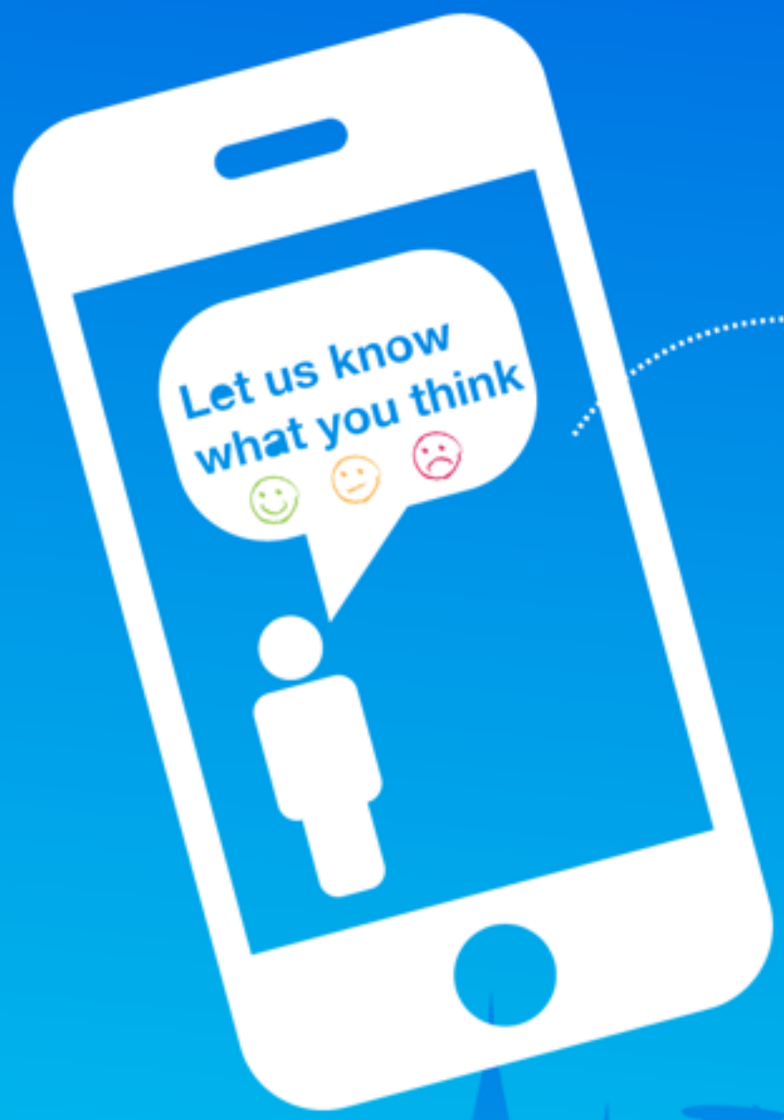
goto; conference

goto; conference



Click ‘engage’
to rate sessions
and ask questions

Join the conversation #gotocph

Drones
- the new mashup target?

Preben Thorö
October 2015

Agenda

- I have a drone, why do I need a use case?
- Where are my things?
- How does a drone work?
- Controlling the drone (from software)
- Final words

Me

- Software Engineer
- Product Manager
- Team Leader
- Trainer/Coach
- Hobby Nerd

I have a drone and a phone,
what can I do?

(for a start, not much!)

SDK level 1: Is for everyone
SDK level 2: For confirmed individuals with a use case

Drone maker DJI bans Washington flights after White House crash

28 January 2015 | Technology

So maybe I do need a use case…

(Geo-)Location is important

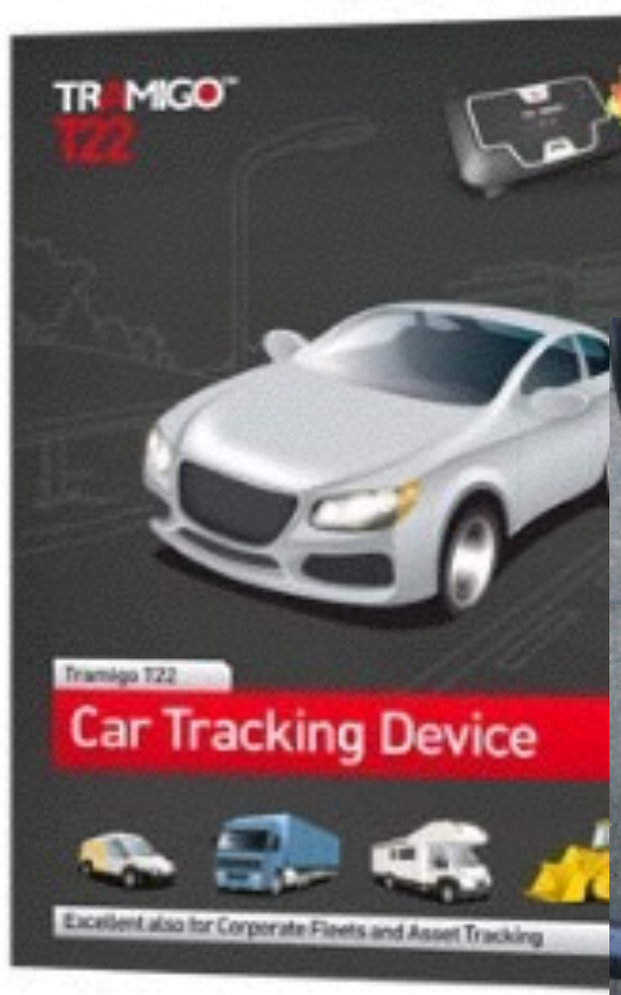
Where are my things?

We all want to know where our things are


TRAMIGO
T22
Tramigo T22
Car Tracking Device
Excellent also for Corporate Fleets and Asset Tracking

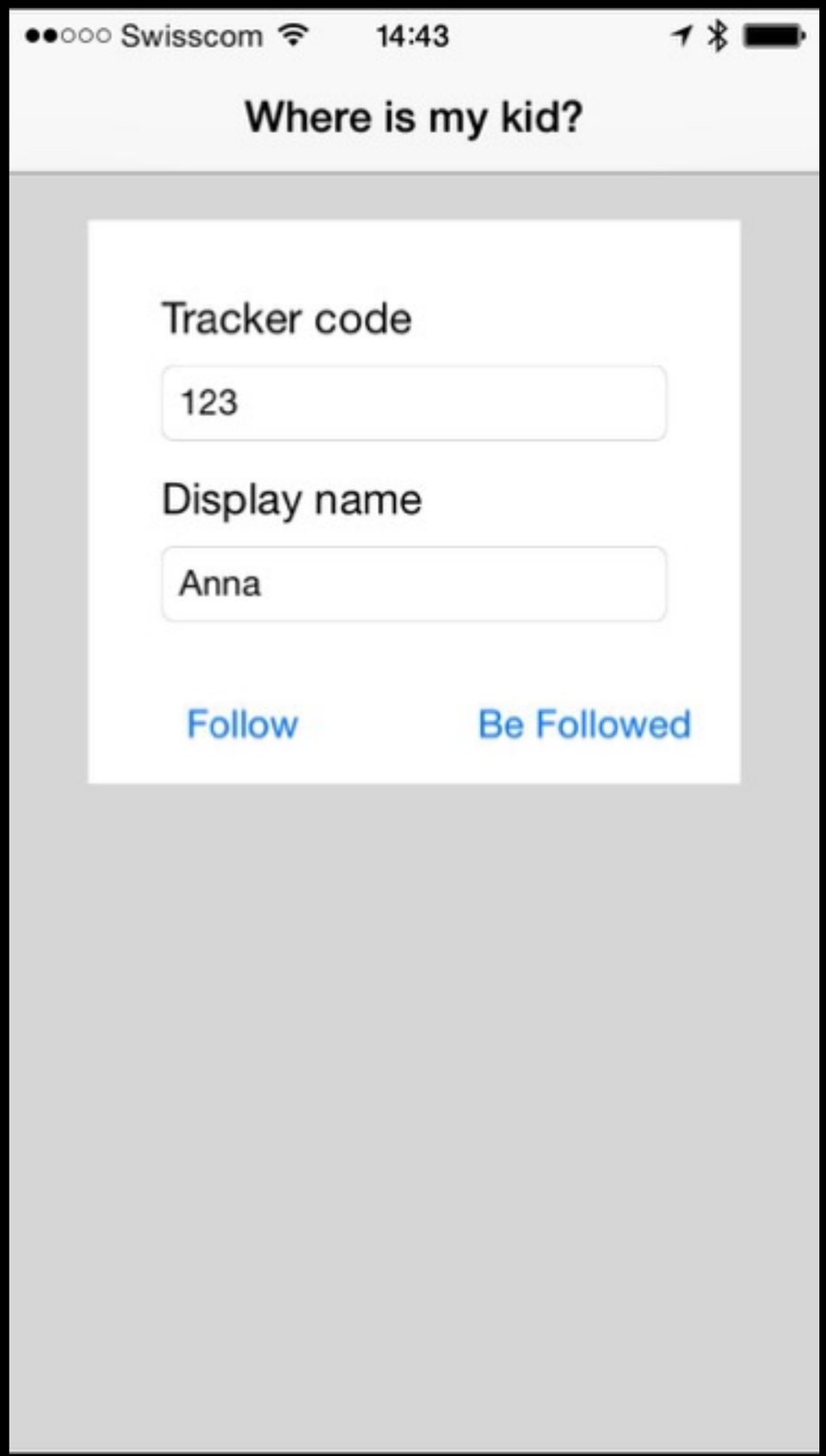

TRAMIGO
T22
Tramigo T22
Car Tracking Device
Excellent also for Corporate Fleets and Asset Tracking

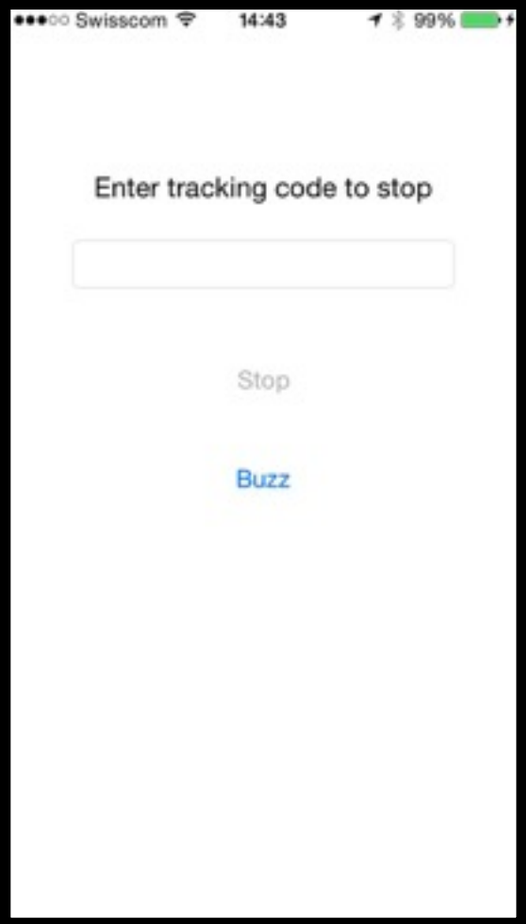

TRAMIGO
T22
Tramigo T22
Boat Tracking Device
Excellent Also for Yachts and Other Marine Vessels

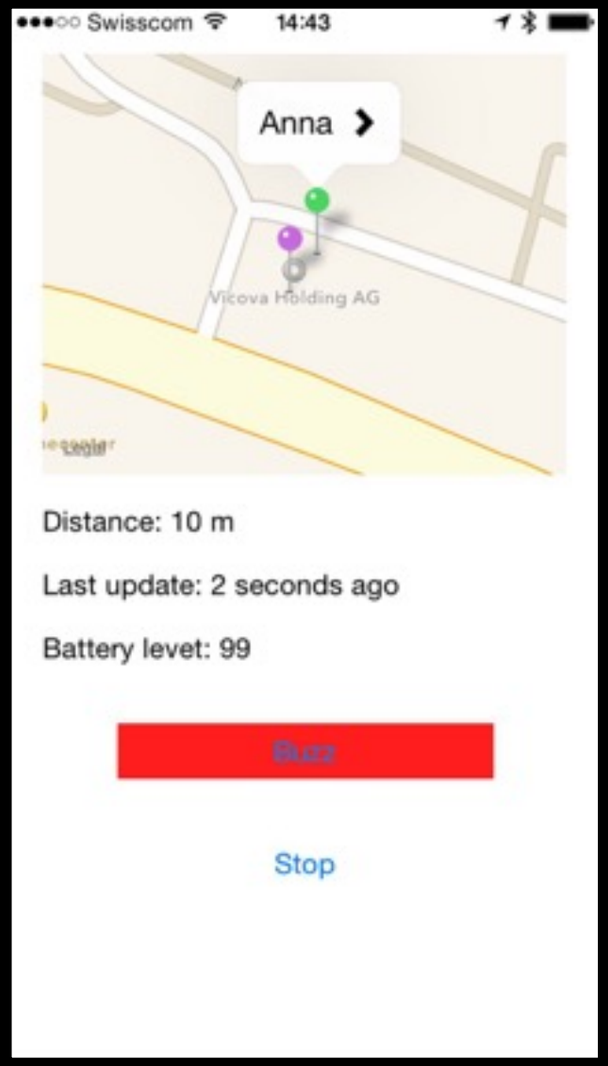

TRAMIGO
T22
Tramigo T22
Car Tracking Device
Excellent also for Corporate Fleets and Asset Tracking

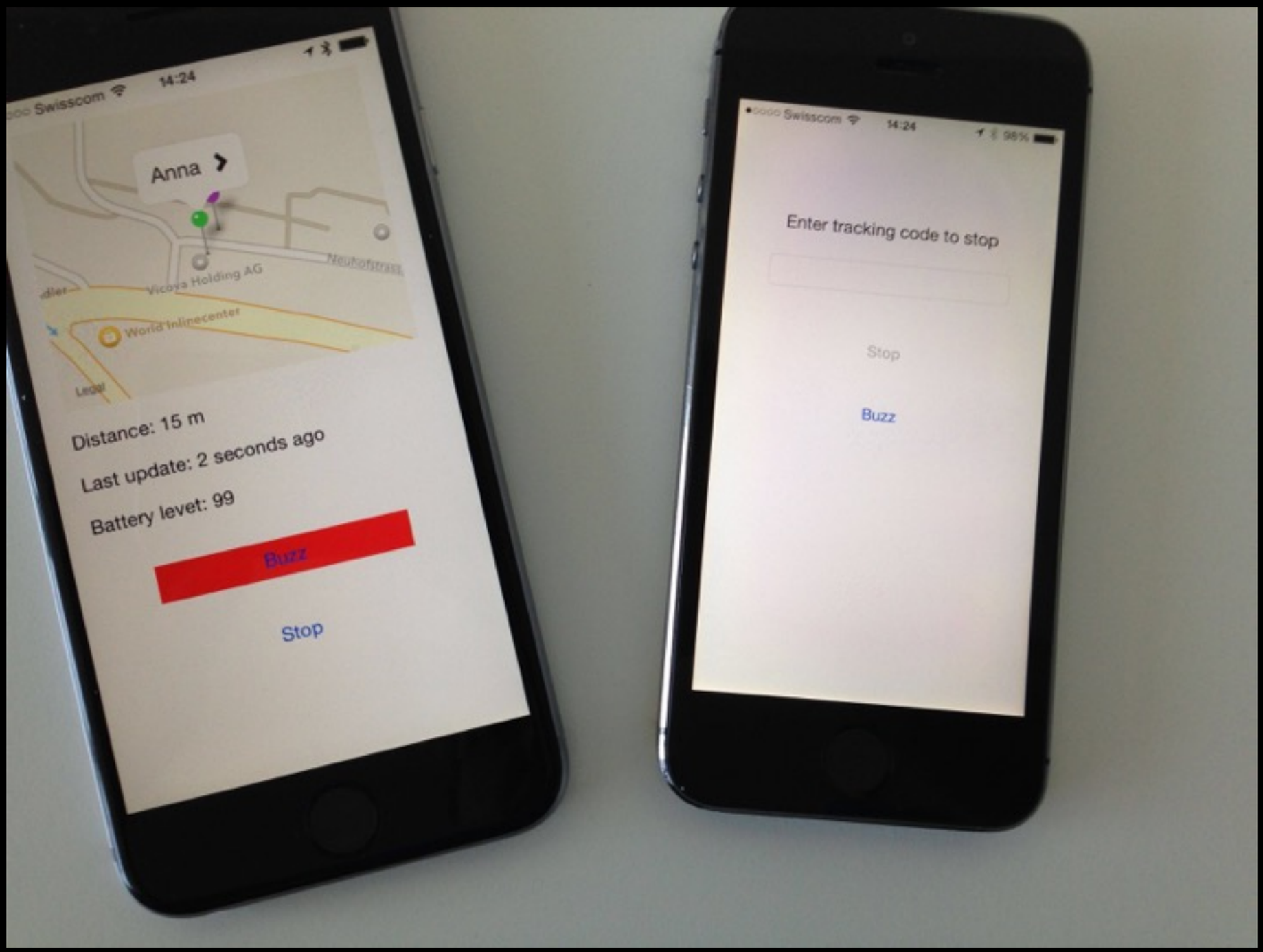

TRAMIGO
T22
Device

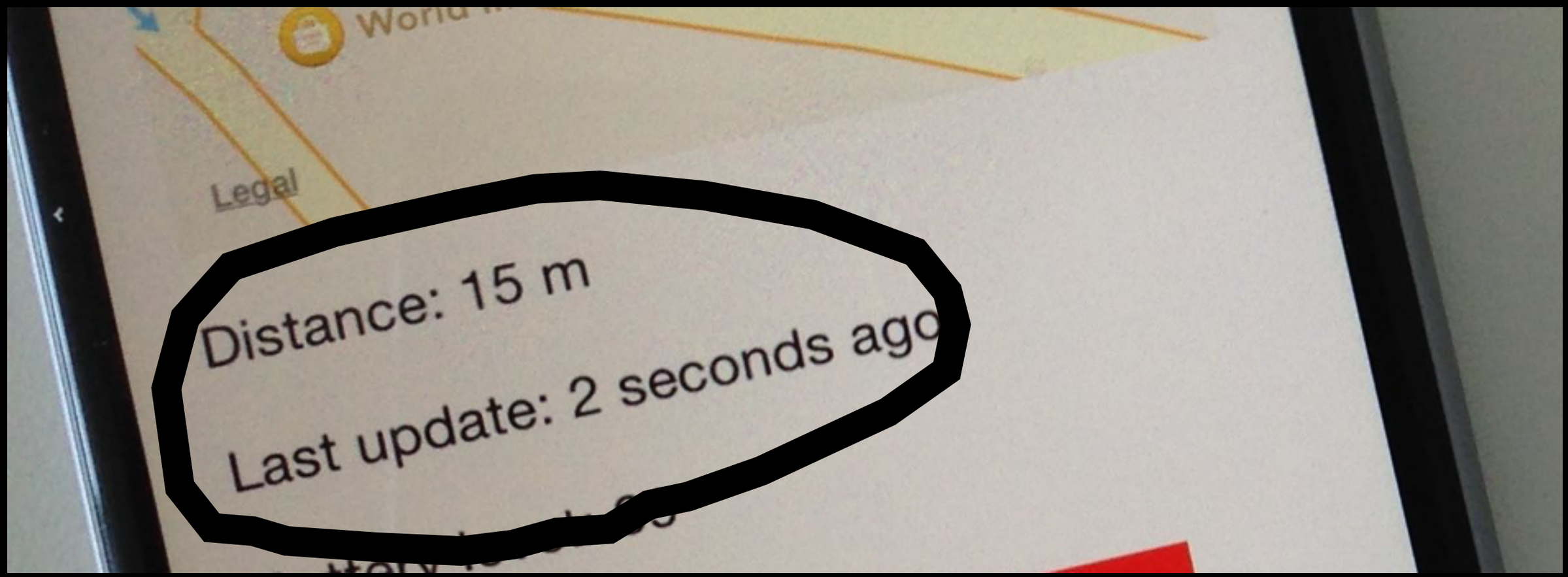
I have two things that I really like to know where are

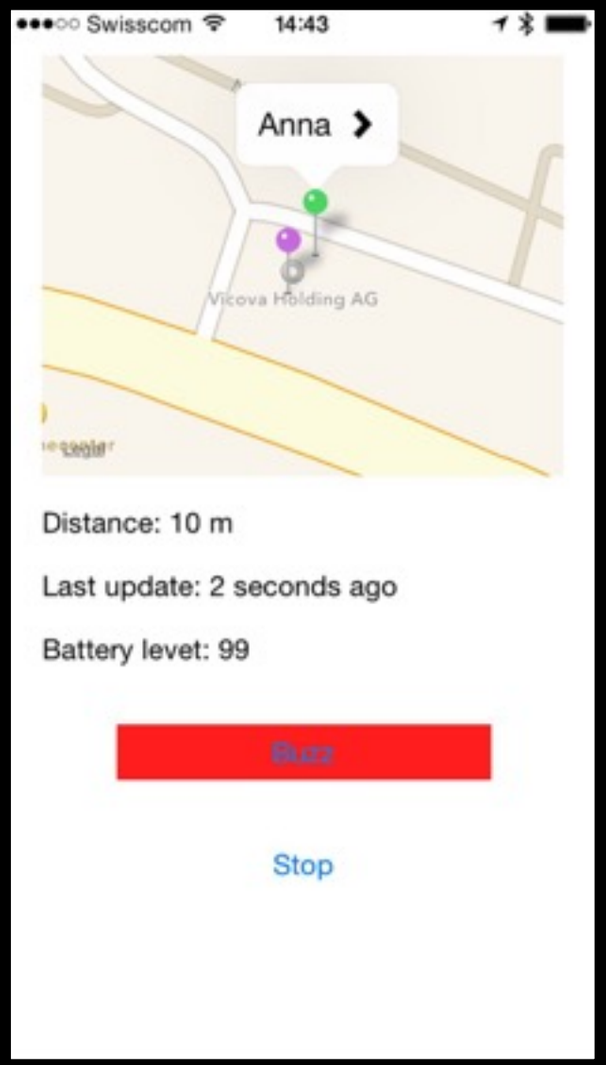

●●○○○ Swisscom
14:43
Where is my kid?
Tracker code
123
Display name
Anna
Follow
Be Followed

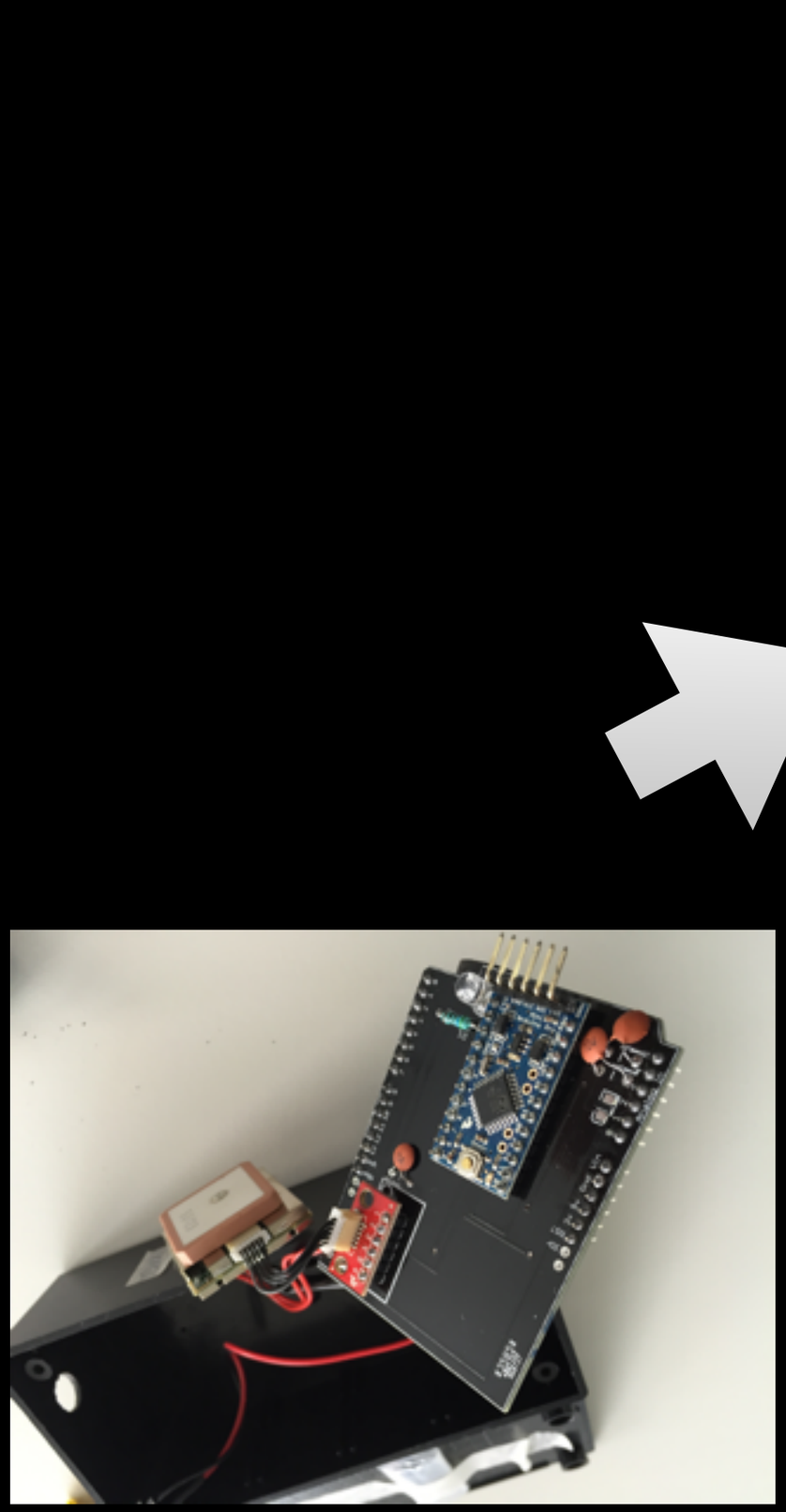

Swisscom
14:43
99%
Enter tracking code to stop
Stop
Buzz

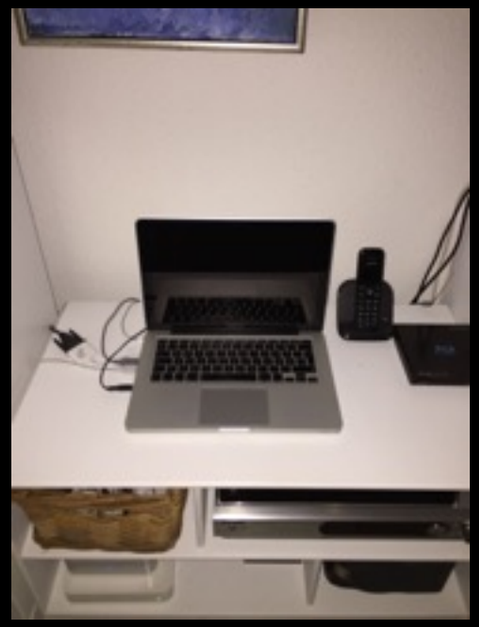

Swisscom
14:43
Anna
Vicova Holding AG
Distance: 10 m
Last update: 2 seconds ago
Battery levet: 99
Buzz
Stop

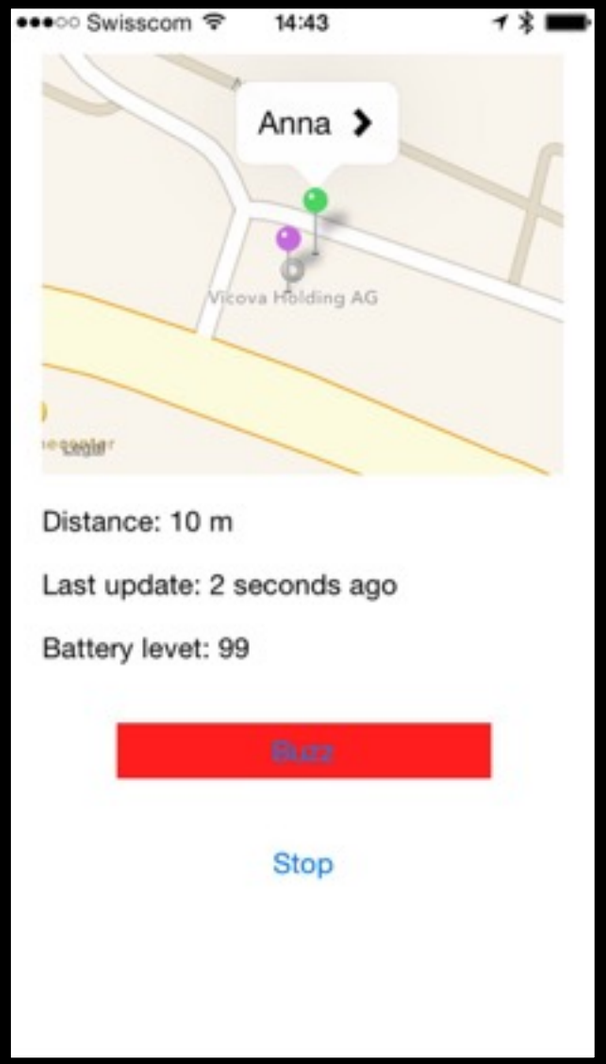

Swisscom
14:24
Anna
Neufeldstrasse
Victoria Holding AG
World Inlinecenter
Distance: 15 m
Last update: 2 seconds ago
Battery level: 99
Buzz
Stop
Swisscom
14:24
98%
Enter tracking code to stop
Stop
Buzz


Legal
Distance: 15 m
Last update: 2 seconds ago


Swisscom
14:43
Anna
Vicova Holding AG
Distance: 10 m
Last update: 2 seconds ago
Battery levet: 99
Buzz
Stop

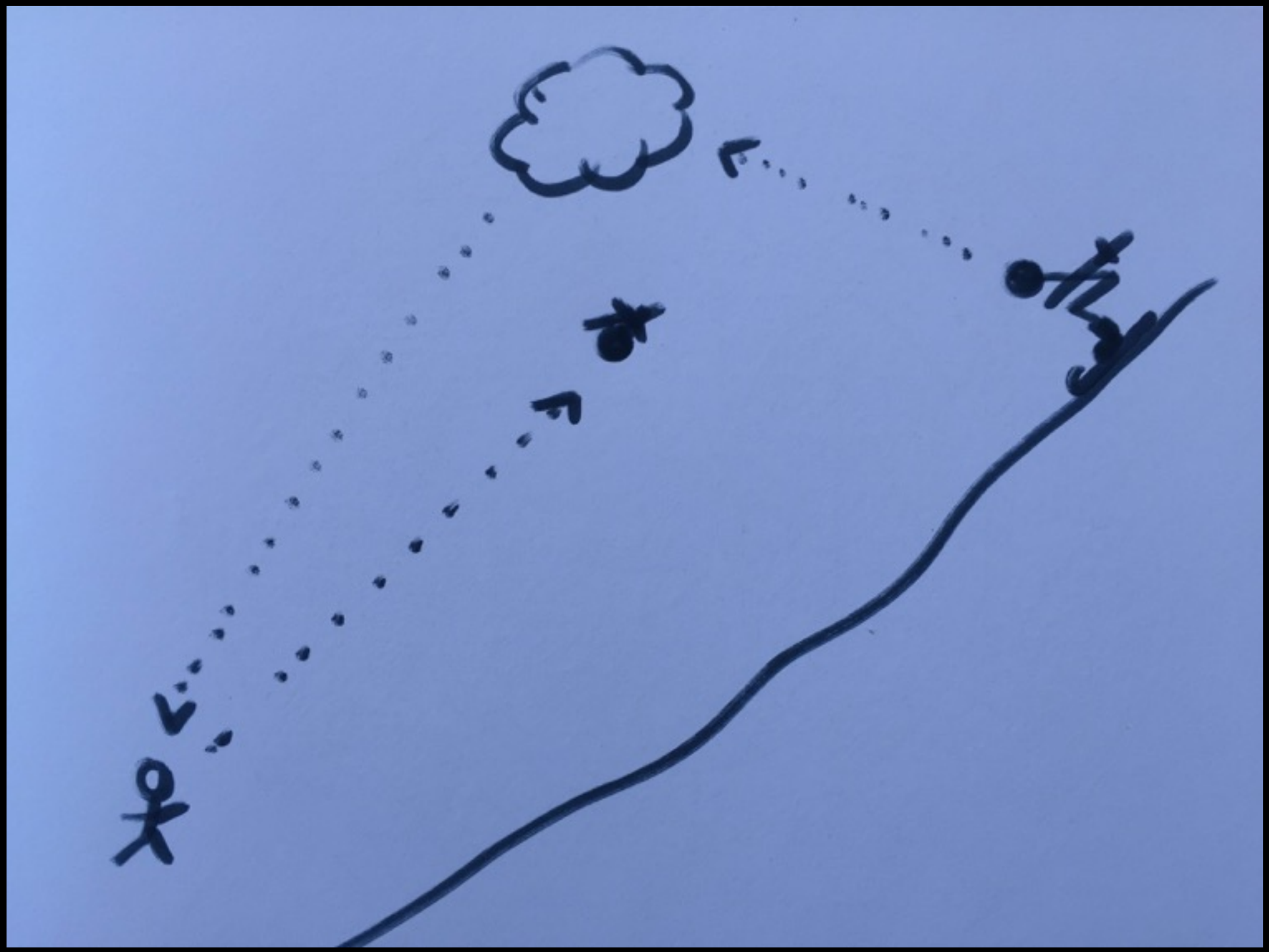




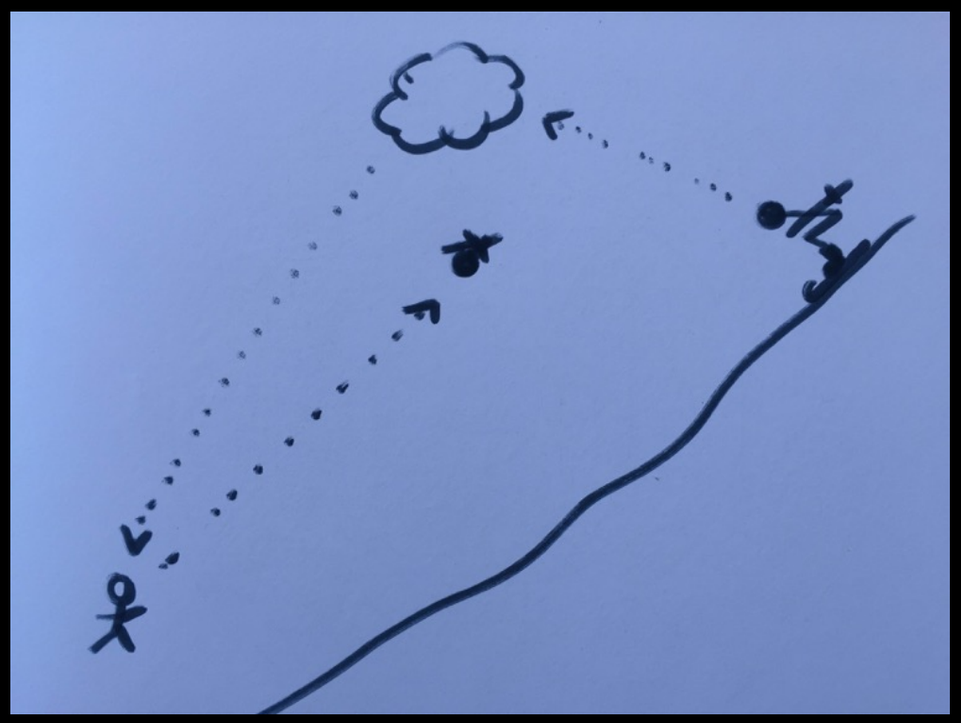

Swisscom
14:43
Anna
Vicova Holding AG
Distance: 10 m
Last update: 2 seconds ago
Battery levet: 99
Buzz
Stop

Lessons learned:

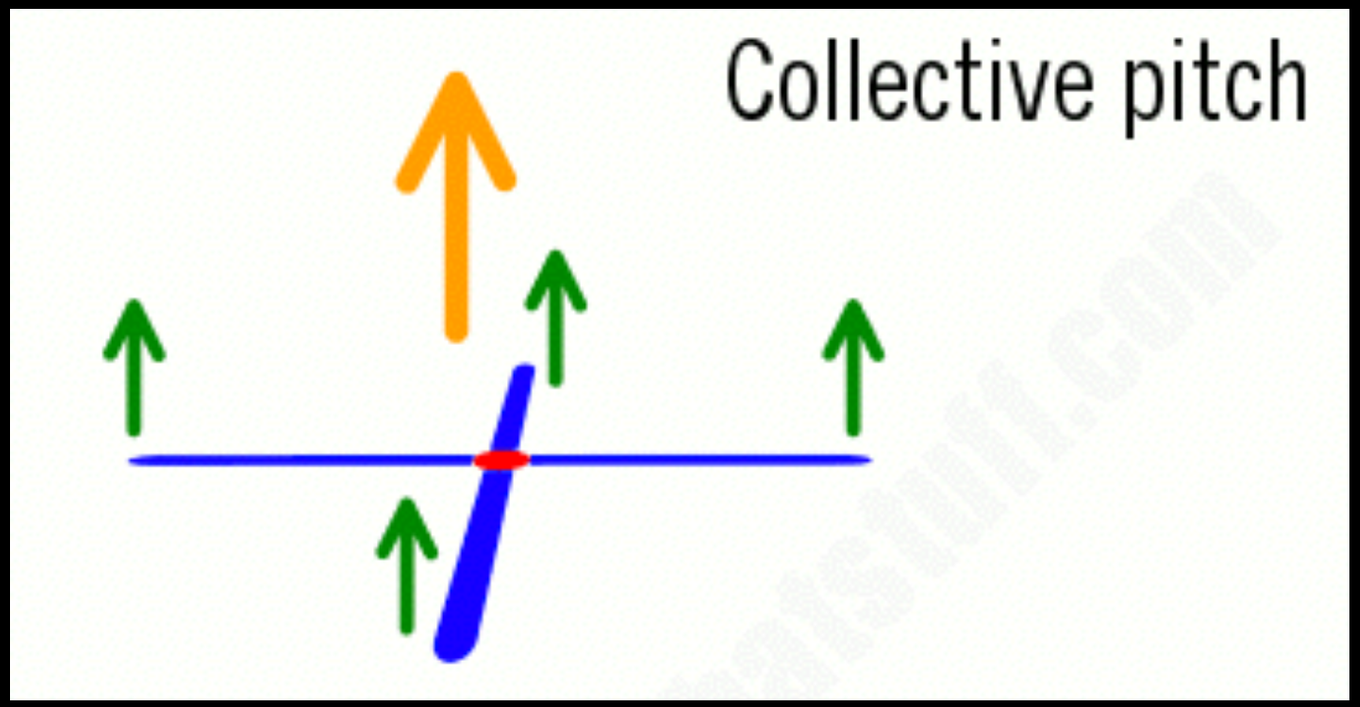
- GPS is inaccurate
- (Speed measure is difficult)
- (Couch DB is pretty cool!)



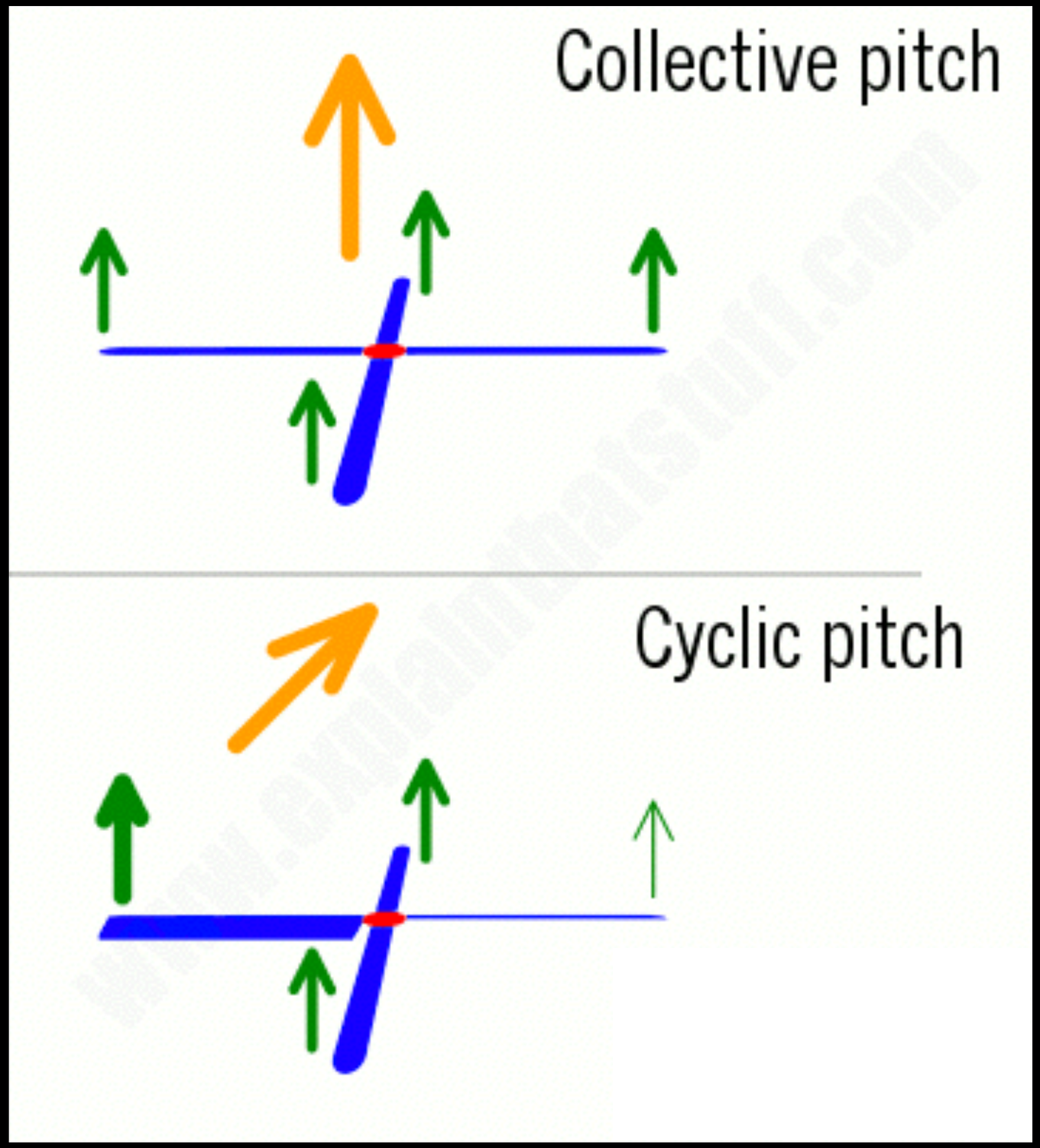
EyeAbove



How does a drone work?



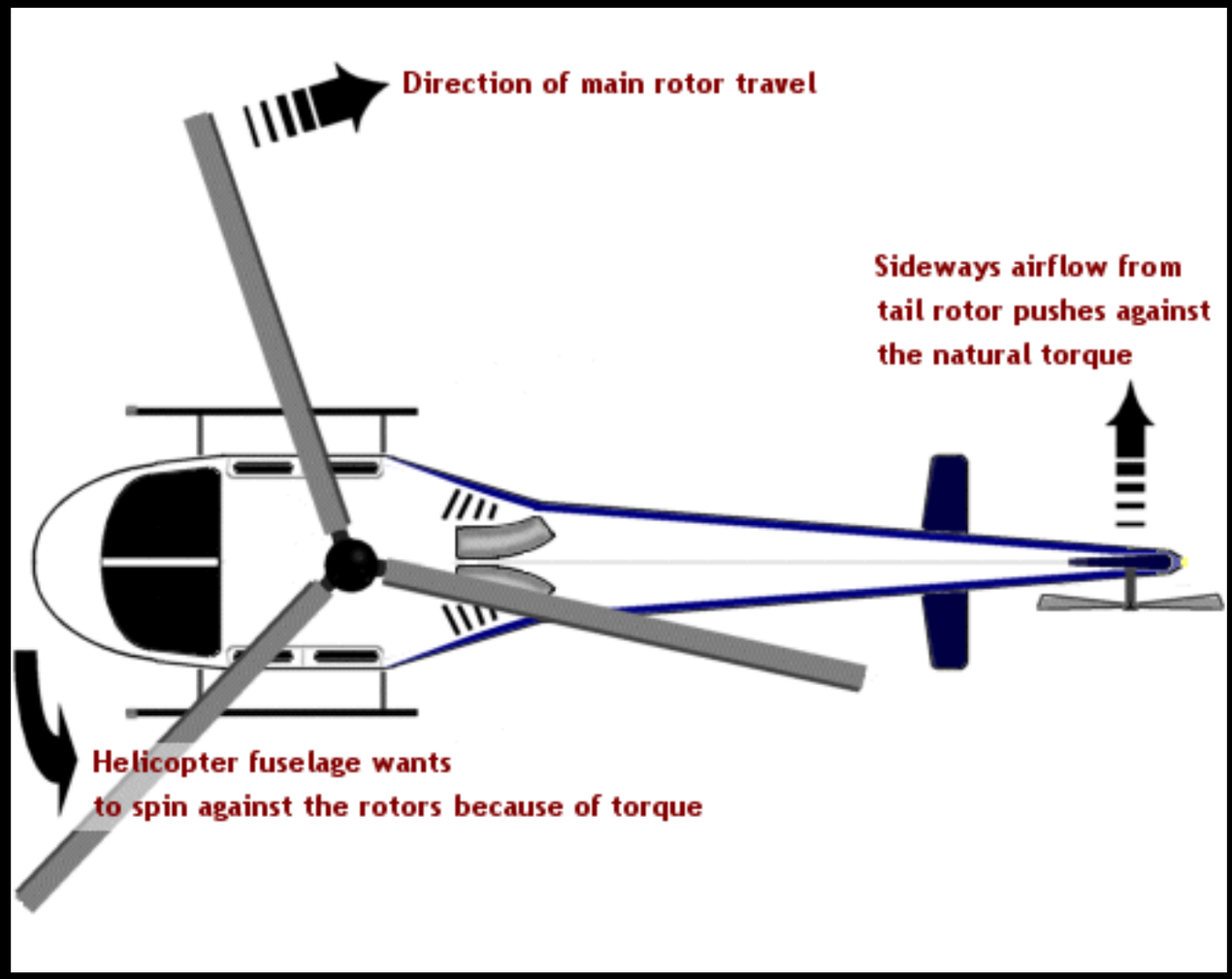
(from www.explainthatstuff.com)



Pitch

(from www.explainthatstuff.com)

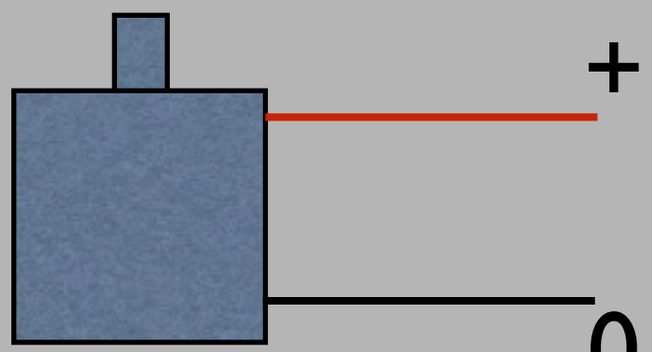
Yaw

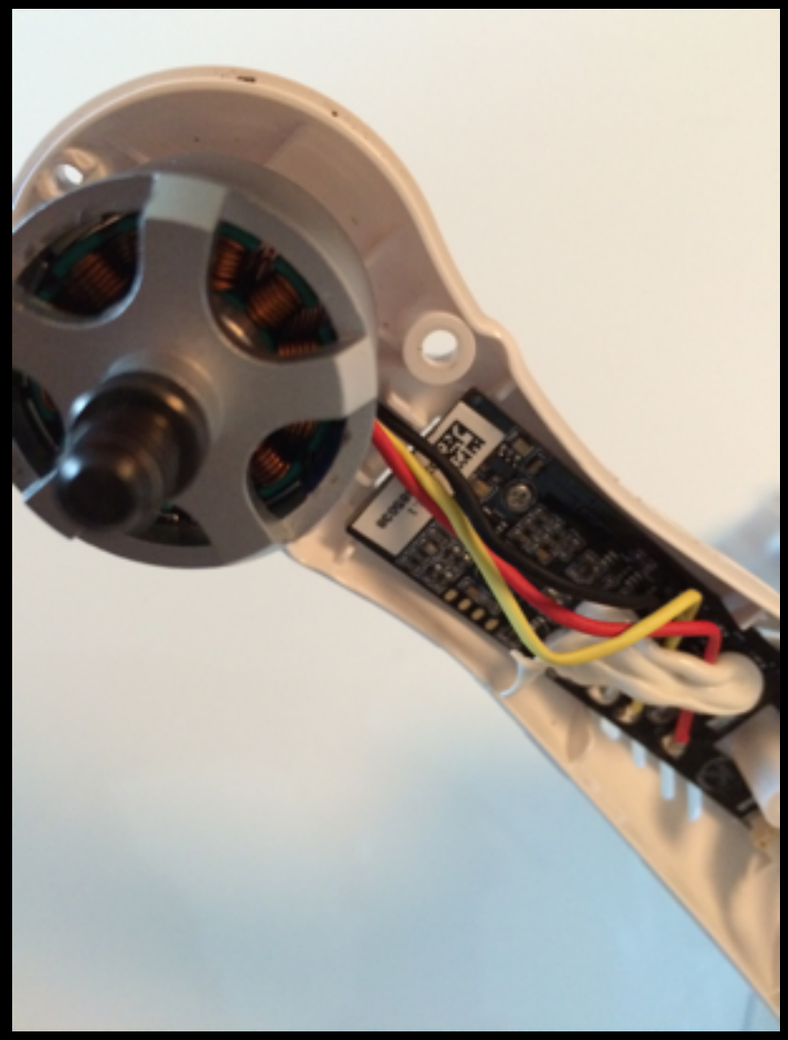


(from www.rc-airplane-world.com)

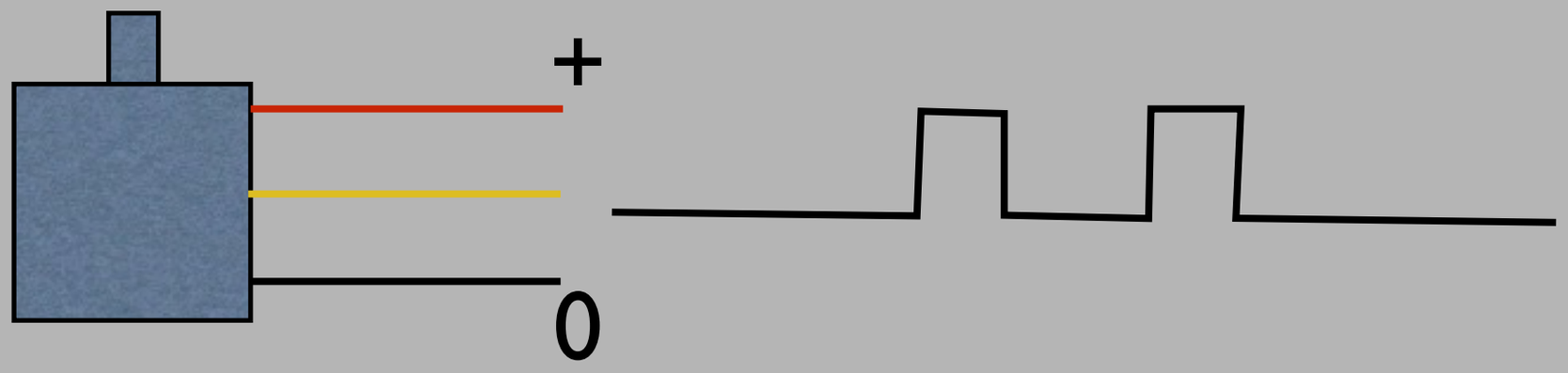
Yaw?
Pitch?

Conventinal motor



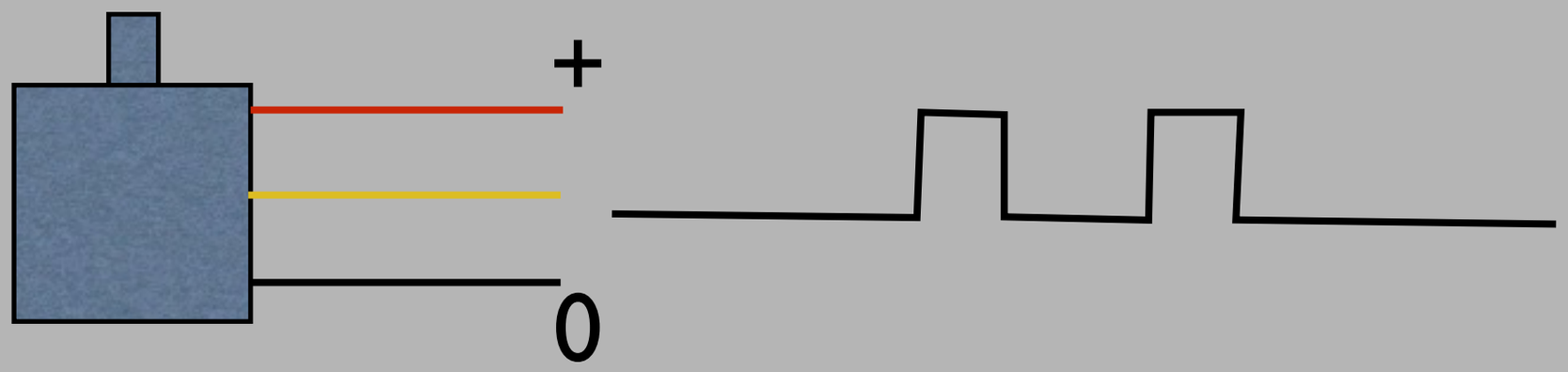


Step motor



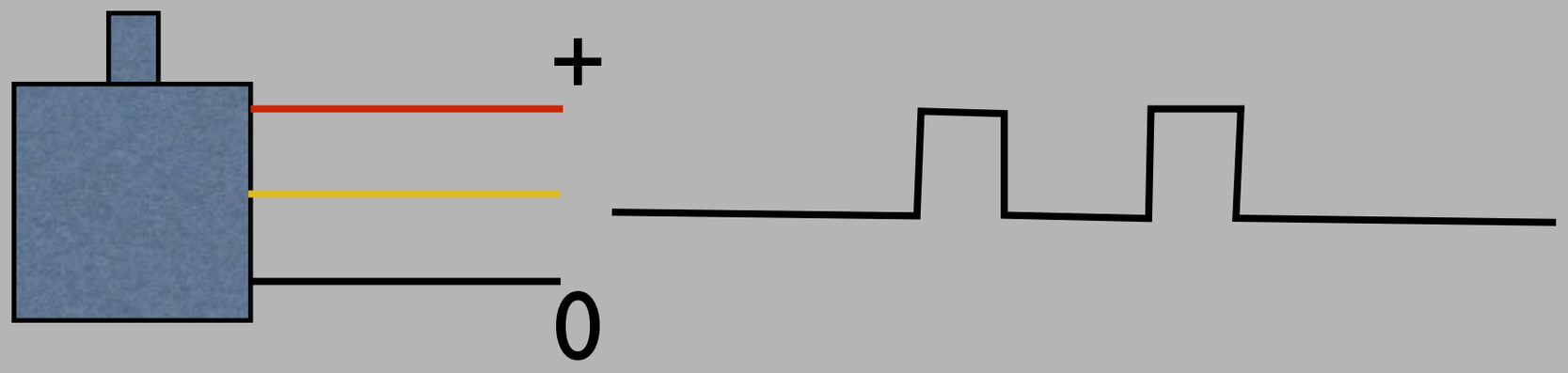
$$1 \text{ pulse} = \frac{360}{24} = 15 \text{ degrees}$$

Step motor



24 pulses/min = 1 rpm

Step motor



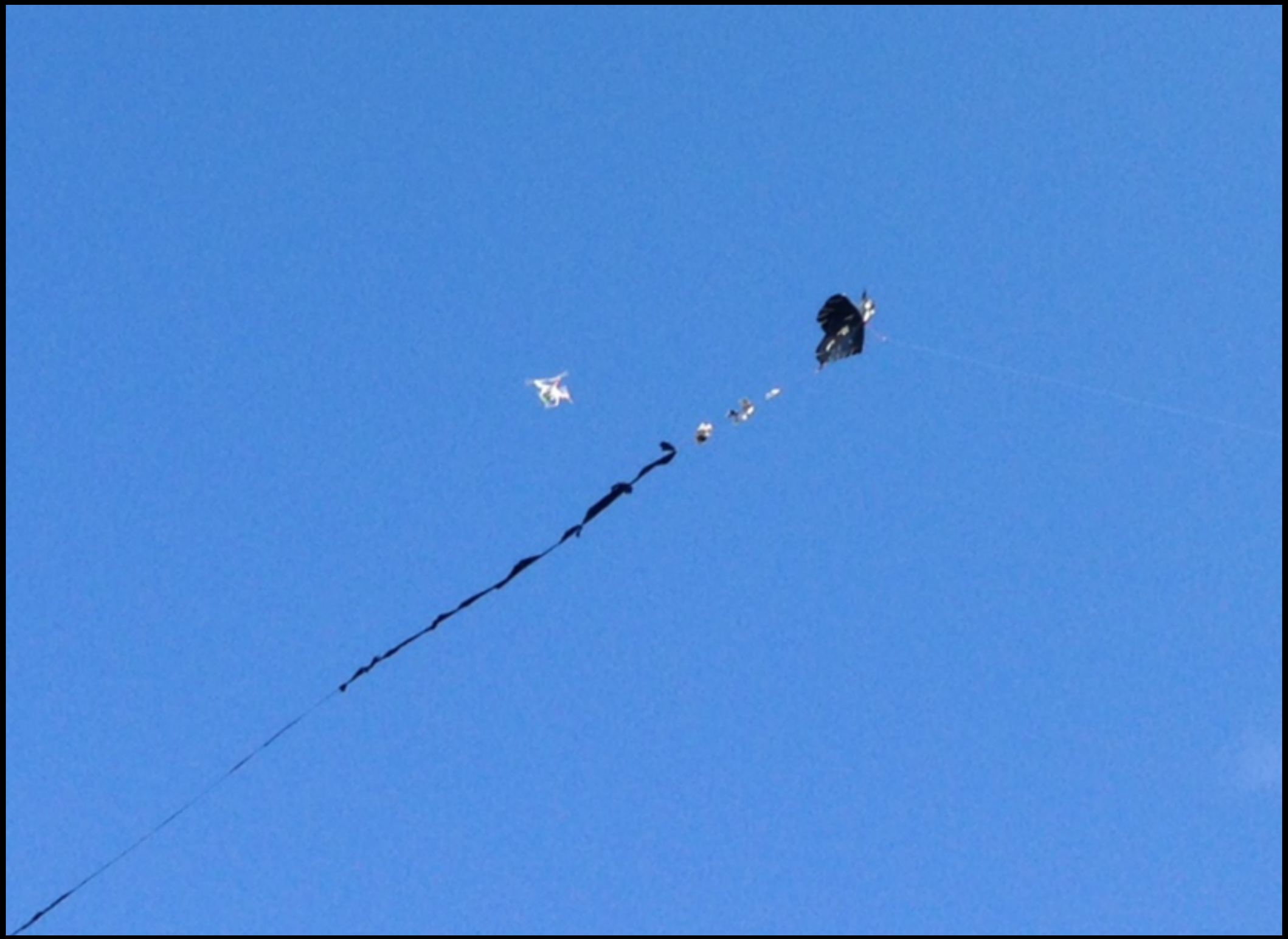
24 x 10000 pulses/min = 10000 rpm
(4000 pulses/sec)

