**Модуль по отображению электронной картографической информации от обеспечивающих систем корабля на «Систему отображения », при применении изделий по различным объектам.**

**Техническое задание**

2017

Оглавление

[1. Введение 3](#_Toc485069434)

[2. Основания для разработки 3](#_Toc485069435)

[3. Назначение разработки 3](#_Toc485069436)

[4. Требования к программному изделию 3](#_Toc485069437)

[4.1. Требования к функциональным характеристикам 3](#_Toc485069438)

[4.2. Требования к надежности 3](#_Toc485069439)

[4.3. Условия эксплуатации 4](#_Toc485069440)

[4.4. Требования к составу и параметрам технических средств 4](#_Toc485069441)

[4.5. Требования к информационной и программной совместимости 4](#_Toc485069442)

[4.6. Требования к маркировке и упаковке 4](#_Toc485069443)

[4.7. Требования к транспортированию и хранению 4](#_Toc485069444)

[5. Требования к программной документации 4](#_Toc485069445)

[6. Технико-экономические показатели 4](#_Toc485069446)

[7. Стадии и этапы разработки 4](#_Toc485069447)

# Введение

Разрабатываемое программное обеспечение «Модуль по отображению электронной картографической информации от обеспечивающих систем корабля», состоящая из двух программ: «Картографическое ядро» и «Модуль графического отображения карт». В прикладную программу встроена авторизация с удаленным ресурсом, на котором содержатся необходимый массив карт. По необходимости происходит запрос к ресурсу. Далее осуществляется передача карт ядру для их дальнейшего отображения. Программное обеспечение написано на C, для его работы необходимо сетевое подключение.

# 2. Основания для разработки

Документ для разработки – учебный план СПбПУ

Организация, утвердившая документ – СПбПУ

Наименование разработки – “ Модуль по отображению электронной картографической информации от обеспечивающих систем корабля на «Систему отображения », при применении изделий по различным объектам.”

# 3. Назначение разработки

Программное изделие предназначено для отображения электронной картографической информации от обеспечивающих систем корабля.

# 4. Требования к программному изделию

## 4.1. Требования к функциональным характеристикам

Программное обеспечение «Картографическое ядро» должно осуществлять следующие функции:

* Отвечать на сетевые запросы модуля графического отображения карт в соответствии с протоколом;
* Проверять корректность полученных от клиентов данных и от сервера хранения карт;
* Осуществлять запись в разделяемую область памяти по запросу от клиента.
* Обрабатывать только одного клиента
* Воспринимать команды модуля графического отображения карт в соответствии с протоколом.

Программное обеспечение «Модуль графического отображения карт» должно осуществлять следующие функции:

* Отображать на экране картографическую информацию в соответствии со своими запросами к картографическому ядру.
* При первом запуске запрашивать картографическую информацию, для того чтобы картографическое ядро могло заранее подготовить информацию.
* Отвечать на сетевые запросы картографического ядра в соответствии с протоколом;
* Воспринимать команды картографического ядра в соответствии с протоколом.

Входными данными для изделия являются:

* Входными данными для картографического ядра является массив карт в формате S-57, а так же команды отданные модулем графического отображения карт.
* Входными данными для модуля графического отображения карт являются разделяемая область памяти в которую записывается массив необходимых данных для графического отображения.

Выходным данными для изделия являются:

* Выходными данными для картографического ядра являются разделяемая область памяти в которую записывается массив необходимых данных для графического отображения.
* Выходными данными для модуля графического отображения карт является вывод графической информации карт формата S-57 на экран.

## 4.2. Требования к надежности

Программное изделие должно обеспечивать следующие параметры надежности:

* При недоступности сервера хранения карт, прикладное приложение должно корректно завершаться с выводом ошибки на экран.
* В случае если ядро не нашло ожидаемые карты в указанной директории должно корректно завершаться с выводом сообщения об ошибке.
* Картографическое ядро должно иметь возможность обслуживания только одним клиентом.
* При отсутствии подключения приложение должно корректно завершать сеанс клиента.

## 4.3. Условия эксплуатации

Для корректного функционирования программного изделия необходимо наличие персонального компьютера с QNX6.

