Chronvertor (v2.1)

User Manual *(For Firmware v2.0)*

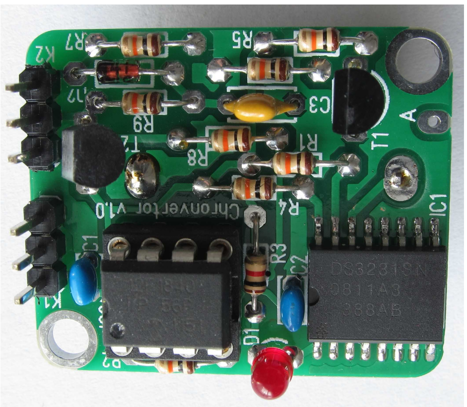
Unusual Electronics

Disclaimer

Unusual Electronics reserves the right to change product specifications without notice.

We shall **NOT BE LIABLE** in any way to you or to any other person, firm or corporation whatsoever for any loss, costs, liability, damages, injury, accident, expense, delay or inconvenience which might occur from the use of this product.
This product is intended for educational or demonstration purposes only.
It is not intended for use in any commercial applications.
If used in such applications the purchaser assumes all responsibility for ensuring compliance with local laws.
It is not suitable for use in medical systems or anywhere that might create a hazardous situation of any kind.

Contents



1. Introduction

The “Chronvertor” is a module containing a very accurate battery backed Maxim DS3231 RTC (Real Time Clock) chip working with a Microchip PIC Microcontroller to output the time and date as either a GPS NMEA datagram or a WWVB, DCF77, MSF or JJY radio time protocol.
It can also be used to generate the periodic impulses needed to step or sync electro-mechanical slave clocks.
It automatically takes care of DST (Daylight Saving Time) changes around the world and provides time-zone offsets to generate any local time.

The RTC's time and date can be set-up by either connecting to an external GPS module for a few seconds or by commands sent by a serial terminal program using a PC.

2. Features

- The Chronvertor module simulates any one of the following serial time protocols:
    - GPS NMEA Time datagram ($GPRMC)
    - WWVB (USA) Radio time protocol.
    - MSF (UK) Radio time protocol.
    - DCF77 (Germany) Radio time protocol.
    - JJY (Japan) Radio time protocol.
- Carrier wave (60,77.5 & 40kHz) generation options for radio protocols.
- Automatic DST (Daylight Saving Time) adjustment for most countries and custom rules
- World Time Zone offset settings allow any local time to be generated
- Automatic Leap Year adjustment
- The module automatically outputs the correct weekday code for radio time protocols
- **New** "MasterPulse" impulses can be generated at periodic intervals to step or synchronise mechanical "Slave" clocks. (external driver interface required).
- Very accurate timekeeping. (within ± 5.4 seconds per month or better)
    - Timekeeping fine adjustment can improve accuracy and compensate for crystal "aging"
    - Each module is tested and fine-calibrated after construction to improve on the published RTC timekeeping accuracy
    - Time is battery-backed with a long life lithium CR2032 coin-cell.
- The module time/date can be very accurately set by syncing it with an external GPS *$GPRMC data connection.*
- Time/date can also be set by serial commands from a PC terminal so that you can set any time or date in the current century.
- Small size with low power consumption making it ideal for replacing existing radio or GPS time sources.
- Could be used for testing or synchronising radio controlled clocks where they cannot receive the appropriate radio signal.
- All configuration settings are safely stored in the microcontrollers' EEPROM memory.
- LED can indicate output pulses or valid GPS sentences being received.
- Configuration settings can be reset back to factory defaults if you wish.

3. Backup Battery (CR2032) Fitting/Replacement

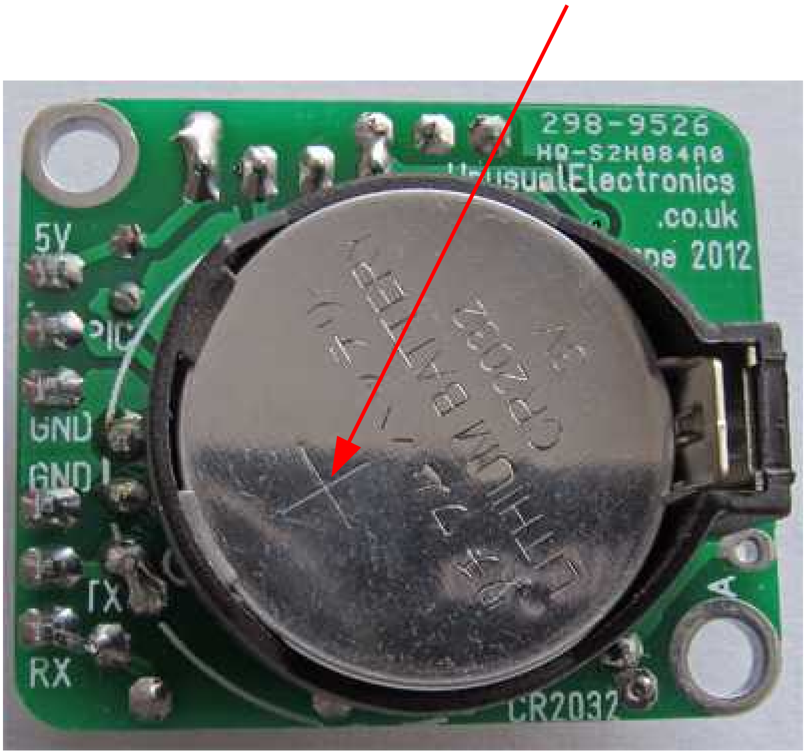
WARNING

- Danger of explosion if battery is incorrectly replaced.
- Be sure to observe the correct polarity when installing the battery
- Do not short-circuit the battery.
- Replace the battery only with a CR2032 lithium battery.
- Use of another battery may present a risk of fire or explosion.
- Battery may explode if mistreated. Do not recharge, disassemble or dispose of in fire.
- Do not expose the battery to excessive heat.
- Dispose of used battery promptly and in accordance with local waste disposal policies.
- Keep away from children.

A **CR2032** Lithium Coin Cell is required for timekeeping backup.

(The module is supplied without a battery to comply with postal regulations)

Insert the battery so that it clips into place with the "**+**" marker visible as shown below.



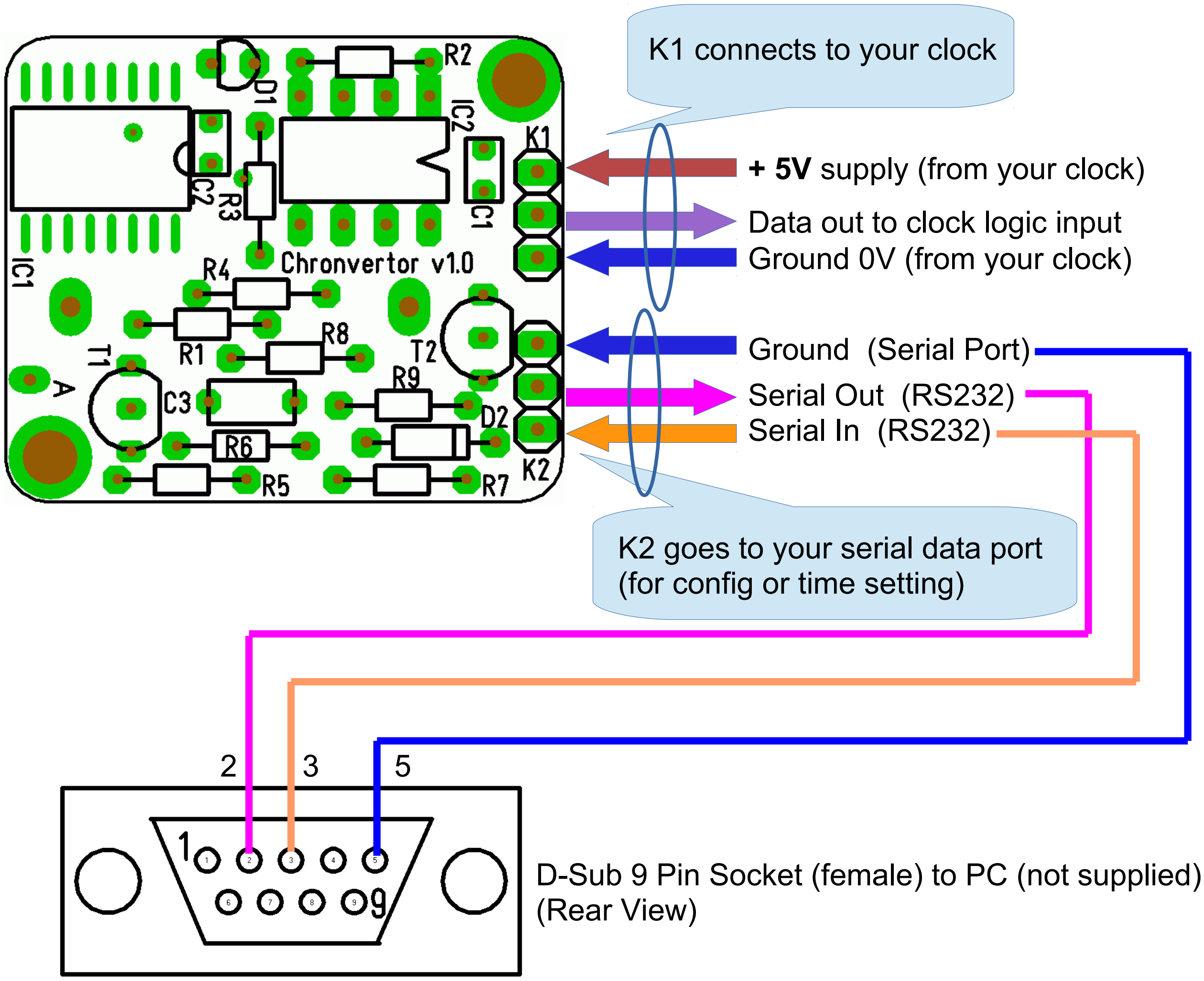
When using the module, be careful to ensure that the exposed battery surface can not come into contact with any casing/metalwork/wiring etc.

4. Connections

Connect the three connections of K1 to your clock as shown below.
Your clocks' 5 Volt supply and ground connections are used to power the module.

(Make sure the supply voltage is correct before connecting the module as it will be damaged by over-voltage or incorrect supply polarity.)

The data out connection to your clock uses standard logic levels (5 Volts) and should only be connected to a compatible logic input of your clock.

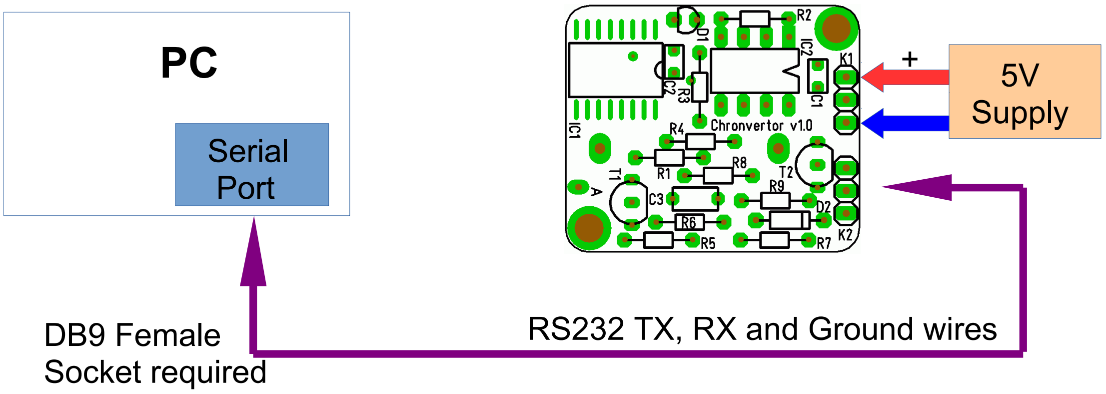


K2 connector is used for connecting the module to an RS232 serial port of a computer or GPS module.
This is needed for changing the module configuration settings or setting the time/date.

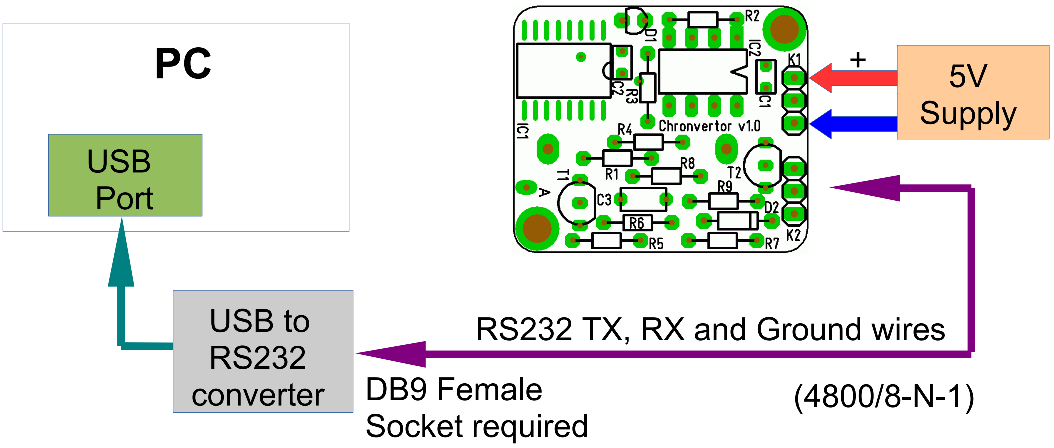
When power is first applied to the module, The Red LED should light for a few seconds to indicate successful start-up.

Connection options for configuration or time & date setting

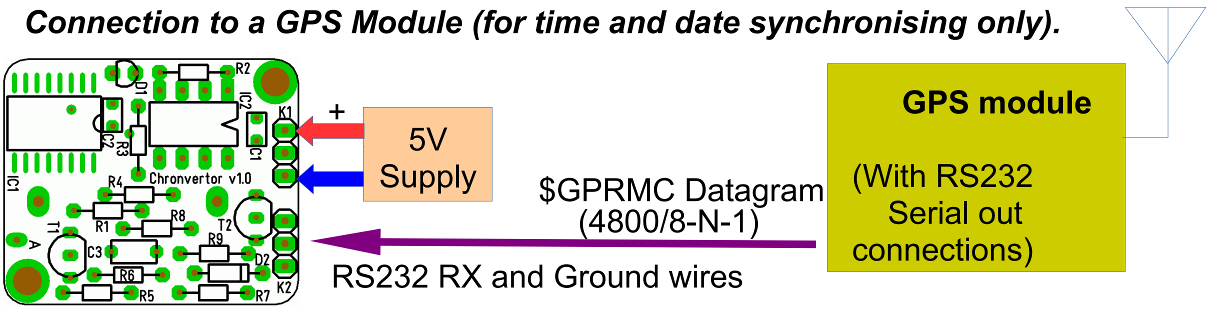
- ***Connection direct to a PC serial port. (4800/8-N-1)***



- ***Connection via a PC USB port.***



- ***Connection to a GPS Module (for time and date synchronising only).***



(More accurate synchronisation may be achieved if the GPS module sends a $GPGGA datagram each second in addition to the required $GPRMC datagram)

5. Module Configuration Set-up

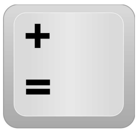
Before configuration, your module must be connected to a 5V power supply via connector K1 (either from your clock or another 5V regulated supply.)

