Appendix 1
Pielikums 1

National Research Programme

„Cyber-physical systems, ontologies and biophotonics for safe&smart city and society"

(SOPHIS)

Project No.1

Development of technologies for cyber physical systems with applications in medicine and smart transport"

(KiFiS)

Scientific report

Period 1 & 2

2016

Contents

Glossary and abbreviations

ADAS - Advanced Driver Assistance Systems;

CPS - Cyber-physical systems;

ITS - Intelligent transport systems;

ECG - Electrocardiogram;

EDI - Institute of Electronics and Computer Science, Riga, Latvia;

EMG - Electromyogram;

KiFiS - Cyber-physical system technology development and their applications in medicine and intelligent transport systems (Project No. 1);

SoC - System on Chip;

SOPHIS - Cyber-physical systems, ontologies, and bio-photonics for safe&smart city and society;

VPP - State Research Programme;

Chapter 1.1

Introduction

State research program "Cyber-physical systems, ontologies, and bio-photonics for safe&smart city and society" (VPP SOPHIS) and the included project No. 1 "Cyber-physical system technology development and their applications in medicine and intelligent transport systems" (KiFiS) include tasks for the development of new generation of embedded systems – cyber-physical systems (CPS).

Cyber-physical systems include communication, data processing, and control elements, as well as interfaces to the physical world. These systems monitor the processes in real world, process the data, decides on the actions of controlling and improving the situation and enacts these decisions in the physical environment. Such cycles happen endlessly, and both on low level (such as a single room) and high level (such as a smart city). CPS provide a way for solving the economic problems, by providing us with "smarter", more intelligent, more energy efficient, more comfortable vehicles and transport systems, medical services, places of employment, communication systems, houses, cities and personal devices.

To make this vision a reality, there is a range of serious scientific and technological problems, that still need to be solved, connected to data gathering, electrical and optical signal processing, monitoring, control functions, while at the same time providing high enough level of security, stability and privacy. In addition, the system must be low energy, small, mobile and adaptable to new circumstances, as well as oriented to development of user friendly software and its usability. Scientific problems are connected to defining of new paradigms, concepts, platforms (hardware and software) and tool sets for the future development of CPS.

Because of limited resources available in this research project a subset of three specific CPS-related research areas were selected, matching the overall goals of the state research programme, and assigned to three groups of researchers:

- Group *TestBed*: To facilitate the production, programming and usage of CPS and by doing so, also to facilitate development of economically competitive innovative CPS based products while also facilitating their everyday use and reducing the digital divide;

- Group *MedWear*: To improve the quality and convenience of medical services, while facilitating more efficient prophylaxis, more timely diagnostics

and more successful treatment and rehabilitation based on innovative solutions both face to face or remotely in telemedicine;

- Group *SmartCar*: To improve the road safety and convenience of using road vehicles, by the use of smart transport system technologies.

In each of these three groups new concepts and platforms are developed based on results of previous state research projects, and improving on existing state of the art. These concepts and platforms are evaluated by comprehensive modeling and simulation research, thus selecting the perspective solutions, which are researched empirically, by creating experimental mock-ups, conceptual demonstrators, software libraries. The technologies which are economically competitive, will be approbated ir real or close to real conditions, in cooperation with partners from the economy.

The following document contains detailed description of these research activities in each of the three groups -

- *TestBed* in section 1.2 describes a testing/prototyping environment for wireless sensor system development and testing;

- *MedWear* in section 1.3 describes smart wearable sensor network infrastructure for energy efficient data gathering for measuring human biomechanics, and other health parameters (such as ECG), with applications in medicine and rehabilitation;

- *SmartCar* in section 1.4 describes development of Advanced Driver Assistance Systems based on smart image and sensor signal processing, and communication with other vehicles as well as development of a car based test platform for these systems and related algorithms.

Chapter 1.2

TestBed - Smart sensor and their network innovative hardware and software platform

1.2.1 Introduction

To ease the production, programming and usage of CPS, thus promoting competitive production of innovative CPS based products in economy, as well as facilitating their everyday usage and bridging the digital divide a set of research was done in the field of sensors, sensor networks and their hardware and software platforms.

Designing Wireless Sensor Networks (WSN) is time consuming process that involves many subsequent steps:

1. definition of WSN use case,
2. design and debugging of hardware,
3. development and debugging of software,
4. evaluation of designed WSN performance,
5. adoption of WSN for real-world operation.

Design of WSN begins with definition of use case – number of sensor nodes, operating environment and desired up-time. There are two possibilities for hardware design (step 2): first – to use commercially available sensor nodes TelosB[1], MicaZ[2] EPIC Mote[3], XM1000[4], second – to build hardware from ground. The first option requires only adaptation of existing hardware to the desired operation and therefore requires less effort compared to the second option. However, by using the second option it is possible to design hardware that is optimized for

specific task and has no redundant components. Sometimes software development (step 3) may reveal that some changes to the hardware or even whole architecture of developed node are necessary or beneficial (to increase performance or reduce software complexity). Therefore, steps 2 and 3 must be iteratively repeated. In step 4 designed WSN is tested in controlled environment to evaluate power consumption, radio communication performance, and other parameters. In step 5 designed WSN is scaled (for operation using 25+ nodes) and/or adapted to real-world operation. Failure to accomplish step 5 may require repetition of previous steps and redesign of WSN.

Without specialized tools efficient execution of mentioned steps can be very challenging. For instance, there are a lot of routine manipulations that slow down the design process, like mounting, reprogramming of sensor nodes and connection of measurement equipment. To decrease development time of WSN, mentioned steps must be simplified.

This project works on solving these problems from two directions - (1) providing tools and know-how for faster development of WSN hardware and software, (2) reducing the time and effort required for testing and validating each of the mentioned steps.

The first direction is concerned with both low level tools such as Operating System for WSN (MansOS)[5], and high level programming tools (Seal/Blockly)[6]. Most research in these tasks is left for periods 3 and 4 of this project, while concentrating the limited resources of periods 1 and 2 (described in this report) on the second direction - testing and validation.

In this direction several new hardware and software solutions were developed, and used in development of a large (25+) WSN TestBed where users can perform different tests at different levels of abstraction - from low to high.

The WSN TestBed design challenges can be divided into three main subcategories - architectural, hardware and software:

1. Architectural problems - Scaling, upgrading, and adding a new custom hardware;

2. Hardware problems - Selected hardware define overall WSN TestBed performance;

3. Software problems - The most efficient way to use available hardware resources for desired functionality. User-friendly front-end implementation for intuitive TestBed usage, without compromising data acquisition, processing, structuring from TestBed user point of view

Because of the limited resources of the project, the scope of the research has been narrowed to work on specific unsolved parts of these research problems, while reusing the existing state-of-the-art knowledge and results from previous projects for other parts of the testing environment surrounding them.

In this section our specific approach to solving these problems is described, as well as the vision for the system as a whole, defining the direction of the future research in the project.

1.2.2 Background

Wireless Sensor Networks (WSN) are broadly used in different types of applications, from agriculture to medicine and on body sensor networks. Essence of WSN is to observe the surrounding environment parameters at macroscopic level. For example WSN can be used to monitor temperature distribution in building or vibration levels at the bridge. The quality of designed WSN is defined by individual autonomous devices, called sensor nodes or simply motes, performance, WSN covered area and count of placed nodes.

Typically WSN consists of sensor nodes, that communicates between each other using radio link. Each mote has specific sensor set that is necessary to measure desired environmental parameters. Acquired data are gathered from sensor nodes to super node or sink. Typically super node is connected to the local or global network and also is used as bridge to provide easy data access for end-users.

1.2.3 Related work

There are many designed TestBeds for WSN. We will review most popular WSN TestBeds.

1.2.3.1 The TKN Wireless Indoor Sensor network TestBed (TWIST)

This TestBed is developed by the Telecommunication Networks Group (TKN) at the Technische Universität Berlin. It is one of the first largest academic WSN TestBed[7] for indoor deployment scenarios, deployed in 2005 year. It is located across 3 floors, resulting in more than 1500 m2 of instrumented office space. Currently they are using two types of sensor nodes - 102 Tmote Sky[8], 102 eyesIFX[9].

TWIST TestBed architecture is hierarchical in nature, consisting of three different levels of deployment: sensor nodes, micro-servers, central server. A high level view of this architecture can be seen in Fig. 1.2.1 below.

1.2.3.2 MoteLab

MoteLab[10] has been deployed on a network of 30 Ethernet-connected MicaZ [2] sensor nodes distributed over three floors of Maxwell Dworkin, the Electrical Engineering and Computer Science building at Harvard University. Also this WSN TestBed is freely available as open source, and several universities and research labs have chosen to use it for their projects.

MoteLab was the first designed WSN TestBed with reduced usage complexity. This was achieved due to the fact that there was implemented a rich-set of features: user-friendly web interface, remote access, automatic data logging for offline data processing, job scheduling, quota system for fairly TestBed usage.

MoteLab consists of several different software components. The main pieces are:

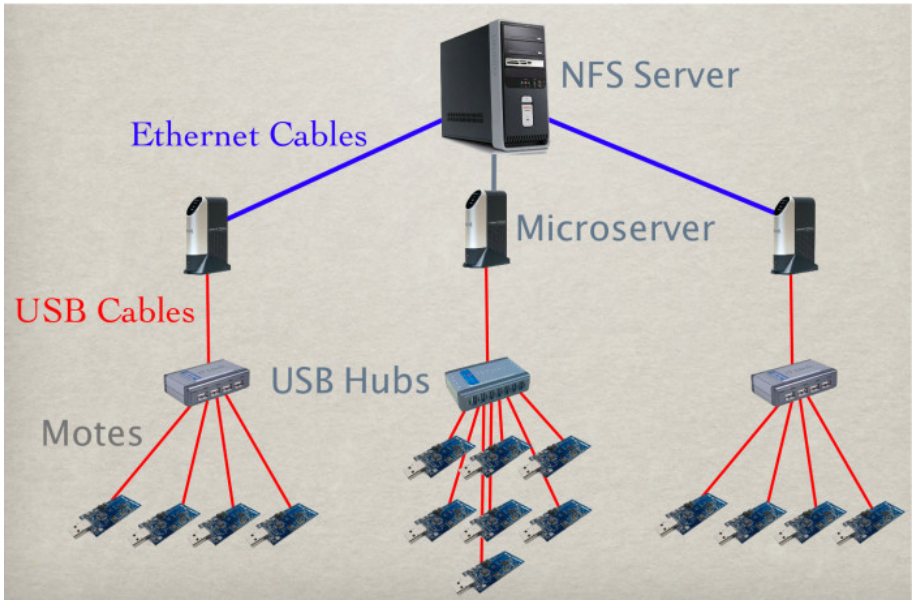


Figure 1.2.1: TWIST TestBed architecture

- **MySQL Database Backend** : Stores data collected during experiments, information used to generate web content, and state driven TestBed operation description,
- **Web Interface** : PHP-generated pages present a user interface for job creation, scheduling, and data collection, as well as an administrative interface to certain TestBed control functionality,
- **DBLogger** : Java data logger to collect and parse data generated by jobs running on the lab,
- **Job Daemon** : Perl script run as a cron job to setup and tear down jobs.

1.2.3.3 Indriya: A Low-Cost, 3D Wireless Sensor Network TestBed

INDRIYA is a three-dimensional wireless sensor network deployed across three floors of the School of Computing, at the National University of Singapore[11]. 100 TelosB[1] nodes and 25 Arduino[12] devices are used in INDRIYA TestBed. The INDRIYA WSN TestBed is build on TWIST architecture[7] with modifications regarding cost reduction. To reduce system cost INDRIYA uses MAC Mini devices that is capable of controlling 127 USB like sensor nodes. In this way micro-server count is reduced, thus cost are reduced.

1.2.4 Our approach

After analyzing the state-of-the-art WSN TestBeds it was decided that there are still many unsolved problems in this field and to properly develop and test

potential solutions to these problems it was decided to develop a TestBed of our own, capable of implementing our research results. TWIST[7] was chosen as the ground truth architecture for the TestBed, but two major modifications were implemented:

1. Ethernet switches were replaced with PoE switches. PoE switch supports data transfers and power delivery. There are two PoE IEEE standards: 802.3af, max power rating is 15.4W, second - 802.3at, max power rating 25.5W. This modification allows us to decrease set-up costs and place micro-servers more freely in desired places, but we can't exceed power limitations, thus power efficient micro servers must be used.

2. Additional module, EDI TestBed adapter (described in subsection 1.2.4.5), is introduced. It is placed between micro-server and sensor node. Our developed module allows users to accurately evaluate designed WSN performance. It provides additional information like: power consumption measurement, battery discharging emulation, real-world sensor data emulation, analog/digital signal debugging.

In the following sections a more detailed view of the developed TestBed is provided, including description of architecture and solutions to specific researched problems, including work on the main challenges:

1. Distribution of the computational resources: The challenge is to split computational power to use full hardware potential, thus reducing computational load from main server;

2. Control all of the TestBed devices: The challenge is to reprogram all sensor nodes (DUT) as well as TestBed adapters remotely and at the same time. This also includes network health monitoring;

3. Efficient data acquisition and structuring: The challenge is to effectively acquire data and structure it in user-friendly manner.

1.2.4.1 TestBed architecture

The architecture of the TestBed (Figure 1.2.2) consists of the server (section 1.2.4.3), routers (section 1.2.4.4), TestBed adapters (section 1.2.4.5) and devices under test (DUT) (section 1.2.4.2).

Routers are connected to the server via Ethernet with PoE. To each of the routers a TestBed adapter is connected via USB connection and to finally the DUT is connected to the adapter either via USB connection (if it supports it) or some other configuration specific wiring.

Physically the TestBed will be installed in EDI building, across five floors - first, second, third, fourth, seventh. About 20 TestBed workstations will be placed in each floor. The placement grid was defined as irregular. Such sensor node placement assures different environment for testing radio communications and a more realistic close-to-real-world testing environment with different types

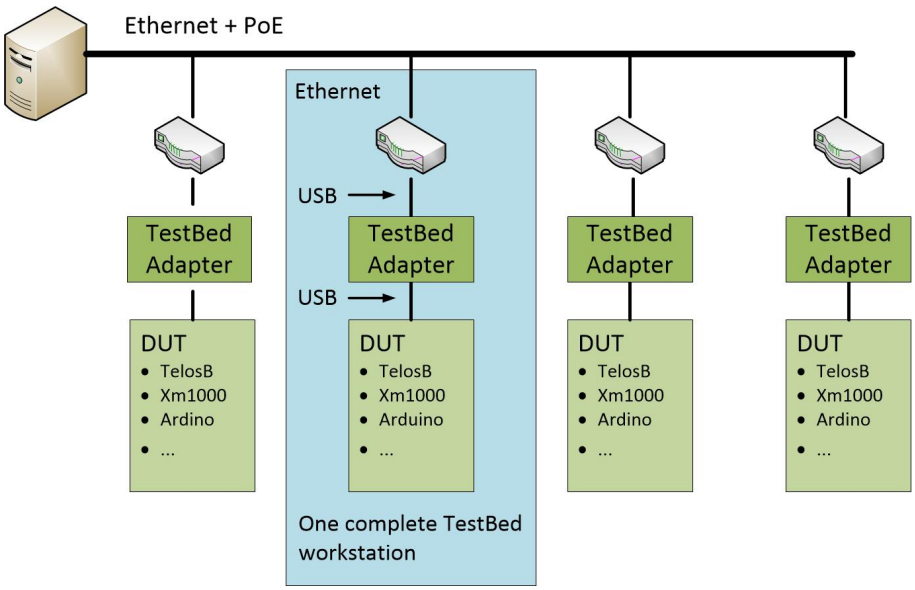


Figure 1.2.2: EDI TestBed Architecture

of rooms, people density and obstacles. It is intended that 100 TestBed workstations will be installed, 90 of them across five floors, 10 outside of The EDI building utilizing also the nearby forest and exterior structures for even more varied test environment. In figure below you can see TestBed workstation placement in the third floor, Fig. 1.2.3.

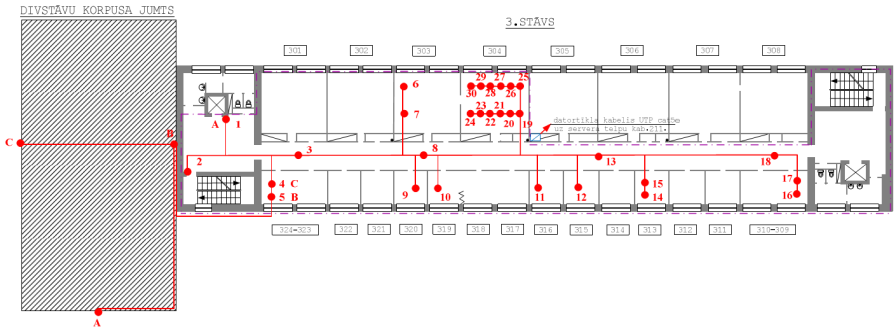


Figure 1.2.3: Sensor node placement in the 3rd floor

At the end of the second period of the project most of the TestBed workstations are already complete and are in the process of being deployed to their planned locations.

1.2.4.2 TestBed Devices Under Test (DUT)

We have designed our WSN TestBed in a way that it puts as few restrictions as possible on devices which can be tested. Devices Under Test can be operated under any WSN operating system that the device is capable of running. As

for the device itself, we have 100 XM1000[4] sensor nodes which we will use as default sensor nodes, but theoretically any device can be used as long as it can operate from USB power source and use Serial communication through USB, the only restriction being that device must be compatible with reprogramming scripts used in routers, but this can be quite flexible since routers are running Linux operating system and we can add new reprogramming scripts quite easy.

Although it is possible to run any operating system on sensor nodes, we recommend using MansOS because we are planning to develop TestBed with close integration into MansOS operating system for more convenient and faster sensor node software prototyping. Also we are planning several hardware and software monitoring features integrated into MansOS.

1.2.4.3 TestBed server

The physical parameters of the server are:

- Manufacturer: ATEA
- Model: sVectron TS26
- CPU: Intel(R) Xeon(R) CPU E3-1220 V2
- RAM: DDR3-1333 4Gb,
- HDD: 1TB, 7200RPM , 64MB cache, SATA3,
- Case: Rackmount type (height 1U).

The server acts as a link between the TestBed and its users as well as a centralized data storage and configuration platform. It provides several critical services, such as:

- Centralized configuration services;
- Web interface for the TestBed;
- Centralized reprogramming control for the connected DUT;
- Online code editor for DUT;
- Data gathering, visualization and analysis utilities.

Centralized configuration To provide the same Linux configuration for all routers, we set specific configuration on the server to provide netboot. Each router before boot-up gets Linux kernel and file system from the server. Configuration was set on both server and router, server is providing and supporting functionality of netboot, but router is set to load boot image from network first. First configuration step on the server is to install DHCP server and manage the IP-addresses, second step is to prepare boot files and set specific protocol for sending files to router. In our case we used DNSmasq server to provide boot files and define MAC address list. TFTDI protocol is used to send boot images to routers. Configuration for both of the softwares is complex and includes multiple sub steps to deal with.

Web interface Web interface backend was programmed in Python using Django 1.8[13] framework. Python was chosen because MansOS operating system, that is used in almost all levels of software in TestBed design, is programmed in Python and it is easier to interact with it. Django framework was chosen because of its wide range of security and customization features. The interface is built as an interactive web application, exchanging data with server through AJAX calls, providing smooth uninterrupted user experience.

Centralized reprogramming One of most time consuming operations in manual sensor network testing is sensor node reprogramming, that is why every wireless sensor TestBed addresses this problem. To reprogram sensor nodes in EDI TestBed interface user first needs to upload .ihex file(compiled code) to server which can later be uploaded on sensor node. In present software state code compiling on server through web interface is not possible, but it is planned for next periods of the project. When compiled code files are uploaded they are saved on the server and can be used by user to reprogram sensor nodes at any time. To start node reprogramming user must choose which file to use for reprogramming and choose group of nodes that should be programmed. Group reprogramming design was developed because most of wireless sensor network designs consist of main tower and leaf nodes or main tower, lower level towers, and leaf nodes and every group of them have different program. Using this interface user can choose group of nodes for every program needed and reprogram them all together. Uploading interface is made to save as much time as possible in setting up the experiment and to be as easy to use as possible.

Online code editor To improve the programming and testing experience in the TestBed, an online code editor is also being developed, which will allow skipping the local compilation and uploading of the code, in favor of online code editing and compiling directly to DUT.

To tackle this task a fast and flexible way to highlight code was needed. ACE[14] JavaScript library was used as basic building block for this part of the system. It was chosen because of its large support of different languages, custom themes, already integrated console view for easy debugging and even primitive auto-complete function.

Currently C and make file syntax highlighting is supported for TestBed editor, also code compiling is possible trough web interface, while debugging and easy one-click upload is planned for the next periods of the project.

Data visualization Data from DUT serial output is sent via the TestBed to the server for storage, analysis and display to the user. To make monitoring of sensor node output easier and less time consuming, two different views for visualizing data were developed.

First view is **plain text view** in which data can be viewed by prefix in plain text. It is useful for looking at string or hex data and is a basic necessity for easy debugging. This visualizing mode is also comfortable for monitoring specific

events outputted by sensor node. For example - time of dawn or low battery warning.

Second is **graphical view** in which data is visualized using Google Charts API[15]. As this API contains a lot of built in data representation styles data can be visualized in most convenient way possible. For example, simple scalar data can be viewed as line graph, vector data can be viewed in plane etc. Currently the graphical interface only fully supports line graphs. Work on supporting pie charts, for displaying relations between values or relative differences, plane charts of visualizing vectors and graph charts for visualizing networks or graphs is ongoing. We also plan to develop timeline graphic to allow convenient monitoring of specific event data.

Restructuring of experiment entity In current development state all data gathered in experiment is saved to database and can be accessed any time after experiment, but it is not marked by experiment ID. In case of couple experiments it is easy to find necessary data, but if saved data size is greater, it is hard to filter out data from necessary experiment. To address this issue a new way of experiment setup was started to be developed. When experiment is being created user is prompted to enter experiment name (using which user can access all data about it), TestBed adapter configuration(necessary modules, module rate of measurements etc.), router configuration(necessary pre-processing etc.), involved DUT and their respective programs. This way greater control over TestBed hardware can be achieved and all experiment data is saved in a structured manner, easy to look up, including all data gathered side by side with configuration settings.

1.2.4.4 TestBed routers

Currently the routers used in the TestBed setup are Alix 2d2. The main parameters of these routers are:

- CPU: AMD Geode LX800, 500 MHz;
- DRAM: 256 MB DDR DRAM;
- Expansion: 2 miniPCI slots, LPC bus;
- Storage: CompactFlash socket, 44 pin IDE header;
- Operating system: Linux.

The routers were set up with 'Ubuntu 12.04 LTS' Linux operating system with custom modifications in kernel. Not all of the Linux kernel modules are needed in the current system and by customizing the kernel, we can achieve better performance and get support for additional hardware. Specifically FTDI drivers were added to the kernel, and redundant modules removed.

The routers provide the base functionality of data transport within the TestBed.

Data transport In the first iterations of the system, a basic data gathering from DUT to Server through Router and TestBed adapter was implemented using MansOS remote access functionality. While testing this approach it was noticed that the server can't handle high amounts of data coming from sensor nodes with sufficient speed, since to gather data it has to ask for data, process it and then store it in local database, and only then the data can be presented to the user.

To make this process more efficient it was decided to split the amount of work between the server and routers by moving data processing part to routers and simplifying data asking process.

This means that the router prepares data and inserts it into database on server directly not by waiting on server to issue these commands. Even though the routers can gather data and insert it into server's database independently, they need to know when server actually needs data and what data is needed, so a high level control interface was developed between server and router instead of the previous polling approach. This was implemented by a socket listening on router, so when ever server needs to do something with a sensor node, it connects to the socket on the appropriate router, to which this node is connected through TestBed Adapter, and sends the router a new configuration. The configuration currently is defined as follows:

```
class TestBedForwarderConfig:
   # None means no update
   ihex = None # ReprogrammingInfo class
   configADC = None # configADC class
   configDAC = None # configDAC class
   # Following request can be optimized by using bits for
each request instead of variables, use if needed
   motelist = 0 # 0 - No, 1 - Yes
   readUART = 0 # 0 - No, 1 - Yes
   readEnergyData = 0 # 0 - No, >0 is interval in seconds
   readADC = 0 # 0 - No, >0 is interval in seconds
   readDAC = 0 # 0 - No, >0 is interval in seconds

class ReprogrammingInfo:
   ihexBinary = None
   moduleNumber = None

class configADC:
   pass # TODO

class configDAC:
   pass # TODO
```

This configuration allows the acquisition of a list of connected sensor nodes, reprogramming of any TestBed adapter modules, as well as controlling the data

reading of all 4 TestBed adapter modules. Note that communication module allows direct communication with the Device Under Test (as described in the next section) so actually we can also read data from it and reprogram it remotely.

