CPE 323
Intro to Embedded Computer Systems
Digital-to-Analog Conversion

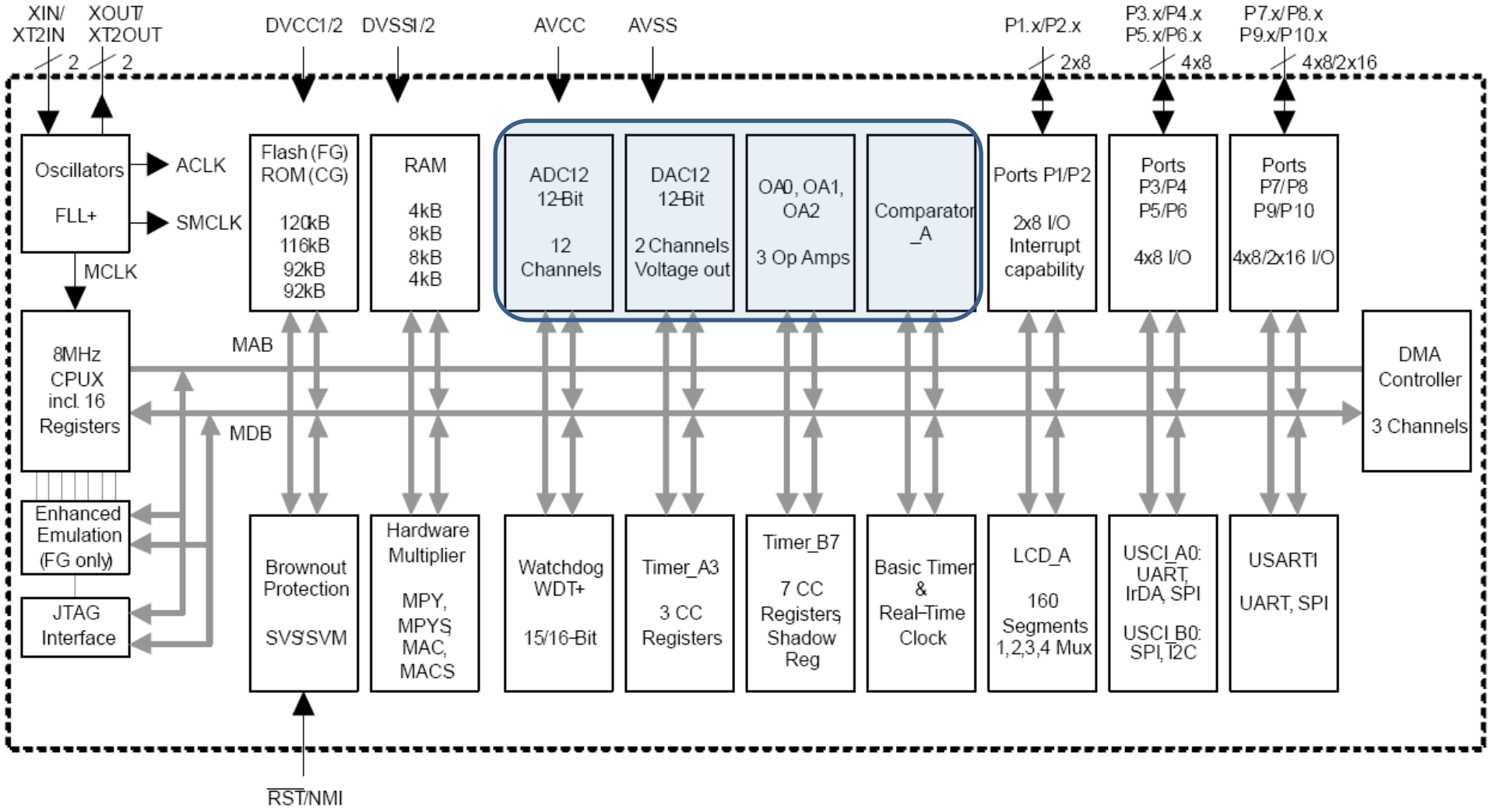
Aleksandar Milenkovic

milenka@uah.edu

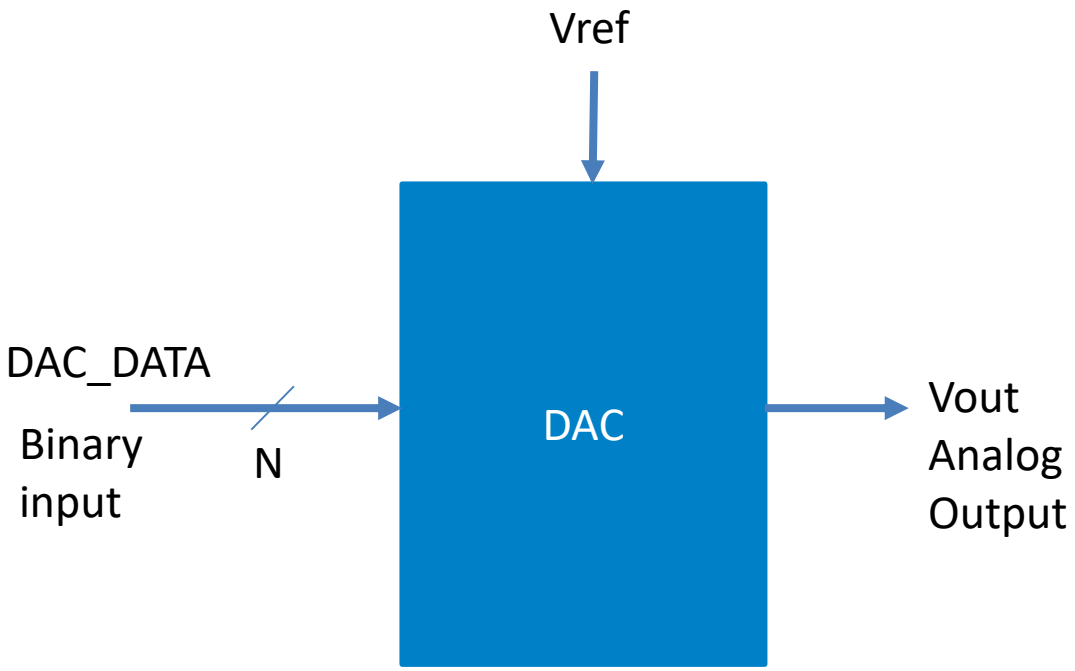
Admin

- Quiz.06 is tomorrow
- HW.5 due Tuesday next week

MSP430FG4618 Block Diagram



DAC – System View



$$Vout = Vref \cdot \frac{DAC\_DATA}{2^N}$$

How does it work?

- Multiple implementations
- One of the simplest: WEIGTHED RESISTOR DAC
- 4-bit DAC; Digital input: $b_3\ b_2\ b_1\ b_0$ (b_3 is the MSB)
- Reference voltage: V_R

