

PROJEKTNO POROČILO

Mikroprocesorski sistemi

WI CUBE

Indikatorska škatlica za vreme

David Selčan

10.6.2013

2000 Maribor

KAZALO STRANI

1	UVOD	4
2	STROJNA OPREMA	5
2.1	MIKROKRMILNIK	5
2.1.1	<i>Headerji za programiranje</i>	6
2.2	WI-FI MODUL	6
2.3	RGB LED IN KRMILJENJE	7
2.4	BATERIJA IN REGULATORSKO VEZJE	8
2.5	TISKANINA	9
2.6	OHIŠJE	10
3	PROGRAMSKA OPREMA	12
3.1	STRUKTURA PROGRAMA	12
3.2	IZVAJANJE PROGRAMA NA FERIFONU	13
3.2.1	<i>Čas, lokacija, vreme</i>	14
3.2.2	<i>Barva</i>	15
3.3	DOLOČITEV BARVE.....	16
4	POVZETEK	17
5	SEZNAM VIROV	18
6	PRILOGE	19

KAZALO SLIK

Slika 1:	Blokovna shema celotnega strojne zgradbe sistema	5
Slika 2:	STM32L mikrokrmilnik.....	6
Slika 3:	RN-131 modul	7
Slika 4:	Shema RGB LED diode	8
Slika 5:	Shema napajalnega sistema.....	9
Slika 6:	Tiskanina WI Cube	10

Slika 7: Načrtovano ohišje (levo je zunanji del, desno pa notranji)	11
Slika 8: Potek programa	13
Slika 9: Program WiColGen za določitev barvnih sekvenc za WI Cube	16

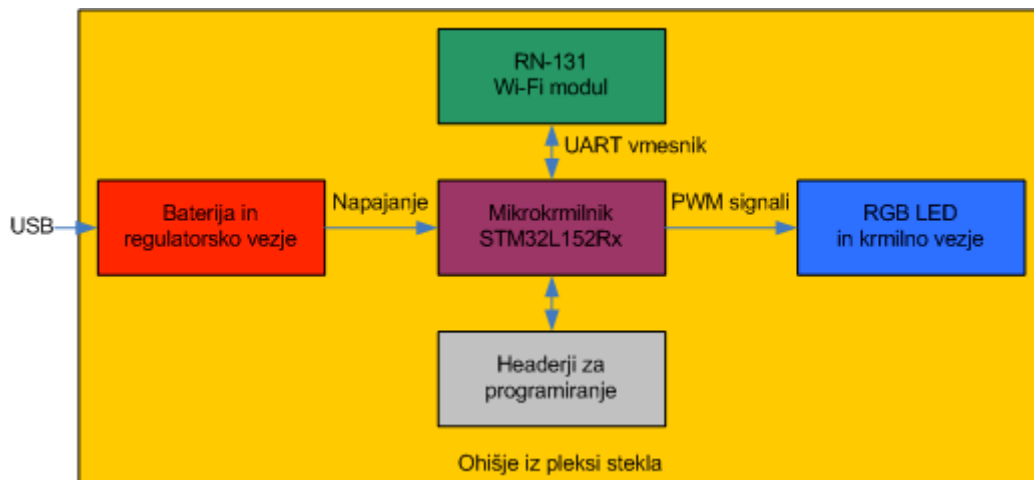
KAZALO TABEL

Tabela 1: Seznam vseh datotek vgrajene programske opreme	13
--	----

1 UVOD

V sklopu projekta smo izdelali indikatorsko škatlica za vreme (Weather Indicator Cube). To je 7 cm velika škatlica iz brušenega pleksi stekla, znotraj katere se nahaja RGB LED dioda, Wi-Fi modul in mikrokontroler. RGB LED dioda sveti na podlagi vremenskih napovedi s strežnika arso.gov.si – za različne vremenske napovedi sveti z različno barvo in različnim časovnim potekom. To nam omogoča, da lahko hipoma vidimo, kakšno vreme bo v bližnji prihodnosti le z pogledom na škatlico. Sistem je načrtovan kot zaključena celot tako da ne potrebuje nobenega vmesnika zunaj svoje škatle - napaja se preko baterije.

