



MICROCHIP

Regional Training Centers

Section 11

Output Compare PWM

What's PWM

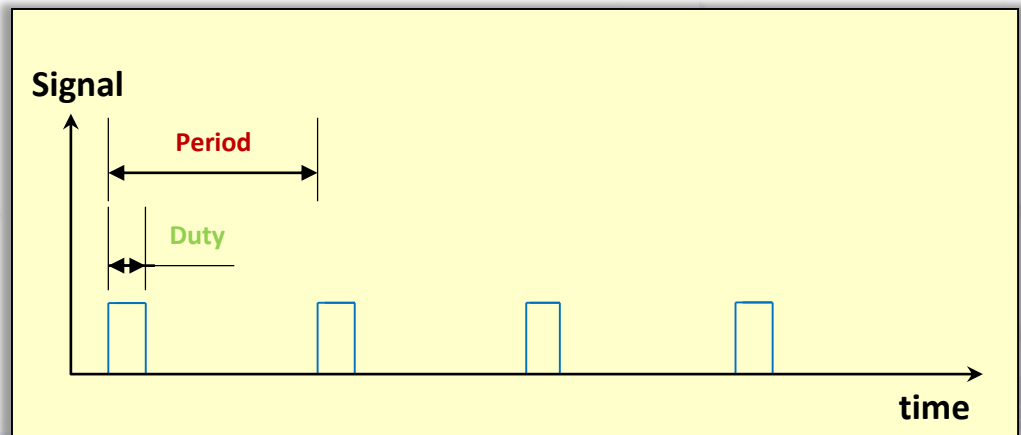
- The Pulse-width modulation (PWM) Waveform is a modulation technique used to encode a message into a pulsing signal.
- Its main use is to allow the control of the power supplied to electrical devices, ex. motor control, ballast, LED, H-bridge, power converters, and other types of power control applications.
- There have two important terms

- ◆ **Period :**

The single square wave duration.

- ◆ **Duty :**

The proportion of active time to the regular interval



PIC24F Output Compare

- PIC24FJGB provide below modes for Output Compare Module (OCM<2:0>):

Simple Compare Match Mode:

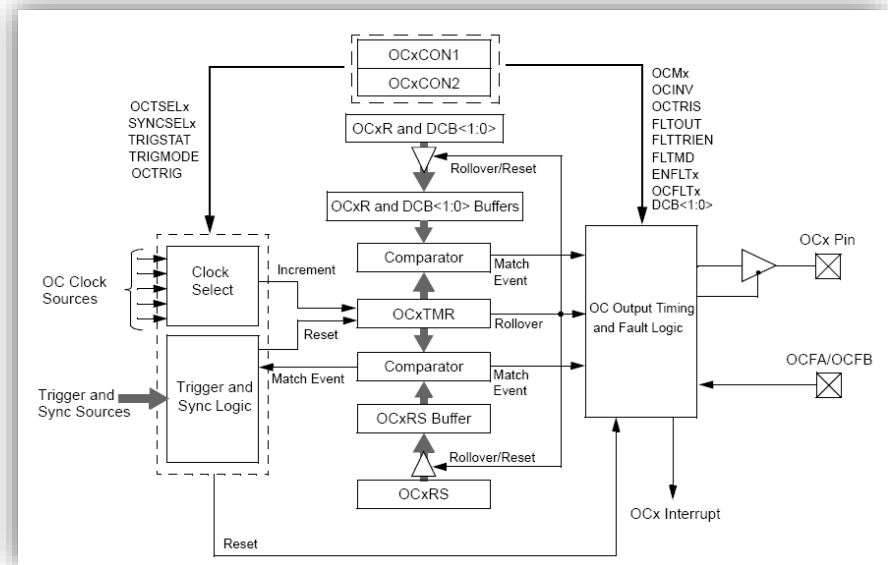
1. Normal Low, High on Match
2. Normal High, Low on Match
3. Toggle on Match

Dual Compare Mode:

1. Single Pulse
2. Continuous pulse

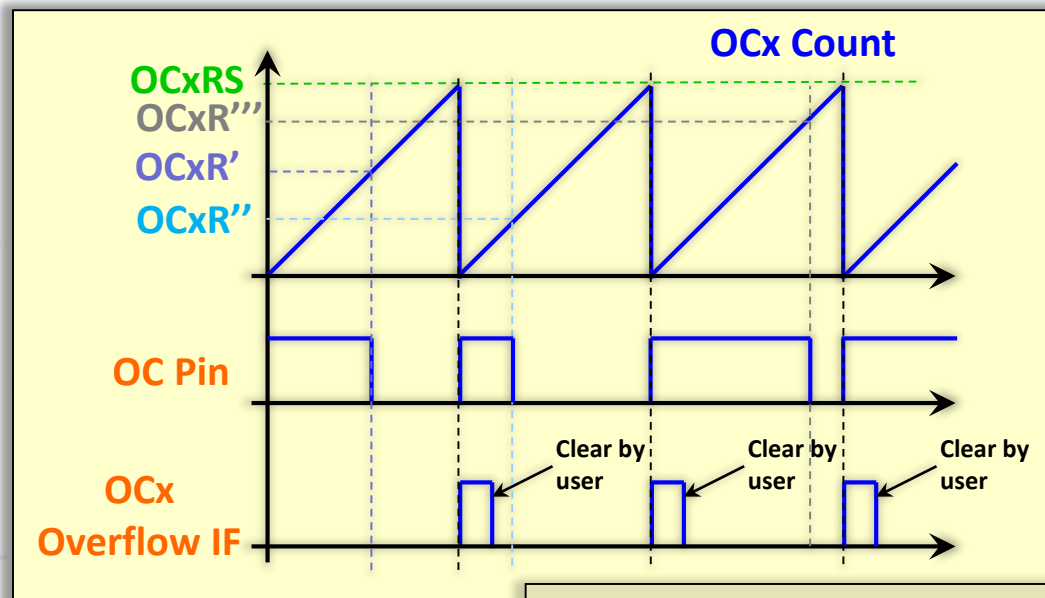
Simple PWM Mode:

1. Edge-Aligned PWM Mode
2. Center-Aligned PWM Mode



Edge-Aligned PWM Mode

- For Edge-Aligned PWM Mode, the period time (T) is controlled by the OCxRS register. OCx pin change stats on each OCx counter = OCxRS and OCxR condition.



