Lunar Clock.

P.J.Leonard   pauljohnleonard@gmail.com

Background.

The side of the moon facing the sun appears as a bright section whilst the other side is dark. As the moon orbits around the earth the amount of the bright hemisphere varies giving us the phases of the moon. New moon is when the moon is in the same direction as the sun whilst a full moon occurs when the moon and sun are opposite sides of the earth.

A complete phase is about 29 days (Synodic month). But because the earth's orbit around the sun and the moon around the earth are not circular the actual period varies from about 29.18 to about 29.93 days.

The tilt of the moon also varies during the day because over own vertical axis is changing relative to ecliptic plane as the earth rotates. The variation will depends on the latitude. At the north pole there is no variation. Frome is about 51 deg north or 39 deg from north the moon tilts varies by about 2 x 39 degrees during the day. However the earth's axis is also tilted by about 23 degrees (which gives us the seasons). So in addition to the daily variation there is a seasonal variation of tilt.

Phase Calculations

We could have assumed that synodic period was constant but this would mean that the clock would have a noticeable discrepancy compared to the real moon. The celestial calculations to determine the phase of the moon are quite complex and require more accuracy than possible with the limited power of the Arduino microprocessor. So the times of full and new moons were calculated and stored in a table. This table stores the times for the next 100 years in the arduino memory. The times stored in the CPU are accurate to a few hours which is probably a far better resolution than can be achieved by the mechanical system.

Tilt

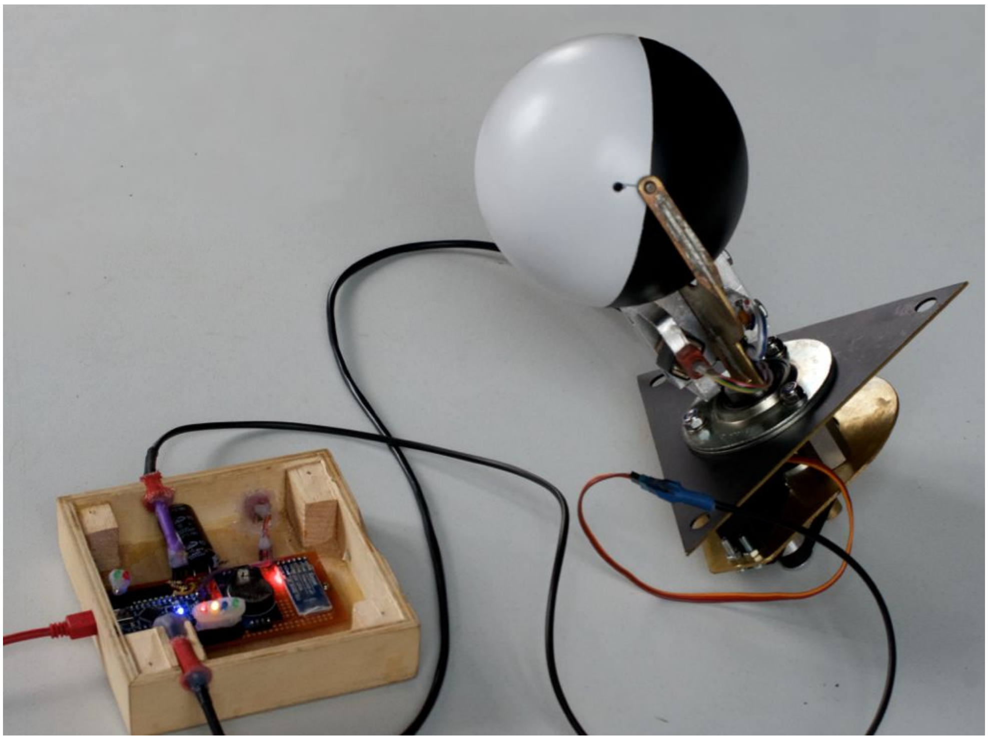
The tilt of the moon can be calculated using vector algebra if you know:

- The time of year after the winter solstice (angle of the earth's tilt).
- The time after midnight (tilt due to our position down from the north pole).
- The phase of the moon (position of the moon relative to the earth).