1.2.4.5 TestBed adapter

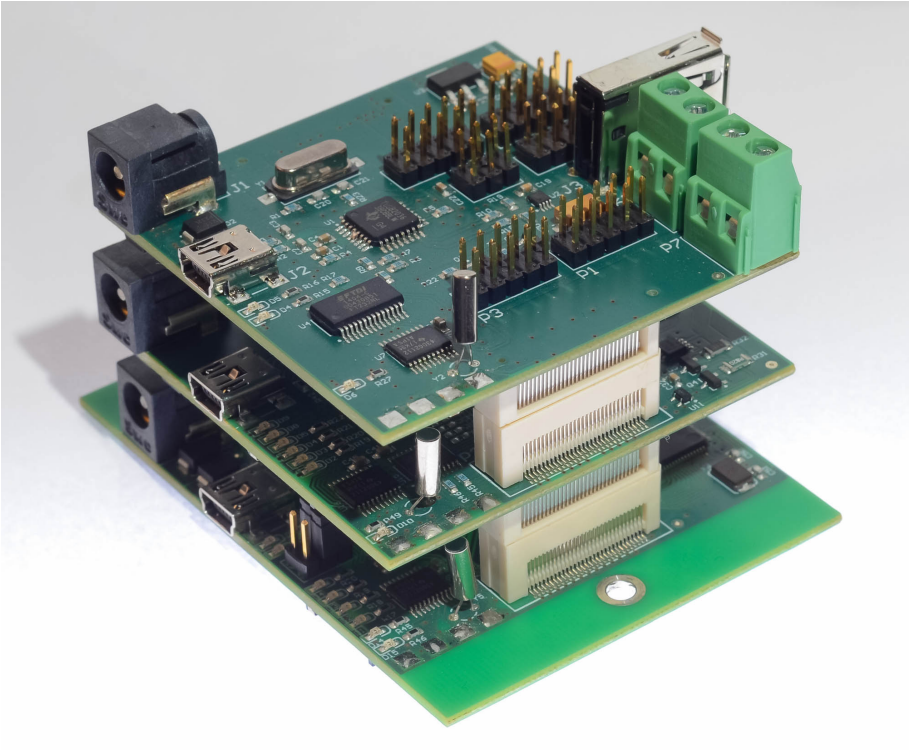


Figure 1.2.4: EDI TestBed adapter

EDI TestBed adapter (Figure 1.2.4) is a result of previous research work in the institute, related to prototyping and profiling low power embedded systems, such as EDIMote[16]. It is intended to test and debug low speed embedded devices, especially wireless sensor nodes.

EDI TestBed adapter has many features to extend testability for embedded devices:

- Emulate battery discharging,
- Measure consumed current of sensor node,
- Generate - digital and analog signals,
- Measure - digital and analog signals,
- Store measured data on local SD cards.

The adapter is designed with modularity in mind, so that new functionality may be added later as required. Three specially synchronized modules are processing and debugging data. Modules are – communication, power meter and signal conversion. Each module has it own task respectively.

- Communication module: Controls data flow between router and sensor node and other adapter modules. With additional software, stored on router, it is possible to access every stacked module, through communication module. Only one connection at the time, between router and modules can be established;
- Power metering module: This module evaluates connected sensor node power consumption and supply voltage stability;
- Signal conversion module stores and processes data to avoid data loss or modifies the current data as needed.

The communication between these modules is controlled by a TinyMCU controller, as described in the sections below.

The design of TestBed adapter was very time consuming and iterative process. The current version is the third complete rework of the system.

TestBed adapter power module sets the output voltage and measures the current. Output voltage is controlled by MCU via LDO (low dropout regulator). Voltage control block diagram is shown in Fig.1.2.5 below.

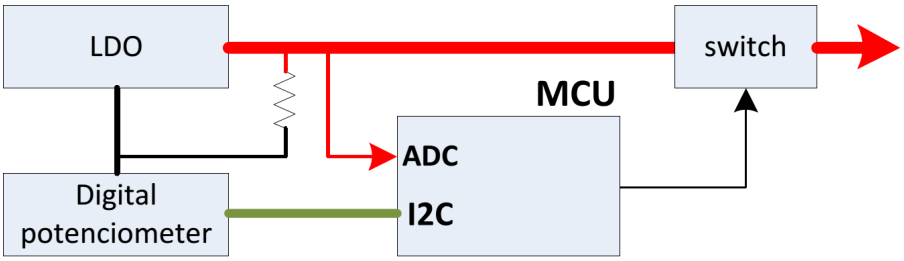


Figure 1.2.5: Voltage control schematics

The voltage drop across the series element - resistor and digital potentiometer. Potentiometer is controlled by MCU. Higher potentiometer value means higher output voltage. MCU has a feedback, to measure the correct value of output voltage.

The second ability of this module is the power consumption measurement. Basically, there are in series connected two elements, resistor R (science) and the load. We know the value of the resistor and the voltage, it is a simple way to calculate the current. But the voltage drop across the resistor depends on load, because the constant current value would be changed, if the load resistance is changed. Simplified circuit architecture shown in Fig.1.2.6 below.

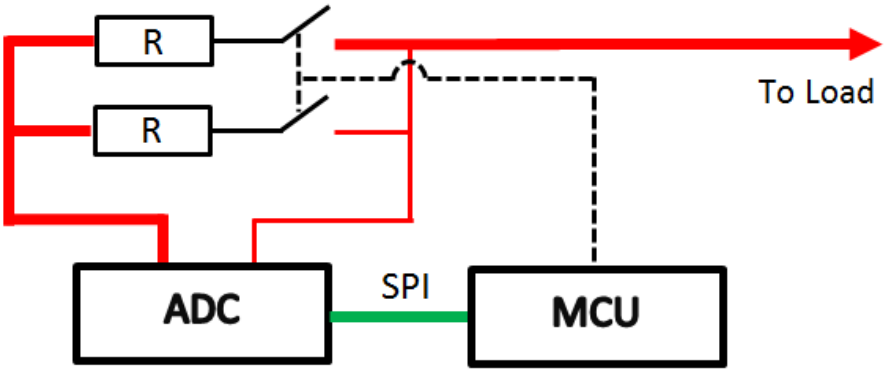


Figure 1.2.6: Voltage control schematics

MCU is measuring the voltage and calculating the current. Filter is connected in circuit between MCU and amplifier to avoid high frequency noise.

Because all of the adapters need to work in the same way to test all of the DUT equally, and because the production process has variable results, the TestBed adapter parameters and performance must be evaluated and calibrated on all 100 devices so that these variations can be taken into account when analyzing measurement data and even more importantly, the DUT can be protected from incorrect or even damaging operational parameters such as supply Voltage.

To run these tests an automated testing setup was developed for TestBed adapters. To test whether measured voltage is the same as real voltage on the pinouts, we performed a lot of tests and data correlations using digital oscilloscope to automatically get real voltage values from TestBed adapter pinouts and compare measured data with MCU calculated values. Oscilloscope was connected to computer via USB to get all values for data correlation. TestBed adapter also was connected with the same computer and after required voltage value was set by software, MCU printed out value from its analog to digital converter(ADC). After both values were logged form oscilloscope and from ADC, we compare these values with reference voltage calculated by formulas of LDO expected values.

In the next periods of the project an even more thorough automatic testing procedure will be developed for the TestBed, so that adding new nodes or improving the existing nodes can be done as securely and smoothly as possible.

1.2.4.6 Communication

To fully use the possibilities of TestBed Adapter hardware we have tried to implement fast, robust and scalable communication protocol for communication between TestBed adapter and Router. There are two levels of communication:

1. Router to TinyMCU
2. Router to Selected Module

Router to TinyMCU TinyMCU is like back-end of TestBed adapter, it performs all the background work necessary for TestBed adapter operation. Since, there is only one FTDI for communication with Router, only one of TestBed adapter modules can communicate with router at any given time, therefore we need a mechanism to switch between modules which can communicate with Router. The module which can communicate with the router is called active module, since its UART at the moment is muxed to the FTDI chip and therefor this module is capable to communicate with router via UART interface.

The main task for TinyMCU is to switch active module on TestBed adapter. But the problem here is that from the router point of view we only have one USB cable connected to TestBed adapter, therefore there are only one Serial connection to communicate with active module and with TinyMCU, but UART can only have one transmitter and one receiver. The solution, as designed in hardware, was to only connect some of the UART pins to TinyMCU(RTS, DTR as inputs and CTS as output), so we could implement two wire communication protocol so Router could tell TinyMCU which module should be activated, and not disrupt communication between original UART endpoints - Router and active module.

As we implemented 2 wire communication for TinyMCU we soon found a problem in this setup, the DTR pin used for 2 wire communication is connected to active module reset pin via UART pinout, so each time we change the pin's state active module gets restarted. This can not be allowed since TestBed adapter modules have way too important role in this setup and such restarts could make any experiment data useless, because module restart could occur at some important point of experiment and it could miss some crucial data. The only solution we could find was to stop using DTR pin for our 2 wire communication protocol, so it effectively became a 1 wire communication protocol.

Now, as it usually works in a 2 wire communication protocols, one wire is used as data and the other wire is clock, when signal on clock wire generates a raising edge, we read signal from data wire, if it is close to 0V the transmitted bit is 0, if data wire's signal is close to 3.3V the transmitted bit is 1, we can collect any amount of bits necessary to understand the transmission from that point. As it was implemented there was a 16 bit unique key, used to avoid any noise being recognized as data, which is followed by 3 bits of data, that allowed to select up to 8 possible modules(only 4 modules currently designed and 4 more slots left for scalability purposes).

Designing a 1 wire communication protocol is quite a challenge considering that the same wire can be used by UART communication protocol, which means that not all the data send by it is intended for TinyMCU. And one thing which made it even more challenging was that TinyMCU(known as MSP430G2112 MCU) had no support for timers in MansOS operating system used to program them. At that time it was decided to try and implement communication without time counting. the implementation is as follows.

Router to TinyMCU: 1-wire, no timers Router uses PC_Control.py script to switch active modules on TestBed adapter. Script uses serial COM

port RTS line to send 10 bit check sequence and 4 bit data sequence to TinyMCU, which performs the module switch on TestBed adapter. 10 bit check sequence consists of 10 high-low transmission bits, all being 7ms long (except the first one "0" bit, that is 3ms long because of serial port being opened `ser = serial.Serial(port, rate, timeout = 1, parity = serial.PARITY_NONE)`. Every data bit length is 24ms to 32 ms depending on device used (Alix router and PC).

External interrupt is used on every falling edge transmission for TinyMCU to synchronize with the waveform, because the timing on different devices may vary in high ranges - for example, same code time.sleep(0.006) in reality makes 6ms delay on PC and 7.8ms delay on Alix router.

The actual PC_controly.py script generated waveform received on TinyMCU can be seen on Figure 1.2.7. Red arrows point to signal edges where logic level is checked. Black arrows point to edges where interrupts are captured.

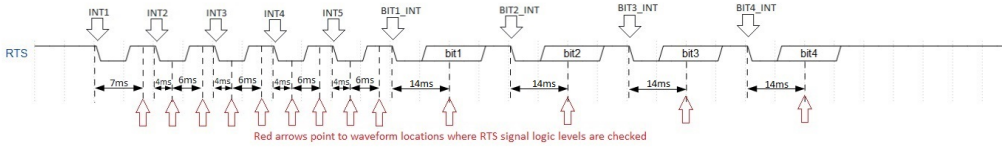


Figure 1.2.7: Intended waveform

The waveform might look complicated, but the TinyMCU does not have other peripherals and functionality useful for capturing data on single wire then external interrupts and mdelay(), msleep() functions. Waveform edge check timing, where logic levels are captured (pointed with red arrows), are calculated based on average bit time executing on PC or Alix router.

There is one extra bit included before and after data bit, for TinyMCU to be able to catch the falling edge transition and synchronize with device that executes PC_control.py.

The same PC_Control.py script executed on PC looks like in Figure 1.2.8.

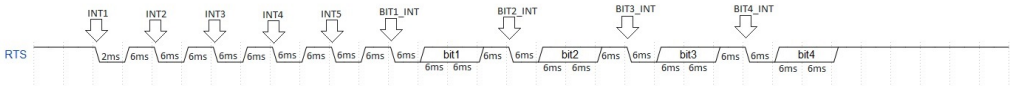


Figure 1.2.8: Waveform when executed on PC

PC_Control.py script executed on Alix router looks like in Figure 1.2.9

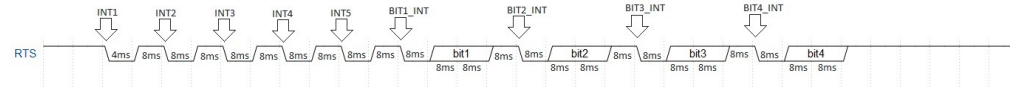


Figure 1.2.9: Waveform when executed on Router

You can see, that specific delay values, when checking received bits on TinyMCU, are chosen based of a difference in bit timings on different devices. Furthermore, the 6ms value in **time.sleep(0.006)** function

is chosen, because **time.sleep(0.001)**, **time.sleep(0.002)**, time.sleep(0.003), **time.sleep(0.004)**, **time.sleep(0.005)**, **time.sleep(0.006)** all results in 7-8 ms delay on Alix router, but **time.sleep(0.007)** to **time.sleep(0.012)** would result in 16ms delay and so on. So the **time.sleep(0.006)** gives the lowest and most accurate value, that is the reason why it is chosen as the delay time for each bit (even the data bits are received using 6ms delays - Interrupt ->Low for 6ms ->High/Low for 6ms ->High/Low for 6ms ->High for 6ms ->Next data bit interrupt).

The acknowledgment is sent by TinyMCU using the COM ports CTS line. Switched modules number is represented by pulling CTS line high and low for modules-number times. If router receives the same number high/low transitions as the modules number that was switched before, it means that active module switching was successful.

To predict the occurrence of the error, while performing the TestBed Adapter module switching, we made a software for this purpose. Based on existing python script 'PC_Control.py' we create a new script, that calls the 'PC_Control.py' and uses it to select a random module. The program called subscript several times and measured switching errors. The test was run on different routers and computers, to ensure that the data are valid and specific hardware parameters didn't influence the results. We performed about 100 000 TestBed Adapter module switching operations and it failed from 5 to 10% of times so it is a relatively frequent error. As mentioned previously, this error is caused by the hardware as well as software, that forms the output signal to switch TestBed Adapter modules. One of the many factors that cause the error is the time delay between calling subscript. When we are decreasing the time delay error occurs more often, but when we are increasing the time delay our best achieved result was about 4 - 5 % of attempts still failing. We are expecting a very often switching frequency when TestBed Adapter will be fully operating in the system, so this error is too regular and will cause a significant delay in all TestBed system.

Router to TinyMCU: 1-wire, with timers Since the communication without usage of timers does not perform quite as good as we expected, we decided to add timer support for TinyMCU in MansOS operating system and then build our communication protocol based on proper time accounting. As of now the timer functionality has been implemented and a new protocol is being developed and tested.

As of recently it was discovered that the problem with DTR pin restarting TestBed adapter modules is still a thing, Router pulls DTR pin low for about 4 ms and then sets it back up at the serial port opening event, which at the moment is beyond our control. We are looking for solutions to this problem as of now.

Router to selected module When we have selected which module we want to communicate to we can start the actual communication, e.g. experiment data retrieval and experiment control. To make this communication protocol robust

and scalable we decided to use High-Level Data Link Control(ISO 13239) style packets, the packet format for both directions is as follows:

```
U8 sflag; //0x7e start of frame flag
U8 cmd; //tells the other side what to do.
U8 len;  // payload length
U8 payload[len];  // could be zero len
U16 crc; // 16bit CRC
U8 eflag; //end of frame flag
```

Total length of packet minus flags is len + 4 bytes. 16 bit Cyclic Redundancy Check(CRC) is used to ensure that received data is valid. Field **cmd** defines what kind of command/data is transmitted and field **payload** contains data transmitted or command parameters depending on command.

Some of the commands will be generic to all modules, like asking for module name, starting experiment, asking for transmitted data etc, but most of control commands will be module specific since each module contains specific hardware and have specific capabilities, at the moment we have developed approximate commands for controlling data acquisition module, which actually contains two modules, one for ADC and the other one for DAC.

1.2.5 Results

During the first two periods of the project a TestBed system is continually developed, including development of a new modular TestBed adapter which allows complex control and analysis of DUT. The adapter hardware has been tested and some problems have been found with communications through TinyMCU, that are currently being resolved.

The TestBed server has been upgraded and moved to a dedicated server room in EDI.

The TestBed user interface has been developed to provide a more convenient way of using TestBed and programming the DUT (including start of development of online code editing tools).

TestBed communications architecture and protocol has been reworked by shifting some work load from the server to the other TestBed components thus resulting in a more decentralized, fast and effective design.

In the second period of the project a lot of tests have been performed to make sure that TestBed is working as expected and during this testing we have already found some crucial problems with overall software architecture as well as with some individual components and their calibration.

1.2.6 Discussion and future work

In future we are planning to achieve multiple long-term and short-term goals to make EDI TestBed a convenient and powerful tool helping people around the world to develop their wireless sensor networks or test their prototypes faster. This includes:

- Improving our user interface to make it more convenient as well as more functional;
- Improving data visualization by introducing a lot of different ways of graphical data representation to suit every need;
- Adding support for cloud program compiling and online code editor linked together with MansOS and SEAL for faster and easier code development;
- Calibrating and improving TestBed adapter to provide more meaningful and precise data about Device Under Test;
- Improving all of our communication protocols to allow faster and more reliable data transmissions and more precise experiment control as well as support scalability for future TestBed improvements;
- Developing more reliable overall TestBed architecture to distribute load evenly among different TestBed components and thus achieve even better overall performance;
- Validating the TestBed system by approbating it for use in other projects and by third parties (such as businesses working with WSNs).

Chapter 1.3

MedWear - Medicine and telemedicine uses of CPS

1.3.1 Introduction

In this section healthcare applications of cyber-physical systems are researched to reach the goal of improving the quality and convenience of face to face medical services as well as remote telemedicine services, while facilitating more efficient prophylaxis, more timely diagnostics and more successful treatment and rehabilitation based on innovative solutions.

The evidence for beneficial applications of wearable computing can be found in many different fields, such as medicine [17] from early detection [18][19][20][21] to treatment [22][23], care for the elderly [24][25][26] and frail[27], exercise [28][29][30], mental health [31], entertainment [32][33] and potentially many others.

Also it is not hard to imagine, that both patients and doctors could benefit from low-cost unobtrusive smart clothing, which could be used for patient compliance monitoring, long term medical data analysis in patients everyday environment as opposed to special testing facilities, telemedicine, unobtrusive vital sign and traumatic event monitoring etc.

Unfortunately each of these examples of previous work in the field of wearable sensor applications have been working on their own custom wearable sensor platform. Because of this a lot of time and energy goes into reinventing the wearable data gathering infrastructure, which could be better spent in researching beneficial applications to wearable technologies.

A standardized architectural, hardware and software process for development of smart clothing could provide tools for development of customizable wearable sensor and actuator networks with little effort and low budget. In addition such wearable sensor ecosystem could provide similar benefits to the growth of the field of wearables as personal computers did for the field of computing - competing companies could work on specific parts, such as specific sensors and not be concerned about the complexity of developing and marketing a full wearable system.

Because of this our team researched the state-of-the-art in these fields and defined a vision of smart clothing architecture capable of providing these benefits. Because of the limited resources available in this research project, priority directions were selected and specific key parts of this vision were researched (such as key architectural elements or use of specific wearable sensors) thus ensuring, that the overall vision of smart wearable systems is moved forward, and an effective foundation is laid for future applications of this research.

A variety of different illnesses or injuries, for example, cerebral palsy, can often lead to posture and different body part control problems in a number of daily situations. Correct posture, body part alignment, etc. is essential to ensure accurate operation of respiratory system, cardio-vascular system and other body functions. In addition, this information can provide context of human activity that allows to better interpret data obtain with other sensors. Currently posture and body alignment monitoring and training in a variety of different rehabilitation programs normally is done in close supervision of medical specialist. This approach limits availability of rehabilitation and maintains high work load of medical staff. In addition, the monitoring duration is limited to special dedicated sessions, making the daily monitoring of patient outside medical facilities practically impossible. To overcome previously stated limitations a variety of aiding technical apparatus are being used, however, the availability of these are limited in terms of functionality and also ease of use. In this project, a particular subtopic is devoted for development of methods that allow monitoring of human posture, different body part alignment and movements in real time with wearable devices.

1.3.2 Architecture for multi-sensor smart wearable systems

While working on wearable computing and smart clothing and reviewing the state-of-the-art, common use case scenarios have been defined and common requirements have been based on those, providing basis of a universal smart textile as described in the introduction[34].

The defined common use scenario of a smart textile, facilitating rapid, low-cost development and deployment of wearable computing, is as follows (Fig. 1.3.1):

1. The cloth with wiring integrated within in a universal grid is mass produced resulting in rolls of smart textile, similar to the rolls of ordinary textile;

2. Clothing is tailored from the smart textile in the same way as ordinary clothing;

3. Wiring from different physical parts of the clothing, such as sleeves and torso, is connected together in the same network;

4. At some point in the clothing a special connector and pocket is added for

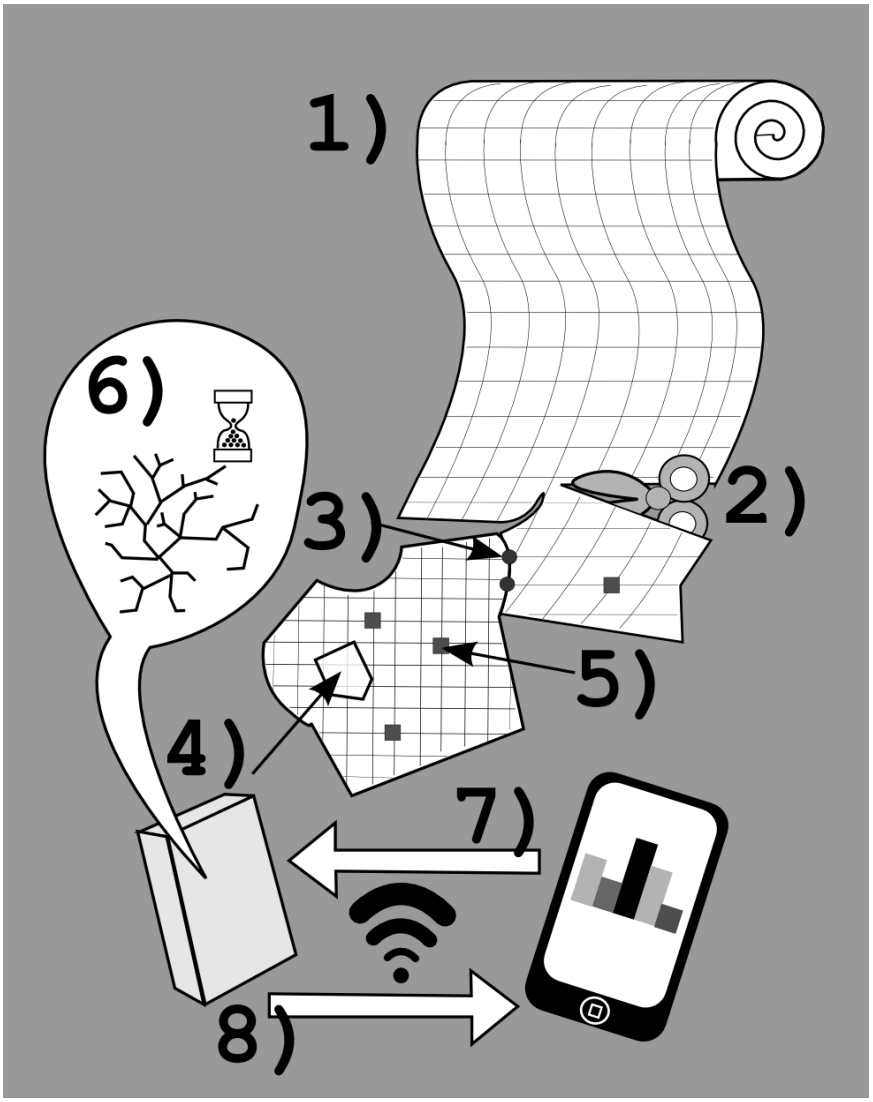


Figure 1.3.1: Envisioned use scenario of a universal smart textile for medical applications

central data gathering endpoint containing a battery and wireless communication module;

5. Depending on the specific task for which the smart clothing is being created sensors are selected and placed in the relevant spots of the ready clothing turning it into a wearable sensor network;

6. The system is turned on and it runs self-diagnosis mapping the network and routing the power, data and clock lines through existing wire connections for optimal data gathering;

7. If needed, specific sensor or actuator drivers are loaded to the whole sensor network, and these drivers are seamlessly distributed throughout the piece

of smart clothing;

8. The data is gathered by the central data gathering endpoint and transmitted over wireless connection to a computer or a mobile device and there software can be developed for the specific purpose envisioned.

To facilitate this scenario a series of basic requirements must be met, which the authors have defined as follows:

- Wires - Wiring system for sensor node connections providing power and signal lines to each of the sensor nodes must be realized with the minimum number of wires. This is because wires are hard to implement in stretchable textiles and they are the most fragile and expensive part of such a smart textile requiring that their number should be kept to a minimum;

- Physical layout - A regular pattern of densely placed sensor node connection points is preferable allowing more universal solutions instead of sensor positioning for a specific problem. Each square meter of textile should include more than 100 sensor nodes - the more potential connections for sensor nodes, the better the chance that a specific node will be close enough to a specific point on the body required for a specific use case. Also a better resolution can provide a more complete picture of the data;

- Connections - Wiring topology connecting the sensor node connection points should provide redundancy for rerouting in cases when some wires are cut while creating clothing from the smart textile - as custom electronics manufacturing is very expensive, it would be more beneficial to develop a universal smart textile. This means, that while tailoring the smart clothing cuts could be made almost anywhere and this potential damage to the network must be taken into account;

- Electronics - The hardware should be small and unobtrusive enough to integrate in the textile and elastic and robust enough to survive everyday use, moisture and other typical stresses. The technology will not gain mass market appeal if it is not easy to wear and will not be robust enough to pay back for its value. Electronic chips and sensors usually are quite resilient. The most fragile part of such smart clothing is usually the wiring and connectors. It must be applied with specific technology to provide elasticity and at the same time maintain the specific electric properties, such as resistance and at the same time protect the electronics from outside elements, such as moisture (The requirement for constant resistance is one of the reasons common methods of developing elastic conductive threads[35] are not a good solution);

- Data gathering endpoint - At any point at the discretion of the clothing designer there should be a possibility to add an endpoint capable of gathering the data from the smart device, providing power to the wearable sensor network and transmitting the gathered data further for processing and displaying. The battery should be small enough to be unobtrusive and

capable enough to power the whole device for a full day of use. This means that the whole system must be built on energy efficient components and principles of energy saving;

- Software - The software accompanying the smart textile should be able to gather the data from all sensor nodes irrespective of the final configuration of the smart clothing and the place where the endpoint is connected. This requires smart mapping of the final garment;

- Updates - Irrespective of the number of sensor nodes and topology of the network it should be easy to reprogram the system with new software or drivers for sensors or actuators without disassembling the network. This means, that sensor nodes should be re-programmable through the network, without direct hardware access to each node. System updates should come from the wireless connection of the endpoint and propagate throughout the network, while it is still powered on;

- Speed - The system should provide data rates of at least several full frames per second for the system to be usable in real time - this includes gathering data of every sensor in the system in each frame and transmitting the gathered data for processing;

In addition - for the system to become widely used and accepted it has to have a potential to be economically viable through mass production, there should be standardized software tools for using the gathered sensor data and the hardware and software standards should be open and accessible for mass acceptance.

