TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology

IDN70LT

Lauri Vihman 130422IABMM

ORGANIZATIONAL SENSORS: COMPARATIVE ANALYSIS, ENHANCEMENT, EXPERIMENT

Master's thesis

Supervisor: Innar Liiv
PhD
Associate Professor

Tallinn 2016

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

IDN70LT

Lauri Vihman 130422IABMM

ORGANISATSIOONI SENSORID: VÕRDLEV ANALÜÜS, EDASIARENDUS, EKSPERIMENT

magistritöö

Juhendaja: Innar Liiv
PhD
dotsent

Tallinn 2016

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Lauri Vihman

09.05.2016

# Abstract

Organizational monitoring of human face-to-face contacts is much needed information for multiple research purposes. Current thesis analyses the different technical approaches and develops nRF51822 prototype solution for collecting contact data using Personal Sensing devices and utilizing radio signals. A cost-effective nRF51822 development stack is introduced and key-points in developing the prototype solution are described. Experiment with the developed solution is conducted and results analysed.

All the developed software — including device firmware application, data converter and graphical representation utility of detected contacts for PC – is freely downloadable and available for free public use as-is. The corresponding chapters are also describing used hardware and used method to develop and download the device firmware application into a wearable nRF51822 device.

Author does not claim the solution to be a finished product, but rather a product in proof of concept stage. Contact detection using proximity and distance estimation leaves much to manual interpretation because of specific circumstances and radio signal fluctuations. In author's opinion the solution can be much improved in further works.

This thesis is written in English and is 65 pages long, including 6 chapters, 22 figures and 9 tables.

Annotatsioon

Organisatsiooni sensorid: võrdlev analüüs, edasiarendus, eksperiment.

Organisatsioonis toimuvate näost-näkku inimkontaktide tuvastamine sensorite abil on vajalik informatsioon erinevate uuringute tarbeks. Käesolevas töös analüüsitakse erinevaid tehnilisi võimalusi ja arendatakse prototüüplahendus, mis suudab isiklike kantavate sensorite abil inimkontakte tuvastada ja koguda. Tutvustatakse ka kuluefektiivset püsivara arendamise vahendite komplekti nRF51822 mikrokiibi tarbeks ja võtmepunkte, millega loodud lahenduse arendamise juures kokku puututi. Arendatud lahenduse abil viiakse läbi eksperiment, mille tulemusi analüüsitakse.

Kogu käesoleva töö raames arendatud tarkvara, kaasa arvatud nRF51822 mikrokiibi püsivara rakendus, andmete teisendaja ja graafiliselt esitamise vahendid PC jaoks, on vabalt alla laaditavad ja autori poolt jagatud tasuta avalikku kasutusse. Vastavates peatükkides on kirjeldatud kasutatud viis rakenduse arendamiseks ja laadimiseks nRF51822 seadmetele.

Autor ei väida, et lahendus on valmistoode, vaid pigem kontseptsiooni tõestamise staadiumis produkt. Kontaktide tuvastamine raadiolainete põhjal ja distantsi hindamise abil on suuresti sõltuvuses konkreetsetest situatsioonidest ja vajab käsitsi tõlgendamist. Autori hinnangul omab pakutav lahendus potentsiaali edasisteks jätkuarendusteks ja parendamiseks.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 65 leheküljel, 6 peatükki, 22 joonist, 9 tabelit.

List of abbreviations and terms

Balun	A device that converts between balanced and unbalanced impedance.
BLE	Bluetooth Low Energy aka Bluetooth Smart (wireless personal area networking)
DFU	Device Firmware Update
GAP	Generic Access Profile (Bluetooth)
GATT	Generic Attribute Profile (Bluetooth)
EWMA	Exponentially Weighted Moving Average
IC	Integrated Circuit
IoT	Internet of Things
ISP	In-System Programming
MIT	Massachusetts Institute of Technology
MMC	Multi Media Card
PC	Personal Computer
P2P	Peer to peer
PCB	Printed Circuit Board
OTA	Over The Air
QFN	Quad Flat No-leads (IC package).
RAM	Random Access Memory
RFID	Radio Frequency Identification
RSSI	Received Signal Strength Indicator
SDC	Secure Digital Card
SDK	System Development Kit
SoC	System on Chip
SoftDevice	Nordic Semiconductors’ binary library
UART	Universal Asynchronous Receiver/Transmitter
WLCSP	Wafer-level Redistribution Chip-scale package (IC package)

Table of contents

List of figures

List of tables

1 Introduction

Contacts between subjects take place frequently in human organizations. The contacts can be planned, spontaneous or serendipitous, they can be formal or informal, take place in indoor or outdoor environment, but often they include face-to-face interaction and information is exchanged. In organizations a contact with face-to-face interaction fosters knowledge transfer and innovation [1]. When being able to detect face-to-face contacts, find patterns and detect points of knowledge transfer, it may be possible to enhance and increase them. Information about contact events can help in researching multiple aspects of organizational behaviour. In addition the contact pattern data is needed for different areas, for example transfer of infectious diseases and collective opinion formation [2].

Collecting contact data via traditional methods like questionnaire, interview, observational or archival methods is slow, inaccurate and intrusive [2], [3]. Computer-aided interactions between human subjects in organizations can be logged and analysed, for example collaboration networks [3] or e-mails [4], but this is not the case with real-life face-to-face interactions where no intermediary exists. The face-to-face human interactions have the property of proximity. There exist multiple different meanings for the term “proximity” [1]. Current thesis uses the term in the meaning of geographical and spatial proximity, also referenced as territorial, local or physical proximity [1]. Temporary physical proximity between members of an organization does not always mean that a face-to-face contact takes place but the probability exists while without proximity it’s not probable. After collecting enough proximity data, contact patterns could be detected and specific interpretation of those can be done via specific dataset analysis.

The aim of current thesis is to analyse alternatives, enhance and implement a solution that could help organizations to detect, measure, collect and analyse contact patterns. Further works could fine-tune the results, discover different uses the collected data provides, and learn the shortcomings of such measurements.

The structure of current paper is divided into 6 chapters that give an overview of related earlier works, analyse different technical approaches, describe the key points in developing the solution, introduce the conducted experiment and results and make a conclusion.

The thesis is interdisciplinary – it includes aspects of hands-on Electronics, the resulting dataset can be used for Social Sciences purposes and it is written from the perspective of Information Technology. The work falls in the section of "Personal Sensing" and "Reality Mining" in Big Data Development research [5].

2 Related work

2.1 SocioPatterns and OpenBeacon

A collaboration unit of researchers and developers called SocioPatterns [6] has done experiments and published papers on measuring contact patterns of human groups. Mainly primary school students [7], high school students [8] and hospital contacts [9] were monitored. Also experiments were made on other dynamic contacts, such as volunteering ACM Hypertext 2009 conference attendees. Attendees were wearing radio badges that monitored their face-to-face proximity [6].

SocioPatterns have used RFID technology in their experiments that work at close range 1–1.5m [9]. The underlying technical platform for experiments have been the leveraged OpenBeacon active RFID platform [2]. OpenBeacon project is open Source and open hardware Active 2.4 GHz RFID reader system and open source and hardware active RFID tags [10]. The latest OpenBeacon new tag includes also BLE technology [11]. OpenBeacon devices are using Nordic Semiconductors microcontrollers. It is unknown from SocioPatterns experiments what specific hardware devices were used in the experiments and whether newer systems based on BLE and nRF51822 SoC were used in later experiments. OpenBeacon has multiple devices with different hardware that are shown in the following table (see Table 1).

Table 1 OpenBeacon products

Product	**Transceiver controller**	**Proximity**	**Accelerometer**	**Storage**	**Protocol**
OpenBeacon Sensor	nRF24L01 + PIC				2.4GHz RF
OpenBeacon USB 2	nRF24L01 + PIC	X	X		2.4GHz RF
OpenBeacon EasyReader	Not specified	X	stationary	MicroSD	2.4GHz RF
OpenBeacon tag NEW	nRF51822	X	X	8MB	BLE

During the time of writing this thesis none of the contact attempts with the representatives of the OpenBeacon project did not succeed in spite of multiple tries. The hardware and the software files of the OpenBeacon devices were found in GIT repository http://www.openbeacon.org/openbeacon.git and a newer nRF51822 based OpenBeacon tag in https://github.com/meriac/openbeacon-ng.

In the next paragraph we are going to have a brief look at the content found from the nRF51822 based OpenBeacon new tag repository. The files in the repository and source code itself are not much documented. The repository contains following firmware and following functionality can be assumed by source code without having OpenBeacon hardware:

- reader-ble – receive beacon packets and write data to UART
- reader-prox – similar to previous firmware
- tag-accel – reading accelerometer hardware values and writing this data to uart
- tag-ble-beacon – reading accelerometer values, writing them to uart and also advertising on radio
- tag-ipv6 – hello world example using Nordic Semiconductors IoT SoftDevice prototype s1xx-iot-prototype2_softdevice.hex. Nordic Semiconductors IoT SoftDevice is using IPv6 protocol for communication.
- tag-mischief
- tag-physical-web – physical web beacon with URL advertisement
- tag-power – power supply level advertising
- tag-sound – sound handling example in form on .wav file.
- tag-proximity – proximity detection and advertising.

Core functionality of the OpenBeacon firmwares uses CMSIS [12] peripheral abstraction library to access peripheral hardware devices. The projects are based on C Makefile building setup. Makefiles suggest that GCC ARM toolchain is used. The programmer for flashing nRF51822 seems to be Segger JLink that is nRF51822 officially supported commercial programmer.

The most complete design seems to be ‘tag_proximity’ type of the beacon. The firmware is using Enhanced ShockBurst (ESB) radio mode, which is Nordic Semiconductors proprietary protocol supporting two-way data packet communication including packet buffering, packet acknowledgment and automatic retransmission of lost packets [13]. Tag_proximity hardware production files are panelised and contain 24 OpenBeacon tags on a single production board. Panelising is commonly used technique to reduce production costs where multiple boards are put side-by-side on a single PCB design. Hardware design files are in Altium Designer [14] format, which is a commercial circuit designing software [14].

Tag hardware design is based on Nordic Semiconductors reference Smart Beacon design [15] with few added components. There is an extra 8MB (64Mb) Flash memory chip and an accelerometer to nRF51822 GPIO ports. Flash storage and accelerometer are connected via SPI and accelerometer is used to detect and report movement and physical state of the tag. Furthermore compared to Nordic Semiconductors reference design [15] there is an additional tactile switch and a CR2032 type coin battery holder for powering. These components can provide additional functionality but also add cost to the tags hardware.

SocioPatterns sensing badges are described [9] as peer-to-peer sensors that exchange ultra-low power radio packets that reach only close range (~1 – 1.5m) and with direct sight.

Direct sight requirement means, that assuming the badge is worn on chest, the facing direction can be detected as the human body acts as the shield for radio frequency of the badge [9]. In case of multiple people in a same room only the people who are actually facing each other are considered to be in contact.

The peer-to-peer topology used is described [9] as beacons detecting other beacons at a low power level. After successful detection they broadcast reporting data which is received by the base stations that are connected to the monitoring infrastructure [2].

The recorded data includes each detected contact between participants, its start and end times with a temporal resolution of approximately 20 s [16]. This way it is possible to monitor the amount of contacts between subjects, the duration of the contacts and the cumulative time spent in contact between the subjects, the frequency of encounters and the evolution of the measured quantities during the observation period [16]. The temporal resolution of required time unqualifies effectively random contacts and errors, for example subjects bypassing each other while moving. Also given the short range and the face-to-face requirement, it is reasonable to assume that the experiment is able to detect an on-going social contact (e.g. a conversation) [2]. The messages between tags and/or stations are encrypted and the data management is completely anonymous [2]. Encryption prevents unknown impersonating devices to pose as a beacon and mimic its behaviour to feed the infrastructure false data or gather the identities from the beacons for the use of own purposes.

2.2 MIT Sociometric badge

The human dynamics group at the MIT media laboratory has shown that commonplace wearable technology can be used to characterize the face-to-face interactions of employees in organization [17].

MIT media laboratory proposes [17] an approach for organizational mining that includes the following steps:

1. Capturing the interactions and social behaviour of employees, managers and customers using wearable and/or environmental sensors. Other sources of information that can be incorporated into the system are any form of digital records (e.g., e-mail, chat, phone logs).

2. Performing data mining and pattern recognition to extract meaningful information from these data.

3 Combining the extracted information with performance data (e.g., sales, tasks, and timing) and finding relationships between objective measurements and performance outcomes.

4 Generating feedback in the form of graphs, interactive visualizations, reports, or real-time audio-visual feedback for employees, managers and/or customers in order to improve organizational performance and customer satisfaction.

5 Designing and implementing organizational interventions based on behaviour simulation and prediction.

6 Continuous measurement and performance assessment.

The first proposed step [17] is capturing the interactions or contacts of the subjects. After that the other activities can be taken with the captured data like mining meaningful information from there, comparing it with other datasets, visualizing patterns etc. Sensor-based data capturing system consists of environmental and wearable sensors, computers, and software, that continuously and automatically measure individual and collective patterns of behaviour, identify organizational structures, quantify group dynamics, and provide feedback to their users.

Capturing interactions is done via wearable sensors. Proposed wearable sensors that employees carry around and that measure human behaviour can be mobile devices such as cell phones, PDAs, or electronic badges that collect data, communicate with a database (via Ethernet or wirelessly) to retrieve information, and provide feedback to their users [17]. In addition to wearable sensors there can be base stations that measure environmental characteristics and relay data to database.

To test their proposed approach the MIT Media Labs developed [17] a Sociometric Badge that has following sensors to measure data:

- Body movement activity Accelerometer to detect body movement activity.
- Microphone to measure speech activity and consistency.
- Infrared sensor to measure face-to-face interaction time, number of participants in face-to-face interaction and centrality of the contacts.
- 2.4GHz radio transceiver is used to send and receive data packets.

Current paragraph is a brief summary of MIT Sociometric Badges functional design [17]. The MIT Sociometric Badges work both as wearable sensors and also as base stations when placed in fixed known locations. The devices broadcast their ID using 2.4GHz transceiver every 5 seconds and using IR every 2 seconds. When a packet is received by another badge or base station it is logged with RSSI and sender’s ID. The badges can transfer data to base stations. Base stations can transfer data to central server. Data can also be stored locally and transferred over USB at the end of the data collection period.

MIT Sociometric badges were originally based on Atmel AT91SAM7S64 microcontroller and CC2500 transceiver [18]. Texas Instruments CC2500 is low-power proprietary 2.4GHz frequency transceiver with integrated modem. Technology is evolving fast and during last years the wireless low-power technology has moved steps forward, so it is reasonable to believe that newer Sociometric badges use newer transceiver and controller microcircuits. However it seems that the MIT Sociometric Badge project has turned over to commercial company called Humanyze [19]. The technical information about recent MIT Sociometric Badge based wearable sensors used for organizational data mining seems not to be freely available any more.

