

# KduPRO

User's guide and  
technical documentation



**Institut  
de Ciències  
del Mar**



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776480



Please cite this document as:

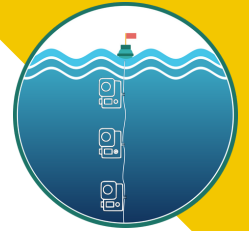
Carlos Rodero, Raul Bardaji, Jaume Piera. 2021. KduPRO. User's guide and technical documentation

doi: <http://doi.org/10.5281/zenodo.5721155>

Special thanks to Xavi Salvador for providing nice pictures of the KduPRO.

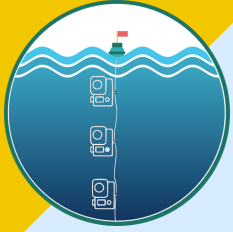


This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776480



# Contents

System description .....	1
Measuring $K_d$ .....	2
Physical meaning .....	3
Reproducibility .....	3
System installation .....	4
Hardware components .....	4
Installation .....	5
Software setup .....	15
System check.....	18
Data gathering .....	19
Data process .....	20
System maintenance .....	21



# System description

The **KduPRO** is a low-cost and Do-It-Yourself moored instrument that measures the diffuse attenuation coefficient ( $K_d$ ), used to assess water transparency.

This parameter is strongly affected by different water quality-related components (such as the presence of phytoplankton, organic matter and sediment concentrations).

## Deployment of the KduPRO

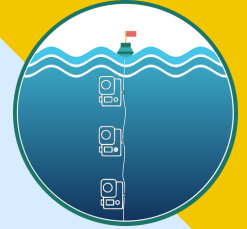


The KduPRO is the modular version of the KduINO, redesigned to obtain accurate  $K_d$  measurements in the first meters of the water column.

Each module contains a light sensor that allows measures in the spectral zone of PAR (Photosynthetically Active Radiation), red, green and blue, and can store data independently.

The depth of each module can be modified according to the requirements of the project or the environment, offering to the user a custom array of sensors.

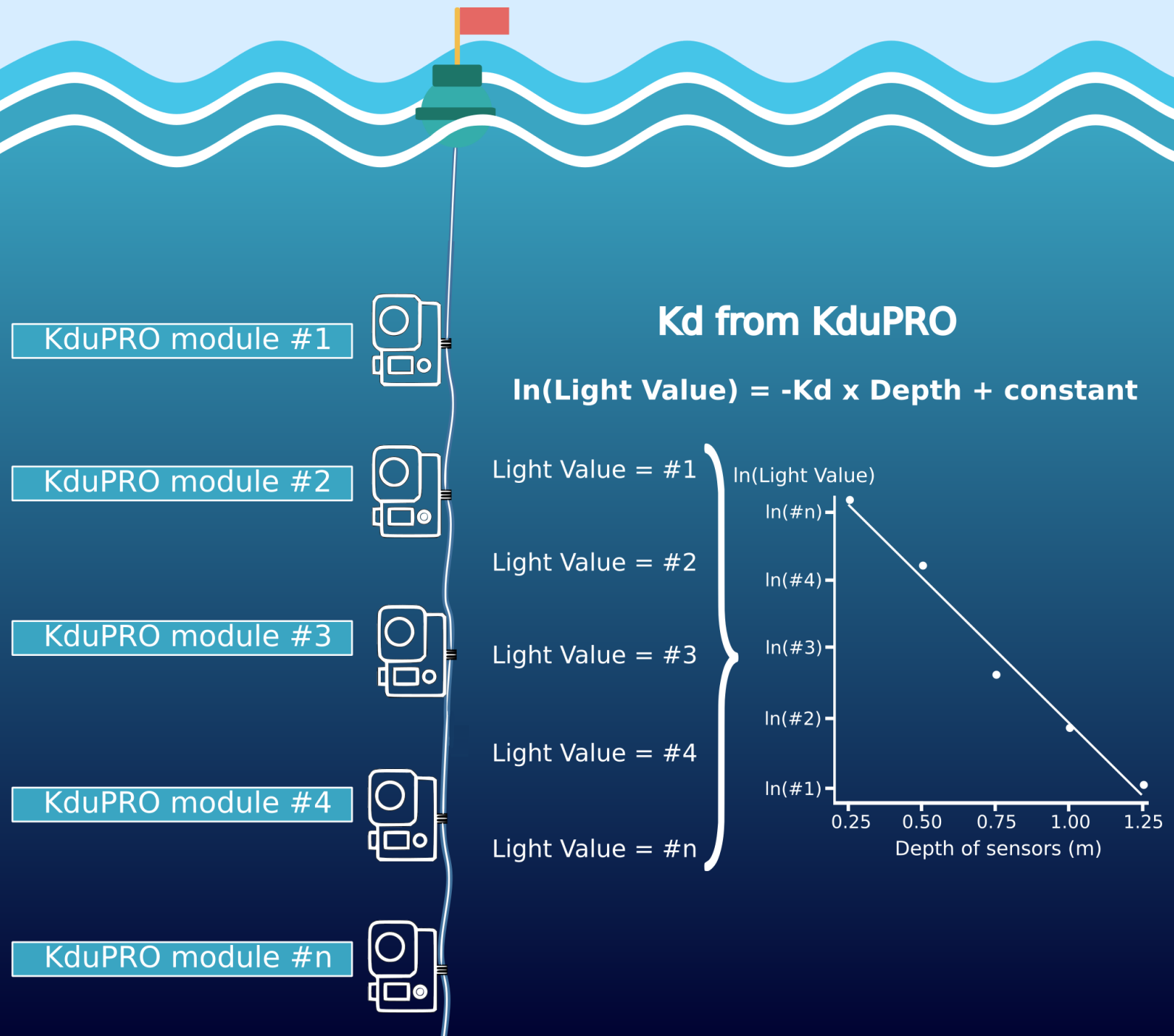
**In this document, we will give general instructions for build and use a KduPRO.**

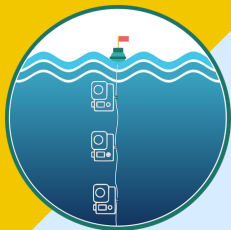


# Measuring Kd

How does **KduPRO** measure the diffuse attenuation coefficient (Kd)?  
Each module of KduPRO is placed at a known depth, and measure the light value by its sensor.

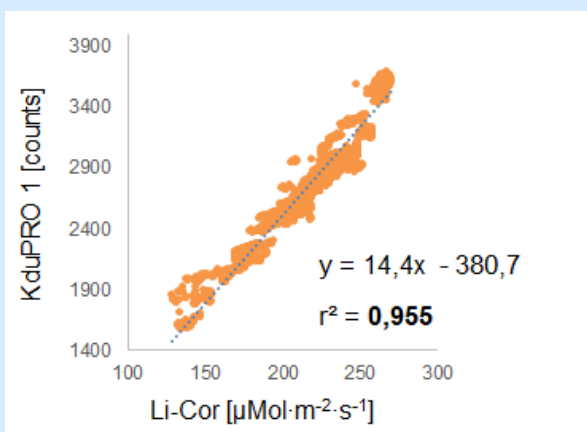
We convert those light values to logarithms and calculates the linear regression. With this parameter, it is obtained the slope, and by means of the negative value of this slope, Kd is obtained.





