

1 Введение

Полное наименование программного продукта: «Интерпретатор командной оболочки для встраиваемых систем на базе микросхем семейства AVR». Продукт создается в рамках курсовой работы по дисциплине “проектирование трансляторов”.

2 Назначение и область применения

Система предназначена для записи на микросхемы и последующего использования для ускорения разработки и отладки аппаратной части встраиваемой системы, создания единого протокола обмена данными между микроконтроллером и терминальным управляющим устройством.

3 Технические характеристики

3.1 Постановка задачи и математические методы

см. ??

3.2 Описание алгоритма функционирования программы

Программа осуществляет анализ входных строк, состоящих из ASCII-символов.

Система состоит из следующего набора модулей:

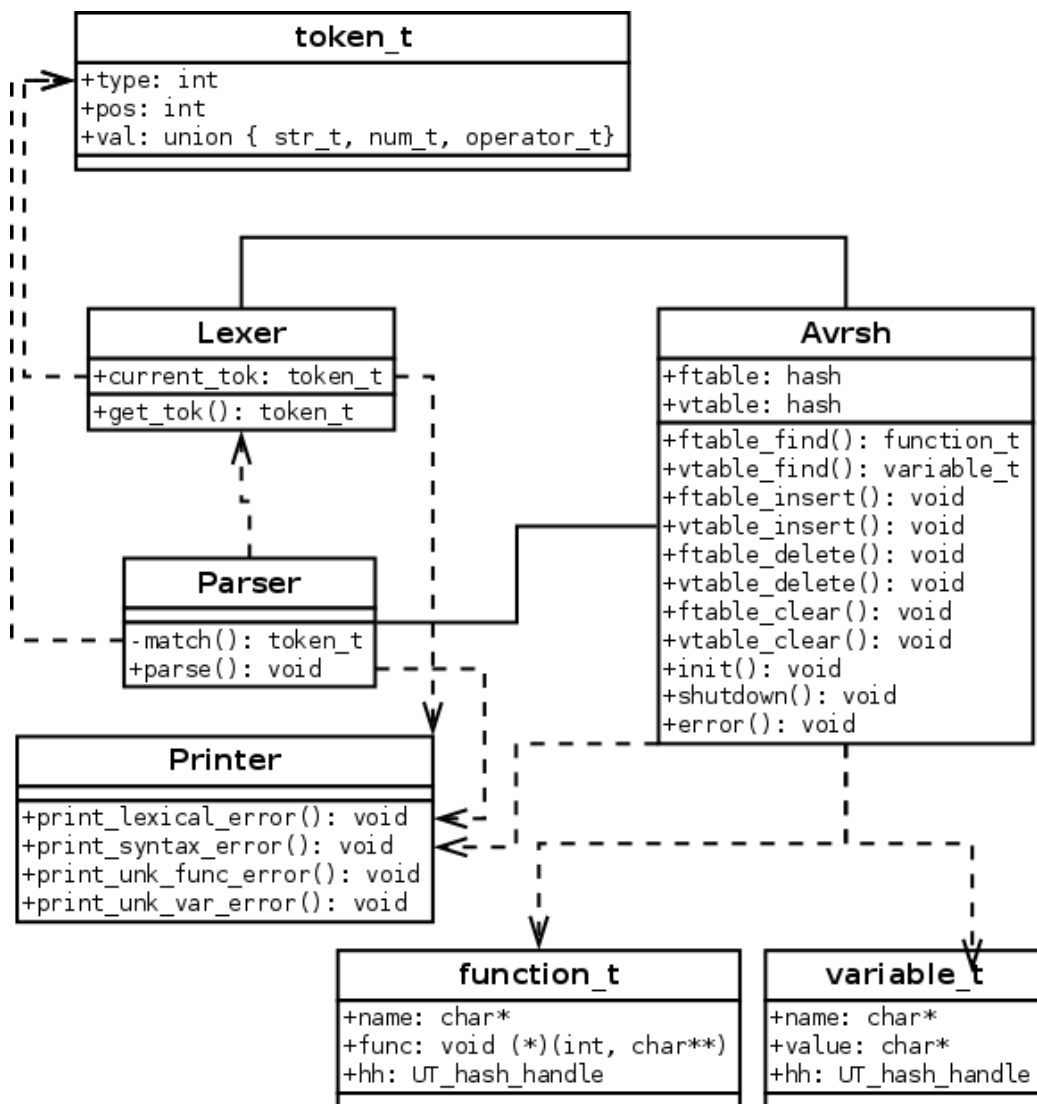


Рис. 1: Диаграмма классов

Модуль **Lexer** осуществляет лексический анализ и токенизацию входной строки. Используя структуру **token_t** передает данные модулю синтаксического анализа. Используется целочисленный тип для передачи типа токена и объединение для передачи значения.

