Actimetre System

Last update: 2024-05-07

Overview

Actimetre is short for "Activity Metre". It is intended to measure the activity of lab animals, typically rats in cages, using commodity electronic components. For a hardware cost of less than 10€ per cage, the system can record 3 to 6 axes of activity (acceleration XYZ and gyroscope XYZ) at up to 8kHz.

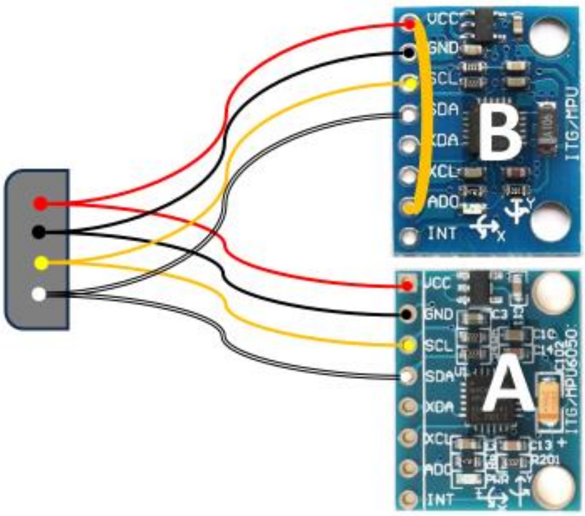
At the base of the system is the MPU-6050/6500 sensor which measures XYZ-axis acceleration (up to 1kHz for the 6050 and 4kHz for the 6500) and XYZ-axis rotational speed at up to 8kHz. At the next level is a ESP32 family MCU which has 2 I2C ports and one or two execution cores. It can therefore control up to 4 sensors (2 addresses on 2 ports) while communicating over WiFi, connected to the Actiserver, which is an OrangePi Zero3 running Debian Linux. Actiserver also provides the WiFi AP service to the MCU, as well as being the transient data repository. It is connected via Ethernet to the central dashboard which helps monitor and manage the system. The whole system can theoretically contain up to 35000 sensors.

The entire system is fairly robust to disruption, except the Central component which is assumed to be highly available. Installation and configuration requires some amount of technical knowledge.

Acticentral is a single central server currently running on a Linux PC. It has a fixed IP address. Actiservers communicate with Acticentral via HTTP.

The MPU-6050/6500 sensor

Datasheets here : [MPU-6050 | TDK InvenSense](), [MPU-6500 | TDK InvenSense](). Invensense says it's not recommended for new designs, but it's the only 6-axis sensor chip sold on a consumer-grade module at low cost. The 6500 part seems to be slightly more stable, but the generally available module is slightly larger.



The AD0 pin can be connected to either GND (0x68) or VCC (0x69) to control the sensor's I2C address. The 0x68 is called the "A" sensor, the 0x69 unit is "B". We need four lines (VCC, GND, SDA, SCL) to reach the ESP32. One "A" sensor and one "B" sensor forms a pair.

MCU variants from Espressif

The Espressif ESP32-S3 chip is described here: ESP32-S3 Wi-Fi & BLE 5 SoC | Espressif Systems. Espressif provides a board support library for Arduino IDE. It is available to the public in several module forms, among which the S3 Mini (Wemos) and the S3 Zero (Waveshare) have been tested.

The ESP32-S2 is the single-core version and is more affordable. However, it is not recommended for controlling more than one sensor. The S2 mini (Wemos) is the most commonly available module.

The ESP32-C3 is an even lower-spec version, with a slower CPU. Again, it can control at most one sensor. The C3 Zero (Waveshare) is a compact and cost-effective module.

<u>The OLED display</u>

The 0.96 inch 128*64 OLED screen with a SSD1306 controller on I2C interface is a ubiquitous component. It can be found from many places from AliExpress.com. Be sure to use the newer, more compact version with yellow and blue lines.

