

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1643

**WEB POSLUŽITELJ ZA UDALJENO  
PRAĆENJE TEMELJEN NA UGRADBENOM  
RAČUNALU**

Marko Ščrbak

Zagreb, siječanj 2011.



## SADRŽAJ

<b>UVOD</b> .....	<b>5</b>
<b>STELLARIS LM3S6965 EVALUATION KIT</b> .....	<b>6</b>
<b>STELLARIS LM3S6965 RAZVOJNA PLOČICA</b> .....	<b>7</b>
STELLARIS LM3S6965 MIKROKONTROLER.....	9
OLED GRAFIČKI ZASLON.....	11
USB KONTROLER .....	11
ZVUČNIK.....	12
UTOR ZA MICRO - SD KARTICU .....	12
TIPKE .....	12
LED DIODE .....	12
<b>KEIL RealView MICROKONTROLLER DEVELOPMENT KIT</b> .....	<b>13</b>
<b>(MDK – ARM)</b> .....	<b>13</b>
<b>VIZUALIZACIJA PROJEKTA</b> .....	<b>18</b>
<b>DIJELOVI PROJEKTA I NJIHOVE FUNKCIONALNOSTI</b> .....	<b>18</b>
OLED ZASLON .....	18
TIPKE ZA NAVIGACIJU .....	20
KOMUNIKACIJA SA RAČUNALOM .....	21
MIKRO SD – KARTICA.....	25
ETHERNET KONTROLER I WEB POSLUŽITELJ.....	27
<b>TIJEK IZVOĐENJA PROGRAMA</b> .....	<b>30</b>
<b>ZAKLJUČAK</b> .....	<b>32</b>
<b>LITERATURA</b> .....	<b>33</b>

<b>NASLOV .....</b>	<b>35</b>
<b>KLJUČNE RIJEČI.....</b>	<b>35</b>
<b>SAŽETAK .....</b>	<b>35</b>
<b>TITLE .....</b>	<b>36</b>
<b>KEYWORDS .....</b>	<b>36</b>
<b>SUMMARY .....</b>	<b>36</b>

## UVOD

U okviru ovog završnog rada izradit ćemo jednostavan ugradbeni računalni sustav za udaljeno praćenje koji će svoje očitane vrijednosti spremati na SD – karticu i putem jednostavne web stranice prezentirati svijetu. Na sustavu će biti implementiran i jednostavan web poslužitelj koji će web stranicu prezentirati klijentu.

Za potrebe rada koristit će se jedan novi sustav. Radi se o Stellaris LM3S6965 razvojnom sustavu. Sustav je realiziran na maloj štampanoj pločici i dolazi u paketu sa svom potrebnom dokumentacijom i softwareom te uz neke primjere omogućava brz razvoj i implementaciju novih i gotovih aplikacija.

## STELLARIS LM3S6965 EVALUATION KIT

Za potrebe završnog rada korišten je Stellarisov evaluation kit, koji sadrži sve dijelove potrebne za realizaciju projekta, razvoj aplikacije te njezino pokretanje na Stellarisovom mikrokontroleru. Stellaris LM3S6965 evaluation kit sadrži slijedeće:

- LM3S6965 razvojnu pločicu(EVB – avaluation board)
- USB kabel
- 20 – pinski JTAG/SWD target kabel
- CD koji sadrži:
  - Keil RealView Microcontroller Development Kit (MDK-ARM)
  - Svu potrebnu dokumentaciju
  - Quickstart vodič i izvorni kod
  - Stellaris Firmware razvojni paket koji sadrži primjere i njihove izvorne kodove
- CAT5 kabel (Ethernet kabel)



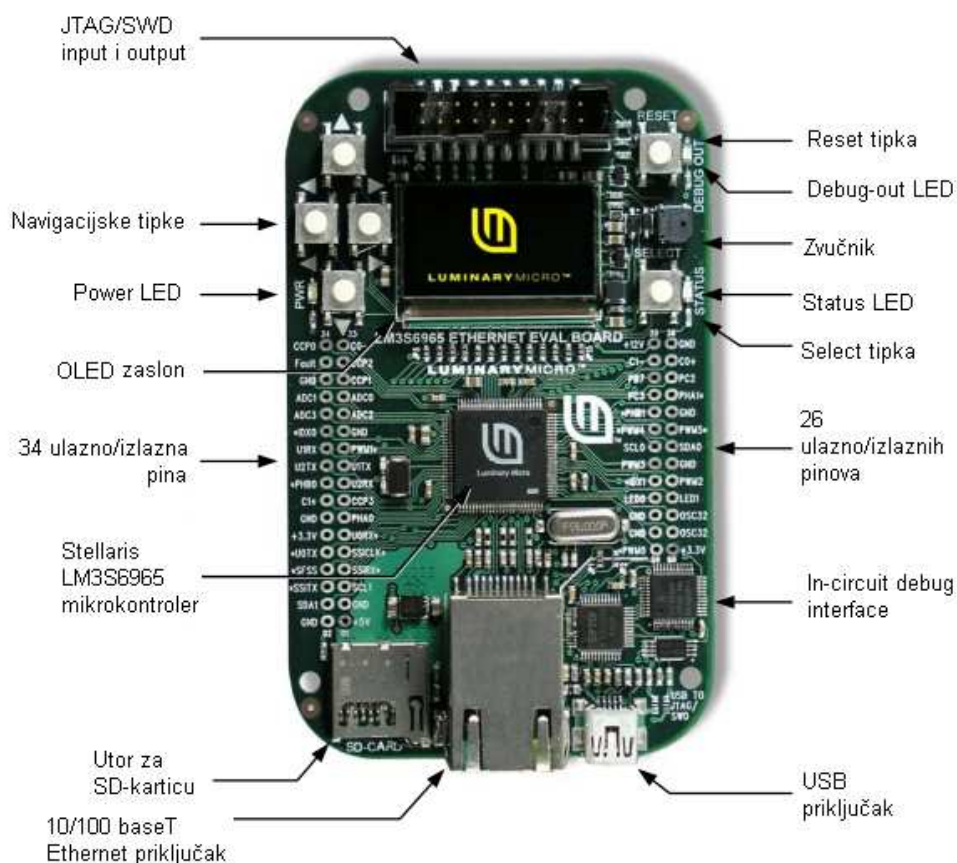
Slika 1: Stellaris LM3S6965 Evaluation Kit

## STELLARIS LM3S6965 RAZVOJNA PLOČICA

Stellaris LM3S6965 razvojna pločica je kompaktna i prilagodljiva razvojna platforma zasnovana na Stellaris LM3S6965 ARM® Cortex™- M3 mikrokontroleru. Razvojni sustav između ostalog sadrži i 10/100 Ethernet kontroler koji je u potpunosti integriran sa LM3S6965 mikrokontrolerom te kao takav služi za demonstraciju WEB servera na ugradbenom računalu. Sama pločica može poslužiti kao razvojna platforma ili kao jeftin ICDI – *in-circuit debug interface*.

Razvojni sustav i njegova programska podrška omogućavaju brz razvoj prototipa i izradu aplikacijsko orijentiranog dizajna za Ethernet mreže. Programska podrška uključuje i velik broj već gotovih biblioteka i primjera sa izvornim kodovima, omogućavajući tako brzu izradu aplikacija pisanih u programskom jeziku

C.

