ANALIZA I IMPLEMENTACIJA ANT+ PROTOKOLA

Rogulj, Roko

Master's thesis / Specijalistički diplomski stručni

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split / Sveučilište u Splitu**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:228:339306

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**



Repository / Repozitorij:

Repository of University Department of Professional Studies

zir.nsk.hr

UNIVERSITY OF SPLIT

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Specijalistički diplomski stručni studij Elektrotehnike

ROKO ROGULJ

ZAVRŠNI RAD

ANALIZA I IMPLEMENTACIJA ANT+ PROTOKOLA

Split, lipanj 2019.

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Specijalistički diplomski stručni studij Elektrotehnike

Predmet: Senzorske mreže

ZAVRŠNI RAD

Kandidat: Roko Rogulj

Naslov rada: Analiza i implementacija ANT+ protokola

Mentor: dr. sc. Tonko Kovačević

Split, lipanj 2019.

SADRŽAJ

Sažetak 3

1. UVOD 4

2. ANT PROTOKOL 6

2.1 ANT kanali 7

2.2 Konfiguracija ANT kanala 8

2.2.1 Tip kanala 9

2.2.2 Frekvencija kanala 10

2.2.3 Identifikator kanala 10

2.2.4 Vremenski period kanala 12

2.2.5 Mreža 12

2.3 Uspostava ANT kanala 14

2.4 Tipovi podataka 16

2.4.1 Broadcast 17

2.4.2 Acknowledged 18

2.4.3 Burst 18

2.4.4 Advanced Burst 19

2.5 Uparivanje ANT uređaja 19

2.6 ANT sučelje 22

3. ANT+ PROTOKOL I UREĐAJI 25

3.1 ANT+ profili 26

3.2 ANT+ uređaji 27

3.3 Analiza ANT+ Hearth Rate profila 30

3.3.1 Formati stranica Hearth Rate profila 31
3.3.2 Analiza ANT+ protokola preko USB sučelja 32
4. IMPLEMENTACIJA ANT+ PROTOKOLA POMOĆU NODE.JS 34
4.1 Node.js 34
4.2 Instalacija ANT+ Node.js modula 34
4.3 Vizualizacija otkucaja srca 36
4.3 Programski kod 38
4.3.1 Index.html datoteka 38
4.3.2 Chart.js datoteka 39
4.3.3 Index.js datoteka 40
6. ZAKLJUČAK 42
LITERATURA 43
POPIS SLIKA 44
POPIS TABLICA 45

Sažetak

Analiza i implementacija ANT+ protokola

Senzorske bežične mreže dio su industrije već dugi niz godina te se potreba za njihovom implementacijom iz dana u dan povećava. Jedan od problema implementacije bežičnih mreža je energetska efikasnost jer uređaji nisu mogli biti izvedeni u kompaktnom obliku zbog velike energetske potrošnje. U tu svrhu osmišljen je ANT protokol, kao alternativa dosadašnjim bežičnim mrežama, kojemu je glavna odlika visoka energetska efikasnost gdje jedan senzor na mreži može raditi i godinu dana na bateriji koja je veličine novčića. ANT+ je upravljana mreža ANT protokola, a koristi se kod mjernih uređaja u sportu i kućnoj medicini. U ovome radu detaljno će se analizirati ANT+ mreža u okvirima ANT protokola te pokazati implementacija vizualizacije mjerenja otkucaja srca ANT+ uređajima preko *Node.js* platforme.

Ključne riječi: *ANT protokol, ANT+ protokol, bežične mreže*

Summary

Analysis and implemetation of ANT+ protocol

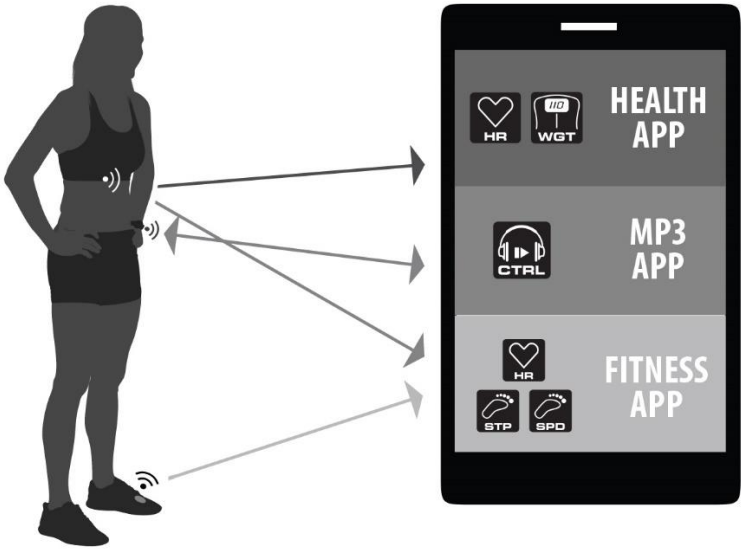
Wireless sensor networks have been part of the industry for many years now, and the need for their implementation has grown from day to day. One of the problems of wireless network implementation was energy efficiency where devices could not be implemented in compact form due to high energy consumption. For this purpose, the ANT protocol was designed as an alternative to wireless networks at the time, the main feature of which is high energy efficiency, where an online sensor can work for a year on a coin-sized battery. ANT + is a managed ANT protocol network, and it is used for sensory measurement in sports and home medicine. This paper deals with analyzation of ANT+ network within ANT protocol framework and implementation of ANT+ Hearth Rate measurement visualization via *Node.js* platform.

Keywords: *ANT protocol, ANT+ protocol, wireless networks*

1. UVOD

Bežične senzorske tehnologije doživjele su značajan tehnološki rast posljednjih godina. Potražnja tržišta uvjetovala je razvitak tehnologija koje se zasnivaju na uređajima visoke energetske efikasnosti te su iz tog skupa iznjedrile dva značajna protokola: *Adaptive Network Topology* (ANT) i *Bluetooth Low Energy* (BLE), gdje oba protokola rade u 2.4 GHz ISM frekvencijskom rasponu.

Oba protokola su zamišljena za primjenu u okvirima WBAN - *Wireless Body Area Network*. BLE dodatno podržava, zbog velikih brzina prijenosa (2 Mbit/s), prenošenje glazbe i videa preko uređaja. Iako se velika brzina prijenosa čini kao prednost, BLE jako je ograničen kod postavljanja same mreže za razliku od ANT protokola. ANT koristi adaptivnu transmisiju podataka, koja omogućava komunikaciju velikog broja uređaja bez interferencije signala te ne zahtijeva postojanje globalnog takta razmjene podataka. ANT je u srži jednostavniji protokol od BLE te se njegova komunikacijska mreža može ostvariti kroz sve topologije, gdje se kod BLE može koristi samo zvjezdasta topologija [1].



Slika 1. ANT bežična senzorska mreža [2]

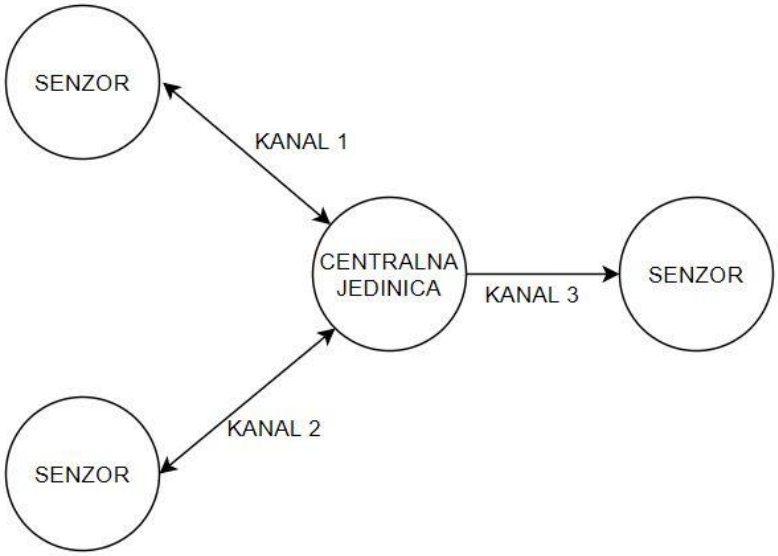
ANT je izrađen i dizajniran od strane tvrtke *ANT Wireless* (podružnica tvrtke *Garmin*). Iz ANT protokola posebno su se počele razvijati upravljane mreže s ciljem ostvarivanja kompatibilnosti svih proizvedenih uređaja na toj mreži. Jedna takva mreža je ANT+ koja je upravljana grupom zvanom *ANT+ Alliance*. *ANT+ Alliance* daje licence za upotrebu i implementaciju ANT+ protokola i osigurava kompatibilnost svih ANT+ uređaja [3].

U prvome dijelu ovoga rada analizirat će se detalji ANT protokola. Posebna pozornost bit će stavljena na način komuniciranja uređaja unutar ANT mreže te konfiguracije same mreže. Drugi dio rada bit će osvrt na ANT+ protokol kao podskup ANT protokola gdje će se opisati dodatne funkcije koje su implementirane unutar ANT+ protokola.

Treći dio rada obuhvaća implementaciju samoga ANT+ protokola kroz Node.js platformu. Prikazat će se jedna od metoda vizualizacije mjerenja otkucaja srca preko ANT+ uređaja različitih proizvođača. Osim vizualizacije podataka biti će izvedeno i spremanje mjerenih vrijednosti u bazu podataka.

2. ANT PROTOKOL

ANT je bežični mrežni protokol koji radi u 2.4 GHz ISM frekvencijskom pojasu namijenjen za prikupljanje senzorskih podataka. Glavna odlika ovoga protokola je u energetskoj efikasnosti gdje je potrošnja električne energije višestruko manja u odnosu na slične protokole iz iste domene. Na slici 2. prikazan je primjer jednostavne ANT mreže.



Slika 2. Jednostavna ANT mreža

ANT mreža definirana je preko kanala. Svaki čvor mreže (prikazan kao krug na slici 1.) može se povezati pomoću kanala s drugim čvorom. Uobičajeno je da svaki kanal povezuje dva čvora, iako se mreža može konfigurirati da kanal povezuje više čvorova.[4]

Svaki kanal, kao minimalni uvjet, mora imati barem jedan nadređeni čvor - *master* i jedan podređeni čvor - *slave*. Svaki čvor može biti i nadređeni i podređeni čvor istovremeno. Također, svaki čvor u mreži može se ponašati kao prijemnik, odašiljač ili oboje (prikazano kroz jednosmjerne i dvosmjerne strelice na slici 1) [4].

ANT mreža može biti izvedena u širokom spektru topologija kao što su *Mesh* topologija, zvjezdasta topologija , *peer-to-peer* topologija , *Broadcast* sustav [4]…

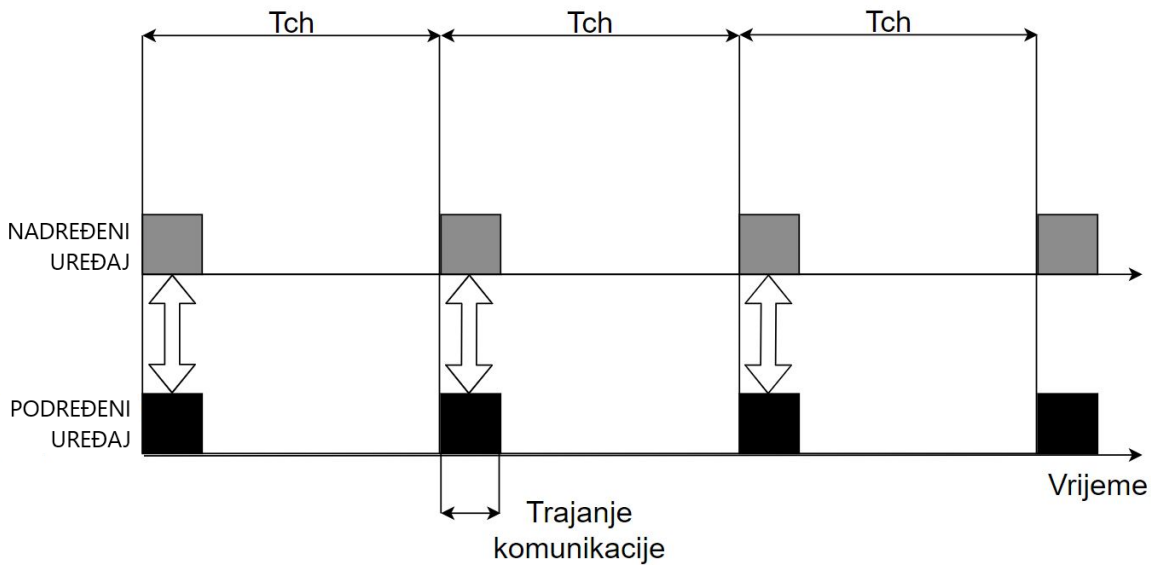
Svi čvorovi mreže sastoje se od ANT transmiterskog sustava i upravljačke jedinice (mikrokontroler ili računalo) gdje transmiterski sustav preuzima odgovornost o svim operacijama potrebnih za uspostavu i održavanje ANT kanala. Upravljačka jedinica preuzima zadaće procesiranja funkcija sustava u kojima se nalazi čvor te komunicira s ANT transmiterom preko serijskog sučelja.

