WEGMATT_{LLC} dAISy HAT AIS Receiver

Contents

1   Disclaimer

dAISy HAT is a reliable, dual-channel AIS Receiver. However, under no circumstances it should be solely relied on for collision avoidance or navigation. It's the user's responsibility to use the product prudently. Neither Wegmatt LLC nor its dealers accept responsibility or liability to the product user or their estate for any accident, loss, injury or damage whatsoever arising out of the use of this product.

2   Configuring the Raspberry Pi for the dAISy HAT

The dAISy HAT communicates with the Raspberry Pi through the serial port broken out one the Raspberry Pi's expansion header. Out-of-the-box, the Raspberry Pi uses these pins for other purposes like a serial terminal or Bluetooth.

Reconfiguring serial ports is straight forward with recent versions of Raspberry Pi OS (since ca. late 2016) and OpenPlotter. Configuration can be more complex with old version of Raspberry Pi OS or OpenPlotter, especially on the Raspberry 3.

For beginners, we recommend starting with a new image of Raspberry Pi OS or OpenPlotter.

2.1   OpenPlotter

To use the dAISy HAT with OpenPlotter, follow the instructions in the official documentation. The dAISy HAT is covered in the section Serial > Connecting the dAISy HAT:
https://openplotter.readthedocs.io/latest/serial/connectingdAISy.html

2.2   Raspberry Pi OS and Raspbian

If you are running Raspberry Pi OS or Raspbian without OpenPlotter, the serial port that connects to the dAISy HAT needs to be manually configured. The steps have changed over time, so make sure to follow the instructions applicable to the version of Raspberry Pi OS you are running.

If you can't get it working, check our website (https://wegmatt.com) for updated documentation. If you already have the latest version of the documentation and still can't get it to work, don't hesitate to contact us by email (daisy@wegmatt.com).

2.2.1   Raspberry Pi OS Bookworm or newer

If you are using the dAISy HAT on a Raspberry Pi with the latest version of Raspberry Pi OS, use the `raspi-config` tool to configure the serial port through which the Raspberry Pi talks to the dAISy HAT.

Open a terminal window on your Raspberry Pi and type:

```
sudo raspi-config
```

This will launch the Raspberry Pi Software Configuration Tool. Then enable the serial port by navigating through the following steps:

```
3 Interface Options -> I5 Serial Port
```
   -> `<No>` (no login shell on serial) -> `<Yes>` (enable serial hardware) -> `<Ok>`
   -> `<Finish>` -> `<Yes>` (reboot)

After completing the reboot, AIS data is available using the following parameters:

Raspberry Pi	Pi 4, 3, Zero	Pi 5
Serial port	/dev/serial0	/dev/ttyAMA0
Baud rate	38400	38400
Data bits	8	8
Parity	None	None
Stop bits	1	1
Flow control	None	None

Use this information to configure OpenCPN or other programs. Keep in mind, that only one application at a time can consume serial data.

You can verify the serial connection with a serial terminal program like for example *screen*. You may have to install *screen* first by running:

```
sudo apt-get install screen
```

With *screen* installed, connect to the dAISy HAT on a Raspberry Pi 5 with:

```
screen /dev/ttyAMA0 38400
```

or for Raspberry 4:

```
screen /dev/serial0 38400
```

If the serial port is properly configured, pressing ESC will bring up the configuration and debug menu. Pressing ESC again will return dAISy into receive mode. Exit *screen* by pressing CTRL-A, release the keys, then press K (kill session) and then Y to confirm.

2.2.2   Raspian or Raspberry Pi OS Buster or older

If you are using the dAISy HAT on an older version of Rasbberry Pi OS or Raspian, use the uart_control shell script by Ilker Temir to configure the serial port.

