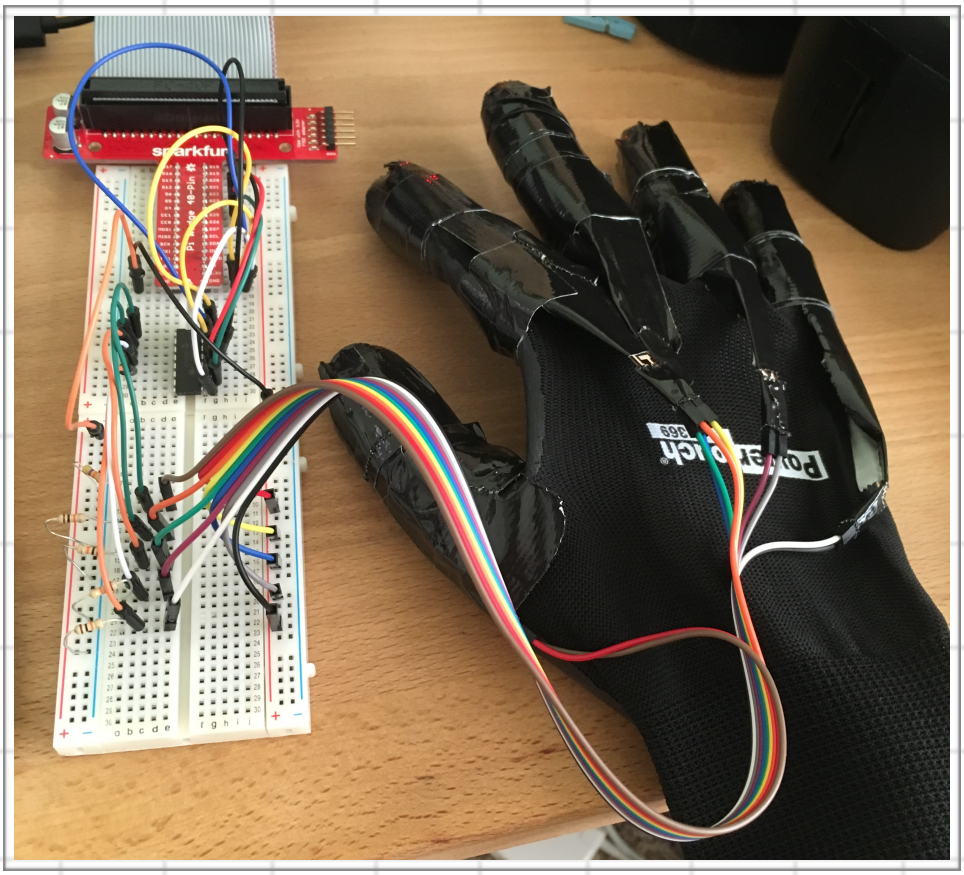
👋 Hands-On AR ✌🏻

(Augmented Reality)

Hardware personal challenge project



All iPad and Raspberry Pi source code available:
https://github.com/incanus/Hands-On-AR

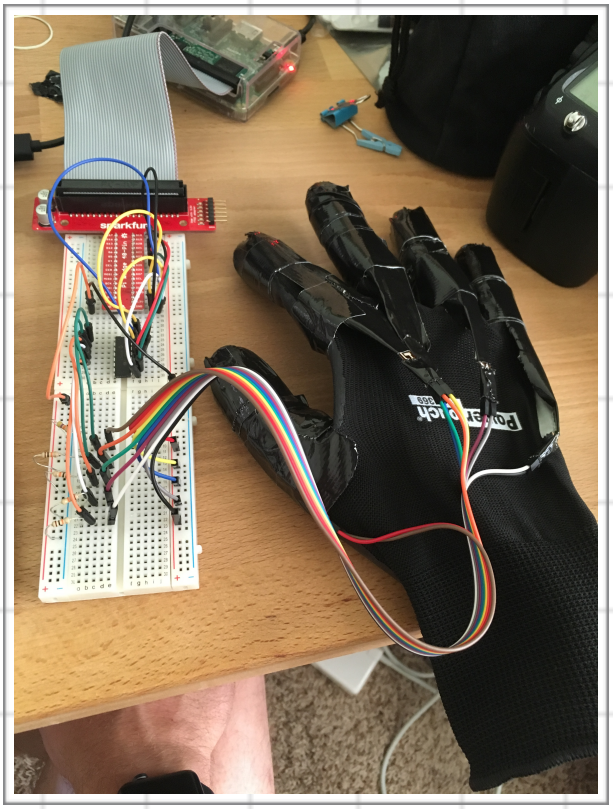
Project Goals

- Build on knowledge of AR workings on iOS
- Experiment with AR & the physical world
- Learn about collaborative AR
- Learn how to build things out of electronics
- Learn more about physical prototyping & construction of projects
- Challenge myself & have fun!

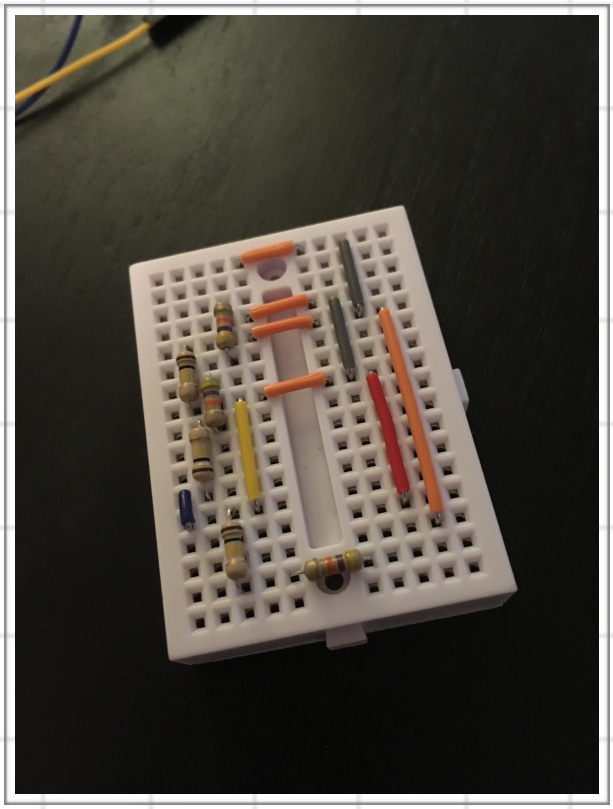
(Also: not rip off the Power Glove too much)

