НИУ «МЭИ»

Кафедра Радиотехнических систем

Расчетно-пояснительная записка к курсовому проекту по

дисциплине «Аппаратура потребителей СРНС»

Выполнил: студ. Бахолдин Н.В.

Группа: ЭР-15-17

Проверил: доц. Корогодин И.В.

Москва 2022

Оглавление

[Цель работы 3](#_Toc98878215)

[Дано 3](#_Toc98878216)

[Решение 3](#_Toc98878217)

# Цель работы

Изучение особенностей сигналов спутников GPS для определения положения спутника по данным с демодулятора его сигнала L1 C/A. На первом этапе происходит моделирование модуля разбора навигационного сообщения до структуры эфемерид.

# Дано

Файл “in.txt” с записанными в нем данными, зафиксированными навигационным приемником по сигналу GPS L1C/A. Файл содержит наблюдения псевдодальностей и прочих радионавигационных параметров, демодулированные и разобранные данные навигационного сообщения.

# Решение

Пункт 1

На первом этапе необходимо создать программу в среде С/ С++, выполняющую функции аналогичные модулю разбора навигационного сообщения. Листинг созданной части программы приведен в приложении.

На данном этапе реализован расчет всех целочисленных параметров спутника. В дальнейшем будет дописана программа для расчета остальных параметров (с плавающей точкой).

