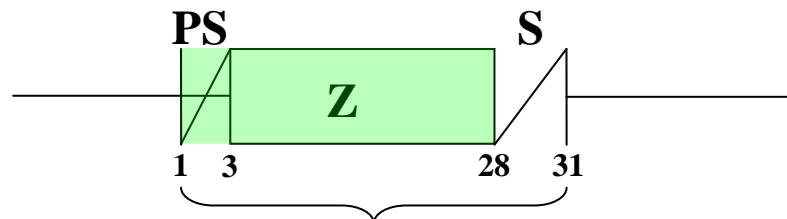


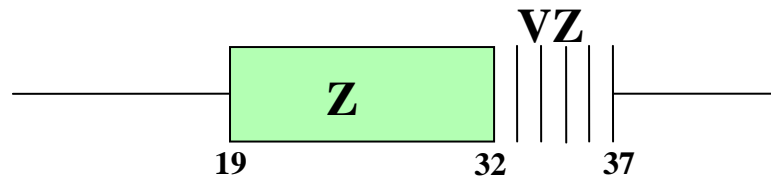
## Házi feladat – ACTROS JAVA programozás

- I. Adja meg az eredeti forgalomtechnikai kódban megtalálható közbenső idő mátrixot!
- II. Rajzolja fel az eredeti forgalomtechnikai kódban megtalálható 1. fix program fázisidő-tervét (FixProg1.java)!
  - Közbenső idő: két egymást követő zöld jelzés közötti idő S és PS nélkül
  - A Jármű jelzőcsoportnál a szabad időt adjuk meg, azaz a  $t^{PS} + t^Z$  időt, pl.: *K.BEKI(Var.j11, 1, 28)*;



Összesen 27 mp, ami tartalmazza a 2 mp PS időt is!

- A Gyalogos jelzőcsoportnál csak a zöldidőt  $t^Z$  adjuk meg, ami után még 5 mp villogó zöld jelzés következik, pl.: *K.BEKI(Var.gy51, 19, 32)*;



- III. Készítsen egy saját fixprogramot FixProgHF.java néven! A program fázissorrendje a FixProg1 program fordítottja legyen: meg kell változtatni a fázissorrendet, éppen az eredetivel ellenkezőre (az első fázis lesz az utolsó, a második az utolsó előtti, és így tovább).

- A ciklusidő legyen  $T_C = 79 [mp]$ ! A zöldidőket ( $t_i^Z = ?$ ) a mértékadó órai forgalmak alapján kell meghatározni (egészre kerekítve!), az  $N_i = \frac{t_i^Z 3600 \lambda_i}{2T_C}$  összefüggés alapján, ahol  $\lambda_i$  az adott irányra vonatkozó sávszám.

A mértékadó forgalmak:

- j11/j21 irány:  $N_{j11} = N_{j21} = 500 [jm/óra]$
- j31 irány:  $N_{j31} = 300 [jm/óra]$
- j41 irány:  $N_{j41} = 550 [jm/óra]$
- A gyalogos jelzők egységesen 14 mp-es zöldidővel működjenek!