Open a terminal window on your Raspberry Pi and execute the following commands:

```
wget https://github.com/itemir/rpi_boat_utils/raw/master/uart_control/uart_control
chmod +x ./uart_control
sudo ./uart_control gpio
sudo reboot now
```

This will download the script, make it executable, run it to reconfigure the serial port for use with a HAT, and rebooting the Raspberry Pi to make the new configuration active.

After completing the reboot, AIS data is available using the following parameters:

Serial port	/dev/serial0
Baud rate	38400
Data bits	8
Parity	None
Stop bits	1
Flow control	None

Use this information to configure OpenCPN or other programs. Keep in mind, that only one application at a time can consume serial data.

You can verify the serial connection with a serial terminal program like for example *screen*. You may have to install *screen* first by running:

```
sudo apt-get install screen
```

With *screen* installed, connect to the dAISy HAT with

```
screen /dev/serial0 38400
```

If the serial port is properly configured, pressing ESC will bring up the configuration and debug menu. Pressing ESC again will return dAISy into receive mode. Exit *screen* by pressing CTRL-A, release the keys, then press K (kill session) and then Y to confirm.

2.3   Configuration in OpenCPN

OpenCPN is a free chart plotter and navigation software. Among many other things, OpenCPN supports input from AIS receivers to display the position of ships on its charts.

You can learn more about OpenCPN on https://opencpn.org

If you are using OpenCPN as part of OpenPlotter, you can ignore this chapter. OpenPlotter already set everything up for you.

There are several tutorials on the web on running OpenCPN on a Raspberry Pi. The consensus is, that you should only attempt this on a Raspberry Pi 3 or 4 with an up-to-date version of Raspberry Pi OS. The official instructions can be found here for Raspberry Pi 3:

https://opencpn.org/wiki/dokuwiki/doku.php?id=opencpn:opencpn_user_manual:getting_started:opencpn_installation:raspberrypi_rpi2

and Raspberry Pi 4:

https://opencpn.org/wiki/dokuwiki/doku.php?id=opencpn:opencpn_user_manual:getting_started:opencpn_installation:rpi4

Alternatively, users have reported good results by following this procedure:

