Vehicle Data Logging Device

Group 17

Sponsored by

Dr Yiannis Papelis

Kyle Fiducia (EE)

Joshua Mahaz (CpE)

Graham Smith (EE)

April 20 2007

Abstract

- Sponsored by Dr. Papelis (UCF CECS)
- Build a compact hardware device that logs data acquired by various sensors on a passenger vehicle
- Vehicle data to be logged includes: geographic location, engine RPM, throttle position, acceleration, speed, yaw rate, and forward looking video
- Device must have an easy way to retrieve the data
- Must allow for widespread data gathering usage
- Device must be portable and easy to install

Motivation

- Dr. Papelis' Motivation:
  - 45,000 automobile accident deaths each year, of which 90% involve driver error.
  - To provide real world data for input into his research.
  - Extensive amounts of logged data to study driver behavior and develop metrics for driver performance.

- Group 17's Motivation:
  - Solid-state device with current standard components
    - Working with standard hardware will be much more versatile than learning a highly specialized task to complete a project.
  - It is also encouraging that if the design is good enough, the device could be actually used after the project is done, rather than just a proof of concept.
  - Sponsored - meaning little money out of pocket.

Vehicle Data Logging Device

- **Data Requirements:**
  - Geographic position
  - Velocity
  - Throttle-position
  - Engine RPM
  - Lateral & longitudinal acceleration
  - Forward looking video images (cost dependent)
  - Yaw rate (optional)
  - Following distance (optional)

- **Administrative Requirements:**
  - Allow for easy data retrieval
  - User Interface
  - Small, portable, easy to install device
  - Maintain system cost reproduction below ~$400
  - Intelligent Power Supply
    - Draws power only when the engine is running

Project Specifications

	Capture Frequencies	
Data	Minimum	Optimal
OBDII - Speed, RPM, Throttle Position	1 Hz	3 Hz
Lateral & Longitudinal Acceleration	>10 Hz	10 Hz
GPS Position, Speed, Time	1 Hz	1 Hz
Forward Looking Video Images	>1 Hz	2 Hz
Yaw Rate	>5 Hz	10 Hz
Following Distance	>1 Hz	5 Hz

Vehicle Data Logging Device

Hardware Requirements

- OBDII Interface (On Board Diagnostics)
  - Vehicle Speed, RPM, Throttle Position (if avail.)
- Accelerometer (dual-axis)
  - Lateral and longitudinal acceleration
- GPS Sensor
  - Vehicle position and accurate time
- Data Storage Device
  - To store all the logged data

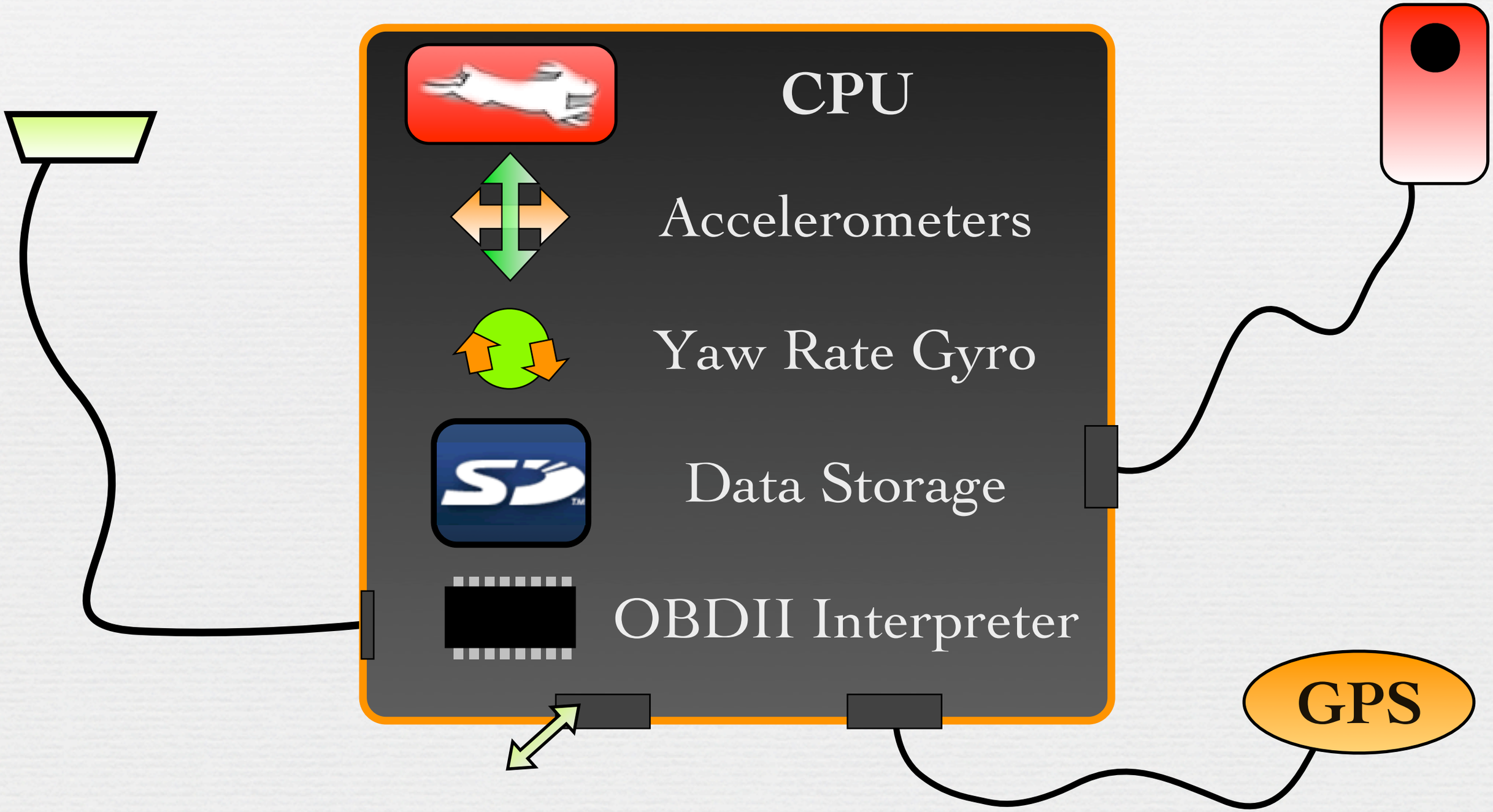
- Image Sensor
  - Captures forward looking video
- Yaw Rate Sensor
  - Measures turning rate of the vehicle
- Range-finder
  - Measures following distance

} Optional

Vehicle Data Logging Device

Hardware Components

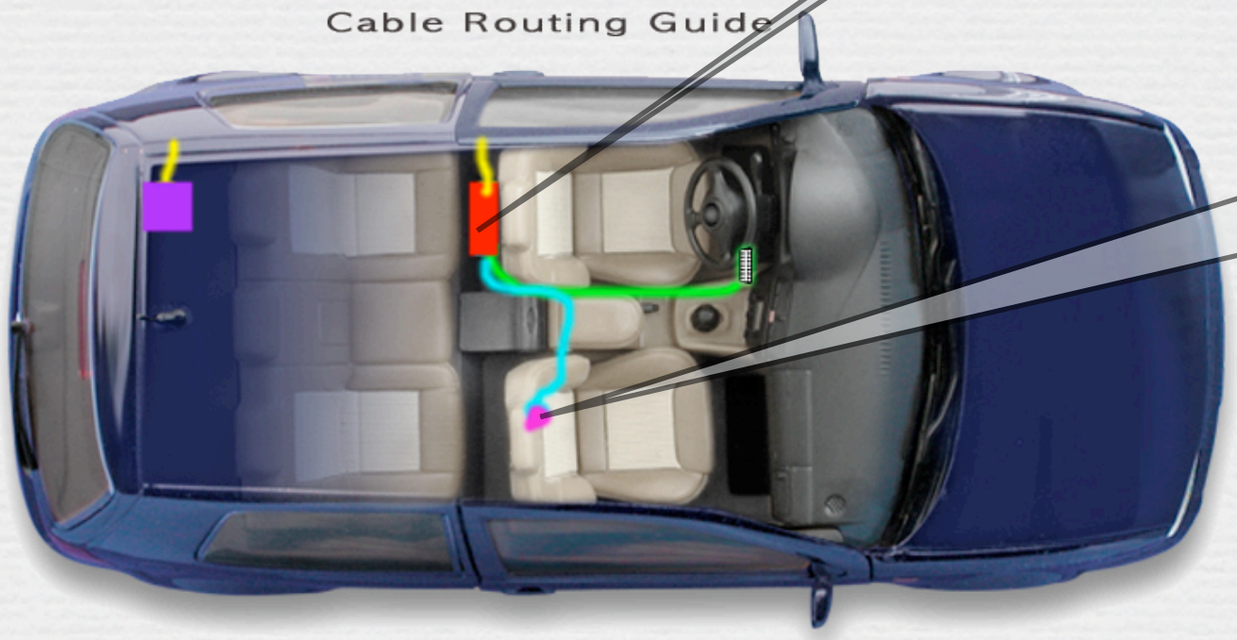
Block Diagram



Vehicle Data Logging Device

Component Layout

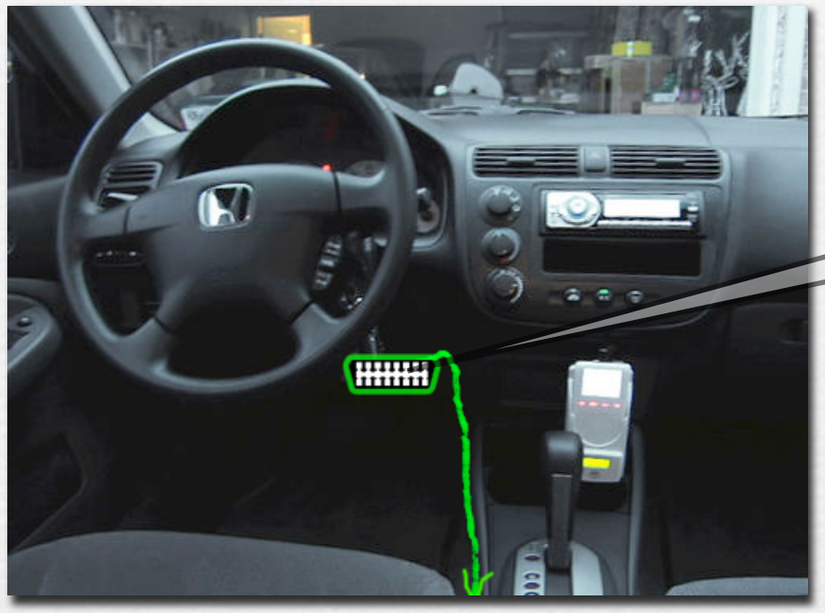
Cable Routing Guide



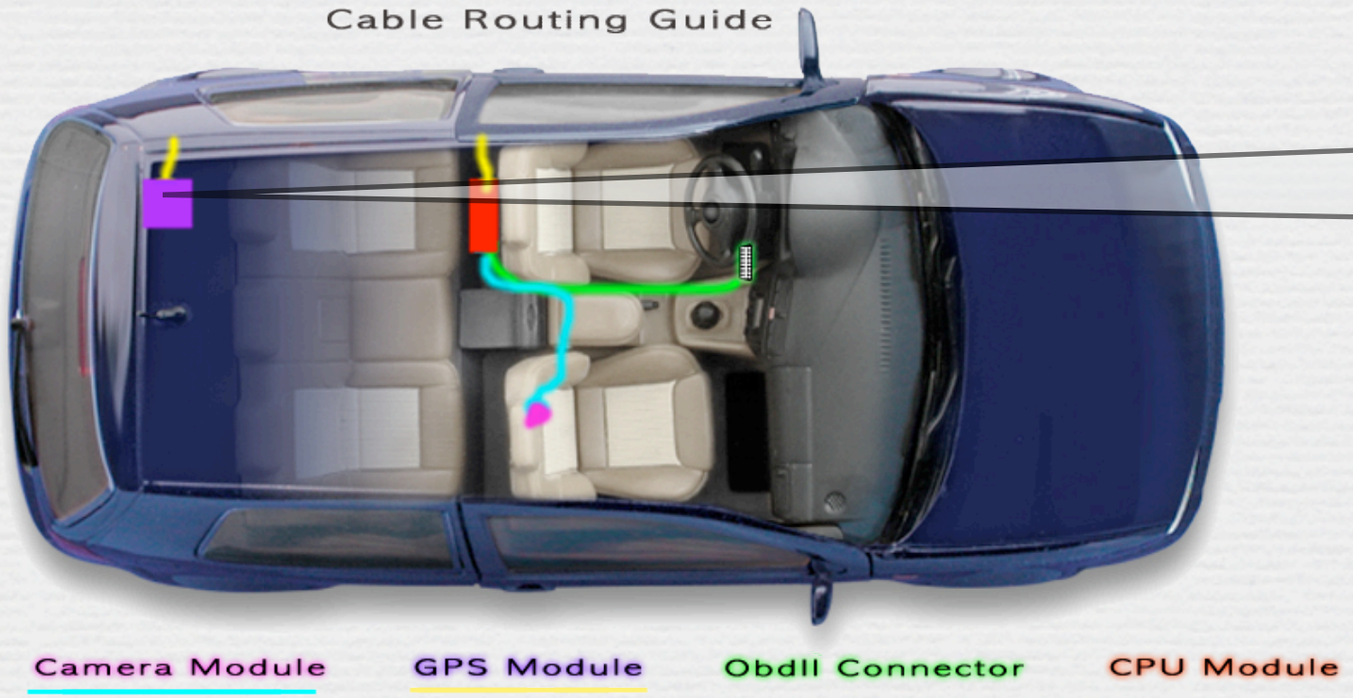
Device CPU placed on the back seat floor

Camera placed in view of driver and road; Attached to headrest of passenger seat

Component Layout

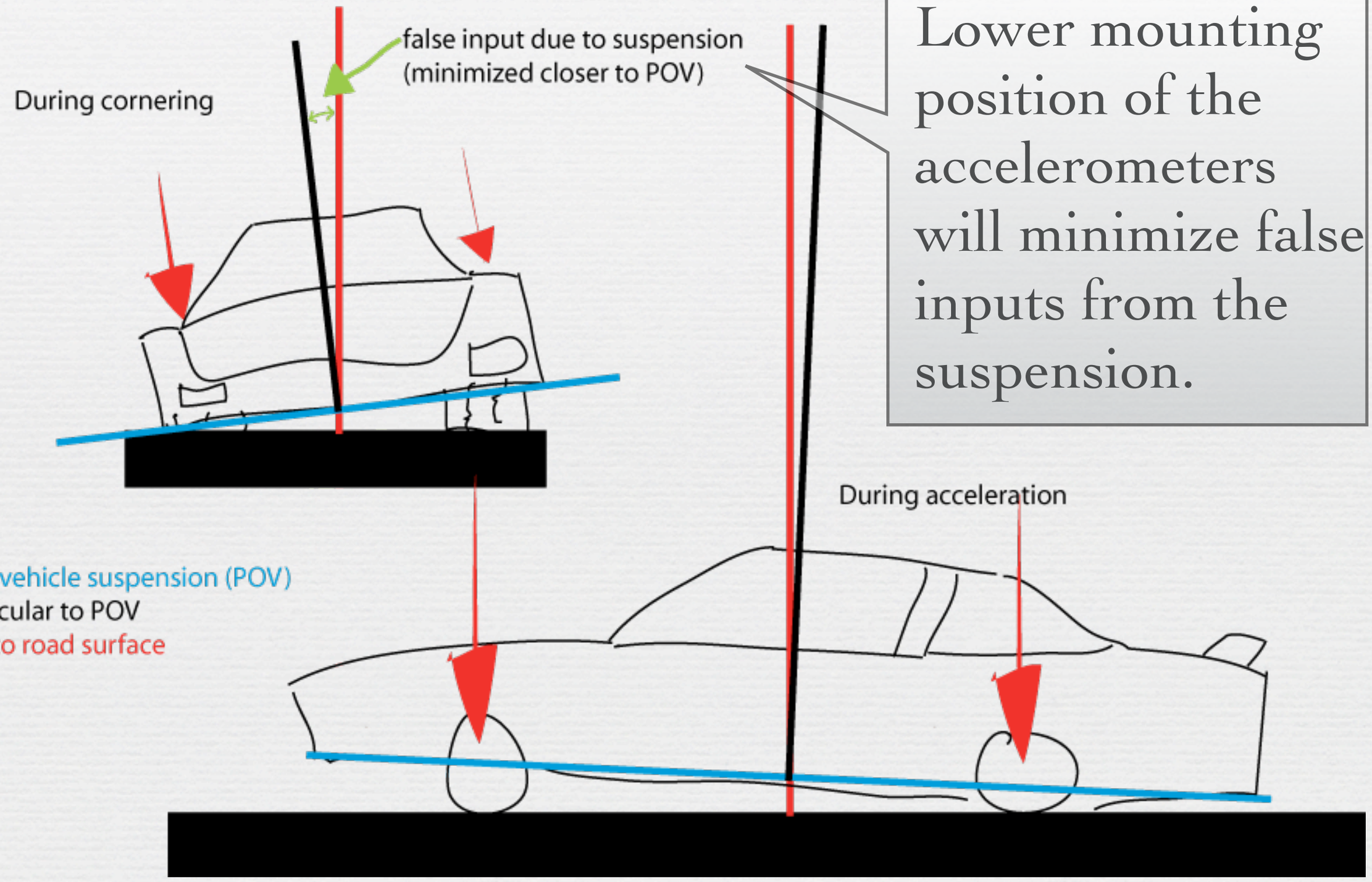


OBDII Standard requires port to be within 36” of steering wheel.



GPS device mounted on roof for clear view of sky and proper ground plane.

Mounting Location Constraints



Lower mounting position of the accelerometers will minimize false inputs from the suspension.

