МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет физический

Кафедра физико-технической информатики

Направление подготовки: 011200

Образовательная программа: _____

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Каньшина Артемия Николаевича

(Фамилия, Имя, Отчество автора)

Тема работы: АВТОМАТИЗАЦИЯ ИЗМЕРЕНИЙ ПОПЕРЕЧНЫХ РАЗМЕРОВ ПУЧКА В БУСТЕРЕ БЭП

«К защите допущена»

Заведующий кафедрой,

ученая степень, звание

Логашенко И. Б. /……..

(фамилия , И., О.) / (подпись, МП)

«……»……20…г.

Научный руководитель

ученая степень, звание

должность, место работы

Сенченко А. И. / ……..

(фамилия , И., О.) / (подпись, МП)

«……»……2017 г.

Дата защиты: 19 июня 2017 г.

Новосибирск, 2017

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Электрон-позитронный коллайдер ВЭПП-2000 был введён в эксплуатацию в 2007 году. С 2010 по 2013 годы было проведено три успешных запуска установки с накоплением данных в диапазоне энергий пучка частиц от 160 до 1000 МэВ.

В течении этой работы комплекс ВЭПП-2000 использовал инжекционный канал его предшественника ВЭПП-2М. Данная установка работала на энергии, меньшей 700 МэВ, и показывала светимость в 30 раз меньшую, чем проектный уровень светимости в 10^{32} $см^{-2}с^{-1}$ у ВЭПП-2000 с энергией пучка 1 ГэВ. В итоге, скорость накопления позитронов была недостаточной для достижения светимости, ограниченной лишь пороговым током пучка. Это ограничение было устранено путем соединения каналом К-500 с новым инжекционным комплексом ВЭПП-5, способным создавать интенсивные электронные и позитронные пучки высокого качества с энергией 450 МэВ [1]. Полная схема инжекционного комплекса ВЭПП-5 и комплекса ВЭПП-2000 показана на рис. 1.

Другое ограничение эффективности ВЭПП-2000 приходило от максимальной энергии работы бустерного кольца БЭП, ограниченного энергией в 800 МэВ [2]. От этого пучок частиц, перепущенный из БЭП в ВЭПП-2000, неизбежно уменьшался после ускорения в кольце коллайдера. Поэтому в 2013 году было принято решение о модернизации бустерного кольца БЭП как для увеличения максимальной энергии работы до 1 ГэВ, так и для успешной инжекции пучков электронов и позитронов из инжекционного комплекса ВЭПП-5, с целью достижения проектной светимости коллайдера ВЭПП-2000 [3].

