ESUM

ANTHORPOCENTRIC
URBAN SENSING BACKPACK

Documentation & Setup Handbook
v1

written by
Constantinos Miltiadis

Chair of Information Architecture

ETH Zürich

ESUM

ANTHORPOCENTRIC
URBAN SENSING BACKPACK

Standfest, Matthias; Mitladis, Constantinos; König, Reinhard; Schmitt, Gerhard

Chair of Information Architecture, ETH Zurich, 2015

This handbook was drafted as a quick but exhaustive guide to setup the ESUM backpack by Constantinos Miltiadis who developed the prototype, for the Chair of Information Architecture, ETHz, 2015.

This handbook is designed to be printed as an A5 booklet on A4 paper.

Contents

ESUM Resources (papers)

******ADD PROJECT CONCEPT, PAPER CITATIONS & LINKS******

Backpack Replication Resources

Code currently at:
https://github.com/cmiltiadis

C# PC programs:
******<ADD REPOSITORY LINKS>******
Waspmote Microcontroller programs:
******<ADD REPOSITORY LINKS>******
Fabrication plans:
******<ADD REPOSITORY LINKS>******

Backpack Constituents List

	Description	Make	Model	List Price
1	PC	Intel NUC	DN2820FYKH	CHF 129
2	Wifi sniffer	Libelium Meshlium	AP Scanner	CHF 1035
3	EEG	Emotiv	EPOC +	$499 +
	Wristwatch	Empatica	E4	(request)
	Gas sensors	Libelium Waspmote	Gas Board	
4	Ambiance sensors	Libelium Waspmote	Smart Cities board	
5	GPS	One Talent GNSS	Ducat10 / neoM8n	E 85.40
6	GPS Antenna	Tallysman	TW2410	$ 93
	USB Clicker	Delcom	706503	$ 56
	USB Hub	Logitech	Premium 4	CHF 40
	Battery	Tracer	8Ah	£125
7	Rucksack	ETHz	ETHz Backpack	CHF 15
8	Devices housing	Custom		~35
9	Sensor housing	Custom		~10

Manufacturer: Intel
Model: DN2820FYKH

Intel NUC

The NUC is a barebone PC from Intel.

Specifications & Features[1]:
- Intel Celeron N2820 @ 2.4Ghz
- 1xDDR3L SODIMM
- Intel HD Graphics
- HDMI adapter
- 1x USB3 port (front)
- 2x USB2 ports (back)
- Ethernet port
- Intel Wireless-N 7260BN (WiFi / Bluetooth)
- Audio Jack

Bought separately:
- 120GB Intel SSD[2]
- 8GB RAM

Current draw:
- Idle : 0.37 Ampere @ 12 Volt DC

Installation[3]

The latest drivers can be found at Intel's website[4].

*Requires extra RAM & SSD drive

---

1 http://www.intel.com/content/www/us/en/nuc/nuc-board-dn2820fykh.html

2 We chose an SSD (Solid State Drive) because is based on Flash Chip technology which does not have moving parts as a traditional HDD and therefore is not susceptible to impacts. It is also significantly faster while consuming less energy.

3 The NUC has a visual BIOS preset for Windows 8. To install Windows 7 you will need to change the respective setting in the PC's Bios.

4 https://downloadcenter.intel.com/product/78953/Intel-NUC-Kit-DN-2820FYKH

Manufacturer: Libelium
Model: Meshlium Scanner AP

Meshlium

Meshlium comes in various models, depending on the required functionalities (Wifi, Wifi API, Bluetooth, 3G). ESUM uses the WiFi Scanner AP model.

It is used to measure Wifi and Bluetooth activity around it.
It runs Debian Linux,

Time to start all services: 90 Seconds

Installation & Test:
Connect the required antennas (the Wifi AP model requires 2).

References:
Meshlium Datasheet, Technical Guide, Quickstart Guide[1]

1 http://www.libelium.com/development/meshlium

Manufacturer: Emotiv[1]
Model: Epoc+

1 https://emotiv.com

Emotiv EPOC+ EEG

EEG (Electroencephalography) device

Contains:

- Headset (wireless) with 14 EEG Channels + 2 reference points and 3-axis accelerometer
- USB Bluetooth LE[1] dongle
- 2x airtight felt pad cases
- Bottle of Electrolyte Solution for moisturizing the pads
- 1x USB to Mini USB cable for charging the headset

Installation

Download and install the SDK Lite, accessible from the Emotiv account that the Epoc device was bought. It can be found on the Emotv website under Applications/Essentials[2]

Further:

Emotiv offers other software for their devices such as the Xavier Premium SDK, along with examples in various programming languages on Epoc+ customers, through their Application Store.

*Note: The Emotiv Forum is highly censored[3] towards negative criticism

ESUM EEG Logger

The logger program for the Epoc+ is written in C# and is heavily based over an example by Jim Galasyn[4].

