Введение

Существует устройство для автоматизированной проверки блоков релейной логики, позволяющее подавать сигналы на входы проверяемого устройства и получать сигналы с его выходов. Результат о исправности или не исправности блока формируется за счет сравнения данных, полученных на его выходах с эталонными значениями. Основным недостатком разработанной модели было отсутствие возможности изменения программы проверки. Для решения данной проблемы возможно создание языка программирования, предоставляющего инструменты для написания программы тестов. Основной концепцией разрабатываемого языка является простота, позволяющая писать программы тестов, не имея большого опыта в программировании.

1. Анализ проблемной области

На первом этапе работы было произведено сравнение существующих языков программирования для микроконтроллеров, выявлены их достоинства и недостатки. На основе анализа было сформировано техническое задание.

* 1. Обзор языков программирования

1.1.1. Ассемблер

Команды языка программирования - инструкции процессора. Каждая команда состоит из названия операции и операндов, к которым будет применена выбранная операция. Операндами могут являться как регистры, так и числа.

Так как любая команда подразумевает взаимодействие с регистрами, то пользователю необходимо обладать знаниями об архитектуре процессора для написания корректно работающей программы. Кроме того, необходимо иметь опыт написания программ на языках низкого уровня, так как они имеют ряд особенностей: наличие команды переходов для возможности организации циклов и ветвлений; наличие типов данных и т.д.

С другой стороны, приближенность команд к инструкциям процессора позволяет создавать компактные и эффективные исполняемые модули. Это достигается за счет отсутствия приведения команд к инструкции, выполняемой процессором. И как следствие, отсутствие избыточных или ненужных команд.

1.1.2. С для Arduino

Предоставляет набор команд для взаимодействия с микроконтроллером. Каждая команда является функцией, описание которой скрыто от пользователя и подключается только в момент компиляции программы. Написание программы подразумевает вызов той или иной функции с передачей в нее параметров. Это значительно упрощает процесс программирования из-за чего пользователем может быть человек, не имеющего опыт в программировании. Но опыт программирования все же должен быть, так как параметрами функции могут быть переменные, чей тип необходимо указывать. Кроме того, название программ на английском, что еще сильнее усложняет процесс программирования для русскоговорящего пользователя.

Основным недостатком такого подхода является отдаленность команд от инструкций процессора. Следовательно, при создании исполняемого модуля необходима трансляция всех выражений в эти инструкции. Из-за этого возникают избыточные команды, дублирующие друг друга команды, что сказывается на объеме исполняемого модуля.

1.1.3. Scratch

Синтаксические конструкции представляют собой визуальные блоки, которые можно последовательно соединять, тем самым задавая алгоритм работы микроконтроллера. Визуальные блоки имеют такой же принцип работы, как функции на С для Arduino, но параметры функции не имеют явной типизации. Это позволяет сделать написание кода доступным для людей, не имеющих опыта в программировании. Кроме того, синтаксические конструкции написаны на русском языке, что упрощает процесс программирования для русскоговорящего населения.

При создании исполняемого модуля все блоки сначала транслируются в команды на языке С, а после в команды процессора. Такая многоуровневая трансляция сказывается на скорости создания исполняемого модуля, на его компактности и эффективности.

Сравнение языков программирования представлено в таблице 1

Таблица 1 – Сравнение языков программирования

|  |  |  |  |
| --- | --- | --- | --- |
|  | Ассемблер | С | Scratch |
| 1. Поддержка русского языка | - | - | + |
| 2. Простота написания программы | - | + | + |
| 3. Компактность исполняемого модуля | + | - | - |
| 4. Эффективность исполняемого модуля | + | - | - |

1.2. Актуальность разработки

Существующие языки программирования имеют ряд преимуществ: поддержку русского языка, простоту написания программы, компактность исполняемого модуля и его эффективность. Но ни один из рассмотренных языков не содержит в себе все эти достоинства.

В результате было принято решение о разработке языка программирования, включающего в себя все преимущества рассмотренных языков.

1.3. Техническое задание

1.3.1. Наименование системы

Полное наименование системы: система для создания тестов автоматизированной проверки блоков релейной логики.

1.3.2. Краткая характеристика области применения

Продукт предназначен для возможности написания тестов автоматизированной проверки блоков релейной логики

1.3.3. Цели создания

Функциональным назначением программы является предоставление возможности написания тестов для устройства проверки блоков релейной логики.

Программа должна эксплуатироваться на ПК, установленных в кабинетах на предприятии. Особые требования к конечному пользователю не предъявляются.

1.3.4. Требования к программе

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

1) функции написания программы;

2) функции сохранения и открытия программы;

3) функции проверки исправности написанной программы;

4) функции компиляции программы в объектный код;

5) функции загрузки объектного кода в микроконтроллер

В состав технических средств должен входить ПК, включающий в себя:

1. 32 или 64 разрядный процессор с тактовой частотой не меньше 1.0

ГГц;

2) дисплей;

3) не менее 1 Гб оперативной памяти;

4) не менее 100 мегабайт свободного дискового пространства;

5) клавиатура и мышь.

Системные программные средства, используемые программой, должны быть представлены ОС Windows 7/8/10;

Программа должна обеспечивать взаимодействие с пользователем по средствам графического интерфейса и предоставлять возможность выполнять наиболее часто используемые операции с помощью сочетаний клавиш на клавиатуре.

1.3.5. Требования к программной документации

Состав программной документации должен включать в себя:

1) техническое задание;

2) руководство пользователя;

3) исходный код.

1.3.6. Стадии и этапы разработки

Разработка должна быть приведена в 3 стадии:

1) разработка технического задания;

2) проектирование;

3) внедрение.

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и разработки технического задания.

На стадии проектирования необходимо выполнить следующие этапы:

1) разработка программы;

2) разработка программной документации;

3) испытание программы.

На этапе внедрения выполняется передача программы заказчику.