Fast, Low-Cost Swarm Robots

Team 17 – Michael Bartmess, Paul Ernst, Peter Cork

Spring 2018 - ECE 445: Design Document

TA: Dongwei Shi

1. Introduction:

1.1 Objective:

Getting numerous robots to work together allows for various novel applications. The multiple robots can be used in parallel to complete multiple tasks at the same time. Or they can come together to complete a large task that would be too daunting for only one. In addition, there is something inherently satisfying about seeing physical objects move in sync. Unfortunately, while a robot tends to be very good at knowing its own state, it often lacks the environmental awareness required to coordinate its movements with others. Current methods of introducing this environmental awareness have been to add more sensors to the robot, but in doing so the price per robot increases.

Our goal is to design and build a robot that is low-cost by reducing the number of sensors required to identify the position of the robot. In addition, we want to allow high precision and but also high responsiveness of the robot. Our method for doing this is to create a base-station that utilizes advances in machine vision to report the position of the swarm robots. In doing so we can thoroughly simplify the swarm robots.

1.2 Background:

The most promising swarm robot interface currently is the Zooid project at Stanford[1]. This project is intended to use the swarm robots as a sort of physical display, where the robots act as physical pixels. The software is very impressive and provides a good demonstration of the uses of swarm robots. However, the price per robot and the base-station price is very high. Each robot costs $50 when not mass produced, and the setup requires a high-speed projector that costs roughly $600, plus the price of the computer and radio station. Our goal would be to create a system that dramatically reduces the price of the setup and of the individual setup.

1.3 High Level Requirements:

- The robots should be able to move faster than .25m/s and be accurate to 0.5cm, and move independently and concurrently, and complete the routines shown in section 2.5
- The robots should have a footprint smaller than $35cm^2$
- The price per robots should be under $35. The price for the base station should be under $150

2. Design:

2.1 Overview and System Block Diagram:

The design will consist of two main functional parts: a vision system, and the swarm robots.

The vision system will consist of a webcam mounted overhead and will be connected to a computer for processing the image data. After computing the position and orientation of each robot, the computer will use an attached router to transmit that information, as well as the desired position and orientation, to the correct robot as UDP packets.

The swarm robots will consist of an SoC with an embedded Wi-Fi stack to handle the connection, and two motors capable of propelling the robot quickly around the table. A USB-to-Serial IC will allow us to program the SoC as well as allow for USB charging of the batteries which will supply power.

