EEE190: SENIOR DESIGN

SEPTEMBER 10, 2013

Design Document

Authors:
Micheal FRITH
David LARRIBAS
Devin MOORE
Benjamin SMITH

Supervisors:
Fethi BELKHOUCHE
Russ TATRO

Abstract

Index Terms

keyword1, keyword2, keyword1, keyword2, keyword1, keyword2, keyword1

I. Addressing the Societal Problem

THE commute provides an opportunity for office workers to work exercise into their otherwise sedentary and busy days. The advantage over a purely motorized vehicle is the requirement for the rider to provide power to some extent enabling physical activity. The responsive control system allows a user to set an exertion amount and tailors itself to the users wants.

II. Skill Sets

THERE will be a variety of different skills required to develop this system. Many of them are currently present in our group, however some will need to be learned and or refined. The most significant technological hurdles will lie within the exertion based control algorithm and the electronic speed control for the CPEs and EEEs respectively.

*a) **Mike Frith:*** Front Flip Masta

*b) **David Larribas:*** Experience with lighting and voltage conversion. Cycle commuter

*c) **Devin Moore:*** Verilog and FPGA development ready. Experience with C++, C, and Java. Worked at bike shop from 2007 to 2010 and cycle commuted 2008 to 2010.

*d) **Ben Smith:*** Verilog, FPGA development, Experience with NIOS II C development. Worked at bike shops since 2004, still employed at the Bicycle Emporium. Cycle commuted from 2005 to 2008 along the American River Parkway.

*e) **Qualifications for success:*** Smart control can be tested by isolating a control variable and manipulating it while recording the systems output data. The control algorithms are to be developed in a simulation environment and tested against generated data. A successful match between theoretical and experimental data will validate the systems performance.

III. "Punch List"

THE "punch list" is a listing of features to be implemented in the final product. These core features represent the bulk of work to be done in the project. They Fall into 5 basic categories

- Smart Assist
  - FPGA based control mechanism
  - Inclanation based speed control
  - Effort based speed control
  - Effort based speed control
- Motor and Motor Control
  - Commercial motor integrated into design
  - Custom build Electronic Speed Control
  - Motor data: temperature, voltage, current, rotation speed
- Power Distrobution
  - 5V subsystem for control components
  - Battery charging circuit
  - Battery monitoring: Charge, Voltage, Current
- Safety and Lighting
  - Function turn and brake light
- User interface and Controls
  - Bluetooth Android control application

A. Smart Assist

The amount of motor assist will be controllable in real time. This combined with a micro controller and sensors will allow motor response tailored to the users need. The motor will have four modes of operation.

- Constant effort via feedback loop (HRT or Torque)
  PI feedback control system based on exertion measurement
- Inclination based assistance
  linear wattage response to change in gradient
- Constant speed mode
  PID based control based on speed sensor
- Constant Wattage mode
  basic wattage setting with no digital control.

These modes offer the user a variety options that require the use of external sensors or not. The constant wattage mode will need to be selectable and controllable via physical buttons on the device. The rest of the modes will be controlled by the Android application over the bluetooth connection.

The motor must also turn off when the brakes are applied. There are a number of systems that detect the application of the brakes. The signal line from these sensors will be run into the FPGA to turn the motor output signal on and off.

To implement the NIOS II soft processor and other HDL entities we will use Alteras Quartus 13sp1 development environment. The actual programming of the Soft Processor will take place in a separate development environment called Nios

II Software Build Tools for Eclipse Indigo. Development of the control system in expected to require significant effort. The MicroC RTOS will be implemented on the NIOS processor to handle task scheduling on the NIOS core.

1) Central Processor: We will be using the NIOS II soft processor implemented on an Altera Cyclone IV FPGA. During the breadboard phase will use the Terasic DE0-Nano development board. This would mainly be used to communicate with peripherals through UART and Bluetooth. This would also package and store user data in a useful manner onto an SD card to be uploaded to PC or smart phone.

