**\**

**НТТУ**

**Киевский политехнический институт**

**ИПСА**

**Кафедра СП**

**Расчетно-графическая работа по курсу:**

**«Методы и инструментарий проектирвания елементной бази информационных технологий»**

|  |  |
| --- | --- |
| **Принял:**  Доц. Стиканов В.Е. | **Выполнил:**  студент  гр. ДА-71мн  Колинько Анжела  . |

**Киев 2017**

# Содержание

[1. Тема работы 3](#_Toc502107949)

[2. Цель работы 3](#_Toc502107950)

[3. Задание на выполнение работы 3](#_Toc502107951)

[4. Краткие теоретические сведения 4](#_Toc502107952)

[5. Расчеты 6](#_Toc502107954)

[6. Эксплуатационные документы](#_Toc502107959) 8

[6.1. Руководство системного программиста 8](#_Toc502107961)

[6.2. Руководство оператора 12](#_Toc502107963)

# 1. Тема работы

Расчет физико-топологических и конструктивных параметров n-канального МОП транзистора.

# 2. Цель работы

Получить навыки разработки САПР-продуктов.

# 3. Задание на выполнение работы

1. Розщитать толщину поззатаорного дталектрика

1. Рассчитать толщину подзатворного диэлектрика n- канальной МОП структуры с поликремнийовим затвором для получения порогового напряжения 0.7 ± 0.1В. Разработать графики зминения порогового напряжения от:

- концентрации примеси в пластине (1E14 - 1E16 sm -3)

- напряжения на подложке (0-10V)

При построении зависимости используются следующие параметры, задаваемые пользователем:

2F - 2 уровня Ферми;

Qox - заряд в окисле;  
ms - разница работ выхода материала затвора и подложки;

sub - относительная диэлектрическая проницаемость подложки;

ox - относительная диэлектрическая проницаемость окисла;

Cox - емкость подзатворного диэлектрика, нормирующая величина;

А также значения для :

1. Порогового напряжения *V*т;

2. Концентрации примесей в положке *N*sub;

Для расчета используются константы:

Абсолютная диэлектрическая проницаемость – E0=8.85\*10-12 Ф/м;

Заряд электрона - 1.6\*10-19;

# 4. Краткие теоретические сведения

1.Пороговое напряжение

При расчете использовалась формула:



(1)

где VT0 = - 2– (Qb /Cox ) – (QSS /Cox )

где – разница работ выходов затвора и полупроводника, (в наших терминах → ), – уровень Ферми, – обьемный заряд подложки, – заряд в диелектрике (считая, что диалектрик идеальный, заряд в нем не накопляется, поэтому ), – емкость подзатворного диелектрика.

Для расчета удельной емкости подзатворного диэлектрике воспользуемся формулой:

(2)

где - диэлектрическая проницаемость оксида, рассчитывается по формуле:

, (3)

Для расчета объемного заряда подложки используется формула:

(5)

где – заряд электрона, - концентрация примеси в подложке, – диелектрическая пронинацельность материала подложки, – напряжение междуж истоком и подложкой.

Коефициент (6)

Отсюда выразим и получим формулу толщины подзатворного диалектрика для заданного порогового напряжения:

(7)

# 5. Расчеты

Рассчитаем необходимую толщину подзатворного диелектрика используя формулу (7).

Для расчета в подложке нам понадобится:

* Уровень Ферми
* Пороговое напряжение (по умолчанию, но может также задаваться пользоателем)
* Удельная емкость подзатворного
* Заряд электрона
* Диелектрическая пронинательность подложки
* пусть разница работ выхода =0.1 eV =0.1\*q=0.1\*1.6\*1e-19=1.6\*1e-20
* вичислим толщину подзатворного диалектрика:

**УТВЕРДЖЕННО**

ГОСТ 19.503-79

ИПСА

Лист 1

Листов 4

Литера

Масса

Масштаб

Изм.

Лист

документа

Подпись

Дата

Разработал

Колинько А. М.

Проверил

Стиканов В.Ю.

