Announcements

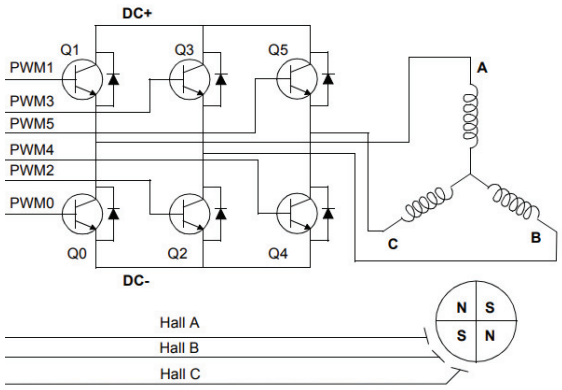
- Upcoming Due Dates
  - Experiment 1 Report: R 1/24 (before class)
  - Experiment 2 Report: T 1/29 (end of class)
  - Experiment 3 Prelab: R 1/31 (before class)

Comments on Motor Control

EXPERIMENT 1

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Motor Driver: Trapezoidal Control



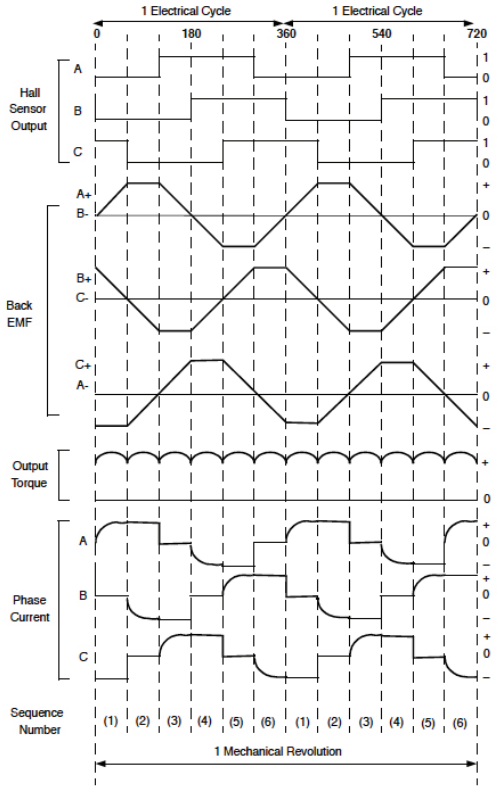
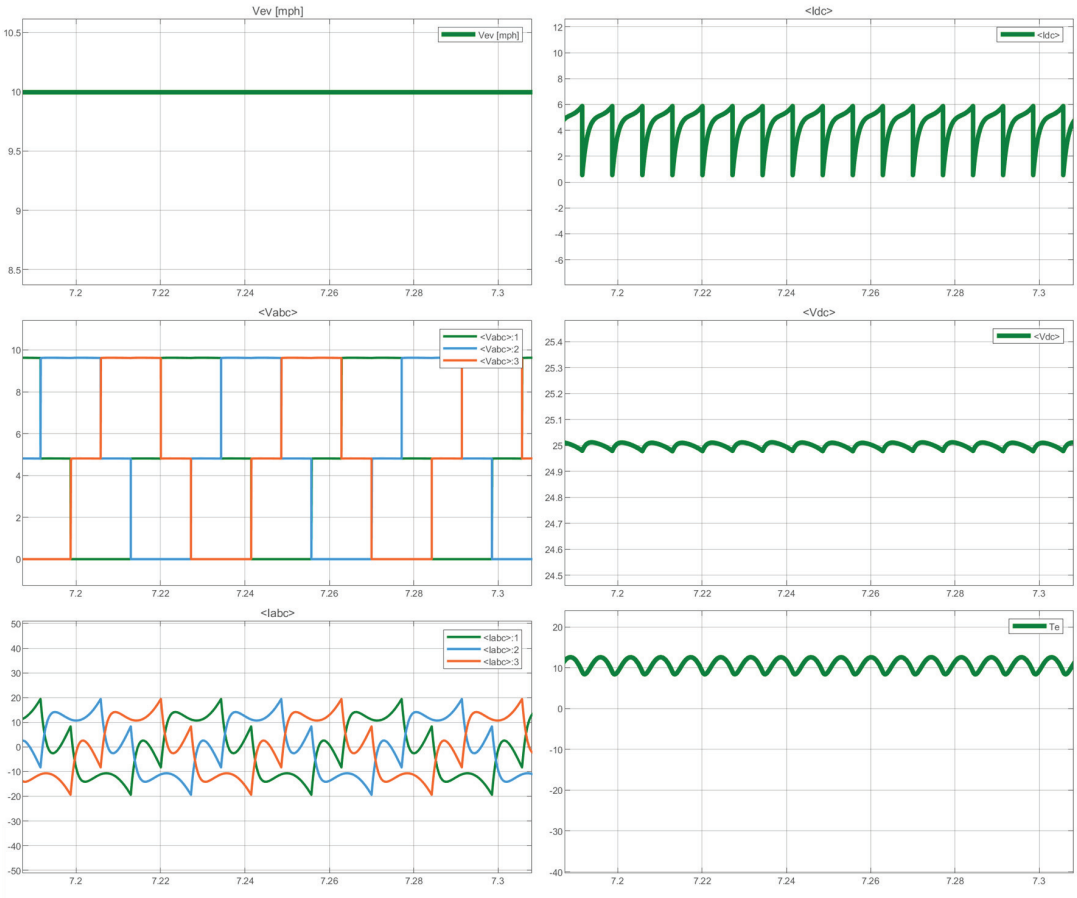


Image credit: Microchip Technology Inc.

