

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Национальный исследовательский университет «МЭИ»

Институт: ИРЭ

Кафедра: РТС

Специальность:

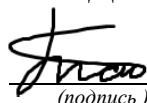
11.05.01 Радиоэлектронные системы и
комплексы

ОТЧЕТ по практике

**Наименование
практики:**

Производственная практика: научно-
исследовательская работа

СТУДЕНТ



(подпись)

/ Тасканов В.Е. /

(Фамилия и инициалы)

Группа ЭР-15-16

(номер учебной группы)

**ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ
ПО ПРАКТИКЕ**

ОТЛИЧНО

(отлично, хорошо, удовлетворительно, неудовлетворительно,
зачтено, не зачтено)

/ Куликов Р.С. /

(подпись)

(Фамилия и инициалы члена комиссии)

/ Шатилов А.Ю. /

(подпись)

(Фамилия и инициалы члена комиссии)

**Москва
2020**

СПИСОК ИСПОЛНИТЕЛЕЙ

студент

Тасканов В.Е.

Содержание

ГЛАВА 1.ТЕОРИЯ	7
1.1.Псевдо дальномерный метод.....	7
1.2.Оценка координат потребителя с учетом ошибок SISRE.....	8
1.2.1.Постановка задачи нахождения оценки координаты потребителя с учетом ошибок SISRE.....	10
1.2.2. Решения задачи нахождения оценки координаты потребителя с учетом ошибок SISRE.....	11
1.3. Описание движения спутников ГЛОНАСС по орбитам	13
1.4. Расчет выбранной точки в декартовой системе координат.....	14
1.5. Расчет видимости спутников	15
2.1. Создание репозитория.....	17
2.2. Проверка правильности работы системы «Git»	21
2.3. Добавления в коллаборацию	23
3.1. Необходимые файлы для сборки проекта	26
3.2.Изменения по коду.....	27
3.2.1.Алгоритм описания движения спутников ГЛОНАСС по орбитам и нахождения координат спутника в определённый момент	28
3.2.3. Алгоритм расчета угла	29
3.2.4. Нахождения оценки координаты потребителя с учетом ошибок SISRE.....	30
3.3. Библиотека “Armadillo”.....	32
ГЛАВА 3.РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ	33

ЗАКЛЮЧЕНИЕ	39
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	39
ПРИЛОЖЕНИЕ	40
Приложение 3	42
Приложение 4	51
Приложение 5	54
Приложение 6	55
Приложение 7	59
Приложение 8	60
Приложение 9	61
Приложение 10	61
Приложение 11	66
Приложение 12	67
Приложение 13	68
Приложение 14	68

Введение

Спутниковые радионавигационные системы (СРНС) являются самыми точными системами по определению координат потребителя. Они стали важной частью в различных сферах нашей жизни. Наиболее распространенными являются системы ГЛОНАСС (Россия), GPS (США), Galileo (Евросоюз).

В 9 семестре стояла **цель работы** – откорректировать предыдущую программу, добавить в программу алгоритм описание движения спутников ГЛОНАСС по орбитам, также добавить алгоритм расчета видимости спутников и нахождения оценки координаты потребителя с учетом ошибок SISRE.

В рамках данной цели решаются следующие задачи:

1. Перерасчет координат потребителя из геодезической системы координат в декартовую систему координат
2. Определение алгоритма описания движения спутников ГЛОНАСС по орбитам
3. Определения алгоритма расчета видимости спутников для системы ГЛОНАСС
4. Нахождения оценки координаты потребителя с учетом ошибок SISRE.

ГЛАВА 1. ТЕОРИЯ

1.1. Псевдо дальномерный метод

Псевдо дальномерный метод или метод определения координат объекта по измерениям квазидальности рассмотрен в работе.

Под псевдо дальностью от i -го НС до потребителя понимают измеренную дальность $D_{изм\ i}$ до этого НС, отличающуюся от истинной дальности D_i на неизвестную, но постоянную за время определения навигационных параметров величину D' . Таким образом, псевдо дальность до i -го НС

$$D_{изм\ i} = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} + D' \quad (1.1)$$

В псевдо дальномерных методах, основанных на измерениях псевдо дальностей, в качестве навигационного параметра выступает $D_{изм\ i}$. Поверхностью положения ПО по-прежнему является сфера с центром в точке центра масс НС, но радиус этой сферы изменен на неизвестную величину D' . Измерение псевдо дальностей до трех НС приводит к системе трех уравнений с четырьмя неизвестными (x, y, z, D'). В решении этой системы уравнений возникает неопределенный параметр, и для устранения возникшей неопределенности необходимо провести дополнительное измерение, т. е. измерить псевдо дальность до четвертого спутника. Полученная таким образом система четырех уравнений имеет точное решение, и следовательно, местоположение потребителя при измерениях псевдо дальностей определяется как точка пересечения четырех поверхностей положения.

Необходимость нахождения в зоне видимости четырех НС предъявляет достаточно жесткие требования к структуре сети НС, которые выполняются только в среднеорбитальных СРНС.

Важно отметить, что если геоцентрическая высота объекта априорно известна, то число минимально необходимых для решения навигационной задачи НС сокращается на один (с четырех до трех). Используя изложенные в

[2] приемы, нетрудно получить алгоритмы решения навигационных задач в конечном виде для подвижного объекта с известной высотой. Однако следует подчеркнуть, что для априорного вычисления геоцентрической высоты требуется знать, в частности, земной радиус-вектор, который является функцией широты места. В этом случае навигационную задачу можно решить с высокой точностью лишь путем последовательных приближений.

Псевдо дальномерный метод не накладывает жестких ограничений на значение погрешности $D' = ct'$ (погрешности временной шкалы) и позволяет одновременно с определением местоположения вычислять отклонение шкалы времени потребителя, однако для его реализации необходимо принимать сигналы от 4 НС. Но данный метод обладает более высокой точностью, по сравнению с дальномерным. [1]

1.2. Оценка координат потребителя с учетом ошибок SISRE

SISRE - эквивалентная погрешность псевдодальности за счёт космического сегмента (signal-in-space range error). Эта величина характеризует погрешность измерения дальности до космического аппарата, обусловленную влиянием так называемых «системных» факторов, не зависящих от внешней среды и характеристик оборудования потребителей.

Для системы ГЛОНАСС эту величину можно наблюдать графически (рис.1.1) на официальном сайте [3].

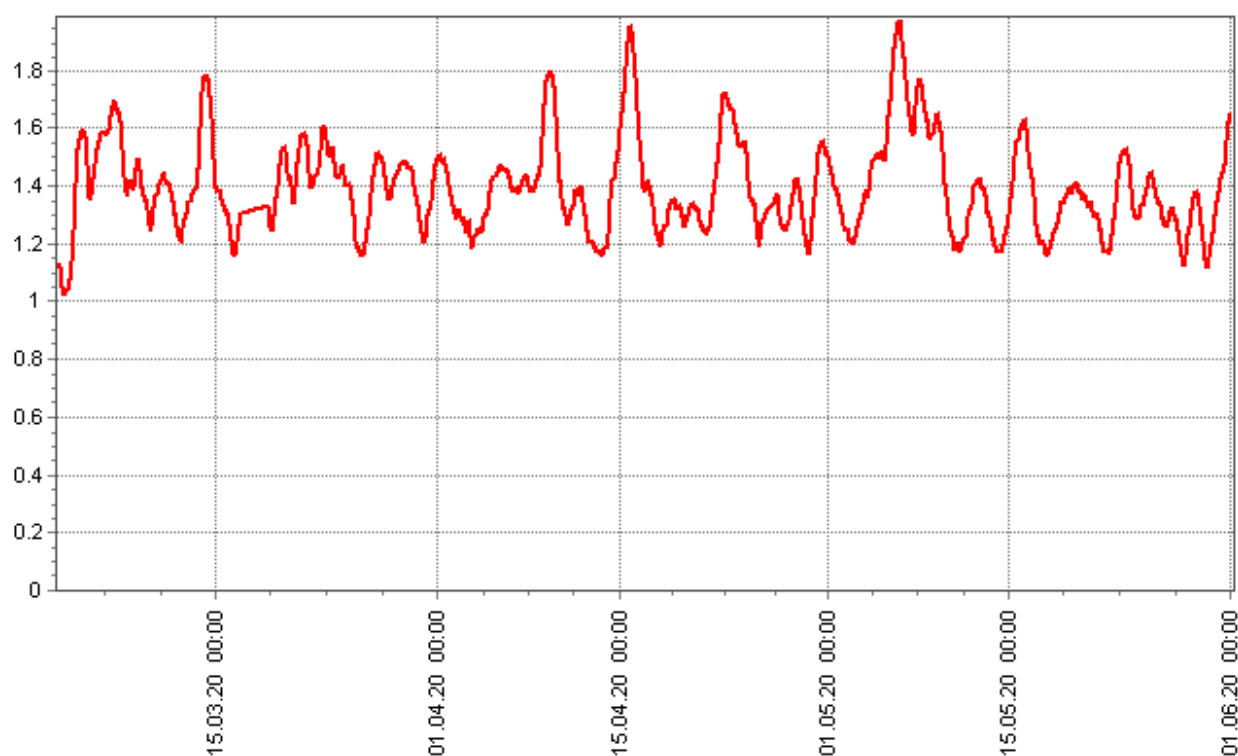


Рис.1.1. Суточные СКП SISRE ГЛОНАСС, м

Оценка функциональных характеристик ГЛОНАСС и GPS проводится в ИАЦ КВНО в режиме регулярной службы на основе измерений глобальной сети станций с помощью собственных программных средств.

Так же данные SISRE можно получить на сервере информационно-аналитического центра координатно-временного и навигационного обеспечения [4].

1.2.1. Постановка задачи нахождения оценки координаты потребителя с учетом ошибок SISRE

Постановка задачи: Оценить координаты потребителя $\{x, y, z\}$ с учетом ошибок SISRE

Введем вектор состояния $\mathbf{x} = \begin{bmatrix} x & y & z & \mathcal{D} \end{bmatrix}^T$, вектор вторичных наблюдений

$$\mathbf{y}_{\tilde{\mathbf{R}},k} = \begin{bmatrix} y_{\tilde{R}_1,k} & y_{\tilde{R}_2,k} & \dots & y_{\tilde{R}_N,k} \end{bmatrix}^T \quad \text{и} \quad \text{вектор погрешностей}$$

$$\mathbf{n}_{\tilde{\mathbf{R}},k} = \begin{bmatrix} n_{\tilde{R}_1,k} & n_{\tilde{R}_2,k} & \dots & n_{\tilde{R}_N,k} \end{bmatrix}^T$$

Линеаризуем вторичные наблюдения $y_{\tilde{R},k} = R_i(x_k, y_k, z_k) + \mathcal{D}' + n_{\tilde{R},k}$ относительно некоторой априорной оценки $\tilde{\mathbf{x}}_k = \begin{bmatrix} \tilde{x}_k & \tilde{y}_k & \tilde{z}_k & \tilde{\mathcal{D}}_k \end{bmatrix}^T$, тогда:

$$y_{\tilde{R},k} = h_{i,k}(\tilde{\mathbf{x}}_k) + \tilde{H}(\tilde{\mathbf{x}}_k)(\mathbf{x}_k - \tilde{\mathbf{x}}_k) + \mathbf{n}_{\tilde{\mathbf{R}},k} \quad (1.2)$$

Выполним преобразования [1, 204 стр.] тогда получим выражение:

$$\Delta \hat{\mathbf{x}}_k = (\tilde{\mathbf{H}}^T \mathbf{D}_{\mathbf{n}_R}^{-1} \tilde{\mathbf{H}})^{-1} \tilde{\mathbf{H}}^T \mathbf{D}_{\mathbf{n}_R}^{-1} \Delta \mathbf{y}_{\tilde{\mathbf{R}},k} \quad (1.3)$$

Где $\mathbf{D}_{\mathbf{n}_R} = M[\mathbf{n}_{\tilde{\mathbf{R}},k} \mathbf{n}_{\tilde{\mathbf{R}},k}^T]$ введенная матрица дисперсий погрешностей вторичных наблюдений.

На данный этап приближенно будем считать $\sigma_n^2 = SISRE_n^2$

Тогда преобразовав матрицу дисперсий и рассмотрим для случая при $k=1$, она примет вид:

$$\mathbf{D}_{\mathbf{n}_R} = M[\mathbf{n}_{\tilde{\mathbf{R}},1} \mathbf{n}_{\tilde{\mathbf{R}},1}^T] = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \sigma_n^2 \end{bmatrix}$$

Необходимо найти матрицу дисперсий вектора состояния \mathbf{x} , которая по определению равна $\mathbf{D}_x = M[\Delta\hat{\mathbf{x}} \cdot (\Delta\hat{\mathbf{x}})^T]$, подставим формулу (1.3), получим:

$$\mathbf{D}_x = M[\Delta\hat{\mathbf{x}} \cdot (\Delta\hat{\mathbf{x}})^T] = (\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \Delta \mathbf{y}_{\tilde{R}} \cdot \left((\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \right)^T \tilde{\mathbf{H}} \left(\mathbf{D}_{n_R}^{-1} \right)^T \Delta \mathbf{y}_{\tilde{R}}^T$$

Учитывая, что $\mathbf{D}_{n_R} = M[\mathbf{n}_{\tilde{R}} \mathbf{n}_{\tilde{R}}^T] = M[\Delta \mathbf{y}_{\tilde{R}} \cdot (\Delta \mathbf{y}_{\tilde{R}})^T]$, тогда матрица дисперсий вектора состояния \mathbf{x} равна:

$$\begin{aligned} \mathbf{D}_x &= (\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \cdot \left((\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \right)^T \tilde{\mathbf{H}} \left(\mathbf{D}_{n_R}^{-1} \right)^T \cdot \mathbf{D}_{n_R} = \\ &= \left((\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \right)^T \left(\mathbf{D}_{n_R}^{-1} \right)^T \cdot \mathbf{D}_{n_R} = \left((\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \right)^T \end{aligned}$$

Получаем, что на диагонали \mathbf{D}_x будут лежать дисперсии компонент вектора состояний \mathbf{x}

1.2.2. Решения задачи нахождения оценки координаты потребителя с учетом ошибок SISRE

Из пункта 1.2.1 следует, что решением данной задачи является матрица \mathbf{D}_x на диагонали которой будут лежать дисперсии компонент вектора состояний \mathbf{x} .