The Actimetre module

[This section is obsolete, please refer to the Newbox section]

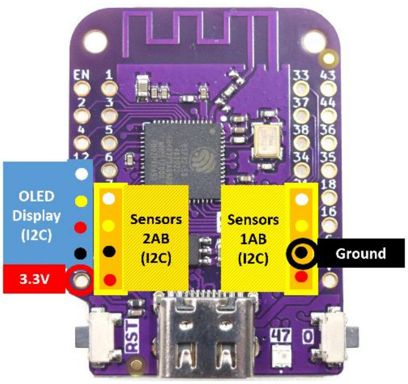
The reference Actimetre is composed of:

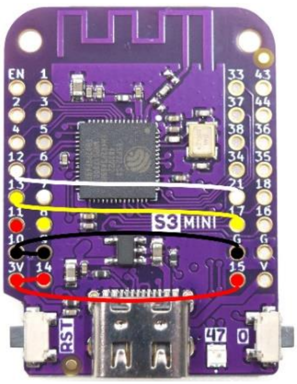
- a ESP32-S3 mini module
- a 0.96-inch diagonal 128x64 OLED I2C display
- two 4-pin JST 2.54mm ports for connecting sensors

Note that these components are directly soldered on the S3 mini module, so the Actimetre doesn't require additional PCB. Each Actimetre draws less than 250mA at most, so 4 of them can be powered by a low-cost USB 2.0 hub and a 1A power supply.

The display is on the "up" side, which is the side of the module with the USB connector. The connectors to the sensors (I2C, 4-pin JST 2.54) will be on the opposite side. This connector carries VCC, GND, SCL, SDA signals. The display shares the I2C port with Sensor pair "1" (I2C port #0).

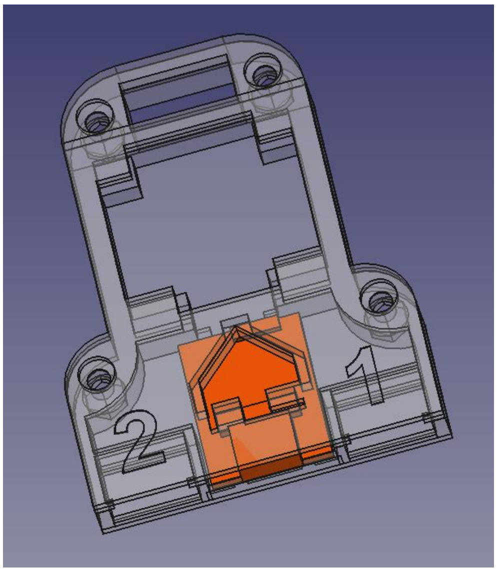
There are 9 lines to connect, preferably by wrapping. The JST connectors must be soldered because the legs don't have enough length for wrapping. The screen will be soldered last and will cover all the wires.

