**Name(s) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ECE/CoE 1188: Cyber-Physical Systems Laboratory**

***Lab #2 Worksheet (60 points)***

Fill out all requested items, export to a PDF document and then upload this document to courseweb. Upload separate source files for each of the three parts

# **Part A – Digital Lock (C Implemenation)**

1. 5 pts **Digtal Lock Source Code (Digital\_Lock.c)**

**All functions implemented**

* void Initialize\_Port1( )
* void Initialize\_Port2( )
* uint8\_t Read\_Port1()
* void Write\_Port2( )
* main( )

# **Part B – Digitally Controlled Oscillator**

1. 5 pts **Digtally Controlled Oscillator Pseudocode**

**#include** <stdint.h>

**#include** "msp432p401r.h"

**void** **Delay\_1ms**(){

//setup of loop takes ~13 clock cycles

//clock rate is 3 MHz = 0.0033 ms

//to get 1ms, must go 3000 clock cycles

//13 taken by loop set up, 1 by breakdown -> loop takes up 2986 cycles

//loop iteration takes up ~24 cycles

//so you need 125 loop iterations to get ~3000 clock cycles

**int** i;

**for**(i = 0; i < 125; i++){

//each iteration of the loop takes ~18 cycles

**asm**(" add R0, R0, #0\n");

}

//breakdown take 1 instruction

}

**void** **Initialize\_Port1**(**void**)

{

P1SEL0 = 0x00;

P1SEL1 = 0x00;

P1REN = 0x12;

P1DIR = 0x00;

}

**void** **Initialize\_Port2**(**void**)

{

P2SEL0 = 0x00;

P2SEL1 = 0x00;

P2REN = 0x07;

P2DIR = 0x07;

}

uint8\_t **Read\_Port1**(**void**)

{

**return** P1IN;

}

**void** **Write\_Port2**(uint8\_t value)

{

P2OUT = value;

}

**int** **main**(**void**){

Initialize\_Port1();

Initialize\_Port2();

uint8\_t LED\_on = 0x01;

uint8\_t LED\_off = 0x00;

Write\_Port2(LED\_on);

**int** LED\_state = 1;

uint8\_t button\_bitmask = 0x02;

**while**(1){

uint8\_t port1\_val = Read\_Port1();

uint8\_t my\_button = port1\_val & button\_bitmask;

**if**(my\_button == 0x00){

Delay\_1ms();

Delay\_1ms();

**if**(LED\_state == 1){

LED\_state = 0;

Write\_Port2(LED\_off);

}

**else**{

LED\_state = 1;

Write\_Port2(LED\_on);

}

}

**else**{

Write\_Port2(LED\_on);

}

}

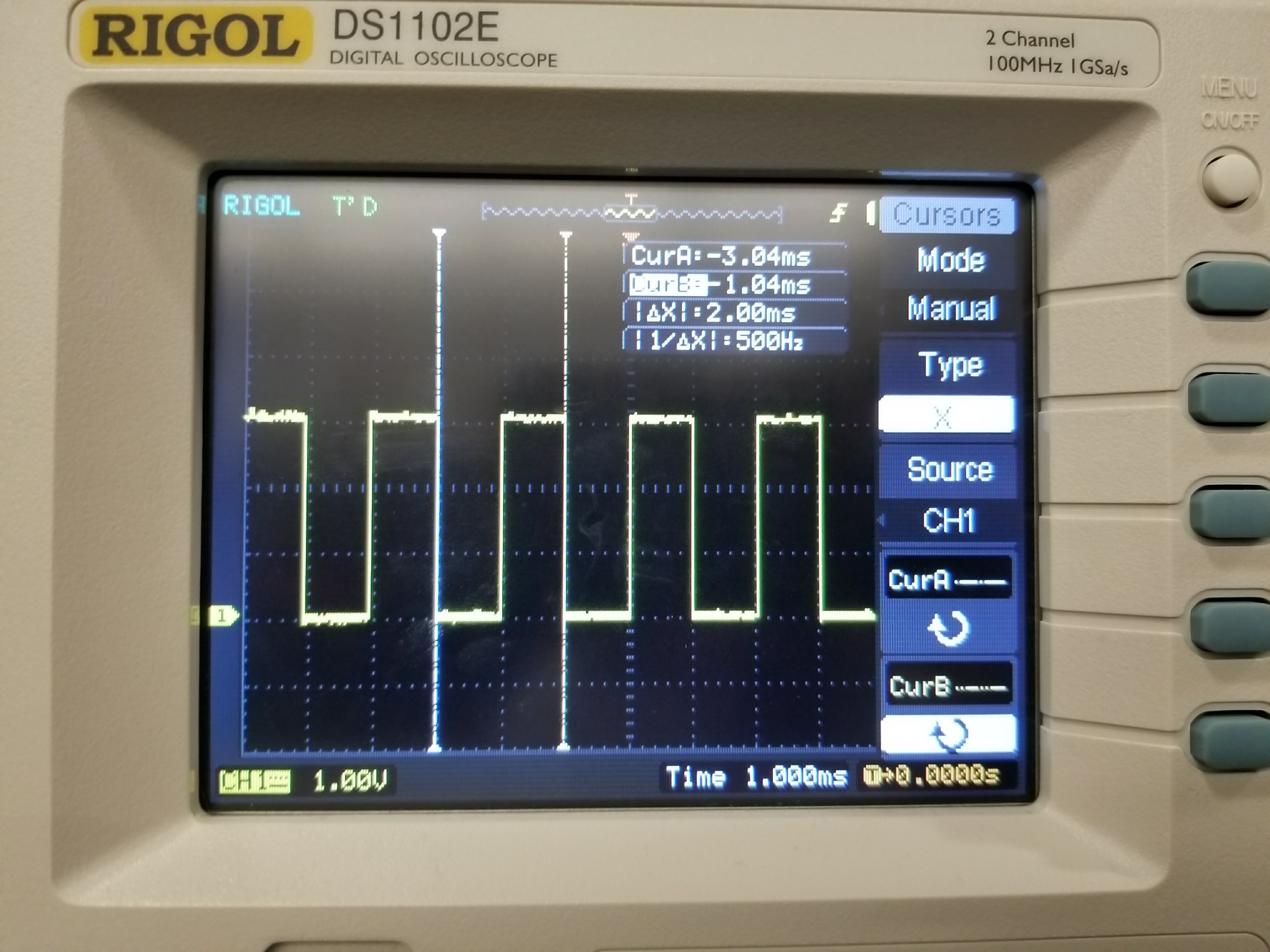
}

1. 5 pts **Digtally Controlled Oscillator Source Code (Digital\_Oscillator.c)**

**All functions implemented**

* Delay function using C/Assembly instructions
* main( ) makes use of delay function to create oscillator

1. 5 pts **Digtally Controlled Oscillator Operating a period of 2ms**



# **Part C – Digitally Controlled Oscillator w/ Debugging Instrumentation**

1. 5 pts **Oscillator w/ Systick and Software Debug Source (Digital\_Oscillator\_SysTick.c)**

**All functions implemented**

* Clock source other than default 3 MHz bus clock used
* Delay subroutine implemented using SYSTICK
* Debug\_Init ()
* Debug\_Capture ()
* In main( ) functions used to create oscillator and capture debugging information

1. 5 pts **5 Oscilloscope measurements of period (*f* = 8Hz) using C/Assembly Delay subroutine**

|  |  |  |
| --- | --- | --- |
| Reading | period |  |
| 1 | 124 ms |
| 2 | 124 ms |
| 3 | 125 ms |
| 4 | 124 ms |
| 5 | 124 ms |

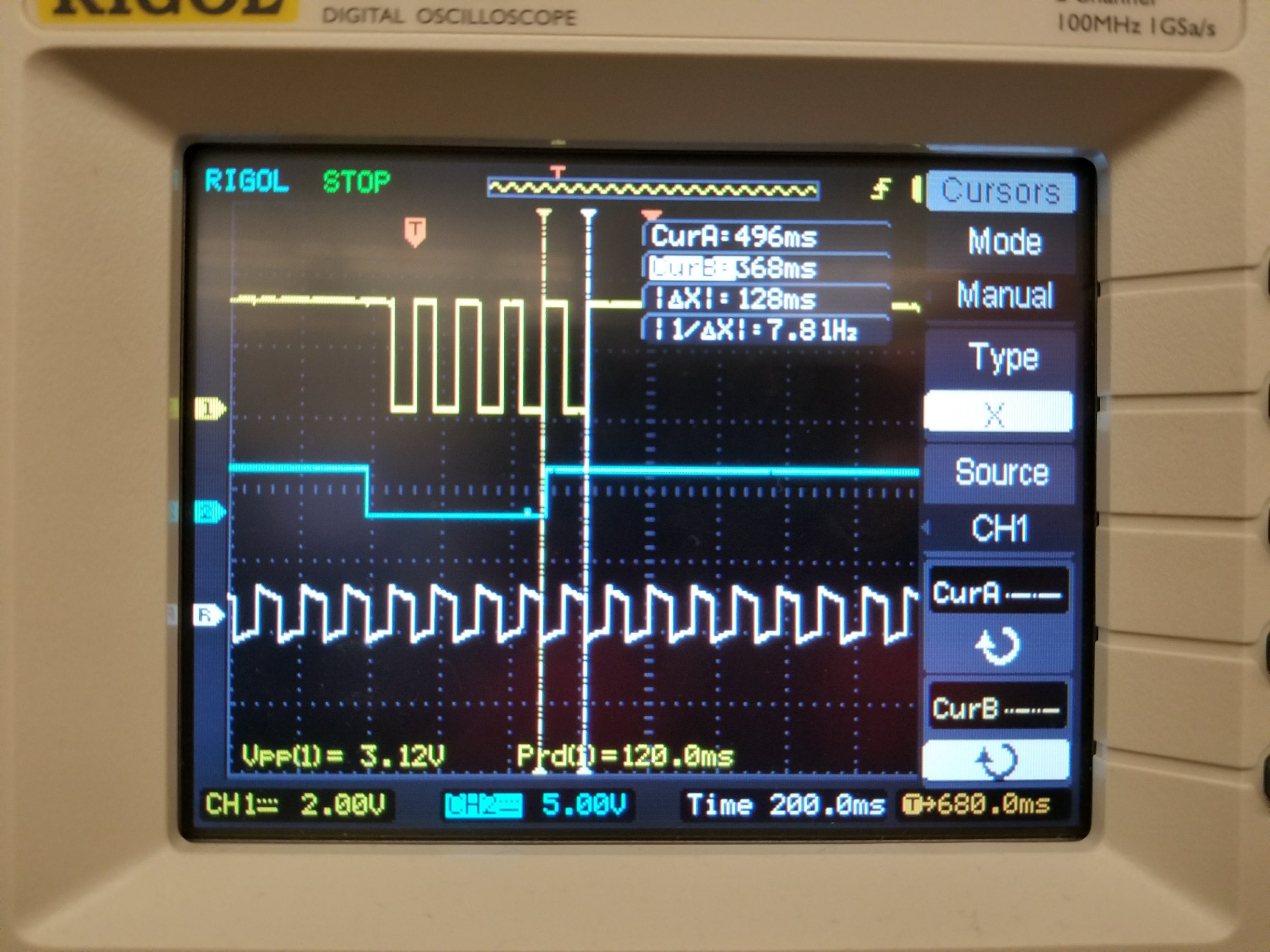
1. 5 pts **5 Oscilloscope measurements of period (*f* = 8Hz) using Systick subroutine**