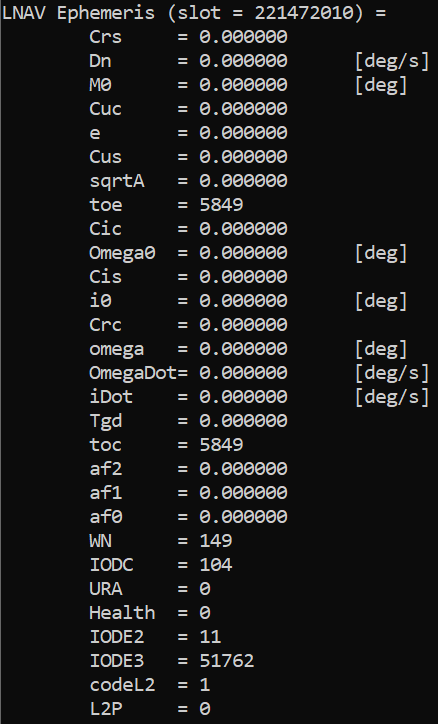


Рисунок 1 – Данные, полученные в ходе компиляции кода через командную строку

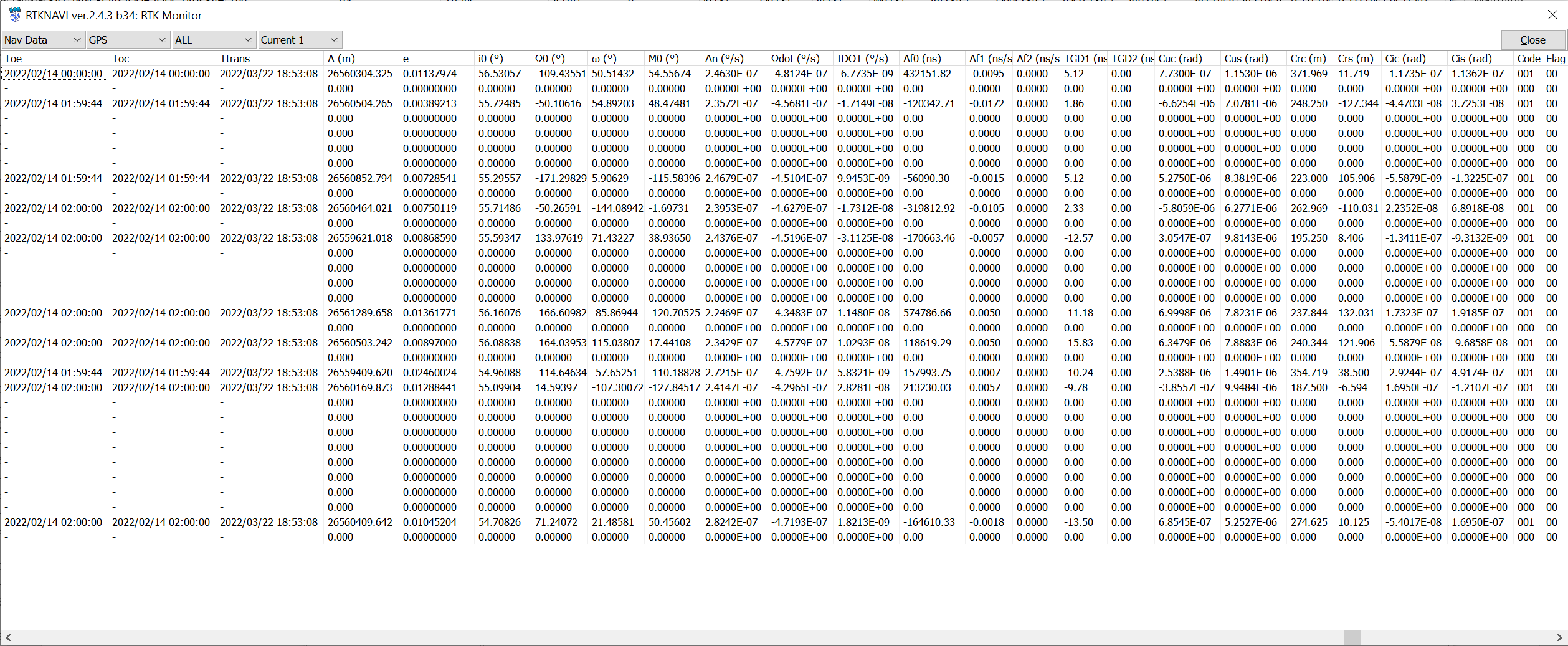


Рисунок 2 – Данные, полученные в программе RTKNAVI

Вывод: Данные, полученные в программе RTKNAVI совпадают с данными, полученными в ходе компиляции кода. На данном этапе программа работает верно.

# Приложение

Листинг программы для пункта 1

#include <stdio.h>

#include <stdint.h>

#include <stdlib.h>

#include <string.h>

struct Ephemeris {

double Crs;

double Dn;

double M0;

double Cuc;

double e;

double Cus;

double sqrtA;

uint32\_t toe;

double Cic;

double Omega0;

double Cis;

double i0;

double Crc;

double omega;

double OmegaDot;

double iDot;

double Tgd;

uint32\_t toc;

double af2;

double af1;

double af0;

uint32\_t WN;

uint16\_t IODC;

uint8\_t URA;

uint8\_t Health;

uint16\_t IODE2;

uint16\_t IODE3;

bool codeL2;

bool L2P;

uint32\_t slot;

};

const int32\_t subFrameLength = 300;

struct SF1\_3 {

uint32\_t slot;

char sf1[subFrameLength+1];

char sf2[subFrameLength+1];

char sf3[subFrameLength+1];

};

void printEmp(Ephemeris\* ep);

int32\_t file2subFrames(SF1\_3\* sf, FILE\* fid, uint8\_t svNum);

int32\_t subFrames2Eph(Ephemeris\* ep, SF1\_3\* subframes);

int main(void)

{

printf(" Hello, World \n");

uint8\_t svNum = 3;

FILE\* fid = fopen("in.txt", "r");

if (fid != nullptr) {

SF1\_3 subframes;

if (!file2subFrames(&subframes, fid, svNum)) {

Ephemeris \*ep = (Ephemeris\*) calloc(1, sizeof(Ephemeris));

if (!subFrames2Eph(ep, &subframes)) {

printEmp(ep);

} else {

printf(" Cannot decode subframes\n ");

}

free(ep);

fclose(fid);

}

else {

printf(" Subframes not found\n ");

}

}

else {

printf(" Cannot open in.txt ");

}

return 0;

}

uint32\_t str2uint(char \*sf, int32\_t start, int32\_t stop) {

uint32\_t ans = 0;

for(int i = start; i < stop; i++) {

bool bit = (sf[i-1] == '1');

ans = ans | (bit << (stop - i - 1));

}

return ans;