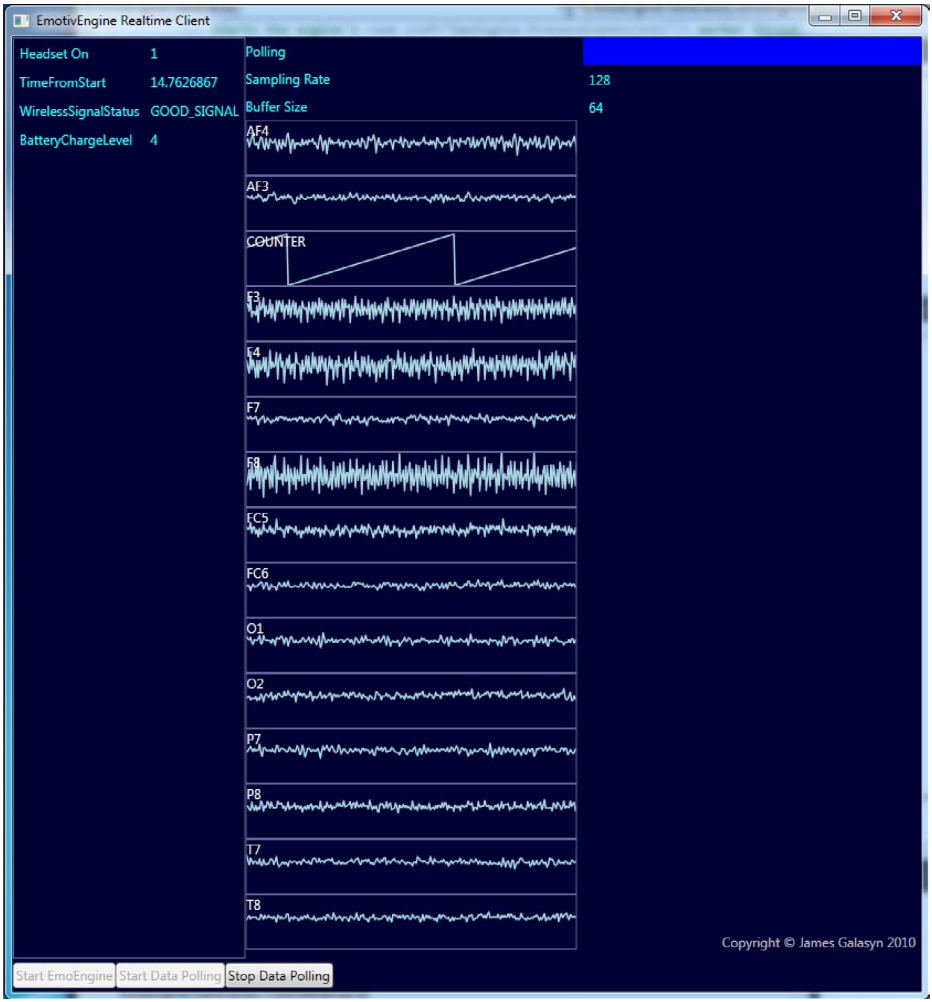
---

1 LE stands for Bluetooth Low Energy. LE is a very different protocol from standard Bluetooth. One of the key differences is that LE devices are not discoverable from a computer's Bluetooth panel, only from applications that "talk" to specific devices.

2 https://emotiv.com/store/product_262.html

3 http://en.wikipedia.org/wiki/Emotiv_Systems

4 http://emotivengineclient.codeplex.com/documentation



The EEG logger running

Hardware preparation:

- Charge the headset with the given micro USB cable.
- Moisturize the pads in their case with the provided solution.

Hardware installation:

- Remove the pads from the case and install on the EEG headset.
- Follow the instructions from Emotiv to place the EEG headset on the test subject.
- Connect the USB dongle on the computer.
- Switch the headset On.

Logging the data:

- Start the ESUM EEG Logger program.
- Click Start Client
- When the program finds the headset, click Start Polling.

Manufacturer: Empatica
Model: E4

Contains: Wristbands / USB base / USB to mini USB cable

Empatica E4 Wristband

The wristband has sensors for:

- Photoplethysmography (PPG) - Continuous Heart Rate (Heart Rate Variability, Stress, Relaxation)
- Electrodermal Activity (EDA) - Skin conductance i.e. Galvanic Skin Response (Arousal, Excitement)
- Temperature and Heat Flux- Activity, Context Info
- 3-axis Accelerometer - Movement, activity

The E4 offers two modes of function

- Streaming mode - Sends the data to a mobile app in real-time
- Record mode - Records the data internally

ESUM uses the Record mode. The Wristband can hold sensor data for 30 hours on its on-board flash memory.