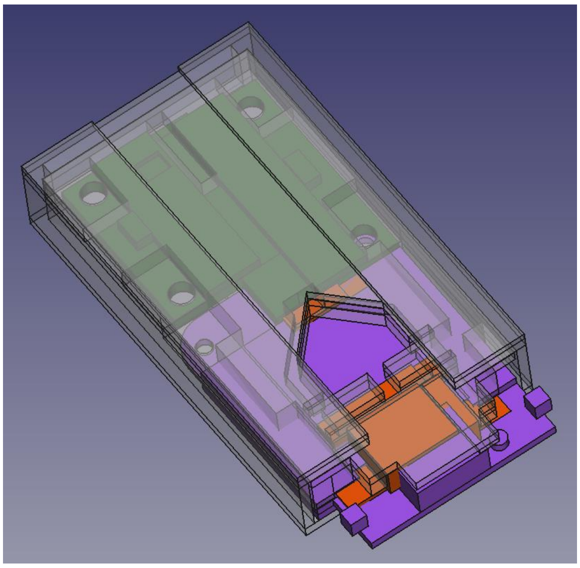




Newbox

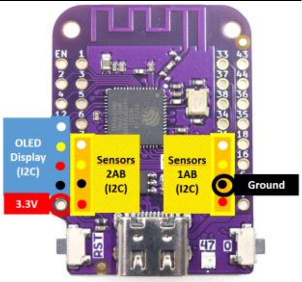
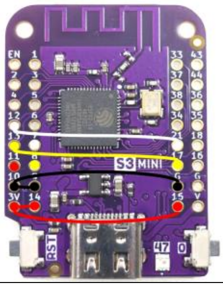
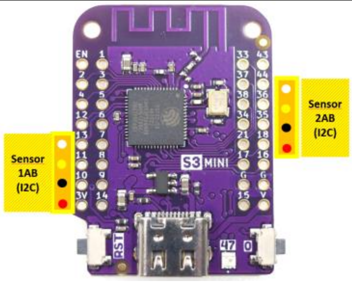
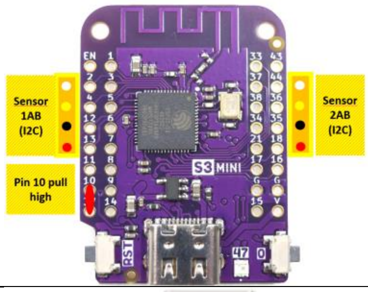
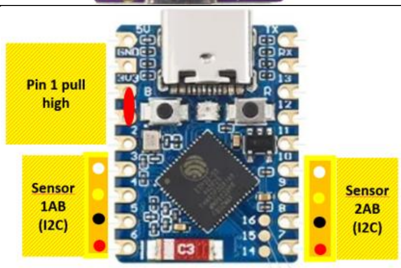
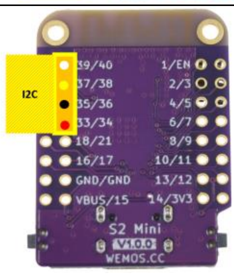
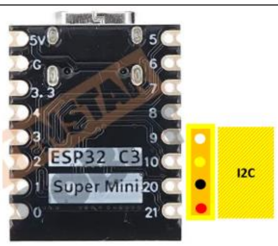
We found that soldering directly on the module made it difficult to service, so a new packaging of Actimetre lays out the components flat in a thin and wide box, and uses the S3 Zero for a smaller footprint (image below left).

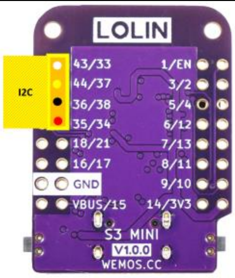
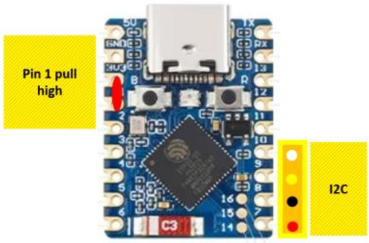




There is also a “Solo” box that matches a single sensor with a single MCU, in a compact box (57x29x11mm) with only two wires for 5V power (image above right).

Board types as of v400:

Board Type	Code	Wiring	Remark
S3i		 	Old-style box for S3; deprecated. Pins 14 and 15 are pulled HIGH, 9 and 10 are pulled LOW electrically.
S3n			Newbox with S3 mini Vcc and GND pins are self-powered (digital HIGH or LOW). Deprecated
S3m			Alternate wiring for newbox. Pin 10 must be pulled high to distinguish from S3n. OLED Vcc should be taken from 3.3V pin.
S3z			Newbox for S3 Zero Pin 1 should be pulled high electrically to distinguish from S3 mini.
S2o			Solo S2
C3o			Solo C3

Board Type	Code	Wiring	Remark
S3o			Solo S3 mini
S3s			Solo S3 Zero Pin 1 must be pulled HIGH electrically to distinguish from S3 mini

Actiserver

The Actiserver can be any Linux-based SBC (single-board computer) with both Ethernet and WiFi. It must support WiFi AP (which some USB dongles don't). 1GB of RAM is recommended. The software has been tested on Raspberry Pi3 (1GB), Pi4 (2GB), CM4102000; NanoPi NEO3 (2GB) with a RTL8812 USB WiFi dongle; Orange Pi Zero2 (1GB) and Zero3 (1GB). The Orange Pi Zero3 (1GB) has the best cost/performance ratio and supports an external antenna for better reception.