## 4.4. Требования к составу и параметрам технических средств

Для возможности функционирования изделия в запущенной операционной системе необходимо наличие на сервере базы данных со списком береговых карт формата S-57.

## 4.5. Требования к информационной и программной совместимости

Программный продукт совместим с QNX6 и не требует установки дополнительных средств.

## 4.6. Требования к маркировке и упаковке

Программное изделие не требует какой-либо маркировки или упаковки.

## 4.7. Требования к транспортированию и хранению

Программное изделие должно храниться и транспортироваться на электронных носителях, обеспечивающих неизменность исходных кодов изделия.

# 5. Требования к программной документации

В состав программной документации должны входить:

• Спецификация

• Техническое задание

• Описание программы

• Руководство системного программиста

• Текст программы

• Программа и методика испытаний

Документация должна подробно и полно описывать функциональные характеристики изделия, его возможности, способы его инсталляции и модификации.

# 6. Технико-экономические показатели

Эффективность программного изделия напрямую зависит от той аппаратной среды, на которой оно функционирует. В частности важнейшим фактором производительности является частота процессора, с которой он может выполнять задачи.

**7. Алгоритм работы**

Картографическое ядро при получении одного из признаков осуществляет подготовку результирующей карты для отрисовки. После завершения загрузки и выбора вида и режима взаимодействия ядро регистрирует себя под именем «949\_55\_mapr1» для , устанавливает связь с необходимыми карт-серверами для выработки результирующей карты и посылает в модуль графического отображения карт сообщение о состоянии связи со всеми карт-серверами.

Далее проверка связи с карт-серверами осуществляется в соответствии с протоколом.

При первом после включения питания обращении модуля графического отображения карт с использованием картографического ядра, отправляет запрос на получение карты с карт-сервера. В запросе содержатся максимальная и минимальная широты и долготы района, охватывающего отображаемую область, на которую требуется карта. Получив от модуля графического отображения карт запрос на считывание карты ядро считывает карту и сохраняет ее для обработки последующих запросов от модуля графического отображения карт на подготовку карты к отрисовке.

После модуль графического отображения карт посылает в ядро запрос на подготовку карты согласно выбранного признака для отрисовки. Далее указанные запросы модуль графического отображения карт направляет в ядро по мере необходимости перерисовать содержимое окна с картой.

В указанном запросе содержатся географические координаты текущего положения носителя (точки старта), расстояния от текущего положения до нижней, левой, верхней и правой границ отображаемой области на экране, максимальная и минимальная широты и долготы района, охватывающего отображаемую область, на которую требуется карты для установленного признака, масштабирующий коэффициент. Получив запрос ядро формирует результирующую карту для отображения в окне, записывает ее в соответствии с форматом (приведен в настоящем протоколе) в разделяемую область памяти, создаваемую картографическим ядром, после чего сообщает процессу в модуль графического отображения карт, зарегистрированному под именем «949\_55\_paintR\_server» , о готовности результирующей карты для отрисовки.

Получив сообщения от картографического ядра о готовности результирующей карты для отрисовки модуль графического отображения карт считывает указанную результирующую карту из разделяемой области памяти и осуществляет отрисовку на экране.

При отсутствии результирующей карты с сервера хранения карт в течение не более 20 с ядро посылает в модуль графического отображения карт сообщение по указанному таймауту. В случае, если сервер хранения карт не предоставил всю запрошенную ядром информацию в течение 10 с ядра посылает в модуль графического отображения карт сообщение по указанному таймауту. Формат указанного сообщения определяется на этапе рабочего проектирования.

По мере движения носителя картографического ядра, анализируя поступающие от модуля графического отображения карт запросы на отрисовку результирующей карты, проводит получение с сервера хранения карт дополнительных элементов карт, обеспечивающих заполнение отображаемого на экране района, изменившегося ввиду перемещения носителя, и сохраняет их для последующей обработки.

