

# EEE3096S Prac 1

HWKLAW001 (Lawrence Hawke) and DWDZAI001 (Zainodine Dawood)

August 2024

## 1 Code

### 1.1 Variables

```
uint32_t current_time = 0;
uint32_t prev_time = 0;
uint32_t delay_led = 500; //500ms delay

uint8_t data[6] = 0b10101010, 0b01010101, 0b11001100, 0b00110011, 0b11110000,
0b00001111;
uint16_t address = 0; //EEprom address
uint32_t adc_val;
```

### 1.2 Write bytes to EEPROM

Iterates through bytes stored in data variable.

```
uint8_t index = 0;
while(index < 6)
write_to_address(address, data[index]);

index
spi_delay(100);
```

### 1.3 While Loop Code

Polls ADC, gets value. Converts to CCR value. Updates the PWM value.

```
// TODO: Poll ADC
adc_val = pollADC();//read analogue adc value from potentiometer
```

```

    // TODO: Get CRR
    CCR = ADCtoCCR(adc_val);

    // Update PWM value ( /4 to make it turn off)
    _HAL_TIM_SetCompare(&htim3, TIM_CHANNEL_3, CCR/4);

```

## 1.4 Exit Trigger Interrupt Handler code

Switches between 2Hz and 1Hz if switch is pressed.

```

    // TODO: Add code to switch LED7 delay frequency
    current_time = HAL_GetTick();

    //ensures unwanted noise within duration is not registered
    if((current_time - prev_time) < 200)
    if(delay_led == 500 ) //if frequency of led is 2Hz
    delay_led = 1000; //toggle the frequency of LED by changing delay
    htim6.Init.Period = delay_led -1;
    else if(delay_led == 1000) //if frequency of led is 1Hz
    delay_led = 500;
    htim6.Init.Period = delay_led -1;

    //update TIM6 with the new period; ensure execution complete
    if (HAL_TIM_Base_Init(&htim6) != HAL_OK)

    Error_Handler();

```

## 1.5 Timer 16 Interrupt Request Handler

Called every second. Writes decimal of bytes in EEPROM, delays for 100ms, and iterates the address. Checks if byte stored is byte read, if it isn't, it outputs an error message.

```

    // TODO: Initialise a string to output second line on LCD
    char charArray[16]; //buffer

    // TODO: Change LED pattern; output 0x01 if the read SPI data is incor-
    rect
    if (address < 5)

        address= 0;

```

```

//validate byte at address
uint8_t num = read_from_address(address);
spi_delay(100);
if (num == data[address])

snprintf(charArray, sizeof(charArray), "writeLCD(charArray);

    else
writeLCD("SPI ERROR!");

//iterate address
address++;

//update TIM16 with the new period; ensure execution complete
if (HAL_TIM_Base_Init(&htim16) != HAL_OK)

Error_Handler();

```

## 1.6 Write to LCD Function

Sets LCD to the second line to output the byte in decimal.

// TODO: Complete the writeLCD function

```

void writeLCD(char *char_in)
delay(3000);
lcd_command(CLEAR);
lcd_putstring("EEPROM byte:");
lcd_command(LINE_TWO);
lcd_putstring(char_in);

```

## 1.7 Polling ADC

Accesses the ADC using HAL libraries, function returns a value from 0 to 4095.

```

// Get ADC value
uint32_t pollADC(void)
HAL_ADC_Start(&hadc); // start the adc
HAL_ADC_PollForConversion(&hadc, 100); // poll for conversion
uint32_t val = HAL_ADC_GetValue(&hadc); // get the adc value
HAL_ADC_Stop(&hadc); // stop adc

```

```
return val;
```

## 1.8 CCR calc

Calculates capture compare register number to use in the function which alters the duty cycle of the 48000 Hz square wave outputted to LED7, hence `adc_val * 47999`.

```
// Calculate PWM CCR value
uint32_t ADCtoCCR(uint32_t adc_val)
// TODO: Calculate CCR value (val) using an appropriate equation
uint32_t val_ccr;

val_ccr = (adc_val * 47999) / 4095;

return val_ccr;
```

## 2 Git Hub Link

<https://github.com/Lawrenceismyname/EEE3096Spracs//>

## 3 Appendix

Full `main.c` file is attached below: