Lab 7: Pulse Width Modulation
Instructor: Prof. Yifeng Zhu
Spring 2016

Goals

1. Understand the concept of Pulse Width Modulation (PWM)
2. Use PWM to control the LED brightness

Pre-Lab Assignment

1. Read Chapter 15.3 PWM Output
2. Complete the pin and timer configuration tables

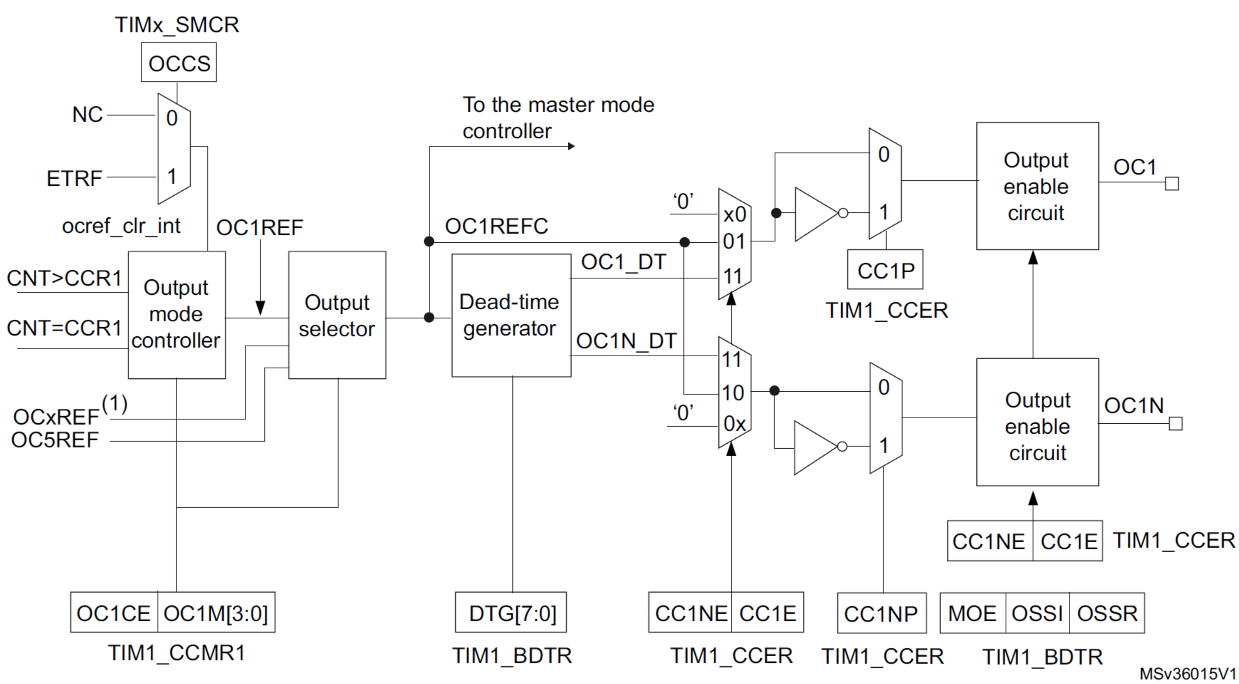
Lab Demo

1. Periodically dimming a LED
2. Use an oscilloscope to measure duty cycles.
3. Something cool. Note that dimming another LED does not count as something cool. The following gives a few example of something cool.
    a. Use PWM to control stepper motors to perform micro-stepping
    b. Use PWM to generate a music tone (a 440-Hz sine wave, tone A), or play a song

Post-Lab Assignment

1. Complete the post lab report and write your answer in Readme.md

Timer Control for Channel 1 of Timer X:



The above diagram shows the functions of each pin. The green LED is connected to the pin PE 8. The following table shows the alternative functions available for these two pins.

