VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Human Behavior Collecting in a Smart Home (SH) within IoT

Monitorování lidského chování v Smart Home (SH) v rámci IoT

Anatoly Ens

Diploma Thesis

Supervisor: doc. Ing. Jan Vaňuš, Ph.D

Ostrava, 2021

Abstract and keywords

Abstraktní

Je široce prozkoumáváno, že za posledních 20 let se nehody s poklesem dramaticky zvýšily. Nejčastěji se to stává u starších lidí, kteří odcházejí sami. Bez včasné pomoci může nehoda s pádem vést k nepředvídatelným následkům.

Technologie inteligentní domácnosti dnes umožňují nejen sledovat, ale také shromažďovat a zpracovávat údaje o lidském chování. Tato diplomová práce využívá nejmodernější technologie ke sběru dat z inteligentní domácnosti založené na technologii KNX pomocí softwaru Home Assistant, jejich předzpracování pomocí modelu strojového učení uloženého ve službě Amazon Web Service a nouzové upozornění na událost pádu prostřednictvím telegramového posla.

Klíčová slova

Inteligentní dům, ADL, rozpoznávání pádů, strojové učení, KNX, Home Assistant, sběr dat.

Abstract

It is widely researched that last 20 years fall accidents increased dramatically. Most often it happens with elderly people who leave alone. Without timely assistance, fall accident may lead to unpredictable consequences.

Nowadays smart home technologies allow not only monitor but collect and process human behavior data. This diploma thesis applies cutting edge technologies to collect data from smart home based on KNX technology with help of Home Assistant software, preprocess it with a machine learning model stored in Amazon Web Service and notify emergency about fall event via Telegram messenger.

Keywords

Smart home, ADL, fall recognition, machine learning, KNX, Home Assistant, data collection.

Table of Content

List of symbols and abbreviations

SH	Smart Home
IoT	Internet of Things
HBC	Human Behavior Collection
ADL	Activity Daily Recognition
ML	Machine Learning
BACS	Building automation and control system
SHC	Smart Home Care (laboratory)
MQTT	Message Query Telemetry Transport
HA	Home Assistant
ETS	Engineering Tool Software
HTTP	Hyper Text Transfer Protocol
AWS	Amazon Web Service
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit

List of images

List of tables

Introduction

One of the fast-growing technologies nowadays is the **Internet of Things** (IoT). The IoT is a concept in which objects such as people, devices, cars, buildings are connected to the internet platform (1). Any connected objects can be interpreted as a source of data. This concept is widely applied in Smart Homes.

There are many alternative terms for “**Smart Home**” (SH) in the professional literature: Automated homes, Intelligent buildings, Adaptive home, Smart House, eHome etc. Smart homes incorporate intelligent devices that control the features of the home. In general, Smart Home represents a home with one or more devices connected in such a way that homeowner can control, customize, and monitor one’s living environment (2).

With technology development, the focus of Smart Homes has shifted from automatization and human’s life simplicity (e.g. light control, energy management, door control etc.) to recognition of human behavior patterns (3). **Human behavior collection** (HBC)can be used in various applications and services.

There are only some possible approaches based on human behavior data. For example, using behavior data can be predicted the occupation of the particular room for energy management (light control, heat control) (4). As one of the methods of indirect monitoring CO2 sensor can be used (5). Such an approach keeps private security comparing with direct methods (cameras) as IoT devices can be hacked (6).

To create a more useful version of the smart home, the home must recognize the user's actions and understand their behavioral patterns or send warnings or suggestions. For example, prediction of user preferences and actions can be used to adjust the temperature (7) or lighting conditions (8) according to their actions. It can also be used to save energy, for example, to turn off unused lights when the user is leaving or sleeping, or to turn on water heating only when the user is expected to take a shower (9).

Using behavior data, it is possible continuously predict the activity which person is doing now (10). The term for that is recognition of **Activities of Daily Recognition** (ADL). The outcome of ADL recognition can be used for assistance or further predictions health monitoring (11).

Accordingly, there is a tendency for homes to become not only smart but also caring (12). A home care environment helps to monitor people’s health and provide necessary assistance without breaking privacy. Data from HBC can be used to improve life quality and prolong a person well health being.

However, HBC is a very wide topic with many applications. The focus of this work is to make fall recognition application for elderly and people with specific diseases which let a person to live at one’s home as long as possible. As a source of data, the wearable sensor is used. The solution is integrated into Smart Home environment and that is the main difference from other products on the market.

What makes home caring is called predictive models. Models are based on mathematics and statistics (13). In Data Science models are built using **Machine Learning** (ML) techniques.

Description

The scope of this diploma thesis is to design Human Behavior Collecting system which would store data from building control system and recognize fall accident in Smart Home. Given chapter explains content of the following chapters and makes short review on the methods applied for solving the assignment.

Infrastructure of research Smart Home Care (SHC) laboratory of VSB was used for practical implementation and testing of the designed system. The laboratory has several floors, but for purpose of this work, only one floor is used. It is equipped with sensors and actuators wired by the KNX bus. The research for smart homes and SH technologies is described in chapters **Building automation** and **Building automation technologies**. Chapter **Design building control** describes the practical implementation of KNX technology in the SHC laboratory.

IoT platform is deployed at single-board computer Raspberry Pi. Instead of remote cloud services, open-source software Home Assistant (HA) was selected. HA is a very flexible and adjustable software used for home automation (14). Chapter **Human behavior collection** covers the connection of the KNX system to the internet of things and the realization of smart home monitoring and data collection.

Data for health monitoring is collected from the wearable device with an accelerometer and gyroscope. As a prototype for an experiment M5Stick device based on ESP32 microcontroller was used. That is a portable device that is perfect for prototyping. Chapter **Pattern recognition in human behavior** contains a description of fall recognition application, its realization, and experimental results.

Accelerometer and gyroscope data is sent to the main control system using MQTT protocol via the wireless connection. Smart home KNX infrastructure is connected to the control system via Ethernet. The sketch of the architecture of the project is presented in figure 1.

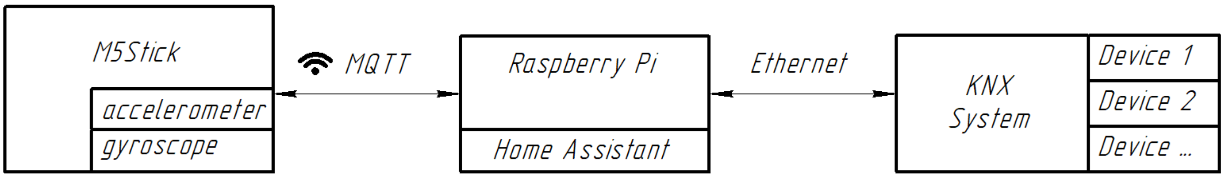

Figure 1 – Sketch of project architecture

1. Building Automation

Building automation and control systems, smart home systems and home automation are terms that are often used synonymously to describe an intelligent building environment. To prevent confusion, the chapter will introduce and use such terms as defined in the international standard DIN EN ISO 16484-2:2004 (15)

Building automation and control system (BACS) – system, comprising all products and engineering services for automatic controls (including interlocks), monitoring, optimization, for operation, human intervention, and management to achieve energy-efficient, economical, and safe operation of building services.

BACS is a common term and can be divided into **building automation** and **building control systems**. While building automation covers the whole measurement, control and management of building services on a higher level, building control is a specialized sub-group of BACS which does not require a central control unit (16).

BACS or building control systems are dedicated to individual room control which means the control of a particular area of a building (e.g. a specific zone or an individual room). A zone is a defined area in a building with some form of control.

Room control (or integrated room automation) – plant/application-specific devices and functions for single zone or individual room control. That includes a wide range of intelligence such as: integrated monitoring, interlocks, open and closed-loop control, and optimization of combined building services such as HVAC&R, lighting, window blinds/shades control, electrical power distribution, and other trades, by communication functions.

The terms like smart home system, home automation, and others are not defined in the standard DIN EN ISO 16484-2:2004. However, they are mostly used as a synonym to one of the defined terms as BACS, building automation, building control, or room control.

Intelligent house – is a residential or office space, the main functions and characteristics of which are provided using intelligent systems. Security including video surveillance, energy management, climate management, etc. are provided with the help of specialized systems – automated control systems.

One of the most inclusive definitions of intelligent buildings was given by the Intelligent Buildings Institute (IBI) in Washington DC: “An intelligent building provides a productive and cost-effective environment through optimization of its four basic elements and the interrelationships between them:

- Building structure
- Building systems
- Services
- Management

Intelligent buildings help building owners, property managers and occupants realize their goals in cost, energy management, comfort, convenience, safety, long term flexibility and marketability” (17).

The degree of intelligence of the building can be represented as 5 level structure, where higher-level covers all below. Most technologies cover only levels 1-3, while levels 4 and 5 are the scope of research.

Level 1 – the house contains separated devices smart devices and systems, which do not communicate with each other.

Level 2 – the house contains devices that can communicate with each other and exchange information about events to improve functionality.

Level 3 – the house and systems inside are connected to the global network, which gives the ability to monitor and act outside of the building.

Level 4 – the system collects data from house owners and adjust its parameters according to historical events.

Level 5 – the system constantly and fully monitors house owners and able to simultaneously react to a person's behavior.

A building can be described as intelligent if it is energy efficient, safe, comfortable, and flexible. Such characteristics can be achieved by **operation and technical functions**. These functions cover both technical operation and comfort features of the building (18).

2. Building automation technologies

Any automation systems differ in the used communication technologies, types of controllers, sensors, visualization, and dispatching systems. Depending on the scope of application, the requirements for the applied technologies differ. Typically, these requirements relate to the materials and equipment used. For example, in the oil and gas industry, explosion protection is important, in the field of commercial metering – metrological accuracy, in fire-fighting systems – resistance to combustion, etc.

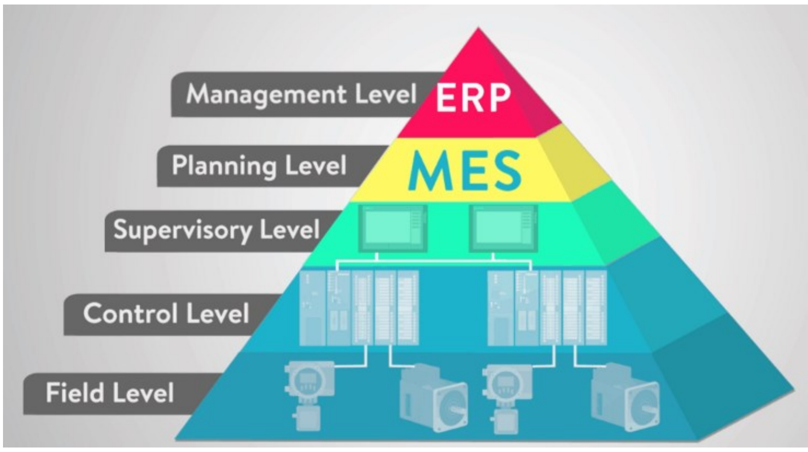

Figure 2 – Automation pyramid (19)

However, the requirements for communication protocols and interfaces are equally important. With the help of communications, information is transferred within a level or between levels of automation. Some protocols are more suitable for the field level, others for the automation level, and some are so versatile that they are used at different levels. Initially, building automation systems used the same communication technologies as in industrial automation, but later specialized protocols were developed, which are actively used today.

The following protocols are most widely used as field ones:

- Modbus is a hardware-oriented protocol for the communication of various sensors and actuators with controllers
- DALI – lighting control interface
- M-bus – a collection of data from various meters, such as electricity, water, heat, etc
- EnOcean – wireless interface, etc.

The next three universal technologies, KNX (EIB), LON and BACnet, are rightfully considered the mainstream in building automation. Each of these technologies has its adherents and opponents, all have their strengths and weaknesses, as well as the prevailing scope.

2.1. LonWorks

LonWorks technology (often referred to as LON – Local Operational Network) is equally good at the field and automation levels as well as inter-level communication.

The foundations of LonWorks technology were laid in the early 90s years of the last century, when Echelon engineers developed the specialized Neuron Chip microprocessor (first introduced in December 1990), the LonTalk communication protocol (ANSI / EIA 709-1), and the first instrumental software for development and design.

LonWorks technology is open, and products from different manufacturers can be easily combined into a common network, provided that the developer adheres to the interoperability rules. The LONMARK International website lists the standard network variable types (SNVT Master List) and standard configuration property type (SCPT Master List). Besides, all devices must comply with the Standard Functional Profile Types (SFPT). A functional profile is a collection of input and output network variables and configuration parameters that are typical of an application.

LonWorks is currently used primarily for the automation of building engineering systems. The following are the main systems using LON technology:

- security and fire alarm
- control of indoor and outdoor lighting
- control of supply and exhaust ventilation and air conditioning
- control of heating and cooling system
- power supply monitoring
- dispatching elevators
- water supply and sewerage, etc.

LonWorks network topology: bus, ring, star, free. Supported environments transmissions: twisted pair, optical cable, coaxial cable, radio channel, power grid, IP networks, IR channel. The most common transmission medium is twisted pair. The physical structure of LonWorks networks is defined by the channel (physical medium) and segment (a section of the physical medium or channel connected to a port router or repeater) of the network.

A commercial feature of LonWorks technology is the need to charge Echelon a $ 5 download and commissioning fee for each node (the so-called loans).

Lonworks has various advantages and disadvantages:

+ Many developers of Lonworks products in the market Open standard (a lot of producers)
+ It has a strong expansion capability
+ System maintenance convenience
– Hardware is specific and requires the Neuron chip for network protocol
– Extensions are allowed only through the LonMark Consortium

2.2. BacNet

BACnet stands for Building Automation Control network and represents a communication protocol for building automation developed by the association ASHRAE (ANSI / ASHRAE standard 135-2001), and recently received the status of a standard ISO 16484- The main purpose of the BACnet protocol is to standardize the interaction between building automation systems from various manufacturers, allowing information exchange and equipment collaboration.

In 1987 ASHRAE attempted to develop a protocol (set of rules) managing the interaction between the various devices used in the systems building automation. The BACnet standard is currently adopted by ANSI (American National Standards Institute) and ASHRAE and has received international recognition and has been adopted in several countries as a national standard. This laid the foundation for the future development of this industry.

BAC net devices are physically similar to other standard devices of building automation systems, but their physical form is not the main thing. BACnet is just a set of rules for communication between devices in systems building automation. The microprocessors of these devices are programmable, which means that they will be able to "understand" each other and meet the general requirements of the BACnet protocol. The physical nature of the device itself remains unchanged.

Each device on a BACnet network is described by a set of standard objects. The number of identical objects making up the device is not limited. The standard defines the types of objects. The standard also defines the classes of applications that devices perform:

- Alarms and events
- Access to files
- Access to objects
- Remote device control
- Virtual terminal

Classes of application tasks are described by a set of services (services) that are used for communication between devices.

All these properties of the standard are the same for all manufacturers of BACnet devices, which allows you to create networks built on common software with practically unlimited capacity. BACnet uses the following technologies as physical layers:

- Ethernet
- BACnet / IP
- RS-232
- MS / TP (Master-Slave / Token-Passing) via RS-485

2.3. KNX

Today KNX is one of the most common solutions for use in medium and large automation systems for homes, offices and commercial buildings. It appeared on the market more than twenty years ago and is supported by many major manufacturers of electrical equipment.

The KNX Association today includes more than 350 companies around the world. The number of certified products is over 7000. The most famous KNX equipment manufacturers today are ABB, Gira and Schneider Electric.

The main key features of KNX are guaranteed compatibility of products from different manufacturers and a single software tool (Engineering Tool Software, abbreviated - ETS) for planning, development and implementation of a project. From a technical point of view, the solution allows to realize all popular automation scenarios including lighting, climate control and security.

2.3.1. KNX Bus

The overwhelming majority of KNX projects are based on the use of a special dedicated wire bus (twisted pair). All controllers, sensors and actuators are connected to the wired bus. In practice, this means the need to develop a project and lay the necessary communications during construction or repair. Formally, there are other transmission media in the standard (in particular, the power supply network and radio communication), but they are relatively rare in projects. As an alternative option, which does not require the laying of an additional control bus, circuits are often used with the output of all individual consumers to a common panel. Both versions have their own pros and cons. At the same time, their combination is also allowed if compliance with the KNX specifications is maintained.

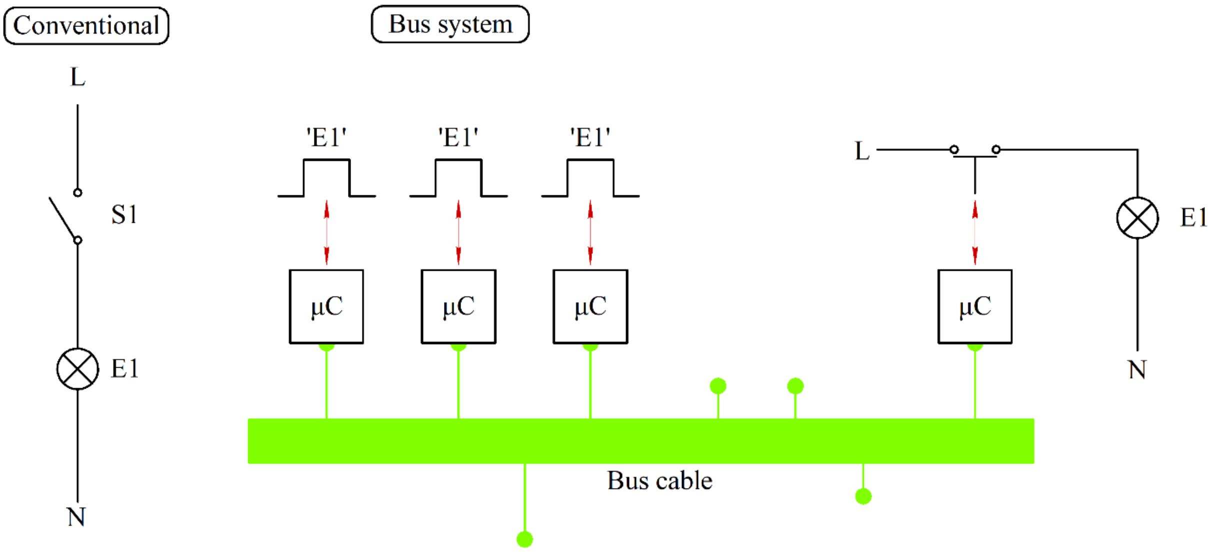

Figure 3 – Conventional connection vs Bus system (20)

The wired bus topology can be selected flexibly. The use of linear busbars, tree and stars is allowed. Termination is not required here, but it is recommended to pay attention to protection against overvoltage and thunderstorms. The basic element of the logical structure is a segment that contains up to 64 nodes. Up to four segments can be combined into lines, which can be combined into an area (up to 15 lines). At the highest level, up to 15 areas can be combined into a system. The total number of devices in one network is about 58 thousand.