https://www.agurney.com/raspberry-pi/pi-chart

Once OpenCPN is running on the Raspberry Pi, configure it for the dAISy HAT. Go to the `Options` dialog, select the `Connections` icon and add a new connection. Set `Properties` to `Serial`. Enter or select the `Dataport` under which dAISy presents itself (see chapters above to determine the correct port name). Set `Baudrate` to `38400`. Leave everything else as preset by OpenCPN.
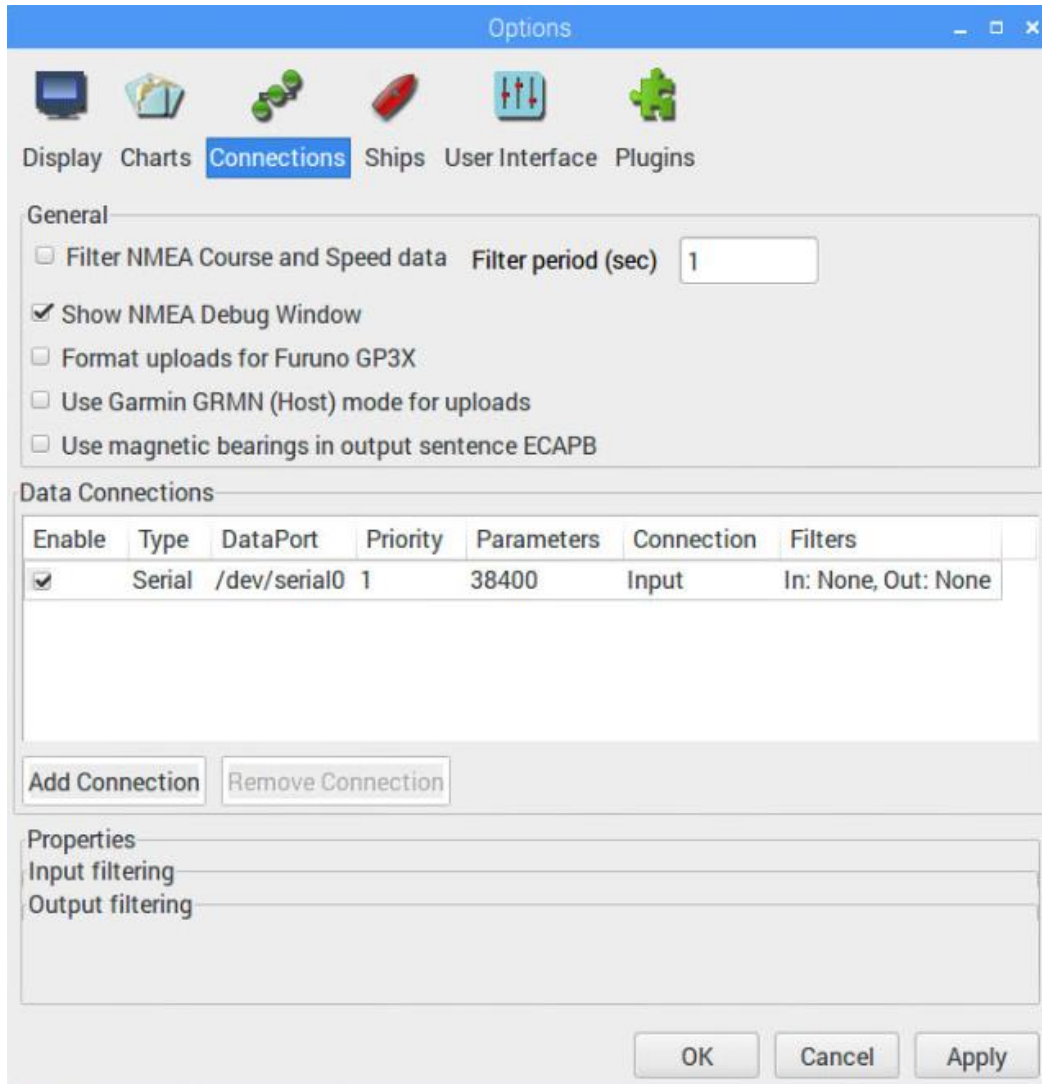
Configuration of a serial connection for dAISy in OpenCPN

2.4   Connecting dAISy with other software

dAISy presents itself to computers as a serial port. See the configuration chapters above about installation to learn how to determine the serial port.

The serial communication parameters of dAISy are:

Baud rate	38400
Data bits	8
Parity	None
Stop bits	1
Flow control	None

Depending on the software dAISy is used with, some or all parameters must be configured. For details, please refer to the instruction manual of the respective application.

3   Using the dAISy HAT

Using the dAISy HAT is as simple as connecting it with an antenna and a PC and powering up your AIS capable application of choice. When plugged in, dAISy automatically starts to listen for AIS transmissions.

3.1   Interpreting the status LEDs

The dAISy HAT has two two-color status LED, giving us some indication about how it's feeling. Each LED represents one of the two AIS channels.

Short GREEN flash every 5 seconds

dAISy is well and alive and in receive mode.

While in receive mode, the mode dAISy starts in when powered up, it will very briefly blink green every 5 seconds to indicate that everything is ok.

Long GREEN flash

Yay! dAISy received a valid AIS packet.

For every packet received and passing all error checks, the LED will turn green for ¼ second. At the same time, dAISy will send the received message to the Raspberry Pi.

Long RED flash

dAISy received an invalid packet.

dAISy caught the start of a package, but the received data did not pass other sanity checks like encoding standard or checksum. It might be a ship just out of range, interference from another radio source, or just random noise.

Don't panic, you will see a lot of red even under optimal conditions.

RED turned on permanently

dAISy failed to initialize the radio IC.

