DESN 204 Advanced 3D Modeling and Prototyping Fall 2013

**Individual Project**

**Decorative Clock Partially Manufactured with 3D Printing**

**Stephen Gallant**

An artistic clock design partially manufactured with Additive Manufacturing is presented. The clock gear and gear frame are 3D Printed on a Dimension printer, while the bearings, off-the-shelf hardware, and electronics are made via traditional manufacturing. A short description of the design is presented.

**0.0 Introduction**

A decorative clock inspired by an image on the web (1) is designed and prototyped. The clock gear and gear box is 3D printed using a Dimension FDM printer. A decision to use traditionally manufactured and readily available bearings is taken to reduce the risk of poor 3D tolerance and, thus, poor friction performance. Similarly, the traditionally manufactured assembly hardware is chosen from off-the-shelf providers, since the cost and efficiencies of these items has long been demonstrated.

The drive to the clock is configured via a stepper motor, stepper motor drive, micro-controller, and power supply. All of these drive elements are sourced from traditional manufacturing since Additive Manufacturing technology for these items is at the earliest research stage, and possible will never commercially be manufacture via Additive Manufacturing.

The drive timing is derived from crystals oscillator technology and the use of the timer/counter features of the modern micro-controller. A very accurate high rate clock is set up via a crystal and crystal oscillator and is then scaled to a lower frequency suitable for the timer/counter. The timer/counter is set up to generate an accurate time pulse, derived from the crystal oscillator, and used to control the stepper motor driver. The stepper motor driver, in turn, drives the motor.

**1.0 Mechanics**

The mechanics design consists of a set of gears to divide down the rotation of the motor to a minute hand and hour hand both formed in the image of gears. The minute hand gear and the hour hand gear are decoratively figured. (2) The gears and gear box are 3D printed while all other elements are traditionally manufactured.

**1.1 Gears and Gear Ratios**

The gear ratios, gear sizing, and number of gears were chosen to minimize design complexity, minimize motor drive complexity, minimize gear meshing errors due to gear pitch, and to ensure that the minute hand and hour hand gears are made of a suitable size.

Given the considerations above, the gears were chosen not to have less than ten teeth, to have smaller pitch, and to be driven from the stepper motor in its fundamental step mode. Further, all gears had to be sized in order to sufficiently step down he rotation speed of the motor, and yet to have the hour hand not too large in diameter. The clock assembly is shown in Figure 1.

