Making Magic

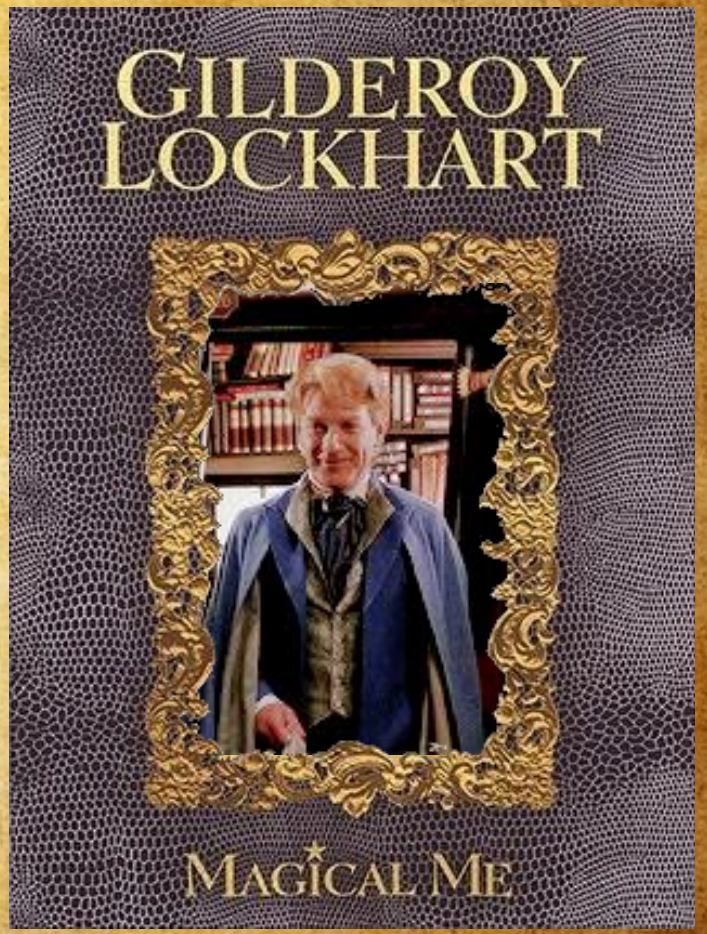
with

LabVIEW™ Community Edition

By Quentin "Q" Alldredge
Q Software Innovations

About "Magical Me"

- Feel Free to call me "Q"
- Work of the Aerospace and Defense Industry
  - First at ATK (Now Northrup Grumman)
  - Now at Hill Air Force Base
- LabVIEW Consulting as Q Software Innovations



Contact Q

- ★Phone/Text: +1 (435)-730-1198
- ★Email: q@qsi.dev
- ★Website: www.qsi.dev
- ★LinkedIn: www.linkedin.com/in/quentin-q-alldredge
- ★LabVIEWWiki.org: Q
- ★Twitter: @QSI_Q
- ★NI Community Forums: TheQ
- ★LAVA Forums: The Q
- ★Stackoverflow: TheQ

Background

- Inspiration from the Harry Potter books by J.K. Rowling
- Four Houses of Hogwarts School of Witchcraft and Wizardry
- Weasley Family's Magic Clock

Background

- Inspiration from the Harry Potter books by J.K. Rowling
- Four Houses of Hogwarts School of Witchcraft and Wizardry
- Weasley Family's Magic Clock

Background

- Inspiration from the Harry Potter books by J.K. Rowling
- Four Houses of Hogwarts School of Witchcraft and Wizardry
- Weasley Family's Magic Clock

Background

★Weasley Family's Magic Clock

Examples

- "Build Your Own "Weasley" Location Clock!" by ppeters0502

  https://www.instructables.com/Build-Your-Own-Weasley-Location-Clock/

- "IoT Location Sensing Picture Frame" by allie.fauer

  https://www.instructables.com/IoT-Location-Sensing-Picture-Frame/

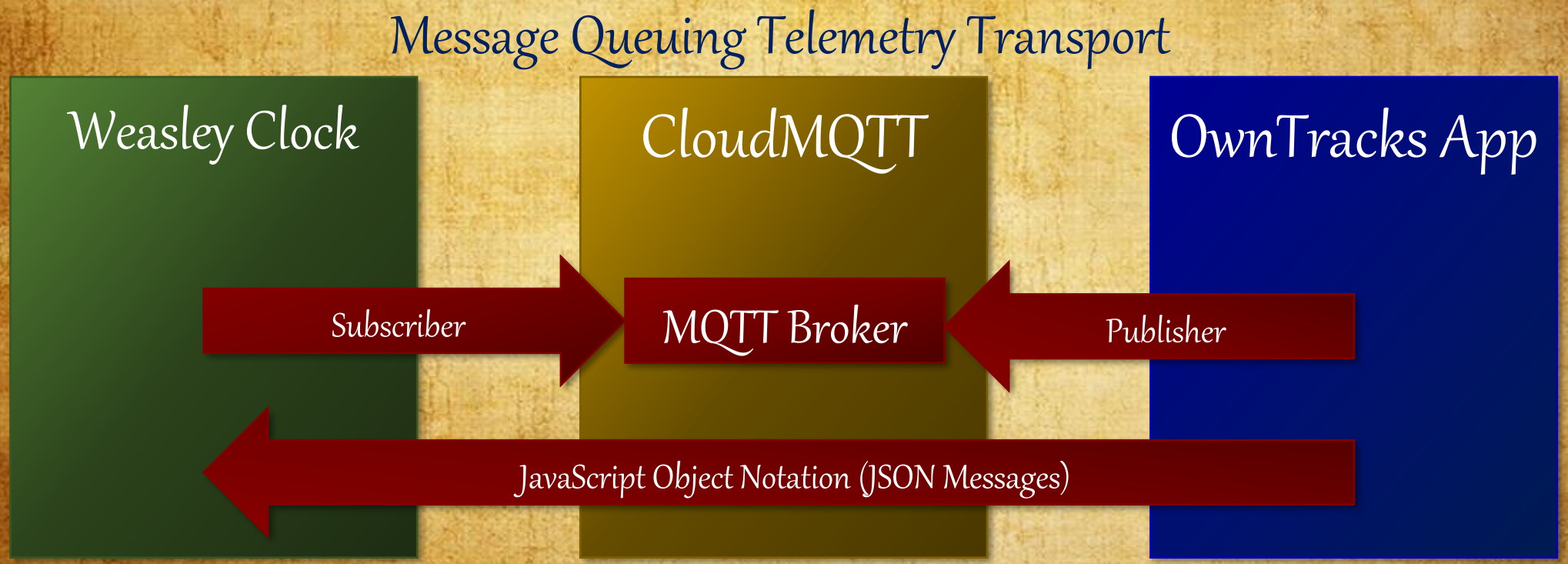
- "Weasley Clock" by jessyjones

  https://www.instructables.com/Weasley-Clock/

Objective

- Recreate the Weasley Family Clock for my own family
- Give each family member their own clock hand that moves based on their location (update based on their phone GPS)
- Keep telling time (include hands to indicate hour, minute, and second)
- Look like a real working grandfather clock (moving pendulum and weights)