И матрица принимает вид: $\mathbf{D}_x = \left((\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \right)^T$

Для решение данной задачи необходимо найти матрицы \mathbf{H} и \mathbf{D}_n :

- Матрица \mathbf{H} принимает вид:

$$\mathbf{H}_{n,1...4} = \left| \begin{array}{ccc} \frac{(x_n - x)}{R_n} & \frac{(y_n - y)}{R_n} & \frac{(z_n - z)}{R_n} \\ 1 & 1 & 1 \end{array} \right| \quad (1.4)$$

где:

$$R_n = \sqrt{(x_n - x)^2 + (y_n - y)^2 + (z_n - z)^2} \quad (1.5)$$

- Матрица \mathbf{D}_n принимает вид:

$$\mathbf{D}_n = \begin{vmatrix} D_1 & 0 & 0 & 0 \\ 0 & D_2 & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & D_n \end{vmatrix} \quad (1.6)$$

где n зависит от количества видимых спутников

После полученных значений матрица \mathbf{D}_x примет вид:

$$\mathbf{D}_x = \left((\tilde{\mathbf{H}}^T \mathbf{D}_{nR}^{-1} \tilde{\mathbf{H}})^{-1} \right)^T = \begin{vmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \sigma_{34} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_{44} \end{vmatrix} \quad (1.7)$$

где:

$$\begin{aligned} \sigma_{11}^{-1} &= \frac{H_{1,1}^2}{D_1} + \frac{H_{2,1}^2}{D_2} + \dots + \frac{H_{n,1}^2}{D_n} \\ \sigma_{22}^{-1} &= \frac{H_{1,2}^2}{D_1} + \frac{H_{2,2}^2}{D_2} + \dots + \frac{H_{n,2}^2}{D_n} \\ \sigma_{33}^{-1} &= \frac{H_{1,3}^2}{D_1} + \frac{H_{2,3}^2}{D_2} + \dots + \frac{H_{n,3}^2}{D_n} \\ \sigma_{44}^{-1} &= \frac{H_{1,4}^2}{D_1} + \frac{H_{2,4}^2}{D_2} + \dots + \frac{H_{n,4}^2}{D_n} \end{aligned}$$

Развернутое вычисление матрицы \mathbf{D}_x в приложении 2.

На диагонали лежат дисперсии компонент вектора состояний \mathbf{x} .

1.3. Описание движения спутников ГЛОНАСС по орбитам

Координаты центра спутника (x_i, y_i, z_i) изменяются в зависимости от положения спутника на орбите

Координаты орбит:

$$\begin{aligned}x &= r \cdot [\cos(\vartheta + \omega_n) \cdot \cos \Omega - \sin(\vartheta + \omega_n) \cdot \sin \Omega \cdot \cos i]; \\y &= r \cdot [\cos(\vartheta + \omega_n) \cdot \sin \Omega + \sin(\vartheta + \omega_n) \cdot \cos \Omega \cdot \cos i]; \\z &= r \cdot \sin(\vartheta + \omega_n) \cdot \sin i.\end{aligned}\tag{1.1}$$

Где:

$\omega_n = 1.5514 \cdot 10^{-4}$ - угол перигея,

ϑ - истинное значение аномалии

Ω - долгота восходящего узла

Значение Ω и ϑ берутся из приложении 1.

Уравнение орбиты спутника в орбитальной плоскости в полярной системе координат с центром, совпадающим с центром Земли:

$$r = \frac{p}{1 + e \cdot \cos(\vartheta - \vartheta_0)},\tag{1.2}$$

где p - фокальный параметр;

e - эксцентриситет;

ϑ_0 - угол между положительным направлением полярной оси и фокальной осью (при $\vartheta_0 = 0$ полярная ось направлена от центра к ближайшей вершине кривой, а при $\vartheta_0 = \pi$ - в противоположную сторону, в дальнейшем расчете полагаем, что $\vartheta_0 = 0$);

i - угол наклона орбиты.

1.4. Расчет выбранной точки в декартовой системе координат

По заданию, необходимо определить координаты выбранной точки:
(55.819715, 37.611680) - координаты наблюдателя.

Переведем координаты точки из геодезической системы координат (Н, L, В) в декартовую систему координат (X, Y, Z), для этого используем формулы:

$$X_0 = (N + H) \cdot \cos B \cdot \cos L,$$

$$Y_0 = (N + H) \cdot \cos B \cdot \sin L,$$

$$Z_0 = ((1 - e^2) \cdot N + H) \cdot \sin B.$$

Где $N = \frac{a}{\sqrt{1 - e^2 \cdot (\sin B)^2}}$ - радиус кривизны в точке местной вертикали;

a – параметр сжатия эллипсоида;

H= 200 м - высота над уровнем моря потребителя;

L=37.611680 - долгота;

B=55.819715 – широта.

Тогда подставив значения координат наблюдателя в формулы перерасчета в декартовую систему координат, значения (X, Y, Z) примут вид:
(2838598.372491, 2186937.017082, 5276631.111979)

1.5. Расчет видимости спутников

На рисунке 1 представлен угол (α) между нормалью к потребителю и спутником, зная который, можно определить угол места ($90^\circ - \alpha$), по которому определим наличия видимости навигационных спутников.

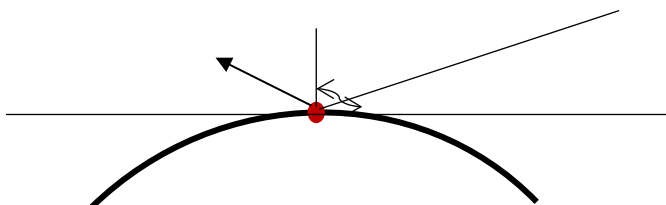


Рисунок 1 – Угол между нормалью к цели и спутником

Исходя из наших данных, для расчета видимости спутников введем вектор единичной длины в направление на НКА, тогда он будет равен:

$$R = \frac{\overrightarrow{ES}}{|\overrightarrow{ES}|} \quad (3.1)$$

Где $\overrightarrow{ES} = (X_i - X_0, Y_i - Y_0, Z_i - Z_0)$.

Так как данные координат даны в геодезической системе координат для расчета необходим перевести в топоцентрические координаты (ENU) [Переведем координаты точки из геодезической системы координат (H, L, B) в декартовую систему координат (X, Y, Z)] – и коэффициентом для перевода из геодезической в топоцентрическую систему координат будет являться матрица C :

$$C = \begin{vmatrix} -\sin L & \cos L & 0 \\ -\sin B \cdot \cos L & -\sin B \cdot \sin L & \cos B \\ \cos B \cdot \cos L & \cos B \cdot \sin L & \sin B \end{vmatrix} \quad (3.2)$$

Теперь умножим вектор единичной длины в направление на НКА, который является вектором столбцом с размером 3×1 , на получившийся

коэффициент перевода, то есть матрицу C с размером 3×3 , и в ответе получим вектор столбец R_{enu} с размером 3×1 :

$$R_{enu} = C \cdot R = \begin{bmatrix} R_e & R_n & R_u \end{bmatrix} \quad (3.3)$$

Тогда угол между нормалью к потребителю и спутником определяется по R_u или z составляющей вектора R_{enu} (которая рассчитывается по формуле 3.3):

$$\alpha = \arccos(R_u) \quad (3.4)$$

Если $(90 - \alpha) < 5^\circ$, то спутник находится в области видимости.

ГЛАВА 2. РАБОТА С СИСТЕМОЙ «GIT»

GitHub — это самый популярный сервис для проектов, использующих систему Git, которая распространена по всему миру, находится по адресу: «github.com»

Git — распределённая система контроля версий, которая даёт возможность разработчикам отслеживать изменения в файлах и работать совместно с другими разработчиками.

Это дает нам возможность более быстрого отслеживания изменения и обменом программы, а также контролирования появления и исправления новых ошибок.

2.1. Создание репозитория

Для этого создадим аккаунт на платформе “GitHub” и заведем в нем собственный репозиторий (место где хранится проект).

Для заведения репозитория программы скачаем «GitHub desktop» - это интерфейсная программа, которая упрощает рабочий процесс разработки.

После установки программы «GitHub desktop» откроем ее и зайдем в ранее зарегистрированный аккаунт:

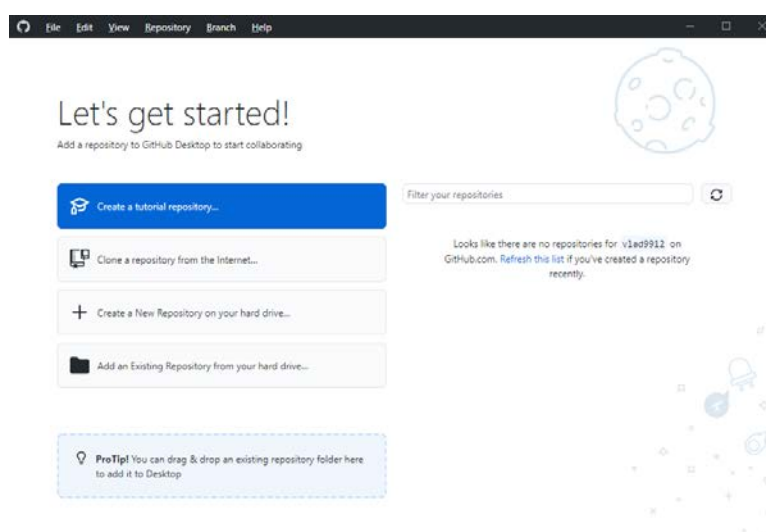


Рисунок 2.1 – Интерфейс программы «GitHub desktop»

Нажмем “Create a New Repository on your hard drive...”, после чего появится окно создания репозитория:

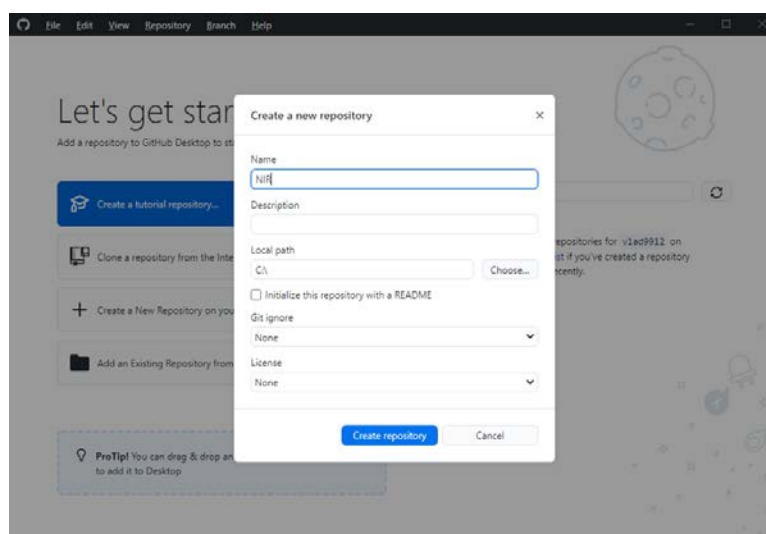


Рисунок 2.2 – Создания репозитория в «GitHub desktop»

Назовем репозиторий – «NIR» и выберем папку, в которой он будет находиться, в моем случае это диск – «С».

Таким образом создастся папка «NIR» по корневому адресу «C:\NIR», а также на сайте «GitHub». Теперь перенесем все необходимые файлы проекта в папку «NIR» на компьютере.

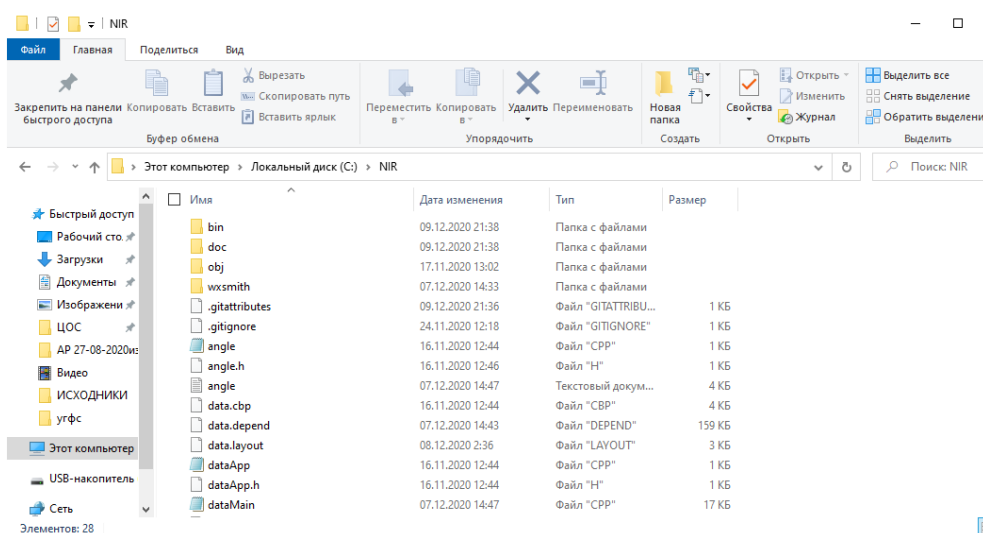


Рисунок 2.3 – Перемещение проекта в папку репозитория

Далее зайдем обратно в программу «GitHub» и нам предложат синхронизировать версию папки NIR на компьютере, с версией NIR на сайте GitHub.

