NEVER GONNA GIVE

ECE 490-ST :: Wireless Computing

- Lesson 19
  - The one where I begin by apologizing

Valparaiso University

HYPERVENTILATE!

NEVER GONNA GIVE

When Modeling

- They will not always match reality.
  - Intrinsic problems in the model
  - Intrinsic problems in the simulation engine
- You need to a) understand the problem b) understand your modeling system and c) understand your measurement system

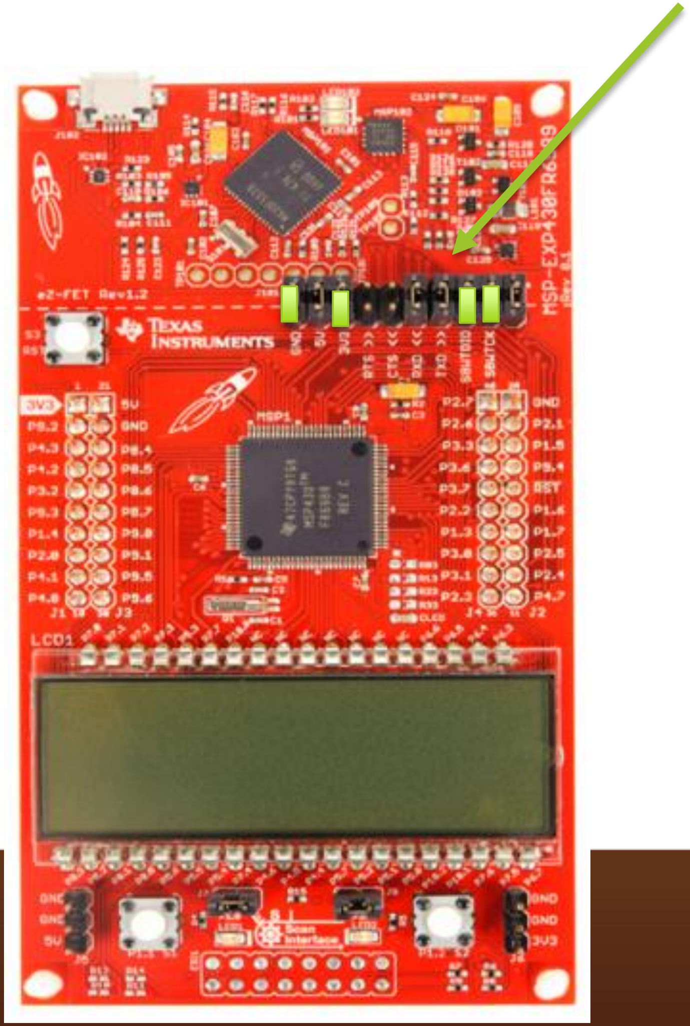
- Always test your equipment!

NEVER GONNA GIVE

ECE 490-ST :: Wireless Computing

- Lesson 19
  - The one where I start by apologizing
  - And then go on to discuss
    Low Power Programming

NEVER GONNA GIVE



These wires will program our microcontroller
Remove jumpers and connect as follows:

REMINDER

- ON 490 BOARD, make sure that the power jumper is between pins 2 and 3. This will let the Launchpad power our MCU.

Goals for today

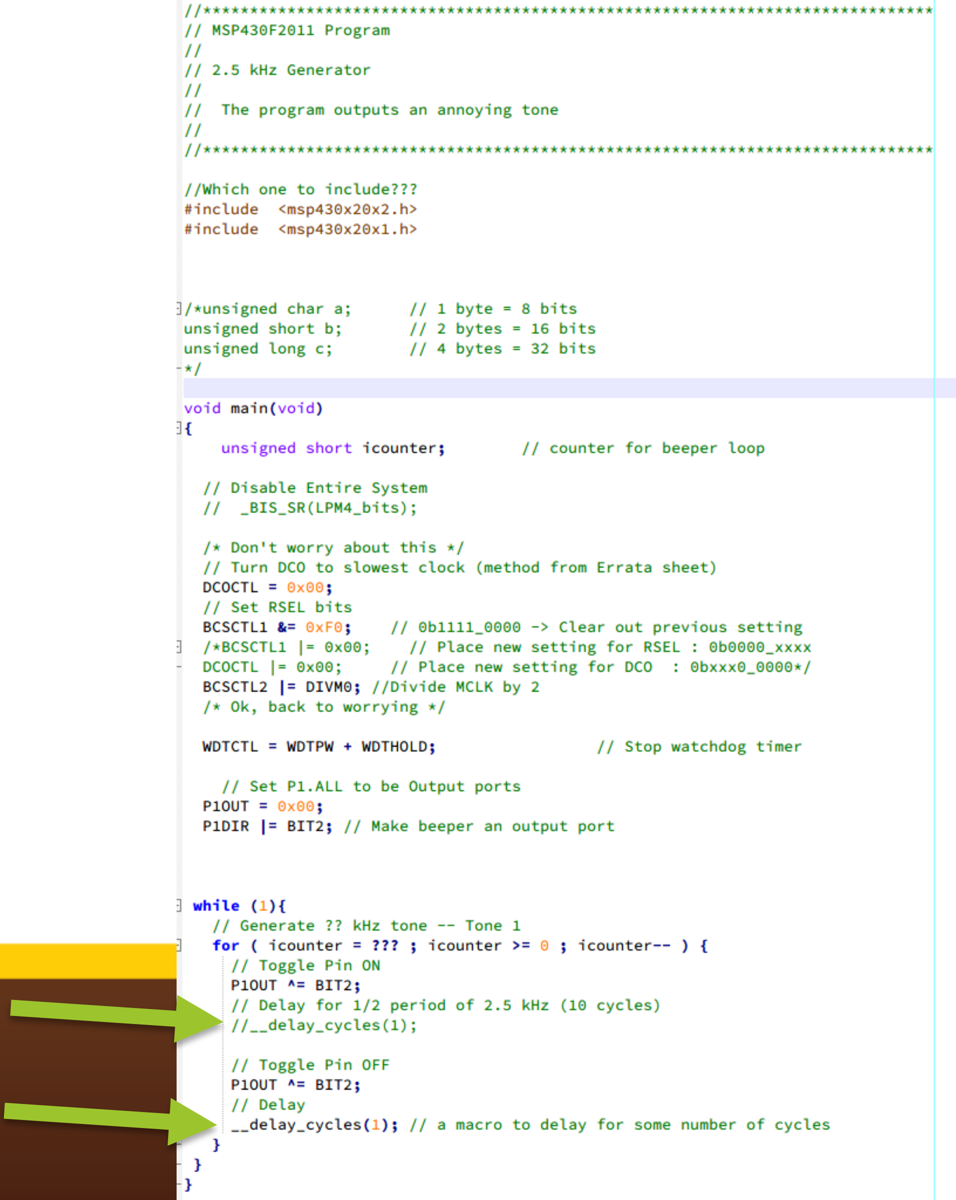
- Make a pulsed beeping program that consumes the lowest power possible

Lesson Plan

- Measure power draw from last lesson’s program
- Remove low-power clocking and measure power draw
- Make it so the beep “pulses” Beep … Beep … Beep
  - Measure power draw between two states, and control duty cycle to make effective resistor

