**Аннотация**

Быков С.К. Система управления механизмом зажигания. – Челябинск, ЧЭнК, 2006, с.75, илл.6. Библиографической литературы – 8 наименований. 3 листа чертежей формата А1, 7 листов карт технологического процесса.

В данном дипломном проекте проводится разработка системы управления механизмом зажигания. В общей части рассматриваются принципы построения систем управления на микроконтроллерах.

В специальной части приведены разработка структурной и принципиальной схемы устройства, анализ семейств микроконтроллеров, разработка программного обеспечения для микроконтроллера и выполнен расчет надежности устройства.

В разделе организация производства представлены виды технической документации, необходимые при производстве радиоэлектронной аппаратуры и маршрутная карта технологического процесса на производство печатных плат.

В экономической части выполнен расчет себестоимости производства проектируемого устройства и проведен анализ потребительского спроса на рынке.

В разделе техники безопасности перечислены правила техники безопасности, соблюдение которых необходимо при проведении электромонтажных работ.

**Содержание**

Введение

1. Системы управления на микроконтроллерах

1.1 Применение микроконтроллеров в современной промышленности

1.2 Принципы построения систем управления на микроконтроллерах

2. Разработка системы управления механизмом зажигания

2.1 Постановка задачи

2.2 Разработка структурной схемы

2.3 Разработка принципиальной схемы

2.4 Выбор элементов принципиальной схемы

2.5 Разработка программного обеспечения

2.6 Расчет надежности

3. Организация производства

3.1 Виды конструкторско технологической документации при производстве электронных устройств

3.2 Маршрутная карта технологического процесса при изготовлении печатной платы

4. Экономическая часть

4.1 Расчет себестоимости изготовления устройства

5. Мероприятия по технике безопасности и противопожарной технике

5.1 Меры безопасности при производстве электромонтажных работ

Заключение

Список литературы

Приложение А Текст программы микроконтроллера

Микропроцессорные технологии уже давно вышли за рамки персональных компьютеров и суперЭВМ. Во всем мире широкое распространение получили микроконтроллеры, как в автоматизированных системах управления, так и в бытовых электроприборах, так как они отличаются надежностью, высокой степенью интеграции и небольшой стоимостью.

В данном дипломном проекте я планирую рассмотреть возможность применения микроконтроллеров в системах зажигания двигателей внутреннего сгорания. А так же описать этапы разработки этой системы зажигания и привести необходимую документацию для организации производства устройства, либо его изготовления в условиях малого предприятия.

**1. Системы управления на микроконтроллерах**

**1.1 Применение микроконтроллеров в современной промышленности**

Микроконтроллеры являются наиболее массовыми представителями микропроцессорной электроники. Интегрируя в одном корпусе микросхемы высокопроизводительный процессор, оперативную и постоянную память, а также набор периферийных устройств, микроконтроллеры позволяют с минимальными затратами реализовать широкую номенклатуру систем управления различными объектами и процессами.

Структурная организация, набор команд и аппаратурнопрограммные средства ввода/вывода информации микроконтроллеров лучше всего приспособлены для решения задач управления и регулирования в приборах, устройствах и системах автоматики, а не для решения задач обработки данных. Микроконтроллеры не являются классическими электронно – вычислительными машинами, так как физическая и логическая разделённость памяти программ и памяти данных исключает возможность модификации или замены (перезагрузки) прикладных программ микроконтроллеров во время работы, что сильно затрудняет их использование в качестве универсальных средств обработки данных.

Поэтому микроконтроллеры находят широкое применение в промышленной автоматике, контрольно измерительной технике, аппаратуре связи, бытовой технике и многих других областях человеческой деятельности.

**1.2 Принципы построения систем управления на микроконтроллерах**

В устройствах управления объектами (контроллерах) на основе микроконтроллеров аппаратурные средства и программное обеспечение существуют в форме неделимого аппаратурно программного комплекса. При проектировании контроллеров приходится решать одну из самых сложных задач разработки, а именно задачу оптимального распределения функций контроллера между аппаратурными средствами и программным обеспечением. Решение этой задачи осложняется тем, что взаимосвязь и взаимовлияние аппаратурных средств и программного обеспечения в микропроцессорной технике претерпевают динамичные изменения. Если в начале развития микропроцессорной техники определяющим было правило, в соответствии с которым аппаратурные средства обеспечивают производительность, а программное обеспечение дешевизну изделия, то в настоящее время это правило нуждается в серьезной корректировке. Так как микроконтроллер представляет собой стандартный массовый (относительно недорогой) логический блок, конкретное назначение которого определяет пользователь с помощью программного обеспечения, то с ростом степени интеграции и, следовательно, функциональнологических возможностей микроконтроллера резко понижается стоимость изделия в пересчете на выполняемую функцию, что в конечном итоге и обеспечивает достижение высоких техникоэкономических показателей изделий на микроконтроллере. При этом затраты на разработку программного обеспечения изделия в 210 раз превышают (за время жизни изделия) затраты на приобретение и изготовление аппаратурных средств.

В настоящее время наибольшее распространение получил методологический прием, при котором весь цикл разработки контроллеров рассматривается как последовательность трех фаз проектирования:

1. анализа задачи и выбора аппаратурных средств контроллера;

2. разработки прикладного программного обеспечения;

3. комплексирования аппаратурных средств и программного обеспечения в прототипе контроллера и его отладки.

Фаза разработки программного обеспечения, т.е. фаза получения прикладных программ, в свою очередь, разбивается на два существенно различных этапа:

1. "от постановки задачи к исходной программе";

2. "от исходной программы к объектному модулю".

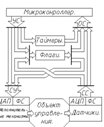
Этап разработки "от исходной программы к объектному модулю" имеет целью получение машинных кодов прикладных программ, работающих в микроконтроллере. Этот этап разработки прикладного программного обеспечения легко поддается формализации и поддержан всей мощью системного программного обеспечения микроконтроллера, направленного на автоматизацию процесса получения прикладных программ. В состав средств системного программного обеспечения входят трансляторы с различных алгоритмических языков высокого уровня, ассемблеры, редакторы текстов, программыотладчики, программы документаторы и т.д. Наличие всех этих системных средств придает инженерной работе на этом этапе проектирования контроллеров характер ремесла, а не инженерного творчества. Так как в конечном изделии (контроллере) имеются только "голый" микроконтроллер и средства его сопряжения с объектом, то выполнять отладку разрабатываемого прикладного программного обеспечения на нем невозможно (изза отсутствия средств ввода, вывода, ОЗУ большой емкости и операционной системы), и, следовательно, разработчик вынужден обращаться к средствам вычислительной техники для выполнения всех формализуемых стадий разработки: трансляции, редактирования, отладки, загрузки объектных кодов в программируемую постоянную память микроконтроллера.

Совсем по другому выглядит инженерный труд на этапе разработки программного обеспечения "от постановки задачи к исходной программе", так как он практически не поддается формализации и, следовательно, не может быть автоматизирован.

Проектная работа здесь носит творческий характер, изобилует решениями, имеющими "волевую" или "вкусовую" окраску, и решениями, продиктованными конъюнктурными соображениями. В силу перечисленных обстоятельств именно на этапе проектирования "от постановки задачи к исходной программе" разработчик сталкивается с наибольшим количеством трудностей.

Качество получаемого прикладного программного обеспечения контроллера всецело зависит от уровня проектных решений, принятых на этапе разработки "от постановки задачи к исходной программе". Уровень проектных решений в свою очередь изза отсутствия теории проектирования программируемых контроллеров определяется только опытом, квалификацией и интуицией разработчика. Однако накопленный опыт убеждает в том, что систематический подход к процессу разработки прикладных программ для контроллеров обеспечивает достижение хороших результатов даже начинающими разработчиками.

Типовая структура микропроцессорной системы управления показана на рис. 1.1 и состоит из объекта управления, микроконтроллера и аппаратуры их взаимной связи.



#### Рисунок 1.1 Структура цифровой системы управления на основе МК

Микроконтроллер путем периодического опроса осведомительных слов (ОС) генерирует в соответствии с алгоритмом управления последовательности управляющих слов (УС). Осведомительные слова это сигналы состояния объекта (СС), сформированные датчиками объекта управления, и флаги. Выходные сигналы датчиков вследствие их различной физической природы могут потребовать промежуточного преобразования на аналогоцифровых преобразователях (АЦП) или на схемах формирователей сигналов (ФС), которые чаще всего выполняют функции гальванической развязки и формирования уровней двоичных сигналов стандарта ТТЛ.

Микроконтроллер с требуемой периодичностью обновляет управляющие слова на своих выходных портах. Некоторая часть управляющего слова интерпретируется как совокупность прямых двоичных сигналов управления (СУ), которые через схемы формирователей сигналов (усилители мощности, реле, оптроны и т.п.) поступают на исполнительные механизмы (ИМ) и устройства индикации. Другая часть управляющего слова представляет собой упакованные двоичные коды, которые через цифро аналоговые преобразователи (ЦАП) воздействуют на исполнительные механизмы аналогового типа. Если объект управлении использует цифровые датчики и цифровые исполнительные механизмы, то наличие ЦАП и АЦП в системе необязательно.

В состав аппаратуры связи, которая как правило, строится на интегральных схемах серии ТТЛ, входит регистр флагов, на котором фиксируется некоторое множество специфицируемых признаков как объекта управления, так и процесса работы контроллера. Этот регистр флагов используется в качестве аппаратурного средства реализации механизма взаимной синхронизации относительно медленных и вероятностных процессов в объекте управления и быстрых процессов в контроллере. Регистр флагов доступен как контроллеру, так и датчикам. Вследствие этого он является удобным местом фиксации сигналов «готов»/«ожидание» при передачах с квитированием или сигналов «запрос прерывания»/«подтверждение» при взаимодействии контроллера и объекта в режиме прерывания. Если МКсистема имеет многоуровневую систему прерываний, то регистр флагов содержит схему упорядочивания приоритетов.

