BRNO UNIVERSITY OF TECHNOLOGY

Faculty of Electrical Engineering
and Communication

MASTER'S THESIS

Brno, 2020 Bc. Filip Šterc

BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF RADIOENGINEERING
ÚSTAV RADIOELEKTRONIKY

BLUETOOTH® LOW ENERGY VEHICLE KEYLESS ENTRY
BEZKLÍČOVÝ PŘÍSTUP DO VOZIDLA POMOCÍ TECHNOLOGIE BLUETOOTH LOW ENERGY

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR — **Bc. Filip Šterc**
AUTOR PRÁCE

SUPERVISOR — **doc. Ing. Tomáš Frýza, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2020

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

Master's Thesis

Master's study field **Electronics and Communication**

Department of Radioengineering

Student: Bc. Filip Šterc ***ID:*** 186210

Year of study: 2 ***Academic year:*** 2019/20

TITLE OF THESIS:

Bluetooth® Low Energy Vehicle Keyless Entry

INSTRUCTION:

The goal of the thesis is to study and understand the fundamentals of vehicle keyless entry systems, Bluetooth low energy (BLE) technology and state-of-the-art Bluetooth radio and in-vehicle networking (IVN) transceivers integrated circuits from ON Semiconductor. Conduct a feasibility study on implementation of automatic unlock feature based on Bluetooth-enabled smartphone to vehicle proximity detection. Using existing BLE, LIN transceiver and LIN-based system basis chip (SBC) evaluation boards, build a basic proof-of-concept system, develop necessary code to enable individual modules data exchange, verify fundamental function and explore capabilities of such a system.

Based on information acquired, design a demonstration kit for vehicle keyless entry combining BLE and IVN technology. The demonstration kit shall allow the user to lock and unlock the door using a regular smartphone with authorized identification key in close vicinity of the vehicle. The system should comprise ON Semiconductor BLE and door lock module integrated circuits, both connected to body control module simulator using suitable IVN bus (CAN/LIN).

RECOMMENDED LITERATURE:

[1] Bluetooth SIG, Inc. Bluetooth Core Specification v5.1 [online]. 21. Leden 2019. https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=457080

[2] ON Semiconductor. RSL10 Bluetooth 5 Radio System-on-Chip (SoC) [online]. Květen 2018, Rev. 3. https://www.onsemi.com/pub/Collateral/RSL10-D.PDF

Date of project specification: 3.2.2020 ***Deadline for submission:*** 28.5.2020

Supervisor: doc. Ing. Tomáš Frýza, Ph.D.
Consultant: Ing. Filip Brtan (ON Semiconductor)

prof. Ing. Tomáš Kratochvíl, Ph.D.
Subject Council chairman

ABSTRACT

This paper deals with the design of demonstration kit for Vehicle Keyless Entry, employing the Bluetooth® Low Energy (BLE) technology and state-of-the-art Bluetooth radio, and In-Vehicle Networking (IVN) transceivers integrated circuits from ON Semiconductor. The demonstration kit enables manual, as well as automatic locking and unlocking of a car door with a remote key realized by a smartphone. The solution deals with a proximity detection of a remote key, security of BLE connection and communication over LIN bus between modules of the demonstration kit.

KEYWORDS

Bluetooth, Keyless Entry, door lock, transceiver, proximity, security, connection, bus, power

ABSTRAKT

Tato práce se zabývá návrhem demonstračního kitu bezlíčového přístupu pro vozidla s využitím nejnovějších integrovaných obvodů Bluetooth® Low Energy a budičů automobilové sběrnice od společnosti ON Semiconductor. Aplikace umožňuje ruční i automatické odemykání dveří vozidla ze vzdáleného klíče realizovaného chytrým telefonem. Řešení se zabývá detekcí vzdálenosti vzdáleného klíče, bezpečností BLE spojení a komunikací přes automobilovou sběrnici LIN mezi moduly demonstračního kitu.

KLÍČOVÁ SLOVA

Bluetooth, bezklíčový přístup, zámek dveří, budič sběrnicem, vzdálenost, bezpečnost, spojení, sběrnice, výkon

ŠTERC, Filip. *Bluetooth® Low Energy Vehicle Keyless Entry*. Brno, Rok, 66 p. Master's Thesis. Brno University of Technology, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. Advised by doc. Ing. Tomáš Frýza, Ph.D.

ROZŠÍŘENÝ ABSTRAKT

Bezklíčový přístup do auta patří mezi velmi rozšířené systémy používané ve vozidlech v současné době. Tento systém umožňuje bezdrátovému zařízení vykonávat funkci klíče bez nutnosti fyzického kontaktu s vozidlem. Podporuje možnost zamknutí a odemknutí vozidla pomocí vyhrazených tlačítek na vzdáleném klíči, případně automaticky na základě vzdálenosti od vozidla. Protože bezklíčový přístup umožňuje přístup do auta, jednou z kritických vlastností tohoto systému je zabezpečení spojení.

Cílem této práce je vytvoření demonstračního kitu pro aplikaci bezklíčového přístupu ve vozidle. Tento systém se skládá z modulu bezklíčového přístupu, který zprostředkovává komunikaci mezi vzdáleným klíčem a vozidlem. Dále se skládá z BCM (Body Control Module), který v autě řídí chování dveřních periferií, a modulu ovladače dveřního zámku. Všechny tyto moduly vzájemně komunikují pomocí LIN (Local Interconnect Network) sběrnice, kde BCM vykonává práci mastera, který řídí a inicializuje komunikaci s oběma moduly a překládá požadavky z jednoho modulu na druhý.

Modul bezklíčového přístupu vyžaduje pro komunikaci se zařízením vzdáleného klíče radiofrekvenční vysílač na krátkou vzdálenost. Toto je realizováno technologií Bluetooth® Low Energy (BLE), za pomoci integrovaného obvodu RSL10 společnosti ON Semiconductor. Tento obvod obsahuje kompletní Bluetooth RF Front-End, stejně tak jako procesorové jádro ARM Cortex-M3. Tento obvod je vybrán z toho důvodu, že nevyžaduje žádné další aktivní komponenty pro realizaci bezklíčového přístupu, s výjimkou LIN budiče sběrnice pro komunikaci s BCM a také nabízí bezkonkurenčně nejnižší spotřebu v porovnání s ostatními komerčně dostupnými BLE obvody pro automotive na trhu. Jakožto budič sběrnice je použit NCV7428 LIN SBC (System Basis Chip), který kombinuje budič sběrnice s LDO (Low-drop Voltage Regulator) s výstupem 3,3 V pro napájení RSL10.

BCM v tomto demonstračním kitu vykonává funkci překladače LIN zpráv mezi modulem bezklíčového přístupu a ovladačem dveřního zámku. BCM periodicky čte status obou těchto modulů a na jejich základu vysílá požadavky pro zamykání a odemykání dveří do ovladače dveřního zámku a vysílá informace o stavu zámku zpět do modulu bezklíčového přístupu. Jakožto hardware pro realizaci BCM je v této práci použito existující PCB LIN_GW_V1 (příloha B). PCB (Printed Circuit Board) se skládá z procesoru od firmy Microchip AT32UC3C, budiče LIN sběrnice NCV7321 a LDO NCV4274 od společnosti ON Semiconductors s výstupem 3,3 V.

Modul ovladače dveřního zámku slouží k přímému zamykání a odemykání dveřního zámku u auta. PCB se opět skládá z procesoru AT32UC3C a LIN SBC NCV7428, tentokrát s výstupem pro napájení 5 V. Samotný ovladač je tvořen integrovaným obvodem NCV7710, který je navržen pro tuto aplikaci a obsahuje kompletní ovladač plného můstku včetně tranzistorů pro ovládání motoru dveřního zámku. Výstup tohoto můstku je řízen pomocí registrů přes SPI (Serial Peripheral Interface) sběrnici.

Zabezpečení BLE komunikace proti odposlouchávání zajišťuje šifrování pomocí AES-CCM (Advanced Encryption Standard - Cipher block Chaining - Message authentication code) používané obecně pro veškerou komunikaci pomocí Bluetooth® Low Energy. Kritickým okamžikem je však předávání klíčů potřebných k tomuto šifrování a autentizaci klíče. Z tohoto důvodu je komunikace mezi modulem bezklíčového přístupu a vzdáleným klíčem rozdělena do dvou módů. Normální mód je použit pro běžnou komunikaci za účelem odemykání a zamykání dveřního zámku. V tomto módu bezklíčový modul komunikuje pomocí rozlišitelných (resolvable) adres a ověřuje autentizaci vzdáleného klíče rozlišováním adresy pomocí IRK (Identity Resolving Key). V případě, že adresa příchozí zprávy byla vygenerována pomocí neznámého IRK, bezklíčový modul tuto zprávu zahodí. IRK se společně s dalšími enkrypčními klíči předává během párování vzdáleného klíče s bezklíčovým modulem. Pro tento účel existuje druhý, párovací mód. V tomto módu bezklíčový modul umožňuje komunikaci jakýmkoliv zařízením, přičemž po spojení se vzdáleným klíčem dojde ke spárování, během kterého si zařízení vymění bezpečnostní údaje. Pro zachování bezpečnosti by k tomuto módu mělo být přistupováno pouze v kontrolovaném prostředí, během prvního připojování vzdáleného klíče.

Jednou z vlastností bezklíčového systému je také určování vzdálenosti na základě přijatého výkonu. Vzdálenost je v tomto případě rozdělena do tří oblastí: bezprostřední (jednotky metrů), blízká a vzdálená (více než 10 metrů). Měřením byly tyto oblasti na základě přijatého výkonu stanovené následovně: bezprostřední $> -60\,\mathrm{dBm}$, blízká $-60\,\mathrm{dBm}$ až $-80\,\mathrm{dBm}$ a vzdálená $< -80\,\mathrm{dBm}$. Funkce automatického zamykání a odemykání dveří umožnuje odemknutí dveří v případě přechodu do bezprostřední vzdálenosti a zamknutí v případě přechodu do vzdálené oblasti. Blízká vzdálenost zde vytváří hysterezní oblast. Určování vzdálenosti na základě přijatého výkonu však vytváří skulinu pro tzv. útok dvou zlodějů, během kterého dva útočníci vytvoří most pro komunikaci mezi vozidlem a vzdáleným klíčem. První útočník přijme signál vyslaný vozidlem a zesílí ho, druhý útočník ho následně vyšle poblíž vzdáleného klíče. Stejným způsobem poté

přenesou odpověď klíče k vozidlu. V tomto případě není šifrování ani autentizace nijak narušena, pouze je zvýšen přijímaný výkon, kterým je určena vzdálenost. V této práci je však klíč tvořen chytrým telefonem, který se odpojí od vozidla v případě zavření aplikace vzdáleného klíče, tudíž nedochází k nepřetržitému vysílání, které by šlo takto zneužít. Zároveň je v aplikaci vzdáleného klíče implementovaná možnost funkci automatického odemykání zcela vypnout.

Poslední částí této práce je program pro simulaci funkce BCM. Demostrační kit je možné předvádět samostatně pouze s nutností připojení na napájení 12 V. V tomto případě je, ale nutné použít tlačítka na modulu bezklíčového přístupu pro vyvolání některých funkcí jako párovací mód, či vymazání paměti s autentizačními klíči. Program simulace BCM tedy existuje pro demostrování reálné funkce, kdy veškeré chování modulu bezklíčového přístupu je řízeno z BCM přes sběrnici LIN. Připojením BCM PCB k počítači s tímto programem přes rozhraní USB (Universal Serial Bus) je možné definovat zprávy, které BCM vysílá na sběrnici a ovládat obecné chování bezklíčového systému.

DECLARATION

I declare that I have written the Master's Thesis titled "Bluetooth® Low Energy Vehicle Keyless Entry" independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the thesis and listed in the comprehensive bibliography at the end of the thesis.

As the author I furthermore declare that, with respect to the creation of this Master's Thesis, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll., Section 2, Head VI, Part 4.

Brno ....

author's signature

ACKNOWLEDGEMENT

I would like to thank my consultant Ing. Filip Brtáň for guidance, suggestions, and help with any problems emerging with the solution.
I would also like to thank my advisor doc. Ing. Tomáš Frýza, Ph.D. for the useful comments, remarks and help with the completion of my thesis.

Contents

List of Figures

Introduction

Keyless entry is one of the most frequently used modern systems in car vehicles today. This system enables a remote key to perform the function of a key without the need for physical contact with the vehicle. It supports locking and unlocking of the car doors by pressing assigned buttons on a remote key itself or automatically based on proximity to the car.

The Keyless entry system requires a short-range radio transceiver for communication with a remote key device. Radio transceivers typically used for Keyless entry systems operate at 433.92 MHz for the European and Asian cars and at 315 MHz for North American cars, using Rolling code encryption for communication between the vehicle and the remote key. In this thesis, a Bluetooth® Low Energy wireless transceiver is used, operating at 2.4 GHz. This enables to use security procedures defined in Bluetooth specification for protection against attacks, and to develop a remote key application for commonly used devices as smart phones, tablets, etc.

In Tab. 1, there are listed BLE transceivers combined with Arm® Cortex® Processor Core for automotive, currently available on the market. From these devices, the RSL10 from ON Semiconductor has the lowest current consumption, making it the most suitable for low energy applications. A small disadvantage of ON Semiconductor's solution, in comparison to MKW36Z from NXP, is the absence of hardware support for LIN driver. Nevertheless LIN communication controller software stack can be based on classical UART (Universal Asynchronous Receiver-Transmitter) peripheral that is implemented in RSL10.

Device	Supply Voltage	Peak Current (TX)	Peak Current (RX)	UART with LIN support
RSL10 (ON semi.)	1.1 V to 3.3 V	4.6 mA	3.0 mA	no
MKW36Z (NXP)	1.71 V to 3.6 V	5.7 mA	6.3 mA	yes
CC2642R-Q1 (TI)	1.8 V to 3.63 V	7.3 mA	6.9 mA	no
nRF51824 (Nordic semi.)	1.9 V to 3.6 V	9.7 mA	8 mA	no

Tab. 1: Parameter comparison of available BLE devices.

As the BLE module requires 3.3 V voltage supply, it calls for a LDO (Low-drop Voltage Regulator) converting 12 V battery supply to lower voltage. For this reason, it is convenient to use an SBC (System Basis Chip) combining both LIN transceiver and LDO. ON Semiconductor offers NCV7428 LIN SBC integrating 3.3 V / 70 mA LDO voltage regulator and LIN transceiver in a small DFN8 package. Similar products are available on the market and can be used as well.

To demonstrate the function of Keyless entry, Keyless Entry module needs to be connected to a BCM unit and to a motor driver that controls a door lock. BCM has a role of LIN master and other modules are LIN slaves. In the demonstration kit, the BCM is going to be realized by a microcontroller with a LIN transceiver and a USB (Universal Serial Bus) interface, enabling for standalone function, as well as remote control from GUI on PC. The door lock driver is going to be realized by an integrated circuit NCV7710 designed for this purpose, communicating with a microcontroller over SPI.

Proposed module diagram for realization of Keyless Entry is shown in Fig. 1.

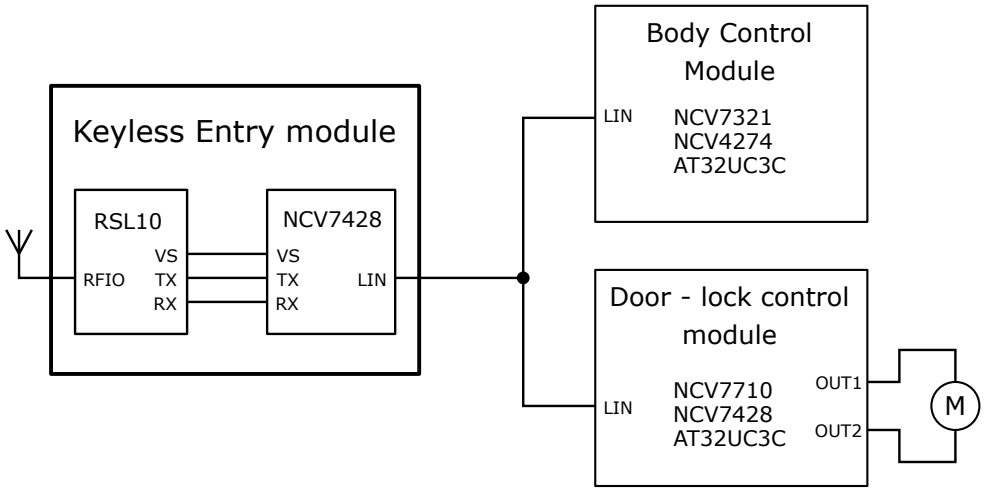


Fig. 1: Block diagram of proposed block connection.