2 STROJNA OPREMA



Slika 1: Blokovna shema celotnega strojne zgradbe sistema

Kot že omenjeno je bil cilj projekta zagotoviti samozadosten sistem. Vse komponente za potrebno funkcionalnost (razen seveda raznih spletnih strežnikov) so vgrajene v format, katerega lahko vstavimo v škatlico. Napajanje baterije je izvezeno preko USB povezave.

Mikrokrmilnik v sistemu tako skrbi za komunikacijo z zunanjim svetom preko Wi-Fi povezave, in nato prižiga ali ugaša LED diodo na podlagi te komunikacije. Dostop do mikrokrmilnika lahko poteka preko WiFi modula (Ad hoc omrežje), ali preko raznih headerjev (programiranje in UART povezava) ter potencialno tudi preko USB povezave.

2.1 Mikrokrmilnik

Osrednji element sistema je STM32L152RX mikrokrmilnik iz linije STM32 32 bitnih ultra low power mikrokrmilnikov podjetja STmicroelektronics.



Slika 2: STM32L mikrokrmilnik

Vsebuje ARM Cortex M3 jedro in je zaradi nizke porabe primeren za sistem, ki se bo napajal preko baterije. Vsebuje vso periferijo, ki jo potrebujemo za komunikacijo/krmiljenje vseh podsistemov (vgrajen UART za komunikacijo z RN 131 modulom, timer z 4 kanali, kar nam omogoča krmiljenje RGB LED diode z enim timerjem). Prav tako ima več uart vmesnikov, kar nam omogoča da uporabimo enega za potrebe razvoja – razhroščevanje preko PCja. Napajan je iz 3.3V, kar je ista napetost kot Wi-Fi modul, torej ne potrebujemo ločenih regulatorjev.

Vgrajeno ima več notranjih virov ur in prav tako podpira mnogo načinov nizke porabe, kar pomeni, da bomo lahko naknadno optimizirali porabo glede na potrebe sistema.

2.1.1 Headerji za programiranje

Za programiranje in razhroščevanje smo uporabili SWD vmesnik, povezan preko 6 pinskega headerja. Vključno z tem imamo pri sistemu še 3 pinski header, preko katerega lahko vodimo UART komunikacijo s PCjem preko UART na USB mosta (za namene razvoja programske opreme. Prav tako smo predvideli še dva 2 pinska mostiča, s katerimi bomo lahko izbirali način vklopa (na primer tovarniška ponastavitev RN modula).

2.2 Wi-Fi modul

Za Wi-Fi modul smo uporabili RN-131 podjetja Roving Networks. Modul podpira ISO 802.11b in 802.11g standarda, prav tako ima vgrajen celoten TCP/IP sklad. To nam močno poenostavi razvoj sistema, saj bi razvoj TCP/IP sklada bil probsežen cilj za čas, ki ga imamo na razpolago. RN-131 ima tudi vgrajeno vso opremo za fizični nivo Wi-Fi povezav, vključno z anteno.



Slika 3: RN-131 modul

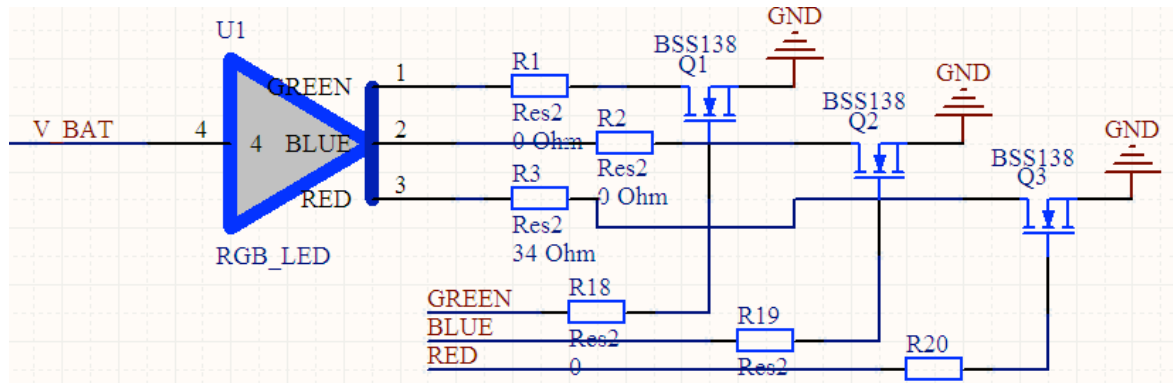
RN-131 komunicira z mikrokrmilnikom preko UART vmesnika s možnostjo kontrolo pretoka (flow control – RTS in CTS), kjer lahko programsko definiramo hitrost in ostale nastavitve povezave.

