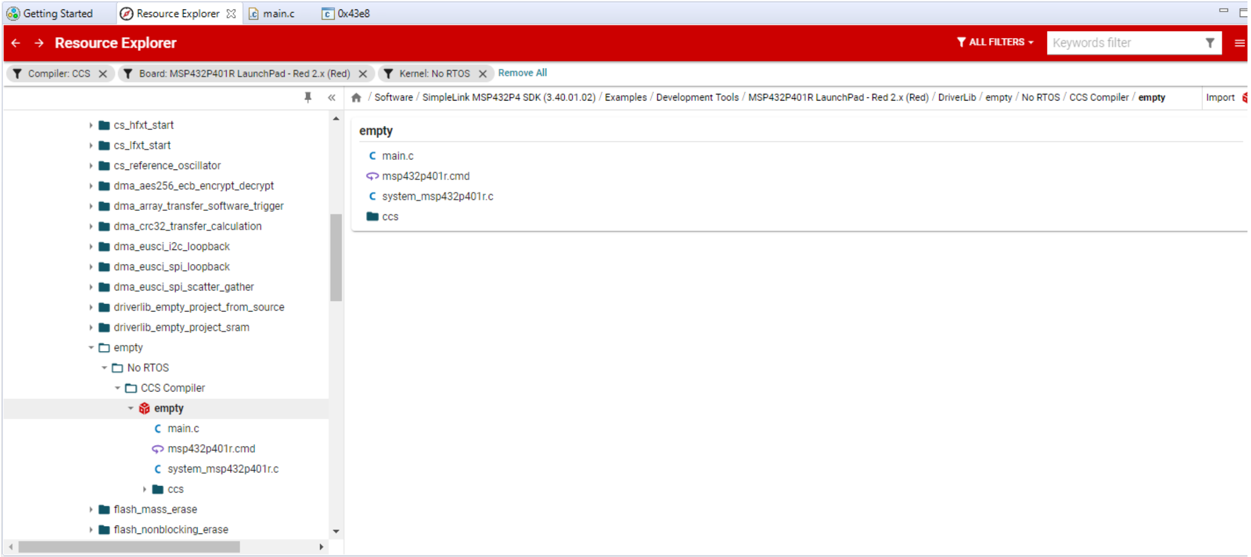
Assessment for Introduction to C and Variables

Introduction

In the embedded world one of the ways you will be able to see what is happening To expand your capability to use the MSP432P401R for your introduction to the C programming language you will find it useful to interact with its processing power more effectively. One of the most effective way to do this is through a serial port, which is a hardware interface that can both send and receive data to the MSP432P401R.

The Code Composer environment provides this through the USB cable interface. This is supported by some high-level functionality that is a part of the IDE Now you can add these to your program. To get started open an empty project:



Compiler: CCS
Board: MSP432P401R LaunchPad - Red 2.x (Red)
Kernel: No RTOS

/ Software / SimpleLink MSP432P4 SDK (3.40.01.02) / Examples / Development Tools / MSP432P401R LaunchPad - Red 2.x (Red) / DriverLib / empty / No RTOS / CCS Compiler / **empty**

```
cs_hfxt_start
cs_lfxt_start
cs_reference_oscillator
dma_aes256_ecb_encrypt_decrypt
dma_array_transfer_software_trigger
dma_crc32_transfer_calculation
dma_eusci_i2c_loopback
dma_eusci_spi_loopback
dma_eusci_spi_scatter_gather
driverlib_empty_project_from_source
driverlib_empty_project_sram
empty
  No RTOS
    CCS Compiler
      empty
        main.c
        msp432p401r.cmd
        system_msp432p401r.c
        ccs
flash_mass_erase
flash_nonblocking_erase
```

empty

```
main.c
msp432p401r.cmd
system_msp432p401r.c
ccs
```

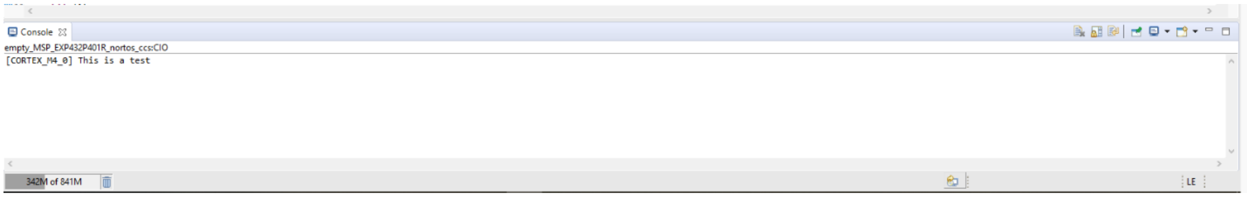
Now let’s add some basic input/output functionality.