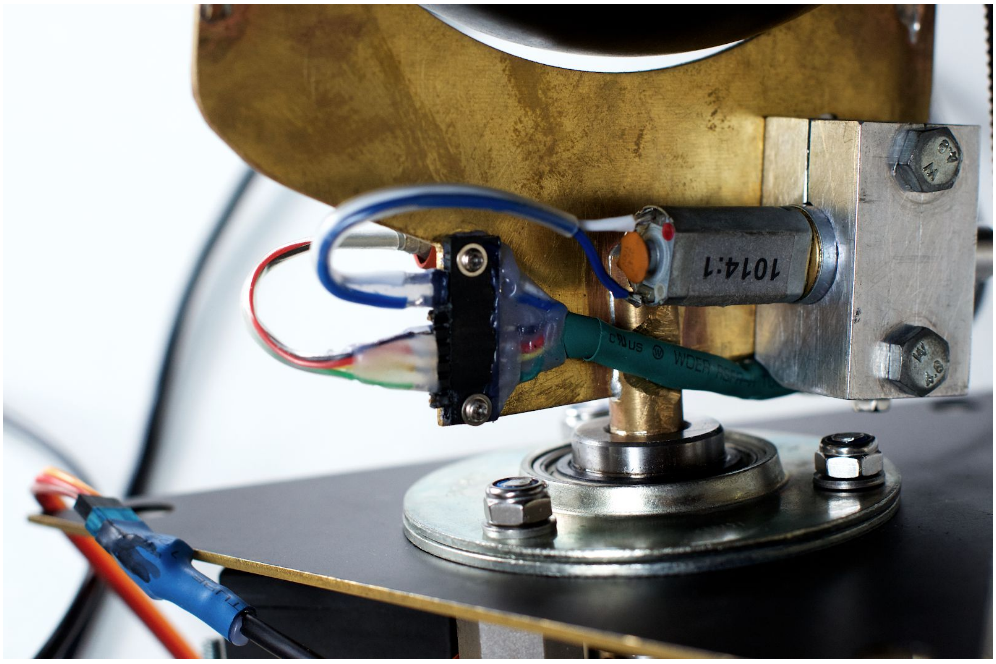
The code to do this can run easily on the Arduino.

The phase and tilt calculations are implemented the moon.cpp module of the arduino code.

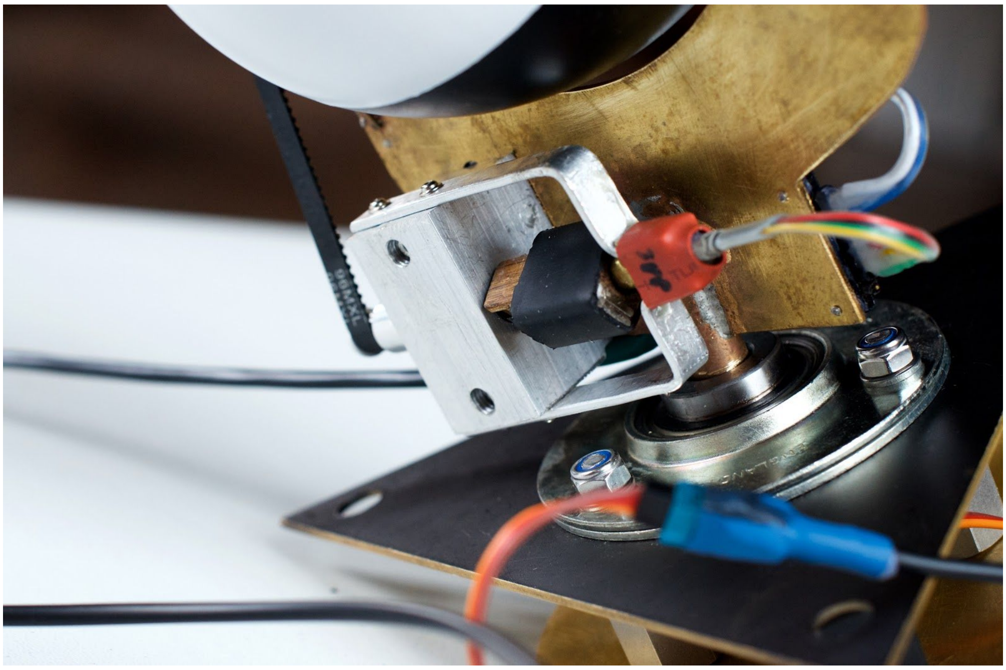
Hardware.