Building & Testing

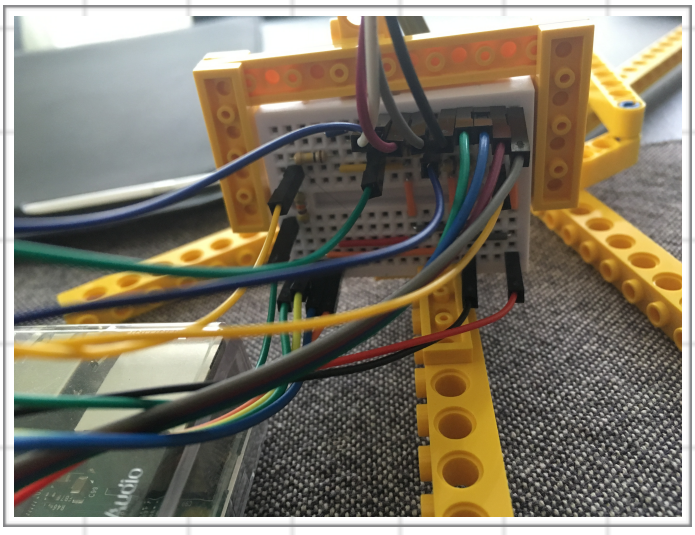
Garden gloves from the store



Early assembly with large prototype circuit

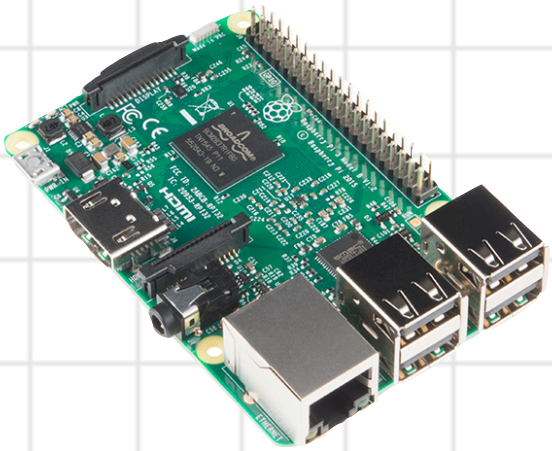


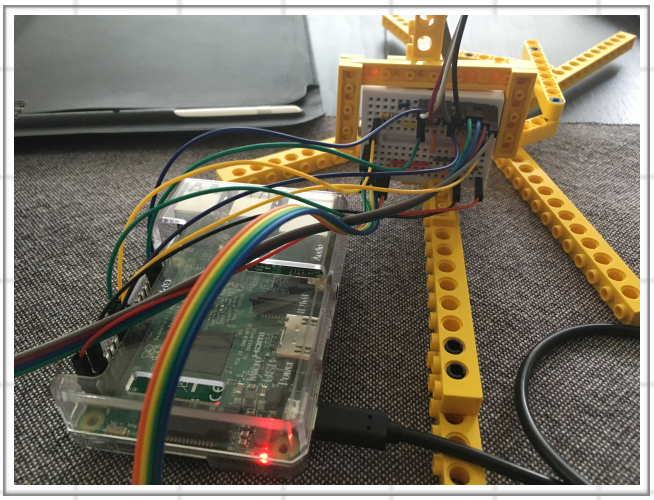
Early circuit layout



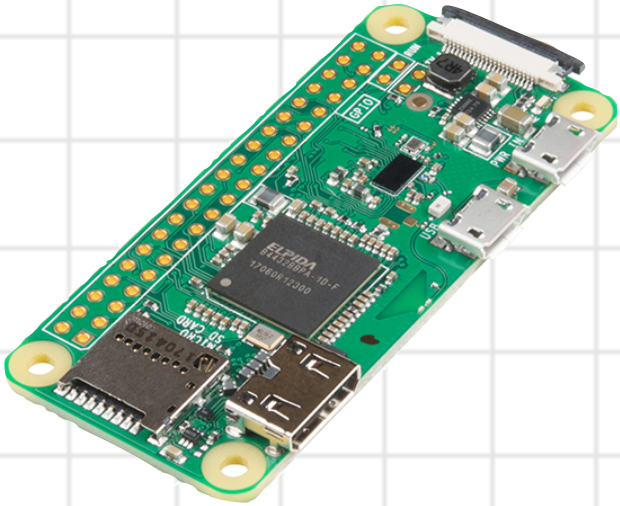
Getting messy!

Computer “Brain”



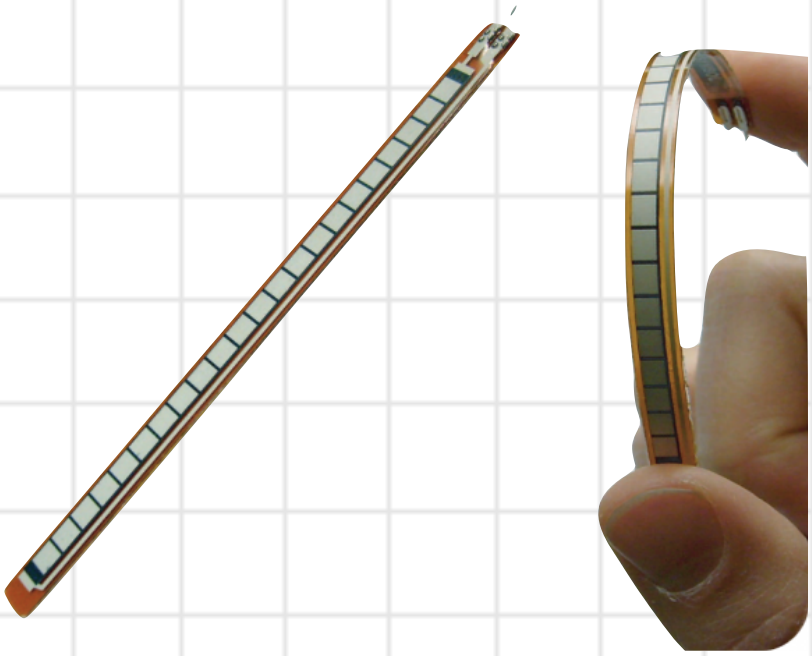


Full size Raspberry Pi computer
Used when prototyping and testing

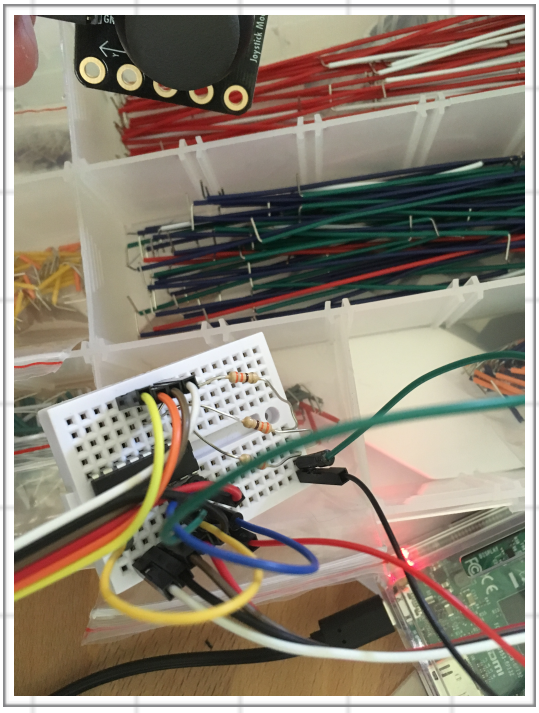


Tiny Raspberry Pi “Zero”
Used when shrinking the final design

Core Components

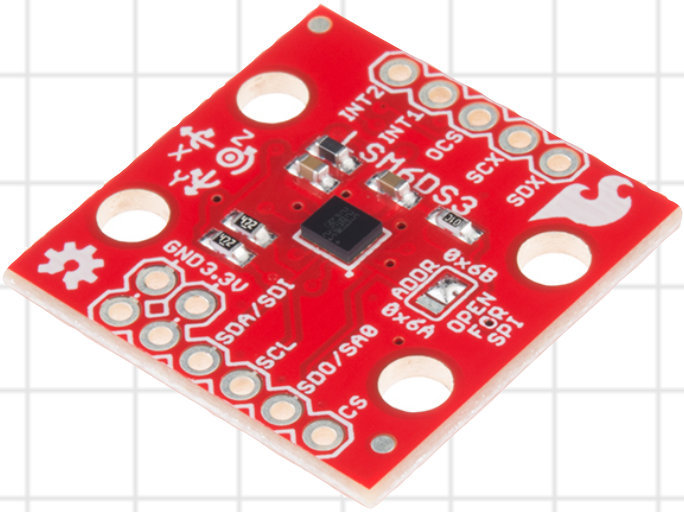


Flex sensors—variable resistors that let different amounts of electricity through at different bends

