MicroPac 515C

HARDWARE REFERENCE MANUAL

for Revision 0 boards

MANUAL Revision 1.1

EMAC, inc.
EQUIPMENT MONITOR AND CONTROL
2390 EMAC Way, Carbondale, Illinois 62901
World Wide Web: http://www.emacinc.com
Phone: (618) 529-4525 Fax: (618) 457-0110

DISCLAIMER

EMAC has made every attempt to ensure that the information in this document is accurate and complete. However, EMAC assumes no liability for any damages that result from use of this manual or the equipment that it documents. EMAC reserves the right to make changes at any time.

FCC COMPLIANCE

EMAC's MicroPac series of computers and accessories are classified as sub-assemblies per FCC CST bulletin, No. 62, May 1984. The bulletin states that sub-assemblies are not themselves subject to the FCC rules. Only the end product is directly subject to the rules.

Table of Contents

INTRODUCTION

The Infineon C515C processor used in the MicroPac 515C has many features, with some requiring trading of one function for another. The MicroPac 515C features are as follows:

- CPU: Infineon C515C with 10 MHz crystal speed. Improved architecture results in a 600 nS instruction cycle time at 10 MHz (compared to 1000nS instructions cycles at 12MHz for most 8051 derivatives).

- CAN MODULE: It is a full CAN module version 2.0 B active with 15 message objects and basic-CAN feature with data rates up to 1 MBaud. The module has 256 register/data bytes allocated in external data memory area.

- DIGITAL I/O: 11 TTL level, bit programmable I/O lines, 8 high current outputs, with an additional 4 inputs and 3 output lines available with the purchase of the DUART upgrade option. There is a 50 pin I/O rack compatible header connector that allows connection to 24 of these I/O lines (consisting of the 8 high current outputs, 11 TTL level programmable lines and the optional 2 inputs and 3 outputs). The A/D input lines can also be used as digital inputs. When the coprocessor is not installed, an additional 6 TTL level, bit programmable I/O lines are available on a 10 pin header.

- ANALOG INPUTS: Fast 8 channel, 10 bit analog to digital converter with a 9.6 microsecond conversion rate for 104K samples per second. A/D inputs may be also be used for an additional 8 digital inputs.

- ANALOG OUTPUTS: The optional D/A has 4 channels with 8 bit resolution.

- COUNTER/TIMERS/PWM: Three, 16 bit counter/timers are included standard. One of these has four I/O lines which can be used as compare outputs, or capture inputs (for Pulse Width Modulation/Demodulation). A fourth 16 bit counter/timer is included with the dual serial port upgrade. A watchdog timer is also provided.

- COMMUNICATION: The 515C is capable of supporting up to 4 RS-232 serial ports, 1 is standard and 3 are included with the purchase of optional upgrades. Conversion of up to 3 ports from RS-232 to RS-422/485 is available optionally. Serial ports are capable of multiprocessor communication using 9 bit protocol and standard baud rates of up to 230.4K baud are available.

- COPROCESSOR: An optional coprocessor may be added which is preprogrammed to provide an additional RS-232 serial port with transmit and receive data buffering and a keypad interface (for decoding up to 16 keys in a 4x4 matrix). This coprocessor may be custom programmed by EMAC for almost unlimited capabilities. Resources available on the coprocessor: 128 bytes of RAM, a UART, two 16 bit timer/counters with counter inputs, 2 interrupt inputs, 12 I/O lines, and an analog comparator.

- LCD INTERFACE: The interface will drive all industry-standard character displays based on the HD44780 LCD controller and compatibles. It will also drive all industry-standard graphic displays based on the HD61830 LCD controller and compatibles.

- PERIPHERAL EXPANSION: The address bus, data bus, reset and interrupt lines are available on a 50 pin bus expansion header for connecting to external peripherals from EMAC, or to peripherals of your own design.

- MEMORY: In addition to the 255 bytes of IRAM and 2K of XRAM memory within the C515C, there is provision for up to 128K of EPROM or in-circuit programmable FLASH and up to 512K of RAM. Total memory capacity is 640K bytes. The C515C directly accesses 128K of memory (64K EPROM and 64K RAM) with the balance being bank selected memory. 32K RAM is included in the standard configuration. A parameter storage area is implemented with a 1K bit serial EEPROM. The EEPROM is a nonvolatile memory organized into 64, 16 bit words with over 10,000 write cycles guaranteed and unlimited read cycles.

POWER REQUIREMENTS

```
 ST1
1 o POWER
2 o GROUND
3 o RESET*
```

The HC16 can be powered by a 5 volt regulated supply or an 8 to 15 volt unregulated DC supply depending on the setting of JP2. Typical unloaded current draw is 85 mA. Power is connected to the 515C through ST1pin 1 and ground to pin 2. The board may be reset by momentarily bringing ST1 pin 3 to ground.

The two configurations of JP2 are shown below:

1 o-o o-o JP2 PWR Default position with regulator in-circuit. Apply 8 to 15 VDC to ST1 pin 1.	1 o o-o o JP2 PWR Position with bypassed regulator. Apply only 5V regulated DC to ST1 pin 1.

MEMORY DEVICES

In addition to the C515C microprocessor's 256 bytes of internal data RAM and 2K bytes of internal XRAM, it also externally accesses 64K of program memory (read only) and 64K of data memory (read/write). With the memory glue logic and option jumpers included on the MicroPac 515C, Flash and EPROM memory space has been expanded to 128K and RAM space to 512K.

RAM

The RAM socket U11 may be populated by 32K,128K or 512K RAMs. There is a jumper under the socket that must be set according to the memory device size: 512K or 32K/128K

EPROM/Flash

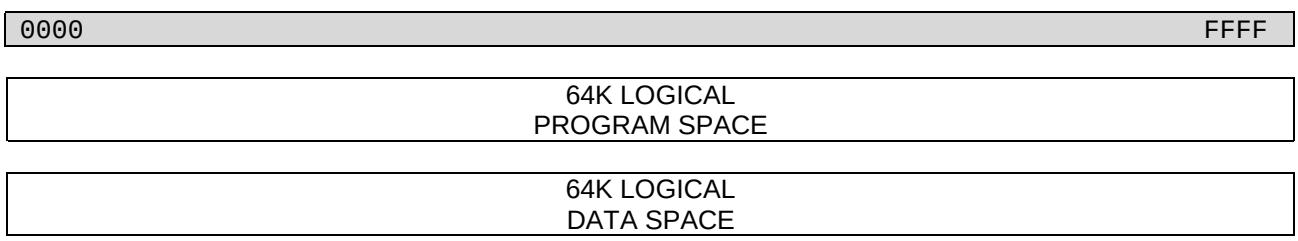
The EPROM/Flash socket U12 may be populated by the following devices:

Description	Part #	Jumper setting
32K EPROM	27C256	(JP 1 Must be in the program/data configuration)
64K EPROM	27C512	(JP 1 Must be in the 64K-128K EPROM/Flash configuration)
128K EPROM	27C010	(JP 1 same as above)
64K Flash	29C512	(JP 1 same as above)
128K Flash	29C010	(JP 1 same as above)
128K Flash	29F010	(JP 1 same as above)

JP5 allows you to write protect the flash. The two jumper positions are marked PR and WE for protect and write enable, respectively.

LOGICAL AND PHYSICAL MEMORY ADDRESSES

The C515C has two types of external memory: program and data. This is illustrated below:



The MicroPac 515C uses port lines to extend the memory map for both the program space and data space.

Therefore we must distinguish between the *logical address* that is part of conventional memory mapping and the *physical address* which is the actual memory location being accessed in a particular device.

The following table shows how port lines P7.0, P5.0 and P5.5 affect each of the 128K EPROM/FLASH/RAM and 512K RAM. The port lines do not affect memories that are 64K or less in size.

PHYSICAL MEMORY ADDRESSES AS AFFECTED BY PORT LINES					
P7.0	P5.0	P5.5	128K EPROM/FLASH (Only affected by P5.0)	128K RAM (Only affected by P5.5)	512K RAM (Affected by all)
0	0	0	00000-0FFFF	00000-0FFFF	00000-0FFFF
0	0	1	00000-0FFFF	10000-1FFFF	10000-1FFFF
0	1	0	10000-1FFFF	00000-0FFFF	20000-2FFFF
0	1	1	10000-1FFFF	10000-1FFFF	30000-3FFFF
1	0	0	00000-0FFFF	00000-0FFFF	40000-4FFFF
1	0	1	00000-0FFFF	10000-1FFFF	50000-5FFFF
1	1	0	10000-1FFFF	00000-0FFFF	60000-6FFFF
1	1	1	10000-1FFFF	10000-1FFFF	70000-7FFFF

Important Notes for 128K EPROM/Flash:

- Since the C515C initializes all port lines to 1 upon reset/power-up, this means P5.0 is selected and the code will boot starting at physical location 10000. Therefore all code must be assembled or compiled to execute as if executing from logical address 0000, as usual, but it must be placed at physical location 10000 within the device. Most PC-based EPROM/Flash device programmers will default to program the device starting at the logical address defined by the Intel hex file or binary file, but code loaded here will not be executed upon reset. To correct this, some offer a load offset or a selectable target burn address.

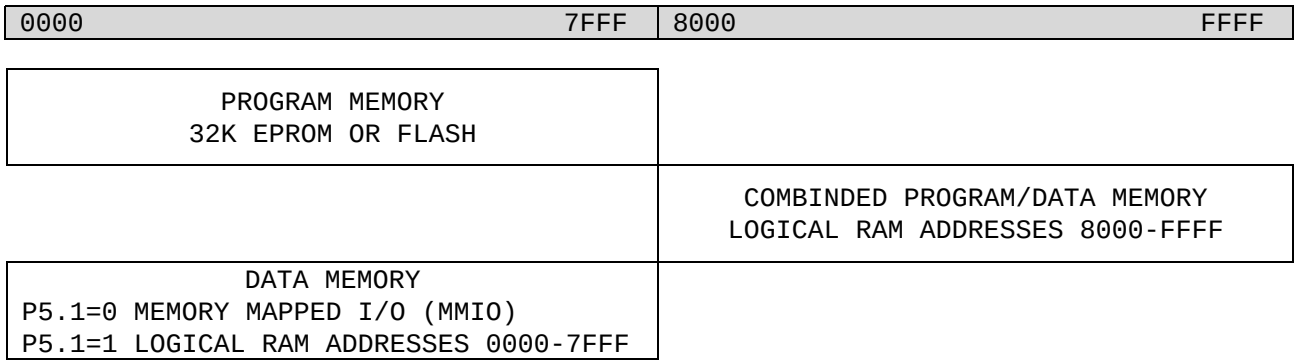
- When a 512K RAM is installed with a128K EPROM or Flash you must consider the shared P5.0 line. Changing P5.0 to access a different bank of RAM will also change the bank of EPROM/Flash and if there is no meaningful code at the next executable location within the device, of course the program will crash. To properly access this RAM, you must have a RAM access function in the128K EPROM/Flash that is mirrored in both the upper and lower 64K banks. For example, if there are RAM access subroutines residing at physical addresses 10200-10452 then the code should be duplicated also at physical addresses 00200-00452, and the code should be executable from logical addresses 0200-0452. If your application uses interrupts, the easiest solution is to disable interrupts around the function (make sure it is written to execute quickly so you don't miss any interrupts). If the code is mirrored, changing P5.0 will not affect the function. The function should set P5.0 back to the value it had before the function was called.

The following sections describe two types of logical memory maps that are available for these external memory devices depending on the positions of the jumpers in JP1.