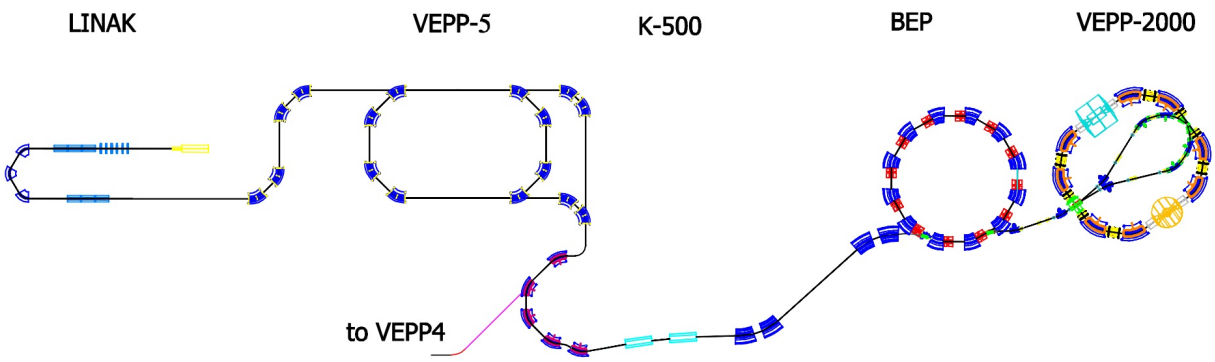

Рис. 1: Схема инжекционного комплекса и комплекса ВЭПП-2000

Эффективная эксплуатация ускорителя заряженных частиц требует измерения параметров циркулирующих пучков, это необходимо как на этапе запуска, так и при ежедневной эксплуатации для настройки оптимальных режимов работы. Поэтому в рамках модернизации бустерного кольца было принято решение оборудовать несколько выводов синхротронного излучения новыми, современными ПЗС матрицами для создания новой системы диагностики пучка в БЭП. В качестве такой ПЗС матрицы была выбрана камера Chameleon CMLN-13S2M, характеристики которой описаны в главе 1. Данная камера отличается от камер, используемых в существующей системе диагностики, поэтому видится целесообразным создание новой системы диагностики пучка.

Разработка новой системы диагностики пучка осложнено наличием в системе управления установкой большого количества различных протоколов передачи данных. В рамках работ по унификации протоколов было решено исследовать возможность применения программной среды TANGO на примере данной задачи.

Данная работа посвящена разработке системы автоматизированного измерения параметров пучка частиц по изображению синхротронного излучения, таких как положение, поперечные размеры и наклон, на базе TANGO.

Глава 1 полностью посвящена постановке задачи, описанию оборудования и программных средств, используемых в данной работе.

Глава 2 посвящена разработке всех компонент системы автоматизированного измерения параметров пучка.

В **главе 3** описан процесс тестирования системы на такие характеристики, как быстродействие и точность вычислений параметров пучка при различных уровнях сигнал/шум на изображении.

В **заключении** подведены итоги данной работы.

Глава 1 . ПОСТАНОВКА ЗАДАЧИ

1.1 Знакомство с TANGO

Для создания новой системы диагностики пучка в БЭП на основе программной среды TANGO, необходимо познакомиться с ней, изучить принципы разработки программ с её использованием, а также изучить API для камер Chameleon под названием FlyCapture2.

TANGO — свободная, с открытым исходным кодом, объектно-ориентированная программная среда, предназначенная для распределенного управления устройствами, различным оборудованием и программным обеспечением. По сути, TANGO — трехуровневая система драйвер-сервер-клиент (см. рис 2), предназначенная для управления в рамках локальной сети. Под драйверами подразумеваются программы, являющиеся серверами устройств, каждый из которых работает под одним сервером на каком-либо компьютере. Все экземпляры серверов устройств регистрируются в базе данных, находящейся на какой-либо машине в рамках этой локальной сети. Сервера TANGO, к которым подключаются сервера устройств, может быть произвольное количество. Клиентские программы могут унифицированным образом управлять и получать данные с каждого устройства, работающего в этой системе. В качестве сетевого протокола используется omniORB реализация CORBA. От разработчика программ на базе TANGO требуется реализовать сервер устройства и клиентскую программу с использованием TANGO API, но для общих целей в TANGO уже присутствует универсальная клиентская программа под названием AtkPanel, которая автоматически генерирует интерфейс, исходя из настроек устройства, то есть из его атрибут, свойств и методов. Взаимодействие с устройством происходит непосредственно через атрибуты и методы. Атрибуты могут иметь три уровня доступа: только чтение, только запись, чтение и запись. А методы позволяют управлять

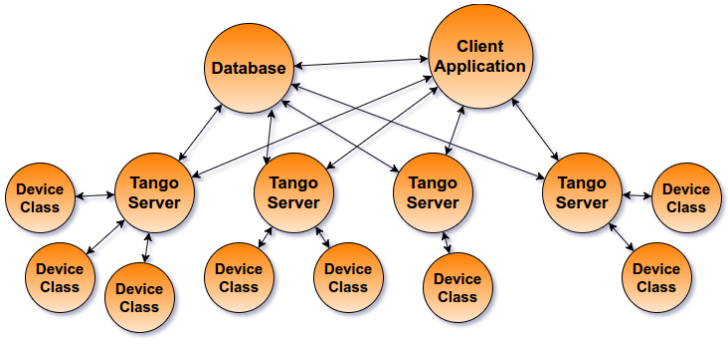

Рис. 2: Схема системы TANGO

состоянием устройств, например, автоматическая калибровка, физическое включение или выключение.

