Context free grammar

Program

<stmts> -> <stmt> <stmts>

| λ

<stmt> -> <if stmt>

| <assign>

| <serial stmt>

Serial

<serial stmt> -> SEND <string>

| BAUD <nr>

<string> -> "[letter|nr]*"

| "[letter|nr]*" + <string>

| <expr>

<letter> \rightarrow [a-z|A-Z]

IF

<if stmt> -> IF <logic expr> <assign>

Assign

<assign> -> SET <assignterm>

<assignterm> -> <id> <expr>, <assignterm>

| <id> <expr>

| <id> [+=|-=] <value expr>

Expression

<expr> -> <value expr>

| <logic expr>

<value expr> -> <term>

| <value expr> [+|-] <term>

<term> -> <value> [*|/|%] <term>

| READ

<logic_expr> -> <logic_value>

| <logic_value> [OR|AND] <logic_expr>

| NOT <logic value>

| <value expr> [<|>|>=|<=] <value expr>

Identifier

<id> -> FLAG#<set>

| PIN#<set> | PINA#<set> | COUNTER#<set> | TIMER#<set>

Number

< nr > -> [0-9] +

 $| [0-9] + \langle suffix \rangle$

 $\langle suffix \rangle$ -> ms | s | m | t | d | u

Set

<set> -> [<nr>|<range>]

| <set>, [<nr>|<range>]

<range> -> <nr>..<nr>

Value

<value> -> <nr>

| <id>

<logic_value> -> HIGH

| LOW

| <id>