AAL Programme

SAVE

SAfety of elderly people and Vicinity Ensuring

AAL-CP-2018-5-149-SAVE

01/09/2019 - 31/08/2022

AAL Programme

Project - SAfety of elderly people and Vicinity Ensuring - "SAVE"

Deliverable: D.1.2 Sensors and Sensors Networking Description

Version: 1.0

Main editor: VS

Contributing partners: UNITBV, ISS

Table of contents

1. Sensor networking

The SAVE solution uses several kinds of sensors (wearable and ambient) to get information about the elderly's well being in his habitat (home, vicinity, etc) and to assure his/her security. Also, the solution has some features that the elderly can use in case of danger (SOS button and fall detection).

Figure 1 presents an overview of the SAVE solution, highlighting the actors and some technical aspects.

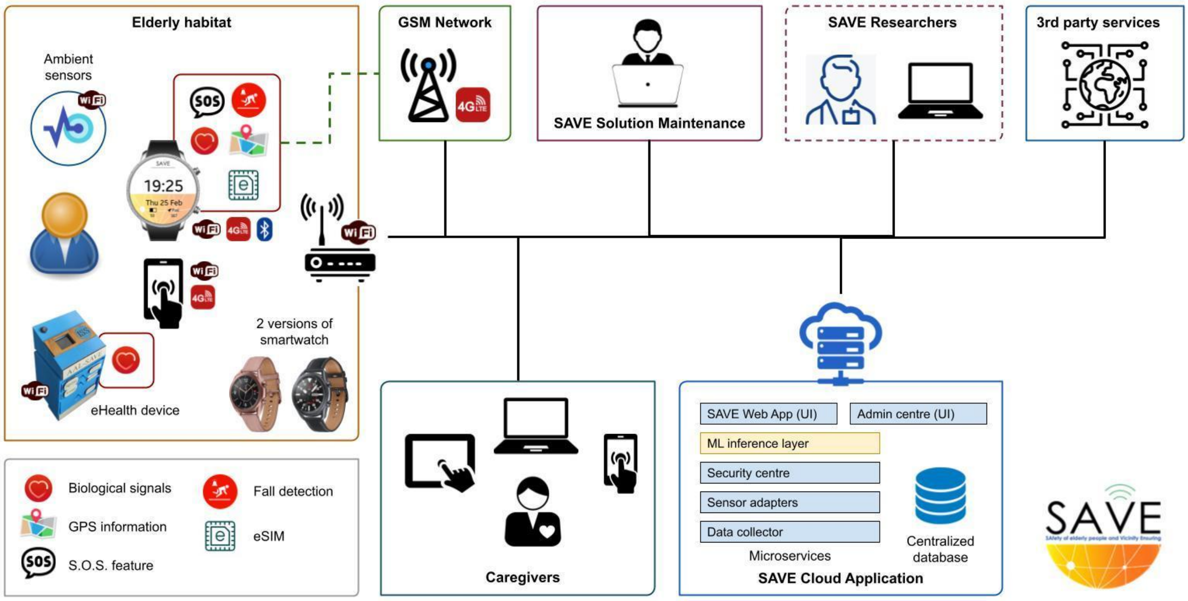

Figure 1 - SAVE solution overview

The SAVE solution presents the elderly with a kit of sensors containing, minimally, an eHealth device and a smartwatch (Samsung Galaxy Watch 3) with the SAVE software pre-installed. The kit can include other ambient sensors (e.g. presence, magnetic contact, flood sensor), that will connect to the SAVE cloud application. Also, the elderly receives the credentials for accessing the SAVE web application. The elderly can request supplementary credentials for his/her caregivers.

For the pilot sites, assistance will be offered, when needed, by the SAVE solution maintenance to the caregivers for configuring the SAVE software on the smartwatch and to do other required configurations (connecting to the WiFi, installing eSim, configuring sensors, etc).

The eHealth device is actually a complex sensor, made up from several individual devices for measuring some biological parameters (e.g. blood pressure, pulse, temperature) and the activity (e.g. body position) of the elderly. This device connects to the Internet through the local WiFi network.

The smartwatch acts as a wearable sensor and connect to the Internet through:

- The GSM network, if the LTE function is used and an eSIM is present (this is the recommended scenario by the SAVE project).
- WiFi, by connecting to the local WiFi network or to a hotspot opened by the elerly's smartphone (that have mobile data activated). In the current phase of the project, for this case some safety features will not be available (SOS and fall detection).
- A tethering Bluetooth connection. In the current phase of the project, for this case some safety features will not be available (SOS and fall detection).

The Samsung Galaxy Smartwatch 3 includes a vast array of sensors that can be used to assess the physical activity (steps, steps frequency, speed of movement) and to read some basic biological signals (pulse) that will be used in the analysis of the user's well-being. The GPS sensor, together with the gyroscope and the accelerometer are used for ensuring the elderly's safety (location tracking, fall detector).

The smartwatch selected for the pilots offers LTE communication and includes in its firmware the SOS and fall detection features. The elderly can call for help by quickly pressing 3 times the power button. The smartwatch sends SMSs containing a standard message and the location of the smartwatch's wearer (Figure 2) and then calls the designated caregiver.

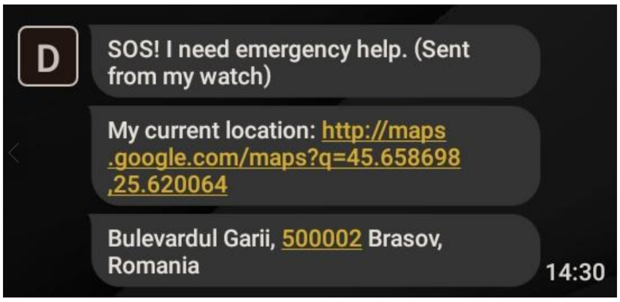

Figure 2 - SOS SMS messages

In case a fall is detected, the smartwatch will display a 60 second warning to the elderly; if the warning is not dismissed, the smartwatch will send SMS messages and call automatically a designated contact person (a caregiver).

The SAVE minimal kit can be complemented by readily available on the market sensor kits (e.g. Mi Smart Home kit). However, these kits have their own user interfaces, but once configured does not require any further human intervention. These kits usually include devices that can emit sound warnings when some situations are detected (door left opened too long, unexpected presence in some perimeter, etc).

Work is carried out inside the SAVE consortium to include organically readily available sensors into the SAVE minimal kit. Also, some cheaper alternatives for the smartwatch are being investigated (e.g LilyGo programmable watch).

All the sensors included in the minimal kit communicates with the SAVE cloud application, through the Internet, feeding it with sensor data and receiving commands, when needed. The preferred way of communicating in the SAVE sensor network is WiFi.

1.1. SAVE solution general architecture and data organizing

The general architecture of the system, described in Figure 3, is composed of several independent components that communicate with each other to serve certain functionalities.

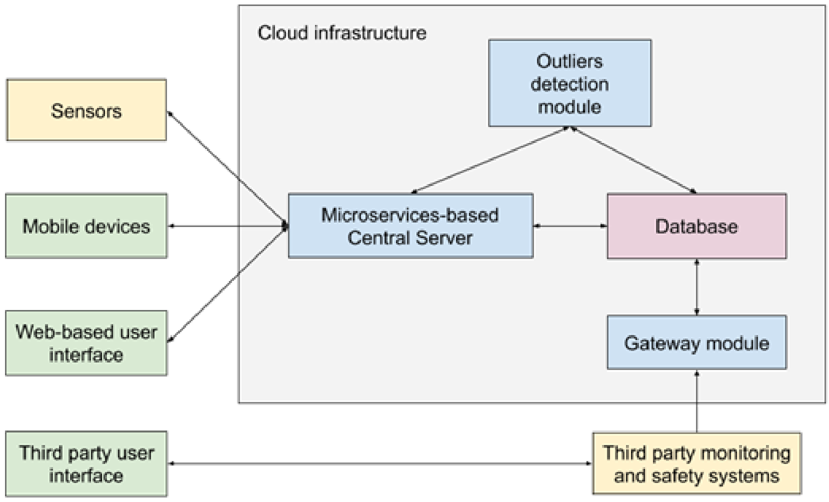

Figure 3 - General architecture of the SAVE system

The system has several main components that are hosted in the cloud and perform specific tasks. Therefore, the system is built upon a microservices-based architecture that allows its running on a cloud infrastructure, scalability, and speed in developing new features.

