UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Gašper Mlinšek

Sistem za rezultatski prikaz in upravljanje baseball tekme

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Dušan Kodek

Ljubljana, 2010

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01615/2009
Datum: 15.10.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **GAŠPER MLINŠEK**

Naslov: **SISTEM ZA REZULTATSKI PRIKAZ IN UPRAVLJANJE BASEBALL TEKME**

SYSTEM FOR A BASEBALL GAME SCORE DISPLAY AND MANAGEMENT

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Analizirajte možne rešitve za izgradnjo sistema, ki je sposoben brezžično pošiljati rezultat baseball tekme na prikazovalni del (semafor) in istočasno shranjevati vse podatke o tekmi. Sistem naj bo zasnovan tako, da se prikazovalni del fizično lahko postavi na dobro vidno oddaljeno lokacijo, upravljalni del pa na zapisnikarsko mizo. Upravljalni del naj bo zgrajen na osnovi osebnega računalnika in naj vsebuje bazo podatkov, v kateri se hranijo statistični podatki o tekmi in o igralcih. Razvijte in izberite oziroma izdelajte strojno in programsko opremo vseh delov sistema. Sistem preizkusite in preverite pravilnost njegovega delovanja.

Mentor:
prof. dr. Dušan Kodek

Fakulteta za računalništvo in informatiko
2
UNIVERZA V LJUBLJANI

Dekan:
prof. dr. Franc Solina

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01615/2009
Datum: 15.10.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **GAŠPER MLINŠEK**

Naslov: **SISTEM ZA REZULTATSKI PRIKAZ IN UPRAVLJANJE BASEBALL TEKME**

SYSTEM FOR A BASEBALL GAME SCORE DISPLAY AND MANAGEMENT

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Analizirajte možne rešitve za izgradnjo sistema, ki je sposoben brezžično pošiljati rezultat baseball tekme na prikazovalni del (semafor) in istočasno shranjevati vse podatke o tekmi. Sistem naj bo zasnovan tako, da se prikazovalni del fizično lahko postavi na dobro vidno oddaljeno lokacijo, upravljalni del pa na zapisnikarsko mizo. Upravljalni del naj bo zgrajen na osnovi osebnega računalnika in naj vsebuje bazo podatkov, v kateri se hranijo statistični podatki o tekmi in o igralcih. Razvijte in izberite oziroma izdelajte strojno in programsko opremo vseh delov sistema. Sistem preizkusite in preverite pravilnost njegovega delovanja.

Mentor:
prof. dr. Dušan Kodek

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:
prof. dr. Franc Solina

I Z J A V A O A V T O R S T V U

diplomskega dela

Spodaj podpisani Gašper Mlinšek,

z vpisno številko 63000225,

sem avtor diplomskega dela z naslovom:

Sistem za rezultatski prikaz in upravljanje baseball tekme

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom

  prof. dr. Dušana Kodeka

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela

- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 7.4.2010 Podpis avtorja:

Zahvala

Za pomoč pri izdelavi naloge se zahvaljujem mentorju, prof. dr. Dušanu Kodeku.
Zahvala gre tudi podjetju LEA d.o.o. za pomoč pri reševanju problemov med izdelavo diplomskega dela.

Kazalo

Seznam uporabljenih kratic in simbolov

ASCII	American Standard Code for Information Interchange
CAN	Controller-Area Network
DCE	Data Circuit-terminating Equipment
DSSS	Direct Sequence Spread-Spectrum
DTE	Data Terminal Equipment
GND	Ground
I2C	Inter-Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
LED	Light-Emitting Diode
LSB	Least Significant Byte
MSB	Most Significant Byte
OFDM	Orthogonal Frequency-Division Multiplexing
RS232	Recommended Standard 232
RS485	Recommended Standard 485
Rx	Recieving Signal
SPI	Serial Peripheral Interface
SQL	Structured Query Language
Tx	Transmitting Signal
WLAN	Wireless Local Area Network

Povzetek

Tema diplomske naloge govori o načrtovanju in izdelavi sistema, ki služi brezžičnemu upravljanju statistike in predstavitvi rezultata baseball tekme. V nalogi so najprej opisani načrtovanje nato zgradba in delovanje električnih vezij. Sistem je glede na funkcijo razdeljen na dva dela. Prikazovalni del je namenjen obveščanju gledalcev o poteku tekme, zato je zasnovan tako, da se ga fizično da postaviti na oddaljeno, višje ležečo in dobro vidno lokacijo. Upravljalni del služi za interakcijo s prikazovalnim delom. Brezžična povezava omogoča poljubno izbiro upravljalnega mesta znotraj njenega dosega. Za povezavo skrbita brezžični modul Avisaro in brezžična mrežna kartica prenosnega računalnika.

V nadaljevanju naloge je predstavljena zasnova programske opreme. Upravljalni del vsebuje napisan program, ki omogoča upravljanje s športnim semaforjem in shranjevanje statističnih podatkov tekme v podatkovno bazo.

V zaključku je sistem ovrednoten glede na podane zahteve, naštetih je tudi nekaj možnih idej za izboljšave, ki so se pojavile tekom razvoja.

Ključne besede:

brezžična povezava, Avisaro WLAN Module 2.0, LED, podatkovna baza

Abstract

The thesis describes a design and realization of a system that is capable of wireless managing statistic data and displaying score of a baseball match. The description begins with a design and structure of electronic circuits. System is divided into two units, each with its own function. Display unit displays the information for viewers on a large LED. Remote unit works as a managing console for display unit. Wireless connection between the units allows comfortable managing of a the scoreboard. The connection is established between the Avisaro WLAN Module and wireless adapter in the PC.

The second half of the thesis describes the softwares design. Custom program was written that manages the display of data on the LED scoreboard and storing of data in local relational database.

In the last chapter the system is compared against given requirements. Also, some additional thoughts on improvements and upgrades are given.

Key words:

wireless communication, Avisaro WLAN Module 2.0, LED, relational database

1 Uvod

Prikazovalniki informacije na osnovi LED so že nekaj časa zelo razširjeni za različne načine obveščanja. Danes jih uporabljamo praktično povsod, vidimo jih na področjih prometne signalizacije, reklamiranja, športnih prireditev ... Njihovi bistveni prednosti sta dobra vidljivost in enostavno upravljanje. Služijo hitremu prikazu informacije relativno velikemu krogu uporabnikov.

Mehanski športni semaforji za prikaz rezultatov so se prvič pojavili že v devetnajstem stoletju. Po pojavu informacijske tehnologije so jih zamenjale digitalne izvedbe, predvsem zaradi lažjega načina upravljanja in hitrejšega pretoka informacije. Zaradi majhne potrebe po takih napravah (vsako igrišče potrebuje največ eno) in zahtevi po uporabi v določenem okolju (vsako igrišče je unikatno) so v večini primerov narejene po naročilu.

Ideja o digitalnem športnem semaforju se je pojavila pri Baseball Softball društvu Ježica. Cilj je bil uporabiti pridobljeno znanje na univerzi ter narediti športni semafor za prikazovanje rezultata in vodenje statističnih podatkov baseball tekme. Zahteve so bile med drugim brezžično upravljanje iz bližnje okolice, shranjevanje rezultata in možnost prikaza preko domače spletne strani. Informacija bi bila tako dostopna tudi osebam, ki se fizično ne bi udeležile dogodka. Poleg tega je bilo potrebno ugoditi še dodatnim zahtevam, kot so možnost enostavnega upravljanja rezultata iz zapisnikarske mize, oddaljene približno 30 metrov ter konsistentnost in centralizacija podatkov. Pomembna je tudi velikost celotnega semaforja, saj je potrebno rezultat razbrati iz vsake točke na igrišču, ki je v najslabšem primeru oddaljena tudi do 100 metrov. V ta namen so bila uporabljena vezja sestavljena iz LED, dimenzije 260 krat 145 milimetrov. Vsako vezje je zmožno prikazati poljuben ASCII znak. Vse skupaj mora biti ustrezno zaščiteno proti podnebnim vplivom in mehanskim poškodbam, saj naj bi bil semafor postavljen na prostem v neposredni bližini igrišča.

Semafor je sestavljen iz dveh glavnih delov, prikazovalnega in upravljalnega. Funkcija prikazovalnega dela je izpisovanje rezultata na športnem semaforju, upravljalni del pa je namenjen upravljanju semaforja iz zapisnikarske mize. Zaradi večje kompatibilnosti so bili pri načrtovanju uporabljeni splošno razširjeni načini brezžične računalniške komunikacije (IEEE 802.11b, RS232). Prednost tega je lažje vzdrževanje semaforja, saj ob morebitni okvari

brez težav zamenjamo upravljalni del, potrebna je le predhodna namestitev programske opreme.

V prikazovalnem delu semaforja je vgrajen brezžični modul, ki sprejema podatke iz oddajnega dela. Preko zaporedne povezave jih pošilja na Atmelov mikroprocesor, ki upravlja z LED vezji. Protokol je osnovan na nivoju sporočil, vsako sporočilo ima vključeno pariteto za preverjanje pravilnosti prenosa podatka.

Upravljalni del uporablja brezžično mrežno kartico, ki ji je dodan program za emulacijo RS232 protokola. Računalnik se fizično poveže preko brezžičnega omrežja, logično povezava izgleda kot trožilni kabel za zaporedni prenos podatkov s signali Rx, Tx in GND. Upravljalni del ima tudi funkcijo zapisa statističnih podatkov v bazo za kasnejšo analizo.