Rule

1. OCx Counter = OCxR -> Output Set to low
2. OCx Counter = OCxRS -> Output Set to high

Lab14 Edge-Aligned PWM



Lab14 Edge-Aligned PWM

- Try to initial OC1 module to Edge-Aligned PWM Mode, than use PWM to control LED brightness.
- OC1 module clock source set to $F_{CY} (F_{osc} / 2)$, Sync Self, Frequency is 1KHz, Duty cycle 10%.
- Please connect **RG6(OC1)** to **LED(D6)** to observe LED brightness.

■ **Let's go!**

Lab14 Edge-Aligned PWM Hits

- *Primary Compare (OCxR, Duty)
- *Secondary Compare (OCxRS, Period)

The screenshot shows the 'OC1 - Editor' window with the following settings and callouts:

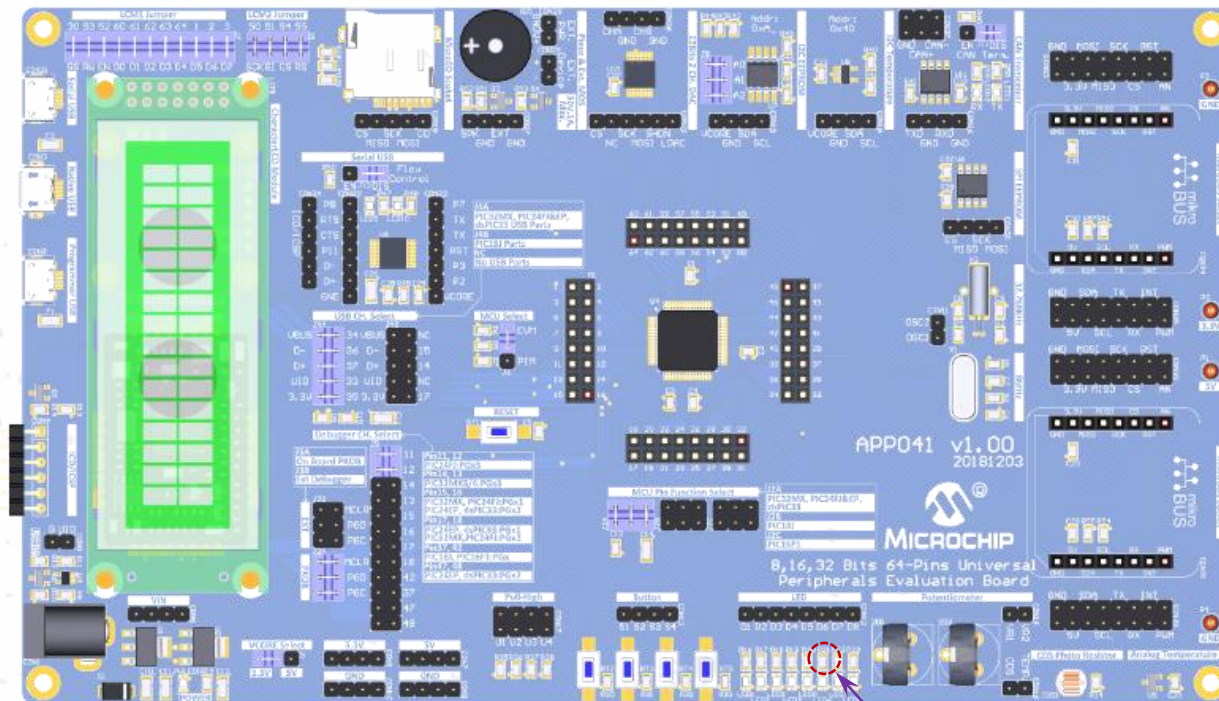
- Mode:** Points to the 'Edge-Aligned PWM mode' dropdown in the 'Hardware Setting' section.
- Clock Source:** Points to the 'FOSC/2' dropdown in the 'Clock Setting' section.
- Trigger/Sync Source:** Points to the 'Self' dropdown in the 'Trigger/Sync Setting' section.
- OCxR, Duty:** Points to the 'Primary Compare Count' field (set to 1600) in the 'Compare Inputs' section.
- OCxRS, Period:** Points to the 'Secondary Compare Count' field (set to 16000) in the 'Compare Inputs' section.

