



Fakultet elektrotehnike i računarstva, Sveučilišta u Zagrebu
Zavod za elektroničke sustave i obradu informacija
Sveučilište u Zagrebu

ToDo-

pametni podsjetnik



- △ Svima zainteresiranima
- △ Osnove programiranja u Pythonu
- △ Arduino sustav i povezivanje različitih modula

Sažetak

Današnje vrijeme traži veliku angažiranost pojedinca u bilo kakvom djelovanju. Zadaci koje danas svatko mora izvršavati su daleko brojniji nego kroz povijest.

Stoga, donosimo sustav za lakše praćenje svih dnevnih zadataka. Glasovni unos zadataka u kalendare ubrzat će svakodnevno djelovanje pojedinca, uz to mu omogućiti rasterećenje i posvećivanje bitnim zadacima.

Temeljna ideja je prepoznavanje glasa i definiranje osobnih želja i njihovo zapisivanje u različite aplikacije. U ovoj dokumentaciji prikazane su osnovne smjernice za izradu sustava za govorni unos podataka kalendar.

Sadržaj

1. UVOD.....	3
2. SUSTAV ZA UPIS PODSJETNIKA.....	4
2.1. Arduino.....	4
2.1.1. Serial.....	4
2.2. Serial Port Bluetooth Module: HC-05.....	5
2.2.1. BT Voice Control for Arduino	5
2.3. Python.....	6
2.3.1. PySerial.....	6
2.3.2. Google Calendar API	6
3. NAČELO RADA SUSTAVA I REALIZACIJA	8
4. ZAKLJUČAK.....	12
5. LITERATURA.....	13
6. POJMOVNIK	14

Ovaj seminarski rad je izrađen u okviru predmeta „Sustavi za praćenje i vođenje procesa“ na Zavodu za elektroničke sustave i obradbu informacija, Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu.

Sadržaj ovog rada može se slobodno koristiti, umnožavati i distribuirati djelomično ili u cijelosti, uz uvjet da je uvijek naveden izvor dokumenta i autor, te da se time ne ostvaruje materijalna korist, a rezultirajuće djelo daje na korištenje pod istim ili sličnim ovakvim uvjetima.

1. Uvod

Pametne kuće nam uvelike pomažu u vođenju što jednostavnijeg i organiziranijeg života u današnjem ubrzanom svijetu. Kako se tehnologija razvijala, rasla je i želja čovjeka za što većom udobnosti života.

Međutim, razvojem mnogih tehnologija i novih zadataka javlja se potreba pojedinca za multitaskingom, odnosno odrađivanjem više zadataka istovremeno. Ljudima koji su zaposleni ili uključeni u bilo kakvu djelatnost postalo je prestresno voditi i naizgled „jednostavan“ život. Život postaje prestresan, stoga se u ovom radu posvećujemo jednom načinu lakše organizacije misli, ideja i zadataka.

Kako bi se pojedinac mogao organizirati te uspješno odrađivati sve svoje zadatke, potrebno je pratiti svoje zadatke te ih na neki način organizirati. S obzirom na to da je bitno da se na jednostavne zadatke troši što manje vremena, razvijamo sustav što jednostavnijeg unošenja zadataka u vlastite kalendare. Glasovnim unosom ideja, zadataka i misli vrlo brzo se mogu zabilježiti zadaci kojih je svaki dan sve više.

Činjenica da neke informacije možemo izgovoriti i zapisati u danom trenutku i više ne misliti na njih, omogućuje nam primanje novih i lakše koncentriranje na možda bitnije i teže zadatke.

2. Sustav za upis podsjetnika

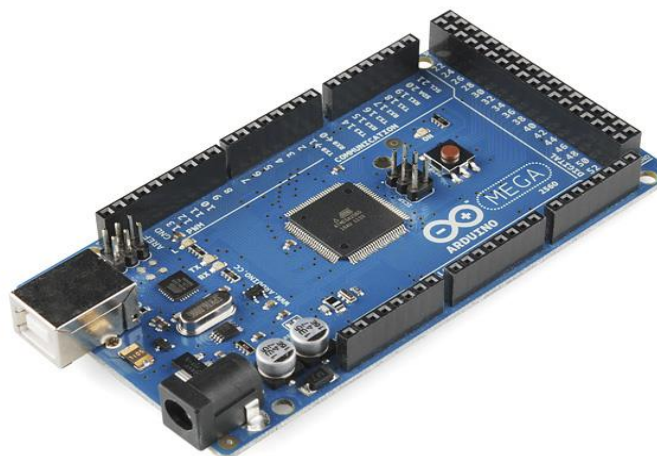
Sustav za upis podsjetnika može dio sustava za upravljanje pametnom kućom, ako se naredbe u kući mogu dati glasovno moguće je i izvesti sustav koji će nam osim upravljanjem različitim funkcija u kući omogućiti i unos zadataka u google kalendar.