Several features are important to know:

1. KNX TP – is a bus that carries out the power supply (24V) and information transfer functions at the same time. Comparing to normal hard-wired systems with operating elements and electrical consumers, all bus devices have microcontrollers and can be easily reprogrammed. It makes the KNX bus very flexible because the function of any device in the building can be changed at any time.
2. Operating elements are separated from electrical consumers. For instance, light control is done with actuators and there is no need to connect the light bulb directly to the switch

button. It brings more safety because actuators are a kind of relay which protect electrical consumers from voltage rises.

2.3.2. Addresses and commands

The basic configuration of KNX projects can be considered as decentralized – data exchange between devices is carried out directly, without the participation of any separate specialized controller. For example, in this way, it is possible to implement an autonomous segment for controlling lighting in a house based on programmed scenes and algorithms. However, one must be aware that the devices themselves are relatively simple. If more complex interaction algorithms are needed, the installation of an additional controller is required.

Device addressing typically uses an area-line-device scheme. The size of the address field is 16 bits. In this case, your addresses must be registered in each device at the stage of programming the system via ETS.

Group addresses are important logical elements of the system. They are functionally assembled devices. A sensor (for example, a button) can send commands to only one group, and actuators (for example, relays) can receive information in several groups at once.

Using such a scheme simplifies the management of a group of devices by sending a single message to a group address instead of unicast. The limitations on the maximum number of group addresses are usually individual and are specified in the hardware specifications.

The group addresses may have a different structure:

- 3-level structure (main group / middle group / subgroup)
- 2-level structure (main group / middle group)
- Free structure

The most common structure is 3-level. Possible structures are presented in figure 4.

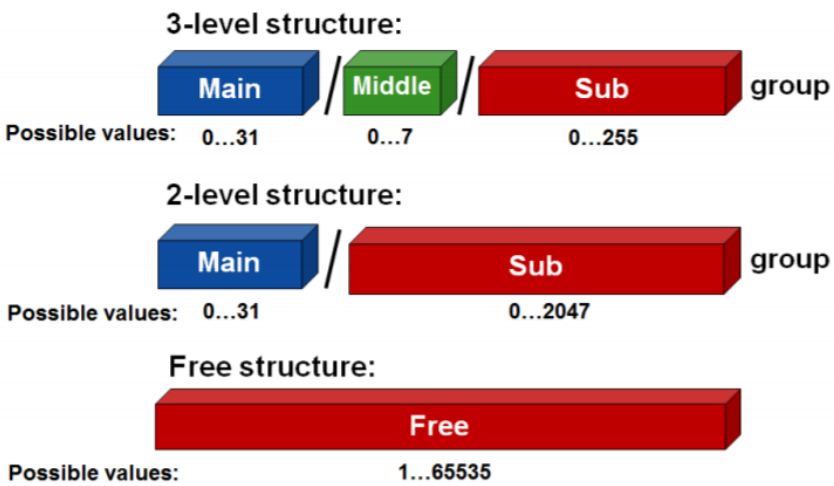

Figure 4– Structure of group addresses (21)

Communication between devices is performed via telegrams. When some event occurs (for example push button is pressed), a new telegram is transmitted over the bus. The structure is following:

- Control field
- Source address
- Target address
- Routing counter
- Length
- Useful data
- Check byte

In general, the signal is a voltage change in the time domain. Telegram is transferred with 8-bit characters (figure 5).

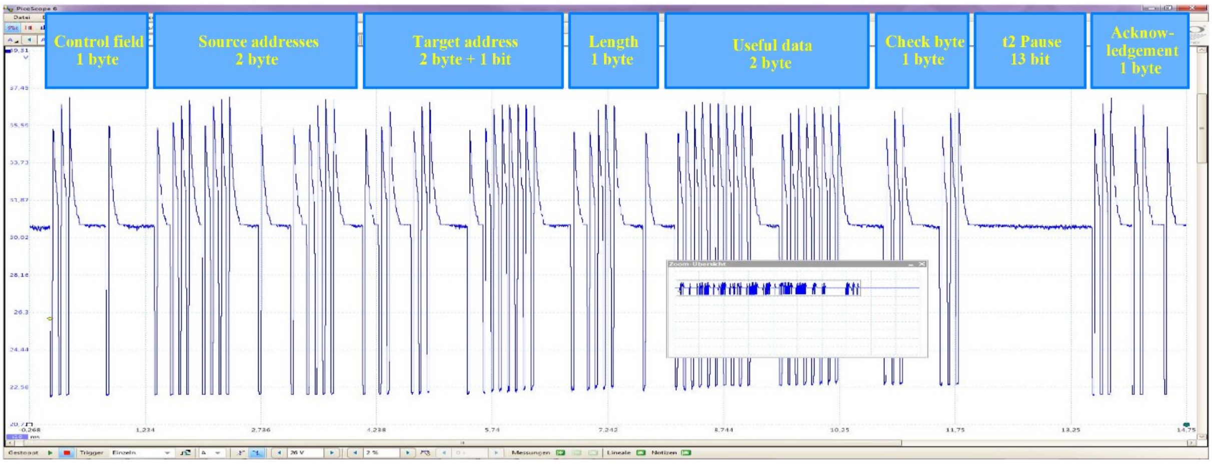

Figure 5 – Complete KNX telegram (22)

All devices connected to the bus can read the telegram. However, only some of them need it. The telegram contains the target address which is the group address. If a related group address changes its state, the device will act.

Pros and cons of KNX technology:

+ The flexibility of making changes
+ Optimal functionality/cable ratio
+ The separation between power (230V) and operation
+ Open standard (a lot of producers)
– Requires interface for management and diagnostic
– KNX has a small community because of its high price

2.3.3. ETS5 software

ETS (Engineering Tool Software) is a unique, versatile software package for the development and configuration of a smart KNX system for automated management and control of buildings and apartments.

Its features:

- ETS is a truly unique software package: anywhere in the world, you can use the same ETS software package for any project based on the KNX standard and for any device certified by the KNX Association.
- This software package is a valid part of the KNX standard.
- Any KNX databases of products from all manufacturers of the KNX Association can be imported into ETS!
- The KNX system, combined with the ETS software package, provides an easy-to-use and vendor-independent toolkit for control system planners and installers.

The working document in ETS is the project. In this case, you can use several projects in one network (for example, in a large building) or several networks with one project (for example, in the same type of hotel rooms). Direct work with the project does not require access to the installation, however, to download the configuration and diagnostics, of course, you will need to connect to the KNX network. This operation is carried out either through local (USB or RS-232) interfaces or an IP bridge.

In general, the process of programming an automation system consists of several steps:

1. Creating a project file
2. Importing information about used devices into the program directory
3. Building structure creation
4. Adding devices from the catalogue to the project
5. Selecting addresses, setting parameters, adding comments for devices
6. Creation of a structure of group addresses
7. Distribution of devices to group addresses
8. Uploading the project to the automation system
9. Performance check, diagnostics

To load a project into an installed automation system, you need to connect a computer to it using one of the supported interfaces. This interface is not required directly for operation, but often it remains in the installation for the convenience of making changes to the configuration.

When startup ETS software, the main window with dashboard appears (Figure 6).

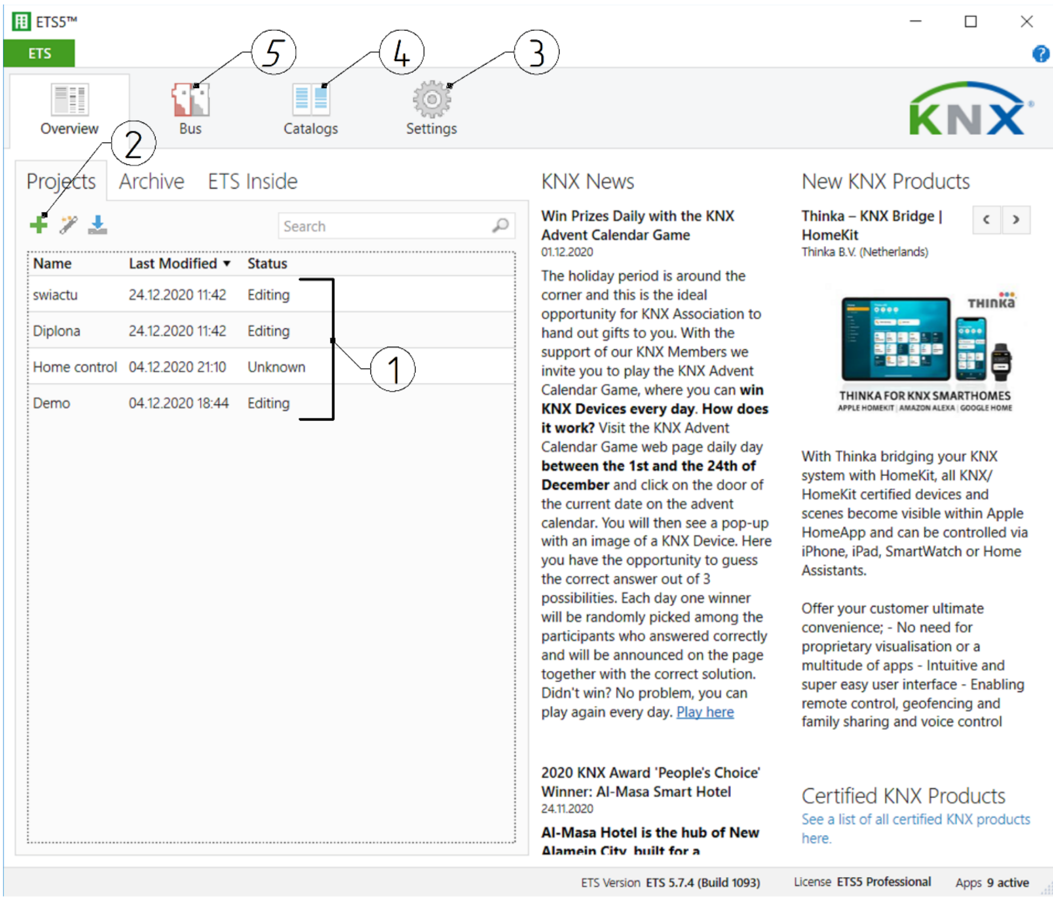

Figure 6 – Main window ETS5

It contains the following elements:

- In area ① all imported projects are stored
- With plus button ② a new project can be created
- Setting tab ③ shows some options like language settings, hotkeys, storage info
- Through Catalogs tab ④ new devices can be uploaded
- In the Bus tab, ⑤ bus monitoring is done. For instance, it can be read all telegrams or check if the device address is taken

When opening a project, the project design view of ETS appears. The following figure shows the elements of this view.

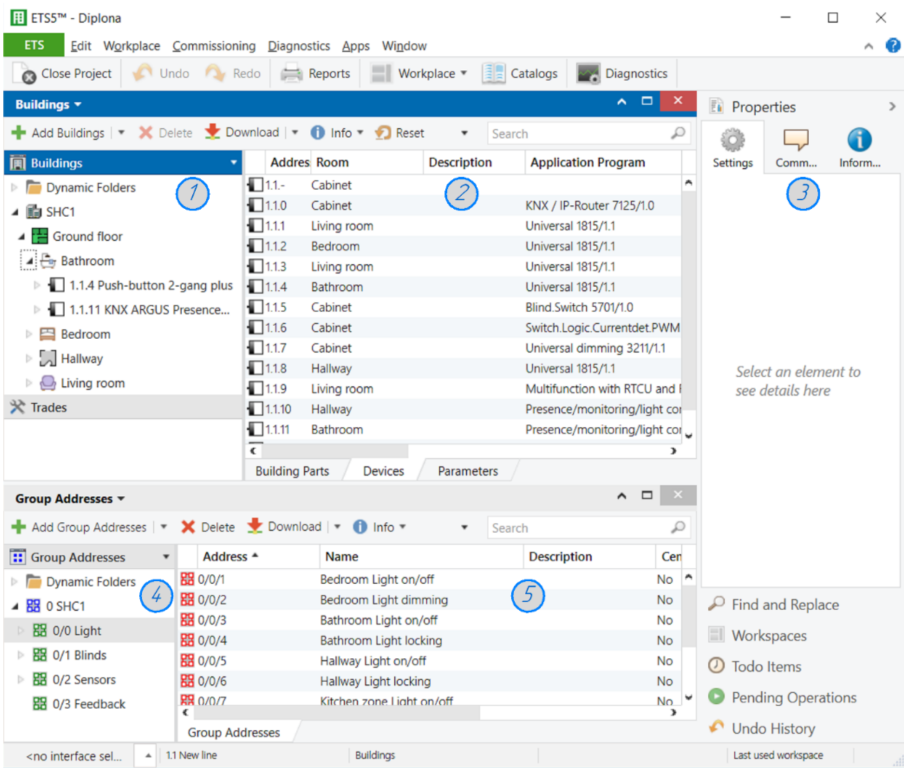

Figure 7 – Project design view ETS5

The Building panel is the central view of ETS. The Buildings panel is used to structure the KNX projects according to the actual building structure ① and to insert the KNX devices. Description view ② is used for setting devices parameters and other manipulations.

The Group addresses panel is used to generate and define group addresses ④. This view is required together with the Building view to link ⑤ the group addresses to the corresponding group objects.

There are also Topology, Catalog, Report, Devices views which can be opened via the Workplace toolbar.

Properties window is needed to specify some settings depending on the active panel. For example, device address for building panel and data type for group addresses panel.

3. Design building control

As a demonstration home automation configuration was designed according to technical specification. Technical specification was drafted with the participation of the diploma supervisor who took the role of customer. For project implementation building Smart Home Care 1 (SHC1) located on the campus of VSB Technical University of Ostrava was chosen.

3.1. Task specification

Technical specification:

Individual home with several rooms is used. List of rooms: Living room, Bedroom, Bathroom and Hallway. A smart home control system has two operation functions – light control and blind control. Requirements according to the room:

1. Living room
    - Light dimming control
    - Blind control
    - Temperature, CO_2, and humidity monitoring
    - Separate light control for kitchen zone
2. Bedroom:
    - Light dimming control
    - Blind control
3. Bathroom:
    - Light control
    - Presence recognition
4. Hallway:
    - Light control
    - Presence recognition

3.2. Equipment sourcing

According to the assignment, the equipment from Schneider Electric was selected. The company offers a single integrated system that meets international standards and a variety of solutions covering various technical areas, while ensuring flexibility, comfort, safety, and profitability, especially in new buildings.

All equipment was selected according to the ideology of flexibility and scalability of KNX. If necessary, in the future it will be possible to add additional equipment without replacing the power source, because it is currently set to 160mA. Free channels in the actuators are also provided for connecting additional equipment. Moreover, KNX/IP interface allows to connect home control system to the world web and extend its possibilities.

Table 1 lists the equipment used in the installation.

Table 1 – KNX devices specification

Order number	Product	Amount
MTN617225	Push-button 2-gang	4
MTN617125	Push-button 1-gang	1
MTN6005	KNX CO2, humidity and temperature sensor	1
MTN630919	Presence detector ARGUS	2
MTN649908	Blind/Switch actuator REGK 8x	1
MTN647595	Switch actuator REGK 4x230	1
MTN646991	Control unit 0-10 V (Dimming)	1
MTN684016	KNX power supply REG-K/160 mA	1
MTN680329	KNX / IP-Router REG-K	1
MTN681829	USB interface REG-K	1

The home plan with KNX symbols is presented in figure 9. A more detailed report with building structure and used devices generated by ETS5 is presented in Appendix A.

For such a small installation, line topology is used. All devices are connected consistently with twisted pair. An automatically generated topology report is presented in Appendix B.

A description of selected KNX devices can be discovered in Appendix C.

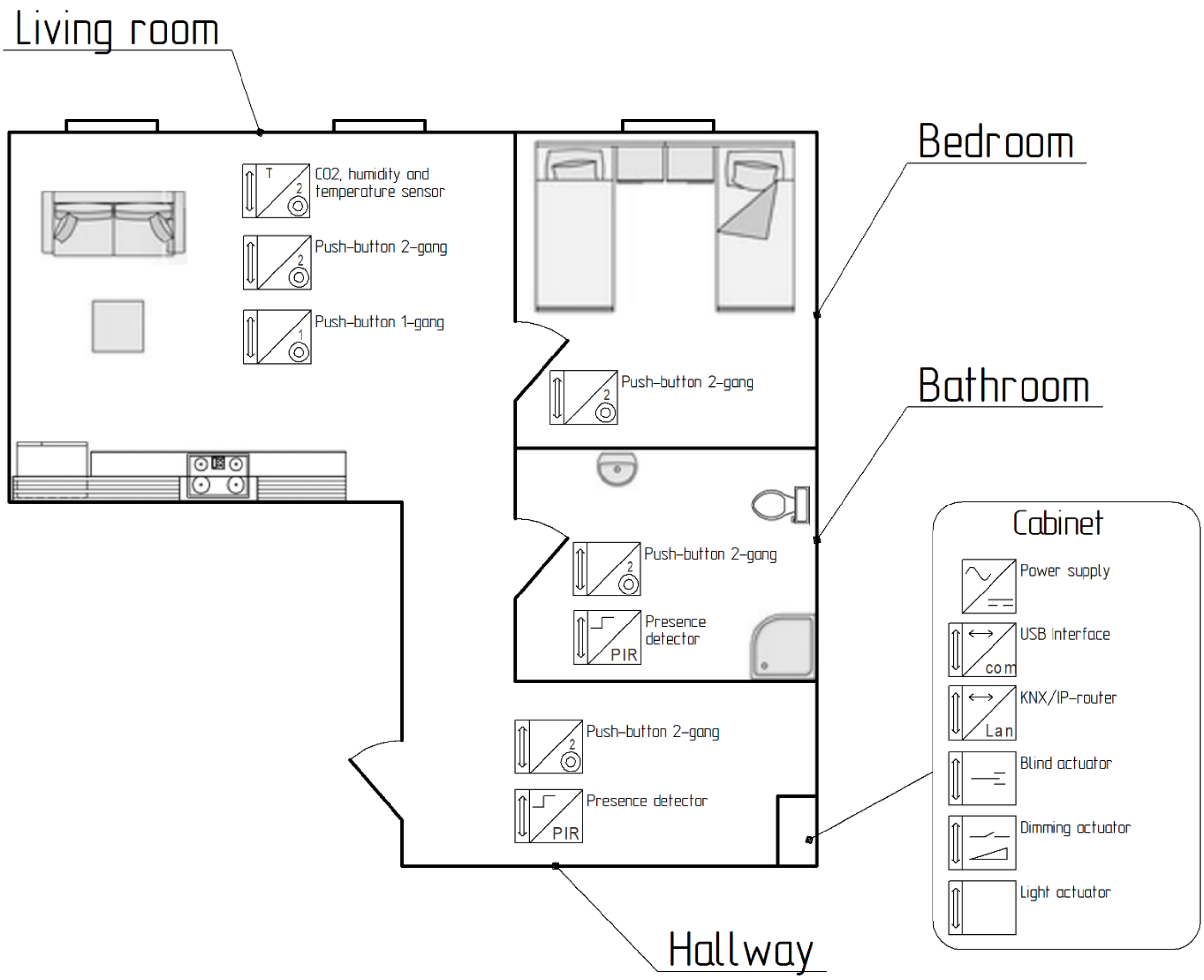

Figure 8 – Home plan with KNX symbols

3.3.Operation functions implementation

As it was mentioned earlier, operation functions can be implemented in KNX technology by the concept of group addresses. Group address represents a particular function and connected to the relevant equipment. For the implementation of operation functions, 3 level group addresses structure was designed. In this case, the main group is the building unit (e. g. floor, room), the middle group is a type of operation function (e. g. light control) and the subgroup – particular installation.