4-bit Weighted Resistor DAC

R-2R Ladder DAC

MSP430 DAC12

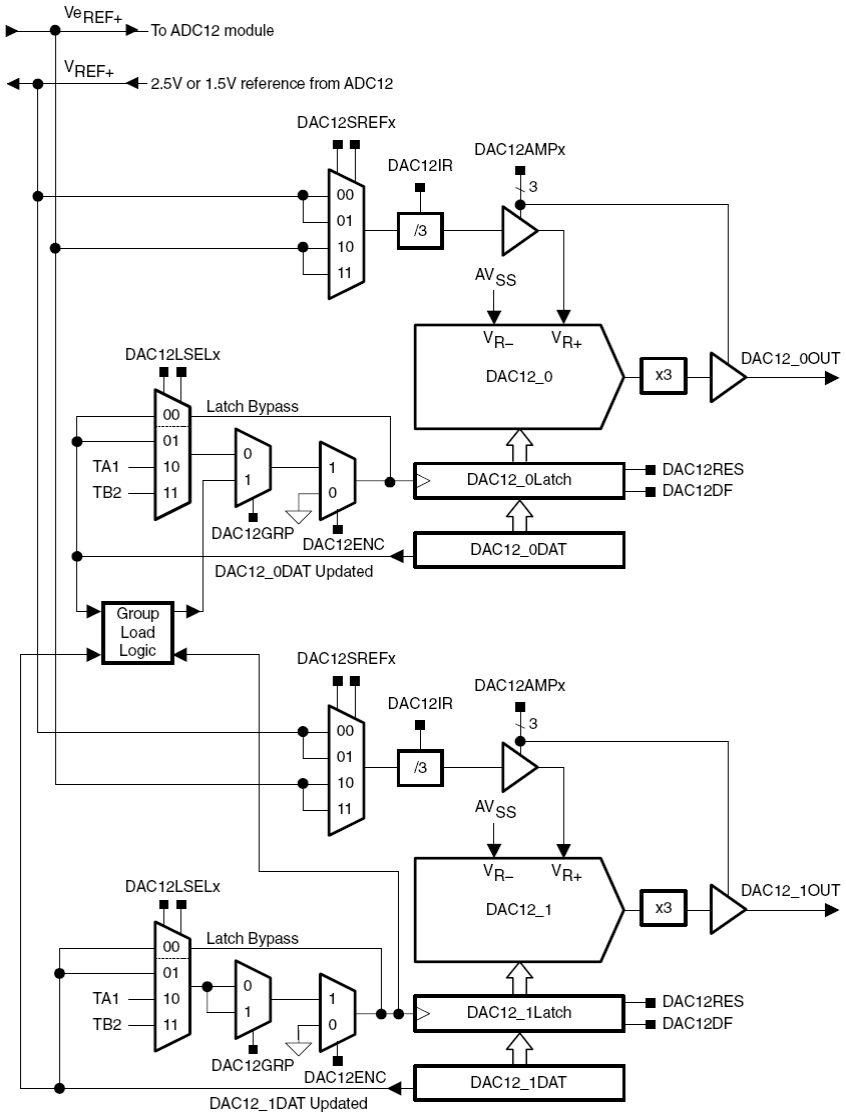
- Modes
  - N = 8 (8-bit mode) or N = 12 (12-bit mode) => DAC12RES control bit
  - unsigned or straight binary mode (0 – 4095) or signed mode (-2048 – 2047) => DAC12DF
  - Voltage output equations for straight binary
  - Amplification (1x or 3x) => DAC12IR

$$Vout = V_{FS} \cdot \frac{DAC\_DATA}{2^N}$$

Resolution	DAC12RES	DAC12IR	Output Voltage Formula
12 bit	0	0	$Vout = Vref \times 3 \times \frac{DAC12\_xDAT}{4096}$
12 bit	0	1	$Vout = Vref \times \frac{DAC12\_xDAT}{4096}$
8 bit	1	0	$Vout = Vref \times 3 \times \frac{DAC12\_xDAT}{256}$
8 bit	1	1	$Vout = Vref \times \frac{DAC12\_xDAT}{256}$

