PDR

TEAM C: Column Robotics

November 3, 2015

TEAM C: Column Robotics

Job Bedford

Cole Gulino

Erik Sjoberg

Rohan Thakker

PROJECT DESCRIPTION

Project Description

Autonomous Exploration and Docking

Failure of undersea oil and gas infrastructure has resulted in billions of dollars of damages.

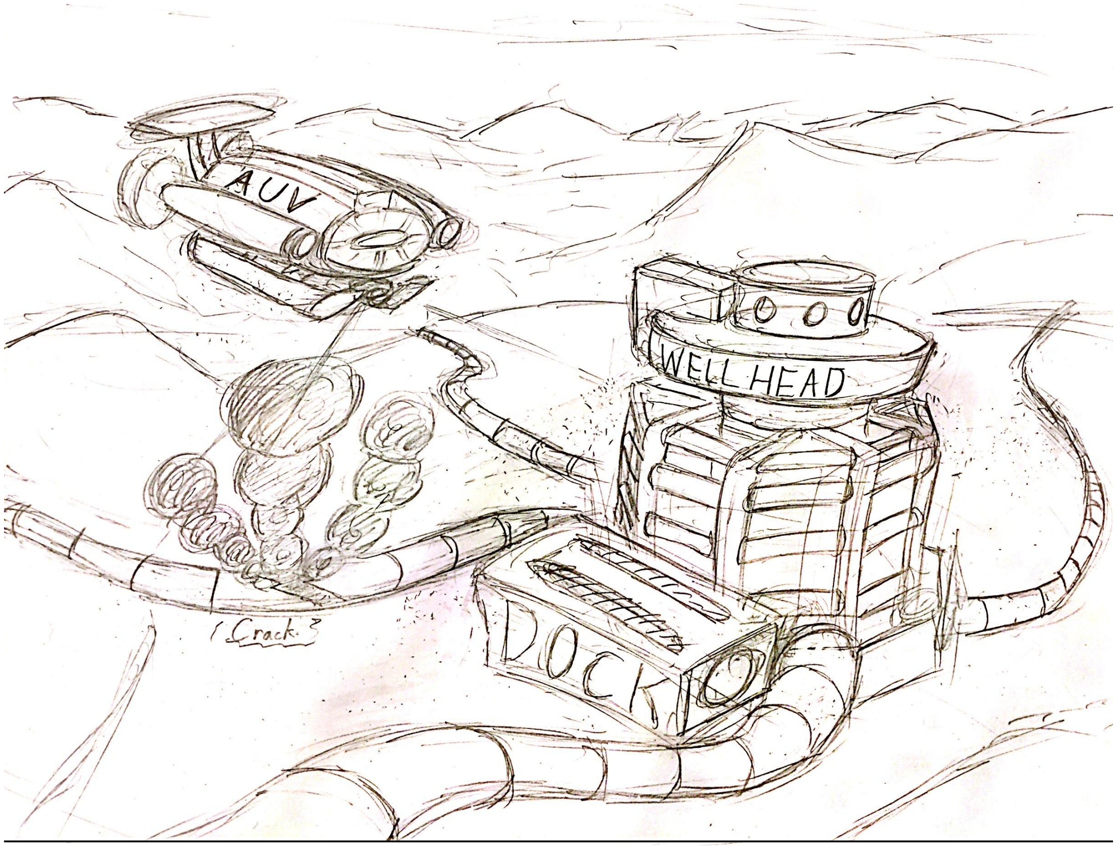
Current maintenance of these wellheads often requires a specialized ship and manual ROV crew, costings hundreds of thousands of dollars per intervention.

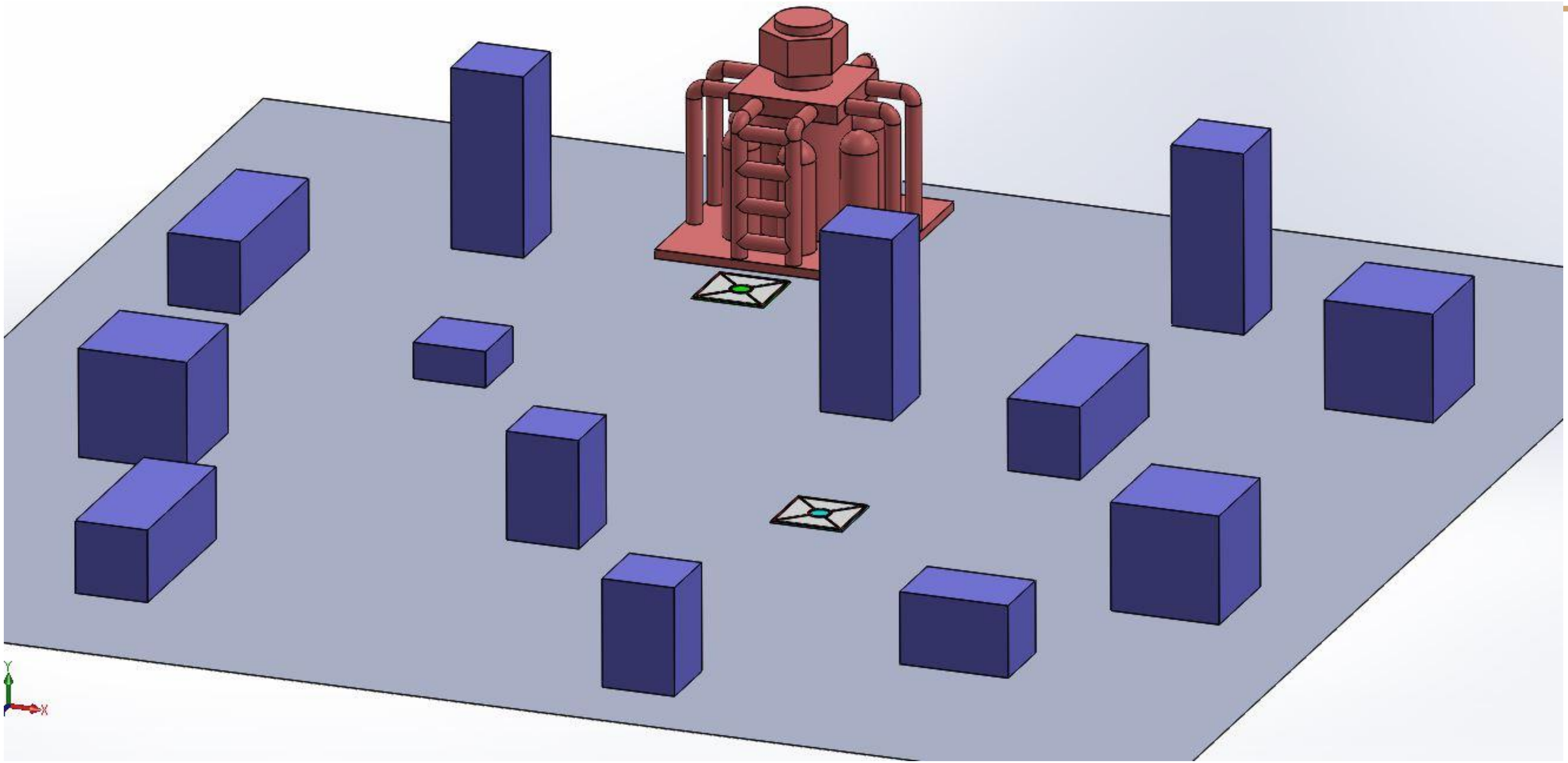
We propose to demonstrate **a terrestrial analogue** to an underwater vehicle capable of autonomously searching for, identifying, and docking with underseas wellhead.

http://persistentautonomy.com/

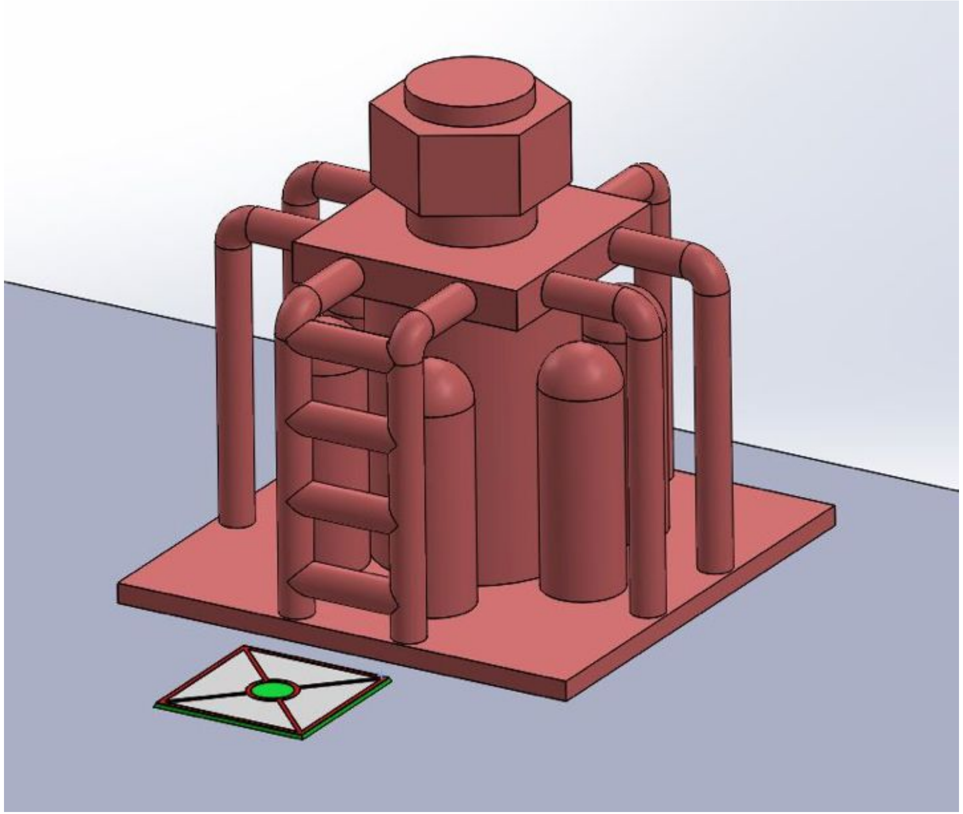
http://3drobotics.com

USE CASE

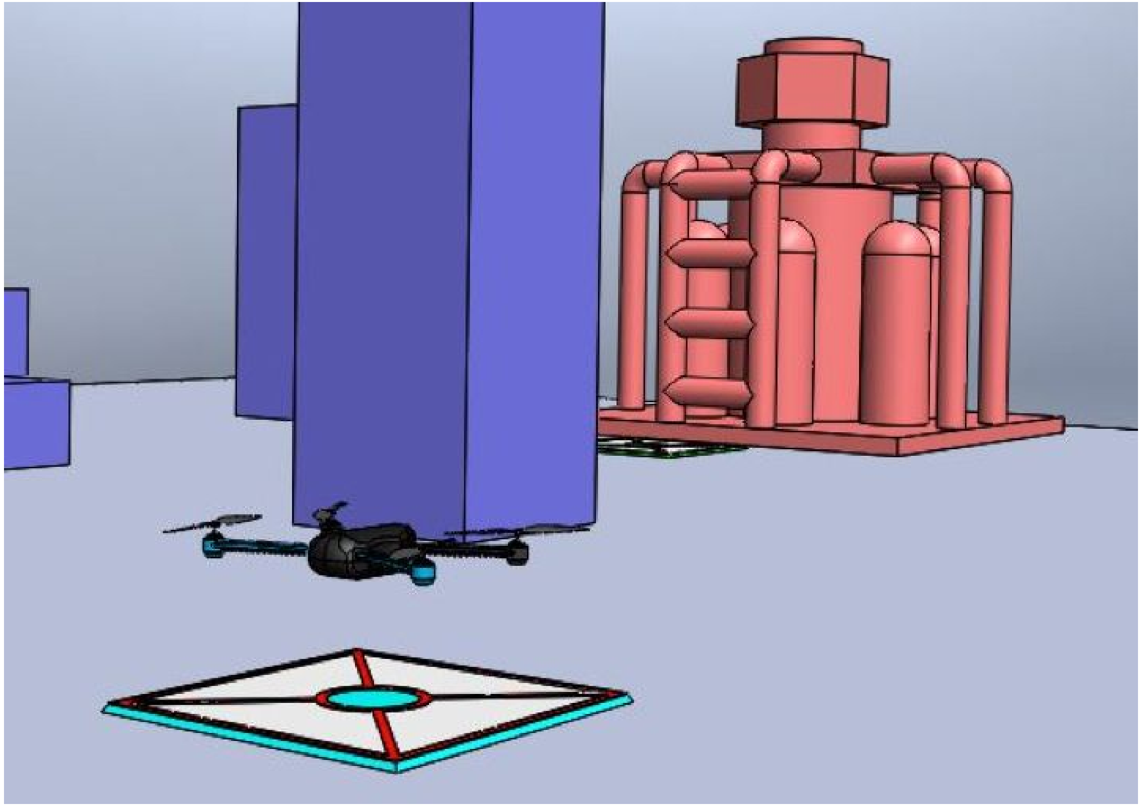

AUV
WELL HEAD
DOCK
Crack



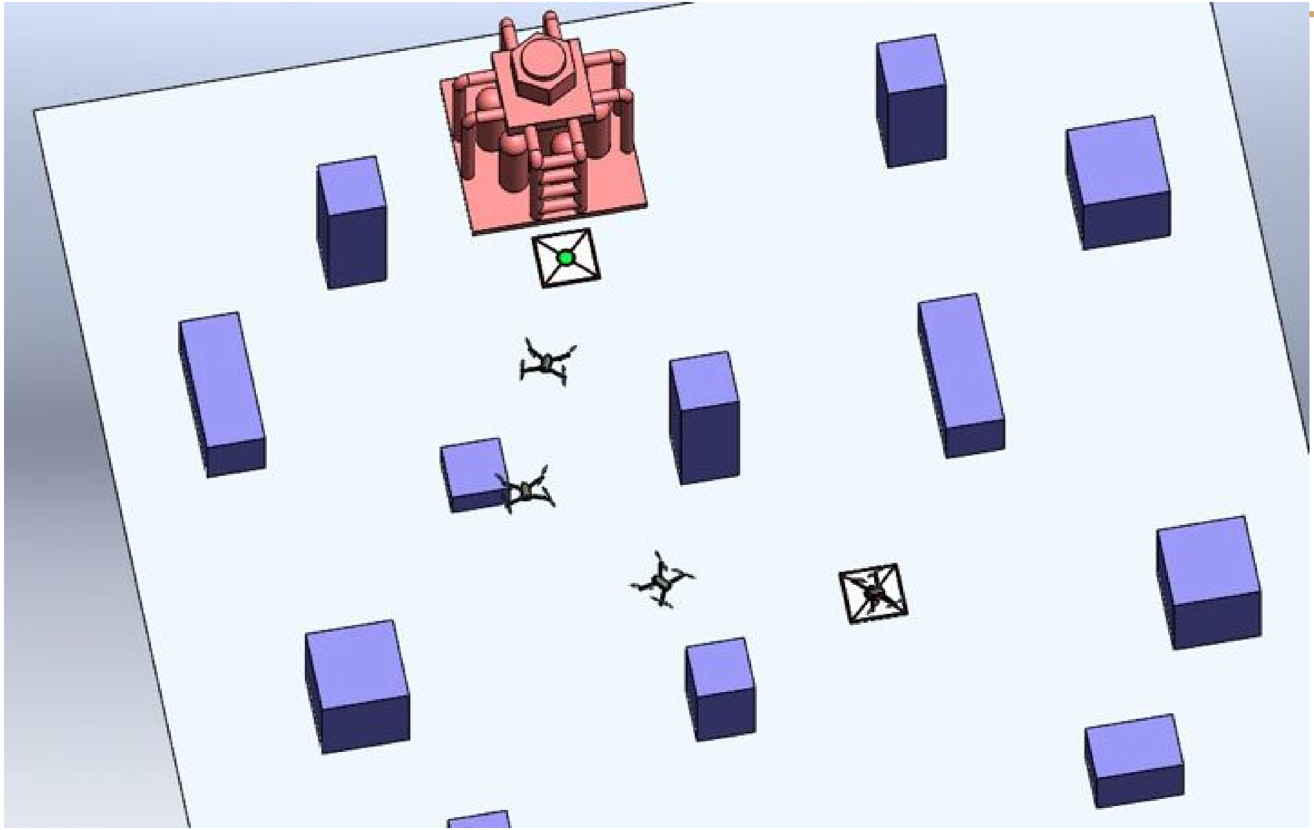
Y
X



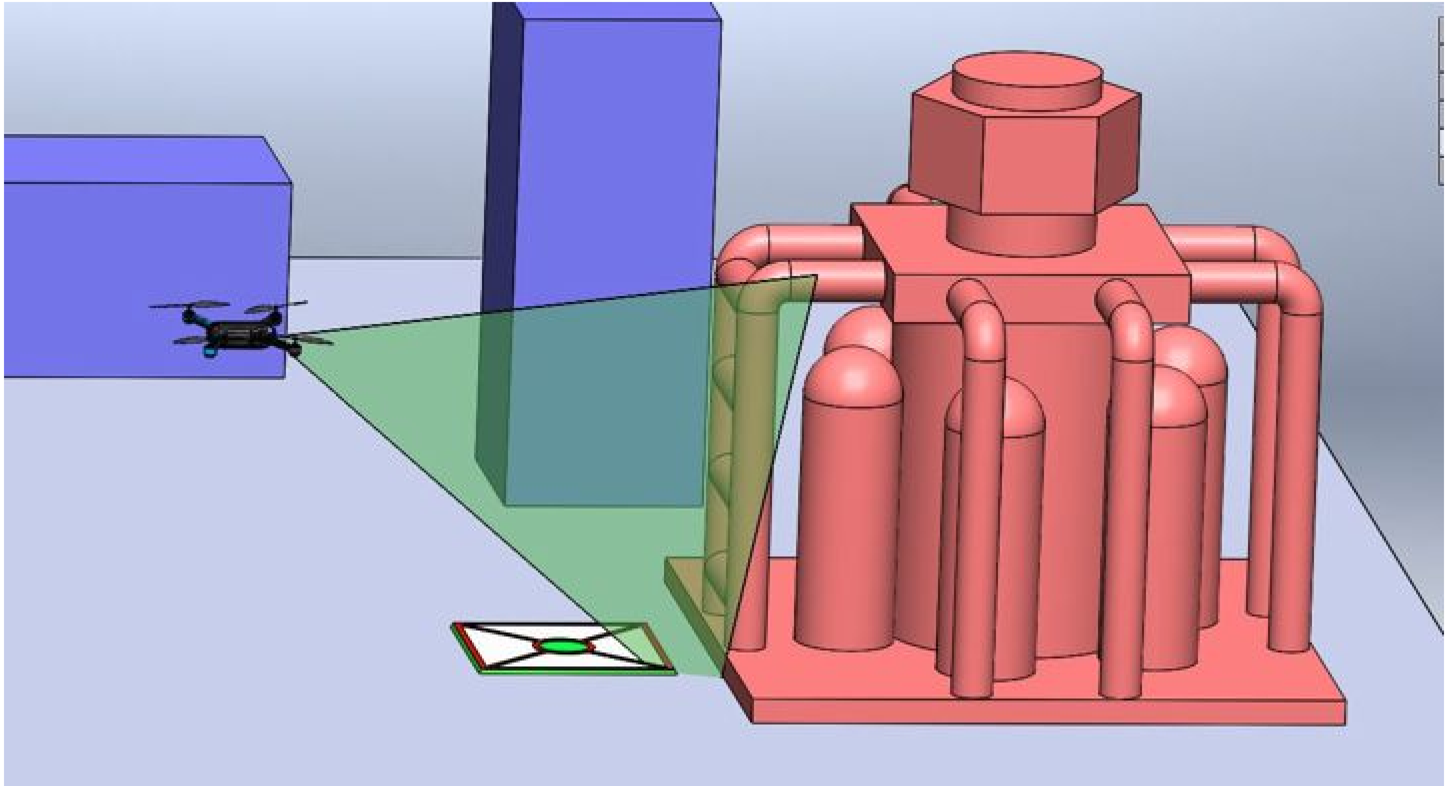
Wellhead



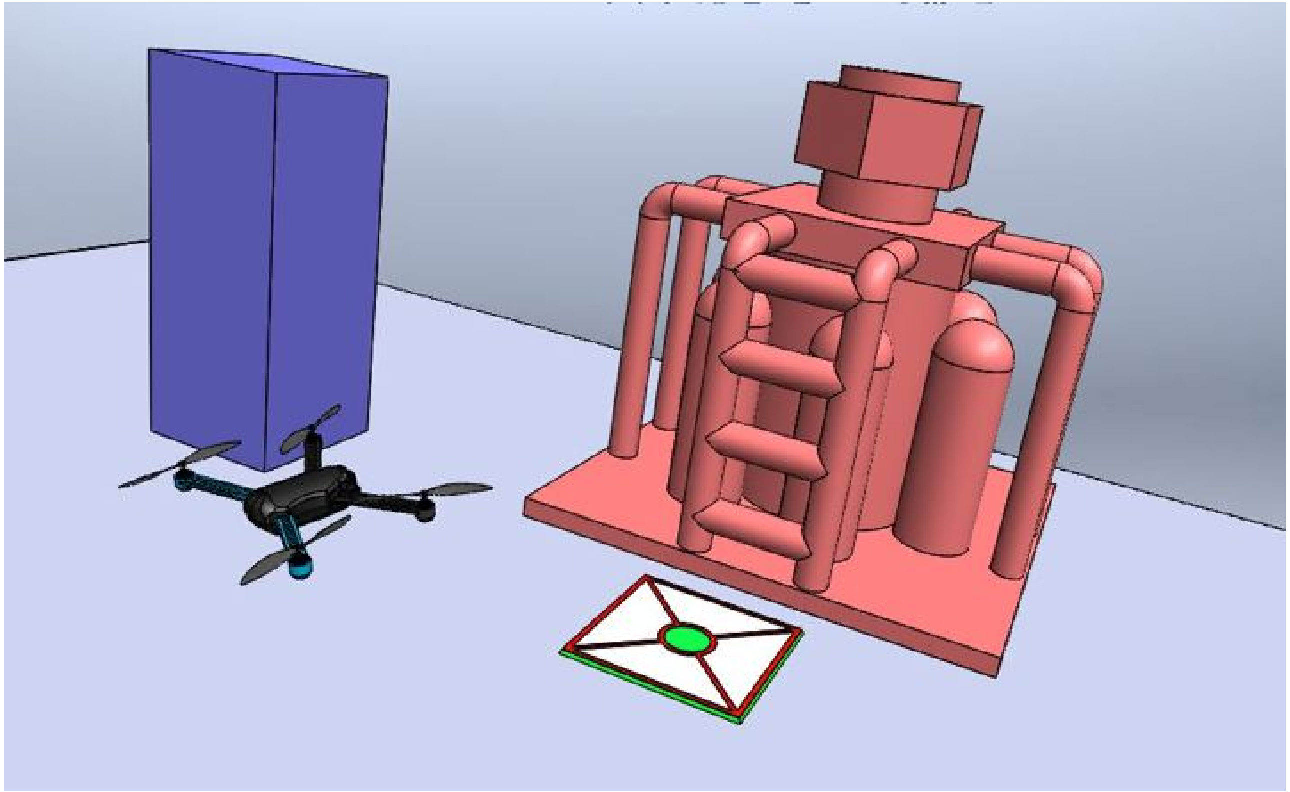
Liftoff from initial Landing Pad



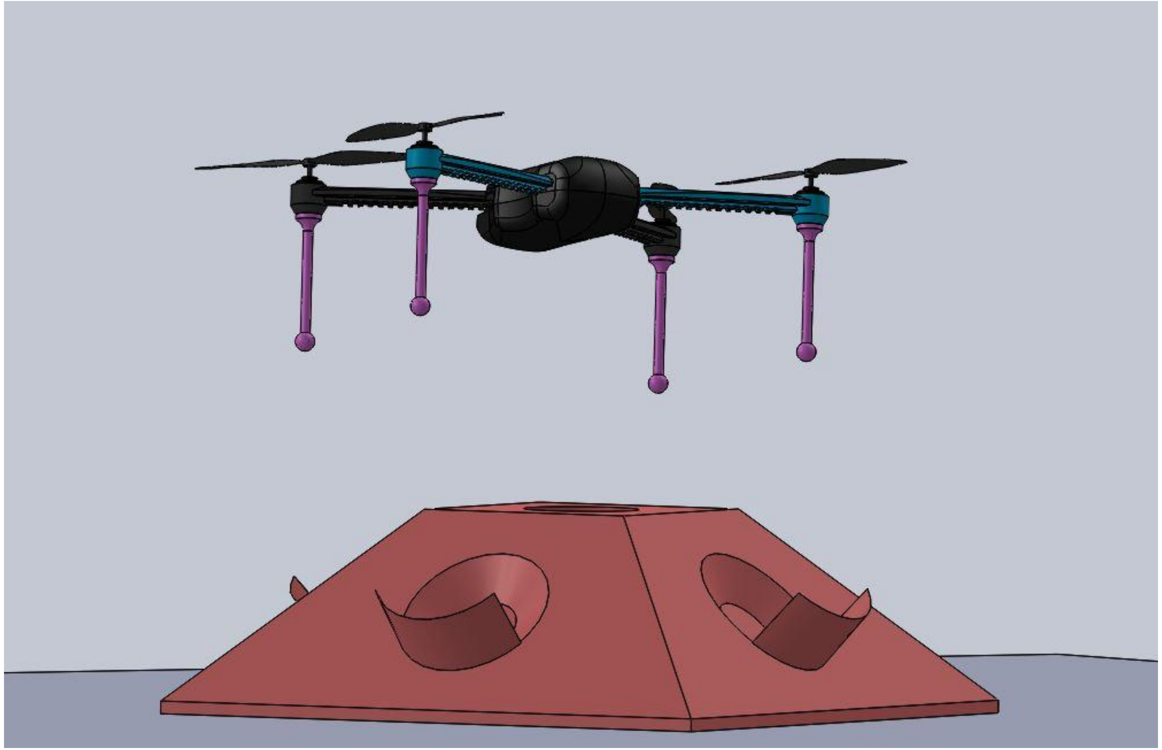
Searching for Wellhead



Identify Wellhead



Orientation before Docking



Docking

SYSTEM-LEVEL REQUIREMENTS

MANDATORY FUNCTIONAL REQUIREMENTS

MF1. Locate Oil/Gas wellhead infrastructure with known heading in 50 m^2 area

- Changed: Area shrunk due to testing constraints

MF2. Autonomously maneuver to wellhead within 1 hour

- Changed: Performance metric deemed to be more valuable

MF3. Positively ID as correct wellhead with 90% confidence

MF4. Obtain visual lock on dock with downward facing camera

MF5. Rigidly dock in 5 DOF

- Changed: 5 DOF more relevant to Quadcopters

MF6. Provide status feedback to user of current state at 0.1 Hz

DESIRED FUNCTIONAL REQUIREMENTS

DF1. Locate oil/gas wellhead infrastructure in low visibility with unknown heading

DF2. Positively ID as correct wellhead from visual object recognition

DF3. Align with dock located behind wellhead

DF4. Rigidly dock in 6 DOF with electrical connection

MANDATORY NON-FUNCTIONAL REQUIREMENTS

MNF1. Operable with simple graphical user interface

MNF2. Provides emergency stop for system

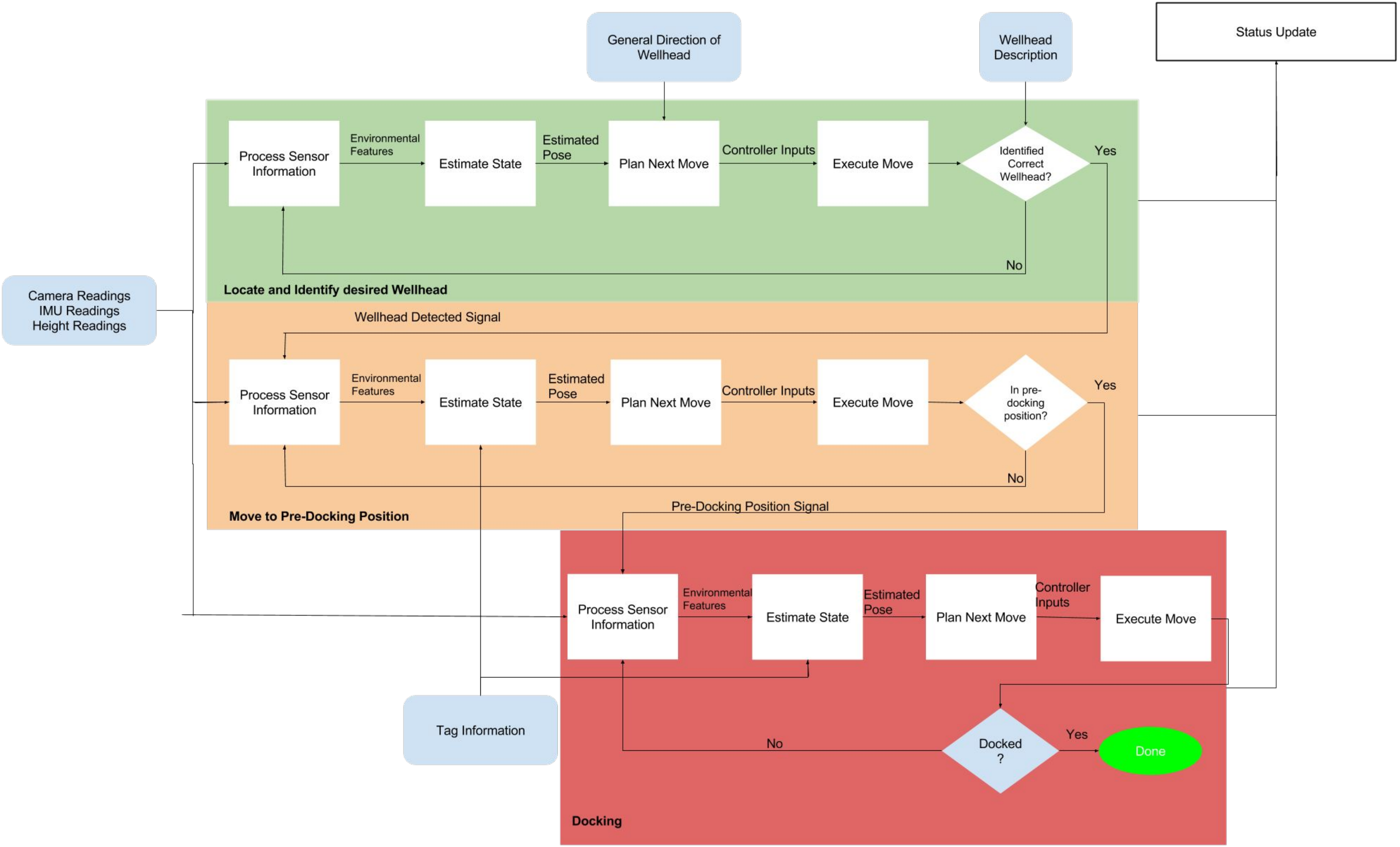
MNF3. Operable by a single person

DESIRED NON-FUNCTIONAL REQUIREMENTS

DNF1. Reduce cost of wellhead intervention

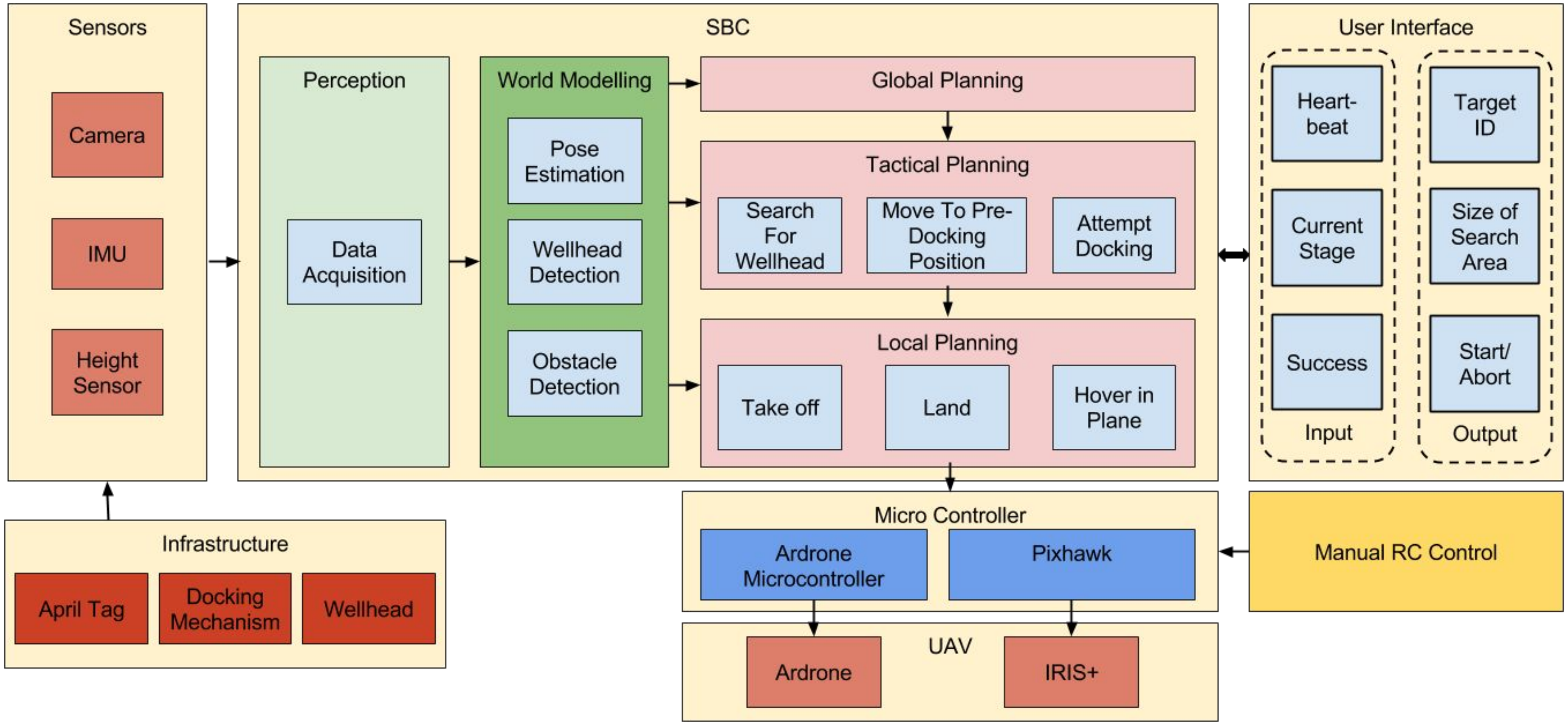
DNF2. Simulate underwater environment

FUNCTIONAL ARCHITECTURE



General Direction of Wellhead
Wellhead Description
Status Update
Process Sensor Information
Environmental Features
Estimate State
Estimated Pose
Plan Next Move
Controller Inputs
Execute Move
Identified Correct Wellhead?
Yes
No
Locate and Identify desired Wellhead
Camera Readings
IMU Readings
Height Readings
Wellhead Detected Signal
Process Sensor Information
Environmental Features
Estimate State
Estimated Pose
Plan Next Move
Controller Inputs
Execute Move
In pre-docking position?
Yes
No
Move to Pre-Docking Position
Pre-Docking Position Signal
Process Sensor Information
Environmental Features
Estimate State
Estimated Pose
Plan Next Move
Controller Inputs
Execute Move
Tag Information
Docked ?
Yes
No
Done
Docking

CYBERPHYSICAL ARCHITECTURE

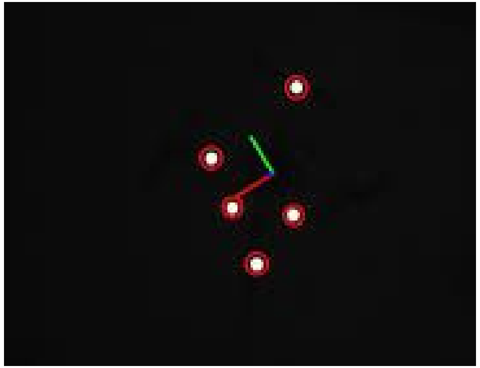

Sensors
Camera
IMU
Height Sensor
SBC
Perception
Data Acquisition
World Modelling
Pose Estimation
Wellhead Detection
Obstacle Detection
Global Planning
Tactical Planning
Search For Wellhead
Move To Pre-Docking Position
Attempt Docking
Local Planning
Take off
Land
Hover in Plane
User Interface
Heart-beat
Current Stage
Success
Input
Target ID
Size of Search Area
Start/ Abort
Output
Infrastructure
April Tag
Docking Mechanism
Wellhead
Micro Controller
Ardrone Microcontroller
Pixhawk
Manual RC Control
UAV
Ardrone
IRIS+

SYSTEM/SUBSYSTEM DESCRIPTIONS

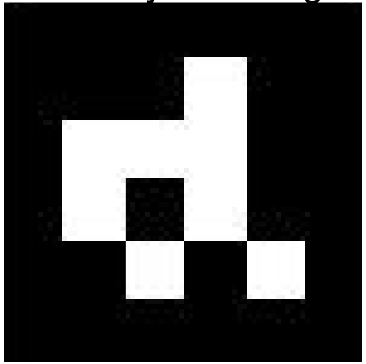
Infrastructure

Tag | Docking Mechanism | Wellhead

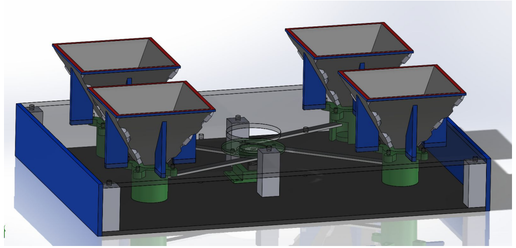
IR LED Based



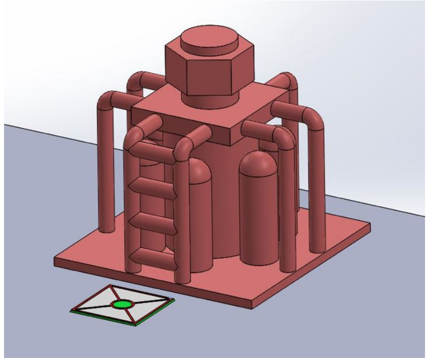
APRIL Tag
University of Michigan

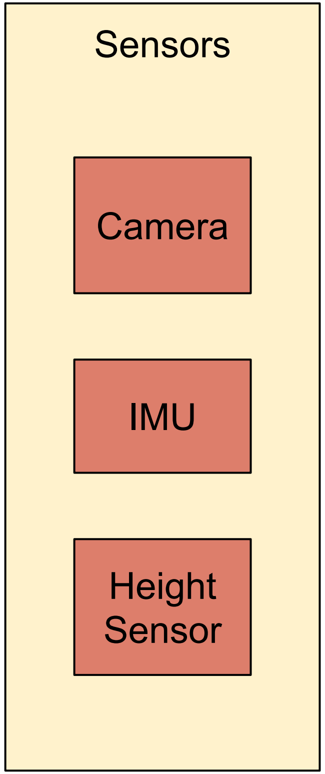


Docking
Mechanism



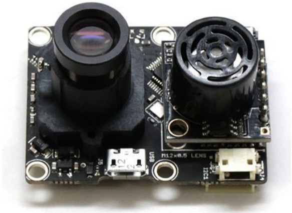
Wellhead Model





Sony Playstation Eye
source: http://amazon.com

ST Micro L3GD20 3-axis 16-bit gyroscope
ST Micro LSM303D 3-axis 14-bit accelerometer / magnetometer
Invensense MPU 6000 3-axis accelerometer/gyroscope
MEAS MS5611 barometer
source: https://pixhawk.org



PX4FLOW KIT
source: https://pixhawk.org

World Modelling

Pose Estimation

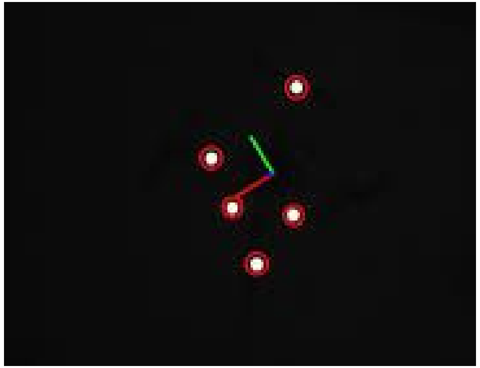
Wellhead Detection

Obstacle Detection

Lucas-Kanade Optical Flow

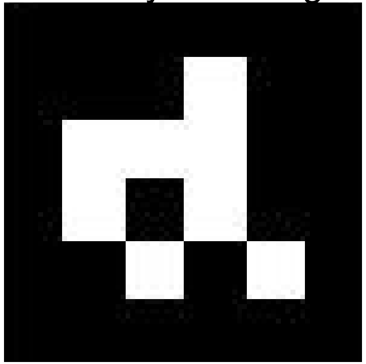
www.hizook.com

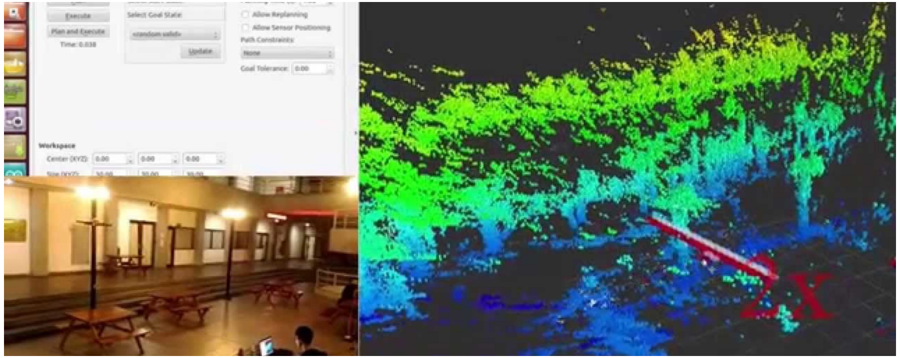
IR LED Based



www.rpg.ifi.uzh.ch

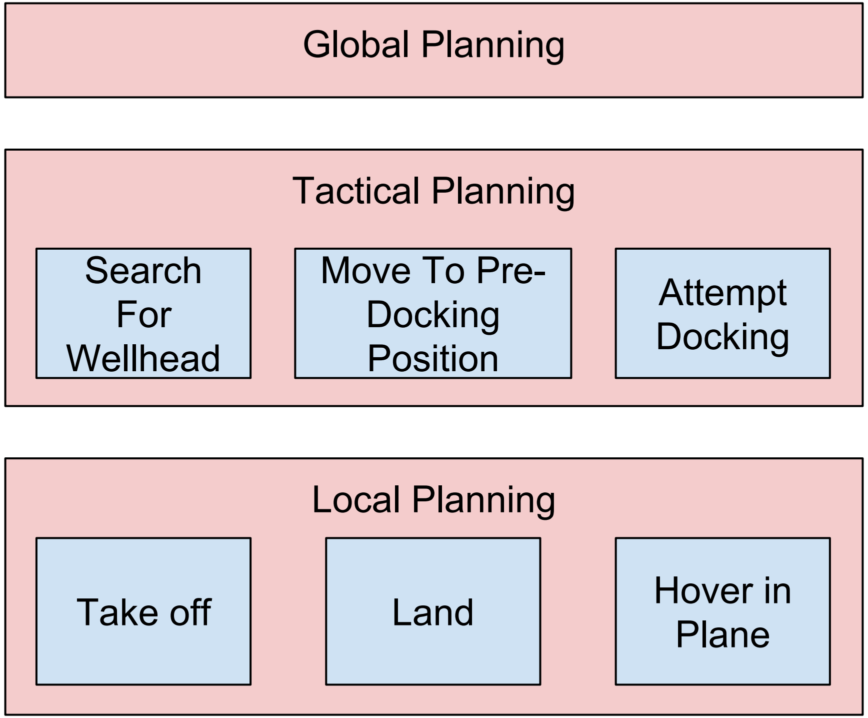
APRIL Tag
University of Michigan





source: https://vision.in.tum.de

LSD-SLAM: Large-Scale.
Direct Monocular SLAM.
Technical University Munich


Global Planning
Tactical Planning
Search For Wellhead
Move To Pre-Docking Position
Attempt Docking
Local Planning
Take off
Land
Hover in Plane

