

TI-RSLK

Texas Instruments Robotics System Learning Kit



TEXAS INSTRUMENTS

Module 13

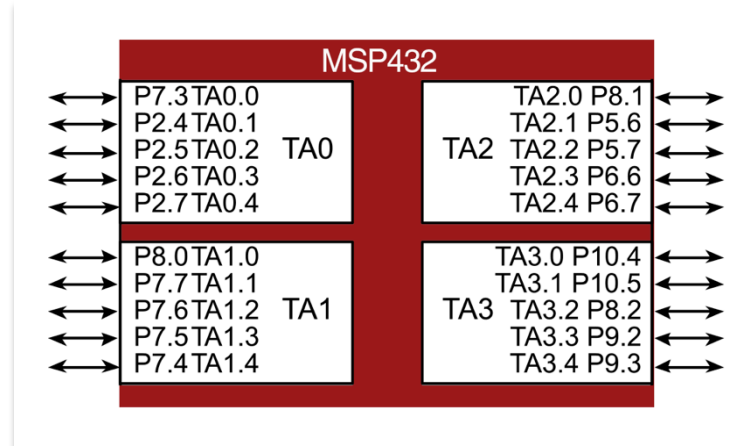
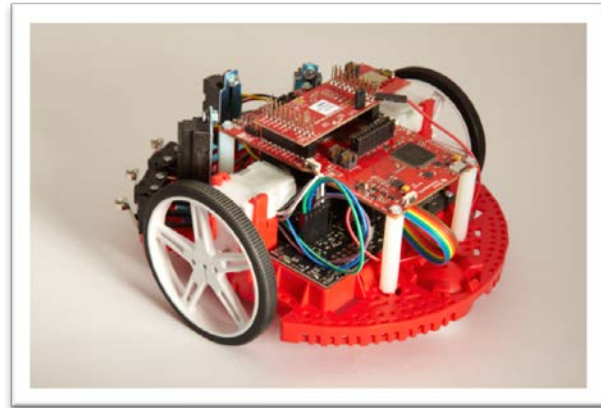
Lecture: Timers – Periodic Interrupts



Periodic interrupts using Timers

You will learn in this module

- Timer A on the TI MSP432
 - Clock input, prescale
 - Counter
- Periodic interrupts
 - Additional threads
 - Priority

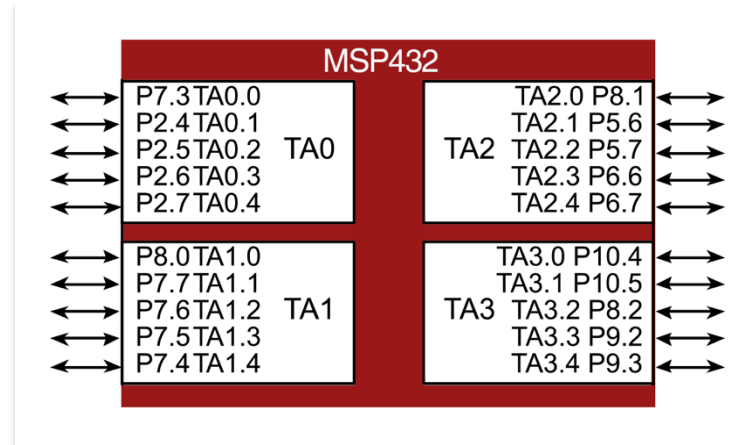




Timer - A for input and output

Timer A Features

- Pins
 - Input capture
 - Output compare
- Precision
 - 16-bits
- Resolution
 - Clock period
 - Prescale