Для аппаратурной реализации временных задержек, формирования сигналов требуемой частоты и скважности в состав аппаратуры связи включают программируемые интервальные таймеры в том случае, если их нет в составе микроконтроллера или их число недостаточно.

Законы функционирования микропроцессорной системы управления со структурой, показанной на рис. 1 всецело определяются прикладной программой, размещаемой в резидентной памяти программ микроконтроллера. Иными словами, специализация контроллера типовой структуры на решение задачи управления конкретным объектом осуществляется путем разработки прикладных программ микроконтроллера и аппаратуры связи микроконтроллера с датчиками и исполнительными механизмами объекта.

**2. Разработка системы управления механизмом зажигания**

**2.1 Постановка задачи**

Одной из проблем двигателя внутреннего сгорания является запаздывание момента зажигания при увеличении частоты вращения коленвала, т.к. скорость преодоления поршнем верхней мертвой точки возрастает вместе с оборотами двигателя, а время сгорания топлива остается неизменным. Это приводит к значительной потере мощности двигателя, повышенному расходу топлива и существенному ограничению максимальных оборотов двигателя.

Для нормализации работы двигателя необходимо применение механизма, изменяющего момент зажигания в зависимости от оборотов двигателя, т.е. опережение момента при увеличении оборотов. В основном это достигается применением механических устройств рис. 2.1, принцип действия которых основан на изменении положения грузиков под воздействием центробежной силы. Недостатками таких механизмов являются нестабильная работа (особенно на низких оборотах) и невозможность достижения нужного угла опережения зажигания на высоких оборотах.



Рисунок 2.1 Механическая система опережения зажигания.

Избавиться от этих недостатков позволяет применение электронного механизма опережения зажигания. За основу этого механизма берется датчик, считывающий обороты распредвала и подающий сигнальные импульсы на устройство управления моментом зажигания.

К устройству управления предъявляются следующие требования:

1. прием сигнала от датчика;

2. преобразование сигнала в зависимости от оборотов двигателя (распредвала);

3. возможность изменять значение угла опережения зажигания.

4. сохранение работоспособности при воздействии высоких температур.

5. устойчивость к воздействию вибраций.

Оптимальным решением в данном случае является построение устройства управления моментом зажигания на микроконтроллере, так как преобразование сигналов датчика обеспечивается программно, что дает возможность регулировки. Термостойкость достигается применением микроконтроллера соответствующего уровня (с индексом). Устойчивость к вибрациям обеспечивается высокой степенью интеграции и малой массой радиоэлементов.

**2.2 Разработка структурной схемы**

Для определения частоты вращения двигателя, как уже говорилось, необходима установка специального датчика, на основе которого будет строиться вся схема устройства рис. 2.2. Следующий элемент механизма – электронная система смещения импульса, основанная на микроконтроллере. Микроконтроллер принимает сигнал от датчика, обрабатывает его с помощью программы, записанной в его ПЗУ, и передает уже обработанный сигнал на выход. Так как микроконтроллер не в состоянии работать с высокими токами, в систему вносится ключевая схема.

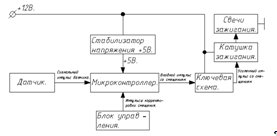


Рисунок 2.2 Структурная схема устройства.

Так же необходимым шагом является включение в схему устройства стабилизатора напряжения. Он нужен для преобразования напряжения бортовой сети транспортного средства +12В в напряжение, приемлемое для питания микроконтроллера, и гашения его скачков.

Блок управления включается для регулировки и точной настройки программы микроконтроллера под конкретный двигатель, корректировки момента искрообразования и угла опережения зажигания.

Катушка зажигания предназначена для преобразования 12 вольтового входного импульса со смещением в усиленный импульс со смещением амплитудой до 15000В. Такое высокое напряжение обеспечивает образование стабильного дугового разряда на контактах свечи зажигания.

**2.3 Разработка принципиальной схемы**

Датчик Холла это индукционный датчик, действующий на основе эффекта Холла. Он применяется в бесконтактных системах зажигания автомобильного типа. Построение системы на основе этого датчика является оптимальным решением, так как он адаптирован для работы с двигателями внутреннего сгорания. Для работы датчика Холла на распредвал устанавливается стальной экран (диск) рис. 2.3, имеющий два выреза, по 120 градусов каждый. Датчик устанавливается так, чтобы диск вращался в его проеме.

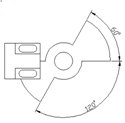


Рисунок 2.3 Экран датчика Холла и расположение датчика относительно экрана.

Датчик имеет три вывода: питание +12В. (красный), земля (черный) и сигнальный провод (зеленый). При входе шторки в проем датчика, он вырабатывает прямоугольный импульс с амплитудой от 9 до 12В (рис. 2.4). При выходе шторки, амплитуда резко падает до минимального значения – не более 0,4В. Сигналом для искрообразования служит окончание импульса.

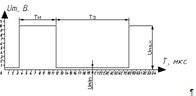


Рисунок 2.4 – Временная диаграмма датчика Холла.

Параметры вырабатываемых датчиком импульсов:

# Ти = Тз / 2 (2.1)

где Ти – длительность импульса,

Тз – длительность задержки импульса.

U max = 9 – 12В. (2.2)

где U max – максимальная амплитуда импульса.

U min = 0 – 0,4В. (2.3)

где U min минимальная амплитуда импульса.

В автомобильных системах зажигания с датчиком Холла работает коммутатор, выполняющий роль ключевой схемы. Плюсами применения коммутатора являются: наличие встроенной схемы питания датчика Холла, подача выходного импульса только при изменении входного, формирование выходного импульса, позволяющего максимально реализовать потенциал катушки зажигания.

Необходимыми частями устройства являются цепи, согласующие микроконтроллер с остальными элементами схемы.

Принципиальная схема системы управления механизмом зажигания приведена в чертеже ДП.230101.802.287.Э3.

Питание микроконтроллера обеспечивает цепь, состоящая из элементов VD1 (снижение уровня пульсаций), C2 и C3 (отфильтровывание несглаженных пульсаций), микросхемой DA1 КР142ЕН5А (преобразование 12В входного напряжения в 5В выходного).

RCцепь, состоящая из элементов R1, R3 и C1 снижает напряжение выходного сигнала датчика Холла (9 – 12В) до приемлемых микроконтроллером 5В.

К выводам микроконтроллера OSC1 и OSC2 подключается резонатор ZQ1, предназначенный для определения программой микроконтроллера временных интервалов сигналов датчика Холла.

Транзистор VT1 работает в ключевом режиме. Он предназначен для усиления выходного импульса микроконтроллера до уровня, приемлемого коммутатором. Оптимальным вариантом является применение высокочастотного маломощного транзистора, например КТ3117А. Резистор R4 предназначены для формирования тока базы транзистора VT1.

Ключи S1 – S5 нужны для подачи сигналов на входы микроконтроллера RB4 – RB7. Данные сигналы предназначены для корректировки переменных программы микроконтроллера.

**2.4 Выбор элементов принципиальной схемы**

Основным элементом устройства управления моментом зажигания является микроконтроллер. Применение микроконтроллера повышает уровень интеграции и надежность устройства. Так же это дает возможность изменять характеристики момента искрообразования.

Сейчас на рынке предлагается множество микроконтроллеров различных производителей. Самыми распространенными семействами микроконтроллеров являются AVR фирмы “Atmel” и PIC фирмы “MicroChip”.

Сравнительный анализ AVR и PIC микроконтроллеров.

AVR – микроконтроллер фирмы “Atmel” – это 8 – разрядные микроконтроллеры с базовой RISC архитектурой. Они привлекают внимание разработчиков наилучшим соотношением быстродействия и энергопотребления. В продаже имеется более 15 типов микроконтроллеров. AVR – контроллеры имеют 89 – 133 инструкции, большинство из которых выполняется за один период тактовой частоты. AVR – контроллеры имеют 32 8 – битных регистра общего назначения. Максимальная частота микроконтроллеров достигает 16 МГц, в основном же колеблется от 8 до 10 МГц. Большое количество инструкций сильно усложняет обучение программированию и само программирование микроконтроллеров, что является существенным недостатком AVR – контроллеров при взгляде со стороны пользователя.

Микроконтроллеры PIC это простые и высокопроизводительные микроконтроллеры с базовой RISC – архитектурой. Несомненное достоинство семейства – минимальное число команд – от 33, большинство из которых выполняется за 4 такта работы процессора. Тактовая частота микроконтроллеров колеблется от 10 до 40 МГц. Общие особенности PIC – контроллеров: внутренний сброс по включению питания, наличие сторожевого таймера, защита памяти программ от несанкционированного считывания, мощные линии ввода – вывода (до 25 мА), возможность внутрисхемного программирования.

Большее количество тактов, затрачиваемых на выполнение одной команды, в достатке компенсируется большими тактовыми частотами PIC – контроллеров. Несомненными достоинствами этих микроконтроллеров являются небольшое количество команд и мощные линии ввода – вывода. Поэтому выбор останавливается на PIC – микроконтроллерах.

Выбор микроконтроллера из семейства PIC контроллеров.

Для сравнения берется три основных семейства PIC – контроллеров: PIC12Cxxx, PIC16F84x и PIC18Fxx.

Микроконтроллеры PIC12Cxxx имеют следующие технические характеристики:

1. Напряжение питания 2,5В.
2. 6(5) линий ввода – вывода.
3. Наличие АЦП в кристаллах PIC12C6xx.
4. Повышенная длина команд (до 14 бит).
5. Тактовая частота – до 10 МГц.
6. Низкая стоимость.

