

ФГБОУ ВО  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ» »  
ИНСТИТУТ РАДИОТЕХНИКИ И ЭЛЕКТРОНИКИ  
НАПРАВЛЕНИЕ РАДИОЭЛЕКТРОННЫЕ СИСТЕМЫ И КОМПЛЕКСЫ  
КАФЕДРА РАДИОТЕХНИЧЕСКИХ СИСТЕМ



**Курсовой проект**  
**по курсу «Аппаратура потребителей СРНС»**

Выполнил студент:

Муратов Николай Сергеевич

группа: ЭР-15-17

Проверил:

к.т.н., доцент

Корогодин Илья Владимирович

Москва, 2022 г.

# Содержание

<b>1</b>	<b>Обработка логов навигационного приемника</b>	<b>3</b>
1.1	Цель проекта . . . . .	3
1.2	Задание . . . . .	3
1.3	Основная часть . . . . .	4
1.3.1	Теоретическая часть . . . . .	4
1.3.2	Решение задачи . . . . .	6
1.3.3	Приложение 1 . . . . .	9
1.4	Выводы . . . . .	19

## Этап 1

### Обработка логов навигационного приемника

#### 1.1 Цель проекта

Конечная цель всего курсового проекта - получить библиотеку функций на Си++, позволяющую рассчитывать положение спутника GPS по данным с демодулятора его сигнала L1 C/A. На первом этапе реализуем модуль разбора навигационного сообщения до структуры эфемерид, сравним результаты со сторонней программой.

#### 1.2 Задание

В неизвестной локации установлен навигационный приемник, принимающий сигналы GPS L1C/A и логирующий результаты этого приема в формате NVS BINR. Собранный на пятиминутном интервале файл приложен в архиве под именем BINR.bin, см. таблицу вариантов. Файл содержит наблюдения псевдодальностей и прочих радионавигационных параметров, демодулированные и разобранные данные навигационного сообщения.

Для удобства студентов данные демодулятора продублированы в текстовый файл in.txt. Каждая строка файла содержит данные одного сабфрейма одного навигационного сигнала в формате:

1 0 013 0R GpsL1CA 13 212130404 29 125 53 100010111010...

где 13 - номер спутника, 212130404 - счетчик сабфреймов в сигнале, 53 - ID сабфрейма в навигационном сообщении, где в первых трех битах содержится номер сабфрейма в фрейме (5 в данном примере), а далее - номер фрейма в сообщении (6 в данном примере), 1000101110... символы с демодулятора в порядке возрастания времени слева направо.

Муратов Николай Сергеевич: спутник №4.

Требуется:

- Разработать программу, обрабатывающую файл in.txt и выводящую в файл out.txt таблицу эфемерид для спутника согласно варианту в заданном формате.
- Обработать файл BINR.bin с помощью программы RTKNAVI из состава RTKLIB. Определить день и место проведения наблюдений, значения эфемерид для спутника согласно номеру варианта (меню открывается в левом нижнем углу экрана по нажатию на квадрат)
- Сравнить полученные таблицы
- Оформить код программы и разместить на Github
- Оформить отчет по этапу и разместить на Github
- Завести Pull Request

Программа должна компилироваться gcc, все входные данные брать из in.txt, весь вывод осуществлять в out.txt.

## 1.3 Основная часть

### 1.3.1 Теоретическая часть

Целью передачи навигационных сообщений является возможность декодирования битового потока в параметры, с помощью которых можно определить положение устройства-приемника. Для определения своего местоположения требуется весь кадр сообщения. Но это не означает, что требуется непрерывный сбор данных. Это было бы очень неэффективно по времени и бесполезно. Как только данные были получены, они имеют срок службы, в течение которого они действительны. GPS эфемериды действительны в течение 2 часов, GPS альманах - в течение 24 часов. С момента получения этих данных, необходимо только постоянно обновлять время, коды дальности (и некоторые другие).

LNAV состоит из пяти подкадров. Подкадры 1-3 не содержат страниц, они декодируются сами по себе, поэтому они рассматриваются как наименьший декодируемый блок и включаются сами по себе. Подкадры 4-5 содержат определенные страницы. Каждая страница декодируется сама по себе, поэтому каждая страница рассматривается как наименьший декодируемый блок, но в данной работе нас интересуют лишь первые 3 подкадра.

Каждый подкадр состоит из 10 слов по 30 бит, то есть подкадр состоит из 300 символов. Одно слово может содержать как один, так и сразу несколько параметров. Но также существуют параметры разделенные на несколько слов (splitted parameters), пример их расположения можно увидеть на рис. 1.3.1.

