Power Electronics Circuits

Prof. Daniel Costinett

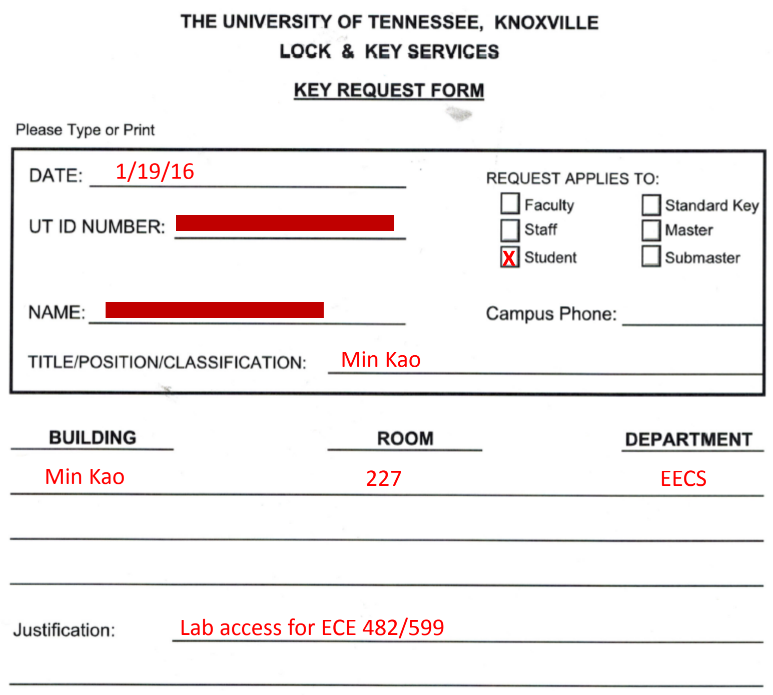
ECE 482 Lecture 2
January 19, 2016

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Announcements

- No prelab for Experiments 1 & 2
- Key Access
- Training needs to be completed and e-mail to Dr. Costinett before Thursday's class
- Meet in MK227 on Thursday

THE UNIVERSITY OF TENNESSEE KNOXVILLE



THE UNIVERSITY OF TENNESSEE, KNOXVILLE
LOCK & KEY SERVICES
KEY REQUEST FORM

Please Type or Print

DATE: 1/19/16

UT ID NUMBER: [redacted]

NAME: [redacted]

TITLE/POSITION/CLASSIFICATION: Min Kao

REQUEST APPLIES TO:
☐ Faculty ☐ Standard Key
☐ Staff ☐ Master
[✓ Student] ☐ Submaster

Campus Phone:

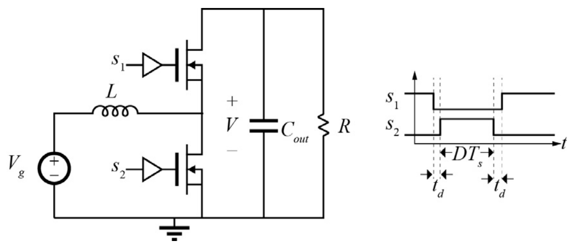
BUILDING	ROOM	DEPARTMENT
Min Kao	227	EECS

Justification: Lab access for ECE 482/599

Experiment 1

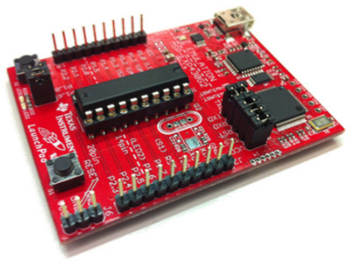
- Begin thinking about Experiment 1 process
  - What tests might you run to determine battery/motor parameters?
- Read through experiment procedure online before Thursday

Experiment 2



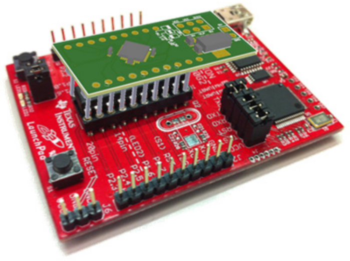
- Experiment 3 will build synchronous boost converter
- To operate open loop, need gate drive signals
- Experiment 2: brief introduction to MSP programming – Generate voltage-controlled PWM signals

Microprocessor: MSP430 Launchpad



- MSP430 microprocessors from Texas Instruments
- 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

MSP430F2172:

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

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

MSP430 Documentation

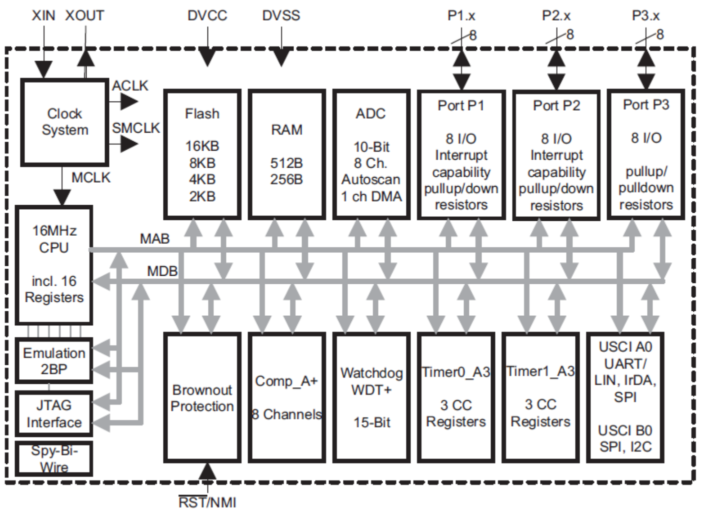
- User’s Guide
  - http://www.ti.com/lit/ug/slau144j/slau144j.pdf
- Datasheet
  - http://www.ti.com/lit/ds/symlink/msp430g2553.pdf
- Errata
  - http://www.ti.com/lit/er/slaz440g/slaz440g.pdf

Example Today

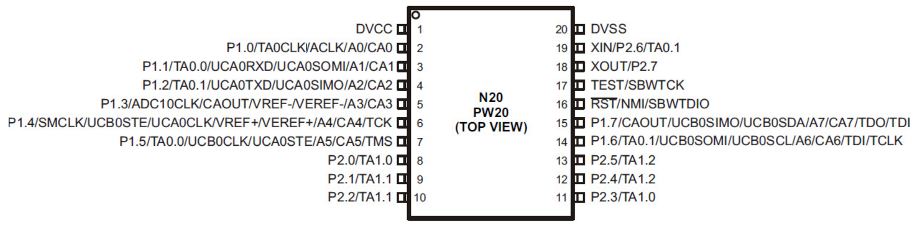
- General Purpose I/O
- System Clock
- TimerA
- Interrupts

MSP430 Internal Block Diagram

Functional Block Diagram, MSP430G2x53


XIN
XOUT
DVCC
DVSS
P1.x
8
P2.x
8
P3.x
8
ACLK
SMCLK
MCLK
Clock System
Flash 16KB 8KB 4KB 2KB
RAM 512B 256B
ADC 10-Bit 8 Ch. Autoscan 1 ch DMA
Port P1 8 I/O Interrupt capability pullup/down resistors
Port P2 8 I/O Interrupt capability pullup/down resistors
Port P3 8 I/O pullup/ pulldown resistors
16MHz CPU incl. 16 Registers
MAB
MDB
Emulation 2BP
JTAG Interface
Spy-Bi-Wire
Brownout Protection
Comp_A+ 8 Channels
Watchdog WDT+ 15-Bit
Timer0_A3 3 CC Registers
Timer1_A3 3 CC Registers
USCI A0 UART/ LIN, IrDA, SPI USCI B0 SPI, I2C
RST/NMI

Pin Assignments


DVCC 1
P1.0/TA0CLK/ACLK/A0/CA0 2
P1.1/TA0.0/UCA0RXD/UCA0SOMI/A1/CA1 3
P1.2/TA0.1/UCA0TXD/UCA0SIMO/A2/CA2 4
P1.3/ADC10CLK/CAOUT/VREF-/VEREF-/A3/CA3 5
P1.4/SMCLK/UCB0STE/UCA0CLK/VREF+/VEREF+/A4/CA4/TCK 6
P1.5/TA0.0/UCB0CLK/UCA0STE/A5/CA5/TMS 7
P2.0/TA1.0 8
P2.1/TA1.1 9
P2.2/TA1.1 10
N20
PW20
(TOP VIEW)
20 DVSS
19 XIN/P2.6/TA0.1
18 XOUT/P2.7
17 TEST/SBWTCK
16 RST/NMI/SBWTDIO
15 P1.7/CAOUT/UCB0SIMO/UCB0SDA/A7/CA7/TDO/TDI
14 P1.6/TA0.1/UCB0SOMI/UCB0SCL/A6/CA6/TDI/TCLK
13 P2.5/TA1.2
12 P2.4/TA1.2
11 P2.3/TA1.0

NOTE: ADC10 is available on MSP430G2x53 devices only.