PROGRAM/DATA CONFIGURATION

This configuration requires the use of a 32K EPROM and allows program execution in logical RAM addresses 8000-FFFF. It causes program memory and data memory, when pointing to addresses above 8000 hex, to select the same RAM addresses. To enable this, the jumpers should be set as shown below:

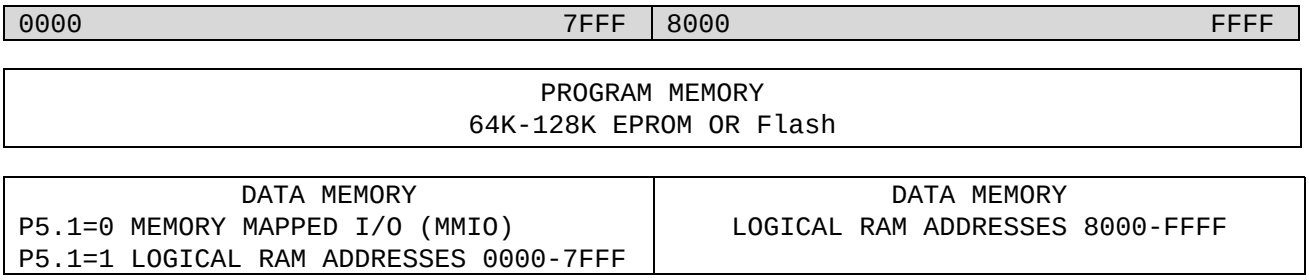
```
    1 2 3 4 5 6
JP1 o o-o o o-o
```



64K-128K EPROM/Flash CONFIGURATION

This requires the use of a 64K-128K EPROM/Flash and disables the combined program/data memory area. Set the jumpers as shown below:

```
    1 2 3 4 5 6
JP1 o-o o-o o o
```



WRITING TO FLASH

The only way to write to the Flash is to allow the C515C to access it as data memory instead of program memory, since program memory is read only. This is made possible when JP5 is in the WE (write enabled) position. In this setting, P5.4 is used to exchange the memory spaces of the RAM device with the Flash device as shown below:

P5.4 OUTPUT	Program memory space (read only with MOVC)	Data memory space (read/writable with MOVX)
1	Flash	RAM
0	RAM	Flash

Any Flash resident subroutine that manipulates P5.4 must be duplicated in the physical RAM addresses that will become program memory when P5.4 is changed. This can be done by adding code to the program initialization that will copy the subroutine from Flash to RAM. Flash can only be written to by code running in RAM.

In-circuit writing to a 64K Flash with a 32K RAM installed requires the following:

- Reset and interrupt vectors and low-level startup code which copies all Flash manipulating subroutines to RAM.
- A subroutine for writing to the Flash which first clears the P5.4 line and sets P5.1 (disables MMIO) before writing.
- The low-level code, Flash subroutines and the application must be programmed into the Flash with an external EPROM/FLASH device programmer.

In-circuit writing to a 128K Flash with a 32K RAM installed requires the same as the previous with the following exceptions:

- The Flash write subroutine must also control the A16 line (P5.0) after P5.4 is cleared and before writing to the Flash.
- The low-level code and subroutines must be programmed into the Flash starting at physical address 10000, instead of 00000.

In-circuit writing to a 128K Flash with a 128K RAM installed requires the same as the previous example. You need to make sure you to have the correct RAM bank enabled which contains the flash writing subroutines.

When designing the Flash write subroutines there are several things to keep in mind:

- The subroutine which copies the low level code and subroutines to RAM must be called before calling a Flash write routine, since there must be code in the RAM space before manipulating P5.4.

- Different Flash devices have different programming algorithms and methods for software write protection. This may involve sector protect options, and/or program commands that the Flash recognizes.

- Be aware that most devices require erasing an entire sector in order to change one byte. This requires the software to copy the sector to RAM, change the desired byte in RAM, then exchange banks, erase the sector, write back the RAM copy and finally exchange back the original memory bank setting.

- Some Flash devices have the capability to write-protect and/or erase-protect sectors of memory when programmed using an external EPROM/FLASH device programmer. This feature should be used to protect the application section(s) of the Flash or any section that needs protection.

SERIAL EEPROM

This non-volatile memory is not part of the memory map and is intended to be used to hold configuration information and other data that doesn't change often, since the device is rated for only 10,000 write cycles. The memory is organized as 64X16 bits and all access is done serially through P5. Since P5.1 is used to select the SEEPROM and also to enable/disable memory mapped I/O, interrupts should be disabled around SEEPROM routines if there is a possibility of an interrupt handler affecting P5.1.

MEMORY MAPPED I/O

Memory mapped I/O (MMIO) is implemented in the external data memory addresses 0000-7FFF when P5.1=0. For example:

```
CLR   P5.1              ; this enables MMIO
MOV   DPH,#<port addr>  ; DPH selects the port (DPL has no effect)
MOVX  A,@DPTR           ; read the current port
MOVX  @DPTR,A           ; write to the current port
```

Note that since the I/O decoding logic uses only the A8-A15 lines, only DPH (data pointer high byte register) is significant; the value of DPL has no effect. Therefore, if DPL changes but neither P5.1 nor DPH are changed, the same port will be selected. If programming in BASIC or C, you must select a full 16 bit address within the XDATA space, but keep in mind that the lower 8 bits of the address are ignored.

I/O PORTS

The I/O ports should not be confused with the special function registers. Though they may have similar purposes, the method of access is different as just shown. The I/O port addresses are as follow:

Physical MMIO Address range	Value for DPH	Description
0000H-0FFFH	00H-0FH	SC26C92 (COM1, COM2, timer/counter and digital I/O ports,)
1000H-1FFFH	10H-13H	D/A OUTPUT (channels A,B,C and D correspond to physical addresses 1000H, 1100H, 1200H and 1300H respectively)
2000H-2FFFH	20H-21H	LCD INTERFACE (2000H=LCD command, 2100H=data)
4000H-4FFFH	40H	HIGH CURRENT LATCH (HCL.0-7)
7000H-7FFFH	70H-7FH	EXTERNAL I/O (Chip select for devices connected to expansion connector)

Below is a detailed list of the SC26C92's ports (**wr** indicates a write register and **rd** a read register).

