Projekt: **Hovering Steward**Schuljahr: **2015/16**

Projektleiter: **Markus Kaiser**

Codingrules

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Bearbeiter** | **Datum** | **Änderung** |
| 0.1 | Markus Kaiser | 04.09.2015 | Erstmaliges Erstellen |
| 1.0 | Lucas, Chrisy | 13.11.2015 | Erstmalige Codingrules |

Tipps vom Fleck:

Funktionelle filebez. auch für funktionen

funktionelle variablenbez. -> variablentyp

Groß kleinschreibung

struktur, union, enumnamen

Bezeichnungen

Files

* Anwendungsort, ausgenommen Hauptplatine
* Bezeichner mit Rückschluss auf Fileinhalt
* Versionsnummer
* Wörter werden mit Underscores getrennt

Beispiel:

    spi\_com\_v1\_0.c

Funktionen

* Funktionskürzel “Fkt\_”
* Filebezeichner
* Erster Buchstabe jedes Wortes groß
* Keine Underscores, Ausnahme Funktionskürzel
* Bei Wertrückgabe kein Variablentypenkürzel

Beispiel:

    Filename: spi\_com\_v1\_0.c

    Funktionsname: Fkt\_SpiExchange8bit()

Variablen

* Variablentypekürzel als Prefix
* Variablenname
* Variable ist klein zu schreiben, Wörter werden mit underscores getrennt
* Initialisierungswert

Beispiel:

    unsigned char uc\_variable\_beispiel = 0;

Konstanten

* Konstantenname in Großbuchstaben
* Wörter werden mit Underscores getrennt
* Wert

Beispiel:

    #define VARIABLE\_KONSTANT 42

Groß- und Kleinschreibung

Programmstruktur

* Erklärung was im File “passiert”
* Inkludieren der Headerfiles
* Funktionen
  + Prolog, erklärung was die Funktion macht
  + Definition der Funktion
  + Lokale Variablen
  + Code der Funktion
  + Allfälliges Return
* Mainfunktion (Nur im Hauptprogramm)

Usage and Operating Instructions, if applicable for program files only

Header File Includes, in this sequence:

#include <stdio.h> (or <stdlib.h>)

#include <other system headers>

#include “user header files”

Defines and Typedefs that apply to the file as a whole, including:

enums

typedefs

constant macro defines

function macro defines

External Data Declarations used by this file

extern declarations of variables defined in other files

non-static external definitions used in this file (and optionally

in others if they are declared in those files using extern)

static external definitions used only in this file

Functions

    Function prolog

Name of the function

was tut die Funktion?

Return value of the function

Function argument declarations

External variable declarations

Internal variable declarations

Automatic internal variable definitions

Static internal variable definitions

Code -> Statements

Return statement

Kommentare

ReadMe File

Filevorspann

Funktionsprolog

Im Funktionsprolog sind die Funktionen der Funktion sowie die verwendeten Variablen auszulisten.

Beispiel:

/\*-----------------------------------------------------------------\*/

/\* Hier wir ein Beispiel für einen Prolog aufgelistet.             \*/

/\* ui\_beispiel\_var Dient als Beispiel der Variablenauflistung.       \*/

/\*-----------------------------------------------------------------\*/

Funktionsvorspann

Funktionskommentare

Typenkürzel