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

Clock Module

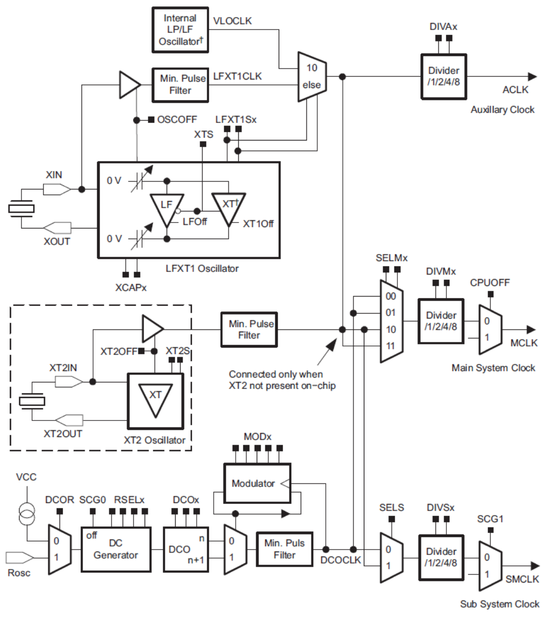


Figure 5-1. Basic Clock Module+ Block Diagram − MSP430F2xx

Timer A Operation – Up/Down Mode

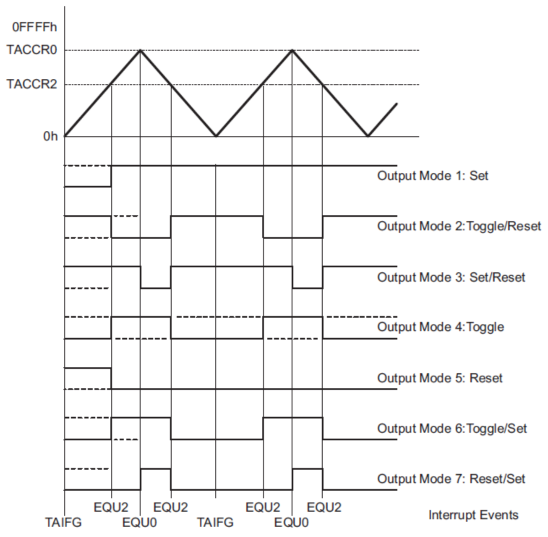


Figure 12-14. Output Example—Timer in Up/Down Mode

Timer A Block Diagram

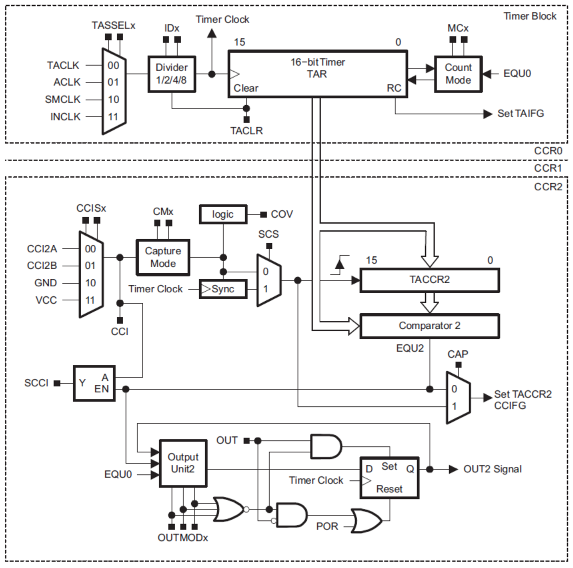


Figure 12-1. Timer_A Block Diagram

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

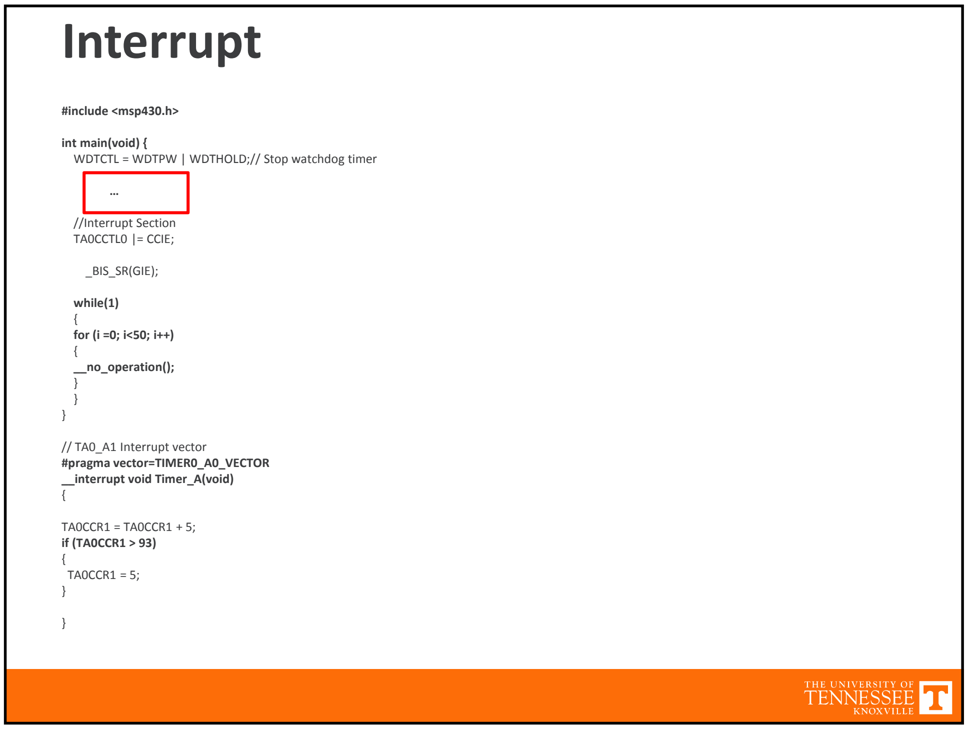
```
int main(void) {
   WDTCTL = WDTPW | WDTHOLD;// Stop watchdog timer


   // 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;
}

}
```