

# PRIMENA MIKROKONTROLERA- MS1PMK

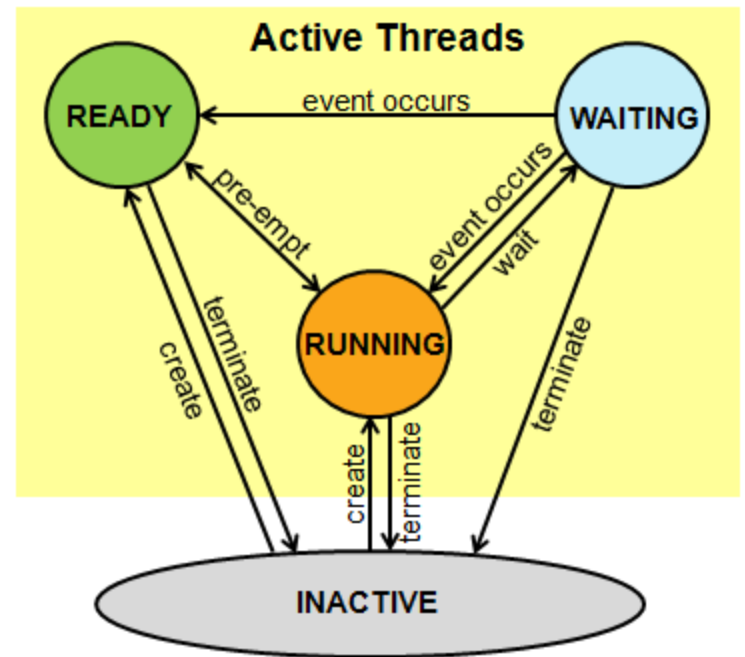
8. deo

2018

Nenad Jovičić

# MBED OS

- Mbed OS postoji u verzijama 2.x i 5.x koje nažalost nisu kompatibilne
- Preporuka je da se koristi verzije 5.x
- <https://os.mbed.com/docs/v5.8>



# Tipičan MBED OS program

- Tipičan program pod operativim sistemom.
- `main(void)` - Default entry point. All the standard application code goes here.
- `mbed_main(void)` - Executed directly before `main`. The user can define this.

```
#include "mbed.h"

Thread thread;
Ticker ticker;
Queue<uint32_t, 5> queue;
DigitalOut myled(LED1);

void queue_isr() {
    queue.put((uint32_t*)2);
    myled = !myled;
}

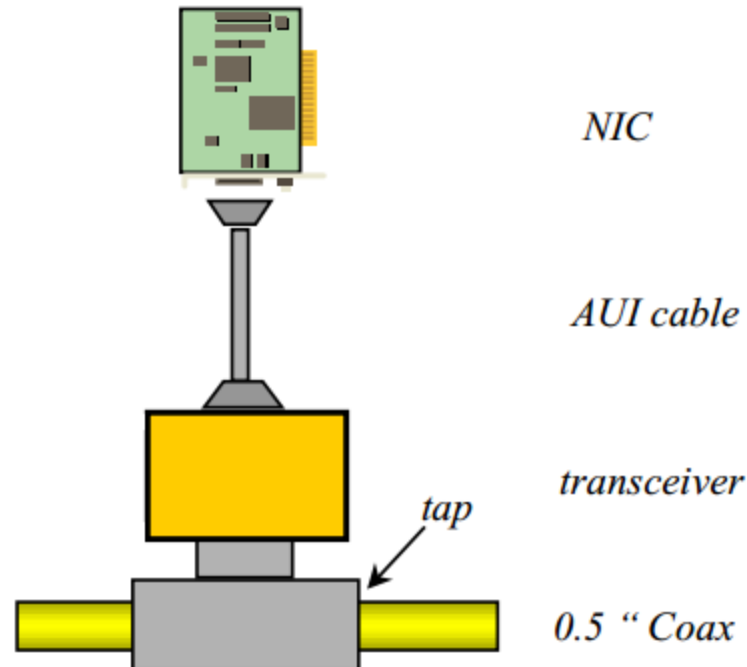
void queue_thread() {
    while (true) {
        queue.put((uint32_t*)1);
        wait(1);
    }
}

int main (void) {
    thread.start(callback(queue_thread));
    ticker.attach(queue_isr, 1.0);

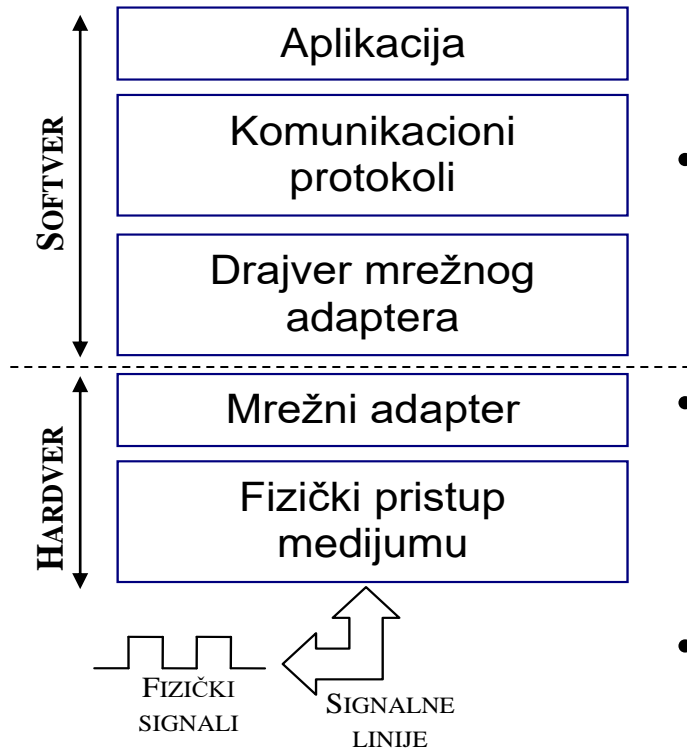
    while (true) {
        osEvent evt = queue.get();
        if (evt.status != osEventMessage) {
            printf("queue->get() returned %02x status\n\r", evt.status);
        } else {
            printf("queue->get() returned %d\n\r", evt.value.v);
        }
    }
}
```