LED	Pin	Available Alternative Functions
Green	PE 8	**TIM1_CH1N**/DFSDM_CKIN2/FMC_D5/SAI1_SCK_B/EVENTOUT

Migrating from STM32L1 to STM32L4:

When you use the flowchart in Figure 15-11 (Page 372 of textbook) to program a timer as PWM output, pay attention to the following when configuring the timer.

	STM32L1	**STM32L4**
Output Enable	**OCx output is enabled by the CCxE bit in the TIMx_CCER register.**	**OCx output is enabled by a combination of the CCxE, CCxNE, MOE, OSSI and OSSR bits (TIMx_CCER and TIMx_BDTR registers).**
OCxN	**Only has main output OCx but no complementary output OCxN**	**Has both OCx and OCxN.**

Refer to the following Table 147 for controlling the outputs OCx and OCxN.

Table 147. Output control bits for complementary OCx and OCxN channels with break feature

Control bits					**Output states[1]**	
MOE bit	**OSSI bit**	**OSSR bit**	**CCxE bit**	**CCxNE bit**	**OCx output state**	**OCxN output state**
1	X	X	0	0	Output disabled (not driven by the timer: Hi-Z) OCx=0, OCxN=0	
		0	0	1	Output disabled (not driven by the timer: Hi-Z) OCx=0	OCxREF + Polarity OCxN = OCxREF xor CCxNP
		0	1	0	OCxREF + Polarity OCx=OCxREF xor CCxP	Output Disabled (not driven by the timer: Hi-Z) OCxN=0
		X	1	1	OCREF + Polarity + dead-time	Complementary to OCREF (not OCREF) + Polarity + dead-time
		1	0	1	Off-State (output enabled with inactive state) OCx=CCxP	OCxREF + Polarity OCxN = OCxREF x or CCxNP
		1	1	0	OCxREF + Polarity OCx=OCxREF xor CCxP	Off-State (output enabled with inactive state) OCxN=CCxNP
0	0	X	X	X	Output Disabled (not driven by the timer: Hi-Z) OCx=CCxP, OCxN=CCxNP	
	1		0	0		
			0	1	Off-State (output enabled with inactive state) Asynchronously: OCx=CCxP, OCxN=CCxNP (if BRK or BRK2 is triggered). Then (this is valid only if BRK is triggered), if the clock is present: OCx=OISx and OCxN=OISxN after a dead-time, assuming that OISx and OISxN do not correspond to OCX and OCxN both in active state (may cause a short circuit when driving switches in half-bridge configuration). **Note**: BRK2 can only be used if OSSI = OSSR = 1.	
			1	0		
			1	1		

1. When both outputs of a channel are not used (control taken over by GPIO), the OISx, OISxN, CCxP and CCxNP bits must be kept cleared.

Play a Song with PWM

You can play a song by using a speaker, such as PKM22EPP.

Murata Electronics PKM22EPPH4001-B0 Speaker (Digikey # 490-4692-ND)

Comments about comments

Properly documenting is especially important for assembly programs.

Thumb of Rule: "*If you are able to easily and quickly modify your codes one year after your initial development, then your code comments are excellent.*"

When writing comments, you can assume that readers of your codes know the assembly syntax but do not know your program goals, logic, and methodology.

- Give meaningful name to registers
- Recommend to using corresponding C code to make comment

Instruction	Bad comment	Good comment
add r2, r2, #1	; add one to a register	; array_index++
sub r2, r2, #1	; r2 = r2 - 1	; counter++
mul r3, r2, r1	; r3 = r2 * r1	; distance = speed * time

Specific Requirements

For each subroutine (also called function or procedure), your program should have a comment header. Your program should include a register usage page. Give each register a meaningful name. Here is an example.

```
; Name: strcat
; Description: Concatenate two strings
; Input: r0 – pointer to the first string
;        r1 – pointer to the second string
; Returns: r0 – pointer to the resulting string
; Register usage:
;        r3 – array index i of the second string
;        r4 – array index j of the resulting string
;        r5 – temporary value to hold a char loaded from memory

strcat   PROC
         ...
         ENDP
```

Lab 7: Pre-Lab Assignment

Student Name: ____________________________

NOTE:

- The Green LED (PE 8) is connected the Channel 1N of Timer 1.

