CEC322 - Section 2

Lab 3 - CubeMX & USART

Lab start date: 02/06/19

Report data: 02/13/19

Andrew Henderson and Tyler Wise

Introduction:

This lab's purpose is to allow students to work directly with the hardware in the code without using the HAL wrapper functions which merely mask CMSIS functionality. The purpose of this is to get students more acquainted with the pointer structure which maps GPIO functionality. As a result, students end with a deeper understanding and appreciation for how the hardware functions getting closer and closer to the underlying assembly code with each iteration.

Code Snippet:

```
/* USER CODE BEGIN 3 */
//If the user is pressing the joystick's "UP" Button
if((JOY_U_GPIO_Port->IDR & JOY_U_Pin) != 0x00u ){
  //Disable both LEDs and wait 2000ms
  //Red
  LED_R_GPIO_Port->BSRR = (uint32_t)LED_R_Pin;
  //Green
  LED_G_GPIO_Port->BSRR = (uint32_t)LED_G_Pin;
  printf("\n\rThe UP KEY is just pressed by Tyler Wise and Andrew Henderson.\n\r");
  HAL_Delay(2000);
}
//If the user is pressing the joystick's "DOWN" Button
else if((JOY_D_GPIO_Port->IDR & JOY_D_Pin) != 0x00u ){
  //Disable both LEDS and wait 2000ms
  //Red
  LED_R_GPIO_Port->BRR = (uint32_t)LED_R_Pin;
  //Green
  LED_G_GPIO_Port->BRR = (uint32_t)LED_G_Pin;
  printf("\n\rThe DOWN KEY is just pressed by Tyler Wise and Andrew Henderson.\n\r");
  HAL_Delay(2000);
}
//If the user is pressing the joystick's "LEFT" Button
else if((JOY_L_GPIO_Port->IDR & JOY_L_Pin) != 0x00u ){
  //Set the Red LED to be enabled, the green one to be disabled, the wait 2000ms
  //Red
  LED_R_GPIO_Port->BSRR = (uint32_t)LED_R_Pin;
  //Green
  LED_G_GPIO_Port->BRR = (uint32_t)LED_G_Pin;
  printf("\n\rThe LEFT KEY is just pressed by Tyler Wise and Andrew Henderson.\n\r");
  HAL_Delay(2000);
}
//If the user is pressing the joystick's "RIGHT" Button
else if((JOY_R_GPIO_Port->IDR & JOY_R_Pin) != 0x00u ){
  //Set the Red LED to be disabled, the green one to be enabled, the wait 2000ms
  //Red
  LED_R_GPIO_Port->BRR = (uint32_t)LED_R_Pin;
  //Green
  LED_G_GPIO_Port->BSRR = (uint32_t)LED_G_Pin;
  printf("\n\rThe RIGHT KEY is just pressed by Tyler Wise and Andrew Henderson.\n\r");
  HAL_Delay(2000);
}
//If the user is pressing the joystick's "CENTER" Button
else if((JOY_C_GPIO_Port->IDR & JOY_C_Pin) != 0x00u ){
  //Set both LEDS to be inactive
  //Red
  LED_R_GPIO_Port->BRR = (uint32_t)LED_R_Pin;
  //Green
  LED_G_GPIO_Port->BRR = (uint32_t)LED_G_Pin;
  printf("\n\rThe CENTER KEY is just pressed by Tyler Wise and Andrew Henderson.\n\r");
  //While 2000ms have not elapsed...
  for(int i = 0; i < 2000; i += 200){
    //Toggle Both LEDS, then wait 200ms
    //Red
    LED_R_GPIO_Port->ODR ^= LED_R_Pin;
    //Green
    LED_G_GPIO_Port->ODR ^= LED_G_Pin;

    HAL_Delay(200);
  }
}
//Otherwise, continuously toggle the LEDs
else {
  // Toggle the red LED, wait 100ms, then toggle the green LED
  //Red
  LED_R_GPIO_Port->ODR ^= LED_R_Pin;
  HAL_Delay(100);
  //Green
  LED_G_GPIO_Port->ODR ^= LED_G_Pin;
  HAL_Delay(100);
}
```

Discussion:

Creating the code for this week was as simple as replacing two commands in each if-else clause, as well as its conditions, from last lab to be CMSIS based rather than HAL based. Following that, a printf statement was added to each user code set to textually identify which direction on the joystick had been pressed.

Results: