Woods Hole Oceanographic Institution

WHOI Micromodem GPIO v1.0

24 December 2003
Document Revision: Draft H

Jim Partan
Phone: 508-289-3534
Email: jpartan@whoi.edu
Fax: 508-457-2195
Mail: WHOI MS-18, Woods Hole, MA 02543

Table of Contents:

Revision History:

Revision Draft H, 24 Dec 2003, Jim Partan: Added Gerbers, added differential impedance calculation, revised BOM to include Digi-Key part numbers and note on how Excel BOM is generated, revised fab & assembly rules to reflect actual production, revised production notes to reflect actual production, revised layout rules to reflect actual layout. Waiting only for new mechanical drawing and test of actual boards before changing to non-draft documentation.

Revision Draft G, 11 Dec 2003, Jim Partan: changed U19 from 56-QFN to 56-SSOP to ease assembly. Changed: Schematics(USB), BOM, netlist, Layout Rules, Fab & Assembly Rules, Coversheet, Changelog. Ideally would have changed layout_top.png and layout_bottom.png and Mechanical Drawing, but didn't. Added Amkor Application Note on 56-QFN package to Datasheets.

Revision Draft F, 10 Dec 2003, Jim Partan: added application-level software notes on $CCCFG,PPS and $CCMEC proposed changes.

Revision Draft E, 10 Dec 2003, Jim Partan: changed RX_SYNC to EXT_SYNC (gpio.c, gpio.doc/pdf; schematics: connectors, interrupts, registers, registers (check), & pdf; netlist; hardware documentation/pdf; gpio_app.doc/pdf; changelog.txt; Coversheet.doc/pdf.

Revision Draft D, 9 Dec 2003, Jim Partan: changed Y1 from ECS-240-20-7S to CS10-24.000MABJ, and hence changed USB schematic, BOM, netlist, and hardware documentation. Changed gpio.c Check Register wrappers to call ERROR_routine(), not ERROR_system(). Updated gpio_app.doc to mention the ordering of RX_ACTIVE, RX_PENDING transitions.

Revision Draft C, 9 Dec 2003, Jim Partan: Added Layout Rules; fixed Schematics (pinout for U19 QFN, USB connector shield, USB_VBUS components, added D+/D- testpoints, added Keystone-5015 PCB footprint, pin numbers for '573); slight Hardware Documentation update; slight update to Fabrication & Assembly Rules (at least one rule still in draft form); very slight change to Production notes; updated Table of Contents.

Revision Draft B, 3 Dec 2003, Jim Partan: a couple of cleanups on the schematics (testpoint for GND, pull-down on USB_5V_REG). Another iteration on the fabrication and assembly rules (still need to finish controlled impedance and reflow rules for Cypress SX2 chip). Still need to add sections in the hardware documentation section on USB build options, perhaps mention testpoints too, and ideally waitstates and power consumption. Main additions were software descriptions, both register level access and application-level examples, and adding Keenan Ball's mechanical model for the board.

Revision Draft A, 2 Dec 2003, Jim Partan: initial release.

Micromodem GPIO v1.0 Hardware Documentation

10 Dec 2003
Jim Partan, WHOI
Document Revision: D

This document is a companion to the Micromodem GPIO v1.0 schematics, providing an overview of the design. The main purpose of the GPIO board is to provide general-purpose inputs and outputs, an external Pulse-Per-Second input, and signals indicating transmitter and receiver state. A secondary purpose is to provide a high-speed offload method for the Micromodem Flash Memory board. The GPIO board connects directly above the Micromodem mainboard, below the Multi-Channel Analog Sampling board and the Flash Memory board.

TABLE OF CONTENTS

CONNECTORS:

JP1:
Connects the Micromodem mainboard to the GPIO board, using the Micromodem's External Interface. The power supply rail is bypassed with a bulk capacitor, and inputs are pulled down or up as appropriate.

JP2:
Connects the GPIO board to the Micromodem flash board, possibly passing through the multi-channel analog sampling board. /Flashbusy is pulled up in case the flash board is not connected. Note that SLE (flash chip chip-Select Latch Enable) is re-generated on the GPIO board, and is not passed directly through to the flash board.

JP3 & JP4:
Connects the Micromodem mainboard to the GPIO board, using the Micromodem's JP2 SPI expansion header. JP3 is almost completely passed through to JP4 and the multi-channel analog sampling board. The only connection the GPIO board makes to these headers is on /EXTINT – an open-drain external interrupt connection. There are no GPIO components above the Micromodem mainboard's analog planes. The GPIO board's power and ground planes do not extend over the Micromodem's analog planes, either.

JP5:
Connector for general-purpose inputs and outputs, as well as several special-purpose inputs and outputs. There are four general-purpose inputs and four general-purpose outputs. The special-purpose I/O lines are:

- TX_ACTIVE: output, high when the Micromodem is transmitting.
- RX_ACTIVE: output, high when the Micromodem is decoding a packet.
- RX_PENDING: output, high when the Micromodem has detected a cycle-init, and is anticipating receiving a packet within the cycle-timeout duration.
- EXTPPS: input, external Pulse-Per-Second line, with rising edge at the start of a second. Causes an /EXTINT interrupt.
- EXT_SYNC: input, for an external system to notify the Micromodem of a synchronization event. (One example would be to let the Micromodem know that a transmission has occurred, and to start the detector.) The rising edge of this line is the synchronization signal, and causes an /EXTINT interrupt.
- TX_INHIBIT: input. When an external system holds this line high, the Micromodem will not transmit.

All inputs and outputs can interface to either 3.3V or 5V logic. All inputs are low when the input is below 0.8V, and high when the input is above 1.7V. The maximum recommended input is 5.5V; any voltage above 6.5V will permanently damage the GPIO board. High-level outputs are 3.3V. All inputs and outputs have 100-Ohm series resistors rated to 130mW, to allow an output to be accidentally shorted to ground. All inputs are pulled down; a disconnected input will be read as low (0).

JP6:
USB connector for a "Mini-B" USB peripheral cable. Case shield is connected to ground through a resistor.

CONTROL STROBES:

There are three address lines on the Micromodem's external interface. The memory map is detailed below (addresses are in '54x I/O space):

0x2000	[Flash board: Flash chip data address (Read/Write)]
0x2001	[Flash board: Flash chip Address Latch Enable (ALE) (Write-only)]
0x2002	[Flash board: Flash chip Command Latch Enable (CLE) (Write-only)]
0x2003	"Check" registers to read back values from GPIO or USB registers (Read-only)
0x2004	[Flash board: chip-Select Latch Enable (SLE) (Write-only)]
0x2005	GPIO register: Write: set outputs. Read: read inputs.
0x2006	USB register: Write: write USB control register. Read: read USB status register.
0x2007	USB chip-select: Read/Write access to the USB chip's data bus.

Note that we generate SLE on the GPIO board to reclaim addresses 0x2005, 0x2006, 0x2007, which would otherwise be used by the flash board. (The flash board's chip-Select Latch Enable (SLE) circuit only looks at A2, so it would use four addresses instead of the one it needs unless we re-generate SLE on the GPIO board.)

INTERRUPT, BUSY, AND RESET:

The reset part simply supplies an active-low, push-pull reset line to clear all GPIO board state at power-up.

The ready circuit combines the USB_READY signal and the /FLASHBUSY signal to generate an /EXTBUSY signal, which can be read from the /DSRA pin of the Micromodem mainboard's UART. The USB_READY signal can also be read from the USB status register (U12, I/O address 0x2006), to help determine whether the USB chip or a flash chip is asserting /EXTBUSY.

All external devices share the /EXTINT external interrupt. On the GPIO board, the rising edges of EXTPPS and RX_SYNC can cause interrupts, as can the USB chip. The EXTPPS, EXT_SYNC, and USB_INT signals can also be read from either the GPIO or USB status registers (I/O addresses 0x2005 and 0x2006), to determine the source of the interrupt. The EXTPPS and EXT_SYNC interrupts can be masked by bits in the USB control register (U11, I/O address 0x2006), and the USB_INT interrupt can be masked in the internal USB control registers. The interrupt service routine should acknowledge and clear the interrupt by setting then resetting the /IACK bit in the GPIO control register (U9, I/O address 0x2005). If /IACK is left low, all interrupts from this board are disabled. /EXTINT is generated with an open-drain part, to allow wired-or connections to other /EXTINT sources (such as the Micromodem Power Amplifier board). Finally, R5 can be removed to disable interrupts from this board completely.

Resistors R1 and R2 are generally Do-Not-Install; if the USB components are not installed, however, these resistors should be installed to define CMOS gate input levels.

REGISTERS:

The GPIO control register contains the external outputs and /IACK. The GPIO status register contains the external inputs and USB_INT. USB_INT is included in the GPIO status register so that all possible /EXTINT sources from this board can be checked in a single read. The TX_ACTIVE and RX_ACTIVE lines light green and red LEDs, respectively.

The USB control register contains the USB control lines, the interrupt mask bits, and the USB_CHECK control signal. The USB_CHECK signal specifies whether the USB control register (written at 0x2006) or the GPIO control register (written at 0x2005) will be read back from the check register at 0x2003. The USB status register contains the USB status lines, all /EXTINT interrupt sources, and a line to sense the +5V USB bus power rail.

REGISTERS (CHECK):

The "check" registers allow a way to verify data in the GPIO and USB control registers. There aren't enough address lines to allow read AND write access to both control register, and also read access to the status registers. So instead, we allow write access to both control registers, read access to both status registers, and one more read address (the "check" register). When the USB_CHECK bit in the USB control register is high, the check register reads the hardware values in the USB control register; when that bit is low, the check register reads the GPIO control register.

USB:

The USB chip is a Cypress EZ-USB SX2 (CY7C68001). This chip presents itself to the Micromodem's DSP as a FIFO, more-or-less, while providing a fast USB 2.0 (480 Mbit/sec nominal speed) connection to a host PC. The details of the USB protocol are handled by the SX2.

The inverter U18 inverts the USB_PKTEND and USB_WAKEUP signals so that those signals reset to inactive in the USB control register, without having to reconfigure the SX2's polarity register (POLAR). It also inverts the USB_INT so it is active-high, as are the other interrupts on this board which feed /EXTINT.

The I2C pins SCL and SDA need 2.2k pull-ups. The reserved pin needs to be tied to ground. The IFCLK clock pin can be left not-connected for asynchronous operation. Data bus pins D8-D15 are pulled down, since we have an 8-bit interface.

The control pins USB_A0-USB_A2 select which USB endpoint fifo is being accessed, or select the command interface. The read and write strobes, along with the chip-select, allow direct memory-mapped access to the USB fifos, either by programmed I/O or potentially by DMA from the DSP. Fifo status is indicated on the USB_FLAGX pins and the USB_READY pin.

Cypress notes that the Y1 crystal should be 24.0 MHz with a parallel load capacitance, but does not specify the load capacitance. With a 20pF parallel load capacitance, there are no good options; the best is the ECS-240-20-7S in the ECX-3S package. With an 18pF parallel load capacitance, there are some better options. The best package looks like the Citizen CS10-24.000MABJ.

See the Cypress Application note "High-speed USB PCB Layout Recommendations" for comments on layout and also various components: R8, C30, R9 VBUS filtering; C31 "USB Peninsula" bypassing; R19, R20 shield connection.

LAYOUT AND CIRCUIT REVISION HISTORY:

Micromodem-GPIO-v1.0, December 2003: Original board, Jim Partan.

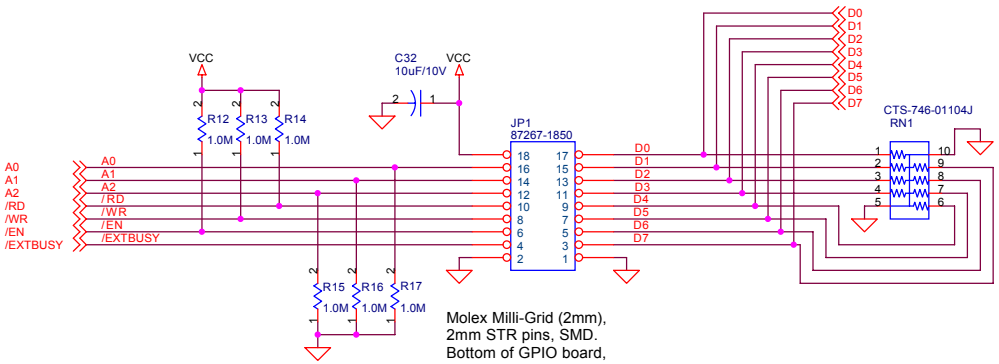
DOCUMENT REVISION HISTORY:

Revision D, 10 Dec 2003, Jim Partan: changed RX_SYNC to EXT_SYNC.

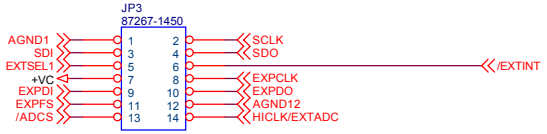
Revision C, 9 Dec 2003, Jim Partan: Changed Y1 from ECS-240-20-7S to CS10-24.000MABJ.

Revision B, 9 Dec 2003, Jim Partan: Noted USB connector case shield connections; noted Cypress Application Note on High-Speed USB layout.

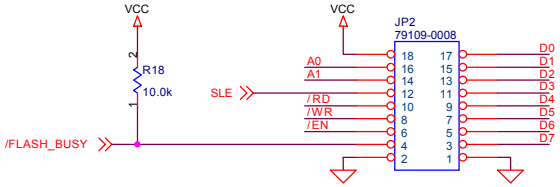
Revision A, 2 Dec 2003, Jim Partan: Initial Micromodem-GPIO-v1.0 release.



Molex Milli-Grid (2mm),
2mm STR pins, SMD.
Bottom of GPIO board,
connects to Micromodem.

