MicroCART Design Document

Group May15-28

Paul Gerver, Tyler Kurtz, Joe Benedict,
Adam Campbell, Matt Vitale, Ravi Nagaraju, Jacob Rigdon

10/28/2014

Contents

List of Figures

List of Tables

1 Definition of Terms

I^2C - A serial bus protocol invented by Philips.
RC Trainer - Device that controls thrust, roll, pitch, and yaw through positional pulse modulation. Another term for RC controller.

2 Abbreviations

ADC: Analog-to-digital converter
BT: Bluetooth
CAD: Computer-aided design
DSP: Digital Signal Processing
ESCs: Electronic Speed Controllers
FPGA: Field Programmable Gate Array
Gyro: Gyroscope
I2C: Inter-Integrated Circuit
IR: Infra-red
MicroCART: Micro-Controller Aerial Research Team
m.o.i: moment of inertia
PID: Proportional-Integral-Derivative
PMOD: Peripheral Module
PPM: Positional pulse modulation
PWM: Pulse width modulation
quad: quadcopter
RAM: Random Access Memory
RC: Radio Controller
XPS: Xilinx Platform Studio
XSDK: Xilinx Software Development Kit

3 Executive Summary

The goal of this project is to design, integrate, and implement a new quad system to replace the current system for the Distributed Sensing and Decision Making Laboratory at Iowa State University. The current system requires an upgrade to remove unnecessary and slow bottlenecks, like an RC mixer, and needs better data analysis tools for comparing measurements from a variety of sensors. These components are all necessary as the system must be able to handle new workloads and environments.

4 System Design

4.1 System Description

4.1.1 Overview

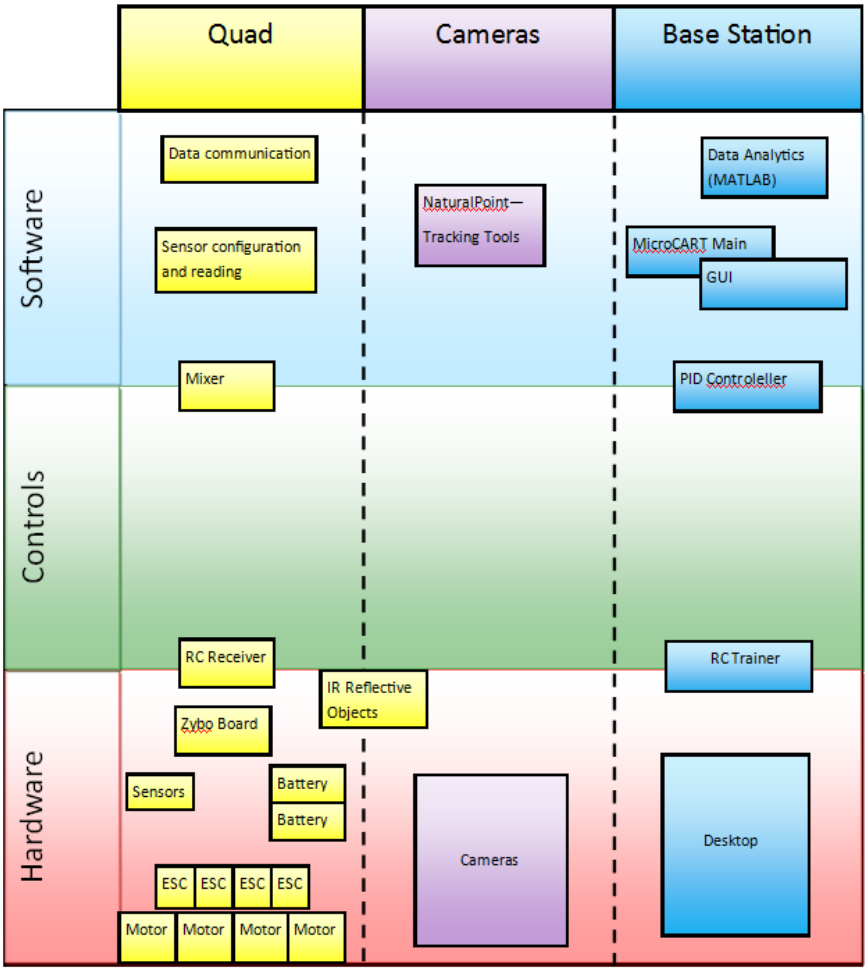

Figure 1: System concept block diagram

This project is an expansion of multiple years of work within the Distributed Sensing and Decision Making lab geared toward preparing a new platform for research. The new system that will be in place allows for multiple projects to be developed at the same time and each can be unique since the

system will be versatile with swappable peripherals and reconfigurable hardware. In other words, the new system can be used for application-specific or research projects depending on the client's interests.

4.1.2 Quad System

The quad system is the mobile unit in the group that performs most of the actuations in the system. Specifically, the quad contains four motors that allow it to move in three dimensional space and receives control signals that is sent by the base station by means of an RC receiver. The hardware for the quad consists of the physical frame, motors, ESCs, reconfigurable processing board, and multiple sensors and peripherals like a Bluetooth module, gyroscope, and accelerometer.

4.1.3 Base Station

The base station is the main processing unit that handles the controls of the system. The software on the desktop includes the system program for running flights through a manual RC trainer or automatically, and it has data analytic tools for interpreting flight data. Since the base station is responsible for controls, PID controllers are incorporated into the main program to handle the error within the feedback control loop to safely control the quad to a desired altitude, longitude, and latitude. For hardware, the base station consists of a standard desktop running Red Hat 6 Linux with a Nexsys2 Spartan-3E FPGA board used as the interconnect between the computer and the RC trainer to send commands through PPM waves to the quad.

4.1.4 OptiTrack Cameras

The 12 high-speed IR cameras located in the Distributed Sensing and Decision Making Lab are used for tracking the location of the quad within its range. The camera system relays positional information to the base station that will incorporate its measurements into its controls. The team is working on removing the requirement to have the cameras in order to fly the quad; however, the team is utilizing them to test and compare with on-board sensors for accuracy.

4.2 System Requirements

4.2.1 Functional

Software

Quad System

The software for the new quad system needs to be able to collect several readings from its main sensor peripheral through I2C protocol, communicate data through Bluetooth to the base station, receive PPMs through an RC receiver, and appropriately mix incoming position signals to pulse wave modulations for each of the four motors on the quad.

