Specifications: 1375/1455 GPS RTC

Rev: B

Description:

The 1375/1455 series modules are complete real time clocks (RTC) that use GPS data and a user supplied local time offset to replicate the register set of the NXP PCF8563 IC so that they can function as drop in replacements for the 1374/1362 RTC modules from TES. Additionally, the full GPS data from the GPGGA, GPGSA, GPGSV and GPRMC NMEA sentences is decoded on the fly to populate a 128 byte I^2C register map with full navigational data. The PPS output is accurate to +/-1ppm.

The RTC data interface is a standard I^2C bus operating at 100kbps. The RTC/Slave side of the interface operates at an internal 3.3V regulated from the master side (VDD) input voltage from 3.3V to 5.5V. The master to slave I^2C interface is level shifted and includes master side bus pull up resistors of 2.21K. All inputs and outputs are ESD protected (See notes on Table 2). Slave addressing information can be found in Fig. 8 and 9.

The local time offset from UTC can be entered directly into the offset registers or can be set indirectly via a time setting function, such as is used in the TES SmartNixie clock backplane. The offset can be plus or minus 23h 59m 59s in 1s increments from UTC. A one hour positive offset function for DST is also included. Local time is computed from the GPS UTC and UDC data at the end of every second and includes date adjustments for leap years. DST can be turned on and off with a short clock set sequence or programmed directly as part of the local time offset..

Once a GPS fix has been established, the battery backed internal RTC will keep time for up to three months with +/-20ppm accuracy with the power off or no GPS fix. Satellite data is stored in battery backed RAM to speed fix acquisition after brief power or satellite view interruptions.

An interrupt output (INT) can be set to pulse high for 560mS every second (1PPS), 30 seconds (1PP30S), minute (1PPM), hour (1PPH) or off, The INT goes high after the GPS data has been fully decoded and the local time calculation has been completed (Useful to control slave clocks). INT is low during NMEA sentence reception and then high during RS232 idle.

Three lamps indicate the status of the module: FIX blinks at 10Hz when the GPS indicates no fix is available, 2Hz if the fix signal is toggling faster than once per two seconds and solid on when a stable fix is indicated. PPS blinks at 1Hz 50% duty when the GPS PPS is stable at 1Hz 10% duty. MON blinks at 5Hz all the time and signals the main processing loop is running.

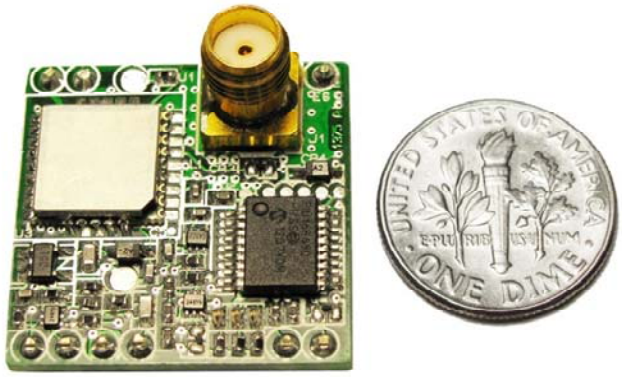


Fig 1. 1375 horizontal GPS RTC

Revision History

Date	FW	Affects	Description
2012/08/05	All	All	Initial release.
2012/09/03	All	Table 4	Direct write prescale (PS) section updated with 1PP30S and nn > Off behavior
2012/09/05	All	Various	Typographical cleanup

Table 1: GPS Receiver specifications

Receiver	GPS L1 C/A-code, SPS	
Chip set	Mediatek MT3339	
Channels	66/22 (search/track)	
Tracking sensitivity	-165 dBm typ.	
Navigation sensitivity	-165 dBm typ.	
Navigation sensitivity, re-acq.	-160 dBm typ.	
Navigation sensitivity, cold acq.	-148 dBm typ.	
External RF amp net gain range	0 to +30 dB	
Antenna type	This module requires an active antenna, 3.3V nominal	
Update rate	1 Hz	
Position accuracy	3.0 m (67%) typ. Horizontal	(With nominal GPS signal levels -130dBm.)
	5.0 m (67%) typ. Vertical	
	0.02 m/s (50%) typ. Velocity	
Max altitude/velocity	<60,000 ft/<1,000 knots	
Time to First Fix, cold acq.	31 s typ.	
Time to First Fix, warm acq.	31 s typ.	
Time to First Fix, hot acq.	1 s typ.	
Power consumption, Full Power	35 mW typ. @ 3.3 V (11mA)	<12 GPS satellites in track
Power consumption, Backup	15 µW typ. @ 3.0 V (4.5uA)	
Host port protocol	NMEA-0183 rev. 3.01	
Serial data format	8 bits, no parity, 1 stop bit	
Serial data speed	9600 baud	
PPS output	100 ms high pulse, rising edge +/-1 µs @ full second GPS epoch	

Table 2: Interface connector signals, horizontal module

Pin	Name	Description
E1	SDA	I^2C serial data. Level shifted
E2	SCL	I^2C serial clock. Level shifted
E3	GND	Power return
E4	VDD	3.3V to 5.5V power input. I^2C master side reference
E5	TXD	PC logic (Idle state is low) RS232 out from GPS, 0V-VDD
E6	VBAT	Test pin for backup battery, do not connect to user circuit
E7	INT	Active high interrupt on time update
E8	PPS	100mS high pulse from GPS, +/-1ppm accuracy (GPS fix)
E9	RXD	PC logic (Idle state is low) RS232 in to GPS, 0V-VDD
SMA	RFIN	GPS RF input, passive or active (3.3V) antenna

Notes:

1. I^2C SDA and SCL are ESD protected to 2KV.
2. INT, PPS, TXD, RXD and RFIN are ESD protected to 20KV.
3. Absolute max input of VDD is 12V, limited by the level shifter transistor gates. Note that SDA and SCL are pulled up to VDD through 2.21K resistors so that SDA and SCL will equal VDD when inactive. The TXD output is also pulled up to VDD via a 2.21K resistor but is voltage limited by its ESD protection ~6V.
4. PPS and INT output current is +/-10mA max.
5. VBAT may be measured to check the charge state of the battery, where the charge current is (3.3V - VBAT) / 249ohms when VDD is >3.4V. Otherwise, charge/discharge current can be measured across R15.