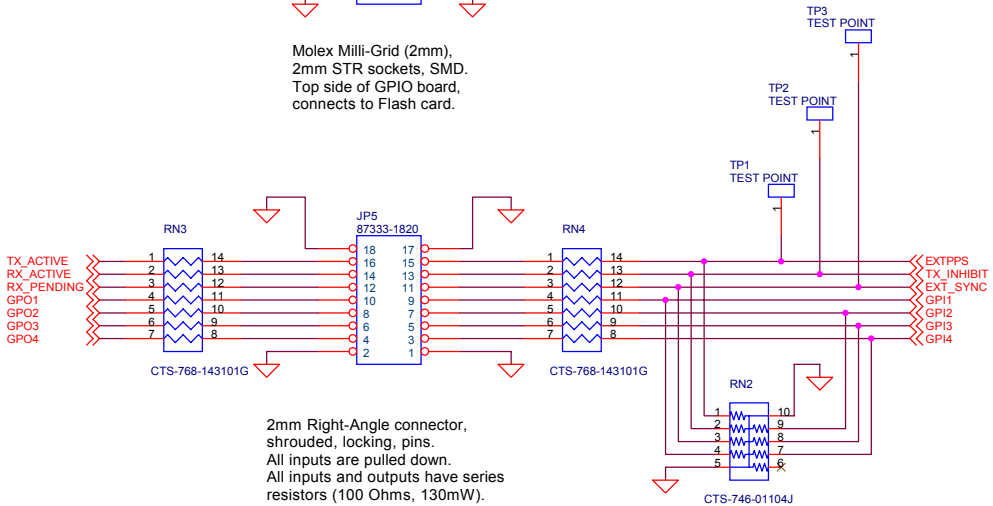


Molex Milli-Grid (2mm),
2mm STR pins, SMD.
Bottom of GPIO board,
connects to Micromodem.



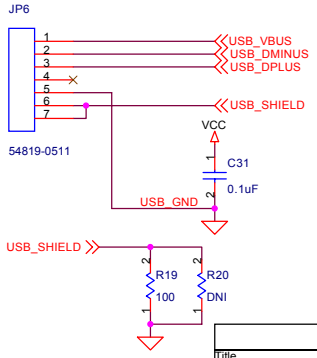
Molex Milli-Grid (2mm),
2mm STR sockets, SMD.
Top side of GPIO board,
connects to Flash card.

Molex Milli-Grid (2mm),
2mm STR sockets, SMD.
Top side of GPIO board,
connects to 4-channel card.

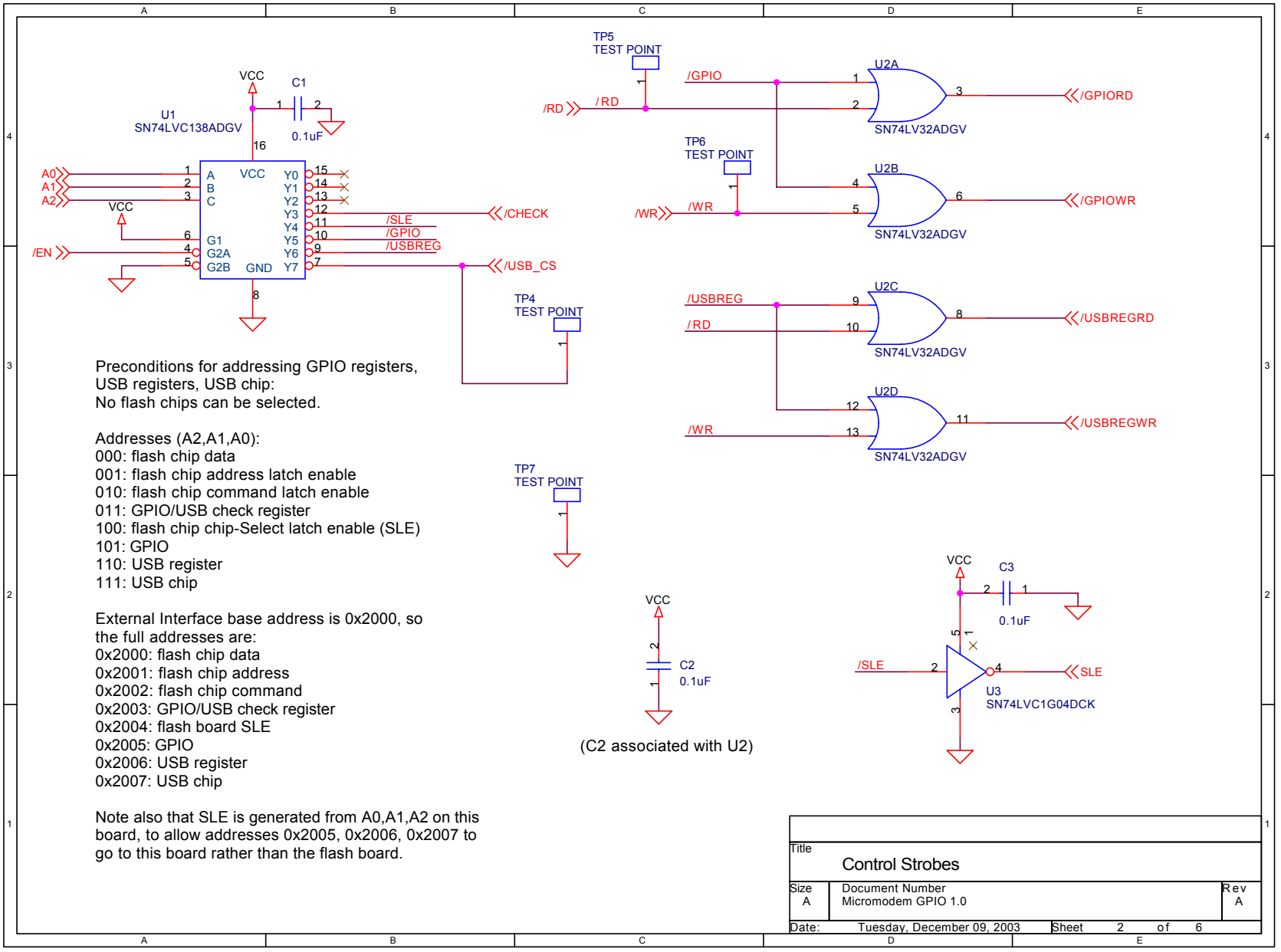


2mm Right-Angle connector,
shrouded, locking, pins.
All inputs are pulled down.
All inputs and outputs have series
resistors (100 Ohms, 130mW).

USB Peripheral Type "Mini B",
Right-Angle PCB mount for USB 2.0.
(Case pins are USB_SHIELD, here
called pins 6 and 7.)



Title: Connectors		
Size: B	Document Number: Micromodem GPIO 1.0	Rev: A
Date: Wednesday, December 17, 2003	Sheet 1 of 6	



Preconditions for addressing GPIO registers, USB registers, USB chip:
No flash chips can be selected.

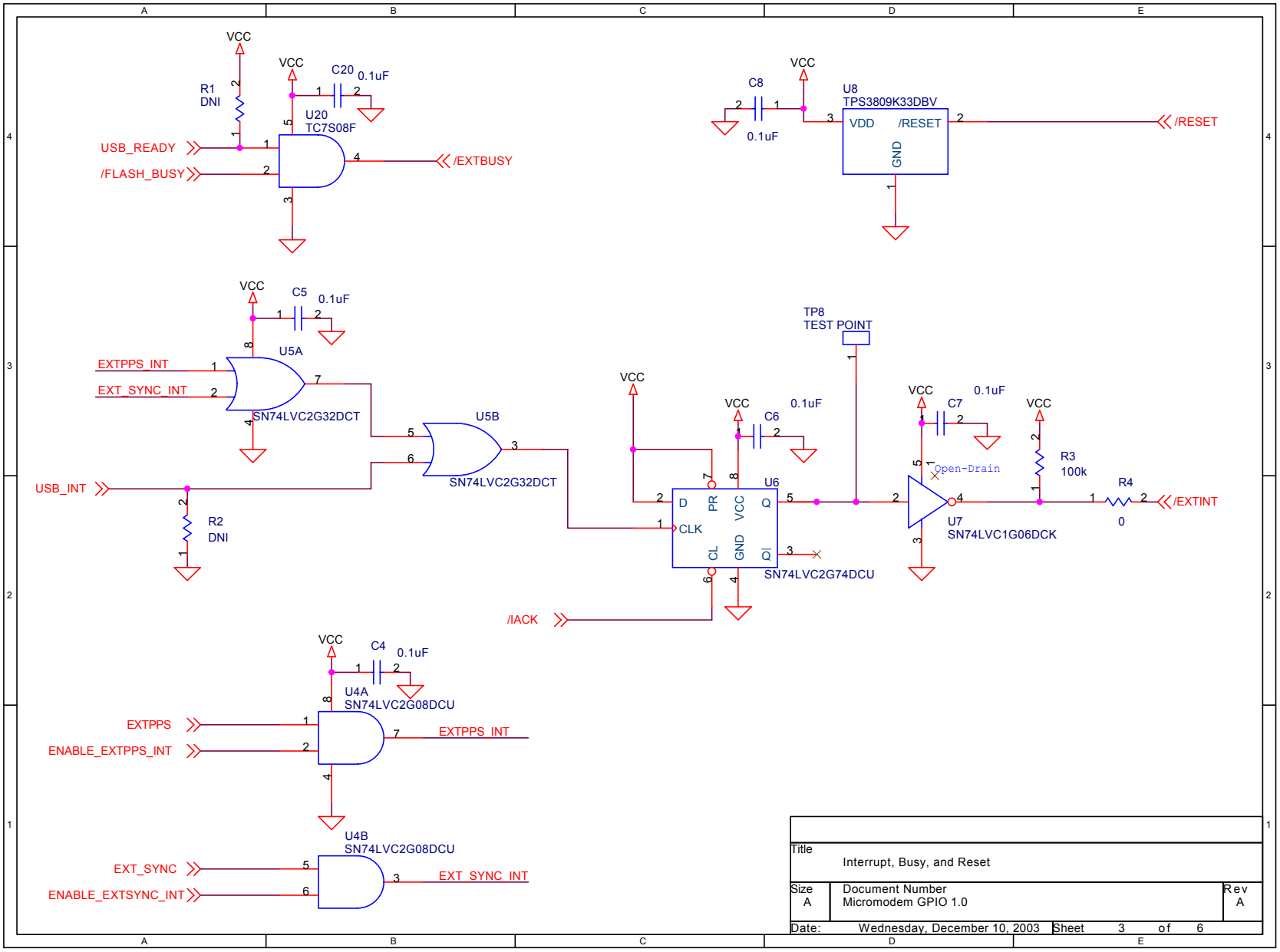
Addresses (A2,A1,A0):
000: flash chip data
001: flash chip address latch enable
010: flash chip command latch enable
011: GPIO/USB check register
100: flash chip chip-Select latch enable (SLE)
101: GPIO
110: USB register
111: USB chip

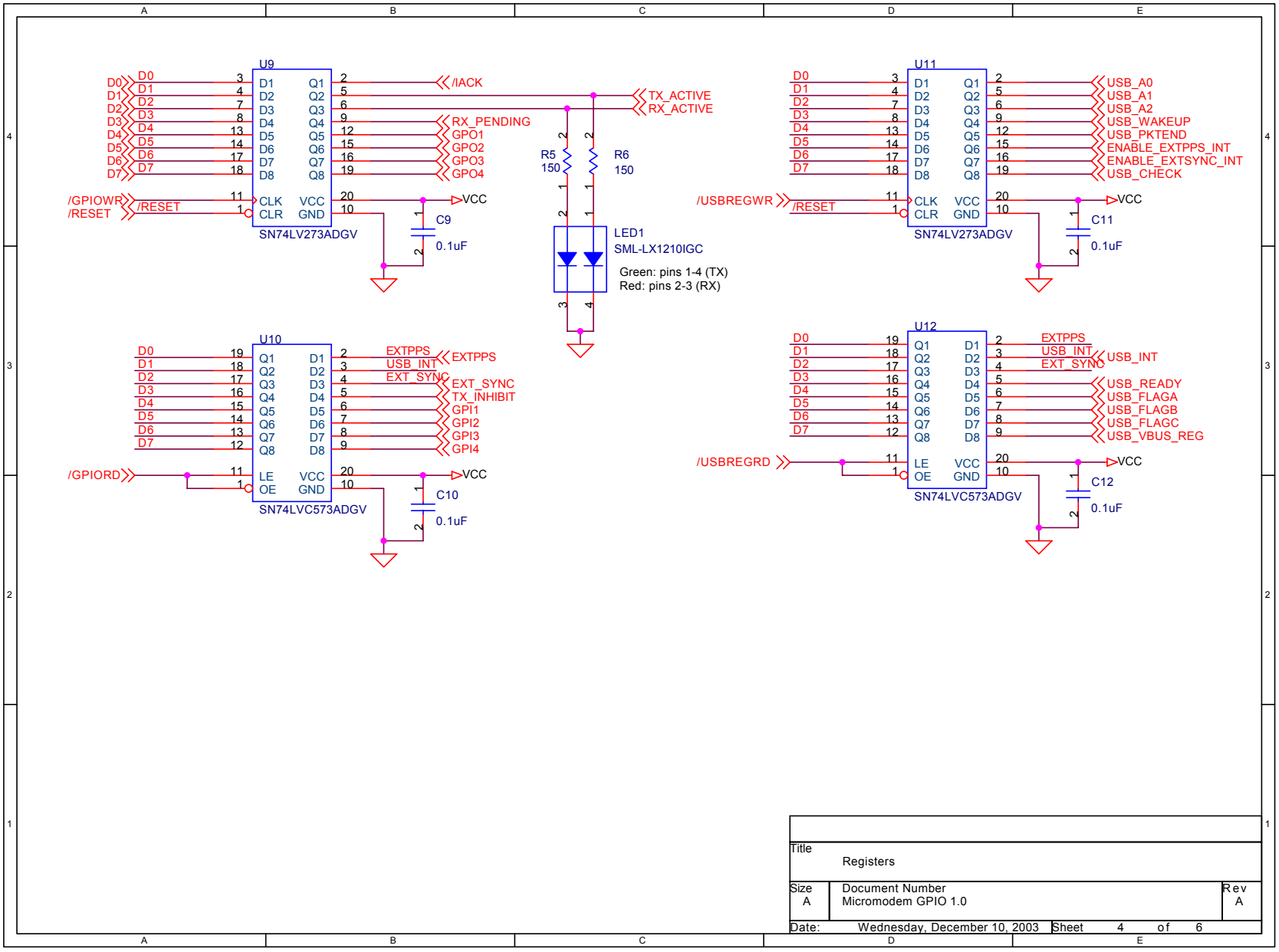
External Interface base address is 0x2000, so the full addresses are:
0x2000: flash chip data
0x2001: flash chip address
0x2002: flash chip command
0x2003: GPIO/USB check register
0x2004: flash board SLE
0x2005: GPIO
0x2006: USB register
0x2007: USB chip

Note also that SLE is generated from A0,A1,A2 on this board, to allow addresses 0x2005, 0x2006, 0x2007 to go to this board rather than the flash board.