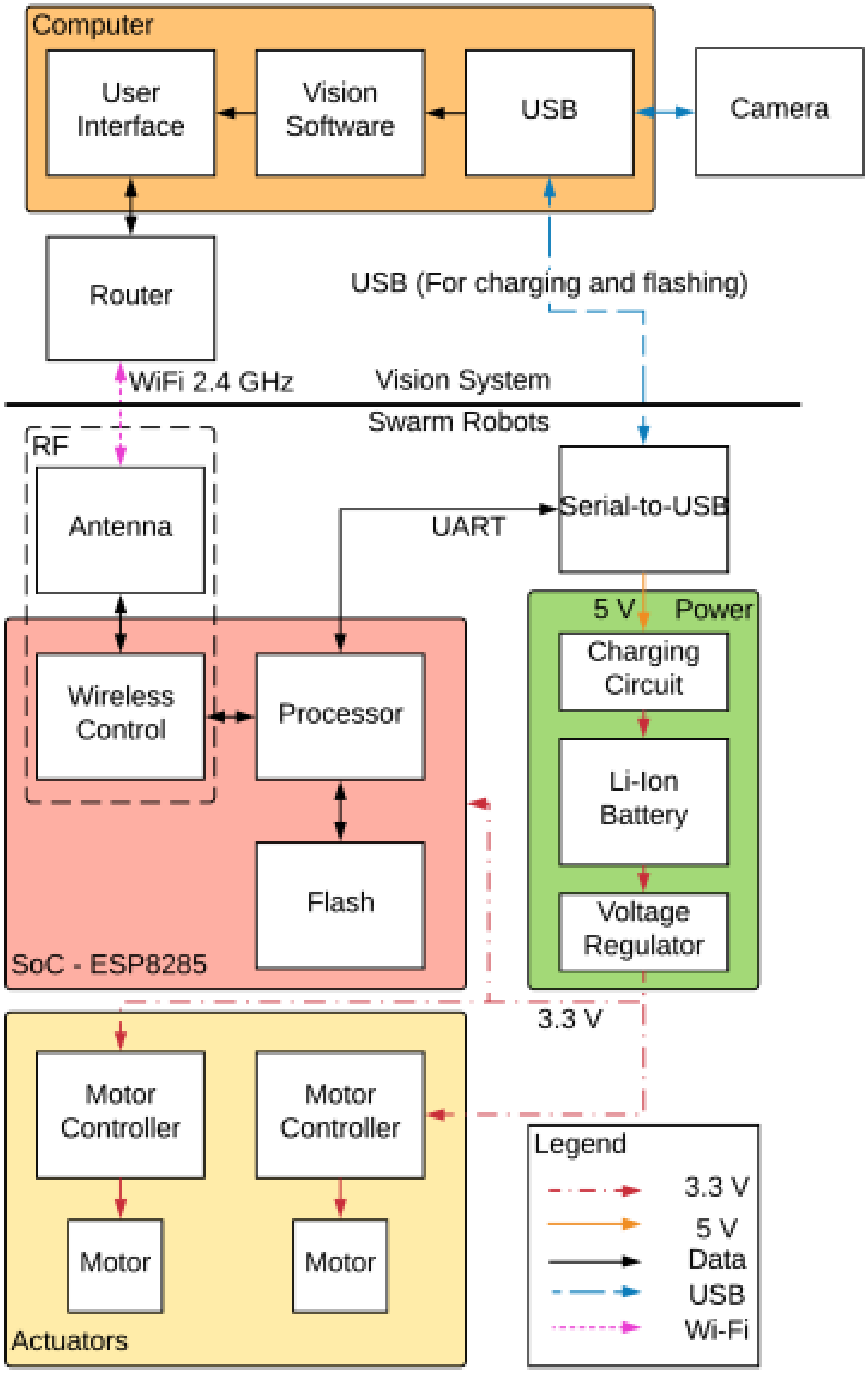


Figure 1: Block diagram for vision system and swarm robot

2.2 Physical Design:

The design of the robots will consist of a rectangular base with 2 wheels on the outside for motion. There will also be some variety of caster wheels to provide stability to the robots. The PCB and battery will ideally be located above the motors, and the top of the robot will have a vision target to assist with the machine vision.

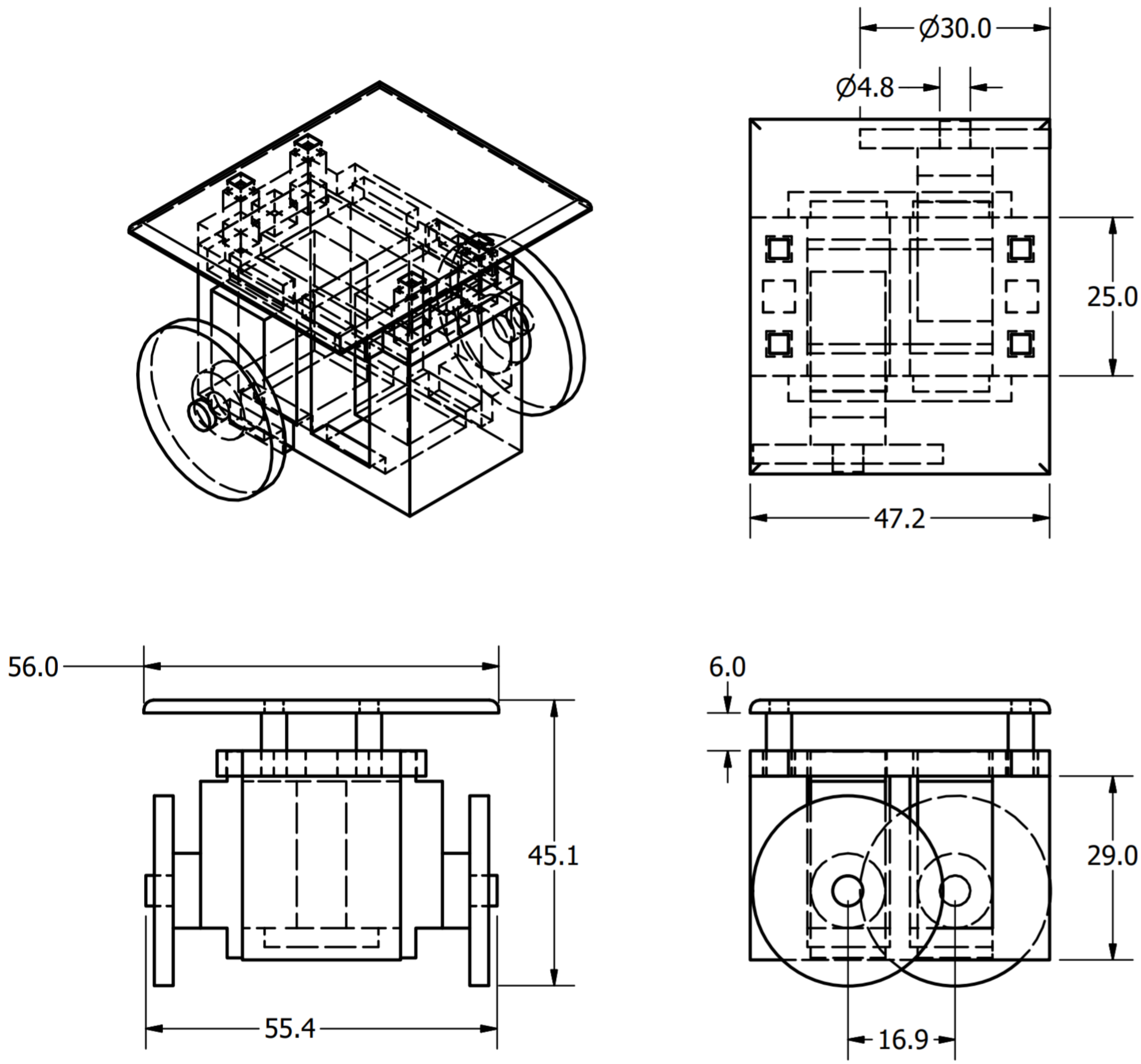


Figure 2: Preliminary physical robot design

2.3 Block Design:

2.3.1 Webcam:

2.3.1.1 Functional Overview:

The vision system will have a webcam mounted over a table. This will allow us to see the table with the robots on it and will connect to the computer via USB. This will be an off the shelf component that lies within our cost restrictions for the vision system. It will receive power over the USB connection.