1. Configure RCC_AHB2ENR to enable the clock of GPIO Port E

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AHB2ENR														RNGEN		AESEN			ADCEN	OTGFSEN					GPIOPHEN	GPIOPGEN	GPIOPFEN	GPIOPEEN	GPIOPDEN	GPIOPCEN	GPIOPBEN	GPIOPAEN
Mask																																
Value																																

Write your assembly code below to enable the GPIO B clock:

2. Configure RCC_APB2ENR to enable the clock of Timer 1

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
APB2ENR								DFSDMEN		SAI2EN	SAI1EN			TIM17EN	TIM16EN	TIM15EN		USART1EN	TIM8EN	SPI1EN	TIM1EN	SDMMC1EN			FIREWALLEN						TIM3EN	SYSCFGEN
Mask																																
Value																																

Write your assembly code below to enable the Timer 1 clock:

3. Configure PE 8(Green LED) as Alternative Function Mode

GPIO Mode: Input (00), Output (01), AlterFunc (10), Analog (11)

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO MODER	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
Mask																																
Value																																

GPIOE Mode Register MASK Value = 0x____________________ (in HEX)
GPIOE Mode Register Value = 0x____________________ (in HEX)

4. Configure and Select the Alternative Function for PE 8 (Green LED)

Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPIO AFR[0]	AFRL7[3:0]				AFRL6[3:0]				AFRL5[3:0]				AFRL4[3:0]				AFRL3[3:0]				AFRL2[3:0]				AFRL1[3:0]				AFRL0[3:0]			
MASK																																
VALUE																																
GPIO AFR[1]	AFRH15[3:0]				AFRH14[3:0]				AFRH13[3:0]				AFRH12[3:0]				AFRH11[3:0]				AFRH10[3:0]				AFRH9[3:0]				AFRH8[3:0]			
MASK																																
VALUE																																

GPIOE Alternative Function Register [0] MASK = 0x____________________ (in HEX)
GPIOE Alternative Function Register [0] = 0x____________________ (in HEX)
GPIOE Alternative Function Register [1] MASK = 0x____________________ (in HEX)
GPIOE Alternative Function Register [1] = 0x____________________ (in HEX)

5. Complete the following table to configure the PWM output for Channel 1N of Timer 1

Note that the alternative function of PE 8 is TIM1N, not TIM1*. *You need to set the CC1NE bit in the TIM1_CCER register. Refer to the table given on Page 2 of the lab handout for controlling OC1N. Note that you need to make minor changes to the flow chart given in Figure 15-11 (Page 372 of textbook).

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	TIM1_CR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	UIFREMAP	Res	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	value																																
0x04	TIM1_CR2	Res	Res	Res	Res	Res	Res	Res	Res	MMS2[3:0]				Res	OIS6	Res	OIS5	Res	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS [2:0]			CCDS	CCUS	Res	CCPC
	value																																
0x08	TIM1_SMCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SMS[3]	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			OCCS	SMS[2:0]		
	value																																
0x0C	TIM1_DIER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	value																																
0x10	TIM1_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	C6IF	C5IF	Res	Res	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	value																																
0x14	TIM1_EGR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	B2G	BG	TG	COM	CC4G	CC3G	CC2G	CC1G	UG
	value																																
0x18	TIM1_CCMR1 Output Compare mode	Res	Res	Res	Res	Res	Res	Res	OC2M[3]	Res	Res	Res	Res	Res	Res	Res	OC1M[3]	OC2CE	OC2M [2:0]			OC2PE	OC2FE	CC2 S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1 S [1:0]	
	Value																																
	TIM1_CCMR1 Input Capture mode	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC2F[3:0]				IC2 PSC [1:0]		CC2 S [1:0]		IC1F[3:0]				IC1 PSC [1:0]		CC1 S [1:0]	
	Value																																
0x1C	TIM1_CCMR2 Output Compare mode	Res	Res	Res	Res	Res	Res	Res	OC4M[3]	Res	Res	Res	Res	Res	Res	Res	OC3M[3]	OC4CE	OC4M [2:0]			OC4PE	OC4FE	CC4 S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3 S [1:0]	
	Value																																
	TIM1_CCMR2 Input Capture mode	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	IC4F[3:0]				IC4 PSC [1:0]		CC4 S [1:0]		IC3F[3:0]				IC3 PSC [1:0]		CC3 S [1:0]	
	Value																																

