SafeCracking on a Budget

About Us

Jay – Works for Indigo IT Labs based in NE Victoria. Has experience in automated production lighting

Luke – plays some CTFs, creator of BitcoinCTF, mostly focuses on web security

What this talk will cover

- Our process of prototyping an autodialer
- Lessons we learnt that can help others
- Live Demo (If the demo gods allow)

Live Demo

Instead of just having the autodialer run, we decided to let the audience crack the safe.

Send a tweet with the hashtag “#ruxsafecrack” with your combination.

Live Demo

The safe has 3 numbers from 0 to 99.

Example Tweet:

#ruxsafecrack 29 70 35

Live Demo

The safe has 3 numbers from 0 to 99.

Example Tweet:

#ruxsafecrack **25** 70 35

Hint: First number is **25**

Why build an autodialer?

- Good excuse to learn some basics about electronics and microprocessors
- We get to learn about mechanical locks
- We get to see the code we write interact with the physical world
- Has a well defined goal
- Expect to be achievable relatively quickly and cheaply
- Commercial solutions are limited and expensive

Building An Autodialer

Problem: We want to open a variety of combination locks that have an unknown code.

First Problem

How do we accurately turn a dial to a chosen number?

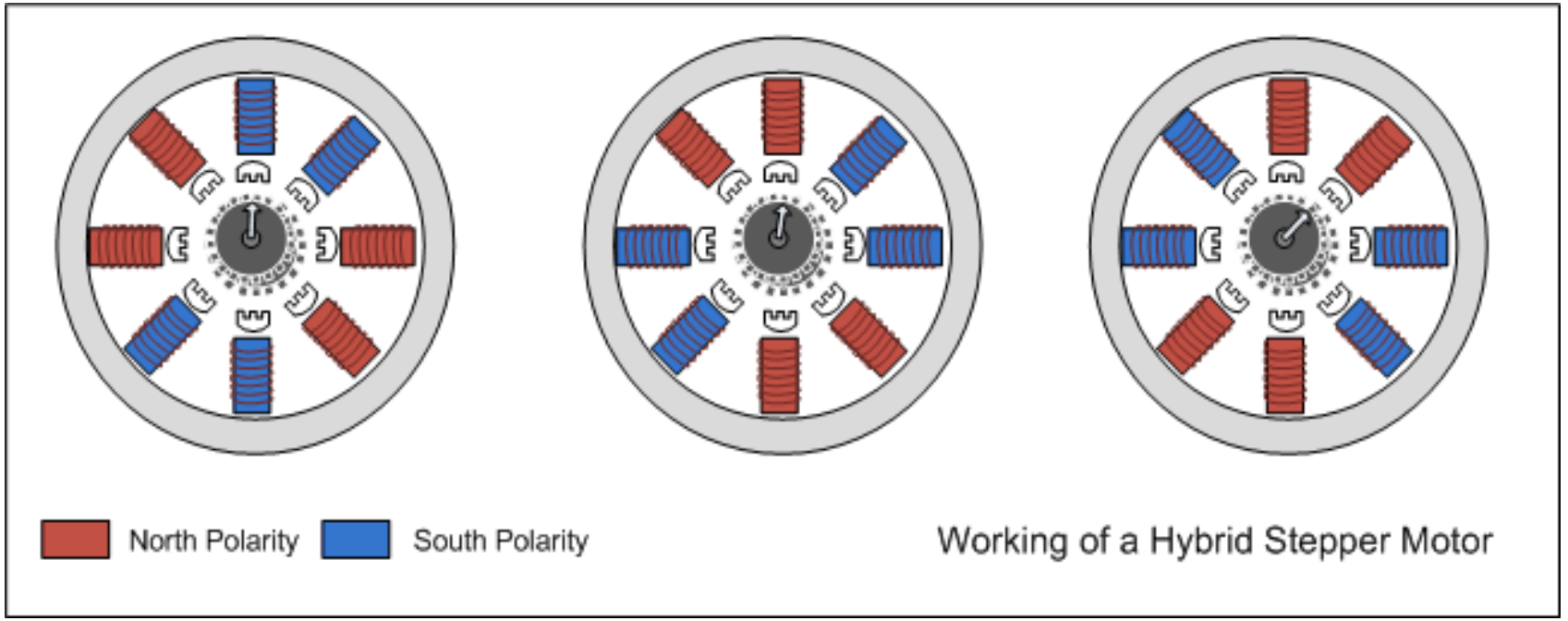
Solution

How Stepper Motors Work

A stepper motor is an electric motor that divides a full rotation into a number of equal steps.

The motor's position can then be commanded to move and hold at one of these steps without any feedback sensor