Т. контр.

Н. контр.

Утвердил

Руководство системного программиста

**ГОСТ 19.503-79**

**Методы и инструментарий проектирования элементной базы информационных технологий**

ГОСТ 19.503-79

Руководство системного программиста.

**Листов 4.**

Описание составных частей; описание логики составных частей (при необходимости следует составлять описание схем программ). При описании логики программы необходима привязка к тексту программы.

1. **Общие сведения о программе**

Данная программа рассчитывает dox и записывает в соответствующие файлы данные для построения граффиков зависимости Vth(Nsub) и Vth(dox)

1. **Структура программы»**

Программа состоит из таких частей:

* Поля для задания параметров;
* Поле с результатами роботы (вычисленное dox);
* кнопки для построения графиков

Начало работы программы происходит с расчета dox из параметров заданних по умолчанию. После ввода нового параметра поле из рассчитаним значением автоматиески обновляется. Задать значения которые менше минимумма или больше максимума нельзя.

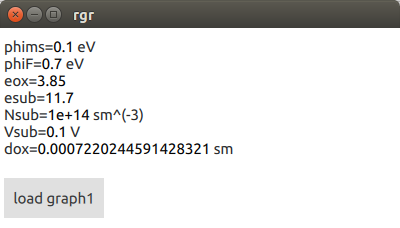
1. **Настройка программы**

Для запуска программы ее необходимо скачать и запустить соответствующий .exe файл. Прорамму также можно запистить в режеме отладки для просмотра диагностических сообщений. Для этого слудует установить среду Qt Creator (если она не устоновлена) и открить в ней соответствующий .pro файл. Далее необходимо нажать кнопку пуск в Qt Creator:

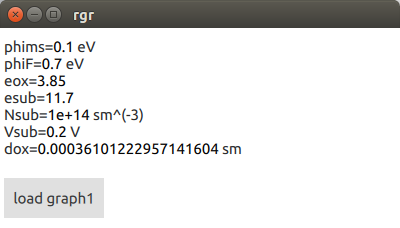


4. **«Проверка программы»**

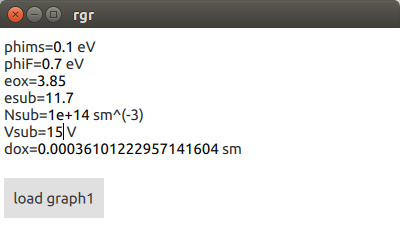
Для проверки работоспособности программы следует запустить программу. При первом запуске отображаются параметры по умолчанию и результат росщетов по етим параметрам



При вводе нового значения параметра результат расчетов обновляется автоматически. Например, введем значение Vsub=0.2 и нажмем “Enter”:



Если введенный параметр меньше минимума или больше максимума, данные не обновляются. Попробуем ввести Vsub=15 и нажать “Enter”:



Допустимые значения для параметров следующие:

"phims",phims,0.1,4,"eV"

"phiF",phiF,0.7,0.9,"eV"

"eox",eox,1.0,2000.0,""

"esub",esub,1.0,2000.0,""

"Nsub",Nsub,1.e14,1.e16,"sm^(-3)"

"Vsub",Vsub,0.,10.,"V"

Если в данных этапах не было ошибок в программе, то это свидетельствует о работоспособности программы.

Изм.

Лист

документа

Подпись

Дата

Лист

3

ГОСТ 19.503-79

5. **Записть данных для построения графиков**

Для получения данных для построения графиков необходимо нажать на кнопку “load graph1”. После нажатия данной кнопки будут разбиты промежутки аргументов на 50 точек и высчитаны значения соответствующих функций. Результаты будут записаны в файлы в той же директории где запускалась программа. Данные по файлам размещены следующим образом:

Файл src1.txt – отсщеты по аргументу Nsub первой функции;

Файл src2.txt – значения первой функции Vth(Nsub)от отсщетов из файла src1.txt;

Файл res1.txt – отсщеты по аргументу Vsub второй функции;

Файл res2.txt – значения второй функции Vth(Vsub)от отсщетов из файла src2.txt;

6. **Сообщения системному программисту**

Для запуска программы, убедитесь о присутствии файла для запуска в Вашей директории.



