# N3T Black Box

| | | |
|---|---|---|
| 31/08/17 | Ver_BK | Some CC2500 radio test results documented. |
| 26/08/17 | Ver_BJ | RED warning LED will flash for 60 seconds before SD card gets written to. Sending/Receiving data by CC2500 RF transceiver with error detection/correction between Logger and base station working. |
| 18/08/17 | Ver_BI | GPS time now saved to RTC when GPS acquires sattelites. |
| 13/08/17 | Ver_BH | Modify SETUP001.TXT to add “erase” command. |
| 04/08/17 | Ver_BG | Modify SETUP001.TXT to add “ZERO” and “ERASE” command |
| 02/08/17 | Ver_BF | Modify SETUP001.TXT file format. |
| 29/07/17 | Ver_BE | Add Temperature Sensor to Technical Notes |
| 26/07/17 | Ver_BD | Add Timing to Technical Notes |
| 26/07/17 | Ver_BC | Add CAN bus data to Start-Up and Operation page. |
| 25/07/17 | Ver_BB | Update technical Notes - Start-up and Operation. |
| 24/07/17 | Ver_BA | Update Technical Notes - Start-up |
| 09/07/17 | Ver_AH | Update Storage Requirements in “Hardware” page. |
| 20/06/17 | Ver_AG | Update Technical Notes |
| 22/05/17 | Ver_AG | Update Technical Notes. |
| 19/05/17 | Ver_AF | Document changes to Agreement after meeting 10/05/17.<br>Add ‘Notes” section.<br>Add Technical Notes section. |
| 10/05/17 | Ver_AE | Add UWSD10 Wind Sensor and BlueTooth (provision only) |
| 03/05/17 | Ver_AD | Update Harware and Software pages |
| 18/04/17 | Ver_AC | Update Hardware pages |
| 13/04/17 | Ver_AB | Add calculations for storage requirments |

# Index

# Hardware

**The N3T PCB contains:**

Communications between main micro and various functional modules/chips:

| Function | Comms type | Coding Status | Ordered 25 sets 27/05/17 |
|---|---|---|---|
| wifi module | Serial UART | Do | ✔ |
| radio | SPI | ✔ Working | ✔ |
| 3 axis accellerometer/gyro | SPI | ✔ Working | ✔ |
| GSM/GPRS module | Serial UART 115200 | ✔ Working | ✔ |
| GPS receiver | Serial UART 9600 | ✔ Working | ✔ |
| 64 MBit flash AND/OR... | SPI | ✔ Working | ✔ |
| Micro SD card (32GB) | SPI to SD card, bit bang LineA, LineB to AT-MEGA328. | ✔ Working | ✔ |
| CAN BUS | Serial UART (via Radio) | Do | ✔ |
| Air temperature sensor | 1 Wire | ✔ Working | ✔ |
| UWSD10 Wind Sensor | Serial UART 4800 | ✔ Working | ✔ |
| BlueTooth Transceiver | Serial UART 115200 | ✔Working | ✔ |

**Power:**

12V 7AH sealed battery. 3V generated by Dc/DC converter, 5V generated by linear voltage regulator. If battery is allowed to be drawn down to 50% capacity and needs to be re-charged once every two days then the average current that can be drawn from the 7 amp-hour battery is 0.072 amps. Circuitry included to minimize power drain by turning ON/OFF functions as required.

**Storage requirements:**

| Variable | #bytes | | |
|---|---|---|---|

1. **Vehicle**

| | | | |
|---|---|---|---|
| Description/Data | 64 | Total=64 NOT INCLUDED IN GRAND TOTAL (ONLY WRITTEN ONCE AT START OF FILE) | |

2. **GPS:** $GPRMC,212433.000,A,3542.3325,S,17417.7205,E,1.57,149.76,090517,,,A*71

| | | | |
|---|---|---|---|
| Time | 212433, | 6+1=7 | |
| Lattitude | 3542.3325, | 9+1=10 | |
| Longitude | 17417.7205 | 10+1=11 | |
| SOG | 120.57, | 6+1=7 | |
| Bearing | 149.76, | 6+1=7 | |
| Date | 090517, | 6+1=7 | Total=49 |

