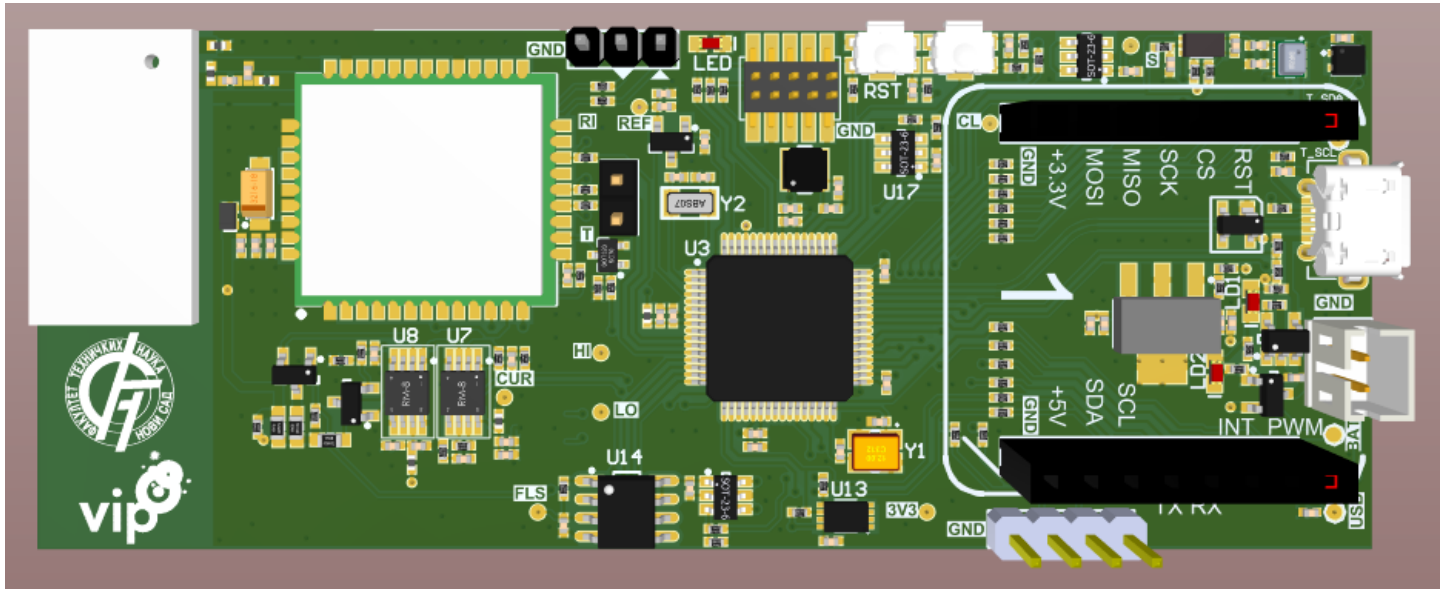
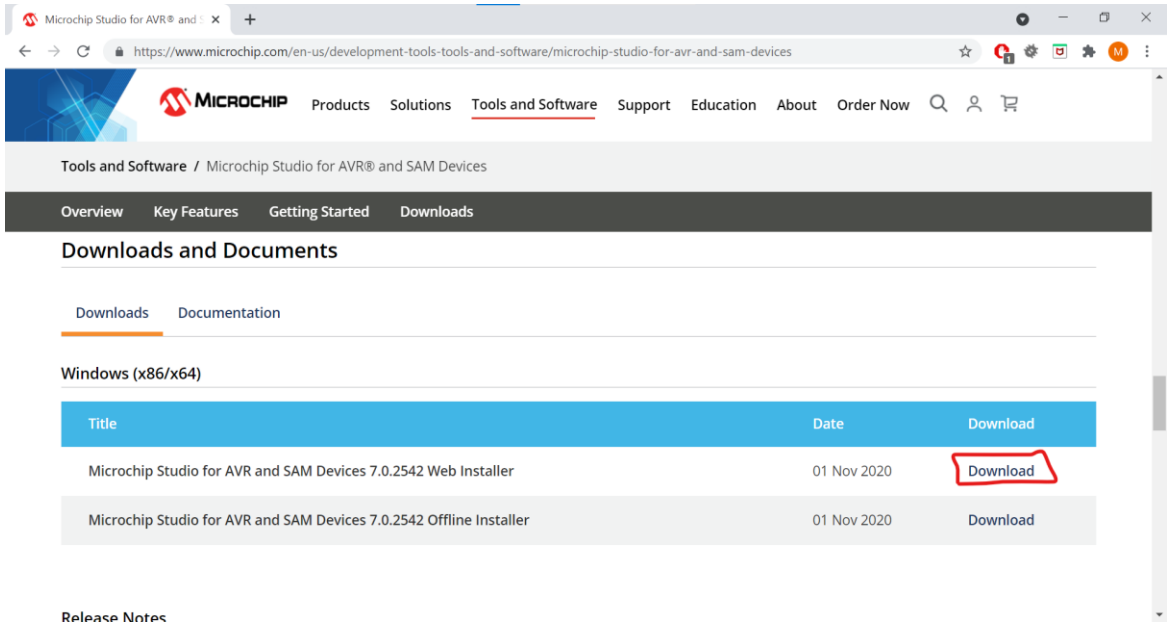
NB-IoT