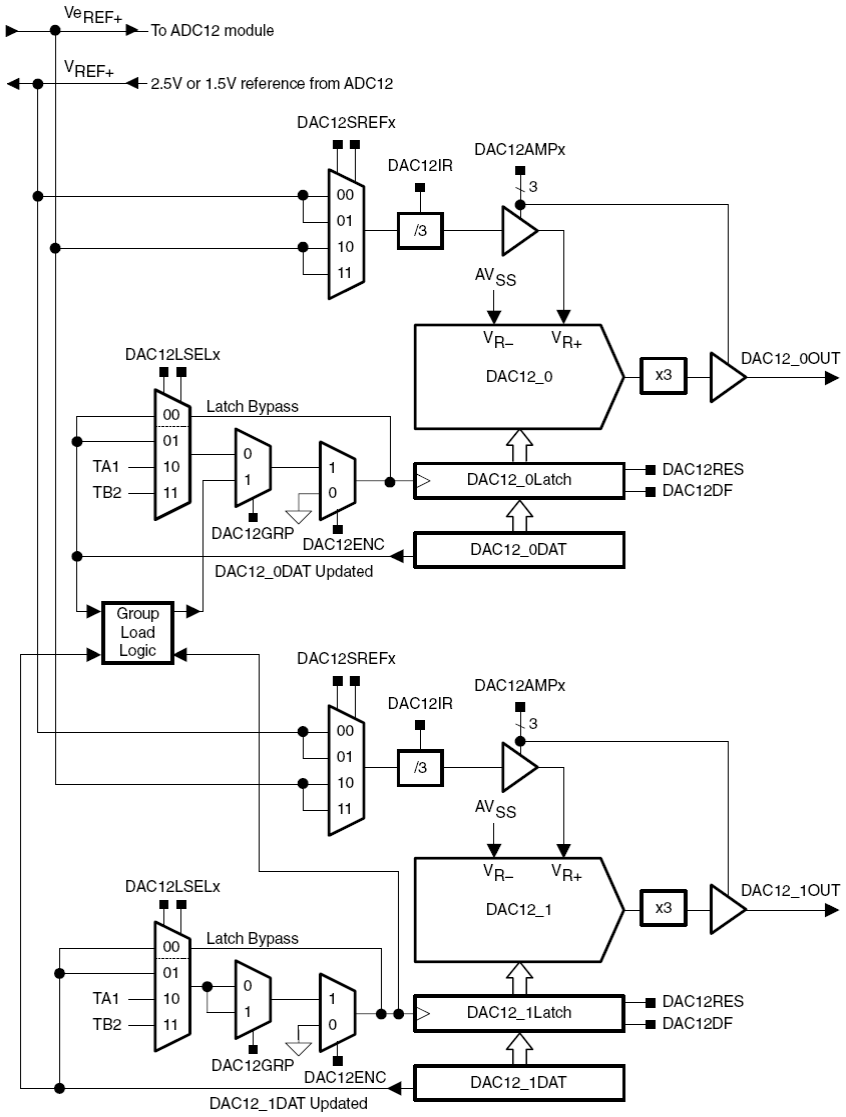
MSP430 DAC12

- 2 channels (0_OUT, 1_OUT)
- Outputs (4618)
  - DAC12OPS=0 (0_OUT => P6.6 and 1_OUT => P6.7)
  - DAC12OPS=1 (0_OUT => VeREF+, 1_OUT => P5_1)
- Reference Voltage
  - Internal 1.5 V or 2.5 (from Voltage Gen in ADC12) or external VeREF+