This architecture concept has been partially based on the line-topology smart clothing architecture described by the institute in previous state research project. In this project specific parts of this architecture have been developed further, as well as some specific sensors and applications for this architecture have been researched.

1.3.3 Our approach

Our approach to this problem is to identify specific parts of the overall solution and concentrate project resources on those parts e.g. specific problems in data gathering architecture or specific sensors and their synergy with other sensor types, as well as energy efficiency and data gathering/transfer.

In below sections these specific tasks, on which our team has worked during the project are described in detail, as well as the resulting innovation from this research.

1.3.3.1 Multi-branch architecture

During State research program “IMIS” project no. 2 „Innovative signal processing technologies for smart and effective electronic system development” a method was developed to form a network with large number of low-power sensors and

acquire their data in real time. This method was based on the daisy-chain principle. It provides a number of benefits when sensors can be connected in series.[36] The proposed enhanced daisy-chained SPI network greatly reduces the amount of wires in the system. New sensors can be easily connected at the end of the chain, with little software adjustments. View of this architecture can be seen in Fig. 1.3.2 below.

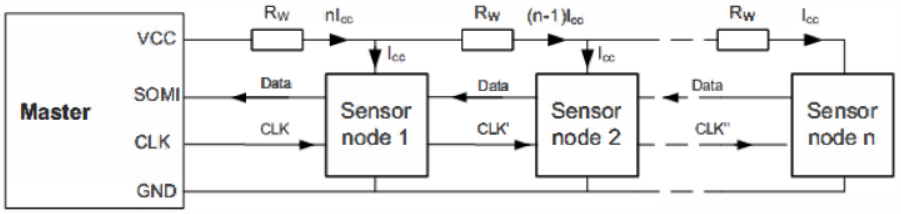


Figure 1.3.2: Enhanced daisy-chained SPI [36]

Several drawbacks arise from this architecture. Because of the series connection and the principle that sensor nodes one by one transfer data in serial manner, data sent over the communication channels has to have similar format. Therefore either sensor nodes have to be homogeneous or their measurements have to be formatted. Also the frequency at which sensors are sampled has to be the synchronized. This architecture is also not very well suited for sensor data acquisition from sensors which are located at various places on whole human body.

Because of these drawbacks a multiple branch network architecture is being considered. Main goal for using this architecture is to increase the rate at which whole system measurements are received as well as to simplify access to remote parts of human body (e.g. feet, palms). This architecture also has to provide the option to connect non-homogeneous sensors and sample them at different frequencies. Proposed architecture consists of one master controller with multiple branches of series connected smart sensors.

This architecture allows to reduce amount of wires needed to connect sensors to the master controller. Multiple branch system is particularly useful for full BSN development, because it provides the possibility to divide a body into sectors of interest. Sensors used to collect data from a particular sector should then be connected in series to form one branch of BSN.

To implement proposed architecture a micro-controller with several communication modules with SPI support running in parallel can be used. This method allows for the user to connect addition sensor branches as long as there is still an unused communication module. It would also allow for branches to be sampled at different frequencies, by setting up communication modules depending on how fast the measured information is changing. Structure of proposed multiple branch architecture can be seen in Fig. 1.3.3 below. For this application, previously described enhanced daisy-chain connection is used for every branch.

During the early stages of development and testing previously designed smart sensor nodes are used. Their software and hardware remains practically unchanged. MSP430F5529 LaunchPad is as master controller used to test the

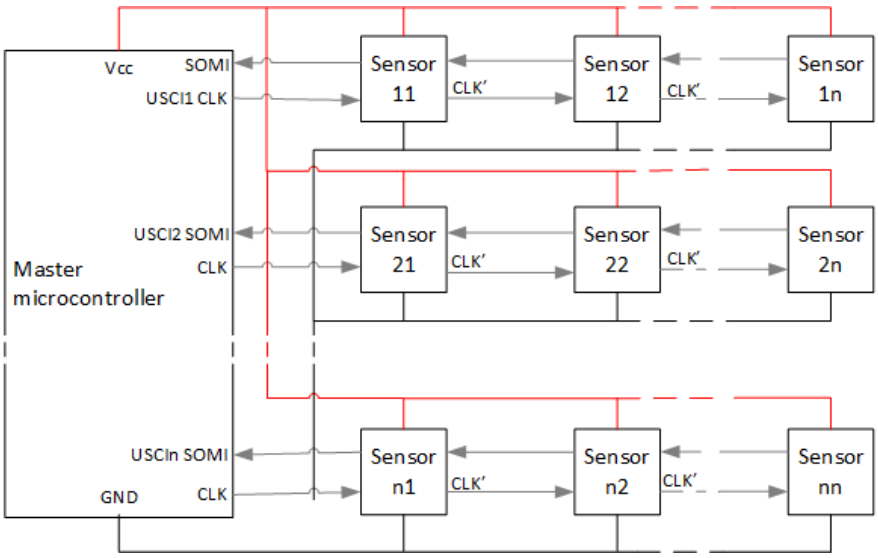


Figure 1.3.3: Multiple branch BSN SPI structure

working principle of this approach. It has 4 communication modules (USCI), one of which is used to send data to Bluetooth module or other wireless link. Remaining three are configured for SPI communication. While the hardware part of the multiple branch BSN is not much more complicated than previously designed enhanced daisy-chain model, software part of the master controller becomes more complex with every connected branch. A simplified flowchart of master controller program is displayed in Fig.1.3.4.

When the network is powered up, the parameterisation is initialized. After the parameterisation is finished master controller sends out pulses, which signal slave devices that data will soon be requested and that it is time to acquire new measurements. Master controller generates clock signals for every branch and stores received data in local memory. Clock signals are generated for each expected byte until every measurement is received. Master controller then passes received data forward via UART connection and repeats the pulses which signal slave devices that data will soon be requested.

Based on what has been accomplished so far we can observe that this architecture relies heavily on the software. Different kinds of sensors should be connected to the network and USCI modules configured differently in order to see how system reacts to different sampling rates.

1.3.3.2 Bluetooth Low Energy

Bluetooth (BT) is widely used wireless communication standard over short distances and it is supported by broad range of mobile devices an operating systems. The latest BT specifications (starting with 4.0) include Bluetooth Low Energy (marked as Bluetooth Smart) technology, which has entirely different lineage and design goals comparing with traditional Bluetooth. It is designed for devices that run for long periods on powers sources such as coin cell batteries and

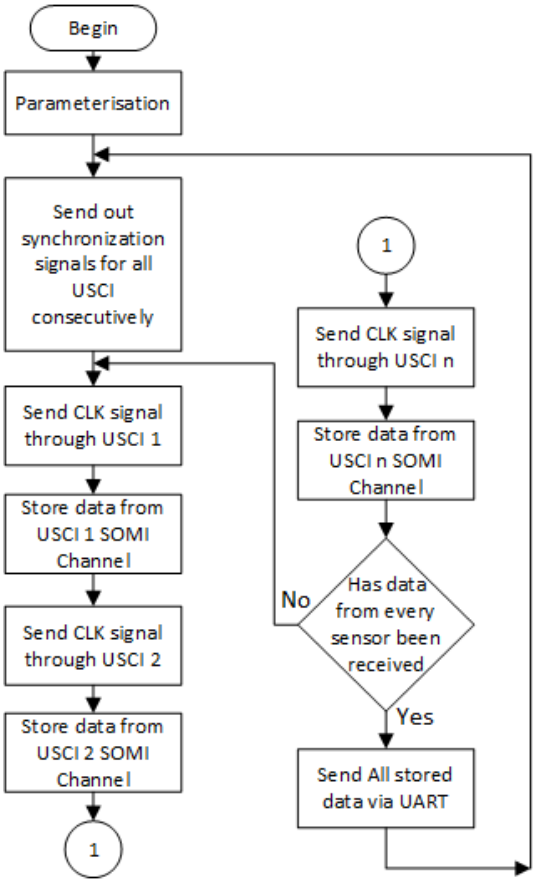


Figure 1.3.4: Master controller code flowchart

energy-harvesting devices. The data throughput of BT Smart is significantly lower than traditional BT (practically 5-10 KB per second[37]), thus it does not replace traditional BT, but provides an alternative for low power applications.

In order to explore BT Smart potential in wearable sensor networks, we tested the TiWi-Ub1 BT Smart module with sensor chain network viewed in Fig. 1.3.2. Developed application used one characteristic to continuously send notifications with acquired sensor data to BT Smart supporting PC or smartphone. Even on maximum throughput we noticed considerable power savings, but for large amount of sensors provided data rate may be insufficient.

BT Smart protocol is very flexible, thus the power consumption and performance highly depends on application and choosing appropriate settings. Further work is required to find the limits of TiWi-Ub1 module energy efficiency and data throughput.

1.3.3.3 Gathering wearable sensor data on a remote server

In various projects we propose systems, that are using mobile platforms in order to visualize, analyze and store data. Mobile platform was chosen taking its availability, ease of use and high computing power into account. Collected data on the smartphone can be stored in cloud infrastructure in order to provide data accessibility and security for both patient and health specialist. By using cloud services it is possible to provide communication in the scope of application between two parties: patient, doing rehabilitation and sharing his statistics with rehabilitation specialist, who can manage prescriptions and view data from patient.

1.3.3.4 Sensors for biomechanic

In this project, methods are being developed, that allow monitoring of human posture, different body part alignment and movements with wearable sensors in real time. In this reporting period a number of new approaches and methods for wearable sensor node signal processing were studied and implemented. Particular focus was aimed towards sensor fusion of inertial and magnetic sensors, which allows to obtain more accurate orientation leading to better measurements of human body biomechanics.

Acceleration and magnetic sensor network for shape sensing

During State research program "IMIS" project no. 2 „Innovative signal processing technologies for smart and effective electronic system development" , an acceleration sensor network [38] was developed that allows to monitor shape of the fabric and can be used in smart wearable garments for posture monitoring [39].

A significant drawback arise from the fact that accelerometers alone can only measure direction of single reference vector, providing incomplete sensor orientation estimate. In order to obtain full 3D surface model another reference direction have to be measured such as Earth magnetic field.

Together acceleration (in static conditions - dynamic acceleration $<< g$) and magnetic sensors provide two vector observations, which is enough for full orientation determination, therefore, to obtain orientation data some deterministic algorithm such as TRIAD [40] can be used. Triad is one of the fastest, singularity free and computationally simple algorithms for orientation calculation from vector observations.

After applying the TRIAD algorithm to sensor measurement vectors the rotation matrix R is obtained. It can be used to calculate surface segment orientation relative to initial position that corresponds to sensor orientation relative to Earth reference frame. Any vector describing surface segment can be simply transformed by multiplying it with R:

$$\vec{v}' = R\vec{v}, \tag{1.3.1}$$

where $\vec{v}$ is the vector and $\vec{v}'$ is the same vector transformed according to sensor orientation.

To reconstruct surface shape following steps are implemented. Acceleration/-magnetic sensor nodes are arranged in regular grid along the surface. The model of the surface is divided in n rigid segments, where $n = i \cdot j$ is the total number of sensors used, so that the segment structure corresponds to the structure of the sensor grid (i and j denote row and column of sensor location in the grid). Each segment is described by four direction vectors, denoted by $\vec{N}[i,j]$, $\vec{E}[i,j]$, $\vec{S}[i,j]$ and $\vec{W}[i,j]$ and segment center point $C[i,j]$. The segment center points are the surface control points which will define the surface geometry. Initially all segments are aligned with global reference system by assigning some base direction vector values such as: $\vec{N_b} = [0;0;\frac{L_1}{2}]; \vec{E_b} = [\frac{L_2}{2};0;0]; \vec{S_b} = [0;0;-\frac{L_1}{2}] = -\vec{N_b}; \vec{W_b} = [-\frac{L_2}{2};0;0] = -\vec{E_b}$, where L_1 is the distance between sensors longways and L_2 is the distance between sensors across in the actual grid. In Fig. 1.3.5 the structure of the surface model can bee seen.

During shape reconstruction the base direction vectors of each segment are translated according to corresponding sensor orientation. By inserting segment direction vectors in the equation (1.3.1), actual orientation of segment can be obtained:

$$\begin{aligned} \vec{N}[i,j] &= R_{ij}\vec{N_b} \\ \vec{E}[i,j] &= R_{ij}\vec{E_b}, \end{aligned} \tag{1.3.2}$$

where R_{ij} is rotation matrix that describes the corresponding sensor orientation and is obtained form TRIAD algorithm. Rest of the direction vectors can be obtained simply as inverse:

$$\begin{aligned} \vec{S}[i,j] &= -\vec{N}[i,j] \\ \vec{W}[i,j] &= -\vec{E}[i,j]. \end{aligned} \tag{1.3.3}$$

Once all segment direction vectors are obtained in actual orientation we can calculate segments center point coordinates, which determine segment locations.According to structure in Fig. 1.3.5, if we denote some $C[i,j] = [0;0;0]$ to serve as reference, all other centre points can simply be obtained using corresponding segment direction vectors, for example:

$$C[i+1;j] = C[i;j] + \vec{N}[i;j] - \vec{S}[i+1;j]. \tag{1.3.4}$$

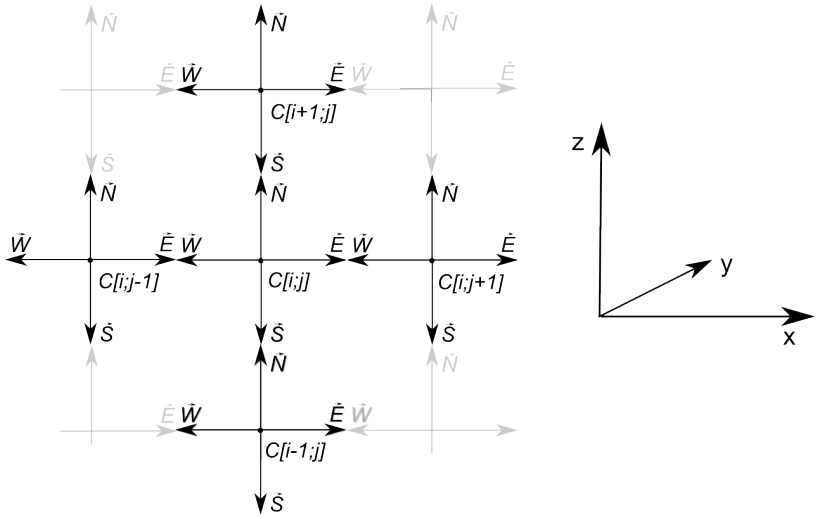


Figure 1.3.5: Surface segment structure. Each segment consists of center C and four direction vectors $\vec{N}$, $\vec{E}$, $\vec{S}$ and $\vec{W}$.

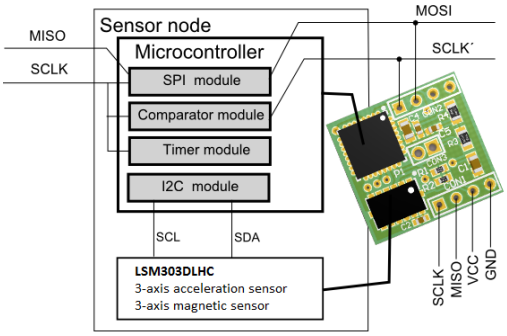


Figure 1.3.6: Structure of sensor node for orientation measurement.

The obtained control points can be seen as 3D point cloud defining the shape of the body similar as in 3D scanner.

Proof-of-concept prototype was designed to demonstrate feasibility of our system and validate performance. A sensor node was designed consisting of acceleration/magnetic sensor LSM303DLHC and low-power microcontroller MSP430G2553 (Fig. 1.3.6). Data from LSM303DLHC is used for orientation estimation. Microcontroller serves as interface between the sensor node and the network and also provides network related tasks. An architecture described in [36] was used to test surface with 63 sensor nodes. Data processing was implemented on PC in Matlab environment and as Java application. In addition, Android application was developed demonstrating ability to implement real-time system on portable devices with limited processing power (Fig. 1.3.7).

To evaluate the performance of proposed shape sensing method, experimental results were compared to commercially available Kinect V2 sensor, that can act as a 3D scanner. The sensor equipped fabric was filmed for several minutes with Kinect v2 sensor while performing various continuous deformations such as

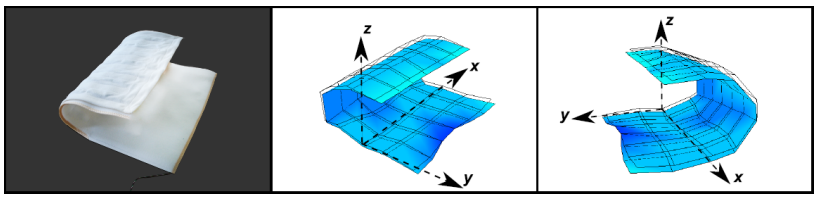


Figure 1.3.7: Sensor sheet and reconstructed shape rendered in Android application with 10 Hz frame rate.

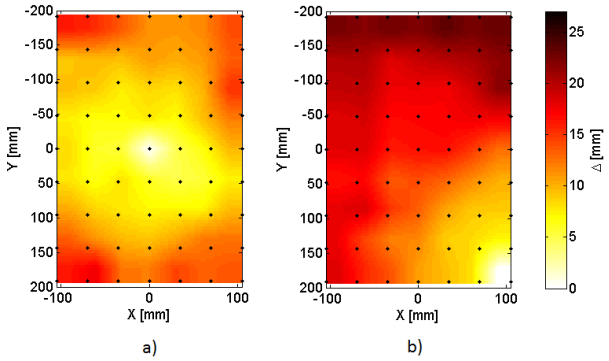


Figure 1.3.8: Difference maps in shape sensing sensor array in comparison to Kinect reconstruction: a) reference sensor located in the center of the grid, b)reference sensor located in the corner of the grid. The black dots indicate the locations of the sensors in the grid (points that are reconstructed).

bending, twisting, etc., to obtain relatively complex and different test shapes. In Fig. 1.3.8 the mean differences of all shapes for each sensor location are mapped on the sensor grid structure.

More detailed information about tests with experimental setup as well as computer simulations can be found in appendix 1.1.

Inertial/magnetic sensor module for motion tracking

Fast dynamic accelerations as well as electromagnetic noise introduce high frequency noise in the reference vector measurements that are used for orientation estimates. This is a particular problem when fast human movements have to be measured which requires bio-mechanics model estimation in dynamic conditions. To eliminate this problem acceleration and magnetic sensor data are often fused with gyroscopes. For this reason an additional sub-task is dedicated for design of small/low-cost/low-power sensor node that can accurately estimate orientations in human bio-mechanics model during dynamic movements.

In theory, orientation can be measured by 3-axis gyroscope alone, by integrating the measured angular velocity of each axis. However, this orientation estimation is severely corrupted by well-known gyroscope drift, which happens due to different measurement error accumulation over integration period. Gyroscope drift can be interpreted as a low frequency noise. This allows acceler-

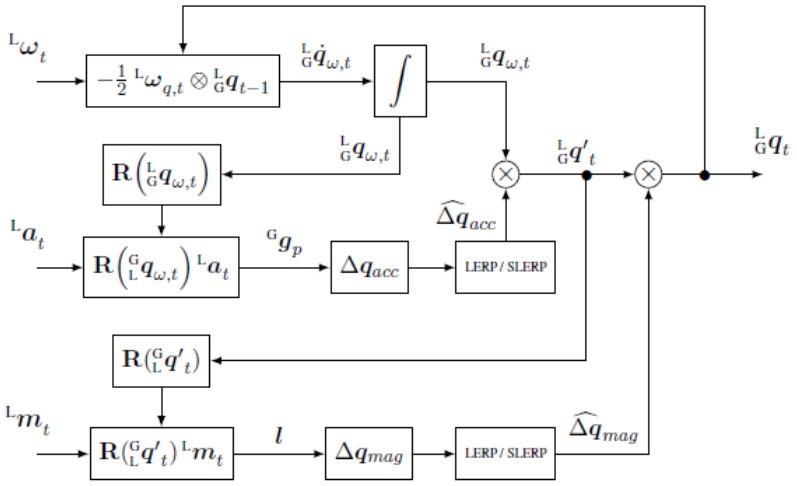


Figure 1.3.9: Complementary filter algorithm

ation/magnetic sensors and gyroscope sensors work together in complementary way, as acceleration/magnetic sensor measurements normally are effected by high frequency noise, but are stable in long run in contrast to gyroscopes.

A number of methods have been proposed for acceleration, magnetic and gyroscope sensor fusion. One of the most widespread is Kalman filter [41]. Despite good performance Kalman filters can be difficult to implement in low-power real-time systems. Also, separate sensor correction terms are not explicitly available, which can be beneficial to adaptively tune the sensor fusion algorithm. Due to this alternative algorithms such as complementary filters have been proposed [42]. This filter is designed for application in Micro Areal Vehicles. It first estimates orientation by gyroscope integration, and then corrects this estimation with corrective term obtained from acceleration sensor and then magnetic sensor. See flowchart of the filter algorithm in Fig. 1.3.9, where L_{ω_t}, L_{a_t}, L_{m_t}, are gyroscope, accelerometer and magnetometer measurements respectively and output is orientation estimate in quaternion form.

We adopted this algorithm for application in wearable sensors for human bio-mechanics measurement. A specific sensor node was designed consisting of 9-axis sensor (accelerometer/magnetometer/gyroscope) and microcontroller (Fig. 1.3.10, that can be connected to our previously used sensor architecture [36].

In Fig. 1.3.11 a demonstration of an early human bio-mechanics monitoring prototype can be seen. Data is acquired from 3 sensor nodes attached to the upper arm, lower arm and palm of human hand. Data from each sensor node is fed to complementary filter in order to obtain orientation of each hand segment. Orientations are then applied to simplified human hand bio-mechanics model to capture motion of the arm.

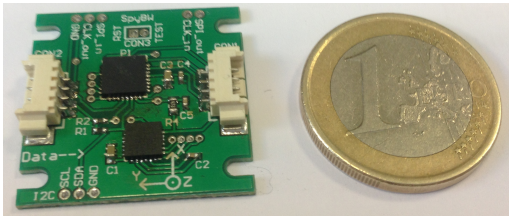


Figure 1.3.10: Prototype of wearable sensor node with accelerometer, magnetometer, gyroscope sensor module.

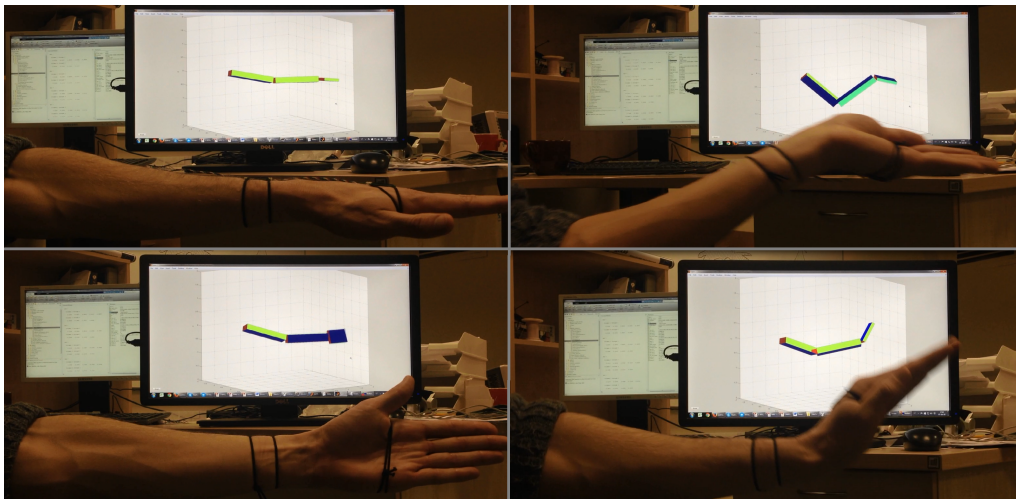


Figure 1.3.11: Human hand movement bio-mechanics reconstruction from wearable sensor node data.

Calibration of inertial/magnetic sensor modules

A major drawback for low-cost MEMS inertial and magnetic sensor application for orientation estimation arise form imperfect sensor parameters such as scale errors, bias errors and axis cross sensitivity and misalignment. In general the output of the 3-axis sensor can be modeled:

$$x' = Sx + b + \mu, \tag{1.3.5}$$

where x - real vector value, x' - measured vector value, S is 3 by 3 matrix consisting of scale, cross axis and misalignment errors, b - offset error and μ - sensor noise. It is crucial to precisely estimate these error parameters in order to calibrate the sensors and obtain reliable orientation measurements. Because of this a particular subtask of the project is set to study and develop new calibration methods for inertial/magnetic sensor nodes.