Ima tudi druge linije, ki se direktno povežejo z mikroprocesorjem, to so linija za reset, linija za prekinitev spanja in linija za povrnitev tovarniških nastavitvev.

Z vidika porabe je Wi-Fi modul aktiven le majhno količino časa (v povprečju enkrat na uro za manjše količine časa, kjer mu poraba ne naraste preko 100mA) torej je njegova poraba zanemarljiva v primerjavi z ostalim sistemom.

2.3 RGB LED in krmiljenje

Za RGB LED diodo smo uporabili 5mm RGB LED z skupno anodo. Posamezne katode različnih barv smo preko upora in N-MOSFET (BS138) tranzistorja speljali proti zemlji. Izbira tranzistorja je potekala predvsem na kriteriju upornost drain-source pri nizkih napetostih na gate-u. Na bazo tranzistorja smo nato pripeljali PWM signal iz mikrokrmilnika. Upore smo dimenzionirani tako, da bo čez LED diodo pri najvišjem napajanju tekel najvišji dovoljen trenutni tok – ker nismo imeli datasheeta za 5mm RGB LED diodo, smo ocenili ta tok na 30 mA. Da ne bomo prekoračili povprečnega dovoljenega toka smo poskrbeli v programski kodi.



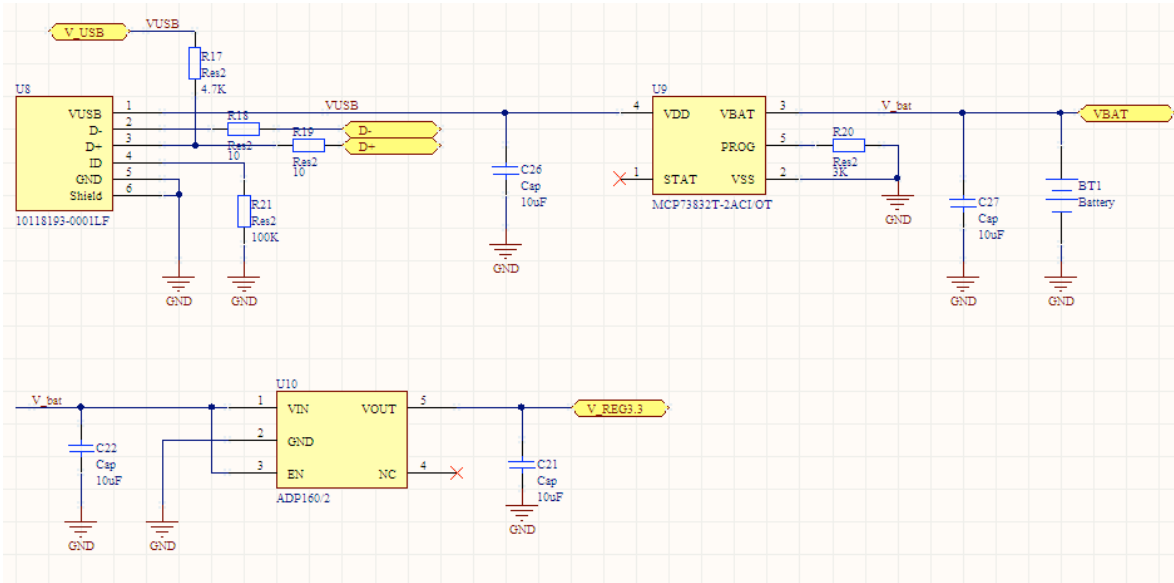
Slika 4: Shema RGB LED diode

Sistem je načrtovan tako, da minimizira izgube, ki nastanejo ob vklopu LED diod. Tukaj je to kritično, saj bo RGB LED delovala večino časa in bo tako večina izgub prav na njej.

Frekvenca PWM signala bo 1kHz, kar je dovolj visoko da ne bo motilo človeka, ko bo gledal v škatlico.

2.4 Baterija in regulatorsko vezje

Baterija je Li-Ion s 950 mAh kapacitete in nominalno napetostjo 3.8V. Iz te baterije je nato napajan mikrokrmilnik in RN-131 modul preko linearnega regulatorja ADP160 proizvajalca Analog devices. Izbira tega regulatorja temelji na dejstvu, da ima zelo nizek izgubni tok pri nizkih tokovih in je zato idealen za naprave, ki preživijo dosti časa v stanju zelo nizke porabe (kar drži za našo škatlico).

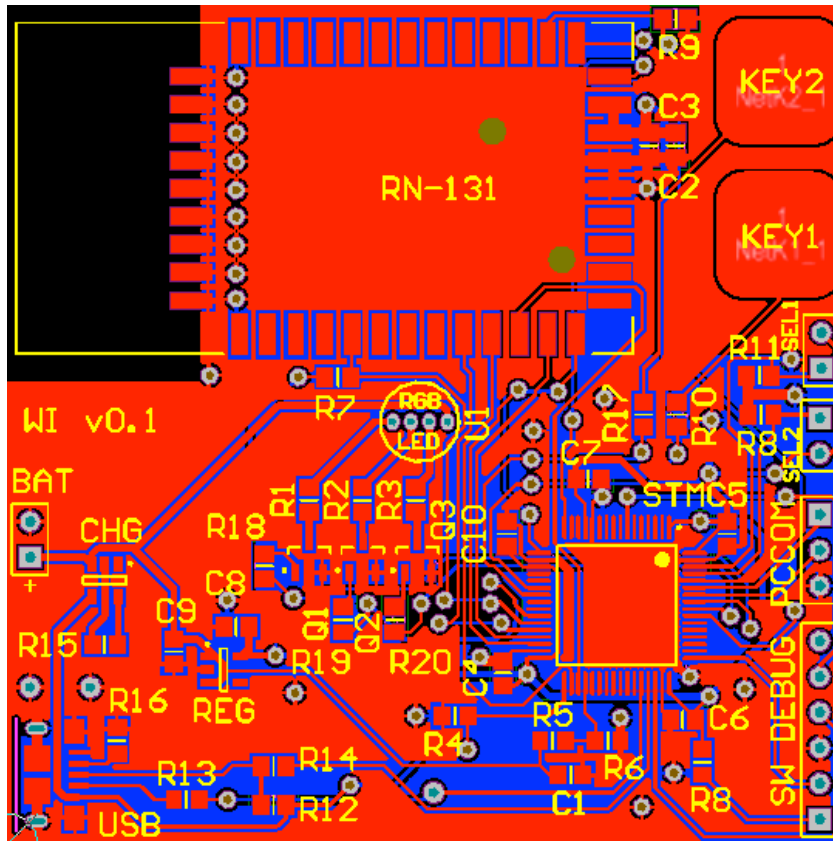


Slika 5: Shema napajalnega sistema

Baterijo napajamo preko micro-USB konektorja z uporabo MCP73832T polnilnega vezja proizvajalca Microchip. Čip popolnoma integrira algoritem polnjenja Li-Ion baterij in je tako zelo primeren za našo aplikacijo, saj se ne rabimo ukvarjati z polnilnim vezjem.

2.5 Tiskanina

PCB smo načrtovali z uporabo programa Altium Designer. Končno vezje smo spravili na velikost 6 cm krat 6 cm. Vezje smo dali izdelati profesionalno, saj je RN-131 modul zaradi kontaktov pod samim modulom težko pajkati na domače izdelan PCB.



Slika 6: Tiskanina WI Cube

Na tiskanini lahko zasledimo tudi dve kapacitivni tipki, ki pa smo ju dodali na tiskanino izključno iz dejstva, da smo za njih imeli prostor, nimata pa nobene funkcionalne vloge za naš modul.

2.6 Ohišje

Ohišje za WI Cube smo načrtovali z programom FreeCAD. V njemu smo ustvarili ohišje, ki je sestavljeno iz dveh delov. En je velikosti 70mm x 70mm x 70mm, in je zunanja površina, v tega se pa vstavi manjša kocka, velikosti 66mm x 66mm x 68mm. Ta vsebuje tudi poseben prostor za tiskanino z vsemi elementi (baterija je največja).

Slika 7: Načrtovano ohišje (levo je zunanji del, desno pa notranji)

Ohišje smo izdelali iz kosa pleksi stekla. Tega smo rezali na način, da smo najprej zarezali del pleksi stekla in nato pretrgali z vzporedno silo rezu. Vsak kos pleksi stekla smo potem zbrusili, da je postal pravilne oblike in da je lepše difuzirana svetloba, ko sveti RGB LED dioda skozi.

Škatlica je bila nato zlepjena skupaj z sekundnim lepilom.

3 PROGRAMSKA OPREMA

Vgrajena programska oprema za WI Cube je bila načrtovana in prevedena v sklopu okolja YAGARTO, to je GCC okolje za Windows operacijski sistem, ki podpira razvoj kode za ARM mikroprocesorje. Vsa koda je bila pisana v programskem jeziku c.

Razvojno okolje uporablja poleg ARM-GCC prevajalnika še knjižice za STM32L serijo mikrokontrolerov ter predlogo za pisanje programov, ki vključuje pomožno kodo za konfiguracijo, program za začetek izvajanja in skript za povezovanje (linker script). Za programiranje in razhroščeavnje pa uporabljamo ST-LINKv2 napravo, ki se nahaja na STM32L-DISCOVERY razvojni plošči.

3.1 Struktura programa

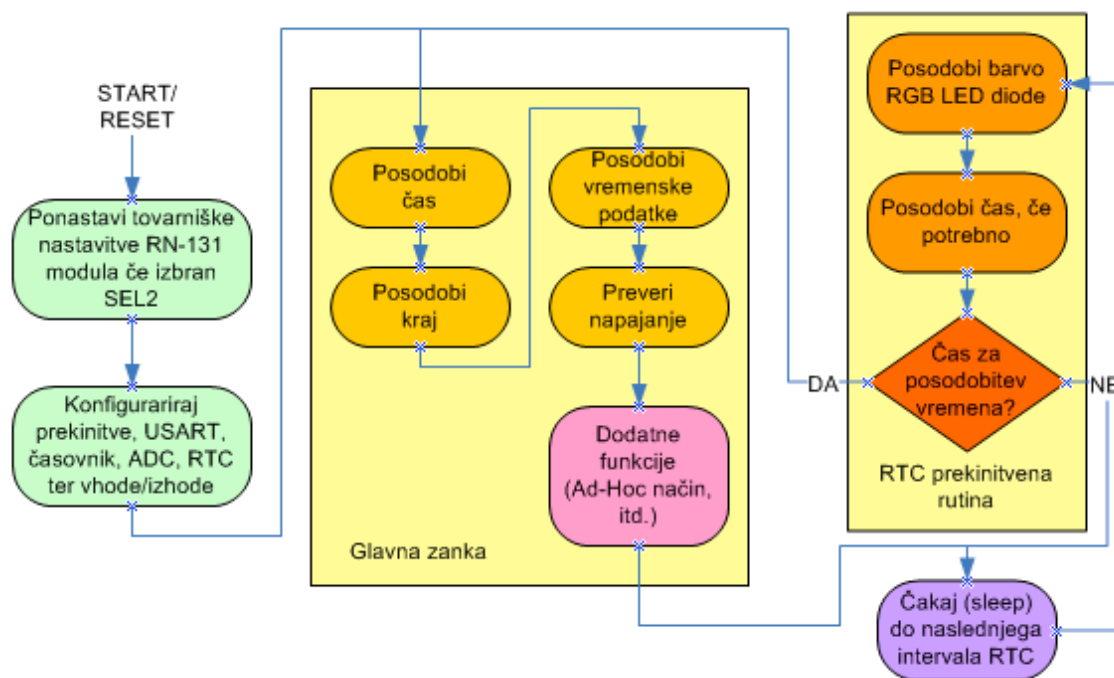
Seznam vseh datotek vključno z njihovim nameni se nahaja v sledeči tabeli.

