GOES Satellite timecode simulator
Or
Using the TrueTime DC468 again

Written by: Paul WB8TSL

Project goal
Recreate the time code signal to allow the TrueTime DC468 GOES receiver to operate again.

Choices of design
Simplicity of implementation of a complex 1970s system.
Remove the need for complex RF mixing schemes and filters.
For the purist this can be implemented if you have the time.
Low cost with modern parts availability.
Synchronization by GPS using the most common sentence GPRMC sequence
Minimum of actual construction effort and soldering.
Simplified and clearly understandable coding. (At least to me) Its Basic language.

This has been achieved.

One slight difference has been implemented however.
The original DC468 displayed the day of the year in day counts.
This has never been very useful to me so the leading digits now display the day of the month. Unfortunately I can’t do anything with the leading number like hex month. It will not allow that though other display digits do support hex. Secondarily I am out of program space in the SXb micro.

Background

This project actually started when another Time-Nut and I were discussing the Austron 2201a GPS timing receiver and we both realized we also had DC468s. The documentations available on the net for the various components. But key tid-bits are missing and had to be reversed out. Over 3-4 weeks I assembled the information, coded, tested and failed ultimately achieving success.

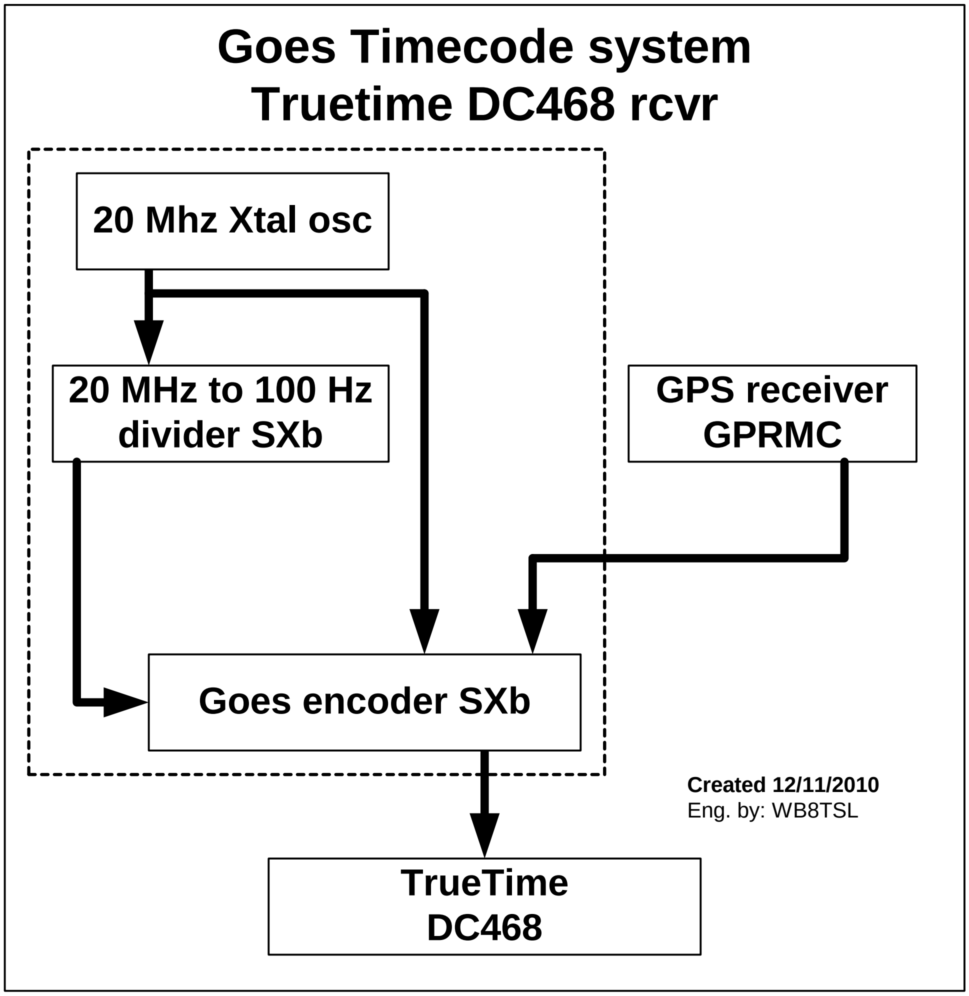
GOES background
In 1976 the GOES satellite time service was implemented. It was to deliver a time reference from orbit primarily for various weather platforms to deliver information for processing on shore.
A majority of the 468 Mhz signals time is spent encoding platform interrogation messages. The timecode actually is a 4 bit nibble and a complete time sequence takes 30 seconds. So either the 00 or 30 second message is what the DC468 uses to establish time.

