

```

1  /*****
2  * Name:      TIM.c
3  * Description: STM32 Timer initialization and functions
4  * Version: V1.00
5  * Author: Ammar Alvi & Shannon D'Souza
6  *
7  * This software is supplied "AS IS" without warranties of any kind.
8  *
9  *****/
10 #include "stm32f10x.h"
11 #include "GPIO.h"
12 #include "UTIL.h"
13
14 /*
15 * Name:      void TIM1_PWM_init(void)
16 * Parameters: none
17 * Description: Writes to TIM1 registers for PWM initialization on
18                PA8, PA9, PA10, PA11
19 */
20
21 void TIM1_PWM_init(void)
22 {
23     TIM1->CR1 |= TIM_CR1_CEN;
24     TIM1->CR2 |= TIM_CR2_OIS1 | TIM_CR2_OIS2 | TIM_CR2_OIS3 | TIM_CR2_OIS4;
25     TIM1->EGR |= TIM_EGR_UG;
26
27     TIM1->CCMR1 |= TIM_CCMR1_OC1PE | TIM_CCMR1_OC1FE | TIM_CCMR1_OC1M_1 | TIM_CCMR1_OC1M_2;
28     TIM1->CCMR1 |= TIM_CCMR1_OC2PE | TIM_CCMR1_OC2FE | TIM_CCMR1_OC2M_1 | TIM_CCMR1_OC2M_2;
29     TIM1->CCMR2 |= TIM_CCMR2_OC3PE | TIM_CCMR2_OC3FE | TIM_CCMR2_OC3M_1 | TIM_CCMR2_OC3M_2;
30     TIM1->CCMR2 |= TIM_CCMR2_OC4PE | TIM_CCMR2_OC4FE | TIM_CCMR2_OC4M_1 | TIM_CCMR2_OC4M_2;
31     TIM1->CCER |= TIM_CCER_CC1E | TIM_CCER_CC2E | TIM_CCER_CC3E | TIM_CCER_CC4E;    //Enable Capture
compare register
32
33     TIM1->PSC = 0x095F;                //Divide 24Mhz by 24, PSC_CLK = 1000000 Hz, 1 count = 0.000001s = 1us
34     TIM1->ARR = 20;                    //20 counts = 20us    f = 50 KHz    (PWM frequency)
35     TIM1->CCR1 = 21;                   //21 countes = 100% duty cycle = logic 1
36     TIM1->CCR2 = 21;
37     TIM1->CCR3 = 21;
38     TIM1->CCR4 = 21;
39     TIM1->BDTR |= TIM_BDTR_MOE | TIM_BDTR_OSSI; //Main Output Enable, Force Idle Level First
40     TIM1->CR1 |= TIM_CR1_ARPE | TIM_CR1_CEN; //Enable Timer 1
41 }
42
43 /*
44 * Name:      void TIM1_Brake(void)
45 * Parameters: none
46 * Description: Writes logic 1 to all 4 ports to stop left and right side motors
47 */
48 void TIM1_Brake(void)
49 {
50     TIM1->CCR1 = 21;
51     TIM1->CCR2 = 21;
52     TIM1->CCR3 = 21;
53     TIM1->CCR4 = 21;
54 }
55
56 /*
57 * Name:      void TIM1_Forward(void)
58 * Parameters: none
59 * Description: Writes logic 1 to PA9, and PA11 but logic 0 to PA8, and PA10
60                to make both sides of the motor run at full speed
61 */
62 void TIM1_Forward(void)
63 {
64     TIM1->CCR1 = 1;
65     TIM1->CCR2 = 21;
66     TIM1->CCR3 = 1;
67     TIM1->CCR4 = 21;
68 }
69
70 /*
71 * Name:      void TIM1_ForwardAdj(void)
72 * Parameters: 8bit value to set the duty cycle of the PWM

```

```
72  * Description:  Writes logic 1 to PA9, and PA11 but varying duty cycle to PA8, and PA10
73                  to make both sides of the motor run at adjustable speed
74  */
75  void TIM1_ForwardAdj(uint8_t val)
76  {
77      TIM1->CCR1 = val;
78      TIM1->CCR2 = 21;
79      TIM1->CCR3 = val;
80      TIM1->CCR4 = 21;
81  }
82
83  /*
84  * Name:          void TIM1_TurnLeft(void)
85  * Paramaters:    none
86  * Description:   Writes logic 1 to PA8, PA9, and PA11 but logic 0 to PA10
87                  to make Right side of the motor run at full speed to make
88                  a left turn (differential steering)
89  */
90  void TIM1_TurnLeft(void)
91  {
92      TIM1->CCR1 = 21;
93      TIM1->CCR2 = 21;
94      TIM1->CCR3 = 1;
95      TIM1->CCR4 = 21;
96  }
97  /*
98  * Name:          void TIM1_TurnRight(void)
99  * Paramaters:    none
100 * Description:   Writes logic 1 to PA9, PA10, PA11 but logic 0 to PA8
101                 to make Left side of the motor run at full speed to make
102                 a Right turn (differential steering)
103  */
104  void TIM1_TurnRight(void)
105  {
106      TIM1->CCR1 = 1;
107      TIM1->CCR2 = 21;
108      TIM1->CCR3 = 21;
109      TIM1->CCR4 = 21;
110  }
111
```