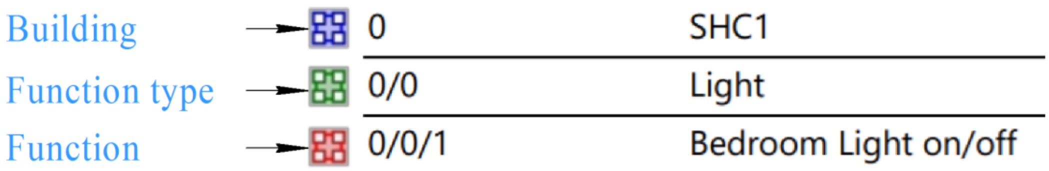

Figure 9 – Group address structure

The automatically generated group addresses report is presented in Appendix D.

Each device in the project needs to be programed using ETS 5. Some steps were done to complete the project:

1. Building structure was created.
2. All necessary devices were imported and placed in the building structure. Individual addresses were assigned. Topology was created.
3. Individual parameters of devices were adjusted.
4. Group address structure was designed and allocated with devices channels.

Next in the chapter detailed documentation is given. It describes how group addresses are allocated with equipment channels and which parameters were adjusted.

3.3.1. Living room

Light control in the Living room and Kitchen zone is implemented by a 2-gang push-button and dimming actuator. By short press PB1 and PB2 light in Living room turns on and off. By long press, it is possible to change brightness. The same functionalities have PB3 and PB4 for the Kitchen zone. Blind control is done by a 1-gang push-button and blind actuator. By short press, blinds make a step change and stop. With long press blinds move up and down.

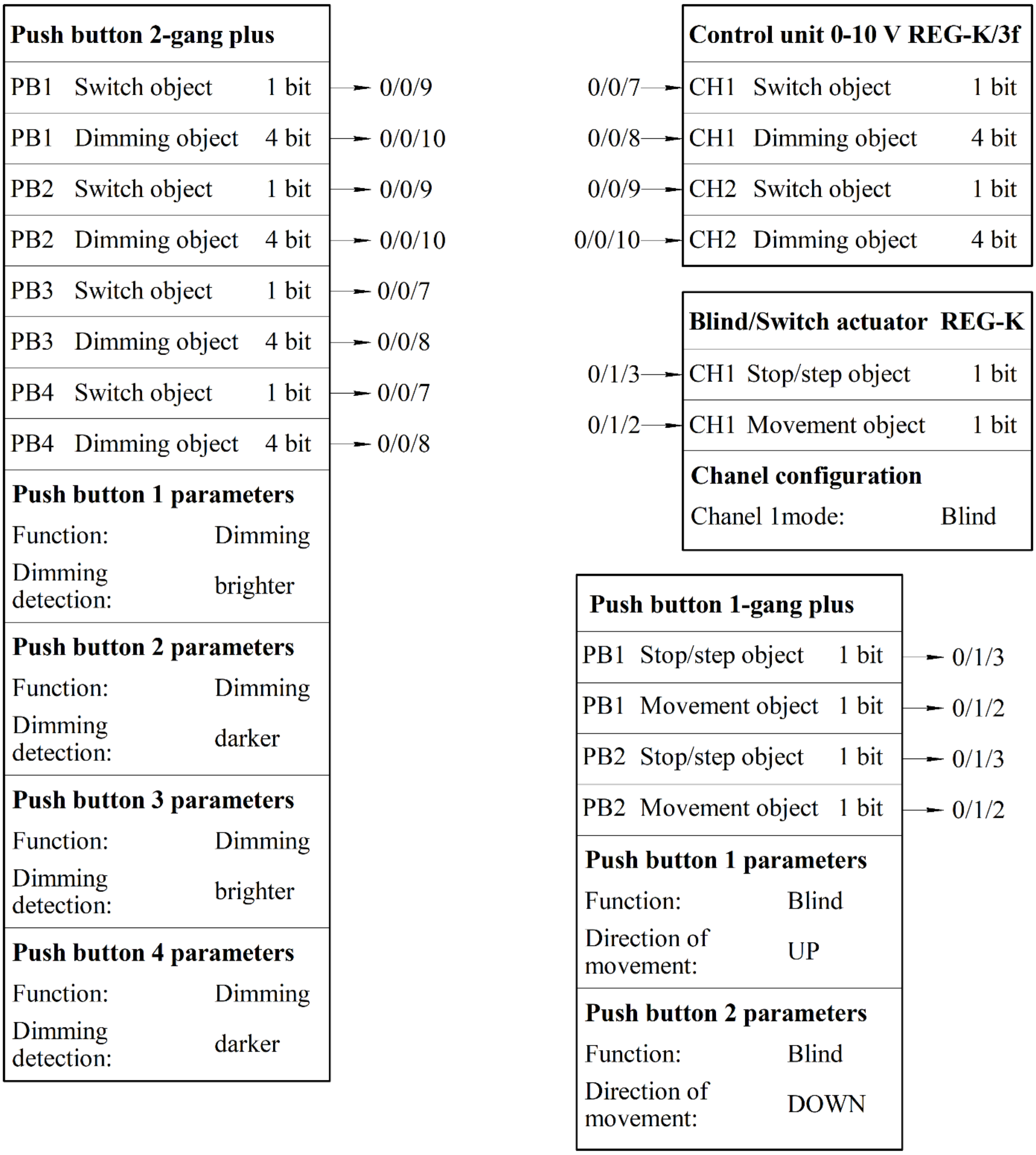

Figure 10 – Device's parameters in Living room

3.3.2. Bedroom

Light control in the Bedroom is implemented by a 2-gang push-button and dimming actuator. By short press PB1 and PB2 light turns on and off. By long press, it is possible to change brightness. Blind control is done by the same 2-gang push-button and blind actuator. By short press, PB3 and PB4 blinds make a step change and stop. With long press blinds move up and down.

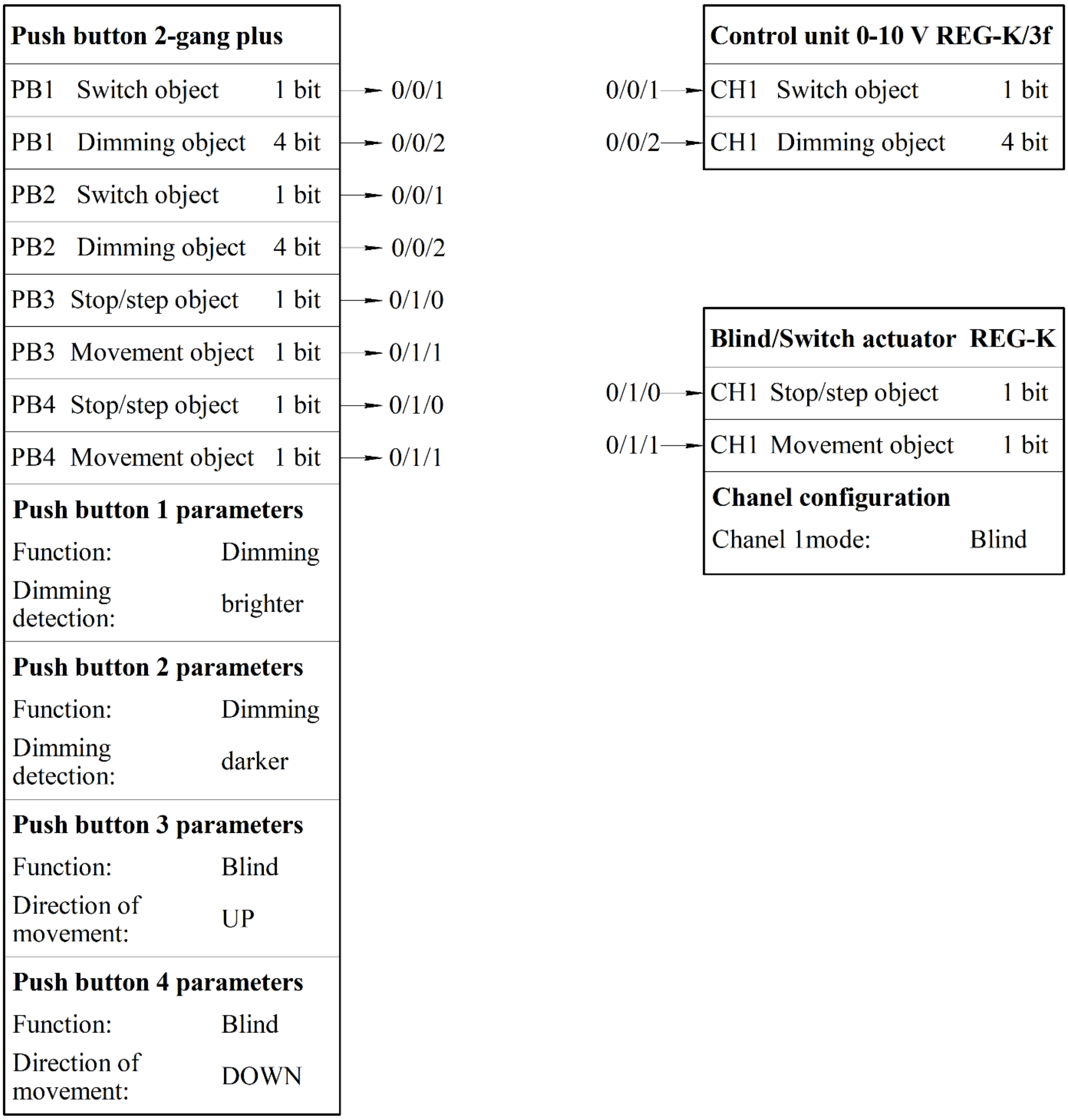

Figure 11 – Device's parameters in Bedroom

3.3.3. Bathroom

Light control in the Bathroom is implemented by a 2-gang push button, switch actuator and presence detector. By short press PB1 and PB2 light turns on and off. With short press PB3 and PB4, an automatic mode can be activated/deactivated. While this mode is active, the light turns on automatically according to the presence in the Bathroom. It also turns off after 5 minutes if no more presence detected.

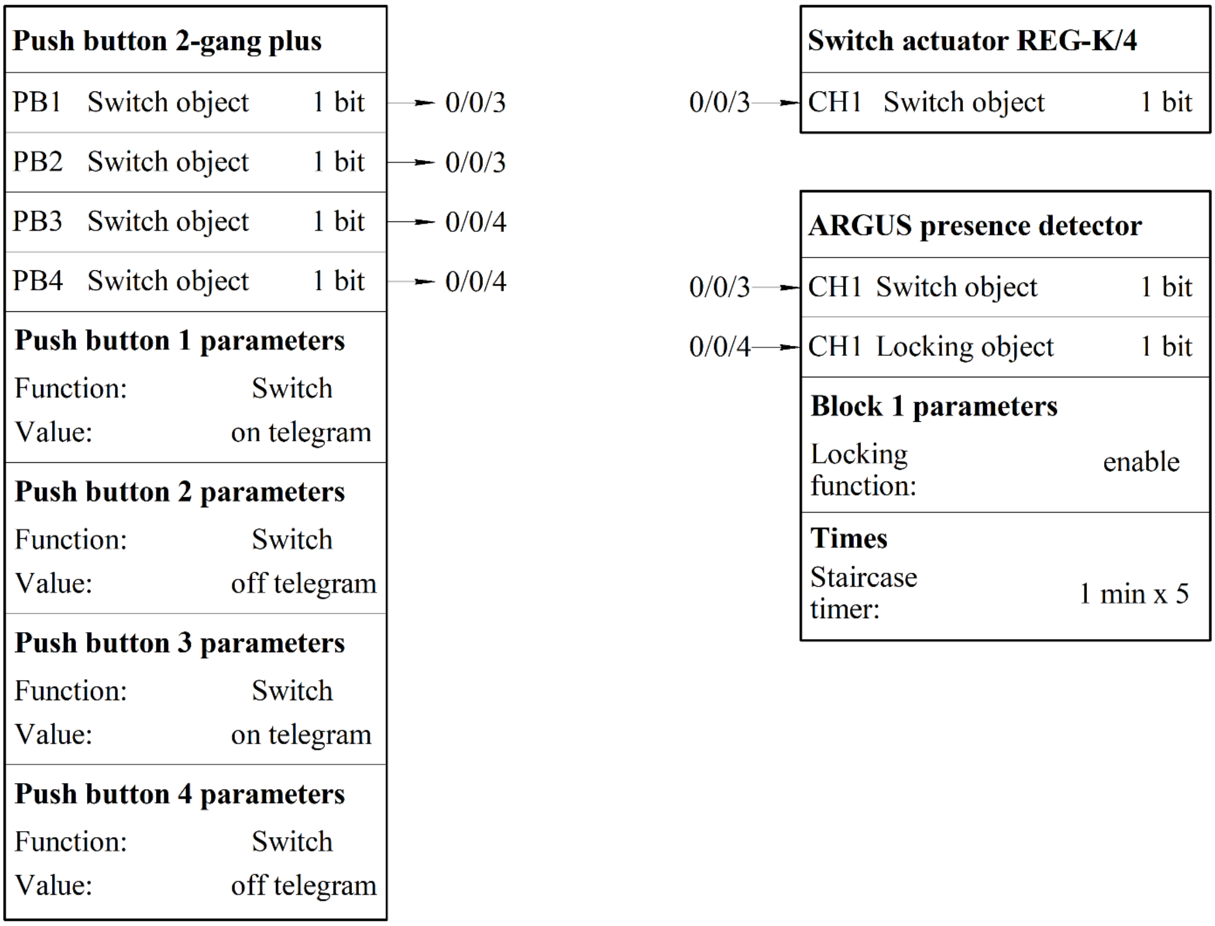

Figure 12 – Device's parameters in Bathroom

3.3.4. Hallway

Light control in the Hallway is implemented by a 2-gang push button, switch actuator and presence detector. By short press PB1 and PB2 light turns on and off. With short press PB3 and PB4, an automatic mode can be activated/deactivated. While this mode is active, the light turns on automatically according to a presence in the Hallway. It also turns off after 5 minutes if no more presence detected.

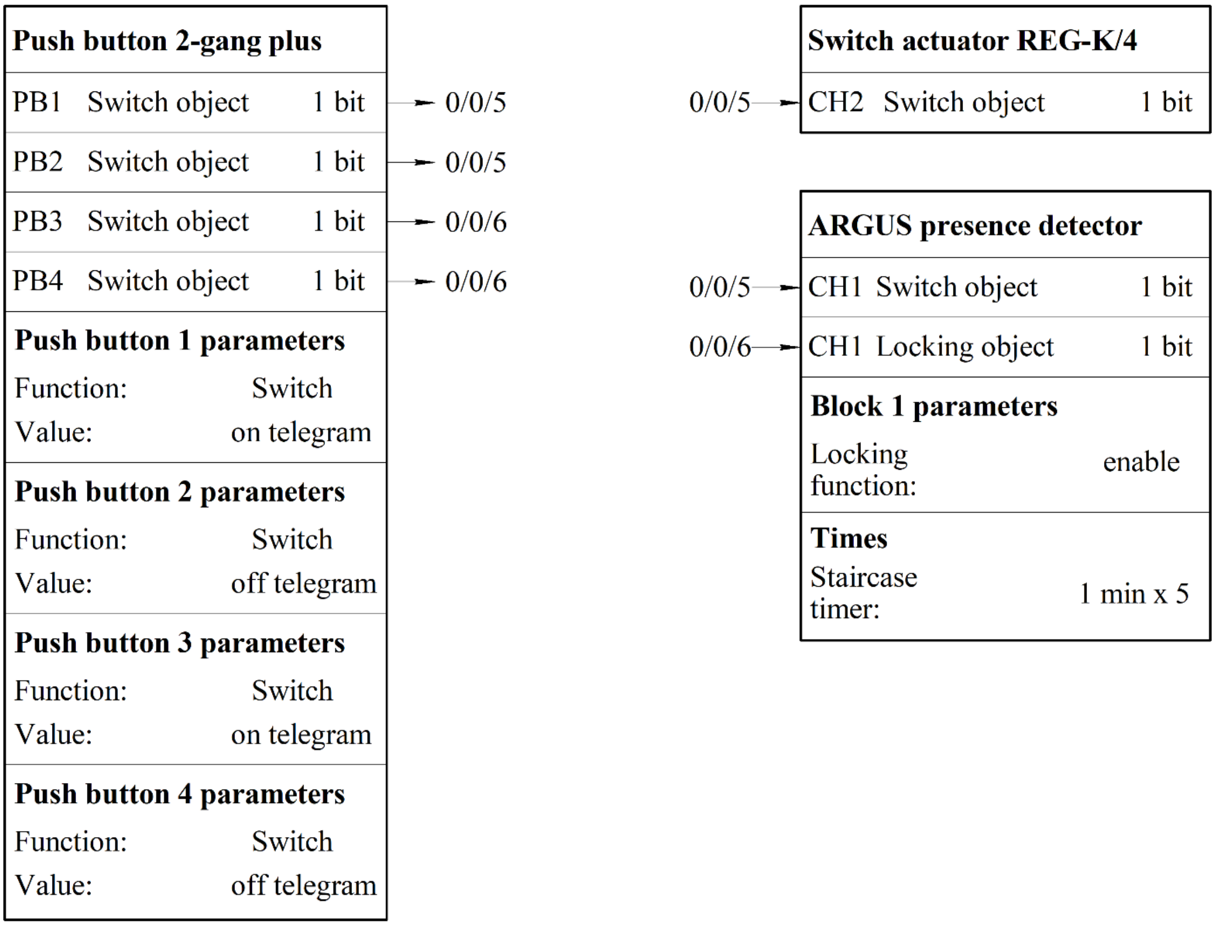

Figure 13 – Device’s parameters in Hallway

4. Human behavior collection

Human behavior collection in a smart home can be understood in many ways. Human behavior data in a smart home can be any interaction between owner and smart home. For example, that can be information of pressing control push buttons or information about time person comes home.

