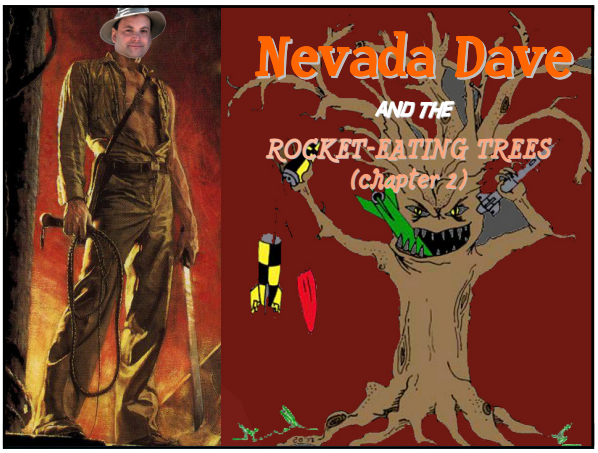
Interim Report 2:
GPS-guided Rocket Recovery

or…

Dave Lindbergh
NerdFever.com
NARCON 2010



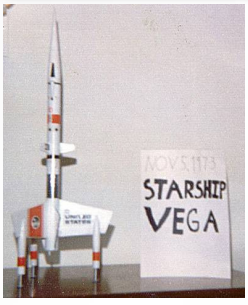
Apologies to NEMROC 2008 folks

- Howard Greenblatt couldn't present today due to a family emergency
- Some of this stuff is new since 2008, really!

Contents

Part 1 – The Wayback Machine

Early 1970s



November 5, 1973

July 1975

…July 1986…

…October 2000…

Part 2 – The Epiphany

November 2005

"Dad, can we go fly airplanes and rockets?"

??? Rockets ???

- But – *Rockets get lost*
- **Trees eat them!**
  - Also power lines, roofs, rivers, lakes, highways, dragons, unfriendly neighbors, trolls, aliens, witches, gnomes, and swamps.
- This is a CHALLENGE

Wee hours of November 17, 2005

- Can't sleep – rockets get lost…
- Must be some way to deal with this

Idea 1 – Radio tracking

- Put a radio transmitter in rocket and DF it
- Estes sold hardware to do this in the late 1970s
  - (MITS, actually…)
  - Lots of others do now (but I didn't know this then)
- Doesn't prevent trees from eating rockets

Idea 2 – Noise maker

- Put a noise maker in rocket
  - Follow the sound
- Simple piezo buzzers can generate > 100 dBA
  - (smoke alarms)
  - Small, cheap
  - Worth trying, but…
- ***Still*** doesn't solve rocket-eating tree problem

Idea 3 – *Don't lose the rocket*

- How?
  - Have it land in the right place, of course.
- Radio controlled return?
  - RC receiver and servos in rocket = Weight & Drag
  - Worse, requires *skill* in flying RC
    - Something I demonstrably do not have
- Automatic return to launch site
  - No RC receiver
  - No skill required!
  - More cool, too.

The simplest thing that will work?

- A string
  - Tug on string to get rocket
  - High drag solution
  - Not cool

Steering vanes

- Simple missile/smart bomb design
  - Simple sensor drives steering vanes
    - Similar to early guided missiles, like the Sidewinder
- Has to home on something
  - Radio? HF source, small antennas…seems hard
  - Light/IR? Very bright source or must track moving target with beam…hard
  - Moving too fast for GPS guidance (at 1 Hz)

Steering vane design

- How to avoid crashing?
  - Rocket is descending clean (fast)
  - Trigger parachute with sensor?
    - Pressure altimeter, ultrasonic ranging, radar…
  - Manual trigger via RC? (Risky if you're slow!)
  - Guide rocket into something big and soft?
  - Any mistake leads to a very broken rocket (at least!)
- Could work, but seems too risky

Glider vs. Steerable Parachute

- Glider design – steer with rudder
  - Could work. Flight is slow enough for GPS guidance.
  - Draggy wings on ascent
    - Or, complex & heavy wing deployment mechanism
- Steerable Parachute – tug on control lines
  - Can use motor deployment – reliable
  - Normal rocket design
  - Risks:
    - Slow airspeed – could be < wind speed
    - Lines can tangle

