

Portovanje Contiki-NG operativnog sistema na OpenMote-B uređaje za industrijske IoT primjene

Studentski rad

Jelena Banović, Sara Danilović

studenti drugog ciklusa studija

Univerzitet u Banjoj Luci, Elektrotehnički fakultet

Banja Luka, Bosna i Hercegovina

e-mail: jelena.banovic@student.etf.unibl.org,

sara.danilovic@student.etf.unibl.org

Sažetak— Contiki-NG je operativni sistem otvorenog koda za Internet of Things (IoT) uređaje sljedeće generacije. Fokusira se na sigurnu i pouzdanu komunikaciju uz malu potrošnju i primjenu standardnih protokola, kao što su IPv6/6LoWPAN, 6TiSCH, RPL i CoAP. OpenMote-B uređaji su otvorena razvojna hardverska platforma za izradu prototipa industrijskih IoT (IIoT) aplikacija. Osnova realizacije je Texas Instruments CC2538 ARM-Cortex-M3 mikrokontroler. Uređaji podržavaju istovremeni rad u ISM opsezima 2,4 GHz i 868/915 MHz sa potpunom podrškom za aktuelne IEEE 802.15.4 standarde. U ovom radu je dat pregled Contiki-NG OS i način implementacije u okviru OpenMote-B platforme. Opisana je praktična realizacija na primjeru kreiranja mreže sa RPL graničnim ruterom i primjenom CoAP aplikacionog protokola za očitavanje statusa senzora.

Ključne riječi – IoT, Contiki-NG, OpenMote-B, RPL, CoAP, IEEE 802.15.4, eksperimentalno testiranje

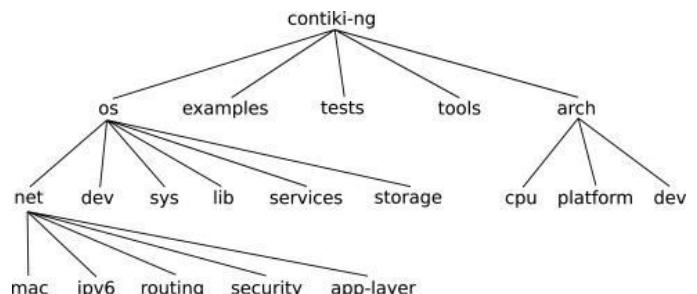
I. UVOD

Pojavom nove generacije senzorskih uređaja (32-bitni ARM Cortex-M uređaji) sa naprednim režimima male potrošnje, omogućeno je realizovanje širokog spektra IoT aplikacija. Brojna istraživanja u oblasti bežičnih senzorskih mreža rezultovala su razvojem IoT tehnologija, pri čemu je od posebnog značaja interoperabilnost rada i primjena standardizovanih rješenja.

Contiki-NG OS obezbeđuje podršku standardnim protokolima kao što su IEEE 802.15.4 TSCH, 6LoWPAN, 6TiSCH, RPL, CoAP, MQTT i LWM2M. Pored toga, cilj je i podrška za moderne hardverske platforme kao i pouzdanost i sigurnost na bazi savremenih rješenja, kontinuirane integracije korišćenjem simulacija i fizičkog testiranja, kao i tehnika testiranja sigurnosti i bezbjednosti prenosa. Contiki-NG definije slojeve hardverske apstrakcije (eng. hardware abstraction layer - HAL) za uobičajene interfejse, uključujući GPIO i SPI [1], [2].

Sl.1 ilustruje strukturu direktorijuma Contiki-NG kodne baze. Kod nezavisno od tipa platforme se može naći u direktorijumu „os“, a svi državari specifični za hardver mogu

