instructables

technology workshop craft home food play outside costumes

Ciclop 3D scanner (BQ & Horus)

by **dtrewren** on August 1, 2016

Table of Contents

Author:dtrewren

I am an electronics engineer working in telecommunications in the UK. Generally I work on chip design for digital radio receivers. I like all things arty and consider chip design a bit arty :) Basically I have a lovely workshop and spend large amounts of time designing and building stuff but also including my little girl Ava as much as possible - trying to teach her to be interested, to question and to think.

Intro: Ciclop 3D scanner (BQ & Horus)

Good day everyone,

Little bit of an introduction and background:

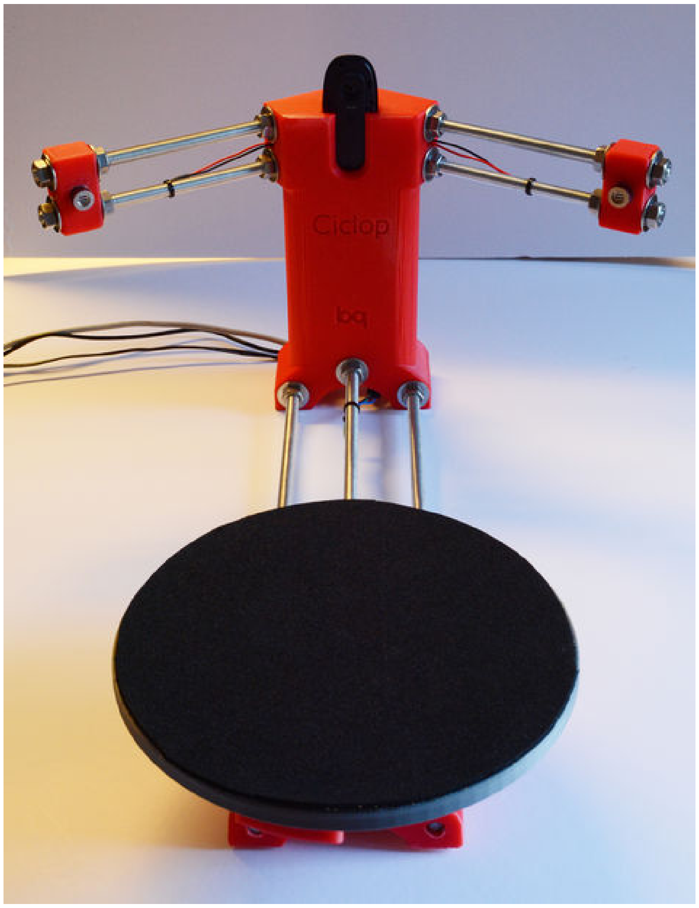
My name is Dave and I am from Bristol in the UK. Late last year I took the plunge and after much research and question asking I bought myself a 3D printer. The printer I opted for was the Lulzbot TAZ5. This printer is a product of Aleph Objects Inc which are based in Colorado USA - I point out now I have no connection with the company at all other than buying one of their printers. I have found the printer to be excellent, very reliable and a joy to use. As important as the printer the customer support from Lulzbot has been amazing, no in fact exceptional !

For anyone interested here is a little video I made going through the printer unboxing and set up:

I have been using the printer to print all sorts of things from artistic sculptures to brackets for bikes. A while ago I came across the open source Ciclop 3D scanner project and thought I had to build the design.

The Ciclop 3D scanner is an open source project that anyone who wants to invest some time can build. So all the designs for the component parts, the hardware, the software and the firmware are available on-line. Therefore, this instructable is not my design but instead a description of how I went about building the Ciclop scanner. I have made some small modifications here and there to the printable parts to better suit my particular 3D printer. These modifications are specifically to the support structures required to print overhangs.

Most of this instructable consists of lots of photographs, screen shots and some video showing the progress I made building the Ciclop and hopefully will help anyone else who may want to have a crack at building one.



Step 1: Ciclop 3D scanner - the basics

The Ciclop 3D scanner allows any object of a suitable size to be scanned in thee dimensions ultimately resulting in a model that after post processing can be edited, manipulated and reproduced using a 3D printer.

The object of choice is placed on a rotating platform. As the object is rotated two line lasers illuminate the object from two different angles. A camera then scans the illuminated object at each rotational position. The data from the camera is then used to build a 3D point cloud of the object.

The Ciclop scanner is constructed from the following basic building blocks:

- A series of plastic components which can be produced using a domestic 3D printer

- A series of threaded connecting rods which hold the plastic parts in position

- A microprocessor control system & firmware used to locally control the platform motor & lasers

- A good quality camera used to capture the lined scanned data

- A PC based software application used to perform calibration, invoke/control scanning and assemble the result

Over the following instructable steps I will go through the above basic component parts in order to share my experience of the build.

Step 2: The Ciclop plastic component parts (Lulzbot TAZ5 printed using HIPS)

Shown above are most of the Ciclop plastic component parts, all these components were printed on my TAZ5 3D printer using red HIPS 3mm printing filament, an extruder temperature of 240C and a print bed temperature of 110C.

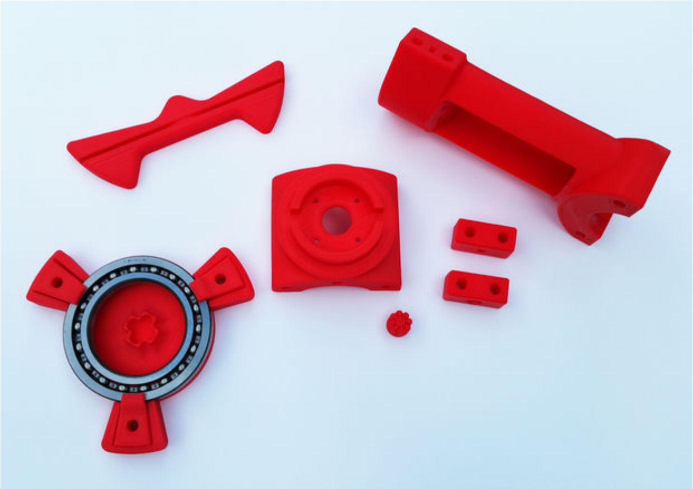
The models for the components were downloaded from thingiverse:

http://www.thingiverse.com/thing:740357

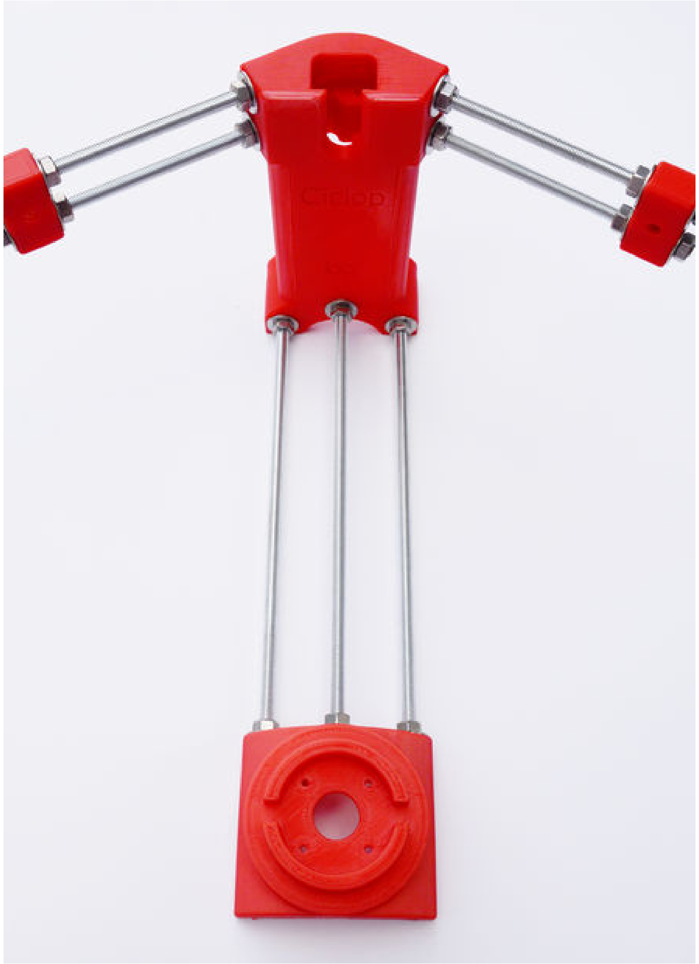
I personally found that the support material used on the camera holder and motor holder did not print particularly well or provide adequate support. I subsequently imported the designs into Meshmixer and removed the support. I then imported the support free models into Simplify3D and added my own custom support structures. Unfortunately, it is not possible to output the models including the support structures from Simplify3D in an STL or OBJ format. The models including support can be exported from Simplify3D but only using the tool's own factory file format. So the upshot is I can only share the models including support with other Simplify3D users or provide the ready to print Gcode. The Gcode option may be ok if you are using a TAZ5 3D printer but may not print as well on other brands of printer.

There are links to other aspects of the project on the thingiverse link. However, I did find more useful links to the required project parts which will be detailed in subsequent steps.

The following steps simply show photographs of the results of my printing and the Ciclop mechanics assembled.