Micro Controller

Ardrone Microcontroller

Pixhawk

Manual RC Control

UAV

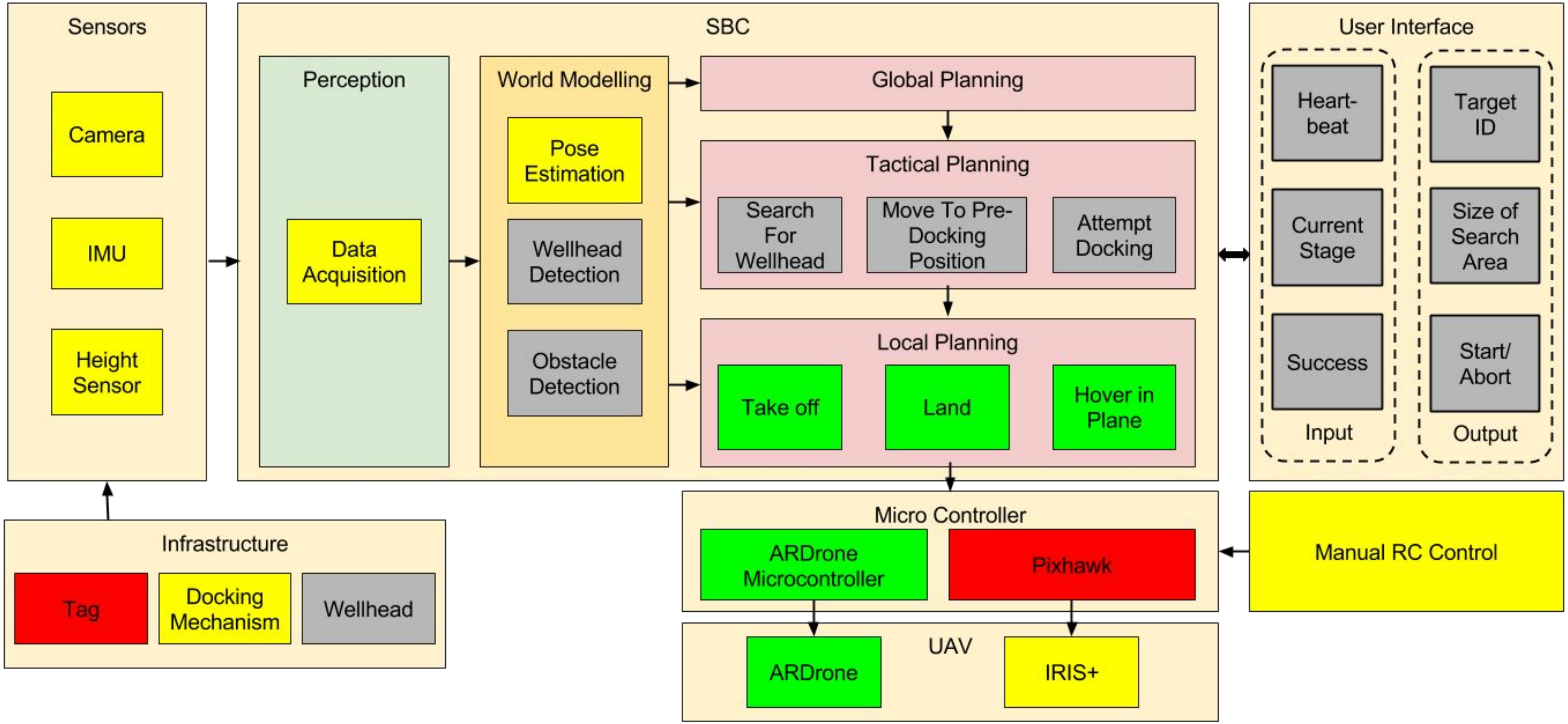
Ardrone

IRIS+

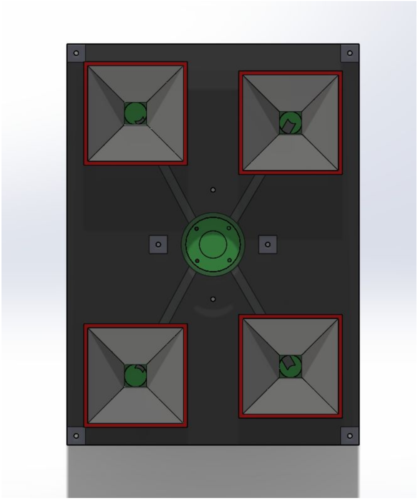
http://ardrone2.parrot.com

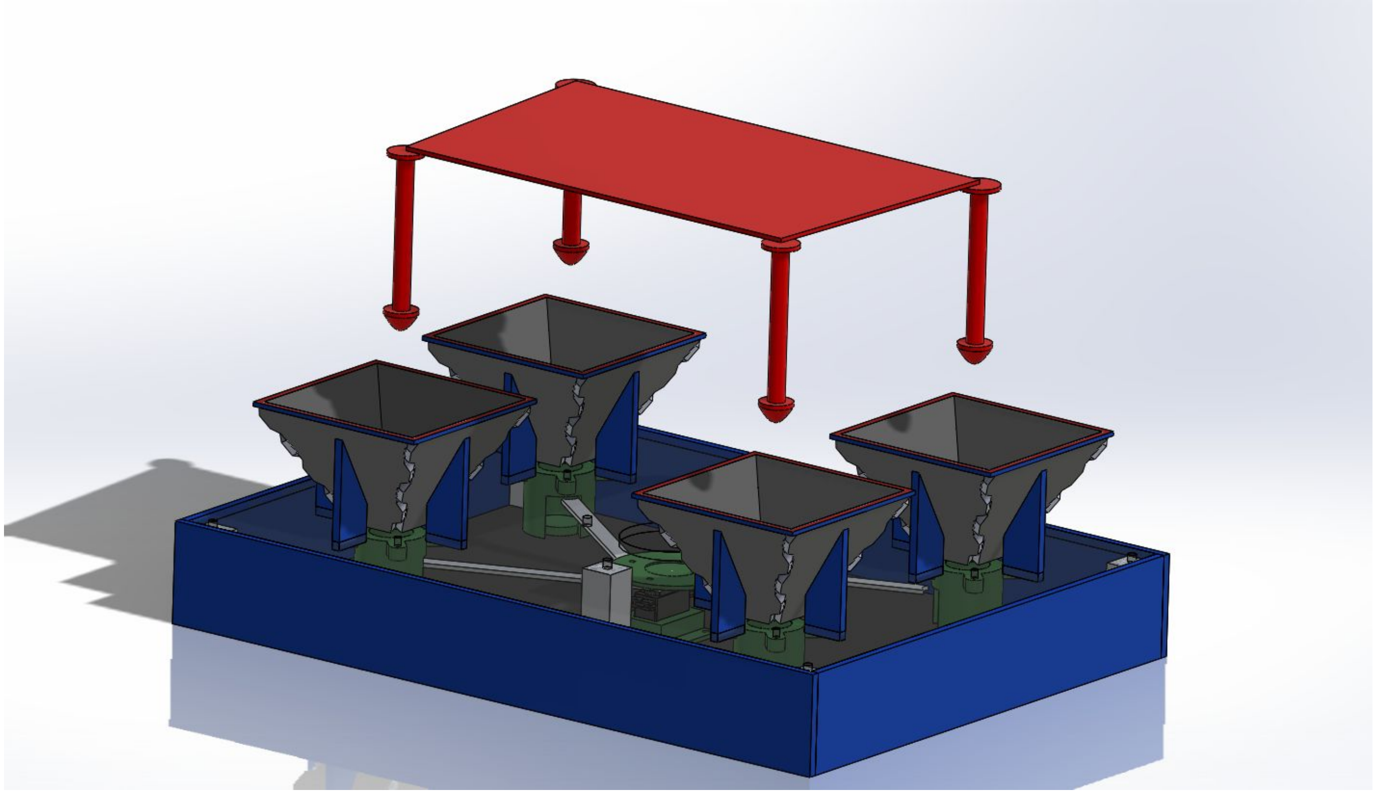
http://store.3drobotics.com/products/iris

CURRENT SYSTEM STATUS

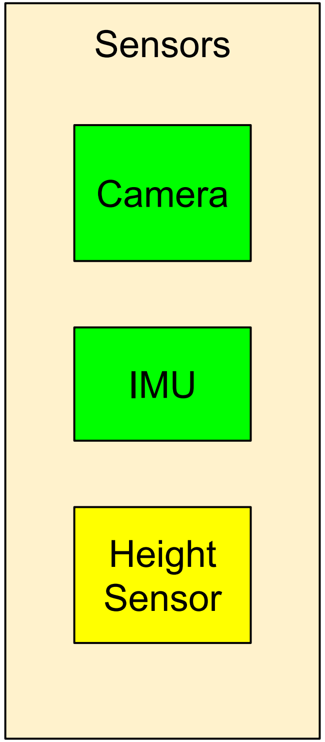

Sensors
Camera
IMU
Height Sensor
SBC
Perception
Data Acquisition
World Modelling
Pose Estimation
Wellhead Detection
Obstacle Detection
Global Planning
Tactical Planning
Search For Wellhead
Move To Pre-Docking Position
Attempt Docking
Local Planning
Take off
Land
Hover in Plane
User Interface
Heart-beat
Current Stage
Success
Input
Target ID
Size of Search Area
Start/ Abort
Output
Infrastructure
Tag
Docking Mechanism
Wellhead
Micro Controller
ARDrone Microcontroller
Pixhawk
Manual RC Control
UAV
ARDrone
IRIS+

Status: Infrastructure





Status: Sensors



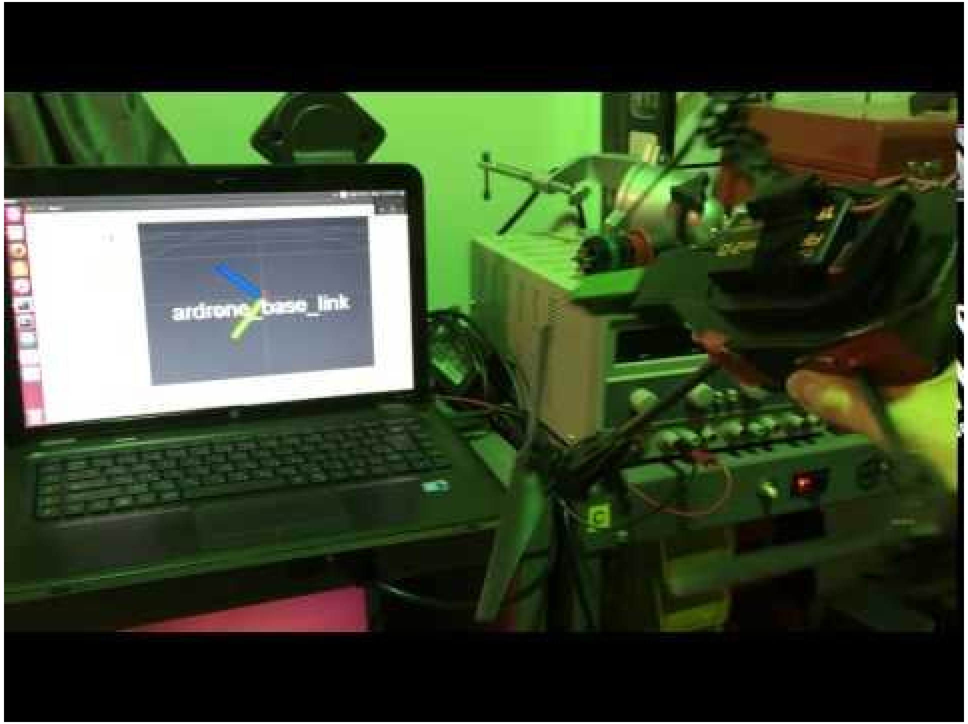
- Sony Playstation Eye ordered and received
- Physically modified to reduce weight

source: http://amazon.com

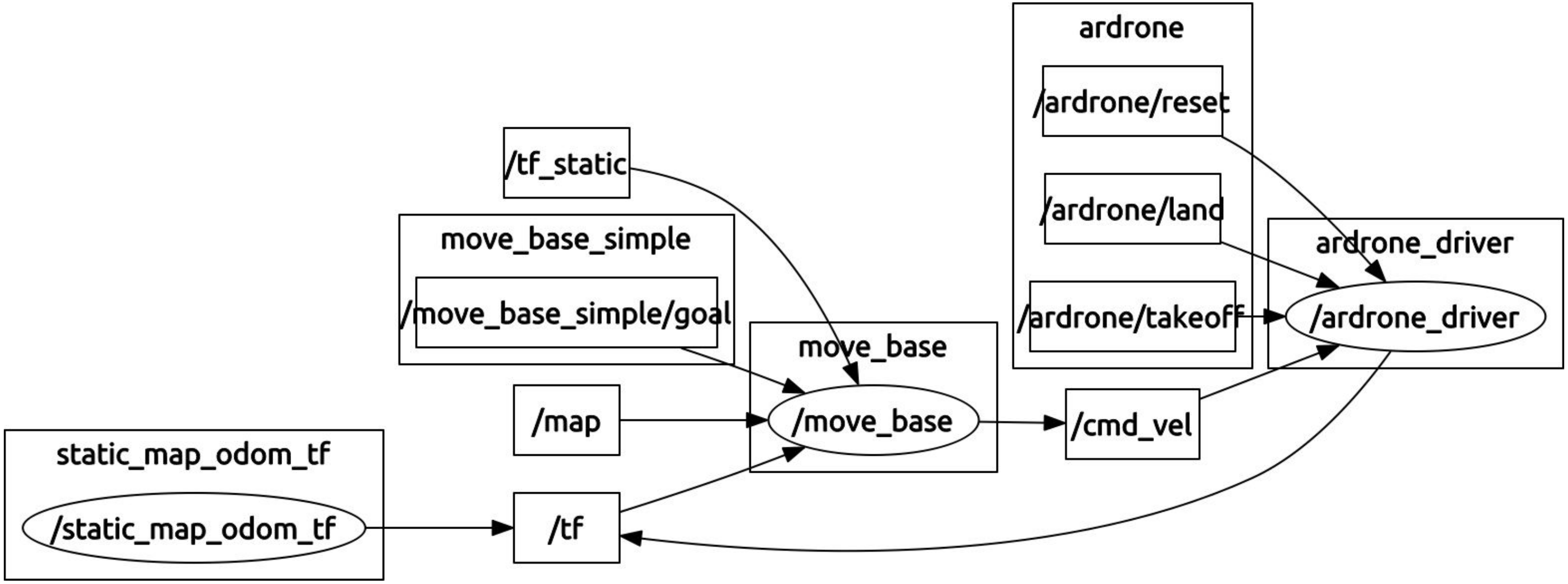
- ST Micro L3GD20 3-axis 16-bit gyroscope
- ST Micro LSM303D 3-axis 14-bit accelerometer / magnetometer
- Invensense MPU 6000 3-axis accelerometer/gyroscope
- MEAS MS5611 barometer
- Pixhawk ordered
- Waiting to receive
- Simulation up and running