Output

The Code Composer IDE supports the printf() command. This command will allow you to print out information from the program running on your MSP432P401R. So add a simple printf() function to your empty project:

```
/* DriverLib Includes */
#include <ti/devices/msp432p4xx/driverlib/driverlib.h>

/* Standard Includes */
#include <stdint.h>
#include <stdbool.h>

int main(void)
{
    /* Stop Watchdog  */

    MAP_WDT_A_holdTimer();

    printf("This is a test\n");

    while(1)
    {

    }
}
```

Now Build and then Debug As -> Code Composer Debug Session. When you hit resume you should see this in the console window:



```
empty_MSP_EXP432P401R_nortos_ccs:CIO
[CORTEX_M4_0] This is a test
```

Now you have a simple way of printing out information from your MSP432P401R.

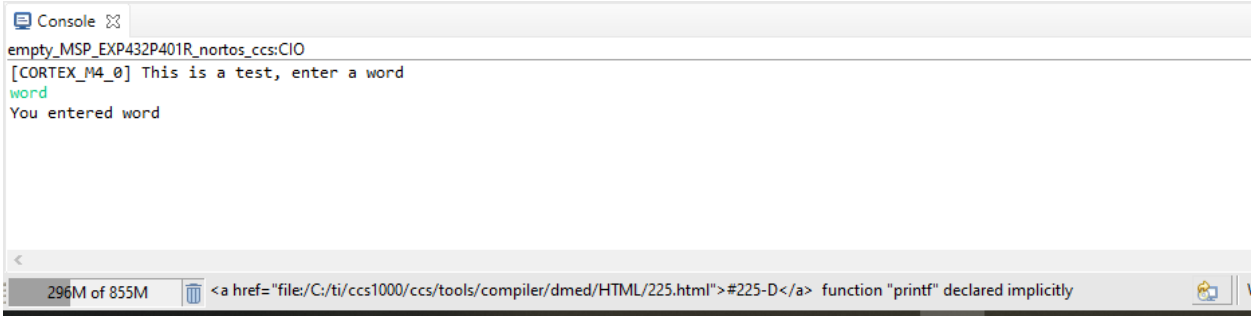
Input

Printing information out is very useful, but you may also want to enter information into the program. You can use the scanf() function to do this. We won't document the specifics of how scanf() works (see https://www.tutorialspoint.com/c_standard_library/c_function_scanf.htm).

Here is some example code:

```
/* DriverLib Includes */
#include <ti/devices/msp432p4xx/driverlib/driverlib.h>

/* Standard Includes */
#include <stdint.h>
#include <stdbool.h>

int main(void)
{
    char input[10];
    /* Stop Watchdog  */

    MAP_WDT_A_holdTimer();

    printf("This is a test, enter a word \n");

    scanf("%s", input);

    printf("You entered %s\n", input);

    while(1)
    {
        }
}
```

If you Build this program, then DebugAs, hit Resume, and you will see the prompt in the Console Window. Select the window, enter a word (this will be displayed as you type in Green), and you will see the result:



```
empty_MSP_EXP432P401R_nortos_ccs:CIO
[CORTEX_M4_0] This is a test, enter a word
word
You entered word
```

```
296M of 855M   <a href="file:/C:/ti/ccs1000/ccs/tools/compiler/dmed/HTML/225.html">#225-D</a>  function "printf" declared implicitly
```

Now you have the capability to enter data into a program, and display results.

Task

The follow is to be completed to verify your learning. You should start with the project shown in the example above. In the process of completing the assignments you will create a set of variables of varying types.

1. Create a program that reads standard input and sends to standard output the hexadecimal values of the characters input, each on its own line. (You’ll need to create an ASCII conversion table.)

   Example input/output:

   ```
   [CORTEX_M4_0] Enter a string: gtfj
   67
   74
   66
   6A
   ```

2. Write a program that converts a string of digits into its equivalent integer value;

   Example input/output:

   ```
   [CORTEX_M4_0] Enter an integer value: 1265
   1265
   ```

Hint: In order to walk through a string of characters, try this:

```
char input_string[100];

int i;

printf(“Enter a string: “);

scanf(“%s”, input_string);

for (i = 0; input_string[i]; i++)   // This will stop when the input_string character at the end of the valid string is 0
```

Compile, Build, and debug your programs.

Take a Screen shot showing the state of the variables with valid data in them.