$$PWM_{Duty} = \left(\frac{OCxR}{OCxRS} \right)$$

$$PWM_{freq.} = \left(\frac{Clock\ Source}{OCxRS + 1} \right)$$

Lab14 Edge-Aligned PWM

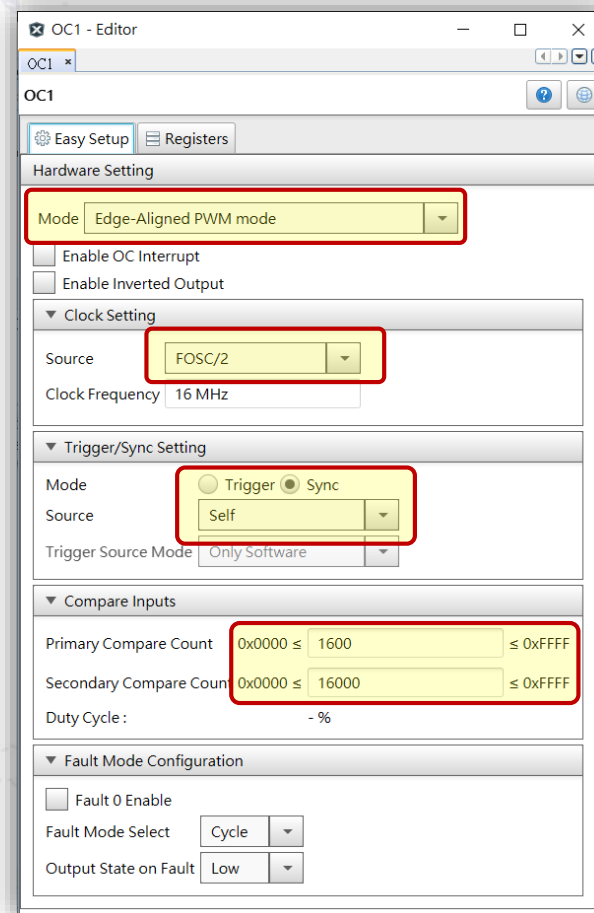
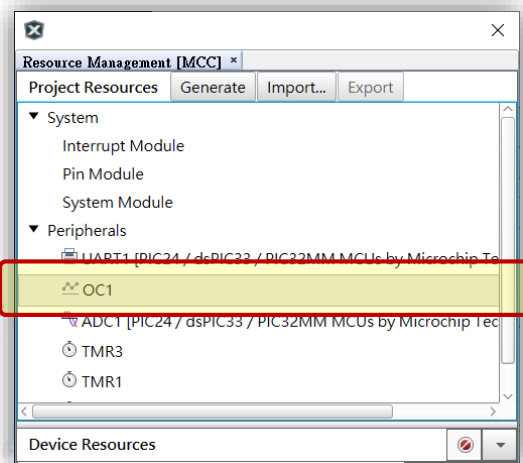
Result



Slightly bright !
(10% Duty)

Lab14 Edge-Aligned PWM

MCC's Setting & Code Example



Lab14 Edge-Aligned PWM

MCC's Setting & Code Example

Pin Manager: Grid View - Editor

Pin Manager: Grid View

Module	Function	Direction	Port D ▼												Port E ▼								Port F ▼					Port G ▼									
			0	1	2	3	4	5	6	7	8	9	10	11	0	1	2	3	4	5	6	7	0	1	3	4	5	2	3	6	7	8	9				
ICD ▼	PGCx	input																																			
	PGDx	input																																			
OC1 ▼	OC1	output	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙																									
	OCFA	input	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙																									
Pin Module ▼	GPIO	input	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙			
	GPIO	output	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙	⚙																									
TMR1	T1CK	input																																			
TMR2	T2CK	input	⚙	⚙	⚙	⚙	⚙	⚙			⚙	⚙	⚙																								
TMR3	T3CK	input																																			
UART1 ▼	U1CTS	input	⚙	⚙	⚙	⚙	⚙	⚙			⚙	⚙	⚙																								
	U1RTS	output																																			
	U1RX	input	⚙					⚙			⚙	⚙	⚙																								
	U1TX	output	⚙	⚙	⚙	⚙	⚙	⚙			⚙	⚙	⚙																								

MCC's O.C.PWM Function

- MCC Provide below common functions:

void OC1_Initialize(void)

// Initial OC.

void OC1_Start(void); void OC1_Stop(void);

// OC Enable & Disable.

void OC1_SingleCompareValueSet(value)

// Single Compare Mode Setting (OC1R <- value)

void OC1_DualCompareValueSet(priVal, secVal)

// Dual Compare Mode Setting (OC1R <- priVal, OC1RS <- secVal)

void OC1_CentreAlignedPWMConfig(priVal, secVal)

// Centre Aligned Mode Setting (OC1R <- priVal, OC1RS <- secVal)

void OC1_EdgeAlignedPWMConfig(priVal, secVal)

void OC1_PrimaryValueSet(uint16_t priVal)

void OC1_SecondaryValueSet(uint16_t secVal)

// Edge Aligned PWM Mode Setting (OC1R <- priVal, OC1RS <- secVal)

100% Duty Issue

How to generate 100% duty PWM ?

Rule Review

1. OCx Counter = OCxR -> Output Set to low
2. OCx Counter = OCxRS -> Output Set to high

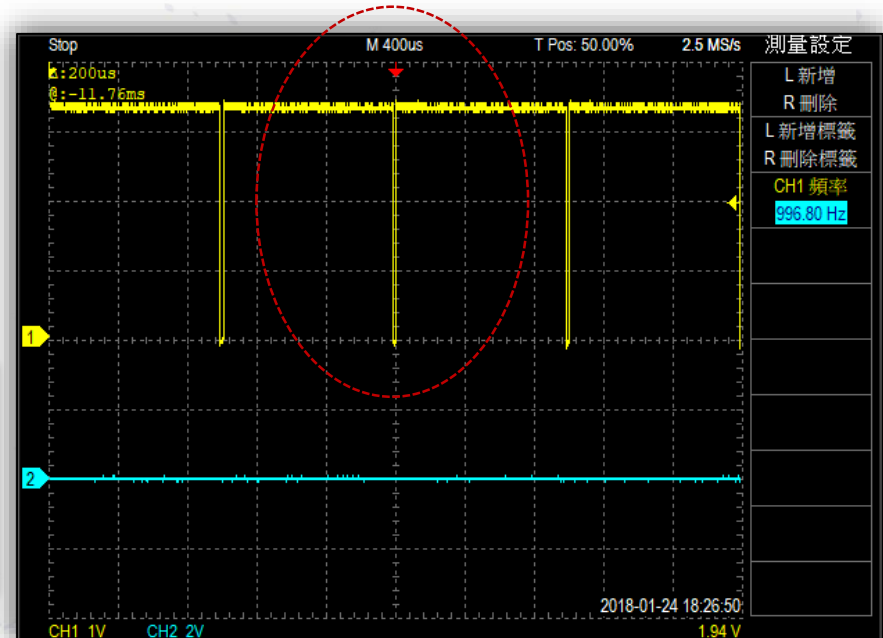
100% is means OCxR = OCxRS must be set for 100% duty output. Which rule be executed ?

