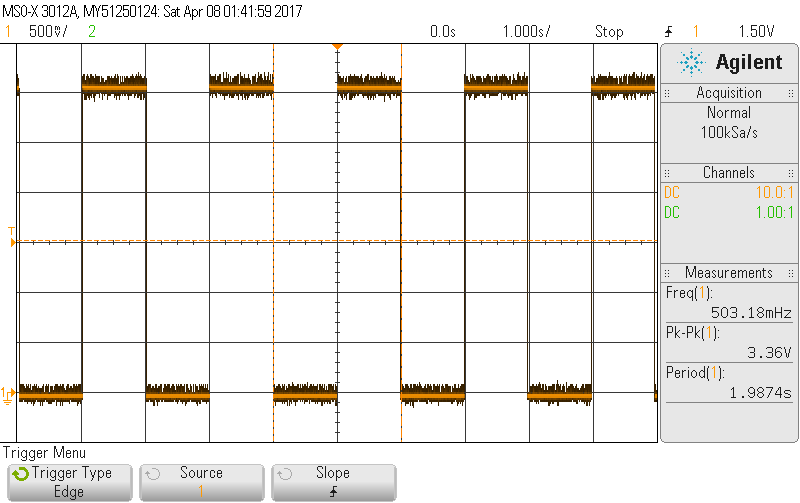
Brendan Baronia

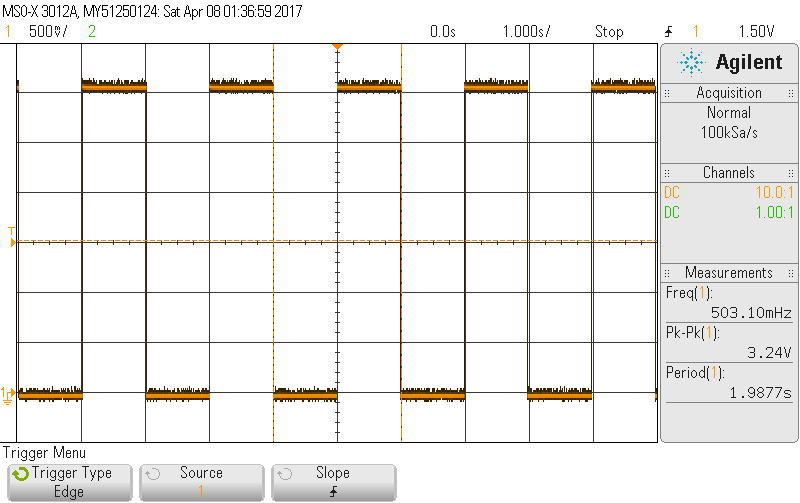
Melinda Ong

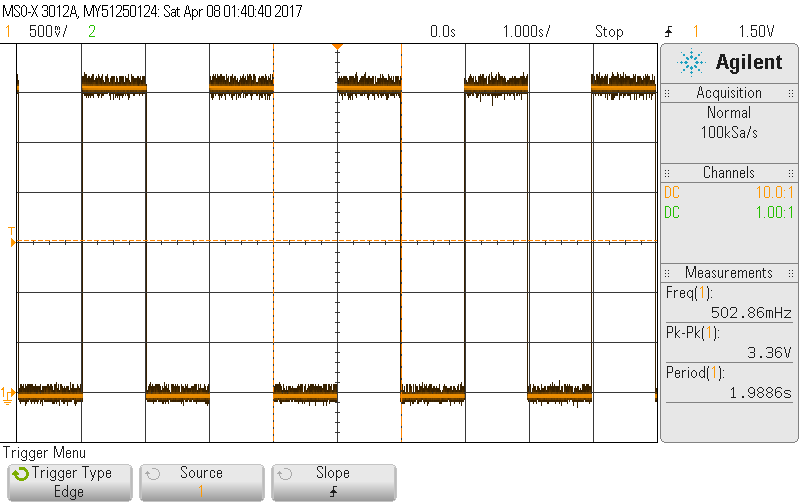
CPE 329-01

Assignment 1: Blinking LED, Clock Control, and Software Delay on MSP432

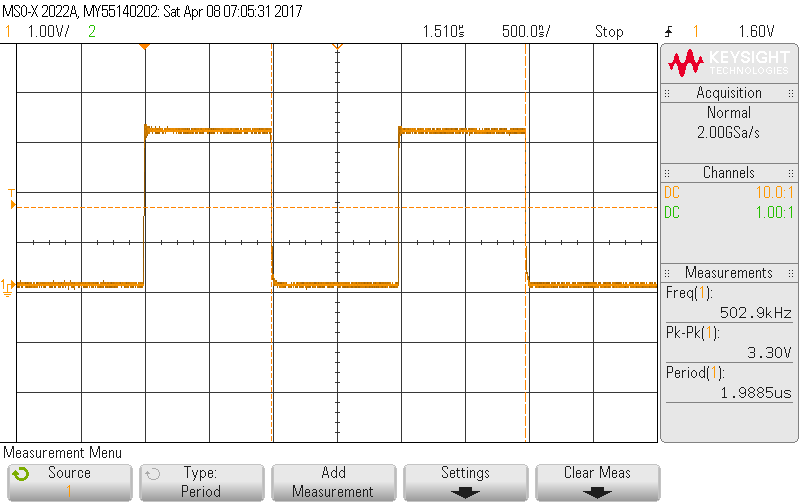
1. Youtube link of one-second pulse for 6 MHz, 12 MHz, and 24 MHz clock frequencies:  
   <https://youtu.be/bHAuoQa0bXo>
2. Documentation of accuracy of the one-second pulses with three clock frequencies:

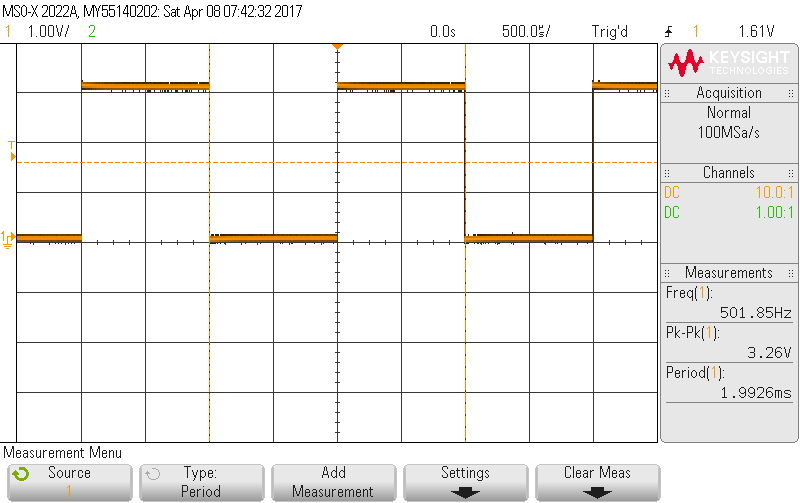
  