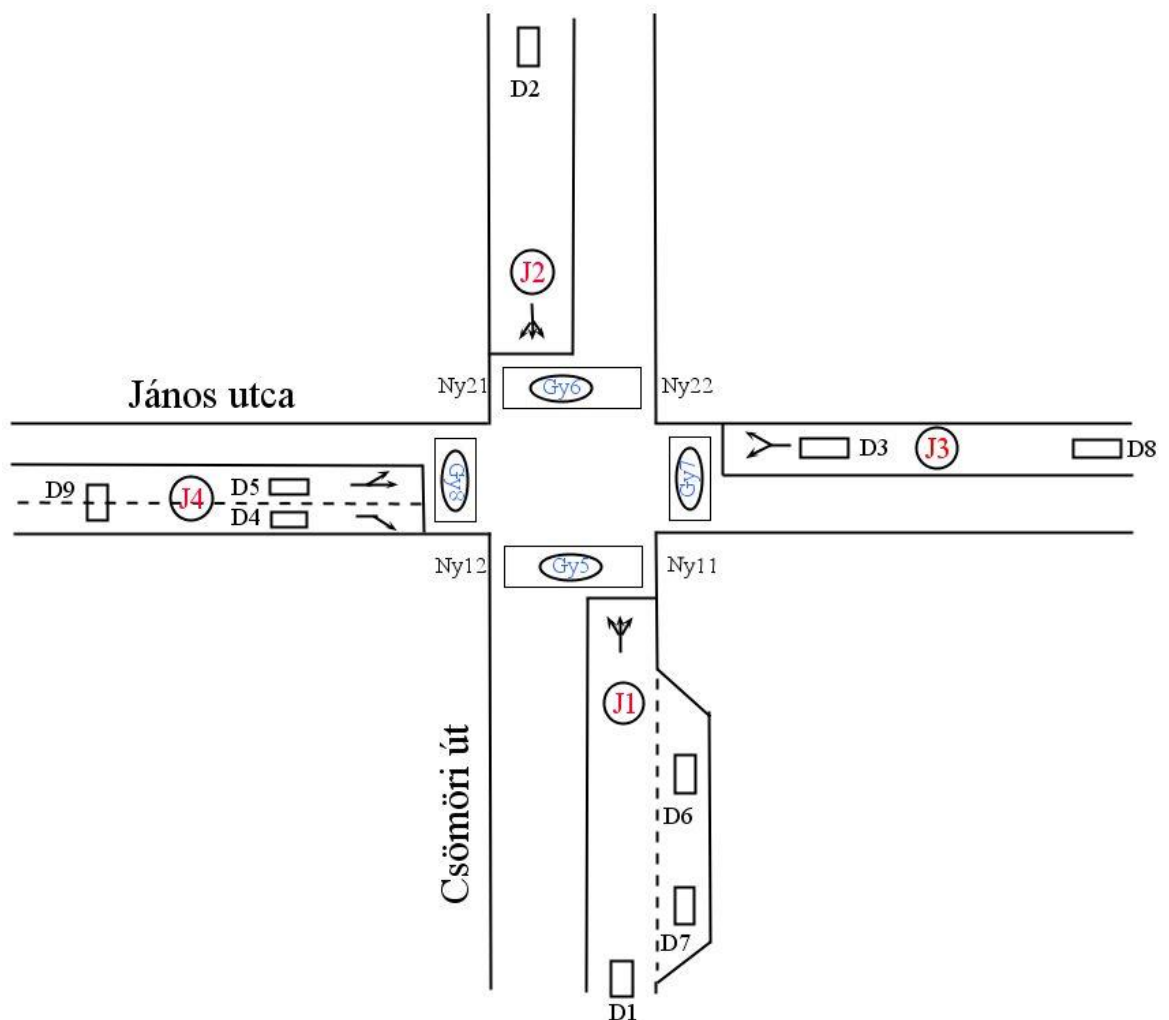
- Úgy állítsa be a programváltást, hogy a FixProgHF működése a HF bemutatás időtartamára essen! (lásd: Init.java → **protected void initialisiereUhr()** {...} eljárás)
- A fázisok felcserélésénél ügyelni kell a közbensőidőkre !
- A tervezésnél a mértékadón túl megmaradó zöldidő-többletet arányosan elosztva adjuk ki! Számítsuk ki az tervezett programunk által elérhető maximális kapacitást irányonként ( $N_i^{MAX} = ?$ )!
- Ábrázolja az elkészített fixprogram fázisidő-tervét!

IV. A FixProgHF felhasználásával készítsen egy forgalomfüggő programot ForgfHF.java néven! A program legalább egyetlenegy forgalomfüggő elemet tartalmazzon!