На момент начала выполнения данной работы последней версией TANGO являлась версия 9.2.5, которая поставлялась только в виде исходного кода. Поэтому пришлось собирать и настраивать TANGO вручную. Задача установки TANGO из исходников оказалась непростой и затратила приличное количество времени.

Для изучения TANGO было решено создать симулятор генератора синусоидального сигнала с шумом с возможностью регулировать параметры сигнала, такие как амплитуда, частота, уровень шума, шаг дискретизации и размер массива данных. В качестве атрибутов чтения записи возьмем все ранее перечисленные параметры сигнала, а в качестве атрибута для чтения создадим массив данных, содержащий синусоидальный сигнал.

В TANGO есть инструмент быстрого генерирования исходного кода серверов устройств, называемый POGO. При помощи POGO создадим сервер устройства (далее будем называть сервер устройства device-классом) под названием *SinNoise*, добавим в него атрибуты ***amplitude***, ***frequency***, ***noise***, ***bufferSize, step*** и ***data*** (см. рис). Как было сказано ранее: ***amplitude, frequency, noise, step, bufferSize*** - атрибуты для чтения и записи, ***data*** - только для чтения. Причем, ***amplitude, frequency, noise, step*** - являются скалярами типа DevDouble (double), ***bufferSize*** - скаляр типа DevLong (int), а ***data*** - спектр типа DevDouble (одномерный массив double). Результатом генерации кода device-класса стал набор из двух классов: *SinNoise* и *SinNoiseClass*. Класс *SinNoise* является основным, он выполняет работу непосредственно с устройством, то есть читает и записывает в него данные. Чтение и запись данных инициализируется запросом из клиентской части. Класс *SinNoiseClass* является классом, описывающим наш device-класс, описывает методы, атрибуты и свойства для того, чтобы клиент мог знать какие есть атрибуты и т.д.. Разработчику необходимо работать только с классом *SinNoise*. Также среди сгенерированных файлов есть такие файлы, как *ClassLoader.cpp*, *SinNoiseStateMachine.cpp*, *main.cpp* и *Makefile*. То есть это полноценная программа, которую уже можно компилировать и запускать.

Чтобы этот device-класс стал работать как генератор синусоидального сигнала с шумом, необходимо создать функцию: **void** *getSinNoise*(**double** *data, **int** size, **double**

amplitude, **double** frequency, **double** noise, **double** step) в отдельных файлах *mysinnoise.h* и *mysinnoise.cpp* (см. рис 3).

```
#include "mysinnoise.h"
#include <math.h>
#include <time.h>
#include <stdlib.h>

void getSinNoise(double *data, int size,
                 double amplitude, double frequency,
                 double noise, double step) {
    srand(time(NULL));
    if (data != NULL && size > 0) {
        for (int i = 0; i < size; i++) {
            data[i] = amplitude * sin(i * step / 1000.0 * frequency);
            if (noise > 0)
                data[i] += (double) (rand() % (int)(noise * 1000)) / 1000.0;
        }
    }
}
```

Рис. 3: Код функции getSinNoise

В классе *SinNoise* есть специальные функции для работы с атрибутами, например, атрибут ***amplitude*** в классе задается как ***attr_amplitude_read***, и к нему прилагается две функции: *read_amplitude*() и *write_amplitude*(), содержимое которых необходимо реализовывать самому. Все функции для работы с атрибутами параметров реализуются очень просто, достаточно читать и присваивать значения соответствующим полям класса. А в функции *read_data*() необходимо вызвать нашу функцию, генерирующую сигнал, но перед этим необходимо проверять не изменился ли атрибут ***bufferSize***. В случае, если изменился, изменяем размер массива данных.

```
void SinNoise::read_amplitude(Tango::Attribute &attr)
{
    DEBUG_STREAM << "SinNoise::read_amplitude(Tango::Attribute &attr) entering... " << endl;
    /*----- PROTECTED REGION ID(SinNoise::read_amplitude) ENABLED START -----*/
    //  Set the attribute value
    attr.set_value(attr_amplitude_read);

    /*----- PROTECTED REGION END -----*/    //  SinNoise::read_amplitude
}
```