Data Logging Constraints

Image Capture Constraints

Resolution	Approx File Size (KB)	Approx Number of Images/Capacity(MB)				
Capacities (MB) ---->		128	512	1024	2048	4096
VGA	32.3 KB	4,058	16,232	32,464	64,927	129,855
QVGA	11.0 KB	11,967	47,869	95,739	191,477	382,954
160x120	6.8 KB	19,361	77,443	154,886	309,771	619,543
Text	0.2 KB	613,788	2,455,154	4,910,307	9,820,615	19,641,230
Hours of Image Capture @ 1Hz						
640x480	VGA	1	5	9	18	36
320x240	QVGA	3	13	27	53	106
160x120	sub-QVGA	5	22	43	86	172
Text		170	682	1364	2728	5456

Image Capture Constraints

Resolution	Approx File Size (KB)	Approx Number of Images/Capacity(MB)				
Capacities (MB) ---->		128	512	1024	2048	4096
VGA	32.3 KB	4,058	16,232	32,464	64,927	129,855
QVGA	11.0 KB	11,967	47,869	95,739	191,477	382,954
160x120	6.8 KB	19,361	77,443	154,886	309,771	619,543
Text	0.2 KB	613,788	2,455,154	4,910,307	9,820,615	19,641,230
Hours of Image Capture @ 1Hz						
640x480	VGA	1	5	9	18	36
320x240	QVGA	3	13	27	53	106
160x120	sub-QVGA	5	22	43	86	172
Text		170	682	1364	2728	5456

Storage Media Comparison

Media Type ->	CompactFlash		SmartMedia	MMC	
Varieties	I	II		MMC	RS-MMC
Maximum storage capacity, MB	8000	12000	128	4096	512
Theoretical maximum capacity	137 GB	137 GB		128 GB	
Data read speed, MB/s	40	40	2		
Data write speed, MB/s	40	40			
Read/write cycles	∞	∞	1,000,000	1,000,000	∞

Media Type ->	Sony Memory Stick (proprietary - not much hardware avail)			
Varieties	Standard	Pro	Pro Duo	Micro
Maximum storage capacity, MB	128	4096	4096	
Theoretical maximum capacity	128 MB	32 GB		
Data read speed, MB/s	2.5	20	20	20
Data write speed, MB/s	1.8	1.8	10	
Read/write cycles	∞	∞	∞	∞

Benefits of SD

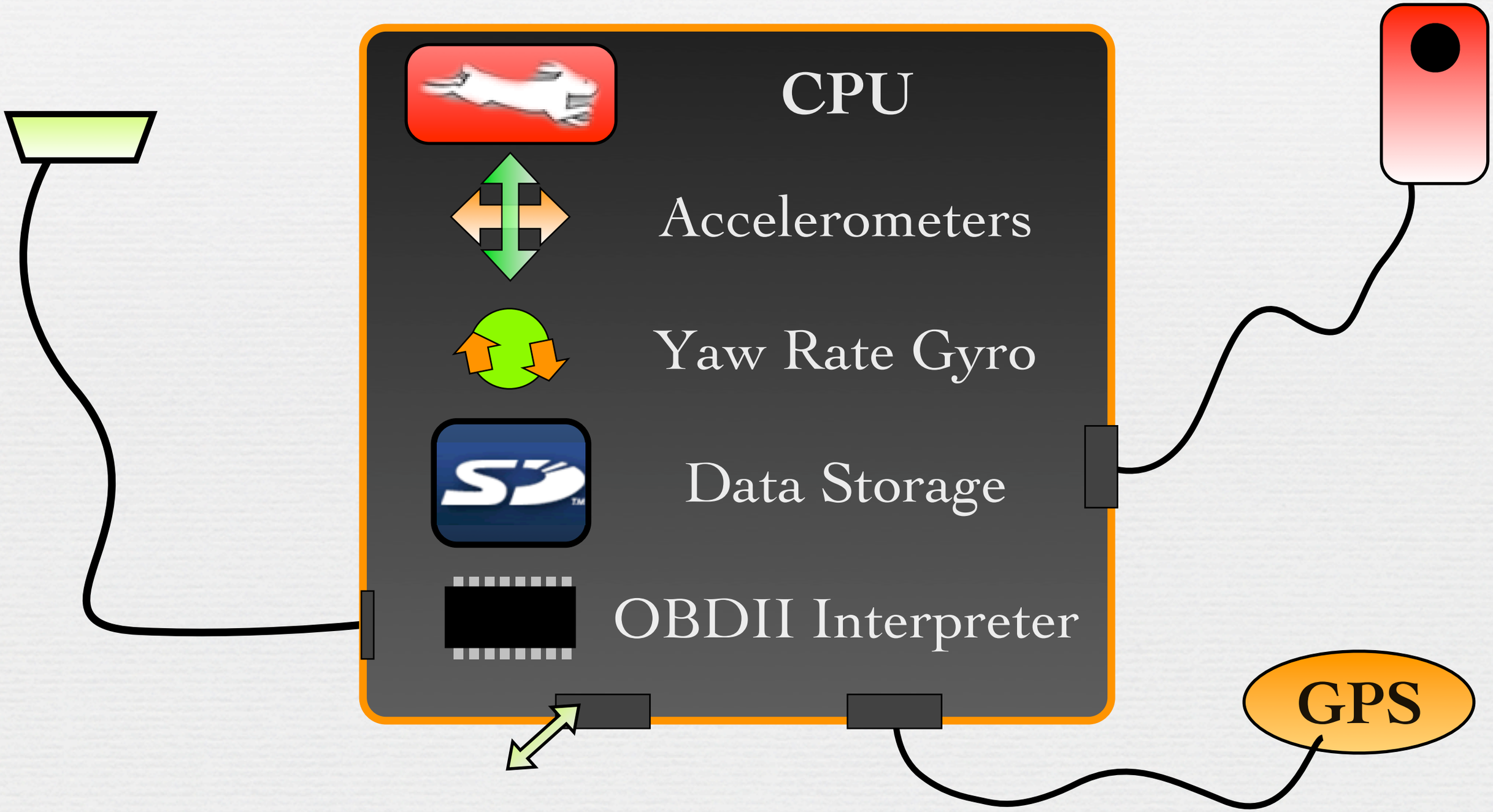
- Uses less power than CF
- Lots of 3rd party hardware available
- Smaller
- Inexpensive cards in many sizes

Media Type ->	xD			Secure Digital (SD)		
Varieties		Type M	Type H	SD	miniSD	microSD
Maximum storage capacity, MB	512	2048	1024	4096	2048	2048
Theoretical maximum capacity	512 MB	8 GB	8 GB	128 GB		
Data read speed, Mb/s	5	4	15	20		
Data write speed, Mb/s	3	2.5	9	20		
Read/write cycles	∞	∞	∞	∞		

Hardware

Hardware Components

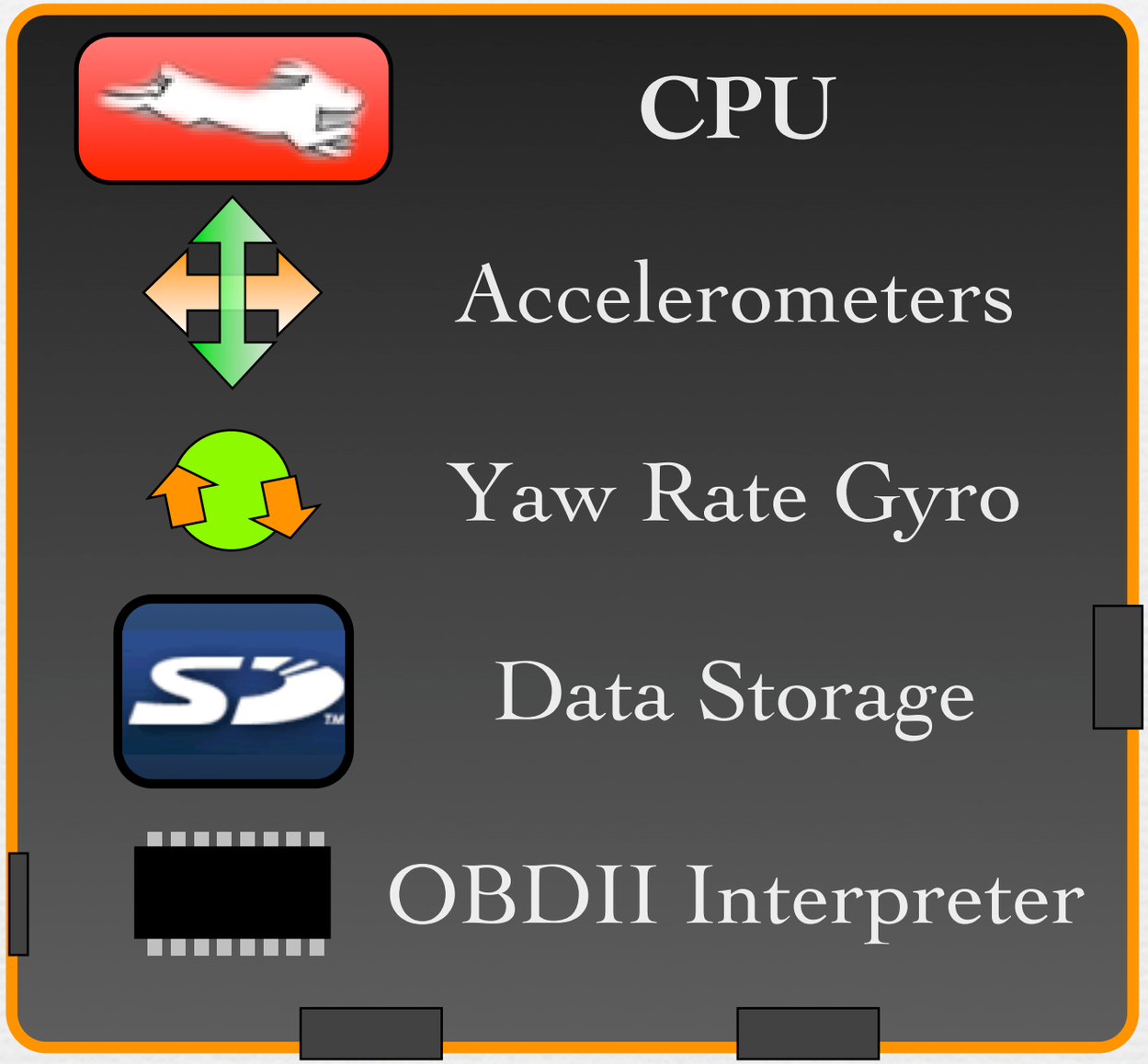
Block Diagram



Vehicle Data Logging Device

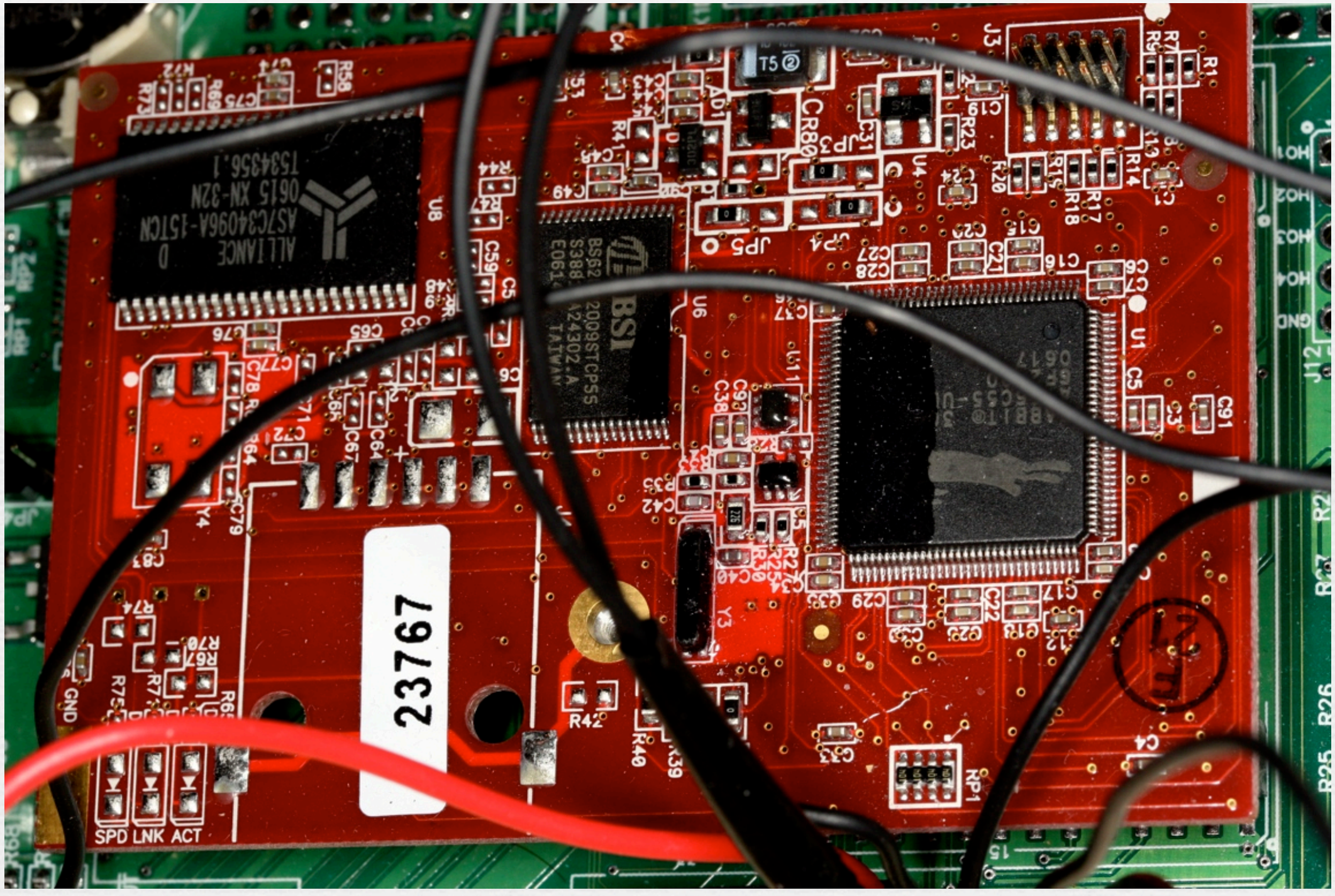
Hardware Components

Block Diagram



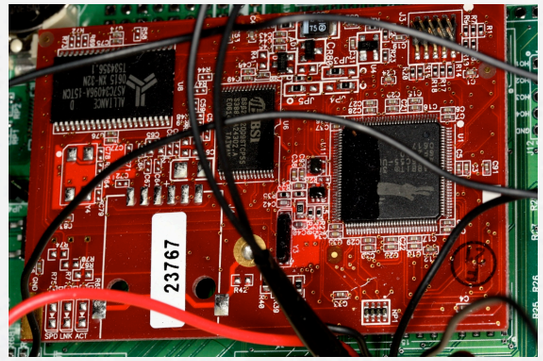
Vehicle Data Logging Device

Micro-Controller



Vehicle Data Logging Device

Micro-Controller



Rabbit 3220

Design Considerations

- **Speed - 44MHz**
  - Adequate speed for data throughput
    *Image capture is largest constraint
- **I/O Lines**
  - Must have adequate I/O lines to support all devices (33)
    Rabbit has 52 I/O Pins
- **Protocol Support**
  - I^2C, RS232, SPI, PWM Input Capture
- **Programming Language**
  - Assembly, Dynamic C
- **Cost**
  - Rabbit is actually a bit expensive, $80.00, but Dr Papelis wanted to use it.
  - PIC is a viable option at $10/unit

Input/Output Lines

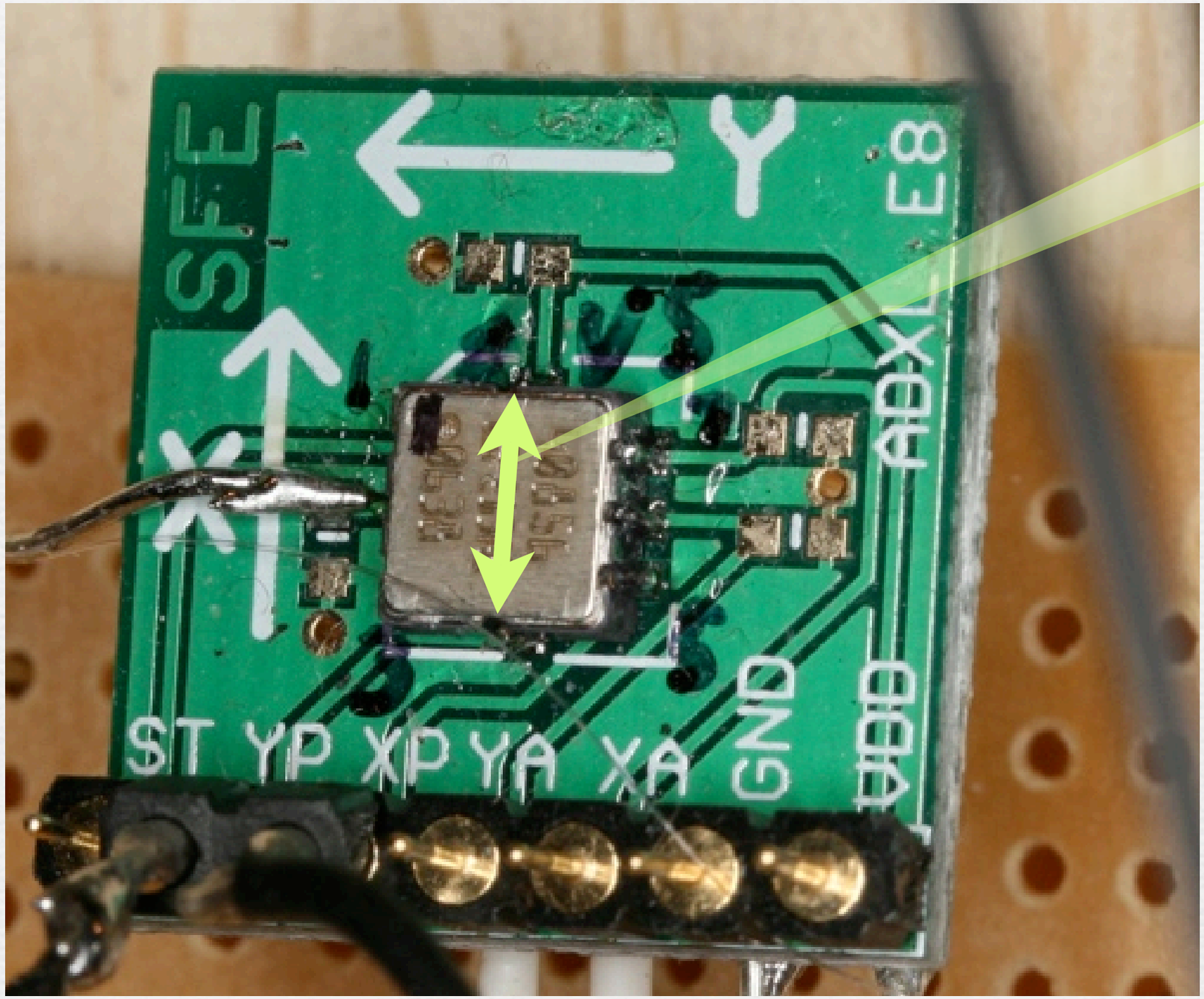
Device	Communication	Data Pins	Other	Total Req
GPS Device	Serial Data	2	0	2
Camera	UART	2	0	2
2-Axis Accelerometer	PWM	2	0	2
Yaw Rate Gyro	SPI	3	1	4
ELM327	Serial	2	2 (Busy/RTS)	4
Diagnostics Port	Serial	2	0	2
Future Expansion	Serial	4	2	6
Input Total				22
uALFAT	UART/SPI	6	0	6
UI LEDs (5)	PWM	0	5	5
Output Total				11
Total				33

Accelerometer

Lateral & Longitudinal Accelerations

ADXL213

shown on breakout board



5mm

Vehicle Data Logging Device

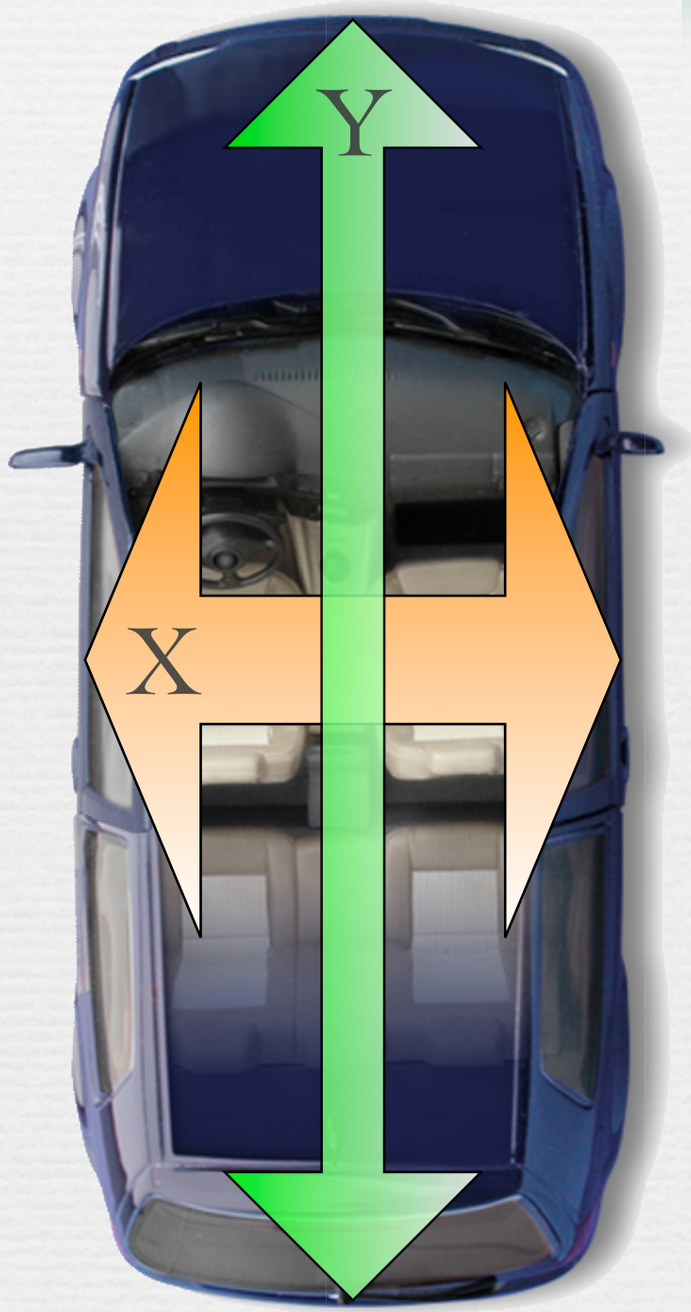
Accelerometer

Lateral & Longitudinal Accelerations

ADXL213

Design Considerations

- **Dual axis**
- **PWM output**
  - Built-in ADC
- **Range**
  - ±1.2g
- **Cost**
  - $40.00



