

---

# **Razvoj Android aplikacija za elektroničare**



**Dogan Ibrahim**

Agencija Eho  
[www.infoelektronika.net](http://www.infoelektronika.net)

---

- Sva prava zadržana. Nijedan deo ove knjige ne sme biti reprodukovan u bilo kom materijalnom obliku, uključujući fotokopiranje ili slučajno ili nenamerno smeštanje na bilo koji elektronski medijum sa ili uz pomoć bilo kog elektronskog sredstva, bez pismenog odobrenja nosioca autorskih prava osim u skladu sa odredbama zakona o autorskim pravima, dizajnu i patentima iz 1988. godine ili pod uslovima izdatim od Copyright Licensing Agency Ltd, 90 Tottenham Court Road, London, England W1P 9HE. Prijave za pismene dozvole radi štampanja bilo kog dela ove publikacije upućuje se izdavaču ove knjige.
- Izjava: Autor i izdavač su uložili najveće napore da bi se obezbedila tačnost informacija sadržanih u ovoj knjizi. Autor i izdavač ne mogu da pretpostave neprijatnosti i ovom izjavom isključuju bilo kakvu odgovornost za bilo koju stranku koja bi imala gubitke ili štetu uzrokovanu greškama ili propustima u ovoj knjizi, bez obzira da li su greške ili propusti nastali usled nemara, nezgode ili bilo kog drugog razloga.

**ISBN 978-86-80134-22-2**

**Razvoj Android aplikacija za elektroničare**

Naslov originala: Android App Development for electronics designers

Izdavač originala: Elektor International Media B.V.

Autor: Dogan Ibrahim

Prevod: Voloda Pezo

Izdaje i štampa: Agencija Eho, Niš

e-mail: redakcija@infoelektronika.net

Tiraž: 500

Godina izdanja: 2018

CIP- Каталогизација у публикацији  
Народна библиотека Србије

004.451.9ANDROID  
004.42:004.451

ИБРАХИМ, Доган, 1954-  
Razvoj Android aplikacija za elektroničare /  
Dogan Ibrahim ; [prevod Voloda Pezo]. - Niš : Agencija Eho, 2018 (Niš : Agencija Eho). - 240 str. : ilustr. ; 24 cm

Prevod dela: Android App Development for electronics designers. - Tiraž 500.

ISBN 978-86-80134-22-2

a) Оперативни систем "Android" - Програмирање

COBISS.SR-ID 270174476

# SADRŽAJ

<b>PREDGOVOR .....</b>	<b>13</b>
<b>POGLAVLJE 1 • ANDROID MOBILNI TELEFONI .....</b>	<b>14</b>
1.1 Pregled .....	14
1.2 Generacije mobilnih telefona.....	14
1.2.1 Prva generacije (1G) .....	14
1.2.2 Druga generacija (2G) .....	14
1.2.3 Treća generacija (3G).....	15
1.2.4 Četvrta generacija (4G).....	15
1.2.5 Peta generacija (5G) .....	15
1.3 Verzije Androida .....	15
1.4 Verzije Android operativnog sistema .....	16
1.5 Rezime.....	17
<b>POGLAVLJE 2 • POČETAK.....</b>	<b>18</b>
2.1 Pregled .....	18
2.2 Instalacija probne verzije B4A .....	18
2.3 Instalacija standardne verzije B4A .....	23
2.4 Dobijanje pomoći .....	24
2.5 Rezime.....	24
<b>POGLAVLJE 3 • MOJ PRVI B4A PROGRAM .....</b>	<b>25</b>
3.1 Pregled .....	25
3.2 Rad sa okruženje B4A IDE .....	25
3.3 Prikaz poruke na Android mobilnom uređaju.....	26
3.4 Rezime.....	28
<b>POGLAVLJE 4 • MOJ DRUGI B4A PROGRAM .....</b>	<b>29</b>
4.1 Pregled .....	29
4.2 Rad sa Designer-om .....	29
4.3 Rezime.....	33

## **POGLAVLJE 5 • MOJ TREĆI B4A PROGRAM .....34**

5.1 Pregled .....	34
5.2 Jednostavan kalkulatorski program .....	34
5.3 Otklanjanje grešaka (debugging) .....	41
5.4 Upotreba USB veze .....	45
5.5 Rezime .....	47

## **POGLAVLJE 6 • NAPOMENE O JEZIKU B4A.....48**

6.3 Pregled .....	48
6.2 Komentari .....	48
6.3 Uvlačenje - zupčanje .....	48
6.4 Razlikovanje velikih slova i odvajanje izraza .....	49
6.5 Konstante .....	49
6.6. Promenljive .....	49
6.7 Nizovi .....	51
6.8 Liste .....	52
6.9 Maps .....	53
6.10 Matematički operatori .....	54
6.11 Logički operatori .....	54
6.12 Operatori odnosa (relational) .....	55
6.13 Promena toka programa .....	55
6.13.1 Uslovni iskazi .....	55
6.13.2 Iteracije .....	56
6.14 Pod programi (Subroutines) .....	59
6.15 Rukovanje sa greškama u programu .....	59
6.16 Vremenski događaji (timer events) .....	61
6.17 Kašnjenja, odlaganja u programima .....	61
6.18 Dijalog .....	62
6.19 Biblioteke (Libraries) .....	66
6.20 Rezime .....	67

## **POGLAVLJE 7 • JEDNOSTAVNI PROJEKTI SAMO NA MOBILNOM UREĐAJU ....68**

7.1 Pregled .....	68
7.2 Projekat 1 - Digitalni hronometar .....	68
7.2.1 Opis .....	68
7.2.2 Svrha .....	68
7.2.3 Listing programa .....	68

7.3 Projekt 2 - Kockica.....	75
7.3.1 Opis .....	75
7.3.2 Svrha .....	76
7.3.3 Listing programa .....	76
7.4 Projekt 3 - Brojevi za lutriju Euro Millions .....	80
7.4.1 Opis .....	80
7.4.2 Cilj .....	81
7.4.3 Listing programa .....	81
7.5 Projekat 4 - Lekcije iz geografije.....	87
7.5.1 Opis .....	87
7.5.2 Svrha .....	87
7.5.3 Listing programa .....	87
7.6 Projekt 5 - Matematika za osnovnu školu .....	94
7.6.1 Opis .....	94
7.6.2 Svrha .....	95
7.6.3 Listing programa .....	95

## **POGLAVLJE 8 • PROJEKTI KOJI KORISTE OSOBINE MOBILNIH UREĐAJA .....100**

8.1 Pregled .....	100
8.2 Telefonski senzori .....	100
8.3 Projekt 6 prikaz atmosferskog pritiska (Ambient Pressure) .....	101
8.3.1 Opis .....	101
8.3.2 Svrha .....	101
8.3.3 Listing programa .....	101
8.3.4 Modifikovani program .....	104
8.4 Projekt 7 - Prikaz nivoa osvetljenosti okoline.....	106
8.4.1 Opis .....	106
8.4.2 Svrha .....	106
8.4.3 Listing programa .....	106

8.5	Projekt 8 - Vibracija telefona na pri slabom svetlu .....	108
8.5.1	Opis .....	108
8.5.2	Svrha .....	108
8.5.3	Listing programa .....	109
8.6	Projekt 9 - Prikaz udaljenosti (proximity) sa tasterima Start/Stop.....	110
8.6.1	Opis .....	110
8.6.2	Cilj .....	110
8.6.3	Listing program .....	110
8.7	Projekt 10- Prikaz ubrzanja i slanje putem SMS-a .....	114
8.7.1	Opis .....	114
8.7.2	svrha.....	114
8.7.3	Listing programa .....	114
8.7.4	Modifikovani program .....	119
8.8	Projekat 11 - Korišćenje više senzora .....	122
8.8.1	Opis .....	122
8.8.2	Svrha .....	122
8.8.3	Listing programa .....	122
8.9	Projekt 12 - Telefonski pozivi.....	125
8.9.1	Opis .....	125
8.9.2	Svrha .....	125
8.9.3	Listing programa .....	126
8.10	Projekt 13/ Čuvanje podataka sa senzora.....	128
8.10.1	Opis .....	128
8.10.2	Cilj	128
8.10.3	Listing programa .....	128
8.11	Projekt 14 - Telefon izgovara vrednost osvetljaja .....	132
8.11.1	Opis .....	132
8.11.2	Svrha .....	133
8.11.3	Listing projekta.....	133
8.12	Ostali senzori telefona .....	136

## **POGLAVLJE 9 • UPOTREBA GPS-A (GLOBAL POSITIONING SYSTEM).....138**

9.1	Pregled .....	138
9.2	Projekt 15- Prikaz podataka mesta (lokacije) .....	138
9.2.1	Opis .....	138
9.2.2	Cilj .....	138
9.2.3	Listing programa .....	138

## **POGLAVLJE 10 • WI-FI INTERFEJS ANDROID NA PC .....143**

10.1 Pregled .....	143
10.2 Projekt 16 • Slanje i prijem podataka sa PC-ja .....	143
10.2.1 Opis .....	143
10.2.2 Svrha .....	143
10.2.3 Blok dijagram .....	143
10.2.4 Listing projekta.....	144
10.2.5 Provera UDP.....	147
10.3 Projekt 17 - Obrtanje reči na PC-ju .....	148
10.3.1 Opis .....	148
10.3.2 Svrha .....	148
10.3.3 Blok dijagram .....	148
10.3.4 Listing programa .....	149

## **POGLAVLJE 11 • WI-FI INTERFEJS ANDROID NA RASPBERRY PI .....151**

11.1 Pregled .....	151
11.2 Računar Raspberry Pi .....	151
11.2.1 Pločica Raspberry Pi 3 .....	151
11.2.2 Podešavanje Wi-Fi i pristupa sa daljine na Raspberry Pi .....	152
11.2.3 Definicije izvoda Raspberry Pi 3 GPIO.....	157
11.2.4 Biblioteka GPIO.....	159
11.2.5 Oznake GPIO izvoda .....	159
11.2.6 Konfiguracija kanala (I/O port pinova).....	159
11.3 Projekt 18 - Kontrola LED sa Android mobilnog telefona .....	162
11.3.1 Opis .....	162
11.2.3 Svrha .....	162
11.3.3 Blok dijagram .....	163
11.3.4 Skica kola.....	163
11.3.5 Konstrukcija .....	164
11.3.5 Android program .....	165
11.3.7 Program na Raspberry Pi.....	167
11.4 Projekt 19 • Prikaz temperature na mobilnom telefonu.....	169
11.4.1 Opis .....	169
11.4.2 Svrha .....	169
11.4.3 Blok dijagram .....	169
11.4.4 Pločica Sense HAT.....	169
11.4.5 Android program .....	172

