

<b>Nastavni predmet:</b>	<b>Ugradbeni računalni sustavi</b>
<b>Vježba: 12</b>	<b>Arduino – Bluetooth komunikacija</b>
<b>Cilj vježbe:</b>	Konfigurirati Bluetooth modul pomoću AT naredbi. Pomoću Bluetooth modula i Android aplikacije upravljati LED diodom, relejom, prikazati poruku s Android aplikacije na LCD zaslonu, te prikazati A/D vrijednost s potencijometra na Android aplikaciji. Kreirati vlastitu Android aplikaciju.

### Upute za izradu pripreme za vježbu:

Budući da je mikroupravljač računalo na čipu koji se programira kako bi upravljao priključenim vanjskim elektroničkim komponentama, priprema za vježbu se sastoji od dva dijela:

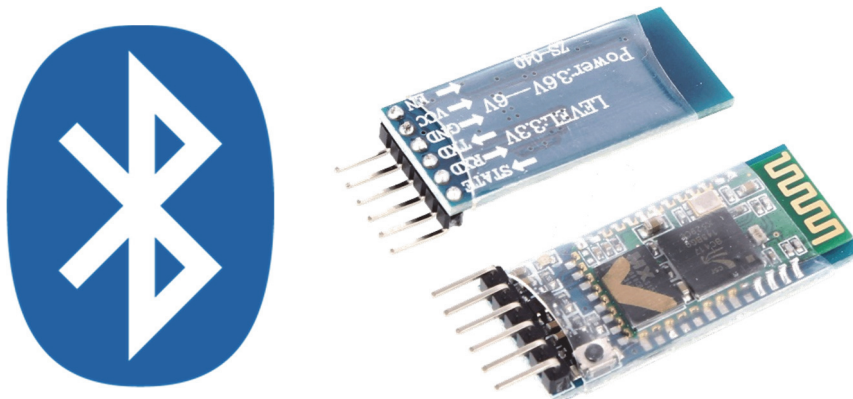
1. **Opis elektroničkih komponenti koje će se koristiti na LV** – proučiti tekst u nastavku, proanalizirati i u bilježnicu ispisati najvažnije informacije za elektroničke komponente.
2. **Opis naredbi korištenih u LV** – proanalizirati programski kod za sve zadatke, ispisati nove naredbe i funkcije, objasniti njihovu namjenu i argumente. Ako ne možeš pronaći sve informacije u kodu priloženih zadataka, posluži se internetom npr. [www.arduino.cc](http://www.arduino.cc) ...

### Upute za izradu vježbe:

- Svaki zadatak treba prije prevođenja (eng. compile) pohraniti u napravljeni folder na Desktopu, tako da, u slučaju pogreške (HW, SW) imaš sačuvan kod.
- Na kraju LV, sve zadatke spremi na USB ili pošalji na svoj mail.
- Nazivi datoteka, zbog preglednosti, neka budu: LV01\_ZAD01, LV01\_ZAD02, itd.
- Vježbe se rade u paru, preporuka - jedan učenik spaja komponente, drugi piše programski kod, a na slijedećoj vježbi se uloge zamjenjuju.
- U zadacima koji zahtijevaju samostalno rješavanje, oba učenika sudjeluju u spajanju i programiranju.
- Za pojedini zadatak potrebno je u bilježnicu nacrtati električnu shemu s vidljivim oznakama korištenih pinova i vezu istih s oznakama u programskom kodu.
- Dobiveno rješenje treba komentirati, tj. dati zaključak što je novo u tom zadatku i kako je to riješeno, ukratko ispisati važniji dio koda (ne prepisivati cijeli kod) te navesti eventualne probleme i kako su isti riješeni.
- Ako su uz neki zadatak postoje pitanja, potrebno je u bilježnicu odgovoriti na ista.
- Ako u kodu postoji greška (negdje će biti namjerno stavljena) kod treba korigirati i objasniti!
- Budući da se na vježbama koriste stvarne komponente, postoji mogućnost da je neka neispravna (pregorena LED, oštećen kontakt tipkala, prekinut vodič...). Ukoliko se sklop ponaša drugačije od očekivanog, predvidjeti i tu mogućnost i pokušati zamijeniti komponentu drugom.
- Prilikom spajanja, za Vcc (+5V) koristi crveni vodič, a za GND (-) crni vodič. Za ostale signale koristiti ostale boje.
- Za zadatke koje nisi stigao odraditi na vježbi, treba kod kuće razmisliti kako bi ih riješio
- Po završetku izvođenja vježbe, na temelju odrađene pripreme te riješenih zadataka, očekuje se da učenik zna odgovoriti na pitanja na kraju ovih materijala.
- Pregledavanje priprema i provjeravanje znanja bit će na svakoj LV, uključujući i prethodne vježbe

## Bluetooth komunikacija i HC-05 Bluetooth modul

Bluetooth modul HC-05 omogućuje bežičnu komunikaciju između dva uređaja. Veza se ostvaruje preko radio valova frekvencije 2,4 GHz putem Bluetooth protokola. Za povezivanje na Arduino, modul HC-05 koristi serijsku komunikaciju i spaja se preko Rx i Tx pinova.



