

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Институт машиностроения  
(наименования института полностью)

Кафедра «Промышленная электроника»  
(наименование)

11.03.04 Электроника и наноэлектроника  
(код и наименование направления подготовки, специальности)

Электроника и робототехника  
(направленность (профиль) / специализация)

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

на тему СТЕНД ДЛЯ ИЗУЧЕНИЯ СВОБОДНОГО ПОЛЕТА ТЕЛА

Студент	<u>М.М. Хамроев</u> (И.О. Фамилия)	<u>_____</u> (личная подпись)
Руководитель	<u>д.т.н., проф. В.П. Певчев</u> (ученая степень, звание, И.О. Фамилия)	<u>_____</u>

Тольятти 2021

## Аннотация

Объем 54 с., 30 рис., 2 табл., 23 источников  
МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ДВИЖЕНИЯ, СВОБОД-  
НЫЙ ПОЛЕТ ТЕЛА, СТЕНД.

Объектом исследования является стенд для изучения свободного полета тела.

Цель работы: разработка стенда для изучения свободного полета тела.

Задачи работы:

1. Обзор состояния вопроса
2. Разработка электрической части стенда
3. Разработка программы для микроконтроллера
4. Математическое моделирование

Данные задачи решены в 5 главах работы.

Степень внедрения: теоретически рассчитаны параметры системы, разработаны схемы и выбраны элементы, написана программа для микроконтроллера.

В данной работе происходит разработка стенда для изучения свободного полета тела. Выполнен обзор состояния вопроса, расчет электрической схемы, разработка печатной платы и программы для микроконтроллера. Разработанный стенд позволяет изучать свободный полет тела. Стенд может быть применен в школе в качестве лабораторного стенда по физике или в ВУЗе в качестве стенда по программированию.

## **Abstract**

The title of the graduation work is «Bench for studying the free flight of the body».

The senior thesis consists of an introduction, 5 parts, a conclusion, 30 figures, 2 tables, list of references including 6 foreign sources and the graphic part on 6 A1 sheets.

The aim of the graduation work is development of bench for studying the free flight of the body.

The object of research is bench for studying the free flight of the body.

We start with the statement of the problem and then logically pass over to its possible solutions. We then analyze the literature concerning our problem.

We give a review of possible solutions and choose the most suitable ones, develop an electrical circuit diagram, carry out theoretical calculations and select elements.

The special part of the project gives details about working out electrical part of the bench, writing a program for the microcontroller and testing mathematical modeling.

In conclusion we'd like to stress that we calculated the system parameters, designed circuits, selected elements and wrote a program for the microcontroller. Nevertheless, more experimental data are required.

This work is of interest to readers involved in physics or programming.

## Содержание

Введение.....	5
1.1 Анализ исходных данных и известных решений .....	7
1.2 Формулировка актуальности, цели и задачи работы .....	10
2 Механическая часть стенда .....	18
3 Электрическая часть стенда .....	19
3.1 Обзор вариантов.....	19
3.2 Разработка электрической схемы.....	21
4 Программная часть.....	28
4.1 Общая блок-схема программы .....	28
4.2 Подпрограмма идентификации задаваемых параметров .....	30
4.3 Подпрограмма вывода задаваемых параметров .....	35
4.4 Подпрограмма управления поворотом ствола.....	38
4.5 Вспомогательные подпрограммы .....	40
5 Математическое моделирование .....	41
Заключение .....	51
Список используемой литературы .....	52
Приложение А. Текст программы .....	55
Приложение Б Перечень элементов .....	70

## Введение

Учебные стенды - это специализированные учебные пособия, предназначенные для проведения практических занятий.

Учебные стенды играют важную роль в вопросах повышения качества образования.

Из всего многообразия учебных стендов был выбран стенд для изучения полета тела, так как он является наглядным и интересным.

В последствии речь пойдет именно о таких стендах.

Эволюция стендов вобрала в себя массу достижений в части конструирования и дизайна.

Существуют механические (баллистический пистолет), электрические (пушка Гаусса) стенды.

Одной из систем, предназначенной для проведения практических работ, является система электромеханическая.

Система основана на следующем принципе действия: обучаемому человеку (группе лиц) предлагается выполнение задач, по учебной программе (задачи зависят от учебной программы). Перечень задач, выполняемых с помощью данной системы:

-Выполнение физических опытов по изучению полета тела при различных начальных параметрах (школа предмет Физика)

-Программирование устройства системы с целью практического освоения программирования (ВУЗ предмет ОМПП или подобный)

-Выполнение опытов по прогнозированию полета тела при различных начальных параметрах (ВУЗ предмет Физика)

Эта система может пригодиться там, где необходимо продемонстрировать работы, выполняемые студентами в ходе обучения (День открытых дверей). Применение этой системы приведет к дополнительным удобствам, например, отпадает необходимость приобретения нескольких устройств, используемых для повышения

наглядности обучения.

Эти удобства обеспечиваются универсальностью системы, каждому обучаемому выдается персональное задание, выполнение которого дает ему возможность лучше понять и глубже освоить предмет учебной программы. Проанализировав основной недостаток механических стендов – низкая степень точности измерений(при достижении высокой точности расходуется много сил и времени), предполагается установить специальную панель, позволяющую человеку с высокой точностью вводить задаваемые параметры и контролировать правильность введенных параметров на протяжении всего времени их ввода и после него.

## **1 Состояние вопроса**

### **1.1 Анализ исходных данных и известных решений**

С древнейших времен учебные пособия играют важную роль в вопросах обучения людей различным навыкам.

Стенды по способу установки делятся на стационарные и мобильные. Мобильные стенды не имеют конкретного места установки и могут, при необходимости быть перенесены на другое место. Стационарные стенды устанавливаются на конкретном месте и требуют больше времени на установку.

Стенды по принципу действия делятся на механические, электрические, электронные и т.д.

#### **1.1.1 Механические стенды**

Одни из самых распространенных и привычных в настоящее время стендов – механические. По принципу устройства механические стенды подразделяются на категории в зависимости от вида выполняемых на стенде задач:

1 Стенды для постановки фронтальных лабораторных работ и физического практикума по механике “Изучение движения тела, брошенного горизонтально” пример стенда изображен на рисунке 1.1;

2 Стенды для демонстрации движения тел, брошенных под различными углами к горизонту пример стенда изображен на рисунке 1.2.

В стендах, предназначенных для постановки опытов по изучению движения тела, брошенного горизонтально [1], нет механизма изменения угла бросания тела. Придание телу начальной скорости производится за счет силы тяжести (редко пружины). Начальные параметры задаются вручную. Дальность полета тела (кроме стендов с пружинами) прямопропорциональна габаритам стенда. Начальная скорость полета сильно зависит от сил трения,

а, следовательно, от формы тела (максимальна, если тело не скользит, а катится по поверхности желоба).

Тело в полете имеет четкую направленность, что облегчает снятие параметров полета. Такие стенды в большинстве случаев мобильны и просты в эксплуатации.

В стендах, предназначенных для постановки опытов по изучению движения тела, брошенного под углом к горизонту [2], есть механизм изменения угла бросания тела. Выставление угла бросания производится вручную и при необходимости повышения точности занимает больше времени. Дальность полета тела прямопропорциональна жесткости пружины. Тело в полете имеет нечеткую направленность, что обусловлено отсутствием направляющих тела при его разгоне, это затрудняет измерение параметров полета.

Такие стенды мобильны.



Рисунок 1.1 – Стенд для изучения движения тела, брошенного горизонтально.





Рисунок 1.2 – Стенд для изучения движения тела, брошенного под углом к горизонту.

### **1.1.2 Электрические стенды**

Электрические стенды представляют собой устройства, управляемые при помощи электричества. Управляющие сигналы формируются при нажатии пользователем кнопок, рычагов и т.п.

Примером такого стенда может быть пушка Гаусса [3]. Принцип ее действия можно описать так: в цилиндрической обмотке (соленоиде) при протекании через нее электрического тока возникает магнитное поле. Это магнитное поле начинает втягивать внутрь соленоида железный снаряд, который от этого начинает разгоняться. Если в тот момент, когда снаряд окажется в середине обмотки ток в последней отключить, то втягивающее магнитное поле исчезнет и снаряд, набравший скорость, свободно вылетит через другой конец обмотки. Чем сильнее магнитное поле и чем быстрее оно отключается – тем сильнее вылетает снаряд.

Электрическая принципиальная схема пушки Гаусса представлена на рисунке 1.3

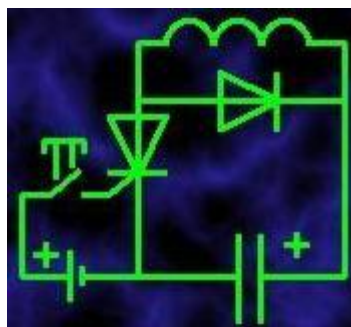


Рисунок 1.3 - Электрическая принципиальная схема пушки Гаусса  
Стенд пушка Гаусса представлен на рисунке 1.4

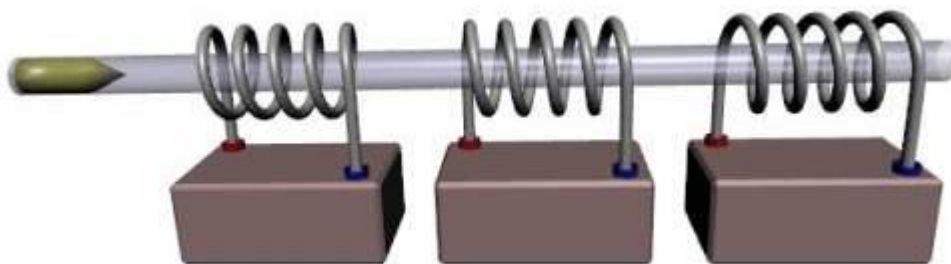


Рисунок 1.4 – Пушка Гаусса

## 1.2 Формулировка актуальности, цели и задачи работы

### 1.2.1 Развитие систем бросания тела

Приспособления для обучения начали применять задолго до рождения Христова. Самым эффективным метательным оружием того времени был лук [4]. Время его рождения относится к эпохе мезолита — от десяти до пяти тысяч лет до нашей эры. При обращении с метательным оружием нужны были определенные навыки. Эти навыки получались как в ходе охоты, так и в ходе тренировок. Для проведения тренировок нужны были оружие и мишени. Когда, например при «нулевой» подготовке, использование оружия было невозможно (не хватало сил и т.п.), оружие заменялось на похожее оружие (макет) с более простой эксплуатацией и пониженными полезными характеристиками. Например, маленькие, до 1 м длиной, детские луки из упругого можжевельника ученые нашли при раскопках Старой Ладogi, Новгорода, Старой Руссы и других городов России. Мишени, совместно с «макетами» и были первыми

приспособлениями для обучения. С накоплением научных достижений совершенствовалась и материальная база. Применение новых технологий и материалов позволило создавать более эффективное оружие.

Изображение лука приведено на рисунке 1.5



Рисунок 1.5 – Усиленный японский лук

Появление арбалета (рисунок 1.6) по своему значению может посоперничать с изобретением лука [5]. В те времена техника стрельбы из лука предполагала спускать тетиву от уха, а не от носа, как это делается сейчас, в связи с чем целиться в цель, в нашем понимании данного процесса, было по сути дела невозможно и для того что бы действительно научиться не просто стрелять но и попадать необходимо было регулярно и подолгу тренироваться, начиная с детства, вырабатывая в себе особое "чутье", чтобы пускать стрелу практически навскидку, а к совершеннолетию выпускать в минуту до десятка стрел на расстояние порядка 200 шагов(это справедливо для европейских лучников, в частности англичан, почти повсеместно использующих простые луки, что же касается восточных народов, применявших т.н. скифский(составной) лук, то там нередко были случаи когда опытный лучник посылал стрелы более чем на 600 метров а то и далее), естественно, услуги таких стрелков обходились недешево. Но с появлением арбалета любой человек, обладая лишь элементарными

навыками стрельбы, мог соревноваться в меткости с профессиональным лучником, а по поражающему действию стрелы и превзойти его. Макетирование арбалетов не было необходимо для учебных целей.

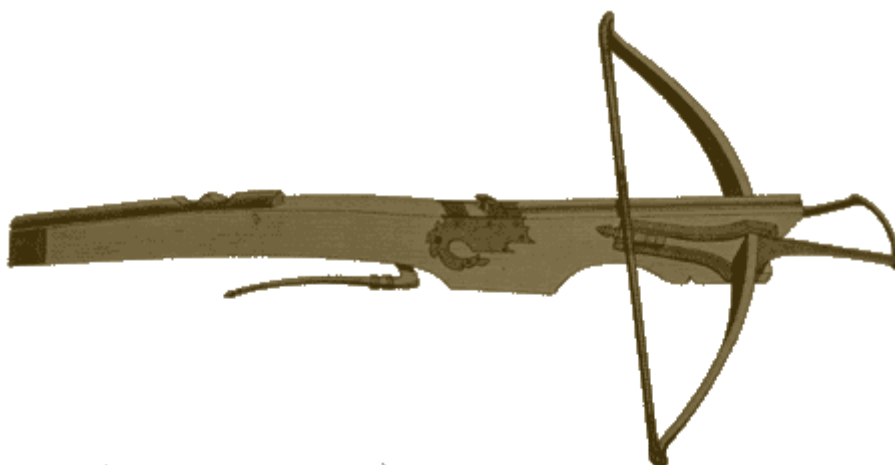


Рисунок 1.6 – Арбалет

Наряду с ручным метательным оружием создавалось и стационарное. Первым стационарным метательным оружием была катапульта [6].

Катапультами называли древние метательные машины, которые в зависимости от конструкции могли стрелять копьями, стрелами или камнями. Впервые начали применяться с 5 века до н.э. в Греции, а затем практически во всех странах Европы вплоть до 15 века. Принцип действия древних катапульти основывался либо на принципе лука (баллисты и скорпионы), либо на принципе пращи. На рисунке 1.7 изображена катапульта, действующая по второму принципу.

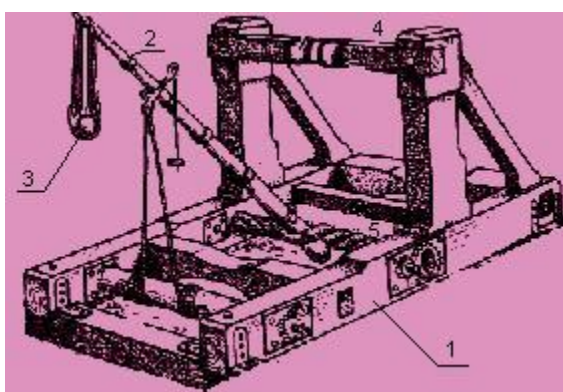


Рисунок 1.7 – Катапульта

Обучение стрельбе из катапульти проходило на штатных орудиях без применения макетов.

На смену катапультам пришли пушки (рисунок 1.8).

До второй половины XIX века в основном применялись орудия гладкоствольные с зарядкой через дульную часть, хотя это вовсе не означает, что не было технологий изготовления казенозарядных типов орудий, просто в то время такие орудия оказались неудобными и сложными в изготовлении [7]. Их время придет только во второй половине XIX века. Из-за отсутствия нарезов в канале ствола, которые придавали бы снаряду вращение в полете, повышающее точность попадания в цель, и большого зазора между каналом и ядром дальность полета ядра была невелика и попадания из пушек на дистанцию более 150 -200 метров были редки, а из мортир — и вовсе случайны.

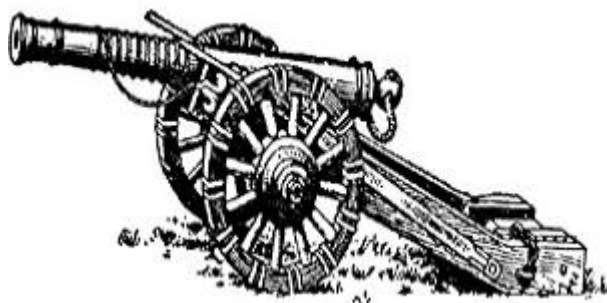


Рисунок 1.8 – Пушка

К середине XVIII века скорострельность, точность эффективность артиллерии заметно улучшились, а к эпохе Наполеоновских войн — уже достигла существенного прогресса. Например, 32-фунтовые пушки уже производил один выстрел каждые 2 минуты, при этом дальность прямого выстрела составляла около 350 метров, а железное ядро на этой дистанции пробивало до 106 см дубового массива. То есть, фактически, любое попадание с такой дистанции наносило очень большое повреждение. Для пушек стали применяться таблицы стрельбы. Развивалось обучение стрельбе.

Советская артиллерия была создана в годы гражданской войны и в своем предвоенном развитии прошла два этапа [8]. В период с 1927 по 1930 гг. была проведена модернизация доставшегося в наследство от царской

армии артиллерийского вооружения, в результате которой существенно улучшились основные тактико-технические характеристики орудий в соответствии с новыми требованиями, причем это было сделано без больших затрат, на базе имевшегося вооружения. Благодаря модернизации артиллерийского вооружения дальность стрельбы артиллерии возросла в среднем в полтора раза. Увеличение дальности стрельбы было достигнуто за счет удлинения стволов, увеличения зарядов, увеличения угла возвышения и улучшения формы снарядов. Увеличение мощности выстрела потребовало также некоторой переделки лафетов. В лафет 76-мм пушки обр. 1902 г. был введен уравнивающий механизм, у 107- и 152-мм пушек поставлены дульные тормоза. Для всех орудий был принят единый прицел образца 1930 г. После модернизации орудия получили новые наименования: 76-мм пушка образца 1902/30 гг., 122-мм гаубица обр. 1910/30 гг. и т.д.

Из новых образцов артиллерии, разработанных в этот период, на вооружение была принята 76-мм полковая пушка обр. 1927 г.

Начало второго этапа в развитии советской артиллерии относится к началу 30-х гг., когда в результате ускоренного развития тяжелой индустрии появилась возможность начать полное перевооружение артиллерии новыми образцами.

