Christian Lang

Neumont university  143 South Main Salt Lake City, UT 84111

Al syntax

# Arduino Language Syntax

## 1 Introduction

This is a syntax guide to the Arduino Language (AL). There are several aspects to the Arduino Language that are different than the conventional programming language: the core being the language is centered on reading like English and is loosely-typed. AL is an imperative and procedural programming language. The intention of this language is to educate people who desire to learn programming. The ease and flexibility of the language allows for a dynamic experience along with a sense of learning.

# 2 Reserved Word List

1. and: the conjunction of two statements in a Boolean expression. Ex: if x == 3 and y == 4
2. array: a collection of objects in AL. Ex: let int y = array[5] (y is a collection of 5 ints)
3. as: operator that allows you to perform certain types of conversions between compatible reference types.
4. begin:  the keyword associated with beginning a block of code
5. break:  used to resume program execution at the statement immediately following the current enclosing block or statement
6. case: the case keyword is used to create individual cases in a switch statement
7. constantly: for defining a variable to be constant and maintain the same value
8. continue:  used to resume program execution at the end of the current loop body
9. do:  used in conjunction with a while to create a do-while loop, which executes a block of statements associated with the loop and then tests a Boolean expression associated with the while
10. else:  used in conjunction with if to create an if-else statement which tests a Boolean expression
11. end:  the key word associated to ending a block of code
12. exit: the keyword used to completely exit the program
13. false: the keyword used to say if something is not true
14. file: the keyword used to access a file that is found on the hard drive
15. finally: used to define a block of statements for a block defined previously by the try keyword
16. function: defines a method in the method section before the main execution block
17. if: the if keyword is used to create an if statement, which tests a Boolean expression
18. in: the keyword used to compare an object and to see if it is ‘in’ a collection

1. is: they keyword used in conjunction with in to form the ‘is in’ statement
2. null: the keyword used to assign a variable nothing.
3. not: literal not, used in Boolean expressions
4. or: the keyword used in Boolean expressions, (booleanExpression1) or (booleanExpression2)
5. program: the keyword used to define a the program name
6. set: the keyword used to define a set of objects
7. switch: the keyword used to define a switch statement
8. this: the keyword used to define ‘this’ program
9. true: the keyword used to mark a Boolean as true.
10. unimplemented: the keyword that marks a method as unimplemented
11. while: the keyword used to create a loop, or put at the end of a do statement to evaluate a Boolean expression

# 3 Basic Structure

AL has a very basic structure so that there isn’t an issue with jumping around class structures and trying to find everything you need. All methods and variables will be in one file and will follow a basic structure as shown:

:alproject"

this program "AL\_Structure"

/\*

\* Comments look like this

\* This is where variables go

\*/

:variables:

Let #*variableName* = (string or number or whatever is needed)

/\*

\* This is where methods go

\*/

:methods:

method *methodName*(list of parameters, or none)

/\*

\*This is where the program will begin and end

\*/

:start program:

:end program:

All AL projects must have the header “:alproject:”. If the file does not have the specified header, then the file will be rejected and will not compile the code. The user is allowed no not have any variables and not have any methods at all if they chose to.