The lunar sphere is mounted on a cut out brass plate.



The drive mechanism comprises a dc motor with a 1:1014 ratio gear box.

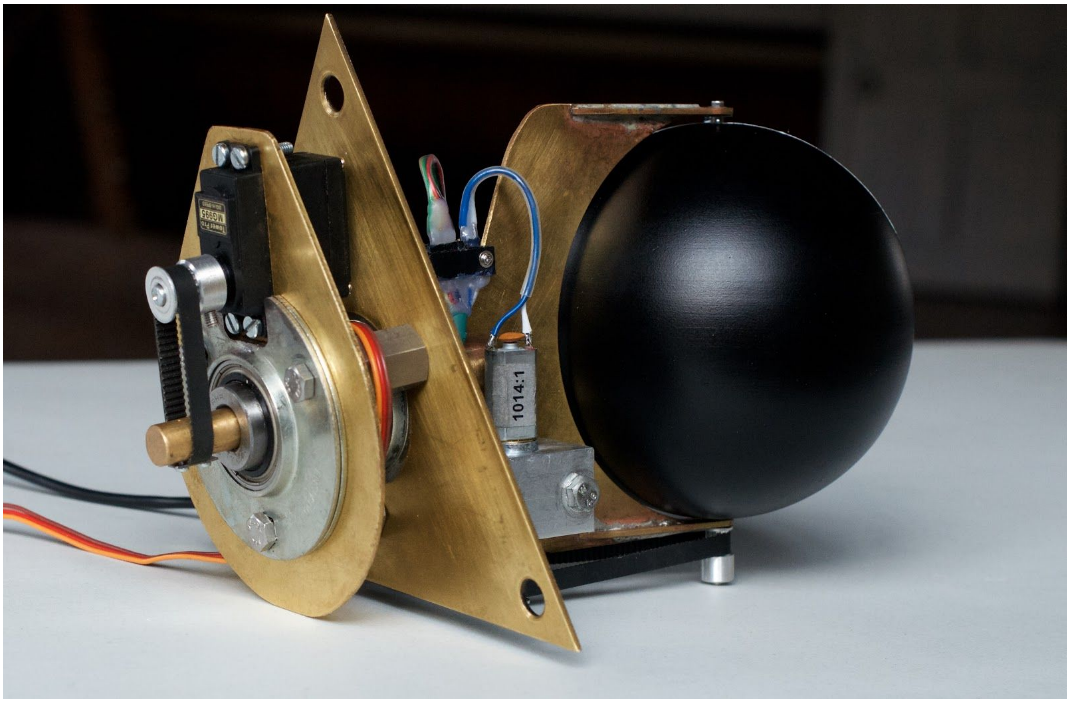


The angle is sensed by 2 hall probes mounted between 2 magnetics.

The cog for the sensor has the same tooth count as the lunar sphere so they rotate at the same rate.

The phase system is mounted on a brass rod which allows the tilt to be varied.

Tilt

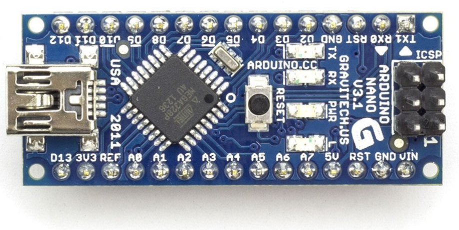


A standard servo is used to control the tilt.

Electronics.

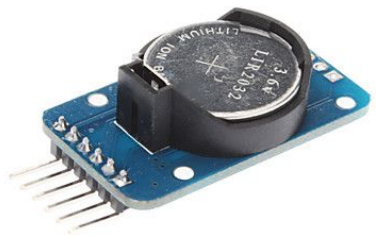
CPU

The system is controlled by an Arduino nano module.