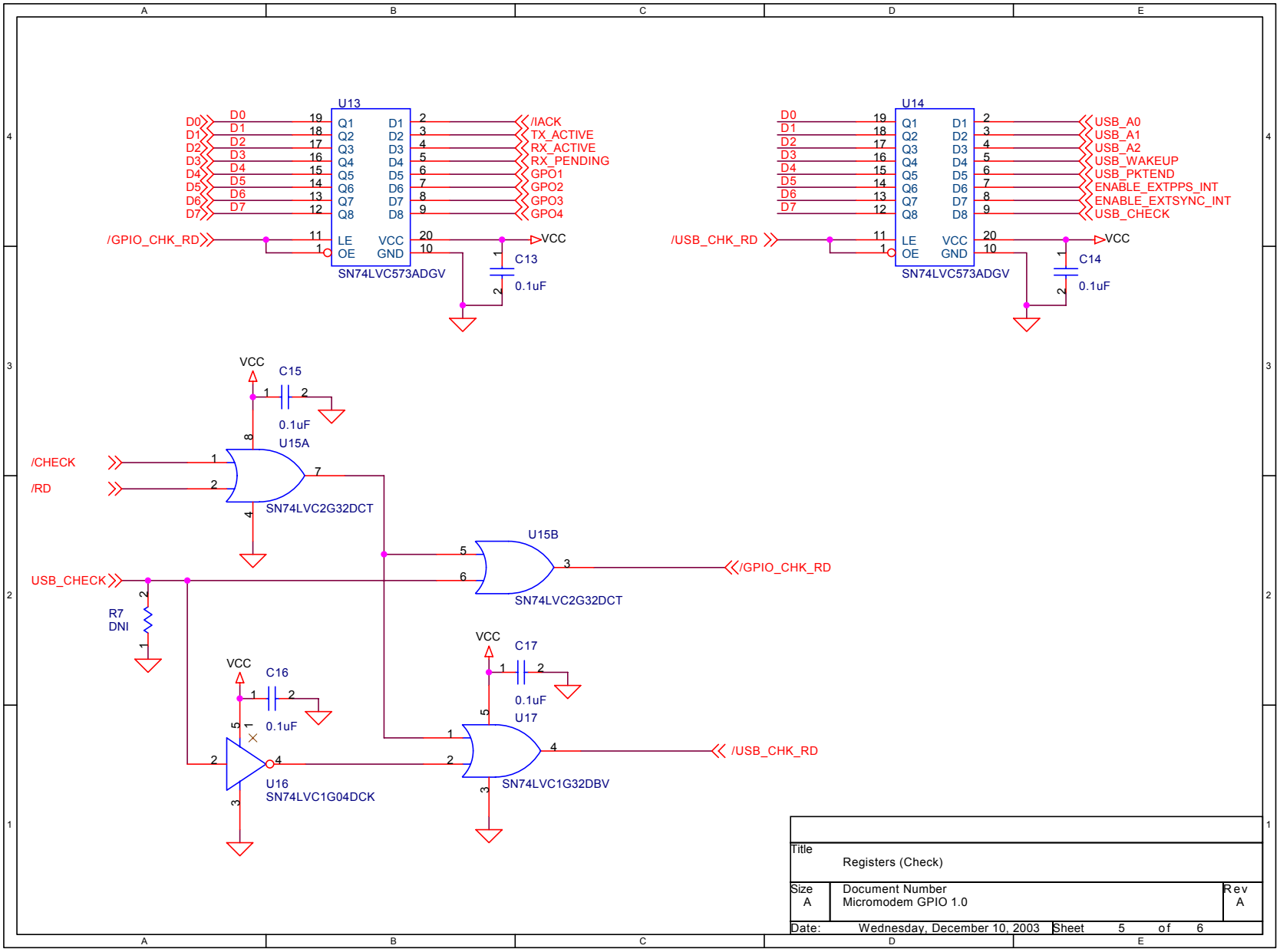
(C2 associated with U2)

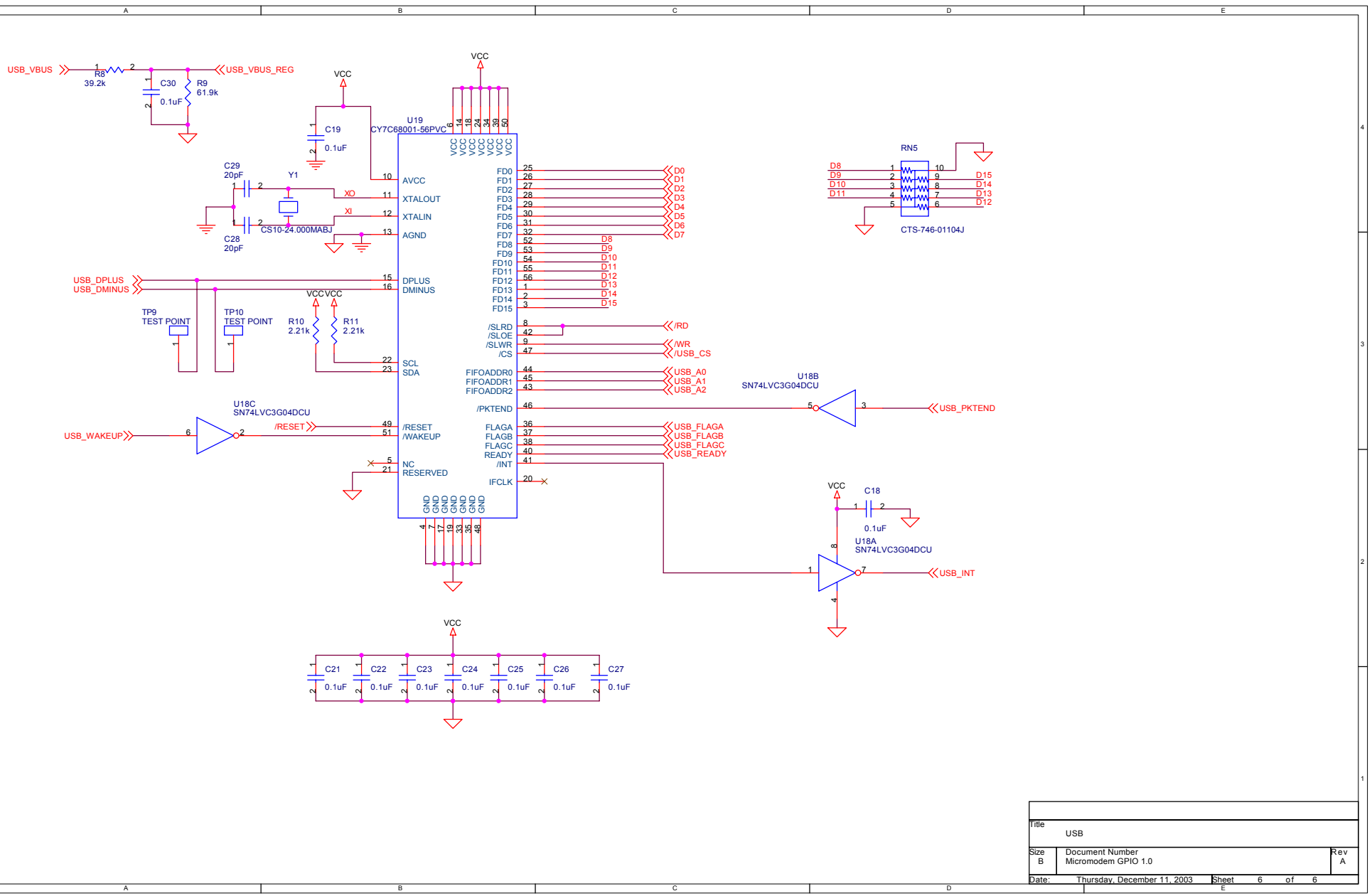
Title: Control Strobes		
Size: A	Document Number: Micromodem GPIO 1.0	Rev: A
Date: Tuesday, December 09, 2003	Sheet 2 of 6	


VCC
R1
DNI
C20 0.1uF
U20
TC7S08F
USB_READY
/FLASH_BUSY
/EXTBUSY
C8
0.1uF
U8
TPS3809K33DBV
VDD
/RESET
GND
/RESET
C5 0.1uF
U5A
EXTPPS_INT
EXT_SYNC_INT
SN74LVC2G32DCT
U5B
SN74LVC2G32DCT
USB_INT
R2
DNI
TP8
TEST POINT
C6 0.1uF
U6
D
PR
VCC
Q
CLK
CL
GND
QI
SN74LVC2G74DCU
/IACK
C7 0.1uF
Open-Drain
R3
100k
R4
0
U7
SN74LVC1G06DCK
/EXTINT
C4 0.1uF
U4A
SN74LVC2G08DCU
EXTPPS
ENABLE_EXTPPS_INT
EXTPPS_INT
U4B
SN74LVC2G08DCU
EXT_SYNC
ENABLE_EXTSYNC_INT
EXT_SYNC_INT
Title
Interrupt, Busy, and Reset
Size
A
Document Number
Micromodem GPIO 1.0
Rev
A
Date:
Wednesday, December 10, 2003
Sheet
3
of
6



Title: Registers		
Size: A	Document Number: Micromodem GPIO 1.0	Rev: A
Date: Wednesday, December 10, 2003	Sheet 4 of 6	


U13
D0
D1
D2
D3
D4
D5
D6
D7
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
D1
D2
D3
D4
D5
D6
D7
D8
/IACK
TX_ACTIVE
RX_ACTIVE
RX_PENDING
GPO1
GPO2
GPO3
GPO4
/GPIO_CHK_RD
LE
OE
VCC
GND
SN74LVC573ADGV
C13
0.1uF
U14
USB_A0
USB_A1
USB_A2
USB_WAKEUP
USB_PKTEND
ENABLE_EXTPPS_INT
ENABLE_EXTSYNC_INT
USB_CHECK
/USB_CHK_RD
C14
C15
C16
C17
U15A
U15B
/CHECK
/RD
SN74LVC2G32DCT
USB_CHECK
R7
DNI
U16
SN74LVC1G04DCK
U17
SN74LVC1G32DBV
/GPIO_CHK_RD
/USB_CHK_RD
Title
Registers (Check)
Size
A
Document Number
Micromodem GPIO 1.0
Rev
A
Date:
Wednesday, December 10, 2003
Sheet
5
of
6



Title	USB	
Size B	Document Number Micromodem GPIO 1.0	Rev A
Date: Thursday, December 11, 2003	Sheet 6 of 6	

WHOI Micromodem GPIO v1.0
Document Revision: G

Jim Partan
email: jpartan@whoi.edu
phone: 508-289-3534
fax: 508-457-2195
mail: WHOI MS-18, Woods Hole, MA, 02543

Bill Of Materials December 24,2003 11:19
Note: Columns 1-7 are taken DIRECTLY from the Orcad BOM, with the exception of Item 1, for which the Panasonic part is substituted.

Item	Qnty	Reference	Part	PCB Footprint	MFG	MFG PN	Alternate Part,Notes	Marking	Digi-Key PN
1	29	C1,C2,C3,C4,C5,C6,C7,C8,	0.1uF	603	Panasonic	ECJ-1VB1C104K	AVX 0603YC104KAT2A		PCC1762CT-ND
		C9,C10,C11,C12,C13,C14,							
		C15,C16,C17,C18,C19,C20,							
		C21,C22,C23,C24,C25,C26,							
		C27,C30,C31							
2	2	C28,C29	20pF	603	Murata	GRM1885C2A200JA01D			490-1334-1-ND
3	1	C32	10uF/10V	TE-X	Panasonic	ECS-T1AX106R		Polarity Bar (+)	PCS2106CT-ND
4	1	JP1	87267-1850	87267-1850	Waldom / Molex	87267-1850			WM18163-ND
5	1	JP2	79109-0008	79109-0008	Waldom / Molex	79109-0008			WM18022-ND
6	1	JP3	87267-1450	87267-1450	MOLEX	87267-1450			WM18161-ND
7	1	JP4	79109-0006	79109-0006	MOLEX	79109-0006			WM18020-ND
8	1	JP5	87333-1820	87333-1820	Waldom / Molex	87333-1820			WM18095-ND
9	1	JP6	54819-0511		Molex	54819-0511			WM5974-ND
10	1	LED1	SML-LX1210IGC	SMLX1210	LUMEX	SML-LX1210IGC		Cathode mark	67-1361-1-ND
11	3	RN1,RN2,RN5	CTS-746-01104J	CTS-746	CTS	746X101104J		104	746X101104JCT-ND
12	2	RN4,RN3	CTS-768-143101G	CTS-768	CTS	768143101G		768143101G, CTS	768-143-R100-ND
13	3	R1,R2,R7	DNI	603	Panasonic	ERJ-3EKF1003V	Do Not Install		
14	1	R3	100k	603	Panasonic	ERJ-3EKF1003V			P100KHCT-ND
15	1	R4	0	603	Panasonic	ERJ-3GEY0R00V			P0.0GCT-ND
16	2	R5,R6	150	603	Panasonic	ERJ-3EKF1500V			P150HCT-ND
17	1	R8	39.2k	603	Panasonic	ERJ-3EKF3922V			P39.2KHCT-ND
18	1	R9	61.9k	603	Panasonic	ERJ-3EKF6192V			P61.9KHCT-ND
19	2	R11,R10	2.21k	603	Panasonic	ERJ-3EKF2211V			P2.21KHCT-ND
20	6	R12,R13,R14,R15,R16,R17	1.0M	603	Panasonic	ERJ-3EKF1004V			P1.00MHCT-ND