Furthermore, the communication between the main components of the system is enhanced by the interaction of the cloud hosted microservices with external systems, such as sensors, mobile devices or web applications that provide data and other monitoring applications that can process data retrieved from the SAVE system.

In a more detailed view of SAVE architecture there are several main components consisting of microservices for collecting data from the devices, for data processing, for user and external systems interaction, for user interfaces, all of these being complemented by a Relational Database Management System (*MySQL*).

The communication, either internal or external, is standardized using HTTP Protocol, hence every microservice in this chain is exposing a REST (Representational State Transfer) interface. Every message exchange in this context is JSON formatted, this way the communication is easy to realise.

Also, the communication channels are secured using the Public Key Infrastructure (PKI). SSL / TLS certificates and a hybrid cryptosystem are used for this purpose. User interfaces access is achieved through the HTTPS secured protocol, requiring an authentication with username and password and authorization mechanism.

Along with the microservices-based architecture the system implemented using Java programming language with the help of Spring Boot Framework in order to take advantage of its features.

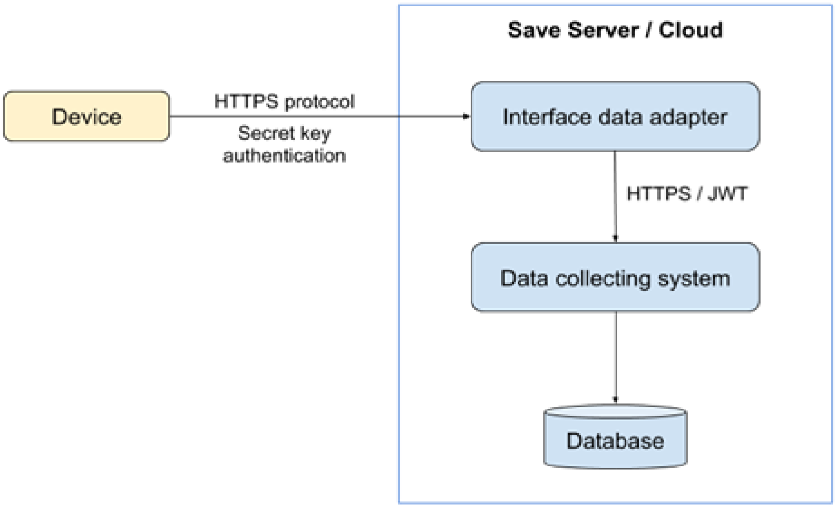

Figure 4 - System communication model

Regarding the data persistence a relational database is used which provides a more efficient solution for organizing information in the system. On top of that resides the most important components of the system, which is the *data collection module*. It is also designed on the principle of microservices-based architecture in order to achieve a standardized data collection process and easy adaptation in case of devices providing data changes.

The flow of collecting data is simple, the device sends the collected data through HTTPS protocol to an *interface data adapter* which transforms it into a standard format and transfer it to the data collecting systems which will save it to the database.

The data adapter interface is a module developed for each type of connected device with the specific role to expose a communication interface with external devices and to convert the data received from them into a standard format recognized by the collection system. This ensures an abstraction and standardization for the communication between the SAVE system and its connected devices.

Data collection system is the module with the purpose of receiving information from an adapter, computing, and persisting it into the database.

The security of these communication channels, internal or external, is done using HTTPS protocol, secret key authentication, and JWT (JSON Web Token)

Once in the system and transposed into a standard format, data from the devices is organized and saved. The figure below illustrates how the data is organized.

Data is collected from specific devices, that need to be prior registered into the system. This registration implies setting a name for the device in order to be easily identified by the user, a type that also defines the adapter interface it needs.

With these attributes of the devices, the data collected from them is store into the database.

In order to optimize the speed of data access, the devices are organized in kits, organization that defines the partitioning scheme for the data collected. in this way, the data from a specific kit is stored in a single specific table. The data storage mode is a general one, allowing the saving of any data structure.

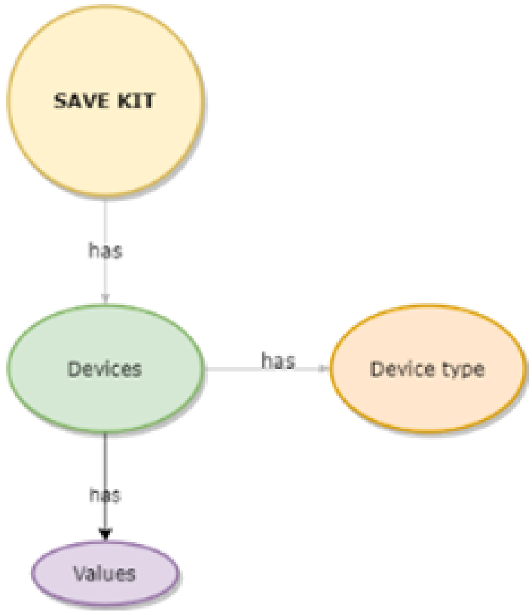

Figure 5 - Data organization in the SAVE system

The data collected from the eHealth sensors and from the Samsung Galaxy Watch 3 is received in a JSON format and it is saved into the database.

An eHealth JSON example is listed below:


```
{
   "EQUID":"1234",
   "USID": 2,
   "SPO2":[78,98],
   "TEMP": 366,
   "BLOODPRESSURE": [125,79,84],
   "SPIROMETER": [22977,22977],
   "GLUCOMETER": 95,
   "BODYPOSITION": [5],
   "SCALE": [9220,283,606,525,1940,33,13],
   "BUTON": [1]
}
```


Where:

- *EQUID* contains the equipment identifier number which transmitted the data.
- *USID* contains the user identifier number.

- *SPO2* represents the oxygen saturation. The first value is the pulse (78 bpm), while the second one shows the oxygen level in the blood (98%).
- *TEMP* stores the registered human body temperature (36.6 degrees Celsius).
- *BLOODPRESSURE* has the values of the measured blood pressure. The first value represents the systolic blood pressure, while the second one is the diastolic blood pressure (125/79 mmHg). The third value corresponds to the heart rate (84 bpm).
- *SPIROMETER* stores the data by digital spirometer,
- *GLUCOMETER* stores the blood sugar level (95 mg/dl);
- *BODYPOSITION* contains values registered by the body position detection sensor: 5 - standing or sitting, 4 - stomach, 3 - right, 2 - left, 1 – back.
- *SCALE* stores the data registered by digital scale: mass, body fat, muscle mass, water, calories, bone mass, visceral fat.
- *BUTTON* stores the state of a button (on/off).

A JSON representing the data collected from the Samsung Galaxy Watch 3 sensors:


```
{
   "stepStatus":"WALKING",
   "speed":4.300000190734863,
   "walkingFrequency":1.5,
   "heartRate":80,
   "steps":44
}
```


Where:

- *stepStatus* contains the user's movement type. Could be "NOT_MOVING", "WALKING", "RUNNING", "UNKNOWN".
- *speed* stores the user's moving speed (4.30 km/h).
- *walkingFrequency* represents the step count per second (1.5).
- *heartRate* contains the value for user's heart rate (80 bpm).
- *steps* stores the cumulative walking and running step count (44).

2. eHealth Monitoring System

The eHealth objectives consist in automatically collecting biometric data (e.g., unique physiological, physical, behavioural data) from elders on site and further transmitting the data to caregivers/volunteers for achieving the proper communication value chain.

According to the Object-Process Diagram in Figure 6, there are two regimes of user interface functionality that may emerge in parallel, as follows:

- Regime of system development and serviceability: the biometric data is displayed on site and further transmitted via the internet, exclusively for the developers.
- Regime of daily usage by the elders: the biometric sensor data is not displayed on site but transmitted exclusively via internet to caregivers/volunteers outside the eHealth system boundary.

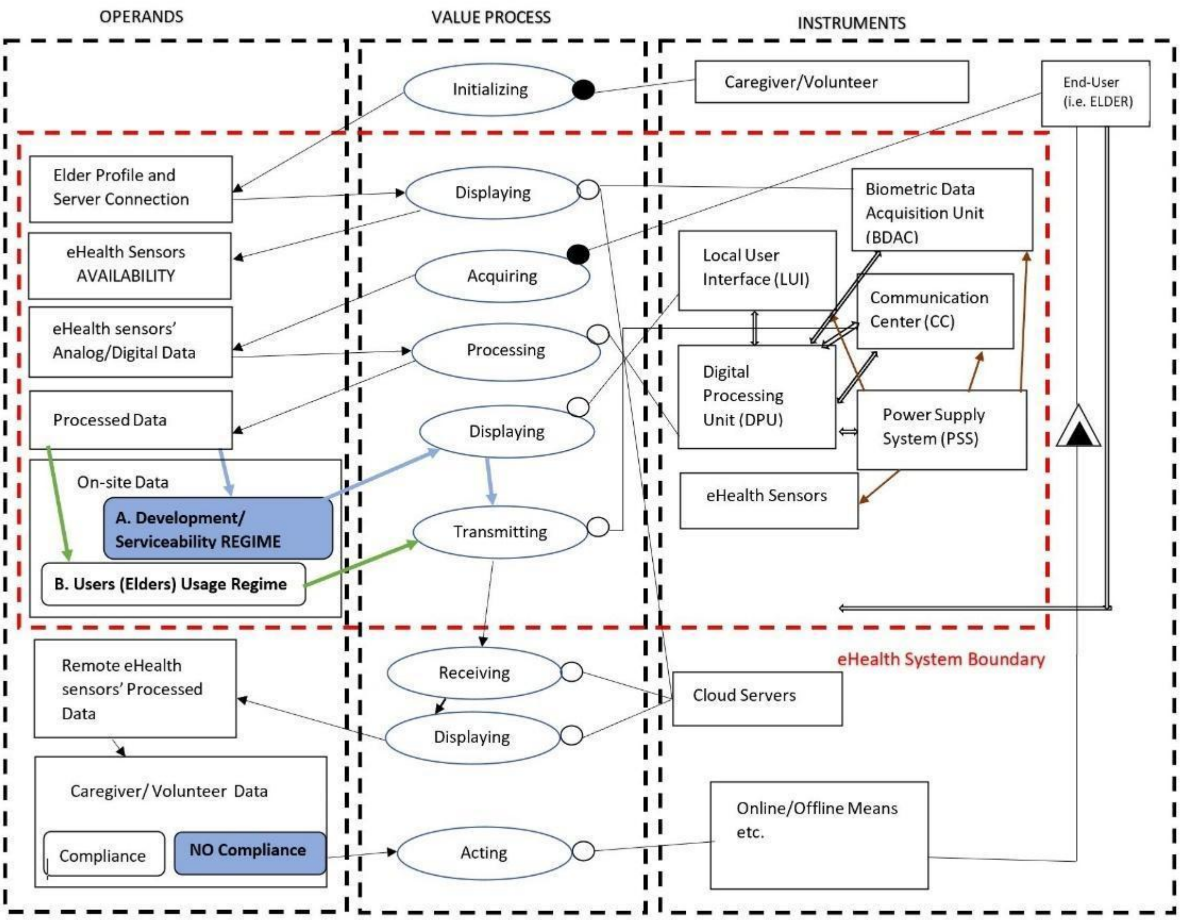

Figure 6 - eHealth system architecture in the form of Object-Process Diagram (OPD).
Figure source: Institute of Space Science

SAVE project's main objective regarding eHealth sensors component is that it can offer eHealth capabilities at home monitoring for elderly people, over 50 years old, suffering from age-related chronic illnesses, mild cognitive issues/disabilities, or cognitive decline.

The service can be implemented as an Online Health Service (OnHS) and/or Offline Health Service (OffHS), both involving e-Health sensors used on-site at home. As settled at the level of technology, for complying with one of the main requirements of the European AAL perspectives in terms of interoperability and open interfaces for achieving a European market, the eHealth system concept is oriented towards Open-Source Hardware (OSHW) and COTS eHealth Platforms including COTS biosensors.

The eHealth system is based on the low-cost eHealth Platform from Libelium Company, namely MySignals Hardware (HW) Development Platform - eHealth and Medical IoT Development Platform for Arduino. The eHealth system concept, comprises short and long-range communication protocols as a Wi-Fi and Bluetooth4-enabled therefore scalable base station that offers two services:

- embedded C++ application: offers connection and readout of biometric sensors. It shares data with the cloud via a web service.

- web application: allows management and configuration for the base station. This option was considered better than OS-dependent smartphone app (iOS/Android). It will enable configuring the specific sensor, triggering a certain measurement, displaying the result of the measurement, and setting the connection parameters used for cloud interaction, from any device connected to the same network as the base station (PC/laptop/smartphone).

2.1. Description of eHealth functional blocks

Taking into account the eHealth system architecture in the form of Object-Process Diagram in Figure 1, it can be extracted from the instruments inside the eHealth System Boundary several eHealth functional blocks which will be briefly described below.

2.1.1. Biometric Data Acquisition System

The biometric data acquisition system must integrate Commercial off the shelf (COTS) Open-Source Hardware (OSHD) components for data acquisition in digital and/or analog format, data processing and transmission using analog-to-digital converters, multiplexers, standard communication protocols, etc.

2.1.2. eHealth Sensors and Devices

The e-Health sensors and devices are a customizable biometric sensors package for the primary target group of the SAVE project. According to the system requirements (SR), the considered biometric sensors are blood pressure sensor, pulse oximetry sensor, body position sensor, temperature sensor, glucometer, spirometer, scale with information on body weight, fat percentage, muscle mass percentage, degree of hydration, calories needed to be burned daily, percentage of bone mass, percentage of visceral fat. The acquired data by the e-Health central unit from the sensors (intended to be used individually) are collected, processed, and automatically sent to the SAVE Cloud

2.1.3. Power Supply System (PSS)

The system must be autonomous for several hours with the screen ON and all sensors in operating condition. The user must be able to charge the system while it is in use. The system must function in "always-on" mode and not enter in standby mode; the e-Health system is turned off using an ON/OFF button.

2.1.4. Digital Processing Unit (DPU)

The Digital Processing Unit (DPU) must communicate to transmit information between the Communication Centre, Local User Interface, Biometric Data Acquisition System and Power Supply blocks.

2.1.5. Client - Server Communication Unit

Using the network card Wi-Fi SoC in order to send information to the Cloud server, the system will communicate via Wi-Fi with the user's home wireless router. The system is intended to communicate with the server using an encrypted channel with HTTP protocol and secured with SSL/TLS certificates, to send data only when it has real values recorded from the sensors and at a configurable period of about 10 seconds.

