

Hemstudieuppgift v108

Inledning

I denna hemstudieuppgift kommer du att titta närmare på en del av SSR-systemet, nämligen ADS-B.

Börja med att titta på följande klipp:

https://youtu.be/M_C2R4U-xug

Och därefter:

<https://youtu.be/wj89g8Ml3aA>

Den materiel du behöver för att genomföra uppgiften är: Raspberry pi och SDR-mottagare med tillhörande antenn (samt ev. kopplingsplatta, motstånd och lysdioder). Till detta också tangentbord, mus, skärm och strömförsörjning samt tillgång till internet.

Om du fastnar på något eller har några frågor, så skicka då ett mail till:

tss.lv6@gmail.com

Förberedelser

Här kan du, om du vill, nyttja den senaste varianten av Raspberry pi operativsystem genom att bränna om SD-kortet.

Tänk på att alla egna filer du har på SD-kortet försvinner om du gör detta!

Ladda ner zip-filen som heter "Raspberry Pi OS with desktop and recommended software" från Raspberry pi foundation till din egen dator:

<https://www.raspberrypi.org/software/operating-systems/#raspberry-pi-os-32-bit>

Packa upp denna och bränn den sedan till SD-kortet med tex programmet Win32DiskImager:

<https://sourceforge.net/projects/win32diskimager/>

Anslut tangentbord, mus, skärm och SDR-mottagaren till din Raspberry pi. Glöm inte att ansluta antennen till SDR-mottagaren. Placera antennen på en plats så att mottagningen kan bli bra. Anslut din Raspberry pi till internet via wifi eller nätverkskabel. För den som har erfarenhet av VNC går det också att fjärrstyra Raspberry pi från egen dator.

Öppna ett kommandofönster och uppdatera operativsystemet med följande kommandon:

```
sudo apt update  
sudo apt upgrade
```

Installation av dump 1090

Installera därefter programvaran dump 1090 med följande kommandon:

```
sudo apt install dump1090-mutability
```

OBS att det skall vara sudo apt install och inte sudo apt-get install.

Svara ja på frågan om dump1090 skall startas automatiskt.

Kör sedan följande kommandon (det första ger dump1090 möjlighet att komma åt SDR-mottagaren):

```
sudo usermod -a -G plugdev dump1090
sudo systemctl restart dump1090-mutability
```

Starta nu om din Raspberry pi med kommandot:

```
sudo reboot
```

Titta på vilka specifika kommandon/inställningar som programmet dump1090 accepterar genom att öppna hjälpavsnittet med följande kommando:

```
dump1090-mutability --help
```

Kontrollera nu att ADS-B data kommer in till din Raspberry pi.

Tips: För att kontrollera så att det faktiskt är några flygplan i närheten så kan du öppna Chromium och kolla på FR24:

www.flightradar24.com

Se på mottagen ADS-B data i kommandofönstret genom att köra följande kommandon:

```
sudo systemctl stop dump1090-mutability
dump1090-mutability --interactive-rtl1090
```

Stoppa med CTRL-C och kör sedan:

```
sudo systemctl restart dump1090-mutability
```

Se på ADS-B data på en kartbild genom att öppna Chromium och skriva in följande i adressfältet:

```
127.0.0.1/dump1090/gmap.html
```

Flytta kartan genom att dra och klicka. Zooma till önskad kartbild.

Se på data i textform i Chromium genom att skriva in följande i adressfältet:

```
127.0.0.1/dump1090/data/aircraft.json
```

Det är denna data som Python-programmet kommer att nyttja.

Vad betyder JSON?

Kontrollera vilka versioner av Python som finns i din Raspberry pi genom att öppna ett kommandofönster och skriva in följande (obs två minustecken framför version):

```
python --version
python3 --version
```

Nyttja fortsättningsvis Python 3 (tex 3.7.3).

I Python kan vi använda oss av en mängd färdiga funktioner genom att importera de bibliotek som dessa återfinns i. Många av dessa bibliotek finns grundinstallerade (tex time och math) medan några måste laddas ner. Python-kommandot `time.sleep(5)` innebär exempelvis att vi vill nyttja sleep-funktionen som återfinns i time-biblioteket. I detta fall anges tiden i sekunder.