Второй подкадр					
2	Preamble	Preamble	8	0	7
2	TLM_MSG	Telemetry_Message	14	8	21
2	ISF	Integrity_Status_Flag	1	22	22
2	Reserved	Reserved	1	23	23
2	P	Parity	6	24	29
2	MSG_TOW	Message_Time_Of_Week_Count	17	30	46
2	ALERT	Alert_Flag	1	47	47
2	ASF	Anti_Spoof_Flag	1	48	48
2	Sub_ID	Subframe_ID	3	49	51
2	t	Parity_Computation	2	52	53
2	P	Parity	6	54	59
2	IODE	E_IOD_Ephemeris	8	60	67
2	C_rs	E_Harmonic_Corr_rs	16	68	83
2	P	Parity	6	84	89
2	DEL_n	E_Mean_Motion_Diff	16	90	105
2	M_0	E_Mean_Anomaly_At_Ref_T	8	106	113
2	P	Parity	6	114	119
2	M_0	E_Mean_Anomaly_At_Ref_T	24	120	143
2	P	Parity	6	144	149
2	C_uc	E_Harmonic_Corr_uc	16	150	165
2	e	E_Eccentricity	8	166	173
2	P	Parity	6	174	179
2	e	E_Eccentricity	24	180	203
2	P	Parity	6	204	209
2	C_us	E_Harmonic_Corr_us	16	210	225
2	A_1/2	E_SQRT_Semi_Major_Axis	8	226	233
2	P	Parity	6	234	239
2	A_1/2	E_SQRT_Semi_Major_Axis	24	240	263
2	P	Parity	6	264	269
2	t_0e	E_Ref_T	16	270	285
2	FIF	Fit_Interval_Flag	1	286	286
2	AODO	Age_Of_Data_Offset	5	287	291
2	t	Parity_Computation	2	292	293
2	P	Parity	6	294	299

Рисунок 1.3.1 — LNAV Подкадр(Subframe) 2 структурирован в электронную таблицу. Строки, выделенные красным цветом являются разделенными параметрами

### 1.3.2 Решение задачи

Имеем входные данные:

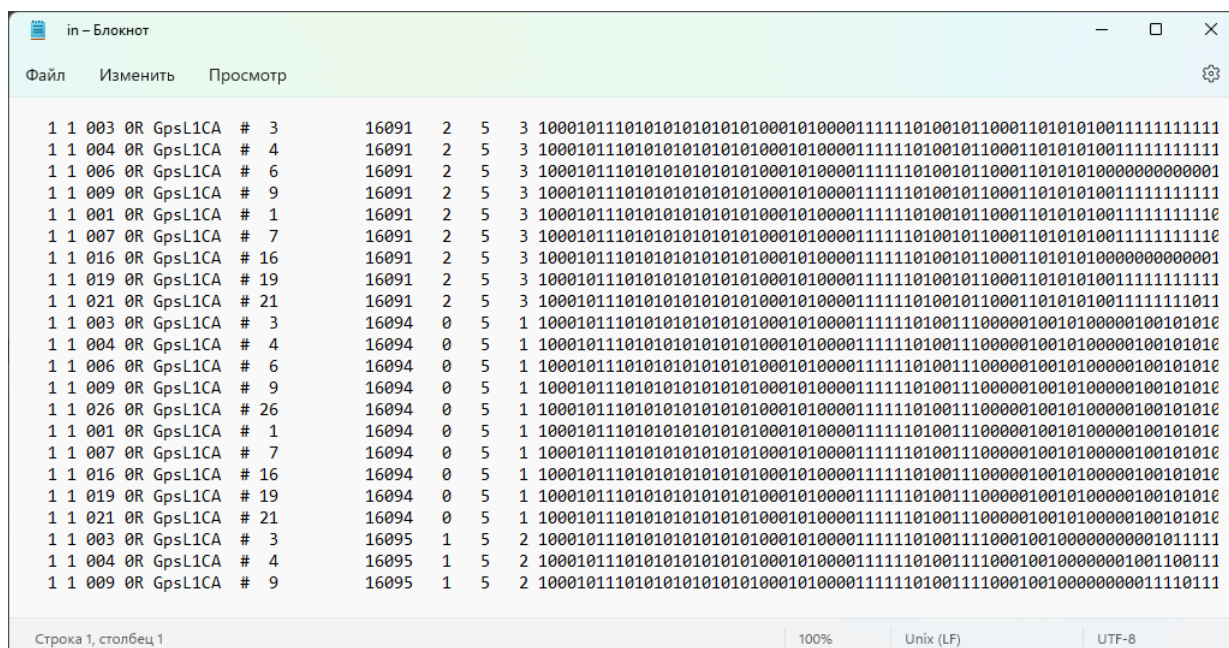
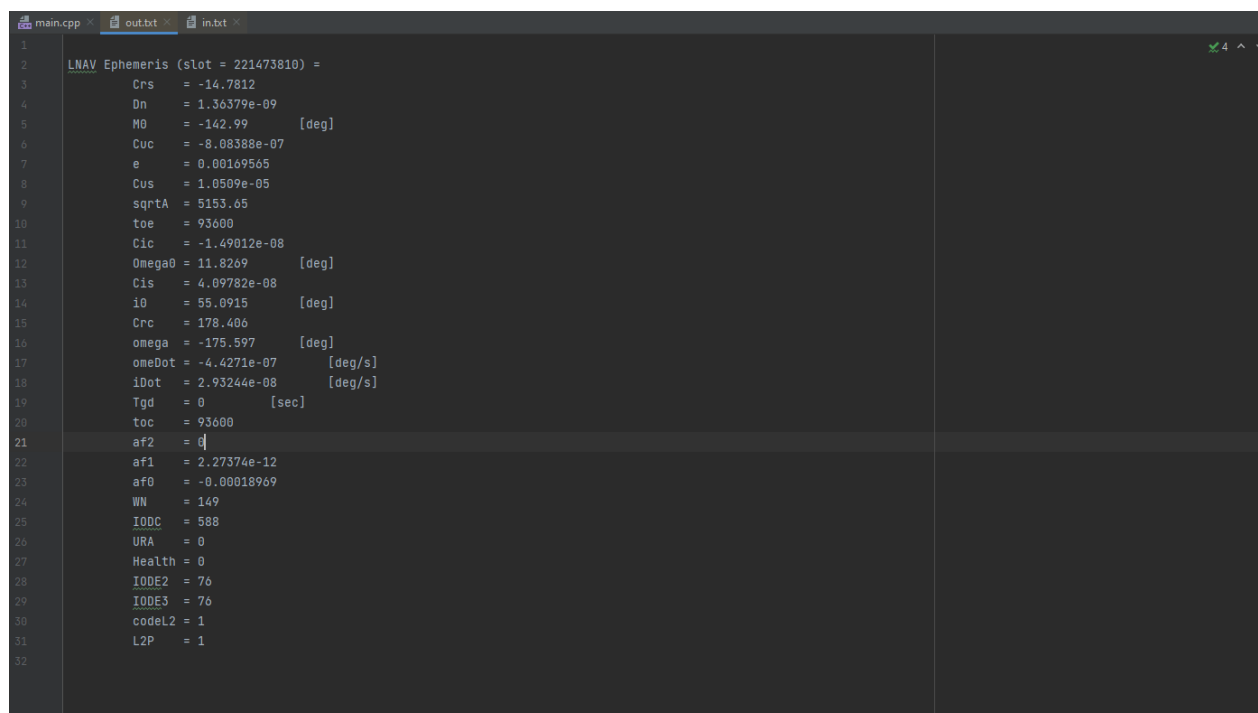


Рисунок 1.3.2 — Входной файл in.txt

Задача решается в несколько следующих этапов:

- Открыть файл;
- Найти в потоке подкадров первые три, исходящие от необходимого спутника;
- Извлечь их в отдельную структуру данных;
- Достать из подкадров данные эфемерид (данный этап для некоторых переменный состоит из двух действий, т.к. необходимо не только выбрать необходимые биты, но и преобразовать их из дополнительного кода в прямой с помощью оператора побитового дополнения - это унарный оператор (работает только с одним операндом). Он обозначается символом  $\sim$  и меняет двоичные цифры 1 на 0 и 0 на 1;
- Вывести найденные параметры в терминал;
- Сохранить в файл out.txt