2.1 ANT kanali

Komunikacija između čvorova ANT mreže odvija se preko kanala. Način na koji se odvija komunikacija ovisi o :

- tipu kanala,
- konfiguraciji kanala,
- vrsti podataka koji se šalju
- smjeru slanja podataka [4].

Većina ANT čvorova koristi sinkronu, nezavisnu i dvosmjernu komunikaciju. Svaki put kada nadređeni čvor otvori sinkroni kanal, prvo provjerava da li se ta komunikacija interferira s transmisijom nekog drugog uređaja. Ako nema interferencije postavlja se određeni vremenski period unutar čvora T_{ch} – **vremenski period kanala** (slika 3), unutar kojega nadređeni uređaj šalje podatke. Kod dvosmjerne komunikacije, nadređeni uređaj ostavlja prijemnik upaljenim nakon slanja podataka da bi podređeni uređaj mogao poslati svoje podatke [4].



Slika 3. Komunikacija između nadređenog i podređenog uređaja

2.2 Konfiguracija ANT kanala

Za ostvarivanje komunikacije između dva ANT uređaja potrebno je definirati sljedeće parametre:

- tip kanala,
- frekvenciju kanala
- identifikator kanala (tip transmisije, tip uređaja i broj uređaja)
- vremenski period kanala,
- mreža [4].

Većina parametara kanala može se mijenjati tijekom komunikacije. Mora se naglasiti da nadređeni uređaj može raditi na više kanala gdje su na svim kanalima različite postavke. U nastavku su pojedinačno opisani parametri ANT kanala.

2.2.1 Tip kanala

Tip kanala određuje vrstu komunikacije koja se odvija između uređaja. Tip je određen 8 bitnim poljem unutar ANT poruke te mora biti definiran prije otvaranja kanala. Tipovi ANT kanala s njihovim vrijednostima unutar ANT sučelja prikazani su u tablici 1.

Tablica 1. Tipovi ANT kanala [4]

Tip	**Vrijednost polja u HEX**
Dvosmjerni *slave* kanal	0x00
Dvosmjerni *master* kanal	0x10
Dijeljeni dvosmjerni *slave* kanal	0x20
Dijeljeni dvosmjerni *master* kanal	0x30
Slave jednosmjerni kanal	0x40
Master jednosmjerni kanal	0x50

Dvosmjerni kanal podržava tok podataka u oba smjera (*slave->master* i *master->slave*). Primarni smjer određen je postavkama čvora tako da čvor konfiguriran kao podređeni čvor primarno će primati podatke ali može i slati podatke, iako mu to nije glavna namjena [4].

Dijeljeni kanali se koriste kod čvorova koji moraju primati i obrađivati podatke od više drugih čvorova. U ovome slučaju više čvorova dijeli isti kanal za komuniciranje prema središnjem čvoru [4].

Jednosmjerni kanali mogu komunicirati samo u jednome smjeru, odnosno nadređeni uređaj ne može primati podatke od nijednog podređenog uređaja. U ovakvoj konfiguraciji koristi se samo *broadcast* tip podataka za slanje. Ovakav tip kanala koristi se u dijagnostičke svrhe [4].

Nezavisni kanali su oni kanali u kojima postoji samo jedan nadređeni i samo jedan podređeni čvor [4].

Dijeljeni kanali su oni kanali u kojima jedan nadređeni uređaj sakuplja i obrađuje podatke za više podređenih uređaja odnosno, postoji samo jedan nadređeni uređaj unutar mreže [4].

Komunikacija između dva čvora može biti ***forward*** ili ***reverse*** tipa. Kod *reverse* tipa komunikacije najčešće se šalju kontrolni podaci, podaci potvrde ili signali greške prijenosa podataka. Takva komunikacija je suprotnog smjera nego što je *forward* komunikacija koja podrazumijeva slanje korisničkih podataka. *Forward* i *reverse* nazivi se koriste kao oznaka „smjera“ kanala.

2.2.2 Frekvencija kanala

ANT tehnologija podržava korištenje 125 jedinstvenih frekvencija. Kanal će raditi na jednoj frekvenciji dokle god postoji takav kanal, a ona mora biti poznata svi podređenim i nadređenim uređajima na istome kanalu. Postoji mogućnost mijenjanja frekvencije tijekom rada ali to može rezultirati desinkronizacijom podređenih uređaja iz kanala [4].

Frekvencija kanala je 8 bitno polje postavljeno unutar ANT poruke gdje se vrijednosti kreču od 0 do 124. Vrijednost unutar poruke označava pomak po 1 MHz inkrementima od osnovne frekvencije 2400 MHz. Tako da bi primjerice vrijednost 50 značila da je trenutna frekvencija kanala 2450 MHz.

2.2.3 Identifikator kanala

Da bi se ostvarila veza između ANT čvorova mora se jedinstveno definirati identifikator kanala. Identifikator kanala je 32-bitna vrijednost (podijeljena na 4 bajta) koja se logički dijeli na 3 polja: **tip komunikacije, tip uređaja i broj uređaja**. Nadređeni uređaji koriste ovu vrijednost da odrede kako će komunicirati s podređenim uređajima dok podređeni uređaji koriste ovu vrijednost da znaju s kojim nadređenim uređajem će komunicirati [4].

U komunikaciju mogu stupiti samo oni uređaji kojima su polja identifikatora kanala kompatibilna. Svaki slave uređaj može postaviti vrijednost ovoga polja u nulu te će pri uspostavi komunikacije promijeniti to polje prema parametrima mrežnog ključa i frekvencije mreže. Kod nadređenih uređaja identifikator kanala mora biti definiran prije uspostave komunikacije. U nastavku su opisana pojedina polja identifikatora kanala [4].

Tip komunikacije je 8-bitno polje koje se koristi za definiranje značajki uređaja u transmisiji. Opis bitova prikazan je u tablici 2.

Tablica 2. Bitovi polja tipa komunikacije [3]

Bit	***Opis***
0-1	00: Rezervirani kanal 01: Nezavisni kanal 10: Dijeljeni kanal s adresom veličine 1 bajt 11: Dijeljeni kanal s adresom veličine 2 bajta
2	0: Globalne stranice se ne koriste 1: Globalne stranice se koriste
3	Nedefiniran
4-7	Dodatno polje za broj uređaja (nije potrebno definirati)

Ovo polje bitova uvijek mora biti definirano na nadređenim uređajima dok se kod podređenih uređaja može ostaviti prazno (identifikator kanala neće biti u cijelosti isti na početku komunikacije dok se u trenutku ostvarivanja komunikacije polje tipa komunikacije ne izjednači na podređenom i nadređenom uređaju) [4].

Tip uređaja je 8-bitno polje koje označava vrstu svih uređaja koji se koriste u mreži. Ovo polje se upotrebljava i kada se u mreži koristi više tipova različitih uređaja (senzori temperature, otkucaja srca, brzine …). ANT uređaj će prema ovome polju odrediti kako se ponaša prema korisničkim podacima koje šalje pojedini uređaj [4].

Broj uređaja je 16-bitno polje koje je jedinstveno za svaki tip uređaja u istoj mreži. U većini slučajeva odgovara serijskom broju uređaja ali može i biti nasumično generirano ako uređaj nema serijski broj. Vrijednosti 0x0000 i 0xFFFF su rezervirane za testiranja i ne mogu se koristiti [4].

2.2.4 Vremenski period kanala

Vremenski period kanala je polje u ANT poruci koje određuje brzinu prijenosa paketa podataka koje šalje nadređeni uređaj. Paketi se šalju i primaju diskretno i to u određenim vremenskim razmacima. Vremenski period kanala računa se prema sljedećoj formuli (1):

$$Period\ kanala = {}^{32768}/_{Broj\ poruka\ u\ Hz} \qquad (1)$$

Primjerice, ako želimo postići brzinu slanja paketa od 4 paketa po sekundi, odnosno 4 Hz, period kanala moramo postaviti na vrijednost 8129. Brzina slanja paketa može biti u rasponu od 0.5 do 200 Hz te se mogu dobiti brzine prijenosa podataka i do 20 kbps [4].

2.2.5 Mreža

Mreža u ANT protokolu može biti javna, upravljana ili privatna te se za svaki čvor mogu postaviti posebna pravila vezana za njihovu komunikaciju unutar mreže. Da bi dva uređaja mogla komunicirati moraju biti dio iste mreže [4].

U **upravljanim mrežama** striktno su definirana pravila korištenja protokola. Primjer takve mreže bio bi ANT+. Ovime se postiže isti standard kod svih proizvođača uređaja te kompatibilnost svih uređaja koji se spajaju na mrežu. ANT+ je mreža ANT protokola koja se bavi pretežito sportskim pomagalima i medicinskim uređajima za kućnu upotrebu. ANT+ mreža razlikuje različite profile podataka s obzirom na to koji se uređaj nalazi na mreži (da li se mjere otkucaji srca, temperatura…) [4].

Privatna mreža osigurava privatnost svih korisnika na toj mreži i onemogućuje pristup uređajima koji nisu dio te mreže. Kanali na uređajima se mogu konfigurirati tako da je moguće spajanje uređaja na više različitih mreža [4].

Mreža je definirana kroz polje podataka od 16 bita unutar ANT poruke. Sastoji se od dva djela:

1) Broj mreže

Broj mreže je 8-bitno polje koje definira sve mreže koje su dostupne ANT uređaju. Kada je vrijednost ovoga polja nula uređaj je dio javne mreže. Za sve ostale mreže definiran je mrežni ključ u paru s brojem mreže. Više kanala može imati isti broj mreže.

2) Mrežni ključ

Mrežni ključ je 8-bitni broj koji jedinstveno opisuje mrežu u kojoj se uređaj nalazi te pruža dodatnu sigurnost samoj mreži [4].

Primjer konfiguracije jednostavnog ANT kanala prikazan je u tablici 3.