NEVER GONNA GIVE

These lines control the frequency of the beep

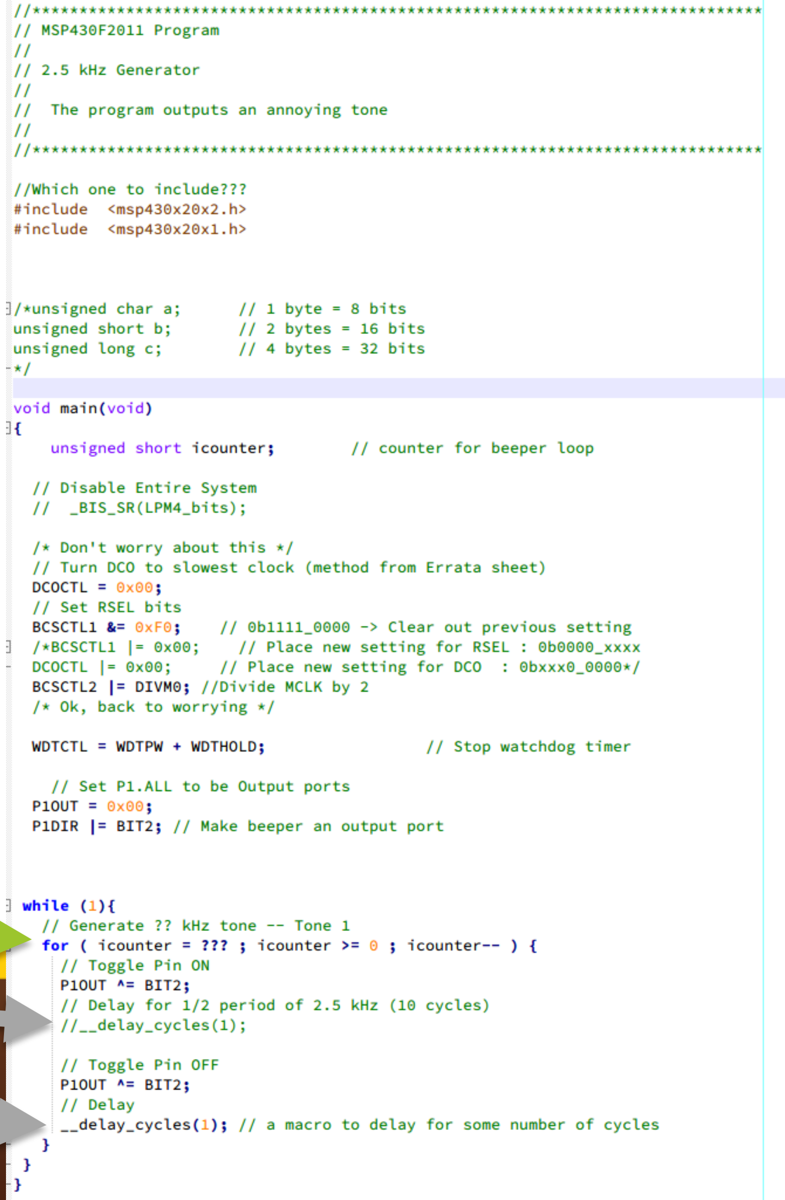


```
//****************************************************************************
// MSP430F2011 Program
//
// 2.5 kHz Generator
//
//  The program outputs an annoying tone
//
//****************************************************************************

//Which one to include???
#include  <msp430x20x2.h>
#include  <msp430x20x1.h>



/*unsigned char a;      // 1 byte = 8 bits
unsigned short b;       // 2 bytes = 16 bits
unsigned long c;        // 4 bytes = 32 bits
*/

void main(void)
{
    unsigned short icounter;        // counter for beeper loop

  // Disable Entire System
  //  _BIS_SR(LPM4_bits);

  /* Don't worry about this */
  // Turn DCO to slowest clock (method from Errata sheet)
  DCOCTL = 0x00;
  // Set RSEL bits
  BCSCTL1 &= 0xF0;    // 0b1111_0000 -> Clear out previous setting
  /*BCSCTL1 |= 0x00;    // Place new setting for RSEL : 0b0000_xxxx
  DCOCTL |= 0x00;     // Place new setting for DCO  : 0bxxx0_0000*/
  BCSCTL2 |= DIVM0; //Divide MCLK by 2
  /* Ok, back to worrying */

  WDTCTL = WDTPW + WDTHOLD;                 // Stop watchdog timer

    // Set P1.ALL to be Output ports
  P1OUT = 0x00;
  P1DIR |= BIT2; // Make beeper an output port



 while (1){
   // Generate ?? kHz tone -- Tone 1
   for ( icounter = ??? ; icounter >= 0 ; icounter-- ) {
     // Toggle Pin ON
     P1OUT ^= BIT2;
     // Delay for 1/2 period of 2.5 kHz (10 cycles)
     //__delay_cycles(1);

     // Toggle Pin OFF
     P1OUT ^= BIT2;
     // Delay
     __delay_cycles(1); // a macro to delay for some number of cycles
   }
 }
}
```

NEVER GONNA GIVE

These lines control the frequency of the beep

This for loop controls duration of beep (actually it does nothing right now)

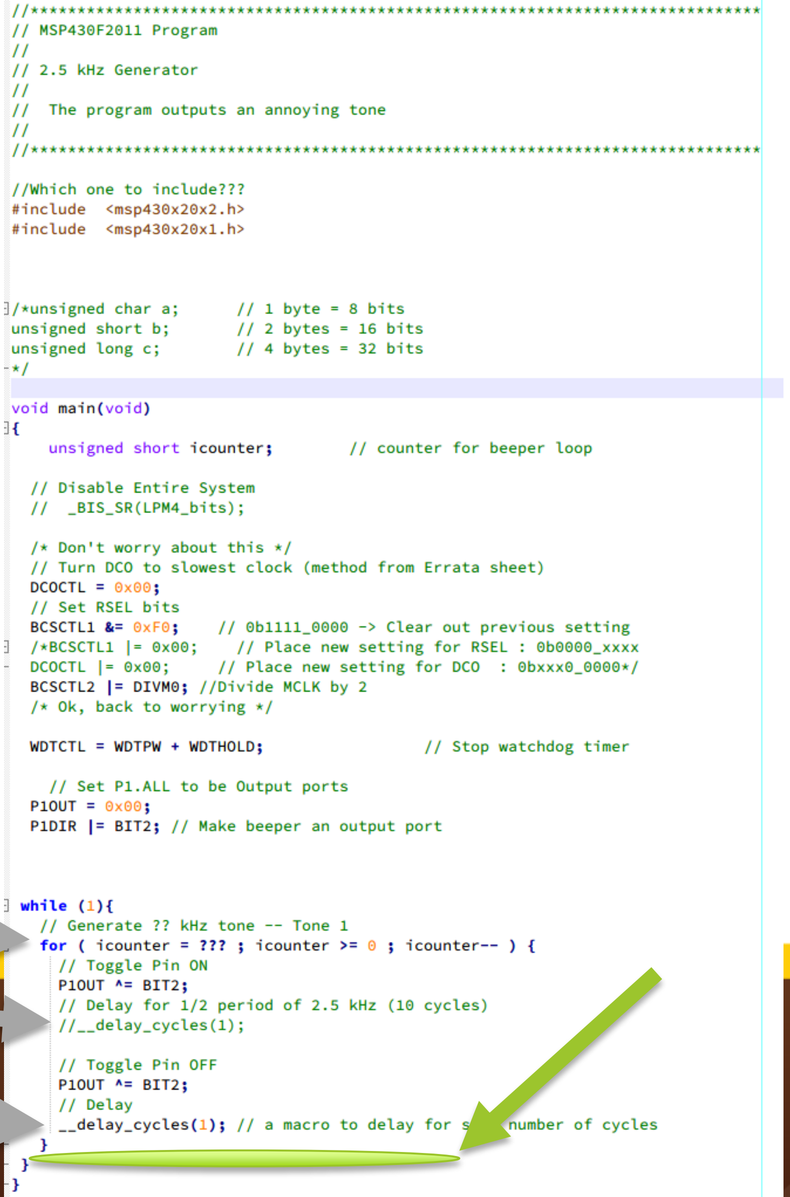


```
//******************************************************************************
// MSP430F2011 Program
//
// 2.5 kHz Generator
//
//  The program outputs an annoying tone
//
//******************************************************************************

//Which one to include???
#include  <msp430x20x2.h>
#include  <msp430x20x1.h>



/*unsigned char a;      // 1 byte = 8 bits
unsigned short b;       // 2 bytes = 16 bits
unsigned long c;        // 4 bytes = 32 bits
*/

void main(void)
{
    unsigned short icounter;        // counter for beeper loop

  // Disable Entire System
  //  _BIS_SR(LPM4_bits);

  /* Don't worry about this */
  // Turn DCO to slowest clock (method from Errata sheet)
  DCOCTL = 0x00;
  // Set RSEL bits
  BCSCTL1 &= 0xF0;    // 0b1111_0000 -> Clear out previous setting
  /*BCSCTL1 |= 0x00;    // Place new setting for RSEL : 0b0000_xxxx
  DCOCTL |= 0x00;     // Place new setting for DCO  : 0bxxx0_0000*/
  BCSCTL2 |= DIVM0; //Divide MCLK by 2
  /* Ok, back to worrying */

  WDTCTL = WDTPW + WDTHOLD;                 // Stop watchdog timer

    // Set P1.ALL to be Output ports
  P1OUT = 0x00;
  P1DIR |= BIT2; // Make beeper an output port


 while (1){
   // Generate ?? kHz tone -- Tone 1
   for ( icounter = ??? ; icounter >= 0 ; icounter-- ) {
     // Toggle Pin ON
     P1OUT ^= BIT2;
     // Delay for 1/2 period of 2.5 kHz (10 cycles)
     //__delay_cycles(1);

     // Toggle Pin OFF
     P1OUT ^= BIT2;
     // Delay
     __delay_cycles(1); // a macro to delay for some number of cycles
   }
 }
}
```

These lines control the frequency of the beep

This for loop controls duration of beep (actually it does nothing right now)