**Figure 1: One-second pulse with clock cycle at 6 MHz**

  
**Figure 2: One-second pulse with clock cycle at 12 MHz**

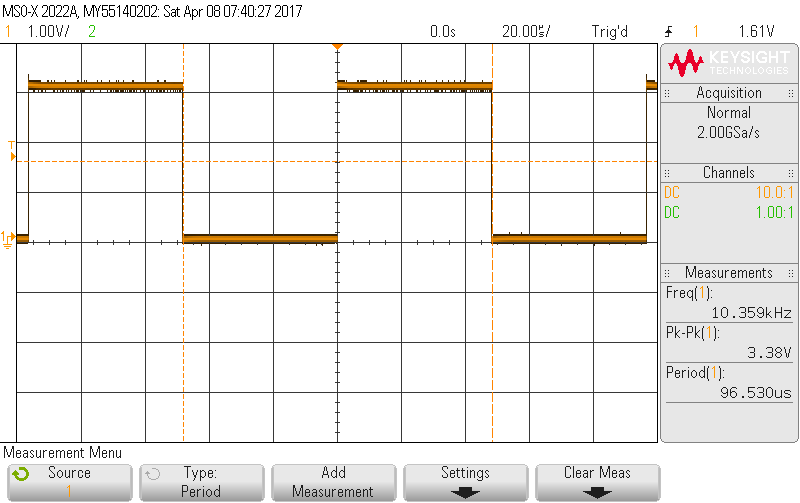


**Figure 3: One-second pulse with clock cycle at 24 MHz**

1.   
    **Figure 4: Screenshot of two 1 us pulses using normal triggering**

1. Shortest pulse generated:   
    

**Figure 5: Delay milliseconds function - 1.99 ms actual period, 2 ms specified**



**Figure 6: Delay nanoseconds function - 96 us actual period, 100 us specified**

2. C-Source code for system:

/\* P2\_1 Toggling green LED in C using header file register definitions.