Additionally, the processing board does contain volatile memory and a reconfigurable FPGA, so the board needs to be configured without having to plug it into a computer.

Base Station
The base station is the heart of the system so it needs to be able to communicate or receive data from all devices in the system including the OptiTrack camera system, the RC Trainer, and the Zybo board located on the quad itself. Some tools already exist to communicate with the RC controller and the cameras, but project members will still learn about these components in order to incorporate them into the design of the new system.

The base stations also requires data analytic tools to interpret the diverse range of data being received and transmitted. Again, the data collected will be utilized ins PID controller as well as post-flight analysis for testing measurements against the camera and the quad's on-board sensors. The PID controller in software needs to be tuned properly to the new system's quad to properly fly it.

Hardware

Board and Peripheral Mounting
For the quad to operate, the Zybo board, among its peripherals, need to be mounted and placed somewhere on the new quad chassis so the Zybo board can operate the motors. Additionally, the main sensor board that measures gyro and accelerometer data needs to be placed at the center of gravity on the quad to receive useful data. In regards to how the board will be powered, an external, mobile, and replaceable power supply is placed on the carrying the platform to power the board, ESCs, and motors so the quad can operate in a wide environment and not be restricted.

Power Supply
There are a few requirements for the power supply batteries, both to the quad motors and the Zybo board, including a need for proper voltage regulation, detecting reverse polarity, and not allowing for overdrain so the batteries do not get damaged.

4.2.2 Non-Functional

Software

Quad System
There are many timing constraints that the quad should meet and the data bottleneck should be minimal when communicating data back to the base station so the quad can have a smooth flight. The system should also be reliable and shouldn't need to be reset or configured after each flight, or behave strangely when launching the main program. Overall, users shouldn't have to touch the software aspects of the quad when starting the basic flying program.

Base Station
The software for collecting data from the IR cameras and sending the correct instructions through the Nexsys2 board should not be of any concern to users wanting to fly the quad. These components should also be reliable and need to be tampered with unless absolutely necessary. The data analytic tools also need to be automated so users do not need to dig through log files and look for scripts to run in order to view results from a flight. Lastly, the PID controllers on the system need to be optimal to make flying the quad easy and safe for users.

Hardware

Board and Peripheral Mounting
The most important aspect of the new system is that it does not injure any of its components when testing. For this reason, the placement of the Zybo board and its peripherals on the quad chassis is critical to not damage equipment which can cost time and money to replace. The mounted board should not be the tallest point on the quad since it is likely the quad will tip over upside at some point, and it should be protected and stable during test flights. Additionally, accessibility should be considered when mounting the board and peripherals so they are fairly easy to access and remove if needed.

Power Supply
The power supply batteries used in the system need to be monitored and handled properly when their power is too low to operate the quad. For maintainability of the quad, the quad needs to safely land rather than stop altogether when in a flight, and the batteries need to not be damaged in the process.

4.3 System Analysis

4.3.1 Modeling and Simulation

Characterization of the new quad system will conducted to obtain key constants that can be used to model the system in Simulink or MATLAB to identify proper control parameters. For the PID controls for the quad, several simulated and actual tests are being conducted to identify appropriate P, I, and D constants for each of the four channels of control: thrust, roll, pitch, and yaw. To make modeling easy at first, these parameters are obtained through extensive testing using equipment that constrain the quad to only one axis of rotation at a time.

The main characterizations that will be collected for the quad are detailed in Table 1.

Symbol	Description
J_{reg}	rotor+motor m.o.i around motor axis of rotation
K_T	propeller thrust constant
K_d	rotor drag constant
R_m	motor winding resistance
K_Q	motor torque constant
i_f	motor internal friction current
K_v	motor back-emf constant
m_q	vehicle mass
m_b	battery mass
α	angular acceleration of disk+rig+quad
Θ	angular displacement
τ	applied torque

Table 1: Quad characterization variables

4.4 Operating Environment

The new system is being developed and tested within the Distributed Sensing and Decision Making Lab. The lab has different electrical equipment for signal debugging, physical component characterization, charging batteries, computers for programming the Zybo board and data analysis, and the OptiTrack IR Camera system for position measurements. This operating environment is ideal for quad testing and flying as there are no natural forces like wind that disrupt conditions.

5 Detailed Design

5.1 Hardware Specification

5.1.1 DJI FlameWheel F450

The new quad system utilizes a four arm DJI FlameWheel to house and carry the proper components in order to fly the system. The chassis has two platforms: one that sits on the intersection of the four arms, and one below that acts as a base. Both of the platforms are conductive and they share a single pair of wire leads for powering the ESCs and motors tightly constrained to the chassis' arms. The exact specifications and recommendations for the FlameWheel can be seen in Table 2.

Spec	**Description**
Frame Weight	282g
Diagonal Wheelbase	450mm
Takeoff Weight	800g $\sim$ 1200g
Recommended Propeller	10 x 3.8in; 8 x 4.5in
Recommended Batter	3S $\sim$ 4S LiPo
Recommended Motor	2212 $\sim$ 2216 (Stator Size)
Recommended ESC	15A $\tilde{2}$5A

Table 2: Specifications of the DJI FlameWheel F450

5.1.2 DJI Motors

The motors are the key actuators in the system that eventually move the quad with the help of propellers. The DJI Motors are stepper motors that are controlled by magnets inside the motor that generating precise magnetic fields to make the motor rotate rapidly. With four total motors on the quad, proper mixing of signals is required for the motors to change their number of resolutions per minute so the quad can turn or rotate correctly. The specifications for the DJI motors can be seen in Table 3.

Spec	**Description**
Stator Size	22 x 12mm
Brushless Motor KV	920rpm/V

Table 3: Specifications of the DJI Brushless Motors

5.1.3 ESCs

The ESCs act as the middle-man between the Zybo board and the motors. The ESCs are responsible for translating the pulse wave modulation sent by the Zybo board into how much power to give the motors in order to spin. The ESCs are attached to the quad frame by zip ties and are powered by the conductive platforms on the frame. The ESCs the team are using are specified in Table 4

Spec	**Description**
Electric Current	30A OPTO
Compatible Signal Frequency	30Hz - 450Hz
Battery	3S $\sim$ 4S LiPo

Table 4: The specifications for the DJI ESCs

5.1.4 Diligent ZYBO Zynq-7000 Development Board

The Zybo board is the core processing unit on the quad system responsible for mixing controls from the base station and handling gyro and accelerometer data for its sensors. Additionally, the board is responsible for communicating data appropriately to the base station through the use of Bluetooth or WiFi. A full list of features the Zybo board offers can be seen in Table 5.