GPS replaced the GOES timecode service in 2005 leaving approximately 2000 DC468s dead. These amazingly still show up at ham-fests and flea markets for $5 or so.
They are a very nice looking time display using red Sperry or panalplex 7 segment displays. In addition these units tend to have various time signals out such as IRIG B or serial timecode etc. Quite a mix actually.

A point never very clearly called out is the fact that the 100hz clock signal that locks a 10 MHZ local reference in the DC468 would have been very accurate and could have actually been a frequency transfer method. I don’t really know. But its very tightly locked and very very picky.

Simplified diagram and discussion

This section discusses the system solution.



In this diagram a 20 MHz dip oscillator (Must be adjustable) feeds two microprocessors. The first one divides 20 Mhz very accurately to 100 HZ and creates the encoding clock used for traditional mancester encoding. Not the more modern version.

The 100 Hz feeds the GOES encoder that takes the 100 Hz clock and GPS time sentences as inputs for synchronization. The output of the GOES encoder is a 100 Hz mancester signal that's injected into the DC468 receiver just past the 4.5 Mhz phase decoder. A small board on the back panel of the receiver. It has to be at this point because you can not directly inject it as a data signal to the DC468 micro because its clock signals will not be synchronized. Tried and failed that approach.

The overall process is to send one long 30 second sequence to the DC468 consisting of sync nibbles the timecode nibble and then the remaining platform interrogation nibbles. Lots of them. The whole message must exactly land on either the 00 or 30 second time mark.