Physical MMIO Address	Value for DPH	LABEL	DESCRIPTION
0000H	00H	MR1A:	Mode register a (MR1A, MR2A) (rd/wr)
0100H	01H	SRA:	Status register A (rd)
0100H	01H	CSRA:	Clock select register A (wr)
0200H	02H	BRGTST:	Baud rate generator test (rd)
0200H	02H	CRA:	Command register A (wr)
0300H	03H	RHRA:	RX holding register A (rd)
0300H	03H	THRA:	TX holding register A (wr)
0400H	04H	IPCR:	Input port change register (rd)
0400H	04H	ACR:	Auxiliary control register (wr)
0500H	05H	ISR:	Interrupt status register (rd)
0500H	05H	IMR:	Interrupt mask register (wr)
0600H	06H	CTU:	Counter timer upper (wr)
0600H	06H	CTUR:	Counter timer upper read (rd)
0700H	07H	CTL:	Counter timer lower (wr)
0700H	07H	CTLR:	Counter timer lower read (rd)
0800H	08H	MR1B:	Mode register B (rd/wr)
0900H	09H	SRB:	Status register B (rd)
0900H	09H	CSRB:	Clock select register B (wr)
0A00H	0AH	TST1X16:	1X/16X test (rd)
0A00H	0AH	CRB:	Command register B (wr)
0B00H	0BH	RHRB:	RX holding register B (rd)
0B00H	0BH	THRB:	TX holding register B (wr)
0D00H	0DH	UARTIN:	Input port (rd)
0D00H	0DH	OPCR:	Output port configuration register (wr)
0E00H	0EH	STRTCNT:	Start counter command
0E00H	0EH	OUTSET:	Set output port bits command
0F00H	0FH	STOPCNT:	Stop counter command
0F00H	0FH	OUTRES:	Reset output port bits command (wr)

HARDWARE DESCRIPTION

The following sections describe various general aspects of the I/O devices available. Further details can be gathered by examining the drivers that were included with your MicroPac 515C.

COPROCESSOR

The standard coprocessor option adds the following features to the MicroPac 515C: an additional serial port with a buffer for input characters and output characters, and a 4 by 4 keypad decoder. All communication to it is through the C515C's SPI interface and the INT8* interrupt used as an attention signal to the C515C. The coprocessor communicates with the C515C through with a complex protocol that is beyond the scope of this manual. Please refer to the drivers for an example.

When special custom features are needed on the MicroPac 515C, many of these can be implemented on the coprocessor. EMAC can custom program the coprocessor's firmware for almost unlimited applications, so please call for a cost estimate for your requirements. The hardware features available for a custom programmed coprocessor are: 128 bytes of RAM, two 16 bit counter/timers, a serial port which can be used in synchronous or asynchronous mode, up to 12 I/O lines, and an analog comparator.

KEYPAD INTERFACE

The standard coprocessor option decodes a keypad according to the "RETURN VALUES" table. A connection between an X and Y pin will produce the hexadecimal value as shown in the table. For example a connection between X2 and Y3 will product the hexadecimal output 4A. The coprocessor communicates with the C515C through a SPI interface, with a complex protocol that is beyond the scope of this manual. Please refer to the drivers for an example.

KEYPAD CN5

```
 1
o  X4
o  X3
o  X2
o  X1
o  Y4
o  Y3
o  Y2
o  Y1
o  GND
 9
```

RETURN VALUES

```
 X1 X2 X3 X4

 41 42 43 44 -Y1

 45 46 47 48 -Y2

 49 4A 4B 4C -Y3

 4D 4E 4F 50 -Y4
```

CAN INTERFACE

The CAN controller within the C515C provides all resources that are required to run the standard CAN protocol (11-bit identifiers) as well as the extended CAN protocol (29-bit identifiers). It provides a sophisticated object layer to relieve the CPU of as much overhead as possible when controlling many different message objects (up to 15). This includes bus arbitration, resending of garbled messages, error handling, interrupt generation, etc. Refer to the C515C User's Manual for more details.

CN6

```
          1     2
P4.0/ADST* o o  Vcc
P4.4/SLS*  o o  CANH/P4.7
P4.3/STO   o o  CANL/P4.6
P4.2/SRI   o o  P4.5/INT8*
P4.1/SCK   o o  GND
          9     10
```

SPI INTERFACE

The SPI pins are available for use when the coprocessor isn't installed. They are as follows:

SCK	SSC Master Clock Output / SSC Slave Clock Input
SRI	SSC Receive Input
STO	SSC Transmit Output
SLS*	Slave Select Input

```
          CN6
        1    2
P4.0/ADST*  o o  Vcc
P4.4/SLS*   o o  CANH/P4.7
P4.3/STO    o o  CANL/P4.6
P4.2/SRI    o o  P4.5/INT8*
P4.1/SCK    o o  GND
        9    10
```

This interface differs slightly from the Motorola standard in that SRI is always an input and STO is always an output regardless of whether it is in master or slave mode. Refer to the C515C User's Manual for more details.

SERIAL PORTS

COM0, 1 and 2 serial ports may be individually configured for RS-232 or RS-422, and may be programmed for 8 or 9 bit protocol. Baud rates from 50 to 230.4k baud are available for COM1 and COM2 as well as higher non-standard baud rates. COM0 offers standard rates with a slight deviation (0.16% at 9600 baud). An optional fourth serial port, COM3, is available with the RS-232/Keypad coprocessor option. It is configured as RS-232 only.

The RS-422 option allows up to 32 serial ports to use the same twisted pair. Each end of the pair should have a terminating resistor that matches the characteristic impedance of the line. Typically 33 ohms can be used.
The serial ports allow for multiprocessor communication using 9 bit protocol. This innovative function, when enabled, will interrupt the processor if the 9th bit of a data byte is set. Using the appropriate software allows for a powerful communication scheme.

COM0

This port is the C515C serial interface which has 4 modes of operation. Since mode 0 is not directly supported by the hardware, only modes 1, 2 and 3 will be described here.

MODE 1: 10 bits are transmitted or received: a start bit of 0, 8 data, and a stop bit of 1. The baud rate is variable.