Component	**Description**
Processor	650MHz dual-core Cortex-A9 Processr
Memory	DDR3 Memory Controller with 8 DMA channels
High-bandwith peripheral controllers	1G Ethernet, USB 2.0, SDIO
Low-bandwith peripheral controllers	SPI, UART, I2c
FPGA	28K logic cells 240KB Block RAM 80 DSP slices On-chip dual channel, 12-bit ADC
Misc.	MicroSD slot 6 pushbuttons, 4 slide switches, 5 LEDs 6 PMOD connectors On-board JTAG programming and UART to USB converter

Table 5: Features for the Diligent ZyBo Board

The Zybo board is used on the new system because it has 6 PMOD connectors (see Figures 2 and 3) and an FPGA that can programmed to be utilize different peripherals including the 9 Degrees of Freedom sensor board, Bluetooth module, and the outputs signals for controlling the motors. Another reason the board was selected is due to its many internal controllers like UART and I2C so developers can use the FPGA for hardware acceleration or even on-board PID controlling.

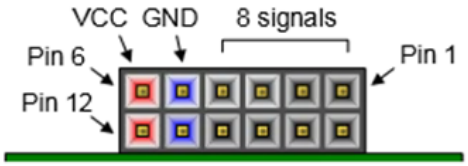

Figure 2: PMOD Diagram

Pmod JA (XADC)	Pmod JB (Hi-Speed)	Pmod JC (Hi-Speed)	Pmod JD (Hi-Speed)	Pmod JE (Std.)	Pmod JF (MIO)
JA1: N15	JB1: T20	JC1: V15	JD1: T14	JE1: V12	JF1: MIO-13
JA2: L14	JB2: U20	JC2: W15	JD2: T15	JE2: W16	JF2: MIO-10
JA3: K16	JB3: V20	JC3: T11	JD3: P14	JE3: J15	JF3: MIO-11
JA4: K14	JB4: W20	JC4: T10	JD4: R14	JE4: H15	JF4: MIO-12
JA7: N16	JB7: Y18	JC7: W14	JD7: U14	JE7: V13	JF7: MIO-0
JA8: L15	JB8: Y19	JC8: Y14	JD8: U15	JE8: U17	JF8: MIO-9
JA9: J16	JB9: W18	JC9: T12	JD9: V17	JE9: T17	JF9: MIO-14
JA10: J14	JB10: W19	JC10: U12	JD10: V18	JE10: Y17	JF10: MIO-15

Figure 3: ZyBo Pin out Table

For programming the Zybo board, users can develop a bitstream file through XPS to be flashed onto the FPGA that can configure the processing system and programming logic portions of chip on the board. The actual process of programming is done either through UART JTAG if connected to a computer in the XSDK or through a microSD card if set up properly.

5.1.5 SpekTrum DX6i RC Controller

The RC trainer is the main device for communicating the thrust, roll, pitch, and yaw levels to the quad system. The RC trainer is connected to the base station through the Nexsys2 FPGA and sends 50 PPMs a second. The RC receiver located on the quad receives the 6 channel PPM and splits the channels to individual PWMs for each flight dynamic. The specifications for the SpekTrum RC trainer can be seen inf Table 6.

Spec	**Description**
Modulation	DSM2
Band	2.4GHz
Receiver	AR6200
Programming Features	Helicopter & Airplane
Model Memory	10
Modes	Mode 2
Transmitter (Tx) Battery Type	AA NiMH 1500mAh batteries
Charger	4-cell 150mAh wall charger

Table 6: Specification of the SpekTrum DX6i RC Controller

5.1.6 Voltage Regulator

The voltage regulator is key to powering the Zybo board while on the quad. The regulator is also responsible for identifying the battery level, detecting reverse polarity, and stopping power from being drawn from the batter after certain level. The specifications for the voltage regulator can be seen in Table 7, the circuit diagram and voltage and power graphs can be seen in Figure 4 and 5, and lastly the bill of materials can be seen in Figure 6.

Spec	Description
Desired Voltage Output	5 V
Desired Current Output	2.5A
Expected input battery voltage range	6.5V - 8.4V