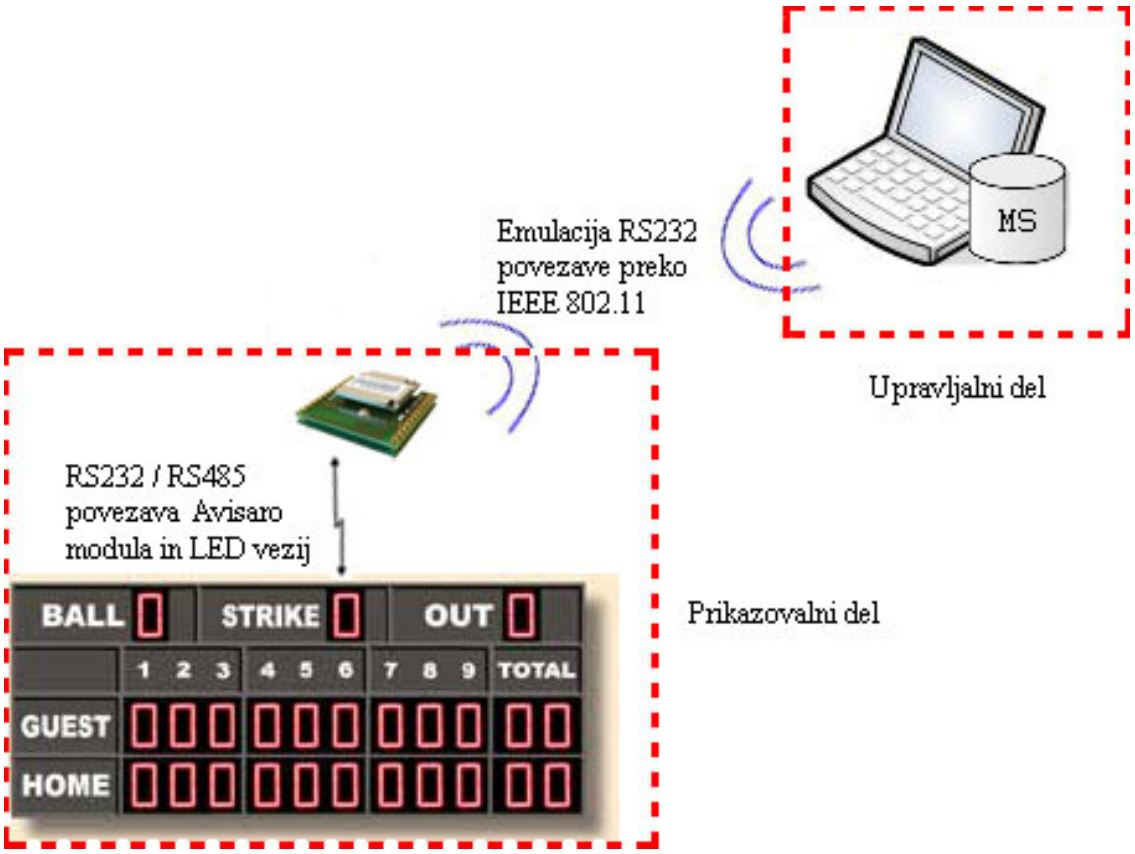
Problem je predstavljal dodaten izziv, saj ima med seboj prepletenih več različnih računalniških področij od načrtovanja strojne opreme, programiranja algoritma do upravljanja s podatkovno bazo.

2 Strojna oprema

Športni semafor je zgrajen iz dveh medsebojno ločenih enot (slika 1). Prikazovalni del predstavlja fizično največji del projekta, njegov namen je vizualni prikaz rezultata. Narejen je izključno iz strojne opreme, s katero upravljamo s posebej formatiranimi ukazi, poslanimi iz upravljalnega dela.

Upravljalni del je narejen za enostavno upravljanje iz neposredne bližine. Narejen je programsko, uporablja prenosnik z vgrajeno brezžično mrežno kartico. Na prikazovalni del pošilja informacijo iz uporabniku prijaznega programa. Program ima vmesnik zasnovan tako, da se vidi vsebina tudi, če je semafor izven vidnega polja.

Upravljalni del ima tudi možnost shranjevanja podatkov v lokalni podatkovni bazi. Namen uporaba lokalne podatkovne baze je shranjevanje podatkov v primeru, če nimamo dostopa do interneta. Strojna oprema upravljalnega dela je izbrana tako, da jo je ob morebitni okvari brez težav zamenjati.

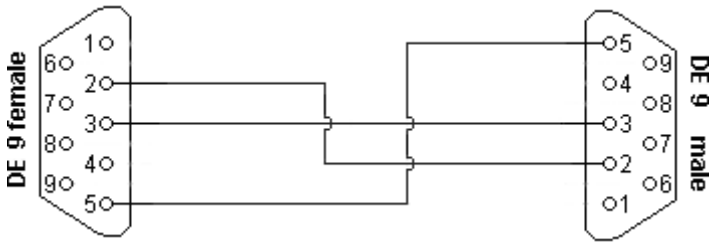


Slika 1: Prikaz športnega semaforja sestavljenega iz prikazovalnega dela (levo) in upravljalnega dela (desno).

2.1 Standardi za prenos informacije

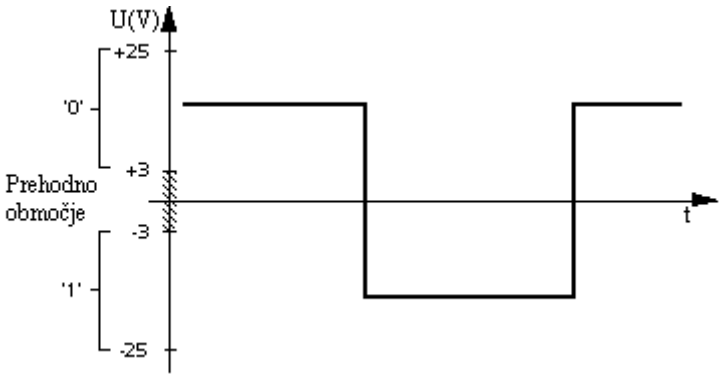
2.1.1 RS232 in RS485

RS232 [2] je telekomunikacijski standard za zaporedni prenos informacije med dvema napravama, imenovanima podatkovna terminalska oprema (DTE) in komunikacijska podatkovna oprema (DCE). Standard določa 22 različnih signalov, preko katerih se napravi sporazumevata. Za povezavo brez dodatnih kontrol zadostujejo signali za oddajo (TxD), sprejem (RxD) in ozemljitev signala (GND). To nam omogoča, da lahko namesto 25-signalnega DB-25 vtiča uporabimo 9-signalni DE-9 vtič (slika 2).



Slika 2: RS232 povezava med DTE in DCE brez strojnega nadzora pretoka.

Vrednost signala je določena bipolarno. Na območju od +3V do +25V naprava zazna visoko stanje ali v tem primeru logično 0. Območje med -3V in -25V predstavlja nizko stanje ali logično 1. Vmesno območje med -3V in 3V imenujemo prehodno območje. V prehodnem območju je prepovedano daljše zadrževanje signala, širina prehodnega območja služi kot odpornost proti šumu (slika 3).



Slika 3: Vrednost signala za visok in nizek logični nivo pri RS232.

RS485 [3] je izboljšana inačica RS232 standarda. Omogoča priklop več kot ene DTE naprave na isto vodilo, pri katerem ima vsaka naprava svoj naslov.

Zaradi diferencialnega prenosa je v primerjavi z RS232 uporaben za prenose preko veliko daljših razdalj in v okolju, kjer je nevarnost šuma večja. Vsak podatek se prenaša s parom invertiranih signalov, vrednost se določi kot razlika njunih potencialov.

2.1.2 IEEE 802.11

IEEE 802.11 [1] je množica standardov, ki opisujejo računalniške komunikacije preko brezžičnih omrežij. Uporablja kombinacije DSSS in OFDM modulacijskih tehnik na frekvenčnih območjih 2.4 GHz, 3.7 GHz in 5 GHz (tabela 1). Različice protokolov so označene s črkami abecede na koncu imena.

802.11 Protokol	Frekvenčno območje (GHz)	Pasovna širina (MHz)	Hitrost (Mbit / s)	Modulacija
-	2.4	20	1, 2	DSSS
A	5	20	6, 9, 12, 18, 24, 36, 48, 54	OFDM
	3.7			
B	2.4	20	1, 2, 5.5, 11	DSSS
G	2.4	20	1, 2, 6, 9, 12, 18, 24, 36, 48, 54	OFDM, DSSS
N	2.4	20	7.2, 14.4, 21.7, 28.9, 43.3, 57.8, 65, 72.2	OFDM
	5	40	15, 30, 45, 60, 90, 120, 135, 150	

Tabela 1: Pregled pogostejših protokolov družine IEEE 802.11.

S frekvenčnim pasom imamo določeno območje, v katerem si naprave izmenjujejo podatke. Problem nastopi, kadar več uporabnikov uporablja isto frekvenčno območje. V tem primeru lahko pride do motenj signala. Da se temu izognemo, je potrebno signal obdelati z eno od sledečih modulacij:

DSSS modulacija:

- signal preko celotnega spektra zmnožimo s naključnim zaporedjem -1 in 1 (šumom) pri veliko višji frekvenci,
- sprejemnik uporabi isto naključno zaporedje za zgradbo osnovnega signala.

OFDM modulacija:

- signal razstavimo na več med seboj ortogonalnih signalov in jih vzporedno pošljemo preko sosednjih podkanalov,
- sprejemnik ortogonalne signale združi skupaj v originalni signal.

802.11b in 802.11g sta danes med najbolj razširjenimi WLAN protokoli.

2.2 Prikazovalni del

2.2.1 Avisaro modul

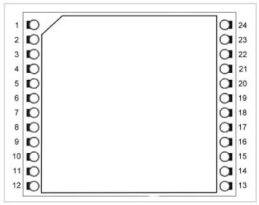
Za vspostavitev povezave s prikazovalnim delom je bil izbran modul »Avisaro WLAN Module 2.0« [4] istoimenskega nemškega proizvajalca. Modul ima 24 nožic, ki jih je možno poljubno nastavljati z BASIC-u podobnim programskim jezikom. Vgrajene ima notranjo anteno in čip za povezavo preko IEEE 802.11 standarda. Nastavi se ga lahko tudi z eno od šestih v naprej pripravljenih nastavitev:

Ime nastavitve	Aktivni vmesniki	Opis
RS232-1	RS232, V/I nožice za preverjanje statusa in kontrolo	RS232 vmesnik deluje v polnem načinu, aktivne so vse kontrolne linije
RS232-2	2x RS232	Dva RS232 vmesnika sta aktivna
SPI-1	SPI, V/I nožice za preverjanje statusa in kontrolo	SPI vmesnik je aktiven in deluje v načinu suženj
I2C-1	I2C	I2C vmesnik je aktiven in deluje v načinu suženj
CAN-1	CAN, V/I nožice za preverjanje statusa in kontrolo	CAN povezava je aktivna
CAN-2	2x CAN	Dve CAN povezavi sta aktivni

Tabela 2: Seznam nastavitev na Avisaro WLAN modulu 2.0.

Za potrebe športnega semaforja je bila izbrana RS232-1 nastavitev, saj omogoča premoščanje signalov med IEEE 802.11 in RS232 standardom (slika 4).