Slika 1. Bluetooth Logotip i HC-05 Bluetooth modul

Iako HC-05 modul dolazi s predefiniranim postavkama (brzina komunikacije 38400 bauda, naziv modula HC-05 i lozinkom „1234“, na laboratorijskim vježbama je zbog većeg broja jednakih modula potrebno promijeniti standardne postavke.

Da bi mogli **promijeniti postavke**, potrebno je **HC-05 pokrenuti u AT konfiguracijskom modu rada**. Za komunikaciju (prijenos naredbi i informacija) potrebno je na Arduino upisati program **HC\_05.ino** (Zadatak 1). Taj program služi samo za podešavanje modula. Kad se završi s podešavanjem, na Arduino se upisuje program koji će izvršavati zadani zadatak (Zadatak 2).

**AT naredbe** koriste se za upravljanje i podešavanje različitih vrsta modema. **Modemi** su uređaji koji služe za komunikaciju između dva računala. Naziv Modem dolazi od riječi **MOD**ulacija/**DEM**odulacija. U vrijeme pojave modema, komunikacija se odvijala putem analognih (telefonskih) linija, pa se digitalni signal iz računala pretvarao u analogni signal radi prijenosa podataka, a potom opet iz analognog u digitalni. Modemi mogu biti žičane ili bežične varijante.

**AT** je skraćenica od **Attention**. Ove naredbe razvio je Denis Hayes za upravljanje Hayes Smartmodem 300 bauda još davne 1981. godine. AT naredbe sastoje se od kratkih tekstualnih nizova, koji se mogu kombinirati, a pomoću njih se pokreće primjerice akcija dizanja slušalice, biranja broja, završetka razgovora i spuštanja slušalice, podešavanje parametara komunikacije i sl.

Primjeri AT naredbi:

AT <enter> - provjeravanje li aktivna komunikacija s modulom

ATA <enter> - naredba za javljanje na poziv

ATD123456789<enter> - zove upisani broj telefona (123456789)

ATP <enter> - postavlja pulsno biranje

ATT <enter> - postavlja tonsko biranje

Većina današnjih modema koristi Hayesove AT naredbe. AT naredbe mogu se koristiti za podešavanje različitih informacija i servisa:

1. Informacije i konfiguracija vezane uz mobilne uređaje ili modem i SIM kartice
2. SMS servisi

3. MMS servisi
4. Fax uređaji
5. Podatkovna i glasovna veza preko mobilne mreže

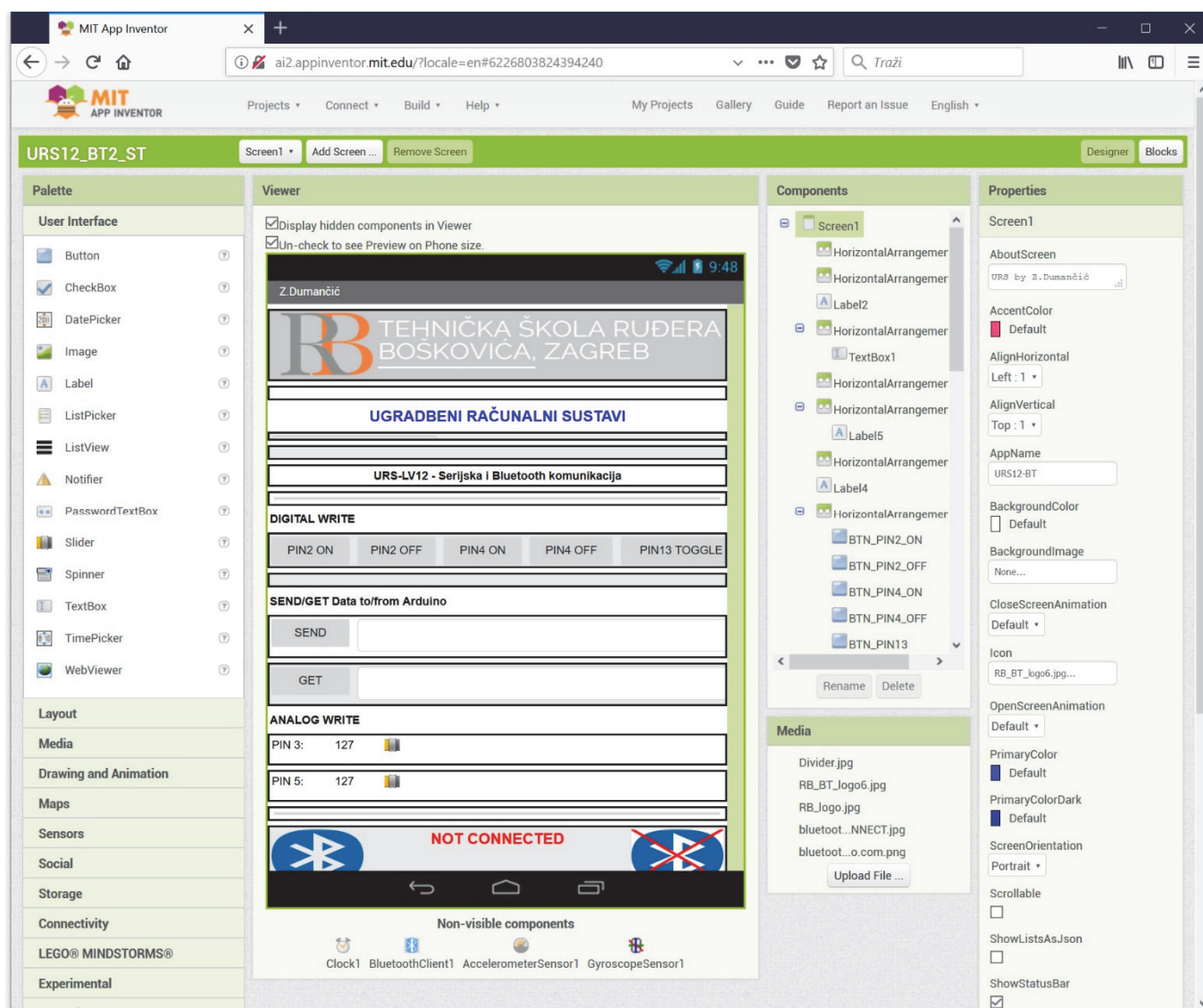
Hayesov niz naredbi naziva se osnovni niz naredbi (Basic commands), dok naredbe specifične za GSM, BT i sl. se zovu prošireni niz naredbi (Extended AT commands).

## MIT App Inventor ([www.appinventor.mit.edu](http://www.appinventor.mit.edu))

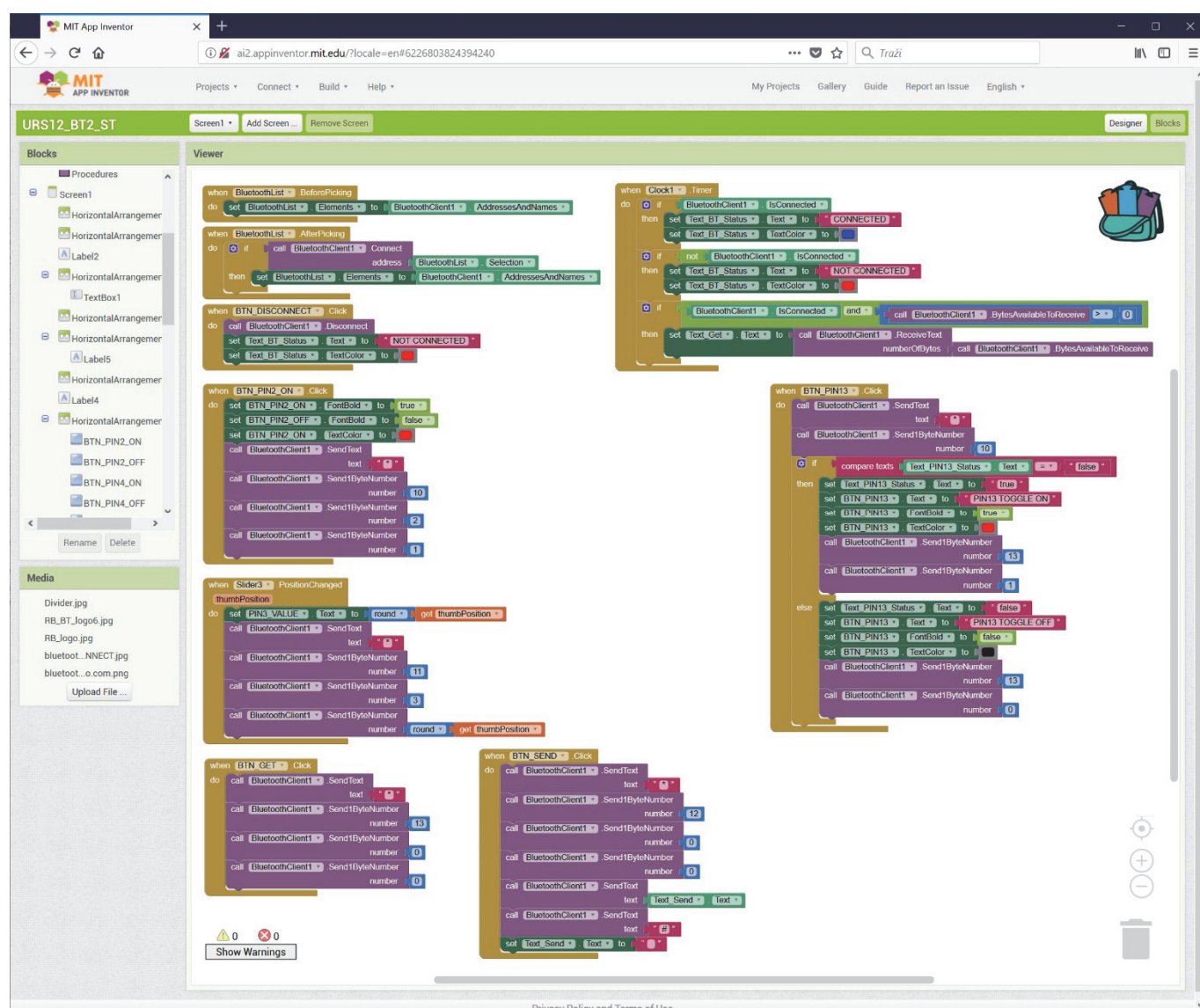
MIT App Inventor je *cloud-based* aplikacija za jednostavno stvaranje mobilnih aplikacija za Android. Aplikacija se stvara tako da se pojedini elementi – kontrole (labele, gumbi, klizači, padajući izbornici,...) postavljaju na željeno mjesto na zaslonu, a potom se za njih piše pozadinski kod koji određuje što će koja kontrola raditi.