# Physical meaning

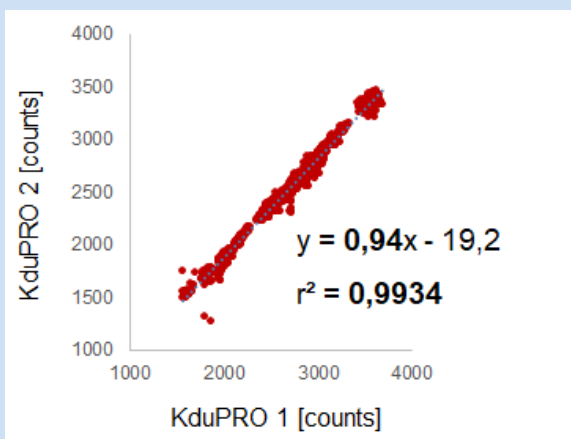
The **KduPRO** has been compared with other reference sensors, obtaining significant results. For this reason, the KduPRO can be expressed in these same units ( $\mu\text{Mol}\cdot\text{m}^{-2}\cdot\text{s}^{-1}$ ).



*Relation between one module of KduPRO and Li-COR Li-192 reference sensor*

# Reproducibility

Different modules of the KduPRO have been compared between them, offering a good reproducibility of the measurements.

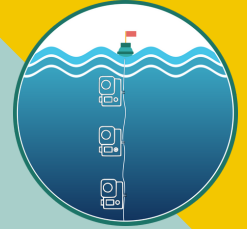


*Relation between different modules of KduPRO*

## Results obtained during MONOCLE field campaign (Loch Leven, 2018)



*Measurements with different modules of KduPRO and Li-COR Li-192 reference sensor*

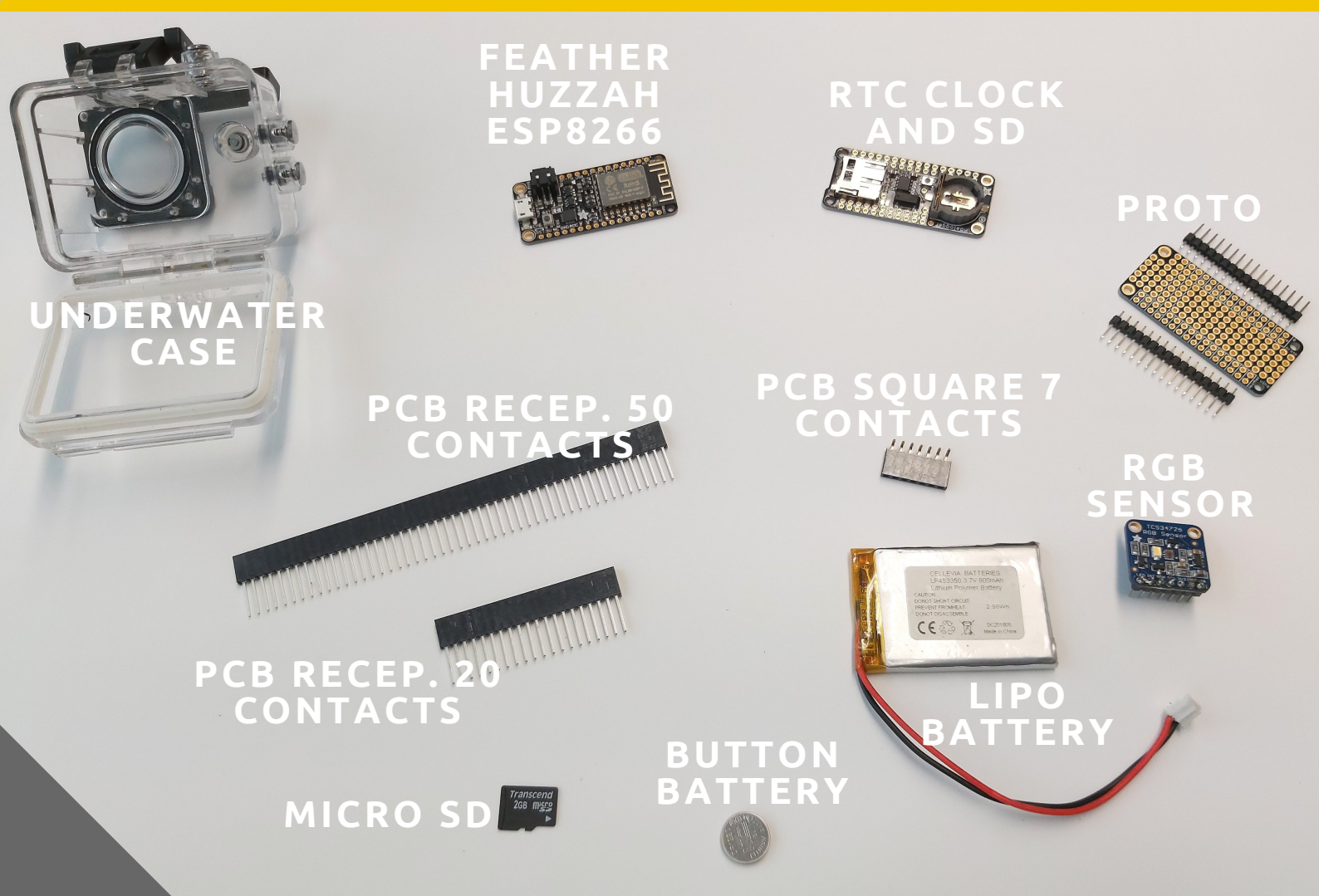


# System installation

## Hardware components

Component	Cost (€)
<u>Feather HUZDAH ESP8266</u>	19.49
<u>Adalogger FeatherWing_RTC Clock and SD</u>	9.60
<u>FeatherWing_Proto</u>	5.31
<u>Underwater case (GoPro or similar).</u>	15.69
<u>TCS34725 color RGB sensor</u>	11.76
<u>Lithium-ion Polymer Battery</u>	8.41
<u>MicroSD</u>	13.39
<u>CR1220 Button Battery 3V</u>	1.33
<u>PCB socket 2.54 mm square 7 contacts</u>	2.19
<u>PCB Receptacle 2.54 mm board-to-board 20 contacts</u>	3.47
<u>PCB Receptacle 2.54 mm board-to-board 50 contacts</u>	6.67
<b>Total</b>	<b>97.31</b>

*Budget for build one module of KduPRO*



**UNDERWATER CASE**

**FEATHER HUZDAH ESP8266**

**RTC CLOCK AND SD**

**PROTO**

**PCB RECEP. 50 CONTACTS**

**PCB SQUARE 7 CONTACTS**

**RGB SENSOR**

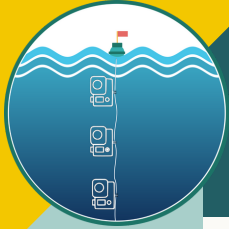
**PCB RECEP. 20 CONTACTS**

**LIPO BATTERY**

**MICRO SD**

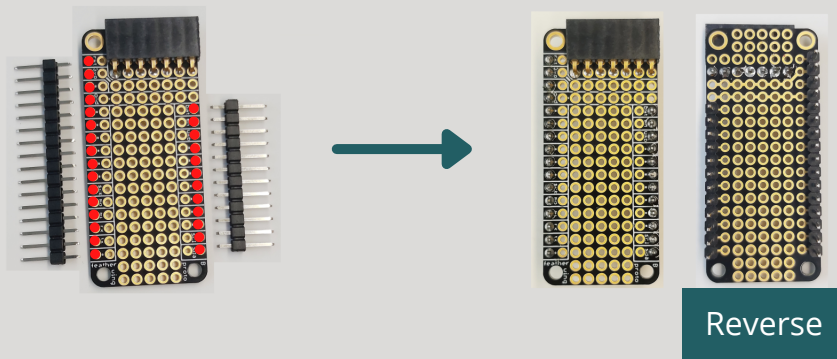
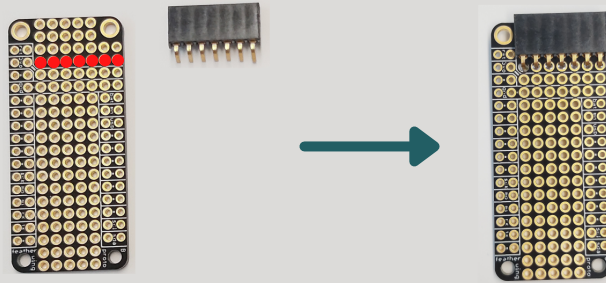
**BUTTON BATTERY**