Микроконтроллеры PIC16F84x имеют следующие технические характеристики:

1. Наличие Flash – памяти программ.
2. Отсутствие АЦП.
3. Тактовая частота – до 20 МГц.
4. 13 линий ввода – вывода.
5. Напряжение питания – от 4 до 6В.
6. Небольшая стоимость.

Микроконтроллеры PIC18Fxxx имеют следующие технические характеристики:

1. Расширенное ядро RISC.
2. Наличие Flash – памяти программ.
3. Тактовая частота – до 40 МГц.
4. До 34 линий ввода – вывода.
5. Наличие АЦП либо ЦАП.
6. Высокая стоимость.

Исходя из данного сравнения, оптимальным семейством микроконтроллеров является семейство PIC16F84x, так как несомненными плюсами являются: наличие Flash – памяти, отсутствие ЦАП и АЦП, большое количество линий ввода – вывода, небольшая стоимость. Из всего семейства, оптимальным для бесконтактной системы зажигания является микроконтроллер PIC16F844I/P (рис. 2.5). Здесь PIC16F84 марка микроконтроллера, 4 тактовая частота микроконтроллера, I индустриальный диапазон рабочих температур (от –40 до +85 градусов Цельсия), P корпус PDIP для монтажа в отверстия.

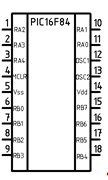


Рисунок 2.6 Условное обозначение микроконтроллера.

Таблица 1 назначение выводов микроконтроллера PIC16F844I/P.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Обозначение | № | Тип (I/O/P) | Тип буфера | Назначение |
| OSC1/CLKIN | 16 | I | ST/CMOS | Вход для подключения кварцевого резонатора, либо RC – цепи, либо вход для внешнего тактового генератора. |
| OSC2/CLKOUT | 15 | O |  | Выход для подключения кварцевого резонатора в режиме в режиме работы с кварцем, в режиме RC – генератора на выходе присутствуют импульсы с частотой 1/4 от OSC1 |
| MCLR | 4 | I/P | ST | Сброс по низкому уровню но входе. При программировании кристалла – вход напряжения программирования. |
| RA0  RA1  RA2  RA3  RA4/T0CKI | 17  18  1  2  3 | I/O  I/O  I/O  I/O  I/O | TTL  TTL  TTL  TTL  ST | Выводы двунаправленного порта А  RA4/T0CKI может быть настроен как вход импульсов для таймера – счетчика TMR0. Выход с открытым стоком. |
| RB0/INT  RB1  RB2  RB3  RB4  RB5  RB6  RB7 | 6  7  8  9  10  11  12  13 | I/O  I/O  I/O  I/O  I/O  I/O  I/O  I/O | TTL/ST  TTL  TTL  TTL  TTL  TTL  TTL/ST  TTL/ST | Выводы двунаправленного порта В. К выводам порта могут быть программно подключены внутренние подтягивающие резисторы Vdd.  RB0/INT может быть также программно настроен как вход внешнего прерывания  RB4 – RB7 могут быть также программно настроены как входы прерывания по изменению уровня на на любом из этих входов. Направление изменения задается программно.  При программировании кристалла RB6 используется как тактовый, RB7 как вход/выход данных. |
| Vss | 5 | P |  | Общий провод. |
| Vdd | 14 | P |  | Положительное напряжение питания |

Обозначения. I – вход, O – выход, I/O вход/выход, P – вывод питания, TTL – стандартный TTL вход, ST – вход с триггером Шмидта.

Таблица 2. Система команд микроконтроллера PIC16F844I/P.

|  |  |
| --- | --- |
| Команда. | Описание команды. |
| ADDWF f, d | Сложение W и f |
| ANDWF f, d | Выполняется побитное “И” содержимого регистров W и f.  Если d=0, результат сохраняется в регистре W.  Если d=1, результат сохраняется в регистре f. |
| CLRF f | Очистить содержимое регистра f и установить флаг Z. |
| CLRW | Очистить содержимое регистра W и установить флаг Z. |
| COMF f, d | Инвертировать все биты в регистре f.  Если d=0, результат сохраняется в регистре W.  Если d=1, результат сохраняется в регистре f. |
| DECF f, d | Декремент содержимого регистра f.  Если d=0, результат сохраняется в регистре W.  Если d=1, результат сохраняется в регистре f. |
| DECFSZ f, d | Вычесть 1 из f и пропустить если 0. Декремент содержимого регистра f с пропуском, если 0.  Если d=0, результат сохраняется в регистре W.  Если d=1, результат сохраняется в регистре f.  Если результат не равен 0 – исполняется следующая инструкция.  Если результат = 0, то следующая инструкция не выполняется (пропускается, вместо нее выполняется “виртуальный” NOP), а команда выполняется за 2 цикла. |
| INCF f, d | Прибавить 1 к f. Инкремент содержимого регистра f.  Если d=0, результат сохраняется в регистре W.  Если d=1, результат сохраняется в регистре f. |
| INCFSZ f, d | Прибавить 1 к f и пропустить если 0. Инкремент содержимого регистра f с пропуском, если 0.  Если d=0, результат сохраняется в регистре W.  Если d=1, результат сохраняется в регистре f.  Если результат не равен 0 – исполняется следующая инструкция.  Если результат = 0, то следующая инструкция не выполняется (пропускается, вместо нее выполняется “виртуальный” NOP), а команда выполняется за 2 цикла. |
| IORWF f,d | Побитное “ИЛИ” содержимого регистров W и f.  Если d=0 – результат сохраняется в регистре W.  Если d=1 – результат сохраняется в регистре f. |
| MOVF f,d | Содержимое регистра f пересылается в регистр адресата.  Если d=0 – значение сохраняется в регистре W.  Если d=1 – значение сохраняется в регистре f.  d=1 используется для проверки содержимого f на ноль. |
| MOVWF f | Переслать содержимое W в f. |
| NOP | Нет операции. |
| RLF f, d | Выполняется циклический сдвиг влево содержимого регистра f через бит C регистра STATUS (через перенос).  Если d=0 – результат сохраняется в регистре W.  Если d=1 – результат сохраняется в регистре f. |
| RRF f, d | Выполняется циклический сдвиг вправо содержимого регистра f через бит C регистра STATUS (через перенос).  Если d=0 – результат сохраняется в регистре W.  Если d=1 – результат сохраняется в регистре f. |
| SUBWF f, d | Вычетание содержимого регистра W из регистра f.  Если d=0 – результат сохраняется в регистре W.  Если d=1 – результат сохраняется в регистре f. |
| SWAPF f, d | Поменять местами старший и младший полубайты регистра f.  Если d=0 – результат сохраняется в регистре W.  Если d=1 – результат сохраняется в регистре f. |
| XORWF f, d | Сравнение содержимого регистров W и f (проверка на “одинаковость”).  Побитное “Исключающее “ИЛИ” содержимого регистров W и f.  Если d=0 – результат сохраняется в регистре W.  Если d=1 – результат сохраняется в регистре f. |
| BCF f, b | Установить в 0 бит b регистра f. |
| BSF f, bf | Установить в 1 бит b регистра f |
| BTFSC f, b | Проверить бит b в регистре f.  Если бит b в регистре f =1, то исполняется следующая инструкция.  Если бит b в регистре f =0, то следующая инструкция не выполняется (пропускается, вместо нее выполняется “виртуальный” NOP), а команда выполняется за 2 цикла. |
| BTFSS f, b | Проверить бит b в регистре f.  Если бит b в регистре f=0, исполняется следующая инструкция.  Если бит b в регистре f=1, то следующая инструкция не выполняется (пропускается, вместо нее выполняется “виртуальный” NOP), а команда выполняется за 2 цикла. |
| ADDLW k | Содержимое регистра W складывается с 8 – разрядной константой k. Результат сохраняется в регистре W. |
| SUBLW k | Вычесть содержимое регистра W из 8 – разрядной константы k. Результат сохраняется в регистре W. |
| MOVLW k | Переслать константу k в регистр W. В неиспользуемых битах ассемблер устанавливает 0. |
| ANDLW k | Выполняется побитное “И” содержимого регистра W и 8 – разрядной константы k. Результат сохраняется в регистре W. |
| IORLW k | Выполняется побитное “ИЛИ” содержимого регистра W и 8 – разрядной константы k. Результат сохраняется в регистре W. |
| XORLW k | Сравнение содержимого регистра W и константы (проверка на “одинаковость”). Выполняется побитное “Исключающее ИЛИ” содержимого регистра W и 8 – разрядной константы k. Результат сохраняется в регистре W. |
| CALL | Выполнить условный переход (переход по стеку). Адрес следующей инструкции (PC+1) “загружается” в вершину стека (TOS). 11 бит адреса “загружаются” из кода команды в счетчик команд PC<10:0>. 2 старших бита “загружаются” в счетчик команд PC<12:11> из регистра PCLATH. |
| GOTO k | Выполнить безусловный переход (стек не задействован). 11 бит адреса “загружаются” из кода команды в счетчик команд PC<10:0>. 2 старших бита “загружаются” в счетчик команд PC<12:11> из регистра PCLATH. |
| RETURN | Возврат из подпрограммы (по стеку). Вершина стека (TOS) “выгружается” в счетчик команд PC. |
| RETLW k | Возврат из подпрограммы (по стеку). В регистр W загружается 8разрядная константа. Вершина стека (TOS) “выгружается” в счетчик команд PC. |
| RETFIE | Возврат из подпрограммы обработки прерываний (по стеку). Вершина стека (TOS) загружается в счетчик команд PC. Осуществляется предварительное разрешение прерываний (бит №7 регистра INTCON {GIE} устанавливается в 1). |
| CLRWDT | Сброс WDT и предделителя (если он подключен к WDT). В регистре STATUS, биты (TO) и (PD) устанавливаются в 1. Коэффициент деления предделителя (если он подключен к WDT) не меняется. |
| SLEEP | Переход в “спящий режим”. Сброс флага включения питания (PD) в 0. Установка флага переполнения WDT (TO) в 1. Сброс WDT и его предделителя. Перевод микроконтроллера в режим SLEEP и выключение тактового генератора. |

В данной системе применяется датчик Холла марки Honeywell 2AV50A (2AV54). Он специально предназначен для бесконтактных систем зажигания. Диапазон рабочих температур датчика от –40 до +150 градусов Цельсия. Разброс напряжения питания от 4,5 до 16В. Так же датчик имеет кронштейны, позволяющие надежно закрепить его.