K2 Must be connected to your computer serial port as described in the previous section. (If your computer doesn't have a serial port, there are USB → serial adapters available at low cost from places such as eBay)

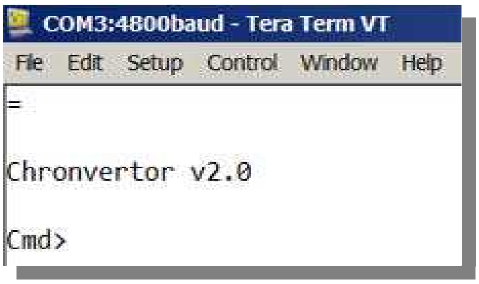
- Open a serial comms program such as HyperTerminal (Windows XP or earlier) or download a freeware program such as TeraTerm if you have Windows 7 or Vista.

- Make sure the baud-rate is set to 4800/8-N-1 and the Com port setting matches the serial port socket setting on your PC.

- Press the "Equal" = key followed by the "Enter" ↵ key to start the config.



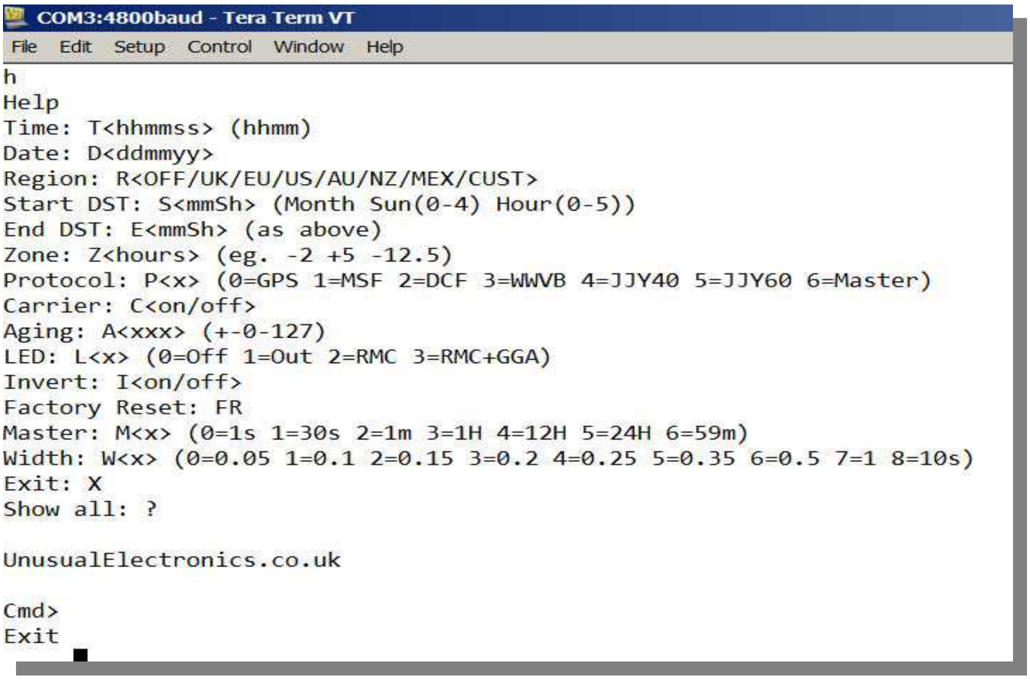
- The module should respond with "Chronvertor" followed by the firmware version. As shown below:



On the next line it shows "Cmd>" to indicate it is now ready to receive your commands. (see next section)

Help

Press "**H**" followed by ↵ to see **H**elp with all the commands and their syntax:



```
COM3:4800baud - Tera Term VT
File  Edit  Setup  Control  Window  Help
h
Help
Time: T<hhmmss> (hhmm)
Date: D<ddmmyy>
Region: R<OFF/UK/EU/US/AU/NZ/MEX/CUST>
Start DST: S<mmSh> (Month Sun(0-4) Hour(0-5))
End DST: E<mmSh> (as above)
Zone: Z<hours> (eg. -2 +5 -12.5)
Protocol: P<x> (0=GPS 1=MSF 2=DCF 3=WWVB 4=JJY40 5=JJY60 6=Master)
Carrier: C<on/off>
Aging: A<xxx> (+-0-127)
LED: L<x> (0=Off 1=Out 2=RMC 3=RMC+GGA)
Invert: I<on/off>
Factory Reset: FR
Master: M<x> (0=1s 1=30s 2=1m 3=1H 4=12H 5=24H 6=59m)
Width: W<x> (0=0.05 1=0.1 2=0.15 3=0.2 4=0.25 5=0.35 6=0.5 7=1 8=10s)
Exit: X
Show all: ?

UnusualElectronics.co.uk

Cmd>
Exit
```

Time

Press **T** followed by *hhmmss* or *hhmm* then ↵ (colon separators can also be used)

hh=hours (00-23) *mm*=minutes *ss*=seconds (if seconds are omitted they default to 00)

Note. if **T** is pressed on it's own, it will just show the current time.

(Time should normally be set to UTC/GMT so that DST and the timezone can be setup.)

(The symbol "**+**" after the time appears if DST is currently in effect.)

Date

Press **D** followed by *ddmmyy* (or *ddmm* (for current year)) then ↵

dd=day *mm*=month *yy*=year (/ separators can also be used eg. dd/mm/yy)

Note. if **D** is pressed on it's own, it will just show the current date.

DST Region (Daylight Saving Time)

This option is used for automatically changing DST according to the rules for your country.

Press **R** followed by the region code then ↵

OFF = DST is not enabled.

UK = United Kingdom   US = United States   EU = Europe

AU = Australia   NZ = New Zealand   MEX = Mexico

CUST = Custom configuration (this option is for other countries or if rules change)

DST Start (Custom Rule)

This option is only used if the DST region is set to **CUST** (Custom)
To set the DST **Start** rule:

Press **S** followed by *mmSh* then ↵

mm=Month, *S*=Sunday in the month (0-4 (0=Last Sunday)), *h*=hour (0-5)

(Pressing **S** ↵ shows the current custom or pre-set region dst start rule settings)

DST End (Custom Rule)

This option is only used if the DST region is set to **CUST** (Custom)
To set the DST **End** rule:

Press **E** followed by *mmSh* then ↵

mm=Month, *S*=Sunday in the month (0-4 (0=Last Sunday)), *h*=hour (0-5)

(Pressing **E** ↵ shows the current custom or pre-set region dst end rule settings)

TimeZone Offset

This option is used for applying a timezone offset for your country if the module is set to UTC time. (Coordinated Universal Time).