You’ll need to insert code here, at the end of the for loop to delay between beeps



```
//******************************************************************************
// MSP430F2011 Program
//
// 2.5 kHz Generator
//
//  The program outputs an annoying tone
//
//******************************************************************************

//Which one to include???
#include  <msp430x20x2.h>
#include  <msp430x20x1.h>



/*unsigned char a;      // 1 byte = 8 bits
unsigned short b;       // 2 bytes = 16 bits
unsigned long c;        // 4 bytes = 32 bits
*/

void main(void)
{
    unsigned short icounter;        // counter for beeper loop

  // Disable Entire System
  //  _BIS_SR(LPM4_bits);

  /* Don't worry about this */
  // Turn DCO to slowest clock (method from Errata sheet)
  DCOCTL = 0x00;
  // Set RSEL bits
  BCSCTL1 &= 0xF0;    // 0b1111_0000 -> Clear out previous setting
  /*BCSCTL1 |= 0x00;    // Place new setting for RSEL : 0b0000_xxxx
  DCOCTL |= 0x00;     // Place new setting for DCO  : 0bxxx0_0000*/
  BCSCTL2 |= DIVM0; //Divide MCLK by 2
  /* Ok, back to worrying */

  WDTCTL = WDTPW + WDTHOLD;                 // Stop watchdog timer

    // Set P1.ALL to be Output ports
  P1OUT = 0x00;
  P1DIR |= BIT2; // Make beeper an output port



 while (1){
   // Generate ?? kHz tone -- Tone 1
   for ( icounter = ??? ; icounter >= 0 ; icounter-- ) {
     // Toggle Pin ON
     P1OUT ^= BIT2;
     // Delay for 1/2 period of 2.5 kHz (10 cycles)
     //__delay_cycles(1);

     // Toggle Pin OFF
     P1OUT ^= BIT2;
     // Delay
     __delay_cycles(1); // a macro to delay for s    number of cycles
   }
  }
}
```

STEP 1

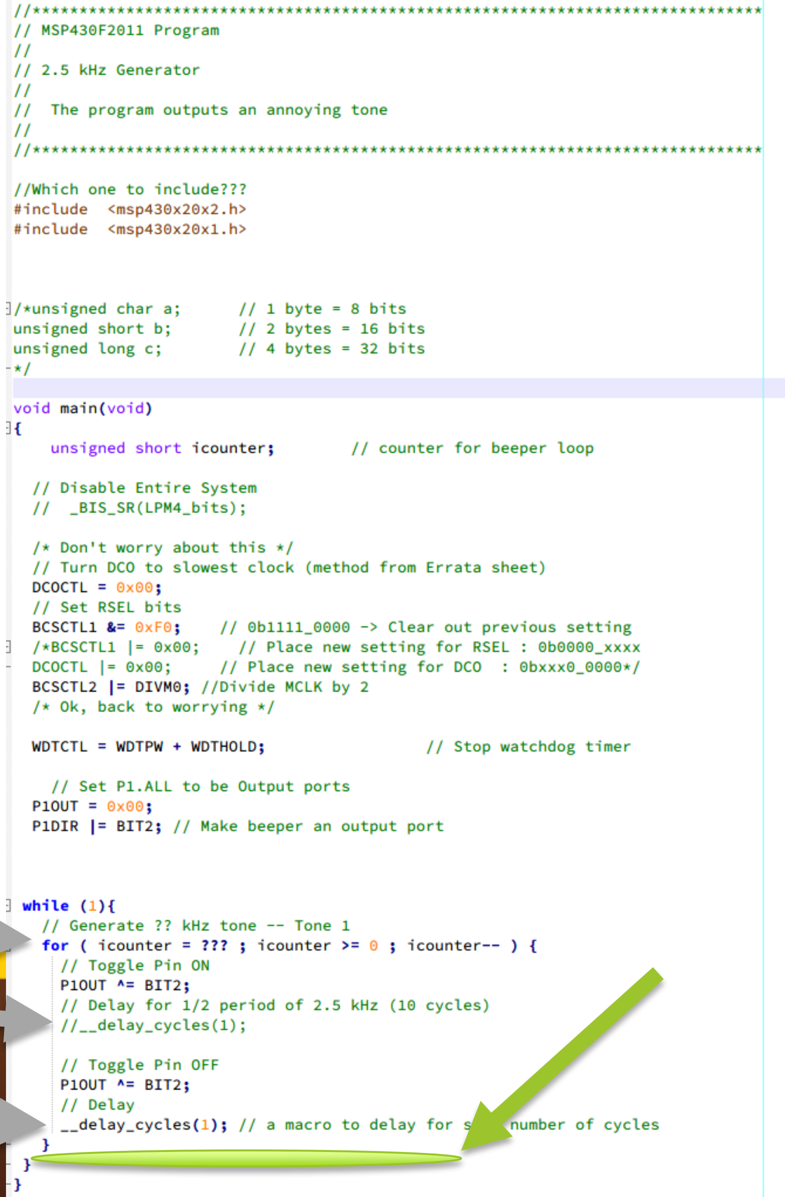
These lines control the frequency of the beep

This for loop controls duration of beep (actually it does nothing right now)

You'll need to insert code here, at the end of the for loop to delay between beeps

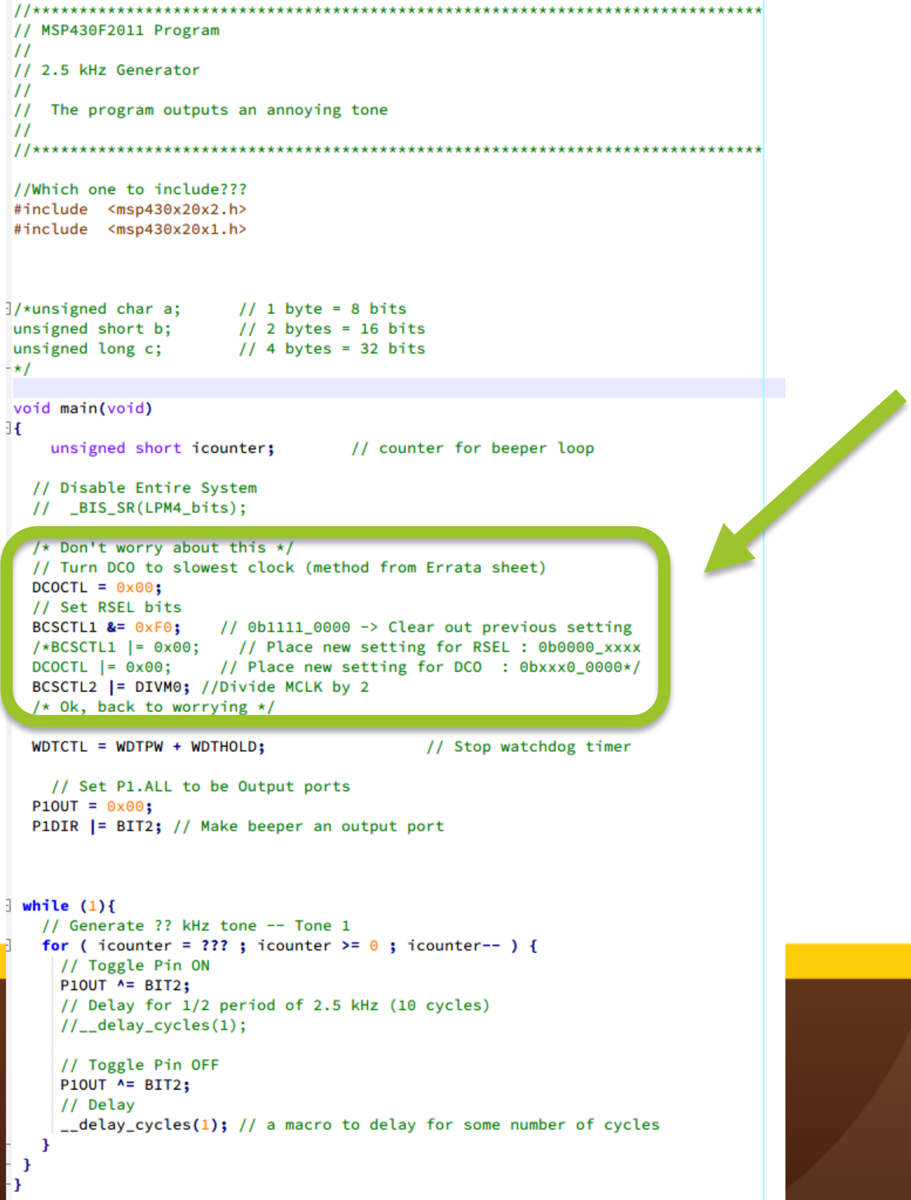
Make it so, and measure power

Remove from Launchpad, send 3.3V to the VDD line, measure the current



```
//*****************************************************************************
// MSP430F2011 Program
//
// 2.5 kHz Generator
//
//  The program outputs an annoying tone
//
//*****************************************************************************

//Which one to include???
#include  <msp430x20x2.h>
#include  <msp430x20x1.h>



/*unsigned char a;      // 1 byte = 8 bits
unsigned short b;       // 2 bytes = 16 bits
unsigned long c;        // 4 bytes = 32 bits
*/

void main(void)
{
    unsigned short icounter;        // counter for beeper loop

  // Disable Entire System
  //  _BIS_SR(LPM4_bits);

  /* Don't worry about this */
  // Turn DCO to slowest clock (method from Errata sheet)
  DCOCTL = 0x00;
  // Set RSEL bits
  BCSCTL1 &= 0xF0;    // 0b1111_0000 -> Clear out previous setting
  /*BCSCTL1 |= 0x00;    // Place new setting for RSEL : 0b0000_xxxx
  DCOCTL |= 0x00;     // Place new setting for DCO  : 0bxxx0_0000*/
  BCSCTL2 |= DIVM0; //Divide MCLK by 2
  /* Ok, back to worrying */

  WDTCTL = WDTPW + WDTHOLD;                    // Stop watchdog timer

    // Set P1.ALL to be Output ports
  P1OUT = 0x00;
  P1DIR |= BIT2; // Make beeper an output port


 while (1){
   // Generate ?? kHz tone -- Tone 1
   for ( icounter = ??? ; icounter >= 0 ; icounter-- ) {
     // Toggle Pin ON
     P1OUT ^= BIT2;
     // Delay for 1/2 period of 2.5 kHz (10 cycles)
     //__delay_cycles(1);

     // Toggle Pin OFF
     P1OUT ^= BIT2;
     // Delay
     __delay_cycles(1); // a macro to delay for s number of cycles
   }
 }
}
```