Рис.1

После открытия окна введите необходимые параметры системы и сохраните данные для построения графиков (или сохраните данные для построения графиков по данным по умолчанию) нажав на кнопку “load graph1”

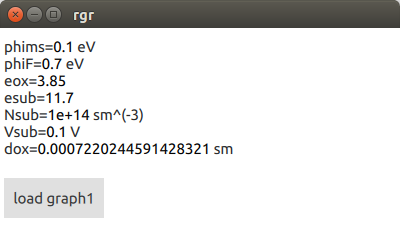


Рис.2

После этого можно скопировать данные с ссответствующих файлов и посроить граффики

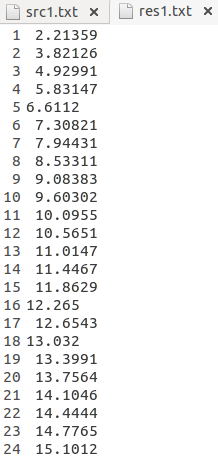


Рис.3

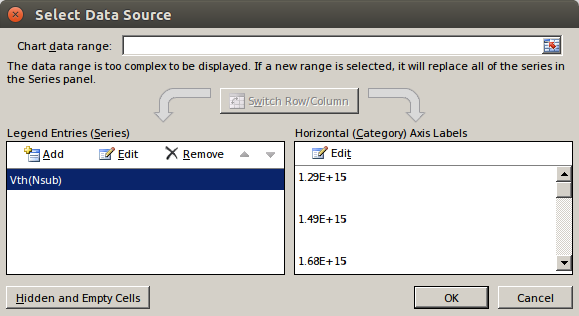


Рис.4

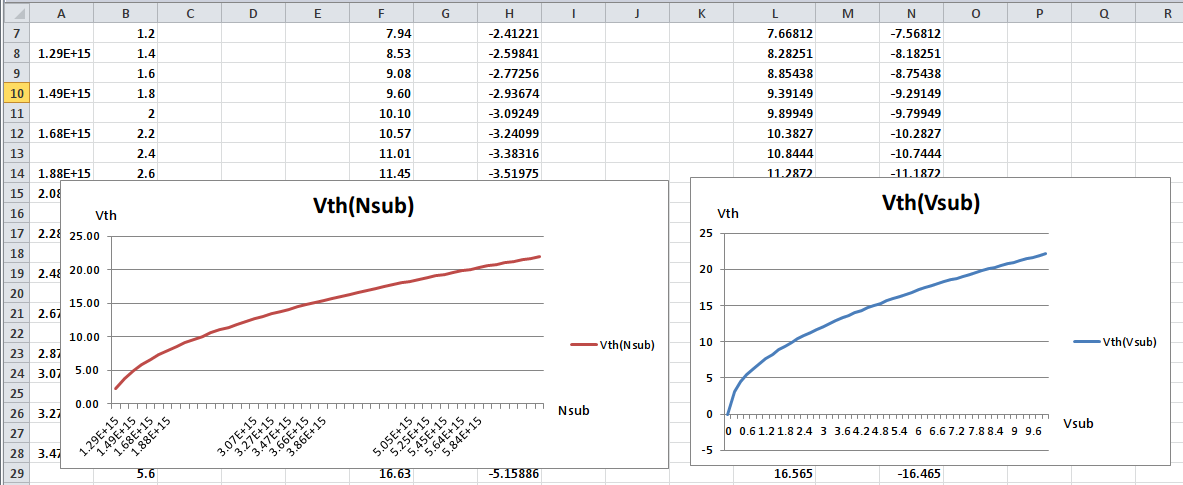


Рис. 5

Окно программы остается активным и в нем можно продолжить роботу. Для закрития окна необходимо нажать на соответствующую кнопку.

