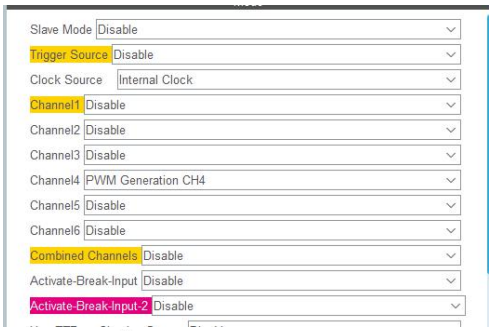
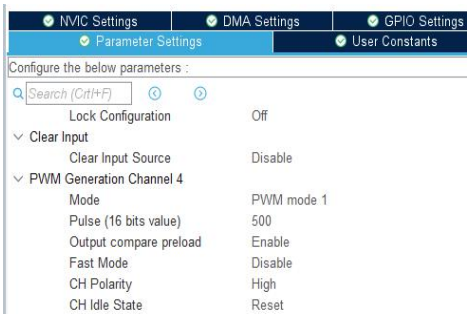
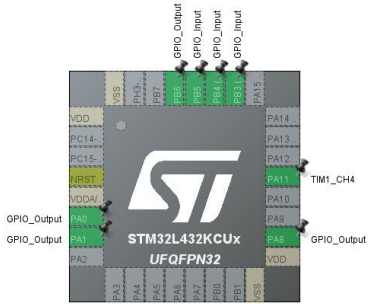
Andrew Stites
EEE 174 – CpE185 Section 2
Summer Session 2
Lab 5
Sean Kennedy

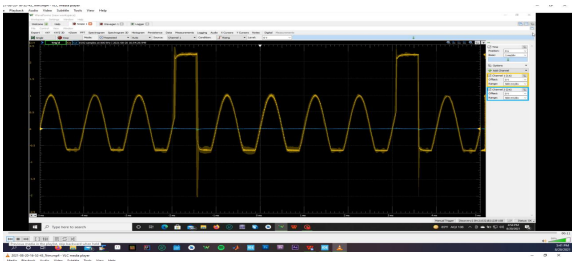
A servo is a DC motor that is coupled with a positional sensor that places the arm at the correct degree due to the Pulse Width Modulated signal given. The GPIO pin PA11 is set as TIM1_CH4 with a wire connecting to the signal pin of the servo. In the “TIM1 Mode and Configuration”, we set “Channel4” to “PWM Generation CH4” set the “Pulse” to 500.

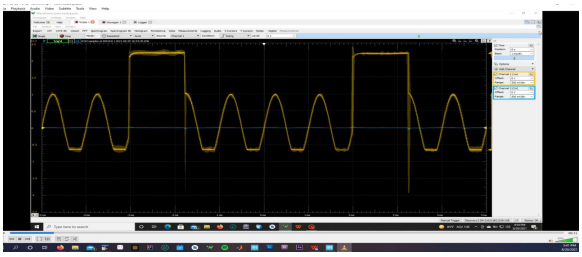


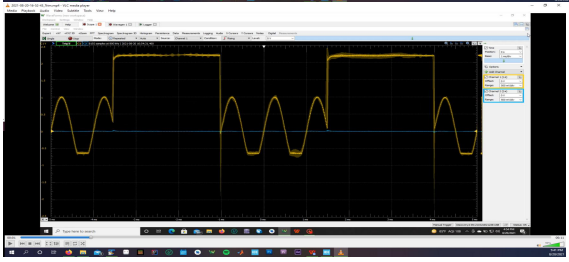



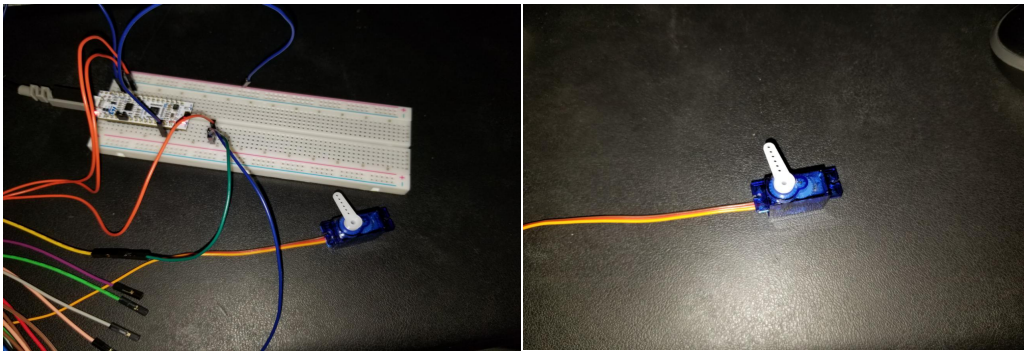

The photos below captured with the AD2 oscilloscope show square waves with edge detection. The first picture shows a smaller square wave when the servo is at 0 degrees. As the servo proceeds to 90 degrees the sqaure wave grows due to the edge detection being different. The final picture shows the largest square wave period due to the PWM being detected by the AD2 oscilloscope. The “Waveform” section was set with a 1kHz frequency, 1ms period, and 1V amplitude.







I placed a jumper wire in “PA11” that connected to the signal line of the servo. I connected the 5V and ground jumper cables from the STM32L432KC to the servo. The code pictured below shows the change in the servo. The first line starts the servo at 0 degrees by putting the “TIM_CHANNEL_4” at a Duty Cycle of 100. The code proceeds through the Duty Cycles of 250 and 500 denoting 90 degrees and 180 degrees. In between the “_HAL_TIM_SetCompare(&hitm1, TIM_CHANNEL_4, 250)” code, there are “HAL_Delay(1000)” lines of code that delay the movement of the arm every 1 second.



```
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{

    __HAL_TIM_SetCompare(&htim1, TIM_CHANNEL_4, 100); // PWM sets servo to 0 degrees
        HAL_Delay(1000);
        __HAL_TIM_SetCompare(&htim1, TIM_CHANNEL_4, 250); // PWM sets servo to 90 degrees
        HAL_Delay(1000);
        __HAL_TIM_SetCompare(&htim1, TIM_CHANNEL_4, 500); // PWM sets servo to 180 degrees
        HAL_Delay(1000);


  /* USER CODE END WHILE */

  /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
```

In this lab, I learned how to setup up a servo motor with the STM32L432KC. In addition, I was able to become more skilled with the Analog Discovery 2 oscilloscope and Waveforms. When setting up the circuit configuration on the breadboard, I had trouble figuring out how to get the servo to move. I naively assumed that the DC motor in the servo would be just that: hook up ground and voltage and be ready to move the arm. However, the introduction of PWM and tying that in with STM32IDE was a arduous journey to say the least. I initially had the servo working on the raspberry pi, but could not find a solution when trying to have the STM32L432KC connect to it and control the signal and failing. I battled with using other sensors due to not properly being able to setup any of them. Finally, the servo code implementation into the STM32L432KC was extremely simplified with the new knowledge of the "TIM1_CHANNEL_4" GPIO pins that can be set and configured to transmit a PWM signal to the servo.

References

https://www.engineersgarage.com/interfacing-servo-motor-with-stm32/