

GPS-based MSF/DCF time-code generator

Design Targets

To use the low cost Garmin 18x LVC OEM GPS receiver (sensor) to generate a highly accurate MSF or DCF time code. The time-code can be fed into a suitably equipped digital clock to create a time-piece with an accuracy in the order of 1 microsecond. The hardware required is very straight-forward and low cost - a dsPic33 and a D-type latch, all the tricks are in the software.

Hardware

The Garmin 18x OEM series of GPS receivers are small, reliable, weather-proof and low-cost. The 18x LVC variant (part number 010-00321-36) features a one microsecond accurate pulse-per-second output in addition to a NMEA standard RS232 serial data output. The leading edge of the positive-going PPS signal is aligned with great accuracy to UTC seconds and buried in the subsequent serial output data burst is the time and date (and position fix) corresponding to the preceding PPS pulse. The D-latch is used to propagate the PPS pulse through to the MSF/DCF output pin forming the leading edge of the negative going time-code second marker pulse, it's the software's job to fill in the remainder of the time-code pulse for each second. The 18x LVC GPS receiver comes with a metric 6-pin JST connector which plugs straight in to the PCB, through this connector flows 5V power (at < 100mA) to the receiver and after a few seconds initial acquisition time back comes the TTL PPS pulse and the +/- 5V RS232 signal. There's also an RS232 serial input to the receiver which can be used to change the 18x LVC's settings, see later. The dsPIC33FJ12GP201 reads the serial data and monitors the PPS pulses (via level shifters etc to protect the 3.3v PIC from over-voltage) its software then generates output signals to drive the D-latch and construct the output time-code pulses. An input pin on the dsPIC is used to select either DCF or MSF output time-code formats, link J1 pins 3 and 4 (grounds RB0) to select DCF format. The dsPIC's internal FRC oscillator is used as the master clock avoiding the need for an external crystal. The Garmin receiver is available from RS in the UK, stock number 445-090 (don't be put off by the misleading picture which shows the wrong connectors) and other sources, Garmin part number: 010-00321-36.

I/O Connectors

J1 PIC in-system programming header 5-pin MOLEX

- J1.1 MCLR
- J1.2 3v3
- J1.3 GND
- J1.4 ISP Data & MSF/DCF select (MSF when open, DCF when linked to pin3, GND)
- J1.5 ISP Clock

J2 Power-in and time-code out header 3-pin MOLEX

- J1.1 Time-code output, low going pulses, open collector with weak pull-up
- J1.2 +5V
- J1.3 GND

J3 Garmin 18x LVC interface header 6-pin JST

- J2.1 YEL Positive-going 5v TTL PPS signal
- J2.2 RED Positive supply, 4.0 - 5.5v, <100mA
- J2.3 BLK GND
- J2.4 WHT RS232 polarity serial NMEA Tx data, +/-5V
- J2.5 BLK GND
- J2.6 GRN RS232 polarity Rx data (tied to GND by GPS2MSF)

Garmin 18x sensor firmware update

Older firmware versions of the 18x may not perform entirely to specification in so far as the serial data can occasionally be more than 1000ms delayed with respect to the PPS pulse it is referring to, this results in GPS2MSF omitting output pulses. This bug is fixed in firmware v3.7 or later so you should make sure you have at least this version installed in your unit, unfortunately there seems to be no way of discovering which firmware version is currently loaded, so in case of problems the first thing to try is an upgrade. You'll need a PC with serial comm port, Garmin's

SNSRCFG_320.EXE sensor configuration utility, their latest rgn firmware image for 18xLVC sensors (eg 079601000370.rgn) and their Updater.exe program. You can download all these from the Garmin website, but they take some finding, alternatively get them from www.marvellconsultants.com/gps/garmin_update.zip. The rather clunky SNSRCFG_320.exe utility runs on a PC and communicates with the sensor over a serial comm port. Once you have the unit connected according to the table below...

Garmin pin 1	YEL	PPS	no connect
Garmin pin 2	RED	+5V	+5v, 100mA supply
Garmin pin 3	BLK	GND	Dee9 female pin 5
Garmin pin 4	WHT	TX -->	Dee9 female pin 2
Garmin pin 5	BLK	GND	0v supply
Garmin pin 6	GRN	Rx <--	Dee9 female pin 3

run the utility and select the 18 PC/LVC option, in Comm->Setup enter the comm port and baud-rate (normally 4800), click OK then click Comm->Connect, if all is well you should get a confirmation dialog. Once connection is established you can use Config->Update Software (or F12) to perform the update, just follow the instructions. Additionally you can use Config->Get Configuration from GPS (F8) followed by Config->NMEA Sentence Selections (F7) to optionally select only the transmission of GPRMC sentences - these contain the time and date fields used by GPS2MSF. If the baud rate is not currently set to 4800 use Config->Sensor Configuration (F6) to change it. Finally use Config->Send Configuration to GPS (F9) to reprogram the sensor. A power cycle is required before any baud rate change takes effect.