No	I/O	Name
1	P	VBAT
2	O	GPIO (LED1)
3	O	GPIO (LED2)
4	I	GPIO (KEY)
5	I	DCD
6	I	DSR
7	O	DTR
8	I	RING
9	O	TXD
10	I	RXD
11	I	CTS
12	O	RTS



Name	I/O	No
VCC (3.3V)	P	24
n.c.	-	23
n.c.	-	22
n.c.	-	21
n.c.	-	20
n.c.	-	19
n.c.	-	18
n.c.	-	17
GPIO (open)	-	16
GPIO (open)	-	15
Reset	I	14
GND	P	13

Slika 4: Avisaro modul v RS232-1 načinu.

Vsaka nastavitev določi pomen nožicam Avisaro modula. Za potrebe RS232 povezave so bile določene nožice 5 do 12.

Nožici 13 in 24 sta v vseh načinih vedno namenjeni za električno napajanje Avisaro modula z napajalno napetostjo Vcc = 3.3V.

Po izbiri nastavitve je potrebno Avisaro modulu podati pravila, kako ravnati pri sprejemu signala. Na voljo imamo več načinov:

- vmesnik za ukaze nam omogoča pisanje in pošiljanje ukazov v ASCII formatu neposredno preko podatkovnih vmesnikov RS232, RS485, SPI, I2C in CAN;
- za komunikacijo z ostalimi programabilnimi napravami in mikrokrmilniki ima možnost uporabe binarnih paketov. Omogoča dostop do več datotek ali TCP povezav hkrati;
- v programskem jeziku BASIC se lahko ukaze združi v krajši program, ki se shrani v notranjem flash pomnilniku. Za pisanje programa zadostuje tekstovni urejevalnik brez naprednih funkcij. Program je možno nastaviti, da se ob vsakem zagonu samodejno zažene.

Ker se pravila za premoščanje signalov med IEEE 802.11 in RS232 med samim delovanjem ne bodo spreminjala, je bila uporabljena rešitev s programom v jeziku BASIC. Program je dodan kot dodatek besedilu (Dodatek B).

Ko je program napisan, se lahko v Avisaro modul prenese na več načinov:

- pred zagonom modula nastavitve shranimo na spominsko kartico SD v datoteko autorun.txt. Ta način deluje tudi, če so vsi ostali dostopi blokirani;
- do nastavitvenega vmesnika lahko pridemo preko WLAN ali LAN mrežne povezave. V brskalnik vpišemo IP naslov modula in se prijavimo z uporabniškim imenom in geslom;
- vse ukaze lahko vnesemo preko podatkovnih vmesnikov (RS232, SPI, CAN …). Ukazi so lahko bodisi v ASCII ali dvojiško paketnem formatu.

Za vnos programa je bil uporabljen spletni brskalnik, dostop se je izvršil preko IEEE 802.11 povezave. Naloženi program je bil nastavljen kot privzeta izbira ob zagonu Avisaro modula.

Modul je po želji razširljiv z dodatnim modulom za baterijsko napajanje ali razširitev spominskega prostora s kartico SD.

2.2.2 Programiranje Avisaro modula

Za upravljanje manjših programov in dodatnih funkcij se lahko uporabi BASIC-u podoben programski jezik. Program se lahko piše v poljubnem tekstovnem urejevalniku. Na izbiro je več različnih množic ukazov:

- ukazi za delo z zankami (FOR, IF, THEN, WHILE …),
- ukazi za delo z nizi in spremenljivkami (DIM X(Y), LEN(string), VAL(string) …),
- ukazi za pregled statusa strojne opreme in trenutnega časa (TIME, DATE, FREEMEM …),
- ukazi za delo z datotekami (GET, PUT, OPEN, CLOSE …),
- ukazi za delo z omrežjem (CONNECT, LISTEN, RESOLVE …),
- ukazi za delo z V/I (INMODE, INPUT, SETLEDS, …),
- ukazi za uporabo parametra kot ukaza znotraj konzole (EXEC).

Pomnilniške lokacije se naslavljajo z naslovi imenovanimi »handles« (tabela 1). Preko njih se programu podaja naslove spremenljivk v pomnilniku. Le-ti so podani kot parametri ukazov, z njimi se posledično določi izvor oz. ponor podatkov.

Program v Avisaro modul vnesemo z ukazom »load«. Kot parameter ukazu load lahko določimo program, ki se nahaja na SD spominski kartici. Za pregled programa, ki se trenutno nahaja v pomnilniku, uporabimo ukaz »list«. Program je potrebno po vnosu v Avisaro modul zagnati. V ta namen uporabimo ukaz »run«. Za vklop oz. izklop samodejnega zagona programa ob resetu ukazu »run« dodamo parameter »auto« oz. »manual«.

Handle	Tip vmesnika	Opis
201 do 300	UDP	Nastavitve UDP kanala
101 do 200	TCP	Nastavitve TCP kanala
1 do 100	Datoteka	Nastavitve datotek na SD spominski kartici
-2	Podatkovni vmesnik	Sinhron način prenosa
-3	Podatkovni vmesnik	Asinhron način prenosa
-4	RS232	Pomožna RS232 vrata
-100 do -105	WEB	Šest spremenljivk za prikaz informacije na web strani
-201 do -224	V/I nožice	Digitalni V/I na modulu

Tabela 3: Pomnilniške lokacije, do katerih lahko dostopamo pri programiranju Avisaro modula.

Podroben seznam in opis ukazov je v priročniku programskega jezika.

2.2.3 Atmel AT8951 in upravljanje z LED vezji

Za upravljanje LED vezij skrbi mikroprocesor Atmel AT8951 [5] (slika A.2), ki je realiziran kot samostojno vezje. Podatki se iz Avisaro modula prenesejo preko RS232/RS485 pretvornika do AT8951, ki poskrbi za preslikavo sprejetega ASCII znaka v zaporedje petih 8-bitnih nizov. Vsak niz predstavlja svoj stolpec na LED vezju. Matrike za preslikavo ASCII znaka v zaporedje 8-bitnih nizov so shranjeni znotraj SST27SF512 flash pomnilnika [11] velikosti 512 kbitov, ki se nahaja na istem vezju v neposredni bližini mikroprocesorja.

Podatki za izpis se interpretirajo znak za znakom. Vsak znak lahko predstavlja ukaz (označen z #) ali številko. V primeru, da interpreter pride do znaka »#«, skoči v proceduro za izvajanje

ukazov. V nasprotnem primeru pogleda v preslikovalno tabelo bajtov, kjer vsak bajt predstavlja en stolpec na LED vezju. Vsi znaki se predhodno zapišejo v predpomnilnik, od koder se nato z ukazom po stolpcih izpišejo na kartico.

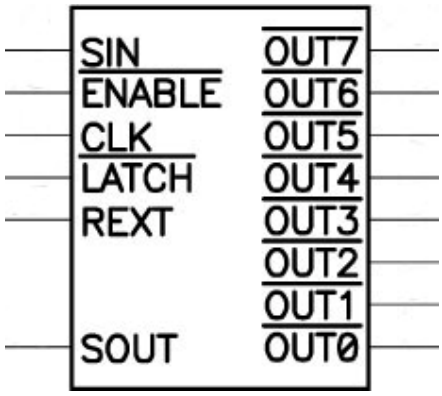
Pri vsaki preslikavi se binarni nizi pošljejo zaporedno na vhod prvega LED vezja. Zraven informacije o znaku se pošljejo še signal za vklop izhodov OE, signal za pomikalni register strobe in urin signal CLK, ki so potrebni za kontrolo izpisa na LED vezju.

2.2.4 LED vezja

Prikazovalni del sistema je sestavljen iz zaporedno vezanih LED vezij z ločljivostjo 5x7 pik in sposobnostjo prikazovanja alfanumeričnih znakov (slika A.1). Zaradi boljše vidljivosti vsak piksel tvorijo štiri zaporedno vezane diode.

Več LED vezij je med seboj zaporedno povezanih, izhod vsakega vezja je povezan z vhodom naslednjega. To omogoča poveza izhoda SOUT na vhod SIN. S tem se doseže, da je za izpis potrebna le ena podatkovna linija.

Vsako LED vezje ima za vsak stolpec svoj TB62705CFG [10] pomikalni register (slika 5). Na vhod SIN sprejema zaporedno poslane podatke. Z uporabo signala strobe, vezanega na vhod LATCH, se zaporedni bit prenese na vzporedne izhode OUT0–OUT7. Prenosi se izvajajo sinhrono ob upoštevanju signala CLK. Signal OE, vezan na vhod ENABLE, sproži osvežitev stolpca na LED vezju.

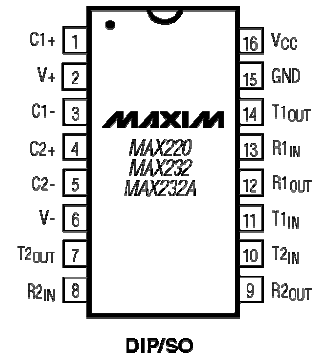


Slika 5: Pomikalni register za vklop LED diod v stolpcu.

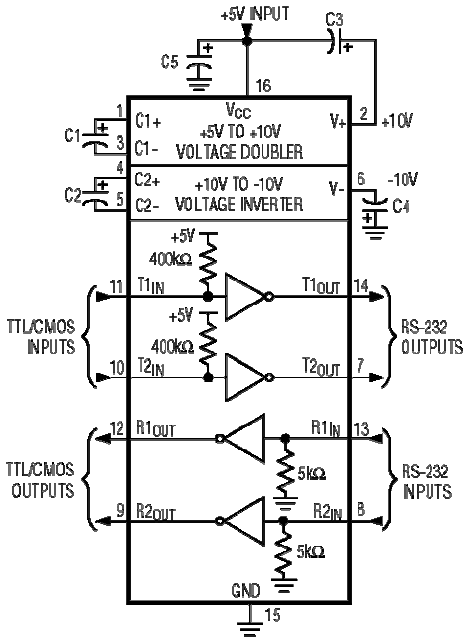
2.2.5 MAX232 in napetostni nivoji

Za povezavo Avisaro modula z LED vezjem je potrebno uskladiti napetostne nivoje signalov. To je potrebno, ker Avisaro modul deluje s signali v TTL napetostnem območju, RS232/RS485 pretvornik pa sprejema/oddaja podatke v RS232 napetostnem območju. Nizek signal pri TTL-u na območju od 0 do 0.8V se preslika na območje med 3V in 25V, visok signal med 2V in 5V se preslika na območje med -3V do -25V.