2.3 Protocols

For reality mining using open bandwidth radio signals it is not preferred to use proprietary own protocol but utilize an existing radio protocol. Although MIT Sociometric Badge uses BlueRadios BR-46AR Bluetooth Module [18] and it can communicate to Bluetooth enabled SmartPhones [18] it is not known if their device is fully Bluetooth compliant. SocioPatterns is mentioned [9] using RFID on 2.4 GHz band, specific details are not specified. When adopting more widely used standard, it may be possible to integrate other devices and not be limited to own technology – Smartphones and other devices supporting necessary technology can be used for communication and data transfer.

2.4GHz radio frequency devices examined in this thesis are classified by European Commission directive [20] as Class 1 equipment that can be placed on the market and be put into service without restrictions. The radio transceiver devices that fall into this category must have following properties (Table 2):

Table 2 Sub-class 21 of Class 1 according Commission Decision 2000/299/EC (6.4.2000)

Description	**Non-Specific Short Range Devices**
Frequency	2 400– 2 483.5 MHz
Class	Sub-class of Class 1 according Commission Decision 2000/299/EC (6.4.2000)
Transmit power / power density	10 mW e.i.r.p

Emerging technologies in this frequency range are Bluetooth, ANT and Wireless LAN 802.11 which are looked upon in following paragraphs.

ANT is a ultra-low-power 2.4 GHz wireless protocol for connecting multiple devices in robust and flexible manner [21]. ANT+ adds device interoperability [21] to ANT standard. ANT can be used in different target applications which include sport, wellness, and home health [21]. ANT seems to be more robust standard with small protocol overhead and support for different network topologies as P2P, tree, star and mesh [22]. Ant is very energy-efficient [22] but the downside of using ANT can be that it is not very widely spread at the moment. ANT can operate concurrently with BLE [23] and both technologies are implemented into nRF51822 microchip [24].

802.11, also known as Wi-Fi, is a protocol family that has multiple generations (a, b, g, n, ac) operating both on 2.4GHz and 5GHz frequency band. Wi-Fi is very widespread for wireless local area networking and Wi-Fi support can be found in great number of smart phones and devices. Wi-Fi standards are much used for wireless Internet and are thus focusing on high data throughput and good range. Wi-Fi can operate in infrastructure mode using star topology or in ad-hoc mode using P2P topology. Wi-Fi is used also for low-latency connection and piloting for some hobbyist unmanned aerial vehicles [25]. The technologies are evolving to intersect and Wi-Fi gets also used in IoT beacons [26]. Although widely used, regarding energy-efficiency for transferring small amounts of data it was not presently considered to be the best choice for mobile sensors used in current thesis.

Bluetooth has been in use for connecting peripheral devices for some time. Bluetooth Low Energy (BLE) was introduced [27] as part of the Bluetooth specification 4.0 introduced in 2010 and is not compliant with earlier versions i.e. “Classic Bluetooth” – it can work in compatibility mode when compliance is needed. BLE was originally designed [27] by Nokia as Wibree and was later adopted by the Bluetooth Special Interest Group. From the very start [27] the power-efficiency with low cost, low bandwidth and low complexity was main focus. In 2013 a major update called Bluetooth 4.1 was announced [27] which has some updates to protocol and user experience although core part of the protocol was left untouched. Bluetooth 4.1 is backward compatible [27] to version 4.0.

BLE has low data transfer rate because by specification every packet can contain only 20 bytes [27] of user data unless negotiated otherwise. Per connection interval [27] there can be up to 6 packets (depending on microchip used) which makes effective transfer rate about 125kbit/s.

There are two generic profiles defined by the BLE specification [27] that are fundamental to interoperability of the devices, those are:

- Generic Access Profile (GAP) defines [27] modes for broadcast data, discover devices, establish and manage connections.

- Generic Attribute Profile (GATT) defines [27] basic data model and procedures to discover, read, write and push data.

There are four roles defined [27] in GAP:

- Broadcaster periodically sends out advertisement packets.
- Observer scans and listens to advertising packets from broadcasters.
- Central is an initiator of connections with other peers.
- Peripheral uses advertising to allow central to find it and establish a connection.

The BLE specification does not impose restrictions regarding of particular device's roles. Each device can operate in one or more GAP roles simultaneously (24).

GATT defines [27] two roles:

- Client sends requests to server and gets responses.
- Server acts as counterpart to Client role and also can initiate update connections if so configured.

GATT servers can provide uniquely identified services with UUID and service characteristics [27].

The identifier of a BLE device is a Bluetooth device address [27]. This is 48-bit number that uniquely identifies a device and is similar to Ethernet MAC address [27]. The address can be either public address and registered in IEEE Bluetooth Registration Authority or a random or so called private address. For higher security the private address of the device can be regenerating and changing at time intervals using IRK – identity resolving key [27]. Nordic Semiconductors nRF51 SDK10 for SoftDevice S130 [13] has following symbols defined for address types in "ble_gap.h" header (see Figure 1).

```
#define BLE_GAP_ADDR_TYPE_PUBLIC    0x00 /**< Public address. */
#define BLE_GAP_ADDR_TYPE_RANDOM_STATIC   0x01 /**< Random Static
address. */
#define BLE_GAP_ADDR_TYPE_RANDOM_PRIVATE_RESOLVABLE 0x02 /**< Private
Resolvable address. */
#define BLE_GAP_ADDR_TYPE_RANDOM_PRIVATE_NON_RESOLVABLE 0x03 /**<
Private Non-Resolvable address. */
```

Figure 1 BLE address types in "ble_gap.h" from nRF51 SDK10

Random resolvable addresses with known IRK and changing at a time intervals should be preferable and provide more security for the purposes of mobile contact detectors. This way it is harder for a possible attacker to pose as a contact detector. Scanning all the available BLE devices in vicinity does not provide any useful data, so whitelisting of only known devices should be preferred. Additionally BLE bonding information could be kept to allow connecting to a device only from a known device. However for the prototyping try-out current work uses only simple static public addresses and whitelisting of the devices is done via predefined address range. Establishing connections is not secured in any way during the phase covered in current thesis.

BLE 4.1 provides star network topology in undirected broadcasting and observing GAP role and P2P topology in Central or Peripheral GAP role. It is announced that in addition to star and P2P the Bluetooth technology is going to have also mesh topology in late 2016 or 2017 specifications [28]. There is already an evolving mesh implementation in GitHub [29] for trying out proof of concept for nRF51822/nRF52832.

2.4 Proximity and facing detection

Spatial proximity can be detected using different approaches, for example, utilizing optical, acoustic or radio transceivers. MIT Sociometric badge detects face-to-face proximity using infrared optical sensors [18]. The benefit of using optical proximity detection can be that detecting light source outside of visible range can be quite precise and reliable and not much dependent on environmental conditions. OpenBeacon relies on RFID radio frequency proximity detection [9]. Many mobile consumer devices also use Hall sensor to detect very close proximity. The advantage of using RF is that it does not need additional components like coils, magnets or optical sensors, the device can be very small and energy efficient and can be adapted and calibrated to needed target usage.

Using radio transceiver to sense proximity relies on the path loss of the signal with increasing distance. The receiving device can use Received Signal Strength Indicator (RSSI) measurement to estimate the distance of the transmitting device. RSSI can be measured in decibels per mill watt (dBm) and in addition to distance it is dependent on multiple different factors such as the location environment, construction, noise and surrounding obstructing objects resulting in diffraction, reflection, phase fluctuations, multipath etc. [30]. Because of problems related to those phenomenon locative technologies are typically not viable at low cost to infer face-to-face contact between individuals [2].

Using free-space radio path loss formula Liu et al [31] brings out that Bluetooth RSSI can be theoretically modelled to distance from simplified distance relation based on Free-Space Path Loss equation (1).

$$\begin{aligned} RSSI &= Ptx + Gtx + Grx + 20\log_{10}(c\frac{c}{4\pi f}) - 10n\log(d) \\ &= Ptx + G - 40.2 - 10n\log(d) \end{aligned} \tag{1}$$

Where d is distance between transmitter and receiver, Ptx is transmit power, n is the attenuation factor, f is the central frequency of 2.44GHz and G is total antenna gain $G = Gtx + Grx$. From previous equation (1) distance can be brought out as shown in equation (2):

$$d = 10^{[(\Pr x - 40.2 - RSSI + G)/10n]} \tag{2}$$

This model can be used only as theoretical reference but in real situations due to different phenomenon mentioned earlier RSSI fluctuates even with constant distance and stationary transceivers. It was proposed [32] in BlueEye System for proximity detection a better model that was considered valid [32] for line of sight measurements of mobile 2.4GHz Bluetooth signal distance estimation. BlueEye model RSSI distance estimation is shown in equation (3)

$$RSSI_m = RSSI_0 + y(d)\log 10(d_E) \tag{3}$$

Where for considering dominant radio mode only y(d) is expressed as BlueEye model distance function equation (4).

$$y(d) = \{c_1 \sin(2\pi d / \Omega + \Theta)\}^2 + c_2 \tag{4}$$

In the equation (4) c_1 and c_2 are constants of the given environment, Ω is spatial channel wavelength, Θ is phase lag from the relative orientation of receiver and transmitter with the possible value in range of $0 < \theta < \pi/2$. The value of environment constants in BlueEye model [32] has to be found by calibration in real environments. It was found [32] that in confined office space at lower heights (<50cm) of the antenna RSSI measurement is less while in moderate heights (~1.2m) it is larger for close distances (<5m) about 8 – 18 dB. From this it can be considered that in order to get higher RSSI value, the mobile receiver should be carried on waist or chest level instead of attached to legs.

It is estimated [31] that face-to-face communication usually takes place in distance between 70cm and 250cm with average about 152cm. This value may depend on cultural context or other circumstances but can be considered as a good enough reference for the purposes of current thesis.

Liu et al [31] compared different methods to compensate distance estimation from RSSI and detect human contacts with Bluetooth Smartphones. Following paragraph summarizes their experiments.

RSSI values range depending on if the receiver is in backpack or has direct line of sight. For detecting this light sensors were used and used to estimate needed RSSI for contact detection. To eliminate RSSI fluctuations different approaches were tried and best results were got using data smoothing using EWMA (exponentially weighted moving average) with smoothing factor of 0.5. EWMA calculation of RSSI is shown in equation (5).

$$E_i = s * RSSI_i + (1 - s)E_{i-1} \tag{5}$$

E_i is EWMA factor of the given time i and s is smoothing factor.

Additionally probabilistic model was created by separating different real life situations with different Smartphone positions and light sensor input into different zones with different RSSI boundaries (from -90dBm to -50dBm) with range of 45dBm. When RSSI value was in high probability contact zone then contact was registered, when in lower probability zone then contact estimation probability was based on actual RSSI and predefined zones range relation.

Liu et al [31] perhaps did not have control over the transmission power of the Nexus 4 type Smartphones used in their experiments and low values (<-90dBm) were dropped.

SocioPatterns have used [16] lowest radio power in their experiments to ensure that exchange of packets between devices is only possible when they are facing each other. When combining multiple approaches then a reasonable solution seems to use fixed position forward facing mobile transceivers worn on chest, waist or arm. EWMA data smoothing and lowered radio power could also be used.

3 Comparative Analysis: Hardware and Architecture

3.1 Integrated Microcircuits

Although BLE Smartphones have been used [31] to estimate Proximity, it may not be the most cost-effective solution as it offers much functionality that is not needed. It would be good to wear detection sensors on chest for receiving stronger signal with person's facing direction to another person. Smartphones are hard to wear on clothes as a token because of weight and size, often they are worn in backpack [31] which renders facing detection useless. Additionally, there would be no full control over hardware, because third-party operating system is usually used. There are few alternatives on the market when trying to build a BLE solution. Our requirement is that the solution should be low-cost and low-power. It helps if it has integrated processor with enough computing power and support for peripherals like accelerometer and storage in case those will be needed at some point. Supporting older classic Bluetooth versions is not needed.

Table 3 BLE SoC comparison

Item	Protocol	CPU	Current consumption	Flash/RAM	Interfaces
TI CC26xx	BLE 4.1	ARM Cortex M3	5.9mA	128kB/20kB	SSI
NS nRF51822	BLE 4.1/ANT	ARM Cortex M0	9.7mA / 8mA	256kB/32kB	UART/I2C/SPI
DS DA14580	BLE 4.0	Cortex M0	4.9mA/4.9mA	32kB/50kB	UART/I2C/SPI
Cypress PSoC 4 BLE	BLE 4.1	Cortex M0	15.6mA/ 16.4mA	256kB/32kB	Comm blocks
CSR CSR101x	BLE 4.1	16-bit RISC	16mA	64kB/64kB	UART/I2C/SPI

Argenox compares [33] BLE microchips and comes to conclusion that in regards to processing power Texas Instruments CC26xx SoC is above the others. BLE SoC-s from Texas Instruments, Nordic Semiconductors, Dialog Semiconductors, Cypress Semiconductors and CSR with some specifications are shown in comparison table (Table 3). It can be seen that Dialog Semiconductors DA14580 supports only BLE 4.0 version while others also new newer BLE 4.1. Nordic Semiconductors nRF51822 is supporting also ANT protocol. While current consumption is best for Dialog, it has low memory and also the program memory is OTP (one-time-programmable). The BLE comparison chart left author mainly with 2 alternatives – TI 26xx and NS nRF51822. While the first one has better processing power, but the latter was chosen because of better support for peripherals, larger memory and it seemed to be more widely used (See next chapter).

3.1.1 nRF51822

Many wireless solutions are based on Nordic Semiconductors nRF51822 [24] microchip. For example a budget fitness watch and sleep monitor called MisFit Flash [34] is built on nRF51822 microchip (Figure 2).

Figure 2 MisFit FLASH activity tracker board

While tracker watches may have a bit different requirements than proximity tracking, nRF51 is also the preferred platform [35] for indoor positioning and tracking. At the time of writing current paper a newer version of the microchip was released called nRF52832 [36] – an upgrade to nRF51822 microchip that has more RAM and storage, improved energy-consumption and some additional features like on-chip NFC tag, on-chip Balun and digital microphone interface [36].

nRF51822 System on chip is based on ARM Cortex-M processor [24]. ARM Cortex-M [37] is power efficient 32-bit RISC processor and most popular choice for embedded applications [37]. ARM Cortex-M processor can operate both in Little Endian and Big Endian mode. Current thesis uses Little Endian also referred to as Least Significant Byte first or LSB. The nRF51822 is an ultra-low power SoC integrating 2.4GHz radio frequency transceiver, 2.4GHz ARM Cortex-M processor, flash memory and peripherals [24]. nRF51822 can operate on supply voltage range 1.8 – 3.6v [24].

The nRF51822 is sold in different packages. There is QFN48 (Quad Flat No-Leads) package [24] with side length of 6mm [24] and there are 4 WLCSP (Wafer-Level-Chip-Scale) packages [24] with minor differences and side length about 3.5mm [24]. QFN package as the name implies has no leads, so the surface-mounted connection has to be made directly to the exposed contacts on the microchip. It is possible to mount QFN packages manually with hobbyist equipment using hot air or gas soldering station, the videos about managing that [38] are available on the Internet. The WLCSP type packages of nRF51822 are smaller than QFN48 and for mounting those packages at least an access to an industrial oven is needed which the author of this paper did not have during the project and we look only QFN48 packages in this paper.