Основным параметром, на который нужно ссылаться при выборе остальных элементов схемы, является их масса, так как устройство будет эксплуатироваться в условиях повышенной вибрации, и массивные элементы будут моментально срезаны.

При выборе резисторов R1 (10 КОм.), R2 (100 КОм.), R3 (200 Ом.), R4 (2,2 КОм.) предпочтение отдается импортным углеродистым резисторам типа CF. Они отличаются высокой надежностью и стабильностью параметров, а так же малыми габаритами и массой. Диапазон рабочих температур от –55 до +125 градусов Цельсия. При работе с микроконтроллером мощность резисторов может быть 0,125 Вт или меньше.

При выборе конденсатора С1 (0,1 мкФ.) предпочтение отдается импортным дисковым керамическим конденсаторам серии NP0, так как они обладают высокой стабильностью при изменении температуры. Диапазон рабочих температур – от –20 до +85 градусов Цельсия. Номинальное напряжение – 50В.

Диод VD1 должен быть мощным, поэтому выбор останавливаем на Д 247. Он предназначен для преобразования переменного напряжения частотой до 1,1 кГц. Средний прямой ток – 10А. Средний прямой ток перегрузки в течение 0,5 с. – 30 А. При выборе конденсаторов С2 (47 мкФ.), С3 (1мкФ.) предпочтение отдается импортным электролитическим конденсаторам серии LL, так как они плотно монтируются на плату и, соответственно, менее подвержены вибрациям. Диапазон рабочих температур – от 40 до +85 градусов Цельсия. Максимально допустимое значение переменного тока, проходящего через конденсатор – 116 мА. Рабочее напряжение – 16В.

Резонатор ZQ1 стоит выбирать керамический. Обычно он трехвыводной и имеет встроенные емкости. Такой резонатор лучше переносит вибрацию.

**2.5 Разработка программного обеспечения**

Для осуществления опережения момента искрообразования, экран датчика Холла смещается по направлению, противоположному направлению вращения распредвала. Таким образом, сигнальный импульс будет поступать раньше. А значит, перед подачей выходного сигнала, микроконтроллер должен осуществлять начальное смещение импульса, компенсирующее смещение экрана. Так как экран невозможно установить с идеальной точностью, начальное смещение должно регулироваться. Так же регулировка начального смещения импульса позволяет ограничивать максимальную величину угла опережения зажигания.

Работа механизма опережения зажигания осуществляется уменьшением (увеличением) начального смещения при увеличении (уменьшении) частоты сигнальных импульсов на величину рабочего смещения. Для адаптации устройства к другим моделям двигателей и более точной настройки, величина рабочего смещения должна быть регулируемой. Исходя из этого, к программе предъявляются следующие требования:

1. Определение момента искрообразования, ссылаясь на сигналы датчика Холла.

2. Корректировка угла опережения зажигания в зависимости от частоты входных импульсов по формуле

у = Ти \* х В. (2.4)

где Ти – длительность импульса, х – коэффициент умножения Ти, В – рабочее смещение импульса.

3. Возможность изменения констант начального и рабочего смещения импульса без перепрограммирования и отключения микроконтроллера.

4. Сохранение измененных констант в память EEPROM.

Алгоритм программы приведен в чертеже ДП.230101.802.\*\*\*\*

При поступлении питания, микроконтроллер загружает программу, а так же переменные х и В в ОЗУ из памяти EEPROM, после чего ожидает входной импульс с датчика Холла, сигнализирующий о запуске двигателя. Обеспечив начальное смещение импульса А, равное Ти \* х, и рабочее смещение В, микроконтроллер подает импульс на ключевую схему, обеспечивая тем самым своевременное искрообразование.

Определение изменений частоты вращения двигателя обеспечивается измерением длительности Ти. Величина рабочего смещения импульса В остается неизменной на протяжении всего времени работы устройства. Опережение момента искрообразования достигается путем постоянного вычитания переменной В из начального смещения импульса А, которое изменяется в зависимости от длительности импульсов Ти, а значит и от оборотов двигателя.

При небольших оборотах, длительность импульса Ти, а следовательно значение начального смещения А, велика. Величина рабочего смещения импульса значительно меньше, и вычитание величины В из смещения А даст небольшой угол опережения зажигания.

При высоких оборотах двигателя, длительность импульса Ти уменьшается, уменьшается и значение начального смещения А. Величина В остается неизменной независимо от длительности импульса Ти, следовательно её вычитание из величины А дает гораздо больший угол опережения зажигания.

Применение данного алгоритма позволяет получать своевременное искрообразование в очень широких пределах оборотов двигателя. Значительным плюсом этого алгоритма является плавное, бесступенчатое, изменение угла опережения зажигания.

Текст программы микроконтроллера приведен в Приложении А к дипломному проекту.

**2.6 Расчет надежности**

На этапе проектирования расчет надежности проводится с учетом применяемых элементов и их типов.

Расчет:

Исходные данные:

Требуемая наработка на отказ Тср = 10000 часов.

Плата является не резервируемой.

Таблица 3. – Интенсивность отказов по типам элементов.

|  |  |  |  |
| --- | --- | --- | --- |
| Элемент | Обозначение | Интенсивность отказа λо\*106 1/ч | Количество |
| Печатная плата | λпп | 0,7 | 1 |
| Паянное соединение | λос | 0,01 | 47 |
| Микроконтроллер | λомк | 0,13 | 1 |
| Микросхема | λомкс | 0,2 | 1 |
| Резонатор керамический | λорез | 0,2 | 1 |
| Транзистор высокочастотный | λотр | 1,7 | 1 |
| Диод кремниевый | λод | 1,0 | 1 |
| Резистор | λор | 0,2 | 4 |
| Конденсатор электролитический | λокэ | 2,4 | 2 |
| Конденсатор керамический | λокк | 0,15 | 1 |
| Ключ | λокл | 0,5 | 4 |

λ=К\*λо\*106 (1/ч) (2.5)

λпп=0,7\*1\*106=0,000007 1/ч.

λос=0,01\*47\*106=0,0000047 1/ч.

λомк=0,13\*1\*106=0,0000013 1/ч.

λомкс=0,2\*1\*106=0,000002 1/ч.

λорез=0,2\*1\*106=0,000002 1/ч.

λотр=1,7\*1\*106=0,000017 1/ч.

λод=1,0\*1\*106=0,00001 1/ч.

λор=0,2\*1\*106=0,000002 1/ч.

λокэ=2,4\*2\*106=0,000024 1/ч.

λокк=0,15\*1\*106=0,0000015 1/ч.

λокл=0,5\*4\*106=0,000005 1/ч.

Общая интенсивность отказа:

λ=λпп+λос+λомк+λомкс+λорез+λотр+λод+λор+λокэ+λокк+λокл (2.6)

λ=0,0000765 1/ч.

Среднее время наработки на отказ:

Тср=1/λ ч. (2.7)

Тср=1/0,0000765=13072 ч.

Наработка на отказ данного устройства превышает требуемую наработку на треть, что в свою очередь говорит о работоспособности устройства.

**3. Организация производства**

**3.1 Виды конструкторско технологической документации при производстве электронных устройств**

При проектировании технологического процесса сборки радиоэлектронной аппаратуры и приборов необходима специальная документация. Выбор документов специального назначения производится в зависимости от типа и вида производства, а также технологических методов изготовления изделия. К основным технологическим документам специального назначения относятся:

1. маршрутная карта;

2. карта технологического процесса;

3. операционная карта;

4. карта типового технологического процесса (КТТП);

5. карта типовой операции и др. (ГОСТ 3.1103 82).

Маршрутная карта (МК) содержит описание маршрута технологического процесса изготовления изделия. Кроме того, дополнительно в нее может входить перечень полного состава технологических операций с указанием данных об оборудовании, технологической оснастке, материальных нормативах и трудовых затрат.

Карта технологического процесса (КТП) содержит операционное описание технологического процесса изготовления изделия в технологической последовательности по всем операциям с указанием переходов, технологических режимов, технологической оснастки, материальных и трудовых затрат.

Карта типового технологического процесса содержит описание типового технологического процесса изготовления изделий.

В операционной карте (ОК) дается описание технологических операций с указанием последовательности выполнения переходов, а также приводятся данные об оснастке, режимах и трудовых затратах.

Ведомость материалов (ВМ) содержит данные о подетальных нормах расходов материала и о заготовках.

Чертеж детали представляет собой документ, в котором дается наглядное изображение детали и приводятся данные, необходимые для изготовления и контроля. Правила выполнения чертежей изделия регламентируются ГОСТ 2.109 – 73.

Сборочный чертеж это документ, в котором дается изображение сборочной единицы и приводятся данные, необходимые для её сборки и контроля. В сборочном чертеже указываются расположение и взаимная связь составных частей, соединяемых по данному чертежу.

Спецификация представляет собой документ, определяющий состав сборочной единицы, комплекса или комплекта. Спецификация (ГОСТ 2.108 68) составляется на отдельных листах на каждую сборочную единицу, комплекс или комплект.

Схема это конструкторский документ, на котором показаны в виде условных изображений или обозначений составные части изделия и связи между ними. При проектировании радиоэлектронной аппаратуры и приборов используется схема электрическая (Э), которая в свою очередь разделяется на следующие типы:

1. структурная – схема, определяющая основные функциональные части изделия, их назначение и взаимосвязь.