Setup:

1. Charge the wristband via USB through its USB charger
2. Create an account on the Website of Empatica
3. Install Empatica Manager[1][2] (OSX & Win)

Button Pressings:

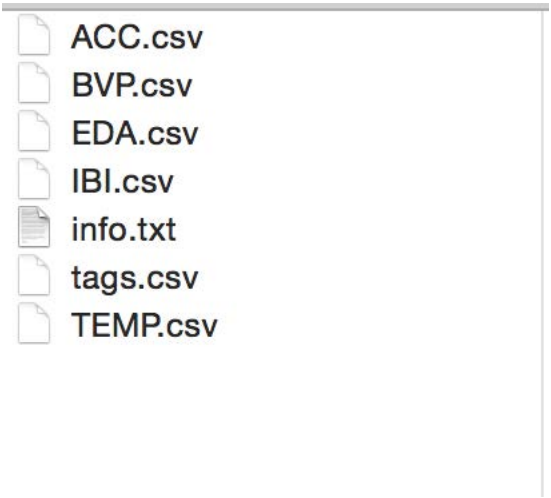
- Short Press: Momentary. When On, will log an event mark. When Off will not result in anything
- Standard Press: 1-3 Seconds - Switches the device On/Off
- Long press: >5 Seconds - Resets the device * CAUTION * after a Reset the device might need to be connected to a computer over USB to be reactivated.

In use

1. Wear the wristband, and do a Standard Press (until the LED flahses

1 https://www.empatica.com/product-e4-download

2 http://support.empatica.com/hc/en-us/sections/200008246-Empatica-Manager-Application-for-OSX-and-Windows



E4 output files

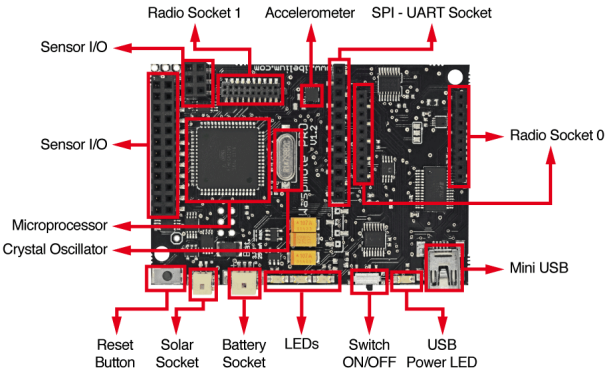
Green - then release it). The wristband searches for a Bluetooth LE connection. If it doesn't find any, in 40 seconds, it goes into Record mode, by blinking a Red LED for 20 seconds.
2. At any moment do a Short press to log an event mark.
3. End the Recording by a Standard Button press.

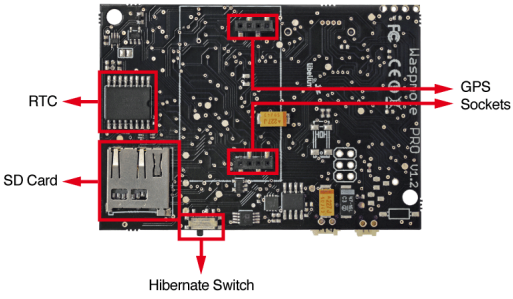
Data Retrieval

1. To get the data, connect the wristband to the computer over USB, and start the Empatica Manager, while being connected to the internet.
2. Go to the Empatica website and log-in.
3. On the calendar you can see your seesions.
4. By selecting one you can View/Download or Delete it.
5. By downloading you get a zip file with 6 CSV files, one for each sensor. There is also an "info.txt" file that explains the data format.

References:
Empatica E4 FAQ page[3] including LED status guide and an operation guide.
Empatica Manager FAQ page[4]

3 http://support.empatica.com/hc/en-us/categories/200023126-E4-Wristband
4 http://support.empatica.com/hc/en-us/sections/200008246-Empatica-Manager-Application-for-OSX-and-Windows





Manufacturer: Libelium
Model: Waspmote
List price: E 158 (with Roving Networks WiFi radio w/out battery)[1]

1 http://www.cooking-hacks.com/shop/waspmote/core-boards/waspmote-wifi-sma-2-dbi

Libelium Waspmote Platform

The Libelium Waspmote[1] is an "Internet of Things" microcontroller platform developed by Libelium and based on the Arduino microcontrollers, for the purpose of sensor data logging in a wireless sensor network.