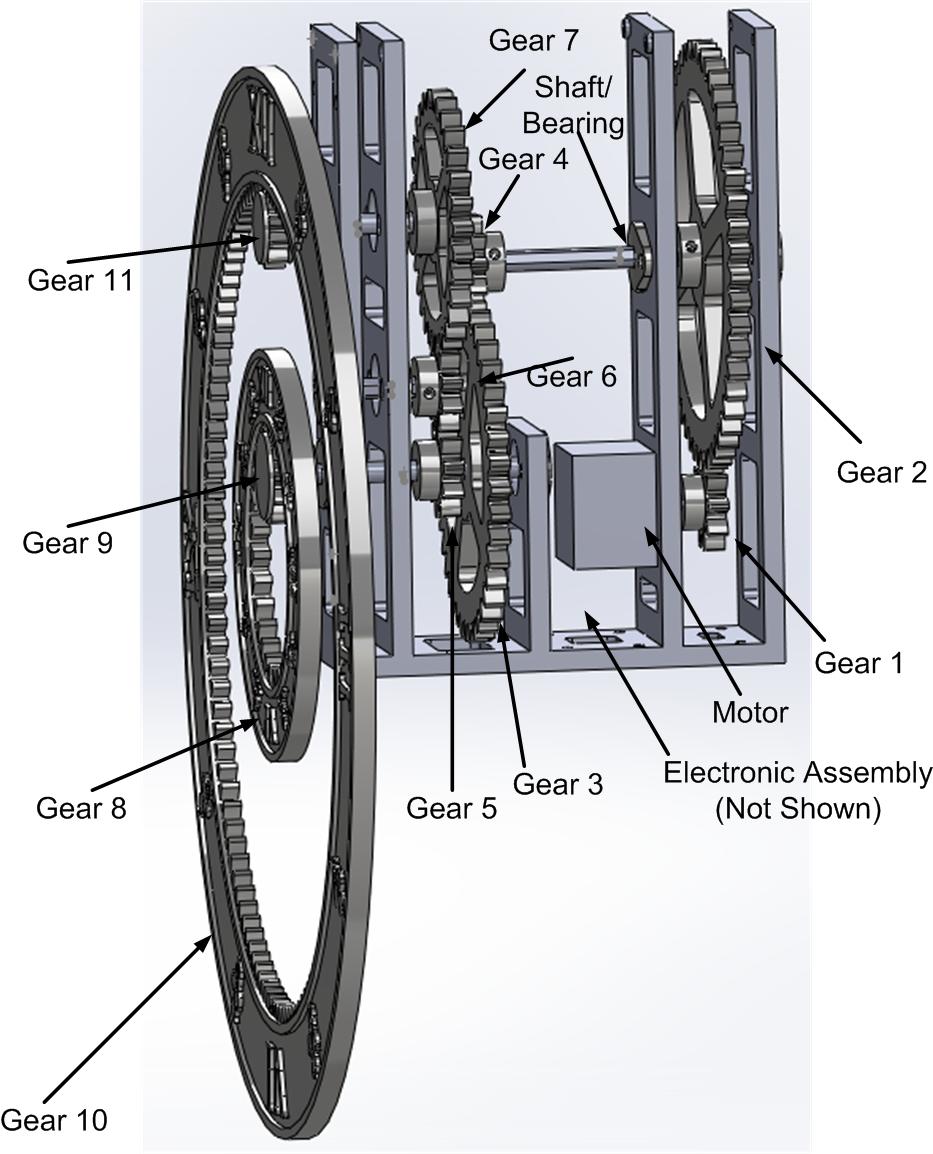


Figure 1. Clock Assembly.

Design simplicity influenced the desired motor speed. This is chosen as 1 rotation per minute. We need 60 motor rotations for one minute hand rotation and 60X12 motor rotations for one hour hand rotations. These speed reductions have been separated into the gears, Table 1, as follows.

The gain reduction in the motor to minute gearing is Gain 1,2 x Gain 4,3 x Gain 9,8. This is 10/50 x 10/40 x 10/30 = 1/60 as needed. Similarly, we need a 1/(60x12) gain reduction in the motor to hour gearing. (See Table 2.) Gain 1,2 x Gain 4,3 x Gain 5,7 x Gain 11,10. This is 10/50 x 10/40 x 10/30 x 10/120 = 1/720.

Note that Gear 6 is just an idler gear that corrects for proper clockwise rotation of the hour hand. This gear does not play a role in the magnitude of the gain.

Table 1. Minute Hand Gears and Gear Ratio.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Gear and Gear Ratio | | | | | |
| Gear 1 | Gear 2 | Gear 4 | Gear 3 | Gear 9 | Gear 8 |
| Gain 1,2 | | Gain 4,3 | | Gain 9,8 | |
| 10/50 | | 10/40 | | 10/30 | |

Table 2. Hour Hand Gears and Gear Ratio.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Gear and Gear Ratio | | | | | | | |
| Gear 1 | Gear 2 | Gear 4 | Gear 3 | Gear 5 | Gear 7 | Gear 11 | Gear 10 | |
| Gain 1,2 | | Gain 4,3 | | Gain 5,7 | | Gain 11,10 | | |
| 10/50 | | 10/40 | | 10/30 | | 10/120 | | |

NOTE: Gear 6 is simply an idler gear and does not play a role in the gear gain.

**1.2 Gear Design**

All gears are spur gears and designed using Rush Gears(3) web gear design application. The gears designs are imported into Solidworks where they are modifies as needed.