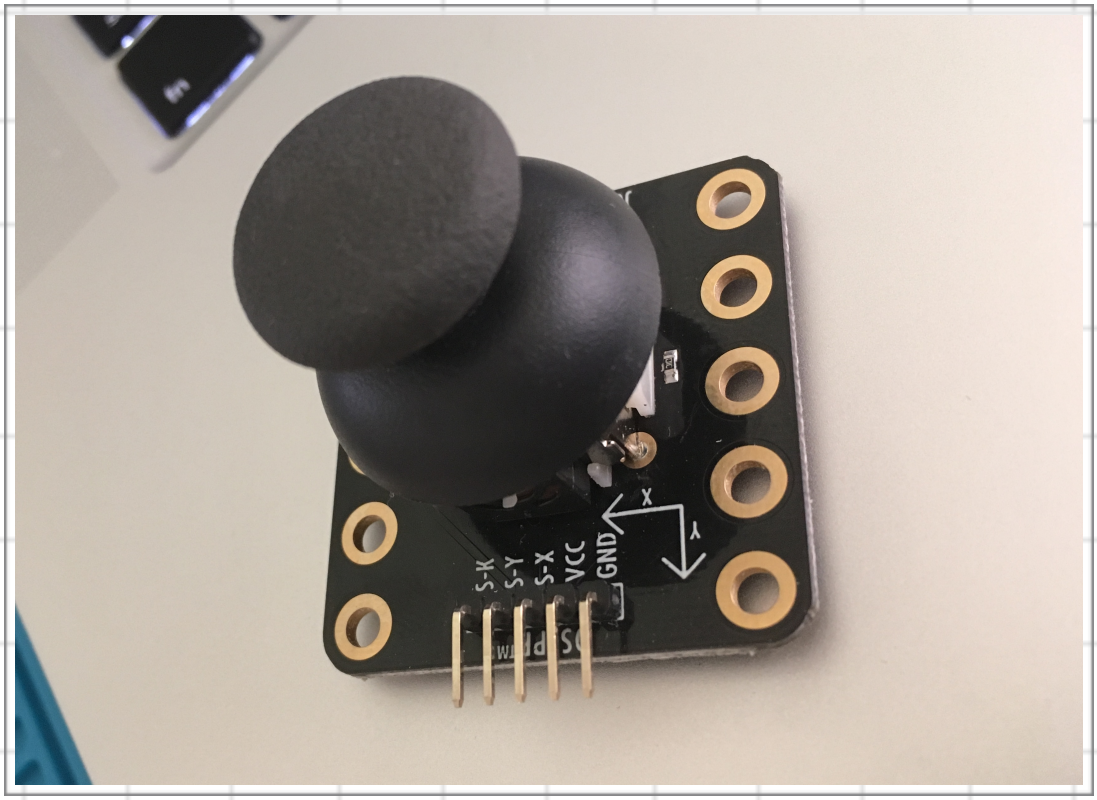


Color-coded connector wires for easier hookup

More Components

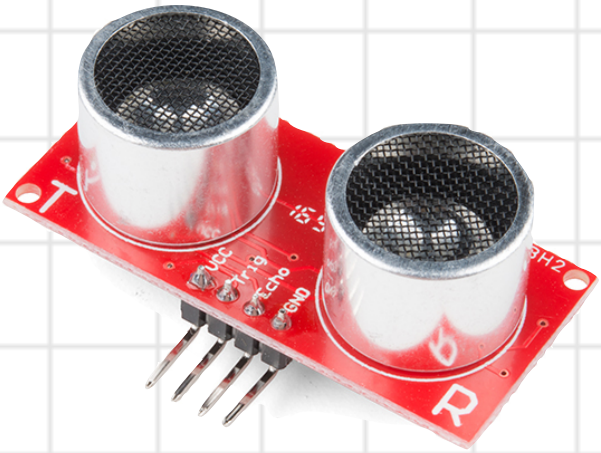


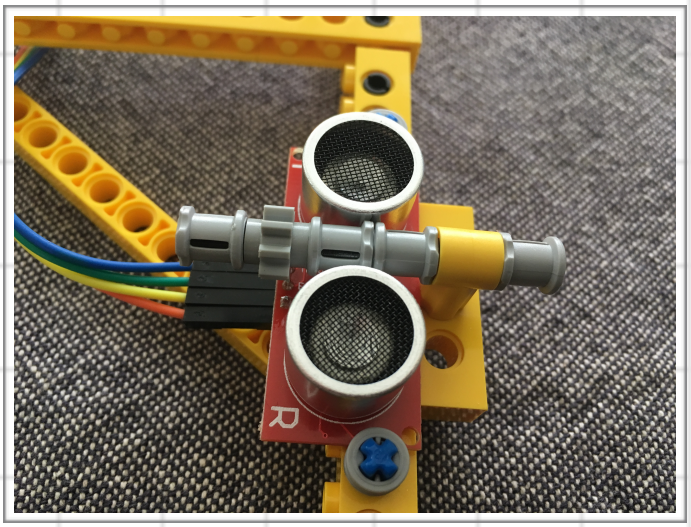
Gyro & accelerometer
Measures rotation in space



Eventually used a joystick for the most reliable positional control

Early Spatial Tracking Idea

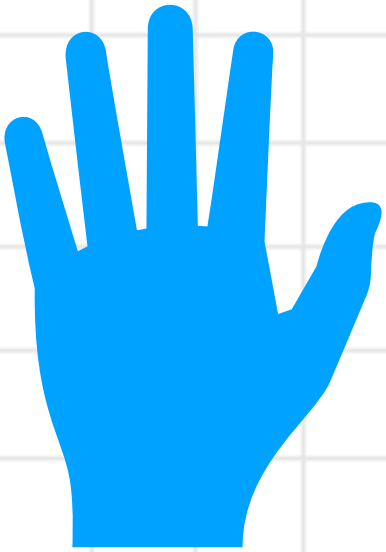




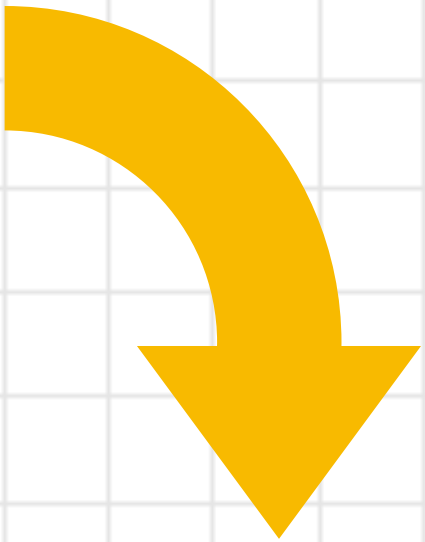
Ultrasonic distance tracker bounces audio & measures the echo time, like a bat's "night vision"