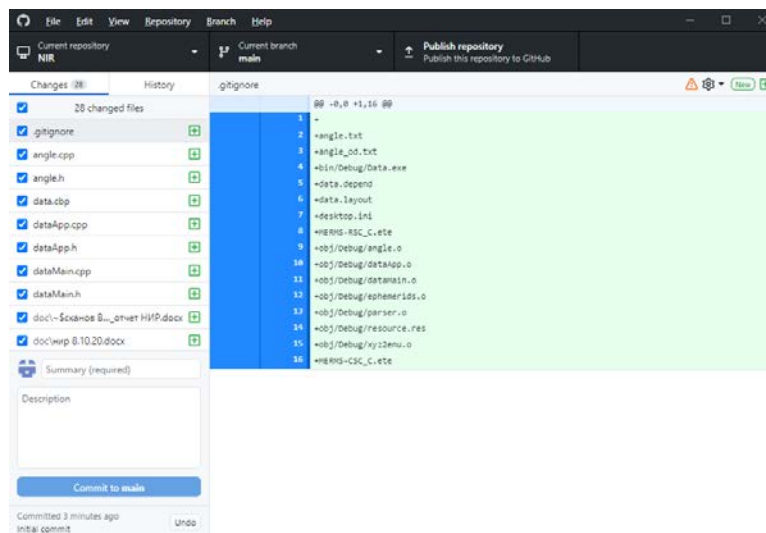


Рисунок 2.4 – Интерфейс программы «GitHub desktop»

В разделе Summary напишем изменения, которых произошли – «добавил нир в репозиторий на гитхаб»

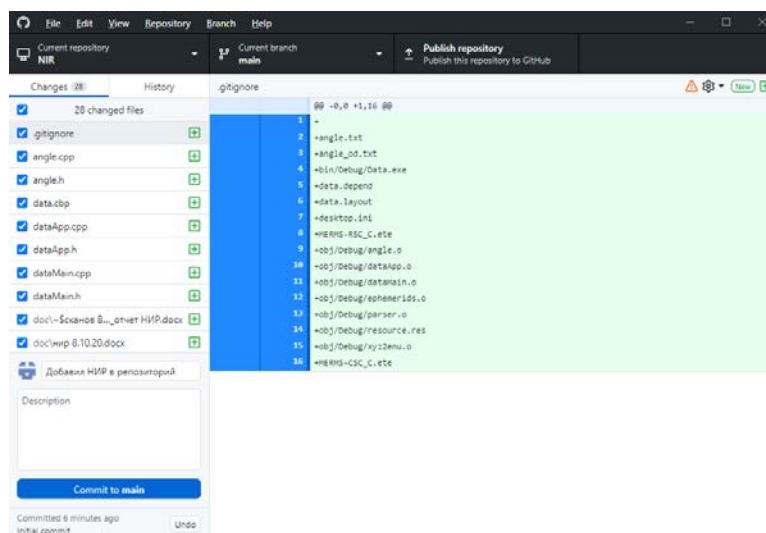


Рисунок 2.5 – Интерфейс программы «GitHub desktop»

Далее нажимаем «commit to main» и кнопку «publish repository»:

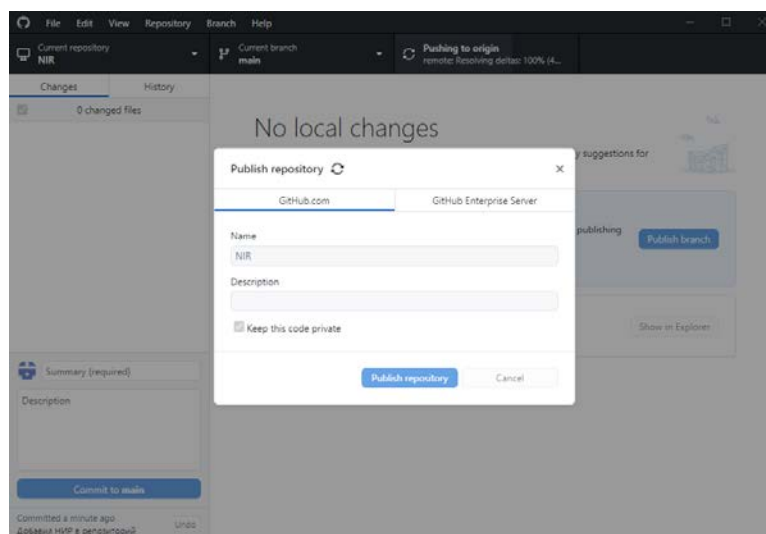


Рисунок 2.6 – Интерфейс программы «GitHub desktop»

Теперь зайдём на сайт «GitHub» и можем пронаблюдать, что содержимое папки появилась на сайте:

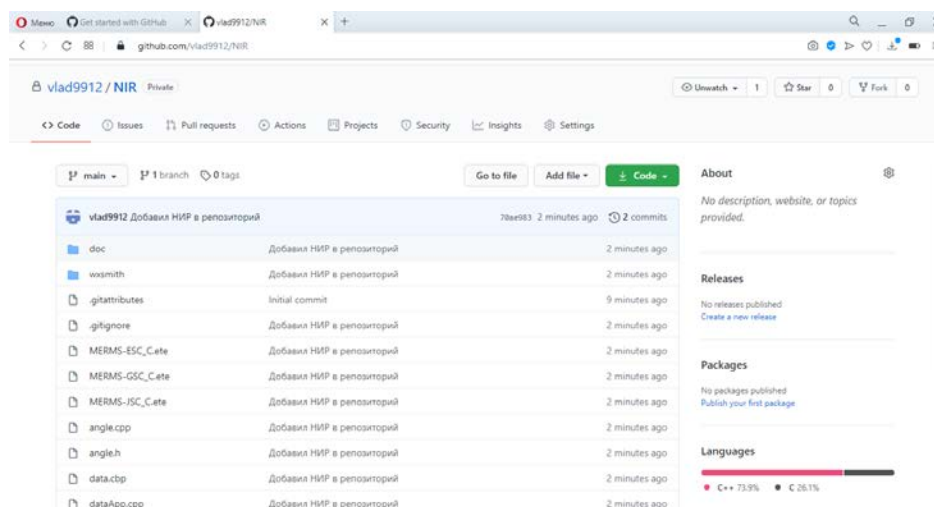


Рисунок 2.7 – Проверка репозитория на сайте «GitHub»

2.2. Проверка правильности работы системы «Git»

Далее проверим работу: очистим код под функцией `ephemerids` в файле `ephemerids`, а в файле «`dataMain`» в 128 строке пропишем функцию вывода «hello world»: проверим работу программы:

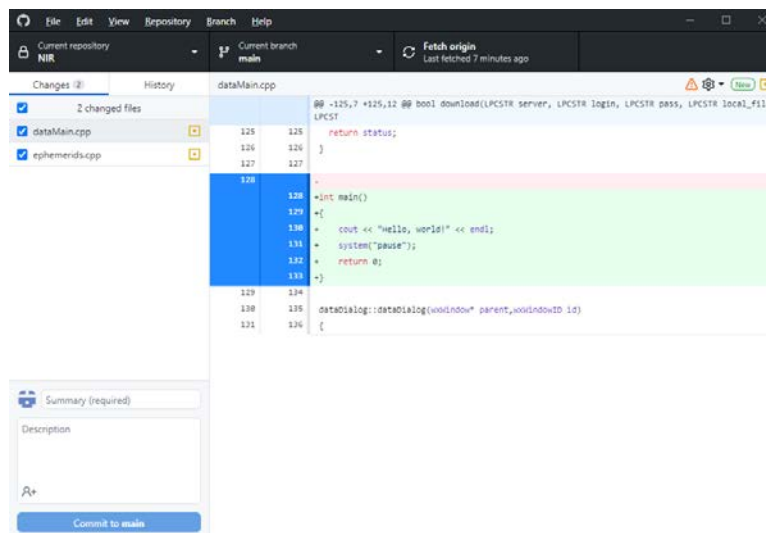


Рисунок 2.8 – Интерфейс программы «GitHub desktop»

Мы можем наблюдать изменения, которые совершили в коде, далее синхронизируем репозитории и зйдем на сайт:

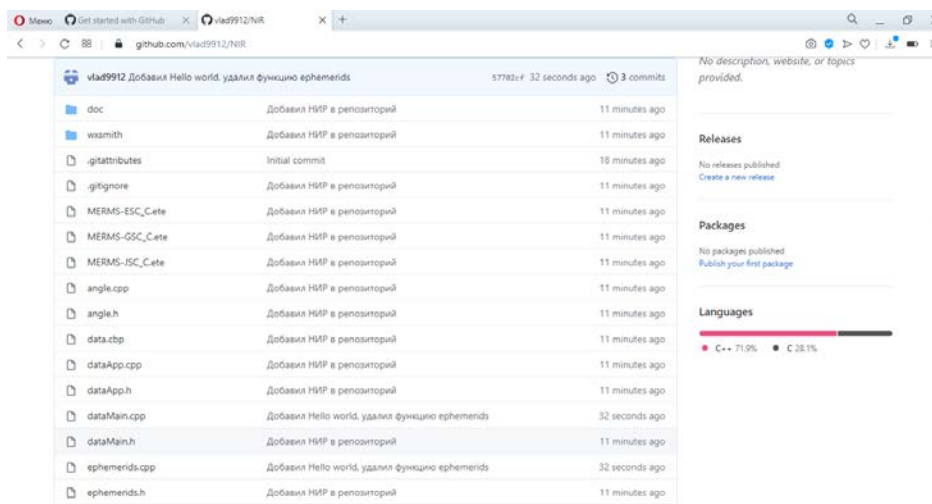


Рисунок 2.9 – Проверка репозитория на сайте «GitHub»

Пройдем по «коммиту» и посмотрим отображаются ли изменения:

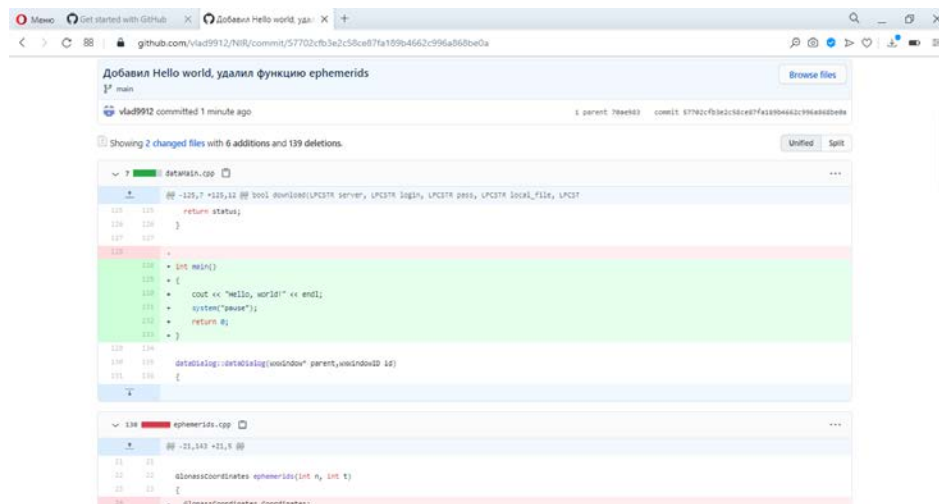


Рисунок 2.10 – Проверка репозитория на сайте «GitHub»

Если вы хотите вернуть предыдущую версию, достаточно пройти во вкладку History в программе «GitHub», найти версию изменения и нажать кнопку Fetch origin

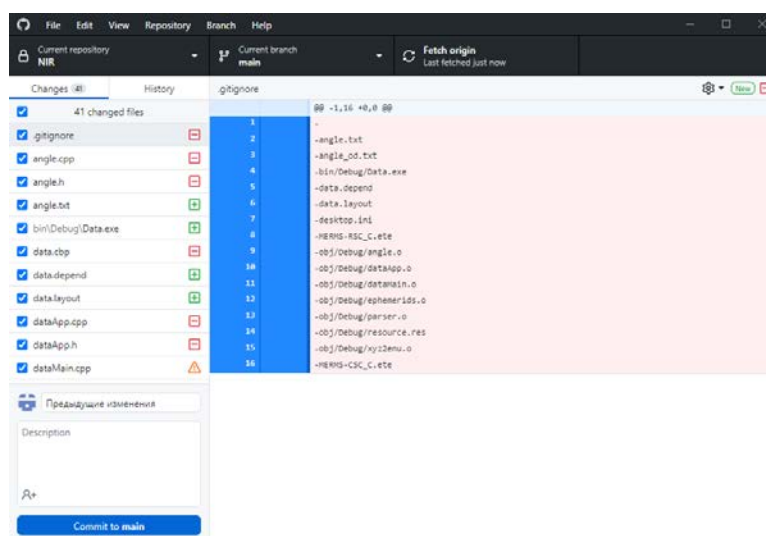


Рисунок 2.11 – Интерфейс программы «GitHub desktop»

2.3. Добавления в коллаборацию

Далее сделаем репозиторий приватным и пригласим для просмотра и исправлений научного руководителя, для этого пройдем во вкладку «setting» далее ->«manage access», нажимаем на кнопку «invite a collaborator»