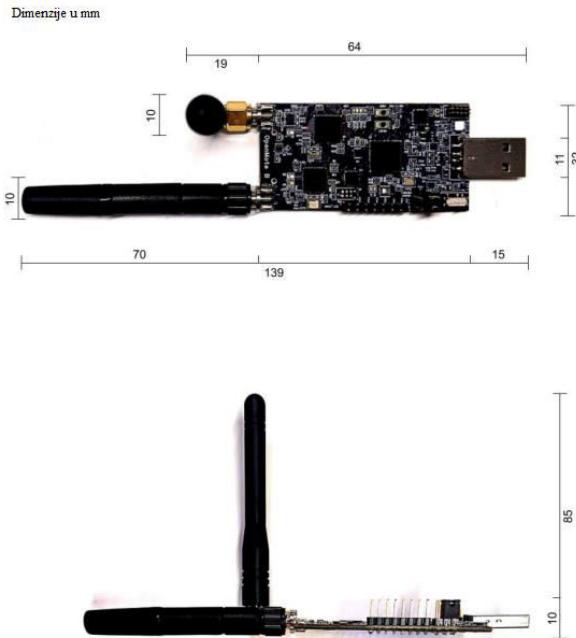
se naći u direktorijumu „arch“. U direktorijumu „tests“ se nalazi automatizovani paket za testiranje kontinuirane integracije, dok „tools“ sadrži pomoćne uslužne programe, kao što su skripte koje se koriste za otpremanje firmvera na podržane uređaje i za generisanje dokumentacije [1] - [4].



Slika 1. Struktura Contiki-NG direktorijuma [1]

Izgled OpenMote-B uređaja koji je korišten u eksperimentalnoj postavci predstavljen je na sl.2. [5] - [7]. OpenMote-B sadrži dva SMA antenska konektora za 868 MHz i 2,4 GHz antene. Sub-GHz antenski konektor je direktno povezan na sub-GHz radio na AT86RF215. Antena za opseg 2,4 GHz je multipleksirana pomoću RF prekidača na TI CC2558 i AT86RF215 radio primopredajnike. Glavna ploča takođe uključuje dva porta (3 pina sa razmakom od 2.54 mm) za mjerjenje trenutne potrošnje cijelog sistema tokom rada. Prvi port omogućava mjerjenje trenutne potrošnje CC2538 čipa, dok drugi port omogućava mjerjenje trenutne potrošnje AT86RF215 čipa.

OpenMote-B uređaji sadrže četiri LE diode (zelena, žuta, narandžasta i crvena) i korisnički taster koji je namijenjen za oticanje grešaka (eng. debugging). Pored toga, glavna ploča uključuje taster za resetovanje koji omogućava resetovanje hardvera. Može da komunicira sa računarcem koristeći UART port na TI CC2538.



Slika 2. Izgled i dimenzije OpenMote-B uređaja [5]

II. PORTOVANJE CONTIKI-NG OPERATIVNOG SISTEMA

A. Portovanje Contiki-NG OS na OpenMote-B uređaj

Za portovanje Contiki-NG OS na OpenMote-B uređaju korišćen je uređaj-računar Raspberry Pi 3 Model B V1.2 sa 64-bitnim operativnim sistemom Raspbian, koji je baziran na Linuxu. Postoje dva načina instalacije Contiki-NG na Raspberry Pi. Prvi način je korišćenje Docker slike (eng. Docker images), pri čemu Contiki-NG obezbeđuje veliki broj slika na DockerHub-u. Drugi način, koji je i izabran za ovaj rad, jeste da se izvrši manuelna instalacija za operativni sistem koji već postoji na uređaju. Kako Raspberry Pi uređaj koristi Linux operativni sistem, praćeno je uputstvo za instalaciju na Linux [8].

Portovanje skripti iz odgovarajućih direktorijuma Contiki-NG operativnog sistema na OpenMote-B uređaje je vršeno na identičan način. Prvo je vršeno kreiranje skripte izabrane aplikacije na Raspberry Pi tako što se pozicionira u direktorijum sa određenim primjerom/skriptom i izvrši komanda make kojom se kreira program sa ekstenzijom .native i podfolder build u kome se nalaze folderi sa skriptama koje su se portovale na odgovarajuću platformu. Kada se u željenom direktorijumu izvrši samo komanda make bez definisanja mete na koju želimo da izvršimo portovanje, to znači da je *TARGET=native*, odnosno platforma nije odabrana i kreiranje skripte se vrši na matičnom uređaju (u ovom slučaju Raspberry Pi). Nakon pokretanja native skripte na Raspberry Pi vršeno je portovanje iste aplikacije na OpenMote-B uređaj što će biti opisano u nastavku.

B. Portovanje leds aplikacije na OpenMote-B

Aplikacija *leds* omogućava uključivanje i isključivanje dioda na OpenMote-B uređaju. Prvo se vrši pozicioniranje u odgovarajući direktorijum (u ovom slučaju direktorijum je contiki-ng/examples/dev/leds), a potom se vrši komanda za portovanje ove skripte na OpenMote-B uređaj. Kako uređaj ima ugrađene LED, vrši se njihovo ciklično uključivanje i isključivanje koje podrazumijeva pet različitih stanja koja se ponavljaju.

Prije izvršavanja komande potrebno je povezati OpenMote-B uređaj na Raspberry Pi i očitati odgovarajuće portove dodijeljene OpenMote-B (OpenMote-B ima dva serijska interfejsa) da bi se omogućilo portovanje skripte. Komanda za provjeru očitanih portova na Raspberry Pi je:

```
$ ls /dev/ttyUSB*
```

Unosom komande očitano je da se OpenMote-B uređaj predstavlja sa /dev/ttyUSB0 i /dev/ttyUSB1.

Komanda za portovanje skripte leds na OpenMote-B uređaj je:

```
$ make TARGET=openmote BOARD=openmote-b  
PORT=/dev/ttyUSB1 leds.upload
```

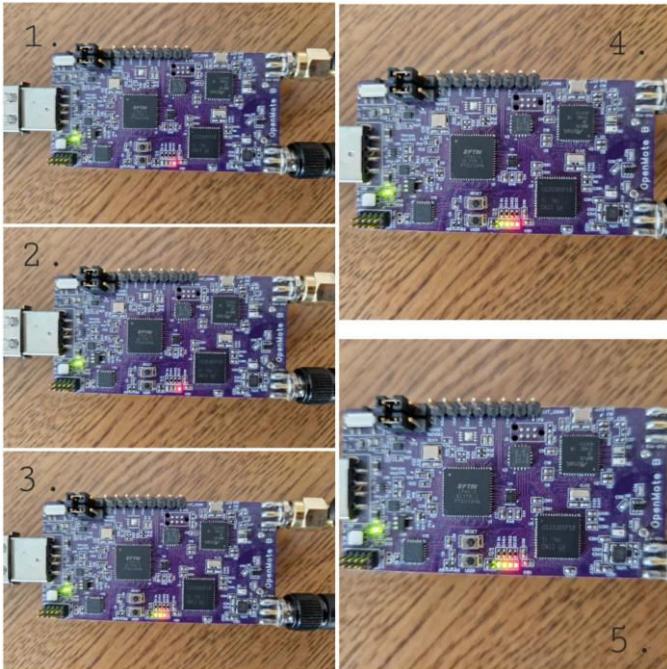
Prilikom portovanja aplikacije *leds* na OpenMote-B bilo je potrebno izvršiti izmjene u skripti Makefile.openmote koja se nalazi u direktorijumu arch/platform/openmote/. Promijenjena je baud brzina (eng. baud rate) sa 450000 na 460800 i dodata je komanda za invertovanje linija serijskog bootloader-a (koji se koristi za portovanje na OpenMote-B uređaje) u BSL_FLAGS na način dat u nastavku, nakon čega je portovanje na OpenMote-B uređaj bilo uspješno.

```
BSL_FLAGS += -e  
ifeq ($(BOARD),openmote-b)  
    BSL_FLAGS += --bootloader-invert-lines  
endif  
BSL_FLAGS += -w -v -b 460800
```

U ovom primjeru nema ispisa u terminalu, već je vidljivo uključivanje i isključivanje dioda na OpenMote-B uređaju i to sljedećim redoslijedom:

1. crvena
2. crvena
3. zelena, žuta, narandžasta
4. zelena, žuta, narandžasta, crvena
5. zelena, žuta, narandžasta, crvena

Primjer uključivanja i isključivanja dioda prikazan je na sl. 3.



Slika 3. Primjer uključivanja i isključivanja dioda pokretanjem skripte leds

III. RPL GRANIČNI RUTER I COAP SERVER

U ovom dijelu rada je kreirana mreža od dva OpenMote-B uređaja povezana na Raspberry Pi. Prvi OpenMote-B je podešen kao granični ruter (eng. border router) portovanjem skripte *border-router.c* koja se nalazi u direktorijumu */home/raspberrypi/contiki-ng/examples/rpl-border-router* u koji se trebalo prvo pozicionirati i izvršiti narednu komandu:

```
$ make TARGET=openmote BOARD=openmote-b
PORT=/dev/ttyUSB1 border-router.upload
```