MSP430 DAC12

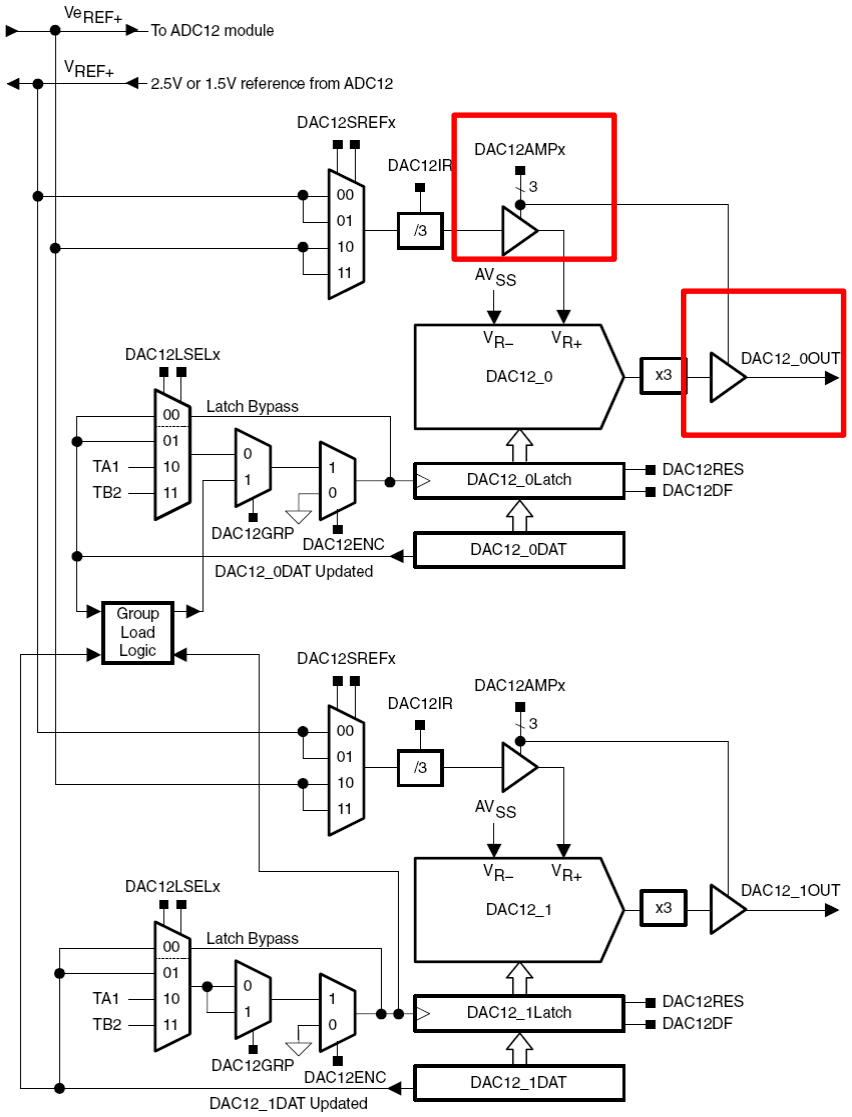
- Voltage Output
  - DAC12_xDAT is double buffered
  - When to update output voltage is controlled by DAC12LSEL (latch selection)
    - 0 – transparent (as soon as you write to into data register it becomes visible to the core)
    - 1 – DAC data is latched and presented to the core on the next write
    - 2, 3 – data is latched on the rising edge of the signal coming from from TimerA CCR1 or TimerB CCR2



MSP430 DAC12

- DAC12AMPx control bits configure DAC12 amplifier as follows
  - these setting control setting time vs. current consumption of the DAC12 input and output amplifiers

DAC12AMPx	Input Buffer	Output Buffer
000	Off	DAC12 off, output high Z
001	Off	DAC12 off, output 0 V
010	Low speed/current	Low speed/current
011	Low speed/current	Medium speed/current
100	Low speed/current	High speed/current
101	Medium speed/current	Medium speed/current
110	Medium speed/current	High speed/current
111	High speed/current	High speed/current



MSP430 DAC12 Registers

Register	Short Form	Register Type	Address	Initial State
DAC12_0 control	DAC12_0CTL	Read/write	01C0h	Reset with POR
DAC12_0 data	DAC12_0DAT	Read/write	01C8h	Reset with POR
DAC12_1 control	DAC12_1CTL	Read/write	01C2h	Reset with POR
DAC12_1 data	DAC12_1DAT	Read/write	01CAh	Reset with POR

DAC12_xCTL, DAC12 Control Register

15	14	13	12	11	10	9	8
DAC12OPS	DAC12SREFx		DAC12RES	DAC12LSELx		DAC12 CALON	DAC12IR
rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)

7	6	5	4	3	2	1	0
DAC12AMPx			DAC12DF	DAC12IE	DAC12IFG	DAC12ENC	DAC12 GRP
rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)

Modifiable only when DAC12ENC = 0

MSP430 DAC12 Registers

Register	Short Form	Register Type	Address	Initial State
DAC12_0 control	DAC12_0CTL	Read/write	01C0h	Reset with POR
DAC12_0 data	DAC12_0DAT	Read/write	01C8h	Reset with POR
DAC12_1 control	DAC12_1CTL	Read/write	01C2h	Reset with POR
DAC12_1 data	DAC12_1DAT	Read/write	01CAh	Reset with POR

DAC12_xCTL, DAC12 Control Register

15	14	13	12	11	10	9	8
DAC12OPS	DAC12SREFx		DAC12RES	DAC12LSELx		DAC12 CALON	DAC12IR
rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)