# Ethernet

- Nastao sredinom 70-tih u okviru kompanije Xerox Palo Alto Research
- 1985 standardizovan kao IEEE 802.3 standard



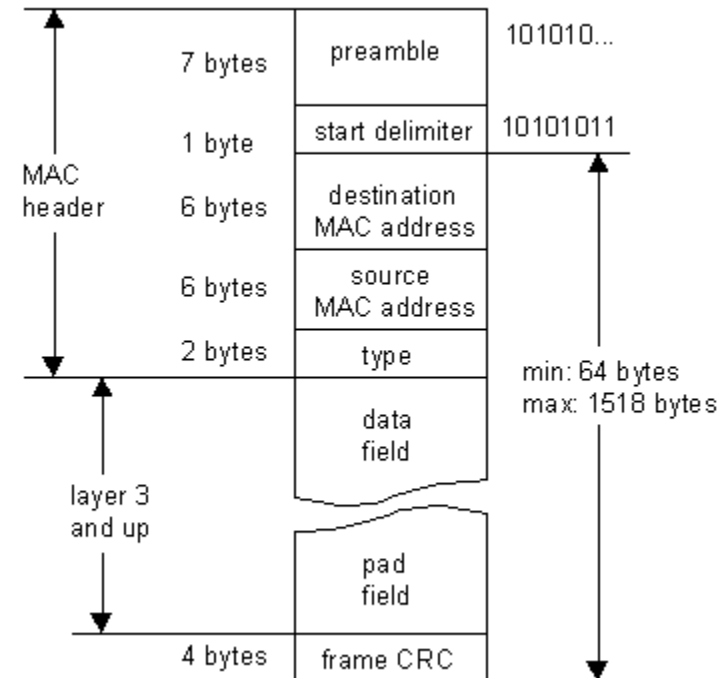
# Ethernet



- **Komunikacioni protokoli** se koriste za implementaciju različitih funkcija u mrežnoj komunikaciji.
- **Drajver mrežnog adaptera** obezbeđuje spregu između softvera koji se izvršava na uređaju i mrežnog adaptera. Drajver može vršiti osnovno formatiranje i upravljati HW mrežnog adaptera.
- **Mrežni adapter** obezbeđuje hardversku vezu sa komunikacionim medijumom, obavlja slanje i prijem podataka u odgovarajućem formatu na osnovu definisanog seta pravila.
- **Fizički pristup medijumu** podrazumeva kabliranje, konektore, generisanje linijskih signala/kodova.