MODE 2: 11 bits are transmitted or received: a start bit of 0, 8 data, a programmable 9th bit, and a stop bit of 1. When transmitting, the 9th bit comes from TB8 in SCON. This could be used to hold the parity of the data. When receiving, the 9th bit goes to RB8 in SCON, while the stop bit is ignored. The baud rate can be either 1/32 or 1/64 of the oscillator frequency

(RS-232 CONFIGURATION)

```
     1   2   CN1
n.c.  o o  n.c.
Tx    o o handshake in (P5.7)
Rx    o o handshake out (P5.6)
n.c.  o o n.c.
GND   o o n.c.
     9   10
```

```
(RS-485 CONFIGURATION)
     1   2   CN1
n.c.  o o  n.c.
TxA   o o  RxB
TxB   o o  RxA
n.c.  o o  n.c.
GND   o o  n.c.
     9   10
```

MODE 3: Same as mode 2 except the baud rate is variable.

For standard baud rates in modes 1 and 3, the overflow rate of TIMER 1 and the setting of SMOD (PCON. 7) can be used to control the baud rate.

COM0 HANDSHAKE LINES

An input and output handshake line is available with the standard RS-232 option which allows you to implement software handshaking. The input line is connected to P5.7 and you will read a 0 or 1 for a + or - RS-232 level, respectively (note that if a 0 has been written to P5.7, 1 must be written to it to make it an input). The output line is connected to P5.6, and similarly a 0 or 1 output will produce a + or - RS-232 level, respectively.

When the RS-422 option is installed for COM0, an output of 1 or 0 on P5.6 enables or disables the transmitter, respectively.

(RS-232 CONFIGURATION)

```
COM1 (CHANNEL A)
     1   2   CN2
n.c.  o o  n.c.
Tx    o o  handshake in (IP4)
Rx    o o  handshake out (OP0)
n.c.  o o  n.c.
GND   o o  n.c.
     9   10

COM2 (CHANNEL B)
     1   2   CN3
n.c.  o o  n.c.
Tx    o o  handshake in (IP5)
Rx    o o  handshake out (OP1)
n.c.  o o  n.c.
GND   o o  n.c.
     9   10

(RS-485 CONFIGURATION)
   CN2 & CN3
     1   2
n.c.  o o  n.c.
TxA   o o  RxB
TxB   o o  RxA
n.c.  o o  n.c.
GND   o o  n.c.
     9   10
```

COM1 AND COM2

These ports are in the SC26C92 with COM1 and COM2 corresponding to channel A and channel B respectively. The ports are very versatile as seen in the following list of features:

Eight character FIFO for each Txer and Rxer
Programmable data format
- 5 to 8 data bits plus parity
- odd, even, no parity or force parity
- 1, 1.5 or 2 stop bits programmable in 1/16-bit increments
- Programmable baud rate for each receiver and transmitter
- Parity, framing, and overrun error detection
- False start bit detection
- Line break, and mid character break detection

Programmable channel mode
- Normal (full-duplex)
- Automatic echo
- Local loopback
- Remote loopback

Automatic wake-up mode for multidrop applications

COM1, COM2 HANDSHAKE LINES

The RS-232 configuration provides handshake inputs for COM1 and COM2 which may be examined by reading IP4 and IP5 respectively. Reading a 0 indicates a + RS-232 input and 1 indicates a - RS-232 input. Also, OP0 and OP1 control the handshake outputs for COM1 and COM2 respectively. Normally these are controlled by using the OUTSET and OUTRES commands, but setting bit 5 of MR2A (for COM1) or MR2B (for COM2) will enable the Request-to Send (RTS) function. When enabled, this function will bring the corresponding handshake out line to a - RS-232 level one bit time after the characters in the Channel A (or B for COM2) transmit shift register and in the THR, if any, are completely transmitted including the programmed number of stop bits, if the transmitter is not enabled. To automatically terminate a string of characters, do the following:

- Enable the RTS function
- Enable the transmitter
- Set handshake out line to + RS-232 level using the OUTSET command
- Send string, and immediately after writing last character to THR, disable transmitter

The last character will be sent and the handshake out line will go to a - RS-232 level one bit time afterward.

In the RS-422 configuration OP0 and OP1 are used to enable the transmitters for COM1 and COM2 respectively. Use the OUTRES command to enable and OUTSET to disable. For more details, see the SC26C92 data sheets.

COM3

This is part of the RS-232/Keypad coprocessor option. The current version (as of the date of this manual) has a fixed baud rate and protocol of 9600,N,8,1. There is no software connection to the handshake lines on this connector. The handshake input pin is converted by the level converting chip and then output again at the same polarity to the handshake output pin. In other words, the handshake output follows the level of the handshake input.

```
     CN4
     1   2
n.c.  o o  n.c.
Tx    o o  handshake in
Rx    o o  handshake out
n.c.  o o  n.c.
GND   o o  n.c.
     9   10
```

DIGITAL I/O

PX0-PX7

The port may be controlled by writing to MMIO address 4000h.
Each line of HCL.0 through HCL.7 is individually capable of sinking 500mA, and the driver package power dissipation limit is 2.25W at 70F.

PX8-PX18

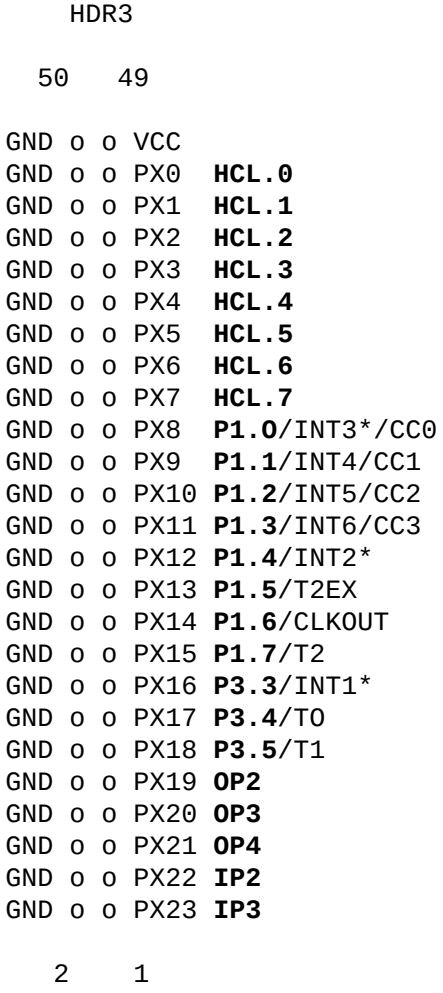
These ports are bidirectional. When a 1 is written to a port line it is configured as a input/output port. It is considered an output port because it is outputing a logical 1, yet as soon as it is externally brought low by an input, reading this port will return a 0. When there is no input, or when a 1 is applied externally, reading this port will return a 1. When a 0 is written to the port, it outputs a 0, but now cannot be used as an input port. During reset all the port latches of the C515C have 1s written to them.

