Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Gašper Kojek

Primerjava učinkovitosti prenosa podatkov v standardih Bluetooth

MAGISTRSKO DELO

MAGISTRSKI PROGRAM DRUGE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2018

University of Ljubljana
Faculty of Computer and Information Science

Gašper Kojek

Comparing data transmission efficiency in Bluetooth Standards

MASTER'S THESIS

THE 2nd CYCLE MASTER'S STUDY PROGRAMME
COMPUTER AND INFORMATION SCIENCE

Supervisor: izr. prof. dr. Patricio Bulić

Co-supervisor: doc. dr. Anton Biasizzo

Ljubljana, 2018

Acknowledgments

First and foremost I would like to thank my fiancee Eva for all the love and support she gave me during the writing of this thesis and through the course of my studies.

Furthermore, I would like to thank my family and friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

I would like to thank my mentor izr. prof. dr. Patricio Bulić for his help with my thesis. I would also like to thank my co-mentor doc. dr. Antoin Biasizzo from Jožef Stefan Institute for his counselling and guidance with the technical challenges I encountered during my research.

A special thanks goes to Foolography GmbH and its CEO Oliver Perialis for providing me with the hardware needed to conduct the measurements and kind-hearted support.

Gašper Kojek, 2018

Contents

List of Figures

List of Tables

List of used acronyms

acronym	**meaning**
IoT	Internet of Things
SIG	Special Interest Group
BR/EDR	Basic Rate/Enhanced Data Rate
BLE	Bluetooth Low Energy
HS	High Speed
LR	Long Range
PHY	physical layer
L2CAP	Logical Link Control and Adaptation Protocol
GAP	Generic Access Profile
ATT	Attribute Protocol
GATT	Generic Attribute Profile
MTU	Maximum Transmission Unit
DLE	Data Length Extension
SoC	System on Chip
FEC	Forward Error Correction
UUID	Universally Unique IDentifier
CE	Connection Event
CI	Connection Interval
SoC	System-on-Chip
MCU	Microcontroller Unit
DK	Development Kit
SDK	Software Development Kit
KB	kilobyte (1024 bytes)
kbps	kilobits per second (1000 bits per second)

Povzetek

Naslov: Primerjava učinkovitosti prenosa podatkov v standardih Bluetooth

Ta magistrska naloga predstavlja opravljeno delo,potrebno za meritve in primerjavo ~~moč~~ Bluetooth Low Energy energetske učinkovitosti. Predstavljene so primerjave porabe električne energije glede na pretočnost med različnimi verzijami specifikacije BLE.

energetske uinkovit

Predstavljena je kratka zgodovina Bluetooth Low Energy, kot tudi glavne izboljšave in razlike med specifikacijami. Meritve so bile opravljene z enim parom naprav, katere so imele parametre nastavljene glede na različico BLE, na kateri je test temeljil. Porabo energije smo merili z različnimi parametri, pri branju, pisanju, obveščanju in pisanju brez odziva.

Namesto termina specifi

dvojina

Po pričakovanjih so rezultati pokazali, da novejše verzije BLE ponujajo večjo pretočnost, pa tudi dejstvo, da je povprečna poraba energije neodvisna od količine podatkov, ki jih je treba poslati. Zanimivo je, da se z vsako novo različico BLE poraba energije na preneseno količino podatkov zmanjša, tudi če se poveča pretočnost. Ugotovljeno je bilo tudi, da je energija, potrebna za prenos določene količine podatkov, konstantna pri spreminjanju pretočnosti pri določeni različici specifikacije. Ugotovitve v tej diplomski nalogi dokazujejo, da ni razloga, da ne bi prešli na razvoj in uporabo naprav, ki uporabljajo tehnologijo Bluetooth Low Energy 5, ker prinaša večjo pretočnost, večji doseg in boljše sožitje z drugimi brezžičnimi tehnologijami, hkrati pa porabi isto ali manjšo količino energije za prenos podatkov.

ne vem, kaj si hot

Ključne besede

Bluetooth, zmogljivost, pretočnost, energetska učinkovitost, meritve, nizka poraba

Abstract

Title: Comparing data transmission efficiency in Bluetooth Standards

This master thesis presents the work done to be able to measure and compare Bluetooth Low Energy power efficiency. More specifically, comparisons between different BLE core specification versions are made in regards to power consumption in relation to throughput.

A brief history of Bluetooth Low Energy is presented, as well as the main improvements and differences between core specification versions. Measurements were performed with a single pair of devices, which had their connection parameters tuned in line with BLE version they were targeting. Power consumption was measured with different parameters, with read, write, notify and write without response BLE operations.

As expected, results revealed that newer BLE versions offer higher throughput, as well as the fact that average power consumption is independent from the amount of data needed to be transmitted. Interestingly enough, with each BLE version iteration, power consumption per transmitted amount of data is decreasing, even when the throughput is increased. It was also discovered that total energy needed to transmit some amount of data is constant when changing the throughput in a single core specification version. The findings in this thesis prove that there is no reason not to switch to developing and using devices that use Bluetooth Low Energy 5, as it brings higher throughput, extended range and better coexistence while using the same or lower amount of power for transmissions.

they bring

consuming

energy

Keywords

Bluetooth, performance, throughput, power efficiency, measurements, low energy

Razširjeni povzetek

V zadnjih letih smo lahko opazili velik napredek v digitalnih tehnologijah, katere prispevajo k vedno večjemu pojmu ”internet stvari”. Velika večina IoT izdelkov uporablja za komunikacijo Bluetooth, tako da je le-ta še vedno v razvoju in se nenehno izboljšuje. Kljub temu nismo zasledili nobene znanstvene študije, ki bi primerjala porabo energije v primerjavi s pretokom pri različnih parametrih povezave in z različnimi Bluetooth Low Energy različicami. To je še posebej zanimivo v povezavi z novim standardom Bluetooth 5, ki obljublja dvakratno hitrost, štirikraten doseg in osemkrat boljšo zmogljivost oddajanja, ki naj ne bi vplivala na porabo moči. Predstavili smo primerjave porabe električne energije glede na pretočnost med različnimi verzijami specifikacije BLE.

(IoT)

I Uvod

Primerjava porabe energije za prenos podatkov z uporabo različnih tehnologij ni nič novega, je pa izjemno zanimivo, da nismo našli nobene take raziskave, ki bi primerjala različne verzije tehnologije Bluetooth. Našli smo nekaj raziskav, ki primerjajo starejše Bluetooth Low Energy tehnologije z ostalimi tehnologijami, kot na primer ZigBee in 802.15.4 [1] ali pa 6LoWPAN, IrDA, NFC ali ANT [2, 3]. Prav tako smo našli nekaj različnih raziskav ali člankov o meritvah porabe energije z uporabo padca napetosti preko šhunt” upora [4, 5, 6].

na zaporednem uporuna napajalnem upor

II Kratka zgodovina standarda Bluetooth

Standard Bluetooth je zasnoval dr. Jaap Haartsen leta 1994. Poimenovan je bil po danskem kralju, katerega so klicali Bluetooth. Leta 1998 je bila ustanovljena posebna skupina interesentov za standard Bluetooth (*Bluetooth Special Interest Group (SIG)*) in leta 1999 je bila izdana prva verzija standarda Bluetooth, imenovana Bluetooth 1.0. Čeprav uporablja isto frekvenčno območje kot Wi-Fi, je bil Bluetooth že od vsega začetka zasnovan kot tehnologija z veliko nižjo porabo energije in manjšim dosegom.

Leta 2002 je bila izdana različica 2.0, katera je prinesla nekaj zelo pomembnih izboljšav. Najpomembnejše izmed izboljšav so bile povečana prepustnost zaradi dodatka EDR (*Enchanced Data Rate*), povečan doseg na okoli 30 m ter za polovico manjša poraba energije. Verzija 2.1 je bila izdana leta 2007 in je prinesla poenostavljen proces vzpostavljanja povezave, kateri je bil hkrati tudi bolj varen. Izboljšali so porabo energije za naprave, katere so večji del časa neaktivne. Leta 2009 je izšla verzija 3.0, katera je izjemno povečala prepustnost, saj je v kombinaciji z Wi-Fi povezavo omogočala hitrosti prenosa podatkov do 24 Mbps.

Bluetooth SIG je leta 2010 uradno izdal specifikacije verzije 4.0, katera prvič doda Bluetooth Low Energy tehnologijo. Le-ta je prinesla nove načine povezovanja, prav tako pa je prvič omogočila delovanje naprav, katere uporabljajo gumbno baterijo, v trajanju nekaj let. Leta 2013 je bila izdana verzijo 4.1, katera je prinesla veliko izboljšav v programskem delu, saj je omogočila razvijalcem več kontrole nad parametri naprave, prav tako pa je omogočila direktno povezovanje naprav Bluetooth med seboj, brez posrednika.

ki

Verzija 4.2 standarda je bila izdana leta 2014 in je prinesla več novosti, katere se s pridom uporabljajo še danes. Največja sprememba je bila povečanje maksimalne dolžine paketka, katera je prinesla približno desetkratno pohitritev. Dodali so tudi možnost direktne povezave na internet ter izboljšali varnost povezave.

Standard Bluetooth 5 je bil uradno izdan leta 2016, vendar smo prve naprave dobili v letih 2017 in 2018. Verzija 5 je prinesla dvakratno pohitritev prenosa podatkov, štirikrat daljši doseg (do 120 metrov) in osem krat večjo zmogljivost oddajanja. Hkrati se je izboljšalo sobivanje z drugimi brezžičnimi tehnologijami ter poenostavilo poimenovanje, saj je sedaj uradno ime Bluetooth 5, brez decimalk.

III Bluetooth Low Energy

Standard *Bluetooth Low Energy (BLE)*, poznan tudi kot *Bluetooth Smart* je bil prvič predstavljen v verziji 4.0 standarda Bluetooth kot lahek del standarda Bluetooth, kateri porabi izjemno malo energije. Čeprav si s klasičnim Bluetooth-om deli nekaj funkcionalnosti in ime, imata ta dva dela standarda povsem drugačne lastnosti in funkcionalnosti. *Bluetooth Low Energy* je razdeljen na tri glavne nivoje: kontroler, gostitelj in aplikacija. Ti so naprej razdeljeni na dele, kot je razvidno na Sliki 3.1. Večinoma so vsi trije nivoji implementirani na enem integriranem vezju, za nas pa so najpomembnejši deli fizični nivo (*Physical Layer - PHY*), generični profil dostopa (*Generic Access Profile - GAP*) in profil splošnih atributov (*Generic Attribute Profile - GATT*).

krmilnik

Fizični nivo je najnižji nivo sklada BLE in vsebuje analogno vezje za prevajanje, pošiljanje in prejemanje digitalnih simbolov preko elektromagnetnih valov. BLE oddajnik deluje v 2.4 GHz ISM frekvenčnem pasu. Standardi BLE verzij v4.0, v4.1 in v4.2 pošiljajo podatke pri 1 Mbps, standard BLE 5 pa doda še tri načine delovanja: visoko-hitrostni 2 Mbps ter dva načina za večji doseg pri 125 kbps in 500 kbps. Zadnja dva načina prav tako uporabljata modulacijo pri 1 Mbps, vendar je ~~tam~~ vsak bit zakodiran z večimi simboli, kar zniža prepustnost, vendar pomeni tudi, da se biti pravilno razpoznajo na večji razdalji. To hkrati pomeni, da za standard BLE 5 oglaševana dvakratna prepustnost in štirikraten doseg v resnici predstavljata dvakratno prepustnost ALI štirikraten doseg.

ali je mišljeno kodiranje

s hitrostjo

vendar omogoa odprav

Povezovalni nivo (*Link Layer - LL*) je nivo, ki povezuje fizični nivo in gostiteljski del. Ker je zadolžen za upravljanje vseh časovnih omejitev je po navadi ločen od višjih nivojev sklada. Ta nivo je zadolžen za oglaševanje, iskanje ter vzpostavitev in vzdrževanje povezav. Povezava v standardu BLE pomeni zaporedje prenosa paket~~k~~ov ob vnaprej določenih časih. Čas med paket~~k~~i je eden izmed parametrov povezave, ki se imenuje povezovalni interval (*connection interval*), skupek paket~~k~~ov poslan ob tem času pa povezovalni dogodek (*connection event*). Ko je povezava vzpostavljena LL skrbi za zanesljiv prenos podatkov na višjih nivojih. Povezovalni interval je ključen pri zagotavljanju nizke porabe energije, saj je oddajnik vključen le ob povezovalnih dogodkih, drugače pa je izključen, kar prihrani izjemno veliko energije.

asovnih intervalih

nivo

Generični profil dostopa (*Generic Access Profile - GAP*) uporablja nižje nivoje za definiranje višjenivojskih vlog, procedur in načinov delovanja, ki omogočajo napravam prenos podatkov, odkrivanje servisov, upravljanje povezav in varnosti. To je temeljni profil naprav BLE, kateri je obvezen za vse naprave. V standardu BLE določa GAP štiri vloge naprav, katere lahko uporabljajo eno ali več vlog hkrati. Vlogi opazovanca in opazovalca se uporabljata za pošiljanje podatkov večim opazovalcem hkrati, kateri niso povezani z opazovancem. Vlogi centralne in periferne naprave se uporabljata za prenos podatkov preko vzpostavljene povezave med dvema napravama. Centralna naprava podpira več hkratnih povezav na več perifernih napravah in je tista naprava, katera vzpostavi in kontrolira povezavo.

Nivo GAP v sta

Tule ni jasno, kdo lahko

Profil splošnih atributov (*Generic Attribute Profile - GATT*) definira načina kako naprave BLE prenašajo podatke z uporabo konceptov imenovanih servisi in karakteristike. GATT definira, kako se uporablja nižje nivoje za odkrivanje, branje, pisanje, oddajanje in sprejemanje podatkov. Podatki so organizirani hierarhično, v skupine imenovane servisi, katere združujejo konceptualno povezane skupke podatkov imenovane karakteristike, katere si lahko predstavljamo kot zabojnike podatkov. Hierarhijo podatkov lahko vidimo na Sliki 3.3. GATT prav tako poda referenčno ogrodje profilov, definiranih s strani Bluetooth SIG. Ti osnovni profili zagotavljajo skupno delovanje naprav različnih proizvajalcev. GATT definira dve vlogi naprav, GATT server, ki hrani podatke, GATT klient, ki se poveže na GATT server ter bere, piše in sprejema podatke. Najbolj pogosta konfiguracija naprav je

kateri

skupinemr

tu ni jasno, ali se to na

doloa

ti konfiguraciji nista edini možni.(ali) so mc

GAP periferna naprava, v vlogi GATT serverja in GAP centralna naprava v vlogi GATT klienta, vendar ta konfiguracija ni edina možna. Vsi GATT atributi, karakteristike in servisi se naslavljajo z uporabo edinstvenih identifikatorjev (UUID). GATT definira postopke za upravljanje s podatki in konfiguracijami naprav. Postopka branja in pisanja podatkov nadzoruje GATT klient. Za ta dva postopka je potrebna potrditev GATT serverja, kar se zgodi v naslednjem povezovalnem dogodku, ki sledi poslanemu zahtevku. Postopek pisanja podatkov brez potrdila začne GATT klient, vendar mu ni potrebno čakati na potrdilo GATT serverja, da lahko pošlje naslednji paketek. Postopka obveščanja in indikacije podatkov začne GATT server ob vnaprejšnji konfiguraciji s strani GATT klienta. Indikacija podatkov zahteva potrditev s strani GATT klienta, medtem ko obveščanje tega ne potrebuje, zaradi česar je prenos podatkov na ta način veliko hitrejši.

