Korištenje Arduino mikrokontrolera za izvođenje mjerenja i prijenos podataka na daljinu

Lovrić, Nikica

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:217:783936

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-25**

Repository / Repozitorij:

Repository of the Faculty of Science - University of Zagreb

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

Nikica Lovrić

Korištenje Arduino mikrokontrolera za izvođenje mjerenja i prijenos podataka na daljinu

Diplomski rad

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
FIZIČKI ODSJEK

INTEGRIRANI PREDDIPLOMSKI I DIPLOMSKI SVEUČILIŠNI STUDIJ
FIZIKA I INFORMATIKA; SMJER NASTAVNIČKI

Nikica Lovrić

Diplomski rad

Korištenje Arduino mikrokontrolera za izvođenje mjerenja i prijenos podataka na daljinu

Voditelj diplomskog rada: Izv. prof. dr. sc. Emil Tafra

Ocjena diplomskog rada: _____

Povjerenstvo: 1. _____

2. _____

3. _____

Datum polaganja: _____

Zagreb, 2024.

Sažetak

U ovom radu sam istražio mogućnost slanja podataka na velike udaljenosti koristeći Arduino i LoRa tehnologiju. Koristeći Arduino MKRWAN 1300 i SHT-21 senzor sam prikupljao podatke o temperaturi zraka i relativnoj vlažnosti i pokazao mogućnosti slanja tih podataka preko LoRaWAN mreže koristeći LoRaPHY signal.

Ključne riječi: Arduino, MKRWAN 1300, PMW, I2C, Arduino Cloud, The Things Network, Internet Stvari, LoRa, LoRaWAN, Bandwidth, Chirp signal, Spreading factor, Frequency shift chirp modulation, SHT21, Temperatura zraka, Relativna vlažnost, Silicijski diodni sensor, Kapacitvni senzor relativne vlage

Using Arduino microcontroller to perform measurements and transmit data remotely

Abstract

In this paper, I explored the possibility of sending data over long distances using Arduino and LoRa technology. Utilizing the Arduino MKRWAN 1300 and the SHT-21 sensor, I collected data on air temperature and relative humidity, demonstrating the capabilities of transmitting this data over the LoRaWAN network using the LoRaPHY signal.

Keywords: Arduino, MKRWAN 1300, PMW, I2C, Arduino Cloud, The Things Network, Internet of Things, LoRa, LoRaWAN, Bandwidth, Chirp signal, Spreading factor, Frequency shift chirp modulation, SHT21, Air temperature, relative humidity, Silicon band gap thermometer, Capacitive humidity sensor

Sadržaj

1 Uvod **1**

2 Mikrokontroleri i tehnologija Arduino platforme **2**

2.1 Što je Arduino? 2

2.1.1 Kratka povijest Arduino-a 2

2.2 Mikrokontroler MKRWAN 1300 i povezivanje komponenti 3

2.2.1 Pinovi na mikrkontroleru MKRWAN 1300 4

2.2.2 Pinovi za napajanje 4

2.2.3 Analogni pinovi 5

2.2.4 Konverzija analognog signala u digitalni signal (ADC) 5

2.2.5 Digitalni I/O pinovi 5

2.3 Modulacija signala : PWM - Pulse Width Modulation 7

2.4 Serijski prijenos podataka 8

2.4.1 Razlika između serijske i paralelne komunikacije 8

2.4.2 UART (Universal Asynchronous Receiver/Transmitter) 9

2.4.3 SPI - Serial Peripheral Interface 11

2.4.4 **I^2C (Inter Integrated Circuit)** 11

2.5 O Arduino programskom jeziku 13

2.5.1 Arduino Integrated Development Enviroment (IDE) 13

2.5.2 Arduino Cloud 13

2.6 Arduino Shields 14

2.7 Arduino senzori 14

2.8 Osnovne komponente Arduino projekta 15

3 Internet of Things - IoT **16**

3.1 Uvod u IoT 16

3.2 Povijest Interneta stvari (IoT) 17

3.3 Tehnološke komponente Interneta stvari 19

3.4 Primjena Interneta stvari (IoT) u svakodnevnom životu 22

4 LoRa i LoRaWAN tehnologija **23**

4.1 Uvod u LoRa tehnologiju 23

4.1.1 Prednosti LoRa tehnologije 24

4.2 Radio modulacija i LoRa 25
4.2.1 Pozicija LoRa-e u OSI modelu 27
4.3 LoRa PHY signal specifikacije 28
4.3.1 Bandwidth 28
4.3.2 Chirp signal 28
4.3.3 SF - Spreading factor 29
4.3.4 Frequency Shift - Chirp Spread Spectrum - modulacija LoRa PHY signala 30
4.3.5 Trenutna frekvencija 31
4.3.6 Modulacija 31
4.3.7 Demodulacija 33
4.4 LoRaWAN 34

5 SHT-21 breakout pločica 35
5.1 SHT21 senzor 35
5.1.1 Silicijski diodni termometar 35
5.1.2 Kapacitivni senzor za mjerenje relativne vlage 36

6 Mjerenja 37
6.1 Postav mjerenje 37
6.1.1 Spajanje mikrokontrolera sa senzorom 37
6.1.2 Komunikacija sa mikrokontrolerom 38
6.1.3 Programiranje mikrokontrolera 39
6.1.4 The Things Network heatmap 40
6.2 Rezultati mjerenja 42
6.2.1 Mjerenje u park šumi Grmošćica 42
6.2.2 Mjerenje kod PMF Fizičkog odsjeka 44
6.2.3 Mjerenje na Sljemenu 45
6.3 Diskusija rezultata i prijedlozi poboljšanja 48

7 Zaključak 49

Dodaci 50

Literatura 51

1 Uvod

U današnjem digitalnom dobu, gdje prijenos podatak ima ključnu ulogu, raste potreba za razvojem sustava koji omogućuju pouzdan i energetski učinkovit prijenos podataka na velike udaljenosti. Ovaj diplomski rad fokusira se na istraživanje i implementaciju Arduino MKR WAN 1300 mikrokontrolera u kombinaciji s senzorom SHT21, koristeci LoraPHY signal sa LoRaWAN protokolom (Long Range Wide Area Network). Cilj je iskoristiti učinkovit sustav za bežični prijenos podataka na velike udaljenosti, a istovremeno omogućiti relativno precizno mjerenje temperature i vlage zraka. Motivacija za ovaj rad je došla iz korištenja balona na velikim visinama za prikupljanje podataka. Baloni za velike visine mogu letiti na nadmorskoj visini od 10-15 km (stratosfera), što čini LoRa tehnologiju jako pogodnom za takav projekt, jer omogućava lako i efikasno prikupljanje podataka senzora koji su na balonu.

2 Mikrokontroleri i tehnologija Arduino platforme

2.1 Što je Arduino?

Arduino je open-source elektronička platforma koja omogućava izradu različitih projekata, od jednostavnih svjetlosnih efekata do složenih uređaja za mjerenja i kontrolu. Platforma je posebno popularna među hobi elektroničarima, inženjerima, i predmetnim nastavnicima zbog svoje jednostavnosti korištenja i dostupnosti.

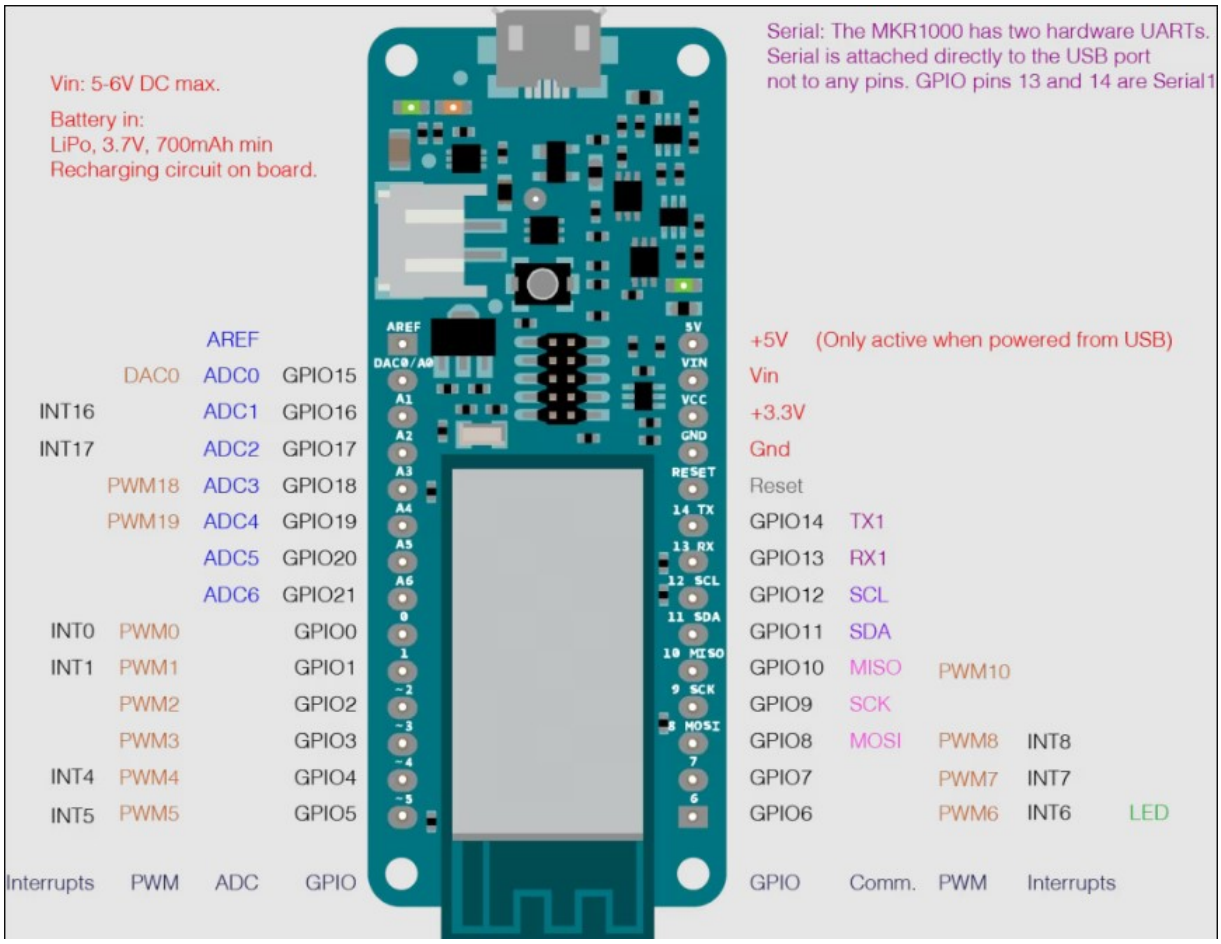
2.1.1 Kratka povijest Arduino-a

Arduino je nastao 2005. godine kao projekt u okviru Interactive Telecommunications Program (ITP) na Sveučilištu New York. Razvijen je kako bi olakšao proces prototipiranja i omogućio brzo eksperimentiranje s elektronikom. Projekt su stvorili Massimo Banzi, David Cuartielles, David Mellis, Tom Igoe i Gianluca Martino.

Slika 2.1: Arduino Logo, slika je preuzeta sa izvora [1]

2.2 *Mikrokontroler MKRWAN 1300 i povezivanje komponenti*

Razvojna ploča je fizička osnova Arduino sustava. Na njoj se nalazi mikrokontroler, USB priključak, digitalni i analogni pinovi, kao i druge komponente koje olakšavaju povezivanje sa senzorima, aktuatorima i drugim uređajima. U ovom radu ja ću raditi s Arduino MKRWAN 1300 mikrokontrolerom.



Slika 2.2: Arduino MKRWAN 1300 mikrokontroler pinout, slika je preuzeta sa izvora [2]

MKRWAN 1300 koristi mikročip AT SAMD21, koji u sebi sadrži 48 MHz 32-bitni MO+ ARM Cortex procesor i Murata CMWX1ZZABZ-078 modul za LoRa komunikaciju

2.2.1 Pinovi na mikrkontroleru MKRWAN 1300

Svi Arduino mikrokontroleri posjeduje digitalne i analogni pinove koji omogućuju povezivanje s različitim uređajima. Digitalni pinovi rade na principu logičkih razina 0 ili 1, dok analogni pinovi omogućuju čitanje analognih vrijednosti, kao što su jačina svjetlosti ili temperatura.

2.2.2 Pinovi za napajanje

- **Li-Po(3.7V)** - mikrokontroler možemo napajati sa Li-Po[1] baterijom koja ima nominalni napon 3.7V
- **Vin** - Mikrokontroler preko ovog pina možemo napajati sa naponom od 5V, najveći napon koji možemo pustiti na ovaj pin je 6V.
- **5V** - ovaj pin možemo koristiti kada mikrokontroler napajamo preko USB-a
- **Vcc** - ovaj pin ima ugrađeni potenciometar s kojim regulira da ovaj izlaz uvijek daje 3.3V
- **GND** pin se koristi da zatvorimo strujni krug. Moramo paziti da su nam Arduino i njegove komponente spojene jedno s drugim i spojene na isti GND pin.
- **RESET** pin - služi za resetiranje mikrokontrolera, kada aktiviramo ovaj pin, on spusti napon na LOW[2] s čime resetira mikrkontroler. Praktična primjena ovog pina je da se doda gumb za resetiranje na neki senzor koji koristimo i onda pritiskom na tipku možemo blokirati jedan od senzora spojenih na mikrokontroler.

[1] Li-Po je Lithium polimerska baterija koja se može puniti, često ju susrećemo u mobilnim uređajima, laptopima i powerbank-ovima.

[2] 'LOW' na engleskom jeziku u ovom kontekstu označava stanje niskog napona.

2.2.3 Analogni pinovi

Arduino MKRWAN1300 ima 7 analognih pinova, koji koriste ADC[3]. Ovi pinovi služe za čitanje analognog signala ; analogni signal je oblik signala kojemu vremenski prikaz u svakom trenutku odgovara fizikalnoj veličini električnog napona. Analogni pinovi se mogu koristiti i za ulaz ili izlaz digitalnog signala, radi ADC-a.

- **A0-A6 pinovi -** imamo 7 analognih pinova koji se koriste za primanje analognih signala u rasponu od 0-3.3V. Pinovi imaju 10-bitnu rezoluciju, što znači da koristeći ADC mogu predstavljati analogni signal na 1024 digitalne razine.
- **DAC0 pin -** dijeli mjesto sa A0 pinom. Služi za konvertiranje digitalnog signala u analogni signal.
- **AREF pin** - postavlja referentni napon za analogni ulaz, koristi se sa funckijom analogReference().

2.2.4 Konverzija analognog signala u digitalni signal (ADC)

ADC je akronim za 'Analog to Digital Converter'. ADC je strujni krug koji se koristi za konvertiranje analognog signala u digitalni. Naprimjer: pretvaranje analognog napona u digitalni broj, koji se onda zapisuje u memoriju.

2.2.5 Digitalni I/O pinovi

Digitalni signal je način prikaza napona kao 1 bit: 0 (niski napon) ili 1(visoki napon). Na Arduinu su digitalni pinovi konfigurirani kao ulaz ili izlaz signala ovisno o potrebama korisnika. Digitalni pinovi su uključeni ili isključeni. Kada su uključeni (eng. 'ON') tada su u stanju visokog (eng.'HIGH) napona koji iznosi 5V, kada su ugašeni su u stanju niskog (eng.'LOW') napona koji iznosi 0V.

Digitalni pinovi konfigurirani kao izlaz su postavljeni na 0V ili 5V.

[3]ADC stoji za 'Analog to Digital Converter'. Česta primjena ADC-a je Voice over Inernet Protocol (VoIP, npr. platforma Whatsapp koristi ovaj protokol), svaki pametni telefon ima mikrofon koji konvertira zuvčne valove (glas) u analogni signal napona, taj signal onda ide kroz ADC i pretvara se u digitalni signal, koji se onda prenosi do drugog krajnjeg uređaja preko interneta.

Digitalni pinovi konfigurirani kao ulaz, znači da signal odnosno napon dolazi iz vanjskog izvora. Napon može varirati između 0 i 5V koja se onda konvertira u digitalni reprezentaciju 0 ili 1. Da bi odredili kada je napon visok ili nizak, postoje 2 praga:

- Ispod 0.8V se smatra logičkom 0
- Iznad 2V se smatra logičkom 1

Kada spajamo komponentu na digitalni pin, moramo paziti da se pragovi napona za logičke razine podudaraju. Ako je napon između dva praga, izlazna vrijednosti će biti nedefinirana.