**УВТВЕРДЖЕННО**

документа

Подпись

Дата

Лист

3

ГО ГОСТ 19.503-79

ГОСТ 19.505-79

ИПСА

Лист 1

Листов 5

Литера

Масса

Масштаб

Изм.

Лист

документа

Подпись

Дата

Разработал

Колинько А. М.

Проверил

Стиканов В.Ю.

Т. контр.

Н. контр.

Утвердил

Руководство оператора

**ГОСТ 19.505-79**

**Методы и инструментарий проектирования элементной базы информационных технологий**

**ГОСТ 19.505-79**

**Руководство оператора.**

**Листов 3.**

1. **Назначение программы**

Программа рассчитывает для построения интегральной схемы физико-топологические и конструктивные параметры n - канального МОП транзистора. Реализовано с помощью графического интерфейса. При запуске программы будет выведено значение толщины подзатворного диэлектрика, необходимой для достижения заданных параметров.

1. **Условия выполнения программы**

Программа работает в непрерывном основном цикле и ожидает ввода от пользователя. При вводе корректных значений искомое значение толщины подзатворого диэлектрика пересчитывается и обновляется в соответствующем поле. При выборе сохранения данных для графиков (кнопка “load graph1”). Соответствующие данные записываются в файлы. Если файлы с соответствующими именами уже существуют в данной директории, они перезаписываются.

1. **Сообщения оператору**

Изм.

Лист

документа

Подпись

Дата

Лист

2

ГОСТ 19.505-79

Программа выполнена с помощью графического интерфейса. Но возможно получение с помощью консоли разработчика диагностических сообщений если программа была запущена в отладочном режиме (с помощью Qt Creator).

В консоль выводятся данные о вызываемых функциях и значениях некоторых переменных в этих функциях.

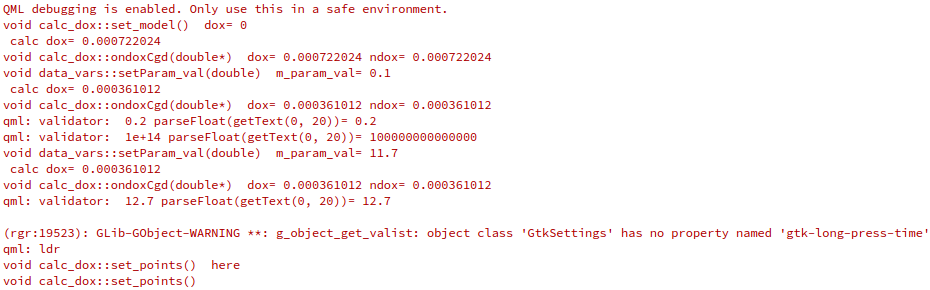


Рис.1

Также в консоли можно получить информацию о коде завершения программы:



Рис. 2

После сохранения данных для построения граффиков их можно скопировать в для построения по точкам.

Для завершения роботы программы необходимо нажать на соответствующую кнопку.

Изм.

Лист

документа

Подпись

Дата

Лист

3

ГОСТ 19.505-79

Код программы в приложениях.

Приложение 1. main.cpp

#include <QGuiApplication>

#include <QQmlApplicationEngine>

#include <QtWidgets/QApplication>

#include <QQmlContext>

#include <QQmlComponent>

#include "calc\_dox.h"

int main(int argc, char \*argv[])

{

QCoreApplication::setAttribute(Qt::AA\_EnableHighDpiScaling);

QApplication app(argc, argv);

//QGuiApplication app(argc, argv);

QQmlApplicationEngine engine;

//QQmlEngine engine\_qml;

QQmlContext \*context = engine.rootContext(); // Создаём корневой контекст

//QQmlComponent component(&engine\_qml,QUrl::fromLocalFile("q1.qml"));

//QObject \*root\_qml\_obj = component.create();

QList<QObject\*> dataList;

calc\_dox calc(dataList,context,&engine); // Создаём ядро приложения

/\* Загружаем объект в контекст для установки соединения,

\* а также определяем имя, по которому будет происходить соединение

\* \*/

context->setContextProperty("calc", &calc);