Specifications[2]:

- Microcontroller: ATmega1281
- Frequency: 14MHz
- SRAM: 8KB
- EEPROM: 4KB
- FLASH: 128KB
- SD Card: 2GB
- Weight: 20gr
- Dimensions: 73.5 x 51 x 13 mm
- Temperature range: [-10ºC, +65ºC]
- Clock: RTC (32KHz)

Features:

- mini USB connector
- micro SD slot
- 3.7v LiPo Battery connector - w/ special plug
- connector for GPS module
- connector for Wireless module
- 3-axis accelerometer

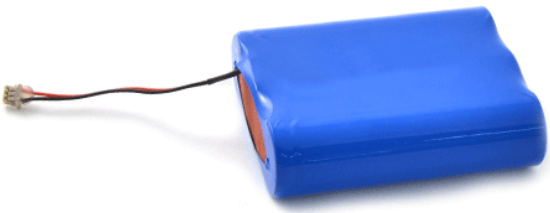
Switching On/Off:

- Push the switches towards the USB connector to power OFF.
- Push the other way for switching ON.

Installation

- To program the Waspmote you need to download the "Waspmote

1 http://www.cooking-hacks.com/documentation/tutorials/waspmote
2 http://www.libelium.com/products/waspmote/hardware/



Pro" IDE[3] which is identical in function to the Arduino IDE, and containes all the required libraries that the Waspmote m/c use.
- Also make sure that you have installed the latest FTDI drivers[4].

Batteries:
- The Waspmote microcontrollers *NECESSARILY* need a battery in order to function, regardless of if they are connected to a power source via USB. Otherwise they will output an error (E#).
- Libelium offers a high capacity 6600MA/H LiPo battery[5] for their Waspmote range.
- The battery level can be monitored (in percentage) by the Waspmote microcontrollers.
- The specific battery needs about 8h to charge.
- Note that the battery plug is not very common - not very easy to interchange between 3rd party batteries.
- It is not adviced to leave the battery empty completely.

References:
- Waspmote Quickstart Guide, Technical Guide, Datasheet[6]
- Libelium has a forum that offers technical support[7]

3 http://www.libelium.com/development/waspmote/sdk_applications/
4 http://www.ftdichip.com/Drivers/VCP.htm
5 http://www.cooking-hacks.com/shop/waspmote/accessories/6600ma-h-rechargeable-battery
6 http://www.libelium.com/development/waspmote/documentation/
7 https://www.libelium.com/forum/

Model: v2
List Price : E 120[1] w/out sensors

1 http://www.cooking-hacks.com/shop/waspmote/sensor-boards/waspmote-gasses-sensor-board

Gas Sensor Board

A Sensor board that plugs on the Waspmote microcontrollers and can measure Temperature, Atmpspheric pressure, plus 7 of 14 different kinds of gases.

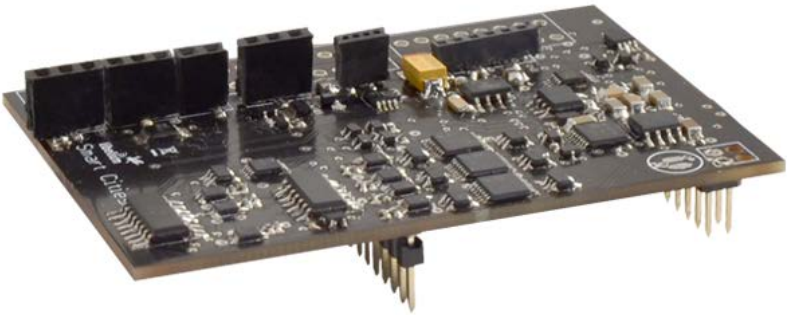
ESUM uses the following sensors:

- Temperature sensor
- Atmospheric pressure sensor
- Humidity sensor
- CO2 - Carbon Dioxide
- O3 - Ozone
- NO2 - Nitrogen Dioxide
- Air pollutants I : C4H10, CH3CH2OH, H2, CO, CH4
- Air pollutants II : C6H5CH3, H2S, CH3CH2OH, NH3, H2

Technical Guide[1]

*Note: You have to have the battery charged and connected in order to read the sensor values.

1 http://www.libelium.com/development/waspmote/documentation/gas-es-board-technical-guide/



Model: v2
List Price : E 120[1] w/out sensors

1 http://www.cooking-hacks.com/shop/waspmote/sensor-boards/waspmote-smart-cities-sensor-board

Smart Cities Board

A Sensor board that plugs on the Waspmote microcontrollers, used measure qualities of outdoor environments.