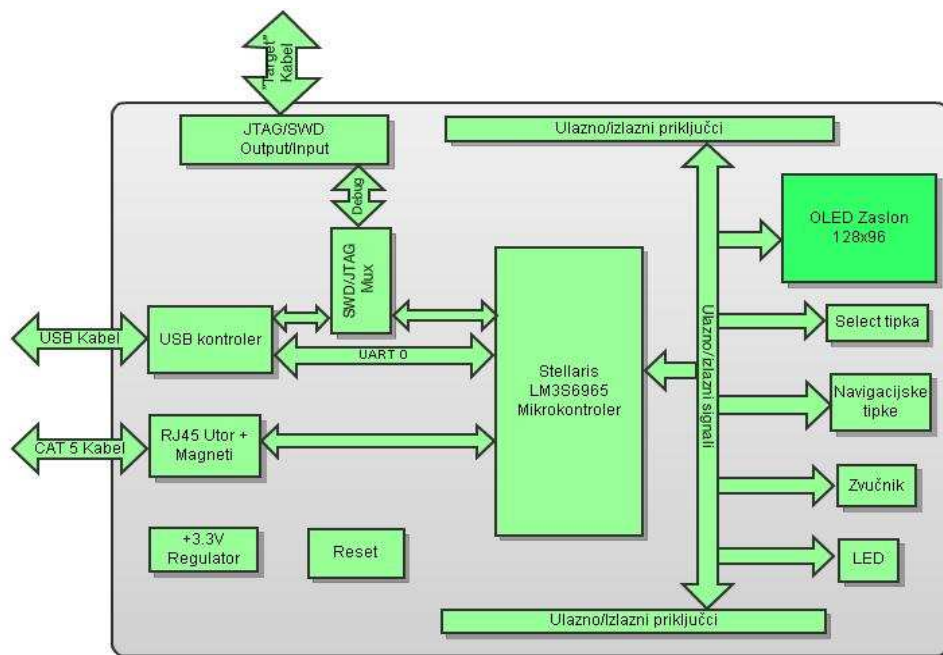


Slika 2: Stellaris LM3S6965 Razvojna pločica

Stellaris LM3S6965 razvojna pločica sadrži slijedeće značajke:

- Stellaris LM3S6965 mikrokontroler sa potpuno integriranim 10/100 ugradbenim Ethernet kontrolerom
- USB priključak; USB kabel pruža serijsku komunikaciju, potrebnu struju te mogućnost debugiranja
- OLED grafički zaslon rezolucije 128x96 piksela
- Korisničke LED diode, 4 navigacijske tipke, select tipku, te tipku za reset
- Zvučnik
- Označene ulazno/izlazne priključke
- Utor za micro-SD karticu
- Standardni ARM 20-pinski JTAG debug konektor sa ulaznim i izlaznim modom

Sve navedene značajke i njihov raspored prikazane su slijedećim dijagramom:



Slika 3: Raspored elemenata na pločici



## **STELLARIS LM3S6965 MIKROKONTROLER**

Srcce Stellarisove razvojne pločice je Stellaris LM3S6965 mikrokontroler. Familija mikrokontrolera Stellaris je prva familija koja se temelji na ARM® Cortex™- M3 jezgri, te kao takva donosi visoke performanse 32-bitnog ugradbenog računala za cijenu ekvivalentnim 8-bitnim i 16-bitnim mikrokontrolerima. Stellaris familija mikrokontrolera nudi učinkovite performanse i visok stupanj integracije čineći tu familiju prikladnom za razvoj aplikacija koje zahtijevaju značajnu kontrolu obrade i sposobnost povezivanja i sve to uz relativno nisku cijenu. Stellaris LM3S6000 serija po prvi put kombinira 10/100 mrežni (EthernetMedia Access Control - MAC) i fizički (Physical - PHY) sloj integriran u jednoj arhitekturi ARM mikrokontrolera.

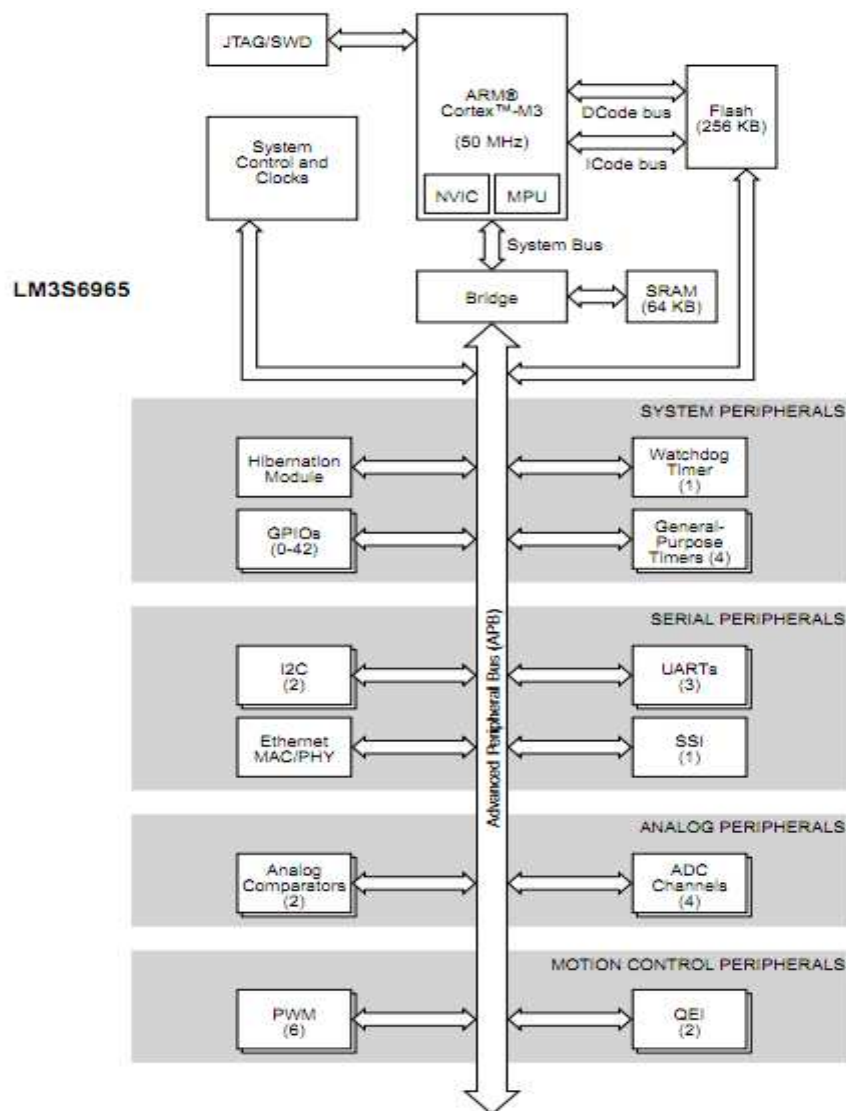
LM3S6965 mikrokontroler primarno je namjenjen za industrijske primjene, uključujući daljinsko praćenje, testiranje i mjernu opremu, rad sa uređajima u mreži, automatizaciju, kontrolu pokreta, medicinske instrumente i sigurnost, itd.

Za potrebe aplikacija koje zahtijevaju nisku potrošnju i visoku uštedu energije, LM3S6965 raspolaže i sa modulom za hibernaciju sa baterijom omogućavajući prelazak u režim smanjene potrošnje za vrijeme kada je mikrokontroler neaktivan. Modul za hibernaciju uz power-up/power-down sekvencer, brojač u stvarnom vremenu (RealTimeCounter – RTC), nekoliko priručnih registara, APB sučelje prema sistemskoj sabirnici, te uz rezerviranu memoriju, čini LM3S6965 prikladnim i za baterijom pogonjene aplikacije.

Mikrokontroler LM3S6965 sadrži slijedeće značajke i komponente:

- 32-bitnu RISC performansu
- ARM Cortex – M3 procesorsku jezgru
- JTAG
- Modul za hibernaciju
- Unutarjna memorija
  - 256KB Flash-a
  - 64 KB SRAM-a
- GPIO
  - 0-42 ulazno izlazna priključka opće namjene

- 4 Vremenska brojača opće namjene
- Integrirani Ethernet MAC i PHY
- 3 programabilna UART priključka
- Watchdog timer
- 4 analogna ulazna kanala
- 2 integrirana analogna komparatora
- 2 I<sup>2</sup>C modula
- SSI (sinkronizirano serijsko sučelje)
- 3 bloka pulsno širinskih modulatora (PWM)
- 2 QEI modula
- On-chip low drop-out (LDO) regulator napona



Slika 4: Blok dijagram mikrokontrolera LM3S6965

Mikrokontroler dolazi sa tvornički ugrađenom quick-start aplikacijom koja se nalazi u flash memoriji mikrokontrolera i pokreće se čim se pločica spoji USB kabelom sa računalom, osim ako smo flash memoriju prebrisali sa vlastitim korisničkim programom.

## **OLED GRAFIČKI ZASLON**

Na razvojnoj pločici nalazi se i OLED grafički zaslon (Organic LED display) za prikaz podataka, rezolucije 128x96 piksela. OLED je nova tehnologija koja pruža puno više mogućnosti od običnog LCD zaslona.