21	1	R18	10.0k	603	Panasonic	ERJ-3EKF1002V			P10.0KHCT-ND
22	1	R19	100	603	Panasonic	ERJ-3EKF1000V			P100HCT-ND
23	1	R20	DNI	603	Panasonic	ERJ-3EKF1000V	Do Not Install		
24	10	TP1,TP2,TP3,TP4,TP5,TP6,	TEST POINT	Keystone-5015	Keystone	5015			5015KCT-ND
		TP7,TP8,TP9,TP10							
25	1	U1	SN74LVC138APW	16-TSSOP	Texas Instruments	SN74LVC138APW		LC138A	296-1223-1-ND
26	1	U2	SN74LVC32APW	14-TSSOP	Texas Instruments	SN74LVC32APW		LC32A	296-1236-1-ND
27	2	U3,U16	TC7S04F	SSOP5-P-0.95	Toshiba	TC7S04F		E5	TC7S04FTCT-ND
28	1	U4	TC7W08FK	SSOP8-P-0.50A	Toshiba	TC7W08FK		W,08	TC7W08FKCT-ND
29	2	U5,U15	TC7W32FK	SSOP8-P-0.50A	Toshiba	TC7W32FK		W,32	TC7W32FKCT-ND
30	1	U6	TC7W74FK	SSOP8-P-0.50A	Toshiba	TC7W74FK		W,74	TC7W74FKCT-ND
31	1	U7	NC7SZ05M5	SOT-23-5	Fairchild	NC7SZ05M5		7Z05	NC7SZ05M5CT-ND
32	1	U8	TPS3809K33DBV	SOT-23	Texas Instruments	TPS3809K33DBV		PDBI	296-2631-1-ND
33	2	U11,U9	74VHC273MTC	20-TSSOP	Fairchild	74VHC273MTC	Mouser, America II		
34	4	U10,U12,U13,U14	TC74LCX573FT	20-TSSOP	Toshiba	TC74LCX573FT			TC74LCX573FTCT-ND
35	1	U17	TC7S32F	SOT-23-5	Toshiba	TC7S32F		E4	TC7S32FCT-ND
36	1	U18	TC7W04FK	SSOP8-P-0.50A	Toshiba	TC7W04FK		W,04	TC7W04FKCT-ND
37	1	U19	CY7C68001-56PVC	SSOP-56	Cypress	CY7C68001-56PVC	Future/Active, America II		
38	1	U20	TC7S08F	SSOP5-P-0.95	Toshiba	TC7S08F		E2	TC7S08FCT-ND
39	1	Y1	CS10-24.000MABJ	CS10	Citizen	CS10-24.000MABJ			300-8103-1ND

```
/* gpio.c:
 *
 * Low-level support for the Micromodem GPIO v1.0 board.
 * 3 Dec 2003, Jim Partan.
 * 9 Dec 2003, Jim Partan: check wrappers now call ERROR_routine()
 * 10 Dec 2003, Jim Partan: changed RX_SYNC to EXT_SYNC, USB_5V to USB_VBUS
 * $Id$
 *
 * Example code for:
 * Low-Level Register Access;
 * Wrapper Functions for the following registers:
 *    GPIO Control, GPIO Status, USB Control, USB Status, CHECK;
 * GPIO /EXTINT Interrupt Service Routine.
 *
 */

#define GPIO_CREG_IACK        (0x01)
#define GPIO_CREG_TXACTIVE    (0x02)
#define GPIO_CREG_RXACTIVE    (0x04)
#define GPIO_CREG_RXPENDING   (0x08)
#define GPIO_CREG_GPO1_OFFSET (4)

#define GPIO_SREG_EXTPPS      (0x01)
#define GPIO_SREG_USBINT      (0x02)
#define GPIO_SREG_EXTSYNC     (0x04)
#define GPIO_SREG_TXINHIBIT   (0x08)
#define GPIO_SREG_GPI1_OFFSET (4)

#define USB_CREG_AMASK        (0x0f8)
#define USB_CREG_WAKEUP       (0x08)
#define USB_CREG_PKTEND       (0x10)
#define USB_CREG_EXTPPSINT    (0x20)
#define USB_CREG_EXTSYNCINT   (0x40)
#define USB_CREG_CHECK        (0x80)

#define USB_SREG_EXTPPS       (GPIO_SREG_EXTPPS)
#define USB_SREG_USBINT       (GPIO_SREG_USBINT)
#define USB_SREG_EXTSYNC      (GPIO_SREG_EXTSYNC)
#define USB_SREG_READY        (0x08)
#define USB_SREG_FLAGA        (0x10)
#define USB_SREG_FLAGB        (0x20)
#define USB_SREG_FLAGC        (0x40)
#define USB_SREG_VBUS         (0x80)


ioport int port2000;    // flash board: data
ioport int port2001;    // flash board: address
ioport int port2002;    // flash board: command
ioport int port2003;    // GPIO board: check registers (read-only)
ioport int port2004;    // flash board: chip-Select Latch Enable (SLE)
ioport int port2005;    // GPIO board: GPIO registers
ioport int port2006;    // GPIO board: USB control and status registers
ioport int port2007;    // GPIO board: USB chip


#define CHECK_REG (port2003)
#define GPIO_REG  (port2005)
```

```
#define USB_REG         (port2006)
#define USB_SX2         (port2007)


static int gpio_creg = 0x00;  // these registers hardware-reset to zero.
static int usb_creg  = 0x00;



/*************************************************************************/
/********************** Low-Level Register Access ************************/
/*************************************************************************/

// just write the 8-bit register, having already masked and shifted.
void GPIO_output_low(int word)
{
  GPIO_REG = word;
  gpio_creg = (word & 0x0ff); // keep a mirror
}


// just read the 8-bit register -- mask and shift individual bits later.
int GPIO_input_low(void)
{
  return (GPIO_REG & 0x0ff);
}


// just write the 8-bit register, having already masked and shifted.
void USB_output_low(int word)
{
  USB_REG = word;
  usb_creg = (word & 0x0ff);  // keep a mirror
}


// just read the 8-bit register -- mask and shift individual bits later.
int USB_input_low(void)
{
  return (USB_REG & 0x0ff);
}


// just read the 8-bit register -- mask and shift individual bits later.
int CHECK_input_low(void)
{
  return (CHECK_REG & 0x0ff);
}


// access to USB_SX2 memory-mapped location is similar, using USB_SX2.
```

```
/*****************************************************************************/
/******************* GPIO Control Register Wrappers **************************/
/*****************************************************************************/



/* assert /IACK (active-low) for a few cycles, if it is not already asserted.
 * This will clear the /EXTINT flip-flop on the GPIO board.
 */
void GPIO_iack(void)
{
  GPIO_REG = (gpio_creg & ~GPIO_CREG_IACK);
  asm("      nop");
  asm("      nop");
  asm("      nop");
  asm("      nop");
  GPIO_REG = gpio_creg;
}


// enable the GPIO /EXTINT interrupt by releasing /IACK.
void GPIO_extint(int enable)
{
  if (enable)
      GPIO_output_low(gpio_creg |  GPIO_CREG_IACK);
  else
      GPIO_output_low(gpio_creg & ~GPIO_CREG_IACK);

}


// set or reset the TX_ACTIVE line, lighting the led.
void GPIO_tx_active(int active)
{
  if (active)
      GPIO_output_low(gpio_creg |  GPIO_CREG_TXACTIVE);
  else
      GPIO_output_low(gpio_creg & ~GPIO_CREG_TXACTIVE);
}


// set or reset the RX_ACTIVE line, lighting the led.
void GPIO_rx_active(int active)
{
  if (active)
      GPIO_output_low(gpio_creg |  GPIO_CREG_RXACTIVE);
  else
      GPIO_output_low(gpio_creg & ~GPIO_CREG_RXACTIVE);
}
```

```
// set or reset the RX_PENDING line.
void GPIO_rx_pending(int active)
{
  if (active)
      GPIO_output_low(gpio_creg |  GPIO_CREG_RXPENDING);
  else
      GPIO_output_low(gpio_creg & ~GPIO_CREG_RXPENDING);
}

// set general-purpose output line N to value b.
void GPIO_gpoutput(int N, int b)
{
  int tmp = 0x01 << ((N-1)+GPIO_CREG_GPO1_OFFSET);
  if (b)
      GPIO_output_low(gpio_creg |  tmp);
  else
      GPIO_output_low(gpio_creg & ~tmp);
}


/*************************************************************************/
/****************** GPIO Status Register Wrappers ************************/
/*************************************************************************/


// polled access to EXTPPS (through GPIO status register)
int GPIO_extpps(void)
{
  return (GPIO_input_low() & GPIO_SREG_EXTPPS);
}


// polled access to USB_INT (through GPIO status register)
int GPIO_usbint(void)
{
  return (GPIO_input_low() & GPIO_SREG_USBINT);
}


// polled access to EXT_SYNC (through GPIO status register)
int GPIO_extsync(void)
{
  return (GPIO_input_low() & GPIO_SREG_EXTSYNC);
}


// poll TX_INHIBIT
int GPIO_tx_inhibit(void)
{
  return (GPIO_input_low() & GPIO_SREG_TXINHIBIT);
}
```

```
// read general-purpose input line N.
int GPIO_gpinput(int N)
{
  int mask = 0x01 << ((N-1)+GPIO_SREG_GPI1_OFFSET);
  return (GPIO_input_low() & mask);
}


/*************************************************************************/
/****************** USB Control Register Wrappers ************************/
/*************************************************************************/

// set the USB_Ax lines (USB_A0, USB_A1, USB_A2).
void USB_address(int address)
{
  USB_output_low((usb_creg & USB_CREG_AMASK) | (address & ~USB_CREG_AMASK));
}


// set or reset the USB_WAKEUP line.
void USB_wakeup(int wakeup)
{
  if (wakeup)
      USB_output_low(usb_creg |  USB_CREG_WAKEUP);
  else
      USB_output_low(usb_creg & ~USB_CREG_WAKEUP);
}


// set or reset the USB_PKTEND line.
void USB_pktend(int pktend)
{
  if (pktend)
      USB_output_low(usb_creg |  USB_CREG_PKTEND);
  else
      USB_output_low(usb_creg & ~USB_CREG_PKTEND);
}


// enable or disable the EXTPPS interrupt.
void GPIO_extpps_int(int enable)
{
  if (enable)
      USB_output_low(usb_creg |  USB_CREG_EXTPPSINT);
  else
      USB_output_low(usb_creg & ~USB_CREG_EXTPPSINT);
}


// enable or disable the EXT_SYNC interrupt.
void GPIO_extsync_int(int enable)
{
  if (enable)
      USB_output_low(usb_creg |  USB_CREG_EXTSYNCINT);
  else
      USB_output_low(usb_creg & ~USB_CREG_EXTSYNCINT);
}
```

```
// set or reset the USB_CHECK line.
void USB_check(int set)
{
  if (set)
      USB_output_low(usb_creg |  USB_CREG_CHECK);
  else
      USB_output_low(usb_creg & ~USB_CREG_CHECK);
}



/*************************************************************************/
/****************** USB Status Register Wrappers *************************/
/*************************************************************************/


// polled access to EXTPPS (through USB status register)
int USB_extpps(void)
{
  return (USB_input_low() & USB_SREG_EXTPPS);
}


// polled access to USB_INT (through USB status register)
int USB_usbint(void)
{
  return (USB_input_low() & USB_SREG_USBINT);
}


// polled access to EXT_SYNC (through USB status register)
int USB_extsync(void)
{
  return (USB_input_low() & USB_SREG_EXTSYNC);
}


// poll USB_READY
int USB_ready(void)
{
  return (USB_input_low() & USB_SREG_READY);
}


// poll USB_FLAGA
int USB_flaga(void)
{
  return (USB_input_low() & USB_SREG_FLAGA);
}


// poll USB_FLAGB
int USB_flagb(void)
{
  return (USB_input_low() & USB_SREG_FLAGB);
}
```

```
// poll USB_FLAGC
int USB_flagc(void)
{
  return (USB_input_low() & USB_SREG_FLAGC);
}



// poll USB_VBUS
int USB_vbus_valid(void)
{
  return (USB_input_low() & USB_SREG_VBUS);
}


/*************************************************************************/
/********************* Check Register Wrappers ***************************/
/*************************************************************************/


int GPIO_creg_check(void)
{
  int usb_old = usb_creg;
  int gpio_read;

  USB_check(0);
  gpio_read = CHECK_input_low();
  USB_output_low(usb_old);    // restore it

  if (gpio_read != gpio_creg) {
      ERROR_routine("GPIO CREG corrupted", gpio_read);
      return 1;
  }

  return 0;
}


int USB_creg_check(void)
{
  int usb_old = usb_creg;
  int usb_read;
  int ret = 0;

  USB_check(1);
  usb_read = CHECK_input_low();
  if (usb_read != usb_creg) {
      ERROR_routine("USB CREG corrupted", usb_read);
      ret = 1;
  }
  USB_output_low(usb_old);    // restore it
  return ret;
}
```

```
/*****************************************************************************/
/***************** GPIO /EXTINT Interrupt Service Routine ********************/
/*****************************************************************************/

interrupt void GPIO_isr(void)
{
  int source = GPIO_input_low();

  if (source & GPIO_SREG_EXTPPS) {
      // do EXTPPS service here
  }

  if (source & GPIO_SREG_EXTSYNC) {
      // do EXT_SYNC service here
  }

  if (source & GPIO_SREG_USBINT) {
      // do USBINT service here
  }

  GPIO_iack();
  return;
}


/*****************************************************************************/
/*************************** end of file gpio.c ******************************/
/*****************************************************************************/
```

Application-Level Software Notes

Jim Partan

Document Revision: Draft D
10 Dec 2003

These notes describe suggested uses of the GPIO board at the application level. The lines are TX_ACTIVE, RX_ACTIVE, RX_PENDING, EXTPPS, EXT_SYNC, TX_INHIBIT, and GPO1-GPO4, GPI1-GPI4. Several items are yet to-be-determined, labeled ***[DRAFT]***.