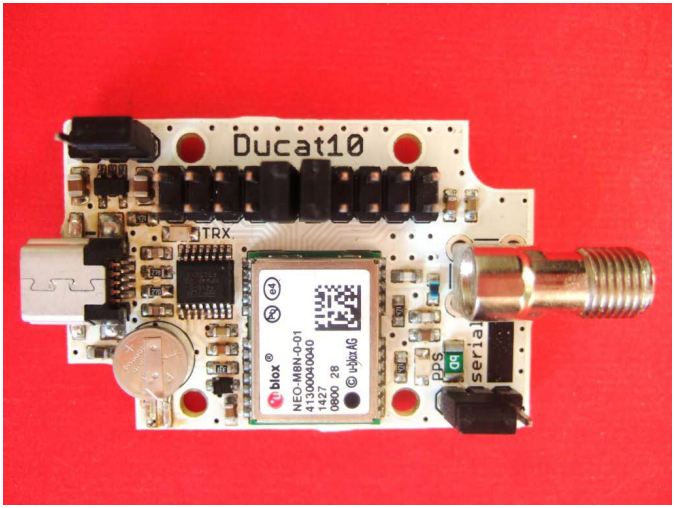
ESUM uses the following sensors:

- Luminosity sensor
- Dust sensor
- Sound pressure sensor

Technical Guide[1]

*Note: You have to have the battery charged and connected in order to read the sensor values.

1 http://www.libelium.com/development/waspmote/documentation/smart-cities-board-technical-guide/



Model: Ducat 10 / Ublox Neo-M8N[1]
Antenna: TW2410[2]

1 http://www.onetalent-gnss.com/ideas/usb-hw-receivers/ducat10
2 http://www.tallysman.com/TW2410.php

GPS Module

Installation
To install the module's drivers, you can follow the instructions found on the M8 product page[1]. After installing and restarting your PC, you should be able to receive GPS data through a Serial Port Monitor (eg. Arduino).

Troubleshooting
In case of trouble you can contact the support team of One Talent which is happy to help[2].

Monitor & Edit
uBlox offers u-Connect[3] a software designed for their GPS modules, which can monitor the data and also edit the parameters of the module.

Notes
GPS technology does function only outdoors and only when it has an optical connection to a satellite. Dense urban environments might obstruct its connectivity.

1 http://www.u-blox.com/en/drivers-a-middleware/usb-drivers/windows-7-driver.html
2 http://www.onetalent-gnss.com/contact-us
3 http://www.u-blox.ch/en/evaluation-tools-a-software/u-center/u-center.html

Manufacturer: Delcom
Model: 706503

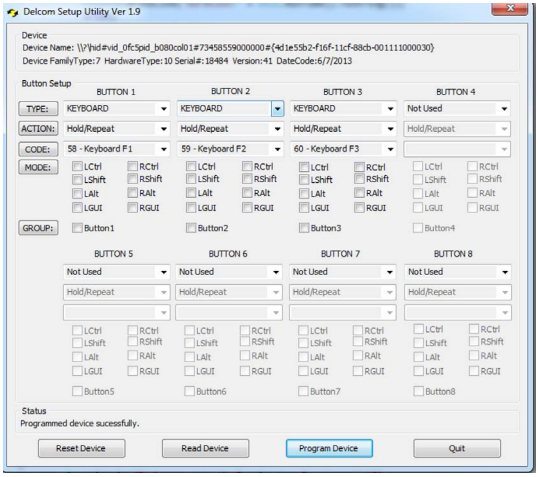
USB Clicker

Installation
The Clicker is HID (Human Interface Device) therefore it needs a software which binds its keys with mouse or keyboard functions. Delcom offers a software for this purpose which can be found on the clicker product page[1].

Key Bindings
Through the software you can set the function for each key, and save the configuration. Once done, you don't need to use the software again (only in case you need to change the configuration).

When connected to a PC the clicker buttons light up.

The bindings for the prototype are (also saved in the logger program)
- Button 1 - F1 (RED)
- Button 2 - F2 (ORANGE)
- Button 3 - F3 (GREEN)



1 http://www.delcomproducts.com/productdetails.asp?PartNumber=706503

Manufacturer: Logitech
Model: Premium 4-Port USB Hub
List price: ~E 40

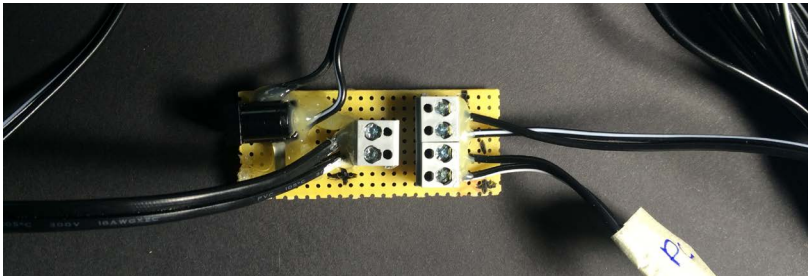
USB HUB

The backpack requires a number of USB connections between sensor devices and the PC. The NUC PC offers already 3 USB Connections.

Required connections:
- Waspmote Smart Cities
- Waspmote Gases
- Emotiv Epoc EEG
- GPS module
- USB Clicker

Extra dock station connections:
- Keyboard
- Mouse

For this reason the ESUM prototype uses a simple 4xUSB 2.0 HUB, that is passive (does not require external power source).