11.4.6 Program na Raspberry Pi-ju .....	174
---	-----

## **POGLAVLJE 12 • ANDROID I RASPBERRY PI SMS INTERFEJS.....176**

12.1 Pregled .....	176
12.2 SIM800C .....	176
12.3 Projekt 20 • Kontrolisanje relea na Raspberry Pi 3 pomoću SMS poruka.....	179
12.3.1 Opis .....	179
12.3.1 Svrha .....	179
12.3.3 Blok dijagram .....	179
12.3.4 Skica kola.....	179
12.3. 5 Android program .....	180
12.3.6 Program na Raspberry Pi 3 .....	180

## **POGLAVLJE 13 • ANDROID NA ARDUINO WI-FI INTERFEJS .....184**

13.1 Pregled .....	184
13.2 Arduino Uno .....	184
13.3 Projekat 21 - Kontrola LED na Arduino Uno .....	186
13.3.1 Opis .....	186
13.3.2 Svrha .....	186
13.3.3 Blok dijagram .....	186
13.3.4 Skica kola.....	186
13.3.5 Android program .....	188
13.3.6 Program na Arduino Uno.....	189
13.4 Projekat 22 • Prikaz temperature i vlažnosti.....	192
13.4.1 Opis .....	192
13.4.2 Svrha .....	192
13.4.3 Blok dijagram .....	192
13.4.4 Električna šema.....	192
13.4.5 Android program .....	194
13.4.6 Program za Arduino Uno .....	196

## **POGLAVLJE 14 • SMS INTERFEJS ANDROID NA ARDUINO .....200**

14.1 Pregled .....	200
14.2 SMS poruke .....	200
14.2.1 Slanje i prijem u Text modu .....	201
14.3 Modul na Arduino SIM900 GSM/GPRS .....	203
14.4 Projekat 23 - Kontrolisanje relea SMS porukom.....	206



14.4.1 Opis .....	206
14.4.2 Svrha .....	206
14.4.3 Blok dijagram .....	206
14.4.4 Električna šema .....	206
14.4.5 Konstrukcija .....	207
14.4.6 Android program .....	207
14.4.7 Program na Arduino Uno .....	209
<b>POGLAVLJE 15 • ANDROID NA ESP32 WI-FI INTERFEJS.....</b>	<b>214</b>
15.1 Pregled .....	214
15.2 Procesor ESP32 .....	214
15.2.1 Arhitektura ESP32 .....	215
15.2.2 Razvojna pločica .....	216
15.3 Projekt 24 - Kontrola LED preko ESP32 DevKitC .....	219
15.3.1 Opis .....	219
15.3.2 Svrha .....	220
15.3.3 Blok dijagram .....	220
15.3.4 Električna šema .....	220
15.3.5 Konstrukcija .....	220
15.3.6 Android program .....	221
15.3.7 program na ESP32 .....	222
15.4 Projekt 25 - Milivoltmetar .....	224
15.4.1 Opis .....	224
15.4.2 Svrha .....	225
15.4.3 Blok dijagram .....	225
15.4.5 Android program .....	225
15.4.6 ESP32 program.....	226
<b>PRILOG A • UPOTREBA ANDROID EMULATORA .....</b>	<b>229</b>
Primer A1 .....	229
Primer A2 .....	230
Prilog B • Objavljivanje aplikacije na Google Play .....	234
B1 Razvoj aplikacije za Google Play .....	234
<b>PRILOG B • OBJAVLJIVANJE APLIKACIJE NA GOOGLE PLAY .....</b>	<b>234</b>

Knjiga je o razvoju aplikacija za mobilne android uređaje ( mobilne telefone i tablete) uz pomoć programskog jezika Basic za Android (B4A) i integrisanog razvojnog okruženja (B4A IDE). U knjizi su uključeni ispitani i radni projekti gde većina projekata zasnovana na hardveru ima pod poglavlja:

- Naslov projekta
- Opis projekta
- Svrha projekta
- Blok dijagram
- Električna šema
- Konstrukcija
- Listing čitavog programa projekta
- Potpuni opis programa

Knjiga je namenjena studentima, hobistima i bilo kome ko je zainteresovan za razvoj aplikacija (App) za android mobilne uređaje. Osobine i sintaksa programa B4A je zatim opisana korak po korak uz pomoć jednostavnih projekata.

Lepa osobina knjige da u većini projekata opisuje korak po korak kako Android mobilni uređaj može komunicirati sa Raspberry Pi, ili Arduino ili procesorom ESP32 preko Wi-Fi linka ili korišćenjem SMS poruka.

Primeri projekata opisuju kako Android mobilni uređaj može slati naredbe uz pomoć UDP protokola ili SMS poruka u cilju upravljanja uređajima vezanim na Raspberry Pi, Arduino ili na procesor ESP32. Pored toga neki projekti prikazuju kako se paketi podataka mogu slati sa Raspberry Pi, Arduino ili sa ESP32 procesora na Android prenosni uređaj i zatim prikazati na mobilnom telefonu.

Svi projekti sa strane Androida u knjizi su razvijeni upotrebom programskog jezika B4A. Raspberry Pi projekti su razvijeni uz pomoć jezika Python. Arduino i ESP32 procesor su uređeni popularnim Arduino IDE. U knjizi su dati potpuni listinzi programa kao i detaljni opisi. Korisnik bi trebalo da je u stanju da koristi projekte kao su izloženi ili da ih izmeni prema svojim potrebama.

## Predgovor

Svetske statistike pokazuju da je broj prodatih pametnih telefona krajnjim korisnicima stalno rastao u poslednjoj dekadi od 2007 do 2017. Godine 2016 krajnjim korisnicima je širom sveta prodato oko 1.5 milijarda pametnih telefona. Godine 2017 broj je porastao na 1.54 milijarde što je značajan porast za samo godinu dana. U četvrtom kvartalu 2016 81.7% od svih pametnih telefona prodatih krajnjim korisnicima su bili sa Android operativnim sistemom. Ovaj broj je porastao na 85.9% u prvom kvartalu 2018 (izvor <https://www.statista.com>).

Razvojne aplikacije za mobilne telefone nisu lak zadatak i zahtevaju veliko znanje programskih veština. Razvoj programa zahteva priličnu količinu vremena. Aplikacije bazirane na Androidu su na raspolaganju u Google Play Store. Većina aplikacija je besplatna i lako se mogu skinuti na mobilni telefon. Problem sa većinom od ovih aplikacija je da nisu zvanično ispitane pa su na raspolaganju kako ih vidite. Dodatno, većina takvih aplikacija sadrži reklame koje mogu smetati korisnicima. Moguće je kupiti neke profesionalne aplikacije bez ugrađenih reklama.

Ova knjiga je o razvoju App za Android mobilne telefone ili tablete uz pomoć jezika **Basic-a za Android (B4A)** i integrisanog razvojnog okruženja. Sintaksa i razvojno okruženje B4A su slični razvojnom okruženju za Visual Basic i zato vrlo pogodni za brz razvoj aplikacija bez previše znanja prethodnih programskih jezika.

B4A uključuje sve osobine Android mobilnih uređaja i zato je savršena platforma za razvoj aplikacija za Android mobilne telefone. Karakteristike kao što su Wi-Fi, SMS, NFC, grafika, serijski port, pomoćni programi Sve podržava B4A uz mnoge druge osobine. Hiljade programera širom sveta uključujući čuvene kompanije kao što su HP, IBM pa NASA, koriste B4A.

Nažalost na tržištu nema na raspolaganju mnogo knjiga o B4A. Jedna od najobimnijih knjiga koju je autor prošao je napisao Wyken Seagrave a zove se B4A Rapid App Development Using BASIC.

Ova knjiga daje uvod u sintaksu B4a i daje primere projekata koji pokazuju kako razviti jednostavne aplikacije uz pomoć B4A. Zanimljiva i značajna stvar ove knjige je da opisuje kako Android mobilni telefon može komunicirati sa Raspberry Pi, Arduino ili ESP32 baziranim razvojnim pločama na daljinu uz pomoć Wi-Fi i SMS poruka. U knjizi su dati potpuni listinzi programa B4A, Raspberry Pi, Arduino i ESP32 procesora sa detaljnim opisima hardvera i softvera za svaki projekat. Nadam se da vam se dopada knjiga i da je nalazite korisnom za vaše sledeće projekte zasnovane na Androidu.

*Prof Dr Dogan Ibrahim*

## Poglavlje 2 • Početak

### 2.1 Pregled

Ovo poglavlje je o instalaciji i početnom korišćenju softvera B4A na Vindovs računaru. B4A se može nabaviti od Anywhere Software ([www.b4x.com](http://www.b4x.com)) gde postoje četiri različite verzije u zavisnosti od podrške ažuriranja i zahteva licence:

- Probna verzija (Trial version)
- Standardna verzija
- Verzija za firme (Enterprise version)
- Site licence

Standardna i 'Enterprise' verzija su za programere gde svaki programer mora da ima posebnu licencu.

Probna verzija je besplatna i opisana je u sledećem odeljku a može se skinuti sa interneta. Ova verzija ima ograničeno vremensko trajanje do 30 dana i ograničenu veličinu programa. Druge verzije imaju trajne licence.

Standardna verzija košta 59 USD u vreme pisanja knjige. Ova verzija uključuje 2 meseca besplatnog nadograđivanja i pun pristup forumima.

'Enterprise' verzija košta 119 USD i uključuje 2 godine nadogradnje pored punog pristupa forumima.

'Site licence' košta 599 USD i uključuje 30 licenci za istu organizaciju. Dodatno, kao kod 'Enterprise' verzije, ova verzija uključuje 2 godine besplatne nadogradnje i pun pristup forumima.

Akadske licence B4A (za studente, nastavnike i istraživače) su na raspolaganju u pola cene ali se svi detalji mogu dobiti sa razvojnog veb sajta:

*<https://www.b4x.com/store.htm>*

### 2.2 Instalacija probne verzije B4A

Probna verzija se može pustiti u rad na dva načina; **Remote Compilation Mode** ili **Local Compilation Mode**. Remote compilation Mode zahteva instalaciju samo nekoliko komponenta ali ima ograničene mogućnosti. U ovom odeljku ćemo pretpostaviti da čitalac koristi probnu verziju kratko vreme a zatim kasnije nadograđuje u punu verziju. Zato ćemo instalirati sve zahtevane komponente tako da probna verzija (a kasnije i puna) radi u Local Compilation Mode sa dodatnim karakteristikama kao što je upotreba emulatora, potpuno otklanjanje grešaka (debugging), koje se naziva **Rapid Debugger** itd.