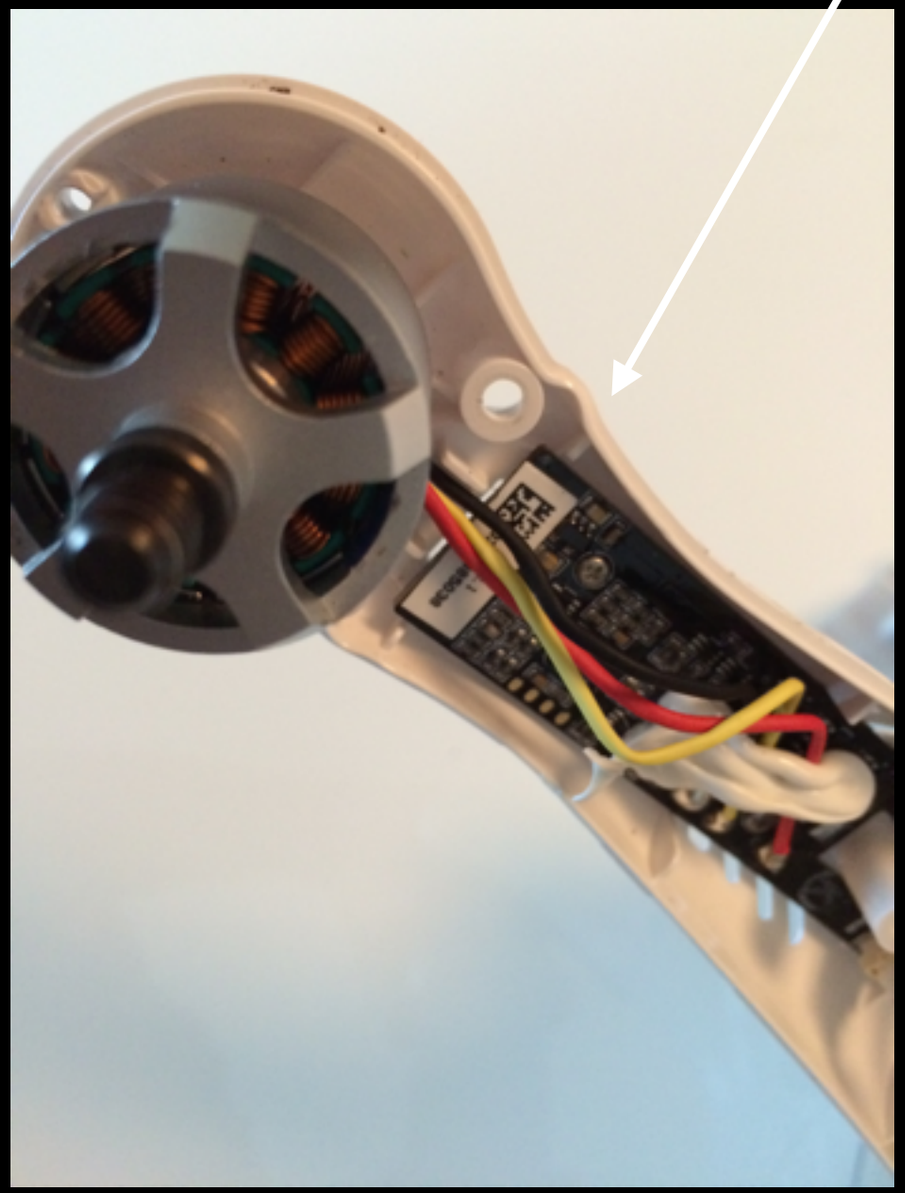
Sensors

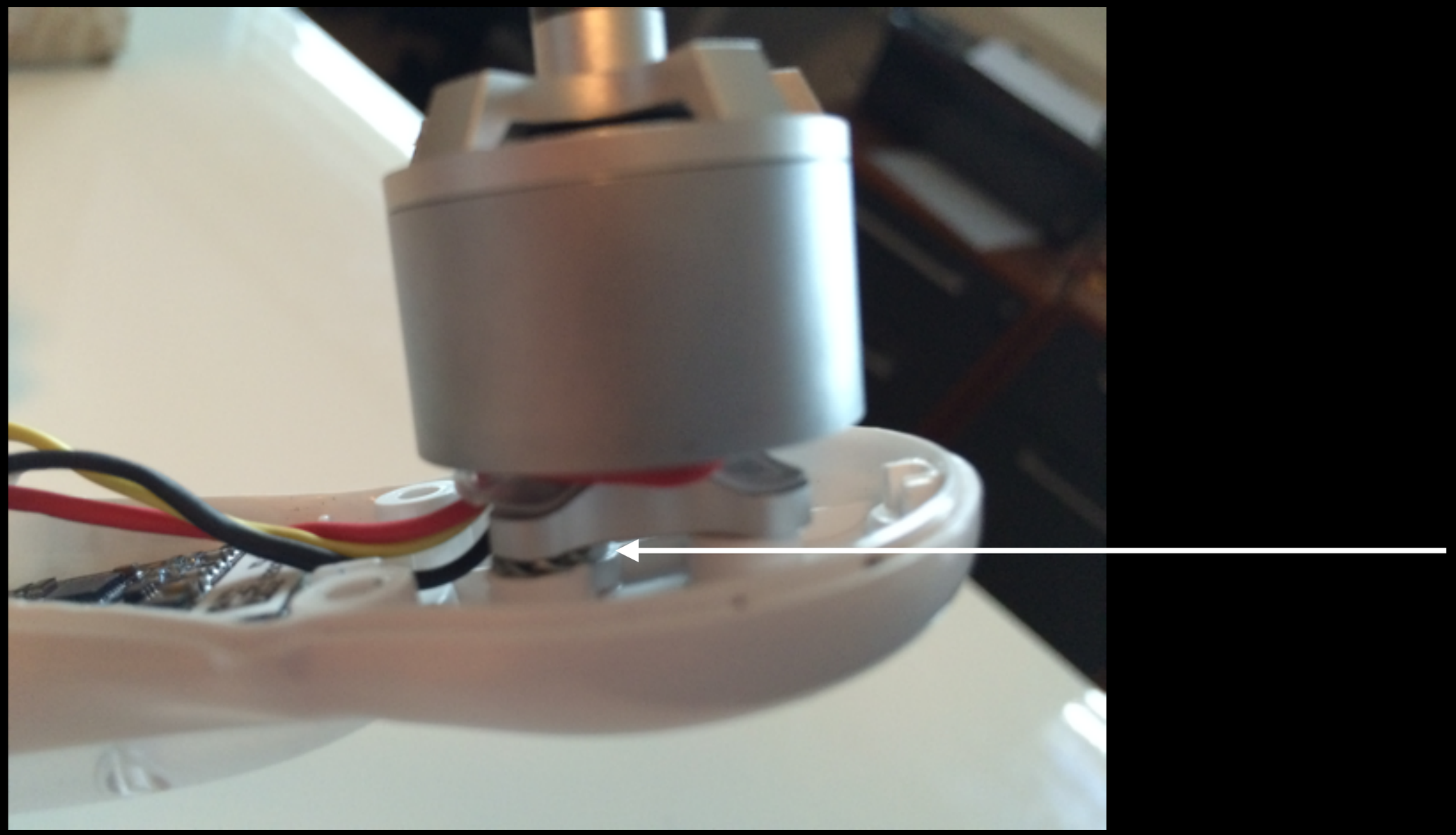
- GPS
- Compass
- Gyroscope
- Air pressure

+ lots of PI regulation

Pure coincidence!







Lessons learned 2:

The drone is constantly adjusting and correcting according to sensor input.

GPS is inaccurate but the drone is stable.

Conclusion: The drone will most likely with high stability be at the wrong position!

Real altitude
= drone altitude + offset

drone altitude

drone 0-altitude

offset

Sea level = 0m

Controlling the drone

- from software

Wifi access point

Remote controller
(ground station)

Main controller

```objc
@interface AppDelegate () <DJIAppManagerDelegate>
@end

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {

    NSString* appKey = @"0b2c15540b62b9bc097f13c4";
    [DJIAppManager registerApp:appKey withDelegate:self];

    return YES;
}

-(void) appManagerDidRegisterWithError:(int)error {
    if (error != RegisterSuccess) {
        NSString* message = @"Register App Failed!";
        UIAlertView* alertView = [[UIAlertView alloc]
        initWithTitle:@"Register App" message:message delegate:nil
        cancelButtonTitle:@"OK" otherButtonTitles:nil];
        [alertView show];
    }
}
```

```objc
DJIDrone *drone = [[DJIDrone alloc] initWithType:DJIDrone_Phantom];
drone.delegate = self;
drone.camera.delegate = self;
```

```objc
@protocol DJIDroneDelegate <NSObject>
-(void) droneOnConnectionStatusChanged:(DJIConnectionStatus)status;
```

```objc
@protocol DJICameraDelegate <NSObject>
-(void) camera:(DJICamera*)camera didReceivedVideoData:(uint8_t*)videoBuffer
    length:(int)length;
```

```objc
DJIPhantomMainController *mainController =
    (DJIPhantomMainController*)drone.mainController;
mainController.mcDelegate = self;
[mainController startUpdateMCSystemState];
```

```objc
@protocol DJIMainControllerDelegate <NSObject>
-(void) mainController:(DJIMainController*)mc
        didUpdateSystemState:(DJIMCSystemState*)state;
```