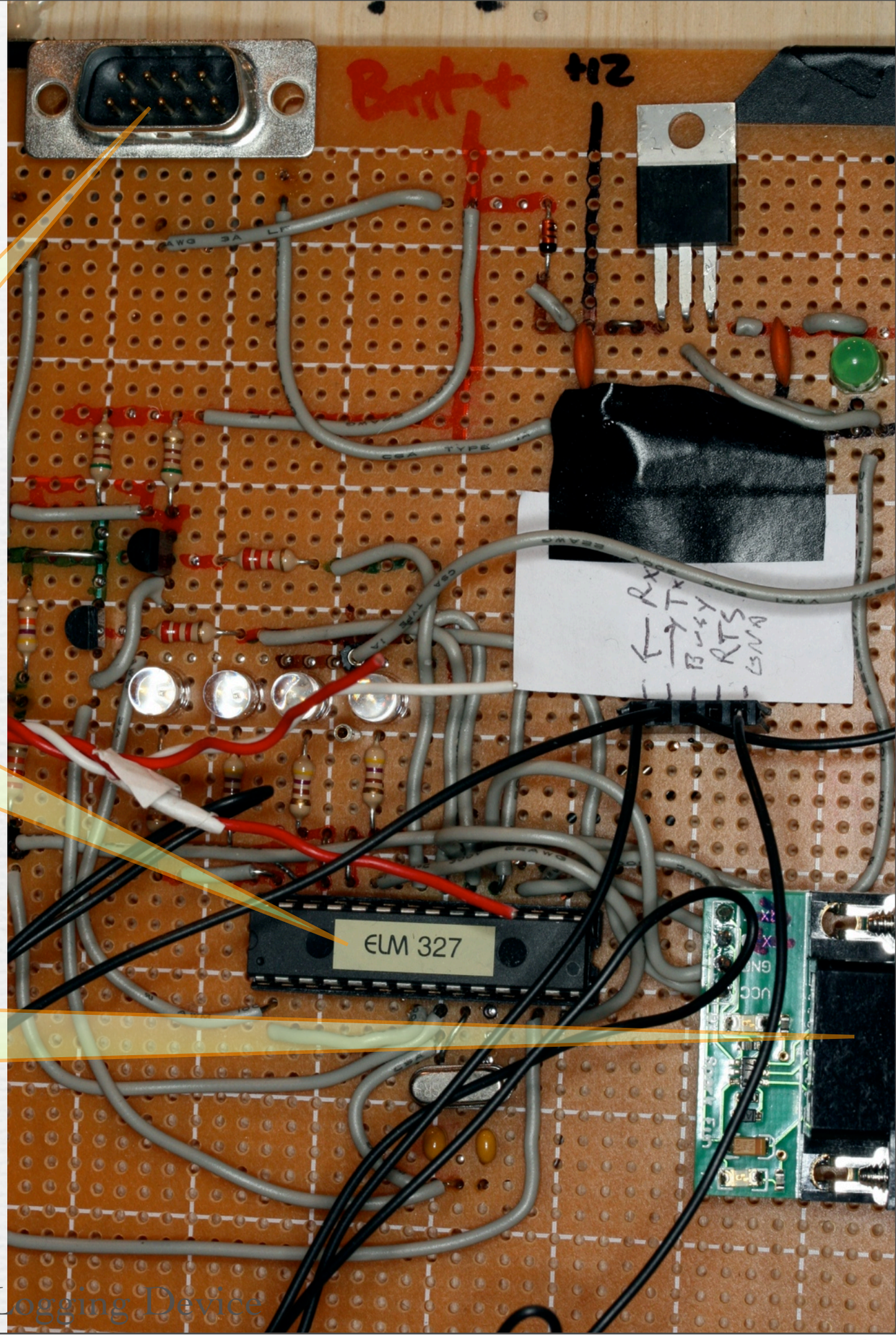
Vehicle Data Logging Device

OBDII Interface

To car's OBDII port

OBDII Interpreter

Data to µController



OBDII Interface

Vehicle RPM & Throttle Position

Design Considerations

- **Compatibility**
- **Ease of Use**
  - Uses standard AT command syntax
  - RS232
  - Handles all the bus initiation and detection itself automatically
- **Cost**
  - $30.

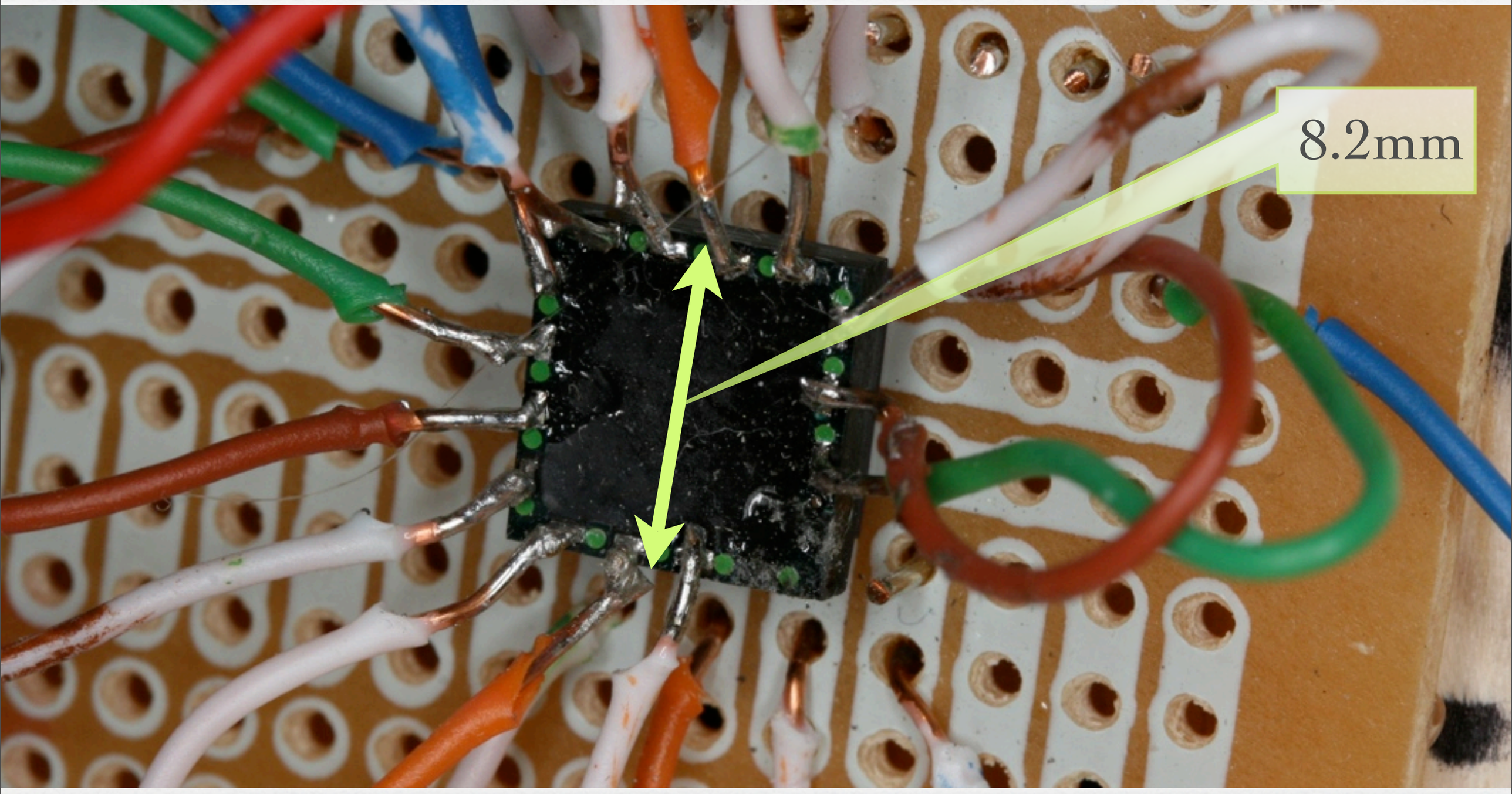
ELM327

Compatible with all OBDII standard protocols:

SAE J1850 PWM	Ford
SAE J1850 VPW	GM
ISO 9141-2	Chrysler, European, Asian
ISO 15765 CAN	all after 2008
ISO 14230	

Yaw-Rate Gyro

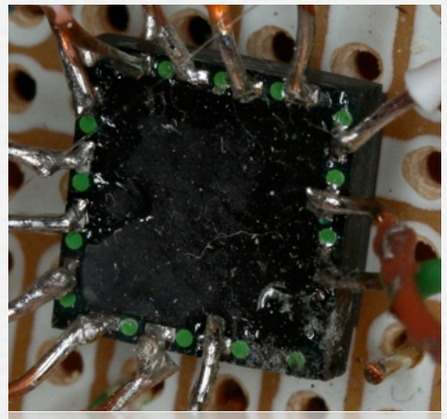
ADIS16100



Vehicle Data Logging Device

Yaw-Rate Gyro

Vehicle Turning Rate



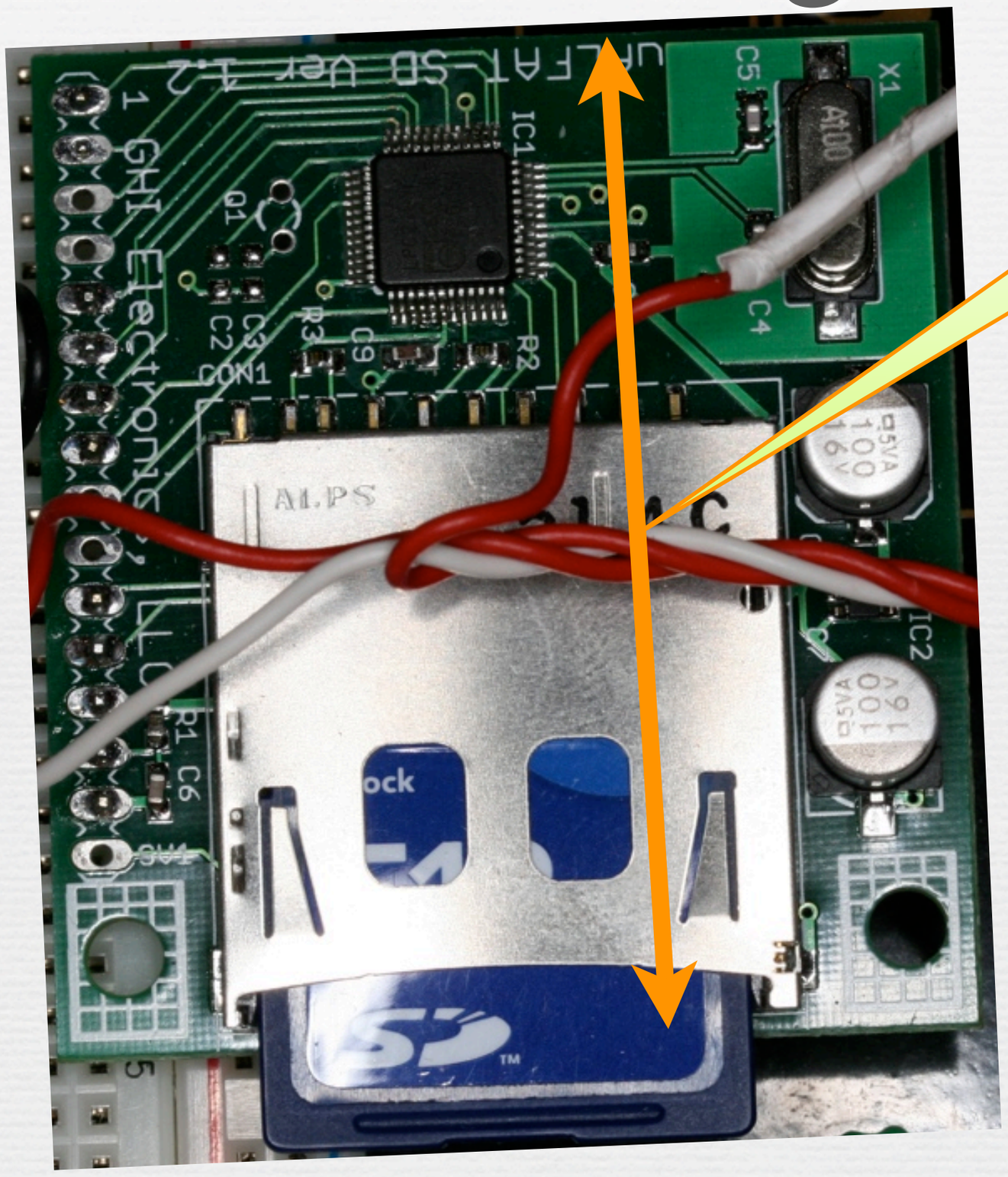
ADIS16100

Design Considerations

- **Built-in ADC**
  - Reduces number of components on the board
- **Resolution**
  - 360°/second
- **Protocol Support**
  - SPI Interface is fast and relatively easy
- **Number of Axes**
  - Single axis measurement (yaw)
- **Cost**
  - A bit expensive at $50 each
- **Difficulties**
  - LGA16

Vehicle Data Logging Device

Data Storage Device



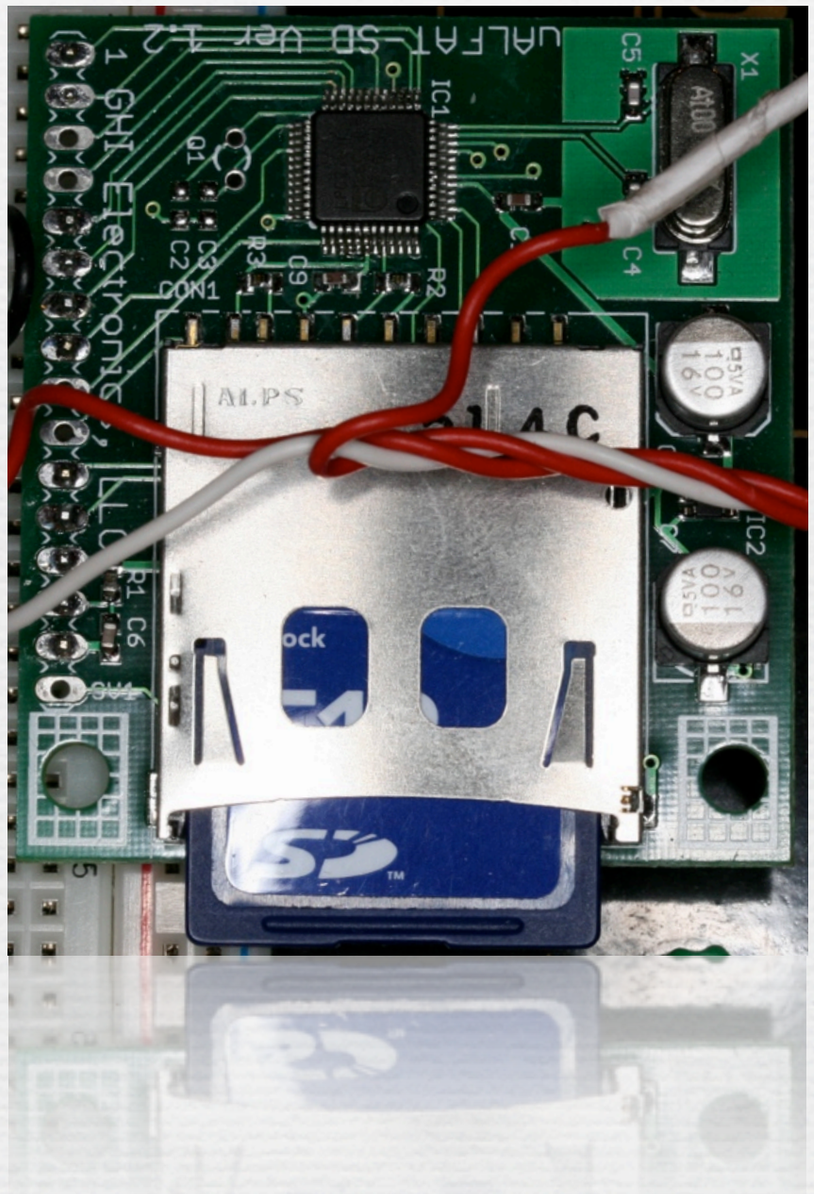
Vehicle Data Logging Device

Data Storage Device

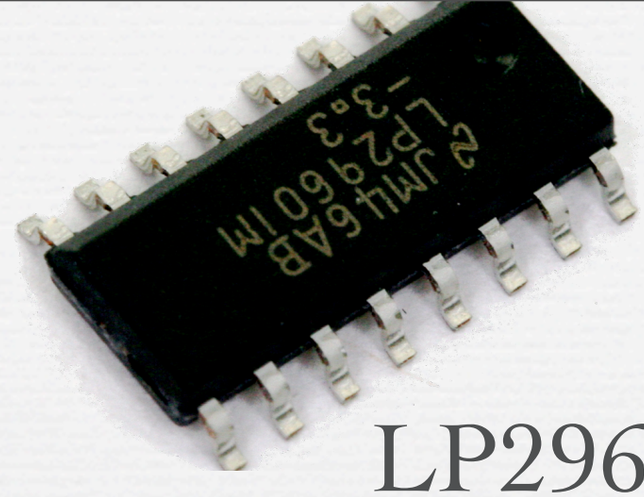
µALFAT

Design Considerations

- **File System Format**
  - FAT16
- **Speed**
  - 57.5 KB/s write speed
- **Physical Media Format**
  - Removable SD Card
- **Communication**
  - RS232
- **Cost**
  - $40.00



Power Supply

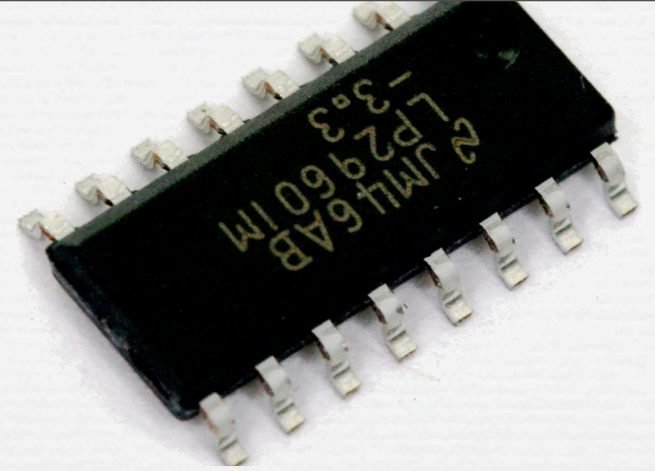


LP2960

<u>Design Considerations</u>

- **Input vs Output Voltages**
  - 11-15 VDC from OBDII port must be converted to a stable 3.3 and 5 VDC for our components.
- **Logic Level Shutdown**
  - Will shut down PSU and attached devices with one pin
- **Cost**
  - $3.50
- **Internally Fused**
  - 500mA internally limited current
- **Automatic Thermal Limiting**
  - Will shut down before heat damages components
  - A well-tested feature

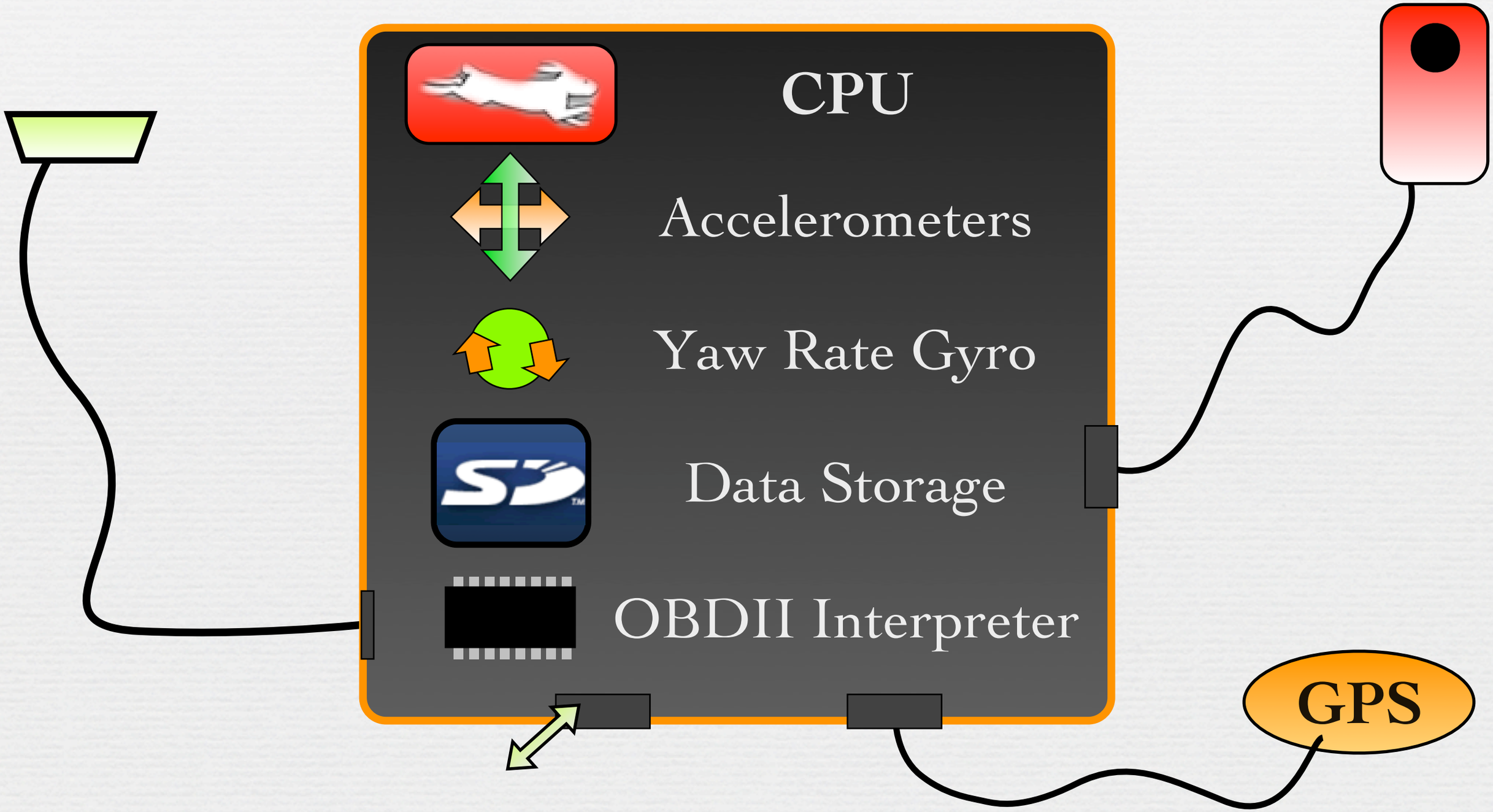
Power Requirements



Part	Voltage (V)	Current (mA)	Power (mW)
Micro-Controller	3.3	255.0	841.5
Storage	3.3	5.0	16.5
Camera	3.3	15.0	49.5
Accelerometers	5.0	0.7	3.5
OBDII IC	5.0	9.0	45.0
GPS	5.0	60.0	300.0
Yaw Rate Gyro	5.0	7.0	35.0
Total			1291