1 Theory

This chapter is dedicated to the comparison of Keyless Entry systems used today, summarization of Bluetooth® Low Energy specification for necessary information needed to understand the functionality of Bluetooth, and to the explanation of the way LIN communication protocol works.

1.1 Keyless Entry system comparison

The most commonly used communication technique in Keyless Entry system is Rolling Code encryption, used to evade replay attacks [1]. The security of Rolling Code encryption is realized by including a value of a counter in the message, which is then encrypted with the use of a secret key. In receiver, the message is decrypted and the value of the counter is checked if it is in an acceptable range of values. This creates dynamic transmission, which offers resistance against recording and later replaying of the communication to get access to the vehicle. For higher security, the counter used in the Rolling Code encryption may be replaced with a pseudo-random number generator [1].

But the Rolling Code encryption may be vulnerable to a scan attacks [2]. In scan attack, the attacker is continuously sending different codes, trying to match the code of the vehicle transceiver. Another encryption technique, which offers higher resistance against these attacks is Challenge Response. The Challenge Response technique utilizes a secret key shared between the transceivers. When a request is received, the vehicle creates a random number and sends it to the remote key. Remote key encrypts this random number and sends it back to the vehicle. The vehicle also encrypts the random number and compares it with the received one. When the numbers match, the request is served [2].

In this thesis, a Bluetooth® Low Energy (BLE) communication protocol is used. In contrast with the previous Keyless Entry communication techniques, the BLE communication requires an establishment of a connection before any data and requests can be transmitted. This provides a protection against scan attacks. Furthermore the BLE utilizes a data signature with the use of a Connection Signature Resolving Key (CSRK), which provides also protection against replay attacks [3].

1.2 BLE Specification

Bluetooth® Low Energy stack consists of several layers shown in Fig. 1.1. This section will mainly focus on GAP (Generic Access Profile), GATT (Generic Attribute Profile) and Profile layers, together with Link Layer handling of connection through White List and Resolving List. These layers are responsible for connection management between devices as well as sharing and storing of information.

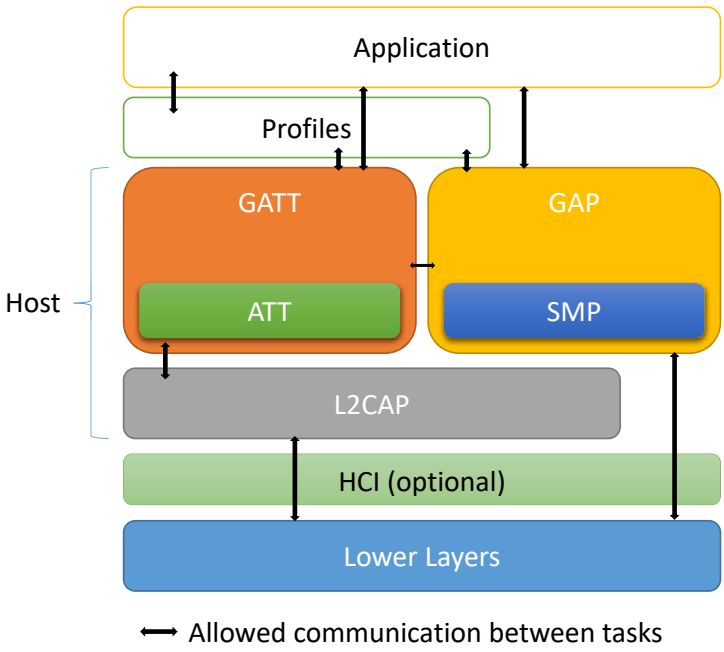


Fig. 1.1: Bluetooth® Low Energy Layers [4].

1.2.1 Generic Access Profile

Generic Access Profile (GAP) [3] is a base profile implemented in all Bluetooth devices and defines the basic requirements of the device. It also defines the methods for device discovery, connection establishment, association models and service discovery. GAP defines four specific roles: Broadcaster, Observer, Peripheral and Central.

Broadcaster is a role which supports transmitter only applications. The Broadcaster device uses advertising to broadcast data, and never creates connections with other devices.

Observer is a role which supports receiver only applications. The Observer device is complementary to the Broadcaster device, and only receives data broadcasted over advertisements.

Peripheral and **Central** roles are only roles that support connection establishment. The Peripheral device acts as a Controller's slave and usually supports single connection. The Central device is the initiator for all connection with the Peripheral devices and supports multiple connections. It also acts as a Controller's master.

Device discovery [3] comprise the advertising procedure and the scanning procedure to discover (or to be discovered by) other Bluetooth devices in nearby area.

The advertising is a procedure used for data broadcasts, with the goal of presenting itself to the nearby devices. This data contains basic information about the device such as name, company ID, list of services, etc. It may be used for connection establishment or to periodically broadcast data to the scanning devices.

The scanning is a procedure used by a scanning device to listen for broadcasts from an advertising device. A scanning device can send a scan request to an advertiser, which sends back a scan response containing additional user data. If a scanning device is in the initiator mode it can initiate a connection by sending a connection request.

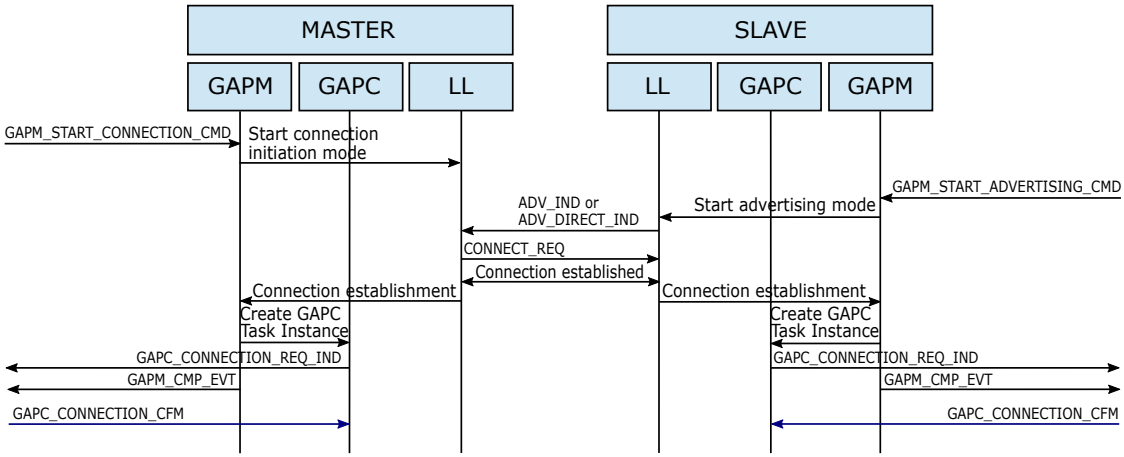


Fig. 1.2: Connection establishment overview [4].

Connection establishment [3] happens when a scanning device in the initiator mode sends a connection request to an advertiser in a connectable mode. Upon connection establishment, both of the devices cease their device discovery procedure

and switch to the connected mode. In this mode, a physical link is created between the two devices and enables them to share a service data over GATT. It is possible to re-enter advertising or scanning procedure in the connected mode, while maintaining the connection. A possible connection establishment scenario is shown in Fig. 1.2.

1.2.2 Generic Attribute Profile

The *Generic Attribute Profile* (GATT) [3] specifies the structure in which profile data is exchanged. This structure defines basic elements such as services and characteristics, used in a profile. GATT lies above the ATT (Attribute Protocol) and acts as a gateway to discover, read and write attributes stored in the server attribute database. GATT defines two roles: GATT Client and GATT Server.

The GATT client sends request and commands to the GATT server over the Attribute Protocol. The GATT server stores the received data, and can be configured to send notifications and indication to the GATT client on a specific events.

Services and Characteristics [3] are units defined within the attribute database, and represent basic functionalities and features of the device. A service is made of a set of characteristics. Each characteristic contains a value, properties defining its capabilities, and may contain descriptor for additional information about the value.

A GATT client can discover services available on the server by performing a service discovery. Each service and characteristic is recognized by its 128-bit UUID (Universally Unique Identifier), which can be also represented by a 16-bit or a 32-bit version to simplify its transfer and storing. Each version of UUID can be computed from others by (1.1) and (1.2).

Conversion of UUID types by specification [3]:
Bluetooth Base UUID: 00000000-0000-1000-8000-00805F9B34FB

$$UUID_{128\text{-}bit} = UUID_{16\text{-}bit} \cdot 2^{96} + Bluetooth\_Base\_UUID \tag{1.1}$$

$$UUID_{128\text{-}bit} = UUID_{32\text{-}bit} \cdot 2^{96} + Bluetooth\_Base\_UUID \tag{1.2}$$

Service discovery loads UUIDs and properties of every service and characteristic available within the server attribute database. Bluetooth® Low Energy specification describes numerous services and profiles pre-defined for a generic Bluetooth device, enabling compatibility between individual devices.

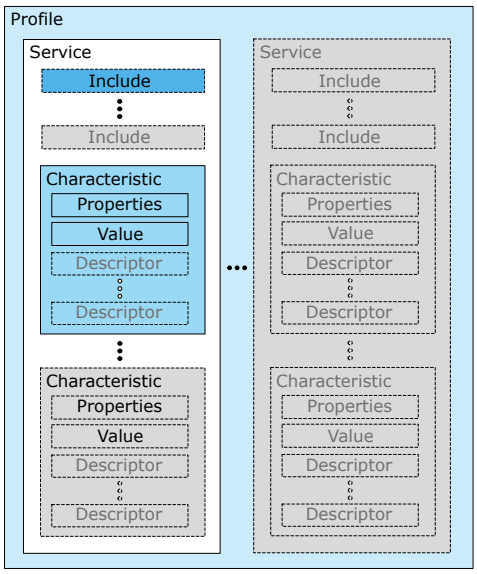


Fig. 1.3: Profile hierarchy [3].

Profiles [3] are a bundles of services designed to perform a certain features used in devices, such as Proximity profile, Hearth Rate profile, etc. Profile hierarchy can be seen in Fig. 1.3

1.2.3 Addresses and device identification

In BLE protocol addresses are used for communication and identification of a remote device. Addresses used by BLE can be divided into 4 types [3]. Public address of the device and Private Static address are used for identification of the device and may be used for communication. The other 2 private address types are used only for communication and they are changed every 15 minutes by default. These addresses are Non-resolvable and Resolvable addresses. Non-resolvable address is generated as a random number and is used only for non-connectable applications such as Bluetooth beacons. Resolvable address has only a higher half of address generated randomly and rest is computed with *Identity Resolving Key* (IRK) by a hash function [3]. Each type of Private address can be recognized by 2 MSBs (Most Significant Bit) (see Fig. 1.4), which are given for them by specification [3].

When connecting with the usage of Resolvable address, devices may request pairing, and exchange their identity information (identity address and IRK) together with security information for encrypted communication. When paired, devices are able to recognize each other even after regeneration of Resolvable address by applying IRK

from bond list (list of paired devices with their information) to received address. This resolving procedure can be managed either by GAP - Host Managed Privacy, or directly by Link Layer - Controller Managed Privacy. When the resolving is performed by Link Layer, received address is replaced by identification address and the message is sent to a higher layer.

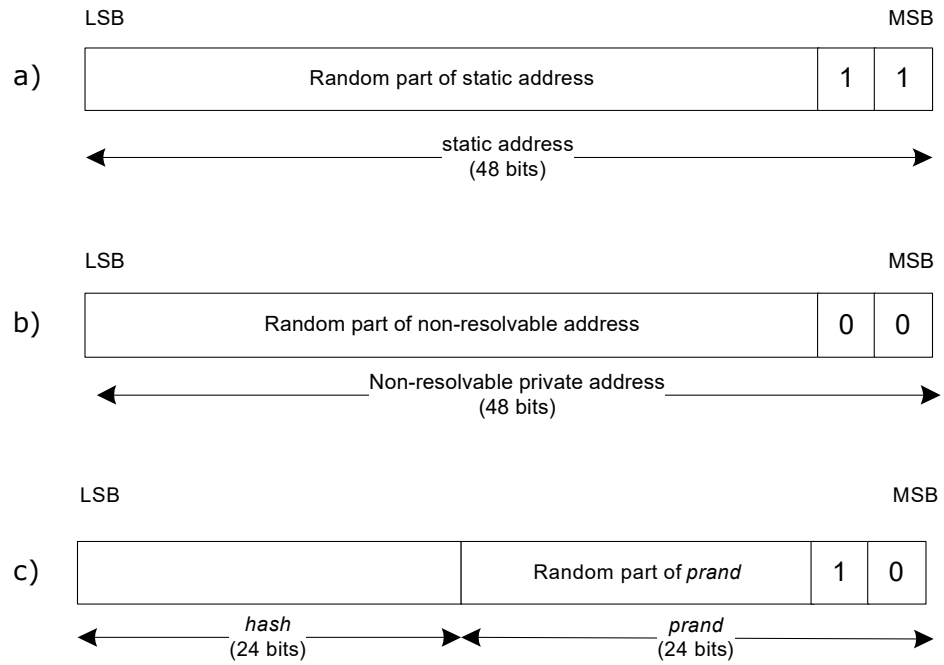


Fig. 1.4: BLE addresses. A) Static b) Non-resolvable c) Resolvable [3].

1.2.4 Security in Bluetooth Low Energy

When transferring data, BLE uses AES-CCM cryptography for data encryption. AES encryption is generally considered very secure [5] and protects BLE communication from passive eavesdropping, where a third device listens to the data being exchanged between the two devices. But the weak point in otherwise secure communication may be the encryption key exchange protocols, which serves for the transmission of encryption keys when establishing a secure connection. In BLE, methods by which the keys are exchanged are called as "pairing method" or "association model", and are critical in the case of *Man-In-The-Middle* (MITM) attacks, where a third device impersonates other two legitimate devices and tries creating a bridge between them [6]. These attack scenarios are shown in Fig. 1.5.

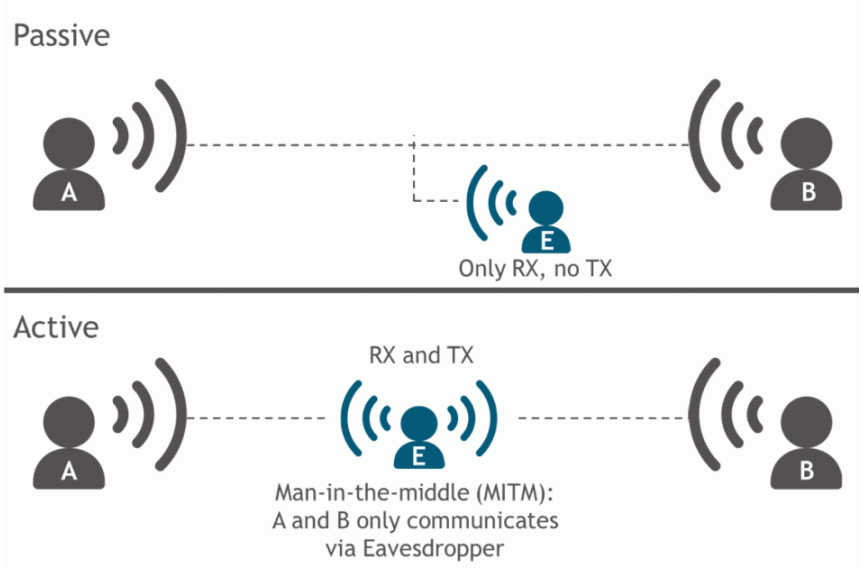


Fig. 1.5: Passive eavesdropping (up) and MITM (down) attack models [7].