2.3.1.2 Requirements and Verification:

Requirement:	Verification:
Be 1080p in order to provide enough resolution	A. Using a computer, connect to the webcam and open a stream B. Capture a still as an image, inspect the image metadata to verify the resolution
Be able to stream greater than 10fps	A. Using a computer, connect to the webcam and open a stream B. Use a software counter to verify the fps of the stream

2.3.2 Computer:

2.3.2.1 Functional Overview:

This will be any sort of computer capable of doing image processing. The images received from the webcam will be analyzed and the position and orientation of each robot calculated. The computer will also handle determining the correct destination for each robot. The information for both will be sent to the router. Power will be provided from the wall, and ideally the machine will be running a linux flavor for easy programming.

2.3.2.2 Requirements and Verification:

Requirement:	Verification:
Be fast enough to analyze a frame in under 100ms	A. Using images captured from the webcam, run the process

	software in a loop for 2000 iterations B. Verify the time to run the program takes less than 200s

2.3.3 Router:

2.3.3.1 Functional Overview:

This will be a Wi-Fi router capable of transmitting a 2.4GHz signal to the robots. This will also be an off the shelf component. The router will be communicating to the robots with a UDP connection to allow for quick and frequent connections. It will connect to the computer using an Ethernet cable

2.3.3.2 Requirements and Verification:

Requirement:	Verification:
Transmit and receive a 2.4 GHz Signal	A. In router settings, disable the 5Ghz band. B. Verify a computer can still connect
Connect to 16 devices at one time	A. Connect to all 16 robots at once B. Verify communication by sending each a packet
Transmit 16 packets of 16kb in under 100ms	A. Connect to all 16 robots at once. B. Send 32000 packets, 2000 to each robot C. Verify that the packets were sent in under 200 seconds

2.3.4 System-on-a-Chip:

2.3.4.1 Functional Overview:

The most complicated block, the SoC will consist of an ESP8285, which contains a processor, embedded flash memory, a full Wi-Fi stack, and various GPIO interfaces. This will be the brain of the robot and will execute the control loop to move the robot to the destination position received from the vision system computer. It must be fast enough to handle the incoming position data as well as calculate the proper signals to send to the motors. It will receive power from the power block at 3.3v, and will be programmed from the USB-to-Serial block. It will use PWM signals to control the motors.

2.3.4.2 Requirements and Verification:

Requirement:	Verification:
Be fast enough to respond to compute and respond to loop corrections in under 100ms	A. Run loop correction routine 2000 times, then flash an LED B. Verify the loop completed in under 200 seconds
Be able to connect to a 2.4GHz signal	A. Connect to the router, verify the connection is made
Can transmit 4 PWM signals simultaneously	A. Connect the 4 channels of an oscilloscope to the 4 PWM output pins B. Run a 1kHz PWM wave for all 4 at the same time C. Verify the PWM signals are visible on all 4 channels
Runs reliably between 2.95v and 3.4v	A. Connect a voltage generator to the SoC's VDD pins B. Set the voltage to 3.3v C. Start communicating with the router, logging packets missed. D. Sweep the voltage between 2.95v and 3.4v, and verify that the packet loss rate does not change
Consumes less than 200mA while transmitting	A. Connect a voltage generator to the SoC's VDD pins B. Set the voltage to 3.3v C. Start communicating with the router D. Check the current draw from the voltage generator, verify less than 200mA

2.3.5 RF Block:

2.3.5.1 Functional Overview:

This block overlaps with the SoC block and contains an antenna for transmission and receiving of the 2.4GHz Wi-Fi signals. The overlap occurs since the SoC contains the remainder of the Wi-Fi stack. Due to the short ranges we will be using, the antenna does not need to be extremely robust.

2.3.5.2 Requirements and Verification:

Requirement:	Verification:
Capable of receiving packets with less than 10% loss at 6ft away	A. Connect SoC to router B. Place robot 6ft away from router antenna C. Send 2000 packets from router to robot D. Verify that the SoC received at least 1800 packets

2.3.6 Power Block:

2.3.6.1 Functional Overview:

The power block provides power to the components on the robot. It will store the energy in a Lithium Ion battery. Charging will come from a micro-USB plug on the robot and will be carefully controlled through the charging circuit to prevent over and undercharging. The power will then flow through a voltage regulator to the various components of the board.

2.3.6.2 Requirements and Verification:

Requirement:	Verification:
Changing Circuit: Able to charge the battery in less than 1 hour	A. Connect the battery to the charging circuit B. Connect the charging circuit to USB power C. Monitor the status pin on the charging circuit with a voltmeter. Verify that it is charged within 60 minutes
Voltage Regulator: Able to maintain a voltage output between 2.95v and 3.4v over an input range of 3.5v to 4.2v	A. Connect input on Voltage Regulator to a voltage generator B. Sweep voltage from 3.5v to 4.2v C. Verify voltage on output stays between 2.95v and 3.4v
Voltage Regulator: Able to provide 500mA of steady state current	A. Connect input of voltage regulator to a voltage generator at 3.3v.

	B. Use an E-Load to pull 500mA output. C. Verify that the voltage at the output pin is still above 2.95v using multimeter
Battery: Able to provide 500mA for 30 minutes	A. Using a fully charged battery, connect an E-Load B. Pull 500mA for 30 minutes C. Verify that the output voltage from the battery is above 3.7v using multimeter D. Verify battery temperature is below 35C using IR sensor or thermometer
Battery: Output voltage stays between 3.5v and 4.2v	A. Charge battery B. Verify that the voltage never rises above 4.2v while charging C. Let battery discharge at 350mA D. Verify that the voltage never falls below 3.5v

2.3.6.3 Supplementary Material:

The following circuits (pulled from Adafruit's Feather Huzzah schematics[2]) demonstrate a possible configuration for the battery charging circuit and the voltage regulator, although we will likely use a different regulator IC.