Вторая мировая война оказала огромное влияние на развитие артиллерии.

Объектом научных достижений того времени можно считать, например, советскую 100-мм пушку (рисунок 1.9)



Рисунок 1.9 – Советская 100-мм пушка БС-3 образца 1944 года

Проанализировав различные способы придания телу начальной скорости за продолжительный исторический период (сила упругости: луки, арбалеты, катапульты; химическая реакция – пушки), остановились на использовании силы упругости.

### **1.2.2 Технология обучения**

Развивающее обучение сегодня рассматривается как историческая перспектива которая заложит основания не только для развития системы образования но и общества [9]. Появляющиеся новые системы и модели обучения в своих построениях отталкиваются от идеологии развивающего обучения.

Теория развивающего обучения сделала огромный прорыв в педагогике, позволив соединить в единый процесс получения знаний, предметное и психологическое развитием самопознание человека. Это удалось сделать благодаря учебной деятельности Учебная деятельность составляет ядро развивающего обучения.

На первый план в учебной (поисково-исследовательской) деятельности выдвигается деятельность по организации учащимися собственного познания. Важной стороной поисково-исследовательской деятельности выступает освоение процессуальной стороны поисково-исследовательской деятельности. Освоение процедур поисковой учебной деятельности становится тем фактором, который обуславливает развитие и саморазвитие человека. Наряду с предметно-содержательными результатами (предметные знания, умения, способы решения проблем) особым результатом становится рефлексивно осмысляемый опыт поисковой деятельности.

Эту сторону учебной деятельности называют надпредметной, потому, что она определяет развитие многих других способностей человека, его успех в любой другой деятельности, в т.ч. профессиональной. С ними связано развитие качеств творческого субъекта.

В настоящее время активно ведутся поиски новых форм учебной деятельности. В плане прогнозирования эволюции учебной (поисково-исследовательской) деятельности сегодня все более активно обоснуются такие ее формы, как: моделирование, конструирование и проектирование. Характерной особенностью этих видов деятельности выступает не изучение того, что существует, а создание новых продуктов и одновременно познание того, что лишь может возникнуть. Возможной целью учебного заведения в логике и технологии развивающего обучения может быть овладение самостоятельной постановкой учебных целей, осваивание пространства возможностей конструируемых способов действия.

Учебное моделирование отличается от других видов моделирования тем, что связано с познавательными процессами, направлено на раскрытие способов познания или на фиксацию раскрытых путем исследования отношений, связей между объектами и внутри объекта.

Компьютеры - универсальные обработчики цифровой информации пришли в образование около двадцати лет назад и в настоящее время занимают в нем центральное место.

Однако компьютер не может решить всех задач, поставленных в ходе обучения, без периферийных устройств. Как раз периферии-то сейчас появилось огромное количество, причём, что очень важно, она стала широко доступной.

Деятельность образовательных учреждений направлена на комплексное внедрение всех современных технологий в образовательный процесс.

Образование, особенно физическое, не может быть «вещью в себе». Приёмы и методики должны гибко реагировать как на новые достижения науки и техники, приборы, так и на новые возможности, которые предоставляют эти достижения и приборы.

Проанализировав историю развития стендов, пришли к выводу, что наиболее лучшим для целей создания систем моделирования полета тела



является стенд электромеханического типа.

Целью ВКР является разработка стенда электромеханического типа с микроконтроллерным управлением.

В ходе работы необходимо решить ряд задач: подобрать механическую и электромеханическую части стенда. Разработать электрическую часть стенда, а в частности: формирователь поворотного импульса, систему идентификации, вспомогательные схемы. А также необходимо составить общую блок-схему программы и написать ряд подпрограмм. Необходимо проверить безопасность и экологичность, а также эффективность работы.

### **Выводы**

В данном разделе сформулированы актуальность, цели и задачи работы. Произведен анализ исходных данных и известных решений.

## 2 Механическая часть станда

Разработка и изготовление механической части станда не является задачей данной ВКР. Для общего понимания работы станда разработали кинематическую схему (рис. 2.1).

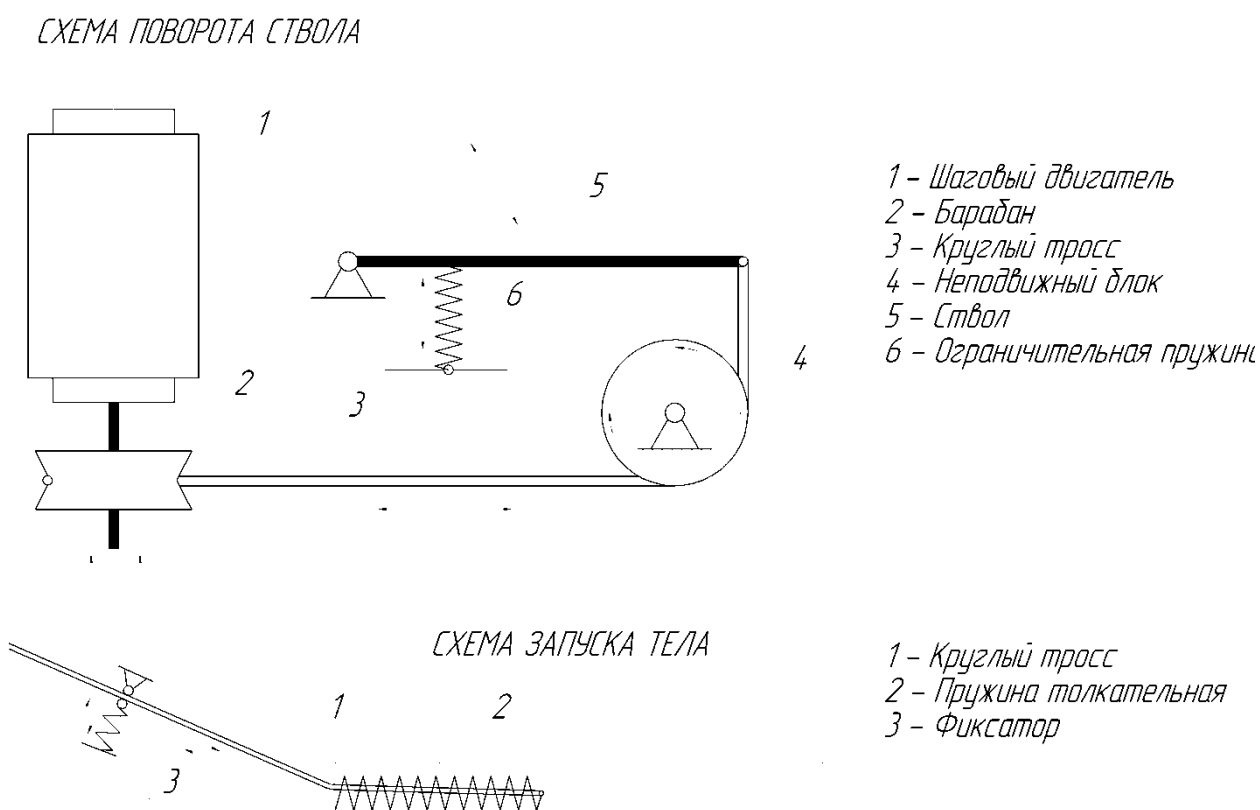


Рисунок 2.1 – Кинематическая схема станда

Согласно этой схеме изменение вертикального угла наклона ствола осуществляется шаговым двигателем 1, а непосредственный запуск снаряда – пружиной 2.

### Выводы

В данном разделе приведена кинематическая схема станда.

## **3 Электрическая часть стенда**

### **3.1 Обзор вариантов**

#### **3.1.1 Создание блока управления**

Управляющий блок должен: иметь систему хранения данных, производить логические и алгебраические операции, иметь порты вывода для подключения внешних устройств, систему программирования, систему ввода задаваемых параметров, систему индикации задаваемых параметров.

##### *3.1.1.1 Система выполнения математических операций*

Функции производства математических операций реализуем на основе микроконтроллера. Среди рассмотренных вариантов микроконтроллера выделим AT89S8252. AT89S8252 – потребляющий небольшую мощность, высокопроизводительный комплементарный металлооксидный полупроводниковый (КМОП) 8-битный микрокомпьютер с 8 килобайтами перезаписываемой флеш памяти, программируемой и стираемой ПЗУ и 2 кБ EEPROMа. Однокристалльная перезаписываемая флеш позволяет программной памяти быть перепрограммированной в системе через последовательный интерфейс SPI или стандартным энергонезависимым способом. Благодаря объединению многофункционального 8ми битного CPU с перезаписываемой флеш памятью на монолитном чипе Atmel AT89S8252 – мощный микрокомпьютер, который обеспечивает гибкие и эффективные решения для управления многими вложенными приложениями.

##### *3.1.1.2 Система программирования*

Программирование будем осуществлять через последовательный интерфейс SPI, предложенный разработчиками микроконтроллера. Программатор реализуем на основе микросхемы MAX232, которая поддерживает интерфейс SPI и посредством COM порта может быть подключена к персональному компьютеру

### 3.1.1.3 Система ввода задаваемых параметров [11]

Вводить задаваемые параметры удобнее всего с клавиатуры. Среди рассмотренных вариантов выделим клавиатуру 3×4 АК-304-N-BBW, представленную на рисунке 3.1. Число кнопок которой будет достаточно для ввода задаваемых параметров.



Рисунок 3.1 - Клавиатура

### 3.1.1.4 Система индикации задаваемых параметров

Удобнее всего для индикации задаваемых параметров воспользоваться семисегментными индикаторами. В нашем случае необходимо и достаточно для индикации использовать три семисегментных индикатора. Используем блок из трех семисегментных индикаторов ВА 56\_12.

### 3.1.2 Создание формирователя импульсов для шагового двигателя

Существует несколько способов управления фазами шагового двигателя. Один из простейших способов обеспечивается попеременной коммутации фаз, при этом они не перекрываются, в один момент времени включена только одна фаза (рис.3.5). Этот способ называют "one phase on"

full step или wave drive mode. Точки равновесия ротора для каждого шага совпадают с «естественными» точками равновесия ротора у незапитанного двигателя. Диаграммы управления шаговым двигателем при полношаговом режиме представлены на рисунке 3.2.

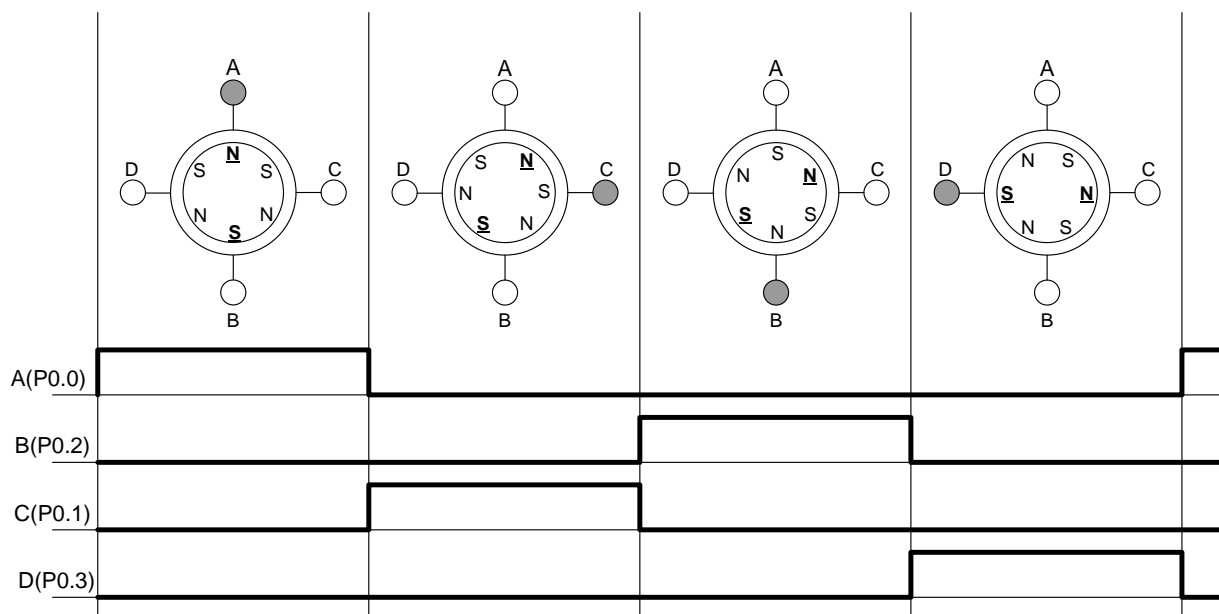


Рисунок 3.2 - Диаграммы управления при полношаговом режиме

В нашей разработке используем именно этот способ

### 3.2 Разработка электрической схемы

#### 3.2.1 Создание блока управления [10]

##### 3.2.1.1 Система выполнения математических операций

Микроконтроллер был выбран в пункте 3.1.1.1.

Для работы микроконтроллера к его входам BQ1 и BQ2 подключить кварцевый генератор.

Собрали кварцевый генератор на элементах  $C_7$ ,  $C_9$  и  $ZQ_1$  на частоту 12мГц.

Выбрали кварцевый резонатор  $ZQ_1$  (HC-49S-12M). Номинальные значения конденсаторов  $C_7$ ,  $C_9$  возьмем из справочных данных на данный резонатор.  $C_7$ : SMD-1206-50B-30мкФ,  $C_9$ : SMD-1206-50B-30мкФ.

### 3.2.1.2 Система программирования

Программатор выполнен на основе микросхемы MAX232 напряжение питания у которой -0,3- +6В (имеется + 5 В). Ток питания 5 мА, поддерживающей интерфейс RS232. Элементы обвязки микросхемы:  $C_1$ ,  $C_2$ ,  $C_5$ ,  $C_6$  выберем по рекомендации на микросхему.  $C_1$ : K10-17Б -40В-1мкФ,  $C_2$ : K10-17Б -40В-1мкФ,  $C_5$ : K10-17Б -40В-1мкФ,  $C_6$ : K10-17Б -40В-1мкФ.

Выберем транзистор для регулирования подачи сигнала на вывод RST микроконтроллера, подаваемый с микросхемы MAX232 в процессе программирования КТ3107 с подключенным к базе токоограничительным резистором  $R_4$  1кОм.

Токоограничительный резистор  $R_5$  возьмем 10кОм.

$C_8$  выберем по рекомендации на микроконтроллер  $C_8$ : K10-17Б -40В-1мкФ

В качестве индикаторов процесса программирования и включенного питания используем светодиоды  $VD_1$ –  $VD_3$ . Выберем светодиоды L-132-XIT с током свечения  $I_{прVD_1-VD_3} = 10\text{мА}$ .

Рассчитаем сопротивление резисторов  $R_1$ -  $R_3$

$$R_{1-3} = U_{num} / I_{прVD_1-VD_3} = 5 / 0,01 \approx 470\text{Ом}, \quad (3.1)$$

Для стабилизации параметров питания выберем конденсаторы  $C_3$  и  $C_4$   $C_3$ : K10-17Б -40В-1мкФ,  $C_4$ : K50-35 -50В-1мкФ.

### 3.2.1.3 Система индикации задаваемых параметров

Блок индикации состоит из трех семисегментных индикаторов. Блоком управляют регистры и транзисторные ключи. Так, при подаче логической единицы на вход  $DD_2(11)$  разрешается трансляция данных P0.0 – P0.7 с микроконтроллера на выхода  $DD_2$ . При подаче на базы транзисторов  $VT_5$  –  $VT_7$  логического нуля транзисторы переходят в режим насыщения и примерно +5 В передается на семисегментный индикатор ( $HG_1$ ), тем самым

включая сегменты трехразрядного индикатора. При подаче нуля на вход DD<sub>2</sub>(11) происходит защёлкивание регистра и запрет трансляции данных.

В качестве регистра DD<sub>2</sub> выберем регистр 74ACT373CN20A, напряжение питания у которого -0,5 - +7 В (имеется + 5 В). Ток питания 20 мА. А в качестве индикатора – красный семисегментный индикатор ВА56-12SRWA, напряжение питания которого 5 В и ток питания  $I_{питHG1}=30$  мА.

Выбираем резисторы R<sub>14</sub> – R<sub>24</sub>:

$$R_{14-24} = U_{num} / I_{numHG1} = 5 / 30 \cdot 10^{-3} \approx 180 \text{ Ом.} \quad (3.2)$$

Транзисторы VT<sub>6</sub> – VT<sub>8</sub> выбираем с большим коэффициентом усиления по току так как управляющий ток с микросхемы небольшой. Выбрали транзистор КТ973А, с коэффициентом усиления по току  $\beta_{VT3-VT5}=750$ . Необходимо для питания семисегментного индикатора подать на него  $I_{кVT5-VT7}=240$  мА. Рассчитаем ток базы транзисторов VT<sub>6</sub> – VT<sub>8</sub>:

$$I_{бVT6-VT8} = I_{кVT6-VT8} / \beta_{VT6-VT8} = 240 \cdot 10^{-3} / 750 = 0,32 \cdot 10^{-3} \text{ А,} \quad (3.3)$$

Рассчитаем резисторы R<sub>18</sub>-R<sub>20</sub>:

$$R_{18-20} = U_{num} / I_{бVT6-VT8} = 5 / 0,32 \cdot 10^{-3} \approx 16 \text{ кОм,} \quad (3.4)$$

Возьмем резисторы R<sub>18</sub> R<sub>20</sub> = 2,2 кОм (МЛТ-0,125 2,2кОм), чтобы транзисторы VT<sub>6</sub>– VT<sub>8</sub>быстрее входили в режим насыщения.