Digitalni I/O pinovi na MKRWAN 1300 su pinovi od 0-14. Za svaki pin preporučena jalina struje je 20mA, a ne smije prekoračiti 40mA. Ukupna jačina struje za rad svih pinova je 200mA. Digitalni pinovi imaju sljedeću podjelu:

- **Serial TX i RX pinovi** - pinovi 13 i 14 se koriste za primanje (engl. 'Receive') Rx i slanje (engl. 'Transmit') Tx serijskih podataka kroz TTL (Tranzistor-Tranzistor Logika koja se koristi u UART[4] krugovima.). Ovi pinovi su spojeni na FTDI USB-TTL čip.
- **PMW pinovi** su 0, 1, 2, 3, 4, 5, 6, 7, 8, 10, A3, A4 ; ovi pinovi omogućavaju 8-bitni PMW[5] izlaz koristeći analogWrite() funkciju.
- **SPI[6] pinovi**
  - pin 8 - MOSI (Master Out Slave In) : kanal 'Master' uređaja za slanje podataka podataka perifernim uređajima
  - pin 9 - SCK (Serial clock) : signal sata koji generira 'Master' uređaj prijenos podataka između kanala
  - pin 10 - MISO (Master In Slave Out) : kanal zadužen za slanje podataka 'Master' uređaju

[4]UART- 'Universally asynchronous receiver/transmitter'
[5]PMW - Pulse Width Modulation
[6]SPI- Serial Peripheral Interface

- **I2C[7] pinovi**
  - pin 11 - SDA (Serial Data line)
  - pin 12 - SCL (Serial Clock line)
- **Pin 6** ima spojenu LED lampicu na sebe. Kada je napon visok, lampica je upaljena, kada je napon nizak, lampica je ugašena.

Da bi bolje razumjeli pinove na Arduino mikrokontroleru, trebamo shvatiti kako Arduino modulira signale i koje protokole koristi za komunikaciju sa različitim komponentama koje možemo spojiti na njega.

2.3 Modulacija signala : PWM - Pulse Width Modulation

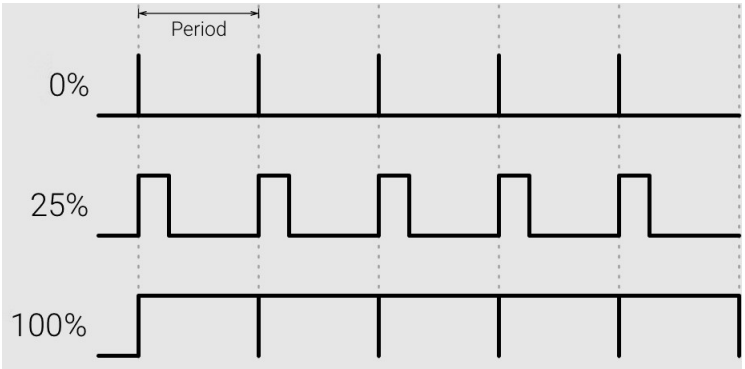
Generalno PWM je modulacijska tehnika koji se koristi za kodiranje poruka u pulsni signal.
PWM se sastoji od dvije ključne komponente:

- Frekvencija
- Duty cycle - omjer vremena unutar jednog perioda u kojem je strujni krug uključen nasprem vremena kada je strujni krug isključen.

Frekvencija PWM određuje koliko je vremena potrebno da se odvije jedan ciklus (period) i koliko brzo signal fluktuira od maksimalne amplitude do minimalne amplitude. 'Duty cycle' određuje koliko dugo signal ostaje na maksimalnoj amplitudi kroz cijeli period. 'Duty cycle' se prikazuje u postotku, kao što je prikazano na slici **??**.

U Arduino mikrokontrolerima, PWM pinovi produciraju konstantnu frekvenciju od 500Hz, dok se 'duty cycle' mijenja ovisno o parametrima postavljenim od strane korisnika. PWM signali se koriste za kontrolu brzine DC motora, gušenje svjetline LED lampica, itd..

[7] I^2C - Inter-Integrated circuit



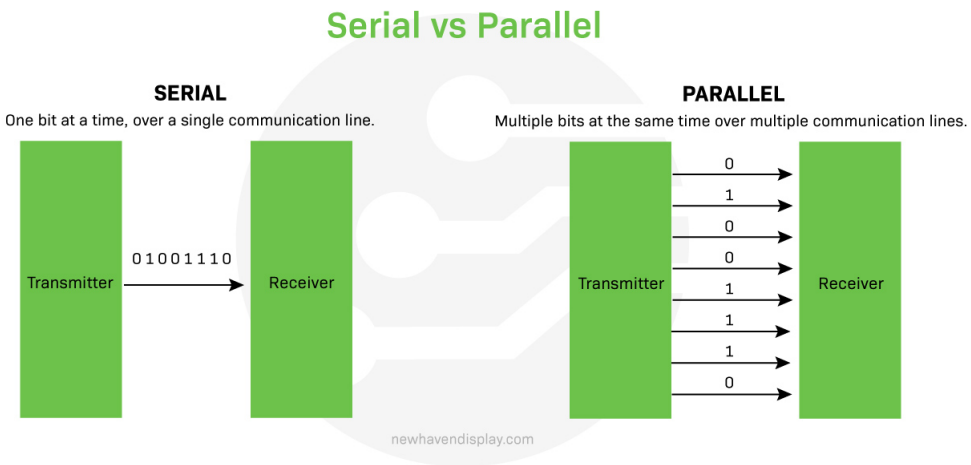
Slika 2.3: PWM signal. Slika je preuzeta sa izvora [3]

2.4 *Serijski prijenos podataka*

Da bi malo bolje razumjeli pinove koje Arduino uređaji koriste, trebamo razumjeti na koje načine mikrokontroler može komunicirati sa kompjuterom ili periferijom spojenom na njega.

2.4.1 Razlika između serijske i paralelne komunikacije

Serijska komunikacija radi na način da šalje bitove jedan po jedan kroz jednu žicu. To možemo zamisliti kao analogiju telefonskom pozivu gdje oba korisnika mogu pričati u isto vrijeme, ali se riječi šalju jedna po jedna.
Dok serijska komunikacija šalje bit po bit na jednoj žici, paralelna komunikacija šalje više bitova u isto vrijeme preko više žica. Razlika je prikazana sa slikom 2.4



Slika 2.4: Serijska komunikacija šalje bit po bit preko jedne žice, dok paralelna šalje više bitova u isto vrijeme preko više žica. Slika je preuzeta sa izvora [5]

2.4.2 UART (Universal Asynchronous Receiver/Transmitter)

UART je akronim za 'Universal Asynchronous Receiver/Transmitter'. Koristi se za serijsko slanje i primanje podataka. Koristi RS-232[8] serijski protokol.
UART danas više nije toliko popularan, jer većina računala danas ne komunicira preko serijskih portova, već koriste tehnologije poput USB-a i Etherneta. UART i dalje pronalazi svoje mjesto u mikrokontrolerima radi svoje jednostavnosti i lagane implementacije.
Za razliku od I2C-a i SPI-a, UART nije protokol već je fizički strujni krug koji možemo naći unutar Arduino mikrokontrolera. Koristi se za komunikaciju između mikrokontrolera i kompjutera ili drugih uređaja kao što su GPS modul ili RFID modul.
UART je jako jednostavan i koristi samo dvije žice između primatelja i pošiljatelja da šalje i prima podatke u oba smjera. Komunikacija u UART sustavu se dijeli na:

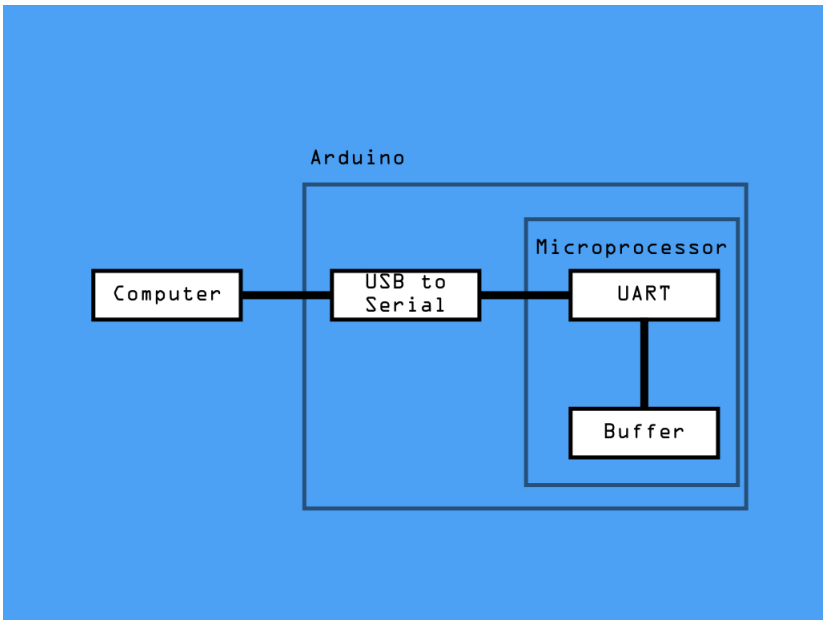
- **simplex** - podaci se šalju u samo jednom smjeru
- **half - duplex** - obostrana komunikacija, ali nemogu obje strane u isto vrijeme slati i primati podatke
- **full-duplex** - obje strane mogu slati i primati podatke u isto vrijeme

U arduino mikrokontrolerima komunikacija je full duplex.

UART serijska komunikacija sa Arduinom

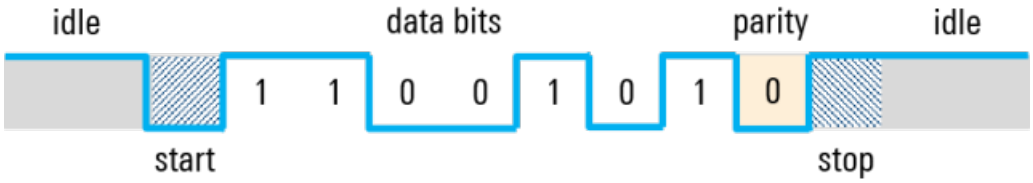
- Kompjuter šalje podatke mikrokontroleru preko USB-a. Tada se taj USB signal pretvara u serijski signal i šalje se UART uređaju. Primljeni podaci se onda spremaju u međuspremnik za daljnje korištenje. Prikazano na slici 2.5
- Za serijsku komunikaciju su potrebne samo 3 žice:
  - **TX** : linija za slanje podataka
  - **RX** : linija za primanje podataka
  - **GND** : uzemljenje

[8] RS-232 je standardni protokol koji se koristi za serijsku komunikaciju, koristi se za spajanje računala sa perifernim uređajima koji omogućuje serijsku izmjenu podataka između njih.



Slika 2.5: Grafički prikaz komunikacije između računala sa mikrokontolerom preko USB-a

- UART šalje podatke asinkronizirano, što znači da ne postoji eksterni sat koji pogoni prijenos bitova. Bez signala sata, UART koristi start i stop bit, između kojih se nalazi paket bitova koji želimo poslati.
  - **Start bit** se koristi da indicira početak podatkovnog teksta, i koristi se za sinkronizaciju pošiljatelja i primatelja. Startni bit je uvijek 0.
  - **Stop bit** indicira kada je zadnji bit podataka poslan. Stop bit je uvijek 1.
  - **Paket bitova** je obično oko 5-9 bitova u duljini i to je informacija koju šalje ili prima UART uređaj. Ta informacija je u obliku binarnog zapisa. Kao što je prikazano na slici 2.6
  - **Parni bit** se koristi za detekciju greški preko algoritma koji se danas rijetko koristi.
- **Baud rate** - kada uređaj koji prima serijski signal detektira startni bit, on kreće procesuirati nadolazeće bitove. Ali ti bitovi trebaju biti procesuirani na određenoj frekvenciji koju koriste i primatelj i pošiljatelj. Ta frekvencija se naziva **baud rate**, ona je mjera kojom se brzinom podaci šalju ili primaju, iskazujemo ju kao bitovi po sekundi. Često korišteni baud rate na Arduino uređajima je 9600, 14400, 19200 ili čak 115200 bitova u sekundi.



Slika 2.6: Paket podataka koji UART uređaj šalje se sastoji od : start bita, paketa bitova, stop bita i opcionalnog parnog bita. Slika je preuzeta sa izvora [7]

2.4.3 SPI - Serial Peripheral Interface

SPI je sinkronizirani protokol za serijsko slanje podataka koji se koristi kod mikrokontrolera za komunikaciju sa jednim ili više perifernih uređaja. SPI sustav koristi 4 žice/pina[9] Pinovi su označeni kao MISO, MOSI,SCK i CS. MKRWAN 1300 nema CS pin. Miso se koristi za slanje podataka, MOSI za primanje podataka i SCK pin za sinkronizaciju razmjene podataka između dva uređaja. SPI je full-duplex komunikacija. Maksimalni baud rate je u praksi oko 10-100 Mbps. Za korištenje ovog protokola na Arduino uređajima na početku koda trebamo uključiti biblioteku SPI.h.

2.4.4 I^{2}C (Inter Integrated Circuit)

I2C protokol koristi samo dva pina da šalje i prima podatke. SCL (Serial Clock) pin koji se brine za sinkronizaciju prijenosa podataka i SDA (Serial Data) pin preko kojeg se podaci šalju između dva uređaja. Kod I2C protokola postoji samo jedan uređaj koji je kontroler i više perifernih uređaja koji su spojeni na taj kontroler preko SCL i SDA pinova. Signal sata se mijenja iz niskog napona u visoki napon ('rising edge of the clock pulse'), jedan bit informacije se šalje od kontrolera do I2C uređaja preko SDA pina. Kako signal sata nastavlja pulsirati tako se šalju bitova sve dok se ne oformi

[9]Arduino više ne koristi terminologiju 'master' i 'slave' i zamjenio je te riječi sa 'controller' i 'peripheral', iako i dalje možemo naići na uporabu te terminologije. Na novijim Arduino uređajima je MISO (Master In Slave Out) postao CIPO (Controller In Peripheral Out), MOSI (Master Out Slave In) je postao COPI (Controller Out Peripheral In) i SS(Slave Select) je CS(Chip Select).

sekvenca od 7 ili 8 bitova, koja onda čini adresu, instrukciju ili podatke. Kada se ta informacija pošalje uređaju on obrađuje zahtjev i šalje podatke nazad istom linijom koristeći signal sata od kontrolera.
Svaki uređaj spojen na I2C pinove je funkcionalno neovisan od kontrolera, ali će dati informaciju kontroleru u bilo kojem trenutku kada ga kontroler traži.
I2C protokol dopušta svakom perifernom uređaju da ima svoju unikatnu adresu o da kontroler i periferni uređaj komuniciraju preko jedne linije naizmjence. Također je onda moguće da Arduino miktrokontroler komunicira nazimjence sa više perifernih uređaja ili drugih mikrokontrolera, koristeći samo dva pina.

Slika 2.7: Format I2C poruke, slika je preuzeta sa izvora [6]

Na slici 2.7 je prikazana struktura poruke koju izmjenjuju pošiljatelj i primatelj u I2C protkolu. Ukratko ću objasniti strukturu te poruke:

- Kontroler šalje poruku kroz SDA pin, svaka poruka počinje sa adresom, tako da 'točan' periferni uređaj 'sluša' poruku.
- Nakon toga ide bit koji određuje da li kontroler želi čitati ili pisati
- Svaka poruka u sebi sadrži bit potvrde (engl. 'acknowledgement'), tako da ne dođe do neočekivanih rezultata. Kada primatelj poruke potvdi prijašnju informaciju to potvrđuje sa bitom potvrde i daje zeleno svjetlo kontroleru da može nastaviti slati sljedeći set bitova.
- Potom ide paket podataka veličine 8-bitova
- Opet bit potvrde
- Paket podataka od 8 bitova
- Još jedan bit potvrde

Treba imati na umu da cijelo vrijeme SCL pin sinkronizira satove kontrolera s uređajima spojenim na njega, osiguravajući pritom da se u isto vrijeme pomiču na

sljedeću instrukciju. Da bi koristili ovaj protokol, ne moramo nužno brinuti o svim ovim stvarima, već Arduino koristi Wire library koji to onda radi za nas.

2.5 O Arduino programskom jeziku

Arduino programski jezik se koristi za programiranje mikrokontrolera kao što je Arduino Uno da interagiraju sa senzorima, aktuatorima i drugim uređajima povezanim na mikrokontroler. Jezik se temelji na programskog jeziku C++ i dizajniran je da ga mogu koristiti korisnici sa različitim razinama iskustva. Koristi se u projektima koji uključuju robotiku, automatizacija domova i Internet of Things(IoT) aplikacije.

2.5.1 Arduino Integrated Development Enviroment (IDE)

IDE je softverski alat koji omogućava korisnicima pisanje, kompajliranje i učitavanje koda na Arduino uređaj. S jednostavnim sučeljem, podrškom za razne platforme i ugrađenim primjerima, IDE olakšava razvoj čak i početnicima. Osim toga, IDE pruža mogućnost praćenja serijske komunikacije s uređajem, što je korisno za ispravak pogrešaka.

2.5.2 Arduino Cloud

Arduino Cloud je cloud platforma koja omogućava jednostavno povezivanje, konfiguriranje i upravljanje Arduino uređajima putem interneta. Integrirajući Internet stvari, omogućuje korisnicima pristup i upravljanjem projektima, koristeći web sučelje ili mobilne aplikacije. Nudi mogućnost stvaranja prilagođenih dashboard-ova za praćenje i kontrolu projekata u stvarnom vremenu.

2.6 Arduino Shields

Arduino Shields su dodatne ploče koje se montiraju na razvojnu ploču i proširuju njen funkcionalitet. Na primjer, WiFi Shield omogućava bežičnu komunikaciju, dok Motor Shield pojednostavljuje kontrolu motora. Ovi dodaci čine Arduino još jednostavnijim i lakšim za prilagoditi različitim projektima.

2.7 Arduino senzori

Arduino senzori omogućuju Arduino platformi da interagira sa svojim okruženjem prikupljajući podatke o različitim fizičkim veličinama. Postoji širok spektar Arduino senzora, a svaki od njih ima specifičnu namjenu.

- Jedan od osnovnih senzora je senzor svjetlosti, koji omogućava mjerenje intenziteta svjetlosti u okolini. Ovi senzori se često koriste u projektima vezanim uz osvjetljenje ili automatsko uključivanje uređaja ovisno o ambijentalnoj svjetlosti.

- Temperaturni senzori, poput SHT21, omogućavaju precizno mjerenje temperature. Ovi senzori su od suštinskog značaja za projekte koji zahtijevaju praćenje temperature, kao što su sustavi kontrole klime ili praćenje okoline.

- Ultrazvučni senzori koriste se za mjerenje udaljenosti pomoću principa eholokacije. Ovi senzori su popularni u robotici i automatizaciji, omogućujući uređajima da detektiraju prepreke i prilagode svoje ponašanje.

- Senzori pokreta, poput PIR senzora, reagiraju na promjene u infracrvenom zračenju iz okoline. Ovi senzori se često koriste u sustavima zaštite od krađe ili za automatizaciju osvjetljenja na temelju prisutnosti.

Mnogi Arduino projekti koriste kombinaciju senzora radi dobivanja složenijih informacija o okolini. Primjerice, kombinacija senzora temperature, vlage i tlaka može se koristiti za izradu meteorološke stanice.

2.8 Osnovne komponente Arduino projekta

Za izradu Arduino projekta, obično se koriste sljedeće komponente:

- Arduino ploča
- Senzori
- Aktuatori
- Elektroničke komponente (otpornici, tranzistori, diode, kondenzatori, itd.)
- Napajanje (baterije ili adapter)
- Računalo za programiranje i praćenje

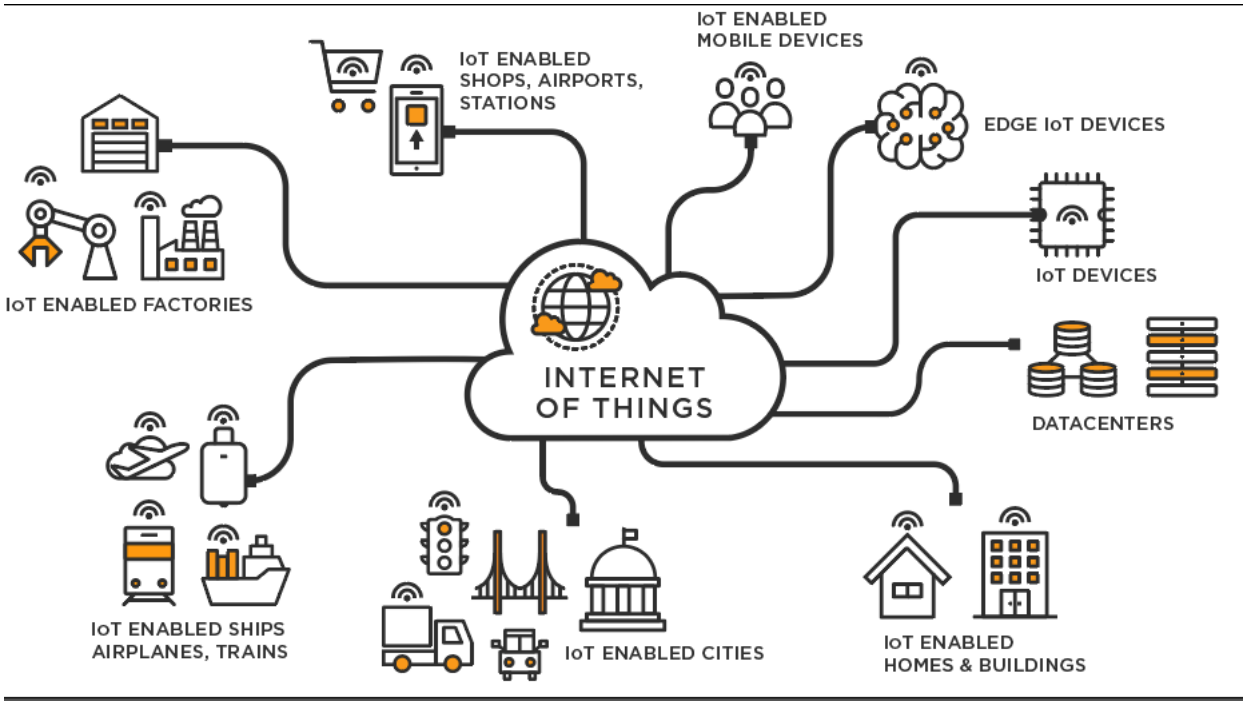
3 Internet of Things - IoT

3.1 Uvod u IoT

Internet stvari (IoT) predstavlja tehnološki koncept koji spaja fizički svijet s digitalnim, otvarajući nove mogućnosti u načinu na koji interagiramo s okolinom, uređajima i uslugama. Ovaj koncept omogućavam uređajima, od kućanskih aparata do industrijskih strojeva, da komuniciraju i razmjenjuju podatake putem računalnih mreža.

Ključna karakteristika IoT-a je njegova sposobnost da omogući predmetima - bilo da je riječ o termostatima, vozilima opremljenim senzorima ili čak pametnim gradovima, da 'osjećaju' i automatski reagiraju na promjene u svojoj okolini. Time smo stvorili ekosustav gdje se podaci neprestano prikupljaju, analiziraju i koriste za poboljšanje svakodnevnih procesa.

U ovom poglavlju ću istražiti osnovne koncepte IoT-a, njegovu povijesti, tehnološke temelje, kao i raznovrsne primjene koje su već počele mijenjati način na koji živimo i radimo. Također ću malo raspraviti o izazovima i sigurnosnim pitanjima koji prate ovu tehnologiju koja se jako brzo razvija.



Slika 3.1: Grafički prikaz IoT-a. Slika je preuzeta sa izvora [8]

3.2 Povijest Interneta stvari (IoT)

Povijest Intereta stvari (IoT) započinje mnogo prije nego što je sam pojam postao popularan. Korijeni IoT-a sežu u kasne 1980-e godine, kada su se počele razvijati ideje o povezivanju uređaja s internetom.

- Rani razvoj (1980-e i 1990-e):
  Prvi koraci prema IoT-u napravljeni su s razvojem tehnologija kao što su RFID [1] i EPC [2] Ove tehnologije omogućile su identifikaciju i praćenje predmeta putem radio valova.

- Pojava termina 'Internet Stvari' (1999.godine):
  Pojam 'Internet stvari' prvi je put upotrijebljen 1999. godine od strane Kevina Ashtona, suosnivača Auto-ID centra na MIT-u. Gospodin Ashton je ovaj termin koristio kako bi opisao mrežu koja povezuje objekte u fizičkom svijetu s internetom putem RFID-a[1]

- Razvoj tehnologije (2000-e):
  U ovom razdoblju dolazi do značajnog tehnološkog napretka u bežičnoj komunikaciji (implemntacija 3G mreže), senzorskim tehnologijama i računalnoj moći. To je omogućilo šire povezivanje uređaja i razvoj prvih praktičnih primjena IoT-a.

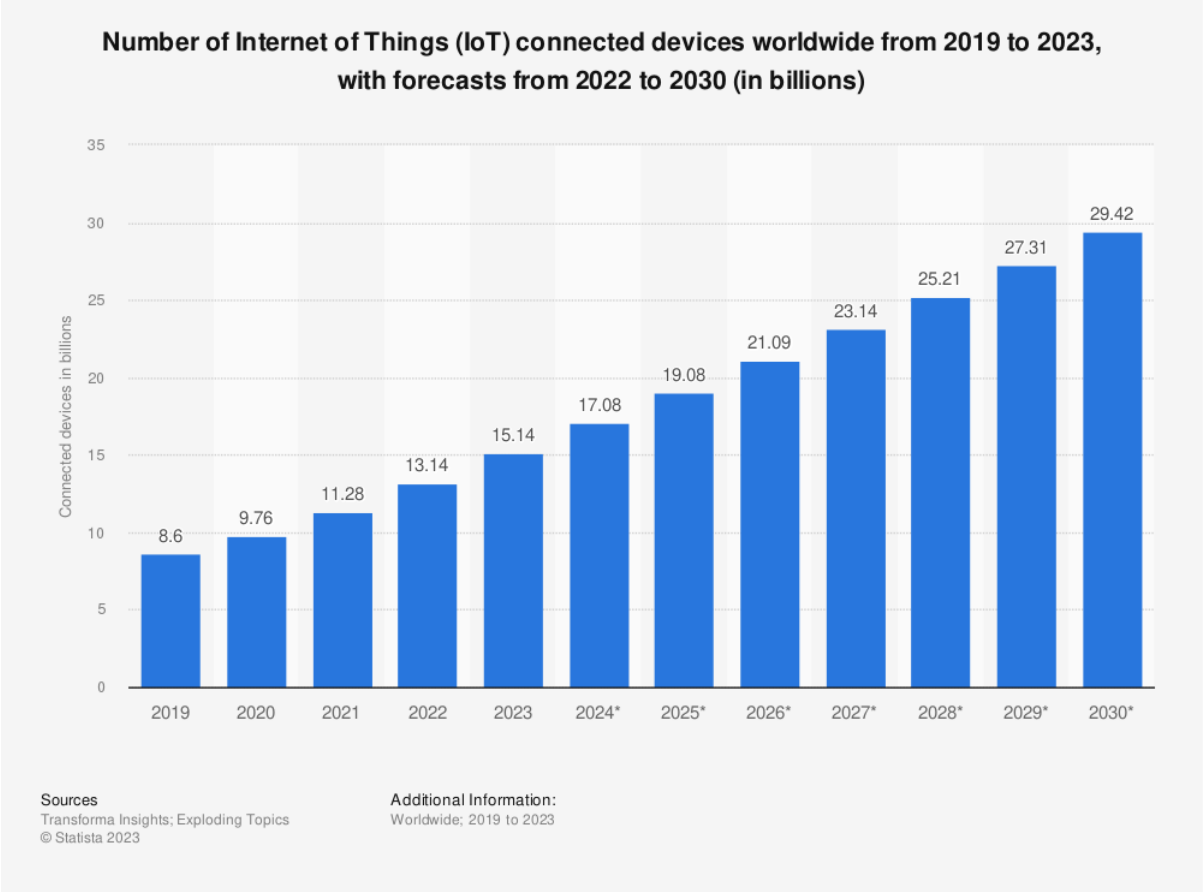
- Ekspanzija i integracija (2010-e):
  dolaskom pametnih telefona i širenjem brzih internet veza, Internet stvari je doživio ogroman rast. Počele su se razvijati različite aplikacije, od pametnih domova i gradova do industrijske automatizacije. Uz to, razvoj tehnologija poput 'cloud-a' i umjetne inteligencije je dodatno doprinjelo mogućnostima IoT-a.

[1]Radio-Frequency Identification (RFID) - koristi elekromagnetska polja da automatski prepozna i prati oznaku (engl. 'tag') koja je dodjeljena objektu. RFID sustav se sastoji od radio transpondera, radio prijamnika i radio odašiljača.

[2]The Electronic Product Code (EPC) je dizajniran kao univerzalni identifikator koji pruža jedinstveni identitet za svaki fizički objekt bilo gdje u svijetu.

- Suvremeni razvoj i budućnost

  Danas IoT predstavlja ključni element u razvoju pametnih tehnologija i digitalne transformacije naše svakodnevnice. U 2023. godini se bilježi da je broj uređaja Internet stvari u cijelom svijetu oko 15.4 miljardi, sa predviđanjem da će taj broj do 2030.godine narasti do 30 milijardi uređaja.



Slika 3.2: Broj IoT spojenih uređaja diljem svijeta od 2019. do 2023., sa prognozom od 2022. do 2030. godine (u milijardama). Slika je preuzeta sa izvora [10]

3.3 Tehnološke komponente Interneta stvari

Internet stvari (IoT) je složeni ekosustav koji se sastoji od niza tehnoloških komponenti. Te komponente su ključne za funkcioniranje i uspješnu implementaciju IoT rješenja. U ovom potpoglavlju ću ukratko opisati glavne tehnološke aspekte koji čine osnovu IoT-a.

1. **Senzori i aktuatori**
   Senzori su zaduženi za prikupljanje podataka iz okoline (npr. Arduino senzori opisani u prošlom poglavlju), kao što su temperatura i vlaga zraka, pokret, svjetlost i zvuk. Aktuatori, s druge strane su mehanički ili elektronički uređaju koji djeluju na osnovu podataka primljenih od senzora, poput otvaranja ventila ili paljenja svjetla.

2. **Povezivost**
   Kako bi senzori i aktuatori mogli komunicirati, potrebna je povezivost. To možemo postići različitim bežičnim tehnologijama, kao što su Wi-Fi, Bluetooth, Zigbee, LoRaWAN i 5G. Izbor tehnologije će ovisiti o potrebama projekta, uključujući domet, brzinu, sigurnost i potrošnju energije.

3. **Obrada podataka**
   Nakon što prikupimo podatke, trebamo ih i obraditi. To može biti jednostavna obrada na samom uređaju (edge computing[3]) ili slanje podataka u oblak (cloud computing[4]) za složeniju analizu. Obrada tih podataka onda omogućuje donošenje odluka i akcija na temelju tih prikupljenih podataka.

4. **Platforme IoT**
   Platforme IoT-a nude okruženje za upravljanje povezanim uređajima, uključujući njihovu konfiguraciju, nadzor i analizu. One omogućuju integraciju s različitih uređaja i aplikacija, pružajući korisničko sučelje za interakciju s IoT sistemom. Ja sam za ovaj rad koristio TTN[5] platformu.

[3]Edge computing je arhitektura distribuirane informacijske tehnologije u kojoj se podaci klijenta obrađuju na periferiji mreže, što je moguće bliže izvoru s kojeg dolaze podaci.

[4]Cloud computing je dostupnost resursa računalnog sustava na zahtjev, posebice pohrane podataka(pohrana u oblaku) i računalne snage, bez izravnog aktivnog upravljanja od strane korisnika. Veliki oblaci šesto imaju funkcije raspoređene na više lokacija, od kojih je svaka podatkovni centar.

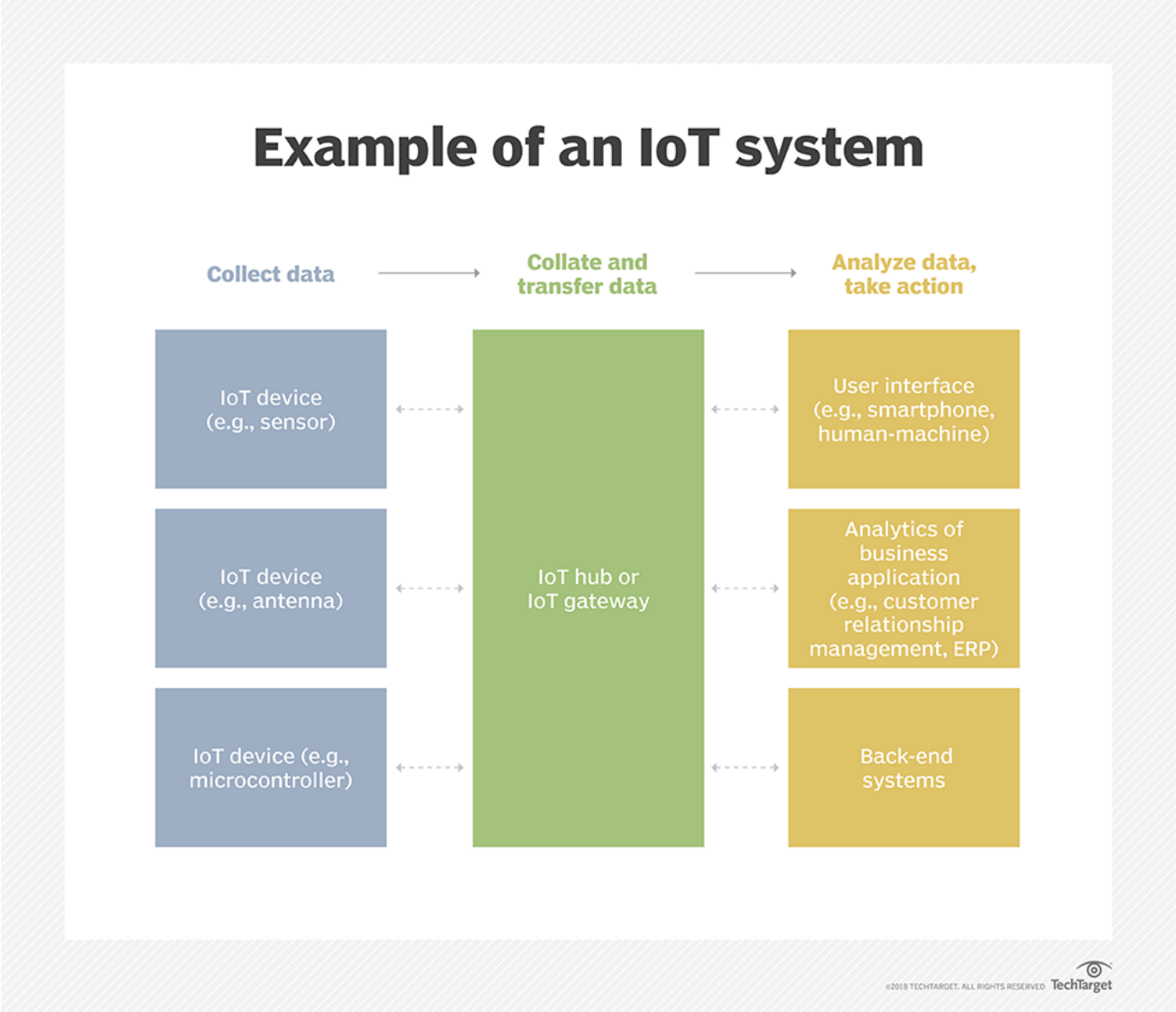
[5]The Things Network

5. **Sigurnost** IoT uređaji prikupljaju i razmjenjuju velike količine podataka, pitanje zaštite tih podataka i uređaja od neovlaštenih pristupa postaje sve važnije. Glavni sigurnosni rizici uključuju neautorizirani pristup podacima, napade na privatnost, i mogućnost udaljenih napada koji mogu dovesti do manipulacije uređajima. Kako bi se osigurala sigurnost Internet stvari, primjenjujemo višeslojne sigurnosne mjere. To uključuje snažnu autentifikaciju i enkripciju podataka kako bi se osigurala povjerljivost i integritet informacija. Bitno je redovito ažurirati softver i firmware uređaja kako bi se sprječili sigurnosni propusti. I na to sve treba implementirati sigurnosne protokole na razini mreže, kao i osigurati fizičku sigurnost uređaja.

6. **Standardi i protokoli** Standardi i protokoli igraju veliku ulogu u ekosustavu IoT-a, jer trebaju omogućiti interoperabilnosti i pouzdanu komunikaciju između različitih uređaja i sistema.
   Jedan od osnovnih protokola u IoT-u je MQTT [6] koji omogućava efikasni prijenos poruka između uređaja. Koristi se CoAP[7], dizajniran za jednostavne uređaje i mreže. Za lokalne mreže se koristi Zigbee, protokol koji omogućava nisku potrošnju energije i efikasnu mrežnu komunikaciju. LoRaWAN[8] je protokol za WAN dizajniran da podržava velike mreže, kao što su pametni gradovi, sa milijunima uređaja koji imaju nisku potrošnju.
   Standardi koje se koriste su IEEE 802.15.4, koji se odnosi na bežične mreže s niskom brzinom prijenosa podataka, i BLE [9] za kratkodometnu komunikciju.Ovi standardi osiguravaju da uređaji mogu učinkovito komunicirati, uz minimalnu potrošnju energije, što je posebno važno za uređaje koji rade na baterije.

[6]MQTT - Message Queuing Telemetry Transport
[7]CoAP-Constrained Application Protocol
[8]LoRaWAN - Long-Range Wide Area Network
[9]BLE- Bluetooth Low Energy

7. **Primjer IoT sustava**



Slika 3.3: IoT sustav skuplja podatke od senzora instaliranih u IoT uređaju i šalje ih kroz IoT gateway da se ti podaci analiziraju od strane aplikacije ili back-end sustava. Slika je preuzeta sa izvora [4]

3.4 Primjena Interneta stvari (IoT) u svakodnevnom životu

Internet stvari (IoT) pronalazi svoje mjesto u mnogim aspektima našeg svakodnevnog života, pružajući pametna i automatizirana rješenja. Evo nekoliko primjera kako IoT utječe na svakodnevni život:

- **Pametni domovi**
  Najčešće vidljiva primjena IoT-a je u pametnim domovima. IoT uređaji omogućuju automatizaciju i daljinsko upravljanje različitim aspektima domaćinstva - od pametnog termostata koji regulira temperaturu, pametnih žarulja, do sigurnosnih sustava koji omogućuju nadzor doma preko pametnih telefona.

- **Tehnologija koju nosimo**
  Pametni satovi su primjer IoT tehnologije koju nosimo. Oni mogu pratiti zdravstvene parametre kao što su otkucaji srca, kvaliteta sna, broj koraka i pružaju povratnu informaciju o fizičkom stanju korisnika.

- **Zdravstvena skrb**
  U zdravstvu, IoT uređaji omogućuju daljinski nadzor pacijenata, pružajući kontinuirane podatke o vitalnim funkcijama, što može biti ključno za pravovremeno otkrivanje i liječenje zdravstvenih problema.

- **Pametna vozila**
  U automobilskoj industriji, IoT omogućuje napredne funkcije popust sustava za pomoć pri parkiranju, autonomne vožne, napredne dijagnostike vozila, poboljšavajući sigurnost i iskutvo vožnje.

- **Pametna poljoprivreda**
  U poljoprivredi, IoT senzori mogu pratiti različite parametre, kao što su vlažnost tla, razina hranjivih tvari, te pomažu u automatiziranom navodnjavanju i upravljanju usjevima, čime se povećava učinkovitost i smanjuju troškovi održavanja.

- **Upravljanje energijom** IoT uređaji pomažu u optimizaciji potrošnje energije kroz uređaje koji reguliraju potrošnju energije u domaćinstvu.

4 LoRa i LoRaWAN tehnologija

4.1 Uvod u LoRa tehnologiju

LoRa je tehnologija koja se bazira na modulaciji radio valova za LPWAN[1]. Ime LoRa je referenca na 'long-range', jer omogućava slanje podataka na velike udaljenosti. Prvo ju je razvila tvrtka Cycleo i 2012. godine ju je kupila tvrtka Semtech. Semtech je dalje razvio tu tehnologiju da standardizira LPWAN mreže. LoRa omogućava komunikaciju u urbanom okruženju do 5 km i do 15 km u ruralnim područjima (dokle možemo vidjeti, engl. 'line of sight'). Ključna karakteristika LoRa tehnologije je što zahtjeva jako malo energije, što omogućava dizajniranje uređaja na baterije koji mogu trajati i do 10 godina. Uređaji koji koriste LoRa tehnologiju su povezani zvjezdastom mrežnom topologijom, koja se bazira na otvorenom LoRaWAN protokolu.
Na slici 4.1 možemo usporediti razlike između LoRa i drugih tehnologija koje se tipično koriste u IoT ili M2M[2] rješenjima za komunikaciju.

Slika 4.1: Mrežne tehnologije koje se koriste u IoT

[1]LPWAN - low-power wide area network
[2]M2M - machine to machine

4.1.1 Prednosti LoRa tehnologije

- **Dalek domet**

  LoRa gateway može primati i slati signale na udaljenosti veće od 15 km u nenaseljenim područjima. U gusto naseljenim područjima, signali mogu putovati i do 5 km, ovisno o tome koliko su duboko u nekoj infrastrukturi smješteni krajni uređaji.

- **Životni vijek baterije**

  Što se tiče baterije, energija potrebna da se šalju podaci je dosta malo, s obzirom da su sami paketi podataka jako mali i šalju se samo par puta na dan. Također kada su krajnji uređaji neaktivni, energetska potrošnja se mjeri u miliwatima (mW), što omogućava bateriji uređaja da traje godinama.

- **Kapacitet LoRaWAN mreže**

  Kapacitet LoRaWAN mreže može podržati milijune poruka, ali taj broj porukla ovisi o broju postavljenih pristupnika (engl. 'gateway'). Jedan 8-kanalni pristupnik može podržati do 1.5M poruka u periodu od 24h, što znači da ako jedan krajnji uređaj šalje jednu poruku svakih sat vremena, jedan takav pristupnik može podržati do 60 000 krajnjih uređaja. Ako u mreži imamo 10 pristupnika, to znači da mreža može podržati oko 100 000 uređaja i 1M poruka, ako je potreban veći kapacitet, sve što treba napraviti je dodati još pristupnika u mrežu.

4.2 Radio modulacija i LoRa

LoRa koristi tehniku modulacije širokog spektra izvedene iz postojeće tehnologije Chirp Spread Spectrum. LoRa radi u fiksnom kanalu propusnosti (engl. 'bandwidth') od 125kHz ili 500 kHz za uplink[3] kanale, te 500 kHz za downlink[4] kanale. LoRa koristi ortogonalni faktor širenja (engl. 'spreading factor', kratica je SF i biti će dodatno pojašnjen malo kasnije u radu). To omogućava mreži da poveća životni vijek baterija krajnjih uređaja, na način da prilagođava snagu i brzinu prijenosa podataka svakog pojedinog krajnjeg uređaja. Na primjer, krajnji uređaj koji se nalazi blizu pristupnika će slati podatke sa malim faktorom širenja (brži prijenos podataka), dok će krajnji uređaj koji je udaljen par km od pristupnika slati podatke s mnogo većim faktorom širenja (sporiji prijenos podataka).

U tradicionalnom sustavu ili DSSS(Direct Sequence Spread Spectrum) sustavu faza nosioca signala koji šalje odašiljač mijenja se prema kodnoj sekvenci kako je prikazano na slici 4.2. Signal koji nosi podatke se množi s unaprijed definiranim nizom bitovima (taj niz bitova se naziva kod širenja, engl. 'spreading code'[5]), s čime se stvara "brži" signal, koji ima veću frekvenciju od izvornog signala. To znači da se širina pojasa signala širi izvan širine pojasa izvornog signala. Kada odašiljani signal stigne u RF prijemnik, taj signal se opet množi s identičnom kopijom koda širenja korištene u RF odašiljaču, što onda rezultira replikom izvornog podatkovnog signala. Ovaj postupak množenja signala kodom sekvence omogućava veći RF 'link budget'[6], što drugim riječima znači da omogućava prijenos podataka na veće udaljenosti.

Processing gain (Gp[7]). - omogućava prijamniku da povrati izvorni podatkovni signal, čak i ako kanal ima negativni SNR[8]. Jedan od nedostataka klasičnog DSSS sustava je to što zahtijeva visoko precizan (i skup) referentni sat, za sinkronizaciju uređaja u mreži. LoRa CSS modulacija taj problem riješava sa generiranjem chirp signala koji kontinuirano varira u frekvenciji. Kako je prikazano na slici 4.3.

[3]uplink označava kanal za slanje podataka

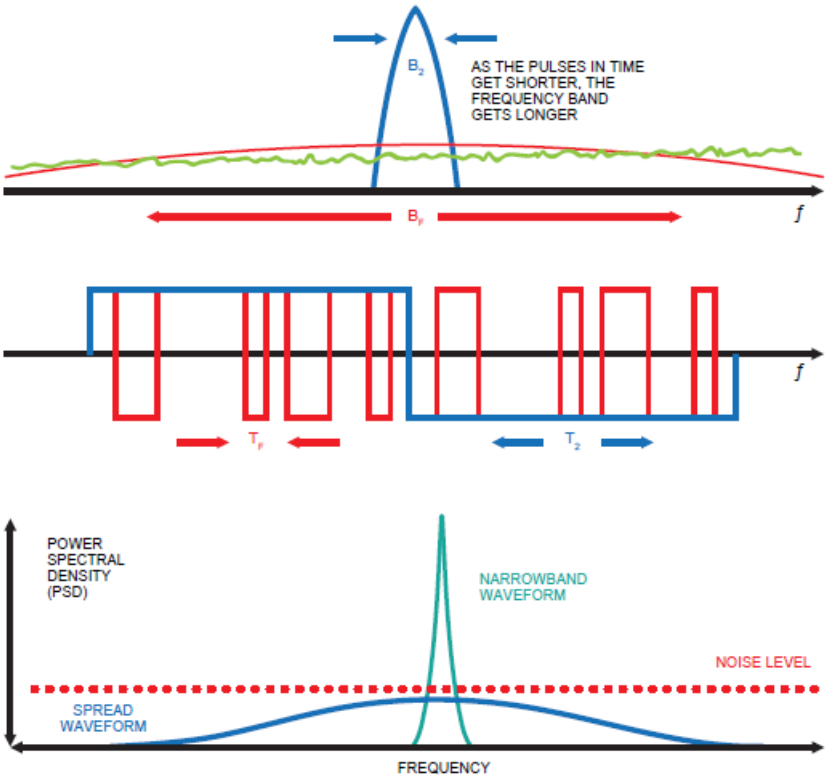
[4]downlink označava kanal za primanje podataka

[5]U slučaju LoRa CSS tehnologije spreading code je spreading factor

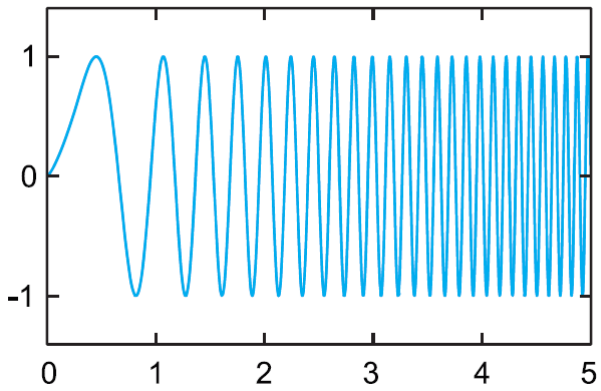
[6]link budget je proračun koji kvantitativno određuje da li će signal biti uspješno poslan.

[7]Gp je logaritam omjera brzine čipova kodne sekvnece i brzina bitova podatkovnog signala na bazi 10.

[8]SNR - Signal to Noise Ration



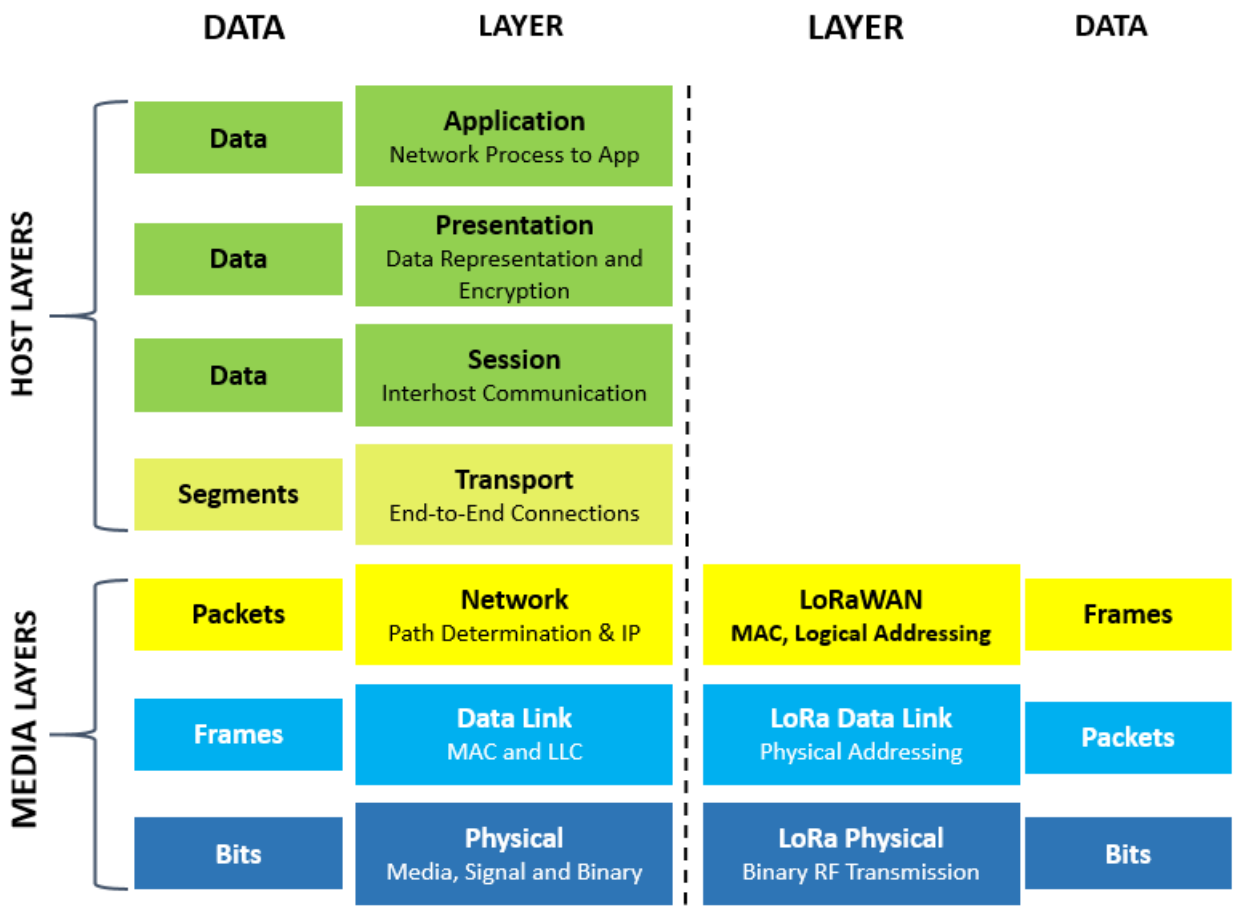
Slika 4.2: Promjene faze nosioca signala predajnika u DSSS sustavu. Slika je preuzeta sa izvora [11]



Slika 4.3: LoRa Chirp Spread Spectrum ilustracija. Slika je preuzeta sa izvora [11]

4.2.1 Pozicija LoRa-e u OSI modelu

U kontekstu OSI[9] modela, LoRa se nalazi u fizičkom sloju ili kolokvijalno u 'bit' sloju (prikazano na slici 4.4), koji je zadužen za prijenos i primanje podataka preko nekog medija. Za razliku od tradicionalnih mrežnih sustava, LoRa je bežična tehnologija. Transmisija podataka se odvija preko slanja radio valova iz RF transmitera koji je ugrađen u IoT uređaj do RF primatelja koji se nalazi u pristupniku. Time smo dobili dvosmjernu komunikaciju koja je jako korisna u IoT primjeni.