Message Queuing Telemetry Transport



Overview

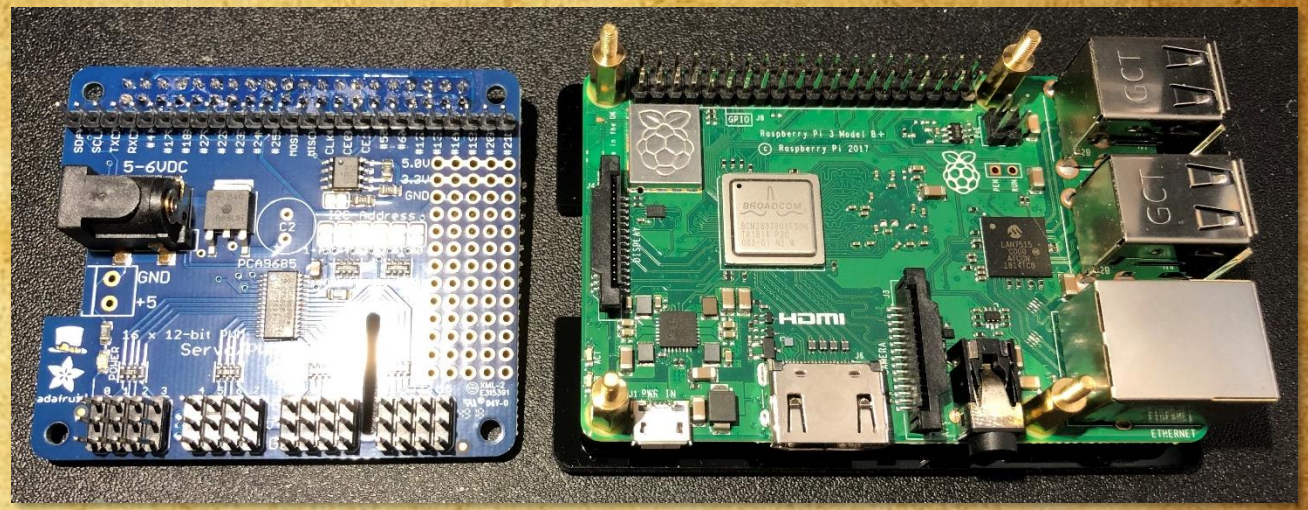
- Materials
- Artistic Design
- Mechanical Design
- Electrical Design
- Software Design
- Final Product
- "Plus It"

Materials

- Old Clock

Materials

- Old Clock
- Raspberry Pi w/ Servo Hat



Qty	Item
1	Raspberry Pi 3 B+
1	Ada-Fruit 16-Channel PWM Servo Hat
1	LabVIEW Community Edition

Materials

- Old Clock
- Raspberry Pi w/ Servo Hat
- Arduino Nano

Qty	Item
7	Arduino Nanos

Materials

- Old Clock
- Raspberry Pi w/ Servo Hat
- Arduino Nano
- Servos

Qty	Item
7	Parallax 360 High Speed, Continuous Rotation Servos
2	W5513-6T-JX 6 Turns 10kg Sail Winch Servo
2/4	MG996R Servo

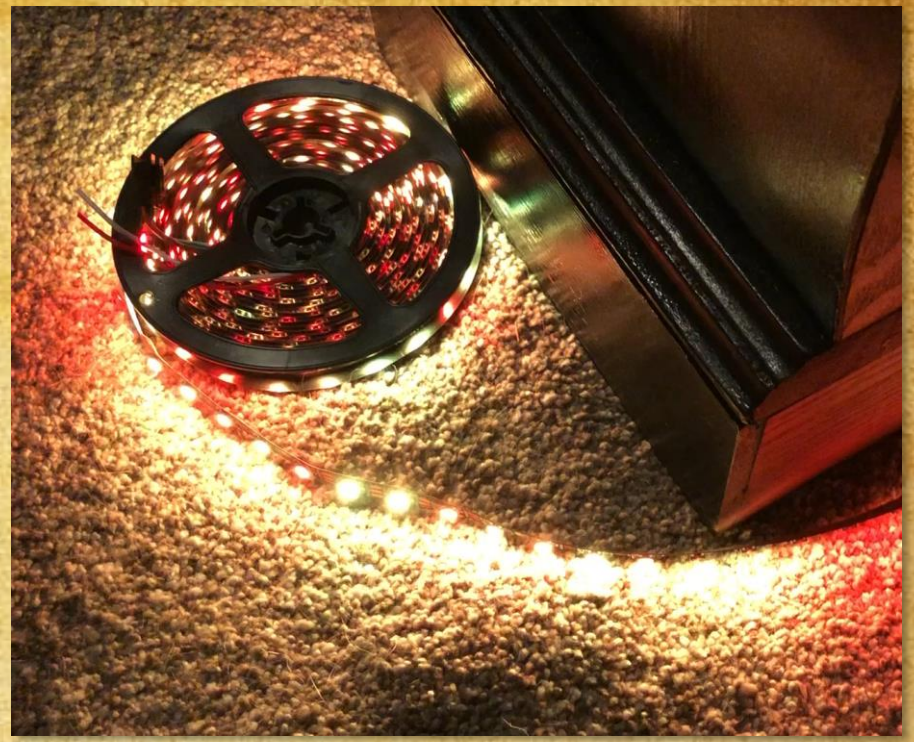
Materials

- Old Clock
- Raspberry Pi w/ Servo Hat
- Arduino Nano
- Servos
- Buttons

Qty	Item
5	WayinTop 12MM Momentary Push Button

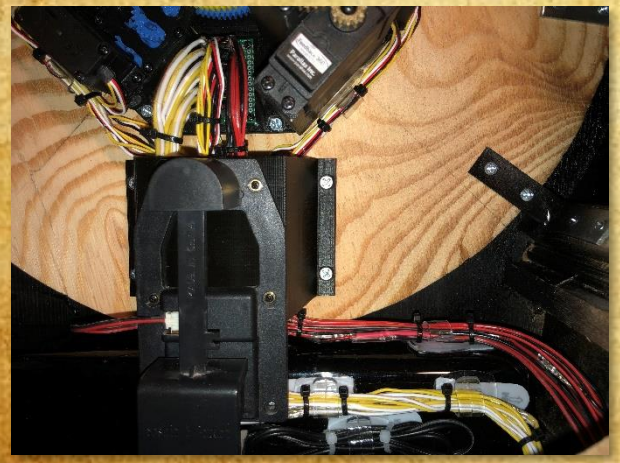
Materials

- Old Clock
- Raspberry Pi w/ Servo Hat
- Arduino Nano
- Servos
- Buttons
- Lights