Battery circuit

Manufacturer: Tracer
Model: 12V 8Ah

Battery

The ESUM prototype uses a high capacity LiPo battery to power the Computer and devices. The devices that require external power are the PC and the Meshlium. Both operate on 12 Volts DC.

Battery requirements:
- Voltage DC: 12 Volt
- Peak current: > 3 Ampere

Device Power Consumption at 12 Volts DC:
- Intel NUC
  - Idle in Windows: 0.37 Ampere
  - Running the logger programs: 0.60 Ampere
- Libelium Meshlium
  - Idle/Running: 0.25 Ampere

Total Power consumption: ~0.85Ampere rounded to 1 Ampere

Therefore a battery of a capacity of X Ah, with a theoretical consumption of 1 Ampere can last X hours.

Battery circuit:
Connect the battery out, to the wires powering the devices, in parallel.
Also solder a connector for charging the battery.
Isolate and fix all connections, on both sides with a glue gun.

*NOTES
Make sure that you apply the correct polarity, otherwise there is a significant risk of burning the devices. Also that there is no short-circuit and the connections are clean and firm.
Use the correct connectors (plugs diameters and polarity) for powering the devices.


ETH zürich

ETHz Rucksack & Housing

In order to make the project portable we needed to find a backpack that could house the devices and sensors, in a way that would be comfortable to carry. Most specialized backpacks are very expensive and difficult to customize so we chose to design our own removable housing for the devices, that could be inserted to a normal backpack. We used the ETHz backpack that can be found at any ETH Store or online[1].

It was selected because it is:

- fairly small
- durable
- made from waterproof fabric
- has padding on the back
- branded
- does not draw attention
- cheap (15-30CHF)
- readily available from any campus any campus (SAAB at Hongger-berg / ETH Store at Central)

1 ETH Store website http://www.eth-store.ch/rucksack.html

device & sensor housing prototype 1

Cost:

- plywood 2x7 CHF
- laser cutting ~7 CHF
- wood glue 5
- Velcro tape ~8 CHF

Device housing

The device enclosure is made of 4mm plywood which has a very good ratio of strength to weight. 4mm is also the largest plywood thickness that a common laser cutter can work with, and its a common material that can be bought in sheets -in our case from the material shop in HIL/ ETH Honggerberg.

The sensor housing was made of 3mm plexiglass to ensure light exposure.

The housing measures 320x250x130x mm

It was cut in under 5 minutes in the Raplab[1] facitilies of ETH Honggerberg, and assembled, ready to go in half an hour.

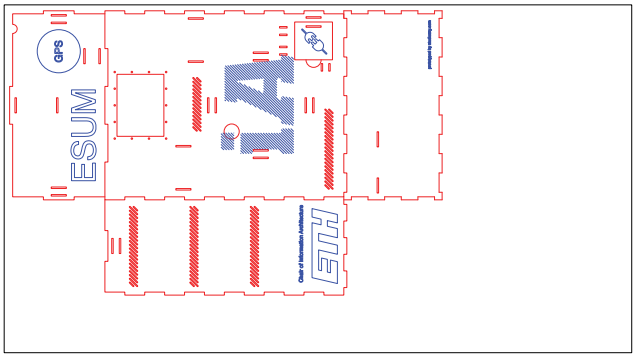
Without any other procedure (eg. sanding) it can be assembled and glued with a standard wood glue, and ready to go in under 30 minutes.

Laser Cut Process
1. Place the plywood sheet on the laser cutter surface and close the lid
2. Press the Auto-focus button
3. First send the Engraving job and then the Cutting
4. Remove the pieces from the machine and repeat for the second sheet

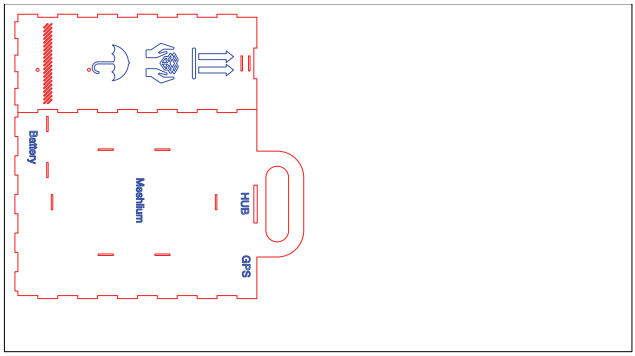
The settings used on a 120 watt Laser Cutter
- Engraving: Speed 50% Power 10%
- Cutting: Speed 40% Power 80%

Assembly
1. Put the pieces on a flat surface and fit them together.
2. Starting with the back panel, the sides and bottom panel, apply glue