Data needs to be aggregated and stored somewhere for further usage. The most common solution is to connect a smart home to an IoT platform.

There are many implementations for collected data. However, all approaches are about predicting the owner's needs, i.e. recognize patterns in human's behavior and take action before.

4.1. IoT platform

The Internet of Things (IoT) is a concept of a computing network of physical objects ("things") equipped with built-in technologies for interacting with each other or with the external environment, considering the organization of such networks as a phenomenon that can restructure economic and social processes, excluding from part of actions and operations the need for human participation (23).

In terms of technology, IoT is a four-tier system: connected devices (sensors, sensors, terminals), the networks over which they interact, IoT platforms and applications for end-users. At the same time, the first two levels cannot be excluded from the structure, the platforms in the solution are variably present, the client interface is still present everywhere, but in the future, perhaps, the application level and other additional elements in management will disappear.

Interaction can be reduced to a backend application that analyzes human actions – and based on this analysis, form interaction with end devices without additional pressing of buttons. For example, a smart home system may know at what time (and at what brightness) the landlord usually turns on the light or at what time to heat the dinner, and when to turn on the TV, because the homeowner's favorite show starts. All this work based on templates stored in the system base, and it forms these actions without human intervention.

IoT platforms are becoming more important for smart homes as a lot of producers uses different communication protocols. With the increasing number of smart devices, more interfaces/applications are required. IoT platform allows to connect them all together and exchange information among them. Another function of the platform is monitoring the smart home state through a convenient user interface.

There are a lot of IoT platforms on the market such as AWS IoT, Google Cloud etc. The main advantage is high performance and the ability to adjust the server's capacity due to the traffic. However, they have two essential disadvantages. Firstly, they are located outside of the home and can be accessed by a third party. Another one, those are commercial products with a limited field of application.

This work focuses on openness**,** availability, and flexibility. Home Assistant meets all these requirements. This work uses a local IoT platform based on Raspberry Pi and Home Assistant software. Home Assistant (HA) is a popular open-source software written in Python. It supports an enormous number of protocols and vendors of smart devices. Because of the huge community around it is well-designed software with a lot of extensions. Due to local hardware, it is much harder to thief private data.

HA uses separate modules (integrations, or components) to control devices. It is easy to create one. The site contains a directory of core (community approved and supported) modules. Any module is a set of entities and services, in fact, functions. Entities store various data received from devices.

HA stores all the data necessary for the user in the settings folder “~ / .homeassistant”. The configuration files are written in YAML format, and the main one is “configuration.yaml”. It combines data from modules, automation, etc. The import feature allows splitting the settings into separate logically organized files. Modules are stored in the “components” and “custom_components” subfolders.

After installation, the HA graphical interface becomes available at http://localhost:8123. The first time you log in, you will need to create a user account.

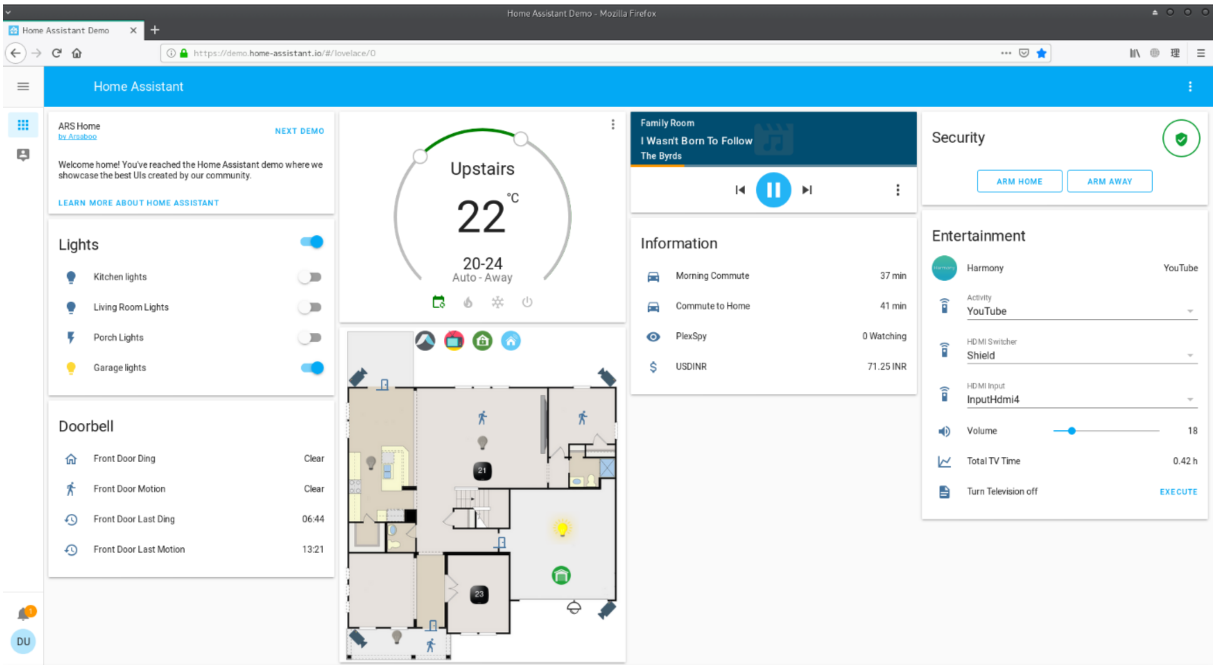

Figure 14 – Example of Home assistant user interface

Home Assistant can also perform monitoring function. Data from smart home can be visualized in the Lovelace interface by a variety of graphical dashboards. Visualization can be adjusted according to requirements. All entities in HA can be controlled from the Lovelace.

Automation can be done with help of .yaml files. For instance, it is possible to connect voice assistant and control light by voice commands. Another example is the automatic start of heating in the house when the person in the radios of 2 km from home. The smart home server tracks location with GPS in a smart phone.

According to the specification of smart devices, an interactive dashboard was designed. It represents states of entities inside Home Assistant. An entity is some element of a smart home e. g. a light bulb or temperature sensor. In the case of related KNX technology, it is a group address from the KNX bus.

The configurational file for KNX Technology is presented in Appendix E. In general, this file contains information about all entities (KNX group addresses), their type, and addresses. A new YAML file should be added to the configuration file.

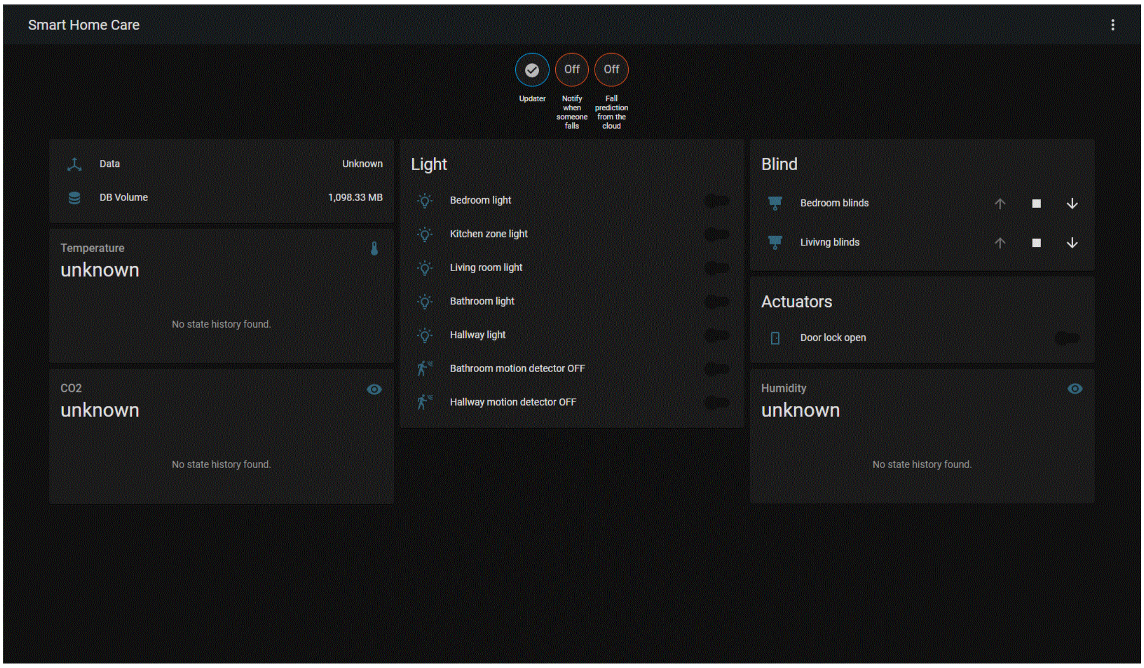

Figure 15 – Configured Lovelace of Home Assistant

4.2. Data collection

Any data needs to be stored somewhere. For this purpose, databases are used. Home Assistant uses SQLAlchemy that is Object Relational Mapper (ORM). In other words, HA supports any SQL backend: PostgreSQL, MySQL, MariaDB. There is a "recorder" integration in HA, which simplifies data collection and requires only some simple settings in configuration.yaml file:

```
recorder:
  db_url: mysql://USERNAME:PASSWORD@core-mariadb/homeassistant?charset=utf8
  purge_keep_days: 20
```

The type of SQL database is defined by db_url. In this case it is MariaDB. It was chosen because it is an open-source technology. There is a belief that only open-source technologies are reliable enough because their source code is open, and anybody can find and fix bugs. Since Home Assistant is installed on Raspberry Pi, the physical database is located on the microSD card of Pi.

Home Assistant gives users two options to extract data from the database. The first option is to use the inbuilt add-on Jupyter Lab. That is Python IDE which is run under HA software and accessible from the browser as well. Developers of Home Assistant designed a Python library for data extraction (24) .

This library based on Pandas and returns pandas data frames. Data from smart home can be presented in form of a table with entity id, state, and date-time mark (figure 24).

```python
from detective.core import db_from_hass_config

db = db_from_hass_config()
db.fetch_all_data(limit=2000000)

df = db.master_df
df[df.entity == 'sensor.co2'].tail()
```

	domain	entity	numerical	last_changed	state
195923	sensor	sensor.co2	True	2021-03-29 14:31:52	2624.0
196705	sensor	sensor.co2	True	2021-03-29 14:30:52	2592.0
198263	sensor	sensor.co2	True	2021-03-29 14:28:52	2572.8
199044	sensor	sensor.co2	True	2021-03-29 14:27:52	2543.36
199827	sensor	sensor.co2	True	2021-03-29 14:26:52	2512.64

Figure 16 – Data extraction with Jupyter Lab

The second option for data export is to apply SQLAlchemy request directly, instead of using a library based on it. To make it, a PC is required which is in the same network as Home Assistant. The IP address of HA and database credentials are used to build an engine and execute SQL query. Python programming language was applied for this purpose.

```python
from sqlalchemy import create_engine
import pandas as pd

SQLALCHEMY_DATABASE_URI =
'mariadb://USERNAME:PASSWORD@192.168.137.82:3306/homeassistant?charset=utf8'

query = """SELECT domain, entity_id, state, last_changed, attributes
         FROM states
         WHERE state NOT IN ('unknown', 'unavailable')
         ORDER BY last_changed DESC"""

with engine.connect() as connection:
    result = connection.execute(text(query)).fetchall()

columns = dict(result[0]).keys()
df = pd.DataFrame(result, columns=columns)
```

5. Pattern recognition in human behavior

The research area of recognition patterns in human behavior data is very wide. The purpose of the current work is to recognize fall accident using the wearable device with accelerometer and gyroscope sensors inside. Analyzing data smart home can recognize an emergency situation and send notifications to relatives or emergency. Moreover, it can send telegram via KNX bus to open the door and let in people to help.

This chapter describes the design of fall recognition application with wearable device in the smart home. The idea of the application is to collect accelerometer and gyroscope data from the wearable device, make fall prediction and call for some help in case of positive prediction.

The general description of the fall prediction algorithm is:

- → Data from sensors is transferred via MQTT to the HA server and stored in the database.
- → Every time it comes HA makes a POST request via HTTP to Amazon Web Service to receive prediction.
- → In AWS machine learning model is stored. It returns predictions in the form of 1/0.
- → In case of positive prediction double-check procedure starts.
- → After fall verification Home Assistant sends a notification via Telegram messenger and opens a door via KNX bus.

The architecture of the solution is presented in figure 25. The figure illustrates protocols and devices involved in the application.

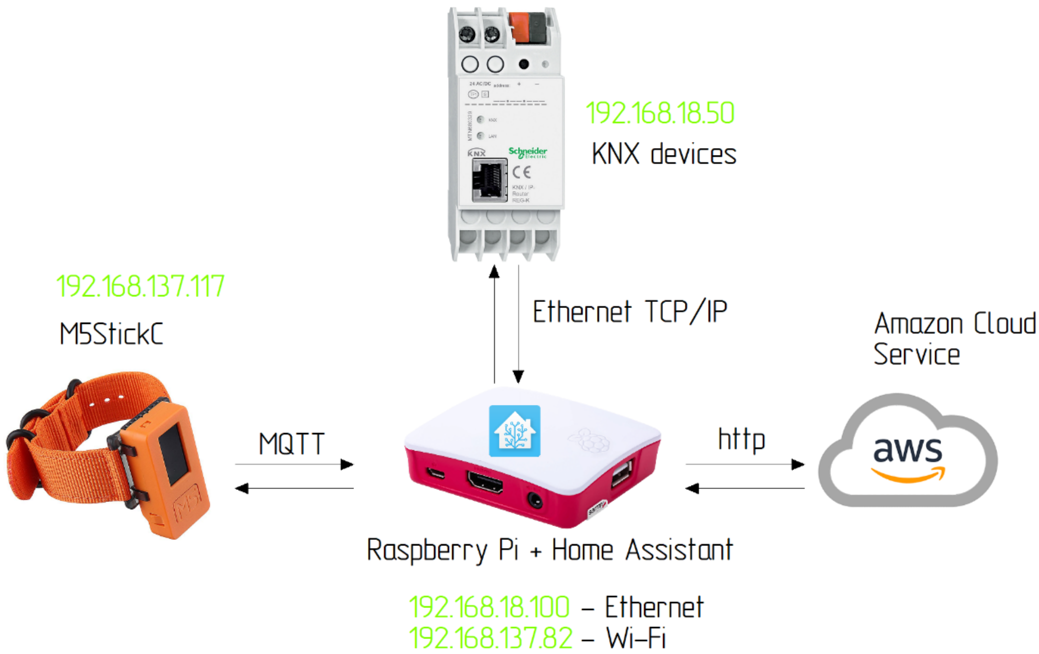

Figure 17 – Architecture of the solution

The next chapters explain the design process of fall prediction application in details and give details about the roles of particular device and technology.

5.1. Literature review (fall detection)

According to the research of the World Health Organization, around 30% of elderly in the age over 64 years fall each year (25). Moreover, people living in nursing homes fall more frequently than those who live in the community. At the same time percentage of elderly people is rapidly growing. United Nations researched that by 2050 proportion of people over 65 years old will increase up to 16% comparing to 9% in 2019 (26). Falls not only lead to physical injuries (fractures, concussions) (27), but also have remarkable psychological, medical, and social effects. Without on-time assistance falls may cause irreversible consequences for health (28).

Mentioned factors make wonder how to decrease costs related to the maintenance of old people and simultaneously improve health care. A new automation method should recognize fall case and inform emergency or one's relatives (29).

There is a lot of research in the field of fall recognition, and it is a well-studied area. The most common practice is to implement detection using a wearable device with an accelerometer and gyroscope (30). Wireless connection simplifies usage and low-powered sensors decrease maintenance level.

Another instance is a system which applies accelerometer data together with ZigBee-based system to localize fall using Received Signal Strength Indicator (RSSI) (31). The system uses ZigBee devices (radio-frequency based wireless devices) installed in several spots around the home to calculate the position of a person.

One more solution is to detect pre-fall and secure the person (32). Most of the research is focused on fall recognition when it has already occurred. It was researched that with modern hardware it is possible to recognize and send notification via a wireless connection.

Fall detection applications can be run on the smartphone also (33) This smartphone-based fall detection system supervises the activities of the patients, detects fall, and requests caregivers for help. It is proven that even with smartphones high accuracy can be achieved by using appropriate prediction techniques (34).

However, fall detection can be presented as an anomaly detection using thermal imaging. Researchers use a novel network that consists of two-channel 3D convolutional autoencoders which reconstruct the thermal data (35). Neural networks can also be applied to visual-based recognition systems which use a sequence of images as a source of data for fall recognition (36).

For this project fall detection using a wearable sensor method was chosen. Such a method provides high accuracy and has a low cost. Moreover, such an approach is very flexible: the number of rooms can be extended without extra expense, new flat can be fast equipped with a monitoring system.

In this work several ML models are used, such as k Nearest Neighbors, Support Vector Machine, Gradient boosting, and their performance is evaluated. These models have proven to be the most effective in terms of fall recognition (37) (38)https://ieeexplore.ieee.org/document/8688609. Publicly available data to tech models were taken from several sources (39).

5.2. Wearable device

The design of the fall prediction application started with a choice of a wearable device. There is a big choice of Arduino like hardware on the market that can be used for prototyping. However, for purposes of current work M5stick device from the M5stack company was chosen. It is ESP-32 based microcomputer with several sensors onboard. It is cheap and good enough for prototyping. Also, it supports MicroPython and it was an essential point because it saved time for the design control program for the device.

The main function of a wearable device is to send accelerometer and gyroscope data to the Home Assistant server. A local Wi-Fi network was utilized as a medium and MQTT was used as a protocol for data exchange.

MQTT is a pub/sub messaging protocol. The original version was published in 1999 by Andy Stanford-Clark of IBM and Arlene Nipper of Cirrus Link. They saw MQTT as a way to maintain communication between machines on networks with limited bandwidth or unpredictable connections.

Considering the harsh operating conditions, the protocol has been made small and light. It is ideal for low power devices with limited battery life. Thus, MQTT has become a protocol for streaming data between devices with limited CPU power and/or battery life, as well as networks with expensive or low bandwidth, unpredictable stability, or high latency. Therefore MQTT is known as the ideal transport for IoT.