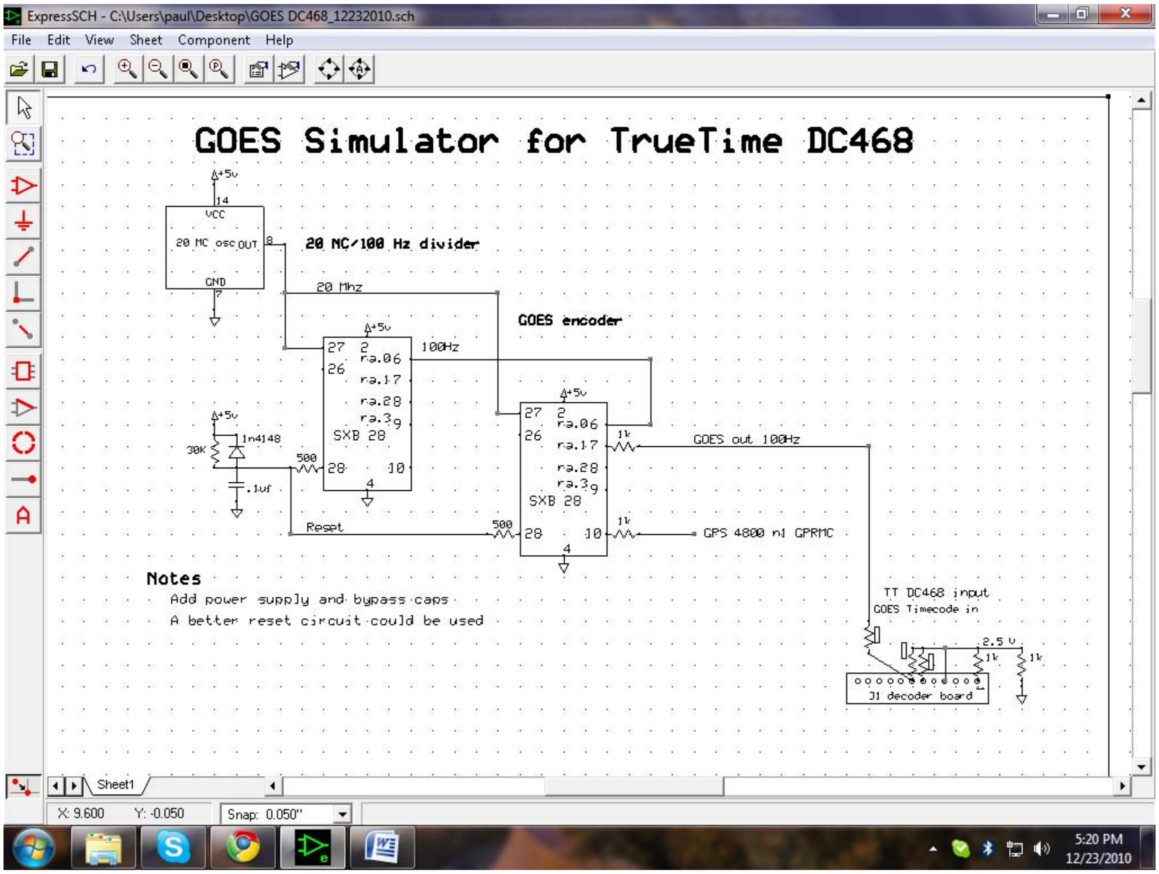
Simplified lock sequence of the DC468

Powerup
Timecode applied
Looks for the hex 05 or A0 sync signal. Colons solid
Looks for satellite orbit position info. Really! Turns on **east** or west GOES LED.
Then decodes several minutes worth of timecode and then displays the time.
The whole process typically is about 2-4 minutes.

Though I could synchronize the system to GPS every 10 minutes I have found that its stable over many hours. Depends on the quality of the 20 Mhz oscillator.
As such I tell it to only sync on any hour with a 2 in the hours position. 02, 12, 22.
Arbitrary time chosen

The board that I created is simply point to point wired and 6-7 wires per micro.
I really tried to force the divider into a single chip with the encoder. It simply did not work reliably and at $ 3.56 each wasn't worth the effort. Many pins on the 28 pin dip are simply not used. The wirings pretty simple.

Schematic and description
The following is the complete schematic of the GOES simulator.



