

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «МЭИ»**

**Утверждаю**

Научный руководитель, доцент каф. РТС

К.Т.Н.,

**Шатилов А.Ю.**

**ОТЧЕТ**

о научно-исследовательской работе

**«АВТОМАТИЗАЦИЯ ОБРАБОТКИ ДАННЫХ О ТОЧНОСТИ  
СПУТНИКОВЫХ НАВИГАЦИОННЫХ СИСТЕМ»**

*Выполнил*

*студент*

*Тасканов В.Е.*

*группа*

*ЭР-15-16*

Москва 2020 г.

## СПИСОК ИСПОЛНИТЕЛЕЙ

студент

Тасканов В.Е.

## Содержание

Введение .....	4
<b>ГЛАВА 1.ТЕОРИЯ .....</b>	<b>5</b>
1.1.Псевдо дальномерный метод .....	5
1.2.Оценка координат потребителя с учетом ошибок SISRE .....	6
1.2.1.Решение нахождения оценки координаты потребителя с учетом ошибок SISRE .....	7
<b>ГЛАВА 2.РАЗРАБОТКА ПРОГРАММЫ .....</b>	<b>9</b>
2.1.Подготовка программного обеспечения .....	10
2.1.1.Настройка MinGW .....	10
2.1.2.Настройка wxWidgets .....	11
2.2. Необходимые файлы для сборки проекта .....	21
2.3.Изменения по коду .....	22
2.3.1.Вывод массива данных .....	22
2.3.2.Исправления ошибок при скачивании файла .....	24
2.3.3.Выведение окна ошибки при неудачном скачивание .....	25
<b>ГЛАВА 3.РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ .....</b>	<b>26</b>
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>30</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....</b>	<b>30</b>
<b>ПРИЛОЖЕНИЕ.....</b>	<b>31</b>
Приложение 1 .....	31
Приложение 2 .....	36
Приложение 3 .....	38
Приложение 4 .....	38
Приложение 5 .....	41
Приложение 6 .....	43
Приложение 7 .....	44

## 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 8 семестре стояла **цель работы** – откорректировать предыдущую программу, устранив ошибки и добавив пользовательский интерфейс, а также изучить теоретическую часть – нахождения ошибок при определении координат.

В рамках данной цели решаются следующие задачи:

1. Изучить новые библиотеки C++ и с помощью них скорректировать программу
2. Изучить теоретическую составляющего для дальнейшего исследования

# 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].

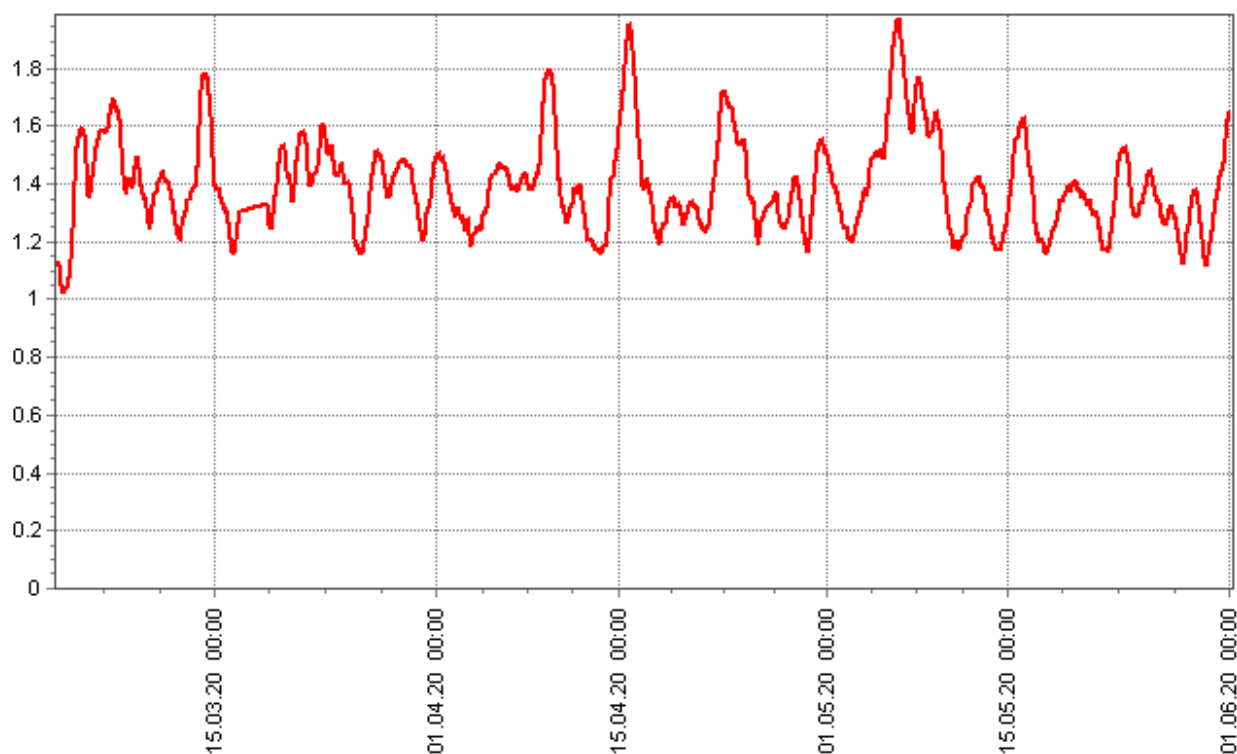


Рис.1.1. Суточные СКП SISRE ГЛОНАСС, м

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. Решение нахождения оценки координаты потребителя с учетом ошибок SISRE

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

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

$\mathbf{y}_{\tilde{\mathbf{R}},k} = \begin{bmatrix} y_{\tilde{R}_1,k} & y_{\tilde{R}_2,k} & \dots & y_{\tilde{R}_N,k} \end{bmatrix}^T$  и 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  $y_{\tilde{R},k} = R_i(x_k, y_k, z_k) + \tilde{D}' + n_{\tilde{R},k}$  относительно некоторой априорной оценки  $\tilde{\mathbf{x}}_k = \begin{bmatrix} \tilde{x}_k & \tilde{y}_k & \tilde{z}_k & \tilde{D}_k \end{bmatrix}^T$ , тогда:

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

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

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

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

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

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

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

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:

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

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

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

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



## ГЛАВА 2. 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.

## 2.1.Подготовка программного обеспечения

### 2.1.1.Настройка MinGW

Для правильной работы программы CodeBlocks, необходимо установить компилятор MinGW, для этого его нужно скачать с официального сайта [5]. Разархивировать в папку, в название которой нет пробелов или кириллицы. Далее открыть программу CodeBlocks, зайти в настройки компилятора и выставить оптимальные настройки (рис.2.1).

