# Automotive Racing Video Data Logger

ECE445 Fall 2012

Project Proposal

Andrew Wesly, Nick Greenway, Tung Do

TA: Igor Fedorov

September 19th, 2012

# Contents

## I. Introduction

The project goal is to develop a data logger video overlay that can be used in a car. The final product should be able to acquire near real-time engine data through the On-Board Diagnostic Port (OBDII) installed in a car. This product will also take video of a ride through the car's windshield with user supplied standard definition camera. Next, it will overlay OBD data onto the captured video and store it in a portable medium. The video stored in the media should be in a computer readable video format like JPEG2000. Raw OBDII data should also be stored as a file in the storage media. The driver can select playback video on the spot with overlay or replay the video with data at a time later.

This product will aid in automotive racing and tuning by being more intuitive for driver's to use and have better functions than existing products on the market.

Currently, products like Dashware create a data video overlay AFTER a user uploads their video and data to a home computer. This feature is not suitable for amateur racers because they may need to review video and data immediately after a race. Also, this style of operation makes system set-up difficult because they cannot test their logging systems in real-time.

Our logger solves this problem with the abilities to:

- Record video and create an overlay in real-time and
- Use logger hardware to playback captured video.
- Capture standalone OBDII data

According to a few internet reviews, some users prefer to have a data logger record video with OBDII data overlayed on the screen, whereas other users prefer to have raw video recorded and race data recorded separately. Currently, there are products like Track vision that record the raw data onto an overlay but do not allow for separate video/OBDII streams.

Our product solves this problem by allowing the user to manually select a recording mode. They can pick either OBD II on video overlay raw video recorded with an associated OBDII data file for later analysis.

We will also implement an intuitive interface for drivers to use with mechanical switches. Interface with the data logger should be intuitive for fast set-up times and can be done by feel. Some products on the market require a computer to set-up the logger, which is not conducive for race conditions when time between race events is critical for success.

## II. Design

### 2.1 Block Diagrams

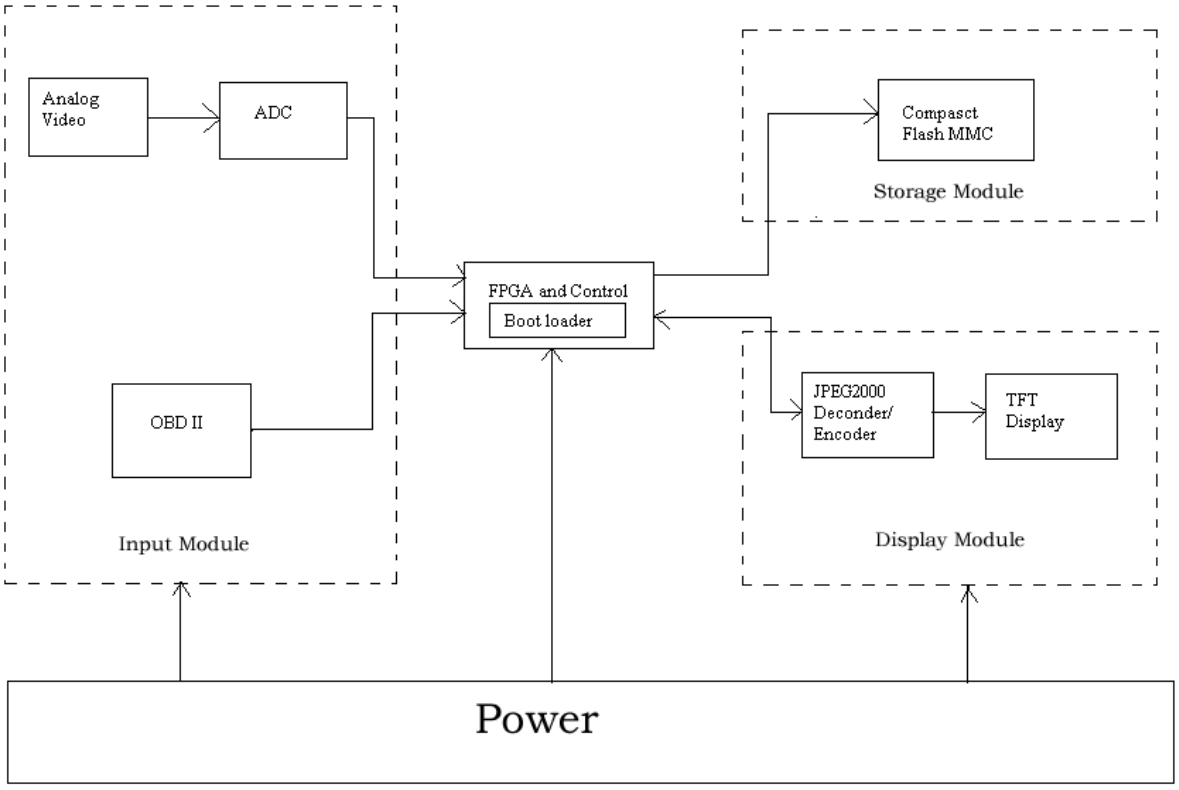


Figure 1: Block diagram

## 2.2 Block Descriptions

1) **Input Data Module:** This module obtains input data such as information from the OBD II and the video images from the camera. The module then converts them into digital form using an ADC and transmits this digital data to the FPGA.
    a. **Analog Video:** The analog video obtained from the camera on the windshield. The image will be sent to the ADC to be converted into digital form.
    b. **ADC:** The Analog-to-Digital converter that converts standard definition analog video input into a digital form so it can be used by the FPGA. ADC input is analog video, which is converted to digital images and are then routed to the FPGA.
    c. **OBD II:** The On-Board Diagnostic system on a car is where information such as the MIL (malfunction indicator light), DTC (diagnostic trouble code), I/M (inspection and maintenance) info, etc. can be extracted from. We will focus on polling operating data like engine RPM, temperature, etc. The information obtained from this block is sent to the FPGA.
2) **FPGA and Control Module:** This module is the "brain" of the entire circuit. It receives the OBD II data and video images from the input module, processes them, overlays the data onto the video stream. It then routes the overlaid video to the storage and display modules. This entity also manages video stream compression/decompression. This block also boots the system.
    a. **Boot loader:** The boot loader is a nonvolatile memory circuit and co-processor that configures the FPGA for use when system is powered on.
    b. **JPEG2000 Decoder/Encoder:** A device that compresses/decompresses digital video for extended storage and computer access. This block receives processed video and data from the FPGA and routes decompressed output back to the FPGA so that video can be routed to other entities.
    c. **FPGA**: The integrated circuit that will be used to design the digital system to process the images and data obtained from the OBD II. This block obtains video from the camera and data from the OBD II, process them, and routes signal through a video compressor/decompressor. FPGA also streams processed to a storage device (the MMC) and controls data display for the attached TFT screen.
3) **Storage Module:** This module receives the overlaid video from the FPGA and control module and stores it on a storage device for later use. A compact flash MMC may be used as the storage device. User can also replay video from the storage medium.
    a. **Compact Flash MMC**: The flash memory data storage device where the video and data obtained will be stored. This block obtains the processed video and data from the FPGA.
4) **Display Module:** This module displays the overlaid video on a LCD screen from storage through the FPGA/control block.
    a. **TFT Display:** The LCD display where the processed video obtained from the FPGA will be displayed.
5) **Power Module:** This module supplies power to the FPGA and Control Module.

## III. Requirements and Verification

### 3.1 Requirements

1) **Input Module:** This module should successfully obtain the captured video from the camera and convert it into digital form using an ADC. Besides, this module also sends the OBD II data to the FPGA and control module without errors.
2) **FPGA and Control Module:** This module should successfully overlay the data obtained from the OBD II on the video received from the input module.
3) **Storage Module:** This module should successfully store the overlaid video without corrupting the file system at a reasonable frame rate.
4) **Display Module:** This module should display the overlaid video without any glitches or freezes.
5) **Power Module:** This module should supply enough steady power to the circuit.