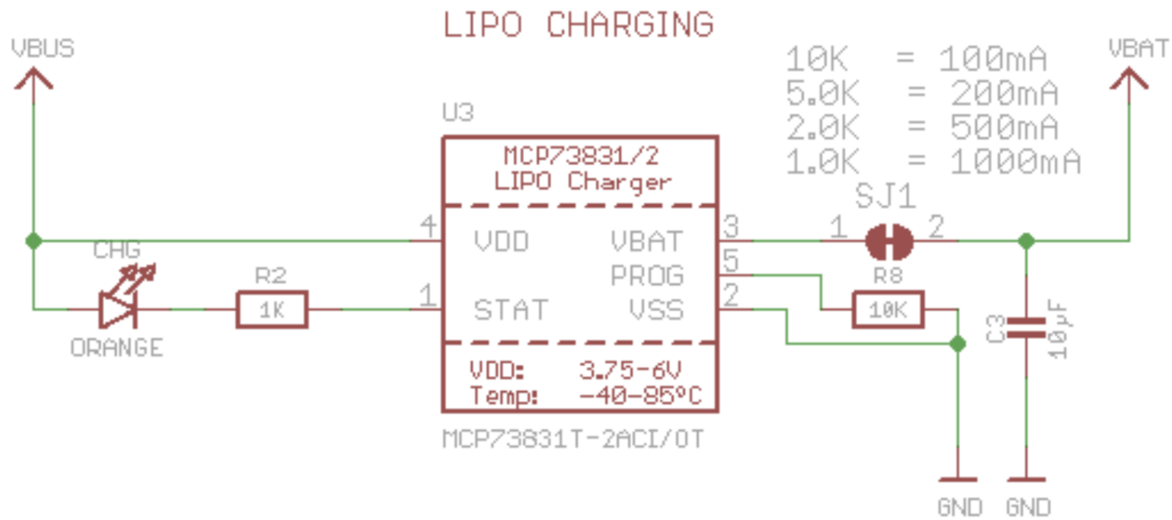


Figure 3: Li-ion battery charging schematic

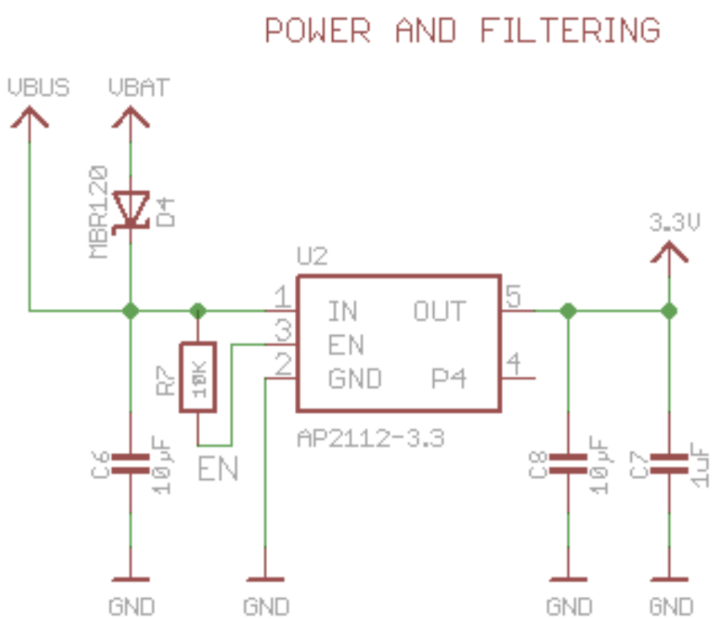


Figure 4: 3.3V LDO regulator schematic

2.3.7 Motors:

2.3.7.1 Functional Overview:

The motor blocks will each consist of a motor controller IC and a DC motor. The controller IC will receive PWM signals from the SoC and use those to drive PWM signals to the motors.

2.3.7.2 Requirements and Verification:

Requirement:	Verification:
DC Motor: Draw less than 150mA with no load at 4v	A. Connect the motor to a voltage generator at 4v B. Verify the current draw is less than 150mA
Motor Controller: Operate from 3.7v to 4.2v	A. Connect the motor controller inputs to a voltage generator and the outputs to a motor B. Sweep the input voltage from 3.7v to 4.2v C. Verify that the motor continues to spin, note any changes in speed.
Motor Controller: Be able to run a DC motor bidirectionally	A. Connect the motor controller to the SoC and select both forward and backwards motion. B. Verify the wheel can turn both ways

2.3.8 USB-to-Serial:

2.3.8.1 Functional Overview:

The USB-to-Serial block will allow for the SoC to be programed from a computer's USB port. It will handle the conversion of the USB packets into UART serial data that will be flashed into memory. In addition this block will set a few pins so that the SoC will be in programmable mode. The power for this chip will come from the 5v USB, but the UART signals will be at 3.3v

2.3.8.2 Requirements and Verification:

Requirement:	Verification:
UART and GPIO signals are at 3.3v	A. Connect the UART and GPIO outputs to an oscilloscope B. Verify the signals are at 3.3v
Has 2 USB-controllable GPIO pins	
UART communicates with baud rate of 9600bps	A. Monitor the UART Signals B. Verify communication happens at 9600bps

2.4 Tolerance Analysis:

For our project one of the most critical components is insuring a robust communication link between the information provided from the web camera and the robots making up the swarm. As rapid calculations are necessary for each robot to know its position and carry out its next task, there is no leeway for data to be lost in transmission. To insure that nothing is lost we needed to have good impedance matching with the antenna. An omnidirectional antenna is selected because in our application it is unknown what the robots position will be relative to the router over time.

