analog signal conditioning circuits: 3-stage amplifiers for Channels 1 and 2 (Channel 2 has the same structure as that of Channel 1, not shown here); and a single differential amplifier for Channels 3 through 8 (Channels 4 through 8 have the same structures as that of Channel 3, not shown here).

**3.6.2 The 12-bit ADC Chips**

Either a MAX186 or a LTC1294 can be used as the 12-bit ADC, depending on the availability of the parts. The ADC is installed with proper configuration in the factory. The MAX186 is a serial, 8-channel,

12-bit ADC which has an internal track/hold, internal reference voltage, and upto 133 KHz sample rate. It also has an internal reference voltage of 4.096V. When in unipolar mode, the range of the valid analog input voltage is from 0 V to +4.096 V. When in bipolar mode, the range is from -2.048 V to +2.048V. The LTC1294 is a serial, 8-channel, 12-bit ADC which has an internal sample and hold, but no internal reference. An external reference (through pin 14 of the U24) must be used with the LTC1294. The external reference may be provided by installing a LM285-2.5 for absolute reference applications, or by using VCC for ratio metric applications. For systems configured with MAX186, use `sw_ad12(ch, mode)`. For LTC1294, use `sw_ad12a(ch, mode)`, For LTC2543, use `sw_ad12b(ch,mode)`.

### 3.7 10-bit DAC

A 10-bit high-speed DAC is composed of a 8-bit DAC (DAC08, U13) and an adder shown in Fig. 3.4. The two least significant bits of the 10-bit data control B0 and B1 of the adder. By proper selection of the values of the resistors in Fig. 3.4, a 10-bit DAC can be constructed. The output DA of the DAC can be in the form of either voltage or current. J15 selects the voltage or current output. With J15 pin 1 and pin 2 connected, the DAC outputs a current (0-20 mA) at the terminal block T1 pin 20, corresponding to the digital input (0-1023). With J15 pin 2 and pin 3 connected, a voltage (0-5 V) is output at T1 pin 20 corresponding to the digital input (0-1023). A software driver `sw_da10(dat)` is available in SW.LIB.

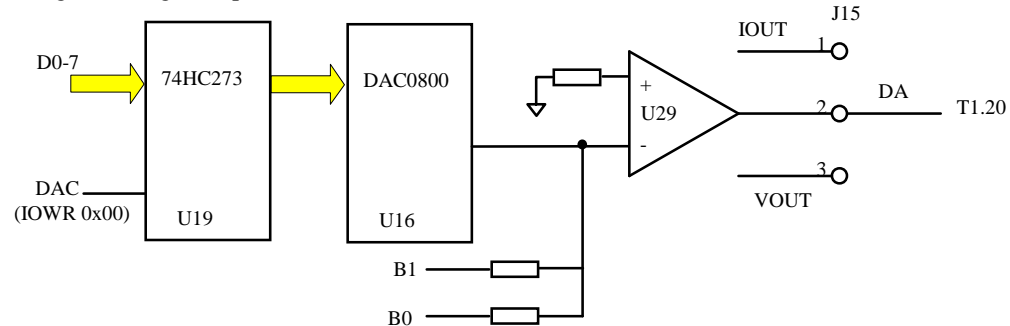


Fig. 3.4 The 10-bit DAC composed of a 8-bit DAC and additional circuitry

### 3.8 PORTT Comparator Inputs and 10-bit ADC

There are eight channels of voltage comparator (PORTT) inputs on the V25-CPU. The Channel 7, PT7, is used by the V25-Engine™ to monitor the PFI input voltage. The rest of the channels, PT0-6, are available on header H14 (Figs. 1.2 and 1.3) of the SensorWatch™.

If the optional 12-bit ADC is not installed, a very low cost 10-bit ADC (7 channels) can be constituted by using the comparator PORTT and the 10-bit DAC mentioned in Section 3.7: The analog input signals are connected to the signal conditioning circuits discussed in Section 3.6.1 and the outputs of the conditioning circuits connected to PT0-6 of the comparator PORTT. The PORTT has a threshold voltage, VTH. In this case, VTH is connected to the DAC output (DA pin) via pins 15-16 of H14. When DA voltage is stepping up by software, VTH is stepping up too. The comparator outputs 0 when the analog input signal is lower than VTH, 1 when the analog input signal is higher than VTH. To measure an analog input signal voltage, it is required to read and save each of the outputs of the comparator PORTT while VTH is stepping up. This 10-bit A/D conversion can be performed by calling a function `sw_ad10()` from SW.LIB, TERN. A block diagram of one of seven channels of the comparator and 10-bit ADC is shown in Fig. 3.5.

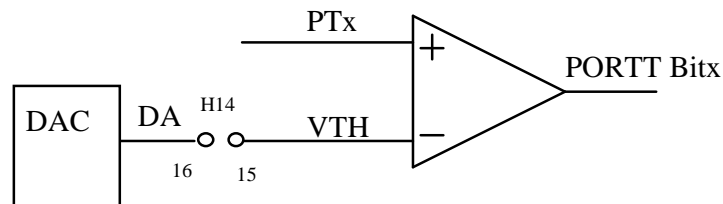


Fig. 3.5. A 10-bit ADC made of the 8-bit DAC and PORTT comparator, where x = 0 through 6

### 3.9 Programmable Timer/Counter (P8254-2 or $\mu$ PD71054)

A 16-bit programmable timer/counter (PTC), either 8254 or  $\mu$ PD71054 (fully compatible), is included in the SensorWatch™. The PTC has three independent 16-bit counters, six programmable count modes for each counter, gate controls, Binary or BCD count, Multiple latch command, clock rate upto 10 MHz. The six programmable counter modes include: interrupt on end of count mode, GATE retriggeable one-shot mode, rate generator mode, square wave generator mode, software-triggered strobe mode, and hardware-triggered strobe mode. The basic operation procedure is shown in Fig. 3.6. For more details, refer to Appendix A.