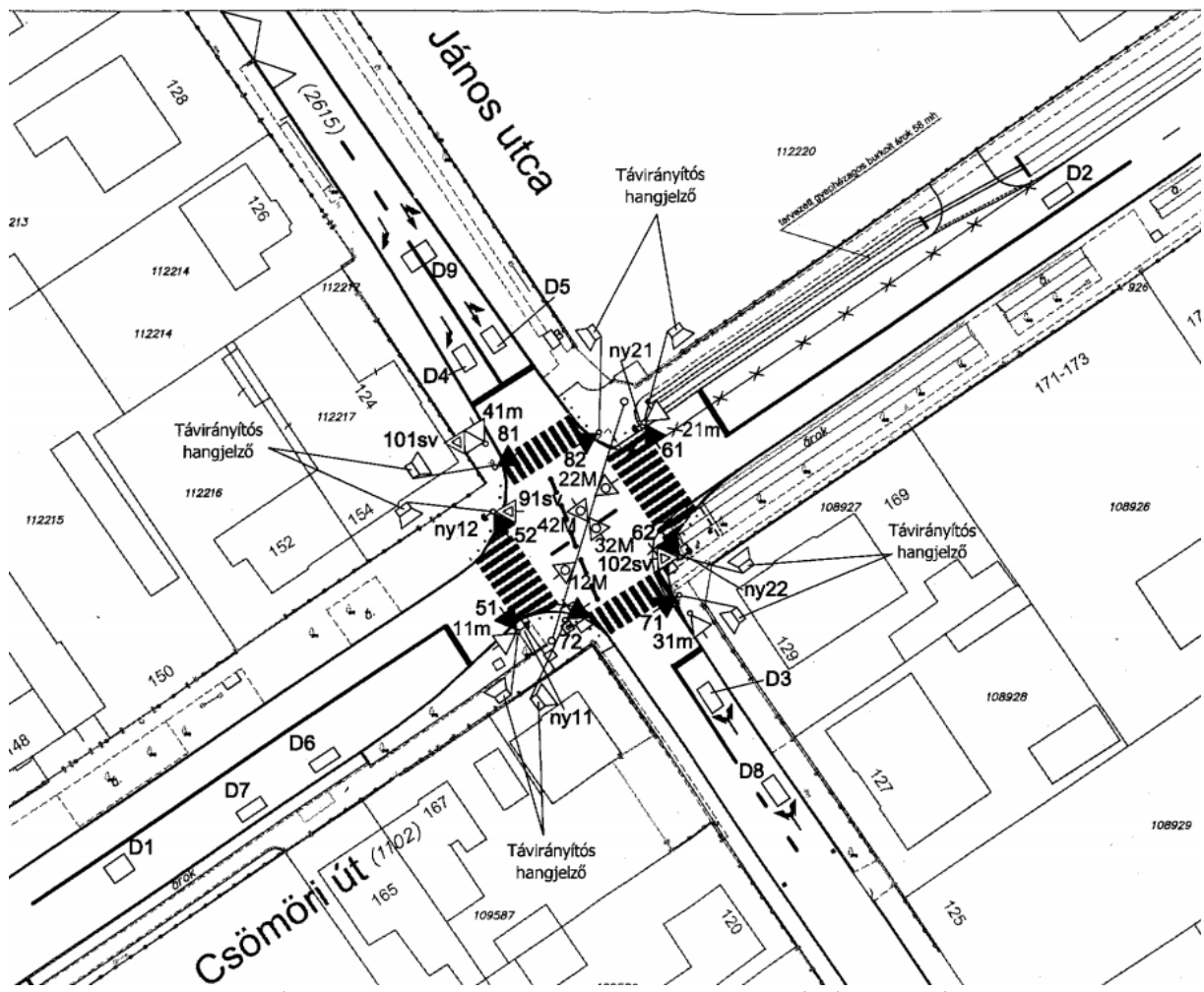
- Néhány eljárás, ami jól használható egy forgalomfüggő program programozásához:
  - `Var.tk1.getProgSek()` → megmondja, hogy hányadik mp-ben tart a ciklusidő
  - `Var.tk1.setProgSek(20);` → legyen a 20. mp-ben a ciklusidő!
  - `Var.ny2.getAnforderung()` → lekérdezi, hogy volt-e bejelentkezés az adott detektoron vagy nyomógombon (Ha volt bejelentkezés, az mindaddig `true` értéken marad, amíg az adott jelzőcsoport zöldet nem kap. Amíg a zöldidő tart a bejelentkezést nem veszi figyelembe a gép!).
  - `Var.d6.belegt()` → lekérdezi a detektort (nyomógombot), hogy éppen foglalt-e?
  - `Var.d1.getBelSek()` → lekérdezi, hogy hány másodperce foglalt már a detektor, amennyiben folyamatosan meghúzott állapotában van.
- Példák egyszerű forgalomfüggőségre:
  - gyalogos bejelentkezés esetén gyalogos zöldidő nyújtása
  - busz bejelentkezés esetén zöldnyújtás a busz irányának
- Úgy állítsa be a programváltást, hogy a ForgfHF működése is a HF bemutatás időtartamára essen!
- Részletezze pár mondatban az elkészített forgalomfüggőség működését!
- FONTOS: a forgalomfüggő program 0.5 másodpercenként fut le az ACTROS-ban, azaz a jelzéstervhez készített feltételeket a gép gyakorlatilag folyamatosan ellenőrzi.

V. A hibátlanul lefordított programot jar fájlba exportálva (VT.JAR néven), ill. a kapcsolódó **forráskódot is** (\*.java fájlokat) pendrive-on kell a HF-bemutatásra elhozni.

- VI. A feladat megoldása során végzett számításokat, a fázisterveket, és a közbensőidőt egy külön dokumentumban kell beadni.
- VII. Néhány hasznos tanács az Eclipse programhoz:
- Az Eclipse fejlesztő környezetben mielőtt lefordítanánk (Build) a forráskódot, minden java fájlt külön-külön mentünk vagy használjuk a „Save All” parancsot!
  - A projektkészítés során 1.3 Java Compiler beállítást használjunk!
  - A jar fájl létrehozása során nulla tömörítést állítsunk be!
- VIII. Szükséges segédletek és szoftverek a házi feladat megoldásához.
- JAVA\_VT\_ACTROS.doc ([www.kjit.bme.hu](http://www.kjit.bme.hu))
  - ACTROS\_VTC\_3000.pdf ([www.kjit.bme.hu](http://www.kjit.bme.hu)):
  - ECLIPSE CLASSIC fejlesztő környezet (<http://www.eclipse.org/downloads>)
  - JAVA SE (Standard Edition)/Java Development Kit (JDK) (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
  - source java fájlok, dokumentáció, vtbib.jar (órán kerül kiadásra)



1. ábra: A csomópont sematikus helyszínrajza



2. ábra : A csomópont forgalomtechnikai helyszínrajza