TX_ACTIVE:

At start of transmit, set TX_ACTIVE high:

```
GPIO_tx_active(1);
```

When finished transmitting, set TX_ACTIVE low:

```
GPIO_tx_active(0);
```

RX_ACTIVE:

When a packet has been detected by MFD, set RX_ACTIVE high:

```
GPIO_rx_active(1);
```

When finished decoding, set RX_ACTIVE low:

```
GPIO_rx_active(0);
```

RX_PENDING:

When a cycle-init has been receiving indicating another packet is coming (e.g. downlink), set RX_PENDING high:

```
GPIO_rx_pending(1);
```

When the packet has timed out or when MFD detects the start of the packet, set RX_PENDING low:

```
GPIO_rx_pending(0);
```

Note: The order of operations with RX_ACTIVE and RX_PENDING is critical, as these are control lines for other systems. The correct order of operations for transitions from pending to active, and vice versa, is specified below. Transitions from inactive to pending, or active to inactive, can be done with simple `GPIO_rx_pending(1);` or `GPIO_rx_active(0);` respectively.

• When going from the pending state to the active state:

```
GPIO_rx_active(1); // set active high, THEN set pending low!
GPIO_rx_pending(0);
```

• When going from the active state to the pending state (e.g. multi-frame packets):

```
GPIO_rx_pending(1); // set pending high, THEN set active low!
GPIO_rx_active(0);
```

EXTPPS:
Generally used as an external Pulse-Per-Second input, causing an /EXTINT interrupt:

In initialization code:

```
time_t ticks;
long start_of_second_samples;
int okay_to_tx;

GPIO_extint(1);           // allow GPIO interrupts
GPIO_extppsint(1);        // enable the EXTPPS interrupt
ticks = mktime(&kdt);     // initialize seconds counter
```

EXTPPS interrupt service routine:

```
interrupt void extpps_isr(void)
{
  // make sure we're really an EXTPPS /EXTINT interrupt!!
  if ((GPIO_input_low() & GPIO_SREG_EXTPPS)==0)
     return;
  ticks++;
  start_of_second_samples = AIN_getsamp();
  okay_to_tx = 1;
  GPIO_iack();
}
```

Then, if we're trying to transmit on the rising edge of EXTPPS, we should be stalling in TXPUT_run():

```
extern volatile int okay_to_tx;
okay_to_tx = 0;
while (!okay_to_tx && not_timed_out)
     ; // wait for EXTPPS
// now tx, or handle timeout
```

[DRAFT] The host computer will need to specify that the external PPS is to be used instead of the internal RTC PPS, perhaps with a new $CCCFG parameter:

$CCCFG,PPS,0	(internal RTC PPS, the default)
$CCCFG,PPS,1	(external PPS)

EXT_SYNC:
Generally used as an external sync line to signal a transmission, as an alternative to acoustic detection. This might be useful if we are in the same unit as the transmitting system -- then (a) we might be able to get a digital line to detect from, instead of running MFD, and (b) MFD might have problems anyway, if the transducers are right next to each other in their near fields, etc. EXT_SYNC can also be used for other forms of external synchronization.

EXT_SYNC can be polled, or can cause an /EXTINT interrupt:

Polled:

```
while ( !GPIO_extsync() && not_timed_out )
     ; // wait for sync line
// handle sync event, e.g. set MFD location, and MFD flag for
// hardware sync detection
```

Interrupt-driven:

```
GPIO_extint(1);          // enable GPIO /EXTINT interrupts
GPIO_extsyncint(1);      // enable EXT_SYNC interrupts
// initialize MFD

interrupt void extsync_isr(void)
{
  // make sure we're really an EXT_SYNC /EXTINT interrupt!!
  if ((GPIO_input_low() & GPIO_SREG_EXTSYNC)==0)
     return;
  // set mfd location and flag for hardware sync detection
  // the block after mfd needs to check for mfd hardware
  // sync detection next time it runs.

  GPIO_iack();
}
```

TX_INHIBIT:

Before transmitting, for instance in TXPUT_run(), check TX_INHIBIT:

```
while ( GPIO_tx_inhibit() && not_timed_out )
     ; // wait til we can transmit.
```

GPIO:

Use as appropriate for the application:

```
GPIO_gpoutput(N, b);     // N: 1-4, b: 0,1
val = GPIO_gpinput(N);   // N: 1-4, val: zero or non-zero.
```

[DRAFT] Perhaps the GPIO can use the $CCMEC command to set/read pins via an acoustic or serial command. The syntax is $CCMEC,SRC,DEST,LINE,MODE,ARG*CS. We have 8 bits to split between Line, Mode, and Arg. Here's a proposed redistribution of the bits (at the expense of Arg) to allow $CCMEC to control all GPIO lines:

Lines (3 bits):		Modes (2 bits):		Arg (3 bits):		
EXTSEL1	000	Read:	00	time:	000	1ms
EXTSEL2	001	Write:	01		001	0.5sec
GP1	010	Toggle High:	10		010	1sec
GP2	011	Toggle Low:	11		011	2sec
GP3	100				100	5sec
GP4	101				101	10sec
Reserved	110				110	20sec
Reserved	111				111	30sec

Document Revision History:

Revision Draft D, 10 Dec 2003, Jim Partan: added Draft sections on $CCCFG,PPS and on $CCMEC.

Revision C, 10 Dec 2003, Jim Partan: changed RX_SYNC to EXT_SYNC.

Revision B, 9 Dec 2003, Jim Partan: specified order of operations for RX_PENDING, RX_ACTIVE transition.

Revision A: 3 Dec 2003, Jim Partan: initial release.

Producing a batch of Micromodem-GPIO-v1.0's
Jim Partan, WHOI, 24 Dec 2003.
Document Revision: C

1.) Get quote from vendor(s) for fabrication and assembly by supplying the following files:
    a.) Micromodem_GPIO_V10_rules.pdf (Fabrication and Assembly rules)
    b.) Micromodem_GPIO_V10_gerbers.zip (Gerber files for board)
    c.) Micromodem_GPIO_V10_assembly.zip (stencil files, pick-and-place files)
    d.) Micromodem_GPIO_V10_BOM.xls (Bill of Materials, Excel format)
    e.) Micromodem_GPIO_V10_IPCnetlist.IPC (IPC-356 netlist to test fabrication)

2.) For complete turnkey operation (fabrication, assembly, and parts procurement), the vendor should be ready to go with the files above. Otherwise, purchase the parts from the Bill of Materials.

3.) Vendors we (WHOI) have used include:

    Phoenix Resources, Rick Costello, rc@phoenixpwb.com, 603-863-9096
    193 Lempster Coach Road, Goshen, NH 03752, Fax: 603-863-9099
    www.phoenixpwb.com [Will not do parts procurement for small quantities.]

    Nexlogic Technologies, Justin Proulx, justin@nexlogic.com, 408-436-8150 x112
    345 East Brokaw Road, San Jose, CA 95112, Fax: 408-436-8156
    www.nexlogic.com

4.) For modified designs, redesign schematics in Orcad Capture, produce new netlist and bill of materials, contract for new layout design (contact Rick Costello, listed above -- Ben Hunter was the layout designer). See Micromodem_GPIO_V10_Layout_rules.pdf for layout rules.

WHOI Micromodem-GPIO-v1.0 Fabrication and Assembly Rules

Document Revision: E
12 Dec 2003

Jim Partan
Phone: 508-289-3534
Fax: 508-457-2195
Email: jpartan@whoi.edu
Mail: WHOI MS-18, Woods Hole, MA, 02543

Fabrication Rules:

1.) Board material is to be FR4, overall thickness not to exceed 0.062" (62 mils).
2.) Board dimensions shall be 4.35" x 1.50".
3.) There shall be a 1.0” x 0.25” notch cut approximately in the middle of the lower board edge. Exact position and size as indicated by the Gerber files.
4.) Board edges shall be finished, not V-scored.
5.) Outer and inner layer copper weight shall be 1 ounce.
6.) Soldermask (solder resist) shall be on top and bottom layers.
7.) Silkscreen shall be on top and bottom layers.
8.) Surface-mount components are on top and bottom layers.
9.) Through-hole components are on the top layer only.
10.) Traces USB_DPLUS (JP6.3 to U19.15) and USB_DMINUS (JP6.2 to U19.16) shall have their differential impedance controlled to 90-Ohms +/- 10%.
11.) "Bed of Nails"-style testing, or similar, shall be performed.

Assembly Rules:

1.) WHOI shall supply all components.
2.) Do not install:
    a.) Item 13 (references R1, R2, R7)
    b.) Item 23 (reference R20)
3.) Finished boards are to be given a unique serial number.
4.) Finished boards are to be given a visual Quality Assurance inspection.
5.) For this initial board run, two boards shall be fully populated and shipped to WHOI for testing. No components at all shall be installed on the remaining boards until WHOI has tested and verified hardware design. At this point there are two possible cases (please quote both cases):
    a.) Full Run: Upon WHOI approval, all remaining boards shall be fully populated.
    b.) Terminated Run: If the hardware design fails testing, WHOI may decide to terminate the assembly run after the first two boards. In this case, the blank boards and uninstalled components shall be shipped to WHOI.
6.) Unused components shall be returned to WHOI at above address.

Layout Design Rules for WHOI Micromodem GPIO v1.0 PCB:
Document revision: C, 24 Dec 2003

Jim Partan
Voice: 508-289-3534
Fax: 508-457-2195
Email: jpartan@whoi.edu
Mail: WHOI MS-18, Woods Hole, MA 02543

Mechanicals (please see attached Mechanical Drawing)
- 4.35" x 1.5" overall dimension.
- Notch on lower board edge, 1.0" x 0.25", as indicated in Mechanical Drawing (drawing labeled "GPIO Board").
- Finished board thickness not to exceed 0.062" (62 mils). Board material intended to be FR-4, 1 ounce copper.
- SMT components on top and bottom layers. Through-hole components on top layer only.
- Soldermask (solder resist) top and bottom.
- Silkscreen top and bottom.
- 4 unplated mounting holes 0.115" diameter, placed as shown on Mechanical Drawing, to include a 0.3" placement and routing keep-out (both top and bottom sides).

Board Stack-Up
- 4 layers as follows:
  - layer 1 signal (called "Top" in this document)
  - layer 2 ground (GND)
  - layer 3 power (VCC)
  - layer 4 signal (called "Bottom" in this document)

Deliverables
- Negatives for each copper layer
- Top and Bottom Soldermask (solder resist) drawings
- Top and Bottom Silkscreen drawings
- Top and Bottom Solder Paste (stencil) drawings
- Assembly / Fabrication drawings
- Component Locator (for pick and place assembly)
- Drill and Aperture files
- Drill Master
- Mechanical Drawing showing board dimensions, mouting hole locations, JP1,JP2,JP3,JP4 locations.
- IPC-356 Netlist
- Pads PCB database

Placement (Final Placement to be approved by WHOI prior to routing)
- Please refer to Mechanical Drawing and Placement Drawings for the placement of JP1-JP6. The four mounting holes and JP1, JP2, JP3, JP4 must align with WHOI Micromodem JP3, JP2. Please contact me if the Mechanical Drawing dimensions appear incorrect.
- Please refer to Placement Drawings 1 and 2 for APPROXIMATE SUGGESTED placement of U19, Y1, C31, R19, R20, and LED1. U19 and Y1 are to be on the top side of the board. LED1 location might be tight.
- 3mm=0.12" keep-out for components on all edges (except for JP5,JP6).
  [If any components must fail keep-out, start with ones which are easy to hand-solder.]
- Decoupling capacitors to be placed as close as possible to, and on the same side as, their associated IC whenever possible. Refer to schematic for associations. (Associate C19, C21-C27 with the various U19 power pins.)
- All test points (TP1-TP10) to be placed on top of board.
- Resistor networks RN1-RN5 can have their resistors swapped if it eases routing. Please suggest pins to swap and request a new netlist if this is desired.

Routing

- Planes shall not extend past 3.1" from the left edge of the board (see Placement Drawings). All components except JP3, JP4 shall be over the planes. All signal traces except those associated with JP3, JP4 shall be over the planes. [Note: in actual layout, two short traces were run along edge of VCC plane to keep board at 4 layers.]
- 0.05" keep-out for traces and planes on all edges.
- Minimum trace width is 0.005".
- Minimum trace separation is 0.005".
- Minimum trace-pad separation is 0.008".
- No internal/hidden vias.
- Power and ground traces to be at least 0.010" width.

Special Routes

- USB Traces USB_DPLUS and USB_DMINUS:
    - shall have their differential impedance controlled to 90 Ohms +/- 10%.
    - shall not have vias. [One via in USB_DMINUS was needed.]
    - shall be kept to within 1.25mm (50 mils) of each other in length,
    - have a preferred length of 20-30mm (1 inch). The absolute maximum length is 75mm (3 inches), and should be shorter than that.
    - shall not have right-angle bends.
    - should be isolated from other non-static traces by at least 6.5mm, and ideally 10mm, when and if possible.
    - should have constant-distance spacing between these traces.
    - Test Points TP9 and TP10 shall not have stubs, but shall be directly in contact with the USB_DPLUS and USB_DMINUS traces, respectively. If including TP9 and TP10 prevents the controlled impedance specification from being maintained, do not include them, and I will generate a new netlist without them.
    - Refer to Cypress "High-Speed USB PCB Layout Recommendations" Application Note and Cypress CY7C68001 datasheet (p.49 in particular) if more information is desired. Please consult with me if you have questions or suggestions to improve layout or performance.
- "USB Peninsula":
    - Make a cut in the Vcc and GND planes around the USB connector (JP6) leaving a 200-mil (5mm) opening for D+ and D-, to preserve their differential impedance. See Placement Drawings.
    - Route USB_DPLUS, USB_DMINUS on the top side of the board. Route USB_VBUS on the bottom side of the board.
    - Place R19 within the USB ground plane cut on the bottom side of the board. Place R20 outside of the USB ground plane cut on the bottom side of the board. Connect trace USB_SHIELD to the the mounting/case pins for JP6 (un-numbered pins in Molex's diagram, numbered pins 6 and 7 in our netlist), routing on the bottom side of the board.
    - Place C31 within the USB ground plane cut on the either side of the board, but placed so it does not interfere with USB_DPLUS and USB_DMINUS.
    - R8, R9, C30 are NOT critical placements, but if possible should be relatively close to JP6.
    - Refer to Cypress "High-Speed USB PCB Layout Recommendations" Application Note if more information is desired.
- Crystal Traces XO, XI:
    - shall not have vias.
    - shall be as short as possible.
- All nets associated with JP3, JP4:
    - All traces except /EXTINT to be at least 0.010" width, ideally 0.015" width.

Finishing

- Tear-drop all pad entries including vias.
- Layout silkscreen shall include a device designator for each part.
- Include orientation indicator for all IC's, connectors, polarized cap (C32), LED (LED1).
- At least three alignment targets on artwork.
- "uM GPIO v1.0, J.Partan, WHOI 12/03" to be placed on front of board in location visible when populated.

Miscellaneous Notes

JP6: Molex does not number the pins connected to the shield case. I have numbered them 6 and 7, connected to netname USB_SHIELD.

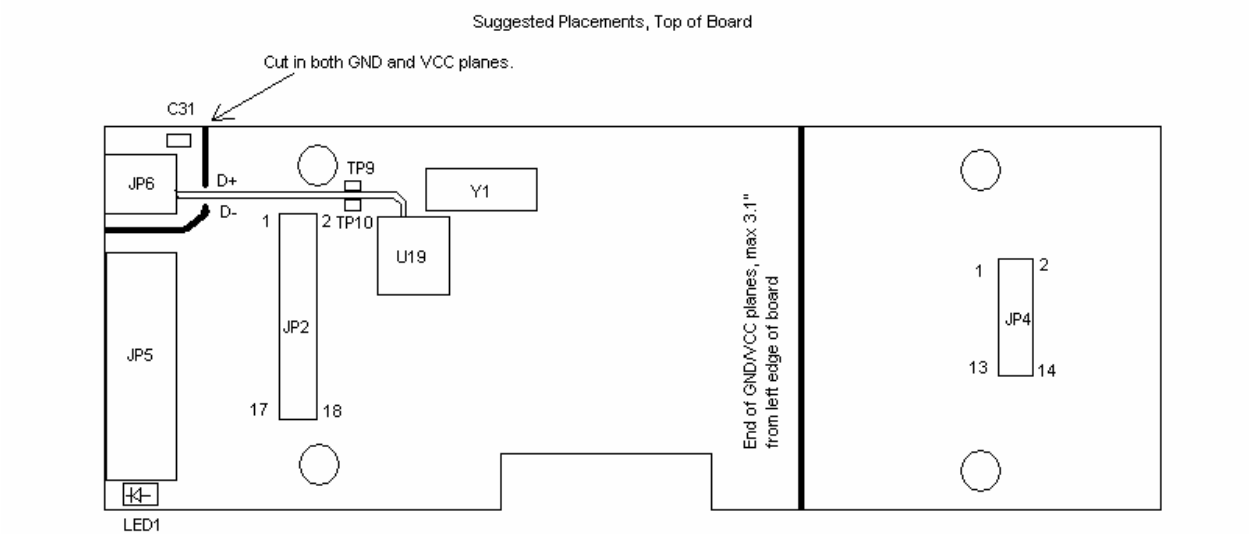
C32 (Tantalum Cap) - netlist/schematic pin 1 is POS
- netlist/schematic pin 2 is NEG

SML-LX1210 (LED1) - netlist/schematic pin 1 is ANODE LED A (Green)
- netlist/schematic pin 2 is ANODE LED B (Red)
- netlist/schematic pin 3 is CATHODE LED B (Red)
- netlist/schematic pin 4 is CATHODE LED A (Green)

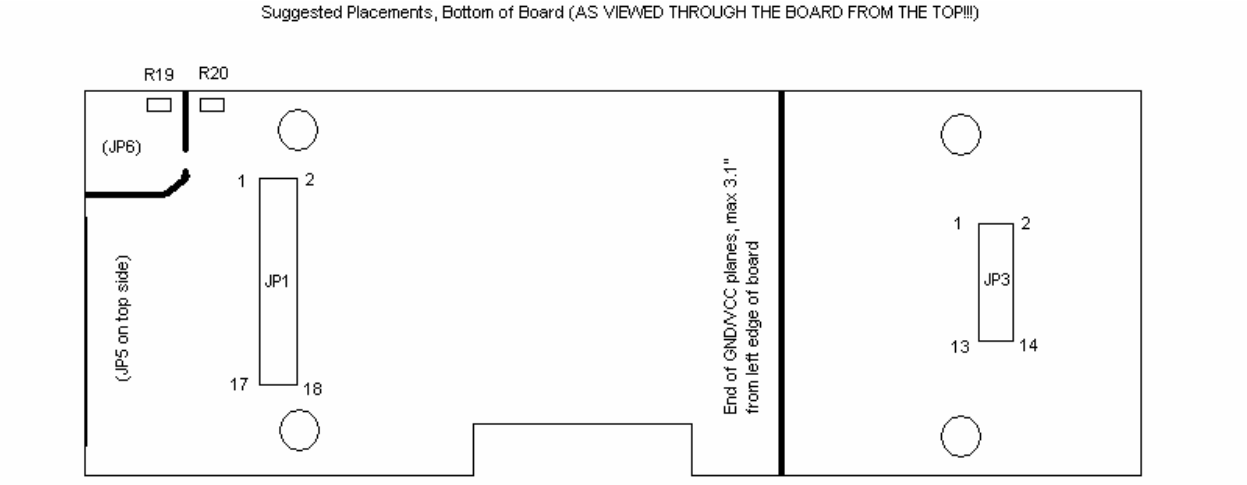
JP1 and JP3 layouts should NOT include holes for locating pegs if the pegs would hit JP2, JP4.

Placement Drawings

Placement Drawing 1 (Top of Board) (Note: U19 and Y1 packages not to scale):



Placement Drawing 2 (Bottom of Board):

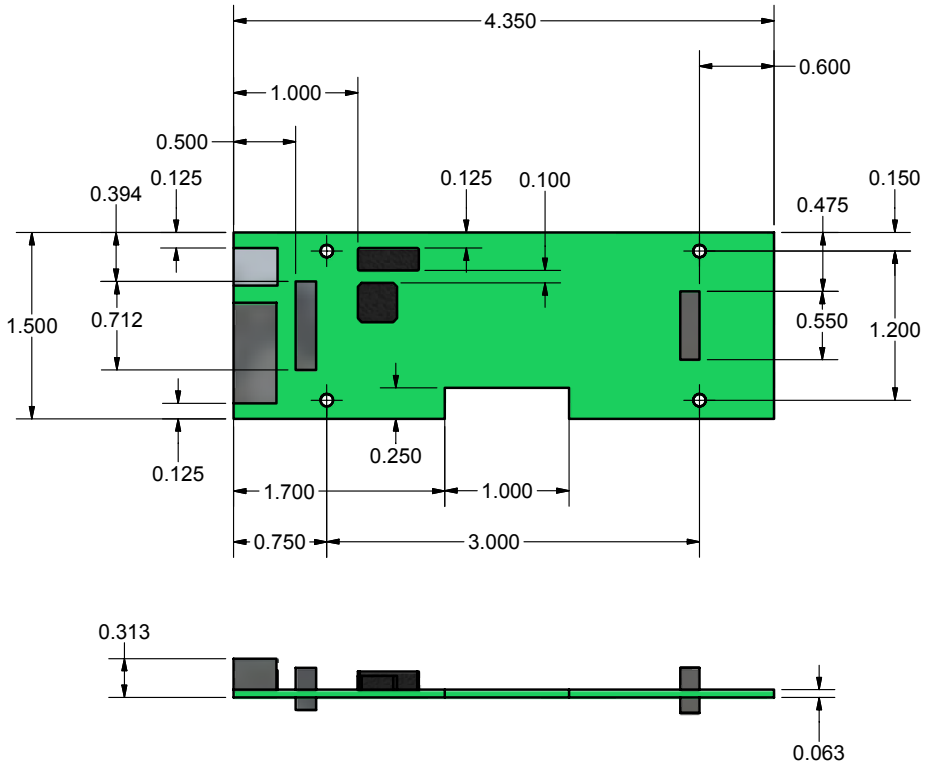


```
% diffimp.m: calculate differential impedance of traces USB_DPLUS, USB_DMINUS
% Our differential impedance target is 90 Ohms +/- 10%.
% The values in this file give a differential impedance of 91.5 Ohms.
% See Cypress Application Note, "High-Speed USB PCB Layout Recommendations"

W = 0.012;      % trace width in inches (12 mils)
H = 0.012;      % height (distance) of traces above ground plane in inches
                % (typically 12 mil for a 4-layer,62mil board, with
                % layer 1 traces, layer 2 ground plane).
T = 2*0.00065;  % trace thickness in inches (0.5oz copper=0.65mil)
S = 0.005;      % separation of differential traces in inches
er = 4.5;       % relative permitivity of FR4 substrate

W/H             % should be between 0.1 and 2.0
S/H             % should be between 0.2 and 3.0

Z0 = (87 / sqrt(er+1.41)) * log( (5.98*H) / (0.8*W + T) )
Zdiff = 2*Z0*(1 - 0.48*exp(-0.96*S/H))
```



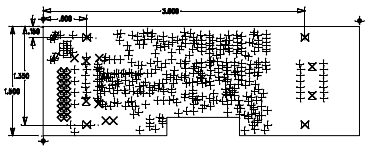
DRAWN K. Ball	12/3/2003	Woods Hole Oceanographic Institution
CHECKED		TITLE
QA		
MFG		
APPROVED		SIZE B / DWG NO GPIO Board / REV
		SCALE / SHEET 1 OF 1

FSCM	DWG NO.	SH	REV

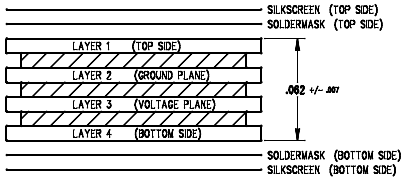
REVISIONS

ZONE	REV	DESCRIPTION	DATE	APPROVED

SIZE	QTY	SYM	PLTD
12	386	+	PLTD
25	5	X	PLTD
28	5	⊠	PLTD
30	18	◇	PLTD
40	4	⧖	NPLTD
115	4	⋈	NPLTD
28	4	Z	NPLTD
28 x 74.8	2	Z	NPLTD



LAYUP DETAIL A
4 LAYER



NOTES:

1. FABRICATE USING GERBER MASTERS MARKED uMODEM GPIO V1.0 DEC 18 2003.
2. FABRICATE AND TEST PER ANSI/IPC-RB-276 AND ANSI/IPC-A-600, CLASS 2. MATERIAL TO BE TYPE FR-4 PER MIL-P-13949/4. ALL LAYERS TO BE ONE OUNCE COPPER.
3. UNLESS OTHERWISE SPECIFIED, ALL HOLES AND CONDUCTORS SHALL BE COPPER PLATED.
4. MINIMUM CONDUCTOR WIDTH TO BE .005" (SPACING IS .005).
5. BOARD THICKNESS SHALL BE .062 +/- .007 INCLUDING ALL PLATINGS AND COATINGS, AND SHALL BE MEASURED ACROSS THE BOARD THICKNESS EXTREMES.
6. VENDOR CAGE CODE NUMBER, DATE AND/OR LOT IDENTIFICATION NUMBER SHALL BE MARKED ON PRINTED WIRING BOARD, LOCATION OPTIONAL.
7. SILKSCREEN COMPONENT OUTLINES AND REFERENCE DESIGNATORS ON BOTH SIDES USING CONTRASTING COLORED INK.
8. SOLDERMASK SHALL BE LIQUID PHOTOIMAGEABLE (LPI) (MATTE FINISH PREFERRED) APPLIED ON BOTH SIDES OVER BARE COPPER AND SHALL MEET SPEC IPC-SM-840 CLASS 3 (LATEST REV). ALL VIAS ARE TENTED.
9. SOLDERCOAT CONDUCTORS NOT COVERED BY SOLDERMASK; HASL.
10. LAYUP PER DETAIL A.
11. 12 MIL TRACES ON L1 TO HAVE DIFFERENTIAL IMPEDANCE CONTROLLED TO 90 OHMS +/- 10%

QTY REQD	ITEM NO.	PART OR IDENTIFYING NO.	NOMENCLATURE OR DESCRIPTION	MATERIAL SPECIFICATION

NEXT ASSY	USED ON

APPLICATION

UNLESS OTHERWISE SPECIFIED
ALL DIMENSIONS ARE IN INCHES
TOLERANCES ARE:
FRACTIONS ± DECIMALS .XX ± .XXX ANGLES ±

MATERIAL

FINISH

DO NOT SCALE DRAWING

CONTRACT NO.