Модуль **Parser** осуществляет синтаксический анализ входной строки. Представляет собой рекурсивный нисходящий предиктивный распознаватель. Грамматика: см. Приложение А.

Модуль **Avsrh** – модуль исполнения команд. В нем хранятся данные о пользовательских командах, о переменных в виде хеш-таблиц.

Значения переменных хранятся только в виде строк. При необходимости в вычислении-подстановке можно получить числовое значение с помощью оператора **#_**. Поддерживаются только целые числа.

Общий алгоритм работы программы:

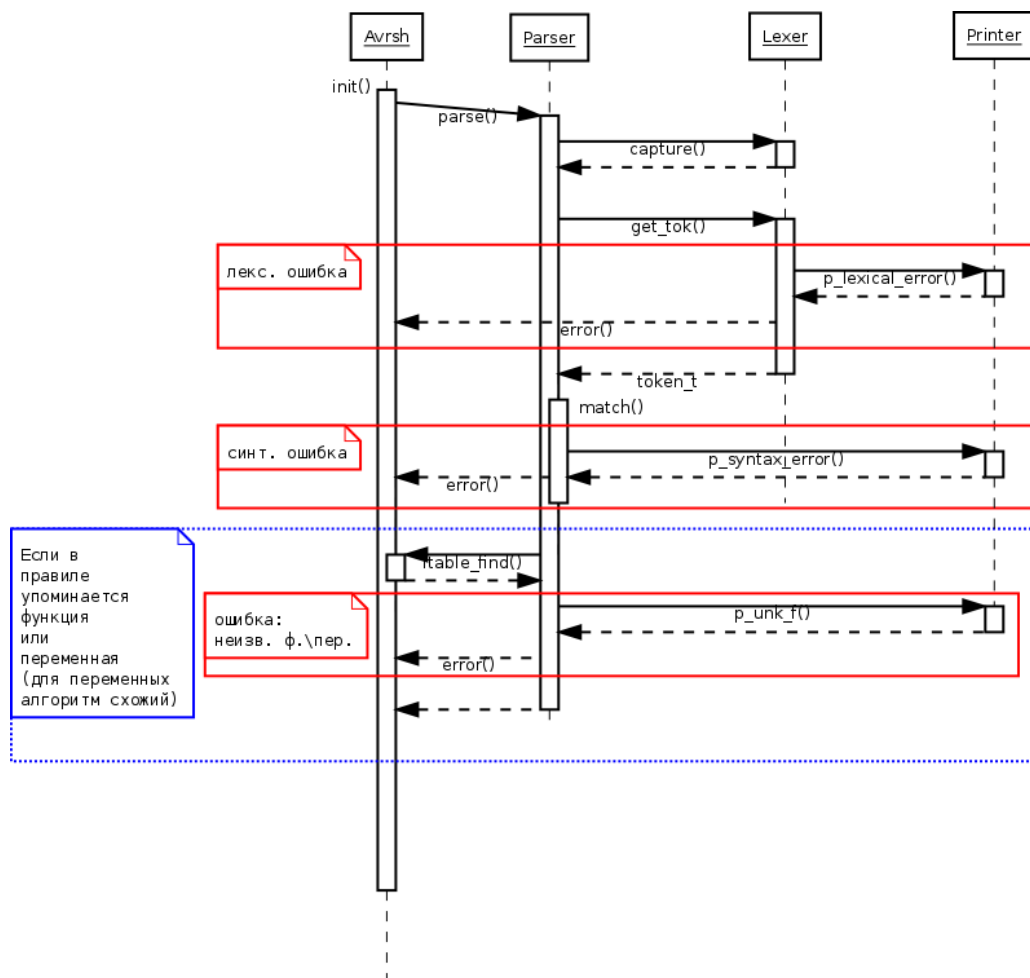


Рис. 2: Диаграмма последовательности

3.3 Описание входных и выходных данных

Приложение

А Приложение А. Грамматика специализированного языка

control	→	statement control
	→	ifst control
	→	loop control
	→	set control
	→	ε
statement	→	ID args DELIM
args	→	ID args
	→	get args
	→	eval args
	→	ε
ifst	→	IF eval branch
branch	→	control maybe_branch
maybe_branch	→	ELSE control END
	→	ELIF eval branch
	→	END
loop	→	WHILE eval control END
set	→	SET ID [=] ID
get	→	\$ ID
getn	→	# ID
eval	→	(aexpr)
aexprlv	→	op
aexprorv	→	+ op
	→	- op
	→	* op
	→	/ op
	→	% op
	→	< op
	→	> op
	→	>= op
	→	<= op
	→	== op
	→	!= op
	→	ε
op	→	NUMBER
	→	' ID '
	→	get
	→	getn

Приложение Б. Исходный код