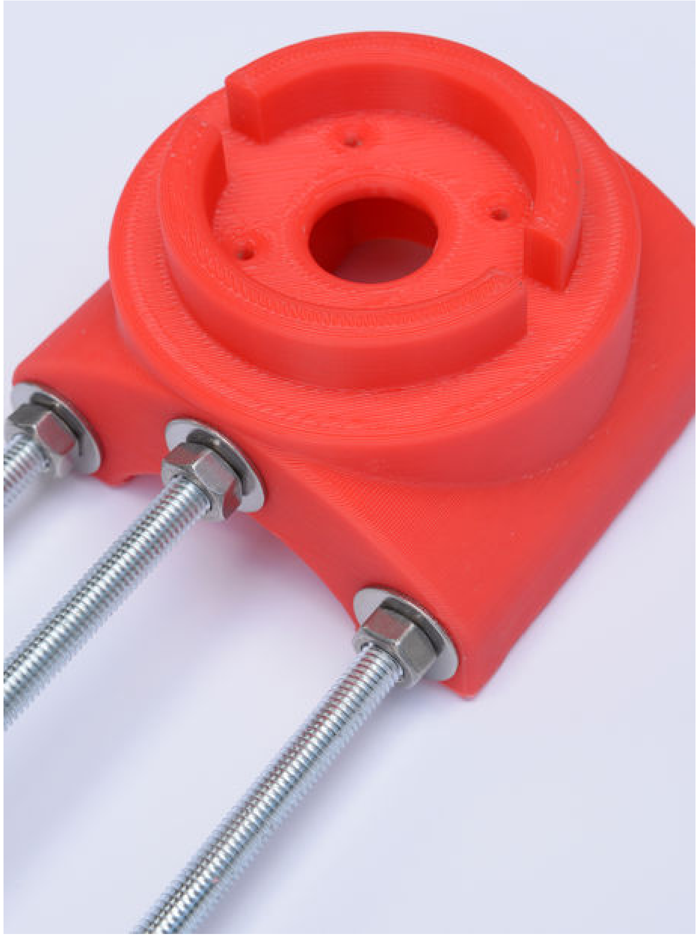


Step 3: Ciclop framework excluding the motor and platform

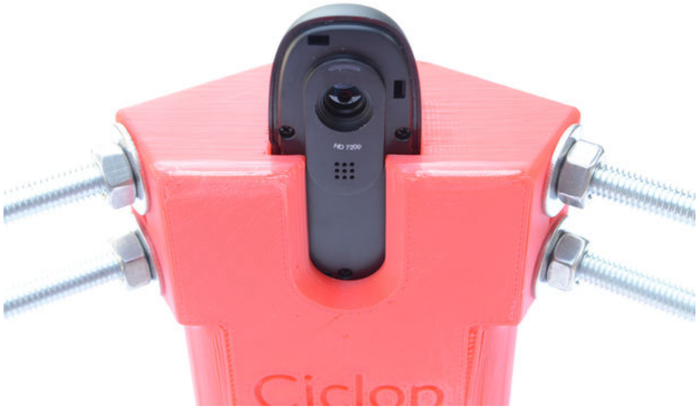


Step 4: Camera holder base

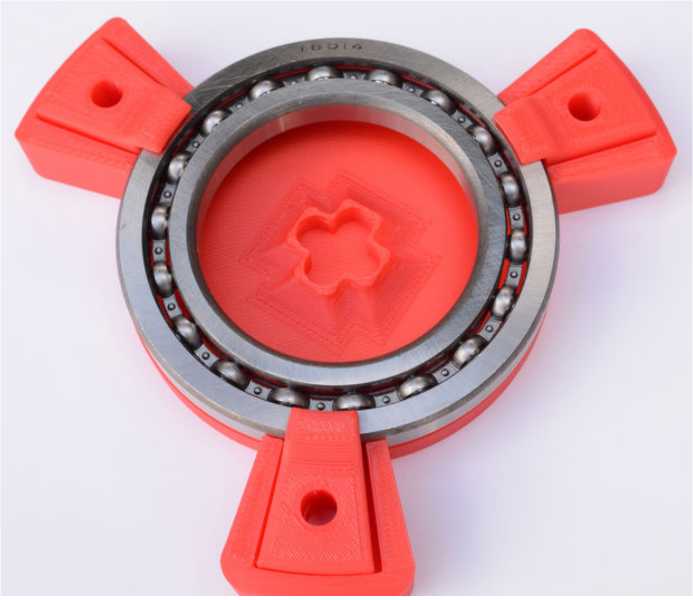
Step 5: Motor holder base



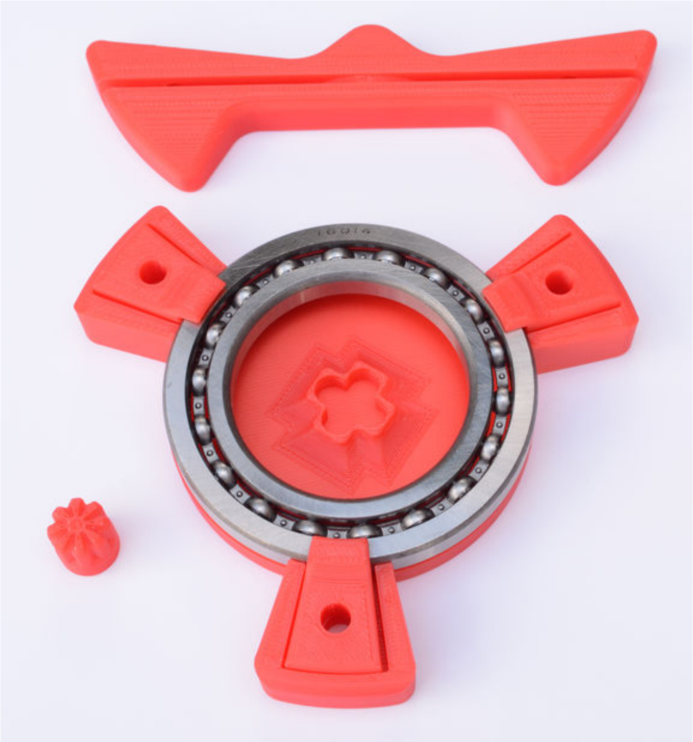
Step 6: Camera holder with camera



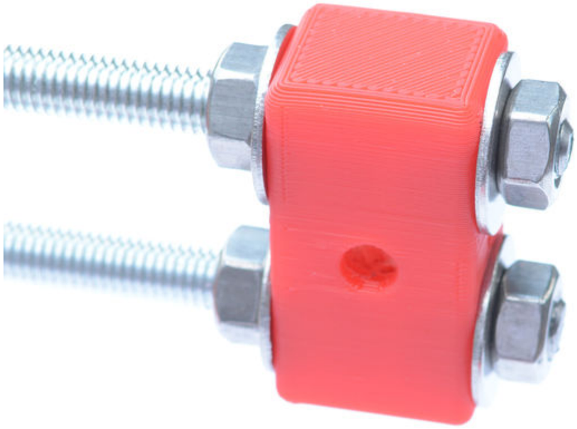
Step 7: Platform holder including the bearing & clips



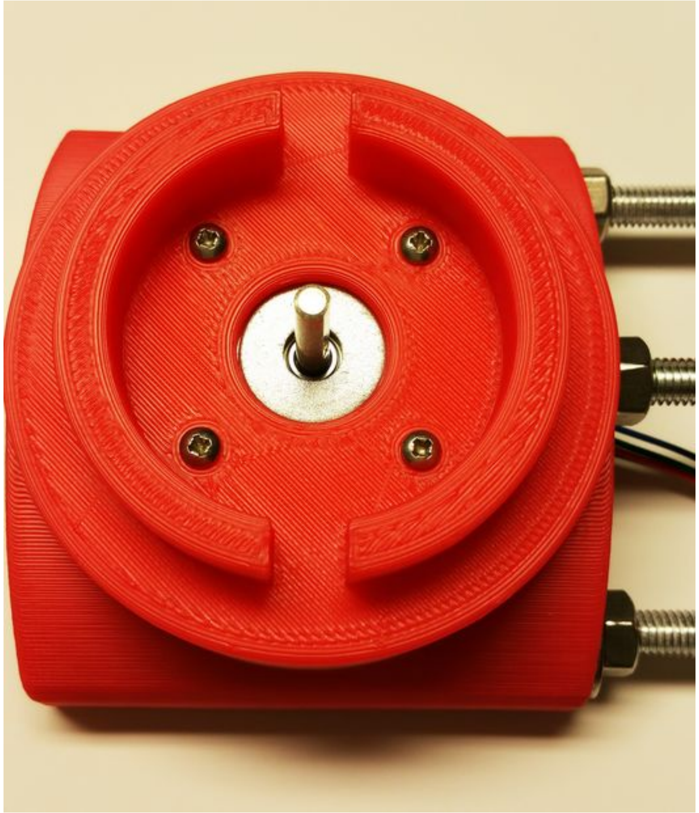
Step 8: Platform holder, bearing & clips, calibration card holder and shaft coupling