Karakteristike OLED zaslona:

- RiT P14201 display
- 128x96 piksela (128 stupaca, 96 redova)
- Visoki kontrast (500:1)
- Odličan sjaj(svjetlost) – 120cd/m<sup>2</sup>
- Brzi odziv (10 us)

OLED zaslon ima ugrađen kontroler sa sinkronim serijskim i paralelnim sučeljem. Na razvojnoj pločici koristi se SSI iz razloga što koristi manje mikrokontrolerskih pinova. Sa zaslona se ne mogu čitati podaci, samo pisati. OLED zaslon dijeli SSI port sa SD – karticom. Stellaris Firmware razvojni paket nudi sve potrebne drivere za OLED zaslon kao i njihove izvorne kodove. Životni vijek OLED zaslona je preko 13000 sati.

## **USB KONTROLER**

USB kontrolerom spajamo sa računalom USB kabelom, te tako pločici osiguravamo potreban napon i struju. Osim toga pomoću USB – a, računalu sa Windows operacijskim sustavom, omogućena je komunikacija pomoću aplikacija

poput HyperTerminala preko virtualnog COM porta sa UART0 priključkom same razvojne pločice. Nakon instalacije svih potrebnih drivera, operacijski sustav Windows dodjeljuje COM port VCP kanalu. USB uređaj provodi i JTAG/SWD operacije pod nadzorom debuggera. Pomoću dva kanala omogućeno je da imamo dvije simultane komunikacijske veze između računala i razvojne pločice koristeći samo jedan USB kabel.

## **ZVUČNIK**

Mali magnetski transduktor spojen preko MOSFET-a na PD1/PWM, omogućavajući nekoliko načina za generiranje jednostavnih i kompleksnih tonova.

## **UTOR ZA MICRO - SD KARTICU**

SD-kartice idealne su za prijenos podataka i za njihovu pohranu, kao što su WEB – stranice, log file-ovi ili neki drugi podaci. Primjeri čitanja i file-system-a nalaze se u primjerima u Stellaris Firmware paketu te su korišteni i u ovom projektu.

## **TIPKE**

Četiri programirane navigacijske tipke, select tipka te reset tipka omogućavaju jednostavnu komunikaciju sa mikrokontrolerom.

## **LED DIODE**

Power-on LED dioda, LED diode na Ethernet kontroleru te korisnička LED dioda omogućavaju raznoliko signaliziranje korisniku.

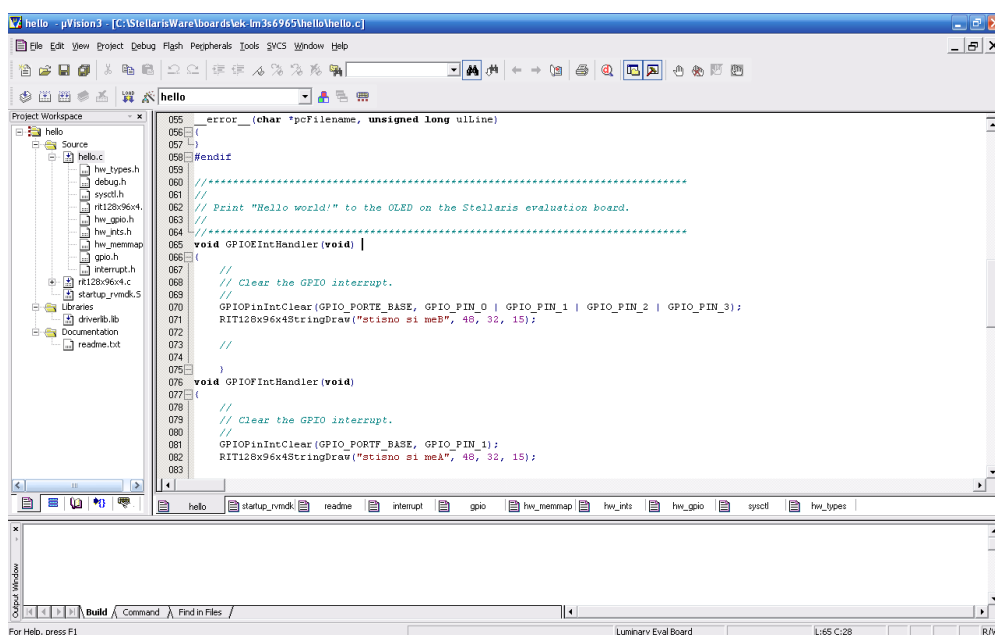
Većina periferije je lako programirajiva od strane korisnika, dok druge periferije imaju već gotove drivere i primjere kako ih koristiti. Stellaris Firmware razvojni paket sadrži sve drivere i primjere pomoću kojih se brzo i lako mogu napraviti gotove aplikacije za rad sa podacima na SD – kartici, mrežom, praćenjem, itd. ARM kao i Texas Instruments nudi cjelovita rješenja za brz proboj na tržište, mnoge gotove biblioteke za lakše korištenje i programiranje te svu odgovarajuću programsku podršku.

# KEIL RealView MICROKONTROLLER DEVELOPMENT KIT (MDK – ARM)

Razvojni alat koji je korišten za ovaj završni rad u vlasništvu je tvrtke ARM i samo je jedan od nekoliko alata pomoću kojih možemo izgraditi valstiti projekt za ciljanu razvojnu pločicu. Za potrebe rada odabran je ovaj alat iz razloga što dolazi u paketu sa razvojnom pločicom, a i jednostavan je za korištenje i primjenjen početnicima.

Keil RealView MDK – ARM instalacija vrši se pomoću installera koji se nalazi na CD-u sa dokumentacijom. Nakon završetka instalacije potrebno je instalirati sve potrebne drivere kao i Stellaris Firmware razvojni paket. Sve to potrebno je za realizaciju projekta čiji će se izvorni kod pisati u Keil uVision razvojnoj okolini

Keil uVision je razvojna okolina za pisanje izvornog koda sa funkcijama prevođenja istog, praćenje, debugiranje te mnogim drugim korisnim funkcijama za realizaciju projekta. Okolina se sastoji od mnoštva prozora, ikona za prečice, alatnih traka te padajućih izbornika.

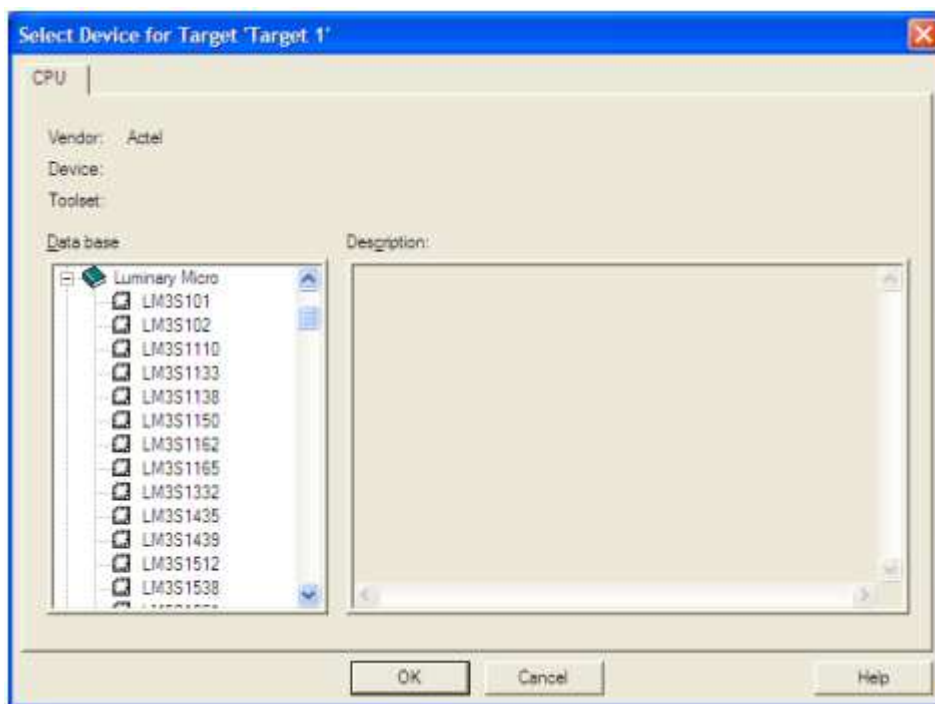


Slika 5: Okolina Keil uVision

Rad u uVisionu započinje posljednim projektom koji je bio pokrenut. Novi projekt se pokreće tako da se u izborniku odabere „Project“ i stisne se na opciju „New uVision Project“. Potom se otvara dijaloški okvir u kojem odlučujemo kako ćemo nazvati naš projekt i gdje ćemo ga spremiti. Ovaj projekt nazvan je LM3S6965\_zavrzni\_rad.Uv2 i nalazi se u:

```
C:\StellarisWare\boards\LM3S6965_zvrzni_rad.Uv2
```

Nakon što smo sačuvali projekt potrebno je odabrati na kojem uređaju radimo. U našem slučaju to je LM3S6965 iz porodice Luminary Micro.