Table 7: Specifications for the voltage regulator

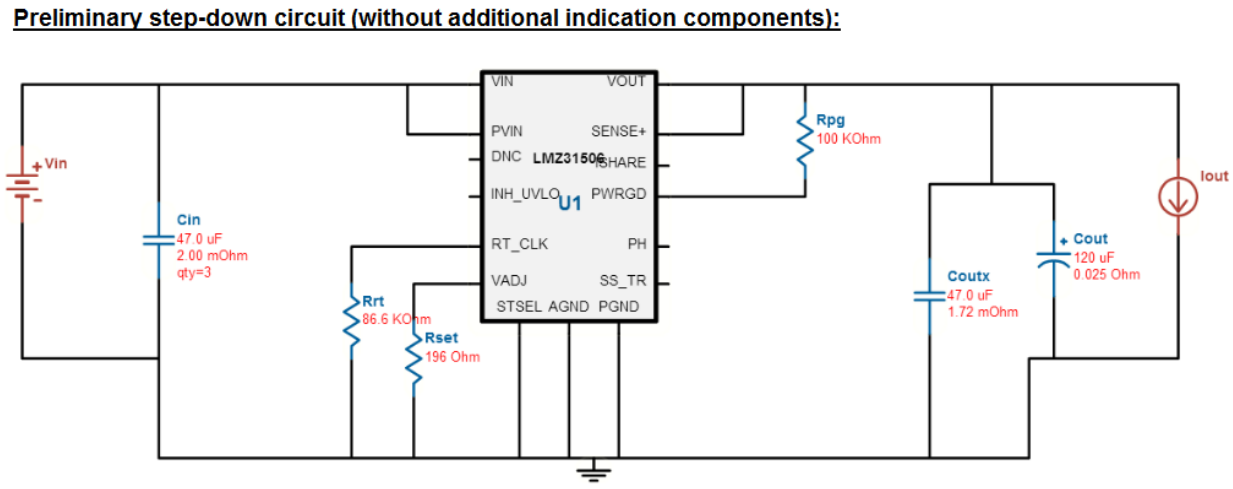

Figure 4: The circuit design for the voltage regulator

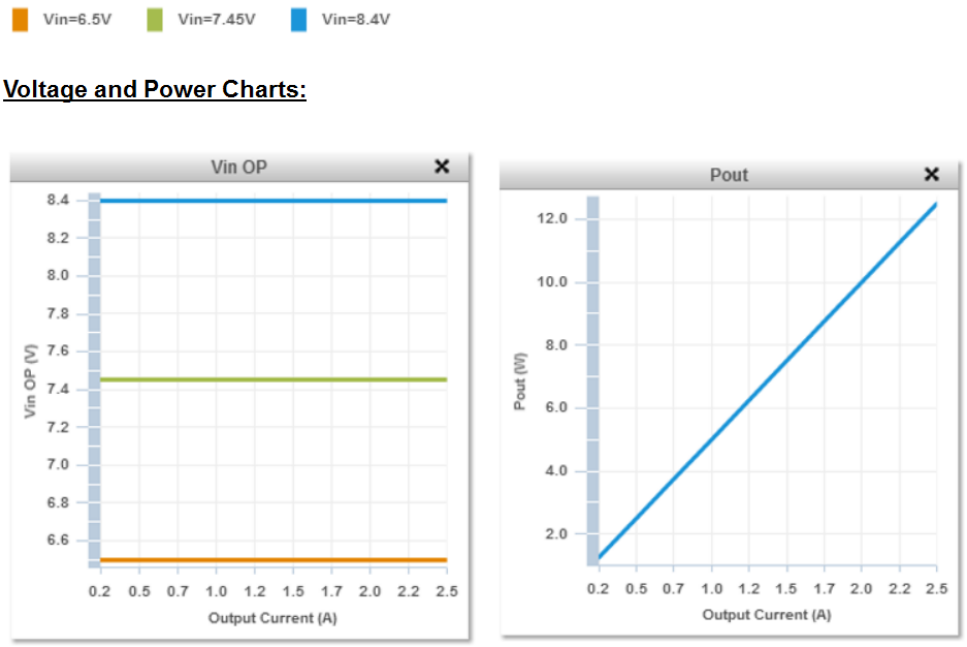

Figure 5: Voltage and power graphs for voltage regulator

Part	Manufacturer	Part Number	Quantity	Price	Attributes	Footprint ▲	Top View
Cout	CUSTOM	CUSTOM	1	NA	Cap=47uF, ESR=4mOhm, VDC=6V	NA	
Cpvin	TDK	C1005X5R1A104K	1	$0.01	Cap=100nF, ESR=0.02Ohm, VDC=10V	3	-
Rset	Vishay-Dale	CRCW0402196RFKED	1	$0.01	Resistance=196Ohm, Tolerance=1%, Power=0.063W	3	-
Rt	Vishay-Dale	CRCW040263K4FKED	1	$0.01	Resistance=63.4kOhm, Tolerance=1%, Power=0.063W	3	-
Cin	MuRata	GRM32ER61E226KE15L	2	$0.28	Cap=22uF, ESR=2mOhm, VDC=25V	15	
Cinx	United Chemi-Con	APXE100ARA121MF61G	1	$0.43	Cap=120uF, ESR=0.025Ohm, VDC=10V	74	
U1	Texas Instruments	LMZ31707RVQR	1	$6.50		144	

Figure 6: Materials needed for voltage regulator

5.1.7 SparkFun MPU9150: 9 Degrees of Freedom

The 9 Degrees of Freedom board contains several useful sensors including a gyroscope, accelerometer, temperature sensor, and a magnetometer. The specifications for the board can be seen in Table 8.

Spec	Description
VDD	2.375 - 3.465V
VLogic	Min = 1.71V Max = VDD 1.8 ± 5% OR VDD
Absolute Maximum Ratings: VDD	-0.5 to 6V

Table 8: Specifications for the SparkFun MPU9150

The board uses a I2C bus protocol to communicate data which the Zybo board will be configured and program to accommodate it. The following figure indicates the pin outs of the board.

Pin Number	Pin Name	Pin Description
1	CLKIN	Optional external reference clock input. Connect to GND if unused.
6	ES_DA	Auxiliary I^2C master serial data
7	ES_CL	Auxiliary I^2C Master serial clock
8	VLOGIC	Digital I/O supply voltage
9	AD0	I^2C Slave Address LSB (AD0)
10	REGOUT	Regulator filter capacitor connection
11	FSYNC	Frame synchronization digital input. Connect to GND if unused.
12	INT	Interrupt digital output (totem pole or open-drain)
3, 13	VDD	Power supply voltage and Digital I/O supply voltage
15, 17,18	GND	Power supply ground
20	CPOUT	Charge pump capacitor connection
22	CLKOUT	System clock output
23	SCL	I^2C serial clock (SCL)
24	SDA	I^2C serial data (SDA)
2, 4, 5, 14, 16, 19, 21	RESV	Reserved. Do not connect.

Figure 7: SparkFun MPU9150 Pin Out and Signal Description

5.1.8 Diligent PMOD-BT2

For non-control communication, a Bluetooth device will be used on the Zybo board to send data to the base station for analysis or for PID controller integration. The team will be utilizing a Diligent PMOD-BT2 device that can easily be plugged into the Zybo board and the pin out for the module can be seen in Figure 8.

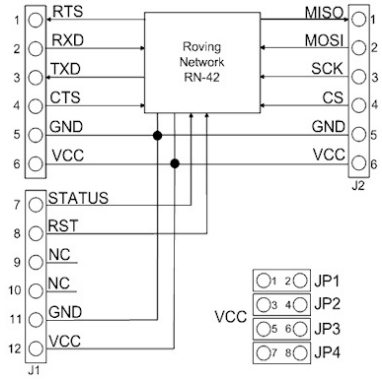

Figure 8: Diligent PMOD-BT2 Diagram

5.1.9 OptiTrack IR Cameras

The OptiTrack cameras in the lab are set to capture reflected IR waves from the quad with the proper pieces of equipment. In order to track the quad's location, the quad frame carries a set of reflective IR balls that the cameras can track. Again, the data produced from the cameras are sent to an intermediate desktop through USB 2.0 where it is then sent to the base station through WiFi. More specifications for the camera system can be seen in Table 9.