Pre instalacije B4A potrebno je instalirati **Java SDK** i **Android SDK**. mi ćemo zatim instalirati **B4A-Bridge** na naš Android uređaj posle instalacije B4A. B4A-Bridge dozvoljava da se IDE poveže na Android uređaj bežično tako da možemo otklanjati greške na

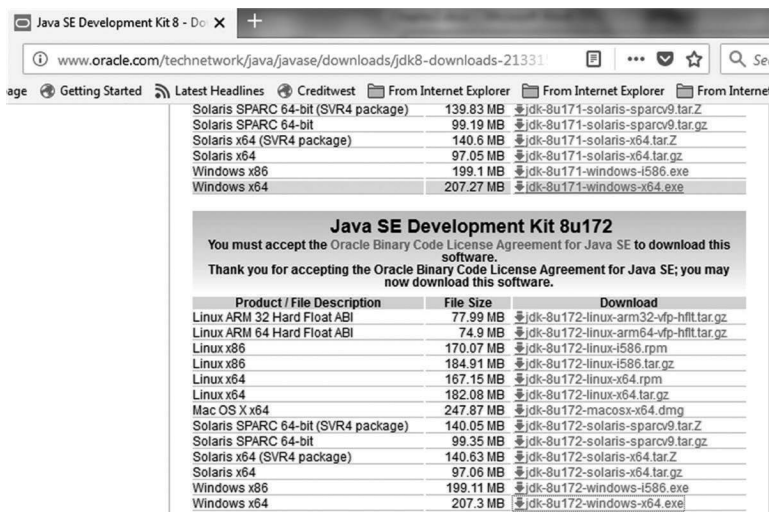
našoj aplikaciji (možemo se vezati preko USB veze kao što ćemo kasnije videti u ovom poglavlju).

Koraci instalacije probne verzije su dati dole (pretpostavlja se da ćete instalirati na Windows PC):

Idite na stranicu za skidanje Java SDK:

<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

- Kliknite na taster Accept Licence Agreement
- Izaberite **Windows x86** (vidi sliku 2.1) čak i ukoliko imate 64-bitnu mašinu i kliknite na odgovarajući **Download** fajl
- Skinite i instalirajte fajl (u vreme pisanja knjige naziv fajla je bio **jdk-8u172-windows-i586.exe** a veličina je bila 199 MB)



Slika 2.1 Izaberite proizvod u skladu sa vašim računarom

- Skinite Android SDK sa sledećeg linka (vidite sliku 2.2) i instalirajte u folder, na primer C:\Android ( nemojte instalirati u folder sa razmaknicama u nazivu foldera kao kod programskih fajlova)

[https://dl.google.com/android/instaler\\_r23.02-windows.exe](https://dl.google.com/android/instaler_r23.02-windows.exe)

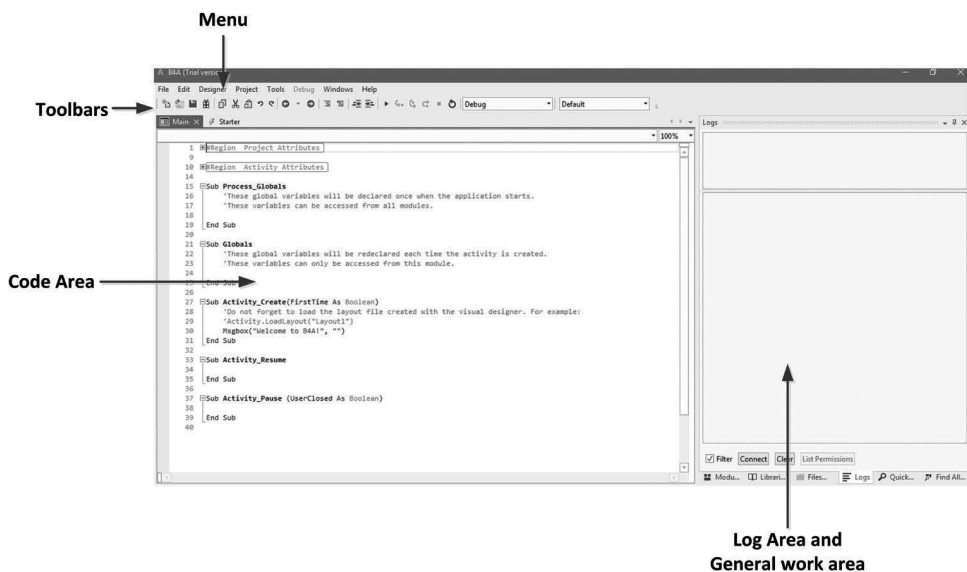
## Poglavlje 3 • Moj prvi B4A program

### 3.1 Pregled

U ovom poglavlju ćemo videti kako instalirati probnu i Standard verziju B4A na Windows računaru. U ovom poglavlju bi trebalo razviti veoma jednostavan program i pogledati osnovne osobine integrisanog razvojnog okruženja za B4A. U ovom poglavlju ćemo prikazati poruku Hello from the B4A na Android mobilnom telefonu.

### 3.2 Rad sa okruženje B4A IDE

Posle pokretanja programa B4A trebalo bi da vidite ekran dobrodošlice. Kliknite Close da bi videli B4A glavni ekran kao na slici 3.1



Slika 3.1 Glavni ekran B4A

Naš kod pišemo i redigujemo na 'Code Area'. Kada se B4A na ekranu se prikaže uzorak šablona da pomogne korisnicima da počnu pisanje svog programa. Prostor kodiranja je podeljen u oblasti gde se sadržaj ovih oblasti može proširiti klikom na '+' na levo od regiona. Klikom na '-' znak na levo od regiona potpuno suzi oblast tako da se sadržaj ne ispisuje. Sledeće oblasti se podrazumevano prikazuju :

- Project Attributes
- Activity Attributes
- Process\_Globals
- Globals
- Activity\_create
- Activity\_Resume
- Activity\_Pause

Project Attributes i Activity Attributes određuju osobine tekućeg projekta odnosno aktivnosti. Po pravilu nemate potrebe da menjate sadržaj ove dve oblasti. Podrazumevani sadržaj ove dve oblasti je prikazan na slici 3.2

```
#Region Project Attributes
#ApplicationLabel: B4A Example
#VersionCode: 1
#VersionName:
'SupportedOrientations possible values: unspecified, landscape or portrait.
#SupportedOrientations: unspecified
#CanInstallToExternalStorage: False
#End Region

#Region Activity Attributes
#FullScreen: False
#IncludeTitle: True
#End Region
```

Slika 3.2 Podrazumevani sadržaj projektnih i atributa aktivnosti

Druge oblasti su u obliku pod rutina. B4A aplikacija se sastoji od jedne ili više aktivnosti gde je aktivnost slična formi u Windows-u. Aktivnost se može ukinuti ukoliko nije u prvom planu da bi se sačuvala memorija. **Process\_Globals** se koristi za smeštaj globalnih promenljivih koje su deklarisanе onda kad je aplikacija započela i ovim promenljivim pristupamo iz svih modula u programu. Globals-ima se može pristupiti samo iz tog modula i one se svaki put ponovo deklarishu kada se kreira aktivnost. **Activity\_Create** je gde je naš glavni programski kod a kod u ovoj oblasti se poziva posle početka programa tj. kad se kreira aktivnost. Parametar FirstTime nam kaže da li je data aktivnost stvarno prvi put kreirana. Ovaj parametar se može iskoristiti da inicijalizuje izvesne promenljive ukoliko je , na primer vrednost 'True'. **Activity\_Pause** se naziva kad se aktivnost ide iz prvog plana u drugi. Ovo se može desiti, na primer, kada startuju različite aktivnosti, pritisnuti tasteri 'Home' ili 'Back' itd. **Activity\_Resume** se poziva posle završetka **Activity\_Create** ili posle ponovnog uspostavljanja posle prekida (pause) tako da se aktivnost vraća iz zadnjeg plana u prednji.

Svi projekti sadrže modul koji se naziva Starter. Ovaj modul se može proširiti klikom na levoj strani ekrana odmah pored 'Main. Ovo je modul se prvi izvršava (ukoliko postoji) kada aplikacija počinje. Podrazumeva se da su pod rutine u Starteru prazne i zato se kod glavne aktivnosti izvršava kada aplikacija počinje. Starter samo startuje jednom i ostaje u memoriji sve dok se uređaj ne ugasi. Preporučljivo je ovde deklarirati **Process\_globals** pošto starter ostaje u memoriji.

### 3.3 Prikaz poruke na Android mobilnom uređaju

Koraci da bi se prikazala poruka Hello from the B4A je data dole. U ovom primeru će biti prikazana poruka na stvarnom Android mobilnom telefonu:

- Ubacite poruku unutar **Activity\_Create** pod rutine kako je prikazano na slici 3.3

## Poglavlje 5 • Moj treći B4A program

### 5.1 Pregled

U prethodnom poglavlju smo videli kako napraviti jednostavnu aplikaciju sa dva tastera. U ovom poglavlju ćemo napraviti jednostavan kalkulatorski program i naučiti kako koristiti ‘dibager’ i kako komunicirati sa Android uređajem preko USB veze.

### 5.2 Jednostavan kalkulatorski program

Svrha primera je pokazati kako upotrebiti druge kontrole (views) B4A. Ovo je jednostavan kalkulatorski program koji može sabirati, množiti, deliti i oduzimati. Korisnik unosi dva broja i onda pritisne jedan od četiri tastera da bi izabrao zahtevanu operaciju. Odgovor računanja se prikazuje preko MessageBox-a.

Koraci projektovanja koda kalkulatorskog programa su dati dole (možete se na Android mobilni uređaj povezati preko Designer opcije WYSIWYG da vidite dizajn na stvarno uređaju):

- Kliknite **Designer**—>**Open Designer** da bi otvorili ekran dizajnera
- Postavite pozadinsku boju na belu u **Activity Properties** (color:#FFFFFF)
- Dodajte Oznaku klikom na **Add View**—>**Label**. Ovu oznaku nazovite **txtLabel** i postavite tekst **SIMPLE CALCULATOR**, postavite veličinu teksta na 18 a boju na crnu (color:#000000)
- Dodajte oznaku (label) i nazovite **txtLabelNo1** i postavite tekst **Enter First Number**, a boju postavite na crnu (color:#000000)
- Dodajte **EditText** okvir pored prethodne i ovaj okvir nazovite **txtNo1**. Postavite Input Type na NUMBERS i postavite boju na crnu (color:#000000)
- Dodajte drugu oznaku i nazovite je **txtLabelNo2** i postavite tekst **Enter Second Number**, postavite boju na crnu (color:#000000)
- Dodajte okvir **EditText** pored oznake i nazovite ga **txtNo2**. Postavite Input Type na NUMBERS i boju na crnu (color:#000000)
- Dodajte četiri tastera sa imenima **txtButtonPlus**, **txtButtonMinus**, **txtButtonMult** i **txtButtonDiv**. Postavite tekst ovih tastera na + . - x. / prema odgovarajućim tasterima i postavite veličinu teksta na 20.