Qty	Item
5m	WS2812b Strip Lights

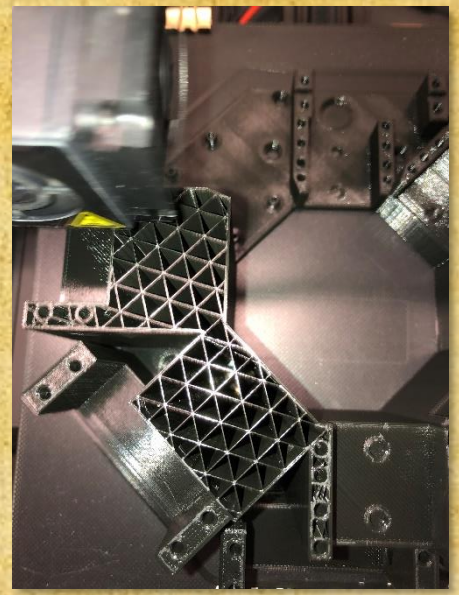
Tools

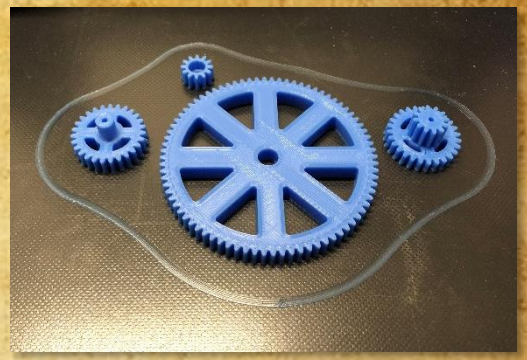


- ★Power Tools: Drill, Jig Saw, Disc Sander, etc.
- ★Soldering Iron (get one with temperature control)
- ★Wire strippers, pliers, screwdrivers
- ★3D Printer!!!

3D Printing

- My Ender 3 Pro was one of my most used tools
- Used Fusion 360 for modelling





Artistic Design

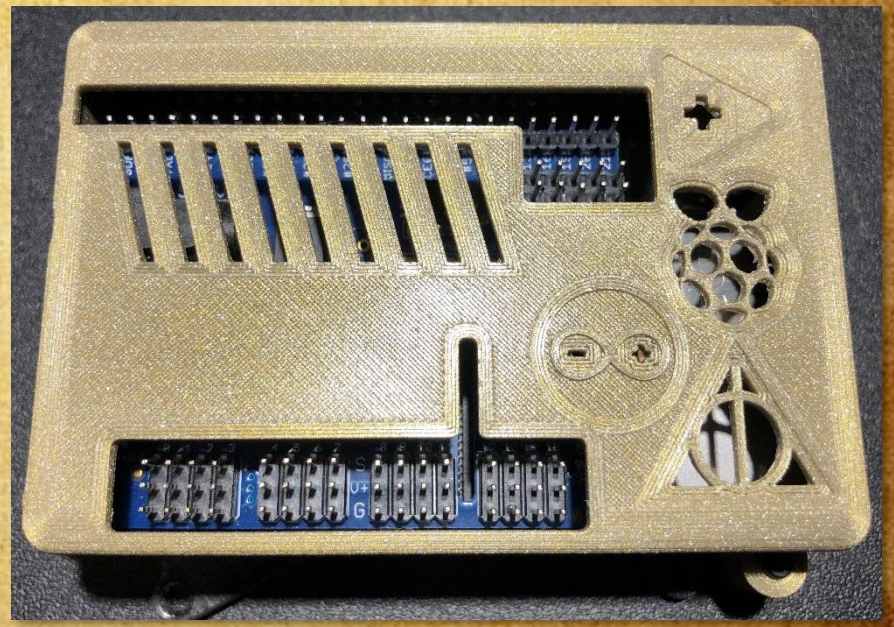
- Like movie, but make it my own
- Bring in Elements from Books and Movies
- Make it whimsical and reminiscent of the Wizarding World

Artistic Design

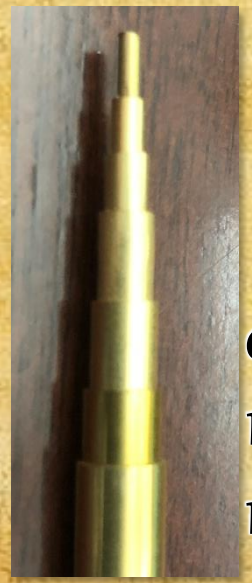
- Like movie, but make it my own
- Bring in Elements from Books and Movies
- Make it whimsical and reminiscent of the Wizarding World

Artistic Design

- Like movie, but make it my own
- Bring in Elements from Books and Movies
- Make it whimsical and reminiscent of the Wizarding World