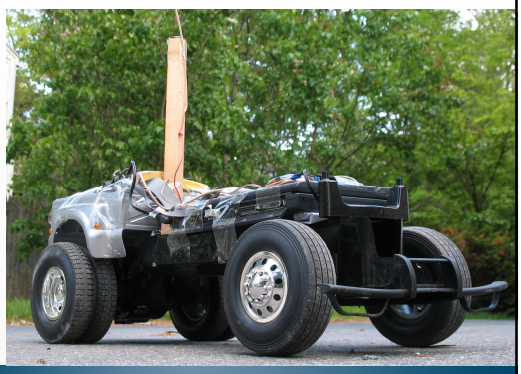
Navigation

- GPS looks really good
  - Cheap, small, low-power
  - Gives both direction & distance to launch site
  - In theory, you can land someplace else
    - (within gliding distance)
- Navigation control algorithm not trivial
  - Response time of control system
  - Winds aloft
  - Computation on small 8-bit MCU
    - Slow, no hardware multiply/divide (let alone floating point)
    - Very little RAM and ROM

Goal

- Self-contained descent steering module between rocket body and steerable parachute
  - GPS
  - Microcontroller
  - Servo
  - Power supply
  - Suitable parachute
- Nice to have:
  - On-board camera
  - Pressure altimeter – logging, electronic ejection
  - Track logger/on-board diagnostics

Rev 1 – The Truck



Truck goals

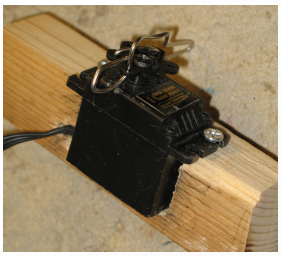
- Prove out steering algorithm on the ground
  - In-flight debugging is difficult
- Method:
  - Build GPS-steered truck
  - Get it to 'home' on a target by itself
- Truck hardware:
  - Radio-controlled truck platform
  - Win95 laptop ($50 on eBay) – runs navigation code
  - DeLorme Tripmate GPS (~1995, from parts box)
  - Hobby servo (steering)
  - PIC18LF2620 microcontroller (control servo)

"Rev 1" hardware – ground test

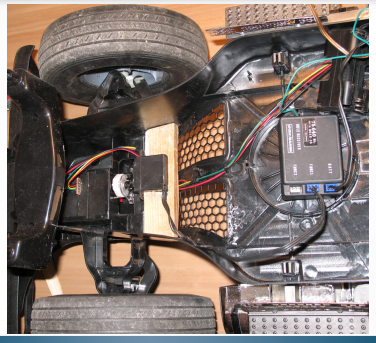
Feb 2006 – Pre-qualification testing

Testing with laptop, battery, GPS

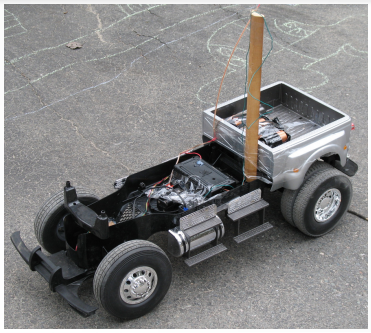
Servo for steering control



Installation with RC receiver

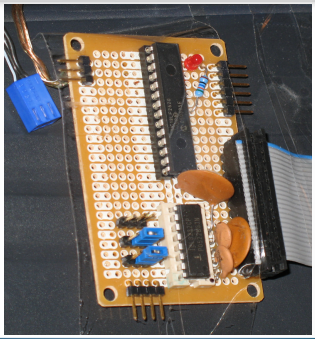


Testing servo operation (RC)



Ready for testing with GPS & nav

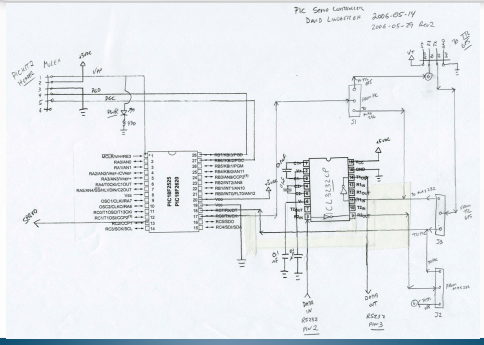
"Rev 1" Microcontroller



"Rev 1" Microcontroller

- Microchip PIC18LF2620
  - 2 MHz, internal oscillator
  - 4 kbytes RAM
  - 64 kbytes Flash (ROM)
  - C18 C compiler
  - Used to drive hobby servo
  - RS-232 input from laptop (steering commands)
- Why PIC18?
  - Very complete MCU in one chip
  - Free MPLAB IDE, free C18 complier
  - Generous free sample policy ☺