Notes

Add power supply and bypass caps

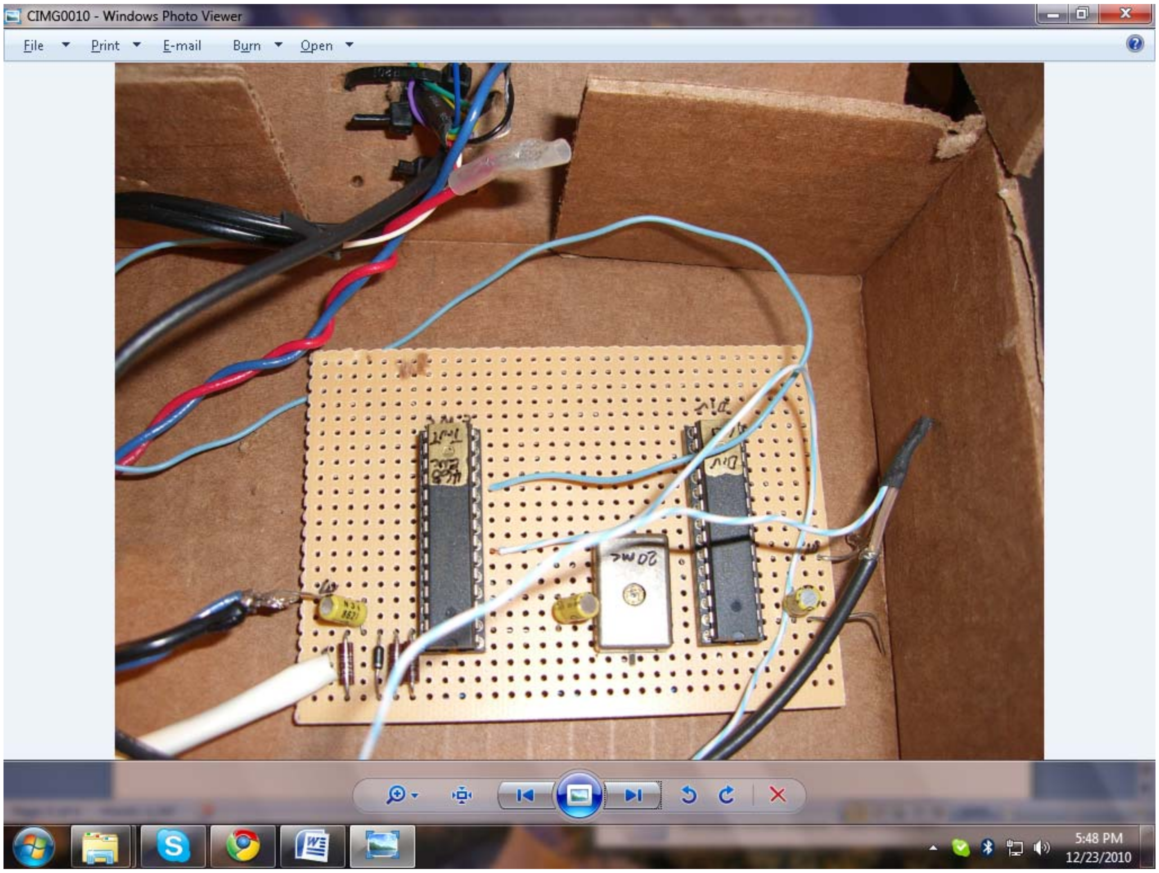
A better reset circuit could be used

On the upper left is the 20 Mhz master oscillator. This should be a cmos or TTL out. 1-2 ppm tcxo. I amusing a vectron that I had. It must be tunable over a small range so that the 100 Hz is 100hz. The Dc 468 expects plus or – 1 Hz stability from what I have seen. The clock recovery pll will not stay locked if a larger variation is presented.
This feeds the first SXb micro processor the 20 Mhz to 100 Hz divider that's loaded with program TrueTime 100 hz divider software. I used a HP5370 counter on a RB reference to set this frequency. Originally in 1976 the uplink must have been very accurate.

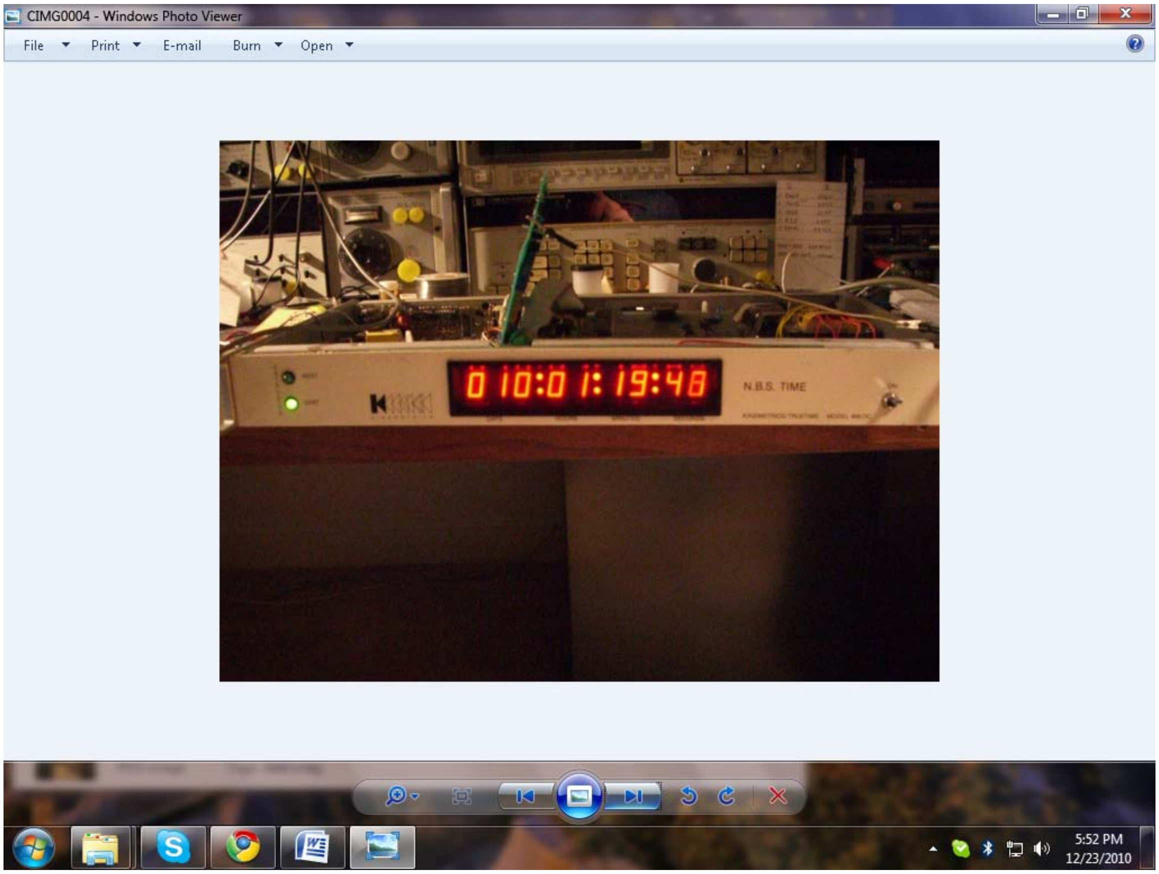
100 Hz from this divider micro feeds the second GOES encoder micro. Programmed with TrueTime_inservice_12102010 software. This micro runs very quick so 99% of its time is spent checking for rising and falling edges of the 100 Hz reference on Port A.0. Then it updates various variables and waits again. Output of the micro is Port A.1 it is a software Manchester encoder. This runs over to the bnc on the back of the DC 468 that's been modified as per the drawing in the bottom right of the schematic.
The software has numerous comments and can be read with a text editor or the Parallax IDE suite free to download.