source: https://pixhawk.org

Status: World Modeling



Status: Planning



Status: Planning

- MOVER node written for low level control of the ARDrone2.0

Status: Drone Hardware

Micro Controller

- Ardrone Microcontroller
- Pixhawk

UAV

- Ardrone
- IRIS+

Manual RC Control

http://ardrone2.parrot.com

http://store.3drobotics.com/products/iris

PROJECT MANAGEMENT

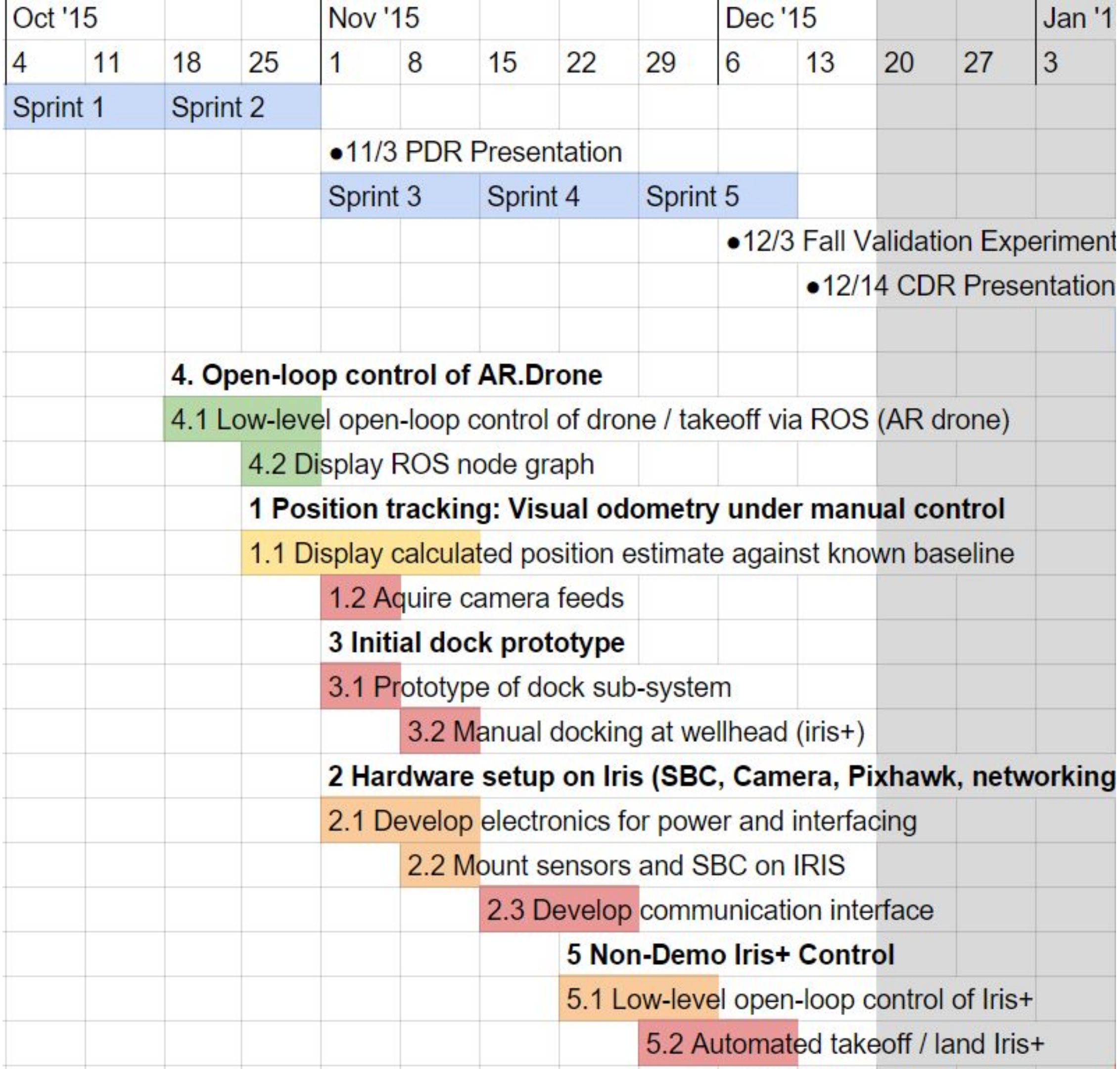
WORK BREAKDOWN STRUCTURE SUMMARY

	WBS for Fall Validation Experiment	Status
1	**Position tracking: Visual odometry under manual control**	
1.1	Display calculated position estimate against known baseline	
1.2	Aquire sensor information from camera and position sensors	
2	**Hardware setup on Iris: Demonstrate camera feed, data feed, SBC talking to Pixhawk**	
2.1	Develop electronics for power and interfacing	
2.2	Mount sensors and SBC on IRIS	
2.3	Develop communication interface	
3	**Prototype of dock: Demonstrate one proof of concept, one actual prototype**	
3.1	Prototype of dock sub-system	
3.2	Manual docking at wellhead (iris+)	
4	**Open-loop ARDrone Control: Demonstrate takeoff, move, land at push of ROS button**	
4.1	Low-level open-loop control of drone / takeoff via ROS (AR drone)	
4.2	Display ROS node graph	
5	**Non-Demo Focus Areas: Iris+ Drone Control**	
5.1	Low-level open-loop control of Iris+	
5.2	Automated takeoff / land Iris+	