//context->setContextProperty("p\_model", QVariant::fromValue(dataList));

engine.load(QUrl(QLatin1String("qrc:/main.qml")));

if (engine.rootObjects().isEmpty())

return -1;

return app.exec();

}

Приложение 2. calc\_dox.h

#ifndef CALC\_DOX\_H

#define CALC\_DOX\_H

#include <QObject>

#include <QDebug>

#include <QVariantList>

#include <QQmlContext>

#include <QQmlApplicationEngine>

#include <QtCharts/QLineSeries>

#include <QtCharts/QtCharts>

#include "data\_vars.h"

using namespace QtCharts;

class calc\_dox : public QObject

{

Q\_OBJECT

public:

calc\_dox(QList<QObject \*> &dataList,QQmlContext\* cntxt, QQmlApplicationEngine\* p\_qml\_obj1, QObject\* parent=0);

~calc\_dox();

void set\_model(/\*QList<QObject\*>& dataList\*/);

Q\_INVOKABLE void set\_points();

signals:

void doxChanged(double\* ch);

public slots:

void calc();

void ondoxCgd(double\* ndox);

private:

double e0=8.854\*1e-14;//F\*sm^(-1);

double qe=1.602\*1e-19;//C;

double \*phims;//def: 0.1\*qe;//phi metal\_semiconductor; 1eV= 1\*qV;

double \*phiF;//def: 0.7\*qe;//Fermi level for n-type semiconductor;

double \*qox;//def: 0;//q\_oxide;

double \*eox;//def: 3.85;//e\_oxide;

double \*esub;//def: 11.7;//e\_sub;

double \*Nsub;//def: 10e14;//sm^(-3);

double \*Vsub;//def: 0;//V;

double \*Vth;//def: 0;//V;

double \*dox;//d\_oxide;

double \*qb;//q\_bulk;

QQmlContext\* qml\_cntxt;

QList<QObject \*> dataList;

QQmlApplicationEngine\* m\_engine;

int cnt;

};

//QList<QObject\*> calc\_dox::dataList;

#endif // CALC\_DOX\_H

Приложение 3. calc\_dox.cpp

#include<QFile>

#include "calc\_dox.h"

calc\_dox::calc\_dox(QList<QObject \*> &in\_dataList, QQmlContext\* cntxt, QQmlApplicationEngine\* p\_engine, QObject\* parent){

Q\_UNUSED(parent); //Q\_UNUSED(in\_dataList);

phims=new double; \*phims=0.1;//\*qe;//phi metal\_semiconductor; 1eV= 1\*qV;

phiF=new double; \*phiF=0.7;//\*qe;//Fermi level for n-type semiconductor;

qox=new double; \*qox=0;//q\_oxide;

eox=new double; \*eox=3.85;//e\_oxide;

esub=new double; \*esub=11.7;//e\_sub;

Nsub=new double; \*Nsub=1e14;//sm^(-3);

Vsub=new double; \*Vsub=0.1;//V;

Vth=new double; \*Vth=0.7;//V;

dox=new double; \*dox=0;

qb=new double; \*qb=0;

connect(this,&calc\_dox::doxChanged,this,&calc\_dox::ondoxCgd);

set\_model(/\*in\_dataList\*/);

in\_dataList=dataList;

for(unsigned int i=0;i<dataList.size()-1;i++){

connect(qobject\_cast<data\_vars\*> (dataList.at(i)), &data\_vars::param\_valChanged,

this,&calc\_dox::calc);

}

qml\_cntxt=cntxt;

//qml\_obj1=p\_qml\_obj1;

cnt=0;

calc();

m\_engine=p\_engine;

}

void calc\_dox::calc(){

//\*qb=sqrt(2\*qe\*(\*esub)\*e0\*(\*Nsub)\*fabs((\*Vsub)-2\*(\*phiF)\*qe)\*1.1);