Three trackers mounted in a LEGO rig

Device Communications



Hand & Raspberry Pi sends simple data about finger bend, tilt, and position over wifi



iPad receives data and updates the augmented reality scene

Raspberry Pi & iPad Programming

```python
#!/usr/bin/env python

import time
import Adafruit_MCP3008
import socket
import sys

sys.path.append('LSM6DS3')
from LSM6DS3 import LSM6DS3

CLK  = 18
MISO = 23
MOSI = 24
CS   = 25
mcp = Adafruit_MCP3008.MCP3008(clk=CLK, cs=CS, miso=MISO,
mosi=MOSI)

lsm = LSM6DS3()

VCC = 3.316
RDIV_THUMB = 47100
RDIV_FINGER = 10000
STRAIGHT_THUMB = 68000
STRAIGHT_FINGER = 101100
BEND_THUMB = 98000
BEND_FINGER = 250000

try:
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
except socket.error, msg:
    print 'Failed to create socket. Error code: ' + str(msg[0]) +
' , Error message : ' + msg[1]
```

```python
print("  " + labels[0] + ratios[0])
for i in range(4):
    print("  " + labels[i + 1] + ratios[i + 1])

print("")

x = str(int(lsm.getXRotation()))
y = str(int(lsm.getYRotation()))
z = str(int(lsm.getZRotation()))
raws.extend([x, y, z])
print("  x:" + x)
print("  y:" + y)
print("  z:" + z)

print("")

try:
    s.sendto("a:" + ",".join(raws), ("192.168.1.7", 8080))
except socket.error:
    print 'Send failed'
    # sys.exit()

time.sleep(0.1)
```

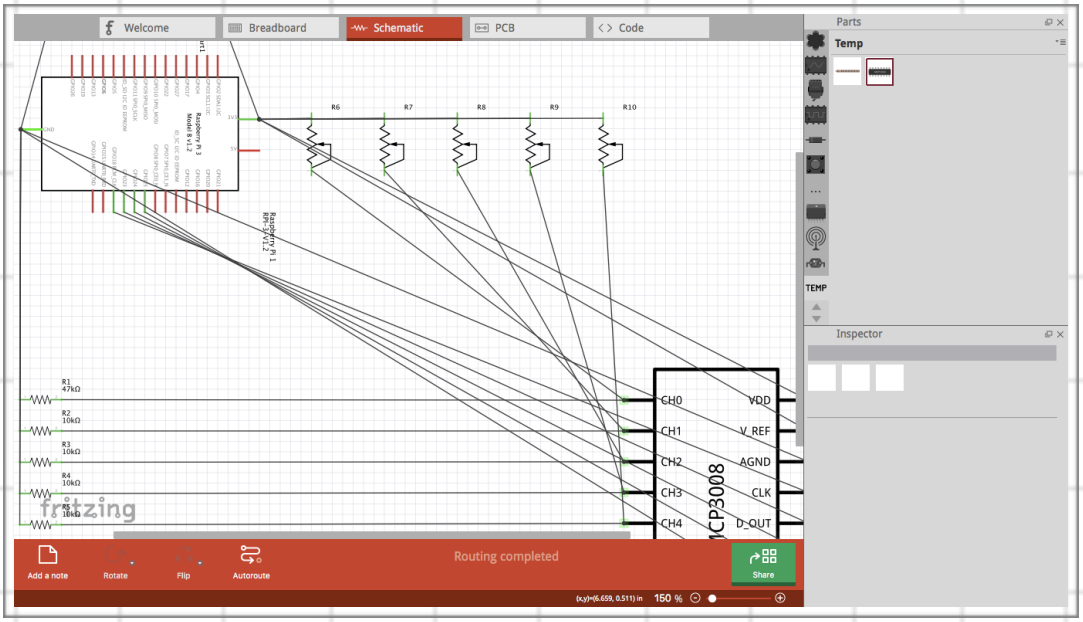
Raspberry Pi uses the Python language, which I didn't know when I started this project

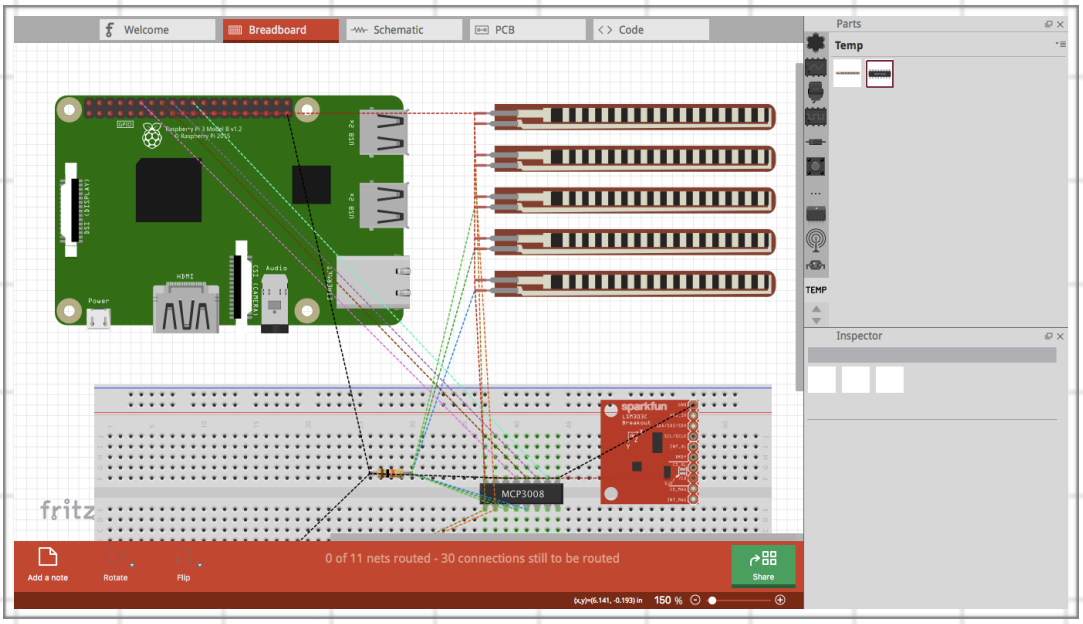
```swift
let palm = SCNTube(innerRadius: 0.35, outerRadius: 0.5, height: 0.1)
palm.materials = [darkMaterial, darkMaterial, colorMaterial, colorMaterial]

let palmNode = SCNNode(geometry: palm)
palmNode.name = "palm"
palmNode.position = SCNVector3(0, 0, 0)
palmNode.transform = SCNMatrix4MakeRotation(.pi / 2, 1, 0, 0)
self.addChildNode(palmNode)

let ls: [CGFloat] = [0.6, 0.75, 0.65, 0.5]
let rs: [Float] = [.pi / 6, .pi / 11, -.pi / 11, -.pi / 6, .pi / 3]
let xs: [Float] = [-0.35, -0.1, 0.1, 0.35, -0.35]
let ys: [Float] = [0.35, 0.5, 0.5, 0.35, 0]

for f in 1...4 {
    let finger = SCNCylinder(radius: 0.1, height: ls[f - 1])
    finger.materials = [colorMaterial, darkMaterial, darkMaterial]

    let fingerNode = SCNNode(geometry: finger)
    fingerNode.name = "finger.\(f)"
    fingerNode.position = SCNVector3(0, 0, 0)
    fingerNode.pivot = SCNMatrix4MakeTranslation(0, -0.5, 0)
    fingerNode.transform = SCNMatrix4MakeRotation(rs[f - 1], 0, 0, 1)
    fingerNode.transform = SCNMatrix4Translate(fingerNode.transform, xs[f - 1],
        ys[f - 1], 0)
    self.addChildNode(fingerNode)
}

let thumb = SCNCylinder(radius: 0.1, height: 0.4)
thumb.materials = [colorMaterial, darkMaterial, darkMaterial]

let thumbNode = SCNNode(geometry: thumb)
thumbNode.name = "finger.0"
thumbNode.position = SCNVector3(0, 0, 0)
thumbNode.pivot = SCNMatrix4MakeTranslation(0, -0.5, 0)
```

```swift
if sections[0] == "a" {
    let values = sections[1].split(separator: ",")

    var fingers = [UInt]()
    for v in 0...4 {
        if (Int(values[v])! > 0) {
            fingers.append(UInt(values[v])!)
        } else {
            fingers.append(0)
        }
    }

    var axes = [UInt]()
    for v in 5...7 {
        axes.append(UInt(values[v])!)
    }

    DispatchQueue.main.sync { [unowned self] in
        self.hand?.setFingers(fingers)
        self.hand?.setTilt(axes)
    }
} else if sections[0] == "b" {
    let values = sections[1].split(separator: ",")

    let x = Float(values[0])!
    let y = Float(values[1])!
    let dropped = (values[2] == "1" ? true : false)

    DispatchQueue.main.sync { [unowned self] in
        self.hand?.setPosition(x: x, y: y, dropped: dropped)
    }
}
```