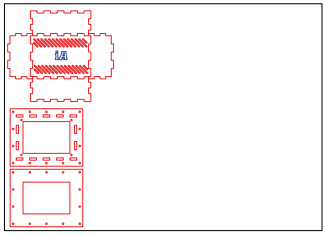
1 Raplab website: https://www.raplab.arch.ethz.ch



Device housing 1, Plywood board 1- 4mm 83x46 cm blank



Device housing 2, Plywood board - 4mm 83x46 cm blank



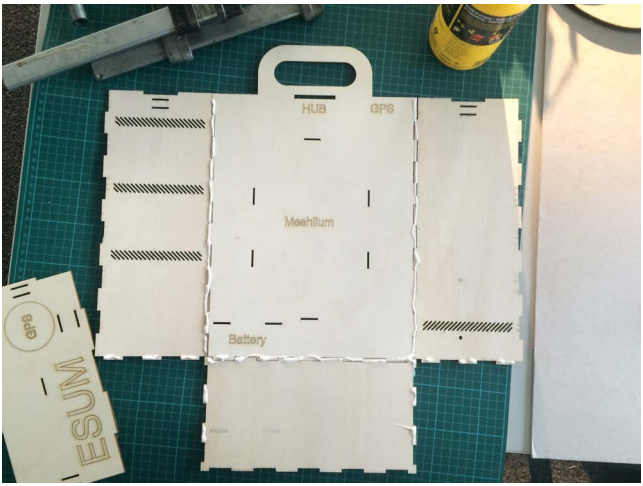
Sensor housing, Plexiglas - 3mm 42x30 cm blank

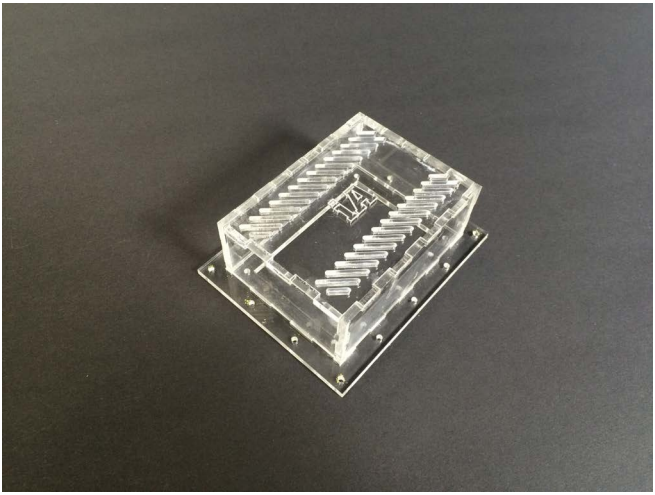
Unrolled surfaces ready for cutting

to the overlapping teeth.
3. Put together and fix tight for 15 minutes for the glue to cure.
4. Place the devices inside, on their marked spots and fix with Velcro tape.
5. Glue the front plate and fix tightly for 15 minutes.
6. Ready!