Pairing, which serves for the creation of a secure link, can be divided into 3 main phases [6].

First phase. In this phase, the two devices exchange their I/O capabilities and determine a suitable method for a set up of secure connection.

Second phase differs based on the chosen type of pairing. In LE Secure Connections Pairing a LTK (Long Term Key) is generated and exchanged using Elliptic Curve Diffie Hellman cryptography, which is then used for encryption and authentication of communication.

Third phase is optional and serves for the bonding process. In this phase, the two devices exchange up to three transport specific keys: LTK, IRK and CSRK (Connection Signature Resolving Key). CSRK is used for data signing at the ATT layer as a protection against replay attacks [13].

The association models influence mainly the second phase of pairing and include methods as Numeric Comparison, Just Works, Out Of Band, and Passkey Entry. Use of the association models is determined by the I/O capabilities of the two devices. Each of these association models is described in the following section.

Numeric Comparison [3] association model is used for a pair of devices, which are both able to display six digit number and are capable of a user entering "yes" or "no", e.g., mobile phone, PC, etc.

Both devices, which are paired together, display a six digit number and the user is asked if the numbers match. If "yes" is entered by the user the pairing is successful.

Aside from protection against MITM attacks, it serves as a confirmation to the user that the correct devices is being paired, as many BLE devices do not have a unique name.

Just Works [3] association model is used in the scenario where at least one of the devices is not capable of displaying a six digit number and does not have a keyboard for entering a six decimal digits, e.g., headset, pedometer, etc.

Just Works association model uses the Numeric Comparison protocol, but the device never shows a number to the user. The application may just ask the user to accept the connection, but the exact implementation depends on the designer.

Just Works association model has the same protection against passive eavesdropping as the Numeric Comparison, but it has no protection against MITM attacks.

Out Of Band (OOB) [3] association model is used in the scenario, where the devices use an Out of Band channel for both the discovery and the exchange of the cryptographic numbers in the pairing process. In this case, the security depends on the security of the Out of Band channel. It should be resistant to the MITM attacks, otherwise, there is a risk of the security being compromised during authentication.

An example of the OOB method is the use of an NFC (Near Field Communication), where the user first touches the two devices together and is asked if a pairing should commence. When "yes" is entered, all of the discovery and cryptographic information is transferred through the NFC channel (which is resistant to the MITM attacks by default). Then the devices use this information to establish a connection over BLE.

The OOB protocol is used only when the pairing process has been commenced by the OOB exchange and at least one of the devices gives OOB as their I/O capability.

Passkey Entry [3] association model is used in the case, where one device has a display capable of displaying six digit number, and the second device has an input capability for entering a six decimal digits.

When pairing, the first device shows six digit number to the user and asks to enter this number on the second device. If the correct value is entered, the pairing is successful.

An important point is that the six digit number is used only for comparison as in the Numeric Comparison model, and is independent of the actual security algorithm. So knowing this number has no benefit in decrypting of the encoded data, transferred in the BLE communication.

1.3 Local Interconnect Network

Local Interconnect Network (LIN) is a serial network protocol widely used in automotive. Its advantages are low cost and simple implementation as it is based on UART hardware. It supports baud rate up to 20 kbit/s, which is lower than its more expensive alternatives (CAN - Controller Area Network, FlexRay, etc.). LIN cluster consists of one master task and several slave tasks (Fig. 1.6). A master node contains the master task, which initializes communication by sending a header of LIN frame, and a slave task responsible for sending of a response, containing data requested by the header. All other slave nodes contain a slave task only [8].

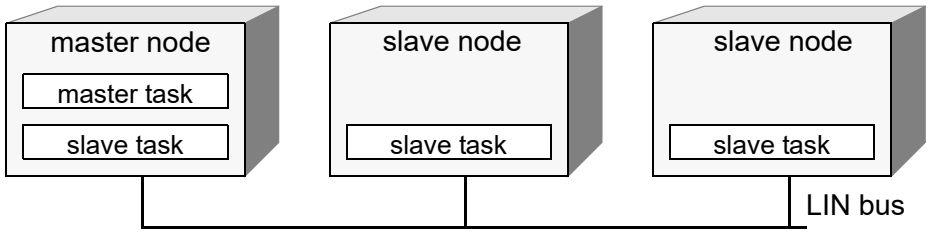


Fig. 1.6: Block diagram of LIN cluster [8].

1.3.1 Frames

To assure each frame is given enough time for a transmission, the master task schedules every LIN frame to the beginning of a frame slot, defined as a multiple of a base time. Base time is the minimal time unit used for a frame transmission, usually set as 5 or 10 ms [8].

The structure of a LIN frame is shown in Fig. 1.7. It consists of 2 main parts: header and response. A header is managed by the master task and is made of a Break field, Synchronization byte and Protected Identifier (PID). A response is managed by the slave task and is made of one to eight data bytes and a Checksum. As LIN protocol

is based on UART hardware, every part of the frame, except for a Break field, is in UART format with 1 start bit, 1 stop bit and no parity [8].

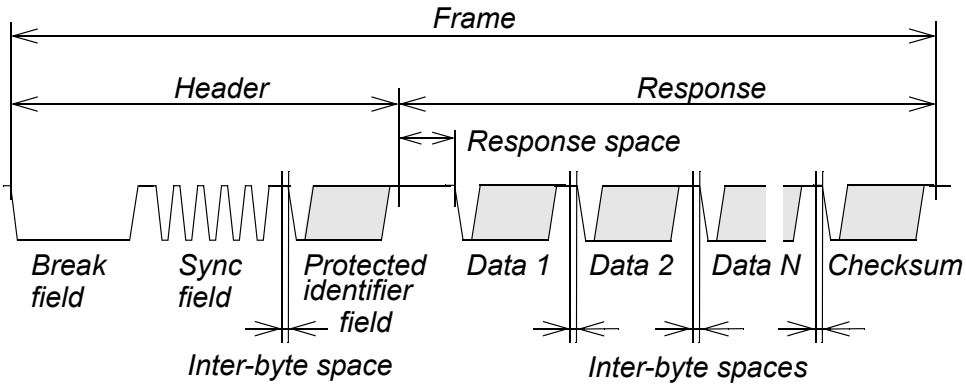


Fig. 1.7: Structure of LIN frame [8].

Break field is used as a symbol signaling the beginning of a new frame. It consists of at least 13 bits in dominant state, followed by a break delimiter made of at least one bit in recessive state. For detection of a break field, slave task shall use a detection threshold of 11 dominant bits [8].

Synchonization byte field is a byte field with the value of 0x55, resulting in an alternating row of dominant and recessive states. When a break/sync field sequence is detected, a slave task shall abort any ongoing transfer and begin new frame processing [8].

Protected Identifier - PID is used for a recognition of LIN frames. As shown in Fig. 1.8, PID byte field is divided into 2 parts: the frame identifier (lower 6 bits) and the parity (2 MSB). The parity is computed by following equations (1.3) and (1.4) [8].

$$P0 = ID0 \oplus ID1 \oplus ID2 \oplus ID4 \tag{1.3}$$

$$P1 = \neg(ID1 \oplus ID3 \oplus ID4 \oplus ID5) \tag{1.4}$$

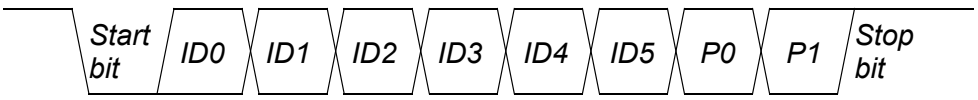


Fig. 1.8: PID field structure [8].

Each frame defined by a unique PID, can have multiple slave tasks assigned as subscribers (nodes receiving a response), but only one slave task assigned as a publisher (node sending a response).

Data bytes are arranged as a little-endian LSB (Least Significant Byte) first. A frame can carry between one to eight data bytes, but the length must be agreed on across all subscribers and a publisher for each frame with a unique PID.

Checksum serves for verification of a frame received on LIN bus. It is computed as an inverted value of a 8-bit sum with carry across all data bytes in classic checksum, or across all data bytes and PID byte in enhanced checksum. 8-bit sum with carry can also be computed as a sum of all values and consequent subtraction of 255 every time the result is greater than 255 [8].

2 Keyless Entry Demonstration kit design

In this chapter, the design of a demonstration kit and implementation of Keyless Entry system will be discussed.

As mentioned in the introduction, the demonstration kit shall consist of 3 modules. The BCM, acting as a LIN master and controlling the demonstration kit as a whole. The door lock module, responsible for driving of a door lock motor. And the main focus of this thesis, the Keyless Entry module, which communicates with a remote key device over Bluetooth® Low Energy, and passes requests for door lock control to the BCM, based on the behavior of the remote key device.

2.1 Body Control Module

This section deals with an overall design of the BCM for the purpose of Keyless Entry demonstration kit.

2.1.1 Circuit design

For the hardware realization of this module, an existing PCB (Printed Circuit Board) LIN_GW_V1, made by my consultant Ing. Filip Brtáň, has been used (inherited schematic and layout can be found in appendix B). This PCB is supplied from 12 V battery supply, which is down converted to 3.3 V voltage supply with NCV4274 LDO. The PCB includes NCV7329 LIN transceiver in master configuration, supplied from 12 V battery supply, which shall be used for realization of BCM as a LIN master. The PCB also includes second LIN transceiver in a slave configuration, which is not used for the purpose of this demonstration kit. Master configuration means connecting a diode and 1 kΩ resistor between LIN bus and a battery supply, and connecting 1 nF capacitor to the connector on LIN bus. Slave configuration means using 220 pF capacitor on LIN bus and no connection between LIN bus and a battery supply [8].

As a core processing unit, the PCB utilizes AT32UC3C processor in 64-pin QFN package, supplied from 3.3 V. The processor uses ADC (Analog-to-Digital Converter) with internal 1 V reference, for battery supply measurement. Used circuit for battery supply conversion to a level measurable by ADC is shown in

Fig. 2.1. It features passive voltage divider converting 12 V battery supply to 0.798 V (2.1). The voltage divider is connected to the battery supply with transistor Q1B, which is opened for current to flow by setting a logical 1 to the base of transistor Q1A with VBB_SENSE_EN pin. This connection is used to reduce overall current consumption caused by the voltage divider and as such should be enabled only when the supply voltage is being measured. The circuit also features a diode connected to a processor voltage supply to limit voltage level on VBB_SENSE in case of high voltage peaks, which can frequently happen on battery supply in automotive applications.

$$V_{\mathrm{BB\_SENSE}} = V_{\mathrm{BB}} \cdot \frac{R_{50}/2}{R_{50}/2 + R_{51}} = 12 \cdot \frac{4.7 \cdot 10^3/2}{4.7 \cdot 10^3/2 + 33 \cdot 10^3} = 0.798\,\mathrm{V} \qquad (2.1)$$

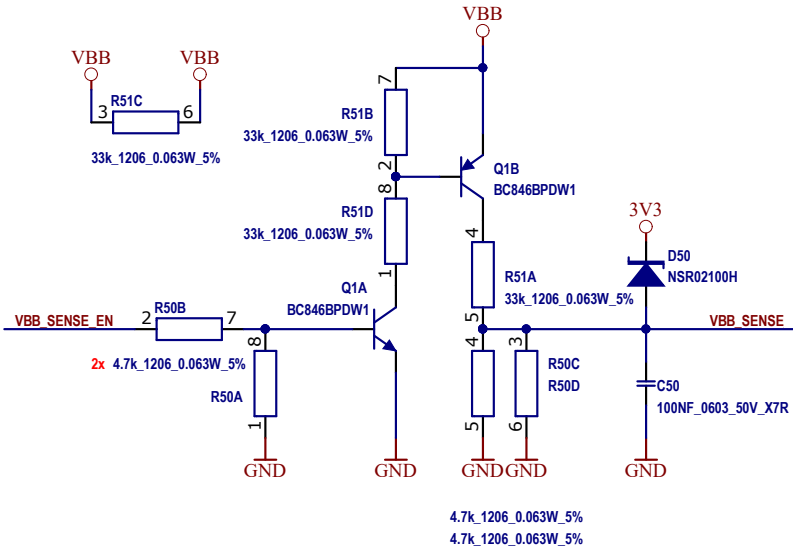


Fig. 2.1: Design of supply measurement with ADC on BCM PCB.

Lastly, the PCB includes a USB interface used for communication with a PC, enabling the application to be controllable with the use of GUI on PC.

2.1.2 Hardware design

The layout is designed as a 2 layered PCB and shall be situated in an ABS box HH-3466. The connectors used for LIN connection are RJ-11, which enables to distribute LIN bus together with a battery supply and a ground signal in one RJ-11 cable. Furthermore, the PCB includes a DC connector for supplying from 12 V

AC/DC adapter, and a mini USB connector. The PCB also features three green and red LED doubles contained in light pipes for indication of function. LED1 indicates level of battery supply, LED2 indicates activity and errors on LIN slave node, and LED3 indicates activity and errors on LIN master node.

Schematic and layout of the PCB can be found in appendix B.

2.1.3 Firmware design

The firmware realizes function of a LIN master, translating requests from the Keyless Entry module to the door lock module commands. The LIN frames are slotted by a timer configured for 10 ms period. Each period, the BCM decides what LIN frame should be send on the bus. In default, the BCM reads out status frames from both connected modules alternately. In case of a request to change the state of a door lock is registered, a command frame is sent to the door lock module to match the request. Similarly, any change in the state of a door lock is reported back to the Keyless Entry module to inform the user of the current state of the door lock. Detailed description of LIN communication protocol and frames used in communication is described in chapter 2.5.2.

As the LIN frames are slotted in 10 ms periods, voltage supply level is checked for under- or over-voltage and the state of PCB is indicated by the LEDs in the same period.

2.2 Door lock module

This section deals with the design of a door lock module for driving of a vehicle door lock motor. For the sake of future reusability in other projects, this PCB has been designed as an EVB (Evaluation Board) for used NCV7710, featuring additional capabilities not used in this demonstration kit.

2.2.1 Circuit design

While the door lock module serves for driving of a door lock motor, NCV7710 has been used as it is designed specifically for this purpose. This Integrated Circuit (IC) features two PWM (Pulse Width Modulation) controllable half-bridges, output for

current monitoring, and is configurable over SPI (Serial Peripheral Interface) bus. The supply of NCV7710 is realized by 12 V battery supply with a reverse protection circuit.

Similarly to the BCM, AT32UC3C is used as a processing unit controlling the NCV7710 over the SPI, and providing the processing of LIN communication. The processor is supplied from 5 V voltage supply, down converted from 12 V battery supply with NCV7428-5 LIN SBC. This SBC also serves as a LIN transceiver, connected in a slave configuration. Alternatively, the EVB contains a footprints for NCV7451 CAN SBC, also supplying 5 V voltage supply down converted from 12 V battery supply and realizing CAN transceiver. But this SBC is not used in this thesis, as there is no CAN communication featured in demonstration kit.

Output for current monitoring on NCV7710 is realized by mirroring of a current from OUT1 or OUT2 divided by 13400 [9]. Current is converted to a voltage with an Ris resistor and measured by channel 1 of ADC. Value of Ris is 6.8 kΩ for the reason, that absolute maximum current on OUT1 and OUT2 is 10 A. In this case voltage converted by the resistor would theoretically be 5.075 V (2.2), which corresponds with the processor voltage supply.

$$V_{\mathrm{ISout}} = \frac{I_{\mathrm{out}} \cdot R_{\mathrm{is}}}{13400} = \frac{10 \cdot 6.8 \cdot 10^3}{13400} = 5.075\,V \tag{2.2}$$

Aside from an output current monitoring, the PCB also features battery supply monitoring, similar to the BCM PCB (Fig. 2.2). In this case, ADC utilizes $0.6\,V_{\mathrm{CC}}$ voltage reference, enabling for conversions of higher voltage levels. V_{CC} is 5 V voltage supply of the processor connected to V_{DDANA} pin. The 12 V battery supply is than divided to 1.579 V (2.3).

$$V_{\mathrm{BAT\_SENSE}} = V_{\mathrm{BAT}} \cdot \frac{R_{\mathrm{m7}}/2}{R_{\mathrm{m7}}/2 + R_{\mathrm{m6}}} = 12 \cdot \frac{10 \cdot 10^3/2}{10 \cdot 10^3/2 + 33 \cdot 10^3} = 1.579\,V \tag{2.3}$$

Lastly, the PCB also includes a USB interface for communication with PC, and control elements for possible standalone function such as switches and two buttons dedicated for locking and unlocking of a car door.