Actiserver software was written in Kotlin using IntelliJ IDEA. It relies on OpenJDK 17's JVM and therefore is portable across a wide range of systems.

Communication protocols

Communication between Actimetre and Actiserver is a simple socket connection. The Actiserver acts as WiFi AP, but doesn't provide a bridge to its Ethernet connection. Since an Actimetre will know the Actiserver's local IP address (192.168.4.1 by default), it can address it directly. Actiserver software listens on port 2883 and 2882 at that address, and accepts socket connections. If a socket connection is broken for any reason, the Actimetre reboots by itself, and Actiserver knows the Actimetre is down.

Port 2882 is used for side-band traffic. At boot, Actimetre scans the air for existing APs, then attempts to connect to them in turn in order of strongest signal. Once the WiFi link is up, it connects to port 2882 on the gateway and sends an Assignment Query message. This message contains information to up to 10 APs whose SSID is of the form "ActisNNN". The message is always 31 bytes in length, even if there are less than 10 APs detected.

Byte no.	Content
0	Number of APs detected (up to 10)
1	MSB of NNN_1
2	LSB of NNN_1
3	RSSI of NNN_1 (dB in positive integer)
4 .. 27	
28	MSB of NNN_{10}
29	LSB of NNN_{10}
30	RSSI of NNN_{10}

Upon receiving this, the Actiserver calls upon Acticentral to assign a server for this Actimetre. See below for details on the HTTP request `action=actimetre-query`. If an error occurs, Actiserver can respond with an error code. The Actimetre will try the next available Actiserver, or failing that, self-assign an Actiserver. The response message is just a single byte containing the index of the Actiserver assigned, or an error code >= 100. The socket is closed after this.

The algorithm used to select the optimal Actiserver relies on 3 data points: the RSSI perceived by the Actimetre, the CPU and disk utilization of the Actiserver.

Once the Actimetre knows which Actiserver it should use, it re-established a WiFi link with it if needed. It connects to port 2883 on the gateway. This socket connection is the single communication channel from this point.

Once a socket is established, Actimetre starts the communication by sending an initial 13-byte packet containing its board type, its MAC address, what sensors are present, and the SW version.

Byte no.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Board type ASCII character 0							
1	Board type ASCII character 1							
2	Board type ASCII character 2							
3	MAC address byte 0							
4	MAC address byte 1							
5	MAC address byte 2							
6	MAC address byte 3							
7	MAC address byte 4							
8	MAC address byte 5							
9	2B is 6500	2A is 6500	2B present	2A present	1B is 6500	1A is 6500	1B present	1A present
10	Version string ASCII character 0							
11	Version string ASCII character 1							
12	Version string ASCII character 2							

Actiserver responds with the unique Actim ID that is assigned to this Actimetre (two bytes, MSB first), as well as the current UTC time in 32-bit epoch seconds since 1970 (four bytes, MSB first). This allows Actimetre to timestamp all its data.

Once Actimetre enters its main loop, it sends a data packet at the sampling frequency it is set to. The default frequency is 1kHz for the S3. This frequency can be changed by pressing the button on the Actimetre. A data packet is composed of a 8-byte header containing the millisecond timestamp and the sampling frequency, followed by payloads of 12 bytes per measurement. The payload is made of 16-bit signed integers for each of the sensor’s output: Accelerometer XYZ axes and Gyroscope XYZ axes.

Byte no.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Seconds since boot, bits 16-23, see [Note3]							
1	Seconds since boot, bits 8-15							
2	Seconds since boot, bits 0-7							
3	I2C port	I2C address	Number of samples in payload					
4	RSSI (0~7)			Sampling mode		Frequency code		
5	is 6500	Heartbeat [Note5]		[Note4]	Microseconds, bits 16-19			
6	Microseconds, bits 8-15							
7	Microseconds, bits 0-7							

- Seconds since boot, this 24-bit unsigned value maxes at 0xFEFFFF (decimal 16711679, representing about 193 days). The Actimetre does a cleaning reboot after 183 days.
- Sampling mode: all axes (0 or 3), only accelerometers (1), only gyroscopes (2).
- Frequency code: 0=100, 1=500, 2=1000, 3=2000, 4=4000, 5=8000
- Microseconds is taken for the last sample of the payload. So if the payload contains 10 samples (byte 3 bits 0-5 contains 10), the time of the first sample of the payload would be 9 sampling periods earlier than this.