Nisi definiral atribut

komunikacijskega paketap

Strukturo paket~~k~~a standarda BLE lahko vidimo na Slikah 3.4 in 3.5. Velikost podatkov je odvisna od različice standarda BLE. V verzijah v4.0 in v4.1 je bila ta maksimalno 27 bajtov, od različice v4.2 naprej pa je 251 bajtov. Od tega je potrebno odšteti 4 bajte za glavo L2CAP nivoja, s čimer pridemo do 23 bajtov, oziroma 247 bajtov, kar je definirano v parametru povezave imenovanem *ATT_MTU*. Paketek nivoja ATT vsebuje še nadaljnje 3 bajte glave, s čimer pridemo do največjega števila podatkov za prenos na nivoju GATT, ki je 20 oziroma 244 za različico v4.2 in kasnejše različice.

IV Metodologija testiranja

Testiranje se je izvajalo s pomočjo dveh Nordic Semiconductor nRF52840 DK [7] razvojnih plošč, od katerih je bila ena konfigurirana kot periferna naprava na kateri se je merila poraba energije, druga pa kot centralna naprava, katera je upravljala s testi.

Za meritve se je uporabilo *Power Profiler Kit* [8] proizvajalca Nordic Semiconductor. Ta sistem je razvit posebej za meritve porabe energije vgrajenih sistemov kakršne so naprave Bluetooth. Sistem ima tronivojsko analogno meritveno vezje, katero za zagotavljanje večje natančnosti razdeli območje merjenja od 1 µA do 70 mA na tri razpone, kar lahko vidimo na Tabeli 4.1. Meritveni sistem prikazuje meritve na priklopljenem računalniku. Meritve so časovno omejene na maksimalno dolžino 120 s. Primer meritve z izbranimi podatki enega testa lahko vidimo na Sliki 4.1.

Periferna naprava, na kateri se je merila poraba je bila konfigurirana tako, da so bili vsi deli vezja, kateri niso nujno potrebni za delovanje povezave Bluetooth izklopljeni, saj bi drugače vplivali na rezultate meritev. Za lažjo sinhronizacijo testov in rezultatov je bil na periferno napravo dodan 470 Ω upor, skozi katerega je v času med dvema testoma tekel električni tok, v samem trajanju testa pa je bil le-ta odklopljen. S tem je bilo v rezultatih meritev zelo enostavno opaziti kdaj se je test začel in končal.

Na ta nain

Firmware je bil spisan posebno za potrebe teh testov. Projekti za centralno in periferno napravo so bili ločeni, vendar so uporabljali iste module za povezovanje. Arhitektura *firmware-a* je vidna na Sliki 4.5. Centralno in periferno jedro sta modula, katera skrbita za izvajanje testov, ki so definirani v modulu testnih parametrov. Centralna in periferna BLE modula ter modul BLE abstrakcije skrbijo za visoko abstrakcijo funkcij potrebnih za izvajanje testov. Delujejo kot lepilo med komponentami sklada BLE, implementiranimi s strani Nordic Semiconductor, ter višjih modulov. Periferna naprava vsebuje lasten servis z dvema karakteristikama, kot je vidno na Sliki 4.6. Karakteristika kontrole testov se uporablja za nastavljanje parametrov testa ter začetek in konec testa. Karakteristika testnih podatkov se uporablja za dejanski prenos podatkov med testom.

Odločeno je bilo, da se ne bo testiralo ~~dodat~~nih načinov delovanja standarda BLE 5, ki kateri zagotavljajo večji doseg, saj te načini niso namenjeni za visoko ti pretočnost podatkov. Prav tako pri razdaljah, katere ti načini omogočajo navadni načini ne delujejo, tako da bi bila primerjava brezpredmetna, saj gre za povsem drug o namembnost~~i~~. Zaradi časovne omejitve 120 s merilnega sistema smo pred testi izračunali teoretičen čas trajanja prenosa podatkov različnih testov, kar lahko vidimo v Tabelah 4.2 in 4.3. Zaradi (pre)dolgega

Odloili smo se, da nainov delovanja standarda BLE 5 za zagotavljanje vejega dosega ne bomo te trajanja prenosa podatkov pri nekaterih testnih parametrih, smo zasnovali dva sklopa testov. Pri prvem testu smo testirali neodvisnost povprečne porabe energije od količine prenesenih podatkov nad neko mejo, pri drugem pa porabo energije v odvisnosti od pretočnosti. Parametre testov lahko vidimo v Tabeli 4.4 za prvi in Tabeli 4.5 za drugi sklop testov.

V Rezultati in analiza

Paketov ne vidimo, lahko

za prenos 100 bajtov pod

smo predvideli

Ne glede na kolii

na nivoju GATT

potrjevanje?

Slike 5.1, 5.2 in 5.3 prikazujejo primere meritev za tri različne primere testov. Na sliki 5.1 lahko vidimo posamezne paket~~k~~e operacije pisanja, poslane v posameznih povezovalnih dogodkih. Kot predvideno v Tabeli 4.2 je bilo potrebnih 5 poslanih paketkov za prenos 100 bajtov podatkov, kateri so zasedli 10 povezovalnih dogodkov in potrebovali 75 ms. Slika 5.2 prikazuje branje podatkov z različico v4.2 standarda BLE, ki pa še vedno zahteva potrditev s strani GATT serverja, zato vsako branje porabi dva povezovalna dogodka. Slika 5.3 prikazuje prenos podatkov z različico BLE 5 z uporabo operacij obveščanja, za kar potrditve niso potrebne, temveč se opirajo le na dogovore nižjega povezovalnega nivoja. Zaradi tega je lahko poslanih več paket~~k~~ov v vsakem povezovalnem dogodku, kar izjemno poveča pretočnost podat~~k~~ov. Vidimo lahko tudi kdaj se začne vsak povezovalni dogodek, saj se pred tem oddajnik za kratek čas izklopi, da pusti morebitnim ostalim povezavam prostor za komunikacijo.

meritev odvisnosti

Rezultate testa neodvisnosti povprečne porabe energije od količine podatkov lahko vidimo na Slikah 5.4 za pisanje in 5.5 za branje. Tu vidimo potrditev teoretičnih pričakovanj, saj vsaka operacija pisanja potrebuje dva povezovalna dogodka, kar pomeni da večje število podatkov enostavno potrebuje večje število operacij, katere v povprečju porabijo enako količino energije.

Primarni načini spreminjanja pretočnosti v standardu Bluetooth Low Energy so spreminjanje parametrov *ATT_MTU*, povezovalnega intervala in hitrosti modulacije PHY nivoja. Ker je smiselno nastaviti največje parametre *ATT_MTU* ter PHY modulacije, ki jih naprave podpirajo, nam za spreminjanje pretočnosti ostane le še povezovalni interval. Slika 5.6 prikazuje spreminjanje pretočnosti ob različnih intervalih, kar je v skladu s pričakovanji - manjši kot je interval, več paketkov lahko pošljemo v eni sekundi, zato bo pretočnost višja. Vidimo lahko tudi, da višji *ATT_MTU* občutno poveča pretočnost v različicah v4.2 in 5. Operacija obveščanja po drug strani ni omejena z odgovori, tako da bi pričakovali, da bo imela dolžina povezovalnega intervala manjši vpliv in da bo z daljšimi povezovalnimi dogodki tudi večja pretočnost, saj bo oddajnik lahko konstanto prižgan. Slika 5.7 prikazuje rezultate pri operaciji obveščanja, kjer vidimo da nad neko dolžino intervala pretočnost prične padati. Do tega pride zaradi dejstva, da se v primeru neuspešnega prenosa paktetka, povezovalni dogodek zapre. Takrat se lahko pošiljanje nadaljuje šele ob naslednjem intervalu. To pomeni, da bo ob večjem intervalu tudi povprečen čas, ko je dogodek zaprt daljši, zaradi česar pretočnost pade. Ob primerjavi s Tabelo 5.1 lahko tudi vidimo da so dosežene hitrosti pretočnosti zelo blizu teoretičnim maksimumom za vsako različico standarda BLE.

Ker sta operaciji branja in pisanja omejeni z odgovori GATT serverja so rezultati obeh operacij praktično isti. Ob pogledu na Sliko 5.8 lahko vidimo da je uspelo razvijalcem standarda Bluetooth z vsako novo različico zmanjšati porabo moči za prenos podatkov glede na pretočnost. To jim je uspelo kljub temu, da različica BLE 5 deluje pri višji hitrosti. To omogoča dejstvo, da se pri višjih hitrosti podatki hitreje prenesejo, kar pomeni, da se lahko oddajnik prej izklopi. To je lepo razvidno tudi na Sliki 5.9, ki kaže skupno energijo potrebno za prenos določene količine podatkov v odvisnosti od pretočnosti. Vidimo da je najbolj energijsko učinkovit prenos pri najvišjih pretočnostih, pri čemer je potrebno poudariti, da je pri merjenju različice v4.1 količina prenesenih podatkov manjša, zaradi česar je ustrezno manjša tudi poraba energije.

Zaradi istih lastnosti je tudi analiza operacij obveščanja in pisanja brez potrdila združena. Ker ni potrebno čakati na odgovor druge naprave je možno opraviti več operacij v enem povezovalnem dogodku. Na Sliki 5.10 je lepo razvidno da različici v4.1 in v4.2 uporabljata isto hitrost modulacije, saj

Kaj misliš s "hitrostjo modulacije"Iz

Ni mi jasno, kako lahko iz najveje in najmanjše moi (na sliki so n

enaki

sta največja in najmanjša ~~poraba~~ moči~~,~~ praktično ~~isti~~, vendar pri različnih pretočnostih. Vidimo, da hitrejša modulacija standarda BLE 5 potrebuje več energije, vendar je zato tudi dosti večja pretočnost podatkov. Bolj zanimiva je Slika 5.11, kjer vidimo velik preskok skupne porabe energije za prenos iste količine podatkov z vsako različico standarda. Za oba preskoka zmanjšanja porabe energije je odgovorno dejstvo, da se lahko oddajnik hitreje izklopi, pri čemer je razlog pri različici v4.2 povečanje količine podatkov v posameznem paketku, pri različici 5 pa višja hitrost modulacije. Zanimivo je tudi, da je količina energije potrebne za prenos neke količine podatkov praktično neodvisna od pretočnosti. Ta informacija je zelo zanimiva, saj nam omogoči, da parametre povezave nastavimo glede na potrebe, brez posebne skrbi za porabo energije pri večjih količinah prenešenih podatkov.

Modulacija nima hitrost

VI Sklep

Nedavno predstavljena različica 5 standarda Bluetooth obljublja dvakratno hitrost, štirikraten doseg in osemkrat boljšo zmogljivost oddajanja, ki naj ne bi vplivala na porabo moči. V magistrski nalogi je predstavljeno delo, potrebno za primerjavo porabe energije med različicami standarda Bluetooth Low Energy.

Da smo premostili omejitve merilnega sistema smo morali najprej potrditi, da je povprečna poraba neodvisna od količine prenešenih podatkov. Nato smo testirali vpliv povezovalnega intervala na pretočnost, kjer smo ugotovili da je pretočnost najvišja ob najnižjem intervalu za operacije branja in pisanja. Za operacije obveščanja in pisanja brez potrdila je za najvišjo pretočnost potrebno najti ravnovesje med velikostjo intervala ter zasedenostjo radijskih valov v danem okolju. Z meritvami potrebne moči in porabe energije za prenos podatkov smo ugotovili, da se energetska učinkovitost standarda Bluetooth izboljšuje z vsako različico. Ugotovili smo, da je za najnižjo porabo energije v vseh primerih potrebno najti najvišjo pretočnost, saj se v tem primeru lahko oddajnik najhitreje izklopi, s tem pa se tudi zmanjša poraba energije.

Rezultati in ugotovitve predstavljene v tej magistrski nalogi dokazujejo, da ni razloga da nadaljnji razvoj naprav Bluetooth ne bi uporabljal različice Bluetooth 5, saj le-ta prinese višjo pretočnost, daljši doseg in boljše sobivanje z ostalimi brezžičnimi tehnologijami, in vse to zagotavlja pri nižji porabi energije za prenos podatkov.

Chapter 1

Introduction

In recent years, we have seen significant progress in digital technologies, that contribute to the ever-present concept of "Internet of Things". Since the vast majority of IoT products use Bluetooth for communication, it is still under development and is continuously improved. At work, we are working on the development of a new product that uses Bluetooth Low Energy for communication. During development we have encountered a problem, where we needed a ~~scientific~~ study, that would compare energy consumption in relation to throughput with different connection parameters applied and across different Bluetooth Low Energy versions. This is especially interesting for the new Bluetooth 5, which promises twice the speed, four times the range and eight times increased broadcast capacity, all of which is supposed not to affect power consumption [9].

To se zelo slabo sliši:

throughput

kaj je to?

1.1 Related Work

Comparing energy consumption when transferring data using different technologies is nothing new, but it is fascinating that we have not been able to find any such research that compares multiple versions of Bluetooth technology. It seems even more interesting because of the arrival of Bluetooth 5 [9], which brings many innovations that are incompatible with previous versions

different

Such comparision have become even more interesting with the arrival of Bluetooth 5, which without modifying the physical level of the device.

A comparison of the power consumption between Bluetooth 4.0 Low Energy and ZigBee / 802.15.4 can be found in an article [1], that upgrades older articles that compare Bluetooth Classic (2.0). The authors also test the effect of interference on Bluetooth and ZigBee technology. Similar articles [2, 3] where authors compare Bluetooth 4.0 Low Energy with some other technologies, such as ZigBee / 6LoWPAN via IEEE 802.15.4, IrDA, NFC, ANT and SimpliciTI, ~~are also available in different publications~~, but there is still nothing comparing BLE with itself. The technique of measuring the voltage drop through the shunt resistor was used for power consumption measurements. Current measurements using the shunt resistor technique are described in ~~multiple articles~~ [4, 5, 6].

supersedes

studies were maid

nov odstavek

different versions

Power consumption

1.2 Thesis content

In Chapter 2, we describe the history of Bluetooth and Bluetooth Low Energy, as well as general differences between the Bluetooth standard versions and when those improvements were introduced.

In Chapter 3, we describe the Bluetooth Low Energy stack, what the layers are responsible for and how they interact, especially the relevant layers and their properties and mechanisms.

are the responsibilities

The properties and mechanisms of the layers that have major imp

In Chapter 4, we describe the hardware and software tools used for power consumption measurements, as well as the tested device firmware implementation development. The testing firmware mechanisms are explained, ~~as well as~~ the test scenarios that were used as the basis for power consumption measurements.

and

are presented

In Chapter 5, we describe measurement examples and results for different tests. We analysed power consumption invariance from data size, above a certain limit, the throughput of BLE standards and power consumption comparison at different connection parameters with Bluetooth Low Energy versions.

We analyzed the relationship between energy consumption and amount of data transmitte

Finally, we conclude the thesis in Chapter 6, recapping the work done and present the most important findings.

Chapter 2

A brief history of Bluetooth

2.1 Version 1.0

The Bluetooth standard was initially conceived by Dr Jaap Haartsen at Ericsson back in 1994. It was named after the Danish king Harald Blatand, nicknamed Bluetooth, who united Denmark, Norway and Sweden in the 10th century. At the time Ericsson wanted to replace the wired standard RS-232 with a wireless alternative, while other companies including Intel and Nokia also had the idea of wirelessly linking cell phones and computers around the same time.