Za pretvorbo skrbi čip MAX232. Na nožice Tx_{in} in Tx_{out} je pripeljan TTL signal, Rx_{in} in Rx_{out} so namenjene priklopu RS232 signala. Funkcijo MAX232 natančno določajo tantal kondenzatorji, s katerimi lahko napetost podvojimo oz. invertiramo. Glede na tabelo (slika 6) so bili izbrani kondenzatorji s kapaciteto 1µF.



CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	0.047	0.33	0.33	0.33	0.33
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1



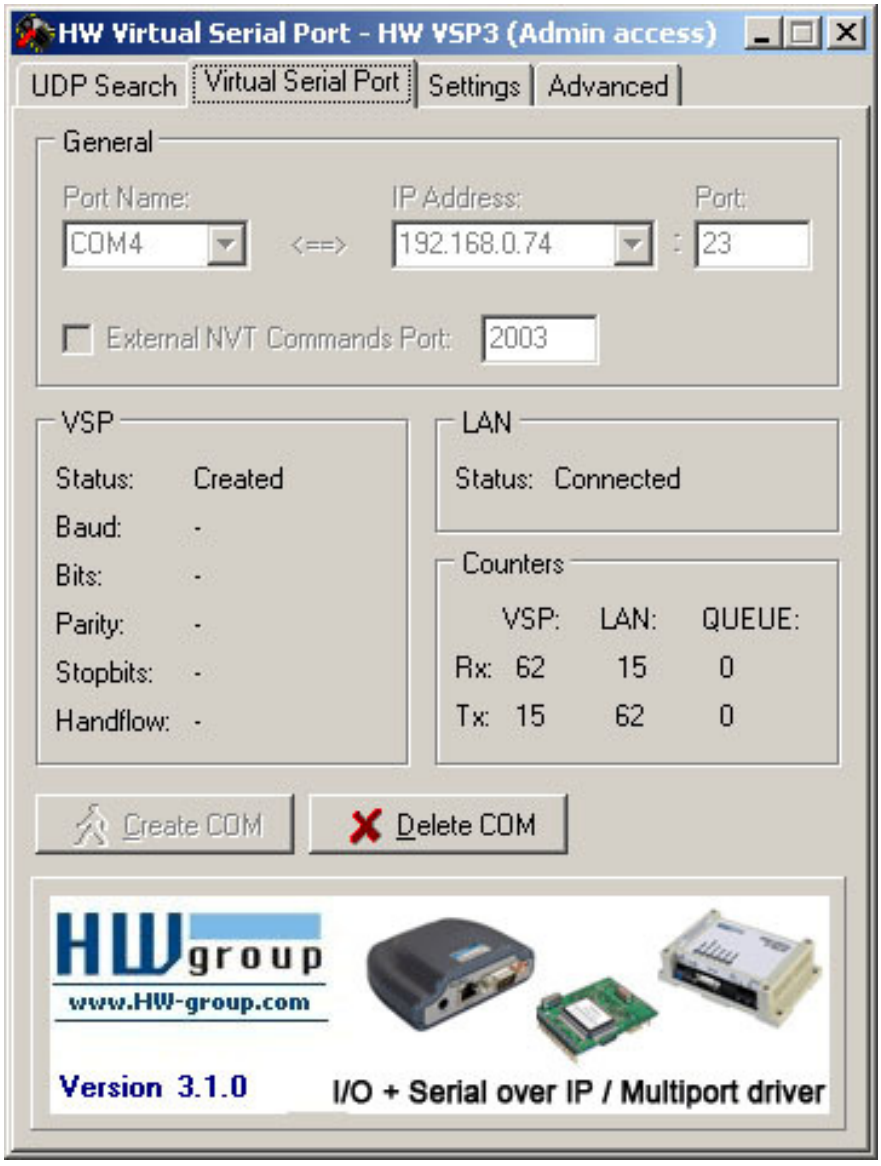
Slika 6: MAX232 čip za preslikavo napetostnih nivojev.

Prednost Avisaro modula je programabilnost. Modul je nastavljen v RS232-1 načinu, kar pomeni, da so nožice 5–12 programirane za RS232 povezavo (slika 4). Za povezavo je uporabljen oddajni signal TxD, nastavljen kot izhod, in sprejemni signal RxD, nastavljen kot vhod. Ozemljitev povezave in ozemljitev napajanja sta povezani na nožico 13.

2.3 Upravljalni del

Za pošiljanje podatkov na Avisaro modul je uporabljen PC računalnik z možnostjo priklopa na IEEE 802.11 WLAN omrežje in operacijskim sistemom MS Windows. Povezava se vzpostavi med mrežnim adapterjem v računalniku in Avisaro modulom (poglavje 2.1.2).

Za nastavitev povezave preko brezžičnega omrežja je bil izbran HW Virtual Serial Port WLAN emulator (slika 7). Ta emulator omogoča, da je fizična povezava preko brezžičnega medija logično videti, kot da se povezuje preko RS232 povezave. V zavihku »Virtual Serial Port« se vnese IP naslov in vrata. Med modulom in računalnikom se vzpostavi brezžična povezava, preko katere se emulirajo RS232 signali.



Slika 7: Nastavitve emulacije RS232 povezave preko IEEE 802.11 omrežja.

3 Programska oprema

Programski vmesnik je narejen v programskem jeziku C#. Njegova naloga je enostavno upravljanje z izpisom LED vezij in pregled nad vnesenimi podatki iz upravljalnega dela. Zato je tudi vizualno narejen v stilu končnega produkta.

Vmesnik upravlja z LED vezjem preko vzpostavljene IEEE 802.11 povezave. Mikroprocesorju pošilja informacijo kodirano po ASCII standardu. Za prenos se uporablja enostaven komunikacijski protokol, ki informaciji za izpis na LED vezje doda 8-bitni naslov naprave in 16-bitno kontrolno vsoto.

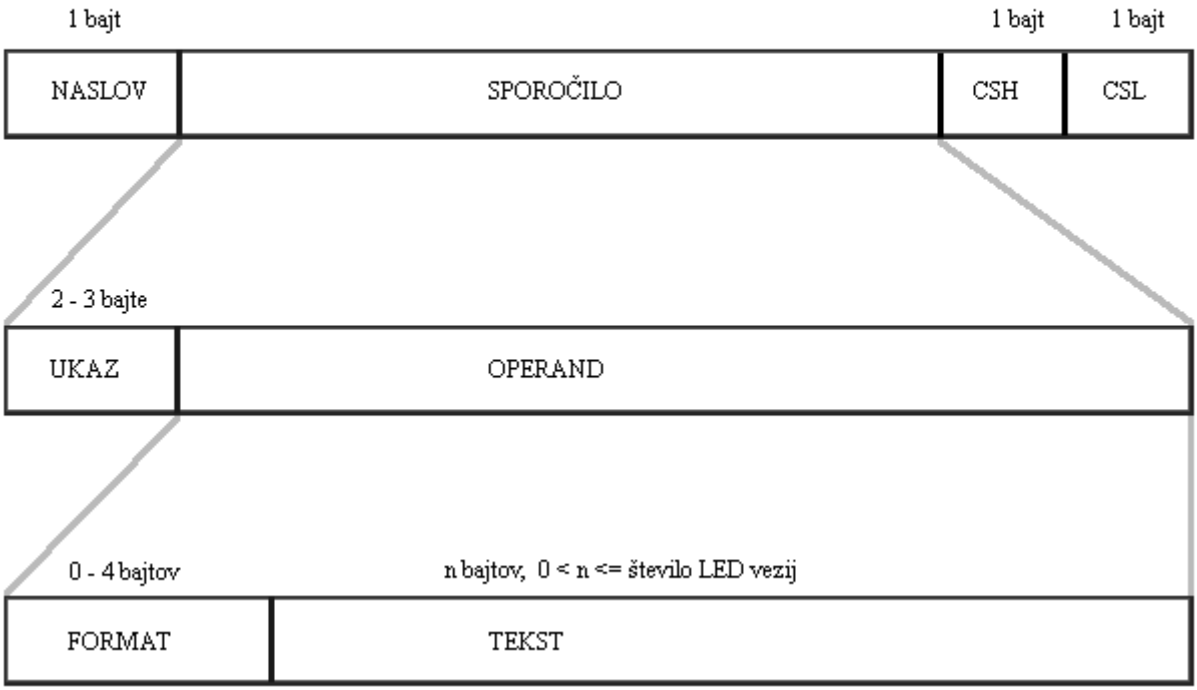
3.1 Komunikacijski protokol

Komunikacijski protokol določa sestavo sporočil, ki se pošiljajo med Atmelovim mikroprocesorjem in uporabniškim vmesnikom na PC računalniku. Protokol je dvosmeren, iz uporabniškega vmesnika se pošilja niz za izpis kodiran po ASCII standardu, sprejema pa se informacija o uspešnosti prenosa.

Protokol pošilja podatke preko RS232 povezave na nivoju bajta. To omogoča pošiljanje 128 različnih 7-bitnih ASCII znakov. Osmi bit podatkovnega bajta (MSB) ima poseben pomen; z njim povemo napravi, kdaj se pošilja naslovni bajt. Naslovni bajt nam pove, na katero napravo so sporočila namenjena, kar omogoča priklop do 128 različnih naprav.

Več paketov zapovrstjo sestavlja sporočilo (grafični prikaz na sliki 8). Sporočilo je sestavljeno iz:

- naslovnega bajta sporočila, namenjenega naslavljanju prikazovalnika. Da gre za naslovni bajt nam pove MSB, ki je enak 1;
- 2-3 bajte dolgega kontrolnega ukaza mikroprocesorju, kaj narediti s poslano vsebino. Pošiljanje ukaza najavi prvi bajt, ki je vedno enak »0x3A«;
- 0-2 bajta namenjena ukazu za določanje formata. Prvi bajt ima vedno vrednost »0x23«;
- niza informacije, kodirane po ASCII standardu, ki se prikaže na LED vezju;
- 16-bitna kontrolna vsota vseh bajtov za odpravo napak med prenosom sporočila. Sestavljena je iz zgornjih 8 bitov (CSH) in spodnjih 8 bitov (CSL).



Slika 8: Zgradba komunikacijskega sporočila. Prvi bajt nam pove naslov, sledi mu sporočilo in na koncu še kontrolna vsota razdeljena na zgornjih (CSH – check sum high) in spodnjih 8 bitov (CSL – check sum low).

3.1.1 Kontrolni ukazi

Uporabljajo se za upravljanje strojne opreme. Vsi kontrolni ukazi se začnejo z znakom 0x3A oz. »:« po ASCII standardu.