2) Heartrate Based Control: The Zephyr HxM uses the bluetooth SPP (slave) protocol to communicate. the RN-41 bluetooth module we plan to use can communicate with the heart rate strap operating in master mode. Adding this feature will allow motor torque to be varied in near real time automatically based on the users need. An example implementation would be to use the Android application UI to specify a heart rate ceiling. As the user approaches the ceiling value the output wattage of the motor is increased to provide assistance. The downside is a 10 to 20 second delay from increase in exertion and heart rate response. The benefit is we can provide a feedback system without having to implement the mechanically complicated torque measurements. [1]

a) Referenced control systems: An SOPC based Intelligent Bike The authors of this paper developed a system to monitor tire pressure and and gear selection the most novel portion of their implementation is the use of cellphone RF to indicate the presence of other riders. [2]

Applying Fuzzy Logic Control to an Electric Bicycle This paper details the Author's control system based on inclination. A control system was built around the rotational speed of the rear wheel. This system generated a PWM signal using a Programmable System On Chip. [3]

Task	**Labor Hours**
SPI for ADC	16
I2C for IMU	20
NIOS II Core in Quartus	2
MicroC OS for NIOS II	2
Verilog PWM Generator	2
Verilog Motor Control	20
Verilog Sensor Parser	10

TABLE I: Estimated Time for smART Control

B. Motor and Motor Control

b) ***Motor Selection:*** Our motor shall be contained within the rear hub of the bicycle. Placing the motor in the rear will ensure the greatest traction while climbing. Rear mounting also attaches the motor to the sturdiest portion of the frame. We have chosen a brushless design due to their availability, durability, and reliability. Unlike a brushed DC motor, brushless designs dont have electrical contacts in the moving portion which can get contaminated or wear out. The motor will additionally have the traditional bicycle gearing attached.

Task	**Labor Hours**
–FILL IN–	16
–FILL IN–	20
–FILL IN–	2

TABLE II: –FILL IN–

c) ***Motor Electronic Speed Controller:*** The motor shall be initially driven using an off the shelf electronic speed controller. This speed controller will accept a pulse width modulated signal generated by our master control unit and convert the the incoming signal to a 3 phase signal to power the motor.

We shall design our own ESC to control the motor using the same pwm signal, however we will be creating our own design to get more data out to the master control unit while incorporating a power supply to get power to our master control unit and auxiliary systems.

The electronic speed controller shall use a microprocessor to trigger 6 MOSFETS to conduct current to the motor windings. It shall process data to feed back to the control unit. Ideally we would like rotational speed, temperature, current, voltage, and torque data.

Task	Labor Hours
–FILL IN–	16
–FILL IN–	20
–FILL IN–	2

TABLE III: –FILL IN–

C. Power Distribution

*d) **USB power:*** This will be a hub that will supply power and operate the various external functions of the project including the lighting and sound. Additional slots will be added providing power for whatever USB device the user chooses. The main purpose for the extra slots are for charging a cell phone which the user would use to adjust settings. This hub would provide six female USB slots; four of these would be controlled by the FPGA and the other two would be always powered. The four controlled ports would be used for the lights and sound. The hub will take power from the battery and step down the voltage to 5V and limit the current to 700mA for each output.

Task	Labor Hours
–FILL IN–	16
–FILL IN–	20
–FILL IN–	2

TABLE IV: –FILL IN–

D. Safety and Lighting

*e) **Motor Safety:*** A number of safety systems need to be considered in the development of the control algorithm. All operation modes must have a speed limit of 25 mph in accordance with state law. Accelerometer data can be utilized to ensure the rider is upright and disable the drive motor in the event of a fall for an added measure of safety.

A hard kill switch is to be between the battery and motor, this relay is to be held closed by the NIOS soft processor. The signal line is to be use a pulldown resistor to ensure the relay will open in the event of a software failure. A physical handlebar switch should be also attached to the motors kill system so the user can kill the motor without having to rely on the FPGA control system.