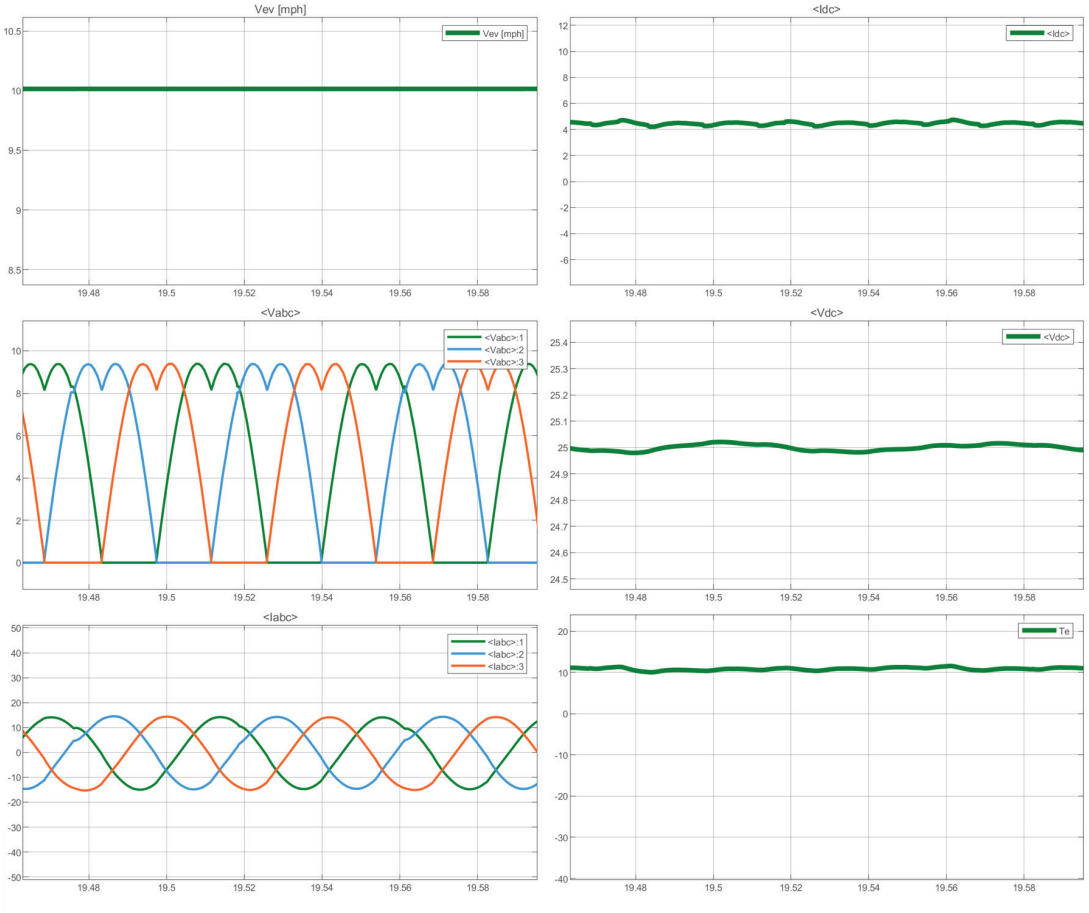
THE UNIVERSITY OF TENNESSEE KNOXVILLE

Trapezoidal Control



THE UNIVERSITY OF TENNESSEE KNOXVILLE

Sinusoidal Control

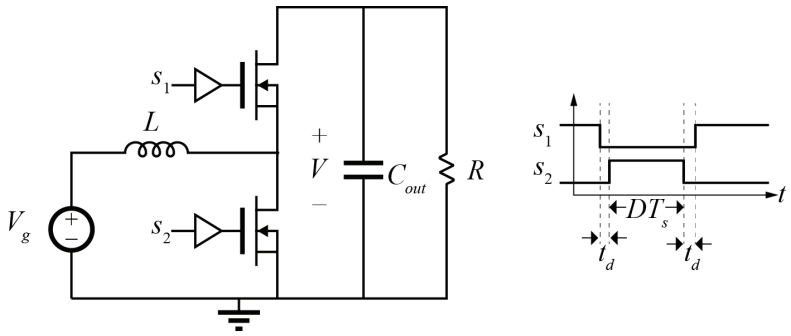


THE UNIVERSITY OF TENNESSEE KNOXVILLE

Microcontroller Programming

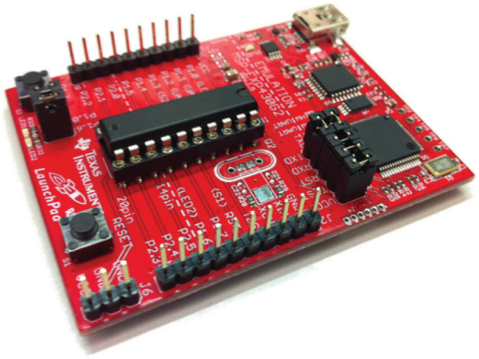
EXPERIMENT 2

Experiment 2



- Experiment 3 will build synchronous boost converter
- To operate open loop, need gate drive signals
- Experiment 2: brief introduction to MSP programming – Generate voltage-controlled PWM signals