GND
LED
RST
RI
REF
Y2
U17
U3
U8
U7
CUR
HI
LO
U14
FLS
Y1
U13
3V3
CL
GND
+3.3V
MOSI
MISO
SCK
CS
RST
GND
+5V
SDA
SCL
INT
PWM
TX
RX
vip

Microchip Studio - instalacija

- https://www.microchip.com/en-us/development-tools-tools-and-software/microchip-studio-for-avr-and-sam-devices
- Prilikom instalacije neophodno je instalirati alate za rad sa ATSAM familijom mikrokontrolera

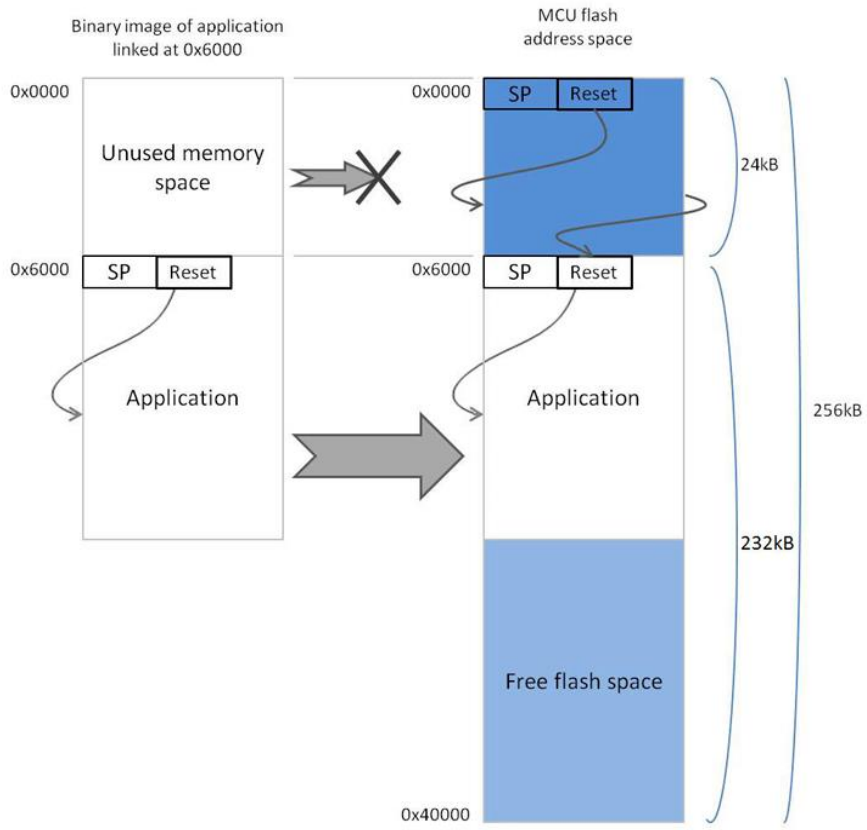


SAM-BA instalacija

- SAM-BA v2.18 je pomoćna aplikacija koja će biti korišćena za programiranje mikrokontrolera ATSAML21J18B
- Instalacija: https://ww1.microchip.com/downloads/en/DeviceDoc/SAM-BA%20v2.18%20for%20Windows.exe