Укрупненная логико-временная диаграмма обмена при запросе считывания карты:

- модуль графического отображения карт -> pulse-сообщения -> картографическое ядро

(получив pulse-сообщения от картографического ядра запрашивает данные от модуля графического отображения карт у процесса с именем «949\_55\_count\_traektR» )

- картографическое ядро -> Send() -> модуль графического отображения карт -> Reply() -> картографическое ядро

(получив в Reply() данные от модуля графического отображения карт картографическое ядро считывают карту и сохраняют для дальнейшего использования)/

Укрупненная логико-временная диаграмма обмена при запросах на подготовку карты к отрисовке:

- модуль графического отображения карт -> pulse-сообщения -> картографическое ядро

(получив pulse-сообщения картографическое ядро запрашивает данные от СО УКСУС у процесса с именем «949\_55\_count\_traektR»)

- картографическое ядро -> MsgSend() -> модуль графического отображения карт -> MsgReply() -> картографическое ядро

(получив данные от модуля графического отображения карт считывает сохраненные карты, преобразует их в пиксельные координаты и записывает в разделяемую область памяти для отрисовки и сообщает об этом в модуль графического отображения карт в процесс «949\_55\_paintR\_server»)

- картографическое ядро -> MsgSend() -> модуль графического отображения карт -> MsgReply() -> картографическое ядро

**8. Структура используемых qnx-сообщений**

Формат сообщения, отправляемого из картографического ядра в модуль графического отображения карт для информирования о состоянии связи c сервером хранения карт.

|  |  |
| --- | --- |
| Структура | Описание параметров |
| struct{  unsigned short type  unsigned short subtype  } \_sysmsg\_hdr;  typedef struct { |  |
| struct \_sysmsg\_hdr Hdr; | Hdr.type=0х1000U+1=4097=0x1001  Hdr.subtype = 610 |
| short check\_kks; | check\_kks – состояние связи с УКС:  0 – есть связи с УКС;  1 – нет связи с УКС. |
| } TYPE\_M\_MP\_PN\_CC; |  |

Формат структуры MsgPulse-сообщения, выдаваемой из, для картографического ядра структуры были реализованы ранее.

|  |  |
| --- | --- |
| Структура | Описание параметров |
| struct \_pulse pulse; | pulse.code = 88; |
|  |  |

Формат сообщения, отправляемого картографическим ядром в модуль графического отображения карт в ответ на полученное pulse-сообщение.

|  |  |
| --- | --- |
| Структура | Описание параметров |
| typedef struct { |  |
| struct \_sysmsg\_hdr Hdr; | Hdr.type=0х1000U+950=5046=0x13b6  Hdr.subtype = 574 |
| } TYPE\_P\_MP\_CT; |  |

Формат структуры возвращаемой функцией Reply() из модуль графического отображения карт в картографическое ядро в ответ на сообщения представлены ниже. Отдельный бит структуры (поле priznak) определяет, что требуется от картографического ядра: либо это запрос на получение карты, либо запрос на отрисовку карты.

|  |  |
| --- | --- |
| Структура | Описание параметров |
| typedef struct { |  |
| struct \_sysmsg\_hdr Hdr; | Hdr.type=0х1000U+950=5046=0x13b6  Hdr.subtype=574 |
| int reserv1; | reserv1 |
| int reserv2; | reserv2 – резерв |
| unsigned short priznak; | бит 1 (счет от 0) в слове priznak обозначает:  1 - запрос на получение карты  0 - запрос на отрисовку карты |
| unsigned short mainzona; | резерв |
| double Bst; | Bst – широта текущего положения носителя, грд. |
| double Lst; | Lst – долгота текущего положения носителя, грд. |
| double Xmin; | Xmin – координата X нижнего левого угла отображаемой карты в СК XOY, м |
| double Ymin; | Ymin – координата Y нижнего левого угла отображаемой карты в СК XOY, м |
| double Xmax; | Xmax – координата X верхнего правого угла отображаемой карты в СК XOY, м |
| double Ymax; | Ymax – координата Y верхнего правого угла отображаемой карты в СК XOY, м |
| double Bmin; | Bmin – широта нижнего левого угла отображаемой карты, грд. |
| double Lmin; | Lmin – долгота нижнего левого угла отображаемой карты, грд. |
| double Bmax; | Bmax – широта верхнего правого угла отображаемой карты, грд. |
| double Lmax; | Lmax – долгота верхнего правого угла отображаемой карты, грд. |
| double deltaangle; | резерв |
| double gk\_Xst; | резерв |
| double gk\_Yst; | резерв |
| double scale; | scale – масштаб отображаемой карты , пиксел/м |
| int ntraj[3]; | ntraj[0..2] - резерв |
| } TYPE\_M\_CT\_MP; |  |
|  |  |
| Примечание - Система координат XOY. Центр O находится в точке текущего положения носителя. Ось OX направлена на север. Ось OY направлена на восток (см. раздел 2). | |