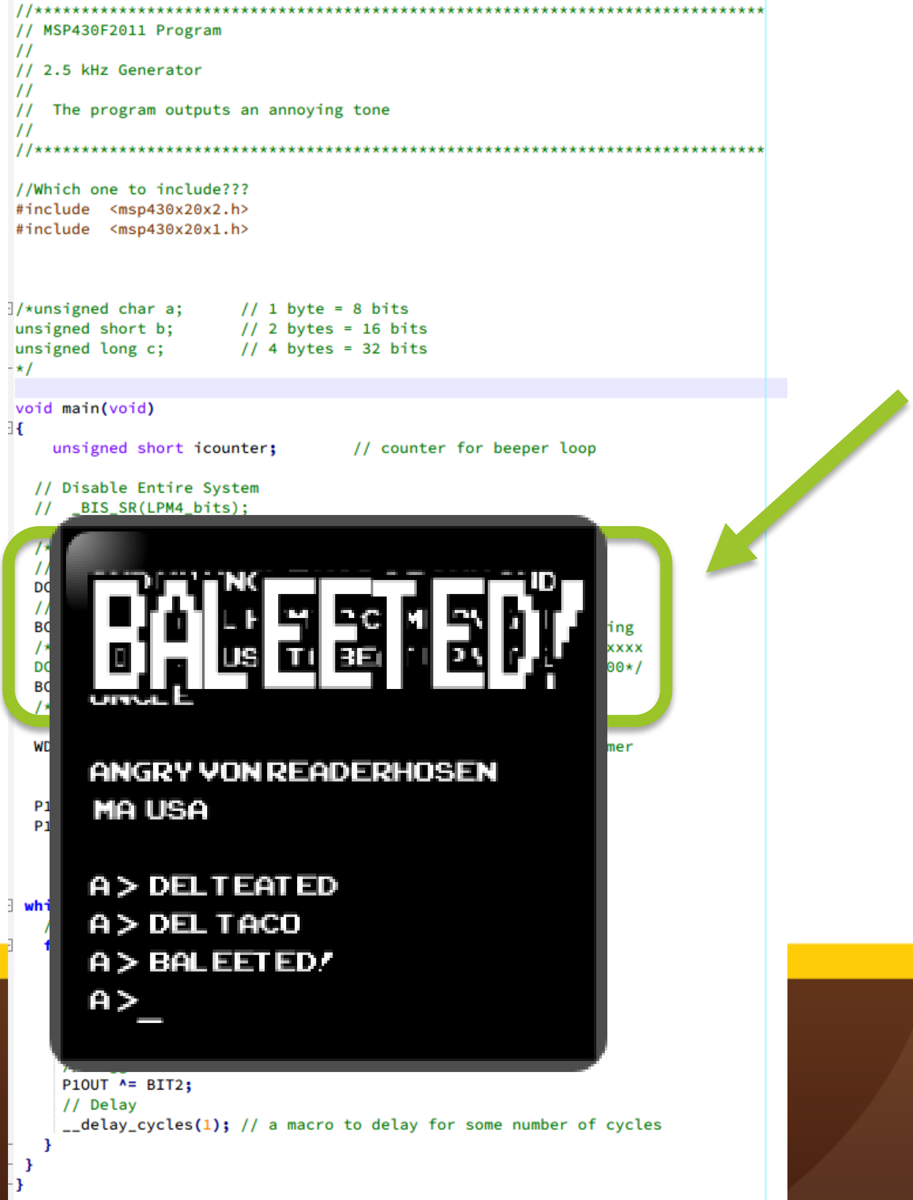
Hmm, what if we delete this nice little section and leave the MSP430 in the default clock mode?



```c
//******************************************************************************
// MSP430F2011 Program
//
// 2.5 kHz Generator
//
//  The program outputs an annoying tone
//
//******************************************************************************

//Which one to include???
#include  <msp430x20x2.h>
#include  <msp430x20x1.h>


/*unsigned char a;      // 1 byte = 8 bits
unsigned short b;       // 2 bytes = 16 bits
unsigned long c;        // 4 bytes = 32 bits
*/

void main(void)
{
    unsigned short icounter;        // counter for beeper loop

  // Disable Entire System
  //   _BIS_SR(LPM4_bits);

  /* Don't worry about this */
  // Turn DCO to slowest clock (method from Errata sheet)
  DCOCTL = 0x00;
  // Set RSEL bits
  BCSCTL1 &= 0xF0;     // 0b1111_0000 -> Clear out previous setting
  /*BCSCTL1 |= 0x00;     // Place new setting for RSEL : 0b0000_xxxx
  DCOCTL |= 0x00;      // Place new setting for DCO  : 0bxxx0_0000*/
  BCSCTL2 |= DIVM0; //Divide MCLK by 2
  /* Ok, back to worrying */

  WDTCTL = WDTPW + WDTHOLD;                 // Stop watchdog timer

    // Set P1.ALL to be Output ports
  P1OUT = 0x00;
  P1DIR |= BIT2; // Make beeper an output port



 while (1){
   // Generate ?? kHz tone -- Tone 1
   for ( icounter = ??? ; icounter >= 0 ; icounter-- ) {
     // Toggle Pin ON
     P1OUT ^= BIT2;
     // Delay for 1/2 period of 2.5 kHz (10 cycles)
     //__delay_cycles(1);

     // Toggle Pin OFF
     P1OUT ^= BIT2;
     // Delay
     __delay_cycles(1); // a macro to delay for some number of cycles
   }
 }
}
```

NEVER GONNA GIVE

Hmm, what if we delete this nice little section and leave the MSP430 in the default clock mode?



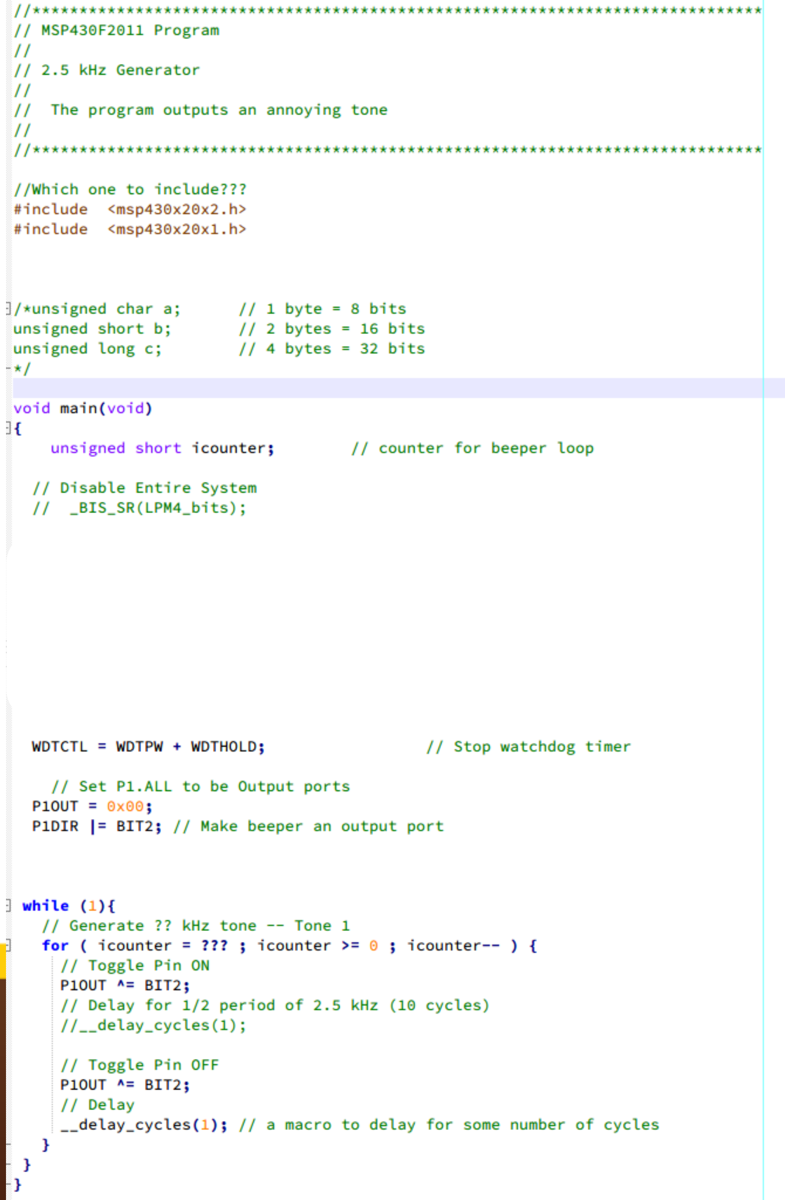
```
//*******************************************************************************
// MSP430F2011 Program
//
// 2.5 kHz Generator
//
//  The program outputs an annoying tone
//
//*******************************************************************************

//Which one to include???
#include  <msp430x20x2.h>
#include  <msp430x20x1.h>



/*unsigned char a;        // 1 byte = 8 bits
unsigned short b;        // 2 bytes = 16 bits
unsigned long c;         // 4 bytes = 32 bits
*/

void main(void)
{
    unsigned short icounter;        // counter for beeper loop

  // Disable Entire System
  //  _BIS_SR(LPM4_bits);

  /*
  //
  DC
  //
  BC                                                          ing
  /*                                                          xxxx
  DC                                                          00*/
  BC
  /*

  WD                                                          mer


  P1
  P1



 whi
    /
    f




       P1OUT ^= BIT2;
       // Delay
       __delay_cycles(1); // a macro to delay for some number of cycles
     }
   }
}
```