Izvedba jednostavnog sustava za unos govora je dana u daljnjem tekstu. Sustav je izveden koristeći android, bluetooth modul, Arduino i računalo.

2.1. Arduino

Arduino je „open source“ platforma temeljena na fleksibilnom i jednostavnom korištenju softvera i hardvera. Arduino je tiskana pločica na kojoj se nalazi mikrokontroler i ostalo sklopovlje za rad mikrokontrolera. Arduino kodovi i različite korisnosti dostupne su na internetu te omogućuju njegovu čestu primjenu.

U ovom radu koristio se Arduino Mega 2560, baziran na Atmega2560. Ima 54 digitalna ulazno/izlazna pina; koji mogu biti korišteni kao PWM izlazi, analogni ulazi, hardverski serijski portovi itd. Sve mogućnosti ovog mikrokontrolera mogu se pronaći u njegovom datasheetu dostupnom na internetu **Error! Reference source not found..**



Slika 1 Arduino Mega 2560

2.1.1. Serial

Serial je skup funkcija koje omogućavaju komunikaciju između Arduino pločice i računala ili drugih uređaja. Sve Arduino pločice imaju barem jedan serijski port (UART, USART). Komunicira se digitalnim pinovima 0 (RX) i 1 (TX) te preko USB-a s računalom.

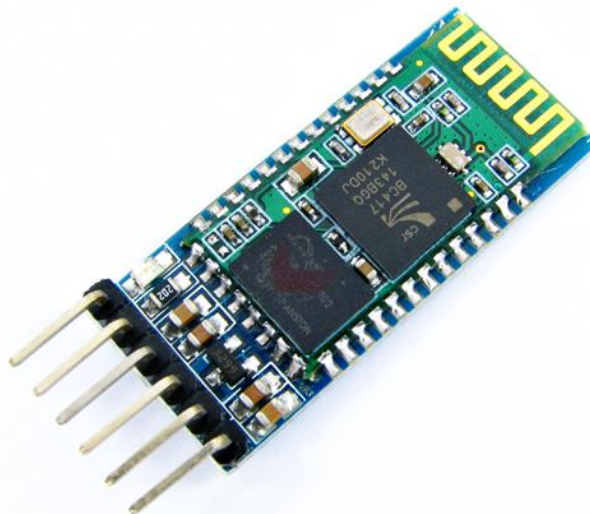
Arduino okruženje ima također opciju ugrađenog serijskog monitora na kojem je moguće pratiti što se točno događa i prenosi upravo preko serijskog porta.

2.2. Serial Port Bluetooth Module: HC-05

HC-05 modul je Bluetooth SPP (Serial Port Protocol) modul, dizajniran za jednostavno bežično povezivanje [1].

Serijski port bluetooth modula je opremljen Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation s kompletnim 2.4 GHz radio primopredajnikom i pojasom.

U sustavu se HC-05 žičano spaja na Arduino, a bluetooth komunikacija se ostvaruje s android mobilnim telefonom.

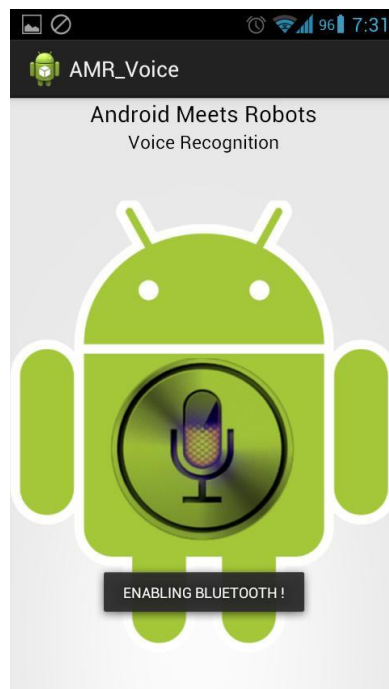


Slika 2 Serial Port Bluetooth Module: HC-05

2.2.1. BT Voice Control for Arduino

BT Voice Control je aplikacija koja koristi glasovno prepoznavanje unutar mobitela te prenosi naredbe do sustava (inicijalno je aplikacija napravljena za upravljanje robotima, ali je primjenjiva i u drugim aplikacijama). Aplikacija se uparuje s bluetooth serijskim modulom i šalje glas, odnosno izrečenu riječ ili rečenicu kao string.