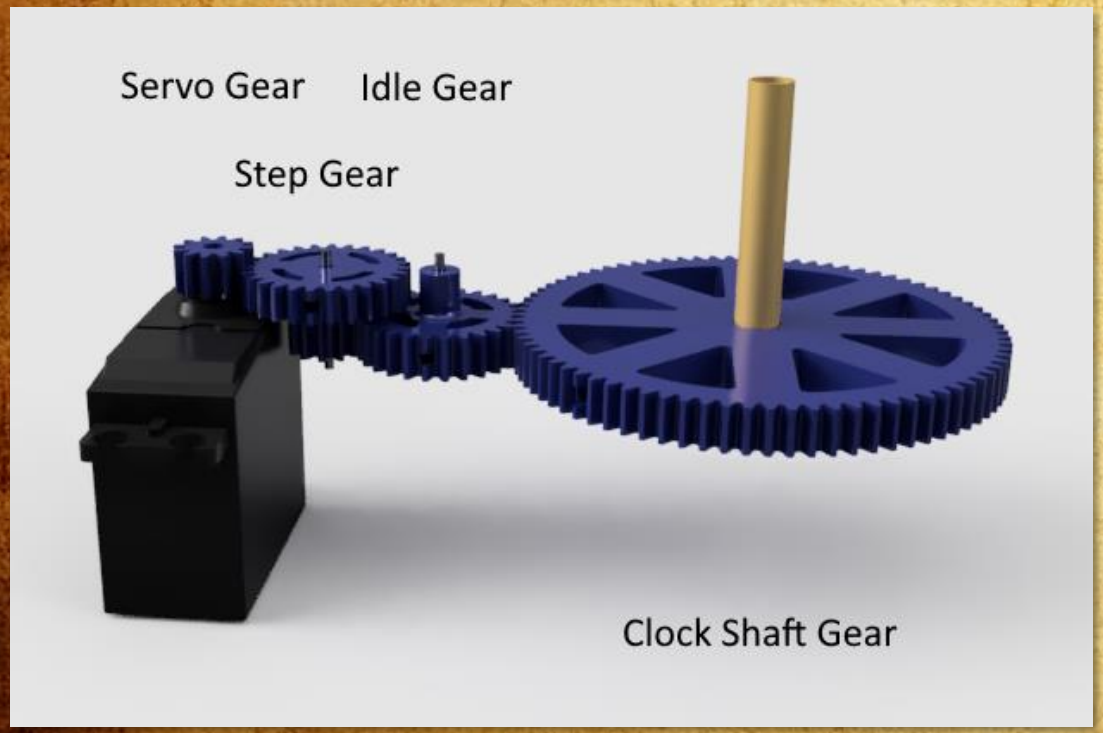
Mechanical Design



Second
Minute
Hour
Slytherin
Gryffindor
Hufflepuff
Ravenclaw

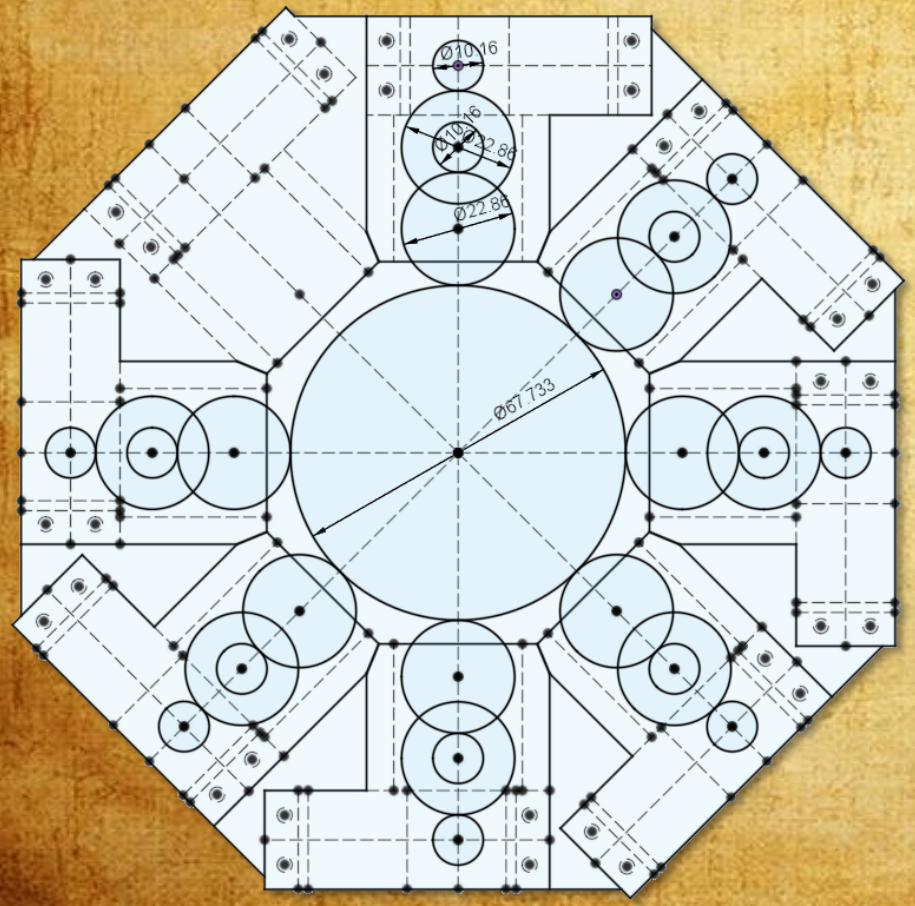
★7 Hands requires 7 servos and 7 nested shafts

Mechanical Design



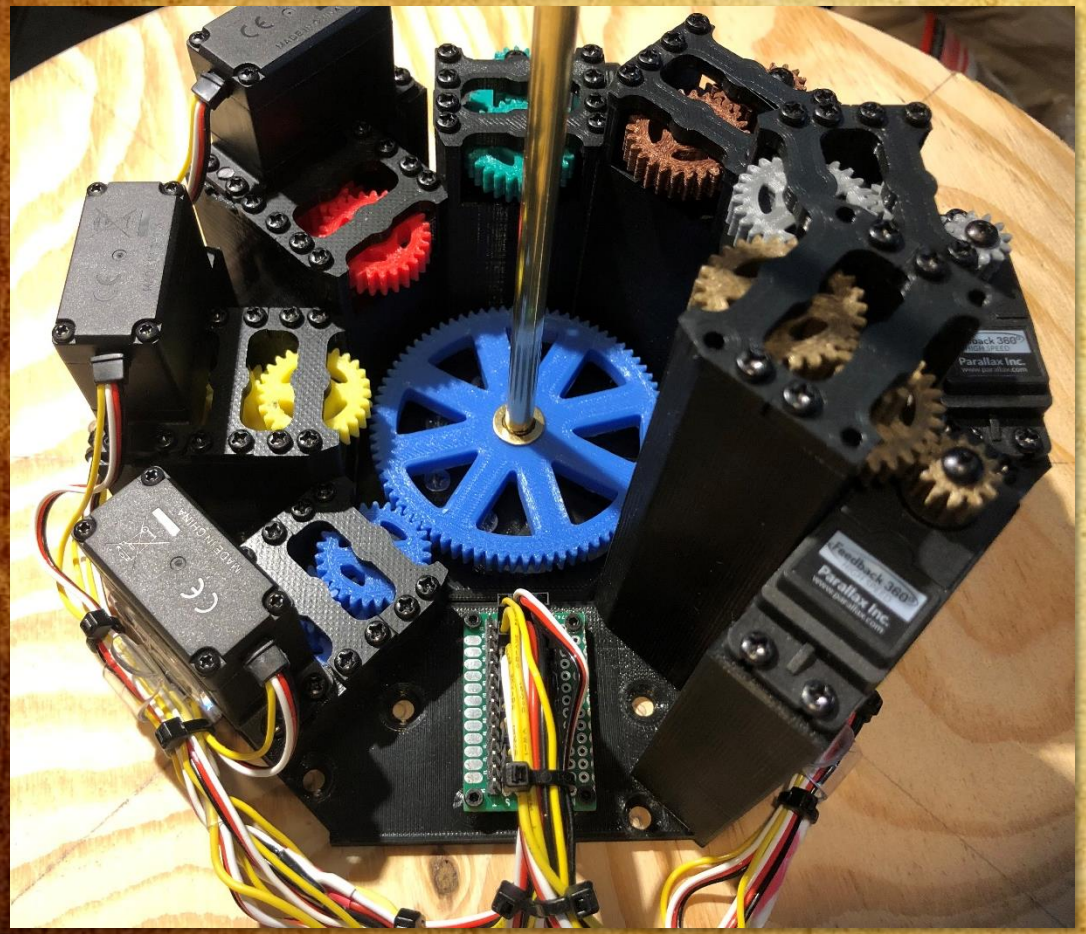
- 7 Hands requires 7 servos and 7 nested shafts
- Decided on a 15:1 Gear Ratio
  - 1° Clock = 15° Servo
  - 1 second = 6° Clock = 90° Servo
  - 360° on the clock = 5400° (15 rotations)