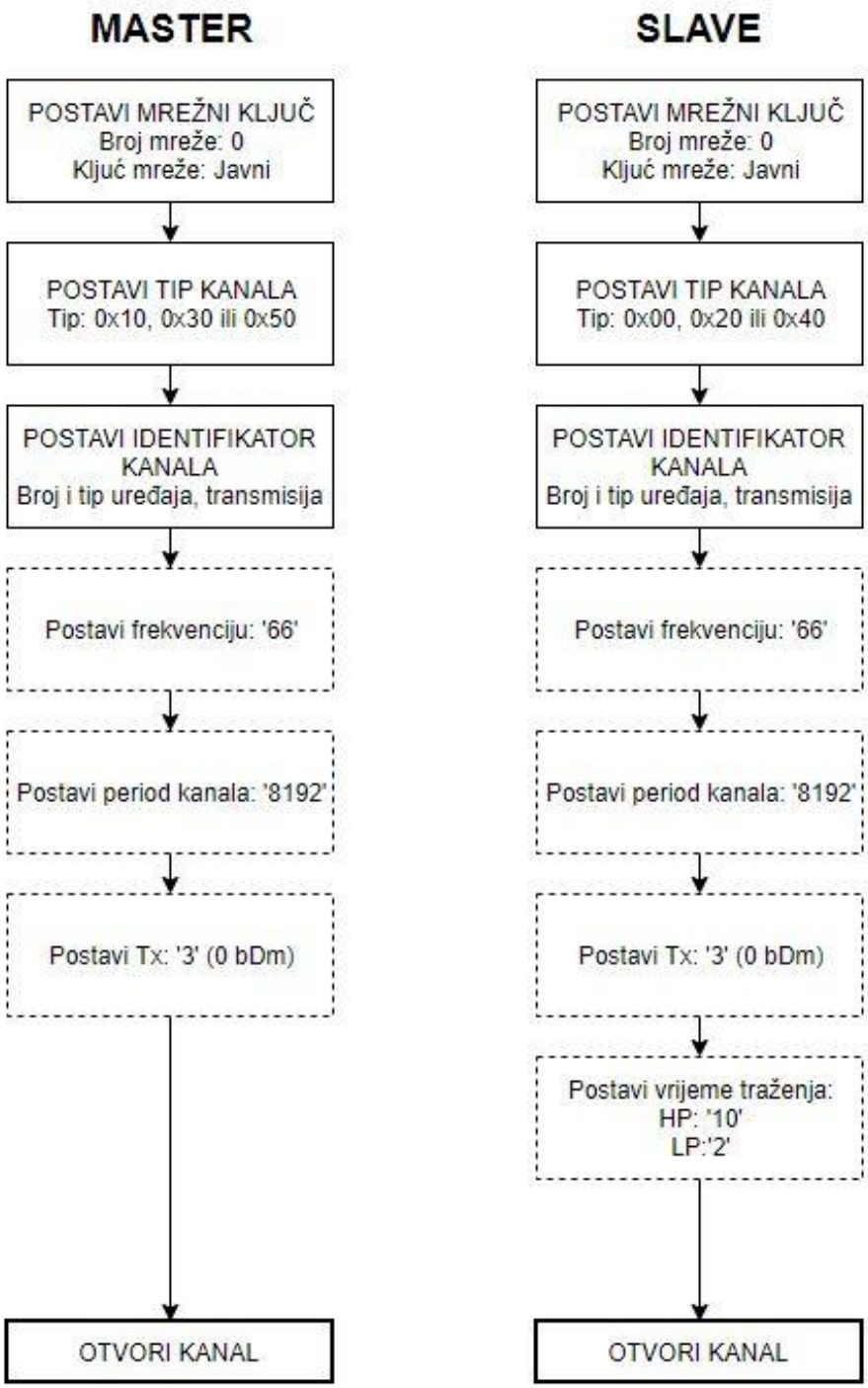
Tablica 3. Primjer konfiguracije kanala

Parametar	***Vrijednost***	***Opis***
Mrežni ključ	0	Ključ javne mreže
Frekvencija	66	Frekvencija od 2466 MHz
Broj uređaja	1	Serijski broj uređaja
Tip prijenosa podataka	1	Osnovni tip
Tip uređaja	1	Osnovni tip
Vrsta kanala	0x10	Dvosmjerni kanal
Period kanala	16384	2 Hz takt poruka
Tip podataka	0x4E	*Broadcast*

2.3 Uspostava ANT kanala

Prije uspostave kanala podređeni i nadređeni uređaj trebaju poznavati konfiguraciju kanala (tablica 3). Na slici 4. prikazan je primjer procesa uspostave kanala. Postavke na slici koje nemaju unaprijed definirane vrijednosti označene su punom linijom dok postavke koje imaju unaprijed definirane vrijednosti označene su iscrtanom linijom (proizvođač uređaja određuje unaprijed definirane postavke) [4].

Osnovna postavka mreže je javna mreža kojoj je pridijeljen broj mreže '0'. Nakon unosa mrežnog ključa mora se postaviti tip kanala. Primjerice, nadređeni uređaj bit će postavljen za slanje podataka dok će podređeni uređaj biti postavljen za primanje podataka. Identifikator kanala mora biti specificiran na nadređenom uređaju dok se na podređenom uređaju, pri konfiguraciji, polje identifikatora kanala može ostaviti prazno. U trenutku kada se postavi mreža, tip i identifikator kanala, sve ostale postavke automatski će se podesiti na osnovne vrijednosti definirane samom izvedbom uređaja. Korisnik po potrebi može mijenjati ove postavke [4].



Slika 4. Primjer uspostave kanala

Nakon što su postavljeni svi parametri kanala, nadređeni uređaj započinje transmisiju podataka u određenim vremenskim razmacima definiranim unutar postavki kanala. Ova transmisija će trajati dok korisnik ili kontroler ne prekinu komunikaciju.

Podređeni uređaji, nakon otvaranja kanala, odmah traže nadređeni uređaj koji zadovoljava kriterije postavki kanala. Kada pronađu nadređeni uređaj, primaju podatke po definiranim brzinama. Ako se ne pronađe nadređeni uređaj unutar definiranog vremena traženja, podređeni uređaj zatvara kanal (nadređeni uređaj nikada ne traži podređene uređaje te nema vrijeme gašenja kanala).

2.4 Tipovi podataka

Razlikujemo četiri tipa podataka u ANT mrežama: *Broadcast*, *Acknowledged*, *Burst* i *Advanced Burst*. Tip podataka se ne definira kao parametar kanala stoga jedan kanal može koristiti sve tipove podataka. Svaki od ova četiri tipa podataka može biti poslan u *forward* ili *reverse* smjeru kod dvosmjerne komunikacije. Jedno ograničenje se postavlja kod jednosmjernih kanala gdje se može koristiti samo *forward-broadcast* tip kanala [4].

Tablica 4. Tipovi podataka u ANT protokolu

Tip podatka	***Smjer kanala***	***Opis***
Broadcast	*Forward*	Osnovni tip podataka. Šalje se u svakom periodu kanala.
	Reverse	Podaci koji se šalju samo na zahtjev podređenog uređaja.
Acknowledged	*Forward*	Šalje se samo na zahtjev.
	Reverse	Podaci koji se šalju samo na zahtjev podređenog uređaja te se ne mogu poslati ako prethodno nije bila primljena poruka.

Burst	*Forward*	Paketi koji su sinkronizirani odvojeno od glavnog takta slanja poruka. Dolazi do retransmisije paketa ako uređaj nije primio nove podatke.
	Reverse	Šalju se samo na zahtjev podređenog uređaja. Šalju se samo jednom te nema retransmisije.
Advanced Burst	*Forward*	Paketi koji su sinkronizirani odvojeno od glavnog takta slanja poruka. Dolazi do retransmisije paketa ako uređaj nije primio nove podatke. Brzine slanja su veće nego kod *Burst* podataka.
	Reverse	Šalju se samo na zahtjev podređenog uređaja. Šalju se samo jednom te nema retransmisije.

2.4.1 Broadcast

Broadcast je osnovni tip podataka kod ANT protokola i postavljen je kao osnovna opcija komunikacije. Ovi podaci se šalju s nadređenog uređaja na kanalu prema podređenome uređaju u svakom periodu kanala. Podređeni uređaj također može slati podatke u *reverse* smjeru samo ako je to zatraženo od strane upravljačke jedinice podređenog uređaja (iako je uobičajeno da se ni jedan podatak ne šalje bez zahtjeva nadređenog uređaja) [4].

Nadređeni uređaji uvijek šalju podatke u *forward* smjeru u svakom vremenskom periodu kanala. Ako nema podataka koje upravljačka jedinica želi poslati u danome trenutku, bez obzira na tip podataka koji se šalje, ponavlja se slanje prethodne poruke kao *broadcast* tip podataka. Poruke *reverse* tipa se ne trebaju slati periodično već se šalju samo jednom [4].

Broadcast podaci nemaju potvrde te zbog toga primatelj tih podataka ne može znati da li je došlo do gubitka paketa u slanju. Zbog toga *broadcast* podaci zauzimaju najmanje RF spektra i upotrebljavaju najmanje energije za prijenos te se koriste tamo gdje gubitak podataka može biti toleriran (npr. mjerenje nekritičnih temperaturnih procesa) [4].

2.4.2 Acknowledged

U svakom trenutku između bilo koje dvosmjerne komunikacije dva čvora, bilo u *forward* ili *reverse* tipu, uređaj može poslati *acknowledged* tip paketa. Uređaj koji je primio taj paket će odgovoriti s drugim *acknowledged* paketom istome uređaju. Upravljačka jedinica uređaja koji je izvorno poslao poruku bit će tako obaviještena da je prijenos paketa bio uspješan. Kod ovoga tipa podataka ne postoji automatska retransmisija nepotvrđenih paketa [4].

Nadređeni uređaj može slati sve pakete kao *acknowledged* tip podatka ili koristiti kombinaciju *broadcast-acknowledged*, zavisno o primjerni. Mora se naglasiti da *acknowledged* paketi troše puno više RF spektra i više energije za slanje. S druge strane, ovakav tip paketa je idealan za prijenos kontrolnih podataka tako osiguravajući da su oba čvora u komunikaciji svjesna o stanju drugoga čvora [4].

2.4.3 Burst

Burst podaci osiguravaju mehanizam razmjene velike količine podataka između uređaja. Više paketa podataka se šalje unutar vremenskog perioda kanala. Ovi podaci se sinkroniziraju jedan naprema drugome (dok su do sada podaci se sinkronizirali prema periodu kanala) te se mogu postići brzine prijenosa podataka do 20 kbps [4].

Slično kao i kod *acknowledged* podaci, upravljačka jedinica uređaja koji šalje podatke bit će obaviještena da li je prijenos *burst* podataka bio uspješan. Uspjeh prijenosa podataka u *burst* modu odnosi se na cijeli skup podataka koji se šalje a ne na pojedinačne pakete. U slučaju da se samo jedan paket pogrešno pošalje dolazi do retransmisije cijeloga skupa podataka iz kojih dolazi taj paket. Ukoliko se takva retransmisija dogodi pet puta, upravljačka jedinica prijemnika bit će obaviještena da je došlo do pogreške u prijenosu [4].

Kod slanja *burst* podataka nema ograničenja vremenskog perioda unutar kojega se mogu slati podaci ali se mora napomenuti da ovakav tip prijenosa uzima presedan nad svim otvorenim kanalima između dva čvora u komunikaciji. Kada nadređeni uređaj prolongirano šalje *burst* podatke podređenome uređaju može doći do desinkronizacije svih ostalih podređenih uređaja

koji su spojeni na taj nadređeni uređaj. Također *burst* podaci mogu uzrokovati interferenciju komunikacije svih ostalih uređaja koji komuniciraju na istoj frekvenciji [4].

2.4.4 Advanced Burst

Advanced burst tipovi podataka su slični b*urst* tipovima podataka samo što se brzina prijenosa povećava do 60 kbps. Mora se naglasiti da se ovaj tip podataka može koristiti samo u kratkim vremenskim intervalima jer se inače može zagušiti cijeli kanal te se mogu desinkronizirati svi podređeni uređaji na kanalu [4].

2.5 Uparivanje ANT uređaja

Uparivanje dva uređaja uključuje uspostavljanje veze između dva čvora radi međusobne komunikacije. Veza može biti trajna (*eng. permanent*), polutrajna (*eng. semi-permanent*) ili prolazna (*eng. transitory*) [4].

Uparivanje uređaja započinje tako da podređeni uređaj pribavi identifikator kanala nadređenog uređaja. Kod trajne veze podređeni uređaj sprema identifikator kanala nadređenog uređaja u trajnu memoriju te se spaja na njega u svakoj sesiji. Polutrajna veza traje samo za vrijeme jedne sesije dok u prolaznim vezama komunikacija traje jako kratko (dok se ne pribave potrebni podaci).