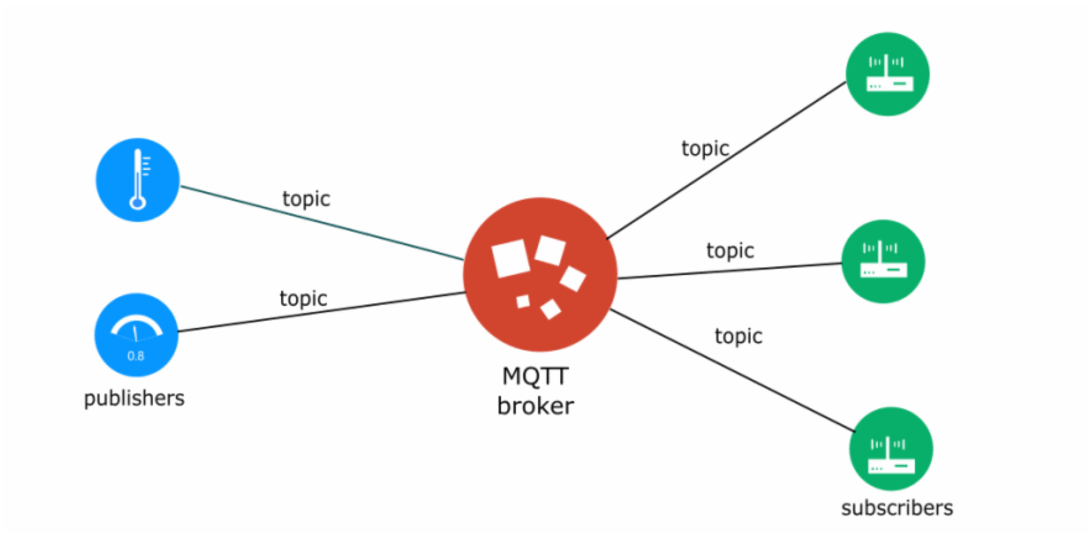

Figure 18 – MQTT protocol

M5stick publishes data to MQTT broker with a frequency of 10 Hz to the topic “m5Stick_data”. Details about the MQTT broker are described in the next chapter.

According to specific functionality, the control program for the device was designed. As mentioned above, M5Stick supports the MicroPython language which is an implementation of the Python programming language for microcontrollers. It enables the audience of this language to work with small computing devices using familiar syntax and programming principles.

M5tack company designed IDE for programming their devices – UI flow. It is a web-based programming tool that supports Blockly Visual Programming Language and Python code. Several libraries from the M5stack community were used in this solution. Their scope is defined in the table below. A full version of Python code is presented in Appendix F.

Table 2 – List of libraries

m5stack	The main library for interaction with the system in device and files.
m5ui	Library for user interface settings. (display, buttons)
m5mqtt	Library for communication via MQTT protocol.
imu	Library used for getting interaction with inertial measurement unit.
hat	Library to control extra hats like PWD unit or speaker.

Fall prediction system requires to be reliable since false positive predictions may cause undesirable consequences. To prevent false-positive predictions, double-check procedure is implemented. The concept of such a procedure is to inform the owner of the wearable device about the positive prediction and ask for a cancelling.

Technically it is realized via two MQTT topics: “fall_prediction” and “fall_event”. As soon as positive prediction is received from the cloud, it is published to the “fall_prediction” topic by Home Assistant. M5Stick subscribes to the topic, change the text label on the screen and starts beeping. If the person does not press the save button, the wearable device publishes the prediction to “fall_event”. Once fall is verified by the wearable device, Home Assistant sends a notification to the Telegram messenger.

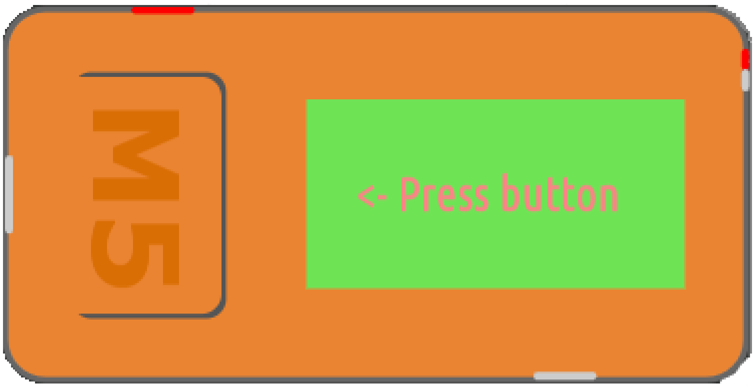

Figure 19 – Verification message

5.3. Data flow in Home Assistant

Home Assistant subscribes to the same MQTT broker and receives data ten times per second. Broker is located on the same device under Home Assistant, and it is Mosquitto broker add-on, which is an open-source lightweight protocol suitable for use on all devices from low power single board computers to full servers. Configuration of MQTT is done in configuration.yaml file.

```
mqtt:
  broker: 192.168.137.82
  port: 1883

sensor:
  - platform: mqtt
    name: "Data_M5stick"
    state_topic: "home-assistant/data"
```

Home Assistant provides many tools for making automation and one of them is Nod-RED. Node-RED is a streaming programming tool originally developed by the IBM Emerging Technology Services team and currently part of the JS Foundation. A key component of Node-RED is the streaming programming paradigm, which was invented in the 70s by Jay Paul Morrison.

Streaming programming is a way of describing the behavior of an application as a network of black boxes or "nodes" as they are called in Node-RED. Each node has a clear purpose - it receives some data, does something with this data, and then passes it on to the next node. The network is responsible for the flow of data between nodes. This model is great for presenting it visually as it becomes more accessible to a wide range of users.

Node-RED runs in the Node.js runtime and uses a browser to create or edit a flow ("Flow"). In the browser, you can create your application by dragging and dropping the required nodes ("Node") from the palette to the stage and linking them together.

All data flow in Home Assistant is built on Node-RED. For instance, accelerometer and gyroscope data is aggregated and sent to AWS via a sequence of nodes presented in figure 28.

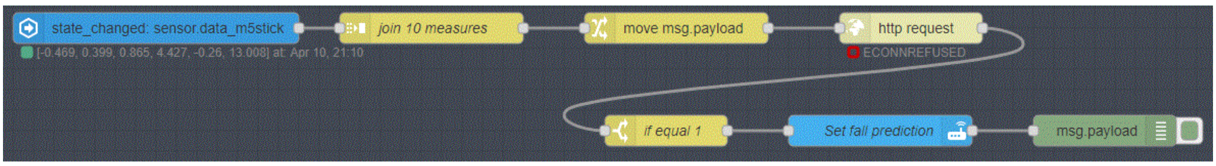

Figure 20 – Data flow in Home Assistant withing Node-RED

This sequence prepares data, sends it via an HTTP request to Amazon Web Service, receives prediction, and change variable state in case of positive fall prediction. JSON file with data is sent with POST request.

HTTP is a widespread data transfer protocol, originally intended for the transfer of hypertext documents (that is, documents that may contain links that allow you to navigate to other documents). HTTP stands for HyperText Transfer Protocol. According to the OSI specification, HTTP is an application (upper, 7th) layer protocol.

The HTTP protocol assumes the use of a client-server data transfer structure. The client application generates a request and sends it to the server, after which the server software processes this request, generates a response, and sends it back to the client. The client application can then continue to send other requests, which will be handled similarly.

A problem that is traditionally solved using the HTTP protocol is the exchange of data between a user application accessing web resources (usually a web browser) and a web server. At the moment, it is thanks to the HTTP protocol that the World Wide Web is supported.

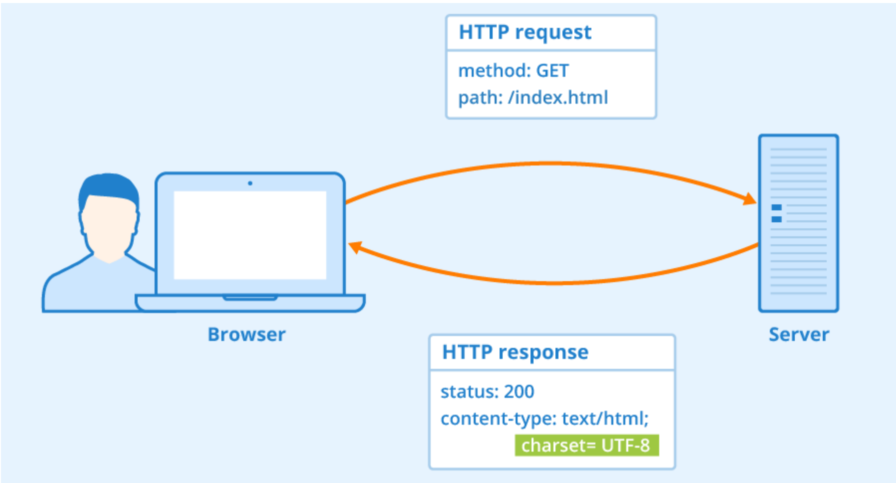

Figure 21 – HTTP protocol

As soon as a positive prediction is received, Node-RED stars other sequences to perform double-check and prevent false-positive predictions. It publishes prediction to the broker and waits for verification from the wearable device.

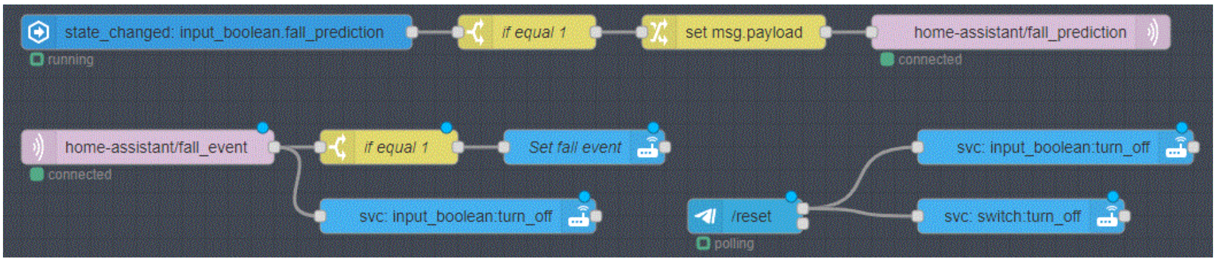

Figure 22 – Double-check sequence in Node-RED

After confirmation is received, Node-RED sends a notification via Telegram messenger. Telegram messenger is a highly encrypted messenger with an audience of more than 500 million users. It was chosen because it has well-configured documentation with API and it supports bots. Notifications from the bot can be addressed to any user who has messenger installed.

Any new bot in Telegram can be created with a help of BotFather, which is also a bot. With several commands, a new bot called SHC1_bot was created. BotFather provided an API key, which is used for the configuration of the Telegram node in Node-RED.

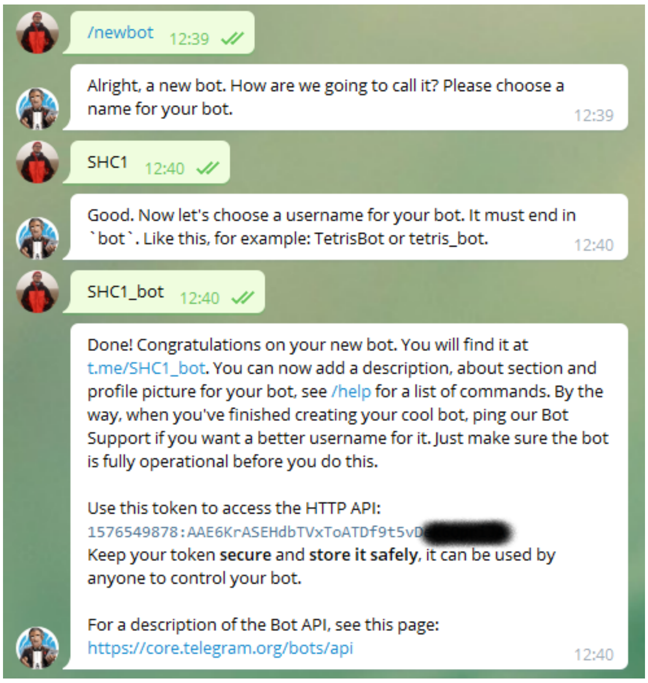

Figure 23 – New bot configuration

The state change of variable “fall_event” executes sequence for sending notification and opening the entrance door via KNX bus. Variable “fall_event” can be reseted by submitting a control command.

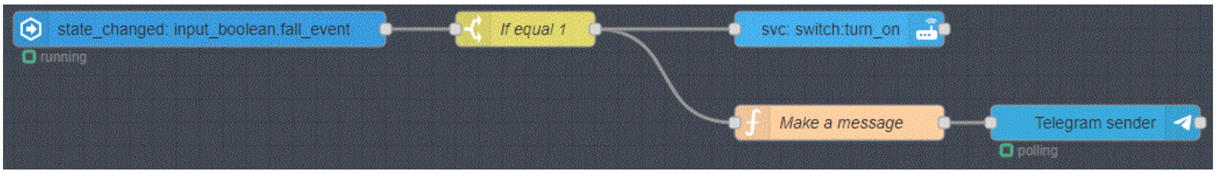

Figure 24 – Post-fall sequence in Node-RED

There is a function node with JavaScript code inside. This code makes the content of the message and specifies the user ID for a created bot.

```
msg.payload = {}
msg.payload.chatId = 421186142
msg.payload.type = "message"
msg.payload.content = "🚨🚨🚨 Habitant of SHC1 has fault. Call emergency!"
return msg;
```

5.3.1. Testing system

Any health monitoring system must meet high-reliability standards. The fall prediction system is not the exception. Daily activity monitoring of Smart Home habitant performed by the wearable device with accelerometer and gyroscope sensors. To have sufficient results, the habitant must wear the device all day long and accelerometer data needs to be constantly transferred to the Home Assistant server. However, a person may forget to wear it, or the device may disconnect from the Wi-Fi. The idea is to design such a testing system which can determine if a person has forgotten to wear a device or the device has lost the connection.

The wearable device is worn on the wrist. The main concept is to check sum acceleration vector at the moment when light or blinds are controlled via push buttons. If the sum vector is less than some value then the habitant forgot to wear the device.

It is not possible to compare Sum Vector (SV) with threshold value exactly at that time when the push button was switched due to synchronization difficulties. However, it is possible to organize a buffer that stores all SV values for the last 1 second. In fact, it looks like a FIFO concept – when new accelerometer data comes, the last value in the buffer is popped.

Figure 33 represents one sequence of nodes that takes a state of the Home Assistant entity with wearable sensor data and calculates the sum vector for current measurement. Then it stores in the global variable “flow” which is a buffer.

Figure 25 – Sequence for sum vector aggregation in Node-RED

As far as MQTT protocol supports only text messages exchange, it is necessary to unpack string and find axis accelerations. Code bellow describes the content of functions.

```
let arr = msg.payload.replace(/[\])}[{(]/g, '').split(',').splice(0, 3)
let x = arr[0]
let y = arr[1]
let z = arr[2]
let SV = Math.sqrt(x*x + y*y + z*z);

msg.SV = SV

flow.set('SV', SV);

return msg;
```

Figure 26 – Function Calculate SV in Node-RED

```
//First In First Out

let SVarr = flow.get('SVarr');

SVarr.unshift(msg.SV)

if (SVarr.length > 10) {
    SVarr.pop()
}

flow.set('SVarr', SVarr);

return msg;
```

Figure 27 – Function Aggregate SV in Node-RED

The next sequence executes only if state changes in light and blind entity domains happens. If the maximum value of Sum vector in a buffer greater than a threshold value, then the Find max SV function triggers out and notification is sent.

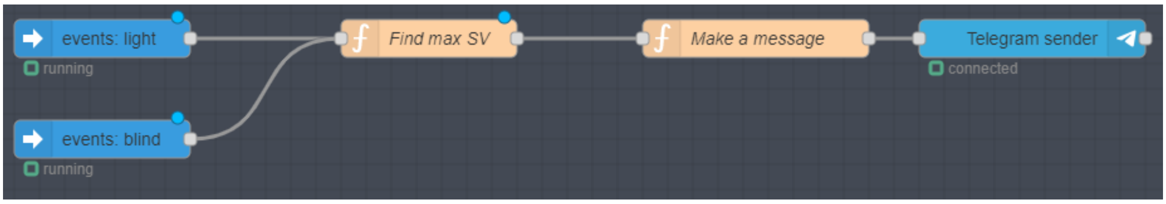

Figure 28 –Sequence for notification in Node-RED

```
let SVarr = flow.get('SVarr')

function getMaxOfArray(numArray) {
  return Math.max.apply(null, numArray);
}

let SVmax = getMaxOfArray(SVarr)

msg.SVmax = SVmax

if (SVmax > 1.2) {
  msg.trigger = true;
} else {
  msg.trigger = false;
}


return msg;
```

Figure 29 – Function Find Max SV in Node-RED

To ensure that the wearable device is constantly connected to the Wi-Fi network another testing system is used. It listens to the variable with data and starts a timer if data flow stopped. After 5 seconds it checks that data flow stepped at the daily time and sends a notification.

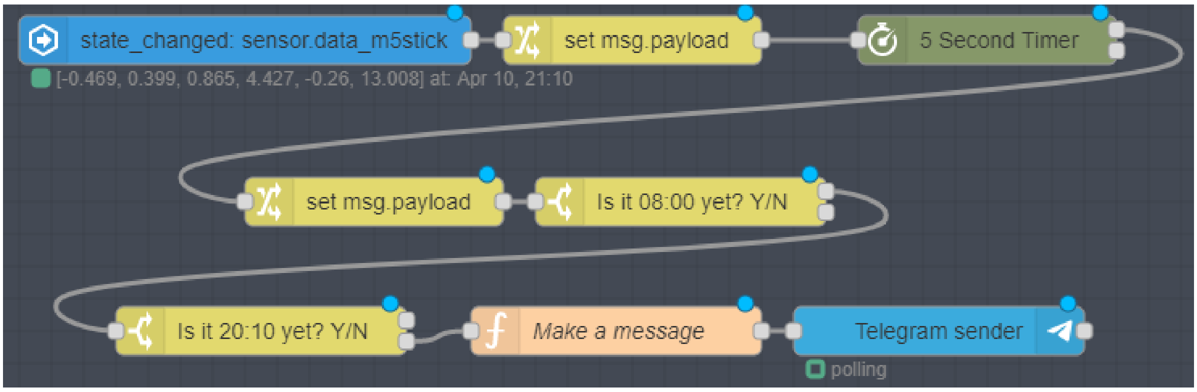

Figure 30 – Testing system to ensure stable Wi-Fi connection in Node-RED

During exploitation Telegram messenger of the end-user may look such way as presented in figure 39.