2.1.6. Local User Interface (LUI)

The Local User Interface (LUI) system must display locally the biometric data if it has found at least one Bluetooth-enabled sensor and the sensor pairing. The system, according to the system architecture in Figure 6, has 2 operating modes, as follows:

i. the biometric parameters will be displayed on-site in the development/service mode;
ii. ii. in normal user mode (use by the elders), the biometric parameters will NOT be displayed on-site, except for the sensor availability information.

2.2. *eHealth sensors networking*

2.2.1. Biometric Data Acquisition System

Biometric Data Acquisition System was developed implementing the Libelium eHealth COTS platform, namely MySignals - eHealth and Medical Internet of Things (IoT) Development Platform (Figure 7), a development board that allows connecting and processing data collected from biometric sensors. More precisely, the platform consists exclusively of a shield with outputs for Libelium sensors. Based on this platform, a whole integration process of hardware and software implementation was implemented coherently, as it can be analyzed in the following sections.

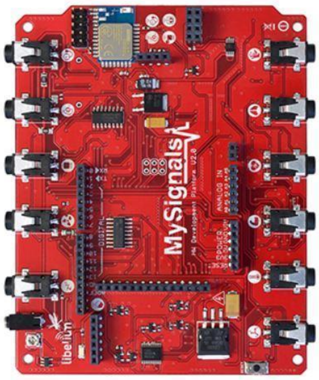

Figure 7 - MySignals hardware. Source: http://www.my-signals.com

2.2.2. eHealth Sensors and Devices

The sensors and peripherals recommended by the MySignals platform are illustrated in Figures 8.1 - 8.8. The platform can integrate any other biometric sensors connected wirelessly via Bluetooth - Bluetooth Low Energy (BLE) or wired. It should be mentioned that the development board supports analog sensors with a maximum of 3 channels (i.e. digital body position detection sensor that acquires information through 4 connections corresponding to the 3 axes of the accelerometer component and a Ground reference connection).

Fig 8.1. BLE Digital puls – oximetre

Fig. 8.2 Analog body thermometer

Fig. 8.3 Digital body position detection sensor

Fig. 8.4 BLE emergency button

Fig. 8.5. BLE Digital blood pressure sensor

Fig. 8.6. Digital Spirometer

Fig. 8.7. BLE Digital glucometer

Fig. 8.8. BLE Digital Scale for body assessment

Figure 8 - Source: http://www.my-signals.com

2.2.3. Power Supply System (PSS)

The power Supply System (PSS) is made of an internal rechargeable battery - Power Bank Voltaic V50 with an autonomy of 47 hours. Recharging the battery can be done during operation without affecting eHealth system functioning. The battery is supplied by a rectifier that receives 230 V AC or is supplied directly with 5V / 12V DC. Figure 9 shows the type of power supply used by the eHealth system to charge the battery. The entire eHealth system has a power consumption of 120mA – with the screen off and 200mA - with the screen on.

Hence, the Power Bank Voltaic V50 offers the possibility of charging during operation with a running time of about 47 hours - screen on (12800mAh * 3.7 / 5 = 9500mAh). In always-on mode and without being connected, the eHealth system has a running time of approximately 1350 hours (i.e., 56 days).

Figure 9 - Power bank Voltaic V50 (left) and rectifier (right)

2.2.4. Digital Processing Unit (DPU)

Digital Processing Unit (DPU) is a logical system that is based on the processing capacity of the Arduino development board used in the eHealth Unit configuration. Data collected from analog sensors and / or digital devices may be in a raw format, which is why filtering multiple signal samples is a way to avoid measurement errors.

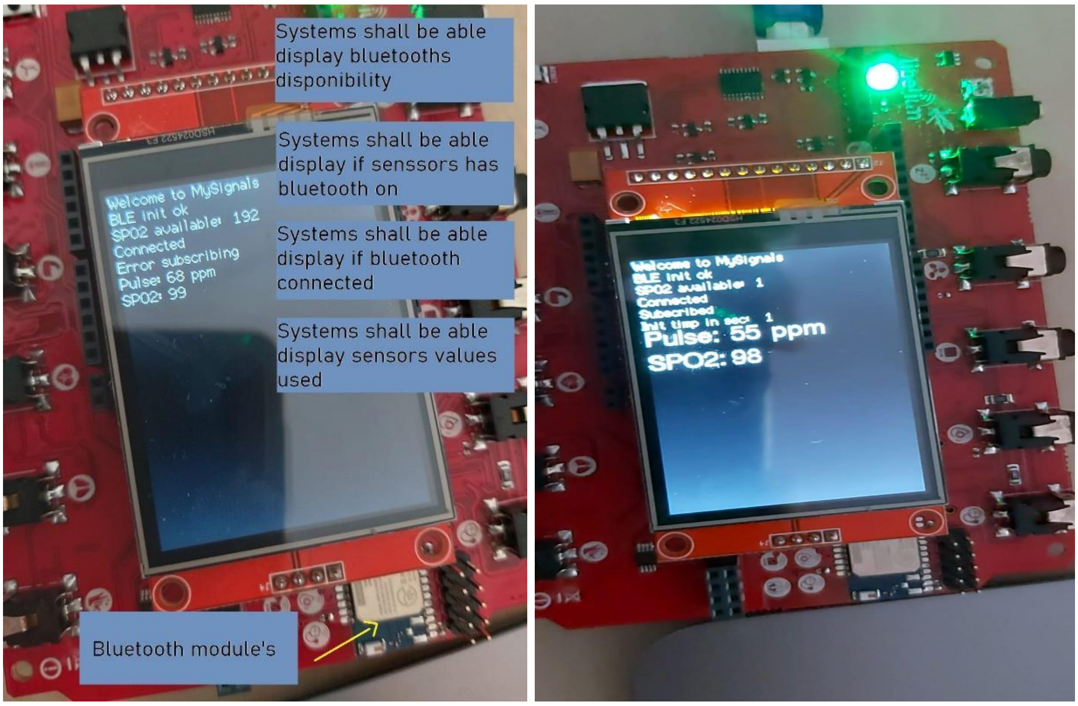

Figure 10 - Data display models for pulse oximetry sensor

2.2.5. Client - Server Communication Unit

Client-Server communication is performed through a built-in SoC Wi-Fi ESP 8266 network card that ensures internet connectivity and allows data transmission to the server through an encrypted channel via HTTPS protocol. The server receives the data, processes it, stores it and displays it in an Admin Center application.

2.2.6. Local User Interface (LUI)

The Local User Interface (LUI) system must send and / or display locally the biometric data collected from the sensors. Although the requirements of Romanian end-users specified that data should not be displayed on the user's display, a number of data display models were tested on the eHealth Unit, as can be seen in Figures 10, 11, 12 for the blood pressure, pulse oximetry and temperature sensors. Certain dynamic data display models were tested using intuitive real-time graphs. Following the Co-Design sessions, according to the users' requirements, the display of the biometric data on the screen was cut out.