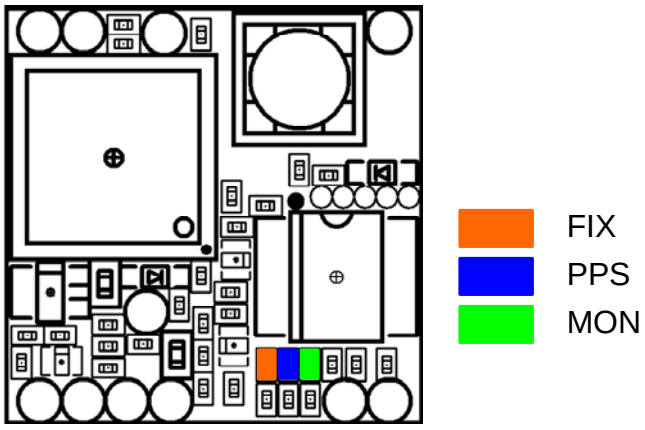


Fig 3. Status lamp locations

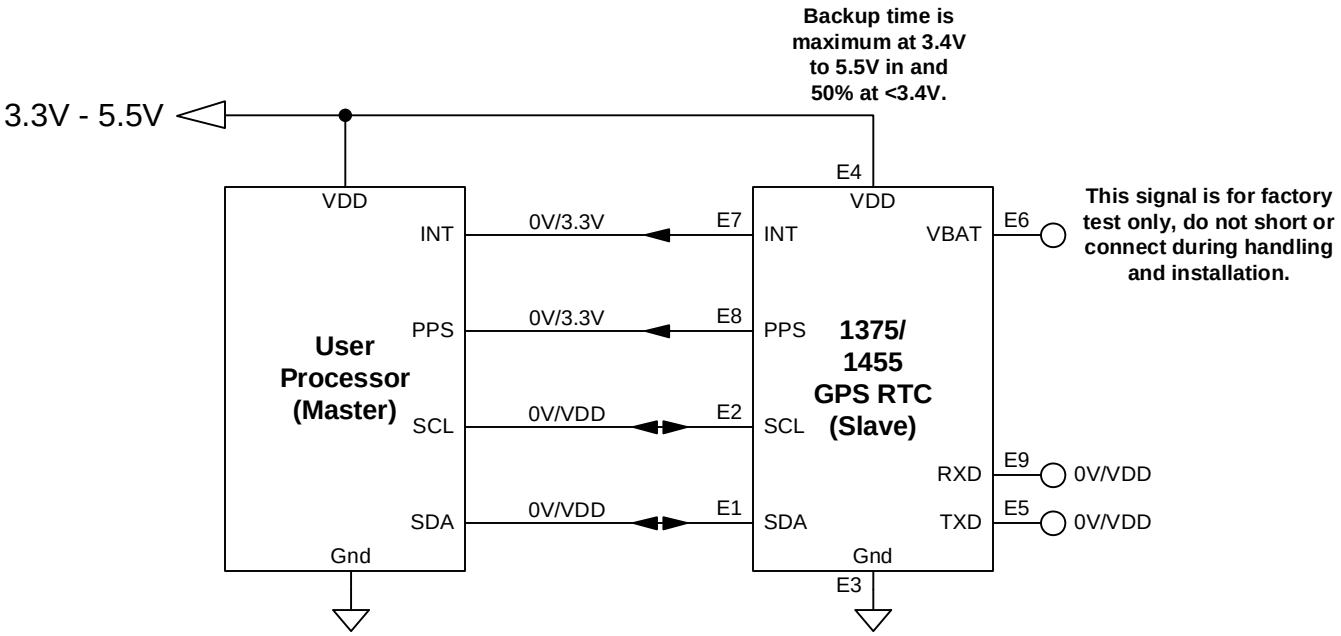


Fig 4. Typical application circuit (1375 pin numbering shown)