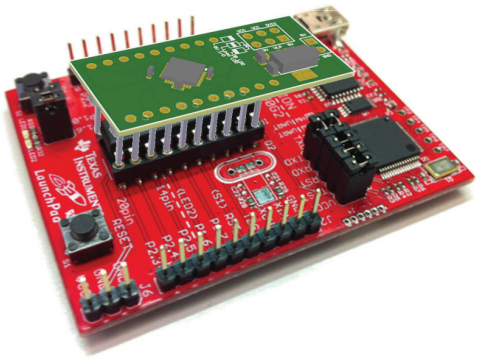
Microprocessor: MSP430 Launchpad



- MSP430 microprocessors from Texas Instruments
  - MSP430G2553
- Programmable in C or ASM
- Ultra-low power (not a focus here)

- On-board USB bootloader
- Two LEDs, one switch
- Two timers, one 5-channel 10-bit ADC
- System clock up to 16 MHz

High Resolution PWM



MSP430G2553:

- 16 MHz clock
  - Max PWM resolution is 62.5ns

MSP430F5172:

- PWM 16x clock multiplier
  - Max PWM resolution is 4ns

- Final decision TBD; same programming approach applies in either case

Notes on Launchpad

- P1.1 and P1.2 are used as part of the digital communication for the debugger
- P1.0, P1.6, P2.1, P2.3, P2.5 can be tied to on-board LEDs for visual debugging
- Do not apply power to Vcc; it is generated on-board
- Launchpad **does not** break out all pins on MSP
  - User guide lists all functionality in family
  - Make sure to take note of what *your* chip can do
- Documentation contains both assembly and C code

MSP430 Documentation

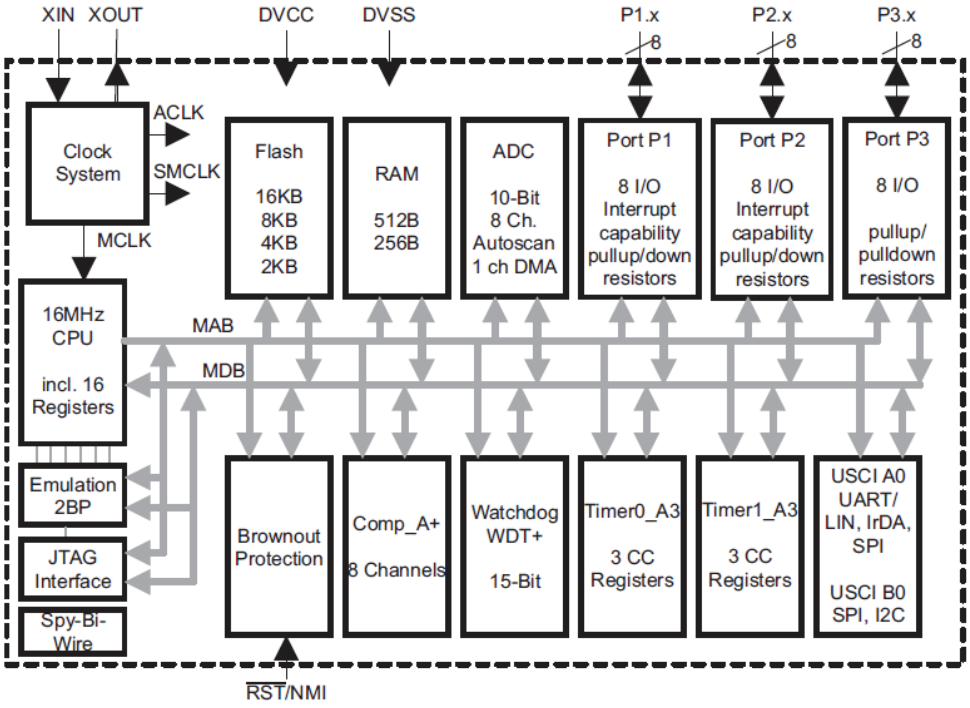
- User's Guide
  - http://www.ti.com/lit/ug/slau144j/slau144j.pdf
- Datasheet
  - http://www.ti.com/lit/ds/symlink/msp430g2553.pdf
- Errata
  - http://www.ti.com/lit/er/slaz440g/slaz440g.pdf

Example Today

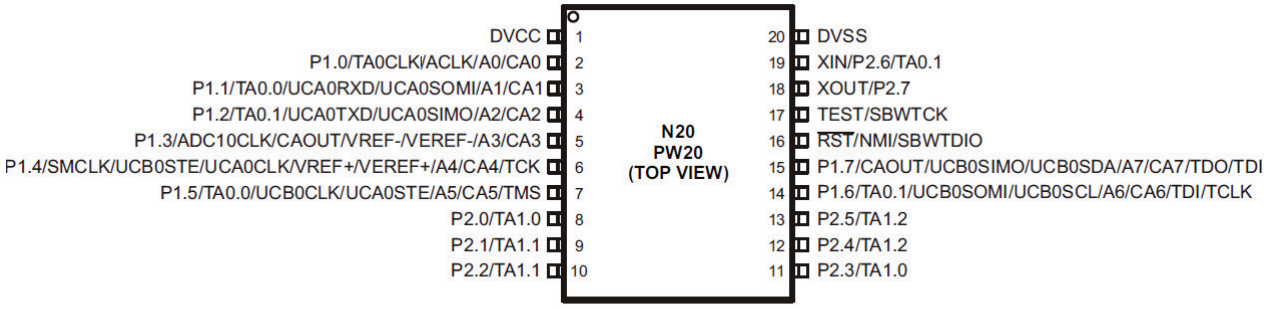
- General Purpose I/O
- System Clock
- TimerA
- Interrupts