Pokretanje bootloader-a i aplikacije

- Pritiskom RST tastera na ploči, uređaj se prebacuje u jedan od dva moguća režima:
  - Ukoliko je u trenutku pritiska RST prethodno bio pritisnut i drugi taster (PRG), uređaj se prebacuje u bootoladerski režim i čeka na prijem programskog koda od strane SAM-BA aplikacije
  - Ukoliko je pritisnut samo RST taster, a PRG je u tom trenutku bio pušten, uređaj će startovati korisničku aplikaciju



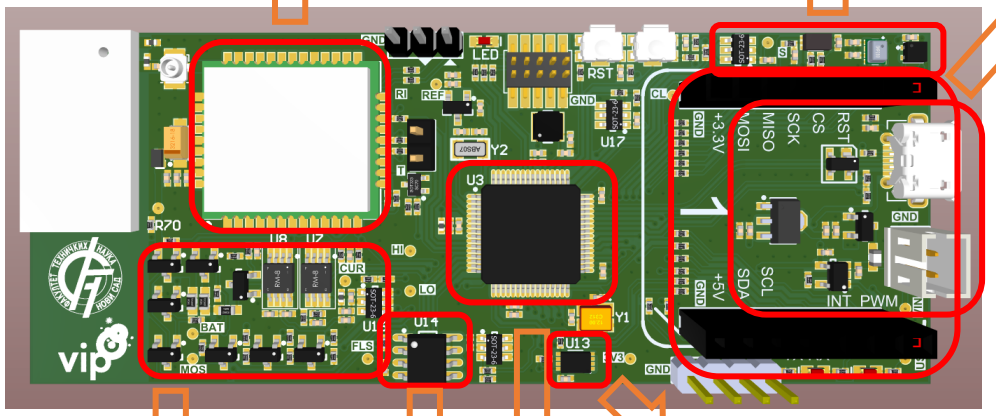
FTN-VIP ploča: Hardver

Senzori:

- Temperatura + vlažnost vazduha
- Temperatura + vazdušni pritisak
- Intenzitet svetla
- 3-osni akcelerometar

Quectel BC68
NB-IoT modul

mikroBUS konektor
za dodatne
senzore/aktuatore



Napajanje:
USB/baterija
+ kolo za punjenje
baterije

Kripto čip = Sigurna memorija za ključeve + TRNG +
kriptografski ko-procesor (ECC, AES, SHA)