The nRF51822 QFN48 microchips have multiple revisions and variants differing from each other, among other attributes like RAM and Flash size [39]. Same variants were produced during different revisions, but the latest SoftDevices are compatible only with 3rd revision [39]. The revision can be determined by the variant build code. In the following table (Table 4) there are different 3rd revision QFN48 packaged nRF51822 variants from Product Compatibility Matrix [39].

Table 4 Variants of 3rd revision QFN48 package nRF51822

Variant	**Build code**	**RAM (kB)**	**Flash (kB)**
QFAA	Hx0	16	256
QFAB	Cx0	16	128
QFAC	Ax0	32	256

NRF51822 allows also for OTA (Over-the-Air) firmware update aka DFU (Device Firmware Update) and using a bootloader. This was not implemented in current project due to the RAM requirements for it and time constraints of the project. During current project the nRF51822 QFAAH0 microchips which are 3rd revision nRF51822 with 256kB flash and 16kB RAM were used.

3.2 Network topology

Different network architectures can be implemented depending on constraints and possibilities of used technologies and requirements of the task. In current chapter some network topologies are looked at for the mobile wearable radio transceivers which can help to detect human contacts.

It is possible to use star topology where all the mobile sensors broadcast, stationary base stations receive and register the signals and send them to network destination to collect and analyse the data. This way the mobile sensors' only responsibility is broadcasting which makes them radio transmitters-only (i.e. radio beacons) and implementation relatively easy. In case of multiple (at least 3) stationary base stations the locations of the beacons can theoretically be pinpointed knowing the exact predetermined locations of the base stations and using RSSI data from all receiving base stations to estimate the distance at the same moment from each specific station (Figure 3). The locating can be conducted via comparing RSSI and time difference of a node signal in multiple base stations.

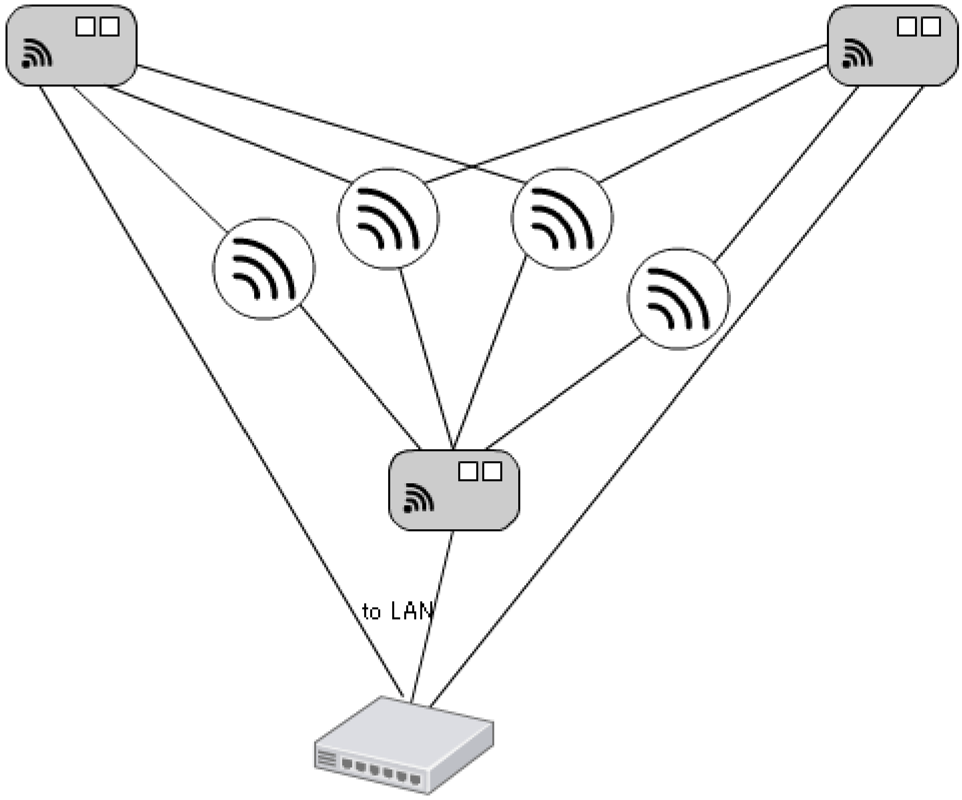

Figure 3 Simple trilateration topology

While approximate position of a person wearing a beacon at a moment is known, human contacts can be determined from extracting other persons who were nearby the same location at the time. Simple star topology with beacons and base stations would be a robust approach without technological limitations to storage or processing power because those tasks can be delegated via network and there are no spatial on temporal constraints for processing the data. However it may be hard to decide the facing direction of persons with such method. In case when persons own or other human body or other not-stationary equipment shields the radio waves somewhat then the location estimating with good enough distance resolution can fail. It can be hard to detect a beacon wearing person's facing direction and thus also contact without any additional sensors. Also there is need to set up the used space beforehand with base stations which should be with good enough density that at any used location multiple base stations are in reach of the beacons. In case persons leave the perimeter, contacts are not detected.

It is known [28] that soon next BLE version will be announced with support for mesh technology. There is already an existing mesh proof-of-concept library [29] for nRF51822 and some other technologies like ANT already support mesh topology [22]. With mesh topology every node forward-broadcasts the received data until the data reaches its destination or gets lost. This can be used for relaying detected nearby RSSI or detected contact data or to network (Figure 4).

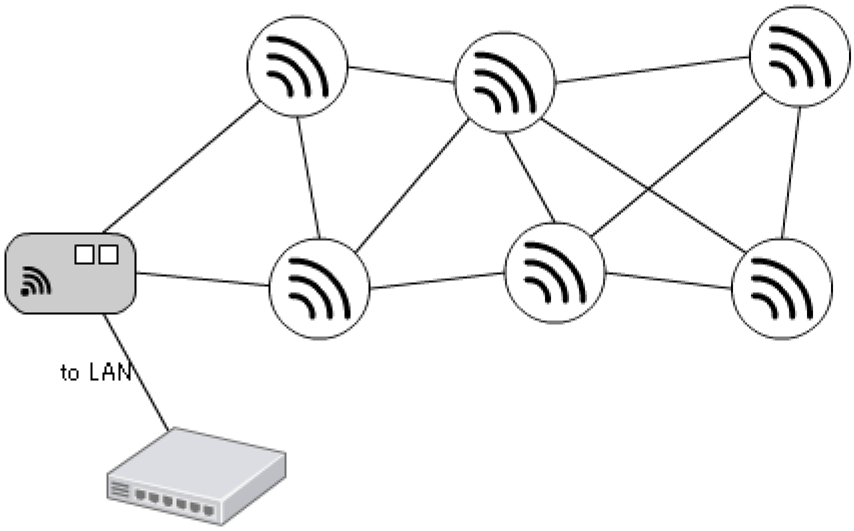

Figure 4 Mesh relaying topology

Using mesh topology requires nearby base station that relays the data to the network but the responsibility to advertise and detect advertisements is put on every beacon itself. It cannot be ensured that the data reaches its destination with mobile transceivers – at any needed moment there can happen situations where no linking node, that can relay mobile sensor data to any base stations, is available. Also this type of mesh may not be very scalable, in case of a number of nodes the network may get congested which may result in data loss.

When every mobile node can detect proximity of other mobile nodes itself, it can store the data and postpone the relaying of the detected data until it is requested from it. This requires storage ability for all the nodes but makes devices architecture more homogenous (Figure 5).

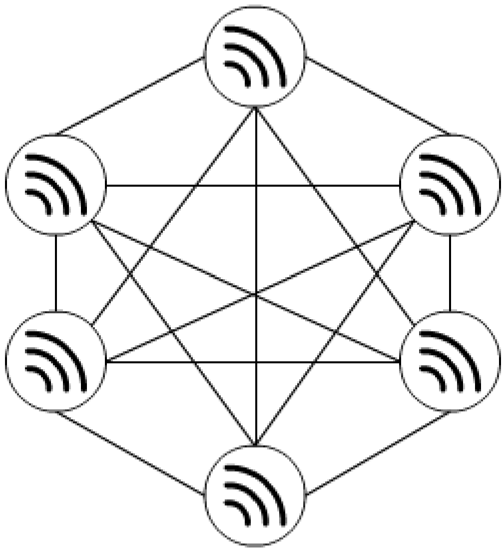

Figure 5 Topology with postponed data relay

There has to exist a method to sync time for all the devices either initially or at least when requesting and downloading the collected data at a later moment. Only by being able to map events of a device to a common clock with enough precision, the events from multiple devices can be mapped together to make sense. Downside of the Postponed Data Relay topology can be considered that additional hardware or software interfacing has to be implemented to allow remote data requesting. The same devices can be worn by organization members and also be stationary — this way locations can be detected and if wished the multilateration location detection tried. With postponed relay approach there is no need to fear network congestion because relaying does not happen and data copying can be done after active time. It is noted that congestion can still happen from overcrowded vicinity with extremely high number of devices broadcasting but this topic is not covered in current paper.

3.3 Data storage

It is clear that to use the sensor measurement data later it has to be stored somewhere. In the final phase the data could reach monitoring infrastructure database storage which can be even in the cloud but after the immediate measurement the proximity tag either has to take storage responsibility itself or delegate it. There are possibilities either to store measurements straight inside the sensor device or to send it directly to the relaying device that forwards it to the monitoring infrastructure storage. OpenBeacon has chosen the second approach to relay data directly although there might be later developments to also store the data. The MIT Sociometric Badge has both abilities [17] to store the data locally and to relay it directly to the base station. The relaying devices may not be available in all the locations, where the contacts take place. The circumstances can possibly sometimes be such that it is not even possible or not relevant to set up relaying devices – for example when the contact area perimeter is not specified, the location environment is not under the control of the interested party or cannot be set up beforehand. On the other hand the base relaying stations can help in addition to relaying the data provide an additional possibility to pinpoint the location of the contacts taking place.

The local storage of the active moveable contact sensors has to be low-powered because the power of the devices is limited. The physical dimensions and mass should also be taken into consideration because carrying the movable sensors around has to be comfortable and not taking any noticeable effort by the person. Storage accessing speed seems not to be of critical importance as long as it can be done in reasonable timeframe. Even if the storing procedure halts the other executing code of the moveable sensor system – few milliseconds of missing a contacts data should not have a great impact on the resulting dataset.

One of the most used relatively low power and lightweight physical data storage technologies today is flash memory cards. Memory card support can be found in many consumer devices like video cameras, Smartphones and tablet computers. It can be argued that the flash memory card technology is aged and is not relevant in future, for example new models of Google branded Android Nexus phones do not support external SD card storage [40]. However flash cards are still found in lot of consumer electronic devices and are relatively cheap. Multi Media Cards (MMC) and Secure Digital (SD) Cards are powered with 2.7 – 3.6v DC voltage. MMC/SD cards have internal microcontroller in them and can in addition to native mode also work in SPI mode [41]. SPI can effectively make connecting the SD Card/MMC to a microcontroller easier. There are 4 signal connections needed for SPI – Master-in-slave-out (MISO), Master-out-slave-in (MOSI), Chip-select (CS) and Serial clock (SCLK) and additionally power and ground connections for powering the card adapter. If not planning to use any other SPI devices on the same connections then the CS connection may not be needed. Signals connections need to have pull-up resistors to eliminate the possible data corruption by high impedance signals. An SD card adapter to the sensor circuit makes the storage easily replaceable – after the measurements are stored the card can be physically removed or exchanged to a new one and the flash card with the stored data processed separately in external controlled environment. Before removing the storage card from the sensor device, it must be ensured that writing is not in progress, otherwise the card data can easily get corrupted.

Another possibility is to use an integrated on-circuit non-volatile memory component. For example OpenBeacon nRF51822-based new tag includes Adesto Technologies 64Mbit AT45DB641E storage chip [11], although it does not seem to be extensively used by firmware code. On-board flash memory chip decreases the storage hardware mobility compared to removable memory card – storage capacity increasing possibility is limited and in case of break-down it cannot be so easily replaced. Nordic Semiconductors nRF51822 BLE SoC chip itself contains 128kB or 256kB of flash memory used for firmware code. Free part of this storage can be accessed programmatically to store needed data, although the capacity is perhaps not enough for contact data storage for reasonable timeframe.

There has been made a comparison research for sensor networks storage alternatives in sense or energy consumption [42] and it was found that parallel NAND flash memory consumes less power than serial memory storage. The presence of internal microcontroller in MMC cards increases the idle current as well as energy consumption for read, write and erase operations [42]. However, the comparatively larger number of I/O pins required for the parallel interface may pose interfacing issues on low power embedded systems with limited number of I/O pins [42]. From the abovementioned reasons it can be seen that in terms of power usage MMC/SD card is probably not the best solution.

Storing information just by continuous blocks is possible but it would have many benefits when a file system is used, let's have a look at few. In the current use case an addressing table of a file system would be handy to allow storing the data in directly accessible named files instead of just direct sequence of bytes. Additionally data validity checking can be provided by file system often via some kind checksum algorithm to ensure that read data was truly the same that was intended to write at the first place. File systems also provide a mechanism to write data not sequentially but into randomly addressed blocks on the storage device. Flash type storage typically has limited number of writes to a block within lifetime. File system can help to save on the write operations on the most used blocks and distribute the data more evenly. In case a file system is not used, some of the storage blocks, for example in the beginning of the storage device, are used for writing more than the other ones. There can be some storage blocks on the device that are corrupted due to manufacturing or other faults or either are worn out, file systems usually allow to mark those blocks as bad and avoid storing data in those places while the good storage still remains usable.

For removable SD/MMC card it would be useful if the file system on the card is supported by PC operating systems. This would allow the information to be copied from the removed sensor card with the PC computer card reader using operating system's own utility functionality.

The file system used should be lightweight because RAM of the integrated microcontroller is usually very limited and the footprint has to be small. There are multiple file system libraries for embedded systems, many of them are supporting FAT-12, FAT-16 or/and FAT-32. FAT is a robust legacy filesystem that is supported by most desktop operating systems like Linux, OS X and Windows. FAT has a boundary of 4GB for storage size [43], so in case of larger storage is needed, some other file system option should be considered.

Popular Atmel RISC microcontroller based Arduino board's software development environment includes support for SD cards with FAT16 and FAT32 file systems based on sdfatlib by William Greiman [44].

There is an SDFilesystem library for The ARM mbed platform that can be used on any other ARM CPU based environment with SD card via SPI connection. SDFilesystem supports FAT12, FAT16 and FA32 [45].

FatFS [46] is a generic FAT module written in ANSI C that supports FAT12, FAT16 and FAT32.

One of the mentioned libraries can possibly be adapted to use in contact sensors for data storage in case file system is needed. File system decision is depending on specific implementation of the contact sensor.

