

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI PROJEKT

**Integracija razvojnog sustava  
ESP32-C3-DevKitM-1 s računalnim  
oblakom platforme AWS u IoT  
sustavima**

*Jelena Gavran*

*Voditelj: prof. dr. sc. Hrvoje Džapo*

Zagreb, siječanj 2024.

# SADRŽAJ

<b>Popis slika</b>	<b>iii</b>
<b>1. Uvod</b>	<b>1</b>
<b>2. Razvojni sustav ESP32-C3-DevKitM-1</b>	<b>3</b>
2.1. Wi-Fi . . . . .	4
<b>3. Amazon Web Services (AWS)</b>	<b>8</b>
3.1. Usluge AWS-a za IoT sustave . . . . .	8
3.1.1. AWS IoT Core . . . . .	9
3.1.2. Ostale dostupne IoT usluge u sustavu AWS . . . . .	12
<b>4. Integracija računalnog oblaka i razvojnog sustava</b>	<b>16</b>
4.1. Probno povezivanje korištenjem demo aplikacije AWS <i>Quick Connect</i>	16
4.2. Biblioteka <i>esp-aws-iot</i> . . . . .	17
4.3. Programski paket <i>iot-reference-esp32c3</i> . . . . .	19
4.4. Buduća razmatranja . . . . .	22
<b>5. Zaključak</b>	<b>23</b>
<b>6. Literatura</b>	<b>24</b>

# POPIS SLIKA

2.1.	Konfiguracija razvojnog sustava ESP32-C3-DevKitM-1 [13] . . . . .	3
2.2.	Blok dijagram modula ESP32-C3 [11] . . . . .	4
2.3.	Wi-Fi RF standardi [12] . . . . .	6
2.4.	Primjer scenarija Wi-Fi povezivanja u načinu rada stanice [13] . . . . .	7
2.5.	Primjer scenarija Wi-Fi povezivanja u načinu rada pristupne točke [13] . . . . .	7
3.1.	Arhitektura usluga AWS-a za IoT [6] . . . . .	9
3.2.	Princip rada usluge AWS IoT Core [6] . . . . .	10
3.3.	Komponente usluge AWS IoT Core [6] . . . . .	10
4.1.	Ispis u konzoli pri pokretanju demo aplikacije Quick Connect . . . . .	17
4.2.	Ispis slanja paketa protokolom MQTT s razvojnog sustava ESP32-C3 . . . . .	18
4.3.	Ispis poruke u AWS brokeru na temu <i>testClient/example/topic</i> . . . . .	19
4.4.	Ispis u konzoli s modula ESP32-C3 . . . . .	21
4.5.	Poruka na AWS brokeru . . . . .	21
4.6.	Ispis u konzoli s modula ESP32-C3 spojenog s DHT11 . . . . .	21

# 1. Uvod

Internet stvari (engl. Internet of things - IoT) je brzo rastuća tehnologija koja urbani svijet pretvara u potpuno mrežno povezani sustav visoke tehnologije. Uz sve veću popularnost, potražnju i rastuće zahtjeve na IoT tehnologiju, sustavi generiraju sve veću količinu podataka koju je gotovo nemoguće obraditi lokalno. Tom je izazovu doskočila tehnologija računarstva u oblaku (engl. *cloud computing*), koja pruža softver, alate i infrastrukturu preko interneta umjesto lokalnog upravljanja. Računanje u oblaku može pružiti skalabilnost i dostupnost potrebnu za pohranu i obradu velike količine podataka koju stvaraju IoT sustavi. Osim toga, računanje u oblaku može pomoći u smanjenju troškova upravljanja IoT sustavima.

Za razvoj IoT uređaja potrebni su mikrokontroleri s mogućnošću bežičnog povezivanja, primarno Wi-Fi komunikacije. Serija ESP32 mikrokontrolera tvrtke *Espressif* ozbiljan je konkurent među bežičnim uređajima zbog niske potrošnje, visoke otpornosti na temperature, te najvažnije, jednostavnom bežičnom povezivosti [5]. Jedan takav čip je ESP32-C3, koji pruža Wi-Fi i Bluetooth povezivanje. Čip je integriran u nekoliko različitih modula, koji su pak dio razvojnih sustava koje proizvodi *Espressif*. Za izradu ovog rada odabran je razvojni sustav ESP32-C3-DevKitM-1. Isto tako, usluga računarstva u oblaku mora biti jednostavna, intuitivna, pouzdana te najvažnije, skalabilna. Platforma AWS tvrtke *Amazon* ističe se kao jedan od najkorištenijih sustava za računarstvo u oblaku.

Ovaj projekt analizira mogućnosti koje pruža ESP32-C3-DevKitM-1 u razvoju aplikacija koje koriste Wi-Fi te kako povezati modul s oblakom platforme AWS. Opisano je i sam modul te mogućnosti koje nudi u sklopu Wi-Fi povezivanja. Dan je osnovni pregled funkcionalnosti koje nudi AWS te su nabrojane njegove usluge za olakšano stvaranje IoT sustava. Isto tako, opisan je postupak povezivanja modula i oblaka koji nudi AWS. Također, uz pomoć perifernih uređaja stvoren je pokazni IoT sustav koji integrira modul i računalni oblak.

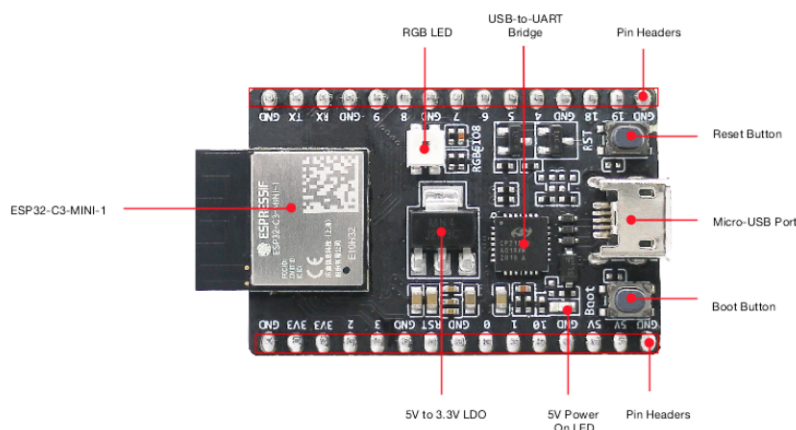
Rad je podijeljen u cjeline kako slijedi. Drugo poglavlje „*Razvojni sustav ESP32-C3-DevKitM-1*“ opisuje osnovne karakteristike korištenog razvojnog sustava kao ci-

ljane hardverske platforme te su opisane najvažnije značajke W-Fi tehnologije. U trećem poglavlju „*Amazon Web Services (AWS)*“ dan je pregled ekosustava AWS u kontekstu IoT sustava te su navedene usluge koje AWS podržava za olakšanu integraciju fizičkih uređaja s oblakom. Četvrto poglavlje „*Integracija računalnog oblaka i razvojnog sustava*“ opisuje postupke povezivanja razvojnog sustava s oblakom AWS usluge te je modeliran manji IoT sustav koji koristi opisane tehnologije uz dodatnu periferiju radi simulacije stvarnog sustava. Za kraj su opisana razmatranja o mogućim proširenjima ovakvog postava.