We selected the 802.11g protocols because it has a maximum data rate of 54Mbps. The actual realistic data rate is 24Mbps and for our project we have a necessary data rate of 20MBps so it is within the bounds.

For the SoC we are using the sensitivity is able to detect -75 dBm signals at 54Mbps. For selecting an antenna we need a well-matched interface so that power was not lost when going from the chip to the antenna for radiation or coming in from the antenna for demodulation. We used the transmission line approximation for reflectance from interfaces to determine the amount of power lost from the antenna to the SoC or the SoC to antenna.

Eq 1. $$|\Gamma|^2 = \left|\frac{Z_L - Z_O}{Z_L + Z_O}\right|^2$$

The output impedance for the esp8285 is $39 + j6\Omega$. And the transmitting power for the SoC is +17 dBm for the 802.11g protocols. We want to minimize the loss so that the Signal to noise ratio is high. A high SNR means that our wireless system is more robust.

We determined the actual amount of power transmitted successfully through:

Eq 2. $$10log_{10}\left(1 - \left|\frac{Z_L - Z_O}{Z_L + Z_O}\right|^2\right)$$

If there is too much loss of power due to the interface the router and the SoC will not be able to detect the packets sent. We determined the maximum amount of signal loss due to the interface to be high as our router will be close to the robots and they should never stray far. The Routers transmitting power is

18.98 dBm so the maximum loss acceptable would be 93.98 dBm for router to robot. A small amount is lost in propagation through free space so there was a large leeway for our selection of the impedance of the antenna. This is important because for any antenna we select the cost is multiplied by the amount of robots in our system.

Power lost due to free space propagation in dB is:

Eq 3. $$10log_{10}FSPL = 10log_{10}(\frac{4\pi d}{\lambda})^2$$

Where D is the distance propagated and the wavelength is the wavelength of light at the operation frequency. For our distance we operate at 2m at most. So about 46 dBm is lost to propagation. The remaining 48 dBm can be lost to the interface. Our matching portion has a tolerance of a 48 dBm loss. This range narrows as the distance we wish to use our devices at increases.

We are using the UDP/IP protocol so there is no need to lower the data rate due to error checking or sending packets multiple times.

2.5 Software:

2.5.1 Vision Software:

Our main control loop is determining the commands our robots' need and sends those commands. The process for doing this begins by obtaining an image from our camera. With the image, we transform it into a black and white image so that an algorithm called Oriented FAST and Rotated BRIEF (ORB)[3] can find our vision targets more easily. ORB produces a list of corners (with position and angle information). After finding the corners, we look through the list of corners and try to find combinations of corners that produce rectangles the size of our vision targets. With those rectangles, we can determine the position and angle of all the rectangles. With the positions of the rectangles, we can now read the ID information stored in our vision targets in the original color image. Then, the position of the robots is compared with our desired positions and calculate desired changes. Once we have our commands we send the specific command to each specific robot using UDP. If the robot has not been found with vision for the past 500 ms, we send a command for it to remain stationary.

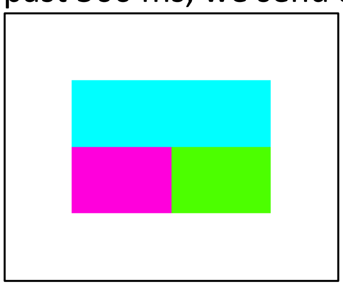


Figure 5: Sample Vision Target

Some of the steps of our main control loop depends on some values that need to be determined through testing. This process consists of us placing a robot into our camera area and taking an image of it. From here, we determine the brightness of the vision target. After determining the threshold for turning the vision target white and the background black, we try to find a single rectangle using ORB. From this rectangle we can find the expected size of a rectangle for use in our main control loop.

2.5.2 Robot Software:

Our robots will all have two main control loops: one for managing router connection and sending status to the control computer and one for managing a state machine for receiving control data and controlling motors. The control state machine will have 5 states; 3 of which set the motors to control either straight, turning, or off; and the other two states dedicated to handling waiting for and reading packets. If there is a timeout, the robot should automatically go to the stationary state and begin waiting for a new command again. The connection and status loop will simply check every 500 ms to check for router connection, if not connected try to activate the Wi-Fi controller to connect to the router. After either finding an existing connection or creating a new connection, the robot sends a status update to the control computer describing the current state that the control loop is in.