# Ethernet frejm

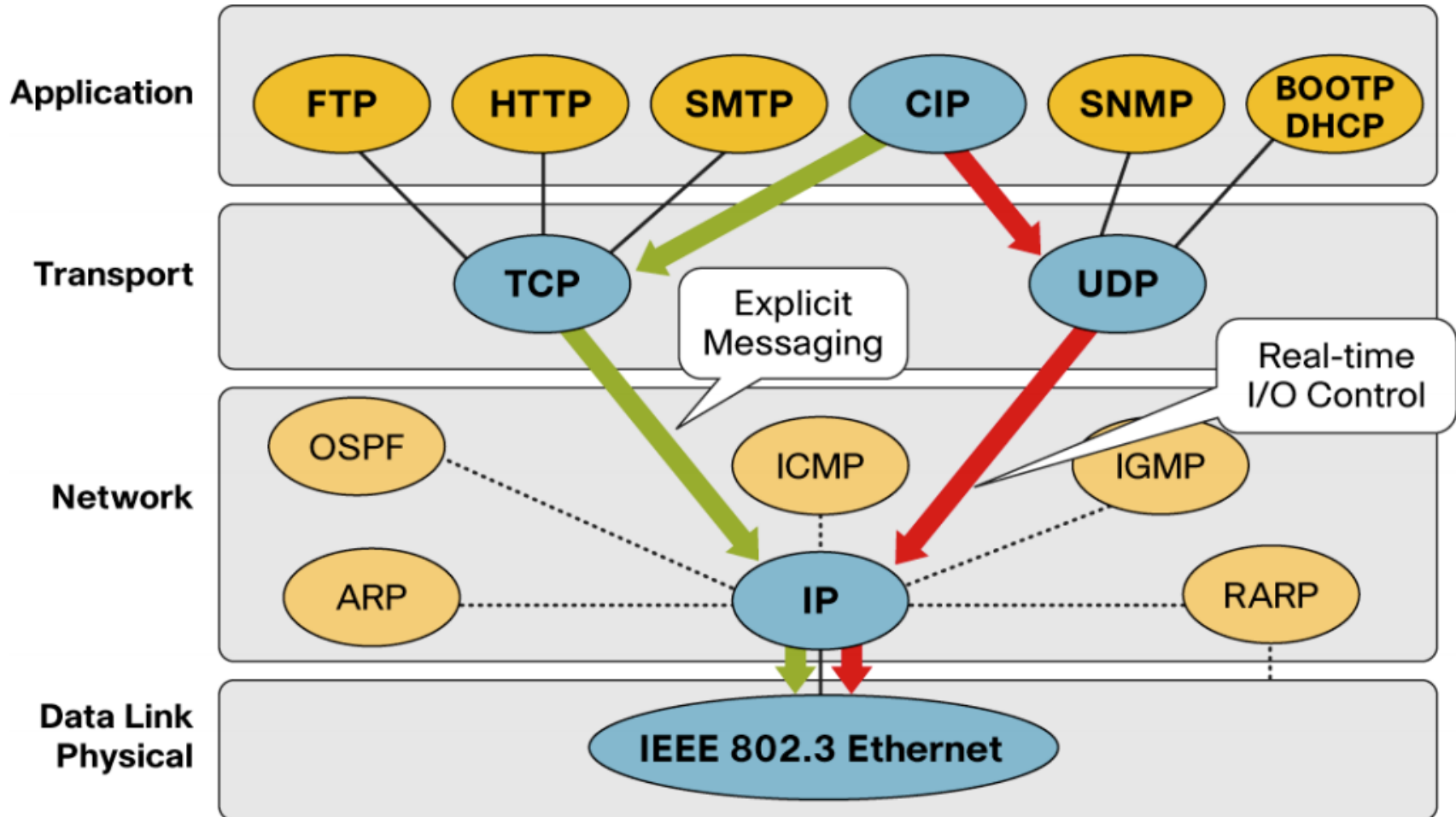
- **Preambula i SFD** imaju zajedničku funkciju da obezbede predvidivu sekvencu bita za namene sinhronizacije.
- Polje sa **MAC izvorišnom/odredišnom adresom** sadrži fizičku adresu pošiljaoca/primaoca frejma. Primaoc frejma može biti individualni intefejs, grupa interfejsa ili svi interfejsi na mreži.
- **Lenght/Type** polje može ukazivati na broj bajta validnih podataka u polju sa podacima ili na protokol koji je korišćen prilikom formatiranja polja sa podacima.
- **Polje sa podacima** sadrži izvornu informaciju dobijenu od strane protokola višeg nivoa i mora imati između 46 i 1500 bajta.
- **FCS** polje ili 32-bitni CRC omogućava prijemnom interfejsu da detektuje greške u primljenom frejmu



# Nulceo F746 Ethernet blok

- Ethernet blok sadrži 10 Mbps/100 Mbps Ethernet MAC kontroler sa DMA hardverskim ubrzanjem.
  - Pulu dupleks/pun dupleks prenos prema IEEE 802.3 std.
  - Kontrola protoka
  - Hardversko ubrzanje pokušaja retransmisije
  - Filtriranje primljenih paketa
  - Wake-up on LAN activity
  - Automatsko slanje/prijem frejmova preko DMA bez podrške CPU.
- **Osnovne karakteristike Ethernet bloka**
  - Ethernet blok je AHB master koji ima pristup RAM memoriji na čipu.
  - Veza prema ethernet PHY je obezbeđena preko RMII i MIIM (MDIO) interfejsa
  - Nezavisni predajni/prijemni baferi u SRAM
  - Podržava multicast i broadcast frejmove i proizvoljnu dužinu frejma
  - Opciono automatsko generisanje FCS polja (CRC32 kod).
  - Automatski backoff mehanizam i retransmisija frejma.

# Ethernet u službi viših protokola





# UDP protokol

- UDP- user datagram protokol – je protokol koji omogućava brzu razmenu poruka između uređaja na mreži.
- Kao i svi internet protokoli paketski je orijentisan.
- Osnovni nedostatak UDP protokola je nesigurnost prenosa podataka.
- Protokol ne daje informaciju pošiljaocu da je poruka stigla na odredište.
- Pošiljalac šalje poruku a primalac je dužan da se postara da je primi.
- Gubitak poruke nastaje najčešće usled loše tj. neblagovremene reakcije primaoca.

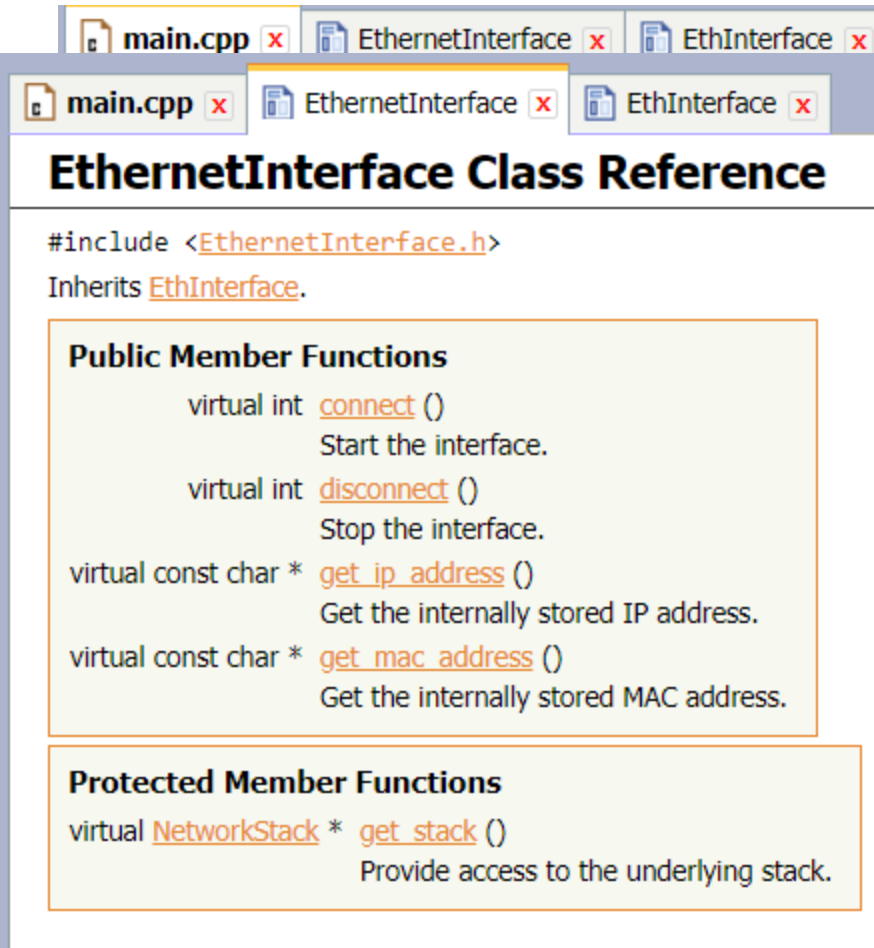
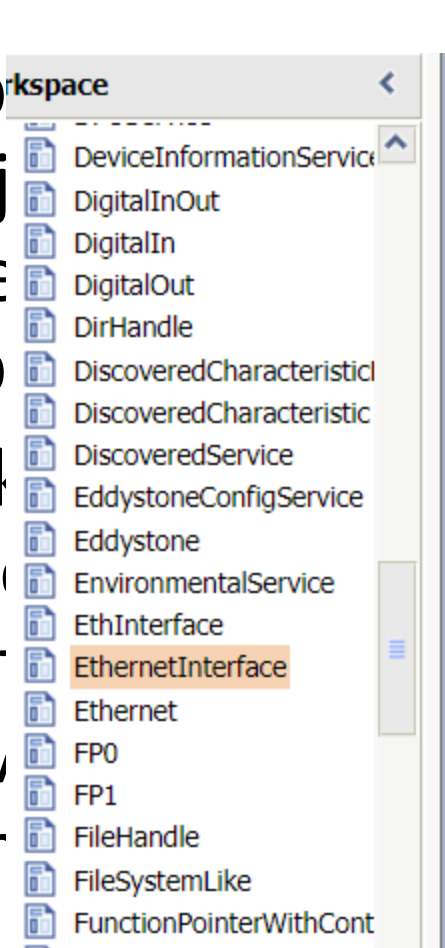
# Pirmer UDP