## 2. Razvojni sustav

### ESP32-C3-DevKitM-1

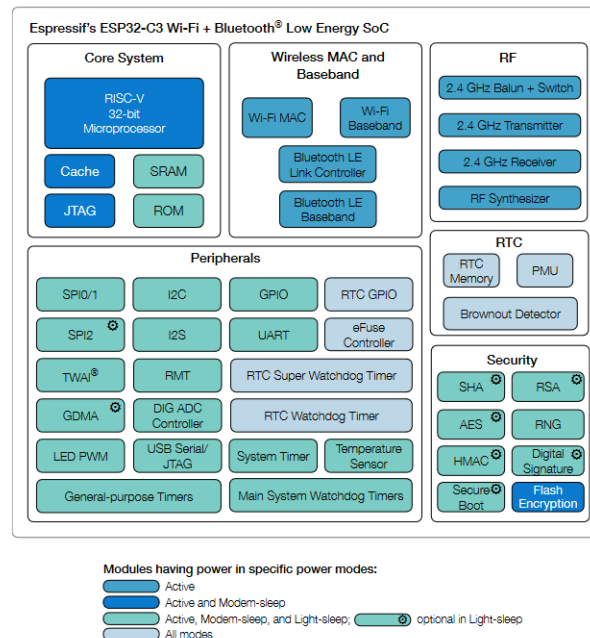
Razvojni sustav temelji se na modulu ESP32-C3-MINI-1. Modul je jedan u nizu ESP32-C3 serije SoC (engl. *System on Chip*) platformi tvrtke *Espressif*, te sadrži jednojezgri 32-bitni procesor s RISC-V arhitekturom koji radi na frekvenciji do 160 MHz. Modul sadrži 400 KB memorije tipa SRAM (engl. *Static random-access memory*), od kojih je 16 KB rezervirano za priručnu memoriju (engl. *cache*), 384 MB memorije tipa ROM (engl. *Read-only memory*) te 4 MB memorije tipa *Flash*. Od periferije sadrži 22 programabilna GPIO pina (engl. *General Purpose Input Output*), te digitalna sučelja SPI, UART, I2C i I2S. Također sadrži upravljače za sučelja USB i JTAG koji se mogu koristiti za efikasnije otklanjanje pogrešaka u kodu (engl. *debugging*). Konfiguracija sustava prikazana je na slici 2.1. [11]



**Slika 2.1:** Konfiguracija razvojnog sustava ESP32-C3-DevKitM-1 [13]

Budući da modul ima funkciju RF (engl. *radio frequency*) primopredajnika, podržava bežično lokalno umrežavanje odnosno Wi-Fi, koji omogućava propusnost do 20 Mbps protokolom TCP te maksimalnu propusnost od 30 Mbps koristeći protokol UDP. Isto tako, podržava protokol Bluetooth s podrškom za velike udaljenosti.

Modul ESP32-C3-MINI-1 bežični je uređaj niske potrošnje energije (engl. *ultra-low-power*) primarno namijenjen razvoju aplikacija koje koriste Wi-Fi ili *Bluetooth Low Energy* (BLE) protokol. Na slici 2.2 nalazi se blok shema modula sa svim dostupnim značajkama.



**Slika 2.2:** Blok dijagram modula ESP32-C3 [11]

## 2.1. Wi-Fi

IEEE 802.11, skupina standarda za bežične lokalne mreže (engl. *WLANs*) [17], nudi nekoliko različitih načina bežične modulacije signala. Pojedini standardi označeni su slovima abecede. Za korisničke mreže postoje dva frekvencijska pojasa: 2,4 GHz i 5 GHz.

Prednosti pojasa od 2,4 GHz su veći doseg, bolje prolaženje kroz fizičke prepreke te bolja podrška jer više bežičnih uređaja koristi pojas od 2,4 GHz nego od 5 GHz. S druge strane, ovaj pojas ima manju propusnost i nudi manje kanala koji se ne preklapaju. Isto tako, može doći do zagušenja mreže jer kućni i Bluetooth uređaji koriste ovaj isti mrežni pojas.

Pojas od 5 GHz nudi brži protok, manje zagušenih kanala te ima više kanala koji se međusobno ne preklapaju. Ipak, ima kraći raspon u usporedbi s mrežama od 2,4 GHz jer teže prolazi kroz prepreke. [14]

U nastavku su opisani ključni standardi Wi-Fi tehnologije [9]:

- 802.11b - najsporiji i najjeftiniji standard, emitira u frekvencijskom pojasu od 2,4 GHz. Može prenijeti do 11 Mbps te koristi komplementarno šifriranje (engl. *complementary code keying* - *CCK*) radi poboljšanja brzine prijenosa.
- 802.11a - transmitira u pojasu od 5 GHz i može prenijeti do 54 Mbps. Koristi ortogonalno frekvencijsko multipleksiranje (engl. *orthogonal frequency-division multiplexing* - *OFDM*), što je efikasnija tehnika u odnosu na CCK koja dijeli radio signal u nekoliko podsignala prije slanja primatelju. Ova metoda značajno umanjuje interferenciju.
- 802.11g - poput standarda 802.11b, koristi frekvencijski pojas od 2,4 GHz. Međutim, može prenijeti do 54 Mbps jer koristi tehniku OFDM.
- 802.11n - kompatibilan je standard sa prethodno opisanim standardima. Nudi znatno poboljšanje u rasponu i brzini u odnosu na svoje prethodnike. Ovaj standard može prenijeti do četiri toka podataka, svaki maksimalno 150 Mbps, no većina usmjerivača (engl. *router*) dopušta dva ili tri toka.
- 802.11ac - radi isključivo u pojasu od 5 GHz, te je kompatibilan s prethodnim standardima. Manje je sklon interferenciji i brži je od prethodnih standarda s maksimalnim prijenosom od 450 Mbps jednim tokom.
- 802.11ax - najnoviji standard koji proširuje nekoliko ključnih mogućnosti svojih prethodnika. Usmjerivači koji podržavaju ovaj standard dopuštaju tok podataka do 9.2 Gbps, što je značajan porast u usporedbi s prethodnicima. Isto tako, moguće je postaviti više antena na jedan usmjerivač, čime je omogućen prihvrat više veza odjednom bez usporavanja i interferencije.

Podsustav modula ESP32-C3 za Wi-Fi u skladu je sa standardom IEEE 802.111 te koristi nelicencirani pojas frekvencija od 2,4 GHz. U tom pojasu podržava propusnost od 20 i 40 MHz. Modul također podržava tehniku raznolikosti antena (engl. *antenna diversity*) za poboljšanje prijema i pouzdanosti signala korištenjem RF komutatora (engl. *switch*). Tim komutatorom upravljaju GPIO priključci i koristi se za odabir najbolje antene u kontekstu pouzdanosti i kvalitete signala. [12]

ESP32-C3 u potpunosti implementira protokol Wi-Fi na temelju standarda 802.11 b/g/n. Podržava osnovni skup (engl. *Basic Service Set* - *BSS*) operacija za značajke pristupne točke (engl. *SoftAP*). Upravljanje napajanjem odvija se automatski s minimalnom intervencijom domaćina kako bi se smanjila aktivnost uređaja.

Tvrtka *Espressif* također nudi biblioteke za povezivanje putem protokola TCP i IP te korištenje Wi-Fi *mesh* tehnologije. Pruža i podršku za protokole TLS 1.0, 1.1 i 1.2. Na slici 2.3 prikazani su Wi-Fi RF standardi koje koristi modul.



Name		Description
Center frequency range of operating channel <sup>1</sup>		2412 ~ 2484 MHz
Wi-Fi wireless standard		IEEE 802.11b/g/n
Data rate	20 MHz	11b: 1, 2, 5.5 and 11 Mbps 11g: 6, 9, 12, 18, 24, 36, 48, 54 Mbps 11n: MCS0-7, 72.2 Mbps (Max)
	40 MHz	11n: MCS0-7, 150 Mbps (Max)
Antenna type		PCB antenna and external antenna connector

**Slika 2.3:** Wi-Fi RF standardi [12]

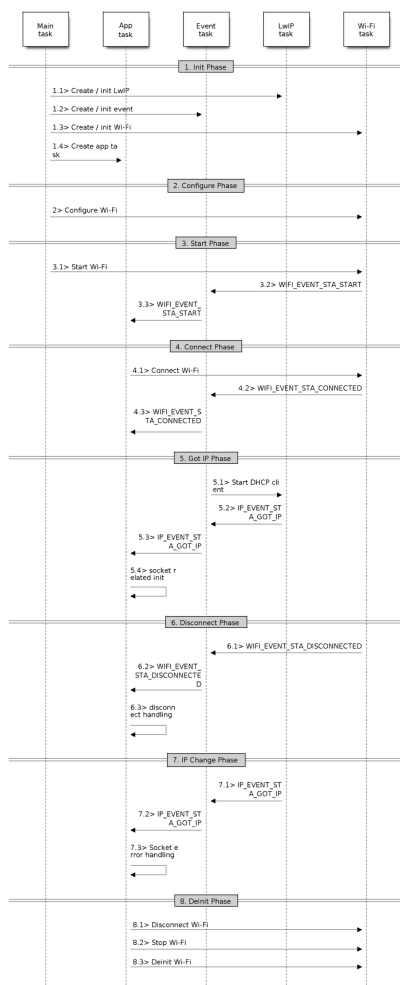
ESP32 nudi nekoliko načina rada pri korištenju Wi-Fi tehnologije [18]:

1. način rada stanice (engl. *station mode*) - ESP32 spaja se na točku pristupa,
2. način rada pristupne točke (engl. *SoftAP mode*) - druge se stanice spajaju na ESP32,
3. miješani - ESP32 radi kao stanica i pristupna točka spojena na drugu pristupnu točku.

