



pypilot_mfd

Official pypilot MFD User Manual

Sean D'Epagnier

GPLv3

Table of contents

1. Introduction	3
1.1 Motivation	3
1.2 Philisophy	3
2. Hardware and Installation	3
2.1 Mounting Instructions	3
2.2 Electrical Connections Overview	3
2.3 Maintainance	3
2.4 Mechanical Installation	4
2.5 Suggested sensors	4
2.6 Power off	4
3. Usage	4
3.1 Keypad	4
3.2 Menu	4
4. Web Configuration	4
4.1 WIFI Network Configuration	4
4.2 Sensors	4
4.3 Data Configuration	5
4.4 Display Pages	7
4.5 Display Options	7
4.6 Alarms	8
4.7 History plots	8
4.8 Software Updates	8

1. Introduction

Currently the pypilot_mfd documentation is incomplete

1.1 Motivation

The pypilot_mfd project is intended to provide a free and open design for a marine multifunction display. This is primarily for sensors using open standards such as SignalK and from free software instruments to integrate with solutions like pypilot and openplotter.

1.2 Philisophy

pypilot is free software (GPLv3) and as such you can trust it for use as intended and for future use without restrictions. Because the freedom to modify is guarenteed, users from around the world continuously contributed improvements, even small ones to the benefit of all users.

2. Hardware and Installation

The pypilot_mfd has an internal esp32 microcontroller. This processor can receive the ESP-Now packets produced from various wireless weather instruments as well as communicate wirelessly to opencpn, signalk server, pypilot and others.

2.1 Mounting Instructions

Find a good location on the boat to mount the pypilot_mfd display. The enclosure is fully waterproof and can be exposed to the elements however using a cover over the screen can extend its useful life when not in operation especially if mounted in direct sun.

The display can be mounted in any orientation. Internal accelerometers detect the orientation of the screen on powerup. It may be an advantage to mount the screen in portrait or landscape depending on the intended use.

2.2 Electrical Connections Overview

diagram

2.2.1 USB-C data connection

USB-C can be used to power the display as well as provide serial communication with other devices. The power consumption is light (0.4 Watts) and is slightly higher when the backlight is enabled. The baud rate can be selected from the web interface.

2.2.2 rs422 data connection

The MFD also supports a galvanically isolated rs422 data connection. This has 2 differential pairs (4 wires) for transmit and recieve of nmea0183 data. The baud rate can be selected from the web interface

2.3 Maintainance

The MFD does not require much maintainance besides covering it when not in use. It is recommended to mount it so that the usb and other wire connections are protected from the elements or sealed.

2.4 Mechanical Installation

2.5 Suggested sensors

2.5.1 Wind sensors

2.5.2 Temperature Pressure Sensors

2.5.3 Outdoor weather sensors

2.5.4 Depth/Water speed sensors

2.6 Power off

The pypilot_mfd uses flash memory and therefore should be able to always poweroff simply by removing power. Should the flash become corrupted it is possible that certain settings would be lost.

3. Usage

3.1 Keypad

The MFD has a simple keypad with 4 keys

- left and right arrows page through different displays. You can enable/disable pages from the web interfaces
- hamburger key toggles through different time scales on history displays.
- power key toggles display on/off

3.2 Menu

To enter the menu, hold the menu key for 1-2 seconds. The menu key selects items in the menu, but holding it can also exit the menu.

< picture of main menu >

3.2.1 Alarms

4. Web Configuration

holding the left and right keys for 3 seconds will enable the access point ensuring you can connect with a device to access the web configuration if the wifi settings are wrong or they cannot connect

4.1 WIFI Network Configuration

4.2 Sensors

pypilot_mfd can receive from wireless weather sensors and display the data directly on screen. It can also relay the received wireless sensor data over USB, rs422 (nmea0183) or wifi to other systems. The

4.2.1 Wind Sensors

Multiple wind sensors can be configured to be recieved. This allows the use of primary, secondary and the ability ignore nearby boats using similar sensors. It is also possible to configure wind sensors as port and starboard. This is often useful if sensors are mounted on either side of the boat to automatically switch to the sensor in clear air.

The sensors

4.2.2 Water Speed Sensors

4.2.3 Temperature and Air Pressure Sensors

4.2.4 UV and lightning Sensors

4.3 Data Configuration

4.3.1 USB data (NMEA0183)

4.3.2 rs422 data (NMEA0183)

4.3.3 WIFI data

The internal esp32 can only operate on one wifi channel at a time. For this reason, it is recommended that if you have an access point on 2.4ghz with significant bandwidth, to ensure it is on a different channel than the one the MFD and wireless sensors are using.

This could unfortunately make it more difficult to interface all of the sensors

4.3.4 Interfacing with pypilot

When interfacing with pypilot it is both possible to communicate directly using the pypilot protocol as well as with nmea data. It is also possible to indirectly interface via signalk.