4 Solution development

4.1 Development stack

NRF51822 SoC chip integrates ARM Cortex M0 processor. By manufacturer the best supported development environment [47] for nRF51822 firmware development is Keil MDK [48]. Keil MDK is a commercial ARM Cortex-M Microcontrollers software development solution that consists of uVision IDE, ARM C/C++ compiler toolchain, Pack Installer for managing development packs and CMSIS. CMSIS [12] is a vendor-independent hardware abstraction layer for the Cortex-M processor series. CMSIS aids to interface peripherals and middleware to the processor [12]. Keil MDK is free up to firmware sizes under 32kB (firmware made for this paper is around ~95kB), otherwise the solution is commercial with prices, depending on installed nodes and used microcontrollers, up to several thousand euros.

There exists an alternative toolchain that can be used. If GNU GPL licensing terms suit the needs then GCC (GNU Compiler Collection) toolchain for ARM Cortex-M & Cortex-R processors [49] can be used for firmware development. This toolchain is officially maintained by ARM [49] and can be used on different host platforms including Ubuntu, MS Windows and OS X [49] to develop firmwares for supported processors. GCC ARM toolchain makes it possible to compile files on PC for supported ARM processor based systems, which is sometimes also known as cross-compiling. In addition to the toolchain also GNU make utility may be needed, which is usually already included in distributed packages on *nix systems, but is available also for different other host platforms, Windows among them.

Used IDE for writing the code, building, debugging and downloading firmware can be very widely used open source Eclipse IDE [50]. Nordic Semiconductors has published a tutorial [51] about how to set up environment to develop software for nRF5x series with GCC toolchain [49] and Eclipse IDE [50]. This tutorial [51] also recommends using Gnu ARM Packs [49] plugin for Eclipse. Packs enable convenient peripheral view in Eclipse that can provide real time structured information about the different registers of the debugged system. However, the tutorial about using GCC ARM toolchain [49] as well as different other Nordic Semi tutorials assumes that Segger J-Link programmer [52] is used.

4.1.1 Programmer

To develop nRF51 software there is a need for separate hardware called programmer. This is used to connect target nRF51822 system to host PC and allows to transfer the code to the target system and in some cases also to trace and debug the source code. There have been different Nordic Semi original development boards [53], [54], [55] and an USB dongle [56]. Latest Nordic Semi original development kit [53] and nRF51 development dongle [56] include Segger J-Link on-board programmer and debugger [52] already on the board. This on board programmer [52] can be used to program without any additional cost the development board itself from host PC as well as to manage and program other nRF51822 systems [57]. In case another not original development board is used what does not have the on-board J-Link, the programming has to be solved separately.

Segger J-Link debug probes [58] programmer is sold also for end use and is a commercial product with different variations and different prices. There are Base, Plus, Ultra+, Pro and a basic and even educational model [58] available with limited functionality. The prices [59] of J-Link probes and original development boards however made the author look for alternatives that fit into the budget of the project better.

A company named Icarus Sensor System has written an article [60] that proposes a way of developing nRF51822 system using alternative low-cost tools and is recommending using ST-Link v2 [61] programmer for developing the nRF51822 systems. The article [60] recommends using also GCC toolchain [49], SRecord [62] for merging the compiled files, mbed ecosystem [63] and OpenOCD [64] for debugging the system. The development environment described in the article [60] is focused on mbed [63] development which might be called "Arduino" of ARM world [60] and was not used in this project.

STMicroelectronics – a company that manufactures the ST-Link programmers – is a leading semiconductor solutions providing company with headquarters in Switzerland [65]. ST-LINK/V2 is an in-circuit debugger and programmer for the STMicroelectronics STM8 and STM32 microcontroller families [61]. The STM32 product family of the company is based on ARM Cortex-M processors similarly to Nordic Semiconductor's nRF51 products – so the internal processor is the same and it can be programmed with similar tools. The JTAG/SWD (serial wire debugging) target interfaces supported by Segger J-Link programmers [58] are also supported by ST-Link [61] and are possible to use for programming nRF51822 systems. Original ST-Link/v2 programmers are better priced than Segger J-Link programmers, but Icarus describes in his article [60] that there are cheap replicas of the ST-Link/v2 programmer on the market that can be used for the given task or even own programmer is not very difficult to be built. ST-Link/v2 type programmer used for this paper was found from eBay for approximately €3 (Figure 6).

Figure 6 ST-Link v2 type programmer used for current thesis

ST-Link type programmer's SWD connection was connected to an nRF51822 system. nRF51822 chip in QFN48 packet has pin 23 for SWDIO/nRESET described in product specification [24] as System reset (active low), hardware debug and flash and programming I/O. Pin 24 is named SWDCLK [24] and described [24] as digital input Hardware debug and flash programming I/O. ST-Link type programmers need WinUSB driver to work under windows operating system [66].

Article from Icarus [60] proposes also using OpenOCD for debugging. OpenOCD or Open On-Chip Debugger [67] is a free programming and debugging software that has among other hardware also support both for nRF51822 chips and ST-Link/v2 programmers [64]. OpenOCD supports GDB [49] hardware debugging and also telnet protocol. This way the ST-Link/v2 programmer can be connected to nRF51822 system and used via telnet commands manually or via gdb as a debugger. There is also a free OpenOCD plugin for Eclipse IDE [68] that integrates the OpenOCD well into Eclipse, make configuration pages available and pipes the OpenOCD daemon into Eclipse.

While development and testing of embedded applications can often be done directly on the final product by soldering ISP (In-System-Programming) pads but is more convenient to do on development kits. Development boards could provide solderless connections and additional peripheral possibilities what could come handy for debugging and testing. As already mentioned in the beginning of this chapter there are multiple different original development boards for nRF51822 [53]–[55], [55], [56]. Additionally there is a cost-effective third party alternative to original development/evaluation boards. A development board [69] originally manufactured by Waveshare is claimed [70] to be software compatible with Nordic Semiconductors older PCA10001 development kit [71] and shown on following photo (Figure 7).

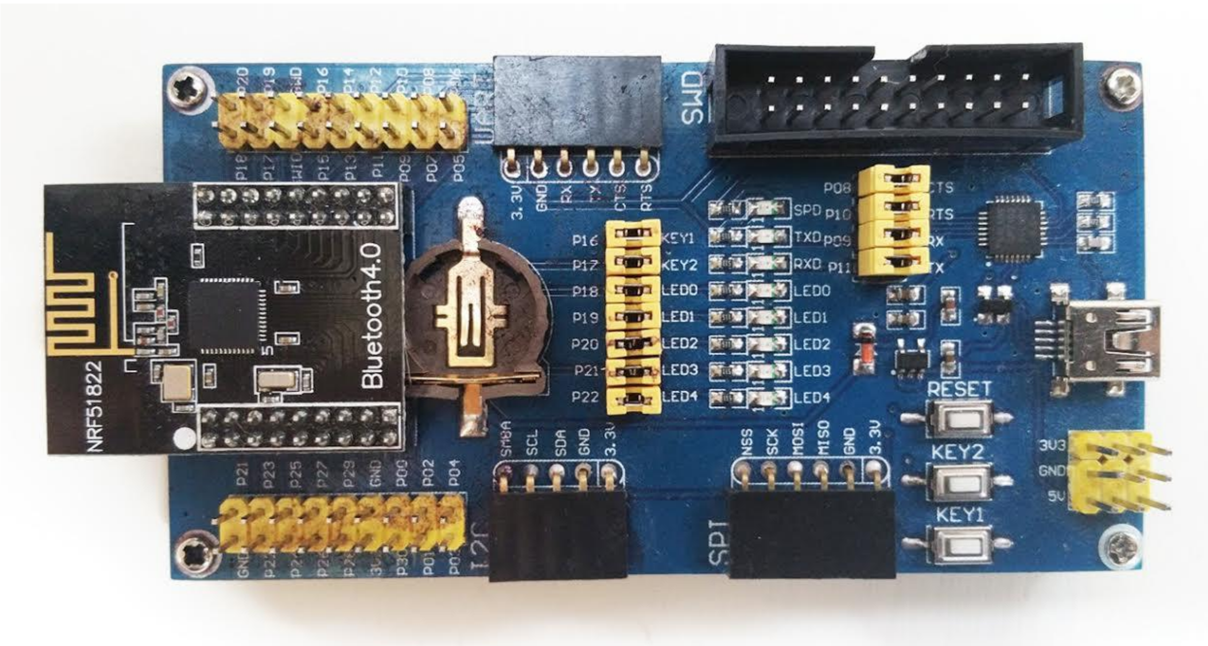

Figure 7 Waveshare BLE400 development board with nRF551822 module

On-board Segger J-Link programmer [52] is not included with the Waveshare board like it is with original development kits. Waveshare board provides 3 buttons and 4 LEDs with ability to disconnect them via separate jumpers. LED-s and buttons can be handy in different stages of development even if they are not needed in the end product. The design is modular – so the nRF51822 microchip module is connected to the board via a socket and can be easily exchanged. Like with many cheaper alternatives, this board is not very well documented. There is also an additional on-board UART CP2102 chip. CP2102 is Silicon Labs chip that can map a serial port to USB port [72]. Drivers are needed for windows for CP2102 [72] but there should exist kernel module to use it under Linux. The CP2102 can be used during development phase for example for trace logging into host PC serial terminal, although such logging is relatively slow and can give timeouts and crashes in radio module precise timings. The Waveshare board CP2102 serial communication works on following nRF51822 pins (Figure 8).

```
#define RTS_PIN_NUMBER 8
#define TX_PIN_NUMBER  9
#define CTS_PIN_NUMBER 10
#define RX_PIN_NUMBER  11
```

Figure 8 Waveshare CP2102 serial pinout

The Waveshare board has also a socket for CR1220 type coin cell and sockets for connecting i2c/SPI/UART/SWD interfaces as well as power 3v3 and 5v power connectors. The design is modular and nRF51822 module connected to the board can be exchanged. These boards are sold on online market for under 20€.

For the purpose of the project described in current paper a Waveshare Evaluation Board, Waveshare nRF51822 QFAAH0 module and ST-Link/V2 type programmer was bought from online market. ST-Link USB driver was installed and official STM32 ST-Link utility [73] recognized the programmer successfully. ST-Link utility was used to update the ST-Link internal firmware successfully. The SWDCLK and SWDIO wires were used to connect nRF51822 module. OpenOCD daemon was used for connection initiating. It was not successful to initiate nRF51822 flashing with ST-Link using external power supply for nRF51822 module and connecting only SWDCLK and SWDIO pins. It turned out that for successful connection the nRF51822 needed to be supplied power by connecting ST-Link 3.3v and GND pins to nRF51822 pins VDD (positive power) and VSS (ground) pins [24]. This requirement may present itself because of nRF51822 SWDIO pin acting also as reset pin but more thorough research was not done.

As a result the nRF51822 products could be programmed and debugged by ST-Link/V2 through OpenOCD software, although connecting 2 signal connections is not enough but 4 is needed.

The development stack that was used for the purpose of this thesis is described in following table (Table 5).

Table 5 Development stack

Component	**Product**	**Price**
IDE	Eclipse IDE for C/C++ developers	-
Toolchain	GCC ARM toolchain	-
Packs	GNU ARM C/C++ Packs	-
Debugger	OpenOCD	-
Build tool	GNU Make	-
Programmer	ST-Link/v2 type	~3€
DK	Waveshare BLE400 evaluation board with QFAAH0 nRF51822 module	~15€

In current chapter author described a cost effective development stack for nRF5x microchips based on ARM CPU-s. Application can be written in Eclipse IDE. Source can be compiled with GCC ARM toolchain. Via SWD programmer the compiled code can be downloaded to microchip using OpenOCD and the code can be debugged utilizing GCC debugger GDB and OpenOCD. OpenOCD can use 4 hardware breakpoints and 2 watchpoints on nRF51822 in debugging mode. However using breakpoints while in middle of radio session can crash the device.

4.1.2 SDK & SoftDevice

The information provided in current chapter is summarized from Nordic Semiconductors Infocenter [13] unless stated otherwise. Nordic Semiconductors provides pre-compiled libraries for nRF5 systems that are called SoftDevices. Only one SoftDevice can be used at a time and the nRF5x microchip can be used with or without a SoftDevice. SoftDevice is basically a binary file that provides necessary hardware functionality and radio controlling behaviour for the firmware code. Implementing supported and required radio protocols for the device is also a responsibility of a SoftDevice. The application can be written without using a SoftDevice but utilizing a SoftDevice makes the application development lot easier and helps to ensure that the radio protocols, packets and timing windows are all following necessary standards so the intercompatibility between different devices can be achieved with less effort. Interfacing application code with SoftDevice can be done with an appropriate SDK.

Nordic Semiconductors SDK (software development kit) comes in different versions with support to different hardware revisions and versions (described in Chapter 3.1.1). There exists also an IoT SDK which is aimed towards Internet of Things devices and offers usage of IPv6 protocol over Bluetooth. This offers complete Internet Protocol Suite and provides the possibility of different devices on different hardware to communicate using a well-established and known protocols. In current thesis however IoT SDK was not used. The latest regular production SDK for nRF51822 at the time of writing current thesis was NRF5 SDK v10. Function signatures and behaviour can differ a bit between different SoftDevice and SDK versions. SDK10 requires a revision 3 hardware [39] and supports following SoftDevices and versions (Table 6)

Table 6 SoftDevices supported by NRF5 SDK v10

SoftDevice type	**Version**	**Usage**
S110	v.8.0.0	BLE peripheral solutions.
S120	v.2.1.0	For both BLE peripheral or central.
S130	v.1.0.0	BLE peripheral and central solutions.
S210	v.5.0.0	Full ANT protocol stack solutions.
S310	v.3.0.0	Combined ANT and BLE peripheral solutions.

NRF5 SDK is a freely downloadable bundle of header and source files divided into libraries, examples and documentation for developing nRF5x based applications.

SoftDevices allocate predetermined region in both volatile and non-volatile memory, so these parts of memory cannot be accessed by application code. Using a SoftDevice has to take notice of the exact specification of the SoftDevice and application code has to be linked according to available memory region alignment.

Table 7 S110 and S130 memory usage comparison

Property	**S110**	**S130**
Code R1 Base	0x16000	0x20000
RAM R1 Base	0x20002000	0x20002800
Stack usage	1536 bytes	1546 bytes
Heap usage	0 bytes	0 bytes

Program memory is divided into regions of 0 – SoftDevice and 1 – program. Code R1 base is the address where application vector table is located. Program code begins from Code R1 base + 0xC0. RAM starts for both SoftDevices from address 0x20 000 000. RAM space is similarly to program space divided into regions and RAM R1 Base is the address where program allocated memory starts. Heap and Stack are taken from RAM. It can be reckoned from the table (Table 7) that S110 reserves 88kB of non-volatile and 8kB of volatile memory and S130 accordingly 128kB of non-volatile and 10kB of volatile memory. QFAA version has 256kB of Flash and 16kB of RAM [39], so using S130 leaves less than 128kB of Flash and less than 6kB of RAM for program use.