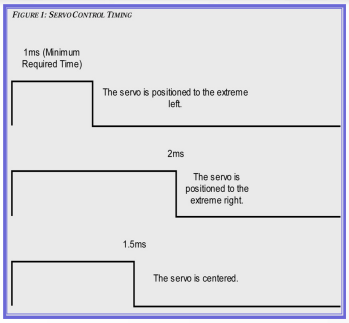
"Rev 1" Schematic



Hobby Servo control

- Pulses every ~ 20 millisecond
- Pulse length controls servo position
  - Range of ~ 1 to 2 ms (1.5 ms = center)

Image from *Hobby Servo Fundamentals* by Darren Sawics (try Google)

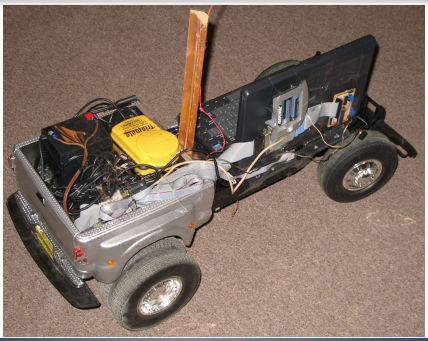


How to drive a servo with a PIC

- Initialize servo control pin low
- Generate interrupts every 20 ms
  - I use TMR1
- In ISR:
  - Raise servo control pin
  - Set hardware to drop control pin at correct time
  - I use CCP1
    - Compare mode, init pin high, force low on match
- All source code is posted on NerdFever.com

Finished test truck

Finished test truck



Test truck power supply

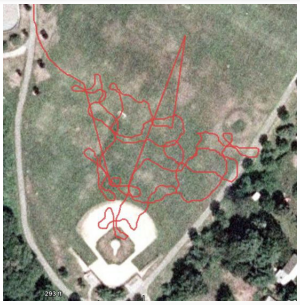
- Truck motion
  - 19.2v NiCd battery pack
- PIC
  - 3 x AA alkaline cells (4.5v)
- GPS, Servo
  - 4 x D alkaline cells (6v)
- Laptop
  - One lead-acid AGM battery (12v)
  - (internal laptop battery was dead)
- Complicated & heavy

Field testing – May 2006

Field testing

Navigation models

- Initial truck auto-navigation tests failed
  - Quite horribly
- Excel modeling of system was tremendously helpful
- Ported working Excel model to C code in laptop

