



Műholdas helyzetmeghatározó rendszerek I.

Intelligens Rendszerek
Gyakorlat



GPS

NMEA log és paramétere



Feladat:

- Adott egy NMEA gps log
- Számítsuk ki a jármű minimális, maximális illetve átlag sebességét
- Rajzoljuk ki egy 2D diagramra a sebességet



GPRMC NMEA mondat felépítése

- % \$GPRMC Recommended Minimum sentence
- % 123519 Fix taken at 12:35:19 UTC
- % A Status A=active or V=Void.
- % 4807.038 Latitude 48 deg 07.038'
- % N North
- % 01131.000 Longitude 11 deg 31.000'
- % E East
- **% 022.4 Speed over the ground in knots**
- % 084.4 Track angle in degrees True
- % 230394 Date - 23rd of March 1994
- % 003.1,W Magnetic Variation
- % *6A The checksum data, always begins with *



Checksum számítása

\$GPGGA,542.600,3314.8961,N,11142.2714,W,1,10,.86,395.2,M,-26.6,M,,*52

- Checksum: 0x52 (maximális értéke 0xFF)

Ciklus (i=2-től mondathossza-3-ig)

{

chk=bitenkénti_xor(chk, mondat(i));

}

Ciklus vége

//vesszük a karakterek decimális kódjainak a xor kapcsolatát!

G xor **P** xor **G** xor **G** xor **A** xor , xor **5** xor **4** xor **2** xor . xor **6** xor **0** xor **0** xor ,
xor xor **M** xor , xor ,



Matlab parancsok

- help parancsnev – gugli mellett az egyik legjobb barátunk
- clc – képernyő törlése
- clear all – összes változó törlése
- cell_tipusu_tomb = importdata('filename.txt');
- string = char(cell_tipusu_tomb(i)); //cell típusból string
- strcmp(a,b) //két string összehasonlítása, visszatérési érték 0 vagy 1
- dec_tipusu_valtozo = hex2dec(hex_valtozo);
- dec_valtozo = unicode2native(character)//karakter-ből szám
- bitszintu_xor_eredmeny = bitxor(a,b); //a és b változó bitszintű xor kapcsolata
- double_valtozo = str2double(string_valtozo); //stringből double típusú változó
- Plot(x), Plot(x,y), Plot(x,y,z) //grafikon megjelenítés



```
format long
```

```
clc
```

```
clear all
```

```
%GPS log betöltése
```

```
filename='gps.txt'; %filenév
```

```
speedOverGround=0;
```

```
gps = importdata(filename); %betöltés után cell típusú tömböt  
kapunk!
```

```
                                %bármilyen típusú adatot  
tartalmazhat. Számoláskor kasztolni kell!
```

```
delimiter=', '; %elválasztó
```

```
s=0; %segéd változó
```

```
idx=1; %altitude tömb indexe
```

```
chk=0; %Cheksum
```

```
nmea_type='$GPRMC'; %keresett NMEA mondat típusa
```

```
calculated_chk=0; %számolt checksum
```



```
for i=1:length(gps)
    s=char(gps(i));
    splitted=strsplit(s,delimiter);%egy GPS mondat feldarabolása az
    elválasztó karakter mentén
    if strcmp(splitted(1),nmea_type)==1 %keresett mondat ellenőrzése
        chk=char(splitted(length(splitted)));%Checksum kinyerése a mondat
        végéről
        chk=chk(length(chk)-1:length(chk)); %chk max FF, a * levágása
        chk=hex2dec(chk); %hexa szám a checksum, így át kell
        konvertálni decimálisba

        %Checksum ellenőrzése, a checksum előállítása a $ után
        karaktertől a
        %a * előtti karakterig tart, XOR kapcsolat (bitszintű)
        calculated_chk=0;
        for j=2:length(s)-3
            num = unicode2native(s(j)); %a karakterből számot számot
            kell konvertálni
            calculated_chk=bitxor(calculated_chk,num); %XOR művelet
        end
    ...
end
```




```
%az eredeti és a számolt checksum ellenőrzése
```

```
    if(chk==calculated_chk)
```

```
speedOverGround(idx)=1.852*str2double(split(8));
```

```
%csomó->kmh
```

```
        idx=idx+1;
```

```
    end
```

```
end
```

```
end
```

```
disp(sprintf('minimum speed: %0.5g km/h', ...  
            min(speedOverGround)));
```

```
disp(sprintf('maximum speed: %0.5g km/h', ...  
            max(speedOverGround)));
```

```
disp(sprintf('average speed: %0.5g km/h', ...  
            mean(speedOverGround)));
```

```
plot(speedOverGround);
```



Sebesség diagram