iOS uses the Swift language, which I knew well going into the project

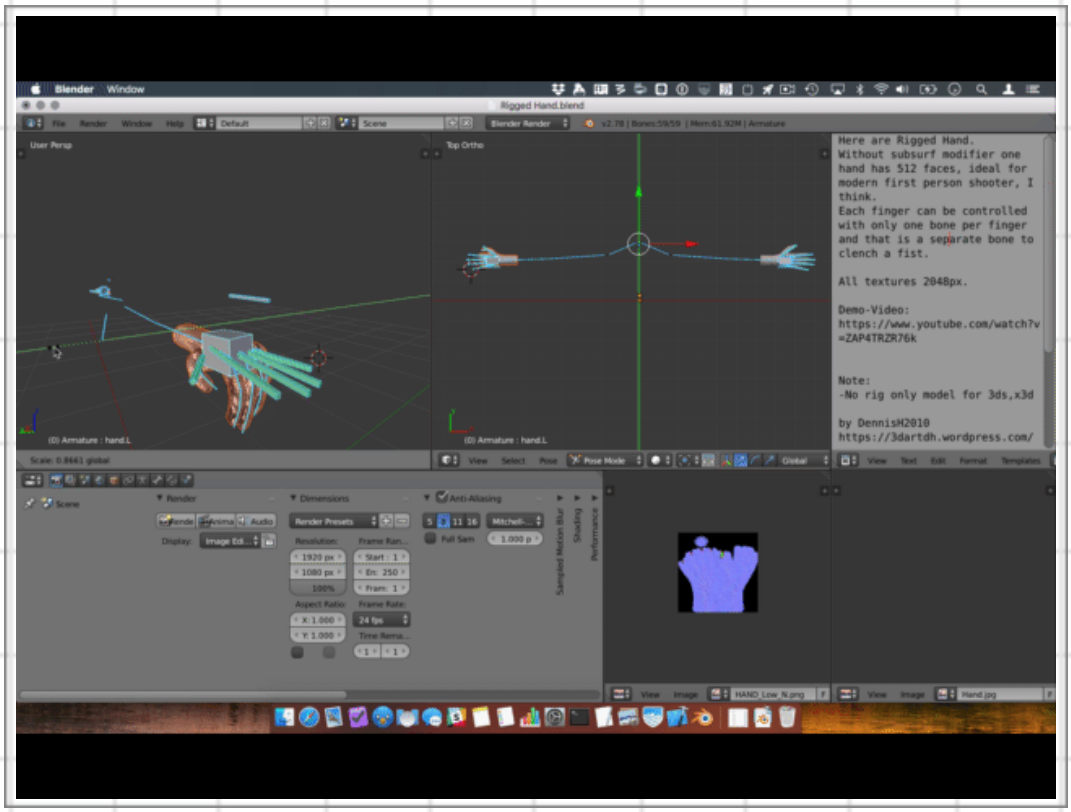
Laying Out Circuit Diagrams

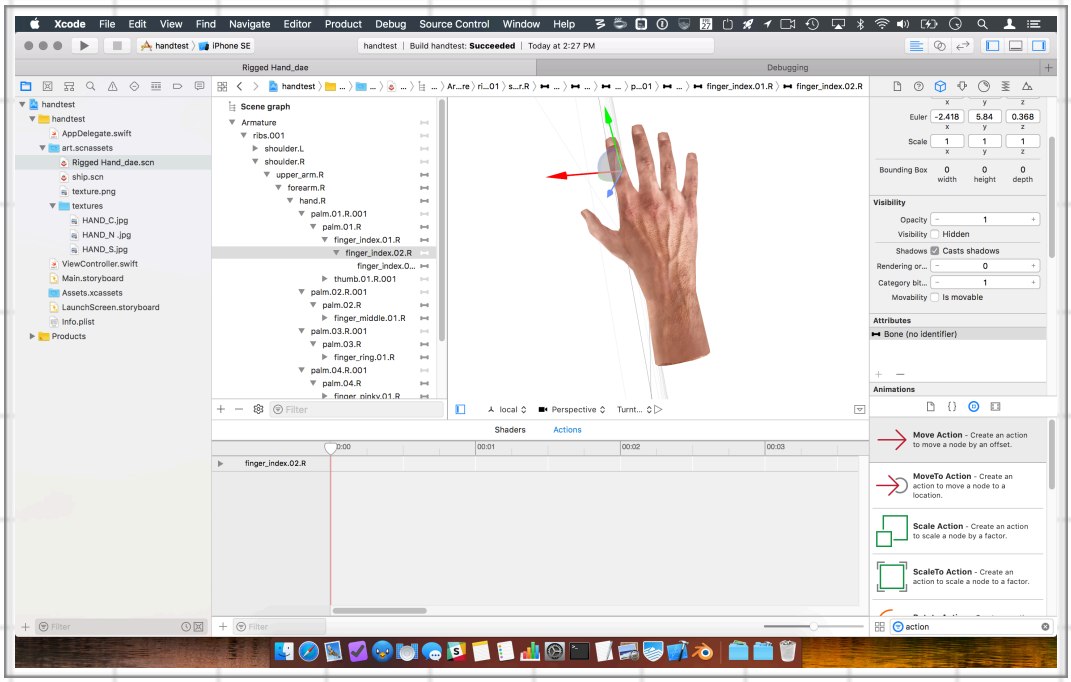




Free app “Fritzing” helped plan with both schematics and physical wiring

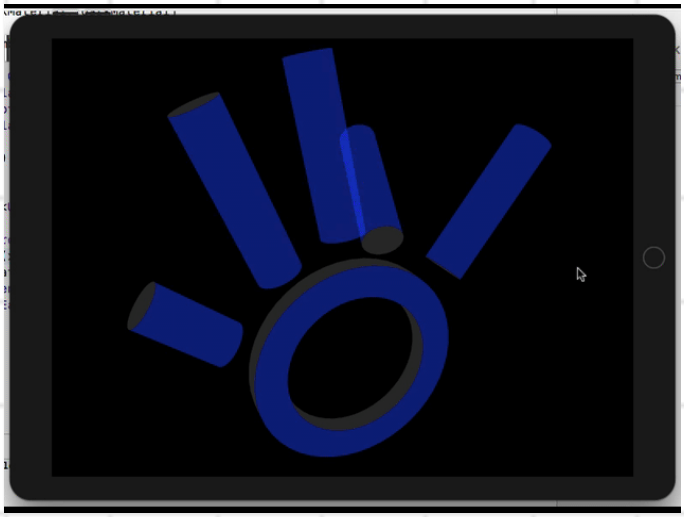
Early Hand Ideas