Kolo za
merenje struje

Eksterna
FLASH
memorija

- ARM Cortex M0+ MCU @16MHz
- 256KB FLASH programske memorije
- 32KB SRAM-a

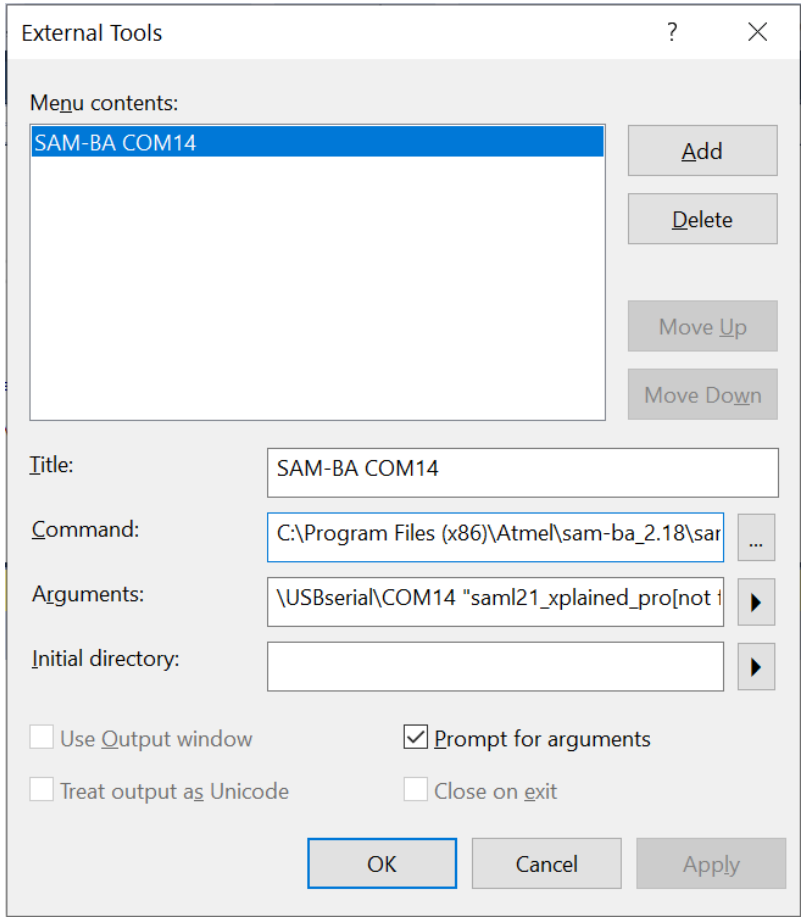
Tehnička dokumentacija

- Mikrokontroler ATSAML21J18B
  https://www.microchip.com/wwwproducts/en/ATSAML21J18B
- Quectel BC68 set AT komandi
  http://www.elektronika.ftn.uns.ac.rs/umrezeni-embeded-sistemi/wp-content/uploads/sites/176/2018/03/Quectel_BC95-GBC68_AT_Commands_Manual_V1.5.pdf
- Šema uređaja
  http://www.elektronika.ftn.uns.ac.rs/umrezeni-embeded-sistemi/wp-content/uploads/sites/176/2018/03/vip-ftn-bc68-nbiot.pdf

- GitHub repozitorijum aplikacije
  https://github.com/milukic/UES_NB-IoT.git

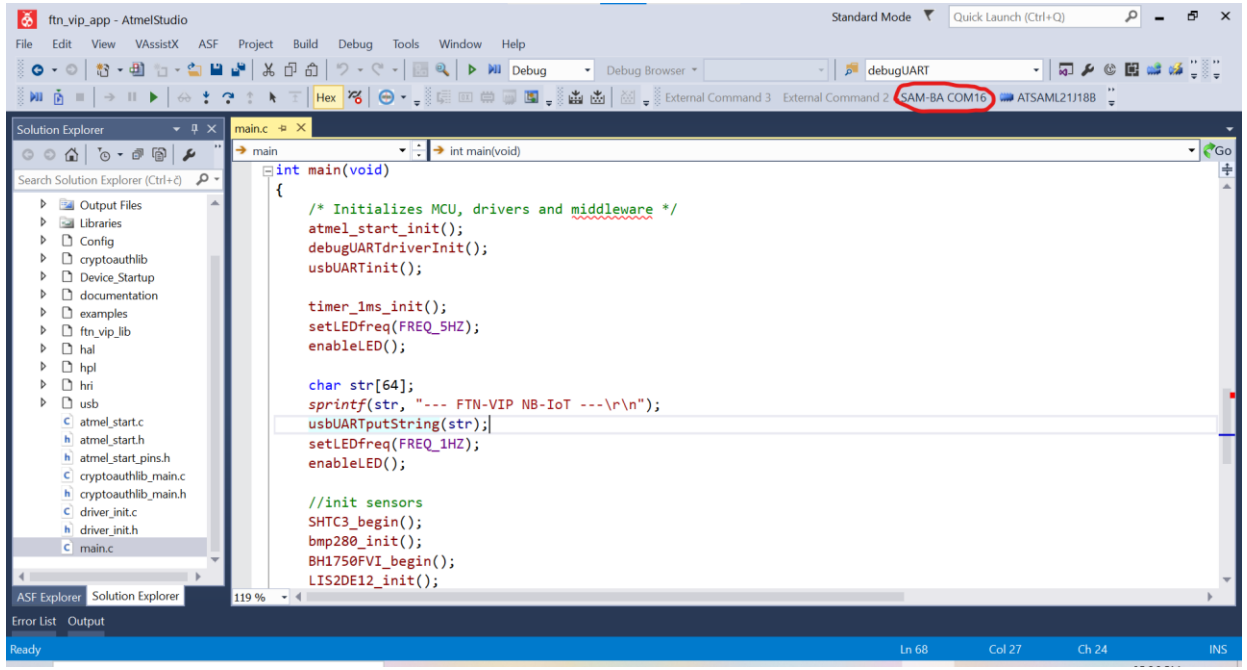
Podešavanje okruženja za rad sa SAM-BA

- U okruženju Microchip Studio izabrati opciju **Tools -> External Tools -> Add**
- U polju **Title** staviti naziv SAM-BA COMxx, gde je xx oznaka COM porta
- U polju **Command** zadati putanju do SAM-BA aplikacije (npr. *C:\Program Files (x86)\Atmel\sam-ba_2.18\sam-ba.exe*)
- U polje **Arguments** upisati: *\USBserial\COMxx "saml21_xplained_pro[not factory programmed]"*