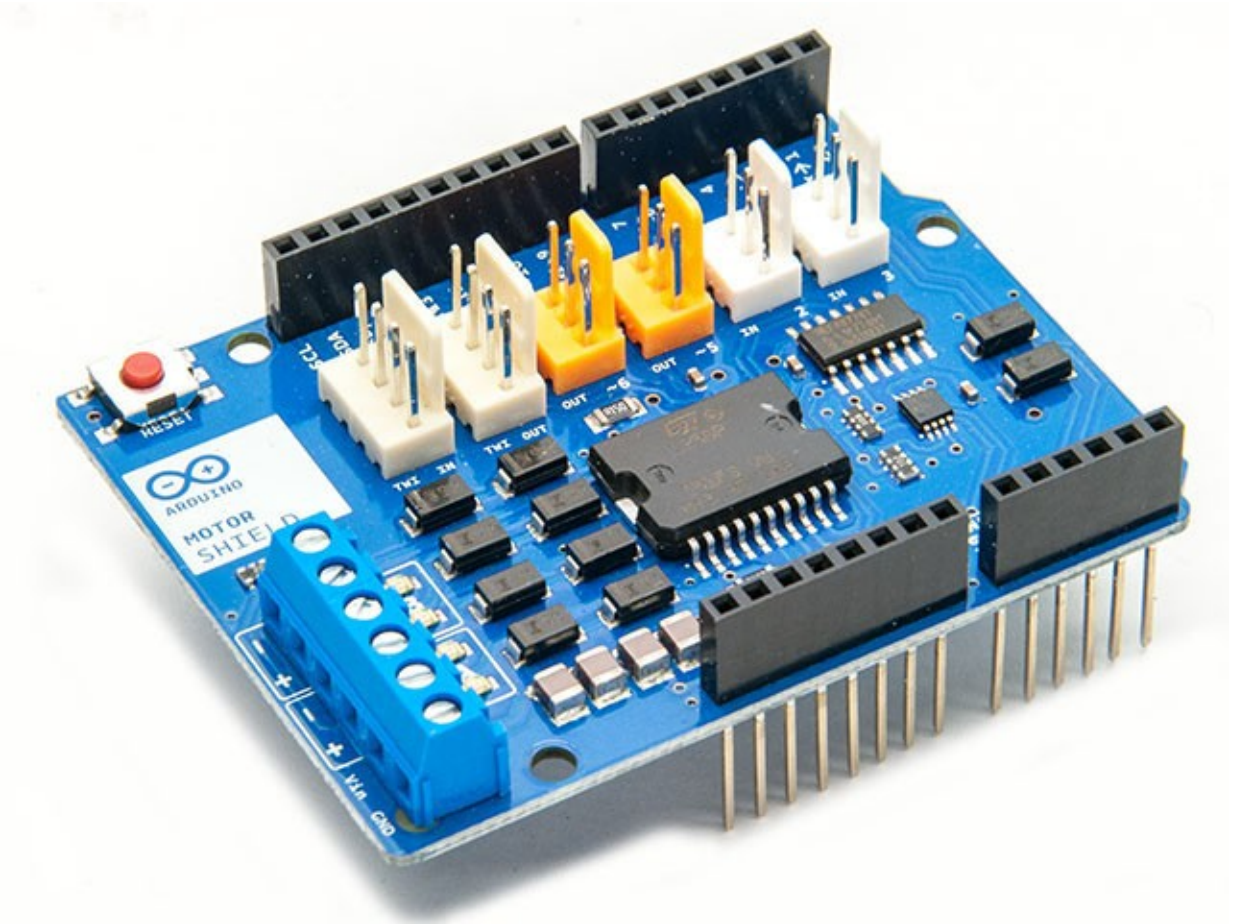
How Stepper Motors Work



How to control it?

An easy option is an Arduino stepper motor shield?

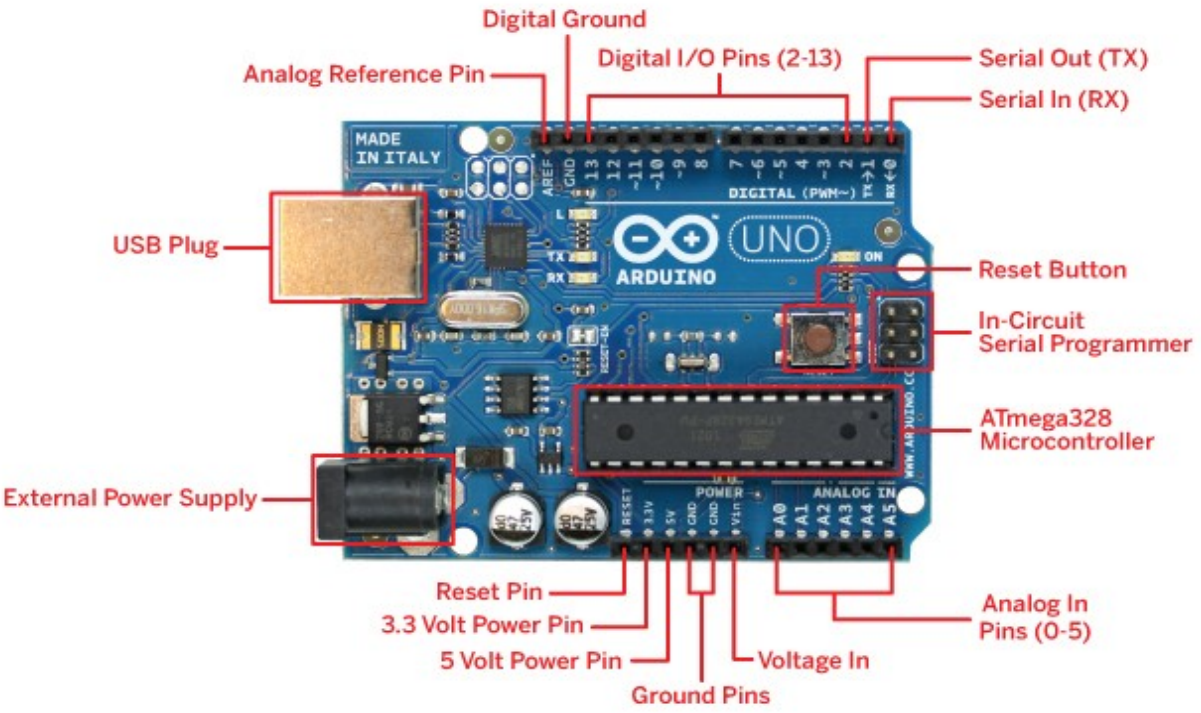
How to control it?