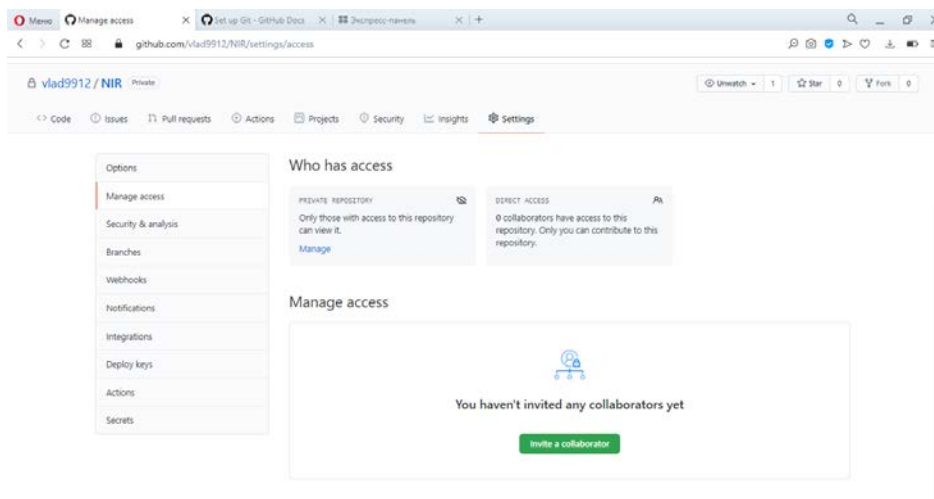


Рисунок 2.12 – Добавление в коллаборацию

Далее необходимо ввести профиль научного руководителя и нажимаем кнопку добавить в этот репозиторий:

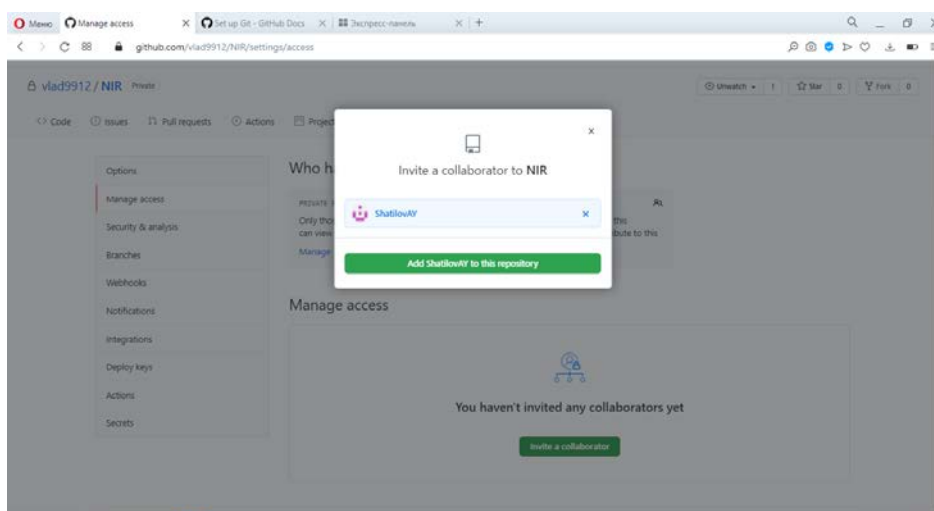


Рисунок 2.13 – Добавление в коллаборацию

Далее преподавателю придет письмо на почту с приглашением просмотра репозитория и при успешном добавлении будет присутствовать в «Direct Access» :

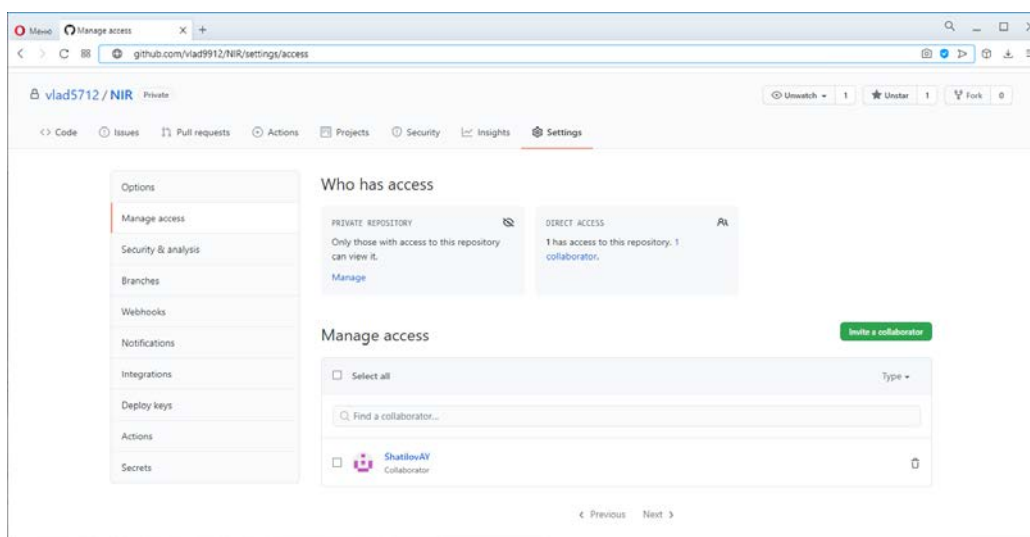


Рисунок 2.14 – Добавление в коллаборацию

Все, система «Git» в «Github» успешна настроена.

ГЛАВА 3. РАЗРАБОТКА ПРОГРАММЫ

Для разработки программы выбрали среду – CodeBlocks.

CodeBlocks - свободная интегрированная среда разработки (IDE) с открытым исходным кодом, которая поддерживает использование различных компиляторов. В качестве компилятора используется MinGW.

MinGW (сокращение "Minimalist GNU для Windows") - набор инструментов разработки программного обеспечения для создания приложений под Windows. Набор MinGW включает инструменты для программирования с открытым исходным кодом, которые не зависят от каких-либо платных сторонних библиотек. Он используется многими средами разработки (Visual Studio использует собственный компилятор). MinGW включает набор GNU Compiler Collection (GCC), включая компиляторы C, C++, ADA и Fortran. Code::Blocks имеет открытую архитектуру, что позволяет ему масштабироваться за счёт подключаемых модулей (можно писать программы как под windows, linux, так и под различные микропроцессоры типа avr и stm32). Отладчик кода в Code::Blocks поддерживает точки останова в исходном коде или в данных, которые обрабатывает программа.

Создание графики происходит с помощью графического интерфейса – wxWidgets.

wxWidgets - это библиотека графических инструментов (по-английски Widget toolkit). Она необходима для того, чтобы в среде CodeBlocks можно было создавать программы с графическим интерфейсом.

3.1. Необходимые файлы для сборки проекта

К отчету прикреплены 10 приложений, в которых содержатся основные файлы кода программы, необходимые для сборки проекта. Так файл: “dataMain.cpp”, код описан в приложение 3, содержит в себе основной алгоритм настройки окна приложения, с помощью библиотек wxWidgets, также содержит алгоритм скачивания файла с сервера и заполнения таблицы данными SISRE и SISVE, путь к файлу: “...\data\dataMain.cpp”.

Файл: “parser.c”, код описан в приложение 4, содержит в себе обработку скаченного файла с сервера, с помощью алгоритма из приложения 1. Под обработкой подразумевается фильтрование нужной нам информации – значения SISRE и SISVE для определенного спутника, путь к файлу: “...\data\parser.c”.

Файл: “parser.h”, код описан в приложение 5, содержит в себе обработчик массива SISerr, для использования этого массива в приложение 1, данный обработчик необходим, так как приложение 2 написано на языке «С», а приложение 1 на языке «C++», путь к файлу: “...\data\parser.h”.

Файл: “data.cbp”, код описан в приложение 6 – это необходимый файл для сборки проекта, в котором прописан используемый компилятор, библиотеки, а также все необходимые заголовочные файлы, путь к файлу: “...\data\data.cbp”.

Файл: “dataMain.h”, код описан в приложение 7 – это заголовочный файл, в котором хранятся применяемые классы и методы, путь к файлу: “...\data\dataMain.h”.

Файл “xyz2enu.cpp”, код описан в приложение 8, в файле реализована функция перевода из геодезической системы координат в топоцентрические координаты (ENU), путь к файлу: “...\data \ xyz2enu.cpp”

Файл “xyz2enu.h”, код описан в приложение 9 - это заголовочный файл, в котором объявляются применяемые классы и методы для функции перевода из геодезической системы координат в топоцентрические координаты (ENU) : “...\data \ xyz2enu.h”

Файл “ephemerids.cpp”, код описан в приложение 10, в файле реализована функция описание движения спутников ГЛОНАСС по орбитам и нахождения координат спутников ГЛОНАСС в определенный момент, путь к файлу: “...\data \ ephemerids.cpp ”

Файл “ ephemerids.h”, код описан в приложение 11 - это заголовочный файл, в котором объявляются применяемые классы и методы для функции “ephemerids”, путь к файлу: “...\data \ ephemerids.h ”

Файл “angle.cpp”, код описан в приложение 12, в файле реализована функция расчета видимости спутников, путь к файлу: “...\data \ angle.cpp ”

Файл “ angle.h ”, код описан в приложение 13 - это заголовочный файл, в котором объявляются применяемые классы и методы расчета видимости спутников, путь к файлу: “...\data \ angle.h ”

Файл: “datadiaslog.wxs”, код описан в приложение 14 – это файл описания графического пользовательского интерфейса для плагина wxSmith, путь к файлу: “...\data\ wxsmith\datadiaslog.wxs”.

3.2.Изменения по коду

Полный алгоритм программы описан в отчете НИР за 8 семестр, в данном отчете рассмотрим лишь ключевые изменения, внесенные в алгоритм.

В программе необходимо было добавить алгоритм описание движения спутников ГЛОНАСС по орбитам и нахождения координат спутника в определённый момент, алгоритм расчета видимости спутников, а также нахождения оценки координаты потребителя с учетом ошибок SISRE.

3.2.1. Алгоритм описания движения спутников ГЛОНАСС по орбитам и нахождения координат спутника в определённый момент

Функция примет тип структуры в связи с необходимостью получить координаты спутников – X,Y,Z.

Структура — это объединение нескольких объектов, возможно, различного типа под одним именем, которое является типом структуры. В качестве объектов могут выступать переменные, массивы, указатели и другие структуры. То есть структура позволяет сгруппировать переменные разных типов в единое целое.

Запишем данные истинного значения аномалии и долготы восходящего узла для каждого спутника, применив функцию «switch».

«switch» - оператор выбора, является очень удобной заменой множественного использования операторов «if», оператор «switch» сравнивает значение одной переменной с несколькими константами. Основной формат для использования оператора множественного выбора «switch case». Значение переменной указанной в условии «switch» сравнивается со значениями, которые следуют за ключевым словом «case». Когда значение в переменной, соответствует значению в строке с оператором «case», компьютер продолжит выполнение программы с этого места.

Когда сравниваемое значение в переменной «variable» совпадет с первым значением оператора «case», программа начнет выполнять код, который находится между текущим оператором «case» и оператором «break». Оператор «break» используется для того, чтобы прерывать ход программы в операторе «switch» и передавать управление следующему оператору, после «switch». Если не использовать оператор «break», то, сразу после того, как выполнится один блок кода, программа переключится на выполнения следующего «case», даже, если константное значение не будет равно значению

в переменной «variable». Поэтому, в операторе выбора «switch», блоки кода после «case» всегда должны предваряться оператором «break».

Далее основной расчет происходит по формулам из пункта 1.3.

3.2.2. Алгоритм перевода из геодезической системы координат в топоцентрические координаты

В данную функцию поступают значения в геодезической системе координат и указатель матрицы **C** - массив данных, который равен 0.

Параметры функции в языке «C++» могут представлять указатели. Указатели передаются в функцию по значению, то есть функция получает копию указателя. В то же время копия указателя будет в качестве значения иметь тот же адрес, что оригинальный указатель. Поэтому используя в качестве параметров указатели, мы можем получить доступ к значению аргумента и изменить его.

Далее в функции по формулам преобразования получается массив, в котором содержатся коэффициенты матрицы **C** из пункта 1.5, которые передаются обратно по указателю в переменную.

3.2.3. Алгоритм расчета угла

Функции «angel» примет тип данных “double”. Так как значения, которые обрабатывает и выдает функция нецелые.

Функцию поступают значения координат спутников и координаты потребителя, далее по формулам из пункта 1.5 рассчитываем вектор единичной длины в направлении на НКА.

Теперь необходимо эти значения умножить на получившийся коэффициент перевода (из пункта 3.2.2), то есть матрицу **C** с размером 3×3 , и получим вектор-матрицу, где угол определяется по z – ой компоненте.

На выходе функции получается угол между нормалью к потребителю и спутником

3.2.4. Нахождения оценки координаты потребителя с учетом ошибок SISRE

Данный алгоритм напишем в основной файл программы – «dataMain.cpp». Он будет срабатывать по нажатию кнопки «Обработка»

Ввод данных координат потребителя, а также его высоты над уровнем моря, будет реализован с помощью форм «wxTextCtrl».

Данный алгоритм создан только для системы ГЛОНАСС, поэтому во избежание появления ошибок, в самом начале функции пропишем условие, для срабатывания алгоритма только для системы ГЛОНАСС, а при выборе других ГНСС будет выводиться окно с ошибкой.

Для этого воспользуемся функций «if», в которой приравняем указатель текста в форме «wxChoice» к ГЛОНАСС. Если это условия удовлетворяет, то будет происходить дальнейший расчет, иначе будет выводить окно с ошибкой, для этого пропишем в операторе «else» функцию «wxMessageBox», которая будет выдавать определенный текст в новом окне.