Slika 6: Odabir uređaja

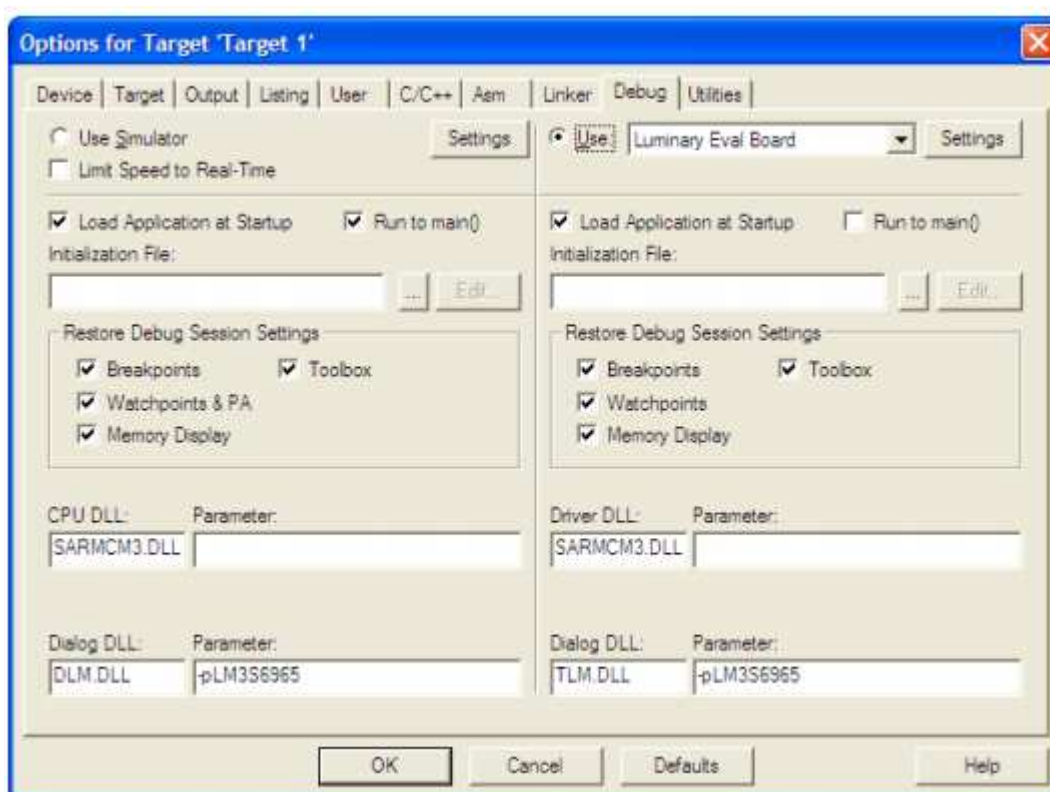
Na kraju još moramo odabrati želimo li Startup kod za naš uređaj. U našem slučaju ćemo odabrati da želimo pošto je on prijeko potreban za rad projekta.



Slika 7: Generiranje startup koda

Prije početka rada potrebno je podesiti i opcije hardware-a na kojem radimo. Opcije podešavamo tako da odaberemo „Project“ iz glavnog izbornika i odaberemo „Options for Target – Završni Rad“. U dijaloškom okviru potrebno je za svaku karticu unijeti postavke koje odgovaraju našem hardware-u i načinu rada.

Za početak je potrebno u kartici Target podesiti frekvenciju Xtal na 8.0 Mhz i provjeriti odgovaraju li veličine memorije. Nakon toga potrebno je promijeniti način debugiranja u kartici debug. Normalno je da koristimo simulator međutim možemo i koristiti pravi hardware pa odaberemo u padajućem izborniku Luminary Eval Board.



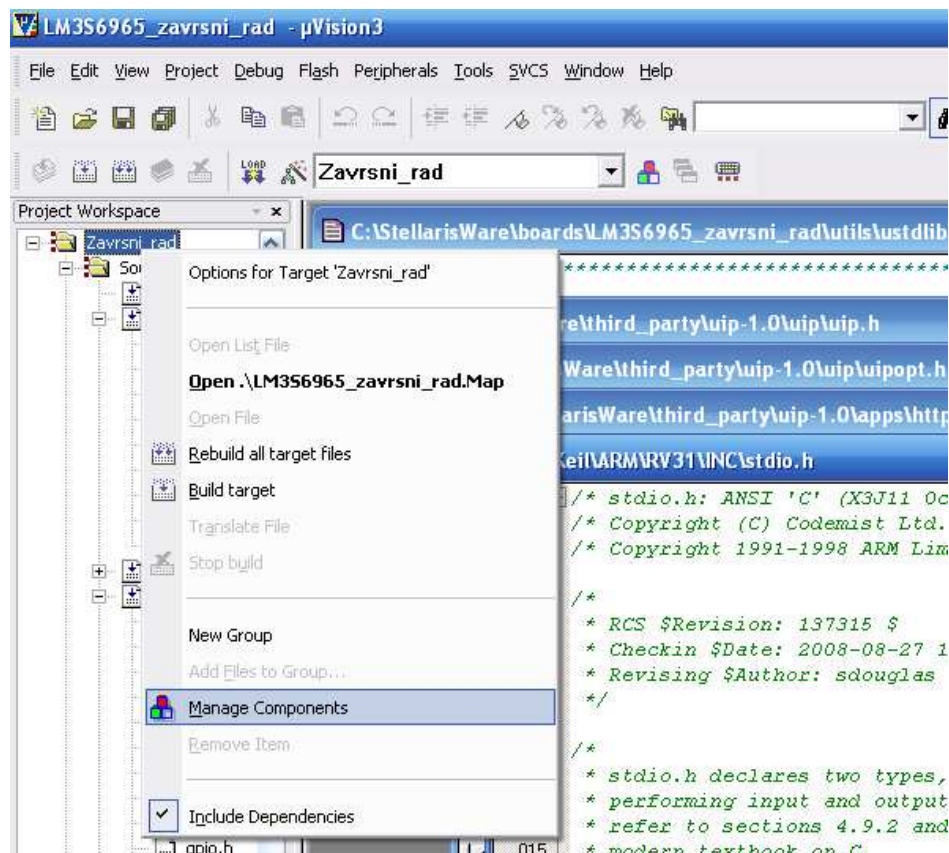
Slika 8: Podešavanje parametara za rad

Kao što smo promijenili hardware za debug, tako moramo isto to napraviti za debug sučelje za programiranje flash memorije. U kartici Utilities odaberemo „use target driver for flash programming“, ta u padajućem izborniku odaberemo našu razvojnu pločicu Luminary Eval Board. Sad ćemo moći spremiti aplikaciju na flash i uspješno ju debugirati.

Još u C/C++ kartici moramo podesiti da koristimo ARM kompajler, pa ćemo staviti „rvmdk PART\_LM3S6965“ pod „preprocessor symbols“. Ostale postavke ostavimo kao „default“ ili ih promjenimo prema potrebi.

Jedino što preostaje je napraviti projekt počevši od nule ili koristeći gotove komponente. Za potrebe rade korištene su komponente i primjeri koji su sadržani unutar Stellaris Firmware paketa, te su dodavane nove funkcije kako bi se postigle željene funkcionalnosti aplikacije.

Za početak potrebno je dodati datoteku „driverlib.lib“ u projekt kako bi aplikaciji dali sve potrebne drivere za vrijeme kompajliranja/linkanja. Kako bi koristili u stvarnosti određene funkcije u našoj aplikaciji potrebno je uključiti i zaglavlja(header files) u source datoteku u kojoj koristimo funkciju pojedinog drivera. Dodavanje izvornih datoteka i drugih datoteka vrši se desnim klikom na ime projekta te odabiranjem funkcije „manage components“.



Slika 9: Dodavanje komponenti u projekt

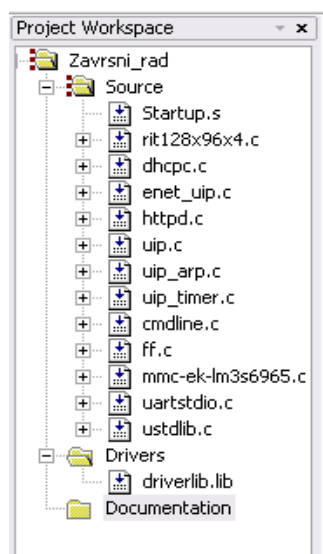


Unutar dijaloškog okvira možemo dodati i mijenjati grupe, te dodavati nove datoteke koje sadrže izvorni kod. Unutar projekta dodane su slijedeće datoteke unutar grupe source:

- rit128x96x4.c
- dhcp.c
- enet\_uip.c
- httpd.c
- uip.c
- uip\_arp.c
- uip\_timer.c
- cmdline.c
- ff.c
- mmc-ek-lm3s6965.c
- uartstdio.c
- ustdlib.c

Pojedinosti oko pojedine source datoteke bit će objašnjene kasnije.

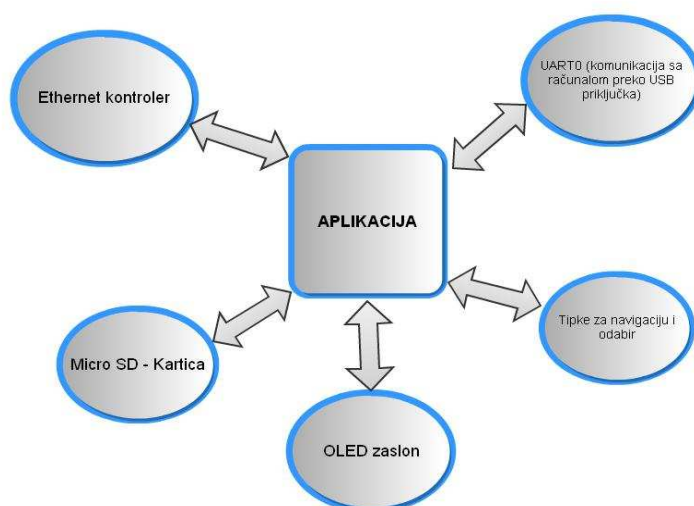
Unutar grupe Drivers smještena je datoteka driverlib.lib, te je dodana još jedna grupa ukoliko se odluči unutra staviti dokumentacija projekta.



Slika 10: Komponente projekta

## VIZUALIZACIJA PROJEKTA

Kako bi lakše pokazali na koji način će biti izvedena aplikacija koja će se vrtiti na mikrokontroleru vizualizirat ćemo dijelove koji će se povezati u jednu cjelinu. Aplikacija mora povezivati nekoliko cjelina, a svaka od cjelina bit će predstavljena svojim izvornim kodovima, a sve cjeline bit će povezane u jednu.



Slika 11: Dijagram komponenti projekta

## DIJELOVI PROJEKTA I NJIHOVE FUNKCIONALNOSTI

### **OLED ZASLON**

Ranije je već opisano da razvojna pločica raspolaže OLED zaslonom rezolucije 128x96 piksela. Unutar ovog projekta zaslon će služiti za prikaz uspostavljana veze, IP adrese, izbornika, te prikaz nekih statističkih vrijednosti. Kako bi se omogućio rad sa zaslonom potrebno je u projekt uključiti datoteku rit128x96x4.c koja predstavlja driver za naš zaslon. Kako bi se u pojedinim .c datotekama koristile funkcije pomoću koji obavljam inicijalizaciju zaslona, te ispis

i slične radnje, potrebno je u svaki od tih izvornih datoteka uključiti header file `rit128x96x4.h` koji sadrži sve potrebne deklaracije funkcija.

Neke od funkcija koje se najčešće koriste su:

- `extern void RIT128x96x4Clear(void);`
  - funkcija za brisanje zaslona
- `extern void RIT128x96x4StringDraw(const char*pcStr, unsigned long ulX, unsigned long ulY, unsigned char ucLevel);`
  - funkcija za ispis stringa na zaslon na poziciju `x(stupac)`, `y(red)`, te razine osvijetljenosti
- `extern void RIT128x96x4Init(unsigned long ulFrequency);`
  - inicijalizacija zaslona
- `extern void RIT128x96x4Enable(unsigned long ulFrequency);`
  - enable funkcija
- `extern void RIT128x96x4Disable(void);`
  - disable funkcija
- `extern void RIT128x96x4DisplayOn(void);`
  - uključi zaslon
- `extern void RIT128x96x4DisplayOff(void);`
  - isključi zaslon

Postoji još i funkcija za prikaz slike na zaslon, no ona nije korištena unutar ovog projekta.

Potrebno je uočiti da se jedino podatak tipa `string` (niz `charactera`) može ispisati na zaslon pa su korištene funkcije koje su varijable tipa `int` pretvarale u podatak tipa `character`.

Također potrebno je paziti da se zaslon onemogućí(disable-a) kada nešto čitamo i pišemo na SD – karticu zato što OLED zaslon i priključak za mikro SD – karticu imau dijeljenu serijsku (SSI) sabirnicu.

## **TIPKE ZA NAVIGACIJU**

Tipke za navigaciju spojene su na GPIO priključke mikrokontrolera i to na slijedeći način.

- Navigacijska tipka GORE – priključak PE0
- Navigacijska tipka DOLJE – priključak PE1
- Navigacijska tipka LIJEVO – priključak PE2
- Navigacijska tipka DESNO – priključak PE3
- Tipka za odabir SELECT – priključak PF1

Kako bi omogućili funkcionalnost pojedinih tipki potrebno je u izvornu datoteku uključiti header fileove koje omogućavaju rad sa GPIO. Nakon toga potrebno je za svaki skup priključaka, u našem slučaju to su skupovi E i F, potrebno je napisati odgovarajuću prekidnu rutinu. Napisanu prekidnu rutinu potrebno je deklarirati unutar Startup koda kao rutinu pisanu u vanjskoj .c datoteci te ju navesti u tablici vektora prekidnih rutina. Na primjeru je pokazano kako je to izvedeno za prekidnu rutinu koja obrađuje prekid pristigao sa skupa priključaka F.

```
void GPIOFIntHandler(void)
{
    // Clear the GPIO interrupt.
    GPIOPinIntClear(GPIO_PORTF_BASE, GPIO_PIN_1);
    // Posao koji obavlja prekidna rutina
    ...
}
```

Prekidna rutina za navigacijske tipke obrađuje prekid tako da ustvrdi koja tipka je pritisnuta te ovisno o tome obavlja traženu funkciju. Sve prekidne rutine moraju se navedene su u tablici prekidnih vektora u Startup kodu.

Primjer deklariranja prekidnih rutina u tablici prekidnih vektora:

...

```
DCD      IntDefaultHandler          ; GPIO Port D
DCD      GPIOEIntHandler            ; GPIO Port E
DCD      UARTIntHandler             ; UART0 Rx and Tx
DCD      IntDefaultHandler          ; UART1 Rx and Tx
```

...

Primjer deklariranja prekidnih funkcija unutar Startup koda:

...

```
EXTERN  UARTIntHandler
EXTERN  GPIOEIntHandler
EXTERN  GPIOFIntHandler
```

...

Unutar glavnog programa potrebno je najaviti rad sa GPIO priključcima, inicijalizirati ih te omogućiti GPIO prekide.

## **KOMUNIKACIJA SA RAČUNALOM**

Komunikacija sa računalom odvija se preko priključka UART0. Priključak je spojen sa USB kontrolerom te se komunikacija sa računalom odvija preko USB kabela. Računalo „vidi“ pločicu spojenu ustvari preko virtualnog COM porta. Prilikom spajanja pločice sa računalom, operacijski sustav Windows dodjeljuje pločici virtualni COM port. Komunikaciju preko virtualnog COM porta možemo pratiti pomoću programa HyperTerminal.

Za rad sa UART-om potrebno je u projekt uključiti sve potrebne source datoteke. U ovom slučaju dodana je datoteka uartstdio.c čije funkcije koristimo u

glavnom programu. Također je potrebno uključiti i sve potrebne header fileove od kojih je najvažniji `uart.h`. Unutar glavnog programa potrebno je omogućiti rad sa UART periferijom te inicijalizirati UART i njegove prekide. Sva komunikacija sa vanjskim svijetom odvija se preko UART0 priključka.