Arduino Intro

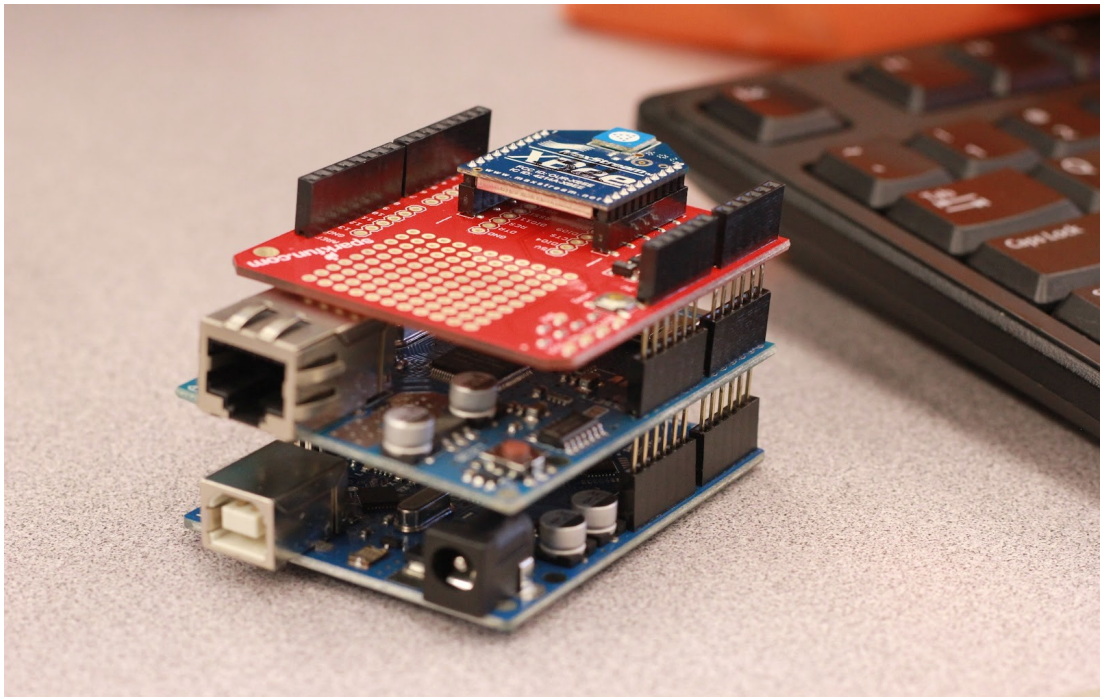
“Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software.”

Microprocessor + supporting infrastructure + easy I/O + IDE =



Shields

“Shields are boards that can be plugged on top of the Arduino PCB extending its capabilities. The different shields follow the same philosophy as the original toolkit: they are easy to mount, and cheap to produce.”



Where to from here?

- We have the concept of using a stepper motor and Arduino to control it.
- Let's work on the immediate problems:
  - How do we couple the motor to the lock and make sure it remains accurate?
  - How do we quickly and cheaply test this with a mechanical lock?

Test Lock

Addressing the second point first, we purchased a cheap master combination padlock to use for a basic PoC

Coupling Method

The first option that we thought of was using a mini lathe head as it would allow for varying dial sizes.

We disregarded as we found it likely to be painful to couple to a 5mm stepper motor shaft.

A better idea...

- Why not 3D print a custom part?
- A good idea but where to start and how to get it printed? I don't own a printer and I'm not paying $1500 for a printer to make a $10 part
- One downside is that the part would only work for one type of lock

3D Printing without a Printer

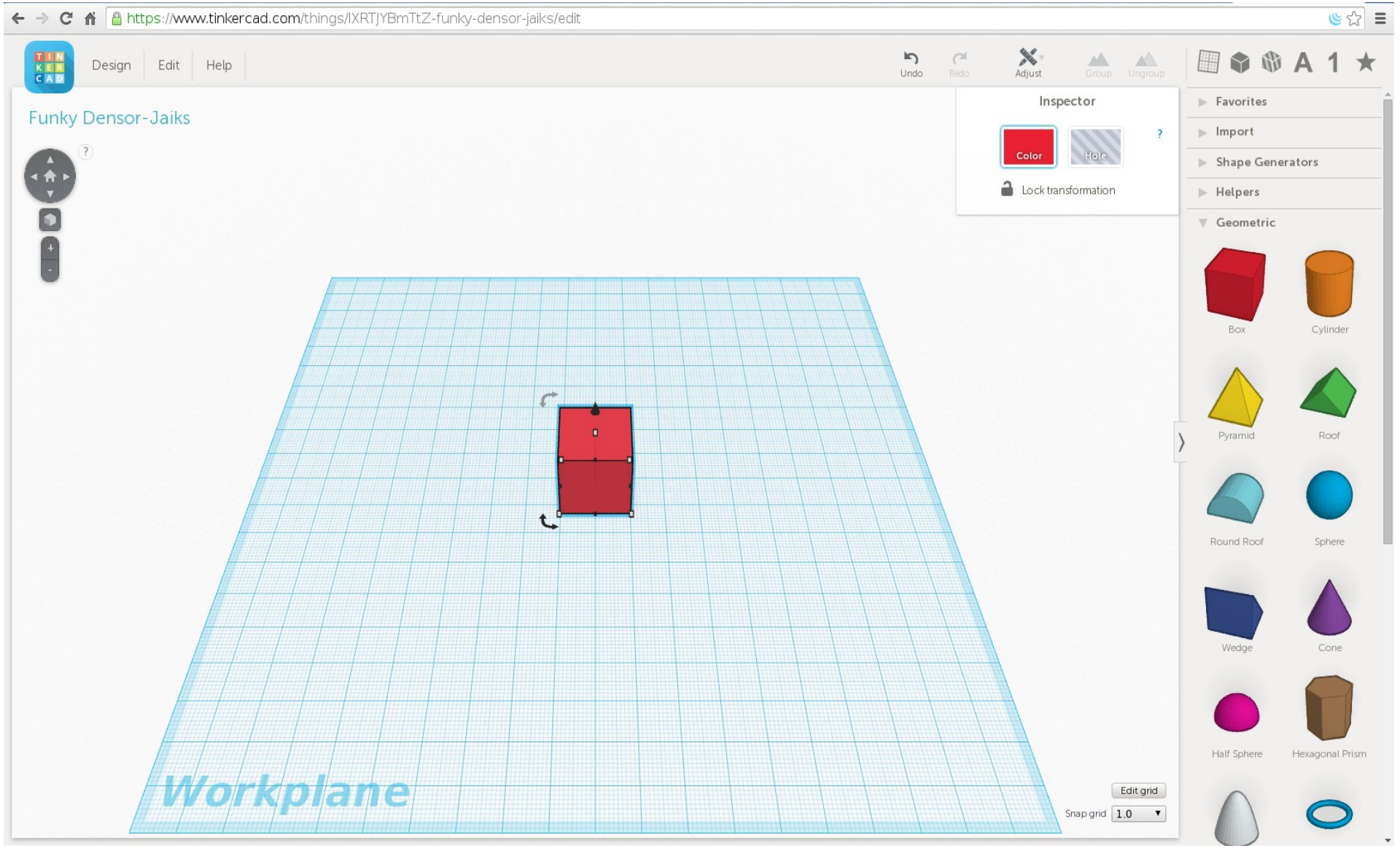
- Services exist to print parts
- People you might know may have a printer
- hackerspaces and libraries might have one you can use

