**Purpose**

The purpose of this lab is to become familiar with the Code Composer Studio software as well as gain a basic understanding of the MSP432P401R board and some of its functionality.

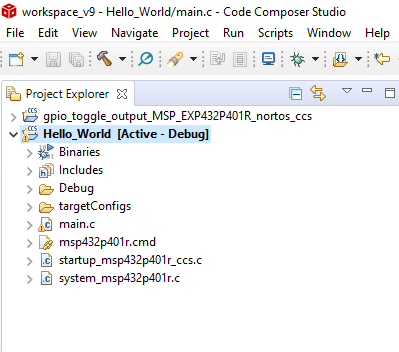
**Exercise 1**

There were no issues with setting up and installing the Code Composer Studio.

**Exercise 2**

The only issue encountered during setup was finding the correct project within the subfolders after the example files were installed. I followed the key titles as listed in the assignment instructions, however the folders were not in order as the tags listed in the exercise.

**Exercise 3**



**Exercise 4**

|  |
| --- |
| **main.c** |
| #include "msp.h"  void main(void)  {  WDT\_A->CTL = WDT\_A\_CTL\_PW | WDT\_A\_CTL\_HOLD; // stop watchdog timer  uint32\_t data;  uint32\_t offset;  int i;  for (i = 0; i < 17; i = i + 1)  {  offset = (uint32\_t)(i \* 4);  uint32\_t\* addr\_ptr = (uint32\_t\*)(0x00201000+offset);  data = \*addr\_ptr;  printf("Data at address %08x: %08x\n", addr\_ptr, data);  }  } |

|  |  |  |  |
| --- | --- | --- | --- |
| **Description** | | **Address** | **Value** |
|  | TLV Checksum | 00201000h | 907FB435h |
| Info Block | Device Info Tag | 00201004h | 0000000Bh |
| Device Info Length | 00201008h | 00000004h |
| Device ID | 0020100Ch | 0000A000h |
| Hardware Revision | 00201010h | 00000044h |
| Boot-Code Revision | 00201014h | 00420044h |
| ROM Driver Library Revision | 00201018h | 03000010h |
| Die Record | Die Record Tag | 0020101Ch | 0000000Ch |
| Die Record Length | 00201020h | 00000008h |
| Die X Position | 00201024h | 00000023h |
| Die Y Position | 00201028h | 00000009h |
| Wafer ID | 0020102Ch | 00000004h |
| Lot ID | 00201030h | 00011412h |
| Reserved | 00201034h | FFFFFC42h |
| Reserved | 00201038h | 00000000h |
| Reserved | 0020103Ch | 00000000h |
| Test Results | 00201040h | 00474345h |

**Exercise 5**

|  |
| --- |
| **gpio\_toggle\_output.c** |
| /\* DriverLib Includes \*/  #include <ti/devices/msp432p4xx/driverlib/driverlib.h>  /\* Standard Includes \*/  #include <stdint.h>  #include <stdbool.h>  //![Simple GPIO Config]  int main(void)  {  volatile uint32\_t ii;  /\* Halting the Watchdog \*/  MAP\_WDT\_A\_holdTimer();  /\* Configuring P1.0 as output \*/  MAP\_GPIO\_setAsOutputPin(GPIO\_PORT\_P1, GPIO\_PIN0);  MAP\_Timer32\_initModule(TIMER32\_0\_BASE, TIMER32\_PRESCALER\_1,  TIMER32\_32BIT, TIMER32\_FREE\_RUN\_MODE);  MAP\_Timer32\_startTimer(TIMER32\_0\_BASE, 0);  uint32\_t clk\_freq = MAP\_CS\_getMCLK();  printf("%u\n", clk\_freq);  while (1)  {  uint32\_t t0 = MAP\_Timer32\_getValue(TIMER32\_0\_BASE);  /\* Delay Loop \*/  for(ii=0;ii<500000;ii++)  {  }  MAP\_GPIO\_toggleOutputOnPin(GPIO\_PORT\_P1, GPIO\_PIN0);  uint32\_t t1 = MAP\_Timer32\_getValue(TIMER32\_0\_BASE);  uint32\_t num\_cycles = t0 - t1;  printf("Cycles: %lu\nTime: %f ms\n", num\_cycles, ((double)num\_cycles/(double)clk\_freq)\*1000);  }  } |

By obtaining the clock frequency (3 MHz), the clock period in milliseconds can be obtained (0.0003333 ms). The countdown timer decrements after each clock cycle and so the total number of clock cycles required for a single iteration of the while loop can be obtained by taking the difference between the initial count and the final count when the loop finishes, denoted by *t1* and *t0*.. The total number of clock cycles (*num\_cycles*) is divided by the clock frequency to give the time in seconds and then further converted (by multiplying by a factor of 1000) to give the time in milliseconds.

The number of cycles is printed along with the time elapsed in milliseconds so that the calculation can be checked. When the value of *ii* is set to 500,000, the average number of clock cycles was 7000066. Dividing this number by the clock frequency and converted to milliseconds yields an average time for a single iteration as 2333 ms. This value was verified using an external calculator.

**Appendix**

**Exercise 2**

|  |
| --- |
| **gpio\_toggle\_output.c** |
| /\* DriverLib Includes \*/  #include <ti/devices/msp432p4xx/driverlib/driverlib.h>  /\* Standard Includes \*/  #include <stdint.h>  #include <stdbool.h>  //![Simple GPIO Config]  int main(void)  {  volatile uint32\_t ii;  /\* Halting the Watchdog \*/  MAP\_WDT\_A\_holdTimer();  /\* Configuring P1.0 as output \*/  MAP\_GPIO\_setAsOutputPin(GPIO\_PORT\_P1, GPIO\_PIN0);  printf("%u\n", MAP\_CS\_getMCLK());  while (1)  {  uint32\_t t0 = MAP\_Timer32\_getValue(TIMER32\_0\_BASE);  /\* Delay Loop \*/  for(ii=0;ii<50000;ii++)  {  }  MAP\_GPIO\_toggleOutputOnPin(GPIO\_PORT\_P1, GPIO\_PIN0);  }  } |

**Exercise 3**

|  |
| --- |
| **main.c** |
| #include "msp.h"  //#include <stdio.h>  /\*\*  \* main.c  \*/  void main(void)  {  //WDT\_A->CTL = WDT\_A\_CTL\_PW | WDT\_A\_CTL\_HOLD; // stop watchdog timer  printf("Hello World!!!\n");  } |