- Importovati projekat mbed-os-example-udp-sockets
- Preko UDP konekcije se kontaktira server koji vraća tačno vreme.

```
main.cpp x EthernetInterface x EthInterface x
13 int main() {
14     // Bring up the ethernet interface
15     printf("UDP Socket example\n");
16     if(0 != net.connect()) {
17         printf("Error connecting\n");
18         return -1;
19     }
20
21     // Show the network address
22     const char *ip = net.get_ip_address();
23     printf("IP address is: %s\n", ip ? ip : "No IP");
24
25     UDPSocket sock(&net);
26     SocketAddress sockAddr;
27
28     char out_buffer[] = "time";
29     printf("sending");
30     wait(1);
31     if(0 > sock.sendto("time.nist.gov", 37, out_buffer, s
32         printf("Error sending data\n");
33         return -1;
34     }
35     printf("sent");
36
37     ntp_packet in_data;
38     int n = sock.recvfrom(&sockAddr, &in_data, sizeof(ntp
39     in_data.secs = ntohl( in_data.secs ) - 2208988800;
40     printf("Time Received %lu seconds since 1/01/1900 00:
```

# Pirmer UDP

- Imp
- proj
- os-€
- udp
- Pre
- kon
- kon
- serv
- tačr



```
36  
37     ntp_packet in_data;  
38     int n = sock.recvfrom(&sockAddr, &in_data, sizeof(ntp  
39     in_data.secs = ntohs( in_data.secs ) - 2208988800;  
40     printf("Time Received %lu seconds since 1/01/1900 00:
```

orkspace

- SafeBool
- Semaphore
- SerialBase
- Serial
- SocketAddress
- Socket
- Specification
- Stream
- String
- TCPService
- TCPService
- Thread
- Ticker
- Timeout
- TimerEvent
- Timer
- Transaction
- UARTService
- UDPSocket
- URIBeaconConfigService
- UUID
- WiFiInterface
- WriteOnlyArrayGattCharacteris
- WriteOnlyGattCharacteris
- base
- iBeacon
- init()

main.cpp x EthernetInterface x EthInterface x UDPSocket x

## UDPSocket Class Reference

```
#include <UDPSocket.h>
```

Inherits [Socket](#).

### Public Member Functions

[UDPSocket](#) ()

Create an uninitialized socket.

template<typename S >

[UDPSocket](#) (S \*stack)

Create a socket on a network interface.

virtual [~UDPSocket](#) ()

Destroy a socket.

int [sendto](#) (const char \*host, uint16\_t port, const void \*data, unsigned size) ); IP");

Send a packet over a UDP socket.

int [sendto](#) (const [SocketAddress](#) &address, const void \*data, unsigned size)

Send a packet over a UDP socket.

int [recvfrom](#) ([SocketAddress](#) \*address, void \*data, unsigned size)

Receive a packet over a UDP socket.

int [open](#) ([NetworkStack](#) \*stack)

Opens a socket.

int [close](#) ()

Close the socket.

int [bind](#) (uint16\_t port)

Bind a specific address to a socket.

int [bind](#) (const char \*address, uint16\_t port)

Bind a specific address to a socket.

```
39     in_data.secs = ntohl( in_data.secs ) - 2208988800;  
40     printf("Time Received %lu seconds since 1/01/1900 00:");  
                                     sizeof(ntp
```