$$SMCLK = 48\text{MHz}/4 = 12 \text{ MHz}, 83.33\text{ns}$$

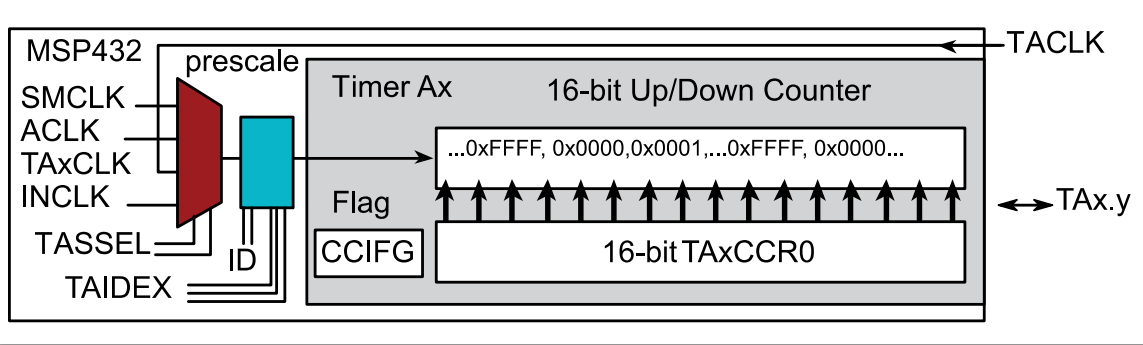


Timer - A registers

	15-10	9-8	7-6	5-4	3	2	1	0	Name				
0.0000		TASSEL	ID	MC		TACLRL	TAIE	TAIFG	TA0CTL				
	15-14	13-12	11	10	9	8	7-5	4	3	2	1	0	
0.0002	CM	CCIS	SCS	SCCI		CAP	OUTMOD	CCIE	CCI	OUT	COV	CCIFG	TA0CCTL0
0.0004	CM	CCIS	SCS	SCCI		CAP	OUTMOD	CCIE	CCI	OUT	COV	CCIFG	TA0CCTL1
0.0006	CM	CCIS	SCS	SCCI		CAP	OUTMOD	CCIE	CCI	OUT	COV	CCIFG	TA0CCTL2
0.0008	CM	CCIS	SCS	SCCI		CAP	OUTMOD	CCIE	CCI	OUT	COV	CCIFG	TA0CCTL3
0.000A	CM	CCIS	SCS	SCCI		CAP	OUTMOD	CCIE	CCI	OUT	COV	CCIFG	TA0CCTL4
0.000C	CM	CCIS	SCS	SCCI		CAP	OUTMOD	CCIE	CCI	OUT	COV	CCIFG	TA0CCTL5
0.000E	CM	CCIS	SCS	SCCI		CAP	OUTMOD	CCIE	CCI	OUT	COV	CCIFG	TA0CCTL6
	15-0												
0.0010	16-bit counter												TA0R
0.0012	16-bit Capture/Compare 0 Register												TA0CCR0
0.0014	16-bit Capture/Compare 1 Register												TA0CCR1
0.0016	16-bit Capture/Compare 2 Register												TA0CCR2
0.0018	16-bit Capture/Compare 3 Register												TA0CCR3
0.001A	16-bit Capture/Compare 4 Register												TA0CCR4
0.001C	16-bit Capture/Compare 5 Register												TA0CCR5
0.001E	16-bit Capture/Compare 6 Register												TA0CCR6
	15-3										2-0		
0.0020											TAIDEX		TA0EX0
	15-0												
0.002E	TAIV												TA0IV



Clock and prescale



TASSEL	Selected Clock
00	TAxCLK
01	ACLK
10	SMCLK
11	INCLK

ID	Prescale
00	/1
01	/2
10	/4
11	/8

Resolution = $T * 2^{ID} * (TAIDEX+1)$
 Range = Precision * Resolution



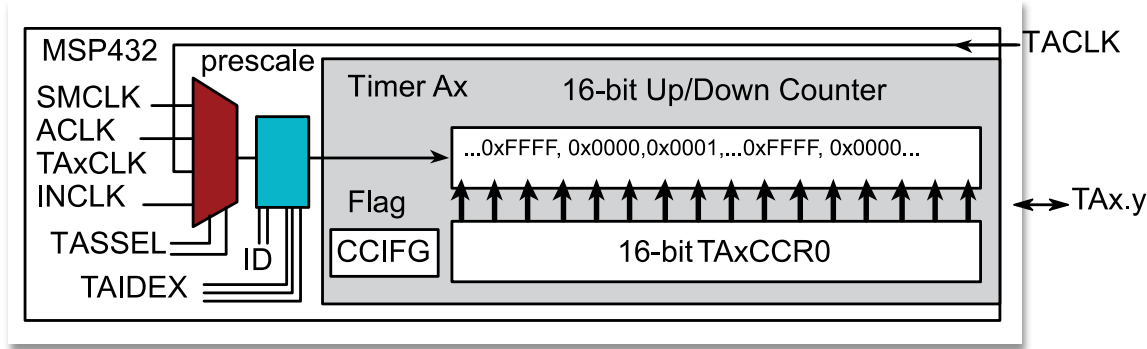
Timer - A Modes

MC	Mode control
00	Stop
01	Up mode: Timer counts up to TAXCCR0
10	Continuous mode: Timer counts up to 0xFFFF
11	Up/down mode: Timer counts up to TAXCCR0 then down to 0x0000

OUTMOD	On match to TAXCCRy	On match to TAXCCR0
000	OUT bit value	
001	Set	
010	Toggle	Reset
011	Set	Reset
100	Toggle	
101	Reset	
110	Toggle	Set
111	Reset	Set



Clock and prescale



- 0) Halt the timer (MC=00),
- 1) Set the timer clock and prescale,
- 2) Set submodule 0 to compare, arm interrupt
- 3) Set the TAXCCR0 to the interrupt period minus 1
- 4) Set the priority in the correct NVIC Priority register
- 5) Enable the interrupt in the NVIC Interrupt Enable register
- 6) Reset the timer and start it in up mode
- 7) Enable interrupts (in the main program after all devices initialized)



Interrupt Vectors, numbers, names, and priority

Vector	Number	IRQ	ISR name	NVIC priority	Priority
0x0000002C	11	-5	SVC_Handler	SCB_SHPR2	31 - 29
0x00000038	14	-2	PendSV_Handler	SCB_SHPR3	23 - 21
0x0000003C	15	-1	SysTick_Handler	SCB_SHPR3	31 - 29
0x00000060	24	8	TA0_0_IRQHandler	NVIC_IPR2	7 - 5
0x00000064	25	9	TA0_N_IRQHandler	NVIC_IPR2	15 - 13
0x00000068	26	10	TA1_0_IRQHandler	NVIC_IPR2	23 - 21
0x0000006C	27	11	TA1_N_IRQHandler	NVIC_IPR2	31 - 29
0x00000070	28	12	TA2_0_IRQHandler	NVIC_IPR3	7 - 5
0x00000074	29	13	TA2_N_IRQHandler	NVIC_IPR3	15 - 13
0x00000078	30	14	TA3_0_IRQHandler	NVIC_IPR3	7 - 5
0x0000007C	31	15	TA3_N_IRQHandler	NVIC_IPR3	15 - 13
0x00000080	32	16	EUSCIA0_IRQHandler	NVIC_IPR4	23 - 21
0x00000084	33	17	EUSCIA1_IRQHandler	NVIC_IPR4	31 - 29
0x00000088	34	18	EUSCIA2_IRQHandler	NVIC_IPR4	7 - 5
0x0000008C	35	19	EUSCIA3_IRQHandler	NVIC_IPR4	15 - 13
0x00000090	36	20	EUSCIB0_IRQHandler	NVIC_IPR5	23 - 21
0x00000094	37	21	EUSCIB1_IRQHandler	NVIC_IPR5	31 - 29
0x00000098	38	22	EUSCIB2_IRQHandler	NVIC_IPR5	7 - 5
0x0000009C	39	23	EUSCIB3_IRQHandler	NVIC_IPR5	15 - 13
0x000000CC	51	35	PORT1_IRQHandler	NVIC_IPR8	23 - 21
0x000000D0	52	36	PORT2_IRQHandler	NVIC_IPR9	31 - 29
0x000000D4	53	37	PORT3_IRQHandler	NVIC_IPR9	7 - 5
0x000000D8	54	38	PORT4_IRQHandler	NVIC_IPR9	15 - 13
0x000000DC	55	39	PORT5_IRQHandler	NVIC_IPR9	23 - 21
0x000000E0	56	40	PORT6_IRQHandler	NVIC_IPR10	31 - 29
					7 - 5

```
void TA2_0_IRQHandler(void){
    TIMER_A2->CCTL[0] &= ~0x0001; // ack
    // body
}
```

Look for `interruptVectors[]`
in the file `startup_msp432p401r_ccs.c`



Periodic Interrupt

```
void (*TimerA2Task)(void); // user function
void TimerA2_Init(void(*task)(void), uint16_t period){
    TimerA2Task = task; // user function
    // bits9-8=10, clock source to SMCLK
    // bits7-6=10, input clock divider /4
    // bits5-4=00, stop mode
    // bit2=0, set this bit to clear
    // bit1=0, no interrupt on timer
    TIMER_A2->CTL = 0x0280;
    // bits15-14=00, no capture mode
    // bit8=0, compare mode
    // bit4=1, enable capture/compare interrupt on CCIFG
    // bit2=0, output this value in output mode 0
    // bit0=0, clear capture/compare interrupt pending
    TIMER_A2->CCTL[0] = 0x0010;
    TIMER_A2->CCR[0] = (period - 1); // compare match value
    TIMER_A2->EX0 = 0x0005; // configure for input clock divider /6
    NVIC->IP[3] = (NVIC->IP[3]&0xFFFFF00)|0x00000040; // priority 2
    NVIC->ISER[0] = 0x00001000; // enable interrupt 12 in NVIC
    TIMER_A2->CTL |= 0x0014; // reset and start Timer A in up mode
}
```