Elementi se postavljaju na zaslon u **Designer modu**, a programiranje se bazira na blokovnom sustavu gdje jedan blok predstavlja jednu naredbu. Blokovi se povlače i vizualno sastavljaju kao **puzzle**. To se može vidjeti u **Blocks modu**.

Detaljnije upute se mogu potražiti na web stranici pod literatura [3] i [4].



Slika 2. Designer mod aplikacije MIT APP inventor



Slika 3. Blocks mod aplikacije MIT APP inventor

**NAPOMENA:** Na laboratorijskim vježbama iz predmeta Ugradbeni računalni sustavi koristi se MIT App Inventor aplikacija jer je relativno jednostavna za početnike koji se nisu susretali s programiranjem aplikacija za Android. Cilj ovog predmeta nije edukacija programiranja Android aplikacija, nego što jednostavnije kreiranje aplikacije radi usvajanja znanja serijske i Bluetooth komunikacije. Napredniji programeri mogu Aplikaciju napraviti u Android studio (<https://developer.android.com/studio>) ili sličnoj aplikaciji.



**Zadatak 1.** Na Arduino učitaj program **HC\_05.ino** koji će omogućiti komunikaciju s HC-05 modulom i podešavanje modula pomoću AT naredbi. Nakon što pročitaš cjelokupne upute u nastavku, poveži HC-05 Bluetooth modul prema shemi ispod. Navedenom procedurom promijeni ime BT modula u skladu s nazivom računala, npr. BT\_WS\_01, te promijeni lozinku.

### Kòd zadatka

```
/*
AUTHOR: Hazim Bitar (techbitar)
DATE: Aug 29, 2013
LICENSE: Public domain (use at your own risk)
CONTACT: techbitar at gmail dot com (techbitar.com)
Program se može preuzeti na web stranici pod literatura [1].
*/

#include <SoftwareSerial.h>

SoftwareSerial BTSerial(10, 11); // RX | TX

void setup()
{
    pinMode(9, OUTPUT); // this pin will pull the HC-05 pin 34 (key pin
                        // or EN pin) HIGH to switch module to AT mode
    digitalWrite(9, HIGH);
    Serial.begin(9600);
    Serial.println("Enter AT commands:");
    BTSerial.begin(38400); // HC-05 default speed in AT command mode
}

void loop()
{
    // Keep reading from HC-05 and send to Arduino Serial Monitor
    if (BTSerial.available())
        Serial.write(BTSerial.read());

    // Keep reading from Arduino Serial Monitor and send to HC-05
    if (Serial.available())
        BTSerial.write(Serial.read());
}
```

Program **HC\_05.ino** radi dvije stvari:

1. Preuzima AT naredbu koju ćete utipkati u Serial monitor i šalje ju na HC-05 modul
2. Program čita izlaz s HC-05 modula, i prikazuje ga na Serial monitoru (može se koristiti i neki drugi terminal emulator, npr. Tera term)

Arduino će komunicirati s HC-05 koristeći softverski emulirani serijski port na pinovima 10 (Rx) i 11 (Tx) koristeći biblioteku SoftwareSerial.h, dok se komunikacija sa Serial monitorom odvija preko standardnog hardverskog serijskog porta na pinovima 0 i 1 (Rx i Tx).