In 1998 the Bluetooth Special Interest Group (SIG) was formed with five companies: Ericsson, Nokia, Intel, Toshiba and IBM. The Bluetooth SIG still to this day publishes and promotes the Bluetooth standard and its subsequent revisions, and although it started as a group of five companies, it reached 4,000 members by the end of its first year. The group now contains over 33,000 member companies at various levels of influence.

Although Bluetooth uses a very similar frequency bandwidth (2.4 - 2.485 GHz) as Wi-Fi, it has been designed as a much shorter range and lower power alternative from the start. Version 1.0, released in 1999, proposed as a replacement for the RS-232 standard, was designed as a flexible packet-based protocol with a wide selection of profiles. The standard came with profiles for wireless voice and headsets, dial-up networking, fax and file transfers, which were the primary use cases for RS-232, but the set of profiles has increased substantially since then. The first version had some problems, such as anonymity issue due to compulsory address broadcasting, connection problems, slow (theoretical) data speeds of just 721kbps and short range of about 10 meters.

2.2 Version 2.0 + EDR

In 2002 IEEE approves 802.15.1 specification to conform with Bluetooth wireless technology and two years later, Version 2.0 [10] with Enhanced Data Rate (EDR) is released. The same year Bluetooth technology reaches an installed base of 250 million devices and product-shipment rate surpasses 3 million per week. Version 2.0 brings some much-needed improvements, such as increased range of around 30 meters, an increased theoretical connection speed of 1 Mbps ~~core~~ and 3 Mbps with EDR. Power consumption was also reduced by 50% in comparison to the previous version.

was

ce že, bi napisal:theoreti

2.3 Version 2.1 + EDR

In 2007 Bluetooth 2.1 [11] standard was released. Secure Simple Pairing (SSP) was the most important feature of this version, which made the pairing process more secure and simpler. SSP made encryption mandatory for all connection, which made man-in-the-middle attacks more difficult. Another important improvement brought in version 2.1 was sniff subrating, which improved battery life in devices that are inactive most of the time, such as keyboards and headsets, by reducing the active duty cycle of Bluetooth devices.

2.4 WiBree acquisition

In June 2007 the Bluetooth SIG acquired the Wibree Alliance, an initiative led by Nokia (together with Nordic Semiconductor, Broadcom Corporation, CSR and Epson) to develop an ultra-low power form of wireless connectivity, which uses much less power than existing Bluetooth technology. Nordic and other brought their knowledge of ultra-low power wireless connectivity to the Bluetooth SIG, enabling it to develop what will become Bluetooth Low Energy (BLE).

2.5 Version 3.0 + HS

Version 3.0 High Speed (HS) [12] specification was released in 2009, with the key feature being high-speed data transfer. In conjuction with 802.11 Wi-Fi radio, it could reach data speeds of up to 24 Mbps. It used the 802.11 link for transferring large amounts of data while still using Bluetooth radio for discovery, connection and configuration, which made it less power hungry than classic 802.11 Wi-Fi link. This version also permitted near-field communication (NFC) for pairing.

2.6 Version 4.0 + LE

The Bluetooth SIG announces the formal adoption of Bluetooth Core Specification Version 4.0 with Low Energy [13] technology in 2010. This version defines two chips, one for Classic Bluetooth v4.0 (BR/EDR) and one for Bluetooth Low Energy (also called Smart Bluetooth). The Bluetooth Low Energy technology was heavily based on WiBree. Being adopted and finished by Bluetooth SIG made it possible for BLE to complement the existing Classic Bluetooth, while still making it possible to run it off a single coin cell battery. Bluetooth LE was mainly designed to frequently transmit small chunks of data to and from small devices while saving power. That made possible a new way to gather data from various sensors, which enabled the development of heart

Adoptation of the Blueto

empower

rate monitors, thermometers, health and fitness trackers, etc., which all took advantage of the new low power feature.

Aside from Bluetooth LE feature, this version also introduced the Security Manager (SM) services with AES encryption and the Generic Attribute Profile (explained in Chapter 3.5), which is the backbone of a Bluetooth LE device, as it provides the profile of the device.

2.7 Version 4.1 LE

Version 4.1 [14], introduced in 2013 brought very few changes in hardware, speed and range, but many improvements on the software side. Bluetooth 4.1 focused on implementing the groundwork to drive IoT devices. It could indirectly connect devices to the cloud that would have otherwise been out of range. Another improvement was coexistence with LTE radios, as previous versions had problems such as poor performance and battery draining when used simultaneously. This version also allowed developers to have more control over connections, making it easier to manage their power as well as the power of a connected device. Additionally, Version 4.1 made it possible to treat any device as both a peripheral and a hub at the same time, enabling the direct connection between peripheral devices without an intermediary.

2.8 Version 4.2

Version 4.2 [15] was released in 2014, and it brought multiple improvements and is still the most widely used standard today. The most notable improvement was the LE Data Length Extension (DLE), which increased the capacity of each packet from 20 bytes to 244 bytes. It also enabled devices to use Internet Protocol version 6 (IPv6) for direct internet access, which allowed sensor and smart devices transmitting data directly over the internet. Another improvement was LE Secure Connections, which made it harder to track devices without permission.

2.9 Version 5

Bluetooth version 5 [16] was officially adopted by Bluetooth SIG in 2016, with most of the major smartphone manufacturers adopting it in 2017 and 2018. Its purpose is to optimise wireless technology by increasing the functionality of Bluetooth. Aside from improvements involving IoT, Bluetooth 5 promises two times faster transfer speed, four times longer range (up to 120 meters) and eight times the data broadcasting capacity. It also improves coexistence with other wireless technologies, which is becoming more and more critical as we enter a complex world of IoT smart devices. Bluetooth SIG dropped the point number of the name, so it is now just called Bluetooth 5 (instead of Bluetooth 5.0 or 5.0 LE), which is supposed to "simplifying marketing, and communicating user benefits more effectively" [17].

Chapter 3

Bluetooth Low Energy

Bluetooth Low Energy (BLE), also known as Bluetooth Smart, was introduced in Bluetooth Core Specification Version 4.0 [13] as a light-weight, ultra-low power subset of classic Bluetooth. While there is some functionality overlap with classic Bluetooth and they share the name ”Bluetooth”, BLE has an entirely different property and functionality set. BLE started its life in 2001, in Nokia labs as an in-house project named WiBree, before it was adopted by Bluetooth SIG and renamed to Bluetooth Low Energy.

This chapter will give a quick overview of BLE, specifically how data is organised in BLE and how devices advertise their presence so that other devices can find and connect to them.

3.1 BLE Stack

The Bluetooth Low Energy stack is split into three main layers, Controller, Host, Application, which in turn consist of multiple parts, as seen in Figure 3.1. In most cases the three layers are implemented on a single SoC, but they can also be distributed onto multiple physical SoCs, with most commonly split between the Host and the Controller [18]. For us, the key elements of the BLE stack are the Physical Layer (PHY), the Generic Access Profile (GAP) and the Generic Attribute Profile (GATT).

and

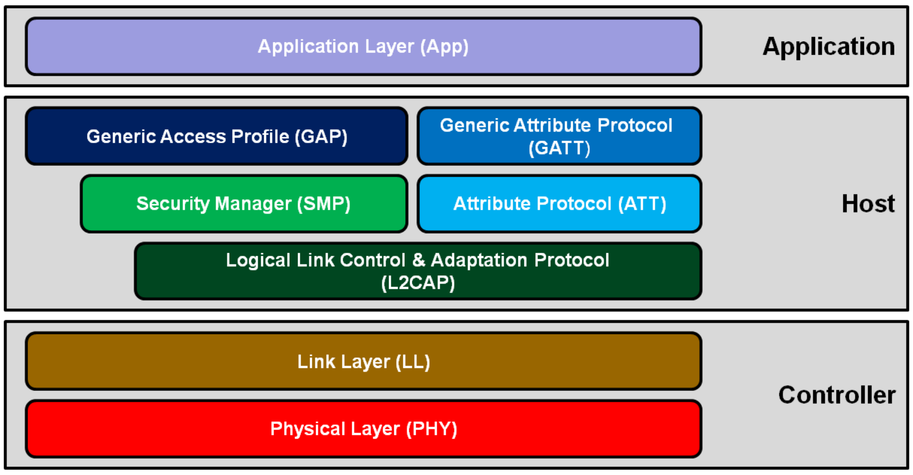


Figure 3.1: Bluetooth Low Energy stack[1].

3.2 Physical Layer (PHY)

The BLE physical layer is the lowest layer of the BLE stack and contains the analogue circuitry responsible for translation, transmission and receiving of digital symbols over the air. BLE radio operates in the 2.4GHz ISM (Industrial Scientific Medical) band. A frequency hopping transceiver is used to combat interference and fading. The BLE system uses 40 RF channels between 2400 MHz and 2483.5 MHz with 2MHz spacing. The 40 channels are divided into 3 Advertising Channels and 37 Data Channels [19].

BLE v4.0, v4.1 and v4.2 used a radio, transmitting at 1 Mbps with one symbol per bit. With the BLE 5 Core Specification [16] Bluetooth SIG added two new modes: High Speed (HS) and Long Range (LR). The BLE 5 LR (also known as coded) uses the same 1 Msym/s physical modulation scheme, but each bit is encoded at either 125 kbps (S=8) or 500 kbps (S=2). That is because Bluetooth 5 LR uses Forward Error Correction (FEC), which replaces a single bit with multiple symbols before modulating and sending, as shown in Table 3.1.

bit

[1]Image via Microchip: `http://microchipdeveloper.com/wireless:ble-introduction`

to isto ne drži, nekje sem prebral, da se 10 bitov zapiše s 15-timi (FEC), potem pa se še pošlje e

Input (from FEC encoder)	Output with S=2	Output with S=8
0	0	0011
1	1	1100

Table 3.1: FEC Pattern Mapper behaviour

It means that the "2x speed and 4x range" advertised by Bluetooth SIG for Bluetooth 5 actually means 2x speed OR 4x range, as those two options are on opposite sides of the spectrum, meaning we cannot have both at the same time.

3.3 Link Layer (LL)

The Link Layer is the part that ties together the PHY and the Host stack. It is usually implemented as a combination of custom hardware and software. Because it is responsible for managing all timing requirements, it is usually kept isolated from the higher layers of the stack, as it is the only hard real-time constrained layer of the whole protocol stack. It is responsible for advertising, scanning, and creating/maintaining connections. At the link layer, the following role pairs are defined:

- Advertiser / Scanner (Initiator)
- Slave / Master
- Broadcaster / Observer

Once the Scanner has acquired enough information about the Advertisers in its range to decide to which to connect it, it becomes an Initiator, initiating a BLE Link Layer connection process. A connection is simply a sequence of packet transmissions between two BLE devices at predefined times. Once connected, the Link Layer acts as a reliable data barrier, meaning any data presented to the upper layers have been reliably transmitted. All packets are checked against a 24-bit CRC. If the check fails, re-transmissions are requested by the Link Layer until the packet is transmitted successfully, with the retries being done in the next connection event and channel.

Prvi stavek mi ni vsec, ker ni natancen (mogoce nekaj takega):Once connected, the data packet exchange (ali tran

starts

duration of the

Once connected, the Master/Slave exchange data packets at regular intervals, called *connection events*, as seen in Figure 3.2. The *Connection Interval* can be set between 7.5 ms and 4000 ms with a step ~~size~~ of 1.25 ms. When ~~in~~ a connection, the master has to transmit a packet to the slave once every connection event. If there is no actual data to send, a 0-byte packet is sent to keep the connection alive. A connection event is the start of a group of data packets ~~that are sent~~ from the master to the slave and back again. The connection event continues until a packet either fails to be received correctly, the sender is satisfied with ending the connection event, or, in an improbable scenario, the end of the interval has been reached. If a packet fails to be received correctly, the connection event is ended, and the failed packet will be retried at the next connection event.

a connection is e

transmittion of a

prej si govoril o enem pa

Connection Events and Connection Intervals are the concepts that enable the Bluetooth Low Energy to actually be Low Energy. That is because between CIs, the radios can be turned off, which saves tremendous amounts of energy. The most power-hungry part of a BLE chip is undoubtedly the 2.4GHz oscillator needed for the radio, and if we can keep the radio turned off as much as we can, the energy consumption will be lowered substantially. The CEs/CIs are thus used to turn on the radio only at predetermined intervals for short amounts of time, keeping the average power consumption low.

[2] Image via Microchip: `https://microchip.wdfiles.com/local--files/wireless:ble-link-layer-connections/connected-phase.png`

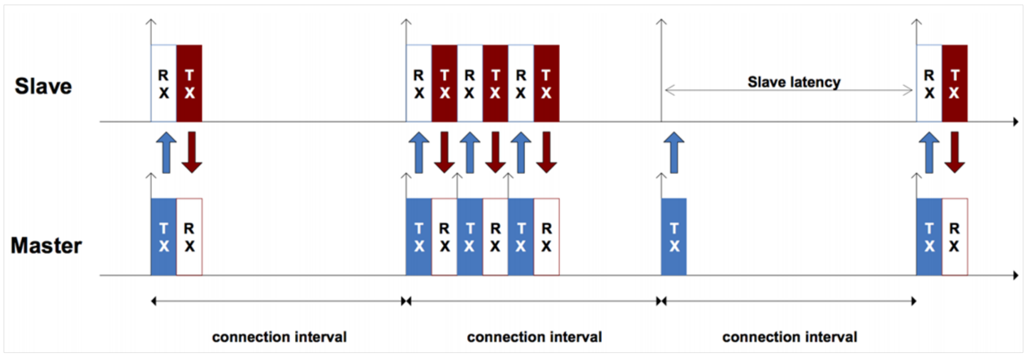


Figure 3.2: Connection events and connection intervals[2].

3.3.2 Connection parameters

Other than the *connection interval*, explained in Chapter 3.3.1, there are two more connection parameters managed by LL: *Slave latency*, which declares the number of connection events that a slave can choose to skip without risking disconnection, and *Connection supervision timeout*, which is the maximum time between two received valid data packets before a connection is considered lost.

The connection parameters are negotiated at the initial connection procedure, but they can also be changed during the active connection. This can be used to dynamically increase throughput for transfer of larger chunks of data, while still maintaining low power consumption on average. This can be achieved by decreasing the connection interval for the data transfer and increasing it back up after the data was sent. That way, each connection can be fine-tuned to provide the best balance between throughput and power consumption.

3.4 Generic Access Profile (GAP)

Generic Access Profile (GAP) covers the usage mode of the lower-level radio protocols to define roles, procedures and modes that allow devices to broadcast data, discover devices, manage connections and negotiate security levels. This is a Bluetooth base profile which is mandatory for all Bluetooth devices.

3.4.1 Roles

In BLE, GAP defines four specific roles: Broadcaster, Observer, Peripheral, and Central. A device can operate in one or more BLE GAP roles at a time.

Broadcaster role is optimised for transmit-only applications that distribute data regularly. Broadcasters send data in advertising packets instead of in data packets, so that data is immediately accessible to any device listening, without the need for a specific connection to the Broadcaster.

The Observer is optimised for receive-only applications that want to collect data from broadcasting devices, without the need to form a connection with them. An Observer listens to data embedded in the advertising packets.