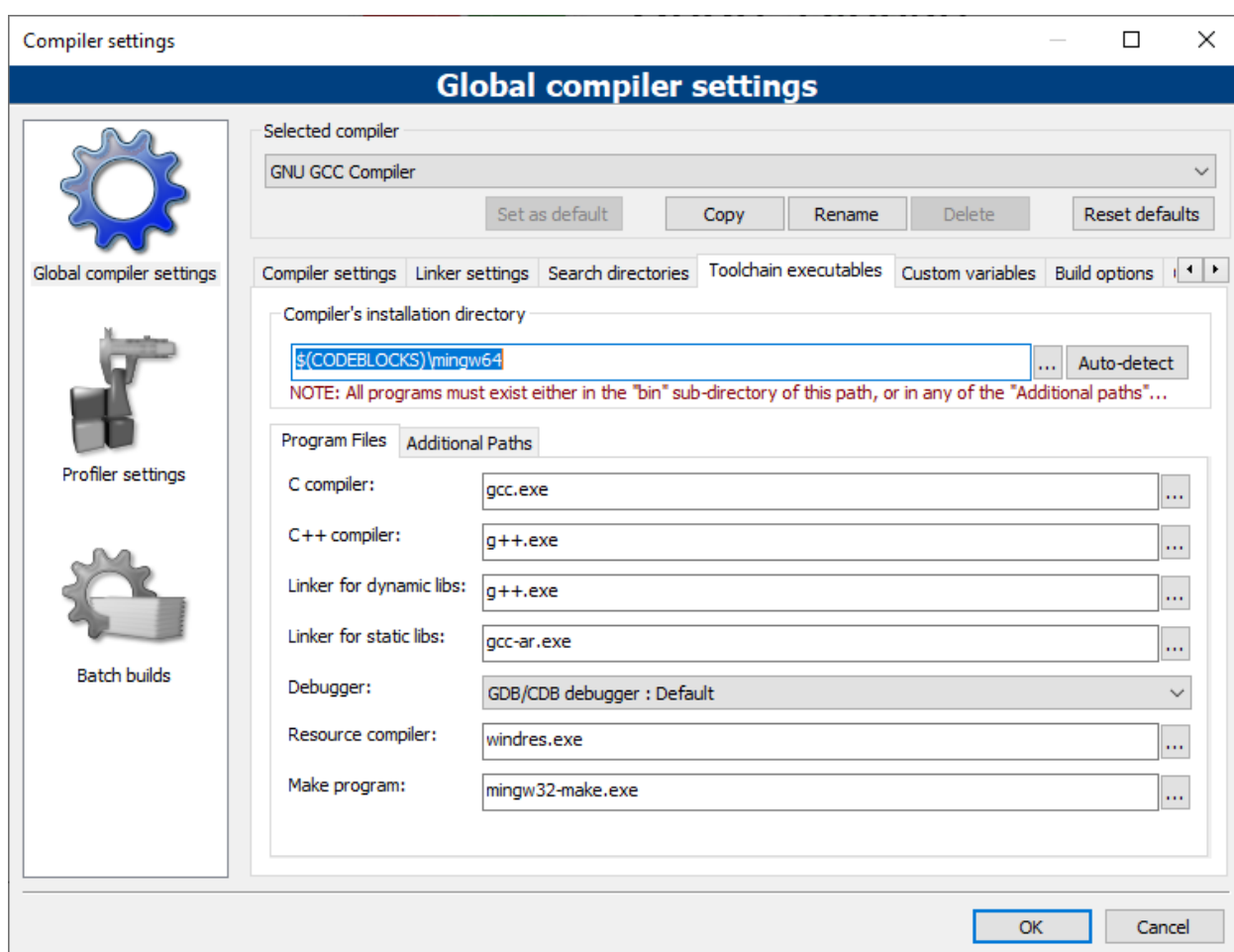


Рис.2.1.Настройки компилятора

## 2.1.2.Настройка wxWidgets

Для разработки программы с использованием wxWidgets необходимо:

1. Скачать исходные коды с официального сайта [6] и разархивировать их в папку, в название которой нет пробелов или кириллицы.

2. Открыть командную строку и вбить, определенные команды, которые необходимо для правильной сборки wxWidgets

3. Перейти в каталог сборки wxWidgets, В архиве wxWidgets есть специальная папка ("build\msw"), в которой хранятся файлы проекта для компиляции библиотеки под Windows с помощью различных сред. Переходим в эту папку с помощью команды:

```
«cd "C:\Library\wxwidgets\build\msw\"»
```

4. Выполните команду сборки. Рекомендованная команда для MinGW/GCC:

```
«mingw32-make -f makefile.gcc BUILD=release SHARED=1  
MONOLITHIC=1 UNICODE=1 CXXFLAGS=-fno-keep-inline-dllexport»
```

Этот этап может занять длительное время, обычно сборка длится от 30 минут до нескольких часов.

Чтобы создать проект с графическим интерфейсом, нужно изначально компилировать проект с использованием wxWidgets.

Для этого создадим новый проект:

1.В CodeBlocks выбираем пункт меню "File - New - Project..."

В открывшемся окне выбирается тип проекта "wxWidgets project"

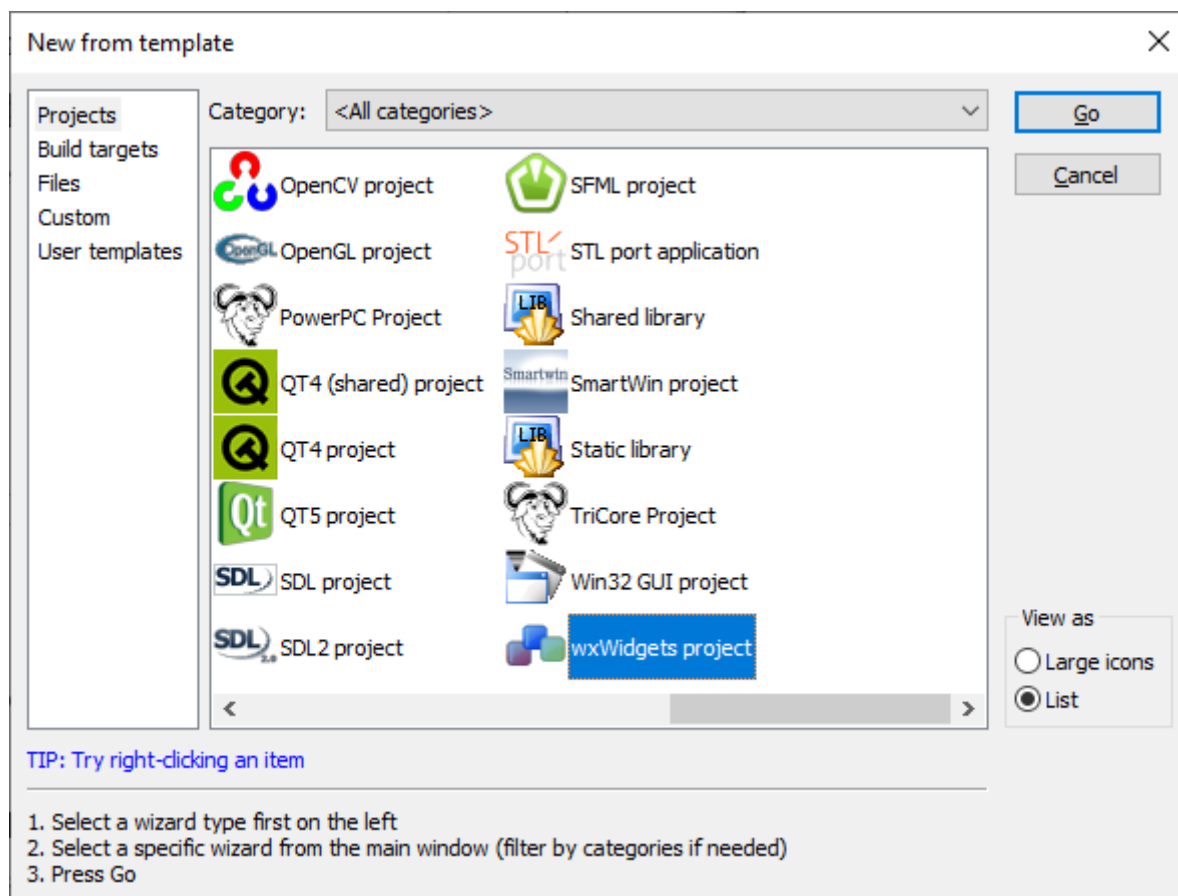


Рис.2.2. Выбор типа проекта

2. В следующем окне выбираем версию wxWidgets, в своем проекте использовал версию wxWidgets 3.0, поэтому выбираем пункт "wxWidgets 3.0.x".