Note. The timezone is automatically set for radio time protocols (e.g. JJY = +9 hours). You can manually override by setting the timezone ***after*** setting the protocol.

To set the time zone:
Press **Z** followed by the number of hours then ↵

e.g. **Z5** = plus 5 hours   **Z-4** = minus 4 hours   **Z-12.5** = minus 12.5 hours

Protocol

This sets the time output protocol
(Note. Setting the protocol may also affect the timezone setting)

To set the protocol:
Press **P***x* then ↵ (*x*=0→5)

0 = GPS (NMEA 0183 $GPRMC satellite navigation datagram)
1 = MSF (Time & Date code broadcast from UK)
2 = DCF77 (Time & Date code broadcast from Germany)
3 = WWVB (Time & Date code broadcast from USA)
4 = JJY40 (40kHz Time & Date code broadcast from Japan)
5 = JJY60 (60kHz Time & Date code broadcast from Japan)
(Japan has two transmitters at different frequencies which use the same code)
JJY options 4 & 5 are identical unless the Carrier wave option is enabled)
6= "MasterPulse" mode (The module sends impulses at pre-set intervals)

(see the protocols section for more info)

Carrier

This turns on/off a 60Khz, 77.5Khz or 40Khz carrier wave applicable to the MSF/DCF77/WWVB/JJY protocols. (not used by GPS or MasterPulse)

Press **Con** or **Coff** then ↵

This feature may only be used for sending a very low power signal to a clock containing a radio receiver placed very close to the module.
(do not use if your clock is directly connected to the module).

Aging

This is used for adjusting the RTC timekeeping accuracy which can vary with age.

It may also be used to improve on the initial manufacturer's calibration of +/-2ppm (+/-5 seconds/month)

Press **A***xxx* then ↵ (*xxx*= +1 to +127 decrease the clock speed)

0=No Compensation (*xxx*= -1 to -127 increase the clock speed)

Each increment = 0.05ppm approx. (about 1.6 sec/year)

The module is supplied calibrated to approx +/- 0.2ppm

(The calibrated value is pre-set and labelled on the module)

Invert Output

This Inverts the MSF/DCF77/WWVB/JJY/GPS or Masterpulse output logic protocols when set to "On"

Press **Ion** or **Ioff** then ↵

This allows the module to be used with clocks that require an inverted logic input.

(some Nixie clocks have an inverted logic input. Using this option allows the Chronvertor to be connected without the need for a separate logic inverter or other modifications)

LED Function

The module LED can flash to show data being sent from or received by the module. Options 2&3 can be used to indicate valid GPS datagram sentences being received.

Press **L***x* then ↵

0=Off (This turns the LED off to reduce power consumption)

1=Output (LED flashes to indicate data being sent from the module)

2=RMC (Indicate a valid GPRMC sentence being received from a GPS module)

(An RMC sentence is required for the module to sync time with a GPS module)

(RMC is deemed valid if the "Validity" char is "A" and checksum is OK)

3=RMC&GGA (indicates both valid RMC & GGA sentences are being detected.

(If a GGA sentence is received in addition to the required RMC, it can improve the time sync accuracy).

Factory Reset

To reset all settings back to the originally supplied defaults.

Press **FR** then ↵

All configuration options including the RTC clock time/date will be reset to defaults.

MasterPulse (Firmware v2.0 onwards)

(Requires the **P**rotocol to be set to option **6** (Masterpulse))

This mode is intended for use with antique electro-mechanical clocks that either require a regular impulse to drive them (options 0-2) or a periodic pulse to re-sync the dial (options 3-6)

Press **M**x then ↩ (x=0→6)

0 = 1 second (pulse every second)

1 = 30 seconds (every 30 seconds)

2 = 1 Minute (every minute)

3 = 1 Hour (every hour)

4 = 12 Hours (every 12 hours at 00:00 and 12:00)

5 = 24 hours (every 24 hours at 00:00)

6 = Minute59 (a special pulse at minute 59, always with a width of 60 seconds)

Notes

DST Changes can be automatically applied for options **0 – 2**

(At DST, Option 0 advances the clock 1 hour, using pulses at 0.5 sec intervals.

Options 1&2 advance 1 hour with pulses at 2 sec intervals.)

At DST end, the impulses will stop for 1 hour (options 0 – 2 only).

Sending pulses manually (to advance slave clocks to the required time)

To be able to advance clocks manually, First enable Masterpulse mode (P6).

Then press **V** ↩ and the terminal screen prompt will change to **>>>**

Now every ↩ press will send an impulse (you can press ↩ repeatedly until you have sent enough pulses, then press any other key followed by ↩ to return to normal.

Physical Interface to electro-mechanical clocks

Do not connect this module directly to any electro-mechanical clock.

The module's low power logic output will need a separate interface/isolator to adapt to the voltage/current pulse requirements of your slave clock.

Due to the variety of slave clock systems that are/were used around the world, we cannot provide any support or accept liability for interfacing to them.

Width (MP) (Firmware v2.0 onwards)

This option configures the pulse width for **MasterPulse** mode only.

Press **W**x then ↵ (x=0→8)

0 = 0.05 sec (50ms)

1 = 0.1 sec

2 = 0.15 sec

3 = 0.2 sec

4 = 0.25 sec

5 = 0.35 sec

6 = 0.5 sec

7 = 1 sec (do not use this option with MasterPulse option 0 (1 sec))

8 = 10 sec (do not use this option with MasterPulse option 0 (1 sec))

Notes

Choose the shortest pulse width that reliably operates your mechanical slave clock.

The default is active pulse low – you can change this using the Invert (**I**) option.

The pulse width during any auto DST one hour advance overrides to limit it to 1 second Max (for the 30sec or 1min pulse modes) or 0.2 sec Max (for the 1 second pulse mode.)

Masterpulse option 6 (min59 mode) overrides to always be a 60 sec pulse.

Exit

Exit from the configuration. (any changes are saved)

Press **X** then **Enter**

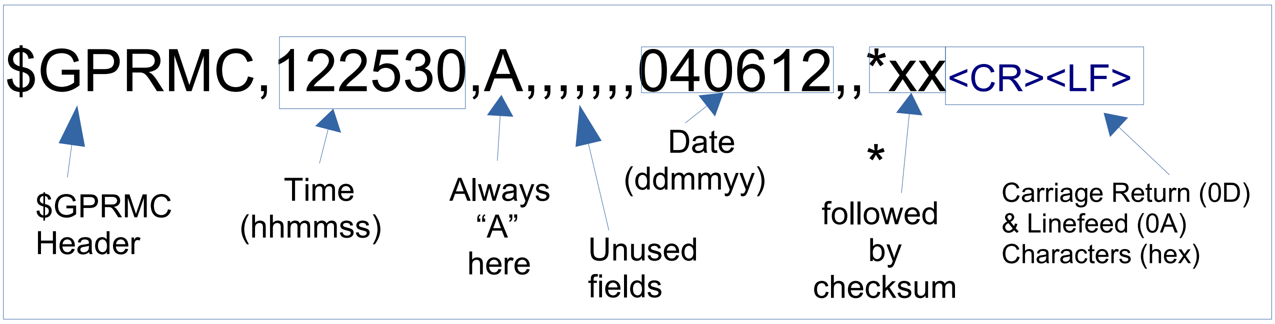
7. GPS Datagram Protocol

The module outputs a simulated **$GPRMC** datagram (NMEA 0183 standard) at the start of each second, (only the time and date are sent).