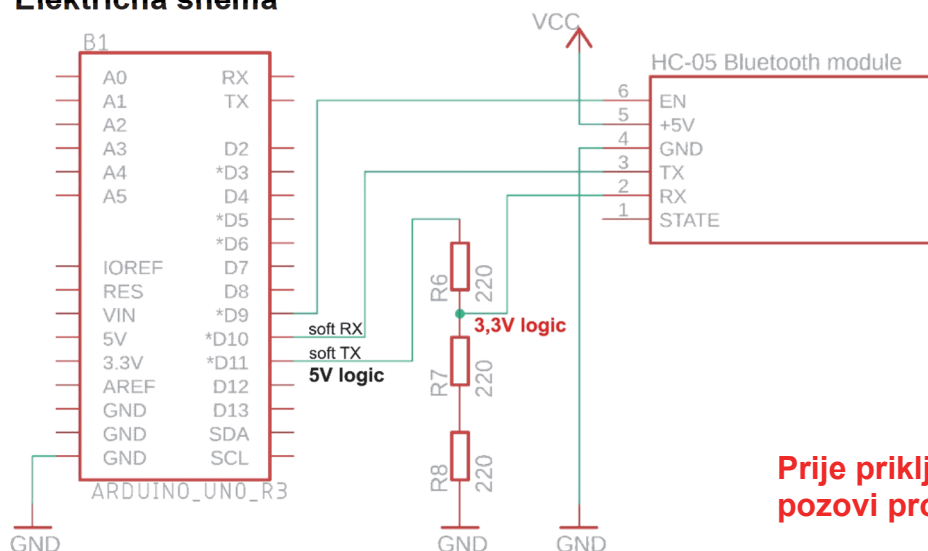
Da bi HC-05 postavili u AT komandni mod, potrebno je nožicu Key (EN) spojiti na visoku razinu (+5V), ali uz određenu proceduru koja će biti dana u nastavku. U komandnom modu, HC-05 komunicira brzinom 38400 bauda.

Za ulazak u komandni mod slijedite slijedeće korake:

1. Povežite HC-05 modul prema priloženoj shemi

**VAŽNO:** RX pin na HC-05 Bluetooth modulu koristi 3,3V digitalnu razinu, pa je prilikom povezivanja na pin Arduino UNO potrebno ugraditi naponsko djelilo koje spušta napon s 5V na 3,3V !!! Ostali pinovi su TTL kompatibilni (5V).

#### Električna shema



#### Raspored pinova:

Arduino UNO Pin	HC-05 Bluetooth modul
9	Key (EN)
10	TXD
11	RXD (3.3V !!!)
5V	VCC
GND	GND
Nije spojeno	STATE

**Prije priključenja Arduino na napon pozovi profesora da pregleda spoj!**

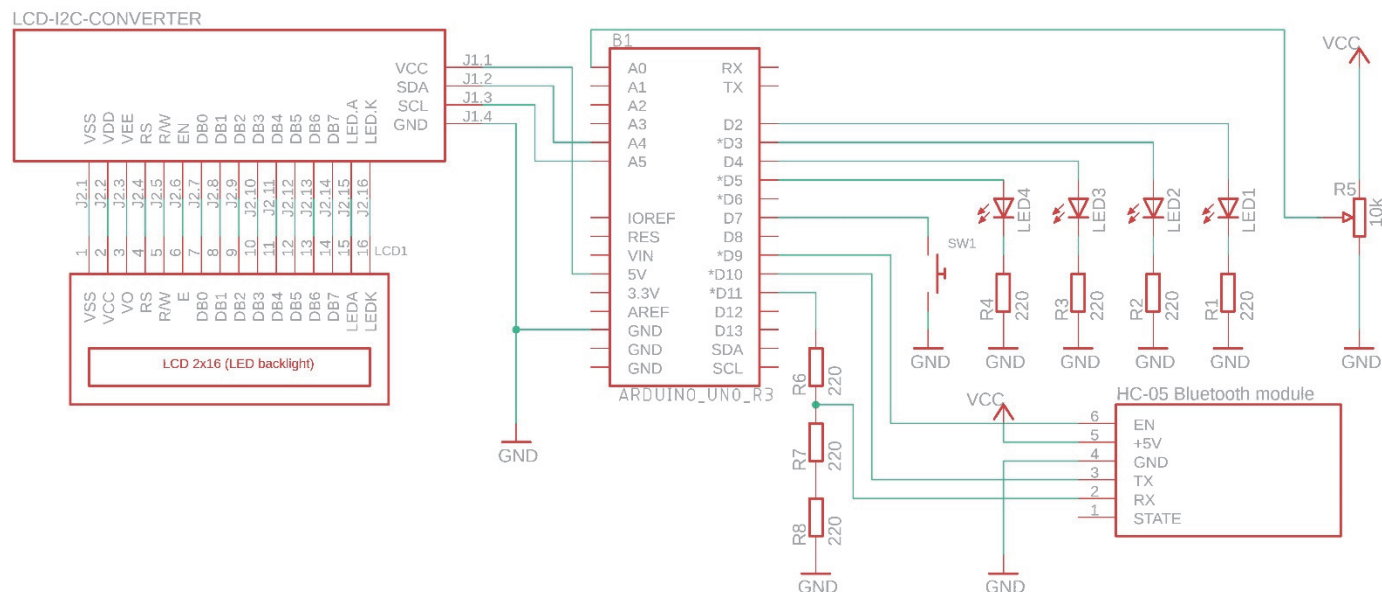
1. PRIJE NEGO POVEŽETE ARDUINO NA USB, odspojite crveni vodič - VCC (napajanje) s HC-05 modula, kako ne bi dobivao napajanje od Arduino. Ostali vodiči trebaju ostati spojeni.
2. Sad povežite Arduino na USB port računala
3. Provjerite da BT modul nije uparen s niti jednim BT uređajem
4. Pritisnite tipku na BT modulu i držite ju pritisnutom
5. Sada povežite 5V s Arduino na VCC od HC-05 modula (crveni vodič) uz pritisnutu tipku na HC-05 modulu
6. LED dioda HC-05 modula će se paliti i gasiti u intervalima od približno 2 sekunde. To označava da je modul u AT komandnom modu, spreman za prihvaćanje naredbi i podešavanje postavki.
7. Otvorite Serial monitor, podesite brzinu prijenosa te podesite „Both NL & CR“
8. Kako bi provjerili da je sve u redu, utipkajte AT i pritisnite SEND. Trebali bi dobiti poruku OK. U suprotnom provjerite ožičenje.