Result : Glitch !

To prevent this issue

set **OCxR** > **OCxRS** for 100% duty.

```
if( DutyVlaue >= OCxRS ) /* 100% duty */  
    OC1_PrimaryValueSet(OCxRS+ 1);  
else  
    OC1_PrimaryValueSet(DutyVlaue);
```



Lab15 PWM ADC Duty Adj



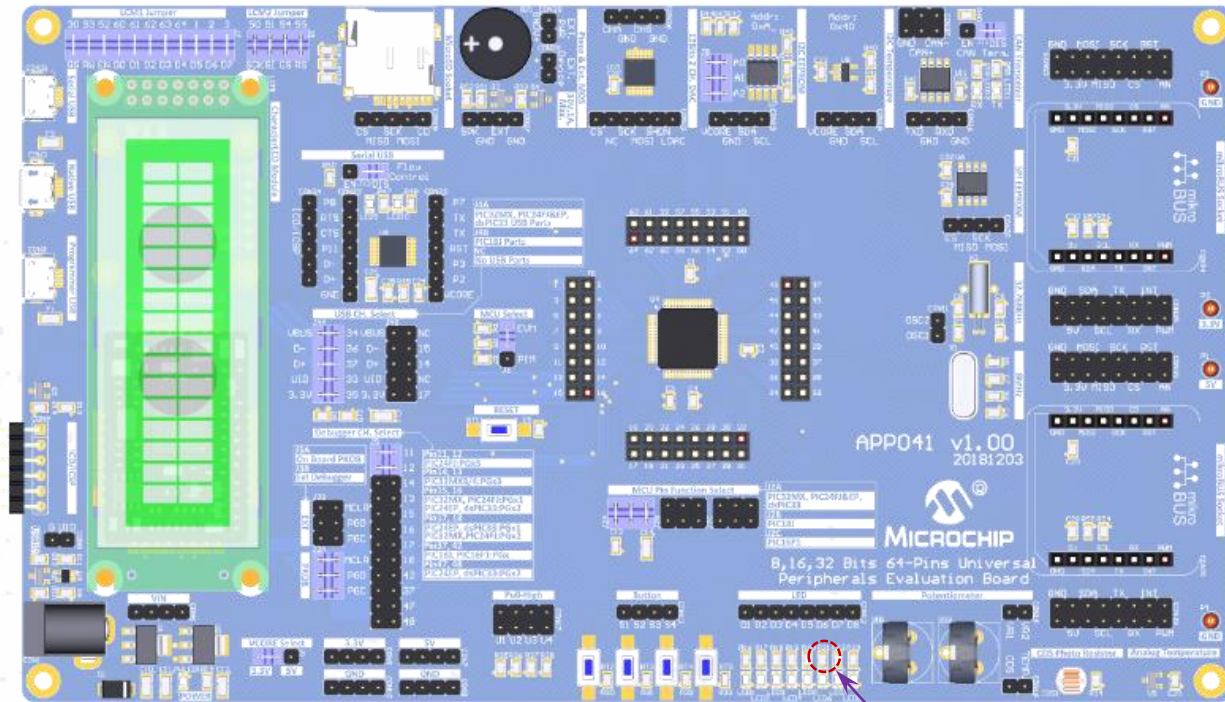
Lab15 PWM ADC Duty Adj

- ◆ Try to use **Potentiometer(VR1)** to control **LED(D6)** brightness.
- ◆ VR Value(0 ~1023) <-> PWM Duty (0% ~ 100%)
- ◆ OC1 Setting same as lab14.

◆ **Let's go!**

Lab15 PWM ADC Duty Adj

Result




dimming Up & Down by VR1

Lab15 PWM ADC Duty Adj

MCC's Setting & Code Example

```
int main(void)
{
    ...
    while (1)
    {
        if (AD1Flag)
        {
            AD1Flag = 0;
            ...

            if (ADCResult[0] >= 1023)
                OC1_PrimaryValueSet(OC1RS + 1);
            else
                OC1_PrimaryValueSet(((long) ADCResult[0] * OC1RS) / 1023);
        }
    }
}
```



```
if( DutyVlaue >= OCxRS ) /* 100% duty */
    OC1_PrimaryValueSet(OCxRS+ 1);
else
    OC1_PrimaryValueSet(DutyVlaue);
```

OC's Duty & Period update

- OC's duty (OCxR) can update at any time, hardware will auto reload the new value at new PWM cycle moment.
- OC's period (OCxRS) only allow update at new PWM cycle moment.
- **Enable OC interrupt more easy to update PWM period.**

- For example:

```
void OC2_CallBack(void)
{
    OC2_SecondaryValueSet(NewPeriodValue);
    OC2_PrimaryValueSet(OC2RS >> 1); /* 50% Duty */
    ...;
}
```