(Serial data Baud-rate is 4800, 8Bits, No parity, 1 Stop Bit.)

The serial data output is sent at logic levels only, and must be connected to the serial **LOGIC** level input of your clock.

An example of the format of the ASCII datagram sent by the module is as shown below:



The checksum is a two-character hexadecimal number. (It is the XOR of all the bytes between the **$** and the ***** in the sentence).

The commas are used as data separators. A real GPS string would also contain navigation data such as latitude, longitude, speed etc. between the commas.
Data can be omitted as long as all the commas are still present to mark each section of the string. (The RMC string should contain 10 commas).

Notes.
Your datagram string parsing software should count the commas in the sentence to locate the time and date.

The checksum is provided to increase transmission reliability. It is optional for your software to check it although is advisable if the module is fitted some distance from your clock or near any interference.

The time is in 24 Hour format and is usually set to UTC unless you have enabled DST or Time Zone options.

More information at:

[Wikipedia]

8. MSF **UK** Radio Time Protocol Information

The module can simulate the MSF time and date code which is broadcast from a 60kHz radio transmitter at Anthorn Radio Station in Cumbria, UK

The full sequence of information is transmitted in binary coded decimal format each minute, using two data bits during every second of the minute except the first.

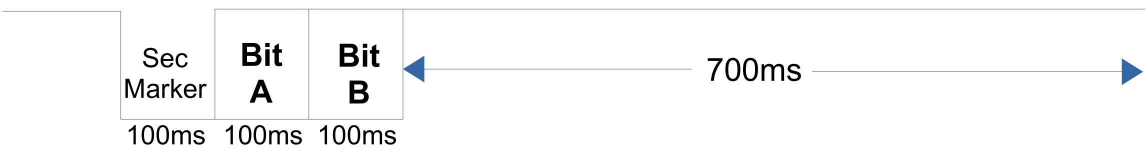
The time code includes the following information:

- year
- month
- day of month
- day of week
- hour
- minute
- British Summer Time (in effect or imminent)

The broadcast time is UTC during winter and UTC+1 during summer (DST).

The first second of each minute (Minute Marker) begins with a 500ms logic low (1) followed by 500ms high (0). (The bit polarity is: High=0 Low=1)

Each normal second begins with at least 100ms low (second marker) followed by bit “A” (100ms) then bit “B” (100ms).



(The bit polarity is: High=logic 0 Low=logic1)
The rest of the second (700ms) is always high.

The module normally simulates the time-code using TTL/CMOS logic levels without any carrier wave so that it can be directly connected to the logic decoder input of your clock.

There is an option to enable a 60kHz (approx) very low power carrier wave.

A suitable 60kHz coil via a current limiting resistor (1kΩ) could be connected between the module data output and ground for testing a clock with a built-in MSF radio receiver over a short distance (a few inches only).

This may require the clock being tested to be screened in some way to prevent it receiving the real MSF broadcast signal.

MSF Time & Date Code

Sec	A	B	INFO	Sec	A	B	INFO	Sec	A	B	INFO
00	1	1	Minute mark	**28**	2	0	Month (01-12)	**44**	1	0	Hour (00-23)
Seconds **01-16** (the "DUT" Atomic-Astronomical time difference) are not able to be simulated. **(A & B are both = 0)**				**29**	1	0		**45**	40	0	Minute (00-59)
				30	20	0	Day of Month (01-31)	**46**	20	0	
				31	10	0		**47**	10	0	
				32	8	0		**48**	8	0	
17	80	0	Year (00-99)	**33**	4	0		**49**	4	0	
18	40	0		**34**	2	0		**50**	2	0	
19	20	0		**35**	1	0		**51**	1	0	
20	10	0		**36**	4	0	Day of Week (Sun=0 Sat=6)	**52**	0	0	Minute Identifier (Start)
21	8	0		**37**	2	0		**53**	1	STW	Summer Time Warning
22	4	0		**38**	1	0		**54**	1	P1	Parity A17-24 (Year)
23	2	0		**39**	20	0	Hour (00-23)	**55**	1	P2	Parity A25-35 (Month&Day)
24	1	0		**40**	10	0		**56**	1	P3	Parity A36-38 (DOW)
25	10	0	Month (01-12)	**41**	8	0		**57**	1	P4	Parity A39-51 (Time)
26	8	0		**42**	4	0		**58**	1	ST	Summer Time
27	4	0		**43**	2	0		**59**	0	0	Minute Identifier (End)

- The time and date is sent for the *following* minute. (e.g. during 01:25 the time is being sent for 01:26)
- The BCD time code is sent with the ***most*** significant bits first.
- Second **00** (Minute Mark) is always 500ms Low (1) followed by 500ms High (0)
- Seconds **01-16** (DUT) data is not simulated by the module (Bits A & B) are set to 0
- Seconds **17-51** contain the Date & Time in Binary Coded Decimal (BCD)
- Bit "B" is always 0 (High) for seconds **17-51** (Date & Time)
- Seconds **52-59** create a minute identifier sequence where Bit "A" = 01111110
- Bit **B53** (STW) is set to 1 for one hour before the start & end of summer time.
- The time code is UTC during winter and UTC+1 during Summer time
- Second bits **B54-57** (Parity Bits) create "Odd Parity" (odd number of ones) when combined with their designated data bits (e.g. P1=Year)
- Bit **B58** (ST) is set to 1 during Summer Time.

More information can be found at:

National Physical Laboratory

Wikipedia

9. DCF77 **Germany** Radio Time Protocol Information

The DCF77 time and date code is broadcast from a 77.5 kHz radio transmitter at Mainflingen, about 25 km south-east of Frankfurt, Germany.
The transmitter is much more powerful than the UK MSF signal and can be received over most of Europe including the UK.

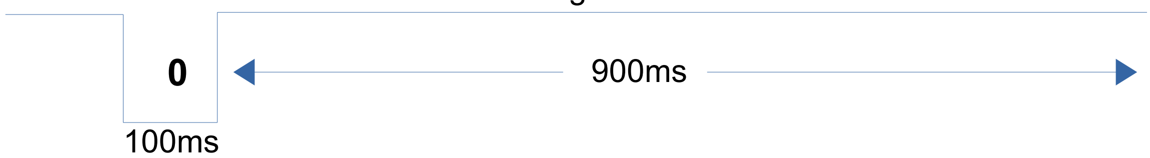
The full sequence of information is transmitted in binary coded decimal format each minute.

The time code includes the following information:

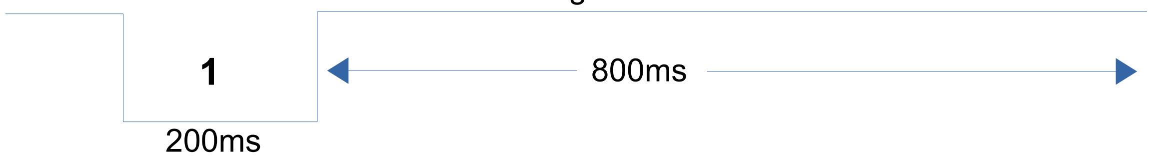
- year
- month
- day of month
- day of week
- hour
- minute
- Central European Summer Time (& time change warning)

The broadcast time is UTC+1 during winter and UTC+2 during summer (DST)

A 100ms low at the start of a second = logic 0



A 200ms low at the start of a second = logic 1



The rest of the second (800ms or 900ms) is high.