Može se koristiti s bilo kojim mikrokontrolerom koji može obrađivati string formate.



Slika 3 Početni zaslon aplikacije BT Voice Control

2.3. Python

Python je programski jezik opće namjene, interpretiran i visoke razine. Python dopušta programerima korištenje nekoliko stilova programiranja. Objektno orijentirano, strukturalno i aspektno orijentirano programiranje su stilovi dopušteni korištenjem Pythona te ova fleksibilnost čini Python programski jezik sve popularnijim. Python se najviše koristi na Linuxu, no postoje i inačice za druge operacijske sustave.

U našem sustavu Python je korišten na Windows operacijskom sustavu.

2.3.1. PySerial

Da bi se uopće Python mogao primijeniti u sustavu koji koristi serijski ulaz koristi se pySerial software koji ima niz funkcija za otvaranje i čitanje sa serijskog porta.

Korištenjem pySeriala naš sustav otvara serijski port i čita njegov ulaz. S obzirom na to da se s Arduina zapravo šalje string kreiran u BT Voice Controlleru taj string najlakše je učitati primjenom funkcija iz ovog sustava.

2.3.2. Google Calendar API

Google Calendar API nam omogućuje prikaz, stvaranje i izmjene događaja u kalendaru kao i rad s različitim povezanim objektima. Rad i korištenje API-ja je objašnjeno na stranicama Google Developers [3]. Korištenje API-ja moguće je u različitim programskim jezicima kao što su Java, PHP, .NET, JavaScript, Ruby, Python, Go, Android, iOS.

Međutim s obzirom na to da je Python jednostavan i lako primjenjiv on je korišten za rad na sustavu.

Za korištenje Google Calendar API-ja potrebno je imati pristup gmail-u, odnosno „google account“.

3. Načelo rada sustava i realizacija

Sustav je sastavljen iz navedenih komponenti se koristi tako da se korisnik poveže preko bluetootha s Arduino mikrokontrolerom.

To je napravljeno aplikacijom BT Voice Control koja se spaja preko bluetootha na modul HC-05. Aplikacija govor pretvara u „string“ tj. niz znakova, kojeg šalje na Arduino. Kod za Arduino dan je u nastavku:

```
SoftwareSerial BT(10, 11); //TX, RX respectively
String voice;

void setup() {
  BT.begin(9600);
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  Serial.println("Goodnight moon!");
}

//-----//
void loop() {
  while (BT.available()){ //Check if there is an available byte to read
    delay(10); //Delay added to make thing stable
    char c = BT.read(); //Conduct a serial read
    if (c == '#') {break;} //Exit the loop when the # is detected after the
word
    voice += c; //Shorthand for voice = voice + c
  }
  if (voice.length() > 0) {
    Serial.println(voice);

    if(voice == "*start")
    {
      digitalWrite(13, HIGH);
    }

    else if(voice == "*stop")
    {
      digitalWrite(13, LOW);
    }

    voice="";} //Reset the variable after initiating
```

Programiranjem Arduina odlučujemo što ćemo s danim nizom. Analiza se može izvesti već na mikrokontroleru ili se može proslijediti preko serijskog porta.

Proslijeđeni niz analiziramo python skriptom i izrađenim funkcijama koje se mogu naći u dostavljenim kodovima:

- funk_analize.py
- funk_mjeseci.py
- vrijeme.py

Analiza se sastoji od traženja datuma, mjeseca i vremena izrade događaja u kalendaru. Zadani format rečenice je:

„Podsjetnik <dan> <naziv mjeseca> u <sati> ... “

Za dodatne provjere potrebno je unaprijediti izrađene funkcije.

Nakon što primimo niz skripta ga obradi i odredi je li uopće moguće izraditi podsjetnik na temelju primljenih riječi. Ako nije dobijemo sustav nas informira o nedostacima, a ako da onda sustav izrađuje događaj te nas o tome obavijesti i uz pošalje link.

Različiti parametri se mogu mijenjati u kodu, ali samo se dan, mjesec i vrijeme mogu definirati glasom. Kao zadana postavka definirano je postavljanje podsjetnika e-mailom 24 h prije zadatka. Osim toga u opis će se spremiti cijela izrečena rečenica kao poblža informacija o zadatku.