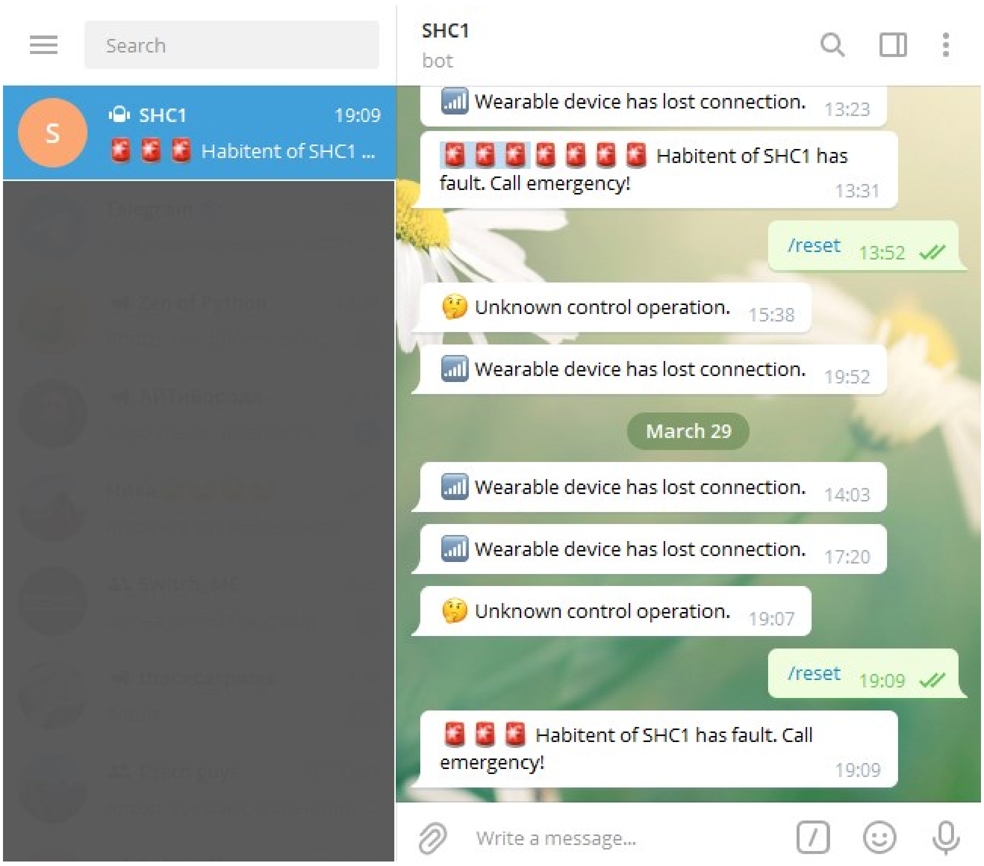

Figure 31 – Notification in Telegram messenger

5.4. Machine learning model

Machine Learning (ML) is an extensive subset of artificial intelligence that studies methods for constructing algorithms that can learn. In more detail, it is a data analysis technique that allows a machine/robot/analytical system to independently learn by solving an array of similar problems.

Thanks to machine learning, the programmer does not have to write instructions that take into account all possible problems and contain all solutions. Instead, a computer (or a separate program) is loaded with an algorithm for finding solutions on its own through the complex use of statistical data, from which patterns are derived and on the basis of which predictions are made.

Machine learning technology based on data analysis dates back to 1950 when the first programs for the game of checkers began to be developed. Over the past decades, the general principle has not changed. But thanks to the explosive growth of the computing power of computers, the patterns, and predictions they create have become many times more complex, and the range of problems and tasks that can be solved using machine learning has expanded.

To start the machine learning process, you first need to download the Dataset (some amount of initial data) to the computer, on which the algorithm will learn to process requests. For example, there may be pictures of dogs and cats that already have tags indicating who they belong to. After the training process, the program itself will be able to recognize dogs and cats in new images without tags. The learning process continues after the forecasts are issued, the more data is analyzed by the program, the more accurately it recognizes the required images.

All problems solved with ML fall into one of three main categories:

- The problem of regression is a forecast based on a sample of objects with different characteristics. The output should be a real number (2, 35, 76.454, etc.).
- The task of classification is to obtain a categorical answer based on a set of features. Has a finite number of answers (usually in the "yes" or "no" format).
- The task of clustering is the distribution of data into groups.

There are plenty of ML algorithms, and most of them are based on statistics. Modern algorithms in fact are statistical methods covered implemented in the code. Due to the increase of PC capabilities, machine learning has become very powerful. There are several algorithms that are widely used in many ML applications:

1. Decision tree

This is a decision-making method based on the tree graph. Model evaluates potential consequences (with the calculation of the probability of an event occurring), efficiency, resource consumption and makes decision.

From the business point of view, this tree is built of a minimum number of questions that can be answered in two ways – yes/no. After giving answers to all those questions, model comes to the correct choice. The advantages of a decision tree are that it structures the problem so that it can be presented to the business, and the final decision is made based on logical decisions.

2. Naive Bayesian classification

Naive Bayesian classifiers belong to the family of probabilistic classifiers. It comes from Bayes' theorem, where it considers functions as independent (strict or naive assumption).

3. Logistic regression

Logistic regression is a way to determine the relationship between variables, where one of them is categorically dependent and the rest are independent. To do this, it uses the logistics function (cumulative logistics distribution). From the practical point of view, logistic regression is a powerful statistical method for predicting events with one or more independent variables.

4. Support vector machine (SVM)

This is a whole set of algorithms needed to solve classification and regression analysis problems. Since an object in N-dimensional space belongs to one of two classes, the support vector machine builds a hyperplane with dimension (N - 1) so that all objects are in one of two groups. On paper, this can be represented as follows: there are points of two different kinds, and they can be linearly separated. In addition to separating points, this method generates a hyperplane in such a way that it is as far away from the nearest point of each group as possible.

5. kNN

k Nearest Neighbours is one of the simplest classification algorithms, also sometimes used in regression problems. It is sometimes called a lazy algorithm because it calculates distance only at the moment of prediction. To classify each of the objects of the test sample, the following operations must be performed:

- Calculate the distance to each sample objects of training set.
- Select k objects that has minimal distance to the predictable object.
- The class of the object to be classified is the class that occurs most frequently among the k nearest neighbours.

6. Ensemble method

This method is based on machine learning algorithms which generate a set of classifiers and split objects from new data based on voting results. Originally, method was some case of Bayesian averaging. However, has it became more complex and added some additional features:

- Boosting - transforms weak models into strong ones by forming an ensemble of classifiers (from a mathematical point of view, this is an improving intersection).

- Bagging - collects sophisticated classifiers, while simultaneously training basic ones (improving union).
- Correction of errors of output coding.

The ensemble method is more powerful than stand-alone forecasting models because:

- It minimizes the impact of chance by averaging the errors of each base classifier.
- Reduces variance, since several different models based on different hypotheses are more likely to arrive at the correct result than one taken separately.
- Excludes going beyond the set: if the aggregated hypothesis is outside the set of basic hypotheses, then at the stage of forming the combined hypothesis, it expands using one or another method, and the hypothesis is already included in it.

When talking about machine learning it is important o mention two more concepts:

- Normalization

Normalization is the conversion of data to certain dimensionless units. Sometimes - within a given range, for example, [0..1] or [-1..1]. Sometimes - with some given property, such as a standard deviation of 1.

The key goal of normalization is to bring different data in a wide variety of units and ranges of values to a single form that will allow you to compare them with each other or use them to calculate the similarity of objects. In practice, this is necessary, for example, for clustering and in some machine learning algorithms.

- Cross-validation

The crucial task of machine learning algorithms is their ability to generalize, that is, to work well on new data. Since the quality of the constructed model cannot be immediately checked on the new data (after all, it is important to make a forecast for them, that is, not know the true values of the target feature for them), need to expense a small portion of data in order to check the quality of the model performance on it.

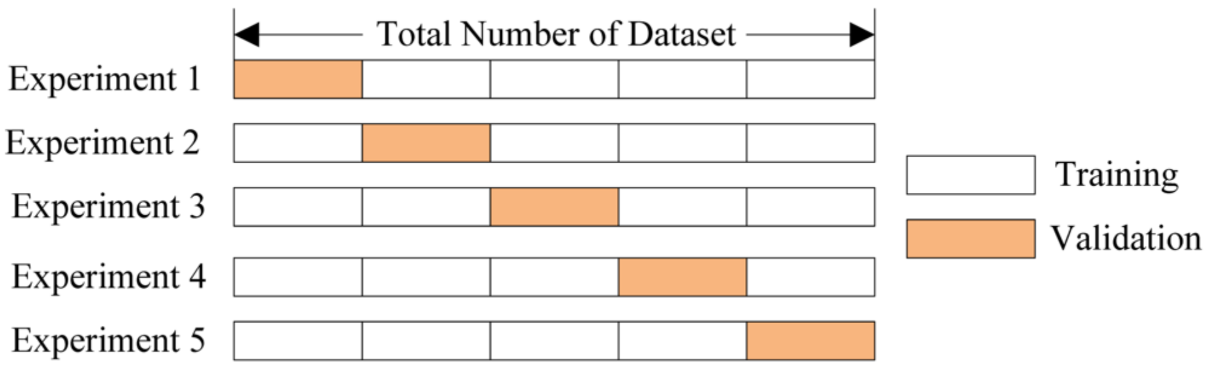

Figure 32 – Cross-validation

Here the model is trained K times on different (K-1) subsamples of the original sample (white) and tested on one subsample (each time on a different, orange colour). It happens that K model quality scores are obtained, which are usually averaged, giving the average cross-validation classification/regression quality score.

Cross-validation gives a better estimate of the quality of the model on new data compared to deferred sampling. But cross-validation is computationally expensive if there is a lot of data.
For training machine learning model for fall prediction publicly available dataset HAR-UP with daily activities is used (39). It includes data from several inertial measurement units (IMU) and infrared cameras. Data was collected from 17 subjects during 11 activities. Among others, it contains data from wrist wearable IMU.

Number	Activity	Duration (s)
1	Falling forwards	10
2	Falling on knees	10
3	Falling backwards	10
4	Falling sideways	10
5	Falling from knees	10
6	Walking	60
7	Standing	60
8	Sitting	60
9	Picking up	10
10	Jumping	30
11	Lying	60

Figure 33 - List of activities performed in the database (40)

Raw acceleration and angular velocity data are not useful in terms of activity recognition. On the other side, features extracted from aggregated data can make an impact on solving such a problem. Before extracting features, it is important to understand how data should be aggregated. Current work researched tree aggregation time-windows: 1&0.5, 2&1 and 3&1.5. For each time-window, the first number refers to the window length (in seconds) in which the features were calculated, while the second number to the period (also in seconds) in which a new calculation was done. This is illustrated in figure 42. The most suitable 1&0.5 time-window was researched.

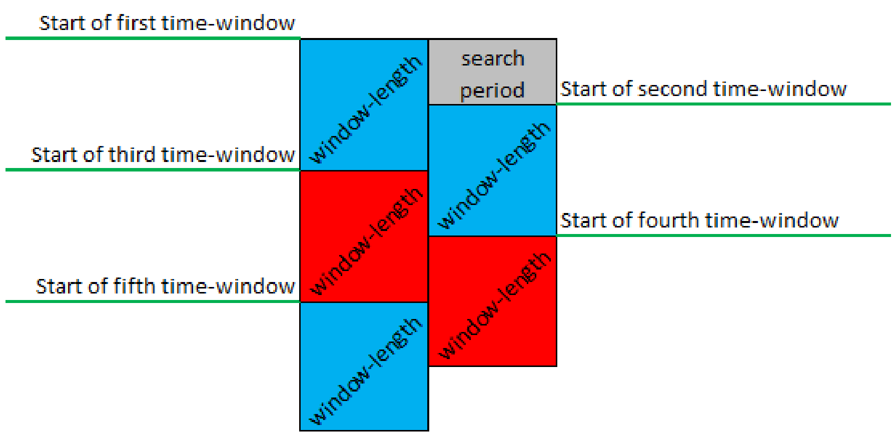

Figure 34 – Data aggregation with time-window

When solving problems with sensors, the main task is to find good signs. Even with a slight change in the orientation of the wearable device in space, the signals received from the sensors can change dramatically. Feature extraction from time series is a separate topic and requires deep research. Thus, features were selected according to recommendations and results of several works which deal with time series of acceleration and angular velocity data (41) (42).

The most promising features selected from the articles and implemented in current work are:

- Mean, max, min, zero-crossing, standard deviation
- Magnitude Mean Value
- Signal Magnitude Area
- Average Absolute Difference

Since fall recognition is a classification problem, tree models were selected for performance comparison: SVM, kNN, and Gradient boosting. The research was held in a Python development environment with Sklearn and XGboost libraries. Both are very common in the data science community because they allow to implementat ML models out of the box.

The machine learning process for different models includes the following steps:

1. Dataset export. For the training model, only wrist IMU data from the initial dataset is required.
2. Change target Activity label. Because the initial dataset contains 11 activities and work's problem is binary classification, target activities (1, 2, 3, 4, 5) are changed label 1, and other activities are changed on label 0.
3. Time-window aggregation 1&0.5.
4. Splitting the data. Preprocessed data is splatted into training and test sets in the proportion of 70/30.
5. Normalization. The train set is rescaled in such a way that mean=0 and std=1.
6. Feature extraction.
7. Model training. With 5-fold cross-validation most suitable hyperparameters were found.

Three models were trained and evaluated on the different test set. Confusion matrices are presented in the figure below.

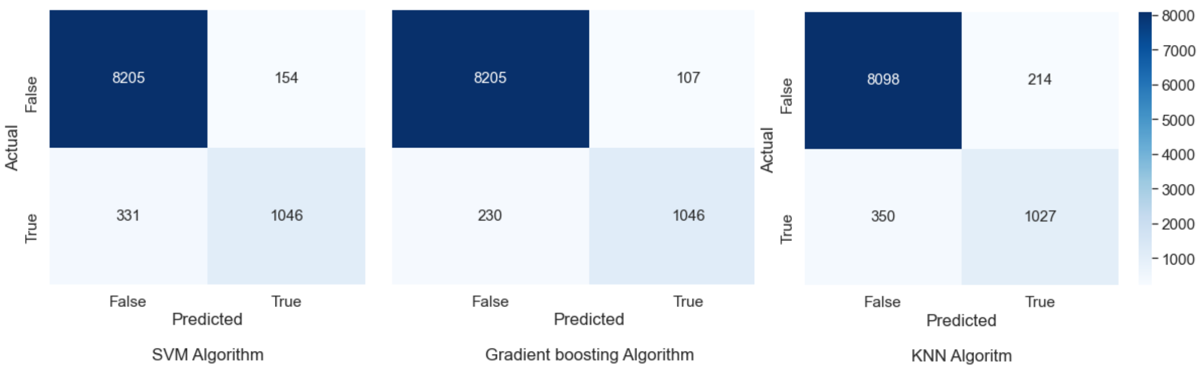

Figure 35 – Confusion matrices

As far as data is imbalanced because of the rare fall event, simple metrics like accuracy are useless for the evaluation performance of the models. Special matrices were selected in order to evaluate the quality of the algorithm for each of the classes separately:

$$precision = \frac{TP}{TP+FP} \tag{1}$$

$$recall = \frac{TP}{TP+FN} \tag{2}$$

Precision can be interpreted as the proportion of objects called positive by the classifier and at the same time positive, and recall shows what proportion of objects of a positive class from all objects of a positive class the algorithm found. It is the introduction of precision that does not allow to write all objects into one class since in this case, an increase in the False Positive level reaches. Recall demonstrates the ability of the algorithm to detect a given class in general, and precision demonstrates the ability to distinguish this class from other classes.

There are several different ways to combine precision and recall into an aggregated measure of quality. F-score – harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision+recall} \tag{3}$$

The F1-score reaches its maximum when the recall and precision are equal to one and is close to zero if one of the arguments is close to zero.

Comparison of algorithms according to introduced metrics is presented in table 3.

Table 3 – Best performance obtained for each model.

Model	Precision	Recall	F1-score
SVM	87.17%	75.96%	81.18%
Gradient boosting	90.72%	81.97%	86.13%
KNN	82.76%	74.58%	78.46%

From the obtained scores it is seen that the maximum F1 score is 86% which is not sufficient for industrial application, but good enough for a minimum viable product. After comparison model's performance Gradient boosting model was chosen. It was programmed with XGboost classifier and exported to the pickle file for further deployment in the fall prediction application.

5.5. Deployment in AWS

A core component of the fall recognition application is a service that makes predictions based on income data. The previous chapter explained how to translate coming data into prediction using a mathematical model. However, any model requires infrastructure to become real services that satisfy requirements of time response and load.

Initial idea was to deploy the ML model in Raspberry Pi and make an interconnection between Home Assistant and Debian OS. Nevertheless, such an approach hardly complicates the solution because some interface between the operation system and HA needs to be designed. Moreover, the operation system would take extra space on an SD card, which is used as data storage.

After researching the best practices of development systems based on machine learning technologies, a cloud solution was chosen. It means that ML service is deployed in some remote machine and accessible from the internet. Cloud service allows to scale the number of connections, support reliable functionality, and reduce costs.

AWS is considered for deployment. Amazon Web Service (AWS) is the largest cloud service, provider. Common cloud services include storage, computing power, databases, analytics, security, monitoring, developer tools, networks. Amazon provides free tariffs like EC2 with instant t2.micro up to 750 working hours a month, which is in fact totally free. A server with such capacity is a proper option for low-loaded services like a fall recognition system.

Application for fall detection was designed with Python. Firstly, code upload scaler and ML model from pickle files. Then code receives JSON with format {key: [arrau1…array10]} where each array consists of 3 axis accelerometer and 3 axis angular velocity data. Data is packed into a pandas Data frame and divided into time-windows. It is also normalized with the mean and std of the original dataset. From raw data, the same features that were used to train the model are extracted. Finally, the model makes a prediction based on pre-processed incoming data. Since there are 2 divided data frames, the model returns 2 predictions and only one most informative is chosen.

Still, the application is not ready to receive and return data. It should be covered in web application with the Flask framework. Flask is a micro-framework for building websites in Python. The application was covered with a Flask to be accessible on the address hhtp://WEBSITE_IP/predict. The full code of an application is presented in Appendix G.

The goal is to run a web service on a cloud virtual machine. The cloud virtual machine may be ran under any OS. Containerized web server does not have the troubles with environment-related issues like invalid libraries. If the containerized code works on one machine, it will definitely run on another irrespective of the characteristics of the machine. Docker is the most common technology for containerization, and it is used in the current application.

The application was placed into a Docker container and uploaded into the AWS server. There were activated a port 80 in the container to make it reachable. After all manipulation web service for fall recognition has become available with HTTP request:

```
POST /predict HTTP/1.1
Host: ec2-3-143-4-97.us-east-2.compute.amazonaws.com
Content-Type: application/json
```

6. Experiment and results

To ensure reliability and functionality of HBC in SH within IoT long term experiment was held in the VSB dormitory. Only one person participated in the experiment. Case with KNX devices was used to simulate operation functions in smart home. During the day participant used KNX devices to simulate daily activity. During this time data from CO2, humidity and temperature sensors had been collecting.

