**Design Assignment 6: Stepper Motor**

****

**Embedded Systems**

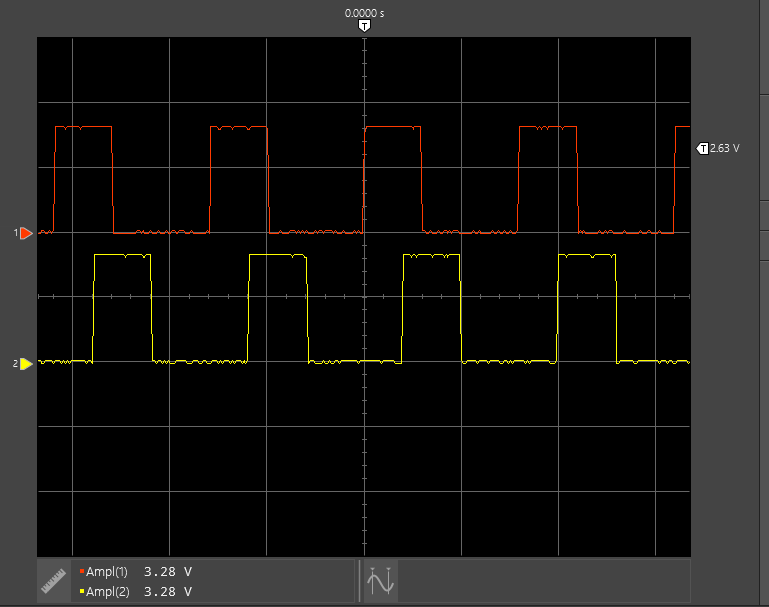
**ELC 411**

**Matthew Strickland and Jacob Levine**

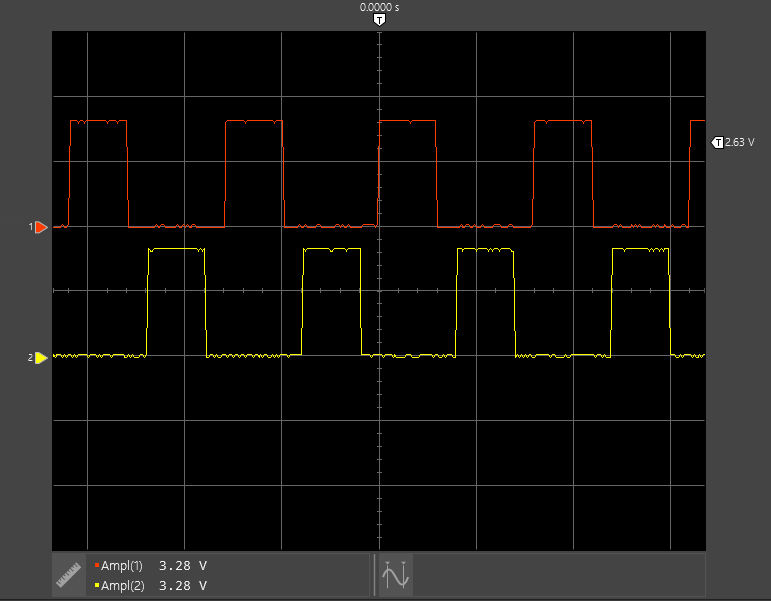
**Submission: 11/23/17**

1. **Diagrams**

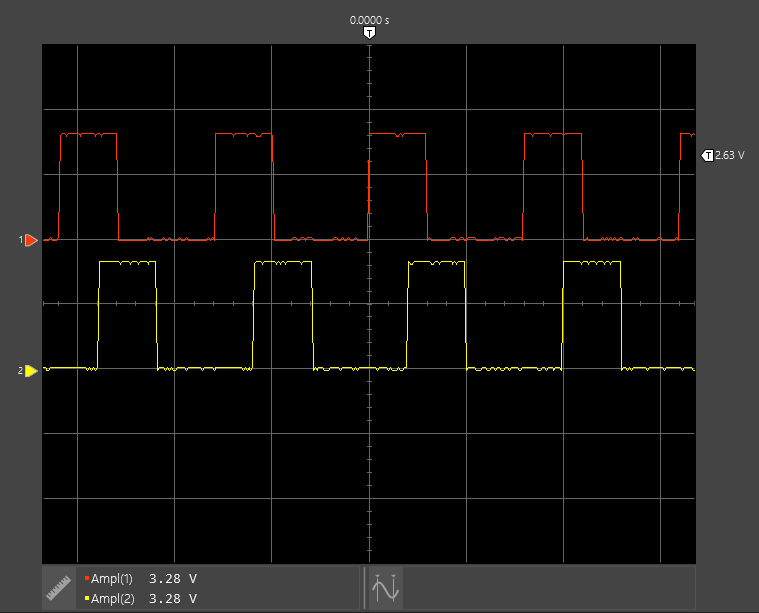
Figures 1 through 3 below show the D1, D2, and D3 (yellow) data lines with respect to D0 (red). As you move further through the ascending signals, a larger phase shift between the periodic signals is observed (in what appears to be 90 degree increments).