- We decided to explore the local market place of 3D printing services

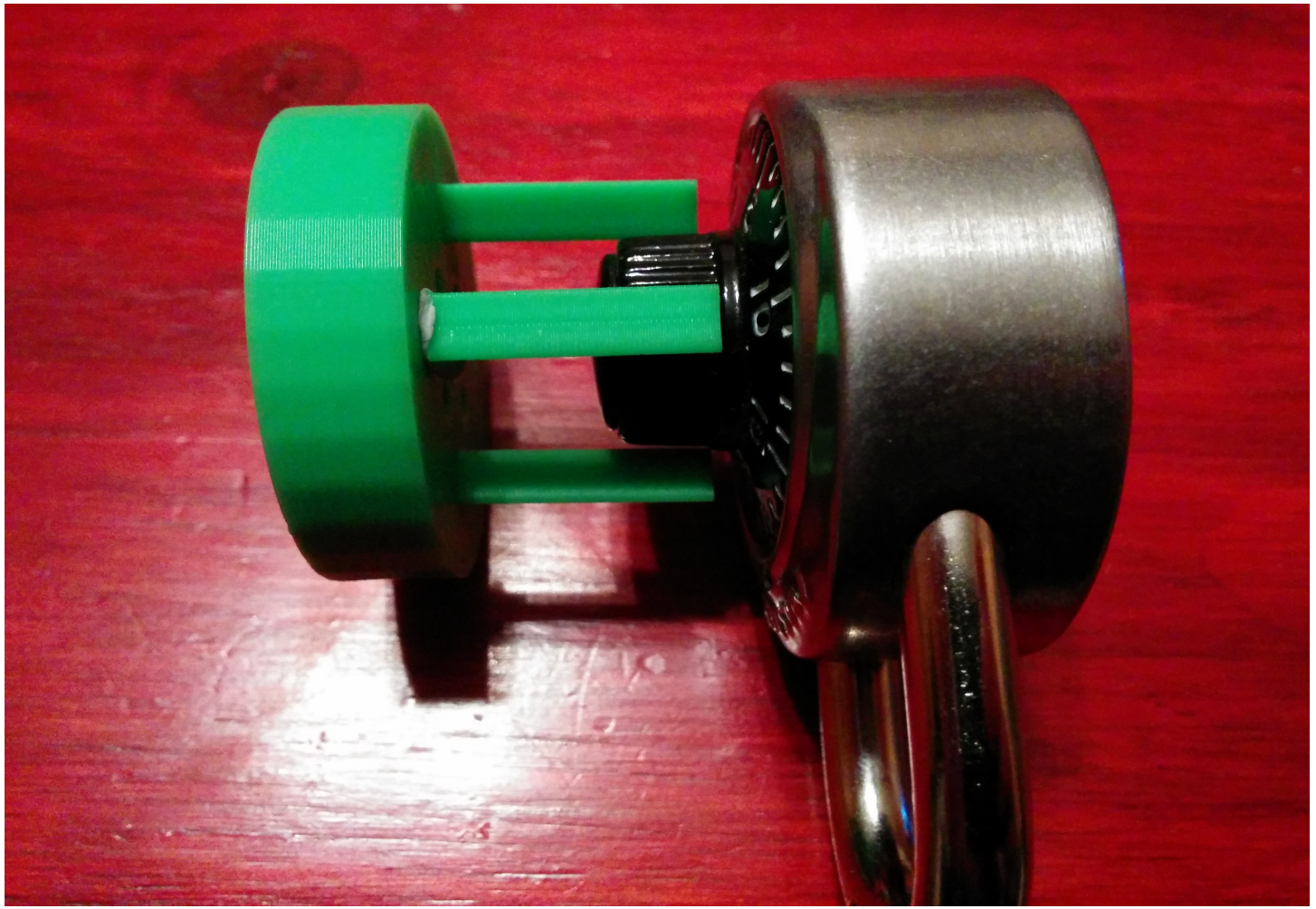
What can we use to design 3D parts?

- Wanted to avoid expensive options such as AutoCAD
- Common file format is STL (StereoLithography) which can be ASCII or binary
- Google search leads us to Tinkercad, a simple 3D design web application