Software

The software, source in C for the Microchip C30 compiler, monitors the PPS pulses and reads the NMEA data bursts from the GPS receiver, it aligns and adjusts it's internal timing to the PPS pulses and extracts time and date info from the serial data stream. From these inputs it generates a time-code bit stream, to either MSF or DCF format, and drives the external D-latch in such a way as to complete each time-code pulse appropriately for the current second. Note that the leading edge of each output pulse is generated in hardware directly from the PPS pulse so as not to compromise timing accuracy. The software does not try to be a real-time clock, if the GPS data or PPS signals stop then it will stop generating time-code pulses until the GPS signals reappear. However it's not quite as simple as it sounds - the GPS serial data is retrospective at the level of seconds ie it's referenced to the preceding PPS pulse. Also the MSF time code is predictive at the level of minutes, ie the time code bits being output at any given point are referring to the up-coming minute. We therefore need to add 1 second and 1 minute to the received GPS time and date in order to be ready for the next time-code pulse. It gets a bit more complicated still with the conversion of UTC to German Legal Time (plus 1 hour) and the adjustment required for daylight saving time during the summer months (plus 1 more hour), remember GPS time is UTC, it does not include DST adjustments. So we need to be able to add 1 second, 1 minute and 0, 1 or 2 hours to the received GPS time in order to generate the output time-code. This requires full calendar capability with day of week calculation and a model for predicting when DST changes are needed, mercifully both MSF and DCF change over to and from DST at the same time. One more complication is that both MSF and DCF include warning bits for DST change-overs which are set an hour prior to the actual change, the software presented here supports all these features.

Limitations

Unfortunately it's not possible to support *all* the features of the MSF and DCF time-code transmissions, firstly leap seconds: the NMEA data provides no prior warning of leap second insertion and the extra second is added (in the form of a repeated second #0) at the beginning of the first minute of the calendar month instead of at the end of the last minute in the (previous) month. Consequently DCF's A2 alert bit #19 is not implemented and the elongated minute occurs in the first minute of the month (rather than the last). This is likely to be an issue only for the most exacting of applications. There are further minor issues: DCF's MeteoTime weather data bits (#1..14) and call bit #15 are clearly not implementable and are set to 0. MSF's DUT1 bits, which indicate the offset between UTC and sidereal time, are similarly unimplementable and are also set to 0.

Debugging

At power-up, with the 18x unit attached and with a good view of the sky, the LED should remain off until first GPS fix. At this point the 18x starts producing PPS pulses, and LED1 should start blinking with MSF or DCF data cadence, if it just stays off then it's possible that the 18x's baud rate isn't set to 4800. If you get missing output pulses then the 18x sensor firmware may need updating - see above.

Further software development

Those wishing to tinker with the source code are welcome to download and experiment with it as they wish but please make bug-fixes and improvements available to all. You'll need Microchip's MPLab IDE development software and C30 compiler, all free to download from:

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=81

To program or re-program the dsPIC33 you'll need a suitable in-circuit programmer, the cheapest unit supported by the MPLab environment is the PICKit3, I used an ICD2, now replaced by the ICD3. You'll need to make up a custom lead to use the board's Molex 5-pin programming port J1.

On-line resources

For Gerber files, part-list, schematic, source code, object code, photos, links etc.

www.marvellconsultants.com/GPS

Developing in C30 for the dsPIC processor range:

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=81

In-system programming PIC microcontrollers:

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2519
¶m=en534451&page=wwwdevMPLABEmulatorDebuggers

Garmin data sheets...

http://static.garmincdn.com/pumac/GPS18x_TechnicalSpecifications.pdf

RS

<http://uk.rs-online.com/web/search/searchBrowseAction.html?method=getProduct&R=0445090>

DCF77 specifics...

http://www.ptb.de/cms/fileadmin/internet/fachabteilungen/abteilung_4/4.4_zeit_und_frequenz/pdf/PTBM_50a_DCF77_engl.pdf

MSF specifics...

<http://www.npl.co.uk/science-technology/time-frequency/time/products-and-services/msf-radio-time-signal>

Daylight saving time...

http://en.wikipedia.org/wiki/Western_European_Summer_Time

Steve Marchant

steve@marvellconsultants.com