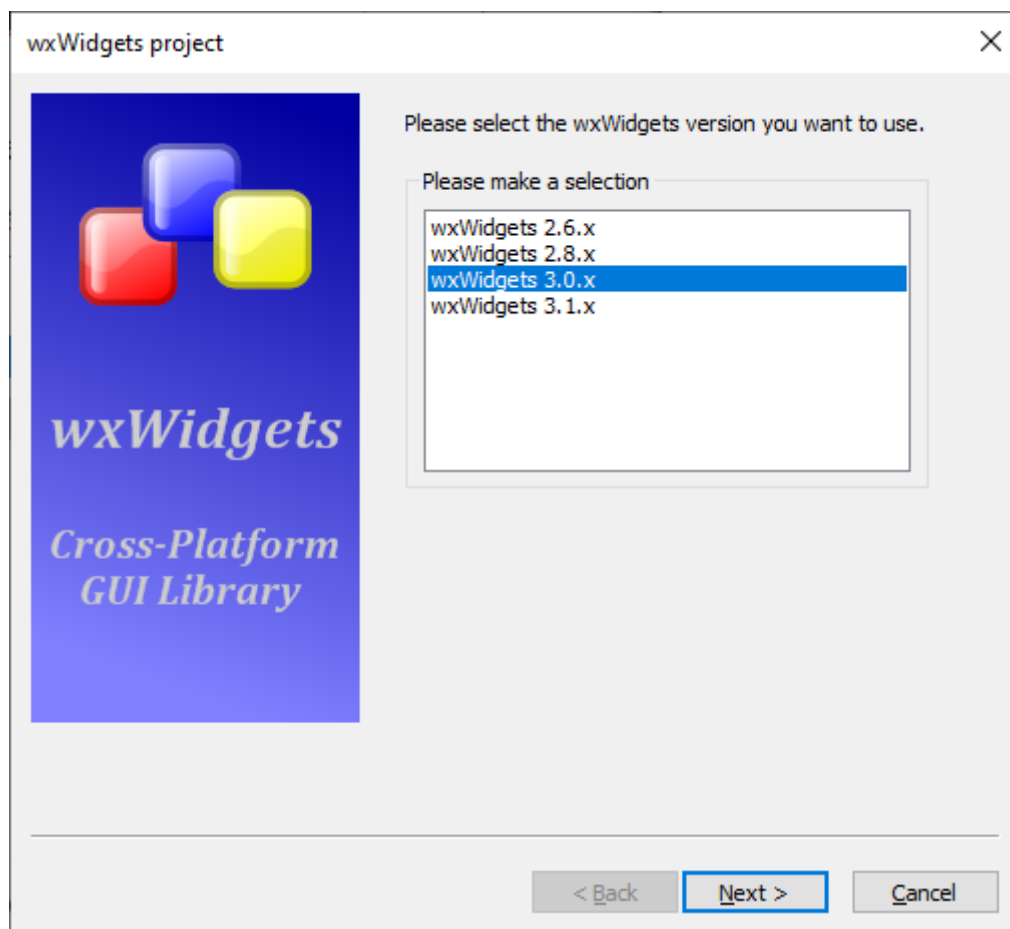


Рис.2.3.Выбор версии wxWidgets

3. Далее вводим название проекта и путь, где он будет создан.

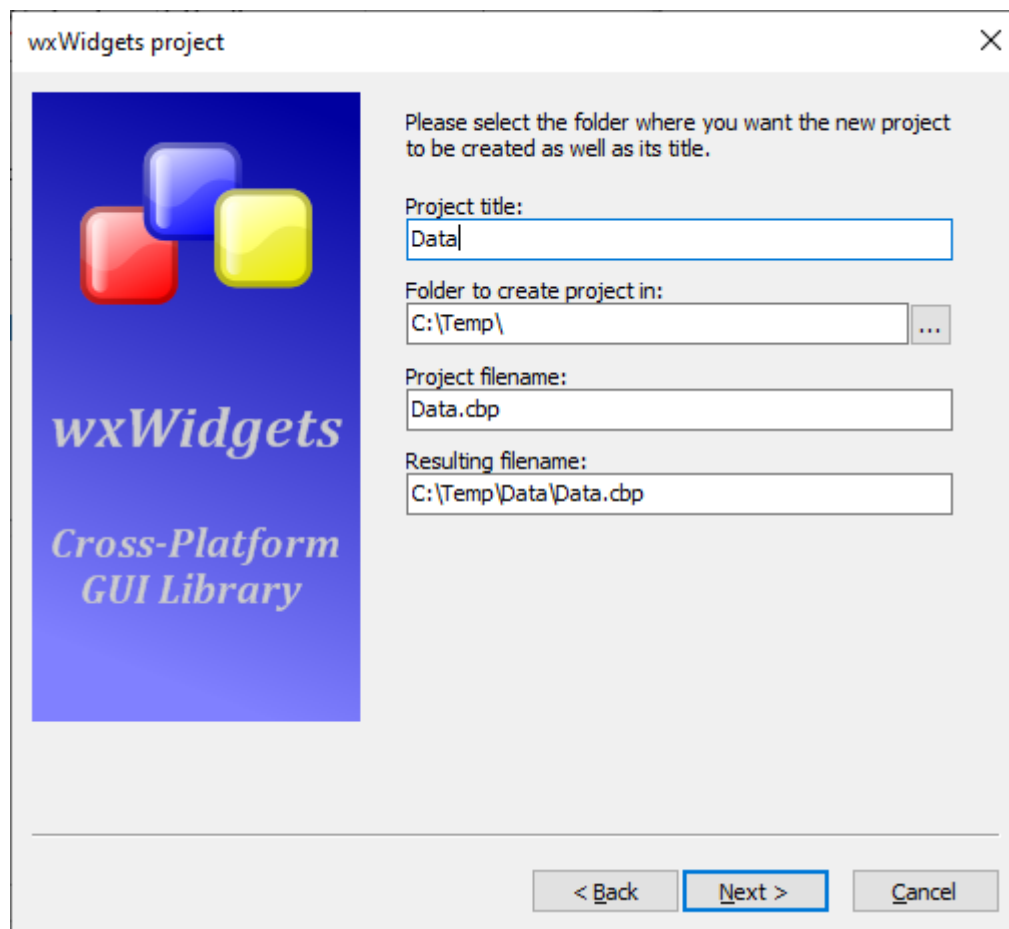
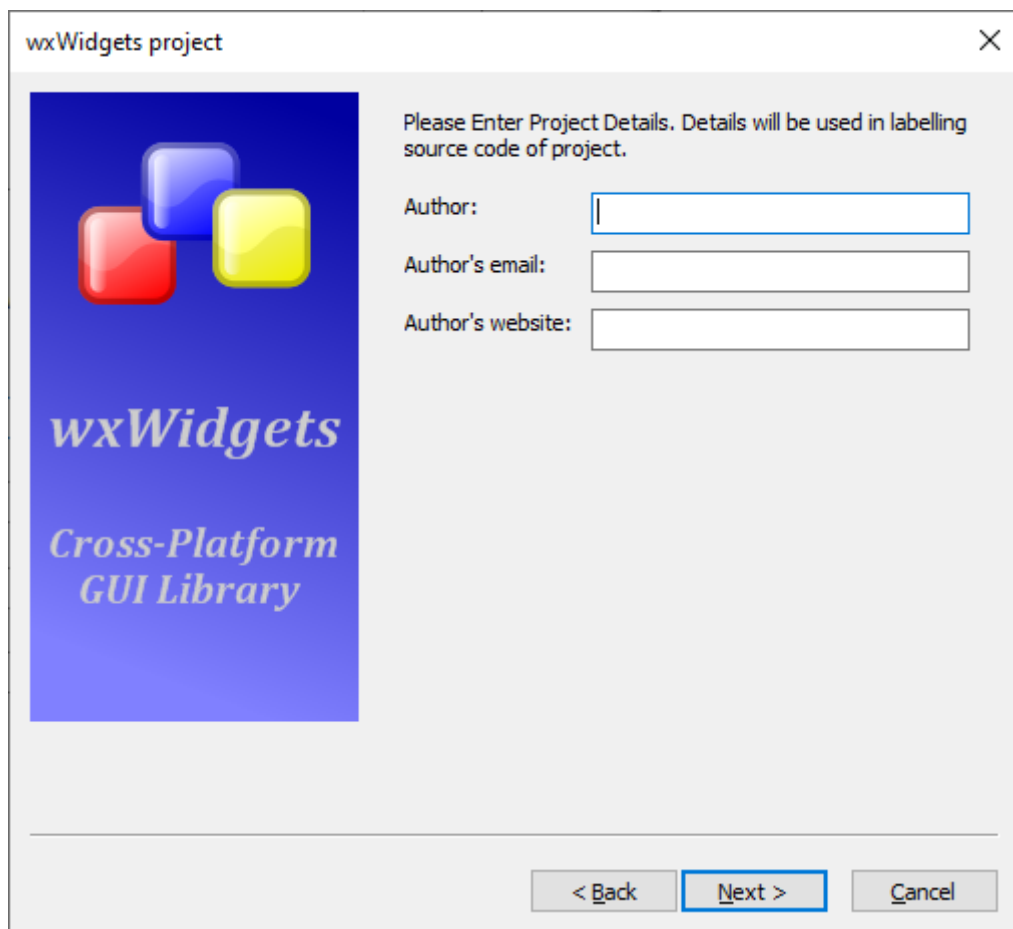