Slika 4.4: OSI 7-slojni model mreže. Slika je preuzeta sa izvora [11]

[9]OSI - Open Systems Interconnection je model računalne mreže prikazane u 7 slojeva

4.3 LoRa PHY signal specifikacije

Cilj LoRa PHY[10] signala je ostvariti veliku kvalitetu veze('link budget') sa malom energetskom potrošnjom za komunikacije na velike udaljenosti.

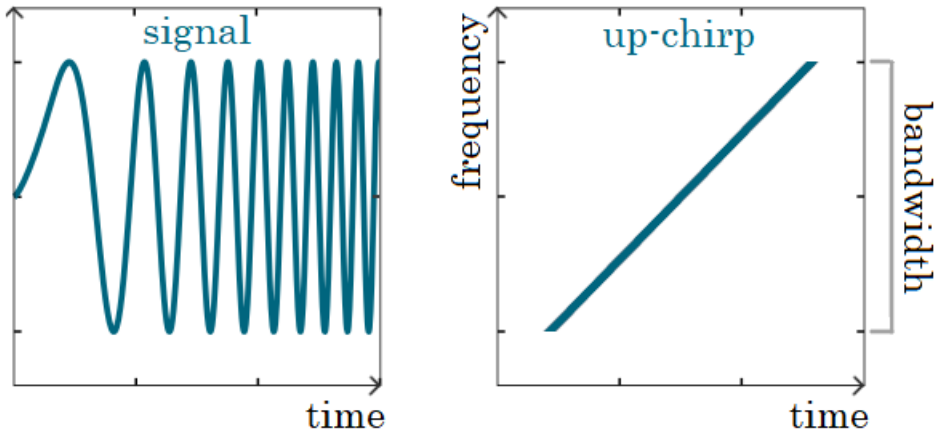
4.3.1 Bandwidth

Bandwidth je parametar u bežičnoj komunikacji koji opisuje frekvencijski raspon preko kojeg se prenose podaci. U našem slučaju bandwidth je dio frekvencijskog spektra kojeg prebrisuje chirp signal. Dopuštene frekvencije su 125kHz, 250kHZ i 500kHz. Označavam bandwidth sa slovom B.

4.3.2 Chirp signal

Chirp[11]signal je u suštini sinusoidalni val kojem se frekvencija mijenja u vremenu. Frekvencija signala se povećava ili smanjuje linearno, eksponencijalno, itd... . U slučaju CSS tehnologije, frekvencija signala se povećava ili smanjuje linearno, kao što je prikazano na slici 4.6. Chirp signali imaju široku primjenu. Od korišenja chirp signala u radarima i sonarima za precizno određivanje udaljenosti i brzine nekog objekta do korištenja u komunikacijskim tehnologijama (bežične mreže i mobilne komunikacije).

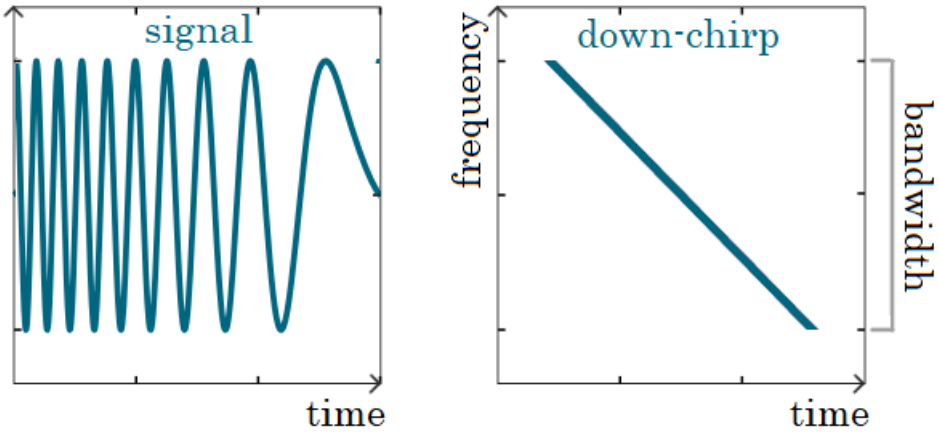
- Chirp signal gdje se frekvencija povećava u vremenu se naziva 'up-chirp'



Slika 4.5: Slika je preuzeta sa izvora [12]

- Chirp signal gdje se frekvecija smanjuje u vremenu se naziva 'down-chirp'

[10]LoRa Physical - je fizički signal u spektru radio frekvencije
[11]Chirp je akronim od 'Compressed High Intensity Radar Pulse'



Slika 4.6: Slika je preuzeta sa izvora [12]

4.3.3 SF - Spreading factor

U terminima LoRe, SF je broj bitova po simbolu. LoRa modulacija sve zajedno ima 6 faktora širenja (SF7 do SF12). Što je veći faktor širenja, to će dalje signal moći putovati. Na slici 4.7 je prikazana tablica za 4 različita faktora širenja [SF7...SF10] koji se mogu koristiti za uplink poruke na kanalu 125kHz. Tablica prikazuje odnos između faktora širenja, brzine prijenosa podataka u bitovima, udaljenosti (udaljenost prijenosa podataka ovisi o području, postići ćemo veće udaljenosti prijenosa podataka u ruralnom području nego u urbano naseljenom područj) i koliko dugo će signal koji prenosi 11-bajtova veliki paket putovati (TOA - 'time on air).

Spreading Factor (For UL at 125 KHz)	Bit Rate	Range (Depends on Terrain)	Time on Air for an 11-byte payload
SF10	980 bps	8 km	371 ms
SF9	1760 bps	6 km	185 ms
SF8	3125 bps	4 km	103 ms
SF7	5470 bps	2 km	61 ms

Slika 4.7: Tablica LoRa faktora širenja, slika je preuzeta sa izvora [11]

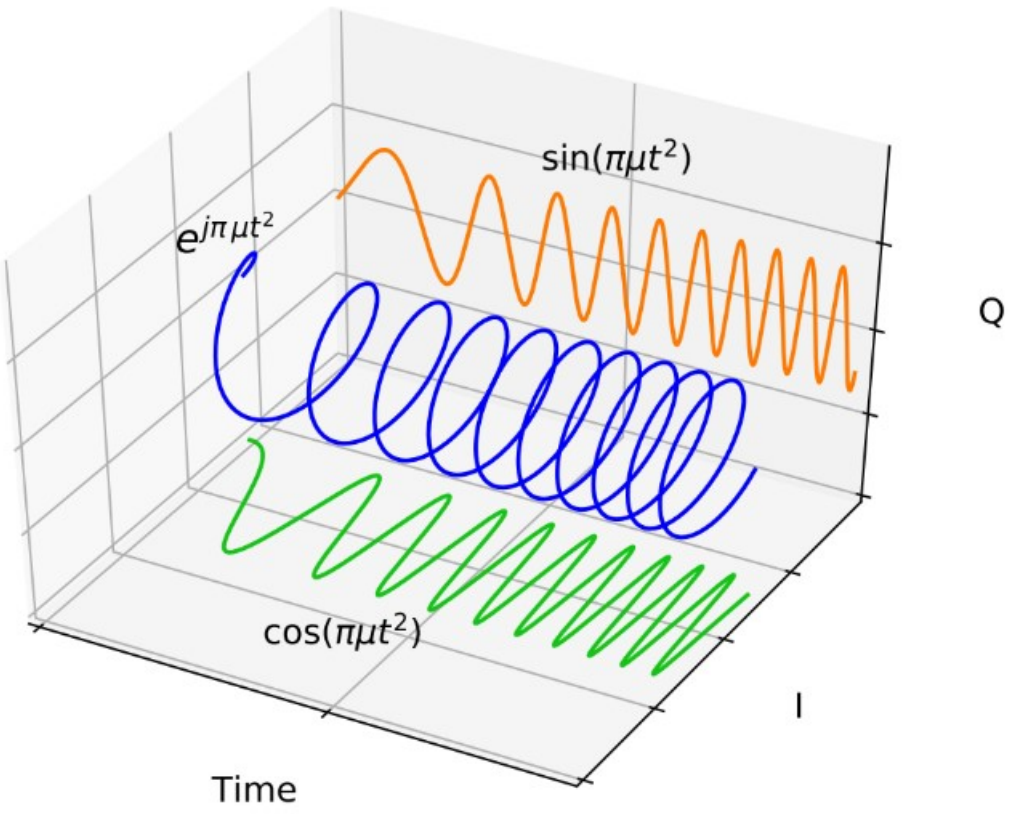
Bitno je naglasiti da su faktori širenja ortogonalni. To znači da kada moduliramo signale na istoj frekvenciji sa različitim faktorima širenja, signali neće međusobno interferirati. Signali na istoj frekvenciji sa različitim SF-om će se međusobno doživljavati kao šum.

4.3.4 Frequency Shift - Chirp Spread Spectrum - modulacija LoRa PHY signala

Bazni chirp signal možemo opisati kako kompleksni sinusoidalni val na sljedeći način:

$$x(t) = e^{j\pi\mu t^2} \tag{4.1}$$

Signal je prikazan na slici 4.8. Možemo uočiti da se faza signala ne mijenja linearno kroz vrijeme već kvadratno. Zbog toga se signal čini sve 'stisnutijim' udesno. Iz Eulerove relacije, realni signal $cos(\pi\mu t^2)$ vidimo kao projekciju kompleksnog sinusoidalnog vala na realnu ravninu. A imaginarni dio signala $sin(\pi\mu t^2)$ vidimo kao projekciju kompleksnog sinusoidalnog vala na kompleksnu ravninu.



Slika 4.8: Osnovni chirp sa realnim i imaginarnim signalom, slika je preuzeta sa izvora [16]

4.3.5 Trenutna frekvencija

Opći izraz za chirp signala, zanemarujući amplitudu, pišemo kao:

$$x(t) = e^{j(\pi\mu t^2 + 2\pi f t + \theta)} \tag{4.2}$$

Faktor π je zapravo 2π skaliran za $1/2$. To radimo zato da možemo normalizirati chirp rate[12] kao linearnu funkciju trenutne frekvencije. Trenutna frekvencija je definirana kao derivacija trenutne faze iz jednadžbe (4.2).

$$\phi(t) = \frac{1}{2\pi}\frac{d}{dt}(\pi\mu t^2 + 2\pi f t + \theta) = \mu t + f \tag{4.3}$$

Možemo uočiti da postoje tri različita parametra u izrazu za trenutnu fazu koja možemo mijenjati za slanje podataka.

- Faza θ
- Frekvencija $\omega = 2\pi f$
- Chirp rate μ

4.3.6 Modulacija

Mijenjajući samo frekvenciju[13], podatke(simbole) možemo slati kroz set od M frekvencija f_m, gdje je $m = 0, 1, ..., M-1$. Set M je definiram kao $M = 2^{SF}$. Za $M = 4$ možemo slati 4 različita signala, gdje svaki signal može prenositi 2 bita (to slijedi iz relacije $\log_2 4 = 2$). Iz jednadžbe (4.2) možemo napisati:

$$x(t) = e^{j(\pi\mu t^2 + 2\pi f_m t + \theta)} \tag{4.4}$$

Prema jednadžbu (4.3) onda vidimo da je trenutna frekvencija dana kao:

$$\phi_m(t) = \mu t + f_m, \quad m = 0, 1, ..., M-1 \quad \text{ili} \quad m = -\frac{M}{2}, ..., \frac{M}{2} - 1 \tag{4.5}$$

[12]Chirp rate opisuje u kojem vremenu T_m bi signal trebao proći kroz dani spekar bandwidtha, definira se kao $\mu = \frac{B}{T_m}$

[13]Kada bi htjeli koristiti sve parametre, uključujući amplitudu, μ, f i θ za kodiranje informacija, dobili bi jako kompleksni modem koji ne bi bio prikladan iz perspektive dizajna i same aplikacije. drugim riječima preskupo i nepoželjno u IoT.

Ako uzmemo $M = 4$ kao primjer. Unutar širine pojasa frekvencija B (engl. 'bandwidth'), te četiri frekvencije zapisujemo kao

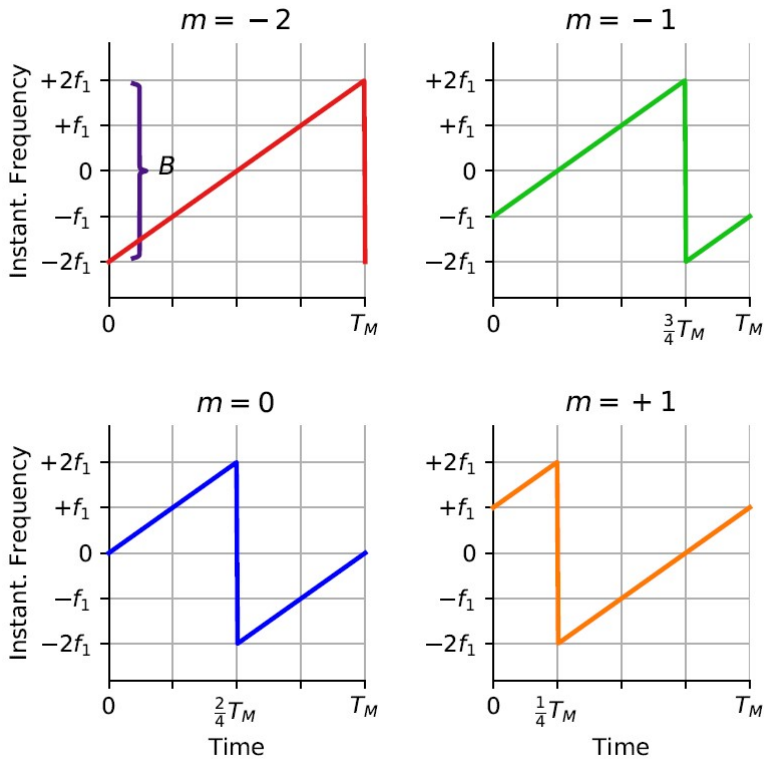
$$f_m = mf_1, \quad f1 = \frac{B}{M} = \frac{B}{4} \tag{4.6}$$

pri čemu je $m = -2, -1, 0, 1$.

Svakoj od tih frekvencija možemo dodjeliti kombinaciju dva bita kao:

$$\begin{aligned} f_{-2} = -2f_1 = -\tfrac{B}{2} \quad &\rightarrow \quad 00 \\ f_{-1} = -1f_1 = -\tfrac{B}{4} \quad &\rightarrow \quad 01 \\ f_0 = 0 \cdot f_1 = 0 \quad &\rightarrow \quad 10 \\ f_{+1} = +1f_1 = +\tfrac{B}{4} \quad &\rightarrow \quad 11 \end{aligned} \tag{4.7}$$

Na slici 4.9 su prikazane 4 moguće trenutne frekvencije za $\mu t + f_m$.

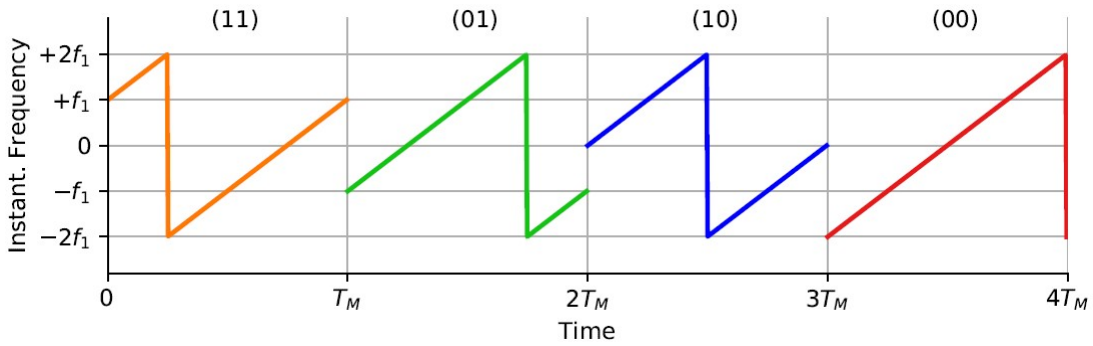


Slika 4.9: Trenutna frekvencija kao funkcija vremena za $M = 4$ FS-CSS simbole. Slika je preuzeta sa izvora [16]

Možemo vidjeti da je raspon fekvencije dan kao $-2f_1 = -\frac{B}{2}$ do $+2f1 = +\frac{B}{2}$. Prema tome vidimo da se modulacija ostvaruje na način da radimo linearni 'sweep' frekvencije signala unutar zadanog bandwidth-a.