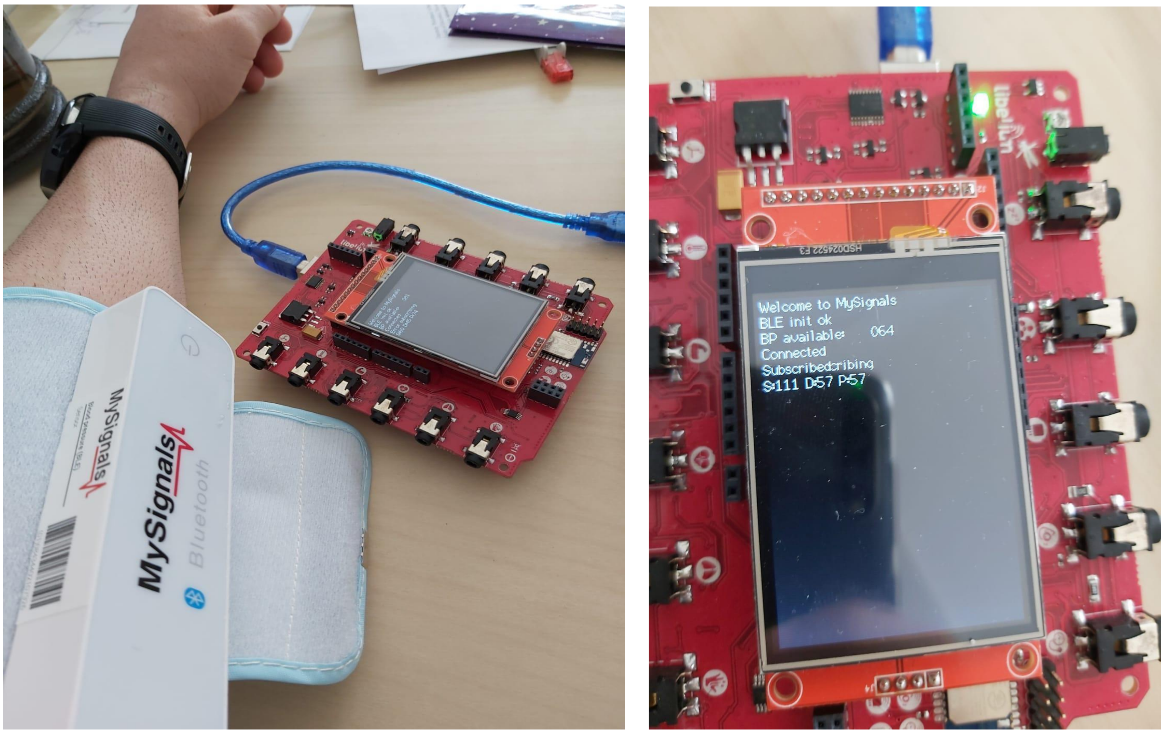

Figure 11 - Data display models for blood pressure sensor

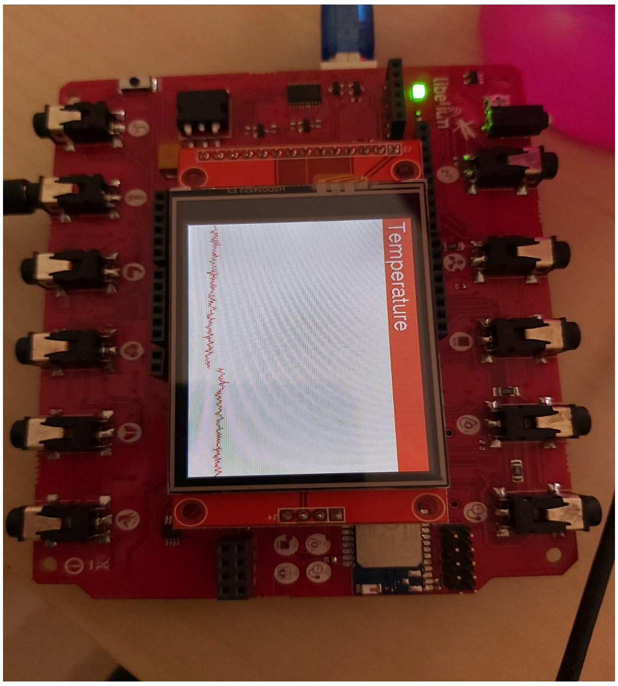

Figure 12 - Dynamic data display of temperature data

2.3. *Hardware implementation*

A Proof-of-Concept (PoC) system was developed using the open-source hardware development board Arduino UNO (ATmega328P) with the Libelium MySignals shield as shown in Figure 13. The PoC system was initially implemented on two sensors. The program memory was limited so in order to implement the other sensors it has been used Arduino Mega (Atmega 2560 - Figure 14). Specifically, the software code for the sensors, together with the BLE code and TFT display required 32116 bytes of program memory out of a total of 32256 bytes and 1697 bytes of RAM memory of 2048 bytes. This means about 98%. To add new sensors, the Arduino UNO development board had to be replaced with another development board with additional memory. Also, the Arduino UNO R3 (ATmega328) together with the network component, ESP8266 (ESP-01), requires a program memory of only 32Kbytes and 2Kbytes SRAM which was enough for the operation of three sensors.

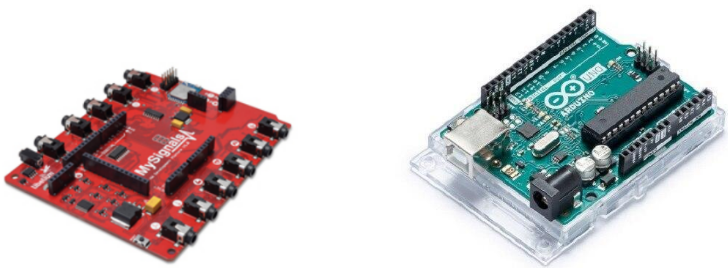

Figure 13 - Libelium MySignals and Arduino UNO
Source: https://www.arduino.cc/Products;https://www.my-signals.com

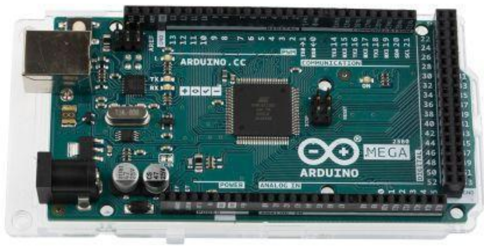

Figure 14 - Arduino Mega (microcontroller Atmega 2560)
Source: https://www.arduino.cc/Products

The PROTOTYPE solution was to use Arduino Mega (ATmega2560) with 256Kbytes and 8Kbytes SRAM. The program memory thus had 32116 Kbytes for all the desired sensors, including software code for BLE communication, display, data storage, network configuration and internet connectivity.

Hardware adaptations have been applied to the MySignals acquisition board to physically connect the Arduino Mega pins. As a result, in order to respect the configuration of the Arduino Mega pins, the height between the boards was increased by the use of special extensions as it can be seen in Figure 15. The use of Ardunio Mega allowed software implementation of all sensors and the PoC system testing was performed successfully.

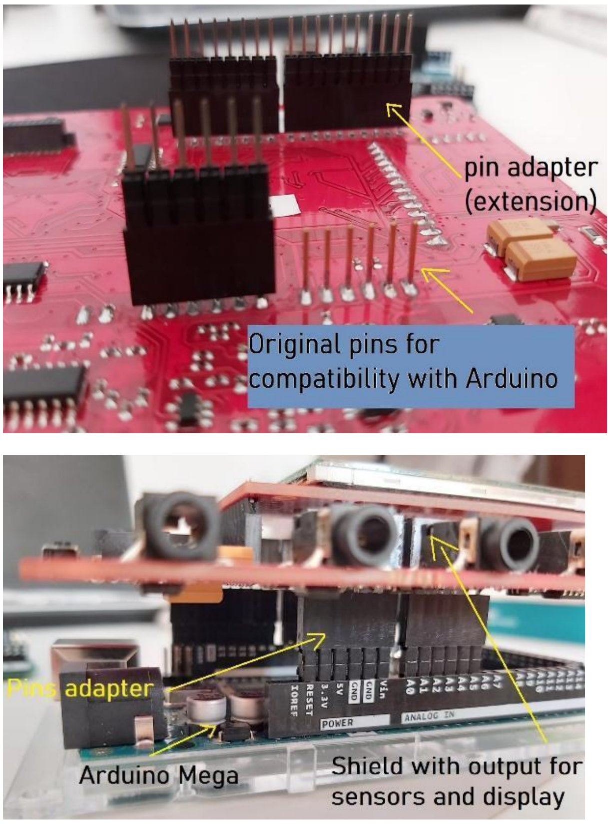

Figure 15 - Hardware adaptation by extending the required Arduino Mega pins

Important code sections have been redesigned and rewritten, reorganized modularly with the effect of considerably increasing the operating efficiency of the eHealth system. The code used for the eHealth Unit and peripherals (biometric sensors, biometric devices, displays, etc.) has been improved so that the entire embedded system exceeds the initial performance previously described by MySignals in terms of significantly lower program memory and operation of the acquisition system at a speed higher than that reported in the literature.

