**ENCM 511 ASSIGNMENT #1**

**Author**: Kyle Derby MacInnis

**UCID**: 10053959

**Email**: kderbyma@ucalgary.ca

**Date**: October 16, 2014

**Professor**: Dr. Smith

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**COMMENTS:**

* The Code Snippets are generally snippets. For more information, please refer to the zipped assignment files.
* The Screenshots are one to a page.
* The Code Formatting was due to the CCES IDE.

**Screenshots:**

|  |  |  |
| --- | --- | --- |
| 1 |  |  |
| Figure 1 – The Coffee Pot has been added to the device, and the Base Address given. |
| 2 |  |  |
| Figure 2 – The First Coffee Pot has been Initialized and told to stay on – Waiting for Device to be Ready. |
| 3 |  |  |
| Figure 3 – The Device Ready Bit has been set, and the Coffee Pot is ready to begin making coffee. |
| 4 |  |  |
| Figure 4 – The Coffee Pot Power has been set on, and the Water is beginning to flow. |
| 5 |  |  |
| Figure 5 – Water is finished filling, water flow set to 0, and heat has been turned on, and the water is beginning to warm up. |
| 6 |  |  |
| Figure 6 - Water is beginning to bubble as temperature increases. |
| 7 |  |  |
| Figure 7 – The Water has finished heating up and the Coffee Pod has been initialized. |
| 8 |  |  |
| Figure 8 – The First Coffee Pot has finished (GUI Problem with Colour), and now the second pot is being prepared. |
|  |  |  |
| Figure 9 – Both Coffee Pots have finished and their pods have been initialized. (Issue with GUI Colour) |
| **COFFEEPOT\_MAIN.CPP** | | |
| **#include** "MyCoffeePotFunctions.h"  /\* Managed drivers and/or services include \*/  **#include** "../system/adi\_initialize.h"  **int main**(**void**) {  /\* Initialize managed drivers and/or services \*/  **adi\_initComponents();**  **printf**("Hello World from the Blackfin\n");  // Use mathematical paragraphs to separate ideas  **printf**("\n");  **printf**("About to start my coffee pot simulator\n");  **int** numCoffeePots = 2;  **int** whichDisplay = USE\_TEXT\_GUI | USE\_CCES\_GUI;  **Init\_CoffeePotSimulation**(numCoffeePots, (WHICHDISPLAY) whichDisplay);  COFFEEPOT\_DEVICE \*coffeePot1\_BaseAddress = (COFFEEPOT\_DEVICE \*)NULL;  COFFEEPOT\_DEVICE \*coffeePot2\_BaseAddress = (COFFEEPOT\_DEVICE \*)NULL;  coffeePot1\_BaseAddress = **Add\_CoffeePotToSystem\_PlugAndPlay**(COFFEEPOT1, "WDS");  **Do\_Simulation\_CPP**(coffeePot1\_BaseAddress);  coffeePot2\_BaseAddress = **Add\_CoffeePotToSystem\_PlugAndPlay**(COFFEEPOT2, "KDM");  **Do\_Simulation\_CPP**(coffeePot2\_BaseAddress);  }  **void** **USE\_CCES\_GUI\_Delay**(void) {  **int** delay = 0xFFFF;  **for** ( **int** count = 0; count < delay; count++) { // 0x1FFFFFF Seemed to work abeit slowly  count = count + 1;  }  } | | |
| **TURNONWATER\_ASM.ASM** | | |
| // LOCAL VARIABLE DECLARATIONS  **.section** data1  // PROGRAM INSTRUCTIONS  **.section** program  //void TurnOnWater\_ASM(COFFEEPOT\_DEVICE \*coffeePt\_BaseAddress, unsigned char waterInputSpeed) {  **.global** \_TurnOnWater\_ASM;  // INPUT PARAMETERS  **#define** PAR1\_R0 R0  **#define** PAR2\_R1 R1  // TEMPORARY REGISTERS  **#define** TEMP\_R2 R2  **#define** TEMP\_R3 R3  // COFFEEPOT\_DEVICE \* BaseAddress  **#define** COFFEEPOT\_DEVICE\_P0 P0  // CONTROL REGISTER OFFSETS  **#define** CONTROLREGISTER\_OFFSET 0  **#define** WATERINFLOWREGISTER\_OFFSET 7  **#define** HEATERREGISTER\_OFFSET 6  **#define** HEATERBOOST\_OFFSET 8  **#define** LED\_OFFSET 12  // CONTROL BITMASKS  **#define** INITPOWERED (1 << 0)  **#define** LEDPOWER (1 << 1)  **#define** WATERPOWER (1 << 2)  **#define** HEATERPOWER (1 << 3)  **#define** DEVICEREADY (1 << 4)  // LED BITMASKS  **#define** LED1 (1 << (LED\_OFFSET + 0))  **#define** LED2 (1 << (LED\_OFFSET + 1))  **#define** LED3 (1 << (LED\_OFFSET + 2))  **#define** LED4 (1 << (LED\_OFFSET + 3))  **#define** LED\_LED\_ON LED2  **#define** LED\_HEAT\_ON LED3  **#define** LED\_WATER\_ON LED4  **\_TurnOnWater\_ASM:**  LINK 20;  // SELECT COFFEE POT BASE ADDRESS  COFFEEPOT\_DEVICE\_P0 = PAR1\_R0;  // SET WATER FLOW RATE  B[COFFEEPOT\_DEVICE\_P0 + WATERINFLOWREGISTER\_OFFSET] = PAR2\_R1;    // TURN ON POWER AND WATER (AND LEDS)  TEMP\_R2 = W[COFFEEPOT\_DEVICE\_P0 + CONTROLREGISTER\_OFFSET];  TEMP\_R3 = WATERPOWER;  TEMP\_R2 = TEMP\_R2 | TEMP\_R3;  TEMP\_R3 = LEDPOWER;  TEMP\_R2 = TEMP\_R2 | TEMP\_R3;  R3.L = lo(LED\_WATER\_ON); R3.H = hi(LED\_WATER\_ON);  TEMP\_R2 = TEMP\_R2 | TEMP\_R3;  R3.L = lo(LED\_LED\_ON); R3.H = hi(LED\_LED\_ON);  TEMP\_R2 = TEMP\_R2 | TEMP\_R3;  W[COFFEEPOT\_DEVICE\_P0 + CONTROLREGISTER\_OFFSET] = TEMP\_R2;  // UPDATE DISPLAY  **.extern** \_\_Z23UpdateSimulationDisplayv;  CALL \_\_Z23UpdateSimulationDisplayv;  CALL \_\_Z23UpdateSimulationDisplayv;  UNLINK;  **\_TurnOnWater\_ASM.END:**  RTS; | | |