/\*qDebug()<<Q\_FUNC\_INFO

<<"qe="<<qe<<" esub="<<(\*esub)<<" e0="<<e0<<" Nsub="<<(\*Nsub)<<" Vsub="<<(\*Vsub)<<" phiF="<<(\*phiF)<<" Vth="<<(\*Vth)

<<" phiF="<<(\*phiF)<<" phims="<<(\*phims)<<" Vsub+phiF\*qe="<<(\*Vsub)+2\*(\*phiF)\*qe<<" esub\*e0\*Nsub="<<(\*esub)\*e0\*(\*Nsub)

<<" eox="<<(\*eox)<<" qb="<<(\*qb);\*/

(\*dox)=((-(\*phims)\*qe+2\*(\*phiF)\*qe+(\*Vth))\*(\*eox)\*e0)

/(sqrt(2\*qe\*(\*eox)\*e0\*(\*Nsub))\*

(fabs(-2\*(\*phiF)\*qe+(\*Vsub))-

fabs(-2\*(\*phiF)\*qe)));

qDebug()<<" calc dox="<<(\*dox);

emit doxChanged(dox);

}

void calc\_dox::set\_model(/\*QList<QObject \*> &dataList\*/){

dataList.append(new data\_vars("phims",phims,0.1,4,"eV"));

dataList.append(new data\_vars("phiF",phiF,0.7,0.9,"eV"));

dataList.append(new data\_vars("eox",eox,1.0,2000.0,""));

dataList.append(new data\_vars("esub",esub,1.0,2000.0,""));

dataList.append(new data\_vars("Nsub",Nsub,1.e14,1.e16,"sm^(-3)"));

dataList.append(new data\_vars("Vsub",Vsub,0.,10.,"V"));

qDebug()<<Q\_FUNC\_INFO<<" dox="<<\*dox;

dataList.append(new data\_vars("dox",dox,\*dox,\*dox,"sm"));

}

void calc\_dox::ondoxCgd(double \*ndox){

qDebug()<<Q\_FUNC\_INFO<<" dox="<<\*dox<<"ndox="<<\*ndox; cnt++;

/\*delete dataList.takeLast();

data\_vars\* tmpdv=qobject\_cast<data\_vars\*> (dataList.at(dataList.size()-1);

tmpdv->setParam\_val(\*dox);

for(unsigned int i=0;i<dataList.size();i++){

data\_vars\* tmpdv=qobject\_cast<data\_vars\*> (dataList.at(i));

qDebug()<<Q\_FUNC\_INFO<<(tmpdv->param\_nm().toStdString()).c\_str()<<' '<<tmpdv->param\_val();

}

dataList.append(new data\_vars("dox",dox,\*dox,\*dox,"sm"));\*/

qml\_cntxt->setContextProperty("p\_model", QVariant::fromValue(dataList));

}

calc\_dox::~calc\_dox(){

delete phims;

delete phiF;

delete qox;

delete eox;

delete esub;

delete Nsub;

delete Vsub;

delete Vth;

delete dox;

delete qb;

}

void calc\_dox::set\_points(){

qDebug()<<Q\_FUNC\_INFO<<" here";

/\*QLineSeries \*series = new QLineSeries();

\*series << QPointF(0, 6) << QPointF(9, 4) << QPointF(15, 20) << QPointF(25, 12) << QPointF(29, 26);

QChart \*chart= new QChart();

chart->legend()->hide();

chart->addSeries(series);\*/

QDebug debug=qDebug();

QFile fl("./src1.txt"); QFile fl1("./res1.txt"); QFile fl2("./res2.txt"); QFile fl3("./src2.txt");

if(!fl.open(QIODevice::Truncate|QIODevice::WriteOnly)

||!fl1.open(QIODevice::Truncate|QIODevice::WriteOnly)

||!fl2.open(QIODevice::Truncate|QIODevice::WriteOnly)

||!fl3.open(QIODevice::Truncate|QIODevice::WriteOnly)

){

debug<<"\nsave: cannot open"; return;

}

QDataStream fl\_stream(&fl),fl1\_stream(&fl1),fl2\_stream(&fl2),fl3\_stream(&fl3);