The module normally simulates the time-code using TTL/CMOS logic levels without any carrier wave so that it can be directly connected to the logic decoder input of your clock.

There is an option to enable a 77.5Khz (approx) very low power carrier wave. A suitable coil via a current limiting resistor (1kΩ) could be connected between the module data output and ground for testing a clock with a built-in DCF radio receiver over a short distance (a few inches only).
This may require the clock being tested to be metallically screened in some way to prevent it receiving the real DCF broadcast signal.

DCF77 Time & Date Code

Sec	Bit	Info	Sec	Bit	Info	Sec	Bit	Info
00	**0**	**Min Start (always=0)**	**26**	20	**Minute** (00-59)	**43**	2	**Day of Week** (Mon=1 Sun=7)
Seconds **01-15** (Weather, Civil Warning and Backup antenna information) can not be simulated. **All bits = 0**			**27**	40		**44**	4	
			28	P1	Parity (Minute)	**45**	1	**Month** (01-12)
			29	1	**Hour** (00-23)	**46**	2	
			30	2		**47**	4	
			31	4		**48**	8	
			32	8		**49**	10	
16	STW	Summer Time Warning	**33**	10		**50**	1	**Year** (00-99)
17	CEST	Summer Time	**34**	20		**51**	2	
18	CET	Winter Time	**35**	P2	Parity (Hour)	**52**	4	
19	**0**	Leap Second (not used)	**36**	1	**Day of Month** (01-31)	**53**	8	
20	**1**	Time Start (always=1)	**37**	2		**54**	10	
21	1	**Minute** (00-59)	**38**	4		**55**	20	
22	2		**39**	8		**56**	40	
23	4		**40**	10		**57**	80	
24	8		**41**	20		**58**	P3	Parity (Date)
25	10		**42**	1	**Day of Week**	**59**	–	Minute Mark (no bit)

- The time and date is sent for the *following* minute. (e.g. during 01:25 the time is being sent for 01:26)
- The BCD time code is sent with the ***least*** significant bits first.
- Second **00** (Minute Start) bit is always 0
- Seconds **01-15** (Weather etc.) are not simulated by the module, all bits = 0
- Second bit **16** (STW) is set to 1 for one hour before the start & end of summer time.
- Second **17** (CEST) is set to 1 during Central European Summer Time (UTC+2)
- Second **18** (CET) is set to 1 during Central European Time (Winter) (UTC+1)
- The time code is UTC+1 during winter and UTC+2 during summer time
- Second **19** (Leap Second)) is not simulated (bit always=0)
- Second **20** marks the start of encoded Time (bit always set to 1)
- Bits **P1**, **P2** and **P3** (Parity Bits) create "Even Parity" (even number of ones) when combined with their designated data bits (e.g. P1=Minute)
- Second bit **59 (Minute Mark)** Has no data (no low pulse during the second)

More information can be found at:

Wikipedia

10. WWVB **USA** Radio Time Protocol Information

The WWVB time and date code is broadcast from a 60 kHz radio transmitter near Fort Collins, Colorado

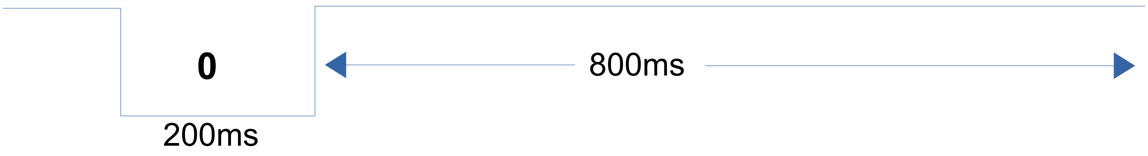
The full sequence of information is transmitted in binary coded decimal format each minute.

The time code includes the following information:

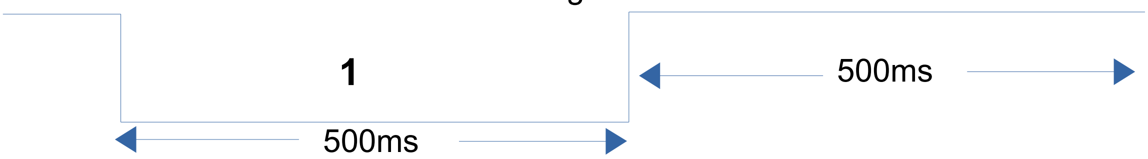
- year
- day of year
- hour
- minute
- DST warning and in effect indicators
- Leap Year indicator

The broadcast time is always UTC.

A 200ms low at the start of a second = logic 0



A 500ms low at the start of a second = logic 1



(There are also seven special seconds where there is an 800ms low marker)

The module normally simulates the time-code using TTL/CMOS logic levels without any carrier wave so that it can be directly connected to the logic decoder input of your clock.

There is an option to enable a 60Khz (approx) very low power carrier wave. A suitable coil via a current limiting resistor (1kΩ) could be connected between the module data output and ground for testing a clock with a built-in WWVB radio receiver over a short distance (a few inches only).
This may require the clock being tested to be metallic-ally screened in some way to prevent it receiving the real WWVB broadcast signal.

WWVB Time & Date code								
Sec	**Bit**	**Info**	**Sec**	**Bit**	**Info**	**Sec**	**Bit**	**Info**
00	FRM	Frame Marker (800ms)	**20**	**0**	**Bit always=0**	**40**	**0**	DUT1 Value (not used) **Always 0**
01	40	**Minute** (00-59)	**21**	**0**	**Bit always=0**	**41**	**0**	
02	20		**22**	200	**Day of Year**	**42**	**0**	
03	10		**23**	100		**43**	**0**	
04	**0**	**Bit always=0**	**24**	**0**	**Bit always=0**	**44**	**0**	**Bit always=0**
05	8	**Minute** (00-59)	**25**	80	**Day of Year**	**45**	80	**Year** (00-99)
06	4		**26**	40		**46**	40	
07	2		**27**	20		**47**	20	
08	1		**28**	10		**48**	10	
09	M1	Marker (800ms)	**29**	M3	Marker (800ms)	**49**	M5	Marker (800ms)
10	**0**	**Bit always=0**	**30**	8	**Day of Year** 1=1st Jan 365= 31st Dec (366 for leap year)	**50**	8	**Year** (00-99)
11	**0**	**Bit always=0**	**31**	4		**51**	4	
12	20	**Hour** (00-23)	**32**	2		**52**	2	
13	10		**33**	1		**53**	1	
14	**0**	**Bit always=0**	**34**	**0**	**Bit always=0**	**54**	**0**	**Bit always=0**
15	8	**Hour** (00-23)	**35**	**0**	**Bit always=0**	**55**	LY	Leap Year
16	4		**36**	**1**	DUT1 Sign (not used) **Always "+"**	**56**	**0**	Leap Sec (not used)
17	2		**37**	**0**		**57**	**TCW**	Time change warning
18	1		**38**	**1**		**58**	**DST**	DST bit
19	M2	Marker (800ms)	**39**	M4	Marker (800ms)	**59**	M0	Marker (800ms)

TCW (Time Change Warning)	**DST** (Daylight Saving Time)	**Meaning**
0	0	**DST not in effect** (DST bit is cleared from 00:00 UTC on day AFTER change)
1	0	**Time changes forward to DST today** (warning bit is set from 00:00 UTC on day of change)
1	1	**DST is in effect** (DST bit is set from 00:00 UTC on day AFTER change)
0	1	**Time changes back from DST today** (warning bit is cleared from 00:00 UTC on day of change)

Notes

- WWVB always sends **UTC** time and date for the ***current*** minute.
- The BCD time code is sent with the ***Most*** significant bits first.
- Seconds with Markers **M0-M5** and **FRM** start with an 800ms low pulse.
- The month, month-day and day of week have to be calculated by your clock from the **Day of Year** and **Leap Year** flag bit data.
- LY (Leap Year) is set to 1 during a leap year.
- DUT1 values are not simulated by the module (DUT1 value is set to 0)
- See the extra table above for an explanation of the DST warning and flag bits.