Downloading a SoftDevice into nRF51 microchip can be done separately from application code or compiled application code can be merged with a SoftDevice to a single file that can be transferred as a whole to a nRF51 microchip. Utilities for managing binary files merging are provided with nRFgo Studio software that is meant exclusively for Segger J-Link programmers. When linking compiled files with ARM GCC linker and with accurate region addresses in link file, then the application hex filed can be downloaded onto nRF51822 microchip right address with OpenOCD. In order to transfer existing SoftDevice file into a fresh unused microchip, OpenOCD command ‘mass erase’ may need to be provided beforehand to allow writing it to the microchip. The application binary file can be transferred alone with OpenOCD without overwriting the SoftDevice.

4.2 Hardware development

nRF51822 [24] microchip does not work as intended without circuit, connecting only a battery to power the IC does not lead to a usable radio transceiver – in addition an external Balun, antenna, crystals and other components are needed. Nordic Semi offers few reference designs, for example a Coin Cell Design [15], for nRF51822 devices depending on wished target usage. Nordic Semi recommends that all other designs use those reference designs as a base and extend them. Even the track length should be based on reference designs – which is understandable, because with high-frequency radio devices – signal strength, component positioning, external noise, timing and other factors matter much.

For hardware development the Altium Designer [14] software is recommended by Nordic Semi and also all the Nordic Semi reference designs are offered in this format. Altium Designer is a not very favourably priced commercial product without free permanent but only with a free evaluation version. Evaluation version was considered not enough for current project and so alternatives were looked for. A widely used PCB designing software with a free limited community version is Eagle Designer [74]. Free Eagle Designer version can be considered one of the most used PCB designing software for open source projects. Author tried to export and import Altium Designer format into Eagle through third party format but the results were not satisfactory. Probably the best bet to get Nordic Semi reference designs from Altium format to Eagle is to redraw them and this can be somewhat time consuming. OpenBeacon designs were also in Altium format and included Gerber and PDF format of production files. Because OpenBeacon team was not accessible author used these designs to try to manufacture an OpenBeacon Proximity Sensor.

When the design is created there are multiple ways to achieve the manufacturing results. There is a possibility to order them manufactured in factory with industrial technology. For instance Eurocircuits [75] offers industrial-grade PCB manufacturing in small (<100 boards) quantities. For even smaller quantities there are also possibilities to try to manufacture them in home environment.

PCB was tried to be manufactured in home/hobbyist environment using photosensitive lacquer on the board copper surface layer, projecting a circuit image into a board through a printed transparency developing it and chemically etching the copper. Because of the microscopic nature of the nRF51822 schematics the printer resolution for printing transparencies needed to be high to get satisfactory results this way. Industrial PCB manufacturing introduces solder mask and silk screen that aid to connect components. These were not available for home-manufacturing experiments.

When a satisfactory PCB is successfully etched there is a need to connect the components on the PCB. In case no access to a temperature calibrated soldering oven is available connecting can be done also with soldering paste and hot air soldering iron. There exist tutorial videos showing QFN soldering with hot air and solder-paste [38]. This is a work that requires precision, firm hands and experience and is not scalable at all for high quantities of devices needed. Trying out this approach with OpenBeacon design PCB-s did not succeed – copper layers came loose because of varying heat applied during multiple tries and solder paste created unwished connections because of missing repellent solder mask on the PCB that is usually present on industrially manufactured PCB-s (Figure 9).

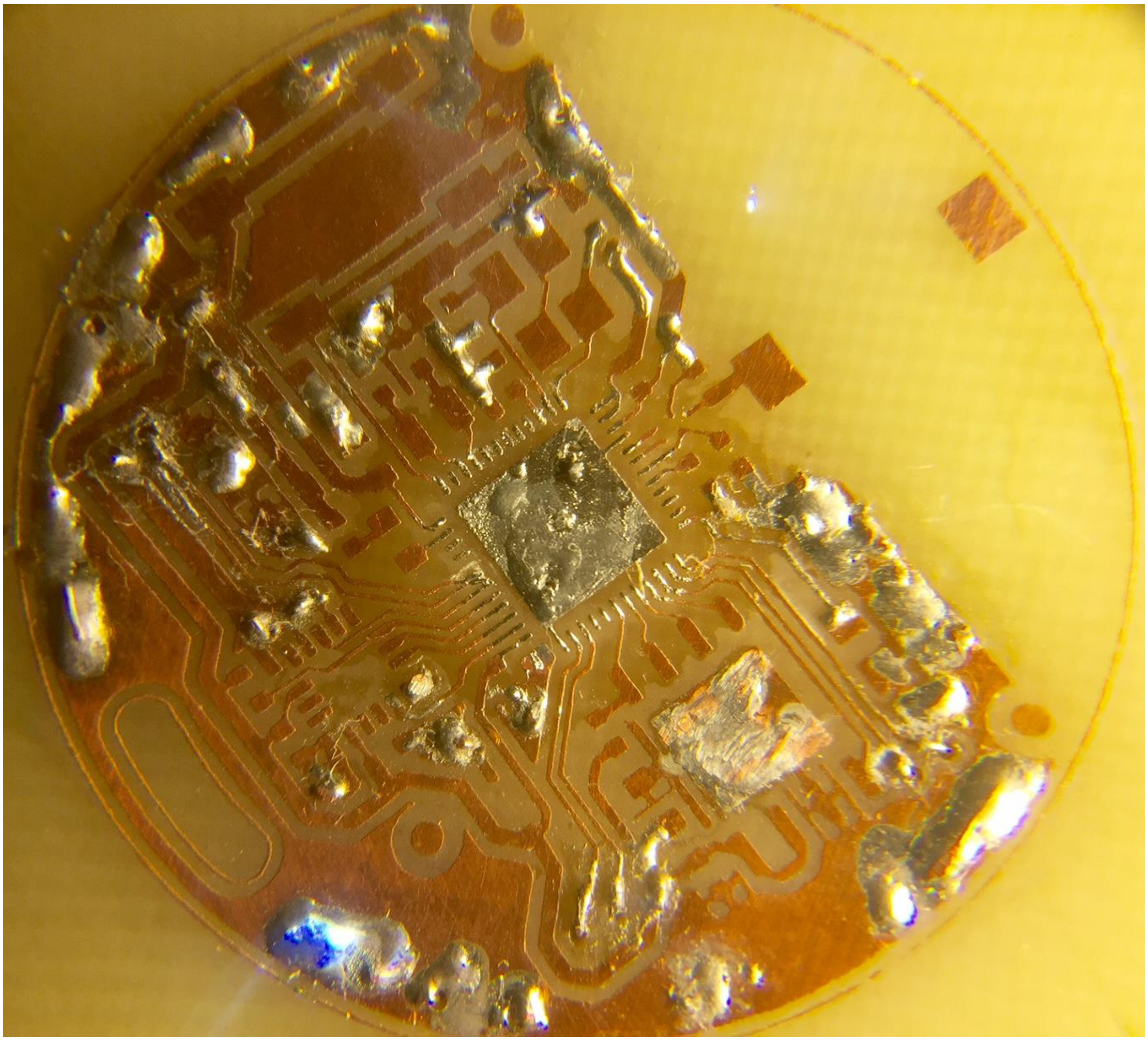

Figure 9 Home-made OpenBeacon proximity tag PCB with solder-paste residue

After multiple unsuccessful attempts of home-manufacturing a board based on OpenBeacon proximity tag design, author's eyes turned to ready manufactured boards. There turned out to be multiple ready-manufactured nRF51822 beacons on the online-supermarkets with reasonable prices. Downside is that SCH board schematics files are often unavailable and it may be hard to visually decide what exactly is on the board or how is it connected. The sellers often provide no documentation and claim false information in exchanged letters, so it can be hard to identify exact specifications. Fortunately nRF51822 beacons are often based on Nordic Semiconductors' Beacon reference design. Author found a beacon from online-dealer that seems to be Radioland-China nRF51822 beacon [76] with PCB antenna and battery holder for CR2032 type cell from online. Seller claimed to have an accelerometer under the bottom left corner of the n51822 chip (Figure 10). However further inquiries were not answered and the chip and its datasheet were not succeeded to identify. Used LED and its pin connection was identified by visual inspection and by trying out experimentally. For using existing pre-manufactured designs, it is also mandatory that the SDIO and SDCLK programming connections are brought out to pads from the microchip. This allows to solder programmer connection and flash the firmware of the device. The original firmware of the Radioland China beacon device turned out to be just a random address BLE Beacon and was erased after delivery.

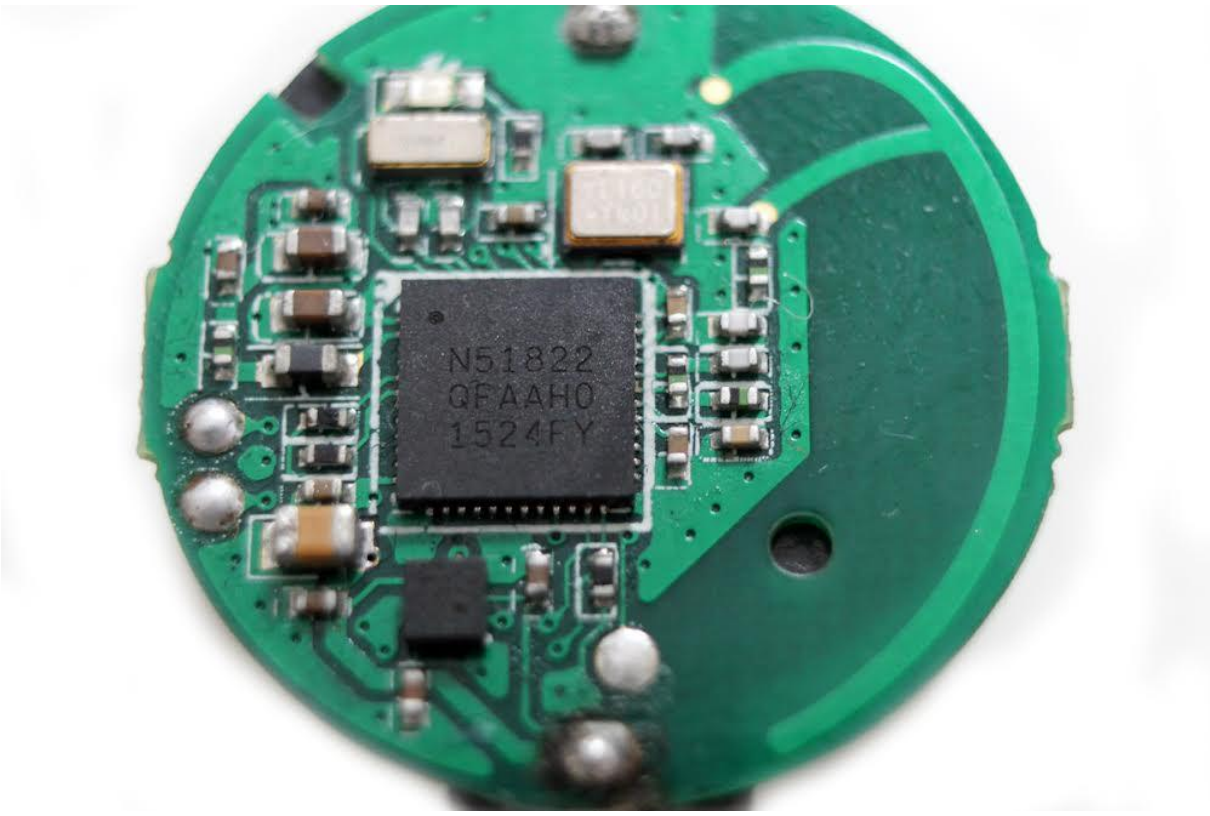

Figure 10 Radioland China nRF51822 beacon

4.3 Reality mining using BLE and nRF51822

A reference framework [27] is provided by Bluetooth SIG that covers among others the use case of Proximity Profile [27]. BLE Proximity profile [27] detects the presence or absence of nearby devices. It can define a behaviour [77] when a device moves away from a peer device so that the connection is dropped or the path loss increases above a preset level, causing an immediate alert. Proximity profile can also be used to define behaviour [77] when devices come together. BLE Proximity profile relays on establishing a connection between devices. S130 SoftDevice can support up to eight [78] simultaneous connections. Author assumes that human contacts can happen simultaneously with more than 8 people, so implementing a BLE Proximity Profile to detect such contacts does not seem to be the best solution.

In order to have homogenous architecture it is needed that advertisers also scan. Those are different GAP roles of broadcaster and observer but the GAP roles are not exclusive [27] by BLE specification, so both can be implemented for one device but behaving in multiple roles simultaneously with a single antenna may be hard to implement due to the physical limitations. nRF51 SDK is providing timeslot API [79] that can provide a timeslot ranging from 100 µs to 100ms during which the application has full control over the radio of the device. This way it is possible to alternately scan and observe in the single application. Nordic Semiconductors is offering an example in GitHub repository [80] for observing and scanning with single device using Timeslots and SoftDevice S110.

However S130 SoftDevice is described [78] to have support for simultaneous observing and broadcasting built-in, so the timeslot usage can be left out from the application and exact timing left to be controlled by SoftDevice itself. The downside of using S130 instead of S110 is the requirement of more memory resources by the SoftDevice.

nRF51 SDK10 includes pstorage library [81] which provides functionality to store application data directly in SoC flash area side-by-side to the firmware code. In prototype phase of current paper the external storage and file system was not used but instead stored data in the nRF51822 internal storage flash remaining free space with pstorage library.

As a fully BLE compliant solution was implemented it is possible to receive data and give commands to the devices using simple Smart phone with BLE support. For this purpose a GATT service was implemented in application. In prototyping phase the service has predefined UUID of NUS (Nordic UART Service) as defined in SDK10 “ble_nus.h” header (Figure 11).

```
#define BLE_UUID_NUS_SERVICE 0x0001                      /**< The UUID of the
Nordic UART Service. */
#define BLE_NUS_MAX_DATA_LEN (GATT_MTU_SIZE_DEFAULT - 3) /**< Maximum length of
data (in bytes) that can be transmitted to the peer by the Nordic UART service
module. */
```

Figure 11 NUS service header file UUID definition in SDK10

The developed firmware has following functionality using NUS service. In case an administrator device (e.g. a Smartphone) initiates a connection to the device advertising and scanning are stopped and GAP role of peripheral and GATT role of server is assumed by the device. When a Smartphone connects to device NUS service it can download data from the device storage, clear the storage, toggle the write-enabled flag and change the device radio transmitting power. Data is divided into 20 byte blocks because GATT MTU size sets limit to maximum amount of information that can be stored in a BLE packet.