Далее необходимо заполнить матрицу дисперсий погрешностей вторичных наблюдений.

Так как дисперсии скачиваются с сервера с помощью функции “parser” и не все спутники находятся в определённый момент для потребителя в области видимости, следовательно необходимо создать алгоритм нахождения спутника в области видимости.

Для этого получив значения углов между нормалью к потребителю и спутником из функции “angle”, определим видимость спутника, для этого воспользуемся функцией «if», в котором создадим условия, если угол больше 5 градусов, то он находится в области видимости.

Пример реализации:

```
if ((alpha) > 5)
```

```

{
    vsb[i]=1;
    Visibles.push_back(i); // добавление элемента в конец вектора
}

```

Где функция «Visibles.push_back(i)» добавление элемента в конец вектора, то есть таким образом рассчитаем суммарное количество видимых спутников.

Для упрощения, если данных дисперсий для определённого спутника отсутствуют, принимаем наибольшее значение из имеющих дисперсий погрешностей.

Для работы с матрицами воспользуемся библиотекой «Armadillo» (пункт 3.3).

Объявим матрицу \mathbf{D}_n : “mat Dn;”

Чтобы не появились аномальные числа выделим необходимый нам массив данных и приравняем его к нулю: “Dn.zeros(inum, inum);”

Матрица \mathbf{D}_n приняла вид:

$$\begin{bmatrix} 0 & \dots & 0_{0,inum} \\ \dots & \dots & \dots \\ 0_{inum,0} & \dots & 0_{inum,inum} \end{bmatrix}$$

Заполним диагональ матрицы значениями: $\sigma_n^2 = SISRE_n^2$ (из п. 1.2).

Объявим матрицу \mathbf{H} : “mat H(inum, 4); H.zeros();” Таким образом создается нулевая матрица размером (inum, 4).

Заполним ее значениями из (п.1.2)

Далее посчитаем матрицу СКО по формуле из (п.1.2.1):

mat sko = (inv(H.t()*inv(Dn)*H)).t();, где функция .t() – транспонирования матрицы.

Выведем полученные значения в форму «wxTextCtrl» с помощью встроенного оператора вывода “ostream”.

Для округления до 3 числа после запятой воспользуемся преобразователем числа в строку, с помощью “wxString”

Например:

```
streamX << wxString::Format("%.3f", sko(0,0)) ;  
streamY << wxString::Format("%.3f", sko(1,1)) ;  
streamZ << wxString::Format("%.3f", sko(2,2)) ;  
streamD << wxString::Format("%.3f", sko(3,3)) ;
```

Получим в 4 формах СКО по X координате, Y координате, Z координате и погрешности временной шкалы.

3.3. Библиотека “Armadillo”

«Armadillo» - это программная библиотека линейной алгебры с открытым исходным кодом для языка программирования «C++» . Он нацелен на обеспечение эффективных и оптимизированных базовых вычислений. Синтаксис намеренно похож на «Matlab».

Поддерживаются целые числа, числа с плавающей точкой и комплексные числа, а также подмножество тригонометрических и статистических функций. Различные разложения матриц обеспечиваются за счет необязательной интеграции с «LAPACK» или одной из его высокопроизводительных замен (например, многопоточных библиотек «MKL» или «ACML»).

Подход с отложенной оценкой используется (во время компиляции) для объединения нескольких операций в одну и уменьшения (или устранения) потребности во временных. Это достигается с помощью рекурсивных шаблонов и шаблонного метапрограммирования.[5].

ГЛАВА 3. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Минимальные требования:

Для запуска программы необходимо иметь windows 7/10 и подключенное устройство к интернету.

Инструкция:

1. Запустите программу “Data.exe” от имени администратора

Если программа не запустится отключите антивирус.

У вас появится диалоговое окно:

The screenshot shows a window titled "Data app" with a close button (X) in the top right corner. Inside the window, there is a dropdown menu currently showing "Glonass". To the right of the dropdown is a button labeled "Загрузить". Below the dropdown is a table with 24 rows and 2 columns. The first column contains numbers from 1 to 24. The second column is labeled "SISRE, m" and the third column is labeled "SISVE, mm/s". To the right of the table, there are four input fields with labels: "Введите значение высоты:", "Введите значение B:", "Введите значение L:", and "Введите значение D:". Below these input fields is a button labeled "Обработка".

	SISRE, m	SISVE, mm/s
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		

Рисунок 3.1. Интерфейс программы

2. Выберите необходимую вам НС:

The screenshot shows a software window titled "Data app". On the left, a dropdown menu is open, displaying a list of satellite systems: Glonass, GPS, Galileo, Beidou, and QZSS. Below the dropdown is a table with 24 rows, numbered 1 to 24. To the right of the table, there are several input fields and buttons. At the top right is a "Загрузить" (Load) button. Below it is a label "Введите значение высоты:" followed by an input field. Then "Введите значение B:" followed by an input field. Then "Введите значение L:" followed by an input field. Below these is an "Обработка" (Processing) button. At the bottom right, there are four more input fields labeled "СКО для x:", "СКО для y:", "СКО для z:", and "СКО для D:", each followed by an input field.

Рисунок 3.2. Интерфейс выбора НС

3. Нажмите кнопку «Загрузить»

Data app

Glonass

Загрузить

	SISRE, m	SISVE, mm/s
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		

Введите значение высоты:

Введите значение B:

Введите значение L:

Обработка

СКО для x:

СКО для y:

СКО для z:

СКО для D:

Рисунок 3.3. Интерфейс выбранной НС

Получили значения SISRE и SISVE для каждого спутника, если значения равны 0.00, то данный спутник отсутствует.

Если при загрузке возникла ошибка существует два варианта решения ее:

- Отключите антивирус,
- Включите брандмауэр.

4. Для того, чтобы скачать данные для других НС, перейдите к п. 2 инструкции.

5. Чтобы закрыть программу нажмите крестик в диалоговом окне

The screenshot shows a window titled "Data app" with a close button (X) in the top right corner. Inside the window, there is a dropdown menu currently set to "Glonass". To the right of the dropdown is a button labeled "Загрузить". Below these is a table with 24 rows and 3 columns. The first column contains numbers 1 through 24. The second column is labeled "SISRE, m" and the third is labeled "SISVE, mm/s". To the right of the table are several input fields and buttons. There is a label "Введите значение высоты:" followed by an input field. Below that is a label "Введите значение B:" followed by an input field. Then a label "Введите значение L:" followed by an input field. Below these is a button labeled "Обработка". Further down are labels "СКО для x:", "СКО для y:", "СКО для z:", and "СКО для D:", each followed by an input field.

	SISRE, m	SISVE, mm/s
1	0.427	0.508
2	1.205	0.650
3	0.863	0.511
4	0.851	0.569
5	0.682	0.597
6	0.000	0.000
7	0.686	0.469
8	2.104	0.506
9	2.751	1.216
10	0.000	0.000
11	0.000	0.000
12	1.390	0.753
13	1.451	0.657
14	0.914	0.540
15	0.519	0.562
16	1.392	0.580
17	0.440	0.701
18	1.481	0.551
19	1.023	0.591
20	1.190	0.768
21	0.849	0.636
22	1.324	0.796
23	0.000	0.000
24	0.690	0.534

Рисунок 3.4. Полученные значения

6. Далее вводим значения из пункта 1.4:

Data app ✕

Glomass ▼ Загрузить

	SISRE, m	SISVE, mm/s
1	0.427	0.508
2	1.205	0.650
3	0.863	0.511
4	0.851	0.569
5	0.682	0.597
6	0.000	0.000
7	0.686	0.469
8	2.104	0.506
9	2.751	1.216
10	0.000	0.000
11	0.000	0.000
12	1.390	0.753
13	1.451	0.657
14	0.914	0.540
15	0.519	0.562
16	1.392	0.580
17	0.440	0.701
18	1.481	0.551
19	1.023	0.591
20	1.190	0.768
21	0.849	0.636
22	1.324	0.796
23	0.000	0.000
24	0.690	0.534

Введите значение высоты:

Введите значение В:

Введите значение L:

Обработка

СКО для x:

СКО для y:

СКО для z:

СКО для D:

Рисунок 3.5. Ввод значений в формы

7. Нажимаем кнопку обработка:

Data app

Glomass

Загрузить

	SISRE, m	SISVE, mm/s
1	0.427	0.508
2	1.205	0.650
3	0.863	0.511
4	0.851	0.569
5	0.682	0.597
6	0.000	0.000
7	0.686	0.469
8	2.104	0.506
9	2.751	1.216
10	0.000	0.000
11	0.000	0.000
12	1.390	0.753
13	1.451	0.657
14	0.914	0.540
15	0.519	0.562
16	1.392	0.580
17	0.440	0.701
18	1.481	0.551
19	1.023	0.591
20	1.190	0.768
21	0.849	0.636
22	1.324	0.796
23	0.000	0.000
24	0.690	0.534

Введите значение высоты:
200

Введите значение В:
37.611680

Введите значение L:
55.819715

Обработка

СКО для x:
0,398

СКО для y:
0,757

СКО для z:
0,605

СКО для D:
0,304

Рисунок 3.6. Полученные значения

Получаем значения СКО.

ЗАКЛЮЧЕНИЕ

Получил значения СКО для x, y, z - координаты и погрешности временной шкалы, которые будут необходимы в дальнейшем для оценки прогнозирования точности ГНСС.

Все поставленные задачи были выполнены.

Так же в отчете рассмотрел ключевые детали, с которыми столкнулся при разработке ПО: настройкой работы системы «Git», с библиотекой «Armadillo», которая упростила математические вычисления и разбором новых алгоритмов, которые создал в этом семестре. Также написал руководство пользователя.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1]. ГЛОНАСС принципы построения и функционирования. Под редакцией А.И.Перова, В.Н.Харисова. Изд.4-е, 2010, 800 с

[2]. ГОСТ 32454-2013 Глобальная навигационная спутниковая система. Параметры радионавигационного поля. Технические требования и методы испытаний [Электронный ресурс]: <http://docs.cntd.ru/document/1200110468>

[3]. «Информационно-аналитического центра координатно-временного и навигационного обеспечения «www.glonass-iac.ru» »

[4]. Сервер «информационно-аналитического центра координатно-временного и навигационного обеспечения «<ftp://glonass-iac.ru>» »

[5]. Официальный сайт «Armadillo»: «<http://arma.sourceforge.net>»

ПРИЛОЖЕНИЕ

Приложение 1.

Значение Ω и ϑ :

Номер спутника	ϑ , гр.	Ω , гр.
1	0	0
2	-45	0
3	-90	0
4	-135	0
5	-180	0
6	-225	0
7	-270	0
8	-315	0
9	15	120
10	-30	120
11	-75	120
12	-120	120
13	-165	120
14	-210	120
15	-255	120
16	-300	120
17	30	240
18	-15	240
19	-60	240
20	-105	240
21	-150	240
22	-195	240
23	-240	240
24	-285	240

Приложение 2.

Расчет матрицы \mathbf{D}_x :

$$\mathbf{D}_x = \left((\tilde{\mathbf{H}}^T \mathbf{D}_{n_R}^{-1} \tilde{\mathbf{H}})^{-1} \right)^T =$$

$$= \left(\left(\begin{array}{c|c} \begin{matrix} H_{1,1} & H_{2,1} & \cdots & H_{n,1} \\ H_{1,2} & H_{2,2} & \cdots & H_{n,2} \\ H_{1,3} & H_{2,3} & \cdots & H_{n,3} \\ H_{1,4} & H_{2,4} & \cdots & H_{n,4} \end{matrix} & \begin{matrix} \frac{1}{D_1} & 0 & 0 & 0 \\ 0 & \frac{1}{D_2} & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \frac{1}{D_n} \end{matrix} \end{array} \cdot \begin{array}{c|c} \begin{matrix} H_{1,1} & H_{1,2} & H_{1,3} & H_{1,4} \\ H_{2,1} & H_{2,2} & H_{2,3} & H_{2,4} \\ \cdots & \cdots & \cdots & \cdots \\ H_{n,1} & H_{n,2} & H_{n,3} & H_{n,4} \end{matrix} & \end{array} \right)^{-1} \right)^T =$$

$$= \left(\left(\begin{array}{c|c} \begin{matrix} H_{1,1} & H_{2,1} & \cdots & H_{n,1} \\ H_{1,2} & H_{2,2} & \cdots & H_{n,2} \\ H_{1,3} & H_{2,3} & \cdots & H_{n,3} \\ H_{1,4} & H_{2,4} & \cdots & H_{n,4} \end{matrix} & \begin{matrix} \frac{H_{1,1}}{D_1} & \frac{H_{1,2}}{D_1} & \frac{H_{1,3}}{D_1} & \frac{H_{1,4}}{D_1} \\ \frac{H_{2,1}}{D_2} & \frac{H_{2,2}}{D_2} & \frac{H_{2,3}}{D_2} & \frac{H_{2,4}}{D_2} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{H_{n,1}}{D_n} & \frac{H_{n,2}}{D_n} & \frac{H_{n,3}}{D_n} & \frac{H_{n,4}}{D_n} \end{matrix} \end{array} \right)^{-1} \right)^T =$$