Spec	**Description**
Resolution	0.3MP (640 x 480)
Frame Rate	100 FPS
Horizontal FOV	46^o
Filter Switcher	Optional
Interface	USB 2.0
No. of LEDs	26
Latency	10ms

Table 9: Specifications for the OptiTrack Cameras

5.2 Software Specification

5.2.1 Zybo Board Program: Sensor Board

For each peripheral added to the Zybo board, appropriate software needs to accommodate for the new module. By developing through the XSDK, project members can initialize, configure, and operate devices through C code and pre-defined function calls, if users are using Xilinx provided files.

One example use case for programming the Zybo board is the software required to speak with the SparkFun MPU9150 sensor board. The sensor peripheral uses I2C to communicate, so the board's software needs to accommodate for the configured hardware's internal I2C controller. A process flow for this specific example can be seen in Figure 9; however, the initial setups seen in the diagram are similar across different peripherals with the only difference being the type of controller or I/O.

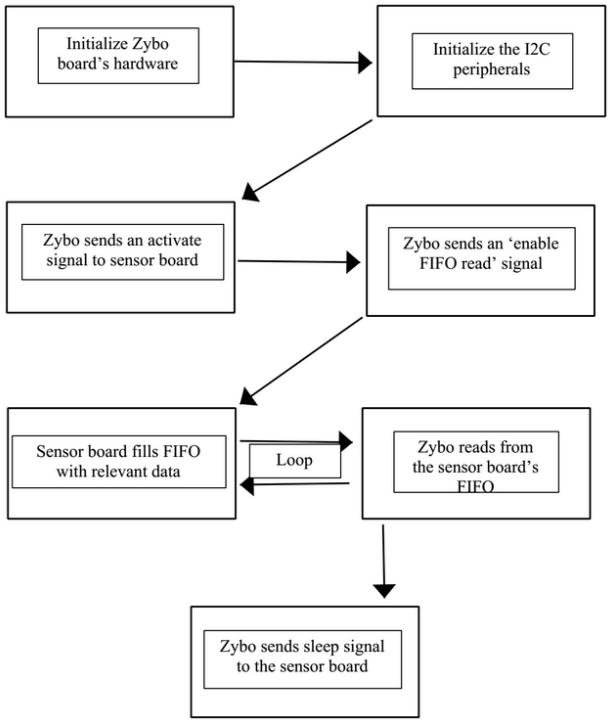

Figure 9: Program flow for reading data from the SparkFun MPU9150

5.2.2 Control Mixer

The control mixer for the system mixes the incoming PWMs from the RC receiver and translates the individual signal for a flight component to a signal that all four motors on the quad react to. This software is run by the Zybo board since it handles all communication for the quad. For example, if a users wants to move the quad horizontally to the right, the RC receiver will change the roll to rotate slightly clockwise which means the two motors on the left side of the quad need to increase power and the right ones need to decrease power.

5.2.3 Data Analytics

The data analytics for the system are done using MATLAB to parse data logs for both the entire system and individual component logs and produces visual plots of the data with respect to time. The data analytics tools are also automated, and the automated flow of the data can be seen in Figure 10.

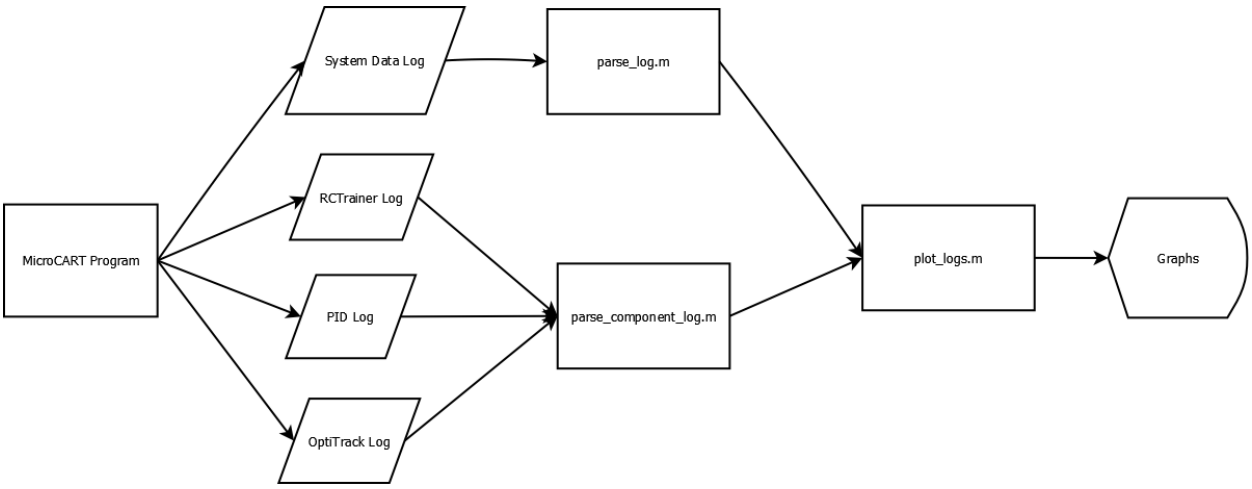

Figure 10: Data flow for automated data analytics

5.2.4 PID Controller

The PID controller is the basis for a steady flight for the quad, because it factors in the present, past, and future error of the system and uses it to correct itself rather than overshoot its ideal destination. Based off a desired location, the controller will take a weighted sum of the different errors and provide feedback to the current system. For the quad, four different PID controllers, one for each flight dynamic, are being created. The psuedo-code for a generic PID controller can be seen below:

```
while(true){
        error = desired value - measured value;
        D = (error -    previous)/dt;
        I += error*dt;
        Output = Kp*error + Ki*I + Kd*D;
        previous = error;
        sleep(dt);
}
```

5.3 User Interface Specification

5.3.1 CLI/GUI

The GUI system is used to interact with the quad using the OptiTrack camera system, the base station FPGA board, and the manual controller. The GUI starts interacting with these by first taking in camera system data. It then does a quick Kalman filter and PID calculation with the information received. Upon start of the GUI, the code implements a home system. It takes the x, y, and z coordinates that the quad is currently at and saves it as the home location. From this location, the GUI directs the quad in the direction it needs to go in, adjusting the pitch, roll, and yaw to keep it in that position until it receives the command to start moving. The full layout of the GUI can be seen in Figure 11.