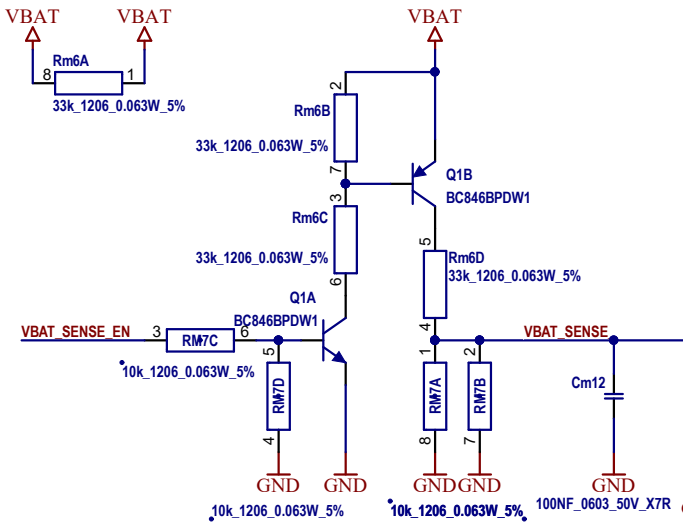


Fig. 2.2: Design of supply measurement with ADC on NCV7710 EVB.

2.2.2 Hardware design

The layout is designed as a 2 layered PCB and shall be situated in an ABS box 1591MSBK. The PCB includes two mounting holes for fastening in the box.

LIN and CAN buses share one connector, which calls for a connector with at least 5 pins. A simple 6 way connector with a lock has been chosen for this PCB. Furthermore, the PCB includes a 2 pin connector with a lock for the motor connection, a DC connector for supplying from 12 V AC/DC adapter, and a micro USB connector.

Schematic and layout of the PCB can be found in appendix C.

2.2.3 Firmware design

During initialization of a door lock module, the processor configures NCV7710 into active mode, sets both outputs to be controlled by PWM1 input, and configures them to a high impedance state. When a command to lock or unlock a door lock is registered from incoming LIN frame, both outputs are activated accordingly for required lock state (high side x low side) to begin transition, and a timer is set to measure 500 ms. After the 500 ms passes, both outputs are set back to a high impedance state and a lock motor becomes inactive again. Output, that is configured for a high side is also set for a passive freewheeling, which means

controlling the output with a PWM signal, toggling it between high side and high impedance state. The output configured for a low side has PWM control disabled. Possible configurations for output can be seen in Tab. 2.1, used configurations are marked in red square.

CONTROL_2	PWM input pin	CONTROL_0		Output pin state
OUTx_PWM1/2	PWM1/2	OUTx_HS	OUTx_LS	OUTx
0 (PWM disabled)	X	0	0	High Impedance
		0	1	L
		1	0	H
		1	1	High Impedance
1 (PWM enabled)	0	0	0	High Impedance
		0	1	High Impedance
		1	0	High Impedance
		1	1	L
	1	0	0	High Impedance
		0	1	L
		1	0	H
		1	1	H

Tab. 2.1: Possible output configurations, using NCV7710 SPI registers [9].

In case of an error occurring on NCV7710, outputs are automatically disabled. The cause of an error might be: supply undervoltage or overvoltage, output current overload or underload, or thermal shutdown. Consequently, the processor reads out the status of NCV7710 and reports the Global Fault Bit [9] to the BCM in the outcoming LIN frame.

For indication purpose, the application uses 3 LEDs. Green LED Dm1 is used to indicate the application being powered up and is set during initialization. Second green LED Dm2 indicates outputs being active and is set with the beginning of the transition, and reset with the end of the transition. And red LED Dm3 indicates any error on NCV7710, and is controlled by the state of the Global Fault Bit.

2.3 Keyless Entry module

This section deals with the design of a Keyless Entry module realizing the core function of Keyless Entry system. This module communicates with the remote key over BLE and requests control over a door lock by responding to LIN frames from the BCM.

2.3.1 Circuit design

The Keyless Entry module features the RSL10 *System-in-Package* (SIP), which includes the RSL10 together with an on-board antenna and all necessary passive components in one package, which helps minimize size of the final PCB. The RSL10 is supplied with 3.3 V, down converted from 12 V battery supply with NCV7428-3 LIN SBC. The SBC also serves as a LIN transceiver connected in a slave configuration.

The PCB includes buttons for execution of a transition into pairing mode, and for the erasement of a flash memory containing bond information. These buttons are meant only for demonstration purpose during standalone showcasing. In application both features can be executed by LIN command from the BCM.

2.3.2 Hardware design

The layout is designed as a 4 layered PCB and shall be situated in an ABS box HH-3466. 4 layered option has been chosen as the RSL10 SIP is only available in BGA-type package. Internal planes are arranged in a way, that the plane closer to the top layer is connected to the ground and the plane closer to the bottom layer is connected to the 3.3 V voltage supply.

To comply with the BCM, Keyless Entry module PCB features RJ-11 connector for connection of a LIN bus, together with a battery supply and ground signal. The PCB also utilizes two headers J3 and J4 for measurement of current consumption of RSL10 and the module as whole.

Similarly to the BCM, LED indicating state of connection to the remote key is contained in a light pipe, to bring it out of the ABS box. Schematic and layout of the PCB can be found in appendix A.

2.3.3 Firmware design

The RSL10 SIP is available with an Eclipse-based ON Semiconductor IDE (Integrated Development Environment) with the support of libraries for hardware configuration, and RSL10 Kernel implementing event and message handling system for configuration of *Bluetooth® Low Energy* (BLE) stack according to the

specification. This IDE is used for a necessary code development of Keyless Entry module firmware.

The firmware can be dived into two parts. First part is dedicated to the configuration of a Bluetooth transceiver with the use of Kernel, and realizes the application layer of Bluetooth stack. Configuration and operational modes are closely describer in the following section 2.4.

Second part realizes a slave task of LIN communication controller software stack, with the use of a DMA and UART peripheral. The implementation of the LIN controller is further discussed in LIN configuration section 2.5.1.

2.4 Bluetooth transceiver

While the Keyless Entry system controls a car door lock, it is desirable for it to be able to connect only to the key devices, which are authorized for it. For this very reason, Keyless Entry application operates in two modes. Normal mode is used for standard function, which enables only known (bonded) devices to connect with it and is invisible for any other devices, not known for the application. Second, Pairing mode is used for pairing with new devices. In this mode, any device is able to connect with the system and exchange authentication and encryption keys necessary for a bond establishment. By pairing, a relationship of known device is established with the system. This mode can be accessed by command over LIN bus. To ensure security, Pairing mode should be accessed only upon first connection in controlled environment.

2.4.1 Normal mode

In this mode the device communicates with the use of resolvable addresses and is configured as connectable with a filter for connection and scan requests limited only to the devices listed on a white list. White List, together with Resolving List needs to be set during the initialization of the device with the bond information from the bond list located in a flash memory. Resolving List is set with an identity address, type of address (whether it is Public or Private Static address) and IRK. It is then used by Link Layer to resolve any resolvable address (Controller Managed Privacy) from incoming communication and compare the assigned identity address with addresses set in the White List. If the identity address of a peer device is also

set in the white list, the resolvable address is replaced by identity address and the message is sent to a higher layer. If the identity address is not set in the white list, communication is discarded.

2.4.2 Pairing mode

This mode is designed for the exchange of a bond information with a peer device, that is not paired yet. The configuration of the device is changed to the Host Managed Privacy, and filter policy for connection and scan requests is set to any device. As a result any peer device can establish a connection and pair with the Keyless Entry module. The Host Managed Privacy is used due to the case, in which a pairing device has already been paired and it is necessary to delete any previous record of the device from the bond list by GAP. In this process information needed for resolving is erased during communication with the peer device, so it is essential for GAP to have full control over address handling.

Upon entering a pairing mode, a timeout timer is set for one minute. If there is no bond request registered during this time, the device goes back to the normal mode. When a bond request is registered and a bonding is successful, the connection established for bonding purpose is canceled and the mode is switched back to the normal mode automatically.

2.4.3 Service configuration

The Keyless Entry application utilizes a Custom service and a Proximity profile, composed of *Immediate Alert Service* (IAS), *Link Loss Service* (LLS) and *Tx Power Service* (TxPS) for realization of its function.

Custom service is a service used for general control over the Keyless Entry module. The Custom service consists of 2 characteristics. First characteristic is used for transmission of data from a key device to the Keyless Entry module. This characteristic is configured for both read and write, and is defined by 2 byte value. First byte controls the state of lock: 0 - unlock, 1 - lock, and the second byte enables automatic locking/unlocking based on the proximity: 0 - disable, 1 - enable. Second characteristic is used for transmission of data from the Keyless Entry module to a key device. This characteristic is configured for read-only and periodic notification, when enabled. Its value also contains 2 data bytes. First byte

reports current state of a door lock stated by the BCM: 0 - unlock, 1 - lock. The second byte reports any errors occurring on a door lock module, which might lead to inaccurate lock state determination: 0 - no error, 1 - error occurred.

Proximity profile [10] is used in order to determine the proximity between the Keyless Entry module and a key device. As defined by the BLE Specification, the proximity profile consists of three services. The Tx Power Service is used for sharing of the current transmit power level of the Keyless Entry module and is configured as read-only. The Immediate Alert Service and the Link Loss Service are used to trigger an alert of certain level within the Keyless Entry module. The alert levels are: No Alert = 0, Mild Alert = 1 and High Alert = 2. The difference between these two is that the IAS triggers the alert immediately, and as such is configured for write-only. The LLS stores the level of alert and triggers it only when a physical link between devices is disconnected. The LLS is configured for both read and write.

In the RSL10, all Bluetooth services and profiles have to be configured with the use of a Kernel message handling system. The communication between Kernel and Application during initialization of the Keyless Entry module is illustrated in Fig. 2.3 [11] [12].

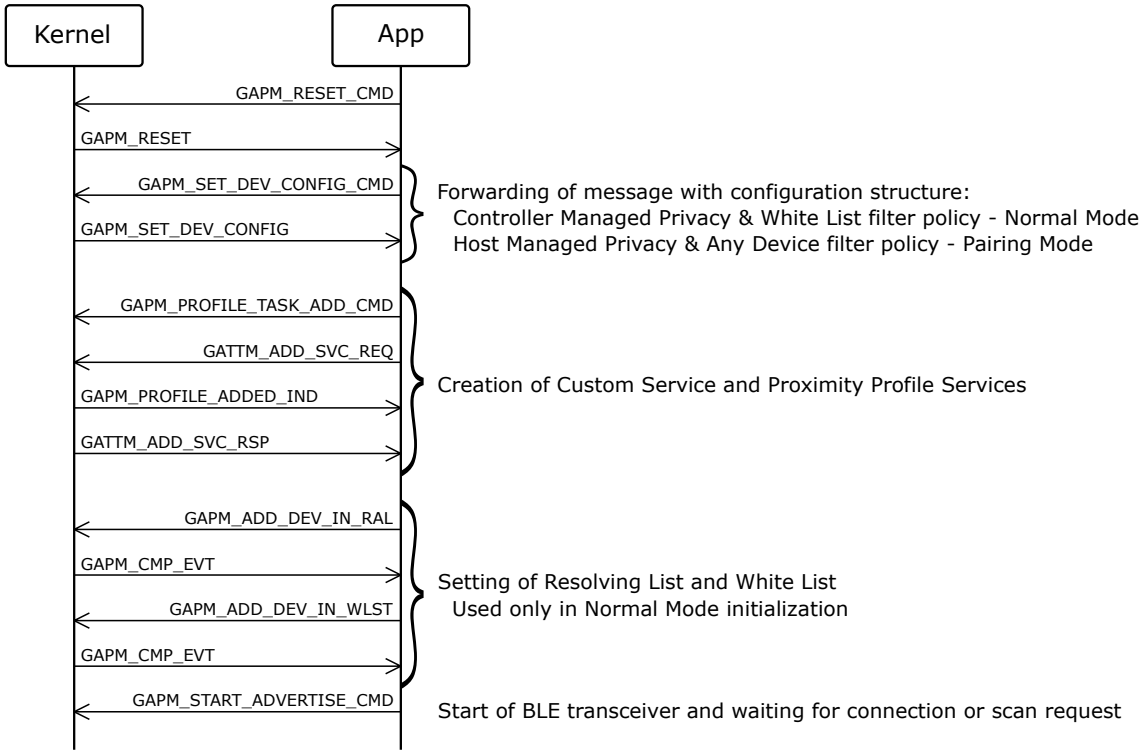


Fig. 2.3: Communication sequence between the Keyless Entry application and Kernel.

2.4.4 Proximity

Alerts defined by IAS are used to share information about the proximity of a key device. The key device computes the level of proximity by the difference of transmitted power and *Received Signal Strength Indicator* (RSSI). The level of proximity is classified into 3 zones: Immediate, Near and Far. Current zone in which a key device is located, is notified to the Keyless Entry module by sending assigned alerts: Immediate - No Alert, Near - Mild Alert, Far - High Alert. On the other hand, the LLS is used purely as an indication of losing connection. In this case, it is configured to automatically lock a car door, when a High Alert is risen by LLS. Thus a High Alert shall be set on LLS by a key device immediately after the establishment of a connection, to ensure automatic locking of a car door, if a loss of connection occurs.

Ability to determine proximity enables the Keyless Entry system to lock and unlock a car door automatically. In this thesis, automatic locking is implemented in a way, that when a remote key transits into a Immediate zone it sends command to unlock a car door, and when it transits into a Far zone it sends command to lock a car door.

However, this features has a known issue in case of a Two-thief attack [13]. During this attack the two thieves create a bridge between a vehicle and a remote key by receiving communication signal from the vehicle, amplifying it and re-transmitting it near the remote key. After that, a response from the key is transmitted back to the vehicle and a car door is automatically unlocked. Because this attack doesn't compromise the communication itself, only adjusts the strength of the signal it will always be successful, as long as there is a communication between the vehicle and the remote key. However, in this thesis a remote key is realized with a smart phone, which can be easily disconnected from a vehicle, either manually of automatically upon closing the Key Device application. On top of that, the feature to manually disable automatic locking and unlocking is implemented, for the case where the user would leave a vehicle unattended for a longer period of time and wanted to leave the Key Device application running for any reason. In the future, this issue shall be fully resolved by using next generation devices utilizing a Time of Flight feature, as it will replace the need to use a received power for determination of proximity.

2.5 LIN Configuration

The communication between each module of the Keyless Entry demonstration kit is realized through LIN bus. The BCM and the door lock driver PCBs both use processor AT32UC3C from Microchip Technology as a controller. This enables the use of the original LIN libraries from the manufacturer for designing LIN driver, which uses a UART peripheral with LIN support and a DMA (Direct Memory Access) controller. On the other hand, the RSL10 used in the Keyless Entry module requires creation of a custom LIN communication controller software stack library. It shall be made in the similar manner as with the AT32UC3C, employing a UART peripheral and a DMA controller. Because the UART doesn't include a LIN support, it may present a few challenges as a break field detection in header recognition. As for general parameters, LIN communication shall be using a baudrate of 20 kbit/s with a 10 ms time slots for frame transmission.

2.5.1 Custom LIN Driver

The Keyless Entry module acts as a slave device in the demonstration kit. For this reason, it shall consist only of a slave task. This task is responsible for a break field detection, as well as, a LIN header recognition and consequential response handling based on the received header.

The break field detection is realized by an external interrupt on a LIN RX pin, which monitors the LIN bus for any transition between dominant and recessive state. Upon triggering of this interrupt, state of a bus is checked. In case of a dominant state, a timer for determination of a break field duration is started. When a recessive state is registered after interrupt, the timer is stopped and the duration of the dominant state is checked, if it meets requirements for the break field. This detection is always active, even during LIN frame handling, as recommended by the specification.