Листинг программы располагается в Приложении 1. Выходной файл out.txt:



```
1
2 LNAV Ephemeris (slot = 221473810) =
3   Crs   = -14.7812
4   Dn    = 1.36379e-09
5   M0    = -142.99      [deg]
6   Cus   = -8.08388e-07
7   e     = 0.00109565
8   Cus   = 1.0509e-05
9   sqrtA = 5153.65
10  toe    = 93600
11  Cic    = -1.49012e-08
12  Omega0 = 11.8269     [deg]
13  Cis    = 4.09782e-08
14  i0     = 55.0915     [deg]
15  Crc    = 178.406
16  omega  = -175.597    [deg]
17  omeDot = -4.4271e-07 [deg/s]
18  iDot   = 2.93244e-08 [deg/s]
19  Tgd    = 0           [sec]
20  toc    = 93600
21  af2    = 0
22  af1    = 2.27374e-12
23  af0    = -0.00018969
24  WN     = 149
25  IODC   = 588
26  URA    = 0
27  Health = 0
28  IODE2  = 76
29  IODE3  = 76
30  codeL2 = 1
31  L2P    = 1
32
```

Рисунок 1.3.3 — Выходной файл

Приведем данные, обработанные open-source программой RTKNAVI  
рис. 1.3.5:

RTKNAVI ver.2.4.2: RTK Monitor

Nav GPS

All

Current

Close

SAT	PRN	Status	MODE	IDOC	Acci	Health	Toe	Toc	Trans	A (m)	e	i0 (deg)	OMEGA0 (deg)	omega (deg)	M0 (deg)	deltan (deg/s)	
G01	1	-	-	5911	23	0	00	2022/02/14 00:00:00	2022/02/14 00:00:00	2022/04/15 14:46:39	26560304.325	0.01137974	56.53057	-109.43551	50.51432	54.55674	2.4630E-07
G02	2	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G03	3	OK	-	2827	11	0	00	2022/02/14 01:59:44	2022/02/14 01:59:44	2022/04/15 14:46:39	26560504.265	0.00389213	55.72485	-50.10616	54.89203	48.47481	2.3572E-07
G04	4	OK	-	1953	588	0	00	2022/02/14 02:00:00	2022/02/14 02:00:00	2022/04/15 14:46:39	26560157.291	0.00169565	55.09149	11.82685	-175.5971E	-142.9901E	2.4548E-07
G05	5	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G06	6	OK	-	2210	86	0	00	2022/02/14 02:00:00	2022/02/14 02:00:00	2022/04/15 14:46:39	26560290.249	0.00269551	56.49500	-109.91291	-55.01959	133.64356	2.4669E-07
G07	7	OK	-	8995	35	0	00	2022/02/14 04:00:00	2022/02/14 04:00:00	2022/04/15 14:46:39	26560462.036	0.01554191	54.46659	70.20050	-131.2703E	136.52698	2.8489E-07
G08	8	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G09	9	OK	-	7710	30	0	00	2022/02/14 02:00:00	2022/02/14 02:00:00	2022/04/15 14:46:39	26559848.321	0.00225987	54.69112	8.79464	107.10939	-94.65112	2.5565E-07
G10	10	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G11	11	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G12	12	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G13	13	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G14	14	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G15	15	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G16	16	OK	-	5140	20	0	00	2022/02/14 04:00:00	2022/02/14 04:00:00	2022/04/15 14:46:39	26559618.285	0.01275745	55.59110	135.03970	40.56887	-7.13304	2.4325E-07
G17	17	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G18	18	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G19	19	OK	-	9766	38	0	00	2022/02/14 02:00:00	2022/02/14 02:00:00	2022/04/15 14:46:39	26560503.242	0.00897000	56.08838	-164.03953	115.03807	17.44108	2.3429E-07
G20	20	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G21	21	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G22	22	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G23	23	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G24	24	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G25	25	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G26	26	OK	-	2313	9	0	00	2022/02/14 02:00:00	2022/02/14 02:00:00	2022/04/15 14:46:39	26560614.536	0.00687526	53.77037	126.89030	21.18201	-13.00067	2.8919E-07
G27	27	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G28	28	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G29	29	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G30	30	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	
G31	31	OK	-	6168	24	0	00	2022/02/14 02:00:00	2022/02/14 02:00:00	2022/02/14 03:01:19	26560409.642	0.01045204	54.70826	71.24072	21.48581	50.45602	2.8242E-07
G32	32	-	-	-	-	0	00	-	-	0.000	0.00000000	0.00000	0.00000	0.00000	0.00000	0.0000E+00	