For each of the 16-bit counters, there are CLK, GATE, and OUT signals. The CLK signals, CK10-2, are available at terminal block T2 pins 20-22 (Figs. 1.2 and 1.3). The GATE signals, G0-2, are controllable by writing 0/1 to I/O address 0x1A-0x1C. The OUT signals, PO0-2, are available at header H0 pins 6, 4, and 2 (Figs. 1.2 and 1.3).

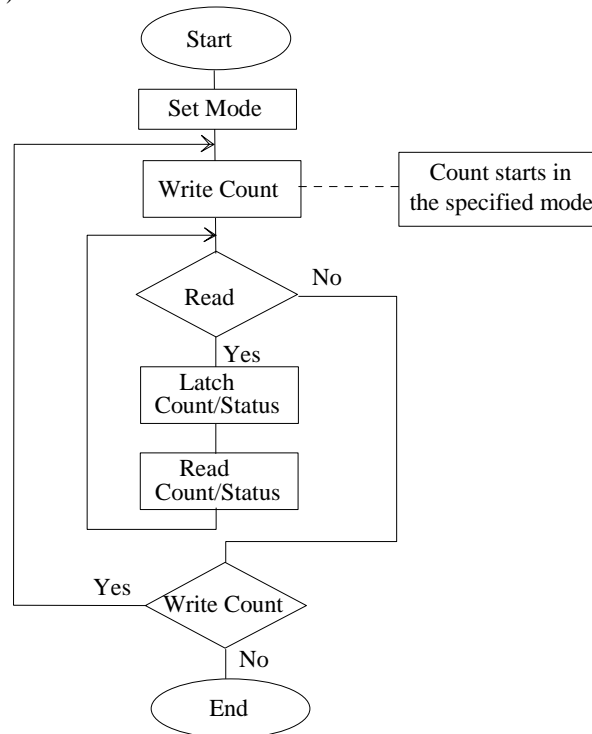


Fig. 3.6. Basic operation procedure of the PTC

### 3.10 High Voltage/Current Driver

A high voltage/current Darlington transistor array (ULN2003, U22) is used to support digital outputs P0-7 to handle 50-100V, 300 mA each pin. The output pins of the high voltage driver, HV6-HV1, are available at terminal blocks T2 pin 7 to pin 12, and HV7 is used in the power relay (Section 3.12). The K pin of the high voltage driver must be connected to the highest voltage level in the system in order to protect from voltage surge for inductive load. The GND pin at the terminal block T2 pin 4 is designed for heavy current to sub-strain out of the board. The user must provide a heavy wire (AWG18) connecting T2 pin GND directly to the high voltage power supply negative terminal.

### 3.11 PDC (Portable Data Carriers) Interface

An RJ11 connector is included on the SensorWatch™ to interface with PDCs (Portable Data Carriers, or serial EEPROMs, up to 8K bytes each).

### 3.12 Power Relay



A mechanical relay is installed on the SensorWatch™. The normally-close pin, normally-open pin and common pin of the power relay are connected to the terminal block T2 (pin 1, pin 2, and pin 3) with heavy trace, supporting at least 5A current. The relay is controlled with `sw_relay(k)`, where `k=0`, relay is off; `k=1`, relay is on.

### 3.13 Interrupt Driven Keypad

A telephone keypad (3 x 4) is provided for the user to input commands. Optional 4 push-buttons P1-P4 can be installed if more keys are needed. The keypad matrix is shown in Fig. 3.7. The row lines, connected to port 2 of V25-CPU (ROW1 = P26, ROW2 = P24, ROW3 = P25, and ROW4 = P27), are pulled up by 10-K resistors, therefore, these lines are read high after the relevant port lines are set to input-port mode. The function `sw_kb_scan()` sets these row lines to input-port mode; sets all the column lines H1, H2, H3, and /KH (open collector output pins) to high; and then scan the column lines by setting each line to low one-by-one and reading all the row lines to determine if there has been a key pressed and which key is pressed. If INTP2 (P13), a V25 external interrupt, is enabled, any key being pressed will generate an interrupt. The interrupt handler is `intp2_interrupt_isr`.