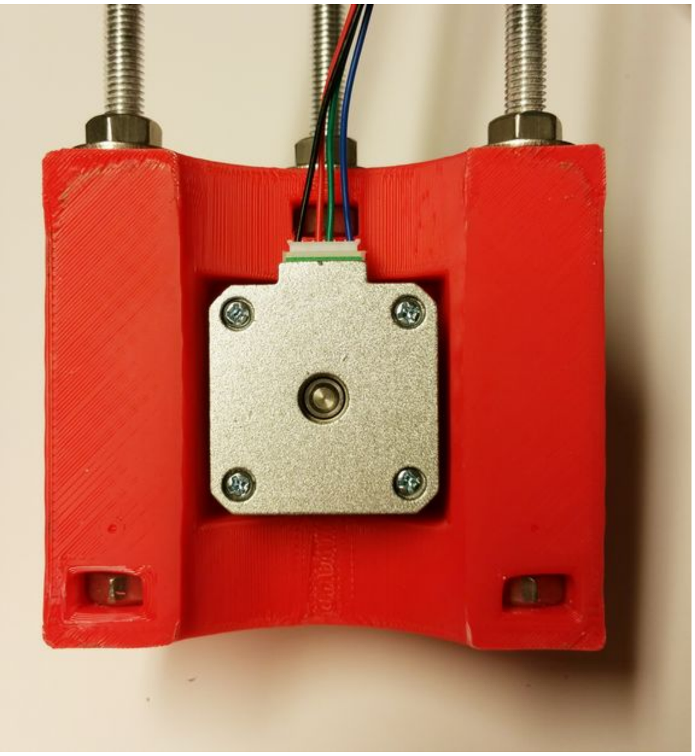


Step 9: Laser holder



Step 10: **Motor mounted into motor holder top/bottom**





Step 11: **Bearing mounted in holder and bolted to the platform**

Step 12: On-board motor & laser control system

The on-board motor and laser control systems is based around an Arduino UNO ATmega328P control board. I don't generally use the Arduino system for my other microprocessor based projects preferring to build bis-spoke based control systems around a PIC microprocessor best suited to the application. However, the Ciclop project is not my design and so I ventured into the world of Arduino.

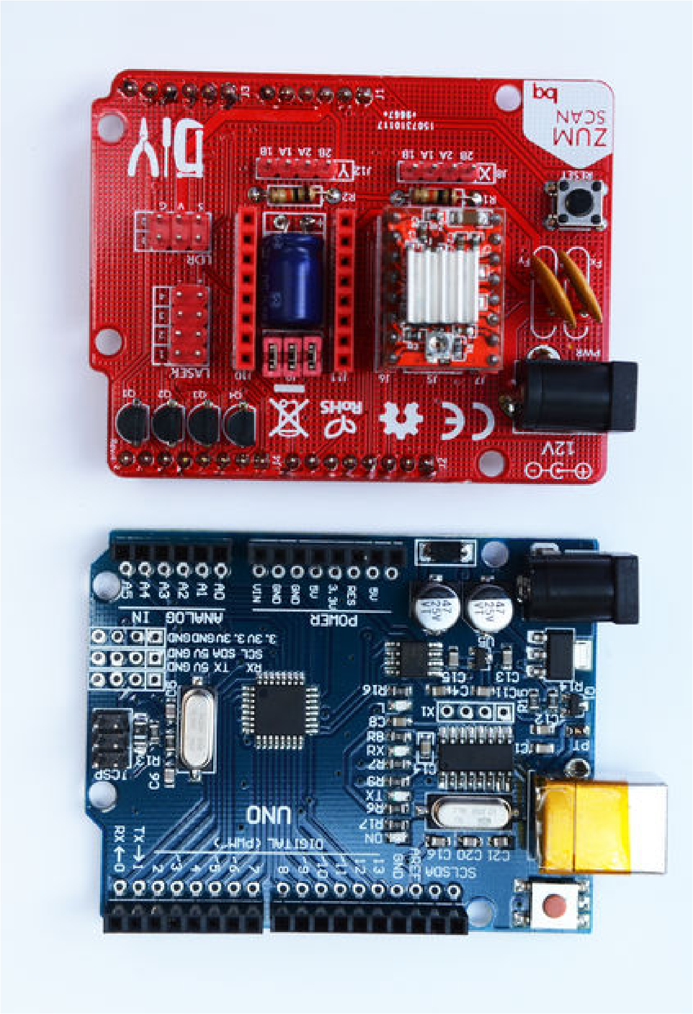
Although I was going to spend money on the Ciclop project I was prepared to cut a few corners and get the price down a bit. To this end I first decided to buy a cheap Arduino clone board from Ebay for the unbelievable price of £3.99 delivered. The Arduino control board is the blue board pictured at the head of this step. The board is based around the ATmega328P microprocessor. The board has a simple linear power supply, a clock source for the micro, a reset button and a few leds. The idea behind the Arduino system is that you take you basic control board and then plug what is termed a shield on top. The shield is an application specific hardware board that acts as an interface between the microprocessor and whatever it is you want to control. This is a simple but very powerful system. The basic control board is generic and mass produced at very low cost. It is then left to the community of developers and anyone else for that matter to go on to produce a staggering range of shields to perform just about any task you care to think of. For example last year I built a quadcopter using a flight control system that is essentially an Arduino platform with an integrated flight control shield. Mounted on my quadcopter I have a two axis stabilized camera gimbal - again an Arduino core with an application specific integrated shield. For the quadcopter it made more sense for the shields to be integrated. However, for the Ciclop the shield is a more traditional piggy back interface board. The Ciclop shield provides the hardware interface control for the bipolar stepper motor and also for the line laser control. In the case of the Ciclop shield the integration is taken one step further and a stand alone bipolar stepper motor driver module is used as these modules are used extensively in the 3D printer and home CNC market and are very inexpensive.

In order to save money I chose to buy my Arduino Ciclop shield as a kit.

The Arduino shield kit I purchased from eBay here :

http://www.ebay.co.uk/itm/252442375077?_trksid=p2060353.m2749.l2649&ssPageName=STRK%3AMEBIDX%3AIT

Support from this supplier was very good and the kit is sourced from the UK.