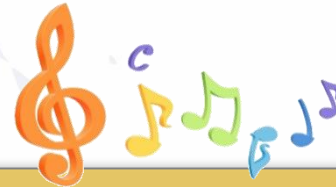
Lab16 PWM Buzzer Tone



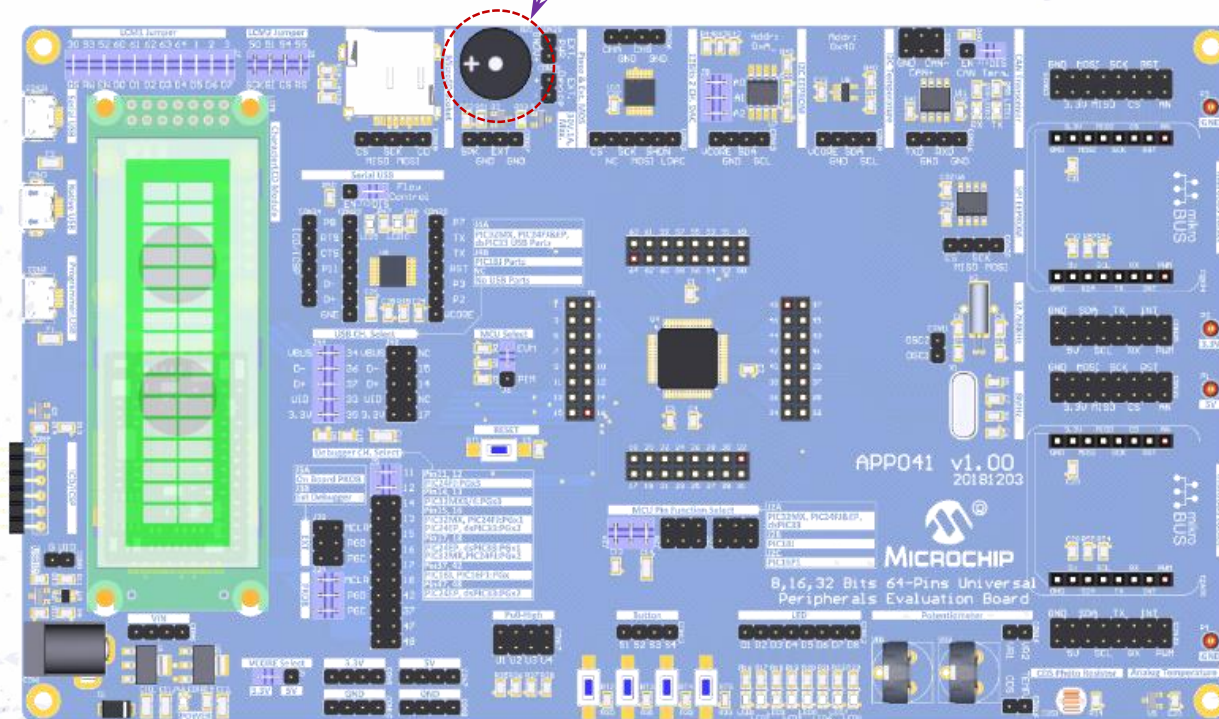
Lab16 PWM Buzzer Tone

- Try to initial OC2 to export special frequency (Tone) to Buzzer, change the tone every second.
 - `Tone[] = {262,294, 330, 349, 392, 440, 494}; /*Do, Re, Mi, etc.. */`
- Enable OC2 interrupt, priority set to **7** (highest).
Try to change OC2 frequency at OC2 callback function.
- Please connect **RG9(OC2)** to **Buzzer(SPK)**.
- Let's go!**

Result



Tone by your code



Lab16 PWM Buzzer Tone

MCC's Setting & Code Example

The image displays the Microchip Configuration Studio (MCC) interface with two windows open: 'OC2 - Editor' and 'Pin Manager: Grid View - Editor'.

OC2 - Editor Window:

- Mode:** Edge-Aligned PWM mode
- Enable OC Interrupt:** Checked
- Enable Inverted Output:** Unchecked
- Clock Setting:**
 - Source: FOSC/2
 - Clock Frequency: 16 MHz
- Trigger/Sync Setting:**
 - Mode: Trigger (selected), Sync
 - Source: Self
 - Trigger Source Mode: Only Software
- Compare Inputs:**
 - Primary Compare Count: 0x0000 ≤ 1600 ≤ 0xFFFF
 - Secondary Compare Count: 0x0000 ≤ 16000 ≤ 0xFFFF
 - Duty Cycle: - %
- Fault Mode Configuration:**
 - Fault 0 Enable: Unchecked
 - Fault Mode Select: Cycle
 - Output State on Fault: Low

Pin Manager: Grid View - Editor Window:

This window shows a grid of pin configurations for various modules. The modules listed are OC1, OC2, GPIO, TMR1, TMR2, TMR3, UART1, and UART2. The pins are organized by port (D, E, F, G). The OC2 module is highlighted in yellow, and its pins (OC2A, OC2B, OC2C, OC2D) are also highlighted in yellow.

Module Pin Configuration Table:

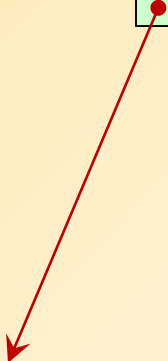
Module	Function	Direction	Port D	Port E	Port F	Port G
OC1	OC1A	output	0	1	2	3
OC2	OC2A	output	4	5	6	7
OC2	OC2B	output	8	9	10	11
OC2	OC2C	output	12	13	14	15
OC2	OC2D	output	16	17	18	19
GPIO	GPIO0	input	0	1	2	3
GPIO	GPIO1	input	4	5	6	7
GPIO	GPIO2	input	8	9	10	11
GPIO	GPIO3	input	12	13	14	15
GPIO	GPIO4	input	16	17	18	19
GPIO	GPIO5	input	20	21	22	23
GPIO	GPIO6	input	24	25	26	27
GPIO	GPIO7	input	28	29	30	31
GPIO	GPIO8	input	32	33	34	35
GPIO	GPIO9	input	36	37	38	39
GPIO	GPIO10	input	40	41	42	43
GPIO	GPIO11	input	44	45	46	47
GPIO	GPIO12	input	48	49	50	51
GPIO	GPIO13	input	52	53	54	55
GPIO	GPIO14	input	56	57	58	59
GPIO	GPIO15	input	60	61	62	63
GPIO	GPIO16	input	64	65	66	67
GPIO	GPIO17	input	68	69	70	71
GPIO	GPIO18	input	72	73	74	75
GPIO	GPIO19	input	76	77	78	79
GPIO	GPIO20	input	80	81	82	83
GPIO	GPIO21	input	84	85	86	87
GPIO	GPIO22	input	88	89	90	91
GPIO	GPIO23	input	92	93	94	95
GPIO	GPIO24	input	96	97	98	99
GPIO	GPIO25	input	100	101	102	103
GPIO	GPIO26	input	104	105	106	107
GPIO	GPIO27	input	108	109	110	111
GPIO	GPIO28	input	112	113	114	115
GPIO	GPIO29	input	116	117	118	119
GPIO	GPIO30	input	120	121	122	123
GPIO	GPIO31	input	124	125	126	127
GPIO	GPIO32	input	128	129	130	131
GPIO	GPIO33	input	132	133	134	135
GPIO	GPIO34	input	136	137	138	139
GPIO	GPIO35	input	140	141	142	143
GPIO	GPIO36	input	144	145	146	147
GPIO	GPIO37	input	148	149	150	151
GPIO	GPIO38	input	152	153	154	155
GPIO	GPIO39	input	156	157	158	159
GPIO	GPIO40	input	160	161	162	163
GPIO	GPIO41	input	164	165	166	167
GPIO	GPIO42	input	168	169	170	171
GPIO	GPIO43	input	172	173	174	175
GPIO	GPIO44	input	176	177	178	179
GPIO	GPIO45	input	180	181	182	183
GPIO	GPIO46	input	184	185	186	187
GPIO	GPIO47	input	188	189	190	191
GPIO	GPIO48	input	192	193	194	195
GPIO	GPIO49	input	196	197	198	199
GPIO	GPIO50	input	200	201	202	203
GPIO	GPIO51	input	204	205	206	207
GPIO	GPIO52	input	208	209	210	211
GPIO	GPIO53	input	212	213	214	215
GPIO	GPIO54	input	216	217	218	219
GPIO	GPIO55	input	220	221	222	223
GPIO	GPIO56	input	224	225	226	227
GPIO	GPIO57	input	228	229	230	231
GPIO	GPIO58	input	232	233	234	235
GPIO	GPIO59	input	236	237	238	239
GPIO	GPIO60	input	240	241	242	243
GPIO	GPIO61	input	244	245	246	247
GPIO	GPIO62	input	248	249	250	251
GPIO	GPIO63	input	252	253	254	255
GPIO	GPIO64	input	256	257	258	259
GPIO	GPIO65	input	260	261	262	263
GPIO	GPIO66	input	264	265	266	267
GPIO	GPIO67	input	268	269	270	271
GPIO	GPIO68	input	272	273	274	275
GPIO	GPIO69	input	276	277	278	279
GPIO	GPIO70	input	280	281	282	283
GPIO	GPIO71	input	284	285	286	287
GPIO	GPIO72	input	288	289	290	291
GPIO	GPIO73	input	292	293	294	295
GPIO	GPIO74	input	296	297	298	299
GPIO	GPIO75	input	300	301	302	303
GPIO	GPIO76	input	304	305	306	307
GPIO	GPIO77	input	308	309	310	311
GPIO	GPIO78	input	312	313	314	315
GPIO	GPIO79	input	316	317	318	319
GPIO	GPIO80	input	320	321	322	323
GPIO	GPIO81	input	324	325	326	327
GPIO	GPIO82	input	328	329	330	331
GPIO	GPIO83	input	332	333	334	335
GPIO	GPIO84	input	336	337	338	339
GPIO	GPIO85	input	340	341	342	343
GPIO	GPIO86	input	344	345	346	347
GPIO	GPIO87	input	348	349	350	351
GPIO	GPIO88	input	352	353	354	355
GPIO	GPIO89	input	356	357	358	359
GPIO	GPIO90	input	360	361	362	363
GPIO	GPIO91	input	364	365	366	367
GPIO	GPIO92	input	368	369	370	371
GPIO	GPIO93	input	372	373	374	375
GPIO	GPIO94	input	376	377	378	379
GPIO	GPIO95	input	380	381	382	383
GPIO	GPIO96	input	384	385	386	387
GPIO	GPIO97	input	388	389	390	391
GPIO	GPIO98	input	392	393	394	395
GPIO	GPIO99	input	396	397	398	399
GPIO	GPIO100	input	400	401	402	403
GPIO	GPIO101	input	404	405	406	407
GPIO	GPIO102	input	408	409	410	411