Tinkercad

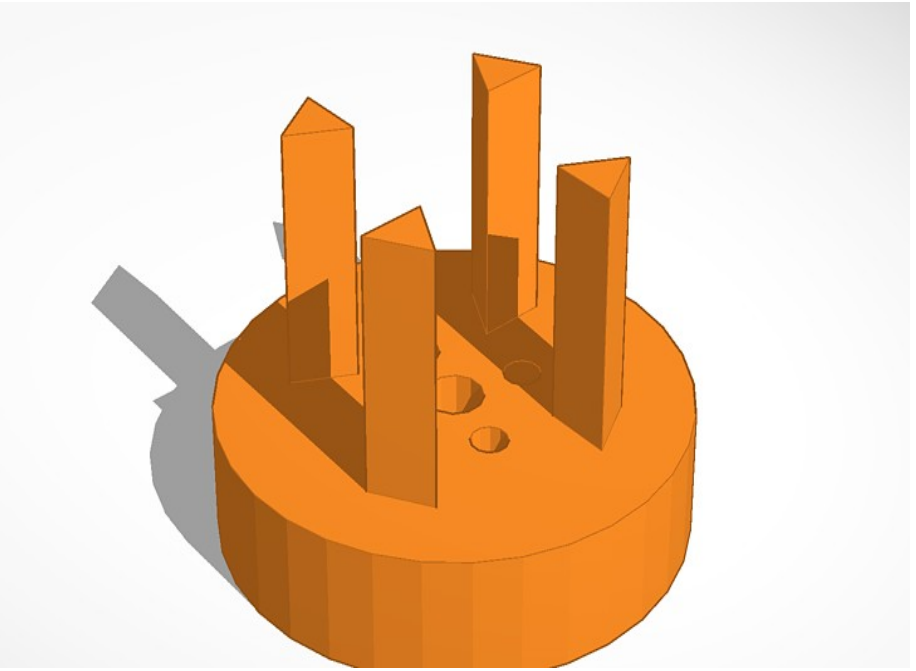

https://www.tinkercad.com/things/IXRTJYBmTtZ-funky-densor-jaiks/edit
Design
Edit
Help
Undo
Redo
Adjust
Group
Ungroup
Funky Densor-Jaiks
Inspector
Color
Hole
Lock transformation
Favorites
Import
Shape Generators
Helpers
Geometric
Box
Cylinder
Pyramid
Roof
Round Roof
Sphere
Wedge
Cone
Half Sphere
Hexagonal Prism
Workplane
Edit grid
Snap grid
1.0

Success!!



Tinkercad

We designed a part to couple the motor to the lock with the use of a stepper motor hub.



Writing Code

- Jay wrote some code to handle stepping and tracking of steps due to 4 steps being a cycle
- Was pretty messy, but it worked
- Luke found this thing called a “library” and got it to work. It even has cool things like acceleration/deceleration. Made everything much cleaner
- Next problem...

PoC Achieved

- We can now control a mechanical lock with an Arduino and enter a valid hardcoded combination

- We then pull on the shackle and it opens

Master Lock != Safe Lock

- What do we have to do to make it work with a more typical combination lock such as ones used on safes?

- Let's just buy a commonly used lock to find the differences

LA Guard

Model 3330 Lock:

LA Guard

Differences:

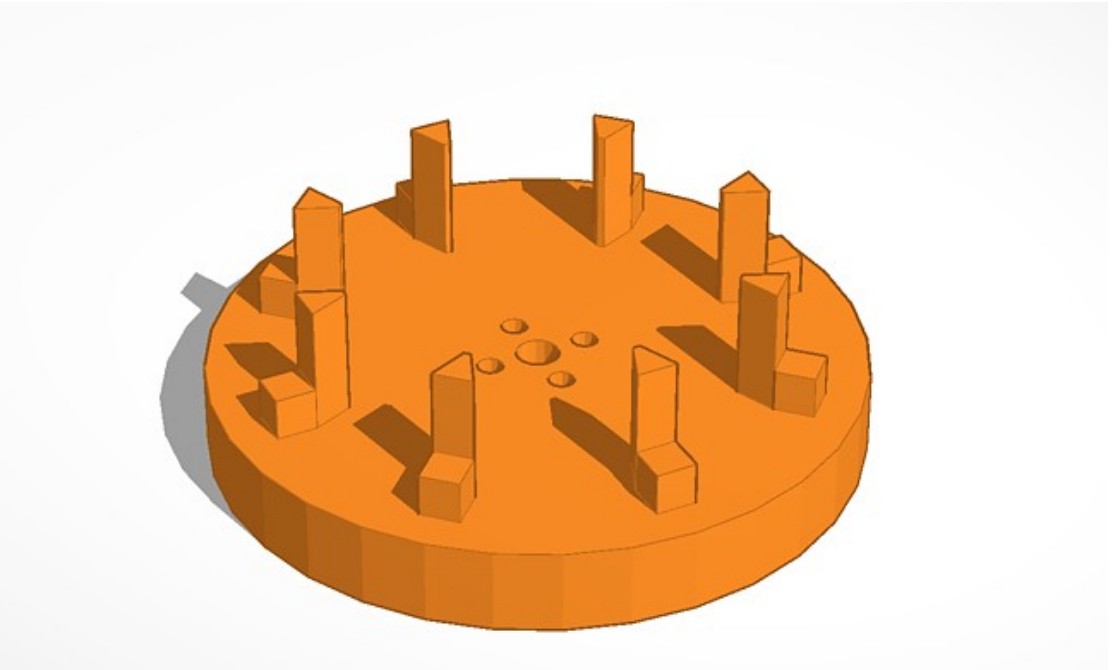
- 100 numbers, not 40
- Dial has a larger diameter
- Doesn't use an external action to open the bolt (instead you turn the dial to retract the bolt rather than manipulating some other mechanical component)

Hint: Combo is 25 45 ??

Tweet: #ruxsafecrack 25 45 ??

More 3D Design

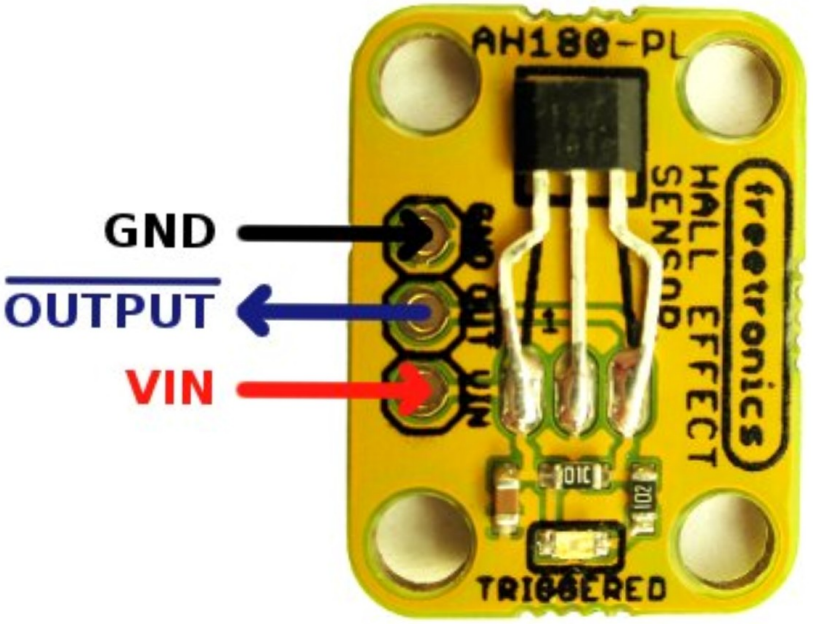
Needed different motor interface, back to Tinkercad. Also a good chance to improve the design.