**1.3 Bearing Design**

All bearings are chosen from McMaster Carr (4) where they have provided Solidworks models. Several of the bearings required a milling operation so that the shafts will pass through them.

**1.4 Shaft Design**

All shafts are chosen from McMaster Carr (4) where they have provided Solidworks models. Several of the shafts required cutting and grinding operations to size the shafts.

**1.5 Mounting Hardware**

All mounting hardware is chosen from McMaster Carr (4) where they have provided Solidworks models.

**1.6 Gear Box Design**

The gear box is designed around the available hardware, shafts, bearings, and gear designs. The gear box is designed in Solidworks, where the primary concern is the quick turn-around of a prototype.

**2.0 Electronics**

The primary design concerns for the electronics is simplicity, reduction of risk, and a quick turn-around of a prototype. Since the motor requires a periodic drive signal to set the clock in motion, and this signal requires accuracy and low drift over time, a micro-controller is chosen for the design. Modern micro-controllers have high quality clock generation from crystal oscillator sources and precision timer/counters that enable high quality waveform generation.

A motor drive circuit is chosen from Pololu(5) for speed to prototype concerns and their use of quality TI parts. (6) A circuit board containing a power supply, micro-controller(7), crystal oscillator, programming port, driver board port, and motor drive port is designed in Altium. This circuit board is paired with an off-the-shelf ‘wall wart’ power supply. (8) The clock assembly is designed to mount this circuit board.

**2.1 Stepper Motor Drive Signal**

The Pololu #1207 (9) stepper motor is chosen for the reasons outlined above. This motor’s fundamental step mode is 200 steps per revolution. It is chosen that the motor will make one rotation for every minute. Thus 200 steps/min or 200 steps/min x 1/60 min/s = 3 and 1/3 Hz is the frequency that the motor needs to be pulsed to drive the clock gear box.

**2.2 Micro-Controller Clock Signal**

The Tiny44A micro-controller is programmed to run from a crystal clock source. The micro-controller is programmed to scales the oscillator frequency to the system clock by 1/8. The micro-controller is programmed to use its Timer/Counter1 in CTC Waveform Generation mode using the system clock.

The programming of the micro-controller, as described above, is chosen based on the MATLAB script listed in the Appendix. This script mathematically models the micro-controller oscillator timer system to find all the combinations of parameters that generate 3 and 1/3 Hz. The script generates more than 100 possibilities. The Table in the Appendix shows listings for 12.288MHz crystal. This crystal frequency is chosen since it is available in very accurate and low drift versions.

The waveform frequency is calculated from the following formula where Fwave is the frequency of the output waveform, Fosc is the crystal oscillator frequency, Fscale is the system clock scale, Tscale is the Timer/Counter scale, and OCR1A is a register that must be written to to generate the desired frequency.

Fwave = (Fosc/Fscale)/[2 x Tscale x (1 + OCR1A)]

Solving for OCR1A and using the values listed in the table in the Appendix, we see that OCR1A is 28799. This value of OCR1A is programmed into the micro-controller

The waveform of the scaled clock signal, and the waveform output is shown in the Appendix along with the C source code. We see that the system clock frequency is measured as 12.288MHz/8 and the output frequency is measured as 3.333… Hz.

**2.3 Motor Driver Board**

The Pololu #1207 motor is driven by the Pololu DRV8824 stepper motor drive. The DRV8824 intelligently translate the 3.333Hz micro-controller waveform into an H-Bridge output suitable for driving the motor. The main control inputs of the DRV8824 is Step and Direction. The step is connected to the microcontroller 3.333Hz waveform, and the Direction can be tied hi or lo depending on the needed direction of rotation.

**2.4 Motor Drive Board Electronics Schematic**

The motor drive schematic is shown in Figure 2. The ‘Wall Wart’ jack power input is shown followed by the motor power supply and the logic power supply. The micro-controller is shown with its crystal oscillator. A programming port, motor drive board port, and the motor connection ports are shown. There are a set of buttons. It is possible to use this board without the buttons.