7	6	5	4	3	2	1	0
DAC12AMPx			DAC12DF	DAC12IE	DAC12IFG	DAC12ENC	DAC12 GRP
rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)	rw–(0)

Modifiable only when DAC12ENC = 0

Example #1

Problem statement: Using DAC12_0 and 2.5V ADC12REF reference with a gain of 1, output 1V on P6.6.

```
//                MSP430xG461x
//             -----------------
//          /|\|               XIN|-
//           | |                  | 32kHz
//           --|RST           XOUT|-
//             |                  |
//             |        DAC0/P6.6|--> 1V
//             |                  |
```

Design:

Step 1: Clocks: ACLK = 32kHz, MCLK = SMCLK = default DCO 1048576Hz, ADC12CLK = ADC12OSC

Step 2: Stop watchdog timer.

Step 3: Ports initialization: P6.6 as special function port (analog output channel A0).

Step 4: DAC12 initialization:

Internal reference voltage (2.5V); software delay to settle down.

Select medium speed/current; DAC12AMPx=5 (101).

Gain should be 1 (DAC12IR=1); Default is gain 3 (DAC12IR=0).

Latch: DAC latches the data when written to DAC12_0DAT (DAC12LSELx=00; default).

DAC12ENC=1 (Enable converter).

Step 5: Software organization. Use TimerA to allow for the internal reference voltage.

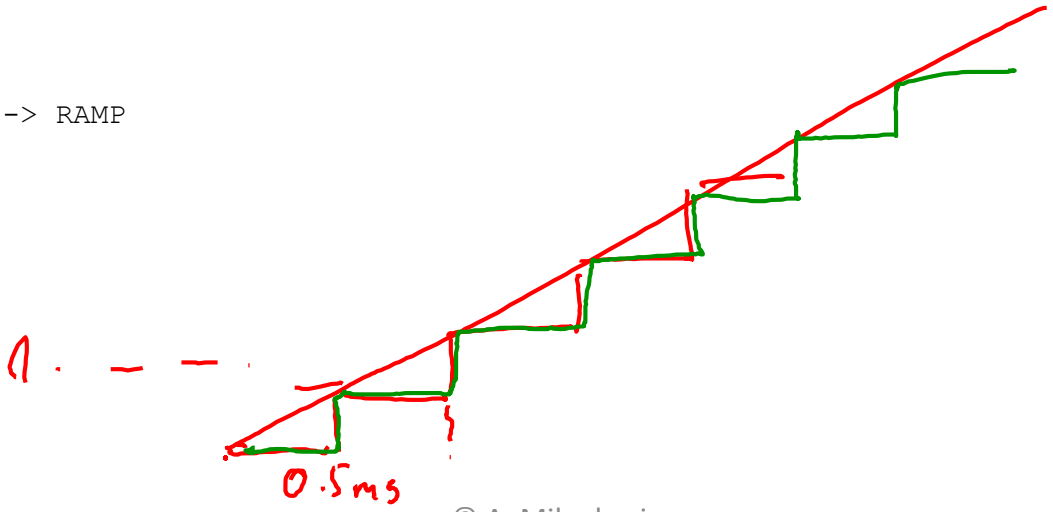
Example #1

```
#include "msp430xG46x.h"

void main(void)
{
  WDTCTL = WDTPW + WDTHOLD;                 // Stop WDT
  ADC12CTL0 = REF2_5V + REFON;              // Internal 2.5V ref on
  TACCR0 = 13600;                           // Delay to allow Ref to settle
  TACCTL0 |= CCIE;                          // Compare-mode interrupt.
  TACTL = TACLR + MC_1 + TASSEL_2;          // up mode, SMCLK
  __bis_SR_register(LPM0_bits + GIE);       // Enter LPM0, enable interrupts
  TACCTL0 &= ~CCIE;                         // Disable timer interrupt
  __disable_interrupt();                    // Disable Interrupts
  DAC12_0CTL = DAC12IR + DAC12AMP_5 + DAC12ENC; // Int ref gain 1
  DAC12_0DAT = 0x0666;                      // 1.0V
  __bis_SR_register(LPM0_bits + GIE);       // Enter LPM0
}

#pragma vector = TIMERA0_VECTOR
__interrupt void TA0_ISR(void)
{
  TACTL = 0;                                // Clear Timer_A control registers
  __bic_SR_register_on_exit(LPM0_bits);     // Exit LPMx, interrupts enabled
}
```