Рис.2.4. Выбор названия проекта и пути его расположения

4. Данные поля не обязательные, в них рассказывается об авторе проекта



The image shows a dialog box titled "wxWidgets project" with a close button (X) in the top right corner. On the left side, there is a blue rectangular area containing the wxWidgets logo (three colored squares: red, blue, and yellow) and the text "wxWidgets" in a large, bold, serif font, followed by "Cross-Platform" and "GUI Library" in a smaller, italicized serif font. On the right side, there is a text area with the instruction "Please Enter Project Details. Details will be used in labelling source code of project." Below this instruction are three input fields: "Author:", "Author's email:", and "Author's website:". At the bottom right of the dialog box, there are three buttons: "< Back", "Next >" (which is highlighted with a blue border), and "Cancel".

Рис.2.5. Данные об авторе проекта

5. Затем выбираем инструмент, с помощью которого мы будем создавать интерфейс (группа переключателей "Preferred GUI Builder").

В этом же окне выбирается тип приложения ("Application Type"): главное окно будет диалогом ("Dialog Based") или полноценным окном ("Frame Based"). Первый тип подходит для небольших приложений, при этом у диалога не может быть строки статуса и главного меню.

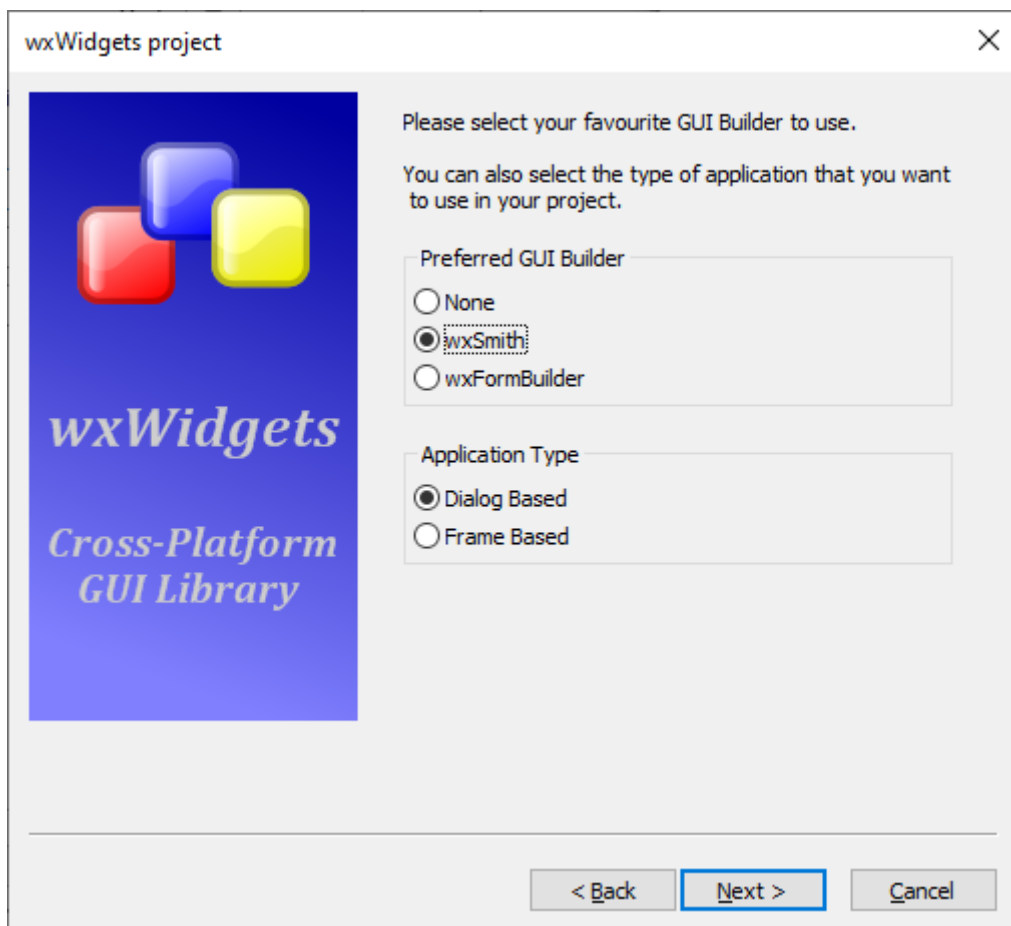


Рис.2.6. Выбор инструмента и типа приложения



6. В следующем окне нужно указать путь до библиотеки wxWidgets, куда вы распаковывали скачанный архив.

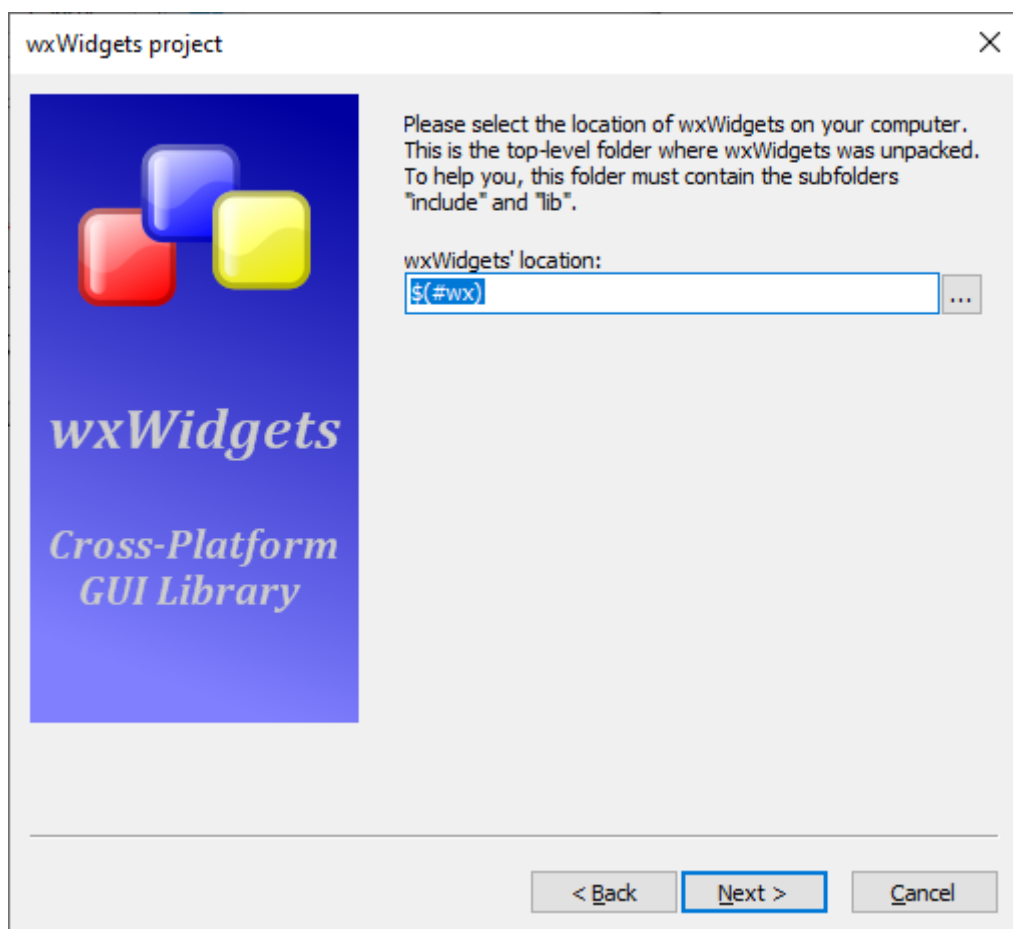


Рис.2.7. Указание пути расположения библиотеки wxWidgets

7. Затем нужно установить (или подтвердить) некоторые параметры для компиляции - тип компилятора и папки для отладочной и релизной версии сборки нашего проекта.