}

int32\_t subFrames2Eph(Ephemeris\* ep, SF1\_3\* subframes) {

ep->slot = subframes->slot;

ep->WN = str2uint(subframes->sf1, 61, 71);

ep->toe = str2uint(subframes->sf2, 271, 287);

ep->toc = str2uint(subframes->sf1, 219, 219+16);

ep->IODC = str2uint(subframes->sf1, 83, 93);

ep->URA = str2uint(subframes->sf1, 73, 77);

ep->Health = str2uint(subframes->sf1, 77, 83);

ep->IODE2 = str2uint(subframes->sf2, 61, 69);

ep->IODE3 = str2uint(subframes->sf3, 219, 219+16);

ep->codeL2 = str2uint(subframes->sf1, 71, 73);

ep->L2P = str2uint(subframes->sf1, 91, 92);

return 0;

}

int32\_t file2subFrames(SF1\_3\* sf, FILE\* fid, uint8\_t svNum){

int32\_t sth1, sth2, sth3, sth4, sth5;

char str\_0R[8];

char str\_GPSL1CA[12];

char str\_reh[8];

char str[1000];

uint32\_t svStr;

uint32\_t slot;

int32\_t subFrameNum;

uint32\_t slot\_SF1 = 0;

uint32\_t slot\_SF2 = 0;

uint32\_t slot\_SF3 = 0;

int32\_t readres = 0;

while(readres != EOF)

{

svStr = 0;

readres = fscanf( fid, "%d %d %d %s %s %s %u\t %u %d %d %d %s", &sth1, &sth2, &sth3, str\_0R, str\_GPSL1CA, str\_reh, &svStr, &slot, &sth4, &sth5, &subFrameNum, str);

if (( svStr == svNum ) && (slot >= (604800/6))) {

if ( subFrameNum == 1 ) {

slot\_SF1 = slot;

strncpy(sf->sf1, str, sizeof(sf->sf1));

}

else if (subFrameNum == 2) {

slot\_SF2 = slot;

strncpy(sf->sf2, str, sizeof(sf->sf2));

}

else if (subFrameNum == 3) {

slot\_SF3 = slot;

strncpy(sf->sf3, str, sizeof(sf->sf3));

}

if ((slot\_SF1 + 1 == slot\_SF2) && (slot\_SF2 + 1 == slot\_SF3)) {

sf->slot = slot\_SF1;

return 0;

}

}

}

return 1;

}

void printEmp(Ephemeris\* ep)

{

printf("LNAV Ephemeris (slot = %u) = \n", ep->slot );

printf("\tCrs = %f \n", ep->Crs );

printf("\tDn = %f \t[deg/s] \n", ep->Dn );

printf("\tM0 = %f \t[deg] \n", ep->M0 );

printf("\tCuc = %f \n", ep->Cuc );

printf("\te = %f \n", ep->e );

printf("\tCus = %f \n", ep->Cus );

printf("\tsqrtA = %f \n", ep->sqrtA );

printf("\ttoe = %u \n", ep->toe );

printf("\tCic = %f \n", ep->Cic );

printf("\tOmega0 = %f \t[deg] \n", ep->Omega0 );

printf("\tCis = %f \n", ep->Cis );

printf("\ti0 = %f \t[deg] \n", ep->i0 );

printf("\tCrc = %f \n", ep->Crc );

printf("\tomega = %f \t[deg] \n", ep->omega );

printf("\tOmegaDot= %f \t[deg/s] \n", ep->OmegaDot );

printf("\tiDot = %f \t[deg/s] \n", ep->iDot );

printf("\tTgd = %f \n", ep->Tgd );

printf("\ttoc = %u \n", ep->toc );

printf("\taf2 = %f \n", ep->af2 );

printf("\taf1 = %f \n", ep->af1 );

printf("\taf0 = %f \n", ep->af0 );

printf("\tWN = %u \n", ep->WN );

printf("\tIODC = %u \n", ep->IODC );

printf("\tURA = %u \n", ep->URA );

printf("\tHealth = %u \n", ep->Health );

printf("\tIODE2 = %u \n", ep->IODE2 );

printf("\tIODE3 = %u \n", ep->IODE3 );

printf("\tcodeL2 = %u \n", ep->codeL2 );

printf("\tL2P = %u \n", ep->L2P );

}