Prevođenje aplikacije

- Po otvaranju aplikacije (ftn_vip_app.atsln), razvoj koda se vrši iz okruženja Microchip Studio IDE (ili Atmel Studio, kao starijoj verziji)
- Nakon unošenja izmena u kodu, kreiranje izvršne verzije koda vrši se pomoću opcije **Build -> Build Solution**, ili pritiskom tastera **F7**
- Nakon što je kompajliranje i linkovanje uspešno izvršeno unutar foldera ftn_vip_app/Debug nalaziće se nova verzija izvršnog koda u datoteci **ftn_vip_app.bin**
- Sledeći korak je prebacivanje ploče u režim bootloadera i pokretanje aplikacije za programiranje (**Tools -> SAM-BA COMxx**)

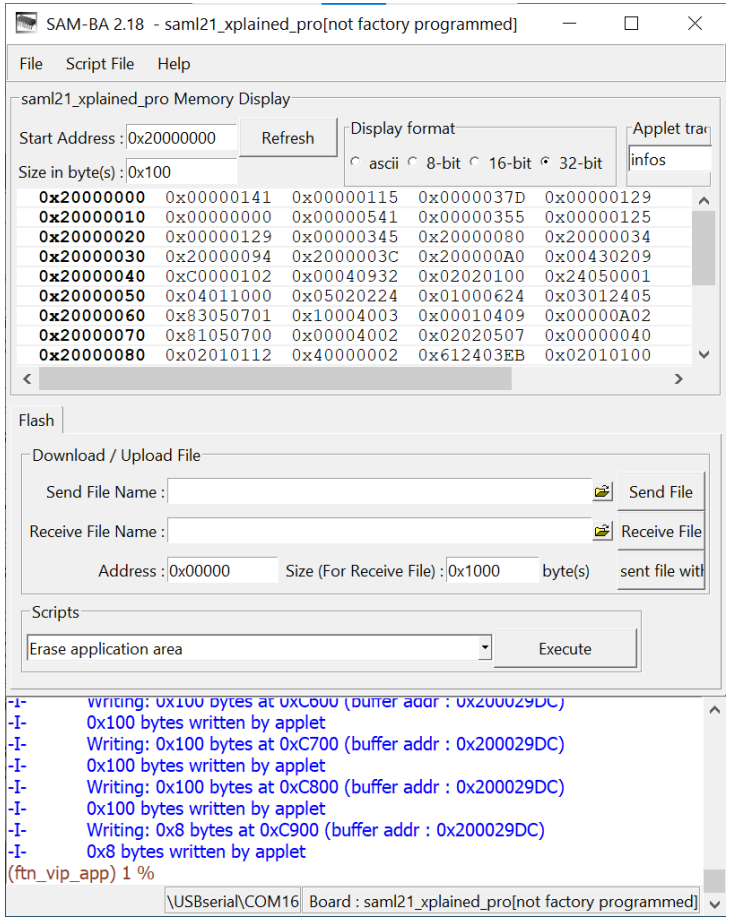


Upis aplikacije u FLASH memoriju mikrokontrolera

- Pokretanjem aplikacije SAM-BA v2.18, omogućen je upis programskog koda u flash memoriju kontrolera ATSAML21J18B
- U okviru datoteke **boot.tcl** smeštene u glavni folder aplikacije **ftn_vip_app**, potrebno je podesiti putanju do izvršne verzije koda, npr:

```
FLASH::Init
FLASH::ScriptGPNMV 1
send_file {Flash} "C:/ftn_vip_app/ftn_vip_app/Debug/ftn_vip_app.bin" 0x06000 0
```

- Pokrenuti opciju **Script File -> Execute Script File**
- Izabrati skriptu **boot.tcl** smeštenu u glavni folder aplikacije ftn_vip_app
- Nakon spuštanja koda, aplikacija se pokreće pritiskom tastera RST



Eho server (UDP)

- Serverska skripta osluškuje dolazni UDP port i za svaki primljeni paket u tekstualnom formatu vraća kapitalizovanu verziju teksta (samo sa velikim slovima)
- Skriptu je neophodno snimiti na server u datoteku **udp_echo.py**
- Pre pokretanja skripte potrebno je u kodu podesiti port na vrednost 50000 + serijski broj pločice, ispisan na nalepnici sa donje strane

  PRIMER:

port=50061

- Pokretanje servera:

  $ python3 udp_echo.py

- Server se zaustavlja kombinacijom tastera **ctrl-z**, nakon čega je potrebno izvršiti komandu:

  $ pkill -9 -f udp_echo.py

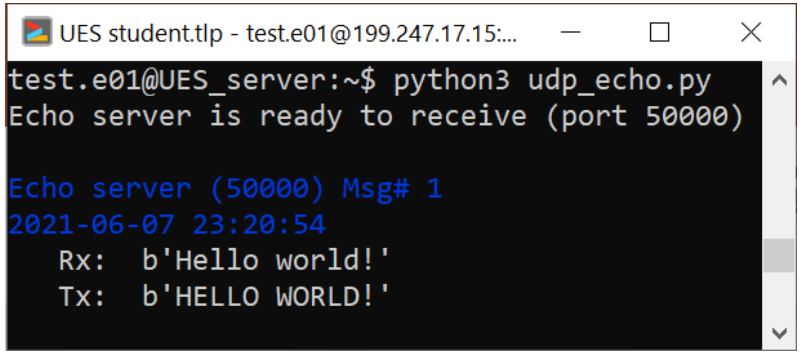
```python
from socket import *
import time

port = 50061
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', port))
print ('Echo server is ready to receive (port ' + str(port) + ')\n')

msgCnt = 1
while True:
    try:
        messageIn, clientAddress = serverSocket.recvfrom(4096)

        ts = time.localtime()
        print('\033[0;34;40mEcho server (' + str(port) + ') Msg#', str(msgCnt))
        print(time.strftime('%Y-%m-%d %H:%M:%S\033[0;37;40m', ts))
        print('    Rx: ', messageIn)
        print('    Tx: ', messageIn.upper())
        serverSocket.sendto(messageIn.upper(), clientAddress)

        msgCnt += 1
    except:
        print('ERROR in Echo UDP')
```

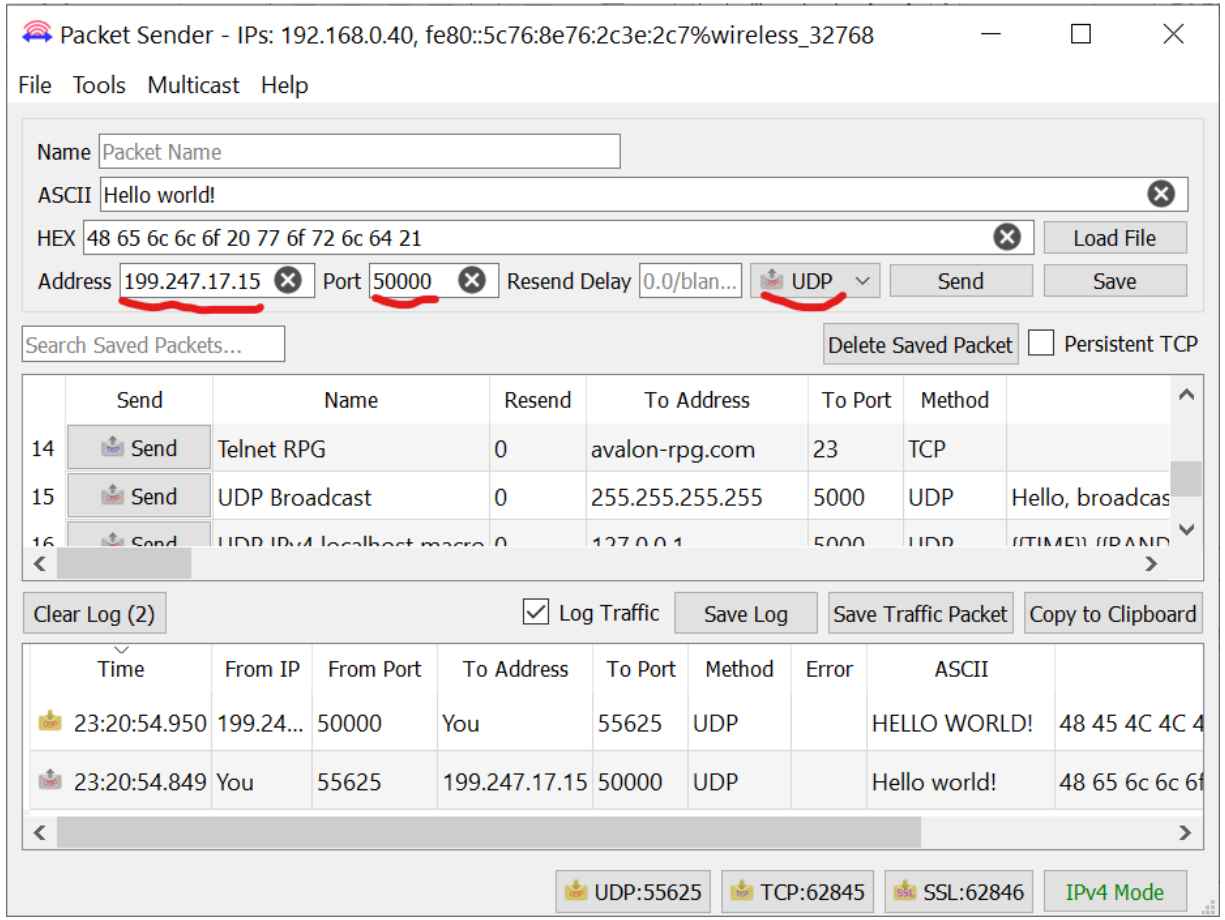
Eho server (TCP)

```
import socket

HOST = ''                    # Symbolic name meaning all available interfaces
PORT = 50061                 # Arbitrary non-privileged port
NUM_OF_CLIENTS = 1

tcpSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
tcpSocket.bind((HOST, PORT))
tcpSocket.listen(NUM_OF_CLIENTS)

print ('Echo server is ready to receive (port ' + str(PORT) + ')\n')

msgCnt = 1
while True:
    try:
        conn, addr = tcpSocket.accept()
        print ('Connected by', addr)
        while True:
            data_in = conn.recv(1024)
            if not data_in:
                break
            data_out = data_in.upper()
            print ('   Rx: ', data_in)
            print ('   Tx: ', data_out)
            conn.sendall(data_out)
    except:
        conn.close()
        print('ERROR in Echo TCP')
```