Рисунок 1.3.4 — RTKNAVI результат 1/2

RTKNAVI ver.2.4.2: RTK Monitor

Nav GPS

All

Current

Close

deltan (deg/s)	OMEGAdot (d)	IDOT (deg/s)	a0 (ns)	af1 (ns/s)	af2 (ns/s2)	TGD (ns)	BGD5a(ns)	BGD5b(ns)	Cuc(rad)	Cus(rad)	Crc(m)	Crs(m)	Cic(rad)	Cis(rad)	Code	Flag
2.4630E-07	-4.8124E-07	-6.7735E-09	432151.8	-0.0095	0.0000	5.1	0.0	0.0	7.7300E-07	1.1530E-06	3.7197E+02	1.1719E+01	-1.1735E-07	1.1362E-07	1	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
2.3572E-07	-4.5681E-07	-1.7149E-08	-120342.7	-0.0172	0.0000	1.9	0.0	0.0	-6.6254E-06	7.0781E-06	2.4825E+02	-1.2734E+02	-4.4703E-08	3.7253E-08	1	0
2.4548E-07	-4.4271E-07	2.9324E-08	-189689.9	0.0023	0.0000	-4.2	0.0	0.0	-8.0839E-07	1.0509E-05	1.7841E+02	-1.4781E+01	-1.4901E-08	4.0978E-08	1	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
2.4669E-07	-4.7165E-07	-7.1623E-10	200809.4	0.0119	0.0000	3.7	0.0	0.0	1.0580E-06	5.7183E-07	3.7962E+02	1.5906E+01	6.5193E-08	-1.3039E-08	1	0
2.8489E-07	-4.5906E-07	7.3464E-09	310124.8	0.0027	0.0000	-11.2	0.0	0.0	3.5018E-07	5.3551E-06	2.7103E+02	1.0562E+01	-6.5193E-08	-2.8871E-07	1	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
2.5565E-07	-4.4940E-07	2.5354E-08	-354265.8	0.0016	0.0000	1.4	0.0	0.0	-1.2182E-06	1.0680E-05	1.7066E+02	-2.2969E+01	-2.4214E-08	1.0431E-07	1	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000	0.0000	0.0	0.0	0.0	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0	0
0.0000E+00	0.0000E+00	0.0000E+00	0.0	0.0000												



### 1.3.3 Приложение 1

Листинг 1.1 — Программа декодирования подкадров навигационного сообщения GPS

```
1 // Made on Earth by Murathon.
2
3
4 #include <iostream>
5 #include <fstream>
6 #include <windows.h>
7 #include <string>
8 #include <stdlib.h>
9 #include <cmath>
10 #include <stdio.h>
11 #include <bitset>
12
13 #define P2_5      pow(2,-5)
14 #define P2_55     pow(2,-55)
15 #define P2_43     pow(2,-43)
16 #define P2_31     pow(2,-31)
17 #define P2_4      pow(2,4)
18 #define P2_19     pow(2,-19)
19 #define P2_29     pow(2,-29)
20 #define P2_33     pow(2,-33)
21 #define SemiCirc  180
22
23
24 using namespace std;
25
26 struct Ephemeris {
27     double    Crs;
28     double    Dn;
29     double    M0;
30     double    Cuc;
31     double    e;
32     double    Cus;
33     double    sqrtA;
34     uint32_t  toe;
35     double    Cic;
```

```

36     double    Omega0;
37     double    Cis;
38     double    i0;
39     double    Crc;
40     double    omega;
41     double    OmegaDot;
42     double    iDot;
43     int16_t    Tgd;
44     uint32_t    toc;
45     double    af2;
46     double    af1;
47     double    af0;
48     uint16_t    WN;
49     uint16_t    IODC;
50     uint32_t    URA;
51     uint32_t    Health;
52     uint16_t    IODE2;
53     uint16_t    IODE3;
54     bool        codeL2;
55     bool        L2P;
56     uint32_t    slot;
57 };
58
59 struct nssrSF {
60     uint32_t    slot;
61     string    sf1;
62     string    sf2;
63     string    sf3;
64 };
65
66 void parsStr(nssrSF *Rdata);
67 uint32_t pickSplitParam(string sf, uint16_t FS, int HMF, uint16_t
    FtS, int HMS);
68 void printEPH(Ephemeris* EPH);
69 void saveEPH(Ephemeris* EPH);
70 void decodeSF(Ephemeris* EPH, nssrSF *data);
71 int64_t pick32(string sf, int32_t FrmN, int HmR);
72
73
74 int main(void)