When PX8-PX18 are used as outputs they can drive 4 LS-TTL inputs. When changing the state of an output bit (or bits), the following assembly language instructions should be used:

ANL,ORL,XRL,JBC,CPL,INC,DEC,DJNZ,
MOV Px.y,C (Move carry bit to bit Y of port X),
CLR Px.y (Clear bit Y of Port X),
SET Px.y (Set bit Y of Port X).

These instructions should be used because they read the latch of a line instead of the signal being output (or input) to the line. This is important because if a load, such as the base of a transistor, is connected to one of the output lines, reading this port with instructions which read the line instead of the latch (such as MOV A,P1) will always return a 0 because of the base's low voltage drop. Instructions such as MOV A,P1 should be used when the ports are input only.

HDR3



PX19-PX23 and IP0, IP1

The remaining lines PX19-PX23 are supplied by the optional SC26C92, with OP2,OP3 and OP4 being output only, and IP0, IP1, IP2, and IP3 being input only. The inputs IP0 and IP1 are on the I/O Expansion Connector (discussed later). The SC26C92 inputs and outputs are defined as follows:

OP2:General purpose output, or Channel A transmitter 1X or 16X clock output, or Channel A receiver 1X clock output.

OP3:General purpose output, or open-drain, active-low counter/timer output or Channel B transmitter 1X clock output or receiver 1X output.

OP4:General purpose output, or open-drain, active-low RxRDYA/FFULLA

IP0:General purpose input, or Channel A clear to send active low input.

IP1:General purpose input, or Channel B clear to send active low input.

IP2:General purpose input, or counter/timer external clock input.

IP3:General purpose input, or Channel A transmitter external clock input (TxCA).

When the SC26C92 is reset, the following registers will be cleared: SRA, SRB, IMR, ISR, OPR, and OPCR. It will also put OP0-OP7 into the high state, stop the counter/timer, and make channel A and B inactive with the TxDA and TxDB outputs in the high state.

After reset OP2-4 default as general purpose outputs. If after reset they have been programmed for another function they may be programmed to outputs again by programming register OPCR (see data sheets). When these ports are configured as outputs they output the complement of the data in OPR (an internal register of the SC26C92) which is controlled by the OUTSET and OUTRES ports.
For example, to make OP2 high we must make bit 2 of OPR low (since the complement of OPR is output). A 1 in any of the bits that are written to the OUTRES port will result in the corresponding bit in OPR being brought low (reset), allowing you to selectively reset any one or multiple bits without affecting the others. The following code does this.

```
CLR   P5.1              ; enable MMIO
MOV   DPH,#OUTRES       ; select OUTRES port
MOV   A,#00000100B      ; bit 2 of A is set
MOVX  @DPTR,A           ; write A to oUTRES resetting bit 2 of OPR only
```

...subsequently writing the same value to oUTSET port will make OP2 high:

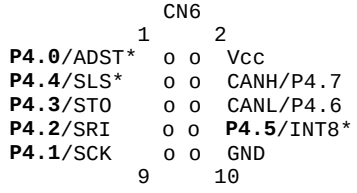
```
MOV   DPH,#OUTSET       ; select OUTSET port
MOVX  @DPTR,A           ; write A to oUTSET setting bit 2 of OPR only
```

As shown above, OUTSET works the same as OUTRES except it selectively sets bits in OPR.

The inputs IP0, IP1, IP2 and IP3 may be programmed to be general purpose inputs; they are unlatched and may be examined by simply reading the UARTIN port and examining bits 0,1,2 and 3. The inputs IP0-IP3 have a unique feature in that they can also detect a change in state. This is defined as a high-to-low, or low-to-high input transition that lasts longer than 50 uS. When the IPCR port is read, bits 0-3 will read the same as the corresponding bits in the UARTIN port, and bits 4-7 if set to 1 will indicate a change of state for IP0-IP3 respectively. After the port is read, bits 4-7 are automatically reset. For more details on SC26C92 input and output, see the SC26C92 data sheets.

P4.0-P4.5

When the coprocessor is not installed, 6 port lines are available for general I/O use on CN6. Refer to the C515C User's Manual for more details.



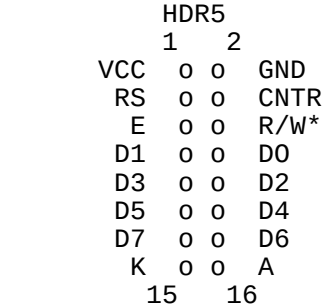
LCD INTERFACE

The LCD interface allows the MicroPac to control character displays and graphic displays depending on the JP3 jumper settings shown below:

1 **o-o o-o o** Graphic mode setting
1 **o o-o o-o** Character mode setting (backlight always on)
1 **o o o o-o** Character mode setting (backlight controlled)

The interface can support character LCD's that have HD44780 or compatible controllers. It can support graphic LCDs which have HD61830 or compatible controllers.

The backlight is controlled by P3.2 (or instead, OP7 if the dual serial port upgrade is installed). The backlight control pin must be turned high to turn on the backlight.