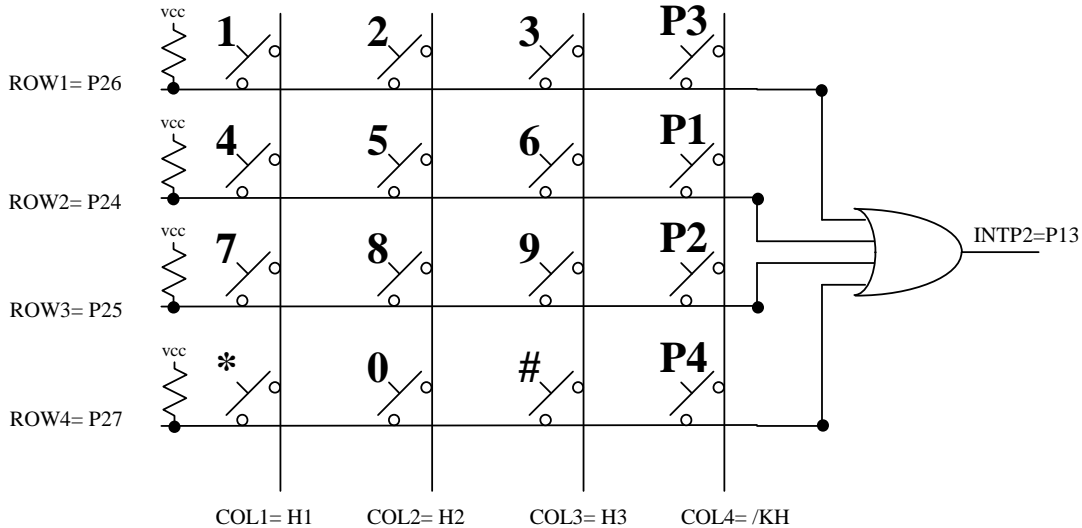


Fig. 3.7. Keypad matrix

### 3.14 RS485 Networking

SCC2691 is a single-chip UART (Universal Asynchronous Receiver/Transmitter, Philips Semiconductors) on the V25-Engine™. When SCC2691 is configured as an RS485 port, RS485 networking is made possible. SCC2691 is mapped in 0xc000 of the I/O address. The V25-Engine™ uses a special mode of the SCC2691 for networking, which provides automatic wake-up of the receiver through address frame recognition for multi-processor communications. Only the master V25-Engine™ sends out destination address bytes. All slaves, whose receivers are normally disabled, examine the received data stream and "wake-up" the CPU only upon receipt of an address character. The CPU compares the received address to its node address and enables the receiver if they match. The node address is stored in EEPROM byte 0 and the node #0 is the master.

N1 and N2 are phone jack connectors (RJ11, 6x6) used for the networking. N1 is used as the network chain-in, and N2 the chain-out. Not only signals 485+ and 485- are included in the RS485 network cable, but also the signals +12V, GND, and /RST are included.

### 3.15 LEDs, Beeper, Reset Push-Button

The SensorWatch™ has two LEDs which can be turned on/off by calling `sw_led(led, i)` function by specifying the adequate values for "led" and "i". The beeper in the SensorWatch™ can be controlled by

**sw\_bEEP (t, delay)** function. A reset push-button(Figs. 1.2 and 1.3) allows the user to reset the program on the SensorWatch™.

### 3.16 DC Power Supply

By default, a linear regulator LM7805 is installed to provide VCC (+5V), an LM723 precision regulator is used to provide +7V, a negative voltage converter 7660 is used to convert +12V voltage to -12V for RS232C communication, and an LM7905 regulator provides -5V for op-amps and the ADC. However, a power-off mode can be constructed to save energy, if the optional LM2575 (U11) switching-mode regulator and TK11408 (U9) regulator are used. The user can program the timer RTC72421 to release the VOFF pin of the LM2575 to high after some tasks are finished, then the whole board is in power-off mode with only 3  $\mu$ A consumption. After certain time period, e.g., 1 minute, the RTC72421 will pull down the VOFF pin and turn on the power. The VOFF pin can also be pulled low manually, in order to turn the board power on.

### 3.17 Jumpers and Headers

The jumpers and headers used in the SensorWatch™ are listed below:

J1, J2	20x2	V25-Engine™
J6	6x2	Header for connecting ADx- to GND
J7	RJ11 6-6	SER0, RS232
J8	RJ11 6-6	SER1, RS232
J9	RJ11 6-6	PDCs, serial EEPROMs
J10	5x2	SER1, RS232
J11	2x1	External main battery
J12	2x1	+12V to +V
J13	DJ005	Center negative DC power jack( $\phi$ 2.5 mm)
J14	5x2	SER0, RS232
J15	3x1	DAC output voltage(0-5V),if J15 2=3 DAC output current(0-20mA), if J15 1=2
H0	6x1	PTC outputs and P12, P15, IN7
H1	7x2	LCD1 for M1632 16x2 CH
H2	6x1	ADC MAX186/LTC1294 configuration
H3	2x1	header for extra keys /KH, P11
H4	2x1	High voltage driver K and +12VI
H6	2x1	VOFF to GND
H11	2x1	LCD2 for M4024
H12	8x2	digital output O0-7
H13	2x2	1-2 network power input +VI=+12VI 3-4 network remote reset /RT=/RST
H14	8x2	PORTT PT0-6, VTH, DA, AD1-7
H15	10x2	Header for graphics LCD TLX711/UG-204-03
H20	2x1	ADC MAX186 jumper on, LTC1294 jumper off
N1	RJ11 6-6	RS485 Network Chain IN
N2	RJ11 6-6	RS485 Network Chain OUT
L1	14x1	Header for 20x4 LCD L2014
T1	22x1	Terminal blocks
T2	22x1	Terminal blocks
K1	7x1	Keypads

---

# Chapter 4

## Software

### 4.1 Programming the SensorWatch™ with C/C++

The SensorWatch™ is programmable with Borland, Microsoft, or Turbo C/C++. (refer to Section 1.1, Chapters 4 and 6 of the V25-Engine™ Technical Manual, or MANUAL).

### 4.2 Software Libraries

The SensorWatch™ needs the VE.LIB and the SW.LIB. You need to check the value of "VE" parameter in the MAKEFILE to make sure it is "3", so that the SensorWatch™ system is selected. To use the V25-Engine™ functions, users need to include the adequate header files (refer to Chapter 6 of MANUAL).

### 4.3 Functions in SW.LIB

SW.LIB includes functions related to SensorWatch™ (V25-Engine™ functions are included in VE.LIB, see Chapter 6 of MANUAL).

#### 1. Keypad Functions

int **sw\_kb\_scan**(void);                sw.h                Scan keypads to identify which key is pressed  
return 0 - no key pressed  
return 1-16 if key is detected,  
1 for top-left key, 16 for lower-right key  
For VE-SW version with TKA2200 4x3 Telephone Keypad LZR 301-921-9440

#### 2. PDC (Portable Data Carrier) Functions

int **sw\_ee\_wr**                sw.h                Random write a character i to the addr of an external serial EEPROM  
(int addr, unsigned char i);

int **sw\_ee\_rd**(int addr);    sw.h                Random read a character from the addr of an external serial EEPROM.

#### 3. LED Function

void **sw\_led**                sw.h                LED is on, if led = 0, i = 1; LED is off, if led=0, i=0;  
(unsigned char led,                LED1 is on, if led = 1,i = 1; LED1 is off, if led=1, i=0;  
unsigned char i);

#### 4. Beeper Function

void **sw\_beeper**            sw.h                Sound the beeper with time duration = t,  
(int t, int f);                      and relative frequency controlled by "f".

