AN4008

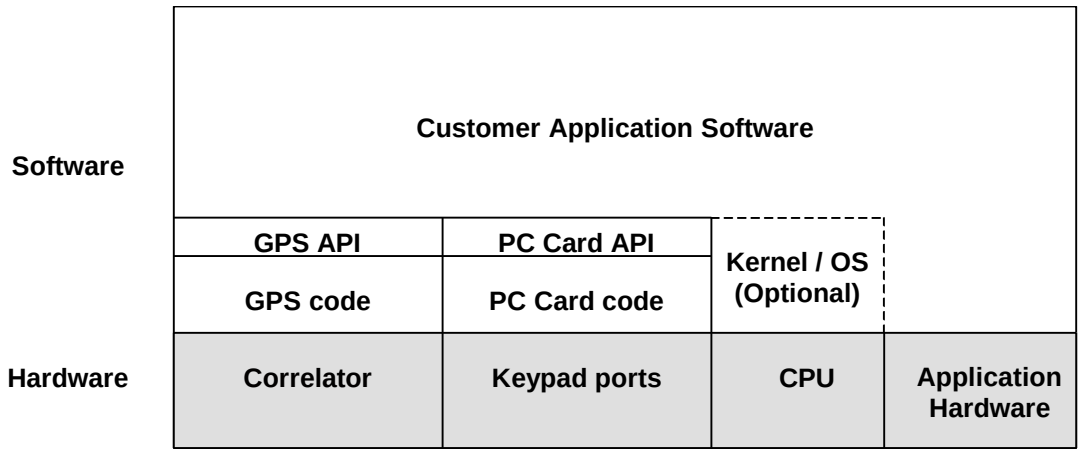
Global Positioning System (GPS) PC Card Reference Design

by Jacky-W.C. Chan
Market Development
Transportation Systems Group
Asia Pacific Region

Introduction

Implementing a GPS receiver in the PC Card form factor has several advantages over the traditional GPS module with RS-232 interface: no extra cable, built-in power and small form factor. Also, the increasing popularity of the notebook computer and hand-held devices nourishes the PC Card market. This application note illustrates GPS PC Card reference design using the Motorola MCU MMC2003 and RF Module PSRF1111A.

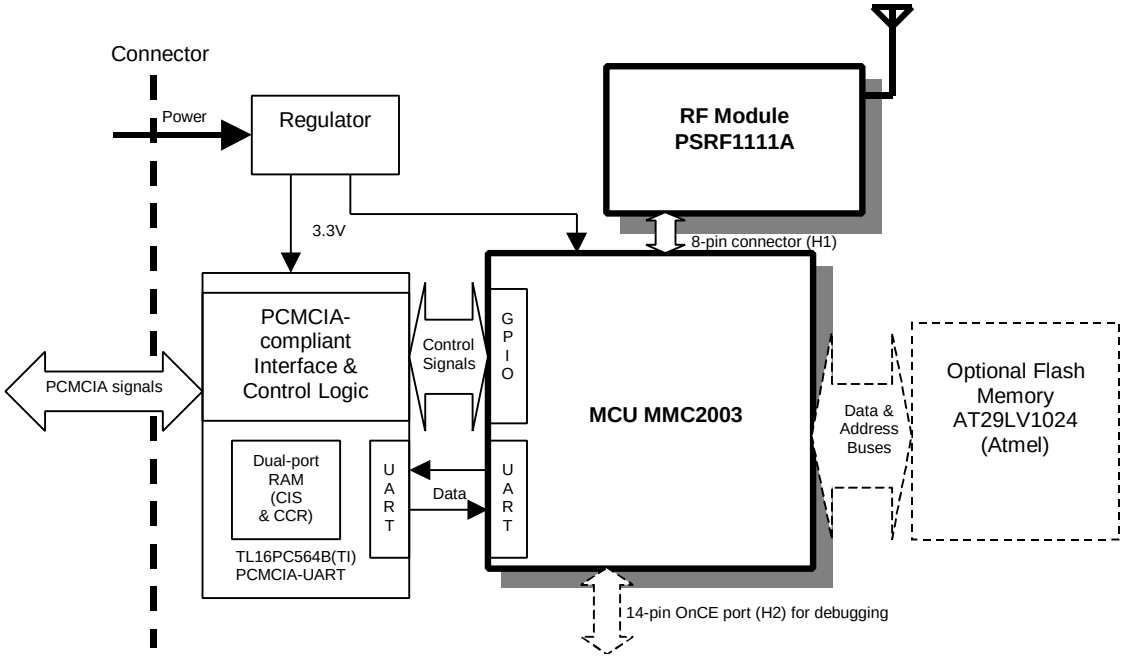
System Architecture



From the application developer perspective, the software all-in-one approach offers potential cost and functionality benefits. However, such an approach could entail undue complexity and risk, with a negative impact on the product development cycle. These concerns can be mitigated with well-defined and very flexible applications programming interfaces (APIs).

In this reference design, GPS API and PC Card API are provided to insure well-defined interactions between the application code and the GPS/PC Card code residing in the same MCU.

Hardware



Essential components are a Motorola MMC2003, an RF Module PSRF1111A and a PCMCIA-UART interface chip TL16PC564B (from Texas Instrument). The optional Flash memory AT29LV1024 (from Atmel) is used for software development before masking the MCU. The 3.3V voltage regulator is used to convert 5V into 3.3V supply. The OnCE port is available for debugging.

The GPS signal is received via the RF module and converted to digital signals. The MCU processes the signals and passes to a host via the PCMCIA-UART interface chip. This chip acts as a bridge between the MCU and the PCMCIA bus. Also, it stores the Card Information Structure (CIS) and the Card Configuration Register (CCR) to comply with the PCMCIA standard. In the host, either a handheld device or a notebook PC, the overall system is emulated as a communication (COM) device with choices of address and interrupt.