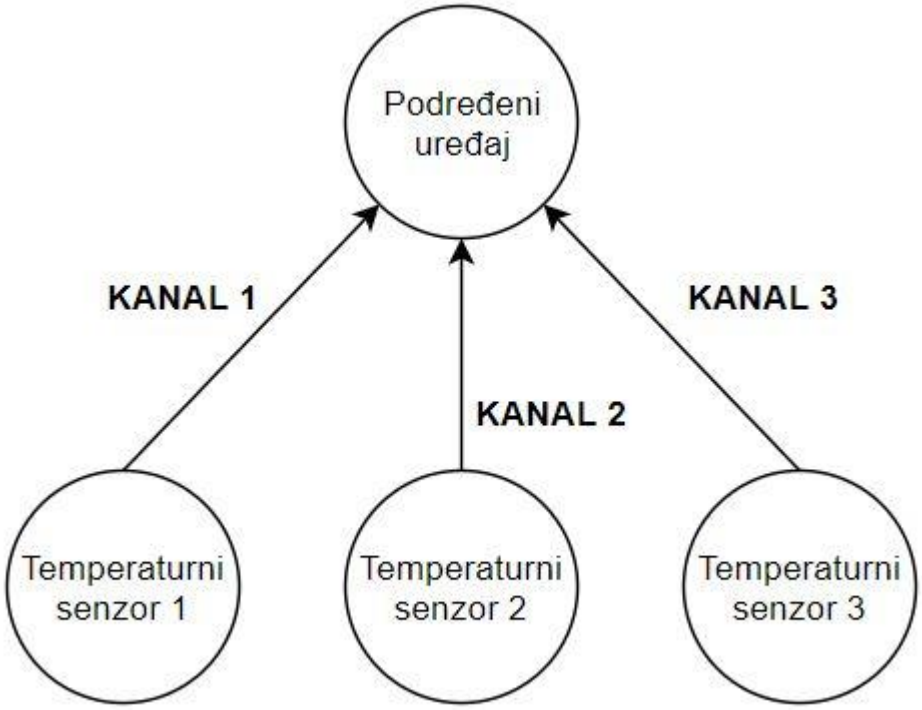
Kada nadređeni uređaj otvori kanal počinje slati *broadcast* poruke gdje u svakoj šalje jedinstveni identifikator kanala. Kada podređeni uređaj otvori kanal započinje s traženjem nadređenog uređaja s istim identifikatorom kanala. Identifikator kanala podređenog uređaja je spremljen u memoriji njegove upravljačke jedinice. Ako podređeni uređaj nema identifikator kanala (odnosno upravljačka jedinica nije poslala jedan) tada se spaja na onaj nadređeni uređaj koji zadovoljava neke druge kriterije. Primjerice, podređeni uređaj poznaje tip nadređenog uređaja na koji se želi spojiti, tada traži sve uređaje s tim tipom te se spaja na prvi uređaj na kojeg naiđe. Podređeni uređaj sprema podatke o identifikatoru kanala nadređenog uređaja nakon prve uspostave komunikacije.

Na slici 5. prikazana je jednostavna ANT mreža koja se sastoji od jednog podređenog uređaja i tri nadređena uređaja (temperaturne sonde). Podređeni uređaj želi ostvariti trajnu jednosmjernu vezu sa svim temperaturnim sondama. Da bi započelo uparivanje svaka temperaturna sonda mora kronološki odraditi sljedeće radnje:

1. Konfiguraciju kanala
2. Postavljanje identifikatora kanala
3. Otvaranje kanala za slanje podataka
4. Slanje podataka u svakom vremenskom periodu kanala

Podređeni uređaj mora se pripremiti za traženje svih nadređenih uređaja odgovarajućeg tipa. On obavlja sljedeće radnje:

1. Konfiguracija kanala
2. Postavljanje identifikatora kanala
3. Otvaranje kanala za prijem podataka
4. Traženje nadređenih uređaja



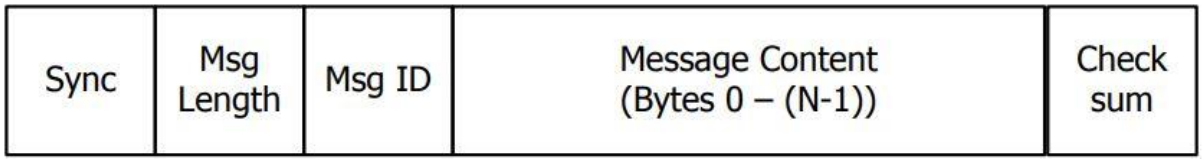
Slika 5. Primjer uparivanja u ANT mreži

Podređeni uređaj pronalazi temperaturne senzore koji imaju bit uparivanja unutar ANT poruke postavljen na '1'. Kada se uspostavi kanal, podređeni uređaj prosljeđuje svoj identifikator kanala temperaturnom senzoru gdje ga on trajno sprema u svoju memoriju. Na takav način se osigurava da se postupak uparivanja ostvaruje samo jednom kod trajne veze gdje se smanjuje opterećenje sustava [4].

Enkripcija kanala može biti postavljena na svim nezavisnim kanalima. Kada je enkripcija aktivna, ANT kanal koristi 128-bitni AES-CTR (*Advanced Encryption Standard – Counter Mode*) način rada. Mora se naglasiti da enkripcija nije aktivna u svim mrežama. Primjerice, ANT+ mreža zabranjuje korištenje enkripcije radi kompatibilnosti uređaja između različitih proizvođača. Kriptirani ANT+ kanali ne mogu koristiti ANT+ mrežni ključ ni radio frekvencije mreže [4].

2.6 ANT sučelje

Upravljačka jedinica i ANT uređaj komuniciraju preko jednostavnog serijskog sučelja. Razlikujemo dva formata poruka unutar ANT protokola: standardni i proširenI format. Na slici 6. prikazan je oblik standardne ANT poruke [4].



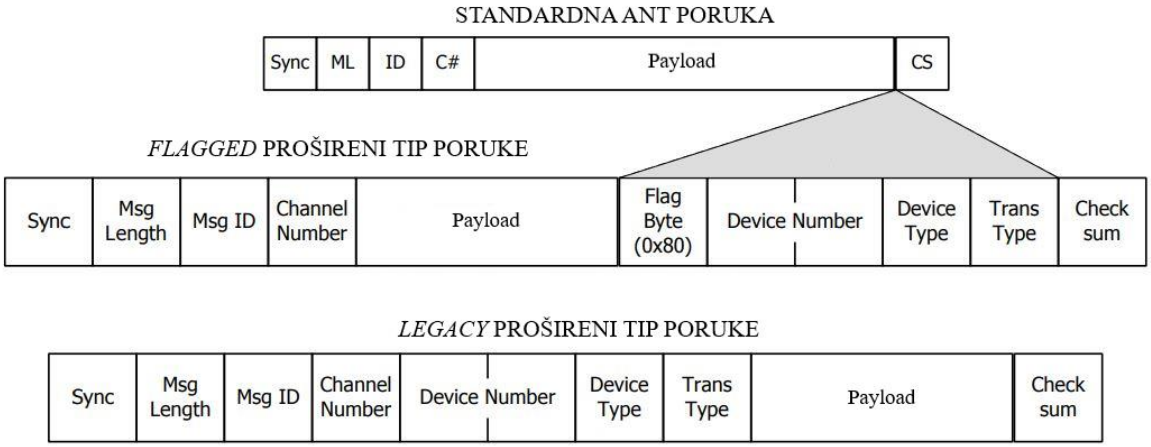
Slika 6. Standardna ANT poruka [4]

Svaka poruka započinje SYNC bajtom te završava CHECK SUM bajtom. Bajtovi se šalju od LSB prema MSB. U tablici 5. prikazana su objašnjena pojedinih polja poruke.

Tablica 5. Opisi pojedinih polja ANT standardne poruke

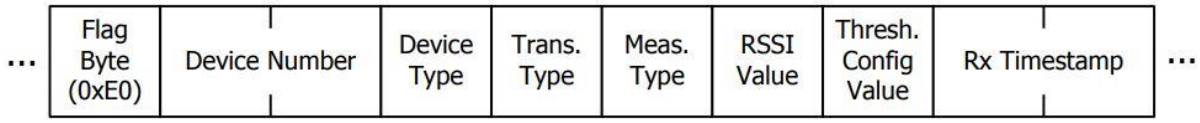
Redni broj	***Ime***	***Veličina***	***Opis***
0	SYNC	1 bajt	*Fiksirana vrijednost 10100100 ili 10100101*
1	MSG LENGTH	1 bajt	*Ukupni broj bajtova u poruci*
2	MSG ID	1 bajt	*Identifikator vrste poruke*
3.. N+2	PAYLOAD	N bajtova	*Sadržaj poruke do 8 bajtova*
N+3	CHECKSUM	1 bajt	*Vrijednost operacije ekskluzivno ILI nad svim prethodnim bajtovima*

Standardna ANT poruka može biti proširena na dva formata: *flagged* i *legacy*. Koji format se koristi ovisi o vrsti uređaja. Proširenim porukama mogu se poslati dodatne informacije. Na slici 7. prikazana je struktura proširenih ANT poruka [4].



Slika 7. Struktura proširenih ANT poruka [4]

Dodatna polja proširenih tipova poruka mogu biti birana po volji proizvođača a služe za prenošenje dodatnih podataka vezanih za samu komunikaciju ili uređaj. Veličina poruke sa svim uključenim porukama iznosi 24 bajta. Dodatna polja prikazana su na slici 7.



Slika 8. Dodatna polja proširenih poruka [4]

Dodatna polja proširenih poruka opisana su u tablici 7.

Tablica 6. Dodatna polja proširenih poruka [4]

Oznaka polja	***Veličina***	***Opis***
Flag Byte	1 bajt	*Oznaka tipa poruke i koliko poruka ima proširenih polja*
Device Number	2 bajta	*Broj uređaja koji šalje poruku*
Device Type	1 bajt	*Tip Uređaja*
Trans. Type	1 bajt	*Tip transmisije*
Meas. Type	1 bajt	*Mjerna jedinica za RSSI vrijednost*
RSSI Value	1 bajt	*Received Signal Strength Indicator – Vrijednost jačine signala u decibelima*
Thresh. Config Value	1 bajt	*Snaga signala tijekom traženja uređaja za komunikaciju*
Rx Timestamp	2 bajta	*16 bitno polje koje mjeri vrijeme unutarnjeg sata (32.768 kHz).*

3. ANT+ PROTOKOL I UREĐAJI

ANT+ je upravljana ANT mreža kojoj je svrha stvaranje jedinstvenog sustava unutar kojega uređaji različitih proizvođača mogu komunicirati. ANT+ se koristi u svrhe sporta i kućne medicine. Primjeri korištenja ANT+ mreže su:

- mjerenje otkucaja srca,
- brzina gibanja na biciklu,
- mjerenje snage bicikla,
- mjerenje mase,
- senzori opreme za tjelovježbu i
- temperaturni senzori.

Svi ANT+ uređaji imaju svoj profil. Svaki profil određuje funkciju pojedinog uređaja. Primjerice, *Environment* profil uvijek šalje iste podatke nezavisno o proizvođaču uređaja. Svi ANT+ uređaji su konfigurirani tako da ANT+ korisnička sučelja uvijek znaju koji tip podatka dobivaju [5].

Bitna razlika ANT+ i ANT mreže su postavke kanala. Primjer postavki kanala za podređeni ANT+ mreže prikazan je u tablici 7. ANT+ može koristiti bilo koji tip poruke za prijenos podataka (najčešće prošireni *flagged* tip) [5].

Tablica 7. Postavke podređenog kanala u ANT+ mreži [5]

Parametar	***Vrijednost***	***Opis***
Mrežni ključ	ANT+ mrežni ključ	ANT+ mrežni ključ se dijeli proizvođačima uređaja na službenim ANT+ internet stranicama
Frekvencija	57	ANT+ radio frekvencija
Broj uređaja	0	Traži sve nadređene uređaje
Tip prijenosa	0	Uparivanje
Tip uređaja	25	*Environment* senzor
Tip kanala	0x000	Podređeni kanal
Period kanala	65535	Takt slanja od 2 Hz

3.1 ANT+ profili

Svaki ANT+ uređaj posjeduje profil koji sadrži određena pravila mreže te podatke tog profila može poslati unutar poruke. Profili se sastoje od stranica (*eng. pages*) te unutar svakoga profila možemo pronaći više različitih stranica. Stranice su označene brojevima a broj stranica koji se šalju određen je samim profilom koji se koristi. Bitno je naglasiti da se podaci senzora šalju preko stranica te je dopušteno slanje ostalih poruka unutar ANT mreže nezavisno o stranicama (kao dijagnostički podaci ili podaci potvrde).

