Variable and Function Naming Convention for C Programming

Description: Document providing Guidelines for Variable and Function Naming Convention for C Programming	**Document Name:** Variable and Function Naming Convention for C Programming

REVISION HISTORY

S. NO	REV. NO	DATE	CHANGE DESCRIPTION	Revised By
1	1.0	05-06-2020	First Version	Kiran A R

PREPARED BY	VERIFIED BY	APPROVED BY
Kiran A R	Pavan B, Sharath K	Kiran A R

Contents

1 Introduction

1.1 Purpose

Purpose of this document is to specify the naming convention for all the variables and functions defined in the firmware written by any developer at iTriangle Infotech. The Purpose is to enhance readability and reviewability of the code. The Coding guidelines related to MISRA C or any Safety requirements are defined in a separate document called "iTriangle Coding Guidelines"

1.2 Scope

Define Guide lines and rules for

- Global Variable Definition and Declaration
- Local Variable Definition and Declaration
- Function Definition and Declaration
- Type Definition and Declaration
- Structure and Union Definition and Declaration
- Macro Definition and Declaration

1.3 Intended Audience of the document

- Firmware Developer
- Firmware Architect
- Code Reviewers
- Unit Tester and Integration tester

1.4 Acronyms & Definitions

Table 1: Acronyms

2 Type String Table

2.1 Basic Standard data types

Type	Type String	Example	Remarks
integer	i	iCounter	
unsigned Integer	ui	uiCounter	
char	ch	chCounter	
Unsinged char	uch	uchCounter	
float	f	fCounter	
Structure	st	stRecord	
Typedefined structure	tst	tstRecord	
union	un	unRecord	
Typedefined union	tun	tunRecord	
double	d	dCounter	
double double	dd	ddCounter	
enum	en	enRecord	
Typedef enum	ten	tenRecord	
Bool	bo	boFLag	

2.2 Type Redefined in the project

Type	Type String	Example	Remarks
U32	u32	u32Counter	
U16	u16	u16Counter	
U8	u8	u8Counter	
U64	u64	u64Counter	
S32	s32	s32Counter	
S16	s16	s16Counter	
S8	s8	s8Counter	
S64	s64	s64Counter	
word	w	wCounter	
double word	dw	dwCounter	
byte	b	bCounter	

2.3 Pointers

Below Section defines the type string for pointer variables

2.3.1 pointer to Basic Standard data types

Type	Type String	Example	Remarks
Integer	pi	piCounter	
unsigned Integer	pui	puiCounter	
Char	pch	pchCounter	

Unsinged char	puch	puchCounter	
Float	pf	pfCounter	
Structure	pst	pstRecord	
Typedefined structure	ptst	ptstRecord	
Union	pun	punRecord	
Typedefined union	ptun	ptunRecord	
Double	pd	pdCounter	
double double	pdd	pddCounter	
Enum	pen	penRecord	
Typedef enum	pten	ptenRecord	
Bool	pbo	pboFLag	

2.3.2 pointer to Type Redefined in the project

Type	Type String	Example	Remarks
U32	pu32	pu32Counter	
U16	pu16	pu16Counter	
U8	pu8	pu8Counter	
U64	pu64	pu64Counter	
S32	ps32	ps32Counter	
S16	ps16	ps16Counter	
S8	ps8	ps8Counter	
S64	ps64	ps64Counter	
Word	pw	pwCounter	
double word	pdw	dwCounter	
Byte	pb	pbCounter	

2.4 Arrays

Below Section defines the type string for array of variables

2.4.1 Array of Basic Standard data types

Type	Type String	Example	Remarks
Integer	ai	aiCounter	
unsigned Integer	aui	auiCounter	
Char	ach	achCounter	

Unsinged char	auch	auchCounter	
Float	af	afCounter	
Structure	ast	astRecord	
Typedefined structure	atst	atstRecord	
Union	aun	aunRecord	
Typedefined union	atun	atunRecord	
Double	ad	adCounter	
double double	add	addCounter	
Enum	aen	aenRecord	
Typedef enum	aten	atenRecord	
Bool	abo	aboFLag	

