Wiki at erau.us

Workshop 1

The standard `printf` function cannot print an integer in the binary form. In this workshop, we write C functions that can be used to print an uint16_t integer in two different binary forms, the compact one and the verbose one, as detailed later.

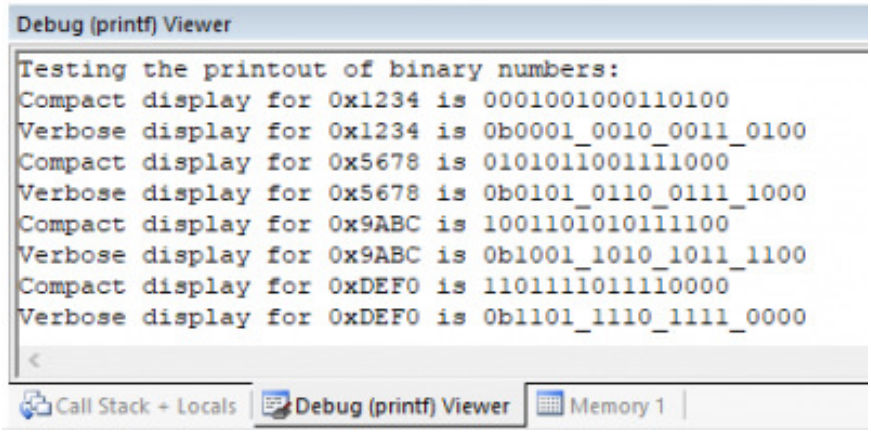
Note that we don't need to use the Discovery Board for this workshop. The simulator with Keil is used for simplicity. Use the following code snippet as a starting point:

```
#include <stdio.h>
#include <stdint.h>
#include <stdbool.h>

char str_c[30], str_v[30];

void uint16_2_binary_str(uint16_t num, char *cPtr);
void print_str_c(uint16_t num, char *cPtr);
void print_str_v(uint16_t num, char *cPtr_c, char *cPtr_v);


int main(void) {
    uint16_t test_num[] = {0x1234, 0x5678, 0x9ABC, 0xDEF0};
    int N = sizeof(test_num) / sizeof(test_num[0]);

    printf("Testing the printout of binary numbers:\n");
    for (int i = 0; i < N; i++) {
        printf("Compact display for 0x%X is ", test_num[i]);
        print_str_c(test_num[i], str_c);
        printf("\n");
        printf("Verbose display for 0x%X is ", test_num[i]);
        print_str_v(test_num[i], str_c, str_v);
        printf("\n");
    }

    while (1);
}
```

Note that we have defined two global strings of 30 chars each:

- `str_c` is used to hold the compact version of the binary expression of a number using ASCII '0's and '1's. For example, if the integer is 0x0039, `str_c` should be `0000000000111001`. Note that you need to end the string with number 0 '\0'.
- `str_v` is used to hold the verbose version of the binary expression of the number. For example, the verbose version of the above compact expression is `0b0000_0000_0011_1001`.

The running result should be similar to



Below are the programming requirements of the assignment.

- (40 points) Write a function named `uint16_2_binary_str` which converts `num` to binary and save the results to `cPtr`.
- (10 points) Write a function named `print_str_c` which can be used to print the compact form of the binary string. Note that you need to call `uint16_2_binary_str` in this function to perform the conversion and then print. Note that you cannot print the newline "\n" since we may join strings in the same line. This is the reason why we have used "printf("\n");" after calling this function.
- (40 points) Write a function named `print_str_v` which can be used to print the results in the verbose form. In this function, you need to call `uint16_2_binary_str` as well. You then need to build `cPtr_v` from `cPtr_c`. Finally, you print out `cPtr_v` in a manner similar to `print_str_c`.

Below are the submission requirements of the assignment (10 points).

- You are supposed to do the assignment in teams of two members.
- You need to submit a renamed version of the `main.c` file of the project. Use this convention: `cec320_sec_x_ws1_lastname1_firstname1__lastname2_firstname2.c`, where `x` is your section number.
- You need also submit a pdf file which contains the screenshots of your C functions and the screenshot of the running results. The running results should be like the one given above but with the names of you and your team member displayed clearly. Use the same naming convention as above but with a suffix of `pdf` for this file.