The bill of materials, circuit schematics and PCB layouts are provided in the Appendices.

Software

The software consists of GPS code and PC Card code. Both will be provided in the form of object files. As interfaces to them, GPS API and PC Card API are available in "C" language.

GPS API

The GPS API performs GPS functionality. The complete GPS API is documented separately. Samples are provided as follows:

`GPS_API_STATUS GPS_Reset(void)` sets receiver to default values.

`GPS_API_STATUS GPS_Get_PVT(void)` returns most recent position, velocity and time.

PC Card API

The PC Card API initializes TL16PC564B and provides access to CIS and CCR.

Reference Card Information Structure (CIS)

This is provided in the Appendices. Refer to the PCMCIA standard also.

Function Calls

`int InitPCMCIA(void)` initializes PCMCIA

`void WriteAttribute(int address, char data)` writes CIS and CCR.

`char ReadAttribute(int address)` returns the value of CIS and CCR.

The header file listing is provided in the Appendices.

Windows 95 Drivers

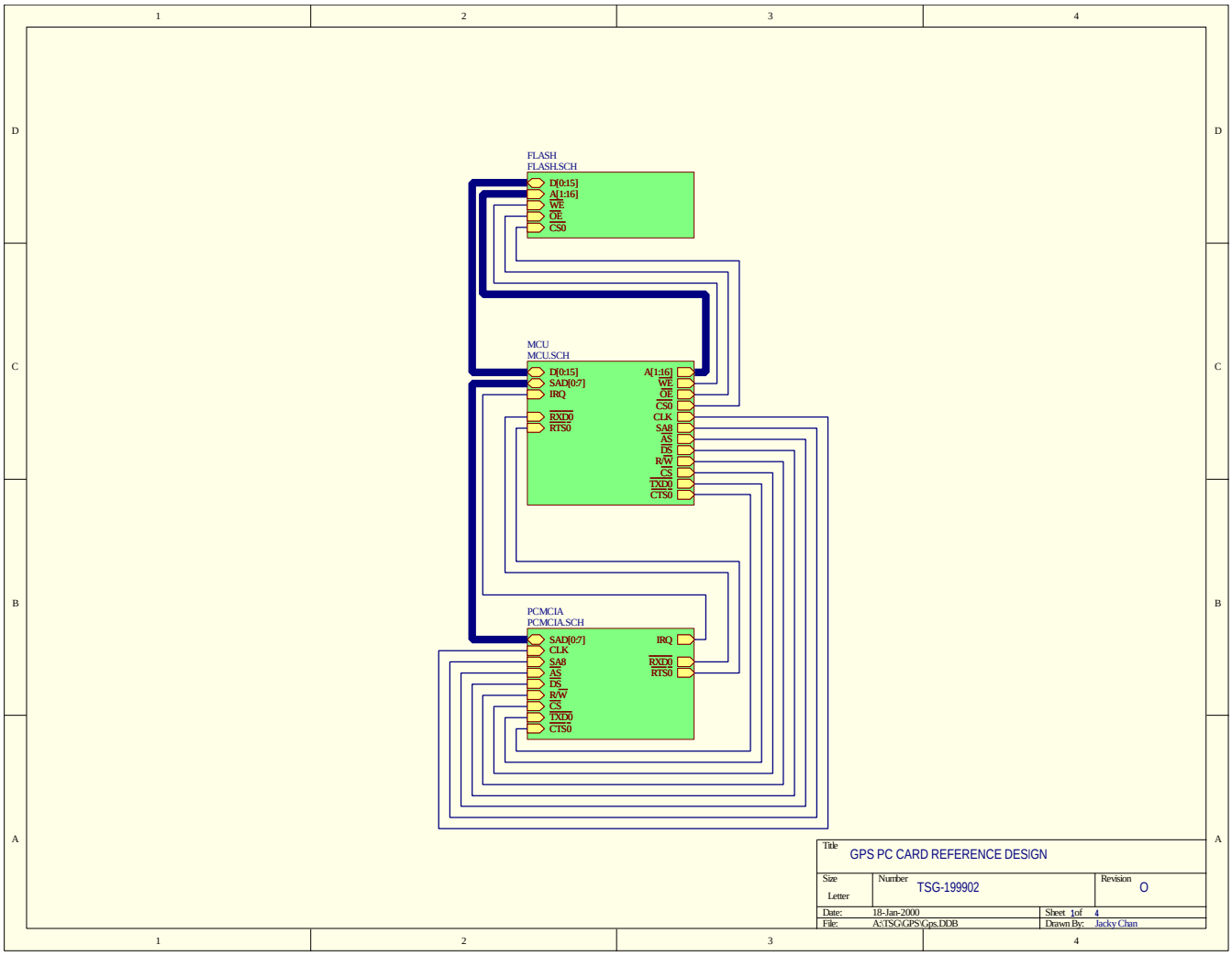
As a demonstration to support Windows 95 Plug-&-Play capability, three driver files, namely serial.vxd, serialui.dll and gps.inf are provided.