Clock

A high precision clock module provides the time this has it’s own battery so the time is preserved in event of a power loss.



Servo

The servo used was found to draw more current than could be supplied by the power supply. To get around this the supply for the servo was decoupled using a resistor and a capacitor. Under normal operation the capacitor provides enough power to drive the servo each small step as the tilt changes. For large changes the servo voltage drops and the system changes the tilt in several steps until the the target position is reached and no volt drop is detected.

If the voltage is still dropping after 100 steps the system assumes there is a fault such as a jam and the system is halted. (see faults section).

Phase system

The phase system drives the DC motor until the phase sensor is at the desired angle. A L239D half bridge driver is used to drive the motor as the arduino itself is unable to provide enough current. If the phase sensor fails to reach the desired position after about 100 seconds we assume that there is a mechanical fault and the system is halted. (see faults section).

Communication

A bluetooth module allows remote monitoring and control of the system (mainly used for debugging and testing).

Control is via a bluetooth serial link. You will probably need a friendly geek to help set this up.

Device: lunar (or something like this).
PASSKEY: 1234

Use a serial comms program like putty/screen minicom to communicate with the device.

For windows use putty.

http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Hitting CR should display something like

```
>
30/08/2016 14:03:31 JD=2457631.085  Vcc = 4855 mV
```

Note that the red light behind that plastic melt pulses fast if there is no connection and slowly (about once every second once the bluetooth is connected).

Setup/calibrate.

DISENGAGE THE TILT SYSTEM e.g loosen servo grub screw before switching on if the tilt system has not been calibrated.

Halt the system by entering H<CR>. This should halt the system and return the tilt to 0 degrees.

```
> H
*!*  SYSTEM IS NOT RUNNING           *!*
>
 Setting SERVO @ 0.0(deg)  1500(us)  .. please wait ..  SERVO DETACHING  (OK)   0   64831
>
```

Tilt

The tilt should be set to zero by adjusting mechanically using the grub screw on the servo cog wheel.

Phase

Setting mechanically.

The phase can be set mechanically using the grub screw on the moon shaft to adjust the sphere to its' correct position. But you might want to wait till new half full moon before doing this!

Setting Electrically

Alternatively the bluetooth interface allows the phase to be set by adjusting an offset value and saving this to the EEPROM.

Halt the system H<CR>.

Set the phase to zero by entering P 0 <CR>

You could now set the phase mechanically OR

It can be adjusted with the software by setting the phase offset O degrees<CR>

Once it is set correctly enter W<CR> to save this offset in the EEPROM (not needed if doing it mechanically).

Setting the Date/Time

Hopefully this should not be needed as the the clock claims to be accurate to 3.5ppm which means even after 100 years we are only going to be out by a few minutes. However it is possible that a power cut might last too long for the rechargeable battery (or it might might have to be replaced).

Use the command

```
D YEAR MONTH DAY HOUR MINUTE SECOND<CR>

E.g.
> D 2016 8 26 16 50 30<CR>
```

All calculations use GMT so subtract 1 from the HOUR in BST.

Clock battery.

The clock has a rechargeable battery Li-ion Coin Cell Lir2032. This should maintain the clock state whilst the device is not powered. If the clock ceases to hold it's date during a power cycle this can be replaced.

Tilt limits.

If the servo is replaced make sure it has enough range to drive the tilt (+/- 50 deg).
The tilt system is control signal limited to a range of -50 to 50 degrees. However the actual tilt depends on the timing values set for these limits. The servo works uses a pulse of variable width about 1000-2000uS.   The limits of the servo can be adjusted using t MIN_VAL MAX_VAL

e.g
>t 1000 2000

These will be the values can be used to adjust the maximum and minimum angles of tilt. To adjust the tilt halt the system (H<CR>).   Set the tilt limits using t and then test using T ANGLE<CR> to set the servo to the desired angle.   Care should be taken because it is possible to drive the tilt mechanism into the stops if the tilt limits are too small/big.