Hardware Components

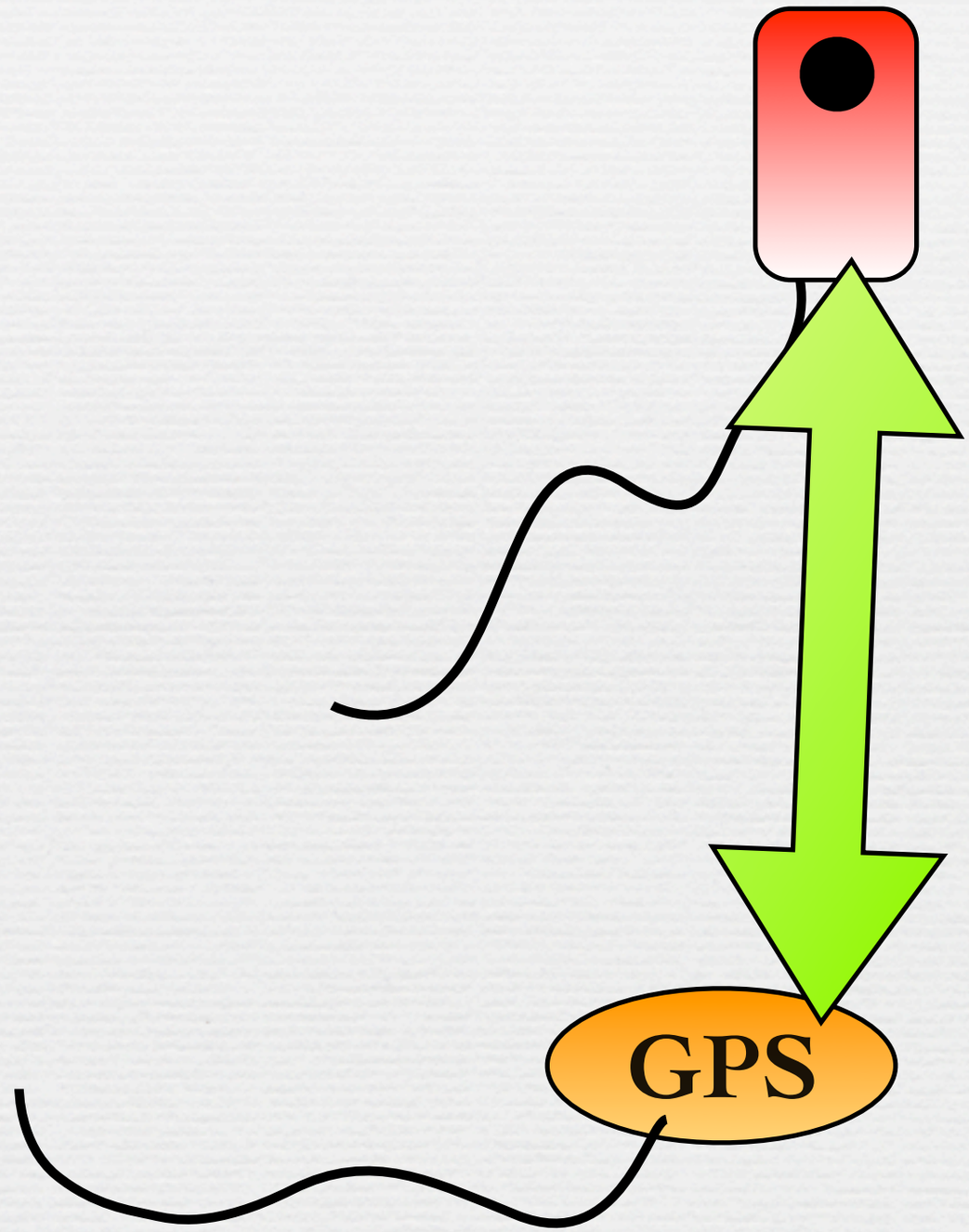
Block Diagram



Vehicle Data Logging Device

Hardware Components

Block Diagram



Vehicle Data Logging Device

Garmin GPS18

GPS Sensor

Geographic Position & Speed

Design Considerations

- **Update Frequency**
  - 1 Hz
- **Communication**
  - Standard NMEA over RS232
- **Time To First Fix**
  - ~40 seconds (cold start)
- **Physical Design**
  - Weatherproof, magnetic enclosure
- **Cost**
  - Already had one; $80 to buy

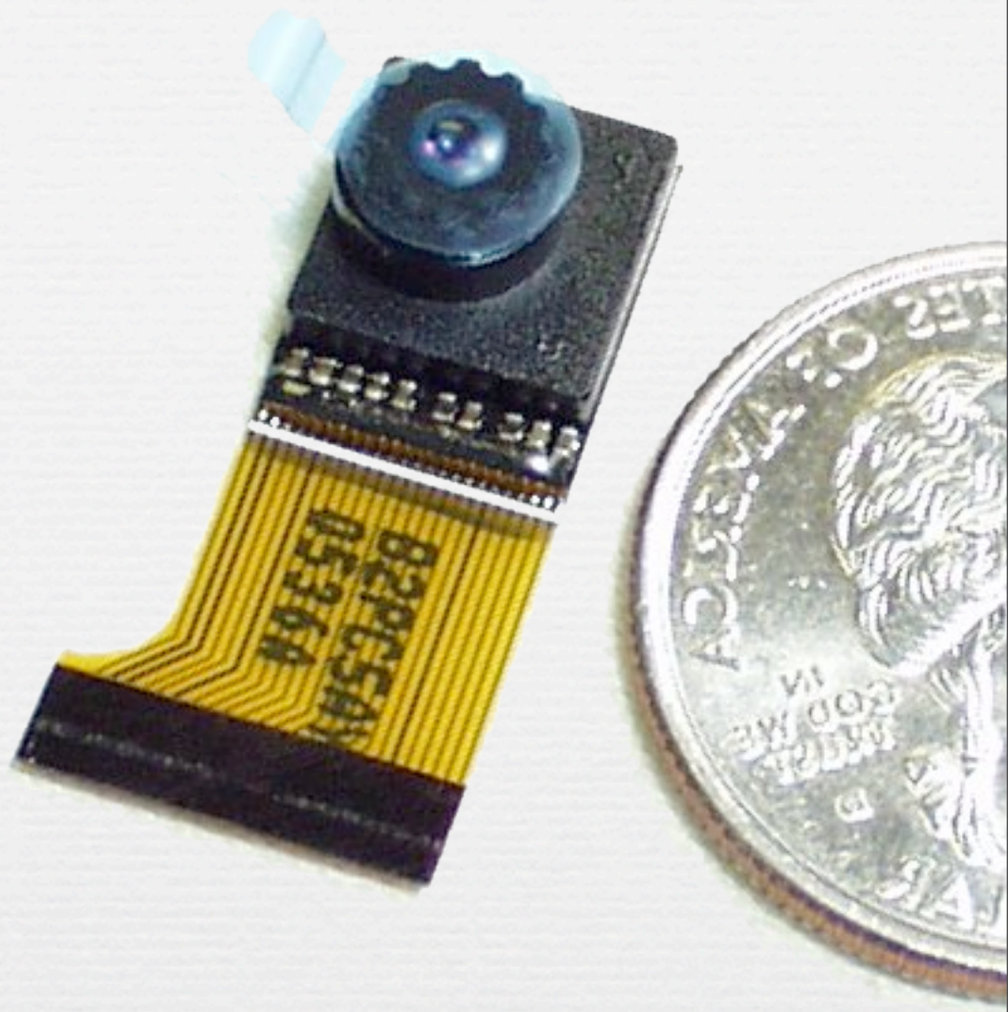
Camera / Video Capture

Design Considerations

- **Resolution**
  - VGA, QVGA, CIF, QCIF
  - Must scale to lower resolutions to enable higher frame rates with the same throughput.
  - Configurable on-the-fly

- **Frame Rate**
  - Only limited by communications speed

- **Video Format/Compression**
  - JPEG output

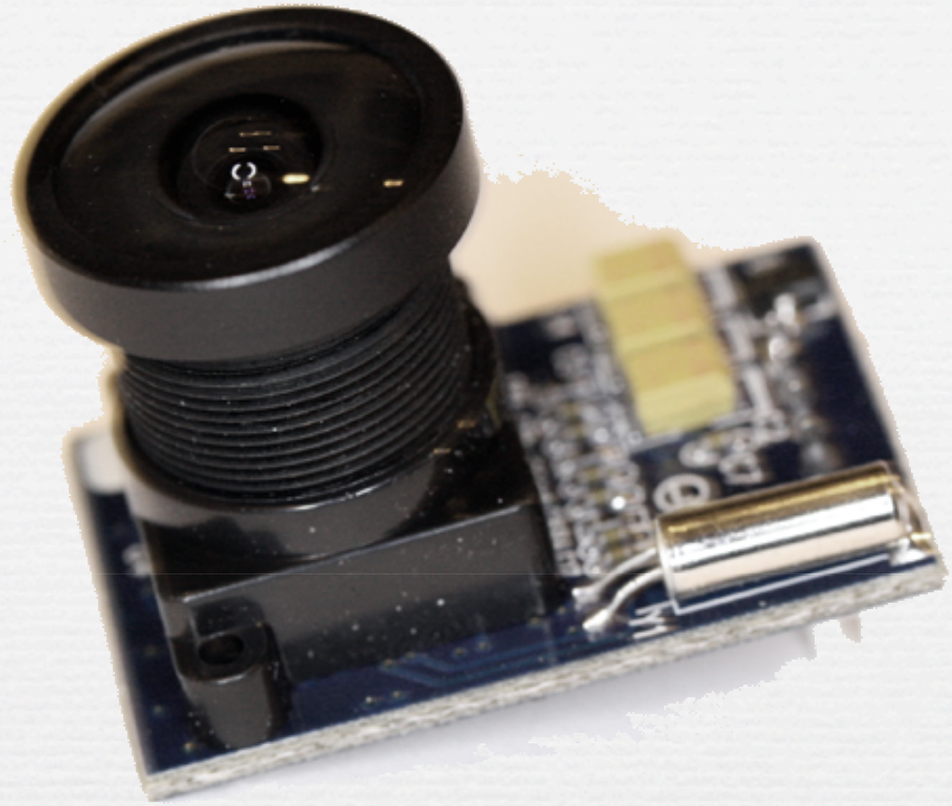


- **Communications**
  - RS232 @ 115.2Kb/s max
- **Cost**
  - $50./module
- **Wide Field of View**
  - 118°

Camera / Video Capture

Design Considerations

- **Resolution**
  - VGA, QVGA, CIF, QCIF
  - Must scale to lower resolutions to enable higher frame rates with the same throughput.
  - Configurable on-the-fly



C328-2225BW

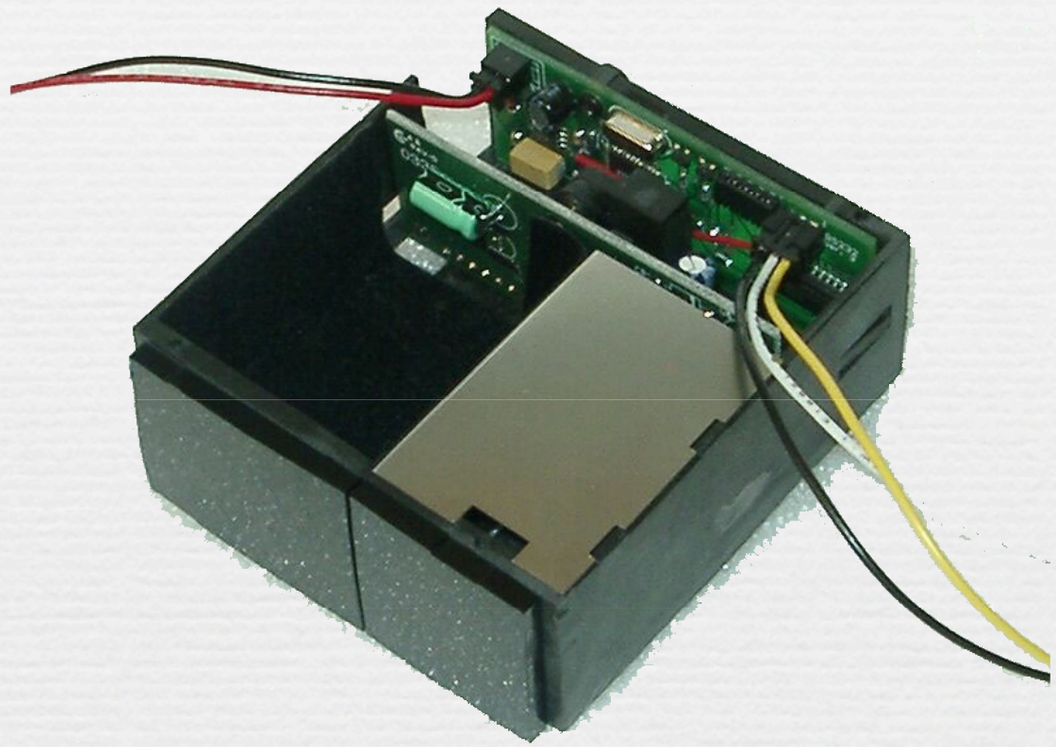
- **Frame Rate**
  - Only limited by communications speed
- **Video Format/Compression**
  - JPEG output
- **Communications**
  - RS232 @ 115.2Kb/s max
- **Cost**
  - $50./module
- **Wide Field of View**
  - **118°**

Range Finder

Following Distance

Design Considerations

- **Cost**
  - Laser - $600
  - Microwave - $10,000
  - Budget still only $400
- **FCC Regulations**



Final Product Package

- Final design is implemented on a custom fabricated printed circuit board (PCB)
- PCB is fabricated so the optional components can be added to the board or not.
  - Allows for lower cost devices with reduced functionality
- The software determines whether or not the sensors are present and log data accordingly.
  - Intelligent software control to reduce user interaction
- Enclosure includes active cooling hardware

Software

Software Design

- Sensor Boot Sequences w/ Error Handling
- User Interface
- Polling Multiple Sensors
- Store to Removable Media Device

Sensor Boot Sequence

- Error handling
- Call initialization function
- Pull test data
- Validate data
- Return passed/failed

```
while(testCounter<15)
{
    ++testCounter;
    AccelInit();
    getAccel();
    if(Accelleration.x>=-10 &&
       Accelleration.x<=10 &&
          Accelleration.y>=-10 &&
             Accelleration.y<=10)
       {
          passed=1;
          break;
       }
    }

    if(passed)
       return 1;
    else
       return 0;
}
```

Sensor Boot Sequence

- Error handling
- Call initialization function
- Pull test data
- Validate data
- Return passed/failed

```
while(testCounter<15)
{
    ++testCounter;
    AccelInit();
    getAccel();
    if(Accelleration.x>=-10 &&
       Accelleration.x<=10 &&
          Accelleration.y>=-10 &&
             Accelleration.y<=10)
       {
          passed=1;
          break;
       }
    }

    if(passed)
       return 1;
    else
       return 0;
}
```

Sensor Boot Sequence

- Error handling
- Call initialization function
- Pull test data
- Validate data
- Return passed/failed

```
while(testCounter<15)
{
    ++testCounter;
    AccelInit();
    getAccel();
    if(Accelleration.x>=-10 &&
       Accelleration.x<=10 &&
          Accelleration.y>=-10 &&
             Accelleration.y<=10)
       {
          passed=1;
          break;
       }
    }

    if(passed)
       return 1;
    else
       return 0;
}
```

Sensor Boot Sequence

- Error handling
- Call initialization function
- Pull test data
- Validate data
- Return passed/failed

```
while(testCounter<15)
{
    ++testCounter;
    AccelInit();
    getAccel();
    if(Accelleration.x>=-10 &&
       Accelleration.x<=10 &&
          Accelleration.y>=-10 &&
             Accelleration.y<=10)
        {
            passed=1;
            break;
        }
    }

    if(passed)
        return 1;
    else
        return 0;
}
```

Sensor Boot Sequence

- Error handling
- Call initialization function
- Pull test data
- Validate data
- Return passed/failed

```
while(testCounter<15)
{
    ++testCounter;
    AccelInit();
    getAccel();
    if(Accelleration.x>=-10 &&
       Accelleration.x<=10 &&
          Accelleration.y>=-10 &&
             Accelleration.y<=10)
       {
          passed=1;
          break;
       }
    }

    if(passed)
       return 1;
    else
       return 0;
}
```

User Interface

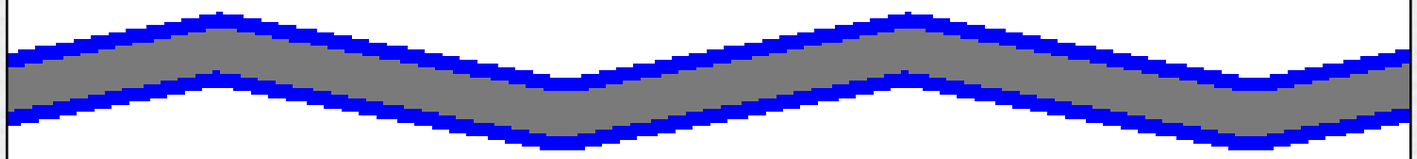
- Power
- GPS Sensor
- SD Card
- OBDII Data
- Camera

- On power up system LED flashes
- each sensor LED flashes as being tested
- System LED signaled between each sensor test
- After all sensors are done testing all LEDs sequence
- A solid sensor LED with a solid system LED during testing indicates an error
- If the system light begins to flash non stop during boot up it means there is an SD failure: either not present, full, or corrupted.
- After boot up and testing each sensor flashes as it is being polled

User Interface

- Software flashes sensor's LED upon activity

```
int getYawRate()
{
    int i;
    int DOUT[17];
    int decimalSum;

    DOUT[0]=0;

    WrPortI(PEB5R, NULL, 0xFF);
    BitWrPortI(PBDR, PBDRShadow,  0, 7);    //CS low
    BitWrPortI(PCDR, PCDRShadow,  1, 4);    //WRITE
```



```
    decimalSum += DOUT[5]*1024;
    decimalSum += DOUT[4]*2048;

    WrPortI(PEB5R, NULL, 0x00);

    return decimalSum;
}
/* END: YAW RATE GYRO FUNCTIONS */
```

User Interface

- Indicate sensor activity
- Mark end of boot sequence
- Notify user of any sensor initialization failures

```
printf("\nMain Function Starting");
printf("\nChecking Devices...");

OBDGTG = prepOBD();
yawGTG = prepYaw();
accelGTG = prepAccel();
GPSGTG = prepGPS();


flashLED1();
flashLED2();
flashLED3();
flashLED4();
flashLED5();


if(accelGTG==0)
   WrPortI(PEB3R, NULL, 0xFF);
if(OBDGTG==0)
   WrPortI(PEB4R, NULL, 0xFF);
if(yawGTG==0)
   WrPortI(PEB5R, NULL, 0xFF);
if(camGTG==0)
   WrPortI(PEB6R, NULL, 0xFF);
if(GPSGTG==0)
   WrPortI(PEB7R, NULL, 0xFF);
```