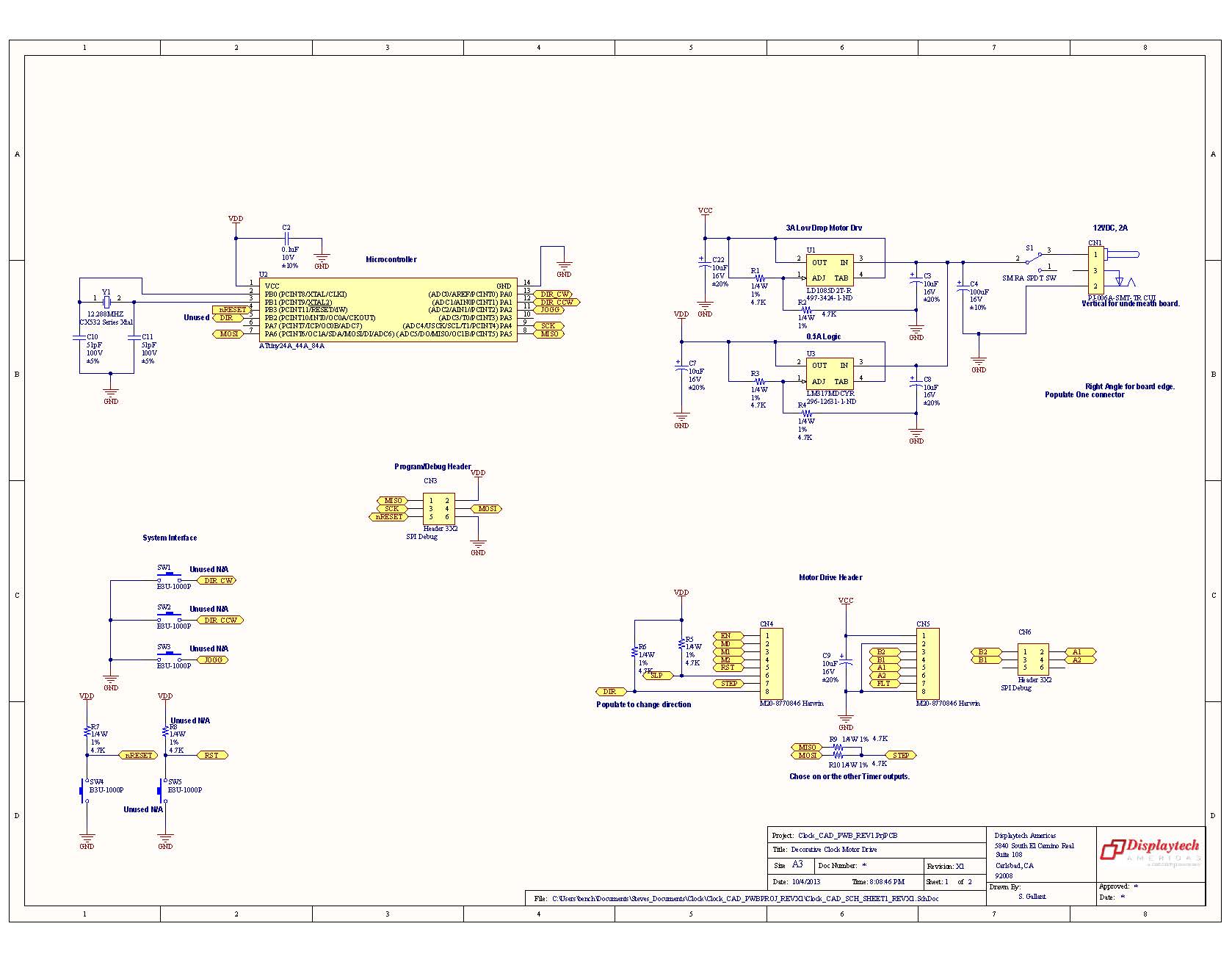


Figure 2. Motor Drive Electronics Schematic.