When a correct break field is registered, DMA channel 0 is configured for the header reception and started. After a correct header is received, the PID is compared with a database of registered frames. In case, that the PID is registered for a response publishment, a checksum is computed for scheduled data and DMA channel 1 is configured for transmission of data bytes and checksum. If the PID is registered for subscription, DMA channel 0 is reconfigured for a response reception and a response is received.

After response handling, the checksum is always checked and a corresponding function registered for data processing is called.

2.5.2 Communication Protocol

For communication within the demonstration kit, a four LIN frames are used in total. Two frames are used for data transmission between the BCM and the Keyless Entry module, the other two for data transmission between the BCM and the door lock driver module. Each frame is defined as following:

Frame with the ID = 0x20 (Fig. 2.4) serves for transmission of a response to the Keyless Entry module and consists of a single data byte.

ID = 0x20:

7b	6b	5b	4b	3b	2b	1b	0b
X	X	Lock state	Error flag	Flash reset	Pairing config		Disconnect

Fig. 2.4: Definition of a data byte used in transmission of a response to the Keyless Entry module.

- **Lock state:**
  - 0: Indicates that the door is currently unlocked.
  - 1: Indicates that the door is currently locked.
- **Error flag:**
  - 0: No error indication.
  - 1: Indicates an error occurred during locking.
- **Flash reset:**
  - 0: No Effect.
  - 1: Clears bond list stored in the flash memory.
- **Pairing config:**
  - 00: No Effect.
  - 01: Triggers a transition to the Pairing mode.
  - 10: Triggers a transition back to the Normal mode.
  - 11: No Effect.
- **Disconnect:**
  - 0: No Effect.
  - 1: Disconnects a key device, in case there is one connected over BLE.

Frame with the ID = 0x10 (Fig. 2.5) serves for transmission of a response from the Keyless Entry module and consists of a single data byte.

ID = 0x10:

7b	6b	5b	4b	3b	2b	1b	0b
X	X	X	X	Pairing	Connected	Execute	Lock state

Fig. 2.5: Definition of a data byte used in transmission of a response from the Keyless Entry module.

- **Pairing:**
  - 0: The device is in the Normal mode.
  - 1: The device is in the Pairing mode.
- **Connected:**
  - 0: A key device is connected to the device.
  - 1: No connection to the device.
- **Execute:**
  - 0: No Effect.
  - 1: Requests a change in the lock state based on the Lock state bit.
- **Lock state:**
  - 0: Requests a door to unlock, when Execute = 1.
  - 1: Requests a door to lock, when Execute = 1.

Frame with the ID = 0x09 (Fig. 2.6) serves for transmission of a response to the door lock module. Response consists of a single data byte, used for control of a door lock motor.

ID = 0x09:

7b	6b	5b	4b	3b	2b	1b	0b
X	X	X	X	X	X	Lock state	

Fig. 2.6: Definition of a data byte used in transmission of a response to the door lock module.

- **Lock state:**
  - 00: No Effect.
  - 01: Triggers a transition of a door lock motor to an unlock state.
  - 10: Triggers a transition of a door lock motor to a lock state.
  - 11: No Effect.

Frame with the ID = 0x0B (Fig. 2.7) serves for transmission of a response from the door lock module. Response consists of a single data byte.

ID = 0x0B:

7b	6b	5b	4b	3b	2b	1b	0b
X	X	X	X	X	X	LIN error	Lock error

Fig. 2.7: Definition of a data byte used in transmission of a response from the door lock module.

- **LIN error:**
  0: No error.
  1: Indicates that an error have occurred during previous LIN communication.

- **Lock error:**
  0: No error.
  1: Indicates that a global error bit on a door lock module has been set.

2.6 RSL10 LIN Demo Application

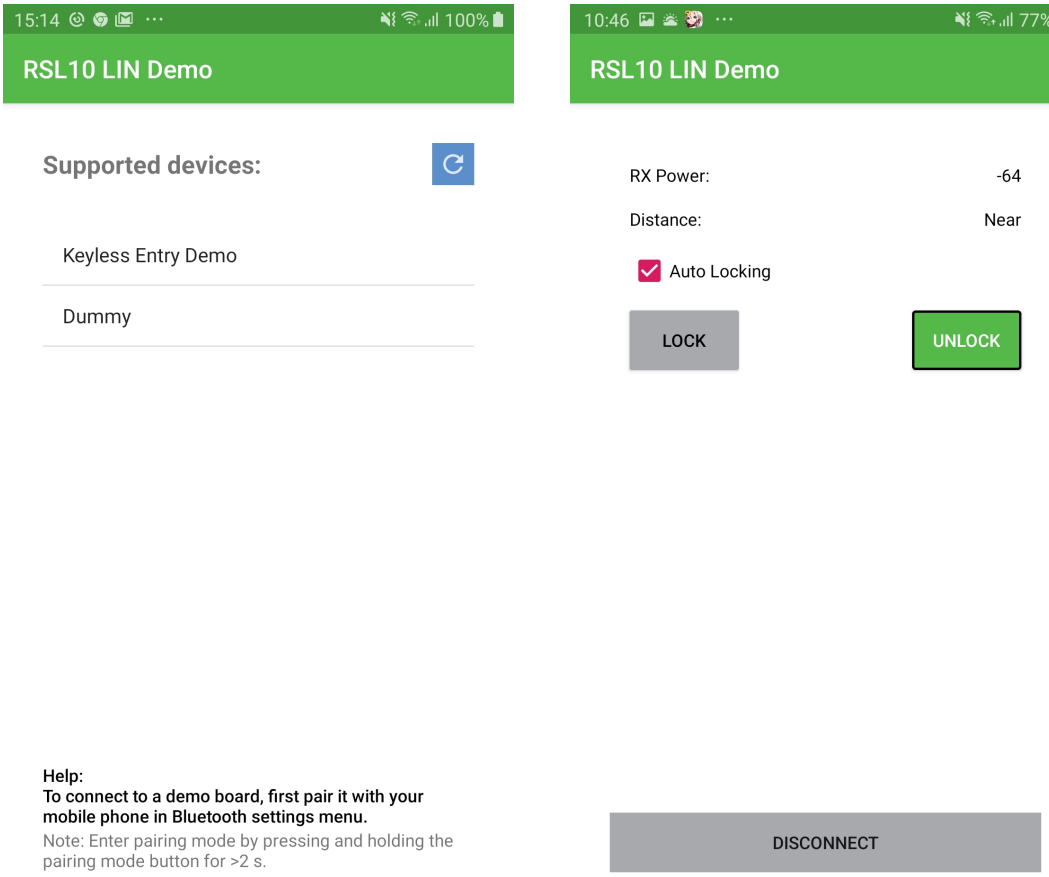
The key device is a device, which advertises its presence to the keyless entry system, and signals if a car door should be locked/unlocked. In this thesis, the key device is realized by a smartphone with an application designed for operational system Android 6 (API - Application Programming Interface, level 23) or higher. It would be possible to use API of lower level as the Bluetooth low energy was included first in android 4.3 (API level 18), but using API of higher level allows usage of more stable and reliable methods for BLE. Especially a class BluetoothLeScanner first included in API level 21, providing methods for scan related operations for BLE devices. The only disadvantage of using higher level API is limiting certain amount of users with lower version of android, but according to the official android developer sites, this affects only around 20% of android devices [14].

2.6.1 Scan Activity

The graphic layout of this activity can be seen in Fig. 2.8a. The main purpose of the Scan activity is to discover any Keyless Entry devices and connect to them when selected by the user. The discovery procedure is realized by a BluetoothLeScanner

object from BluetoothAdapter [15]. Because the scan procedure has high demands on energy usage, it should always be time-restricted and executed only when needed. In this case, the scan is limited to 10 sec duration and starts automatically only on the startup of the application. But it can be re-executed anytime by the blue refresh button when needed. To restrict scan only for supported devices, controllable by the application, the scanner is set with a filter to discover only devices with a name set as "Keyless Entry Demo".

When the Keyless Entry device is discovered by the scanner it is sent to a callback function, where it gets inserted to the list of supported devices. From this list, it can be selected by the user, in which case the Control activity is started and address of the device is transferred to it as an Intent [16].



(a) Scan activity.

(b) Control activity.

Fig. 2.8: Screenshots of the key application activities.

2.6.2 Control Activity

The graphic layout of the Control activity can be seen in Fig. 2.8b. This activity manages the connection between the application and the Keyless Entry device and handles the transmission of data to the Bluetooth services used in Keyless Entry Demo.

First, it connects to the selected Keyless Entry device and creates GATT instance for the sharing of Bluetooth characteristics. The application acts as a GATT client, so it initializes any read/write commands with the Keyless Entry module. Immediately after the establishment of connection, service discovery is executed and Link Loss Service is written to a High Alert (viz. 2.4.3).

After initialization, the Control activity periodically measures RSSI and determines a proximity to the device. Both of these information are displayed in the top part of the user interface. When the estimated distance changes to immediate or far, it signals this transition to the device, by writing an IAS accordingly to the change.

It is also possible for the user to control a door lock motor directly from the application by pressing the assigned buttons. This action is sent to the device with the use of a Custom Service. It triggers an immediate change of the motor state regardless of proximity, but doesn't interfere with an automatic locking of the system, as the motor state is updated with the next proximity change. The last function of this activity is the ability to disable/enable automatic updates of a motor state based on the proximity with a checkbox. This is also configured with the use of a Custom Service, and signals to the device how it should behave when it receives proximity state change through IAS.

Upon disconnection of the device both GATT instance and the control activity are destroyed and the application resumes the scan activity.

2.7 BCM Interface for PC

Graphical User Interface (GUI) for BCM is designed to enable control over the behavior of the Keyless Entry demonstration kit without the need to physically interact with the hardware, thus simulating real life application, where every command is initialized over a communication bus. The GUI is developed using Qt creator IDE with Qt widget toolkit.

The GUI communicates with the BCM over USB, using simple text based protocol. Each message starts with a key word "set" for data transmission to the BCM, or "get" for data request from the BCM. The key word is followed by one or more commands specifying type of data to be transmitted, divided by a space character. In case of data transmission, the command is followed with an equal sign and data represented by a decadic number in text format. The commands available for BCM GUI are following:

BLEcmd - "get BLEcmd" reads last command sent to the Keyless Entry module (LIN frame with ID = 0x20). "set BLEcmd=X" writes value X to the LIN frame buffer and sends it to the LIN bus.

BLEstat - "get BLEstat" reads last LIN frame data received from the Keyless Entry module (LIN frame with ID = 0x10).

BCMlock - "set BCMlock=X" triggers transmission of a command to lock (X = 1) or unlock (X = 0) a car door lock directly. This means sending a LIN frame with ID = 0x09 to a door lock module as defined in 2.5.2. No "get" command is implemented, as the current state of a car door lock, and error bit are already included in data from "BLEstat" command.

BCMstatread - "set BCMstatread=X" is used to control automatic status read-outs in BCM standalone function. The value X tells how many status read-outs shall be performed after reception of this command. When X = 255, the BCM is set for continuous read-outs any time the LIN slot is free.

2.7.1 Description of function

The main window widget of the BCM GUI automatically searches for any BCM devices connected to a COM port upon startup, or the search can be triggered by pressing refresh button in the bottom left part of the widget (Fig. 2.9). When a valid device is connected to the USB COM port, it can be selected from the list located also in the bottom left part of the widget and connected by pressing adjacent connect/disconnect button. When a BCM device is connected, BCM widget is automatically opened. The widget consists of four control parts.

"BLE command" group box contains check boxes and a list representing bits in a LIN frame with ID = 0x20 (ref. 2.5.2). Upon pressing a "Send" button, the value of LIN frame is transferred over the USB, using "set BLEcmd=X" command and sent to the LIN bus. When "Read Last Command" button is pressed, the value of the LIN frame is requested over the USB, with "get BLEcmd" and filled in the check

boxes and a list.

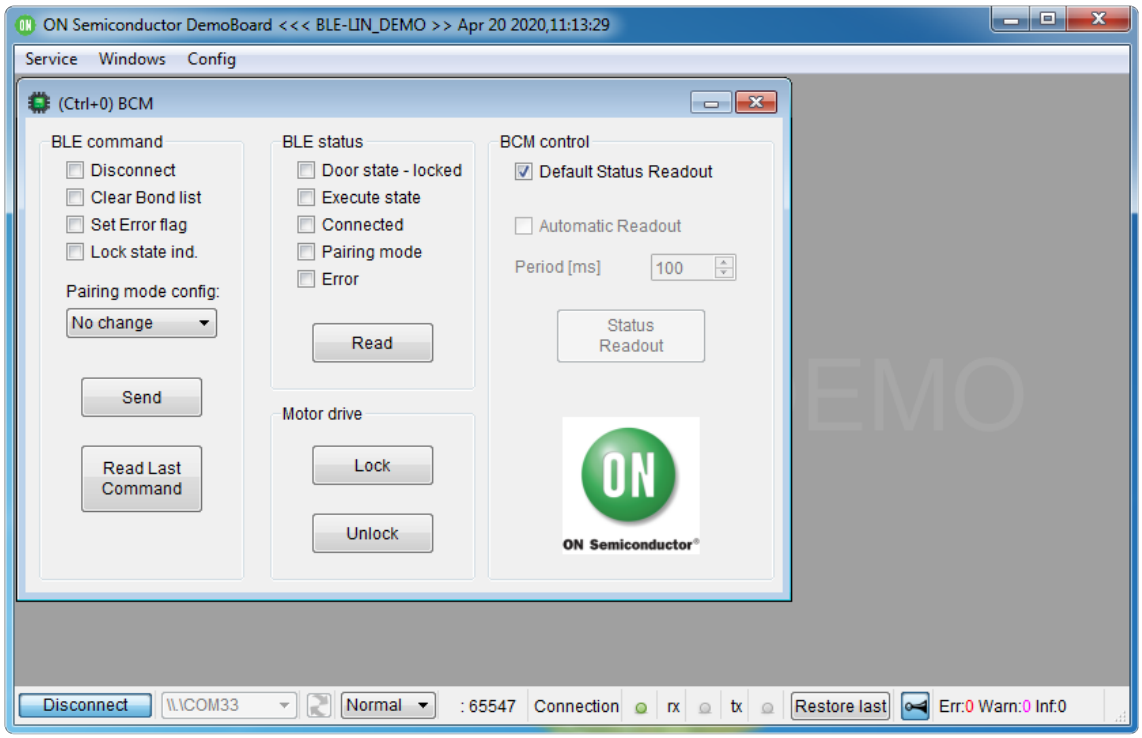


Fig. 2.9: BCM GUI with a BCM device connected.

"BLE status" group box, similarly to the "BLE command" consists of a check boxes representing bits in a LIN frame with ID = 0x10. The difference is that the LIN frame data value is set as read-only and cannot be changed by the user. The value is read from the BCM device by pressing "Read" button, which triggers "get BLEstat" request.

"Motor drive" group box serves for a control over a door lock module directly from the BCM. Both buttons in the group box trigger a "set BCMlock=X" command with a corresponding value.