T_m je vrijeme jednog simbola. To je vremenski period u kojem šaljemo jedan od svih mogućih moduliranih simbola. Možemo uočiti na slici 4.9 kako trenutna frekvencija radi skok na umnošku $\frac{1}{M} = \frac{1}{4}$ puta vrijeme simbola T_m za svaki m.

Primjer signala koji nosi bitnu sekvencu $11\ 00\ 10\ 00$ je prikazan na slici 4.10.



Slika 4.10: Graf trenutne frekvencije u vremenu za bitove 11011000. Slika je preuzeta sa izvora [16]

4.3.7 Demodulacija

Kada chirp signal dođe do prijamnika, bazni dio signala se briše, jer ne nosi informaciju. Množimo jednadžbu (4.2) sa konjugiranom bazom chirp signala:

$$e^{j(\pi\mu t^2+2\pi_m t+\theta)} \cdot e^{-j\pi\mu t^2} = e^{j(2\pi f_m t+\theta)} \tag{4.8}$$

Od originalnog signala ostaje modulirana frekvencija i pomak u fazi. Argument u eksponentu je pravac koji nam sa svojim nagibom otkriva frekvenciju f_m[14], koju onda možemo prevesti u bitove.

Ja u ovom radu neću dalje ulaziti u detalje kako se detektira i sinkronizira zaprimljeni signal.

[14]Pronalaženje nagiba pravca f_m je isti problem kao pronalaženje kompleksnog sinusoidalnog vala u nekom šumu

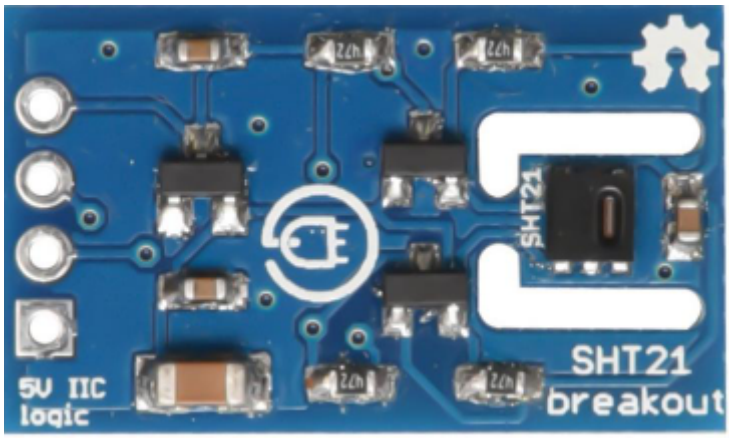
4.4 LoRaWAN

LoRaWAN je Media Access Control (MAC) protokol koji se gradi na LoRa PHY modulaciju. To je sofverski sloj koji definira kako uređaji koriste LoRa hardver, na primjer kada se šalje signal, kako se formatira poruka, itd... .
LoRaWAN protokol je razvijen i održavan od strane LoRa Alliance. [15]

[15] LoRa Alliance je neprofitna organizacija koji se brine o LoRaWAN protokolima i izdaje certifikate za krajnje uređaje koji koriste LoRaWAN.

5 SHT-21 breakout pločica

Senzor za mjerenje temperature i vlage se nalazi na SHT-21 breakout pločici prikazanoj na slici 5.1. Brekaout pločica nam omogućava spajanje senzora na Arduino uređaj putem I2C logike.



Slika 5.1: SHT-21 breakout pločica

5.1 SHT21 senzor

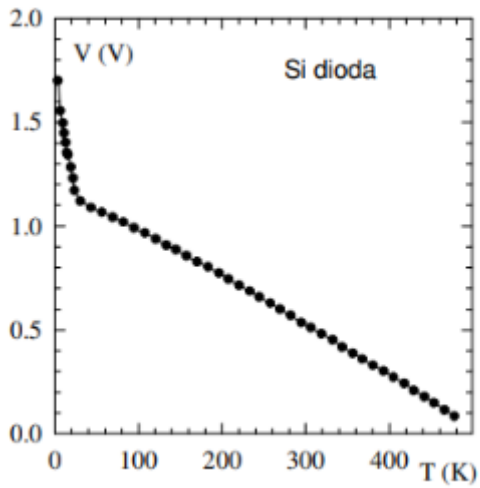
SHT21 senzor služi za mjerenje relativne vlagu i temperaturu zraka. Senzor se sastoji od dva senzora: kapacitivni senzor za mjerenje relativne vlage zraka i silicijsko diodnog temperaturnog senzora.

5.1.1 Silicijski diodni termometar

Silicijeve diode služe kao termometri u širokom temperaturnom području (1-400 K). Koriste se u režimu stalne struje, tj. pri konstantnoj struji (tipično $10\ \mu\mathrm{A}$) kroz diodu, mjeri se napon diode, koji ovisi o temperaturi. Na slici 5.2 je prikazana temperaturna ovisnost napona silicijeve diode. U tablici 5.1 je prikazana preciznost mjerenja i u kojem temperaturnom rasponu senzor SHT21 radi.

Preciznost mjerenja temperature zraka	0.3 °C
Operativni raspon mjerenja	-40 do 125 °C

Tablica 5.1: Specifikacija SHT21 senzora za mjerenje temperature



Slika 5.2: Temperaturna ovisnost napona Si diode. Slika je preuzeta sa izvora [17]

5.1.2 Kapacitivni senzor za mjerenje relativne vlage

Kapacitivni senzor za mjerenje relativne vlage je elektronički uređaj koji mjeri relativnu vlagu (RH) u okolini koristeći princip električnog kapaciteta. Senzor radi na način da između ploča kondenzatora ima dielektrični materijal osjetljiv na vlagu. Sa promjenom razine vlage mijenja se dielektrična konstanta i samim time kapacitet kondenzatora. Ako se koristi u LC krugu, frekvencija kruga će se mijenjati ovisno o promjeni kapaciteta. Korištenjem sklopa koji pretvara frekvenciju u analogni napon, možemo jednostavno detektiranjem promjene napona, detektirati promjenu kapaciteta. Taj signal se onda obrađuje u mikrokontroleru i pretvara izlaz u mjerenje relativne vlage zraka u postocima. (%RH). Na tablici 5.2 su prikazana preciznost i raspon mjerenja SHT21 senzora.

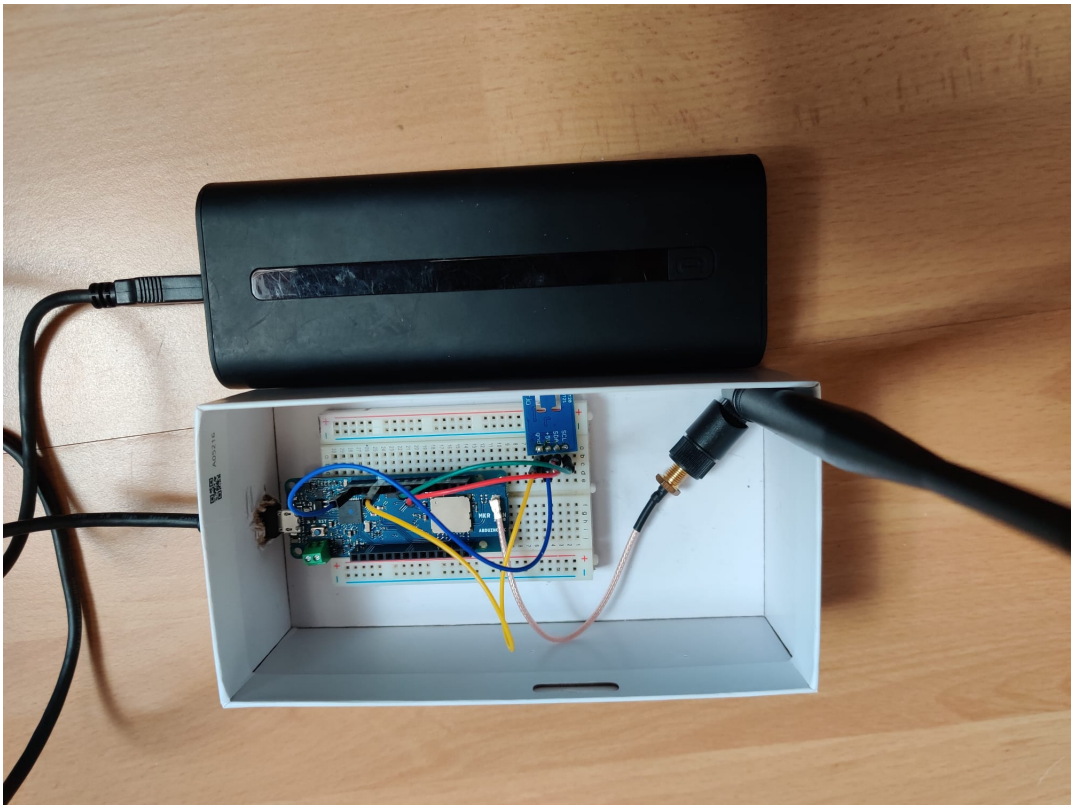
Relativna vlaga zraka	
Preciznost mjerenja relativne vlage zraka	2 %RH
Operativni raspon mjerenja	0-100 %RH

Tablica 5.2: Specifikacija SHT21 senzora za mjerenje relativne vlage zraka

6 Mjerenja

6.1 Postav mjerenje

Za mjerenje temperature i relativne vlage zraka sam koristio MKRWAN 1300 mikrokontroler spojen sa SHT21 senzorom, antenom i powerbankom. Kao što je prikazano na slici 6.1



Slika 6.1: MKRWAN 1300 spojen sa SHT21 breakout-om, antenom i powerbank-om

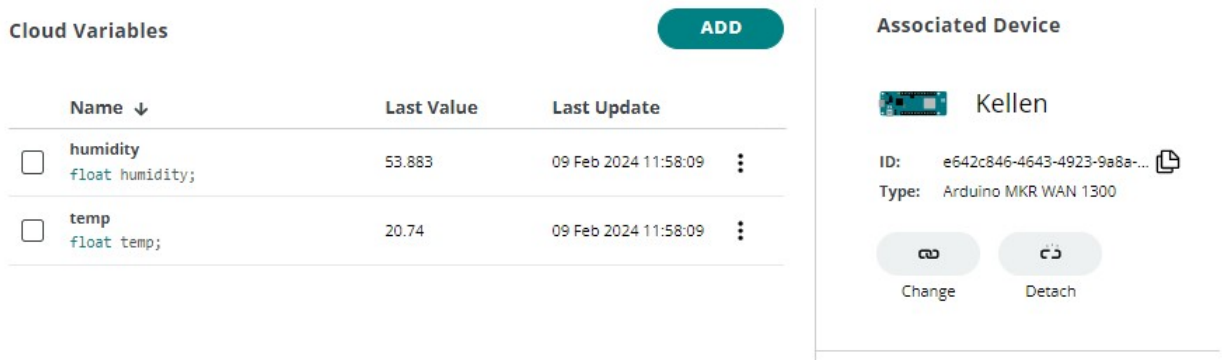
6.1.1 Spajanje mikrokontrolera sa senzorom

SHT21 koristi I2C protokol za komunikaciju, tako da pinovi koji su korišteni za spajanje su:

- VDD pin
- GND pin
- SCL pin
- SDA pin

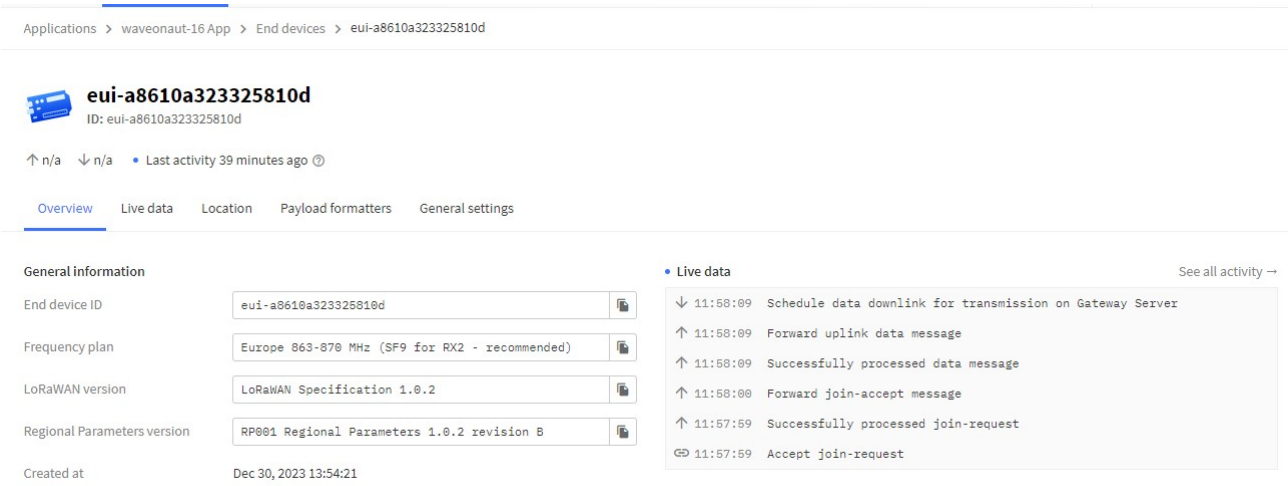
6.1.2 Komunikacija sa mikrokontrolerom

Mikrokontroler sam povezao sa Arduino Cloud-om kao 'Thing'. Pri registraciji uređaja na Arduino Cloud, uređaj se automatski registrira na The Things Network (koji omogućava praćenje aktivnosti uređaja i omogućava korištenje LoRa tehnologije). Korištenjem Arduino Cloud-a sam zaobišao korištenje Arduino IDE i nisam morao paziti koje library trebam instalirati. Kada je uređaj registriran, u Arduino cloud-u sam napravio dvije 'cloud' varijable tipa podataka float ('temp' i 'humidity') koje sam onda koristio za očitavanje prikupljenih podataka od strane senzora. Na slici 6.2 se vidi povezan uređaj sa varijablama koje će biti rezultati mjerenja.



Slika 6.2: Sučelje Arduino Cloud-a

The Things Network je LoRaWAN mrežni server. Kada je uređaj povezan sa TTN mrežom, mreža pruža sučelje preko The Things Stack Cloud-a na kojem mogu pratiti aktivnost svojeg uređaja. Kao što je prikazano na slici 6.3



Slika 6.3: Sučelje The Things Stack Cloud-a

6.1.3 Programiranje mikrokontrolera

Korišteni library su:

- thingProperties.h
- SHT21.h
- Wire.h

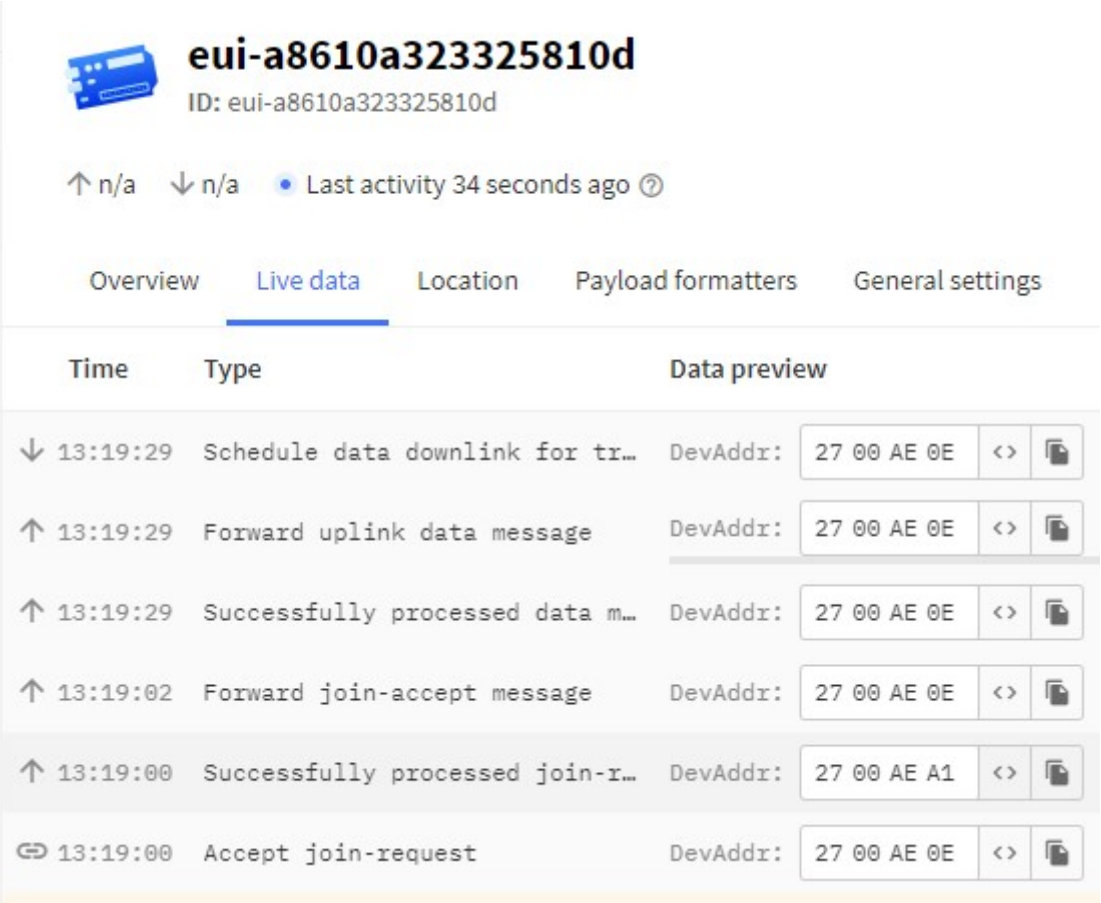
Kod koji sam koristio za programiranje mikrokontrolera je prikazan na slici 6.4.

```
#include "thingProperties.h"
#include <SHT21.h>
#include<Wire.h>

SHT21 sht;

void setup() {

  //Započni I2C protokol komunikacije
  Wire.begin();
  /* Ovaj delay radim da Serial monitor ne blokira pokretanje programa,
     pošto ne koristim Serial Monitor
  */
  delay(1500);
  // Definirano u thingProperties.h
  initProperties();
  // Spajanje na Arduino Cloud
  ArduinoCloud.begin(ArduinoIoTPreferredConnection, false);

 /*
  Ova funkcija nam daje uvid u stanje mreže i povezivanje sa Arduino Cloudom.
  Što je veći broj, informacija je više granularna.
  Postavljena vrijednost je 0 (to su samo greške)
  Maksimum je 4
  */
  setDebugMessageLevel(0);
  ArduinoCloud.printDebugInfo();
}

void loop() {
  //Šalje update o cloud varijablama u Arduino Cloud
  ArduinoCloud.update();

  //temp i humidity su cloud varijable i već im je određen tip podatka na Cloud-u
  temp = sht.getTemperature();  // očitavanje temperature sa SHT21
  humidity = sht.getHumidity(); // očitavanje relativne vlage zraka sa SHT21

}
```

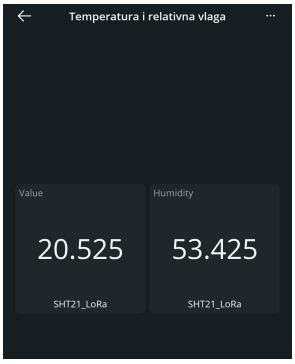
Slika 6.4: Screenshot koda koji sam koristio za programiranje mikrokontrolera

Interpretacija koda : kada se uređaj upali, pokušati će se povezati sa Arduino Cloud-om (koji je također povezan sa The Things Cloudom). Ako je povezivanje uspješno, uređaj sprema podatke mjerenja u cloud varijable 'temp' i 'humidity' i ažurira

ih na Arduino Cloud. Nakon toga se uređaj gasi. Kako to izgleda na The Things Cloudu je prikazano na slici 6.5. Ako je sve uspješno prošlo, moje cloud varijable će biti ažurirane i prikazane na Arduino Cloud dashboard-u, što je prikazano na slici 6.6.



Slika 6.5: Prikaz aktivnosti uređaja na The Things Stack Cloud-u

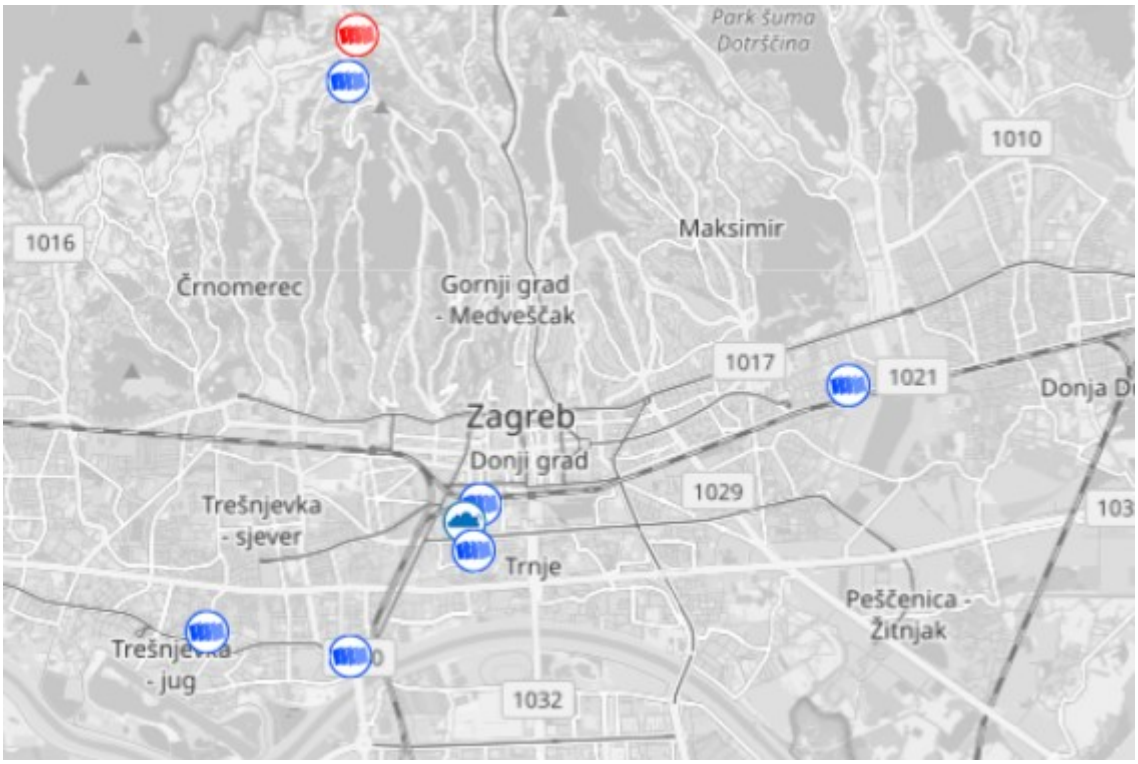


Slika 6.6: Prikaz cloud varijabli na Arduino Cloud dashboard-u

6.1.4 The Things Network heatmap

TTN heatmap daje informaciju gdje su postavljeni gateway-i u svijetu za LoRa komunikaciju. Tu mapu sam koristio da vidim odakle mogu komunicirati sa LoRaWAN

mrežom. Ako u području nema postavljenih gateway-a, onda spajanje nije moguće i ja bi trebao sam postaviti gateway za to područje. Također sam koristio tu mapu da vidim s kojim gateway-om je moj uređaj komunicirao.TTN heatmap sa prikazom dostupnih gateway-a u gradu Zagrebu je prikazan na slici 6.7.



Slika 6.7: Screenshot TTN heatmap-a koji prikazuje dostupne gateway-e na području grada Zagreba

6.2 *Rezultati mjerenja*

6.2.1 Mjerenje u park šumi Grmošćica

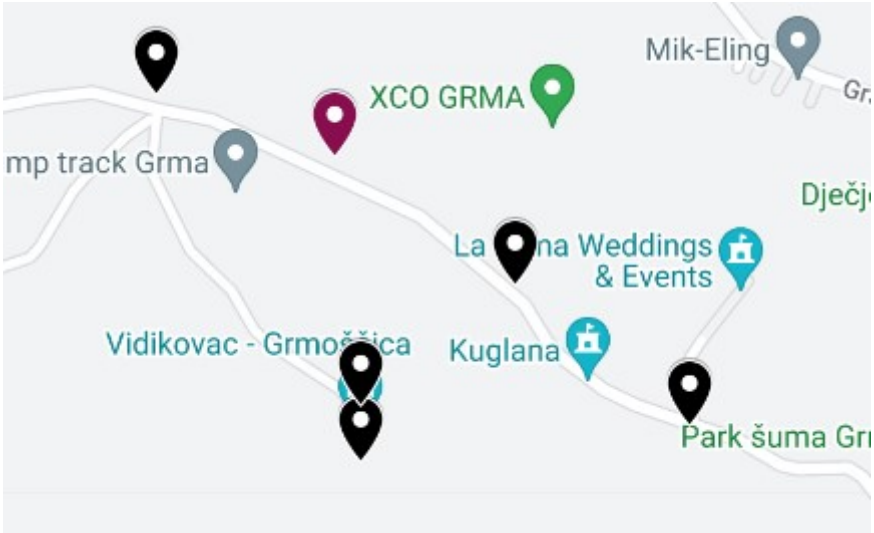
Mjerenje provedeno u park šumi Grmošćica je prikazano u tablici 6.1. Napravio sam 7 mjerenja, od čega je 1 mjerenje bilo neuspješno. Lokacije mjerenja su prikazane na slici 6.9 (Crni pinovi na karti označavaju uspješno poslan signal, dok ljubičasti pin označava lokaciju sa koje nisam mogao poslati signal). Signal je bio slab; RSSI je bio u rasponu od -103 do -108. Signal je bio uspješno detektiran sa SNR-om od 10.8 do -20. Neuspješno mjerenje je radi lošeg 'line of sight-a'. Prosječna udaljenost s koje je uređaj komunicirao s gateway-om je $\approx$ 3.26km, kao što je prikazano na slici 6.9. DHMZ-ovi službeni podaci za 10.2.2024 na Maksimiru su prikazani u tablici 6.2. Prva tri mjerenja temperature koja su bila uspješna na Grmošćici odstupaju od podataka mjerenja na Maksimiru, dok se mjerenja relativne vlažnosti poklapaju. Mjerenja od 13:30 do 13:51 se poklapaju sa službenim podacima mjerenja na Maksimiru.

Vrijeme mjerenja	Lokacija	Temperatura zraka (°C)	Relativna vlažnost (%)
12:47	45°49'3,96" N 15°55'12" E	20,86	44,15
13:03	45°49'7,42" N 15°55'5,95" E	20,89	46,97
13:13	45°49'3,105" N 15°55'0,48" E	Neuspješno mjerenje	Neuspješno mjerenje
13:20	45°49'12" N 15°55'53,34" E	20,13	48,87
13:29	45°49'4,98" N 15°55'59,87" E	18,88	46,72
13:35	45°49'4,44" N 15°55'0,54" E	18,48	46,92
13:51	45°48'58,51" N 15°55'26,77" E	18,85	47,45

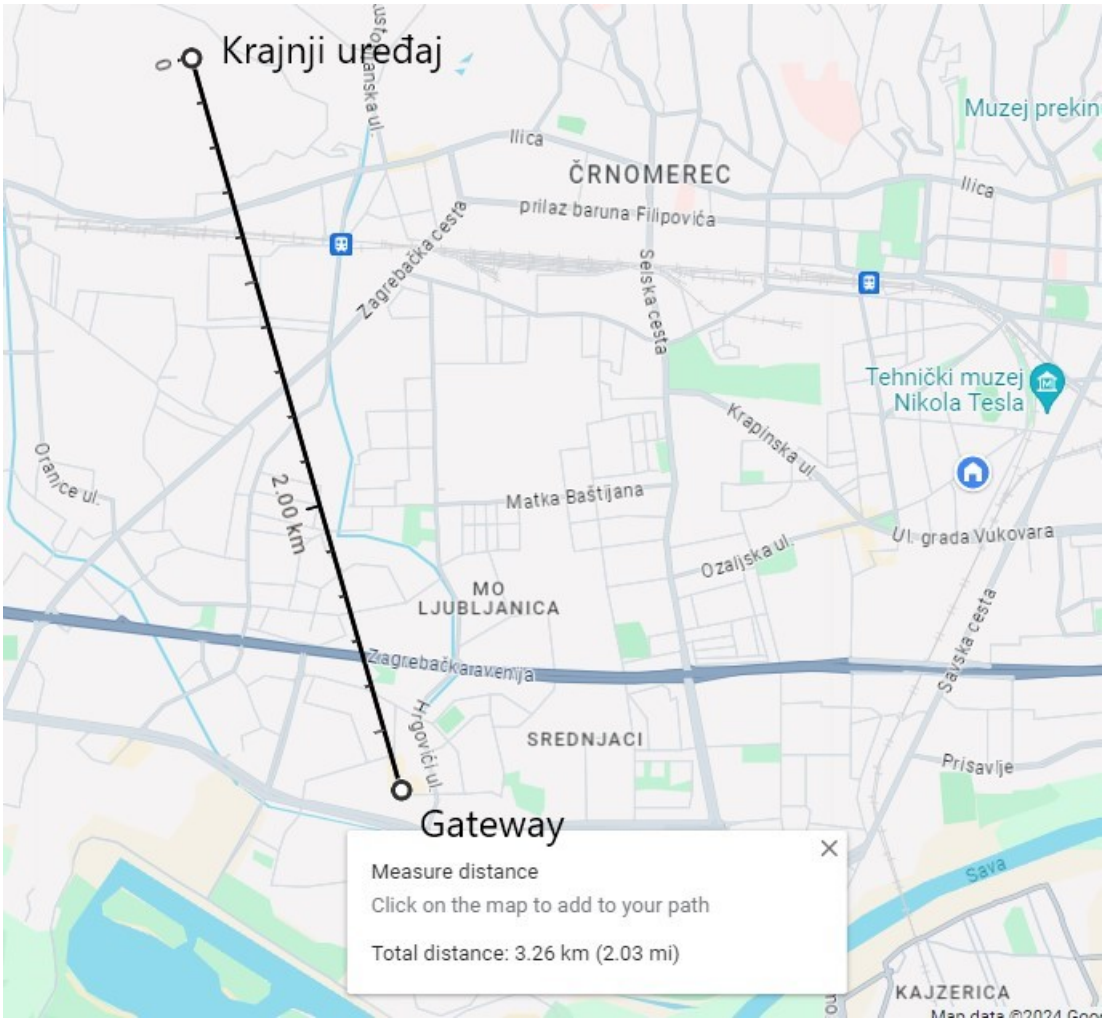
Tablica 6.1: Tablica mjerenja u park šumi Grmošćica 10.2.2024.

Sat	Temperatura zraka (°C)	Relativna vlažnost (%)
12	17,4	45
13	18,2	43
14	17,8	43
15	16,8	47
16	16,0	51

Tablica 6.2: Službeni podaci DHMZ-a 10.2. Maksimir



Slika 6.8: Lokacije mjerenja u park šumi Grmošćica



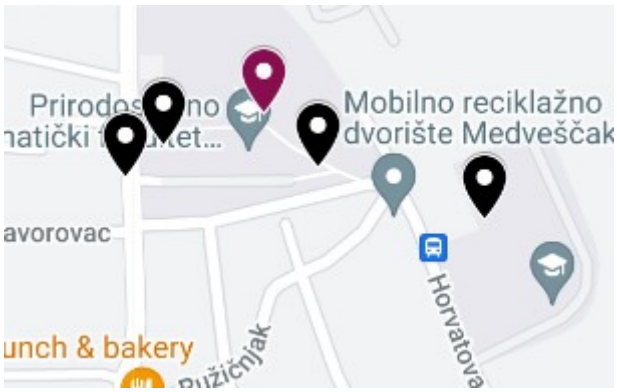
Slika 6.9: Prosječna udaljenost između mojeg uređaja i gatewaya s kojim je uređaj komunicirao $\approx$ 3.26 km

6.2.2 Mjerenje kod PMF Fizičkog odsjeka

Mjerenje u okolici PMF Fizičkog odsjeka je prikazano u tablici 6.3. Napravio sam 5 mjerenja, lokacije mjerenja su prikazane na slici 6.10 (Crni pinovi na karti označavaju uspješno poslan signal, dok ljubičasti pin označava lokaciju sa koje nisam mogao poslati signal). Signal je bio dosta slab; RSSI je bio od -114 do-116. Signal je bio uspješno detektiran 4 puta sa SNR-om[1] od 0.2 do -4. Neuspješno mjerenje je radi slabog 'line of sight'. Prosječna udaljenost uređaja od gatewaya s kojim je komunicirao je bila $\approx$ 3.39 km , kao što je prikazano na slici 6.11.
DHMZ-ovo službeno mjerenje 10.2. na Maksimiru je u 15 sati bilo 16.8°C i 47% relativne vlažnosti, kao što je prikazano na tablici 6.2. Prvo mjerenje ima veliko temperaturno odstupanje. Razlog tome je što je senzor bio u autu i nisam mu dao dovoljno vremena da se 'aklimatizira'. Zadnja dva mjerenja temperature su puno preciznija i odgovaraju službenim podacima DHMZ-a. Mjerenje relativne vlažnosti se isto poklapa sa DHMZ-ovim podacima.