Kod glavne funkcije koja čita, analizira i kreira događaj korištenjem google calendar API-a.

```
# -*- coding: utf-8 -*-

from datetime import datetime
import os

from apiclient.discovery import build
from httplib2 import Http
import oauth2client
from oauth2client import client
from oauth2client import tools

import serial
import unicodedata

import funk_analize
import funk_mjeseci
import vrijeme
import ucitaj_serial

try:
    import argparse
    flags = argparse.ArgumentParser(parents=[tools.argparser]).parse_args()
except ImportError:
    flags = None

SCOPES = 'https://www.googleapis.com/auth/calendar.readonly'
CLIENT_SECRET_FILE = 'client_secret.json'
APPLICATION_NAME = 'Google Calendar API Quickstart'

def get_credentials():
    """Gets valid user credentials from storage.

    If nothing has been stored, or if the stored credentials are invalid,
    the OAuth2 flow is completed to obtain the new credentials.

    Returns:
        Credentials, the obtained credential.
    """
    home_dir = os.path.expanduser('~')
    credential_dir = os.path.join(home_dir, '.credentials')
    if not os.path.exists(credential_dir):
        os.makedirs(credential_dir)
```

```

credential_path = os.path.join(credential_dir,
                                'calendar-quickstart.json')

store = oauth2client.file.Storage(credential_path)
credentials = store.get()
if not credentials or credentials.invalid:
    flow = client.flow_from_clientsecrets(CLIENT_SECRET_FILE, SCOPES)
    flow.user_agent = APPLICATION_NAME
    if flags:
        credentials = tools.run_flow(flow, store, flags)
    else: # Needed only for compatibility with Python 2.6
        credentials = tools.run(flow, store)
    print 'Storing credentials to ' + credential_path
return credentials

def initialize():
    credentials = get_credentials()
    service = build('calendar', 'v3', http=credentials.authorize(Http()))

    return credentials, service

credentials, service = initialize()

def example_event(linija, pocetak, kraj):
    event = {
        'summary': 'Podsjetnik',
        'description': linija,
        'start': {
            'dateTime': pocetak,
            'timeZone': 'Europe/Zagreb',
        },
        'end': {
            'dateTime': kraj,
            'timeZone': 'Europe/Zagreb',
        },
        'reminders': {
            'useDefault': False,
            'overrides': [
                {'method': 'email', 'minutes': 24 * 60},
            ],
        },
    }

    return event

def event_put(event, calendar_name, service):
    return service.events().insert(calendarId=calendar_name,
body=event).execute()

def upcoming_events(event_count):
    now = datetime.utcnow().isoformat() + 'Z' # 'Z' indicates UTC time
    print 'Getting the upcoming event_cnt events'
    eventsResult = service.events().list(
        calendarId='primary', timeMin=now, maxResults=event_count,
singleEvents=True,
        orderBy='startTime').execute()
    events = eventsResult.get('items', [])

    if not events:
        print 'No upcoming events found.'
    for event in events:
        start = event['start'].get('dateTime', event['start'].get('date'))

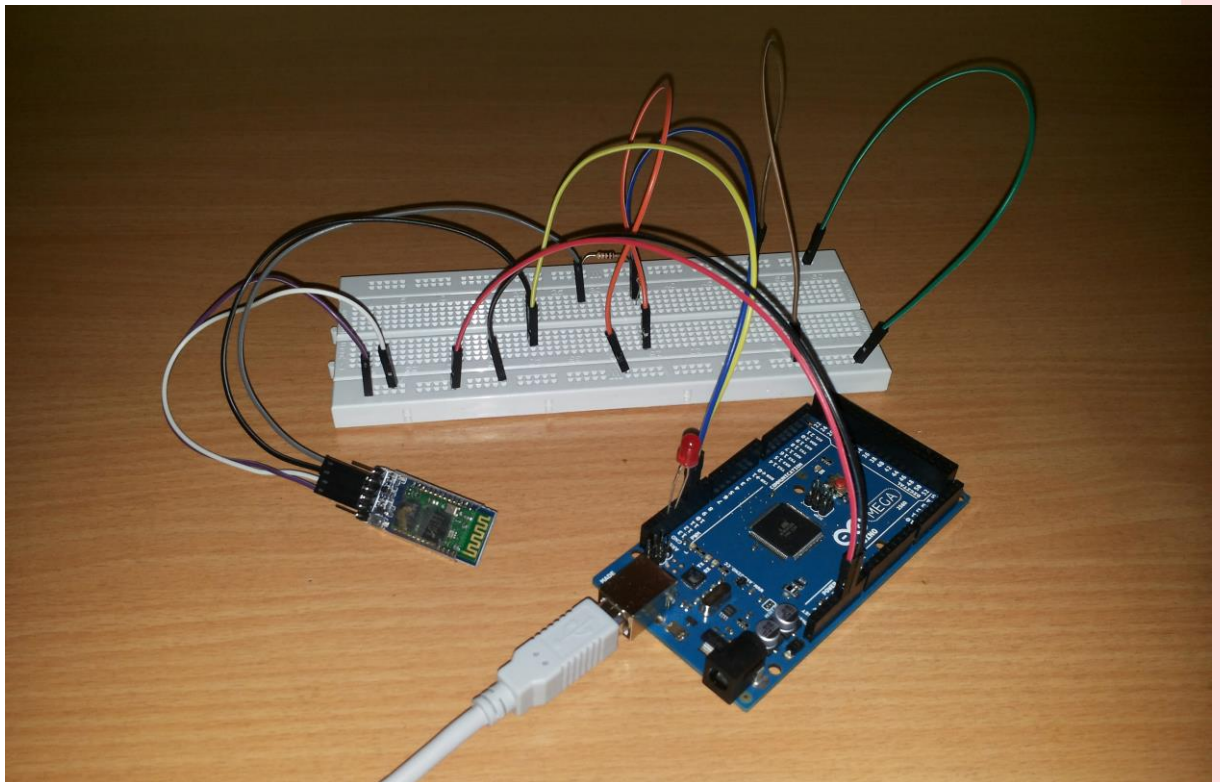
```

```
print start, event['summary'].encode('utf-8')

def main():
    arduino = serial.Serial( 9, 9600, timeout=.1)

    while True:
        data = arduino.readline()#[::-2] #the last bit gets rid of the new-line
        chars
        if data:
            data = unicode(data, 'utf-8')
            data = unicodedata.normalize('NFKD', data).encode('ascii','ignore')
            print data
            dan, mjesec, sati = funkc_analize.analiza(data)
            if dan != '0' and mjesec != '0':
                pocetak, kraj = vrijeme.vrijeme(dan, mjesec, sati)
                barbara_event = example_event(data, pocetak, kraj)
                event = event_put(barbara_event, 'primary', service)
                print 'Event created: %s' % (event.get('htmlLink'))
            else:
                print 'Podsjetnik se ne moze izdraditi!'
        arduino.close()

if __name__ == '__main__':
    main()
```