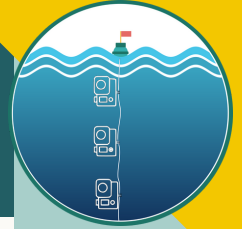




# Installation

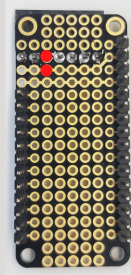
- 1 Solder the PCB socket-7contacts and the assembled headers to the FeatherWing Proto



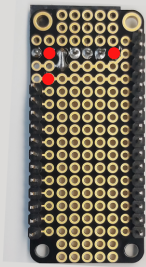
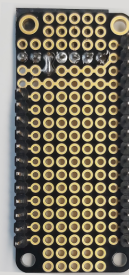


# Installation

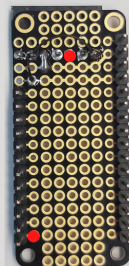
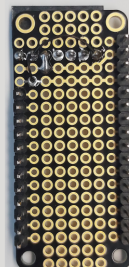
## 2 Solder the circuit of the FeatherWing Proto



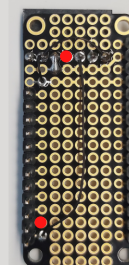
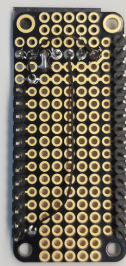
3V3  
→



GND and LED  
→

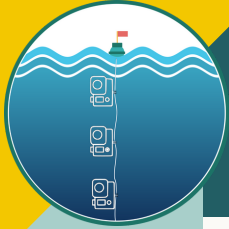


SDA  
→



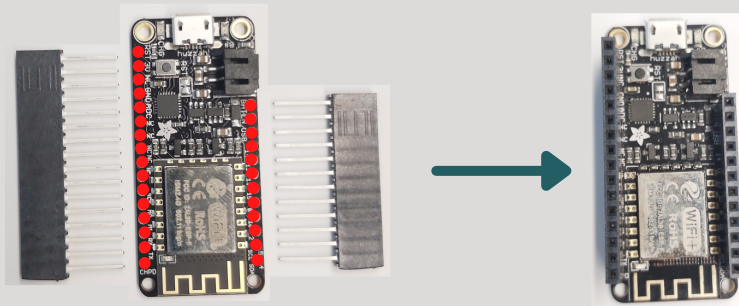
SCL  
→



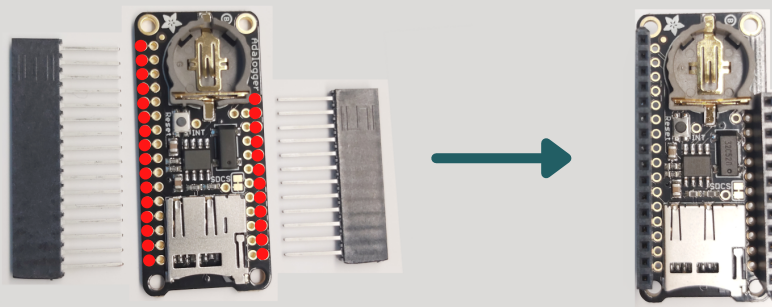


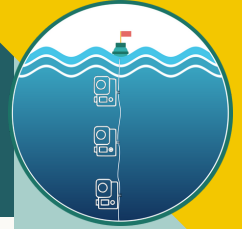
# Installation

- 3 Solder the PCB receptacles to the Feather HUZAZH ESP8266



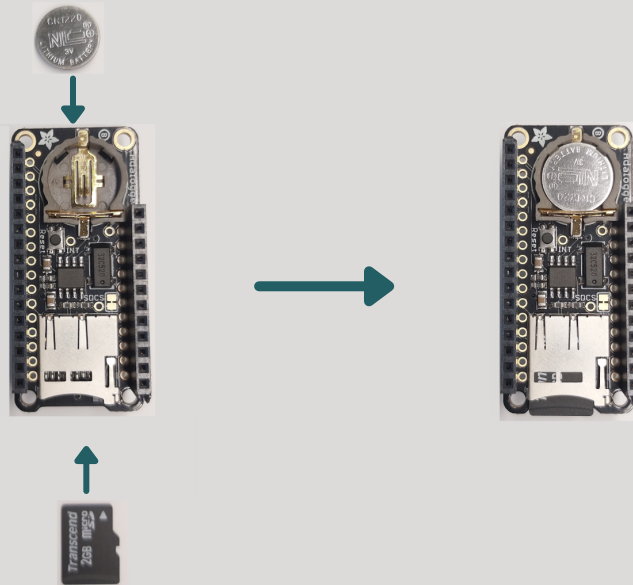
- 4 Solder the PCB receptacles to the Adalogger FeatherWing





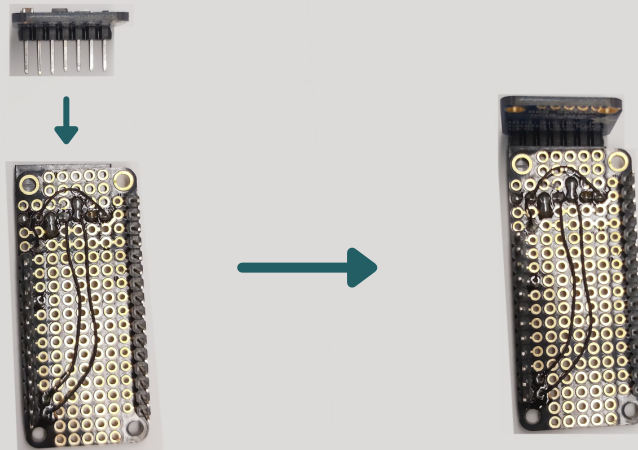
# Installation

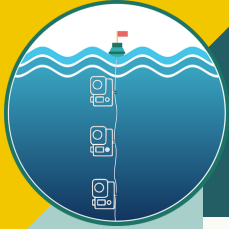
- 5 Insert microSD card and battery button to Adalogger FeatherWing RTC Clock and SD



- 6 Insert the TCS34725 sensor into the PCB socket of FeatherWing Proto.