```

```

75 {
76     nssrSF data;
77     parsStr(&data);
78     cout << data.sf1 << endl << data.sf2 << endl << data.sf3 << endl
;
79     Ephemeris *EPH = (Ephemeris*) calloc(1, sizeof(Ephemeris));
80     decodeSF(EPH,&data);
81     printEPH(EPH);
82     saveEPH(EPH);
83     free(EPH);
84
85
86 }
87
88 void parsStr(nssrSF *Rdata)
89 {
90
91     SetConsoleOutputCP(CP_UTF8);
92     string path = "in.txt";
93
94     ifstream fin;
95     fin.open(path);
96
97     if(fin.is_open()) {
98
99         cout << "Файл " << "открыт." << endl;
100         while (!fin.eof()) {
101             int N = 3;
102             string rubber;
103             int nmbr_stlt;
104             uint32_t slot;
105             uint32_t subFrameNum;
106             string str;
107
108             uint32_t slot_SF1;
109             uint32_t slot_SF2;
110             uint32_t slot_SF3;
111
112             int mass[N];
113             for(int i=0;i<N;++i)

```

```

114         {
115             fin >> mass[i];
116         }
117         fin >> rubber >> rubber >> rubber;
118         fin >> nmbr_stlt >> slot >> rubber >> rubber >>
subFrameNum;
119         fin >> str;
120
121         if (nmbr_stlt == 4 and slot >= 604800/6){
122
123             if (subFrameNum == 1)
124             {
125                 slot_SF1 = slot;
126                 Rdata->sf1 = str;
127             }
128             else if (subFrameNum == 2)
129             {
130                 slot_SF2 = slot;
131                 Rdata->sf2 = str;
132             }
133             else if (subFrameNum == 3)
134             {
135                 slot_SF3 = slot;
136                 Rdata->sf3 = str;
137             }
138
139
140             if (slot_SF1 + 1 == slot_SF2 and slot_SF2 + 1 ==
slot_SF3) {
141                 Rdata->slot = slot_SF1;
142                 return;
143             }
144
145         }
146         //
147     }
148 }
149 else
150 {
151     cout << "Ошибка" открытия файла!!!" << endl;

```

```

152     }
153     fin.close();
154
155
156 }
157 void saveEPH(Ephemeris* EPH)
158 {
159     ofstream fout;
160     string path = "out.txt";
161     fout.open(path);
162     if(fout.is_open()) {
163
164         cout << "Выгружаю" << endl;
165         fout << endl << "LNAV Ephemeris (slot = " << EPH->slot << ")
=< endl;
166         fout << "\t\t Crs = " << EPH->Crs << endl;
167         fout << "\t\t Dn      = " << EPH->Dn << endl;
168         fout << "\t\t M0      = " << EPH->M0 << "\t\t[deg]" << endl;
169         fout << "\t\t Cuc = " << EPH->Cuc << endl;
170         fout << "\t\t e      = " << EPH->e << endl;
171         fout << "\t\t Cus = " << EPH->Cus << endl;
172         fout << "\t\t sqrtA = " << EPH->sqrtA << endl;
173         fout << "\t\t toe = " << EPH->toe << endl;
174         fout << "\t\t Cic = " << EPH->Cic << endl;
175         fout << "\t\t Omega0 = " << EPH->Omega0 << "\t\t[deg]" <<
endl;
176         fout << "\t\t Cis = " << EPH->Cis << endl;
177         fout << "\t\t i0      = " << EPH->i0 << "\t\t[deg]" << endl;
178         fout << "\t\t Crc = " << EPH->Crc << endl;
179         fout << "\t\t omega = " << EPH->omega << "\t\t[deg]" << endl
;
180         fout << "\t\t omeDot = " << EPH->OmegaDot << "\t\t[deg/s]"
<< endl;
181         fout << "\t\t iDot  = " << EPH->iDot << "\t\t[deg/s]" <<
endl;
182         fout << "\t\t Tgd = " << EPH->Tgd << "\t\t\t[sec]" << endl;
183         fout << "\t\t toc = " << EPH->toc << endl;
184         fout << "\t\t af2 = " << EPH->af2 << endl;
185         fout << "\t\t af1 = " << EPH->af1 << endl;
186         fout << "\t\t af0 = " << EPH->af0 << endl;