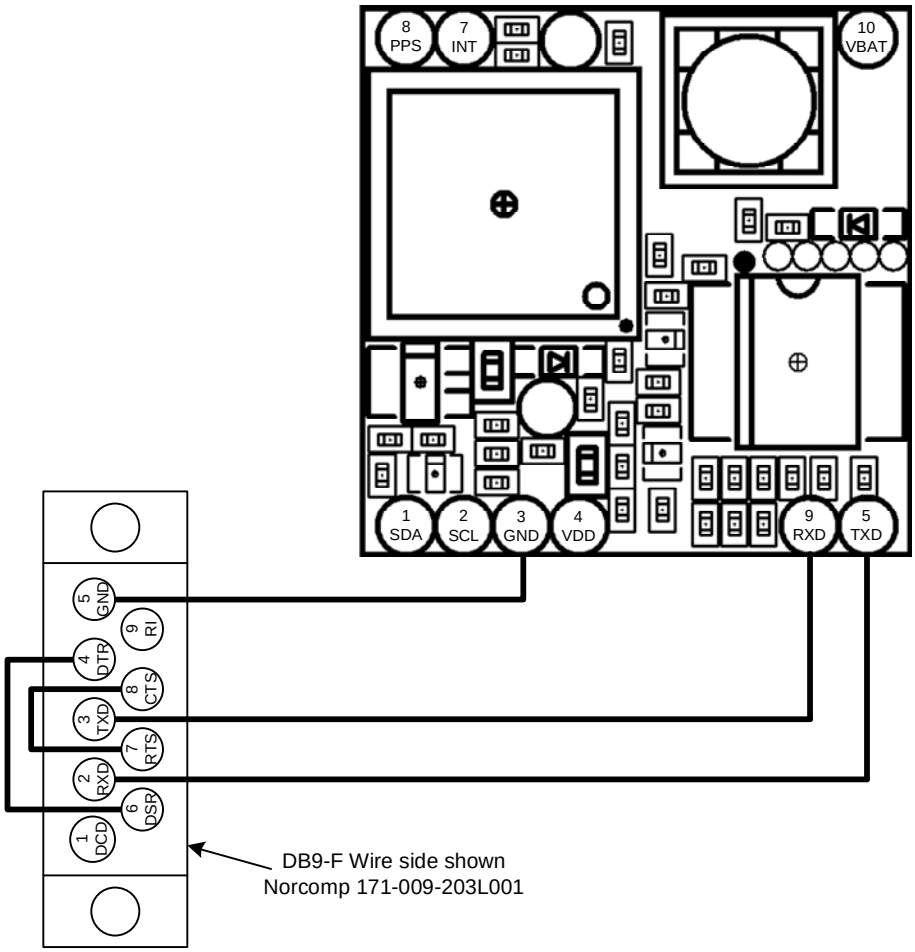


Fig 5. Module RS232 Connection

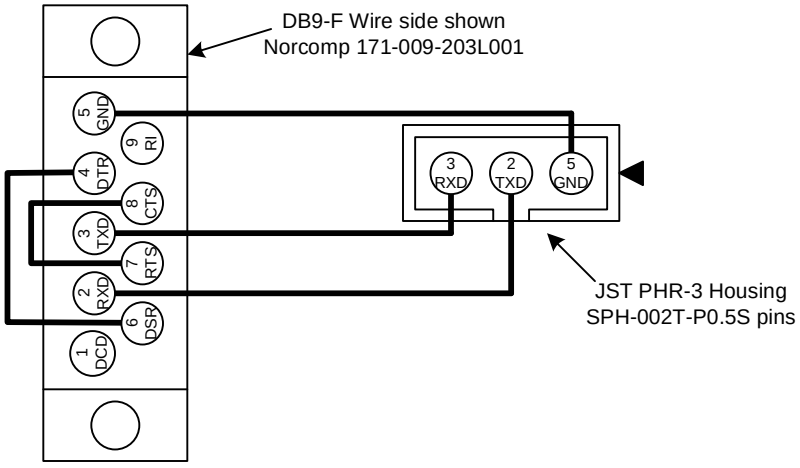
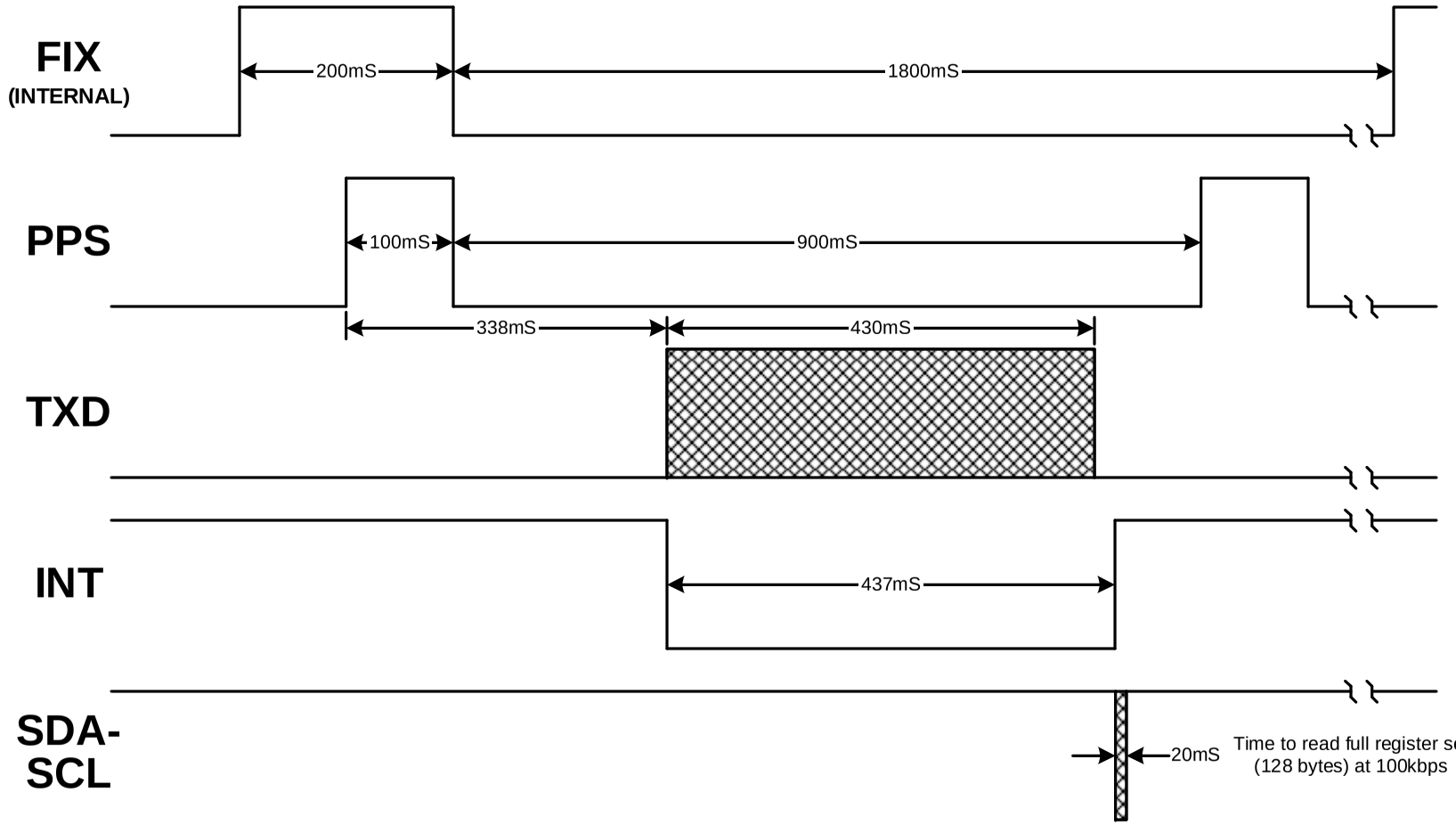