Central role corresponds to the Link Layer master and supports multiple connections. It is the initiator and controller of connections to peripheral devices. The device supporting the central role needs access to more processing and memory resources to be able to manage multiple connections and generally supports more complex functions compared to other BLE GAP devices. The central starts by scanning for other devices advertising packets and then initiates a connection with the selected device.

Peripheral role corresponds to the Link Layer slave. It is optimised for devices that support a single connection and are less complex than central devices. The peripheral device starts with advertisements to allow the central devices to find and establish a connection with it.

It is common for developers to associate BLE GATT *client* and *server* roles (explained in Chapter 3.5) with GAP *central* and *peripheral* roles, even though there is no connection between them, as any combination is allowed and used in different devices and use cases.

3.4.2 Modes and procedures

GAP layer is responsible for the following procedures, which are applicable to specific roles [16]: broadcasting, observing, advertising, discovery, connection establishment, connection parameter update, connection termination and many more.

Advertising packets are blindly sent at fixed intervals unidirectionally, and they make up the basis for broadcasting (and observing), as well as discovery. An observing or scanning device may receive the advertising packet if it happens to scan while the advertising packet is transmitted, after which it can initiate the connection. A connection requires two devices that synchronously perform data exchanges at set regular intervals and provide guarantees on data transmission.

3.5 Generic Attribute Profile (GATT)

Generic Attribute Profile (GATT) defines the way BLE devices transfer data between themselves, using concepts called *services* and *characteristics*. GATT defines how to use Attribute Protocol (ATT) as its transport protocol to discover, read, write, broadcast and obtain indications of data. It is used by the application for data communication between two connected devices. The data is organised hierarchically in groups called *services*, which group conceptually related pieces of data called *characteristics*.

where group consists of c

Attributes are the smallest data item defined by GATT (and ATT). They are addressable pieces of information that can contain user data or metadata. Metadata houses informations about the structure and grouping of the different attributes contained within the server.

Dobro bi bilo opisati relac

GATT also provides the reference framework for all GATT-based profiles, which are defined by Bluetooth SIG. These profiles cover precise use cases and ensure interoperability between devices from different manufacturers and range from mandatory ones, such as *Device Name*, to more use case specific such as *Heart Rate Measurement*.

3.5.1 Roles

As with any other profile or protocol in Bluetooth specification, GATT starts with defining the roles that devices can implement:

The GATT Server contains the data to be monitored or changed, organised in *attributes*. It receives requests from a client and sends back responses. It can also send server-initiated updates when configured to do so.

The GATT Client sends requests to a server and receives responses (and server-initiated updates) from it. Because the GATT client inherently has no information about the server's attributes, it must first inquire about those attributes by performing service and characteristic discovery. After completing service and characteristic discovery, it can then start reading and writing attributes (organised in services and characteristics) supported by the server.

While the most common configuration is a smaller device configured as GAP Peripheral and GATT Server, and a second device (most often a smartphone) configured as GAP Central and GATT Client, it is by no means mandatory, as already mentioned in Chapter 3.4.1.

3.5.2 UUIDs

] A universally unique identifier (UUID) is a globally unique 128-bit (16-byte) number that is used to identify Profiles, Services and Data Types in a Generic Attribute (GATT) Profile. BLE specification adds support for shortened 16-bit and 32-bit UUIDs. These shortened formats can only be used with UUIDs that are defined by the BLE specification (i.e., that are listed by the Bluetooth SIG as standard Bluetooth UUIDs), while vendor-specific UUIDs always use 128-bit.

3.5.3 Attribute and Data Hierarchy

The data in a GATT server is grouped into *services*, each of which can contain zero or more *characteristics*, as seen in Figure 3.3. Each service distinguishes itself from other services by a UUID. This hierarchy is strictly enforced for any device claiming GATT compatibility (essentially, all BLE devices sold).

Characteristics can be thought of as containers for user data. They always include at least two attributes: the characteristic declaration (metadata) and the characteristic value (user data). They can also contain additional attributes called *descriptors*, which expand the metadata contained in the characteristic declaration. The most common descriptor used is the *Client Characteristic Configuration Descriptor (CCCD)*, which provides a mechanism for Server-Initiated updates (explained in Chapter 3.5.4).

Attributes are the smallest addressable data entity defined by GATT (and ATT). The most common attributes are:

ali je attribut handle (type,

- The attribute **handle** is a unique 16-bit identifier for each attribute on a particular GATT server.
- The attribute **type** is nothing other than a UUID (see Chapter 3.5.2).
- **Permissions** are metadata that specify which operations can be executed on each particular attribute. ATT and GATT define the following permissions:
  - Access Permissions: *None, Readable, Writeable, Readable and Writeable.*
  - Encryption: *No encryption, Unauthenticated encryption, Authenticated encryption.*
  - Authorization: *No authorization required, Authorization required.*
- **Value** holds the actual data content.

3.5.4 Features

GATT features are strictly defined procedures that allow GATT-based data exchanges to take place. There are 11 features defined in the GATT Profile: *Server Configuration*, *Primary Service Discovery*, *Relationship Discovery*,

[3]Image via Adafruit: `https://learn.adafruit.com/assets/13828`

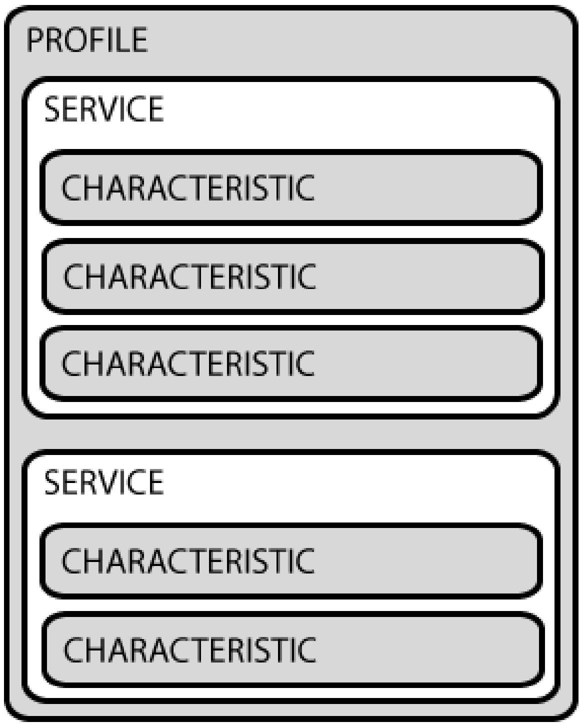


Figure 3.3: GATT data hierarchy[3].

Characteristic Discovery, *Characteristic Descriptor Discovery*, *Reading a Characteristic Value*, *Writing a Characteristic Value*, *Notification of a Characteristic Value*, *Indication of a Characteristic Value*, *Reading a Characteristic Descriptor* and *Writing a Characteristic Descriptor*. Here, the features are relevant for this work, will be explained in more details.

Server Configuration: Exchange MTU

This (sub-)procedure is used by the client to set the *ATT_MTU* parameter (see Chapter 3.5.5) to the maximum value that is supported by both devices. The client will send an *Exchange MTU request*, to which the server will reply with either *Exchange MTU Response* or *Error Response*. The *ATT_MTU* parameter defines the largest data payload allowed by the ATT layer, which means that the maximum data payload for GATT operations is $ATT\_MTU - 3$ bytes, as 3 bytes are used for the header.

Service and Characteristic Discovery

Because the client by itself has no knowledge about the server attributes, services and characteristics themselves and their handles it usually issues a series of sub-procedures to gather that information when it first connects to the server. These sub-procedures return the list of primary services and their handles, which can then be used to issue characteristic discovery for each returned service.

Characteristic Value Read

Tole je malce cudno, ker j

This procedure is used by the client to read a Characteristic Value from a server. Four sub-procedures can be used to read a Characteristic Value: *Read Characteristic Value*, *Read Using Characteristic UUID*, *Read Long Characteristic Values*, and *Read Multiple Characteristic Values*.

To je zavajujoce, najprej s

→ The *Read Response* to the *Read Characteristic Value* contains the complete Characteristic Value if that is less than $ATT\_MTU - 3$ bytes ~~in length~~.

its length (mogoce bolje s polnim nazivom: the length of the characteristic value) functionprocedu

Otherwise, it only contains the first part, and the *Read Long Characteristic Value* can be used to obtain the rest of the value.

The *Read Long Characteristic Values* sub-procedure request is issued with increasing offset as a request parameter until the reply is shorter than $ATT\_MTU - 3$ bytes.

Characteristic Value Write

This procedure is used by the client to write a Characteristic Value to a server. Five sub-procedures can be used to write a Characteristic Value: *Write Without Response*, *Signed Write Without Response*, *Write Characteristic Value*, *Write Long Characteristic Values* and *Reliable Writes*.

Write Characteristic Value can be used by the client to write a Characteristic Value if it is smaller than $ATT\_MTU - 3$ bytes. ~~and~~ the server will acknowledge the operation with a response.

The server acknowledge

When the value ~~to be written~~ is larger than $ATT\_MTU - 3$ bytes, the client can use *Write Long Characteristic Values*. Here the data (up to 512 bytes long) is split into multiple *Prepare Write Requests*, each of which can have at most $ATT\_MTU - 5$ bytes in its payload, as two bytes are needed to transfer the offset. Each *Prepare Write Request* will get a *Prepare Write Response* from the server, which sends the offset, as well as data for the client to confirm. After all the data is transferred, an *Execute Write Request* is used to write the complete value.

Ali so:Prepare Write Requ

Reliable Writes sub-procedure is used when the client wants to write multiple characteristic values, so the operations are queued and at the end, a final packet to commit the pending write operations and execute them is sent.

Ali koncen paket dodas v v

Notification-e se

Write Without Response is the equivalent to notifications, except that here the data stream is from client to server. This uses Write Command packets, which are not acknowledged at the application layer, only by the link layer, which means that the client will get the acknowledgement that the packet was successfully transferred to the server, but not if the server

This write sub-procedure does not trigger the server acknowledgement at the GATT layer, however the link layer still accepted the packet. The data payload limit of *ATT_MTU* − 3 bytes applies to *Write Without Response* as well.

The *Signed Write Without Response* uses the same mechanism as *Write Without Response*, with the difference being it is only used if the authenticated bit is enabled and the client and server share a bond and that the data payload is *ATT_MTU* − 15 bytes.

Characteristic Value Notification

This procedure is used when a server is configured to notify a Characteristic Value to a client without expecting any acknowledgement that the notification was successfully received. Notifications can be configured using the Client Characteristic Configuration descriptor.

Characteristic Value Indication

This procedure is used when a server is configured to indicate a Characteristic Value to a client and expects an acknowledgement that the indication was successfully received. Indications can be configured using the Client Characteristic Configuration descriptor.

3.5.5 BLE Packet

A BLE packet containing application data and transferred over the air has the structure seen in Figure 3.4. The size of the Data field is dependent on the Bluetooth specification version. In Bluetooth 4.0 and 4.1, the maximum size of the Data field was 27 bytes(+4 bytes for the message integrity check). Bluetooth 4.2 brought the new feature called *Data Length Extension (DLE)*, which enabled the Data field size to be up to 251 bytes (+4 bytes for the message integrity check).

maximumlargest

[4]Image via PunchThrough: https://punchthrough.com/blog/posts/maximizing-ble-throughput-part-2-use-larger-att-mtu

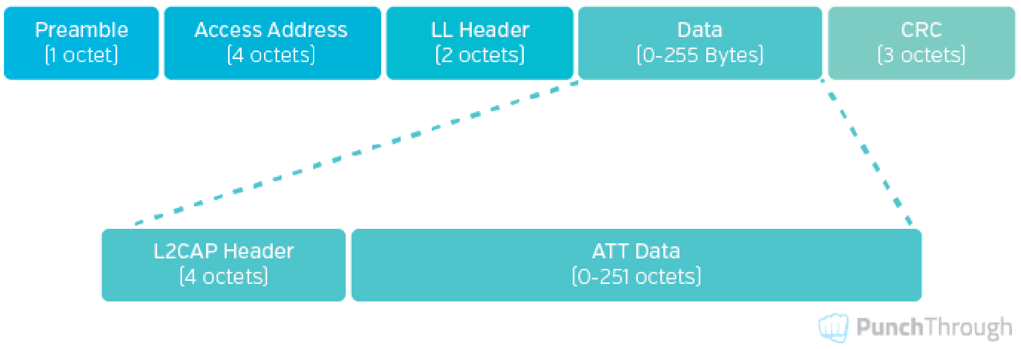


Figure 3.4: BLE packet structure[4].

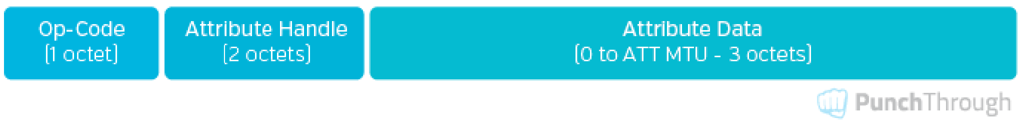


Figure 3.5: ATT packet structure[5].

That Data field is an L2CAP packet, which includes 4 bytes for the L2CAP header, yielding 23 (BLE v4.0, v4.1) or 247 (BLE v4.2, v5) bytes of ATT Data, which is also known as *ATT_MTU*. The ATT packet includes 1 byte for the ATT operation code and another 2 bytes for the attribute handle, as seen in Figure 3.5, from which we get the maximum for the application layer data payload of 20 bytes in BLE v4.0 and v4.1, and 244 bytes for BLE v4.2 onwards.

po sliki je 251!!!preve

[5] Image via PunchThrough: `https://punchthrough.com/blog/posts/maximizing-ble-throughput-part-2-use-larger-att-mtu`

Chapter 4

Testing methodology

4.1 Hardware

Testing was done using two Nordic *nRF52840 DK* [7] development kit boards, one as the central device and the other as the peripheral. The *nRF52840 DK* is a single board development kit for Bluetooth development on *nRF52840 SoC* [20], with support for the on-board SEGGER J-Link OB Program/Debug probe [21]. The *nRF52840 SoC* is designed around an ARM Cortex-M4 CPU and has a 1MB flash with cache and 256kB RAM. It has the full hardware and software support for all the new Bluetooth 5 features: Long Range (Coded) and High-Speed PHY layers, Advertising extensions and improved coexistence. The Nordic protocol stacks are known as *SoftDevices* [22]. The nRF52840 is supported by the S140 SoftDevice, which is a 20-link Bluetooth 5 pre-qualified Bluetooth Low Energy protocol stack.

The central device was the test initiator and controller and was connected to a PC for test monitoring using the on-board SEGGER J-Link Real Time Transfer technology [23]. The peripheral device was the device under test and was configured in *nRF only mode*, where the *nRF52840 SoC* under test is decoupled from peripheral devices such as LEDs, interface MCU and external memory using analogue switches. It was then powered with 3.000V using the *external supply* connector on the board from the measurement system.

to implicira toleranco vira < +- 0.005, kar

Range	Resolution	Accuracy [1]	Offset
1-70 µA	0.2 µA	±20%	±2µA
70 µA-1 mA	3 µA	±15%	±2µA
1-70 mA	50 µA	±15%	±2µA

Table 4.1: Power Profiler measurement resolution and accuracy.