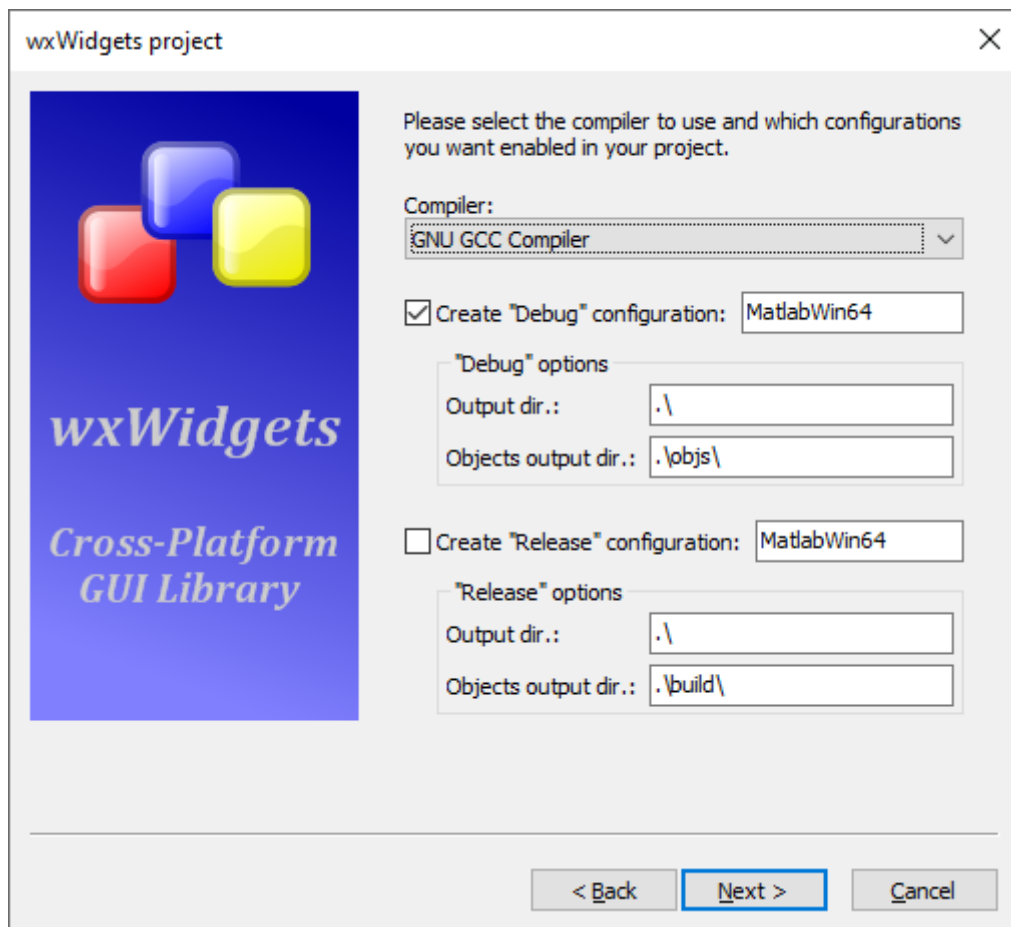


Рис.2.8. Параметры компиляции

8. В этом окне, необходимо установить параметры, с которыми была скомпилирована библиотека wxWidgets

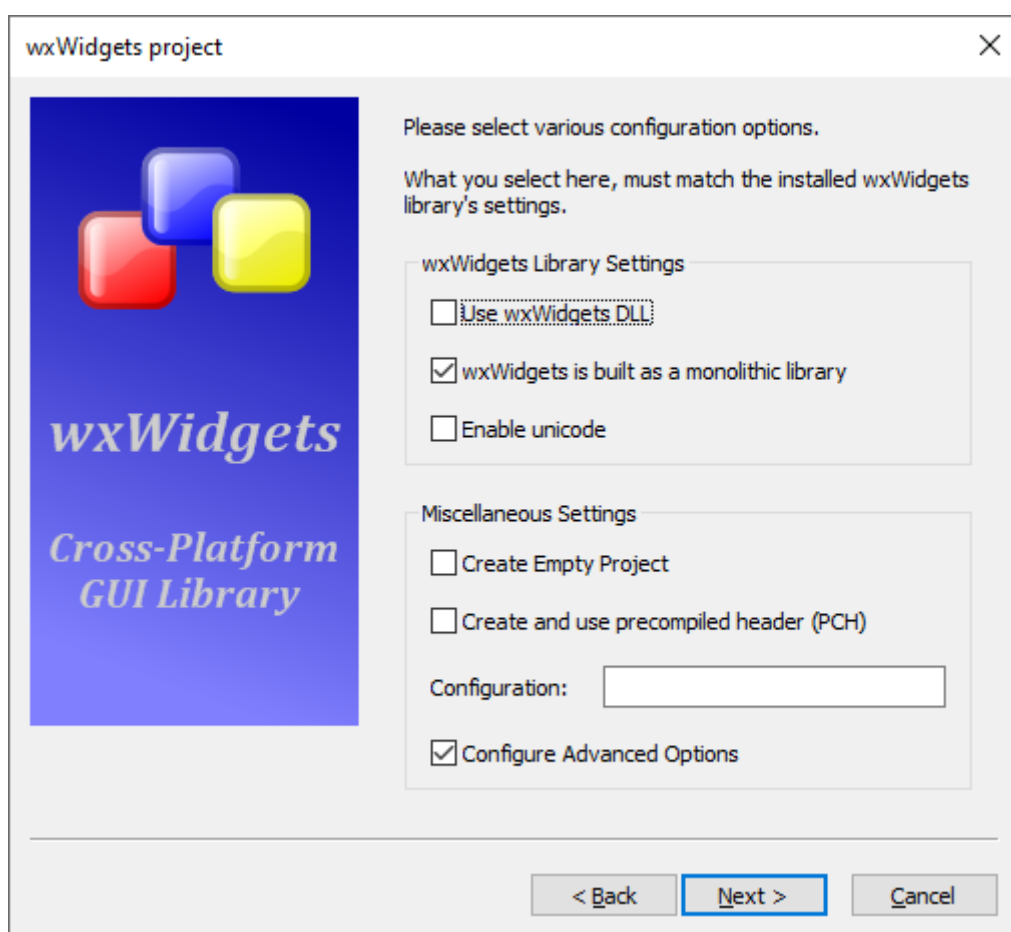


Рис.2.9. Параметры с которыми была скомпилирована библиотека wxWidgets

9. После этого мастер создания проекта завершит свою работу, и будет создан новый проект