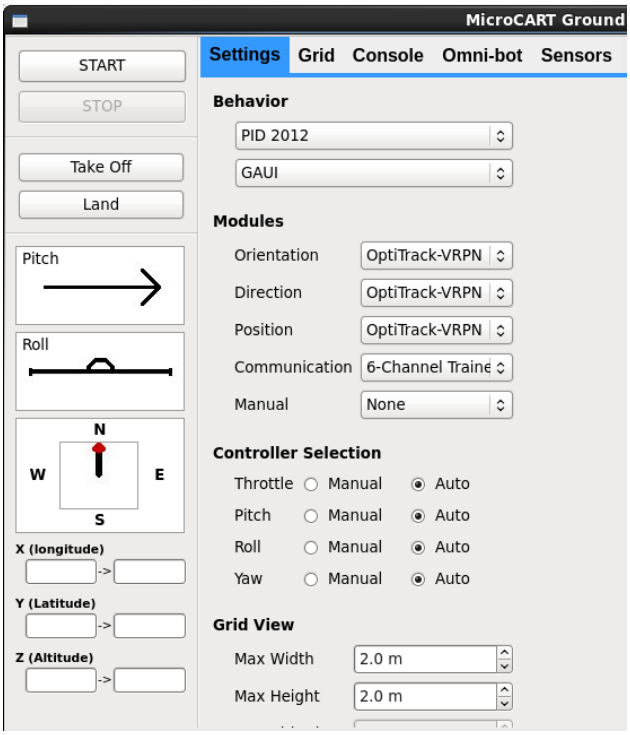

Figure 11: GUI diagram

Side Bar

Upon initially starting up the GUI, it connects to the base station FPGA board. The image on the left shows the sidebar of the GUI that is static throughout each view. The bottom of the sidebar contains the x, y, and z coordinates that the quad is at in relation to the home position. The pitch and roll graphics in the center show the orientation the quad is currently in. The compass shows the quad's current heading in relation to the camera system.

The buttons toward the top of the sidebar show the quad takeoff and landing functionality as well as the ability to start or stop the communication link. When clicked, the START button begins communications with the camera system and initializes the PPMs sent over to the quad. The STOP button stops the system from reading camera data and stops the transfer of data to the quad.

The Take Off and Land button are only functional after the START button has been pushed. The Take Off button is an auto takeoff feature for the quad to get in the air on its own. It works within the code by slowly raising the throttle PPM signal sent through the base station FPGA board. The GUI also uses camera system data to keep the quad stable as it is raising off the ground. The auto-takeoff feature is currently set to raise the quadcopter until it gets 0.5 meters off the ground.

The Land button brings the quad onto the ground from the air. The GUI uses the quads current position to keep it balanced while slowly reducing throttle. Once the quad gets to about 0.1 meters off the ground, the GUI shuts off all the motors at once so that the quad doesnt bounce upon hitting the ground.

Settings

The settings view in the GUI is pictured in Figure 3. This area is divided into the four following categories: Behavior, Modules, Controller Selection, and Grid View. The Behavior section is used to determine what system it will be running on. It currently only has two behaviors, PID 2012 and Test. The only system actually used to fly the quad is the PID 2012. The other option is the GUIs visual style, but there is only one option at the moment.

The Modules and Controller Selection determine which controller will be used to fly the quad. The default option is OptiTrackVRPN, and that is currently the only option capable of running the flight controls. Communication can be set to the trainer. The Manual control selection determines whether the quad is fully automatic, or controlled manually when manual control is selected. The Controller Selection allows you to change which specific controls will be either automatic or manual.

Grid
The Grids basic functionality is to have the user draw a flight path for the quad. The desired flight path is drawn in the white box using. Once the path is drawn, the user can click the send button to start the quad new flight path. When the user clicks the clear button, the drawn flight path is erased, allowing a new flight path to be created. The load button is used to load an existing flight path, and the save button is used to save the path to reuse later.

Console
The console is used as a textbased controller for the quad. The current supported commands are the following:

- alt - used to increase or decrease altitude
- lon - used to increase or decrease longitude
- lat - used to increase or decrease latitude

The console receives and outputs data given from the settings section in real-time displays that data. The user types in the command to give toward the bottom of the interface and clicks the send button to initiate the command. The quad measures all changes given from the console in meters.

Sensors
The Sensors area of the GUI is non-interactive. It shows where the quad is going on the left side, and where the quad is currently at on the left. The user can see the longitude, latitude, and altitude the quad is currently at and he or she can this the information for testing purposes. This information can also be used to make sure the quad is reading information correctly, ensuring the quad does not leave its designated area.

CLI
The CLI (command line interface) will be a menu driven interface that is simpler and faster than the GUI. The supported commands will allow for incrementing and decrementing the altitude, latitude, and longitude for movement, as well as incrementing and decrementing throttle, pitch, roll, yaw for testing.

5.4 Test Specification

Testing the individual components and pieces of the system will be conducted as each task is accomplished. For initial hardware and signal debugging, an oscilloscope will be used to identify issues

in the system. For example, proper I2C communication can be observed when probing the sensor board when attached to the Zybo board. For software, an agile approach will be taken by creating a small working prototype and building more functionality upon the prototype. As more functionality is introduced and problems arise, a developer can roll back the source control to a working state.

As the system becomes more elaborate, the following order of logically questions will be asked while stepping through the hardware and software flow of the system:

- Is the quad's power supply connected?
- Is the Zybo board being programmed? Is the "Done" LED on?
- Is the main program on the base station started?
- Are the proper PID constants used?
- Is the RC Trainer plugged in, set to the proper mode, and armed?
- Is the camera system software on and set for the quad?

In regards to controls, heavy testing and tuning is being done for the flight dynamics PID controllers. Testing for a single axis is done by constraining two axes by means of a turn-table or horizontal pole. By oscillating the quad back and forth, the appropriate amounts of present, past, and future error factors can be identified as the quad should stabilize and move back to a stead state if displaced. Again, this stabilization is keen on having good factors as to not overshoot or compensate too quickly.

5.5 CAD Drawings

5.5.1 Mechanical CAD

The quad frame requires several mounts for the Zybo board, peripherals, and testing equipment. Physical support pieces were created using CAD software for each case and were, more specifically, a mount reflective IR balls, a base platform for securing the quad to a testing platform, and support beams for mounting the Zybo board to the chassis; all of these pieces can be seen in Figures 12, 13, and 14.

