**Слайд 1**

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.

**Слайд 2**

На первом этапе работы было произведено сравнение существующих языков программирования для микроконтроллеров, выявлены их достоинства и недостатки. В результате было принято решение о создании нового языка программирования, который бы включал в себя все преимущества: поддержку русского языка, простоту написания программы, компактность исполняемого модуля и его эффективность.

**Слайд 3**

Для создания языка программирования необходимо написать компилятор, который будет его обрабатывать.

Для написания компилятора необходимо понимать его структуру. Для этого была построена диаграмма прецедентов. После определения составных частей компилятора началась их разработка.

**Слайд 4**

Для лексического анализатора была разработана диаграмма переходов состояний, позволяющая выделять лексемы из входного текста. Далее было произведено кодирование и тестирование.

**Слайд 5**

Синтаксический анализатор строит дерево разбора при помощи грамматики, которая так же была разработана. После ее разработки она была програмно реализована, и анализатор был протестирован.

**Слайд 6**

На этапе семантического анализатора была определена структура таблицы имен, а также разработан алгоритм работы анализатора. Выполнена программная реализация и тестирование

**Слайд 7**

Для генератора промежуточного кода также был разработан алгоритм работы, программно реализован и протестирован.

**Слайд 8**

Аналогичные действия были выполнены и для генератора кода.

**Слайд 9**

Для понимания взаимодействия между частями компилятора была разработана диаграмма последовательности. На ее основе был сделан вывод о необходимости использования try-catch для программной реализации.

**Слайд 10**

В результате выполнения курсового проекта был разработан программный продукт, с содержанием всех преимуществ, которые были заявлены в начале.