Ett bibliotek du behöver nu är PIL (Python Image Library). Installera detta med följande kommando:

```
pip3 install pillow
```

Hämta ADS-B programmet

Öppna Chromium och gå in på:

<https://github.com/TSS-Lv6/ADS-B>

Ladda ner följande program:

ADS-B_mottagare_med_karta.py

ADS-B_mottagare_extra_1.py

ADS-B_mottagare_extra_2.py

Bakgrundskarta_Sverige.png

Alternativt kan du öppna ett kommandofönster och skriva in följande:

```
wget https://github.com/TSS-Lv6/ADS-B/archive/main.zip
```

Flytta det du laddat ner till en plats där du lätt kommer åt det, tex skrivbordet (/home/pi/Desktop).

Öppna en programmeringsmiljö (IDE). Förslagsvis kan du använda Thonny, som du kan hitta under applikationsmenyn -> programmering. Här kan du välja "switch to regular mode" och starta om Thonny. Kontrollera så att programmet nyttjar någon variant av Python 3 (tex 3.7.3). Du ser detta i den nedre "Shell"-rutan.

Öppna **ADS-B_mottagare_med_karta.py** och kör detta aningen med grön knapp eller F5-tangenten. Stoppa med röd knapp, ESC-tangenten eller genom att stänga fönstret som programmet skapade.

Prova också att starta programmet från kommandofönstret genom att köra följande kommando:

```
python3 ADS-B_mottagare_med_karta.py
```

OBS att du först måste flytta dig till den mapp där filen ligger. Se vad som finns i aktuell mapp med kommandot:

```
ls
```

Flytta dig sedan med:

```
cd
```

följt av mappens namn. Här finns också en bra autokompletteringsfunktion. Då du skrivit någon/några bokstäver kan du trycka på TAB-tangenten.

Här hittar du övrigt en bra referens för linux-kommandon:

<https://www.raspberrypi.org/documentation/linux/usage/commands.md>

Du kan också göra programmet exekverbart med följande kommando:

```
chmod +x ADS-B_mottagare_med_karta.py
```

Och sedan starta det med:

```
./ADS-B_mottagare_med_karta.py
```

Att vi kan göra så förutsätter dock att det i inledningen av vår Python-fil finns med raden `#!/usr/bin/python3`. Detta så att datorn vet vilket program det skall starta vår fil med. Vad kallas `#!` ?

ADS-B programmet

Sätt dig nu in i hur Python-programmet fungerar genom att titta igenom kod och kommentarer.

Du kan tex avkommentera rad 97 (ta bort # framför print), för att se hela det mottagna meddelandet i "Shell"-rutan.

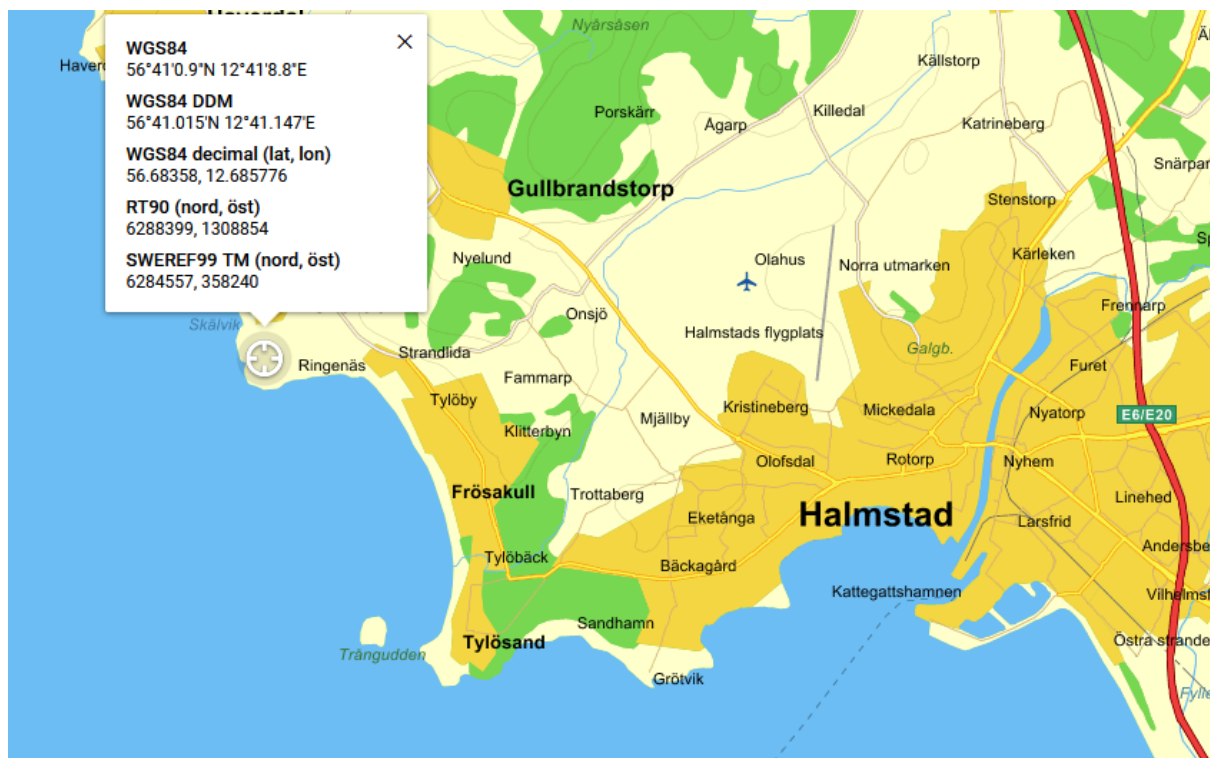
```
# Detta är en kommentar i Python

'''
Flera rader kan också kommenteras bort på en gång
OBS att det är tre st '''
Och inte tre st ...
'''
```