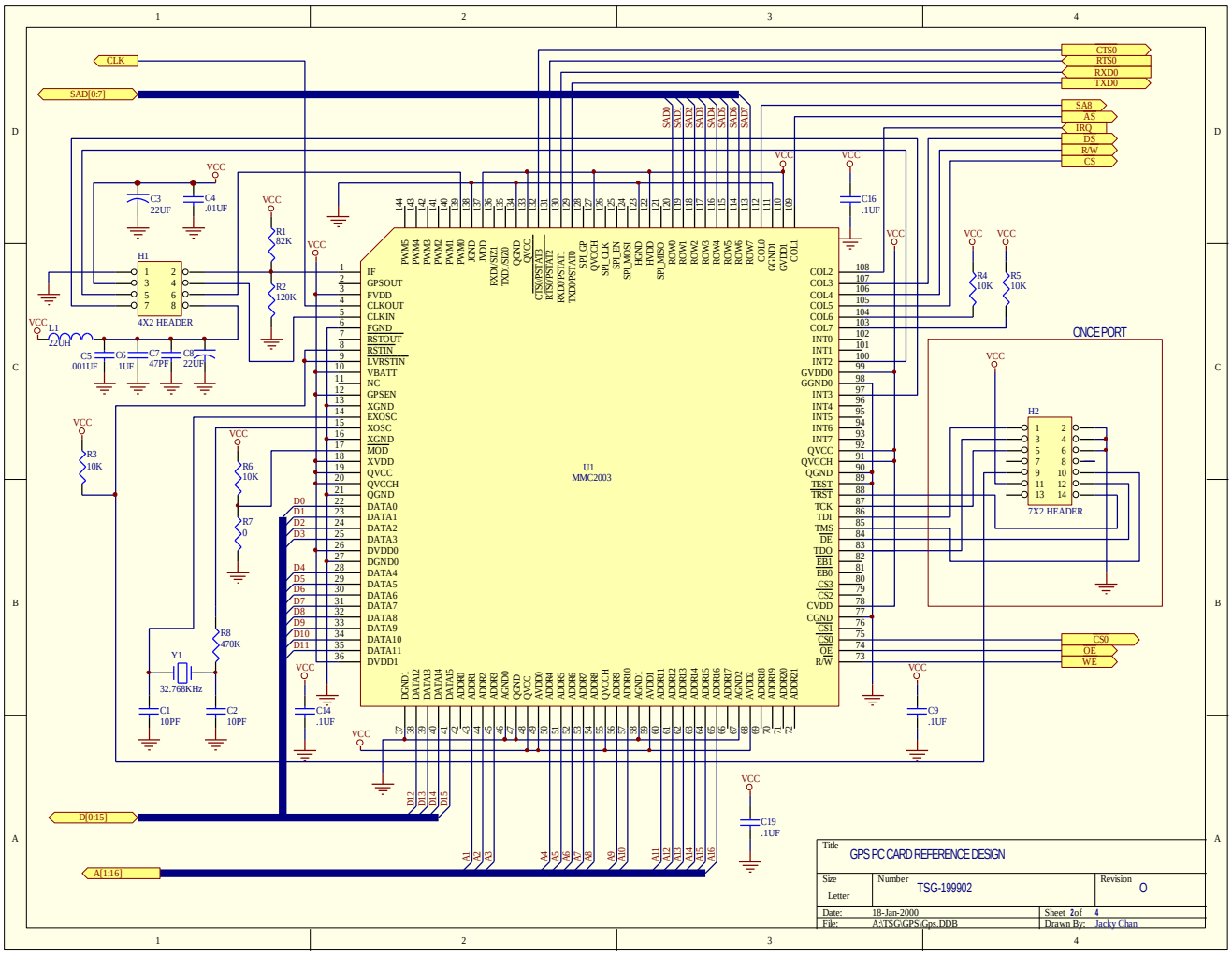
Appendices

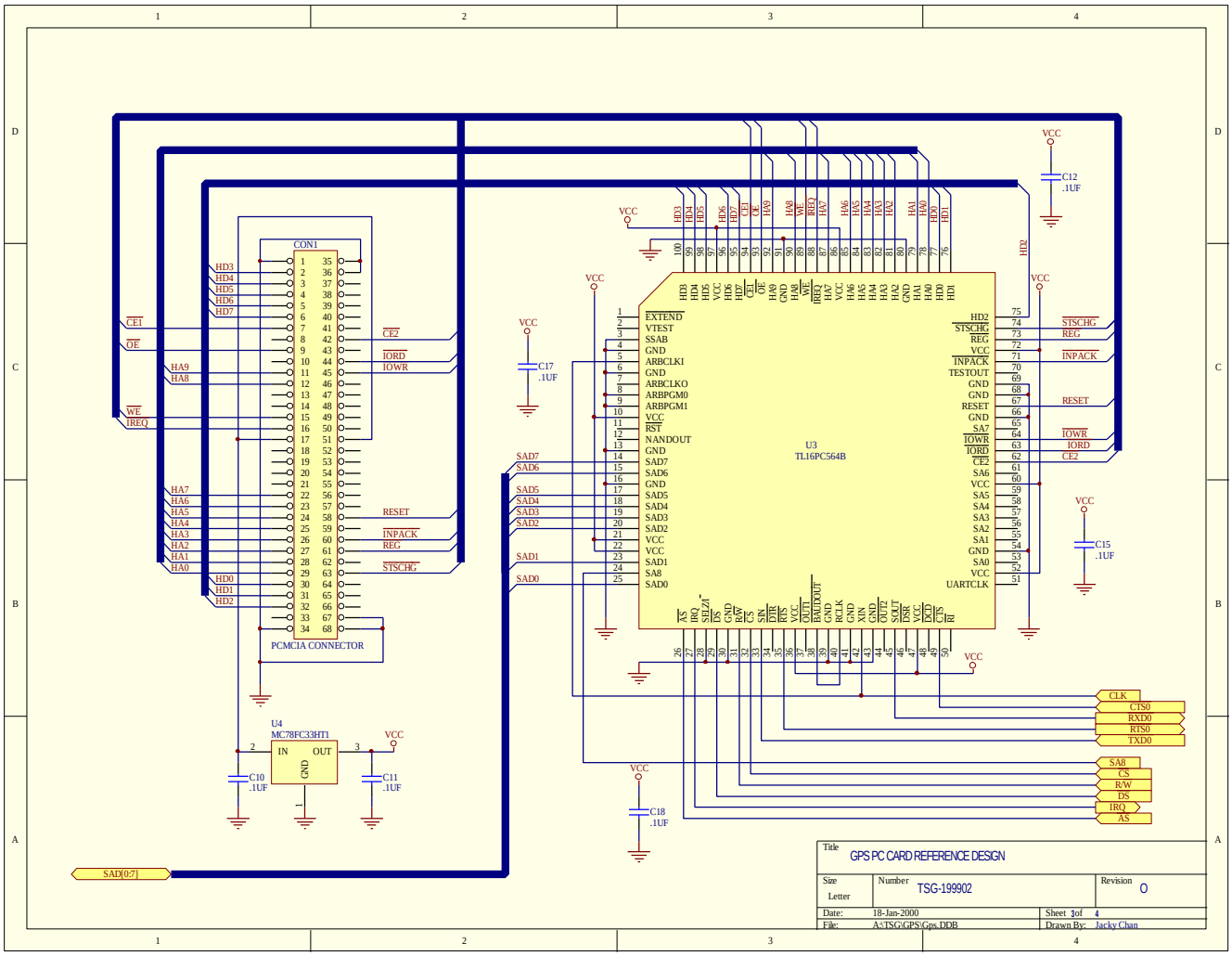
Bill of Materials

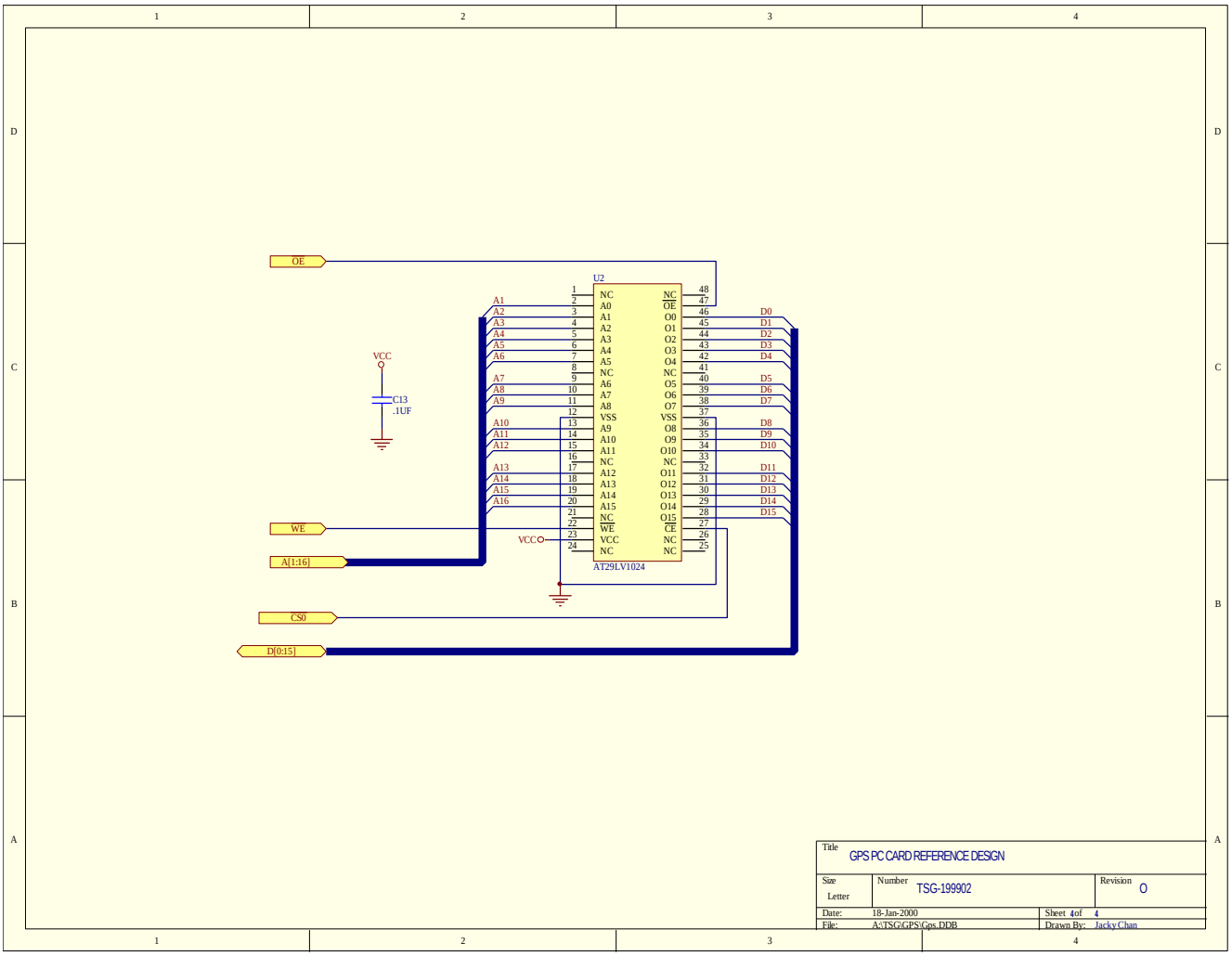
Item	Qty.	Part	Reference
1	1	.01µF	C4
2	1	.001µF	C5
3	6	.1µF	C6 C9 C10 C11 C12 C13 C14 C15 C16 C17 C18 C19
4	1	0Ω	R7
5	1	4X2 Header	H1
6	1	7X2 Header	H2
7	4	10KΩ	R3 R4 R5 R6
8	2	10pF	C1 C2
9	2	22µF	C3 C8
10	1	22µH	L1
11	1	32.768kHz	Y1
12	1	47pF	C7
13	1	82KΩ	R1
14	1	120KΩ	R2
15	1	470KΩ	R8
16	1	AT29LV1024	U2
17	1	MC78FC33HT1	U4
18	1	MMC2003	U1
19	1	PCMCIA Connector	CON1
20	1	TL16PC564B	U3
21	1	RF Module	PSRF1111A