ANT+ profili su sljedeći:

- ***Bicycle Power*** – mjerenje izlazne snage biciklista,
- ***Controls*** – kontrola glazbenih elektroničkih uređaja,
- ***Geocache*** – mjerenje geografske lokacije,
- ***Multi Sport Speed & Distance*** – mjerenje brzine i prijeđene udaljenosti,
- ***Stride Based Speed & Distance*** -mjerenje brzine i prijeđene udaljenosti hodom,
- ***Bicycle Speed & Cadence*** -mjerenje brzine i broja koraka,
- ***Enviroment*** – mjerenje temperature,
- ***Hearth Rate Monitor*** – mjerenje otkucaja srca,
- ***Muscle Oxygen Monitor*** – mjerenje koncentracije kisika u mišićima,
- ***Sync*** – spremanje podataka za kasniju analizu,
- ***Blood Pressure*** – mjerenje krvnog tlaka,
- ***Fitness Equipment*** – profil za komuniciranje s opremom za tjelovježbu,
- ***Light Electric Vehicle*** – upravljanje i nadzor električnih vozila malih snaga,
- ***Racquet*** – senzor za mjerenje parametara teniskih reketa,
- ***Weight Scale*** – mjerenje mase,
- ***Extended Display*** – komunikacija prema udaljenom zaslonu,
- ***Suspension*** – mjerenje amortizirajućih sila u vozilima,
- ***Dropper Seatpost*** – nadzor i podešavanje sjedišta bicikla,
- ***Tracker*** – pronalaženje uređaja pomoću GPS tehnologije [5].

Svaki podređeni uređaj pri slanju podataka šalje određene stranice koje nadređeni uređaj poznaje (spremljene u memoriji nadređenog uređaja). Primjerice, *Environment* profil može slati tri tipa stranica: podatkovne stranice (temperaturni podaci), stranice o proizvođaču uređaja te stranice o tehničkim specifikacijama samoga podređenog uređaja. Broj stranice koji se šalje određen je prvim bajtom unutar samih korisničkih podataka. Sve stranice prikazane su u tablici 8. Mora se naglasiti da se stranice šalju po potrebi tako da stranice o tipu uređaja i proizvođaču će biti poslane svakih 80 ciklusa (zavisno o profilu) dok stranice o podacima će biti slane svakim ciklusom. Nadređeni uređaj može zatražiti bilo koju stranicu od podređenog uređaja po potrebi.

Tablica 8. Stranice ANT+ profila [5]

Broj stranice	***Opis***
0x00 – 0x3F	*Stranice proizvođača ANT+ Alliance (gore navedeni profili)*
0x40 – 0x5D	*Opće stranice gdje su prikazane informacije o uređaju, proizvođaču itd.*
0x5E – 0x6F	*Stranice rezervirane za buduću upotrebu.*
0x70 – 0x7F	*Stranice rezervirane za proizvođače koji ne koriste prvi bit prvog bajta unutar korisničkog djela ANT+ poruke.*
0x80 – 0xDF	*Stranice rezervirane za buduću upotrebu.*
0xE0 – 0xFF	*Stranice u ovom rasponu mogu biti korištene od proizvođača uređaja na bilo koji način.*

3.2 ANT+ uređaji

ANT+ uređaji dolaze u mnogo oblika, funkcijskih izvedbi i tehnologija. Najčešći princip izvedbe je podređeni uređaj u nosivome obliku (narukvica ili ogrlica) dok se za nadređeni uređaj koristi ANT+ kompatibilni pametni telefon ili računalo s kompatibilnim ANT+ USB uređajem. U okviru ovoga rada bit će predstavljeni ANT+ senzori otkucaja srca. Senzori na kojima se testirao ANT+ protokol u okvirima ovoga rada prikazani su na slici 9.



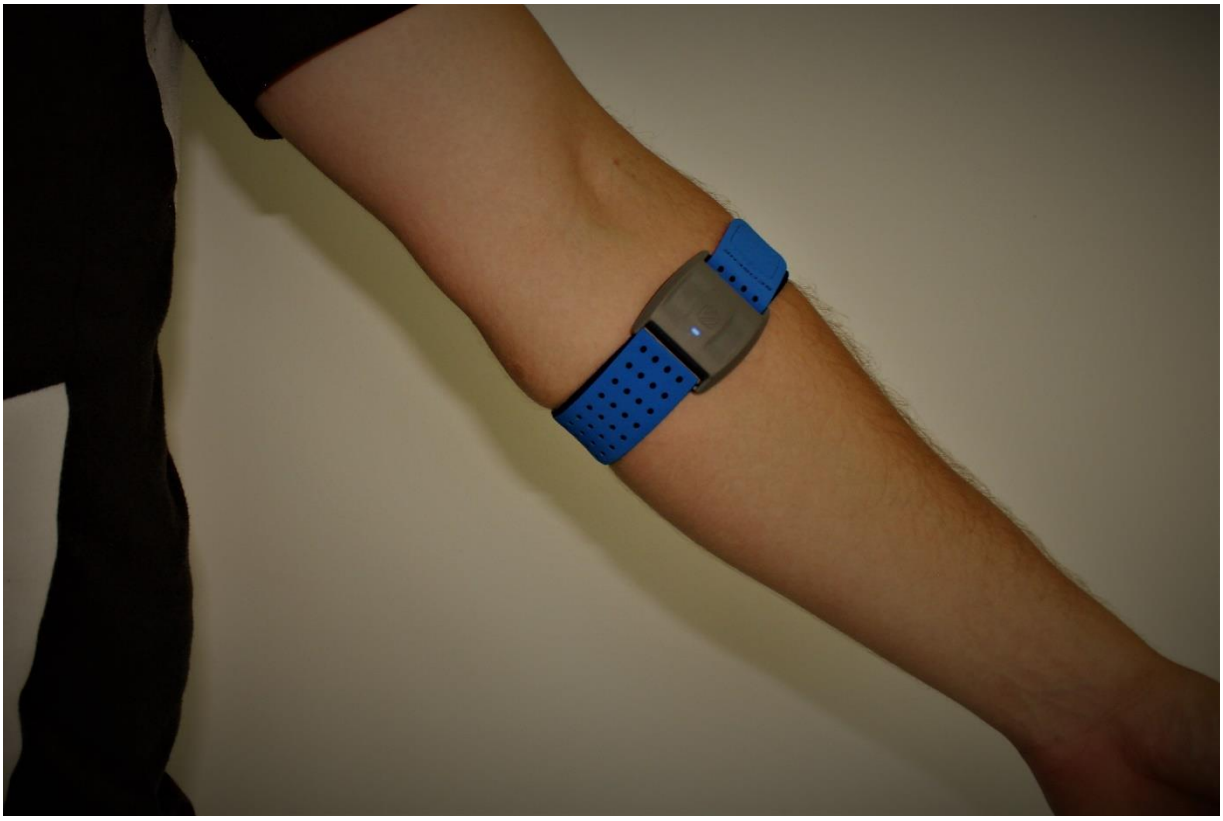
Slika 9. ANT+ senzorski uređaji za mjerenje otkucaja srca.

Svi uređaji dolaze s prikladnom opremom za nošenje na tijelu. Uređaji se mogu postaviti oko gornjeg dijela tijela kraj srca ili ispod podlaktice. Modeli proizvođača „Garmin“ i „Wahoo“ (slika 11) započinju s radom kada ostvare kontakt s kožom dok model proizvođača „Scosche“ (slika 12) započinje s radom kada se pritisne tipkalo. Svi uređaji su otporni na znoj i vlagu te zaštićeni od udara unutar posebnog kućišta. Uređaj proizvođača „Garmin“ posebno je namijenjen za mjerenje otkucaja srca plivaća te je dodatno zaštićen vodonepropusnim pokrovom.

Za sve uređaje dostupna je prilagođena mobilna aplikacija proizvođača u kojoj je predstavljen prikaz trenutnih mjernih vrijednosti te dodatne mogućnosti kao praćenje lokacije i trajanje vježbe (ostvareno preko funkcija pametnog mobitela). Izgled korisničkog sučelja aplikacije proizvođača „Garmin“ prikazan je na slici 10. Korisnik nije dužan koristiti mobilnu aplikaciju proizvođača, već zbog kompatibilnosti samoga protokola može koristit bilo koju ANT+ aplikaciju.



Slika 10. Garmin Fit mobilna aplikacija



Slika 11. Mjerač otkucaja srca tvrtke „Scosche“



Slika 12. Mjerač otkucaja srca tvrtke „Wahoo“

3.3 Analiza ANT+ Hearth Rate profila

Podređeni uređaji svakog ANT+ profila rade po posebnim pravilima. Stranice podataka koje će se poslati ovise o funkciji te izvedbi svakoga pojedinačnoga uređaja. Naravno, svaki uređaj mora poštovati pravila koja su postavljena unutar samih profila. U ANT+ *Hearth Rate* profilu razlikujemo dva skupa stranica koje se transmitiraju: pozadinske stranice (*eng. Data Pages*) i glavne stranice (*eng. Main Pages*). ANT+ *Hearth rate* profil podržava pet različitih brojeva stranica [5].

Pozadinske stranice su numerirane brojevima '1','2' i '3' (oznaka se nalazi unutar *payload*-a prvoga bajta profila). Ove stranice daju informacije o vremenskom trajanju rada podređenog uređaja s nadređenim uređajem te podatke o proizvođaču uređaja. Pozadinske stranice se šalju svako 65 poruka ili minimalno jedna stranica svako 64.03 sekunde (mogu se i kontinuirano slati – ovisi o proizvođaču uređaja) [5].

Glavne stranice uključuju stranice '0' i '4'. One se kontinuirano šalju od podređenog prema nadređenome uređaju uz iznimku 65-te poruke koja mora biti pozadinska stranica. U nastavku opisane su pojedine stranice *Hearth Rate* profila.

3.3.1 Formati stranica Hearth Rate profila

Sve stranice su definirane kao paketi veličine 8 bajtova. Posebna pravila se primjenjuju na ovaj profil a to su:

1. Najznačajniji bit prvoga bajta poruke uvijek je rezerviran za *toggle* bit. Svako četiri poruke transmiter mora promijeniti vrijednost ovoga bita što dopušta prijemniku da prima podatke o otkucajima srca po manjim frekvencijama a da je još sposoban razaznati da je došlo do promjene formata stranice. Drugim riječima, tek nakon što se promjeni vrijednost *toggle* bita mogu se obrađivati novi tipovi stranica.
2. Bajtovi od 1 do 3 su jedini kojima se funkcija mijenja zavisno o broju stranice koja se šalje.
3. Bajtovi 4 do 7 imaju istu definiciju za svaku stranicu [5].

U tablici 9. prikazana je struktura stranice '0'. Ova stranica omogućava slanje vremenskog trenutka posljednjeg otkucaja srca, brojač otkucaja srca te izračunati broj otkucaja srca.

Tablica 9. ANT+ *Hearth Rate* profil – stranica '0' [5]

Redni broj	***Opis***	***Duljina***	***Vrijednost***	***Mjerne jed.***	***Rollover***
0	*Toggle* bit	1 bit	*Transmiter mora mijenjati vrijednost ovoga bita svako četiri poruke. Prijemni uređaj ne može procesirati ostatak poruke dok ovaj bit nije promijenjen iz 0 u 1.*	-	-
0	Broj stranice	7 bitova	*Broj stranice koji se šalje*	-	-
1	Rezervirani bajt	3 bajta	*Transmiter postavlja ovu vrijednost u 0xFF. Prijemnik ne čita ove bajtove.*	-	-
2	Rezervirani bajt				
3	Rezervirani bajt				
4	Otkucaji srca LSB	2 bajta	*Vrijeme koje je proteklo od trenutka kada je zabilježen zadnji otkucaj srca.*	1/1024 s	64s
5	Otkucaji srca MSB				
6	Takt otkucaja	1 bajt	*Brojač kojemu se inkrementima vrijednost sa svakim otkucajem*	-	225
7	Izračunati otkucaji	1 bajt	*Izračunati otkucaji srca*	bpm	-

Stranice '1','2','3' i '4' prikazane su u tablicama 11, 12, 13 i 14. Te stranice se razlikuju od stranice 0 samo po bajtovima 1,2 i 3 te su tako u tablicama i navedene.