**Appendix**

Matlab

% File to Calculate Frequencies Near 3 1/3 Hz

% Available Crystal Frequencies between 1MHz and 20MHz. (... is a line continuation)

crystal3 = [1.0000 1.8432 2.0000 2.0480 2.4576 2.5000 2.9491 ...

3.0000 3.0720 3.2768 3.5795 3.6400 3.6864 3.9322 ...

4.0000 4.0320 4.0960 4.0963 4.1943 4.4336 4.4336 ...

4.5000 4.7547 4.8000 4.8970 4.9150 4.9152 5.0000 ...

5.0680 5.0688 5.1850 5.2234 5.5000 5.9904 6.0000 ...

6.1440 6.4983 6.5000 6.5536 6.6128 6.7458 7.1505 ...

7.2000 7.3720 7.3728 7.5000 7.6800 8.0000 8.0002 ...

8.1920 8.3880 8.4320 8.5000 8.9120 9.0000 9.2160 ...

9.5450 9.6000 9.8156 9.8300 9.8304 9.8438 10.0000 ...

10.2400 10.2450 10.7386 11.0000 11.0460 11.0592 11.2280 ...

11.2896 11.5200 11.9814 12.0000 12.0004 12.0960 12.1875 ...

12.2880 12.3520 12.5000 12.6880 12.8000 12.9600 13.0000 ...

13.0252 13.0625 13.2250 13.2256 13.2256 13.2623 13.4916 ...

13.5000 13.5168 13.5213 13.5486 13.5600 13.8240 14.0000 ...

14.0892 14.2764 14.3180 14.3182 14.3359 14.4000 14.4637 ...

14.6900 14.7456 14.8225 14.8500 15.0000 15.3600 16.0000 ...

16.0003 16.2570 16.3676 16.3680 16.3840 16.6660 16.6700 ...

16.8000 16.9344 17.7345 17.7456 18.0000 18.0800 18.0896 ...

18.4320 18.8690 18.9375 19.0625 19.0699 19.2000 19.2800 ...

19.4400 19.6600 19.6608 19.6800 19.6875 19.8000 20.0000];

% Sysem Clock Scale.

sys\_scale = [1 2 4 8 16 32 64 128 256];

% Timer/Counter Scale.

N = [1 8 64 256 1024];

% Output Compare Register Setting (OCRA)

OCRA = 0:(2^16)-1;

% Change display format

format long

% Output Frequency focna = (fosc/sys\_scale)/(2N(1 + OCRA))

fosc = (1e6)\*crystal3;

data = [0 0 0 0 0];

x\_fosc = 1;

x\_sys\_scale = 1;

x\_N = 1;

x\_OCRA = 1;

for x\_fosc = 1:length(fosc)

crystal3(x\_fosc)

for x\_sys\_scale = 1:length(sys\_scale)

for x\_N = 1:length(N)

for x\_OCRA = 1:length(OCRA)

focna = (fosc(x\_fosc)/sys\_scale(x\_sys\_scale))./(2\*N(x\_N)\*(1+OCRA(x\_OCRA)));

if (focna < 3.3333333333333338)

if (focna > 3.3333333333333328)

data = [data; [focna crystal3(x\_fosc) sys\_scale(x\_sys\_scale) N(x\_N) OCRA(x\_OCRA)]];

end

end

end

end

end

end

% Write a comma delimited file for Excel view

dlmwrite('Freq\_Data\_Results\_csv', data, ',')

Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Frequency | Crystal | System | Counter/Timer | Counter Top |
| Hz | MHz | Clock Scale | Scale N | OCRA |
| 3.3333 | 12.288 | 4 | 64 | 7199 |
| 3.3333 | 12.288 | 4 | 256 | 1799 |
| 3.3333 | 12.288 | 4 | 1024 | 449 |
| 3.3333 | 12.288 | 8 | 8 | 28799 |
| 3.3333 | 12.288 | 8 | 64 | 3599 |