Na ovaj način mote postaje *DAG root*, ali i Internet pristupna tačka za sve čvorove u mreži. Makefile u ovim primjerima uključuje niz meta podataka koje pomažu u postavljanju graničnog rutera na strani hosta (npr. linux). Komanda "make target" će pozvati tunslip6 koristeći neke zadane parametre, uključujući podrazumijevano ime serijskog porta (*/dev/ttyUSB1* u ovom slučaju):

```
$ make TARGET=openmote BOARD=openmote-b
PORT=/dev/ttyUSB1 connect-router
```

Prethodnom komandom se pokreće program *tools/tunslip6*, koji povezuje Contiki-NG granični ruter sa hostom (u ovom slučaju linux) preko tun interfejsa. Podrazumijevani prefiks adrese mreže je *fd00::1/64* (može se konfigurisati preko make *PREFIX* varijable). Hostu je dodijeljena adresa *fd00::1*. RPL granični ruter, koji radi na OpenMote-B uređaju, ima automatski konfiguriran adresu koja se može pročitati na terminalu nakon pokretanja tun interfejsa, što je prikazano na

sl. 4. Može se očitati da je IPv6 adresa graničnog rutera *fd00::212:4b00:14b5:d354*.

```
raspberrypi@raspberrypi:~ $ /dev/ttyUSB1
raspberrypi@raspberrypi:~ cd /home/raspberrypi/contiki-ng/examples
raspberrypi@raspberrypi:~/contiki-ng/examples$ cd rpl-border-router
raspberrypi@raspberrypi:~/contiki-ng/examples/rpl-border-router$ make TARGET=openmote BOARD=openmote-b PORT=/dev/ttyUSB1 connect-router
make: *** No targets specified and no makefile found in ./rpl-border-router.  Stop.
***** SLIP started on /dev/ttyUSB1
opened tun device '/dev/tun0'
ifconfig tun0 up mtu 1500 up
ifconfig tun0 add fd00::1/64
ifconfig tun0 add fe80::0:1/64
ifconfig tun0
```

Slika 4. Pokretanje tun interfejsa i konekcija graničnog rutera

Tun interfejs sve vrijeme izvršavanja primjera mora ostati aktivan u jednom od terminala. Za lokalni ping samog graničnog rutera potrebno je otvoriti novi terminal i izvršiti sljedeću komandu:

```
$ ping6 fd00::212:4b00:14b5:d354
```

Rezultat ping komande je prikazan na sl. 5.

```
raspberrypi@raspberrypi:~ $ ping6 fd00::212:4b00:14b5:d354
PING fd00::212:4b00:14b5:d354(fd00::212:4b00:14b5:d354) 56 data bytes
64 bytes from fd00::212:4b00:14b5:d354: icmp_seq=1 ttl=64 time=30.1 ms
64 bytes from fd00::212:4b00:14b5:d354: icmp_seq=2 ttl=64 time=20.2 ms
64 bytes from fd00::212:4b00:14b5:d354: icmp_seq=3 ttl=64 time=26.0 ms
64 bytes from fd00::212:4b00:14b5:d354: icmp_seq=4 ttl=64 time=32.4 ms
64 bytes from fd00::212:4b00:14b5:d354: icmp_seq=5 ttl=64 time=22.6 ms
64 bytes from fd00::212:4b00:14b5:d354: icmp_seq=6 ttl=64 time=28.4 ms
64 bytes from fd00::212:4b00:14b5:d354: icmp_seq=7 ttl=64 time=34.5 ms
64 bytes from fd00::212:4b00:14b5:d354: icmp_seq=8 ttl=64 time=24.5 ms
...
--- fd00::212:4b00:14b5:d354 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7013ms
rtt min/avg/max/mdev = 20.155/27.394/34.462/4.583 ms
raspberrypi@raspberrypi:~ $
```

Slika 5. IPv6 ping graničnog rutera

Za nastavak rada je potrebno fizički povezati drugi OpenMote-B uređaj na Raspberry Pi i provjeriti kojim serijskim interfejsima, odnosno portovima se predstavlja (u ovom slučaju OpenMote-B platforma na kojoj se izvršava skripta za granični ruter se predstavlja portovima USB0 i USB1, dok se OpenMote-B uređaj na kome će biti pokrenut CoAP server predstavlja sa portovima USB2 i USB3).