Step 13: Downloading the firmware to the Arduino platform

In this step we will go over the process of downloading or programming the Arduino platform with the firmware.

Firmware in this context is essentially the code that runs on the Arduino ATmega328P microprocessor. The board is supplied already programmed up with the bootloader so all that is necessary is to use the Arduino Integrated Design Environment or IDE to program the board with the Ciclop firmware.

The process is as follows :

- Install the Arduino IDE after downloading from the internet

https://www.arduino.cc/en/Main/Software

- Download the latest version of the firmware, you need the horus-fw.ino sketch file
https://github.com/bqlabs/horus-fw

- Download & install the CH340G driver if you have a clone version of the Arduino board. I will elaborate slightly on this point. Authentic (non clone) Arduino control boards are fitted exclusively (as far as I am aware) with FTDI USB to serial interface devices and therefore require the appropriate FTDI software driver. However, for various reasons outside the scope of this instructable most of the clone boards are now fitted with a different type of interface device that requires a different driver (CH340G). In my experience once this driver is installed the board appears as a COM port and everything else operates as expected.
http://www.arduined.eu/ch340g-converter-windows-7-driver-download/

- Connect the IDE to the board via the appropriate COM port (look in the Windows device driver list to determine the correct COM port) (see screen shot)

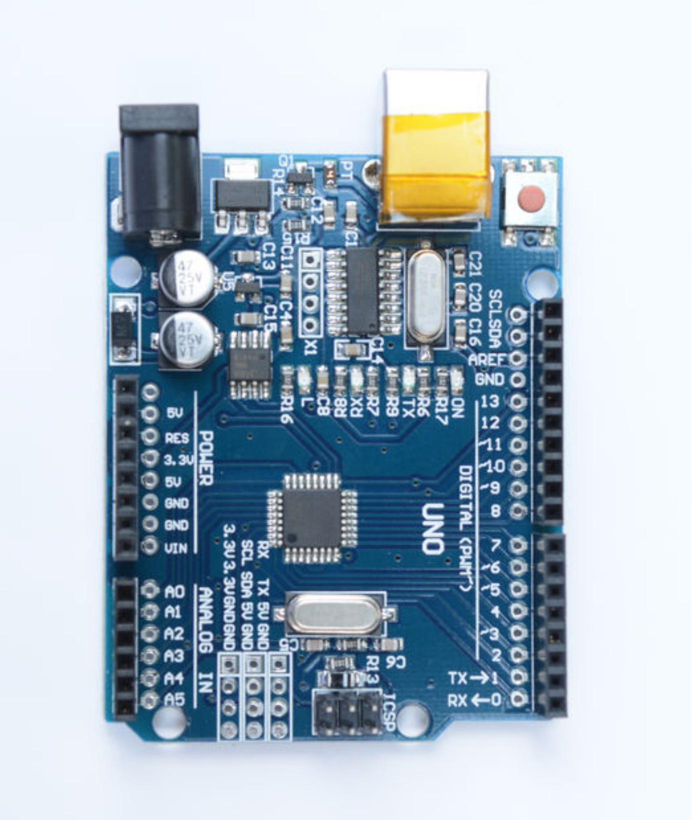
- Select the board type (see screen shot)

- Select the programmer (see screen shot)

- Load the firmware sketch (see screen shot)

- Upload the sketch to the board

One thing that is worth trying if you are new to all this Arduino business is to up load an example sketch first just to give you the confidence that everything is working correctly. Under the IDE file menu there are a bunch of example sketches - one of which is simply called blink. This sketch once uploaded just flashes an LED on the Arduino UNO board. So you don't need the Ciclop shield fitted at this stage just upload blink and sanity check that the board LED is flashing. Once happy you can go onto load up the Ciclop sketch and program the board. The entire programming process for the Ciclop firmware takes just a few seconds.



Step 14: The camera connection and driver

The camera used is a Logitech C270 HD webcam, sourced from eBay. Once you have your camera you need to install the camera driver.

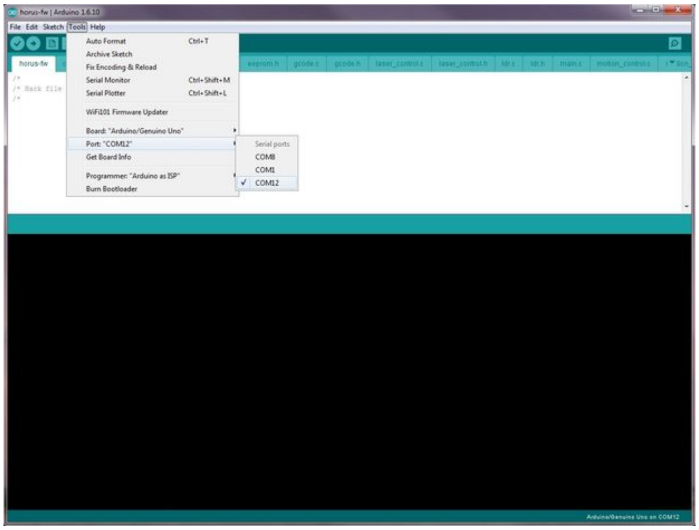
I downloaded the driver from here :

http://support.logitech.com/en_ca/product/hd-webcam-c270

The camera simply slots snuggly into the top of the camera holder. The USB cable runs down through the camera holder and out of the base.

Step 15: Using the Arduino IDE to select the COM port

Note the COM port in MY case was COM12. However, for you it may (probably will be) different. Under the Windows control panel open the device manager and check which COM port has been allocated.