Primer kontrolnih ukazov:

a.) inicializacija teksta

0x3A 0x38 0x30 (ASCII :80)

Ukaz nastavi kazalec na prvo mesto v medpomnilniku LED vezja. Kazalec se lahko postavi na poljubno mesto, to pa dosežemo s spreminjanjem zadnjega bajta v ukazu.

b.) vpis ASCII znaka v pomikalni register LED vezja

0x3A 0x39 <niz kodiran po ASCII standardu in vrednostjo najtežjega bita 0>

(ASCII :9<niz>)

Ukaz vpiše niz v pomikalni register LED vezja, vendar ne sproži izpisa.

c.) prenos ASCII znaka iz pomikalnega registra v LED vezje

0x3A 0x3A

(ASCII »::«)

Ukaz sproži signal OE in prenos podatkov iz vzporednih izhodov pomikalnega registra na LED diode.

3.1.2 Ukazi za določanje formata

Uporabljajo se za oblikovanje izpisa znakov na LED vezjih. Ti ukazi se začnejo z znakom 0x23 oz. # po ASCII standardu, sledi koda ukaza (3.1.1.b).

Primer podatkovnih ukazov:

a.) #C

Ukaz povzroči brisanje vseh registrov LED vezja.

b.) #M

Ukaz povzroči takojšnji izpis teksta na LED vezje, besedilo se centrira na sredino. V primeru, da je besedilo daljše, kot ga lahko prikaže, se izpiše od prvega znaka naprej.

c.) #Si

Vezje ima možnost prikaza informacije v tekočem načinu. To pomeni, da se vsak n-ti stolpec na LED vezju preslika v (n+1)-ti stolpec. Z nastavljanjem parametra i določimo časovno periodo preslikave stolpcev in s tem hitrost premikanja besedila v tekočem načinu. Parameter i lahko zavzame vrednosti od 1 (najvišja hitrost) do 9 (najnižja hitrost).

3.1.3 Kontrolna vsota sporočila

Vsako sporočilo ima na koncu izračunano še 16-bitno kontrolno vsoto vseh 8-bitnih paketov. Kontrolna vsota je 16-bitna vsota vseh bajtov sporočila brez upoštevanja prenosa. Vsota se pošlje v dveh delih, najprej se pošlje zgornjih 8 bitov (CSH), nato še spodnjih 8 bitov (CSL). Pred pošiljanjem kontrolne vsote nad obema deloma naredimo še sledečo operacijo:

```
int ModifiedCSL = (CSL & 127) | 64;
int ModifiedCSH = (CSH & 127) | 64;
```

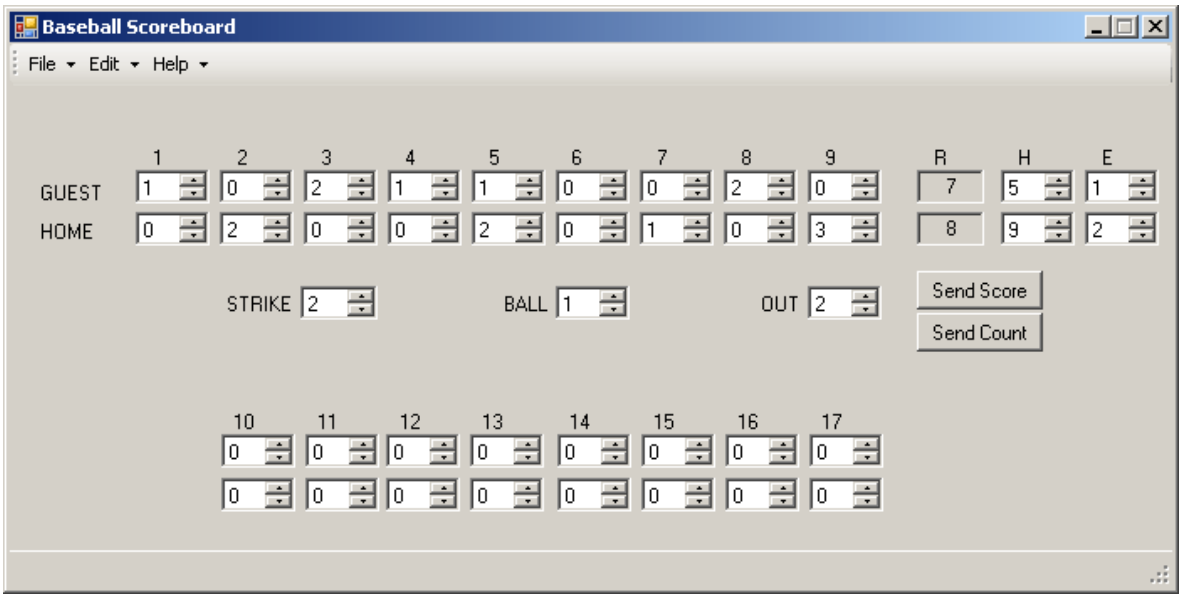
S tem preprečimo, da se MSB postavi na 1 (določa naslovni bajt) in hkrati preprečimo znak 0x00 (zaključitveni bajt paketa – NULL). Na koncu se kot zadnji paket vedno pošlje še zaključitveni bajt sporočila 0x00.

Primer paketa za incializacijo besedila:

8 bitni paket	ASCII znak	pomen
0xD6		7-bitni naslov naprave, bit7 = 0
0x3A	:	znak za začetek ukaza
0x38	8	ukaz inicializacije medpomnilnika
0x30	0	mesto, na katerega se postavi kazalec
0x41	A	zgornjih 8 bitov kontrolne vsote
0x78	x	spodnjih 8 bitov kontrolne vsote

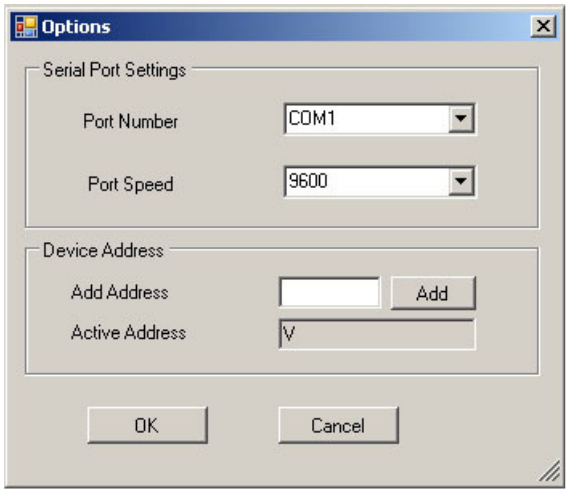
3.2 Uporabniški vmesnik

Preko vmesnika je možno nastaviti vrednost posameznega LED vezja, s pritiskom na gumb »send score« sprožimo prenos in izpis podatkov. Pri tem se v programu iz vnesenih podatkov sestavi komunikacijsko sporočilo. Podatki se kot niz ASCII znakov pošiljajo preko RS232 vrat v obliki sporočil (poglavje 3.1).



Slika 9: Glavno okno programa za upravljanje semaforja.

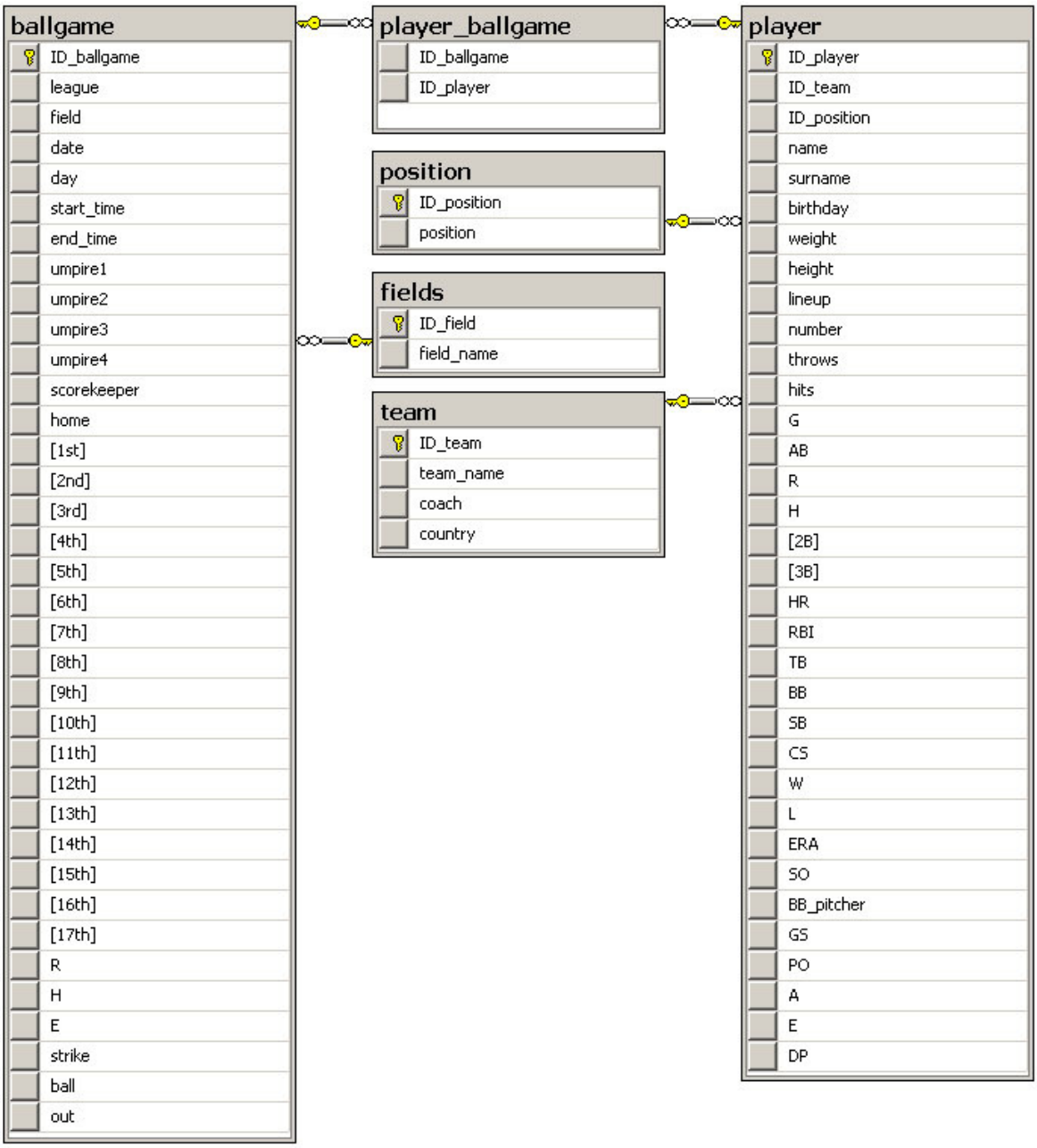
Komunikacijske nastavitve povezave se lahko nastavijo v meniju Edit -> Options, kjer program samodejno poišče razpoložljiva RS232 vrata. Hitrost prenosa podatkov je 9600 bitov/sekundo. Omenjene vrednosti se zaradi lažjega upravljanja samodejno privzamejo.