NEVER GONNA GIVE

Hmm, what if we delete this nice little section and leave the MSP430 in the default clock mode?

Now, adjust to keep roughly the same beeping output (beep tone, and beep duty cycle)

And …



```
//******************************************************************************
// MSP430F2011 Program
//
// 2.5 kHz Generator
//
//  The program outputs an annoying tone
//
//******************************************************************************

//Which one to include???
#include  <msp430x20x2.h>
#include  <msp430x20x1.h>


/*unsigned char a;       // 1 byte = 8 bits
unsigned short b;       // 2 bytes = 16 bits
unsigned long c;        // 4 bytes = 32 bits
*/

void main(void)
{
    unsigned short icounter;        // counter for beeper loop

  // Disable Entire System
  //   _BIS_SR(LPM4_bits);











  WDTCTL = WDTPW + WDTHOLD;                    // Stop watchdog timer

    // Set P1.ALL to be Output ports
  P1OUT = 0x00;
  P1DIR |= BIT2; // Make beeper an output port



 while (1){
   // Generate ?? kHz tone -- Tone 1
   for ( icounter = ??? ; icounter >= 0 ; icounter-- ) {
     // Toggle Pin ON
     P1OUT ^= BIT2;
     // Delay for 1/2 period of 2.5 kHz (10 cycles)
     //__delay_cycles(1);

     // Toggle Pin OFF
     P1OUT ^= BIT2;
     // Delay
     __delay_cycles(1); // a macro to delay for some number of cycles
   }
 }
}
```

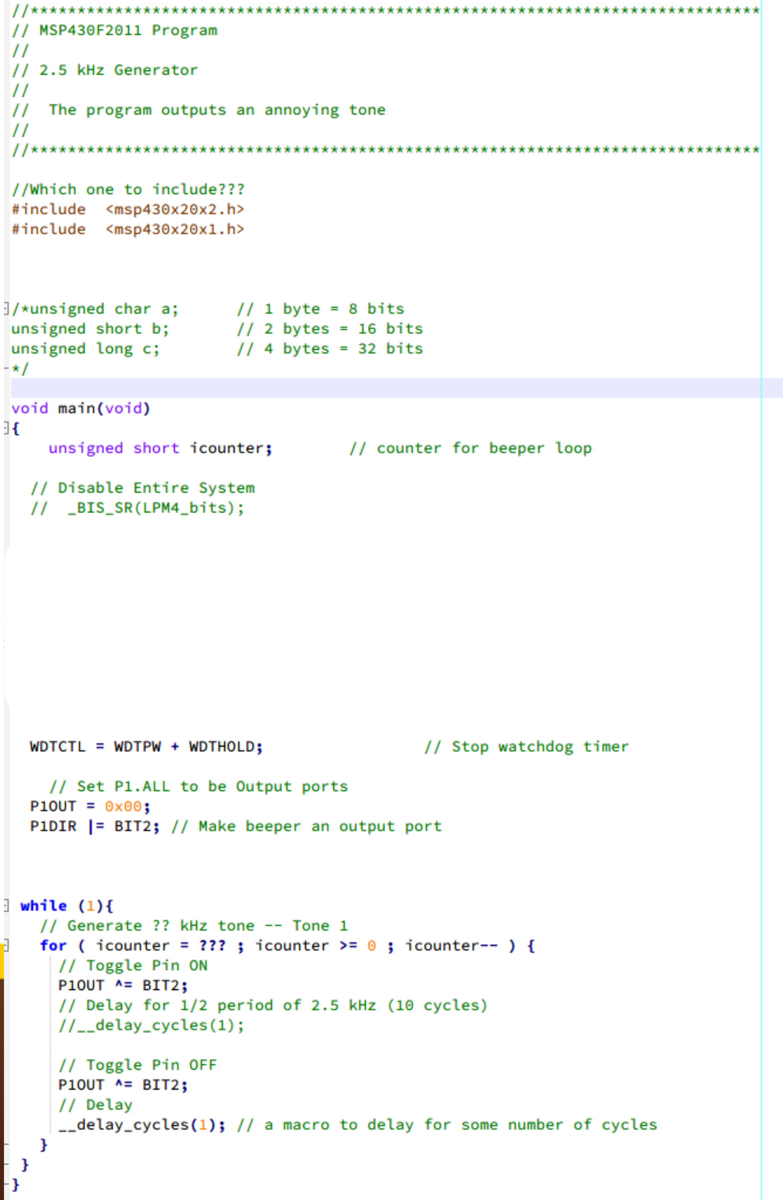
STEP 2

Hmm, what if we delete this nice little section and leave the MSP430 in the default clock mode?

Now, adjust to keep roughly the same beeping output (beep tone, and beep duty cycle)

And …

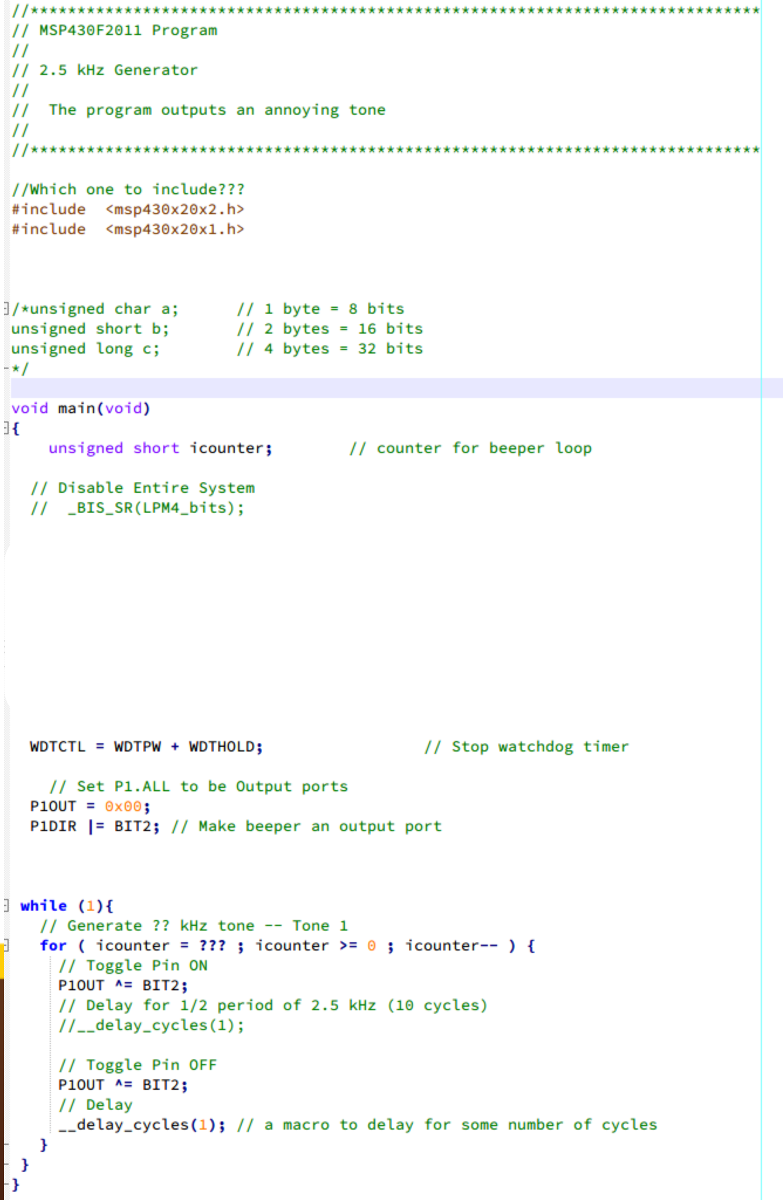
Measure the power



```
//******************************************************************************
// MSP430F2011 Program
//
// 2.5 kHz Generator
//
//  The program outputs an annoying tone
//
//******************************************************************************

//Which one to include???
#include  <msp430x20x2.h>
#include  <msp430x20x1.h>



/*unsigned char a;       // 1 byte = 8 bits
unsigned short b;       // 2 bytes = 16 bits
unsigned long c;        // 4 bytes = 32 bits
*/

void main(void)
{
    unsigned short icounter;        // counter for beeper loop

  // Disable Entire System
  //   _BIS_SR(LPM4_bits);









  WDTCTL = WDTPW + WDTHOLD;                  // Stop watchdog timer

    // Set P1.ALL to be Output ports
  P1OUT = 0x00;
  P1DIR |= BIT2; // Make beeper an output port



 while (1){
   // Generate ?? kHz tone -- Tone 1
   for ( icounter = ??? ; icounter >= 0 ; icounter-- ) {
     // Toggle Pin ON
     P1OUT ^= BIT2;
     // Delay for 1/2 period of 2.5 kHz (10 cycles)
     //__delay_cycles(1);

     // Toggle Pin OFF
     P1OUT ^= BIT2;
     // Delay
     __delay_cycles(1); // a macro to delay for some number of cycles
   }
 }
}
```

