CS691: Introduction to Aerial Robotics

Agile Multirotor Flight and Control through Smart Devices

Daniel Mendez

Motivation and Problem Description

- Goal of Project:
  - No previous exposure to MAVs, so as broad-as-possible project, including:
    - Hardware
    - Flight Software
    - Middleware
    - Control Software
- Problem tackled
  - A scaled down, prototype-level version of the ongoing lilycamera commercial project
  - Use iOS phone instead of dedicated gizmo
  - Eventual use cases:
    - Follow runner
    - Ultimate goal: Follow skier at 60mph, open-terrain

- Est. Release Date September 2016
- $900, 20 minute max. flight time
- Berkeley Students
- 15M raised, seed money and Series A

Original Proposed Approach

- Early Design Decisions
  - Use iOS for control
    - Use Swift, Xcode 7 for programming
  - Use quadrotor for speed, agility
    - off-the-shelf basic X240 frame
  - Use OpenPilot 3CCD Autopilot hardware
    - Modify software directly on Autopilot
  - Use WiFi for communication iOS to MAV

- First Lessons Learned
  - Flying is hard!
    - Transformation matrix in your head – nah!
    - Result – PTSD MAV
  - Batteries are very delicate
  - "Soldering! Are you kidding me?"
    - In Powerpoint speak: "Industry is nascent"
    - Interconnectivity very limited
    - Everything is its own, huge world
      - Batteries, ESC, motors
      - Connector alphabet soup
  - Very rapid change
    - E.g., OpenPilot Near-Death-Experience

Very similar to PC Industry in early 1980s, Internet in early 90s

Therein lies the fun!

Result: Needed to make significant changes

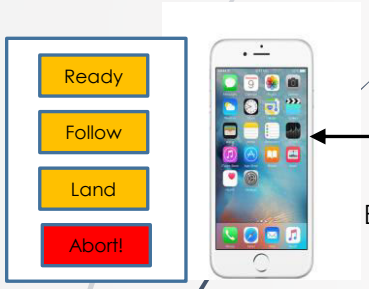
New approach and current System Description

Key Changes:

- Add a new high-level general-purpose processor to control Autopilot
- Gave up on 3CCD: Use PixRacer PX4 brand-new autopilot board/software
- Build new quadrotor from scratch

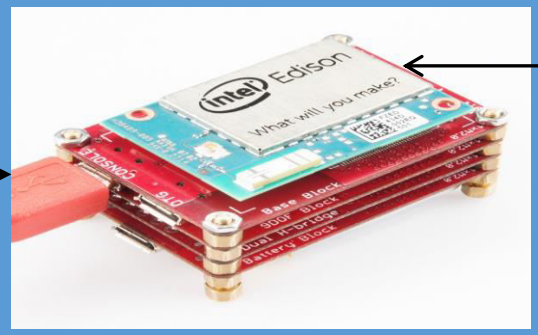
Backup Manual RC

Current Architecture



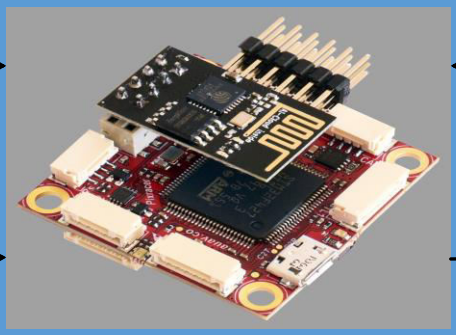
Wi-Fi
Edison creates
MAVgul WiFi AP

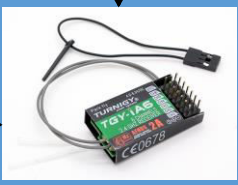
Basic Commands
GPS Info

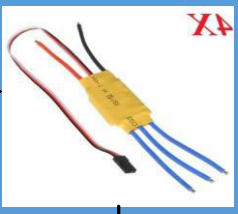


MAVROS
Serial:
UART /FTDI

GPS







Simple Custom App
Swift
Xcode IDE

Intel Edison
- Atom-based SoC
- Full Linux
- Buil-in WiFi (AP and Client)
- Expansion "Blocks":
  - Base (connectivity)
  - UART
  - IMU
  - Battery
- Many others (but not GPS)

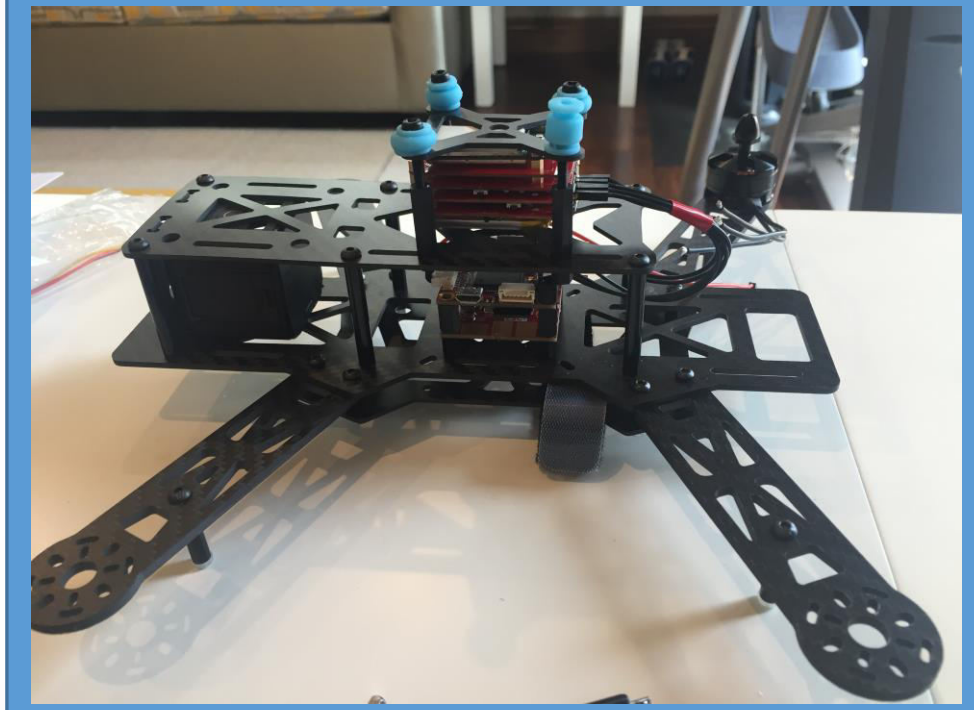
Standalone
GoPro Hero

PixRacer Autopilot Board
- Supports MAVROS
- Lots of Connectivity
- Uses QGC for Setup and Control

On-Board **MAVgul**

Results

- Completed
  - Basic iOS Swift app
  - Phone <-> MAVgul WiFi connectivity
  - Basic Edison C Programming done
    - Socket listener
    - UART interface
    - MAVROS Interface
  - Pixracer autopilot configured using QGC
- Currently working on
  - Edison <-> Pixracer connectivity
    - UART Serial to Telem2
  - Adding backup RC connection
- Next Steps
  - GPS integration (iOS and Pix4 <-> Edison)
  - Fly!
- Summer Improvements
  - Replace Hero with FPV connected camera



- LHI Emax 250mm CF Frame
- Simonk 12A ESC
- MT2204 2300KV Motor
- 6030 CF Propellers
- 3S 1300 mAh Battery