### 3.2 Verification:

1) **Input Module:** The video input can be tested simply by displaying the image from the camcorder onto a screen. The latest OBD II information will be stored in a register, so we can display the contents of the register onto the screen to be sure they are correct. If possible we will use an OBD II simulator to verify our measurements.
2) **FPGA and Control Module:** This can be verified by checking the information from the OBD II in relation to what is present on the screen. Since most of the information needs to have a calculation done to it, we will have to make sure that this calculation is done accurately.
   The signal inputs and outputs will be verified with a logic analyzer to ensure accuracy.
3) **Storage Module:** We will do a simple check to make sure that both of our videos (the raw video and the video with the overlay) are stored properly by playing them back off of the storage device. One method to check this is by displaying the videos through a computer and the other is local playback.
4) **Display Module:** We will verify that the real-time video is displayed on our screen.
5) **Power Module:** We will do a simple check to see if the power supply supplies the expected voltage and that the circuit stays powered on during an extended period of time.

**Details on Requirements and Verifications**

| Requirements | Verifications |
|---|---|
| Input Module:<br>1) Input video signal is analog SD NTSC-M signal<br>2) Output video is ITU-R.656 uncompressed digital video<br>3) Input data signal is OBDII data from car<br>4) Output data is OBDII info over RS-232 interface | Input Module:<br>1) Ensure analog source is powered on and output is enabled<br>2) Check ADC chip power and input format settings<br>3) Ensure car is powered on and right OBD II signal scheme is used<br>4) Make sure OBDII module is powered on and is set correctly |

| | |
|---|---|
| FPGA and Control Module:<br>1) Module passes ITU.656 video with overlay to ENCODER/DECODER chip during record<br>2) Module overlays OBD II data onto ITU.656 stream<br>3) Module takes in OBD II data.<br>4) Module passes compressed JPEG2000 stream to storage module during record<br>5) Module passes compressed JPEG2000 video from storage module to JPEG2000 ENCODER/DECODER chip during playback<br>6) Module processes decompressed JPEG2000 to ITU.656 video and passes it to display module during playback<br>7) User selects video playback and record | FPGA and Control Module:<br>1) Ensure FPGA is powered on<br>2) Ensure digital timings and internal character mappings are at standard<br>3) Ensure FPGA configures OBD II circuit module for correct operation.<br>4) Ensure FPGA is set to route video from each module and has device settings configured properly<br>5) Ensure FPGA is set to route video from each module and has device settings configured properly<br>6) Ensure FPGA is set to route video from each module and has device settings configured properly<br>7) Ensure mechanical switches are connected to FPGA and FPGA can query storage, ADC, OBDII data and display |
| Storage:<br>1) Video stored in computer readable format<br>2) Video read from device is realtime JPEG2000 format | Storage:<br>1) Ensure FPGA commands sent to storage SoC properly set-up file saves<br>2) Ensure FPGA control commands sent to storage SoC properly set-up file reads provided condition in step 1 is met. |
| Display:<br>1) Displays decompressed video equivalent to capture rate<br>2) Video output size maximum of 800X480 resolution | Display:<br>1) Ensure interface clock is configured properly<br>2) Ensure JPEG2000 video is decompressed and scaled properly |
| Power:<br>1) Supply enough voltage to power on the board<br>2) Supply constant power to the board | Power:<br>1) The board is powered on<br>2) The board doesn't power off suddenly |

### 3.3 Tolerance Analysis:

**Oscillator Frequencies**: Verify that the ADC oscillator crystal has a frequency of 27 MHz, plus or minus 50 ppm. We can verify this using either an oscilloscope, an ADC output with logic analyzer, or a frequency analyzer.

## IV. Cost and Schedule

### 4.1 Cost Analysis

a. Labor

| Name | Rate | Hours | Total | Total x 2.5 |
|---|---|---|---|---|
| Tung Do | $50/hr | 180 | $9000 | $22500 |
| Nick Greenway | $50/hr | 180 | $9000 | $22500 |
| Andrew Wesly | $50/hr | 180 | $9000 | $22500 |
| Total | | | | $67500 |