System Clock Measurement and Output Waveform Measurement

|  |  |
| --- | --- |
|  |  |

C-Code

/\*

\* DESN204\_Clock\_Code\_00.c

\*

\* Created: 10/13/2013 9:14:56 PM

\* Author: stephen@displaytech

\*/

#include <avr/io.h>

// 3 and 1/3 Hz Square Wave Clock Driver for DESN204 3D Printed Clock Project.

/\* Clock Source is set for Crystal Oscillator. See Section 6.2 Table 6-1. Crystal Osc Operating Mode set to 8MHz and higher. See Section 6.2.5 and Table 6-9.

Page 29. The Start-Up Time is set to 16K CK + 64ms Crtstal Osc slowly rising power. See 6.2.5 Table 6-10. Oscillator is 12.288MHz with two 22pF capacitors as in Fig 6.3.

The Crystal prescale value is /8 as in Table 6-11. (All these values are set using the Tools/Device Programming Fuse Setting GUI of the IDE.) \*/

int main (void)

{

DDRA |= (1 << DDA6); // Set as Clock Output to drive Stepper Motor Driver on PA6, OC1A, PIN7 as output. See PortA, DDRA on P66 Section 10.3.2 and 10.3.3 .

TCCR1B |= (1 << WGM12); // Configure timer 1 for CTC mode (Clear Timer on Compare). See 12.11.2 and Table 12-5 P108.

TCCR1A |= (1 << COM1A0); // Enable timer 1 Compare Output Channel A in toggle mode. See 12.11.1 Table 12-2.

OCR1A = 28799; // Set CTC compare value to (3 and 1/3)Hz at Fcrystal = 12.288MHz, clock prescale of /8, timer prescale /8 and OCR1A = 28799.

/\* Timer 1 output frequency is (Fcrystal/Clk\_Pre)/(2\*Tim\_Pre\*(1+OCR1A)) . (12.288MHz/8)/(2\*8\*(1+28799)) = (3 and 1/3)Hz \*/

TCCR1B |= (1 << CS11); // Start timer at Fclk\_I/O/8 where Fclk\_I/O = Fcrystal/Clk\_Pre. (See 12.11.2 and Table 12.6)

for (;;)

{

}

}

**References**

1. <https://www.google.com/search?rlz=1C1LENP_enUS492US492&es_sm=93&biw=884&bih=445&noj=1&tbm=isch&sa=1&q=gear+clock+hands+image&oq=gear+clock+hands+image&gs_l=img.3...129241.130799.1.131678.6.6.0.0.0.0.322.995.3j1j0j2.6.0....0...1c.1.29.img..11.0.0.QTFNEi6tVTs#facrc=_&imgrc=SXi3euPfKVM1jM%3A%3Bft4CM_EoPSux7M%3Bhttp%253A%252F%252Fwww.wired.com%252Fimages_blogs%252Fgadgetlab%252Fclocks-mirrors-25012007181230.jpg%3Bhttp%253A%252F%252Fwww.wired.com%252Fgadgetlab%252F2007%252F05%252Fwall_gear_clock%252F%3B400%3B404>
2. <https://grabcad.com/library/flower--23>
3. <http://www.rushgears.com/Tech_Tools/PartSearch8/partSearch.php?cadAdHeader&gearType=SPUR&fromSpurGearBody>
4. <http://www.mcmaster.com/>
5. <http://www.pololu.com/catalog/product/2131>
6. <http://www.ti.com/lit/ds/symlink/drv8824.pdf>
7. <http://www.atmel.com/Images/doc8183.pdf>
8. <http://www.digikey.com/product-detail/en/EMSA120300K-P5P-SZ/T1116-P5P-ND/2352109>
9. <http://www.pololu.com/catalog/product/1207>