

```

/*
 * Copyright (c) 2013 - 2014, Freescale Semiconductor, Inc.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without modification,
 * are permitted provided that the following conditions are met:
 *
 * o Redistributions of source code must retain the above copyright notice, this list
 *   of conditions and the following disclaimer.
 *
 * o Redistributions in binary form must reproduce the above copyright notice, this
 *   list of conditions and the following disclaimer in the documentation and/or
 *   other materials provided with the distribution.
 *
 * o Neither the name of Freescale Semiconductor, Inc. nor the names of its
 *   contributors may be used to endorse or promote products derived from this
 *   software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND
 * ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
 * DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR
 * ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES;
 * LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON
 * ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
 * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
 * SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */

```

```

#ifndef __print_scan_h__
#define __print_scan_h__

```

```

#include <stdio.h>
#include <stdarg.h>
#include <stdint.h>
#include <stdbool.h>
#include <string.h>

```

```

// #define PRINTF_FLOAT_ENABLE 1
// #define PRINT_MAX_COUNT 1
// #define SCANF_FLOAT_ENABLE 1

```

```

#ifndef HUGE_VAL
#define HUGE_VAL (99.e99) // wrong value
#endif

```

```

typedef int (*PUTCHAR_FUNC)(int a, void *b);

```

```

/*!
 * @brief This function outputs its parameters according to a formatted string.
 *
 * @note I/O is performed by calling given function pointer using following
 * (*func_ptr)(c,farg);
 *
 * @param[in] farg Argument to func_ptr.
 * @param[in] func_ptr Function to put character out.
 * @param[in] max_count Maximum character count for snprintf and vsnprintf.
 * Default value is 0 (unlimited size).
 * @param[in] fmt_ptr Format string for printf.
 * @param[in] args_ptr Arguments to printf.
 *
 * @return Number of characters
 * @return EOF (End Of File found.)
 */

```

```

int _doprint(void *farg, PUTCHAR_FUNC func_ptr, int max_count, char *fmt, va_list ap);

```

```

/*!

```

```

* @brief Writes the character into the string located by the string pointer and
* updates the string pointer.
*
* @param[in] c The character to put into the string.
* @param[in, out] input_string This is an updated pointer to a string pointer.
*
* @return Character written into string.
*/
int _sputc(int c, void * input_string);

/*!
* @brief Converts an input line of ASCII characters based upon a provided
* string format.
*
* @param[in] line_ptr The input line of ASCII data.
* @param[in] format Format first points to the format string.
* @param[in] args_ptr The list of parameters.
*
* @return Number of input items converted and assigned.
* @return IO_EOF - When line_ptr is empty string "".
*/
int scan_prv(const char *line_ptr, char *format, va_list args_ptr);

#endif

```