### 3.2.2 Создание формирователя импульсов для шагового двигателя

При подаче на биты P0.0 - P0.4 порта P0 в заданной последовательности напряжения высокого уровня (2.4 В) транзисторы VT<sub>2</sub> – VT<sub>5</sub> открываются и переходят в режим насыщения. Напряжение +5 В прикладывается к обмоткам шагового двигателя. Транзисторы VT<sub>2</sub> – VT<sub>5</sub> выбираем с большим коэффициентом усиления по току так как управляющий ток с микросхемы небольшой. Выбрали транзисторы КТ829Б, с коэффициентом усиления по току  $\beta_{VT2-VT5}=750$ . Необходимо для работы шагового двигателя подавать на его обмотки в заданной последовательности  $I_{кVT2} - I_{кVT5} = 0,4$  А. Рассчитаем ток базы транзисторов VT<sub>2</sub> – VT<sub>5</sub>:

$$I_{бVT2-VT5} = I_{кVT2-VT5} / \beta = 0,4 / 750 = 0,53 \cdot 10^{-3} \text{ А,} \quad (3.5)$$

Рассчитаем резисторы  $R_6, R_7, R_{10}, R_{11}$ :

$$R_{6,7,10,11} = U_{num} / I_{\sigma VT_2-VT_5} = 5 / 0,53 \cdot 10^{-3} = 9,4 \text{ кОм}, \quad (3.6)$$

Возьмем резисторы  $R_6, R_7, R_{10}, R_{11} = 2,2 \text{ кОм}$  (МЛТ-0,125 2,2кОм). Выберем заведомо намного меньше, чтобы транзисторы  $VT_2 - VT_5$  быстрее входили в режим насыщения.

Резисторы  $R_6 - R_{13}$  необходимы для корректной работы порта P0

Возьмем резисторы  $R_8, R_9, R_{12}, R_{13} = 2,2 \text{ кОм}$  (МЛТ-0,125 2,2кОм): такие же, как  $R_6, R_7, R_{10}, R_{11}$ .

Диоды  $VD_4 - VD_8$  выберем по току, проходящему через обмотки шагового двигателя. Возьмем диоды 1N4007 с большим запасом по току.

Схема лабораторного стенда электрическая принципиальная приведена на рисунке 3.3. Перечень элементов приведен в приложении.



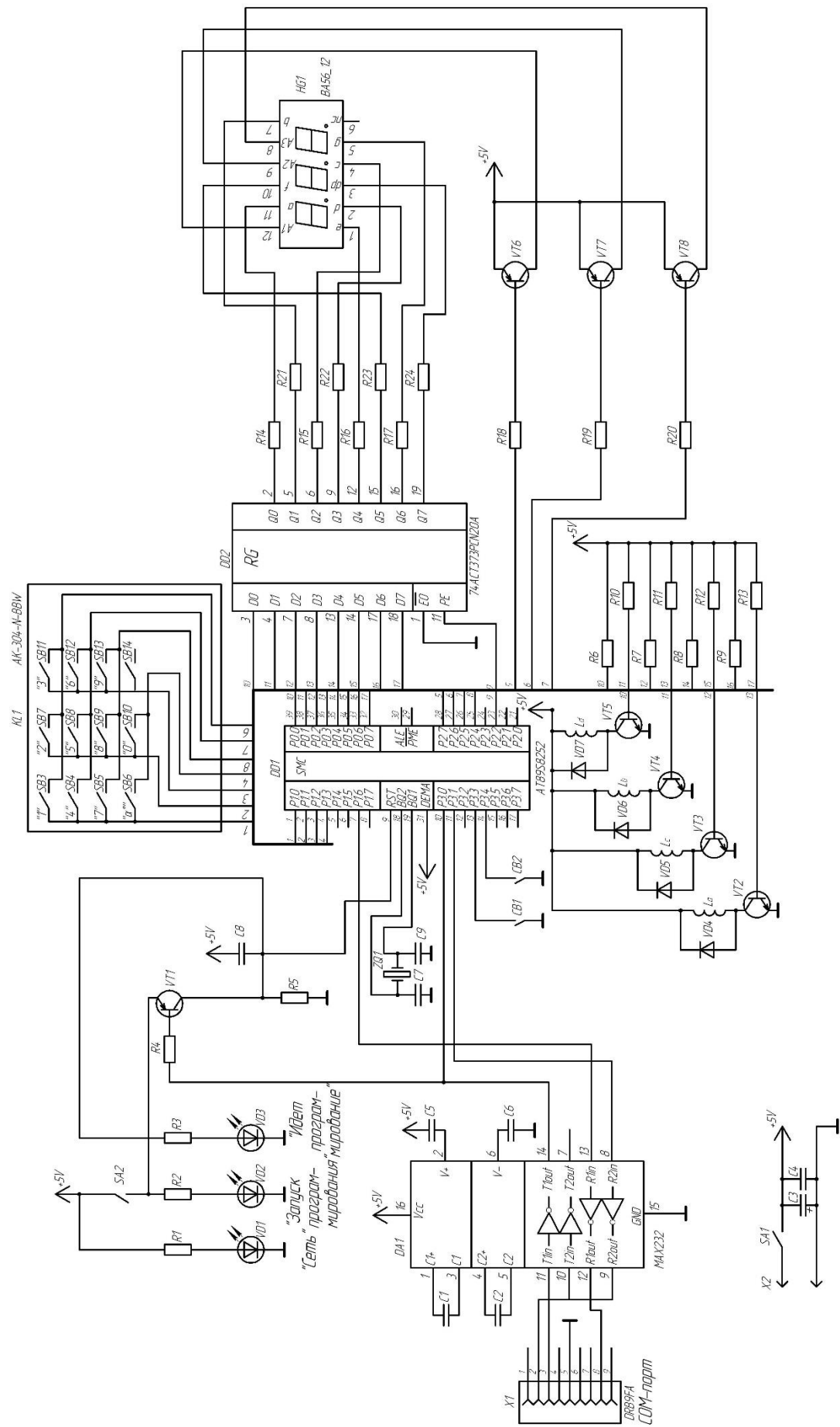


Рисунок 3.3 – Схема электрическая принципиальная

Управление шаговым двигателем может быть осуществлено с помощью формирователя импульсов (драйвера) (рисунок 3.4).

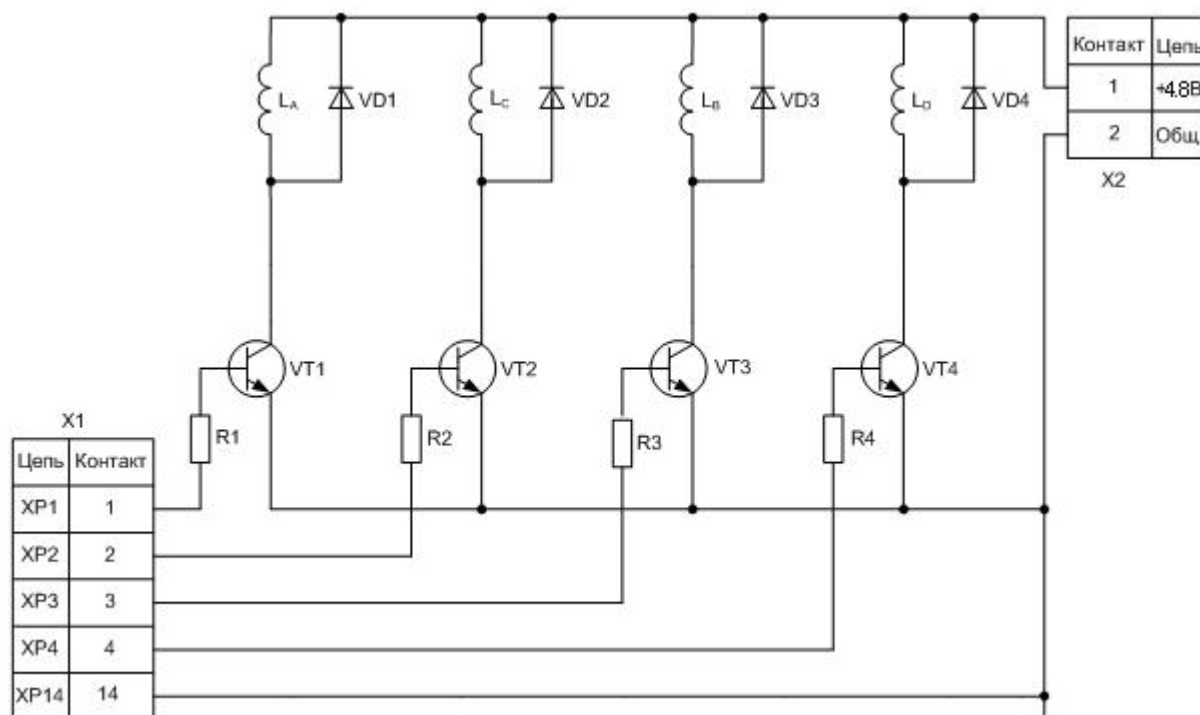


Рисунок 3.4 – Схема формирователя импульсов для шагового двигателя

В качестве системы управления формирователя импульсов для шагового двигателя можно воспользоваться стендом МКС-51 [12], как наиболее удобным и доступным, по программе подготовки инженеров по специальности «Промышленная электроника» ТГУ (рисунок 3.5). Стенд содержит микроконтроллер, программатор, порты вывода, систему индикации, систему ввода, питается от 5 Вольт, имеет прочный корпус и эстетический дизайн. Что обеспечивает возможность его применения в создаваемом устройстве без внесения изменений. Соединение блоков электрической схемы может быть произведено посредством имеющихся на стенде разъемов.



Рисунок 3.5 – Стенд МКС-51

### **Выводы**

В данном разделе произведена разработка электрической принципиальной схемы устройства, рассчитаны и выбраны ее элементы.

## **4 Программная часть**

Текст программы приведен в приложении.

### **4.1 Общая блок-схема программы**

Общая блок-схема программы приведена на рисунке 4.1.

Программа начинается с опроса клавиатуры

На блоке индикации отображается результат нажатых кнопок

Если все поля заполнены, проверяется, нажата ли кнопка «Наводить»

Если кнопка нажата, проверяется, набран угол или расстояние, если не нажата, продолжение индикации

После выявления единицы измерения вводимого параметра, начинается его обработка

Если значение в пределах нормы, запускается таймер, если нет, на блоке индикации высвечивается «ошибка» (Err)

Подпрограмма прерывания таймера поворачивает ствол на заданный угол, по одному градусу за срабатывание, после поворота ствола на заданный угол, таймер выключается и продолжает работать подпрограмма индикации

Инструкция по эксплуатации лабораторного стенда приведена в приложении.

При составлении программы использовалась литература [12], [13].

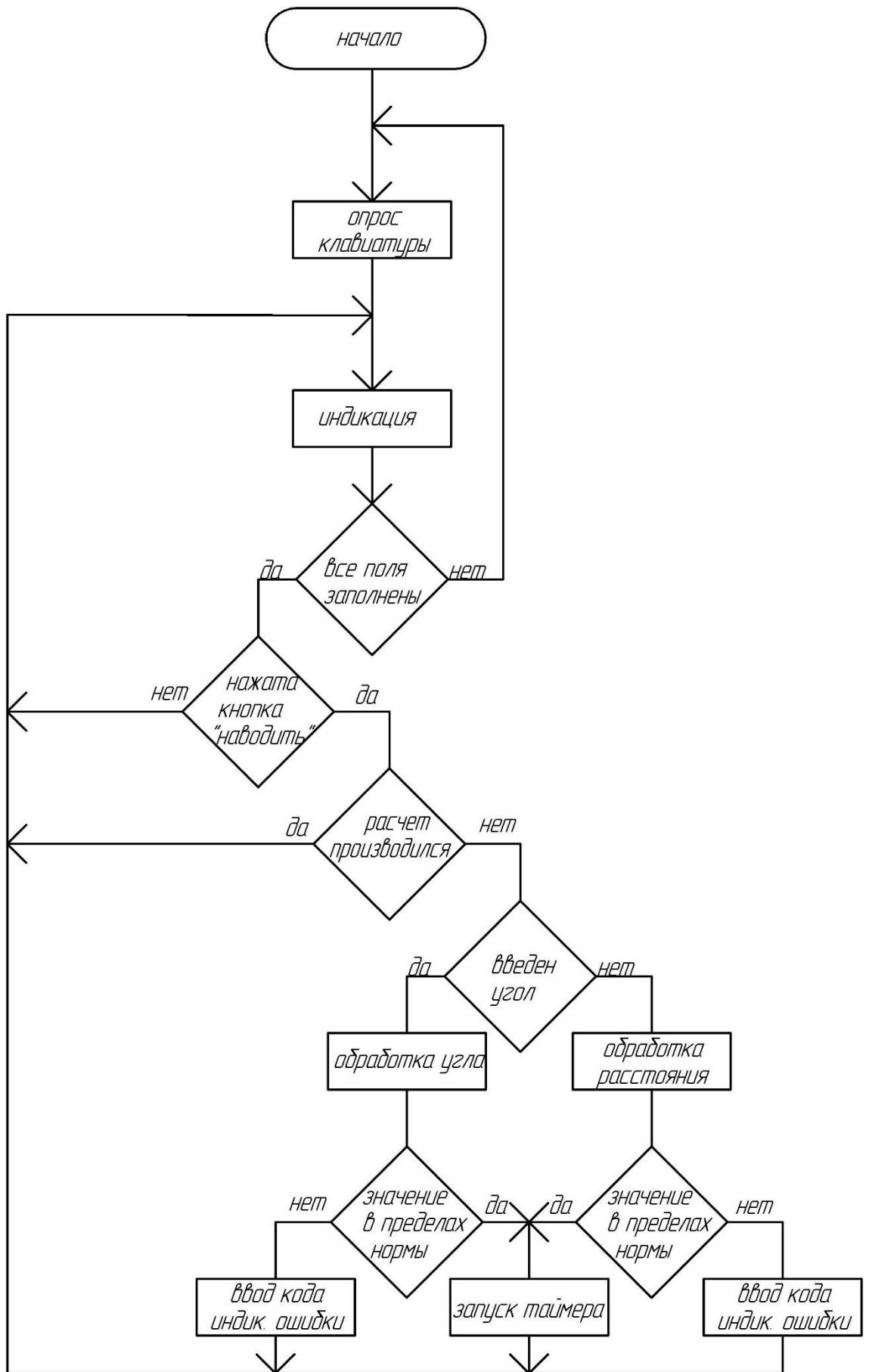


Рисунок 4.1 - Общая блок-схема программы

## 4.2 Подпрограмма идентификации задаваемых параметров

### 4.2.1 Опрос клавиатуры

Обнуляем бит, отвечающий за первую строку

Считываем данные бита отвечающего за первый столбец

Если он равен нулю, значит, нажата кнопка «1»

Если он не равен нулю, считываем данные бита отвечающего за второй столбец

Если он не равен нулю, считываем данные бита отвечающего за третий столбец

Повторяем со второй и третьей строкой, пока значение бита столбца не будет равно нулю

Информацию о нажатой кнопке (в рассмотренном примере «1») записываем в аккумулятор в виде кода символа «1» из базы данных

Переходим на метку сравнения информации в регистре, соответствующем крайнему левому знаку, вводимому с клавиатуры, с исходной, если совпадает, изменяем ее на информацию из аккумулятора

Выводим информацию о нажатой кнопке на индикатор

Проверяем, нажата ли кнопка, если «да», продолжаем индикацию

Если «нет», начинаем опрос клавиатуры

Цикл повторяется до введения трех символов с клавиатуры (второй символ, записывается в регистр соответствующий среднему знаку, вводимому с клавиатуры, третий – крайнему правому)