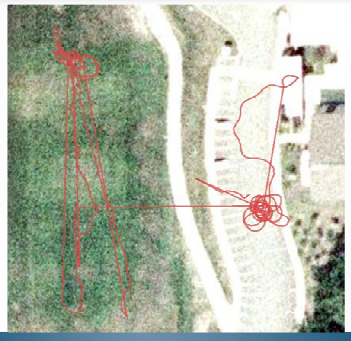


May 25, 2006

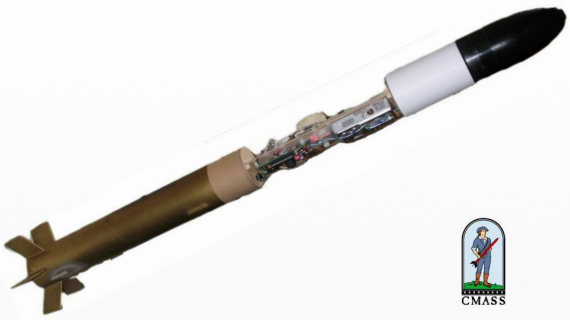
Field testing hazards

Field testing hazards

August 6, 2006 - Success



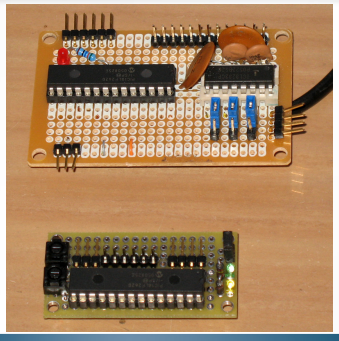
Rev 2 – Saturn Zero: Proof of Concept



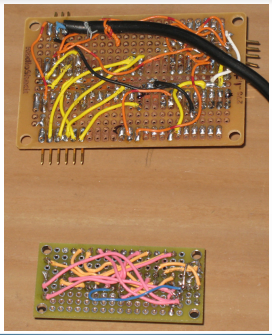
Saturn Zero Goals

- Test microcontroller hardware in flight
- Test GPS performance in flight
- Test parachute in flight
- Attempt steering of parachute via radio control
- Observe flight characteristics

"Rev 2" board – first flight hardware



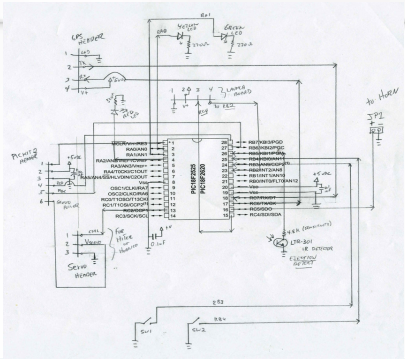
"Rev 2" board – first flight hardware



"Rev 2" Board – Flight hardware

- Same PIC18LF2620
- Serial I/O to EM-406 GPS (1 Hz)
- Drives servo
- Ejection detection via IR phototransistor
  - Nose cone had to be painted black
- 2 buttons
- 2 LEDs
- Piezo speaker output
- "Loud horn" (smoke detector horn) output

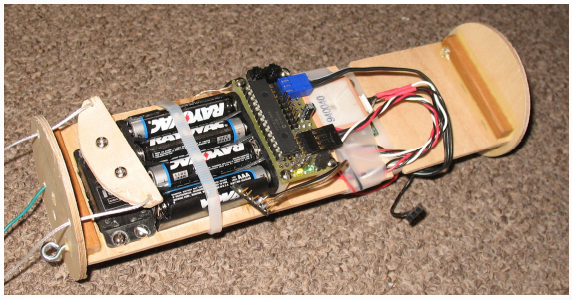
"Rev 2" schematic



"Saturn Zero" test rocket

- Estes BT-80 body tube (2.6")
- 29 mm motor mount
- NASA NPW-5 steerable parachute
  - "SuperChute" kite
  - $6 on eBay

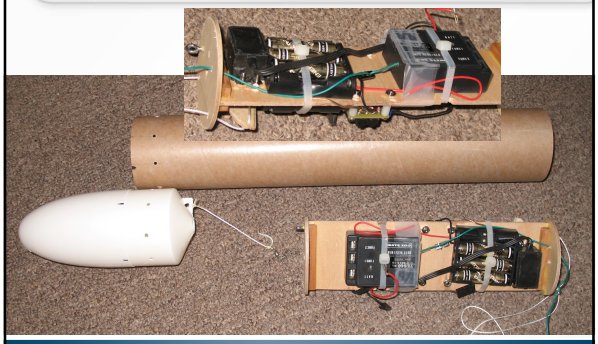
"Rev 2" Mounted on e-sled



Saturn Zero power supply

- PIC
  - 3 x AA alkalines (4.5v)
- GPS, RC Receiver, Servo
  - 4 x AA alkalines (6v)
- Each set of AA in spring battery holder
  - This a Bad Idea…

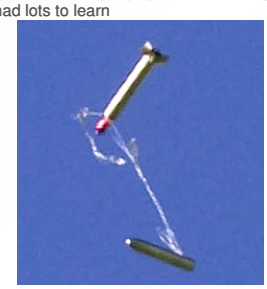
Reverse side of e-sled, RC receiver



Prepping for first flight

First flight – October 6, 2006

- Parachute didn't deploy (packed too tight)
  - I still had lots to learn



Flights 2 & 3 – November 4, 2006

- Flight 2 – too low!
  - D24-4T is way too small
    - 50' altitude
  - Parachute deployed (good)
  - CPU reset on landing
    - Spring-clipped AA batteries not a good idea
- Flight 3 – Crash landing
  - Improper parachute packing