\*\* In case of prepare a **corrective cosine filter**, please follow steps 7 to 16.  
If not, continue to step 17





# Installation

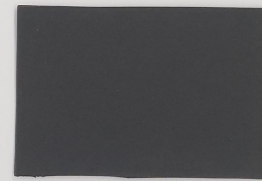
- 7 Material list to do a corrective cosine filter in TCS34725 sensor

## TRANSPARENT GLASS METHACRYLATE PLATE

3MM

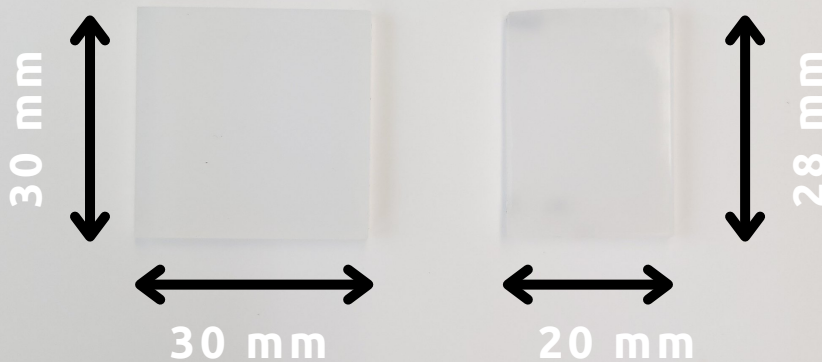


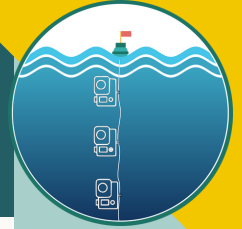
## PTFE TAPE 0.2 MM THICKNESS



## FOAM BLACK 2MM

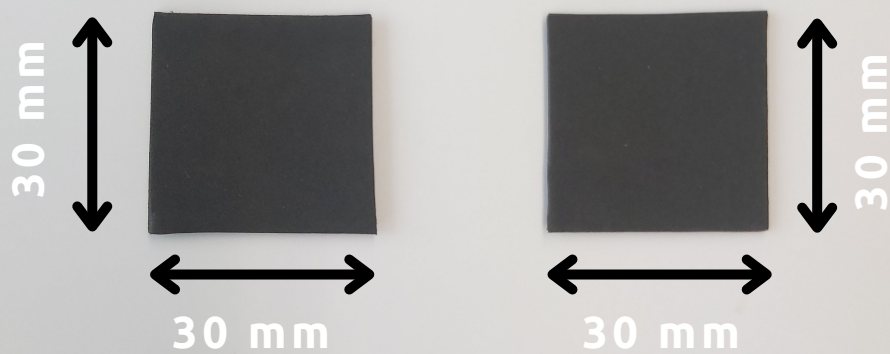
- 8 Cut transparent glass methacrylate plate into 2 pieces with these dimensions: 28mm x 20mm, and 30mm x 30mm



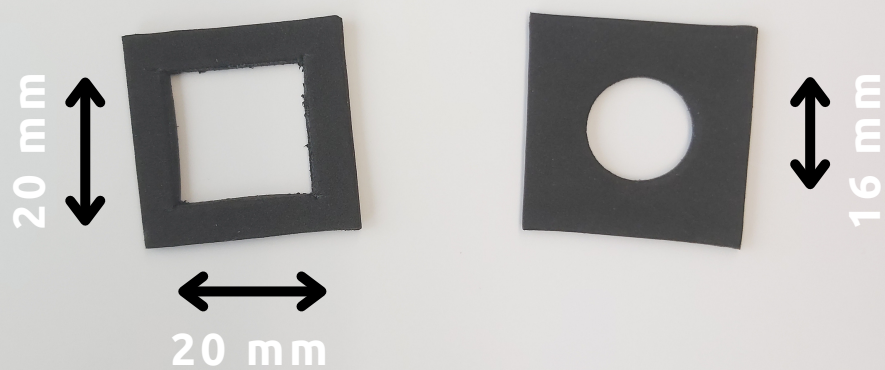


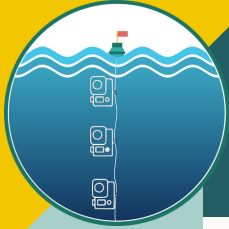
# Installation

- 9 Cut the foam black 2mm into 2 pieces with these dimensions: 30 mm x 30 mm.



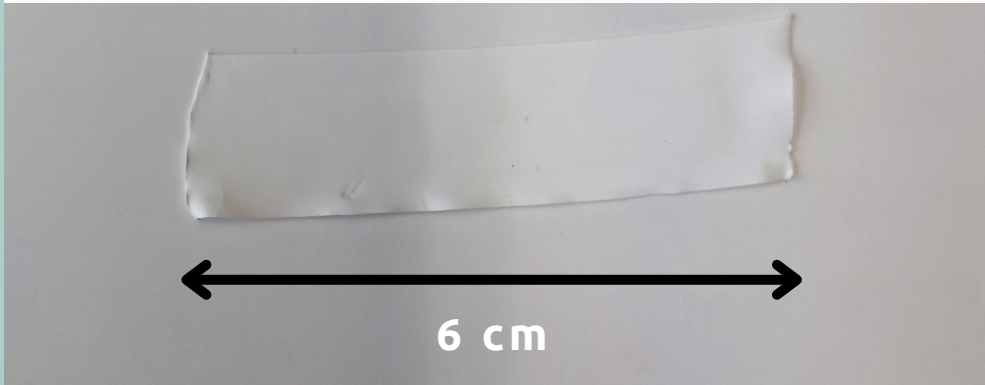
- 10 Cut a circular hole of 16 mm diameter from the centre, and a square hole of 20 mm x 20 mm from the centre.



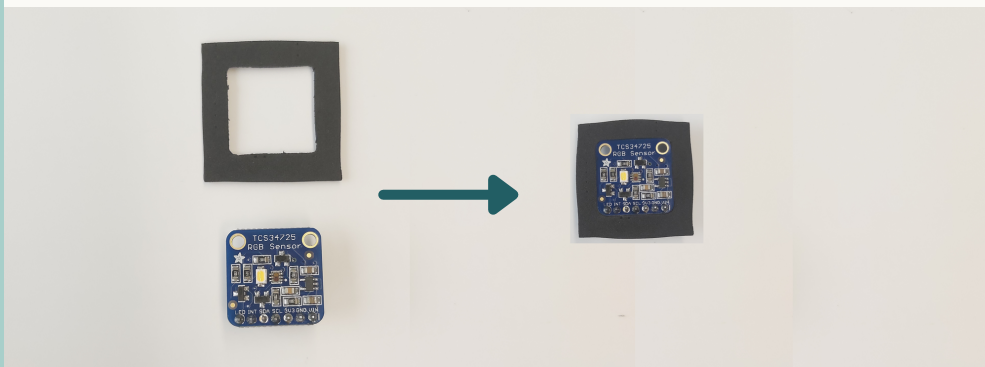


# Installation

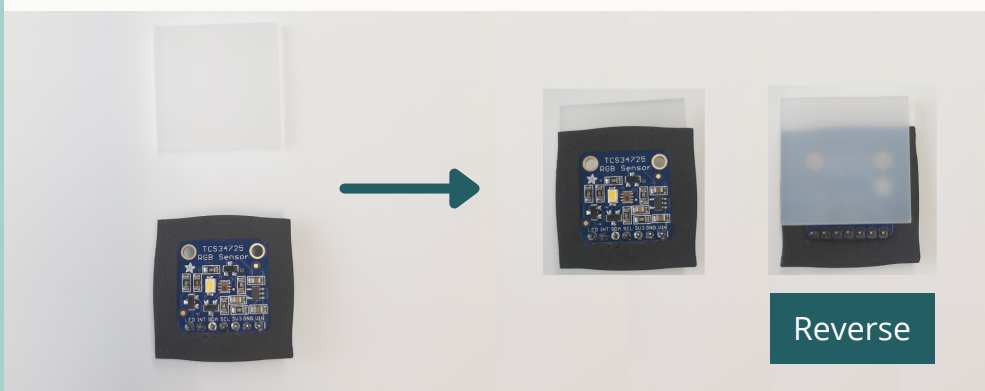
- 11 Cut a strip of 6 cm length of PTFE tape 2mm