Mechanical Design



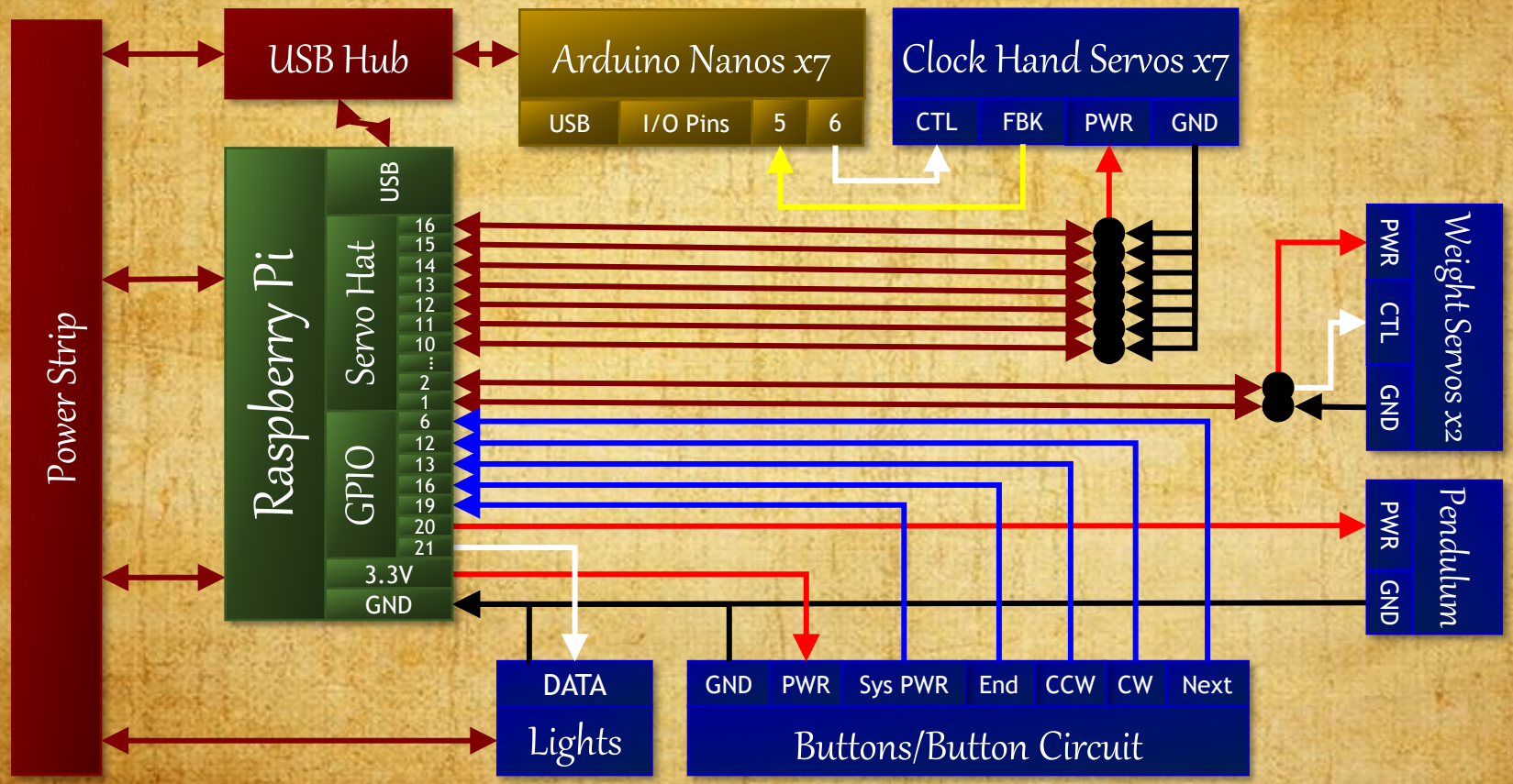
- ★7 Hands requires 7 servos and 7 nested shafts
- ★Decided on a 15:1 Gear Ratio
- ★Pack all Servos around Shafts

Mechanical Design



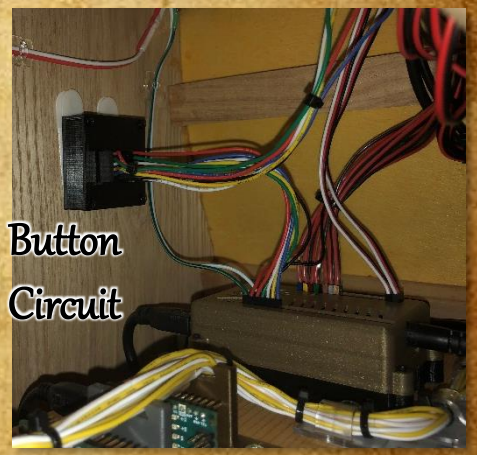
- ★7 Hands requires 7 servos and 7 nested shafts
- ★Decided on a 15:1 Gear Ratio
- ★Pack all Servos around Shafts
- ★Make it all fit behind the clock face

Electrical Design



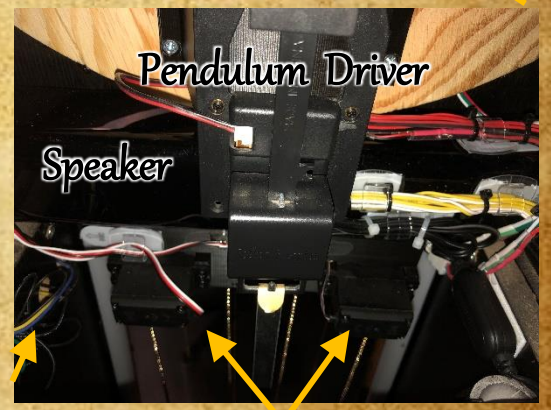
Electrical Design

- Base of Clock houses:
  - Raspberry Pi
  - Arduino Nanos
  - USB Hub
  - Button Circuit Board
  - Power Supplies