```

```

187     fout << "\t\t WN      = " << EPH->WN << endl;
188     fout << "\t\t IODC   = " << EPH->IODC << endl;
189     fout << "\t\t URA    = " << EPH->URA << endl;
190     fout << "\t\t Health  = " << EPH->Health << endl;
191     fout << "\t\t IODE2   = " << EPH->IODE2 << endl;
192     fout << "\t\t IODE3   = " << EPH->IODE3 << endl;
193     fout << "\t\t codeL2  = " << EPH->codeL2 << endl;
194     fout << "\t\t L2P    = " << EPH->L2P << endl;
195 }
196 else
197 {
198     cout << "Ошибка " открытия файла !!!" << endl;
199 }
200 fout.close();
201 cout << "Готово" !";
202 }
203
204 void printEPH(Ephemeris* EPH)
205 {
206
207     cout << endl << "LNAV Ephemeris (slot = " << EPH->slot << ") ="
<< endl;
208     cout << "\t\t Crs = " << EPH->Crs << endl;
209     cout << "\t\t Dn   = " << EPH->Dn << endl;
210     cout << "\t\t M0   = " << EPH->M0 << "\t\t[deg]" << endl;
211     cout << "\t\t Cuc = " << EPH->Cuc << endl;
212     cout << "\t\t e   = " << EPH->e << endl;
213     cout << "\t\t Cus = " << EPH->Cus << endl;
214     cout << "\t\t sqrtA = " << EPH->sqrtA << endl;
215     cout << "\t\t toe = " << EPH->toe << endl;
216     cout << "\t\t Cic = " << EPH->Cic << endl;
217     cout << "\t\t Omega0 = " << EPH->Omega0 << "\t\t[deg]" << endl;
218     cout << "\t\t Cis = " << EPH->Cis << endl;
219     cout << "\t\t i0   = " << EPH->i0 << "\t\t[deg]" << endl;
220     cout << "\t\t Crc = " << EPH->Crc << endl;
221     cout << "\t\t omega = " << EPH->omega << "\t\t[deg]" << endl;
222     cout << "\t\t omeDot = " << EPH->OmegaDot << "\t\t[deg/s]" <<
endl;
223     cout << "\t\t iDot  = " << EPH->iDot << "\t\t[deg/s]" << endl;
224     cout << "\t\t Tgd   = " << EPH->Tgd << "\t\t\t[sec]" << endl;

```

```

225     cout << "\t\t toc = " << EPH->toc << endl;
226     cout << "\t\t af2 = " << EPH->af2 << endl;
227     cout << "\t\t af1 = " << EPH->af1 << endl;
228     cout << "\t\t af0 = " << EPH->af0 << endl;
229     cout << "\t\t WN = " << EPH->WN << endl;
230     cout << "\t\t IODC = " << EPH->IODC << endl;
231     cout << "\t\t URA = " << EPH->URA << endl;
232     cout << "\t\t Health = " << EPH->Health << endl;
233     cout << "\t\t IODE2 = " << EPH->IODE2 << endl;
234     cout << "\t\t IODE3 = " << EPH->IODE3 << endl;
235     cout << "\t\t codeL2 = " << EPH->codeL2 << endl;
236     cout << "\t\t L2P = " << EPH->L2P << endl;
237
238 }
239 int64_t pick32(string sf, int32_t FrmN, int HmR) {
240     int64_t ans = 0;
241     int64_t Rans = 0;
242     for (int i = FrmN; i < FrmN+HmR; i++) {
243         ans = (ans | ((sf[i - 1] == '1') ? 1 : 0));
244         cout << sf[i-1];
245         if (i < FrmN+HmR-1){
246             ans = ans<<1;
247         }
248
249     }
250     return ans;
251
252
253 }
254
255 int64_t compl2int(uint64_t ans, int HmZ){
256     int64_t Rans = 0;
257     if (HmZ == 8){
258         if (bool((1<<7) & ans)){
259             ans /= 0xFFFFFFFFFFFFFFF00;
260             Rans = ~(ans-1);
261             /*cout<< endl << bitset<64>(Rans).to_string() << endl;*/
262             return -Rans;
263         }
264

```

```

265 }
266 if (HmZ == 14){
267     if (bool((1<<13) & ans)){
268         ans |= 0xFFFFFFFFFFC000;
269         Rans = ~(ans-1);
270         return -Rans;
271     }
272
273 }
274 if (HmZ == 16){
275     if (bool((1<<15) & ans)){
276         ans |= 0xFFFFFFFFF0000;
277         Rans = ~(ans-1);
278         return -Rans;
279     }
280
281 }
282 if (HmZ == 22){
283     if (bool((1<<21) & ans)){
284         ans |= 0xFFFFFFFFC00000;
285         Rans = ~(ans-1);
286         return -Rans;
287     }
288
289 }
290 if (HmZ == 24){
291     if (bool((1<<23) & ans)){
292         ans |= 0xFFFFFFFF000000;
293         Rans = ~(ans-1);
294         return -Rans;
295     }
296
297 }
298 if (HmZ == 32){
299     if (bool((1<<31) & ans)){
300         ans |= 0xFFFFFFFF00000000;
301         Rans = ~(ans-1);
302         return -Rans;
303     }
304