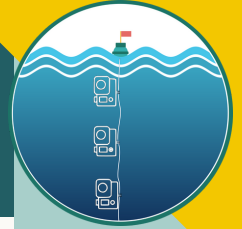


- 12 Put the TCS34725 sensor inside the 20 mm x 20 mm square hole of the black foam piece



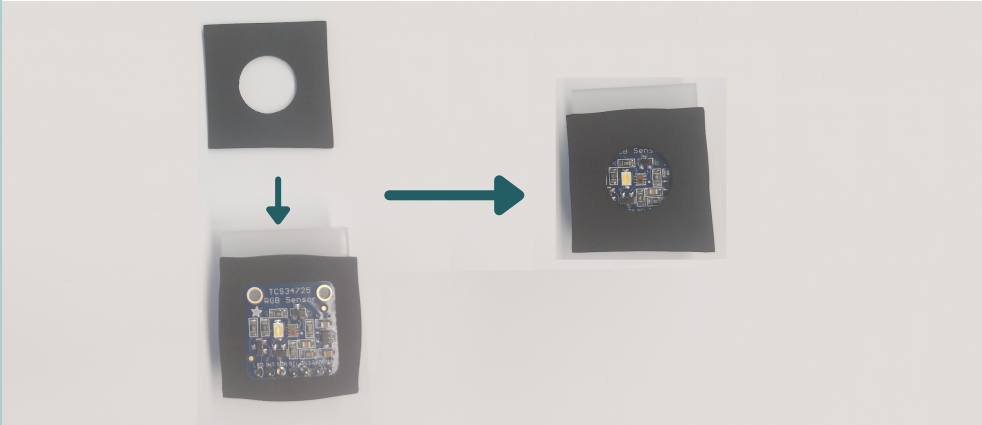
- 13 Stick it (with superglue) to the 30 mm x 30 mm transparent glass methacrylate plate piece (function as a support)



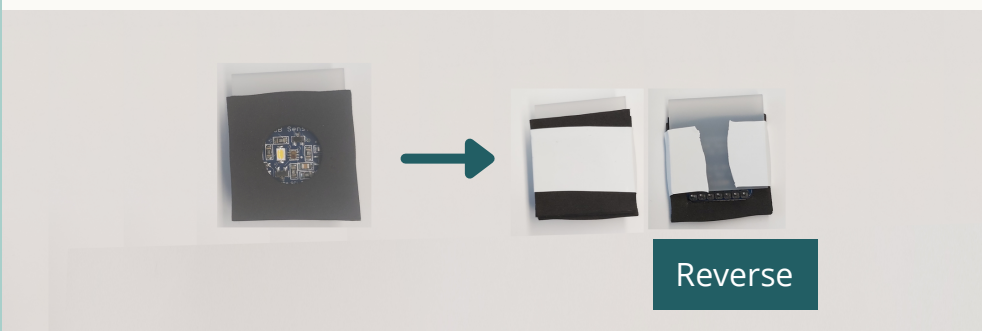


# Installation

- 14 Stick those piece to the 20 mm x 20 mm circular hole of the black foam piece



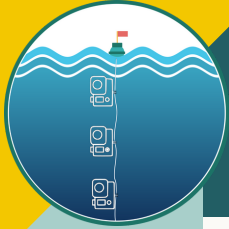
- 15 Put the strip of 6 cm length of PTFE tape 2mm above the piece, surrounding the hole piece



- 16 Stick it to the 28 mm x 20 mm transparent glass methacrylate plate piece

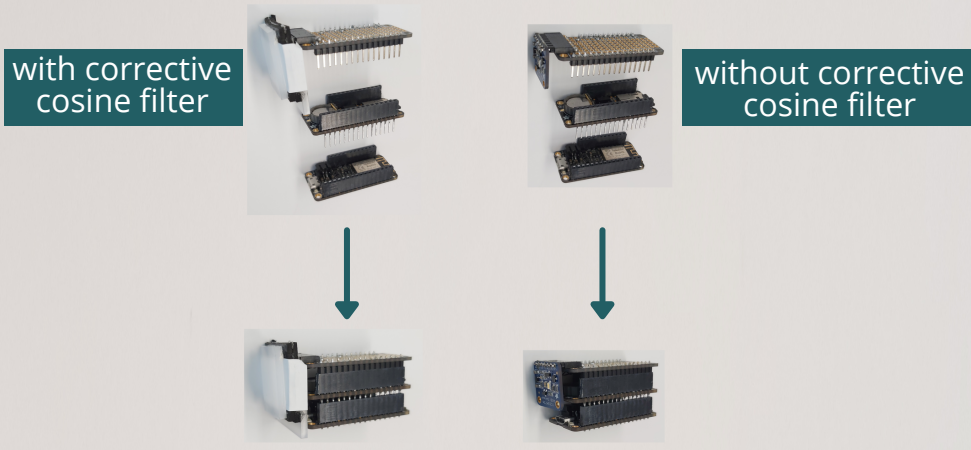




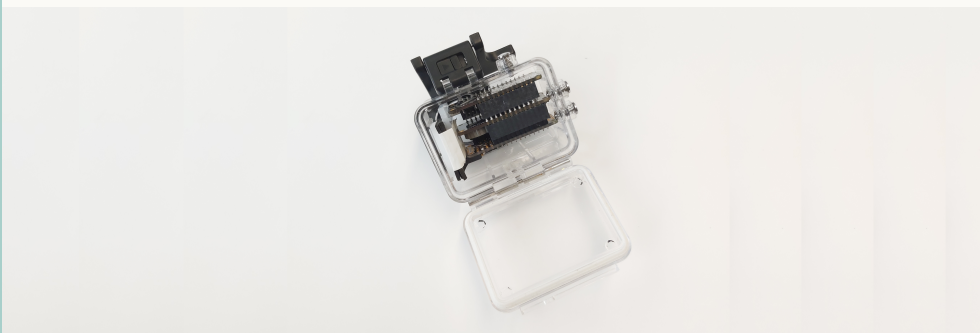


# Installation

- 17 Join the FeatherWing Proto, the Adalogger FeatherWing and the Feather HUZDAH ESP8266



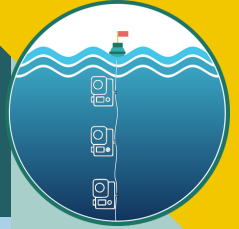
- 18 Put it inside the underwater case as it is shown in the picture below



- 19 Connect the battery and close the underwater case

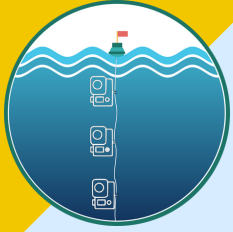


# Installation



Finally, you have finished one module of the KduPRO!



