Buzzer

Timer2, PWM를 활용하여 구현

초기 설정

[Stm32l4xx_it.h]

```
void TIM2_IRQHandler(void);
```

[main.h]

```
void HAL_TIM_MspPostInit(TIM_HandleTypeDef *htim);

#define Buzzer_Pin              GPIO_PIN_0     //PA0
#define Buzzer_GPIO_Port        GPIOA
```

[stm32l4xx_hal_msp.c]

timer, pwm 설정

```
void HAL_TIM_Base_MspInit(TIM_HandleTypeDef* htim_base)
{
  if(htim_base->Instance==TIM6)
  {
  /* USER CODE BEGIN TIM6_MspInit 0 */

  /* USER CODE END TIM6_MspInit 0 */
    /* Peripheral clock enable */
    __HAL_RCC_TIM6_CLK_ENABLE();
    /* TIM6 interrupt Init */
    HAL_NVIC_SetPriority(TIM6_DAC_IRQn, 2, 0);
    HAL_NVIC_EnableIRQ(TIM6_DAC_IRQn);
  /* USER CODE BEGIN TIM6_MspInit 1 */

  /* USER CODE END TIM6_MspInit 1 */
  }
  else if(htim_base->Instance==TIM7)
  {
  /* USER CODE BEGIN TIM7_MspInit 0 */

  /* USER CODE END TIM7_MspInit 0 */
    /* Peripheral clock enable */
    __HAL_RCC_TIM7_CLK_ENABLE();
    /* TIM7 interrupt Init */
    HAL_NVIC_SetPriority(TIM7_IRQn, 2, 0);
```

```
    HAL_NVIC_EnableIRQ(TIM7_IRQn);
  /* USER CODE BEGIN TIM7_MspInit 1 */

  /* USER CODE END TIM7_MspInit 1 */
  }
  else if(htim_base->Instance==TIM2)
  {
  /* USER CODE BEGIN TIM2_MspInit 0 */

  /* USER CODE END TIM2_MspInit 0 */
    /* Peripheral clock enable */
    __HAL_RCC_TIM2_CLK_ENABLE();
    /* TIM2 interrupt Init */
    HAL_NVIC_SetPriority(TIM2_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(TIM2_IRQn);
  /* USER CODE BEGIN TIM2_MspInit 1 */

  /* USER CODE END TIM2_MspInit 1 */
  }

}

void HAL_TIM_MspPostInit(TIM_HandleTypeDef* htim)
{
  GPIO_InitTypeDef GPIO_InitStruct = {0};
  if(htim->Instance==TIM2)
  {
  /* USER CODE BEGIN TIM2_MspPostInit 0 */

  /* USER CODE END TIM2_MspPostInit 0 */

    __HAL_RCC_GPIOA_CLK_ENABLE();
    /**TIM2 GPIO Configuration
    PA0     ------> TIM2_CH1
    */
    GPIO_InitStruct.Pin = Buzzer_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_AF_PP;
    GPIO_InitStruct.Pull = GPIO_PULLDOWN;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    GPIO_InitStruct.Alternate = GPIO_AF1_TIM2;
    HAL_GPIO_Init(Buzzer_GPIO_Port, &GPIO_InitStruct);

  /* USER CODE BEGIN TIM2_MspPostInit 1 */

  /* USER CODE END TIM2_MspPostInit 1 */
  }

}

void HAL_TIM_Base_MspDeInit(TIM_HandleTypeDef* htim_base)
{
  if(htim_base->Instance==TIM6)
  {
  /* USER CODE BEGIN TIM6_MspDeInit 0 */

  /* USER CODE END TIM6_MspDeInit 0 */
    /* Peripheral clock disable */
    __HAL_RCC_TIM6_CLK_DISABLE();
```

```
    /* TIM6 interrupt DeInit */
    HAL_NVIC_DisableIRQ(TIM6_DAC_IRQn);
  /* USER CODE BEGIN TIM6_MspDeInit 1 */

  /* USER CODE END TIM6_MspDeInit 1 */
  }
  else if(htim_base->Instance==TIM7)
  {
  /* USER CODE BEGIN TIM7_MspDeInit 0 */

  /* USER CODE END TIM7_MspDeInit 0 */
    /* Peripheral clock disable */
    __HAL_RCC_TIM7_CLK_DISABLE();

    /* TIM7 interrupt DeInit */
    HAL_NVIC_DisableIRQ(TIM7_IRQn);
  /* USER CODE BEGIN TIM7_MspDeInit 1 */

  /* USER CODE END TIM7_MspDeInit 1 */
  }
  else if(htim_base->Instance==TIM2)
  {
  /* USER CODE BEGIN TIM2_MspDeInit 0 */

  /* USER CODE END TIM2_MspDeInit 0 */
    /* Peripheral clock disable */
    __HAL_RCC_TIM2_CLK_DISABLE();

    /* TIM2 interrupt DeInit */
    HAL_NVIC_DisableIRQ(TIM2_IRQn);
  /* USER CODE BEGIN TIM2_MspDeInit 1 */

  /* USER CODE END TIM2_MspDeInit 1 */
  }
}

void HAL_TIM_PWM_MspInit(TIM_HandleTypeDef* htim_pwm)
{

  if(htim_pwm->Instance==TIM2)
  {
  /* USER CODE BEGIN TIM3_MspInit 0 */

  /* USER CODE END TIM3_MspInit 0 */
    /* Peripheral clock enable */
    __HAL_RCC_TIM2_CLK_ENABLE();
  /* USER CODE BEGIN TIM3_MspInit 1 */

  /* USER CODE END TIM3_MspInit 1 */
  }

}

void HAL_TIM_PWM_MspDeInit(TIM_HandleTypeDef* htim_pwm)
{

  if(htim_pwm->Instance==TIM2)
```

```
  {
  /* USER CODE BEGIN TIM3_MspDeInit 0 */

  /* USER CODE END TIM3_MspDeInit 0 */
    /* Peripheral clock disable */
    __HAL_RCC_TIM2_CLK_DISABLE();
  /* USER CODE BEGIN TIM3_MspDeInit 1 */

  /* USER CODE END TIM3_MspDeInit 1 */
  }

}
```