Shut down the Raspberry Pi and remove power. Plug it in again and see if this resolves the problem. If the condition persists, there might a hardware issue with your device. Please contact us.

Fast regular blinking RED

dAISy encountered an unexpected system error.

Shut down the Raspberry Pi and remove power. Plug it in again and see if this resolves the problem. If the condition persists, there might a hardware issue with your device. Please contact us.

No LED activity

The dAISy HAT may not receive power. Shutdown the Raspberry Pi and remove power. Then visually inspect the expansion header and make sure that the dAISy HAT is properly seated on the Raspberry Pi.

Another possibility dAISy is not in receive mode because you were using the configuration menu.

Connect to dAISy with a serial terminal and press ESC. A message should tell you that it entered AIS receive mode and the green LED should start to blink every 5 seconds.

Another possibility is, that dAISy failed to boot or crashed. In this case, shut down the Raspberry Pi and remove power. Plug it in again and see if this resolves the problem.

If the condition persists, there might a hardware issue with your device. Please contact us.

3.2   Configuration and debug menu

When connecting to dAISy with a serial terminal (for example *screen*), pressing ESC will bring up a menu with a few configuration and debug options.

```
dAISy v5.11 - dAISy HAT (5232) (C)2014-2022 Adrian Studer

Radio 1: 161.975MHz AIS1       Radio 2: 162.025MHz AIS2       Format: A/B
Serial 2: 9600 8N1, NMEA in     Debug: off
Reset GPIO18: on         Sleep GPIO17: off       Sleep Serial IO: on

Configure:
0-3     Set Serial 2 to 4800, 9600 or 38400 baud (1-3) or disable port (0)
S       Toggle function of Serial 2 as NMEA output or NMEA input
C       Configure AIS channels
R       Toggle control of reset via Raspberry Pi GPIO18
L       Toggle control of sleep via Raspberry Pi GPIO17
I       Toggle control of sleep via Serial IO
#       Save current settings

Debug:
D       Toggle debug messages on/off
N       Read current RSSI (indication for noise)
V       Show statistics for valid messages received
T       Output NMEA test messages
B       Launch bootloader to update firmware

ESC     Return to AIS receive mode
>
```

The most used options are:

0-3: Controls the Serial 2 header on the dAISy HAT. Pressing 1 enables the serial port with a speed of 4800 baud, 2 is for 9600 baud, and 3 is for 38400 baud. Pressing 0 turns Serial 2 off.

S: Selects the function of the Serial 2 header. There are two modes:

1. **NMEA output**: Output AIS messages, for example to a Bluetooth module, RS-422/NMEA0183 adapter, or a data logger.
2. **NMEA input**: Act as an input for NMEA messages, for example from a GPS module. All data received through Serial 2 will be forwarded to the Raspberry Pi.

D: Toggles debug messages. When enabled and in AIS receive mode, signal strength and other information about valid and invalid AIS packets will be displayed. It is normal to receive a lot of broken

packages as there always will be radio noise and ships just out of range. With some experience, you can use the debug messages to analyze and improve reception issues.

#: Save current settings (serial, channel, reset and debug) as startup defaults. This is particularly useful when you want to persist the settings for the auxiliary serial output.

ESC: Resume receiving AIS messages.

Other options:

N: Keep pressed to get an idea of the current noise level. -120 to -100 dBm is normal. dAISy will only be able to receive valid packets roughly 15 dBm above that level, consistent reception 25+ dBm above noise.

V: If valid packages were received, this gives a summary of the signal strength at which AIS messages arrived (min, max, average).

T: Enter test mode. In this mode, dAISy outputs a test message every 5 seconds. You can enter your own message or have it output a preprogrammed AIS message. This mode is useful when you want to test dAISy but do not have AIS reception at your current location. You can exit test mode by pressing **ESC** or power-cycling dAISy.