Particular focus is directed to automatic calibration without external equipment, that would allow on-the-fly calibration during the use of the device. This approach can provide essential contribution to inertial/magnetic sensor systems applications in general by removing the necessity of cumbersome calibration routines, that are usually required before each use of the system. Results of these studies are expected during the next periods of the project.

1.3.3.5 Mobile device for ECG monitoring

Abstract

Judging from Disease prevention and control center data the main cause of death in Latvia still is the heart and vascular diseases. The most important heart and vascular disease test is Electrocardiogram (ECG). There are three different ways to do this test – stationary, load, and long term. The most common is stationary, which is also the most accurate, but the length of the test is less than a minute. Load test is for patients who complain about problems when doing some physical activities. Long term test is for patients who need to monitor heart for longer periods, for example 24 hours. Using Holter system for these long tests is uncomfortable for patient because it is necessary to deliver this system back to doctor for analysis. As a part of our MedWear sub-project we are developing a sensor for patients heart monitoring with potential to integrate it in the overall smart clothing architecture and gain the added benefits of activity monitoring together with ECG monitoring. We are making an embedded system that is easy to use for the patients, doctors and supervising staff. The collected data is sent to the database over the air using the advantages of wireless sensor networks. Doctor can monitor patient's health from distance and decide what to do before the system is even returned. Human movement and environmental factors are interfering with data that is why there is a need for data analysis to filter noise and get more believable results. Using other MedWear sensors we can monitor patients movements and predict noise that needs filtering.

Introduction

Electrocardiogram (ECG) is the most important test that is used for patients with potential heart diseases. Wilhelm Einthoven was the first who invented this system in 1903, later in 1924 he got the Nobel Prize in medicine for his invention.

The same method is still used nowadays. Different electrical signals from heartbeats are monitored using electrodes that are connected to patients flesh. ECG test provides important information about inner workings of the heart and helps diagnose serious heart rhythm disturbances. To use this test firstly patient has to have some complaints about dizziness, blackouts, palpitations etc. ECG is important for patients of all ages, because it provides important information about patient's health.

Most common is the stationary test that is concluded in doctor's cabinet and is no more than a minute long. That means that there is a small chance that in this moment doctor can see the real problem with a patient's heart for example arrhythmia, if it appears only once in a week. To get more accurate readings doctor sends patient to get Holter monitoring system which monitors patient's heart for at least 24 hours. This system lets patient go on with his everyday life while it collects the data. In addition patient has to fill a diary about his feelings, like dizziness, coming blackout, or even an argument with a neighbor. Using this system patient has some discomfort, because, patient can't get wet not to break the system. But the biggest discomfort of all is that patient has to deliver the system to doctor on his own, so that the data can be read and analyzed.

Our group is making an ECG system that includes, data gathering, analysis, and delivery to all interested parties using wireless radio. Using wireless radio eases work for doctors and patients. This functionality is ensured using wireless sensor network principles.

Review of the market and last records in the field of heart monitoring device (ECG, holters)

Production of the following companies has been reviewed:

- **SCHILLER AG**, Switzerland, production 21 type of ECG, including:
  - holter MS-12 BLUE with wireless output;
  - software Medilog Darwin 2 - for holter data processing;
  - holter data registers Medilog: AR4plus; AR12plus; FD5plus; FD12plus.
- **BTL Industries**, USA, has 33 divisions and representatives across the world, manufacturing:
  - ECG for rest and stress conditions - 27 types;
  - holters (3 sorts): BTL Cardio- Point Holter H100/H300/H600;

  - software for holters BTL Cardio Point.

- **SEIVA**, Prague, the Czech Republic, producer of:
  - ECG systems, 2 types - Cardiotouch and Cardiowriter;
  - precise holter system Cardiotrack;
  - short time blood pressure monitor Touchtrack;
  - dealer of holters DL800, DL900, 1200W, produced by **NORAV Medical**, USA.

- **Contec Medical Systems Co, Ltd**, China, producer of ECG - 21 type, including:
  - small size holters 5 types with wireless output.

- **North East Monitoring, Inc.**, USA, producer of holters:
  - DR180, DR181 - with 12 input wires and options - price 750 USD;
  - DR200, DR200/E, DR200/HE - the same - price 1400 USD;
  - CR160 - smallest size - price 500 USD.

- **Mortara Instrument, Inc.**, USA, holter Vision 5L - price 1995 USD.

- **Forest Medical, LLC**, USA, holters Trillium1000 - - Trillium5000, price 1300-1700 USD, software Trillium Silver/Gold/Platinum.

- **GE Medical**, USA, MARS Holter Monitoring System - price 6300 USD;
  - Holters Cardio Day, Getemed CM4000.

- **Midmark Corp.**, USA, holters IQ ECG, IQ vitals.

- **Welch Allyn**, USA, holter HR100, software PCH-100, PCH-200.

- Some holters available from sales company **QUIRUMED**, Spain, Valencia:
  - Micro Holter 939-M1CP - price 249 EUR*;
  - holter HOLT5000 (for using 400 hours) - price 1449 EUR*, additional complete of Input electrodes - 5 pcs- 280 EUR; 7 pcs- 300 EUR; 10 pcs- 320 EUR;
  - Digital Holter Recorder with bluetooth, for using 200 hours, 127-87019308 - price 1350 EUR*;
  - holter recorder, 3 channel, 127-109603 - price 1200 EUR*, the same with software - price 2499 EUR*;
  - digital holter recorder with bluetooth, 3-12 channels, for using 400 hours - price 1800 EUR*, the same with software - price 3499 EUR*.

*) all prices have been given without VAT and delivery costs.

These are WEB sites of some additional medical equipment sales companies:

- www.ebneuro.biz/en/;
- www.flukebiomedical.com;
- www.heinstruments.com;
- www.biomedequip.com;
- www.emedhome.com.

All producers of ECG and holters at the same time are the direct sellers of their own production first and only then their production has been delivered by other dealers. There are at least 15 companies in the Europe, which are dealers of medical electronic equipment, including ECG and holters, made by producers worldwide. Nearest suppliers of ECG and holters for Latvia region are:

- JSC Spectramed - Vilnius, Lithuania;
- Arbor Medical Corp. LT - Kaunas, Lithuania;
- A.Medical Ltd - Riga, Latvia;
- NMS Group Ltd - Riga, Latvia;
- RoLa Ltd - Riga, Latvia - calibration service of all types of ECG and holters.

Target parameters for scientific R&D of new holters

Before connecting ECG monitor to the MedWear we decided to make it a standalone ECG monitor, so that development process would be easier, and after that we create a connection with MedWear. We organized gatherings with Latvian cardiologists to get their view of the situation and requirements for the system. Summarizing the requirements we made a flowchart of the systems prototype (fig. 1.3.12).

Holters as autonomous heart monitoring devices have following most significant parameters and possible range of them:

- number of input signal lines - 1-12;
- stability of electrical contact between electrodes and human body - transition resistance between patient body and electrode less than 1 Kohm;
- sensitivity of analog input and valid number of digital output signal bits after ADC of analog input signals - analog input signal $\geq 1mV$, ADC- 24 bit, data rate $0,125 - 8kSPS$;
- method of holter‘s output signal transferring to the central ECG analyzer:

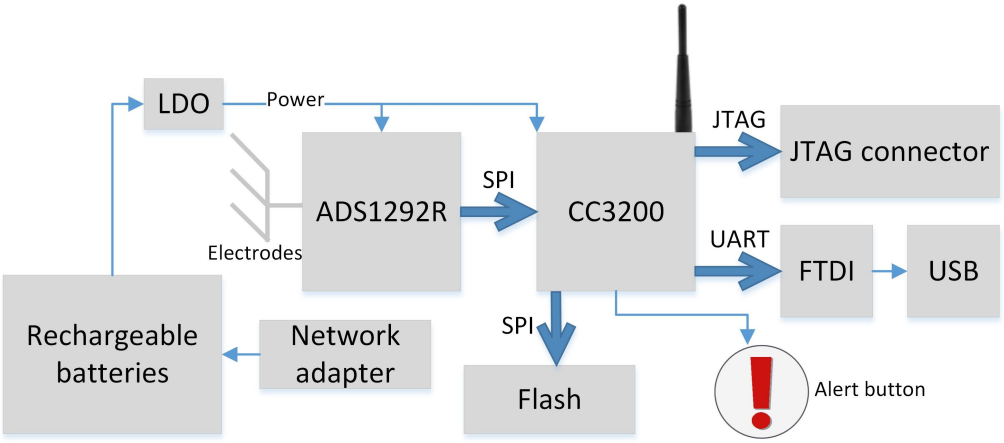


Figure 1.3.12: ECG flowchart

  - periodic reading of accumulated ECG data to PC across the USB cable;
  - taking out the repleted memory card from holter and reading them in PC or central ECG analyzer;
  - wireless non-stop transferring of ECG digital data and alarm signals to the patient mobile phone or directly to ambulance or clinic ECG analyze system;
  - memory capacity - from $256kB$ to several GB;
- mechanical size and placement on the patient body - possible minimal size and weight;
- maximal using time without reading of data, accumulated in the memory, and without changing or charging of the power supply unit (battery) - 24-400 hours;
- possibility to generate an alarm signal in cases, when patient any heart signal condition has gone outside of allowable borders;
- retail price - depending of the holter and ECG monitoring system functions - from 250 EUR till 5875 EUR.

Our aim in R&D work is constructing of new holter with at least 4 input electrodes, wireless output, minimal size and weight, continuous non-stop using time at least 2 weeks (336 hours) and retail price less than 200 EUR.

Choice of the chipset and structure for new experimental wireless holter

First, analysis of the product portfolio of worldwide known biggest companies was done, such as Analog Devices and Texas Instruments, which are producing

wide range of microcircuits and processors applicable for building of medical electronic equipment, including ECG and holters.

Following microcircuits and documents from Analog Devices have been reviewed:

- Low Power Analog Front End microcircuits type ADAS1000/1000-3/1000-4 for 3/5 electrode ECG;
- Possible Low Power, Low Cost, Wireless ECG Holter Monitor, based on operational amplifiers AD623,AD8500, AD8641, microcontroller ADuC702x and wireless output chip AD702x.

But company Texas Instruments offered broad range of solutions for building various ECG devices and heart monitoring systems. Following technical articles, selection guides, application notes and software packs have been reviewed:

- Medical Applications Guide (rev. H, 5.2013.);
- HealthTech Health Guide (1.2015, 5.2013.);
- Signal Chain Guide (1.2015.);
- ECG Quick Reference Guide (4.2012.);
- ECG System Functionality and Evolution;
- Power Management Guide (9.2011.);
- Improving Common- Mode Rejection Using the Right- Leg Drive Amplifier (6.2011.);
- Thermal Noise Analysis in ECG Applications (3.2011.);
- Respiration Rate Measurement Using Impedance Pneumography (1.2011.);
- Analog Front- End Design for ECG Systems Using Delta- Sigma ADCs (rev.A, 4.2010.);
- Wireless Connectivity;
- Possible Structure of Holter Using Microcircuits CC2541 and CC2590, with Wireless Output;
- Code Composer Studio, v.6 - IDE for Embedded Processors (sprt 696);
- CC31xx - CC32xx- Service Pack-1,0,0,10- Installation (12.2015.).

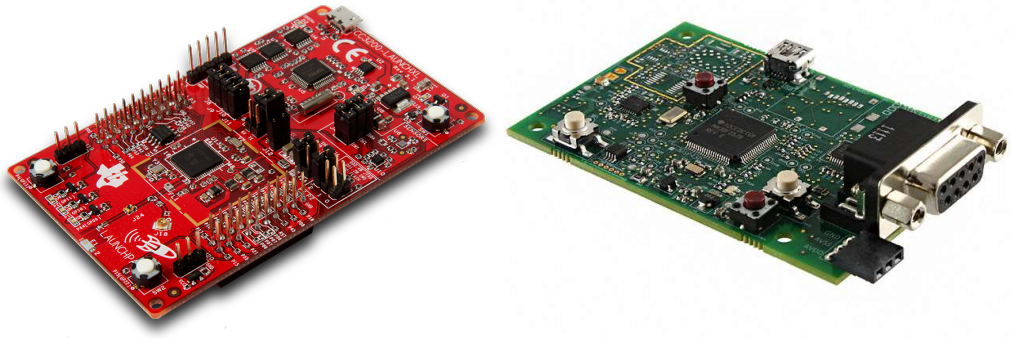
After analysing of all the documents mentioned above there has been accepted decision to build up an experimental holter on basis of following Texas Instruments made microcircuits:

- ADS1292 - analog front- end;

- CC3200 - a single chip wireless MCU;
- some auxiliary chips for programming and power supply.

To learn about the device we used various evaluation modules and development boards of components that are planned to be used in our system:

- CC3200 development board (fig. 1.3.13a);
- ADS1292R development board (fig. 1.3.13b).



(a) CC3200 launchpad (b) ADS1292R development board

Figure 1.3.13: Used development boards

Circuit diagram, printed circuit board and construction of the experimental Holter

Continuing work we created prototype schematics. The main component is CC3200 microprocessor (fig. 1.3.14), which has two integrated devices – powerful ARM Cortex M4 microprocessor and IEEE 802.11b/g standard WiFi wireless radio adapter.

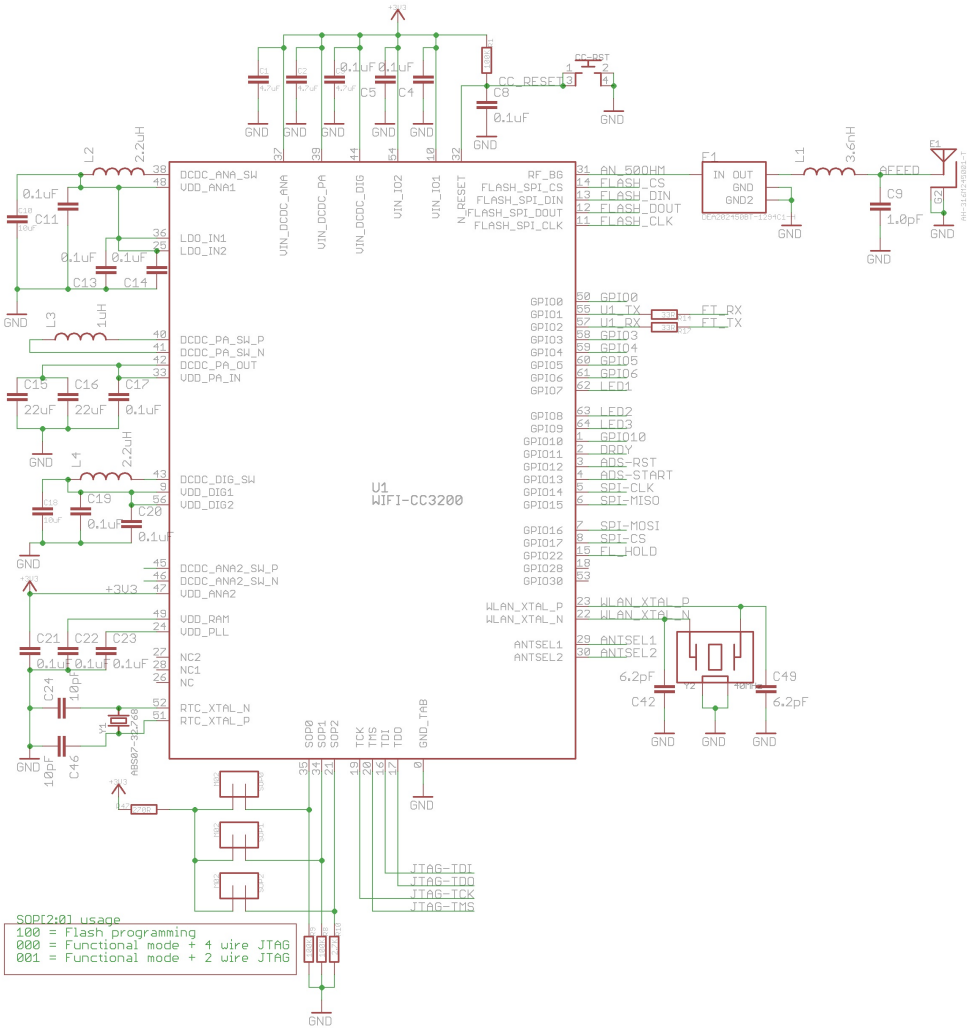


Figure 1.3.14: CC3200 schematic

For gathering of heart signals we use ADS1292R analog to digital converter with integrated repertory function (fig. 1.3.15).

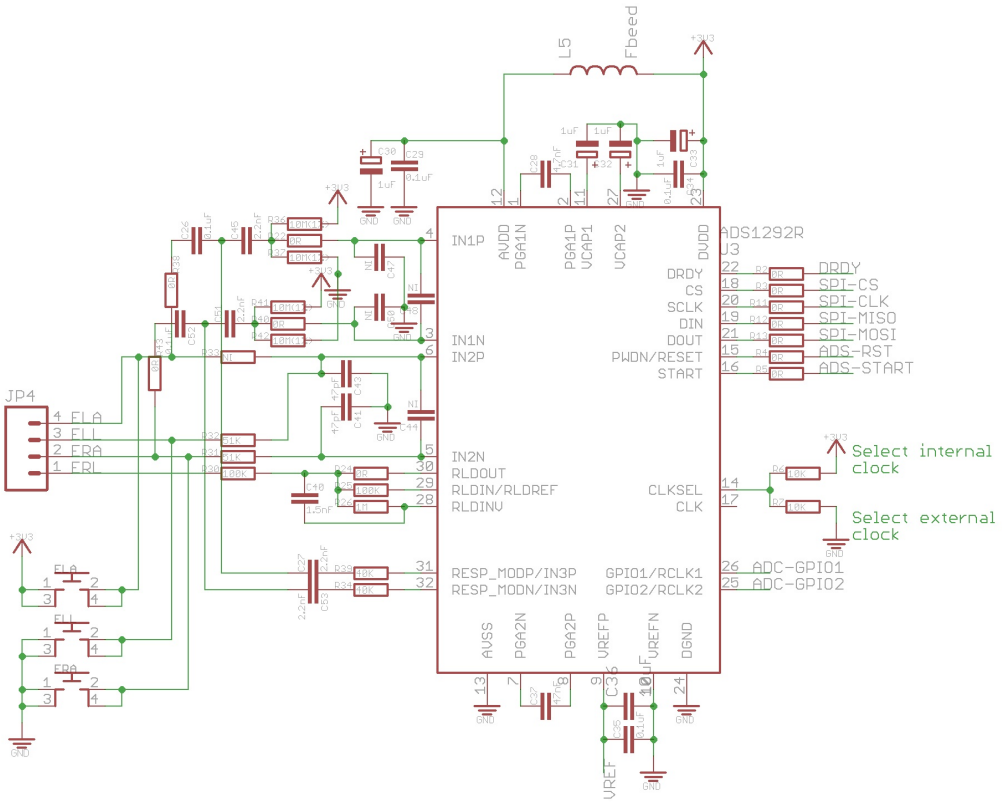


Figure 1.3.15: ADS1292R schematic

For communication with USB devices we use Future Technology Devices International FT232RL component (fig. 1.3.16). To connect with embedded device we use RS232 interface.

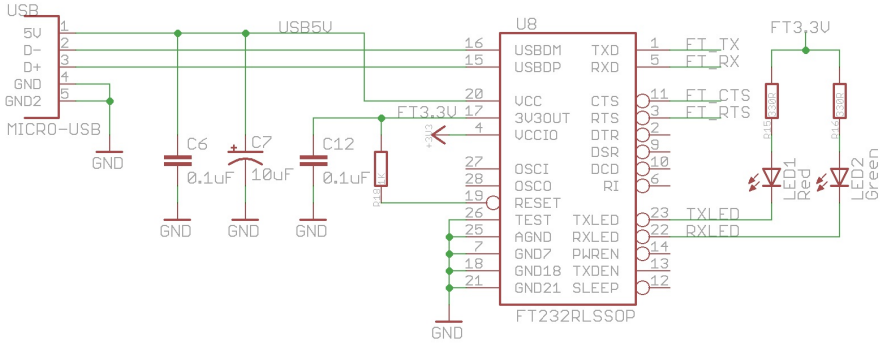
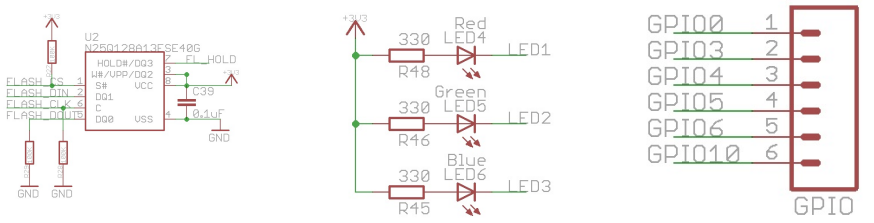


Figure 1.3.16: FTDI FT232RL schematic

CC3200 comes without external flash for data storage, that is why we use N25Q128A13ESE40G (fig. 1.3.17a) as external flash and communicate with it via SPI interface.

For easier debugging we added 3 programmable LED's (fig. 1.3.17b) and couple of unused microcontroller GPIO pins for future use if needed (fig. 1.3.17c).



(a) External flash schematic (b) LED schematics (c) GPIO schematics

Figure 1.3.17: External flash, LED and GPIO schematics

All of used components need 3.3V digital supply to run, so we used LDO to change from USB supplied 5V to needed 3.3V. Also the device can be powered by rechargeable LiIon batteries, so on prototype there is a physical switch to change power source from USB to batteries (fig. 1.3.18)

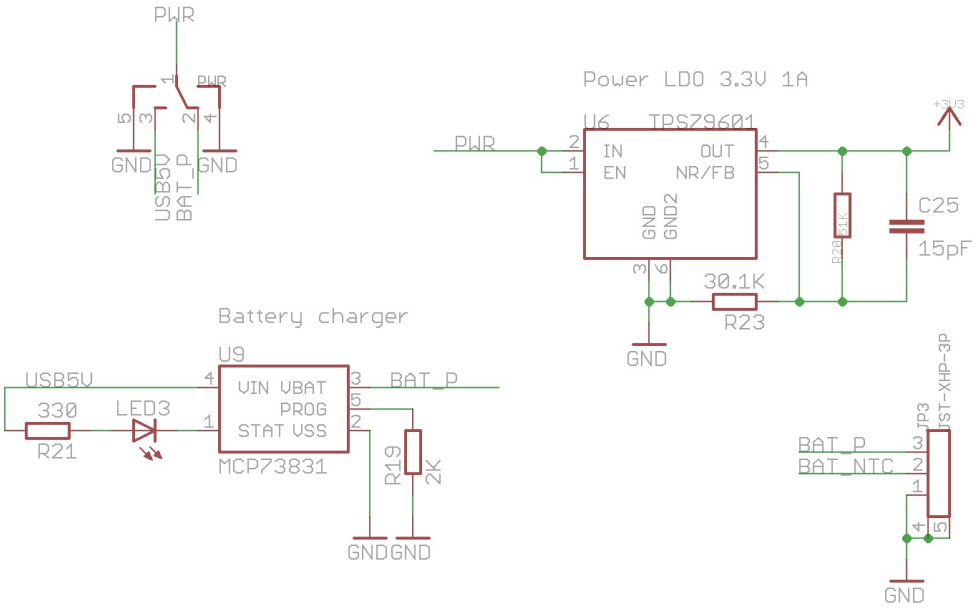


Figure 1.3.18: Power schematics

To program CC3200 microcontroller there are two options:

- USB flashing (external flash is needed);
- JTAG (possible code debugging).

To use USB connection Micro USB cable and Uniflash v3.4 or later is needed. Uniflash flashes program output file into external flash memory,

If there is a need for program debugging we use JTAG connection. To use JTAG Blackhawk XDS100v2 ARM debugging probe is also needed.

After schematic design was done we created PCBs to test out our new device (fig. 1.3.19).

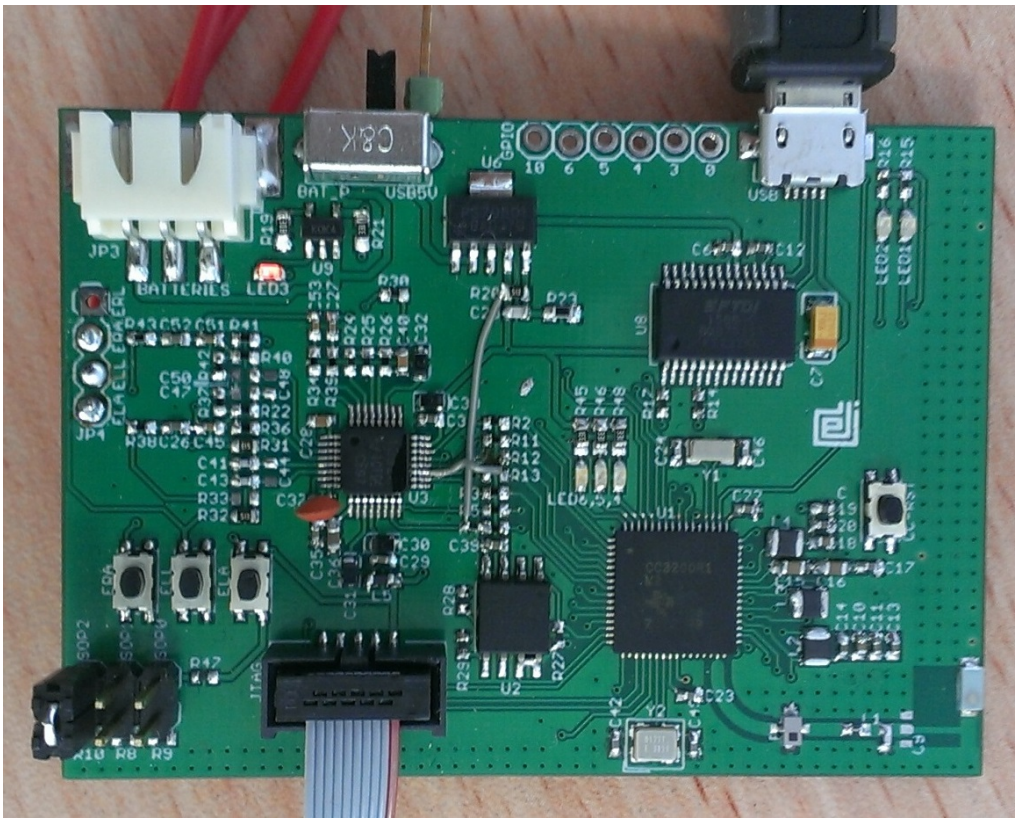


Figure 1.3.19: Heart monitoring device prototype

1.3.3.6 EMG device in wearable network system

Electromyography (EMG) is technique for evaluating and recording the electrical activity of muscles. It is often used in clinical diagnostics and research of human movements. Moreover EMG signals in combination with other sensors can be used as control signals for prosthetic devices and human-computer interface.