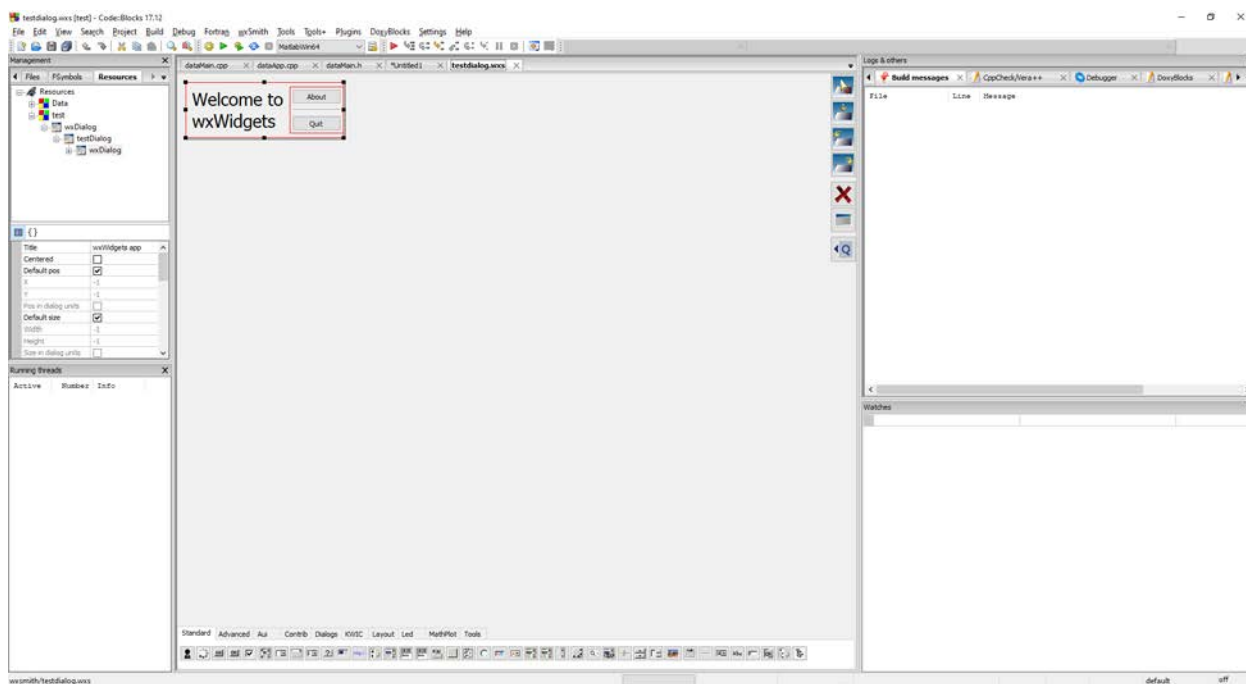


Рис.2.10. Пример собранного проекта

## 2.2. Необходимые файлы для сборки проекта

К отчету прикреплены 7 приложений, в которых содержатся основные файлы кода программы, необходимые для сборки проекта. Так файл: “dataMain.cpp”, код описан в приложение 1, 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 2, 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 3, содержит в себе обработчик массива SISerr, для использования этого массива в приложение 1, данный обработчик необходим, так как приложение 2 написано на языке «C», а приложение 1 на языке «C++», путь к файлу: “...\data\parser.h”.

Файл: “data.cbp”, код описан в приложение 4 – 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 5 – это заголовочный файл, в котором хранятся применяемые классы и методы, путь к файлу: “...\data\dataMain.h”.

Файл: “parser.h”, код описан в приложение 6, содержит в себе метод обработки массива из «C» файла, путь к файлу: “...\data \ parser.h”.

Файл: “datadiaslog.wxs”, код описан в приложение 7 – это файл описания графического пользовательского интерфейса для плагина wxSmith, путь к файлу: “...\data\ wxsmith\datadiaslog.wxs”.

## 2.3.Изменения по коду

Полный алгоритм программы описан в отчете НИР за 7 семестр, в данном отчете рассмотрим лишь ключевые изменения, внесенные в алгоритм.

В программе необходимо было переделать алгоритм вывода массива данных: выводить данные в таблицу. Ранее было реализован вывод в командную строку. Также необходимо было исправить ошибки по алгоритму загрузки файла.

### 2.3.1.Вывод массива данных

Для вывода массива данных воспользовался отдельным классом «wxGrid».

WxGrid и связанные с ним классы используются для отображения и редактирования табличных данных.

Они предоставляют богатый набор функций для отображения, редактирования и взаимодействия с различными источниками данных. Для простых приложений и для того, чтобы помочь вам начать работу, wxGrid - единственный класс, к которому вам нужно обращаться напрямую. Он установит экземпляры по умолчанию других классов и будет управлять ими за вас. Для более сложных приложений вы можете получить свои собственные классы для пользовательских видов сетки, таблиц данных сетки, редакторов ячеек и средств визуализации. WxGrid Обзор имеет примеры простых и более сложных приложения, объясняет взаимосвязь между различными классами сетки и имеет краткое сочетание клавиш и функции мыши, предоставляемое wxGrid.

Чтобы воспользоваться данным классом, сначала в заголовке программы необходимо добавить данную библиотеку `#include "wx/grid"`

Далее придумываем название нашей таблицы, например, grid

И делаем указатель на класс wxGrid: `wxGrid* Grid;`

Чтобы создать форму, необходимо прописать окно в котором оно будет находиться, идентификатор ID, местоположение таблицы, ее размеры, стиль и имя.

Пример: `Grid = new wxGrid(SashWindow1, ID_GRID, wxPoint(33,72), wxSize(244,sizeY), 0, _T("ID_GRID"));`

Чтобы создать сетку с nr- количество строк, nc- количество столбцов, необходимо прописать функцию: `Grid->CreateGrid(nr,nc)`, где в моем случае nr- количество спутников определенной НС.

Для подписи столбцов воспользуемся функция `SetColLabelValue`, например подпишем 0-ой столбец, как SISRE, а 1-ый, как SISVE:

```
Grid->SetColLabelValue(0, _T("SISRE, m"));
```

```
Grid->SetColLabelValue(1, _T("SISVE, mm/s"));
```

Далее заполним таблицу значениями SISRE, SISVE для НС с количеством спутников равных `max_sats`:

```
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));
}
```

`wxString` используется для преобразования числовых значений в строки.

Самая большая сложность использования класса `wxGrid` состояла в том, что по умолчанию устанавливаются размеры таблицы и далее, чтобы их изменять для определенной НГСС нельзя будет просто обратиться к классу, а приходилось в цикле обработки удалять и создавать новую таблицу.

### 2.3.2.Исправления ошибок при скачивании файла

Часто программа давала сбой при скачивании файла с сервера из-за того, что файл, который скачивался уже хранился в папке, для этого был написан алгоритм, который перед скачиванием файла проверял, существует ли такой файл в папке, если да, то удалял его, если нет, то продолжал дальнейший цикл программы. Пример кода, который выполняет алгоритм: «Проверка если такой файла в папке программы «MERMS-RSC\_C.ete», если да, то удаляет его: `wxRemoveFile(file)`, если нет, то продолжения дальнейшего цикла»:

Для этого воспользуемся классом `wxTextFile`. Этот класс позволяет работать с текстовыми файлами построчно и понимает символ переноса строки для различных систем, позволяя преобразовывать файл из одной системы в другую.

Класс не позволяет работать с большими файлами (порядка 1 Мб и более), так как считывает содержимое всего файла в память, однако с небольшими файлами она справляется.

Воспользуемся функцией `Bool Exists()` – возвращает `true`, если файл существует, имя файла должно быть указано в конструкторе перед вызовом функции `Exists()`.

Пример кода:

```
wxTextFile file11(wxT("MERMS-RSC_C.ete"));  
    if (file11.Exists())  
    {wxRemoveFile(file);  
    }
```



### 2.3.3. Выведение окна ошибки при неудачном скачивании

Для функции `download` поменяли тип функции на `bool`. Это стало необходимо для более адекватной работы программы путем введения дополнительного окна об ошибке, когда не удастся скачать файл.

Тип `bool` - тип данных, принимающий два возможных значения, иногда называемых истиной (`true`) и ложью (`false`). Присутствует в подавляющем большинстве языков программирования как самостоятельная сущность или реализуется через численный тип данных. В некоторых языках программирования за значение истина полагается 1, за значение ложь — 0.

Далее перед тем, как загрузить таблицу добавим цикл проверки на скачивание файла. Для этого обозначим новую переменную, `bool down = download( "glonass-iac.ru", NULL, NULL, File1, file);`

При правильном выполнении функции будет возвращаться значение `true`, тогда добавим проверку: если значение `down == false`, тогда выдадим ошибку.

Пример реализации:

```
if (!down)
{
    wxMessageBox(_("Error"), _("Error"));
    return;
}
```

В C++ использование `!down` равносильно `down == false`.

Если `down == true`, то цикл программы будет продолжаться.