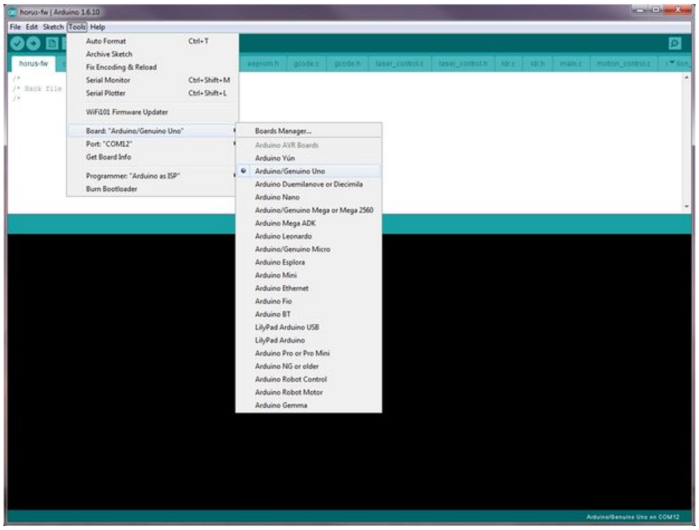
Board: "Arduino/Genuino Uno"

Port: "COM12"

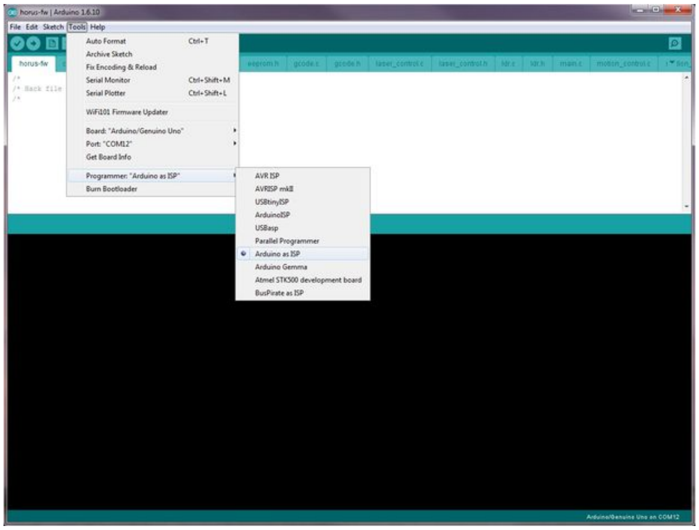
Programmer: "Arduino as ISP"

Arduino/Genuino Uno on COM12

Step 16: Using the Arduino IDE to select the board type

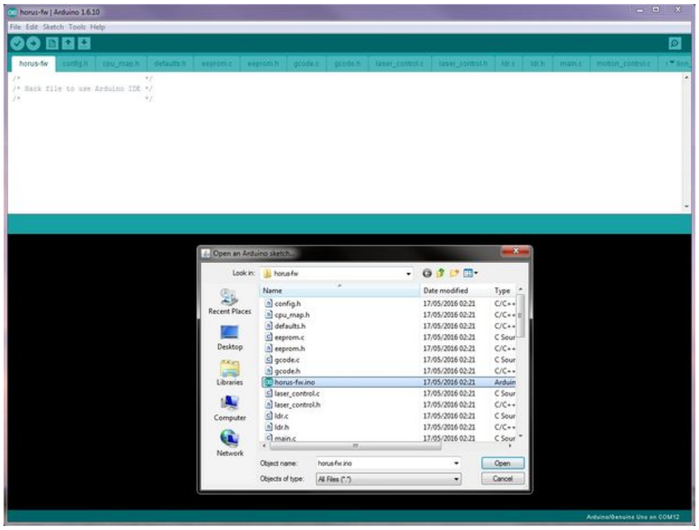


Step 17: Using the Arduino IDE to select the programmer type



Step 18: Using the Arduino IDE to select the Sketch (Ciclop firmware)

Once the firmware has been successfully programmed you can plug the laser/motor control shield onto the Arduino board - the controller is now ready for integration into the Ciclop scanner.



```
/*
/* Back file to use Arduino IDE */
/*
```

Look in: horus-fw

Name	Date modified	Type
config.h	17/05/2016 02:21	C/C++
cpu_map.h	17/05/2016 02:21	C/C++
defaults.h	17/05/2016 02:21	C/C++
eeprom.c	17/05/2016 02:21	C Sour
eeprom.h	17/05/2016 02:21	C/C++
gcode.c	17/05/2016 02:21	C Sour
gcode.h	17/05/2016 02:21	C/C++
horus-fw.ino	17/05/2016 02:21	Arduin
laser_control.c	17/05/2016 02:21	C Sour
laser_control.h	17/05/2016 02:21	C/C++
ldr.c	17/05/2016 02:21	C Sour
ldr.h	17/05/2016 02:21	C/C++
main.c	17/05/2016 02:21	C Sour

Object name: horus-fw.ino

Objects of type: All Files (*.*)

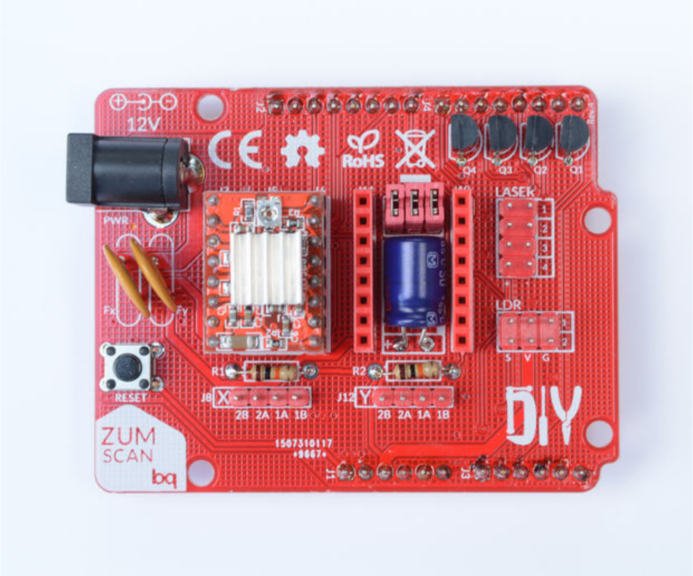
Step 19: Arduino UNO and laser/motor control sheild fitted together

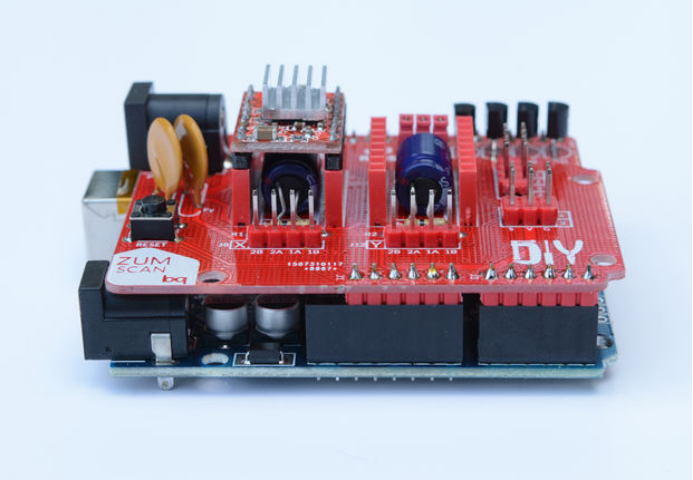
This series of images show the Arduino UNO control board and the laser/motor control board fitted together.