Nakon fizičkog povezivanja druge OpenMote-B platforme na Raspberry Pi potrebno je pozicionirati se u direktorijum */home/raspberrypi/contiki-ng/examples/coap/coap-example-server* i izvršiti sljedeću komandu za učitavanje *coap-example-server.c* na drugi OpenMote-B:

```
$ make TARGET=openmote BOARD=openmote-b
PORT=/dev/ttyUSB3 coap-example-server.upload
```

Prije izvršavanja komande potrebno je u skripti *coap-example-server.c* izvršiti odgovarajuće promjene. Linije kôda koje se odnose na platforme koje imaju senzor za temperaturu

se označe kao komentar kako bi se moglo izvesti ispravno kompajliranje, jer OpenMote B uređaj ne sadrži ugrađene senzore temperature i Contiki-NG ne podržava direktno senzorske podatke preko CoAP servera.

Poslije uspješnog kompajliranja .c skripte na OpenMote-B, on postaje CoAP server i povezuje se u mrežu. Da bismo saznali IPv6 adresu CoAP server, potrebno je izvršiti komande:

```
$ wget -6 "http://[fd00::212:4b00:14b5:d354]/"  
$ cat index.html
```

Prvom komandom se sve adrese koje su povezane na granični ruter snimaju u fajl index.html, a drugom komandom se izlistavaju te adrese tako da je nakon izvršavanja ovih komandi dobijena adresa CoAP servera - fd00::212:4b00:14b5:d332. Rezultat ping komande ka toj adresi prikazan je na sl. 6.

```
raspberrypi@raspberrypi:~ $ ping6 fd00::212:4b00:14b5:d332
PING fd00::212:4b00:14b5:d332(fd00::212:4b00:14b5:d332) 56 data bytes
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=1 ttl=63 time=101 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=2 ttl=63 time=43.5 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=3 ttl=63 time=97.6 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=4 ttl=63 time=71.5 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=5 ttl=63 time=110 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=6 ttl=63 time=84.4 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=7 ttl=63 time=90.9 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=8 ttl=63 time=65.4 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=9 ttl=63 time=55.5 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=10 ttl=63 time=61.7 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=11 ttl=63 time=51.6 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=12 ttl=63 time=89.4 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=13 ttl=63 time=95.9 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=14 ttl=63 time=70.6 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=15 ttl=63 time=60.8 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=16 ttl=63 time=51.7 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=17 ttl=63 time=58.3 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=18 ttl=63 time=80.7 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=19 ttl=63 time=54.6 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=20 ttl=63 time=60.8 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=21 ttl=63 time=67.8 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=22 ttl=63 time=74.6 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=23 ttl=63 time=80.7 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=24 ttl=63 time=70.7 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=25 ttl=63 time=92.7 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=26 ttl=63 time=99.6 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=27 ttl=63 time=41.7 ms
64 bytes from fd00::212:4b00:14b5:d332: icmp_seq=28 ttl=63 time=82.9 ms
```

Slika 6. IPv6 ping CoAP server

Veza sa CoAP serverom nije uvek bila stabilna zbog interferencije sa drugim mrežama, pa je zbog toga pušten kontinualni ping ka CoAP serveru za konstantnu provjeru veze.