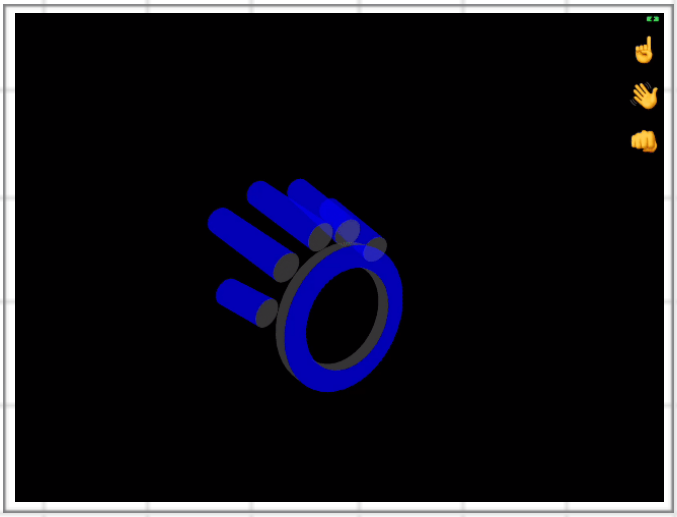


Very difficult to make a realistic looking and acting hand from a 3D model

Hand Simplification & Refinement

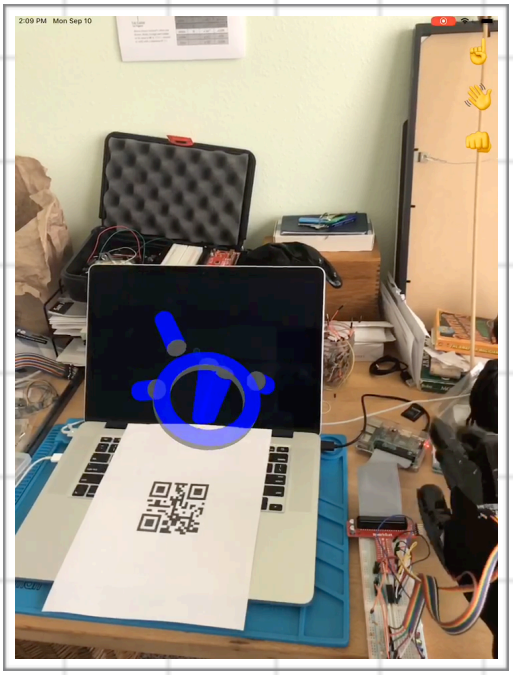


Early computer-driven pre-recorded gestures



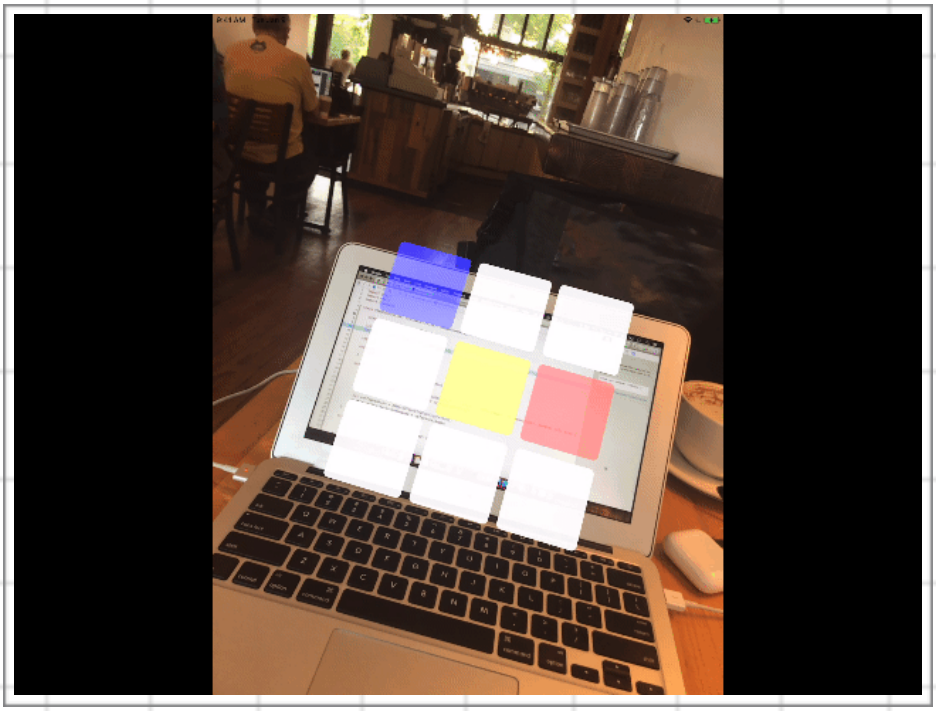
Adding an emoji menu for quick gesture testing

Finally hooking up the glove!