U nastavku se nalaze nekoliko interesantnih AT naredbi:

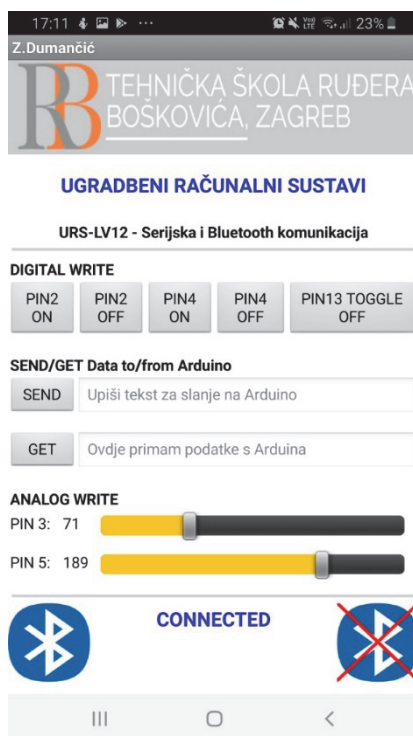
- a) Vraćanje HC-05 na standardne postavke: **AT+ORGL**
  - b) Dobivanje informacije o verziji HC-05 modula: **AT+VERSION?**
  - c) Za promjenu imena iz standardnog HC-05 u BT\_WS\_01 utipkati: **AT+NAME=BT\_WS\_01**
  - d) Za promjenu pristupnog pina iz standardnog 1234 u npr. 4321 utipkati: **AT+PSWD=4321**
  - e) Za promjenu brzine komunikacije HC-05 sa standardne 9600 na 115200, 1 stop bit, 0 parity utipkati: **AT+UART=115200,1,0**
9. Nakon što podesite željene postavke, odspojite napajanje Arduino (USB) i ponovo priključiti napajanje. Na taj način će se HC-05 pokrenuti u normalnom moda rada.

**Zadatak 2.** Proširi spoj iz prethodnog zadatka u skladu s električnom shemom. Potrebno je dodati 4 LED diode, tipkalo, potencijometar i LCD zaslon.

### Električna shema



**Zadatak 3.** Iz repozitorija preuzmi „URS12\_BT.apk“ i instaliraj na Android mobitel. To je gotova i potpuno funkcionalna aplikacija za testiranje koda iz zadatka 4.



**Zadatak 4.** Koristeći program u nastavku potrebno je pripremiti Arduino za prihvatanje informacija sa HC-05 Bluetooth modula i upravljanje priključenim komponentama (dijelove programa moguće je kopirati iz prethodne vježbe „URS-LV11 - Serijska komunikacija.pdf“ . Provjeri kod i po potrebi prepravi.