2. функциональная – схема, разъясняющая определенные процессы, протекающие в функциональных цепях изделия или изделий в целом.

3. принципиальная (полная) – схема, определяющая полный состав элементов и связей между ними и дающая детальное представление о принципах работы изделия

4. монтажная (схема соединений) – схема, показывающая конструктивное выполнение электрических соединений составных частей изделия и определяющая марки провода, жгута и кабеля, которыми эти соединения осуществляется, а так же места их ввода.

5. подключения – схема, показывающая внешние подключения изделий.

6. общая – схема, определяющая составные части комплекса и соединения их между собой на месте эксплуатации.

7. расположения – схема, определяющая относительное расположение составных частей изделия.

**3.2 Маршрутная карта технологического процесса при изготовлении печатной платы**

Как уже говорилось, маршрутная карта (МК) содержит описание маршрута технологического процесса изготовления изделия. Кроме того, дополнительно в нее может входить перечень полного состава технологических операций с указанием данных об оборудовании, технологической оснастке, материальных нормативах и трудовых затрат.

Маршрутная карта применяется в мелкосерийном и серийном производстве. Наименование операций в зависимости от степени детализации технологического процесса может быть кратким, состоящим из одного слова, соответствующего характеру операции или полным. Повторяющиеся наименования операций нумеруются по порядку римскими цифрами. При операционном описании технологического процесса операции обозначаются двузначными числами по порядку их выполнения, переходы каждой операции обозначаются также двузначными числами по порядку их выполнения.

Существует несколько основных способов промышленного производства печатных плат:

1. Химический негативный;
2. Химический позитивный;
3. Метод с применением активирующих паст;
4. Электрохимический метод;
5. Метод фрезерования.

Текст маршрутной карты технологического процесса при промышленном мелкосерийном производстве печатной платы системы управления механизмом зажигания методом химического негативного травления приведен в Приложении В к дипломному проекту. Химический негативный метод производства печатной платы является в данном случае оптимальным, так как он не требует наличия дорогостоящего оборудования и затрачивает небольшое количество химических реактивов.

**4. Экономическая часть**

**4.1 Расчет себестоимости изготовления устройства**

Себестоимость — это удельные эксплуатационные расходы, отнесенные на единицу произведенной продукции или работы (услуги).

Себестоимость является важнейшим, а по своему экономическому содержанию и наиболее совершенным показателем эффективности производства, поскольку в ней комплексно отражается уровень трудовых материальных и финансовых затрат. Себестоимость изготовления единицы продукции представляет собой выраженные в денежной форме все затраты предприятия, объединения или отрасли в целом, отнесенные к объему этой продукции за расчетный период. Следовательно себестоимость включает все затраты прошлого труда и часть затрат живого труда, эквивалентом которых является заработная плата.

Себестоимость это одна из важнейших величин, ориентируясь на которые можно определить уровень прибыльности, востребованности производимой продукции, уровень технического развития производства и конкретно предприятия. Так же на величине себестоимости серьезно сказывается уровень изношенности различного оборудования, уровень квалификации рабочего персонала, величина производительности труда, правильность подбора и распределения персонала и т. д. На величине себестоимости сказываются многие параметры, но одним из важнейших, оказывающих наибольшее влияние, является востребованность продукции предприятия на рынке. От вышеупомянутого параметра напрямую завися объемы производства продукции, а увеличение объемов производства автоматически приводит к снижению себестоимости производимого товара или услуги.

Расчет себестоимости.

Производственные показатели работы мастерской.

Мастерская по производству печатных плат занята непосредственным изготовлением. В мастерской работают 2 электромонтажника, 1 программист, 1 контролер.

Режим работы мастерской:

1. 5ти дневная рабочая неделя – 40 часов.
2. Количество смен – 1.
3. Начало рабочего дня с 9 часов. Обеденный перерыв с 12 до 13.

Каждый специалист лично собирает свое изделие, контролер проверяет продукцию на исправность, программист занимается обновлением текста программы и прошивкой микроконтроллеров.

На сборку одного изделия мастерской установлен норматив времени – 2 часа.

Количество продукции, производимой мастерской за месяц:

Qм=Тм/Тпр (шт.), (4.1)

где Тм – время работы мастерской за месяц в часах, Тпр – нормативное время изготовления единицы продукции в часах.

Тм=Д\*8 (часа), (4.2)

где Д – среднее количество рабочих дней в месяце – 21,5 дней.

Тм=21,5\*8=172 (часа)

Qм=172/2=86 (шт.)

Требуется производство партии (Qобщ) в 100 изделий.

Время, необходимое для производства партии:

Тобщ=Qобщ/Qм (мес.) (4.3)

Тобщ=100/86=1,16 (мес.)

Данное значение для удобства расчетов округляется до 1,2.

Издержки мастерской на производство включают в себя:

1. Материальные затраты – стоимость основных комплектующих, элементов и деталей, основных и вспомогательных материалов;
2. Оплата труда – основная и дополнительная;
3. Отчисления на социальные нужды;
4. Амортизационные отчисления;
5. Накладные расходы.

Материальные затраты.

Таблица 3. Затраты на приобретение основных комплектующих, элементов и изделий

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование элементов. | Количество, шт. | Цена ед. руб. | Общая стоимость, руб. |
| Интегральные микросхемы |  |  |  |
| PIC16F844I/P. | 1 | 108 | 108 |
| KP142EH5A. | 1 | 6 | 6 |
| Диоды |  |  |  |
| К247. | 1 | 7 | 7 |
| Конденсаторы |  |  |  |
| Серии NP0 0.1 мкФ. 50В. | 1 | 2 | 2 |
| Серии LL 47 мкФ 16В. | 1 | 1,5 | 1,5 |
| Серии LL 1 мкФ 16В. | 1 | 1,5 | 1,5 |
| Резисторы |  |  |  |
| Серии CF 0.125 10КОм. | 1 | 1 | 1 |
| Серии CF 0.125 100КОм. | 1 | 1 | 1 |
| Серии CF 0.125 200Ом. | 1 | 1 | 1 |
| Серии CF 0.0125 2.2КОм. | 1 | 1 | 1 |
| Транзисторы |  |  |  |
| КТ3117А. | 1 | 2,5 | 2,5 |
| Резонатор керамический |  |  |  |
| Г3 105 4МГц. | 1 | 10 | 10 |
| Итого |  |  | 142,5 |

Таблица 4. – Затраты на приобретение основных материалов.

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование элементов. | Количество, шт. | Цена ед. руб. | Общая стоимость, руб. |
| Стеклотекстолит, 1дм. кв. | 1 | 30 | 30 |
| Железо хлорное, 100г. | 1 | 50 | 50 |
| Итого |  |  | 80 |

Таблица 5. – Затраты на приобретение вспомогательных материалов.

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование элементов. | Количество, шт. | Цена ед. руб. | Общая стоимость, руб. |
| Припой ПОС61 10гр. | 1 | 15 | 15 |
| Флюс 15мл. | 1 | 15 | 15 |
| Смола эпоксидная 30мл. | 1 | 30 | 30 |
| Итого |  |  | 60 |

Итоговые материальные затраты:

Имз=Иоз+Иом+Ивм (руб.), (4.4)

где Иоз – основная затраты, Иом – основные материалы, Ивм – вспомогательные материалы.

Имз=142,5+80+60=282,5 руб.

Материальные затраты на производство партии:

Имзобщ=Имз\*Qобщ. (4.5)

Имзобщ=282,5\*100=28250 руб.

Оплата труда.

Основная заработная плата отражает затраты на оплату труда производственного персонала, исходя из тарифных ставок и должностных окладов в соответствии с принятыми на предприятии формами и системами оплаты труда.

Дополнительная заработная плата – это все виды доплат, надбавок, премий.

Таблица 6. – Основная заработная плата.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование должности специалистов. | Количество сотрудников. | З/п одного сотрудника, руб. | Общая з/п за месяц, руб. | Общая з/п, руб. |
| Электромонтажники | 2 | 6000 | 12000 | 14400 |
| Программист | 1 | 5000 | 5000 | 6000 |
| Контролер | 1 | 4000 | 4000 | 4800 |
| Изп | 5 |  |  | 25200 |

Дополнительная заработная плата:

Идз=Изп\*0,2 (руб.) (4.6)

Идз=25200\*0,2=5040 (руб.)

Полная оплата труда:

Иот=Изп+Идз (руб.) (4.7)

Иот=25200+5040=30240 (руб.)

Отчисления на социальные нужды осуществляются по определенным нормативам от фонда оплаты труда – 26%:

Исн=26/100\*Иот (руб.)

Исн=26/100\*30240=7862,4 (руб.)

Расчет амортизационных отчислений производится только на оборудование, стоимостью не менее 10 тыс. рублей и сроком эксплуатации более 1 года. При варианте малого производства, рассматриваемом в данном расчете, оборудование, подходящее под данную классификацию, отсутствует.

Накладные расходы.

Это издержки, затраченные на организацию управления предприятием и прочие вопросы, связанные с его функционированием.

Инр=0,8\*Иот (руб.) (4.8)

Инр=0,8\*30240=24192 (руб.)

Общие издержки на производство продукции за год:

Иобщ=Имзобщ+Иот+Исн+Инр (руб.) (4.9)

Иобщ=28250+30240+7862,4+24192=90544,4 (руб.)

Себестоимость единицы продукции:

Sо=Иобщ/Qобщ (руб.) (4.10)

Sо=90544,4 /100=905,5 (руб.)

Составляющие себестоимости.

1. Материальные затраты.

Sмз=Имзобщ/Qобщ (руб.)

Sмз=28250/100=282,5 (руб.)

2. Оплата труда.

Sот=Иот/Qобщ (руб.)

Sот=30240/100=302,4 (руб.)