Рис. 4: Код функции read_amplitude

```
void SinNoise::read_data(Tango::Attribute &attr)
{
    DEBUG_STREAM << "SinNoise::read_data(Tango::Attribute &attr) entering... " << endl;
    /*----- PROTECTED REGION ID(SinNoise::read_data) ENABLED START -----*/
    //  Set the attribute value

    if (sizeChanged) {
        Tango::DevDouble *tmp = attr_data_read;
        attr_data_read = new Tango::DevDouble[*attr_bufferSize_read];
        delete[] tmp;
        sizeChanged = false;
    }

    getSinNoise(attr_data_read, *attr_bufferSize_read,
                *attr_amplitude_read, *attr_frequency_read,
                *attr_noise_read, *attr_step_read);

    attr.set_value(attr_data_read, *attr_bufferSize_read);

    /*----- PROTECTED REGION END -----*/    //  SinNoise::read_data
}
```

Рис. 5: Код функции read_data

После реализации всех этих функций, необходимо скомпилировать программу и зарегистрировать device-class в базе данных под именем, например, test/SinNoise/dev1. Запустим device-класс и попробуем поуправлять параметрами через программу AtkPanel, автоматически генерирующую клиентский интерфейс. Результат запуска симулятора генератора представлен на рис. 6 и 7.

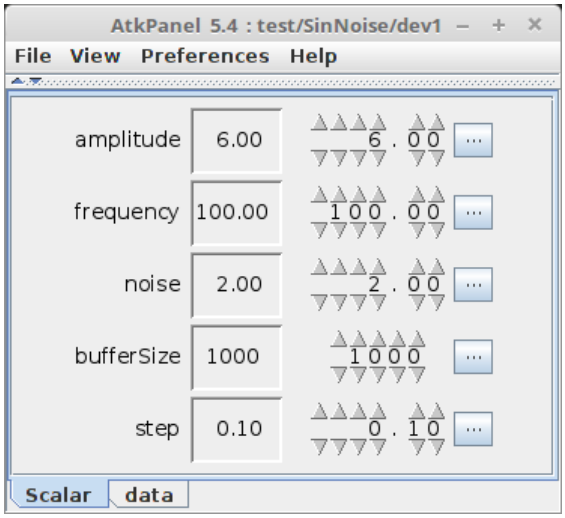

Рис. 7: Управление параметрами

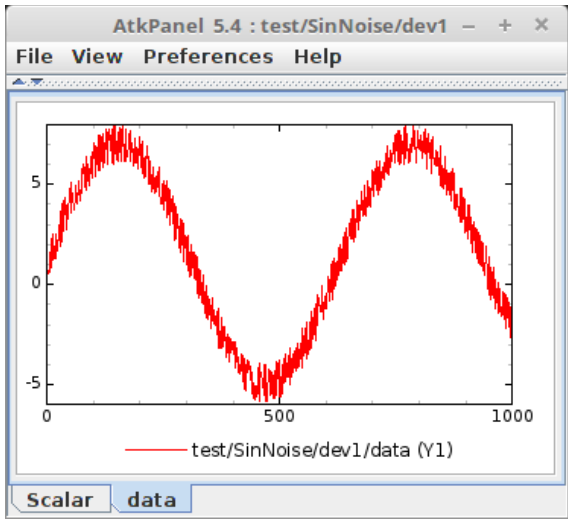

Рис. 6: График сигнала

Таким образом было изучено каким образом работать с TANGO, как создавать device-классы, как их запускать и как с ними работать. Данного опыта достаточно для создания device-класса для работы с ПЗС камерой.

1.2 ПЗС Камера Chameleon

Пучок заряженных частиц, двигаясь по траектории, искривленной магнитным полем, испускает по касательной к траектории электромагнитное излучение, называемое синхротронным. На энергиях пучка в БЭП синхротронное излучение имеет длины волн в диапазоне от 5500 Å до 5 Å, то есть от зеленого света до ультра-фиолетового.