The eHealth system uses the configuration composed by Libelium MySignals shield, Arduino Mega and the ESP8266 network card for reasons related to program memory that reaches about 30-40% of maximum capacity and perhaps may facilitate software and hardware development perspectives.

Figure 16 describes the computing architecture of the e-Health system based on the ATMEGA AVR microcontroller and the communication channels through which it communicates with peripheral devices: sensors and/or biometric devices, Thin Film Transistor (TFT) screen, etc.

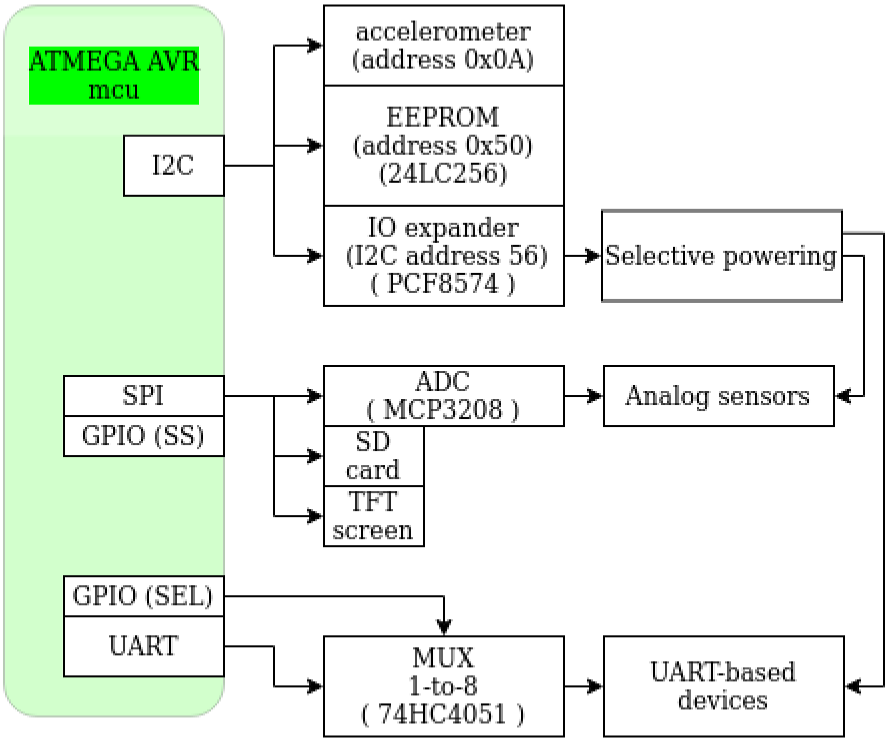

Figure 16 - Computing architecture of the e-Health system

2.4. *eHealth System Software specification*

2.4.1. Basic Embedded Software

The eHealth system is using MySignals configuration, Arduino Mega and an ESP8266 network card for reasons related to program memory that reaches about 30-40% of maximum capacity. Arduino UNO Wi-Fi Rev2 has come very fast very close to the upper limit of the program memory (approximately 98%), so for now it is desirable to use Arduino Mega.

2.4.2. Additional Embedded Software Modules

eHealth Unit system benefits from a series of global software improvements that can be presented modularly in terms of results:

- Better Bluetooth manager (by recognizing the MAC addresses at boot time and specific search of the devices by the MAC address);
- Better Wi-Fi driver (data communication between eHealth Unit and SAVE Cloud regulated to communicate minimum required data)
- Added brightness sensor (in order to control the brightness of the screen)
- Analog and/or digital signal filtering (some sensors require data filtering in order to eliminate certain specific artifacts of the acquired signal, e.g. temperature sensor induces a noisy signal - the solution found was to apply a median filter, etc.)

2.4.3. Software Relationship: eHealth System to Cloud

Currently, the data transmission to the SAVE Cloud has been successfully completed and the data transmission format has been established as a JSON template based on JavaScript principles. Prior to this step, the data transmission between PC and HTTP POST was simulated in order to simulate the data transmission in the SAVE Cloud. Finally, tests and demonstrations were performed to observe the operation of the system in real cases.

2.5. Final hardware trade-off analysis

As an additional remark, regarding the secure connection of eHealth Unit and SAVE Cloud, Arduino UNO Wi-Fi Rev2 was the optimal option, despite the limited program memory. The compatibility of the Arduino UNO and Arduino UNO Wi-Fi Rev2 development boards (Figure 17) was achieved and any inconsistencies of the pins or at the architectural level were identified due to the different microcontrollers of the ATmega328P and ATmega4809 development boards, according to Table 1. Changes were made to the software code in order to be able to use the I2C data bus for the ATmega 4809 controller.

Figure 17 - Arduino UNO Wi-Fi Rev2 (microcontroller Atmega 4809)
Source: https://www.arduino.cc/Products

After several tests, the decision was made to replace the Arduino Mega development board with Arduino UNO Wi-Fi Rev 2 (ATmega4809) for the following reasons: program memory is sufficient (it has 48K bytes and 6K bytes SRAM), integrated network board and options for easier implementation of the SSL / TLS encryption component. Thus, it would no longer be necessary to use the external SOC Wi-Fi ESP card. In order to work properly, the internal driver for the SPI bus and the timer driver have been modified, but as specified before, Arduino UNO Wi-Fi Rev2 is limited in terms of the program memory, which makes the development board Arduino UNO Wi-Fi Rev2 can no longer be used.

Table 1 - Arduino UNO Rev3 and UNO Wi-Fi Rev2

Board	UNO Rev3	UNO Wi-Fi Rev2
Microcontroller	**ATmega328P**	**ATmega4809**
Operating Voltage	5V	5V
Input Voltage (recommended)	7-12V	7 - 12V
Input Voltage (limit)	6-20V	6-20V
Digital I/O Pins	14 (6 PWM output)	14 (5 PWM Output)
PWM Digital I/O Pins	6	5
Analog Input Pins	6	6
DC Current per I/O Pin	20 mA	20 mA
DC Current for 3.3V Pin	50 mA	50 mA
Flash Memory	32 KB	48 KB
SRAM	2 KB	6,144 Bytes
EEPROM	1 KB	256 Bytes
Clock Speed	16 MHz	16 MHz
LED_BUILTIN	13	25
Length	68.6 mm	68.6 mm
Width	53.4 mm	53.4 mm
Weight	25 g	25 g

After several tests, the decision was made to replace the Arduino Mega development board with Arduino UNO Wi-Fi Rev 2 (ATmega4809) for the following reasons: program memory is sufficient (it has 48K bytes and 6K bytes SRAM), integrated network board and options for easier implementation of the SSL / TLS encryption component. Thus, it would no longer be necessary to use the external SOC Wi-Fi ESP card. In order to work properly, the internal driver for the SPI bus and the timer driver have been modified, but as specified before, Arduino UNO Wi-Fi Rev2 is limited in terms of the program memory, which makes the development board Arduino UNO Wi-Fi Rev2 can no longer be used.

