

Hands-on 3: Automated environmental monitoring with telemetry

Professor Wouter Buytaert
Imperial College London

Download the code

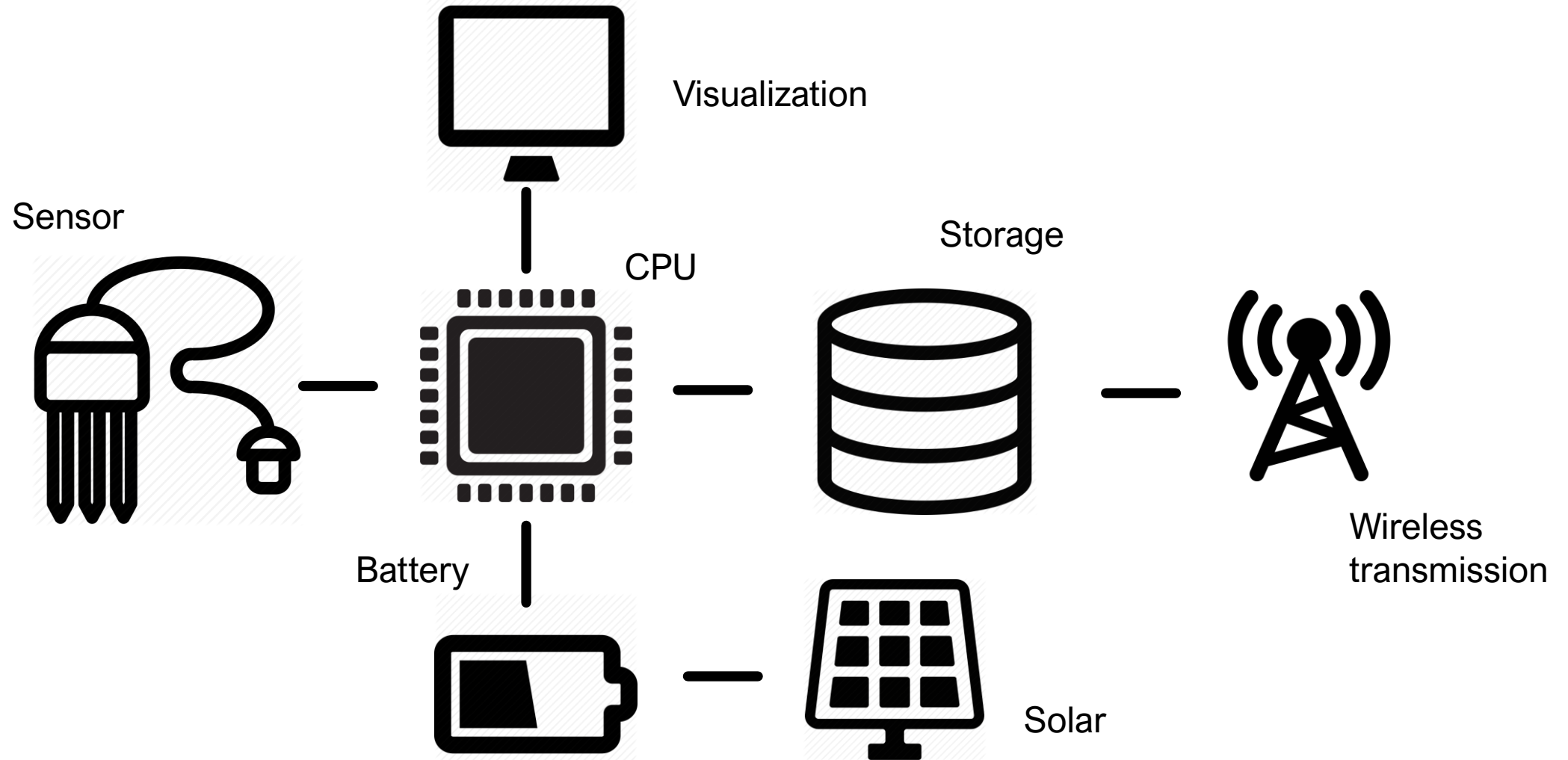
1

2

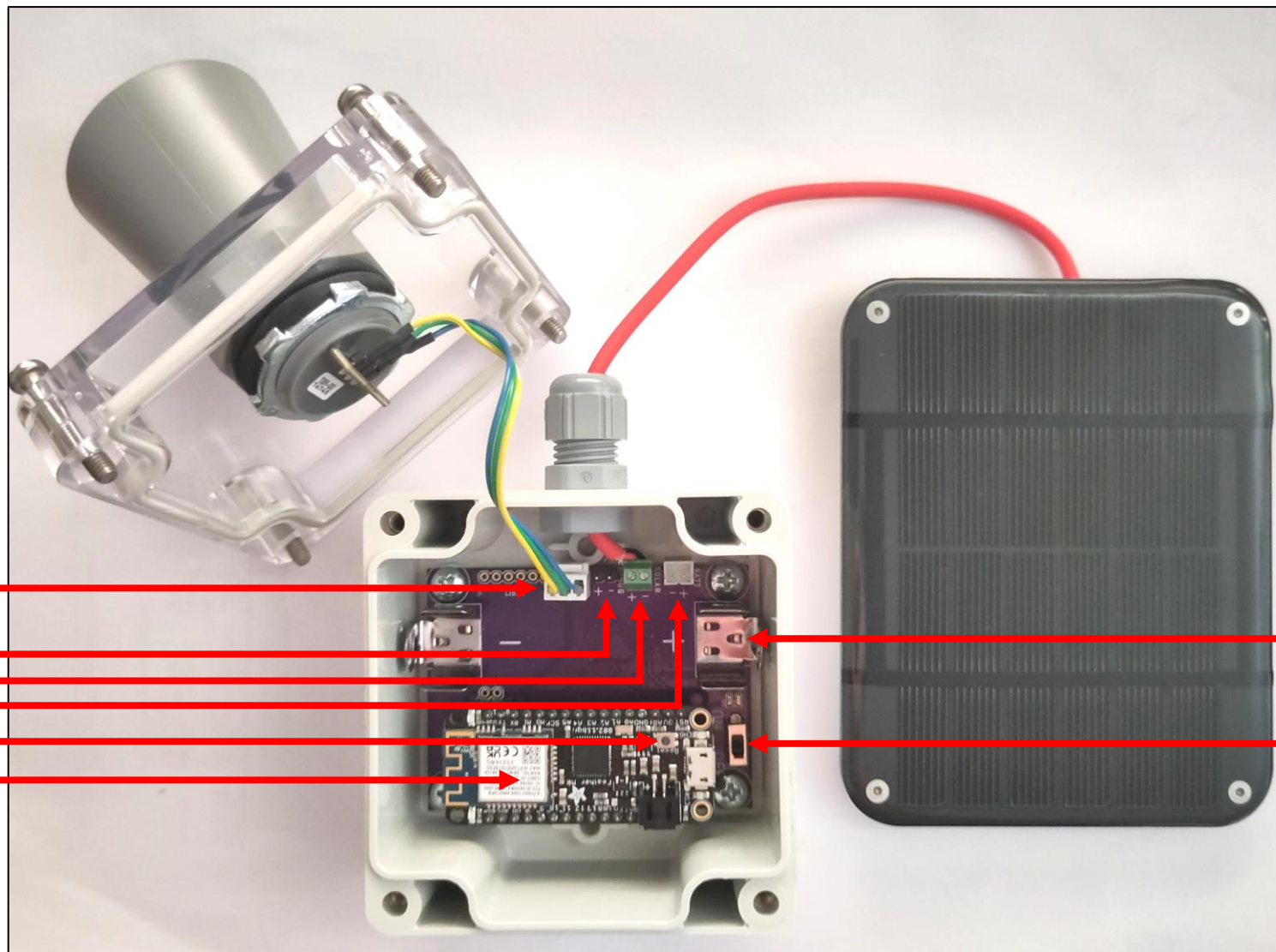
The screenshot shows the GitHub interface for the repository 'UNESCO_Open_Hardware_Cookbook' by 'ICHydro'. The repository is public and has 1 star and 0 forks. The 'Code' button is highlighted with a red arrow labeled '1'. A dropdown menu is open, showing options to clone the repository using HTTPS, GitHub CLI, or the web URL, or to open it with GitHub Desktop. The 'Download ZIP' option is highlighted with a red arrow labeled '2'. The repository's README is visible, stating that it contains materials for an online course on water resources management. The repository is licensed under GPL-3.0.

https://github.com/ichydro/UNESCO_Open_Hardware_Cookbook

Logger architecture



Overview of the logger



Level sensor connector

Debug serial port

Solar panel connector

Battery connector

Reset button

Adafruit feather board

Battery holder

On/off switch

The Real Time Clock

Dataloggers need a real time clock to timestamp the recorded measurements. Many embedded CPUs have an embedded clock (RTC), including the SAMD M0.

Explore and run the following script:

SetClockM0.ino

Install the following libraries:

- Time by Michael Margolis
- RTCZero by Arduino

Testing the sensor connection

The maxbotix sensor has both an analog and a digital interface. The digital interface uses the serial communication protocol. Once turned on, it reports measurements at its internal frequency, which is 6 measurements per second.

Explore and run the following script:

`ultrasound_test.ino`

SD card logger

Now, we bring all the functionality together in a single sketch that performs all the steps required from a logger system:

- retrieve a measurement from the sensor
- write the measurement to the SD card
- repeat this sequence at a predetermined interval

Explore and run the following sketch:

`ultrasound_sd.ino`

Wifi logger

Instead of logging to an SD card, we can also send the data via telemetry. This is quite a bit more complicated, and requires a telemetry module such as wifi, but also a remote server to receive the data, and a communication protocol with the server. In this case we will use MQTT.

Explore and run the script:

`ultrasound_wifi.ino`

Note: you will need the connection details and credentials provided in the course