Ime datotek	Namen
main.c	Glavna datoteka, v kateri se nahaja glavni program vključno z inicializacijo splošnih funkcij.
stm32l1xx.h, cgpio.h	Datoteke knjižice STM32L, katere vsebujejo funkcije in strukturo za upravljanje raznih naprav mikrokontrolerov.
defines.h	Header datoteka, ki vsebuje nekatere spremenljivke, ki se jih lahko spremeni po potrebi.
usart.c, usart.h	Vsebuje kodo in vmesnik za upravljanje z USART komunikacijo.
rtc.c, rtc.h	Vsebuje kodo za upravljanje z uro realnega časa, vključno z funkcijami za zakasnitev.
Inet.c, inet.h	Vsebuje kodo za komunikacijo in upravljanje z RN-131 modulom, prav tako vso kodo za parsanje spletnih strani (čas, lokacija, vreme).
Regex.c, regex.h	Vsebuje pomožne funkcije za parsanje spletnih strani – iskanje specifičnih zaporedij znakov.

Location.c, location.h	Vsebuje kodo za pretvorbo pridobljene lokacije v lokacije, ki jih pozna arso strežnik za vreme.
Time.c, time.h	Vsebuje strukturo in funkcije za delo z časom – trenutnim in časom napovedi.
Weather.c, weather.h	Vsebuje kodo za delo z vremenskimi podatki – kako se podatki iz vremenskega strežnika preslikajo v naše, ki jih znamo prikazati.
adc.c, adc.h	Vsebuje kodo za inicializacijo in upravljanje z ADC enoto.
Timer.c, timer.h	Vsebuje kodo za inciiializacijo in upravljanje z časovniki.
Color.c, color.h	Vsebuje kodo za prižiganje pravih barv ob pravih časih.

Tabela 1: Seznam vseh datotek vgrajene programske opreme

3.2 Izvajanje programa



Slika 8: Potek programa

Program na mikrokrmilniku teče približno tako, kot kaže shema. Ob zagonu se primarno preveri, če je potrebno ponastaviti RN-131 modul na tovarniške nastavitve. Sledi konfiguracija vseh podsistemov, ki jih potrebujemo, prav tako se konfigurira RN-131 modul (nastavitve, ki so po resetu drugačne). Celoten čas po tem nam teče prekinitvena rutina s frekvenco 100 Hz (s pomočjo vgrajenega števca realne ure (Real Time Clock)).

Sistem se nato preko RN-131 modula poveže s strežnikom za čas, za lokacijo in za vreme ter posodobi te podatke. Prav tako preveri tudi napajanje (če prisoten USB kabel in če dovolj visoka napetost). V primeru potrebe po drugih načinih delovanja (npr. prisoten USB napajanje in tako mogoč Ad-Hoc način) se to tudi izvede tukaj. Ko vse to zaključimo čaka, dokler ne mine 1 ura, da znova izvede to rutino (z izjemo preverjanja napajanja, kar izvaja enkrat na sekundo).

Celoten čas med tem teče prekinitvena rutina, v kateri se posodablja trenutni čas (če to potrebno) in se posodablja RGB LED osvetlitev po vnaprej določenem vzorcu, glede na vreme.

3.2.1 Čas, lokacija, vreme

Čas se prebere iz strežnika DAYTIME, in se shrani v strukturo z podatki za leto, mesec, dan, uro in minuto.