Interrupts enabled in the main program after all devices initialized

$$T * 2^{ID} * (TAIDEX+1)(TAxCCR0+1) = 2\mu s * period$$

```
void TA2_0_IRQHandler(void){
    TIMER_A2->CCTL[0] &= ~0x0001;
    *TimerA2Task();
}
```

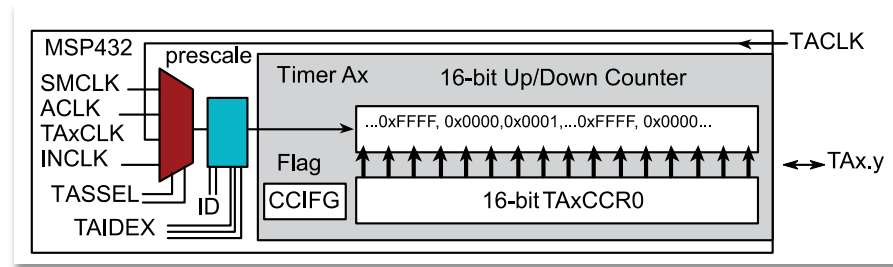
```
NVIC->IP[2]= (NVIC->IP[2]&0xFF00FFFF)|0x00400000;
NVIC->ISER[0] = 0x00000400; // enable interrupt 10 in NVIC
```



Summary

Timer overview

- Clock input
- Prescale
- Counter



Periodic interrupts

- Additional threads
- Priority

```

void TA0_0_IRQHandler(void){
    TIMER_A0->CTL[0] &= ~0x0001;
    // ack
    // body
}
void TA1_0_IRQHandler(void){
    TIMER_A1->CTL[0] &= ~0x0001;
    // ack
    // body
}
void TA2_0_IRQHandler(void){
    TIMER_A2->CTL[0] &= ~0x0001;
    // ack
    // body
}

```

Arrows point from the code blocks to labels: TimerA0.c, TimerA1.c, and TimerA2.c.

Module 13

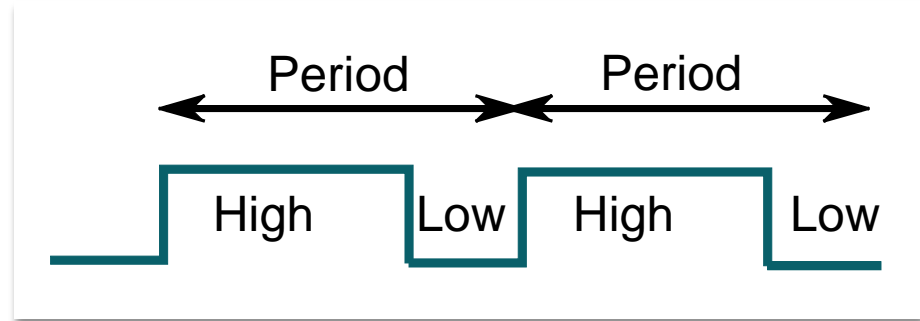
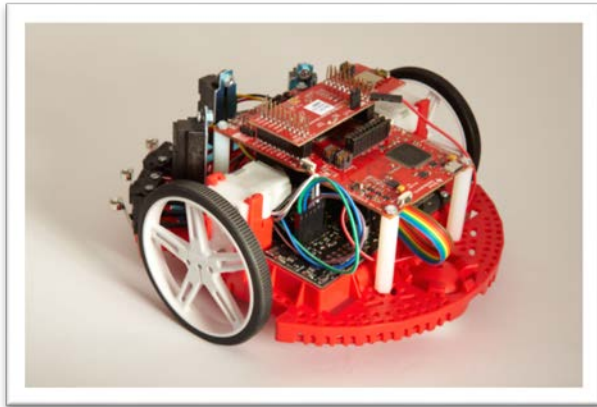
Lecture: Timers – Pulse Width Modulation