Slika 10: Nastavitve parametrov RS232 prenosa.

3.3 Pogled v bazo in dostop do podatkov

Baseball je igra, kjer se veliko uporabljajo verjetnost in statistični podatki. Razlog za to je zahteva po medsebojni primerjavi igralcev. Zato ima program za upravljanje s športnim semaforjem dodatno možnost shranjevanja in kasnejše uporabe rezultatov ter ostalih statističnih podatkov v lokalno podatkovno bazo (slika 12). Digitalna oblika statističnih podatkov omogoča lažji prikaz le-teh na spletni strani kluba.



Slika 11: Struktura baze rezultata in statističnih podatkov.

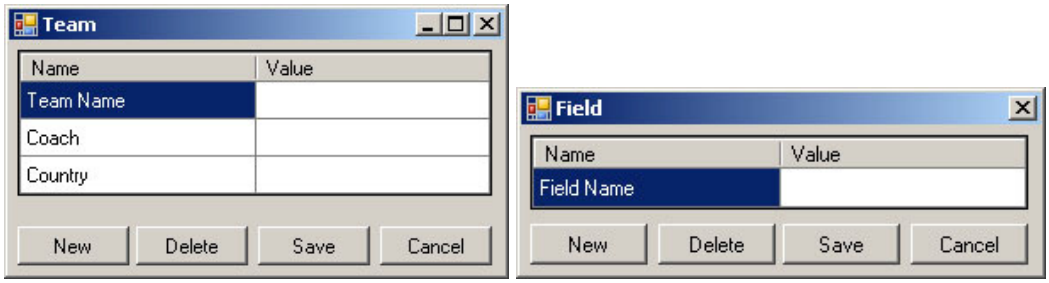
Podatki se shranjujejo v dve glavni tabeli. Prva tabela, imenovana »player«, shranjuje podatke vezane na igralca. Ti se delijo na osebne podatke in statistiko uspešnosti na različnih področjih igre, kot so število doseženih točk, število odigranih tekem, odstotek odbitih žog, itn.

Druga tabela »ballgame« vsebuje podatke o tekmi, kdaj in kje je bila odigrana, kdo so bili sodniki in kakšen je bil rezultat tekme.

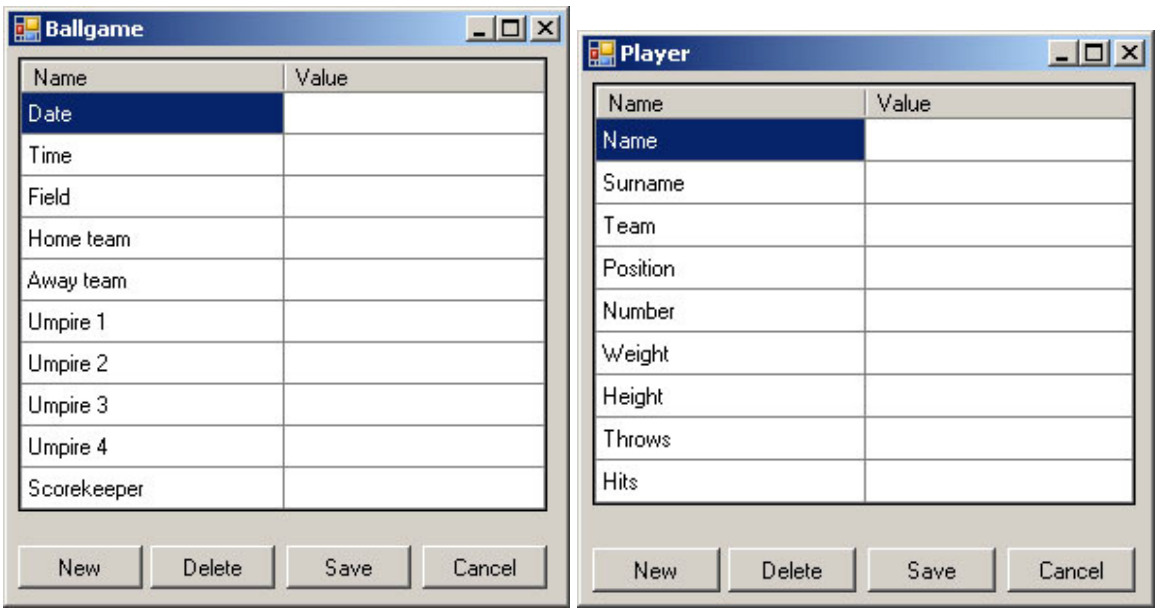
Tabela »ballgame« vsebuje zapise o različnih tekmah. Vsak zapis ima več povezav na tabelo »player«, saj v vsakem moštvu sodeluje vsaj devet igralcev. Ker vsak igralec tekom sezone igra na različnih tekmah, ima tudi tabela »player« lahko več povezav na tabelo »ballgame«. Za rešitve tega problema vpeljemo vmesno tabelo, ki med seboj povezuje različne igralce in tekme.

Poleg treh omenjenih tabel so v bazi še tabele zapisov, ki delujejo kot šifranti. Vsak šifrant ima listo dovoljenih zapisov, ki so preko tujega ključa povezani z zapisom v glavni tabeli. Tako je potrebno npr. pred vnosom igralca v bazo najprej vnesti podatke o moštvu, za katerega nastopa. Pri vnosu igralčevih podatkov se pri pogoju, da v tabeli »team« ne obstaja izbrano moštvo, javi napaka. To onemogoča vnos igralcev brez podatka o klubu (ID_team) in igralni poziciji (ID_position) ali tekme brez podatka o igrišču (ID_fields). Dodatna prednost je samo en zapis za določeno igrišče oz. moštvo v bazi, vsi ostali objekti se nanj sklicujejo. Tako odpravimo večkratni vnos in potencialne napake znotraj baze.

Novi vnosi se v bazo lahko vnašajo preko menija Edit. Pri vnosu je potrebno zaradi hierarhije paziti na vrstni red (slika 13 in slika 14).



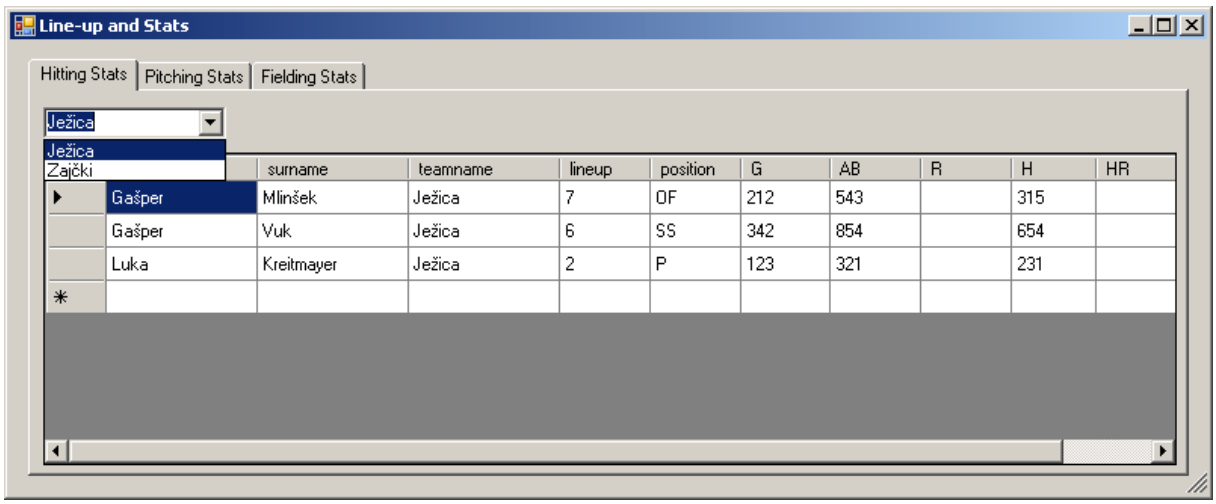
Slika 12: Vnos podatkov o moštvu (levo) in igrišču (desno) v podatkovno bazo.



Slika 13: Vnos podatkov o igri (levo) in igralcu (desno) v podatkovno bazo. Tabeli se navezujeta na prejšnji tabeli o igrišču in moštvu.

Statistični podatki se pri baseballu delijo v tri večje skupine: napadalni, obrambni in metalčevi. Program ima možnost prikaza vsake posebej. Tabela za prikaz znotraj programa pri prenosu podatke uredi s pomočjo SQL strukturiranega stavka. Primer SQL stavka za izpis statistik metalca:

SELECT name, surname, G, W, L, ERA, SO, GS
FROM scoreboard.dbo.player
WHERE ID_team = (
SELECT ID_team
FROM scoreboard.dbo.team
WHERE team_name='Ježica'
AND ID_position = 1)";



Slika 14: Dostop do statističnih podatkov igralcev celotnega moštva preko grafičnega vmesnika.

Podatki se v tabeli lahko poljubno urejajo, vsaka sprememba se odraža neposredno na bazi. Da pri vpisu v bazo ne bi prišlo do nepopolnega vnosa oz. nekonsistentnosti, so vsi dostopi opravljeni s pomočjo t.i. »store procedure«.

3.3.1 Store procedura

Pri vnosu podatkov v bazo je potrebno paziti na pravilni zapis vseh podatkov. Če pride do napake med zapisom, je potrebno postopek ponoviti. Pri tem se v bazo ne smejo vpisati delni oz. nepopolni rezultati. V ta namen se uporablja zaporedje ukazov, ki se ne smejo prekiniti, imenovano »store procedura«.

Store procedura je sestavljena iz vhodnih parametrov, izhodnega rezultata in osrednjega dela procedure. Osrednji del je sestavljen iz zaporedja SQL ukazov in odločitvenih stavkov, ki določajo operacije nad podatki. V primeru spremembe podatka znotraj baze se ponovno osveži celoten zapis za določenega igralca.