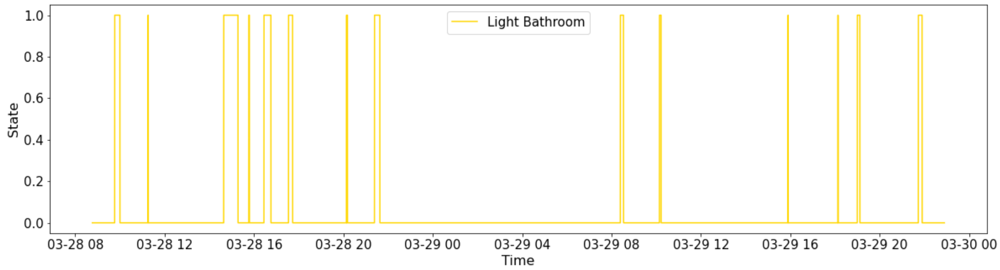

Figure 36 – State change of the light in Bathroom

For example, it is visible that bathroom was used around 7 times a day. It was used only daily.

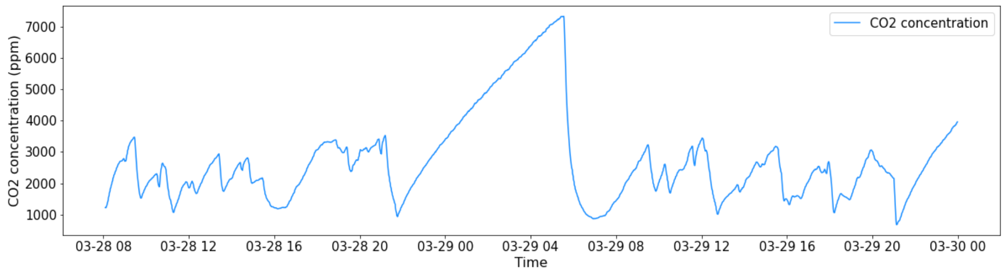

Figure 37 – State change of CO2 concentration

Another illustration of a pattern in collected data is CO2 concentration. It is visible that at night time concentration increases dramatically. During the day it falls when the door or a window is open. The same situation happens with humidity and temperature.

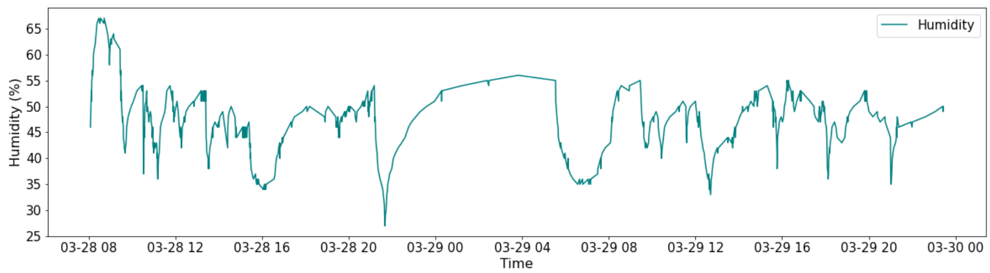

Figure 38 – State change of humidity

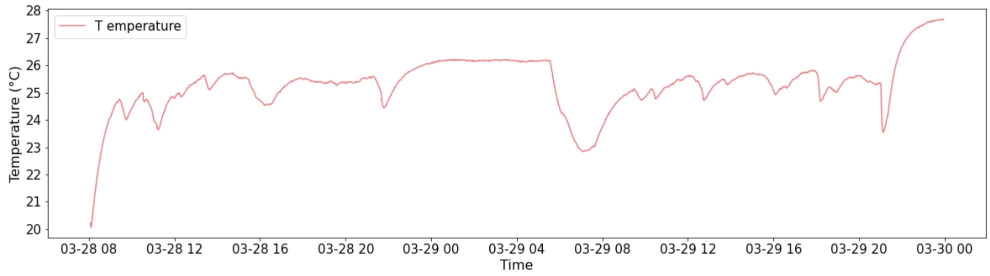

Figure 39 – State change of temperature

Also, participant wore M5Stick device daily to ensure the functionality of the fall prediction system. Through the experiment, false predictions were not obtained. The participant performed several different falls: forward, backward and slideway.

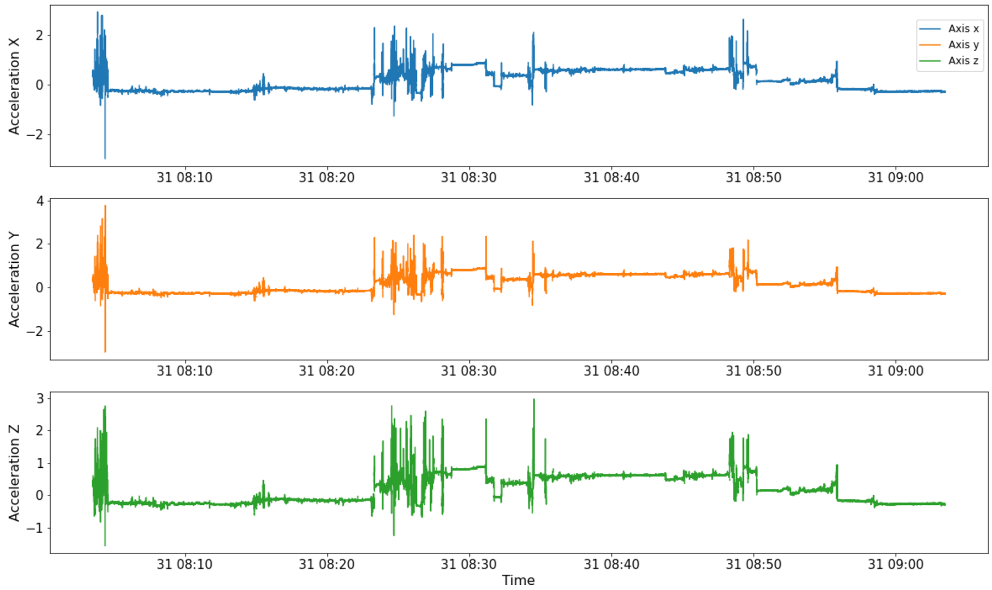

Figure 40 – One hour sample of acceleration data

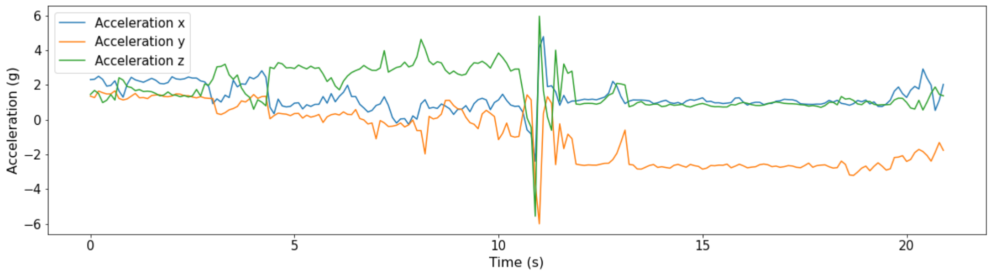

Figure 41 – Falling forward acceleration data sample

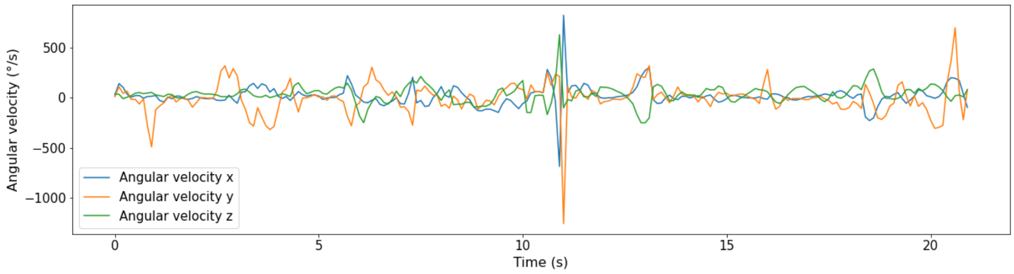

Figure 42 – Falling forward angular velocity data sample

As a result, it was obtained that the HBC system properly collects data from smart home and IoT devices and stores it in a database. The fall recognition system works fine as well and does not have time lags in predictions. However, full testing of the system is only possible with a bigger number of participants to test more different conditions.

Conclusion

During work on this diploma, the HBC system was designed. It is based on Raspberry Pi hardware and Home Assistant software and collects data from KNX building control technology and wearable IoT device M5Stick. There were developed data flow for processing data from wearable device and recognition one of the patterns in human behavior – fall.

Machine learning models were designed and compared in order to select one with the best performance. Different approaches and concepts were implemented to increase the correctness of fall predictions.

While solving the diploma thesis assignment, knowledge of many complex technologies was necessary. There is a list of technologies that were studied and implemented in this work:

1. KNX technology
2. Home Assistant software
3. SQL databases, database connection and extraction
4. Programming of microcontrollers with MicroPython
5. MQTT protocol
6. HTTP protocol
7. Node-Red development environment
8. JavaScript for making function nodes on Node-RED
9. Machine learning concepts
10. Flask framework to build web service
11. AWS deployment procedure
12. Docker container

During the experiment, the HBC system proved its functionality and reliability as well as fall recognition application. However, the fall prediction model still makes false positive predictions and has an insufficient quality for industrial usage. To avoid it, further research with more experiments and advanced preprocessing techniques should be conducted.

The current state of the solution has a status of the minimum viable product and not ideal from the technical point of view. The industrial solution should contain device with smaller dimensions and reduced power consumption as well as extra functionality which will allow competing with other producers on the market.

References

1. AGNIHOTRI, S., P. TIWARI AND M. ZYMBLER Internet of Things is a revolutionary approach for future technology enhancement: a review. Journal of Big Data, 12/01 2019, 6.

2. DEMIRIS, G. AND B. HENSEL Technologies for an Aging Society: A Systematic Review of "Smart Home" Applications. Yearbook of medical informatics, 08/01 2008, 3, 33-40.

3. SKOCIR, P., P. KRIVIC, M. TOMELJAK, M. KUSEK, et al. Activity Detection in Smart Home Environment. Procedia Computer Science, 12/31 2016, 96, 672-681.

4. GORDON, D., J. CZERNY, T. MIYAKI AND M. BEIGL Energy-Efficient Activity Recognition Using Prediction. Edtion ed., 2012. 29-36 p. ISBN 978-1-4673-1583-8.

5. VAŇUŠ, J., O. MAJIDZADEH GORJANI AND P. BILIK Novel Proposal for Prediction of CO2 Course and Occupancy Recognition in Intelligent Buildings within IoT. Energies, 11/28 2019, 12, 4541.

6. APTHORPE, N., D. REISMAN, S. SUNDARESAN, A. NARAYANAN, et al. Spying on the Smart Home: Privacy Attacks and Defenses on Encrypted IoT Traffic 08/16 2017.

7. JATAIN, A. Temperature Monitoring in Home Automation System using Predictive Analytics & IOT. International Journal of Research in Advent Technology, 06/10 2019, 7, 89-96.

8. M. C. Mozer, "Lessons from an Adaptive Home," in Smart Environments. Wiley-Blackwell, 2005, pp. 271–294.

9. Jo, H., Yoon, Y.I. Intelligent smart home energy efficiency model using artificial TensorFlow engine. Hum. Cent. Comput. Inf. Sci. 8, 9 (2018).

10. KRISHNAN, N. AND D. COOK Activity Recognition on Streaming Sensor Data. Pervasive and mobile computing, 02/01 2014, 10, 138-154.

11. MLINAC, M. E. AND M. C. FENG Assessment of Activities of Daily Living, Self-Care, and Independence. Archives of Clinical Neuropsychology, 2016, 31(6), 506-516.

12. VADILLO, L., M. MARTÍN-RUIZ, J. MALAGÓN HERNÁNDEZ, M. VALERO DUBOY, et al. The Role of Smart Homes in Intelligent Homecare and Healthcare Environments. In., 2016, p. 345-394.

13. JAKKULA, V., D. COOK AND G. JAIN Prediction Models for a Smart Home Based Health Care System. Edtion ed., 2007. 761-765 p. ISBN 0-7695-2847-3.

14. Home Assistant. 2020. Main Page. [online] Available at: <https://www.home-assistant.io/> [Accessed 26 November 2020].

15. International Standardization Organization. ISO 16484-2:2004(en): Building automation and control systems (BACS) — Part 2: Hardware. ISO, 2004.

16. FRENKEN, M., J. FLESSNER AND J. HURKA. Building Automation and Control Systems for Healthcare in Smart Homes. In S.U. KHAN, A.Y. ZOMAYA AND A. ABBAS eds. Handbook of Large-Scale Distributed Computing in Smart Healthcare. Cham: Springer International Publi.

17. CLEMENTS-CROOME, D. Intelligent buildings. In., 2004, p. 3-24.

18. Vaňuš, J., 2021. Introduction To Building Automation. [online] Lms.vsb.cz. Available at: <https://lms.vsb.cz/pluginfile.php/1392197/mod_resource/content/1/Introduction%20to%20Building%20Automation.pdf> [Accessed 8 January 2021].

19. Kumar, R., 2021. (92) What Is The Five Layer Automation Pyramid?. [online] Medium. Available at: <https://medium.com/world-of-iot/92-what-is-the-five-layer-automation-pyramid-d0ccc1b903c3> [Accessed 5 January 2021].

20. Youtube.com, 2021, KNX Principles. [online]. 2020. [Accessed 5 January 2021]. Available at: https://www.youtube.com/watch?app=desktop&v=K3FyVoG_X_o.

21. Association., 2020. KNX System Overview. KNX.

22. Library.e.abb.com. 2021. KNX ETS5 And Group Addresses. [online] Available at: <https://library.e.abb.com/public/02d03f979e59478c845af18759dd2030/OnlineTraining-KNX-ETS-Groupaddresses-April-2020_PR_EN_V1-0_9AKK107680A5823.pdf> [Accessed 5 January 2021].

23. https://ru.wikipedia.org/wiki/Интернет_вещей.

24. Cole, Robin. Home Assitant. [Online] [Cited: April 1, 2021.] https://data.home-assistant.io/blog.

25. Who.int. 2020. [online] Available at: <https://www.who.int/ageing/projects/1.Epidemiology of falls in older age.pdf> [Accessed 26 November 2020].

26. Un.org. 2020. [online] Available at: <https://www.un.org/development/desa/pd/sites/www.un.org.development.desa.pd/files/files/documents/2020/Jan/un_2019_worldpopulationageing_report.pdf> [Accessed 26 November 2020].

27. SADIGH, S., A. REIMERS, R. ANDERSSON AND L. LAFLAMME Falls and fall-related injuries among the elderly: a survey of residential-care facilities in a Swedish municipality. J Community Health, Apr 2004, 29(2), 129-140.

28. Euro.who.int. 2020. [online] Available at: <https://www.euro.who.int/__data/assets/pdf_file/0018/74700/E82552.pdf> [Accessed 26 November 2020].

29. November2020], Consumersadvocate.org. 2020. Best Medical Alert Systems Of 2020 | Consumersadvocate.Org. [online] Available at: <https://www.consumersadvocate.org/medical-alerts/> [Accessed 26.

30. YACCHIREMA, D., J. S. DE PUGA, C. PALAU AND M. ESTEVE Fall detection system for elderly people using IoT and Big Data. Procedia Computer Science, 2018/01/01/ 2018, 130, 603-610.

31. HUANG, C.-N. AND C.-T. CHAN A ZigBee-based location-aware fall detection system for improving elderly telecare. International journal of environmental research and public health, 2014, 11(4), 4233-4248.

32. THELLA, A. K., V. K. SURYADEVARA, M. RIZKALLA AND G. HOSSAIN. Smart unit care for pre fall detection and prevention. In 2016 IEEE National Aerospace and Electronics Conference (NAECON) and Ohio Innovation Summit (OIS). 2016, p. 220-224.

33. ABBATE, S., M. AVVENUTI, F. BONATESTA, G. COLA, et al. A smartphone-based fall detection system. Pervasive and Mobile Computing, 2012/12/01/ 2012, 8(6), 883-899.

34. FallDroid: An Automated Smart-Phone-Based Fall Detection System Using Multiple Kernel Learning," in IEEE Transactions on Industrial Informatics, vol. 15, no. 1, pp. 35-44. Kim, A. Shahzad and K.

35. MEHTA, V., A. DHALL, S. PAL AND S. KHAN Motion and Region Aware Adversarial Learning for Fall Detection with Thermal Imaging. Edtion ed., 2020.

36. NÚÑEZ-MARCOS, A., G. AZKUNE AND I. ARGANDA-CARRERAS Vision-Based Fall Detection with Convolutional Neural Networks. Wireless Communications and Mobile Computing, 2017/12/06 2017, 2017, 9474806.

37. OZDEMIR, A. AND B. BARSHAN Detecting Falls with Wearable Sensors Using Machine Learning Techniques. Sensors (Basel, Switzerland), 06/01 2014, 14, 10691-10708.

38. PEÑAFORT-ASTURIANO, C. J., N. SANTIAGO, J. P. NÚÑEZ-MARTÍNEZ, H. PONCE, et al. Challenges in Data Acquisition Systems: Lessons Learned from Fall Detection to Nanosensors*. In 2018 Nanotechnology for Instrumentation and Measurement (NANOfIM). 2018, p. 1-8.

39. Sites.google.com. 2020. HAR-UP. [online] Available at: <https://sites.google.com/up.edu.mx/har-up/página-principal?authuser=0> [Accessed 26 November 2020].

40. Challenges in Data Acquisition Systems: Lessons Learned from Fall Detection to Nanosensors*. C. J. Peñafort-Asturiano, N. Santiago, J. P. Núñez-Martínez, H. Ponce and L. Martínez-Villaseñor. Mexico City, Mexico : Nanotechnology for Instrumentation and Measurement (NANOfIM), 2018.

41. Accelerometer Placement for Posture Recognition and Fall Detection. H. Gjoreski, M. Lustrek and M. Gams. Nottingham, UK : Seventh International Conference on Intelligent Environments, 2011 .

42. Fall-Detection Algorithm Using 3-Axis Acceleration: Combination with Simple Threshold and Hidden Markov Model. Dongha Lim, Chulho Park, Nam Ho Kim, Sang-Hoon Kim, Yun Seop Yu. s.l. : Journal of Applied Mathematics, 2014.

43. Se.com. 2021. Products Overview | Schneider Electric. [online] Available at: <https://www.se.com/ww/en/all-products> [Accessed 5 January 2021].

