**Lab 1 Blink LED – ECE 5780**

Nate Sheffield

A02268057

Nathan Critchfield

A02283426

**Objective**

The purpose of this lab is to learn how to implement a toggling LED when a button on the STM32 is pressed.

**Procedure**

**Results**

**Figure 1.** STM32 Board with LED activated and deactivated after button press

**Conclusion**

In conclusion,

**Appendix**

***Blink\_LED code***

1. #include "FreeRTOS.h"

2. #include "stm32l476xx.h"

3. #include "system\_stm32l4xx.h"

4. #include "task.h"

5. #include "timers.h"

6. #include "stdint.h"

7.

8. static uint32\_t led\_state;

9. void LED(void \*pvParameters);

10. void Button(void \*pvParameters);

11.

12. //Function to toggle led\_state

13. void LED(void \*pvParameters){

14. while(1){

15. //If the LED is on turn it off

16. if(led\_state == 1){

17. GPIOA->BSRR |= GPIO\_BSRR\_BS5;

18. }

19. //If the LED is off turn it on

20. else {

21. GPIOA->BSRR |= GPIO\_BSRR\_BR5;

22. }

23. }

24. }

25.

26. //Function to read in button state and led\_state

27. void Button(void \*pvParameters){

28. while(1){

29. uint32\_t button\_in;

30. //Read in the value of the button

31. button\_in = GPIOC->IDR;

32. button\_in &= GPIO\_IDR\_ID13\_Msk;

33.

34. //If the button is pressed toggle the LED

35. if(button\_in == 0){

36. while(button\_in == 0){

37. button\_in = GPIOC->IDR;

38. button\_in &= GPIO\_IDR\_ID13\_Msk;

39. }

40. if(led\_state == 0){

41. led\_state = 1;

42. }

43. else {

44. led\_state = 0;

45. }

46. }

47. }

48. }

49.

50. int main(void) {

51. //Initialize System

52. SystemInit();

53.

54. //Turn Clock on GPIOA and GPIOC

55. RCC -> AHB2ENR |= RCC\_AHB2ENR\_GPIOAEN;

56. RCC -> AHB2ENR |= RCC\_AHB2ENR\_GPIOCEN;

57.

58. //Set PA5 to output mode for LED

59. GPIOA->MODER &= ~GPIO\_MODER\_MODE5\_1;

60. GPIOA->MODER |= GPIO\_MODER\_MODE5\_0;

61. //Turn LED on

62. GPIOA->BSRR |= GPIO\_BSRR\_BS5;

63. led\_state = 1;

64.

65. //Set PC13 to input mode for Button

66. GPIOC->MODER &= ~GPIO\_MODER\_MODE13; //0xf3ffffff

67.

68.

69.

70. xTaskCreate( //Task for LED

71. LED,

72. "LED",

73. 16,

74. NULL,

75. 1,

76. NULL);

77.

78. xTaskCreate( //Task for Button

79. Button,

80. "Button",

81. 16,

82. NULL,

83. 1,

84. NULL);

85.

86. //Start Task Scheduler

87. vTaskStartScheduler();

88. while(1);

89. }

90.