Fig 6. SmartNixie Backplane RS232 Connection



Notes:

1. The INT line goes low at the end of the first byte (1.04mS) of the first NMEA sentence (GPGGA). INT goes high after the UTC to local time conversion has been completed and transferred to the PCF time registers which happens after the last byte of the last NMEA sentence (GPRMC) and next 10mS tick (Internal).
2. Reading of all registers except the PCF time registers should be done while INT is high to prevent reading GPS data from a previous NMEA block mixed with data from a block in the process of decoding. As shown in the timing diagram, reading the entire register stack of 128 bytes takes (conservatively) only 20mS at 100kbps.
3. When the INT frequency is set to 1PP30S, the INT line goes high at 00 and 30 seconds, for 1PPM and 1PPH INT goes high at the beginning of a new minute and new hour respectively. The high period is always the same (~560mS) regardless of the INT frequency selection. INT does not toggle when set to off however PPS always toggles at once per second.
4. The NMEA data stream is decoded as it is transmitted so that errors in the stream will result in errors in the decoded data since the checksum is at the end of each sentence. The validity of the decoded data however can be verified by examining the sentence error flags and counts at register locations 0x39 and 0x3A.

Fig 7. Timing (GPS fix is stable, INT set to 1PPS)

PCF8563 Register map emulation page

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x00	00	00	se	mn	hr	dy	00	mo	yr	00	00	00	00	00	00	00
0x10	hr	mn	se	dy	mo	yr	hd	dd	mm	mm	mm	hd	dd	mm	mm	mm
0x20	sa	aa	aa	aa	sa	aa	aa	ss	ss	ss	0h	hh	hh	rv	vv	vv
0x30	pp	pp	hh	hh	vv	vv	F1	F2	RR	EF	EC	PS	dl	hr	mn	se
0x40	PF	QD	VV	SS	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12
0x50	EE	0A	AA	SN	EE	0A	AA	SN	EE	0A	AA	SN	EE	0A	AA	SN
0x60	EE	0A	AA	SN	EE	0A	AA	SN	EE	0A	AA	SN	EE	0A	AA	SN
0x70	EE	0A	AA	SN	EE	0A	AA	SN	EE	0A	AA	SN	EE	0A	AA	SN

0x00: Read / Write — UTC, UDC (Adjusted for local time); Local time offset from UTC

0x10: **UTC** Raw / Unadjusted; **UDC**; **Latitude** h0ddmm.mmmm, **H**emisphere: 0+N/8-S; **Longitude** hdddmm.mmmm, **H**emisphere: 0+E/8-W

0x20: **Altitude - M** saaaaaa.a, **S**ign: 0+/8-; **Geo Alt - M** saaaa.a, **S**ign: 0+/8-; **Speed - knot** ssss.ss; **Heading - Deg** 0hhh.hh; **MagVar - Deg** rvvv.vv, **R**ef: 0+E/8-W

0x30: **Dilution of Precision** **PDP** Position pp.pp, **HDP** Horizontal hh.hh, **VDP** Vertical vv.vv; **Firmware** 02 21 00, Rev 00 etc; **Local Time Offset** dlhhmmss, **d**st: 0 UTC/1 UTC+1, **l**ead/**l**ag: 0 UTC+/1 UTC-

0x50: P01 P02 P03 P04

0x60: P05 P06 P07 P08

0x70: P09 P10 P11 P12

00	01	02	03	04	05	Smart Nixie set index
hr	mn	se	mo	dy	yr	
hr	mn	se	01	DL	00	Set UTC offset (Below)
xx	xx	xx	02	xx	00	Turn DST off
xx	xx	xx	03	xx	00	Turn DST on
xx	xx	xx	08	xx	00	INT = 1PPS
xx	xx	xx	09	xx	00	INT = 1PP30S
xx	xx	xx	10	xx	00	INT = 1PPM
xx	xx	xx	11	xx	00	INT = 1PPH
xx	xx	xx	12	xx	00	INT = Off
All other combinations						[Transaction ignored]

DL	1x	Daylight savings time is off
	2x	Daylight savings time is on, UTC+1 hour
	x1	Local time is UTC + offset
	x2	Local time is UTC - offset
hr:mn:se		Offset from UTC in hours, minutes, seconds

Setting Local Time vs UTC Offset

This functionality allows setting the local time vs UTC offset and the interrupt frequency from a clock application, duplicating the functionality or writing directly to registers 0x3B to 0x3F.

When writing to the page 0x00 time registers, if the year is set to 00 and the month is set to 01 to 12 (the setting mode) the settings are updated according to the table to the left and stored in EEPROM after the the years register is written. DL is checked for valid input of 11, 12, 21 or 22 however hr:mn:se is not checked so non-BCD/non-Time values will give unpredictable results when local time is calculated.

EF	Error flags (Writing anly value to the error flags register clears both the flags and the counts)	0x40	RS232 bus overflow
		0x20	I²C bus OF/collision
		0x10	I²C bus lockup
		0x08	GPRMC checksum
		0x04	GPGSV checksum
		0x02	GPGSA checksum
		0x01	GPGGA checksum
EC	Error count (Clear via EF write)	0 - 255 (Count stop at 255)	
PS	INT prescaler factor Output high (~500mS) at change in second, minute or hour. Setting stored in EEPROM.	0x00	INT = 1PP Second
		0x01	INT = 1PP 30 Sec
		0x02	INT = 1PP Minute
		0x03	INT = 1PP Hour
		0x04	INT = Off
		Else	Off (Reset=1PPS)

P	Position status	0	Void
		1	Active
F	Fix mode	0	No fix
		1	2D
		2	3D
Q	Fix quality	0	No fix
		1	GPS
		2	DGPS (Assisted GPS)
		3	PPS
		4	RTK (Real Time Kinematic)
		5	Floating RTK
		6	Estimated (Dead reckoning)
		7	Manual input mode
		8	Simulated
D	2D/3D fix selection mode	1	Auto
		2	Manual
VV	Satellites currently in view		
SS	Satellites used to compute fix, up to 12		
Pnn	PRNs of satellites used to compute fix		
EE	Satellite elevation, 00-90 degrees		
0AAA	Satellite azimuth, 0000-0360 degrees		
SN	Signal/Noise ratio, 00-99dB (higher is better)		

Table 4. I²C Register Map

An I²C master communicates with the TES1375 module configured as a slave via a 128 byte register stack using standard I²C commands at a maximum data rate of 100kbps. See Table 4 for a complete list of the slave stack register functions.

For register writes a Start command is issued followed by the slave address where the R/W bit is cleared to zero. The next byte written is the pointer to the desired element in the register stack from 0x00 to 0x7F. The following bytes written are the register data and after the last register data byte has been sent, a Stop command is issued which completes the transaction. Each write by the master is followed by a low ACK bit generated by the slave indicating an acknowledgement of the transaction, where a high NAK bit generally indicates no slave at the specified address is present. A single write consisting of the pointer byte alone sets the pointer for the next read transaction.

After each write the internal pointer is incremented allowing from 0 to 128 consecutive registers to be written in the same transaction. Writes beyond the last writeable register position (0x7F) are ignored. Read only registers (The majority of registers in this device) are not affected by being written to, while other registers such as the PCF time group are acted on indirectly.

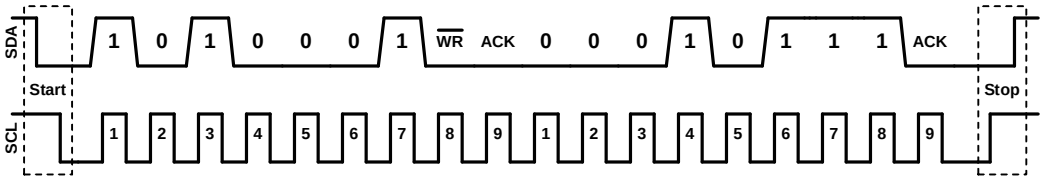


Figure 8a: Setting the pointer to 0x17 (For Latitude register read)

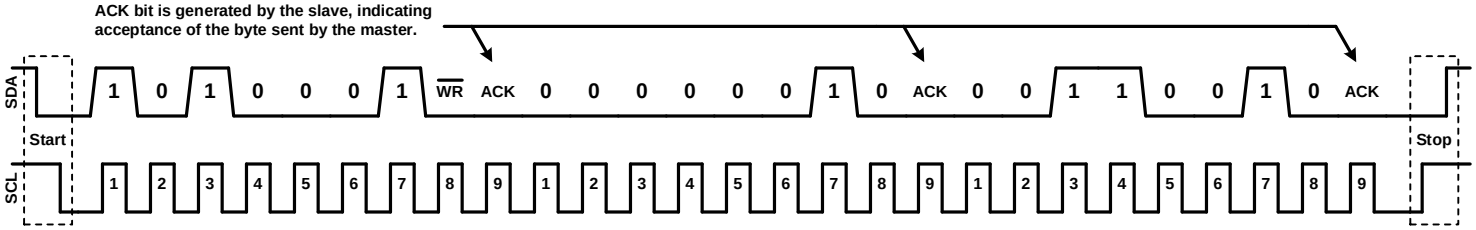


Figure 8b: Write to Register 0x02 with Data 0x32 - Seconds = "32"

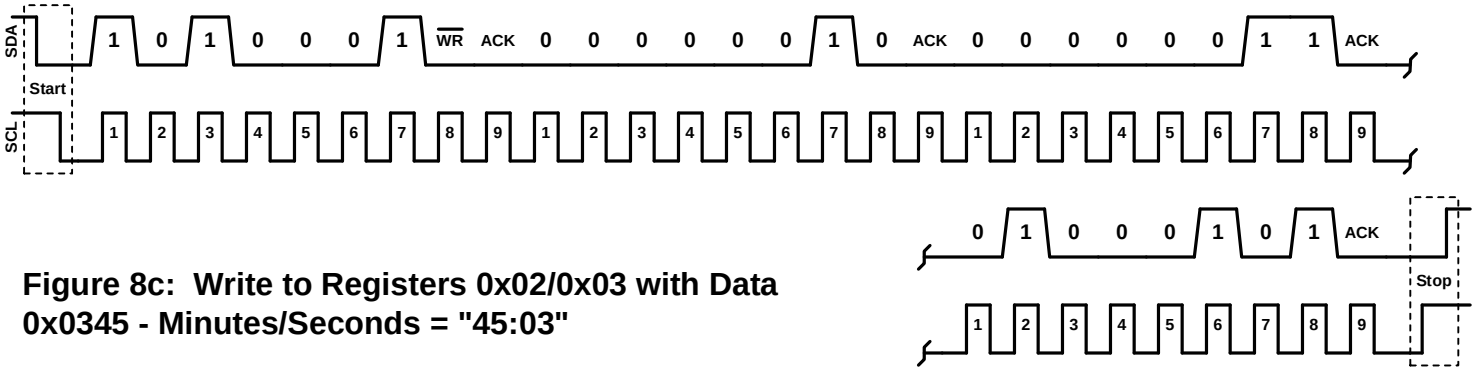


Figure 8c: Write to Registers 0x02/0x03 with Data 0x0345 - Minutes/Seconds = "45:03"

For register reads a Start command is issued followed by the slave address where the R/W bit is set to one. The first byte read is at the current pointer address. This address can be set by a single write, the pointer resulting from the last series of writes, i.e. the last byte written plus one, or the pointer resulting from the last series of reads, i.e. the last byte read plus one. Reads past the top of the register stack at 0x7F return zeros. After each read except the last, the master issues an ACK indicating the data was received. After the last read of a transaction, the master issues a NAK indicating that the data was received and no more data will be read. Following the NAK, the master issues a Stop command to terminate the transaction.