SCHEDULE

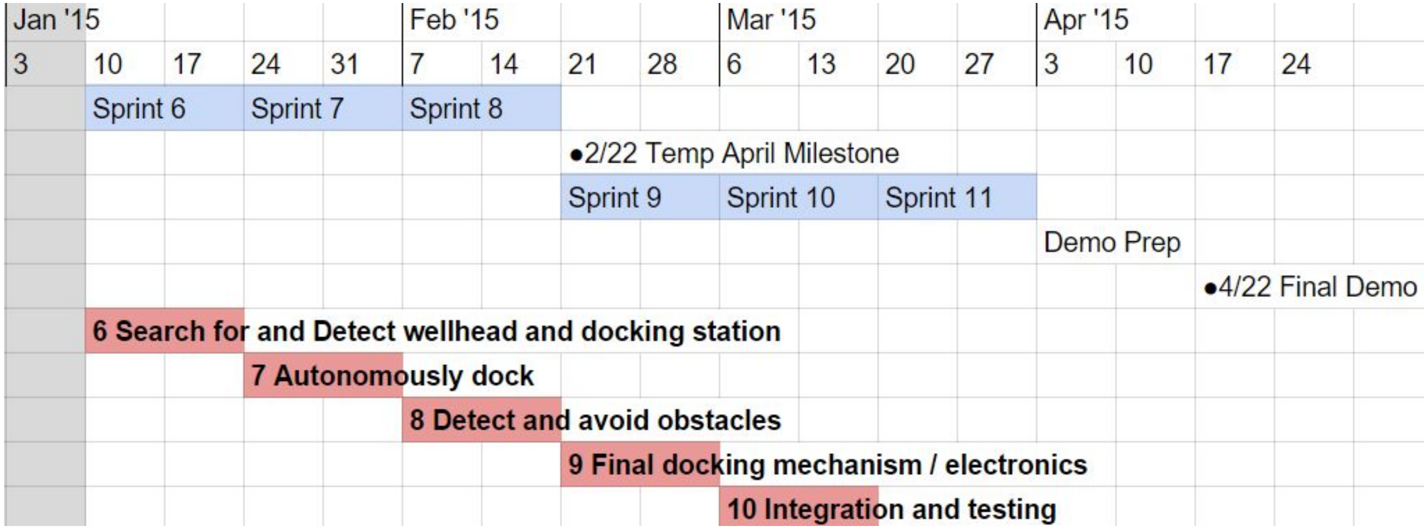
Fall Semester

On track to finish demo deliverables



SCHEDULE

Spring Semester

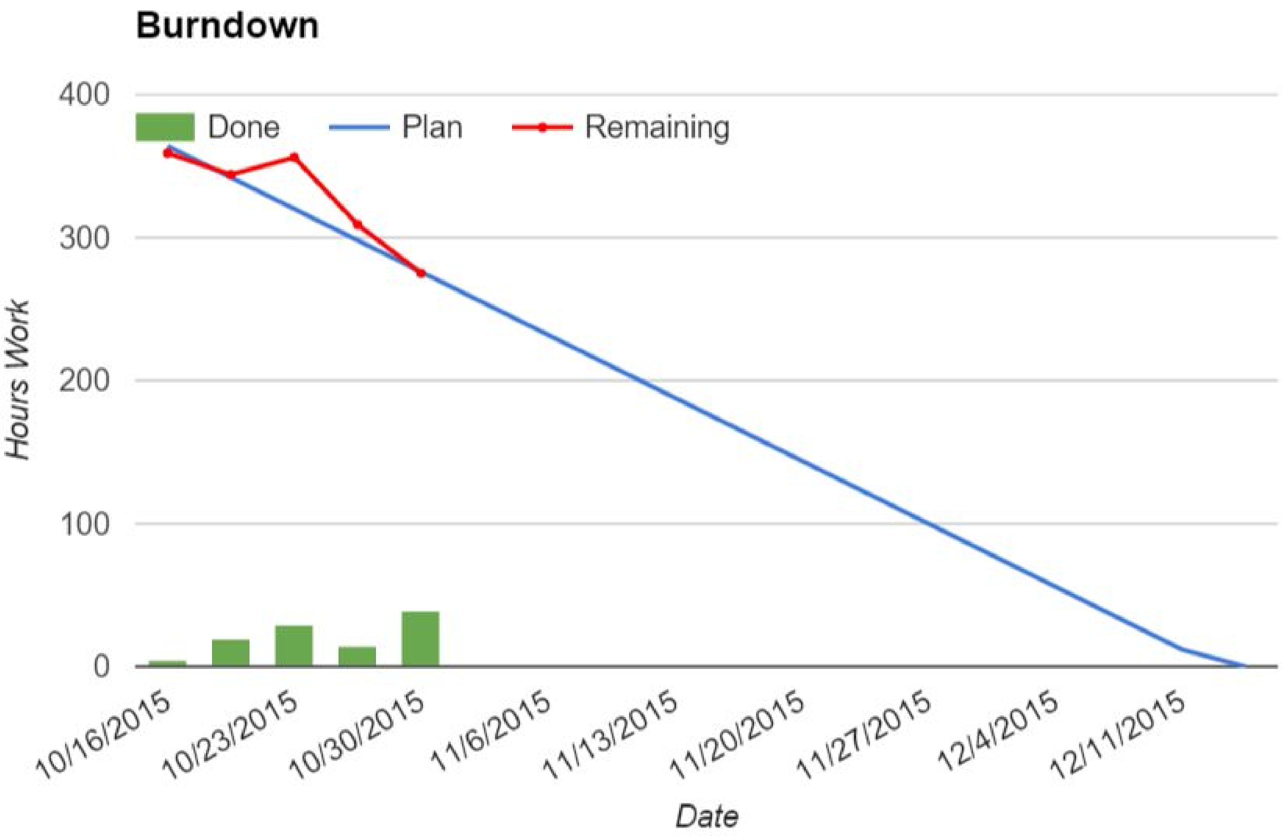


On track to finish demo deliverables

SCHEDULE

Fall Demo Preparation

On track to finish demo deliverables



HIGH LEVEL TEST PLAN

Deadline	**Deliverable Functionality**	**Method to Test**
Nov 12 Progress Review 3	Initial visual odometry implementation on AR.Drone	Pilot drone; measure displacement; compare to reported measurement.
Nov 24 Progress review 4	Initial dock prototype	Visual inspection, manual physical mating of Iris+ and dock
Dec 3 Progress Review 5 (FVE)	Hardware setup on Iris+ Drone	Launch ROS infrastructure onboard SBC, display sensor readings and state
Jan 31 Spring PR 1	Low-level control of Iris+ drone	Stable, teleoperated control of iris+ via ROS. Demonstrate in net.
Feb 28 Spring PR 2	Search for and detect docking station and wellhead	Initialize drone in random orientation. Autonomously search/find. Record success rate.
Mar 27 Spring PR 3	Autonomously dock, Final docking mechanism / electronics	With drone hovering over dock, initiate automated landing. Record success rate.
Apr 22 Final Demonstration	Integrated working system	Full demo: Take off, Search for wellhead, Orient to dock, land on dock, send signal.

Fall Validation Experiment

Test stage 1: Accurate AR.Drone2 Odometry

Location: NSH B-level hallway **Equipment:** AR.Drone2; Caution tape; Target marker

Test process:

1. Cordon off section of hallway
2. Place AR.Drone on ground
3. Place target area identifier of diameter 1 m at target location from (0,0) location
4. Hit button for takeoff. Confirm ARDrone is stable
5. Once stable, move drone to (0,0) location decided in step 3
6. Input target coordinate in meters into console
7. After movement is completed, mark position of drone
8. Confirm drone is partially within target area marker

Fall Validation Experiment

Test stage 2: Hardware Setup of Dock and Iris+ Drone

Location: NSH B-level MRSD Lab **Equipment:** Dock Hardware, Iris+ hardware

Test process:

1. Position Dock prototype hardware on benchtop
2. Physically mate Iris+ to dock and demonstrate physical fit
    a. Confirm rigidity in 5 DOF
3. Boot SBC and Pixhawk on Iris+ and:
    a. Observe ROS connectivity to external laptop at > 0.1Hz
    b. Observe sensor reading updates from Iris+ SBC
    c. Observe communication between SBC and Pixhawk
        i. View Iris+ orientation PC

Spring Validation Experiment

Accurate Iris+ Takeoff and Search for Wellhead

Location: NSH B-level hallway **Equipment:** Iris; Caution tape; dock; mock wellhead

Test Process:
1. Cordon off section of hallway
2. Place wellhead and dock within search area
3. Place Iris+ on ground at desired position within search area
4. Hit START button to initiate sequence
5. Confirm Iris lifts off and begins searching for wellhead (marker)
6. Confirm Iris arrives within 3 meter radius of wellhead
7. Confirm Iris orients above dock in pre-docking position (within 1 meter of dock)
8. Confirm Iris successfully lands in dock, constrained in 5 DOF

Mandatory goal: Dock with docking station constraints 5 DOF
Desired goal: Dock constraints 6 DOF with electrical connection

BUDGET

Total budget: $4000

Total spent to date: $888

Big ticket items:

- 3DR Iris+ Drone: $599
- Minnowboard Max x86 SBC: $150
- Odroid XU-4 Arm SBC: $83

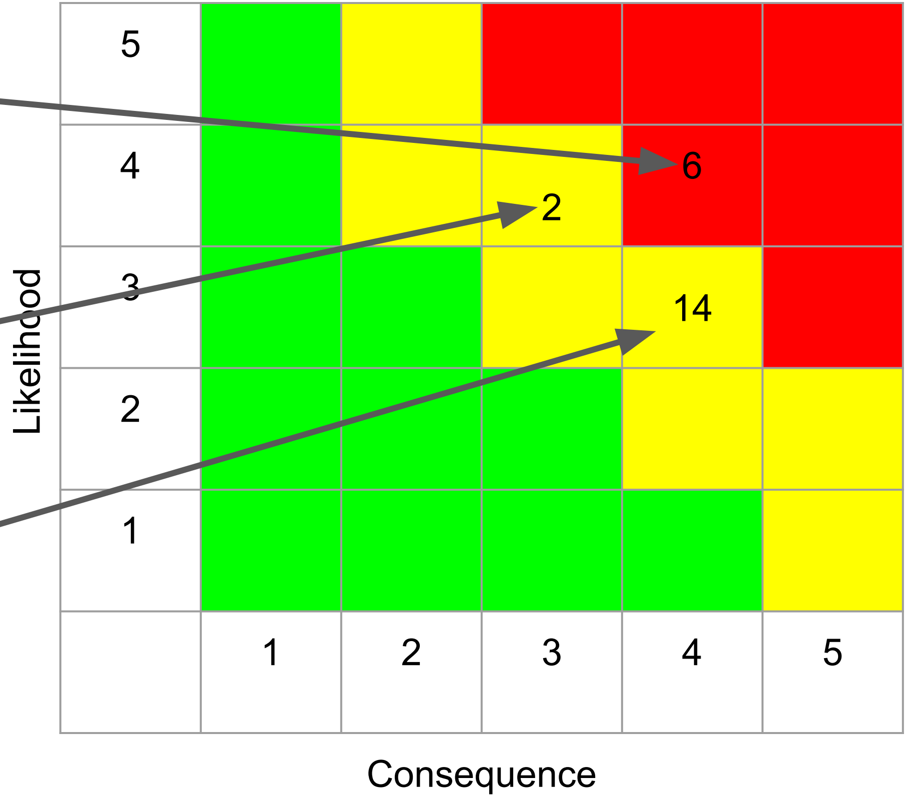
Risk Management

severity = consequence * likelihood

RIsk #	RIsk	Requiren	Type	Description	Owner	Consequence	Likelihood	Severity	Risk Reduction Plan
6	Cannot get the UAV to dock successfully	3.1.5	Programmatic	Dock design and manufacturing does not have the properties needed to successfully dock, or the quadcopters dynamics or structural properties stop the quad from successfully docking.	Job	4	4	16	Prototype multiple dock designs early and often Maintain existing AR.Drone system as fallback Focus on precision landing ASAP
14	Quadcopter goes wild during run	3.1	Technical	Quadcopter has unexpected motion that can be damaging to the quad or others around it	Job	4	3	12	Create an ABORT button on the computer to take control of quad and land it if it has unsafe motion.
8	Drone is damaged	3.1	Schedule + Cost	Drone is damaged while testing and operation	Cole	3	4	12	Use a net while testing Buy multiple backup parts Save budget for a replacement drone ($600)
2	Extra payload on UAV throws off dynamics	3.1.2	Technical	The extra payload on the quadcopter might change the dynamics of the system and will require modification of controller	Rohan	3	4	12	Test the manual dynamics with weights as soon as possible Test integrated control systems as soon as possible Keep AR.Drone as backup
15	Not enough battery life	3.3	Technical	Quadcopter does not have enough power to successfully meet requirements	Job	2	5	10	Keep extra batteries on hand for hot swap and possibly add extra battery power to payload
13	System error while in flight	3.1	Programmatic	There is some system error that occurs while Quad is in flight, resulting in a loss of control	Cole	2	5	10	Every exception must be handled correctly. Develop E-Stop / abort system.
1	Cannot get accurate localization of system	3.1.2	Technical	We cannot get accurate localization from our sensors	Rohan	3	3	9	Don't rely on accurate global positioning
9	Accurate sensing requires expensive sensor	3.1.3 3.1.4	Technical	The inexpensive sensors we have in the lab or get early cannot detect dock and/or obstacles	Erik	3	3	9	Save money on the budget for expensive sensors
4	Not able to detect obstacles and other objects	3.1.2 3.1.3	Technical	There is not enough processing or payload capacity to be able to have a good enough system to detect obstacles and other objects	Erik	3	3	9	Buy multiple processors and test them for speed and low weight. Use methods of visual recognition which require less processing and memory: like tags.
10	Dock does not rigidly connect with UAV	3.1.5	Technical	Dock that is designed does not rigidly dock	Job	3	3	9	Design a mechanism to attach rigidly to quad externally
12	Software packages do not work on ARM architecture	3.1	Technical	Software packages that we need to do certain tasks do not work on our ISA or our operating system. Reduces effectiveness and creates extra work	Erik	3	3	9	Buy an extra x86 based OBC and test for compatibility with needed packages
7	High and low level software system dependencies	3.1.1 3.1.2 3.1.3 3.1.5	Schedule	There are high level dependencies on high level and low level designs which is hard to work in parallel	Cole	2	4	8	Use the AR.Drone2 to work on the high level software design, and use the Iris+ for low level software design

TOP RISKS

1. We cannot get the UAV to dock successfully
2. Extra payload on UAV throws off dynamics of system
3. UAV goes completely out of control during the run damaging itself or someone/something else



RISK MATRIX

Likelihood

5		15, 13			
4		10, 12	2, 8	6	
3			1, 9, 4 10, 12	14	
2					
1					
	1	2	3	4	5

Consequence

Questions