2.4.2 Array of Type Redefined in the project

Type	Type String	Example	Remarks
U32	au32	au32Counter	
U16	au16	au16Counter	
U8	au8	au8Counter	
U64	au64	au64Counter	
S32	as32	as32Counter	
S16	as16	as16Counter	
S8	as8	as8Counter	
S64	as64	as64Counter	
Word	aw	awCounter	
double word	adw	awCounter	
Byte	ab	abCounter	
Void	v	vPtr	

Note:

When an array of pointer is defined, the type String to start with "ap" followed by the Repective Data String. For example array of Pointer to Integer would have a type string as "api".

If any data type which cannot be covered with the above Type string tables, please contact the Author or the Maintainer of the document.

3 Global Variable Definition and Declaration

3.1 General Structure of the Global Variable Name

The General Structure for Global Variable Name shall be as per below structure

XXX_TypeVariableName

Section	Description	Length	Remarks
XXX	1 to 5 Letter of the Module in which the Global Variable in declared. As short for of module Name. Should be always in CAPS	3 to 5	For example Q or QUE for QUEUE.C module DATRC for Data recorder Module
Type	Indicated the type of the variable. Should be always in Small Letter	NA	As per type string table
VariableName	Actual Name of the variable. Should always start with Capital Letter	Should be as per the coding guidelines	Should be meaningful and easily understandable

3.1.1 Examples:

For A Global Variable in the module" command Handler" with integer type for counting purpose will be defined as

int CMD_iCounter ;

4 Local Variable Definition and Declaration

4.1 General Structure of the Local Variable Name

The General Structure for Local Variable Name shall be as per below structure. The only difference b/w Global Variable Naming and Local Variable Naming is that, There is no Module Name attached to it.

TypeVariableName

Section	Description	Length	Remarks
Type	Indicated the type of the variable.	NA	As per type string table

	Should be always in Small Letter		
VariableName	Actual Name of the variable. Should always start with Capital Letter	Should be as per the coding guidelines	Should be meaningful and easily understandable

4.1.1 Examples

For A local Variable in any function with integer type for counting purpose will be defined as

Int iCounter ;

5 Function Name Definition

The General Structure for Function Name shall be as per below structure

XXX_returntypeFucntionName(parameter list)

Section	Description	Length	Remarks
XXX	1 to 5 Letter of the Module in which the function is declared. As short for of module Name. Should be always in CAPS	3 to 5	For example Q or QUE for QUEUE.C module DATRC for Data recorder Module
retuntype	Indicated return type of the function. Should be always in Letter	NA	As per type string table
FucntionName	Actual Name of the variable. Should always start with Capital Letter	Should be as per the coding guidelines	Should be meaningful and easily understandable
parameter list	Function parameter list	No Special Rule. As per C program Standards	

5.1.1 Examples:

For A Function to Set IMEI Number in ” command Handler” Module with return type integer will be defined as

int CMD_iGetIMEI(void);

For A Function to Module Initialization in ” command Handler” Module with return type void will be defined as

void CMD_vFirstInit(void);

6 Advantages of The Naming Convention guidelines.

Some of the key advantage of adopting naming Convention for variables and function are

- Improves the readability of the code to a large extent and
- Uniformity in the code irrespective of the developer who develops it
- Reduces a lot of time in code review with respect to datatypes as reviewer spends lot of time is searching back and forth through the code to know the data types while reviewing assignment, copy/move, or any operation. Just looking at the name itself gives us the details of the data type of the variables and functions.
- Ensures meaning full name declaration

7 Applicability

These naming rules are applicable for any code written by iTriangle. Its mandatory to adopt the standard for all the code written by the company and Optional on any third party code or SDK or any Library which is not developed or written by iTriangle Firmware team.

8 Reference

For detailed understanding with Examples, please review the code of the following Modules in the existing Bharat101 or TS101 Basic or TS101 Advance code bases.

- QUEUE
- DATREC