U daljem postupku se očitavao status tastera USER na OpenMote-B uređaju na kome je pokrenut CoAP server. Logovanjem na pomenuti OpenMote-B (komanda *login*) može se u terminalu vidjeti status tastera *USER*. Svaki pritisak tastera se broji i ispisuje se redni broj pritiska u terminalu (na sl. 7 TICK 1, 2, 3, ... for /sensors/button). Pritiskom tastera RESET na OpenMote-B se brojanje poništava, vrši se odgovarajući ispis u terminalu i kreće se u novi ciklus brojanja pritisaka na taster USER kao što je ilustrovano na sl. 8.

```
[nagori@raspberrypi:~/raspberry_pi/serial3/coolexamples/cool-example-server]$ make TARDISopenremote BOARD=openremote-ls PORT=/dev/ttyUSB3 login  
[nagori@raspberrypi:~/raspberry_pi/serial3/coolexamples/cool-example-server]$ ./cool-example-server  
connecting to /dev/ttyUSB3 [OK]  
[000 : App ] TICK 0 For /sensors/button  
[000 : App ] [***** BUTTON*****]  
[000 : App ] TICK 1 For /sensors/button  
[000 : App ] [***** BUTTON*****]  
[000 : App ] TICK 2 For /sensors/button  
[000 : App ] [***** BUTTON*****]  
[000 : App ] TICK 3 For /sensors/button  
[000 : App ] [***** BUTTON*****]  
[000 : App ] TICK 4 For /sensors/button  
[000 : App ] [***** BUTTON*****]  
[000 : App ] TICK 5 For /sensors/button  
[000 : App ] [***** BUTTON*****]  
[000 : App ] TICK 6 For /sensors/button  
[000 : App ] [***** BUTTON*****]  
[000 : App ] TICK 7 For /sensors/button  
[000 : App ] [***** BUTTON*****]  
[000 : App ] TICK 8 For /sensors/button  
[000 : App ] [***** BUTTON*****]  
[000 : App ] TICK 9 For /sensors/button  
[000 : App ] [***** BUTTON*****]  
[000 : App ] TICK 10 For /sensors/button  
[000 : App ] [***** BUTTON*****]  
[000 : App ] TICK 11 For /sensors/button  
[000 : App ] [***** BUTTON*****]  
[000 : App ] TICK 12 For /sensors/button  
[000 : App ] [***** BUTTON*****]  
[000 : App ] TICK 13 For /sensors/button  
[000 : App ] [***** BUTTON*****]  
[000 : App ] TICK 14 For /sensors/button  
[000 : App ] [***** BUTTON*****]  
[000 : App ] TICK 15 For /sensors/button
```

Slika 7. Očitavanje statusa tastera USER u terminalu

```
[DBG : App] TICK 2 for /sensors/button
[DBG : App] *****'BUTTON'*****
[DBG : App] TICK 3 for /sensors/button
[DBG : App] *****'BUTTON'*****
[DBG : App] TICK 4 for /sensors/button
[DBG : App] *****'BUTTON'*****
[DBG : App] TICK 5 for /sensors/button
[DBG : App] *****'BUTTON'*****
[DBG : App] TICK 6 for /sensors/button
[DBG : App] *****'BUTTON'*****
[DBG : App] TICK 7 for /sensors/button
[DBG : App] *****'BUTTON'*****
[DBG : App] TICK 8 for /sensors/button
[DBG : App] *****'BUTTON'*****
[DBG : App] TICK 9 for /sensors/button
[INFO@: Main] Starting Contiki-NG-develop/v4.9-334-gbe094ba77-dirty
[INFO@: Main] - Routing: RPL Lite
[INFO@: Main] - Net: sicslowpan
[INFO@: Main] - MAC: CSMA
[INFO@: Main] - 802.15.4 PANID: 0xabcd
[INFO@: Main] - 802.15.4 Default channel: 26
[INFO@: Main] Node ID: 54666
[INFO@: Main] Link-layer address: 0012.4b00.14b5.d332
[INFO@: Main] Tentative link-local IPv6 address: fe80::212:4b00:14b5:d332
[INFO@: OpenMote]
[INFO@: App] Starting Erbium Example Server
[DBG : App] *****'BUTTON'*****
[DBG : App] TICK 1 for /sensors/button
[DBG : App] *****'BUTTON'*****
[DBG : App] TICK 2 for /sensors/button
[DBG : App] *****'BUTTON'*****
[DBG : App] TICK 3 for /sensors/button
[DBG : App] *****'BUTTON'*****
[DBG : App] TICK 4 for /sensors/button
[DBG : App] *****'BUTTON'*****
[DBG : App] TICK 5 for /sensors/button
[DBG : App] *****'BUTTON'*****
[DBG : App] TICK 6 for /sensors/button
[DBG : App] *****'BUTTON'*****
[DBG : App] TICK 7 for /sensors/button
```

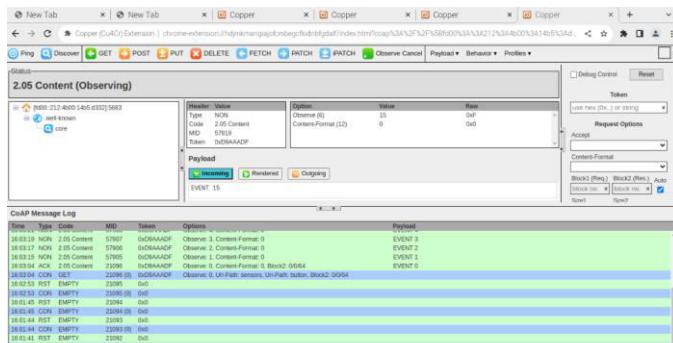
Slika 8. Resetovanje ciklusa brojanja pritiskanja tastera USER pritiskom tastera RESET

Za pristup IoT uređajima baziranim na CoAP-u, tj. OpenMote-B uređaju na kome je pokrenut CoAP server i očitavanje podataka putem pretraživača korištenjem IPv6 adrese, potrebno je instalirati tzv. Copper dodatak (eng. Copper add-on). U ovom slučaju je Copper dodatak instaliran na default pretraživač na Raspberry Pi uređaju – Chromium pretraživač. Postupak instalacije na Chromium urađen je na osnovu uputstva [10].