### Kod zadatka

/\*  
Neki dijelovi koda preuzeti iz:

```
PROJECT: ArduDroid
PROGRAMMER: Hazim Bitar (techbitar at gmail dot com)
DATE: Oct 31, 2013
FILE: ardudroid.ino
LICENSE: Public domain
*/

#define START_CMD_CHAR '*'
#define END_CMD_CHAR '#'
#define DIV_CMD_CHAR '|'

#define CMD_DIGITALWRITE 10
#define CMD_ANALOGWRITE 11
#define CMD_TEXT 12
#define CMD_SEND 13

#define PIN_HIGH 1
#define PIN_LOW 0

#include <JC_Button.h>           // https://github.com/JChristensen/JC_Button
const byte pin_SW1 = 7;
Button myButton(pin_SW1);      // kreiranje objekta myButton

int PIN_POT_A0 = A0;
int nPOT_A0_value = 0;

#include <SoftwareSerial.h>
SoftwareSerial BTSerial(10, 11); // Software RX | TX

String inText;

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3f, 16, 2);

void setup()
{
    Serial.begin(9600);          // inicijalizacija HARDVERSKE serijske komunikacije
                                // za ispis na Serial monitor
    BTSerial.begin(38400);       // inicijalizacija SOFTVERSKE serijske komunikacije
                                // za komunikaciju s BT modulom preko pinova 10 i 11
    myButton.begin();            // inicijalizacija objekta myButton

    pinMode(2, OUTPUT);          // pin 2 za LED izlazni
    pinMode(3, OUTPUT);          // pin 3 za LED izlazni
    pinMode(4, OUTPUT);          // pin 4 za LED izlazni
    pinMode(5, OUTPUT);          // pin 5 za LED izlazni
    pinMode(13, OUTPUT);         // pin 13 za LED izlazni

    pinMode(PIN_POT_A0, INPUT);

    lcd.begin();                 // inicijaliziranje LCD-a
    lcd.backlight();             // uključivanje pozadinskog osvjetljenja LCD-a
    lcd.setCursor(0, 0);
    lcd.print("Cekam tekst...");

    Serial.print("Start...\n");
```



```
}

void loop()
{
    static bool ledState;          // varijabla koja sadrži stanje LED
    myButton.read();               // čitanje stanja tipkala

    nPOT_A0_value = analogRead(PIN_POT_A0);

    if (myButton.wasPressed())     // ako je tipkalo bilo jednom pritisnuto
    {
        ledState = !ledState;
        digitalWrite(13, ledState);
        BTSerial.print(nPOT_A0_value);    // Slanje vrijednosti na SOFTVERSKI
                                           // serijski port 10(RX) 11(TX)
    }

    BTSerial.flush();              // čeka dovršetak primanja znaka na serijski spremnik

    if (BTSerial.available() < 1) return; // čekam podatke sa softverskog
                                           // serijskog porta

    char start_flag = " ";
    int ard_command = 0;
    int pin_num = 0;
    int pin_value = 0;

    //ard_command = Serial.parseInt(); // čita 1. vrijednost (naredba)
    //pin_num = Serial.parseInt();      // čita 2. vrijednost (PIN)
    //pin_value = Serial.parseInt();    // čita 3. vrijednost (vrijednost)

    start_flag = BTSerial.read(); // čita jedan znak iz serijskog spremnika
    delay(20);

    ard_command = BTSerial.read(); // čita 1. vrijednost (naredba)
    delay(20);
    pin_num = BTSerial.read();      // čita 2. vrijednost (PIN)
    delay(20);
    pin_value = BTSerial.read();    // čita 3. vrijednost (vrijednost)

    Serial.print("start flag\t");
    Serial.print(start_flag);

    Serial.print("\tard_command\t");
    Serial.print(ard_command);

    Serial.print("\tpin_num\t");
    Serial.print(pin_num);

    Serial.print("\tpin_value\t");
    Serial.print(pin_value);

    Serial.print("\n");

    if (start_flag != START_CMD_CHAR) return; // ako znak nije oznaka za početak
                                           // komunikacije, izlaz u loop()
```

```

if (!(ard_command == 10 | ard_command == 11 | ard_command == 12 |
    ard_command == 13)) return; // provjera naredbe

if (ard_command == CMD_DIGITALWRITE) {
    Serial.println("CMD_DIGITALWRITE");
    if (!(pin_num == 2 | pin_num == 4 | pin_num == 13)) return; // provjera
                                                                // pina

    if (pin_value == PIN_LOW) pin_value = LOW;
    else if (pin_value == PIN_HIGH) pin_value = HIGH;
    else return; // za bilo koju drugu primljenu vrijednost
                // vrati se petlji loop()

    digitalWrite(pin_num, pin_value);

    return;
}

if (ard_command == CMD_ANALOGWRITE) {
    Serial.println("CMD_ANALOGWRITE");
    if (!(pin_num == 3 | pin_num == 5)) return // provjera primljenog pina
    pin_value = constrain(pin_value, 0, 255); // ograničava primljenu
                                            // vrijednost u zadani
                                            // interval 0 do 255

    analogWrite(pin_num, pin_value);
    return;
}

if (ard_command == CMD_SEND) {
    Serial.println("CMD_SEND");
    nPOT_A0_value = analogRead(PIN_POT_A0);
    BTSerial.print(nPOT_A0_value); // Slanje vrijednosti na SOFTVERSKI
    return;
}

if (ard_command == CMD_TEXT) {
    Serial.println("CMD_TEXT");
    inText = ""; //clears variable for new input
    while (BTSerial.available()) {
        char c = BTSerial.read(); //prihvati jednog bajta iz spremnika
        delay(10);
        if (c == END_CMD_CHAR) { // kad primim dogovoreni znak za
                                // završetak znakovnog niza
                                // odradi slijedeće

            Serial.print("Primio string: ");
            Serial.println(inText);

            lcd.setCursor(0, 0);
            lcd.print("Primio tekst:");

            lcd.setCursor(0, 1);
            lcd.print(" ");
            lcd.setCursor(0, 1);
            lcd.print(inText);
            break;
        }
    }
    else {