Блок – схема подпрограммы представлена на рисунке 4.2

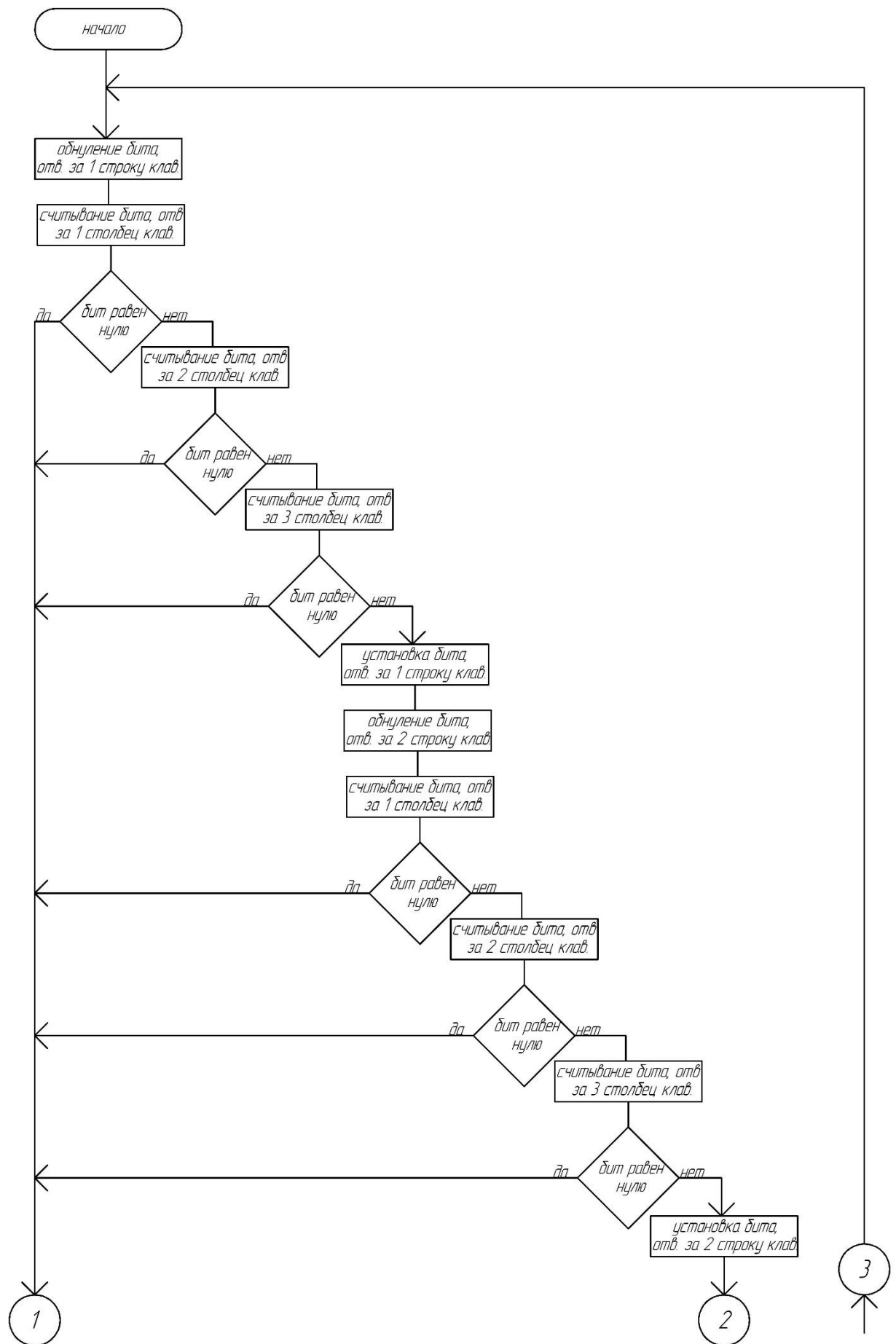


Рисунок 4.2

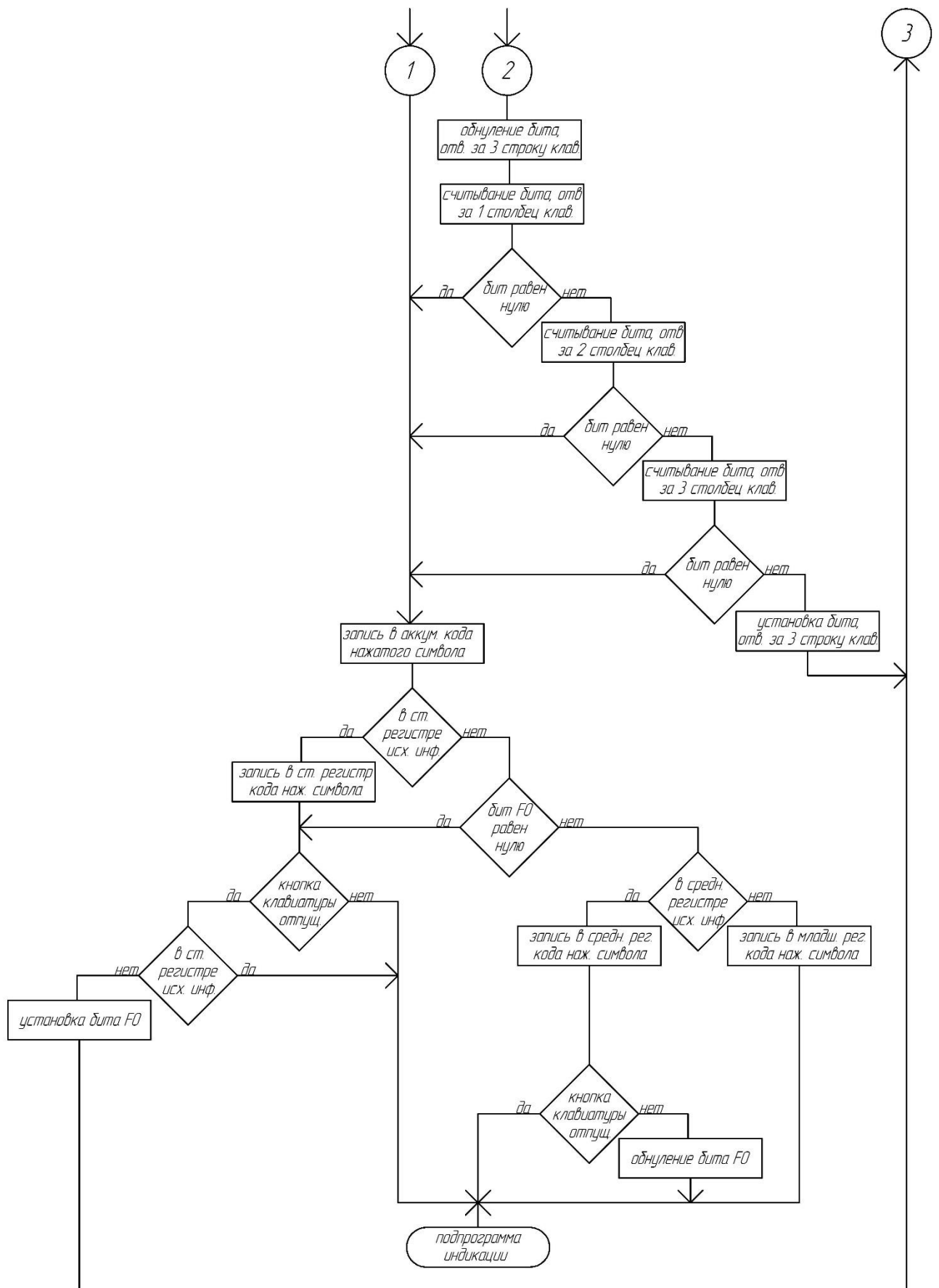


Рисунок 4.2 – Блок схема опроса клавиатуры



#### **4.2.2 Определение единиц измерения задаваемого параметра**

Если в регистре, соответствующем крайнему правому разряду вводимого кода, записан код символа «градус» из базы данных, то информация введена в градусах, если нет, в сантиметрах.

#### **4.2.3 Перевод набранного значения в угол поворота ствола**

Если информация введена в градусах, число, записанное в старшем разряде, умножается на 10 и складывается с числом, записанным в младшем разряде

Если введено число в пределах допустимой нормы, то вычитаем из него число, вводимое ранее (если оно было), информацию о результате записываем в регистр, отвечающий за поворот ствола.

Если введенное число выходит за пределы допустимой нормы, на индикаторе высвечивается «ошибка» (Err), информация в регистре, отвечающем за поворот ствола, не изменяется.

Если информация введена в сантиметрах, число, записанное в старшем разряде, умножается на 100 и складывается с числом, записанным в среднем разряде, умноженном на 10 и числом из младшего разряда.

Если введено число в пределах допустимой нормы, из базы данных в регистр, отвечающий за поворот ствола, извлекается количество градусов, на которое необходимо повернуть ствол, чтобы тело улетело на введенное расстояние, за вычетом ранее вводимого параметра (если он был).

Если введенное число выходит за пределы допустимой нормы, на индикаторе высвечивается «ошибка» (Err), информация в регистре, отвечающем за поворот ствола, не изменяется.

Блок – схема подпрограммы представлена на рисунке 4.3

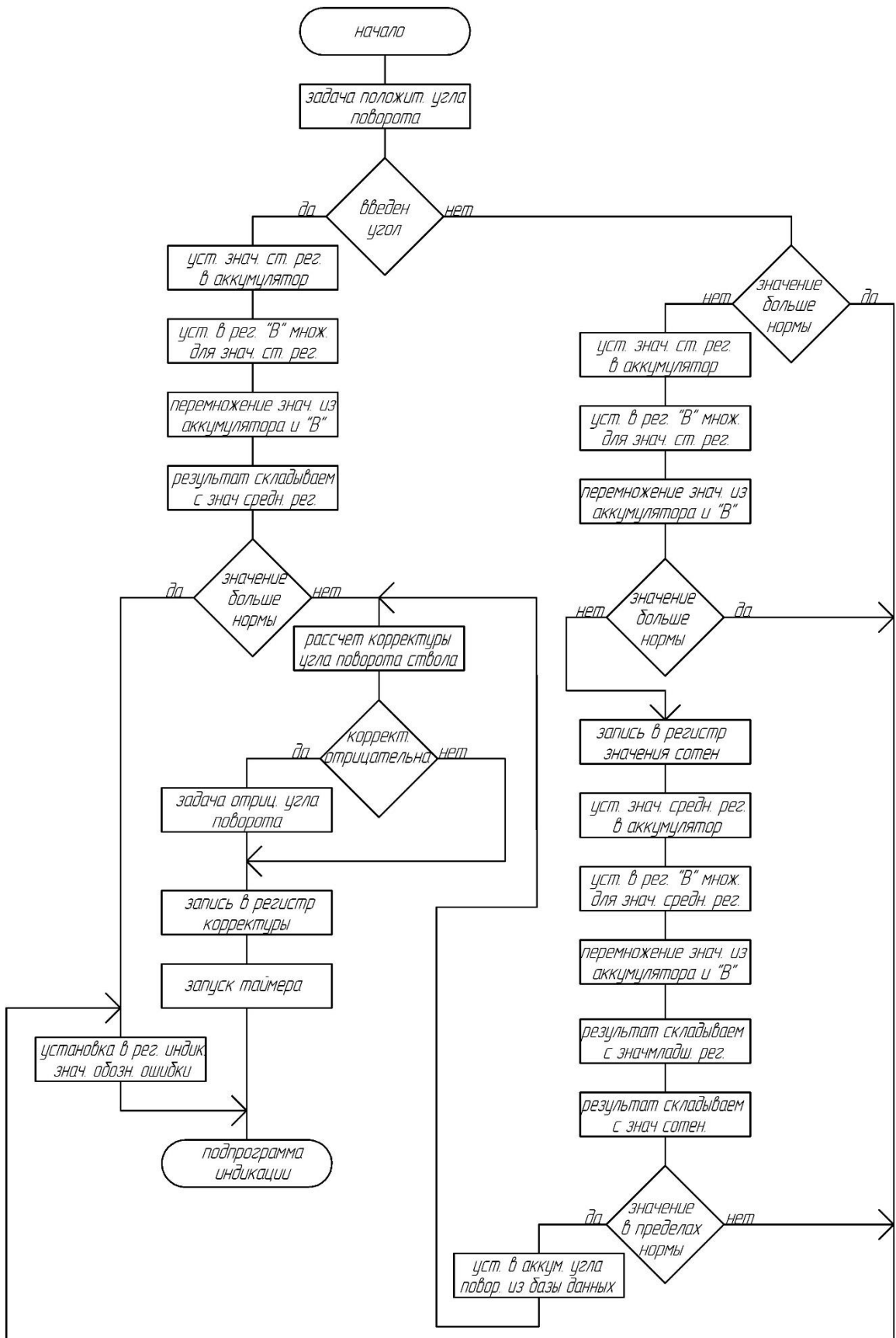


Рисунок 4.3 – Блок схема перевода набранного значения  
в угол поворота ствола

### 4.3 Подпрограмма вывода задаваемых параметров

Закрываем транзисторы, отвечающие за индикацию

Выставляем потенциал логической единицы на катоды светодиодов индикации

Открываем транзистор, отвечающий за индикацию старшего разряда

Из базы данных код цифры старшего разряда вводимого числа записываем в порт, соединенный с катодами светодиодов (индицируется цифра старшего разряда числа).

Закрываем транзистор, отвечающий за индикацию старшего разряда

Выставляем потенциал логической единицы на катоды светодиодов индикации

Открываем транзистор, отвечающий за индикацию среднего разряда

Из базы данных код цифры среднего разряда вводимого числа записываем в порт, соединенный с катодами светодиодов (индицируется цифра среднего разряда числа).

Закрываем транзистор, отвечающий за индикацию среднего разряда

Выставляем потенциал логической единицы на катоды светодиодов индикации

Открываем транзистор, отвечающий за индикацию младшего разряда

Из базы данных код цифры младшего разряда вводимого числа записываем в порт, соединенный с катодами светодиодов (индицируется цифра младшего разряда числа).

Закрываем транзистор, отвечающий за индикацию младшего разряда

Выставляем потенциал логической единицы на катоды светодиодов индикации

Если младшей разряд не равен исходному, то проверяем нажата ли кнопка «наводить»

Если нажата кнопка «сброс», переходим на опрос клавиатуры, выставив начальные параметры.

Блок – схема подпрограммы представлена на рисунке 4.4

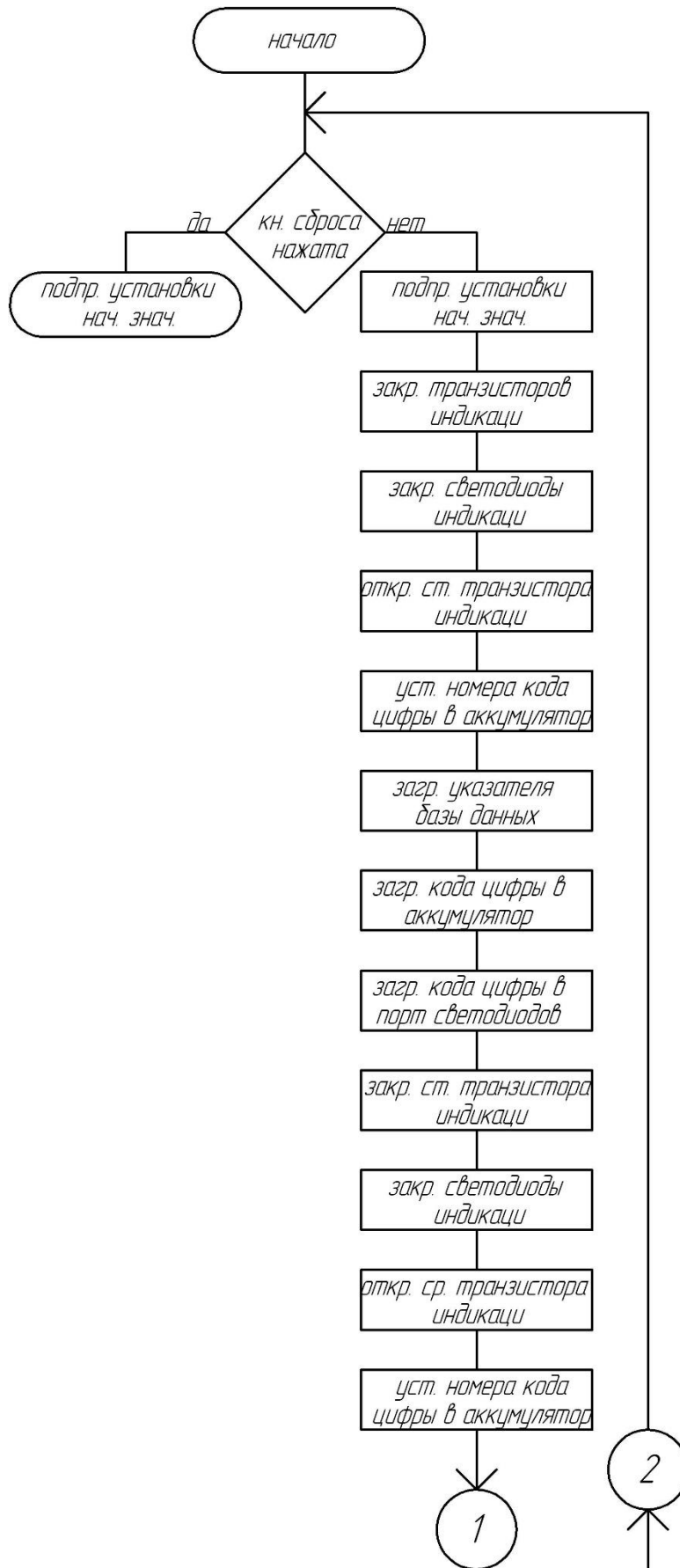


Рисунок 4.4

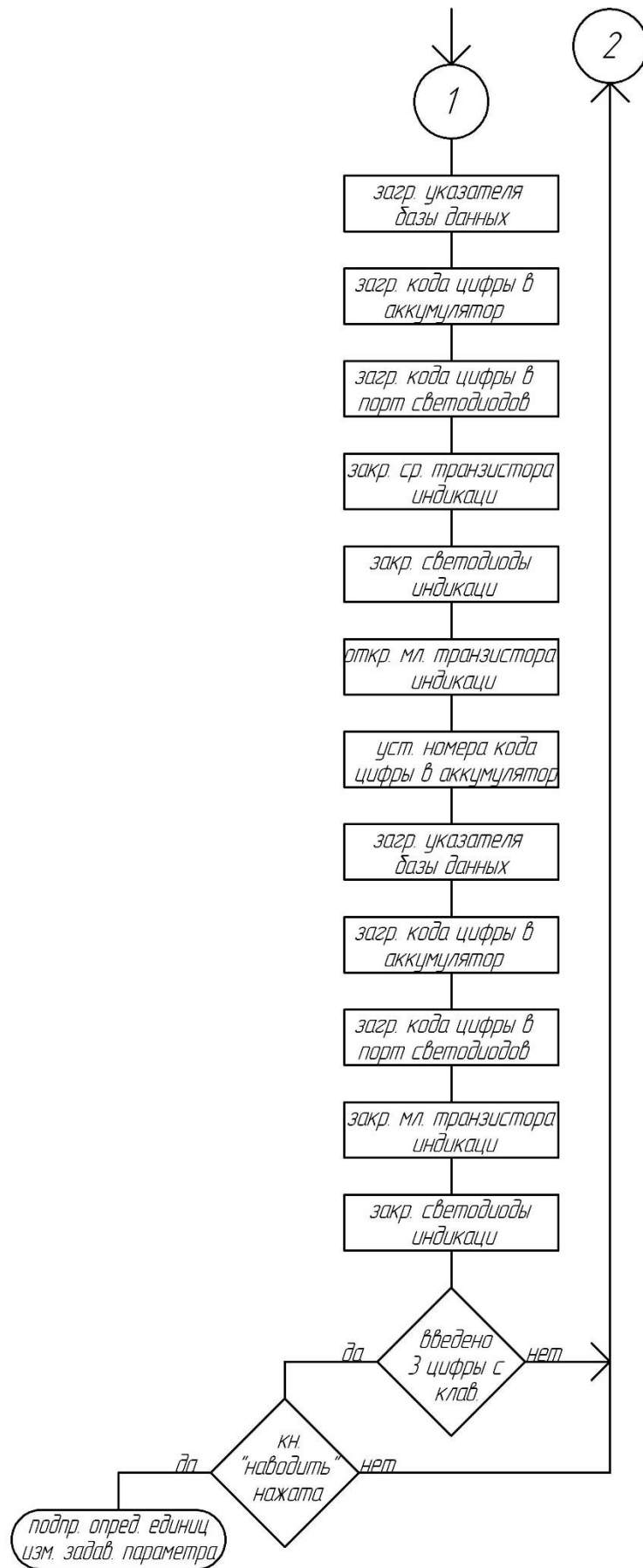


Рисунок 4.4 – Блок схема подпрограммы индикации

#### 4.4 Подпрограмма управления поворотом ствола

Расположена в подпрограмме прерывания таймера

При обнулении таймера она запускается

За один запуск программа подает один импульс на обмотку шагового двигателя

Программа предусматривает необходимую длительность импульса за счет вставки пустых циклов