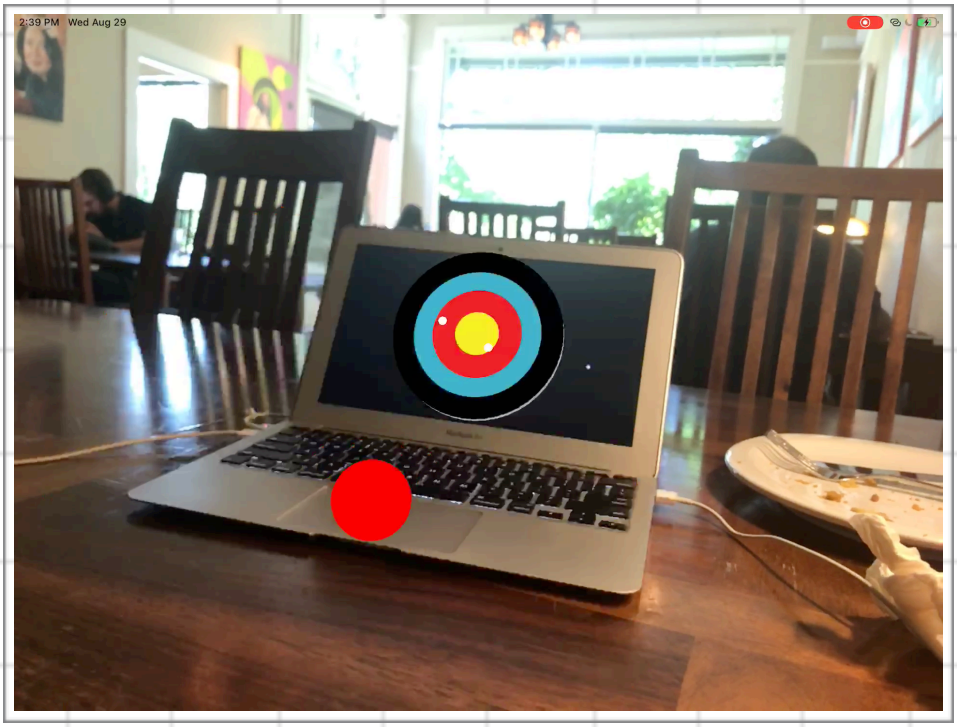


First tests in AR, hovering over a QR code

Augmented Reality Game Ideas

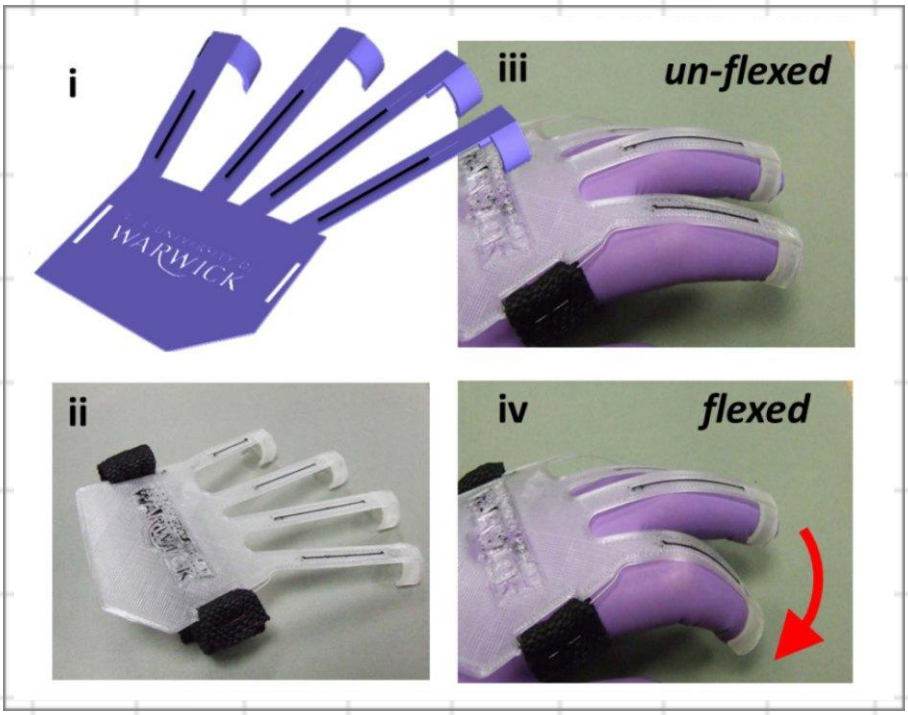


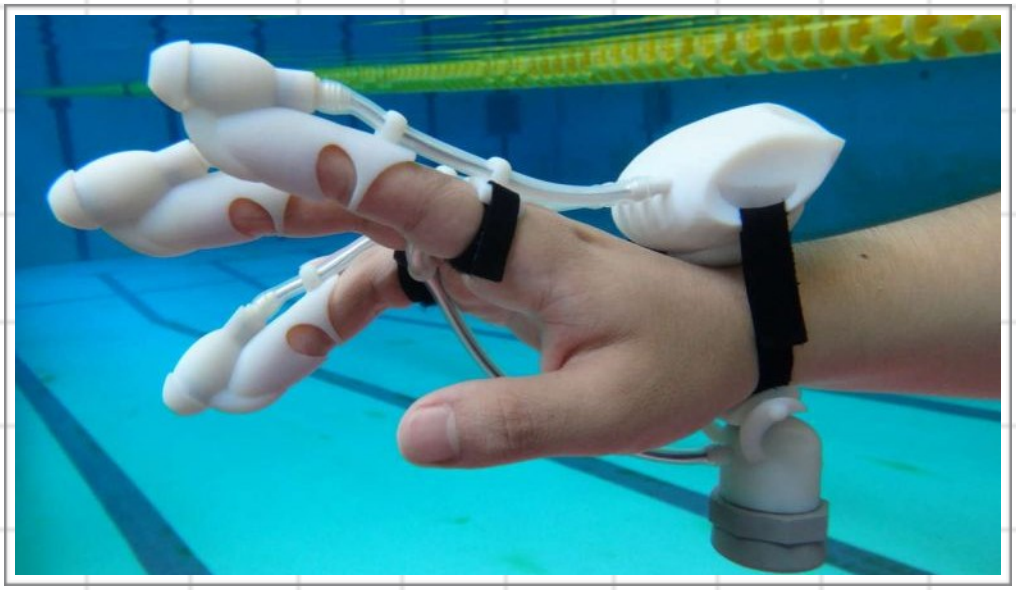
Early experiment with 3D tic-tac-toe
Could be played by a player on each side in 3D space



3D target or throwing practice

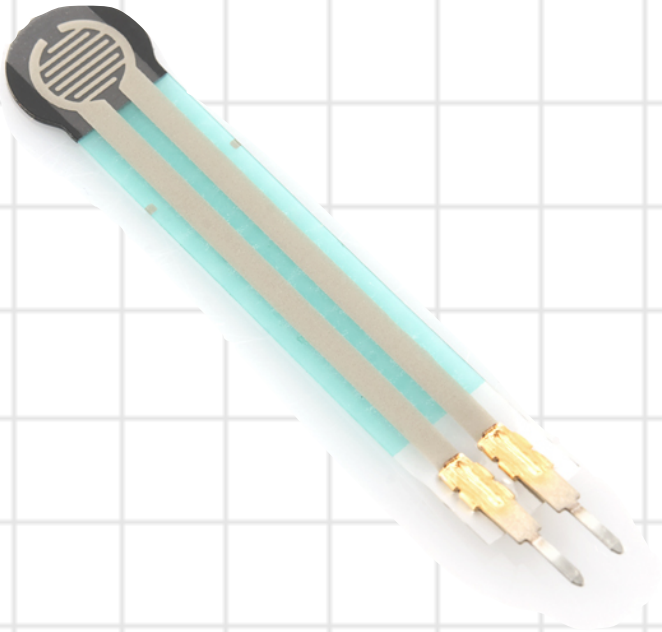
Future Directions





Possible 3D printed structure

Future Directions



Pinch sensors work similar to flex sensors but measure grip

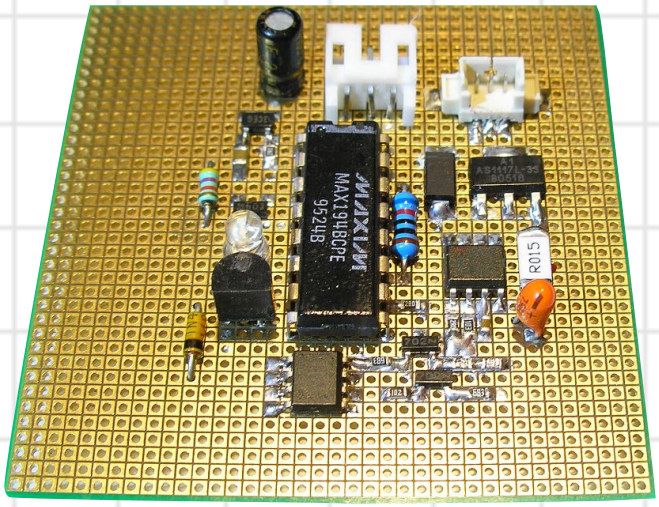
Mini vibration motor
Let user “feel” objects at fingertips

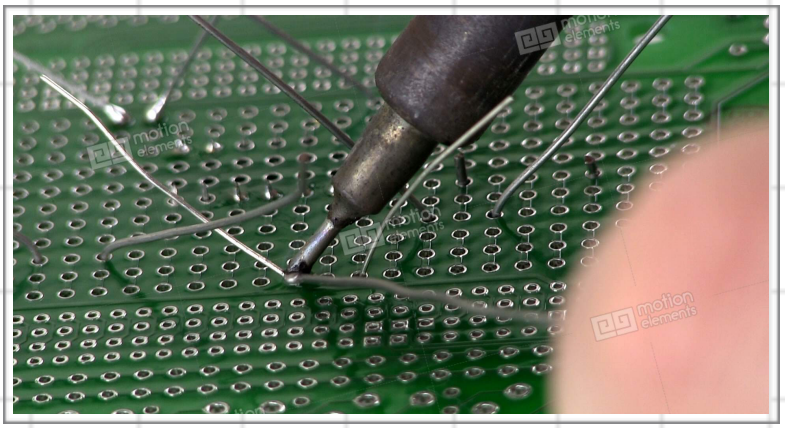
Future Directions

LEGO exoskeleton idea
Flickr user **Milan Sekiz**, *"A.R.M"*

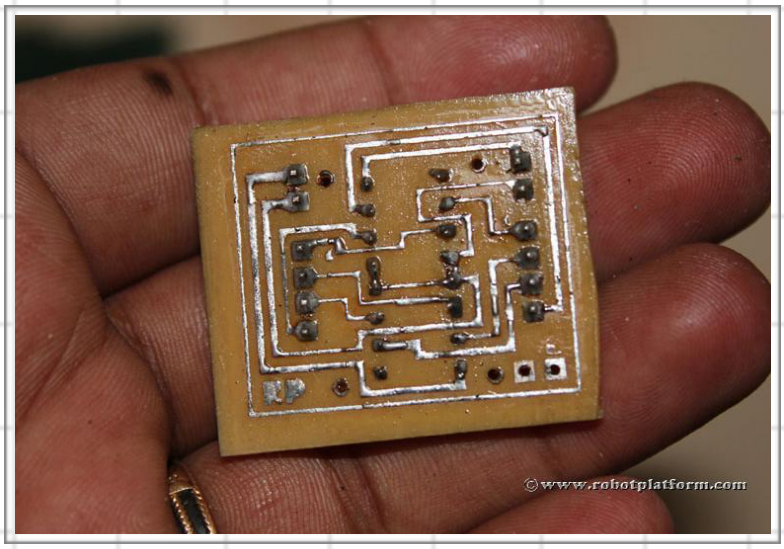
Aluminum foil-based capacitance 3D spatial tracking
https://makezine.com/projects/a-touchless-3d-tracking-interface/

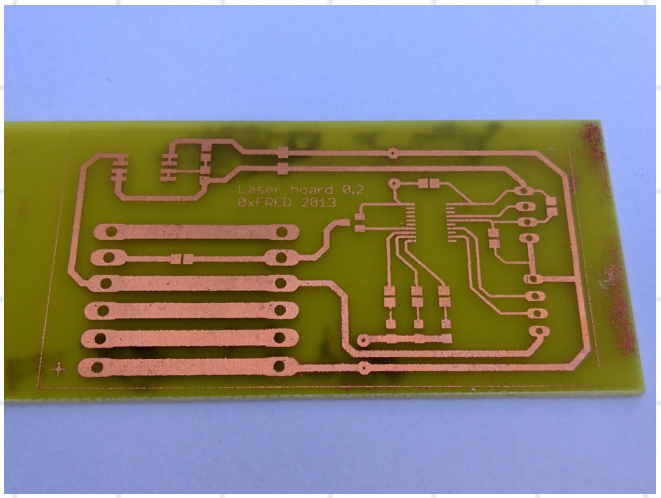
Future Directions





Learn how to through-hole or surface mount circuit for lower profile





Learn PCB etching for more streamlined and professional look

Project Log

Jul 25

- Maker Faire application accepted! Now to get started…

Jul 27

- Experiments with 3D hand models
- Learned about 3D file formats & animations/rigging

Jul 31

- Started interactive iPad hand model app

Aug 1

- Brainstorming game concepts

Aug 5

- Get a Windows install going to experiment with AutoDesk 3ds Max

Aug 13

- Lots of experiments with 3D hand animations

Aug 20

- Brainstorming more AR software concepts

Aug 27

- ARKit experimentation
- Lots of learning about ways to have 3D objects interact

Aug 29

- Bought first glove components
- Trying out iPad stands
- Working version of ball target practice

Aug 30

- Research other gloves prior work
- Tic tac toe first cut
- Learn about USB OTG and why you need it

Sep 4

- Learn how to make DIY USB OTG cable
- Learn about SSH over USB for Pi setup
- Other experiments with Pi headless setup
- Learn about analog vs. digital circuits on the Pi
- Learn about MCP3008 microcontroller for ADC

Sep 5

- Learn about 3v vs. 5v logic levels and bidirectional conversion
- Learn about SPI for 6DOF gyro chip
- Testing all the components

Sep 6

- Learn how to hook up MCP3008 chip
- Learn about Fritzing for circuit design & documentation
- Basic glove hookup (flex sensors)
- Start to learn Python
- Learn Python sockets
- Learn about Python strings, loops, arrays/lists, and exceptions
- Try out various libraries for Swift sockets on the iPad
- Compare TCP vs. UDP for project needs
- Got realtime animations working
- Started to figure out incremental rotation

Sep 7

- More learning about rotations
- Learn about breakaway headers for chip mounting
- Learn about Python module paths & loading
- Learn how to solder

Sep 8

- Learn I2C protocol for gyro chip

Sep 10

- Getting hand into AR
- AR node scaling/transform
- Safely tapping into Pi 5v power, logic level conversion, external power, resistance
- Built LEGO ultrasonic sensor rig
- Trying to get all three proximity sensors reading

Sep 11

- Basic triple proximity working
- Possible interference echos between sensors?
- Interference actually from LEGO gears in front of sensors!
- Basic dual-client messaging
- Rough position tracking
- It’s gonna be a challenge to make this look good…

Sep 12

- Researching other position ideas - Hall effect for magnetic proximity?
- Joystick testing
- Learn how to hook up joystick hookup & send data packets
- Try soldering headers onto the Pi Zero 😬
- Learn about ways to make circuits tiny
- Start thinking about enclosure/ wearable
- Debug wrong SPI pins on replicated flex sensor circuit

Sep 13

- Flex tuning & simplification
- Creating AR anchor laminated cards
- Create project binder
- Experiment with through-hole breadboarding

Sep 14

- Final, last-minute prep…

Thanks for stopping by!

All iPad and Raspberry Pi source code available:
https://github.com/incanus/Hands-On-AR