|  |
| --- |
| **TURNONHEAT\_ASM.ASM** |
| // LOCAL VARIABLE DECLARATIONS  **.section** data1  // PROGRAM INSTRUCTIONS  **.section** program  // void TurnOnHeat\_ASM(COFFEEPOT\_DEVICE \*coffeePot\_BaseAddress, unsigned char heaterLevel, unsigned char heaterBoost);  **.global** \_TurnOnHeat\_ASM;  // INPUT PARAMETERS  **#define** PAR1\_R0 R0  **#define** PAR2\_R1 R1  **#define** PAR3\_R2 R2  // TEMPORARY REGISTERS  **#define** TEMP\_R3 R3  **#define** TEMP\_R4 R4  // COFFEEPOT\_DEVICE \* BaseAddress  **#define** COFFEEPOT\_DEVICE\_P0 P0  // CONTROL REGISTER OFFSETS  **#define** CONTROLREGISTER\_OFFSET 0  **#define** WATERINFLOWREGISTER\_OFFSET 7  **#define** HEATERREGISTER\_OFFSET 6  **#define** HEATERBOOST\_OFFSET 8  **#define** LED\_OFFSET 12  // CONTROL BITMASKS  **#define** INITPOWERED (1 << 0)  **#define** LEDPOWER (1 << 1)  **#define** WATERPOWER (1 << 2)  **#define** HEATERPOWER (1 << 3)  **#define** DEVICEREADY (1 << 4)  // LED BITMASKS  **#define** LED1 (1 << (LED\_OFFSET + 0))  **#define** LED2 (1 << (LED\_OFFSET + 1))  **#define** LED3 (1 << (LED\_OFFSET + 2))  **#define** LED4 (1 << (LED\_OFFSET + 3))  **#define** LED\_LED\_ON LED2  **#define** LED\_HEAT\_ON LED3  **#define** LED\_WATER\_ON LED4  **\_TurnOnHeat\_ASM:**  LINK 20;  // SET COFFEE POT BASE ADDRESS  COFFEEPOT\_DEVICE\_P0 = PAR1\_R0;    // TURN WATER OFF - \*\*DOES NOT WORK WITH GUI  //TEMP\_R4 = ~WATERPOWER;  //TEMP\_R3 = TEMP\_R3 & TEMP\_R4;  //R4.L = lo(~LED\_WATER\_ON); R4.H = hi(~LED\_WATER\_ON);  //TEMP\_R3 = TEMP\_R3 & TEMP\_R4;    // TURN WATER FLOW RATE DOWN TO ZERO - \*\*WORKS WITH GUI  TEMP\_R3 = 0;  B[COFFEEPOT\_DEVICE\_P0 + WATERINFLOWREGISTER\_OFFSET] = TEMP\_R3;    // SET HEATER AND BOOST LEVEL  B[COFFEEPOT\_DEVICE\_P0 + HEATERREGISTER\_OFFSET] = PAR2\_R1;  B[COFFEEPOT\_DEVICE\_P0 + HEATERBOOST\_OFFSET] = PAR3\_R2;    // TURN ON HEAT (AND LED)  TEMP\_R3 = W[COFFEEPOT\_DEVICE\_P0 + CONTROLREGISTER\_OFFSET];  TEMP\_R4 = HEATERPOWER;  TEMP\_R3 = TEMP\_R3 | TEMP\_R4;  R4.L = lo(LED\_HEAT\_ON); R4.H = hi(LED\_HEAT\_ON);  TEMP\_R3 = TEMP\_R3 | TEMP\_R4;  W[COFFEEPOT\_DEVICE\_P0 + CONTROLREGISTER\_OFFSET] = TEMP\_R3;  // UPDATE DISPLAY  **.extern** \_\_Z23UpdateSimulationDisplayv;  CALL \_\_Z23UpdateSimulationDisplayv;  CALL \_\_Z23UpdateSimulationDisplayv;  UNLINK;  **\_TurnOnHeat\_ASM.END:**  RTS; |