```



```

305     }
306     return ans;
307 }
308 uint32_t pickSplitParam(string sf, uint16_t FS, int HMF, uint16_t
    FtS, int HMS) {
309     uint32_t ans = 0;
310
311     for (int i = FS; i < FS+HMF; i++) {
312         ans = (ans | ((sf[i-1] == '1')? 1 : 0)) << 1;
313     }
314     for (int i = FtS; i < FtS+HMS; i++) {
315         ans = ans | ((sf[i-1] == '1')? 1 : 0);
316         if (i < FtS+HMS-1){
317             ans = ans<<1;
318         }
319     }
320
321     return ans;
322 }
323 }
324
325 void decodeSF(Ephemeris* EPH, nssrSF *data){
326
327     EPH->slot = data->slot;
328
329     EPH->Crs = compl2int(pick32(data->sf2,69,16),16)*P2_5;
330
331     EPH->Dn = compl2int(pick32(data->sf2,91,16),16)*P2_43;
332
333     EPH->M0 = compl2int(pickSplitParam(data->sf2,107, 8, 121, 24)
    ,32)*P2_31*SemiCirc;
334
335     EPH->Cuc = compl2int(pick32(data->sf2,151,16),16)*P2_29;
336
337     EPH->e = pickSplitParam(data->sf2,167, 8, 181, 24) * P2_33;
338
339     EPH->Cus = compl2int(pick32(data->sf2,211,16),16)*P2_29;
340
341     EPH->sqrta = pickSplitParam(data->sf2,227, 8, 241, 24) * P2_19;
342

```

```

343  EPH->toe = pick32(data->sf2,271,16)*pow(2,4);
344
345  EPH->Cic = compl2int(pick32(data->sf3,61,16),16)*P2_29;
346
347  EPH->Omega0 = compl2int(pickSplitParam(data->sf3,77, 8, 91, 24)
,32)*P2_31*SemiCirc;
348
349  EPH->Cis = compl2int(pick32(data->sf3,121,16),16)*P2_29;
350
351  EPH->i0 = compl2int(pickSplitParam(data->sf3,137, 8, 151, 24)
,32)*P2_31*SemiCirc;
352
353  EPH->Crc = compl2int(pick32(data->sf3,181,16),16)*P2_5;
354
355  EPH->omega = compl2int(pickSplitParam(data->sf3,197, 8, 211, 24)
,32)*P2_31*SemiCirc;
356
357  EPH->OmegaDot = compl2int(pick32(data->sf3,241,24),24)*P2_43*
SemiCirc;
358
359  EPH->iDot = compl2int(pick32(data->sf3,279,14),14)*P2_43*
SemiCirc;
360
361  EPH->Tgd = compl2int(pick32(data->sf1,197,8),8)*P2_31;
362
363  EPH->toc = compl2int(pick32(data->sf1,219,16),16)*P2_4;
364
365  EPH->af2 = compl2int(pick32(data->sf1,241,8),8)*P2_55;
366
367  EPH->af1 = compl2int(pick32(data->sf1,249,16),16)*P2_43;
368
369  EPH->af0 = compl2int(pick32(data->sf1,271,22),22)*P2_31;
370
371  EPH->WN = pick32(data->sf1,61,10);
372
373  EPH->IODC = pickSplitParam(data->sf1,83, 2, 211, 8);
374
375  EPH->URA = pick32(data->sf1,73,4);
376
377  EPH->Health = EPH->IODE2 = pick32(data->sf1,73,6);

```

```

378
379     EPH->IODE2 = pick32(data->sf2,61,8);
380
381     EPH->IODE3 = pick32(data->sf3,271,8);
382
383     EPH->codeL2 = pick32(data->sf1,71,2);
384
385     EPH->L2P = data->sf1[90];
386
387 }

```

## 1.4 Выводы

На первом этапе были получены эфемериды для дальнейшего написания функцию расчета положения спутника GPS на языке python.

## Литература

- [1] Interface Control Contractor: SAIC (GPS SEI) 200 N. Pacific Coast Highway,  
Suite 1800 El Segundo, CA 90245
- [2] Интернет-ресурс:  
<https://www.programiz.com/cpp-programming/bitwise-operators>