User Interface

- Indicate sensor activity
- Mark end of boot sequence
- Notify user of any sensor initialization failures

```
printf("\nMain Function Starting");
printf("\nChecking Devices...");

OBDGTG = prepOBD();
yawGTG = prepYaw();
accelGTG = prepAccel();
GPSGTG = prepGPS();


flashLED1();
flashLED2();
flashLED3();
flashLED4();
flashLED5();


if(accelGTG==0)
   WrPortI(PEB3R, NULL, 0xFF);
if(OBDGTG==0)
   WrPortI(PEB4R, NULL, 0xFF);
if(yawGTG==0)
   WrPortI(PEB5R, NULL, 0xFF);
if(camGTG==0)
   WrPortI(PEB6R, NULL, 0xFF);
if(GPSGTG==0)
   WrPortI(PEB7R, NULL, 0xFF);
```

User Interface

- Indicate sensor activity
- Mark end of boot sequence
- Notify user of any sensor initialization failures

```
printf("\nMain Function Starting");
printf("\nChecking Devices...");

OBDGTG = prepOBD();
yawGTG = prepYaw();
accelGTG = prepAccel();
GPSGTG = prepGPS();

flashLED1();
flashLED2();
flashLED3();
flashLED4();
flashLED5();

if(accelGTG==0)
   WrPortI(PEB3R, NULL, 0xFF);
if(OBDGTG==0)
   WrPortI(PEB4R, NULL, 0xFF);
if(yawGTG==0)
   WrPortI(PEB5R, NULL, 0xFF);
if(camGTG==0)
   WrPortI(PEB6R, NULL, 0xFF);
if(GPSGTG==0)
   WrPortI(PEB7R, NULL, 0xFF);
```

User Interface

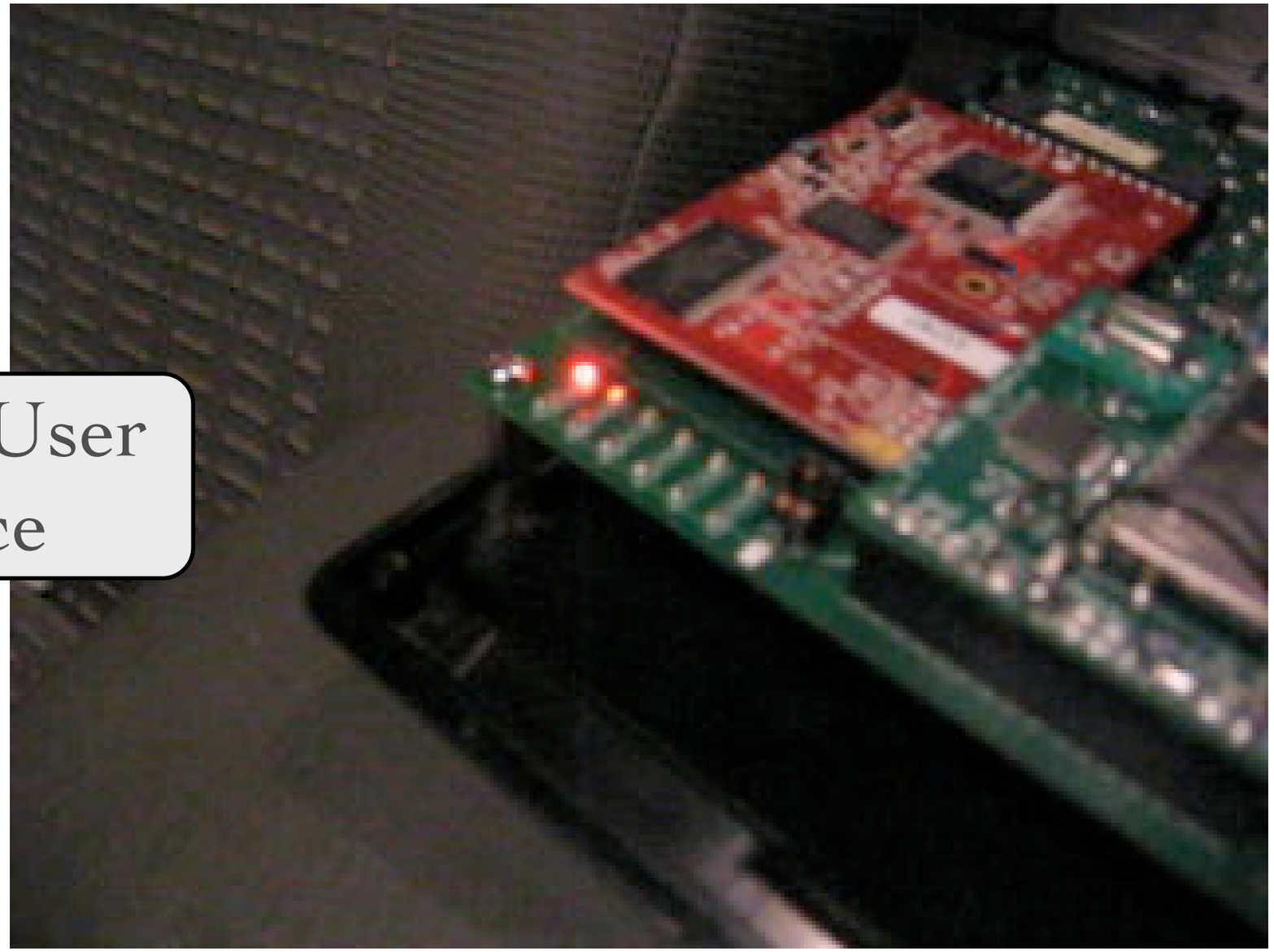
- Indicate sensor activity
- Mark end of boot sequence
- Notify user of any sensor initialization failures

```
printf("\nMain Function Starting");
printf("\nChecking Devices...");

OBDGTG = prepOBD();
yawGTG = prepYaw();
accelGTG = prepAccel();
GPSGTG = prepGPS();


flashLED1();
flashLED2();
flashLED3();
flashLED4();
flashLED5();


if(accelGTG==0)
   WrPortI(PEB3R, NULL, 0xFF);
if(OBDGTG==0)
   WrPortI(PEB4R, NULL, 0xFF);
if(yawGTG==0)
   WrPortI(PEB5R, NULL, 0xFF);
if(camGTG==0)
   WrPortI(PEB6R, NULL, 0xFF);
if(GPSGTG==0)
   WrPortI(PEB7R, NULL, 0xFF);
```

Intelligent Power Control

(Sleepy Mode)

Sleepy Mode Demo



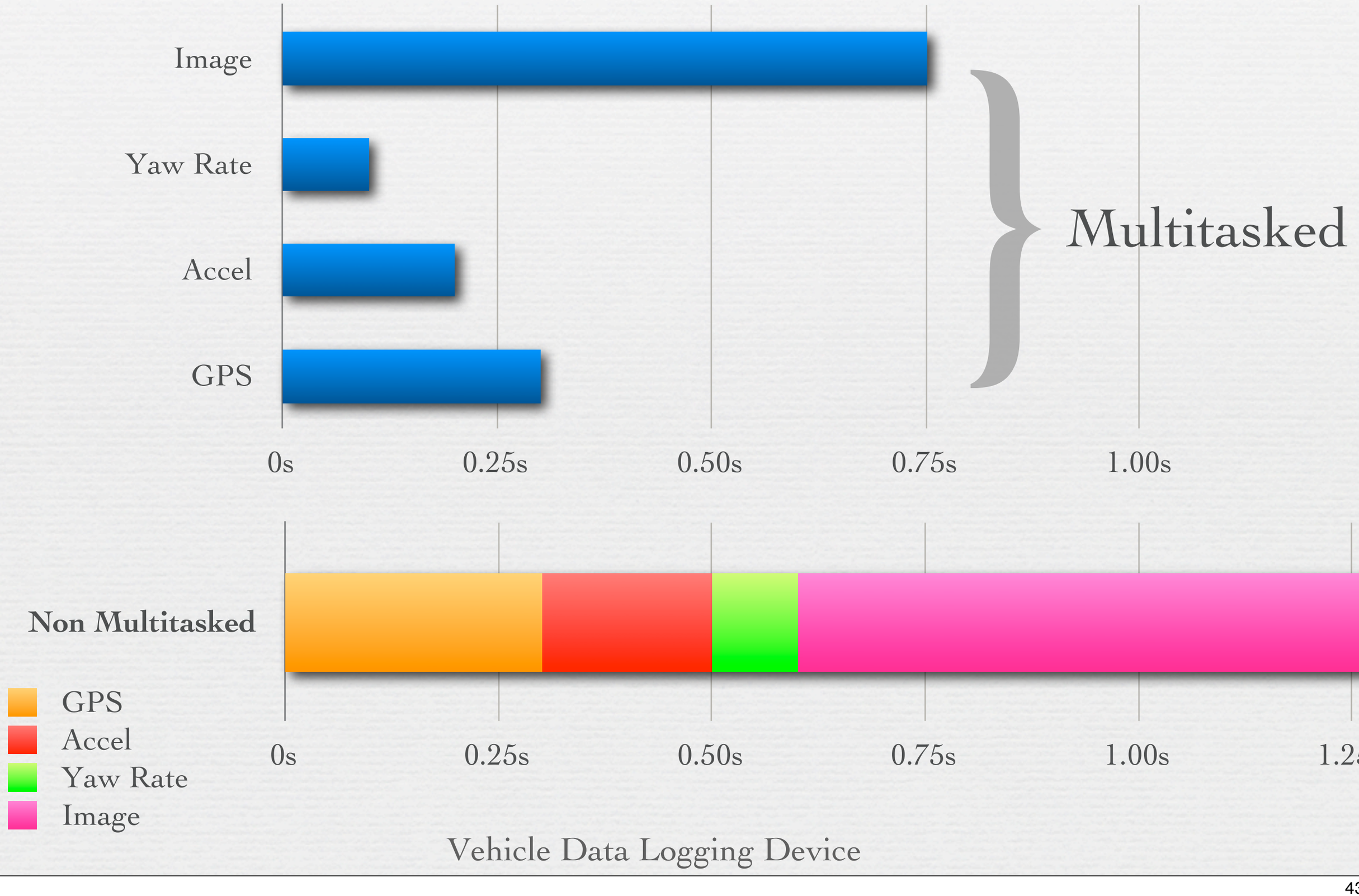
Also note User Interface

- Device powers down when RPM = 0
- Accelerations wake the unit up to check the RPM status as the accelerometer can be polled using less power

Sensor Polling

The need for multitasking.

Multitasking Timing Illustration



Polling Sensors

Multitasking

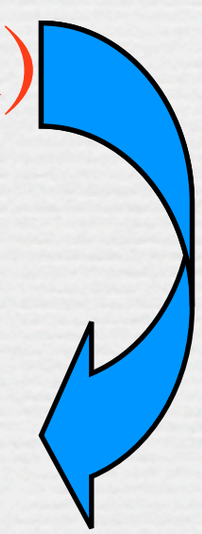
```
while(1){
  costate { … }
  costate { … }
  costate { … }
}
```

- costate descriptor
- Resides within infinite loop
- costate flow proceeds in three ways

Polling Sensors

Multitasking

```
costate{
   statement 1
   statement 2
   waitfor(condition)
   statement 3
   statement 4
}
```

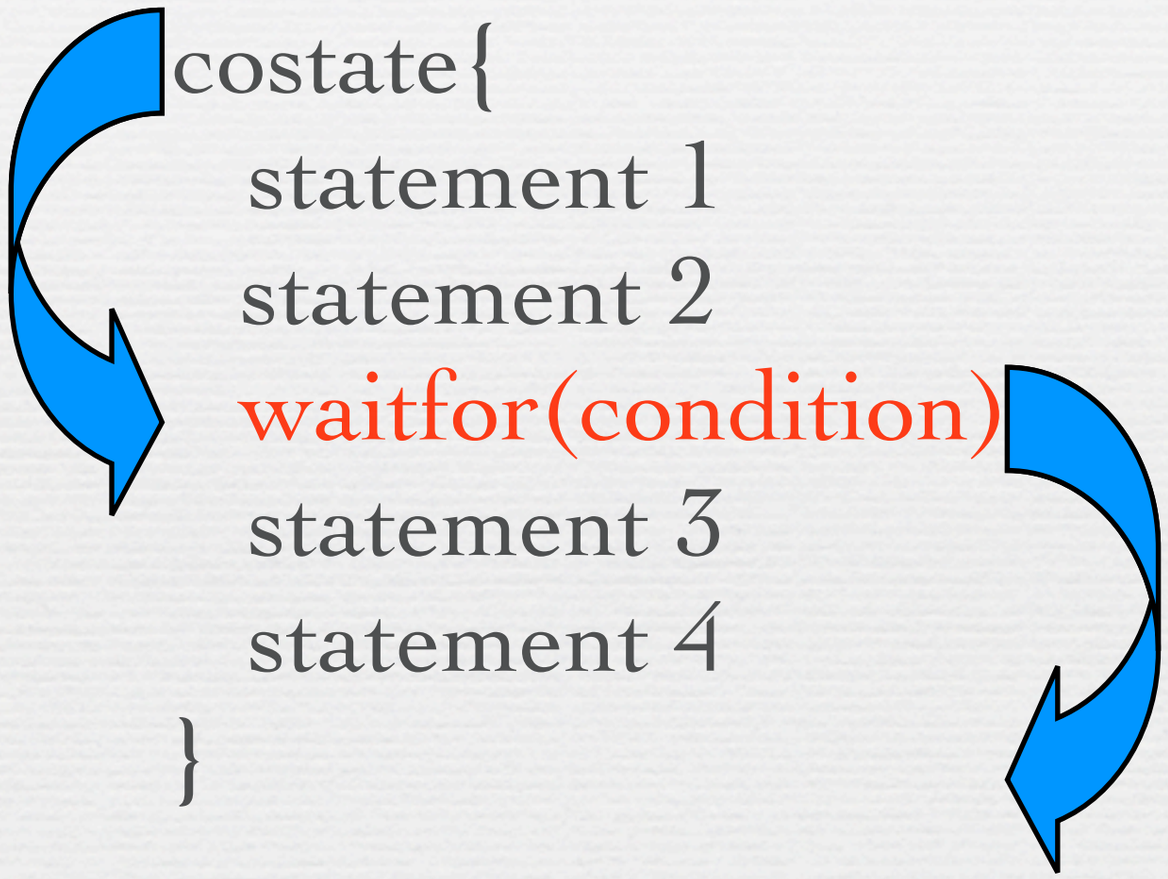


Initial entry into costate

- **Statement 1 & 2 executed**
- **waitfor evaluated to zero (0)**
- **Flow jumps out of the costate**

Polling Sensors

Multitasking



Subsequent entry into costate

- **Flow jumps directly to waitfor**
- **waitfor evaluated to zero (0)**
- **Flow jumps out of the costate**

Polling Sensors

Multitasking

```
costate{
    statement 1
    statement 2
    waitfor(condition)
    statement 3
    statement 4
}
```

Subsequent entry into costate

- **Flow jumps directly to waitfor**
- **waitfor evaluated to one (1)**
- **Statement 3 & 4 executed**
- **System reference to costate reset**

Polling Sensors

Multitasking

```
costate{
   statement 1
   statement 2
   waitfor(condition)
   statement 3
   statement 4
}
```

<u>After costate reference is reset</u>

- **Next cycle will be executed as an initial entry**
- **Cycle starts anew**
- **Statement 1 & 2 executed**
- **waitfor evaluated to zero (0)**

Polling Sensors

Multitasking

Allows Rabbit to devote time to other tasks while…

- Waiting for ELM327 Ready to Receive Flag
- Waiting for ELM327 to fill buffer with data

```
cofunc int getThrot()
{
    char readSentence[25];
    char OBDresponse[5];
    int carThrot;
    int x;

    carThrot=1;

    serErdFlush();
    serEwrFlush();
   waitfor(0==BitRdPortI(PGDR, 4));
    serEputs("01 11\r");
   waitfor(20==serErdUsed());
    serEread(readSentence, 20, 10);
```

Polling Sensors

Multitasking

Allows Rabbit to devote time to other tasks while…

- Waiting for ELM327 Ready to Receive Flag
- Waiting for ELM327 to fill buffer with data

```
cofunc int getThrot()
{
    char readSentence[25];
    char OBDresponse[5];
    int carThrot;
    int x;

    carThrot=1;

    serErdFlush();
    serEwrFlush();
    waitfor(0==BitRdPortI(PGDR, 4));
    serEputs("01 11\r");
    waitfor(20==serErdUsed());
    serEread(readSentence, 20, 10);
```

Polling Sensors

Multitasking

Allows Rabbit to devote time to other tasks while…

- Waiting for ELM327 Ready to Receive Flag
- Waiting for ELM327 to fill buffer with data

```
cofunc int getThrot()
{
    char readSentence[25];
    char OBDresponse[5];
    int carThrot;
    int x;

    carThrot=1;

    serErdFlush();
    serEwrFlush();
   waitfor(0==BitRdPortI(PGDR, 4));
    serEputs("01 11\r");
   waitfor(20==serErdUsed());
    serEread(readSentence, 20, 10);
```

Storing Data to SD Card

- New text file is created on system boot
  - **system boots each time ignition is cycled**
- Each cycle's data is appended to this text file

```
UTC Time              $$ UTC Position(Lat,Lon)          $$ Yaw $$ Accel(x,y)
$$ OBD(RPM,Speed,Throttle)
-------------------------------------------------------------------------------
04-19-2007 07:32:36 $$ 28 25.463701 N 081 15.402101 W $$ +01 $$  0.63 -0.55
$$ 0812 000 000
04-19-2007 07:32:36 $$ 28 25.463701 N 081 15.402101 W $$ +01 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:36 $$ 28 25.463701 N 081 15.402101 W $$ +01 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:36 $$ 28 25.463701 N 081 15.402101 W $$ +01 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:36 $$ 28 25.463701 N 081 15.402101 W $$ +00 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:36 $$ 28 25.463701 N 081 15.402101 W $$ +00 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:36 $$ 28 25.463701 N 081 15.402101 W $$ +01 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:48 $$ 28 25.463301 N 081 15.399800 W $$ +01 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:48 $$ 28 25.463301 N 081 15.399800 W $$ +01 $$  0.02  0.03
```

```
*00-00-0000
00:00:02*FFD8FFE000114A4649460001020304050607080900
785C666864FFDB00430112121216161630 1A1A306442384264
0000000000000102030405060708090A0BFFC400B510000201
36373839 3A434445464748494A535455565758595A63646566
E1E2E3E4E5E6E7E8E9EAF1F2F3F4F5F6F7F8F9FAFFC4001F01
1
3223281081442 91A1B1C109233352F015627 2D10A162434E12
999AA2A3A4A5A6A7A8A9AAB2B3B4B5B687B8B9BAC2C3C4C5C6
48E69314F229B8A0618A314B45218014B8A4CD2E6900A2A64A
A598E147535CFCD235C4CD230EBD07A5696A92958D621FC7C9
1E7DDA605FEEB1AD5AE7FC2AE5ADEE14FF000B0FEB5D054D89
8
A43466B31851499A4CD201D4F5A8F34A1A901614D4AA6AAAB5
CD2B0E1C803F0CD7454D753196E1451453 2428A28A6073C501
E14D4ADE5D9A7DE006440444792 5C7248E3A600FC7F0AECC59
71918E4567D9DB9B9620B615464D6C12ACEA57A13FD2B29EC6
39AA2B4828A291890A48EB564ADC5A2A389CBAE4AE2A4A10E4
4
BD0B5B8756B70CAA125CE1E3DD92A79F6F403FC8AF2ECD5CD3
9B3819A18F4BD56E2DE272D103C647623207E19C55A8984891
62AD8499DC9F3286033903A9C0E2B3847B9A4E5D8F53D37EEE
304D349A09A69AA2C334B494B400668CD145075FE03BF8E38
```