The master may perform a pointer write followed by a Stop command and then a Start command for the read sequence, or a repeated start may be used between the write and read sequences.

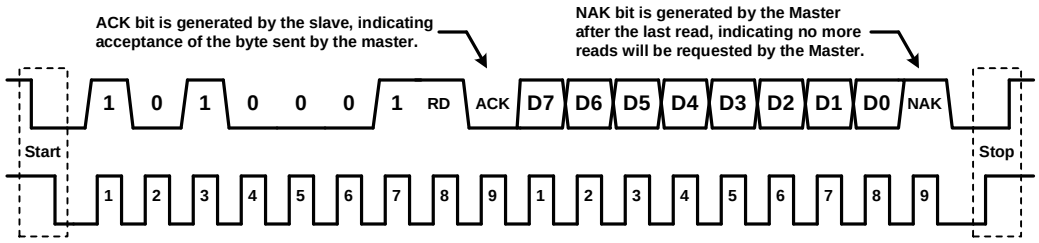


Figure 9a: One Byte Read Sequence from the current pointer

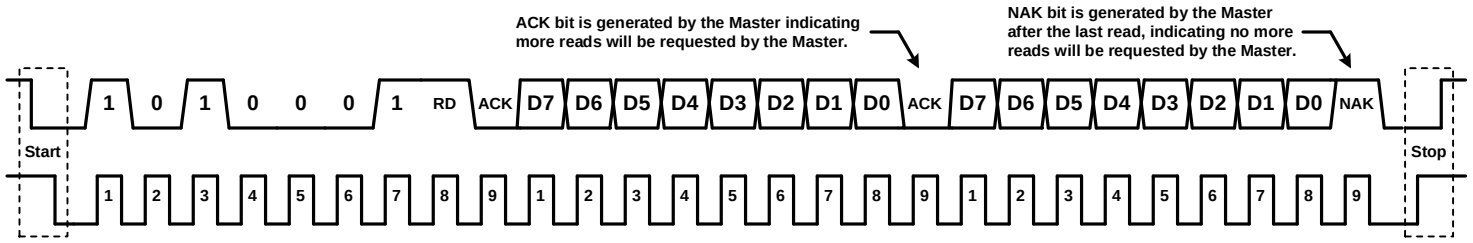


Figure 9b: Multi Byte Read Sequence from at the current pointer

```
//Figure 8a
I2C_START();
I2C_WRITE(0xA2);        //slave address, R/W set low
I2C_WRITE(0x17);        //set pointer to latitude start
I2C_STOP();

//Figure 8b
I2C_START();
I2C_WRITE(0xA2);        //slave address, R/W set low
I2C_WRITE(0x02);        //set pointer to seconds register
I2C_WRITE(0x32);        //write "32" to seconds register
I2C_STOP();

//Figure 8c
I2C_START();
I2C_WRITE(0xA2);        //slave address, R/W set low
I2C_WRITE(0x02);        //set pointer to seconds register
I2C_WRITE(0x03);        //write "03" to seconds register
I2C_WRITE(0x45);        //write "45" to minutes register
I2C_STOP();

//Figure 9a
I2C_START();
I2C_WRITE(0xA2);        //slave address, R/W set low
I2C_WRITE(0x17);        //set pointer to latitude start
I2C_START();            //repeat start
I2C_WRITE(0xA3);        //slave address, R/W set high
data1=I2C_READ(1);      //read register 0x17
I2C_STOP();             //last read is always (1)=NAK

//Figure 9b
I2C_START();
I2C_WRITE(0xA2);        //slave address, R/W set low
I2C_WRITE(0x17);        //set pointer to latitude start
I2C_START();            //repeat start
I2C_WRITE(0xA3);        //slave address, R/W set high
data1=I2C_READ(0);      //read register 0x17
data2=I2C_READ(1);      //read register 0x18
I2C_STOP();             //last read is always (1)=NAK

//Alternate code for Figure 9b
I2C_START();
I2C_WRITE(0xA2);        //slave address, R/W set low
I2C_WRITE(0x17);        //set pointer to latitude start
I2C_STOP();
I2C_START();
I2C_WRITE(0xA3);        //slave address, R/W set high
data1=I2C_READ(0);      //read register 0x17
data2=I2C_READ(1);      //read register 0x18
I2C_STOP();
```

Figure 10: Example coding (CCS PICC)