Slika 5.1 prikazuje izgled dizajna za ovaj primer. Kliknite File da bi sačuvali projekat. U ovom primeru projektu je dat naziv **Calculator**.

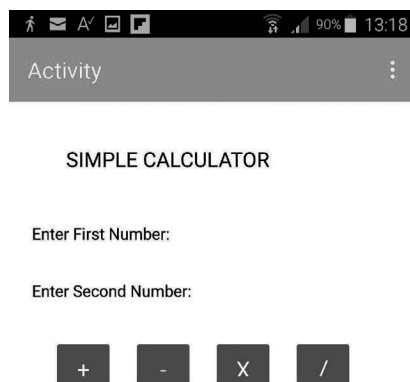




Slika 5.1 Izgled dizajna

Ovde okviri **EditTEXT** čuvaju dva broja nad kojima će se vršiti operacija. Rezultat operacije će biti prikazan pomoću MessageBOX-a.

Sada ćemo videti kako će dizajn izgledati na stvarnom uređaju. Pokrenite B4A Bridge na vašem Android uređaju i povežite ga klikom na **Tools—>B4A—>Connect**. Kliknite WYSIGWYG na ekranu Designera i zatim kliknite Connect. Trebalo bi da vidite izgled ekrana kao na slici 5.2. **Zapamtite da kad se povežete možete podesiti mesto i dimenzije kontrola na ekranu stvarnog uređaja rukujući u meniju dizajnera.**



Slika 5.2 Izgled ekrana na stvarnom android uređaju

Sad je potrebno generisati kod za razne kontrole na ekranu. Kao u prethodnom projektu kliknite na **Tools—>Generate Members** i postavite način rada tastera na **Click** tako da ta-

## Poglavlje 6 • Napomene o jeziku B4A

### 6.3 Pregled

U ovom poglavlju ćemo detaljnije pogledati na jezik B4A i naučiti o promenljivim, linijama komentara, uslovnim iskazima, nizovima itd.

### 6.2 Komentari

Linije primedbi u B4A počinju sa znakom jednostrukog apostrofa. Primer je dat dole:

- Ovo je linija primedbi B4A
- Linije sa komentarima su informativne i ne izvršavaju se
- Linije komentara su korisne za održavanje programa

Možemo koristiti blokove programa gde se može dati primedba o bloku. Blok komentari se ubacuju i izbacuju uz pomoć ikona prikazanim na slici 6.1. Možete naglasiti linije koje ćete komentarisati i zatim kliknuti na ikonu Block Comment radi primedbi na izabrane linije.

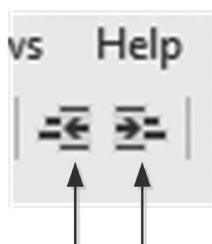


**Block Comment    Uncomment**

Slika 6.1 Ikona Blok koment

### 6.3 Uvlačenje - zupčanje

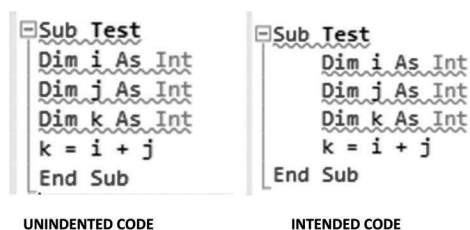
Dobra praksa je nazupčavanje koda da bi se lakše pratio. Možete zupčati kod za vreme pisanja ili kliknuti na taster za zupčanje da bi kod uvlačili automatski posle pisanja. Ikona uvlačenja je data na slici 6.2. Ikona za obratan postupak uklanja zupčanje.



**Outdent    Indent**

Slika 6.2 Ikona zupčenja

Na slici 6.3 je deo koda koji nije zupčan. Izaberite kod koga je potrebno nazupčati i kliknite na ikonu uvlačenja. Nazupčani kod će biti prikazan na slici 6.3



Slika 6.3 Nazupčani kod

## 6.4 Razlikovanje velikih slova i odvajanje izraza

Jezik B4A je osetljiv na to da li su u pitanju mala ili velika slova gde editor automatski menja veličinu slova u slučaju ključnih reči. Dozvoljeno je koristiti više od jednog iskaza u liniji odvajanjem izraza tačkom i zapetom (;). Primer je dat dole:

```
Dim i as Int; Dim J as Int; Dim k as Int
```

## 6.5 Konstante

Konstante se deklarišu pomoću ključne reči Const. Vrednost konstante se ne može menjati unutar programa. Primer je dat dole:

```
Dim Const MyPi as Double=3.14159
```

## 6.6. Promenljive

Promenljive su određene nazivima. Imena promenljivih moraju počinjati slovima i mogu koristiti znake od A-Z, a-z, brojeve 0-9 i donja crtica (\_). Razmaci, zagrade, matematički simboli ili bilo koji simbol se ne može koristiti kao naziv promenljive.

U jeziku B4A postoje dva tipa promenljivih: Primitive types i Non-Primitive Types.

### Primitive types

Ove promenljive prenose vrednosti kada se prenose u pod programe (Sub) ili kada su dodeljene drugim promenljivim. Tabela 6.1 prikazuje primitivne promenljive koje su dozvoljene u B4A:

Primitive Type	Type	No of bits	Min value	Max value
Boolean	boolean	1	FALSE	TRUE
Byte	byte	8	-128	127
Short	integer (signed)	16	-32768	32767
Int	integer (signed)	32	-231	231 - 1

## Poglavlje 7 • Jednostavni projekti samo na mobilnom uređaju

### 7.1 Pregled

U ovom poglavlju ćemo razviti nekoliko jednostavnih projekata uz pomoć B4A i samo za mobilni uređaj (tj. Android mobilni telefon). Nadamo se da će čitaoci naučiti kako da koriste razne osobine B4A za vreme razvojnog rada na projektima. Projekti su potpuno bazirani na B4A za mobilne uređaje.

Sledeća zaglavlja će biti data za svaki projekat

- Naslov projekta
- Opis projekta
- Svrha projekta
- Listing čitavog programa projekta
- Opis programa
- Preporuke za dalji rad (ukoliko je moguće)

### 7.2 Projekat 1 - Digitalni hronometar

#### 7.2.1 Opis

U ovom projektu je dizajniran digitalna štoperica. Koriste se tri tastera, sa nazivima START, STOP i CLEAR. Hronometar počinje i zaustavlja se kada se pritisnu tasteri Start odnosno Stop. Klikom na taster Clear briše se brojanje tako da štoperica bude spremna za novo brojanje. Brojanje se prikazuje u sekundama.

#### 7.2.2 Svrha

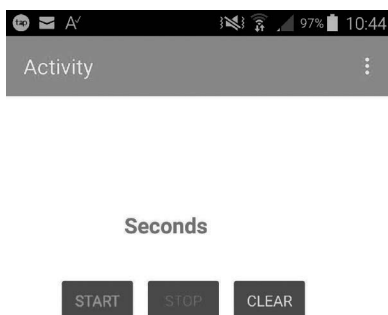
Svrha projekta je pokazati kako se koristi tajmerski modul B4A da bi se projektovala digitalna štoperica.

#### 7.2.3 Listing programa

U programu se koristi Label (tablica) da prikaže vreme i tri tastera za kontrolu hronometra. Koraci dizajna projekta su dati dole:

- Pokrenite B4A
- Pokrenite B4A-Bridge na vašem mobilnom telefonu i kliknite START
- Povežite se na mobilni telefon klikom na **Tools —>B4A Bridge —> Connect**
- Kliknite **Designer** a zatim **Open Designer**
- Kliknite **WYSIWYG** pa **Connect** da bi se vezali na mobilni uređaj tako da možete videti dizajn na vašem mobilnom telefonu
- Postavite **Color** u **Activity Properties** na belu boju.

- Kliknite na **Add View** i dodajte novu Label-u
- Promenite ime Labele na **SECONDS**
- Postavite **Text Color** na crnu, njegov **Style** na BOLD a veličinu **Size** na 20
- Kliknite **Add View** i dodajte 3 tastera na dno tablice (labele). Tastere nazovite **START**, **STOP** i **CLEAR** i respektivno njihov **Texts** na **START**, **STOP** i **CLEAR**.
- Postavite boju teksta tastera **START**, **STOP** i **CLEAR** respektivno na zelenu, crvenu i žutu.
- Slika 7.1 prikazuje izgled ekrana mobilnog uređaja.



Slika 7.1 Izgled kрана mobilnog uređaja

- Sada generišemo kod a stavke na ekranu  
Kliknite **Tools** —> **Generate Members** i kliknite **Select All Views**
- Kliknite **START**, **STOP** i **CLEAR** i postavite **Click** tako da tasteri odgovaraju kada se klikne na njih (slika 7.2)
- Kliknite **Generate Members** i zatvorite ekran

## Poglavlje 8 • Projekti koji koriste osobine mobilnih uređaja

### 8.1 Pregled

U ovom poglavlju ćemo razviti nekoliko jednostavnih projekata uz pomoć B4A i biblioteke **Phone Library** koja uključuje mnoštvo osobina mobilnog uređaja (Android mobilnog telefona). Neke od tih funkcija uključenih u **Phone Library** su:

- Call log (pristup pozivima)
- Spisak kontakata (pristup zapisanim kontaktima)
- Elektronska pošta
- Informacije o telefonu
- Senzori na telefonu
- Događaji
- Rukovanje porukama (prijem i slanje)
- Kontrola vibracija telefona
- Rad sa zvukovima zvona
- Prepoznavanje glasa

Nadamo se da će čitaoci naučiti kako da iskoriste razne osobine mobilnih uređaja razvojem projekata uz pomoć projekata pisanih u B4A. Projekti će biti opisani kao u prethodnom poglavlju.