Формулы, которые необходимо использовать при преобразовании геодезических координат геодезической системы координат в систему XOY и обратно, представлены ниже.

Координаты в пикселях отсчитываются от левого верхнего угла окна (центр пиксельной СК - OП) в направлении вправо (ось XП) и вниз (ось YП).

Формат сообщения, отправляемого картографического ядра в модуль графического отображения карт по результатам отработки запроса на отрисовку карты.

|  |  |
| --- | --- |
| Структура | Описание параметров |
| typedef struct { |  |
| struct \_sysmsg\_hdr Hdr; | Hdr.type=0х1000U+1=4097=0x1001  Hdr.subtype = 571 |
| } TYPE\_M\_MP\_PN; |  |

**9. Структура разделяемой области памяти**

Формат информации общей области памяти "mapshare2.ft", в которую сохраняется информация для отрисовки:

#define PAINTMAX 800000

unsigned short paint[PAINTMAX]

Paint[0] = 0

Paint[1] = 1

Paint[2..3] // - (unsigned long) число заполненных элементов массива, начиная с Paint[8]

Paint[4..5] // - резервные поля

Paint[6..7] // - (float) широта текущего положения носителя, град., полученная в запросе, грд.

Paint[8..9] // - (float) долгота текущего положения носителя, град., полученная в запросе, грд.

Paint[10..11] // - (float) координата левой границы окна относительно т. текущего положения носителя, м

Paint[12..13] // - (float) координата нижней границы окна относительно т. текущего положения носителя, м

Paint[14..15] // - (float) координата правой границы окна относительно т. текущего положения носителя, м

Paint[16..17] // - (float) координата верхней границы окна относительно т. текущего положения носителя, м

// Данные отображаемых объектов:

Paint[18..19] // - (unsigned long) число (K) отображаемых объектов (примитивов) в массиве

Paint[20..21] // - (unsigned long) порядковый номер 1-го отображаемого примитива (=1)

Paint[22] // - тип отображаемого примитива (1 - полилиния, 2 - многоугольник,

//3 - точка, 4 - окружность, 5 – дуга, 6 – текст и т.п.)

Paint[23] // - тип полилинии (границы многоугольника, окружности, дуги)

// 0 – граница не отображается

Paint[24] // - цвет полилинии (границы многоугольника, окружности, дуги)

// 0 – граница не отображается

Paint[25] // - толщина полилинии (границы многоугольника, окружности, дуги)

// 0 – граница не отображается

Paint[26] // - тип заливки в границах многоугольника (окружности) 0 – заливка отсутствует

Paint[27] // - резервное поле

Paint[28..29] // - (unsigned long) число (N1) точек полилинии (многоугольника), заранее определено:

// для точки -1 (Bp, Lp), для окружности – 2 (Bс, Lc, R, резерв),

// для дуги – 3 (Bс, Lc, R, Ab, Ae, резерв)

Paint[30] // - координата x 1-ой точки 1- го примитива, пиксели

Paint[31] // - координата y 1-ой точки 1-го примитива, пиксели   
 // …

Paint[30+2\*N] // - координата x N1-ой точки

Paint[31+2\*N] // - координата y N1-ой точки

Paint[M1.. M1+1] // - M1 = 30+2\*N+3 порядковый номер 2-го отображаемого примитива (=2) и далее аналогично тому, как представлена информация для 1-го примитива

..

**10. Исходные данные по привязке системы координат к системе координат СО УКСУС с СПО ГКИ ЭКИ.**

Система координат XOY. Центр O находится в точке текущего положения носителя. Ось OX направлена на север. Ось OY направлена на восток.