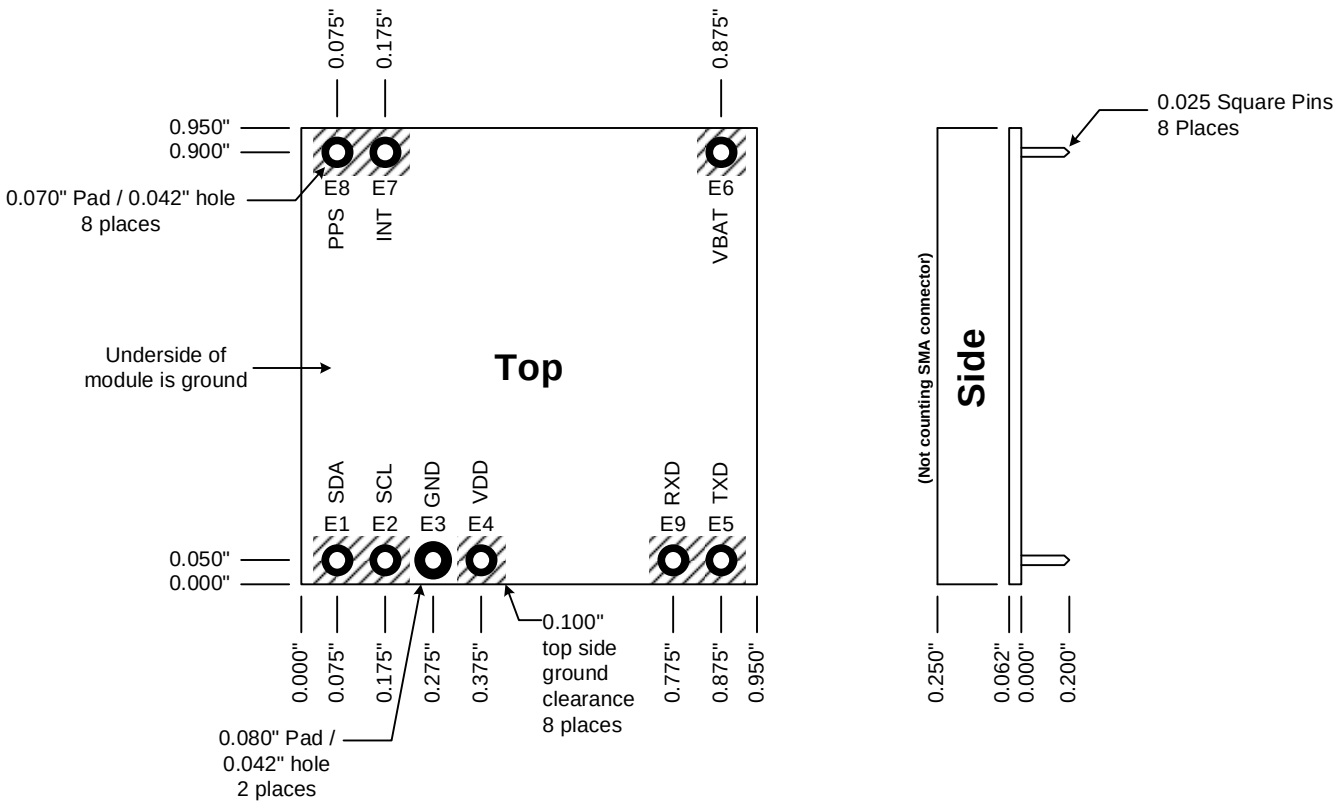
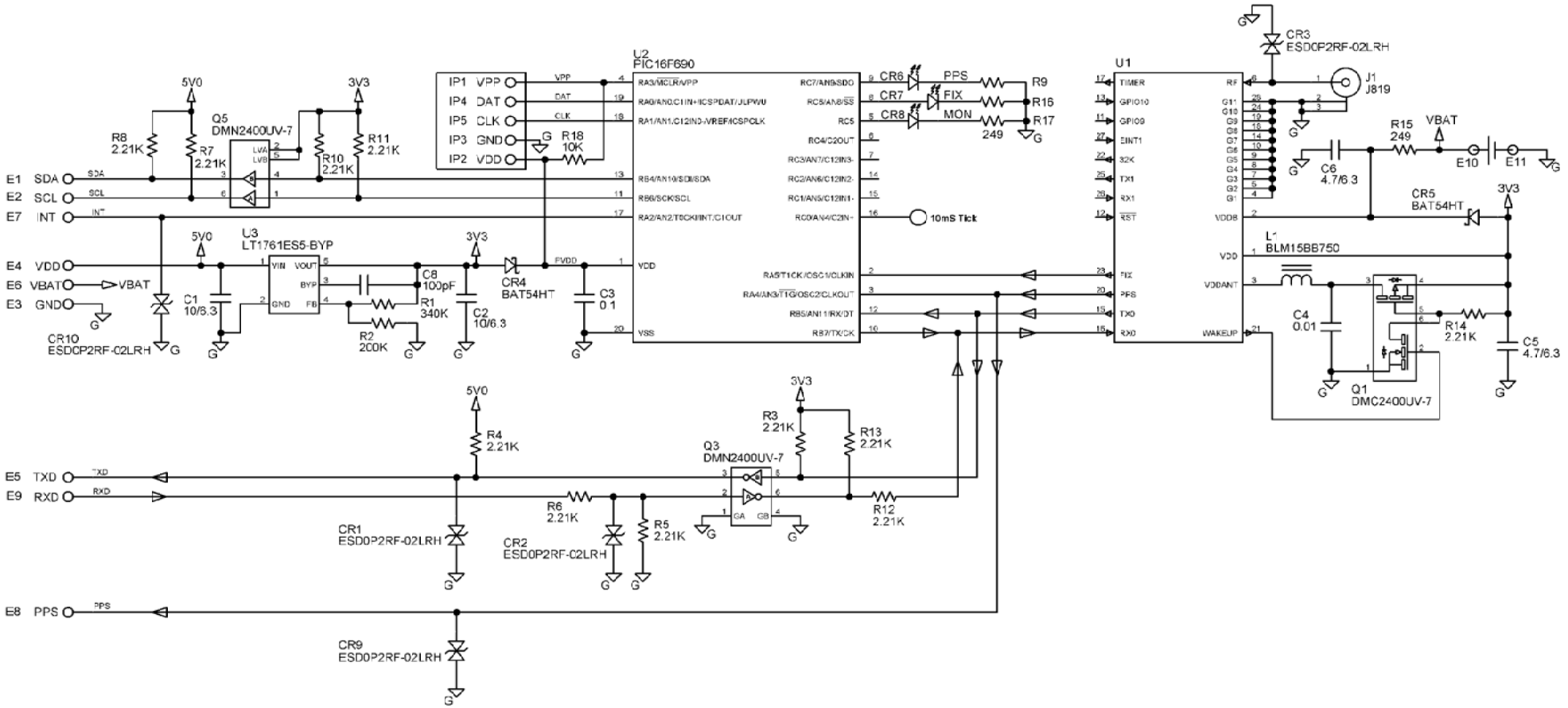