3. Well-being System

Well-being system aims to adapt physical exercise and social activities for elderly people, driven by voluntary organizations, via stress/cognitive assessment services and actigraphy based services.

3.1. Cognitive Assessment System

The connection between well-being states and mild cognitive impairments for the elderly has been highlighted in the literature. As also aging influences the decline of cognitive components, such as alertness, attention, reduced stimulus perception, and decision making, several experiments through Choice Reaction Time (CRT) and Simple Reaction Time (SRT) have been considered, especially in recent decades for evaluating reaction time to visual stimuli.

Assuming that the reaction time to visual stimuli increases with age, especially in terms of chromatic light discrimination, the Choice Reaction Time (CRT) methodology of this initiative is based on a visual CRT in a specific paradigm based on several visual stimuli and response buttons.

The methodology requires a sensor for cognitive assessment based on a trichromatic light emitting diode (LED) stimulus (red, green, and blue) with the following procedures: the colour blue is the target and when the blue stimulus lights up, the elder must discriminate, select and execute the right button; the red or green colours are distractors and when the red or green stimulus lights up, the elder must discriminate, select and execute the left button.

Regarding the CRT configuration, the elder sits on the chair and manipulates the CRT device with both hands, holding the device with his index fingers and pressing the two buttons with his thumbs. Stimulus duration and stimulus onset synchronization are set to 200 ms and 2500 ms, respectively. If the old man presses the correct button for the respective LED (Right for Blue and Left for Red and Green) the timer stops and the stimulus-to-response time interval is recorded; otherwise, the timer continues to run until the stop state appears (the correct button is pressed or 2500 ms have passed since the last stimulus). The whole experiment lasts 5 minutes. The CRT sensor consists of two touch switches and a 5 mm RGB LED. The LED segments are CA (common anode), and the current through them is limited by resistors of 1KOhm (Figure 13). The touch switches ground the microcontroller inputs whenever they are pressed. The microcontroller's switch inputs have the weak pull-up enabled.

4. Full-programmable wearable devices

The TTGO T-Watch-2020 produced by LilyGO (model: LILYGO-H414 – Figure 18) is a 1.54 inch touch display programmable wearable device with environmental interaction, based on the ESP32 SoC produced by Espressif.

Specifications**:** main chip ESP32, dual core MCU (integrated dual mode Bluetooth/wifi), FLASH QSPI flash 16MB, 520 KB SRAM / PSRAM 8MB, AXP202 PMU power management, 1.54 inch LCD capacitive touch screen, BMA423 three-axis accelerometer, MAX98357A I2S PCM input Class D audio amplifier, support for RTC, IR, Vibration Motor function

Development platform**:** ESP-IDF (native SDK), Arduino, Lua, MicroPython, Scratch, PlatformIO in Visual Studio Code

The freely available software development tools are the main future-proof advantage, providing full control over the data sent by the sensors, data analytics using neural networks programmed by Tensor Flow Lite.

Figure 18 - LilyGo programable smartwatch

Functional description:

The data from the BMA423 accelerometer is used to analyse the situation of the person wearing the watch. If a programmable threshold is exceeded the data is sent to the local neural network for real-time analysis and to the cloud for more extensive post-analysis. The data is buffered if the available storage until a fluctuant internet connection is available again. To conserve power the watch wakes up only on interrupts generated by the accelerometer and the telegrams can be uploaded into the Cloud directly via the WiFi router of the smart-house (not only via a BT-connected smartphone).

Figure 19 - TTGO boot screen

5. Full-scalable smart sensor kit

The Mi Smart sensor kit is produced by Xiaomi and was intended to be integrated in the security of the house. The main idea is based on security of the home and neighbourhood and we started with this versatile set as is mentioned on the product page. Is designed to be like a central hub connected to the cloud and also connected to many sensors, like doors and windows sensors, motion sensors, smart buttons, flooding sensors, temperature sensors and so on.

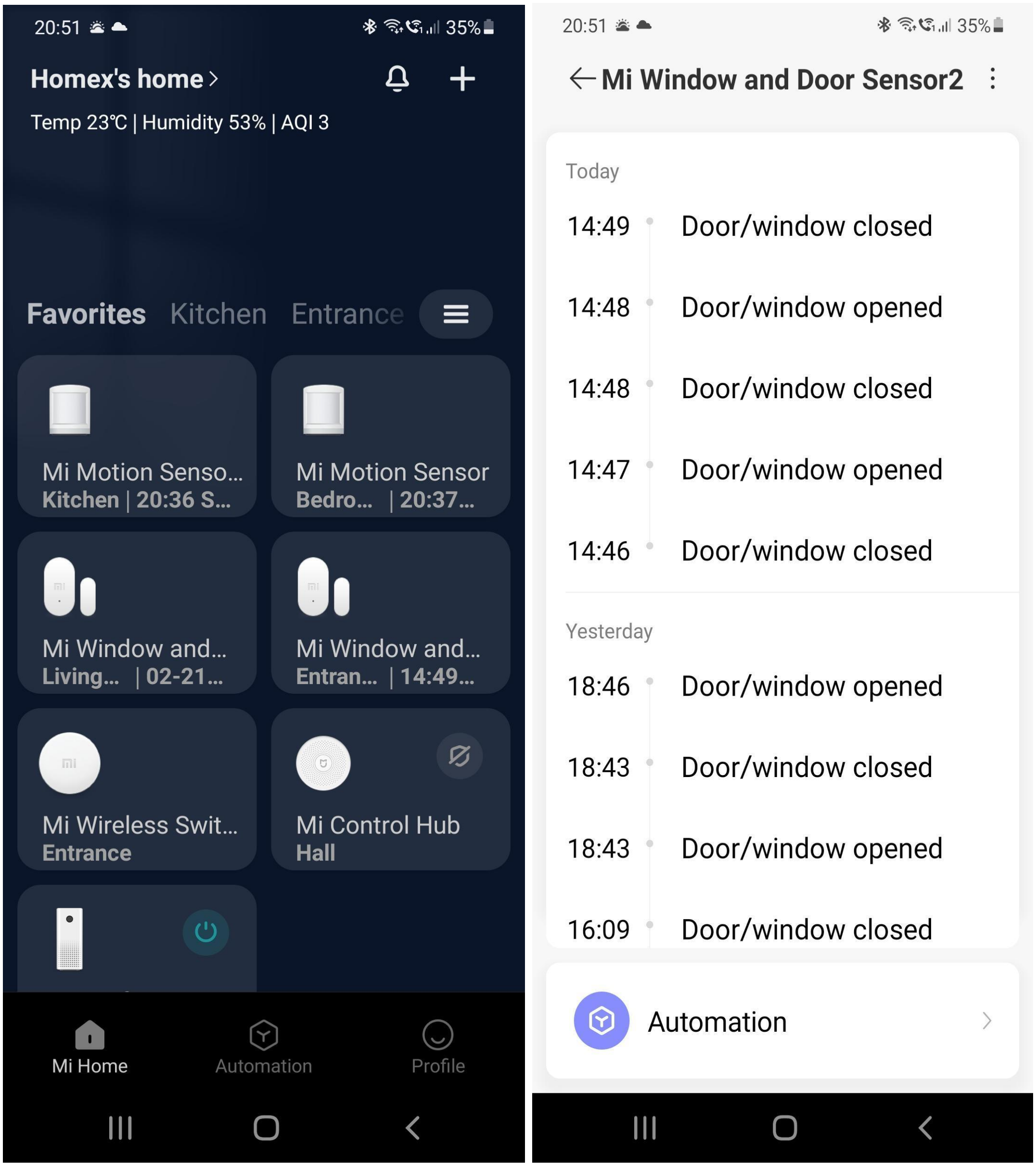

Figure 20 - Mi Home app

All these sensors have local intelligence and connected to Mi Control Hub provide beside the security level all the information about moving in the house and also on the doors and windows. Could act like a pro alarm system and also extended to various security issues like fire, flooding or entrance breach.