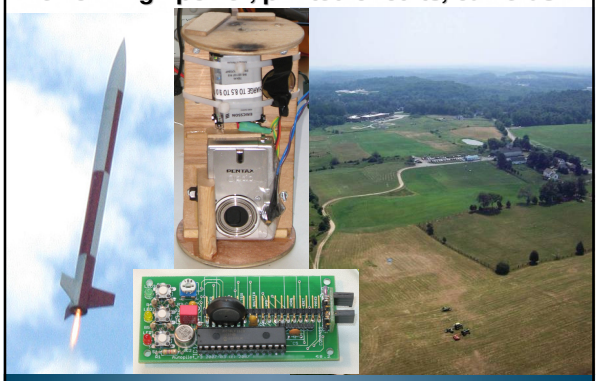
Flight 4 – November 18, 2006

- Flight 4 – Parachute deployed but lines tangled
  - Some control of rate of turn via radio

Flight 5 – November 18, 2006

- Flight 5 – Successful radio control!
  - Proof of concept
  - NPW-5 + servo can work

Rev 3 – High power, printed circuits, cameras



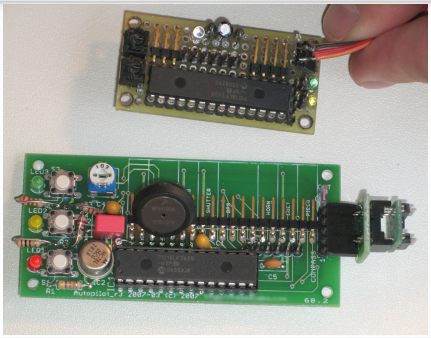
Lessons learned from Saturn Zero

- Higher altitude flights ➔ More descent time
  - Higher power
  - Electronic ejection ➔ Pressure altimeter
- Logging of flight events critical for debugging
- Hardware reliability
  - Soldered connections (no springs)
  - Potted connectors
  - Printed-circuit board (PCB) construction

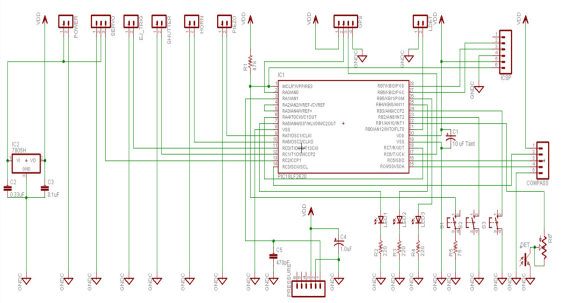
Rev 3 Flight Computer - Goals

- PCB construction – more robust
- Pressure altimeter – logging, electronic ejection
  - Also launch detection
- Drive on-board camera payload
  - Direct shutter trigger, infrared signal generation
- Complete log: Lat, Lon, Altitude, Status

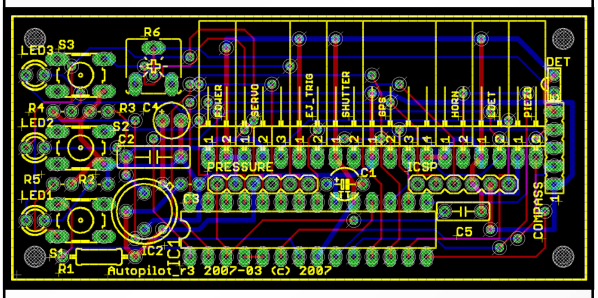
May 2007 – Rev 3 Flight Computer



Rev 3 Schematic (Eagle PCB)



Rev 3 PCB Layout (Eagle PCB)



PCB Fabrication

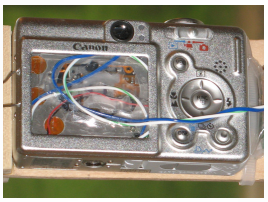
- BatchPCB.com
  - 4 boards (2 of these, 2 smaller ones), $55
  - Delivered (from China)
  - Directly from Eagle PCB output files
  - 2 layers
  - Silk screen both sides
  - Unlimited holes
  - Took 2 months (slow, but cheap)