#### 5. Relay Function

void **sw\_relay**              sw.h                Turn relay on, if k = 1; off, if k = 0.  
(unsigned char k);

#### 6. 10-Bit DAC Function

void **sw\_da10** (int dat);    sw.h                Output dat (0-1023) to 10-bit DAC (0-5 V).

#### 7. 10-Bit ADC Function

void **sw\_ad10**(int\* ad);    sw.h                Return results of the 10-bit ADC in the integer array pointed by "ad".

#### 8. 12-Bit ADC Functions

unsigned char **adch**(unsigned char ch);    sw.h                Convert adc channel number to a control format

---

		Input : ch = 1 to 8, AD channel number
	Returns:	0, if ch=1 4, if ch=2 1, if ch=3 5, if ch=4 2, if ch=5 6, if ch=6 3, if ch=7 7, if ch=8
<b>int sw_ad12</b> (unsigned char c, unsigned char mode);	sw.h	Read the MAX186 12-bit ADC conversion results unsigned char c = input channel c = 0, input ch = AD0 c = 1, input ch = AD4 c = 2, input ch = AD1 c = 3, input ch = AD5 c = 4, input ch = AD2 c = 5, input ch = AD6 c = 6, input ch = AD3 c = 7, input ch = AD7 unsigned char mode = 0, Poweroff Mode unsigned char mode = 1, Unipolar Mode, Input Range 0 to +4.096 V unsigned char mode = 2, Bipolar Mode, Input Range -2.048 to +2.048 V Return: 12 bit AD data Bipolar: int 0x0800 = -2.048V, 0x0fff=-0.001V, 0x000=0V, 0x07ff=+2.048V Unipolar: int 0x0000 = 0 V, 0x07ff = + 4.096V 1 wait state for Memory and I/O without RDY, < 400 us execution time
<b>int sw_ad12a</b> (unsigned char ch, unsigned char mode);	sw.h	Read the LTC1294 12-bit ADC conversion results unsigned char c = input channel c = 0, input ch = AD0 c = 1, input ch = AD4 c = 2, input ch = AD1 c = 3, input ch = AD5 c = 4, input ch = AD2 c = 5, input ch = AD6 c = 6, input ch = AD3 c = 7, input ch = AD7 unsigned char mode = 0, Poweroff Mode unsigned char mode = 1, Unipolar Mode, Input Range 0 to REF+ unsigned char mode = 2, Bipolar Mode, Input REF- to REF+ Return: 12 bit AD data Bipolar: int 0x0800 = REF-, 0x0fff=-0.001V, 0x000=0V, 0x07ff = REF+ Unipolar: int 0x0000 = 0 V, 0x07ff = REF+ 1 wait state for Memory and I/O without RDY, < 400 us execution time
<b>int sw_ad12b</b> (unsigned char c);		Analog to Digital conversion using TLC2543 Input: unsigned char c = input channel c = 0-a, c = b, input ch = (vref+ - vref-) /2 c = c, input ch = vref- c = d, input ch = vref+ c = e, software power down

---

 Unipolar:

(Vref+ - Vref-)=0x7ff

Vref- = 0x000

Vref+ = 0xff

Use 1 wait state for Memory and I/O without RDY, &lt; 350 us execution time

**9. High Voltage Function**

void **sw\_hv** sw.h Turn channel "hv" of the high voltage driver  
 (unsigned char hv, on if k = 1, off if k = 0.  
 unsigned char k);

**10. Digital Input/Output Functions**

int **sw\_di** sw.h Reads the digital channel i from T2 (Fig. 1.2)  
 (unsigned char i); for i = 0-6:  
 return 0 if the channel input is low, 1 if high  
 for i = other value: return -1

void **sw\_do** sw.h Write k (0 or 1) to digital  
 (unsigned char digout, output channel, "digout" = 0-7.  
 unsigned char k);

**11. PTC (Programmable Timer/Counter) Functions**

int **sw\_ptc\_mode** sw.h Set mode for PTC  
 (unsigned char chnrl, chnrl: 0, 1, or 2 for channel 0,1, or 2;  
 unsigned char rwmode, rwmode: 1:low-byte r/w,  
 unsigned char cntmode, 2: high-byte r/w,  
 unsigned char bcd); 3: low-high r/w;  
 cntmode: 0, 1, 2, 3, 4, or 5, for Mode 0-5;  
 bcd: 0:hex, 1:BCD; return 1 if successful, 0 otherwise.

int **sw\_ptc\_cltch** sw.h Count latch command  
 (unsigned char chnrl); chnrl: 0, 1, or 2, for Channel 0-2;  
 return 1 if successful, 0 otherwise.

int **sw\_ptc\_mlts** sw.h Multiple latch command  
 (unsigned char ch0, ch0: 1 Channel 0 latched, 0 not latched;  
 unsigned char ch1, ch1: 1 Channel 1 latched, 0 not latched;  
 unsigned char ch2, ch2: 1 Channel 2 latched, 0 not latched;  
 unsigned char ltch\_status, ltch\_status: 0 status latched, 1 not latched;  
 unsigned char ltch\_count); ltch\_count: 0 count latched, 1 not latched.

void **sw\_ptc\_wr** sw.h Write counter  
 (unsigned char chnrl, chnrl: 0, 1, or 2 for Channel 0, 1, or 2;  
 unsigned char wmd, wmd: write mode, 1: for low-byte write,  
 unsigned int dat); 2: for high-byte write, 3: for low-high write.

unsigned int **sw\_ptc\_rd** sw.h Read counter or status  
 (unsigned char chnrl, chnrl: 0, 1, or 2 for Channel 0, 1, or 2;  
 unsigned char rmd); rmd: read mode, 1 for low-byte/status read\*,  
 2 for high-byte read, 3 for low-high read.