БЭП оборудован выводами синхротронного излучения, обладающего узкой апертурой, чтобы как можно лучше отфильтровать свет, распространяющийся под необходимым углом. На конце вывода синхротронного излучения устанавливается ПЗС матрица, собирающая этот свет. По изображению с ПЗС матрицы можно измерить поперечные размеры и положение пучка заряженных частиц. Принцип работы вывода синхротронного излучения представлен на рис. 8.

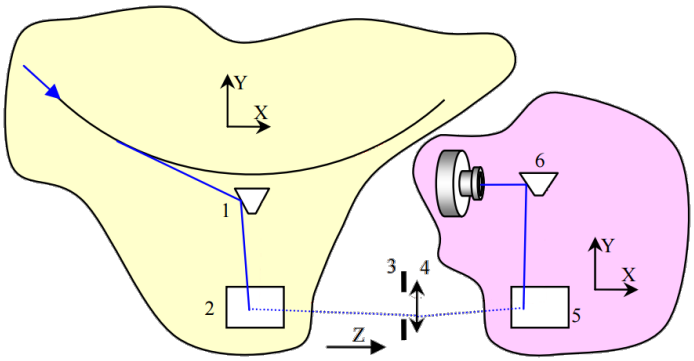

Рис. 8: Принцип работы вывода СИ

В качестве ПЗС матрицы, используемой в данной работе, научным руководителем была выбрана камера Chameleon CMLN-13S2M. Данная камера обладает следующими основными характеристиками:

Название	**Величина**
Разрешение изображения	1280x960
Размер пикселя	3.75 мкм
Цветность	моно
Количество бит на пиксель	8
Максимальная частота кадров	16.28 Гц
Потребляемая мощность	2 Вт
Габаритные размеры	25.5 мм x 44 мм x 41 мм
Масса	37 г

Таблица 1. Основные характеристики камеры Chameleon

Рис. 9: Chameleon CMLN-13S2M

Для программной работы с этой камерой компания-производитель PointGrey поставляет программный инструментарий FlyCapture2. Данный инструментарий является программной библиотекой, реализованной на C++, при помощи которой можно создавать собственное программное обеспечение для работы с камерой, регулировать её параметры, такие как частота кадров, выдержка, усиление, гамму и т.д.

Таким образом, является возможным создать device-класс, который будет способен управлять камерой, получать с неё данные, измерять параметры пучка и отправлять готовые результаты клиенту. Но библиотека FlyCapture2 предоставляет доступ к камере очень непростым и громоздким способом, используя собственные реализации типов данных, например, класс *Image*. Следовательно, для удобной работы с камерой необходимо создать библиотеку-обёртку, предоставляющую удобный интерфейс для работы с камерой, используя при этом стандартные типы данных. Реализовав данную обёртку, мы сможем использовать её в device-классе.

1.3 Измерение параметров пучка

После того, как мы научимся получать изображения с камеры, необходимо каким-то образом измерять по нему параметры пучка заряженных частиц. Как известно, в идеальном случае, когда сигнал не вызывает пересвет на пзс матрице и когда установка работает правильно, изображение синхротронного излучения представляет из себя двумерную гауссову функцию. Эта гауссова функция может иметь центр масс в любой точке изображения, иметь произвольные размеры и угол поворота.

Если мы имеем дело с аппроксимацией какого-либо сигнала какой-либо функцией, то мы можем воспользоваться библиотекой GSL[1] (GNU Scientific Library), которая позволяет методом наименьших квадратов аппроксимировать различные данные произвольными функциями. Данная библиотека реализована на языке C.