PWM using Timers

You will learn in this module

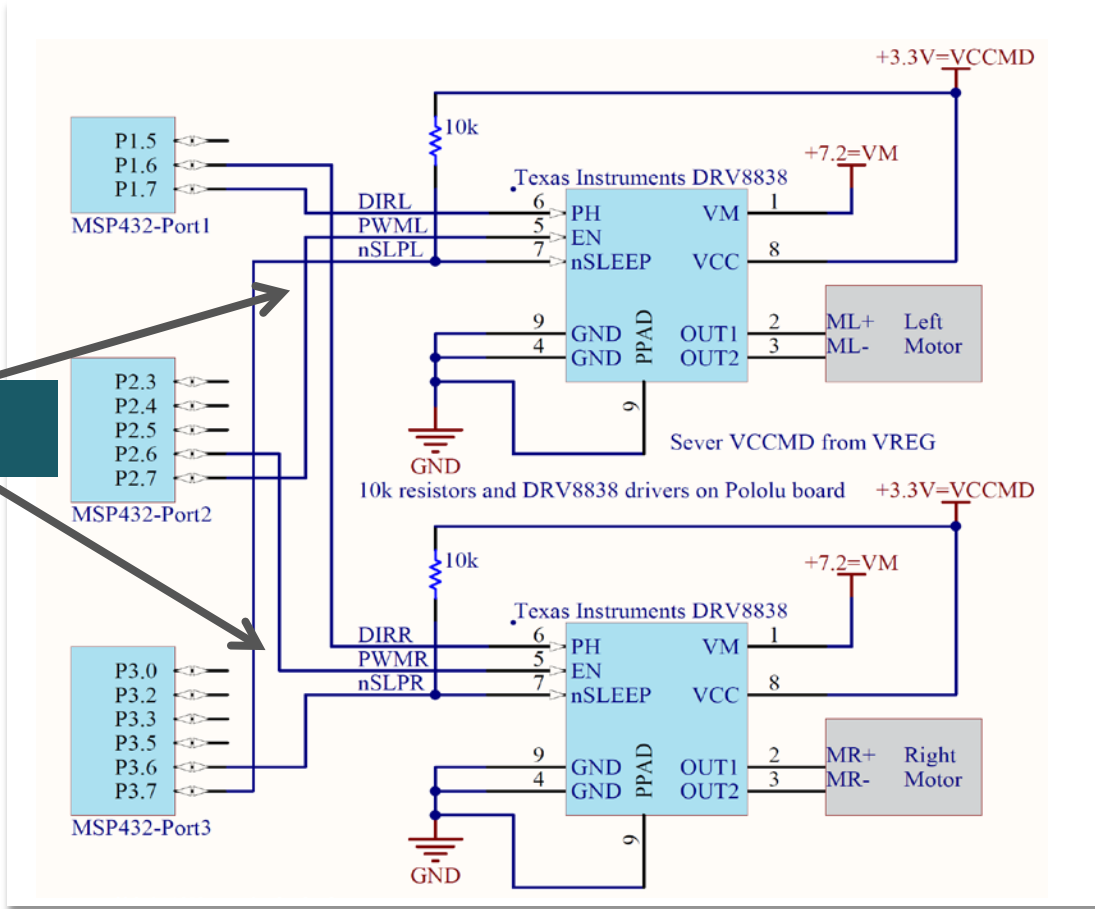
- Timer A on the TI MSP432
 - Clock input, prescale
 - Counter
- PWM output
 - Adjust power to motors
 - Two independent outputs





Interface circuit using the Motor Drive Power Distribution Board (Review)