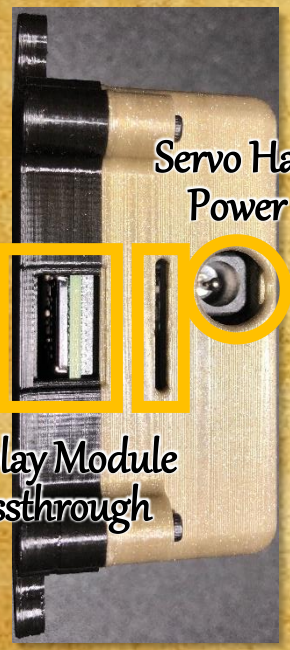


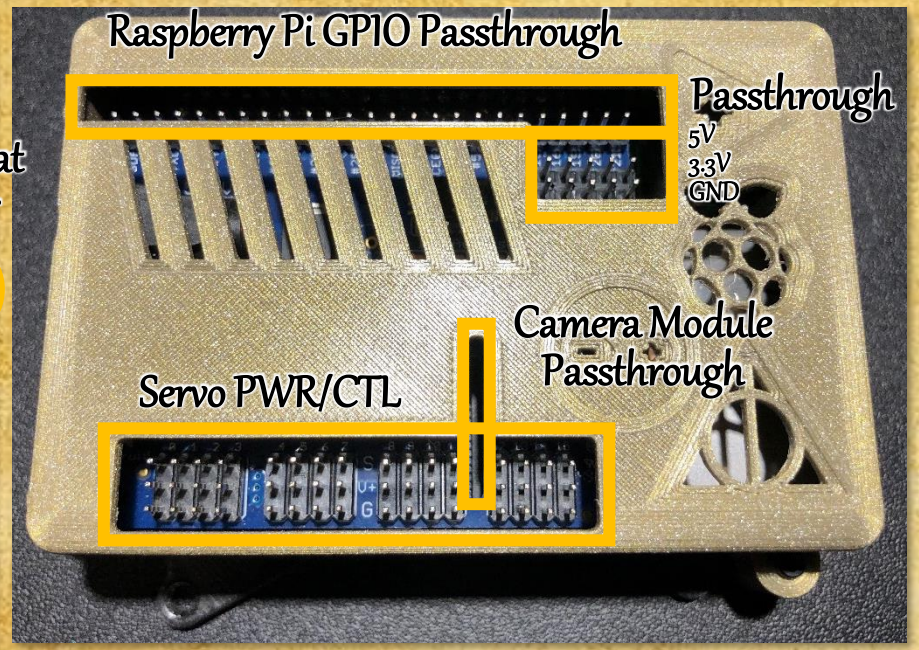
Electrical Design

- Top of Clock houses:
  - Servo Assembly
  - Pendulum Driver
  - Weight Servos
  - Speaker
  - Buttons

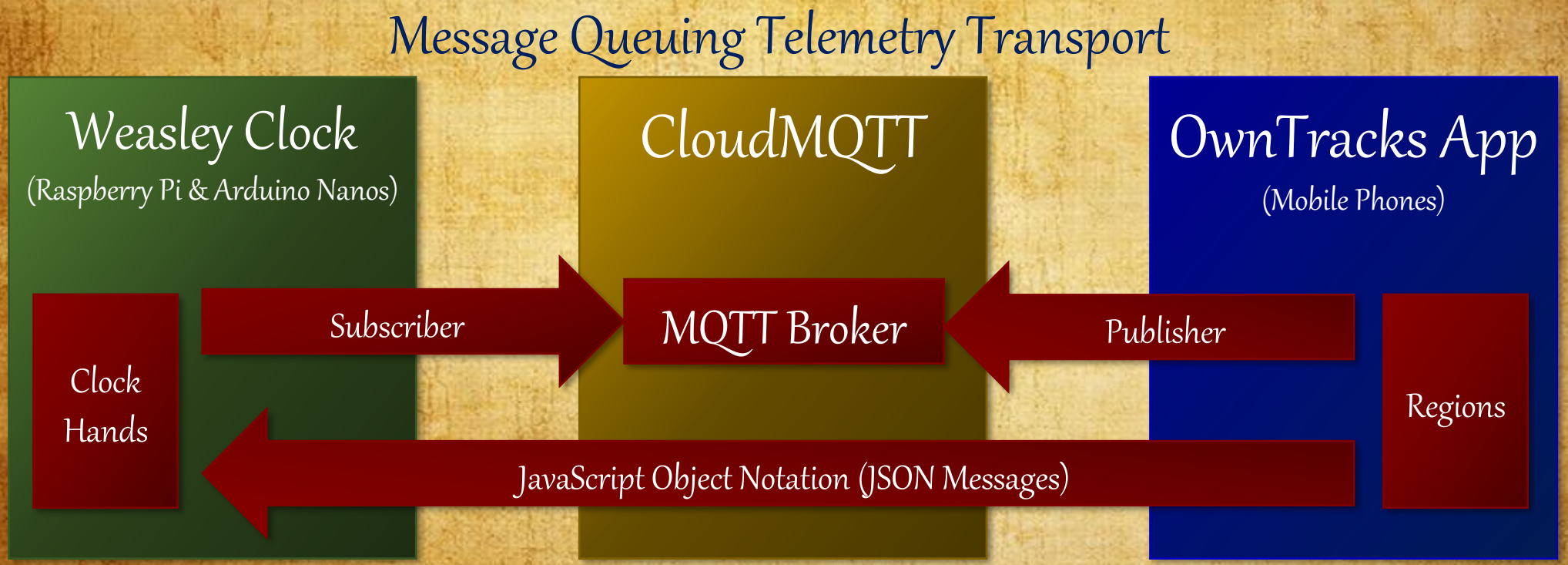


Raspberry Pi 3 B+ with Servo Hat





Software Design



Software Design

Software Components

Raspberry Pi

G (LabVIEW)

Main Controller

Receive MQTT Messages (TCP)

Control Servos through Hat

Interface with Python via Command Line

Interface with Arduino Nanos via Serial (USB)