"BCM control" group box controls the frequency of status read-outs from slave modules, inside of the BCM device. In default mode, a "set BCMstatread=255" is sent and statuses are read whenever possible. In automatic mode, a timer is started and with each period, set by a spin box, a "set BCMstatread=1" is sent to the BCM to read one status LIN frame from each slave module. Alternatively both modes can be disabled, in which case only way to read status is by pressing a "Status Readout" button, which also sends a "set BCMstatread=1" command. This group box affects only read-outs inside of the BCM device for better control over the system. Reading of an actual value of the status frame needs to be preformed by

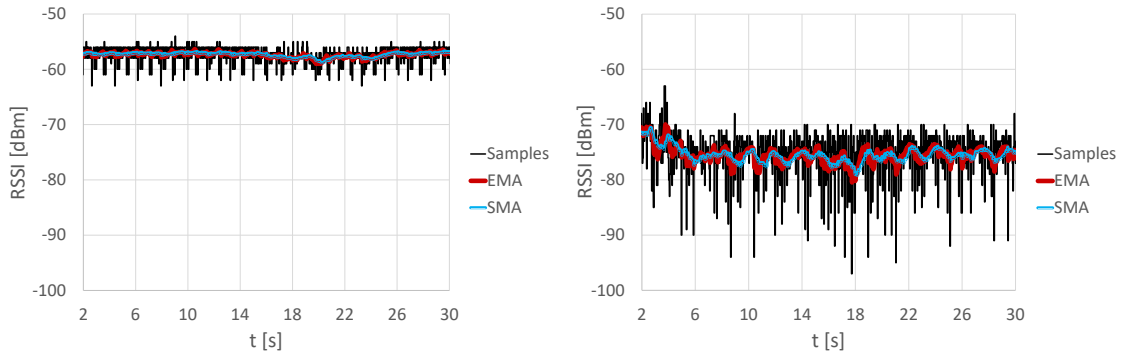
pressing an assigned button in "BLE status" group box. Important note is that for correct function of the Keyless Entry system, periodic status read-outs are necessary. For this reason, disabling both of the read-out modes should be performed only for debugging purpose.

2.8 Measurements

This section is dedicated to the measurements of RSSI for a proximity determination, and a current consumption of Keyless Entry module.

2.8.1 RSSI filtration

As mentioned earlier, RSSI measurement is key for a proximity detection. the RSSI is measured by the remote key (smartphone with an RSL10 LIN Demo app) every 40 ms and stored for later processing. In Fig. 2.10, there are measured samples of RSSI in time duration of 30s. The samples were measured in a distance of 1 and 8 meters between the remote key and Keyless Entry module, while the remote key was placed flat on a table.



(a) Proximity of 1 meter.

(b) Proximity of 8 meters.

Fig. 2.10: Measured RSSI by the Key device in still position.

It is shown, that the RSSI measurement has a high dispersion, especially for further distances, which may be caused by reflection on near objects and interference of reflected signals, as well as interference with signals from other BLE devices. In order to determine a stable value of RSSI, filtration is necessary. The filtration shall be realized with the use of an averaging. There are two general methods of averaging:

Simple Moving Average (SMA) (2.4) and *Exponential Moving Average* (EMA) (2.5). The advantage of EMA is that it does not compute with any previously measured values, so there is no need for an additional buffer. But as can be seen in Fig. 2.10, it has slightly worse averaging ability than SMA. As the dispersion gets very high in RSSI measurement, SMA was preferably used to filter the dispersion.

$$y_i = \frac{1}{N} \cdot \sum_{j=0}^{N-1} x_{i-j} \tag{2.4}$$

$$y_i = y_{i-1} \cdot \lambda + x_i \cdot (1 - \lambda) \tag{2.5}$$

The constants in equations (2.4) and (2.5) were determined as $N = 50$ and $\lambda = 0.965$, for sufficient suppression of a dispersion and because averaging with these values has very similar delay property, as can be seen in Fig. 2.11, which shows measured samples of RSSI together with averaging output values, while the remote key was moving over the Keyless Entry module to generate time varying signal.

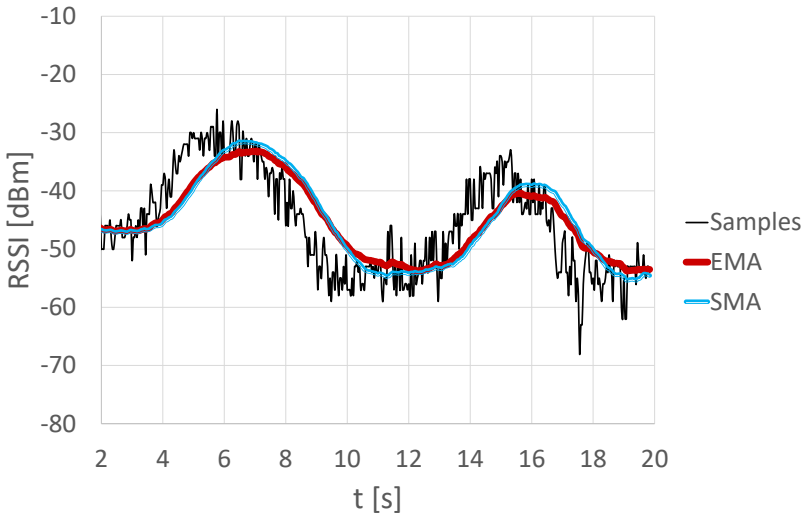


Fig. 2.11: Delay between measured RSSI and averaged value of RSSI.

2.8.2 Proximity dependency on receiver position

Determination of a proximity with the use of RSSI is based on path losses in free space. For isotropic transceivers in free space with no obstacles and interference,

the RSSI can be determined with the equation (2.6) [17].

$$\text{RSSI} = P_0 - 20 \cdot log\left(\frac{4 \cdot \pi \cdot r}{\lambda}\right), \tag{2.6}$$

where P_0 is transmitted power, r is a distance between transceivers and λ is wavelength of signal. In case of non-isotropic transceivers, this equation can be generalized to equation (2.7).

$$\text{RSSI} = P_0 - 20 \cdot log\,(r) - L_0, \tag{2.7}$$

where L_0 is attenuation dependent on signal wavelength and additionally on the directivity of the antennas. In case of any nearby obstacles, this attenuation also includes losses caused by absorption and reflection of signal on the obstacles.

The characteristics of RSSI dependent on the distance between the remote key and the Keyless Entry module can be seen in Fig. 2.12. Characteristics were measured for two general position of the smartphone realizing remote key. First in horizontal position, where the smartphone laid flat with a screen up, directed with a top side towards the Keyless Entry module. And second in vertical position, where the smartphone stood up, facing the Keyless Entry module with its back.

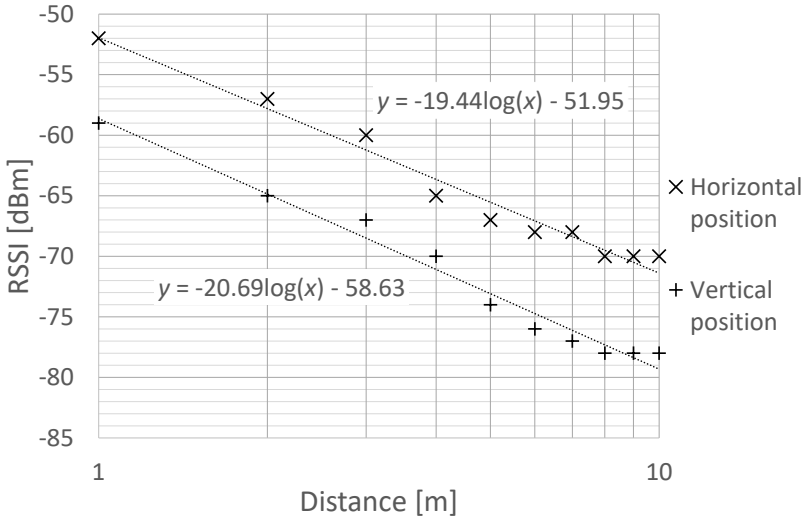


Fig. 2.12: Characteristic of measured RSSI in different positions of the remote key.

To minimize influence of losses on nearby obstacles, measurement has been conducted in an open space of a parking lot. Similarly to the previous measurement, RSSI was measured by the remote key, realized by a smartphone

with RSL10 LIN Demo app, and filtered with the use of SMA. Picture of used setup is shown in Fig. 2.13.

Fig. 2.13: Setup of RSSI measurement in open space.

By approximation of measured characteristics we can determine parameter L_0 from (2.7). Since transmitted power of Keyless Entry module is 0 dBm, L_0 is equal to the constant in logarithmic formulas in Fig. 2.12. This enables to compute the difference of directivity in measured positions of used smartphone as approximately $|L_{0\mathrm{L}} - L_{0\mathrm{S}}| = |51.95 - 58.63| = 6.68\,\mathrm{dB}$. This difference is approximate, as the characteristics are influenced by an error, which can be seen by dispersion of values, and that the characteristics trend is not exactly, but only close to $-20 \cdot log(x)$. The error is the most likely caused by the interference of nearby Wi-Fi transceivers, which emit signal on the same frequency bandwidth as BLE, and are hard to avoid nowadays.

The difference in directivity of 6.68 dB leads to the conclusion, that the RSSI cannot be used to determine exact proximity, as it would require knowledge of antenna radiation pattern for currently used smartphone, and positioning of the remote key and the Keyless Entry module. But it is sufficient for the used division of proximity into the three general zones of proximity.

2.8.3 Proximity measurement with obstacle

Aside from directivity of antenna, the path losses are also highly dependent on surrounding obstacles. Measurement of RSSI with and without nearby obstacles can be seen in Fig. 2.14. The measurement has been conducted in an open office with the use of an obstacle made of ferrite sheets arranged to a sheet (1x2) m in size (Fig. 2.15b). Similarly to the previous measurements, RSSI was measured by the remote key smartphone, laid with the screen up, top side directed towards the Keyless Entry module. Setup is pictured in the Fig. 2.15. Due to the space restriction, the characteristics has been measured only to the proximity of 6 m. In case of obstacle, the obstacle was positioned in the proximity of 1 m from the Keyless Entry module.

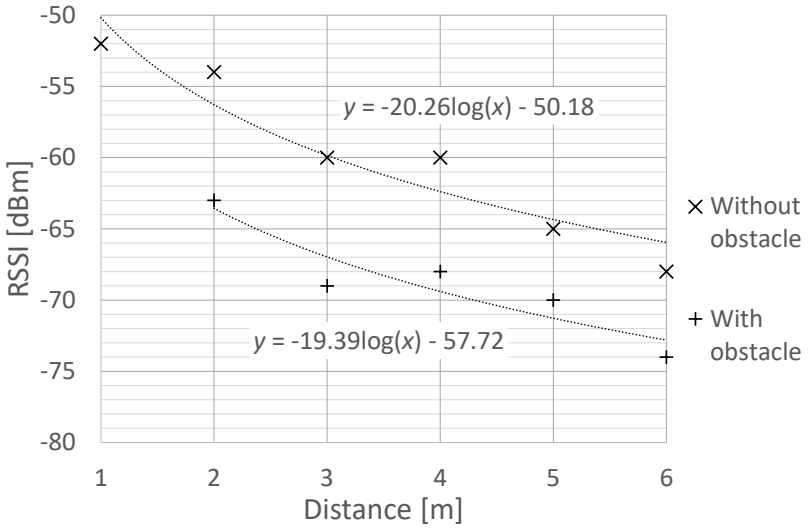
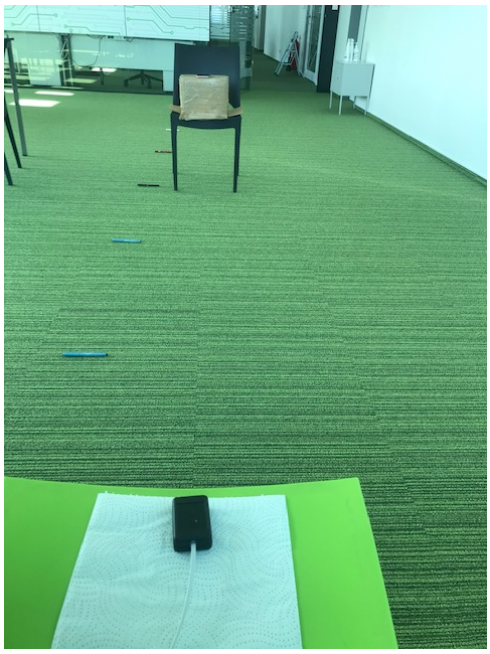


Fig. 2.14: Characteristic of measured RSSI with an obstacle.

The parameter L_0 can be again determined from approximation equations displayed in Fig. 2.14. Approximate attenuation caused by an obstacle is then computed as a difference between L_0 parameters of each characteristic $L_{0\text{obs}} - L_{0\text{noobst}} = 57.72 - 50.18 = 7.53\,\text{dB}$. This points towards further inaccuracy in proximity determination in case of an obstacle being present.

For this reason, the division of space into three proximity zones shall be done in a way, where a Near zone is at least 20 dB wide to compensate for worst case scenarios e.g. positioning the remote key in the direction of the highest gain versus positioning the remote key in the direction of the highest attenuation and someone walking/driving between the vehicle and the remote key. Furthermore, as the Immediate zone should be somewhere around 1 - 2 m, the threshold is set to

-60 dBm, and Far zone threshold shall be -80 dBm, which both makes Near zone 20 dB wide, and corresponds with the proximity of more than 10 m.



(a) Setup of RSSI measurement.

(b) Obstacle used in the measurement.

Fig. 2.15: Setup of RSSI measurement with obstacle.

2.8.4 Current consumption

The current consumption of Keyless Entry module has been performed by connecting an ammeter to the designated headers J3 and J4. For the purpose of the measurement, indicating LED has been disabled, as it is not necessary for the function of Keyless Entry module. Table of current consumption during advertising and connection with the remote key is shown in Tab. 2.2. It is noted, that the current consumption of the whole Keyless Entry module is always below 2 mA, which makes it suitable as a low power application.

I [μA]	Advertising	Connected
Module	1850	1659
RSL10	1195	867

Tab. 2.2: Table of current consumption of Keyless Entry module.

Conclusion

In this thesis, a demonstration kit for vehicle keyless entry have been designed, with the use of Bluetooth® Low Energy. The Keyless Entry demonstration kit consists of BLE Keyless Entry module and a door lock module, both connected to BCM utilizing a LIN communication protocol. The Keyless Entry module communicates with authenticated remote key realised by a smartphone with a key device application, which enables the key to lock and unlock a car door, and provides feedback to inform the user about current state of a door lock. Authentication of a remote key is based upon BLE identity resolving with Identity Resolving Key exchanged during bonding process. To ensure security of the exchange, bonding can be carried out only when the Keyless Entry system is initialized in a pairing mode, by the BCM. In normal mode, the Keyless Entry module doesn't respond to any other than bonded devices. General security of a connection between remote key and Keyless Entry module, after the bonding, is established by the usage of AES-CCM encryption and data signing, for protection against passive eavesdropping and replay attacks.

The Keyless Entry system has also ability to determine proximity based on the RSSI. The proximity is divided into three zones from Immediate (around 1-2 m) to Near and Far (more than 10 m). This feature is used to lock and unlock car door automatically, when the key device gets closer or further from a car. But this feature introduces a weakness to otherwise secure connection in case of a two-thief attack, which is critical for many of the current keyless entry systems. To resolve this issue, it is possible to disconnect a remote key in case it is not used at the moment. Furthermore, it is possible to temporarily disable this feature in case of typical scenario for such attack e.g. leaving a key connected to a car and in a place, where the signal from it can be easily received and amplified by the attackers. In the future, this issue shall be resolved with the use of next generation devices utilizing a Time of Flight feature to determine proximity instead of the RSSI.

It is most possible to showcase the demonstration kit as a standalone application, but in such case it relies on hardware buttons to execute certain asynchronous operations e.g. transition to pairing mode, which is not typical for a real-life application. Therefore a BCM software for Windows PC have been developed. When the Body Control Module is connected to the PC with USB, the software can be used to execute any operation on the Keyless Entry module from the BCM, without the need to physically interact with the PCBs.

Bibliography

[1] A. Moradi and T. Kasper, "A new remote keyless entry system resistant to power analysis attacks," *2009 7th International Conference on Information, Communications and Signal Processing (ICICS),* Macau, 2009, pp. 1-6, doi: 10.1109/ICICS.2009.5397727.

[2] T. Glocker, T. Mantere and M. Elmusrati, "A protocol for a secure remote keyless entry system applicable in vehicles using symmetric-key cryptography," *2017 8th International Conference on Information and Communication Systems (ICICS),* Irbid, 2017, pp. 310-315, doi: 10.1109/IACS.2017.7921990.

[3] Bluetooth SIG, Inc. *Bluetooth Core Specification v5.1* [online]. 21. January 2019, [cit. 3. 11. 2019]. Available from URL: <`https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=457080`>.