We tested a minimal set of these sensors as is shown in the sensors' movie-clip. Unfortunately, the producer did not mention that the Mi Control Hub could only connect to his own cloud and not on Google or Amazon Cloud.

The system can work reliably on its own application on smartphones and could provide real information that can be proved that is useful, but in order to integrate we need to choose another set, like Agora.

The Figure 20 shows the mobile application and the sensor logs can be seen.

6. Samsung Galaxy Smartwatch

6.1. Tizen operating system

Tizen is a Linux-based mobile operating system, supported by the Linux Foundation, but developed and used mainly by Samsung Electronics. The project was originally designed as an HTML5-based platform for mobile devices.

Samsung merged its previous effort based on the Linux operating system, Bada, into Tizen, and has since used it primarily for portable devices and smart TVs.

Much of Tizen is open-source software, although the software development kit contains components owned by Samsung, and portions of the operating system are licensed under the Flora License, a derivative of the Apache 2.0 License that grants a patent license only to "certified" platforms. Tizen".

The Tizen project was formed by the Linux Foundation in 2011 as the successor to MeeGo, another Linux-based mobile operating system, with the help of its main Intel supporter joined by Samsung Electronics, Access Co., NEC Casio, NTT DoCoMo, Panasonic Mobile, SK Telecom, Telefónica and Vodafone as trading partners.

Tizen is designed to use HTML5 applications and target mobile and embedded platforms such as netbooks, smartphones, tablets, smart TVs and in-car entertainment systems.

Sprint Corporation (which supported MeeGo) joined the Tizen Association in May 2012. On September 16, 2012, Automotive Grade Linux announced its intention to use Tizen as the basis for its reference distribution.

In January 2013, Samsung announced its intention to launch several Tizen-based phones that year. In February 2013, Samsung merged its Bada operating system with Tizen.

In October 2013, the first Tizen tablet was delivered by Systena. The tablet was part of a development kit exclusively for Japan.

In 2014, Samsung launched the Gear 2 smartwatch that used a Tizen-based operating system.

On February 21, 2016, Samsung announced Samsung Connect Auto, a connected car solution that offers diagnostic services, Wi-Fi and other car-connected services. The device connects directly to the OBD-II port under the steering wheel.

On November 16, 2016, Samsung said it would work with Microsoft to bring .NET Core support to Tizen.

According to the research of the Analytical Strategy, approximately 21% of smart TVs sold in 2018 run on the Tizen platform, thus becoming the most popular smart TV platform.

Releases:

30 april 2012: realease Tizen 1.0

18 february 2013: realease Tizen 2.0

20 may 2017: realease Tizen 3.0

2018: realease Tizen 4.0

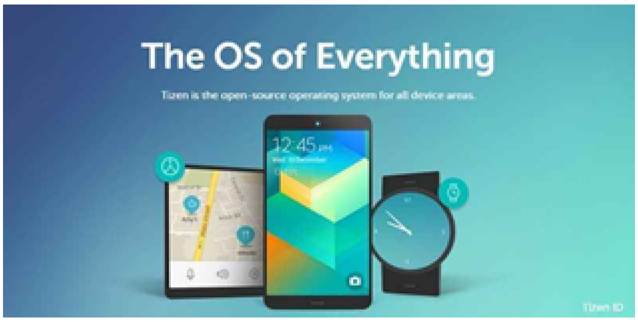

Figure 21 - Tizen advert

6.2. *Compatible devices*

Smart watches:

- Samsung Galaxy Gear
- Samsung Gear S
- Samsung Gear S2
- Samsung Gear S3
- Samsung Gear 2
- Samsung Gear Fit 2
- Samsung Gear Fit 2 Pro
- Samsung Gear Sport
- Samsung Galaxy Watch
- Samsung Galaxy Watch Active
- Samsung Galaxy Watch Active 2
- Samsung Galaxy Watch 3

Tablets:

- Samsung Galaxy S6 lite Tablet

Cameras:

- Samsung NX200
- Samsung NX300
- Samsung NX1

Smartphones:

- Samsung Z
- Samsung Z1
- Samsung Z2
- Samsung Z3
- Samsung Z4

Appliances:

- Family Hub 3.0 Refrigerator

LED Wall controllers:

- SBB-SNOWJ3U

6.3. Samsung Galaxy Watch 3

Galaxy Watch3 combines smartphone productivity and sophisticated health technology in a classic, premium device (Figure 22).

Figure 22 - Samsung Galaxy Watch 3

Specifications:

	Galaxy Watch3 45mm	Galaxy Watch3 41mm
Form Factor		
Design	Circular	
Display		
Screen Size	1.4" inch (45 mm)	1.2" inch (41 mm)
Display Type	Super AMOLED	
Screen Resolution	360 x 360 pixel resolution	
Touch Type	Multi-touch	
Body		
Size	46.2 x 45 x 11.1 mm	42.5 x 41 x 11.3 mm
Weight	53.8 g	48.2 g
Casing	Stainless steel, Titanium	
Platform		
OS Platform	Tizen for Wearables	
Processor		
CPU	Exynos 9110 Dual Core 1.15 Ghz Cortex A53 processor	
Memory		
Internal Storage	8 GB	
RAM	1 GB	
Card Slot	none	
Connectivity		
SIM Slot	4G LTE (eSIM)	
WIFI	Yes	
NFC	Yes	
Bluetooth	Bluetooth 5.0	
USB	none	
Battery		
Battery Capacity	247 mAh battery for 41 mm version and Li-Ion 340 mAh battery	
Charging Technology	Wireless Charging	
Body Resistance		
Body Protection	IP68 Waterproof up to 50 meters	
Glass Technology	Corning Gorilla Glass DX	
Sensors		
Sensors/Functions	Accelerometer, Gyroscope, Barometer, Optical heart rate sensor, ECG, blood pressure sensor, ambient light sensor.	
Media		

Audio	Voice Assistant
Radio	-
Speaker	Yes
Microphone	Yes
Camera	
Camera Type	none
Features	
Features	independent call and messages for 4G LTE, call and message notifications, weather, alarm, social app support, downloadable fitness apps, sleep monitoring, find my phone and more
Color Availability	
Colors	mystic black, mystic silver, and mystic bronze
Package Contents	
Package Contents	1 x Samsung Galaxy Watch 3 1 x User Manual 1 x Charging Cable 1 x Warranty Card
Compatibility	
Compatibility	Android and iOS

6.4. Smartwatch as a sensor

For the smartwatch, a specific SAVE interface was developed in the form of a watch face (Figure 23, which fulfils a double role: wearable user interface for the elderly, regardless of his/her location (inside or outside the home), and wearable sensor (for basic biological signals - pulse, activity monitor - and location).

Figure 23 - Watch face for smartwatch

The watch face has been designed to display in an aesthetic and efficient way the necessary information and to collect data provided by the sensors, the location and to display the notifications defined by the user (end-user and/or caregiver) in the web application interface (developed for this purpose), all with a minimum energy consumption.

REFERENCES

[1] "LiliyGO", [Online]. Available: http://www.lilygo.cn/prod_view.aspx?TypeId=50053&Id=1290&FId=t3:50053:3

[2] "Tenserflow," [Online]. Available: https://www.tensorflow.org/lite/microcontrollers