Next Problem

After playing with the La Guard test lock and understanding the method to open it, we needed to devise a way to detect that it's open.

Hall effect sensors to the rescue!



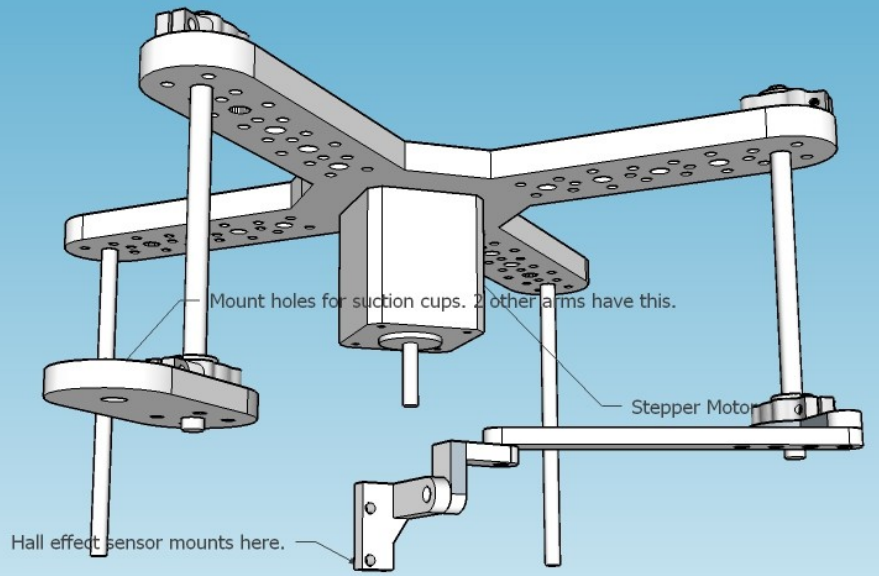
Next Problem

How to package this thing? What would it look like if it was used in Payday 2?

More 3D Design

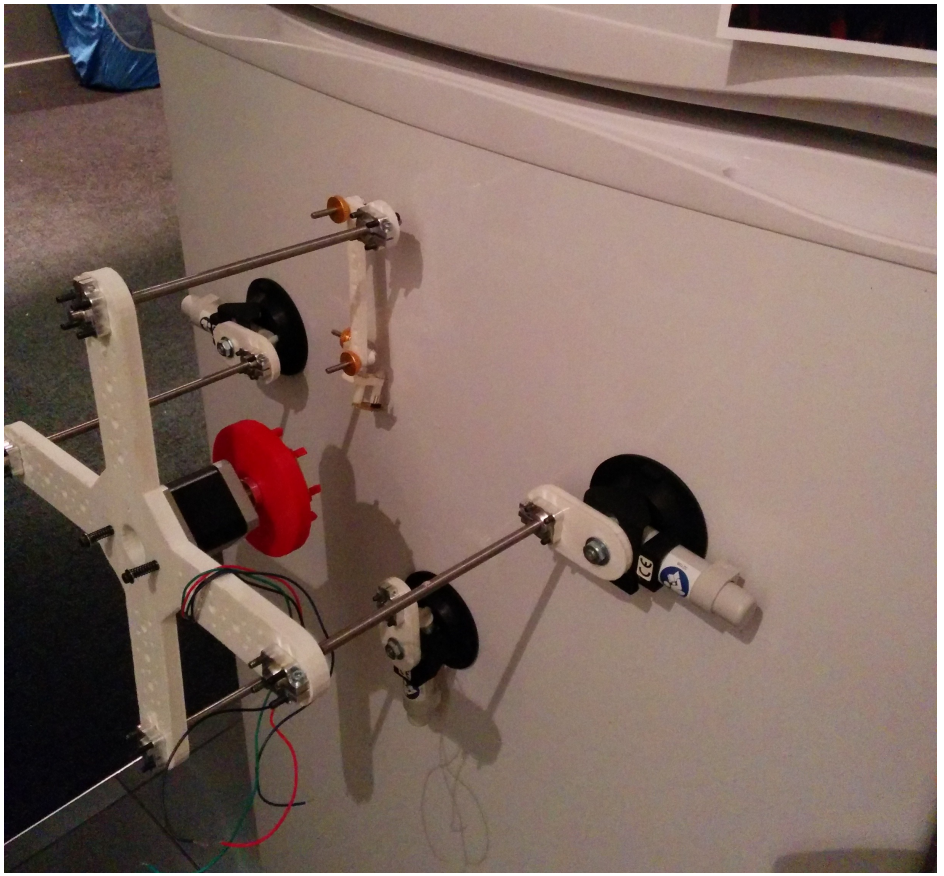
- Started build of chassis parts
- Tried to use Tinkercad again
- Flipped a table
- Found an alternative called SketchUp
- Remade parts in a fraction of the time
- Was confident in measurement, spacings and hole sizes. Yay!