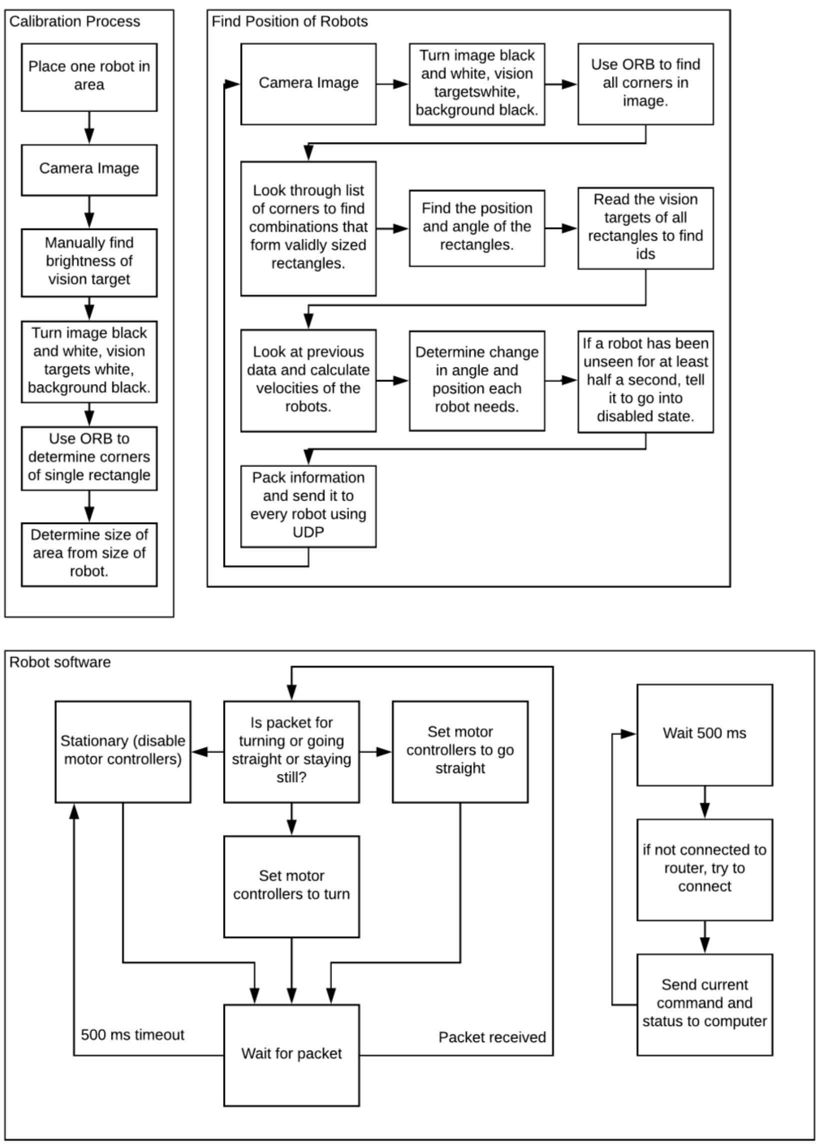


Figure 6: Software Flowcharts for calibration, vision analysis, and robot control

The UDP packets sent back and forth will each have 24 bytes of payload being sent (Layouts of each in Figure 7). Control packets being sent to the robots will contain their current location, their desired location, and the time we want them to be there by. The status packets sent from the robots to the computer contain the robots estimated position and the robot's current command that it is following. The packets contain a CRC-16 checksum to verify that they contain valid information. We will determine the exact conversion factors between real location to the location in the packets when we start testing. The integer sizes are chosen such that we should have more than enough resolution to describe the locations (at least 0.1 mm and 0.04 degrees).

Control Diagram (In bytes)

1	2	3	4
"CTRL"			
version	robot_id	x	
y		current_rotation	
start_rotation		final_x	
final_y		final_rotation	
final_time		crc	

Status Diagram (In bytes)

1	2	3	4
"STAT"			
version	robot_id	est_x	
est_y		est_rotation	
start_rotation		final_x	
final_y		final_rotation	
final_time		crc	

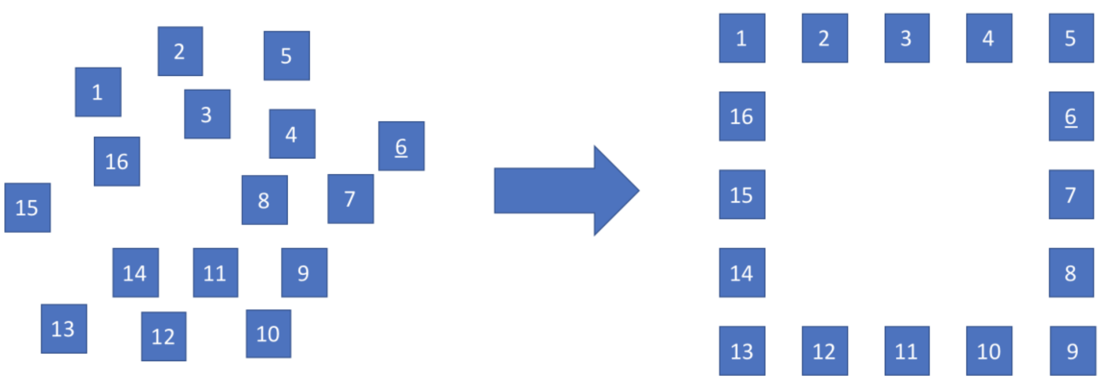
Figure 7: Packet Layout

2.5.3 Robot Routines:

The following routines should be able to be completed by the robots.