*f) **Battery Safety:*** The volatile lithium polymer will need to have it's operation monitored to deal with a number of conditions which can harm the battery or lead to catastrophic failure. A robust charging system will be needed to prevent the battery from harmful overcharge or undercharge conditions. Short circuit protection will be needed as lithium polymer batteries are specifically susceptible to damage from high current conditions. Temperature and charge control mechanisms can be built into the FPGA.

*g) **Lighting:*** The lighting will consist of a headlight, brake light, and front and rear blinkers. The headlight will illuminate the area in front of the bicycle in order to increase the riders visibility and perceptibility. The lighting must not interfere with a drivers ability to see and thus must not shine directly into a drivers eyes. LEDs will be the illuminator because of their small viewing angle which allows them to have a concentrated beam of light without the use of reflection and their low energy consumption. Some number of LEDs are to stay illuminated at all times the bike is in operation to increase visibility to drivers. The rest of the LEDs in the headlight, which will be used for illuminating the road in front of the rider, will be activated manually or through light detection to activate automatically in dark areas.

The blinkers will consist of lights mounted on the rear, and possible front, of either side of the bicycle. The left light flashes when the rider flips the toggle up and the right light flashes when the toggle is down to indicate which direction the rider intends to turn. Both lights remain off when the toggle is in the center position. Blinkers may be combined with the brake lights to reduce number of units.

The brake and tail light will be a red light or set of lights will be on at all times the vehicle is in operation at a reduced intensity. When the rider squeezes the braking lever, the brake lights will engage with full intensity to alert people behind that the rider is slowing down significantly or stopping.

This feature will not require any software to run and will only need simple hardware: resistors, LEDs, wires, switches and maybe some others if different modes are to be hardwired.

Task	Labor Hours
Design Schematics	4
Proto Wiring and Testing	2
Design Casings and Mount	2
Fabricate Lights	15
Fabricating Control Unit	3
FPGA Logic Module	1
–FILL IN–	10

TABLE V: Estimated Time for Lighting and Safety

E. *User Interface and Controls*

*h) **Cell Phone Application:*** The cellphone application will primarily act as the riders dashboard. It will display the riders speed, heart rate, location, and state of the system including battery charge, rate of discharge, distance to empty estimation and power mode. The app is meant to be used from a cell phone mounted to the handlebars much like the commercially available gps systems. The application will communicate with the NIOS II through Bluetooth and store required data on the SD card.

The user will also be able to change the power mode and alter the system settings from the application. This should be the only interface to the system that the rider has to worry about, besides a physical kill switch in case a reset is required.

The implement this system we will use Nios II Software Build Tools for Eclipse Indigo to program the FPGAs Bluetooth utility and use the Eclipse IDE with built-in Android Developer Tools and Android SDK components for programming the cell phone application.

Task	Labor Hours
Application UI	2
Data Processing/Presentation	5
Data Storage Procedure	2
NIOS II Bluetooth	2
Cell Phone App Bluetooth	2

TABLE VI: Estimated Time for Cell Phone Integration

IV. Estimated Costs

Part	Cost in Dollars($)	Purpose
DE0-Nano	100	Central Processor
9 DOF IMU	100	Orientation Sensors
Bluetooth	50	Communication
Electric Motor	150	Propulsion
Motor Controller	100	PWM to Motor
Battery	400	Power
BT4 Heart Rate	70	HR Monitering
Bicycle	FREE	Framework
Cell Phone	FREE	UI/Controls
Miscellaneous	75	Framework
TOTAL	1050	N/A
TOTAL +35%	1417	Room For Error

TABLE VII: Cost Estimation

References

[1] Zeypher. (2010, Mar.) Hxm bluetooth api guitde. [Online]. Available: http://www.zephyranywhere.com/media/pdf/HXM1_API_P-Bluetooth-HXM-API-Guide_20100722_V01.pdf

[2] C. I. . K. J. . V. A. . R. R. Prabhu, S. ; PARK Eng. Coll., "An sopc based intelligent bike," *Conference on Computer Modeling and Simulation*, vol. 4, 2010.

[3] B. C. Chi-Ying Liang, Wai-Hon Lin, "Applying fuzzy logic control," *An SOPC Based Intelligent Bike*, 2006.