|  |
| --- |
| **TURNONHEAT\_CPP.CPP** |
| **void** **TurnOnHeat\_CPP**(COFFEEPOT\_DEVICE \*coffeePot\_BaseAddress, **unsigned** **char** heaterLevel, **unsigned** **char** heaterBoost)  {  **#define** DO\_CORRECT\_WAY true  **#if** DO\_CORRECT\_WAY  **#warning**("Compiling Correct Code")  **unsigned** **short** **int** currentRegisterValue = coffeePot\_BaseAddress->controlRegister;  **unsigned** **short** **int** updatedRegisterValue = currentRegisterValue | HEATERPOWERON\_BIT;  updatedRegisterValue = updatedRegisterValue | LED\_HEATERPOWERON\_BIT;  // Set Water Flow Rate to 0  coffeePot\_BaseAddress->waterInFlowRegister = 0;  // Set Heat Level and Boost Level for Heater  coffeePot\_BaseAddress->heaterRegister = heaterLevel;  coffeePot\_BaseAddress->heaterBoostRegister = heaterBoost;  **UpdateSimulationDisplay**();  // Turn on Heat  coffeePot\_BaseAddress->controlRegister = updatedRegisterValue;  **UpdateSimulationDisplay**();  **#else**  **#warning**("Compiling Incorrect Code")  **unsigned** **short** **int** newValue = HEATERPOWERON\_BIT | LED\_HEATERPOWERON\_BIT;  coffeePot\_BaseAddress->controlRegister = newValue;  UpdateSimulationDisplay();  coffeePot\_BaseAddress->heaterRegister = heaterLevel;  UpdateSimulationDisplay();  **#endif**  } |

|  |
| --- |
| **TURNONCOFFEE\_CPP.CPP** |
| **void** **TurnOnCoffee\_CPP**(COFFEEPOT\_DEVICE \* coffeePot\_BaseAddress)  {  // BIT MASK FOR COFFEE POD  **unsigned** **short** **int** currentRegisterValue = coffeePot\_BaseAddress->controlRegister;  **unsigned** **short** **int** updatedRegisterValue = currentRegisterValue | COFFEE\_INSERT\_BIT;  // TURN HEAT DOWN LOW  coffeePot\_BaseAddress->heaterBoostRegister = 1;  coffeePot\_BaseAddress->heaterRegister = 1;  // TURN ON COFFEE  coffeePot\_BaseAddress->controlRegister = updatedRegisterValue;  **printf**("\n\tCoffee Pod Initialized: Coffee Intensity 42/42\n\n");  **UpdateSimulationDisplay**();  } |
|  |
| **PREPARETHECOFFEEPOT\_CPP.CPP** |
| **void** **PrepareTheCoffeePot\_CPP**(COFFEEPOT\_DEVICE \* coffeePot\_BaseAddress)  {  // BIT MASK FOR INITIALIZATION  **unsigned** **short** **int** currentRegisterValue = coffeePot\_BaseAddress->controlRegister;  **unsigned** **short** **int** updatedRegisterValue = currentRegisterValue | INITandSTAYPOWEREDON\_BIT;  updatedRegisterValue = updatedRegisterValue | LED\_POWERON\_BIT;  **bool** deviceReady = false;  // TURN ON INIT BIT  coffeePot\_BaseAddress->controlRegister = updatedRegisterValue;  **UpdateSimulationDisplay**();  // WAIT UNTIL DEVICE READY BIT IS SET  while(!deviceReady)  {  **UpdateSimulationDisplay**();  if(((coffeePot\_BaseAddress->controlRegister & DEVICE\_READY\_BIT\_RO) ^ DEVICE\_READY\_BIT\_RO) == 0x0)  deviceReady = true;  else  deviceReady = false;  }  } |