Circuit Schematics

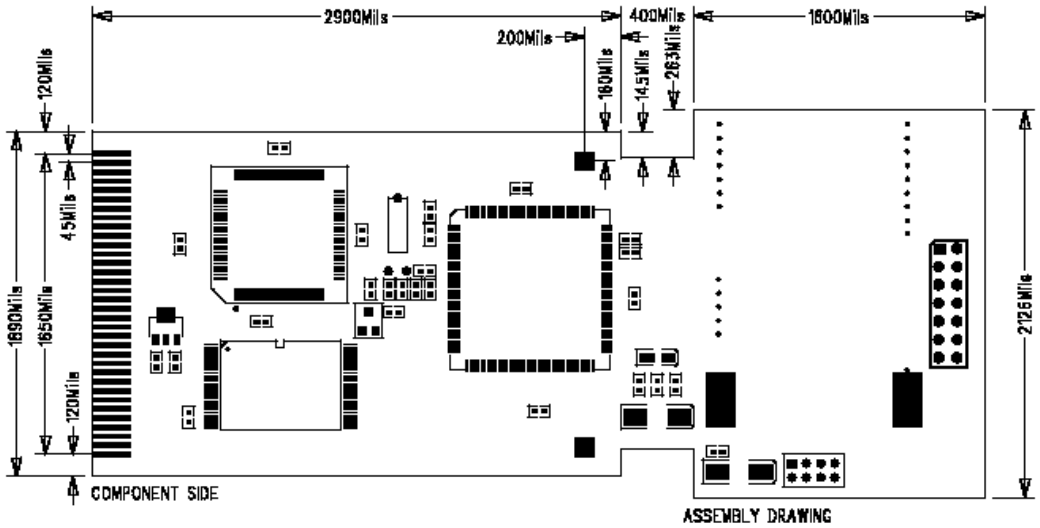


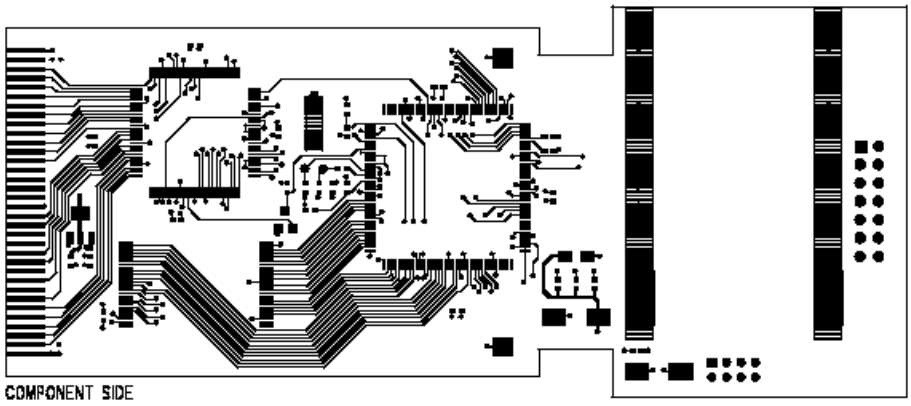

CLK
SAD[0:7]
D[0:15]
A[1:16]
CTS0
RTS0
RXD0
TXD0
SA8
AS
IRQ
DS
R/W
CS
CS0
OE
WE
U1
MMC2003
4X2 HEADER
7X2 HEADER
ONCE PORT
Y1
32.768KHz
Title
GPS PC CARD REFERENCE DESIGN
Size
Letter
Number
TSG-199902
Revision
0
Date:
18-Jan-2000
Sheet 2of 4
File:
A:\TSG\GPS\Gps.DDB
Drawn By:
Jacky Chan

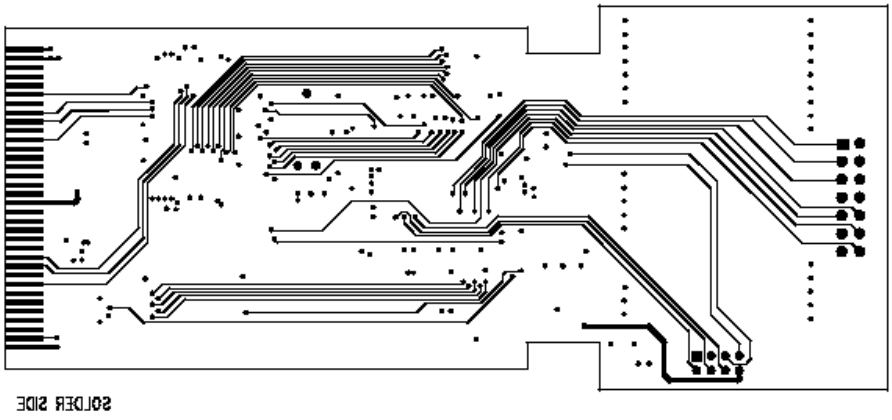

CON1
PCMCIA CONNECTOR
U3
TL16PC564B
U4
MC78FC33HT1
IN
OUT
GND
VCC
C10 .1UF
C11 .1UF
C12 .1UF
C15 .1UF
C17 .1UF
C18 .1UF
CE1
OE
WE
IREQ
HA9
HA8
HA7
HA6
HA5
HA4
HA3
HA2
HA1
HA0
HD0
HD1
HD2
HD3
HD4
HD5
HD6
HD7
CE2
IORD
IOWR
RESET
INPACK
REG
STSCHG
SAD7
SAD6
SAD5
SAD4
SAD3
SAD2
SAD1
SAD0
SAD[0:7]
CLK
CTS0
RXD0
RTS0
TXD0
SA8
CS
R/W
DS
IRQ
AS
Title
GPS PC CARD REFERENCE DESIGN
Size
Letter
Number
TSG-199902
Revision
O
Date:
18-Jan-2000
Sheet 3of 4
File:
A:\TSG\GPS\Gps.DDB
Drawn By:
Jacky Chan