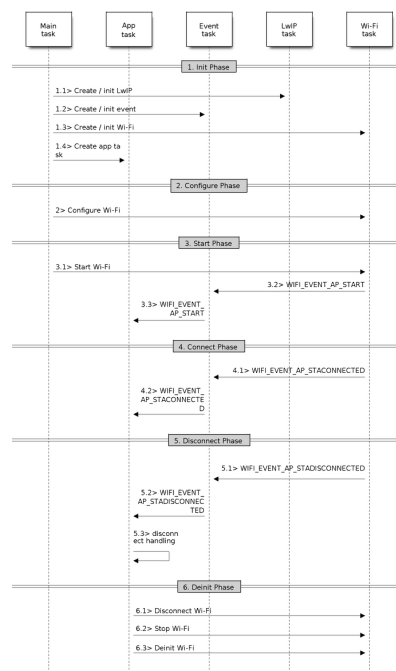
U nastavku su opisani scenariji Wi-Fi povezivanja modula ESP32-C3 u načinu rada stanice i pristupne točke.

Na slici 2.4 prikazan je sekvencijski dijagram zadataka koje ESP32 obavlja u cijelom ciklusu spajanja i komunikacije s pristupnom točkom. Iz slike je vidljivo da se ciklus sastoji od osam faza. Prva faza služi za inicijalizaciju upravljačkih programa i pokretanje zadataka odnosno dretvi koje će obavljati zadatke vezane uz svoju dužnost. Glavni zadatak pokreće četiri različite dretve izvršavanja: aplikacijski zadatak, zadatak za događaje, zadatak za IP protokol, te zadatak za Wi-Fi. U drugoj fazi konfigurira se upravljački program za Wi-Fi. U sljedećoj se fazi pokreće upravljački program, nakon koje slijedi faza pretraživanja mreže i povezivanja na usmjerivač ili pristupnu točku. Nakon inicijalizacije DHCP klijenta, započinje faza dohvata IP adrese. Šesta faza odvija se nakon prekida Wi-Fi veze, čime se također uklanjaju i sve UDP i TCP konekcije. U aplikaciji se može omogućiti radno čekanje na ponovno uspostavljanje veze. Sedma faza pokreće se pri detekciji promjene IP adrese. Posljednja faza služi za programsko odspajanje s mreže i zaustavljanje upravljačkog programa za Wi-Fi.

Slika 2.5 modelira slučaj u kojem ESP32 ima ulogu pristupne točke. Scenarij je vrlo sličan ranije opisanom slijedu događaja, no razlikuje se u dvije faze i događajima koji su pohranjeni u sustavu. Ovaj način rada nema fazu detekcije promjene IP adrese, jer je u ovom načinu ESP32 upravo taj uređaj čija se IP adresa može promijeniti. Isto tako, ne postoji faza dohvata IP adrese.



**Slika 2.4:** Primjer scenarija Wi-Fi povezivanja u načinu rada stanice [13]



**Slika 2.5:** Primjer scenarija Wi-Fi povezivanja u načinu rada pristupne točke [13]

U modulu ESP32 stavljen je veliki naglasak na mehanizme uštede energije, što se također preslikava na korištenje Wi-Fi veze. Modul pruža načine uštede energije i pri radu kao stanica i pristupna točka, no neke značajke nisu podržane u pristupnoj točki. Modul pri neaktivnosti može otići u stanje mirovanja (engl. *sleep mode*). Postoje dva načina uštede energije u načinu rada stanice: minimalna i maksimalna ušteda. Pri minimalnoj uštedi stanica se budi iz stanja mirovanja nakon svakog DTIM intervala (engl. *Delivery Traffic Indication Message*). Ovim se načinom ne gube globalno emitirane poruke (engl. *broadcast*) jer se one prenose nakon DTIM intervala. Međutim, ova metoda ne štedi puno energije ako je pristupna točka na koju je spojen modul postavila malen interval. Pri maksimalnoj uštedi moguće je znatno produžiti vrijeme mirovanja u odnosu na DTIM interval, no ovime se riskira gubitak globalno emitiranih poruka.

### 3. Amazon Web Services (AWS)

Amazon usluge za web (engl. *Amazon Web Services* - AWS) sveobuhvatna je platforma za računarstvo u oblaku koju pruža Amazon te sadrži brojne usluge u oblaku, uključujući infrastrukturu (engl. *Infrastructure as a Service* - IaaS), platformu (engl. *Platform as a Service* - PaaS) i softver (engl. *Software as a Service* - SaaS). AWS usluge nude organizacijske alate kao što su računalna snaga, baza podataka i usluge isporuke sadržaja [16].

AWS je podijeljen u više različitih usluga koje se mogu pojedinačno konfigurirati na temelju korisnikovih potreba. Neke od usluga koje nudi AWS su: pohrana, baze podataka, migracija, povezivanje, monitoriranje, sigurnost, analitika, umjetna inteligencija te razvoj mobilnih aplikacija.

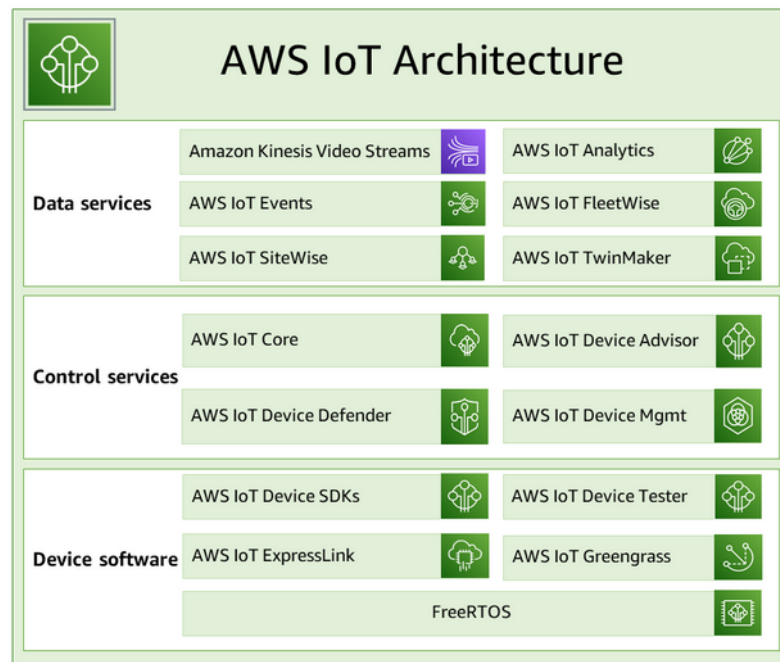
AWS pruža usluge iz mnogo podatkovnih centara (engl. *data center* - DC) koji su raspodijeljeni po zonama dostupnosti (engl. *availability zone* - AZ) diljem regija cijelog svijeta. Jedna regija obuhvaća nekoliko fizički bliskih zona povezanih mrežom niske latencije.

Također, nude se brojne mogućnosti za razvojne inženjere u sklopu AWS-a. Nudi alate naredbenog retka (engl. *command-line tools*) i pakete za razvoj programa (engl. *Software Development Kit* - SDK) za puštanje aplikacija u produkciju (engl. *deployment*) i upravljanje vlastitim uslugama i aplikacijama. Paketi za razvoj programa dostupni su u raznim programskim jezicima, uključujući programske jezike C++, Android, iOS, Java, Node.js, Python i Ruby.

#### 3.1. Usluge AWS-a za IoT sustave

AWS isto tako nudi brojne usluge za razvoj IoT sustava. Usluga AWS-a za IoT pruža platformu za upravljanje IoT uređajima te obradu podataka i njihovu pohranu na druge AWS usluge, poput baze podataka. AWS IoT pruža usluge u oblaku koje povezuju IoT uređaje s drugim uređajima i uslugama AWS-a u oblaku. Također pruža software za uređaje, poput paketa za razvoj programa, za jednostavniju integraciju s uslugama

AWS-a za IoT. Na slici 3.1 nalazi se prikaz arhitekture usluga koje AWS nudi za razvoj IoT sustava.