Som du ser finns det i inledningen av programmet ett parti där du skall lägga in din aktuella position så att målens relativa position i förhållande till var du befinner dig blir rätt.

Nyttja tex Eniro kartor för att ta ut positionen i WGS84 decimalt (lat, long).

<https://kartor.eniro.se/>



Efter inledningen där bla kartbilden laddas och passas in så har programmet en loop som går med 60Hz. Det finns sedan delar i loopen som bara utförs varje hel sekund.

Vilka är flygplanen i listan som automatiskt märks om till fientliga i programmet?

Prova att skriva till något av de just nu mottagna flygplanen så att dessa märks som fientliga.

Egna ändringar

Det är nu dags för dig att göra någon eller några egna ändringar i programmet.

Välj ur förslagen nedan eller kom själv på någon/några förändringar du vill göra.

- Koppla till en lysdiod och gör så att denna blinkar då det finns flygplan närmare än x km
- Ändra skalan på ppi-ytan beroende på hur långt bort du kan maximalt kan detektera flygplan från den plats där just du befinner dig
- Gör så att presentationen från programmet fyller hela datorskärmen
 - o Läs in skärmstorleken med följande Python kommandon:
width, height = pygame.display.Info().current_w, pygame.display.Info().current_h
OBS att detta måste föregås av import pygame och pygame.init()
- Lägg till/ta bort information i textfältet
- Gör olika målsymboler beroende på flygplanskategori eller ändra målsymbolerna så att dessa följer mil-standard 2525
- Lägg till en höjdvyn på motsvarande sätt som i LT-91
 - o Gör så att höjdvyn kan väljas att vara antingen linjär eller logaritmisk
- Gör så att det går att klicka på mål med musen för att få extra info i ppi-bilden tex höjd
- Gör så att flygplatser som finns inom ppi-bilden markeras på något sätt i kartbilden
- Hämta data från FR24 istället för från ADS-B mottagaren
 - o Se exempelkod och noteringar i filen **ADS-B_mottagare_extra_1.py**
- Nyttja data från både ADS-B mottagaren och från FR24.
 - o Markera mål som mottagits från båda källorna.
 - o Här kan du också särskilt markera mål som enbart tas emot av din ADS-B mottagare. FR24 måste nämligen ta bort vissa flygplan/helikoptrar från den information som de lägger ut på nätet. Exempel på detta är polisens helikoptrar.
 - o Här kan du också råka ut för det dilemma som även vi drabbas av, nämligen att positionsinformationen skiljer sig lite åt ifrån de olika källorna, bland annat på grund av tidsfördröjningar. Kan du hitta en bra lösning på detta?
- Gör så att alla flygplan som är registrerade i Sverige märks som egna. Dessa har alla en Hex-kod i spannet 4A8000-4AFFFF.
 - o http://www.aerotransport.org/html/ICAO_hex_decode.html
- Gör så att programmet läser orienteringar om målen (espeak).
 - o Se exempelkod och noteringar i filen **ADS-B_mottagare_extra_2.py**
- Lägg till Python-programmet i autostarten, så att detta startar automatiskt då din Raspberry pi kraftsätts.