3. **3 axis accelerometer/Gyro**

| | | | |
|---|---|---|---|
| X acceleration | -32768, | 6+1=7 | |
| Y acceleration | -32768, | 6+1=7 | |
| Z acceleration | -32768, | 6+1=7 | |
| X gyro | -32768, | 6+1=7 | |
| Y gyro | -32768, | 6+1=7 | |
| Z gyro | -32768, | 6+1=7 | Total=42 |

| | | | |
|---|---|---|---|
| 4. **Air temperature** | -054.06, | 7+1=8 | Total=8 |
| 5. **CANBUS data** | | | |
| RPM | 6543, | 4+1=5 | |
| Oilo Pressure | 110.1, | 5+1=6 | |
| Engine Temperature | 189, | 3+1=4 | |
| Battery Voltage | 12.7, | 4+1=5 | Total=20 |
| 6. **UWSD10 Wind Sensor** | | | |
| Wind Speed | 102.61, | 6+1=7 | |
| Direction | 359, | 3+1=4 | Total=11 |
| | | | **Total=130** |
| 7. **General purpose spares** | 126 | | Total=256 |

Total time available for download of all trip data is 7 minutes minimum. At 115200 baud this allows for 4.83 Mbytes of data to be downloaded. At 256 bytes per sample, 18,867 samples per download is possible. Assuming a sample rate of one 256 byte record per two second then approximately 10.48 hours of data can be downloaded in 7 minutes.

Formula: 256 Bytes/Record * 1 Sample/2 Secs. *60 Sec./Min. *60 Min./Hr. = 460,800 bytes/Hr. With 64 Mbits of flash memory a total of 17.7 hours of data can be recorded on this memory chip.

# Software

**The following routines have been coded and tested:**

1) Code to reading and parse $GPRMC NMEA 0183 sentence from GPS module working.

2) Reading and writing of files to/from slave micro <--> SD card working. Code to transfer data between main micro and SD card micro written and working.

3) Code to read air temperature from digital air temperature sensor is working.

4) Code to transmit and receive data to/from Radio transceiver module via SPI is working.

5) Code to save/recall data to 64 MBit flash memory chip is working

6) Code to read/write to/from 3 axis accellerometer/gyro working.

7) Code to read/parse UWSD10 wind sensor has been written and working.

8) Code to transmit/receive via Bluetooth has been written and working. (Note: Provisions for Bluetooth module will be made on PCB layout but not populated.

9) Code written to transfer data to a website working but needs more work to save data to a file on a website. (Agreed to use RF radio link in place of WiFi, see below).

10) Code to send text message via GSM/GPRS module working.

**To do:**

1) Write/test code to save log files to a website data file via WiFi. - Decided at meeting on 10/05/17 that uploading data to local PC using RF Transceiver module (used to talk to CAN bus PCP) would be best option at this point.

2) CAN BUS get details about what sentences and code same.

# Notes

1) Decided at meeting on 10/05/17 that uploading data to local PC using RF Transceiver module (used to talk to CAN bus PCP) would be best option at this point. CruzPro to supply RF Transceiver to connect to PC.

2) FAT32 file format. Name of data file on micro SD card is: YYMMDDxx.Iyy where: YY=Year, MM=Month, DD=Day, xx=Trip No (01,02,...99) on that day, I=Truck Type, yy= Truck ID.

3) 256 bytes of new data written to 64Mbit flash memory chip once every two seconds. New data written to Micro SD card once every 10 minutes.

4) When truck has not moved for ssss seconds (See SETUP001.TXT), it will be assumed that truck has parked for the night and all non-essential circuitry will be turned off until acceleration higher than aa.a (see SETUP001.TXT) is detected by 3D accelerometer. No data to be written when datalogger is “asleep”.

5) Accelerometer scale is +/- 2.5 G’s full scale. Gyro is +/- 250 degrees per second full scale.

6) CAN bus data to collect is TBD.

7) PC will be at end of truck route and upload of data will be via the RF Transceiver module, not the WiFi module. Bert to supply RF Transceiver module for the PC side and Windows software to save uploaded data to data files. Whenever truck speed falls below x.x Km/Hr (See SETUP001.TXT) the logger will try to contact and upload data saved to 64Mbit flash memory. Flash memory will be erased if upload sucessful.