**Note:**

\* The **status read** can be realized only after a multiple latch command is issued,  
 in which status read is enabled by setting ltch\_status = 0.

## A. Programmable Timer/Counter (PTC)

The  $\mu$ PD71054 is a high-performance, programmable counter for microcomputer system timing control. The description for the  $\mu$ PD71054 applies to 8254, a PTC fully compatible with the  $\mu$ PD71054. Three 16-bit counters, each with its own clock input, gate input, and OUT pin, can be clocked from dc to 10 MHz. Under software control, the  $\mu$ PD71054 can generate accurate time delays. Initializing the counter, the  $\mu$ PD71054 counts the delay and interrupts the CPU when the task is complete. This eliminates the need for software timing loops.

The three counters in the  $\mu$ PD71054 are capable of binary or BCD operations. There are six programmable count modes. The counters operate independently and each can be set to a different mode. Use address lines A1, A0 to select a counter and perform a read/write operation.

For each of the 16-bit counters, there are CLK, GATE, and OUT signals. The CLK signals, CKI0-2, are available at terminal block T2 pins 20-22 of the SensorWatch (Figs. 1.2 and 1.3). The GATE signals, G0-2, are controllable by writing 0/1 to I/O address 0x1A-0x1C (for hardware control of G0-2, the user needs to cut the connections between U1 pin 11 and U4 pin 6, between U1 pin 14 and U4 pin 7, and between U1 pin 16 and U4 pin 9, then U1 pins 11, 14, and 16 are available for external G0-2 inputs). The OUT signals, PO0-2, are available at the header H0 pins 6, 4, and 2 (Figs. 1.2 and 1.3).

### A.1 Mode 0: Interrupt on End of Count

In this mode, the OUT of the counter changes from low to high level when the end of the specified count is reached. Table A.1 summarizes the Mode 0 operation:

Table A.1. Mode 0 Operation

Function	Result
Initial OUT	Low level
GATE High	Count enable
GATE Low	Count disable
Count Write	The OUT pin goes low independent of the CLK pulse. In 2-byte mode, the count is disabled when the first byte is written. The OUT pin goes low when a new mode of new count is written.

Table A.1 (Continued)

Function	Result
----------	--------

Count Transfer * and Operation	When the count is written with GATE high: Transfer is performed at the first CLK pulse after the count value is written. The down counter is decremented beginning at the first CLK pulse after data transfer. If a count of n is set, the OUT pin goes high after n+1 CLK pulses. When the count is written with GATE low: Transfer is performed at the first CLK pulse after the count is written. The down counter is decremented beginning at the first CLK pulse after the GATE signal goes high. If a count of n is set, OUT is low for a period of n CLK pulses after GATE goes high.
Count Zero	The signal at the OUT pin goes high. The count operation does not stop and counts down to FFFFH (hexadecimal) or 9999 (BCD) and continues to count down.
Minimum Count	1

Note:

\* "Count Transfer" refers to the transfer from the count register to the down-counting counter (or "down counter"). The down counter is decremented at the falling edge of the CLK pulse.

### A.2 Mode 1: GATE Retriggerable One-Shot

In this mode, the  $\mu$ PD71054 functions as a retriggerable one-shot. A low-level pulse triggered by the GATE input is output from the OUT pin. Table A.2 summarizes Mode 1 operations.

**Table A.2. Mode 1 Operations**

Function	Result
Initial OUT	High level
GATE Trigger*	Count data is transferred at the CLK pulse after the trigger.
Count Write	The count is written without affecting the current operation.
Count Transfer and Operation	Transfer is performed at the first CLK pulse after the trigger. At the same time, the signal at the OUT pin goes low to start the one-shot pulse operation. The count is decremented beginning at the next CLK pulse. If a count of n is set, the one-shot output from the OUT pin continues for n CLK pulses.
Count Zero	The signal at the OUT pin becomes high. Count operation does not stop and wraps to FFFFH (hexadecimal) or 9999 (BCD) and continues to count.
Minimum Count	1

Note:

\* The trigger is ignored when the count has not been written after the mode is set, or when only one byte of the count has been written in 2-byte count mode.

### A.3 Mode 2: Rate Generator

In this mode, the signal from the OUT pin cyclically goes low for one clock period when the counter reaches 0001H. The counter operates as a frequency divider. Table A.3 lists the Mode 2 operations.

Table A.3. Mode 2 Operations

Function	Result
Initial OUT	High level
GATE High	Count enable
GATE Low	Count disabled. If GATE goes low when OUT is low, OUT will go high (independent of the CLK pulse).
GATE Trigger*	Transfer is performed at the first CLK pulse after the trigger.
Count write	Count is written without affecting the current operation.
Count Transfer and Operation	Transfer is performed at the CLK pulse after the count is written following the mode setting. The counter is then decremented. transfer is again performed at the first CLK pulse after the count becomes 1. When the trigger is used, transfer is performed at the next CLK pulse. When the contents of the down counter becomes 1, OUT goes low for one CLK pulse and returns to high. If a count of n is set, OUT repeats this sequence every n CLK pulses.
Count Zero	Never occurs in this case.
Minimum count	2

Note:

\* The trigger is ignored when the count has not been written or when only one byte of the count has been written in 2-byte mode.

#### A.4 Mode 3: Square Wave Generator

This mode is a frequency divider similar to mode 2, but with a different duty cycle. Table A.4 lists the Mode 3 operations.

Table A.4. Mode 3 Operations

Function	Result
Initial OUT	High level
GATE High	Count enable
GATE Low	Count disabled. If GATE goes low when OUT is low, OUT will go high (independent of the CLK pulse).
GATE Trigger*	Transfer is performed at the first CLK pulse after the trigger.
Count Write	Current operation is not affected. The count is transferred at the end of the half-period of the current square wave and the OUT pin goes high.