Tablica 10. ANT+ *Hearth Rate* profil – stranica '1' [5]

Bajt	***Opis***	***Vrijednost***
1	Vrijeme rada – bitovi 0 do 7	Inkrementira se svako dvije sekunde i resetira kada nestane napona napajanja uređaja.
2	Vrijeme rada – bitovi 8 do 15	
3	Vrijeme rada – bitovi 16 do 23	

Tablica 11. ANT+ *Hearth Rate* profil – stranica '2' [5]

Bajt	***Opis***	***Vrijednost***
1	Identifikator proizvođača	ID proizvođača dodijeljen od strane ANT+
2	Serijski broj LSB	Serijski broj uređaja
3	Serijski broj MSB	

Tablica 12. ANT+ *Hearth Rate* profil – stranica '3' [5]

Bajt	***Opis***	***Vrijednost***
1	Verzija sklopovlja	Postavlja se od strane proizvođača.
2	Verzija programa	
3	Broj modela uređaja	

Tablica 13. ANT+ *Hearth Rate* profil – stranica '4' [5]

Bajt	***Opis***	***Vrijednost***
1	Bajt proizvođača	Prijemni uređaj neće obrađivati ovo polje.
2	Prethodno vrijeme otkucaja srca LSB	Ovi bajtovi predstavljaju prethodno validni vremenski trenutak otkucaja srca.
3	Prethodno vrijeme otkucaja srca MSB	

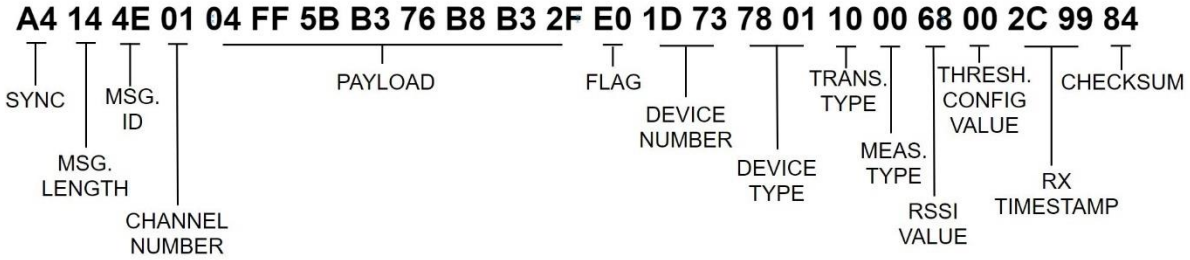
3.3.2 Analiza ANT+ protokola preko USB sučelja

Radi utvrđivanja valjanosti ANT+ *Hearth Rate* profila, u okviru ovoga rada izvršen je nadzor serijskog sučelja na nadređenom uređaju. Na osobno računalo postavljen je ANT+ kompatibilni USB uređaj te je ostvarena komunikacija s mjeračem otkucaja srca proizvođača „Sosche“. Promet je stavljen pod nadzor pomoću programa „SysNucleus USBTrace“. Primjer očitane komunikacije prikazan je na slici 13.

52.100975	BULK_OR_INTERRUPT_TR...	IN	81	\Device\USBPDO-3	0x27434B...	STATUS_SUCCESS	A4 14 4E 01 04 FF 40 AE 5B B3 B2 2F E0 1D...	24
52.101022	BULK_OR_INTERRUPT_TR...	OUT	81	\Device\USBPDO-12	0x27434B...	STATUS_SUCCESS		0
52.101027	BULK_OR_INTERRUPT_TR...	OUT	81	\Device\USBPDO-3	0x27434B...	STATUS_SUCCESS		0
52.101046	BULK_OR_INTERRUPT_TR...	IN	81	\Device\USBPDO-3	0x27434B...	STATUS_PENDING		0
52.346971	BULK_OR_INTERRUPT_TR...	IN	81	\Device\USBPDO-3	0x27434B...	STATUS_SUCCESS	A4 14 4E 01 04 FF 40 AE 5B B3 B2 2F E0 1D...	24
52.347005	BULK_OR_INTERRUPT_TR...	OUT	81	\Device\USBPDO-12	0x27434B...	STATUS_SUCCESS		0
52.347009	BULK_OR_INTERRUPT_TR...	OUT	81	\Device\USBPDO-3	0x27434B...	STATUS_SUCCESS		0
52.347024	BULK_OR_INTERRUPT_TR...	IN	81	\Device\USBPDO-3	0x27434B...	STATUS_PENDING		0
52.593975	BULK_OR_INTERRUPT_TR...	IN	81	\Device\USBPDO-3	0x27434B...	STATUS_SUCCESS	A4 14 4E 01 04 FF 5B B3 76 B8 B3 2F E0 1D...	24
52.594012	BULK_OR_INTERRUPT_TR...	OUT	81	\Device\USBPDO-12	0x27434B...	STATUS_SUCCESS		0
52.594015	BULK_OR_INTERRUPT_TR...	OUT	81	\Device\USBPDO-3	0x27434B...	STATUS_SUCCESS		0
52.594025	BULK_OR_INTERRUPT_TR...	IN	81	\Device\USBPDO-3	0x27434B...	STATUS_PENDING		0
52.839988	BULK_OR_INTERRUPT_TR...	IN	81	\Device\USBPDO-3	0x27434B...	STATUS_SUCCESS	A4 14 4E 01 04 FF 5B B3 76 B8 B3 2F E0 1D...	24

Slika 13. Nadzor USB sučelja na ANT+ protokolu

Sve podatkovne poruke su duljine 24 bajta te dolaze u vremenskim intervalima od 0.25 sekundi (4 Hz). Poruke se čitaju u heksadekadskom obliku te je primjer prikazan na slici 14.



Slika 14. Očitana ANT+ poruka

Može se primijetiti da poruke odgovaraju proširenom *legacy* formatu ANT protokola te se iz poruke može iščitati:

- Broj uređaja = 29 449
- Tip podataka = 4E (što odgovara *broadcast* podacima)
- Broj kanala = 01

Payload prikazane poruke ima vrijednost *04 FF 5B B3 76 B8 B3 2F*. Iz prvoga bajta može se primijetiti da se radi o stranici '4' dok se iz zadnjega bajta može primijetiti su izmjereni otkucaji srca 47 bpm.

4. IMPLEMENTACIJA ANT+ PROTOKOLA POMOĆU NODE.JS

Implementacija ANT+ protokola može se izvršiti na više načina. ANT+ Alliance nudi programska rješenja kroz C++ programski jezik unutar „Microsoft Visual Studio okruženja“. U slučaju kada bi htjeli napraviti Internet aplikaciju, gdje korisnici mogu spremati podatke na udaljeni server, morali bi se poslužiti drugim alatima. U tu svrhu upotrijebiti će se Node.js izvršno okruženje.

4.1 Node.js

Node.js je izvršno okruženje koje izvršava JavaScript kod izvan internet pretraživača. Ono omogućava korisnicima izvršavanje JavaScript programa na strani servera dok se obrađeni podaci šalju korisniku preko internet pretraživača. Ovime su reducirani zahtjevi prema performansama korisnikovog računala te kvaliteta usluge najvećim djelom ovisi o kvaliteti samoga servera [7].



Slika 15. Node.js logo [7]

4.2 Instalacija ANT+ Node.js modula

Implementacija ANT+ protokola u okviru ovoga rada treba omogućiti vizualizaciju mjerenja otkucaja srca preko više uređaja i spremanje podataka mjerenja u bazu podataka. Programski kod je napravljen unutar sučelja *Microsoft Visual Studio Code* koji posjeduje dodatke za Node.js izvršno okruženje.

Node.js radi na principu modula. Moduli se ponašaju slično kao i biblioteke unutar C++ programa te daju pristup već unaprijed definiranim klasama i funkcijama. Moduli se nalaze na različitim serverima ili mogu biti spremljeni na korisnikovo računalo. Svaki modul može unutar sebe imati druge dodatne module tako da instalacija modula podrazumijeva instalaciju svih programskih podskupova od kojih se modul sastoji.

ANT+ Node.js modul je napravljen za integraciju ANT+ protokola preko USB sučelja. Instalacija se izvršava upisom sljedeće linije koda unutar *Windows PowerShell* sučelja unutar *Visual Studio Code* programa:

```
npm install ant-plus
```

Modul se uključuje sljedećom naredbom unutar samoga *JavaScript* programa:

```
const Ant = require('ant-plus');
```

Nakon toga potrebno je definirati varijable USB sučelja i senzora koji se koriste:

```
// Inicijaliziramo ANT USB uređaj
let stick = new Ant.GarminStick2;
// Inicijaliziramo Heart Rate senzor
let sensor = new Ant.HeartRateScanner(stick);
```

U trenutku kada je USB uređaj spreman za rad mora se pokrenuti traženje senzora:

```
stick.on('startup', function () {
  sensor.scan();
  console.log("Scanning...");
});
```

Nakon što se pronađu senzori potrebno je postaviti sve senzore na određene kanale. Funkcija *senzor.attach* prima dva parametra: identifikator kanala i identifikator uređaja. Ako se u *channel_id* postavi parametar '0', uređaj se spaja na prvi senzor kojega skenira. U kodu ispod prikazano je postavljanje uređaja '42763' :

```
stick.on('startup', function () {
  sensor.attach(0, 42763);
});
```

Nakon primarnoga postavljanja mogu se očitati vrijednosti koje se šalju s uređaja. Zavisno o tipu i vrsti uređaja mogu se čitati različiti profili. U sljedećem kodu prikazano je čitanje otkucaja srca:

```
sensor.on('hbData', function (data) {
  console.log(data.DeviceID, data.ComputedHeartRate);
});
```

4.3 Vizualizacija otkucaja srca

Zadatak ovoga projekta je vizualizacija otkucaja srca s više ANT+ uređaja te spremanje mjerenja senzora u bazu podataka. Vizualizacija mora biti *RealTime* te baza podatka mora biti konstruirana tako da je jednostavna za čitanje i upisivanje.

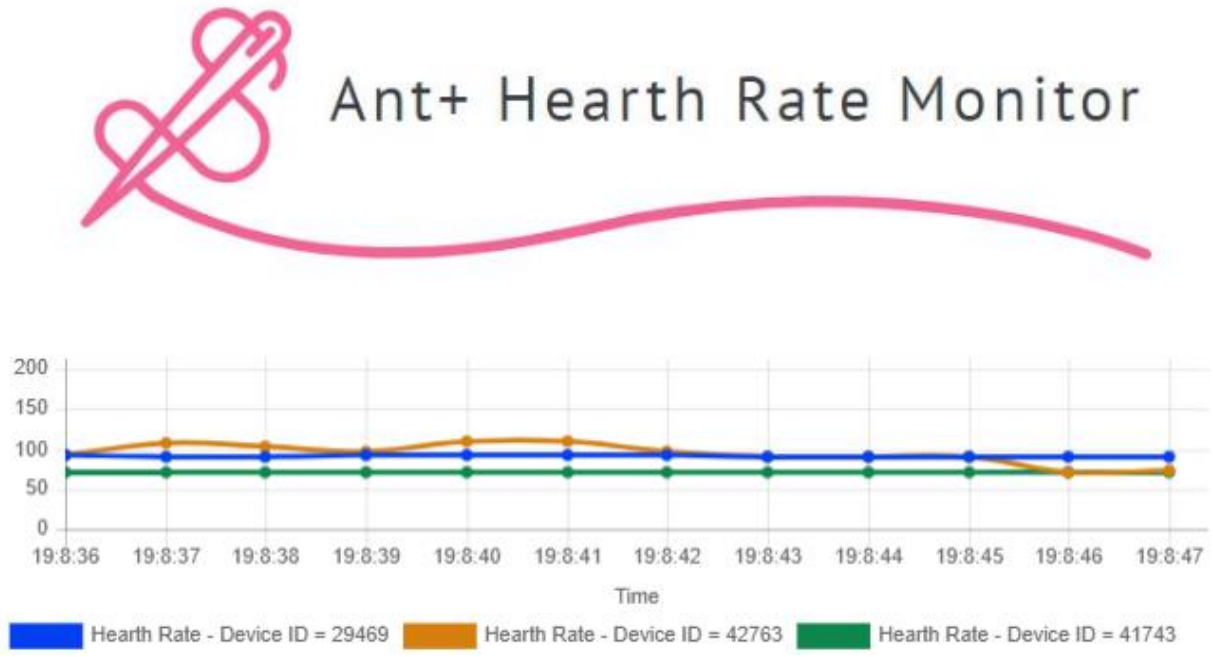
Vizualizacija podataka ostvarena je unutar Internet pretraživača primjenom grafičkog alata Charts.js. Chart.js je JavaScript alat za izradu grafikona s mnoštvom mogućnosti za dizajn i interakciju na samim grafikonima.