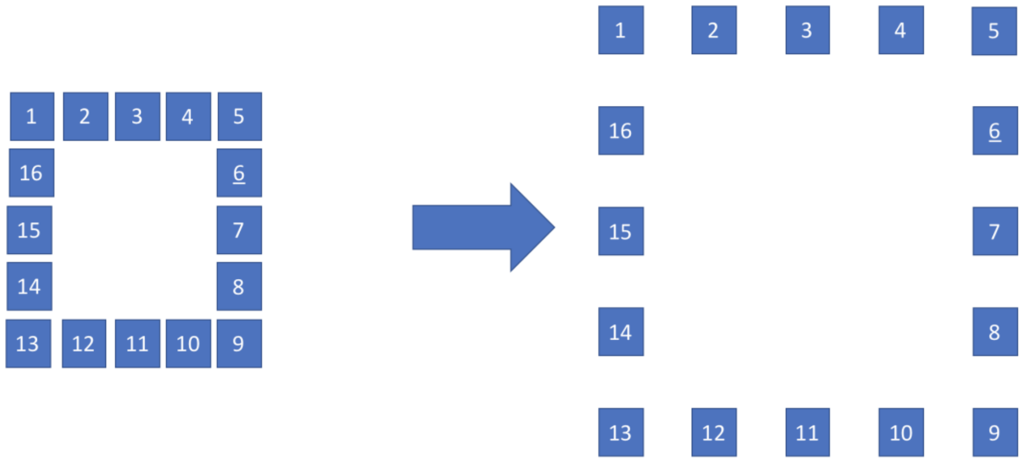
A) Move to static formation:

The robots should form an ordered box from an initially scattered arrangement



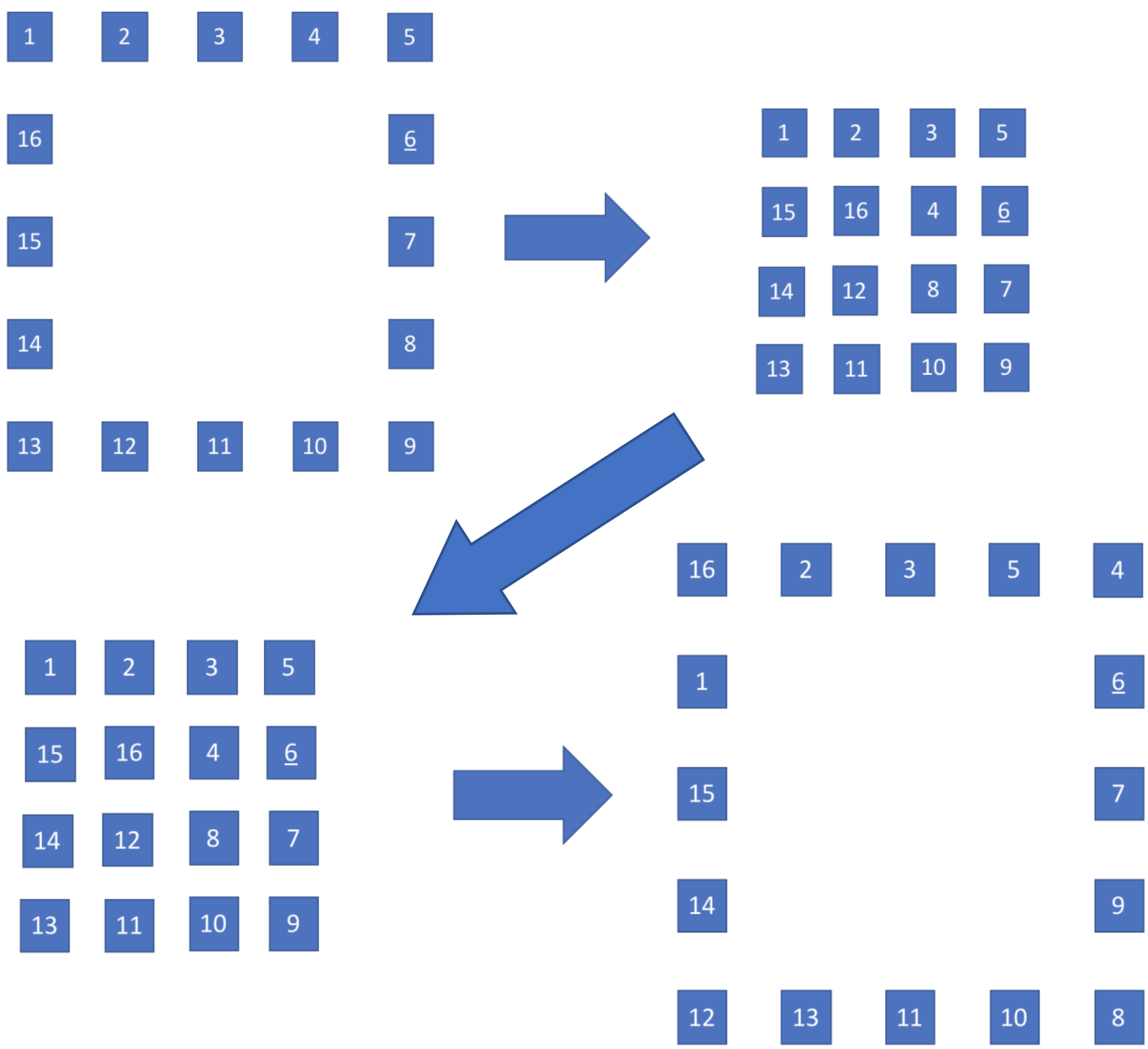
B) Expanding Formations:

The robots should move concurrently and smoothly to give the impression that the box is expanding. Poor control loops will cause this to look bad.



C) Kaleidoscopic Formation:

The robots move in a high-effect motion, collapsing and expanding in on themselves giving the effect of rotation. This requires coordination and timing as well as accurate positioning.