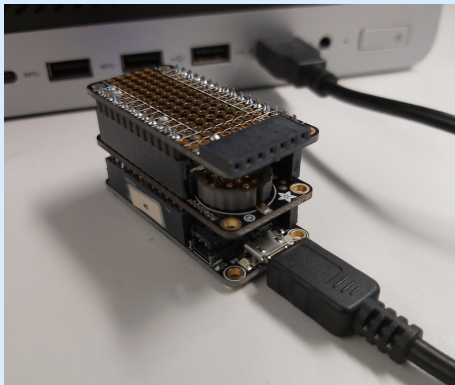


# Software setup

- 1 Download the current firmware from this repository:

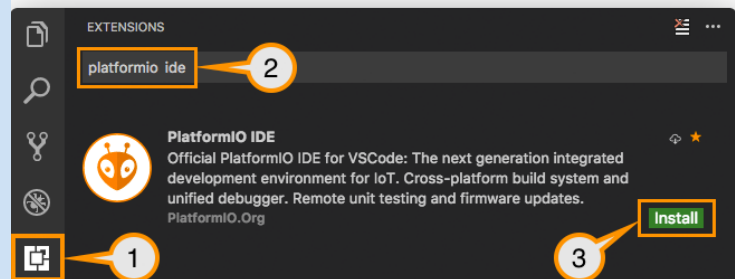
<https://git.csic.es/kduino/kdupro>

- 3 Connect your Feather HUZZAH ESP8266 through a USB

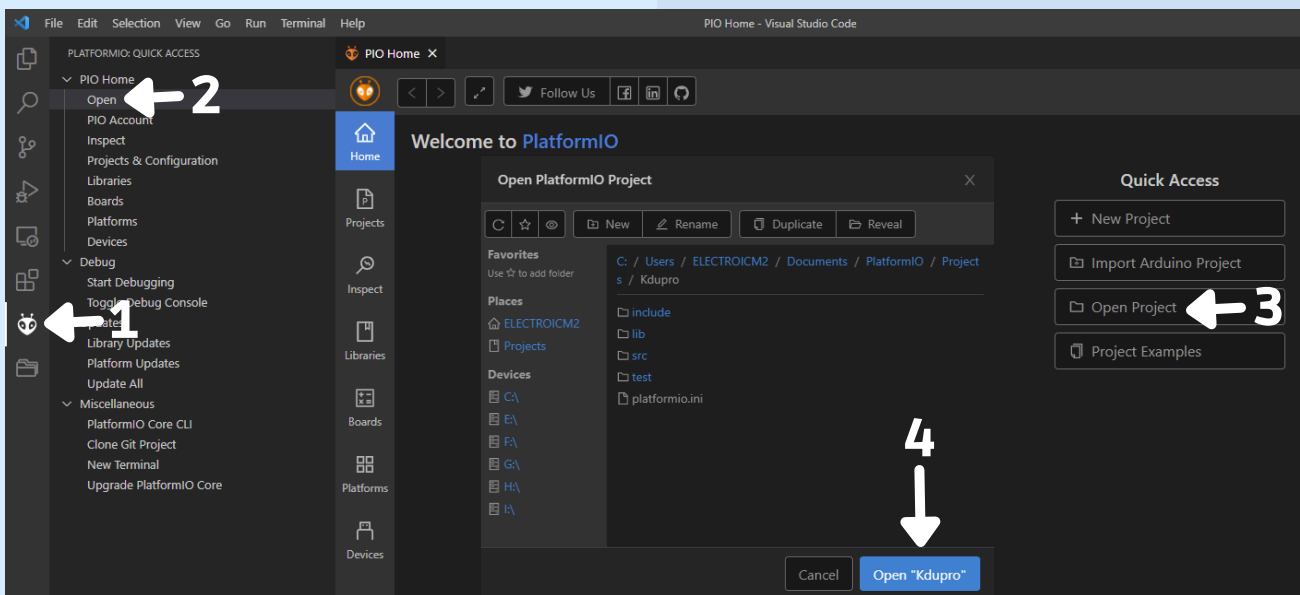


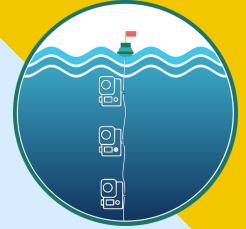
- 2 We suggest using the Visual Studio Code (works on Windows, Linux and Mac) and installing the PlatformIO extension to upload this firmware.

1. Open VSCode Extension Manager
2. Search for official PlatformIO IDE extension
3. Install PlatformIO IDE.



- 4 Open Visual Studio Code, click on the PlatformIO extension (1), open (2) and open project (3). Find the downloaded firmware and open the project (4).





# Software setup

- 5 Edit metadata fields in the `/src/main.cpp` file for each module of the KduPRO (depth, latitude, longitude, owner contact...)

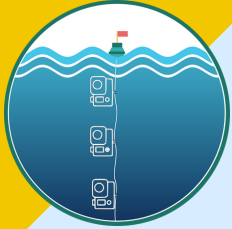
```

src > main.cpp > ...
4  #include <Wire.h>
5  #include "RTClib.h"
6  #include "Adafruit_TCS34725.h"
7  #include <ESP8266WiFi.h>
8
9  // Settings
10 int initial_wait = 60;    // Time to wait before start the loop (in seconds)
11 int measures = 60;      // Number of measurements to do [1, 59]
12 int period = 1;        // Sampling period (in minutes) [1, 60]
13 float depth = 1.0;     // Absolute depth of the device [0.1, 30] (in meters)
14 int sample_counter = 1; // Counter of measurements
15
16 // CSIC Metadata
17 String name = "Kdupro1"; // Name of the module
18 String maker = "ICM-CSIC"; // Maker name
19 String curator = "ICM-CSIC"; // Curator name
20 String email = "jpiera@icm.csic.es"; // Email of the curator
21 String sensors = "TCS34725"; // List with name of used sensors "Sensor 1, ..., Sensor n"
22 String description = "KduPro buoy-1 module-1 totaldepth-4.0m country-spain place-ICM_lab measurements-60"; // Des
23 String place = "lab_ICM"; // Text with place of deployment
24 String units = "counts, counts, counts, counts"; // Units of the measurements "Unit 1, ..., Unit n"
25
26 // MONOCLE Metadata
27 float latitude = 0; // Latitude
28 float longitude = 0; // Longitude
29 float altitude = 0; // Altitude
30 String ref_coord_system = "WGS84"; // Reference Coordinate System
31 String location_source = "GNSS"; // Source of the Geodesic information
32 String time_source = "internet time pool"; // Source of the Time information
33 int processing_level = 0; // Defined by manufacturer and described in the reference documentation.
34 String processing_procedure = "https://git.csic.es/kduino/kdupro"; // Reference to protocols and algorithm
35 String processing_version = "build"; // Version of the data processing software
36 String processing_revision = "0"; // Incremental version of the processed data
  
```

- 6 Click on the bottom PlatformIO toolbox **build** (marked in white in the screenshot below) and then **upload** (marked in red in the screenshot below)

```

22 String description = "KduPro buoy-1 modul
23 String place = "lab_ICM";
24 String units = "counts, counts, counts, c
25
26 // MONOCLE Metadata
27 float latitude = 0;
28 float longitude = 0;
29 float altitude = 0;
30 String ref_coord_system = "WGS84";
31 String location_source = "GNSS";
32 String time_source = "internet time pool"
33 int processing_level = 0;
34 String processing_procedure = "https://gi
35 String processing_version = "build";
36 String processing_revision = "0";
  