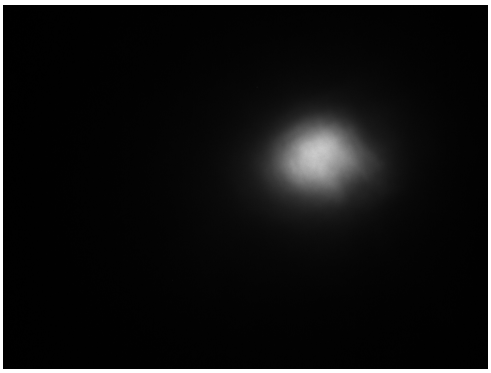

Рис. 10: Изображение СИ

Также, поскольку мы имеем дело с обработкой изображений, разумно воспользоваться библиотекой для работы с изображениями, позволяющей, например, фильтровать изображение, менять размер и т. д. Одной из популярнейших библиотек для работы с изображениями является

1 GSL — GNU Scientific library. - www.gnu.org/software/gsl/

библиотека машинного зрения OpenCV[2]. Возможности этой библиотеки очень большие. Она позволяет обрабатывать изображения, искать объекты, контуры и многое другое.

Следовательно, необходимо создать программный модуль, способный измерять параметры пучка по изображению СИ с использованием этих библиотек. Этот модуль будет использоваться в device-классе.

1.4 Клиентская программа

Для управления множеством камер типа Chameleon и наблюдением изображений и результатов измерений, необходимо создать программу с графическим пользовательским интерфейсом, предоставляющий такие возможности. Существует множество способов реализовать программу с графическим интерфейсом. Разумным шагом будет использовать фреймворк Qt для создания такой программы.

Qt[3] — кроссплатформенный инструментарий для создания программ с графическим пользовательским интерфейсом на языке C++. Конечно, есть и реализация Qt для языка Python, называемая PyQt, но решено реализовывать систему диагностики пучка полностью на C/C++. Qt предоставляет большой набор инструментов для создания программ. К этому набору относятся: среда разработки Qt Creator, интегрированный в него инструмент Qt Designer и многое другое. Главной особенностью Qt является реализация механизма сигналов и слотов. Этот механизм позволяет инициировать какие-либо действия или передавать данные в один объект из другого вне зависимости от того, в каких потоках работают эти объекты, также этот механизм обеспечивает принцип инкапсуляции. Слот одного объекта может быть подписан на сигнал другого объектом, стоящим выше по иерархии инициализации объектов. Например в паттерне MVC (Model View Controller) есть три различных объекта: view, model

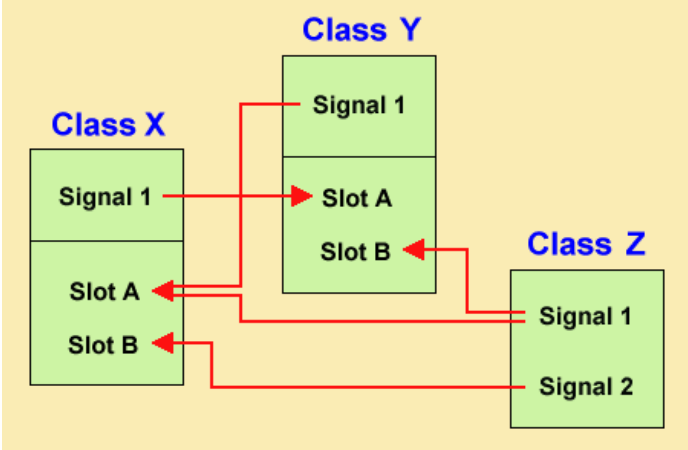

Рис. 11: Схема механизма сигналов и слотов в Qt

2 OpenCV - opencv.org

3 Qt - www.qt.io

и controller. На view могут быть созданы такие сигналы, как нажатие кнопки, окончание ввода текста и т. д. Model может сигнализировать о том, что данные изменились. Соединив сигналы этих объектов с их предполагаемыми обработчиками через объект controller, можно обеспечить потоково независимую работу, в которой объекты не будут знать о существовании каких-то других объектов, подслушивающих их сигналы. Поэтому фреймворк Qt подходит для задачи создания программы с графическим пользовательским интерфейсом на языке C++ лучше всего.

1.5 Итог постановки задачи

Главными целями этой работы являются:

- Создать новую систему автоматизированного измерения параметров пучка заряженных частиц в БЭП на комплексе ВЭПП-2000;
- Исследовать применимость программной среды TANGO в данной задаче, то есть исследовать её возможности в скорости и сложности написания программ, а также в её быстродействии.