Bluetooth Low Energy is backwards compatible down to version 4.0, which is why a single chip (*nRF52840 SoC*) was used for all the tests. Tests for versions BLE v4.0/v4.1 and BLE v4.2 were done with the same hardware as tests for BLE 5, with parameters set to that specific version capabilities, as described in Chapter 4.5.

exhibit

4.2 Measuring system

For measurements, the *Power Profiler Kit* [8] from Nordic Semiconductor was used. The Power Profiler Kit is a current measurement tool for embedded development, specifically optimised for measuring the power consumption of Bluetooth devices. It has a three-stage analogue measurement circuit, which splits the total current range into three ranges: 1 µA to 70 µA, 1 µA to 1 mA and 1 mA to 70 mA. It can be used to power external boards that are being measured with Vcc level configurable between 1.8 V and 3.6 V with up to 70 mA.

The power supply voltage is

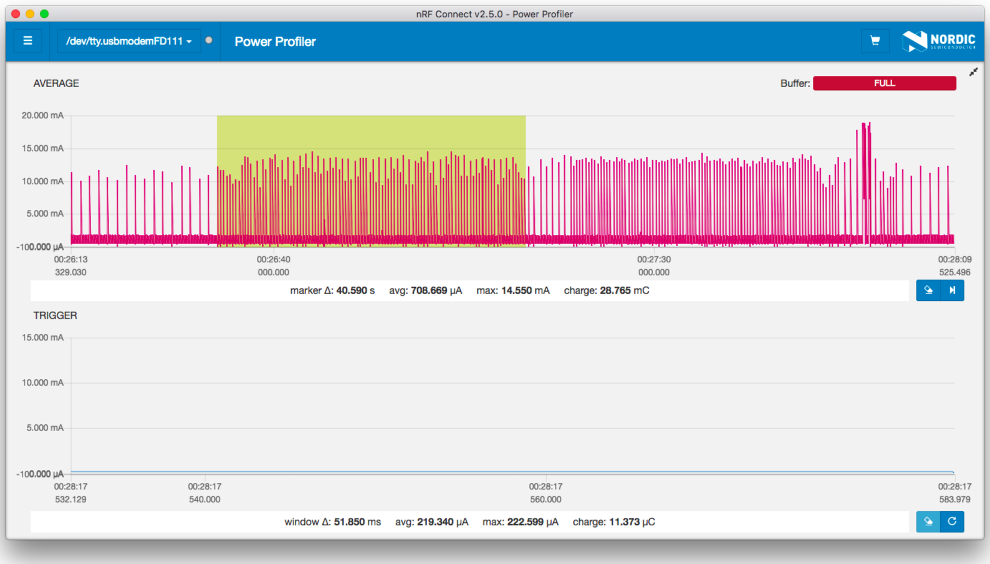
, and it can drive the current of up to 70mA cu

Table 4.1 shows the measurement resolution and accuracy of the Power Profiler Kit. It should also be noted that the measurement frequency is 77 kHz, which nets the time resolution of 13 µs and that the maximum recording / averaging time in the software is 120 s. Although ~~we can see that~~ the accuracy of the Power Profiler Kit is not the greatest, it is sufficient for our needs. That is because of the nature of the tests and the fact that they were all conducted in the same environment and under the same conditions.

for low power measurem

[1] Accuracy of average readout

That is because we are interested in power consumption during the data transfer when I



marker Δ: **40.590** s avg: **708.669** µA max: **14.550** mA charge: **28.765** mC

window Δ: **51.850** ms avg: **219.340** µA max: **222.599** µA charge: **11.373** µC

Figure 4.1: Power Profiler software with test data selected.

No drift of the baseline (radio off) consumption was detected during testing, which means that comparisons can still be made. Tests were also conducted multiple times and averaged to get the results. more accuratemore reliable

The interface MCU of the child development kit is used for data connection to the Power Profiler Kit so that the data can be displayed in the PC software. The measurement data is displayed in dual measurement windows: one with longer, averaged acquisition times and one with high time resolution to show trigger events (if configured). A subsection of data can be selected for average current and energy calculations, as seen in Figure 4.1.

and the acquired

4.3 Hardware configuration

was used as device under test

As explained in Chapter 4.2 a nRF52840 DK board is needed for Power Profiler Kit operation, which can also measure the consumption of the nRF52840 SoC on the DK board. It has been decided to separate the measurement system and the device under test. This was done to avoid any interference

can be configured to

power

Tole mi zdaj ni cisto jasno: ali lahko nRF52840 DK meri svojo porabo? V tem primeru so gornji popravki narobe

Na sliki imas akronim PPK. Dobro bi bilo, ce bi v tekstu to opisal (mogoce znotraj enega ze napisanega stav

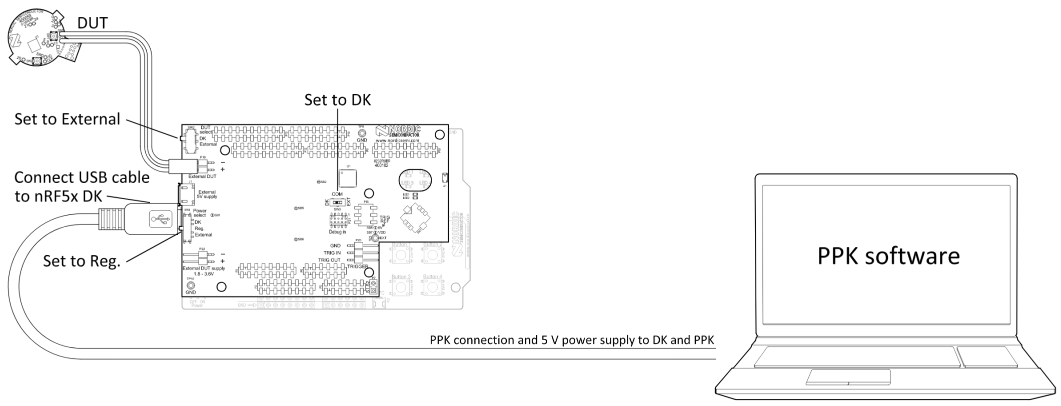

DUT
Set to External
Set to DK
Connect USB cable to nRF5x DK
Set to Reg.
PPK software
PPK connection and 5 V power supply to DK and PPK

Figure 4.2: Power Profiler connected to the external device under test (DUT)[2].

that the Power Profiler Kit and the interface MCU might present to the current measurement, as well as the Bluetooth connection. The connection can be seen in Figure 4.2, with the DUT being our peripheral device, connected to the external supply connector.

The actual hardware connections can be seen in Figure 4.3, where a small resistor can be observed between pin 13 and ground. The resistor is 470 Ω 1%, measured at 473 Ω. When the voltage is applied to pin 13, this resistor will increase the constant consumption by ~5.5 mA, as per the Ohm's Law:

Note that on DUT board a

$$I = \frac{V}{R} = \frac{V_{OH}}{R} = \frac{V_{dd} - 0.4\,\text{V}}{R} = \frac{3.0\,\text{V} - 0.4\,\text{V}}{473\,\Omega} = 5.5\,\text{mA} \qquad (4.1)$$

, where V_{OH} is output high voltage, which is equal to $V_{dd} - 0.4\,\text{V}$ [20]. This is used as an anchoring mean when conducting tests to roughly mark the start and end of the test. While the peripheral device is waiting for commands, the pin 13 is kept high, allowing the current to flow through the resistor and thus increasing the consumption. Once we get the *test start* command, pin 13 is kept low until the test is done, to not interfere with the test measurements,

[2]Image via Nordic: `http://infocenter.nordicsemi.com/topic/com.nordic.infocenter.tools/dita/tools/power_profiler_kit/PPK_user_guide_PPK_on_customHW_with_nRF5xDK.html?cp=5_6_5_4`

Vrednost upora in racune bi dal proti koncu

Tu bi dal vrednost upora in gornje racune, m

but still generate an edge, visible on the measurement software. An example of how the edge looks in the measurement software can be seen in Figure 5.1, on the very left side of the measurements, where the consumption falls from ~5 mA to ~0.6 mA.

Figure 4.4 shows the devices while testing and measurements are undergoing. The peripheral device under test is flipped on top of the central device. That is done to minimise the effect of interference on tests, as the signal to noise ratio will be best if devices are close.

4.4 Firmware implementation

A highly modular system was built specifically for the purpose of these tests. The projects for the central and the peripheral device are separated, but they re-used a lot of lower layer abstraction layers, which have been built on top of *Nordic BLE Stack Components* to provide an even higher abstraction to the topmost layers. The abstraction stack architecture can be observed in Figure 4.5.

The layers and their functions are:

- **Central / Peripheral Core** is the module that contains the testing logic. This is implemented using a state machine, which initialises the device on boot and awaits for further instructions.

- **Test Params** is the module, shared between central and peripheral devices by the use of *git submodule* mechanism [24] to keep it up to date in both projects. It is used to load the test parameters for different Bluetooth versions, as well as data building and checking.

- **Central / Peripheral BLE** is the module that maps the internal handles to UUIDs, as well as forwarding relevant Bluetooth events to the Core layer. On the central device, it is also responsible for scanning, connecting and service/characteristic discovery procedures.

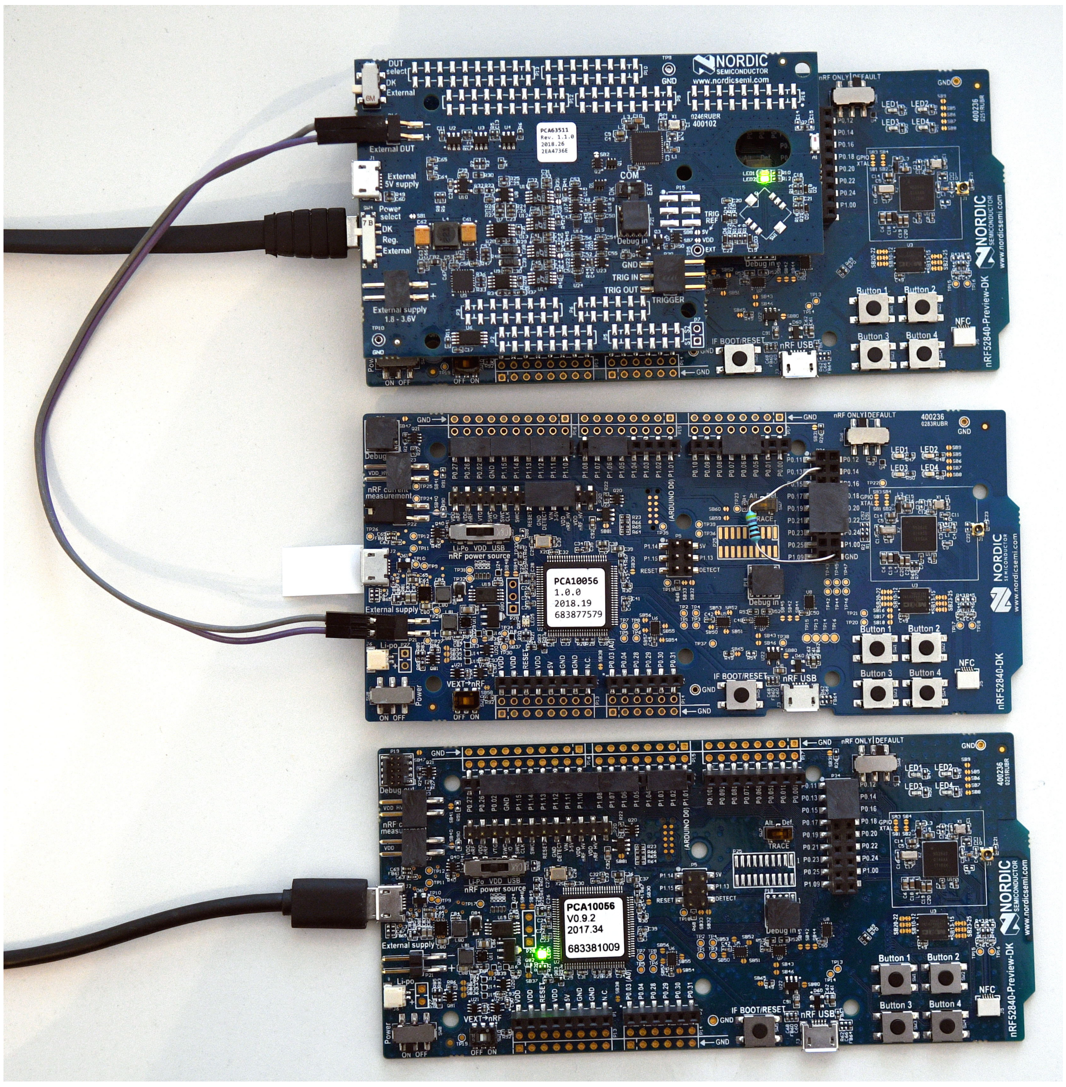


Figure 4.3: Hardware connections. Top board: Power Profiler on nRF5240 DK, the middle board: peripheral device under test, the bottom board: central device.

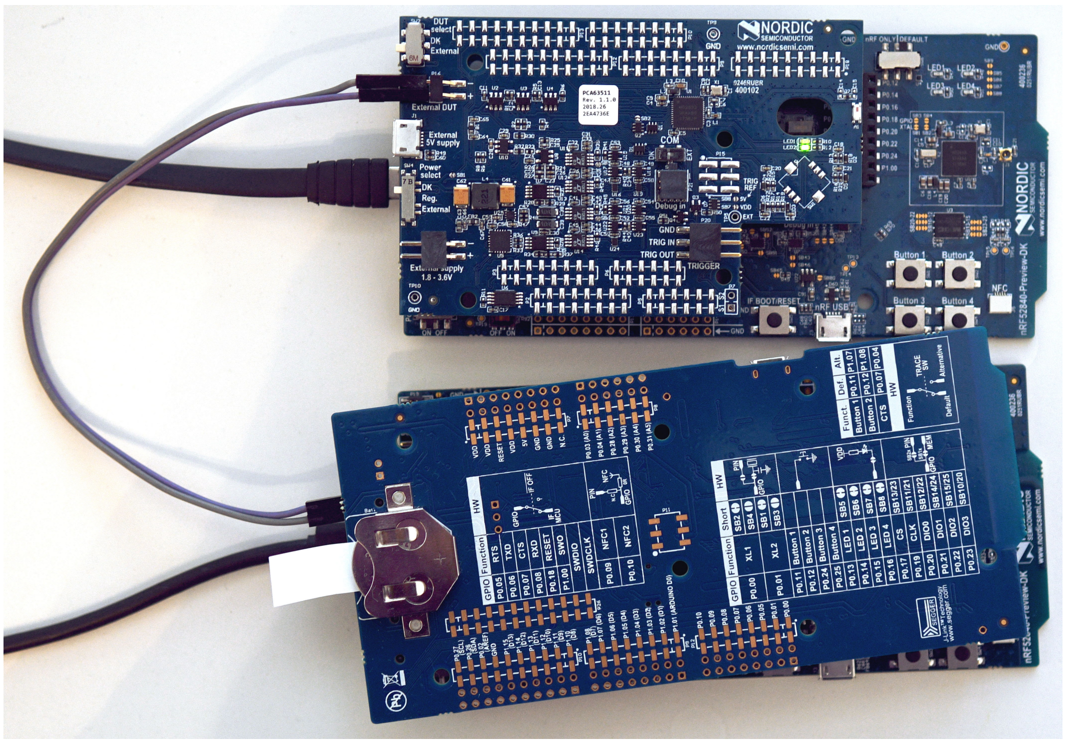


Figure 4.4: Hardware connections during testing. Top boards: Power Profiler on nRF5240 DK, bottom boards: peripheral device under test flipped on top of the central device.