3. Cost and Schedule:

3.1 Cost Analysis:

3.1.1 Labor:

We estimate the time to build each robot will come to 8 man hours, and since we are building 16 that is a total of 128 hours assembly time. We estimate that the design time is roughly 6 hours per week each for 3 people, so for a 16 week project. Assuming a 2.5x multiplier and a $30/hr salary for engineers, the labor cost of our project comes out to **$31,000**.

3.1.2 Parts:

The following table defines the part costs per robot:

Block – Part (Manufacturer):	Cost (Single):	Cost (Bulk):
SoC – ESP8285 (Espressif)	$3.38	$1.46
Power – MCP73831 (Microchip)	$0.59	$0.44
Power – Li-Ion Battery 500mAh (Adafruit)	$7.95	$7.16
Power – LDL1117S33R (STMicroelectronics)	$0.41	$0.11
USB-to-Serial – CP2104 (Silicon Labs)	$1.65	$1.53
Motor – DC Motor x2 (Adafruit)	$7.00	$5.60
Motor – DRV8834 (Texas Instruments)	$2.76	$1.26
Other – Assorted Passives (Digikey)	$5.00	$2.50
PCB – PCB (PCBway)	$2.20	$0.59
Shell – Shell (Self)	$0.50	$0.25
Total:	$31.44	$20.90

The following table defines the part costs for the vision system, excluding the computer:

Block – Part:	Cost (Single):
Camera – Logitech C920	$51.70
Router – Linksys EA6350	$89.99
Total:	$141.99

The total parts cost to build 16 robots and 1 vision system (assuming the computer is provided and the robots parts are not at bulk prices) would be **$645.03**

3.1.3 Total Cost:
The total estimated cost for the parts and labor is: **$31,645.03**

3.2 Schedule:
The following schedule is proposed for the remaining weeks before the demo:

Week of:	General Task:	Mike:	Paul:	Peter:
2/26	Prototype motion	Develop firmware for ESP8285 for running motors	Work on vision software, complete 16 target classification	Put together motor testbench
3/5	Complete PCB Design	Work on schematics and layout	Complete 16 target classification on real image	Work on schematics and layout
3/12	Complete RF tests, buffer week	Work on firmware for communication from router to robot	Complete position computations for vision targets	Put together crude test robot, no PCB
3/19	Spring Break	None	None	None
3/26	Assemble 1st PCB	Assemble PCB,	Complete vision system software	Assemble robot
4/2	Single Unit, Full test	Test SoC communications	Test vision software accuracy	Test power and rf blocks
4/9	Assemble all PCB's	Assemble remaining PCBs	Make software more robust	Assemble remaining PCBs
4/16	Demo presentation preperation	Prepare for demo, emphasis on higher level	Prepare for demo, emphasis on software	Prepare for demo, emphasis on hardware
4/23	Demo	Demo	Demo	Demo

4. Safety and Ethics:

There are multiple safety considerations for our swarm robotics. The most important of which, is the lithium ion battery in each robot. Lithium Ion batteries have been known to, and can, fail. Their failure can cause injury in the event of an explosion when the batteries cell and discharge rate create a positive feedback loop leading to overheating.

To prevent this situation from occurring we plan to design a charge protection element. The protection element separates the battery from the circuit charging elements. It monitors the temperature of the power source insuring that there is no runaway heating. The other precaution is to prefabricated batteries from a trustable source. Each robots charging circuitry will be checked to insure that the proper voltage is drawn from each battery. We will not make any modifications to the prefabricated batteries to uphold our responsibilities to section 9 of the IEEE code of ethics[4] – to prevent injuries to others. Our design will comply with the International Technical Commission standards for ion battery safety specifically 62133 for portable electronics.

A second concern is that in our project there is a web cam. Web cams are easily tampered with and provide the possibility for someone unwanted to spy on your property or person. We shall implement a method to automatically turn off the web cam when robots are no longer detected. As in section 1 of the IEEE code of ethics our design should concern the welfare of the safety of the public, specifically the users of our design. Privacy is a right that should be enjoyed. Warnings will be provided for situations in which our electronics should not be used. For damaged components of our devices such as those caused by an element falling off the desk, we suggest discontinuing the specific robots use. The same is true of any parts of our project damaged by water. Other guidelines for proper usage will be established so that our project is enjoyed safely and ethically. Our robots are not designed at a size where children could consume them and provide a choking hazard. All parts of our project are not large enough to cause a threatening physical force to humans.

To properly adhere to minimizing conflicts of interest in our project as per Section 2 and 3 of the IEEE code of ethics, we will be specifically careful in our citations as to avoid plagiarism. Our project was inspired by the zooids project done at Stanford. All parts of our project will make sure to provide a citation when they are derived or closely related to their work. Special care will be made to clearly state what is our own design and work.

5. References:

[1] Mathieu Le Goc, Lawrence Kim, Ali Parsaei, Jean-Daniel Fekete, Pierre Dragicevic, et al.. Zooids: Building Blocks for Swarm User Interfaces. Proceedings of the Symposium on User Interface Software and Technology (UIST), Oct 2016, New York, NY, United States. pp.97 - 109, 2016, .

[2] Ada, L. (2016, March 19). Adafruit_product_schem.png [PNG]. New York City: Adafruit Industries. ESP8266 Feather Schematic

[3] OpenCV: ORB (Oriented FAST and Rotated BRIEF). (n.d.). Retrieved February 23, 2018, from https://docs.opencv.org/3.4.0/d1/d89/tutorial_py_orb.html

[4] Institute of Electrical and Electronics Engineers, Inc.. (2006). Code of Ethics IEEE, http://www.ieee.org/. Retrieved at July 28, 2009