$$= \left(\begin{pmatrix} \frac{H_{1,1}^2}{D_1} + \frac{H_{2,1}^2}{D_2} + \dots + \frac{H_{n,1}^2}{D_n} & \frac{H_{1,1} \cdot H_{1,2}}{D_1} + \frac{H_{2,1} \cdot H_{2,2}}{D_2} + \dots + \frac{H_{n,1} \cdot H_{n,2}}{D_n} \\ \frac{H_{1,2} \cdot H_{1,1}}{D_1} + \frac{H_{2,2} \cdot H_{2,1}}{D_2} + \dots + \frac{H_{n,2} \cdot H_{n,1}}{D_n} & \frac{H_{1,2}^2}{D_1} + \frac{H_{2,2}^2}{D_2} + \dots + \frac{H_{n,2}^2}{D_n} \\ \frac{H_{1,3} \cdot H_{1,1}}{D_1} + \frac{H_{2,3} \cdot H_{2,1}}{D_2} + \dots + \frac{H_{n,3} \cdot H_{n,1}}{D_n} & \frac{H_{1,3} \cdot H_{1,2}}{D_1} + \frac{H_{2,3} \cdot H_{2,2}}{D_2} + \dots + \frac{H_{n,3} \cdot H_{n,2}}{D_n} \\ \frac{H_{1,4} \cdot H_{1,1}}{D_1} + \frac{H_{2,4} \cdot H_{2,1}}{D_2} + \dots + \frac{H_{n,4} \cdot H_{n,1}}{D_n} & \frac{H_{1,4} \cdot H_{1,2}}{D_1} + \frac{H_{2,4} \cdot H_{2,2}}{D_2} + \dots + \frac{H_{n,4} \cdot H_{n,2}}{D_n} \end{pmatrix}^{-1} \right)^T$$

$$\begin{pmatrix} \frac{H_{1,1} \cdot H_{1,3}}{D_1} + \frac{H_{2,1} \cdot H_{2,3}}{D_2} + \dots + \frac{H_{n,1} \cdot H_{n,3}}{D_n} & \frac{H_{1,1} \cdot H_{1,4}}{D_1} + \frac{H_{2,1} \cdot H_{2,4}}{D_2} + \dots + \frac{H_{n,1} \cdot H_{n,4}}{D_n} \\ \frac{H_{1,2} \cdot H_{1,3}}{D_1} + \frac{H_{2,2} \cdot H_{2,3}}{D_2} + \dots + \frac{H_{n,2} \cdot H_{n,3}}{D_n} & \frac{H_{1,2} \cdot H_{1,4}}{D_1} + \frac{H_{2,2} \cdot H_{2,4}}{D_2} + \dots + \frac{H_{n,2} \cdot H_{n,4}}{D_n} \\ \frac{H_{1,3}^2}{D_1} + \frac{H_{2,3}^2}{D_2} + \dots + \frac{H_{n,3}^2}{D_n} & \frac{H_{1,3} \cdot H_{1,4}}{D_1} + \frac{H_{2,3} \cdot H_{2,4}}{D_2} + \dots + \frac{H_{n,3} \cdot H_{n,4}}{D_n} \\ \frac{H_{1,4} \cdot H_{1,3}}{D_1} + \frac{H_{2,4} \cdot H_{2,3}}{D_2} + \dots + \frac{H_{n,4} \cdot H_{n,3}}{D_n} & \frac{H_{1,4}^2}{D_1} + \frac{H_{2,4}^2}{D_2} + \dots + \frac{H_{n,4}^2}{D_n} \end{pmatrix}$$

Приложение 3

dataMain.cpp

```
#include "dataMain.h"
#include "parser.h"
#include "angle.h"
#include "ephemerids.h"
#include "xyz2enu.h"
#include <wx/msgdlg.h>
#include <windows.h>
#include <wininet.h>
#include <iostream>
#include <string>
#include <stdio.h>
#include <armadillo>
#include <fstream>
#include <wx/string.h>
#include <wx/textfile.h>
#include <wx/dialog.h>
#include <wx/msgdlg.h>
#include <wx/spinctrl.h>
```

```

#include <wx/intl.h>
#include <wx/settings.h>
#define SQUARE(val) val * val
using std::cout;
using std::cin;
using std::endl;
using namespace std;
using namespace arma;
//int numberSput;
//(*InternalHeaders(dataDialog)
#include <wx/intl.h>
#include <wx/string.h>
//*)
//helper functions
enum wxbuildinfoformat
{
    short_f, long_f
};
wxString wxbuildinfo(wxbuildinfoformat format)
{
    wxString wxbuild(wxVERSION_STRING);
    if (format == long_f )
    {
#ifdef __WXMSW__
        wxbuild << _T("-Windows");
#elif defined(__UNIX__)
        wxbuild << _T("-Linux");
#endif
#ifdef wxUSE_UNICODE
        wxbuild << _T("-Unicode build");
#else
        wxbuild << _T("-ANSI build");
#endif // wxUSE_UNICODE
    }

    return wxbuild;
}
//(*IdInit(dataDialog)
const long dataDialog::ID_CHOICE1 = wxNewId();
const long dataDialog::ID_BUTTON2 = wxNewId();
const long dataDialog::ID_NOTEBOOK1 = wxNewId();
const long dataDialog::ID_BUTTON1 = wxNewId();
const long dataDialog::ID_TEXTCTRL1 = wxNewId();
const long dataDialog::ID_TEXTCTRL2 = wxNewId();
const long dataDialog::ID_TEXTCTRL3 = wxNewId();
const long dataDialog::ID_TEXTCTRL4 = wxNewId();

```

```

const long dataDialog::ID_TEXTCTRL5 = wxNewId();
const long dataDialog::ID_TEXTCTRL6 = wxNewId();
const long dataDialog::ID_TEXTCTRL7 = wxNewId();
const long dataDialog::ID_STATICTEXT1 = wxNewId();
const long dataDialog::ID_STATICTEXT2 = wxNewId();
const long dataDialog::ID_STATICTEXT3 = wxNewId();
const long dataDialog::ID_STATICTEXT4 = wxNewId();
const long dataDialog::ID_STATICTEXT5 = wxNewId();
const long dataDialog::ID_STATICTEXT6 = wxNewId();
const long dataDialog::ID_STATICTEXT7 = wxNewId();
const long dataDialog::ID_SASHWINDOW1 = wxNewId();
/*)
const long dataDialog::ID_GRID = wxNewId();
BEGIN_EVENT_TABLE(dataDialog,wxDialog)
    /*(*EventTable(dataDialog)
    /**)
END_EVENT_TABLE()

bool download(LPCSTR server, LPCSTR login, LPCSTR pass, LPCSTR local_file, LPCSTR remote_file)
{
    bool status;
    HINERNET hOpen, hConnection;

    hOpen = InternetOpen(NULL, INTERNET_OPEN_TYPE_DIRECT, NULL, NULL, 0);
    if (hOpen == NULL)
        return false;

    hConnection = InternetConnectA(hOpen, server, 21, login, pass, INTERNET_SERVICE_FTP,
INTERNET_FLAG_PASSIVE, 0);
    if (hConnection == NULL)
    {
        InternetCloseHandle(hOpen);
        return false;
    }

    status=FtpGetFileA(hConnection, local_file, remote_file, true, 0, FTP_TRANSFER_TYPE_UNKNOWN, 0);
    InternetCloseHandle(hConnection);
    InternetCloseHandle(hOpen);
    return status;
}

dataDialog::dataDialog(wxWindow* parent,wxWindowID id)
{
    /*(*Initialize(dataDialog)
    Create(parent, wxID_ANY, _("Data app"), wxDefaultPosition, wxDefaultSize, wxDEFAULT_DIALOG_STYLE,
_T("wxID_ANY"));
    SetClientSize(wxSize(502,571));

```

```

        SashWindow1 = new wxSashWindow(this, ID_SASHWINDOW1, wxPoint(56,40), wxSize(432,520),
wxSW_3D|wxCLIP_CHILDREN, _T("ID_SASHWINDOW1"));

        Choice1 = new wxChoice(SashWindow1, ID_CHOICE1, wxPoint(33,16), wxSize(244,21), 0, 0, 0, wxDefaultValidator,
_T("ID_CHOICE1"));

        Choice1->SetSelection( Choice1->Append(_T("Glonass")) );
        Choice1->Append(_T("GPS"));
        Choice1->Append(_T("Galileo"));
        Choice1->Append(_T("Beidou"));
        Choice1->Append(_T("QZSS"));

        Button2 = new wxButton(SashWindow1, ID_BUTTON2, _T("Загрузить"), wxPoint(305,16), wxSize(127,23), 0,
wxDefaultValidator, _T("ID_BUTTON2"));

        Notebook1 = new wxNotebook(SashWindow1, ID_NOTEBOOK1, wxPoint(124,214), wxDefaultSize, 0,
_T("ID_NOTEBOOK1"));

        Down = new wxButton(SashWindow1, ID_BUTTON1, _T("Обработка "), wxPoint(305,216), wxSize(127,23), 0,
wxDefaultValidator, _T("ID_BUTTON1"));

        TextCtrlH = new wxTextCtrl(SashWindow1, ID_TEXTCTRL1, _T(""), wxPoint(305,95), wxSize(127,-1), 0,
wxDefaultValidator, _T("ID_TEXTCTRL1"));

        TextCtrlB = new wxTextCtrl(SashWindow1, ID_TEXTCTRL2, _T(""), wxPoint(305,140), wxSize(127,-1), 0,
wxDefaultValidator, _T("ID_TEXTCTRL2"));

        TextCtrlL = new wxTextCtrl(SashWindow1, ID_TEXTCTRL3, _T(""), wxPoint(305,185), wxSize(127,-1), 0,
wxDefaultValidator, _T("ID_TEXTCTRL3"));

        TextCtrlX = new wxTextCtrl(SashWindow1, ID_TEXTCTRL4, wxEmptyString, wxPoint(305,279), wxSize(127,-1), 0,
wxDefaultValidator, _T("ID_TEXTCTRL4"));

        TextCtrlY = new wxTextCtrl(SashWindow1, ID_TEXTCTRL5, wxEmptyString, wxPoint(305,324), wxSize(127,-1), 0,
wxDefaultValidator, _T("ID_TEXTCTRL5"));

        TextCtrlZ = new wxTextCtrl(SashWindow1, ID_TEXTCTRL6, wxEmptyString, wxPoint(305,369), wxSize(127,-1), 0,
wxDefaultValidator, _T("ID_TEXTCTRL6"));

        TextCtrlD = new wxTextCtrl(SashWindow1, ID_TEXTCTRL7, wxEmptyString, wxPoint(305,414), wxSize(127,-1), 0,
wxDefaultValidator, _T("ID_TEXTCTRL7"));

        StaticText1 = new wxStaticText(SashWindow1, ID_STATICTEXT1, _T("Введите значение высоты:"), wxPoint(305,77),
wxDefaultSize, 0, _T("ID_STATICTEXT1"));

        StaticText2 = new wxStaticText(SashWindow1, ID_STATICTEXT2, _T("Введите значение B:"), wxPoint(305,122),
wxDefaultSize, 0, _T("ID_STATICTEXT2"));

        StaticText3 = new wxStaticText(SashWindow1, ID_STATICTEXT3, _T("Введите значение L:"), wxPoint(305,167),
wxDefaultSize, 0, _T("ID_STATICTEXT3"));

        StaticText4 = new wxStaticText(SashWindow1, ID_STATICTEXT4, _T("СКО для x:"), wxPoint(305,261), wxDefaultSize,
0, _T("ID_STATICTEXT4"));

        StaticText5 = new wxStaticText(SashWindow1, ID_STATICTEXT5, _T("СКО для y:"), wxPoint(305,306), wxDefaultSize,
0, _T("ID_STATICTEXT5"));

        StaticText6 = new wxStaticText(SashWindow1, ID_STATICTEXT6, _T("СКО для z:"), wxPoint(305,351), wxDefaultSize,
0, _T("ID_STATICTEXT6"));

        StaticText7 = new wxStaticText(SashWindow1, ID_STATICTEXT7, _T("СКО для Д:"), wxPoint(305,396), wxDefaultSize,
0, _T("ID_STATICTEXT7"));

        SashWindow1->SetSashVisible(wxSASH_TOP, true);
        SashWindow1->SetSashVisible(wxSASH_BOTTOM, true);
        SashWindow1->SetSashVisible(wxSASH_LEFT, true);

```

```

        SashWindow1->SetSashVisible(wxSASH_RIGHT, true);

Connect(ID_BUTTON2,wxEVT_COMMAND_BUTTON_CLICKED,(wxObjectEventFunction)&dataDialog::OnButton2Click);

Connect(ID_BUTTON1,wxEVT_COMMAND_BUTTON_CLICKED,(wxObjectEventFunction)&dataDialog::OnButton1Click1);

Connect(ID_TEXTCTRL1,wxEVT_COMMAND_TEXT_UPDATED,(wxObjectEventFunction)&dataDialog::OnTextCtrl1Text1);

Connect(ID_SASHWINDOW1,wxEVT_SASH_DRAGGED,(wxObjectEventFunction)&dataDialog::OnSashWindow1SashDragged);
    Connect(wxID_ANY,wxEVT_INIT_DIALOG,(wxObjectEventFunction)&dataDialog::OnInit);
    /*)
Grid = new wxGrid(SashWindow1, ID_GRID, wxPoint(33,72), wxSize(244,490), 0, _T("ID_GRID"));
Grid->CreateGrid(24,2);
Grid->SetColLabelValue(0, _("SISRE, m"));
Grid->SetColLabelValue(1, _("SISVE, mm/s"));
Grid->SetDefaultCellFont( Grid->GetFont() );
Grid->SetDefaultCellTextColour( Grid->GetForegroundColour() );
}

dataDialog::~dataDialog()
{
    /*(*Destroy(dataDialog)
    /**)
}

void dataDialog::OnQuit(wxCommandEvent& event)
{
    Close();
}

void dataDialog::OnInit(wxInitDialogEvent& event) { };
void dataDialog::OnAbout(wxCommandEvent& event)
{
    wxString msg = wxbuildinfo(long_f);
    wxMessageBox(msg, _("Welcome to..."));
}

void dataDialog::OnChoice1Select(wxCommandEvent& event)
{
}

void dataDialog::OnSashWindow1SashDragged(wxSashEvent& event)
{
}

void dataDialog::OnButton2Click(wxCommandEvent& event)
{
    wxString s;
    const char* File1 ;
    const char* file ;