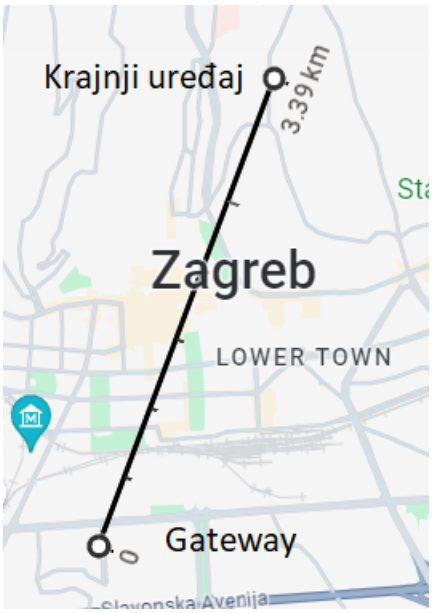
Vrijeme mjerenja	Lokacija	Temperatura zraka(°C)	Relativna vlažnost (%)
15:26	45°49'36.24" N 15°59'06.08" E	19,14	46.41
15:36	45°49'37.08" N 15°59'07.38" E	17.55	50.65
15:44	45°49'37.79" N 15°59'10.97" E	Neuspješno mjerenje	Neuspješno mjerenje
15:48	45°49'36.54" N 15°59'12.81" E	16.21	54.72
15:54	45°49'35.25" N 15°59'18.65" E	16.33	54.18

Tablica 6.3: Tablica mjerenja na lokaciji PMF Fizičkog odsjeka 10.2.2024.



Slika 6.10: Lokacije mjerenja kod PMF Fizičkog odsjeka

[1]SNR - Signal to Noise Ratio



Slika 6.11: Prosječna udaljenost mojeg uređaja do gateway-a s kojim je uređaj komunicirao. Prema Google maps-u je udaljenost ≈ 3.39 km

6.2.3 Mjerenje na Sljemenu

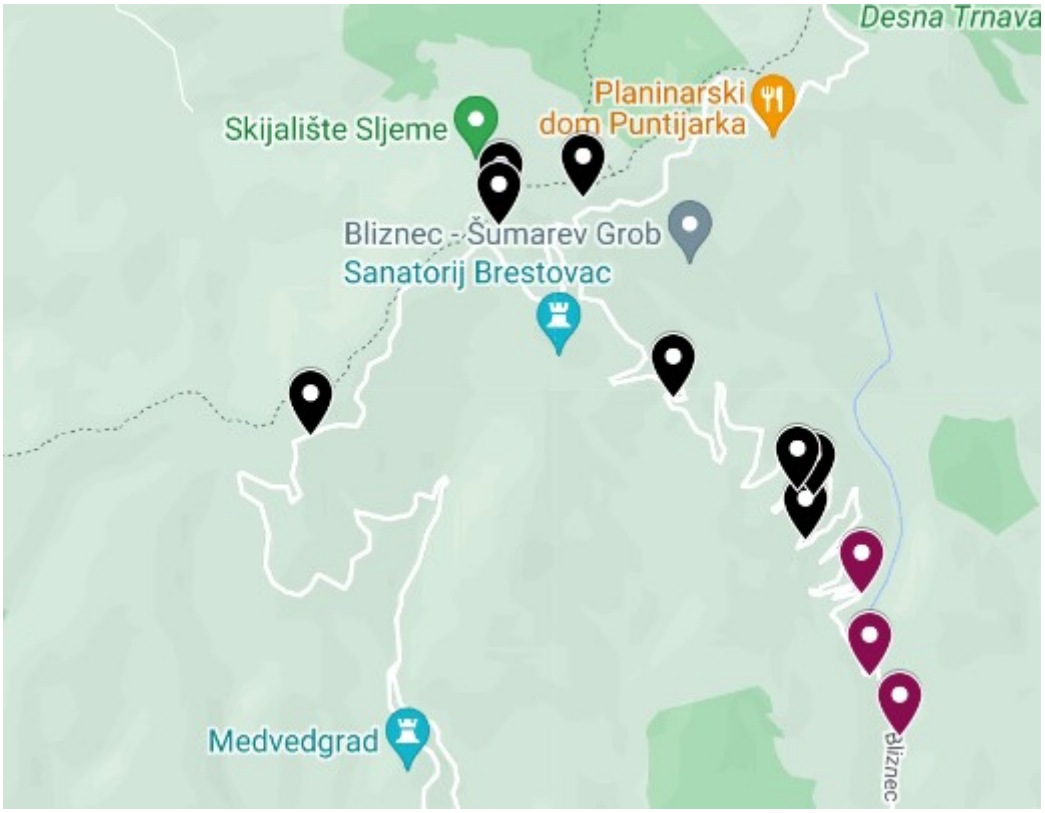
Podaci mjerenja sa Sljemena su prikazani u tablici 6.4. Lokacije mjerenja su prikazane na slici 6.12 (Crni pinovi na karti označavaju uspješno poslan signal, dok ljubičasti pinovi označavaju lokacije sa kojih nisam mogao poslati signal). Na nižoj nadmorskoj visini su 3 neuspješna mjerenja radio lošeg 'line of sight'-a. Na nižoj nadmorskoj visini prvi uspješno poslani signal je bio dosta slab sa RSSI = -108. Kako sam radio mjerenja na sve većoj nadmorskoj visini tako se i kvaliteta signala popravljala, pa je veliku većinu mjerenja RSSI = -60. SNR je bio negativan u svim mjerenjima u rasponu od -2.5 do -17.5, ali je signal bio uspješno detektiran. Najveća udaljenost s koje je moj uređaj uspio poslati signal gateway-u je bila 11.37 km, kao što je prikazano na slici 6.13. Službeni podaci mjerenja DHMZ-a na Puntijarci su prikazani u tablici 6.5. Prva dva uspješna mjerenja je temperatura bila puno viša od realnosti, to je iz razloga što se senzor nije stigao prilagoditi vanjskim uvjetima. Mjerena temperatura je postajala manja, kako sam dobivao na nadmorskoj visini, sve do zadnjeg mjerenja koje je napravljeno po povratku sa vrha Sljemena i ima smisla da je temperatura veća nego na vrhu. Moja mjerenja se ne poklapaju sa mjerenjima na Puntijarci, što ima smisla za mjerenja koja sam radio dok sam se penjao prema vrhu Sljemena, ali najniže mjerenje koje sam zabilježio blizu vrha je bilo 8.2°C sa relativnom vlažnosti zraka 86,6 %, za razliku od 7.2°C i kiše zabilježene na Puntijarci.

Vrijeme mjerenja	Lokacija	Temperatura zraka (°C)	Relativna vlažnost (%)
13:05	45°52’ 15.7” N 15°58’43.73” E	Neuspješno mjerenje	Neuspješno mjerenje
13:12	45°52’27.72” N 15°58’35.36” E	Neuspješno mjerenje	Neuspješno mjerenje
13:19	45°52’43.88” N 15°58’33.46” E	Neuspješno mjerenje	Neuspješno mjerenje
13:31	45°52’54.48” N 15°58’17.31” E	18.6	57.96
13:39	45°52’54.48” N 15°58’17.31” E	17,88	56,97
13:47	45°53’02.75” N 15°58’19.77” E	16.12	59.78
13:59	45°53’02.75” N 15°58’19.77” E	15.91	61.95
14:07	45°53’04.44” N 15°58’15.96” E	16.4	63.99
14:17	45°53’04.44” N 15°58’15.96” E	14.67	63.57
14:30	45°53’22.38” N 15°57’40.22” E	12.55	71.77
14:33	45°53’22.38” N 15°57’40.22” E	12.02	73.69
14:49	45°54’01.64” N 15°57’15.45” E	9.98	79.89
14:53	45°54’01.64” N 15°57’15.45” E	8.94	85.48
15:08	45°53’59.88” N 15°56’52.44” E	9.26	78.1
15:11	45°53’59.88” N 15°56’52.44” E	9.16	82.23
15:17	45°53’55.85” N 15°56’51.55” E	8.22	86.57
15:44	45°53’15.17” N 15°55’59.37” E	10.93	80.5

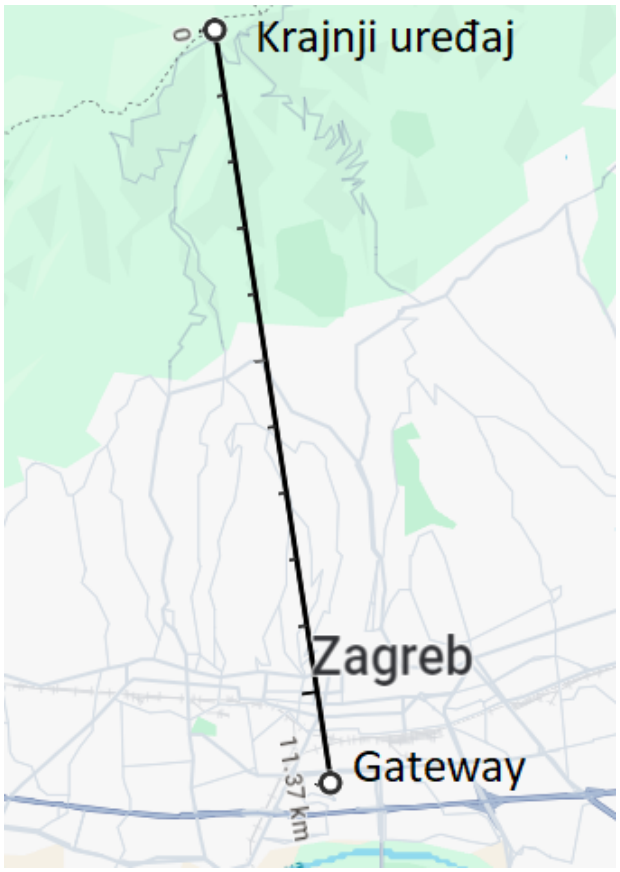
Tablica 6.4: Tablica mjerenja na području Sljemena 11.2.2024.

Sat	Temperatura zraka (°C)	Relativna vlažnost (%)
13	6,7	99
14	7,4	98
15	7,2	99
16	7,2	99

Tablica 6.5: Službeni podaci DHMZ-a; Puntijarka, 11.2.2024.



Slika 6.12: Lokacije mjerenja na Sljemenu



Slika 6.13: Najveća udaljenost s koje je moj uređaj komunicirao s gateaway-om je prema Google maps-u bila 11.37km.

6.3 Diskusija rezultata i prijedlozi poboljšanja

Koristeći MKRWAN 1300 sa SHT21 senzorom sam prikupljao podatke o temperaturi zraka i relativnoj vlažnosti. Koristeći LoRa tehnologiju sam ta prikupljenja mjerenja uspješno slao kao LoRaPHY signal prijemnicima u blizini. LoRa način komunikacije se pokazao odličnim za ovakav projekt, jer omogućuje komunikaciju na velike udaljenosti uz jako malu energetsku potrošnju. Smatram da je ova tehnologija jako dobra za bilo koji projekt koji treba slati podatke kao što su očitanje senzora i gdje nije potreban kontinuirani prikaz podataka. Koristeći LoRu ograničenje na slanje signala je svakih 2-3 minute, sa limitom do 30 uplinkova dnevno, što treba uzeti u obzir ako projekt zahtjeva još češće prikupljanje podataka. Kod programa koji sam koristio je bio dosta trivijalan, jer sam ja fizički kontrolirao kada će uređaj raditi i slati podatke, tako što bi upalio powerbank. U slučaju kada bi htjeli ostaviti uređaj sa senzorom negdje, trebali bi napraviti preinake u kodu gdje uređaj u određenim intervalima šalje očitanje podataka senzora i kada je uređaj gotov sa slanjem podataka, da uređaj uđe u stanje mirovanja. To se lako da napraviti, pošto Arduino nudi klasu LowPower sa funkcijom DeepSleep() sa kojom možemo zadati koliko dugo će uređaj mirovati i na taj način štediti bateriju, a da se uređaj ne ugasi. Korištenje memorijske kartice sa MKRWAN 1300 je isto moguće i pružilo bi veću pouzdanost prikupljanja podataka. U principu jako sam zadovoljan korištenjem MKRWAN 1300 mikrokontrolera za LoRa komunikaciju, zato što sam uz relativno jednostavan postav uspio poslati signal preko 10km udaljenosti od prijamnika. Jedino što je bitno za slanje podataka na velike udaljenosti je dobar 'Line of Sight'. Kada sam bio negdje duboko u šumi u podnožju planine ili u parku Maksimir nisam nikako mogao poslati signal radi lošeg 'Line of Sight'-a. Još je zanimljivo za uočiti kako je moguća komunikacija između pošiljatelja i primatelja iako je SNR bio negativan većinu vremena. Što se tiče SHT21 senzora, smatram da senzor jako dobro radi i stvarno odgovara industrijskim standardima. Jedino treba paziti da pri mjerenjima senzor bude malo dulje vremena izložen vanjskim uvjetima, tako da odstupanja u temperaturi ne budu toliko velika, iako to ne bi trebao biti problem, ako je senzor postavljen na lokaciji mjerenja 24/7. Ograničenje senzora je to što može mjeriti samo temperaturu i relativnu vlagu zraka,ako postoji još potreba za mjerenjem tlaka zraka, Arduino nudi dobru alternativu u obliku MKR ENV shielda (može mjeriti temperaturu zraka, relativnu vlažnost, tlak i ambijentalnu svjetlost).

7 Zaključak

U ovom radu sam prošao kroz osnove o Arduino uređajima, pokazao kako mikrokontroleri komuniciraju sa računalom i koriste senzore. Pošto je Arduino korišten u ovom radu bio namjenjen za upotrebu kao 'Thing', pokazao sam važnost 'Internet of Things' u današnje doba i koliko raste upotreba istih.Objasnio sam kako se modulira LoRaPHY signal koristeći 'Frequency Shift Chirp Modulation' i neke osnovne značajke tog signala. Koristeći LoRa tehnologiju za komunikaciju sam pokazao koliko daleko se može poslati LoRaPHY signal u urbanom okruženju.

Na kraju zaključujem da ova tehnologija uz odgovarajuće predznanje pruža jednostavno realiziranje bilo kakvog IoT projekta koji se bavi prikupljanjem podataka senzora i upravljanjem aktuatora.

Dodaci

```
#include "thingProperties.h"
#include<SHT21.h>
#include<Wire.h>


SHT21 sht;

void setup(){

  //Zapo ni I2C protokol za komunikaciju sa senzorom
  Wire.begin();

  //Radim delay od 1.5 sekunde da Serial monitor ne blokira pokretanje
    programa
  delay(1500);

  //Definirano u thingProperites.h
  initProperties();
  //Spajanje na Arduino Cloud
  ArduinoCloud.begin(ArduinoIOTPreferredConnection, false);


  setDebugMessageLevel(0);
  ArduinoCloud.printDebugInfo();
}

void loop(){
  // alje  update o cloud varijablama u Arduino Cloud
  ArduinoCloud.update();

  //temp i humidity su cloud varijable i tip podataka im je float
  temp=sht.getTemperature();
  humidity= sht.getHumidity();
}
```

Listing 1: Kod za MKRWAN 1300 spojen sa SHT21 senzorom u Arduino Cloud environment-u

Bibliography

[1] https://en.m.wikipedia.org/wiki/File:Arduino_Logo.svg

[2] MKRWAN 1300 pinout shema, https://community.element14.com/products/arduino/w/documents/8511/arduino-mkr-wan-1300-pinout-samd21-pin-mapping-tech-specs-schematics-eagle-fil

[3] https://www.circuito.io/blog/arduino-uno-pinout/

[4] Internet of Things (IoT), https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT

[5] https://newhavendisplay.com/blog/serial-vs-parallel-communication/

[6] https://docs.arduino.cc/learn/communication/wire/

[7] ://www.geeksforgeeks.org/universal-asynchronous-receiver-transmitter-uart-prot

[8] https://www.tibco.com/glossary/what-is-the-internet-of-things-iot

[9] What is Iot? https://www.oracle.com/internet-of-things/what-is-iot/

[10] https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/#:~:text=The%20number%20of%20Internet%20of,around%208%20billion%20consumer%20devices.

[11] https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/

[12] https://blog.ttulka.com/lora-spreading-factor-explained/

[13] https://en.wikipedia.org/wiki/Chirp

[14] ”IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs): Amendment 1: Add Alternate PHYs,” in IEEE Std 802.15.4a-2007 (Amendment to IEEE Std 802.15.4-2006) , vol., no., pp.1-210, 31 Aug. 2007, doi: 10.1109/IEEESTD.2007.4299496. keywords: Wireless personal area networks;IEEE 802.15 Standard;Ad hoc

networks;Wireless communication;Media Access Protocol;ad hoc network;low data rate;low power;LR-WPAN;mobility;PAN;personal area network;radio frequency;RF;short range;wireless;wireless personal area network;WPAN,

[15] L. Vangelista, ”Frequency Shift Chirp Modulation: The LoRa Modulation,” in IEEE Signal Processing Letters, vol. 24, no. 12, pp. 1818-1821, Dec. 2017, doi: 10.1109/LSP.2017.2762960. keywords: Frequency shift keying;Chirp modulation;Power capacitors;Demodulation;AWGN channels;Chirp modulation;frequency shift chirp modulation (FSCM);Internet of Things (IoT);LoRa;low power wide area networks (LPWAN),

[16] `https://wirelesspi.com/understanding-lora-phy-long-range-physical-layer/#mjx-eqn-equation-basic-complex-sinusoid`

[17] prof.dr.sc. Amir Hamzić, Niskotemperaturna fizika i supravodljivost: Zagreb, 2005.