Character Mode
K = backlight cathode
A = anode

Graphic Mode.
K = CS*
A = RES*

TIMER/COUNTERS

HDR3

50 49

GND o o VCC
GND o o PX0 HCL.0
GND o o PX1 HCL.1
GND o o PX2 HCL.2
GND o o PX3 HCL.3
GND o o PX4 HCL.4
GND o o PX5 HCL.5
GND o o PX6 HCL.6
GND o o PX7 HCL.7
GND o o PX8 P1.0/INT3*/**CC0**
GND o o PX9 P1.1/INT4/**CC1**
GND o o PX10 P1.2/INT5/**CC2**
GND o o PX11 P1.3/INT6/**CC3**
GND o o PX12 P1.4/INT2*
GND o o PX13 P1.5/**T2EX**
GND o o PX14 P1.6/**CLKOUT**
GND o o PX15 P1.7/**T2**
GND o o PX16 P3.3/**INT1***
GND o o PX17 P3.4/**TO**
GND o o PX18 P3.5/**T1**
GND o o PX19 **OP2**
GND o o PX20 **OP3**
GND o o PX21 OP4
GND o o PX22 **IP2**
GND o o PX23 IP3

2 1

TIMER 0 AND TIMER 1

These timers are each 16 bits and can be programmed individually to be in counter or timer mode, in addition to 4 different modes of operation. Note that mode 0 for T0 is not available since INT0 is used to detect the interrupt output of the SC26C92.

When using counter mode for TIMER 0 the signal is input through P3.4/T0, or for TIMER 1, through P3.5/T1. These pins as well as P1.5/T2EX, P1.7/T2, CCO, CC1, CC2 and CC3 must be configured as inputs by writing 1 to them if their default settings have been changed.

TIMER 2

This 16 bit timer is also referred to as PTRA Unit (Programmable Timer/Counter Register Array) which can be used as a timer, event counter, or gated timer. In each configuration an interrupt may be generated on counter overflow. The counter input signal (or, in gated timer mode, the timer gate) comes from P1.7/T2.

Once the timer or counter function has been enabled, 3 other modes of operation may be selected: reload, compare, and capture.

By default, when TH2 and TL2 are incremented above FFFFh, they roll over to 0000. In reload mode 1, TH2 and TL2 are loaded with the values stored in CRCH and CRCL respectively, on a falling edge of P1.5/T2EX. In reload mode 0, these values are loaded when the timer rolls over.

Compare mode is used to provide 4 PWM (pulse width modulated) output signals with 16 bits of resolution from the CC0, CC1, CC2 and CC3 lines.

Capture mode 0 can be used to decode 4 PWM signals with 16 bits of resolution again, using the CC0, CC1, CC2 and CC3 lines as capture input signals (if P1.0-1.3 are configured as inputs). Capture mode 1 can be used to read timer 2 while it is running to eliminate the problem of a wrong reading caused by a rollover occurring between the time of reading the lower and upper byte of the timer.

C/T

The designation C/T is used in the SC26C92 data sheets, to refer to the Counter/Timer. This C/T counts down instead of up as with the other timer/counters. The minimum count that can be loaded into the C/T is 0002h. The terms *counter* (which commonly refers to a device that produces an output based on the number of external signals received) and *timer* (which refers to a device with a programmable output based on a constant clock source) are used differently in the data sheets. What they refer to as timer mode shares features of a counter, and vice-versa. For example, there are 2 timer modes which use an external input (IP2) to clock the timer. Though this may be confusing, we will follow their definitions in this section.

In timer mode, a square wave output will be produced which has a period of 2 * (value loaded into C/T registers) * (period of clock source). This output will set the Counter Ready interrupt flag (ISR[3]) each period (this flag is reset by the STOPCNT command, without stopping the timer), and if OPCR[3]=0 and OPCR[2]=1 the square wave will be output on OP3.

In counter mode the C/T is enabled to decrement the value loaded into it after the STRTCNT command, and upon reaching 0000 the Counter Ready interrupt flag (ISR[3]) is set and the counter will continue to decrement until a STOPCNT command, which also clears the interrupt flag. If OP3 is programmed to be a C/T output, it will remain high until it reaches the terminal count and goes low. The output goes high again after a STOPCNT command (see the SC26C92 data sheets for more details).

SIGNAL OUTPUTS

For applications that need a signal output, there are 3 sources available:

CLOCKOUT	this can be programmed to provide a system clock output of 1/6 the oscillator frequency.
OP2	this can output channel A's transmitter clock, transmitter clock times 16, or receiver clock.
OP3	this can output channel B's transmitter clock, receiver clock, or the C/T output

WATCHDOG TIMER

The watchdog timer is built into the C515C processor and is used to reset the MicroPac 515C in the event that a hardware or software crash occurs. The watchdog timer can be enabled and configured so that the software must clear the timer at least once every 0.6 seconds, otherwise a reset will occur. This reset is the same as an external reset, except that bit WDTS (bit 6 of IP0) is set. Using this, the software can differentiate between a watchdog reset and an external reset. See the section titled “Programmable Watchdog Timer” in the C515C User’s Manual for programming details.

INTERRUPTS

The C515C has 5 internal interrupt sources and 7 external sources, and the SC26C92 has a single interrupt output with eight maskable interrupt sources. As these devices are configured on the MicroPac 515C, this works out to 9 external interrupt sources (10 available when timer 2 reload mode is disabled) and 9 on-board. The external interrupts IP0 and IP1 are available on the expansion connector (the connector is discussed later), INT8* is on CN6 (if RS-232/Keypad upgrade are not installed)and the rest are available on HDR3. The interrupt sources are as follows:

EXTERNAL

INT1* -programmable for falling edge or low level trigger.
INT2* -programmable for rising or falling edge trigger.
INT3* -programmable for rising or falling edge trigger.
INT4 -rising edge triggered.
INT5 -rising edge triggered.
INT6 -rising edge triggered.
INT8* (CN6)-low level triggered.

T2EX/P1.5 -when timer 2 reload mode is disabled, it is falling edge. Triggered

T0,T1,T2 -each can be configured as falling edge interrupt inputs by setting them in auto-reload counter mode and loading the count and reload registers with the maximum count value. A falling edge on the respective timer’s input will cause the counter to roll over and generate an interrupt.