Программа определяет, обмотку, на которую нужно подавать импульс по двум параметрам: предыдущий импульс и наличие (отсутствие) логической единицы в бите «F0», логическая единица при подъеме ствола, логический ноль при опускании

Наличие логической единицы в бите «F0» определяет подпрограмма обработки введенных значений

Таймер отключается при обнулении числа, обозначающего количество градусов, на которое нужно повернуть ствол, оно задается в подпрограмме обработки введенных значений

Блок – схема подпрограммы представлена на рисунке 4.5

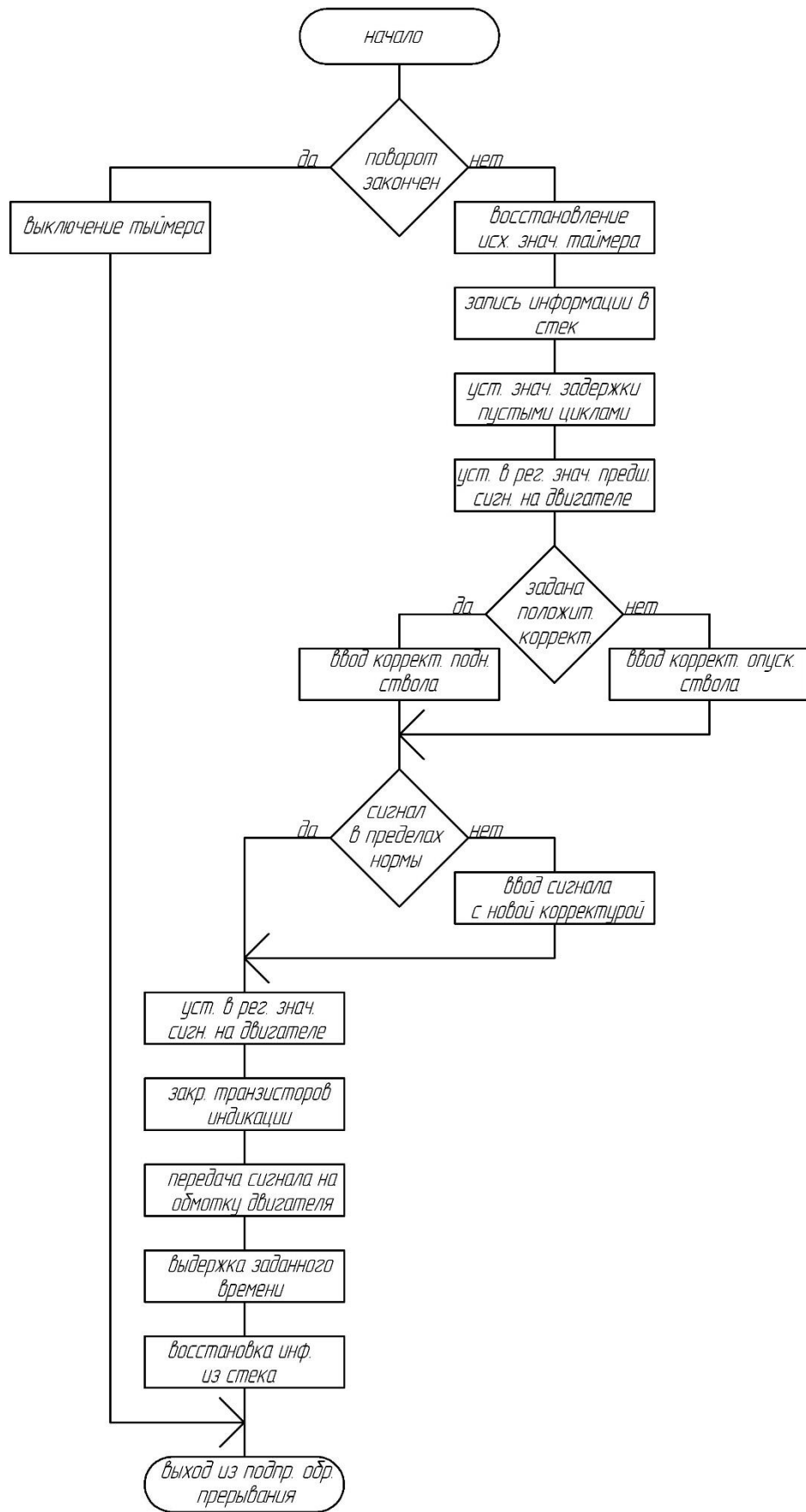


Рисунок 4.5 – Блок схема подпрограммы управления поворотом ствола

## 4.5 Вспомогательные подпрограммы

### - Подпрограмма задания параметров работы таймера

В ней задаются режим работы таймера, условия прерывания, отсчитываемое время

### - Подпрограмма создания базы данных

В ней формируется база данных кодов символов для индикации и набор углов, на которые необходимо поворачивать ствол, чтобы тело улетело на задаваемое расстояние(для каждого сантиметра дистанции свой угол)

### - Подпрограмма проверки нажата ли кнопка клавиатуры

Нужна для того, чтобы программа за одно нажатие не заполнила все три значения вводимых параметров(время нажатия много больше циклов выполнения подпрограмм). Если кнопка на клавиатуре нажата, программа не разрешает запись значения в два и более регистра, соответствующие знакам, вводимым с клавиатуры

### - Подпрограмма проверки отпущена ли кнопка клавиатуры

Нужна для того, чтобы продолжить опрос клавиатуры, и запись значений в следующий регистр.

## **Выводы**

В данном разделе приведена блок-схема алгоритма программы и разработана сама управляющая программа.



## 5 Математическое моделирование

Выберем систему координат следующим образом (рисунок 5.1). Ось  $OX$  направим горизонтально и совместим ее направление с направлением вектора начальной скорости  $V_0$ . Ось  $OY$  направим вертикально вверх. В этом случае траектория движения тела расположится в плоскости  $XOY$  («плоская» задача, координату  $z$  отбросим).

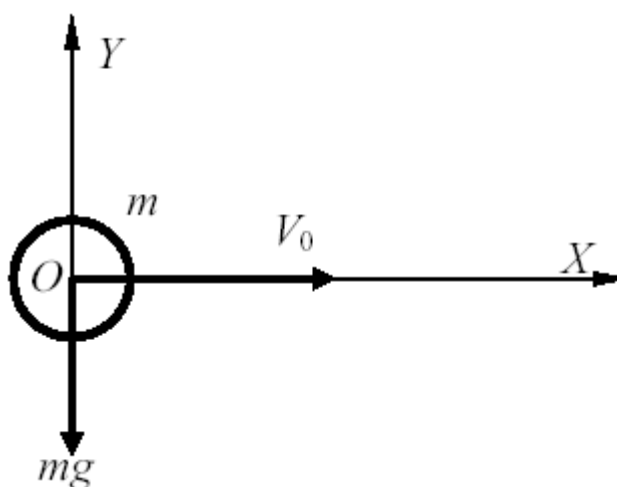


Рисунок 5.1 - Система координат

Предположим, что тело имеет форму шара (т.е. его ориентация не влияет на силу трения). Уравнения движения запишем, опираясь на второй закон Ньютона (сумма действующих на тело внешних сил должна уравниваться силой инерции), рассмотрев его в проекциях на оси в рассматриваемой системе координат.

$$\begin{cases} ma_x = -kV_x \\ ma_y = -mg - kV_y \end{cases} \quad (5.1)$$

Здесь  $a_x$  и  $a_y$  - проекции ускорения тела;  $V_x$ ,  $V_y$  - проекции скорости;  $m$  - масса тела,  $k$  - коэффициент вязкого трения тела о воздух.

Разделив обе части равенств на  $m$ , и выражая ускорения через скорости, получим:

$$\begin{cases} \dot{V}_x = -\frac{k}{m}V_x \\ \dot{V}_y = -g - \frac{k}{m}V_y \end{cases} \quad (5.2)$$

Дополнив уравнения (3) выражениями для расчета координат, получим окончательное уравнение состояния рассматриваемой системы в форме Коши:

$$\begin{cases} \dot{V}_x = -\frac{k}{m}V_x \\ \dot{V}_y = -g - \frac{k}{m}V_y \\ \dot{x} = V_x \\ \dot{y} = V_y \end{cases} \quad (5.3)$$

Для ее реализации средством визуального моделирования **Scicos** пакета **Scilab**, нам потребуются четыре интегратора, сумматор, два аналоговых перемножителя, построитель графика в координатах  $x, y$ .

Процесс визуального моделирования начнем с рассмотрения фрагментов модели, соответствующих каждому из четырех уравнений системы. В основе каждого фрагмента лежит решающий блок - интегратор (рисунок 5.2).

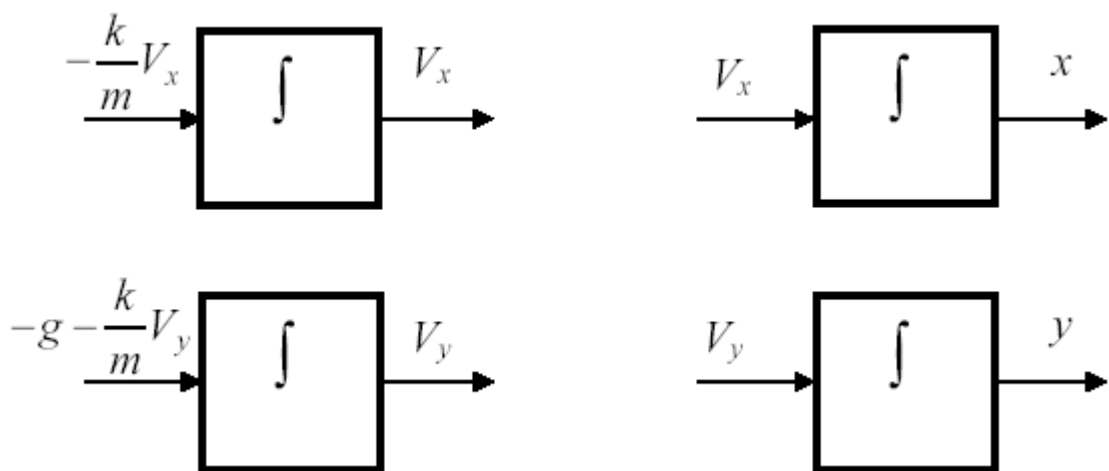


Рисунок 5.2 - Фрагменты модели, соответствующие уравнениям системы (4)

Дальнейший процесс составления модели заключается в соединении одноименных (выражающих одну и ту же физическую величину) выходов и входов и формировании алгебраических выражений с помощью умножителей и сумматоров. Например, формирование алгебраического выражения  $-\frac{k}{m}V_x$  иллюстрируется рисунком 5.5, а выражение  $-g - \frac{k}{m}V_y$  - рисунком 5.3

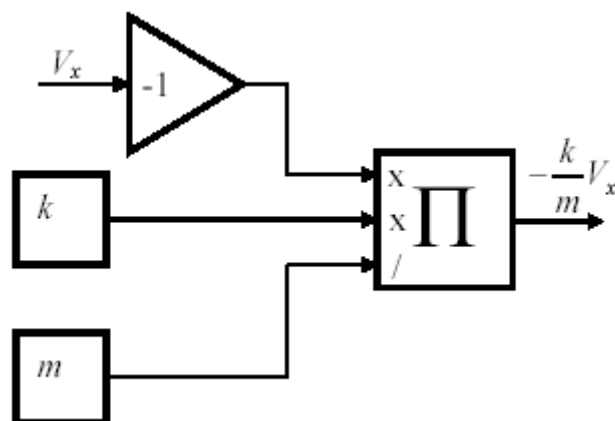


Рисунок 5.3 - Формирование алгебраического выражения

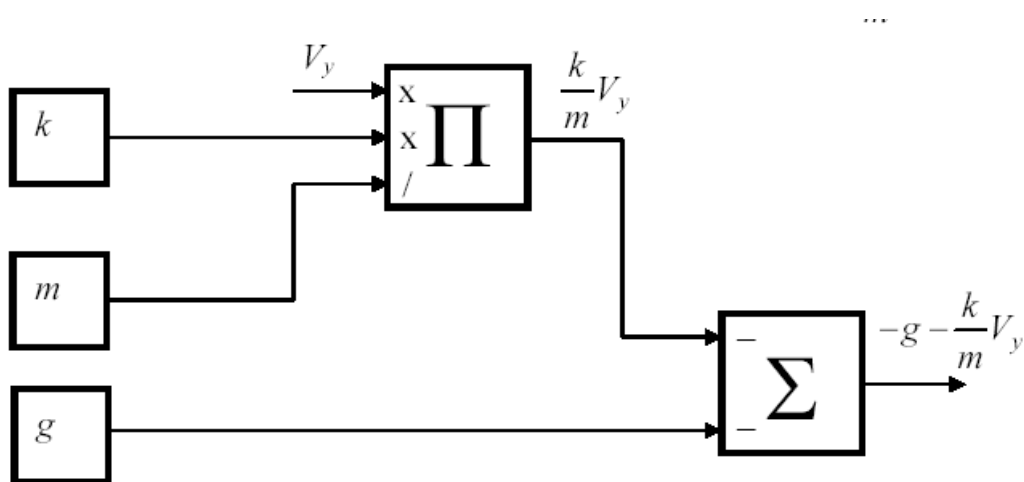


Рисунок 5.3 - Формирование алгебраического выражения

Модель системы, составленная в рабочем поле **Scicos**, приведена на рисунке 5.4.

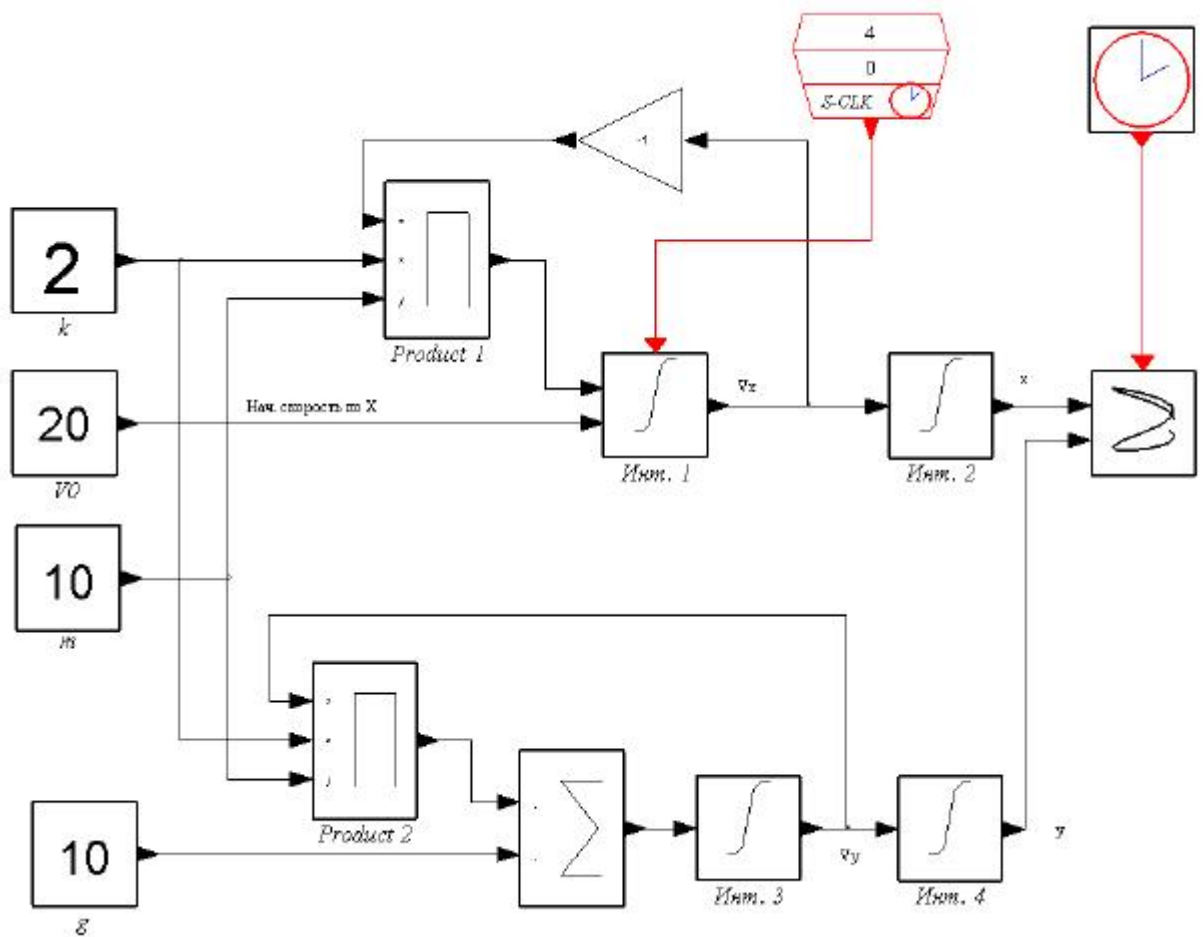


Рисунок 5.4 - модель системы моделирования полета тела, брошенного горизонтально

После этого запускаем симуляцию (меню **Simulate — Run**) и наблюдаем в раскрывшемся графическом окне траекторию полета тела (рисунок 5.5).