Spremanje podataka bilo je prvotno zamišljeno kroz popularne sustave kao *MySQL* ili *MongoDB*. Zbog toga što se unutar ovoga zadatka spremaju samo podaci o trenutku mjerenja i trenutni otkucaji srca, nije bilo potrebno koristiti baze podataka koje unose dodatna opterećenja u sustav. Odlučeno je da će se za svaki senzor posebno raditi tekstualna datoteka u JSON formatu.

JSON (*JavaScript Object Notation*) je format otvorenog standarda koji koristi podatkovne objekte za spremanje podataka. Objekti se sastoje od imena i vrijednosti. Podaci senzora će se spremati u sljedećem JSON obliku:

[{"hr":10,"time":"start"},{"hr":0,"time":"12:36:13"},{"hr":0,"time":"12:36:14"
},{"hr":0,"time":"12:36:15"},{"hr":0,"time":"12:36:16"},{"hr":0,"time":"12:36:
17"},{"hr":0,"time":"12:36:19"},{"hr":0,"time":"12:36:20"},{"hr":0,"time":"12:
36:22"},{"hr":0. . . . . .

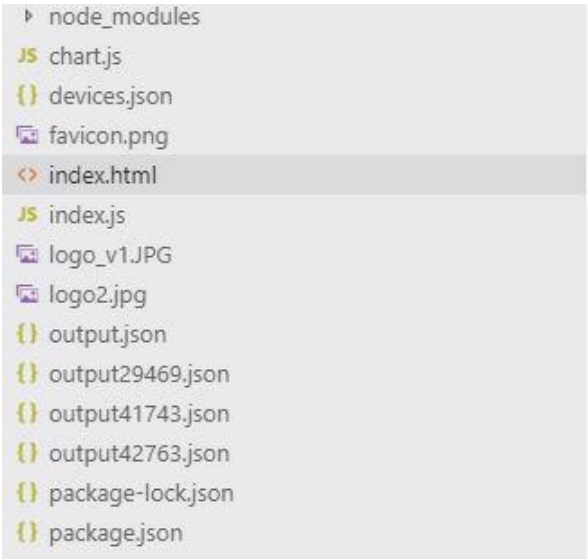
JSON datoteke svakog senzora se pišu i čitaju svake sekunde. Na slici 16. prikazan je izgled vizualizacijskog sučelja.



Slika 16. Izgled vizualizacijskog sučelja ANT+ *Hearth Rate* monitora

4.3 Programski kod

Struktura programa postavljena je tako da se programi *chart.js* i *index.js* pozivaju unutar *index.html* datoteke. Program *chart.js* služi za vizualizaciju podataka dok program *index.js* služi za prikupljanje i spremanje mjerenja senzora u bazu podataka. Datotečna struktura programa prikazana je na slici 17. Program je nadopunjen s dodatnim datotekama kao što su slike i dodatne postavke okruženja. JSON datoteke se spremaju unutar glavnoga direktorija.



Slika 17. Datotečna struktura programa

4.3.1 Index.html datoteka

Uz standardne dijelove HTML zaglavlja u ovoj datoteci se dodatno pozivaju JavaScript program chart.js te se uključuju biblioteke za obradu JSON datoteka i biblioteke za rad sa Chart.js skriptama:

```
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"
type="text/javascript"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.1.4/Chart.min.
js">
</script>
<script  src = "chart.js">
```

4.3.2 Chart.js datoteka

Unutar chart.js datoteke nalaze se funkcije za prikaz podataka na grafikonu te funkcije za čitanje JSON datoteka. Program u radu stvara dva tipa JSON datoteka a to su:

1. Podaci o uređajima – „devices.json“

Za ostvarivanje komunikacije između ANT+ uređaja i računala potrebno je da računalo poznaje identifikator uređaja. Podaci o identifikatoru uređaja mogu se pronaći na stranicama proizvođača ili se skeniraju putem aplikacije. Za tri navedena uređaja iz poglavlja 3.2 sadržaj JSON datoteke je sljedeći:

```
[{"device":29469},{"device":42763},{"device":41743}]
```

Podaci ove datoteke mogu se unositi ručno ili preko neke druge aplikacije. Broj uređaja zapisanih unutar datoteke određen je brojem kanala koji USB ANT+ uređaj podržava jer se svaki uređaj spaja na posebni kanal.

2. Mjerenja senzora – „output.json“

Senzorski podaci se spremaju kao uređeni par vremena mjerenja i vrijednosti mjerenja (otkucaji srca). Naziv JSON datoteke određen je identifikatorom uređaja pa primjerice sadržaj datoteke „output29469.json“ bio bi sljedeći:

```
{"hr":10,"time":"start"},{"hr":0,"time":"12:33:23"},{"hr":0,"time":"12:33:24"},{"hr":70,"time":"12:33:25"},{"hr":70,"time":"12:33:26"},{"hr":70,"time":"12:33:27"},...
```

Vrijeme mjerenja se zapisuje u formatu *h:m:s* izuzev prvoga mjerenja koje je označeno s vrijednošću *start*. Time se označava početak rada senzora.

Chart.js datoteka se sastoji od jedne globalne varijable *HRChart* i dvije funkcije: *getDevice()* i *getData()*. Varijabla *HRChart* predstavlja definiciju samoga grafikona na kojemu će se

vizualizirati podaci mjerenja. Osim postavki za dizajn i funkcionalnosti grafikona potrebno je definirati vrijednosti apscise i ordinate samoga grafikona preko *data* varijable. *Labels* varijabla predstavlja oznake na apscisi (vremenski trenutci mjerenja) dok *datasets* predstavlja vrijednosti mjerenja senzora. Kod inicijalizacije, vrijednosti *data* varijable ostavljene su prazne:

```
data: {
    labels: [' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',],
    datasets: []
  }
```

Funkcijom *getDevice()* se očitava sadržaj datoteke *devices.json* tako da se dinamički dodaju novi skupovi podataka unutar varijable *HRChart*. Također se za svaki novi uređaj dodaje legenda ispod grafikona radi lakšeg prepoznavanje uređaja. Ova funkcija je periodična, te se izvršava svake sekunde.

Funkcijom *getData()* se čitaju mjerenja senzora te iz JSON datoteka. Podaci se zapisuju u varijablu *HRChart*. Ova funkcija se također izvršava svake sekunde. Iako se mjerenja senzora uzimaju u prosjeku svako 0.25 sekundi, ovdje se zapisuje svako četvrto mjerenje da bi se reduciralo zagušenje podataka te povećala preglednost čitanja podataka.

4.3.3 Index.js datoteka

Unutar index.js datoteke, osim uključivanja node.js modula i postavki servera, očitavaju se podaci uređaja koji su spojeni na ANT+ USB uređaj. Prvo se mora ostvariti veza između svakoga ANT+ uređaja i računala funkcijom *sensor.attach()* i to za više senzora na sljedeći način:

```
stick.on('startup', function () {
    for(var j = 0; j < jsonDevice.length; j++){
      sensor[j].attach(j+1, jsonDevice[j].device);
    }});
```

Funkcijom *sensor.attach()* se ostvaruje dvosmjerna veza između senzora i računala tako da svaki uređaj komunicira preko zasebnoga kanala. Ostvarena veza je aktivna sve dok korisnik ne prekine rad programa.

Nakon definicije korištenih uređaja potrebno je očitati mjerenja uređaja te podatke spremiti u JSON datoteku. U tu svrhu koristi se funkcija *senzor.on()* i to na sljedeći način:

```
for(var j = 0; j < jsonDevice.length; j++){
  sensor[j].on('hbData', function (data) {
    var today = new Date();
    var tim = today.getHours() + ":" + today.getMinutes() + ":" +
    today.getSeconds();

    appendObject({"hr":data.ComputedHeartRate,time":tim},'./output'+
    data.DeviceID+'.json');
});}
```

Ova funkcija posebno čita svaki senzor te zapisuje podatke u pripadajuću JSON datoteku. Vrijeme očitanja senzora uzima se unutar samoga programa u trenutku mjerenja te se skupa s vrijednosti mjerenja zapisuje u datoteku. Potpuno objašnjenje programa sa uputama za instalaciju sustava vizualizacije može se pregledati u prilogu.

6. ZAKLJUČAK

Rezultati analize ANT protokola pokazali su da je ANT robusna i jednostavna mreža. Upravljanje i nadzor ANT uređaja pokazao se kao jednostavan proces gdje je zahvaljujući doprinosu proizvođača jako olakšana integracija ANT protokola unutar računalnih okruženja.

Najveća prednost ANT protokola je u tome što su ANT uređaji jako energetski efikasni. To se i posebno pokazalo u izradi ovoga rada gdje baterije uređaja nisu ni jednom trebale biti napunjene u vremenu korištenja od tri mjeseca. Osim primjene ANT uređaja unutar sportske opreme, nudi se i prilika za korištenje senzora unutar nekritičnih industrijskih procesa zbog jake otpornosti ANT protokola na interferencije i mogućnosti integracije raznih mrežnih topologija.

Mane ANT protokola nisu toliko očite ali problem se može naći u brzini prijenosa podataka od 60 kbps što je dovoljno za prijenos senzorskih podataka, ali ne i glazbenih i video formata. Može se reći da je to dijelom propuštena prilika ANT protokola, jer bi integracija takvog sustava privukla više korisnika, posebno u segmentima sportske opreme. Drugu manu predstavlja mali izbor senzorskih modula i stranica podataka gdje se primjerice ANT uređaji još uvijek oslanjaju na funkcije računalnih sustava za čitanje GPS podataka.

U budućnosti se zasigurno može očekivati nastavak razvijanja ANT protokola. BLE još uvijek dominira tržištem niskoenergetskih bežičnih senzora dok ANT protokol tek pronalazi mjesto među korisnicima.