**Slika 3.1:** Arhitektura usluga AWS-a za IoT [6]

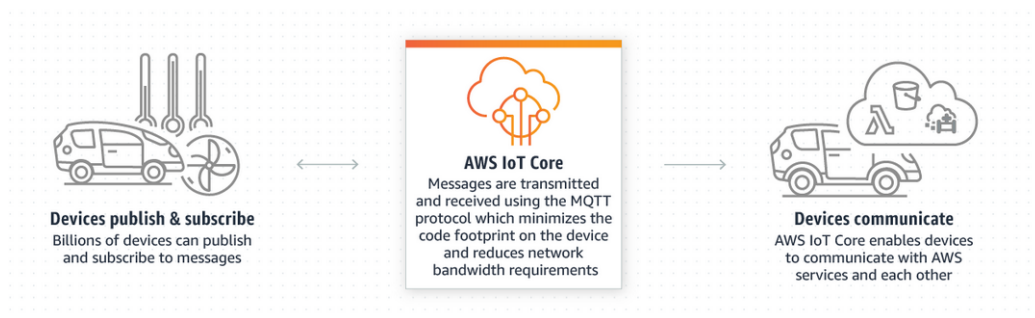
AWS IoT podržava sljedeće komunikacijske protokole:

- MQTT (engl. *Message Queuing Telemetry Transport*)
- HTTPS
- LoRaWAN (engl. *Long Range Wide Area Network*)
- TLS

U nastavku su opisane usluge za razvoj IoT sustava u sklopu AWS-a.

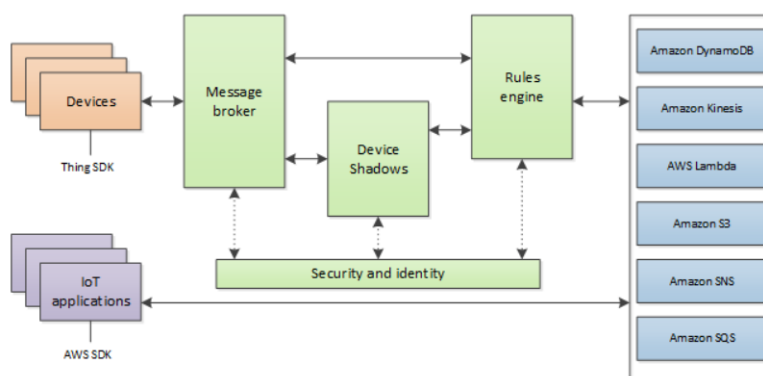
### 3.1.1. AWS IoT Core

AWS IoT Core ključna je komponenta za integraciju oblaka i fizičkih uređaja. Omogućava povezivanje uređaja i preusmjeravanje poruka na usluge AWS-a. Koristi MQTT koji je standardni protokol za razmjenu poruka u IoT sustavima. To je lagan (engl. *lightweight*) protokol za prijenos poruka temeljen na objavi/pretplati sustavu, te je pogodan za povezivanje udaljenih uređaja uz minimalnu potrošnju [15]. AWS IoT Core pruža paletu značajki za razmjenu poruka temeljenih na protokolu MQTT, koje pomažu pri izradi prilagodljive i skalabilne IoT arhitekture [6]. Na slici 3.2 nalazi se pregled rada usluge AWS IoT Core.



**Slika 3.2:** Princip rada usluge AWS IoT Core [6]

AWS IoT Core pruža usluge koje povezuju oblak AWS-a s IoT uređajima kako bi se ostale usluge u oblaku i aplikacije mogle međusobno komunicirati s tim uređajima. Na slici 3.3 nalaze se svi segmenti usluge AWS IoT Core te kako oni komuniciraju s vanjskim dijelovima. Zelenom bojom označena je sama usluga, narančastom bojom fizički uređaji, odnosno stvari (engl. *Things*), sivom bojom druge IoT aplikacije unutar AWS-a, dok su plavom bojom označene ostale usluge koje se nude u ekosustavu AWS.



**Slika 3.3:** Komponente usluge AWS IoT Core [6]

U nastavku su navedene ključne usluge koje pokriva AWS IoT Core.

### Usluge za slanje poruka

AWS IoT Core usluge za povezivanje pružaju sigurnu komunikaciju s IoT uređajima i upravlja porukama koje prolaze između uređaja i oblaka.

Prilazni uređaj omogućuje uređajima sigurnu i efikasnu komunikaciju sa sustavom AWS. Komunikacija je osigurana sigurnosnim protokolima koji koriste X.509 certifikate.

Broker za poruke pruža mehanizam uređajima i aplikacijama slanje i primanje po-

ruka. Moguće je koristiti protokol MQTT ili direktno WebSocket za objavu i pretplatu na teme. Uređaji i klijenti koriste HTTP REST sučelje za objavu poruka brokeru. Broker zatim distribuira podatke na uređaje koji su se pretplatili na određene teme kao i na druge AWS aplikacije te usluge koje prate teme na brokeru.

AWS IoT Core za protokol LoRaWAN omogućava postavljanje privatne LoRaWAN mreže tako što poveže LoRaWAN te prilazne uređaje na AWS bez potrebe za razvojem mrežnog servera za LoRaWAN (engl. *LoRaWAN Network Server* - LNS). Poruke primljene od LoRaWAN uređaja šalju se na stroj za pravila (engl. *rules engine*) gdje se formatiraju i prosljeđuju ostalim AWS uslugama.

Stroj za pravila povezuje podatke iz brokera s drugim AWS IoT uslugama za pohranu i dodatnu obradu. Primjerice, moguće je umetati ili pretraživati po podatkovnim tablicama ili pozvati određene definirane funkcije na temelju izraza definiranog u stroju. Isto tako, moguće je obraditi te podatke i proslijediti novostvoreni format poruka drugim uslugama ili bazama podataka.

## Upravljačke usluge

Upravljačke usluge AWS IoT Core komponente pružaju sigurnost uređaja te značajke za upravljanje i registraciju novih uređaja.

Moguće je definirati vlastite autorizatore radi upravljanja autentifikacijskim i autorizacijskim strategijama koristeći vlastiti servis za autentifikaciju i Lambda funkcije za računanje koje AWS nudi.

Usluga za postavljanje (engl. *provisioning*) uređaja omogućava konfiguriranje i prijavu uređaja u AWS koristeći predložak koji opisuje resurse potrebne uređaju: *stvar*, certifikat i nekoliko politika. *Stvar* je unos u registar koji sadrži attribute opisa uređaja. Uređaji koriste certifikate za autentifikaciju s AWS IoT sustavom. Politike određuju koje operacije uređaj može izvršiti.

Isto tako, moguće je definirati grupe za lakšu kategorizaciju i upravljanje uređajima. Grupe također mogu imati podgrupe, i tako graditi hijerarhiju grupa. Sve akcije izvršene na roditeljima propagiraju se do najdubljih podgrupa. Dozvole dodijeljene grupi primjenjuju se na sve uređaje u toj grupi i svim njihovim podgrupama.

Usluga za poslove (engl. *jobs*) omogućava definiranje udaljenih operacija koje se pošalju i izvrše na fizičkim uređajima spojenih u oblak. Posao se može odnositi preuzimanje i instalaciju aplikacija, ažuriranje sustava, ponovno pokretanje, obnova certifikata i slično.

Registar organizira resurse pridružene uređajima u oblaku. Moguće je registrirati

uređaje i pridružiti im maksimalno po tri atributa. Isto tako, uređajima se mogu pridružiti certifikati i klijentski ID za MQTT komunikaciju kako bi se olakšalo otklanjanje budućih grešaka.

Usluga za sigurnost i identitet pruža dijeljenu odgovornost za sigurnost unutar AWS oblaka. Fizički uređaji moraju držati vjerodajnice na sigurnom kako bi se podaci mogli slati brokeru na siguran način. Značajke za sigurnost koriste i stroj za pravila kao i broker za poruke radi sigurnog prijenosa podataka drugim uslugama unutar AWS ekosustava.

### **Usluge za uređaje**

AWS IoT Core osigurava pouzdano aplikacijsko iskustvo iako uređaji nisu uvijek povezani.