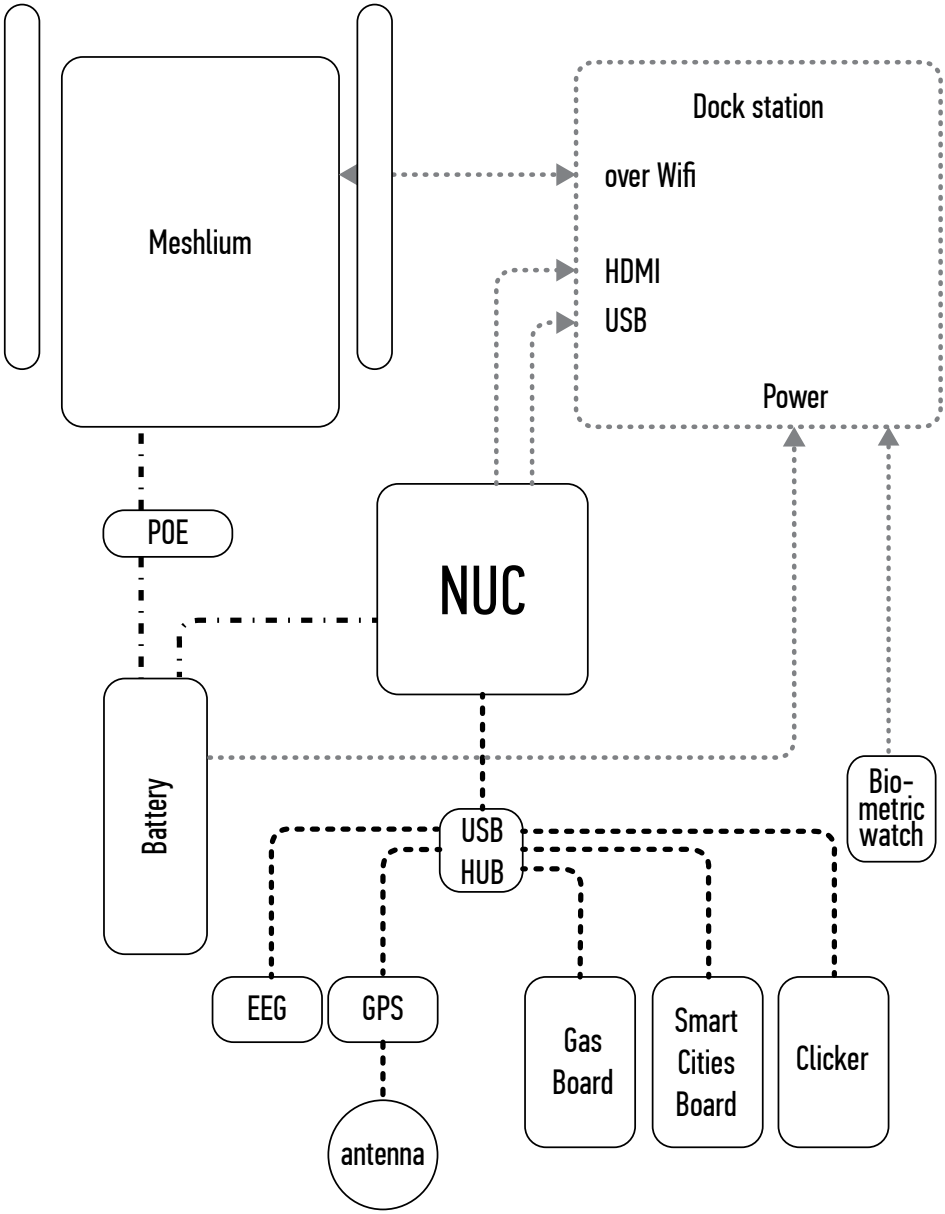
Backpack Preparation:
Some holes must be cut on the backpack fabric. The first is to have access to the PC power button, the second for charging cables and clicker accessibility and the last for the atmospheric sensors to be exposed to air and light. To cut these holes, place the electronics housing in the backpack and mark a cutting outline. Make sure to cut rounded edges, to minimize the risk of tear.
For fixing the area around the holes, since the backpack's fabric has a plastic coating, instead of sewing, we used a silicon based textile binder glue (GD200[2]). Ideally apply one binder coating before the cutting and a second afterwards.
To pass the screws that connect the plexiglas sensor housing to the plywood box through the backpack fabric, we used a leather hole punch tool - which can also be used to make small ventilation holes on the fabric.

2 http://www.cht.com/cht/web.nsf/id/pa_en_productdetail.html?Open&pID=00029521.104_EN


HUB
GPS
Meehilum
Battery
GPS
ESUM



Device Assembly



ESUM Logger Program

The Esum Logger program was written in Visual Studio C#.

The basic class structure is:

- Program Class - calls and creates all the objects, contains loop
- Device Class - Handles Serial Devices
- Util Class - holds the program variables and auxiliary functions, such as: time stamps, beeps and file exporting.

Program Structure:
The basic structure follows symmetrical arrays and enumerators

- Device [] devices
- Enum DeviceName
- Enum DeviceComPort
- Util.deviceBaudRate []

*NOTE: These are symmetrical and should not be altered appart from the DeviceComPort enumerator values which differs from one PC to another.

Important Variables (Util class):

```
//Start automatically, or request the Rescan function from input
//clicker
public static bool StartPortsOnAwake = false;

//save data buffer to file after this amount of input frames
static int numOfFramesToExport = 2;
```

Setup:
Before you run the program, you have to now which COM port corresponds to which device. By default Windows assigns a unique COM port to each device (eg. COM11).

Export path:

The program finds the Documents folder of the PC and exports the data in a sub-folder of the structure:
.../Documents/ESUM_Export/Session_<TIME>/<SensorName>.CSV

Export format:
The data are exported in CVS files, in the form of <DeviceName>.csv . CSV format files can be read with both text editors and Excel type software.
The structure of the file is:
<DetailedTimeStamp> , <Value1>, <Value2>,<Value3>,....
<DetailedTimeStamp> , <Value1>, <Value2>,<Value3>,....

The program has some Commands that can be typed in the console:
- Status - prints the status of the devices
- Rescan - rescans ports
- Clear - cleans the screen
- Monitor - toggles printing device input
- Beep - toggles beep sounds
- Duration - prints the amount of time since the program started
- Help - prints the console commands
- Exit - exits the program

Notifications:
- On start, the program does an ascending bleep.
- On device synchronization, the amount of bleeps equivalent to the index of the device, and then a long bleep.
- On exit, it does a descending bleep.

Known issues:
- The program cannot know -for the moment- if a device that its already connected to has been disconnected.

Issues to look into:
- The maximum file size of the CSV format.

Chair of Information Architecture