In this project EMG is one of the sensor types researched for addition to our MedWear system, to complement other measurements with additional data, as well as benefit from other measurements, such as motion or ECG, in usual EMG applications.

A basic EMG recording device requires biopotential amplifier and at least 3 electrodes - two for biopotential sensing and one as signal reference. Choosing the electrodes, the most common approach is to use wet Ag/AgCl electrodes bathed in electrolyte gel or solution applied on layers of the skin, because they

have reduced motion artifacts and a smaller effect of frequency on electrode impedance compared to dry-contact or non-contact electrodes, especially at low frequencies. Because of the excellent signal quality and good fixation on the skin Ag/AgCl electrodes are common choice in clinical applications.

Other approach is to use dry-contact electrodes designed to operate without explicit electrolyte. Instead, it is usually supplied by moisture on the skin (i.e., sweat). Because of increased skin-electrode impedance, these electrodes suffer from considerable induced motion artifacts and stray interference. Also instability and lack of firming material makes dry-contact electrodes unattractive for clinical applications. But nevertheless the universal and simple design makes them suitable for short measurements and many other applications.

In its simplest form dry-contact electrode consists of conductive layer (i. e. metal disc) in contact with the skin. For more comfort the conductive layer can be made of conductive rubber, foam or fabric. Additionally softer materials better comply to skin reducing contact resistance and improve signal quality.

Comfort and easy application is very important for wearable devices, thus we chose the dry-contact electrode design, which could be adapted for conductive textile. The fig. 1.3.20 shows the test PCB of electrode made from schematic shown in Fig. 1.3.21. The electrode PCB is composed of 3 layers - sensing (bottom) layer, shielding (middle) layer and routing (top) layer.

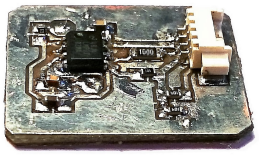


Figure 1.3.20: Active shielded dry-contact biopotential electrode PCB

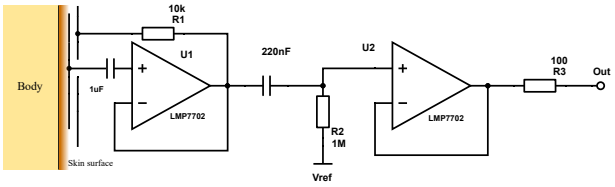


Figure 1.3.21: Active shielded dry-contact biopotential electrode schematic adapted from [43]

Biosignal measurements on the skin surface involve voltages at very low levels, typically ranging between 1 μV and 1 mV, with high source impedance and superimposed high level interference signals and noise. Amplifiers to measure these signals have to provide amplification selective to EMG signal, reject superimposed noise and interference, and provide protection from voltage and current surges for both patient and electric equipment.

Interference generated by electric or magnetic coupling with body and cabling greatly surpass biosignal level, but to amplifier inputs it shows only very small difference in amplitude and phase, appearing as common mode voltage. Thus, a

good biopotential amplifier has to provide strong common mode rejection ratio (CMRR).

To achieve sufficient signal quality, good subject grounding to amplifier has to be established. The best results provides actively driven grounding schemes. Despite commonly referred to as the driven right leg (DRL) circuit, it is important to note that the DRL does not necessarily involve driving the right leg and that the nomenclature stems from its first use in ECG applications. Generally the DRL electrode may be placed anywhere on the body [43].

In order to amplify EMG signal, a commonly used schematic with instrumental amplifier and actively driven ground was adapted (fig. 1.3.22) and tested with dry-contact electrodes described before in fig. 1.3.20.

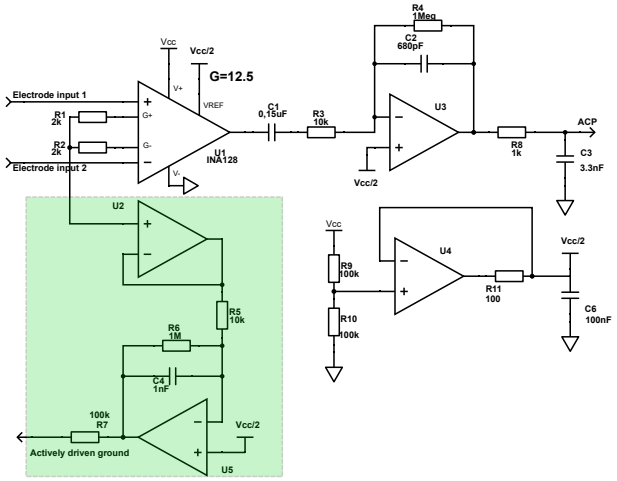


Figure 1.3.22: Schematic of analog biopotential amplifier

Alternative communication for people with motor disabilities is one of the applications, where EMG signals can be used. For example, it is possible to capture EMG signal caused by squeezing eyes and use it as a way of communication.

In order to capture EMG from eye squeezing muscles, electrodes were placed on the temples close as possible to the subject eyes (fig.1.3.23). The obtained signal from amplifier output during three consecutive eye squeezes is demonstrated in fig. 1.3.24. We can see that noise and interference level is high, but applying sufficient signal processing squeezing events could be distinguished with certainty.

Further work will be concentrated on including the EMG device in wearable sensor network, finding sufficient signal processing approach for EMG signal detection and interference suppression and adapting electrode design for textile. Also we are planning to use EMG signal as an answer acceptance for alternative communication application using head position.

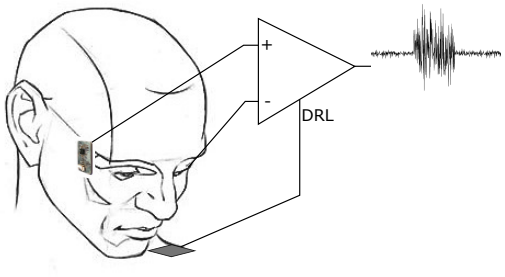


Figure 1.3.23: Electrode placement for eye squeezing detection

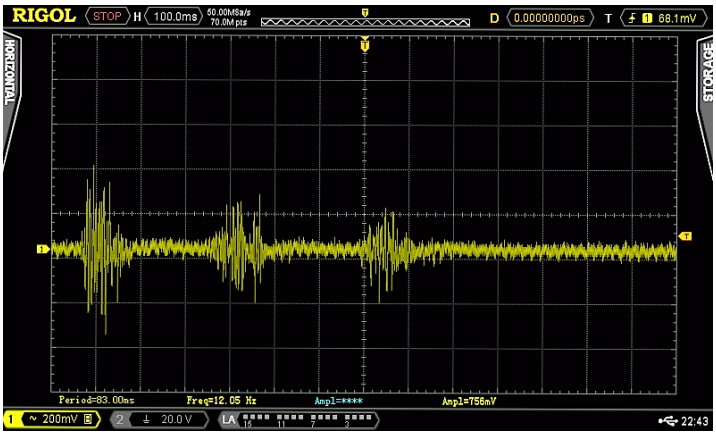


Figure 1.3.24: Captured EMG using prototype amplifier

1.3.4 Applications of technologies developed in the project

The technologies developed in this project has already been applied in several applications, initiated both by our research institute and approbated with external institutions, and also developed as contract research using knowledge and results reached in this research.

In the next subsections main examples of these applications are described.

1.3.4.1 Head position monitoring in therapy

There are patients with cerebral palsy or people after serious injuries who have difficulty maintaining vertical head position. In rehabilitation there are therapies to improve patient ability to maintain normal body position. Therapy with patients who suffer cerebral palsy can be problematic, because communication with some patients is troublesome or almost impossible. In collaboration with rehabilitation center "MEL" idea emerged for system that could improve rehabilitation process for head position therapy. The system consists of previously developed accelerometer/magnetometer sensor device node using the same architecture for data transmission to smart phone or tablet. Sensor module consists of accelerometer and magnetometer sensors, bluetooth communication module

for data transmission to processing device and battery. Module is attached to wearers head with elastic head band as seen in Figure 1.3.25. By using sen-

Figure 1.3.25: Sensor module for head position monitoring.

sor integrated in module it is possible to estimate tilt of the head relative to gravitation vector of earth.

Also specialized software was developed with simplified graphical user interface that provides feedback about head position to the wearer(Figure 1.3.26). The object, seen in application screenshot depicted in Figure 1.3.26, is controlled by wearers head movements. System is calibrated for preferred position of the head before the session. During session the task for the wearer of the sensor is to maintain the object in the marked area in the center of the screen. If object leaves marked area, feedback is triggered changing background of the screen to red color and producing sound alert. This simplified approach allows to perform therapy on patients who are hard to communicate with.

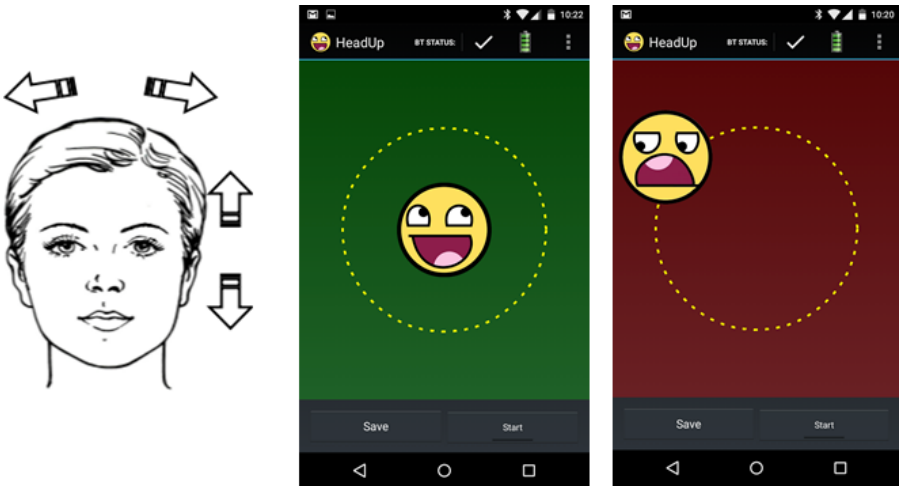


Figure 1.3.26: Head position monitoring application screenshot.

1.3.4.2 Head device for alternative communication

As mentioned in previous section, there are patients that are problematic to communicate with. For this reason the idea for system providing means of alternative communications emerged. Previously described system can be used as computer mouse like human-computer interface device, that is controlled by head motion.

In Figure 1.3.27. screen shot of smart phone/tablet application is shown used as tool for alternative communication. The application consists of moving object (yellow face) and two coloured regions (red and green). Regions represent two options: yes/no type answers. The medical staff can ask yes/no type questions, and respondent can move the object to region representing corresponding answer by moving his head. In current implementation answer is accepted if yellow face marker has been in answer region for specified amount of time. The accept of answer is represented with sound alert. The future implementation will have answer acceptance by using eyelid blinking using electromiography.

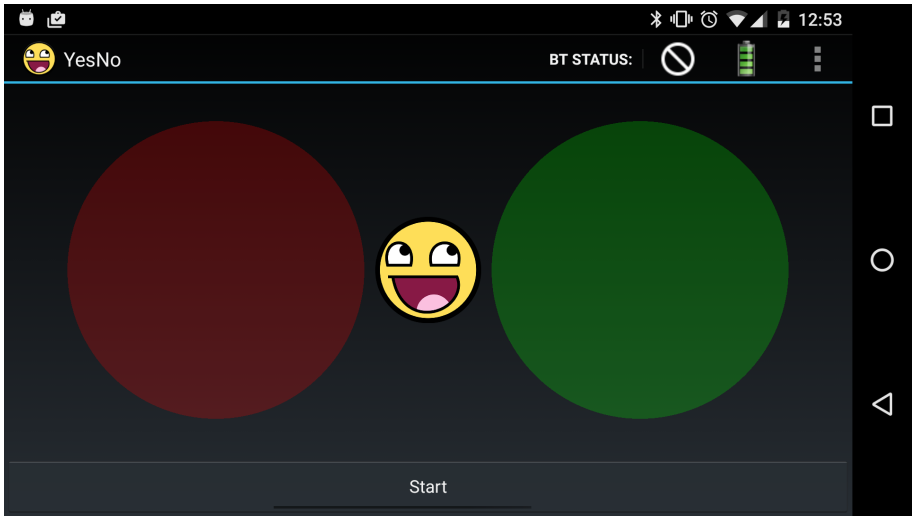


Figure 1.3.27: Application for alternative communication.

1.3.4.3 Posture and head position monitoring in therapy

To obtain combined information about patients head position and posture, prototype was developed in collaboration with rehabilitation center "MEL", that combines head position and posture monitoring functions. Specialized Vest was developed embedding 20 sensor nodes for posture monitoring and additional sensor for head position monitoring. Specialized application was developed for posture and head tilt monitoring providing feedback for the wearer of the system. Application also provides with data logging feature, that allows the data later to be visualized to provide better insight of the session as seen in Figure 1.3.28.

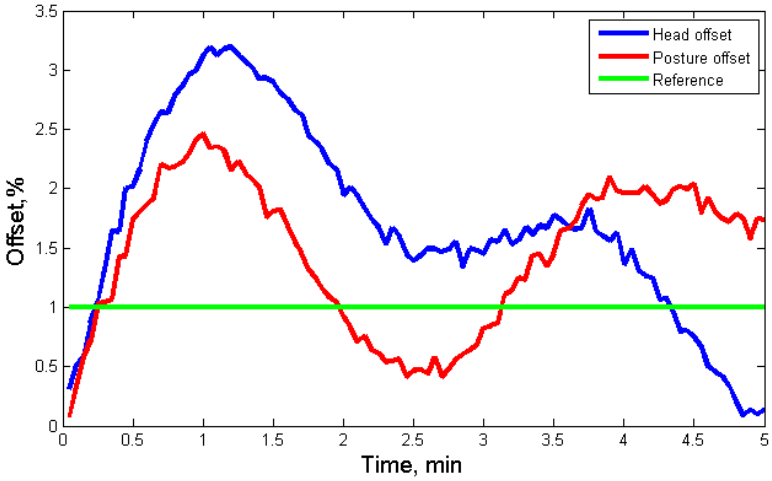


Figure 1.3.28: Head position and posture data visualization.

1.3.4.4 Knee device

Knee joint is the largest joint in the body, and one of the most easily injured. Knee injury is one of the most common reasons people see their doctors. In 2010, there were roughly 10.4 million patient visits to doctors' offices because of common knee injuries such as fractures, dislocations, sprains, and ligament tears. We propose a combination of a wearable sensor system and a mobile application in order to help patients successfully complete rehabilitation procedure. Wearable system consists of 4 sensor nodes used for knee joint flexion/extension angle calculation. After successful vital signs data acquisition from sensor nodes, data is being transmitted to mobile application. As part of the project a mobile application was developed to make calculations, analyze collected data from sensor nodes, store and visualize it 1.3.29. Another important functionality of application - communication with a patient using developed notification system. Notification system was implemented based on health specialist – patient communication style aiming to create a similar feeling of safety when a patient is near physiotherapist. Patient is notified when exceeding the flexion limit threshold and also when reaching the end of rehabilitation session. Developed solution was tested in dynamic conditions – patients used the system during their rehabilitation session in real life. System received positive feedback from participants and rehabilitation specialists 1.3.30. During rehabilitation session developed prototype data was compared to industrial digital Precision of the implemented solution was calculated: 0.79 degrees, fulfilling the requirements received from doctors on current condition.

1.3.4.5 BT Smart Inhaler

Asthma patients use inhalers to deliver medication into the body via lungs. To help them to comply with their medication schedule and improve treatment, we

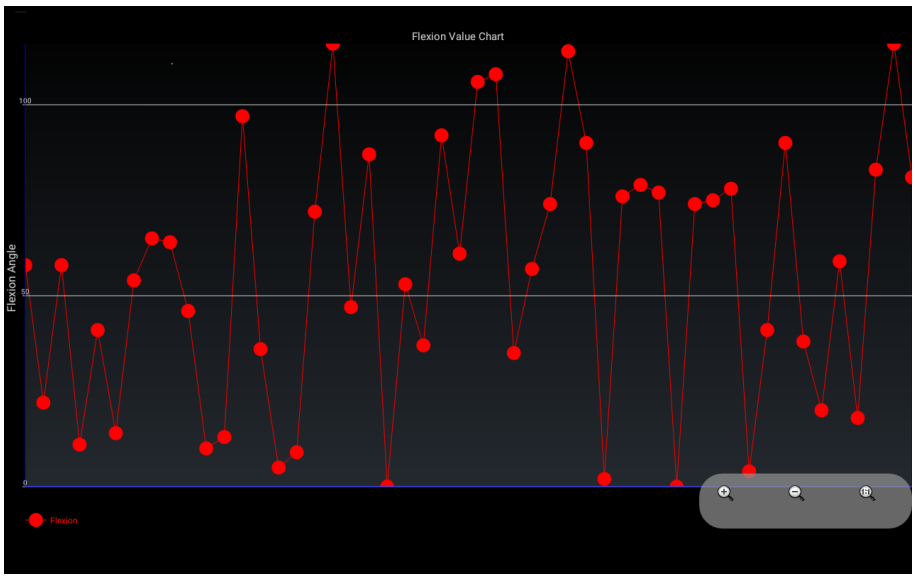


Figure 1.3.29: Knee flexion data visualization.

Figure 1.3.30: Patient participates in system prototype testing.

proposed to use BT Smart technology to send notifications from inhaler to a smartphone or a tablet.

We developed and tested a proof-of-concept prototype using TiWi-Ub1 BT Smart module with minimum circuitry required for device programming, registering push events and sending notifications to BT Smart enabled device (fig. 1.3.31). We also made an Android application, which logs received notifications and assigns corresponding time stamps (1.3.32). In current consumption test,

logging averagely 4 push events every day, device could continuously operate for two weeks powered by two AA batteries in series.

In future we are planning to improve device and application power efficiency to extend battery life to more than a month and implement time tracking into the inhaler device to eliminate necessity of connection to master device during push event to assign valid time stamp.

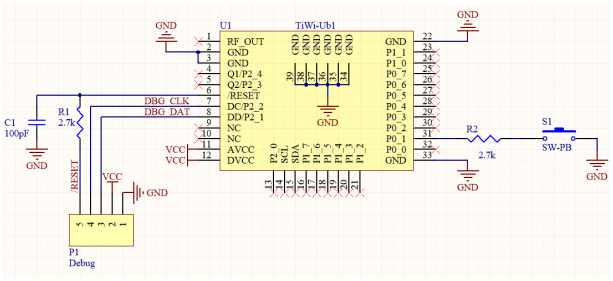


Figure 1.3.31: Proof-of-concept prototype schematic of inhalator monitoring device using TiWi-Ub1 module

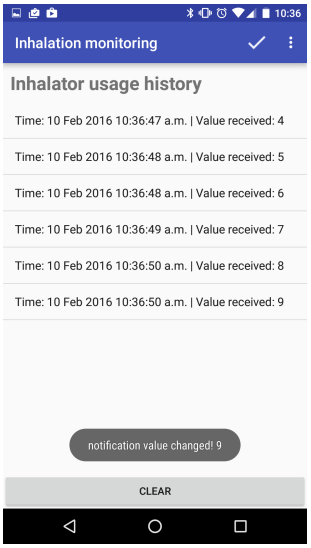


Figure 1.3.32: Inhalator monitoring device Android application screenshot.

1.3.4.6 Palm prosthesis dynamics monitoring in therapy (in co-operation with Wide.Tech)

During this project a wearable sensor system for prosthesis motion tracking in order to improve rehabilitation process was proposed. We developed a sensor system for prosthetics that will be able to track angle of a joint flexion/extension in real time. Acquired data will be used to monitor rehabilitation process and patient progress. For palm prosthesis flexion/extension angle calculation was used a network that consists of two 3-axial accelerometers and magnetometers

on-board. Special cases were designed and printed to ensure sensors, master board and battery protection (mechanical influence etc.) and calculation precision 1.3.33. Position of sensors can influence on received data precision – sensor nodes should be located on one straight line (one plane). Sensors should be attached to the prosthetic in a way to minimize free movement inside the printed cases. As part of the project a mobile application was developed to make calculations, analyze collected data from sensor nodes, store and visualize it. Cloud services were used to provide data storage and communication between patient and health specialists, so that patient will be using up-to-date rehabilitation recommendations and health specialist will be informed on actual status of patient's rehabilitation progress. In the scope of this project, prototype version uses Microsoft Azure cloud services 1.3.34. Proposed prototype version was created in collaboration with start-up company "Wide.Tech", participating in "Fabulous" EU acceleration program.

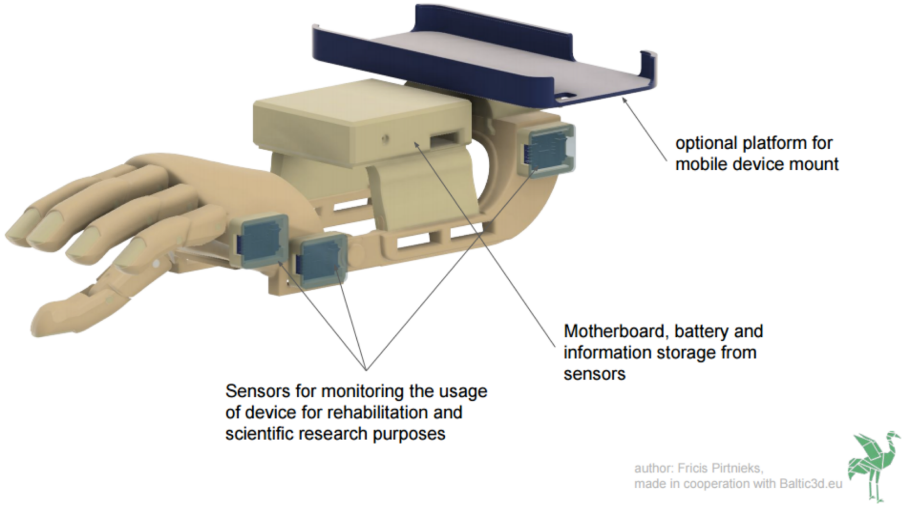


Figure 1.3.33: Palm prosthesis model with sensors attached.

1.3.4.7 Results

During the first two stages of the project a wearable sensor system is being developed, by both improving the architecture and developing new sensors for use in such system. In addition several of these results have been promoted to general public through dissemination efforts, and some have already attracted interested companies, thus resulting in contract research producing prototypes for specific applications.

Results of this research has also been presented in several conferences and research papers, as described in the main report, to which this report is appended to.

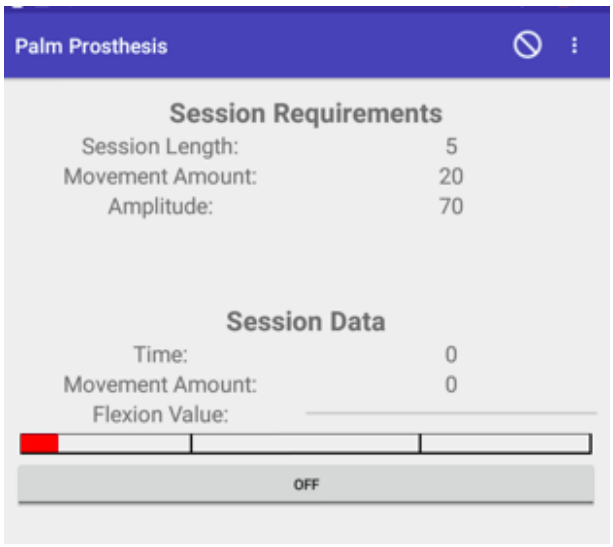


Figure 1.3.34: Prototype version of mobile application for palm prosthesis dynamics monitoring.

1.3.4.8 Discussion and future work

In the next stages of the MedWear project the main goal is to further improve the data gathering architecture and research the potential uses of added sensor data for patient monitoring, such potential for motion/activity data to improve usefulness of gathered ECG results, or EMG data to improve the overall activity monitoring results in rehabilitation.

Authors see system future research as a part of global human vital signs monitoring solution – developed device can be used in synergy with other systems to get more data and provide them to health specialist to better diagnostics. We are planning to create cloud based solution to analyze data from various patients with sensor nodes attached, in order to define risk factors of rehabilitation process.

Chapter 1.4

SmartCar - Intelligent transport systems

1.4.1 introduction

Every year, many car accidents due to driver fatigue and distraction occur around the world and cause many casualties and injuries. Various studies have suggested that around 20% of all road accidents are fatigue-related, up to 50% on certain roads[44][45].

To reduce the number of these incidents, generally improve road safety and improve driving convenience Advanced Driver Assistance Systems (ADAS) can be used.

In this project existing ADAS systems were reviewed and analyzed, and several research directions were selected in which potential improvements can be achieved through research in the confines of this project. Also to test and validate the results of this research a test platform (self-driving car) with an innovative control platform was developed which will be used not only for running real-life tests, but also for validating the devices in actual competition environment in Grand Cooperative Driving Challenge (GCDC).