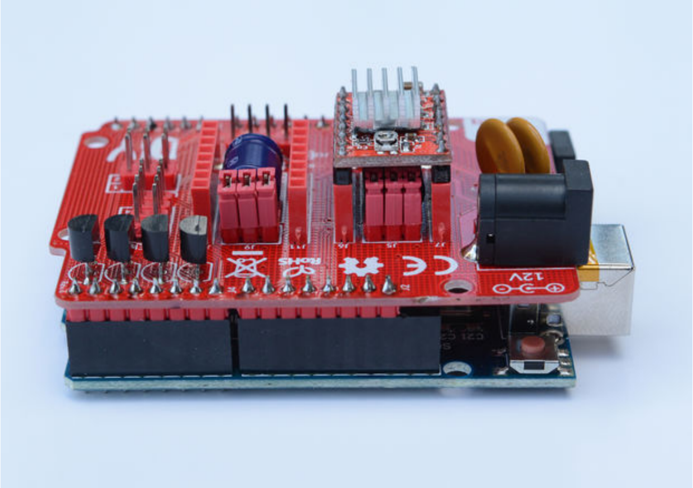
This is a good point to issue a word of WARNING as there is a bit of a design flaw with the board combination.

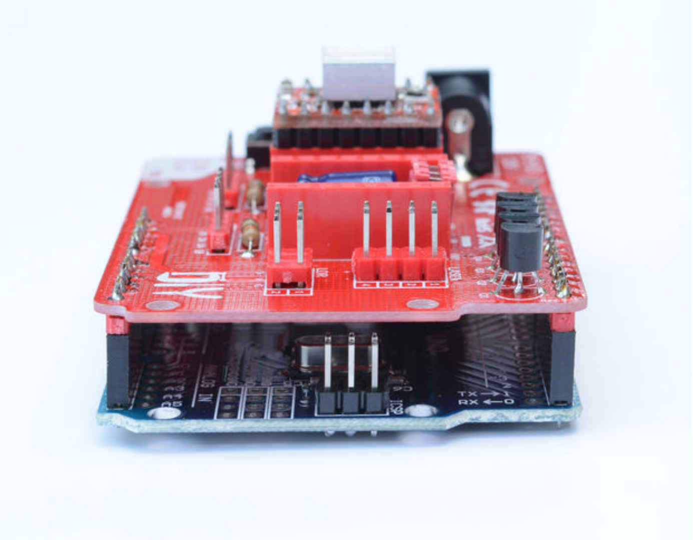
When the boards are plugged together it becomes apparent there is a design flaw. The soldered connections on the base of the power connector foul the top of the USB socket and potentially the pins of the capacitor immediately to the right of the power connector. This is bad news as the USB socket is connected to ground and the power connector is delivering +12V onto the board.

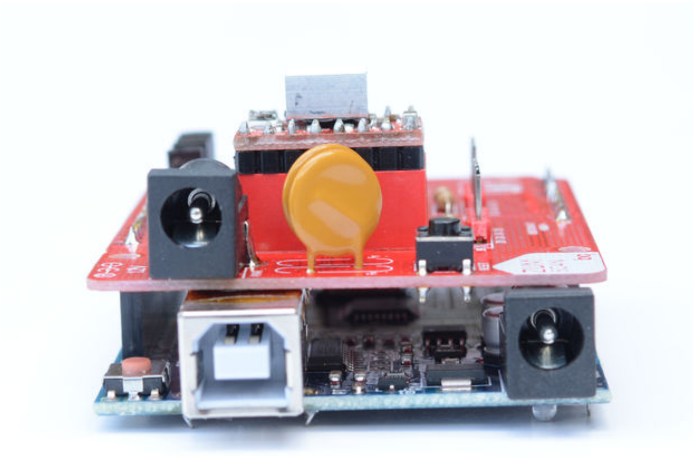
My solution was to file down the solder connections on the base of the power connector and then insulate both the board and the USB connector with a double layer of Kapton tape.











Step 20: The motor connection

I think it is worth just going over the motor connection to the laser/motor control shield. I bought my motor from eBay and it was supplied with a header connection on the side and a plug with about 1m of cable - very nice. However, it may not be obvious how the four connections of the motor connect to the motor controller. I do quite a bit of work with stepper motors and it never ceases to amaze me how much misinformation there is on the internet regarding their connections. There are two main types of stepper motor Unipolar and Bipolar - this particular motor is a Bipolar type with four wires. Essentially inside the motor are two coils with two wires connected to each coil - hence a four wire interface.

The laser/motor control shield has a four pin header labelled 2B 2A 1A 1B

Pins 1A & 1B connect to one of the motor coils and 2A & 2B connect to the second motor coil.