4.4 Real-time clock

There are two clock sources implemented in nRF51822 SDK – RTC0 and RTC1 [24]. NRF51822 SoftDevice uses RTC0 for internal purposes. nRF51822 RTC1 can be used by application and in our device ticked in frequency of 2^{14} Hz that is 32.768 kHz. There is built-in 12 bit prescaler in SDK but author didn't use it because different modules could set restrictions for RTC1 prescaler. For example a defined macro in nRF51 SDK10 timer library for converting milliseconds to timer_ticks declared in "app_timer.h" header file has a comment describing that, although the prescaler is 12-bit which means that max value can be 0xFFF i.e. 4095, the maximum possible value of the prescaler in this macro may be 6 when using 32768 Hz clock (Figure 12).

```
* @param[in]  MS          Milliseconds.
* @param[in]  PRESCALER   Value of the RTC1 PRESCALER register (must be * the
same value that was passed to APP_TIMER_INIT()).
* @return     Number of timer ticks.
*/
#define APP_TIMER_TICKS(MS, PRESCALER)
```

Figure 12 Declaration of APP_TIMER_TICKS macro in SDK10 timer library.

Author made a RTC1 clock imprecise comparison with NTP-synchronized PC clock. As the RTC1 counter is 24 bits [24] the maximum value is 0xFFFFFF which is for 32768 Hz clock about 512 seconds, then a timer function was made that counts epochs into a static variable. It turned out that after powered on for 12 hours and 9 minutes the RTC1 showed 85 epochs and 6536640 ticks.

$$
\begin{aligned}
&6536640 + 85 * 0xFFFFFF = 1432599915;\\
&1432599915 / 32768 \approx 43719;\\
&(12 * 60 + 9) * 60 = 43740;\\
&43740 - 43719 = 21 \text{ seconds}
\end{aligned}
$$

Figure 13: Calculating RTC1 difference

As a conclusion the difference of RTC1 in 12 hours and 9 minutes was less than measurement error of used reference clock which should be enough for the purpose of timing human contacts.

The developed firmware uses RTC1 repeating timer with timeout. The timeout value can be defined by `TIMER_TIMEOUT_TICKS` constant in developed “`timer.c`” file. The value was left to 65536 ticks, which is 2 seconds and is used for epoch increasing as well as for indicating status of the device asynchronously without fear of timeout for the radio event handling. The implementation uses member variable of pointer to void function with the signature as follows (Figure 14)

```
void (*timer_event_indication)(void);
```

Figure 14: Timer indicator variable declaration

When notification should be indicated then handler can omit value of the variable to a needed function like shown on following code example (Figure 15).

```
timer_event_indication = indicate_proximity;
```

Figure 15 Example of setting indicator

In case of timeout event the function is executed and cleared using following code (Figure 16)

```
if (timer_event_indication)
{
		(*timer_event_indication)();
		timer_event_indication = NULL;
}
```

Figure 16 Executing timer indication function

This way only latest notification (if existing) is executed each time the timer runs out but there is no fear of delaying too long and crashing the SoftDevice, because the timeout function is executed in low priority and may differ in exact execution intervals because of high priority radio operations.

4.5 Data structures

The contact detection data is divided into blocks to store information. In order to save on data capacity this information can be optimized. In current chapter we will look at needed data fields length and how much it can be shortened to store only useful information (Table 8).

Data	**Length**	**Useful length**
RTC timer	48 bit	8bit
Epoch	8 bit	8bit
RSSI	8 bit	8bit
Device address	48 bit	8bit
Total	112 bit	32bit

Table 8 Length of data fields of received advertisement

RTC timer of nRF51822 is 48 bits. This means that when nRF51822 RTC reaches 0x00FFFFFF = 16777215 ticks, it is overflowed and if not stopping it will restart from 0. With prescaler value of 0 (Paragraph 4.4) it is less than 512 seconds, i.e. less than 9 minutes. To measure longer values with prescaler 0, an epoch counter can be used. Using epoch counter of 8 bits allows to use device without using RTC1 prescaler for maximum of 255 * 511.9999 = 130559.9922 seconds = 2175.9999 minutes = 36.2667 hours. The RTC1 clock of 48 bit measures time with ~32 kHz clock accuracy but it should be enough to use only 8 bits for this. When dividing maximum value of 48bit RTC1 to maximum value of 8 bits as follows 0x00FFFFFF / 0xFF = 0x10101 = 65793 we can get that the

RTC1 value should be divided with 0x10101 to receive 8bit value of RTC1. With such approach the accuracy resolution of time measurement decreases to 65793 / 32768 = 2.007843 seconds. 2 seconds was considered accurate enough for human contact detection.

Full BLE device address consists of 48 bits. As long as no more than 256 devices are used for contact detection, it should be enough to keep only the 8 bit part of this address to identify a device.

This optimization reduces the needed stored data from 112 bits to only 32 bits for one detected broadcast of the mobile sensor. Later this data can be downloaded, analysed and human contacts extracted from it for deeper analysis.

In terms of device storage space it would be even more sparing if the contacts and their lasting is detected by the device already. Even more storage space can be saved if not the advertisement broadcasts are saved but the contacts are detected. What is actually needed from data, is a contact party and time and length. So instead of RSSI it may be feasible to measure contact length like shown on Table 9.

Table 9 Length of data fields of detected contact

Data	**Length**
Timer start	8 bit
Epoch start	8 bit
Timer end	8 bit
Epoch end	8 bit
Address	8 bit
Total	40 bit

The contact data is not closed and written down until the contact actually ends. This type of contact record assumes that no contact is longer than maximum of timer and epoch length which is over 36 hours. For storing data on nRF51822 Flash drive it is recommended [13] that block size is at least 0x10 = 16 bytes i.e. 128 bits and it should be multiple of word size of the platform – 4 bytes. To fulfill this recommendation it is reasonable to collect multiple records before writing them to the storage. The least common denominator of 32 bits and 40 bits is 160 bits i.e. 20 bytes. In the software for current thesis the storage block size was changed to 40 bytes and 8 records were collected into volatile memory before initiating a write procedure to non-volatile memory. If a contact record length is 40 bits then 40 bytes is 8 contacts. 8 contacts should be collected to volatile memory before write procedure is initialized. The downside of this approach could be that some contacts (up to 7) could never be written. In case of small number of contacts it could affect the resulting dataset much when some data never gets written down. In case it turns out to be a problem it should be taken care with application methods, so that unwritten data would also be usable for data analysis.

4.6 RAM management

When using very limited RAM (16kB for nRF51822 versions used for current thesis) and wishing to implement more functionality, there comes a point where optimizing is needed in order to move further. Program variables can be kept in heap or stack regions in RAM. According to Paul Gribble [82] the difference of using both regions can be summarized as follows:

Stack – very fast access, don't have to explicitly de-allocate variables, space is managed efficiently by CPU, memory will not become fragmented.

Heap – (relatively) slower access, no guaranteed efficient use of space, memory may become fragmented, you must manage memory (you're in charge of allocating and freeing variables), variables can be resized using realloc().

From the heap and stack main difference it was thought by author of current paper that efficient code for the purpose of current thesis can be written without using heap at all. However, this did not succeed, because heap space was still needed for other purposes.

It is claimed that SoftDevice does not use any of the heap but it was not successful to set heap usage to 0 in link file. The attempts to reduce heap size to 0 in application resulted in device crash during initialization. The reason for this can be that some used SDK functionality may need heap space (e.g. UART logging and other libraries). The heap depends on amount and type of data used for dynamic allocation, therefore there is no mechanism to precisely determine the heap usage [83]. In order to be able to allocate storage blocks as much as possible the stack size should be increased as much as possible when using pstorage. nRF51 SDK v10 defines heap and stack size in "`gcc_startup_nrf51.s`" assembly file which should be included in assembly phase of the project building with GCC toolchain [49].

```
#ifdef __HEAP_SIZE
    .equ    Heap_Size, __HEAP_SIZE
#else
    .equ    Heap_Size, 2048
#ifdef __STACK_SIZE
    .equ    Stack_Size, __STACK_SIZE
#else
    .equ    Stack_Size, 2048
```

Figure 17 Heap and Stack definition in SDK10

The code block to define used stack and heap size in SDK10 (Figure 17) clarifies that defining appropriate symbols for assembly phase can override the allocated stack and heap size. Assembly files are not built with separate assembly compiler but with GCC with command line argument '`-x assembler-with-cpp`'. This allows to define symbols for assembly same as usual. It was tried experimentally that reducing the heap and increasing the stack in build file as shown in next code snippet increases usable pstorage size to approximately 6kB which should be enough for prototyping proof of concept although is not nearly enough for production usage.

```
ASMFLAGS +=-D__STACK_SIZE=0xB00
ASMFLAGS +=-D  HEAP SIZE=0x500
```

Figure 18 RAM optimization in project Makefile

5 Experiment

5.1 Experiment design

The experiment can be divided into 3 phases – data gathering, relaying and processing. The first phase is done in three rooms in a stone house with inner walls built from plasterboard. 4 actors are moving and stopping emulating short conversations with 2 to 4 participants. The doors between the rooms is open. Sensors are hold by actors on chest. Actors' height ranges from 1.40cm to 1.85cm. Sensor power is set to -30dBm. First phase of the experiment is recorded with stationary video camera on stand. The used sensors have BLE advertising interval set to 100ms. Scanning interval is set to 100ms with scan window of 50ms. The exact timings are varying because of radio delays introduced by SoftDevice. RSSI pruning is used for storing only signals not strictly greater than -98dBm. All the received advertisement packets are collected and saved in optimized data format and timer resolutions of 2007.843 milliseconds as explained earlier (4.5 Data structures).

In the second phase of the experiment gathered data is collected from sensors and uploaded to Google Drive service with Android Smartphone. Using Google Drive service is not mandatory, it just seemed to be a convenient service for holding files and Google Drive connector was present and handy in Android Smartphone. The files can be sent by e-mail, uploaded to FTP server or forwarded by any other means convenient. Collecting gathered data is done with nRF Toolbox app [84] and over BLE NUS service (explained in 4.3). Smartphone also records its own time and timer and epoch value of the sensor device at the same moment. This action allows to backtrack all the data records and map real clock information to all the events' timestamp. Otherwise Smartphone does no process data in any way, Smartphone acts only as a relay device – data remains in binary format and is aligned to 20 byte blocks because of limitations of NUS and BLE GATT.

In the final phase of the experiment, the data is downloaded to a PC from Google Drive using a browser. Files are processed and converted to CSV format with a written Python v3 script (Appendix 1). The records that are sharing timestamp and address (i.e. received advertisement packets from the same device during 2007.843 ms interval) are counted, average and mean RSSI calculated and combined into one record. Smartphone time and Sensor device timestamp are combined and CSV file is written with real clock date and time information instead of optimized timer ticks and epochs. CSV file has following data fields:

- Date and time value of the event(s)
- Address of the detected device
- Average RSSI of the packets
- Mean RSSI of the packets
- Count of the packets.

The data resolution decreasing approach smooths the received data and additional algorithms like EWMA (see Chapter 2.4) are not used.

5.2 Experiment results

Diagrams are created from contact data with developed script (Appendix 1) and Matplotlib library [85]. Y axis represents median RSSI in dBm of the received packets and X axis is time in minutes and seconds. The experiment took place in evening about half past eight, this is reflected on time axis. Bubble size is related to packet count received during time interval– it does not seem to correlate to distance but may correlate to subjects changing rapidly distance in relation of each other – it was not researched further.

During the minutes of 33 until 35 all the actors were standing in the circle with diameter about 1.5m and facing inside. Afterwards Actor 3 and Actor 4 were leaving side-by-side to other separate room. Actor 1 and Actor 2 remained stationary for a while after what Actor 1 left the room and joined Actor 3. Actor 3 returned and had a conversation with distance about 1.5 meters with Actor 1. Finally around 41:30 all the actors gather together and form a circle again.

Data collected from Actor 1 (Figure 19) shows that at that time all the signals are present. Actor 1 received Actor 2 signal stronger than the others. It seems from video recording that they were not closer to each other, but were standing side-by in the circle. It can be seen from the diagram that other actors leave the vicinity around 35 minutes and around 37 minutes the Actor 2 RSSI fades while Actor 3 appears. Around 39:30 Actor 3 disappears and Actor 1 remains alone until around 41:30 all the actors appear again. The actors are standing in the circle in same positions as in the beginning but still the Actor 2 RSSI is slightly higher.

Actor 2 was stationary during whole experiment. It can be seen from the diagram (Figure 20) that after the leaving of the Actor 3 and Actor 2 at about 35 minutes, only Actor 1 remained in vicinity and at around 37 Actor 1 was left alone. Around 39:30 Actor 3 reappeared and around 41:30 everybody reappears. RSSI was stronger between Actors 1 and 2 – a phenomenon that was present already on Actor 1 diagram.

Actor 3 diagram (Figure 21) had everybody in vicinity until at around 35 minutes only Actor 4 remains and shortly also disappeared. About 37:30 Actor 1 is detected and 39:30 Actor 1 is exchanged to Actor 2 and at around 41:30 everybody appeared.

Actor 4 has signal from everybody, until around 35 only Actor 3 remains and also leaves shortly as seen from the corresponding diagram (Figure 22). This is the moment where Actors 3 and 4 left the main room side-by-side and entered different rooms. It is seen that Actor 4 does not detect anybody until returning to main room at around 41:30.

From the diagrams it seems that presence of Actor 2 tends to have slightly higher RSSI for everyone. In addition to phenomenon caused by radio it might be caused from the fact that Actor 2 was the tallest and thus wore the sensor higher. It seems from diagrams that coming to vicinity begins often with row of small size of packets with low RSSI and then both increase. The same pattern can be seen with leaving but vice versa.

