```
tc@box:/mnt/mmcblk0p2/tinypilot/pypilot_git/pypilot$ git diff
diff --git a/pypilot/autopilot.py b/pypilot/autopilot.py
old mode 100644
new mode 100755
index 10d762d..e7ec1fd
--- a/pypilot/autopilot.py
+++ b/pypilot/autopilot.py
@@ -220,6 +220,9 @@ class Autopilot(object):
     self.heading_command = self.Register(HeadingProperty, 'heading_command',
0)
     self.enabled = self.Register(BooleanProperty, 'enabled', False)
     self.lastenabled = False
+    self.trace = self.Register(BooleanProperty, 'trace', False,
persistent=True)
+    self.bell_server = self.Register(EnumProperty, 'bell_server',
'10.10.10.1', ['10.10.10.1', '10.10.10.2', '10.10.10.4', '192.168.178.129'],
persistent=True)
+    self.wlan_mode = self.Register(EnumProperty, 'wlan_mode', 'manual',
['manual', 'home client', 'openplotter client', 'access point'], persistent=True)

     self.preferred_mode = self.Register(Value, 'preferred_mode', 'compass')
     self.lastmode = False
@@ -369,11 +372,13 @@ class Autopilot(object):
       heading = self.heading.value

       # keep same heading if mode changes
-      if self.mode.value != self.lastmode:
+      # marcb 2019-11-02 Disabled because poorly understood
+      if self.mode.value != self.lastmode and False:
           error = self.heading_error.value
           if 'wind' in self.mode.value:
               error = -error
           self.heading_command.set(resolv(heading - error, 180))
+
           self.lastmode = self.mode.value

       # compute heading error
diff --git a/pypilot/nmea.py b/pypilot/nmea.py
index 5957879..02ff06c 100644
--- a/pypilot/nmea.py
+++ b/pypilot/nmea.py
@@ -32,12 +32,20 @@ from signalk.values import *
 from signalk.pipeserver import NonBlockingPipe
 from sensors import source_priority
 import serialprobe
+import os

 import fcntl
 # these are not defined in python module
 TIOCEXCL = 0x540C
 TIOCNXCL = 0x540D

+g_last_waypoint_id = ""
+g_last_track = 0
+g_last_xte = 0
+g_client = ""
+
```

```
+def normalize_heading(heading):
+    return (heading if heading <= 180 else heading - 360)

 # nmea uses a simple xor checksum
 def nmea_cksum(msg):
@@ -154,21 +162,57 @@ def parse_nmea_apb(line):
    ** 14) M = Magnetic, T = True
    ** 15) Checksum
         '''
+
+    global g_last_track
+    global g_last_xte
+    global g_last_waypoint_id
+    global g_last_values
+    global g_client
+
     if line[3:6] != 'APB':
         return False
     try:
         data = line[7:len(line)-3].split(',')
-        mode = 'compass' if data[13] == 'M' else 'gps'
-        track = float(data[12])
-        xte = float(data[2])
-        xte = min(xte, 0.15) # maximum 0.15 miles
-        if data[3] == 'L':
-            xte = -xte
-        return 'apb', {'mode': mode, 'track':  track, 'xte': xte, '**':
line[1:3] == 'GP'}
+        if g_last_values['ap.mode'] == 'gps' and g_last_values['ap.enabled']:
+            track = float(data[12])
+            xte = float(data[2])
+            xte = min(xte, 0.15) # maximum 0.15 miles
+            if data[3] == 'L':
+                xte = -xte
+            waypoint_id = data[9]
+            if waypoint_id != g_last_waypoint_id or g_last_waypoint_id == "":
+                # if waypoint_id changes, switch to compass mode on previous
track+xte, resulting in same commanded heading
+                mode = 'compass'
+                g_client.set('ap.mode', mode) # as not to rely on subtlety
+
+                track = g_last_track
+                xte = g_last_xte
+
+                # Switch ray.py to MODE_WAYPOINT, awaiting confirmation to turn
to next waypoint
+                # That confirmation will come in the form of switching ap.mode
to 'gps' again by ray.py
+                f=open('/tmp/remote', 'w+')
+                if normalize_heading(float(data[12])) >
normalize_heading(g_last_values['ap.heading_command']):
+                    f.write('1000')
+                else:
+                    f.write('1001')
+                f.close()
+            else:
+                mode = 'gps'
```

```
+
+            g_last_track = track
+            g_last_xte = xte
+            g_last_waypoint_id = waypoint_id
+
+            return 'apb', {'mode': mode, 'track':  track, 'xte': xte, '**':
line[1:3] == 'GP'}
+        else:
+            return False
+
+
     except Exception as e:
         print('ex', e)
         return False

+
 nmea_parsers = {'gps': parse_nmea_gps, 'wind': parse_nmea_wind, 'rudder':
parse_nmea_rudder, 'apb': parse_nmea_apb}

 # because serial.readline() is very slow
@@ -471,13 +515,14 @@ class Nmea(object):


 class NmeaBridgeProcess(multiprocessing.Process):
+
     def __init__(self):
         self.pipe, pipe = NonBlockingPipe('nmea pipe', True)
         self.sockets = False
         super(NmeaBridgeProcess, self).__init__(target=self.process,
args=(pipe,))

     def setup_watches(self, watch=True):
-        watchlist = ['gps.source', 'wind.source', 'rudder.source',
'apb.source']
+        watchlist = ['gps.source', 'wind.source', 'rudder.source',
'apb.source', 'ap.mode', 'ap.heading_command', 'ap.enabled']
         for name in watchlist:
             self.client.watch(name, watch)

@@ -548,9 +593,14 @@ class NmeaBridgeProcess(multiprocessing.Process):
         sock.close()

     def client_message(self, name, value):
+        global g_last_values
         self.last_values[name] = value
+        g_last_values = self.last_values

     def process(self, pipe):
+        global g_last_values
+        global g_client
+
         import os
         self.pipe = pipe
         self.sockets = []
@@ -563,6 +613,7 @@ class NmeaBridgeProcess(multiprocessing.Process):
             time.sleep(2)
             try:
```

```
                self.client = SignalKClient(on_con, 'localhost',
autoreconnect=True)
+               g_client = self.client
                break
            except:
                pass
@@ -582,6 +633,7 @@ class NmeaBridgeProcess(multiprocessing.Process):
        server.listen(5)

        self.last_values = {'gps.source' : 'none', 'wind.source' : 'none',
'rudder.source': 'none', 'apb.source': 'none'}
+       g_last_values = self.last_values
        self.addresses = {}
        cnt = 0

diff --git a/pypilot/servo.py b/pypilot/servo.py
index 4ceaab4..e045637 100755
--- a/pypilot/servo.py
+++ b/pypilot/servo.py
@@ -129,9 +129,9 @@ class ServoFlags(Value):
        if self.value & self.BAD_FUSES:
            ret += 'BAD_FUSES '
        if self.value & self.FWD_FAULT:
-           ret += 'FWD_FAULT '
+           ret += 'PORT_FAULT '
        if self.value & self.REV_FAULT:
-           ret += 'REV_FAULT '
+           ret += 'STBD_FAULT '
        if self.value & self.DRIVER_TIMEOUT:
            ret += 'DRIVER_TIMEOUT '
        if self.value & self.SATURATED:
```