Control Pendulum and Receive Button Presses via GPIO

Python

Control Sound

Control Lights

Get Timezone

Call Shutdown

Arduino Nanos

C

Control Servos

Read Feedback Line

Calculate Angle and Maintain Position

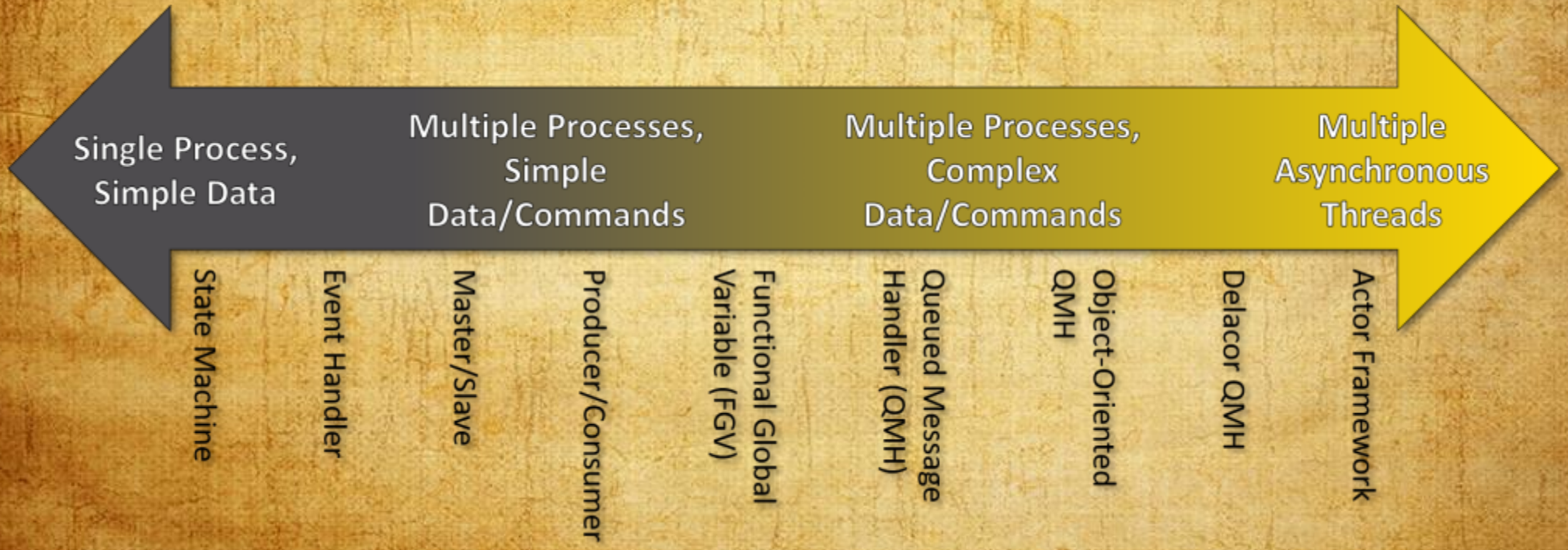
Software Components

G Classes and Interfaces

- Weasley Clock (Main Actor)
- Location Hand Controller
- Raspberry Pi I/O Controller
- Time Hand Controller
- LED Effects Controller
- MQTT Controller
- Calibration Controller
- Controller Interface
- Test Interface
- OwnTracks Message Parser
- MQTT
- Clock Hand Classes
- Servo
- LINX
- LINX PCA9865 PWM
- Raspberry Pi Utilities
- RPi-WS281x

Why Actor Framework?

★Multiple Asynchronous Modules with Communication between them



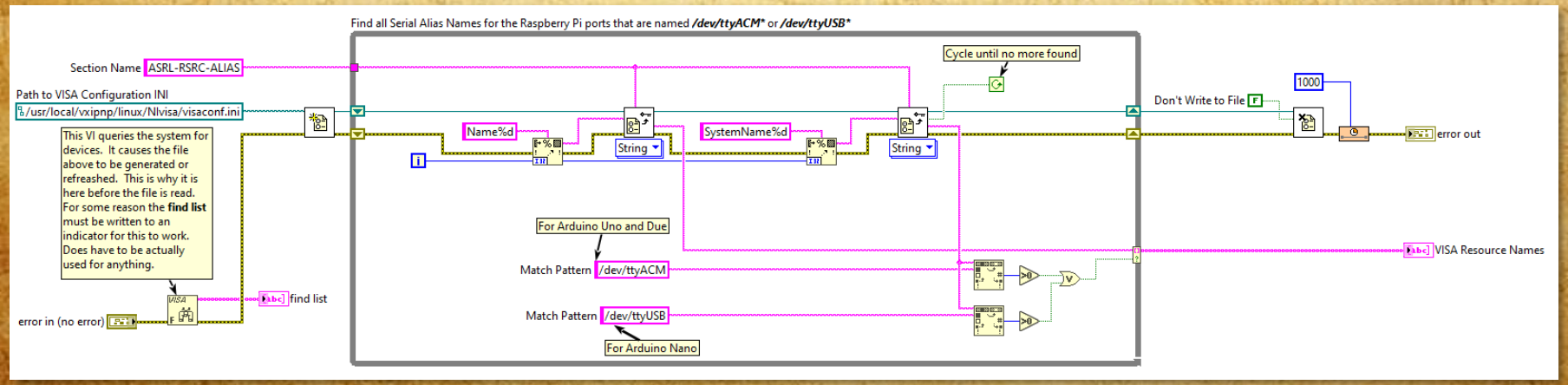
Finding the Arduino Nanos

- Problems:
  - The Arduinos wouldn't always be at the same VISA Resource Names
  - The name the Raspberry Pi assigns isn't the VISA Resource Name

Finding the Arduino Nanos

- Solution:
  - Run the *VISA Find Resource.vi*
  - Don't have to using anything from it but it causes a file to be generated on the Rasp Pi at:

    /usr/local/vxipnp/linux/NIvisa/visaconf.ini
  - Look in the file to correlate the System Name with the VISA Resource Name