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x20	TIM1_CCER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CC6P	CC6E	Res	Res	CC5P	CC5E	Res	Res	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	Value																																
0x24	TIM1_CNT	UIFCPY	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CNT[15:0]															
	Value																																
0x28	TIM1_PSC	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PSC[15:0]															
	Value																																
0x2C	TIM1_ARR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ARR[15:0]															
	Value																																
0x30	TIM1_RCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REP[15:0]															
	Value																																
0x34	TIM1_CCR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR1[15:0]															
	Value																																
0x38	TIM1_CCR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR2[15:0]															
	Value																																
0x3C	TIM1_CCR3	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR3[15:0]															
	Value																																
0x40	TIM1_CCR4	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR4[15:0]															
	Value																																
0x44	TIM1_BDTR	Res	Res	Res	Res	Res	Res	BK2P	BK2E	BK2F[3:0]				BKF[3:0]				MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DT[7:0]							
	Value																																
0x48	TIM1_DCR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DBL[4:0]					Res	Res	Res	DBA[4:0]				
	Value																																
0x4C	TIM1_DMAR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DMAB[15:0]															
	Value																																
0x50	TIM1_OR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TI1_RMP	ETR_ADC3_RMP		ETR_ADC1_RMP	
	Value																																

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x54	**TIM1_CCMR3 Output Compare mode**	Res	Res	Res	Res	Res	Res	Res	OC6M[3]	Res	Res	Res	Res	Res	Res	Res	OC5M[3]	OC6CE	OC6M [2:0]			OC6PE	OC6FE	Res	Res	OC5CE	OC5M [2:0]			OC5PE	OC5FE	Res	Res
	Value																																
0x58	**TIM1_CCR5**	GC5C3	GC5C2	GC5C1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR5[15:0]															
	Value																																
0x5C	**TIM1_CCR6**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CCR6[15:0]															
	Value																																
0x60	**TIM1_OR2**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ETRSEL [2:0]			Res	Res	BKCMP2P	BKCMP1P	BKINP	BKDFBK0E	Res	Res	Res	Res	Res	BKCMP2E	BKCMP1E	BKINE
	Value																																
0x64	**TIM1_OR3**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	BK2CMP2P	BK2CMP1P	BK2INP	BK2DFBK1E	Res	Res	Res	Res	Res	BK2CMP2E	BK2CMP1E	BK2INE
	Value																																

ECE 271 Microcomputer Architecture and Applications
Lab 7: Pulse Width Modulation
Lab Demo

Student Name: ____________________________

Demo 1: Diming an LED
Demo 2: Use an oscilloscope to measure duty cycles.

Keep TIM1_ARR fixed but set TIM1_CCR1 to three different values.

TIM1_ARR = ______________________
TIM1_PSC = ______________________

Case		TIM1_CCR1	Pulse Width Measured	Pulse Period Measured	Duty Cycle Measured
#1	TIM1_CCR1 = 1/6* TIM1_ARR				
#2	TIM1_CCR1 = 1/3* TIM1_ARR				
#3	TIM1_CCR1 = 1/2* TIM1_ARR				

ECE 271 Microcomputer Architecture and Applications
Lab 7: Pulse Width Modulation
Post-Lab Assignment

Suppose the HSE (high-speed external clock) of 16 MHz is selected the clock of a timer. In order to generate 1 Hz square wave with duty cycle of 50%, how would set up the timer? Indicate your counting mode and show the value of ARR, CRR, and PSC registers.