## ГЛАВА 3.РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

### *Минимальные требования:*

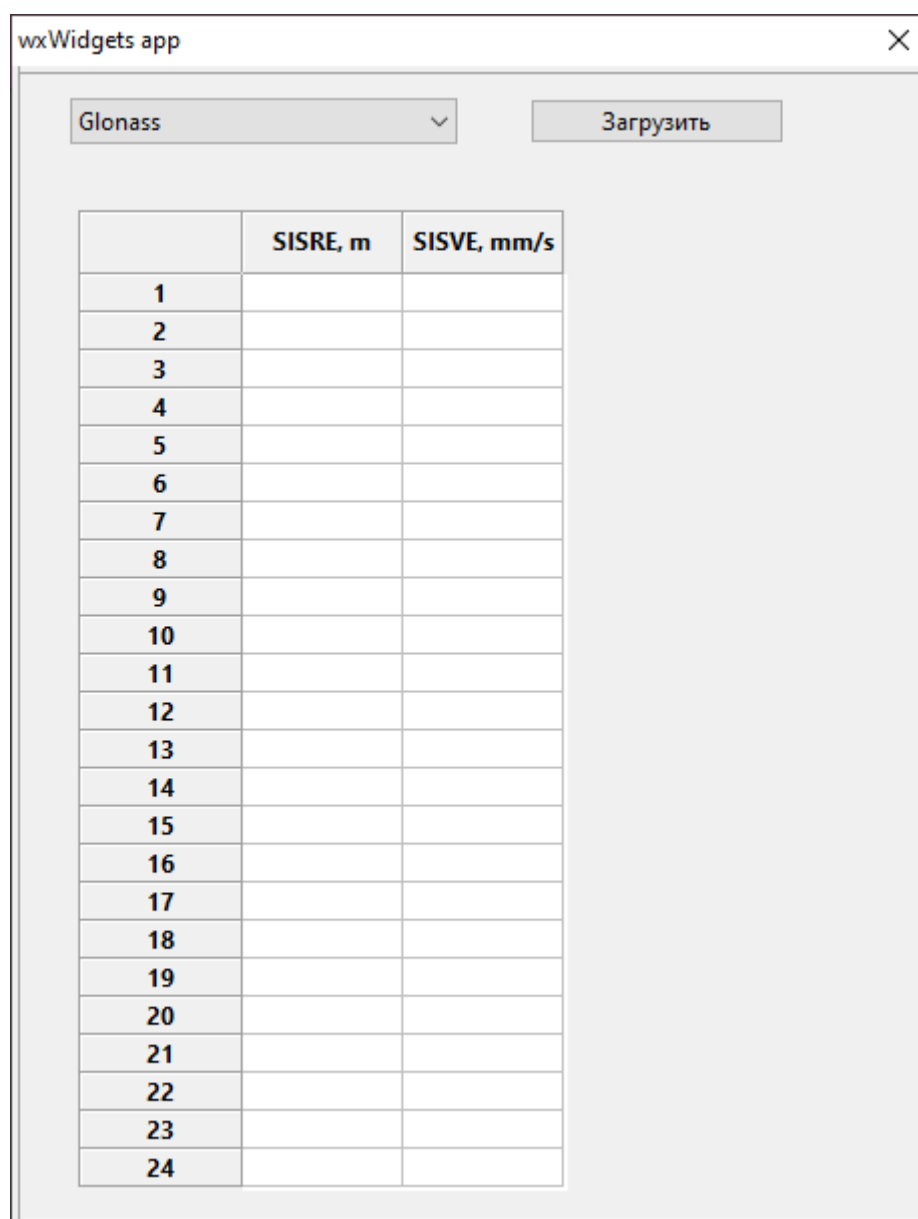
Для запуска программы необходимо иметь windows 7/10 и подключенное устройство к интернету.

### *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 "wxWidgets app". Inside the window, there is a dropdown menu currently showing "Glonass" with a downward arrow. To the right of the dropdown is a button labeled "Загрузить". Below these elements 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".

	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		

Рис.3.1. Интерфейс программы

2. Выберите необходимую вам НС:

wxWidgets app

Глонасс

Глонасс

GPS

Galileo

Beidou

QZSS

Загрузить

/E, mm/s

2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		

Рис.3.2. Интерфейс выбора НС

3. Нажмите кнопку «Загрузить»

The screenshot shows a window titled "wxWidgets app" with a close button (X) in the top right corner. Inside the window, there is a dropdown menu currently set to "Glonass" and a button labeled "Загрузить" (Load). Below these elements is a table with 24 rows. The first column of the table contains numbers from 1 to 24. The second column is labeled "SISRE, m" and the third column is labeled "SISVE, mm/s". All cells in the second and third columns are currently empty.

	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		

Рис.3.3. Интерфейс выбранной НС

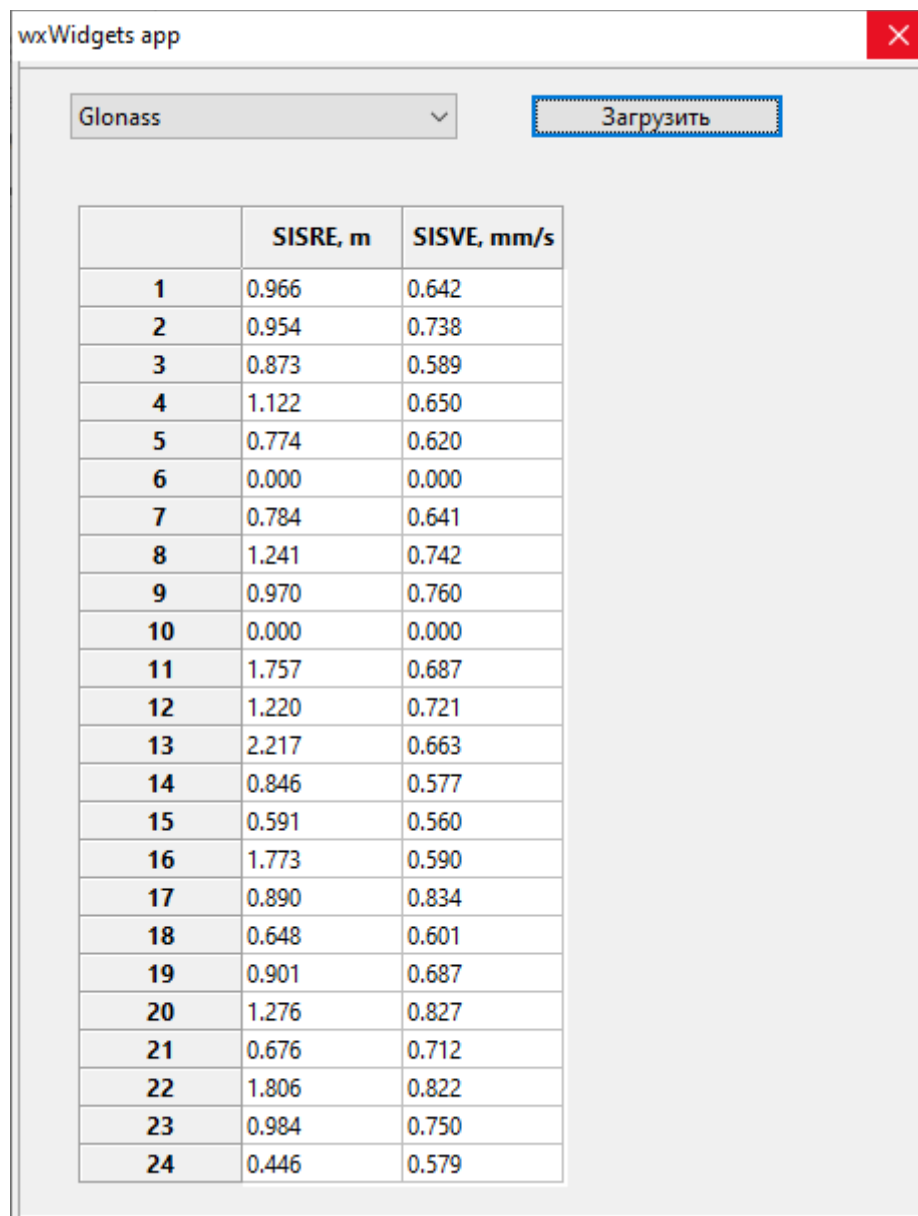
Получили значения SISRE и SISVE для каждого спутника, если значения равны 0.00, то данный спутник отсутствует.