Lokacija se prebere iz strežnika, ki ima povezavo do baze, kjer se IP naslovi lahko dekodirajo v geolokacijo – geografsko dolžino in širino. Ta se nato po algoritmu najbližje točke primerja z vsemi lokacijami vremenskih napovedi, kjer se določi pravilna lokacija. Naprava podpira slednje lokacije (regije): AT_KAERNTEN, AT_STEIERMARK, HR_PRIMORSKO-GORANSKA, HR_ZAGREBACKA, SI_BELOKRANJSKA, SI_BOVSKA, SI_DOLENJSKA, SI_GORENJSKA, SI_GORISKA, SI_KOCEVSKA, SI_KOROSKA, SI_OSREDNJESLOVENSKA, SI_NOTRANJSKO-KRASKA, SI_OBALNO-KRASKA, SI_PODRAVSKA, SI_POMURSKA, SI_SAVINJSKA, SI_SPODNJEPOSAVSKA, SI_ZGORNJESAVSKA.

Vremenska lokacija se pridobi, da se povežemo na server arso.gov.si, kjer se v .xml datoteki hrani napoved za vse zgoraj omenjene regije z intervalom treh ur. Ta datoteka se prebere, in se za vsako napoved, ki je napovedana za današnji ali jutrišnji dan, izlušči podatek za oblačnost, padavine, temperatura in vetrno hitrost.

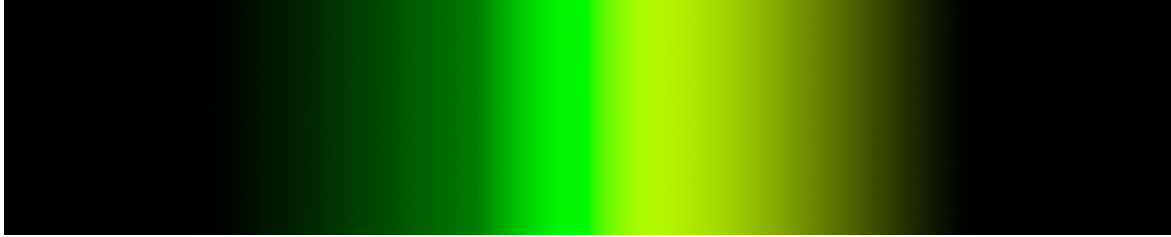
```
<tsUpdated>10.06.2013 21:50 CEST</tsUpdated>
<valid>11.06.2013 8:00 CEST</valid>
<valid UTC>11.06.2013 6:00 UTC</valid UTC>
<tp_3h_cum>0.2</tp_3h_cum>
<sn_3h_cum>0</sn_3h_cum>
<tp_6h_cum/>
<sn_6h_cum/>
<tp_30min_acc/>
<sn_30min_acc/>
<tp_1h_acc/>
<sn_1h_acc/>
<tp_3h_acc>0.2</tp_3h_acc>
<sn_3h_acc>0</sn_3h_acc>
<tp_6h_acc/>
<sn_6h_acc/>
<tp_12h_acc>4.5</tp_12h_acc>
<sn_12h_acc>0</sn_12h_acc>
<tp_24h_acc/>
<sn_24h_acc/>
<rrHh>3</rrHh>
<wwsyn_shortText>plohe</wwsyn_shortText>
<nn_shortText>oblačno</nn_shortText>
<nn_icon-wwsyn_icon>overcast_lightSHRA</nn_icon-wwsyn_icon>
<ts_icon/>
<tnsyn>12</tnsyn>
<txsyn/>
<t>15.2</t>
<ddff_icon>lightW</ddff_icon>
<ff_val>0.7</ff_val>
<ffmax_val>4.4</ffmax_val>
```

Slika 9: Prikaz enega vnosa vremenskih podatkov

Ti podatki se nato dekodirajo v grobe kategorije: Lepo vreme, oblačno, dež, sneg, toča, megla. Od tega ima vsaka kategorija podkategorije, ki se ločijo po moči pojava, po temperaturi in moči vetra.

3.2.2 Barva

Barvo, ki jo mora WI Cube prikazovati se hrani v strukturah, kjer je shranjena 24 bitna barvna koda in 8 bitna trajanja (2.55 sekund naenkrat maksimum). Barve nato interpretiramo fizično tako ,da je vrednost na vsakem kanalu (rdeče, zeleno in modro) linearno sorazmerna z trajanjem osvetlitve, kar kontroliramo z PWM signalom. Na primer, za barvo rdeče 255, zeleno 127, modro 0, bo rdeči del diode osvetljen celoten čas, zelena polovico časa in modra sploh ne.

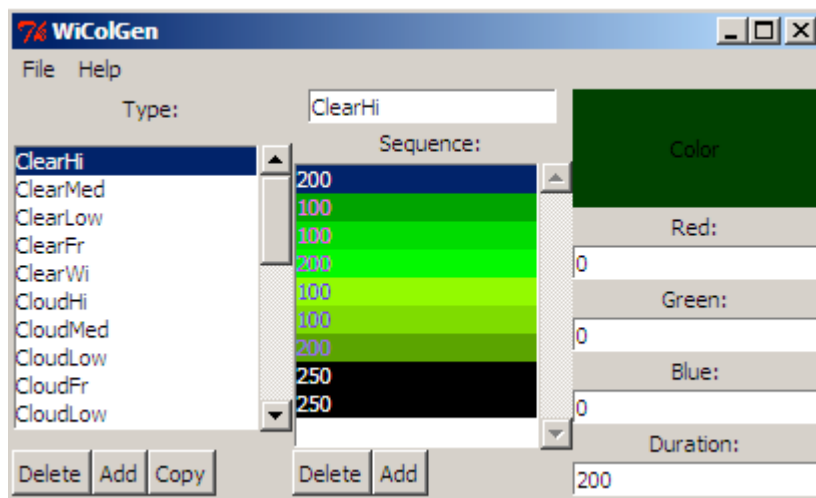


Slika 10: Primer barvne sekvence za jasen, topel dan.

Te barvne strukture se nato nanizajo ena za drugo in se izvede vmes linearna interpolacija, da lahko svetloba utripa. Vsakih 10 ms zamenjamo vrednost na naslednjo.

3.3 Določitev barve

Za določitev barvnih zaporedij za osvetlitev smo napisali program z grafičnim uporabniškim vmesnikom, preko katerega se direktno generira koda, katero vključimo v vgrajeno programsko opremo. Program smo napisali v Python-u.



Slika 11: Program WiColGen za določitev barvnih sekvenc za WI Cube

S programom lahko vstavljamo in zbrisujemo sekvence in znotraj sekvenc barvo in čas trajanja vsakega okvirja. To nam nato program prikaže kako bo približno naša sekvenca zgedala. Lahko tudi shranjujemo in ponovno nalagamo sekvence teh zaporedij, za lažjo določitev različnih barvnih sekvenc.

4 POVZETEK

V sklopu projekta smo razvili indikatorsko škatlico (RGB LED diodo v škatlici iz pleksi stekla). Preko Wi-Fi modula se poveže s strežnikom za vreme in preber vremensko napoved. Nato kontrolira svetlobo iz RGB LED diode glede na napoved vremena, kategorizirano v naprej definirane kategorije. Sistem je popolnoma avtonomen, z močnostjo polnjenja baterije preko micro USB povezave.

Funkcionalno smo napravo spravili do te mere, da omogoča vse glavne funkcije. Lahko bi še dalje razvili USB povezavo, spletni strežnik za konfiguracijo preko interneta, boljšega odpravljanja in signaliziranja težav, mogoče celo možnost posodabljanja barv, ter boljši vmesnik za upravljanje dostopnih točk. Kljub temu pa naprava služi svoji funkciji prikazovanja prihajajočega vremena na jasen, vendar nevdoren način.

5 *SEZNAM VIROV*

1. Twitter Mood Light The Worlds Mood in a Box. Dostopno na: <http://www.instructables.com/id/Twitter-Mood-Light-The-Worlds-Mood-in-a-Box/> [12. 3. 2012].
2. LED Cube Night Light Dostopno na: <http://www.instructables.com/id/LED-Cube-Night-Light/> [12. 3. 2012].

6 PRILOGE

- Programska koda.
- Altium Designer PCB project.
- WiColGen.py program, vključno z barvno sekvenco.
- FreeCAD datoteke za ohišje.