Нам же необходимо синтезировать модель полета тела, позволяющую вводить как начальную скорость  $V_0$ , так и угол  $\beta$  вектора начальной скорости по отношению к направлению горизонта (рисунок 5.6).

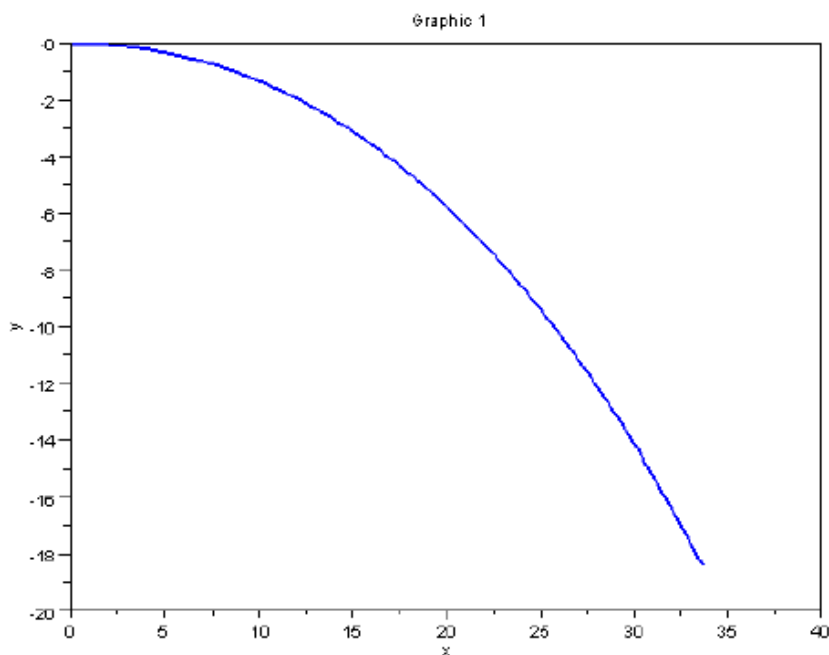


Рисунок 5.5 - Результат работы симулятора

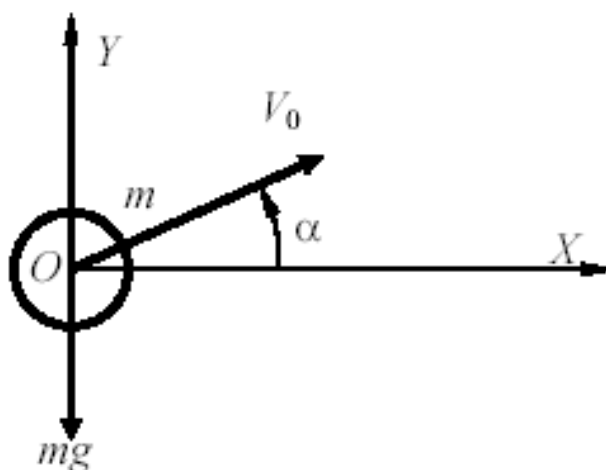


Рисунок 5.6 - Исследуемая система с вводом угла бросания тела

Для этого Вводимый угол на входе в модель нужно перемножить на синус угла (для вертикальной составляющей) и косинус угла (для горизонтальной составляющей) скорости. Для удобства ввода угла (в градусах, а не радианах), необходимо разделить вводимую величину угла на 57 (отношение градуса к радиану =  $1/57$ ). Модель системы с учетом угла бросания тела, составленная в рабочем поле **Scicos**, приведена на рисунке 5.7.

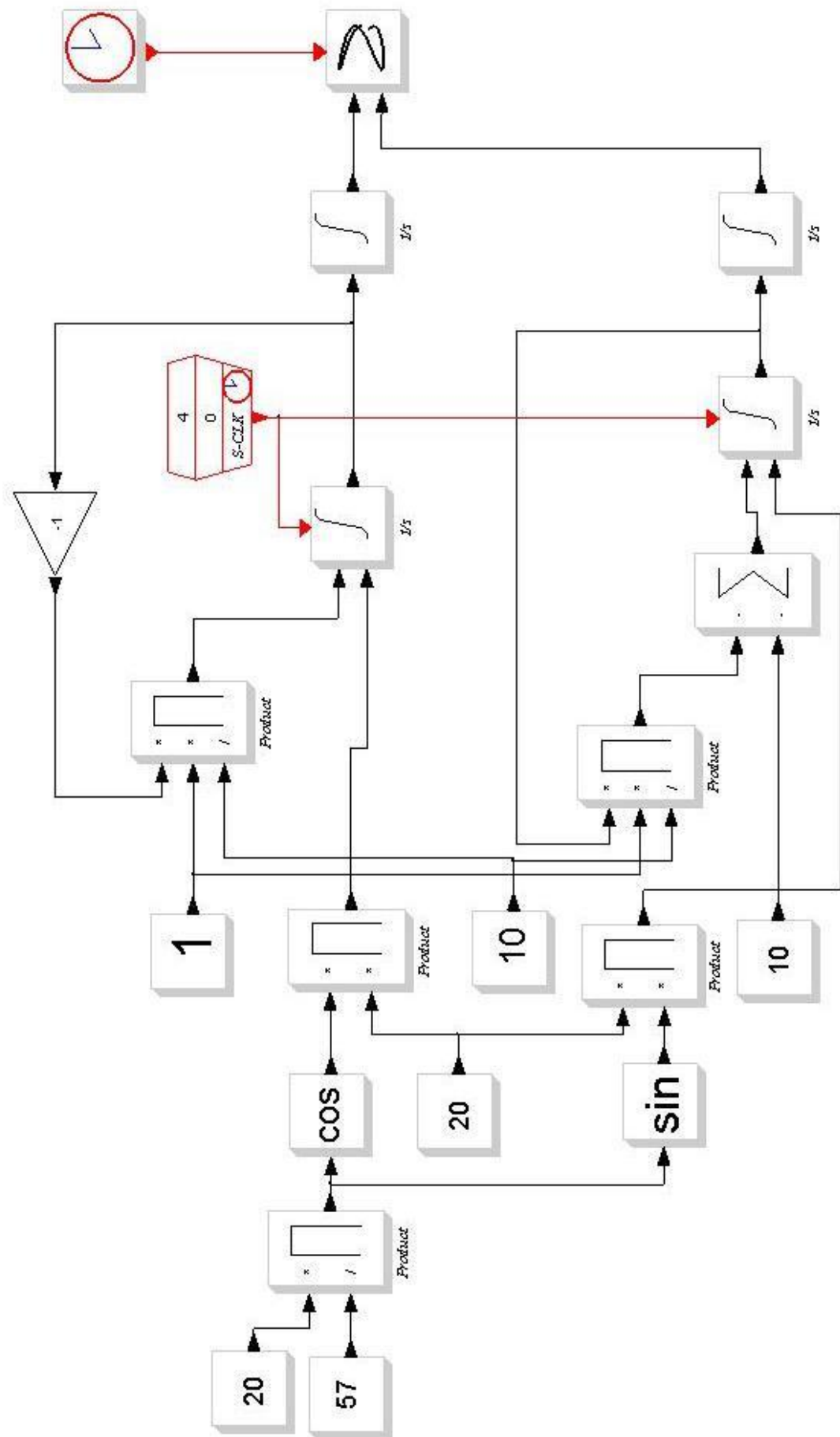


Рисунок 5.7 – Модель системы моделирования полета тела, брошенного под углом к горизонту

Результат моделирования полета тела по модели (рис. 5.7) приведен на рисунке 5.8.



Рисунок 5.8 – Результат математического моделирования полета тела.

Поскольку в нашем конкретном случае тело не имеет форму шара (точное учитывание сопротивления воздуха невозможно), а также нельзя точно определить массу тела (отсутствие точных приборов), мы в нашей математической модели пренебрегаем коэффициентом трения тела о воздух и массой тела. В результате получаем математическую модель, приведенную на рисунке 5.10, используемую для проведения исследований. При вводе параметра «высота точки вылета тела» использовались данные, полученные в ходе измерений на физической модели. Результаты измерений представлены в таблице 5.1.

Таблица 5.1– Результаты измерений высоты точки вылета тела

Угол $\alpha$ , градус	0	5	10	15	20	25	30	35	40	45	50	55
Высота, см	8,3	8,6	8,8	9,1	9,3	9,6	9,8	10	10,2	10,4	10,6	10,8

При введении в математическую модель значений, из таблицы 5.1 получили данные, которые объединили кривой и отобразили на рисунке 5.8.

Для упрощения использования математической модели приняли высоту вылета тела равной 9,6 см.

Результат исследований, проведенных с упрощенным использованием математической модели, отражен на рисунке 5.9.

Поскольку упрощенное использование математической модели не вносит существенных изменений в результаты исследования, приняли решение о возможности работы с математической моделью в упрощенном варианте.

Таблица 5.2 – Результаты опыта на математической модели

Угол, градус	0	5	10	15	20	25	30	35	40	45	50	55
Дальность, м	0,71	0,94	1,18	1,41	1,6	1,76	1,86	1,92	1,92	1,86	1,77	1,63

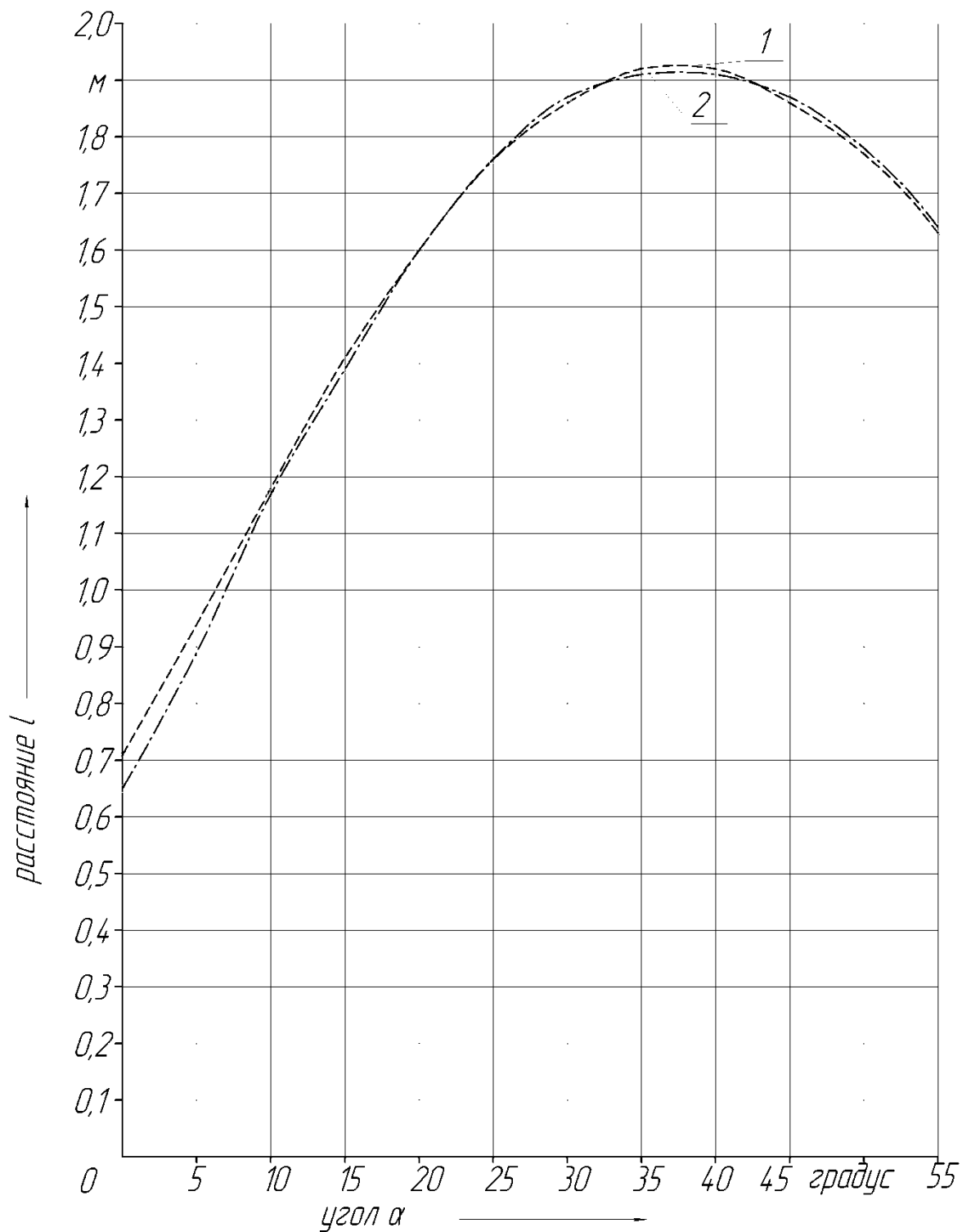
Сравнили результаты уточненной и упрощенной математических моделей. Сделали вывод о возможности использования упрощенной модели. Для наглядности соединили на одном рисунке 5.9 графики зависимостей дальности полета тела от угла поднятия ствола.

### **Выводы**

В данном разделе разработана математическая модель, проведено математическое моделирование и приведены и проанализированы его результаты.



ЗАВИСИМОСТЬ ДАЛЬНОСТИ ПОЛЕТА ТЕЛА ОТ УГЛА ПОВОРОТА СТВОЛА



1 – математическая модель,  
2 – упрощенная математическая модель

Рисунок 5. 9 – Результаты математического моделирования

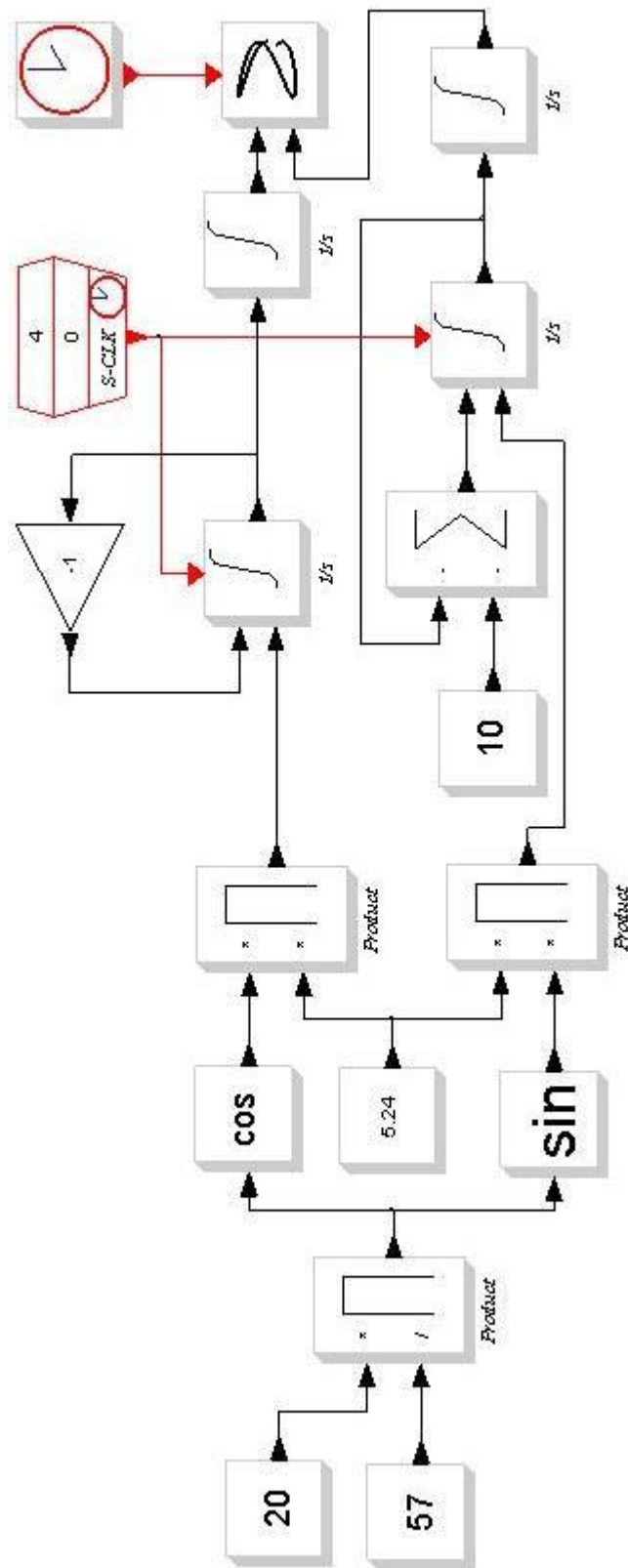


Рисунок 5.10 – Конечная модель системы моделирования полета тела

## Заключение

В данной ВКР была проведена разработка учебного стенда с микроконтроллерным управлением. Разработана система управления стендом, на основе микроконтроллера AT89S8252.

Достигнута поставленная цель работы: разработка стенда для изучения свободного полета тела.

Выполнены запланированные задачи:

1. Обзор состояния вопроса
2. Разработка электрической части стенда
3. Разработка программы для микроконтроллера
4. Математическое моделирование

Степень внедрения: теоретически рассчитаны параметры системы, разработаны схемы и выбраны элементы, написана программа для микроконтроллера.

В данной работе произведена разработка стенда для изучения свободного полета тела. Выполнен обзор состояния вопроса, расчет электрической схемы, разработка печатной платы и программы для микроконтроллера. Разработанный стенд позволяет изучать свободный полет тела. Стенд может быть применен в школе в качестве лабораторного стенда по физике или в ВУЗе в качестве стенда по программированию.

В ВКР было приведено большое количество иллюстраций, позволяющих глубоко вникнуть в суть изучаемой проблемы.

В результате проделанной работы можно сделать следующие выводы:

- 1) поставленные задачи успешно выполнены, цель работы достигнута;
- 2) предложенные решения поставленных проблем позволяют достигать заданных результатов.