GPIO	GPIO103	input	412	413	414	415
GPIO	GPIO104	input	416	417	418	419
GPIO	GPIO105	input	420	421	422	423
GPIO	GPIO106	input	424	425	426	427
GPIO	GPIO107	input	428	429	430	431
GPIO	GPIO108	input	432	433	434	435
GPIO	GPIO109	input	436	437	438	439
GPIO	GPIO110	input	440	441	442	443
GPIO	GPIO111	input	444	445	446	447
GPIO	GPIO112	input	448	449	450	451
GPIO	GPIO113	input	452	453	454	455
GPIO	GPIO114	input	456	457	458	459
GPIO	GPIO115	input	460	461	462	463
GPIO	GPIO116	input	464	465	466	467
GPIO	GPIO117	input	468	469	470	471
GPIO	GPIO118	input	472	473	474	475
GPIO	GPIO119	input	476	477	478	479
GPIO	GPIO120	input	480	481	482	483
GPIO	GPIO121	input	484	485	486	487
GPIO	GPIO122	input	488	489	490	491
GPIO	GPIO123	input	492	493	494	495
GPIO	GPIO124	input	496	497	498	499
GPIO	GPIO125	input	500	501	502	503
GPIO	GPIO126	input	504	505	506	507
GPIO	GPIO127	input	508	509	510	511
GPIO	GPIO128	input	512	513	514	515
GPIO	GPIO129	input	516	517	518	519
GPIO	GPIO130	input	520	521	522	523
GPIO	GPIO131	input	524	525	526	527
GPIO	GPIO132	input	528	529	530	531
GPIO	GPIO133	input	532	533	534	535
GPIO	GPIO134	input	536	537	538	539
GPIO	GPIO135	input	540	541	542	543
GPIO	GPIO136	input	544	545	546	547
GPIO	GPIO137	input	548	549	550	551
GPIO	GPIO138	input	552	553	554	555
GPIO	GPIO139	input	556	557	558	559
GPIO	GPIO140	input	560	561	562	563
GPIO	GPIO141	input	564	565	566	567
GPIO	GPIO142	input	568	569	570	571
GPIO	GPIO143	input	572	573	574	575
GPIO	GPIO144	input	576	577	578	579
GPIO	GPIO145	input	580	581	582	583
GPIO	GPIO146	input	584	585	586	587
GPIO	GPIO147	input	588	589	590	591
GPIO	GPIO148	input	592	593	594	595
GPIO	GPIO149	input	596	597	598	599
GPIO	GPIO150	input	600	601	602	603
GPIO	GPIO151	input	604	605	606	607
GPIO	GPIO152	input	608	609	610	611
GPIO	GPIO153	input	612	613	614	615
GPIO	GPIO154	input	616	617	618	619
GPIO	GPIO155	input	620	621	622	623
GPIO	GPIO156	input	624	625	626	627
GPIO	GPIO157	input	628	629	630	631
GPIO	GPIO158	input	632	633	634	635
GPIO	GPIO159	input	636	637	638	639
GPIO	GPIO160	input	640	641	642	643
GPIO	GPIO161	input	644	645	646	647
GPIO	GPIO162	input	648	649	650	651
GPIO	GPIO163	input	652	653	654	655
GPIO	GPIO164	input	656	657	658	659
GPIO	GPIO165	input	660	661	662	663
GPIO	GPIO166	input	664	665	666	667
GPIO	GPIO167	input	668	669	670	671
GPIO	GPIO168	input	672	673	674	675
GPIO	GPIO169	input	676	677	678	679
GPIO	GPIO170	input	680	681	682	683
GPIO	GPIO171	input	684	685	686	687
GPIO	GPIO172	input	688	689	690	691
GPIO	GPIO173	input	692	693	694	695
GPIO	GPIO174	input	696	697	698	699
GPIO	GPIO175	input	700	701	702	703
GPIO	GPIO176	input	704	705	706	707
GPIO	GPIO177	input	708	709	710	711
GPIO	GPIO178	input	712	713	714	715
GPIO	GPIO179	input	716	717	718	719
GPIO	GPIO180	input	720	721	722	723
GPIO	GPIO181	input	724	725	726	727
GPIO	GPIO182	input	728	729	730	731
GPIO	GPIO183	input	732	733	734	735
GPIO	GPIO184	input	736	737	738	739
GPIO	GPIO185	input	740	741	742	743
GPIO	GPIO186	input	744	745	746	747
GPIO	GPIO187	input	748	749	750	751
GPIO	GPIO188	input	752	753	754	755
GPIO	GPIO189	input	756	757	758	759
GPIO	GPIO190	input	760	761	762	763
GPIO	GPIO191	input	764	765	766	767
GPIO	GPIO192	input	768	769	770	771
GPIO	GPIO193	input	772	773	774	775
GPIO	GPIO194	input	776	777	778	779
GPIO	GPIO195	input	780	781	782	783
GPIO	GPIO196	input	784	785	786	787
GPIO	GPIO197	input	788	789	790	791
GPIO	GPIO198	input	792	793	794	795
GPIO	GPIO199	input	796	797	798	799
GPIO	GPIO200	input	800	801	802	803
GPIO	GPIO201	input	804	805	806	807
GPIO	GPIO202	input	808	809	810	811
GPIO	GPIO203	input	812	813	814	815
GPIO	GPIO204	input	816	817	818	819
GPIO	GPIO205	input	820	821	822	823