[stm32l4xx_it.c]

```
extern TIM_HandleTypeDef htim2;

void TIM2_IRQHandler(void)
{
  /* USER CODE BEGIN TIM2_IRQn 0 */

  /* USER CODE END TIM2_IRQn 0 */
  HAL_TIM_IRQHandler(&htim2);
  /* USER CODE BEGIN TIM2_IRQn 1 */

  /* USER CODE END TIM2_IRQn 1 */
}
```

[main.c]

```
static void MX_GPIO_Init(void)
{
  /*Configure GPIO pins : PA0 */   // Buzzezr Control
  GPIO_InitStruct.Pin =  GPIO_PIN_0 ;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_PULLDOWN;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
}
```

```
static void MX_TIM2_Init(void)  // Buzzer PWM Timer
{

  /* USER CODE BEGIN TIM2_Init 0 */

  /* USER CODE END TIM2_Init 0 */

  TIM_ClockConfigTypeDef sClockSourceConfig = {0};
  TIM_MasterConfigTypeDef sMasterConfig = {0};
  TIM_OC_InitTypeDef sConfigOC = {0};
```

```
  /* USER CODE BEGIN TIM2_Init 1 */

  /* USER CODE END TIM2_Init 1 */
  htim2.Instance = TIM2;
  htim2.Init.Prescaler = 47;
  htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
  htim2.Init.Period = 1000;
  htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
  htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
  if (HAL_TIM_Base_Init(&htim2) != HAL_OK)
  {
    Error_Handler();
  }
  sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
  if (HAL_TIM_ConfigClockSource(&htim2, &sClockSourceConfig) != HAL_OK)
  {
    Error_Handler();
  }
  if (HAL_TIM_PWM_Init(&htim2) != HAL_OK)
  {
    Error_Handler();
  }
  sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
  sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
  if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) != HAL_OK)
  {
    Error_Handler();
  }
  sConfigOC.OCMode = TIM_OCMODE_PWM1;
  sConfigOC.Pulse = 250;
  sConfigOC.OCPolarity = TIM_OCPOLARITY_HIGH;
  sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
  if (HAL_TIM_PWM_ConfigChannel(&htim2, &sConfigOC, TIM_CHANNEL_1) != HAL_OK)
  {
    Error_Handler();
  }
  /* USER CODE BEGIN TIM2_Init 2 */

  /* USER CODE END TIM2_Init 2 */
  HAL_TIM_MspPostInit(&htim2);

}
```

```
//전역변수
TIM_HandleTypeDef htim2;
static void MX_TIM2_Init(void);
```

부저 기능 관련 함수

```
int main(void)
{
```

```
  MX_TIM2_Init();
}
```

음계 정의

```
enum notes
{
  N1 = 460,
  N2 = 450,
  N3 = 440,
  N4 = 430,
  N5 = 420,
  N6 = 410,
  N7 = 400,
  N8 = 390,
  N9 = 380,
  N10 = 370
};

enum notes on[] = { N1,N2,N3,N5 };
enum notes low[] = { N8,N7 };
```

체결 시 알림

```
void Buzzer_ON()
{
//  for (uint8_t i = 0; i < sizeof(on) / sizeof(enum notes); i++)
//  {
//    __HAL_TIM_SET_AUTORELOAD(&htim2, on[i]);
//    __HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_1, on[i]/2);
//    HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);  //BUZZER
//    HAL_Delay (300);
//    HAL_TIM_PWM_Stop (&htim2, TIM_CHANNEL_1);
//    HAL_Delay (150);
//  }
  for(int count=0; count<5; count++)
  {
    HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);  //BUZZER
    HAL_Delay (300);
    HAL_TIM_PWM_Stop (&htim2, TIM_CHANNEL_1);
    HAL_Delay (300);
  }
}
```

Low Batt 시 알림

```
void Buzzer_Low()
{
  for(int count=0; count<5; count++)
```

```
  {
    for (uint8_t i = 0; i < sizeof(low) / sizeof(enum notes); i++)
    {
      __HAL_TIM_SET_AUTORELOAD(&htim2, low[i]);
            __HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_1, low[i]/2);
      HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);  //BUZZER
      HAL_Delay (300);
      HAL_TIM_PWM_Stop (&htim2, TIM_CHANNEL_1);
      HAL_Delay (150);
    }
  }
}
```