### 8.2 Telefonski senzori

**Phone Sensors** su unutar **Phone Library**. U zavisnosti od izabranog modela Android mobilni telefoni imaju ugrađene razne senzore. Skoro svi noviji Android mobilni telefoni podržavaju sledeće senzore

- Senzor atmosferskog pritiska
- Senzor ubrzanja
- Žiroskop
- Svetlosni senzor
- Senzor gravitacije
- Senzor magnetskog polja
- Blizinski senzor

Neki modeli poseduju dodatne senzore kao što je senzor spoljašnje temperature i relativne vlažnosti. U programima B4A podaci senzora se dobijaju korišćenjem biblioteke PhoneSensors. Pre korišćenja senzora mora se izvršiti inicijalizacija određenog senzora koga želimo da upotrebimo. U vreme pisanja knjige mogli su se inicijalizovati sledeći senzori:

- TYPE\_ACCELEROMETER
- TYPE\_GYROSCOPE
- TYPE\_LIGHT
- TYPE\_MAGNETIC\_FIELD

- TYPE\_ORIENTATION
- TYPE\_PRESSURE
- TYPE\_PROXIMITY
- TYPE\_TEMPERATURE

Posle inicijalizacije senzora možemo pogledati da li mobilni telefon podržava određeni senzor. Ovo se radi uz pomoć iskaza **StartListening** koja vraća **True** ukoliko je izabrani senzor podržan. Ako je senzor podržan od strane našeg mobilnog telefona tada podatke sa senzora možemo dobiti u subrutini **SensorChanged**. Ovaj pod program ima argument niza sa tekućim zarezom koji se naziva **Values** gde se smeštaju podaci sa senzora. Na primer kada se koristi senzor ubrzanja, Values smešta tri vrednosti koje su ubrzanja po x, y i z osi. Ako se na primer koristi senzor pritiska onda Values čuva samo jedan podatak.

Neki primeri korišćenja senzora mobilnog telefona su dati u sledećim odeljcima.

### 8.3 Projekt 6 prikaz atmosferskog pritiska (Ambient Pressure)

#### 8.3.1 Opis

U ovom projektu se sa senzora Android mobilnog telefona očitava atmosferski pritisak i zatim ispisuje uz pomoć labele.

#### 8.3.2 Svrha

Smisao ovog projekta je pokazati kako se može očitati atmosferski pritisak sa senzorskog modula Android mobilnog telefona.

#### 8.3.3 Listing programa

U ovom programu se koristi labela za prikaz atmosferskog pritiska. Koraci dizajna projekta su dati dole:

- Pokrenite B4A
- Pokrenite B4A-Bridge na vašem mobilnom telefonu i kliknite START
- Povežite mobilni telefon klikom na Tools—>B4A Bridge—>Connect
- Klik na **Designer** pa zatim **Open Designer**
- Klik na **WSIWYG** pa onda **Connect** da bi se povezali sa mobilnim uređajem tako da možete videti tok projekta na vašem mobilnom telefonu.
- Podesite **Color** u **Activity Properties** na belu.
- Kliknite **Add View** i dodajte novu labelu
- Promenite naziv na labeli u **AmbientPressure**
- Podesite tekst **Color** na crnu a **Style** na BOLD i **Size** na 14

## 8.8 Projekat 11 - Korišćenje više senzora

### 8.8.1 Opis

U ovom projektu se koriste senzori pritiska i svetlosti a njihovi podaci se prikazuju na dve tablice.

### 8.8.2 Svrha

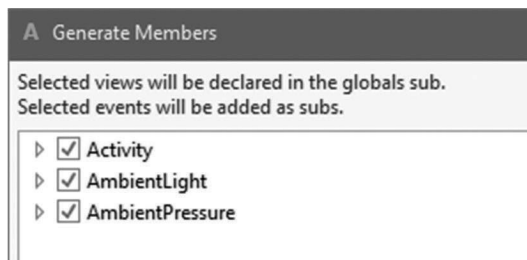
Cilj projekta je pokazati kako se u projektu koristi više od jednog senzora.

### 8.8.3 Listing programa

U ovom projektu se koriste dve labele da prikažu atmosferski pritisak i nivo svetlosti. Prva se naziva **AmbientPressure** a druga **AmbientLight**.

Koraci dizajna projekta su dati dole:

- Pokrenite B4A
- Pokrenite B4A-Bridge na vašem mobilnom telefonu i kliknite **START**
- Povežite mobilni telefon klikom na **Tools—>B4A Bridge—>Connect**
- Klik na **Designer** pa zatim **Open Designer**
- Klik na **WSIWYG** pa onda **Connect** da bi se povezali sa mobilnim uređajem tako da možete videti tok projekta na vašem mobilnom telefonu.
- Podesite **Color** u **Activity Properties** na belu.
- Kliknite **Add View** i dodajte novu Label-u
- Promenite **Text** na prvoj tablici u **AmbientPressure** a drugu u **AmbientLight**
- Podesite tekst **Color** u **Activity Properties** na belu.
- Kliknite **Add View** i dodajte dve nove labele
- Promenite naziv prve labele u **AmbientPressure** a druge u **AmbientLight**
- Podesite tekst **Color** na crno, **Styles** na **BOLD** i **Size** 14
- Kliknite **Tools—>Generate members** i kliknite **Select All Views** a zatim kliknite **Generate Members** pa zatvorite ekran (vidi sliku 8.19)



Slika 8.19 Generate members



- Sada sačuvajte dizajnirani ekran pod imenom, na primer, **MULTIPLE**
- U naš program moramo uključiti biblioteku **Phone**. Kliknite na Libraries Manager na donjoj desnoj strani ekrana glavnog programa i izaberite **Phone** klikom na 'box' sa desne strane
- Sada je potrebno da ispunimo kodom. Unesite sledeće iskaze unutar **Sub Process\_Globals**:

```
Dim apressure As PhoneSensors
Dim alight As PhoneSensors
```

- Ubacite dizajnirani izgled **MULTIPLE**, podesite naslov na **Ambient Prssure and Light Level** unutar **ACTIVITY\_Create** i ubacite sledeći kod. U ovom kodu **alight** se inicijalizuje kucanjem **TYPE\_LIGHT** a **apressure** sa **TYPE\_PRESSURE**. Ubacite **StartListening** iskaz za senzore pritiska i osvetljaja i odredite ime subrutine kao **SensorTrig** da bi se aktivirala kada dođe do promene svetlosnih ili podataka pritiska. Subrutina **Activity\_Create** bi trebalo da ime sledeći sadržaj:

```
Sub Activity_Create(FirstTime As Boolean)
    Activity.LoadLayout("MULTIPLE")
    Activity.Title = "Ambient Pressure and Light Level"
    alight.Initialize(alight.TYPE_LIGHT)
    apressure.Initialize(apressure.TYPE_PRESSURE)
    If alight.StartListening("SensorTrig") = False Then
        MsgBox("Light sensor is not supported", "Error")
    End If

    If apressure.StartListening("SensorTrig") = False Then
        MsgBox("Pressure sensor is not supported", "Error")
    End If
End Sub
```

- Program poziva subrutinu **SensorTrig\_SensorChanged** ukoliko postoji promena podataka na senzoru. Unesite sledeći kod unutar ove subrutine. Promenljiva **ps** se definiše kao tip **PhoneSensors**. Pošiljalac vraća objekat koji okida događaj. U donjem kodu ukoliko je događaj aktiviran svetlosnim senzorom tada se prikazuje izlaz senzora. Slično, u slučaju da je aktiviran senzor pritiska onda se prikazuje izlaz senzora pritiska.

```
Sub SensorTrig_SensorChanged(Values() As Float)
    Dim MyLight As String
    Dim MyPressure As String
    Dim ps As PhoneSensors
    ps = Sender
```

## Poglavlje 9 • Upotreba GPS-a (Global Positioning System)

### 9.1 Pregled

U ovom poglavlju ćemo pogledati kako koristiti ugrađeni GPS Android mobilnog telefona. Projekt izložen u ovom poglavlju izdvaja i prikazuje lokalne GPS podatke.

Android GPS funkcije obrađuje GPS biblioteka B4A. U GPS biblioteci su na raspolaganju tri tipa objekata: **GPS**, **GPSSatellite** i **Location**.

**GPS** radi sa konekcijama i događajima, inicijalizuje pokreće i zaustavlja **GPS**, **GPSSatellite** vraća različite informacije o GPS satelitima, kao što su azimut, pseudo slučajni brojevi (Prn) za satelite i odnos signal šum (Snr). **Location** vraća podatke lokacije mesta kao što su geografska širina, dužina, visina, azimut, brzina, vreme i funkcije pretvaranja

### 9.2 Projekt 15- Prikaz podataka mesta (lokacije)

#### 9.2.1 Opis

U ovom projektu ćemo izdvojiti i prikazati podatke lokalnog mesta kao što su geografska širina, dužina i visina na našem Android mobilnom telefonu.

#### 9.2.2 Cilj

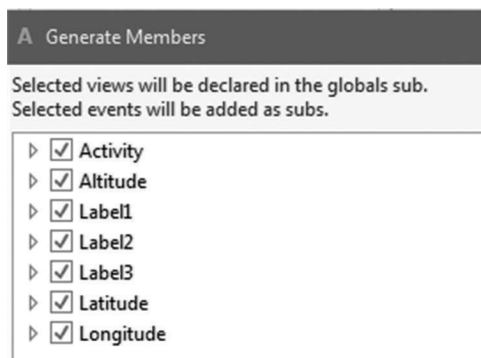
Cilj ovog projekta je pokazati kako se može ugrađeni GPS upotrebiti na Android mobilnom telefonu.

#### 9.2.3 Listing programa

U ovom programu se koriste tri labele da prikažu zaglavlja sa geografskom širinom, dužinom i visinom. Uz to se nalaze tri okvira za prikaz vrednosti geografske širine, dužine i visine. Koraci dizajna projekta su dati dole:

- Pokrenite B4A
- Pokrenite B4A-Bridge na vašem mobilnom telefonu i kliknite START
- Povežite mobilni telefon klikom na **Tools**—>**B4A Bridge**—>**Connect**
- Klik na **Designer** pa zatim **Open Designer**
- Klik na **WSIWYG** pa onda **Connect** da bi se povezali sa mobilnim uređajem tako da možete videti dizajn na vašem mobilnom telefonu.
- Podesite **Color** u **Activity Properties** na belu.
- Dodajte tri labele sa **Text Color** crne boje, **Size** 14 i ispišite **Text** ovih labela na **Latitude:**, **Longitude:** , odnosno **Altitude:**.
- Dodajte tri tekst okvira Edit Text gde je **Text Color** crne boje, **Size** 12 a njihova imena neka budu **Latitude**, **Longitude**, odnosno **Altitude**. Smestite ove okvire pored labela sa istim imenima.

- Kliknite **Tools**—> **Generate Members**. Kliknite **Select All Views** a zatim na **Generate Members** (vidi sliku 9.1) Sačuvajte izgled pod imenom GPS i zatim zatvorite dizajnirani ekran.



Slika 9.1 Generate members

Sada smo spremni da upišemo kod u naš GPS projekat. Kliknite **Libraries Manager** sa donje desne strane ekrana i kliknite da uključite **GPS** biblioteku i biblioteku **RuntimePermissions** u naš kod.

Kreirajte promenljivu nazvanu **MyGPS** tipa **GPS** i promenljivu **rp** tipa **RuntimePermissions** unutar **Sub Process\_Globals**:

```
Sub Process_Globals
    Dim MyGPS As GPS
    Dim rp As RuntimePermissions
End Sub
```

Unutar subrutine **Activity\_Create** definišemo izgled koji će se skinuti kao GPS, postaviti naslov aktivnosti na **MY LOCATION** i dozvoliti radnu dozvolu za pristup GPS. Kod zatim inicijalizuje GPS biblioteku pod imenom **MyGPS** tako da subrutina **MyGPS\_LocationChaged** bude pozvana automatski kada postoji promena na primljenim GPS podacima. Ako GPS nije dobio dozvolu na Android uređaja, tada će se privremeno prikazati poruka **Enable GPS on your device** i poslaće vas na odgovarajuću stranicu podešavanja da dozvolite GPS. Ukoliko je u drugu ruku GPS već aktiviran on počinje da radi. Zapazite da funkcija **MyGPS.start** ima dva argumenta: Prvi argument određuje najkraće vreme između događaja (mereno u milisekundama) a drugi argument određuje najkraću promenu rastojanja (merenu u metrima). Podešavanje oba parametra na 0 podrazumeva najvišu učestanost. Ako je, na primer, prvi parametar podešen na 5000 a drugi na 10 tada će GPS podaci biti ažurirani svakih 5 sekundi ili ukoliko se rastojanje promeni za 10 metara ili više:

## Poglavlje 10 • Wi-Fi interfejs Android na PC

### 10.1 Pregled

U ovom poglavlju ćemo pogledati kako Android uređaj (mobilni telefon ili tablet) može komunicirati sa PC-jem preko Wi-Fi mreže.

Komunikacija preko Wi-Fi veze koristi mrežnu biblioteku B4A. Ova biblioteka sadrži objekte za rad sa **TCP** (serverSocket i Socket) i **UDP** (UDPSocket i UDPPacket) tipom podataka komunikacionih protokola. Oba protokola, TCP i UDP se koriste za slanje i prijem paketa podataka preko interneta gde se podaci šalju ili primaju sa određenih IP adresa i brojeva portova. TCP znači ‘Transmission Control Protocol’, a to je pouzdan i protokol baziran na vezi za slanje i prijem paketa podataka preko interneta. UDP znači ‘User Datagram Protocol’ i on je protokol bez konekcije za slanje i prijem paketa podataka preko interneta TCP je veoma pouzdan komunikacioni protokol jer obezbeđuje proveru grešaka i obaveštenja o primljenim i poslatim paketima podataka. Bilo kakav neispravan ili nedostajući paket podataka se ponovo emituje. UDP je u drugu ruku manje pouzdan i nema potvrde slanja ili da li je primljen paket podataka. Iz toga potiče da TCP nije tako brz kao UDP protokol jer ima veće opterećenje. TCP protokol zahteva stranu klijenta i servera koji su spojeni pre razmene podataka. UDP s druge strane ne zahteva ni klijenta i ni servera da se vežu pre razmene podataka.

U ovom poglavlju ćemo razviti nekoliko projekata da pokažemo kako se podaci mogu razmeniti između PC-ja i Android mobilnog telefona upotrebom UDP paketa podataka.

### 10.2 Projekt 16 • Slanje i prijem podataka sa PC-ja

#### 10.2.1 Opis

U ovom projektu ćemo poslati poruku sa Android mobilnog telefona na PC preko interneta uz pomoć UDP protokola gde će poruka biti prikazana na PC-ju. U isto vreme podaci poslani sa PC-ja e biti prikazani na Mobilnom telefonu.

#### 10.2.2 Svrha

Svrha ovog projekta je da pokažemo kako se može koristiti UDP protokol za slanje i prijem podataka preko interneta između PC-ja i Android mobilnog telefona.

#### 10.2.3 Blok dijagram

Slika 1.1 prikazuje blok dijagram projekta. Komunikacija između PC-ja i Android mobilnog telefona se vrši preko lokalnog Wi-Fi rutera uz pomoć UDP protokola. IP adrese mobilnog telefona i PC-ja su respektivno “192.168.1.78” i “192.168.1.71”.

## Poglavlje 11 • Wi-Fi interfejs Android na Raspberry Pi

### 11.1 Pregled

U ovom poglavlju ćemo pogledati kako Android uređaj (mobilni telefon ili tablet) može komunicirati sa računarom Raspberry Pi preko Wi-Fi veze koristeći UDP protokol. Pre detaljnog pregleda interfejsa Android na Raspberry Pi, vredno je osmotriti neke osnovne funkcije Raspberry Pi za one čitaoce kojima je Raspberry Pi novost

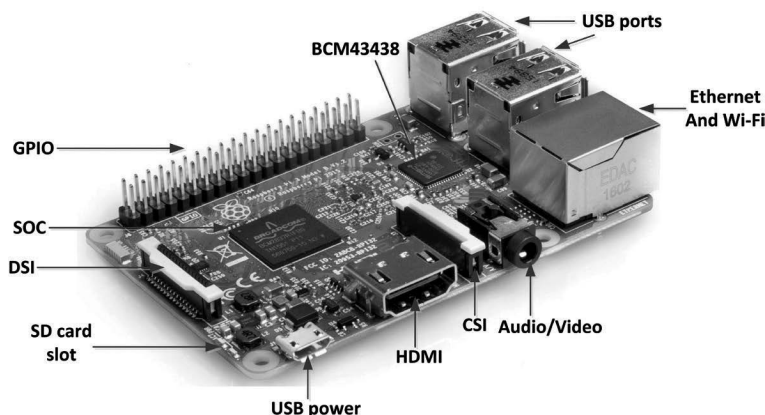
### 11.2 Računar Raspberry Pi

Računar Raspberry Pi je jeftin, snažan računar na jednoj pločici sposoban da radi sa punim operativnim sistemom i sposoban da radi sve što laptop ili stoni računar može da radi kao što je pravljenje i redigovanje dokumenata, izlaska na internet, prijem i slanje elektronske pošte, igranje igrica, razvoj programa i kontrola okoline preko elektronskih senzora i aktuatora i još mnogo toga.

Postoje nekoliko različitih modela Raspberry Pi, sa blago različitim karakteristikama. Osnovne osobine svih računara Raspberry Pi su slične, sve koriste ARM procesor, sve imaju operativni sistem instaliran na SD kartici, sve imaju memoriju na pločici i ulazno izlazne konektore. Neki modeli kao što je Raspberry Pi 3 i Raspberry Pi Zero W imaju ugrađene Wi-Fi i Bluetooth mogućnosti čineći ih lakim za izlazak na internet i komunikacije sa sličnim uređajima koji imaju Wi-Fi ili Bluetooth sposobnosti. U ovom poglavlju ćemo koristiti Raspberry Pi 3 koji je trenutno jedan od najpopularnijih modela Raspberry Pi.

#### 11.2.1 Pločica Raspberry Pi 3

Slika 11.1 prikazuje pločicu Raspberry Pi 3 sa označenim glavnim komponentama. Neki detalji svake od tih komponenti su dati u ovom odeljku.



Slika 11.1 Pločica Raspberry Pi 3

**USB portovi:** Raspberry Pi ima 4 USB porta da bi se povezali miš, tastatura, kamera itd.

## 11.4 Projekt 19 • Prikaz temperature na mobilnom telefonu

### 11.4.1 Opis

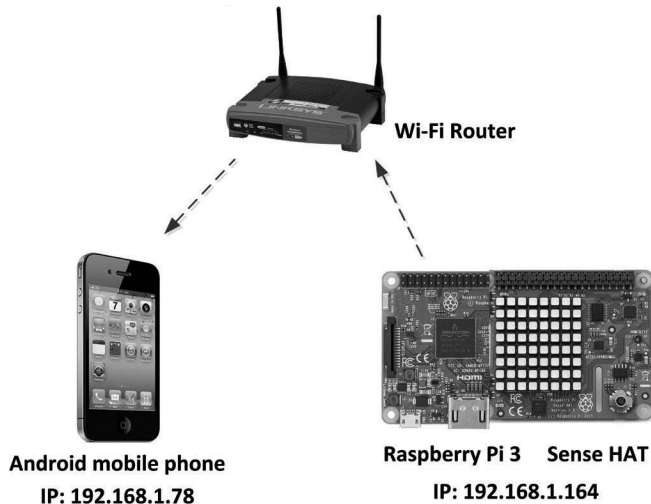
U ovom projektu temperatura okoline se očitava pomoću Raspberry Pi 3i šalje kao UDP paket na Android mobilni telefon koji onda prikazuje temperaturu na ekranu u Edit Text box-u svake sekunde. Senzorska pločica **Sense HAT** se koristi sa Raspberry Pi 3 za očitavanje temperature okoline.

### 11.4.2 Svrha

Svrha projekta je da pokaže kako se očitava temperatura preko pločice **Sense HAT**, vezane na Raspberry Pi 3 i kao se ti podaci šalju i prikazuju na Android mobilnom telefonu.

### 11.4.3 Blok dijagram

Slika 11.18 prikazuje blok dijagram projekta. Pločica Sense HAT je povezana na vrh računara Raspberry Pi3. Komunikacija između Raspberry Pi 3 i Android mobilnog telefona se vrši preko lokalnog Wi-Fi rutera uz pomoć UDP protokola.



Slika 11.18 Blok dijagram projekta

### 11.4.4 Pločica Sense HAT

Vredi pogledati detalje pločice Sense HAT pre nego što se upotrebi u projektu sa računom Raspberry Pi 3.

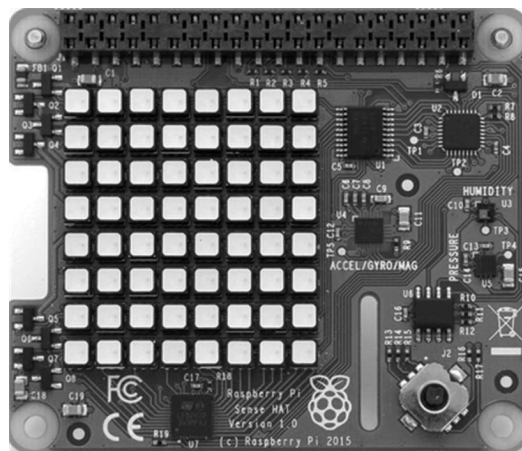
Sense HAT je dodatna pločica za Raspberry Pi i sadrži brojne korisne senzore i niz LED. HAT je skraćenica od **H**ardware **A**ttached on **T**op. Sense HAT je važna komponenta projekta Astro Pi koji je bio obrazovni projekat Raspberry Pi poslat na međunarodnu svemirsku stanicu sa britanskim astronautom Tomom Pikom da pokrene kod koji su pisala deca. Stvarni Astro Pi ima neke modifikacije i poseduje metalno kućište da bi bio pogodan za rad u svemiru.

Pošto HAT poseduje senzore za merenje temperature, vlažnosti, pritiska, ubrzanja, ima žiroskop i magnetometar a pored toga ima na pločici matricu niza 8x8 programabilnih LED koje se mogu programirati da prikazuju tekst i male slike. Slika 11.19 prikazuje pločicu Sense HAT. Na pločici možemo prepoznati sledeće komponente:

Niz od 8x8 LED sa kolor rezolucijom od 15 bita

Čip koji sadrži akcelerometar, žiroskop i magnetometar da meri brzinu, orijentaciju i jačinu i pravac magnetskog polja

- Čip sa sensorom temperature i vlažnosti
- Senzor barometarskog pritiska sposoban da vrši merenje pritiska malih molekula vazduha
- Čip grafičke kontrole
- Džojstik sa pet tastera za pokrete levo, desno, gore, dole i unutar



Slika 11.19 Pločica Sense HAT

Biblioteka Sense HAT se mora uvesti u vaš Python program i na početku programa se mora kreirati objekat **sense**, tj., sledeća dva iskaza se moraju uključiti na početak programa:

```
from sense_hat import SenseHAT
sense = SenseHAT()
```

### Očitavanje temperature, pritiska i vlažnosti

Da bi se očitala temperatura, pritisak i vlažnost okoline moraju se upotrebiti sledeći iskazi na Sense HAT pločici.

## Poglavlje 12 • Android i Raspberry Pi SMS interfejs

### 12.1 Pregled

U ovom poglavlju ćemo pogledati kako jedan Android uređaj (mobilni telefon ili tablet) može komunicirati sa računarom Raspberry Pi 3 slanjem SMS tekst poruka. U ovom poglavlju biće korišćena pločica za nadogradnju/ modul (eng. Shield) SIM800 GSM/GPRSD da bi se dobila SMS mogućnost na računaru Raspberry Pi.

Pre detalja projekta vredi kratko pogledati specifikacije pločice SIM800C.

### 12.2 SIM800C

SIM800C GSM/GPRS je modem sa četiri opsega podešen na interfejs Raspberry Pi-ja. Modem kontrolišu standardne AT komande preko serijske veze.

SIM800C ima sledeće osobine:

- Kompatibilna sa Raspberry Pi
- Četvorostruki opseg u opsezima 850/900/1800/1900 MHz
- Priključci za zvučnik i mikrofon
- Mikro USB konektor
- 'Grove' interfejs
- Eksterno napajanje 5 V, 2 A, opseg 5 - 20 V
- Podrazumevana brzina serijske komunikacije 9600 bod, opseg 1200 - 115200
- Prorez (slot) za SIM karticu

Postoji nekoliko modela nadgradnje SIM800C a slika 12.1 prikazuje jedan takav model. Modul (shield) je opremljen antenom. Slot SIM kartice je pločici. Modul se može montirati direktno na vrh Raspberry Pi. Napajanje za modul mora biti eksterno preko mikro USB priključka sa napajanjem od 5 V (2.5 A) ili vezivanjem napajanja na dva izvoda sa leve strane na žutom konektoru (j1) na gornjem levom uglu pločice (slika 12.2) Pin sa leve strane je +5 V a onaj pored je pin mase (GND). Napajanje može napajati i modul i Raspberry Pi. Priključci za mikrofon i zvučnik su na pločici. Na pločici se takođe nalazi 'Grove' interfejs.

Pre korišćenja nadgradnje (šilda) proverite da li ste ubacili odgovarajuću karticu u slot i takođe vežite antenu na odgovarajući priključak



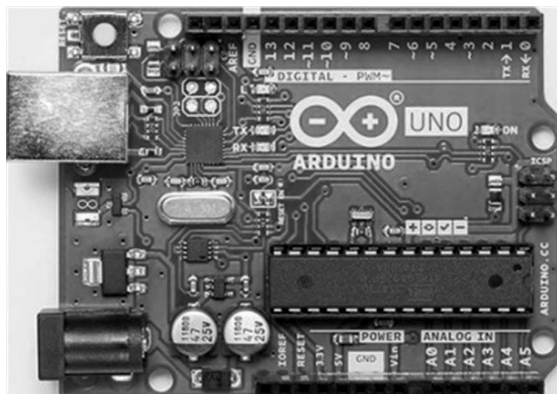
## Poglavlje 13 • Android na Arduino Wi-Fi interfejs

### 13.1 Pregled

U ovom poglavlju ćemo pogledati kako Android uređaj (mobilni telefon ili tablet) može komunicirati sa računarom Arduino Uno preko Wi-Fi linka koristeći UDP protokol. Pre detaljnog pregleda interfejsa Android na Arduino važno je pogledati neke osnovne osobine Arduino Uno za one čitaoce koji su novi u svetu Arduino.

### 13.2 Arduino Uno

Arduino Uno je trenutno, možda najpopularniji računar na jednoj pločici u okolini. Arduino je platforma otvorenog koda podržana mnoštvom biblioteka čineći ga veoma lakim za razvoj projekata. To je jeftin i korišćen je u hiljadama jednostavnih kao i složenih projekata tokom godina gde je učestvovalo mnogo ljudi. Na internetu postoji ogroman broj informacija o arhitekturi i programiranju familije računara Arduino. Izdato je puno knjiga sa različitim projektima da nauče ljude korišćenju Arduino pločica. Neki od razloga zašto je Arduino familija popularna godinama su: niska cena, jednostavno programsko okruženje, jednostavno prebacivanje kompletnog koda na ciljnu pločicu, otvoren kod, veliki broj besplatnih biblioteka, proširene biblioteke i veliki broj uputstava, tehničkih podataka, projekata i primera koda na internetu. U ovom poglavlju ćemo koristiti Arduino Uno pločicu u Android baziranim projektima. Arduino Uno je početnička pločica. Slika 13.1 prikazuje pločicu Arduino Uno. Neki detalji Arduino Uno pločice su dati u ovom odeljku.



Slika 13.1 Pločica Arduino Uno

#### Osobine Arduino Uno

Mikrokontroler	ATmega328P
Brzina takta	16 MHz
Radni napon	5 V
Ulazni napon	7 - 12 V
Digitalni ul. /izl. izvodi	14 (6 PWM izlaza)

## Poglavlje 14 • SMS interfejs Android na Arduino

### 14.1 Pregled

U ovom poglavlju ćemo razviti projekt da naučimo kako Android uređaj (mobilni telefon ili tablet) može komunicirati sa Arduino Uno preko SMS poruka.

Pre detalja projekta vredno je pogledati ukratko detalje SMS poruka i takođe, kako se koristi modul GSM/GPRS sa Arduino Uno da šalje i prima SMS poruke

### 14.2 SMS poruke

SMS (Short Message Service) se koristi za slanje i prijem poruka između mobilnih uređaja. Prvi put se koristilo 1992 godine i upisano u GSM standarde a trenutno je podržan od strane svih GSM mobilnih telefona. Poruka SMS je kratka i može sadržati sve do 160 znakova podataka sa 7-bitnim kodiranjem (kao i engleski znaci) i 70 znakova podataka sa 16-bitnim Unicode kodiranjem (kao kineski znaci). Pored teksta binarni podaci mogu biti slani i primani preko SMS servisa poruka. Zato slike, logo i tonovi zvona i drugi binarni podaci mogu biti obrađeni SMS porukama.

Neke prednosti SMS poruka:

- SMS poruke se mogu slati i primati u bilo koje doba dana
- SMS poruke se mogu slati na telefon koji nije uključen. GSM sistem čuva poruku i šalje je na izabrani telefon kada se priključi na mrežu
- SMS poruke podržavaju svi mobilni telefoni
- SMS poruke se mogu slati i primati na svim jezicima
- Upozorenja i napomene, kao što su transakcije na kreditnim karticama, promene na berzi i tome slično se mogu poslati preko SMS poruka
- Daljinski uređaji se mogu nadzirati i kontrolisati korišćenjem SMS poruka.
- SMS poruke se mogu koristiti za oglase i kao marketinški alat.
- SMS poruke se mogu upotrebiti za učenje

Sms porukama upravlja SMS centar (SMSC). Kada se SMS pošalje prvo stiže do SMS centra odakle se zatim šalje na odredište ako je priključeno na mrežu ili smešta ako je odredišni uređaj isključen sa mreže a isporučuje se primaocu kada postane ponovo prisutan. Broj SMS mrežnog operatorskog centra se mora uneti u uređaj pre korišćenja SMS usluga.

SMS poruka se privremeno smešta ako je prijemni uređaj van mreže. Ovo vreme može biti određeno od strane pošiljaoca (naziva se period važnosti -validity period) gde se posle tog vremena poruka briše od strane SMS centra i poruka neće biti predata na odredište.

SMS poruke mogu biti poslane na uređaje na istoj mreži ili na različite mreže. Kada se koriste različite mreže za rukovanje porukama se koriste SMS tranzicije (gateways). Poruke se mogu slati i međunarodno. Poruke se mogu slati u dva formata koristeći AT naredbe: preko PDU (Protocol Description Unit) format i TEXT format. PDU format nudi

da šalje binarne podatke kao 7-bitne ili 8-bitne i pogodan je kada se žele slati komprimovani podaci ili binarni podaci. PDU format se sastoji od heksadecimalnog stringa znakova, uključujući adresu SMS centra usluga, broj onog ko šalje, vremena itd. U Text formatu se šalju i primaju alfanumerički podaci do 160 znakova sa 7-bitnim kodiranjem i 140 znakova sa 8-bitnim kodiranjem.