NEVER GONNA GIVE

Now we basically have 2 states:



```
//******************************************************************************
// MSP430F2011 Program
//
// 2.5 kHz Generator
//
//  The program outputs an annoying tone
//
//******************************************************************************

//Which one to include???
#include  <msp430x20x2.h>
#include  <msp430x20x1.h>



/*unsigned char a;      // 1 byte = 8 bits
unsigned short b;       // 2 bytes = 16 bits
unsigned long c;        // 4 bytes = 32 bits
*/

void main(void)
{
    unsigned short icounter;        // counter for beeper loop

  // Disable Entire System
  //   _BIS_SR(LPM4_bits);







  WDTCTL = WDTPW + WDTHOLD;                 // Stop watchdog timer

    // Set P1.ALL to be Output ports
  P1OUT = 0x00;
  P1DIR |= BIT2; // Make beeper an output port



 while (1){
   // Generate ?? kHz tone -- Tone 1
   for ( icounter = ??? ; icounter >= 0 ; icounter-- ) {
     // Toggle Pin ON
     P1OUT ^= BIT2;
     // Delay for 1/2 period of 2.5 kHz (10 cycles)
     //__delay_cycles(1);

     // Toggle Pin OFF
     P1OUT ^= BIT2;
     // Delay
     __delay_cycles(1); // a macro to delay for some number of cycles
   }
  }
}
```

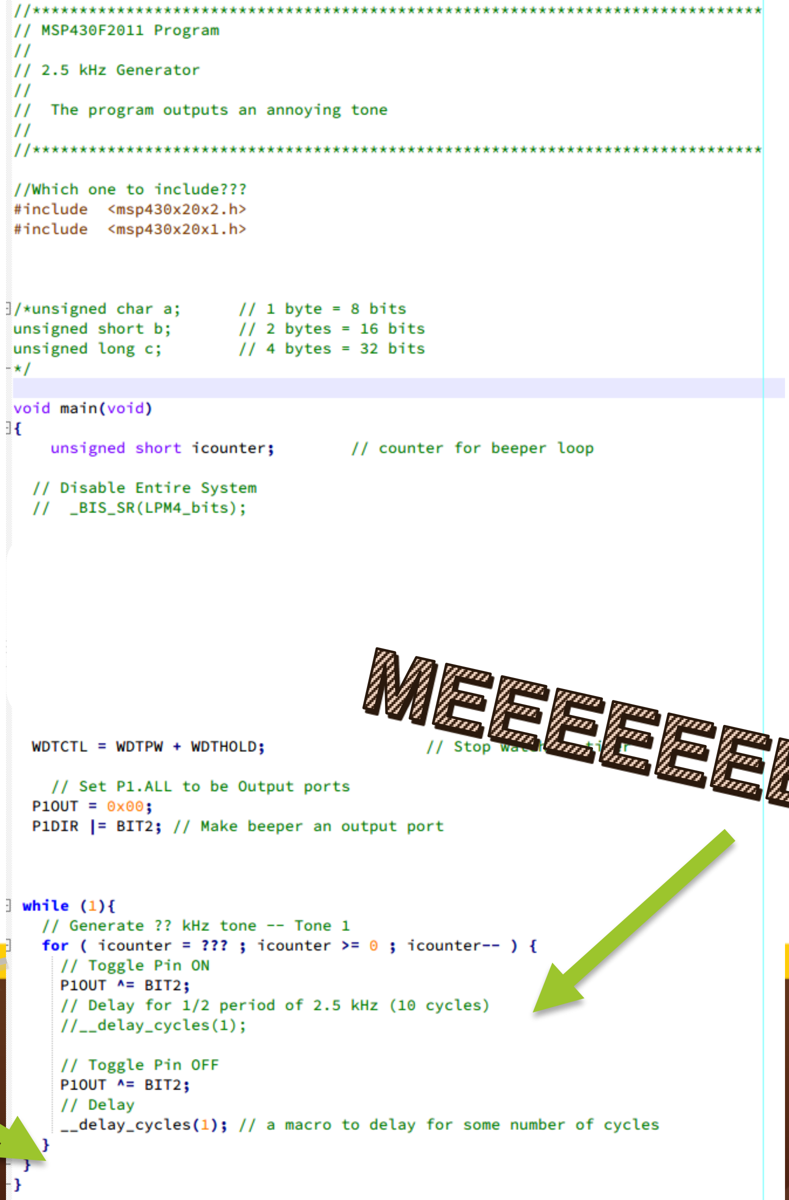
Now we basically have 2 states:

1) Doot – Doot’ing a beep.

2) Waiting until we can Doot –Doot a beep.

Zzzzzzzzzz

MEEEEEEEEEP



```
//******************************************************************************
// MSP430F2011 Program
//
// 2.5 kHz Generator
//
//  The program outputs an annoying tone
//
//******************************************************************************

//Which one to include???
#include  <msp430x20x2.h>
#include  <msp430x20x1.h>


/*unsigned char a;        // 1 byte = 8 bits
unsigned short b;       // 2 bytes = 16 bits
unsigned long c;        // 4 bytes = 32 bits
*/

void main(void)
{
    unsigned short icounter;        // counter for beeper loop

  // Disable Entire System
  //   _BIS_SR(LPM4_bits);







  WDTCTL = WDTPW + WDTHOLD;                 // Stop watchdog timer

    // Set P1.ALL to be Output ports
  P1OUT = 0x00;
  P1DIR |= BIT2; // Make beeper an output port



 while (1){
   // Generate ?? kHz tone -- Tone 1
   for ( icounter = ??? ; icounter >= 0 ; icounter-- ) {
     // Toggle Pin ON
     P1OUT ^= BIT2;
     // Delay for 1/2 period of 2.5 kHz (10 cycles)
     //__delay_cycles(1);

     // Toggle Pin OFF
     P1OUT ^= BIT2;
     // Delay
     __delay_cycles(1); // a macro to delay for some number of cycles
   }
 }
}
```