Fig 11. 1375 module outline and recommended PCB layout

NOTES: UNLESS OTHERWISE SPECIFIED

REVISION HISTORY			
REV	DESCRIPTION	APPROVED	DATE



APPROVALS	DATE
DRAWN J TAYLOR	2012/06/01
CHECKED	
ENGINEER	
PROJ ENGINEER	
APPROVED	
QA	

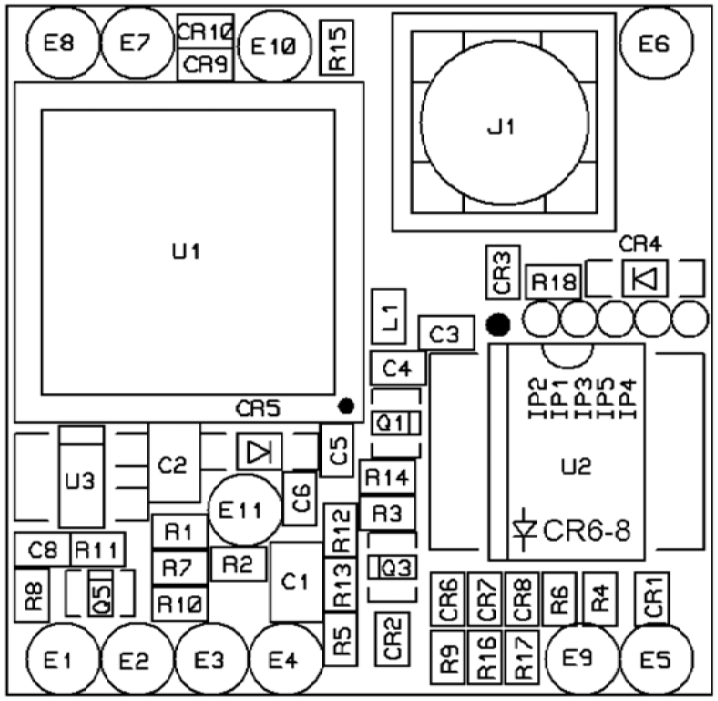
TITLE: SmartNixie GPS Real Time Clock

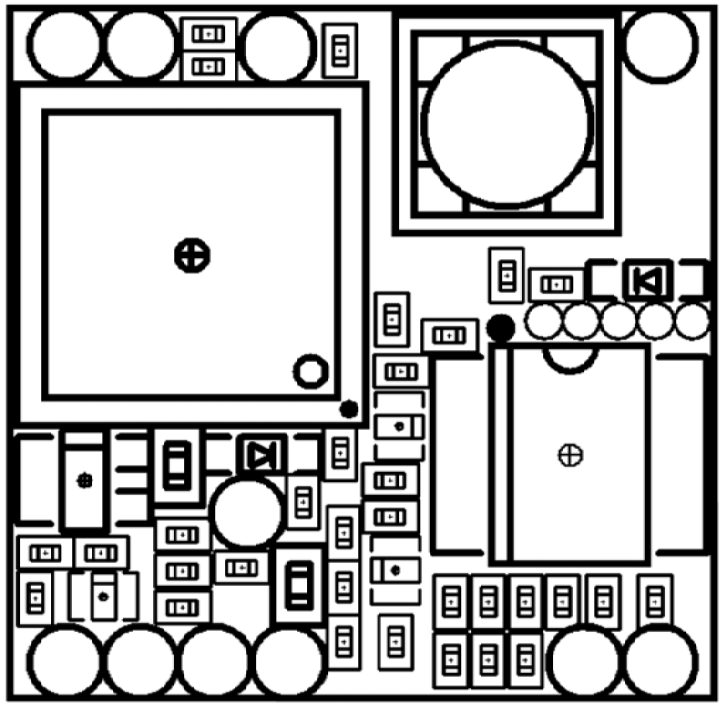
SIZE	FSCM NO.	DRAWING NO.	REV
B		1375B.DSN	B

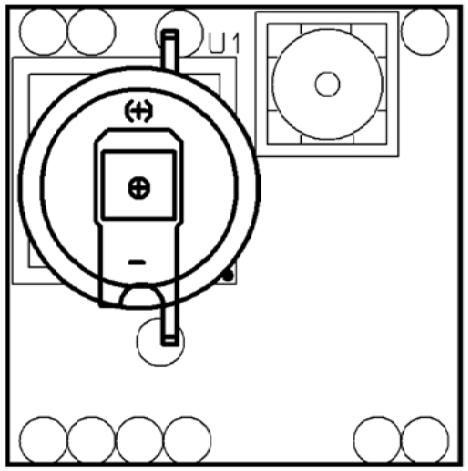
SCALE	© 2012 TES	MODIFIED: 09/02/12 12:09:12 PM	SHEET 1 OF 2
NONE			

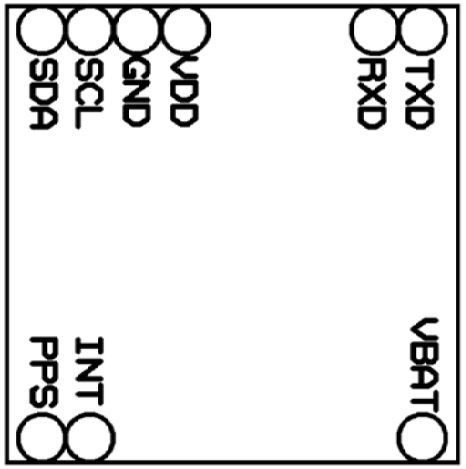
NOTES: UNLESS OTHERWISE SPECIFIED

REVISION HISTORY			
REV	DESCRIPTION	APPROVED	DATE









APPROVALS	DATE	TITLE: SmartNixie GPS Real Time Clock			
DRAWN J TAYLOR	2012/06/01				
CHECKED					
ENGINEER					
PROJ ENGINEER		SIZE B	FSCM NO.	DRAWING NO. 1375B.DSN	REV B
APPROVED		SCALE NONE	© 2012 TES	MODIFIED: 06/02/12 12:20:21 PM	SHEET 2 OF 2
QA					