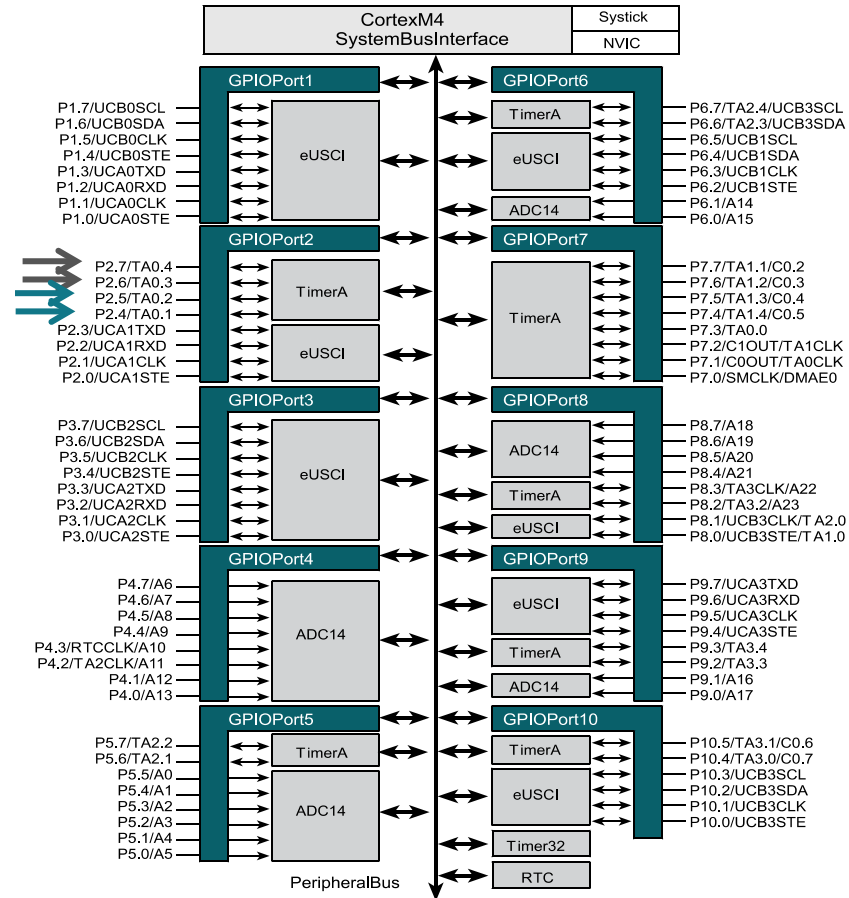
PWM on P2.6, P2.7





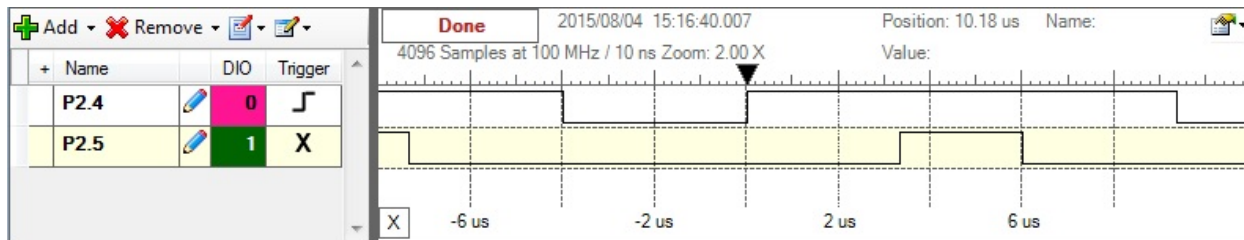
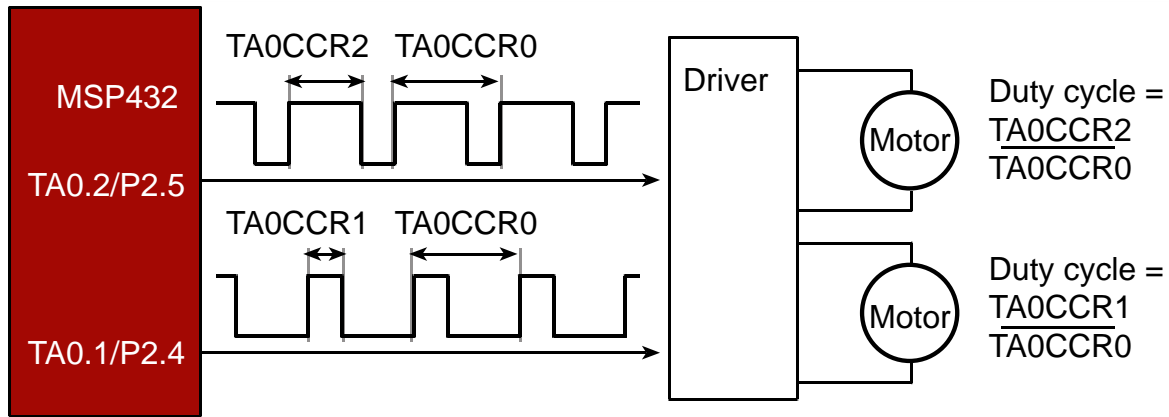
MSP432 Input/Output