Parametri se predhodno nastavijo preko grafičnega vmesnika. Tip in dolžina parametra v programu se nastavi tako, da se ujema z zapisom v podatkovni bazi.

```
SqlConnection sqlcon = new SqlConnection(CString.GlobalConnString);
SqlCommand insert = new SqlCommand("sp_insert_player", sqlcon);

insert.CommandType = CommandType.StoredProcedure;

insert.Parameters.Add("@name", SqlDbType.NVarChar, 30);
insert.Parameters["@name"].Direction = ParameterDirection.Input;
insert.Parameters["@name"].Value =
(string)dataGridView1.Rows[0].Cells[1].Value;

insert.Parameters.Add("@surname", SqlDbType.NVarChar, 30);
insert.Parameters["@surname"].Direction = ParameterDirection.Input;
insert.Parameters["@surname"].Value =
(string)dataGridView1.Rows[1].Cells[1].Value;

insert.Parameters.Add("@team_name", SqlDbType.NVarChar, 30);
insert.Parameters["@team_name"].Direction = ParameterDirection.Input;
insert.Parameters["@team_name"].Value =
(string)dataGridView1.Rows[2].Cells[1].Value;

insert.Parameters.Add("@position", SqlDbType.NVarChar, 2);
insert.Parameters["@position"].Direction = ParameterDirection.Input;
insert.Parameters["@position"].Value =
(string)dataGridView1.Rows[3].Cells[1].Value;

insert.Parameters.Add("@number", SqlDbType.Int, 2);
insert.Parameters["@number"].Direction = ParameterDirection.Input;
insert.Parameters["@number"].Value = dataGridView1.Rows[4].Cells[1].Value;
```

Slika 15: Nastavitev parametrov pred zagonom store procedure.

Store procedura se kliče iz programa z ukazom `ExecuteNonQuery()` (primer store procedure v Dodatku C). Procedura poskrbi, da se vnos podatkov v bazo opravi kot zaključena celota.

4 Sklepne ugotovitve

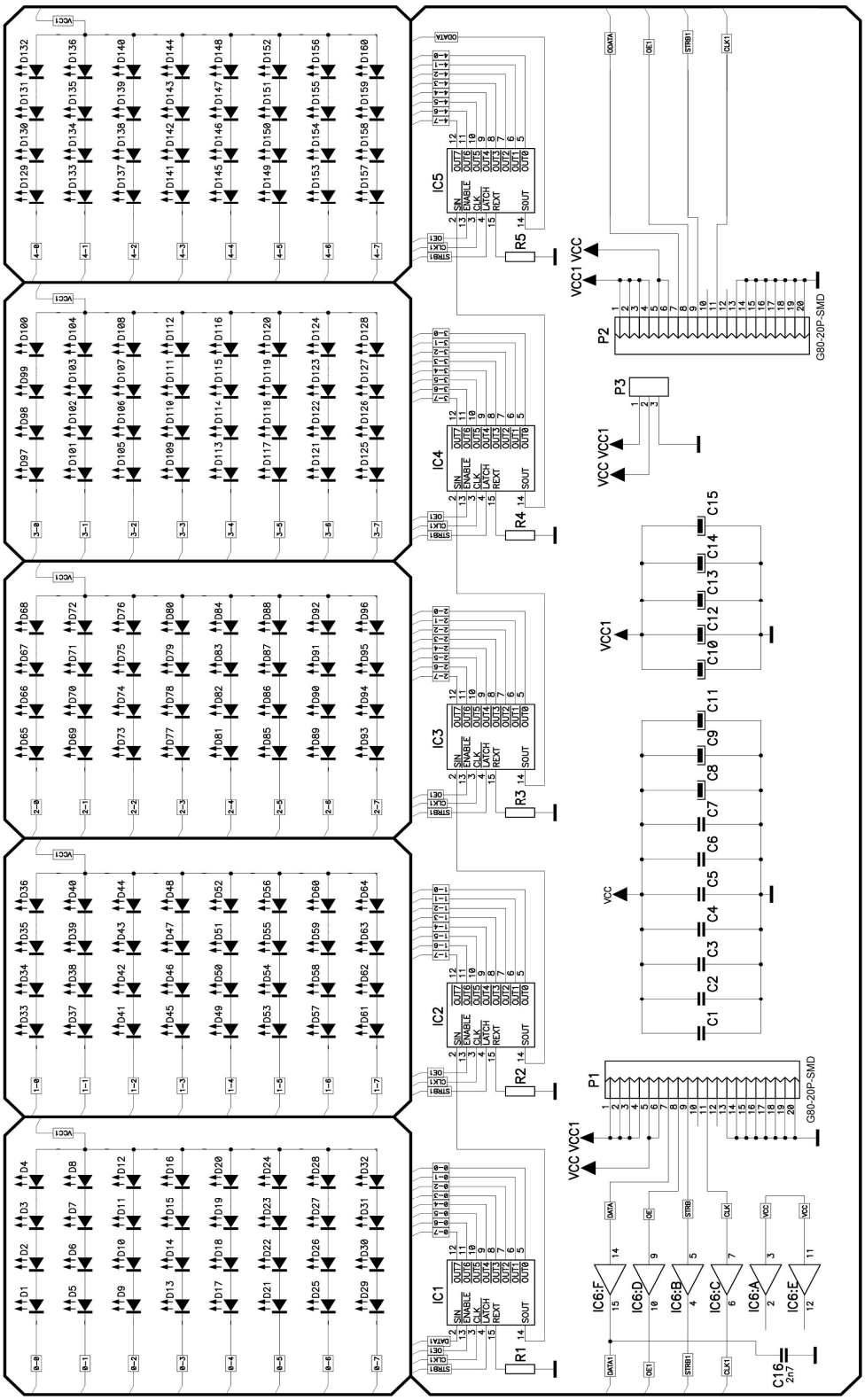
Končni izdelek izpolnjuje vse uvodoma podane zahteve. Semafor je dovolj velik, da je možno rezultat prebrati iz katere koli pozicije na igrišču. Upravljanje semaforja je možno brezžično iz bližnje okolice z uporabo prenosnega računalnika preko WLAN povezave. Vsi podatki se tekom tekme shranjujejo v podatkovno bazo, kar omogoča enostaven dostop in urejanje večje količine statističnih podatkov. Podatki so preko baze na razpolago tudi za realnočasovni prikaz na domači spletni strani kluba.

Pri izdelavi projekta sem pridobil veliko praktičnih izkušenj, pojavilo pa se je tudi nekaj idej, kako bi bilo možno športni semafor še izboljšati. Med glavnimi idejami je uporaba novejših LED za prikaz večje palete barv in sestava LED vezij z bistveno višjo ločljivostjo. Prav tako sem med izdelavo spoznal novejše mikrokrmilnike, katerih prednost je vgrajeni in predvsem večji pomnilnik. To vse omogoča uporabo bistveno bolj kvalitetnega grafičnega prikaza.

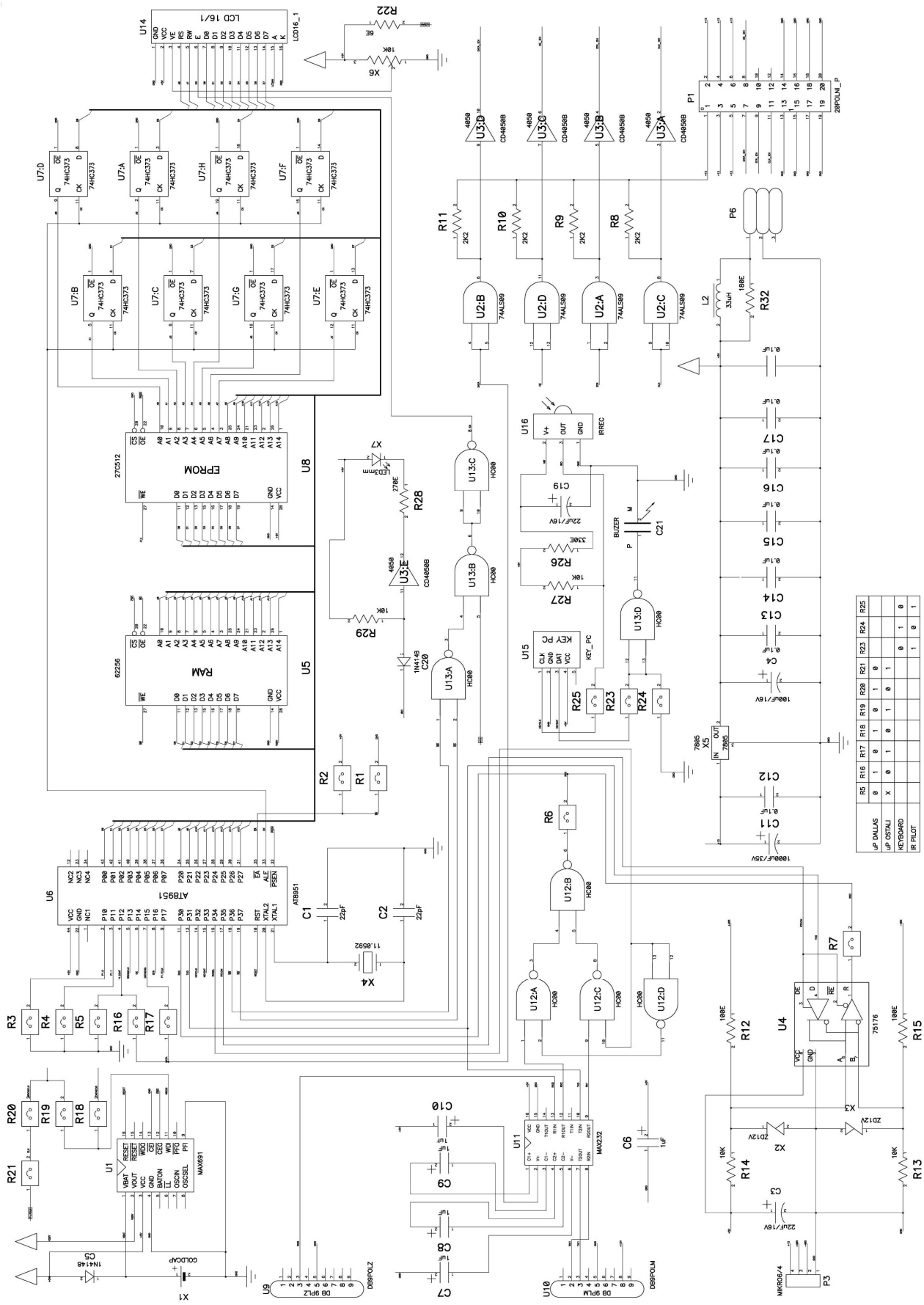
Po drugi strani so bili največji strošek pri projektu ravno veliko število LED in Avisarov WLAN modul. Končna cena je tako predvsem po zaslugi velikega števila LED presegla nekaj tisoč evrov, kar tudi za baseball društvo predstavlja relativno visok strošek. To je bil tudi glavni razlog, da se je število LED vezij minimiziralo in omejilo le na prikaz najpomembnejše informacije.