|  |  |  |
| --- | --- | --- |
| Reading | period |  |
| 1 | 125 ms |
| 2 | 124 ms |
| 3 | 125 ms |
| 4 | 124 ms |
| 5 | 125 ms |

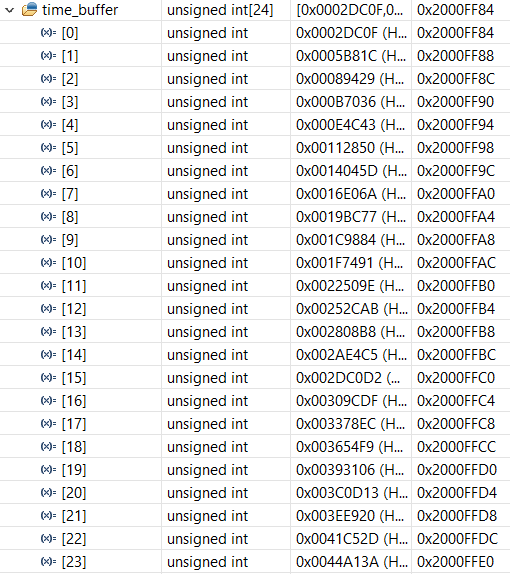
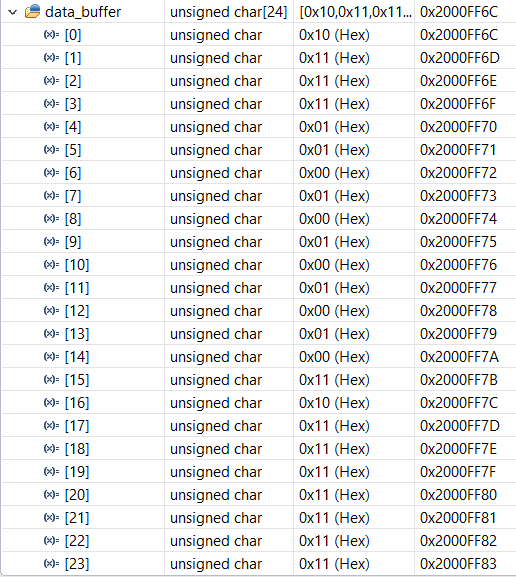
1. 5 pts **Estimation of Software Debugging Overhead**

56 instructions \* 2 = 112 cycles -> 20.833 \* 112 = 2333.3 ns = 2.33 ms overhead

1. 10 pts **Digtally Controlled Oscillator Operating at 8 Hz with heartbeat**



1. 5 pts **Software Debug Oscillator**



1. 5 pts **Software Debug Measurements of Oscillator**

0x2DC0F -> 187407 decimal

0x89429 -> 562217 decimal

562,217 – 187407 = 374810 cycles per period

Used 3MHz digital clock, each cycle = 1/(3\*10^6) = 33.3 ns

374810\*33.3ns = **124.9 ms = T**