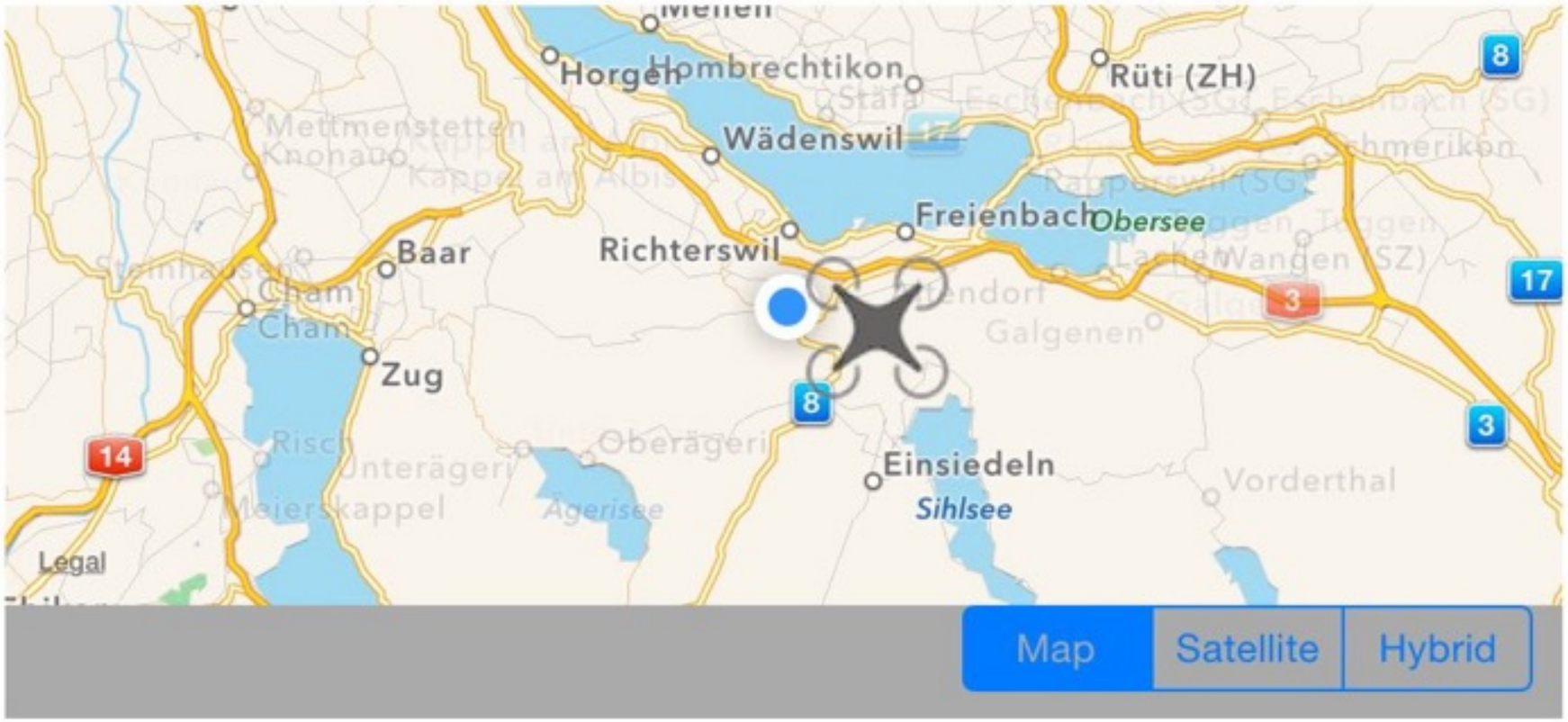
```objc
@interface DJIMCSystemState : NSObject
@property(nonatomic, readonly) int satelliteCount;
@property(nonatomic, readonly) CLLocationCoordinate2D homeLocation;
@property(nonatomic, readonly) CLLocationCoordinate2D droneLocation;
@property(nonatomic, readonly) float velocityX;
@property(nonatomic, readonly) float velocityY;
@property(nonatomic, readonly) float velocityZ;
@property(nonatomic, readonly) float altitude;
@property(nonatomic, readonly) int powerLevel;
@property(nonatomic, readonly) BOOL isFlying;
@property(nonatomic, readonly) DJIMainControllerFlightMode flightMode;
@property(nonatomic, readonly) DJIMainControllerNoFlyStatus noFlyStatus;
@property(nonatomic, readonly) CLLocationCoordinate2D noFlyZoneCenter;
@property(nonatomic, readonly) int noFlyZoneRadius;
@property(nonatomic, readonly) DJIMCSmartGoHomeData* smartGoHomeData;
@property(nonatomic, readonly) DJIGpsSignalLevel gpsSignalLevel;
@property(nonatomic, readonly) BOOL isFailsafe;
@property(nonatomic, readonly) BOOL isCompassError;
@property(nonatomic, readonly) BOOL isVisionWorking;
@property(nonatomic, readonly) BOOL isMotorWorking;
```

```objc
DJIGroundStation* groundStation = (DJIGroundStation*)mainController;
groundStation.groundStationDelegate = self;


-(void) uploadGroundStationTask:(DJIGroundStationTask*)task;
-(void) startGroundStationTask;
-(void) stopGroundStationTask;
-(void) pauseGroundStationTask;
-(void) continueGroundStationTask;
-(void) gohome;
-(BOOL) setAircraftPitchSpeed:(int)pitchSpeed;
-(BOOL) setAircraftYawSpeed:(int)yawSpeed;
```

```objc
@protocol GroundStationDelegate <NSObject>
-(void) groundStation:(id<DJIGroundStation>)gs
 didUpdateFlyingInformation:(DJIGroundStationFlyingInfo*)flyingInfo;



@interface DJIGroundStationFlyingInfo : NSObject
@property(nonatomic, readonly) int satelliteCount;
@property(nonatomic, readonly) CLLocationCoordinate2D homeLocation;
@property(nonatomic, readonly) CLLocationCoordinate2D droneLocation;
@property(nonatomic, readonly) CLLocationCoordinate2D targetWaypointLocation;
@property(nonatomic, readonly) float velocityX;
@property(nonatomic, readonly) float velocityY;
@property(nonatomic, readonly) float velocityZ;
@property(nonatomic, readonly) float altitude;
@property(nonatomic, readonly) float targetAltitude;
@property(nonatomic, readonly) DJIAttitude attitude;
@property(nonatomic, readonly) GroundStationControlMode controlMode;
@property(nonatomic, readonly) GroundStationGpsStatus gpsStatus;
```

```objc
@interface DJIGroundStationTask : NSObject
@property(nonatomic, readonly) int waypointCount;
@property(nonatomic, assign) int startWaypointIndex;
@property(nonatomic, assign) BOOL isLoop;
@property(nonatomic, assign) float maxVerticalVelocity;
@property(nonatomic, assign) float maxHorizontalVelocity;
@property(nonatomic, assign) float maxAngularVelocity;
@property(nonatomic, assign) uint16_t maxExecuteTime;
@property(nonatomic, assign) DJIGSTaskFinishedAction finishedAction;
@property(nonatomic, assign) DJIGSHeadingMode headingMode;
+(id) newTask;
-(void) addWaypoint:(DJIGroundStationWaypoint*)waypoint;
-(void) removeWaypoint:(DJIGroundStationWaypoint*)waypoint;
-(void) removeAllWaypoint;
```

```
float skiMovementLongitude = skiLocation.coordinate.longitude -
     oldSkiLocation.coordinate.longitude;
float skiMovementLattitude = skiLocation.coordinate.latitude -
     oldSkiLocation.coordinate.latitude;
float skiMovementAltitude = skiLocation.altitude - oldSkiLocation.altitude;

float newDroneLocationLongitude = droneLocation.coordinate.longitude +
      skiMovementLongitude;
float newDroneLocationLattitude = droneLocation.coordinate.latitude +
      skiMovementLattitude;
float newdDroneLocationAltitude = droneLocation.altitude + skiMovementAltitude;

wayPoint  = [[DJIGroundStationWaypoint alloc]
     initWithCoordinate:CLLocationCoordinate2DMake(newDroneLocationLattitude,
     newDroneLocationLongitude)];
wayPoint.altitude = newdDroneLocationAltitude;
[groundStationTask removeAllWaypoint];
[groundStationTask addWaypoint:wayPoint];
[groundStation uploadGroundStationTask:groundStationTask];
[groundStation startGroundStationTask];
```



Distance to me
15.6 m

Speed
0 m/s

Satellites:
5

Camera

Follow

Go Home

Drone position
8.705818
47.181728
4.2m

My position
8.705947
47.181620
682.8m

Distance to me
27.2 m

Speed
0 m/s

Satellites:
5

Map

Follow

Go Home

Drone position
8.705812
47.181724
4.4m

My position
8.705900
47.181488
683.8m

The next generation

Sensors

- GPS
- Compass
- Gyroscope
- Air pressure
- Ultrasonic

```
@interface DJIMainController : DJIObject
@property(nonatomic, readonly) NSObject<DJINavigation>* navigationManager;



@protocol DJINavigation <NSObject>
@property(nonatomic, readonly) NSObject<DJIWaypointMission>* waypointMission;
@property(nonatomic, readonly) NSObject<DJIHotPointMission>* hotpointMission;
@property(nonatomic, readonly) NSObject<DJIFollowMeMission>* followMeMission;
```

```
@protocol DJIFollowMeMission <DJINavigationMission>
@property(nonatomic, assign) CLLocationCoordinate2D userCoordinate;
@property(nonatomic, assign) DJIFollowMeHeadingMode headingMode;
-(void) updateUserCoordinate:(CLLocationCoordinate2D)coordinate
        withResult:(DJIExecuteResultBlock)block;
```

```
@protocol DJIHotPointMission <DJINavigationMission>
@property(nonatomic, assign) CLLocationCoordinate2D hotPoint;
@property(nonatomic, assign) float altitude;
@property(nonatomic, assign) float surroundRadius;
@property(nonatomic, assign) BOOL clockwise;
@property(nonatomic, assign) float angularVelocity;
@property(nonatomic, assign) DJIHotPointEntryPoint entryPoint;
@property(nonatomic, assign) DJIHotPointHeadingMode headingMode;
```

Final words

- which is where the mashup parallel comes in

Our own imagination is the limit

Infrared camera.

Send out the drones!

My pulse has gone.

Human transportation?

Is it only the DJI Drone series ?

I guess not…

THANKS!

Catch me at the
TPA booth !

goto; conference

Please

Remember to rate this session

Thank you!

Join the conversation #gotocph