Specialized Computing and Robotics Language (SCARL) Grammar *(first draft)*

*program statement\_list*

*statement\_list statement | statement statement\_list*

*statement device\_declarator\_statement | primitive\_definition\_statement | function\_definition\_statement*

*block\_statment* ***‘*{‘** *statement\_list\_block\_level* **‘}’**

*statement\_list\_block\_level statement\_block\_level | statement\_block\_level statement\_list\_block\_level*

*statement\_block\_level primitive\_definition\_statement | block\_statement | variable\_set\_statement | function\_invocation\_statement | if\_block\_statement | while\_block\_statement*

*device\_declarator\_statement device\_type* **‘IDENTIFIER’ ‘;’**

*primitive\_declarator primitive\_type* **‘IDENTIFIER’**

*primitive\_definition\_statement primitive\_declarator* **‘=’** *expression* **‘;’**

*function\_definiton\_statement primitive\_declarator* **‘(‘** *formal\_parameter\_list* **‘)’** *block\_statement*

*variable\_set\_statement* **‘IDENTIFIER’ ‘=’** *expression* **‘;’**

*function\_invocation* **‘IDENTIFIER’ ‘(‘** *parameter\_list* **‘)’**

*function\_invocation\_statement function\_invocation* **‘;’**

*if\_block\_statement* **‘if’ ‘(‘** *expression* **‘)’** *block\_statement |***‘if’ ‘(‘** *expression* **‘)’** *block\_statement* **‘else’** *block\_statement*

*while\_block\_statement* **‘while’ ‘(‘** *expression* **‘)’** *block\_statement*

*formal\_parameter\_list primitive\_declarator |  
 primitive\_declarator* **‘,’** *formal\_parameter\_list |*

*parameter\_list expression | expression* **‘,’** *parameter\_list* |

*expression logical\_expression*

*logical\_expression logical\_and\_expression |  
 logical\_expression* **‘||’** *logical\_and\_expression*

*logical\_and\_expression equality\_expression |   
logical\_and\_expression* **‘&&’** *equality\_expression*

*equality\_expression relational\_expression|  
 equality\_expression* **‘==’** *relational\_expression |  
 equality\_expression* **‘!=’** *relational\_expression*

*relational\_expression bool\_expression |   
relational\_expression* **‘>’** *bool\_expression |  
relational\_expression* **‘<’** *bool\_expression |  
relational\_expression* **‘>=’** *bool\_expression |  
relational\_expression* **‘<=’** *bool\_expression*

*bool\_expression arithmetic\_expression |***‘!’** *arithmetic\_expression*

*arithmetic\_expression arithmetic\_factor |   
arithmetic\_expression* **‘+’** *arithmetic\_factor |  
arithmetic\_expression* **‘-‘** *arithmetic\_factor*

*arithmetic\_factor arithmetic\_unary |  
arithmetic\_factor* **’\*’** *arithmetic\_unary |  
arithmetic\_factor* **‘/’** *arithmetic\_unary*

*arithmetic\_unary unit |* **‘-‘** *arithmetic\_unary |* **‘(‘** *arithmetic\_expression* **‘)’**

*unit* **‘IDENTIFIER’** *| integer\_value | bool\_value | function\_invocation*

*integer\_value* **‘DECIMAL’** *|* **‘OCTAL’** *|* **‘HEX’** *|* **‘BINARY’**

*bool\_value* **‘true’** *|* **‘false’**

*primitive\_type* **‘bool’** *|* **‘int’** *|* **‘char’** *|* **‘pointer’** *|* **‘void’**

*device\_type* **‘LightActuator’** *|* **‘ServoActuator’** *|* **’SoundSensor’** *|* **‘LightSensor’** *|* **‘DistanceSensor’**| **‘TemperatureSensor’**