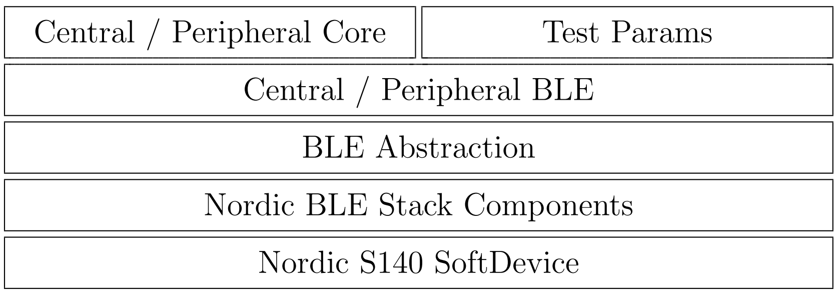


Figure 4.5: Firmware abstraction stack architecture.

- **BLE Abstraction** provides a layer of abstraction over the different Nordic BLE Stack Components.
- **Nordic BLE Stack Components** are the Host part components abstractions of the BLE Stack, provided by Nordic Semiconductor. These include GATT, GAP, L2CAP and other BLE layers and modules, as well as hardware abstraction layer.
- **Nordic S140 SoftDevice** is a pre-compiled binary image and functionally verified according to the wireless protocol specification, revealing only its Application Programming Interface (API), which is abstracting the actual procedures within.

The Nordic SDK version used was 13.1.0, with SoftDevice S140 v5.0.0-2.alpha, as that was the latest SDK and SoftDevice combo that Nordic offered at the start of programming. Because SDK version 14.x.x was still using the same SoftDevice and did not offer any improvements related to testing I we decided not to migrate to the newer SDK. With SDK version 15.0.0 came SoftDevice S140 v6.0.0, which is no production tested for nRF52840 SoC and brought support for LE Coded PHY S=2 (500 kbps). At that point, the migration would have taken too much time, and the implementation of LE Coded PHY S=2 was of no direct interest for these tests, so it has been decided against it. The same goes for SDK v15.2.0 and SoftDevice S140 v6.1.x.

Jaz tega ne bi omenj

The peripheral device implemented one proprietary Service, with two proprietary Characteristics, as seen in Figure 4.6. The test control and test data streams were split to ensure better control over tests. The first byte of the control characteristic was the control command enumeration, while the rest of the bytes were command dependent.

4.4.1 Central Core

The *Central Core* is used on the central device. After initialisation, it starts the scanning procedure, waits for the connection to be made and

Test Service UUID: F001800-F001-F001-F001-F001F001F001	
Test control characteristic UUID: F0010000-F001-F001-F001-F001F001F001	
Value	read, write
Test data characteristic UUID: F0010001-F001-F001-F001-F001F001F001	
Value	read, write, notify
Descriptor: CCCD	read, write

Figure 4.6: Peripheral device BLE API.

service/characteristic discovery to finish, after which it tests reads, writes and waits for a notification from the peripheral. After the self-tests are finished, it waits for the user to start the tests by pressing one of the hardware buttons. On button press, the specified tests are queued up and ran one after the other. Before the start of each test, the test parameters are transmitted to the peripheral via control characteristic, after which the test parameters are applied. Once all the parameters are confirmed by both sides, the *start test* command is sent to the control characteristic. The actual test is performed over the data characteristic. Received data is checked and the test is terminated if it is incorrect.

4.4.2 Peripheral Core

The *Peripheral Core* is used on the peripheral device. After initialisation, it starts the advertising procedure and waits for the central device to connect to it. Once the central is connected, the central device delegates all the following operations by the use of the control characteristic. The peripheral core is also responsible for decoding the control commands.

4.5 Test scenarios

We

It was decided not to test the BLE 5 Long Range, using 125 or 500 kbps Coded PHY, as the increased range offered can't be achieved without it, which would justify lower throughput and consecutively higher consumption for the transmitted amount of data. Because of the much-increased range, it also makes no sense to compare it to uncoded 1 Mb/s and 2 Mb/s PHYs, which can not even connect at the extended range that coded PHYs are offering.

is achieved on the exper

energy

same

Before doing the comparison tests, some theoretical calculations were done in regards to the expected time needed to transfer data with larger connection intervals (CIs), because of the 120 s recording limit of the measurement software. This was especially important for read and write operations, as they take two connection events (CE) to complete. In the first CE, the read/write command is sent to the peripheral and in the second CE the reply is sent to the central device.

$$packets = \frac{payload}{ATT\_MTU - 3} \tag{4.2}$$

$$time = \frac{packets}{CI} * 2 \tag{4.3}$$

From Equations 4.2 and 4.3 we have generated Tables 4.2 and 4.3. ~~Here~~ it can be seen that with bigger payloads at long connection intervals, the time that a single test would take is larger than the measuring time limit of 120 s that the measuring system used. To overcome this barrier, tests were done that are designed to test invariance of average consumption from data payload above a certain limit at different connection events. The test parameters for these tests can be observed in Table 4.4.

is bound to

energy

and

the relationship

As seen in Table 4.3, this is much less of a problem with BLE v4.2 and BLE 5, ~~as~~ they allow a maximum payload of 244 bytes ($ATT\_MTU - 3$), which decreases the number of packets needed for data transmission and thus the time needed for this transmission by a factor of ~10, as we increase payload size.

Time limitation of

Jaz ne bi govoril o testiranju invariance porabe, ampak o odvisnosti porabe od kolicine podatkov, invarianca pa je re

		Time [s] @ connection interval				
Payload	Packets	@ 7.5ms	@ 50ms	@ 400ms	@ 1000ms	@ 4000ms
20 B	1	0.015	0.1	0.8	2	8
100 B	5	0.075	0.5	4	10	40
400 B	20	0.3	2	16	40	160
1 KB	50	0.75	5	40	100	400
10 KB	500	7.5	50	400	1000	4000
100 KB	5000	75	500	4000	10000	40000

Table 4.2: Packets and time needed to transfer different payloads in relation to connection intervals for BLE v4.0 / v4.1 with *ATT_MTU* set to 23, using read/write operation.

		Time [s] @ connection interval				
Payload	Packets	@ 7.5ms	@ 50ms	@ 400ms	@ 1000ms	@ 4000ms
20 B	1	0.015	0.1	0.8	2	8
100 B	1	0.015	0.1	0.8	2	8
400 B	2	0.03	0.2	1.6	4	16
1 KB	5	0.075	0.5	4	10	40
10 KB	41	0.615	4.1	32.8	82	328
12.5 KB	52	0.78	5.2	41.6	104	416
100 KB	410	6.15	41	328	820	3280

Table 4.3: Packets and time needed to transfer different payloads in relation to connection intervals for BLE v4.2 with *ATT_MTU* set to 247, using read/write operation.

	Payload [bytes]	Connection Intervals [ms]
Read, Write	20, 100, 400, 10^3, 10^4	7.5, 50, 400, 1000, 4000

average energy (ali power) consumption measurement

Table 4.4: Test parameters for testing of BLE v4.0 / v4.1 ~~invariance of average consumption from data payload~~. for selected data payloads.

Jaz bi naredil tako tabelo: Read / Write operations--

Mislim, da bi morala biti ena v

Tabelo 4.4 bi lahko nadomestil z barvanjem izbranih parametrov v tabeli 4.2, kot sem naredil jaz z zeleno barvo. Teks

BLE version	Parameters	Connection Intervals [ms]	Procedure	Payload
v4.0/v4.1	*ATT_MTU*: 23 TX power: 8 dBm PHY: 1 Mbps	7.5, 30, 75, 150, 400, 1000	Read, Write Notify Write w/o rsp	1KB 100 KB
v4.2	*ATT_MTU*: 247 TX power: 8 dBm PHY: 1 Mbps	7.5, 30, 75, 150, 400, 1000	Read, Write Notify Write w/o rsp	12.5 KB 100 KB
v5	*ATT_MTU*: 247 TX power: 8 dBm PHY: 2 Mbps	7.5, 30, 75, 150, 400, 1000	Read, Write Notify Write w/o rsp	12.5 KB 100 KB

Table 4.5: Test parameters for testing the energy efficiency of different Bluetooth standards.

In general, tests with small data payloads were not carried out, as with small payloads (smaller than *ATT_MTU* for example) the benefits of newer and faster Bluetooth standards do not come in to play. Tests of medium data payloads (a couple of *ATT_MTU*s) were skipped ~~for comparisons~~ as well because the differences would be smaller, and we could not reap the full benefits of Bluetooth 5, so tests for comparison between standards were done with large data payloads, as seen in Table 4.5. The improvements that BLE 4.1 brought over BLE 4.0 were virtually non-existent as far as hardware, throughput, range and power consumption go. Because of that, tests for BLE v4.0 and BLE v4.1 were bundled together and done by using precisely the same parameters.

razlike med cim?

omitted

Tudi tabelo 4.5 bi lahko zamenjal s osencenim (obarvanim) delom tabel 4.2 in 4.3, pa se bolj jas

Chapter 5

Results and analysis

5.1 Power consumption measurement for different BLE standards and different BLE operations opera

~~Figures 5.1, 5.2 and 5.3 show examples of how the measurements looked like for three completely different test cases.~~

First power consumption

Figure 5.1 shows separate packets nicely, as well as radio activity at the start of each connection event. It can also be observed that the radio activates, where data was actually transmitted consumed a bit more energy than the ones where the GATT data confirmation reply was sent. Keep in mind that Link Layer reply is sent for each packet to ensure reliability. As predicted by calculations ~~in Chapter 4.5 and~~ shown in Table 4.2, five packets were needed to transfer 100 bytes of data, which occupied ten connection events and took 75 ms to complete.

when

since the device

when

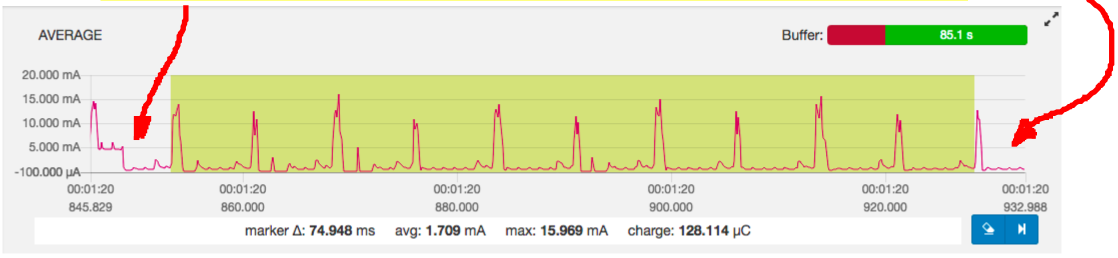
Next power consumption

Figure 5.2 is showing measurements for BLE 4.2, with *ATT_MTU* set to 247 bytes for the reading of 12.5 KB of payload data, using connection interval of 75 ms. Even though *ATT_MTU* is increased, the read operation is still bound by the need for the GATT replies, which means that a single read will occupy two connection events.

Last

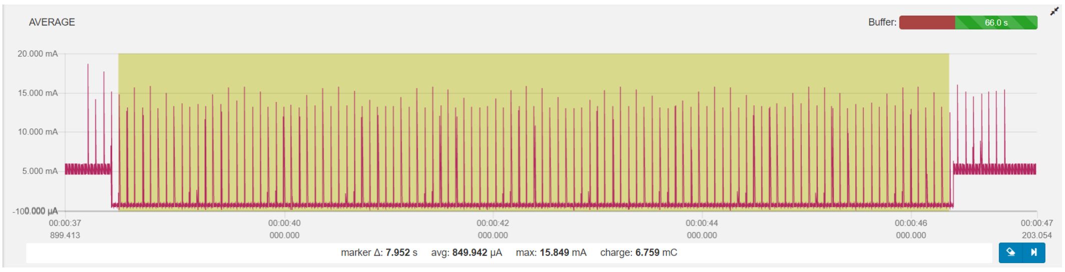
Measurement example for BLE 5 can be seen in Figure 5.3, where *ATT_MTU* is set to 247 and 2 Mbps PHY is used for transferring 100 KB of data using notifications, with the connection interval set to 150 ms. Because

Pri sliki 5.1 bi pricakoval, da bo po zadnjem paketu poraba narasla na cca



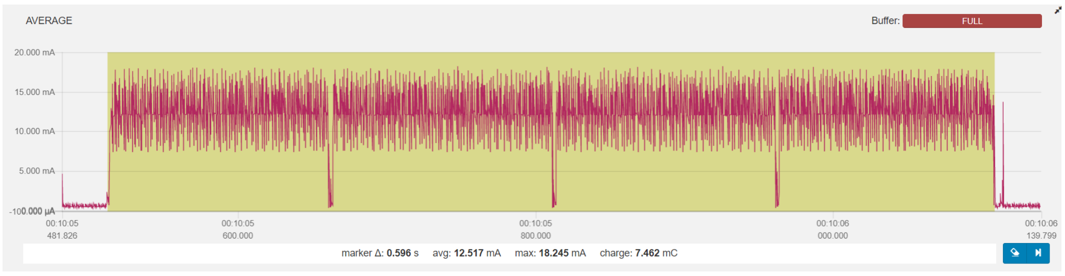
marker Δ: **74.948** ms avg: **1.709** mA max: **15.969** mA charge: **128.114** µC

Figure 5.1: Measurement example: BLE 4.0/4.1 (*ATT_MTU*: 23), write 100 bytes, connection interval 7.5ms.



marker Δ: **7.952** s avg: **849.942** µA max: **15.849** mA charge: **6.759** mC

Figure 5.2: Measurement example: BLE 4.2 (*ATT_MTU*: 247), read 12.5 KB, connection interval 75ms.



marker Δ: **0.596** s avg: **12.517** mA max: **18.245** mA charge: **7.462** mC

Figure 5.3: Measurement example: BLE 5 (*ATT_MTU*: 247), notify 100 KB, connection interval 150ms.

Ce to drzi za vse tri merit

notifications do not need the GATT layer replies but rely solely on Link Layer replies, multiple packets can be sent in the single connection event. We can see that in this instance, data transmission took just shy of four full connection events. At the start of each connection event, the central device still needs to start the connection event, as well as allow a bit of space for other possible communications to occur.

energy consumption

5.2 BLE 4.0/4.1 consumption invariance

of the

device

Power consumption ~~independence from data payload size with~~ Bluetooth v4.0 and v4.1 can be observed in Figures 5.4 for writing and 5.5 for reading. This falls in line with theoretical expectations, as a single packet is transferred every two connection events (the second connection event is used for GATT reply) and the power consumption for a single packet in a set connection event is constant. Larger data sizes simply mean that more packets are sent, which in turn averages the consumption over the number of packets sent. It is because of this invariance that we were able to use different data sizes for BLE 4.0 / 4.1 tests than for BLE 4.2 and BLE 5. BLE 4.2 and BLE 5 use a larger *ATT_MTU*, so the test data size needed to be increased, in order to maintain roughly the same amount of packets sent.

5.3 Throughput

Since the parameters AT

The primary means of changing throughput in Bluetooth Low Energy are *ATT_MTU*, connection interval and PHY modulation speed. Since it makes sense to have the highest *ATT_MTU* and PHY speed that a device ~~supports~~, the only parameter left to tune throughput is the connection Interval.