MSP430 Internal Block Diagram

Functional Block Diagram, MSP430G2x53



Pin Assignments



NOTE: ADC10 is available on MSP430G2x53 devices only.

NOTE: The pulldown resistors of port P3 should be enabled by setting P3REN.x = 1.

Digital I/O Registers

8.2.1 Input Register PxIN

Each bit in each PxIN register reflects the value of the input signal at the corresponding I/O pin when the pin is configured as I/O function.

Bit = 0: The input is low

Bit = 1: The input is high

8.2.2 Output Registers PxOUT

Each bit in each PxOUT register is the value to be output on the corresponding I/O pin when the pin is configured as I/O function, output direction, and the pullup/down resistor is disabled.

Bit = 0: The output is low

Bit = 1: The output is high

If the pin's pullup/pulldown resistor is enabled, the corresponding bit in the PxOUT register selects pullup or pulldown.

Bit = 0: The pin is pulled down

Bit = 1: The pin is pulled up

8.2.3 Direction Registers PxDIR

Each bit in each PxDIR register selects the direction of the corresponding I/O pin, regardless of the selected function for the pin. PxDIR bits for I/O pins that are selected for other functions must be set as required by the other function.

Bit = 0: The port pin is switched to input direction

Bit = 1: The port pin is switched to output direction

8.2.4 Pullup/Pulldown Resistor Enable Registers PxREN

Each bit in each PxREN register enables or disables the pullup/pulldown resistor of the corresponding I/O pin. The corresponding bit in the PxOUT register selects if the pin is pulled up or pulled down.

Bit = 0: Pullup/pulldown resistor disabled

Bit = 1: Pullup/pulldown resistor enabled

8.2.5 Function Select Registers PxSEL and PxSEL2

Port pins are often multiplexed with other peripheral module functions. See the device-specific data sheet to determine pin functions. Each PxSEL and PxSEL2 bit is used to select the pin function - I/O port or peripheral module function.

Clock Module

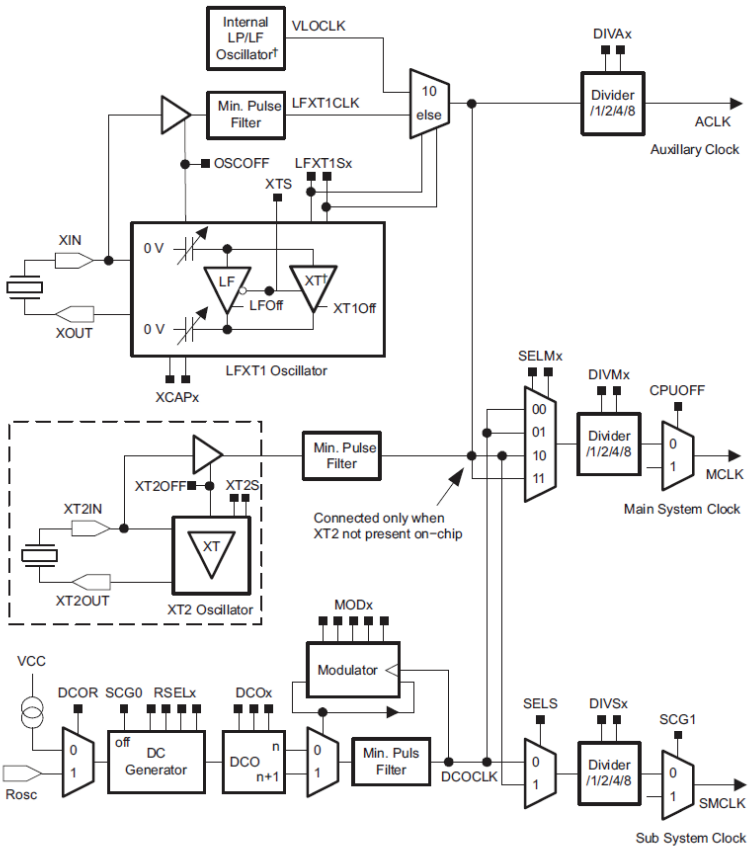


Figure 5-1. Basic Clock Module+ Block Diagram – MSP430F2xx

Clock Registers (1/2)

5.3.1 DCOCTL, DCO Control Register

7	6	5	4	3	2	1	0
DCOx			MODx				
rw-0	rw-1	rw-1	rw-0	rw-0	rw-0	rw-0	rw-0

DCOx	Bits 7-5	DCO frequency select. These bits select which of the eight discrete DCO frequencies within the range defined by the RSELx setting is selected.
MODx	Bits 4-0	Modulator selection. These bits define how often the f_{DCO+1} frequency is used within a period of 32 DCOCLK cycles. During the remaining clock cycles (32-MOD) the f_{DCO} frequency is used. Not useable when DCOx = 7.