Table A. 4. (Continued)

Function	Result
----------	--------



Count Transfer and Operation	Count data is transferred at the first CLK pulse after the count write following the mode setting. Transfer is performed at the end of the current half-cycle and the OUT pin is inverted. Transfer is also performed at the CLK pulse after the trigger. The operation performed depends on whether count n is even or odd. When n is even, the count is decremented by two on each following clock pulse. At the end of the count of two, the count is again transferred and the OUT pin is inverted. This is taken as a half-cycle and repeated. When n is odd, n-1 is transferred and the count is decremented by two on each following clock pulse. The half-cycle when the OUT pin is high continues until the end of count 0 and n-1 is transferred again at the next CLK pulse. The half-cycle while OUT is low continues until the end of count 2. Thus, the half-cycle while OUT is high is one CLK longer than the half-cycle while OUT is low.
Count Zero	Occurs only when the count is odd.
Minimum Count	3

#### A. 5 Mode 4: Software-Triggered Strobe

In this mode, when the specified count is reached, OUT goes low for one CLK pulse. Table A.5 summarizes the Mode 4 operations.

Table A.5. Mode 4 Operations

Function	Result
Initial OUT	High level
GATE High	Count enable
GATE Low	Count disable
Count Write	Count is transferred at the next CLK pulse when the count is written. In 2-byte mode, data is transferred after the second byte is written.
Count Transfer and Operation	Count is transferred at the first CLK following the count write. If GATE is high, the down counter begins to decrement from the next CLK. If GATE is low, decrement begins at the first CLK after GATE goes high.
Count zero	OUT is low for one CLK pulse and returns to high. The down counter wraps to FFFFH (hexadecimal) or 9999 (BCD) without stopping counter operation.
Minimum Count	1

#### A.6 Mode 5: Hardware-Triggered Strobe [Retriggerable]

This mode is similar to Mode 4 except that operation is triggered by the GATE input and can be retriggered. Table A.6 lists Mode 5 operations.

Table A.6. Mode 5 Operations

Function	Result
Initial OUT	High level
GATE Trigger*	The count is transferred at the CLK pulse after the trigger. The GATE has no effect on the OUT signal.
Count Write	The count is written without affecting the current operation.

Count Transfer and Operation	Count is transferred at the first CLK pulse after a trigger, providing that the mode and count have been written. Decrement begins from the first CLK pulse after a data transfer. If a count of n is set, OUT goes low for n+1 CLK pulses after the trigger.
Count zero	OUT is low for one CLK and goes high again. The down counter counts to FFFFH (hexadecimal) or 9999 (BCD) without stopping the counter operation.
Minimum count	1

### A. 7 Control Register Format

A control command must be written into the control register (when A1 = 1 and A0 = 1) to set the counter mode before operating the counter. Table A.7.1-A.7.3 list the format of the three 8-bit control commands.

Table A. 7.1 Control Register Format

Bit (Name)	Meaning			
B0 (BCD)	0: binary count (16 bits); 1:BCD count (4 digits)			
B1 (CM0)	CM2	CM1	CM0	Count Mode
B2 (CM1)	0	0	0	Mode 0
B3 (CM2)	0	0	1	Mode 1
	x	1	0	Mode 2
	x	1	1	Mode 3
	1	0	0	Mode 4
	1	0	1	Mode 5
B4 (RWM0)	RWM1	RWM0		Read/Write Mode
B5 (RWM1)	0	1		low byte only
	1	0		high byte only
	1	1		low byte and then high byte
B6 (SC0)	SC1	SC0		Select Count
B7 (SC1)	0	0		Count 0
	0	1		Count 1
	1	0		Count 2

The Count Latch Command is used to latch a counter value for reading without affecting the counter operation. This value is latched until it is read or a new mode is set. If the counter value that was latched into the count latch is not read before a second count latch command is executed, the second latch command is ignored, because the latch is not released until a read or a mode-set command is issued.

Table A. 7.2. Control Register Format for Count Latch Command

Bit (Name)	Meaning
B0	0
B1	0
B2	0
B3	0

B4 (RWM0) B5 (RWM1)	RWM1 RWM0 0 0	Indication of Count Latch Command
B6 (SC0) B7 (SC1)	SC1 SC0 0 0 0 1 1 0	Target Latched Counter 0 Counter 1 Counter 2

The Multiple Latch Command is used to latch count and status value of any counter for reading without affecting the counter operations. When both count and status value of a counter are latched, the status value is always read first, and the count value is read by the next read operation. If the data that was latched is not read before a second multiple latch command is executed, the second latch command is ignored, because the latch is not released until a read or a mode-set command is issued.

Table A. 7.3. Control Register Format for Multiple Latch Command

Bit (Name)	Meaning
B0	
B1 (CNT0)	1: Counter 0 selected, 0: not selected
B2 (CNT1)	1: Counter 1 selected, 0: not selected
B3 (CNT2)	1: Counter 2 selected, 0: not selected
B4 (STATUS)	0: Status latched, 1: not latched
B5 (COUNT)	0: Count latched, 1: not latched
B6 (SC0) B7 (SC1)	SC1 SC0 1 1 Indication of Multiple Latch Command

## Appendix B. 12-bit ADC (TI, TLC2543) with 11 inputs, 3 Self-test-mode, and Software Power-down

The TLC2543 from TI is a 12-bit analog to digital converter with serial interface and 11 analog inputs. This device has an on-chip 14 channel multiplexer that can select any one of 11 inputs or any one of three internal self-test voltages. The sample-and hold function is automatic.

In order to provide 11 channels of analog inputs, modifications are made on the SensorWatch™ analog signal conditional circuits. The new definitions of the terminal block pin functions are listed below.