Из проведенного в предыдущих параграфах разбора всех инструментов, с которыми предстоит работать, и поставленных выше целей вытекают следующие задачи:

- Создать device-класс для камеры Chameleon;
- Реализовать библиотеку, являющуюся обёрткой библиотеки FlyCapture2, для удобной работы с камерой Chameleon;
- Разработать алгоритм измерения параметров пучка по изображению СИ;
- Сконструировать клиентскую программу с графическим пользовательским интерфейсом, позволяющим работать с произвольным числом камер.

Следующая глава посвящена описанию процесса выполнения поставленных целей и задач.

Глава 2 . РАЗРАБОТКА

2.1 Библиотека для работы с камерой Chameleon

2.2 Сервер устройства

2.3 Клиентская программа

2.4 Алгоритм измерения параметров пучка по изображению синхротронного излучения

Глава 3 . ТЕСТИРОВАНИЕ

3.1 Быстродействие системы

3.2 Реакция алгоритма на шумы на изображении

ЗАКЛЮЧЕНИЕ

Итогом данной работы служит созданная с нуля система автоматизированного измерения параметров пучка БЭП, такие как положение центра масс, поперечные размеры и угол наклона. Эти параметры измеряются в сервере камеры синхронно с получением нового изображения в модуле, работающим непосредственно с самой камерой. Изображение и измеренные данные отправляются по локальной сети в клиентское приложение для отображения в графическом интерфейсе. Клиентская программа позволяет работать с произвольным количеством камер, что является важной характеристикой системы.

Главными задачами этой работы были реализация алгоритма измерения параметров пучка и достижение максимальной производительности. Главным требованием к алгоритму было время выполнения на одном изображении меньше 60 мс, чтобы успевать измерять параметры вслед за камерой. Исследование показало, что TANGO обладает пропускной способностью порядка 10 МБ/с, что является достаточным для данной задачи. Также в рамках этих 60-ти миллисекунд необходимо было проводить необходимые меры по сжатию данных для обеспечения требуемого быстродействия системы. В результате данная система способна работать одновременно с более 20-ю камерами.

Как показало тестирование системы на установке, изготовленный программный продукт удовлетворяет необходимые потребности научного сотрудника лаборатории в работе с изображениями синхротронного излучения, а именно возможности: наблюдать изображения и параметры пучка с множества камер; регулировать параметры камер в этом множестве; сохранять изображение для каких-либо дальнейших личных исследований; управлять быстродействием системы, а именно регулировать частоту кадров и качество изображения; сохранять и восстанавливать состояние системы при возникновении каких-либо неполадок.

СПИСОК ЛИТЕРАТУРЫ

1. Д. Е. Беркаев и др. - «Comissioning of Upgraded VEPP-2000 Injection Chain» - Proc. of International Particle Accelerator Conference IPAC-2016, Busan, Korea.
2. Ю. А. Роговский и др - «Recomissioning and perspectives of VEPP-2000 complex» - Proc of International Particle Accelerator Conference RuPAC-2016, Saint-Petersburg, Russia.
3. Д. Б. Шварц и др - «Booster of electrons and positrons (BEP) upgrade to 1 GeV» - Proc. of International Particle Accelerator Conference IPAC-2014, Dresden, Germany.
4. Техническое описание Chameleon CMLN-13S2M-CS [Электронный ресурс]. - Режим доступа: www.ptgrey.com/chameleon-13-mp-mono-usb-2-sony-icx445-camera (2016).
5. TANGO Control System [Электронный ресурс]. - Режим доступа: www.tango-controls.org (2016).
6. TANGO — Википедия [Электронный ресурс]. - Режим доступа: ru.wikipedia.org/wiki/TANGO (2016).
7. GNU Scientific Library [Электронный ресурс]. - Режим доступа: www.gnu.org/software/gsl/ (2017).
8. OpenCV [Электронный ресурс]. - Режим доступа: opencv.org (2017).