Nakon instalacije, pokretanjem copper ekstenzije, moguće je realizovati CoAP interfejs. U otvorenu ekstenziju je potrebno upisati CoAP krajnju tačku na sljedeći način:

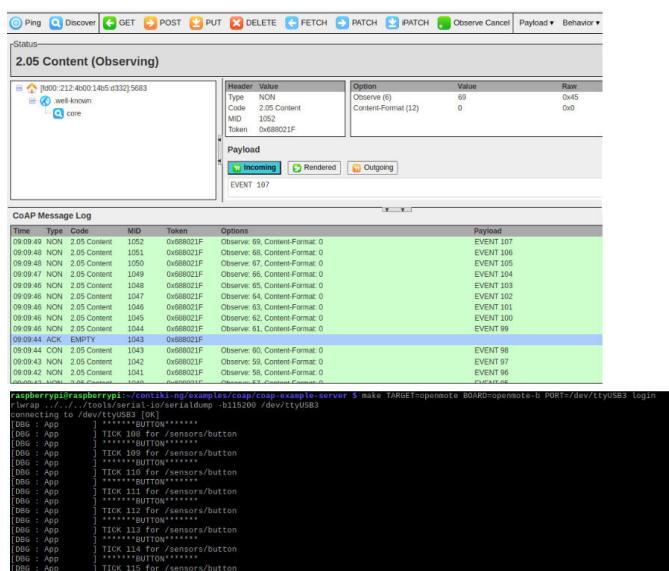
Nakon pokretanja CoAP interfejsa potrebno je konekciju sa CoAP serverom provjeriti pokretanjem ping komande. Ako je konekcija ispravna i postoji odgovor od CoAP servera, može se pokrenuti opcija posmatranja/osluškivanja poruka koje se dobijaju od CoAP servera izborom zelenog tastera Observing

(opcija dobijanja CoAP zahtjeva – eng. get CoAP request). Na sl. 9 je prikazan CoAP interfejs za krajnu tačku sensors/button kao i očitavanje statusa tastera USER na OpenMote-B uređaju na kome je pokrenut CoAP server. Svako pritiskanje tastera se bilježi kao jedan događaj (eng. EVENT).



Slika 9. CoAP interfejs za OpenMote-B uređaj

Ukoliko se desi da se izgubi konekcija sa CoAP serverom, u pretraživaču se neće očitavati status tastera USER što se može uočiti na sl. 10, gdje je nakon izgubljene konekcije sa CoAP serverom nastavljeno pritiskanje tastera USER a redni broj pritiska se nastavlja bilježiti u terminalu nakon logovanja na dati OpenMote-B.



Slika 10. Izgubljena konekcija sa CoAP serverom

ZAKLJUČAK

Contiki-NG OS predstavlja odličnu bazu za rad sa IoT uređajima. Ovaj OS ima veoma detaljnu dokumentaciju i podršku korisničke zajednice. Moguće ga je implementirati na veliki broj aktuelnih IoT platformi. Za komunikaciju Contiki-NG OS sa OpenMote-B platformom se koristi serijski interfejs posredstvom „serial boot loader-a“. Naravno, uvijek su potrebna određena prilagođenja i izmjene programa za implementiranje pojedinih ličnih primjera.

Izvršeni i kompajlirani programi na OpenMote-B platformu predstavljaju dobru osnovu za dalji rad i proširivanje upotrebe sa dodatnim senzorima i očitavanjem njihovih mjerjenja (npr. dodatna ploča se senzorima koja može direktno da se poveže na OpenMote-B platformu, a sadrži senzore za mjerjenje temperature, količine svjetlosti i vlažnosti).