For

4.3.5 Interfacing with OpenCPN

OpenCPN can directly interface (via nmea0183 sentences) or indirectly via SignalK. To directly interface, put the MFD into tcp nmea server mode with a specific port, and make a tcp nmea connection to this address:port in OpenCPN. Conversely it is possible for opencpn to listen on a particular port and to make the MFD act as a tcp client.

4.3.6 Interfacing via NMEA0183

NMEA0183 is an old open standard for marine data communication. It can still serve an important purpose as the data is generally fairly concise, readable and supported by a wide variety of equipment. The MFD supports this format for input and output on both USB and rs422 ports as well as via wifi.

Supported sentences for receiving NMEA0183

- \$xxMWV - wind speed and direction apparent/true
- \$xxMWD - wind speed and direction true
- \$xxVWR - wind speed and direction apparent
- \$xxVWT - wind speed and direction true
- \$xxMDA - air pressure, air temperature, water temperature
- \$xxMTA - air temperature
- \$xxRMB - route following information
- \$xxRMC - GPS time, position, speed and heading
- \$xxVHW - Water speed
- \$xxHDM - Compass heading
- \$xxXDR - Pitch and Roll
- \$xxROT - Rate of Turn
- \$xxRSA - Rudder Angle

- \$xxAPB - Autopilot Bearing
- !xxVDM - AIS ships

Supported sentences for sending NMEA0183

If wireless sensors are connected, the data can be relayed and send out via NMEA0183. Other data received is not relayed.

- \$QYMVW - Apparent Wind Speed and Direction
- \$QYMDA - Barometric Pressure and Relative Humidity
- \$QYMTA - Air Temperature
- \$QYVHW - Water Speed
- \$QYDBT - Depth
- \$QYMTW - Waterm Temperature

4.3.7 Interfacing with SignalK

pyplot_mfd can send and receive SignalK data. The signalk server requires access requests to be granted to receive data

Supported keys for receiving data from SignalK

- environment.wind.speedApparent
- environment.wind.angleApparent
- environment.wind.angleTrue
- environment.wind.speedTrue
- environment.outside.temperature
- environment.outside.pressure
- depth.belowSurface
- navigation.speedOverGroundTrue
- steering.rudderAngle
- steering.autopilot.target.headingTrue
- navigation.headingMagnetic
- navigation.rateOfTurn
- navigation.speedThroughWater

Supported keys for sending data to Signak

If wireless sensors are connected, the data can be relayed and send out via SignalK. Other data received is not relayed

Wireless wind sensors:

- environment.wind.angleApparent
- environment.wind.speedApparent

Wireless pressure/humidity sensors:

- environment.outside.temperature
- environment.outside.pressure
- environment.outside.relativeHumidity

Wireless depth/speed sensors:

- navigation.speedThroughWater
- environment.depth.belowSurface
- environment.water.temperature

For other sensors either no key is supported

4.4 Display Pages

4.5 Display Options

There are several options that can be set to configure the displayed information.

- use360 - This will display gauges from 0-180 in port and starboard rather than 0-360

These options select imperial units rather than metric

- usefahrenheit - display temperatures in fahrenheit (F) rather than celcius (C)
- useinHg - displays pressure using inches mercury instead of millibars
- usedepthft - displays depth in feet rather than meters

The latitude and longitude can be displayed in 3 basic formats:

- degrees - eg: 78.234410 N 32.932330 E
- minutes - eg: 78 14.065' N 32 55.940' E
- seconds - eg: 78 14 3.60" N 32 55' 56.39" E

The LCD contrast and backlight brightness can be set here.

It is also possible to set the rotation of the display. Normally the accelerometers detect the rotation, but in the event they do not, you can change the rotation. The unit should be power cycled to utilize the new orientation.

The mirror should normally be set with a factory jumper, but in the event the display shows everything in a mirror image, this option can be toggled.

- powerdown

The powerdown option specifies if the power button should simply turn off the display and backlight, but continue to receive from wireless sensors, or actually powerdown the microprocessor and go into a sleep mode using the lowest amount of power consumption (microamps)

4.5.1 Pages

The MFD has various pre-programmed display pages. Most users will only want to enable the pages that can display information that is useful to them. If you have multiple MFD displays it might be useful to program different ones to different pages. If for example you have programmed 4 pages, then paging through them with the arrow buttons will not page through information that is not helpful or available.

4.5.2 Automatic Configuration

Automatic configuration button will scan the available incoming sensors, and enable relevant display pages. This should be performed

4.6 Alarms

Various alarms can be set and configured from the web interface rather than the menu

4.7 History plots

Historical data from the sensors can also be viewed on the web browser

4.8 Software Updates

It is possible to update the software either via usb or wirelessly.

4.8.1 OTA Update

stands for Over The Air Updates. It is possible either to select a firmware file in the web configuration to update the software, or if the device is connected on a network with internet access, it can automatically initiate an update from the pypilot.org website.

4.8.2 USB Update

Updatting via usb is more fool-proof and should work in all conditions. To initiate the bootloader, hold the power button on the keypad while applying power to the mfd. The screen should not display anything and the backlight will not turn on. At this point the device is in the bootloader and the commandline tool "esptool.py" can be used to flash new data to the device.