5.3.2 BCSCTL1, Basic Clock System Control Register 1

7	6	5	4	3	2	1	0
XT2OFF	XTS(1)(2)	DIVAx		RSELx			
rw-(1)	rw-(0)	rw-(0)	rw-(0)	rw-0	rw-1	rw-1	rw-1

XT2OFF	Bit 7	XT2 off. This bit turns off the XT2 oscillator 0 XT2 is on 1 XT2 is off if it is not used for MCLK or SMCLK.
XTS	Bit 6	LFXT1 mode select. 0 Low-frequency mode 1 High-frequency mode
DIVAx	Bits 5-4	Divider for ACLK 00 /1 01 /2 10 /4 11 /8
RSELx	Bits 3-0	Range select. Sixteen different frequency ranges are available. The lowest frequency range is selected by setting RSELx = 0. RSEL3 is ignored when DCOR = 1.

Clock Registers (2/2)

5.3.3 BCSCTL2, Basic Clock System Control Register 2

7	6	5	4	3	2	1	0
SELMx		DIVMx		SELS	DIVSx		DCOR(1)(2)
rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0	rw-0

SELMx	Bits 7-6	Select MCLK. These bits select the MCLK source. 00 DCOCLK 01 DCOCLK 10 XT2CLK when XT2 oscillator present on-chip. LFXT1CLK or VLOCLK when XT2 oscillator not present on-chip. 11 LFXT1CLK or VLOCLK
DIVMx	Bits 5-4	Divider for MCLK 00 /1 01 /2 10 /4 11 /8
SELS	Bit 3	Select SMCLK. This bit selects the SMCLK source. 0 DCOCLK 1 XT2CLK when XT2 oscillator present. LFXT1CLK or VLOCLK when XT2 oscillator not present
DIVSx	Bits 2-1	Divider for SMCLK 00 /1 01 /2 10 /4 11 /8
DCOR	Bit 0	DCO resistor select. Not available in all devices. See the device-specific data sheet. 0 Internal resistor 1 External resistor