Osnovni uočeni nedostatak portovanja Contiki-NG na OpenMote-B platformu jeste što se ne može jednostavno izvršiti portovanje OS u svakom novom primjeru. Za portovanje nove aplikacije bilo je neophodno koristiti novi OpenMote-B uređaj, bez mogućnosti reflashing-a. Jedini način brisanja učitanih programa sa OpenMote-B platforme jeste potpuno resetovanje platforme i vraćanje na fabrička podešavanja.

U opštem slučaju, korišćenje i portovanje Contiki-NG OS na OpenMote-B platforme je „user friendly“ i poprilično intuitivno. Najviše vremena je potrebno posvetiti razumijevanju i internoj organizaciji operativnog kao i pronalaženju rješenja za konfigurisanje i kompajliranje aplikacija na platformu.

ZAHVALNICA

Zahvaljujemo se prof. dr Gordani Gardašević za mentorstvo pri izradi projektnog zadatka na temu Portovanje Contiki-NG operativnog sistema na OpenMote-B uređaje za industrijske IoT primjene iz predmeta Bežične senzorske mreže na drugom ciklusu studija Elektrotehničkog fakulteta, Univerziteta u Banjoj Luci.

LITERATURA

- [1] George Oikonomou, Simon Duquennoy, Atis Elsts, Joakim Eriksson, Yasuyuki Tanaka, Nicolas Tsiftes, “The Contiki-NG open source operating system for next generation IoT devices.” <https://www.sciencedirect.com/science/article/pii/S2352711022000620>, Posjećeno 05.09.2023.
- [2] “Contiki.”, Wikipedia, <https://en.wikipedia.org/wiki/Contiki>, Posjećeno 05.09.2023.
- [3] “Contiki-NG Documentation.”, <https://docs.contiki-ng.org/en/develop/>, Posjećeno 12.09.2023.
- [4] “Contiki-NG.”, Fit IoT-LAB, <https://iot-lab.github.io/docs/os/contiki-ng/>, Posjećeno 12.09.2023.
- [5] Open Mote B, Open Mote B User Guide, Industrial Shields, 2019
- [6] Industrial Equipment based on Arduino and Raspberry Pi, *Industrial Shields*
- [7] “Industrial Shields OpenMote B Boards.”, Mouser Electronics, <https://eu.mouser.com/new/industrial-shields/industrial-shields-openmote-b-boards/>, Posjećeno 12.09.2023.
- [8] “Toolchain installation on Linux.”, Github, <https://docs.contiki-ng.org/en/master/doc/getting-started/Toolchain-installation-on-Linux.html>, Posjećeno 13.09.2023.
- [9] “Raspberry Pi.”, Raspberry Pi Fondation, <https://www.raspberrypi.org/>, Posjećeno 13.09.2023.
- [10] “Copper4Cr.”, Github, <https://github.com/mkovatsc/Copper4Cr>, Posjećeno 21.09.2023.
- [11] I. Tomasic, K. Khosraviani, P. Rosengren, M. Jörntén-Karlsson and M. Lindén, “Enabling IoT based monitoring of patients’ environmental

- parameters: experiences from using OpenMote with OpenWSN and Contiki-NG”, Conference Paper, Maj 2018
- [12] “Tutorial: RPL border router.”, Github, <https://github.com/contiki-ng/contiki-ng/wiki/Tutorial:-RPL-border-router/347383e912249e0ccae4f3cec6315ebcb88cc8a3>, Posjećeno 04.10.2023.
- [13] Augus Kurniawan, “Practical Contiki-NG, Programming for Wireless Snesor Networks”

ABSTRACT

Contiki-NG is an open source OS for next-generation IoT devices. It focuses on secure and reliable low-power communication and standard protocols such as IPv6/6LoWPAN, 6TiSCH, RPL and CoAP. OpenMote-B board is a hardware development and prototyping platform for

the Industrial IoT (IIoT) applications. It is built around the Texas Instruments CC2538 ARM-Cortex-M3 microcontroller and provides simultaneous multiband operation in ISM bands 2.4GHz and 868/915MHz with support for current IEEE 802.15.4 standards. This paper provides an overview of the Contiki-NG OS and the implementation of its various functionalities on OpenMote-B platforms. The practical application with an emphasis on creating a network with an RPL border router and the application of the CoAP protocol for sensors' reading has been described.

Porting Contiki-NG operating system to OpenMote-B devices for Industrial IoT applications

Jelena Banović, Sara Danilović