Below sections describe the current state-of-the art, our vehicle control/test platform architecture and specific research in ADAS systems.

1.4.2 Background and state of the art

Because many road incidents are a cause of driver error or drowsiness a significant part of ADAS systems is dedicated to driver monitoring and providing additional information to the driver, that normally is not accessible.

Some of the current systems learn driver patterns and can detect when a driver is becoming drowsy. Various technologies can be used to try to detect driver drowsiness[46] and generally assist driver in driving more safely and comfortably, thus improving road traffic safety through intelligent transport system technologies:

- **Steering Pattern Monitoring.** Primarily uses steering input from electric power steering system;
- **Vehicle Position in Lane Monitoring.** Uses lane monitoring camera;
- **Driver Eye/Face Monitoring.** Requires a camera watching the driver's face[47];
- **Physiological Measurement.** Requires body sensors for measure parameters like brain activity, heart rate, skin conductance, muscle activity;
- **Thermal imaging.** Requires thermographic camera to increase a driver's perception and seeing distance in darkness or poor weather.

1.4.2.1 Steering behavior

Features of steering wheel movement can be divided into time, frequency and state space domains. This assignment follows the first processing step of computing frame level descriptors, independent of feature characteristics of the second, contour describing, functional based processing step.

Time domain features: Within the time domain the following features can be extracted: regression descriptors (e.g. regression slope, intercept, maximum of regression error), class distribution measures (e.g. number of values within steering angle bin 0.0-0.1), peak amplitudes and distances (e.g. mean distance of peaks; maximum of peak amplitude), entropy, zero crossing distances and slope (e.g. maximum of distance between consecutive zero crossings; mean velocity of steering angle in zero crossings)

Frequency domain features: It is possible to obtain a set of frequency-related parameters. Usually a signal processing is applied to obtain signal spectrum which is evaluated in a frequency regions from which the descriptors are extracted. Frequency area is generally defined depending on the application. It is important how many samples will be used to calculate the spectrum of the signal and necessary descriptors will be extracted. Spectral features include: DCT coefficients, statistical moments, the standard deviation, the zero crossing frequency, etc.

State space: To extract the nonlinear properties of the steering angle signal, a three-dimensional state space (steering wheel position, steering wheel velocity, steering wheel acceleration) is computed, and a phase space is reconstructed. The geometrical properties of the resulting attractor figures can be described by trajectory based descriptor contours (angle between consecutive trajectory parts, distance to centroid of attractor, length of trajectory leg). The temporal information of the contours can be captured by computing functionals.

1.4.2.2 Vehicle Position in Lane Monitoring

The goal of the image processing is to extract information about the position of the vehicle with respect to the road from the video image. Two major processes are usually implemented: the pre-processing process and then the lane detection process. The goal of pre-processing is to remove image noise and make the images sharper. The goal of the lane detection is to detect the desired lane of the vehicle in order to obtain the look-ahead distance and the lane angle. This process is based on the real-time data of video sequences taken from a vehicle driving on the road. The processing steps of the example lane detection algorithm could be: image segmentation, edge detection, Hough Transform, and lane tracking.

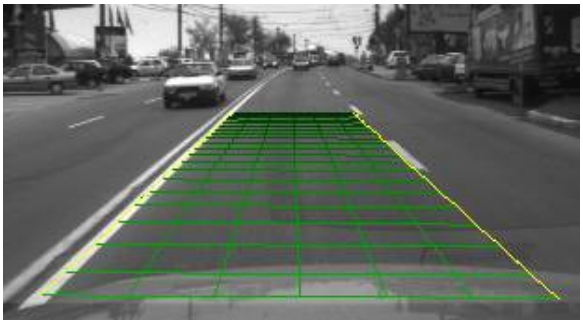


Figure 1.4.1: Lane tracking

1.4.2.3 Driver eye/face monitoring

Driver face monitoring: The driver face monitoring system is a real-time system that investigates driver physical and mental condition based on processing of driver face images. Driver status can be detected from eyelids closure, blinking, gaze direction, yawning and head movement. This system will alarm in hypo-vigilance states such as drowsiness, fatigue and distraction. The systems based on the driver face monitoring can be divided into two general categories. In first category, driver fatigue and distraction is detected only by processing of eye region. There are many researches based on this approach. The main reason of this large amount of researches is that main symptoms of fatigue and distraction appear in driver eyes. Moreover, processing of eye region instead of total face region has less computational complexity.

In the other category, symptoms of fatigue and distraction are detected not only from eyes, but also from other regions of face and head. In these approaches, not only activities of eyes are considered, but also other symptoms such as yawning and head orientation is extracted.

Software is the most important part of driver face monitoring system and is divided into two main parts: image processing algorithms and decision-making algorithms.

The main goals of image processing algorithms include preprocessing, detection and tracking of face, eyes and other facial components, and extraction of appropriate symptom from facial images.

After extraction of appropriate symptom from images, decision-making algorithms determine the level of driver alertness based on extracted symptoms. Finally, an appropriate output is generated for the system. In most of the driver face monitoring systems, face detection is the first part of image processing operations. In systems that fatigue and distraction is detected based on processing of facial region, face detection is considered as very important part of the system. Also in most of methods based on processing of eye region, due to difficulty of eye detection directly, face is detected at first and then eyes are detected. The most important problems of face detection are[48]:

- In-plane face rotation,
- Out-of-plane face rotation,
- The presence or the absence of makeup, beard and glasses,
- Mental conditions (happiness, crying, and etc.),
- Illumination conditions,
- Covering part of the face with an object,
- Real-time processing.

Face detection methods can be divided into two general categories[48]: (1) feature-based and (2) learning-based methods. Learning-based methods usually more robust than feature-based methods, but they often take more computational resources. However, these methods can achieve a detection rate about 80-90% or higher in laboratory conditions, but both of them usually fail in real conditions, especially in night light.

In feature-based methods, the main assumption is that face in the image can be detected based on some simple features, independent of ambient light, face rotation and pose. These methods are usually used for detection of one face in image. In noisy image or the environment with low illuminations, these algorithms have low accuracy[48].

Learning-based methods deal with face detection using a number of training samples. These methods benefit from statistical models and machine learning algorithms. Generally, learning-based methods have less error rates in face detection, but these methods usually have more computational complexity. Viola et al.[49] presented an algorithm for object detection, which uses very simple features named Haar-like features. In this algorithm, many Haar-like features are extracted from the image, and a number of effective features are selected using AdaBoost algorithm, and then these features are processed in a hierarchical structure similar to the decision tree. Due to the simple extracted features and selection of the best features, this algorithm is relatively fast and robust.

Driver eye monitoring: In all driver face monitoring systems, eye region is always processed for symptom extraction, because the most important symptoms are related to the eyes activity. Therefore, eye detection is required before processing of eye region. Eye detection methods can be divided into three general categories: (1) methods based on imaging in IR spectrum, (2) feature-based methods and (3) other methods.

Seeing Machines builds image-processing technology that tracks the movement of a person's eyes, face, head, and facial expressions. They've developed an ADAS technology which monitors driver fatigue and distraction events in real time, enacting an intervention strategy that improves driver and environmental safety.

Figure 1.4.2: Eye tracking device developed by Seeing Machines

1.4.2.4 Relationship between Physiological Signals and Drowsiness

Bioelectricity is generated on the cell level and acts as the charge flow on human surface. The electrical charges on the skin off the chest are mainly caused by the depolarization of heart muscles during each heartbeat cycle. In each cycle, nerve excitability is triggered by sinoatrial node, and then spreads through atrium, intrinsic conduction pathways and ventricles. As a result, it causes the change of action potential in cells manifested as the form of tiny rises and falls of potential on body surface. The electrical activity of heartbeat cycle is adjusted rhythmically by central and peripheral nervous system. Fatigue causes changes in spontaneous rhythmic activity, breathing, cardiovascular reflex activity, blinking, nodding, etc. The comprehensive regulation of these changes by the central nerve system will finally cause changes in the physiological signals.

A few physiological signals of drivers have been found to be good drowsiness indicators. It is generally believed that fatigue is the behavior of the central nervous system. When stress response of organs occurs during fatigue, cardiovascular nervous system will adjust accordingly. Therefore onset of fatigue causes changes in the bioelectrical signals, such as the electrocardiogram (ECG),

a recording of electrical signals produced by the electro-dynamic functioning of the heart.

The methods for drowsiness identification based on ECG signal include Heart Rate (HR) analysis, Heart Rate Variability (HRV) analysis and amplitude analysis of T wave.

In period 1 and 2 of our research project we chose to use drivers face monitoring, because of the ease of use, non-intrusiveness and high reliability, based on the previous research results and commercial success.

1.4.2.5 Thermal imaging

An automotive night vision system uses a thermographic camera to increase a driver's perception and seeing distance in darkness or poor weather beyond the reach of the vehicle's headlights. Such systems are offered as optional equipment on certain premium vehicles. It is crucial to detect persons or animals on or on the side of the road before they appear in the drivers viewing range. This gives additional time for a driver to react and eliminate the accident. Many research groups have been working on this problem using methods like basic background subtraction, sliding window, keypoint detection and other approaches. In [50] researchers propose to use Maximally Stable Extremal Regions (MSER) to detect hot spots, verify detected hot spots using a Discrete Cosine Transform (DCT) based descriptor and a modified Random Naïve Bayes (RNB). Keypoint detection is prone to detect only few or no keypoints for low resolution objects. This leads to partial or missed detections. The sliding window approach is time consuming especially when many different object scale levels are considered. Background subtraction cannot be used with moving camera. Because of the low computational demand MSER method has been chosen for further research in the project. MSERs are the result of a blob detection method based on thresholding and connected component labeling[51].

1.4.3 Our solution

1.4.3.1 Introduction

During this project several innovative solutions in ADAS systems and ITS have been developed already and will be developed in the next periods.

To test and validate these solutions a decision was made to develop a test platform - self driving car capable of housing these technologies and reliably validating them.

Additionally this testing platform was developed with a goal of approbation of these technologies in i-GAME *(interoperable GCDC*[1] *AutoMation Experience)* which is a cooperative autonomous driving competition that will take place in Helmond, Netherlands on May 23-31 of 2016. Goal of the competition is to efficiently perform several typical scenarios for highway and for city by a group

[1] Grand Cooperative Driving Challenge

of vehicles, and the solutions developed within the project should improve the chances to reach this goal.

The testing platform consists of the physical car and mechanical control mechanisms, electronic actuators, feedback systems, data gathering, analysis and decision making framework, and sensing systems for both the external and internal environment, as described in the sections below.

1.4.3.2 Background - GCDC competition

The GCDC competition differs from other autonomous vehicle projects in sense that here vehicles participating in a maneuver communicate precise information about their state (ie. coordinates, speed, direction) to each other via network. For this purpose cars involved in the competition will be equipped with communication infrastructure of unified standard (ETSI[2] ITS G5 network stack). So every vehicle knows about others timely and has a possibility to plan maneuvers without unnecessary stress and without getting into situations where extreme braking or another extreme action is necessary while this communication standardization requirement is held. Another consequence from communication standardization: competition between different technical solutions from different vendors becomes possible.

In the GCDC-2016 competition there will be following scenarios evaluated: "Cooperation on highway", "Cooperative intersection" and "Emergency vehicle"[3].

In the GCDC competition scenarios should be performed efficiently not individually but cooperatively by the entire group of cars participating in the maneuver. Also in this competition there are additional requirement to provide also an automated lateral control of vehicles while in 2011 there was requirement to provide longitudinal control only.

The GCDC 2011[52] was the first competition to implement such a realistic, heterogeneous scenario. It was organized by the Netherlands Organization for Applied Scientific Research (TNO) in Helmond. Participating teams had to come up with strategies that were able to perform as good as possible without knowing the algorithms and technical equipment of other vehicles in the platoon. Control strategies had to cope with unexpected behavior of other vehicles, varying data quality, and sudden failure of communication, among others. Fig.1.4.3 shows one heat of the GCDC, illustrating the large variety of vehicles and technical solutions in the competition [53].

Cooperation among traffic participants plays an important role in everyday life to ensure traffic safety and traffic flow [54]. E.g., resigning one's right of way at a crossroads or allowing other vehicles to merge on one's own lane are frequent behaviors which are most beneficial to all traffic participants and which may even resolve critical situations. While human drivers are still superior to automated vehicles in many situations, machines are able to negotiate cooperative driving maneuvers significantly faster and with fewer misunderstandings

[2]European Telecommunications Standards Institute

[3]http://gcdc.net/en/

Figure 1.4.3: Grand Cooperative Driving Challenge of 2011

than humans.[53]

In the the first GCDC competition in 2011 a team from Latvia was among the participants[4]. In GCDC 2016 we also plan to participate and started to take part in semi-monthly webinars, meetings and other preparation activities for participants since last year. Our autonomous vehicle solution development is also in process.

1.4.3.3 The physical test platform

As a base vehicle for our *experimental autonomous vehicle laboratory* we are using a *Mazda 6* 2004 year's model (Fig.1.4.4) with several modifications (the dataflow of proposed solution is shown in Fig.1.4.7).

Figure 1.4.4: An experimental vehicle (Mazda 6)

[4]http://www.edi.lv/en/home/events/gcdc/

Acceleration, braking and steering control

We have added electronically controllable actuators for steering, braking and acceleration control.

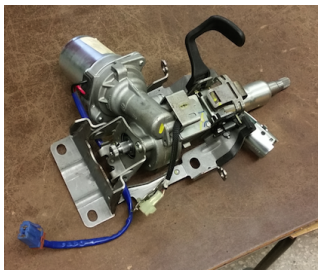


Figure 1.4.5: Steering motor

Steering: Our base vehicle has a hydraulic power assisted steering system that is not controllable *by-wire*. To add *steer-by-wire* functionality we are using a electrical steering assistant device from *Renault Clio II* (Fig.1.4.5) that is assembled into vehicle steering column. Motor is driven by voltage $\pm 0..12V$ that is controlled by PWM signal from *IBT-2* board based on 2 BTS7960 half-bridge chips.

Steering motor device has also a torsion sensor that could measure turning force of steering wheel by driver. We will use this feature to detect a driver's intrusion in control process when vehicle *Control center* (host computing device network) system must switch to manual mode.

For steering wheel angle detection we use CAN digital angle sensor from *Renault* that communicates via CAN interface (Fig.1.4.6).

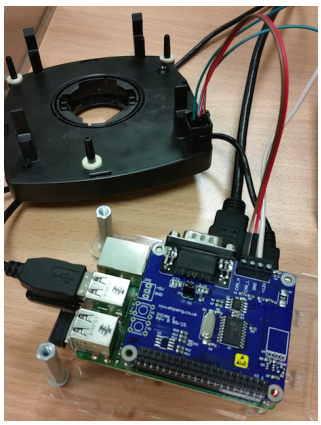


Figure 1.4.6: Steering angle sensor

Braking: To add *brake-by-wire* functionality to the vehicle we are using same motor from *Renault Clio* but in this case as motor angle sensor we use multiturn potentiometer *POT2218M2-5*.

To detect touching brake pedal by driver, we use push button.

Acceleration: To provide *gas-by-wire* control we use the same solution as in 2011. Our base vehicle allows to control gas pedal throttle by two voltages $\pm 0..3.3V$ that are provided by gas pedal. We will emulate those voltages by DAC in the *Interface circuit* (Fig.1.4.9).

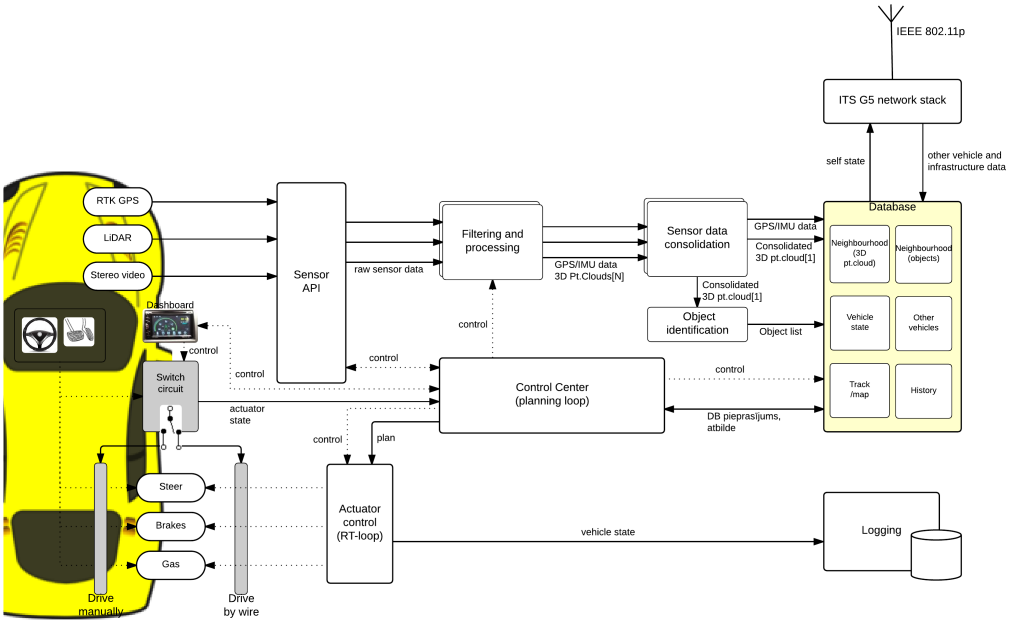


Figure 1.4.7: Dataflow in autonomous vehicle solution

Communication with other vehicles (V2V) and infrastructure (I2V)

For communication with other vehicles as well as with road infrastructure we are *PC Engines APU platform* board with *Voyage linux* and *BTP/GeoNetworking/IEEE 802.11p* protocol stack.

Module placement on the vehicle

Roof module: A control system's module that is fortified on the vehicle roof contains a following setup:

- Communication infrastructure - IEEE 802.11p antenna and PC Engines APU device (1.4.3.3) - all in a waterproof containers;
- 12V Red and Green lights;
- Two GPS antennas;
- Velodyne HDL-32S LiDAR device;
- Stereo Video camera *Point Grey Bumblebee XB3 1.3 MP Color*.

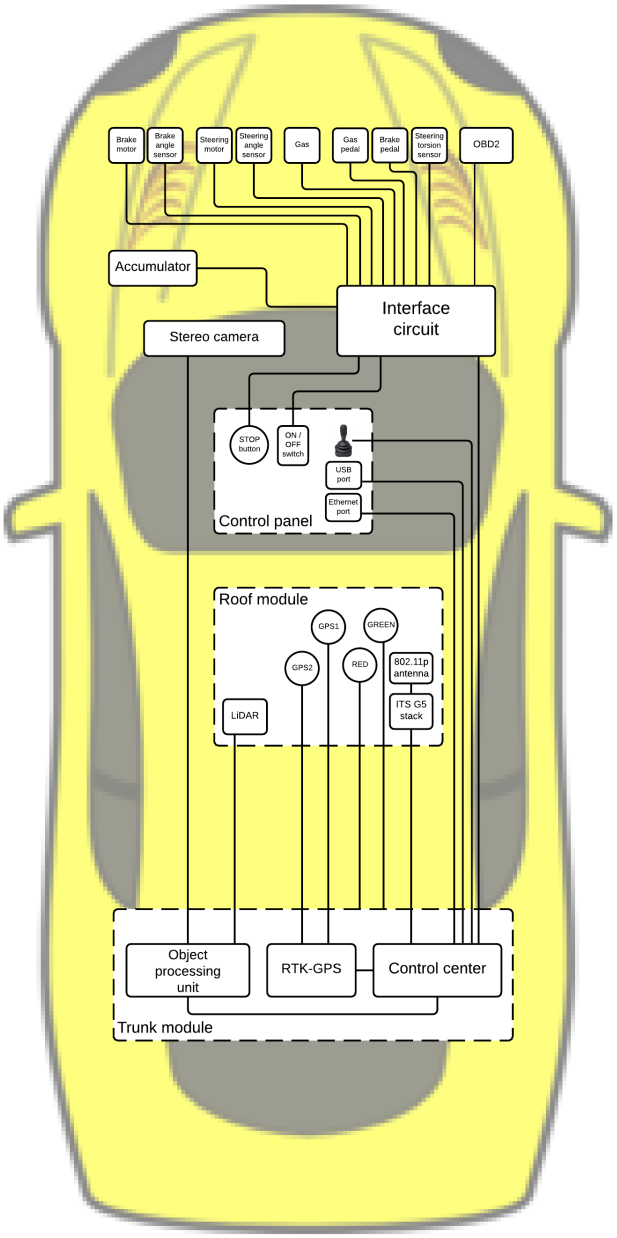


Figure 1.4.8: Module placement on the vehicle

.

Interface circuit: This circuit (Fig.1.4.9) provides an interface between devices and the *Control center*. It contains terminal ports for interfacing to other modules:

- To all sensor/actuator devices - steering motor, braking motor, steering angle sensor, braking motor angle sensor, gas, brake pedal, gas pedal, steering torsion sensor, OBD2[5];

[5]On-Board Diagnostics II

- To vehicle power supply (accumulator);
- To control panel - ON/OFF switch, STOP button.
- To roof module - Red light, Green light;
- To trunk module - 12V power supply link, Ethernet and CAN-bus lines.

Interface circuit's heart is a *Raspberry Pi Model B* computer with extension board with two 12-bit DAC (MCP4822) and two 12-bit ADC (MCP3202). The DAC is used for gas control but ADS is used to read sensor data. To increase number of ADC channels we use analog 8-channel multiplexer SN74HC4051. To control high-current parts of the circuit in motors and ON/OFF switch we use *Songle SRD-05VDC-SL-C* relays. To measure power consumptions on steering and braking motors and overall system we use current sensors ACS712.

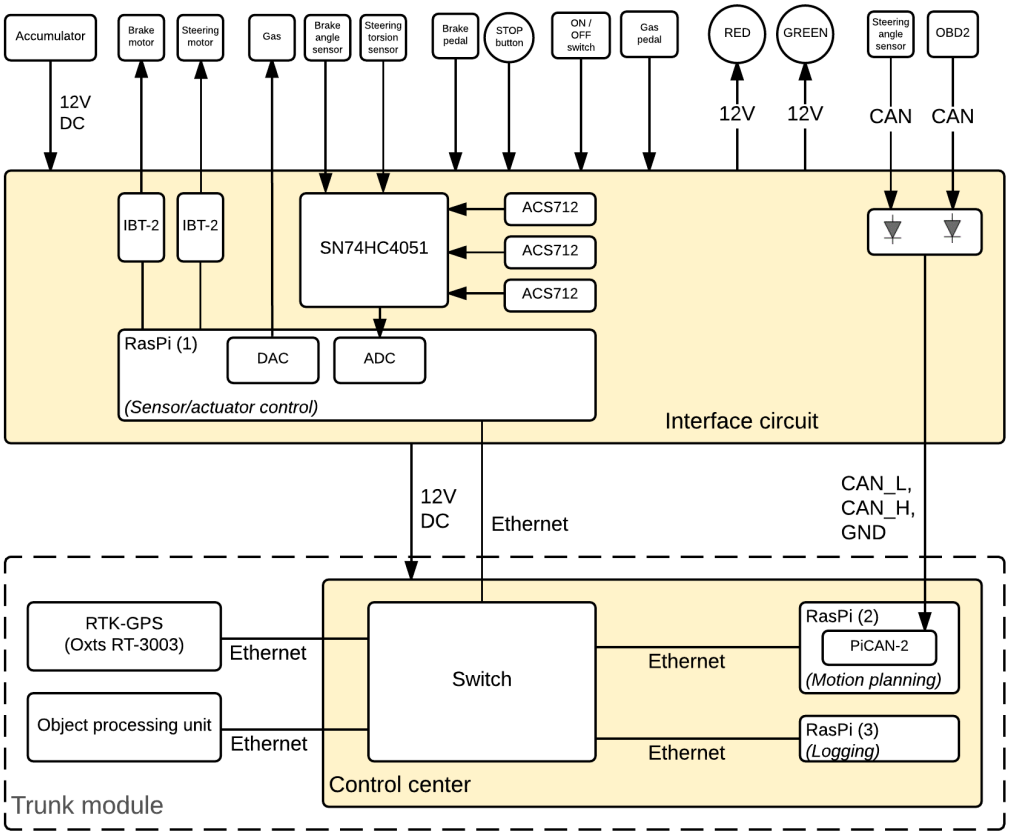


Figure 1.4.9: Interface circuit and Control center modules

.

Trunk module: In the trunk of the vehicle is a platform on which following devices are placed:

- *Control module* that consists of two *Raspberry Pi 2 Model B* devices (1.for motion control, 2.for logging) communicating via Ethernet;

- *Oxts RT-3003* RTK-GPS device;
- *Object processing unit* device for 3D point-cloud processing and detection of objects around the vehicle.
- *100Mbps Ethernet switch*;

Control panel: Near driver's right hand a panel is provided with following controls that are used most often:

- An ON/OFF switch that turns on or off all vehicle control system added by us to the base vehicle. If the switch is in state *OFF* then vehicle is in manual mode just like a base vehicle. If switch is set to *ON* then control center computers are booted up, sensors are initialized and data gathering from them and showing up on *software dashboard* is started, actuators are initialized. Vehicle control center is set to *Manual* mode and *roof module* lights are set to - RED:ON, GREEN:OFF. It is possible now to drive the vehicle by steer/pedals, to monitor sensor state and perform further switch between modes using buttons on the *software dashboard*.
- An USB port to connect a *software dashboard* device with application that is used to monitor system state and switch between vehicle modes and GCDC competition scenarios.
- The software dashboard is an Android-based application that displays vehicle parameters (position, speed, vehicle angle, steering wheel angle etc.) and provides control buttons for switching between driving modes or GCDC scenarios. Each button has an enable/disable condition. For instance, it should not be possible to enter GCDC *Scenario 1* when the vehicle is not in platoon with detected front vehicle or if relative position, direction or speed is differing from vehicle in the front more than by limits allowed for the scenario.