NAME	T1	Function with TLC2543
AD1+	○	AIN8+ G=(1+2*10K/R21) Differential Input
AD1-	○	AIN8-
GND	○	GND
AD2+	○	AIN9+ G=1+10K/10K=2 Single Ended Input 0 to REF/G
AD2-	○	AIN10+ G=1+10K/10K=2 Single Ended Input 0 to REF/G
GND	○	GND
AD3+	○	AIN2+ G=10K/10K=1 Single Ended Input 0 to REF/G
AD3-	○	AIN0+ G=1+10K/10K=2 Single Ended Input 0 to REF/G
AD4+	○	AIN3+ G=10K/10K=1 Single Ended Input 0 to REF/G
AD4-	○	AIN1+ G=1+10K/10K=2 Single Ended Input 0 to REF/G
AD5+	○	AIN4+ G=10K/10K=1 Differential or Single Ended Input
AD5-	○	AIN4-
AD6+	○	AIN5+ G=10K/10K=1 Differential or Single Ended Input
AD6-	○	AIN5-
AD7+	○	AIN6+ G=10K/10K=1 Differential or Single Ended Input
AD7-	○	AIN6-
AD8+	○	AIN7+ G=10K/10K=1 Differential or Single Ended Input
AD8-	○	AIN7-
GND	○	GND
DA	○	DA
-5V	○	-5V
VCC	○	VCC

The VCC is applied to the REF+, and the REF- is connected to GND.

A modified schematics SW01.SCH is for the SensorWatch™ with TLC2543

The software driver for using the TLC2543 ADC is

```
int sw_ad12b(unsigned char ch);
```

ADC with TLC2543 on the SensorWatch™  
Input:

---

unsigned char ch = input channel  
c = 0, input ch = A3-  
c = 1, input ch = A4-  
c = 2, input ch = A3+  
c = 3, input ch = A4+  
c = 4, input ch = A5+  
c = 5, input ch = A6+  
c = 6, input ch = A7+  
c = 7, input ch = A8+  
c = 8, input ch = A1+ and A1-  
c = 9, input ch = A2+  
c = a, input ch = A2-  
c = b, input ch = (vref+ - vref-) /2  
c = c, input ch = vref-  
c = d, input ch = vref+  
c = e, software power down

Output: 12 bit AD data :

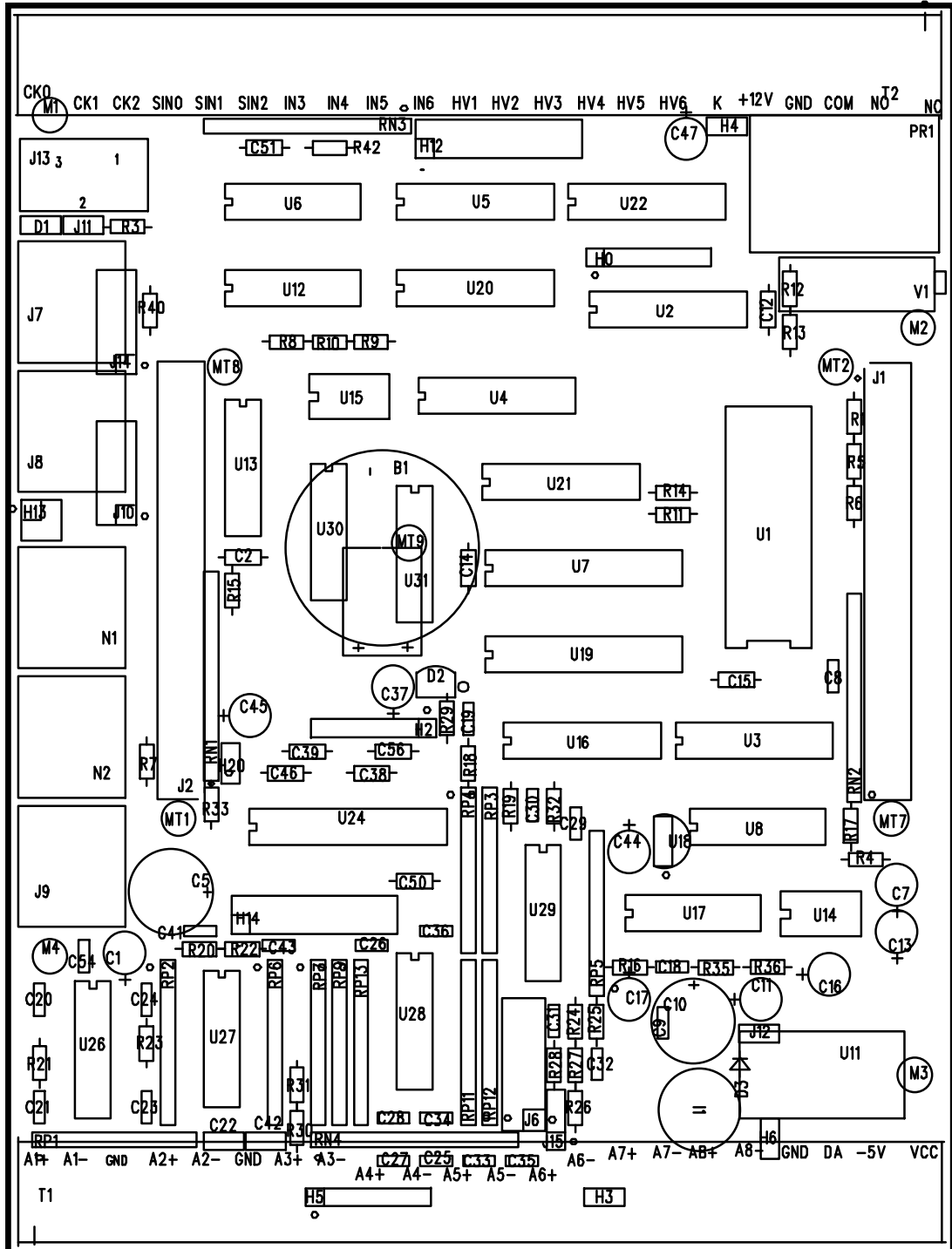
$(V_{ref+} - V_{ref-})/2 = 0x7ff$   
 $V_{ref-} = 0x000$   
 $V_{ref+} = 0xff$