Lab code
Starter code



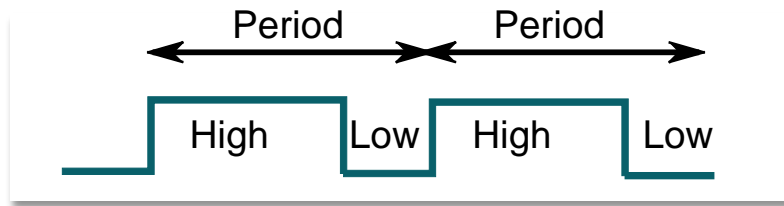
Using Timer - A for Two PWM Outputs

You will convert it to
TA0.4/P2.7
TA0.3/P2.6



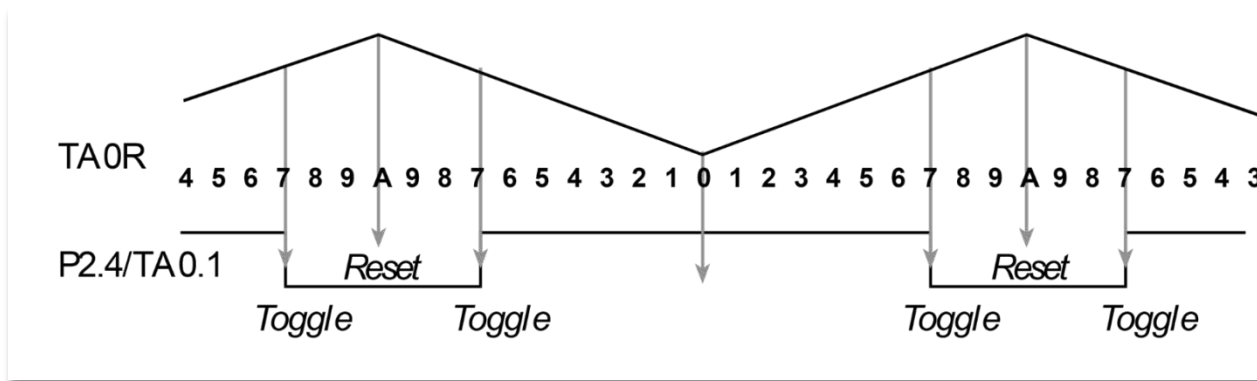
Using Timer - A for Two PWM Outputs

$$\text{DutyCycle} = \frac{\text{High}}{\text{High} + \text{Low}} = \frac{\text{High}}{\text{Period}}$$



PWM Mode

- Up/down count
 - Up to CCR0 (10)
 - Down to 0
- P2.4=1 when timer equals TA0CCR1 (7) on way down
- P2.4=0 when timer equals TA0CCR1 (7) on way up



$T = \text{Prescale}/12\text{MHz}$
 $\text{Period} = 2 * T * \text{CCR0}$
 $\text{DutyCycle} = \text{CCR1}/\text{CCR0}$



PWM Outputs on P2.4 and P2.5

```
// SMCLK = 48MHz/4 = 12 MHz, 83.33ns
// Counter counts up to TA0CCR0 and back down
// Let Timerclock period T = 8/12MHz = 666.7ns
// Period of P2.4 is period*1.333us, duty cycle is duty1/period
// Period of P2.5 is period*1.333us, duty cycle is duty2/period
void PWM_Init12(uint16_t period, uint16_t duty1, uint16_t duty2){
    P2->DIR |= 0x30;          // P2.4, P2.5 output
    P2->SEL0 |= 0x30;         // P2.4, P2.5 Timer0A functions
    P2->SEL1 &= ~0x30;       // P2.4, P2.5 Timer0A functions
    TIMER_A0->CCTL[0] = 0x0080; // CCI0 toggle
    TIMER_A0->CCR[0] = period; // Period is 2*period*8*83.33ns is 1.333*period
    TIMER_A0->EX0 = 0x0000;    // divide by 1
    TIMER_A0->CCTL[1] = 0x0040; // CCR1 toggle/reset
    TIMER_A0->CCR[1] = duty1;  // CCR1 duty cycle is duty1/period
    TIMER_A0->CCTL[2] = 0x0040; // CCR2 toggle/reset
    TIMER_A0->CCR[2] = duty2;  // CCR2 duty cycle is duty2/period
    TIMER_A0->CTL = 0x02F0;    // SMCLK=12MHz, divide by 8, up-down mode

// bit          mode
// 9-8  10      TASSEL, SMCLK=12MHz
// 7-6  11      ID, divide by 8
// 5-4  11      MC, up-down mode
// 2    0       TACLR, no clear
// 1    0       TAIE, no interrupt
// 0    0       TAIFG
}
```