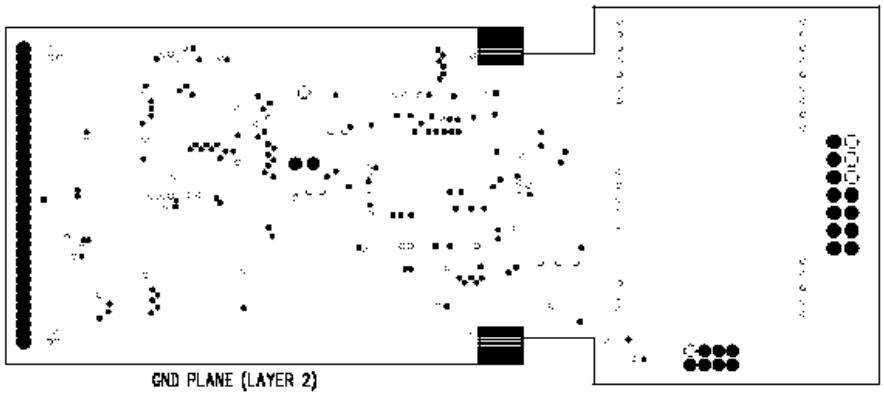

OE
U2
NC
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
VSS
WE
VCC
NC
O0
O1
O2
O3
O4
O5
O6
O7
O8
O9
O10
O11
O12
O13
O14
O15
CE
AT29LV1024
VCC
C13
.1UF
WE
A[1:16]
CS0
D[0:15]
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
Title
GPS PC CARD REFERENCE DESIGN
Size
Letter
Number
TSG-199902
Revision
0
Date:
18-Jan-2000
Sheet 4 of 4
File:
A:\TSG\GPS\Gps.DDB
Drawn By:
Jacky Chan

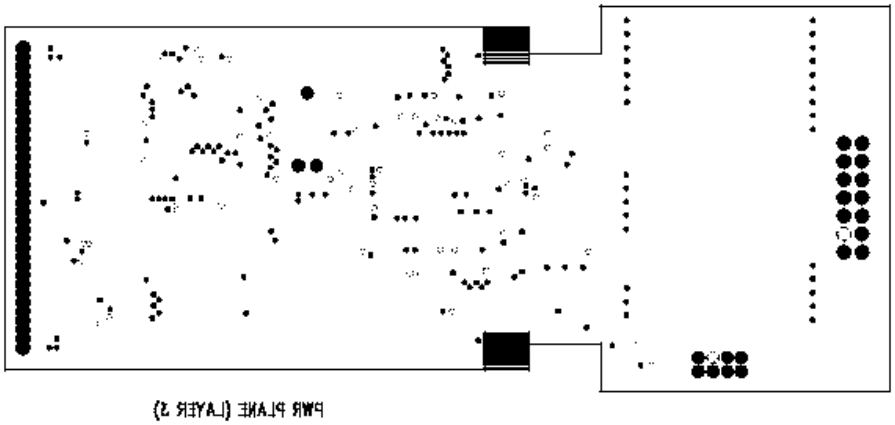
PCB Layouts (Not to scale)