Use 1 wait state for Memory and I/O without RDY, < 350 us execution time

Use 0 wait state for Memory and I/O with VEP010, < 300 us execution time

Appendix C: SensorWatch™ Layout

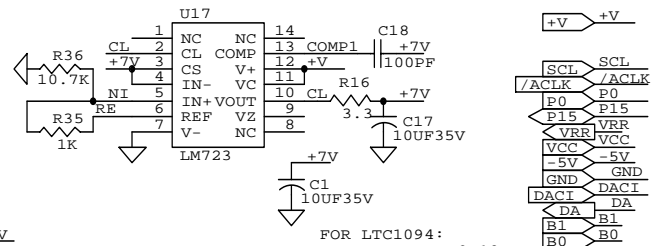
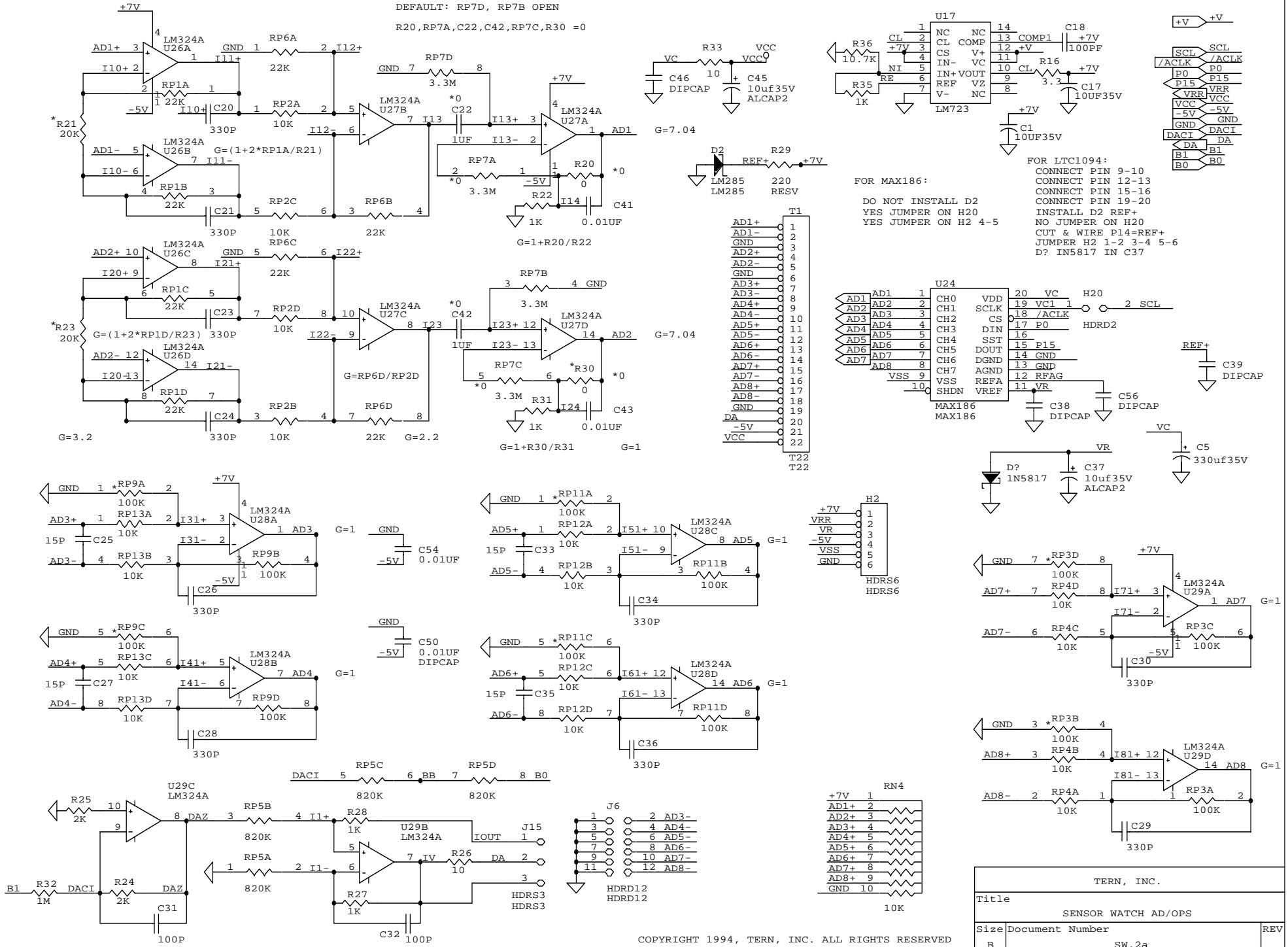
4.30, 5.70



0.00, 0.00



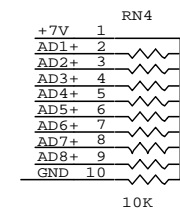
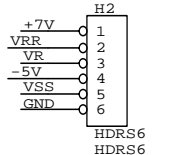
DEFAULT: RP7D, RP7B OPEN  
R20, RP7A, C22, C42, RP7C, R30 = 0



FOR MAX186:

DO NOT INSTALL D2  
YES JUMPER ON H20  
YES JUMPER ON H2 4-5

FOR LTC1094:  
CONNECT PIN 9-10  
CONNECT PIN 12-13  
CONNECT PIN 15-16  
CONNECT PIN 19-20  
INSTALL D2 REF+  
NO JUMPER ON H20  
CUT & WIRE P14=REF+  
JUMPER H2 1-2 3-4 5-6  
D? IN5817 IN C37



TERN, INC.		
Title		
SENSOR WATCH AD/OPS		
Size	Document Number	REV
B	SW.2a	
Date:	November 20, 1994	Sheet 2 of 2