HDR3

50			49	
GND	o	o	VCC	
GND	o	o	PX0	HCL.0
GND	o	o	PX1	HCL.1
GND	o	o	PX2	HCL.2
GND	o	o	PX3	HCL.3
GND	o	o	PX4	HCL.4
GND	o	o	PX5	HCL.5
GND	o	o	PX6	HCL.6
GND	o	o	PX7	HCL.7
GND	o	o	PX8	P1.O/**INT3***/CC0
GND	o	o	PX9	P1.1/**INT4**/CC1
GND	o	o	PX10	P1.2/**INT5**/CC2
GND	o	o	PX11	P1.3/**INT6**/CC3
GND	o	o	PX12	P1.4/**INT2***
GND	o	o	PX13	P1.5/**T2EX**
GND	o	o	PX14	P1.6/CLKOUT
GND	o	o	PX15	P1.7/**T2**
GND	o	o	PX16	P3.3/**INT1***
GND	o	o	PX17	P3.4/**TO**
GND	o	o	PX18	P3.5/**T1**
GND	o	o	PX19	OP2
GND	o	o	PX20	OP3
GND	o	o	PX21	OP4
GND	o	o	PX22	**IP2**
GND	o	o	PX23	**IP3**
2			1	

Through INT0/P3.2 the following SC26C92 interrupts are detected (each generates an interrupt on the is rising *and* falling edge):

IP0 (HDR1, pin 18)
IP1 (HDR1, pin 16)
IP2
IP3
IP4 (via COM1 handshake-in)
IP5 (via COM2 handshake-in)

INTERNAL

A/D Converter
Timer 0
Timer 1
Timer 2
COM0 (C515C serial port)

Through INT0/P3.2 the following SC26C92 interrupts are detected:

COM1 (channel A)
COM2 (channel B)
Counter Ready

SC26C92 INTERRUPTS

All of the SC26C92 interrupts (internal and external) are detected by the C515C's INT0*/P3.2 input and it is suggested that INT0 be set up as low level triggered. To enable SC26C92 interrupts you must first select which of its interrupts will be enabled by setting the corresponding bits in the IMR port. After this you must set bits EAL and EX0 of IEN0. All interrupts are globally disabled unless EAL is set, and setting EX0 enables INT0. Now any SC26C92 interrupt will cause an INT0 interrupt.

When an INT0 interrupt occurs, its interrupt handler must find the source or sources of the SC26C92 interrupt. When there is only 1 enabled, you know that the source is the interrupt that is enabled, but if more than 1 is enabled, some extra code is needed in the handler to vector to the various SC26C92 device handlers.

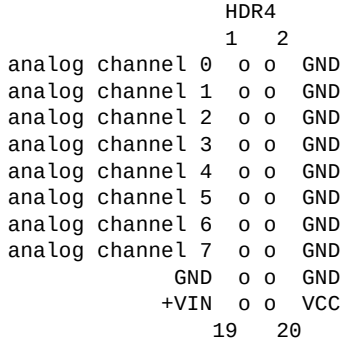
Since the ISR port can indicate pending interrupts even when the interrupts are masked off by IMR, we want to determine which of the devices have both an ISR bit set and a corresponding IMR bit set. This may be done by logically ANDing the value read from the ISR port with the value written to the IMR port (since IMR is write only, the value written to it must be a constant or must be stored in a memory location to be retrieved by the interrupt handler). Once the sources have been determined, the INT0 handler must vector to each of the corresponding SC26C92 device handlers.

The SC26C92 can be programmed to generate interrupts when a change of state occurs on any of the IP0-IP3 inputs. See Digital I/O section for description of change of state, and see the SC26C92 data sheets for detailed information on its interrupts and associated registers.

ANALOG INPUTS

The 8 channels of the10 bit A/D converter have a voltage range of 0-5V. With a 9.6 microsecond conversion rate, it can provide up to104K samples per second. A/D inputs may be also be used as 8 additional digital inputs. See the driver source code which was included with the MicroPac 515C for programming examples, or the C515C User's Manual for more information.

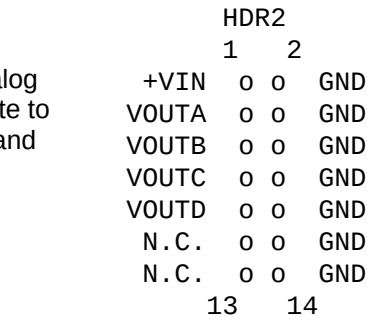
In the diagram of HDR4, +VIN is the voltage being applied to pin 1 of ST1.



DIGITAL TO ANALOG CONVERTER

The D/A converter has four channels of eight bit resolution with 0 to 5 volt analog output range. Writing to MMIO ports 1000H, 1100H, 1200H and 1300H will write to VOUTA, B, C, and D respectively. Writing 0 to a port causes 0V to be output and writing FF hex causes 5V to be output.

+Vin is the unregulated DC that is applied to ST1



EXPANSION CONNECTOR

This connector allows connection to EMAC peripherals, as well as peripherals of your own design. Included are:

AD0-AD7	8 bit data bus
A8-A15	upper byte of the address bus
EXTIO*	external I/O select line which is asserted low when accessing MMIO addresses 7000h to 7FFFh. This allows up to 16 distinct external I/O ports to be decoded from this connector (see section on Memory Mapped I/O).
WR*	active low write select line
RD*	active low read select line
RESET	active high RESET controlled by the power-on reset circuit and when pin 3 of ST1 is pulled low
INT0, INT1	These are the SC26C92 IP0 and IP1 inputs respectively which can be programmed as general purpose inputs, UART CTS's, or interrupts that are triggered on both the rising and falling edge of an input. These pins are not to be confused with the C515C's pins of the same name. These names were used for compatibility with existing EMAC peripherals.
GND, +VIN	These provide ground and the voltage supplied to pin 1 of ST1.

The reserved pins are not connected on this revision of the MicroPac 515C but may be in future revisions, so to maintain compatibility they should be left unconnected in peripherals of your own design.