```



# Software setup

- 7 When the upload is complete, the message **SUCCESS** appears

```
Writing at 0x00034000... (93 %)
Writing at 0x00038000... (100 %)
Wrote 316960 bytes (230306 compressed) at 0x00000000 in 20.3 seconds (effective 125.0 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
===== [SUCCESS] Took 26.78 seconds =====
```

- 8 Finally, we need to synchronize the real-time clock (it is a function included in the KduPRO code). Open the **serial monitor** (marked in red in the screenshot below).

```
39 String calibration_time = "0";           // Date/time stamp of applic
40 String calibration_version = "0";       // Version of the calibratio
41 String sensor_id = "kdupro1_1_1_4_20211118171348"; // Uniqu
42 String platform_id = "lab_ICM_20211118171348"; // Platfo
43 String deployment_id = "1_20211118171348"; // Randomly a
44 String sample_id = "1_4_20211118171348"; // Randomly
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN **TERMINAL**

> Executing task: C:\Users\ELECTROICM2\.platformio\penv\Scripts\platformio.exe device m

--- Available filters and text transformations: colorize, debug, default, direct, esp8266\_exception\_decoder, hexlify, log2file, nocontrol, printable, send\_on\_enter, time  
--- More details at <https://bit.ly/pio-monitor-filters>  
--- Miniterm on COM4 9600,8,N,1 ---  
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---

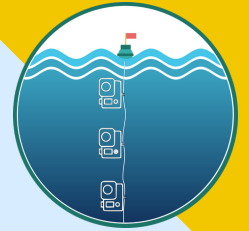
> ESQUEMA

- 9 And enter the date and time in the following format (it is not visible when the user typing in the serial monitor): YYYYMMDDhhmmss