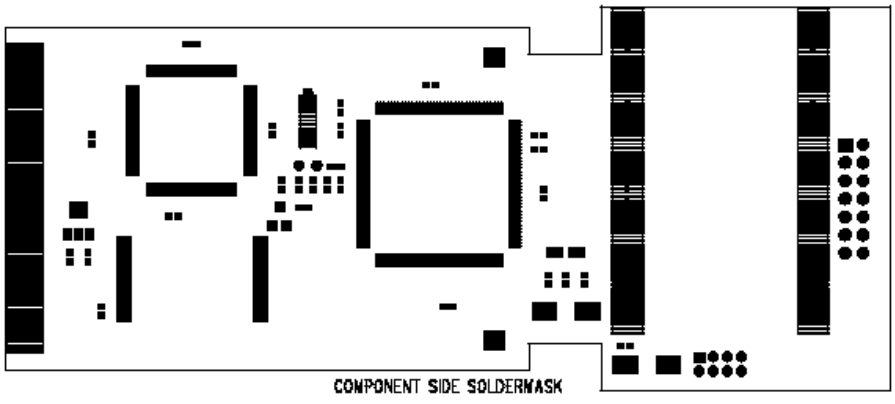


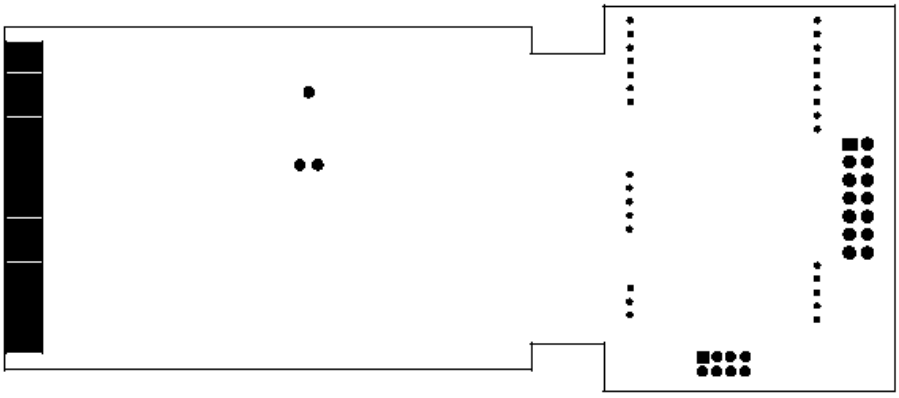




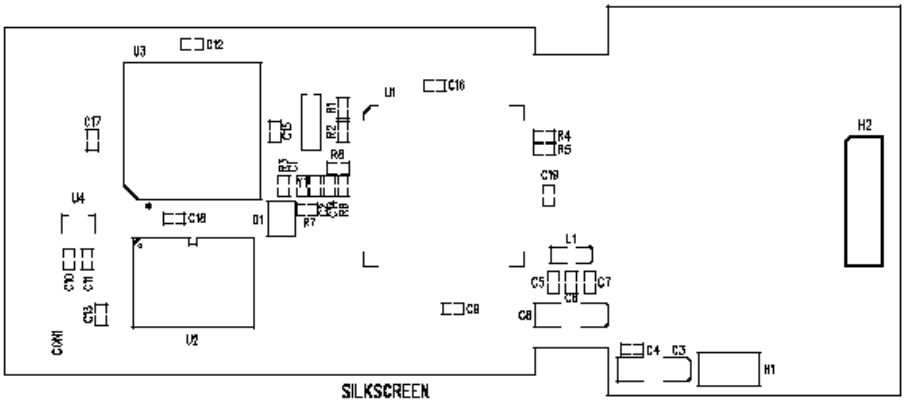

GND PLANE (LAYER 2)

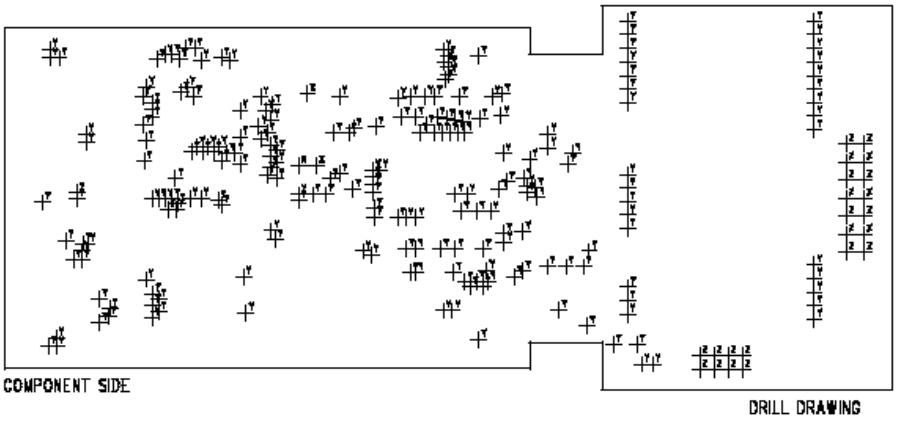

PWR PLANE (LAYER 3)


COMPONENT SIDE SOLDERMASK



SOLDER SIDE SOLDERMASK





COMPONENT SIDE

DRILL DRAWING

Reference CIS

Address (Hex)	Data (Hex)	Description and interpretation
0	01	Device Info Tuple
2	02	Link to next tuple
4	00	Not a memory device
6	FF	Termination byte
8	20	Manufacturer Identification Tuple
A	04	Link to next tuple
C	09	Manufacturer 09(low byte); 01(high byte) = 0109
E	01	
10	00	Product code = 00
12	00	Revision information = 0
14	15	Level 1 Version / Product Information Tuple
16	2B	Link to next tuple
18	04	Major version number = 4
1A	01	Minor version number = 1 (Release 2.0 or 2.1)
1C	4D	'M'
1E	6F	'o'
20	74	't'
22	6F	'o'
24	72	'r'
26	6F	'o'
28	6C	'l'
2A	61	'a'
2C	00	<End manufacturers name>
2E	47	'G'
30	50	'P'
32	53	'S'
34	20	<space>
36	50	'P'
38	43	'C'
3A	20	<space>
3C	43	'C'
3E	61	'a'
40	72	'r'
42	64	'd'
44	20	<space>
46	52	'R'
48	65	'e'
4A	66	'f'
4C	65	'e'
4E	72	'r'
50	65	'e'

Address (Hex)	Data (Hex)	Description and interpretation
52	6E	'n'
54	63	'c'
56	65	'e'
58	20	<space>
5A	44	'D'
5C	65	'e'
5E	73	's'
60	69	'i'
62	67	'g'
64	6E	'n'
66	00	<End product name>
68	30	'0'
6A	00	<End version information>
6C	FF	Termination byte (End tuple)
6E	21	Function Identification Tuple
70	02	Link to next tuple
72	02	Function code 2 = serial interface
74	01	Initialization byte = init during POST and no ROM
76	22	Function Extension Tuple
78	04	Link to next tuple
7A	00	Tuple function extension type = 0
7C	02	UART type = 2 (16550 UART)
7E	0F	UART capabilities = even, odd parity; mark, space; 2, 1.5, 1 stop bit and 8, 7, 6, 5 bit characters
80	7F	
82	1A	Configuration Table Tuple
84	05	Link to next tuple
86	01	2-byte base address register; 1-byte configuration mask
88	24	Index number of last configuration table entry = 24
8A	00	Base address of configuration registers = 0200
8C	02	
8E	0F	Configuration mask = Config Option, Status, Pin Replacement and Socket and Copy
90	1B	Configuration Table Entry Tuple
92	09	Link to next tuple
94	E0	Config Index = E0 (Interface byte used, default entry, Index = 20)
96	41	Interface description byte = Mem I/O interface, READY active, BVD inactive, WP inactive, MWait inactive.
98	18	Feature selection byte = (I/O, INTR defined)
9A	A3	I/O description byte (3 address lines, 8-bit device, includes range)
9C	60	Length size descriptor = 1 address and 1 length