Image Capture

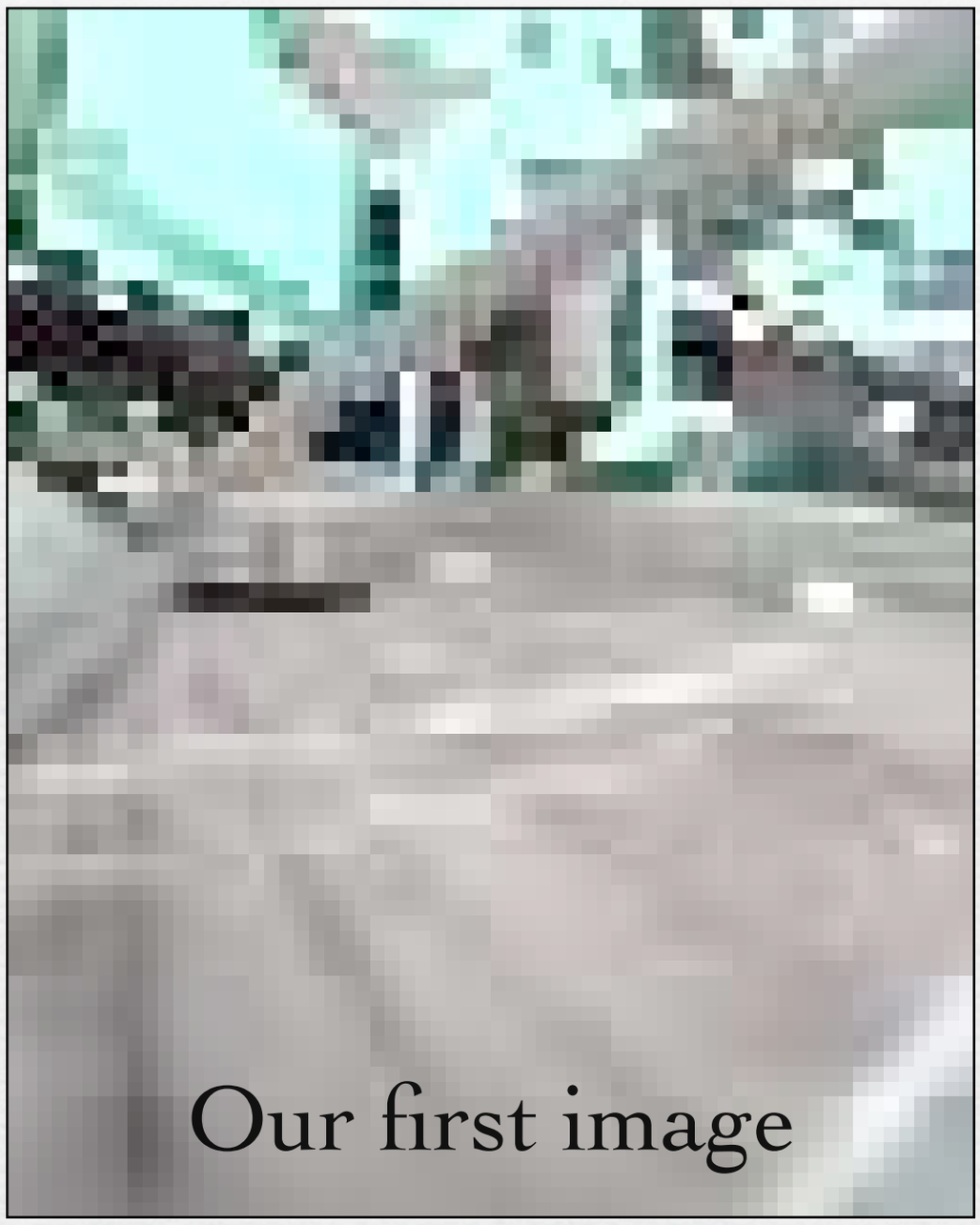
- Stored to a separate file

  - **images stored in one text file per trip**

  - **images sorted by time within text file**

- Post Processing

  - **ASCII data is sorted and converted to JPEGs via web application**

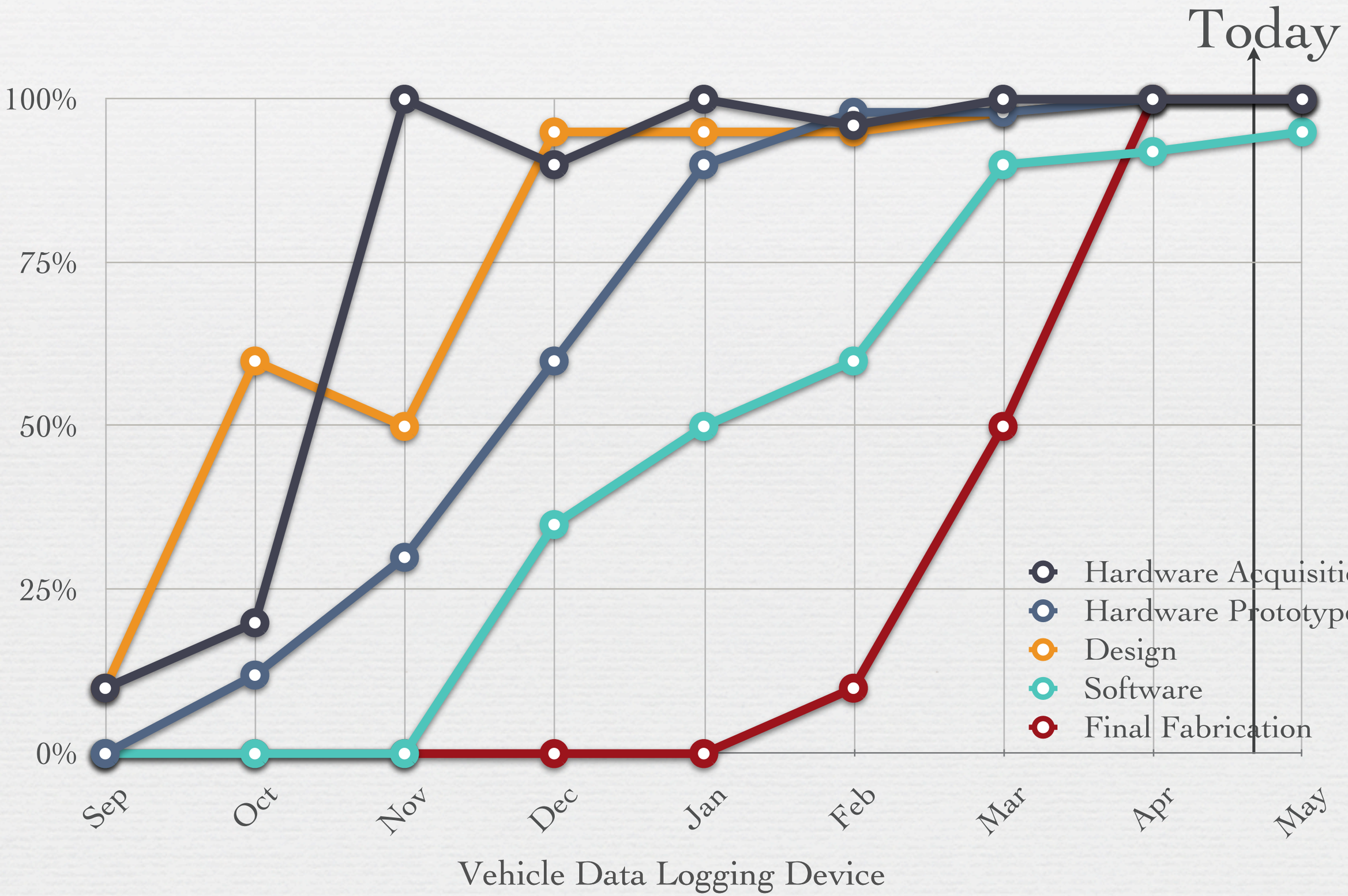


Software to Do

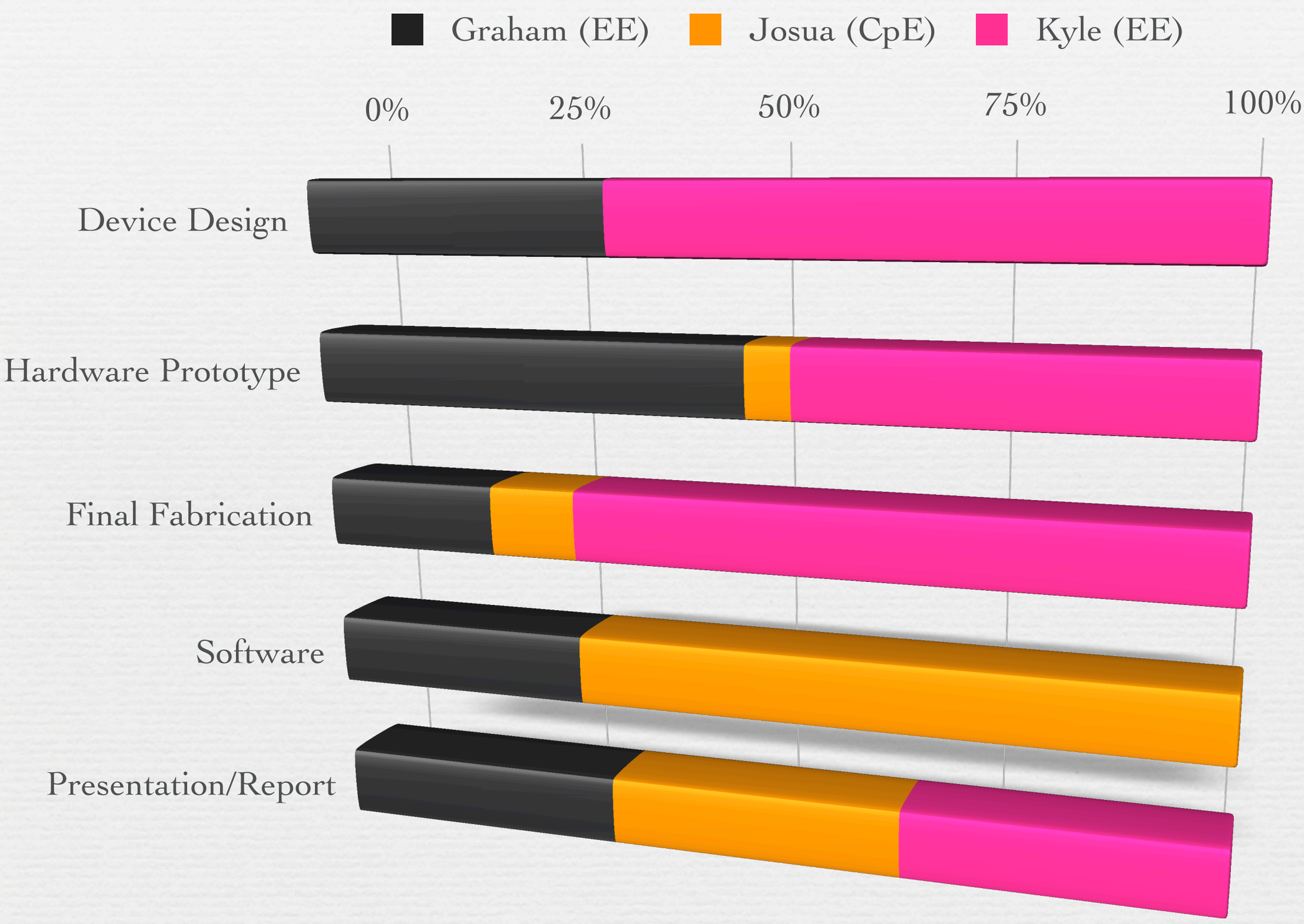
- Improve Image Capture
- Implement System Configurability
  - **Allow user to edit text file on SD to change system settings**
- Fully Incorporate Intelligent Power Supply

Progress & Budget

Project Progress

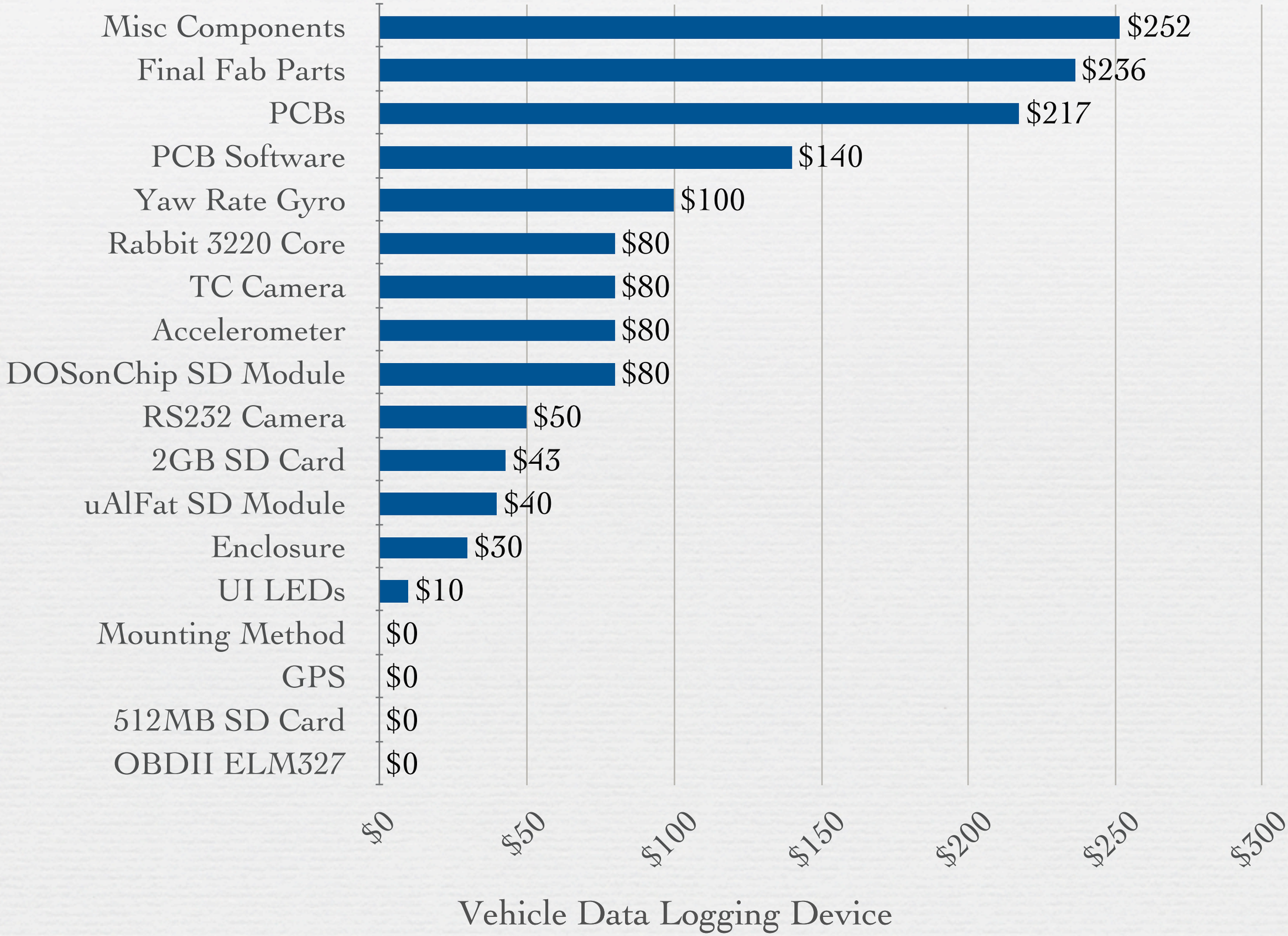


Work Breakdown



Vehicle Data Logging Device

Budget


Misc Components
$252
Final Fab Parts
$236
PCBs
$217
PCB Software
$140
Yaw Rate Gyro
$100
Rabbit 3220 Core
$80
TC Camera
$80
Accelerometer
$80
DOSonChip SD Module
$80
RS232 Camera
$50
2GB SD Card
$43
uAlFat SD Module
$40
Enclosure
$30
UI LEDs
$10
Mounting Method
$0
GPS
$0
512MB SD Card
$0
OBDII ELM327
$0
$0
$50
$100
$150
$200
$250
$300
Vehicle Data Logging Device

Miscellaneous Costs

Capacitors, power regulators, wire, perf board, tools, solder, SOIC adapters, DIP adapters, diodes, transistors, DB9 connectors, markers, PCB etching materials, pre-made breakout boards, RS232 level shifters, breadboards, test leads, and more...

Budget Specifics

Part	Actual Cost	Our Cost	# Acquired	Total Spent
GPS Device	$80.00	$0.00	2	$0.00
CMOS Image Sensor	$40.00	$40.00	2	$80.00
RS232 Camera	$50.00	$50.00	1	$50.00
2-Axis Accelerometer	$40.00	$40.00	2	$80.00
Yaw Rate Gyro	$50.00	$50.00	2	$100.00
ELM327	$35.00	$0.00	2	$0.00
DOSonChip	$40.00	$40.00	2	$80.00
uALfat	$40.00	$40.00	2	$80.00
UI LEDs (4)	$10.00	$10.00	0	$0.00
2GB SD Card	$43.00	$43.00	1	$43.00
512MB SD Card	$20.00	$0.00	1	$0.00
Rabbit 3220 Core	$80.00	$80.00	1	$80.00
Misc Components	??	$251.60	1	$251.60
Enclosure	$30.00	$30.00	0	$30.00
Mounting Method	$5.00	$5.00	0	$0.00
PCB Software	$140.00	$140.00	1	$140.00
Final Fab Parts (3)	$236.17	$236.17	1	$236.17
PCBs (5)	$217.35	$217.35	1	$217.35
Parts Totals	$455.00	$819.60	22	$1,468.12

Device Reproduction Costs

Required Parts	Actual Cost
Rabbit 3220 Core	$80.00
Accelerometer	$15.00
ELM327	$35.00
GPS Device	$80.00
512MB SD Card	$20.00
uALfat	$40.00
Misc Components	$30.00
Required Parts Total	$300.00
Camera	$40.00
Yaw Rate Gyro	$50.00
Parts Totals	$390.00

Without optional components, the total cost of fabrication is easily within the budget constraints.

With optional components we must have a large production run in order to meet the specified system cost.

We suggest in this instance the Rabbit be swapped for a PIC thereby saving $80, about the cost of the additional components.

Number of Boards	2	4	6	8	10	15
PCB Fabrication	$130.00	$130.00	$135.00	$140.00	$145.00	$155.03
Cost/Board	$65.00	$32.50	$22.50	$17.50	$14.50	$10.34
Totals	$455.00	$422.50	$412.50	$407.50	$404.50	$400.34
w/o opt components	$365.00	$332.50	$322.50	$317.50	$314.50	$310.34

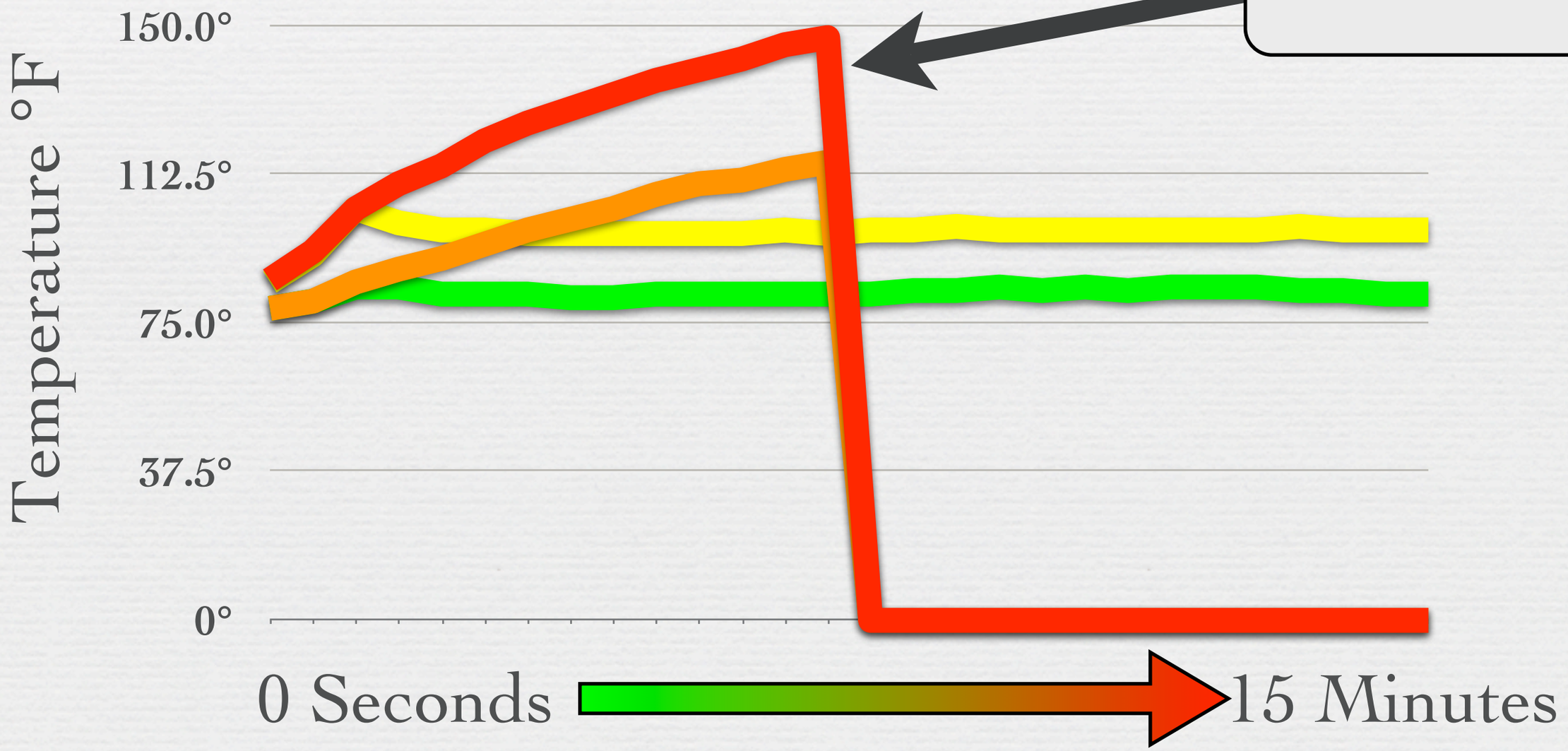
Difficulties

- LGA 16 package (yaw rate gyro) was difficult to work with
- SPI driver had to be rewritten
- 24 Pin ZIF (first camera) also difficult to work with
- Hardware discontinued mid-development (first camera)
- Reworking storage device due to insufficient write speed (DOSonCHIP)
- Thermal dissipation issues with varying voltages