C: Configure the radios of the dAISy HAT to listen to specific AIS channels. The dAISy HAT has two radios, which by default are configured to listen on channel A at 161.975 MHz and channel B at 162.025 MHz respectively. The dAISy HAT also supports the less common "Long range AIS" channels at 156.775 and 156.825 MHz.

R: Enable or disable Reset via GPIO. When enabled, the Raspberry Pi can reset the dAISy HAT through GPIO18 (pin 12 on the Raspberry Pi expansion header). To reset the HAT, set GPIO18 high for 150ms or longer, then set GPIO18 back to low.

L: Enable or disable low power sleep via GPIO. When enabled, the Raspberry Pi can put the dAISy AIS Receiver into a low power sleep state by setting GPIO17 high. In sleep state, the radios are turned off and the AIS receiver pauses operation. Current consumption drops from approximately 35 mA to less than 0.8 mA. Operation is resumed within 100ms from the time GPIO17 goes back to low. This feature requires firmware version 5.11 or newer.

I: Enable or disable low power sleep via Serial IO. When enabled, the dAISy AIS Receiver can be put into a low power sleep state by applying 3.3V to the IO pad on the Serial 1 or Serial 2 ports. In sleep state, the radios are turned off and the AIS receiver pauses operation. Current consumption drops from approximately 35 mA to less than 0.8 mA. Operation is resumed within 100ms from the time Serial IO goes back to 0V. This feature requires firmware version 5.11 or newer.

B: Enter boot loader mode for firmware updates. dAISy exits this mode through power-cycling, after a 20 second timeout without further activity, or after completing firmware upgrade.

4 Using the breakout connections

The dAISy HAT AIS Receiver breaks out several interfaces as unpopulated footprints for 0.1" headers.

All digital interfaces are 3.3V only.

4.1 Raspberry Pi I/O header

The dAISy HAT uses the following connections for its operation.

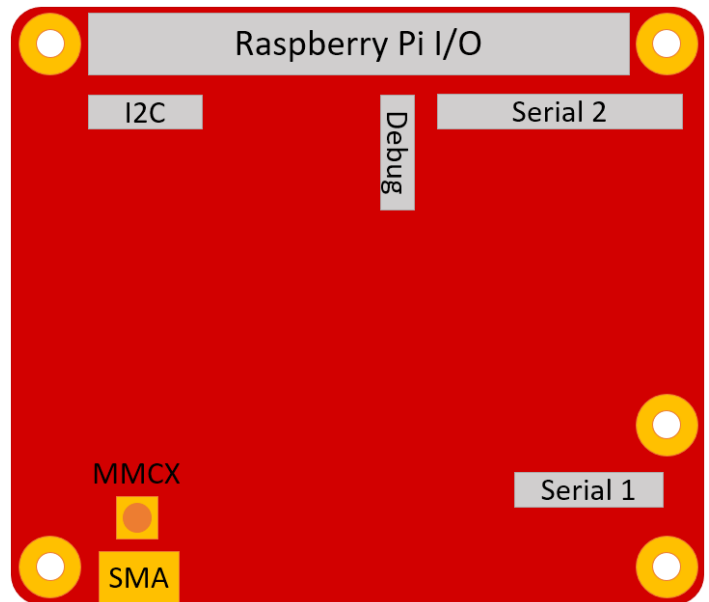| Raspberry Pi	dAISy
Pin 2, 5V	5V power, the AIS receiver will draw less than 35mA in operation
Pin 6, GND	Connected to the GND plane of the HAT
Pin 8, UART0_TXD	Serial data from Raspberry Pi to dAISy, 3.3V signal level, 38400 bps
Pin 10, UART0_RXD	Serial data from dAISy to Raspberry Pi, 3.3V signal level, 38400 bps
Pin 11, GPIO17	Input, 3.3V, normally low, set high to enter sleep mode, see section 3.2
Pin 12, GPIO18	Input, 3.3V, normally low, pulse high for 150ms to reset AIS receiver, see section 3.2

4.2 Serial 1 header

The header Serial 1 exposes the primary serial interface of the dAISy AIS Receiver. It is connected in parallel with the serial interface of the Raspberry Pi and offers the same data and functionality.