Address (Hex)	Data (Hex)	Description and interpretation
9E	F8	Start address = 03F8
A0	03	
A2	07	Length of address block = 0-7 (8 bytes)
A4	24	Interrupt descriptor byte = IRQ4, level mode
A6	1B	Configuration Table Entry Tuple
A8	08	Link to next tuple
AA	21	Config Index = 21, no interface description byte
AC	18	Feature selection byte
AE	A3	I/O description byte (3 address lines, 8-bit device, includes range)
B0	60	Length size descriptor = 1 address and 1 length
B2	F8	Start address = 02F8
B4	02	
B6	07	Length of address block = 0-7 (8 bytes)
B8	23	Interrupt descriptor byte = IRQ3, level mode
BA	1B	Configuration Table Entry Tuple
BC	08	Link to next tuple
BE	22	Config Index = 22, no interface description byte
C0	18	Feature selection byte
C2	A3	I/O description byte (3 address lines, 8-bit device, includes range)
C4	60	Length size descriptor = 1 address and 1 length
C6	E8	Start address = 03E8
C8	03	
CA	07	Length of address block = 0-7 (8 bytes)
CC	24	Interrupt descriptor byte = IRQ4, level mode
CE	1B	Configuration Table Entry Tuple
D0	08	Link to next tuple
D2	23	Config Index = 23, no interface description byte
D4	18	Feature selection byte
D6	A3	I/O description byte (3 address lines, 8-bit device, includes range)
D8	60	Length size descriptor = 1 address and 1 length
DA	E8	Start address = 02E8
DC	02	
DE	07	Length of address block = 0-7 (8 bytes)
E0	23	Interrupt descriptor byte = IRQ3, level mode
E2	1B	Configuration Table Entry Tuple
E4	06	Link to next tuple
E6	24	Config Index = 24, no interface description byte
E8	18	Feature selection byte
EA	23	I/O description byte (3 address lines, no range)
EC	30	Interrupt descriptor byte (use IRQ mask)

Address (Hex)	Data (Hex)	Description and interpretation
EE	FF	Permissible IRQ lines = IRQ0 - 15
F0	FF	
F2	14	No Link Tuple
F4	00	Link to next tuple
F6	FF	Termination Tuple (end of tuple list)

Header File Listing

pcmcia.h

```
#define NUM_CIS 124
#define NUM_CCR 4
#define CCR_OFFSET 256
#define CONTROL_REGISTER 272
#define PGMCLK_REGISTER 288

#define RSCR 0x10001000
#define KDDR 0x10003004
#define KPDR 0x10003006

#define CKOS_MASK 0x00000200
#define CKOE_MASK 0x00000100
#define KDDR_MASK 0x3BFF
#define ALE_MASK 0x0200
#define IRQ_MASK 0x0400
#define RD_MASK 0x0800
#define WR_MASK 0x1000
#define CS_MASK 0x2000

void InitPCMCIA(void);
void WriteAttribute(int, char);
char ReadAttribute(int);

const char CIS_data[NUM_CIS] =
{0x01, 0x02, 0x00, 0xFF,   /* Device Info Tuple */
0x20, 0x04, 0x09, 0x01, 0x00, 0x00,     /* Manufacturer ID Tuple */
0x15, 0x2B, 0x04, 0x01, 'M', 'o', 't', 'o', 'r', 'o', 'l', 'a', 0x00, 'G', 'P', 'S', '  ',
'P', 'C', ' ', 'C', 'a', 'r', 'd', ' ', 'R', 'e', 'f', 'e', 'r', 'e', 'n', 'c', 'e', ' ',
'D', 'e', 's', 'i', 'g', 'n', 0x00, '0', 0x00, 0xFF,        /* Product Info Tuple */
0x21, 0x02, 0x02, 0x01,          /* Function ID Tuple */
0x22, 0x04, 0x00, 0x02, 0x0F, 0x7F,     /* Function Extension Tuple */
0x1A, 0x05, 0x01, 0x24, 0x00, 0x02, 0x0F,      /* Configuration Tuple */
0x1B, 0x09, 0xE0, 0x41, 0x18, 0xA3, 0x60, 0xF8, 0x03, 0x07, 0x24,  /* COM 1 */
0x1B, 0x08, 0x21, 0x18, 0xA3, 0x60, 0xF8, 0x02, 0x07, 0x23,        /* COM 2 */
0x1B, 0x08, 0x22, 0x18, 0xA3, 0x60, 0xE8, 0x03, 0x07, 0x24,        /* COM 3 */
0x1B, 0x08, 0x23, 0x18, 0xA3, 0x60, 0xE8, 0x02, 0x07, 0x23,        /* COM 4 */
0x1B, 0x06, 0x24, 0x18, 0x23, 0x30, 0xFF, 0xFF,             /* IRQ 0 - 15 */
0x14, 0x00, 0xFF};  /* Termination Tuple */
char CCR_data[NUM_CCR] = {0x00, 0x00, 0x00, 0x00};
```

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution; P.O. Box 5405, Denver, Colorado 80217. 1-800-441-2447 or 1-303-675-2140
JAPAN: Nippon Motorola Ltd. SPD, Strategic Planning Office 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141, Japan. 03-5487-8488
Mfax™, Motorola Fax Back System: RMFAX0@email.sps.mot.com; http://sps.motorola.com/mfax/; TOUCHTONE 1-602-244-6609;
US and Canada ONLY 1-800-774-1848
HOME PAGE: http://motorola.com/sps/