That data is decoded by Actiserver who appends each reading in a text format to the appropriate file in the repository. Data repository is in `/media/actimetre` and the file name encodes the Actim ID as well as the start time of the data capture. Actiserver can be configured to start a new file when a certain size is reached (default 1GB) or when a certain time has elapsed (default 24 hours). See the installation instructions for details on how to configure Actiserver.

Actimetre also listens on the TCP connection, and is ready to receive commands in the form of a single byte. See [Note1] for a list of commands.

Actiserver, when it boots up, waits to have a reliable clock via NTP before it starts accepting clients. Then, it seeks connection to Acticentral, which is at `actimetre.fr`. Acticentral provides a HTTP CGI entry point, `https://actimetre.fr/cgi/acticentral.py`, that is used for all communication. Note that there is no open back-channel to Actiserver. All communication from Acticentral to Actiserver is handled in the response message to GET or POST requests.

Request	Request type	URL parameters	POST data	Response
Actiserver active (deprecated)	POST	`action=actiserver` `serverId=`*server ID*	Full data on status of Actimetres connected to this Actiserver	Registry data
Actiserver active (current)	POST	`action=actiserver3` `serverId=`*server ID*	Full data on status of Actimetres connected to this Actiserver	`OK` or [Note1]
Fetch Registry	GET	`action=registry` `serverId=`*server ID*		Registry data

Request	Request type	URL parameters	POST data	Response
Fetch Projects DB	GET	`action=projects` `serverId=`*server ID*		Projects data
Request assign	POST	`action=actimetre-query`	List of APs detected by Actimetre	Assignment, as index to the list
New Actimetre	GET	`action=actimetre-new` `mac=`*MAC of Actimetre* `version=`*SW version of Actimetre* `serverId=`*server ID* `bootTime=`*boot time of Actimetre*		Unique (based on MAC) Actimetre ID assigned to this Actimetre
Actimetre offline	GET	`action=actimetre-off` `serverId=`*server ID* `actimId=`*ID of Actimetre*		None
Actimetre removed	GET	`action=actimetre-removed` `serverId=`*server ID* `actimId=`*ID of Actimetre*		None See [Note2]
Actimetre report	POST	`action=report` `serverId=`*server ID* `actimId=`*ID of Actimetre*	Plain text message as sent by Actimetre	OK See [Note3]

The full data Actiserver sends in the POST request contains information on itself, including: its configured server ID, its machine type, SW version number, and disk status. It also contains information on all the Actimetres currently connected to this server, including: MAC, board type, SW version, last boot time, sensor configuration, sampling frequency, disk usage, RSSI and connection quality. This information is used by Acticentral to display the Dashboard.

[Note1] Response `!` means Registry or Project data have changed, and Actiserver needs to fetch new data using Fetch Registry and Fetch Projects queries. Response **`+A:C`** with A=actimId and C=byte to send command C to specific Actimetre. Available command are listed below.

Command	Relayed to Actimetre?	Action
0x10	Yes	Simulate button press
0x20	No	Clean up stale Actimetre data (call `SYNC_EXEC`)
0x30	Yes	Stop Actimetre (see Acticentral section on data management)
0xF0	Yes	Reboot Actimetre

[Note2] This query is issued after all data files for the given Actimetre have been synchronized, i.e. no more data files exist on this Actiserver.

[Note3] When Actimetre sends a Report packet (instead of a data packet), Actiserver relays this information to Acticentral so that the latter can display it on the Dashboard. A Report packet is flagged by the value 0xFF in byte 0, and bits 0-5 of byte 3 contains the number of bytes of payload to interpret as text. Other values in the header are irrelevant.