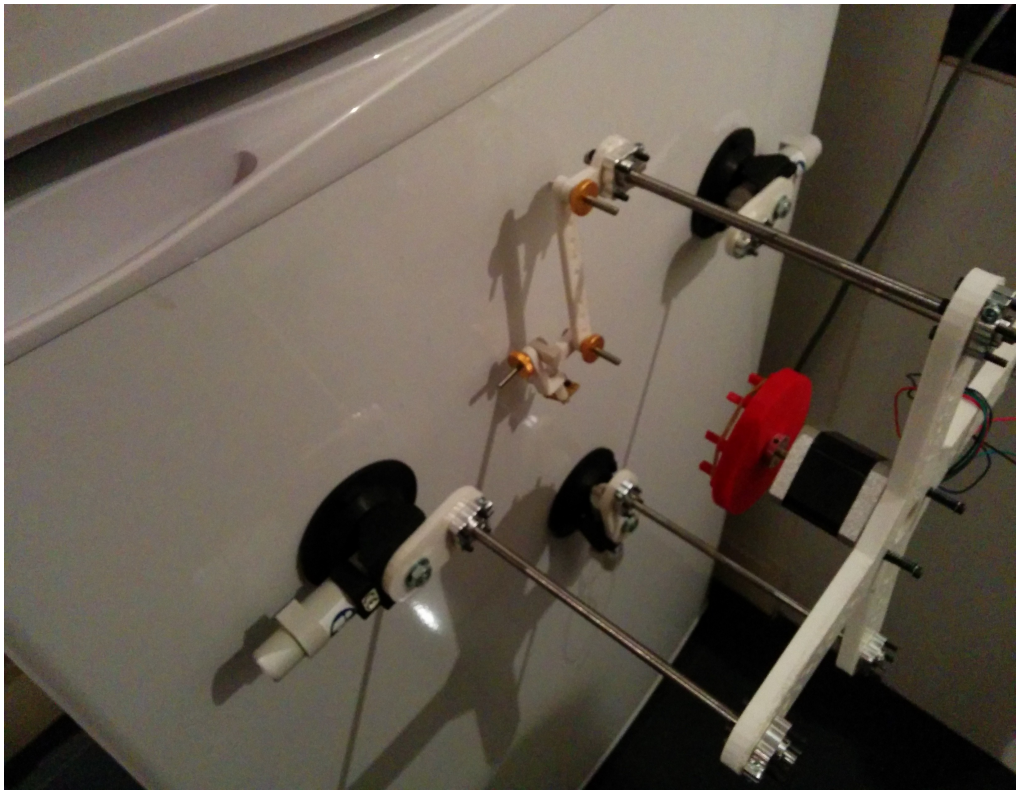
Concept


Mount holes for suction cups. 2 other arms have this.
Stepper Motor
Hall effect sensor mounts here.

Other Parts

First Assembly





Software time

- Calibration setup

- Enter Combination
- Check if lock is unlocked
- Repeat..

Software time

- AccelStepper library
- We wrote a layer on top of AccelStepper so we only deal with moving increments on the dial

- Reading from hall effect sensor is simple, digitalRead returns 0 or 1

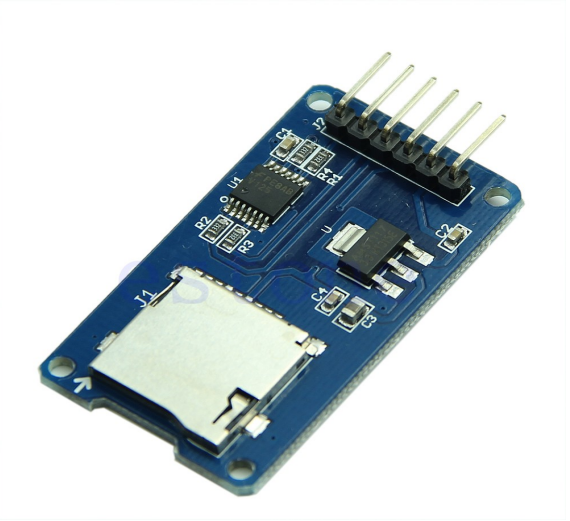
- Time to try one combination: 5 seconds

Which combinations to try?

- Try combinations from a list we provide (either SD card or over serial)
- Dumb bruteforce

Software time

- Let's use an SD Card:



- Allows us to generate the combination file offline and load it onto the SD Card
- Will also allow a tracking file to be written that can allow us to resume in case of an environmental event

Offline Combination Generator

- As mentioned before need to be smart about trying combinations as keyspace is not impossibly high but the time taken per try is not fairly slow
- Current methods are: Tryout/Default combinations, 10's, 5's, dates, recon numbers + permutations, everything else
- Also have randomization function and method order function
- Even accounts for manufacturing tolerances and forbidden zones

Forbidden What?

The lock we are using is classified as a UL Group 2 lock. This means that it needs to comply with the following parameters:

- “it is recommended to avoid an approximate 20 digit range for the last number on a three-number combination lock”

How many combinations?

- 3 Numbers
- First Number: 0-99
- Second Number: 0-99
- Third Number: 0-99

- Total: 1,000,000
- Time: 127 days (11 seconds per combination)

How many combinations?

- 3 Numbers
- First Number: 0-99
- Second Number: 0-99
- Third Number: 21-99

- Total: 800,000
- Time: 102 days (11 seconds per combination)

Manufacturing Tolerances

The lock we are using is classified as a UL Group 2 lock. This means that it needs to comply with the following parameters:

- “The lock can be dialed up to 1.25 digits above or below the actual set number and still open, essentially giving you a 2.5 digit window to hit”

How many combinations?

- 3 Numbers
- First Number: 0-99 (every second number)
- Second Number: 0-99 (every second number)
- Third Number: 21-99 (every second number)

- Total: 100,000
- Time: 13 days (11 seconds per combination)

Optimisation

- We have already optimised which combinations to try, now we want to optimise how we try the combinations

Optimisation

- After entering a combination, it is possible to try other third number possibilities without resetting the state of the first two numbers

- This is also possible with the second number

Optimisation

- 10,20,20
- 10,20,22
- 10,20,24
- ...
- 10,20,16
- 10,20,18

How fast can we try combinations?

- 3 Numbers
- First Number: 0-99 (every second number)
- Second Number: 0-99 (every second number)
- Third Number: 21-99 (every second number)

- Total: 100,000
- Time: 4.6 days (4 seconds per combination)

Controller Construction

Controller Construction

Twitter API

- Ruby and Python implementations were terrible
- PHP surprisingly better (J7mbo) for simple search

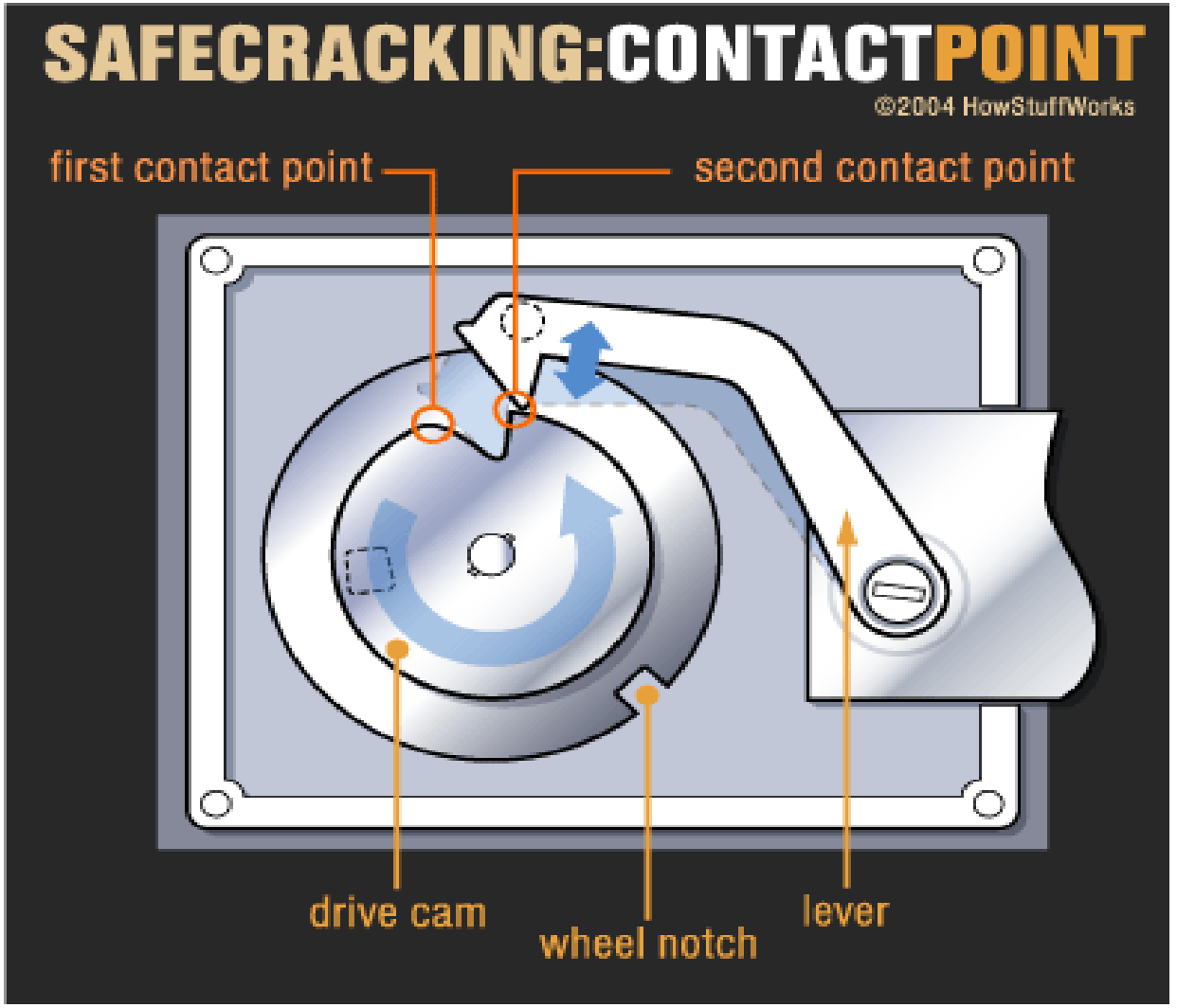
- Additional script takes stdin and writes it to serial (PySerial)

  Hint: Combo is 25 45 6?

  Tweet: #ruxsafecrack 25 45 6?

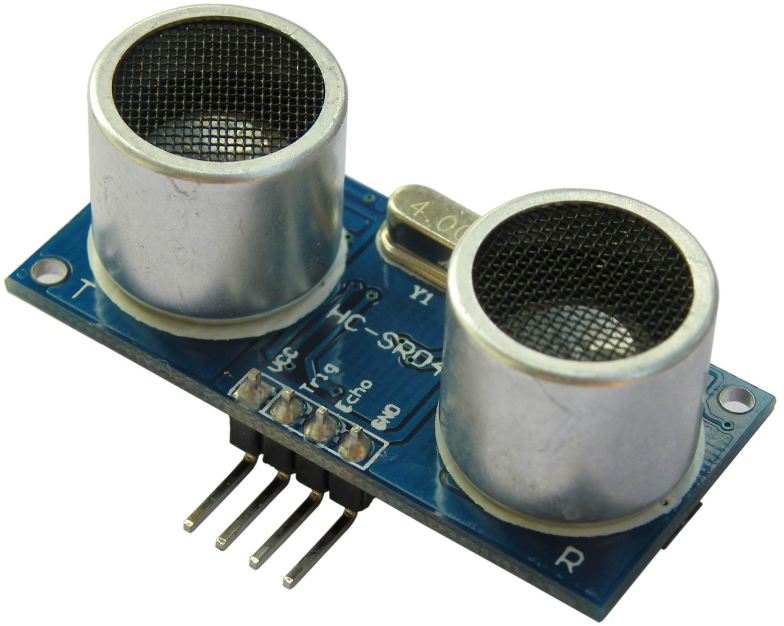
...Into the future

Manual lock manipulation



...Into the future

Automatic lock manipulation potentially using ultrasonic sensors or microphones



Thanks

3D PRINTER SUPERSTORE

your one stop 3d printer shop

resolution X

LIGHTING + RIGGING

Little Bird

ELECTRONICS

SecurusGlobal

3d Print Express

Questions?