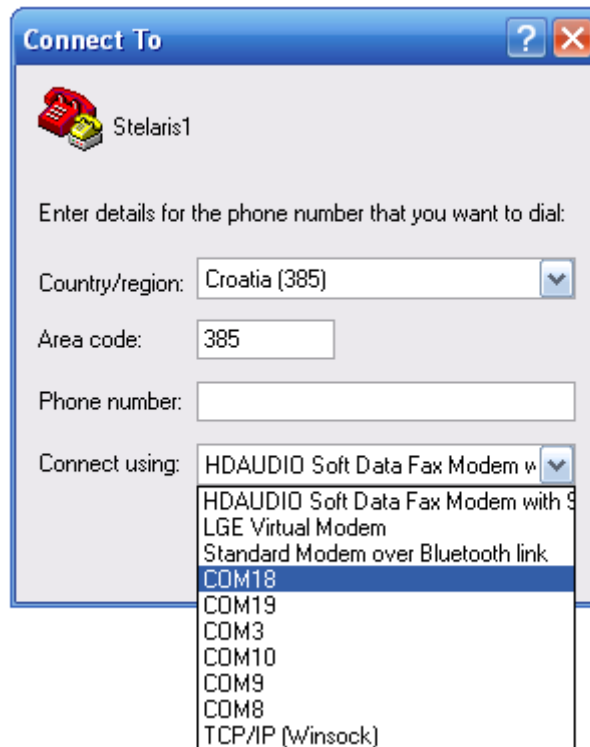
Primjer inicijalizacije UART-a:

```
SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);
UARTStdioInit(0);
// Configure the UART for 115,200, 8-N-1 operation.
UARTConfigSetExpClk(UART0_BASE, SysCtlClockGet(), 115200,
                    (UART_CONFIG_WLEN_8 | UART_CONFIG_STOP_ONE |
                     UART_CONFIG_PAR_NONE));
// Enable the UART interrupt.
IntEnable(INT_UART0);
UARTIntEnable(UART0_BASE, UART_INT_RX | UART_INT_RT);
```

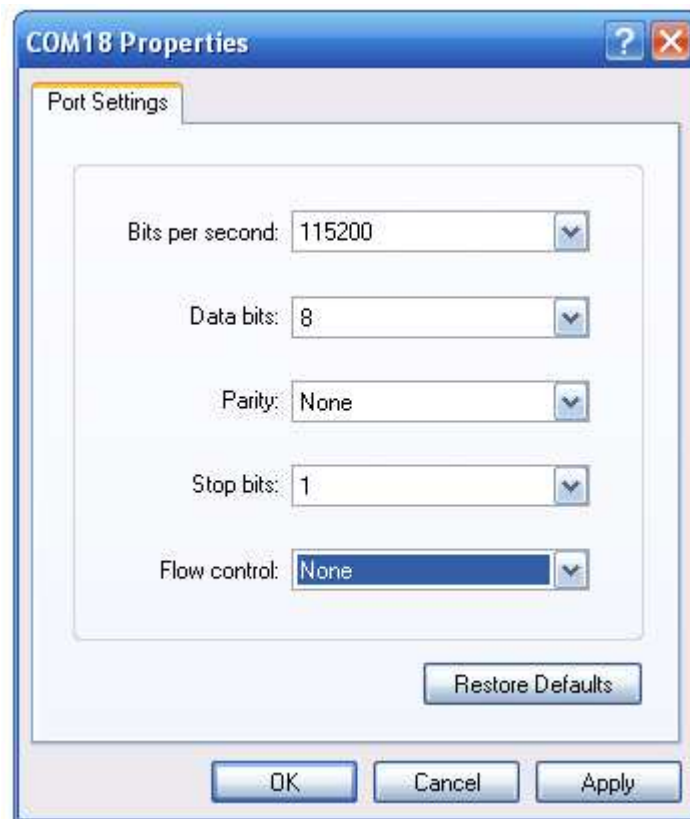
UART je inicijaliziran na način da ima baud rate 115 200, 8 data bits, no parity bits, 1 stop bit i no flow control. Kako bismo mogli pratiti ispis programa u HyperTerminalu potrebno je podesiti i opcije u HyperTerminalu na način kako je inicijaliziran UART. Kako bismo vidjeli koji virtualni COM port je operacijski sustav dodijelio pločici, pogledat ćemo u Device Manager u stavci Hardware u funkciji System control panela. Ukoliko smo podesili HyperTerminal na gore opisan način trebali bismo vidjeti ispis u konzoli HyperTerminala. Način postavljanja komunikacije pomoću HyperTerminala opisan je slikama:



Slika 12: Odabir konekcije



Slika 13: Odabir virtualnog COM porta



Slika 14: Podešavanje postavki veze sa UART-om

Nakon postavljanja parametara HyperTerminal je spreman za komunikaciju. Primjer komunikacije bit će opisan u poglavlju rada sa SD – karticom pošto se koristi naredbeni redak radi lakše navigacije.

Sadržaj datoteke `uartstdio.c` su funkcije za rad sa UART-om. Jedna od češće pozvanih funkcija je `UARTprintf(...)`; pomoću koje ispisujemo niz znakova na konzolu HyperTerminala. Neke od funkcija za čitanje su `UARTgets(...)` i `UARTgetc(...)`. Niti jedna od tih funkcija nije korištena u izradi ove aplikacije iz razloga što su blokirale procesor dok čekaju na upis, pa iz tog razloga nisu radili prekidi Ethernet kontrolera. Umjesto tih funkcija korištena je prekidna rutina za „hvatanje“ jednog karaktera sa UART – a i spremanje u jedan globalni međuspremnik, te funkcija koja prazni taj međuspremnik i vraća korisniku string tek nakon što se pritisne tipka „enter“. Prototip funkcije koja vraća string je: `char strScan(char *retBuff);` Korisniku će se prilikom poziva funkcije vratiti string preko parametra. U glavnom programu je umjesto čekanja da se pročita nešto sa UART – a korištena dotična funkcija, koja se poziva tek kada se



pritisne „enter“, te se onda izvodi jedna od operacija koje su izvedene kao u primjerima za SD – karticu, a za to cijelo to vrijeme poslužitelj neometano radi.

## **MIKRO SD – KARTICA**

Rad sa SD – karticom je, uz realizaciju Web poslužitelja, najteži u cijelom razvoju aplikacije. Pošto je ideja koristiti SD – karticu za pohranu i čitanje jednostavnog log-filea potrebno je implementirati Fat File System (FatFS), koji je standardan za SD – kartice. Za potrebe implementacije FatFS-a korišten je `ff.c`, a za komunikaciju i kontrolu `mmc-ek-lm326965.c`.

Datoteka `ff.c` predstavlja Fat File System modul i dio je eksperimentalnog projekta za realizaciju FatFS – a na jeftinim mikrokontrolerima. Unutar izvornog koda implementirane su i funkcije za rad sa FatFS-om koje omogućavaju čitanje, pisanje, brisanje datoteka te mnoge druge mogućnosti. Neke funkcije koje su korištene za razvoj aplikacije:

```
FRESULT f_mount (BYTE, FATFS*);
```

- postavlja logički drive

```
FRESULT f_open (FIL*, const char*, BYTE);
```

-otvaranje/stvaranje datoteke

```
FRESULT f_read (FIL*, void*, WORD, WORD*);
```

-čitanje podataka

```
FRESULT f_write (FIL*, const void*, WORD, WORD*);
```

-pisanje podataka

Gore navedene funkcije korištene su u funkcijama za realizaciju pisanja podataka određenog tipa u datoteku, za realizaciju čitanja podataka iz datoteke te brisanje datoteke čiji naziv prenosimo u funkciju preko parametra. Funkcije prvo provjeravaju postoji li datoteka te ju otvaraju za čitanje/pisanje ili ju brišu sa SD –

kartice. Kada prenosimo ime datoteke treba paziti da je root direktorij „/“ i da drive nema labelu poput C:/ ili nešto slično.

Datoteka u koju se piše i koju čitamo je tekstualna datoteka logfile.txt, koja se nalazi u root direktoriju SD –kartice i koja služi kao jedan svojevrsni log-file.

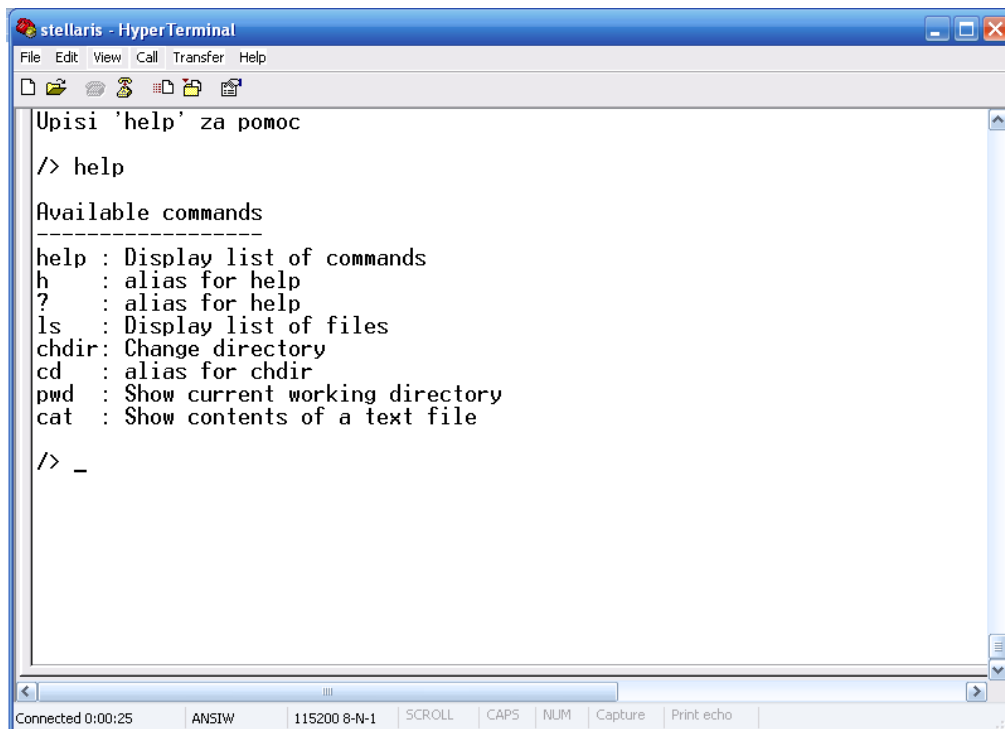
Za pisanje u datoteku korištena je funkcija `int WriteFile(char *file, char *data);`

Za čitanje datoteke korištena je funkcija `int Cmd_cat(int argc, char *argv[]);` koja je implementirana u primjeru za SD – karticu te je ovdje iskorištena u svrhu ispisa sadržaja datoteke na konzolu HyperTerminala.

Za brisanje datoteke korištena je funkcija `void DeleteFile(char *file);`

Radi lakšeg rada sa SD – karticom korišteni su gotovi moduli iz Stellarisovog primjera za SD – karticu kao što su naredbeni redak (command line), te tablica naredbi i njihove funkcije za realizaciju kao što su `pwd`(print working directory), `ls`(list), `cat`(ispis sadržaja datoteke na ekran), `cd`(change directory), te naredba `help` koja ispisuje listu naredbi i njihove funkcionalnosti. Za rad sa naredbenim retkom potrebno je uključiti datoteku `cmdline.c` u projekt. Unutar glavnog programa provjerava se je li ukucana neka naredba, ako je provjerava se postoji li takva. Ukoliko postoji izvršit će se njena funkcionalnost u protivnom će biti ispisano „Bad command“.

Primjer prikaza naredbenog retka i izvršenja naredbe unutar HyperTerminala nakon uspostave komunikacije nalazi se na slijedećoj stranici.



```
stellaris - HyperTerminal
File Edit View Call Transfer Help
Upisi 'help' za pomoc
/> help
Available commands
-----
help : Display list of commands
h    : alias for help
?    : alias for help
ls   : Display list of files
chdir: Change directory
cd   : alias for chdir
pwd  : Show current working directory
cat  : Show contents of a text file
/> _
Connected 0:00:25  ANSIW  115200 8-N-1  SCROLL  CAPS  NUM  Capture  Print echo
```

Slika 15: Naredbeni redak u HyperTerminalu

Valja napomenuti kako treba paziti na to da SD – kartica i OLED zaslon imaju dijeljenu serijsku sabirnicu, te kako treba biti pažljiv u radu sa SD – karticom pogotovo prilikom njenog umetanja i odstranjivanja.

## ***ETHERNET KONTROLER I WEB POSLUŽITELJ***

Realizacija jednog Web poslužitelja na ugradbenom računalnom sustavu nije nimalo jednostavna te se može smatrati radom za jedan diplomski rad ili slično. Unutar ovog projekta korišten je jedan od dva dostupna primjera za realizaciju jednog Web poslužitelja. Poslužitelj se može realizirati na dva načina: „ethernet with uIP“ ili „ethernet with leightweight IP“.

### **ETHERNET WITH uIP**

„uIP“ predstavlja open source implementaciju TCP/IP stoga za ugradbena računala. Razvio ga je Adam Dunkels. Kod je optimiziran tako da zahtjeva manje memorijskog prostora za kod i za podatke. Veličina koda može iznositi i manje od

4 KB, dok veličina podataka ovisi o veličini Ethernet paketa koji se procesiraju (od 100 do 1500 bytea). Unutar izvornog koda implementirani su svi protokoli važni za normalan rad poslužitelja ARP, UDP, TCP, IP, ICMP i SLIP protokol. Dostupni su i različite aplikacije koje se implementiraju uz te protokole: web poslužitelj, web klijent, DNS poslužitelj i još mnogo drugih. Za rad projekta korištene su izvorne datoteke enet\_uip.c, uip.c, uip\_arp.c i uip\_timer.c koje služe za implementaciju protokola za rad sa ethernet kontrolerom sa „uIP“ načinom, dok datoteke httpd.c i dhcp.c služe za implementaciju HTTP poslužitelja i DHCP protokola. Unutar datoteke httpd.c je i Web stranica koja će biti slana klijentu na zahtjev, nakon što pločica dobije svoju IP adresu od strane DHCP poslužitelja. enet\_uip.c je datoteka u kojoj se nalazi i glavni program unutar kojeg se odvija komunikacija između klijenta i poslužitelja te se izvode operacije za rad sa UART-om i SD – karticom. Prije samog rada sa poslužiteljom potrebno je inicijalizirati rad sa ethernet kontrolerom i omogućiti prekide.

```
// Enable and Reset the Ethernet Controller.
SysCtlPeripheralEnable(SYSCTL_PERIPH_ETH);
SysCtlPeripheralReset(SYSCTL_PERIPH_ETH);

EthernetIntDisable(ETH_BASE, (ETH_INT_PHY | ETH_INT_MDIO |
ETH_INT_RXER | ETH_INT_RXOF | ETH_INT_TX | ETH_INT_TXER | ETH_INT_RX));

ulTemp = EthernetIntStatus(ETH_BASE, false);
EthernetIntClear(ETH_BASE, ulTemp);

// Initialize the Ethernet Controller for operation.
EthernetInitExpClk(ETH_BASE, SysCtlClockGet());

// Configure the Ethernet Controller for normal operation.
// - Full Duplex
// - TX CRC Auto Generation
// - TX Padding Enabled
EthernetConfigSet(ETH_BASE, (ETH_CFG_TX_DPLXEN | ETH_CFG_TX_CRCEN |
ETH_CFG_TX_PADEN));

// Wait for the link to become active.
```

```

RIT128x96x4StringDraw("Cekam vezu...", 12, 8, 15);

ulTemp = EthernetPHYRead(ETH_BASE, PHY_MR1);

while((ulTemp & 0x0004) == 0)

{ulTemp = EthernetPHYRead(ETH_BASE, PHY_MR1);

}

RIT128x96x4StringDraw("Veza uspostavljena", 12, 16, 15);

// Enable the Ethernet Controller.

EthernetEnable(ETH_BASE);

// Enable the Ethernet interrupt.

//IntEnable(INT_ETH);

// Enable the Ethernet RX Packet interrupt source.

EthernetIntEnable(ETH_BASE, ETH_INT_RX);

```

Nakon inicijalizacije etehrnet kontrolera slijedi inicijalizacija uIP stoga, te dohvaćanje IP adrese od strane DHCP poslužitelja. Nakon što se dobije važeća IP adresa HTTP poslužitelj može početi posluživati klijenta tako da klijent zatraži web stranicu sa IP adresom poslužitelja.