\*Vth=0.5;// \*Nsub=1e14;

for(\*Nsub=1e14;\*Nsub<1e16;\*Nsub+=(1e16-1e14)/50){

fl\_stream<<QString::number(\*Nsub).toStdString().c\_str()<<'\n';

//\*qb=-sqrt(2\*qe\*(\*esub)\*e0\*(\*Nsub)\*fabs((\*Vsub)-2\*(\*phiF)\*qe));

fl1\_stream<<QString::number(

(\*phims)\*qe-2\*(\*phiF)\*qe+((\*dox)\*

sqrt(fabs(2\*qe\*(\*eox)\*e0\*(\*Nsub)))\*

(sqrt(fabs(-2\*(\*phiF)\*qe+(\*Vsub)))-sqrt(fabs(-2\*(\*phiF)\*qe)))

)/((\*eox)\*e0)).toStdString().c\_str()<<'\n';

}

\*Vth=0.5;// \*Nsub=1e14;

for((\*Vsub)=0;(\*Vsub)<10;(\*Vsub)+=(10-0)/50.){

fl3\_stream<<QString::number((\*Vsub)).toStdString().c\_str()<<'\n';

\*qb=-sqrt(2\*qe\*(\*esub)\*e0\*(\*Nsub)\*fabs((\*Vsub)-2\*(\*phiF)\*qe));

fl2\_stream<<QString::number(

(\*phims)\*qe-2\*(\*phiF)\*qe+((\*dox)\*

sqrt(fabs(2\*qe\*(\*eox)\*e0\*(\*Nsub)))\*

(sqrt(fabs(-2\*(\*phiF)\*qe+(\*Vsub)))-sqrt(fabs(-2\*(\*phiF)\*qe)))

)/((\*eox)\*e0)).toStdString().c\_str()<<'\n';

}

fl.close(); fl1.close(); fl2.close(); fl3.close();

qDebug()<<Q\_FUNC\_INFO;

}

Приложение 4. data\_vars.h

#ifndef DATA\_VARS\_H

#define DATA\_VARS\_H

#include <QObject>

#include <QDebug>

class data\_vars : public QObject

{

Q\_OBJECT

Q\_PROPERTY(QString param\_nm READ param\_nm CONSTANT)

Q\_PROPERTY(double param\_val READ param\_val WRITE setParam\_val NOTIFY param\_valChanged)

Q\_PROPERTY(double l\_bound READ l\_bound CONSTANT)

Q\_PROPERTY(double t\_bound READ t\_bound CONSTANT)

Q\_PROPERTY(QString param\_units READ param\_units CONSTANT)

public:

data\_vars(QString p\_nm, double\* p\_val, double lbnd, double tbnd, QString p\_units, QObject\* parent=0);

~data\_vars();

QString param\_nm() const;

double param\_val() const;

double l\_bound() const;

double t\_bound() const;

QString param\_units() const;

void setParam\_val(const double n\_param\_val);

signals:

void param\_valChanged();

private:

QString m\_param\_nm;

double m\_param\_val;

double\* pm\_param\_val;

double m\_l\_bound;

double m\_t\_bound;

QString m\_param\_units;

};

#endif // DATA\_VARS\_H

Приложение 5. data\_vars.cpp

#include "data\_vars.h"

data\_vars::data\_vars(QString p\_nm, double \*p\_val, double lbnd, double tbnd, QString p\_units, QObject\* parent):

m\_param\_nm(p\_nm), m\_param\_val(\*p\_val),pm\_param\_val(p\_val), m\_l\_bound(lbnd), m\_t\_bound(tbnd), m\_param\_units(p\_units){

Q\_UNUSED(parent);

pm\_param\_val=new double; pm\_param\_val=p\_val;

//qDebug()<<Q\_FUNC\_INFO<<" p\_val="<<\*p\_val<<' '<<m\_param\_val;

}

QString data\_vars::param\_nm() const{

return m\_param\_nm;

}

double data\_vars::param\_val() const{

//qDebug()<<Q\_FUNC\_INFO<<" pm\_param\_val="<<\*pm\_param\_val<<" m\_param\_val="<<m\_param\_val;

return \*pm\_param\_val;

}

double data\_vars::l\_bound() const{

return m\_l\_bound;

}

double data\_vars::t\_bound() const{

return m\_t\_bound;

}

QString data\_vars::param\_units() const{

return m\_param\_units;

}

void data\_vars::setParam\_val(const double n\_param\_val){

if(n\_param\_val!=\*pm\_param\_val){

qDebug()<<Q\_FUNC\_INFO<<" m\_param\_val="<<\*pm\_param\_val;

\*pm\_param\_val=n\_param\_val; m\_param\_val=n\_param\_val;

emit param\_valChanged();

}

}

data\_vars::~data\_vars(){

delete pm\_param\_val;

}

Приложение 6. main.qml

import QtQuick 2.7

import QtQuick.Controls 2.0

import QtQuick.Layouts 1.0

ApplicationWindow {

visible: true

width: 400

height: 200

title: qsTr("rgr")

ListView{

id: g\_view

width: parent.width

height: parent.height-parent.height\*0.1

anchors.top: parent.top

anchors.left: parent.left

anchors.topMargin: parent.height\*0.05

anchors.leftMargin: parent.width\*0.01

interactive: false

model: p\_model

delegate: Rectangle{

id: rect

width: childrenRect.width

height: childrenRect.height

//border.color: "red"

Label{

id: curr\_lbl

text: model.modelData.param\_nm+'='

}

TextInput{

id: curr\_ti

anchors.left: curr\_lbl.right

text: model.modelData.param\_val;

validator: DoubleValidator{

id: curr\_validator

bottom: model.modelData.l\_bound; top: model.modelData.t\_bound

}

onEditingFinished: {

if(model.modelData.param\_nm!="dox"){

model.modelData.param\_val=curr\_ti.text;

}

else{

console.log("qml: dox");

}

console.log("validator: ",curr\_ti.text,"parseFloat(getText(0, 20))=",parseFloat(getText(0, 20)));

}

}

Label{

anchors.left: curr\_ti.right

text: ' '+model.modelData.param\_units

}

}

}

Button{

text: "load graph1";

anchors.bottom: parent.bottom

anchors.left: parent.left

anchors.bottomMargin: parent.height\*0.05

anchors.leftMargin: parent.width\*0.01

MouseArea{

anchors.fill: parent

onClicked:{

console.log("ldr");

calc.set\_points();

//ldr.source="q1.qml";

}

}

}

/\* ListView{

width: 200

height: 500

focus: true

highlight: Component {

Rectangle {

width: 200

height: 20

color: "red"

}

}

model: p\_model

delegate: Label{

anchors.left: parent.left;

color: "black"

text: model.modelData.param\_nm

}

}\*/

/\*TextInput{

id: inppfims

anchors.left: lblphims.right;

text: calc.pphims/calc.pqe;

validator: DoubleValidator{bottom: 0.1; top: 4.0}

onEditingFinished: {

calc.pphims=parseFloat(getText(0,20))\*calc.pqe;

console.log("getText(0,20)=",getText(0,20)," parseFloat(getText(0,20))=",parseFloat(getText(0,20)));

}

}

Label{

id: lblphiF;

text: "phiF=";

}

TextInput{

id: inpphiF

anchors.left: inppfims.right;

text: calc.pphiF/calc.pqe;

validator: DoubleValidator{bottom: 0.7; top: 0.9}

onEditingFinished: {

calc.phiF=parseFloat(getText(0,20))\*calc.pqe;

console.log("getText(0,20)=",getText(0,20)," parseFloat(getText(0,20))=",parseFloat(getText(0,20)));

}

}\*/

Loader{

id:ldr;

anchors.fill: parent;

Component.onCompleted:{//dubug mode;

//calc.set\_points();

//source="q1.qml";

}

}

}