```
42 String platform_id = "lab_ICM_20211118171348"; // Platform serial number or randomly assigned ident
43 String deployment_id = "1_20211118171348"; // Randomly assigned identifier (UUID) specific to depl
44 String sample_id = "1_4_20211118171348"; // Randomly assigned identifier (UUID) generated with
```

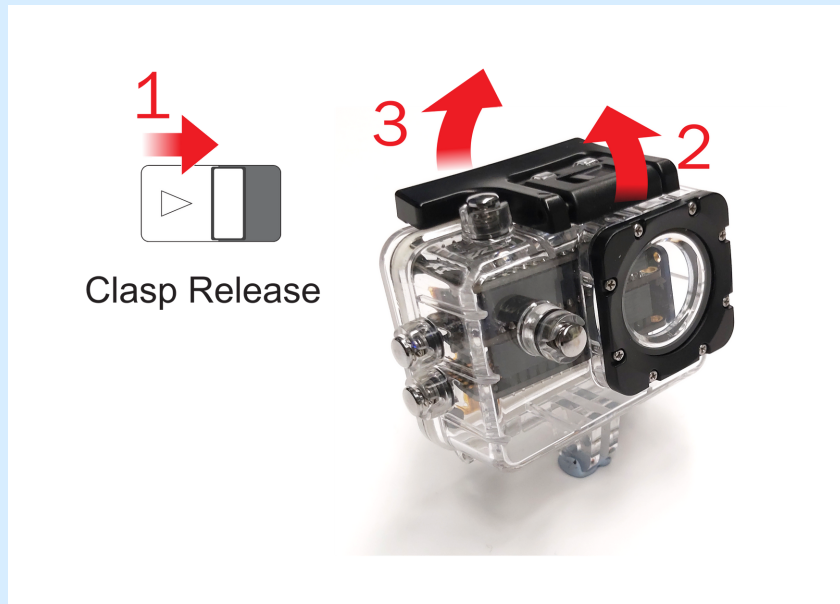
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN **TERMINAL**

--- Available filters and text transformations: colorize, debug, default, direct, esp8266\_exception\_decoder, hexlify, log2file, nocontrol, printable, send\_on\_enter, time  
--- More details at <https://bit.ly/pio-monitor-filters>  
--- Miniterm on COM4 9600,8,N,1 ---  
--- Quit: Ctrl+C | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---  
2021-11-22T16:11:20Z  
Real Time Clock Updated  
Waiting 60 seconds

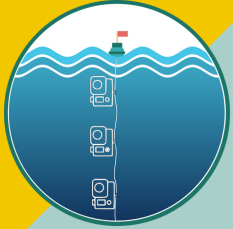


# System check

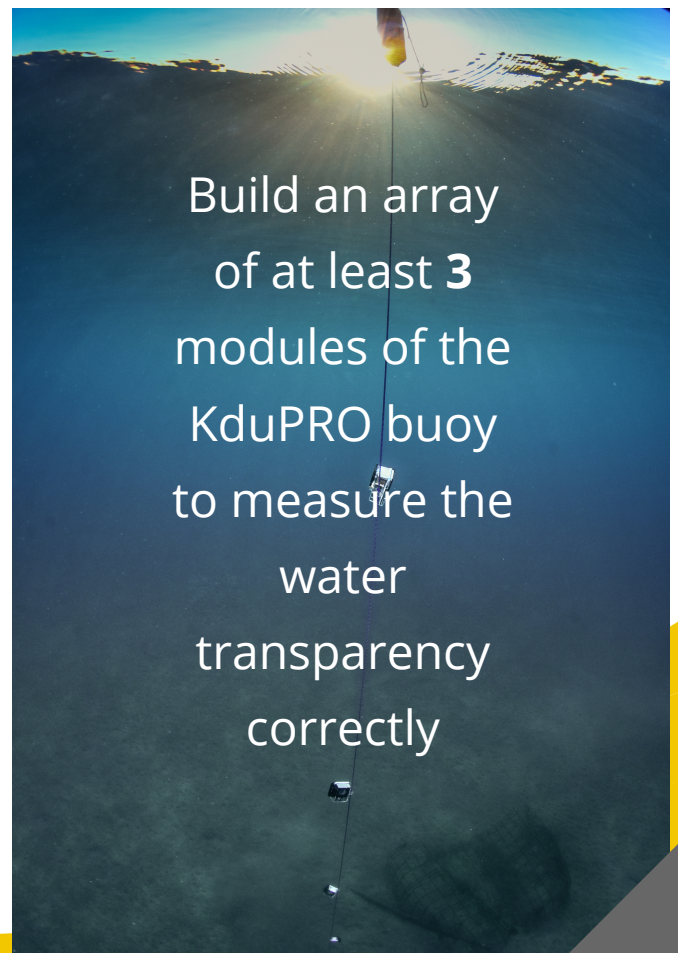
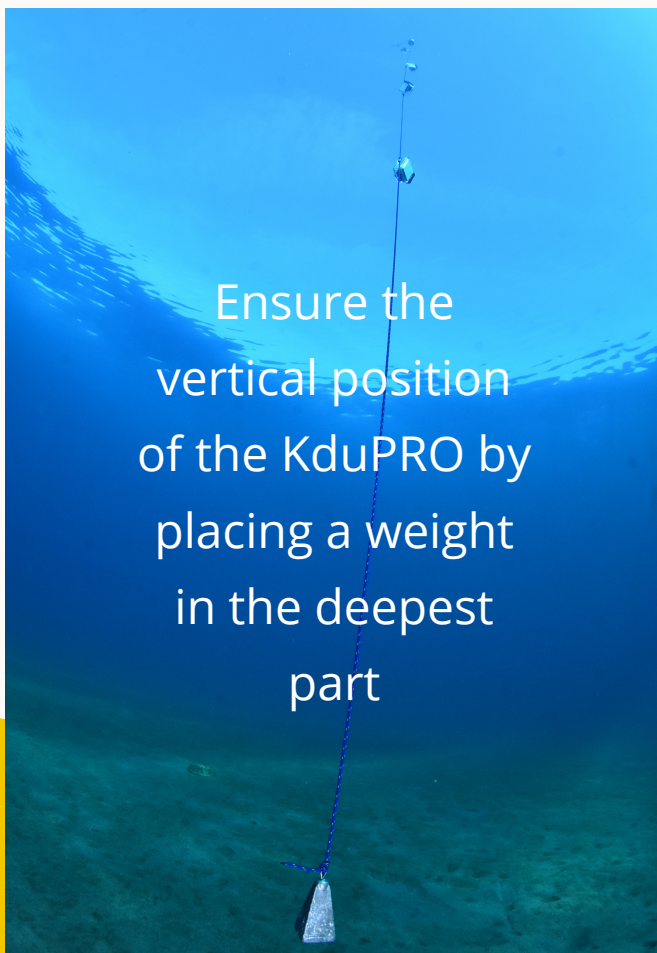
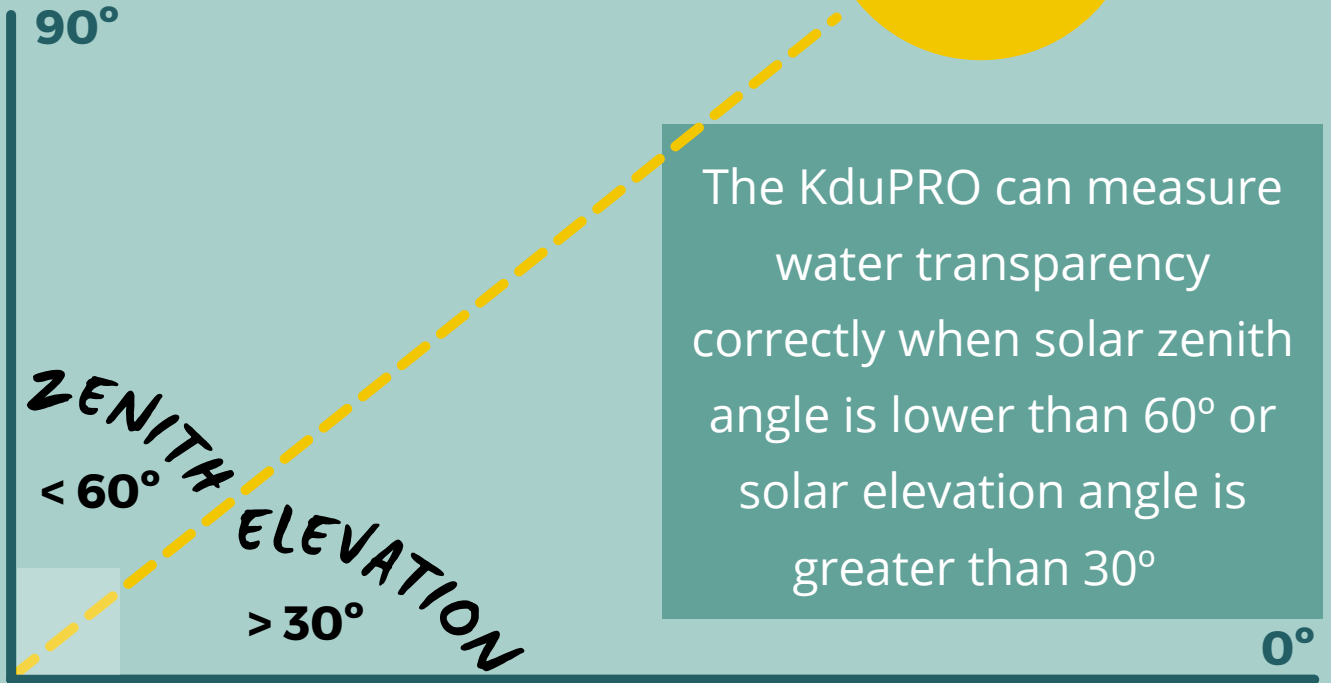
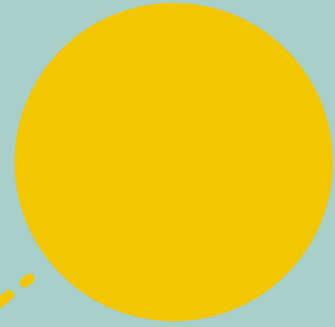
- Check the locking system

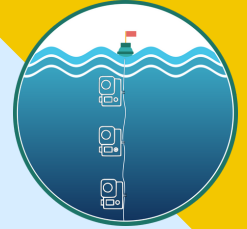


- Battery charged
- Battery button charged
- SD card empty or enough space available
- Battery cable secure
- LED blue when the battery is connected. LED off after initial wait in seconds



# Data gathering





# Data process

- 1 Download the KduINO Data Analysis to process the KduPRO data from this repository:

<https://git.csic.es/kduino/kdupro>

- 2 Follow the instructions of the README file

README.md

## Kduino Data Analysis

Kduino data analysis is a python module designed to open and analyze data files from Kduino instrumentation. Also, provide methods to generate plots and convert data files in netCDF and CSV format.

This module works with different versions of Kduino:

- Kdupro
- Kdustick
- Kdumod (in development)

### Installation


Clone this repository in your local system (`git clone git@git.csic.es:kduino/kduino-data-analysis.git`) and then launch the `main.py` file to run the Kduino Data Analysis.

### Usage

You need to add your data file inside `data` folder and create a new configuration file inside the properties folder. There is a template and so many examples to create your own configuration.

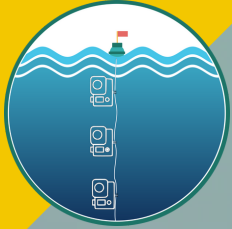
### License

MIT



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776480 (MONOCLE).





# System maintenance

- Clean the sealing
- Clean the underwater case (biofouling) to avoid contamination between samples

- Check the solder joints
- Update Real Time Clock in future deployments

# Contact information

Contact information for KduPRO can be obtained from:

Carlos Rodero  
rodero@icm.csic.es

Raul Bardaji  
bardaji@utm.csic.es

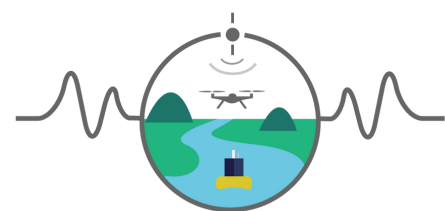
Jaume Piera  
jpiera@icm.csic.es

For additional material and resources, please visit the url:  
[https://monocle-h2020.eu/Sensors\\_and\\_services/KdUINO](https://monocle-h2020.eu/Sensors_and_services/KdUINO)

If you encounter a problem testing the KduPRO, please e-mail us the following:

- A description of the problem, error messages, or other pertinent information.
- The error.log file, which is found in the error directory

This information will greatly increase the speed at which we can troubleshoot the problem



MONOCLE



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776480



Institut  
de Ciències  
del Mar