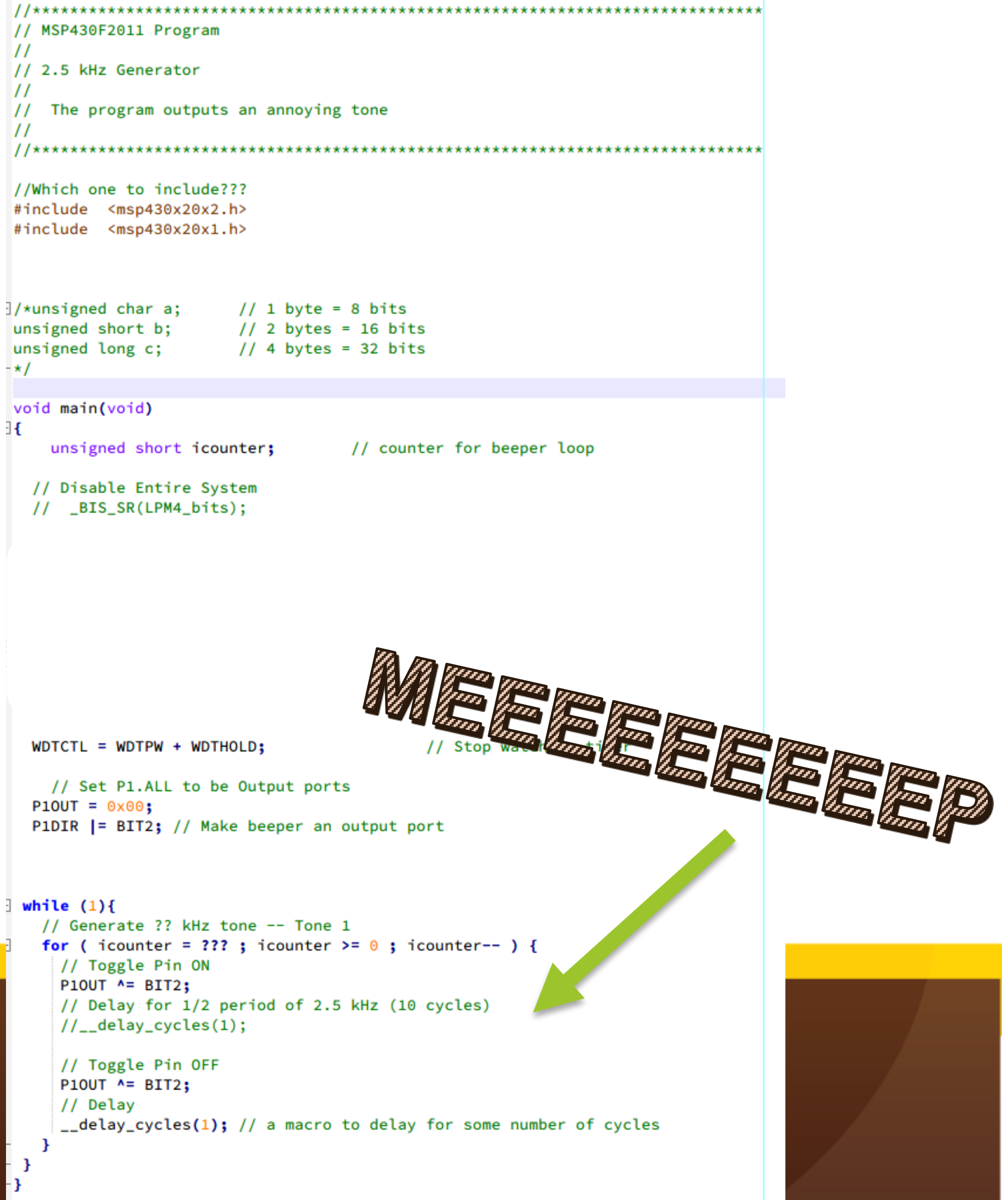
NEVER GONNA GIVE

Now we basically have 2 states:

1) Doot – Doot’ing a beep.

2) Waiting until we can Doot –Doot a beep.

If we made a program that ONLY **beeped**, how much power does it consume? What is its effective load?



```
//******************************************************************************
// MSP430F2011 Program
//
// 2.5 kHz Generator
//
//  The program outputs an annoying tone
//
//******************************************************************************

//Which one to include???
#include  <msp430x20x2.h>
#include  <msp430x20x1.h>


/*unsigned char a;        // 1 byte = 8 bits
unsigned short b;       // 2 bytes = 16 bits
unsigned long c;        // 4 bytes = 32 bits
*/

void main(void)
{
    unsigned short icounter;         // counter for beeper loop

  // Disable Entire System
  //   _BIS_SR(LPM4_bits);








  WDTCTL = WDTPW + WDTHOLD;                 // Stop watchdog timer

    // Set P1.ALL to be Output ports
  P1OUT = 0x00;
  P1DIR |= BIT2; // Make beeper an output port



 while (1){
   // Generate ?? kHz tone -- Tone 1
   for ( icounter = ??? ; icounter >= 0 ; icounter-- ) {
     // Toggle Pin ON
     P1OUT ^= BIT2;
     // Delay for 1/2 period of 2.5 kHz (10 cycles)
     //__delay_cycles(1);

     // Toggle Pin OFF
     P1OUT ^= BIT2;
     // Delay
     __delay_cycles(1); // a macro to delay for some number of cycles
   }
 }
}
```

NEVER GONNA GIVE

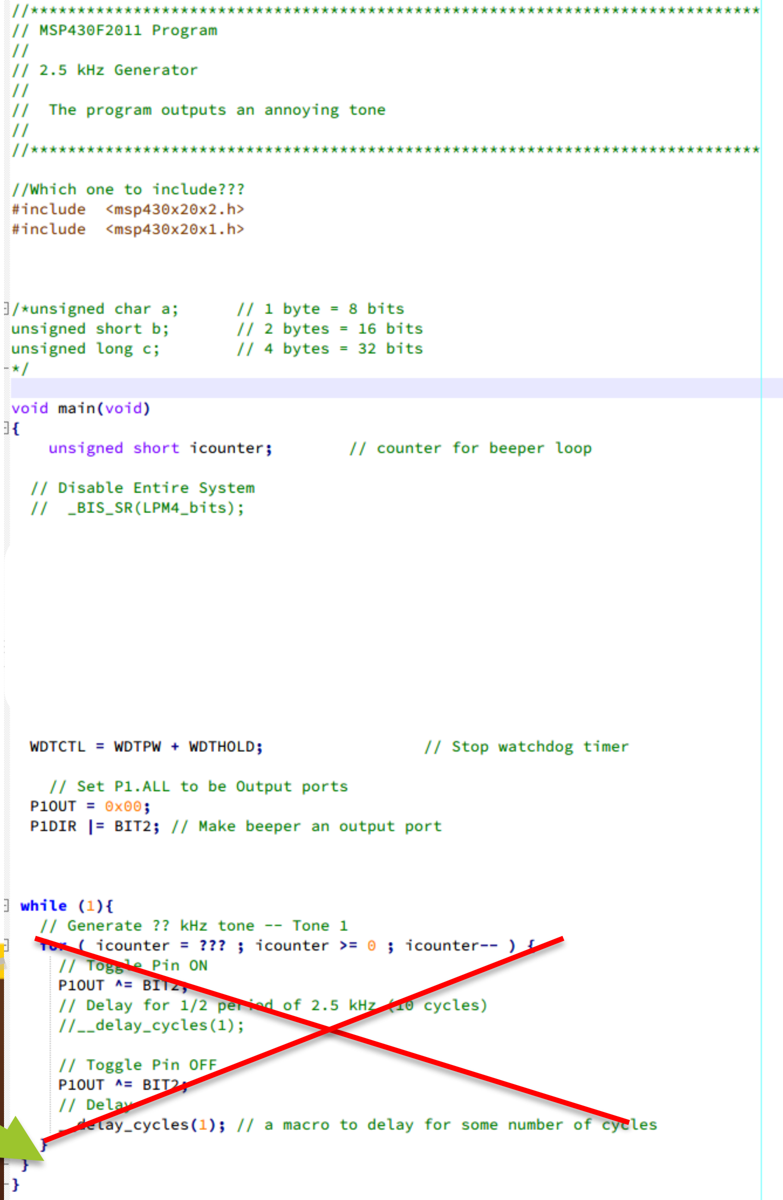
Now we basically have 2 states:

1) Doot – Doot'ing a beep.

2) Waiting until we can Doot –Doot a beep.

If we made a program that ONLY **waited**, how much power does it consume? What is its effective load?

Zzzzzzzzzz



```
//*****************************************************************************
// MSP430F2011 Program
//
// 2.5 kHz Generator
//
//  The program outputs an annoying tone
//
//*****************************************************************************

//Which one to include???
#include  <msp430x20x2.h>
#include  <msp430x20x1.h>



/*unsigned char a;      // 1 byte = 8 bits
unsigned short b;       // 2 bytes = 16 bits
unsigned long c;        // 4 bytes = 32 bits
*/

void main(void)
{
    unsigned short icounter;        // counter for beeper loop

  // Disable Entire System
  //   _BIS_SR(LPM4_bits);








  WDTCTL = WDTPW + WDTHOLD;                   // Stop watchdog timer

    // Set P1.ALL to be Output ports
  P1OUT = 0x00;
  P1DIR |= BIT2; // Make beeper an output port



 while (1){
   // Generate ?? kHz tone -- Tone 1
   for ( icounter = ??? ; icounter >= 0 ; icounter-- ) {
     // Toggle Pin ON
     P1OUT ^= BIT2;
     // Delay for 1/2 period of 2.5 kHz (10 cycles)
     //__delay_cycles(1);

     // Toggle Pin OFF
     P1OUT ^= BIT2;
     // Delay
     __delay_cycles(1); // a macro to delay for some number of cycles
   }
 }
}
```