You will convert it to
TA0.4/P2.7
TA0.3/P2.6

$T = 8/12\text{MHz}$
 $\text{Period} = 2 * T * 15000 = 10\text{ms}$
 $\text{DutyCycle} = \text{CCR1}/15000$



PWM Outputs on P2.4 and P2.5

You will convert it to
TA0.4/P2.7
TA0.3/P2.6

```

//*****PWM_Duty1*****
// change duty cycle of PWM output on P2.4
// Inputs:  duty1
// Outputs: none
// period of P2.4 is 2*period*666.7ns, duty cycle is duty1/period
void PWM_Duty1(uint16_t duty1){
    TIMER_A0->CCR[1] = duty1;        // CCR1 duty cycle is duty1/period
}

//*****PWM_Duty2*****
// change duty cycle of PWM output on P2.5
// Inputs:  duty2
// Outputs: none// period of P2.5 is 2*period*666.7ns, duty cycle is duty2/period
void PWM_Duty2(uint16_t duty2){
    TIMER_A0->CCR[2] = duty2;        // CCR2 duty cycle is duty2/period
}

```

DutyCycle

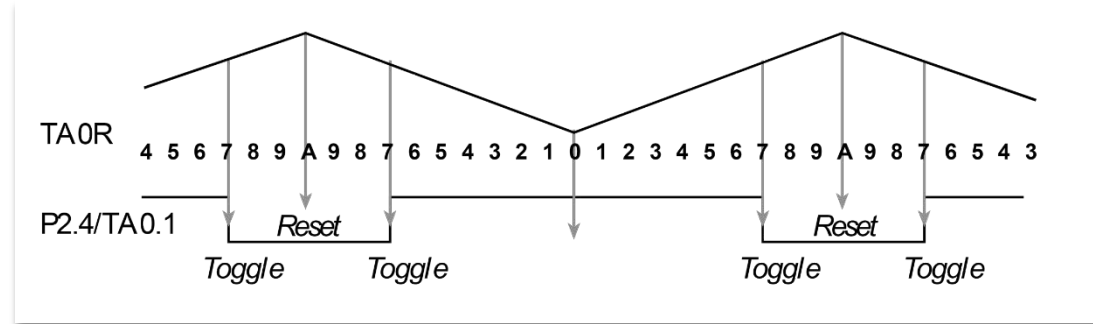
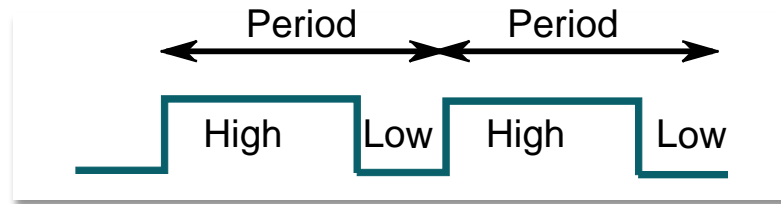
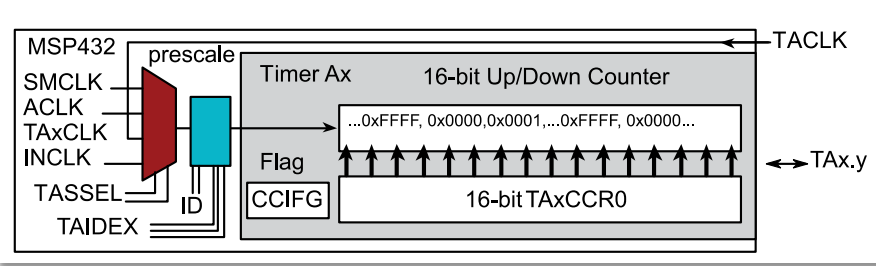
- Precision 14999 alternatives (0 to 14998)
- Range 0 to 99.99%
- Resolution 0.0067%



Summary

PWM output

- Adjust power to motors
- Two independent outputs



IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), [evaluation modules](#), and [samples](http://www.ti.com/sc/docs/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2018, Texas Instruments Incorporated