Temperature Analysis

- On Heatsink No Enclosure
- Above Heatsink No Enclosure
- On Heatsink With Enclosure
- Above Heatsink With Enclosure

PSU Thermal



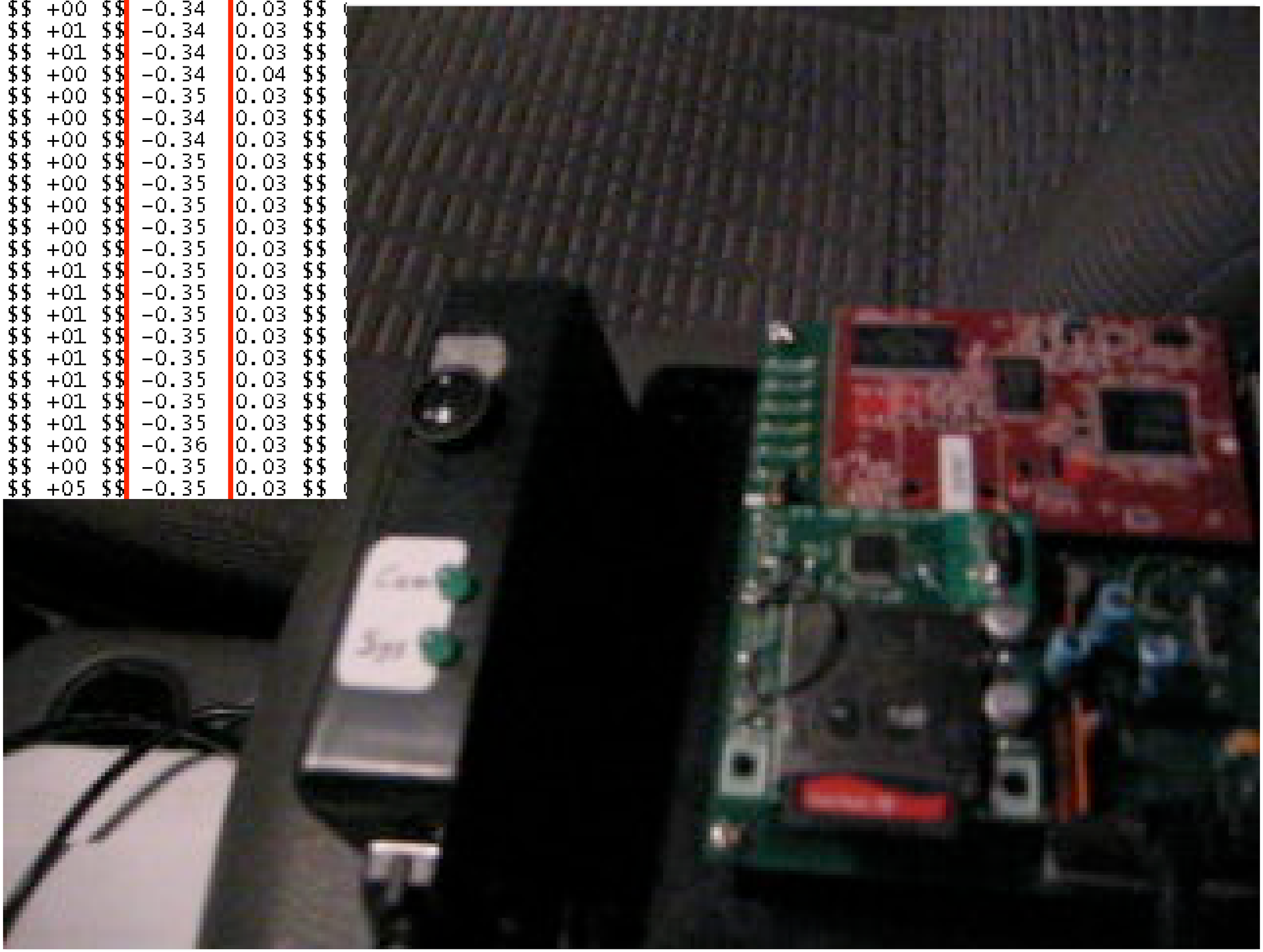
Future Progress

- Optimize Software
- Improve Image Capture
- Implement System Configurability
- Further Calibrate Sensors
- Optimize PCB for better cooling
  - ★ Reduce Size of Device

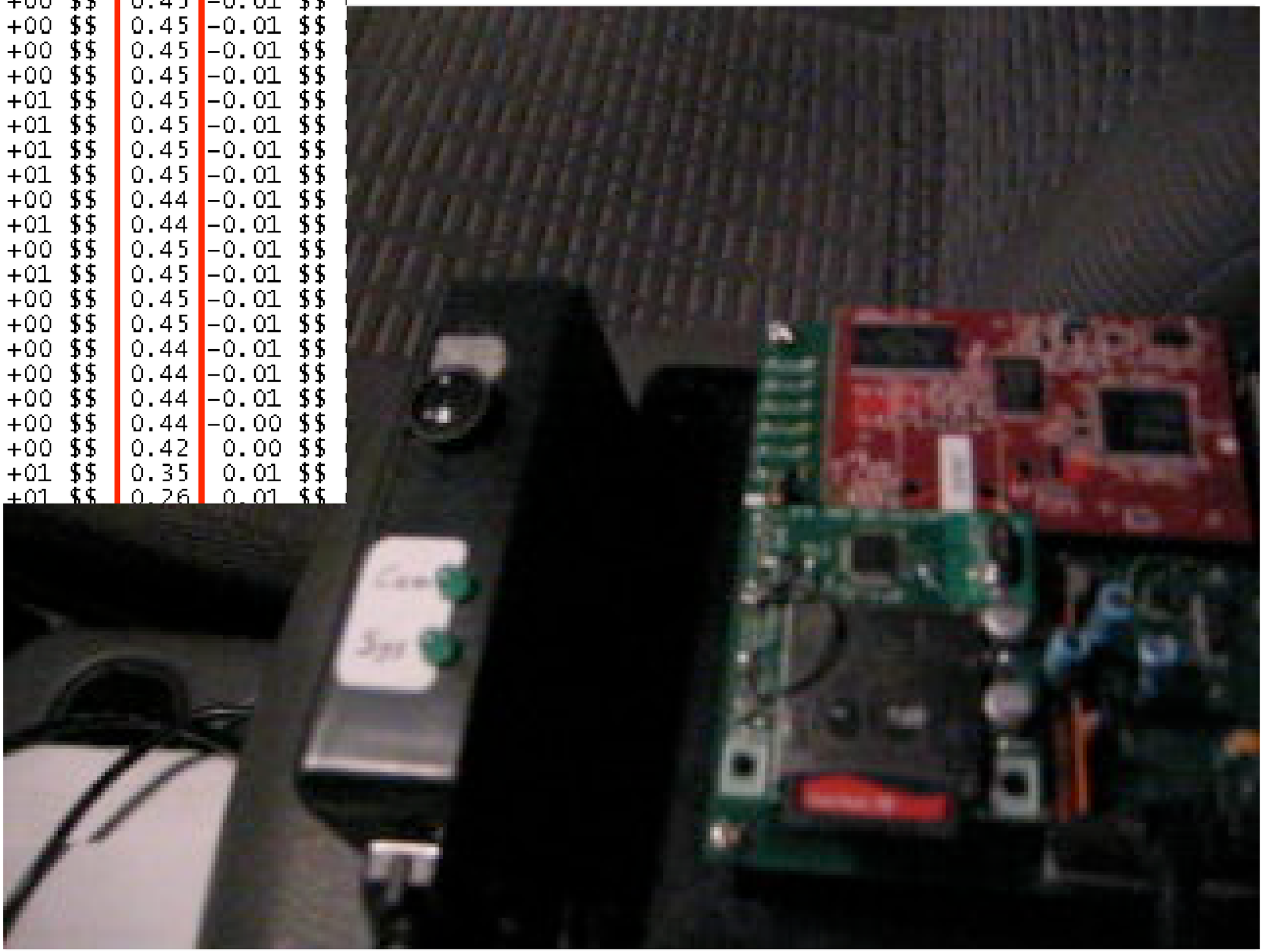
Demo!