Serial 1 is always active and runs at 38400 baud. It can be used to run the dAISy HAT standalone, without being connected to a Raspberry Pi.

When connected to a Raspberry Pi, Serial 1 should only be used to receive the same data as the Raspberry Pi. Do not send data through the RX pin while the HAT is connected to a Raspberry Pi.

The 5V pad is connected to the 5V power rail of the Raspberry Pi. When the dAISy HAT is not connected to the Raspberry Pi, the 5V pad can be used to power the AIS Receiver. When running standalone, the dAISy HAT will draw a maximum current of 40mA in receive mode.

dAISy Serial 1	dAISy MCU	Raspberry Pi	Comment
5V	5V to 3.3V LDO	Pin 2, 5V	5V power rail of Raspberry Pi and HAT. Can be used to power HAT standalone.
GND	GND	GND	Connected to the GND plane of the HAT
TX	Serial 1 TXD	Pin 10, UART0_RXD	Serial data out, 3.3V signal level, 38400 bps
RX	Serial 1 RXD	Pin 8, UART0_TXD	Serial data in, 3.3V signal level, 38400 bps. Don't use when HAT is connected to Raspberry Pi.
IO	GPIO	-	I/O, 3.3V signal level, configurable to enter sleep mode, see section 3.2

4.3   Serial 2 header

The header Serial 2 exposes the secondary serial interface of the dAISy AIS Receiver. This serial port is controlled by dAISy's MCU and not connected to the Raspberry Pi.

Speed and function of Serial 2 are configured in the debug menu. It can operate as secondary output or as NMEA input. The pinout is compatible with the Adafruit Ultimate GPS module.

The 5V pad is connected to the 5V power rail of the Raspberry Pi. When the dAISy HAT is not connected to a Raspberry Pi, the 5V pad can be used to power the AIS Receiver. When running standalone, the dAISy HAT will draw a maximum current of 40mA in receive mode.

dAISy Serial 2	dAISy MCU	Raspberry Pi	Comment
5V	5V to 3.3V LDO	Pin 2, 5V	5V power rail of Raspberry Pi and HAT. Can be used to power HAT standalone.
GND	GND	GND	Connected to GND plane of dAISy HAT
TX	Serial 2 TXD	-	Serial data out, 3.3V signal level, configurable speed.
RX	Serial 2 RXD	-	Serial data in, 3.3V signal level, configurable speed.
IO	GPIO	-	I/O, 3.3V signal level, configurable to enter sleep mode, see section 3.2

4.4   I2C header

This header breaks out the I2C interface of the Raspberry Pi. This allows applications running on Raspberry Pi to communicate with I2C sensors. For example, OpenPlotter supports temperature, pressure, humidity and light sensors.

The I2C interface is independent of the AIS Receiver and the pads are directly connected to the Raspberry Pi expansion header. For more information about the I2C interface, refer to the documentation of the Raspberry Pi.

dAISy I2C	Raspberry Pi	Comment
3V3	Pin 1, 3V3	3.3V power from Raspberry Pi, 50mA max current
SDA	Pin 3, SDA1 / GPIO2	I2C data, 3.3V signal level
SCL	Pin 5, SCL1 / GPIO3	I2C clock, 3.3V signal level
GND	GND	Connected to GND plane of dAISy HAT

4.5   Debug header

The debug header breaks out the debug and programming interface of the MCU controlling the AIS receiver. The MCU is a Texas Instrument MSP430F5232.