### 14.2.1 Slanje i prijem u Text modu

#### Slanje

Slanje SMS u Text modu je vrlo lako i primer je dat na slici 14.1 koji pokazuje poruku Sending test SMS Message! koja je poslata na mobilni broj +447415987063. Pretpostavlja se da je broj SMS centra usluga već postavljen na SIM kartici i da je uneta lozinka:

- Postavite SMS format na Text:

```
AT+CMGF=1
```

- Podesite znakove za GSM:

```
AT+CSCS="GSM"
```

Podesite SMS parametre:

```
AT+CSMP=17,167,8,241
```

Podesite prijemni mobilni telefonski broj i poruku:

```
AT+CMGS="+447415987053"  
>Sending test SMS Message![Ctrl-z]
```

```
AT+CMGF=1  
OK  
AT+CSCS="GSM"  
OK  
AT+CSMP=17,167,0,241  
OK  
AT+CMGS="+447415987053"  
> Sending test SMS Message!→  
+CMGS: 4  
OK
```

Slika 14.1 Slanje SMS tekst poruke

Zapazite da posle unosa prijemnog broja modem odgovara sa znakom > i u tom trenutku se mora uneti poruka koja se šalje. Poruka se mora završiti znakom **Ctrl-Z**.

## Poglavlje 15 • Android na ESP32 Wi-Fi Interfejs

### 15.1 Pregled

U ovom poglavlju ćemo pogledati kako Android uređaj (mobilni telefon ili tablet) može komunicirati sa ESP32 procesorom preko Wi-Fi veze uz pomoć UDP protokola. U ovom poglavlju ćemo programirati procesor ESP32 uz pomoć Arduino IDE. Pre detaljnijeg razmatranja interfejsa Android na ESP32 vredno je razmotriti osnovne funkcije procesora ESP32 za one čitaoce koji su možda novi u svetu ESP32.

### 15.2 Procesor ESP32

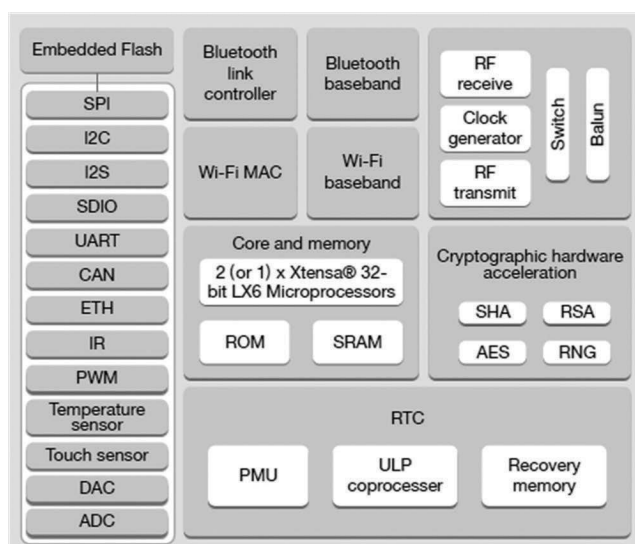
Procesor ESP32 je razvila firma Espressif, ona koja je razvila procesor ESP8266. Iako ESP32 nije razvijen da zameni ESP8266 on ga poboljšava u mnogo aspekata. Novi procesor ESP32ne samo da ima podršku za Wi-Fi nego, takođe, ima Bluetooth komunikacioni modul čineći da procesor može komunicirati sa Bluetooth kompatibilnim uređajima. CPU procesora ESP32je 32-bitna Xtensa LX6 slična CPU od ESP8266 ali ima dva jezgra, više memorije za podatke, više GPIO, veću brzinu takta CPU, ADC pretvarače sa većom rezolucijom, DAC pretvarač i mogućnost vezivanja na CAN sabirnicu. Osnovne specifikacije procesora ESP32 su sabrane dole:

- 32-bitni Xtensa RISC CPU: Tensilica Xtensa LX6 dvojezgarni mikrokontroler
- Brzina rada 160 do 240 MHz
- 520KB SRAM memorije
- 448 KB ROM
- 16 KB SRAM (u RTC)
- IEEE 802.11 b/g/n/e/I Wi-Fi
- Bluetooth 4.2
- 18 kanala x 12 bita ADC
- kanala x 8 bita DAC
- 10 senzora osetljivih na dodir (tač)
- senzor temperature
- 36 GPIO
- 4 x SPI
- 2 x I2C
- 2 x I2S
- 3 x UART
- 1 x CAN sabirnica (CAN bus 2.0)
- Podrška za SD memorijsku karticu
- rad na 2.2 V - 3.3 V
- RTC tajmer i
- Holov senzor (Hall)
- 16 kanala PWM
- Ethernet interfejs
- Interni takt 8 MHz, RC oscilator

- Eksterna 2 MHz - 60 MHz, i 32 kHz oscilator
- hardversko kriptografsko ubrzanje (AES, HASH, RSA, ECC, RNG)
- IEEE 802.11 bezbednosne karakteristike
- struja u servisnom režimu rada 5  $\mu$ A

### 15.2.1 Arhitektura ESP32

Slika 15.1 prikazuje funkcionalni blok dijagram procesora ESP32 (vidi tehničke podatke **ESP32, Expressif Systems, 2017**). U srcu bloka je dvojezgarni Xtensa LX6 procesor i memorija. Sa leve strane možemo videti blok perifernog interfejsa kao što je SPI, I2C, I2S, SDIO, UART, CAN, EZH, IR PWM, temperaturni senzor, tač senzor, DAC i ADC. Bluetooth i Wi-Fi moduli su smešteni u sredini na vrhu blok dijagrama. Generator takta i RF primopredajnik su smešteni u bloku gore desno. Sredina desno je rezervisana za hardverske module ubrzanja kriptovanja kao što su SHA, RSA, AES i RNG. Na kraju sredina dole je deo gde je RTC. PMU, koprocesor i memorija za oporavak.



Slika 15.1 Funkcionalni blok dijagram procesora ESP32

Slika 15.2 prikazuje strukturu sistema koja se sastoji od dva CPU jezgra Hardvard arhitekture po imenu PRO\_CPU (za protokol CPU) i APP\_CPU (CPU za aplikacije). Moduli u sredini dva CPU su zajednički za oba CPU. Detaljne informacije o internoj arhitekturi ESP32 se može dobiti iz **ESP32 Technical Reference Manual** (Expressif Systems, 2017).

## Prilog A • Upotreba Android emulatora

Android emulator je Virtual Device manager (AVD) koji imitira Android mobilni telefon. AVD obezbeđuje Google kao deo njihove podrške korisnicima SDK za vreme razvoja Android baziranih aplikacija. Kao što ćemo videti kasnije u prilogu AVD radi sa B4A IDE klikom na **Tools—> Run AVD Manager**.

Emulator može biti koristan za vreme razvoja programa gde programer može emulirati rad stvarnog mobilnog telefona na ekranu PC-ja. Emulator radi na PC-ju bez potrebe za stvarnim telefonom. Iako emulator može biti koristan alat autor smatra boljim upotrebu stvarnog mobilnog telefona kao i instalaciju i test aplikacije na stvarnim uređajima za vreme razvoja programa. Možda je najveća prednost upotrebe emulatora da se razvijeni program može ispitati na različitim modelima Android mobilnih telefona raznih veličina.

U ovom prilogu ćemo kreirati nekoliko veoma jednostavnih aplikacija i videti kako rade na emulatoru.

### Primer A1

U ovom primeru će biti prikazana poruka **Hello from the emulator** na emulatorskom ekranu.

Koraci početaka i upotrebe emulatora su dati dole:

- Pokrenite B4A IDE
- Kliknite **Tools—> Run AVD Manager**
- Kreirajte novi uređaj i API nivo (level)
- Trebalo bi da vidite Android emulatorski ekran kao na slici A.1 sa tasterima sa desne strane a ekran sa leve



Slika A.1 Ekran emulatora

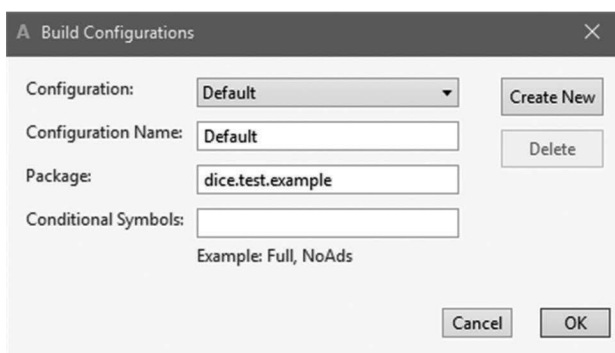
## Prilog B • Objavljanje aplikacije na Google Play

Možda želimo da objavimo i prodamo našu aplikaciju na Google Play. U ovom prilogu ćemo napraviti veoma jednostavnu aplikaciju i videti kako se može objaviti na veb sajtu Google Play.

Primer aplikacije razvijen u ovom prilogu je jednostavna kockica. Imaćemo taster i Edit Text box (kao u prilogu A, primer A2). Klikom na taster će generisati slučajan broj kockice od 1 do 6.

### B1 Razvoj aplikacije za Google Play

Svaka B4A aplikacija mora imati jedinstveno ime paketa. Podrazumevano ime je B4A. Naziv paketa (Package Name) se može postaviti klikom na **Project**—> **Build Configuration** kao što je prikazano na slici B.1. Ostavimo polje Configuration na **Default** tako da se izbor moda rada DEBUG ili RELEASE bira automatski u zavisnosti od tekućeg stanja kompajlera. Dajte ime polju **Configuration Name**. Naziv paketa je važan i trebalo bi da bude jedinstveno ime u vašem projektu. U ovom primeru postavićemo da naziv paketa bude **dice.test.example**. Naziv paketa mora sadržati najmanje dve reči odvojene tačkom. Ostavite praznim **Conditional Symbols**.



Slika B.1 Konfiguracija Build

Sada bi trebalo konfigurisati **Project Attributes** i **Activity Attribute** po želji. Jedna aplikacija mora, takođe, da ima svoju ikonu. Postoje nekoliko tipova ikona kao što su 'tab', 'status bar', 'menu', 'launcher' itd. U ovom primeru ćemo koristiti ikonu kompatibilnu sa Google Play. Ikona Google Play bi trebalo da je PNG fajl sa 512 x 512 tačkica. Veličina ikone ne bi trebalo da je veća od 1 MB. Za pravljenje ikona na internetu postoji dosta besplatnih programa. Takođe, postoje programi koji pretvaraju slike u ikone. U ovom primeru je korišćena ikona kockice sa 512 x 512 tačkica. Kliknite **Project**—>**Choose Icon** i izaberite ikonu koju želite da koristite u vašoj aplikaciji.

Sada, kreirajte vašu aplikaciju kockice. Dodajte taster i Edit Text box tako da kada se pritisne taster, ispiše se broj između 1 i 6 u Edit Text box-u. Nazovite vaš program aplikacije i dizajnirani izgled kao **exampledice**.