Looking at the attached image for the particular motor I have the connections are as follows :

1A - Black

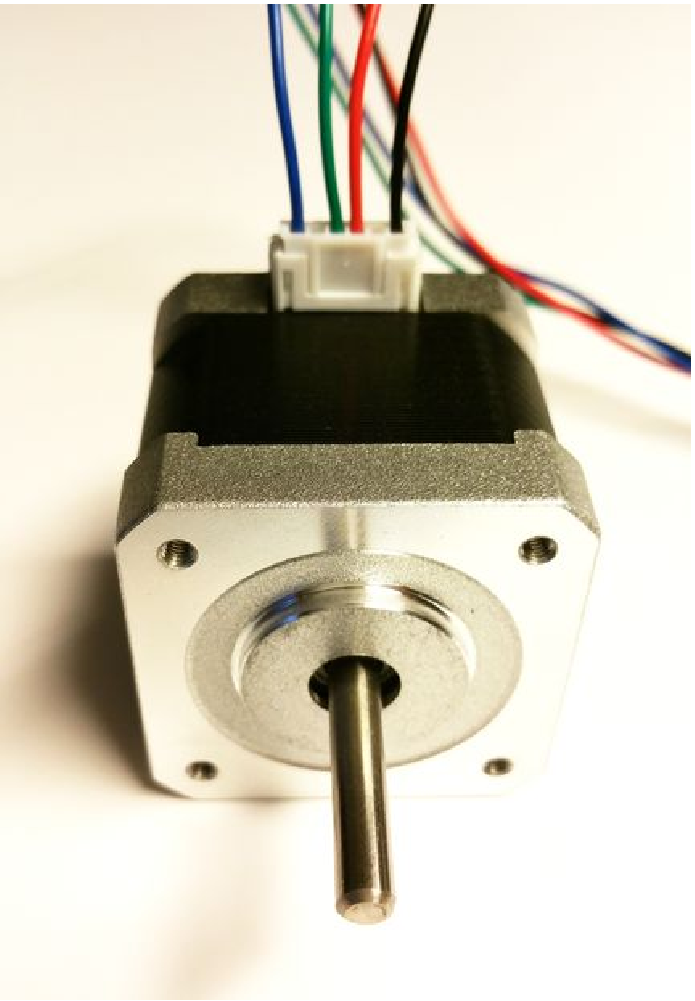
1B - Green

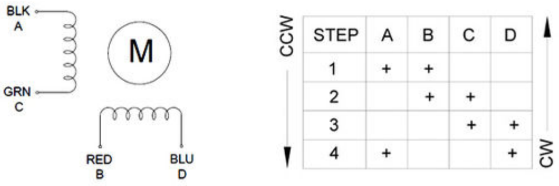
2A - Red

2B - Blue

This connection ensures the motor runs correctly and also rotates in the correct direction.

The motor I used is a Nema 17 type labelled : JK42HS40-1704-13A





STEP	A	B	C	D
1	+	+		
2		+	+	
3			+	+
4	+			+

Step 21: The lasers

The lasers used in the project were 5mW focusable line lasers, again sourced from eBay. Each laser has a rotatable diffraction lens mounted on the front which projects a laser line and by rotating allows the laser line to be focused.

The laser holder plastic parts as printed were obviously intended for a laser with a smaller diameter body and so my lasers didn't fit. I could have imported the laser holder model and modified it and then re-printed ..... it was late by that stage so I just drilled out the plastic to 12mm and popped the lasers inside.

I extended the fly leads to each laser with spare red/black wire that was left over after cutting the motor lead to length. All leads were soldered and then made secure with a heat shrinkable covering.

The two lasers connect to the laser/motor control shield as follows :

Left laser connection port 1

Right laser connection port 2

The red lead connects to the inner most pin closest to the motor controller, the black lead connects to the outer most pin closest to the board edge.

Step 22: Mechanical construction

The mechanical construction of the scanner is reasonably obvious. However, there is an excellent video available showing everything in detail :

Step 23: Horus application software and documentation

The PC/Linux Horus application software can be found here :

http://horus.readthedocs.io/en/release-0.2/

I didn't see much point in explaining the software operation in detail when all the documentation has already been done so well.

In order to get good results I would suggest reading all the documentation rather than just diving in. There is not that much to read and after ten minutes you will be better equipt to get a good result.

Step 24: Finished Ciclop scanner quick video look around

This video is just a quick look over the finished scanner prior to performing calibration.

Step 25: Ciclop scanner calibration workbench

After reading the documentation I decided to use the set up wizard and simply follow the steps. Once the calibration grid has been put into the platform the first thing.

This video shows the calibration sequence if not using the wizard.

Step 26: Ciclop scanner scanning workbench

This video shows the scanning operation.

Step 27: Post processing the results in Meshlab

Once you have successfully scanned an object what you have is called a point cloud. This point cloud is not something you can just load up and 3D print. As the name suggests it is simply a cloud of scanned point and not a printable closed 3D mesh.

I am sure there are lots of ways and tools to turn the point cloud into a 3D closed printable mesh but my tool of choice is Meshlab which is free, open source and truly excellent :

http://meshlab.sourceforge.net/

I would not admit to using or fully understanding every aspect of Meshlab and it can be quite daunting when you first start using it. I use the bits I need and try and understand as much as possible.

So I thought I would include a step at the end of this instructable showing how I use MeshLab to convert the point cloud obtained from the scanner into a solid printable mesh and also transfer colour texture information from the scanned point cloud onto the mesh. Transferring the colour maybe not be that useful if you are going to 3D print the scanned object but you might want to use the scan for some other purpose where colour from the original point cloud scan would be nice to have.

Step 28: Ciclop 3D scanner project - conclusion

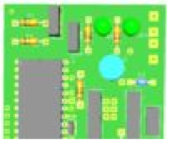
This has been a great deal of fun to build and I think I have learn some new things which is the whole point. It is a good project as it covers many different areas - 3D printing, electronics, mechanics, optics, software, firmware, microprocessor control, 3D image manipulation and post processing.

I am not sure how much I will use the scanner and for what purpose but that is not the point. The point is to buy cool stuff, put it all together in order to learn new things and gain a few more skills.

Please point out any inaccuracies or errors, I will correct. Also please fire away with any comments or input as I would be very keen to hear what people think.

Remember .... build and have fun !

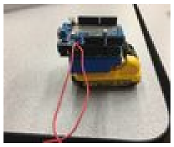
Related Instructables



3D Scanner control board arduino nano + drv8825 by jasonmarkham

3D Printing: From Design to Print! by jocomakerspace

Making Metal Parts With Your 3d Printer by DivideWorks



Arduino Drill Battery Adapter by DoyceC

3D Printed Back to the Future TIme Circuit Clock by gg2inc

Hard Drive NeoPixel 3D Printed Clock by gg2inc

Comments