[4] ON Semiconductor. *RSL10 Firmware Reference* [online]. RSL10 Documentation Package, August 2019 , M-20818-015, [cit. 3. 11. 2019]. Available from URL: <`https://www.onsemi.com/support/design-resources/software?rpn=RSL10`>.

[5] Kavun, E. B., Mihajloska, H., and Yalçin, T. (2018). *A Survey on Authenticated Encryption–ASIC Designer's Perspective.* ACM Computing Surveys, 50(6), 1–21. doi: 10.1145/3131276

[6] Bon, M. *A Basic Introduction to BLE Security* [online]. 15. October 2016, [cit. 5. 3. 2016]. Available from URL: <`https://www.digikey.com/eewiki/display/Wireless/A+Basic+Introduction+to+BLE+Security`>.

[7] Lydersen L. *Wireless & IoT protocols & their security tradeoffs* [online]. 25. July 2017, [cit. 12. 3. 2016]. Available from URL: <`https://www.edn.com/wireless-iot-protocols-their-security-tradeoffs/`>.

[8] LIN Consortium. *LIN Specification Package* [online]. 31. December 2010, [cit. 22. 11. 2019]. Available from URL: <`https://www.cs-group.de/wp-content/uploads/2016/11/LIN_Specification_Package_2.2A.pdf`>.

[9] ON Semiconductor. *NCV7710 Door-Module Driver-IC (Lock Driver-IC)* [online]. April 2016 , Rev. 0, [cit. 13. 05. 2020]. Available from URL: <https://www.onsemi.com/pub/Collateral/NCV7710-D.PDF>.

[10] RivieraWaves. *RW BLE Proximity Profile Interface Specification* [online]. RSL10 Documentation Package, 12. November 2018 , Ver. 8.01, [cit. 8. 12. 2019]. Available from URL: <https://www.onsemi.com/support/design-resources/software?rpn=RSL10>.

[11] RivieraWaves. *GAP Interface Specification* [online]. RSL10 Documentation Package, 06. May 2019 , Ver. 8.23, [cit. 8. 12. 2019]. Available from URL: <https://www.onsemi.com/support/design-resources/software?rpn=RSL10>.

[12] RivieraWaves. *GATT Interface Specification* [online]. RSL10 Documentation Package, 30. January 2019 , Ver. 8.05, [cit. 8. 12. 2019]. Available from URL: <https://www.onsemi.com/support/design-resources/software?rpn=RSL10>.

[13] A. I. Alrabady and S. M. Mahmud, "Analysis of attacks against the security of keyless-entry systems for vehicles and suggestions for improved designs," in *IEEE Transactions on Vehicular Technology,* vol. 54, no. 1, pp. 41-50, Jan. 2005, doi: 10.1109/TVT.2004.838829.

[14] Android Developers. *Distribution dashboard* [online]. 7. May 2019, [cit. 21 02 2020]. Available from URL: <https://developer.android.com/about/dashboards>.

[15] Android Developers. *BluetoothAdapter* [online]. 27. Dec 2019, [cit. 17 04 2020]. Available from URL: <https://developer.android.com/reference/android/bluetooth/BluetoothAdapter>.

[16] Android Developers. *Intent* [online]. 05. May 2020, [cit. 13 05 2020]. Available from URL: <https://developer.android.com/reference/android/content/Intent>.

[17] M. Kasal, *Směrové a družicové spoje: přednášky* Brno: Brno University of Technology, 2003. p. 13. ISBN: 80-214-2496-6.

List of symbols, physical constants and abbreviations

ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
API	Application Programming Interface
ATT	Attribute Protocol
BCM	Body Control Module
BLE	Bluetooth® Low Energy
EMA	Exponential Moving Average
CAN	Controller Area Network
CCM	Cipher block Chaining - Message authentication code
CSRK	Connection Signature Resolving Key
DMA	Direct Memory Access
EVB	Evaluation Board
GAP	Generic Access Profile
GATT	Generic Attribute Profile
GUI	Graphical User Interface
IDE	Integrated Development Environment
IAS	Immediate Alert Service
IC	Integrated Circuit
IRK	Identity Resolving Key
IVN	In-Vehicle Networking
LDO	Low-drop Voltage Regulator
LIN	Local Interconnect Network
LLS	Link Loss Service
LTK	Long Term Key
LSB	Least Significant Bit or Byte
NFC	Near Field Communication
MITM	Man-In-The-Middle
MSB	Most Significant Bit or Byte
OOB	Out Of Band
PCB	Printed Circuit Board
PID	Protected Identifier
prand	parallel random number
PWM	Pulse Width Modulation
RF	Radio Frequency
RSSI	Received Signal Strength Indicator

SBC	System Basis Chip
SIP	System-in-Package
SMA	Simple Moving Average
SPI	Serial Peripheral Interface
SoC	System on Chip
TxPS	Tx Power Service
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus
UUID	Universally Unique Identifier

List of appendices

A Keyless Entry module PCB

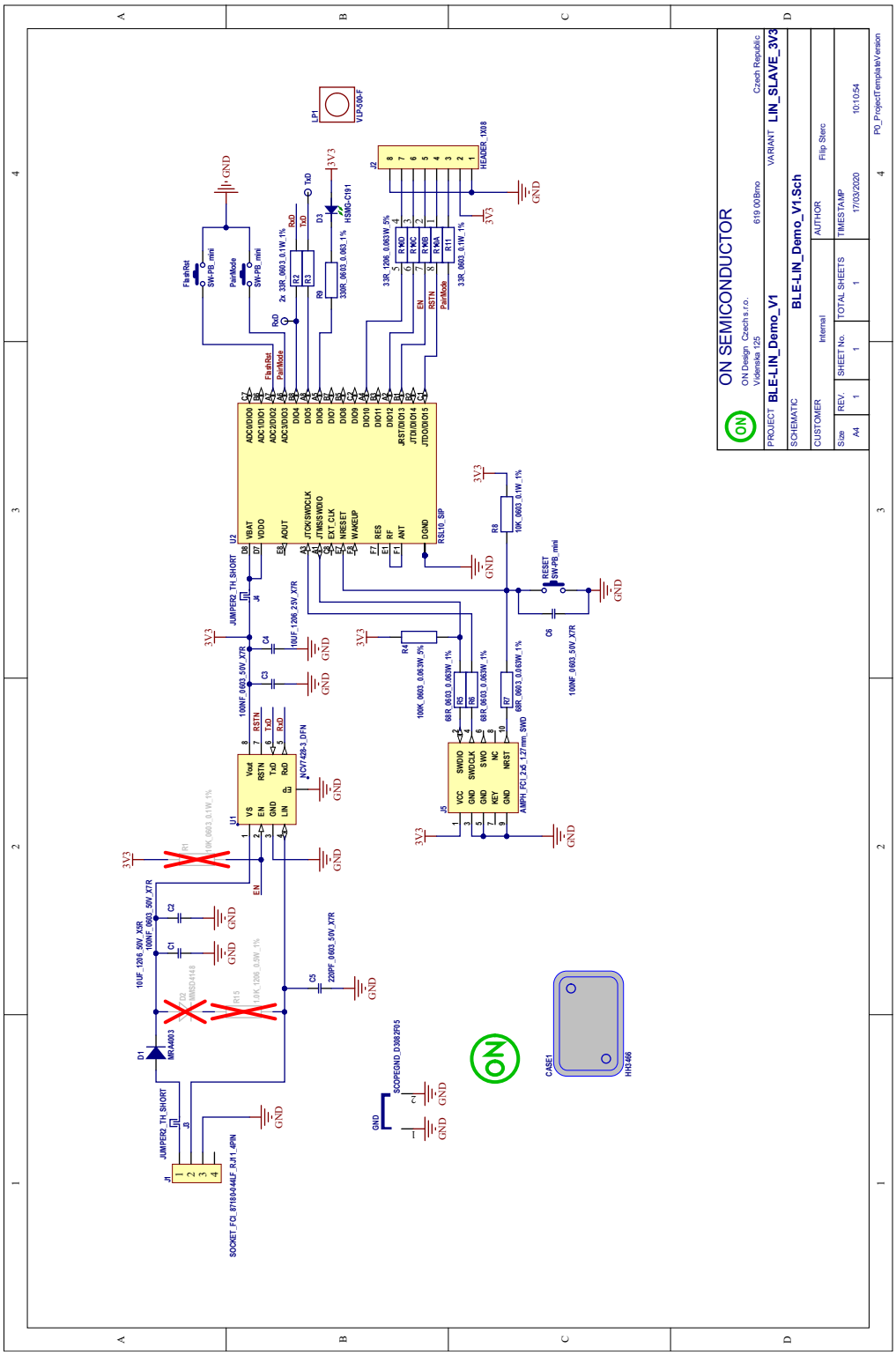


Fig. A.1: Schematic of Keyless Entry module.

CASE1

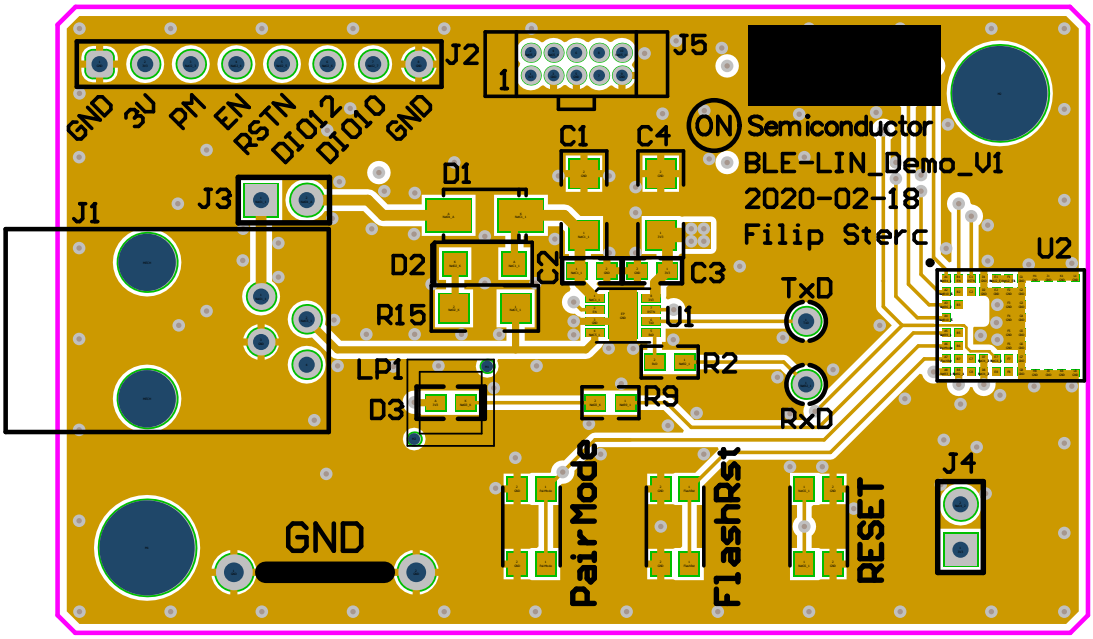


Fig. A.2: Top layout of Keyless Entry module.

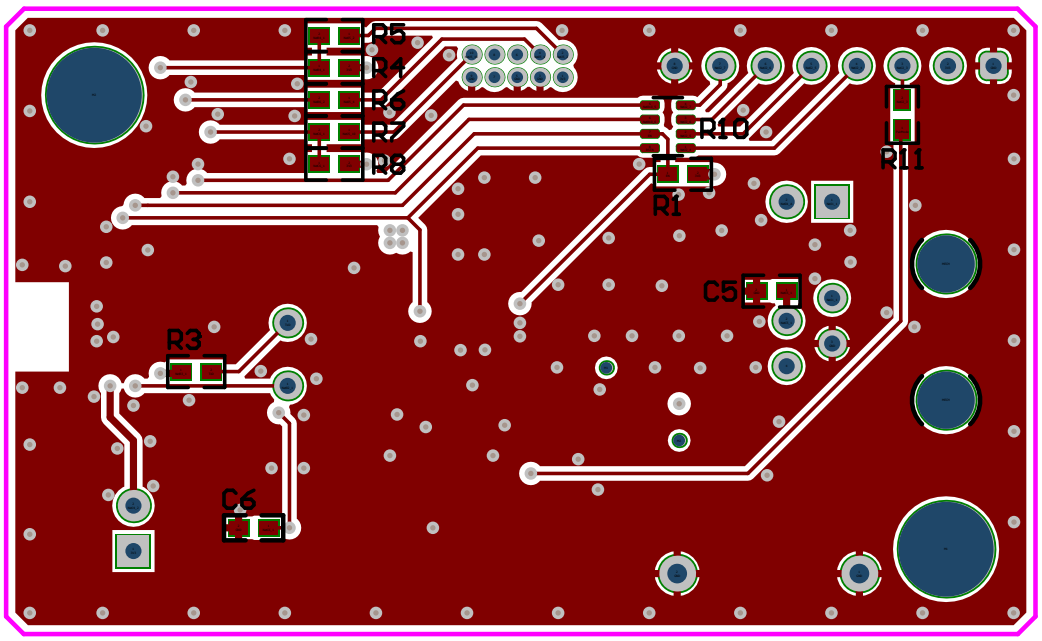


Fig. A.3: Bottom layout of Keyless Entry module.

Fig. A.4: Picture of Keyless Entry module realization.

B Body Control Module PCB

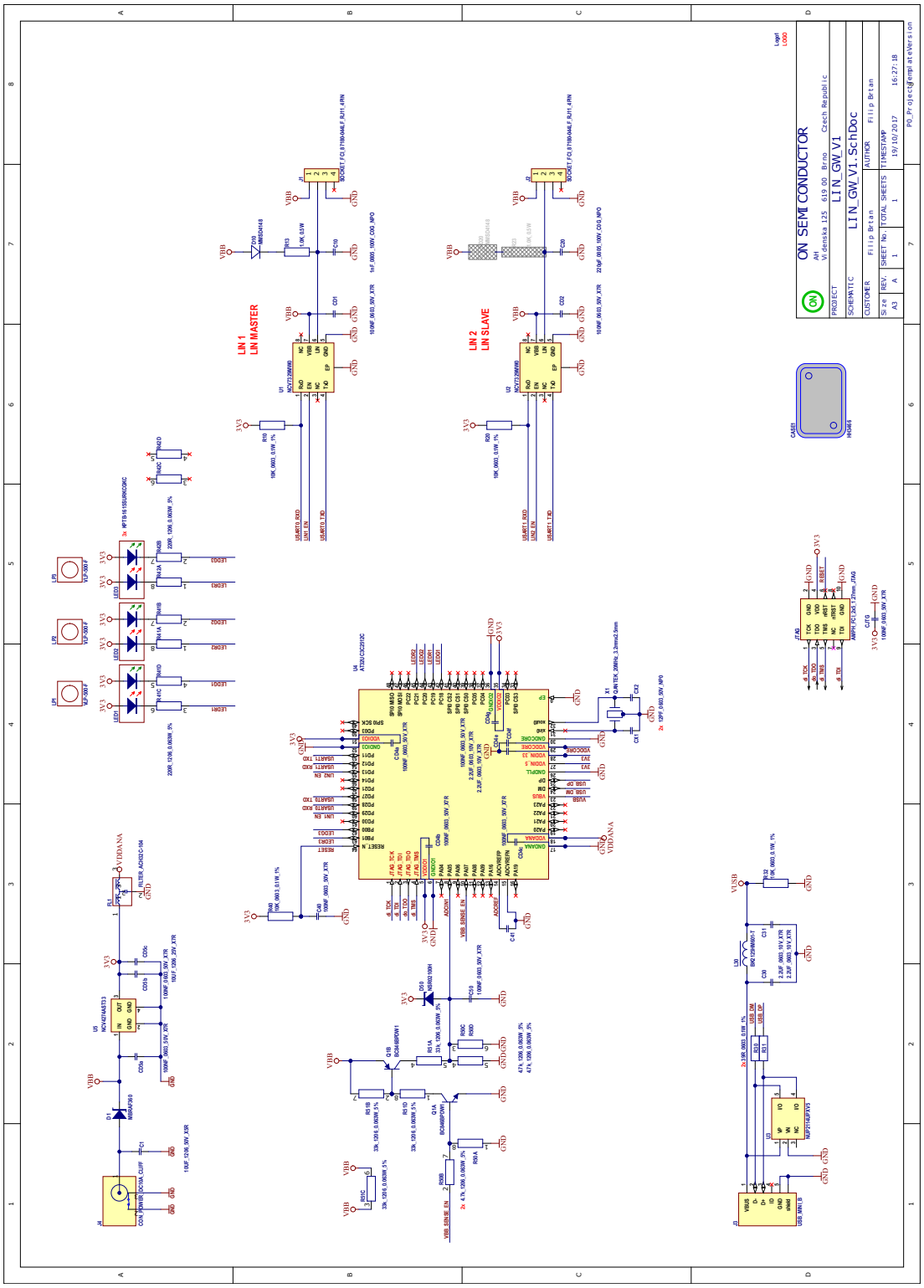


Fig. B.1: Schematic of Body Control Module.