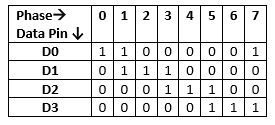
*Figure 1: D0/D1 = Blue/Pink*

**

*Figure 2: D0/D2 = Blue/Yellow*

**

*Figure 3: D0/D3 = Blue/Orange*

**

*Figure 4: Bit Mapping for angular position (0 to 360 degrees, 90 degree increments)*

|  |  |
| --- | --- |
| **Wires** | **Measured winding resistance** |
| **Blue/Pink** | **45Ω - no, each winding is to red!** |
| **Pink/Yellow** | **45Ω** |
| **Orange/Red** | **23Ω - yes!** |
| **Red/Any** | **23Ω** |

*Figure 5: Measured Winding Resistances*

1. **Questions**
   1. The Shaft goes 180° using the code we created for this lab but is capable of full bidirectional rotation depending upon the implementation
   2. One full revolution would take 4096 steps, however some have observed that the actual gear ratio differs from the manufacturer specified one (64:1) at 63.68395:1 so it would likely take 4076 steps unless the manufacturer has rectified the issue.
   3. Peak current would exist across any of the 23Ω winding resistances, which with a 5V supply yields a .217 A peak current from V=IR.
   4. This current cannot be driven directly from the PSoC 5LP, which has an absolute maximum gpio current rating of .041 A.

**Appendix:** Commented Source Code

#include <project.h>

int main(void)

{

   CyGlobalIntEnable; /\* Enable global interrupts. \*/

   int phase;

   int dir = 1;

   uint16\_t count;

   /\* Place your initialization/startup code here (e.g. MyInst\_Start()) \*/

   for(;;)

   {

       for(count = 0; count < 2048/8; count++)

       {

           for(phase = 0; phase < 8; phase++)

           {

// Extra indentation here!

                   if (dir == 1) // if dir positive, Rotation = CW

                   {

                   switch (phase)

                       {

// case looks at phase and based on position writes to data lines

                           case 0:

                               DigOut\_Write(1); // write 0001

                               break;

                           case 1:

                               DigOut\_Write(3); // write 0011

                               break;

                           case 2:

                               DigOut\_Write(2); // write 0010

                               break;

                           case 3:

                               DigOut\_Write(6); // write 0110

                               break;

                           case 4:

                               DigOut\_Write(4); // write 0100

                               break;

                           case 5:

                               DigOut\_Write(12); // write 1100

                               break;

                           case 6:

                               DigOut\_Write(8); // write 1000

                               break;

                           case 7:

                               DigOut\_Write(9); // write 1001

                               break;

                       }

                   }

                   if (dir == 0)

                   {

                   switch (phase) // if dir negative, rotation = CCW, negate previous values

                       {

// case looks at phase and based on position writes to data lines

                           case 7:

                               DigOut\_Write(1); // write 0001

                               break;

                           case 6:

                               DigOut\_Write(3); // write 0011

                               break;

                           case 5:

                               DigOut\_Write(2); // write 0010

                               break;

                           case 4:

                               DigOut\_Write(6); // write 0110

                               break;

                           case 3:

                               DigOut\_Write(4); // write 0100

                               break;

                           case 2:

                               DigOut\_Write(12); // write 1100

                               break;

                           case 1:

                               DigOut\_Write(8); // write 1000

                               break;

                           case 0:

                               DigOut\_Write(9); // write 1001

                               break;

                       }

                   }

               CyDelay(10);

           }

       }

       CyDelay(500);

       dir = 1 - dir;

   }

}

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Expected** | **Points** | **Pts. Available** |
| Cover sheet |  | 0.5 | 0.5 |
| 4 waveforms on single time axis |  | 1 | 1 |
| Measured winding resistance | About 23 ohms | 0.8 | 1 |
| How far does the shaft go? | Half way | 1 | 1 |
| How many steps for full rev? | 4096 steps (technically half steps) | 1 | 1 |
| Peak current? | 5/Rwinding, about 200 mA | 1 | 1 |
| Can it be driven directly? | No - maximum GPIO current is either 30 mA or 41 mA, depending on how one interprets the specification (appears to be ambiguous). Typically negative currents indicate current into pin (sinking) and positive are currents flowing out from pin (sourcing). But also typical chips (including PSoC) are better at current sinking than sourcing! The stepper application requires current sinking, since the common point is connected to VDD. | 1 | 1 |
| Code - fully commented and properly formated |  | 3.3 | 3.5 |
| Bonus | Find max motor speed | 0 |  |
| **TOTAL** |  | **9.6** | **10** |