  Following modes are distinguished in the system and could be controlled using the *software dashboard:*

  – "Manual" - manual mode that is default option when control system is turned on. Roof lights - RED:ON, GREEN:OFF;
  – "Auto" - the same as "Manual" except a possibility to control the vehicle using joystick resided on the *Control panel*. Roof lights - RED:ON, GREEN:OFF;
  – "ACC" - the vehicle will drive at the same speed as a vehicle in the front. Steering remains manual. Roof lights - RED:ON, GREEN:OFF;
  – "CACC" - the vehicle will drive in a platoon at the same speed as a vehicle in the front and with a lateral position as of the vehicle in the front. Roof lights - RED:ON, GREEN:OFF. When the vehicle is in the "CACC" mode, it becomes possible to start GCDC Scenario 1 or 3;

– "V-CACC" - the vehicle will drive in a "virtual" platoon. Roof lights - RED:ON, GREEN:OFF. When the vehicle is in the "V-CACC" mode, it becomes possible to start GCDC Scenario 2;
– "Scenario1","Scenario2" or "Scenario3" - the vehicle will enter a sequence of a chosen scenario. Roof lights - RED:OFF, GREEN:ON. When execution of the scenario is finished successfully or interrupted by *STOP button* or touching a steer or pedal, the vehicle control center is set to *Manual* mode and roof lights are set to - RED:ON, GREEN:OFF;

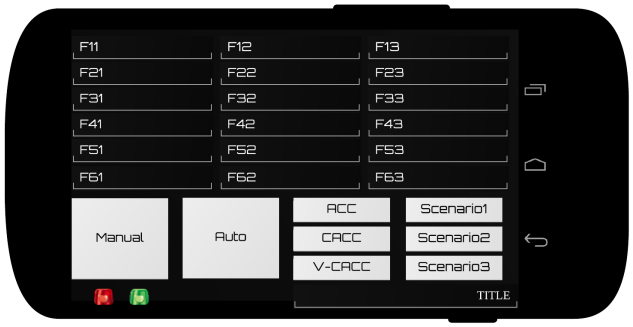


Figure 1.4.10: Software dashboard

.

- A STOP button interrupts any GCDC scenario or any other vehicle mode that is not "Manual" to "Manual". Roof lights are set to - RED:ON, GREEN:OFF.

  When the vehicle is in GCDC Scenario 1, 2 or 3, touching a gas or brake pedal or steering wheel leads to the same effect as pressing the "STOP button": the scenario is interrupted and vehicle mode is set to "Manual". Roof lights are set to - RED:ON, GREEN:OFF.
- Ethernet port for possibility to connect to Control center hardware from driver's seat for diagnostic and debugging purposes.

1.4.3.4 Localization and position detection

We use a combined inertial and satellite-based navigation system *Oxts RT-3003* to obtain location and position of our vehicle. Device can be augmented optionally by ground reference stations. Device uses RTK[6] correction that provides precise position of the vehicle with $1cm$ accuracy at $100Hz$. Data from *RT-3003* is sent via Ethernet cable.

1.4.3.5 Object detection around the vehicle

Although it is not required by GCDC rules we are adding some sensors for detection of objects surrounding our experimental vehicle. It would be more safe

[6]Real-Time Kinematics

to have such stand-by ability when GPS and/or communication device work is interrupted unexpectedly.

We will use a distinct computing device (see *Object processing unit* in Fig.1.4.8) to perform in real-time detection of objects of different size in 3D point-cloud of surrounding of the vehicle.

We plan to use following data sources for 3D point-cloud:

LiDAR: We are using a *Velodyne HDL-32S* device to get cloud of distances to objects around the vehicle. Device has 32 infrared 905nm lasers that measure a distance to objects from $1m$ to $70m$ with precision $2.5cm$ at $10Hz$ speed. Data from *HDL-32S* device is sent via Ethernet cable. Obtained data packets are processed and point-cloud is extracted from them and sent fgor processing to *Object processing unit*.

Stereo Video: For detection of objects in the front of the vehicle we use double video camera *Point Grey Bumblebee XB3 1.3 MP Color*. Video shoots are captured simultaneously by two cameras located in in parallel at a fixed distance from each other. Then shoots are sent to *Object processing unit* where distances to points in shoots are calculated (for more information on video processing algorithms used see 1.4.3.8).

Object processing unit: 3D point-clouds from LiDAR and Stereo Video devices is then received by *Object processing unit* (Fig.1.4.8). It is basically a powerful PC that is continuously performing detection of objects of different sizes in point-clouds that are received from different sources. Device also performs tracking of those objects - recognizing that object in a shoot is the same object as in previous shoot. Output of this is list of objects that is then sent to vehicle *Control center*.

For additional information on software development for Stereo Vision and object recognition in 3D point-clouds please refer to Section 1.4.3.8.

1.4.3.6 Signal and image processing for environment evaluation for intelligent transport systems

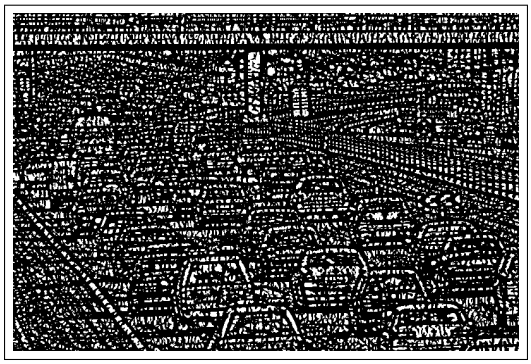
Introduction

Additional tasks that are usually performed by ADAS, like lane detection or traffic sign recognition, require fast and reliable low-level signal processing. In this project, as one of the activities, we have developed line extraction filter (Line Non-Halo Complex Matched Filter[55], L-NH-CMF), published in [56]. The main contribution of L-NH-CMF is fast, angle-invariant line-only detection with background-gradient-independent responses that contain additional line-direction information. The proposed L-NH-CMF can be used to extract object contours while ignoring gradients and edges, producing fewer seed pixels for the segmentation stage, c.f. Fig. 1.4.11 and Fig. 1.4.12.

Figure 1.4.11: L-NH-CMF improvements over NH-CMF for object contour extraction - black pixels in images (b) and (c) represent filter responses

(a) input image

(b) NH-CMF line-like object detection



(c) L-NH-CMF line only detection

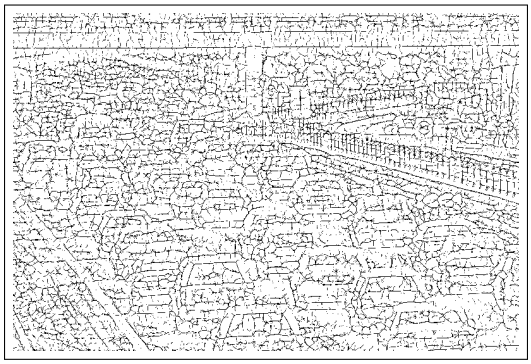
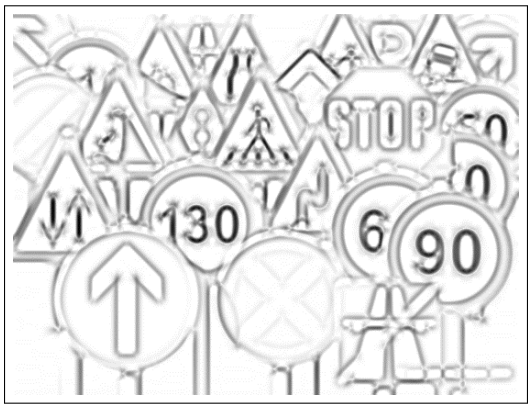


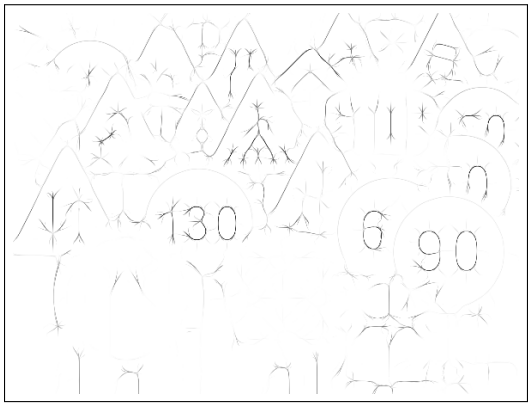
Figure 1.4.12: L-NH-CMF improvements over NH-CMF for traffic sign line extraction - black pixels in images (b) and (c) represent filter responses

(a) input image

(b) NH-CMF line-like object detection



(c) L-NH-CMF line only detection



In the report of this research programme's first period, the theoretical reasons for L-NH-CMF algorithm were already discussed, therefore, only the main idea behind the algorithm will be mentioned here. The L-NH-CMF is based on detection of left and right edge profiles of the line object separately, providing left and right edge detector responses s_L and s_R, as illustrated in Fig. 1.4.13:

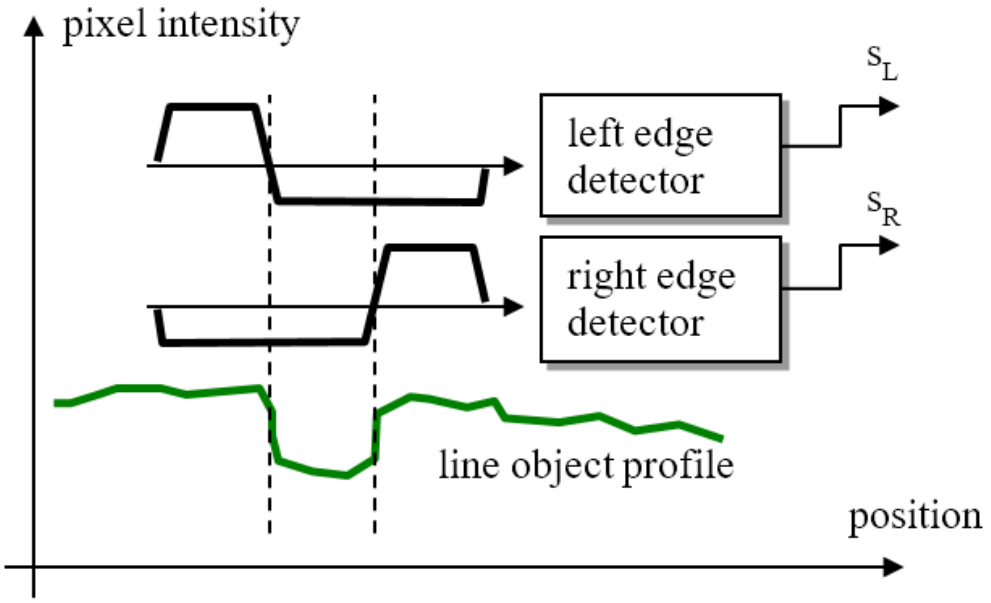


Figure 1.4.13: The principle of operation of left and right edge detectors of L-NH-CMF, shown using 1D pixel profile model

Then, the response of L-NH-CMF is calculated using $s = R[2 \cdot (s_L + s_R) - R[s_L] - R[s_R]]$, where $\forall x \in \mathbb{R} : R[x] \equiv \frac{x+|x|}{2}$. L-NH-CMF responses for different angles are combined using 2Φ algorithm, described in [57].

The rationale behind this algorithm is that when background gradient changes, the sum $s_L + s_R \approx const$, but in case of non-line object $|s_L| >> |s_R|$ or $|s_L| << |s_R|$, making $2 \cdot (s_L + s_R) - R[s_L] - R[s_R]$ negative or positive but small, see [56] for more details.

The developed filter might aid automated lane recognition for ACC systems, because road markings consists of line objects. Additionally, it might help detecting and recognizing road sign texts and in image segmentation tasks.

1.4.3.7 Driver Eye/Face Monitoring

An embedded driver face recognition system is presented in this section. Currently the system includes the facial recognition functionality only, however the eye-based monitoring of the driver condition can be added to the functionality in the future. The system already employs an image acquisition, face detection and eye localization modules which will be the same initial steps for the eye-based driver monitoring system.

Brief overview of the driver face recognition system

An embedded face recognition system based on the Raspberry Pi single-board computer is described in this document. Face recognition system consists of

face detection and face localization using Haar feature-based cascade classifier. Face features are extracted using weighted Local Binary Pattern algorithm. Developed system performs one full face analysis in 110 ms. Comparison of two biometric samples is performed in 2 ms. The proposed embedded face recognition system was tested on FERET database and achieved the accuracy of CMC at rank-1: 99.33% and EER: 1%.

Real time human identification systems are important for security in many areas, including automotive. Human identification can be performed by analysing its biometric information, such as fingerprints, face, iris, palm prints, palm veins etc. However, for fast and convenient person recognition, still the most suitable biometric parameter is facial information. Moreover, the facial recognition system can easily be enhanced to monitor the condition of the driver based on the facial/eye images.

Identification of humans by using facial biometrics is still challenging task, due to the variable illumination, changing facial expressions according to mood changes, head orientation and pose. Over the years, various face detection algorithms have been developed. Some face recognition methods analyse the geometric features of facial images, such as location and distance between nose, eyes, and mouth [58], [59]. However, these methods are sensitive to the changes in illumination and facial expression. Because of this drawback, most of the face recognition systems try to extract some holistic features from the original face images for matching. By using holistic methods face is recognized using descriptions based on the entire image rather than on local features of the face [60]. Many subspace learning based holistic feature extraction methods have been developed, including Eigenfaces, Fisherfaces [61], 2D PCA and others. In this document we describe the holistic method called local binary pattern (LBP) [62]. The introduced system can be used not only for driver recognition but in other applications as-well.

The general block-scheme of the system is introduced in Figure 1.4.14. A more detailed description of the system is given in the upcoming sections.

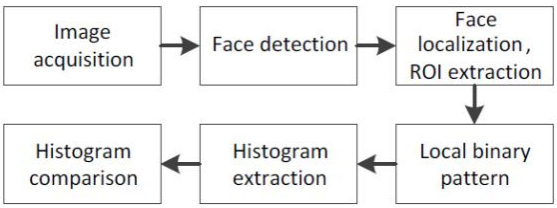


Figure 1.4.14: The block-scheme of the introduced driver face recognition system

Automatic face recognition algorithm

The developed algorithm for face recognition can be divided into several steps. The sequence of steps of the algorithm used for face recognition is shown in Figure 1.4.14. The first step is to acquire the image. Next, face detection has

to be performed, to find whether the face appears in the captured image or not. The next step is to locate the position of the face in the image. Face detection and face localization is performed using the Haar feature-based cascade classifier [63] [62]. The rectangular features needed for Haar classifier are computed using an intermediate representation for the image which is called an integral image [63].

The integral image at location x; y contains the sum of the pixels above and to the left of x; y, inclusive:

$$i(x, y) = \sum_{x' \leq x, y' \leq y} f(x', y'), \tag{1.4.1}$$

where $i(x, y)$ is the integral image and $f(x, y)$ is the original image. Using the following pair of recurrences:

$$s(x, y) = s(x, y - 1) + f(x, y); i(x, y) = i(x - 1, y) + s(x, y), \tag{1.4.2}$$

where $s(x, y)$ is the cumulative row sum, $s(x, -1) = 0$, and $i(-1; y) = 0$ the integral image can be computed in one pass over the original image [63].

Using the integral image any rectangular sum can be computed by referencing four array locations (shown in Figure 1.4.15). Difference between two rectangular sums can be computed in eight references. Since the two-rectangle features defined above involve adjacent rectangular sums they can be computed in six array references, eight in the case of the three- rectangle features, and nine for four-rectangle features [63].

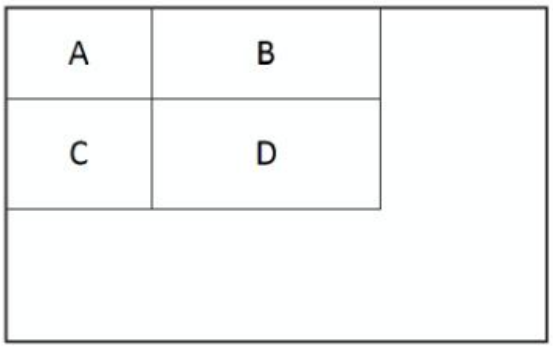


Figure 1.4.15: The sum of the pixels within rectangle D can be computed with four array references. (A+B +C +D)+(A) - ((A + B) + (A + C))

If the image contains a face, the algorithm returns a rectangle with coordinates where face was found. However, it is not the final region of interest (ROI) that we use. To calculate the necessary ROI, we use the coordinates of a rectangle and recalculate the ROI position. We use FERET face image database for algorithm evaluation that contains many frontal face images with their respective eye coordinates. Using the obtained rectangle from the Haar classifier and the known eye positions, we developed a method to statistically determine

where in the image the eyes are most likely to be located. We use this method to determine the approximate location of the person's eyes to calculate the ROI based of the distance between the eyes d. Distance to each of the sides is calculated by coefficients shown in Figure 1.4.16. However, our approach does not localize the position of the eyes and therefore cannot correct the rotations of the face.

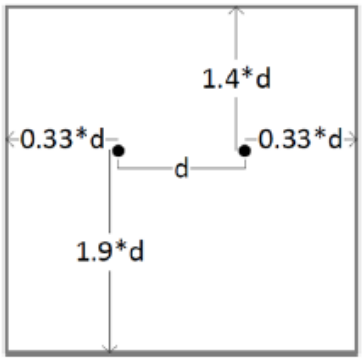


Figure 1.4.16: Coefficients for face region extraction

After the ROI is found, we perform LBP transformation. LBP labels each pixel by thresholding the neighborhood pixels with central pixel value and represents the result as a binary number. An example of the labeling procedure for 3x3 region is shown in Figure 1.4.17. Binary value usually is written as a decimal number and stored in the center pixel position of the output image. After LBP transform a histogram of labels is used as a descriptor of the image. LBP operator can be performed by using different amount of pixels P on the sampling circle and different radius R from the central pixel.

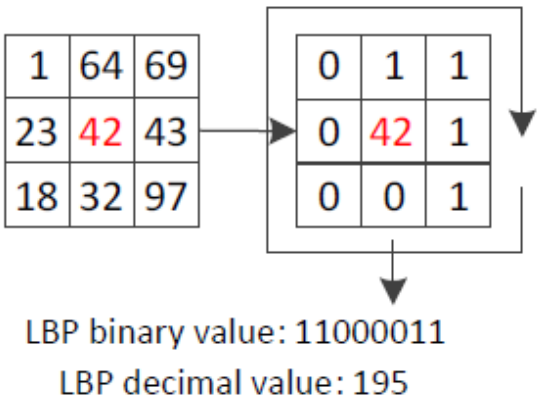


Figure 1.4.17: LBP example of 3x3 neighborhood, (P = 8, R = 1)

In order to save the spatial information about the object, LBP transformed image is divided into several sub-regions. A spatially enhanced histogram is calculated by concatenating the region histograms into a single feature histogram.

An example of LBP transform and face feature histogram forming with 6x6 regions of LBP image is shown in Figure 1.4.18. Feature histogram is used as a descriptor for each of the given images. To compare two feature histograms many methods can be used. For example, correlation of the histograms, Chi-Square, histogram intersection, Bhattacharyya distance and other methods. In our algorithm we chose histogram intersection for two histogram comparison.

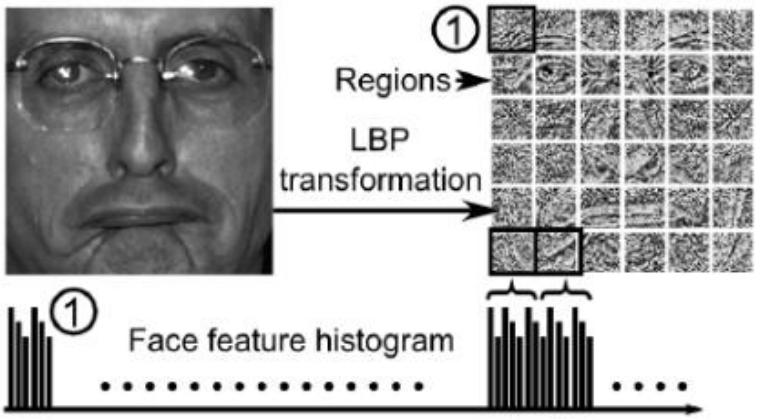


Figure 1.4.18: Example of LBP transform and formation of face feature histogram from 6x6 regions

Hardware details: face recognition system

In the heart of the embedded face recognition system is a Raspberry Pi single-board computer, which controls all of the peripherals. The board has a 700MHz ARM CPU and 512 MB of RAM. For image acquisition we use a USB web camera with a 320x240 resolution. To visualize the information and provide user friendly identification procedure, we use mini LCD. Block diagram of the system is shown in Figure 1.4.19. Raspberry Pi runs the Raspbian OS which is a Linux operating system derived from Debian. Our program is written in C++ and uses the OpenCV library for image acquisition and face detection.

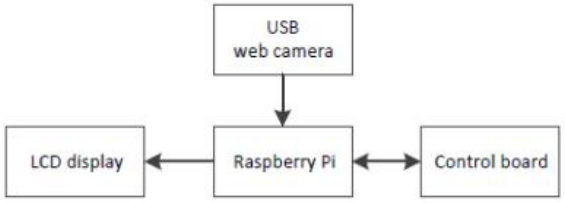


Figure 1.4.19: Hardware block scheme

Experimental setup and results

For evaluation of performance we use two parameters. One is equal error rate (ERR). The rate at which both accept and reject errors are equal. The value

of the EER can be easily obtained from the receiver operating characteristic (ROC) curve. The EER is a quick way to compare the accuracy of devices with different ROC curves. The lower the rate, the lower the error of the system. The other one is cumulative match characteristic (CMC), a CMC curve plots the probability of identification against the rank. The greater the rate at rank 1, the higher the accuracy of the system. To evaluate the performance of the system we use publicly available FERET database. Probe set and gallery set form the FERET database are combined. The combined data contains data from 992 persons, 2 samples from each person, for a total of 1986 images. 25% of persons are used for the algorithm training and the other 75% are used for testing. During the training process, the optimal values for three parameters were found by performing the exhaustive search. The first parameter is the resolution of the ROI, the second one is the radius for the LBP transform and the third one is the amount of regions for the histogram calculation. Optimal parameters are ROI width = 110, R = 4, m = 5. Using the FERET database we achieved the results of CMC rank-1: 99.33% and EER: 1%. CMC curve is shown in Figure 1.4.20.

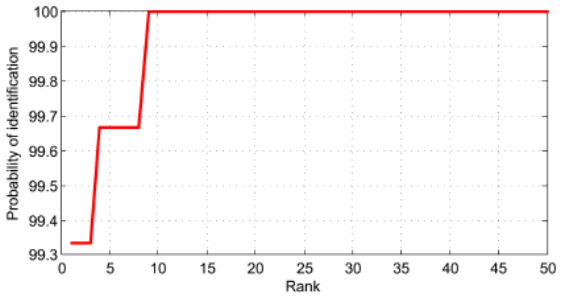


Figure 1.4.20: Cumulative Match Characteristic

1.4.3.8 Stereo Vision System

Introduction

As a basis for Stereo Vision System, we use the software that was developed by H. Grinbergs et al. in EDI during the Fourth Stage of previous State Reseach programme's Project Nr.2 "Innovative signal processing technologies for smart and effective electronic system development" (January-December, 2013)[64]. His group's software was designed to use for Adaptive Cruise Control (ACC) and in Automatic Driver Assistance System (ADAS) and was approbated in several near-real condition scenarios related to application of 3D computer vision and collection and processing of sensor data.

However, the operating conditions of this software were different than in the currently developing system, see Fig. 1.4.21.

Since the results achieved previously were satisfying, e.g.:

- ability to detect and track general objects with specified width-height-

Figure 1.4.21: ESV-1 during approbation in 2013[65], from left to right: camera set, processing unit, monitor and GUI

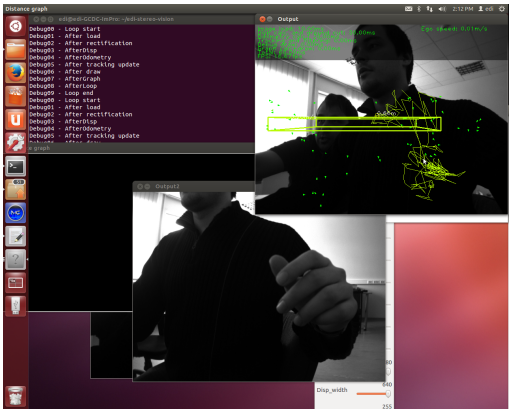


Figure 1.4.22: The GUI elements of ESV-1 when adapted for Linux

length parameter range (including non-moving objects like trees, or pedestrians),

- ability to track objects within 50m range (using industrially-calibrated Bumblebee XB3 stereo camera[66] with 24cm of camera distance),
- performance of 40fps using 2.9GHz Intel i5 (70fps on 4.4GHz),

it was decided to keep the existing principle of software operation and improve the software for a broader usage scope. Next subsections provide more detailed description of these principles and applied improvement strategy.

For convenience, the software provided by H. Grinbergs et al. will be further referred to as edi-stereo-vision-1 (ESV-1).

Characteristics of ESV-1

The ESV-1 was developed on Windows operating system, and was OpenCV-GUI-based standalone program, see Fig. 1.4.22.

Despite the great achieved parameters, there were few inconvenient moments in the initially provided software, for example:

1. it had to be adapted for usage in Linux environment (read next subsection for more details),

2. it had too many GUI elements that can't fit on one less-than-HD1080 screen, and in some usage scenarios are unnecessary at all,

3. its initial used parameters were hardcoded in the C++ code and packaged in a command line options,

4. some of its parameters could later be changed by using OpenCV GUI slider elements,

5. there were too many keyboard shortcuts to toggle separate features and/or GUI windows,

6. it had multithreading support for calculation but sometimes crashed on exit (known issue).