LITERATURA

[1] Wikipedia, https://en.wikipedia.org/wiki/ANT_(network), pristupljeno: lipanj, 2019

[2] Developing ANT in Android, https://www.thisisant.com/developer/ant/ant-in-android, pristupljeno: lipanj, 2019

[3] This Is ANT, https://www.thisisant.com/business/go-ant/levels-and-benefits , pristupljeno: lipanj, 2019

[4] Dynastream Innovations Inc., *ANT Message Protocol and Usage*, 5. izdanje, 2018

[5] Dynastream Innovations Inc., *ANT+ Common Pages*, 3. izdanje, 2018

[6] Dynastream Innovations Inc., *ANT+ Device Profile, HEARTH RATE MONITOR*, 1. izdanje, 2018

[7] Wikipedia, https://en.wikipedia.org/wiki/Node.js, pristupljeno: lipanj 2019

POPIS SLIKA

Slika 1. ANT bežična senzorska mreža [2] 4
Slika 2. Jednostavna ANT mreža 6
Slika 3. Komunikacija između nadređenog i podređenog uređaja 8
Slika 4. Primjer uspostave kanala 15
Slika 5. Primjer uparivanja u ANT mreži 20
Slika 6. Standardna ANT poruka [4] 22
Slika 7. Struktura proširenih ANT poruka [4] 23
Slika 8. Dodatna polja proširenih poruka [4] 23
Slika 9. ANT+ senzorski uređaji za mjerenje otkucaja srca. 28
Slika 10. Garmin Fit mobilna aplikacija 29
Slika 11. Mjerač otkucaja srca tvrtke „Scosche“ 29
Slika 12. Mjerač otkucaja srca tvrtke „Wahoo“ 30
Slika 13. Nadzor USB sučelja na ANT+ protokolu 33
Slika 14. Očitana ANT+ poruka 33
Slika 15. Node.js logo [7] 34
Slika 16. Izgled vizualizacijskog sučelja ANT+ *Hearth Rate* monitora 37
Slika 17. Datotečna struktura programa 38

POPIS TABLICA

Tablica 1. Tipovi ANT kanala [4] 9
Tablica 2. Bitovi polja tipa komunikacije [3] 11
Tablica 3. Primjer konfiguracije kanala 13
Tablica 4. Tipovi podataka u ANT protokolu 16
Tablica 5. Opisi pojedinih polja ANT standardne poruke 22
Tablica 6. Dodatna polja proširenih poruka [4] 24
Tablica 7. Postavke podređenog kanala u ANT+ mreži [5] 25
Tablica 8. Stranice ANT+ profila [5] 27
Tablica 9. ANT+ *Hearth Rate* profil – stranica '0' [5] 31
Tablica 10. ANT+ *Hearth Rate* profil – stranica '1' [5] 32
Tablica 11. ANT+ *Hearth Rate* profil – stranica '2' [5] 32
Tablica 12. ANT+ *Hearth Rate* profil – stranica '3' [5] 32
Tablica 13. ANT+ *Hearth Rate* profil – stranica '4' [5] 32

PRILOG

U ovome prilogu nalaze se upute za instalaciju vizualizacijskog sučelja *ANT+ HR Monitor* koji je napravljen u okviru ovoga rada kao i objašnjenja napravljenog programskog koda. Programski kod objašnjen je preko komentara unutar samoga koda. Sučelje je realizirano u *Windows 10* operativnome sustavu.

1. INSTALACIJA PROGRAMSKIH OKRUŽENJA

Prvi korak zahtjeva instalaciju *Visual Studio Code* okruženja koji se može preuzeti na sljedećoj poveznici:

https://code.visualstudio.com/download

Izbor okruženja nije bitan te se cijeli projekt može pokrenuti samo putem *Windows Command Prompt*-a. Rad u *Visual Studio Code* znatno olakšava postupak izrade i rada s programom. Nakon instalacije *Visual Studio Code* okruženja potrebno je instalirati *Node.js* koji se može preuzeti na sljedećoj poveznici:

https://nodejs.org/en/

Unutar *Visual Studio Code* terminala potrebno je provjeriti da li je pravilno instaliran *Node.js* naredbom:

node -v

ANT+ USB uređaj zahtjeva posebne upravljačke programe (drivers) koji se mogu instalirati preko programa *Zadig* koji se može preuzeti preko sljedeće poveznice:

https://zadig.akeo.ie/

Unutar *Zadig* programa potrebno je odabrati instalaciju *WinUSB* upravljačkih programa.

Nakon instalacije svih programskih komponenti potrebno je sve datoteke iz direktorija program na priloženome disku otvoriti unutar *Visual Studio Code* okruženja zatim upisati unutar terminala sljedeću naredbu:

```
npm install
```

Ovom naredbom se instaliraju svi potrebni *Node.js* moduli koji se koriste unutar programa.

HRMonitor program se pokreće naredbom:

```
node index.js
```

Aplikacija se pregledava putem internet pretraživača (npr. Google Chrome) na adresi *localhost:3000*. Sva mjerenja spremaju se u obliku JSON datoteke unutar glavnog direktorija programa.

2. PROGRAMSKI KOD INDEX.HTML DATOTEKE

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Diplomski rad</title><!-- NASLOV STRANICE -->

    <link rel="icon"
      type="image/png"
      href="favicon.png"> <!-- POSTAVLJANJE FAVICONA -->
</head>
<body>
  <div style="text-align: center">
  <img src="logo_v1.jpg" alt="logo" class="center">
  </div>
<!-- UCITAVNJANJE JAVASCRIPT PROGRAMA -->
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"
type="text/javascript"></script>
<script src=
"https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.1.4/Chart.min.js">
</script>
<!-- POSTAVLJANJE VELIČINE GRAFIKONA -->
</canvas><canvas id="HRChart" width="400" height="100"></canvas>
<script  src = "chart.js">
</script>
</body>
</html>
```

3. PROGRAMSKI KOD CHART.JS DATOTEKE

```
//INICIJALIZACIJA NOVOGA GRAFIKONA
var ctx = document.getElementById('HRChart').getContext('2d');
    var HRChart = new Chart(ctx, {
  type: 'line',
  data: {
//DEFINIRANJE PRAZNIH PODATAKA ZA POČETAK RADA
    labels: [' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',],
    datasets: []
  },
  options: {
     //PRIKAZ LEGENDE
              legend: {
                      display: true,
                      labels: {
                      fontSize: 30
                      }
                },
                responsive: true,
                legend: {
                    position: 'bottom',
                    labels:{
                      usePointStyle: true
                    }
                },
                hover: {
                    mode: 'label'
                },
                scales: {
     //DEFINICIJA X-OSI
                    xAxes: [{
                            display: true,
                            scaleLabel: {
                                display: true,
                                labelString: 'Time'
                            }
                        }],
     //DEFINICIJA Y-OSI
                    yAxes: [{
                            display: true,
                            labelString: 'Hearth Rate',
                            ticks: {
```

```
                                beginAtZero: true ,
                                max: 200,
                                min: 0
                            }
                        }]
                },
     //NAZIV GRAFIKONA
                title: {
                    display: false,
                    text: 'HR Ant +  Monitor'
                },
                gridLines:{
                  display: true,
                  drawBorder: true,
                  drawOnChartArea:true,
                  drawTicks:false
                }
              }
});
//BOJE KORIŠTENE U GRAFIKONU
var colors = ["#0440f2","#d67f0e","#0e874d"];

var i = 0; //POMOČNA VARIJABLA ZA ODREĐIVANJE INDEKSA UREĐAJA
var getDevice = function() {
  $.ajax({ // DOHVAČANJE JSON DATOTEKE KORIŠTENJEM AJAX SKRIPTE
    url: 'devices.json',
    success: function(data) {
      if(i < data.length){ //ZA SVAKI UREĐAJ U JSON DATOTECI
        var newDataset = {//NAPRAVI NOVI DATASET
          label: 'Hearth Rate - Device ID = ' + data[i].device,
          data: [0,0,0,0,0,0,0,0,0,0,0,0],
          fill: false,
          borderColor:colors[i],
          backgroundColor: colors[i],
          pointBackgroundColor: colors[i],
          pointBorderColor: colors[i],
          pointHoverBackgroundColor: colors[i],
          pointHoverBorderColor: colors[i]
        };
       //DODAJ NOVI DATASET UNUTAR GRAFIKONA
        HRChart.data.datasets.push(newDataset);
       //AŽURIRAJ GRAFIKON
        HRChart.update();
        i++;
```

```
      }
    }
  });
};
//U PERIODU OD JEDNE SEKUNDE DOHVATI SVE UREĐAJE KOJI SU ZAPISANI
UNUTAR JSON DATOTEKE
setInterval(getDevice, 1000);

var getData = function() { //DOHVACANJE JSONA POMOCU AJAX SKRIPTE
    for(var j = 0 ; j < HRChart.data.datasets.length ;j++){
     //FUNKCIJA SE POZIVA PREKO j VARIJABLE
    (function(j){
    $.ajax({
     //DOHVAČANJE JSON DATOTEKE SA MJERENJIMA SENZORA
      url: 'output'+HRChart.data.datasets[j].label.substring(26,31)
      +'.json',success: function(data) {
        //RADI DOJMA ANIMACIJE U GRAFIKONU, BRIŠE SE PRVI ČLAN POLJA
TE SE CIJELO POLJE POMIČE ULIJEVO. ZATIM SE DODAJE NOVI ČLAN POLJA.
        if(j == 0){
          HRChart.data.labels.shift();//IZBRIŠI PRVI ČLAN DATA POLJA
          HRChart.data.labels.push(data[Object.keys( data ).
          length - 1 ].time); //DODAJ PROTEKLO VRIJEME
        }

        HRChart.data.datasets[j].data.shift
        HRChart.data.datasets[j].data.push
        (data[Object.keys( data ).length - 1 ].hr);

        HRChart.update();


      }
    });
  }(j));
}
};
//DOHVATI PODATKE O MJERENJU SVAKE SEKUNDE
setInterval(getData, 1000);
```

4. PROGRAMSKI KOD INDEX.JS DATOTEKE

```
//UKLJUČIVANJE NODE.JS MODULA
var http = require('http');
var url = require('url');
var path = require('path');
var fs = require('fs');

//IZRADA LOKALNOG SERVERA NA PORTU 3000
var server = http.createServer(handleRequest);
server.listen(3000);
console.log('Pokrenut server na portu: 3000');


//SLANJE SVIH DATOTEKA IZ PRIMARNOG DIREKTORIJA NA SERVER
function handleRequest(req, res) {
  var pathname = req.url;
  if (pathname == '/') {
    pathname = '/index.html';
  }
  var ext = path.extname(pathname);

  // DEFINICIJA TIPOVA DATOTEKA KOJE SE ŠALJU NA SERVER
  var typeExt = {
    '.html': 'text/html',
    '.js':   'text/javascript',
    '.css':  'text/css',
    '.json': 'text/json',
    '.ico': 'image/ico'
  };
  var contentType = typeExt[ext] || 'text/plain';
  fs.readFile(__dirname + pathname,
    function (err, data) {
      if (err) {
        res.writeHead(500);
        return res.end('Error loading ' + pathname);
      }
      res.writeHead(200,{ 'Content-Type': contentType });
      res.end(data);
    }
  );
}
```

```
//ID SVIH KORIŠTENIH UREĐAJA
var jsonDevice =[
  {"device": 29469},
  {"device": 42763},
  {"device": 41743}
]
fs.writeFileSync('./devices.json', JSON.stringify(jsonDevice));

//ANT PLUS DEFINICJE I IZVRŠAVANJE
const Ant = require('ant-plus');
let stick = new Ant.GarminStick2;

//POLJE UNUTAR KOJEGA SE SPREMAJU IDENTIFIKATORI SENZORA
let sensor = [];

//ZA SVAKI UREDJAJ U JSON FILEU STVORI NOVI SENZOR
for(var j = 0; j < jsonDevice.length; j++){
    sensor[j] = new Ant.HeartRateSensor(stick);
}

//PROVJERA POSTOJANJA ANT USB UREĐAJA UNUTAR SUSTAVA
if (!stick.open()) {
    console.log('Stick not found!');
  }


//JSON VARIJABLA ZA SPREMANJE PODATAKA U BAZU PODATAKA
var jsonData = [
     {"hr":10 , "time":"start"}
]
jsonData =  JSON.stringify(jsonData);

 //OSTVARIVANJE KOMUNIKACIJE USB UREĐAJA SA ANT UREĐAJIMA
stick.on('startup', function () {
    for(var j = 0; j < jsonDevice.length; j++){
      sensor[j].attach(j+1, jsonDevice[j].device);
    }
});


//IZRADA JSON DATOTEKE KOJA SADRŽI IDENTIFIKATORE ANT UREĐAJA
for(var j = 0; j < jsonDevice.length; j++){
  fs.writeFileSync('./output'+jsonDevice[j].device+'.json',jsonData);
}
```

```
//OČITAVANJE PODATAKA SVAKOGA UREĐAJA I SPREMANJE VRIJEDNOSTI UNUTAR
JSON DATOTEKE
for(var j = 0; j < jsonDevice.length; j++){
sensor[j].on('hbData', function (data) {
      var today = new Date();
      var tim = today.getHours() + ":" + today.getMinutes() + ":" +
today.getSeconds();

      appendObject({"hr": data.ComputedHeartRate, "time":
      tim},'./output'+data.DeviceID+'.json');
  });
}

//FUNKCIJA ZA DODAVANJE PODATAKA U JSON DATOTEKU
function appendObject(obj,a){ //APPEND FUNCKIJA
  var configFile = fs.readFileSync(a);
  var config = JSON.parse(configFile);
  config.push(obj);
  var configJSON = JSON.stringify(config);
  fs.writeFileSync(a, configJSON);
}
```