```

```

//wxMessageBox(Choice1->GetString(Choice1->GetSelection()), _(""));
if ((Choice1->GetString(Choice1->GetSelection()))== "GPS")
{
    File1 = "/MCC/PRODUCTS/LATEST/MERMS-GSC_C.ete";
    file = "MERMS-GSC_C.ete";
    wxTextFile file11(wxT("MERMS-GSC_C.ete"));
    if (file11.Exists())
    {wxRemoveFile(file);
    }
}
if ((Choice1->GetString(Choice1->GetSelection()))== "Glonass")
{
    File1 = "/MCC/PRODUCTS/LATEST/MERMS-RSC_C.ete";
    file = "MERMS-RSC_C.ete";
    wxTextFile file11(wxT("MERMS-RSC_C.ete"));
    if (file11.Exists())
    {wxRemoveFile(file);
    }
}
if ((Choice1->GetString(Choice1->GetSelection()))== "Galileo")
{
    File1 = "/MCC/PRODUCTS/LATEST/MERMS-ESC_C.ete";
    file = "MERMS-ESC_C.ete";
    wxTextFile file11(wxT("MERMS-ESC_C.ete"));
    if (file11.Exists())
    {wxRemoveFile(file);
    }
}
if ((Choice1->GetString(Choice1->GetSelection()))== "Beidou")
{
    File1 = "/MCC/PRODUCTS/LATEST/MERMS-CSC_C.ete";
    file = "MERMS-CSC_C.ete";
    wxTextFile file11(wxT("MERMS-CSC_C.ete"));
    if (file11.Exists())
    {wxRemoveFile(file);
    }
}
if ((Choice1->GetString(Choice1->GetSelection()))== "QZSS")
{
    File1 = "/MCC/PRODUCTS/LATEST/MERMS-JSC_C.ete";
    file = "MERMS-JSC_C.ete";
    wxTextFile file11(wxT("MERMS-JSC_C.ete"));
    if (file11.Exists())
    {wxRemoveFile(file);
    }
}

```

```

    }
    bool down = download( "glonass-iac.ru", NULL, NULL, File1, file);
if (!down)
{
    wxMessageBox(_("Error"), _("Error"));
    return;
}
Gridd(file);
}

void dataDialog::Gridd(const char* file)
{
    int k=0;
    int sizeY;
    if (Grid != NULL)
    {
        delete Grid;
    }
    memset(&SISerr,0, sizeof(SISerr));
    int max_sats = parse(file);
    sizeY=490;
    Grid = new wxGrid(SashWindow1, ID_GRID, wxPoint(33,72), wxSize(244,sizeY), 0, _T("ID_GRID"));
    wxString s;
    Grid->CreateGrid(max_sats,2);
    Grid->SetColLabelValue(0, _("SISRE, m"));
    Grid->SetColLabelValue(1, _("SISVE, mm/s"));
    Grid->SetDefaultCellFont( Grid->GetFont() );
    Grid->SetDefaultCellTextColour( Grid->GetForegroundColour() );
    for (k=0; k<max_sats; k++ )
    {
        Grid->SetCellValue((k), 0, wxString::Format("%.3f", SISerr[k].SISRE));
        Grid->SetCellValue((k), 1, wxString::Format("%.3f", SISerr[k].SISVE));
    }
}

void dataDialog::OnChoice1Select3(wxCommandEvent& event)
{
}

void dataDialog::OnTextCtrl1Text1(wxCommandEvent& event)
{
}

void dataDialog::OnSpinCtrl1Change(wxSpinEvent& event)
{
}

void dataDialog::OnButton1Click1(wxCommandEvent& event)
{
    TextCtrlX->Clear();
    TextCtrlY->Clear();
}

```



```

TextCtrlZ->Clear();
TextCtrlD->Clear();
ostream streamX(TextCtrlX);
ostream streamY(TextCtrlY);
ostream streamZ(TextCtrlZ);
ostream streamD(TextCtrlD);
if ((Choice1->GetString(Choice1->GetSelection()))== "Glonass")
{
//wxString s;
double h;
double Bgrad;
double Lgrad;
TextCtrlH->GetValue().ToDouble(&h);
TextCtrlB->GetValue().ToDouble(&Bgrad);
TextCtrlL->GetValue().ToDouble(&Lgrad);
double PI = M_PI;
double B; //Latitude
double L; //Longitude
B=Bgrad*PI/180;
L=Lgrad*PI/180;
double N;
double e=0;
double a=6378136; // радиус З
// Получение координат потребителя
N=a/sqrt(1-(e*e)*(sin(B))*(sin(B)));
double Coord_x;
double Coord_y;
double Coord_z;
Coord_x = (N+h)*cos(B)*cos(L);
Coord_y = (N+h)*cos(B)*sin(L);
Coord_z = ((1-e*e)*N+h)*sin(B);
double Coord_user[3];
Coord_user[0]=Coord_x;
Coord_user[1]= Coord_y;
Coord_user[2]= Coord_z;
double Coord_sput[3];
double alpha;
using namespace std;

// Расчет матрицы Dn, Hn, SKO
//int numberSput;
int numberSput = 24;

```

```

int vsb[numberSput] ;
int sumvsb = 0;
vector<int> Visibles; //вектор из кол-во элементов - visibles
for (int i=1; i<=numberSput; i++)
{
    // Получение коорд спутников
    GlonassCoordinates Coord_sp = ephemerids(i,10000);
    Coord_sput[0] = Coord_sp.X;
    Coord_sput[1] = Coord_sp.Y;
    Coord_sput[2] = Coord_sp.Z;
    // Определение угла
    alpha = 90 - (angle(Coord_sput, Coord_user, B, L)*180/PI);
    // определение видимости спутника
    vsb[i]=0;
    if ((alpha)>5)
    {
        vsb[i]=1;
        sumvsb++;
        Visibles.push_back(i); // добавление элемента в конец вектора

    }
}
int i = 0;
mat Dn;
Dn.zeros(sumvsb, sumvsb);
for (int k=1; k<=numberSput; k++)
{
    if ((vsb[k]) == 1)
    {
        Dn(i,i) = SISerr[i].SISRE;
        i++;
    }

}
double max_val_Dn = Dn.max();
for (int i= 0; i<sumvsb; i++)
{
    if ( Dn(i,i) == 0)
    {
        Dn(i,i) = max_val_Dn;
    }
}
double dx;
double dy;
double dz;
double Ri;

```

```

mat H(sumvsb, 4);
H.zeros();
int numspout = 0;

for (int k=1; k<=numberSput; k++)
{
    if ((vsb[k]) == 1)
    {
        GlonassCoordinates Coord_sp = ephemerids((k),10000);
        dx=(Coord_sp.X-Coord_x);
        dy=(Coord_sp.Y-Coord_y);
        dz=(Coord_sp.Z- Coord_z);
        // Ri = sqrt (SQUARE(dx)+SQUARE(dy)+SQUARE(dz));
        Ri = sqrt (pow(dx,2)+pow(dy,2)+pow(dz,2));

        H(numspout, 0 ) = dx/Ri;
        H(numspout, 1) = dy/Ri;
        H(numspout, 2) = dz/Ri;
        H(numspout, 3) = 1;
        numspout++ ;
    }
}
mat Htr = H.t();
mat sko = sqrt((inv(Htr*inv(Dn)*H)).t());
streamX << wxString::Format("%.3f", sko(0,0)) ;
streamY << wxString::Format("%.3f", sko(1,1)) ;
streamZ << wxString::Format("%.3f", sko(2,2)) ;
streamD << wxString::Format("%.3f", sko(3,3)) ;
}
else
{
    wxMessageBox(_("Выберите другую ГНСС"), _("Error"));
}
streamX.flush();
streamY.flush();
streamZ.flush();
streamD.flush();
}

```

Приложение 4

parser.c

```
#include <stdio.h>
```

```

#include "parser.h"

/*****

* Extern Data

data_t SISerr[75];

/

* Function Definitions

*****/

int parse(const char* file)
{
    int i;

    // int k;

    FILE* fd;

    char systype;

    char dummy[30];

    //char hr[30];

    // char min[30];

    //char sec[30];

    // fd = fopen("MERMS-ESC_C.ete", "r");

    fd = fopen(file, "r");

    while ( !feof(fd) )
    {
        fscanf(fd, "%c", &systype);

        fscanf(fd, "%c", &systype);

        /* if (systype !=('G' || 'R'))
        {
            printf("Sync error\n");

            break;

        }*/

```

```

fscanf(fd, "%d", &i);

SISerr[i-1].systype = systype;

fscanf(fd, "%d", (int *)dummy);

fscanf(fd, "%u", (unsigned int *)dummy);

fscanf(fd, "%c", dummy);

fscanf(fd, "%u", (unsigned int *)dummy);

fscanf(fd, "%c", dummy);

fscanf(fd, "%u", (unsigned int *)dummy);

fscanf(fd, "%u", (unsigned int *)dummy);

fscanf(fd, "%c", dummy);

fscanf(fd, "%u", (unsigned int *)dummy);

fscanf(fd, "%c", dummy);

fscanf(fd, "%u", (unsigned int *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%u", (unsigned int *)dummy);

fscanf(fd, "%u", (unsigned int *)dummy);

fscanf(fd, "%u", (unsigned int *)dummy);

fscanf(fd, "%u", (unsigned int *)dummy);

fscanf(fd, "%u", (unsigned int *)dummy);

fscanf(fd, "%u", (unsigned int *)dummy);

fscanf(fd, "%u", (unsigned int *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", (double *)dummy);

```

```

fscanf(fd, "%lf", &(SISerr[i-1].SISRE));

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", (double *)dummy);

fscanf(fd, "%lf", &(SISerr[i-1].SISVE));

fscanf(fd, "%c", &systype);

}

int imax=i;

fclose(fd);

return imax;

}

```

Приложение 5

Parser.h

```

#ifndef PARSER_H

#define PARSER_H

#ifdef __cplusplus

extern "C" {

#endif

typedef struct

{

```

```

char systype;

double SISRE;

double SISVE;

} data_t;

extern data_t SISerr[75];

int parse(const char* file);

#ifdef __cplusplus
}

#endif

#endif

```

Приложение 6

data.cbp

```

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>

<CodeBlocks_project_file>

    <FileVersion major="1" minor="6" />

    <Project>

        <Option title="Data" />

        <Option pch_mode="2" />

        <Option compiler="gcc" />

        <Build>

            <Target title="Debug">

                <Option platforms="Windows;" />

                <Option output="bin/Debug/Data" prefix_auto="1" extension_auto="1" />

                <Option object_output="obj/Debug/" />

                <Option type="0" />

                <Option compiler="gcc" />

                <Option projectLinkerOptionsRelation="2" />

                <Compiler>

                    <Add option="-g" />

```

```

        <Add option="-D__WXDEBUG__" />

    </Compiler>

    <Linker>

        <Add option="-static-libstdc++" />

        <Add option="-static-libgcc" />

        <Add option="-static" />

        <Add library="libwxmsw30u.a" />

        <Add library="libwxpng.a" />

        <Add library="libwxjpeg.a" />

        <Add library="libwxtiff.a" />

        <Add library="libwxzlib.a" />

        <Add library="libwininet.a" />

    </Linker>

</Target>

</Build>

<Compiler>

    <Add option="-Wall" />

    <Add option="-pipe" />

    <Add option="-mthreads" />

    <Add option="-D__GNUWIN32__" />

    <Add option="-D__WXMSW__" />

    <Add option="-DwxUSE_UNICODE" />

    <Add directory="$(#wx)/include" />

    <Add directory="$(#wx)/lib/gcc_lib/mswud" />

    <Add directory="." />

</Compiler>

<ResourceCompiler>

    <Add directory="$(#wx)/include" />

    <Add directory="$(#wx)/lib/gcc_lib/mswud" />

</ResourceCompiler>

```


<Linker>

```
<Add option="-static-libstdc++" />
<Add option="-static-libgcc" />
<Add option="-static" />
<Add option="-mthreads" />
<Add library="libwxmsw30u.a" />
<Add library="libwxpng.a" />
<Add library="libwxjpeg.a" />
<Add library="libwxtiff.a" />
<Add library="libwxzlib.a" />
<Add library="libkernel32.a" />
<Add library="libuser32.a" />
<Add library="libgdi32.a" />
<Add library="libwinpool.a" />
<Add library="libcomdlg32.a" />
<Add library="libadvapi32.a" />
<Add library="libshell32.a" />
<Add library="libole32.a" />
<Add library="liboleaut32.a" />
<Add library="libuuid.a" />
<Add library="libcomctl32.a" />
<Add library="libwsck32.a" />
<Add library="libodbc32.a" />
<Add library="libuxtheme.a" />
<Add library="libopenblas.a" />
<Add library="libshlwapi.a" />
<Add library="libversion.a" />
<Add library="liboleacc.a" />
<Add directory="$(#wx)/lib/gcc_lib" />
<Add directory="." />
```

```

</Linker>

<Unit filename="angle.cpp" />

<Unit filename="angle.h" />

<Unit filename="data.cbp" />

<Unit filename="dataApp.cpp" />

<Unit filename="dataApp.h" />

<Unit filename="dataMain.cpp" />

<Unit filename="dataMain.h" />

<Unit filename="ephemerids.cpp" />

<Unit filename="ephemerids.h" />

<Unit filename="parser.c">

    <Option compilerVar="CC" />

</Unit>

<Unit filename="parser.h" />

<Unit filename="resource.rc">

    <Option compilerVar="WINDRES" />

</Unit>

<Unit filename="wxsmith/datadialog.wxs" />

<Unit filename="xyz2enu.cpp" />

<Unit filename="xyz2enu.h" />

<Extensions>

    <code_completion />

    <envvars />

    <debugger />

    <lib_finder disable_auto="1" />

    <wxsmith version="1">

        <gui      name="wxWidgets"      src="dataApp.cpp"      main="dataDialog"
init_handlers="necessary" language="CPP" />

        <resources>

            <wxDialog wxs="wxsmith/datadialog.wxs" src="dataMain.cpp"
hdr="dataMain.h" fwddecl="0" i18n="1" name="dataDialog" language="CPP" />