Dodatek A

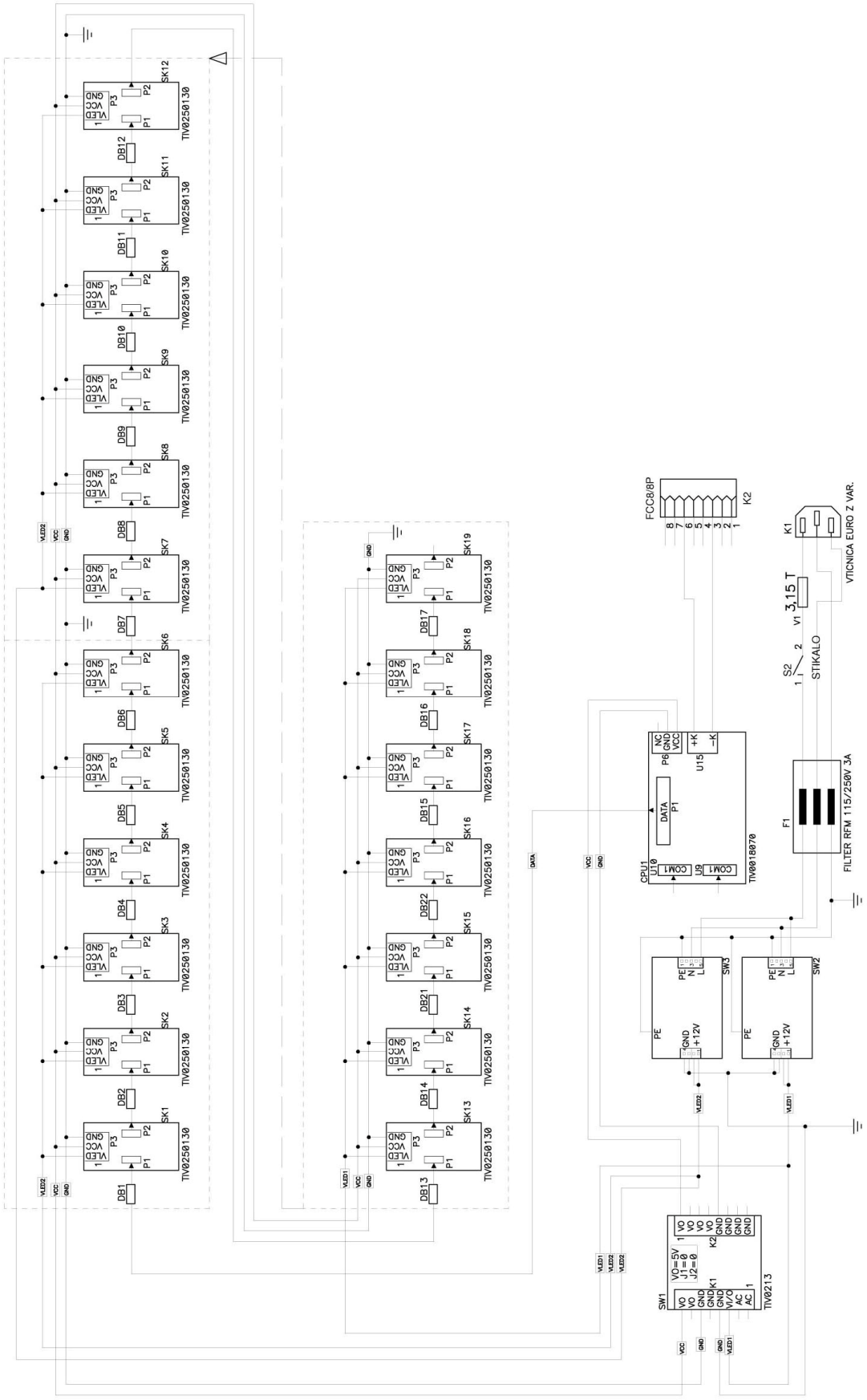
Električne sheme



Slika A.1: Električna shema LED vezja.



Slika A.2: Električna shema mikrokrmilniškega dela z mikrokrmilnikom AT8951.



Slika A.3: Shema povezave mikrokrmilniškega dela TIV0018070 z LED vezjem TIV0250130.

Dodatek B

BASIC program na Avisaro modulu

Vir: [4]

```
setleds 32+128

exec "stpseq +\097+v+i+s+"
sleep 100
exec "prompt"
sleep 100
exec "sched 0"
sleep 100

let n = 0

if ((KEYS & 1) = 1) then
  for n = 0 to 100
    if ((KEYS & 1) = 0) then
      goto STARTING
    end if
    setleds 255
    sleep 10
    setleds 0+128
    sleep 10
  next n
  exec "restart clear"
  sleep 100
  setleds 255
  do
    sleep 1
  loop
end if
STARTING:

sleep 400

DIM A(500)
DIM B(0)
let n = 0
let t = TIME
let f = 60
let m = 0
let u = 0
let y = 0

let x$ = "Connect to (IP, =0 for listen):"
```

```
put -100, x$, len(x$)
let x$ = "Connect to / Listen (Port):"
put -102, x$, len(x$)
let x$ = "Status (WR1 V14):"
put -104, x$, len(x$)

load 0, t$
put -101, t$, len(t$)


load 25, u
if (-1 = u) then
  let u = 23
  save 25, u
end if
let x$ = str$(u)
put -103, x$, len(x$)


TRY_CONNECT:

inmode -3
input A

if (t$ = str$(0)) then
  let x$ = "listening"
else
  let x$ = "try to connect"
end if

gosub load_web

if (t$ = str$(0)) then
  sleep 50
  listen 101, u, 0
  setleds 0+128
  sleep 500
  setleds 32+128
  sleep 50
else
  let y = RESOLV (t$)
  sleep 500
  connect 101, y, u, 5
  setleds 0+128
  sleep 50
  setleds 32+128
end if

let y = status(101)
```

```
if y = 9 then
   let x$ = "Connected"
   inmode 0
   gosub load_web


   REM Connected

   put -212,#1
   sleep 20

   exec "stream 101"

   sleep 30
   put -212,#0

   goto MAIN
else
   close 101
end if

goto TRY_CONNECT

MAIN:

setleds 48+128

if ((KEYS & 1) = 1) then
    close 101
    setleds 32
    gosub load_web
    goto TRY_CONNECT
end if

sleep 10

let y = status(101)
if (y <> 9) then
   close 101
   setleds 32
   goto TRY_CONNECT
end if

goto MAIN:

load_web:

put -105, x$, len(x$)

get -103, x$
```

```
if val(x$) <> u then
    let u = val(x$)
    save 25, u
end if

get -101, x$
if t$ <> x$ then
   let t$ = x$
   save 0, t$
end if

return
```

Dodatek C

Store procedura za konsistentni dostop do podatkovne baze

```
USE [scoreboard]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[sp_insert_player]
	@name varchar(30) = 0,
	@surname varchar(30) = 0,
	@team_name varchar(30) = 0,
	@position varchar(2) = 0,
	@number int = 0
AS
BEGIN
	SET NOCOUNT ON;
	DECLARE	@position_int		INT,
		@team_name_int		INT,
		@count			INT
	SET @position_int = (SELECT ID_position
				FROM scoreboard.dbo.position
				WHERE position = @position)
	SET @team_name_int = (SELECT ID_team
				FROM scoreboard.dbo.team
				WHERE team_name = @team_name)
	SET @count = (
	SELECT COUNT(*)
	FROM scoreboard.dbo.player
	WHERE name=@name
```

```
        AND surname=@surname
        AND ID_team=@team_name_int
        AND ID_position=@position_int
        AND number=@number)

        IF @count < 1
        INSERT INTO scoreboard.dbo.player ("name", "surname", "ID_team", "ID_position",
        "number") VALUES (@name, @surname, @team_name_int, @position_int,
        @number)
END
```

Literatura

[1] (03/2010) IEEE 802.11
http://en.wikipedia.org/wiki/IEEE_802.11

[2] (03/2010) RS232
http://en.wikipedia.org/wiki/RS-232

[3] (03/2010) RS485
http://en.wikipedia.org/wiki/EIA-485

[4] (03/2010) AVISARO WLAN Module (2.0) - Avisaro
http://www.avisaro.com/eng/html/wlan_module__2_0_.html

[5] (03/2010) AT80C32X2 - 8-bit Microcontroller 8 Kbytes ROM/OTP, ATMEL Corp.
http://pdf1.alldatasheet.com/datasheet-pdf/view/257066/ATMEL/AT80C32X2.html

[6] (03/2010) BS62LV256SIP55 - CMOS SRAM 32K X 8 bit - Brilliance Semiconductor
http://pdf1.alldatasheet.com/datasheet-pdf/view/201751/BSI/BS62LV256SIP55.html

[7] (03/2010) CD74HC373M - Texas Instruments
http://pdf1.alldatasheet.com/datasheet-pdf/view/27050/TI/CD74HC373M.html

[8] (03/2010) MAX691A - Maxim Integrated Products
http://pdf1.alldatasheet.com/datasheet-pdf/view/73729/MAXIM/MAX691A.html

[9] (03/2010) SN75176BP - Texas Instruments
http://pdf1.alldatasheet.com/datasheet-pdf/view/28198/TI/SN75176BP.html

[10] (03/2010) TB62705CPG, TB62705CFG, TB62705CFNG - Toshiba
http://www.marktechopto.com/pdfs/Toshiba/ACF368.pdf

[11] (03/2010) SST27SF512-70-3C-PG - Silicon Storage
http://pdf1.alldatasheet.com/datasheet-pdf/view/46512/SST/SST27SF512-70-3C-PG.html

[12] (03/2010) 54HC4050 - Maxwell
http://pdf1.alldatasheet.com/datasheet-pdf/view/97650/ETC/54HC4050.html

[13] (03/2010) LEA
http://www.lea.si/