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 "wxWidgets app" with a red close button in the top right corner. Inside the window, there is a dropdown menu set to "Glonass" and a button labeled "Загрузить" (Load). Below these is a table with 24 rows of data. The table has three columns: an index from 1 to 24, "SISRE, m", and "SISVE, mm/s".

	SISRE, m	SISVE, mm/s
1	0.966	0.642
2	0.954	0.738
3	0.873	0.589
4	1.122	0.650
5	0.774	0.620
6	0.000	0.000
7	0.784	0.641
8	1.241	0.742
9	0.970	0.760
10	0.000	0.000
11	1.757	0.687
12	1.220	0.721
13	2.217	0.663
14	0.846	0.577
15	0.591	0.560
16	1.773	0.590
17	0.890	0.834
18	0.648	0.601
19	0.901	0.687
20	1.276	0.827
21	0.676	0.712
22	1.806	0.822
23	0.984	0.750
24	0.446	0.579

Рис.3.4. Полученные значения

## ЗАКЛЮЧЕНИЕ

Теоретически рассмотрел влияние ошибок SISRE на оценку координат потребителя, выяснил, что на диагонали матрицы  $\mathbf{D}_x$  будут лежать дисперсии компонент вектора состояний  $\mathbf{x}$ .

Все поставленные задачи были выполнены: переделаны алгоритмы скачивания и добавлен свод данных в таблицу.

Так же в отчете рассмотрел настройку программного обеспечения для написания программы на универсальной платформе Windows, разобраны изменения ошибок в алгоритме программы и добавление нового алгоритма для определенных задач, а также написана 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](http://www.glonass-iac.ru)» »

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

[5]. Официальный сайт MinGW: «<http://www.mingw.org>»

[6]. Официальный сайт WxWidgets: « <http://www.wxwidgets.org>»

# ПРИЛОЖЕНИЕ

## Приложение 1

### *dataMain.cpp*

```
#include "dataMain.h"
#include <wx/msgdlg.h>
#include <windows.h>
#include <wininet.h>
#include <iostream>
#include "parser.h"
#include <string>
#include <stdio.h>
#include <wx/string.h>
#include <wx/textfile.h>
#include <wx/dialog.h>
#include <wx/intl.h>
#include <wx/settings.h>
using std::cout;
using std::cin;
using std::endl;
//(*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;
}

const long dataDialog::ID_CHOICE1 = wxNewId();
const long dataDialog::ID_BUTTON2 = wxNewId();
const long dataDialog::ID_NOTEBOOK1 = 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;
    HINTERNET 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)
{
    Create(parent, wxID_ANY, _("wxWidgets app"), wxDefaultPosition, wxDefaultSize, wxDEFAULT_DIALOG_STYLE,
    _T("wxID_ANY"));
    SetClientSize(wxSize(457,579));
    SashWindow1 = new wxSashWindow(this, ID_SASHWINDOW1, wxPoint(56,40), wxSize(408,520),
    wxSW_3D|wxCLIP_CHILDREN, _T("ID_SASHWINDOW1"));
    Choice1 = new wxChoice(SashWindow1, ID_CHOICE1, wxPoint(29,16), wxSize(193,96), 0, 0, 0, wxDefaultValidator,
    _T("ID_CHOICE1"));
    Choice1->SetSelection( Choice1->Append(_("Glonass")) );
    Choice1->Append(_("GPS"));
    Choice1->Append(_("Galileo"));
    Choice1->Append(_("Beidou"));
    Choice1->Append(_("QZSS"));

```



```

    Button2 = new wxButton(SashWindow1, ID_BUTTON2, _("Загрузить"), wxPoint(258,16), wxSize(127,23), 0, wxDefaultValidator,
    _T("ID_BUTTON2"));
    Notebook1 = new wxNotebook(SashWindow1, ID_NOTEBOOK1, wxPoint(124,214), wxDefaultSize, 0, _T("ID_NOTEBOOK1"));
    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_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() );
/*
    s.Printf("=====\n");
    int imax1=5;
    for (int i=0; i<imax1; i++)
    {
        s << wxString::Format(L"SISRE = %.3f\t\n", SISerr[i].SISRE);
    }
    for (int i=0; i<imax1; i++)
    {
        s << wxString::Format(L"SISVE = %.3f\t", SISerr[i].SISVE);
    }
    // s.Printf("=====\n");
    //s << wxString::Format(L"GDOP = %.2f\t", SISerr[i].SISRE);
    //s << wxString::Format(L"GDOP = %.2f\t", i);
    // s << wxString::Format(L"VDOP = %.2f\t", SISVE);

    StaticText1->SetLabel(s);
    */
}
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::OnButton2Click(wxCommandEvent& event)
{
    //int n=5;
    wxString s;
    //while (!Close())
    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);
        }
    }
    // n++;
}

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;
    }
    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));
    }
    for (k=0; k<max_sats; k++ )
    {
        SISerr[k].SISRE=0;

```

```

    SISerr[k].SISVE=0;
}
}

```

## Приложение 2

### *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.etc", "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, "%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 3

### *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 4

### *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="data.cbp" />
<Unit filename="dataApp.cpp" />
<Unit filename="dataApp.h" />
<Unit filename="dataMain.cpp" />
<Unit filename="dataMain.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" />
<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 5

### *dataMain.h*

```

#ifndef dataMAIN_H
#define dataMAIN_H

```

```

//(*Headers(dataDialog)

#include <wx/button.h>
#include <wx/choice.h>
#include <wx/dialog.h>
#include <wx/notebook.h>
#include <wx/sashwin.h>

//*)

#include <wx/grid.h>
//#include <wx/gdicmn.h>

class dataDialog: public wxDialog
{
public:

    dataDialog(wxWindow* parent,wxWindowID id = -1);
    virtual ~dataDialog();

private:
    //(*Handlers(dataDialog)
    void OnQuit(wxCommandEvent& event);
    void OnAbout(wxCommandEvent& event);
    void OnChoice1Select(wxCommandEvent& event);
    void OnSashWindow1SashDragged(wxSashEvent& event);
    void OnChoice1Select12(wxCommandEvent& event);
    void OnChoice1Select2(wxCommandEvent& event);
    void OnButton1Click(wxCommandEvent& event);
    void OnButton2Click(wxCommandEvent& event);
    void OnComboBox1Selected(wxCommandEvent& event);
    void OnButton3Click(wxCommandEvent& event);
    void OnChoice1Select1(wxCommandEvent& event);
    void OnTextCtrl1Text(wxCommandEvent& event);
    void OnInit(wxInitDialogEvent& event);
    void OnGrid1CellLeftClick(wxGridEvent& event);
    void OnChoice1Select3(wxCommandEvent& event);

    //*)

    void Gridd(const char* file);

    //(*Identifiers(dataDialog)
    static const long ID_CHOICE1;

```

```

static const long ID_BUTTON2;

static const long ID_NOTEBOOK1;

static const long ID_SASHWINDOW1;

/*)

static const long ID_GRID;

/*(*Declarations(dataDialog)

wxButton* Button2;

wxChoice* Choice1;

wxNotebook* Notebook1;

wxSashWindow* SashWindow1;

/*)

wxGrid* Grid;

DECLARE_EVENT_TABLE()

};

```

## Приложение 6

### *parser.h*

```

#ifndef PARSER_H
#define PARSER_H

#ifdef __cplusplus
extern "C" {
#endif

typedef struct
{
    char systype;

    double SISRE;

    double SISVE;

    // double imax;

} data_t;

extern data_t SISerr[75];//

int parse(const char* file);

#ifdef __cplusplus
}
#endif

#endif

```

## Приложение 7

*wxsmith\datadiaslog.wx*

```
#ifndef PARSER_H
#define PARSER_H

#ifdef __cplusplus
extern "C" {
#endif

typedef struct
{
    char systype;
    double SISRE;
    double SISVE;
    // double imax;
} data_t;

extern data_t SISerr[75];

int parse(const char* file);

#ifdef __cplusplus
}
#endif
#endif
```