```

```

</resources>

</wxsmith>

</Extensions>

</Project>

</CodeBlocks_project_file>

```

Приложение 7

dataMain.h

```

/*****

* Name:    dataApp.h

* Purpose: Defines Application Class

* Author:   aaa ()

* Created:  2019-08-14

* Copyright: aaa ()

* License:

*****/

#ifndef dataAPP_H

#define dataAPP_H

#include <wx/app.h>

class dataApp : public wxApp

{

public:

    virtual bool OnInit();

};

#endif // dataAPP_H

```

Приложение 8

xyz2enu.cpp

```
#include <math.h>

void xyz2enu(const double lat, const double lon, double * xyz2enu)

{

    /*% *****
    %*   Copyright c 2001 The board of trustees of the Leland Stanford   *
    %*           Junior University. All rights reserved.           *
    %*   This script file may be distributed and used freely, provided   *
    %*   this copyright notice is always kept with it.           *
    %*
    %*   Questions and comments should be directed to Todd Walter at:   *
    %*   twalter@stanford.edu
    % *****33*****

    %

    %FINDXYZ2ENU find the rotation matrix to go from XYZ ECEF coordinates
    %   to a local East North Up frame

    % [xyz2enu] = FINDXYZ2ENU(LAT, LON)

    % LAT, LON specify the coordinates of the center of the local frame in radians

    % XYZ2ENU is the rotation matrix such that DELTA_ENU = XYZ2ENU*DELTA_XYZ */

    xyz2enu[0] = sin(lon);

    xyz2enu[1] = cos(lon);

    xyz2enu[2] = 0.0;

    xyz2enu[5] = cos(lat);

    xyz2enu[8] = sin(lat);

    xyz2enu[3] = -xyz2enu[1] * xyz2enu[8];

    xyz2enu[4] = -xyz2enu[0] * xyz2enu[8];
```

```

xyz2enu[6] = xyz2enu[1] * xyz2enu[5];

xyz2enu[7] = xyz2enu[0] * xyz2enu[5];

xyz2enu[0] = -xyz2enu[0];

}

```

Приложение 9

xyz2enu.h

```

#ifndef xyz2enu_H

#define xyz2enu_H


void xyz2enu(const double lat, const double lon, double * xyz2enu);


#endif // dataAPP_H

```

Приложение 10

ephemerids.cpp

```

#include "ephemerids.h"


#define _USE_MATH_DEFINES

#include <math.h>


#include <windows.h>

#include <wininet.h>

#include <iostream>

#include <string>

#include <stdio.h>

```

```
#include <wx/msgdlg.h>
```

```
#include <wx/string.h>
```

```
#include <wx/textfile.h>
```

```
#include <wx/dialog.h>
```

```
#include <wx/msgdlg.h>
```

```
#include <wx/spinctrl.h>
```

```
#include <wx/intl.h>
```

```
#include <wx/settings.h>
```

```
GlonassCoordinates ephemerids(int n, int t)
```

```
{
```

```
    GlonassCoordinates Coordinates;
```

```
    double TETTA;
```

```
    double OMEGA;
```

```
    switch (n)
```

```
    {
```

```
        case 1:
```

```
            TETTA = 0;
```

```
            OMEGA = 0;
```

```
            break;
```

```
        case 2:
```

```
            TETTA = -45;
```

```
            OMEGA = 0;
```

```
            break;
```

```
        case 3:
```

```
            TETTA = -90;
```

```
            OMEGA = 0;
```

```
            break;
```

```
        case 4:
```

```
            TETTA = -135;
```

```
    OMEGA = 0;

    break;

case 5:

    TETTA = -180;

    OMEGA = 0;

    break;

case 6:

    TETTA = -225;

    OMEGA = 0;

    break;

case 7:

    TETTA = -270;

    OMEGA = 0;

    break;

case 8:

    TETTA = -315;

    OMEGA = 0;

    break;

case 9:

    TETTA = 15;

    OMEGA = 120;

    break;

case 10:

    TETTA = -30;

    OMEGA = 120;

    break;

case 11:

    TETTA = -75;

    OMEGA = 120;
```

```
        break;

case 12:

    TETTA = -120;

    OMEGA = 120;

    break;

case 13:

    TETTA = -165;

    OMEGA = 120;

    break;

case 14:

    TETTA = -210;

    OMEGA = 120;

    break;

case 15:

    TETTA = -255;

    OMEGA = 120;

    break;

case 16:

    TETTA = -300;

    OMEGA = 120;

    break;

case 17:

    TETTA = 30;

    OMEGA = 240;

    break;

case 18:

    TETTA = -15;

    OMEGA = 240;

    break;

case 19:
```



```

        TETTA = -60;

        OMEGA = 240;

        break;

case 20:

        TETTA = -105;

        OMEGA = 240;

        break;

case 21:

        TETTA = -150;

        OMEGA = 240;

        break;

case 22:

        TETTA = -195;

        OMEGA = 240;

        break;

case 23:

        TETTA = -240;

        OMEGA = 240;

        break;

case 24:

        TETTA = -285;

        OMEGA = 240;

        break;

}

double PI = M_PI;

TETTA = TETTA*PI/180;

OMEGA = OMEGA*PI/180;


double ci;

double si;

```

```

ci = cos((64.8*PI/180));

si = sin((64.8*PI/180));


double V_sat_glns = 1.5514*pow(10,-4);

double V_earth = 0; //7.292115*10^(-5);


double Tsat = t*V_sat_glns;

double Tear = -t*V_earth;


double e = 0;

double p = 19100000+6378136;

double r = p/(1+e*cos(TETTA));


Coordinates.X = r*(((cos(TETTA+Tsat))*(cos(OMEGA+Tear)))-((sin(TETTA+Tsat))*(sin(OMEGA+Tear)*ci)));

Coordinates.Y = r*(cos(TETTA+Tsat)*sin(OMEGA+Tear)+sin(TETTA+Tsat)*cos(OMEGA+Tear)*ci);

Coordinates.Z = r*(sin(TETTA+Tsat))*si;

return Coordinates;

}

```

Приложение 11

ephemerids.h

```

#ifndef EPHEMERIDS_H

#define EPHEMERIDS_H


typedef struct

{

    double X;

    double Y;

    double Z;

```

```

} GlonassCoordinates;

GlonassCoordinates ephemerids(int n, int t);

#endif

```

Приложение 12

angle.cpp

```

#include "angle.h"

#include "xyz2enu.h"

#define _USE_MATH_DEFINES
#include <math.h>

#define SQUARE(val) val * val

double angle(double Coord_sput[3], double Coord[3], double B, double L)
{
    double ES[3];

    double Renu[3];

    double R[3];

    double lengthES;

    double C[9];

    double a;

    ES[0]=(Coord_sput[0]-Coord[0]);

    ES[1]=(Coord_sput[1]-Coord[1]);

    ES[2]=(Coord_sput[2]- Coord[2]);

```

```

lengthES = sqrt (pow((Coord_sput[0]- Coord[0]),2)+pow((Coord_sput[1]-Coord[1]),2)+pow((Coord_sput[2]-
Coord[2]),2));

//lengthES = sqrt (SQUARE(Coord_sput[0]- Coord[0])+SQUARE(Coord_sput[1]-
Coord[1])+SQUARE(Coord_sput[2]- Coord[2]));

R[0]= ES[0]/lengthES; // вектор столбец

R[1]= ES[1]/lengthES;

R[2]= ES[2]/lengthES;


xyz2enu(B, L, C); // получение матрицы ENU в -> C

Renu[0]=C[0]*R[0]+C[1]*R[1]+C[2]*R[2]; //вектор столбец

Renu[1]=C[3]*R[0]+C[4]*R[1]+C[5]*R[2];

Renu[2]=C[6]*R[0]+C[7]*R[1]+C[8]*R[2];


a = acos (Renu[2]); // rad

return a;

}

```

Приложение 13

angle.h

```

#ifndef ANGLE_H

#define ANGLE_H

double angle(double Coord_sput[3],double Coord[3], double B, double L);

#endif

```

Приложение 14

wxsmith\datadiaslog.wxs

```

<?xml version="1.0" encoding="utf-8" ?>

```

```

<wxsmith>

    <object class="wxDialog" name="dataDialog">

        <title>Data app</title>

        <size>502,571</size>

        <id_arg>0</id_arg>

        <handler function="OnInit" entry="EVT_INIT_DIALOG" />

        <object class="wxSashWindow" name="ID_SASHWINDOW1" variable="SashWindow1"
member="yes">

            <pos>56,40</pos>

            <size>432,520</size>

            <handler function="OnSashWindow1SashDragged" entry="EVT_SASH_DRAGGED" />

            <object class="wxChoice" name="ID_CHOICE1" variable="Choice1" member="yes">

                <content>

                    <item>Glonass</item>

                    <item>GPS</item>

                    <item>Galileo</item>

                    <item>Beidou</item>

                    <item>QZSS</item>

                </content>

                <selection>0</selection>

                <pos>33,16</pos>

                <size>244,21</size>

                <handler function="OnChoice1Select4" entry="EVT_CHOICE" />

            </object>

            <object class="wxButton" name="ID_BUTTON2" variable="Button2" member="yes">

                <label>P—P°PiCṪCfP·P&C,CḦ</label>

                <pos>305,16</pos>

                <size>127,23</size>

                <handler function="OnButton2Click" entry="EVT_BUTTON" />

            </object>

```

```

member="yes">
    <object class="wxNotebook" name="ID_NOTEBOOK1" variable="Notebook1"

        <pos>124,214</pos>

    </object>

    <object class="wxButton" name="ID_BUTTON1" variable="Down" member="yes">

        <label>PhP±CᄒP°P±PsC,PcP° </label>

        <pos>305,216</pos>

        <size>127,23</size>

        <handler function="OnButton1Click1" entry="EVT_BUTTON" />

    </object>

    <object class="wxTextCtrl" name="ID_TEXTCTRL1" variable="TextCtrlH"

member="yes">

        <value>0</value>

        <pos>305,95</pos>

        <size>127,-1</size>

        <handler function="OnTextCtrl1Text1" entry="EVT_TEXT" />

    </object>

    <object class="wxTextCtrl" name="ID_TEXTCTRL2" variable="TextCtrlB"

member="yes">

        <value>0</value>

        <pos>305,140</pos>

        <size>127,-1</size>

    </object>

    <object class="wxTextCtrl" name="ID_TEXTCTRL3" variable="TextCtrlL"

member="yes">

        <value>0</value>

        <pos>305,185</pos>

        <size>127,-1</size>

    </object>

    <object class="wxTextCtrl" name="ID_TEXTCTRL4" variable="TextCtrlX"

member="yes">

        <pos>305,279</pos>

```

```

<size>127,-1</size>

</object>

<object    class="wxTextCtrl"    name="ID_TEXTCTRL5"    variable="TextCtrlY"

member="yes">

<pos>305,324</pos>

<size>127,-1</size>

</object>

<object    class="wxTextCtrl"    name="ID_TEXTCTRL6"    variable="TextCtrlZ"

member="yes">

<pos>305,369</pos>

<size>127,-1</size>

</object>

<object    class="wxTextCtrl"    name="ID_TEXTCTRL7"    variable="TextCtrlD"

member="yes">

<pos>305,414</pos>

<size>127,-1</size>

</object>

<object    class="wxStaticText"    name="ID_STATICTEXT1"    variable="StaticText1"

member="yes">

<label>P'PIPμPrPëC,Pμ P·PSP°C‡PμPSPëPμ PIC<CfPsC,C<:</label>

<pos>305,77</pos>

</object>

<object    class="wxStaticText"    name="ID_STATICTEXT2"    variable="StaticText2"

member="yes">

<label>P'PIPμPrPëC,Pμ P·PSP°C‡PμPSPëPμ B:</label>

<pos>305,122</pos>

</object>

<object    class="wxStaticText"    name="ID_STATICTEXT3"    variable="StaticText3"

member="yes">

<label>P'PIPμPrPëC,Pμ P·PSP°C‡PμPSPëPμ L:</label>

<pos>305,167</pos>

</object>

```

```

member="yes">
    <object class="wxStaticText" name="ID_STATICTEXT4" variable="StaticText4"
        <label>PŸPљPh PrP»CЏ C...:</label>
        <pos>305,261</pos>
    </object>
member="yes">
    <object class="wxStaticText" name="ID_STATICTEXT5" variable="StaticText5"
        <label>PŸPљPh PrP»CЏ y:</label>
        <pos>305,306</pos>
    </object>
member="yes">
    <object class="wxStaticText" name="ID_STATICTEXT6" variable="StaticText6"
        <label>PŸPљPh PrP»CЏ z:</label>
        <pos>305,351</pos>
    </object>
member="yes">
    <object class="wxStaticText" name="ID_STATICTEXT7" variable="StaticText7"
        <label>PŸPљPh PrP»CЏ P”:</label>
        <pos>305,396</pos>
    </object>
</object>
</object>
</wxsmith>

```