Sjena uređaja (engl. *Device Shadow*) dokument je u JSON formatu koji se koristi za pohranu i dohvat trenutnog stanja i informacija o uređaju. AWS nudi uslugu koja održava stanje uređaja (engl. *Device Shadow service*) kako bi aplikacije mogle komunicirati s uređajem bez obzira je li uređaj na mreži ili ne. Kada uređaj nije priključen na mrežu, usluga sjene uređaja upravlja podacima za povezane uređaje. Kada se uređaj ponovno spoji na mrežu, sinkronizira stanje sa sjenom uređaja koja se nalazi u oblaku. Uređaji također mogu objavi svoje stanje usluzi u bilo kojem trenutku kako bi bilo na raspolaganju drugim aplikacijama i uređajima.

### **3.1.2. Ostale dostupne IoT usluge u sustavu AWS**

Uz ranije opisanu glavnu komponentu IoT Core koju nudi AWS za stvaranje IoT aplikacija, u samom ekosustavu nalazi se još mogućnosti za jednostavniju integraciju oblaka i fizičkih uređaja. U nastavku su ukratko opisane ostale usluge koje se mogu integrirati uz jezgrenu uslugu AWS IoT Core.

Važno je napomenuti kako nisu sve usluge dostupne u svim regijama unutar AWS-a.

#### **IoT Analytics**

AWS IoT Analytics automatizira korake potrebne za analizu podataka prikupljenih od IoT uređajima. Filtrira, transformira i obogaćuje podatke prije nego ih pohrani u vremensku bazu podataka za daljnju analizu. Moguće je postaviti uslugu da prikuplja podatke s uređaja samo koji su potrebni, vrši matematičke operacije i dopunjava

podatke raznim metapodacima, primjerice o lokaciji. Zatim se podaci mogu analizirati koristeći ugrađeni sustav za pretraživanje koji koristi SQL sintaksu ili pak vršiti kompleksniju analizu koristeći usluge umjetne inteligencije. Isto tako, ova usluga nudi vizualizaciju podataka integracijom s dodatnom uslugom Amazon QuickSight.

### **IoT Device Defender**

AWS IoT Device Defender potpuna je usluga koja pomaže pri osiguranju IoT uređaja. Kontinuirano revidira IoT konfiguracije radi provjere jesu li sve u skladu s najboljim sigurnosnim praksama. Također pruža kontinuirano monitoriranje sigurnosnih metrika s uređaja i AWS IoT Core kako bi se detektirale anomalije u ponašanju pojedinih uređaja.

Ova usluga također omogućuje slanje alarma na konzolu AWS IoT sustava i na uslugu za monitoriranje Amazon CloudWatch. Koriste se ugrađene mitigacijske akcije kako bi se izolirali nesigurni uređaji.

### **IoT Events**

AWS IoT Events usluga služi za praćenje događaja u sustavu. Ova usluga prati ulazne podatke s više IoT uređaja i aplikacija radi prepoznavanja uzoraka i pokretanja prikladnih operacija na određene događaje. Moguće je pratiti ne samo fizičke uređaje, nego i druge AWS aplikacije integrirane u IoT sustav.

### **IoT FleetWise**

AWS IoT FleetWise jest usluga koja se koristi za prikupljanje podataka od vozila i njihovu organizaciju u oblaku. Prikupljeni se podaci mogu koristiti za poboljšanje kvalitete, performansa i autonomije vozila. Također podržava više različitih protokola i podatkovnih formata. Ova usluga pomaže pri transformaciji *low-level* poruka u oblik čitljiv čovjeku i standardizira podatke radi lakše analize u oblaku. Moguće je također definirati vrstu podataka i trenutak u kojem se ti podaci šalju u oblak.

Kada su podaci o vozilu u oblaku, mogu se koristiti u aplikacijama koje analiziraju zdravlje vozila. Ove informacije mogu pomoći pri identifikaciji potencijalnih problema u održavanju i pri unapređenju naprednih tehnologija poput autonomne i asistirane vožnje integracijom strojnog učenja.



## **IoT Greengrass**

AWS IoT Greengrass jest usluga otvorenog koda (engl. *open source*) za računarstvo na rubu (engl. *edge computing*) i u oblaku koja pomaže pri izradi, objavi i upravljanju IoT aplikacija na uređajima. Može se koristiti za omogućavanje uređajima lokalno reagiranje na podatke koje generiraju, pokretanje modela strojnog učenja za predikciju, te filtriranje i agregaciju podataka s uređaja. Omogućava uređajima da prikupljaju i analiziraju podatke ne u oblaku, nego ili na samom uređaju ili drugom mjestu koje je bliže izvoru tih podataka. Također može komunicirati na siguran način s uslugom AWS IoT Core i izvoziti podatke u oblak. Karakteristika računarstva u rubu, koje omogućava ova komponenta, jest približavanje računanja izvorišnim uređajima, čime se poboljšava vrijeme odziva i štedi propusnost [7].

## **IoT Roborunner**

AWS IoT RoboRunner nova je usluga koja pruža infrastrukturu za optimizaciju robota iz jedne točke gledišta. Uz pomoć ove usluge moguće je izgraditi aplikacije za jednostavniji međusobni rad robota. Namijenjena je za industrijske robote i automatizirane sustave za olakšano upravljanje opremom. Pruža centralne repozitorije podataka za pohranu te podržava različite podatkovne formate od raznih robota i autonomnih sustava.

## **IoT TwinMaker**

AWS IoT TwinMaker usluga je za kreiranje operativnih digitalnih dvojnika fizičkih i digitalnih sustava. Stvara digitalne vizualizacije koristeći mjerenja i analize iz raznih senzora i kamera radi praćenja stvarnog stanja i uvjeta u kojima se objekt, zgrada ili kompleks nalazi. Podaci iz stvarnog svijeta se mogu koristiti za dijagnostiku i ispravljanje pogrešaka ili pak optimizaciju operacija.

Digitalni dvojnik (engl. *digital twin*) digitalna je reprezentacija sustava i svih njegovih fizičkih i digitalnih komponenti. Dinamički se ažurira primitkom novih podataka kako bi simulirao stvarno stanje i ponašanje sustava.

## **IoT SiteWise**

AWS IoT SiteWise jest usluga koja skalabilno prikuplja, modelira, analizira i vizualizira podatke iz industrijske opreme. Usluga pruža kreiranje web aplikacija za operativne korisnike radi prikaza i analize industrijskih podataka u stvarnom vremenu.

Moguće je dobiti uvide u podatke i operacije konfiguriranjem i praćenjem raznih metrika, primjerice efektivnost i efikasnost opreme. Ovu je uslugu moguće koristiti jedino uz ranije opisan IoT TwinMaker.

## 4. Integracija računalnog oblaka i razvojnog sustava

Ovo poglavlje opisuje biblioteke i primjere korištenje za povezivanje modula ESP32-C3 s računalnim oblakom AWS sustava. Za početno povezivanje korištena je demo aplikacija koja automatski povezuje razvojni sustav s *backend* dijelom AWS sustava. U idućim poglavljima korištene su razne biblioteke za povezivanje uređaja i oblaka.

### 4.1. Probno povezivanje korištenjem demo aplikacije *AWS Quick Connect*

Za početno probno povezivanje razvojnog sustava ESP32-C3 sa sustavom AWS, korištena je službena demo aplikacija *Quick Connect* projekta *FreeRTOS*, koji je trenutno u vlasništvu AWS-a. Ova aplikacija, nakon učitavanja binarne datoteke u razvojni sustav i definiranje vjerodajnica za povezivanje na Wi-Fi, putem protokola MQTT šalje informacije s uređaja u AWS [3]. Račun za uređaj se automatski stvori te nije potrebna nikakva dodatna prijava u sam sustav. Na slici 4.1 nalazi se ispis u konzoli nakon što se pokrene demo aplikacija na razvojnem sustavu. Iz slike je vidljivo postavljanje (engl. *provisioning*) uređaja, kao i kreiranje certifikata za siguran prijenos podataka.

Slika 4.2 također prikazuje ispis u konzoli, no ovaj puta sadržaj paketa koji se šalju protokolom MQTT. Podaci su prikazani u formatu JSON.