Demo Mode.

The system can be made to run faster than real time using S MINS_PER_SEC<CR>.
Resume normal speed by entering R<CR>

Error messages.

The set of 4 LEDS indicate the state of the system.

If the the system is running correctly the GREEN led (set of 4) will blink.

The BLUE led lights up when the tilt servo is active.

If the RED LED (in set of 4) is constantly on this means the system has detected a fault.
Hitting <CR> should display a message

If the phase system fails to set the phase then:

```
*!*  SYSTEM IS NOT RUNNING           *!*
*!*  PHASE SYSTEM BROKEN FLAG IS SET *!*
```

If the tilt servo keeps drawing current :

```
*!*  SYSTEM IS NOT RUNNING          *!*
*!*  TILT SYSTEM BROKEN FLAG IS SET *!*
```

In either case investigate any mechanical problem.
The error flags can be cleared using

```
!<CR>
```

The red light should turn off leaving just the YELLOW led which indicates that the system is not running.

Restart the system using R<CR>

Faults

If the system is designed to detect jams or mechanical faults e.g. if the phase angle takes too long to set or the servo voltage stays low for too long. In this case the system may halt and the red warning light will stay on. In this case the cause of the failure should be investigated and once rectified the system turn on. If the fault has caused the belts to slip you should recalibrate the system as described in Setup/calibrate

Command summary.

Debug/display	
M n	Display the next n full/new moons dates.
L [N incHour year month day hour minute sec]	Display N states of moon starting at the given date/time in increments of incHour. By default N=1 and incHour=1 the start time is the current time.
?	DIsplay help.
S MINS_PER_SEC	Speed mode demo.
P DEGREES	Manually set the phase (system must be halted).
T DEGREES	Manually set the tilt. (System must be halted).
Setup/Calibrate	
C	Calibrate phase sensor by turning sphere and noting min/max values. Write to EEPROM to save values.
O degrees	Set phase offset
t MIN_US MAX_US	Tilt servo limits
p val	-1 or 1 change parity of tilt

H	Halt system
W	Write to EEPROM (use after O C t p) TO persist your changes
Misc	
!	Clear errors
R	Resume running

Note that any input during the phase control or commands that list stuff will stop halt the system.