More Information can be found at:

Wikipedia

11. JJY **Japan** Radio Time Protocol Information

The JJY time and date code is broadcast from two radio transmitters in Japan:
A 40 kHz transmitter (JJY40) on Mount Otakadoya in Fukushima prefecture in the north.
A 60 kHz transmitter (JJY60) on Mount Hagane, Kyushu island in the south.
The transmitter frequencies are different to prevent interference, the time code is identical.

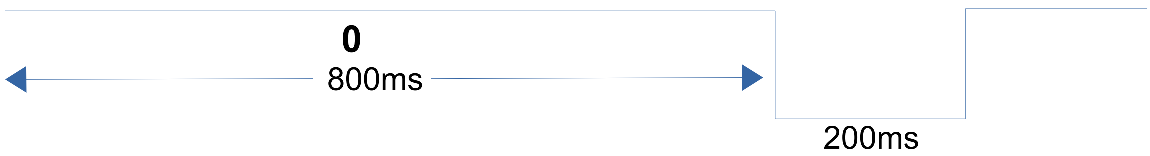
The full sequence of information is transmitted in binary coded decimal format each minute. (parts of the code are very similar to the WWVB code)

The time code includes the following information:

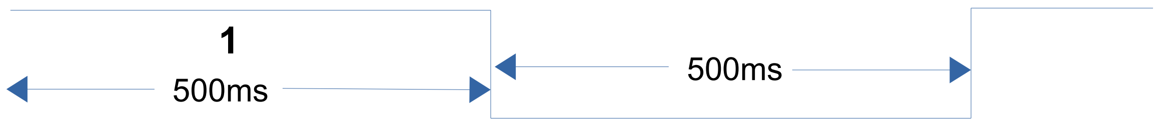
- year
- day of year
- hour
- minute
- day of week

The pulses are the opposite to other codes such as WWVB, each second starts with a **high** pulse.

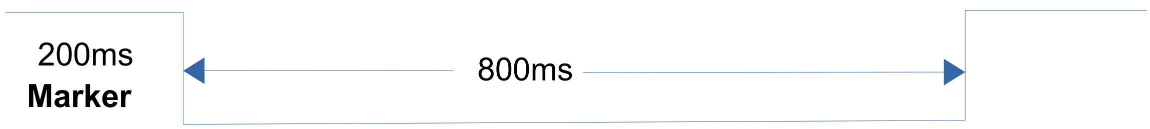
An 800ms **high** at the start of a second = logic **0**



A 500ms **high** at the start of a second = logic **1**



A 200ms **high** at the start of a second = **Marker bit** (there are 7 marker seconds per minute)



The module normally simulates the time-code using TTL/CMOS logic levels without any carrier wave so that it can be directly connected to the logic decoder input of your clock.

There is an option to enable a 60Khz or 40Khz (approx) very low power carrier wave as before.

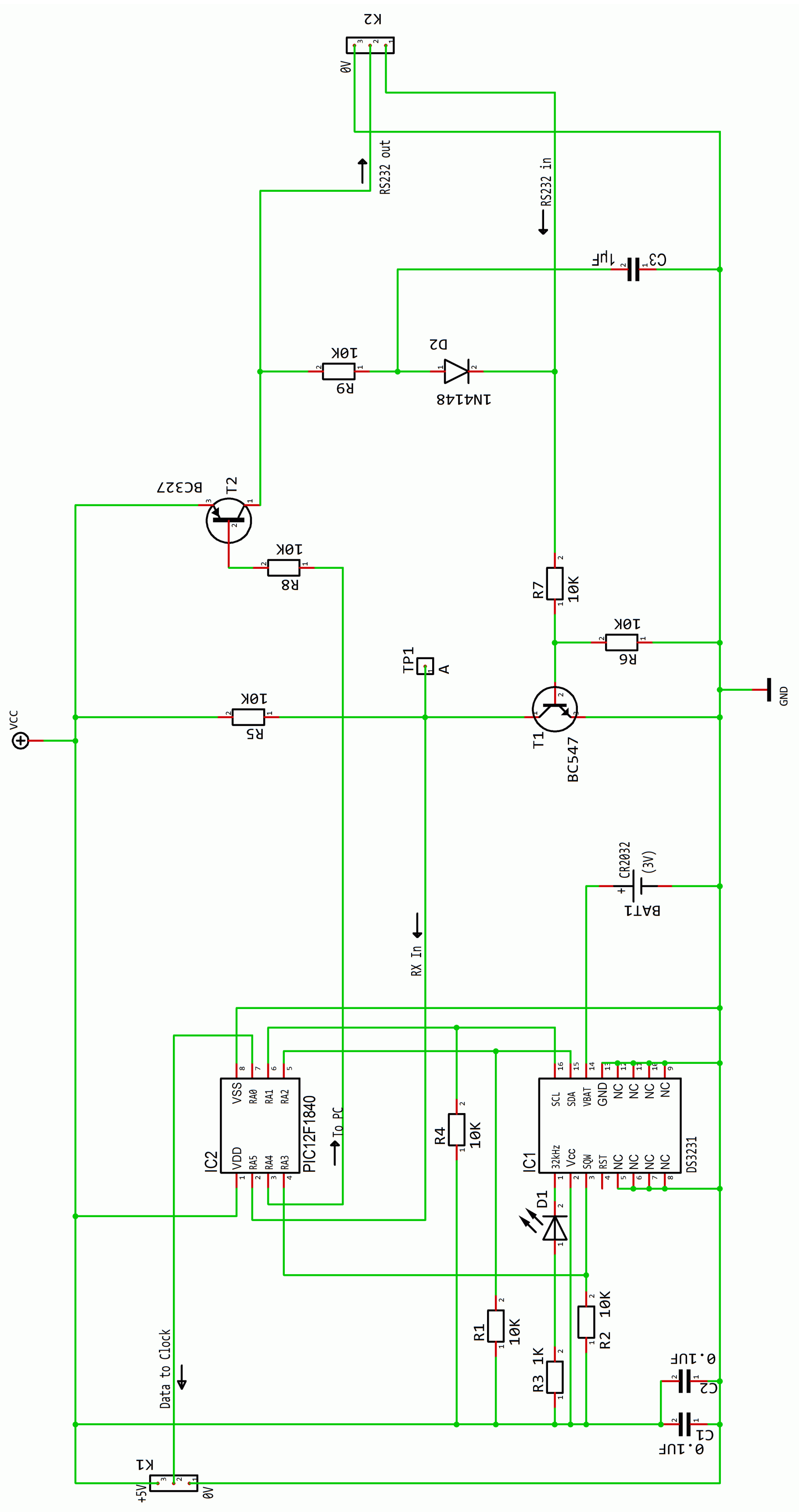
JJY Time & Date code								
Sec	**Bit**	**Info**	**Sec**	**Bit**	**Info**	**Sec**	**Bit**	**Info**
00	FRM	Frame Marker (200ms)	**20**	**0**	**Bit always=0**	**40**	**0**	**ST** (reserved)
01	40	**Minute** (00-59)	**21**	**0**	**Bit always=0**	**41**	80	**Year** (00-99)
02	20		**22**	200	**Day of Year**	**42**	40	
03	10		**23**	100		**43**	20	
04	**0**	**Bit always=0**	**24**	**0**	**Bit always=0**	**44**	10	
05	8	**Minute** (00-59)	**25**	80	**Day of Year**	**45**	8	
06	4		**26**	40		**46**	4	
07	2		**27**	20		**47**	2	
08	1		**28**	10		**48**	1	
09	M1	Marker (200ms)	**29**	M3	Marker (200ms)	**49**	M5	Marker (200ms)
10	**0**	**Bit always=0**	**30**	8	**Day of Year** 1=1st Jan 365= 31st Dec (366 for leap year)	**50**	4	**Day of Week** (Sunday=0 Saturday=6)
11	**0**	**Bit always=0**	**31**	4		**51**	2	
12	20	**Hour** (00-23)	**32**	2		**52**	1	
13	10		**33**	1		**53**	**0**	Leap second warning
14	**0**	**Bit always=0**	**34**	**0**	**Bit always=0**	**54**	**0**	Leap second type
15	8	**Hour** (00-23)	**35**	**0**	**Bit always=0**	**55**	**0**	Unused **Bits always=0**
16	4		**36**	**P1**	**Hours Parity**(even)	**56**	**0**	
17	2		**37**	**P2**	**Mins Parity** (even)	**57**	**0**	
18	1		**38**	**0**	**STW** (reserved)	**58**	**0**	
19	M2	Marker (200ms)	**39**	M4	Marker (200ms)	**59**	M0	Marker (200ms)