Packet Sender – pomoćni alat za slanje paketa

- https://github.com/dannagle/PacketSender/releases/download/v7.2.3/PacketSenderPortable_v7.2.3.zip



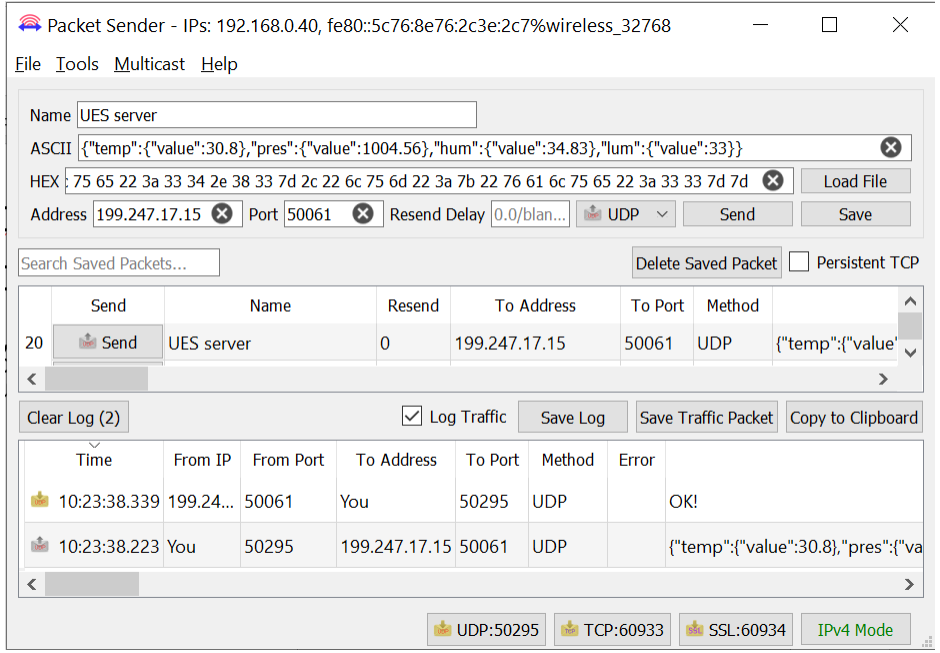


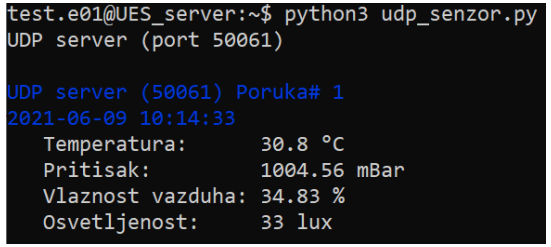
Primer: Upis sadržaja UDP paketa u bazu

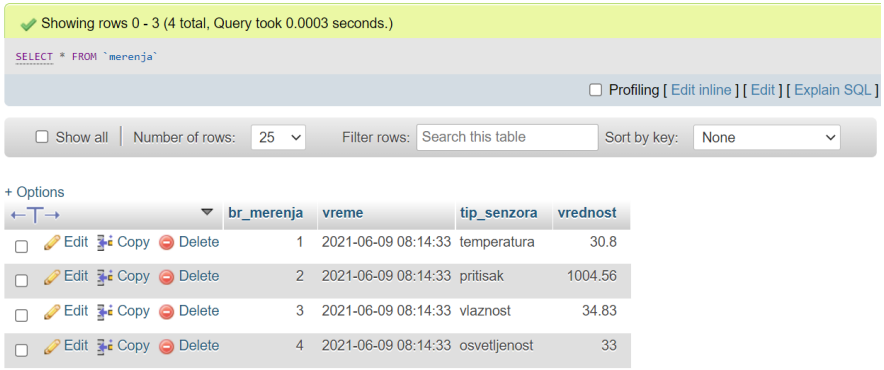
- U ovom primeru, sadržaj UDP paketa (očitavanja senzora u JSON formatu) upisuje se u bazu podataka, u postojeću tabelu **merenja**

PRIMER:

```
{
    "temp": {
        "value": 30.8
    },
    "pres": {
        "value": 1004.56
    },
    "hum": {
        "value": 34.83
    },
    "lum": {
        "value": 33
    }
}
```







```python
from socket import *
import time
import mysql.connector
import json

mydb = mysql.connector.connect(
  host = "localhost",
  user = "test.e01",
  password = "...",
  database = "db_test_e01"
)

port = 50061
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('', port))
print ('UDP server (port ' + str(port) + ')\n')

msgCnt = 1
while True:
    try:
        messageIn, clientAddress = serverSocket.recvfrom(4096)

        ts = time.localtime()
        print('\033[0;34;40mUDP server (' + str(port) + ') Poruka#', str(msgCnt))
        print(time.strftime('%Y-%m-%d %H:%M:%S\033[0;37;40m', ts))

        data = json.loads(messageIn.decode("utf-8"))
        print ("   Temperatura:     ", data["temp"]["value"], "\u00b0C")
        print ("   Pritisak:        ", data["pres"]["value"], "mBar")