The throughput of reac

is needed

and it

Reading is limited by the fact that ~~it~~ ~~needs~~ a GATT reply, ~~which~~ is sent in the connection event after the read command. This can be seen in Figure 5.6,

that follows

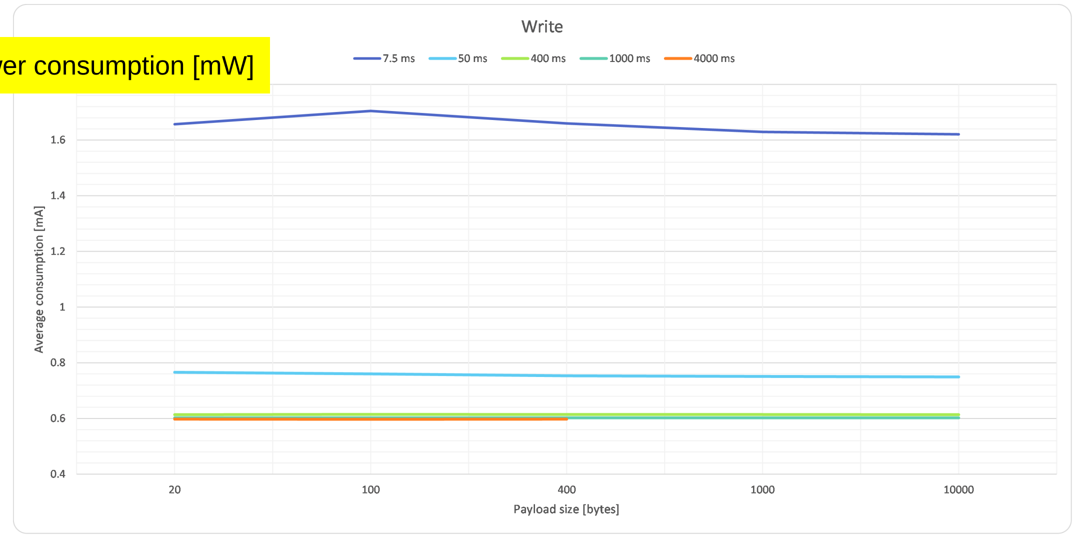
~~as~~ there is a clear correlation between connection interval and throughput.

We can see that

The reason for the correlation is that as we shorten the connection interval,

The relationship between data throughput a

Average power consumption [mW]



average

Figure 5.4: BLE 4.0/4.1 ~~invariance result~~ for write operation: Power consumption in relation to data payload size.

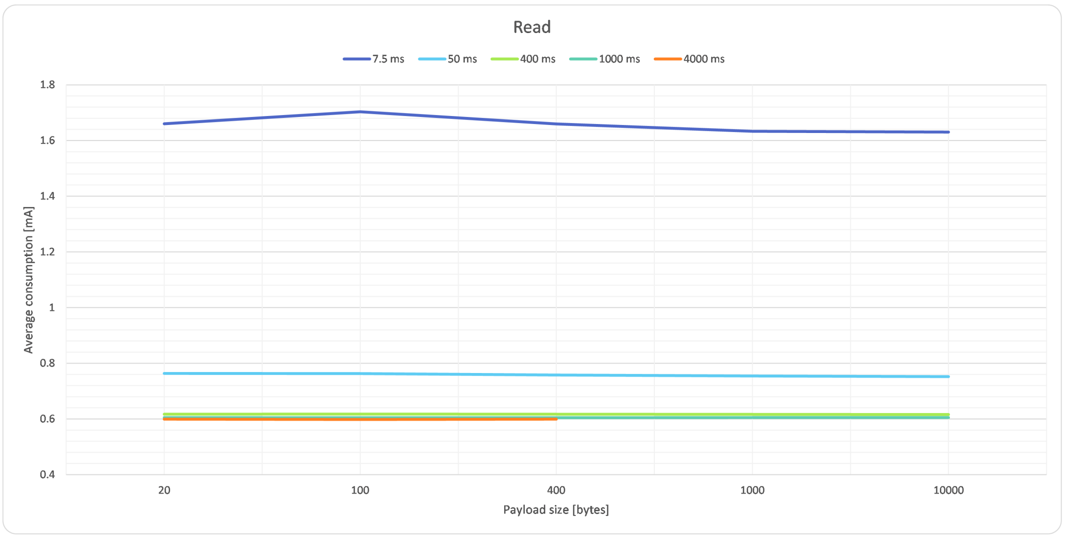


Figure 5.5: BLE 4.0/4.1 invariance result for read operation: Power consumption in relation to data payload size.

Podobno kot zgora

Nenavadno je, da je pre

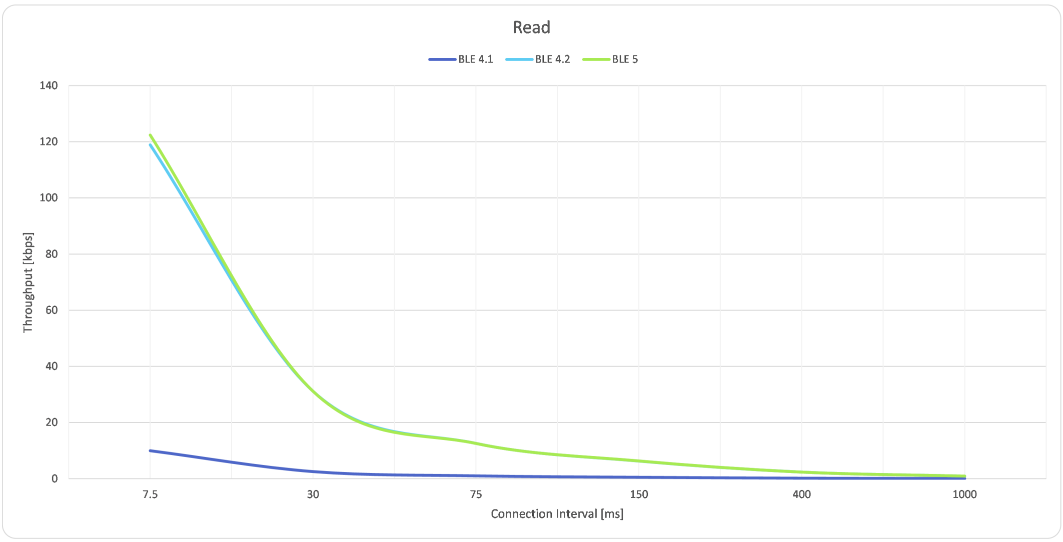


Figure 5.6: Read throughput of BLE versions at different connection intervals

the faster we get the reply, meaning we can send the next read command faster.

there is a fixed amount o

Notifications, on the other hand, are not limited by higher layer replies, as they rely solely on Link Layer acknowledgements, which are sent in the same connection event. This means that in theory, if we can transmit all the data in a single connection event, the highest throughput will be achieved. In practice, this is not true, as it can be seen in Figure 5.7. Starting from the shortest CI of 7.5 ms and going to the longest, we can see that at the beginning the throughput is increasing as we increase the connection interval. That is because there is less interrupts for connection event end and start and more time for notifications. Somewhat surprisingly, as we increase the connection interval further, throughput starts to decrease at some point. This happens in practice, as there is interference from other devices in the area, which causes some packets to fail in transmission. If a notification packet fails, the connection event ends and the devices wait for the start of the next connection event. This causes the decrease in throughput, which can

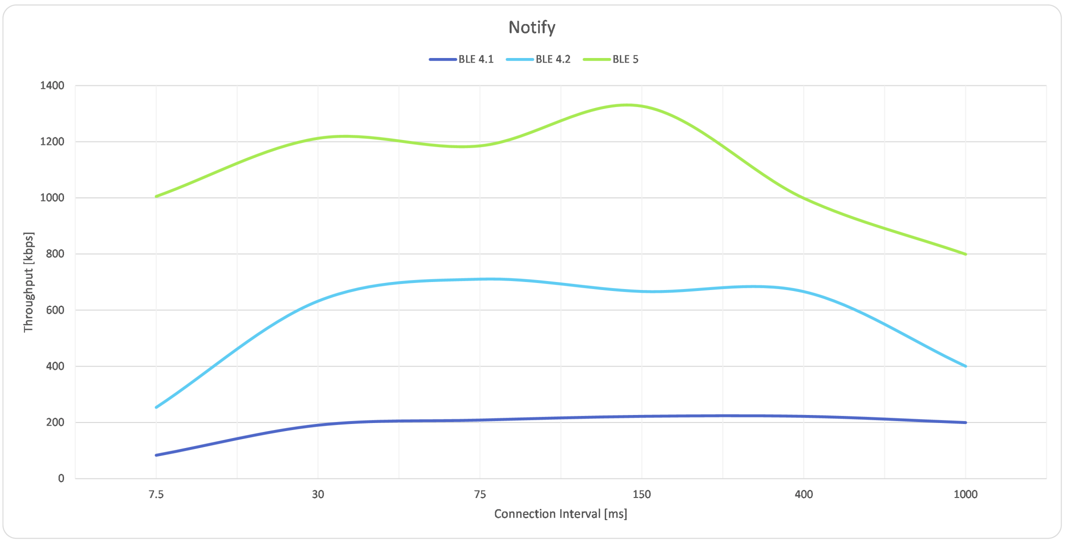


Figure 5.7: Notification throughput of BLE versions at different connection intervals

Ce so v clanku 25 teoreticne meje, potem to povsod pisi, sicer povsod

be observed in Figure 5.7 after connection interval is increased sufficiently.

Here we can also observe that the achieved throughput is at the peak of what is theoretically possible with BLE v4.1, v4.2 and 5, as it is seen in Table 5.1.

Theoretical maximum data throughput of different BLE standards were studied in [25]. The resu

BLE version	Modulation rate	Packet time	Max Throughput
v4.0 / v4.1	1 Mb/s	708 µs	0.305 Mb/s
v4.2	1 Mb/s	2500 µs	0.803 Mb/s
5	2 Mb/s	1400 µs	1.4 Mb/s

Table 5.1: Max throughput and time to transmit a single packet variance on BLE specification versions [25]

5.4 BLE version comparison: read and write

Read and write operations are bound by same limits of waiting for the response in the next connection event, so they also act the same under testing and gave ~~extremely~~ similar results. This is the reason ~~I~~ have included a single graph for read and a single graph for write results. As shown in Figure 5.6 and explained in Chapter 5.3, with read and write throughput is increased when connection interval is shortened and vice versa.

we

The relationship betwee

By looking at Figure 5.8 we can see the improvements through the BLE standard iterations, and it is interesting to see that BLE 5 is consistently using less power at one throughput than BLE 4.2, ~~already with reads and writes and the limits they pose~~. That is because even though the 2Mbps PHY modulations needs more power, it also almost halves the time to send a single packet in comparison to BLE v4.2, as seen in Table 5.1. Shorter time to send a packet means that the device can stop the radio and go into power saving mode sooner, which has an enormous impact on power consumption. Even with fragmented transmissions, as present in reading/writing operations this effect is stronger than the increase in momentary power consumption, thus improving overall power consumption.

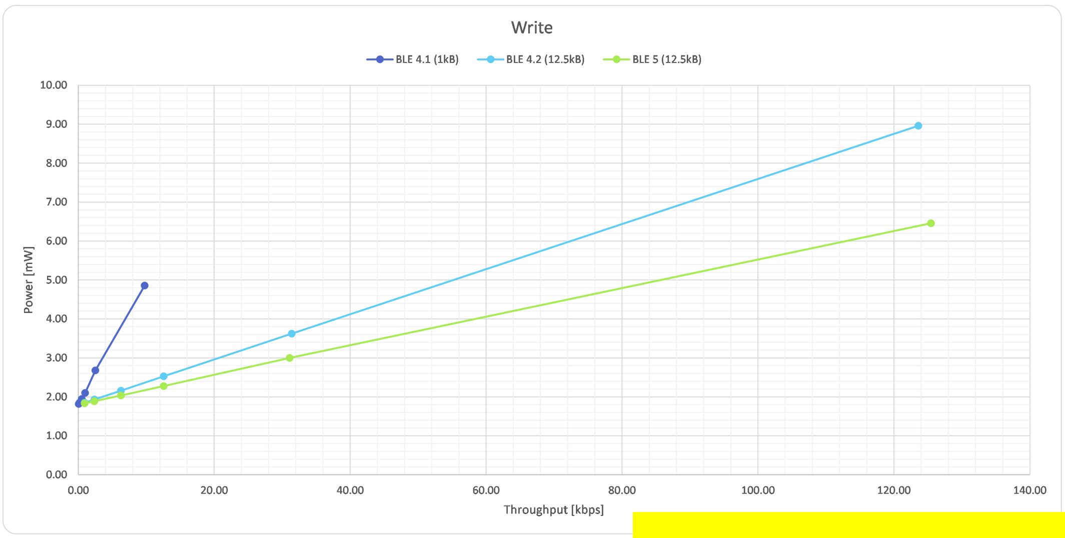
same

malce pretirano

Another interesting thing to observe is just how much BLE has evolved with v4.2 and 5, both in throughput and power consumption domains. This can be nicely seen in Figure 5.9, where we can observe that the energy needed to transmit a certain amount of data lowers with each Bluetooth iteration. Keep in mind that data size differs between BLE 4.1 and BLE 4.2 / 5. To make a fair comparison of BLE 4.1 with Ble 4.2 and BLE 5, the energy consumption for BLE 4.1 is multiplied by 12.5 in the second line (projected 12.5kB). The calculation for that is based on ~~proof of~~ consumption invariance to data size in Chapter 5.2.

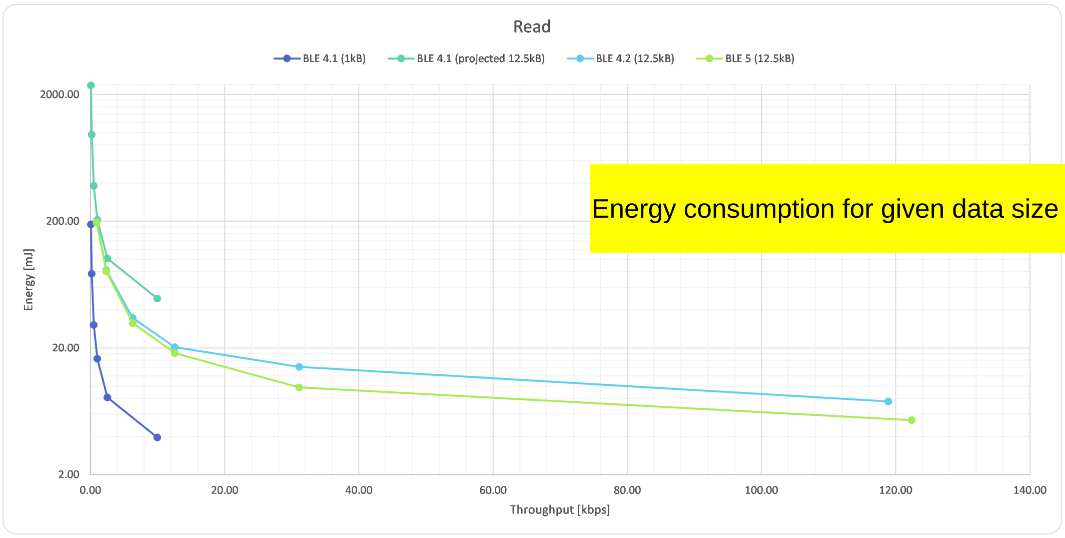
presented

Meritev se ni dokaz!



Power consumption in relation to data throughput for w

Figure 5.8: Comparison of BLE standards with write, shown as power consumed for transmission in relation to throughput. Note: Data size and transmit time differ between BLE v4.1 and BLE 4.2 / 5.