The debug header also gives access to the 3.3V power rail of the AIS receiver, which can be used to power peripherals. Maximum current available is 150mA. Recommended current for continuous operation is 50mA. Back-powering the AIS receiver through the 3.3V pad is possible but may degrade RF performance.

dAISy Debug	dAISy MCU	Raspberry Pi	Comment
GND	GND	GND	Connected to GND plane of dAISy HAT
TST	TEST / SBWTCK	-	TI SBW programming interface
RST	RST / SBWTDIO	-	TI SBW programming interface, reset
3V3	3V3	-	Connected to 3.3V rail of dAISy HAT. 150mA max current, 50mA continuous.

4.6   RF connectors

The edge-mount antenna connector of the dAISy HAT is female SMA, standard polarity. There is also an unpopulated footprint for a surface mount vertical MMCX connector, for example Molex 73415-2061.

The dAISy HAT is matched for an antenna with an impedance of 50 ohms, which is standard for marine VHF communication.

5   Tips for good AIS reception

dAISy is not the most sensitive AIS receiver out there. But you we can help it to excel with some basic measures.

5.1   Antenna

If everything else is optimal, almost any antenna tuned to the VHF band will do. I used a simple wire cut to length for my first prototypes. However, a good antenna will greatly improve results.

AIS messages are broadcast on two channels around 162 MHz, which is in the maritime VHF band. This means that any antenna sold as "*marine VHF antenna*" will be a good start.

The short VHF "rubber duck" antennas are cheap and will work at short distance. As long as you check whether 162 MHz is inside the supported frequency range, you can even ignore the "marine" and save a few bucks. However, the more broadband the antenna, the worse the reception quality.

A step up are VHF whip antennas. These are steel rods about 1 meter (3.5ft) long. While bulky, these don't cost much more than the "rubber duck" but provide superior results. Personally, I use the TRAM 1600-HC which cost around US $40.

The AIS antenna should be separated as far as possible from the voice VHF antenna to avoid unnecessary interference. The best separation is achieved by vertically separating the antennas or at least mounting them on opposite sides of a vessel.

To avoid damaging dAISy's receiver, the antenna should be mounted at least 3 meters away from, or at least out of the transmitting beam, of high-power transmitters such as radars or other VHF antenna installations.

IMPORTANT: Do NOT directly connect the dAISy HAT to the same antenna as your VHF radio, this will damage the AIS receiver. To share an existing antenna with a VHF radio, use an ACTIVE splitter. Active splitters protect the AIS receiver by automatically disconnecting it when the VHF radio is transmitting. Several customers reported good results with the inexpensive Glomex RA201 VHF/AIS/Radio Splitter.

5.2   Location, location, location

Good AIS reception requires line of sight.

A few buildings and trees between you and your targets aren't ideal, but you will still be able to catch a few messages. Hills and mountains however are almost certain showstoppers. A clear view of the water and your targets is ideal.

In my experience the most important factor is the height of the antenna above sea-level (or lake-, river-, wherever-level the ships are). Sitting at the beach or on the deck of a boat will work. Mounting the antenna on the mast of your boat or standing on the observation deck of a cruise ship is better. A hill overlooking the harbor is great. Climbing that 2000m mountain that towers over the coast is AIS heaven.

In a nutshell: The farther you can see, the better.

5.3   Radio noise

The main weakness of dAISy, and probably most low-cost and SDR-based AIS receivers, is susceptibility to radio noise. The effect of noise is reduced range and few successfully received packets. Unfortunately, there's not much we can do except finding a spot that's less impacted by noise. Outdoors is better than indoors (computers, power-supplies, wireless networks, ...). Countryside is better than down-town (taxi radios).

dAISy has an integrated 12 MHz wide bandpass filter to reduce the impact of radio noise outside the 162 MHz target frequency. However, to keep cost and size down, the integrated filter is rather simple. A very narrow-band antenna or an external bandpass filter can improve reception. Please report back any results if you go down this route.