Iako je demo primjer uspješno slao podatke putem MQTT protokola, podaci se nisu prikazivali na stranici. Slika ?? prikazuje stranicu projekta *FreeRTOS* te identifikacijski broj uređaja koji se spojio, no sami podaci koje uređaj šalje nisu prikazani budući da nisu ni primljeni. Nije provedeno daljnje otklanjanje pogrešaka jer je ova aplikacija pružena *out-of-the-box*, te za detaljniju analizu problema potrebno je prolaziti kroz izvorni kod aplikacije. Stoga su odabrana druga rješenja za povezivanje razvojnog sustava s uslugama AWS-a.

```
C:\Users\Jelena\Desktop\QuickConnect_Esspressif-ESP32C3_windows.x64\Start_Quick_Connect.exe
19:05:43 - This program will configure your ESP32C3 to connect to the internet
          using Wi-Fi, and then to AWS. You will be asked to enter your Wi-Fi
          network credentials so that the device can connect. This information
          will be written to the device to allow it to connect to the internet.
          The information will not be transmitted to the internet or AWS.

19:05:43 - -----
19:05:43 - Enter Wifi Network (network SSID): Gavran
19:05:48 - Enter Wifi Security (1:Open, 2:WPA, 3:WPA2): 3
19:06:03 - Enter Wifi Password:

19:06:07 - -----
19:06:07 - Provisioning device to connect. This may take several minutes.

19:06:27 - Flashing complete.
19:06:30 - WiFi connection verified.
19:06:30 - Device has started generating a private key and code-signing request.
19:07:43 - Device has finished generating a private key and code-signing
          request.
19:07:44 - Device has started the self-claiming process.
19:07:50 - Device has finished self-claiming a certificate.

19:07:50 - Provisioning complete. Retrieving thing certificate and name from the
          device.

19:07:50 - Got device certificate.
19:07:51 - Got thing name.
```

Slika 4.1: Ispis u konzoli pri pokretanju demo aplikacije Quick Connect

## 4.2. Biblioteka *esp-aws-iot*

Kao drugi primjer korištena je biblioteka *esp-aws-iot* [1], koja također sadrži gotove primjere za povezivanje sa sustavom AWS. U nastavku su navedene glavne značajke koje biblioteka podržava:

1. *Fleet Provisioning* - usluga koju nudi AWS, a odnosi se na konfiguriranje uređaja u oblaku bez unaprijed definiranih certifikata,
2. *Thing Shadow* - ranije opisane sjene stvarnih uređaja,
3. poslovi (engl. *jobs*) - akcije koje se mogu izvršiti na samom uređaju ili u oblaku,
4. OTA (engl. *Over The Air*) - podržava ažuriranje postavki i koda uređaja putem mreže, podržani protokoli HTTP i MQTT,
5. HTTP povezivanje,
6. MQTT povezivanje.

OTA ažuriranje uređaja vrlo je moćna značajka koja ne zahtijeva ponovni unos koda na uređaj, nego se mrežnim putem ažuriraju željene značajke i tako izmijeni izvorni kod. Biblioteka i AWS primarno podržavaju protokol MQTT za OTA prijenos, no eksperimentalno je omogućen i protokol HTTP, upravo jer nema razinu sigurnosti koja je pružena protokolom MQTT.

```
C:\Users\Jelena\Desktop\QuickConnect_Espressif-ESP32C3_windows.x64\Start_Quick_Connect.exe
19:11:27 - I (299438) QuickConnectMain: Establishing an MQTT
connection...
19:11:27 - I (299838) QuickConnectMain: MQTT CONNECTED!
19:11:27 - I (299848) QuickConnectNetworking: MQTT publish succeeded.
19:11:27 - Sent: [{"label" : "Temperature","display_type" :
"line_graph","values" : [{"unit" : "Celsius","value" :
35.650600,"label" : ""}]]
19:11:28 - I (300848) QuickConnectNetworking: MQTT publish succeeded.
19:11:28 - Sent: [{"label" : "Temperature","display_type" :
"line_graph","values" : [{"unit" : "Celsius","value" :
36.089199,"label" : ""}]]
19:11:29 - I (301858) QuickConnectNetworking: MQTT publish succeeded.
19:11:29 - Sent: [{"label" : "Temperature","display_type" :
"line_graph","values" : [{"unit" : "Celsius","value" :
37.843601,"label" : ""}]]
19:11:30 - I (302868) QuickConnectNetworking: MQTT publish succeeded.
19:11:30 - Sent: [{"label" : "Temperature","display_type" :
"line_graph","values" : [{"unit" : "Celsius","value" :
35.650600,"label" : ""}]]
```

**Slika 4.2:** Ispis slanja paketa protokolom MQTT s razvojnog sustava ESP32-C3

Ova biblioteka koristi mogućnosti koje pruža usluga AWS IoT Core za povezivanje s fizičkim uređajima. Prije pokretanja primjera na samom uređaju, potrebno je napraviti AWS račun za korisnika, te mu dodijeliti potrebna prava i politike. Isto tako, kako bi se uređaj spojio na AWS koristeći protokol TLS za sigurno povezivanje, potrebno je generirati i sigurnosne certifikate. Kako bi uređaj komunicirao s oblakom, potrebno ga je registrirati u sustavu AWS. Pri stvaranju odnosno registraciji pojedine *Stvari*, generiraju se i certifikati koji je zatim potrebno učitati u memoriju uređaja. Ovaj korak nije nužan ako se uređaj spaja putem usluge *Fleet Provisioning*.

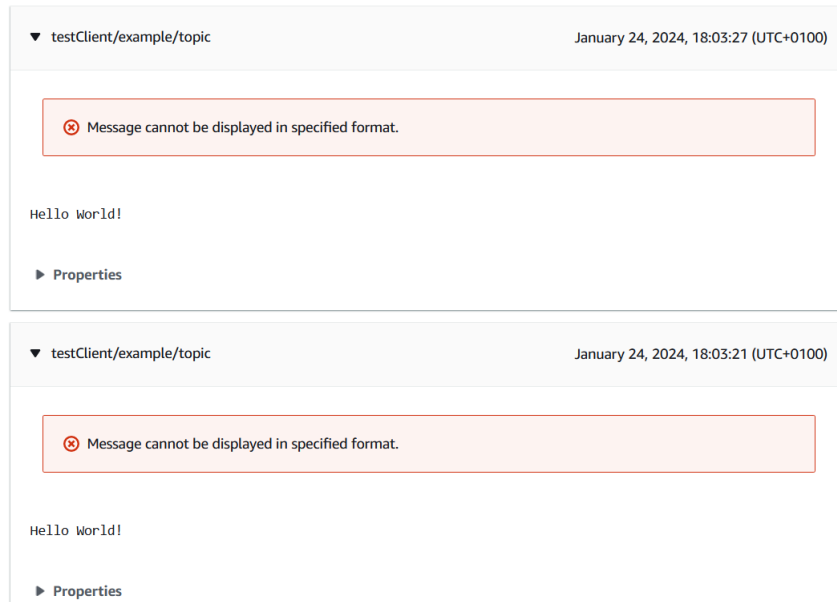
Generirana sigurnosna politika daje dozvolu za sljedeće radnje:

- povezivanje - dopušta uređaju povezivanje na brokera u oblaku s bilo kojim klijentskim identifikatorom,
- objava poruka - uređaj može objavljivati poruke protokolom MQTT na bilo koju temu,
- pretplata na teme - moguća je pretplata na bilo koju temu u brokeru,
- prihvata poruka - uređaj može primati poruke od brokera s bilo koje teme.

Detaljniji koraci stvaranja računa, prava i potrebnih certifikata mogu se pronaći ovdje [4].

Ova je biblioteka korištena za povezivanje uređaja na AWS broker. Korišten je testni broker dostupan u sklopu usluge AWS IoT Core. Povezivanje na internet u ovom primjeru omogućeno je programatski, te je potrebno unaprijed definirati vjerodajnice

Wi-Fi veze. Uređaj je, povežavši se na internet i broker pomoću ranije generiranih certifikata, kontinuirano slao testnu poruku na broker. Prikaz poruka koje je broker primio na testnu temu nalazi se na slici 4.3. Budući da nije postavljen način parsiranja poruka, javlja se greška, iako su poruke uredno prikazane.



Slika 4.3: Ispis poruke u AWS brokeru na temu *testClient/example/topic*

### 4.3. Programski paket *iot-reference-esp32c3*

Ovaj programski paket [2] proširenje je prethodno opisane biblioteke, te integrira više njezinih značajki za potpun prikaz mogućnosti koje biblioteka nudi.