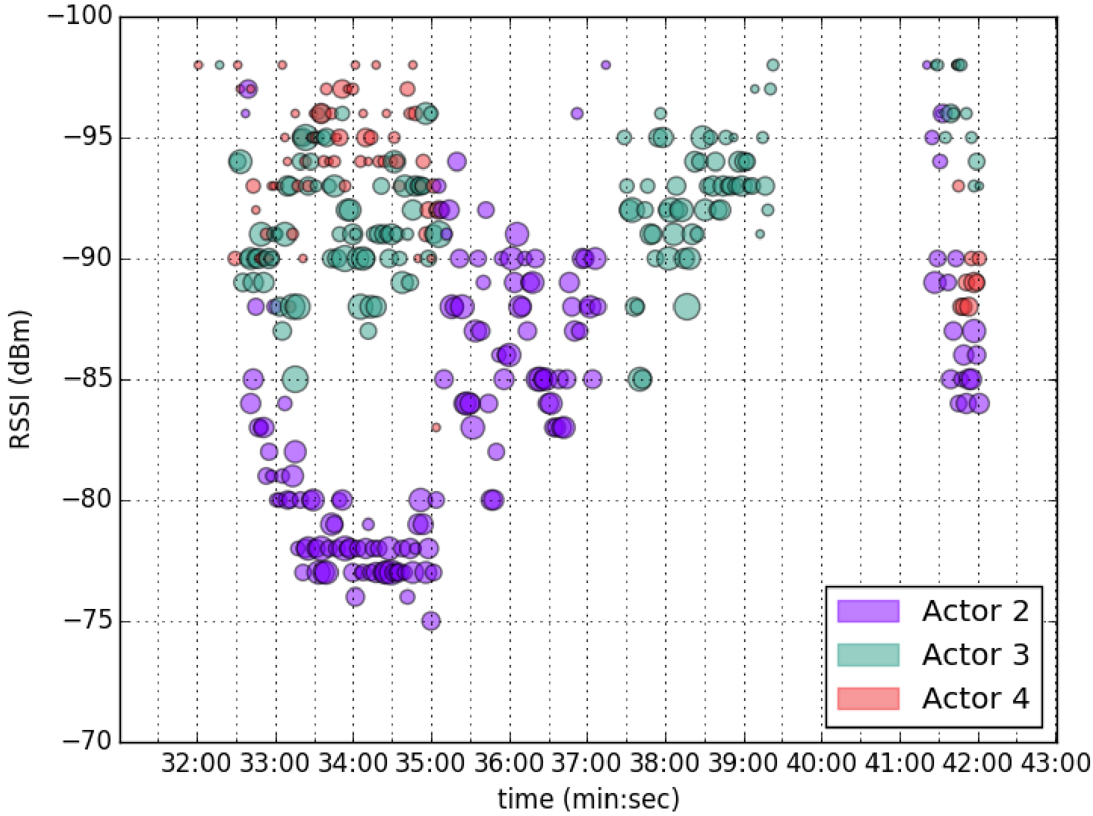

Figure 19 Experimental data from Actor 1

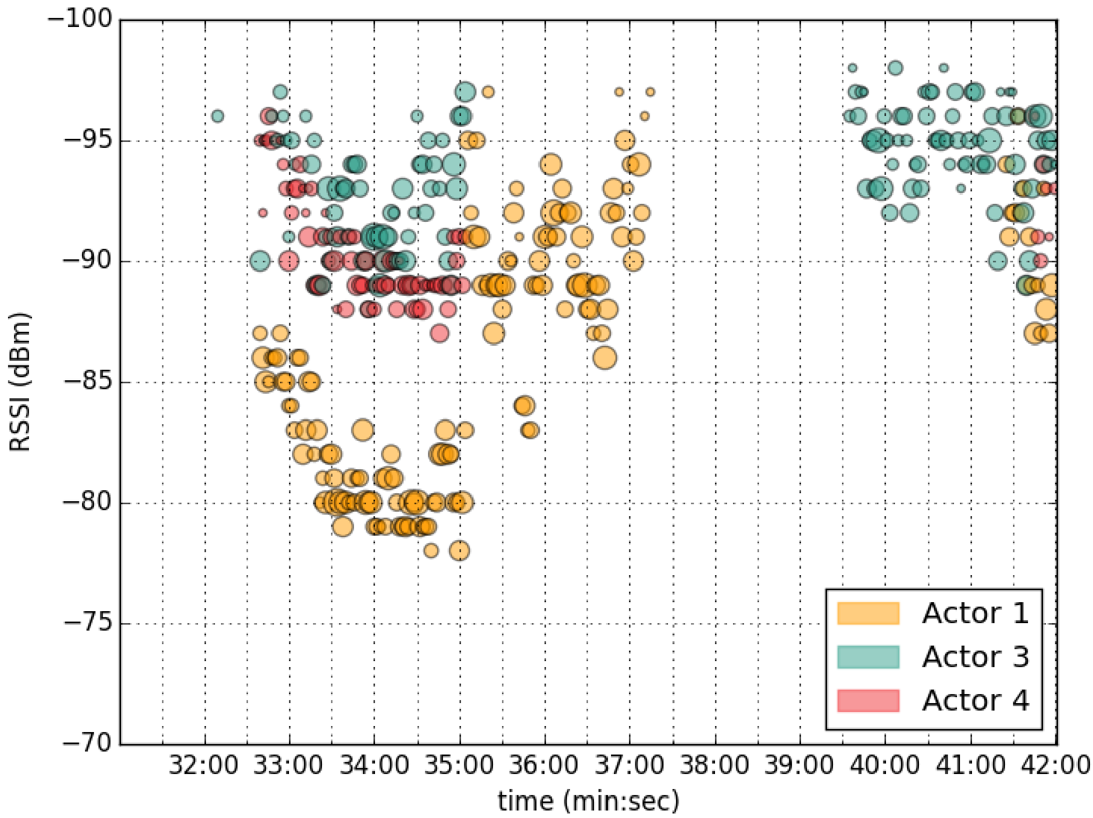

Figure 20 Experimental data from Actor 2

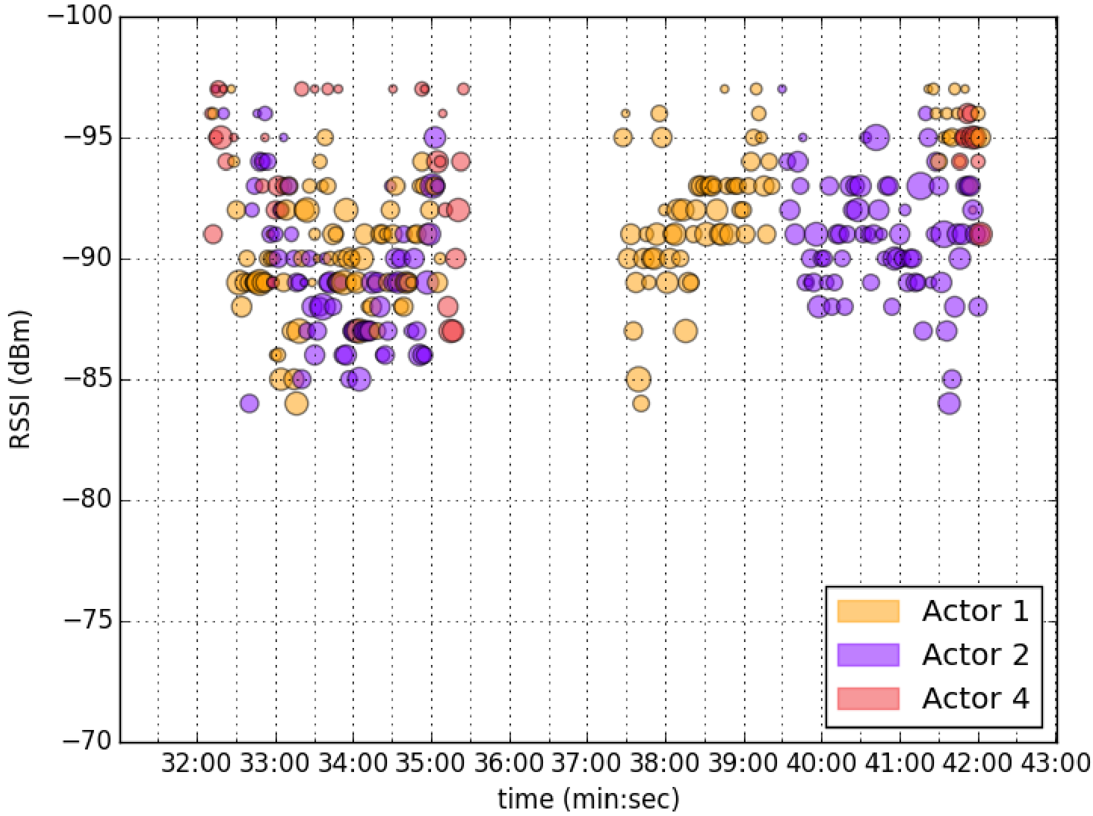

Figure 21 Experimental data from Actor 3

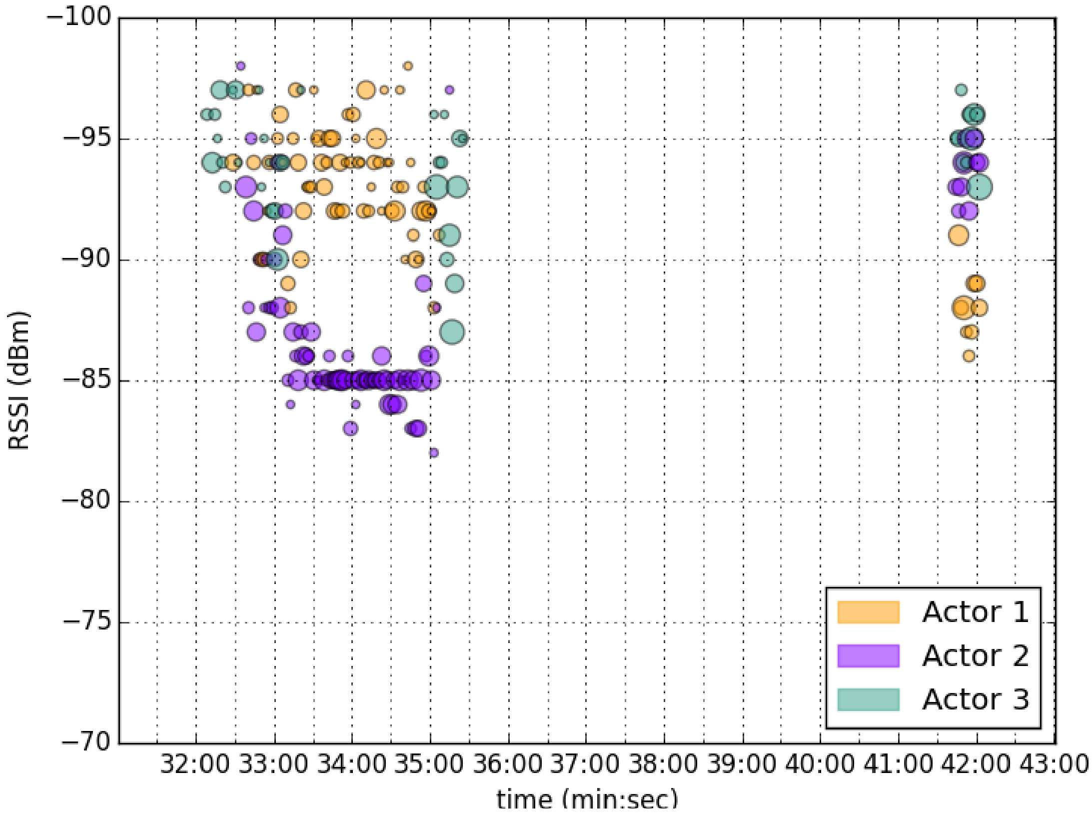

Figure 22 Experimental data from Actor 4

6 Summary

Current thesis analysed methods to monitor human contacts using Personal Sensing devices and developed own prototype solution to solve the problem. Although previous works have used different hardware and software for the same purpose – existing solutions were not successful to achieve or to reproduce with available resources. Multiple radio protocols were analysed and because of low energy requirements, wide use in IoT devices and support by multiple devices the decision was made to use BLE. For network topology author decided to use sensors for simultaneously broadcasting advertisements, scanning and storing received advertisements. This approach was chosen for independence of physical locations, simplicity and in regards of more homogenous and robust architecture. The stored contact data can be downloaded over BLE with an administrator Smartphone.

Smartphones could also be used as sensors for the given task but they tend to be bulky, heavy and costly. There are multiple BLE solutions on the market, which were analysed and one of the most widely used, light-weight, expandable and energy efficient IC-s is ARM Cortex M0 processor based nRF51822 SoC by Nordic Semiconductors. Free or low-cost alternatives were searched and found for nRF51822 firmware development and a BLE contact monitoring firmware was developed for nRF51822 SoC. To prove the concept, nRF51822 Coin Cell Radio Beacons were found and ordered from online-merchants, flashed with developed firmware and close to real life experiments conducted.

The aim that was set in Introduction chapter was met. Author analysed possibilities for detecting and monitoring human contacts and created a usable prototype solution. The solution differs from previous approaches in that the devices used are light-weight and small in size, it is fully BLE compliant, does not use stationary base stations nor physical connections and have more frequent data intervals. The solution and all the developed software is free for everyone to use as-is and available in Appendix 1 of current thesis.

Further works could port the developed prototype solution to new nRF52832 [36] SoC that was released during writing current thesis. nRF52832 has also better support for microphone input and a microphone could be utilized to detect speech and possibly emotional markers of the speech. Storage capacity should be improved for the solution devices for instance by integrating micro SD. The collected data is not highly accurate and leaves much for interpretation – contact detection could be improved and more accurate algorithms found and implemented. Security of the current prototype solution is practically non-existent and needs implementation. Ethical and Juridical side of the wearable monitoring devices could be researched and the social fears that might exist. There could also be multiple research possibilities utilizing the collected data.

References

[1] J. Knoben and L. A. G. Oerlemans, “Proximity and inter-organizational collaboration: A literature review,” *Int. J. Manag. Rev.*, vol. 8, no. 2, pp. 71–89, Jun. 2006.

[2] A. Barrat, C. Cattuto, V. Colizza, J.-F. Pinton, W. Van den Broeck, and A. Vespignani, “High Resolution Dynamical Mapping of Social Interactions With Active RFID,” *ArXiv E-Prints*, Nov. 2008.

[3] R. Guimera, B. Uzzi, J. Spiro, and L. A. N. Amaral, “Team assembly mechanisms determine collaboration network structure and team performance,” *Science*, vol. 308, no. 5722, pp. 697–702, 2005.

[4] H. Ebel, L.-I. Mielsch, and S. Bornholdt, “Scale-free topology of e-mail networks,” *Phys. Rev. E*, vol. 66, no. 3, p. 035103, 2002.

[5] B. Vaitla, “The Landscape of Big Data for Development,” 2014.

[6] “SocioPatterns.org.” [Online]. Available: http://www.sociopatterns.org/. [Accessed: 04-Mar-2016].

[7] J. Stehlé, N. Voirin, A. Barrat, C. Cattuto, L. Isella, J.-F. Pinton, M. Quaggiotto, W. Van den Broeck, C. Régis, B. Lina, and P. Vanhems, “High-Resolution Measurements of Face-to-Face Contact Patterns in a Primary School,” *PLOS ONE*, vol. 6, no. 8, p. e23176, 2011.

[8] R. Mastrandrea, J. Fournet, and A. Barrat, “Contact Patterns in a High School: A Comparison between Data Collected Using Wearable Sensors, Contact Diaries and Friendship Surveys,” *PLoS ONE*, vol. 10, no. 9, p. e0136497, 2015.

[9] A. Barrat, C. Cattuto, V. Colizza, L. Isella, C. Rizzo, A. E. Tozzi, and W. Van den Broeck, “Wearable Sensor Networks for Measuring Face-to-Face Contact Patterns in Healthcare Settings,” in *Proceedings of the 3rd International ICST Conference on Electronic Healthcare for the 21st century (eHealth 2010)*, 2011.

[10] “OpenBeacon Active RFID Project - OpenBeacon.” [Online]. Available: http://www.openbeacon.org/. [Accessed: 04-Mar-2016].

[11] “Physical Web on OpenBeacon – Bluetooth LE Tracking – Active 2.4 GHz RFID tracking.” [Online]. Available: http://get.openbeacon.org/. [Accessed: 04-Mar-2016].

[12] “CMSIS - Cortex Microcontroller Software Interface Standard - ARM.” [Online]. Available: http://www.arm.com/products/processors/cortex-m/cortex-microcontroller-software-interface-standard.php. [Accessed: 30-Mar-2016].