To further develop the system and make it compatible with the proposed test platform architecture or other ADAS or ACC systems it was necessary to port ESV-1 for usage in general Linux-based, possibly embedded, solutions as a modular library and/or TCP/IP service. Usage of a separate libraries/services for image preprocessing, point cloud acquisition, point cloud processing and object detection/tracking, will provide possibilities to conveniently study impact of each module's architecture on overall performance of a system. Additional benefit of such approach includes reusability of modules in other software projects.

Preparation of ESV-1

Introduction to Linux and GNU Coding Standards In the end of second period of the project, the peculiarities of Linux systems and principles of writing and packaging OpenSource software (The GNU Coding Standard[67]) were practically studied on examples by configuration of different (mostly Debian-based) Linux distributions like Ubuntu / Kali / LinuxMint / Kubuntu and so on so that the resulting system can be properly formulated for use with other systems and potentially released as OpenSource software. The practices of software build configuration and building from sources were extensively tested on a broad range of popularly distributed libraries that include OpenCV (CMake-based configuration) for image processing and wxWidgets (GNU-configure script-based configuration) for GUI building. As a result, in January of 2016 the ESV-1 was ported and ran on Ubuntu Linux using Bumblebee XB3 camera set, see Fig. 1.4.22.

Taking into account a new emerging trend of FPGA-based System-on-Chip (SoC) embedded systems with ARM Hard-Processor System (HPS) and to understand the portability issues regarding to this class of systems, additional effort was put into:

- learning the cross-compilation procedure for different types of architecture (mostly ARMv7l using Linaro gnueabihf crosscompiler chain),

- learning the cross-compilation of Linux kernel (v4.4.0 ARCH=arm) from scratch,

- learning the cross-compilation of Linux kernel objects (.ko) from scratch and adapting kernel parameters in case of compilation errors,
- running and testing custom built systems on an Terrasic DE1-SoC device[7].

This was done in order to research the possibilities of offloading part of the image processing onto FPGA for system's faster performance. Regarding this last activity, there are still many unsolved issues like generation of proper Device Tree Blob for Linux kernel to support memory-mapped drivers (for easier access to FPGA) and HPS-FPGA communication using DMA/IRQ to minimize computing effort of HPS that will be solved in the next periods of the project.

Particularly, we plan to study the possibilities of implementing the calculation of disparity map and point cloud in SoC system. The algorithm will be separated into functional blocks that will be divided between FPGA and HPS. The purpose of this is to find the optimal distribution of these blocks in SoC to get higher computation efficiency and to implement optimal communications protocol between HPS and FPGA for the respective amount of data.

Program File Organization for ESV-1 It was decided to keep all dependency libraries in a separate/local directories to exclude any interaction with the installed libraries of the same soname with other subversion and make project easily movable to other Linux platform with the same architecture without rebuilding the libraries. The following libraries were gathered:

- OpenCV-2.4.5 with FFMPEG support for video acquisition,
- Libelas[8][68] for 3-dimensional point-cloud extraction from a pair of input images (left and right),
- Libviso2[9][69] for visual odometry, i.e. own movement estimation from acquired 3-dimensional point-cloud,

and compiled in static library format (.a) for easy inclusion to the ESV-1.

Proposed Stereo Vision Software

For convenience, the new edi-sterero-vision software will be referred to as ESV-2.

Proposed Software Alterations It was decided to make the new software library-based and TCP/IP service-based chronologically in the same order. To achieve this, main components of the system, each performing only a single complete task, were identified:

- video acquisition module (either from Bumblebee XB3 camera, either from a prerecorded scenario dataset (for reproducible tests)),

[7] http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=836
[8] http://www.cvlibs.net/software/libelas/
[9] http://www.cvlibs.net/software/libviso/

- point cloud calculation module, based on libelas for 3-dimensional point-cloud extraction from a pair of input images (left and right),
- ego movement estimation and compensation module, based on libviso2 performing visual odometry on acquired 3-dimensional point-cloud,
- object detection, labeling and tracking module,
- visualization module (GUI).

Each component is then identified in currently almost monolithic code and isolated to be put in a separate library. Since there are two final realizations of the software (libraries and TCP/IP services), each has its own architecture features. So far, only the library-based software has a defined specification.

Architecture of library-based ESV-2 The data structure used for description of point cloud was defined in ESV-1 as follows:

```
typedef vector< vector< vector< double > > > PointCloud;
```

Here, the first coordinate of PointCloud 3D array is X, the second is Y, relative to the front viewpoint of camera (shown in Fig. 1.4.23), the third coordinate sets up a parameter number for point. There are 5 point parameters:

1. x coordinate in 3D space (absolute, in cm),
2. y coordinate in 3D space (absolute, in cm),
3. z coordinate in 3D space (absolute, in cm),
4. point validity (0.0 if point is invalid, 1.0 if valid),
5. disparity value (acquired using libelas).

Considering best C++ practices and C++11 standard[70], it was proposed to edit the point cloud description as follows:

```
const struct PointCloudPointStruct {
        bool is_valid,
        auto x_abs_cm,
        auto y_abs_cm,
        auto z_abs_cm,
        double disparity
} PointCloudPointStruct_default = {.is_valid = false, .
    x_abs = 0.0_cm, .y_abs = 0.0_cm, z_abs = 0.0_cm, .
    disparity = 0.0};
typedef struct PointCloudPointStruct
    PointCloudPointStruct;
```

Then, the point cloud can be defined simpler as:

```
typedef vector< vector< PointCloudPointStruct > >
    PointCloud;
```

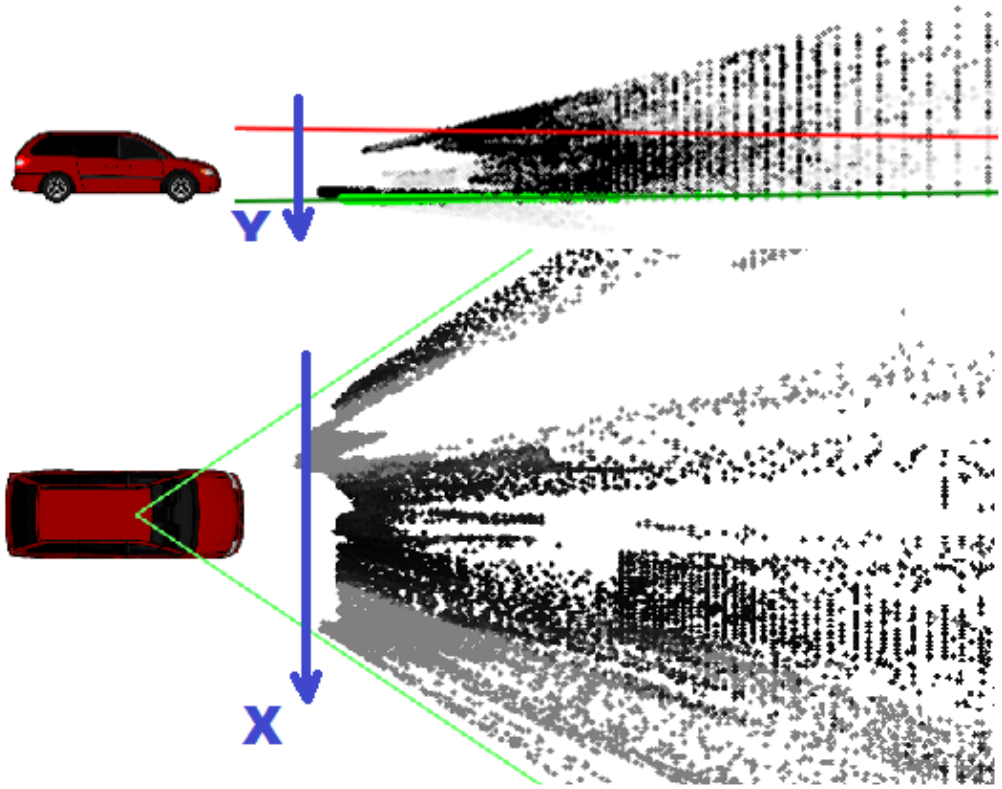


Figure 1.4.23: Coordinate System of ESV-1

The specification defines the following functions that are called inside a loop of the main data gathering and processing process:

For **point cloud calculation module**:

```
int calculate_point_cloud (PointCloud *p_point_cloud,
    const ImgType *p_img_left, const ImgType *p_img_right,
     const SetupStruct *p_setup_struct);
```

where:

- *p_point_cloud* is pointer to a previously defined structure describing the 3D point cloud (OUT),
- *p_img_left* and *p_img_right* is either of type $cv :: Mat$, either $vector < vector < unsigned\ char >>$, depicting captured images (IN), or NULL to omit image processing,
- *p_setup_struct* is a pointer to a structure of parameters, NULL to skip setting/updating the parameters (IN),
- the RETURN VALUE is 0 in case of success and has error code otherwise.

The function performs the following operations in a given order:

1. updating the *SetupStruct* parameters, iff requested,
2. calculating point cloud data from images, and writing point cloud data at $*p\_point\_cloud$, iff requested.

For **object detection, labeling and tracking module**:

```
int track_objects (const PointCloud& point_cloud,
    ObjectList& object_list, ImgType *p_image, const
    SetupStruct *p_setup_struct);
```

where:

- *point_cloud* is a reference to previously calculated 3D point cloud (IN),
- $object_list$ is a reference to continuously updating list of labeled tracked objects (BIDIR),
- *p_image* is a pointer where to write annotated output image (for GUI), NULL to skip this operation (OUT),
- the RETURN VALUE is 0 in case of success and has error code otherwise.

Note that *track_objects* is not expected to keep information about tracked objects - it reuses its own previously returned value as input. The *object_list* with minimum expression is defined as follows:

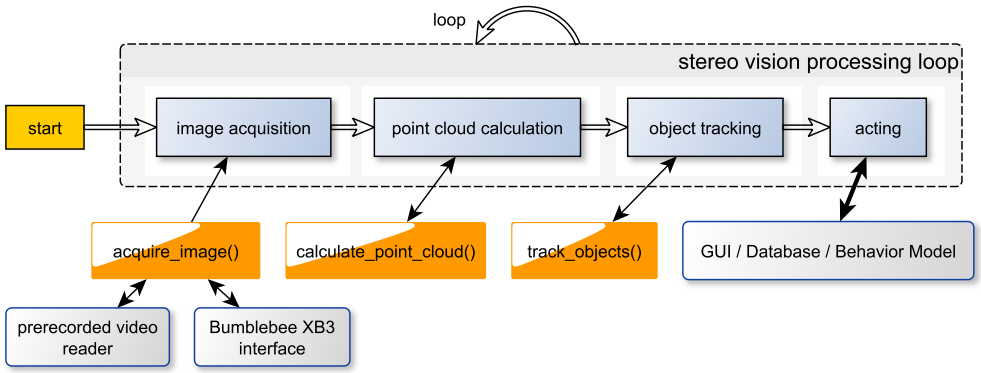


Figure 1.4.24: The simplified model of ESV-2 library version

```
const struct ObjectDescriptionStruct {
    uint64_t id,
    auto x_abs _cm,
    auto y_abs _cm,
    auto z_abs _cm,
    auto dx_abs _cmps,
    auto dy_abs _cmps,
    auto dz_abs _cmps
} ObjectDescriptionStruct_default = {.id = 0, .x_abs =
    0.0_cm, .y_abs = 0.0_cm, z_abs = 0.0_cm, dx_abs = 0.0
    _cmps, dy_abs = 0.0_cmps, dz_abs = 0.0_cmps};
typedef struct ObjectDescriptionStruct
    ObjectDescriptionStruct;
typedef vector< ObjectDescriptionStruct > ObjectList;
```

For convenience and increased precision of tracking, the *ObjectDescriptionStruct* will be later complemented with object size, object class (car / pedestrian / tree / etc), and Kalman filter parameters.

The simplified block diagram of ESV-2 library version (incl. test loop) is shown in Fig. 1.4.24 and depicts all previously described elements.

1.4.4 Results

During the first two periods of the project a an ADAS/ITS test and validation platform has been developed, capable of gathering data from different kinds of sensors and systems, processing this data and generating feedback or deciding on direct car control accordingly.

Most of the research has gone into three categories - new signal and image processing methods for applications in ITS, driver eye/face monitoring for feedback, and object detection and tracking using stereovision camera.

Results from this research can be fed into the test platform and together with the results from other research directions, that have just been recently started

(such as positioning/location with Lidar and GPS/IMU) used for validation of complex ADAS system verification which will be done in GCDC competition in the next period of the project.

1.4.5 Discussion and future work

In the future periods of the projects we plan to achieve several goals:

- To approbate the ADAS/ITS self driving test plafrom in GCDC competition;
- To implement the stereo-vision algorithms on SoC;
- To augment/extend the senses of the driver, e.g. make infrared vision spectrum visible;
- Develop innovative systems for driver monitoring and feedback.

Bibliography

[1] Joseph Polastre, Robert Szewczyk, and David Culler. Telos: enabling ultra-low power wireless research. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 364–369. IEEE, 2005.

[2] *MICAz*.

[3] Prabal Dutta and David Culler. Epic: An open mote platform for application-driven design. In *Proceedings of the 7th international conference on Information processing in sensor networks*, pages 547–548. IEEE Computer Society, 2008.

[4] Advanticsys. *XM1000 datasheet*.

[5] Girts Strazdins, Atis Elsts, and Leo Selavo. Mansos: easy to use, portable and resource efficient operating system for networked embedded devices. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 427–428. ACM, 2010.

[6] Atis Elsts, Janis Judvaitis, and Leo Selavo. Seal: A domain-specific language for novice wireless sensor network programmers. In *Software Engineering and Advanced Applications (SEAA), 2013 39th EUROMICRO Conference on*, pages 220–227. IEEE, 2013.

[7] Andreas Willig Adam Wolisz Vlado Handziski, Andreas Köpke. Twist: A scalable and reconfigurable wireless sensor network testbed for indoor deployments. In *REALMAN '06 Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 63–70. ACM, 2006.

[8] Harward. *Tmote SKY datasheet*.

[9] Vlado Handziski, Joseph Polastre, Jan-Hinrich Hauer, and Cory Sharp. Flexible hardware abstraction of the ti msp430 microcontroller in tinyos. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 277–278. ACM, 2004.

[10] Patrick Swieskowski Geoffrey Werner-Allen and Matt Welsh. Motelab: a wireless sensor network testbed. In *REALMAN '06 Proceedings of the 2nd*

international workshop on Multi-hop ad hoc networks: from theory to reality, pages 483 – 488. IEEE, 2005.

[11] Mun Choon Chan Manjunath Doddavenkatappa and Ananda A.L. Indriya: A low-cost, 3d wireless sensor network testbed. In *Testbeds and Research Infrastructure. Development of Networks and Communities*, pages pp 302–316. Springer Berlin Heidelberg, 2012.

[12] Arduino. *Arduino homepage*.

[13] Django framework.

[14] ACE. *ACE homepage*.

[15] Google charts api.

[16] Rinalds Ruskuls and Leo Selavo. Edimote: a flexible sensor node prototyping and profiling tool. In *Real-World Wireless Sensor Networks*, pages 194–197. Springer, 2010.

[17] Paolo Bonato. Wearable sensors and systems. *Engineering in Medicine and Biology Magazine, IEEE*, 29(3):25–36, 2010.

[18] Rune Fensli, Einar Gunnarson, and Torstein Gundersen. A wearable ecg-recording system for continuous arrhythmia monitoring in a wireless tele-home-care situation. In *Computer-Based Medical Systems, 2005. Proceedings. 18th IEEE Symposium on*, pages 407–412. IEEE, 2005.

[19] Winston H Wu, Alex AT Bui, Maxim A Batalin, Duo Liu, and William J Kaiser. Incremental diagnosis method for intelligent wearable sensor systems. *Information Technology in Biomedicine, IEEE Transactions on*, 11(5):553–562, 2007.

[20] Peter Leijdekkers and Valérie Gay. A self-test to detect a heart attack using a mobile phone and wearable sensors. In *Computer-Based Medical Systems, 2008. CBMS'08. 21st IEEE International Symposium on*, pages 93–98. IEEE, 2008.

[21] Alexandros Pantelopoulos and Nikolaos G Bourbakis. A survey on wearable sensor-based systems for health monitoring and prognosis. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(1):1–12, 2010.

[22] Shyamal Patel, Hyung Park, Paolo Bonato, Leighton Chan, and Mary Rodgers. A review of wearable sensors and systems with application in rehabilitation. *Journal of neuroengineering and rehabilitation*, 9(1):21, 2012.

[23] A Hermanis, K Nesenbergs, R Cacurs, and M Greitans. Wearable posture monitoring system with biofeedback via smartphone. *Journal of Medical and Bioengineering Vol*, 2(1), 2013.

[24] Yu-Jin Hong, Ig-Jae Kim, Sang Chul Ahn, and Hyoung-Gon Kim. Activity recognition using wearable sensors for elder care. In *Future Generation Communication and Networking, 2008. FGCN'08. Second International Conference on*, volume 2, pages 302–305. IEEE, 2008.

[25] Louis Atallah, Benny Lo, Guang-Zhong Yang, and Frank Siegemund. Wirelessly accessible sensor populations (wasp) for elderly care monitoring. In *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on*, pages 2–7. IEEE, 2008.

[26] Jay Chen, Karric Kwong, Dennis Chang, Jerry Luk, and Ruzena Bajcsy. Wearable sensors for reliable fall detection. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 3551–3554. IEEE, 2006.

[27] R Ganea, Anisoara Paraschiv-Ionescu, Arash Salarian, Christophe Bula, Estelle Martin, Stephane Rochat, Constanze Hoskovec, and Chantal Piot-Ziegler. Kinematics and dynamic complexity of postural transitions in frail elderly subjects. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 6117–6120. IEEE, 2007.

[28] Miikka Ermes, Juha Parkka, Jani Mantyjarvi, and Ilkka Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *Information Technology in Biomedicine, IEEE Transactions on*, 12(1):20–26, 2008.

[29] Florian Michahelles and Bernt Schiele. Sensing and monitoring professional skiers. *Pervasive Computing, IEEE*, 4(3):40–45, 2005.

[30] Weijun Tao, Tao Liu, Rencheng Zheng, and Hutian Feng. Gait analysis using wearable sensors. *Sensors*, 12(2):2255–2283, 2012.

[31] Christian Peter, Eric Ebert, and Helmut Beikirch. A wearable multi-sensor system for mobile acquisition of emotion-related physiological data. In *Affective Computing and Intelligent Interaction*, pages 691–698. Springer, 2005.

[32] Flavia Sparacino. The museum wearable: Real-time sensor-driven understanding of visitors' interests for personalized visually-augmented museum experiences. 2002.

[33] Konrad Lorincz, Bor-rong Chen, Geoffrey Werner Challen, Atanu Roy Chowdhury, Shyamal Patel, Paolo Bonato, Matt Welsh, et al. Mercury: a wearable sensor network platform for high-fidelity motion analysis. In *SenSys*, volume 9, pages 183–196, 2009.

[34] Krisjanis Nesenbergs and Leo Selavo. Smart textiles for wearable sensor networks: Review and early lessons. In *Medical Measurements and Applications (MeMeA), 2015 IEEE International Symposium on*, pages 402–406. IEEE, 2015.

[35] Robert Broadbent, Sidney Melamed, and Robert G Minton. Conductive nylon substrates and method of producing them, April 15 1975. US Patent 3,877,965.

[36] A. Hermanis, R. Cacurs, K. Nesenbergs, and M. Greitans. Efficient real-time data acquisition of wired sensor network with line topology. In *Open Systems (ICOS), 2013 IEEE Conference on*, pages 133–138, Dec 2013.

[37] Akiba Kevin Townsend, Carles Cufí and Robert Davidson. *Getting Started with Bluetooth Low Energy*. O'Reilly Media, 2014.

[38] A. Hermanis and K. Nesenbergs. Grid shaped accelerometer network for surface shape recognition. In *Electronics Conference (BEC), 2012 13th Biennial Baltic*, pages 203–206, Oct 2012.

[39] A. Hermanis, K. Nesenbergs, R. Cacurs, and M. Greitans. Wearable Posture Monitoring System with Biofeedback via Smartphone. *Journal of Medical and Bioengineering*, 2(1):40–44, 2013.

[40] M. D. Shuster and S. D. Oh. Three-axis attitude determination from vector observations. *Journal of Guidance and Control*, 4:70–77, 1981.

[41] A. M. Sabatini. Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing. *IEEE Transactions on Biomedical Engineering*, 53(7):1346–1356, July 2006.

[42] Roberto Valenti, Ivan Dryanovski, and Jizhong Xiao. Keeping a Good Attitude: A Quaternion-Based Orientation Filter for IMUs and MARGs. *Sensors*, 15:19302–19330, 2015.

[43] Y. M. Chi, T. P. Jung, and G. Cauwenberghs. Dry-contact and noncontact biopotential electrodes: Methodological review. *IEEE Reviews in Biomedical Engineering*, 3:106–119, 2010.

[44] Royal Society for the Prevention of Accidents. (2001, February) DRIVER FATIGUE AND ROAD ACCIDENTS A LITERATURE REVIEW and POSITION PAPER. [Online]. Available: `http://www.rospa.com/rospaweb/docs/advice-services/road-safety/drivers/fatigue-litreview.pdf`.

[45] COMPASS. (2015, November) 4.1.03. Driver Drowsiness Detection System for Cars. [Online]. Available: `http://81.47.175.201/compass/index.php?option=com_content&view=article&id=506:413-driver-drowsiness-detection-system-for-cars&catid=22:smart-cars`.

[46] Robert Bosch LLC Frank Sgambati. (2012, January) Driver Drowsiness Detection. [Online]. Available: `http://www.sae.org/events/gim/presentations/2012/sgambati.pdf`.

[47] Dimitri J Walger, Toby P Breckon, Anna Gaszczak, and Thomas Popham. A comparison of features for regression-based driver head pose estimation under varying illumination conditions. In *Computational Intelligence for Multimedia Understanding (IWCIM), 2014 International Workshop on*, pages 1–5. IEEE, 2014.

[48] Ming-Hsuan Yang, David J Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(1):34–58, 2002.

[49] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[50] Michael Teutsch, Thomas Muller, Marco Huber, and Jurgen Beyerer. Low resolution person detection with a moving thermal infrared camera by hot spot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 209–216, 2014.

[51] Jiri Matas, Ondrej Chum, Martin Urban, and Tomás Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 22(10):761–767, 2004.

[52] Ellen van Nunen, RJAE Kwakkernaat, Jeroen Ploeg, and Bart D Netten. Cooperative competition for future mobility. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3):1018–1025, 2012.

[53] Andreas Geiger, Martin Lauer, Frank Moosmann, Benjamin Ranft, Holger Rapp, Christoph Stiller, and Jens Ziegler. Team annieway's entry to the 2011 grand cooperative driving challenge. *Intelligent Transportation Systems, IEEE Transactions on*, 13(3):1008–1017, 2012.

[54] Steven E Shladover. Cooperative (rather than autonomous) vehicle-highway automation systems. *Intelligent Transportation Systems Magazine, IEEE*, 1(1):10–19, 2009.

[55] M. Pudzs, M. Greitans, and R. Fuksis. Complex 2d matched filtering without halo artifacts. In *The 18th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 1–4, June 2011.

[56] M. Pudzs, R. Fuksis, A. Mucenieks, and M. Greitans. Complex matched filter for line detection. In *The 9th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pages 93–97, Sept 2015.

[57] M. Pudzs and M. Greitans. Rotation-invariant object detection using complex matched filters and second order vector fields. In *The Proceedings of the 22nd European Signal Processing Conference (EUSIPCO)*, pages 1312–1316, Sept 2014.

[58] Peter N. Belhumeur, João P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):711–720, July 1997.

[59] Keun-Chang Kwak and W. Pedrycz. Face recognition using an enhanced independent component analysis approach. *Neural Networks, IEEE Transactions on*, 18(2):530–541, March 2007.

[60] Jian Yang, David Zhang, Alejandro F. Frangi, and Jing-yu Yang. Two-dimensional pca: A new approach to appearance-based face representation and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(1):131–137, January 2004.

[61] Stefano Arca, Paola Campadelli, and Raffaella Lanzarotti. *Audio- and Video-Based Biometric Person Authentication: 4th International Conference, AVBPA 2003 Guildford, UK, June 9–11, 2003 Proceedings*, chapter A Face Recognition System Based on Local Feature Analysis, pages 182–189. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[62] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–900–I–903 vol.1, 2002.

[63] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511–I–518 vol.1, 2001.

[64] EDI. (2013, December) State Research Programme "Innovative signal processing technologies for smart and effective electronic system development", Project No. 2. [Online]. Available: `http://www.edi.lv/en/projects/state-research-p-projects/project-nr2/`.

[65] H. Grinbergs. (2013, December) Tracking 3D Objects using passive stereovision on moving platform (In Latvian). [Online]. Available: `http://www.edi.lv/media/uploads/UserFiles/VPP_PrNr2_5_Grinbergs.ppt`.

[66] Anonymous. (2016) Bumblebee XB3 1.3 MP Color FireWire 1394b. [Online]. Available: `https://www.ptgrey.com/bumblebee-xb3-stereo-vision-13-mp-color-firewire-1394b-38mm-sony-icx445-camera`.

[67] Anonymous. (2016, January) GNU Coding Standards. [Online]. Available: `https://www.gnu.org/prep/standards/`.

[68] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Asian Conference on Computer Vision (ACCV)*, 2010.

[69] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV)*, 2011.

[70] Bjarne Stroustrup. *The C++ programming language*. Pearson Education, 2013.

Appendix 1.1
Pielikums 1.1

Acceleration and Magnetic Sensor Network for Shape Sensing

IEEE Snsors Journal, Volume: 16, Issue: 5