Example #2 (Voltage Ramp)

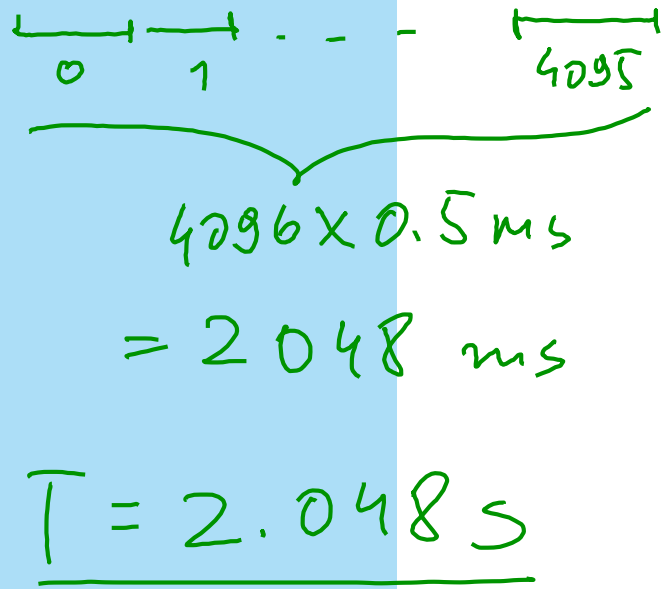
- Problem statement: Using DAC12_0 and 2.5V ADC12REF reference with a gain of 1, output positive ramp on P6.6. Use WDT to provide ~0.5ms interrupt used to wake up the CPU and update the DAC with new sample. Use the internal 2.5V Vref.

```
//                MSP430xG461x
//             -----------------
//         /|\|              XIN|-
//          | |                 | 32kHz
//          --|RST          XOUT|-
//            |                 |
//            |        DAC0/P6.6|--> RAMP
//            |                 |
```



Example #2 (Voltage Ramp)

```
#include "msp430xG46x.h"

void main(void) {
  WDTCTL = WDT_MDLY_0_5;                // WDT ~0.5ms interval timer
  IE1 |= WDTIE;                         // Enable WDT interrupt
  ADC12CTL0 = REF2_5V + REFON;          // Internal 2.5V ref on
  TACCR0 = 13600;                       // Delay to allow Ref to settle
  TACCTL0 |= CCIE;                      // Compare-mode interrupt.
  TACTL = TACLR + MC_1 + TASSEL_2;      // up mode, SMCLK
  __bis_SR_register(LPM0_bits + GIE);   // Enter LPM0, enable interrupts
  TACCTL0 &= ~CCIE;                     // Disable timer interrupt
  __disable_interrupt();                // Disable Interrupts
  DAC12_0CTL = DAC12IR + DAC12AMP_5 + DAC12ENC; // Int ref gain 1

  while (1)
  {
    __bis_SR_register(LPM0_bits + GIE);   // Enter LPM0, interrupts enabled
    DAC12_0DAT++;                         // Positive ramp
    DAC12_0DAT &= 0x0FFF;
  }
}
```



Example #2 (cont'd)

```
#pragma vector = TIMERA0_VECTOR
__interrupt void TA0_ISR(void)
{
  TACTL = 0;                              // Clear Timer_A control registers
  __bic_SR_register_on_exit(LPM0_bits);   // Exit LPMx, interrupts enabled
}

#pragma vector = WDT_VECTOR
__interrupt void WDT_ISR(void)
{
  __bic_SR_register_on_exit(LPM0_bits);    // TOS = clear LPM0
}
```