3. Социальные нужды.

Sсн=Исн/Qобщ (руб.)

Sсн=7826,4/100=78,3 (руб.)

4. Накладные расходы.

Sнр=Инр/Qобщ (руб.)

Sнр=24192/100=241,9 (руб.)

Структура затрат отражает удельный вес каждого элемента в общих издержках мастерской.

1. Материальные затраты.

Имзобщ%=Имзобщ/Иобщ\*100%

Имзобщ%=28250/90544,4\*100=31,3 %

2. Оплата труда.

Иот%=Иот/Иобщ\*100%

Иот%=30240/90544,4\*100=33,4 %

3. Социальные нужды.

Исн%=Исн/Иобщ\*100%

Исн%=7826,4/90544,4\*100=8,6 %

4. Накладные расходы.

Инр%=Инр/Иобщ\*100%

Инр%=24192/90544,4\*100=26,7 %

Таблица 7. – Структура годовых затрат.

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Наименование затрат. | Себестоимость затрат, руб. | Структура затрат, %. |
| 1 | Материальные затраты. | 282,5 | 31,3 |
| 2 | Оплата труда. | 302,4 | 33,4 |
| 3 | Социальные нужды. | 78,3 | 8,6 |
| 4 | Накладные расходы. | 241,9 | 26,7 |

На рынке предлагается два типа подобных устройств екатеринбургского (1800 руб.) и ирбитского производства (3500 руб.). Оба этих устройства являются нерегулируемыми и неремонтопригодными. Исходя из этого, можно вывести на рынок систему управления механизмом зажигания по цене до 2000 руб.

**5. Мероприятия по технике безопасности**

**5.1 Меры безопасности при производстве электромонтажных работ**

Предприятия радиоэлектронной промышленности оснащаются большим количеством разнообразных, а во многих случаях весьма сложных машин, станков и механизмов, облегчающих труд рабочего. Однако все они могут представлять для обслуживающих их рабочих определенную опасность. Это прежде всего относится к машинам, станкам и механизмам, приводимым в движение электрическим током, отдельные элементы которых находятся под давлением сжатого воздуха, пара или газа.

Правильная организация труда, четкость и аккуратность в работе способствует безопасности рабочих на производстве. При выполнении сборочных, монтажных и намоточных работ необходимо строго соблюдать правила техники безопасности.

Общие типовые требования.

1. К самостоятельному выполнению работ следует приступать только после прохождения инструктажа на рабочем месте и усвоения безопасных приемов и методов их выполнения.

2. Выполнять необходимо только ты работу, которая поручена администрацией.

3. При переводе на новый вид работы, приступать к работе можно только после получения от мастера соответствующего инструктажа о безопасных способах её выполнения.

4. Работу следует выполнять только после на исправном оборудовании, исправными инструментами и приспособлениями.

5. Инструмент должен использоваться только по его прямому назначению.

6. Обо всех замеченных неисправностях оборудования и возникших во время работы опасностях для окружающих следует немедленно сообщить мастеру.

7. Запрещается поднимать вручную тяжести, превышающие допустимые нормы: 20 кг. – для женщин и 50 кг. – для мужчин.

8. В случае заболевания или получения травмы следует прекратить работу, сообщить об этом мастеру и обратиться в медпункт за получением первой медицинской помощи.

9. В помещениях, где производятся работы, запрещается хранить личные вещи, принимать пищу и пить воду.

10. Необходимо строго соблюдать правила передвижения в цехах и на территории предприятия. При движении транспорта и перемещении грузов кранами следует отходить в сторону с пути их движения. Запрещается проходить и стоять под поднятым грузом.

11. Рабочая одежда не должна стеснять движений и иметь развевающихся и свисающих концов. На голову следует надеть плотно облегающий головной убор, при этом необходимо убрать под него волосы.

12. Перед началом работы следует внимательно осмотреть рабочее место и привести его в порядок, для чего:

убрать все лишние и мешающие работе предметы;

проверить наличие инструментов;

требующиеся инструменты, приспособления, детали и материал расположить в удобном и безопасном порядке, придерживаясь следующего принципа: то, что берется левой рукой, должно находиться слева, а то, что правой – справа;

подготовить индивидуальные средства защиты и проверить их исправность;

установить сиденье в положение, удобное для работы, чтобы при выполнении рабочих операций не приходилось делать лишних движений руками и корпусом тела;

расположить светильник так, чтобы при выполнении работы не была видна нить накаливания, и свет не слепил глаза.

13. В процессе работы необходимо следить за исправностью используемого оборудования, инструментов и приспособлений; поддерживать на рабочем места чистоту и порядок; не отвлекаться самому и не отвлекать от работы других посторонними разговорами и делами.

14. В случае невыполнения требований настоящей инструкции, виновные несут ответственность согласно правилам внутреннего трудового распорядка.

При написании дипломного проекта была создана принципиальная схема устройства. На её основе была разработана печатная плата, отличающаяся малыми габаритами, небольшим количеством элементов, адаптированная под заданные условия эксплуатации. Исходя из этого можно говорить о высокой надежности системы.

Кроме того, текст программы, написанный для устройства, отличается от аналогов бесступенчатым изменением момента искрообразования и возможностью регулировки устройства в широких пределах. Так же существенным отличием данного программного обеспечения является возможность работы с любой моделью двигателя.

Несомненным плюсом разработанного устройства, кроме вышеперечисленных, является невысокая стоимость, а следовательно, значительная конкурентоспособность.

**Список литературы**

1. Быков К.П., Грищенко П.В. «Мотоциклы «Урал», «Днепр». Эксплуатация и ремонт». Чернигов 2004 208с.

2. Вярочкина Г.О. «Радиоэлектронная аппаратура и приборы. Монтаж и регулировка» М. 2004 352с.

3. Гаряев Л. «Нюансы настройки систем зажигания с датчиком Холла.» МОТО. 2004. №6. с100.

4. Каталог «Промэлектроника. Активные элементы» 2003 108с.

5. Каталог «Промэлектроника. Пассивные элементы» 2003 114с.

6. Косарев С.Н. «Руководство по ремонту автомобилей ВАЗ – 2121 и ВАЗ 21219» М. 1998 184с.

7. Самсонов В.С., Вяткин М.А. «Экономика предприятий энергетического комплекса». М. 2001 315с.

8. Яценков В.С. «Микроконтроллеры MicroCHIP практическое руководство» М. 2005 280с.

П**риложениеА**

Программа микроконтроллера системы управления механизмом зажигания.

; Контроллер: PIC16F84, 4MHz

; Контроллер listp=16f84

; Конфигурация

\_\_CONFIG3ff9h

; Параметры

#define TMR\_VALUE.182; Квантование таймера 100 мкс

; Назначение портов ввода/вывода

#define pCOILPortB,2; Выход управления коммутатором (RB2)

; Направление работы портов ввода/вывода (0 вывод, 1 ввод)

#define PORTB\_IO0xF2; Порт B (11110010) RB1 и RB4RB7 работают на вход, остальные на выход.

; Сбросовые состояния портов

#define PORTB\_RESET0x00; Предустановка портов B (Обнуление)

; Системные константы

Fsrequ 0x04

Indf equ 0x00

Tmr0 equ 0x01 ; Регистр TMR0.

OptionR equ 0x01 ; Option банк1

Status equ 0x03 ; Регистр Status

PortB equ 0x06 ; Порт B

TrisB equ 0x06 ; Tris B Банк1

IntCon equ 0x0B ; Регистр IntCon

EEData equ 0x08 ; EEPROM данные

EECon1 equ 0x08 ; EECON1 банк1.

EEAdr equ 0x09 ; EEPROM адрес

EECon2 equ 0x09 ; EECON2 банк1.

; Память данных

Flags equ 0x10; Флаги состояния

MXHequ 0x0C; Старший байт целой части переменной Х

MXLequ 0x0D; Младший байт целой части переменной Х

EXequ 0x0E; Дробная часть Х, 09;

BHequ 0x1D; Старший байт переменной В

BLequ0x1E; Младший байт переменной В

TimerH equ 0x1A; Счетчик продолжительности сигнала от датчика Холла, старший байт

TimerM equ 0x1B; Средний байт

TimerL equ 0x1C; Младший байт

Secequ 0x11; Счётчик арифметических операций

WrtH equ 0x12; Счётчик времени записи в EEPROM, старший байт

WrtL equ 0x13; Младший байт

MulH equ0x14;Переменные арифметических операций

MulL equ0x15

MulM equ0x16

TmpH equ0x17

TmpM equ0x18

TmpL equ 0x19

Mul1Lo equ0x20

Mul1Hi equ0x21

ResHi equ0x22

DivLo equ0x0F

DivHi equ0x1F

Res1 equ0x23

Res2 equ0x24

Res3 equ0x25

Temp equ0x26

Temp1 equ0x27

Temp2 equ0x28

Temp3 equ0x29

Cntequ0x2A; Счётчик арифметических операций

Res11 equ0x2B

Res22 equ0x2C

Res33 equ0x2D

TmpW equ 0x2E ; Регистр для сохранения состояния W (Аккумулятора)

TmpStatus equ 0x2F ; Регистр для сохранения состояния Status

Системные флаги W equ 0 ; Результат направить в аккумулятор.

F equ 1 ; Результат направить в регистр.

C equ 0 ; Бит флага переносазаема.

Z equ 2 ; Бит флага нулевого результата.

RP0 equ 5 ; Бит выбора банка.

GIE equ 7 ; Бит глобального разрешения прерываний.