Energy consumption for given data size in relation to data throug

Figure 5.9: Comparison of BLE standards with read, shown as energy consumed for transmission in relation to throughput. Note: Energy shown on the logarithmic scale. Data size and transmit time differ between BLE v4.1 and BLE 4.2 / 5.

5.5 BLE version comparison: notify and write without response

Notifications and writes without response are very similar operations, ~~with~~ the only difference being the direction of data, which in our case does not play a significant role. They are the fastest operations that are supported by the majority of BLE devices. They are not bound by the limits of waiting for higher layer replies to commands. They still do need to respect the connection event start/end points, as well as the fact that if a packet fails to send in the current connection event, the ~~CE~~ will be ended, and the failed packet will be retried in the next CE, with the queued packets being moved to the next CE as well.

transmittion

With notify and write without response the differences between Bluetooth versions become much more apparent, as seen in Figures 5.10 and 5.11. Because with these two operations the connection events are filled with packets the differences in throughput, as well as consumption are much more apparent.

Ne vem, ce lahko iz ma

In Figure 5.10 we can see that BLE v4.1 and BLE v4.2 use the same PHY layer, as the minimum and maximum powers are practically the same. We see such clear separation between the two BLE versions because of the increase of maximum *ATT_MTU* in BLE v4.2, which enables BLE v4.2 to have a much higher throughput. When comparing BLE v4.2 and BLE 5 we can see that they use different PHY layers, as the min/max power consumption values are no longer the same. Here we can see that BLE 5 2 Mb/s PHY consumes more momentary power, but it also nets a much higher throughput (roughly 1.7x increase).

Once we look at Figure 5.11 it gets even more interesting, as we can see that the energy needed to transmit a single blob of data is lowered substantially with each BLE version. There is an immense difference in energy needed between BLE 4.1 and BLE 4.2, which can ~~again~~ be attributed solely to the increase of maximum *ATT_MTU* in BLE v4.2. This change

pretirano: signif

for fixed data transfer size

decreased the number of packets to be sent for a set amount of data on the expense of each packet taking longer to send because it can now send more data. That is a good thing because each packet needs some overhead, and an increase of payload size will decrease the ratio between overhead and payload size, which actually means fewer bits to transmit over the air. This nets a total decrease of power consumption, as well as the decrease of time to transmit the whole blob of data, which in turn decreases energy needed to do so because the radio is turned on for a shorter amount of time.

are larger.

packet

When we look at BLE 5, we can see it lowers the energy needed for transmission of the same amount of data even further. This can also be explained the same way as BLE v4.2 improvements, where we have lowered power consumption by means of shortening the amount of time the radio needs to be turned on, with the difference that for BLE 5 this is achieved by using the 2 Mb/s PHY. Even though we can see in Figure 5.10 that BLE 5 uses more power, in Figure 5.11 we can see that it uses less energy for transferring the same amount of data as BLE v4.2. This happens because the gains of shorter radio on time ~~far~~ outweigh the increase of power consumption because of the use of a faster 2 Mb/s modulation PHY layer.

v5 in 4.2 imata identicno v

Another interesting and useful observation can be made when looking at energy consumption, and that is that BLE v4.2, and even more so BLE 5, appear to use the same amount of energy regardless of throughput (connection interval) selected. This information can be beneficial, as it enables us to tune our application to suit our needs in regards to throughput and latency of read/write commands without worrying about energy consumption.

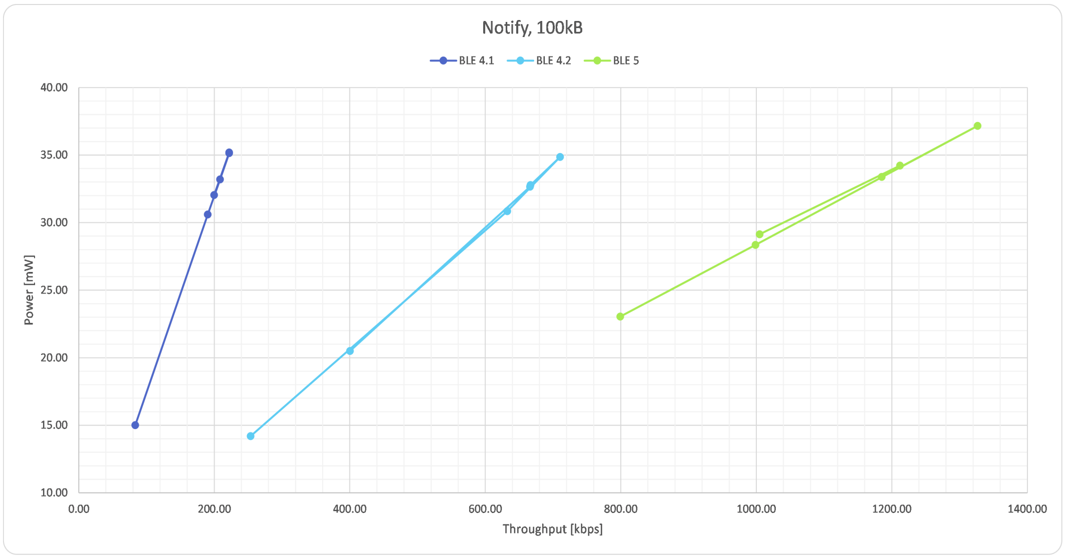


Figure 5.10: Comparison of BLE standards with notifications, shown as power consumed for transmission in relation to throughput.

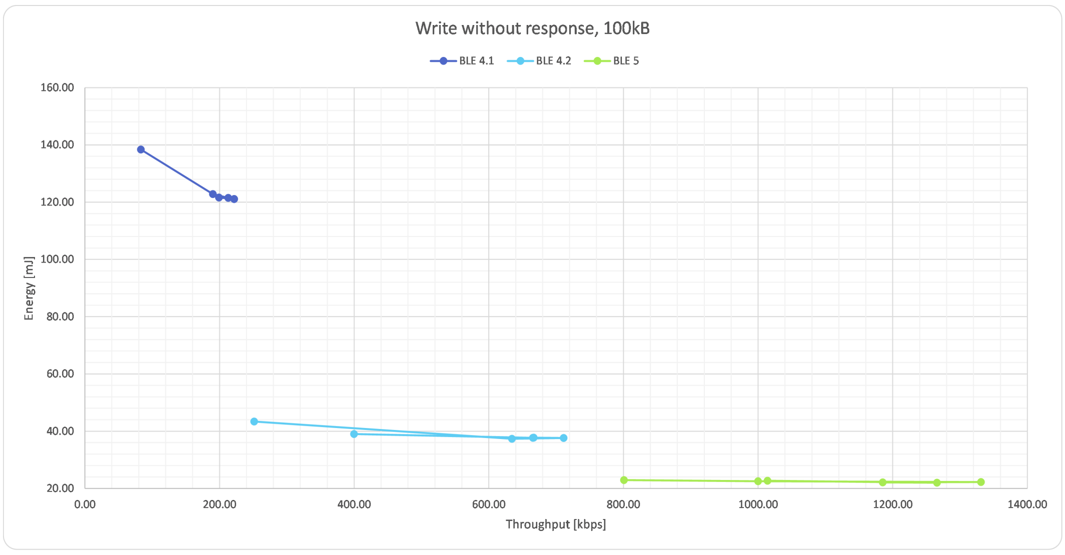


Figure 5.11: Comparison of BLE standards with write without response, shown as energy consumed for transmission in relation to throughput.

Chapter 6

Conclusions

Recently introduced Bluetooth 5 promises twice the speed, four times the range and eight times increased broadcast capacity, all of which is supposed not to affect power consumption [9]. Because no articles were found that would compare the effect of different connection parameters and Bluetooth Low Energy versions on power consumption in relation to throughput, the comparison was therefore made in this thesis. We have delved into the Bluetooth core specifications to understand the inner workings of Bluetooth Low Energy devices, what the parameters that are available to the developer are and how do they affect throughput and consumption.

In this thesis, the work done to compare power efficiency of Bluetooth Low Energy versions is presented. This includes implementing and performing tests, as well as the measurements to get the data needed. Test cases were carefully selected to highlight the improvements offered by BLE versions. To be able to overcome the limitations of our power consumption measurement system we started by confirming that above a certain data size the average consumption does not change, meaning we can compare the average power consumption of older BLE 4.1 with the newer and faster BLE 4.2 and BLE 5. This is important because of the limitation to capture at most 120 s of consumption data in one go. This posed a problem because the throughput of BLE versions is vastly different, so using large test data size breached the

power

limit with the older BLE 4.1 and using a small enough data size for BLE 4.1 when testing BLE 4.2 and BLE 5 proved to be too small to sufficiently stress the devices. Fortunately, because of the invariance of power consumption from data size, we could safely compare BLE standards even if we used different test sizes.

Next, we tested the effect of Connection Interval (see Chapter 3.3.1) on the throughput for the write and notify procedure, where we found that lowering the CI will increase throughput for writes (and reads), but that is not the case for notifications (and writes without response). For notifications, a CI that is too short will decrease the throughput because of extra overhead needed to open and close the connection event too often. On the other side, a connection interval that is too big will in practice decrease throughput because of interference and packet failures. The reason for decreased throughput, in that case, is that a connection event is closed when a packet failure happens and the packet is retried on next connection event, meaning that the rest of the closed connection event is unused, the average time of which increases as we increase connection ~~event~~ length.

thus

transmittion

with the

of

interval

Finally, we did the tests needed for comparing power efficiency in relation to throughput with different BLE versions and different operations. We grouped the comparisons by read/write operations and notify/write without response. That is because read and write are conceptually the same operations, with the difference being data flow direction, and the same goes for notify and write without response operations.

similar

When looking at read/write power consumption, we can see that Bluetooth improves with each version, with the interesting thing being that BLE 5 further improves power consumption by using the 2 Mb/s PHY (see Chapter 3.2). That is interesting because using a higher speed PHY by itself consumes more power, but since it almost halves the time needed to transmit a packet it also means that it can turn the radio off sooner, hence decreasing power consumption. That can also be seen when looking at total energy consumed to transmit the same amount of data (comparing BLE 4.2 and BLE 5), which

Razlog je v tem, da je mo

decreases with BLE 5 because of the use of 2 Mb/s PHY.

The power consumption improvements of BLE 5 in relation to throughput become even more apparent once we look at results from testing notify and write without response. When looking at power consumption, we can clearly see that 2 Mb/s PHY of BLE 5 uses more power, with the benefit of almost doubling throughput. However, if we compare total energy needed to transmit the same amount of data we can clearly see that using 2 Mb/s PHY is the most efficient option when transmission of more substantial amounts of data is needed. Another interesting fact is that BLE 4.2 and BLE 5 actually consume practically the same amount of energy to transmit an amount of data, regardless of Connection Interval setting (and consecutively the throughput). This information can be beneficial, as it enables us to tune our application to suit our needs in regards to throughput and latency of read/write commands without worrying about energy consumption when large amounts of data need to be transmitted.

The findings in this thesis prove that there is no reason not to switch to developing and using devices that use Bluetooth Low Energy 5, as it brings higher throughput, extended range and better coexistence while using the same or lower amount of power for transmissions.

Bibliography

[1] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen, "How Low Energy is Bluetooth Low Energy? Comparative Measurements with ZigBee/802.15.4," *IEEE*, 2012.

[2] K. Mikhaylov, N. Plevritakis, and J. Tervonen, "Performance Analysis and Comparison of Bluetooth Low Energy with IEEE 802.15.4 and SimpliciTI," *Journal of Sensor and Actuator Networks*, August 2013.

[3] F. Touati, O. Erdene-Ochir, W. Mehmood, A. Hassan, A. BenMnaouer, B. Gaabab, M. F. A. Rasid, and L. Khriji, "An experimental performance evaluation and compatibility study of the bluetooth low energy based platform for ecg monitoring in wbans," *International Journal of Distributed Sensor Networks*, p. 12, 2015.

[4] S. Kamath, "Application Note AN092: Measuring Bluetooth Low Energy Power Consumption," 2010.

[5] M. Imran, "Power Measurements Techniques For Embedded Systems," *Mid Sweden University*, 2015.

[6] Ž. Nakutis, "Embedded Systems Power Consumption Measurement Methods Overview," *MATAVIMAI*, pp. 29–35, 2009.

[7] Nordic Semiconductor, "nRF52840 Development Kit." Available: `https://www.nordicsemi.com/eng/Products/nRF52840-DK`, 2017. acquired: 27.11.2017.

[8] Nordic Semiconductor, “Power Profiler Kit.” Available: `https://www.nordicsemi.com/eng/Products/Power-Profiler-Kit`, 2018. acquired: 27.8.2018.

[9] M. Woolley, “Bluetooth 5 / Go Faster. Go Further,” tech. rep., Bluetooth SIG, 2018.

[10] Bluetooth SIG, *BLUETOOTH SPECIFICATION Version 2.0 + EDR*, 2004.

[11] Bluetooth SIG, *BLUETOOTH SPECIFICATION Version 2.1 + EDR*, 2007.

[12] Bluetooth SIG, *BLUETOOTH SPECIFICATION Version 3.0 + HS*, 2009.

[13] Bluetooth SIG, *BLUETOOTH SPECIFICATION Version 4.0*, 2010.

[14] Bluetooth SIG, *BLUETOOTH SPECIFICATION Version 4.1*, 2013.

[15] Bluetooth SIG, *BLUETOOTH SPECIFICATION Version 4.2*, 2014.

[16] Bluetooth SIG, *BLUETOOTH SPECIFICATION Version 5.0*, 2016.

[17] CNX, “Bluetooth 5 Promises Four times the Range, Twice the Speed of Bluetooth 4.0 LE Transmissions.” Available: `https://www.cnx-software.com/2016/06/10/bluetooth-5-promises-four-times-the-speed-twice-the-range-of-bluetooth-4-0-le-` 2016. acquired: 7.9.2018.

[18] R. Heydon, *Bluetooth Low Energy: The Developer’s Handbook*. Prentice Hall, October 2012. ISBN: 9780132888394.

[19] R. D. Akiba, C. Cufí, and K. Townsend, *Getting Started with Bluetooth Low Energy*. O’Reilly Media, Inc., May 2014. ISBN: 9781491949511.

[20] Nordic Semiconductor, "nRF52840." Available: `https://www.nordicsemi.com/eng/Products/nRF52840`, 2017. acquired: 27.11.2017.

[21] SEGGER Microcontroller, "J-Link-OB probe." Available: `https://www.segger.com/products/debug-probes/j-link/models/j-link-ob/`, 2017. acquired: 27.11.2017.

[22] Nordic Semiconductor, "SoftDevices." Available: `https://www.nordicsemi.com/eng/Products/SoftDevices`, 2017. acquired: 27.11.2017.

[23] SEGGER Microcontroller, "Real Time Transfer." Available: `https://www.segger.com/products/debug-probes/j-link/technology/about-real-time-transfer/`, 2017. acquired: 27.11.2017.

[24] git-scm.com, "Git Tools - Submodules." Available: `https://git-scm.com/book/en/v2/Git-Tools-Submodules`, 2018. acquired: 7.9.2018.

[25] K. Ren, Bluetooth SIG, "Exploring Bluetooth 5 - How Fast Can It Be?." Available: `http://blog.bluetooth.com/exploring-bluetooth-5-how-fast-can-it-be`, 2017. acquired: 7.9.2018.