SJT

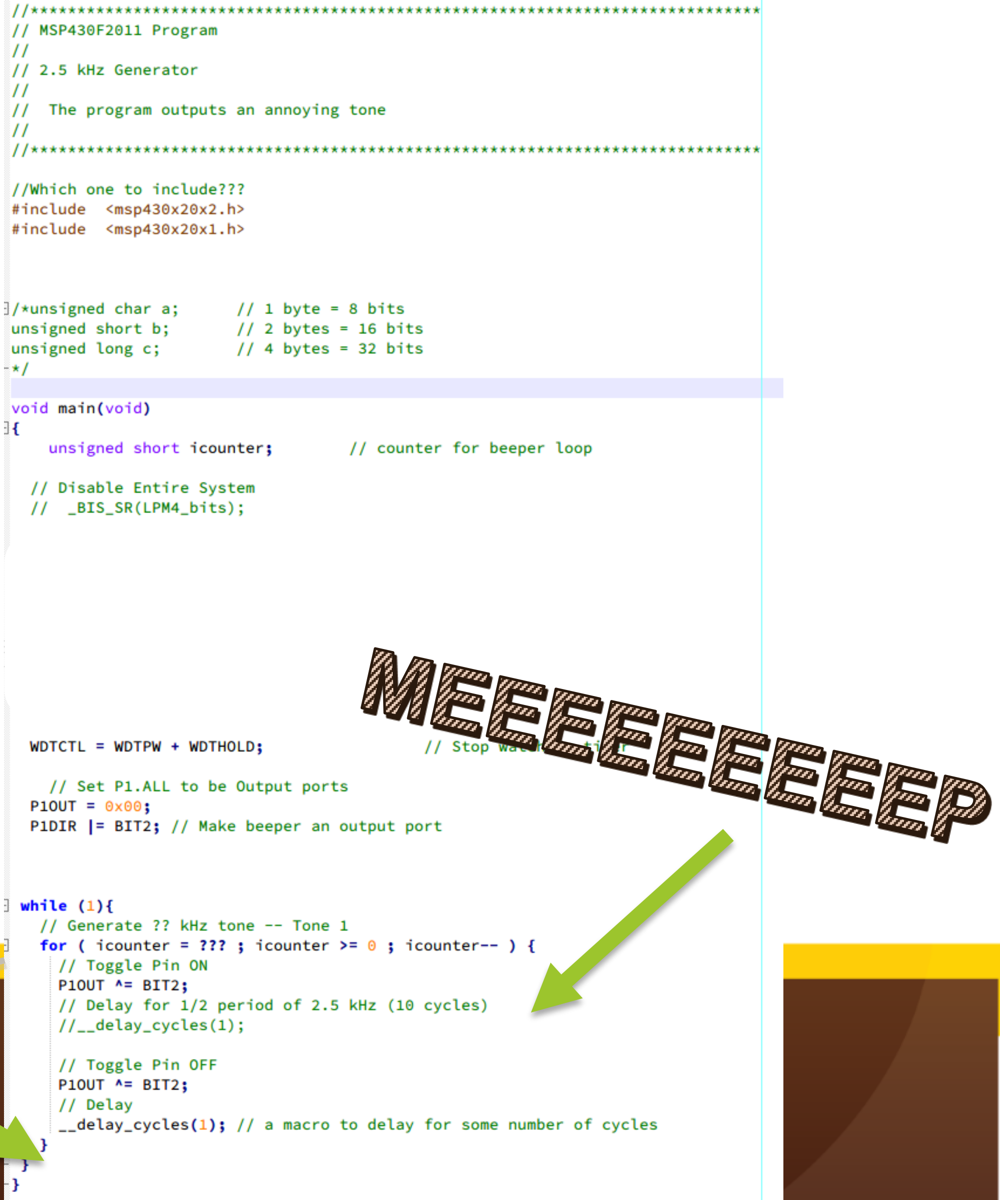
NEVER GONNA GIVE

Now we basically have 2 states:

1) Doot – Doot'ing a beep.

2) Waiting until we can Doot –Doot a beep.

If we made a program that spent X time making a BEEP and Y time waiting, we can create it to be an effective load!

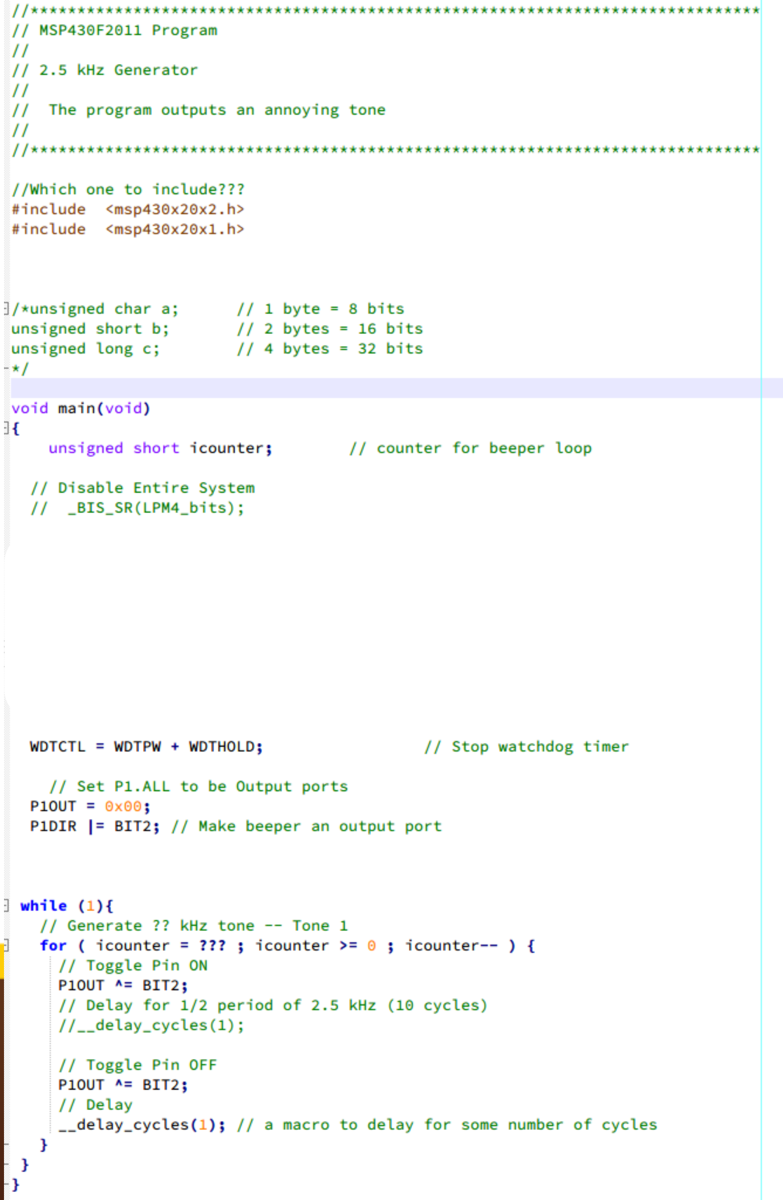


```
//****************************************************************************
// MSP430F2011 Program
//
// 2.5 kHz Generator
//
//  The program outputs an annoying tone
//
//****************************************************************************

//Which one to include???
#include  <msp430x20x2.h>
#include  <msp430x20x1.h>


/*unsigned char a;       // 1 byte = 8 bits
unsigned short b;       // 2 bytes = 16 bits
unsigned long c;        // 4 bytes = 32 bits
*/

void main(void)
{
    unsigned short icounter;        // counter for beeper loop

  // Disable Entire System
  //   _BIS_SR(LPM4_bits);







  WDTCTL = WDTPW + WDTHOLD;                 // Stop watchdog timer

    // Set P1.ALL to be Output ports
  P1OUT = 0x00;
  P1DIR |= BIT2; // Make beeper an output port



 while (1){
   // Generate ?? kHz tone -- Tone 1
   for ( icounter = ??? ; icounter >= 0 ; icounter-- ) {
     // Toggle Pin ON
     P1OUT ^= BIT2;
     // Delay for 1/2 period of 2.5 kHz (10 cycles)
     //__delay_cycles(1);

     // Toggle Pin OFF
     P1OUT ^= BIT2;
     // Delay
     __delay_cycles(1); // a macro to delay for some number of cycles
   }
  }
}
```

TO DO LIST

(It’s an assignment)

1) Count the cycles.
    1) Time on = for loop iterations * (total number of delay cycle)
    2) Time off = for loop iterations * (number of delay cycles)
2) Fill out the sheet that will:
    1) Alter beep code to pulse and measure power
    2) Remove default clock and measure power
    3) Control duty cycle to make an effective 7 ohm resistor



```
//******************************************************************************
// MSP430F2011 Program
//
// 2.5 kHz Generator
//
//  The program outputs an annoying tone
//
//******************************************************************************

//Which one to include???
#include  <msp430x20x2.h>
#include  <msp430x20x1.h>


/*unsigned char a;      // 1 byte = 8 bits
unsigned short b;       // 2 bytes = 16 bits
unsigned long c;        // 4 bytes = 32 bits
*/

void main(void)
{
    unsigned short icounter;        // counter for beeper loop

  // Disable Entire System
  //  _BIS_SR(LPM4_bits);







  WDTCTL = WDTPW + WDTHOLD;                   // Stop watchdog timer

    // Set P1.ALL to be Output ports
  P1OUT = 0x00;
  P1DIR |= BIT2; // Make beeper an output port


 while (1){
   // Generate ?? kHz tone -- Tone 1
   for ( icounter = ??? ; icounter >= 0 ; icounter-- ) {
     // Toggle Pin ON
     P1OUT ^= BIT2;
     // Delay for 1/2 period of 2.5 kHz (10 cycles)
     //__delay_cycles(1);

     // Toggle Pin OFF
     P1OUT ^= BIT2;
     // Delay
     __delay_cycles(1); // a macro to delay for some number of cycles
   }
 }
}
```