6 Troubleshooting

Problem	Solution
No activity of the status LEDs	Verify that the dAISy HAT is properly seated.
Device in reception mode, but not valid AIS messages received (no ¼ second green flashes)	Verify the antenna connection and location. Good AIS reception requires line-of-sight, verify your setup from a location that is certain to have traffic, for example near a harbor.
Device indicates valid AIS messages (¼ second green flashes), but no messages received on the Raspberry Pi.	Verify that the serial port shows up, if not verify the hardware configuration. Verify the serial configuration of your software. Verify the serial communication with a program like *screen*.
Software receives positions, but fewer than expected and/or with worse range than expected.	Verify the antenna connection and location. Good AIS reception requires line-of-sight. Move the antenna away from sources of electrical noise.
After some time, dAISy indicates a device error (fast blinking red LED)	This should not occur with the dAISy HAT. Power cycle the Raspberry Pi to resolve the issue.

If a problem persists don't hesitate to contact us.

7 Specifications

Power	
Power input ports	Raspberry Pi Header (RPi), Serial 1 & 2, Debug header
Input voltage RPi / Serial 1 / Serial 2	5V
Input voltage Debug	3.3V
Input current RPi / Serial 1 / Serial 2 / Debug	< 35 mA
Input current in sleep mode	< 0.8 mA
Power output ports	Serial 1, Serial 2, Debug header
Output voltage Serial 1 / Serial 2	5V
Output voltage Debug	3.3V
Maximum output current Serial 1 / Serial 2	Limited by external 5V supply
Maximum output current Debug	150mA
Radio	
Independent receivers	2
Frequency bands	161.975MHz, 162.025MHz, 156.775MHz, 156.825MHz
AIS channels	2, freely configurable
Bandpass filter	SAW 156.3 - 162.025MHz pass band
RF connector	SMA female
Input impedance	50Ω
Data I/O	
AIS message encoding	NMEA 0183 (!AIVDM)
Supported AIS message types	1-27
Serial speed RPi / Serial 1	38400 Baud fixed
Serial speed Serial 2	4800, 9600, 38400 Baud configurable
Serial protocol	8 data bits, no parity, 1 stop bit (8N1)
Serial mode RPi / Serial 1	NMEA out fixed
Serial mode Serial 2	NMEA out, NMEA in configurable
Serial signal voltage RPi	3.3V
Serial signal voltage Serial 1 / Serial 2	3.3V (5V tolerant)
Dimensions	
Circuit board without connectors	65mm x 56mm
Circuit board with connectors	65mm x 66mm x 14mm (6mm w/o RPi conn)
Circuit board mounting holes	4x Size: 2.75mm (M2.5) Distance: 58mm x 49mm
See also: https://github.com/raspberrypi/hats/blob/master/hat-board-mechanical.pdf	
Weight	
Product	18g
w/o 40-pin Raspberry Pi header	14g

8   Source code and schematics

Currently dAISy is only partially open source.

The core of dAISy, how the Silicon Labs Si4362 radio IC is programmed to receive AIS, is available on Github. While the project does not include all the functionality of dAISy as sold, we do backport any bug fixes that are relevant to the task of receiving AIS. https://github.com/astuder/dAISy

There is also a long thread on 43oh that documents the journey of dAISy from its humble beginnings. https://forum.43oh.com/topic/4833-potm-daisy-a-simple-ais-receiver/

If you want to experiment with the real code, brick your device with custom firmware or even build your own dAISy from scratch, contact us. Tinkerers welcome!

9   Contact

The best way to reach us is by email: daisy@wegmatt.com

Please don't hesitate to contact us if you need support. We also love to hear about your projects and are happy to share your related blog posts, videos, and pictures on our social media.

You can also reach us on GitHub, and Twitter.

https://wegmatt.com

Wegmatt LLC
6356 138th Ave NE, #212
Redmond, WA 98052
United States of America

Wegmatt LLC
6356 138th Ave NE #212
Redmond, WA 98052
United States of America
https://wegmatt.com

Designed and assembled in USA