```

```

        if (c != DIV_CMD_CHAR) {
            inText += c;
            delay(10);
        }
    }
}
}
}

```

**Zadatak 5.** Sa Google Play Store na Android mobilni uređaj instalirati **MIT AI2 Companion** aplikaciju pomoću koje je moguće povezivanje i preuzimanje aplikacije koju ćemo napraviti tijekom laboratorijske vježbe.

**Zadatak 6.** Pomoću Google Chrome ili Mozilla Firefox preglednika, otvoriti web stranicu MIT APP Inventor 2 i logirati se pomoću vlastitog Google računa.

**Zadatak 7.** Iz repozitorija preuzmi pripremljeni predložak za izradu laboratorijske vježbe „**URS12\_BT\_Student.aia**“. U MIT App Inventoru odabrati: **Projects → Import project (.aia) from my computer** i importiraj predložak.

U Designer modu:

- Prouči koje su sve kontrole dodane na formu (label, button, slider, textbox,...)
- Klikom na pojedinu kontrolu, prouči naziv svake kontrole u okviru Components
- Prouči svojstva svake komponente u okviru Properties

U Blocks modu:

- U okviru Blocks prouči koje su sve funkcije ponuđene za pojedinu kontrolu
- U okviru Blocks prouči koje su funkcije standardno ugrađene u MIT App inventor (Built-In), neovisno o kontrolama (Control, Logic, Math, Text,...)
- U okviru Viewer prouči postojeće blokove naredbi.

**Zadatak 8.** Predložak za izradu laboratorijske vježbe „**URS12\_BT\_Student.aia**“ nema potpunu funkcionalnost svih tipki. Doradi program tako da dodaš funkcionalnost preostalim kontrolama (tipke, klizači,...).

- Dodavanje kontrole u Designer modu i/ili dodavanje koda u Blocks modu može se u realnom vremenu pratiti na mobilnom uređaju koristeći MIT AI2 Companion aplikaciju
- Iz izbornika na vrhu prozora odabrati: **Connect → AI Companion**
- Pokrenuti **MIT AI2 Companion** aplikaciju na mobilnom telefonu i odabrati **scan QR code**

**Zadatak 9.** Po dovršetku izrade Android aplikacije, kompilirati aplikaciju i prenijeti instalacijsku APK datoteku na mobilni telefon. Iz izbornika na vrhu prozora odabrati: **Build → App (provide QR code for .apk)**, ili App (save .apk to my computer).

## LITERATURA:

1. Wikipedia, Hayes command set, [https://en.wikipedia.org/wiki/Hayes\\_command\\_set](https://en.wikipedia.org/wiki/Hayes_command_set), (pregledano 09. ožujka 2019.)
2. EngineersGarage, AT Commands, GSM AT command set, Nikhil Agnihotri, <https://www.engineersgarage.com/tutorials/at-commands>, (pregledano 09. ožujka 2019.)
3. MIT App Inventor, <http://appinventor.mit.edu/>, (pregledano 09. ožujka 2019.)
4. Howtomechatronics: How To Build Custom Android App for your Arduino Project using MIT App Inventor, <https://howtomechatronics.com/tutorials/arduino/how-to-build-custom-android-app-for-your-arduino-project-using-mit-app-inventor/>, (pregledano 09. ožujka 2019.)
5. Instructables: Modify the HC-05 Bluetooth Module Defaults Using AT Commands, <http://www.instructables.com/id/Modify-The-HC-05-Bluetooth-Module-Defaults-Using-A/>, (pregledano 09. ožujka 2019.)
6. Instructables: ArduDroid: a Simple 2-Way Bluetooth-based Android Controller for Arduino, <http://www.instructables.com/id/Andruino-A-Simple-2-Way-Bluetooth-based-Android-C/>, (pregledano 09. ožujka 2019.)