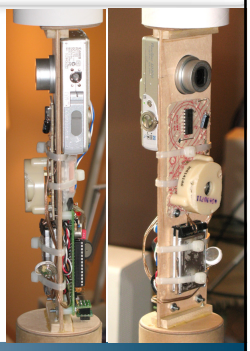
"Rev 3" power supply

- Main battery: 2 x LiIon cellphone cells (7.2v)
  - Or: 2 x NiMH 3.6v cellphone paks (7.4v)
  - Directly power servo (when present)
  - Directly power parachute ejection (flashbulb)
- PIC and GPS
  - Main battery regulated down to 5.0v by 7805
  - This later turned out to be unnecessary; PIC and GPS will both run OK on 3.6v (1 x LiIon or 3 x NiMH)
- Camera (when present)
  - Internal 3.6v LiIon cell (one)

Camera payload

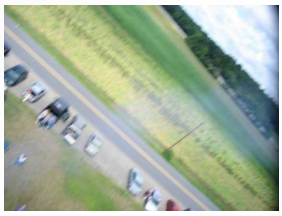
- Canon SD200, 3.2 Mpixel
  - About $30 on eBay (because of "broken screen")
  - Soldered wires onto shutter button





Canon SD200 images

- Lousy. Can't focus when camera moving fast.



- The rest are much worse.

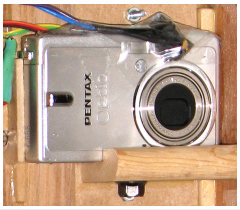
August 18, 2007

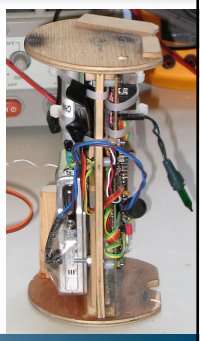
Some other stuff that didn't work

- Launch detection using a pin on launch rod
  - Very unreliable
  - Pressure altimeter works much better
- Honeywell HM55B electronic compass
  - Very slow, damped response
- Drilling out delay elements on HPR motors
  - A good reason to do only electronic ejection
- Fins that extend below base of rocket
  - They break off on landing
- "Drag disc"
  - Enough said

Things that worked better

- Pentax Optio S6 camera, 6 Mpixel
  - Can force infinity focus in "Landscape"
  - Also cheap with "broken screen"
  - Infrared remote shutter release
    - No soldering





How to trigger a camera via IR

- Connect IR LED to digital output
  - PIC outputs will source 25 mA, which is plenty
- Modulate IR LED at 48 kHz (square wave)
  - I did this using TMR2 and CCP2 on the PIC18
  - TMR2 ticks fast
  - CCP2 in PWM mode
- Emit ~32 bit code at 1 pulse/millisecond
  - For the Pentax, it's 10101010101010001111111111111 (right to left)
  - Look up the code you need using Google
  - Or reverse engineer a IR clicker with a oscilloscope

Pentax Optio 6 images

July 19, 2008

July 26, 2008 NYPOWER 2008, Geneseo NY

Pressure altimeter

- Freescale MPX4105A
  - Outputs analog voltage corresponding to pressure
  - Pressure(kPa) = (Vout/Vs + 0.09)*100
- Read voltage with a PIC ADC pin
  - 10 bit resolution, oversampled to 13 bits
    - Provides resolution around 1/2 foot of altitude
    - Thanks to Robert DeHate for that idea
- Then $z = \left(1 - \left(\frac{P_o}{101.325}\right)^{0.190263}\right) \times \frac{288.15}{0.00198122}$
  - Straight from http://en.wikipedia.org/wiki/Pressure_altitude

Firmware functions

- Detects launch within ¼ second
- Detects apogee within 1/25 second
  - Triggers parachute ejection
- Continuity check on ejection circuit
- Honks "Find Me" horn on & after descent
  - Salvaged from a broken smoke alarm
- Flashes LED on each GPS fix
- Piezo buzzer gives internal status (armed/safe)