Demonstration

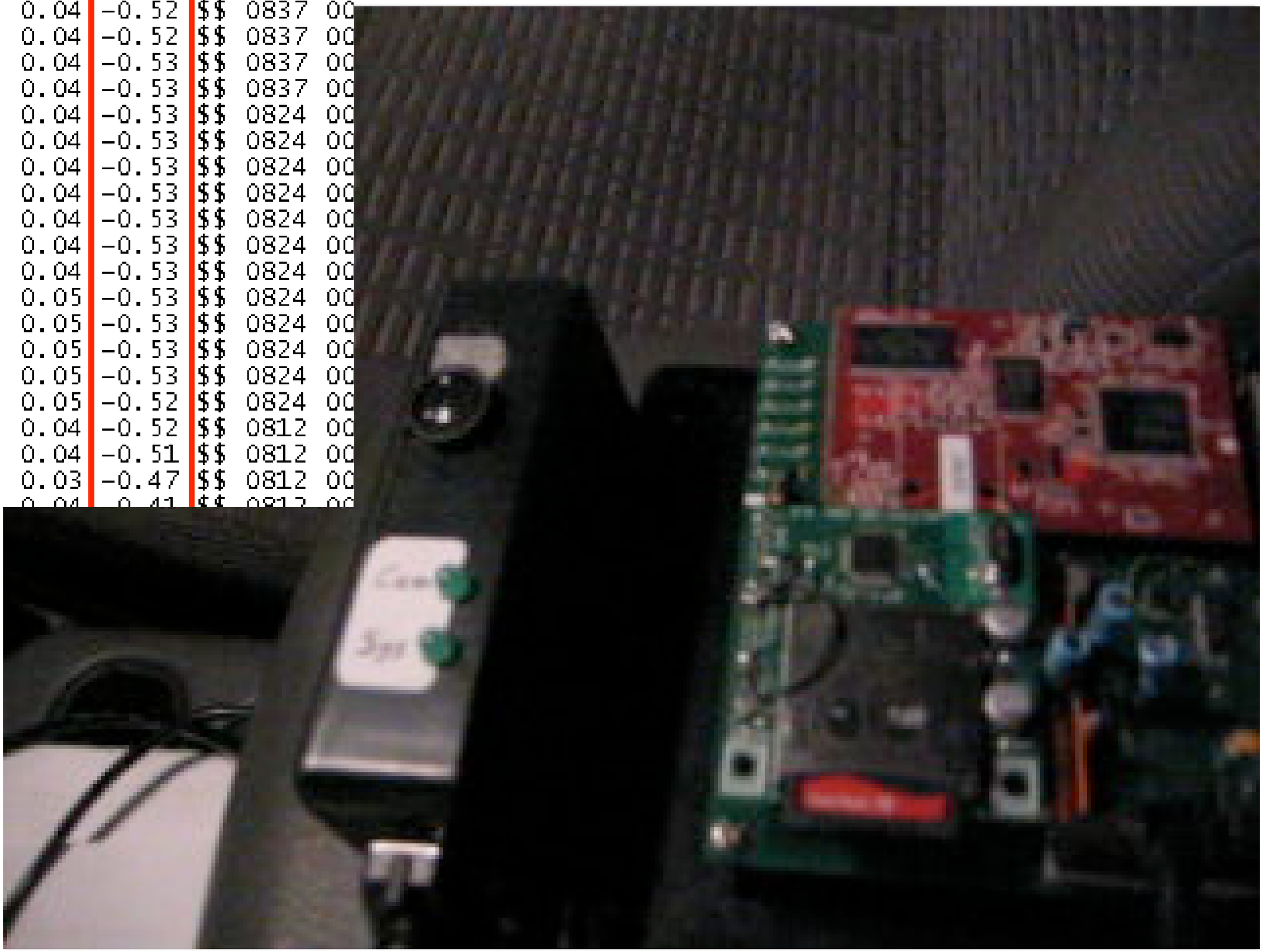
Demonstration



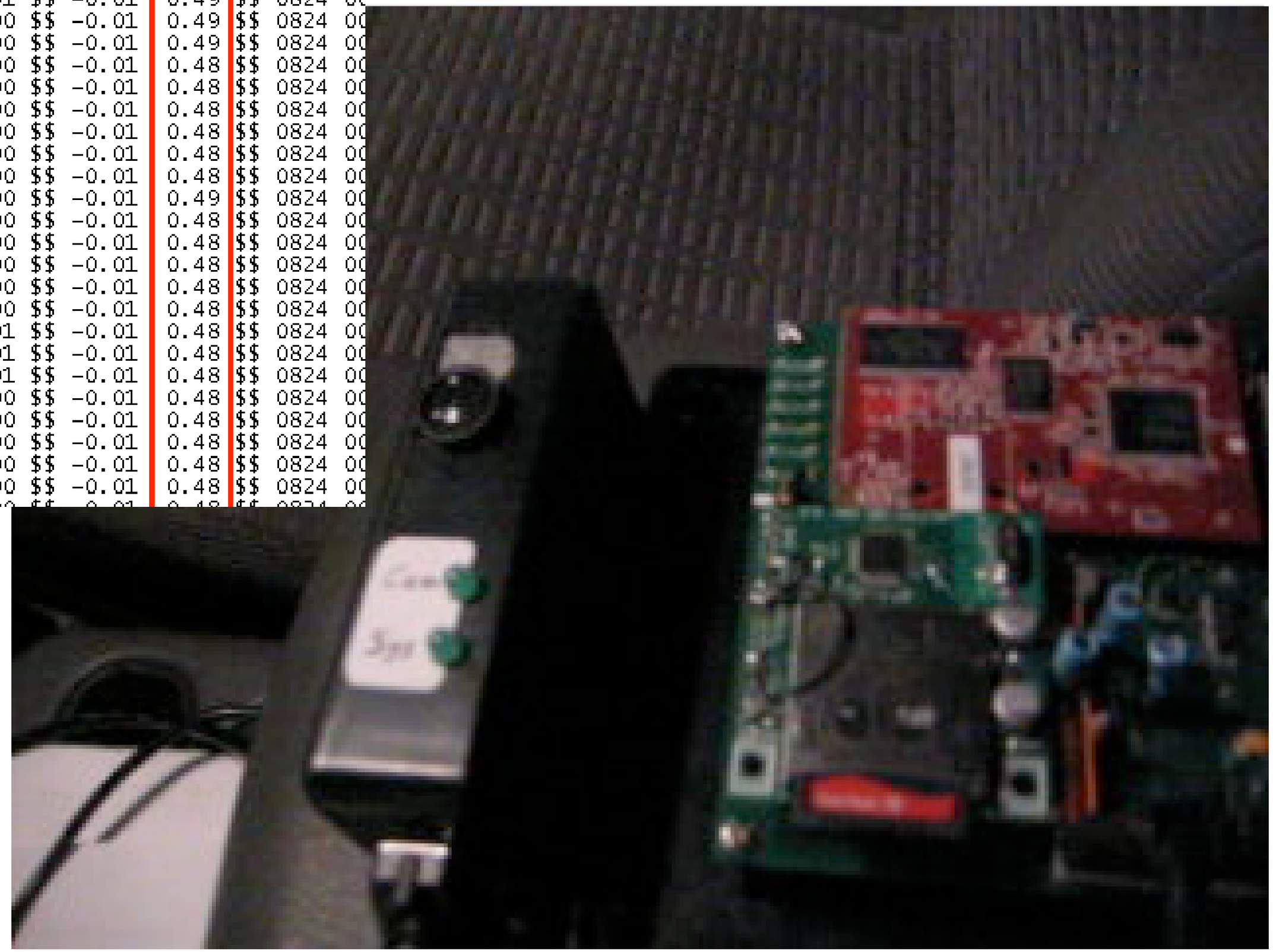
Demonstration



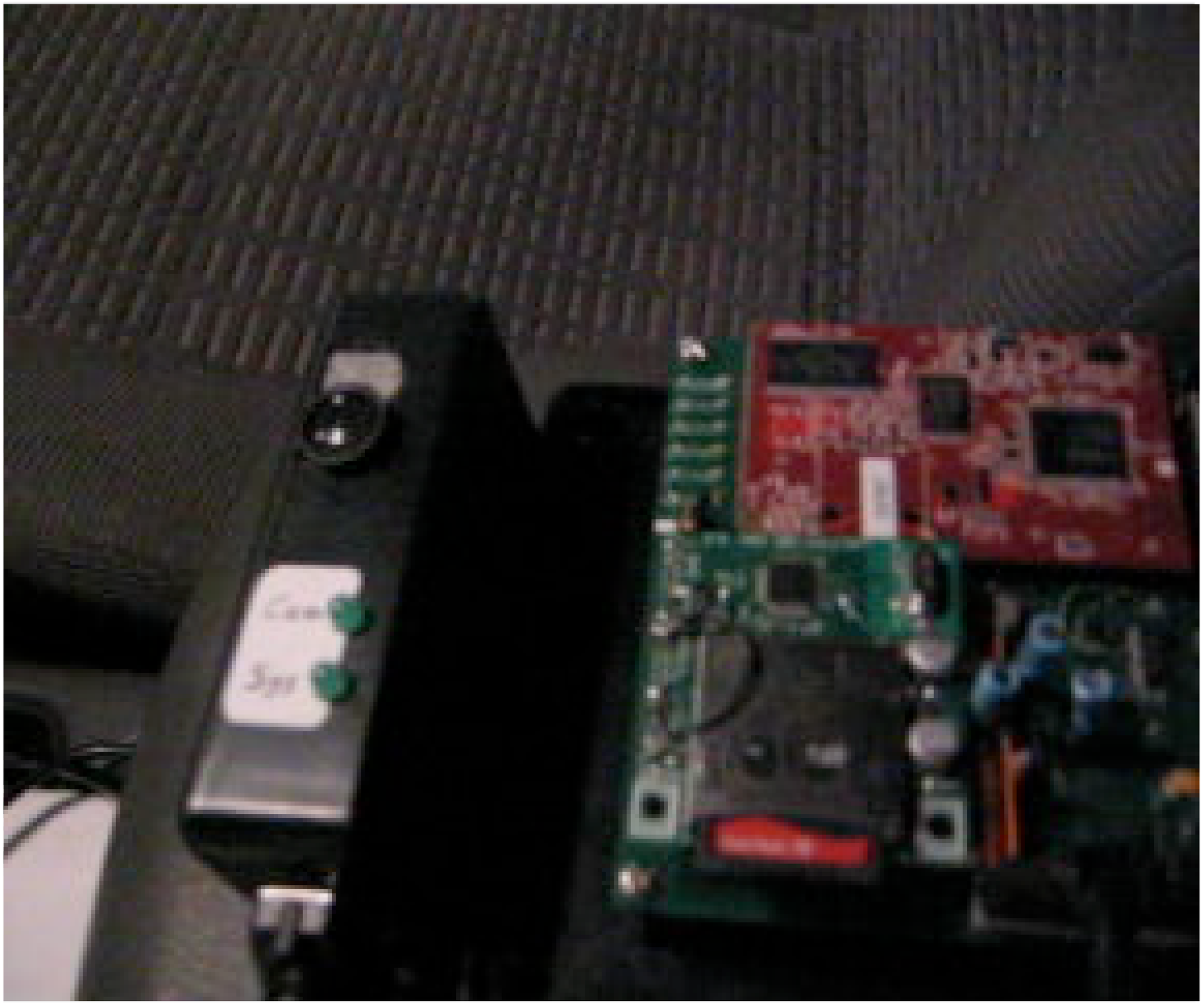
Demonstration



Demonstration

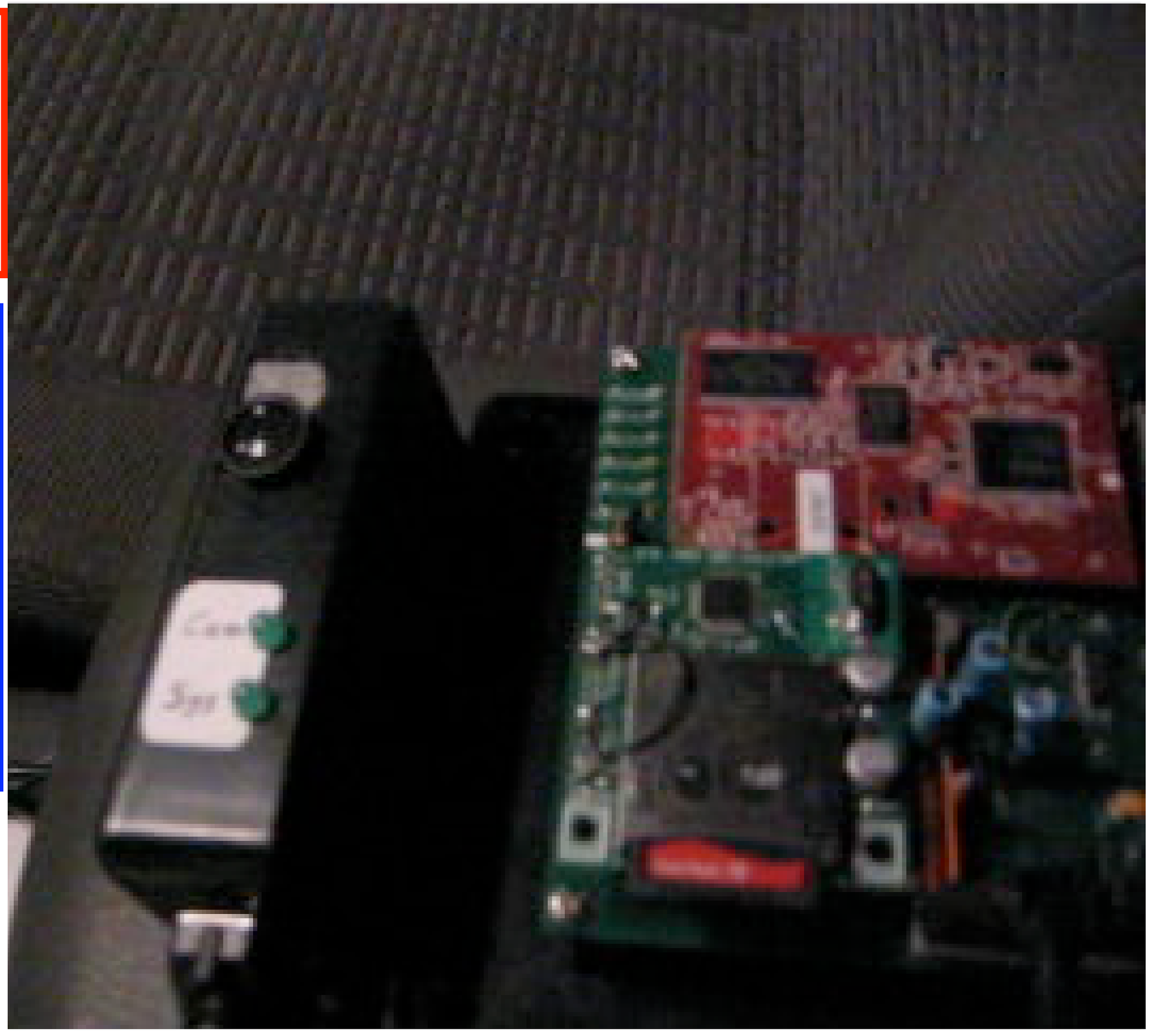


Demonstration



Demonstration

```
$ 0812 000 009
$ 0812 000 009
$ 0812 000 009
$ 0812 000 009
$ 0812 000 009
$ 0812 000 009
$ 0812 000 009
$ 3737 000 009
$ 3737 000 005
$ 3737 000 005
$ 3737 000 005
$ 2949 000 005
$ 2949 000 005
$ 2949 000 006
$ 2949 000 006
$ 2949 000 006
$ 2949 000 006
$ 2749 000 006
$ 2749 000 006
$ 2749 000 006
$ 2749 000 006
$ 2749 000 000
$ 2749 000 000
$ 2749 000 000
$ 2749 000 000
$ 2749 000 000
$ 2749 000 000
$ 2749 000 000
$ 1462 000 000
$ 1462 000 000
```



Demonstration

Data Output

Yaw Rate $$ X Y $$ RPM $$ Speed $$ Throttle Pos.

```
$$ +30 $$ -0.12 -0.35 $$ 1162 012 000
$$ +31 $$ -0.12 -0.36 $$ 1162 012 000
$$ +31 $$ -0.12 -0.37 $$ 1162 012 000
$$ +31 $$ -0.13 -0.38 $$ 1162 012 000
$$ +32 $$ -0.13 -0.38 $$ 1149 012 000
$$ +32 $$ -0.13 -0.39 $$ 1149 012 000
$$ +33 $$ -0.13 -0.39 $$ 1149 012 000
$$ +32 $$ -0.14 -0.41 $$ 1149 012 000
$$ +33 $$ -0.14 -0.40
$$ +33 $$ -0.14 -0.40 $$ 1149 012 0
$$ +34 $$ -0.14 -0.39 $$ 1062 012 0
$$ +35 $$ -0.14 -0.39 $$ 1062 012 000
$$ +35 $$ -0.14 -0.38 $$ 1062 012 000
$$ +36 $$ -0.14 -0.37 $$ 1062 011 000
$$ +36 $$ -0.14 -0.37 $$ 1062 011 000
$$ +36 $$ -0.15 -0.37 $$ 1062 011 000
$$ +36 $$ -0.15 -0.37 $$ 1062 011 000
$$ +36 $$ -0.15 -0.36 $$ 0999 011 000
```

Note harder acceleration

Tightening Turn

Data Log File

DL190732.TXT

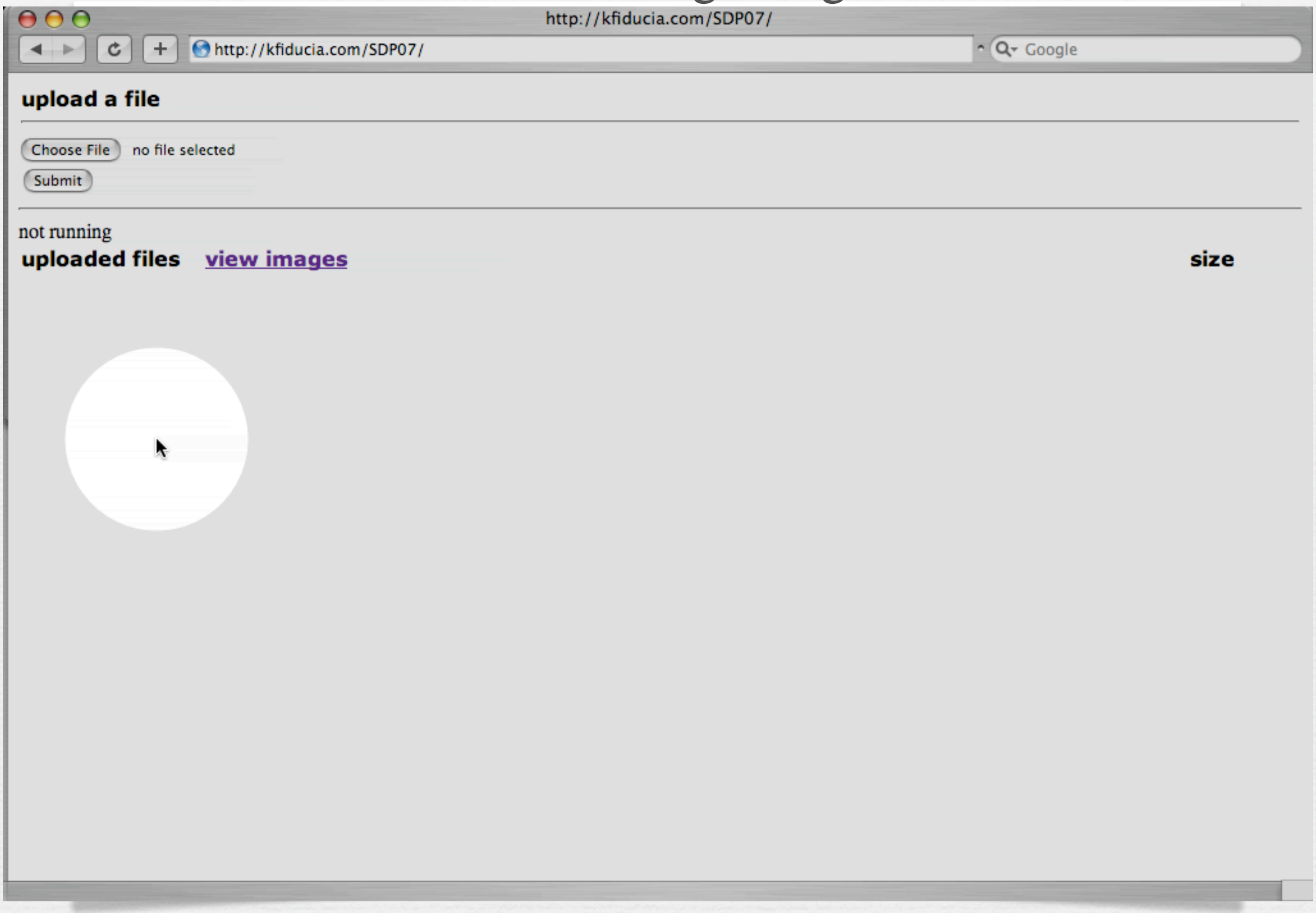
```
UTC Time            $$ UTC Position(Lat,Lon)          $$ Yaw $$ Accel(x,y)
$$ OBD(RPM,Speed,Throttle)
-------------------------------------------------------------------------------------------------
04-19-2007 07:32:36 $$ 28 25.463701 N 081 15.402101 W $$ +01 $$  0.63 -0.55
$$ 0812 000 000
04-19-2007 07:32:36 $$ 28 25.463701 N 081 15.402101 W $$ +01 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:36 $$ 28 25.463701 N 081 15.402101 W $$ +01 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:36 $$ 28 25.463701 N 081 15.402101 W $$ +01 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:36 $$ 28 25.463701 N 081 15.402101 W $$ +00 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:36 $$ 28 25.463701 N 081 15.402101 W $$ +00 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:36 $$ 28 25.463701 N 081 15.402101 W $$ +01 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:48 $$ 28 25.463301 N 081 15.399800 W $$ +01 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:48 $$ 28 25.463301 N 081 15.399800 W $$ +01 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:48 $$ 28 25.463301 N 081 15.399800 W $$ +00 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:48 $$ 28 25.463301 N 081 15.399800 W $$ +00 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:48 $$ 28 25.463301 N 081 15.399800 W $$ +00 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:48 $$ 28 25.463301 N 081 15.399800 W $$ +01 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:48 $$ 28 25.463301 N 081 15.399800 W $$ +01 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:48 $$ 28 25.463301 N 081 15.399800 W $$ +01 $$  0.02  0.03
$$ 0812 000 000
04-19-2007 07:32:49 $$ 28 25.463499 N 081 15.399600 W $$ +01 $$  0.02  0.03
$$ 0774 000 000
04-19-2007 07:32:49 $$ 28 25.463499 N 081 15.399600 W $$ +01 $$  0.03  0.03
$$ 0774 000 000
04-19-2007 07:32:49 $$ 28 25.463499 N 081 15.399600 W $$ +01 $$  0.03  0.03
$$ 0774 000 000
04-19-2007 07:32:49 $$ 28 25.463499 N 081 15.399600 W $$ +00 $$  0.03  0.03
$$ 0774 000 000
04-19-2007 07:32:50 $$ 28 25.463701 N 081 15.399400 W $$ +00 $$  0.02  0.03
$$ 0774 000 000
04-19-2007 07:32:50 $$ 28 25.463701 N 081 15.399400 W $$ +01 $$  0.03  0.03
$$ 0824 000 000
```

Picture Log File

PLOG1377D.TXT

```
*00-00-0000
00:00:02*FFD8FFE000114A46494600010203040506070809 0AFFDB004300100C0C0E0C0A100E0E0E1212101418281A181616183224261E283A343E3C3A34383840485C4E4044584638385
785C666864FFDB0043011212121616161301A1A3064423842646464646464646464646464646464646464646464646464646464646464646464646464646464646464646464FF
0000000000000102030405060708090A0BFFC400B5100002010303020403050504040000017D01020300041105122131410613516107227114328191A1082342B1C11552D1F02433627282
363738393A434445464748494A5354555657585956363465666768696A737475767778797A838485868788898A929394959697989999AA2A3A4A5A6A7A8A9AAB2B3B4B5B6B7B8B9BAC2C3C4
E1E2E3E4E5E6E7E8E9EAF1F2F3F4F5F6F7F8F9FAFFC4001F0100030101010101010101010000000000000102030405060708090A0BFFC400B5110002010204040304070504040001027700
1
3223281081442 91A1B1C109233352F0156272D10A162434E125F11718191A262728292A35363738393A434445464748494A5354555657585956363465666768696A737475767778797A828
999AA2A3A4A5A6A7A8A9AAB2B3B4B5B6B7B8B9BAC2C3C4C5C6C7C8C9CAD2D3D4D5D6D7D8D9DAE2E3E4E5E6E7E8E9EAF2F3F4F5F6F7F8F9FAFFC000110800800 0A00301220002110103110 1
48E69314F229B8A0618A314B4521801 4B8A4CD2E6900A2A64A8454B19A00B71D5A4AA919AB28D4C4595A78A895A9E1A988EDEC2432D842E4E72B566A868ADBB4987DB23F535788C906B3E9
A598E147535CFCD235C4CD230EBD07A5696A92958D621FC7C9FC2B3074AB8A3482D2E702C39C5308A99D706A1231435A1284A28A291414514520145 886A2A7A9A00B686AC235534353AB5
1E7DDA605FEEB1AD5AE7FC2AE5ADEE14FF000B0FEB5D054D8996E14514532428A28A00C6BF666BD704F0A001F95448BB980F5A6CB26F95DCF724D4962E24B8C62B45A2374B43829C056C55
8
A43466B31851499A4CD201D4F5A8F34A1A901614D4AA6AAAB55B88855421B2C7A8C74A5276571C55D8F8999D73B4834F9321390696495A389994608154E3BC794E0F4A9526CA7048EE3C29
CD2B0E1C803F0CD7454D753196E14514532428A28A6073C501157F4DB5441E6753DAAACA9B6465F438AD5B45D86EB54DE86F27A5CF2D9DF26AAB1A748D9351D5DF420334D66DAA4D2D2119
E14D4ADE5D9A7DE0064404447925C7248E3A600FC7F0AECC59DB8E918FCCD1A7521C9DEC78D79A7FBBFAD39652481B71F8D7B27D961FEE7EA6B375CD2FEDBA54D6D0111B48319EB9208207
71918E4567D9DB9B9620B615464D6C12ACEA57A13FD2B29EC6D4F712440C8CA46723A7AD531008CE44257FE060D6832E73DC567F9655BFE3D367BF9B9ACE37359A3D174180DB69C909FBC0
39AA2B4828A291890A48EB564ADC5A2A389CBAE4AE2A4A10E4ACEC655DA8172E07D6B4A218894553BF401D5C753D6AEC7FEAC553D8A7F0A3C758F34DCD0D4C355701F9A426999A33401341
4
BD0B5B8756B70CAA125CE1E3DD92A79F6F403FC8AF2ECD5CD33529B4BBB13C5923A3A67018504C95CF5D0C1864547708D242C8B8C9EF5CE69BE20D4F5287CFB5D1C4880ED63F6A51CFD08A
9B3819A18F4BD56E2DE272D103C647623207E19C55A898489138006735 F88E39EE6FC3FD80C12AFDFFDE2B6470474FC7F3A86D626114098C119C8A9A966AE6D4F7242715942 B62E36CD2
62AD8499DC9F3286033903A9C0E2B3847B9A4E5D8F53D37EEE7DAAFD73F61A83A6B09A6945DE80F9A43671C640AE8288688C6A3BBB85145156408001D052D14530643731F9909C0CB0E453
304D349A09A69AA2C334B494B400668CD145007 5FE03BF8E3BAB8B291B0F300D10C75C673FA1FD2BBCAF1FD32E5AD353827 5B9FB3152479DE5EFDA08209C77EB5D35DF886EE0B612DBEBD0
B1775CD83539D9A400F00AE0F1F28AC99B64103BB4BB245FBA36E7AD67DF6B606A6D7904AD3CB8F924618D99183C1EBE959725C4B700CB23EE7CF19E28DC6AE8B4DA832CB22C1B8F9B80C7
D
5AD1148D6116619 5DA720739E3DAB362196C9A6D5D149D8ED7C3769756DACDAC9743E69D247DD9C9246473F4AEE6B92B2B5169E314895B2162DB8F4F9074AEB6A086145145020A28A2800A
34868A6682E696928A005CD1494B400521A28A77018466A472BF65455C67273FA53714633C668B81674CBB1672B4BB32F8C06CE31EB4DB452F77E493B558E48033D0669DA6DAC772CC5FEE
9A97A2B86EAC753667ED1E258EEF6ED3247B8AE73B7E4518CFE15D3573BE1DBA96F02CF719F31B386C603750781F415D0EE14A4947416AC5A4CD0477A42703148A490A0F38A5A414B4C4D2
E18755F4A9A440E854D73B286B798E0E0F634D2B9492944F35349521E951E6A89169696 8A004A5A5C52E2900DC518A7EDA6BFCAA48A603714A91EE941E98EB491032C91A1C9DCC01FC4D5C
D21B8F2C0C061C7E15A5024B2EA70A8CB0909180327A1158DE4400E7ED673FF5CCD5DB591E4BD86DD48676C1572A39C8CF43EC6A5AD06AEDD91DAF86E1789FCA2C1961CA823BE727FC3F3A
5
0471FCBF5ADF0C1C71C5672776696D4703902949151676E714F5396C9A49835D472E7BD3B34C2E0D566BD884C62279AAB872B65B0C0F4ACAD4AD8B7CCA2B4E3DA1723BD23A075C1A130564
00:00:06*FFD8FFE000114A46494600010203040506070809 0AFFDB004300100C0C0E0C0A100E0E0E1212101418281A181616183224261E283A343E3C3A34383840485C4E4044584638385
785C666864FFDB0043011212121616161301A1A3064423842646464646464646464646464646464646464646464646464646464646464646464646464646464646464646464FF
0000000000000102030405060708090A0BFFC400B5100002010303020403050504040000017D01020300041105122131410613516107227114328191A1082342B1C11552D1F02433627282
363738393A434445464748494A5354555657585956363465666768696A737475767778797A838485868788898A929394959697989999AA2A3A4A5A6A7A8A9AAB2B3B4B5B6B7B8B9BAC2C3C4
E1E2E3E4E5E6E7E8E9EAF1F2F3F4F5F6F7F8F9FAFFC4001F0100030101010101010101010000000000000102030405060708090A0BFFC400B5110002010204040304070504040001027700
1
3223281081442 91A1B1C109233352F0156272D10A162434E125F11718191A262728292A35363738393A434445464748494A5354555657585956363465666768696A737475767778797A828
999AA2A3A4A5A6A7A8A9AAB2B3B4B5B6B7B8B9BAC2C3C4C5C6C7C8C9CAD2D3D4D5D6D7D8D9DAE2E3E4E5E6E7E8E9EAF2F3F4F5F6F7F8F9FAFFC000110800800 0A00301220002110103110 1
48A4A795A6E28B0C4C52D18A314AC31696928A2C03854C86A01532500 5B8CD5A4AA695650D3132D2E29E0D42A6A406811DBD84865B085C9CE56ACD67E8ADBB4987DB23F5357C8C907D2A3D
598E1475358134AD713348DC67A0F4AD2D5242225887F1F27F0ACC0A715515D4D20B4B9C0B75A61A99D706A12314DA120A4A31454B28296928A4028352A1A841A916802D230AB08D8AA686
1DBF87A4DFA605FEEB1FE75AB5CFF855C9B6B853D987F5AE82A4996E14514532428A28A00C6BF766BC707EEA8007E59A8D006602992BEF95DCF724D4962C24B8C015A2D11BA5A1C14F80C4
4
3454300A4CD14953601734F53CD474E068193AB54CADDAAAA9AB7161421 0D963D463A526EC87157761D13B38CED20FA53E424274A749234713328C102A9C776F29C1A9526CA70B1DC784CE
523E57200FC335D1535BB3196E14514504851451 4C0E748C8AD0D32D954193A9ED55648F6BB2FA1C56ADA26CB75F7AB7B1B49E8796CF265AAA93934E91B26A3CD55F426E04D359B6A934EA
2F0A6A56F36CB0BC00C883111E4971C9238E9802BB31676E3A463F334F486A4393BD8F1BF34FF77F5A559496C6DFD6BD93ECB0FF0073F5359BAE697F6DD2A7B680F96D20C67AE482081F89
```

Post Processing Images



Thank you.