[Note4] This bit indicates the packet is a Detailed Report message. Bytes 0-2 and 5-7 are accurate, as well as bits 6-7 of byte 3. Byte 4 is meaningless. Bits 0-5 of byte 3 is the size of message, divided by 4. This size includes the null-terminator, which is guaranteed to be present, and zero-padded to the 4-byte boundary.

[Note5] This bit indicates that the packet is a heartbeat message, i.e. has no content at all. This is used to fill the gap when an Actimetre is "Stopped" (not storing data), but the connection needs to be maintained in order to not reboot.

Acticentral

The Acticentral Registry contains the complete list of all Actimetres known to the system, with their corresponding MAC address and unique ID. This data is JSON-encoded and sent to Actiserver as a response the corresponding GET request. The Projects database is a dictionary matching actimId with projectId, and is also replicated in Actiserver. Actiserver needs this information to store data files in the proper subdirectory.

Acticentral publishes monitoring information on its HTTP server at `https://actimetre.fr`. This data is refreshed dynamically, based on the latest data received from the Actiservers. Note that Acticentral does not store any sensor data.

Sensor data is accumulated in the Actiserver serving the given Actimetre, under a folder names `ProjectNN`, under the directory designated by `REPO_ROOT` in `actiserver.conf`. Once a data file exceeds the size limit set by `MAX_REPO_SIZE` or age limit set by `MAX_REPO_TIME`, it is closed and `SYNC_EXEC` is invoked. The command will typically safely copy the data file to a central server, then it must erase the copy in Actiserver.

Actimetres are grouped into Projects for ease of management. This assignment impacts where the data file is stored. When we wish to move an Actimetre to another Project:

1. "Stop" the Actimetre from the Project page. This triggers syncing all remaining data from Actiserver.
2. When the Actiserver is free of remaining data from that Actimetre, the Project page shows a "Move" button to assign this Actimetre to another project.
3. Then we can click "Reboot" to restart the Actimetre, or physically power-cycle it. From that point, the Actimetre's data will go into the new project's folder.

The "Stop" command [Note1] is also relayed to the Actimetre so it can stop gathering data. The Actimetre keeps sending a single Heartbeat packet every second to keep the TCP connection open [Note5]. Whether there is remaining data file in the Actiserver is determined by looking at the data received from its `action=actiserver3` `POST` request. The "Reboot" command is described in [Note1].

When an Actimetre disconnects and doesn't reconnect for more than 5 minutes, Actiserver triggers SYNC_EXEC anyway. The "Sync" button triggers this action manually.

Acticentral software is mostly a large Python 3.9 script that is called via HTTP CGI for handling requests from Actiserver and actions on the HTML page. The same script is also triggered by a `systemd` timer for periodic refreshes of sub-pages. HTTP is served by Apache HTTPD 2.4.

Future work

More testing must be done in order to estimate the optimal ratio of Actimetres to Actiserver. This includes the WiFi bandwidth available to share among all the Actimetres connected to an Actiserver.

Appendix 1 – Cost analysis

Total BOM for 40-sensor system with 40 Solos (ESP32-S2 with MPU-6500) and 1 Actiserver. Prices are taken from the lowest found on AliExpress.com, including delivery to France (free if purchased as a lot). Unit: EUR(€). Acticentral cost is not taken into account.

Fabrication costs not included: wrapping wire, soldering, 3D printing, glue, shrink tubes, labor.

Component	Cost for system	Cost per sensor	Note
Sensor (MPU-6500)	54.00	1.35	
S2 module	96.00	2.40	C3 are slightly cheaper
Power supply (1A)	11.65	0.29	1 per 4 Actimetres
USB 2.0 hub 4-port	17.90	0.45	1 per 4 Actimetres
USB connector	5.60	0.14	lot 10 units
Zero3 (1GB)	23.00	0.58	
128GB microSD (A1 spec)	13.00	0.33	
Power suppy (2A)	3.00	0.08	with cable
External antenna	2.00	0.05	with cable
Total cost excl. fabrication	226.15	5.65	