Lab16 PWM Buzzer Tone

MCC's Setting & Code Example

```
const unsigned int Tone[] = {262, 294, 330, 349, 392, 440, 494};
volatile unsigned char T2Flag = 0;
int main(void)
{
    ...
    while (1)
    {
        ...
    }
}
void TMR2_CallBack(void)
{
    ...
    T2Flag = 1;
}
void OC2_CallBack(void)
{
    static unsigned char i = 0;
    if (T2Flag)
    {
        T2Flag = 0;

        OC2_SecondaryValueSet(16000000 / Tone[i]);
        OC2_PrimaryValueSet(OC2RS >> 1);
        if (++i > 6)
            i = 0;
    }
}
```

$$PWM_{freq.} = \left(\frac{\text{Clock Source}}{OCxRS + 1} \right)$$




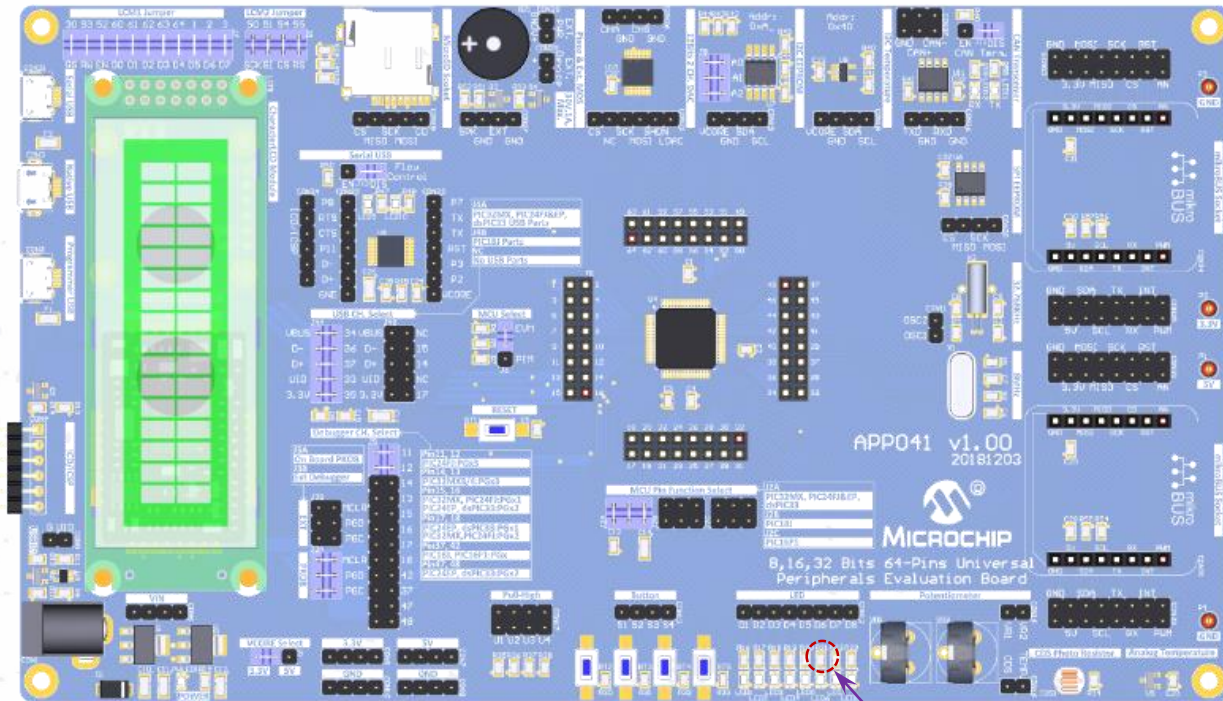
Lab17 PWM Auto Duty Adj



Lab17 PWM Auto Duty Adj

- ◆ Try to use timer to control OC1's PWM duty, automatically.
- ◆ Duty 0% -> Duty 100% -> Duty 0% -> ...
- ◆ Change duty one time per **50ms**.
(2% - 5% duty change every time suggest)

Result



**dimming up and down
automatically**

Lab17 PWM Auto Duty Adj

MCC's Setting & Code Example

TMR4 - Editor

TMR4

Easy Setup | Registers

Hardware Settings

☒ Enable TMR ☐ Enable Gate

Timer Clock

Clock Source: FOSC/2

Input Frequ...: 16 MHz

Prescaler: 1:64

Bit Mode ☐ 32 Bit ☒ 16 Bit

Timer Period

Period Count: 0x0 ≤ 0x30D4 ≤ 0xFFFF

Timer Period 4 us ≤ 50 ms ≤ 262.14 ms

Calculated Period: 50 ms

☒ Enable Timer Interrupt

Software Settings

Callback Function Rate: 0x1 xTimer Period = 50 ms

Interrupt Module - Editor

Interrupt Module

Easy Setup

Interrupt Manager

Module	Interrupt	Description	IRQ Nu...	Enabled	Priority
Pin Module	CNI	CN - Change Notifica...	19	<input type="checkbox"/>	1
ADC1	ADI	ADC1 - A/D Converte...	13	<input checked="" type="checkbox"/>	5
UART1	UERI	U1E - UART1 Error	65	<input checked="" type="checkbox"/>	6
UART1	UTXI	U1TX - UART1 Trans...	12	<input checked="" type="checkbox"/>	6
UART1	URXI	U1RX - UART1 Receiver	11	<input checked="" type="checkbox"/>	6
OC1	OCI	OC1 - Output Compa...	2	<input type="checkbox"/>	1
TMR4	TI	T4 - Timer4	27	<input checked="" type="checkbox"/>	1
TMR4	TNI	T5 - Timer5	28	<input type="checkbox"/>	1
TMR3	TI	T3 - Timer3	8	<input type="checkbox"/>	1
TMR2	TI	T2 - Timer2	7	<input checked="" type="checkbox"/>	3
TMR2	TNI	T3 - Timer3	8	<input type="checkbox"/>	1
TMR1	TI	T1 - Timer1	3	<input checked="" type="checkbox"/>	4
OC2	OCI	OC2 - Output Compa...	6	<input checked="" type="checkbox"/>	7

Lab17 PWM Auto Duty Adj

MCC's Setting & Code Example

```
volatile unsigned char T4Flag = 0;
unsigned int Duty = 50;
char DutyDistance = 2;
int main(void)
{
    ...
    while (1)
    {
        if (T4Flag)
        {
            T4Flag = 0;

            Duty += DutyDistance;

            if (Duty >= 100 || Duty <= 0)
                DutyDistance = -DutyDistance;

            if (Duty >= 100)
                OC1_PrimaryValueSet(OC1RS + 1);
            else
                OC1_PrimaryValueSet(((long) Duty * OC1RS) / 100);
        }
        ...
    }
}

void TMR4_CallBack(void)
{
    T4Flag = 1;
}
```