Paket podržava tri demo simulacije implementirane kao tri FreeRTOS zadatke, od kojih svaka koristi istu MQTT vezu. Simulacije koriste biblioteku *coreMQTT*, dok se biblioteka *coreMQTT-Agent* koristi radi osiguranja pouzdane međudretvenosti unutar MQTT konekcije. Simulacije su sljedeće:

- *ota\_over\_mqtt\_demo*: praćenje OTA ažuriranja putem protokola MQTT,
- *sub\_pub\_unsub\_demo*: Kružna pretplata i odjava s tema u brokeru također koristeći protokol MQTT,
- *temp\_sub\_pub\_and\_led\_control\_demo*: Objava poruka i kontroliranje LED lampice na uređaju.

Simulacija *ota\_over\_mqtt\_demo* koristi OTA uslugu koju nudi AWS u sklopu FreeRTOS projekta za konfiguriranje i kreiranje OTA ažuriranja. OTA klijentski softver

na razvojnom sustavu koristi AWS IoT OTA biblioteku i u pozadini je vrtil unutar FreeRTOS zadatka. Simulacija se pretplati i sluša na temi zaduženoj za OTA ažuriranja. Po primitku obavijesti o nadolazećem ažuriranju, preuzima novu verziju i potvrđuje potpis preuzetog ažuriranja certifikatom. Po uspješnoj verifikaciji, uređaj se resetira i aktivira se nova verzija.

Zadaci unutar simulacije *sub\_pub\_unsub\_demo* pretplaćuju se na temu u oblaku AWS IoT Core, šalju konstantan string na istu temu na koju su pretplaćeni, te se zatim odjave s teme, te tako u krug.

Simulacija *temp\_sub\_pub\_and\_led\_control\_demo* kreira zadatak koji se pretplati na temu unutar oblaka AWS IoT Core. Zatim se čita temperatura sa senzora za temperaturu integriranog u modul te se objavljuje u formatu JSON na istu temu na koju je i pretplaćen. Također, omogućava korisniku primanje paketa za paljenje ili gašenje LED lampice.

Sve tri demo zadaće mogu se paralelno izvršavati kao odvojeni zadaci.

Glavna značajka koju koristi ovaj paket jest da se povezivanje na Wi-Fi mrežu odvija pomoću BLE tehnologije. Wi-Fi mreža konfigurira se putem sigurnosnih sesija uspostavljenih Bluetooth kanalom, i tako uklanja potrebu za upisivanjem vjerodajnica direktno u programski kod. Razvojni sustav pri pokretanju generira QR kod koji je potrebno skenirati mobilnom aplikacijom koju nudi tvrtka *Espressif* [10] kako bi se uređaj povezao na mrežu. Uređaj, tako jednom spojen na mrežu, nema potrebe ponovno se spajati pri svakom uključivanju i isključivanju jer se vjerodajnice zapisuju u memoriju. Jedino brisanjem particije NVS (engl. *Non-Volatile Storage*) memorije tipa *Flash* moguće je spojiti se na novu Wi-Fi mrežu.

Na slici 4.4 nalazi se ispis u konzoli s modula kada se pokrenu sve tri simulacije odjednom. Crvenom su bojom istaknuti zadaci koje logiraju informativne poruke, dok je narančastom bojom istaknut zadatak koji šalje podatkovni paket u JSON formatu protokolom MQTT. Budući da razvojni sustav nema integrirani temperaturni senzor, vrijednost koja bi trebala biti očitana postavlja se na nulu. Na slici 4.5 prikazan je taj podatak (ali u drugoj iteraciji slanja) u brokeru unutar sustava AWS.

Radi manje pokazne simulacije stvarnog IoT sustava koji bi koristio opisane integrirane tehnologije, u sustav je priključen modul DHT11 koji vraća digitalni izlaz proporcionalan temperaturi i vlazi. Za čitanje podataka sa senzora korištena je službena biblioteka tvrtke *Espressif* u okvir radnog okvira ESP-IDF koja nudi upravljački program za korištenje modula DHT11. Uklonjena je simulacija za OTA ažuriranje, te su ostavljene druge dvije vezane za MQTT brokera. Izmijenjen je dio programskog koda koji koristi integrirani temperaturni senzor kako bi se koristio novospojeni modul.

```

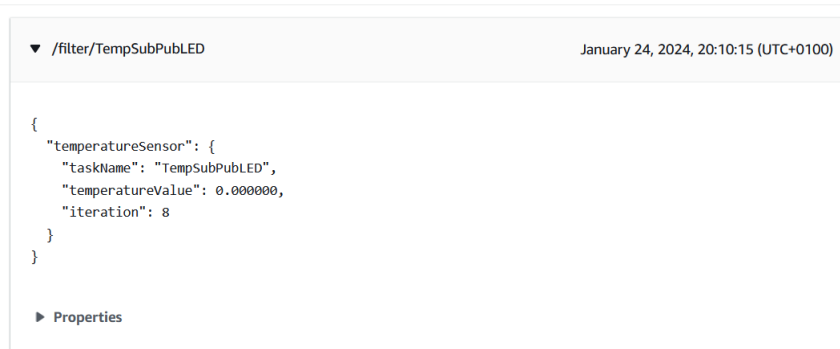
I (66049) sub_pub_unsub_demo: Task "SubPub0" sending publish request to coreMQTT-Agent with message "SubPub0" on topic "/filter/subPub0" with ID 2.
I (66069) sub_pub_unsub_demo: Task "SubPub0" waiting for publish 2 to complete.
I (66259) coreMQTT: Ack packet deserialized with result: MQTTSuccess.
I (66259) coreMQTT: State record updated. New state=MQTTPublishDone.
I (66259) coreMQTT: De-serialized incoming PUBLISH packet: DeserializerResult=MQTTSuccess.
I (66269) coreMQTT: State record updated. New state=MQTTPubAckSend.
I (66279) temp_sub_pub_and_led_control_demo: Received incoming publish message {"temperatureSensor":{"taskName": "TempSubPubLED", "temperatureValue": 0.000000, "iteration": 2}}
I (66289) temp_sub_pub_and_led_control_demo: Rx'ed ack for QoS1 publish from Tx to /filter/TempSubPubLED (P3:F0).
I (66829) ota_over_mqtt_demo: Received: 0   Queued: 0   Processed: 0   Dropped: 0
I (67289) coreMQTT: Publishing message to /filter/subPub0.

I (67829) ota_over_mqtt_demo: Received: 0   Queued: 0   Processed: 0   Dropped: 0
I (68379) coreMQTT: Ack packet deserialized with result: MQTTSuccess.
I (68379) coreMQTT: State record updated. New state=MQTTPublishDone.
I (68379) coreMQTT: De-serialized incoming PUBLISH packet: DeserializerResult=MQTTSuccess.
I (68389) coreMQTT: State record updated. New state=MQTTPubAckSend.
W (68399) sub_pub_unsub_demo: Error or timed out waiting for ack for publish message 2. Re-attempting publish.
I (68409) sub_pub_unsub_demo: Task "SubPub0" sending publish request to coreMQTT-Agent with message "SubPub0" on topic "/filter/subPub0" with ID 2.
I (68419) sub_pub_unsub_demo: Task "SubPub0" waiting for publish 2 to complete.
I (68829) ota_over_mqtt_demo: Received: 0   Queued: 0   Processed: 0   Dropped: 0
I (69399) coreMQTT: Publishing message to /filter/subPub0.

I (69829) ota_over_mqtt_demo: Received: 0   Queued: 0   Processed: 0   Dropped: 0
I (69919) temp_sub_pub_and_led_control_demo: Sending publish request to agent with message {"temperatureSensor":{"taskName": "TempSubPubLED", "temperatureValue": 0.000000, "iteration": 3}} on topic "/filter/TempSubPubLED"
I (69929) coreMQTT: Publishing message to /filter/TempSubPubLED.