Формулы, которые необходимо использовать при преобразовании геодезических координат геодезической СК в систему XOY и обратно, представлены в разделе 5.

**X’’**

**X’’’**

**Y’’**

**(X’’min,Y’’min)**

(**X’’max,Y’’max)**

**Y’’’**

Точка местоположения носителя

Точка старта

**11. Исходные данные по работе с «Системой отображения» при выборе М55**

Необходимо, чтобы «Системой отображения М55» была обеспечена возможность работать с картами в диапазоне запрашиваемой области 1000\*1000 км с центром нахождения носителя, а также запрашиваемой области диапазоне 5км\*5км с центром объекта в конечной точке. Результирующая карта должна получаться из СПО ГКИ и отображаться на СО УКСУС.

**12. Исходные данные по работе с «Системой отображения » при выборе М55мод**

Необходимо, чтобы «Системой отображения М55мод» была возможность работать с картами в диапазоне запрашиваемой области 2000\*2000 км с центром нахождения носителя, а также запрашиваемая область в диапазоне 5км\*5км в конечной точке. Результирующая карта должна получаться из картографического ядра и отображаться на модуль графического отображения карт.

**13. Исходные данные для преобразования картографической информации из картографического ядра в используемую для работы модуля графического отображения карт.**

Определение дальности и пеленга по географическим координатам  
 (P; D) = F1(φST; λST; φ; λ)

Входные данные: географические координаты точки (), пеленг и дальность, которые подлежат определению, и географические координаты точки, относительно которой дальность и пеленг определяются ().

Выходные данные: дальность и пеленг .

(1.1)

(1.2)

(1.3)

(1.4)

(1.5)

(1.6)

(1.7)

(1.8)

(1.9)

(1.10)

(1.11)

(1.12)

(1.13)

(1.14)

(1.15)

(1.16)

(1.17)

(1.18)

(1.19)

(1.20)

(1.21)

(1.22)

(1.23)

Определение географических координат по дальности, боковому отклонению и пеленгу (φ; λ) = F'1(φST; λST; P; Z; D)

Входные данные: пеленг , дальность и боковое отклонение точки, географические координаты которой подлежат определению, географические координаты точки, относительно которой определены дальность и пеленг ().

Выходные данные: географические координаты ().

(1.24)

(1.25)

(1.26)

(1.27)

(15.28)

.(1.29)

(1.30)

(1.31)

(1.32)

(1.33)

(1.34)

(1.35)

(1.36)

(1.37)

(1.38)

Определение дальности и бокового отклонения по географическим координатам и пеленгу (D; Z) = F2(φST; λST; φ; λ; P)

Входные данные: пеленг , географические координаты центра полярной системы координат, относительно которой рассчитан пеленг  
() и цели ().

Выходные данные: дальность и боковое отклонение .

(1.39)

(1.40)

(1.41)

(1.42)

(1.43)

(1.44)

(1.45)

(1.46)

(1.47)

(1.48)

(1.49)

(1.50)

(1.51)

(1.52)

(1.53)

(1.54)

(1.55)

Определение дальности и пеленга по линейным координатам   
(P; D) = F2(X; Y):

P = arctg(Y/X), D = √(X2+Y2).

Определение линейных координат по дальности и пеленгу (X; Y) = F'2(P; D): X = D · cos(P), Y = D · sin(P).

# 8. Стадии и этапы разработки

Разработка программного изделия включает в себя три этапа:

1. Разработка и утверждение технического задания и спецификации программного изделия – целью данного этапа является выработка функциональных и нефункциональных требований к программному изделию и утверждение состава документации программного изделия.

2. Проектирование и разработка программного изделия – целью этапа является создание программного изделия и утвержденной документации. Разработка методов для сетевого взаимодействия, методов сбора информации о системе, методов отображения графической информации на экране. В разработку также включено тестирование программного изделия.

3. Сдача программного изделия – целью этапа является сдача разработанного программного изделия заказчику – передача исходных кодов и документации по изделию.

Срок разработки программного изделия – с 11.02.2017 по 14.06.2017, исполнитель – студент СПбПУ ИКНТ КСПТ Шаляпин Н.С.