[13] “Nordic Semiconductor Infocenter.” [Online]. Available: http://infocenter.nordicsemi.com/index.jsp. [Accessed: 17-Mar-2016].

[14] “Altium Designer Overview.” [Online]. Available: http://www.altium.com/altium-designer/overview. [Accessed: 04-Mar-2016].

[15] “nRF51822 CoinCell design.” Nordic Semiconductors.

[16] A. Barrat, C. Cattuto, A. E. Tozzi, P. Vanhems, and N. Voirin, “Measuring contact patterns with wearable sensors: methods, data characteristics and applications to data-driven simulations of infectious diseases,” *Clin. Microbiol. Infect.*, vol. 20, no. 1, pp. 10–16, 2014.

[17] D. Olguín-Olguín and A. Pentland, “Sensor-based organisational design and engineering,” *Int. J. Organ. Des. Eng.*, vol. 1, no. 1–2, pp. 69–97, 2010.
[18] D. O. Olguin, B. N. Waber, Taemie Kim, A. Mohan, K. Ara, and A. Pentland, “Sensible Organizations: Technology and Methodology for Automatically Measuring Organizational Behavior,” *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 39, no. 1, pp. 43–55, Feb. 2009.
[19] “Humanyze.” [Online]. Available: http://www.humanyze.com/. [Accessed: 06-Mar-2016].
[20] “Radio and Telecommunication Terminal Equipment (R&TTE) Directive - European Commission.” [Online]. Available: http://ec.europa.eu/growth/sectors/electrical-engineering/rtte-directive/. [Accessed: 14-Mar-2016].
[21] “ANT+ Basics - THIS IS ANT.” [Online]. Available: https://www.thisisant.com/developer/ant-plus/ant-plus-basics. [Accessed: 04-Apr-2016].
[22] L. Frenzel, “What’s The Difference Between Bluetooth Low Energy And ANT?” Electronic Design, 29-Nov-2012.
[23] P. M. Østhus, “Concurrent operation of Bluetooth low energy and ANT wireless protocols with an embedded controller,” 2011.
[24] “nRF51822 Product Specification v3.1.” .
[25] “AR.Drone 2.0. Parrot new wi-fi quadricopter - AR.Drone.com - HD Camera - Civil drone - Parrot.” [Online]. Available: http://ardrone2.parrot.com/. [Accessed: 04-Apr-2016].
[26] “Beacons Meet WiFi: Cisco Meraki Has Its Eye on Connected Spaces | BEEKn.” [Online]. Available: http://beekn.net/2015/03/beacons-meet-wifi-cisco-meraki-eye-connected-spaces/. [Accessed: 22-Feb-2016].
[27] K. Townsend, R. Davidson, Akiba, and C. Cufí, *Getting started with Bluetooth low energy: tools and techniques for low-power networking*, Revised First Edition. Sebastopol, CA: O’Reilly, 2014.
[28] “Bluetooth Technology to Gain Longer Range Faster Speed Mesh Networking in 2016.” [Online]. Available: https://www.bluetooth.com/news/pressreleases/2015/11/11/bluetooth-technology-to-gain-longer-range-faster-speed-mesh-networking-in-2016. [Accessed: 11-Mar-2016].
[29] “NordicSemiconductor/nRF51-ble-bcast-mesh,” *GitHub*. [Online]. Available: https://github.com/NordicSemiconductor/nRF51-ble-bcast-mesh. [Accessed: 24-Jan-2016].
[30] H. L. Bertoni, *Radio propagation for modern wireless systems*. Pearson Education, 1999.
[31] S. Liu, Y. Jiang, and A. Striegel, “Face-to-Face Proximity EstimationUsing Bluetooth On Smartphones,” *IEEE Trans. Mob. Comput.*, vol. 13, no. 4, pp. 811–823, Apr. 2014.
[32] A. Ghose, C. Bhaumik, and T. Chakravarty, “BlueEye: A system for proximity detection using bluetooth on mobile phones,” in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, 2013, pp. 1135–1142.
[33] “A Guide to Selecting a Bluetooth and BLE Chipset · Argenox Technologies.” [Online]. Available: http://www.argenox.com/bluetooth-low-energy-ble-v4-0-development/library/a-guide-to-selecting-a-bluetooth-chipset/. [Accessed: 30-Apr-2016].

[34] “Misfit Flash | Sporty Fitness + Sleep Monitor.” [Online]. Available: http://misfit.com/products/flash. [Accessed: 31-Mar-2016].
[35] “BlooLoc chooses Nordic #nRF51 for indoor positioning and tracking | Blooloc.” [Online]. Available: https://www.blooloc.com/content/blooloc-chooses-nordic-nrf51-indoor-positioning-and-tracking. [Accessed: 01-Apr-2016].
[36] “nRF52832 / Bluetooth Smart/Bluetooth low energy / Products / Home - Ultra Low Power Wireless Solutions from NORDIC SEMICONDUCTOR.” [Online]. Available: https://www.nordicsemi.com/eng/Products/Bluetooth-Smart-Bluetooth-low-energy/nRF52832. [Accessed: 31-Mar-2016].
[37] “Cortex-M Series - ARM.” [Online]. Available: http://www.arm.com/products/processors/cortex-m/. [Accessed: 31-Mar-2016].
[38] “Soldering a QFN and QFP package using 2 techniques (50FPS) - YouTube.” [Online]. Available: https://www.youtube.com/watch?v=lhOKSqipwqg. [Accessed: 31-Mar-2016].
[39] “nRF51 Series Compatibility Matrix v2.1.” .
[40] “Review: Google Nexus 5X | WIRED.” [Online]. Available: http://www.wired.com/2015/10/review-google-nexus-5x/. [Accessed: 08-Mar-2016].
[41] “How to Use MMC/SDC.” [Online]. Available: http://elm-chan.org/docs/mmc/mmc_e.html. [Accessed: 08-Mar-2016].
[42] G. Mathur, P. Desnoyers, D. Ganesan, and P. Shenoy, “Ultra-low power storage for sensor networks,” 2006.
[43] “FAT File System (Windows Embedded CE 6.0).” [Online]. Available: https://msdn.microsoft.com/en-us/library/ee489982(v=winembedded.60).aspx. [Accessed: 08-Mar-2016].
[44] “Arduino - SD.” [Online]. Available: https://www.arduino.cc/en/Reference/SD. [Accessed: 08-Mar-2016].
[45] “SDFileSystem - a mercurial repository | mbed.” [Online]. Available: https://developer.mbed.org/teams/mbed/code/SDFileSystem/. [Accessed: 08-Mar-2016].
[46] “FatFs - Generic FAT File System Module.” [Online]. Available: http://elm-chan.org/fsw/ff/00index_e.html. [Accessed: 08-Mar-2016].
[47] “developer.nordicsemi.com:/.” [Online]. Available: https://developer.nordicsemi.com/. [Accessed: 30-Mar-2016].
[48] “Keil Embedded Development Tools for ARM, Cortex-M, Cortex-R4, 8051, C166, and 251 processor families.” [Online]. Available: http://www.keil.com/. [Accessed: 30-Mar-2016].
[49] “GCC ARM Embedded in Launchpad.” [Online]. Available: https://launchpad.net/gcc-arm-embedded. [Accessed: 30-Mar-2016].
[50] “Eclipse - The Eclipse Foundation open source community website.” [Online]. Available: http://www.eclipse.org/. [Accessed: 30-Mar-2016].
[51] “Tutorials - Nordic Developer Zone - Development with GCC and Eclipse.” [Online]. Available: https://devzone.nordicsemi.com/tutorials/7/. [Accessed: 24-Jan-2016].
[52] “SEGGER - The Embedded Experts - J-Link Debug Probes - Model Overview - Other J-Links - J-Link OB.” [Online]. Available: https://www.segger.com/jlink-ob.html. [Accessed: 30-Mar-2016].
[53] “nRF51 DK / Products / Home - Ultra Low Power Wireless Solutions from NORDIC SEMICONDUCTOR.” [Online]. Available: https://www.nordicsemi.com/eng/Products/nRF51-DK. [Accessed: 30-Mar-2016].

[54] “nRF51822 Development Kit / Bluetooth Smart/Bluetooth low energy / Products / Home - Ultra Low Power Wireless Solutions from NORDIC SEMICONDUCTOR.” [Online]. Available: https://www.nordicsemi.com/eng/Products/Bluetooth-Smart-Bluetooth-low-energy/nRF51822-Development-Kit. [Accessed: 30-Mar-2016].
[55] “nRF51822 Evaluation Kit / Bluetooth Smart/Bluetooth low energy / Products / Home - Ultra Low Power Wireless Solutions from NORDIC SEMICONDUCTOR.” [Online]. Available: https://www.nordicsemi.com/eng/Products/Bluetooth-Smart-Bluetooth-low-energy/nRF51822-Evaluation-Kit. [Accessed: 30-Mar-2016].
[56] “nRF51 Dongle / Products / Home - Ultra Low Power Wireless Solutions from NORDIC SEMICONDUCTOR.” [Online]. Available: https://www.nordicsemi.com/eng/Products/nRF51-Dongle. [Accessed: 30-Mar-2016].
[57] “Blogs - Nordic Developer Zone.” [Online]. Available: https://devzone.nordicsemi.com/blogs/686/programming-the-nrf51822-beacon-kit-with-the-nrf51/. [Accessed: 29-Mar-2016].
[58] “SEGGER - The Embedded Experts - J-Link Debug Probes - Model Overview.” [Online]. Available: https://www.segger.com/jlink-model-overview.html. [Accessed: 30-Mar-2016].
[59] “SEGGER - The Embedded Experts - Purchase - Pricing - Price list J-Link related.” [Online]. Available: https://www.segger.com/pricelist-j-link-related.html. [Accessed: 30-Mar-2016].
[60] “Icarus Sensor System.” [Online]. Available: http://icarus-sensors.github.io/general/starting-with-nRF51822.html. [Accessed: 30-Mar-2016].
[61] “ST-LINK/V2 ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32 - STMicroelectronics.” [Online]. Available: http://www.st.com/web/catalog/tools/FM146/CL1984/SC724/SS1677/PF251168. [Accessed: 30-Mar-2016].
[62] “SRecord 1.64.” [Online]. Available: http://srecord.sourceforge.net/. [Accessed: 30-Mar-2016].
[63] “Home | mbed.” [Online]. Available: https://www.mbed.com/en/. [Accessed: 30-Mar-2016].
[64] “Open On-Chip Debugger.” .
[65] “Home - STMicroelectronics.” [Online]. Available: http://www2.st.com/content/st_com/en.html. [Accessed: 30-Mar-2016].
[66] “ST-LINK in-circuit debugger/programmer for STM8 and STM32 microcontrollers - CD00221563.pdf.” [Online]. Available: http://www.st.com/st-web-ui/static/active/en/resource/technical/document/user_manual/CD00221563.pdf. [Accessed: 31-Mar-2016].
[67] “nrf51822-macosx-build-scripts/Makefile.OpenOCD at master · sprhawk/nrf51822-macosx-build-scripts.” [Online]. Available: https://github.com/sprhawk/nrf51822-macosx-build-scripts/blob/master/Makefile.OpenOCD. [Accessed: 02-Feb-2016].
[68] “The OpenOCD debugging Eclipse plug-in.” [Online]. Available: http://gnuarmeclipse.github.io/debug/openocd/. [Accessed: 30-Mar-2016].
[69] “NRF51822 Eval Kit BLE4.0 Bluetooth 2.4G Development/Evaluation Kit designed for nRF51822.” [Online]. Available: http://www.waveshare.com/NRF51822-Eval-Kit.htm. [Accessed: 30-Mar-2016].

[70] "NRF51822 Eval Kit - Waveshare Wiki." [Online]. Available: http://www.waveshare.com/wiki/NRF51822_Eval_Kit. [Accessed: 29-Mar-2016].
[71] "Nordic Semiconductor PCA10001 Board Support Package." [Online]. Available: http://www.rowleydownload.co.uk/arm/packages/Nordic_PCA10001.htm. [Accessed: 30-Mar-2016].
[72] "USB to UART Bridge VCP Drivers | Silicon Labs." [Online]. Available: https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx . [Accessed: 30-Mar-2016].
[73] "STSW-LINK004 STM32 ST-LINK utility - STMicroelectronics." [Online]. Available: http://www.st.com/web/en/catalog/tools/PF258168. [Accessed: 30-Mar-2016].
[74] "CadSoft EAGLE PCB Design Software|Support, Tutorials, Shop." [Online]. Available: http://www.cadsoftusa.com/. [Accessed: 05-Mar-2016].
[75] "Eurocircuits - Online PCB Prototype and small series specialist." [Online]. Available: http://www.eurocircuits.com/. [Accessed: 03-May-2016].
[76] "NRF51822 Beacon-SHENZHEN RadioLand Technology CO.,LTD." [Online]. Available: http://radioland-china.com/English/Product/6035982553.html. [Accessed: 30-Apr-2016].
[77] "Bluetooth.org." [Online]. Available: https://developer.bluetooth.org/gatt/profiles/Pages/ProfileViewer.aspx?u=org.bluetooth.profile.proximity.xml. [Accessed: 10-Mar-2016].
[78] "S130 SoftDevice Specification v0.5." .
[79] "Tutorials - Nordic Developer Zone - Setting up the Timeslot API." [Online]. Available: https://devzone.nordicsemi.com/tutorials/16/. [Accessed: 12-Feb-2016].
[80] "NordicSemiconductor/nRF51-multi-role-conn-observer-advertiser: Runs an Observer or Advertiser role (timeslot API) with a Connection. Provides a HCI interface to the Observer and Advertiser. Advertiser provides SCAN REQs back to application." [Online]. Available: https://github.com/NordicSemiconductor/nRF51-multi-role-conn-observer-advertiser. [Accessed: 04-Apr-2016].
[81] "Write Application Data to Flash with pstorage - Nordic Developer Zone." [Online]. Available: https://devzone.nordicsemi.com/question/14569/write-application-data-to-flash-with-pstorage/. [Accessed: 20-Mar-2016].
[82] "7. Memory : Stack vs Heap." [Online]. Available: http://gribblelab.org/CBootcamp/7_Memory_Stack_vs_Heap.html. [Accessed: 23-Apr-2016].
[83] "Stack and Heap size requirements - Texas Instruments Wiki." [Online]. Available: http://processors.wiki.ti.com/index.php/Stack_and_Heap_size_requirements. [Accessed: 23-Apr-2016].
[84] "nRF Toolbox App / Nordic mobile Apps / Products / Home - Ultra Low Power Wireless Solutions from NORDIC SEMICONDUCTOR." [Online]. Available: https://www.nordicsemi.com/eng/Products/Nordic-mobile-Apps/nRF-Toolbox-App. [Accessed: 01-May-2016].
[85] "matplotlib: python plotting — Matplotlib 1.5.1 documentation." [Online]. Available: http://matplotlib.org/. [Accessed: 03-May-2016].

Appendix 1 – Created Software

The software created for current thesis is available from GitHub repository https://github.com/shunran/ble-mon