\* This program toggles green LED for 0.5 second ON and 0.5 second OFF.

\* The green LED is connected to P2.1.

\* The LEDs are high active (a '1' turns ON the LED).

\*

\* XMS432P401R Rev C.

\*/

#include "msp.h"

#define FREQ\_1\_5\_MHZ 1500000

#define FREQ\_3\_MHZ 3000000

#define FREQ\_6\_MHZ 6000000

#define FREQ\_12\_MHZ 12000000

#define FREQ\_24\_MHZ 24000000

#define FREQ\_48\_MHZ 48000000

#define MILLI 1000

#define NANO 1000000000

#define FOR\_CYCLES 10

#define CALC\_TIME 35000;

void delay\_ms(int n, int frequency);

void delay\_ns(int n, int frequency);

void set\_DC0(int frequency);

int main(void) {

int frequency = FREQ\_24\_MHZ;

P2->SEL1 &= ~2; /\* configure P2.1 as simple I/O \*/

P2->SEL0 &= ~2;

P2->DIR |= 2; /\* P2.1 set as output pin \*/

set\_DC0(frequency);

//Method for two 1us pulses

/\*

int counter = 1, secCounter = 1, thirCounter = 1;

P2->OUT |= 2; /\* turn on P2.1 green LED \*/

//delay\_ns(1000000000, frequency);

while(counter--);

P2->OUT &= ~2; /\* turn off P2.1 green LED \*/

while (secCounter--);

//delay\_ns(1000000000, frequency);

P2->OUT |= 2; /\* turn on P2.1 green LED \*/

while (thirCounter--);

//delay\_ns(1000000000, frequency);

P2->OUT &= ~2; /\* turn off P2.1 green LED \*/

secCounter = 1;

while (0);

//delay\_ns(1000000000, frequency);

\*/

while (1) {

P2->OUT |= 2; /\* turn on P2.1 green LED \*/

delay\_ms(1, frequency);

P2->OUT &= ~2; /\* turn off P2.1 green LED \*/

delay\_ms(1, frequency);

}

}

/\* Delays a number of milliseconds

\* @param n is the number of ms to delay

\* @param frequency is the current clock frequency, as defined

\* at the top of this file

\*/

void delay\_ms(int n, int frequency) {

int i;

int delayTime = frequency / FOR\_CYCLES / MILLI;

for (i = delayTime \* n; i > 0; i--); /\* Delay 1 ms\*/

}

/\* Delays a number of nanoseconds

\* @param n is the number of ns to delay

\* @param frequency is the current clock frequency, as defined

\* at the top of this file

\*/

void delay\_ns(int n, int frequency) {

int i;

n -= CALC\_TIME;

//Find correct count to delay

int delayTime = 1.0 \* frequency \* n/ NANO / FOR\_CYCLES;

for (i = delayTime; i > 0; i--); /\* Delay n\*/

}

/\* set DCO to a defined frequency

\* @param frequency is the clock frequency to set, as defined

\* at the top of this file

\*/

void set\_DC0(int frequency) {

CS->KEY = CS\_KEY\_VAL; // unlock CS registers

CS->CTL0 = 0; // clear register CTL0

//Switch block to determine and set frequency

switch (frequency) {

case FREQ\_1\_5\_MHZ :

CS->CTL0=CS\_CTL0\_DCORSEL\_0;

break;

case FREQ\_3\_MHZ :

CS->CTL0=CS\_CTL0\_DCORSEL\_1;

break;

case FREQ\_6\_MHZ :

CS->CTL0=CS\_CTL0\_DCORSEL\_2;

break;

case FREQ\_12\_MHZ :

CS->CTL0=CS\_CTL0\_DCORSEL\_3;

break;

case FREQ\_24\_MHZ :

CS->CTL0=CS\_CTL0\_DCORSEL\_4;

break;

case FREQ\_48\_MHZ :

CS->CTL0=CS\_CTL0\_DCORSEL\_5;

break;

default:

CS->CTL0=CS\_CTL0\_DCORSEL\_1;

}

CS->CTL1 = CS\_CTL1\_SELA\_2 | CS\_CTL1\_SELS\_3 | CS\_CTL1\_SELM\_3; // select clock sources

CS->KEY = 0; // lock the CS registers

}