        print ("   Vlaznost vazduha:", data["hum"]["value"], "%")
        print ("   Osvetljenost:    ", data["lum"]["value"], "lux")

        mycursor = mydb.cursor()
        mycursor.execute("INSERT INTO merenja (tip_senzora, vrednost) VALUES ('temperatura'," + str(data["temp"]["value"]) + ")")
        mycursor.execute("INSERT INTO merenja (tip_senzora, vrednost) VALUES ('pritisak'," + str(data["pres"]["value"]) + ")")
        mycursor.execute("INSERT INTO merenja (tip_senzora, vrednost) VALUES ('vlaznost'," + str(data["hum"]["value"]) + ")")
        mycursor.execute("INSERT INTO merenja (tip_senzora, vrednost) VALUES ('osvetljenost'," + str(data["lum"]["value"]) + ")")
        mydb.commit()

        serverSocket.sendto(bytearray("OK!", "utf-8"), clientAddress)

        msgCnt += 1
    except:
        print('Greska!!!')
```

Source kodovi

- Serverske skripte (Python):

https://github.com/milukic/UES_Python.git

- FW za FTN-VIP ploču:

https://github.com/milukic/UES_NB-IoT.git

Zadaci za vežbu

- Modifikovati FW za VIP-FTN ploču tako da na pritisak tastera šalje očitavanja senzora
- Kreirati web stranicu koja grafički prikazuje pristigle rezultate očitavanja senzora

- **Nešto teži zadatak**: Na web serveru napraviti u HTML-u jednostavnu “Hello world” stranicu. Zatim u FW za VIP-FTN ploču preuzeti sadržaj web stranice. U tu svrhu, potrebno je realizovati aplikativni sloj HTTP protokola, koji koristi TCP konekciju na portu 80.