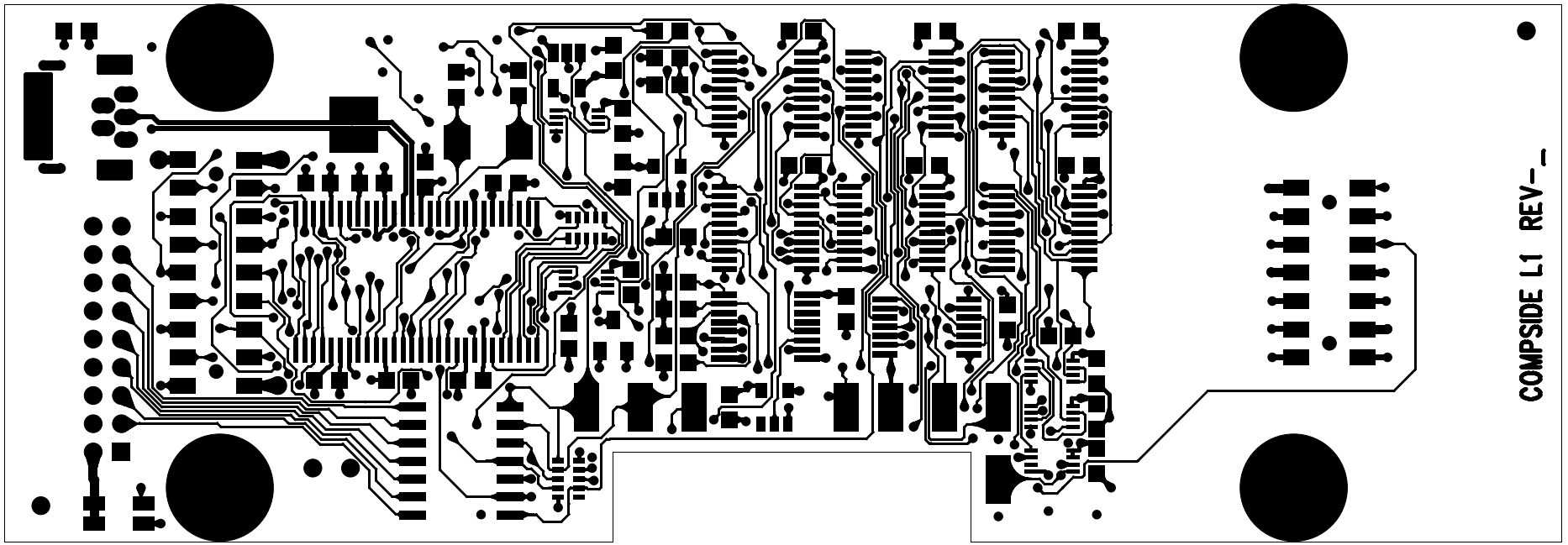
APPROVALS	DATE
DRAWN	
CHECKED	
ISSUED	

CUSTOMER
WOODS HOLE OCEANOGRAPHIC INSTITUTION
WOODS HOLE, MA 02543

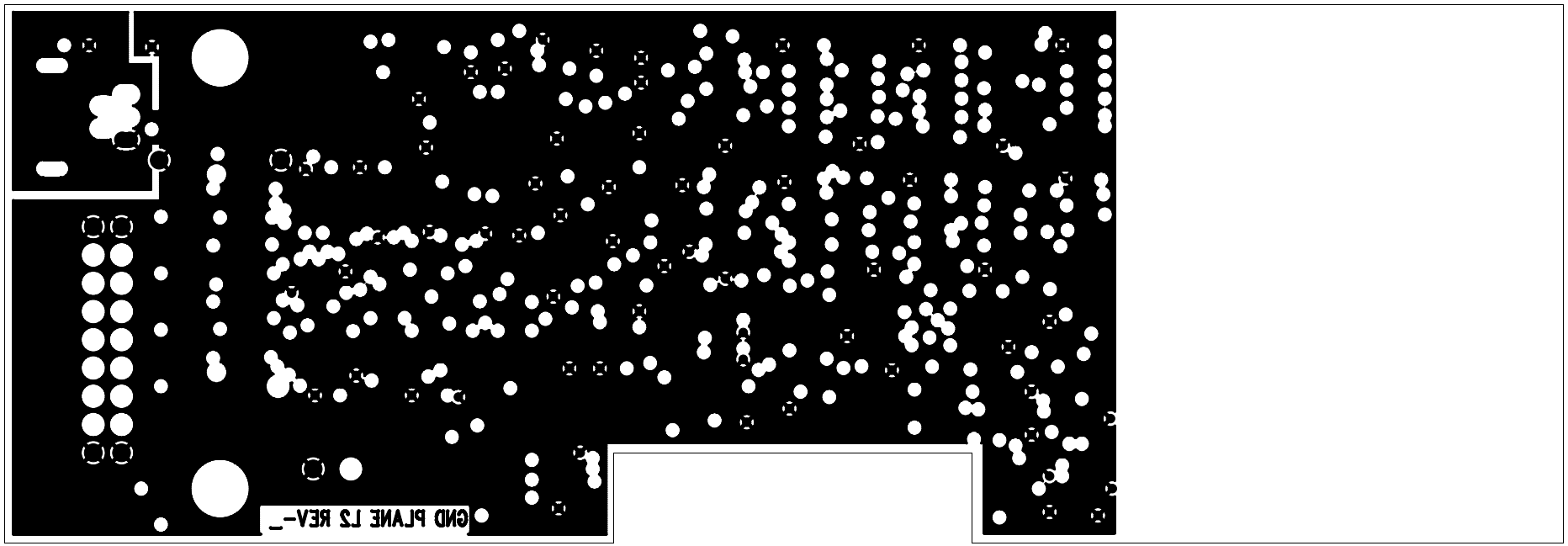
TITLE
FABRICATION DRAWING
MICROMODEM GPIO V1.0

SIZE	FSCM NO.	DWG NO.	REV.
C			-

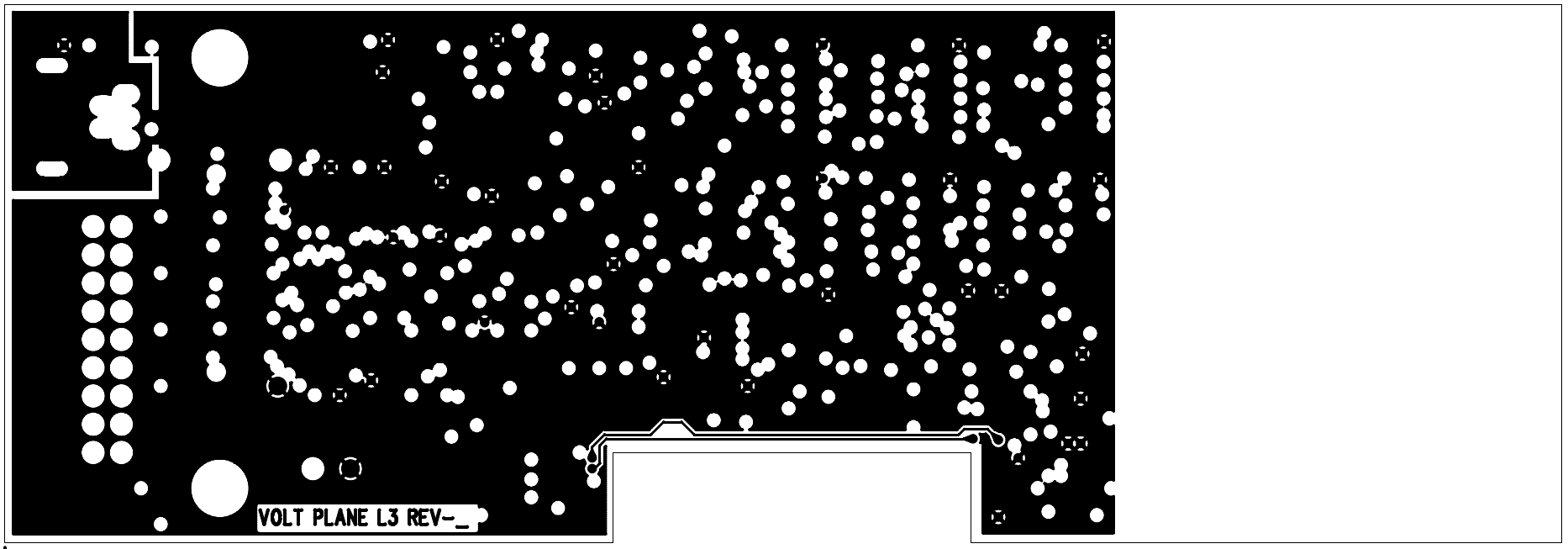
SCALE | SHEET OF



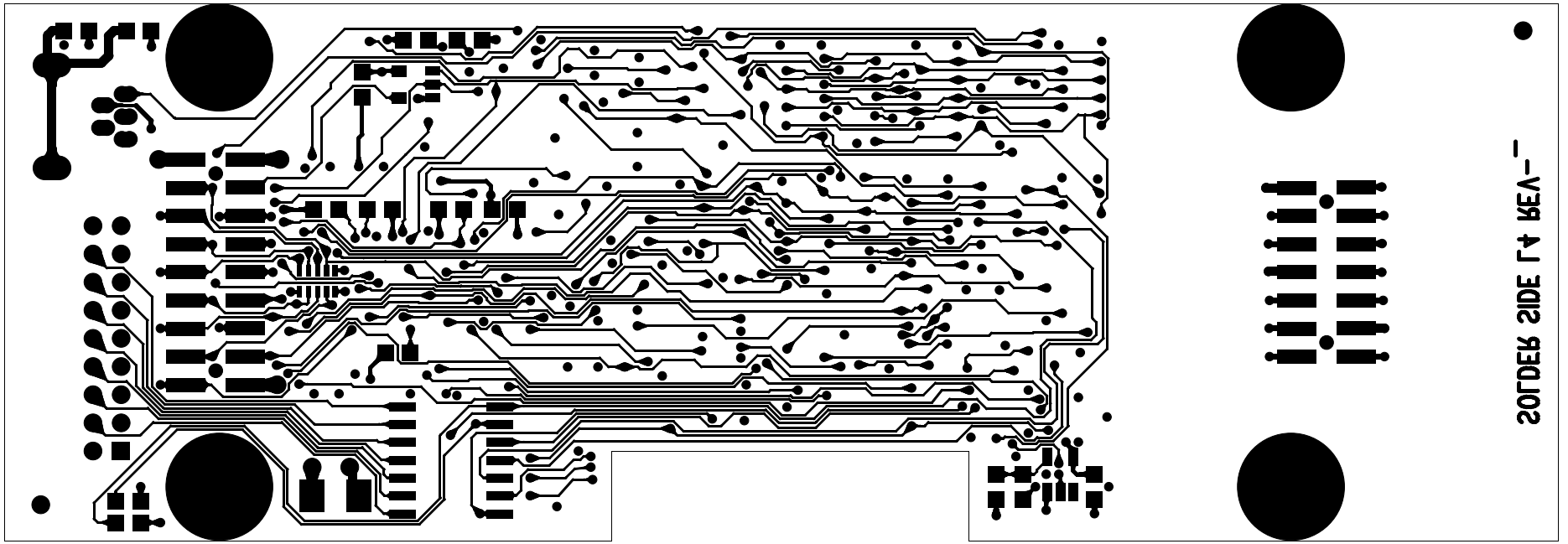
WOODS HOLE OCEANOGRAPHIC INSTITUTION
uMODEM GPIO V1.0
COMPSIDE L1



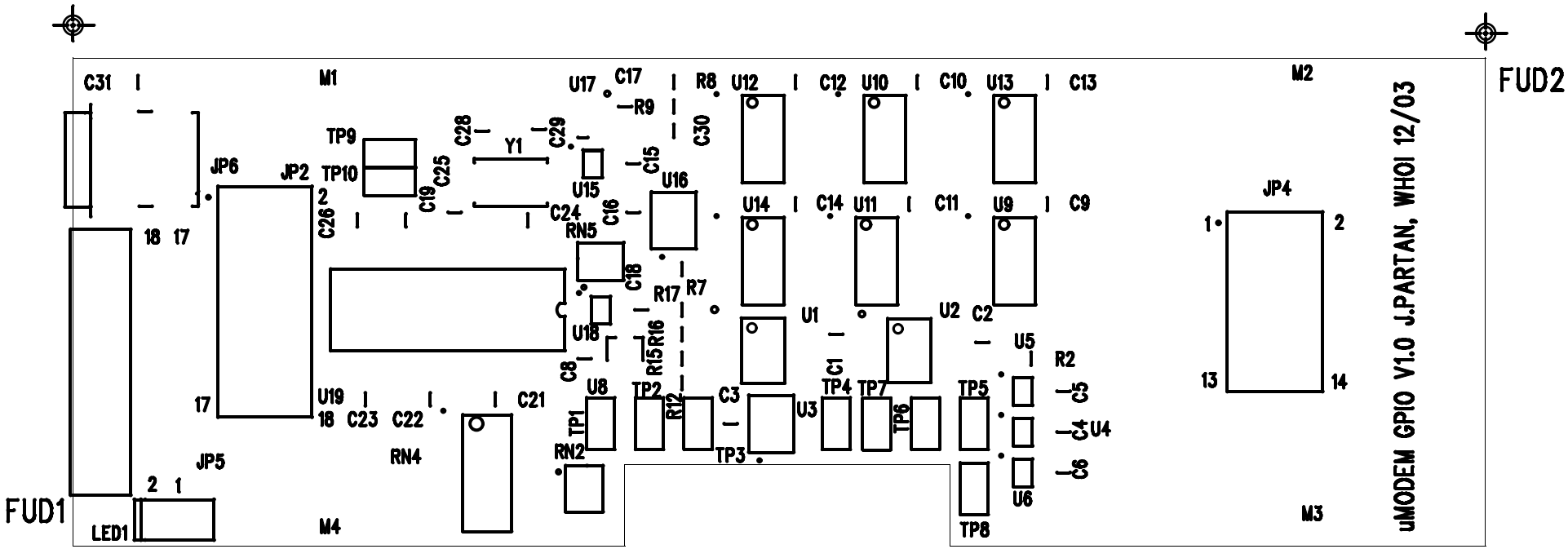
WOODS HOLE OCEANOGRAPHIC INSTITUTION
uMODEM GPIO V1.0
GND PLANE L2



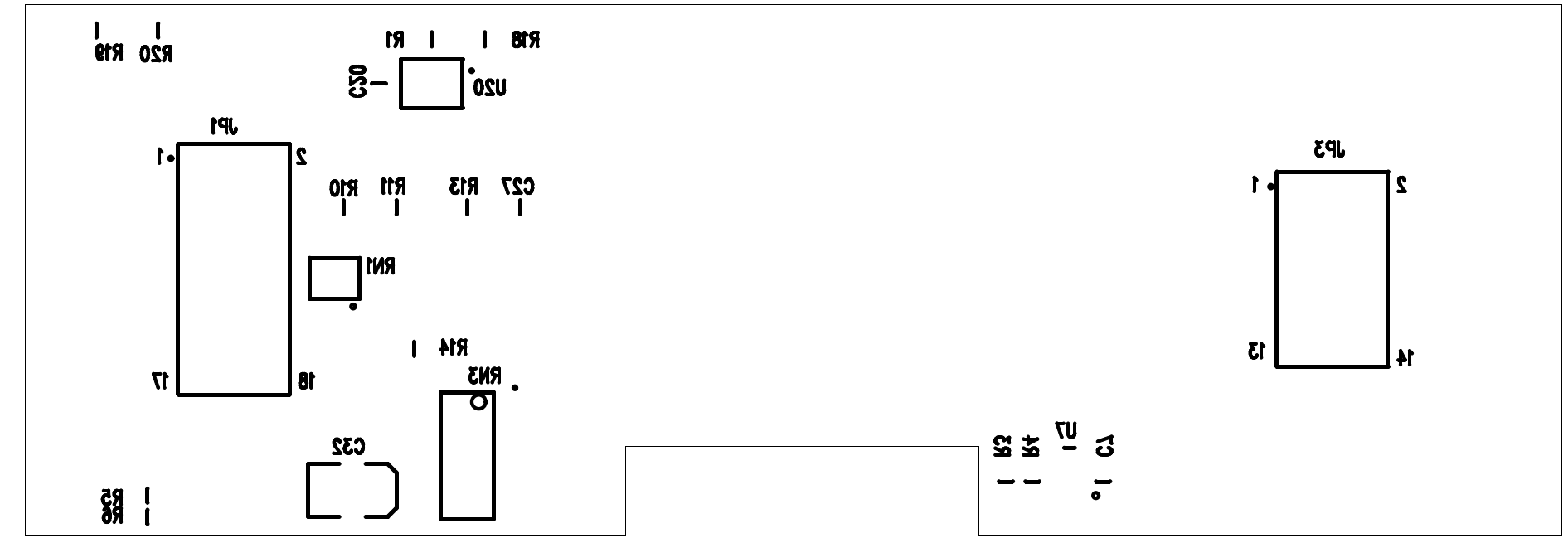
WOODS HOLE OCEANOGRAPHIC INSTITUTION
uMODEM GPIO V1.0
VOLTAGE PLANE L3



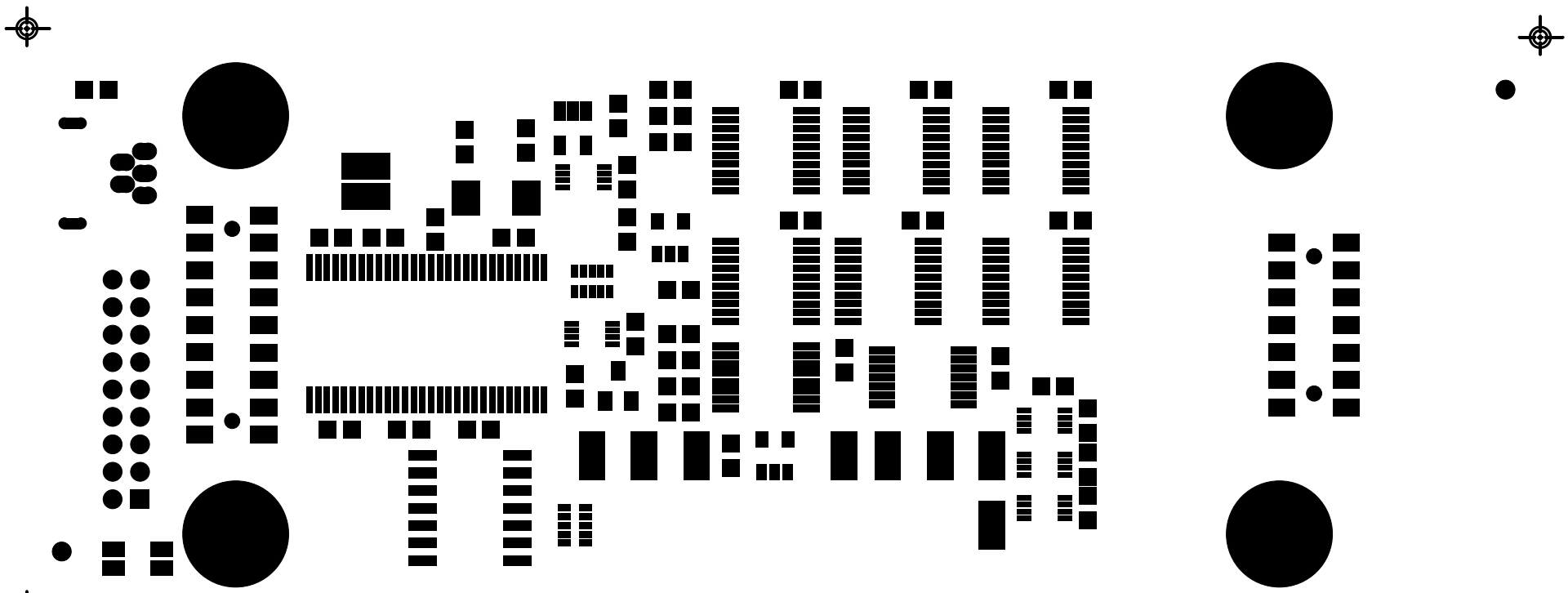
WOODS HOLE OCEANOGRAPHIC INSTITUTION
uMODEM GPIO V1.0
SOLDER SIDE L4



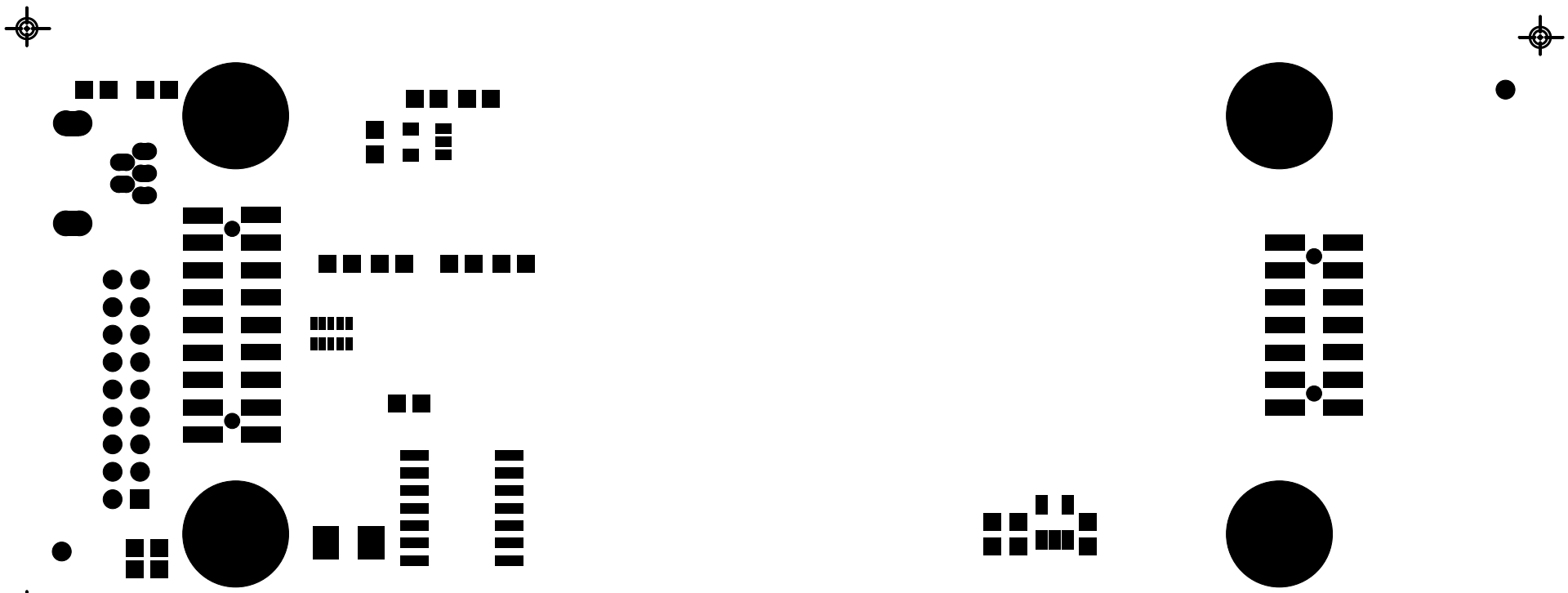
WOODS HOLE OCEANOGRAPHIC INSTITUTION
uMODEM GPIO V1.0
SILKSCREEN L1



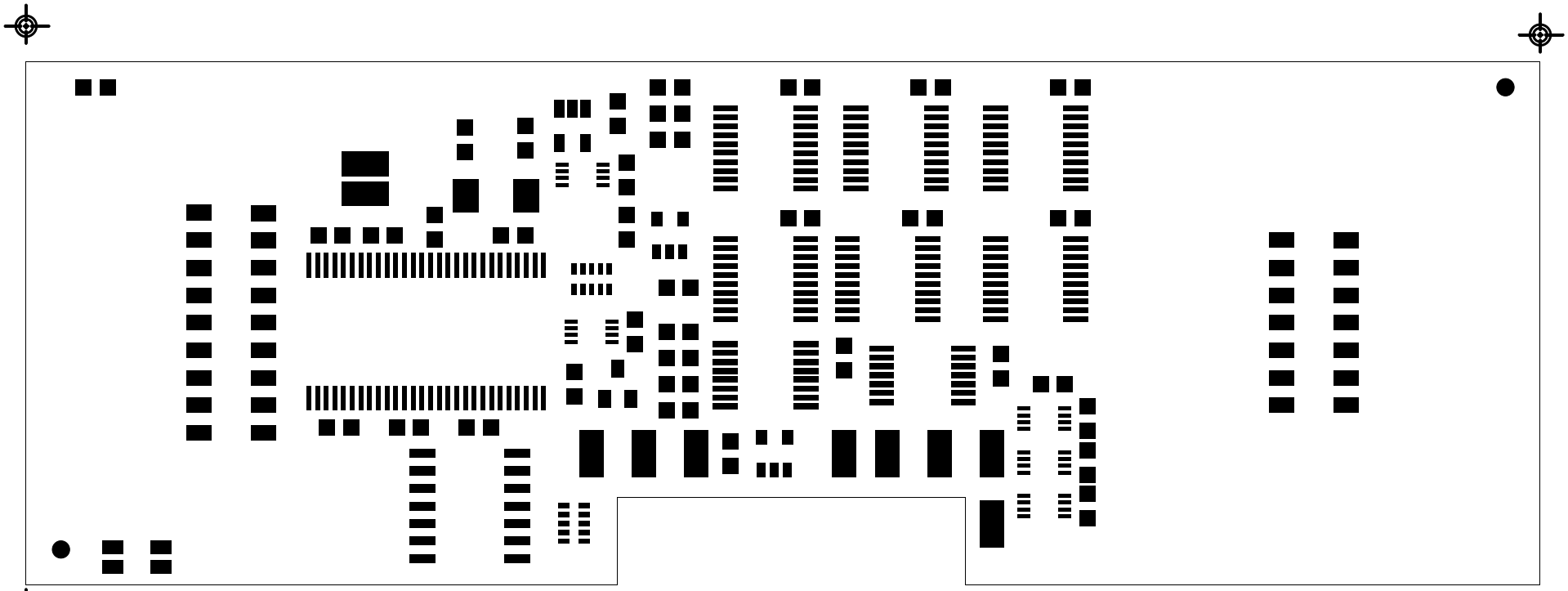
WOODS HOLE OCEANOGRAPHIC INSTITUTION
uMODEM GPIO V1.0
SILKSCREEN L4



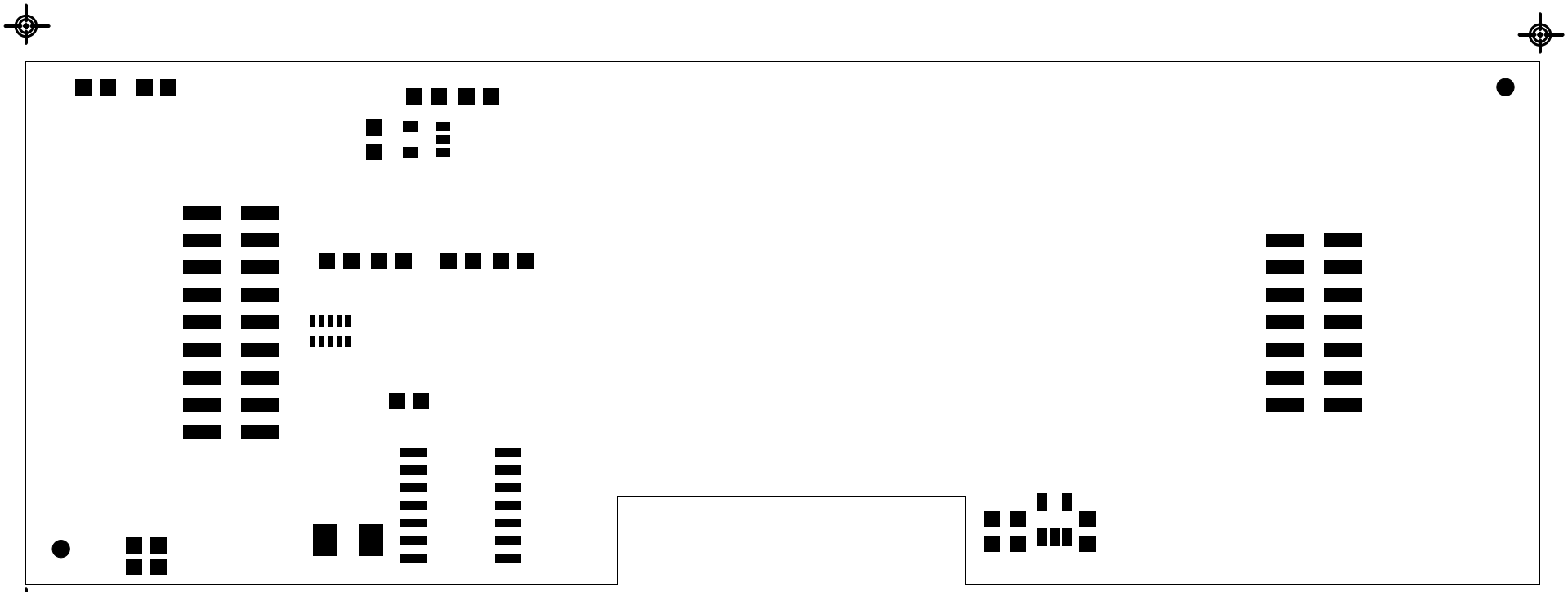
WOODS HOLE OCEANOGRAPHIC INSTITUTION
uMODEM GPIO V1.0
SOLDER MASK L1



WOODS HOLE OCEANOGRAPHIC INSTITUTION
uMODEM GPIO V1.0
SOLDER MASK L4



WOODS HOLE OCEANOGRAPHIC INSTITUTION
uMODEM GPIO V1.0
PASTE MASK L1



WOODS HOLE OCEANOGRAPHIC INSTITUTION
uMODEM GPIO V1.0
PASTE MASK L4