



# **Operativni sistemi i Edge platforme za IoT**

Teodora Kocić, 1457



# Sadržaj

IoT OS arhitektura i komponente

Aplikacije razvijene nad IoT OS

Literatura

# IoT OS arhitektura i komponente

Uloga IoT operativnih sistema je da obezbede funkcionalnosti koje su neophodne u efektivnom razvoju IoT rešenja. Operativni sistem kontroliše hardver i softver uređaja. Arhitektura IoT operativnih sistema biće detaljnije opisana kroz primere dva predstavnika IoT OS – ZephyrOS i MbedOS.

# ZephyrOS i MbedOS

## ZephyrOS

Sveobuhvatan, lightweight, kernel i pomoćni servisi

Prenosiv (portabilan) i siguran

Omogućava konekciju (BLE, Wi-Fi, Ethernet, USB, kao i IoT protokoli)

Jednostavan za korišćenje (evidentiranje, praćenje, debug, shell, podrška za Windows/Linux/macOS)

## MbedOS

Podržava niz uređaja i komponenti

Izvršenje softvera u realnom vremenu (RTOS)

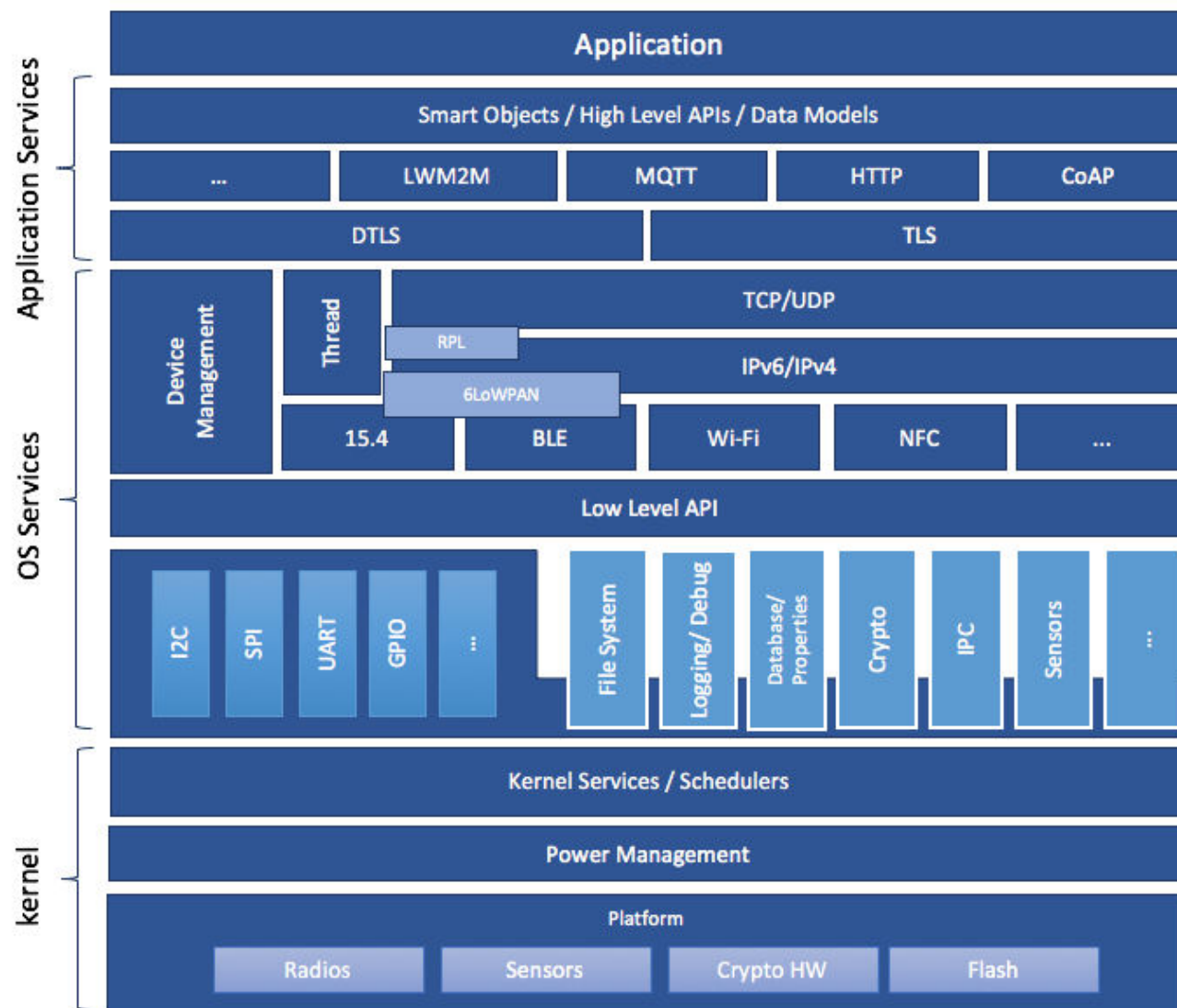
Kod je otvorenog tipa (open source)