Lastly a GPS signal that has a GPRMC sequence feeds the encoder. It looks for the GPRMC sequence and then counts positions and commas to obtain the time and date information. This can be a rs232 signal and no other conditioning is required. Possibly better protection with 2 diodes but it works well as is. Its set at 4800 baud N 8 1. Most of the GPS pucks I have run into out of the box seem to have this as a setting. My puck is a garmin GPS35. Picked up at a Hamfest for $5 or less.

Lastly the diode and cap are the power on reset. This works but can be improved with any small power reset IC these days. I plan to add one. If you are resetting leave if off for 10 seconds or so. It takes that long. The reset chip would be an improvement.



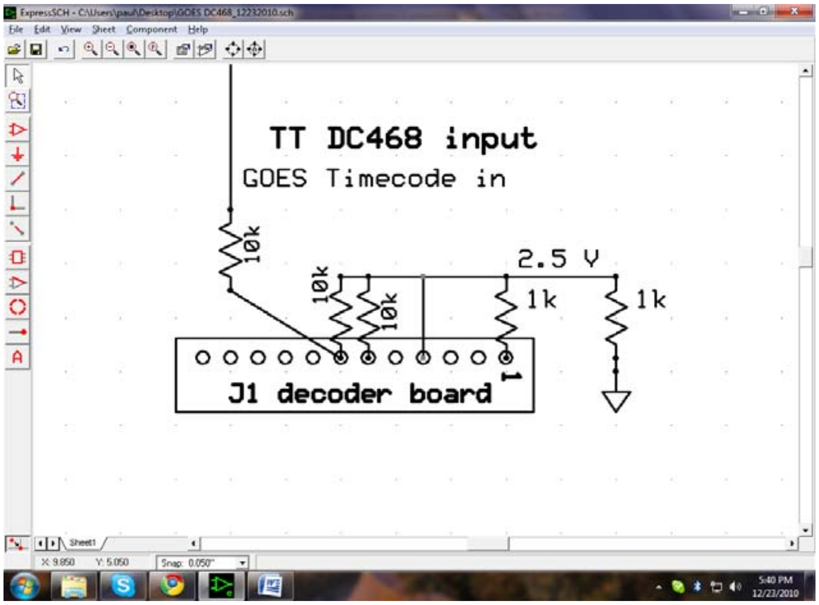
Assemble in a cardboard box for test
Operation