T0IE equ 5

T0IF equ 2

; Битовые переменные

#define bLastStateFlags,0; Последнее состояние датчика Холла

#define bSensFlags,1; Флаг состояния датчика Холла после программного фильтра

#define bINCBPortB,7; Флаг нажатия кнопки "Увеличить В"

#define bDECBPortB,6; Флаг нажатия кнопки "Уменьшить В"

#define bINCXPortB,5; Флаг нажатия кнопки "Увеличить Х"

#define bDECXPortB,4; Флаг нажатия кнопки "Уменьшить Х"

#defineTmpFlags,2; Флаг арифметических операций

; Вектор сброса

ORG0x0000

goto Start; Переходим к началу программы

; Вектор прерываний

ORG0x0004

; Interrupt

; Описание: Прерывание от TMR0. Измерение времени прохода шторки по сигналу с датчика Холла.

Interrupt;

movwfTmpW; Сохраняем состояние W и Status в регистры TmpW и TmpStatus

swapfStatus,W

movwfTmpStatus

movlwTMR\_VALUE; Квантование таймера

movwfTmr0

bcfIntCon,T0IF; Сбрасываем флаг прерывания от таймера

movfPortB,W; Программный фильтр по входу датчика Холла

xorwfFlags,W

btfssbSens; Проверяем флаг датчика (1 шторка в датчике, 0 вне датчика)

goto \_Low; Если 0 преходим

btfscbLastState; Проверяем предидущее состояние датчика

goto \_HighToHigh; Если 1 преходим

\_LowToHigh; Шторка вошла в датчик bSens=1 bLastState=0

bsfbLastState; Обновляем данные о состоянии датчика bLastState=1

bsfpCOIL; Включаем комутатор (Выставляем бит в RB2)

PLUS00 incf TimerL,F ; Начинаем отсчёт времени (TimerL(H,M) + 1)

btfss Status,Z

goto YES00

incf TimerM,F

btfss Status,Z

goto YES00

incf TimerH,F

YES00

goto \_ExitInterrupt; Завершаем прерывание

\_HighToHigh; Шторка уже в датчике bSens=1 bLastState=1

PLUS01 incf TimerL,F ; Продолжаем отсчёт времени (TimerL(H,M) + 1)

btfss Status,Z

goto YES01

incf TimerM,F

btfss Status,Z

goto YES01

incf TimerH,F

YES01

goto \_ExitInterrupt

\_Low; Шторка вне датчика bSens=0

btfssbLastState; Проверяем предидущее состояние датчика

goto \_LowToLow

\_HighToLow; Шторка вышла из датчика

bSens=0 bLastState=1

bcfbLastState; Обновляем данные о состоянии датчика bLastState=0

call CalculateSparkTime; Переходим к вычислению функции Y=f(T)

goto \_ExitInterrupt

\_LowToLow; Шторка уже вне датчика

\_ExitInterrupt

DecWrt

DEC002

incf WrtH,W ; Проверяем было ли изменение констант, если было (WrtH<>0xFF), то осчитываем время до записи в память (2 сек)

btfss Status,Z

goto DEC003

goto OK00

DEC003

decf WrtL,F

movlw0xFF

subwfWrtL,W

btfss Status,Z

goto OK00

decf WrtH,F

movlw0xFF

subwfWrtH,W

btfss Status,Z

goto OK00

movlw0xFF; Выставляем 0хFF в WrtH (признак изменения констант)

movwfWrtH

clrfWrtL

call WriteC; Истекло время до записи изменённых переменных в память, переход на подпрограмму записи

OK00

swapfTmpStatus,W; Восстанавливаем соотояние W и Status и выходим из прерывания

movwfStatus

swapfTmpW,F

swapfTmpW,W

retfie

; Start

; Описание: Начало программы, инициализация контроллера и установка в начальное состояние

Start;

clrwdt

clrfIntCon; Обнуляем регистр прерываний

bsfStatus,RP0; Выбор банка памяти 1

movlwPORTB\_IO

movwfTrisB

movlw0x88

movwfOptionR

bcfStatus,RP0; Выбор банка памяти 0

movlwPORTB\_RESET

movwfPortB

clrf TimerH

clrf TimerM

clrf TimerL

movlw0xFF

movwfWrtH

clrfWrtL

movfPortB,W; Выставляем флаги в начальное состояние

andlw0xF2

movwfFlags

movlwTMR\_VALUE; Квантование таймера

movwfTmr0

; Чтение констант из памяти

; movlw 0 ; Записать в регистр W константу 0 (адрес ячейки в памяти).

movwf EEAdr ; Скопировать 0h из регистра W в регистр EEAdr.

bsf Status,RP0 ; Переход в первый банк.

bsf EECon1,0 ; Инициализировать чтение.

bcf Status,RP0 ; Переход в нулевой банк.

movf EEData,W ; Скопировать число из ячейки EEPROM с адресом 0h в регистр W.

movwf MXH ; Скопировать число из регистра W в регистр MXH.

nop; Пустой оператор (ожидание)

movlw 1

movwf EEAdr

bsf Status,RP0

bsf EECon1,0

bcf Status,RP0

movf EEData,W

movwf MXL

nop

movlw 2

movwf EEAdr

bsf Status,RP0

bsf EECon1,0

bcf Status,RP0

movf EEData,W

movwf EX

nop

movlw 3

movwf EEAdr

bsf Status,RP0

bsf EECon1,0

bcf Status,RP0

movf EEData,W

movwf BH

nop

movlw 4

movwf EEAdr

bsf Status,RP0

bsf EECon1,0

bcf Status,RP0

movf EEData,W

movwf BL

movlw0xA0; Разрешаем прерывания от TMR0, и прерывания глобально, сбрасываем флаги прерываний

movwfIntCon

Cycle; Главный цикл программы

bsfIntCon,T0IE; Разрешаем прерывания от TMR0

bsfIntCon,GIE; Разрешаем прерывания глобально

movfPortB,W

xorwfFlags,W

btfsc bINCB;Если RB7 = 1, то переходим к увеличению переменной В

goto IncB

btfsc bDECB;Если RB6 = 1, то переходим к уменьшению переменной В

goto DecB

btfsc bINCX;Если RB5 = 1, то переходим к увеличению переменной Х

goto IncX

btfsc bDECX;Если RB4 = 1, то переходим к уменьшению переменной Х

goto DecX

gotoCycle

IncB; Подпрограмма увеличения переменной В

PLUS02

incf BL,F

btfss Status,Z

goto YES02

incf BH,F

btfss Status,Z

goto YES02

movlw0xFF

movwfBH

movwfBL

YES02

gotoYES07

DecB; Подпрограмма уменьшения переменной В

PLUS03

decf BL,F

movlw0xFF

subwfBL,W

btfss Status,Z

goto YES03

decf BH,F

movlw0xFF

subwfBH,W

btfss Status,Z

goto YES03

clrfBH

clrfBL

YES03

gotoYES07

IncX; Подпрограмма увеличения переменной Х

PLUS04

incf EX,F

movlw0x0A

subwfEX,W

btfss Status,Z

gotoYES05

clrfEX

PLUS05

incf MXL,F

btfss Status,Z

goto YES05

incf MXH,F

btfss Status,Z

goto YES05

movlw0xFF

movwfMXH

movwfMXL

movlw0x09

movwfEX

YES05

gotoYES07

DecX; Подпрограмма уменьшения переменной Х

PLUS06

decf EX,F

movlw0xFF

subwfEX,W

btfss Status,Z

goto YES07

decf MXL,F

movlw0xFF

subwfMXL,W

btfss Status,Z

goto YES06

decf MXH,F

movlw0xFF

subwfMXH,W

btfss Status,Z

goto YES06

clrfMXH

clrfMXL

clrfEX

gotoYES07

YES06

movlw0x09

movwfEX

YES07; Выход из подпрограм изменения переменных, инициализация отсчёта времени для записи в память (2 сек, 0x4E20)

movlw0x4E

movwfWrtH

movlw0x20

movwfWrtL

nop; Ожидание отпускания кнопок изменения переменных и переход на главный цикл программы

nop

nop

nop

gotoCycle

; Сохранение констант в энергонезависимую память

;WriteC

Bcf IntCon,GIE ; Глобальный запрет прерываний.

bcf Status,RP0 ; Переход в нулевой банк.

movlw 0 ; Записать в регистр W константу 0 (адрес ячейки в памяти).

movwf EEAdr ; Скопировать константу 0 из регистра W в регистр EEAdr.

movf MXH,W ; Скопировать число из регистра МХН в регистр W.

movwf EEData ; Скопировать число из регистра W в ячейку EEPROM с адресом 0.

bsf Status,RP0 ; Переход в первый банк.

bsf EECon1,2 ; Разрешить запись.

movlw 0x55 ; Обязательная

movwf EECon2 ; процедура

movlw 0xAA ; при записи.

movwf EECon2 ; "

bsf EECon1,1 ; "

WR0

Btfsc EECon1,1;Ожидание записи

gotoWR0

bcf EECon1,4 ; Сбросить флаг прерывания по окончании записи в EEPROM.