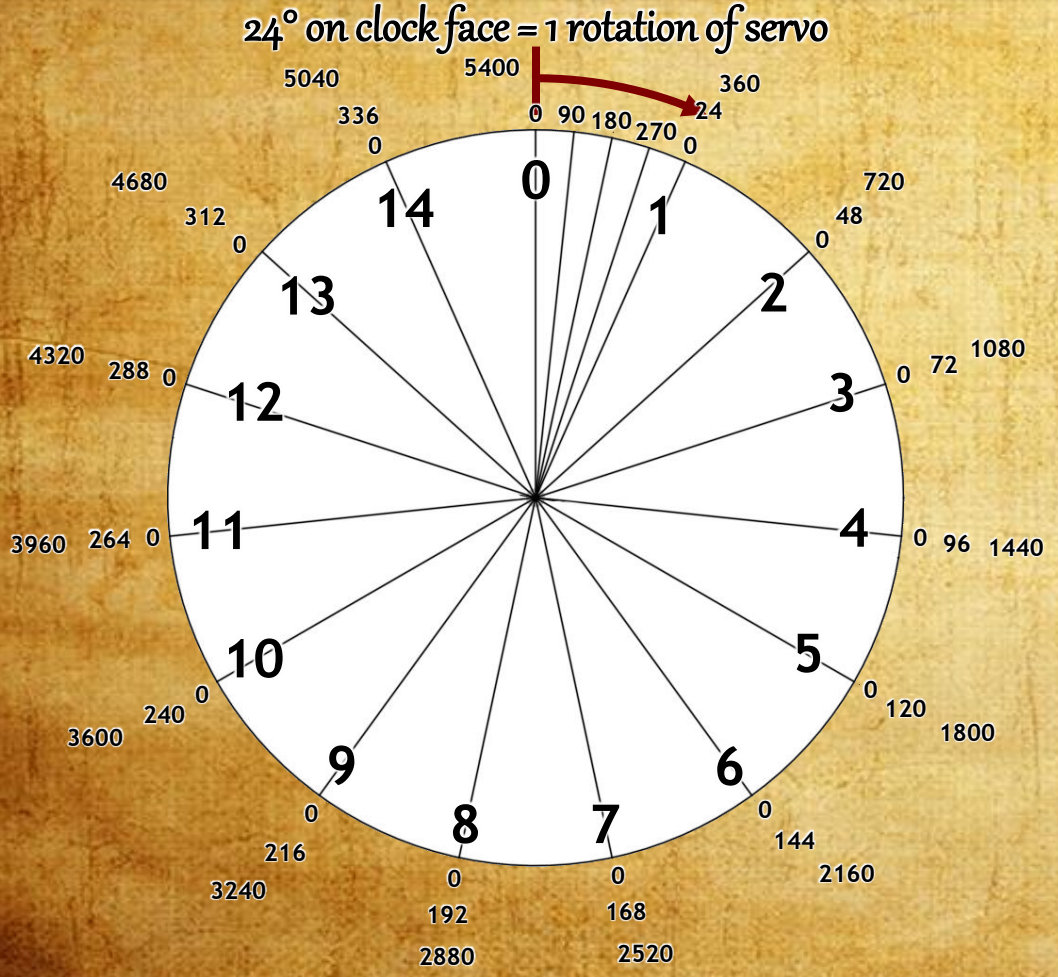
Finding the Arduino Nanos

- Solution:
  - Try to open the VISA Resource
    - Old resources could be in the file so ignore ones that fail
    - Ones that connect, send a command to the Arduino and see if it responds
  - I programmed the Arduino code to return a Servo ID letting me know which one it was

Controlling the Servos

- Via the Arduino Nanos
  - Serial communication set up between LabVIEW and Arduino Sketch
  - Send degrees on the clock face Arduino Sketch take care of conversion due to gear ratio.
- See Tim Robinson's VIWeek presentation:
  https://labviewwiki.org/wiki/VIWeek_2020/Proper_way_to_communicate_over_serial

Number of Turns Problem



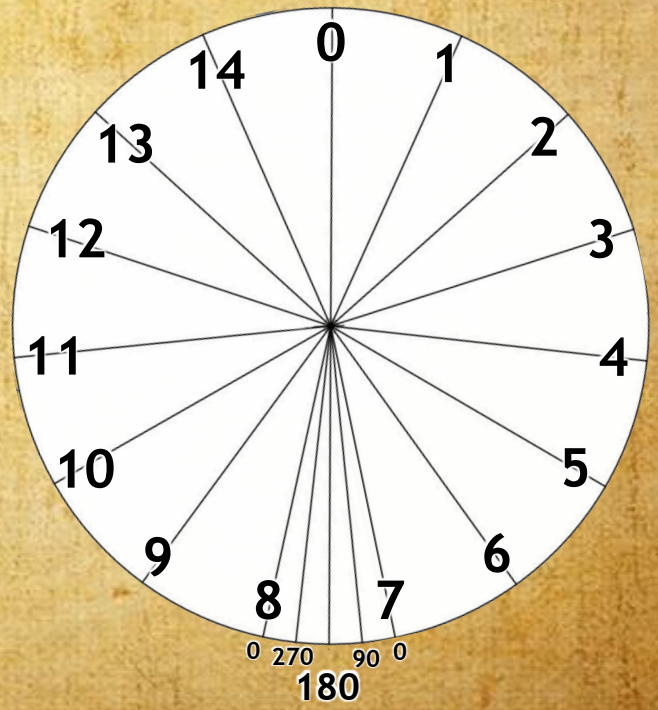
- What is the actual hand positions on startup?
  - The servo knows its position from 0° to 359°
  - Due to the gear ratio I have 15 "0°" positions
- I need to keep track of the number of rotations (turns) the servo has gone to know its clock face position.

Number of Turns Problem

- Tried Arduino Nano's EEPROM
  - 1024 bytes
  - 100k write limit per byte
  - at once every second 4 seconds = 4.63 days
- Write to file on Raspberry Pi
  - Never kept up, would always be off
  - Hard to reset, would have to remote into Rasp Pi to edit files manually

Calibration

- On startup put the hands in a known position
- Best place to set it is all hands pointing straight down
- Use buttons to "manually" command hands to move then tell the Arduino the turns at "7"



Controlling the Servos

★ Via the LINX PCA9865 PWM and the Servo Hat

★ Check out MediaMongrels' DemocracyBot:
https://www.mediamongrels.com/democracybot-rpi-linx-websockets-nxg-webvis/

SSH Trick

- LabVIEW runs in its own root (or chroot)
  - This is partially the reason you can't have UI on the Rasp Pi
  - LabVIEW Run-Time Engine runs as a pseudo-real time deployment
- Like to running computers: Main Root & LabVIEW Root
  - To run code in the main root you have to create a "remote-local" connection
  - https://labviewwiki.org/wiki/Chroot_SSH_trick

Time and Time Zones

- Raspberry Pi does not have its own clock
  - It has to sync with the internet to get correct time
  - Time zone is set in the setting to calculate offset from UTC
- LabVIEW time VIs will output only UTC
  - Manually correct for time zone
  - Python call to get time zone offset from the Rasp Pi

Playing Sound

- Sound VIs don't work on the Raspberry Pi
- I use a Python script to play sound instead
  - Uses the SSH Trick
  - Uses the Pygame Library

Controlling the Lights

- SSH Trick
- Three Options:
  - Through PWM (GPIO18 - conflicts with audio)
  - Through PCM (GPIO21)
  - Through SPI (GPIO10 – more setup)

Stopping the Lights

- Shared Array – used as a single element array
- Python code running lights in a loop check for state of Shared Array
  - 1 = effect running
  - 0 = stop effect

Final Product

"Around here, however, we don't look backwards for very long. We keep moving forward, opening up new doors and doing new things, because we're curious ... and curiosity keeps leading us down new paths."

--Walt Disney

Plus It – Version 2.0

- Add Interactive Spells
  - Wand from Universal Studios Wizarding World Theme Park (Passive IR Reflector)
  - Raspberry Pi NoIR Camera Module
  - Spell Recognition via Optical Character Recognition
  - Animate spell effects (i.e. lights, move weight/hands, stop pendulum)

Plus It – Version 3.0

- Add Interaction with Home Lights
  - Replace some lights with Red-Green-Blue-Warm White (RGBWW) LED Strips
  - Replace some lights with Philips Hue or equivalent
  - Control strips using Arduino Nano 33 IoT

Plus 11 – Version 4.0

- Add "Olivanders" Wand Holder
  - Get more IR reflective wands
  - Create a holder with switches
  - Create wand selection effects

Any sufficiently advanced technology is indistinguishable from magic.

--Clarke's Third Law, Arthur C. Clarke

Questions, Incantations, or Hexes