44. M. C. Mozer, "Lessons from an Adaptive Home," in Smart Environments. Wiley-Blackwell, 2005, pp. 271–294.

List of Appendices

Appendix A: Building structure report

(Report with building structure of the project generated in ETS5)

Address / Description / Comments / Installation Notes	Manufacturer	Order Number	Product	Application	Status
SHC1					
Ground floor					
Bathroom					
4.1.33	Schneider Electric Industries SAS	MTN6172xx	Push-button 2-gang plus	Universal 1815/1.1	
4.1.38	Schneider Electric Industries SAS	MTN6309xx	KNX ARGUS Presence with light control + IR	Presence/monitoring/light control/IR 1335/1.0	
Bedroom					
4.1.32	Schneider Electric Industries SAS	MTN6172xx	Push-button 2-gang plus	Universal 1815/1.1	
Hallway					
4.1.1	Schneider Electric Industries SAS	MTN6309xx	KNX ARGUS Presence with light control + IR	Presence/monitoring/light control/IR 1335/1.0	
4.1.8	Schneider Electric Industries SAS	MTN6172xx	Push-button 2-gang plus	Universal 1815/1.1	
Cabinet					
4.0.-	Schneider Electric Industries SAS	MTN684016	KNX power supply REG-K/160 mA		
4.0.0	Schneider Electric Industries SAS	MTN680329	KNX / IP-Router REG-K	KNX / IP-Router 7125/1.0	
4.0.11	Schneider Electric Industries SAS	MTN646991	Control unit 0-10 V REG-K/3f with manual mode	Universal dimming 3211/1.1	
4.0.100	Schneider Electric Industries SAS	MTN681829	USB interface REG-K		
4.1.10	Schneider Electric Industries SAS	MTN649908	Blind/Switch actuator REG-K/8x/16x/10 manual mode	Blind.Switch 5701/1.0	
4.1.12	Schneider Electric Industries SAS	MTN647595	Switch actuator REG-K/4x230/16	Switch.Logic.Currentdet.PWM 4806/1.1	
Living room					
4.1.30	Schneider Electric Industries SAS	MTN6171xx	Push-button 1-gang plus	Universal 1815/1.1	
4.1.31	Schneider Electric Industries SAS	MTN6172xx	Push-button 2-gang plus	Universal 1815/1.1	
4.1.35	Schneider Electric Industries SAS	MTN6005-0001	KNX CO2, Humidity and Temperature Sensor	CO2-,Humidity-&Temp.Sensor 4216/1.2	

Appendix B: Topology report

(Report with topology structure of the project generated in ETS5)

Address / Description / Room / Comments / Installation Notes	Manufacturer	Order Number	Product / Trade	Application	Status
0	Backbone area				
0.0	IP	Backbone line			
4	New area				
4.0	TP	Main line			
4 devices in line					
4.0.- Cabinet	Schneider Electric Industries SAS	MTN684016	KNX power supply REG-K/160 mA		
4.0.0 Cabinet	Schneider Electric Industries SAS	MTN680329	KNX / IP-Router REG-K	KNX / IP-Router 7125/1.0	
4.0.11 Cabinet	Schneider Electric Industries SAS	MTN646991	Control unit 0-10 V REG-K/3f with manual mode	Universal dimming 3211/1.1	
4.0.100 Cabinet	Schneider Electric Industries SAS	MTN681829	USB interface REG-K		
4.1	TP	New line			
10 devices in line					
4.1.1 Hallway	Schneider Electric Industries SAS	MTN6309xx	KNX ARGUS Presence with light control + IR	Presence/monitoring/light control/IR 1335/1.0	
4.1.8 Hallway	Schneider Electric Industries SAS	MTN6172xx	Push-button 2-gang plus	Universal 1815/1.1	
4.1.10 Cabinet	Schneider Electric Industries SAS	MTN649908	Blind/Switch actuator REG-K/8x/16x/10 manual mode	Blind.Switch 5701/1.0	
4.1.12 Cabinet	Schneider Electric Industries SAS	MTN647595	Switch actuator REG-K/4x230/16	Switch.Logic.Currentdet.PWM 4806/1.1	
4.1.30 Living room	Schneider Electric Industries SAS	MTN6171xx	Push-button 1-gang plus	Universal 1815/1.1	
4.1.31 Living room	Schneider Electric Industries SAS	MTN6172xx	Push-button 2-gang plus	Universal 1815/1.1	
4.1.32 Bedroom	Schneider Electric Industries SAS	MTN6172xx	Push-button 2-gang plus	Universal 1815/1.1	
4.1.33 Bathroom	Schneider Electric Industries SAS	MTN6172xx	Push-button 2-gang plus	Universal 1815/1.1	

Address Description Room Comments Installation Notes	Manufacturer	Order Number	Product Trade	Application	Status
4	New area				
4.1	TP	New line			
4.1.35 Living room	Schneider Electric Industries SAS	MTN6005-0001	KNX CO2, Humidity and Temperature Sensor	CO2-,Humidity-&Temp.Sensor 4216/1.2	
4.1.38 Bathroom	Schneider Electric Industries SAS	MTN6309xx	KNX ARGUS Presence with light control + IR	Presence/monitoring/light control/IR 1335/1.0	

Appendix C: Description of devices

(Description of KNX devices used in the design of building control application)

Push buttons MTN617125 and MTN617225

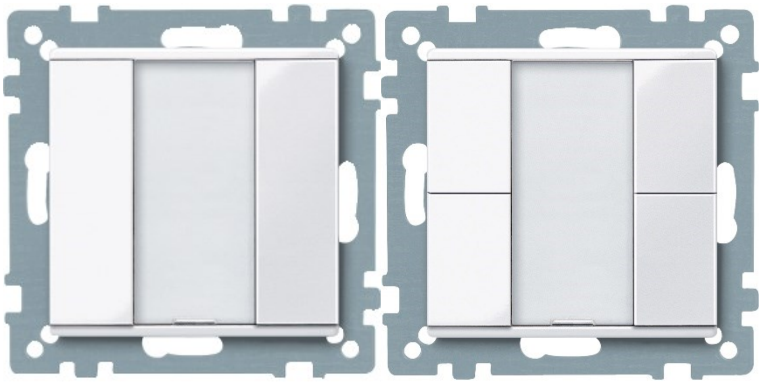

Figure 43 – Push buttons SE (43)

The KNX push-button provides you with two, four or eight operating surfaces, two in the case of 1-gang pushbuttons, four in the case of 2-gang push-buttons, and eight in the case of 4-gang push-buttons. The push buttons can be set to perform various functions, allowing you, for example, to switch lighting on and off or dim it, control the blinds or retrieve stored scenes.

KNX CO2, humidity and temperature sensor MTN6005-0001

Figure 44 – KNX CO2, humidity and temperature sensor SE (43)

The sensor serves to detect carbon dioxide (CO2), relative humidity and temperature in a variety of rooms (offices, schools, meeting rooms etc).

- Temperature detection range: 0–40 °C –
- Setting range thresholds: 500–2550 ppm –
- “Relative humidity” detection range: 1 %–100 %

Presence detector ARGUS MTN630919

Figure 45 – Presence detector ARGUS SE (43)

The KNX ARGUS Presence with light control and IR receiver (called ARGUS in the following) is a KNX presence detector for interior ceiling mounting. It detects smaller movements within a circumference of 360° and a radius of 7 m (at a mounting height of 2.5 m). When movement is detected, a data telegram defined by the programming is transmitted and then evaluated to control the lighting, blinds, or heating, for example.

Blind/Switch actuator REGK 8x MTN649908

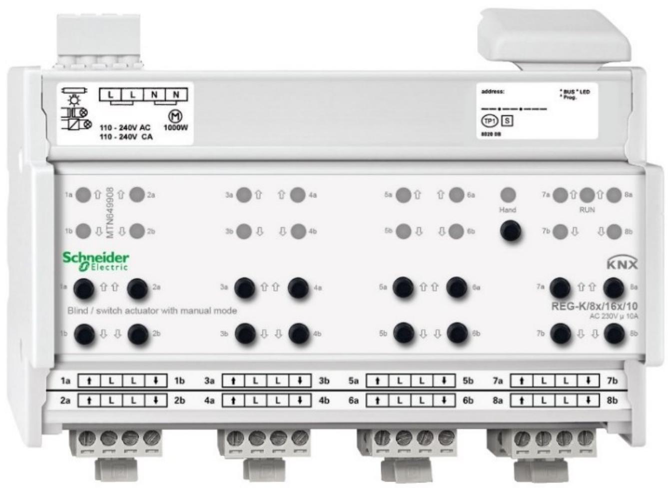

Figure 46 – Blind/Switch actuator SE (43)

The blind/switch actuator REG-K/8x/16x/10 with manual mode can:

- control blind/roller shutter drives with end position switch independently of one another,
- switch loads (luminaires) via separate, floating make contacts.

You can operate each channel either as a blind channel or as two switching channels. In the "Switching" channel operating mode, one blind channel is divided into two equal switching channels. You must change over the channels at the device itself and in the ETS. You can switch the actuator to manual operation and check that it works even without ETS programming, and you can control the connected loads directly at the actuator in the event of a bus failure.

Switch actuator REG-K/4x230/16 MTN647595

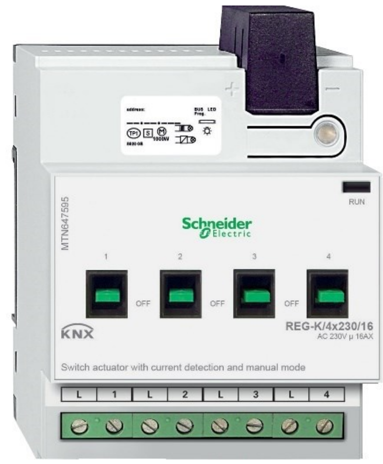

Figure 47 – Switch actuator SE (43)

The switch actuator REG-K/4x230/16 with current detection and manual mode can switch four loads via separate, floating make contacts. You can also manually switch the connected loads with manual switches on the actuator without bus voltage. The actuator has a bus coupler. It is installed on a DIN rail (DIN 60715), with the bus connection made via a bus connecting terminal. It is supplied with power from the bus voltage.

Control unit 0-10 V MTN646991

Figure 48 – Dimming actuator SE (43)

The control unit 0-10 V REG-K/3-gang with manual mode dims and switches fluorescent lamps using electronic ballasts with a 0-10 V/1-10 V interface and LV halogen lamps using transformers with a 0-

10 V/1-10 V interface. The control voltage range can be set with the ETS via a parameter. You can also manually switch the connected loads with the manual switches on the control unit without bus voltage.

KNX/IP router REG-K MTN680329

Figure 49 – KNX/IP router SE (43)

The KNX/IP router REG-K enables KNX telegrams of a TP line to be forwarded to a LAN (IP) as a rapid backbone. The KNX telegrams are forwarded in both directions. In a KNX system, you can use KNX/IP routers, couplers or both devices in mixed operation. The KNX/IP router can also be used as an interface for bus access via IP (e.g. for ETS programming).

USB interface REG-K MTN681829

Figure 50 – USB interface SE (43)

The USB interface REG-K allows linking to a PC for addressing, programming and diagnosis of bus components with USB1.1 or USB2 interfaces

Appendix D: Group Addresses report

(Report with group addresses structure of the project generated in ETS5)

Address Description Comments	Name	Length	Central	Pass through Line Coupler
0	SHC1			No
0/0	Light			No
0/0/1	Bedroom Light on/off	switch	No	No
0/0/2	Bedroom Light dimming	4 bit	No	No
0/0/3	Bedroom Light brightness	1 byte	No	No
0/0/4	Bathroom Light on/off	1 bit	No	No
0/0/5	Bathroom Light locking	1 bit	No	No
0/0/6	Kitchen zone Light on/off	1 bit	No	No
0/0/7	Kitchen zone Light dimming	4 bit	No	No
0/0/8	Kitchen zone Light brightness	1 byte	No	No
0/0/9	Hallway Light on/off	switch	No	No
0/0/10	Hallway Light locking	1 bit	No	No
0/0/11	Living room main Light on/off	1 bit	No	No
0/0/12	Living room main Light dimming	4 bit	No	No
0/0/13	Living room Light brightness	1 byte	No	No
0/1	Blinds			No
0/1/0	Bedroom Blind movement	1 bit	No	No
0/1/1	Bedroom Blind stop/step	1 bit	No	No
0/1/2	Bedroom Blind position	1 byte	No	No
0/1/3	Living room Blind stop/step	1 bit	No	No
0/1/4	Living room Blind movement	1 bit	No	No
0/1/5	Living room Blind position		No	No
0/2	Sensors			No
0/2/0	Living room Temperature	temperature (°C)	No	No
0/2/1	Living room Humidity	1 byte	No	No
0/2/2	Living room CO2	2 bytes	No	No
0/3	Switches			No
0/3/0	Door lock	1 bit	No	No

Appendix E: KNX configuration in Home Assistant

(Configuration file for connection KNX and Home Assistant)

```
###########################
#KNX IP interface
###########################
knx:
  tunneling:
    host: '192.168.18.50'
    #port: 3671

###########################
# LIGHTS
###########################

  light:
    - name: 'Bathroom light'
      address: '0/0/4'

    - name: 'Hallway light'
      address: '0/0/9'

    - name: 'Bathroom light locking'
      address: '0/0/5'

    - name: 'Hallway light locking'
      address: '0/0/10'

###########################
# DIMMING LIGHT
###########################

    - name: 'Bedroom light'
      address: '0/0/1'
      brightness_address: '0/0/3'

    - name: 'Living room light'
      address: '0/0/11'
      brightness_address: '0/0/13'

    - name: 'Kitchen zone light'
      address: '0/0/6'
      brightness_address: '0/0/8'

###########################
# BLINDS
###########################
  cover:
    - name: "Bedroom blinds"
      move_long_address: "0/1/0"
      stop_address: "0/1/1"
      position_address: "0/1/2"

    - name: "Living blinds"
      move_long_address: "0/1/3"
      stop_address: "0/1/4"
      position_address: "0/1/5"

###########################
# SENSOR
###########################
  sensor:
    - name: 'Temperature'
      state_address: '0/2/0'
      type: 'temperature'

    - name: 'Humidity'
      state_address: '0/2/1'
      type: 'percent'

    - name: 'CO2'
      state_address: '0/2/2'
      type: 'ppm'

###########################
# ACTUATORS
###########################
  switch:
    - name: 'Door lock'
      address: "0/3/0"
```

Appendix F: Control program for wearable device

(Control program for M5Stick device to transfer data over MQTT)

```
from m5stack import *
from m5ui import *
from m5mqtt import M5mqtt
import imu
import json
import hat
setScreenColor(0x5fff86)
label0 = M5TextBox(30, 139, "", lcd.FONT_Ubuntu, 0x54f24a, rotate=270)
hat_spk0 = hat.get(hat.SPEAKER)
imu0 = imu.IMU()
m5mqtt = M5mqtt('M5Stick', '192.168.137.82', 1883, 'SHC1', '1111', 300)
fall_prediction = 0
def func_fall_prediction(topic_data):
  global fall_prediction
  fall_prediction = int(topic_data)
  pass

m5mqtt.subscribe(str('home-assistant/fall_prediction'), func_fall_prediction)
m5mqtt.start()
m5mqtt.publish(str('home-assistant/fall_prediction'), str(0))
while True:
    data = [imu0.acceleration[0], imu0.acceleration[1], imu0.acceleration[2],
            imu0.gyro[0], imu0.gyro[1], imu0.gyro[2]]
    m5mqtt.publish(str('home-assistant/data'), json.dumps(data))
    if fall_prediction == 1:
      setScreenColor(0xd26565)
      label0.setColor(0x54f24a)
      label0.setText("<- Press button")
      hat_spk0.setVolume(40)
      for i in range(6):
        hat_spk0.tone(900, 200)
        if btnA.wasPressed():
          fall_prediction = 0
        wait_ms(500)
      if fall_prediction == 1:
        m5mqtt.publish(str('home-assistant/fall_event'), str(1))
        fall_prediction = 0
      else:
        m5mqtt.publish(str('home-assistant/fall_event'), str(0))
      setScreenColor(0x5fff86)
    wait_ms(50)
```

Appendix G: Web application with Machine learning model

(Source code for fall recognition web application written in Python)

```python
import pickle
import numpy as np
import pandas as pd
from flask import Flask, request

model = None
scaler = None
columns_features = []
df1 = pd.DataFrame(columns=['a_x', 'a_y', 'a_z', 'g_x', 'g_y', 'g_z'])
app = Flask(__name__)

def load_model():
    global model, scaler
    # model variable refers to the global variable
    with open('Model dev/XGB_model.pkl', 'rb') as f:
        model = pickle.load(f)
        pass
    with open('Model dev/scaler.pkl', 'rb') as f:
        scaler = pickle.load(f)
        pass

def format_data(s):
    return (list(map(lambda x: float(x) * 8 / 2, s.strip('[]').split(',
')[:3])) +
            list(map(lambda x: float(x), s.strip('[]').split(', ')[3:])))

def zero_crossing(arr):
    zero_index = np.where(np.diff(np.sign(arr)))[0]
    return len(zero_index)

def SV(df):
    return df.apply(lambda x: np.sqrt(x.a_x**2 + x.a_y**2 + x.a_z**2),
axis=1).to_list()


def columns_names():
    global columns_features
    selected_features = ['Mean', 'StandardDeviation', 'MaximalAmplitude',
                         'MinimalAmplitude', 'Number of zero-crossing']
    # Select columns
    for m, u in [('WristAccelerometer', '(g)'), ('WristAngularVelocity',
'(deg/s)')]:
        for ax in ['x', 'y', 'z']:
            for metric in selected_features:
                columns_features.append('{}: {}-axis {}{}'.format(m, ax, u,
metric))

def make_features(df):
    global columns_features
    pred = []
    for column in df.columns:
        arr = df[column].tolist()

        pred.append(np.mean(arr))
        pred.append(np.std(arr))
        pred.append(np.amax(arr))
        pred.append(np.amin(arr))
        pred.append(zero_crossing(arr))

        SumVector = SV(df)
        pred.append(SumVector)
```

```
        SV_MaxMin = max(pred[-1]) - min(pred[-1])
        pred.append(SV_MaxMin)

    pred = list(map(lambda x: round(x, 2), pred))
    return pd.DataFrame([pred], columns=columns_features + ['SV',
'SV_MaxMin'])

@app.route('/')
def home_endpoint():
    return 'Application for Fall recognition'

@app.route('/predict', methods=['POST'])
def get_prediction():
    global df1, model, scaler

    if request.method == 'POST':
        content = request.get_json()  # Get data posted as a json
        data = content['key'].split('\n')
        data = list(map(format_data, data))

        data_scaled = scaler.transform(pd.DataFrame(data))

        df2 = pd.DataFrame(data, columns=['a_x', 'a_y', 'a_z', 'g_x',
'g_y', 'g_z'])

        window_1 = df1
        window_2 = df1.iloc[5:].append(df2.iloc[:5])

        try:
            prediction1 = model.predict(make_features(window_1))[0]
            pass
        except:
            prediction1 = 0

        prediction2 = model.predict(make_features(window_2))[0]

        df1 = df2.copy()


        if prediction1 or prediction2:
            return str(1)
        else:
            return str(0)

if __name__ == '__main__':
    load_model()  # load model at the beginning once only
    columns_names()
    app.run(host='192.168.137.1', debug=True, port=80)
```