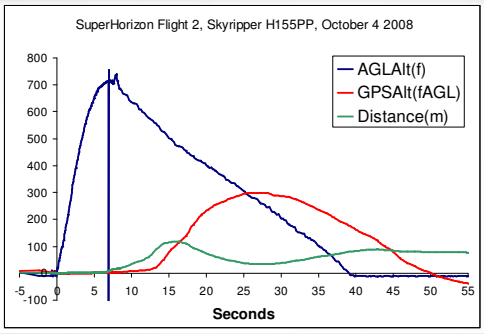
Logging functions

- Logs altitude & status
  - Every 3 seconds before flight
  - 25 times/second during flight
- Logs GPS Lat, Lon, Altitude, Speed, Course
  - Every 3 seconds before flight
  - 1 to 5 times/second during flight
- Trick: PIC stops execution during write to Flash
  - Can't respond to interrupts from incoming GPS data
  - Solution: Predict when GPS data will arrive
    - Buffer data until no GPS data expected soon,
    - THEN write data to Flash

More firmware functions

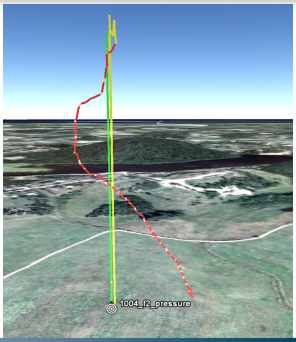
- Beeps out maximum altitude reached
- Test modes
  - Test pressure sensor
  - Simulated arming, "timed launch" for ground testing, apogee
- Simple state machine software design
  - C code, on naked hardware
  - No "real time kernel" or RTOS
    - I don't believe in them
      - Yes, I know they exist ☺

Analysis of logged data



Analysis of logged data

Start UTC 2008-10-04 18:20:32.000.
End UTC 2008-10-04 18:26:46.876. (374.9 s)
Log covers 369.2 seconds by RTC.
1 init records processed.
1436 status records processed (25 lagged).
6 absolute GPS records processed.
279 differential GPS records processed.
285 total GPS records (280 position fixes).
1722 total records processed.
RTC runs at 98.6% of real time. (1.4% slow).
Estimated ground level -56.3 feet MSL.
Ejection altitude 713.8 feet AGL.
Ejection at 6.8 seconds.
Average ascent speed 71.1 mph. (vertical)
Average descent speed 14.9 mph. (vertical)
Descent time (estimated) 33.0 seconds.
Descent wind speed (est) 4.4 mph.
Total flight time (est) 39.8 seconds.

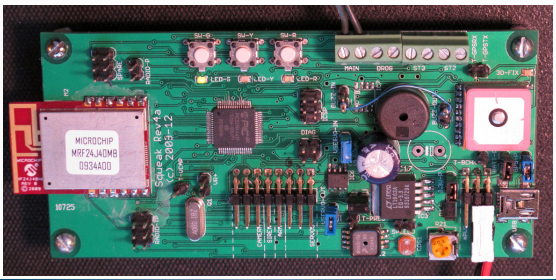


What can you do with that data?

- Google Earth – flight 2 of 2008-10-04
- Google Earth – flight 2 of 2008-11-01

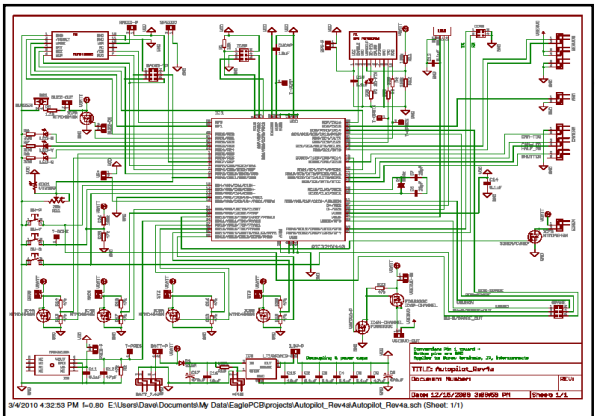
What I'm doing now (2010)

- "Rev 4a" Flight Computer



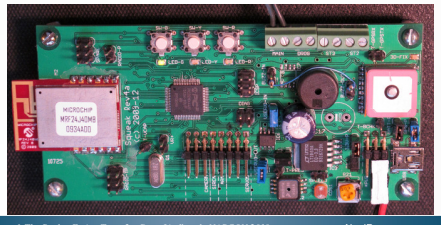
"Rev 4" will fly back to the launch pad

- This time for sure!
- Surface mount (SMD) components
- Integrated ejection / igniter firing
  - Drogue, main parachutes
  - 2nd and 3rd stage in-flight electronic ignition
- Integrated piezo buzzer



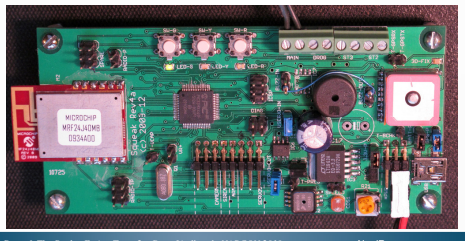
"Rev 4" Flight Computer

- New PIC32 (MIPS) 32-bit MCU
  - 40x faster @ same clock speed
  - More RAM (64 kBytes), Flash ROM (512 kBytes)
  - Will probably allow floating-point nav calculation



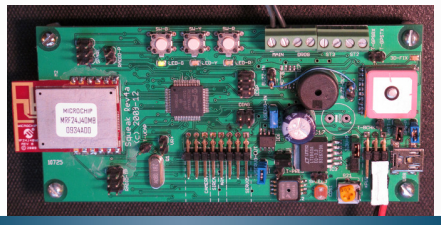
"Rev 4" Flight Computer - 2

- On-board GPS (5 Hz fixes)
- IEEE 802.15.4 telemetry radio (Zigbee/MiWi)
- USB interface to PC (dump flight log)



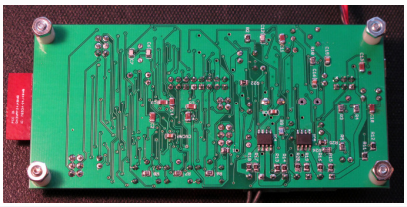
"Rev 4" Flight Computer - 3

- 2nd UART for debugging
- Simplified & improved power supply
  - *Remove* shorting pin to turn ON



"Rev 4" Flight Computer - 4

- Maybe – Gated launch igniter current
  - Prevent launch without arming electronics
  - Provide a T=0 signal from LCO
  - Use to start high-speed camera before liftoff



Still to-do

- Rev4b PCB
  - Smaller, lighter, simpler
- Port navigation code onto Rev4 hardware
  - Runs only in MATLAB/Octave on PC now
- New cameras – Canon SD400-1100 w/CHDK
  - CHDK will support fixed focus & exposure
  - Much faster frame rates
  - Higher resolution, better optical quality
  - Dedicated, soldered, shutter connection

2010

- Automatic return to launch site???
  - No promises (still).

THANK YOU

- Apologies to:
  - Stephen Spielberg & George Lucas
  - www.RocketTshirts.com (rocket eating tree)
  - Bullwinkle J. Moose
- Watch **NerdFever.com** for progress reports

Appendix

Suppliers (1)

- eBay.com – Computers, cameras, etc…
  - First place to look for everything
- Goldmine-Elec.com – Cheap surplus parts
- Futurlec.com – Cheap commodity parts
- MPJA.com – Parts, tools (soldering esp.)
- BatchPCB.com – Custom PC boards (cheap)
- SparkFun.com – GPS, power supply, parts
  - Not so cheap, but lots of cool stuff & well organized

Suppliers (2)

- HarborFreight.com
  - Cheap tools (and they're cheap)
- Microchip.com – MCUs
  - Free MPLAB IDE environment, C compliers
  - Generous free sample policy
- Freescale.com – Pressure sensors
  - Free samples often available
- Mouser.com – General industrial electronics
  - No minimum order, reasonable shipping
  - When you can't get it on eBay, surplus, or free

Most useful tools

- Soldering iron, mini-vise (MPJA.com)
- Stationary belt/disc sander ($99, Home Depot)
- Scroll saw ($85, HarborFreight.com)
- DMM ($5, Futurelec.com)
- Bench power supply (Mastech 3005 or similar)
- Oscilloscope (eBay, $60 or so)
- Eagle PCB (free PCB layout software)
  - Cadsoft.de
- Microchip ICD2 clone ($60, eBay)