## Список используемой литературы

1. AT89S8252 Datasheet (PDF) - ATMEL Corporation [Электронный ресурс]. URL: <http://www.alldatasheet.com/datasheet-pdf/pdf/175000/ATMEL/AT89S8252.html> (дата обращения: 27.05.2021)
2. AT89S8252 Primer [Электронный ресурс]. URL: <http://www.shrubbery.net/~heas/willem/PDF/ATMEL%20Flash%20Microcontroller/8051-Architecture/Application%20Notes/AT89S8252%20Primer.pdf> (дата обращения: 27.05.2021)
3. Electrical resistance and conductance [Электронный ресурс]. URL: [https://en.wikipedia.org/wiki/Electrical\\_resistance\\_and\\_conductance](https://en.wikipedia.org/wiki/Electrical_resistance_and_conductance) (дата обращения: 17.05.2021)
4. Electronic Components Datasheets [Электронный ресурс] URL: <http://www.datasheets.ru> (дата обращения: 17.02.2021)
5. LM78XX / LM78XXA 3-Terminal 1 A Positive Voltage Regulator [Электронный ресурс]. URL: <https://static.chipdip.ru/lib/164/DOC001164647.pdf> (дата обращения: 17.05.2021)
6. S. A. Prasad, B. S. Kariyappa, R. Nagary, S. K. Thakur, Microcontroller Based AC Power Controller, Wireless Sensor Network, Vol. 1, Issue 2, Jul 2009, pp. 76-81.
7. Голубцов М.С. Микроконтроллеры AVR: от простого к сложному. –М.:Солон-Пресс.2005.-304с.:ил.
8. Гребнев В.В. Микроконтроллеры семейства AVR.– М.:РадиоСофт, 2002. -176с.:ил.
9. Гусев В.Г., Гусев Ю.М. Электроника: Учеб. Пособие для приборостроит. Спец. Вузов. –М.: Высш. шк.1991.- 622 с.
10. Евстифеев А.В. Микроконтроллеры AVR семейства Classic фирмы ATMEL.–М.:Додэка, 2004.-288 с.:ил.
11. Забродин Ю.С. Промышленная электроника: Учебник для вузов. –М: Высш. школа, 1982. -496 с.: ил.

12. Конденсаторы: Справочник / Четверков И.И., Дьяконов М.Н., Присняков В.И. и др.: Под ред. Четверкова И.И., Дьяконова М.Н.-М.: Радио и связь. -1993. -392с.: ил.
13. Масленников М.Ю. Справочник разработчика и конструктора РЭА. –М.: Радио и связь, 1988.-300с.: ил.
14. Методические указания по оформлению выпускных квалификационных работ по программам бакалавриата, программам специалитета, программам магистратуры : [Электронный ресурс]. URL: [https://yadi.sk/d/Fs-9ts\\_VInrE3Q/BKP\\_\(Diplom\)](https://yadi.sk/d/Fs-9ts_VInrE3Q/BKP_(Diplom)) / Оформление ВКР (Дата обращения: 18.02.2021)
15. Положение о выпускной квалификационной работе: утв. решен. учен. совет. от 21.11.2019 решение №254 : [Электронный ресурс]. URL: [https://yadi.sk/d/Fs-9ts\\_VInrE3Q/BKP\\_\(Diplom\)](https://yadi.sk/d/Fs-9ts_VInrE3Q/BKP_(Diplom)) / Положение о ВКР (Дата обращения: 18.02.2021)
16. Полупроводниковые приборы: Справочник / Гитцевич А.Б.-М.: Радио и связь, 1988.-528с.:ил.
17. Порядок обеспечения самостоятельности выполнения письменных работ в ТГУ : [Электронный ресурс]. URL: [https://yadi.sk/d/Fs-9ts\\_VInrE3Q/BKP\\_\(Diplom\)](https://yadi.sk/d/Fs-9ts_VInrE3Q/BKP_(Diplom)) / Положение о Антиплагиате (Дата обращения: 18.02.2021)
18. Прядилов, А.В. Лабораторный практикум на базе лабораторного стенда «МКС-51» для курса основы микропроцессорной техники / А.В. Прядилов. - Тольятти: ТГУ, 2009. – 53с.
19. Прядилов, А.В. Основы микропроцессорной техники: учебно-методическое пособие / А.В.Прядилов; под ред. М.В. Позднова. – Тольятти: ТГУ, 2007. – 94с.
20. Резисторы: Справочник / В.В. Дубровский, Д.М. Иванов, Н.Я. Пратусевич и др. Под ред. И.И. Четверткова и В.М. Терехова. -2-е изд., перераб. И доп. – М.: Радио и связь, 1991.-528 с.:ил.

21. Сидоров И.Н. и др. Малогабаритные трансформаторы и дроссели: Справочник / И.Н. Сидоров, В.В. Мукосеев.-М.: Радио и связь, 1985. – 416с., ил.
22. Фрунзе А.В. Микроконтроллеры это же просто. –М.:Скимен, 2002.-336с.ил
23. Электроника и наноэлектроника, управление в технических системах, электроэнергетика и электротехника. Выполнение бакалаврской работы / сост. Позднов М.В., Прядилов А.В. - Тольятти: ТГУ, 2019. - 41 с.

## Приложение А. Текст программы

```
.include '89s8252.inc'; подключаем файл описания регистров
процессора
.ASEG ROM, code; Эта директива объявляет новый непереключаемый
сегмент в области памяти программ
time .equ (65535-50000); Устанавливаем время, через которое
сработает таймер (интервал изменения положения ствола на 1 градус)
TimeH .equ .HIGH(Time); Присваиваем старшему биту таймера
символическое имя
TimeL .equ .LOW(Time); Присваиваем младшему биту таймера
символическое имя
JMP L; Переход на начало программы
.ORG 0BH; Устанавливаем номер ячейки памяти, в которой
расположим вектор прерывания
jmp T0_obr; Переходим к программе обработки прерывания
.org 400h; Устанавливаем номер ячейки памяти, в которой
расположим базу данных
.db1 11000000b; Код цифры "0" на семисегментном индикаторе
.db1 11111001b; Код цифры "1" на семисегментном индикаторе
.db1 10100100b; Код цифры "2" на семисегментном индикаторе
.db1 10110000b; Код цифры "3" на семисегментном индикаторе
.db1 10011001b; Код цифры "4" на семисегментном индикаторе
.db1 10010010b; Код цифры "5" на семисегментном индикаторе
.db1 10000010b; Код цифры "6" на семисегментном индикаторе
.db1 11111000b; Код цифры "7" на семисегментном индикаторе
.db1 10000000b; Код цифры "8" на семисегментном индикаторе
.db1 10010000b; Код цифры "9" на семисегментном индикаторе
.db1 10011100b; Код символа "градус" на семисегментном индикаторе
.db1 10111111b; Код символа "черта" на семисегментном индикаторе
.db1 10000110b; Код символа "Е" на семисегментном индикаторе
.db1 10101111b; Код символа "г" на семисегментном индикаторе
.db1 0; угол, на который должно измениться положение ствола при
заданной дальности 70 см
.db1 0; -//- 71 см
.db1 0; -//- 72 см
.db1 0; -//- 73 см
.db1 1; -//- 74 см
```

## Продолжение Приложения А

.db1 1; -//- 75 см  
.db1 1; -//- 76 см  
.db1 1; -//- 77 см  
.db1 1; -//- 78 см  
.db1 1; -//- 79 см  
.db1 2; -//- 80 см  
.db1 2; -//- 81 см  
.db1 2; -//- 82 см  
.db1 2; -//- 83 см  
.db1 2; -//- 84 см  
.db1 2; -//- 85 см  
.db1 3; -//- 86 см  
.db1 3; -//- 87 см  
.db1 3; -//- 88 см  
.db1 3; -//- 89 см  
.db1 3; -//- 90 см  
.db1 3; -//- 91 см  
.db1 4; -//- 92 см  
.db1 4; -//- 93 см  
.db1 4; -//- 94 см  
.db1 4; -//- 95 см  
.db1 4; -//- 96 см  
.db1 4; -//- 97 см  
.db1 5; -//- 98 см  
.db1 5; -//- 99 см  
.db1 5; -//- 100 см  
.db1 5; -//- 101 см  
.db1 5; -//- 102 см  
.db1 6; -//- 103 см  
.db1 6; -//- 104 см  
.db1 6; -//- 105 см  
.db1 6; -//- 106 см  
.db1 7; -//- 107 см  
.db1 7; -//- 108 см  
.db1 7; -//- 109 см  
.db1 7; -//- 110 см  
.db1 8; -//- 111 см  
.db1 8; -//- 112 см  
.db1 8; -//- 113 см



## Продолжение Приложения А

.db1 8; -//- 114 см  
.db1 9; -//- 115 см  
.db1 9; -//- 116 см  
.db1 9; -//- 117 см  
.db1 9; -//- 118 см  
.db1 10; -//- 119 см  
.db1 10; -//- 120 см  
.db1 10; -//- 121 см  
.db1 10; -//- 122 см  
.db1 11; -//- 123 см  
.db1 11; -//- 124 см  
.db1 11; -//- 125 см  
.db1 11; -//- 126 см  
.db1 12; -//- 127 см  
.db1 12; -//- 128 см  
.db1 12; -//- 129 см  
.db1 12; -//- 130 см  
.db1 13; -//- 131 см  
.db1 13; -//- 132 см  
.db1 13; -//- 133 см  
.db1 13; -//- 134 см  
.db1 14; -//- 135 см  
.db1 14; -//- 136 см  
.db1 14; -//- 137 см  
.db1 14; -//- 138 см  
.db1 15; -//- 139 см  
.db1 15; -//- 140 см  
.db1 15; -//- 141 см  
.db1 15; -//- 142 см  
.db1 16; -//- 143 см  
.db1 16; -//- 144 см  
.db1 16; -//- 145 см  
.db1 16; -//- 146 см  
.db1 16; -//- 147 см  
.db1 17; -//- 148 см  
.db1 17; -//- 149 см  
.db1 17; -//- 150 см  
.db1 17; -//- 151 см  
.db1 17; -//- 152 см

## Продолжение Приложения А

.db1 18; -//- 153 см  
.db1 18; -//- 154 см  
.db1 18; -//- 155 см  
.db1 18; -//- 156 см  
.db1 18; -//- 157 см  
.db1 19; -//- 158 см  
.db1 19; -//- 159 см  
.db1 19; -//- 160 см  
.db1 19; -//- 161 см  
.db1 19; -//- 162 см  
.db1 20; -//- 163 см  
.db1 20; -//- 164 см  
.db1 20; -//- 165 см  
.db1 20; -//- 166 см  
.db1 21; -//- 167 см  
.db1 21; -//- 168 см  
.db1 21; -//- 169 см  
.db1 22; -//- 170 см  
.db1 22; -//- 171 см  
.db1 22; -//- 172 см  
.db1 23; -//- 173 см  
.db1 23; -//- 174 см  
.db1 23; -//- 175 см  
.db1 24; -//- 176 см  
.db1 24; -//- 177 см  
.db1 24; -//- 178 см  
.db1 25; -//- 179 см  
.db1 25; -//- 180 см  
.db1 25; -//- 181 см  
.db1 26; -//- 182 см  
.db1 26; -//- 183 см  
.db1 26; -//- 184 см  
.db1 27; -//- 185 см  
.db1 27; -//- 186 см  
.db1 27; -//- 187 см  
.db1 28; -//- 188 см  
.db1 28; -//- 189 см  
.db1 28; -//- 190 см  
.db1 29; -//- 191 см

## Продолжение Приложения А

.db1 29; -//- 192 см  
.db1 29; -//- 193 см  
.db1 30; -//- 194 см  
.db1 30; -//- 195 см  
.org 30h; Устанавливаем номер ячейки памяти, в которой расположим начало программы  
L: ; Начало программы  
mov R0, #0; устанавливаем начальное значение угла поднятия ствола  
mov R5, #00000001b; Устанавливаем для корректной работы подпрограммы обработки прерывания  
L0: ; Подпрограмма установки начальных значений  
mov TMOD, #00000001b; Устанавливаем таймер в режим 16 битного счетчика  
setb EA; Разрешаем все прерывания  
setb ET0; Разрешаем прерывание от нулевого таймера  
mov TH0, #TimeH; Записываем старший байт таймера  
mov TL0, #TimeL; Записываем младший байт таймера  
clr F0; Сбрасываем (обнуляем), чтобы после нажатия кнопки сброса подпрограмма записи в регистры данных о нажатиях кнопок клавиатуры работала правильно  
mov R2, #11; (регистр - младший разряд устанавливаемого с клавиатуры числа) устанавливаем номер кода символа "черта" в базе данных  
mov R3, #11; (регистр - средний разряд устанавливаемого с клавиатуры числа) устанавливаем номер кода символа "черта" в базе данных  
mov R4, #11; (регистр - старший разряд устанавливаемого с клавиатуры числа); устанавливаем номер кода символа "черта" в базе данных  
mov R6, #1; Устанавливаем, чтобы перейти на метку "огон" 1 раз за цикл (до нажатия кнопки сброса)  
mov P1, #11111111b; Устанавливаем для надежной индикации кнопок клавиатуры, трансляции шинного формирователя в прямом направлении  
clr TR0; Останавливаем таймер  
start: ; Подпрограмма опроса клавиатуры  
mov A, #11; Устанавливаем номер кода символа "черта" в базе данных  
CLR P2.4; Устанавливаем для опроса первой строки клавиатуры на предмет нажатой кнопки  
JB P1.0, M1; Если кнопка "1" не нажата, продолжаем опрос  
MOV A, #1; Устанавливаем номер кода цифры "1" в базе данных  
jmp M11; Переход на подпрограмму записи в регистры данных о нажатиях кнопок клавиатуры

## Продолжение Приложения А

M1:

JB P1.1, M2; Если кнопка "2" не нажата, продолжаем опрос  
MOV A, #2; Устанавливаем номер кода цифры "2" в базе данных  
jmp M11; Переход на подпрограмму записи в регистры данных о нажатиях кнопок клавиатуры

M2:

JB P1.2, M3; Если кнопка "3" не нажата, продолжаем опрос  
MOV A, #3; Устанавливаем номер кода цифры "3" в базе данных  
jmp M11; Переход на подпрограмму записи в регистры данных о нажатиях кнопок клавиатуры

M3:

setb P2.4; Устанавливаем для окончания опроса первой строки клавиатуры на предмет нажатой кнопки  
clr P2.5; Устанавливаем для опроса второй строки клавиатуры на предмет нажатой кнопки

JB P1.0, M4; Если кнопка "4" не нажата, продолжаем опрос

MOV A, #4; Устанавливаем номер кода цифры "4" в базе данных  
jmp M11; Переход на подпрограмму записи в регистры данных о нажатиях кнопок клавиатуры

M4:

JB P1.1, M5; Если кнопка "5" не нажата, продолжаем опрос  
MOV A, #5; Устанавливаем номер кода цифры "5" в базе данных  
jmp M11; Переход на подпрограмму записи в регистры данных о нажатиях кнопок клавиатуры

M5:

JB P1.2, M6; Если кнопка "6" не нажата, продолжаем опрос  
MOV A, #6; Устанавливаем номер кода цифры "6" в базе данных  
jmp M11; Переход на подпрограмму записи в регистры данных о нажатиях кнопок клавиатуры

M6:

setb P2.5; Устанавливаем для окончания опроса второй строки клавиатуры на предмет нажатой кнопки  
clr P2.6; Устанавливаем для опроса третьей строки клавиатуры на предмет нажатой кнопки

JB P1.0, M7; Если кнопка "7" не нажата, продолжаем опрос

MOV A, #7; Устанавливаем номер кода цифры "7" в базе данных  
jmp M11; Переход на подпрограмму записи в регистры данных о нажатиях кнопок клавиатуры

M7:

## Продолжение Приложения А

JB P1.1, M8; Если кнопка "8" не нажата, продолжаем опрос  
MOV A, #8; Устанавливаем номер кода цифры "8" в базе данных  
jmp M11; Переход на подпрограмму записи в регистры данных о нажатиях кнопок клавиатуры

M8:

JB P1.2, M9; Если кнопка "9" не нажата, продолжаем опрос  
MOV A, #9; Устанавливаем номер кода цифры "9" в базе данных  
jmp M11; Переход на подпрограмму записи в регистры данных о нажатиях кнопок клавиатуры

M9:

setb P2.6; Устанавливаем для окончания опроса третьей строки клавиатуры на предмет нажатой кнопки

clr P2.7; Устанавливаем для опроса четвертой строки клавиатуры на предмет нажатой кнопки

JB P1.0, M10; Если кнопка "градус" не нажата, продолжаем опрос  
MOV A, #10; Устанавливаем номер кода символа "градус" в базе данных

jmp M11; Переход на подпрограмму записи в регистры данных о нажатиях кнопок клавиатуры

M10:

JB P1.1, M11; Если кнопка "0" не нажата, переходим на проверку: установлено ли число в регистре-старшем разряде с клавиатуры

MOV A, #0; Устанавливаем номер кода цифры "0" в базе данных

M11: ; Подпрограмма записи в регистры данных о нажатиях кнопок клавиатуры

cjne R4, #11, opr3; Если установлено число в регистре-старшем разряде с клавиатуры, переход на метку проверки нажатия кнопки клавиатуры

MOV R4, A; Установка в регистр-старший разряд номера кода числа, нажатого с клавиатуры, в базе данных

provn: ; Подпрограмма проверки нажатия кнопки клавиатуры

jnb P1.0, vivod; Если кнопка в левом столбце клавиатуры нажата, переход на подпрограмму вывода чисел (символов) на семисегментные индикаторы

jnb P1.1, vivod; Если кнопка в среднем столбце клавиатуры нажата, переход на подпрограмму вывода чисел (символов) на семисегментные индикаторы

## Продолжение Приложения А

`jnb P1.2, vivod`; Если кнопка в правом столбце клавиатуры нажата, переход на подпрограмму вывода чисел (символов) на семисегментные индикаторы