Timer A Block Diagram

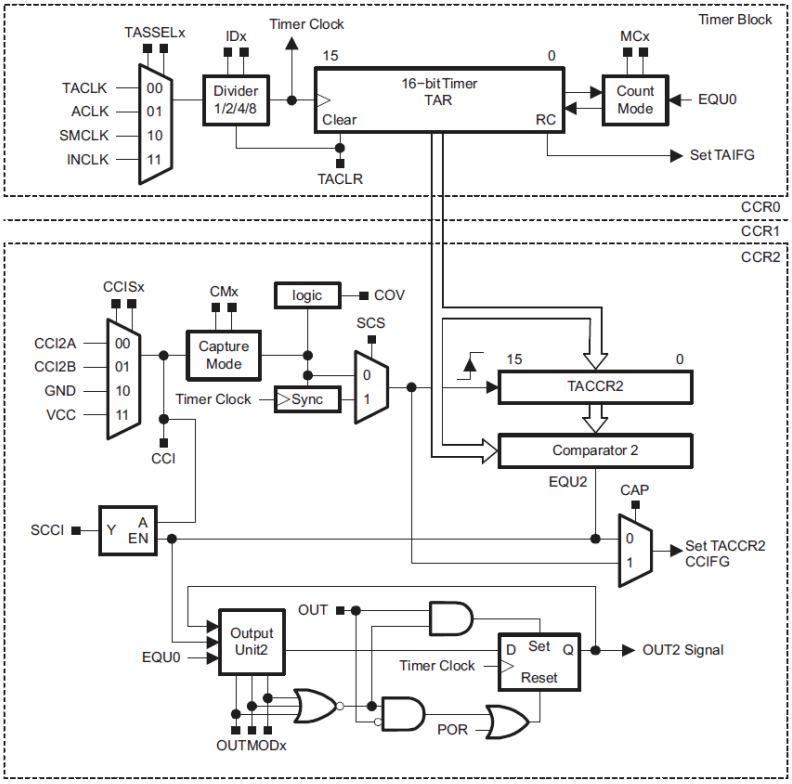


Figure 12-1. Timer_A Block Diagram

Timer A Operation – Up/Down Mode

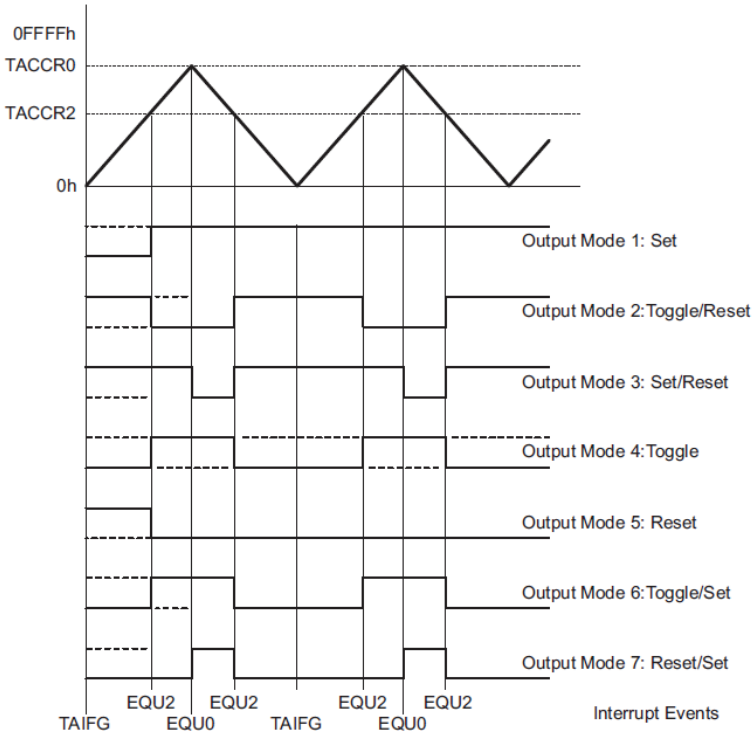


Figure 12-14. Output Example—Timer in Up/Down Mode

Timer A Registers (1/2)

12.3.1 TACTL, Timer_A Control Register

15	14	13	12	11	10	9	8
Unused						TASSELx	
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
IDx		MCx		Unused	TACLR	TAIE	TAIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Unused	Bits 15-10	Unused
TASSELx	Bits 9-8	Timer_A clock source select 00 TACLK 01 ACLK 10 SMCLK 11 INCLK (INCLK is device-specific and is often assigned to the inverted TBCLK) (see the device-specific data sheet)
IDx	Bits 7-6	Input divider. These bits select the divider for the input clock. 00 /1 01 /2 10 /4 11 /8
MCx	Bits 5-4	Mode control. Setting MCx = 00h when Timer_A is not in use conserves power. 00 Stop mode: the timer is halted. 01 Up mode: the timer counts up to TACCR0. 10 Continuous mode: the timer counts up to 0FFFFh. 11 Up/down mode: the timer counts up to TACCR0 then down to 0000h.
Unused	Bit 3	Unused
TACLR	Bit 2	Timer_A clear. Setting this bit resets TAR, the clock divider, and the count direction. The TACLR bit is automatically reset and is always read as zero.
TAIE	Bit 1	Timer_A interrupt enable. This bit enables the TAIFG interrupt request. 0 Interrupt disabled 1 Interrupt enabled
TAIFG	Bit 0	Timer_A interrupt flag 0 No interrupt pending 1 Interrupt pending

Timer A Registers (2/2)

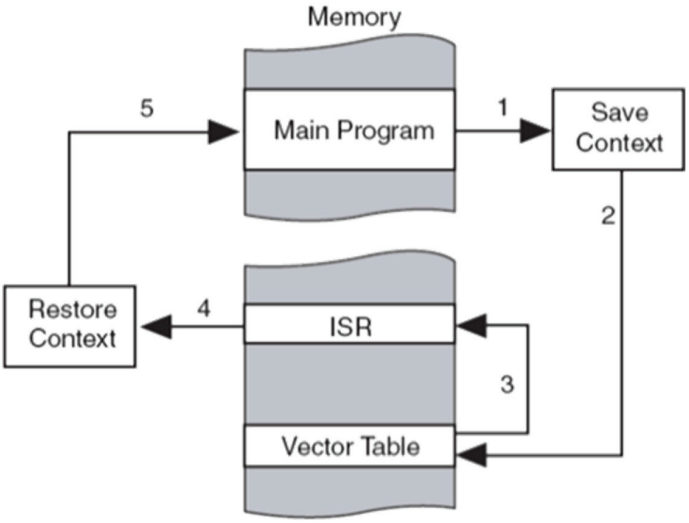
12.3.4 TACCTLx, Capture/Compare Control Register

15	14	13	12	11	10	9	8
CMx		CCISx		SCS	SCCI	Unused	CAP
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r	r0	rw-(0)
7	6	5	4	3	2	1	0
OUTMODx			CCIE	CCI	OUT	COV	CCIFG
rw-(0)	rw-(0)	rw-(0)	rw-(0)	r	rw-(0)	rw-(0)	rw-(0)

CMx	Bit 15-14	Capture mode 00 No capture 01 Capture on rising edge 10 Capture on falling edge 11 Capture on both rising and falling edges
CCISx	Bit 13-12	Capture/compare input select. These bits select the TACCRx input signal. See the device-specific data sheet for specific signal connections. 00 CCIxA 01 CCIxB 10 GND 11 V_{CC}
SCS	Bit 11	Synchronize capture source. This bit is used to synchronize the capture input signal with the timer clock. 0 Asynchronous capture 1 Synchronous capture
SCCI	Bit 10	Synchronized capture/compare input. The selected CCI input signal is latched with the EQUx signal and can be read via this bit
Unused	Bit 9	Unused. Read only. Always read as 0.
CAP	Bit 8	Capture mode 0 Compare mode 1 Capture mode
OUTMODx	Bits 7-5	Output mode. Modes 2, 3, 6, and 7 are not useful for TACCR0, because EQUx = EQU0. 000 OUT bit value 001 Set 010 Toggle/reset 011 Set/reset 100 Toggle 101 Reset 110 Toggle/set 111 Reset/set