bcf Status,RP0

movlw 1

movwf EEAdr

movf MXL,W

movwf EEData

bsf Status,RP0

bsf EECon1,2

movlw 055h

movwf EECon2

movlw 0AAh

movwf EECon2

bsf EECon1,1

WR1

btfscEECon1,1

gotoWR1

bcf EECon1,4

bcf Status,RP0

movlw 2

movwf EEAdr

movf EX,W

movwf EEData

bsf Status,RP0

bsf EECon1,2

movlw 055h

movwf EECon2

movlw 0AAh

movwf EECon2

bsf EECon1,1

WR2

btfscEECon1,1

gotoWR2

bcf EECon1,4

bcf Status,RP0

movlw 3

movwf EEAdr

movf BH,W

movwf EEData

bsf Status,RP0

bsf EECon1,2

movlw 055h

movwf EECon2

movlw 0AAh

movwf EECon2

bsf EECon1,1

WR3

btfscEECon1,1

gotoWR3

bcf EECon1,4

bcf Status,RP0

movlw 4

movwf EEAdr

movf BL,W

movwf EEData

bsf Status,RP0

bsf EECon1,2

movlw 055h

movwf EECon2

movlw 0AAh

movwf EECon2

bsf EECon1,1

WR4

btfscEECon1,1

gotoWR4

bcf EECon1,4

return; Выход из подпрограммы записи в память

;Арифметические подпрограммы

SUM00 ; Операция сложения регистров MulL, MulM, MulH и TmpL, TmpM, TmpH, с учётом разряда

movf MulL,W

addwf TmpL,F

btfss Status,C

goto SUM01

incf TmpM,F

btfss Status,Z

goto SUM01

incf TmpH,F

SUM01

movf MulM,W

addwf TmpM,F

btfsc Status,C

incf TmpH,F

movf MulH,W

addwf TmpH,F

return

MIN00; Операция вычитания регистров MulL, MulM, MulH и TmpL, TmpM, TmpH, с учётом разряда

movf MulL,W

subwf TmpL,F

btfsc Status,C

goto MIN01

btfsc Status,Z

goto MIN01

decf TmpM,F

movlw0xFF

subwfTmpM,W

btfss Status,Z

goto MIN01

decf TmpH,F

movlw0xFF

subwfWrtH,W

btfss Status,Z

goto MIN01

clrfTmpH

clrfTmpM

clrfTmpL

MIN01

movf MulM,W

subwf TmpM,F

btfsc Status,C

goto MIN02

btfsc Status,Z

goto MIN02

decf TmpH,F

movlw0xFF

subwfTmpH,W

btfss Status,Z

goto MIN02

clrfTmpH

clrfTmpM

MIN02

movf MulH,W

subwf TmpH,F

btfsc Status,C

goto MIN03

btfsc Status,Z

goto MIN03

clrfTmpH

MIN03 ; Операция умножения регистров MulL, MulH на множитель MulM, с учётом разряда

return

MUL00

movlw0x09

movwfSec

movfMulM,W

clrfMulH

bcfStatus,C

MUL01

rrfMulH,F

rrfMulL,F

btfssStatus,C

gotoMUL02

addwfMulH,F

MUL02

decfszSec,F

gotoMUL01

return

Div24\_16 ; Операция деления регистров Mul1L, Mul1H, ResHi на делитель DivLo, DivHi, с учётом разряда

clrf Temp

clrf Temp2

clrf Temp3

clrf Cnt

clrf Res1

clrf Res2

clrf Res3

movlw 0xf0

andwf DivHi,0

btfsc Status,Z

goto chknxt

movf ResHi,0

movwf Temp2

movf Mul1Hi,0

movwf Temp

movlw .2

movwf Sec

goto StartDiv

chknxt

movf DivHi,1

btfsc Status,Z

goto chknxt1

movlw 0xf0

andwf ResHi,0

movwf Temp2

swapf Temp2,1

movlw 0x0f

andwf ResHi,0

movwf Temp

swapf Temp,1

movlw 0xf0

andwf Mul1Hi,0

movwf Temp3

swapf Temp3,0

iorwf Temp,1

clrf Temp3

movlw .3

movwf Sec

goto StartDiv

chknxt1

movlw 0xf0

andwf DivLo,0

btfsc Status,Z

goto chknxt2

movf ResHi,0

movwf Temp

movlw .4

movwf Sec

goto StartDiv

chknxt2

movlw 0xf0

andwf ResHi,0

movwf Temp

swapf Temp,1

movlw .5

movwf Sec

StartDiv

sbagn

call Sub

btfsc Tmp

goto FinSub

incf Cnt,1

goto sbagn

Sub

bcf Tmp

movf DivLo,0

subwf Temp,1

btfsc Status,C

goto sub2nd

decf Temp2,1

movlw 0xff

subwf Temp2,0

btfss Status,Z

goto sub2nd

decf Temp3,1

movlw 0xff

subwf Temp3,0

btfsc Status,Z

bsf Tmp

sub2nd

movf DivHi,0

subwf Temp2,1

btfsc Status,C

return

decf Temp3,1

movlw 0xff

subwf Temp3,0

btfsc Status,Z

bsf Tmp

return

FinSub

movf DivLo,0

addwf Temp,1

btfss Status,C

goto add2nd

incfsz Temp2,1

goto add2nd

incf Temp3,1

add2nd

movf DivHi,0

addwf Temp2,1

btfss Status,C

goto jpqm

incf Temp3,1

jpqm

bcf Status,C

rrf Sec,0

movwf Fsr

movlw Res1

addwf Fsr,1

btfss Sec,0

goto nhjuq

swapf Cnt,0

iorwf Indf,1

goto ffr

nnhjuq

movf Cnt,0

iorwf Indf,1

ffr

decf Sec,1

movlw 0xff

subwf Sec,0

btfss Status,Z

goto notFin

goto FinDiv

notFin

bcf Status,C

rlf Temp,1

rlf Temp2,1

rlf Temp3,1

bcf Status,C

rlf Temp,1

rlf Temp2,1

rlf Temp3,1

bcf Status,C

rlf Temp,1

rlf Temp2,1

rlf Temp3,1

bcf Status,C

rlf Temp,1

rlf Temp2,1

rlf Temp3,1

bcf Status,C

rrf Sec,0

movwf Fsr

movlw Mul1Lo

addwf Fsr,1

btfsc Sec,0

goto nnhq

movlw 0x0f

andwf Indf,0

iorwf Temp,1

goto jjddsak

nnhq

swapf Indf,0

andlw 0x0f

iorwf Temp,1

jjddsak

clrf Cnt

goto sbagn

FinDiv ; Операция округления при делении

bcf Status,C

rrf DivHi,1

rrf DivLo,1

movf DivHi,0

subwf Temp2,0

btfss Status,C

goto dsgfadsg

btfss Status,Z

goto cxv

movf DivLo,0

subwf Temp,0

btfss Status,C

goto dsgfadsg

cxv

incfsz Res1,1

goto dsgfadsg

incfsz Res2,1

goto dsgfadsg

incf Res3,1

dsgfadsg

movf Res1,0

movwf Mul1Lo

movf Res2,0

movwf Mul1Hi

movf Res3,0

movwf ResHi

return

;Подпрограмма расчёта времени удержания сигнала коммутатору Y=f(T) (Y=T\*XB)

CalculateSparkTime;

clrfTmpH

clrfTmpM

clrfTmpL

movfTimerL,W ; Множим TimerL(H,M) на MX(L,H)

movwfMulL

movfMXL,W

movwfMulM

callMUL00

movfMulH,W

movwfMulM

clrfMulH

callSUM00

movfTimerM,W

movwfMulL

movfMXL,W

movwfMulM

callMUL00

movfMulL,W

movwfMulM

clrfMulL

callSUM00

movfTimerH,W

movwfMulL

movfMXL,W

movwfMulM

callMUL00

movfMulL,W

movwfMulH

clrfMulL

clrfMulM

callSUM00

movfTimerL,W

movwfMulL

movfMXH,W

movwfMulM

callMUL00

movfMulL,W

movwfMulM

clrfMulL

callSUM00

movfTimerM,W

movwfMulL

movfMXH,W

movwfMulM

callMUL00

movfMulL,W

movwfMulH

clrfMulL

clrfMulM

callSUM00 ;Делим TimerL(H,M) на 10

movf TimerH,W;Старший байт делимого

movwf ResHi

movf TimerM,W;Средний байт делимого

movwf Mul1Hi

movf TimerL,W;Младший байт делимого

movwf Mul1Lo

movlw 0x00;Старший байт делителя

movwf DivHi

movlw 0x0A;Младший байт делителя

movwf DivLo

callDiv24\_16

movfTmpH,W

movwfTimerH

movfTmpM,W

movwfTimerM

movfTmpL,W

movwfTimerL

clrfTmpH

clrfTmpM

clrfTmpL; Множим TimerL(H,M) на EX

movfMul1Lo,W

movwfMulL

movfEX,W

movwfMulM

callMUL00

movfMulH,W

movwfMulM

clrfMulH

callSUM00

movfMul1Hi,W

movwfMulL

movfEX,W

movwfMulM

callMUL00

movfMulL,W

movwfMulM

clrfMulL

callSUM00

movfResHi,W

movwfMulL

movfEX,W

movwfMulM

callMUL00

movfMulL,W

movwfMulH

clrfMulL

clrfMulM

callSUM00

movfTmpH,W

movwfMulH

movfTmpM,W

movwfMulM

movfTmpL,W

movwfMulL

movfTimerH,W

movwfTmpH

movfTimerM,W

movwfTmpM

movfTimerL,W

movwfTmpL

callSUM00; Вычитаем из TimerL(H,M) B(L,H) clrfMulH

movfBH,W

movwfMulM

movfBL,W

movwfMulL

callMIN00

movfTmpH,W

movwfTimerH

movfTmpM,W

movwfTimerM

movfTmpL,W

movwfTimerL; Подпрограмма отсчёта времени удержания комутатора

DecTime

bsfpCOIL; Подаём сигнал на комутатор (1 в RB2)

movlw0xFF; Отсчитываем квант времени таймера

movwfSec

movlwTMR\_VALUE

subwfSec,F

PAUSE

decfszSec,F

gotoPAUSE

PLUS08

decf TimerL,F; Начинаем отсчёт времени (TimerL(H,M) 1)

movlw0xFF

subwfTimerL,W

btfss Status,Z

goto YES08

decf TimerM,F

movlw0xFF

subwfTimerM,W

btfss Status,Z

goto YES08

decf TimerH,F

movlw0xFF

subwfTimerH,W

btfss Status,Z

goto YES08

clrf TimerH ; Отсчёт времени (TimerL(H,M) 1) завершён

clrf TimerM

clrf TimerL

bcfpCOIL; ; Выставляем 0 в RB2 (отключаем сигнал коммутатору)

return

YES08

gotoDecTime

end