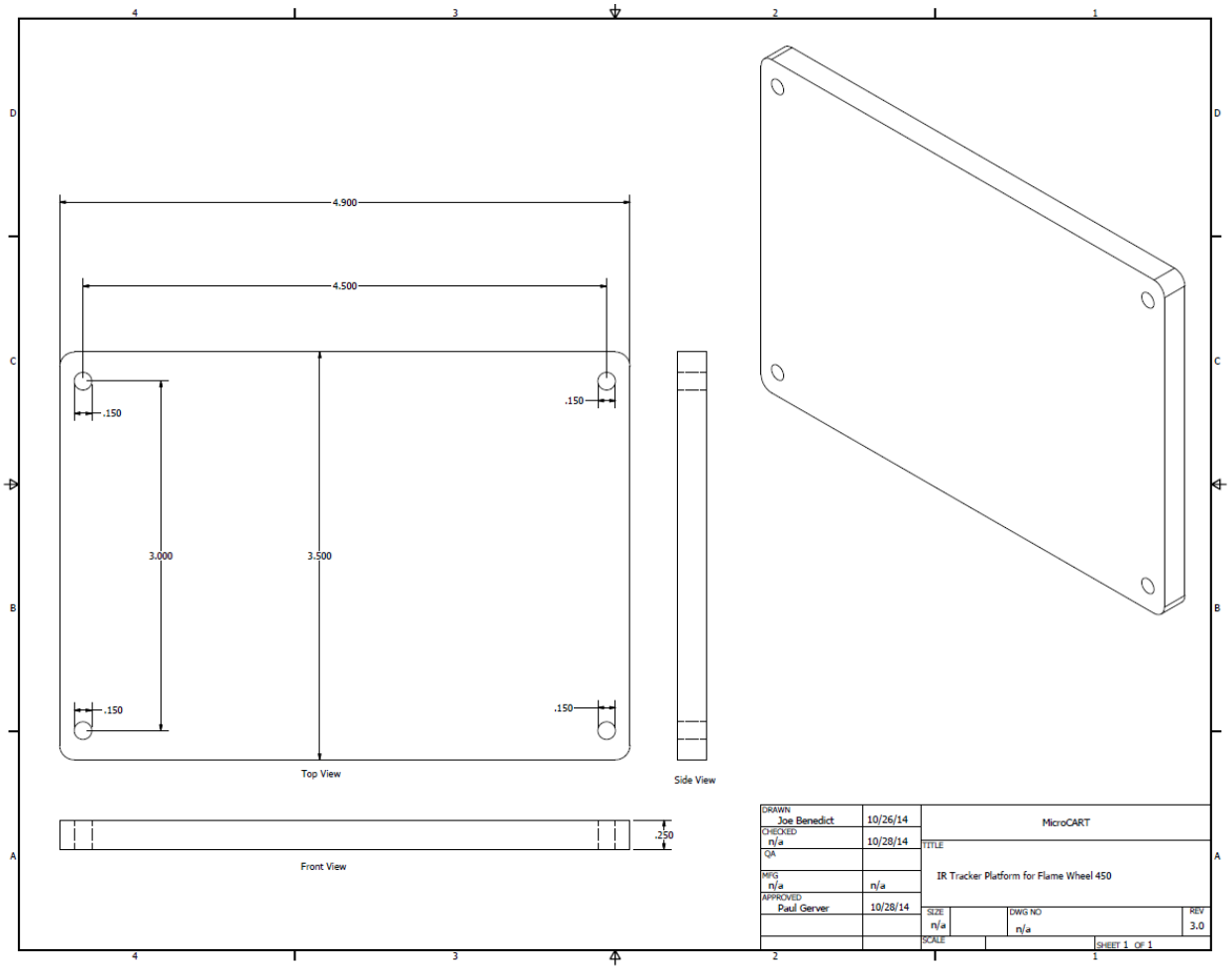

Figure 12: CAD diagram for chassis' IR mount

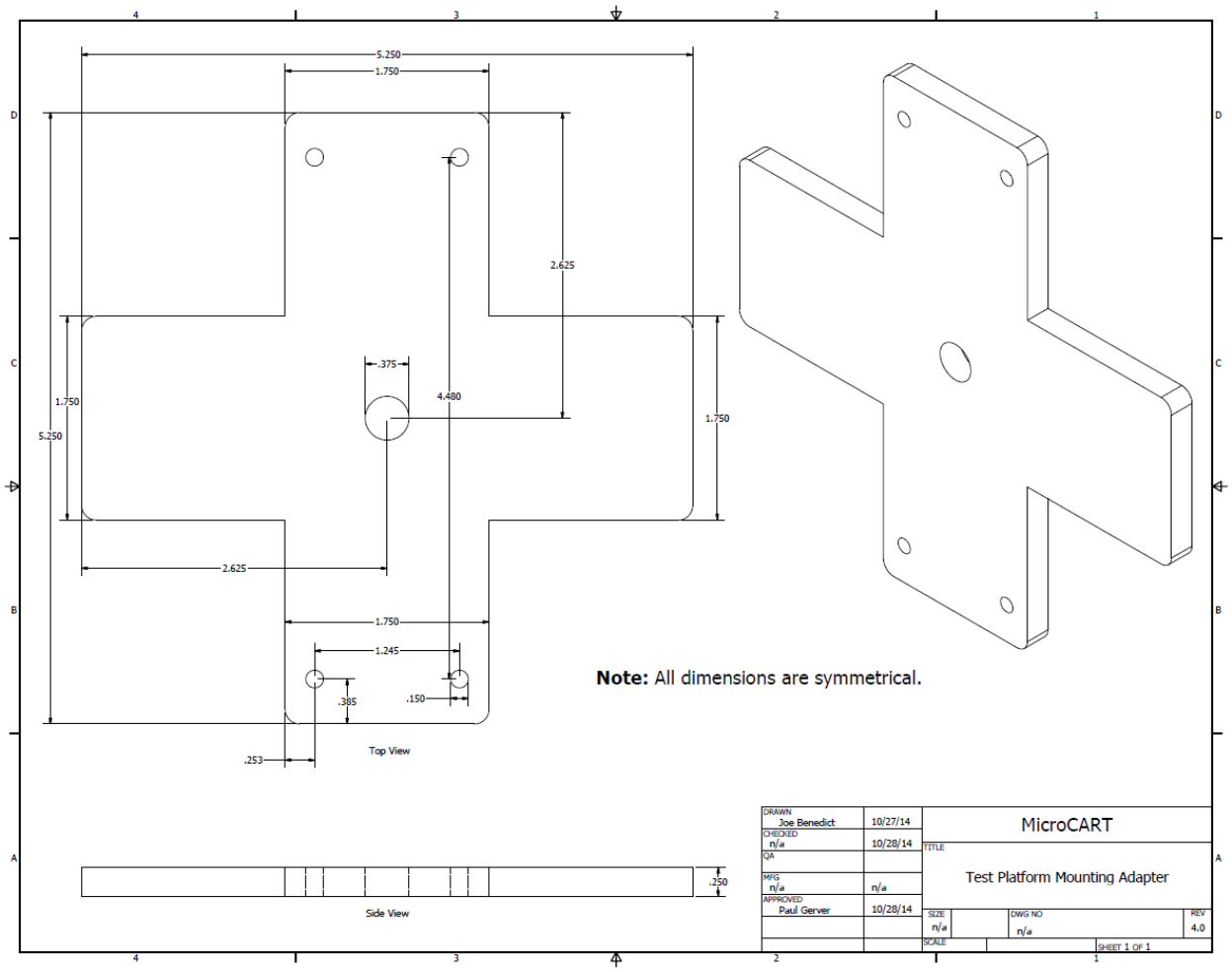

Figure 13: CAD diagram for chassis' test platform

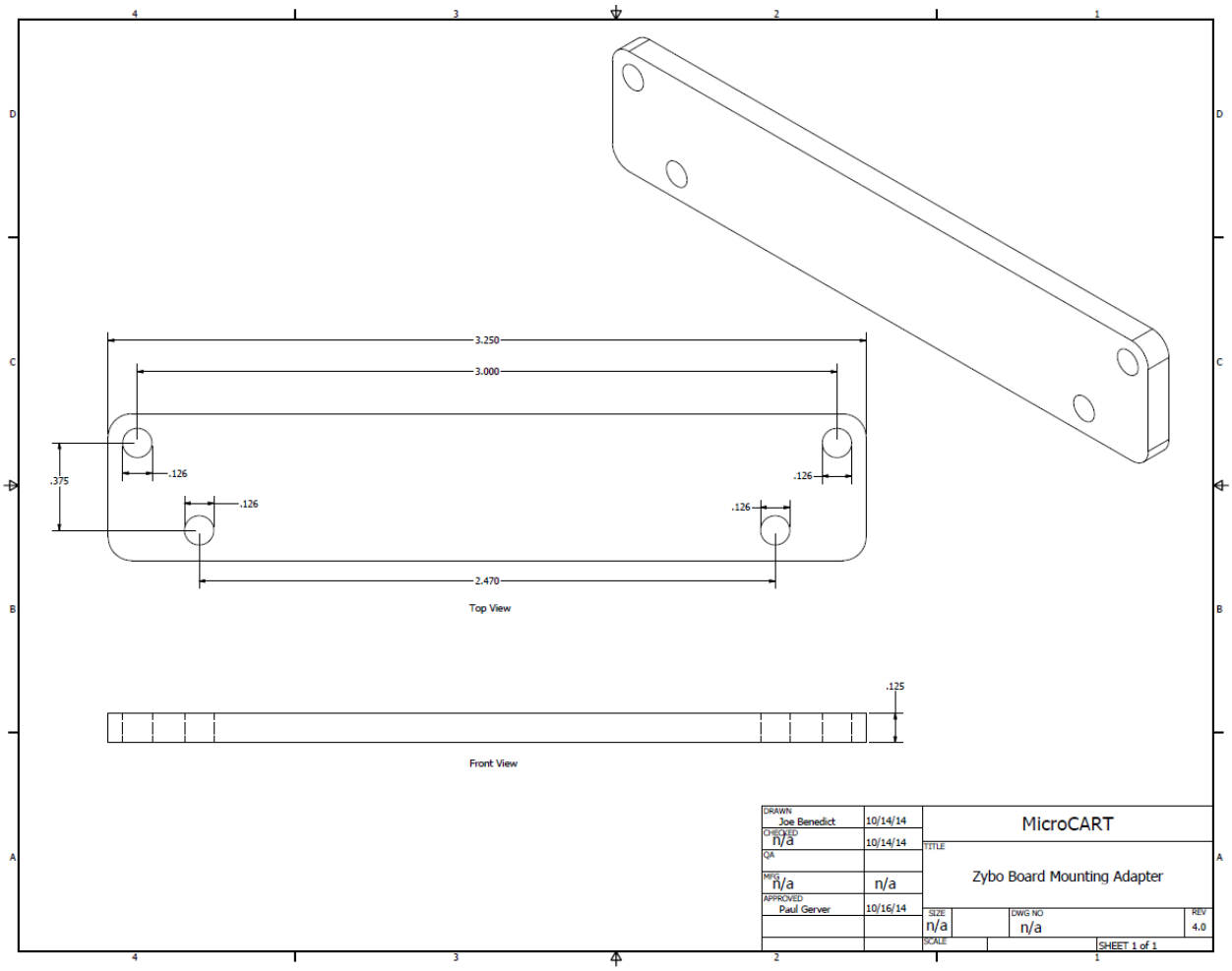

Figure 14: CAD diagram for Zybo board mounting adapter

5.6 Implementation Issues

The main issue the teams is focused on is the safety of the quad system when the batteries' power levels are too low. The team needs to ensure that all components function properly and safely which may slow or hinder progress of the building the system. Additionally, this project consists of many different components that need to be integrated well with one another, and problems can arise from different pieces and subsystems.

6 Deliverables

The team will deliver the assembled and working system to the client at the end of their second term. The final system will consist of the quad system with a mounted board, sensors, and receivers as well as the mixing, configuration, and communication software. Additionally, the team will relinquish the source code for base station, its data analytics, and documentation.

7 Closing Material

7.1 Client

Distributed Sensing and Decision Making Laboratory, Iowa State University
Contact: Dr. Philip Jones III

7.2 Team Info

Physical Hardware, Power Management, and Controls
Joe Benedict (Electrical Engineering) - joeb@iastate.edu
Ravi Nagaraju (Electrical Engineering) - nagaraju@iastate.edu

Peripheral
Tyler Kurtz (Electrical Engineering) - tkurtz@iastate.edu
Jacob Rigdon (Computer Engineering) - jrigdon@iastate.edu
Matt Vitale (Computer Engineering) - mvitale@iastate.edu

Software and Data Analytics
Adam Campbell (Software Engineering) - awjc@iastate.edu
Paul Gerver (Computer Engineering) - pfgerver@iastate.edu

Advisor
Dr. Philip Jones III (Assistant Professor, Dept. of ECpE)