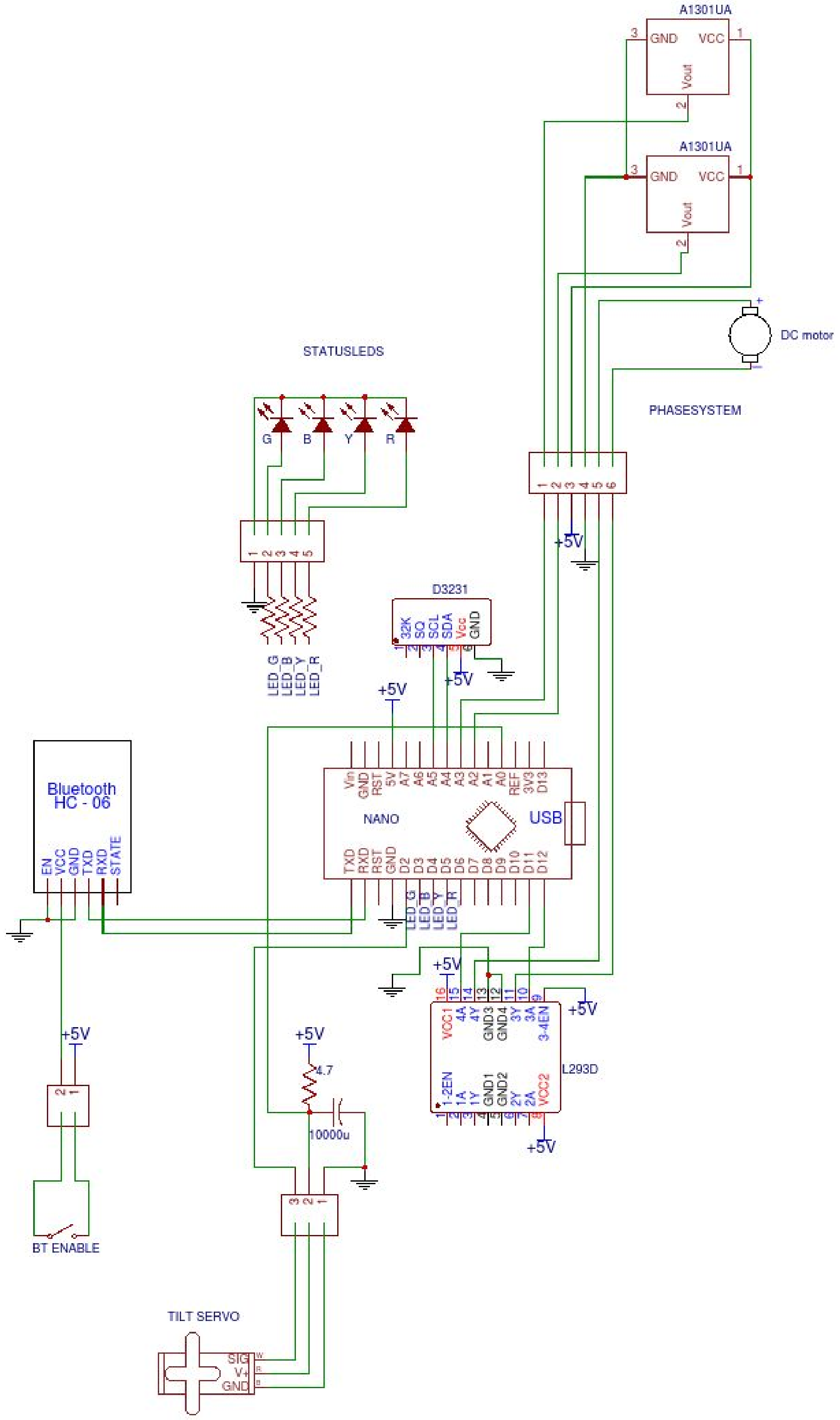
Source Code

The source code for the arduino can be found here.

`https://github.com/pauljohnleonard/moonclock`

This repository also has the code used to generate full/new moon table used by the arduino code.

PCB circuit diagram



Bill Of Materials.

Mechanical Stuff from Motioncol:
Aluminium MXL Pulley, 10T, 3mm Bore PLMXL010AL03
Aluminium MXL Pulley, 12T, 4mm Bore PLMXL012AL04 X2
Planetary Gearmotor, 1014:1 ratio GMP16D1014
Plain Polymer Flange Bearing, 4mm Bore, 5.5mm O/D BG04M
Collar, Plated Steel,4mm I/D   x4 CL04M
303 Stainless Steel Shaft, 4mm Diameter SH04M X2
MXL Rubber Timing Belt, 63 T BMXL063
MXL Rubber Timing Belt, 120 TBMXL120
Aluminium MXL Pulley, 20T, 6mm Bore PLMXL020AL06
Plain Shaft Servo with 180 degrees of rotation SRV180MG2BB

Phase Sensors:
Brass tube for sensor
ALLEGRO MICROSYSTEMS Hall Effect Sensor Linear 3Sip A1302KUA-T x2
20 x 10 x 2mm thick N42 Neodymium Magnet X4

Electronic components
Cable Hornby / Peco Multicore Cable 7/0.22 6 cores
Cable Hornby / Peco Multicore Cable 7/0.22 4 cores
Arduino Nano Atmega328P
Capacitor   10000uF   16V
DS3231 Atmel AT24C32 Module Precision Real Time Clock Module
Vero Board
MISC Headers/pins/shrink wrap
L239D driver chip
HC-05 bluetooth

Power supply   USB   2Amps.
USB cable

Acknowledgements.

Evgeny Fedorischenko: Daff Moon
https://play.google.com/store/apps/details?id=com.dafftin.android.moon_phase&hl=en_GB

Arnold Barmettler: CalSky
https://www.calsky.com/cs.cgi/Moon

Bennett Centre:

http://bennettcentre.org.uk/