b. Parts

| Description | # | Manufacturer | Vendor | Cost/unit | Total Cost |
|---|---|---|---|---|---|
| ADV212BBCZRL-150 JPEG2000 Encoder/Decoder | 1 | Analog Devices | Digikey | $48.984 | $48.984 |
| Development Board | 1 | Altera DE2 | NA | $269 | 0 - Have |
| ELM327 OBD Interpreter | 1 | Elm Electronics | ELM Electronics | $23.50 | $23.50 |
| MAX9526 Video Decoder | 1 | Maxim Integrated | Mouser | $7.97 | $7.97 |
| OBD-II UART | 1 | Sparkfun Electronics | | $49.95 | $49.95 |
| Control Switches | 6 | C&K Compnents | Mouser | $6.25 | $37.50 |
| ALTERA EP2C35F672C6 FPGA | 1 | Altera | Altera | $149.50 | $149.50 |
| Digilent Basys2 Development Board – used as bootloader | 1 | DigilentInc | DigilentInc | $100.00 | NA - Have |
| ALFAT-SD-338 – FAT32 SD card writer | 1 | GHI Electronics | GHI Electronics | $44.95 | $44.95 |
| K50DWN2-VI-FE – TFT Display | 1 | Kentec | Newark | $65.22 | $65.22 |
| 10085901-6015elf– TFT Display Connector | 1 | FCI | Mouser | $1.46 | $1.46 |
| G.SKILL 32GB Micro SDHC Flash Card w/ SD Adapter Model FF-TSDG32GA-C10 – for storage. Equivalent is fine | 1 | G.SKILL | Newegg | $27.99 | $27.99 |
| PCB – All boards | 1 | PCBFABEXPRESS | PCBFABEXPRESS | $200 | $200 |
| Total | | | | | $657.024 |

| | | | | |
|---|---|---|---|---|
| Total Project Cost | = | Total Labor Cost | + | Total Parts Cost |
| | = | $67,500 | + | $657.024 |
| | = | $68157.024 | | |

### 4.2 Schedule

| Week | Task Description | Group Members |
|---|---|---|
| 9/16 | **Proposals Due** | |
| | Software design for storage and boot loader | Tung Do |
| | Software design for OBD II and ITU-R BT. 656 | Nick Greenway |
| | Electronic hardware specifications/designs | Andrew Wesly |
| 9/23 | **Design Reviews Sign-up Closes** | |
| | Software development for storage interface | Tung Do |
| | Software development for OBD II interface | Nick Greenway |
| | Finalize the schematic | Andrew Wesly |
| 9/30 | **Design Reviews** | |
| | Test storage interface | Tung Do |
| | Test OBD II interface | Nick Greenway |
| | Board layout/order components | Andrew Wesly |
| 10/7 | | |
| | Software development for Boot loader | Tung Do |
| | Software development for Video interface | Nick Greenway |
| | Submit board for fab | Andrew Wesly |
| 10/14 | | |
| | Test boot loader | Tung Do |
| | Test video interface | Nick Greenway |
| | Assemble the boards | Andrew Wesly |
| 10/21 | **Individual Progress Reports Due** | |
| | Test overlaid video on storage device | Tung Do |
| | Verify correctness of OBD II data obtained | Nick Greenway |
| | Test assembled boards | Andrew Wesly |
| 10/28 | | |
| | Assemble boot loader | Tung Do |
| | Assemble video | Nick Greenway |
| | Power board layout& Assemble FPGA | Andrew Wesly |
| 11/4 | **Mock-up Demos and Mock Presentation Sign-up closes** | |
| | Mock Presentation – Control Slides | Tung Do |
| | Mock Presentation – Test Slides | Nick Greenway |
| | Mock Presentation – Design Slides | Andrew Wesly |
| 11/11 | **Last Day to Request 1st Revision PCB Fabrication** | |
| | Final software testing | Tung Do |
| | Final software testing | Nick Greenway |
| | Assemble everything in an enclosure | Andrew Wesly |
| 11/18 | **Thanksgiving Break, Last day to Request Final Revision PCB fab** | |

| | | |
|---|---|---|
| 11/25 | **Demo and Presentation Sign-up closes** | |
| | Final tests and verifications | Tung Do |
| | Final tests and verifications | Nick Greenway |
| | Final tests and verifications | Andrew Wesly |
| 12/2 | **Demos and Presentations** | |
| | Demo, Final Paper – Intro & Conclusion | Tung Do |
| | Presentation, Final Paper – Testing & Verification | Nick Greenway |
| | Demo, Final Paper – Design Procedure | Andrew Wesly |
| 12/9 | **Presentations, Checkout, Final Paper, Lab Notebooks** | |
| | Final Paper & Notebook review and hand-in | Tung Do |
| | Final Paper & Notebook review and hand-in | Nick Greenway |
| | Final Paper & Notebook review and hand-in | Andrew Wesly |