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Interrupts



Example Codes From Class

Setting I/O

```
#include <msp430.h>

int main(void) {
   WDTCTL = WDTPW | WDTHOLD;// Stop watchdog timer


   // Set P1.0 to output (high)
   P1DIR |= BIT0;
   P1OUT |= BIT0;

   while(1)
   {
   __no_operation();
   }
}
```


Pulsing I/O

```
int main(void) {
   WDTCTL = WDTPW | WDTHOLD;// Stop watchdog timer


   // Set P1.0 to output (high)
   P1DIR |= BIT0;
   P1OUT |= BIT0;

   while(1)
   {
   P1OUT ^= BIT0;
   __no_operation();
   }
}
```

Setting Up Clocks

```
int main(void) {
   WDTCTL = WDTPW | WDTHOLD;// Stop watchdog timer


   // Set P1.0 to output (high)
   P1DIR |= BIT0;
   P1OUT |= BIT0;

   // Set System Clock to 16 MHz; Set ACLK to VLO
   DCOCTL = DCO0 + DCO1 + DCO2;
   BCSCTL1 = DIVA0 + DIVA1 + RSEL0 + RSEL1 + RSEL2 + RSEL3;
   BCSCTL2 = SELM_0 + DIVM_0;
   BCSCTL3 = LFXT1S_2;

   while(1)
   {
   P1OUT ^= BIT0;
   }
}
```


Problems with CPU PWM

```
int main(void) {
   WDTCTL = WDTPW | WDTHOLD;// Stop watchdog timer


   int i;

   // Set P1.0 to output (high)
   P1DIR |= BIT0;
   P1OUT |= BIT0;

   // Set System Clock to 16 MHz; Set ACLK to VLO
   DCOCTL = DCO0 + DCO1 + DCO2;
   BCSCTL1 = DIVA0 + DIVA1 + RSEL0 + RSEL1 + RSEL2 + RSEL3;
   BCSCTL2 = SELM_0 + DIVM_0;
   BCSCTL3 = LFXT1S_2;

   while(1)
   {
   P1OUT ^= BIT0;
   for (i =0; i<50; i++)
   {
   __no_operation();
   }
   }
}
```

Using TimerA

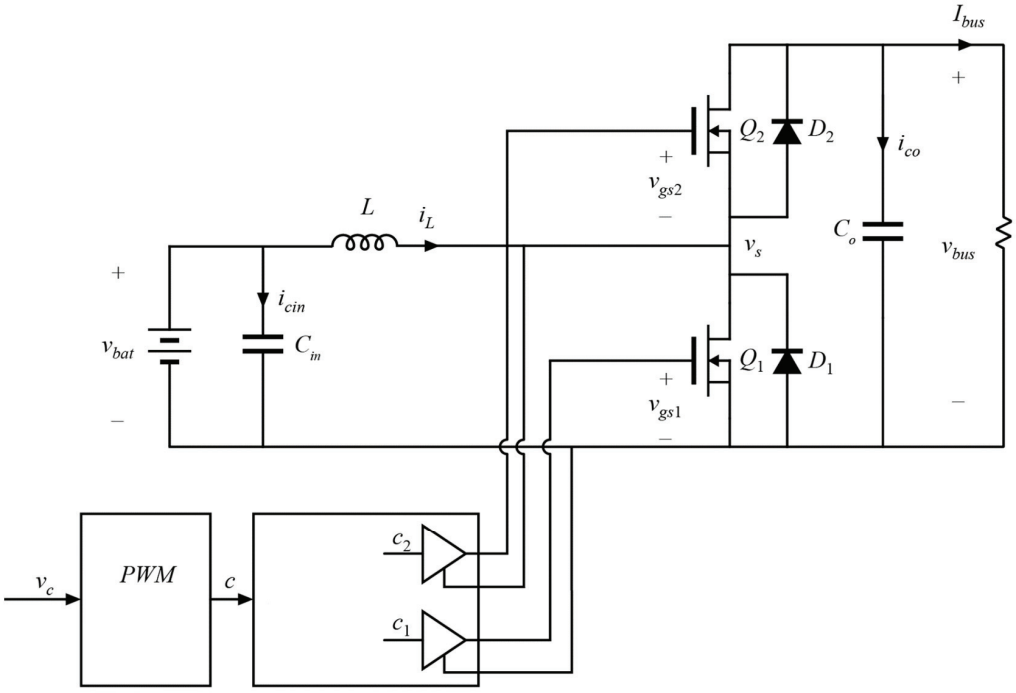
```
int main(void) {
   WDTCTL = WDTPW | WDTHOLD;// Stop watchdog timer

   // Set P1.0 to output (high)
   P1DIR |= BIT0;
   P1OUT |= BIT0;

   // Set P1.6 to TA0.1; Set P1.0 to TA0CLK; Set P1.1 to TA0.0
   P1DIR |= BIT0 + BIT6 + BIT1;
   P1SEL |= BIT0 + BIT6 + BIT1;
   P1SEL2 &= ~(BIT0 + BIT6 + BIT1);

   TA0CTL = ID_3 + MC_3 + TASSEL0; // 8x divider, up/down mode, ACLK source -> 12kHx/8/8 = 187.5 Hz
   TA0CCR0 = 93; // ~ 1Hz period.
   TA0CCTL1 = OUTMOD_2; // toggle/reset
   TA0CCR1 = 46; // 50% duty

   // Set System Clock to 16 MHz; Set ACLK to VLO
   DCOCTL = DCO0 + DCO1 + DCO2;
   BCSCTL1 = DIVA0 + DIVA1 + RSEL0 + RSEL1 + RSEL2 + RSEL3;
   BCSCTL2 = SELM_0 + DIVM_0;
   BCSCTL3 = LFXT1S_2;

   while(1)
   {
         __no_operation();
   }
}
```