Slika 4 Realizacija sustava

4. Zaključak

Sustavom za upis podsjetnika omogućena je lagana kontrola vlastitih zadataka. Međutim iako naizgled jednostavan sustav, ukoliko se želi unaprijediti potrebno je mnogo rada na analizama rečenica.

Sustav sa zadanim formatom naredbe je lako izvediv jer se zna što očekivati, ako pak rečenice nemaju definiran format analiza se znatno zakomplicira. Ipak, valjalo bi se držati forme jer tako sustav možemo nadograđivati, osigurati mu budućnosti i daljnji razvoj.

Pa bi se tako za odgovarajuće forme moglo upisivati sastanke, rođendane, seminare i sl. u kalendar, definirajući alarm i podsjetnik na upisane „evente“.

S obzirom na to da sustav prepoznaje govor to valja iskoristiti i za upravljanje drugim različitim funkcijama pametne kuće. Na taj način održavanje pametne kuće će se pojednostaviti te se funkcije kao npr. paljenje i gašenje svijetla neće morati izvoditi ručno, već samo jednostavnom glasovnom naredbom.

5. Literatura

- [1] Arduino. URL: <http://www.arduino.cc/en/Main/ArduinoBoardMega2560>
- [2] DATASHEET BLUETOOTH TO SERIAL PORT MODULE HC05. URL: <http://www.electronica60norte.com/mwfls/pdf/newBluetooth.pdf>
- [3] Google Developers. URL: <https://developers.google.com/google-apps/calendar/>

6. Pojmovnik

Pojam	Kratko objašnjenje	Više informacija potražite na
API	Programsko sučelje	http://hr.wikipedia.org/wiki/API
Arduino	Razvojna platforma	http://www.arduino.cc/
HC-05	Bluetooth modul (SPP)	http://www.electronica60norte.com/mwfls/pdf/newBluetooth.pdf
Python	Programski jezik	https://www.python.org/