CASE1

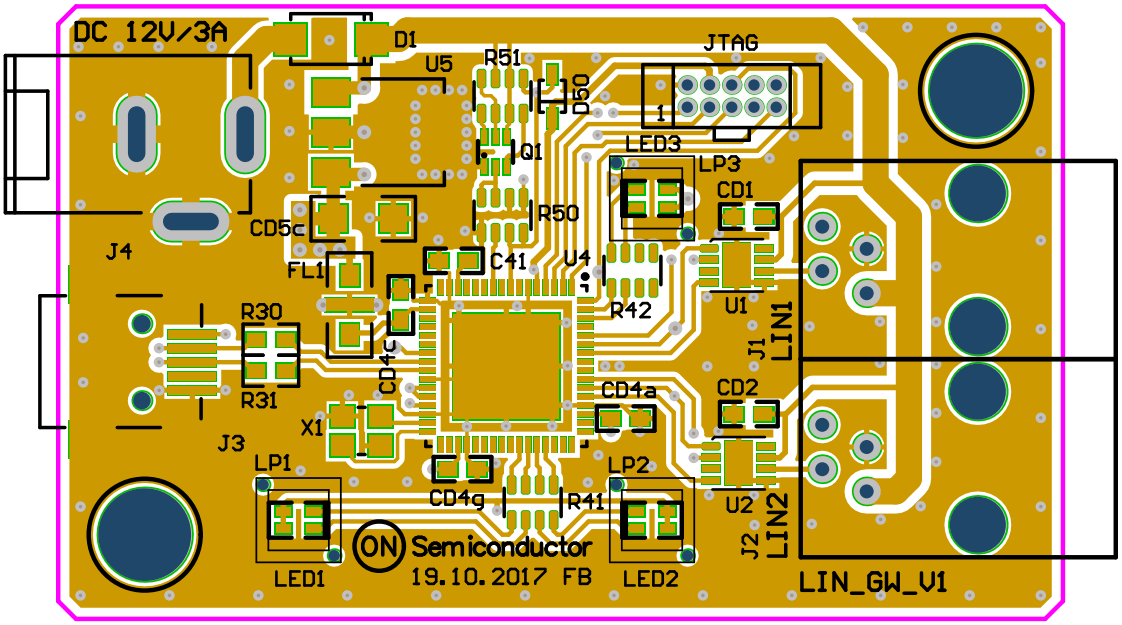


Fig. B.2: Top layout of Body Control Module.

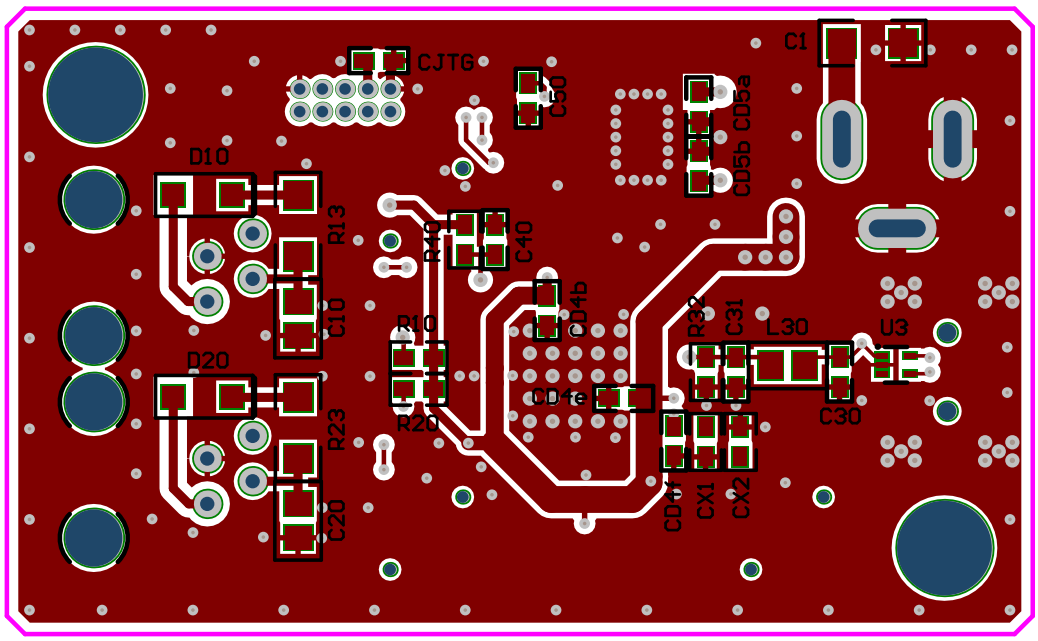


Fig. B.3: Bottom layout of Body Control Module.

Fig. B.4: Picture of Body Control Module realization.

C Door lock module PCB

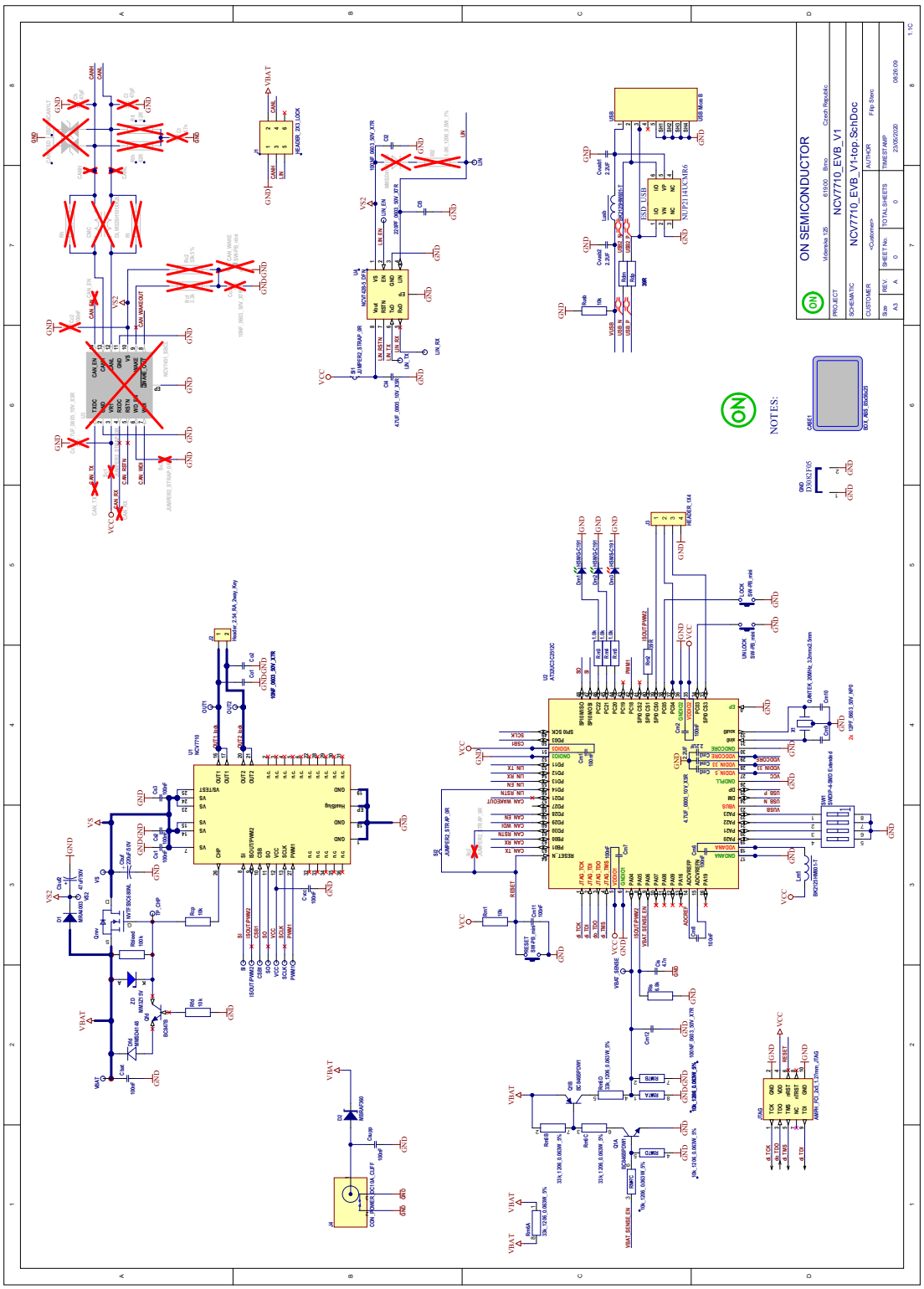


Fig. C.1: Schematic of door lock module.

CASE1

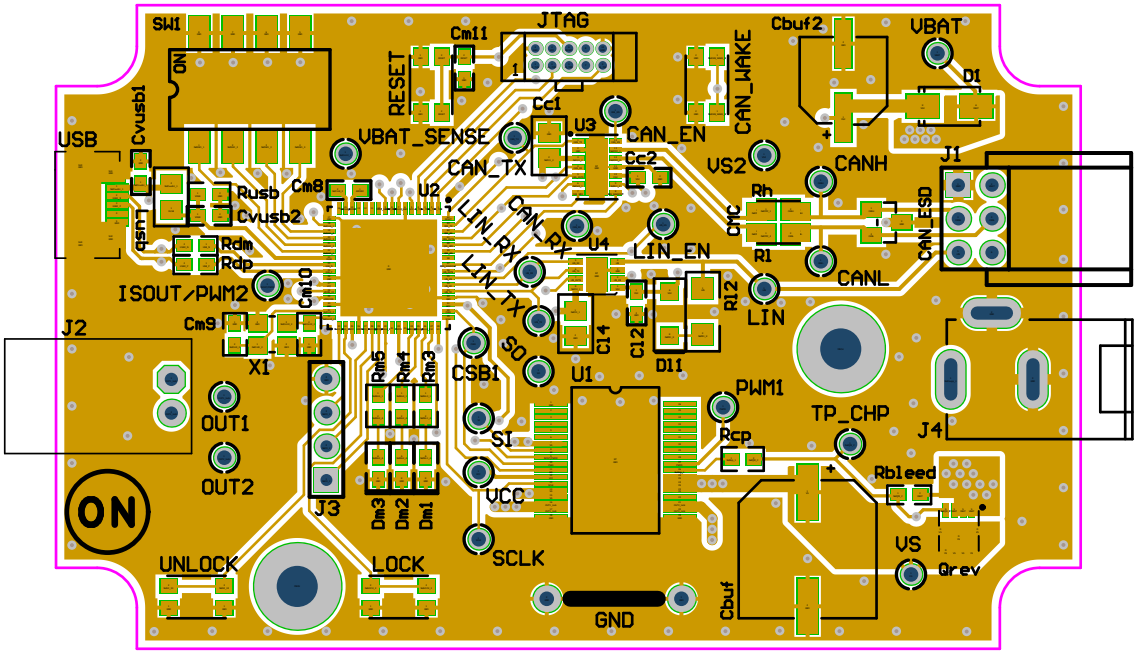


Fig. C.2: Top layout of door lock module.

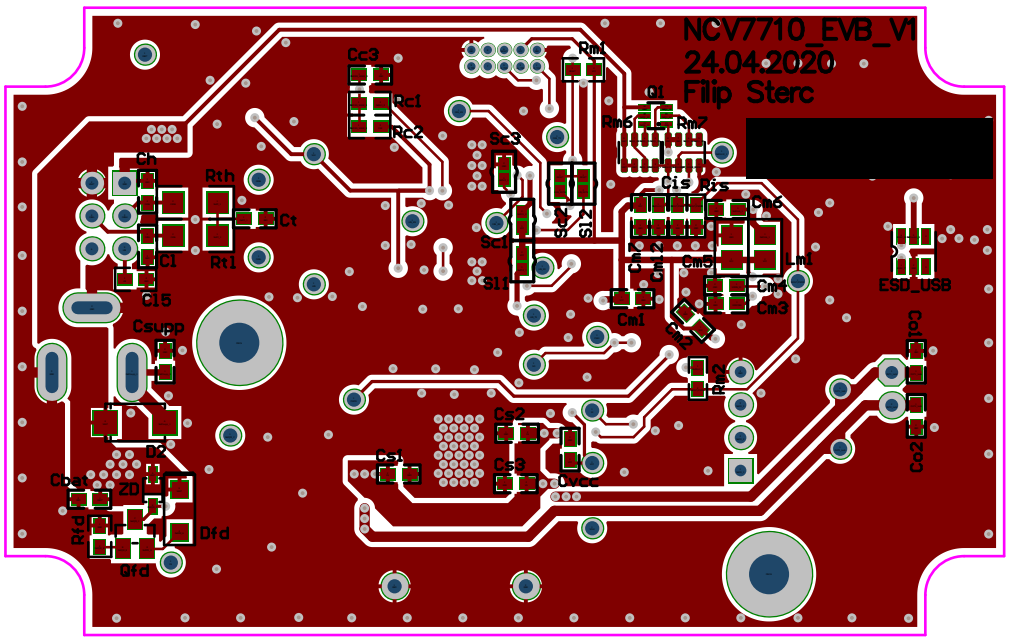


Fig. C.3: Bottom layout of door lock module.

Fig. C.4: Picture of door lock module realization.

D Supplement content

Supplement includes all of the code used in this master's thesis. The firmware for BCM and door lock module has been created in Atmel Studio 7. Keyless Entry module firmware has been created in ON Semiconductor IDE 3.2. The RSL10 LIN application has been made for Android Studio 3.5 in Kotlin programming language. And the BCM GUI software has been made with a Qt creator IDE 3.6.

```
/ ..... Root directory of supplement
├── Keyless_Entry ..... Firmware for Keyless Entry module PCB
│   ├── Include ..... Directory with .h files
│   ├── Source ..... Directory with .c files
│   ├── .cproject ..... Project startup file
│   └── ...
├── BCM_firmware ..... Firmware for BCM PCB
│   ├── src
│   │   ├── board ..... Configuration and PCB initialization source code
│   │   ├── modules ..... Libraries for AT32UC3C peripherals
│   │   ├── main.c
│   │   └── ...
│   ├── LIN_GW_N430.atsln ..... Project startup file
│   └── ...
├── NCV7710_EVB_V1 for BLE_LIN Demo ..... Firmware for door lock module PCB
│   ├── src
│   │   ├── board ..... Configuration and PCB initialization source code
│   │   ├── modules ..... Libraries for AT32UC3C peripherals
│   │   │   ├── ncv7710 ..... Source code for NCV7710 control
│   │   │   └── ...
│   │   ├── main.c
│   │   └── ...
│   ├── N710_EVB.atsln ..... Project startup file
│   └── ...
├── BLE_LIN_Demo ..... Project for RSL10 LIN kotlin application
│   ├── app/src/main/java/com/example/ble_lin_demo
│   │   ├── MainActivity.kt ..... Scan activity code
│   │   ├── DeviceControl.kt ..... Control activity code
│   │   ├── BLEService.kt
│   └── ...
└── BCM_Software ..... Project for BCM GUI in Qt
    ├── bench ..... Main window source code directory
    ├── bcm.cpp ..... Source code for BCM GUI widget
    ├── bcm.ui
    ├── bcm.h
    ├── project_app.pro ..... Project startup file
    └── ...
```