Notes

- JJY always sends "JST" (**J**apan **S**tandard **T**ime) which is **UTC + 9 Hours.**
- The time and date are for the ***current*** minute.
- The BCD time code is sent with the ***Most*** significant bits first.
- Seconds with Markers **M0-M5** and **FRM** start with a 200ms high pulse.
- The month, month-day and day of week have to be calculated by your clock from the **Day of Year** data (no leap-year data is sent.)
- Parity bits **P1** and **P2** make "even" parity for the Time (Bit is set if there are an odd number of bits to make an even number of bits)
- **STW** and **ST** are Summer Time indicators reserved for future use (always=0).
- Leap second warning (seconds 53 & 54 are not simulated (always=0))

More Information can be found at:
Wikipedia

12. Schematic Diagram



13. Circuit Description

Very accurate timekeeping (within ± 5.4 seconds/month) is achieved by the use of a Maxim DS3231 RTC (Real Time Clock) chip (IC1).
The RTC has an integrated temperature compensated oscillator and crystal.

It has battery backup (3V lithium cell) to maintain time and temperature compensation when there is no main power.

It communicates using an I^2C Interface (pins 15&16) to a PIC12F1840 microcontroller (IC2).

IC2 reads the time and converts it into either GPS, MSF DCF77 or WWVB protocol and sends it out on pin 7.
The data is synchronised to be sent exactly every second by the use of a 1Hz pulse sent from the RTC (Pin 3).

The microcontroller can be configured using a serial port (RX pin 2) and (TX pin 3).
Transistors T1 and T2 are used to convert between the microcontroller's logic voltage and the ± voltages used by RS232 Serial ports.

T1 buffers the incoming data ± voltages to between 0 and +5V and also inverts the data to make it compatible with the microcontroller serial input.

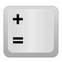
T2 has to be able to convert the 0 to +5V serial output from the microcontroller to RS232 compatible ± voltage levels and invert the data.
Diode D2 conducts only the negative voltage obtained from the input RS232 serial connection (from your PC's RS232 port) and C3 smooths it during periods when incoming data causes the voltage to go positive.

(Another solution may have been to use a dedicated RS232 converter chip such as a MAX232 but these constantly consume about 7-8mA of power which is twice that of the microcontroller and RTC chip combined!)

Test Point 1 (Marked “A” on the PCB) is provided to allow checking the incoming data with a logic probe.
It can also be used to directly connect a GPS module that only has a logic level serial output (this bypasses transistor T1 which would otherwise invert the data making it not compatible).

The Red status LED (D1) is used for indicating outgoing/incoming data pulses.
As the microcontroller doesn't have any more port pins available, it uses a spare 32kHz output (pin1) on the RTC chip which is controlled by the microcontroller to turn on/off the LED using the 32kHz pulses from the RTC.

14. Fault Diagnosis

Problem	Possible Solution
Module does not work at all. (Red LED does not flash when power is applied.)	• Power off and remove the Coin Cell for a few minutes then re-fit it. • Check your 5v regulated power supply is correctly connected (see section 4).
Module does not respond to the serial  terminal config start (= equal) key.	• Check the terminal is set to 4800 baud, 8bits, No parity, 1 Stopbit. • Check the correct serial Com port has been selected. • Check the RS232 connections to the module are correct (see section 4) (make sure you have the RX and TX wires connected correctly.) • Ensure that the module actually works when powered on (see “Module does not work” problem above.
Your Clock does not receive a time sync.	• Some clocks require inverted pulses. Try using the “Invert” command. • Check the logic data output pin is correctly connected to your clock logic data (TTL level) data input. • Check your clock time protocol configuration settings. • Ensure that the “Carrier” option is set to “Off” if the clock is physically connected to the module.
Your radio-controlled clock does not receive a time sync	• Follow the instructions for connecting a coil or short length of wire to act as an aerial. • The clock must be within a maximum of few inches of the coil/aerial. (The use of any RF amplifier is not supported) • Any other transmission on the same frequency may prevent reception from the module.

15. Specifications

Power supply requirements:
5 Volts DC maximum (Regulated) for the module.

3 Volts lithium coin cell (CR2032) for the RTC timekeeping battery backup.

Main Power consumption:
4mA (with LED flashing)
3mA (LED off)

Backup timekeeping battery power consumption:
0.85µA (Estimated run-time >10 years (or the shelf life of the CR2032 battery))

Dimensions:
PCB only - 33mm X 27.5mm X 1.6mm (1.3" X 1.1" X 0.06") (PCB has rounded corners)
Height approx 20mm (0.79") including battery holder

Timekeeping Accuracy:
Unadjusted RTC accuracy ±2ppm from 0°C to +40°C. (about ± 5.4 seconds per month)

Crystal Aging:
First year: ±1ppm
0-10 Years ±5ppm (RTC Manufacturer's estimate)

Weight:
Approx 11 Grams (0.4 oz)

16. Feedback

We welcome your comments and suggestions. If there is a feature you particularly like or dislike, or you have any feature requests:
Please email using the contact form on the website: unusualelectronics.co.uk

Manual revisions

v1.0 - Original version.
v1.1 - Added "Inverted Output" config option (for firmware v1.3 or later).
v2.0 - "MasterPulse" mode added (for firmware 2.0)
v2.1 - Added fault diagnosis section