

```

1  /*-----
2  * Name:      Square.c
3  * Purpose:   Pulse gerator, with a duty cyle variable by the user.
4  * Note(s):   Code modified from ST MicroElectronics Application Teams,
5              TIM_PWM_Output example project.
6  *-----
7  *
8  *-----*/
9
10 #include "STM32F4xx.h"
11 #include "LCD.h"
12 #include "Sqaure.h"
13
14 #define TIM3_CLK_OUT 42000
15 #define TIM3_CNT_CLK 28000000
16 #define TIM3_ARR 665 //((TIM3_CNT_CLK / TIM3_CLK_OUT) - 1)
17 #define FIFTY_PERCENT 333
18
19 void Pulse_Config(void) {
20     // Run timer config and initialise pulses to 50:50 duty cycle
21     TIM3_Config();
22     PWM_Config(TIM3_ARR);
23 }
24
25 void TIM3_Config(void) {
26     GPIO_InitTypeDef GPIO_InitStructure;
27
28     /* TIM3 clock enable */
29     RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM3, ENABLE);
30
31     /* GPIOC clock enable */
32     RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE);
33
34     /* GPIOC Configuration: TIM3 CH1 (PC6) */
35     GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6 ;
36     GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
37     GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
38     GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
39     GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP ;
40     GPIO_Init(GPIOC, &GPIO_InitStructure);
41
42     /* Connect TIM3 pins to AF2 */
43     GPIO_PinAFConfig(GPIOC, GPIO_PinSource6, GPIO_AF_TIM3);
44 }
45
46 void PWM_Config(int period)
47 {
48     TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;
49     TIM_OCInitTypeDef TIM_OCInitStructure;
50     uint16_t PrescalerValue = 0;
51
52     /* Compute the prescaler value */
53     PrescalerValue = (uint16_t) ((SystemCoreClock /2) / 28000000) - 1;
54
55     /* Time base configuration */
56     TIM_TimeBaseStructure.TIM_Period = 665;
57     TIM_TimeBaseStructure.TIM_Prescaler = PrescalerValue;
58     TIM_TimeBaseStructure.TIM_ClockDivision = 0;
59     TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
60
61     TIM_TimeBaseInit(TIM3, &TIM_TimeBaseStructure);
62
63     /* PWM1 Mode configuration: Channell */
64     TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;
65     TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
66     TIM_OCInitStructure.TIM_Pulse = FIFTY_PERCENT; // 50:50 duty cyle
67     TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;
68
69     TIM_OC1Init(TIM3, &TIM_OCInitStructure);
70
71     TIM_OC1PreloadConfig(TIM3, TIM_OCPreload_Enable);
72
73     TIM_ARRPreloadConfig(TIM3, ENABLE);
74
75     /* TIM3 enable counter */
76     TIM_Cmd(TIM3, ENABLE);
77 }
78

```

```
79 void PWM_SetDC(uint16_t dutycycle)
80 {
81     uint16_t newDutyCycle;
82
83     // Calculate the new duty cycle
84     newDutyCycle = (dutycycle * TIM3_ARR) / 100;
85
86     // set the new duty cycle into the capture compare register
87     TIM_SetCompare1(TIM3, newDutyCycle);
88 }
89
```