8) Messages to be transmitted via GSM/GPRS module TBD. Cell phone numbers and rules for doing so are in file on SD card called SETUP001.TXT.

9) Prototype area for future expansion to be incorporated into data loger PCB.

10) Bert to email Martin estimated cost for long lead items so 20 sets of these can be purchased early. (DONE 20/05/17).

11) Agreed 24/8/17 that N3T Logger will upload data to stripped down version of N3T Logger PCB and store data on SD card instead of a PC at receiving end.

12) 7-10 minutes to process & tag logs (email 28/03/17). after truck stopped for 30seconds N3T Logger will attmept to connect to base station to upload data. So worst case we have 6.5 minutes to upload data. Actual upload speed tests show that 6117 bytes of data can be uploaded per second under good signal conditions. Less if signal is poor and re-sending of some packets of data is required (A packet is 52 bytes long and error checking is performed. Packet is re-sent continuously untill received correctly). At 6117 bytes of data per second 6117*65*60=2.38563 MBytes of data can be uploaded in 6.5 minutes. At an average of one 256 byte record per 1.3 seconds, 9318.87 records or 12,114 seconds of data (3.365 hours) can be uploaded in 6.5 minutes. Transmission speed is 250 H Bits/Sec. Actual measured thru-put is 48,936 BPS.

## Logger Start-up

When ppower is applied to the logger PCB it opens a file on the micro SD card called SETUP001.TXT which contains the following information in ASCII format **[Note no spaces after ':', All fields except Vehicle Description and Cell phone numbers must be exactly the length shown (e.g. Max Speed=3 characters, etc.)]. Cell phone #'s can be 9 or 10 digits.**

01,Vehicle ID:Iyy <-- See paragraph 2 on page 7

**Optional alternates to line 01:**

ZERO,Vehicle ID:Iyy <--This command will cause Logger to Zero out the 3D accelerometer/ 3D Gyro offsets at start-up. Use with caution.

erase,Vehicle ID:Iyy <--This command will cause Logger to Erase the 64Mbit flash memory at start-up.. Use with caution. Takes 30 seconds.

ERASE,Vehicle ID:Iyy <--This command causes Logger to Erase the entire SD card contents. System stops until power cycled. Use with caution.

02,Vehicle Description:40 ton lumber truck traveling to Marsden Cove <-- **64 characters max. See line 1 on page 3**

03,Cell1:021525839

04,Cell2:0211650314

05,Cell3:0000000000

06,Max speed:999 <-- Call all non zero) cell phone numbers above if vehicle travels faster than this speed (Km/Hr) and send a text message with date, time, truck ID, GPS location and speed. Will not text again unless higher speed seen. Full digits (i.e. 105, not 75.5 Km/Hr).

07,Parked After:ssss <-- Seconds to wait before deciding truck is parked for the night and turn OFF everything except the 3D accelerometer/Gyro.

08,Min Speed:ss <--- If truck is moving slower than this speed (Km/Hr) try connecting to radio to upload micro SD card trip data. If within radio range then upload data. Full digits. If less than 10 K/Hr use 0x (e.g.02).

09,Start Acceleration:aaaa <--- Acceleration value that determines that truck has started moving in the morning. if less than 9999 us 0xxx or 00xx.

10,Battery Warning Volts:vv.v <--- Send text message if battery voltage drops below this value. If less than 10.y use 0x.y.

11,Battery Sleep Volts:ww.w <--- Send text message and go into permanent sleep mode if battery voltage drops below ww.w

When the truck starts "moving" in the morning after being parked at night, the logger turns ON the GPS and starts a "ssss" second countdown timer (used to put system into "Sleep" mode if truck is not moving). This timer is decremented each time the GPS NMEA 0183 sentence is read and speed is less than s.s. This timer gets reset back to "ssss" seconds if speed is greater than s.s. Zero speed assumed if GPS has not locked onto satelites.

Every two seconds the logger PCB does this:

1) Talks to DS18B20 temperature sensor and commands it to start a temperature reading.
2) Listens to GPS and captures the $GPRMC sentence data to RAM.
3) Listens to UWSD10 wind sensor and captures wind speed and direction to RAM.

3b) Capture CAN bus data from PCB2 radio.
4) Processes the data above, converts to ASCII (where required) and fills in the information in the right places in the 256 byte data record . Data comma delimited in CSV format.
5) Writes the 256 byte data record to the 64Mbit flash memory.
6) Checks GSM/GPRS rules and sends text message(s) to cell phone numbers (if not zero).
7) If speed is less than s.s then try to connect via radio to see if able to upload data from 64Mbit flash memory. If upload is sucessful, then erase 64mBit flash memory and increment “trips”.
8) Decrement “ssss” timer if speed below “s.s” and if timer reaches zero go into sleep mode.

    If in Sleep Mode: Power down everything and put micro into low power mode. Wake up every ten seconds and look at accelerometer. If truck acceleration less than aaaa then go back into sleep mode. If acceleration greater than aaaa then come out of sleep mode and start logging data as above.

Every ten minutes the logger PCB does this:

1) Writes data from 64Mbit flash memory to SD card.
2) Send text message if battery voltage is less than vv.v
3) Send text message and go into permanent sleep mode if battery voltage drops below ww.w
4) Writes GPS Time to Real-Time-Clock chip if GPS has aquired satelites.

# Technical Notes

**8051<-->ATMEGA328P MicroSD card communications protocol**

The ATMEGA328P microprocessor is used to store ASCII and binary data to a Micro SD card formatted with the FAT32 filestructure. The ATMEGA328P is programmed with the Arduino SD card program:
Documents and Settings\Bert van den Berg\Arduino\CruzProSD\CruzProSD.ino

The 8051 communicates as Master and the ATMEGA328P as Slave over two data lines called LineA, LineB.

The SLAVE is told what functions to perform by the MASTER through a series of simple instructions detailed here.
.
A) All data from MASTER to SLAVE is sent as:
1) The '@' character, one 8 bit INSTRUCTION byte and
2) A variable number of DATA bytes, depending on INSTRUCTION byte

B) All data from SLAVE to MASTER (only in response to an INSTRUCTION) is sent as a variable number of bytes, depending on the instruction received.

Note: Not all INSTRUCTIONS cause the SLAVE to send data to the MASTER.

## MASTER INSTRUCTIONS

One '@' character followed by:

'A') Initialize SD card and report SD status. Slave returns one byte:
'0' - initialization failed
'1' - initialization successful
'2' - already initialized

'B') Save filename to Slave in form YYMMDDxx.Iyy where YY=Year, MM=Month, DD=Day, xx=Trip No (01,02,...99) on that day, I=Truck Type, yy= Truck ID. Slave returns one byte of status:
'0' - save filename failed
'1' - save filename successful

'C') Open file with name YYMMDDxx.Iyy for writing where:
'0' - File open failed
'1' - File open successful

'D') Close file. Slave returns one byte:
'0' - File closed failed
'1' - File closed successful

‘E’) Does file with name YYMMDDxx.Iyy exist?:

‘0’ - File does not exist
‘1’ - File exists

‘F’) Save x,y bytes of data (x,y=1 to 65,535), followed by x,y bytes of data to save, slave returns

‘0’ - writing data failed
‘1’ - writing data successful

‘G’) Open file with name YYMMDDxx.Iyy for reading:

‘0’ - File open failed
‘1’ - File open successful

‘H’) Read/Send x,y bytes of data (x,y=1 to 65,535). Returns x,y bytes of data. File remains open.

‘I’) Read/Send all bytes of data from open file. Returns all bytes of data and then closes file.

‘J’) Delete file with name YYMMDDxx.Iyy. Slave returns one byte:

‘0’ - File delete failed
‘1’ - File delete successful

‘K’) List all files/directories on SD Card along with file sizes

## C8051F380 Code Organization

The trip number ‘TripNo’ for that day is stored in NVM and incremented by one each time the truck stops for more than “Parked After” seconds. A stopped truck is assumed when SOG less than “Min Speed” Km/Hr. for “Parked After” seconds. TripNo is set to ‘1’ when the GPS day changes. See Logger Start-Up for descriptions of “Parked After” and “Min Speed”.

The GPS module sets the overall timing for all data collection and storage operations. Once per 2 seconds when the $GPRMC sentence is received, the Date,Time,GPS coordicates and SOG are collected and saved to RAM. The raw 3D gyro, accelleration data, air temperature, analog battery voltage, CAN Bus and wind speed/direction are then saved to RAM in ASCII format. At this time all data is written to the 64Mbit NVM.

Once every ten minutes the latest set of data are then written to the micro SD Card (see page 7 for Filename specification).

Whenever the truck speed is less than x.x knots (See SETUP001.TXT file) the RF Radio is checked to see if the data logger is within range of the ‘upload station’. If so, all data recorded on the 64Mbit NVM are sent to the ‘upload station’ radio transceiver and when completed, the contents of the 64Mbit NVM is erased.

**TCC2500 Radio Test Results**

TheN3T logger CC2500 transceiver transmits packets of 54 bytes (one packet length byte, 52 data bytes and one Checksum byte). The N3T Base CC2500 transceiver receives the 52 data bytes and calculates the checksum and compares it to the transmitted checksum. If they agree the base transmits back an ACK (Acknowledgement) byte, if not it transmits back a NAK (Not Acknowledged). If the N3T logger receives back an ACK it transmits the next packet, if it received back a NAK it re-transmits the faulty packet until it has been acknowledged. If the N3T logger can't get an ACK within 200 tries it gives up and will try again the next time that the truck has stopped for 30 seconds.

The CC2500 can transmit/receive on various frequencies in the 2400-2483.5 MHz ISM (Industrial, Scientific and Medical) and SRD (Short Range Device) frequency band. Channels 0 to 252 are supported with a channel spacing of 199.95 Khz.

Using channel 83 with varying delays between receiving and sending resulted in the following data:

Time to sucessfully transfer 337,920 bytes (1,329 256 byte records, 5 packets per record=6,600 52 byte packets)

| Delay | # of packet errors | Time (Seconds) | Thru-put (Bytes/sec) |
|---|---|---|---|
| 20 mSec | 440 | 564 | 599 |
| 10 mSec | 1024 | 147 | 2,299 |
| 5 mSec | 1464 | 104 | 3,249 |
| 2 mSec | 2016 | 96 | 3,520 |
| 0 mSec | ???? | 91 | 3,713 |
| ZERO (fastest) | ???? | 92 | 3,673 |

Using channel 126 with varying delays between receiving and sending resulted in the following data:

Time to sucessfully transfer 337,920 bytes (1,329 256 byte records, 5 packets per record=6,600 52 byte packets)

| Delay | # of packet errors | Time (Seconds) | Thru-put (Bytes/sec) |
|---|---|---|---|
| 20 mSec | | | |
| 10 mSec | | | |
| 5 mSec | | | |
| 2 mSec | | 89 | 3,797 |
| 0 mSec | | | |
| ZERO (fastest) | | | |

## Timing

The two second timing cycle starts when the $GPRMC sentence is received:

| | | |
|---|---|---|
| Read $GPRMC data (72 chars/4800) = | 0.015 | Seconds |
| Start temperature conversion | 0.001 | |
| Wait for and receive Wind data (0.015 to 1.015 sec max) | 1.015 | |
| Capture CAN bus data (16 bytes at 115,200 baud) | 0.002 | |
| | -------- | |
| Total | 1.033 | Seconds |

Time available to process above data and save to 64Mbit flash memory = 0.967 to 1.967 Secs.

## Timing for saving 64mBit flash memory data to SD card every 10 minutes:

In 10 minutes we save 10*60/2 = 300 records of 256 bytes each for a total of 76,800 bytes of data. At 10,000 bytes per second this will take 7.68 seconds, during which time we will have to still capture new data to the 64Mbit flash every two seconds.

At the ten minute mark, have logger talk to ATMEGA micro and start initializing the SD card. This takes 4 seconds. When SD card is initalized start saving Flash data to SD card at the rate of 4096 bytes (0.41 secs) once every 2 seconds. Close the file and turn off SD card when the flash data has been saved to SD card.

It takes up to 30 seconds to fully erase the 64Mbit flash memeory.