# TCP protokol

- Napredniji i komplikovaniji protokol od UDP protokola.
- Osnovna prednost u odnosu na UDP protokol je sigurnost u prenosu informacija.
- U samom protokolu je inherentno ugrađena kontrola prenosa podataka.
- Pošiljalac može da šalje podatke jedino ako postoji sigurna veza sa primaocem.
- Prekid veze se jasno vidi i signalizira tako da se ne može desiti da pošiljalac šalje podatke u “prazno”.
- Cena koja se plaća za ovo je sporiji protokol jer se u pozadini prenosi znatno više informacija i to u oba smera.

# Primeri programa koji komuniciraju preko TCP protokola











- [https://os.mbed.com/users/nenad/code/mbed-os-PMK2018\\_tcp\\_client/](https://os.mbed.com/users/nenad/code/mbed-os-PMK2018_tcp_client/)
- [https://os.mbed.com/users/nenad/code/mbed-os-PMK2018\\_tcp\\_server/](https://os.mbed.com/users/nenad/code/mbed-os-PMK2018_tcp_server/)

# UDP ili TCP

- Dobar primer za prednosti UDP nad TCP protokolom je situacija kada je komunikacija pretežno jednosmerna i sa velikom količinom podataka.
- Kod UDP protokola primalac jednostavno šalje podatke jedan za drugim. Ako je arhitektura sistema dobro osmišljena i poznata i ako je primalac provereno sposoban da primi veliki protok podataka od pošiljaoca onda neće biti problema. UDP protokol se na primer koristi pri prenosu video signala.

# Viši protokoli

- <http://mbed.org/handbook/TCP-IP-protocols-and-APIs>

Protocol	Library	Example Application	Documentation	Author
<a href="#">HTTP Client</a>	<a href="#">HTTPClient</a>	<a href="#">HTTPClient_HelloWorld</a>	<a href="#">Documentation</a>	 <a href="#">Donatien Garnier</a>
<a href="#">HTTP Client</a>	<a href="#">TinyHTTP_b</a>	<a href="#">denki-yohou_b</a>	<a href="#">Documentation</a>	 <a href="#">Suga koubou</a>
<a href="#">NTP Client</a>	<a href="#">NTPClient</a>	<a href="#">NTPClient_HelloWorld</a>	<a href="#">Documentation</a>	 <a href="#">Donatien Garnier</a>
<a href="#">Websocket Client</a>	<a href="#">WebSocketClient</a>	<a href="#">WebSocket_HelloWorld</a>	<a href="#">Documentation</a>	 <a href="#">Samuel Mokrani</a>
<a href="#">HTTP Server</a>	<a href="#">RPC over HTTP Server</a>	<a href="#">RPC over HTTP Server</a>	<a href="#">Documentation</a>	 <a href="#">François Berder</a>
<a href="#">HTTP Server</a>	<a href="#">HTTPServer_echoback</a>	<a href="#">HTTPServer_echoback</a>	<a href="#">Documentation</a>	 <a href="#">Takuya Urakawa</a>
<a href="#">SMTP Client</a>	<a href="#">SimpleSMTPClient</a>	<a href="#">SimpleSMTPClient_HelloWorld</a>	<a href="#">Documentation</a>	 <a href="#">Tadao Iida</a>
<a href="#">Twitter API</a>	<a href="#">HTTPClient_Twitter</a>	<a href="#">HTTPClient_Twitter</a>	<a href="#">Documentation</a>	 <a href="#">Ryan Pannell</a>
<a href="#">IEEE1888(FIAP)</a>	<a href="#">FiapV2</a>	<a href="#">temp_FIAP</a>	<a href="#">Documentation</a>	 <a href="#">Yasushi TAUCHI</a>
<a href="#">Evrythng Api</a>	<a href="#">EvrythngApi</a>	<a href="#">EvrythngApiExample</a>	<a href="#">Documentation</a>	 <a href="#">Evry Thng</a>



# UMTS

<http://mbed.org/cookbook/VodafoneUSBModem>

- UMTS je fizički nivo baziran na upotrebi naprednih tehnika prenosa u GSM tehnologiji.
- Danas se koristi kao osnova za mobilni internet, pa samim tim podržava sve rasprostranjene internet protokole.
- Kada se jednom uspostavi veza UMTS prenosni put se ne razlikuje od ethernet prenosnog puta.
- UMTS je zgodan za razne vrste bežičnih senzora.