`cjne R4, #11, u1`; Если установлено число в регистре-старшем разряде с клавиатуры, переходим на метку установки бита "Ф0"

`jmp vivod`; Переход на подпрограмму вывода чисел (символов) на семисегментные индикаторы

`u1: ;` Метка установки бита "Ф0"

`setb F0`; Устанавливаем (равным единице) чтобы программа не переходила на подпрограмму проверки нажатия кнопки клавиатуры

`jmp start`; Переход на подпрограмму опроса клавиатуры

`opr3: ;` Метка проверки нажатия кнопки клавиатуры

`jnb F0, prouv`; Если бит "Ф0" не равен единице, переход на подпрограмму проверки нажатости кнопки клавиатуры

`cjne R3, #11, opr2`; Если установлено число в регистре-среднем разряде с клавиатуры, переход на метку записи в регистр-младший разряд номера кода числа, нажатого с клавиатуры, в базе данных

`MOV R3, A`; Установка в регистр-средний разряд номера кода числа, нажатого с клавиатуры, в базе данных

`prouv: ;` Подпрограмма проверки отпускания кнопки клавиатуры

`jnb P1.0, vivod1`; Если кнопка в левом столбце клавиатуры нажата, переход на метку сброса бита "Ф0"

`jnb P1.1, vivod1`; Если кнопка в среднем столбце клавиатуры нажата, переход на метку сброса бита "Ф0"

`jnb P1.2, vivod1`; Если кнопка в правом столбце клавиатуры нажата, переход на метку сброса бита "Ф0"

`jmp vivod`; Переход на подпрограмму вывода чисел (символов) на семисегментные индикаторы

`vivod1: ;` Метка сброса бита "Ф0"

`clr F0`; Сбрасываем (обнуляем) чтобы программа переходила на подпрограмму проверки нажатия кнопки клавиатуры

`jmp vivod`; Переход на подпрограмму вывода чисел (символов) на семисегментные индикаторы

`opr2: ;` Метка записи в регистр-младший разряд номера кода числа, нажатого с клавиатуры, в базе данных

`MOV R2, A`; Установка в регистр-младший разряд номера кода числа, нажатого с клавиатуры, в базе данных

`vivod: ;` Подпрограмма вывода чисел (символов) на семисегментные индикаторы

## Продолжение Приложения А

`jnb P3.3, st`; Если нажата кнопка сброса, переход на промежуточную метку подпрограммы установки начальных значений

`mov P2, #11111111b`; Устанавливаем, чтобы закрыть транзисторы, используемые для индикации чисел (символов), чтобы триггеры, подключенные к порту "P0" и светодиодам не проводили сигнал

`MOV P0, #11111111B`; Устанавливаем, чтобы светодиоды на семисегментных индикаторах не горели

`CLR P2.7`; Открываем транзистор, отвечающий за индикацию цифр (символа) старшего разряда

`MOV A, R4`; Устанавливаем в аккумулятор из регистра-старшего разряда номер кода цифры (символа) в базе данных

`MOV DPTR, #400h`; Загрузка указателя базы данных

`MOVC A, @A+DPTR`; Пересылка в аккумулятор кода цифры (символа) из базы данных

`MOV P0, A`; Устанавливаем на порт "P0" кода цифры (символа) из базы данных

`SETB P2.7`; Закрываем транзистор, отвечающий за индикацию цифр (символа) старшего разряда

`MOV P0, #11111111B`; Устанавливаем, чтобы светодиоды на семисегментных индикаторах не горели

`CLR P2.6`; Открываем транзистор, отвечающий за индикацию цифр (символа) среднего разряда

`MOV A, R3`; Устанавливаем в аккумулятор из регистра-среднего разряда номер кода цифры (символа) в базе данных

`MOV DPTR, #400h`; Загрузка указателя базы данных

`MOVC A, @A+DPTR`; Пересылка в аккумулятор кода цифры (символа) из базы данных

`MOV P0, A`; Устанавливаем на порт "P0" кода цифры (символа) из базы данных

`SETB P2.6`; Закрываем транзистор, отвечающий за индикацию цифр (символа) среднего разряда

`MOV P0, #11111111B`; Устанавливаем, чтобы светодиоды на семисегментных индикаторах не горели

`CLR P2.5`; Открываем транзистор, отвечающий за индикацию цифр (символа) младшего разряда

`MOV A, R2`; Устанавливаем в аккумулятор из регистра-младшего разряда номер кода цифры (символа) в базе данных

`MOV DPTR, #400h`; Загрузка указателя базы данных

## Продолжение Приложения А

MOVC A, @A+DPTR; Пересылка в аккумулятор кода цифры (символа) из базы данных

mov P0, A; Установка на порт кода цифры (символа) из базы данных  
SETB P2.5; Закрываем транзистор, отвечающий за индикацию цифр (символа) младшего разряда

MOV P0, #1111111B; Устанавливаем, чтобы светодиоды на семисегментных индикаторах не горели

cjne R2, #11, Tir3; Если установлено число в регистре-младшем разряде с клавиатуры, переход на метку опроса кнопки "наводить" на предмет нажатия

jmp start; Переход на подпрограмму опроса клавиатуры

st: ; Промежуточная метка подпрограммы установки начальных значений

jmp L0; Переход на подпрограмму установки начальных значений

Tir3: ; Метка опроса кнопки "наводить" на предмет нажатия

jnb P3.4, ogon; Если кнопка "наводить" нажата, переход на подпрограмму записи в память значения угла поворота ствола

jmp vivod; Переход на подпрограмму вывода чисел (символов) на семисегментные индикаторы

ogon: ; Подпрограмма записи в память значения угла поворота ствола

cjne R6, #0, os; Чтобы до нажатия кнопки сброса программа записи в память значения угла поворота ствола проводила расчет и запись угла один раз

jmp vivod; Переход на подпрограмму вывода чисел (символов) на семисегментные индикаторы

os:

dec R6; Вычитаем из регистра единицу, чтобы отметить, что программа записи в память значения угла поворота ствола уже произвела расчет и запись угла

cjne R2, #10, rasstojanie; Если в регистре - младшем разряде не записан номер кода символа "градус", переход на метку расчета и записи угла поворота ствола по введенному расстоянию

ugol: ; Метка расчета и записи угла поворота ствола по введенному градусу

setb F0; Устанавливаем (равным единице) чтобы угол поворота ствола был положительным

mov A, R4; Устанавливаем в аккумулятор из регистра-старшего разряда цифру, для последующих вычислений (количество десятков)



## Продолжение Приложения А

mov B, #10; Устанавливаем, чтобы, перемножив с количеством десятков, получить число для дальнейших вычислений

MUL AB; Перемножаем для определения числа, которое вносят в конечный угол десятки

Add A, R3; Получаем конечный угол поворота ствола

Subb A, #56; Чтобы определить разрешенный интервал ввода угла

jnb CY, eror; Проверка, превышает ли введенный угол предельно допустимый, если да, переход на метку "error"

Add A, #56; Восстанавливаем число, стертое в ходе команды определения разрешенного интервала ввода угла

mov R1, A; Запись в память конечного угла поворота ствола, для последующего использования

Subb A, R0; Расчет корректуры угла поворота ствола

jnb CY, G; Если корректура положительная, переход на метку "далее"

mov A, R0; Выполняем пересчет корректуры

Subb A, R1; Получаем новую корректуру

clr F0; Сбрасываем (обнуляем) чтобы угол поворота ствола был отрицательным

G: ; Метка "далее"

Add A, #1; Вводим корректуру

XCH A, R1; Запись в память корректуры угла поворота ствола для непосредственного поворота

mov R0, A; Запись в память корректуры угла поворота ствола, чтобы после нажатия кнопки сброса, посчитать корректуру угла

setb TR0; Запуск таймера, чтобы обеспечить поворот ствола

jmp vivod; Переход на подпрограмму вывода чисел (символов) на семисегментные индикаторы

rasstojanie: ; Метка расчета и записи угла поворота ствола по введенному расстоянию

setb F0; Устанавливаем (равным единице) чтобы угол поворота ствола был положительным

cjne R4, #2, d; Проверяем, в пределах ли допустимых значений введен параметр

jmp eror; Переход на метку «error»

d:

mov A, R4; Устанавливаем в аккумулятор из регистра-старшего разряда цифру, для последующих вычислений (количество сотен)

mov B, #100; Устанавливаем, чтобы, перемножив с количеством сотен, получить число для дальнейших вычислений

## Продолжение Приложения А

MUL AB; Перемножаем для определения числа, которое вносят в конечный угол сотни

mov R7, B; Устанавливаем, чтобы проверить умещается ли число, которое вносят в конечный угол сотни, в аккумуляторе

cjne R7, #0, erog; Если не вмещается, переход на метку "erog"

mov R1, A; Запись в память числа, которое вносят в конечный угол сотни, для последующего использования

mov A, R3; Устанавливаем в аккумулятор из регистра-среднего разряда цифру, для последующих вычислений (количество десятков)

mov B, #10; Устанавливаем, чтобы, перемножив с количеством десятков, получить число для дальнейших вычислений

MUL AB; Перемножаем для определения числа, которое вносят в конечный угол десятки

Add A, R2; Получаем число, которое вносят в конечный угол единицы и сотни

Add A, R1; Получаем число, используемое для получения конечного угла поворота ствола

Subb A, #70; Чтобы определить разрешенный интервал ввода расстояния

jnb CY, dal; Проверка, меньше ли введенное расстояние предельно допустимого, если нет, переход на метку дальше

erog; ; Метка "erog"

mov R4, #12; Устанавливаем номер кода символа "Е" в базе данных

mov R3, #13; Устанавливаем номер кода символа "г" в базе данных

mov R2, #13; Устанавливаем номер кода символа "г" в базе данных

jmp vivod; Переход на подпрограмму вывода чисел (символов) на семисегментные индикаторы

dal; ; Метка "дальше"

Add A, #70; Восстанавливаем число, стертое в ходе команды определения разрешенного интервала ввода расстояния

Subb A, #196; Чтобы определить разрешенный интервал ввода расстояния

jnb CY, erog; Проверка больше ли введенное расстояние предельно допустимого, если да, переход на метку "erog"

Add A, #196; Восстанавливаем число, стертое в ходе команды определения разрешенного интервала ввода расстояния

subb A, #55; Расчет номера числа в базе данных, соответствующего заданной дальности

MOV DPTR, #400h; Загрузка указателя базы данных

## Продолжение Приложения А

MOVC A, @A+DPTR; Пересылка в аккумулятор числа из базы данных(конечный угол поворота ствола)

mov R1, A; Запись в память конечного угла поворота ствола, для последующего использования

Subb A, R0; Расчет корректуры угла поворота ствола

jnb CY, G1; Если корректура положительная, переход на метку "далее р"

mov A, R0; Выполняем пересчет корректуры

Subb A, R1; Получаем новую корректуру

clr F0; Сбрасываем (обнуляем) чтобы угол поворота ствола был отрицательным

G1: ; Метка "далее р"

Add A, #1; Вводим корректуру

XCH A, R1; Запись в память корректуры угла поворота ствола для непосредственного поворота

mov R0, A; Запись в память корректуры угла поворота ствола, чтобы после нажатия кнопки сброса, посчитать корректуру угла

setb TR0; Запуск таймера, чтобы обеспечить поворот ствола

jmp vivod; Переход на подпрограмму вывода чисел (символов) на семисегментные индикаторы

T0\_obr: ; Подпрограмма обработки прерывания

cjne R1, #0, o0; Если ствол еще не повернулся на заданный угол, переход на подпрограмму движения ствола

clr TR0; Если ствол уже повернулся на заданный угол, выключаем таймер

reti; Выход из подпрограммы обработки прерывания

o0: ; Подпрограмма движения ствола

dec R1; Отмечаем, что ствол повернулся на один градус

mov TH0, #TimeH; Записываем старший байт таймера (он обнулится, нужно выставить исходное значение)

mov TL0, #TimeL; Записываем младший байт таймера (он обнулится, нужно выставить исходное значение)

PUSH ACC; Записываем в стек значение аккумулятора (чтобы после выхода из подпрограммы обработки прерывания восстановить его и не повлиять на работу основной программы)

PUSH P2; Записываем в стек значение порта (чтобы после выхода из подпрограммы обработки прерывания восстановить его и не повлиять на работу основной программы)

## Продолжение Приложения А

PUSH PSW; Записываем в стек значение ячейки (чтобы после выхода из подпрограммы обработки прерывания восстановить его и продолжить работу основной программы с того места, где она была прервана)

PUSH P0; Записываем в стек значение порта (чтобы после выхода из подпрограммы обработки прерывания восстановить его и не повлиять на работу основной программы)

mov R6, #250; Устанавливаем, чтобы впоследствии, значение приняло участие в продлении времени подачи сигнала на обмотку двигателя для его корректной работы

mov R7, #25; Устанавливаем, чтобы впоследствии, значение приняло участие в продлении времени подачи сигнала на обмотку двигателя для его корректной работы

mov A, R5; Устанавливаем значение предшествующего сигнала на обмотке двигателя, чтобы скорректировать сигнал

jb F0, boev; Если бит установлен, переход на метку поднятия ствола

RL A; Ввод корректуры для опускания ствола

jnb 0E4H, y1; Если сигнал после ввода корректуры не вышел за установленные пределы, переход на метку "продолжить"

mov A, #00000001b; Ввод сигнала с корректурой (сигнал не выходит за установленные пределы)

jmp y1; Переход на метку "продолжить"

boev: ; Метка поднятия ствола

RR A; Ввод корректуры для поднятия ствола

jnb 0E7H, y1; Если сигнал после ввода корректуры не вышел за установленные пределы, переход на метку "продолжить"

mov A, #00001000b; Ввод сигнала с корректурой (сигнал не выходит за установленные пределы)

y1: ; Метка "продолжить"

mov R5, A; Запись в память сигнала на обмотке двигателя

x1: ; Метка продления времени

mov P2, #11111111b; Устанавливаем, чтобы закрыть транзисторы, используемые для индикации чисел (символов), чтобы триггеры, подключенные к порту "P0" и светодиодам не проводили сигнал

mov P0, A; Установка сигнала на обмотку двигателя

clr P3.5; Разрешаем трансляцию шинному формирователю

DJNZ R6, x1; Продлеваем время (пока не обнулится)

mov R6, #250; Восстанавливаем обнуленное значение

DJNZ R7, x1; Продлеваем время (пока не обнулится)

## Продолжение Приложения А

mov R6, #0; Устанавливаем, чтобы отметить, что расчет и запись угла произвелись

setb P3.5; Запрещаем трансляцию шинному формирователю

pop P0; Восстанавливаем из стека значение порта (чтобы после выхода из подпрограммы обработки прерывания не повлиять на работу основной программы)

pop PSW; Восстанавливаем из стека значение ячейки (чтобы после выхода из подпрограммы обработки прерывания продолжить работу основной программы с того места, где она была прервана)

pop P2; Восстанавливаем из стека значение порта (чтобы после выхода из подпрограммы обработки прерывания не повлиять на работу основной программы)

pop Acc; Восстанавливаем из стека значение аккумулятора (чтобы после выхода из подпрограммы обработки прерывания не повлиять на работу основной программы)

reti; Выход из подпрограммы обработки прерывания

.END; Эта команда указывает на окончание сегмента, объявленного директивой .ASEG

## Приложение Б Перечень элементов

		Поз. обозначение	Наименование	Кол.	Примечание
Перв. примен.			Конденсаторы		
		C1,C2,C4-C6,C8	K10-17Б - 40В - 1 мкФ	8	
		C3	K50-35 - 50В - 1 мкФ	1	
		C7,C9	SMD-1206 - 50В - 30мкФ	1	
Справ. №			Микросхемы		
		DA1	MAX 232	1	
		DD1	AT89S8252	1	
		DD2	74ACT373PCN20A	1	
			Индикаторы		
		HG1	BA56_12	1	
Подп. и дата			Устройства ввода		
		KL1	AK-304-N-BBW	1	
Инд. № дораб.			Резисторы		
		R1-R3	M/T 0,125 - 470 Ом ± 20%	3	
Взам. инв. №		R4	M/T 0,125 - 1 кОм ± 20%	1	
		R5	M/T 0,125 - 10 кОм ± 20%	1	
		R6-R7,R10-R20	M/T 0,125 - 2,2 кОм ± 20%	11	
		R11-R17,R21-R24	M/T 0,125 - 180 Ом ± 10%	8	
Подп. и дата					
Инд. № подл.	Изм.	Лист	№ докум.	Подп.	Дата
	Разраб.	Хамроев М.М.			
	Проб.	Певчев В.П.			
	Н.контр.				
	Утв.	Шедцов А.А.			
.33					
Стенд				Лист	Листов
Перечень элементов				1	2

Копировал

Формат А4

Продолжение Приложения А

		<i>Поз. обозначение</i>	<i>Наименование</i>	<i>Кол.</i>	<i>Примечание</i>
			<i>Диоды</i>		
		<i>VD1-VD3</i>	<i>L-132-XIT</i>	<i>3</i>	
		<i>VD4-VD7</i>	<i>1N4007</i>	<i>4</i>	
			<i>Транзисторы</i>		
		<i>VT1</i>	<i>KT3107</i>	<i>1</i>	
		<i>VT2-VT5</i>	<i>KT829Б</i>	<i>4</i>	
		<i>VT6-VT8</i>	<i>KT973А</i>	<i>3</i>	
			<i>Разъемы</i>		
		<i>X1</i>	<i>DRBFA</i>	<i>1</i>	
		<i>X2</i>	<i>K375K</i>	<i>1</i>	
			<i>Кварцевые резонаторы</i>		
		<i>ZQ1</i>	<i>HC-49S-12M</i>	<i>1</i>	
Подп. и дата Инв. № докум. Взам. инв. № Подп. и дата Инв. № подл.					
					<i>Лист</i>
	<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подп.</i>	<i>Дата</i>
					<i>2</i>

Копировал

Формат А4