Interrupt

```
#include <msp430.h>

int main(void) {
   WDTCTL = WDTPW | WDTHOLD;// Stop watchdog timer
      ...
   //Interrupt Section
   TA0CCTL0 |= CCIE;

      _BIS_SR(GIE);

   while(1)
   {
   for (i =0; i<50; i++)
   {
   __no_operation();
   }
   }
}

// TA0_A1 Interrupt vector
#pragma vector=TIMER0_A0_VECTOR
__interrupt void Timer_A(void)
{

TA0CCR1 = TA0CCR1 + 5;
if (TA0CCR1 > 93)
{
  TA0CCR1 = 5;
}

}
```

EXPERIMENT 3

Experiment 3



Prelab Assignment
Experiment 3
ECE 482

Fig. 1 shows the power stage of the drivetrain boost converter to be assembled in experiment 3. For all parts of this prelab, consider operation of the converter at an operating point around which:

- $V_{bat} = 25$ V
- $V_{bus} \leq 50$ V
- $5\text{ kHz} \leq f_s \leq 1\text{ MHz}$
- $\Delta v_{out} \leq 1\text{V}$

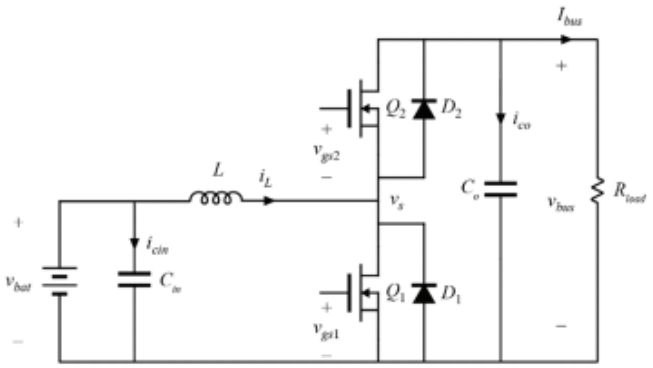


Figure 1: Open loop boost converter (implementation shown with MOSFET devices)

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Design Assessment

In experiment 3, a portion of your grade will be the performance of the design that you choose to build. A 20% segment of the lab grade will be determined by the following formula, which rewards designs with small size, high efficiency, and high power capability:

$$\text{Grade [\%]} = 25 - \kappa_{core} - 100\cdot(0.98 - \eta_{Pout=100}) - \left|\frac{P_{\max} - 250}{50}\right|,$$

where

$$\kappa_{core} = \begin{cases} 0, & ETD29/EFD25 \\ 3, & ETD39 \\ 6, & ETD44 \\ 9, & ETD49 \end{cases}$$

According to the inductor core you have chosen for your design. P_{max} is the maximum power tested, which must be at least 100W, and may be as high as 250W.

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Boost Design

Magnetics Library

Core Geometry	Material
EFD25	Ferroxcube 3C90 Ferroxcube 3F3 Ferroxcube 3F4
ETD29	Ferroxcube 3C90 Ferroxcube 3F3
ETD39	Ferroxcube 3C90 Ferroxcube 3F3
ETD44	Ferroxcube 3C90 Ferroxcube 3F3
ETD49	Ferroxcube 3C90 Ferroxcube 3F3

Core Loss Parameters

Wire Gauge	Diameter [cm]
AWG 10	0.267
AWG 12	0.213
AWG 14	0.171
AWG 16	0.137
AWG 20	0.0874

Full AWG table

Power Semiconductors

Part No.	Description
AOT2500L	150 V, 150 A High Voltage Trench MOSFET
FDP083N15A	150 V, 117 A PowerTrench MOSFET
IPP200N15N3	150 V, 50 A OptiMOS Power MOSFET
irfb4615pbf	150 V, 35 A HEXFET Power MOSFET
CSD19535KCS	100 V, 150 A NexFET Power MOSFET
IPP023N10N5	100 V, 120 A OptiMOS Power MOSFET
FGPF50N33BT	330 V, PDP Trench IGBT
ISL9V3040D	400 V, N-Channel IGBT

Supplemental Lectures

Device	Loss Mechanism	ECE 481	ECE 581
MOSFET	R_{on}	Lecture 7-8	
	C_{oss}		Lecture 7
	Overlap		Lecture 5-6
	P_g		Lecture 5
Diode	V_F	Lecture 7-8	
	R_d	Lecture 7-8	
	t_d cond		Lecture 5
	C_d		Lecture 7 (see: C_{oss})
	Reverse-Recovery	Lecture 11	
Inductor	R_{dc}	Lecture 38	
	Skin Effect	Lecture 39	
	Core Loss	Lecture 39	
	Fringing		
	Proximity	Lecture 39	

ECE581: http://web.eecs.utk.edu/~dcostine/ECE581/Fall2018/schedule.php
ECE481: http://web.eecs.utk.edu/~dcostine/ECE481/Fall2017/schedule.php