Jednostavan za korišćenje

Bezbedna komunikacija i obezbeđuje sigurnost hardvera

Veliki broj dostupnih drajvera i biblioteka

# Arhitektura ZephyrOS-a



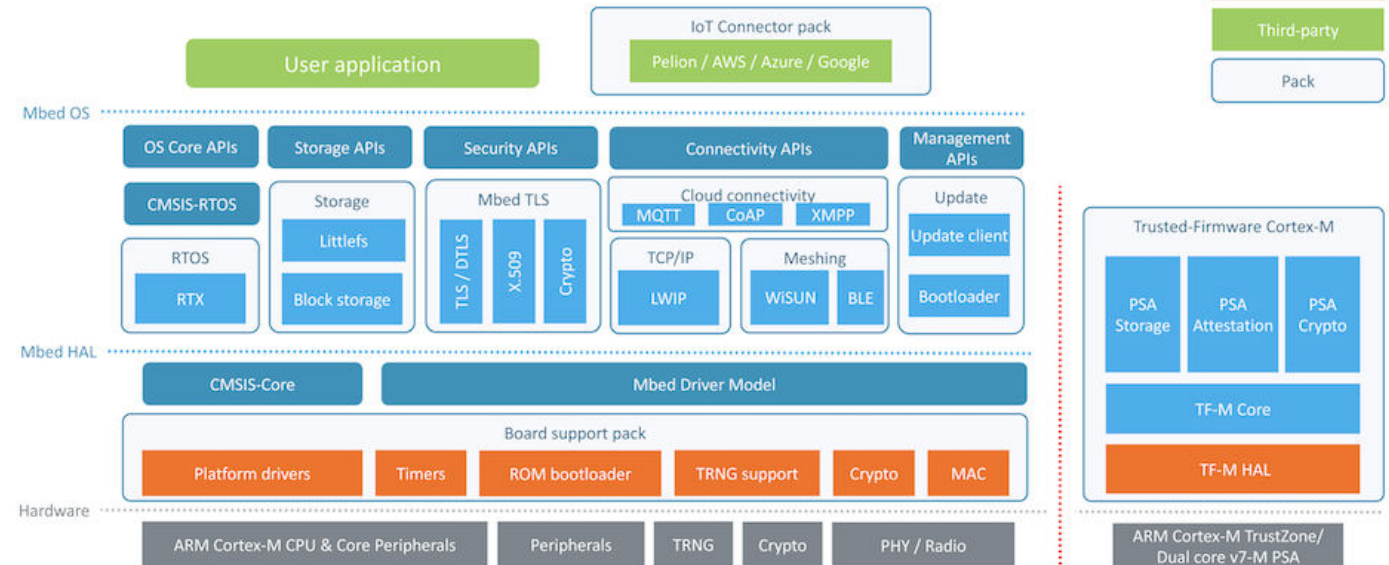
Arhitektura Zephyr operativnog sistema podeljena je u dva dela:

- OS deo – jezgro (kernel) i servisi operativnog sistema
- Deo koji je namenjen korišćenju od strane korisnika – aplikativni servisi

# Arhitektura MbedOS-a

## Mbed OS 6 Conceptual Architectural

Componentized, Layered Architecture



Mbed OS koristi sloj hardverske apstrakcije (HAL) kao podršku delovima mikrokontrolera koji se često susreću (tajmeri). Ova osnova olakšava pisanje aplikacija na osnovu zajedničkog API-a.

HAL je ulazna tačka prilikom dodele novih feature-a ili target-a.

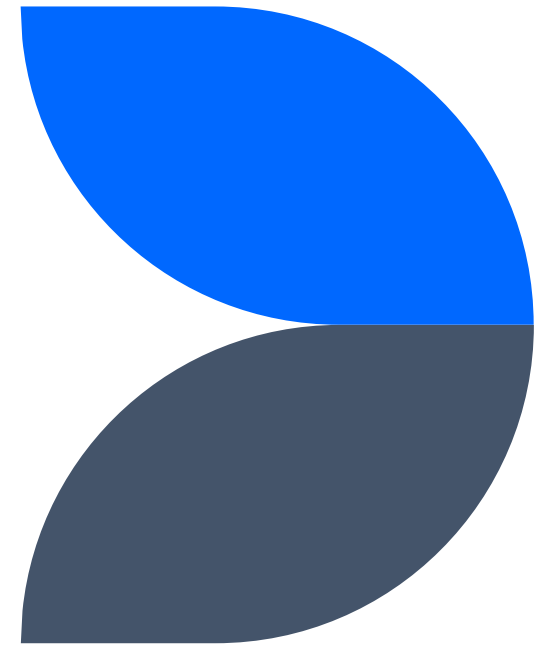
Struktura MbedOS-a obezbeđuje usklađivanje aplikacija i sistema za skladištenje podataka.

MbedOS implementira i sloj za odabir novih target-a i integraciju procesa pokretanja svakog podržanog lanca alata.



# Aplikacije

Razvijene aplikacije u IoT  
operativnim sistemima



# ZephyrOS aplikacija

Build sistem ZephyrOS-a zasnovan je na CMake-u. Kada se pokrene build aplikacije vrši se kontrolisanje konfiguracije i kreiranje same te aplikacije, kao i build Zephyr-a, kompajlirajući ih u jednu binarnu datoteku. Aplikacija mora da ima sledeći sadržaj:

/app |—— CMakeLists.txt |—— prj.conf |—— src |—— main.c

# Komponente aplikacije

## CMakeList.txt

Datoteka koja specificira build sistemu gde da pronađe ostale aplikacione fajlove. Ovde se nalaze konfiguracioni fajlovi koji su vezani za određenu komponentu, kao i dodaci koji omogućavaju izvršenje i debugovanje kompajliranih binarnih fajlova na pravim ili virtuelnim hardverima.

## Konfiguracioni fajlovi jezgra

Definiše vrednosti koje su specifične za samu aplikaciju (uglavnom je ovaj fajl imenovan **prj.conf**). Ove vrednosti se kombinuju kasnije sa specifikacijama za sami board sa kojim se radi kako bi se dobila konfiguracija jezgra.

## Izvršni kod aplikacije

Najčešće postoji jedan (može se napisati i veći broj) aplikacioni fajl, napisan u C-u ili nekom asemblerskom jeziku. Ovaj fajl (fajlovi) smešten je u pod-folderu **src**.

# Izvršni kod aplikacije

- Aplikacija prikuplja podatke sa senzora za temperaturu, pritisak i relativnu vlažnost vazduha, kao i podatke o vrednostima x, y i z komponenti: linearnog ubrzanja sa dva akcelerometra (LSM6DSL i LSM303AGR), rotacionog ubrzanja izmerenih na žiroskopu (LSM6DSL) i vrednosti magnetnog polja očitanih sa magnetometra (LSM303AGR).
- Ukoliko se svi spoljni parametri (temperatura, pritisak i relativna vlažnost vazduha) nalaze u definisanim granicama štampaju se u konzoli numeričke vrednosti linearnog (ukoliko ubrzanje odgovara gravitacionoj konstanti između Španije i Norveške, gledano po vrednosti latituda), rotacionog ubrzanja, kao i vrednost magnetnog polja očitane sa senzora.

# Pokretanje aplikacije i njeno izvršenje - ZephyrOS

- Da bi se aplikacija kreirala na željenom board-u (u demo aplikaciji koristi se ploča kompatibilna sa Arduino Nano 33 BLE Sens - [nRF52DK](#)) treba izvršiti komandu:

**west build -b nrf52dk app/src**

- Nakon pozicioniranja u folder **zephyr** (nakon povlačenja čitavog projekta **zephyrproject** u okviru ovog foldera nalaze se pod-folderi **zephyr**, **bootloader**, **modules** i **tools**) potrebno je izvršiti komandu:

**west flash**

# Rezultat izvršenja aplikacije na FIT IoT Lab-u

## Experiment sensors\_zephyr #363727

User **kocic**

Submitted **2023-04-02 20:21:50**

Started **2023-04-02 20:22:01**








Duration **0 minutes (0%)** of 5 minutes

Nodes **1**

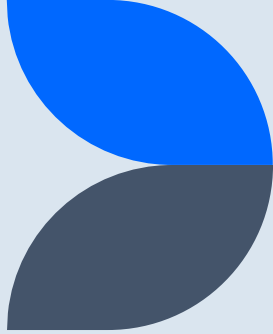
State **Running**

**Stop** **Download**

Actions on selected nodes ▾

Nodes	UID	Firmware	Monitoring	Deployment	Actions
nrf52dk-10.saclay.iot-lab.info	e6d3	zephyr.elf		Success	      
<pre>HTS221: Temperature: 25.2 C HTS221: Relative Humidity: 39.0% LPS22HB: Pressure:100.645 kpa LPS22HB: Temperature: 25.5 C LSM6DSL: Gyro (dps): x: 0.010, y: -0.036, z: -0.033 -&gt; rotational acceleration = 0.063 7:: lsm6dsl trig 2527 LSM303AGR: Accel (m.s-2): x: -0.1, y: -0.4, z: 9.5 -&gt; linear acceleration = 9.553 LSM303AGR: Magn (gauss): x: -0.315, y: 0.162, z: -0.341 -&gt; magnetic field = 0.491 3 Sensor dashboard HTS221: Temperature: 25.4 C HTS221: Relative Humidity: 39.0% LPS22HB: Pressure:100.645 kpa LPS22HB: Temperature: 25.5 C LSM6DSL: Gyro (dps): x: 0.011, y: -0.034, z: -0.031 -&gt; rotational acceleration = 0.062 8:: lsm6dsl trig 2946 LSM303AGR: Accel (m.s-2): x: -0.1, y: -0.5, z: 9.6 -&gt; linear acceleration = 9.632 LSM303AGR: Magn (gauss): x: -0.321, y: 0.168, z: -0.332 -&gt; magnetic field = 0.491</pre>					

# MbedOS aplikacija



Kod Mbed operativnog sistema aplikacija sadrži konfiguracioni fajl, komponente (kod demo aplikacije to su biblioteke vezane za sve senzore koji se koriste), može da sadrži i dodatne biblioteke koje su neophodne kako bi izvršni fajl mogao da se kreira i nakon toga pokrene. Pored konfiguracionih fajlova i biblioteka, aplikacija sadrži jedan **json** fajl u kojem su sadržani opisi komandi za kompajliranje i **cpp** fajl koji sadrži izvršni kod aplikacije.

# Okruženja za pokretanje izvršnog fajla

Kod aplikacije se može izvršavati:

- a) Lokalno – ukoliko postoji instalacija Mbed Studio-a ili korišćenjem Mbed CLI 1
- b) U cloud-u (Keil Studio Cloud) - bira se projekat koji će se izvršavati i bira se board za koji se dati kod treba da kompajlira (kod demo aplikacije radi se o [Nordic nRF52-DK](#) ploči, kompatibilna sa Arduino Nano 33 BLE Sens). Nakon kompajliranja dobija se izvršna datoteka koja se može nakon povezivanja hardvera pokrenuti i njeno izvršenje se može debugovati i pratiti na taj način korak po korak. Pokretanje koda je moguće i korišćenjem virtualnog uređaja.



# Izvršni kod aplikacije

U okviru ove aplikacije prate se vrednosti za x, y i z komponentu očitane sa senzora LSM303AGR i LSM6DSL. Prikazuju se orijentacije vektora linearnog i rotacionog ubrzanja očitanih sa akcelerometra i žiroskopa pomenutih senzora.

Ukoliko su podaci o orijentaciji dostupni dioda svetli, dok se istovremeno i podaci o vrednostima x, y i z komponente vektora linearnog i rotacionog ubrzanja, kao i ukupne vrednosti ovih fizičkih veličina očitane sa senzora se štampaju u konzoli.

```

LSM6DSL_Event_Status_t status;
acc_gyro->get_event_status(&status);
if (status.D6D0rientationStatus) {
    /* Send 6D Orientation */
    send_orientation();

    double acc_value, gyro_value;
    acc_value = sensor_data_value(acc_axes[0], acc_axes[1], acc_axes[2]);
    gyro_value = sensor_data_value(gyro_axes[0], gyro_axes[1], gyro_axes[2]);

    printf("Measured values of linear acceleration's components: x = %i, y = %i, z = %i -> acc = %f\n", acc_axes[0], acc_axes[1], acc_axes[2], acc_value);
    printf("Measured values of rotational acceleration's components: x = %i, y = %i, z = %i -> gyro = %f\n", gyro_axes[0], gyro_axes[1], gyro_axes[2], gyro_value);

    /* Led blinking. */
    myled = 1;
    wait(0.2);
    myled = 0;
}

```

```

/* Enable LSM6DSL accelerometer and gyroscope*/
acc_gyro->enable_x();
acc_gyro->enable_g();

int32_t *acc_axes, *gyro_axes;
acc_gyro->get_x_axes(acc_axes);
acc_gyro->get_g_axes(gyro_axes);
/* Enable 6D Orientation. */
acc_gyro->enable_6d_orientation();

```

# Kompajliranje MbedOS aplikacije i njeno izvršenje

- Nakon odabira ploče za koju će dati kod biti kompajliran, ukoliko nema grešaka u kodu pritiskom na prvu ikonicu u nizu započinje proces kompajliranja



- Ukoliko je uspešno završeno kompajliranje da bi se kod pokrenuo najpre se poveže virtualni ili fizički uređaj i ikonica za pokretanje (druga u nizu) postaje dostupna i klikom na nju ukoliko je kod ispravan počinje izvršenje istog

# Literatura

1. <https://www.zephyrproject.org/#>
2. <https://os.mbed.com/docs/mbed-os/v6.16/introduction/index.html>