**Naming Conventions**

**Files Naming**

File names should be a set of words in UpperCamelCase (with the first letter of every word capitalised). It is important to avoid acronyms and abbreviations (unless the abbreviation is much more widely used than the long form, such as PID).

E.G.

PidControl.c

PidControl.h

**Functions Naming**

Function names should be a set of words in UpperCamelCase (with the first letter of every word capitalised). It is important to avoid acronyms and abbreviations (unless the abbreviation is much more widely used than the long form, such as PID).

E.G.

GetAdcData();

**Variables Naming**

All sort of variables (locals and globals) should be written in lowerCamelCase. Variable names should not start with special characters. Variable names should be short yet meaningful. The choice of a variable name should be mnemonic, that is, designed to indicate to the casual observer the intent of its use. One-character variable names should be avoided except for temporary "throwaway" variables. Common names for temporary variables are i, j, k, m, and n for integers; c, d, and e for characters.

E. G.

uint8\_t adcData;

**Constants Naming**

Constants should be written in UpperCamelCase and defined within a enum data type.

E.G.

enum DutyCycleBounds

{

DutyCycleMaxValue = 255,

DutyCycleMinValue = 0

};

**Macros**

Macros should only be used in header files (\*.h). The naming convention for this purpose should be as follow (upper case with “\_” between words):

#ifndef MACRO\_NAME\_H

#define MACRO\_NAME\_H

.

.

.

#endif

**Data Types**

**The following data types should be used in the software:**

int

char

uint8\_t

uint16\_t

uint32\_t

**Project Structure**

The developers shall implement each module or function through a pair of files (.c and .h). Also, some files will be created to handle the threads. Each of these shall have a clear and descriptive name following the naming conventions.