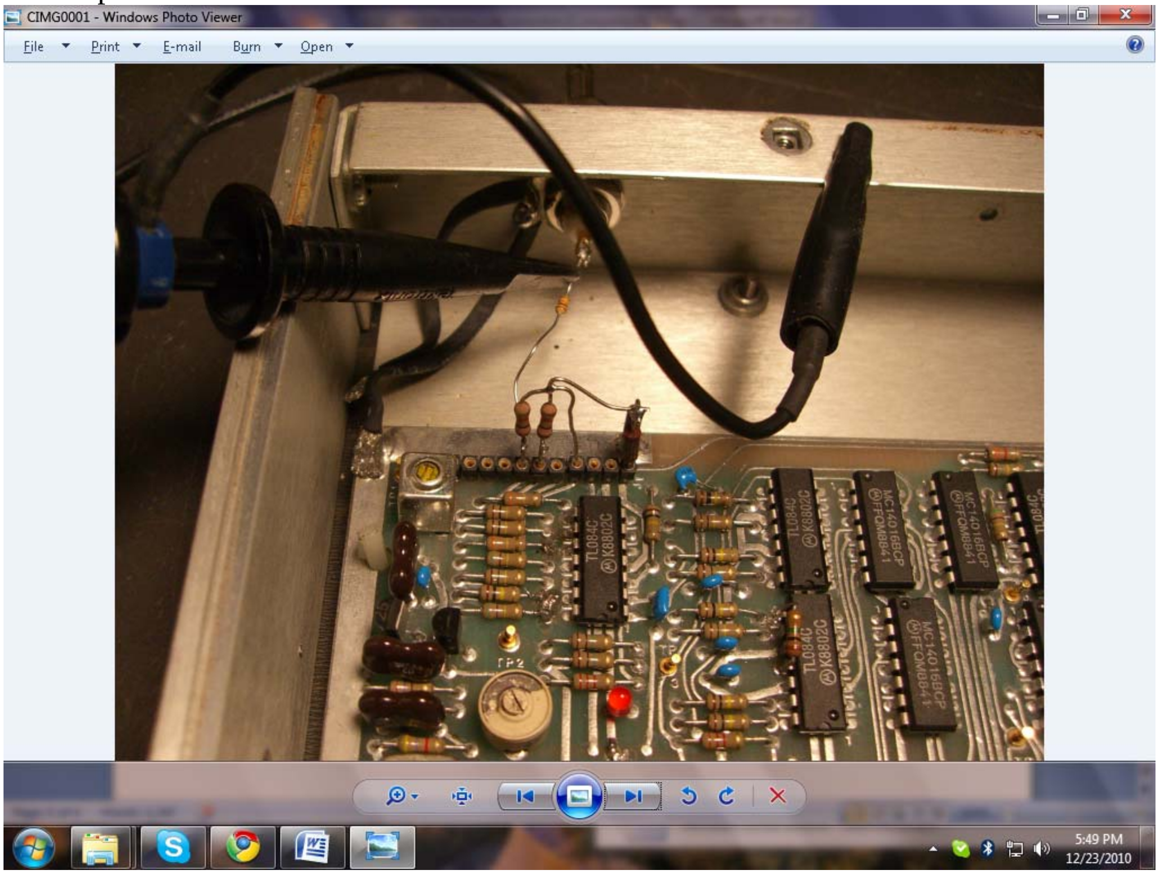
Connecting to the DC468

Connection to the DC468 is through the antenna BNC. This actually has signals and power on it. The best way to free up the BNC is to pull the 4.5 Mhz board off the back of the receiver. It’s the small board with 3 crystals connected to the BNC.
To remove lift the 12 pin jumper from the main decoder board. Cut the bnc wire somewhat hard to get to. Unscrew either 1 or 2 screws to the back panel. It should drop out in your hands. Keep the board who knows one day maybe someone will build a 4.5 MC phase encoder and filters.

The 12 pin jumper as 5 Volts on pin 1. That is the pin on the right side when looking from the front of the chassis. A small resistor pack is added to these 12 pins that bias the decoder circuits and inject the Manchester encoding and also signal a satellites acquired. See schematic and JI connector on the data board..



Actual picture



Final comment.

The systems working well and the design is offered up free. No warranties or other support are implied. No plans for kits or boards and such.

Good luck
Paul
WB8TSL