```

**Slika 4.4:** Ispis u konzoli s modula ESP32-C3



```

▼ /filter/TempSubPubLED January 24, 2024, 20:10:15 (UTC+0100)

{
  "temperatureSensor": {
    "taskName": "TempSubPubLED",
    "temperatureValue": 0.000000,
    "iteration": 8
  }
}

► Properties

```

**Slika 4.5:** Poruka na AWS brokeru

Slika 4.6 tako prikazuje dio ispisa konzole, te ne postoji oznaka za OTA ažuriranje budući da taj posao nije pokrenut. Isto tako, vidljivo je da paket koji se šalje ima temperaturu postavljenu na ne-nul vrijednost, nego stvarnu temperaturu prostora u kojem se modul nalazi.

```

I (5880410) coreMQTT: Publishing message to /filter/TempSubPubLED.
I (5880420) temp_sub_pub_and_led_control_demo: Task TempSubPubLED waiting for publish 1032 to complete.
I (5881420) coreMQTT: Ack packet deserialized with result: MQTTSuccess.
I (5881420) coreMQTT: State record updated. New state=MQTTPublishDone.
I (5881420) coreMQTT: De-serialized incoming PUBLISH packet: DeserializerResult=MQTTSuccess.
I (5881430) coreMQTT: State record updated. New state=MQTTPubAckSend.
I (5881440) temp_sub_pub_and_led_control_demo: Received incoming publish message {"temperatureSensor":{"taskName": "TempSubPubLED", "temperatureValue": 21.000000, "iteration": 1032}}
I (5881460) temp_sub_pub_and_led_control_demo: Rx'ed ack for QoS1 publish from Tx to /filter/TempSubPubLED (P1033:F0).

```

**Slika 4.6:** Ispis u konzoli s modula ESP32-C3 spojenog s DHT11



## 4.4. Buduća razmatranja

Ranije opisane biblioteke i programski paketi nude paletu mogućnosti rada s razvojnim sustavom ESP32-C3 te mnoštvo značajki za integraciju sa uslugom AWS. Jedna od značajki koju je potrebno uzeti u obzir prilikom izrade budućeg stvarnog IoT sustava jest povezivanje uređaja u Wi-Fi mrežu. Kao što je opisano, biblioteke nude gotova rješenja za taj problem, omogućujući spajanje putem QR koda. Prvotni primjer, *Quick Connect*, također dinamički unosi vjerodajnice u uređaj tako što se ručno upisuju kada se sustav pokreće. Idealan sustav bi imao zlatnu sredinu ova dva sustava - pri svakom pokretanju sustava da nije potrebno unositi Wi-Fi vjerodajnice, no isto tako da nije potrebna dodatna mobilna aplikacija za prijavu uređaja na mrežu.

Nadalje, u korištenoj biblioteci te programskom kodu postoji mogućnost korištenja biblioteke za binarnu serijalizaciju podataka koji se šalju putem protokola MQTT. Format CBOR (engl. *Concise Binary Object Representation*), za razliku od formata JSON, pretvara podatke u binarni oblik te ih tako šalje mrežom, što rezultira manjom latencijom i nosivošću (engl. *payload*) [8]. Pri prijenosu velike količine podataka pri ograničenim resursima, poput u IoT sustava, ušteda na veličini poslanih podataka znatno utječe na efikasnost sustava. Isto tako, binaran je format prilagodljiv u odnosu na JSON, gdje mora postojati unaprijed definirana shema za primitak podataka. Iduće iteracije sustava trebale bi razmotriti korištenje binarne serijalizacije radi poboljšanih performansi.

## 5. Zaključak

Detaljno je opisan postupak integracije razvojnog sustava ESP32-C3-DevKitM-1 s računalnim oblakom platforme AWS. Također su opisana tehnička svojstva razvojnog sustava te je detaljnije opisana Wi-Fi tehnologija. Dan je pregled ključnih funkcionalnosti koje pruža oblak u sustavu AWS za IoT sustave, te su ukratko opisane dodatne značajke koje se mogu koristiti u skalabilnim IoT sustavima. Opisana su programska rješenja za spajanje modula i usluge AWS koje je potrebno implementirati u razvojnom sustavu. Također, navedena su moguća poboljšanja postojećih rješenja za buduće integracije.

Integracijom opisanih tehnologija dobiva se kompletan IoT uređaj koji se može koristiti u svakodnevnom životu. Korištene demo implementacije mogu se skalirati na više uređaja unutar platforme AWS. Kombinacija ESP32-C3-DevKitM-1 razvojnog sustava, Wi-Fi veze i integracije s platformom AWS pruža snažno i fleksibilno rješenje za razvoj IoT aplikacija. Ovaj integrirani sustav omogućuje prikupljanje, pohranu i upravljanje podacima iz različitih senzora ili uređaja te njihovo daljnje korištenje unutar platforme AWS, gdje se pak razne usluge računarstva u oblaku mogu koristiti za obradu podataka. Zahvaljujući ovakvoj konfiguraciji, korisnici mogu stvoriti napredne IoT sustave s visokom razinom kontrole, praćenja i automatizacije.

## 6. Literatura

- [1] esp-aws-iot. URL <https://github.com/espressif/esp-aws-iot/blob/release/202012.04-LTS/examples>.
- [2] iot-reference-esp32-c3. URL <https://github.com/FreeRTOS/iot-reference-esp32c3>.
- [3] Aws quick connect application. URL <https://github.com/espressif/aws-quickconnect>.
- [4] Aws setup. URL <https://github.com/FreeRTOS/iot-reference-esp32c3/blob/main/AWSSetup.md>.
- [5] 2021. URL <https://www.intuz.com/guide-on-top-iot-development-boards>.
- [6] *AWS Documentation*. Amazon.com, Inc., 2024. URL <https://docs.aws.amazon.com/index.html>.
- [7] Stephen J. Bigelow. What is edge computing? everything you need to know. 2021. URL <https://www.techtarget.com/searchdatacenter/definition/edge-computing>.
- [8] C. Bormann. *CBOR*, 2020. URL <https://cbor.io/>.
- [9] Marshall Brain i Talon Homer. How wifi works. 2021. URL <https://computer.howstuffworks.com/wireless-network.htm>.
- [10] Espressif. Esp ble provisioning app. URL [https://play.google.com/store/apps/details?id=com.espressif.provble&hl=en\\_US](https://play.google.com/store/apps/details?id=com.espressif.provble&hl=en_US).
- [11] *ESP32-C3 Series Datasheet*. Espressif Systems, 2023. URL [https://www.espressif.com/sites/default/files/documentation/esp32-c3\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-c3_datasheet_en.pdf).

- [12] *ESP32-C3-Mini 1 Datasheet*. Espressif Systems, 2023. URL [https://www.espressif.com/sites/default/files/documentation/esp32-c3-mini-1\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-c3-mini-1_datasheet_en.pdf).
- [13] *ESP-IDF Programming Guide*. Espressif Systems, 2023. URL <https://docs.espressif.com/projects/esp-idf/en/v5.0.2/esp32c3/index.html>.
- [14] Microsoft. Wi-fi problems in your home. URL <https://support.microsoft.com/hr-hr/windows/problemi-s-wi-fijem-i-raspored-va%C5%A1lega-doma-e1ed42e7-a3c5-d1be-2abb-e8fad00ad32a>.
- [15] *MQTT*. MQTT.org, 2022.
- [16] A. S. Gillis N. Barney. Amazon web services (aws). URL <https://www.techtarget.com/searchaws/definition/Amazon-Web-Services>.
- [17] *IEEE 802.11 Wireless Local Area Networks*. The Working Group for WLAN Standards, 2023. URL <https://www.ieee802.org/11/>.
- [18] Random Nerd Tutorials. Esp32 useful wi-fi library functions. 2021. URL <https://randomnerdtutorials.com/esp32-useful-wi-fi-functions-arduino/>.

## **Integracija razvojnog sustava ESP32-C3-DevKitM-1 s računalnim oblakom platforme AWS u IoT sustavima**

### **Sažetak**

U ovom radu opisan je postupak integracije razvojnog sustava ESP32-C3-DevKitM-1 s računalnim oblakom platforme AWS. Opisan je sam modul i mogućnosti koje nudi u sklopu Wi-Fi povezivanja. Dan je pregled glavnih funkcionalnosti koje pruža oblak u sustavu AWS za IoT sustave, te su ukratko opisane dodatne dostupne značajke za IoT sustave. Detaljno su opisana moguća programska rješenja za spajanje modula i usluge AWS. Navedena su buduća razmatranja koja bi poboljšala opisane integracije.

**Ključne riječi:** IoT, ESP32-C3-DevKitM-1, AWS, MQTT, računarstvo u oblaku