Web stranica koja se prikazuje u pregledniku pisana je u html standardu 1.0, nalazi se unutar statičkog međuspremnika u httpd.c datoteci, te koristi „loose“ document type definition. Izgled web stranice u jednom od preglednika prikazan je slijedećom slikom:



**Slika 16: Slika Web stranice**

## ETHERNET WITH LEIGHTWEIGHT IP

„Leightweight IP“ je također jedna open source implementacija TCP/IP stoga i također ga je razvio Adam Dunkels. Razlika između te implementacije i uIP je što je ova implementacija zamišljena kako bi se smanjilo korištenje RAM-a, korištenjem memorijske kartice sa FatFS - om. Naime, aplikacija će prvo provjeriti je li umetnuta SD – kartica. Ako je tako, onda će se svi http zahtjevi proslijediti na SD – karticu. Ako SD – kartica nije umetnuta prikazivat će se standardni set stranica zapisan u internoj memoriji. Ova implementacija nije korištena za izradu ovog zadatka iz razloga što nije bilo moguće kompajlirati izvorni kod jer je za razvoj prototipova omogućeno sam kompajliranje 32 KB koda što je uz potrebno znanje bila glavna prepreka za izradu rješenja zadatka.

## TIJEK IZVOĐENJA PROGRAMA

Nakon što je cjelokupan projekt uspješno kodiran potrebno je prevesti aplikaciju i preseliti je u flash mikrokontrolera. Prevođenje obavlja ARM – ov kompajler odabirom opcije „Project“, „Build target...“. Nakon prevođenja kompajler će dojaviti eventualne greške ili upozorenja. Neka upozorenja mogu se zanemariti iako nije preporučeno, dok se sve greške trebaju ukloniti kako bi se aplikacija uspješno prevela. Nakon uspješnog prevođenja potrebno je programirati flash memoriju mikrokontrolera. To se obavlja na vrlo jednostavan nači. Odaberemo opciju „Download“ iz padajućeg izbornika „Flash“. Alat će prebrisati memoriju, programirati ju i verificirati. Nakon uspješnog programiranja možemo restartati mikrokontroler i pratiti tijek izvođenja aplikacije na OLED zaslonu i konzoli HyperTerminala.

Na početku se čeka da na vezu odnosno da se spoji Ethernet kabel sa utorom u hub-u. Nakon toga na ekran se ispisuje „Veza uspostavljena“ te se čeka da pločica dobije IP adresu od DHCP poslužitelja. Nakon što pločica dobije svoju jedinstvenu IP adresu ispisat će se na OLED zaslon. Nakon toga možemo

pritisnuti „Select“ za glavni izbornik u kojem možemo odabrati jednu od opcija pritiskom na određenu navigacijsku tipku. Ukoliko se odabere opcija brisanja logfilea, potrebno je dotičnu opciju potvrditi pritiskanjem tipke „Select“. Za to vrijeme možemo vidjeti web stranicu na kojoj se ispisuju neki osnovni podaci i broj zabilježenih pristupa pločici. Također možemo navigirati kroz file system SD – kartice korištenjem HyperTerminala i konzole, te tako možemo pregledati logfile u koji se zapisuje pristup koji se odabire pritiskanjem tipke „Gore“. Ovo je jednostavna implementacija sustava za praćenje. Pametno bi bilo u logfile zapisati vrijeme kad se pristupilo, no funkcija za to premašuje size limit koda od 32KB. Također u zadatku je navedeno da bi se osim digitalnih pratili i analogni ulazi, no to nažalost nije implementirano zbog nedostatka vremena.

## ZAKLJUČAK

Stellaris LM3S6965 evaluation kit pogodan je za izradu mnogih jednostavnih aplikacija za mikrokontroler LM3S6965 koji je sadržan na razvojnoj pločici uz razne periferije. Velik broj gotovih biblioteka te široka dokumentacija omogućavaju svakome da izrade vlastiti prototip aplikacije za dotično ugradbeno računalo. Iako samo kodiranje zahtjeva puno znanja iz programiranja ugradbenih sustava, neke jednostavne aplikacije mogu su relativno brzo napisati i prevesti uz pomoć razvojne okoline dostupne na CD-u i gotovih primjera.

Glavna prepreka u rješenju zadatka bila je veličina koda koja je morala biti naknadno optimirana zbog prevođenja izvornog koda, te potrebno znanje i vještine koje su sprječavale implementaciju nekih složenijih primjera.

Na kraju valja reći da je rad na ovom zadatku pridonio novim stečenim znanjima na području izrade aplikacija za ugradbene računalne sustave, te iskustvo u razvoju istih u svrhu razvoja sustava za udaljeno praćenje.



## LITERATURA

1. Texas Instruments, LM3S6965 – Datasheet.pdf, 11.lipanj 2009., *Stellaris LM3S6965 Microcontroller – DATA SHEET*, Luminary Micro Stellaris LM3S6965 Evaluation Kit – Documentaiton+software CD, 2.10.2010.
2. Texas Instruments, LM3S6965 – Quickstart-Eval-Kit\_Kiel.pdf, 10.srpanj 2009., *Quickstar - Keil Tools*, Luminary Micro Stellaris LM3S6965 Evaluation Kit – Documentaiton+software CD, 2.10.2010.
3. Texas Instruments, 6965\_EvalBoard\_User\_C.pdf, 9.srpanj 2009., *Stellaris LM3S6965 Evaluation Board – User's Manual*, Luminary Micro Stellaris LM3S6965 Evaluation Kit – Documentaiton+software CD, 3.10.2010.
4. Texas Instruments, AN01270.pdf, 24.lipanj 2009., *Software UART for Stellaris Microcontrollers – Aplication note*, Luminary Micro Stellaris LM3S6965 Evaluation Kit – Documentaiton+software CD, 11.10.2010.
5. Texas Instruments, AN01261.pdf, 26.lipanj 2009., *Aplication note - Using the Stellaris Ethernet Controller with Lightweight IP (lwIP)*, Luminary Micro Stellaris LM3S6965 Evaluation Kit – Documentaiton+software CD, 11.10.2010.
6. Texas Instruments, AN01260.pdf, 26.lipanj 2009., *Aplication note - Using the Stellaris Ethernet Controller with Micro IP (uIP)*, Luminary Micro Stellaris LM3S6965 Evaluation Kit – Documentaiton+software CD, 11.10.2010.

7. Texas Instruments support community, *TI E2E community*  
[http://e2e.ti.com/?DCMP=TIHeaderTracking&HQS=Other+OT+hdr\\_s\\_e2e\\_community](http://e2e.ti.com/?DCMP=TIHeaderTracking&HQS=Other+OT+hdr_s_e2e_community), 5.11.2010.
8. Vučić Mladen, UPOTREBA MIKROKONTROLERA U UGRAĐENIM RAČUNALNIM SUSTAVIMA, Zagreb, Sveučilište u Zagrebu, 2007.,

## NASLOV

WEB POSLUŽITELJ ZA UDALJENO PRAĆENJE TEMELJEN NA UGRADBENOM RAČUNALU

## KLJUČNE RIJEČI

Stellaris LM3S6965, Stellaris evaluation kit, Keil uVision, uIP, Web poslužitelj, SD – kartica, FatFS, UART, GPIO.

## SAŽETAK

Cilj ovog projekta bila je realizacija ugradbenog računalnog sustava za udaljeno praćenje. Za računalo korišten je Stellaris LM3S6965 razvojni sustav sa istoimenim mikrokontrolerom. Sustav omogućava rad sa tipkama, OLED zaslonom, SD – karticom, Ethernet kontrolerom te UART-om. Očitane digitalne vrijednosti (pritiskanje tipke – prijava) bilježene su u logfile-u i prikazivane na web stranici koju poslužuje web server temeljen a uIP implementaciji. Omogućena je i komunikacija preko UART – a sa računalom na kojem pomoću HyperTerminala možemo navigirati kroz SD – karticu pomoću naredbenog retka te prikazivati logfile. Izvorni kod je pisan u okolini uVision, te je u istoj prevođen. Također je pomoću te okoline programirana i sama flash memorija mikrokontrolera.

## TITLE

WEB SERVER FOR REMOTE MONITORING BASED ON AN EMBEDDED COMPUTER

## KEYWORDS

Stellaris LM3S6965, Stellaris evaluation kit, Keil uVision, uIP, Web server, SD – card, FatFS, UART, GPIO.

## SUMMARY

The aim of this project was the realization of an embedded computer for remote monitoring. For this use we used the Stellaris LM3S6965 development system using the LM3S6965 microcontroller. The system allows you to work with switches, OLED display, SD – Card, Ethernet controller and UART. Digital values were read from the GPIO (pressed button) and were recorded in the logfile and screened-in on the website that was served by a web server based on the implementation of uIP. Communication with provided also via UART which communicates with the computer through HyperTerminal and allows navigation through SD – Card using a command line. It also enables you to view the logfile. Source code is written and compiled using uVision. The same environment is used to program the microcotrollers flash memory.