FISHMO
Final Presentation

Capstone 2015
McMaster University

Who are we?

- Rhett Amin - 4th Year Honours Comp. Sci.
- Brandon Da Silva - 4th Year Honours B.I.
- Daniel Esteves - 4th Year Honours B.I.
- Michael Liut - 4th Year Honours Comp. Sci.

The Problem

- Providing care and monitoring the species inside the aquarium in an effective manner
- Methods for effective care becomes a time consuming, repetitive, and tedious.
- Fish may not receive the best treatment most of the time.

Why FISHMO?

- FISHMO is something that we are passionate about.
- We were all instantly drawn to the idea of FISHMO.
- FISHMO allowed us to integrate our favourite courses and external interests in Computer Science into a final project.

Hardware

- Physical Components
  - LED’s
  - Heater,
  - Air Pump
  - Temperature Sensor
  - Water Flow Metre
  - USB Camera

Hardware - Programming

```
global tick
global starttime
global speed
global speed1
global counter
global constantOfWater

starttime = time.time()
pulseFreq = 7.5 #sensor frequency in Hz
counter = 0


def my_callback(channel):
    print ("falling edge")
    global counter
    counter += 1
    print ("counter: ", counter)
    global starttime
    print(time.time() - starttime, "seconds")


GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.IN)
GPIO.setup(17, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.add_event_detect(17, GPIO.FALLING, callback=my_callback, bouncetime=100)

def signal_handler(signal, frame):
    print("Ctrl+C detected...")
    GPIO.cleanup()
    sys.exit(0)

signal.signal(signal.SIGINT, signal_handler)


while 1:
    time.sleep(0.05)
    if ((time.time()-starttime) > 5.00):
        speed = (counter/(10*pulseFreq))
        speedOne = (speed*60)
        print ""
        print("speed: ", speed, "L/Sec")
        print "or"
        print("speed: ", speedOne, "L/Min")

        # Write Speed Data to File - in Litres per Minute -- seperated by line
        file.write("Speed: " + str(speedOne) + " L/Min" + "\n")
        print ""
        starttime = time.time()
        counter = 0
```

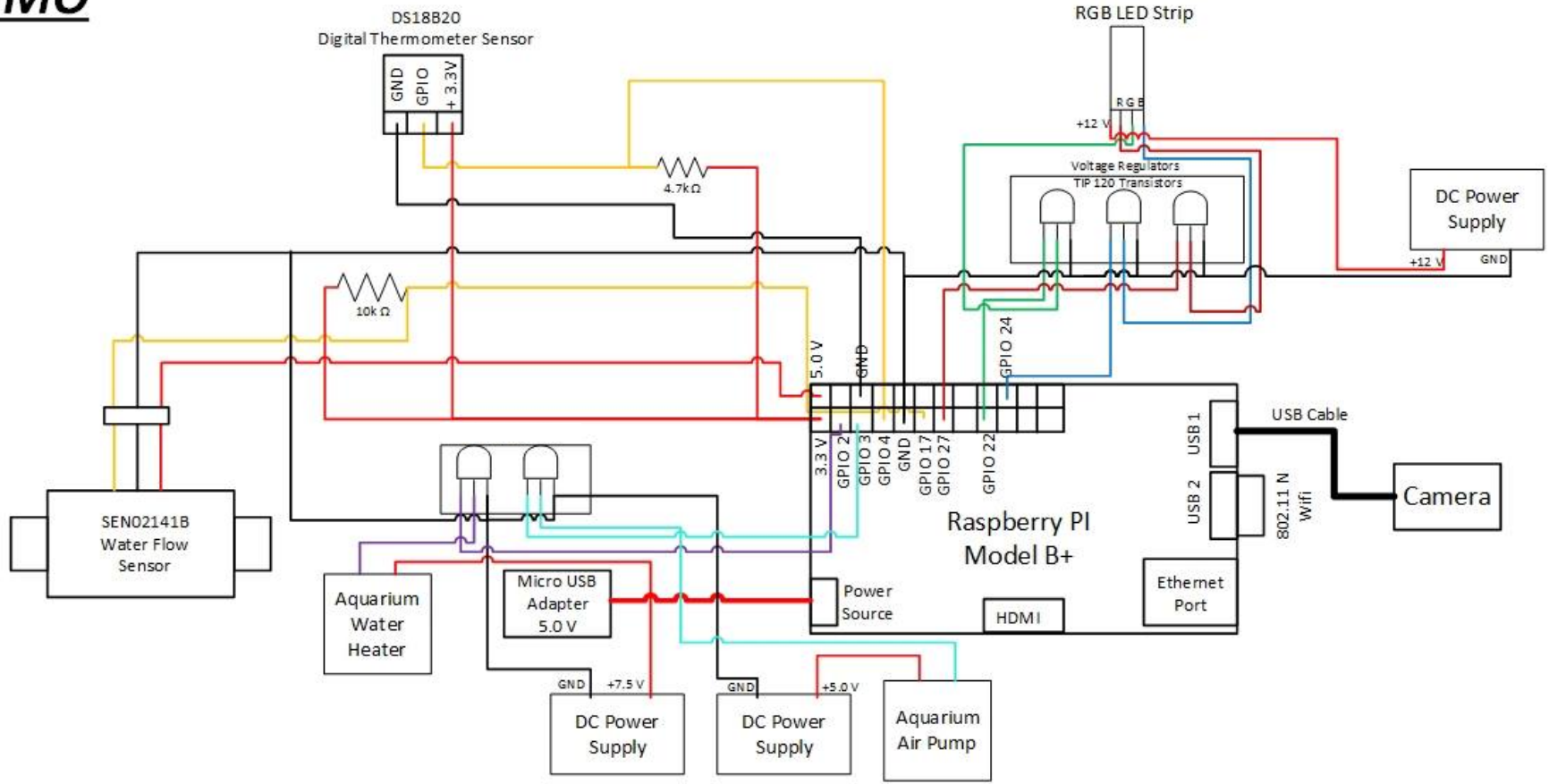
All of the Raspberry Pi's coding was written in Python.

Installation and utilization of unique libraries for the Pi components and peripherals.

Hardware - Wiring

- All hardware was soldered in ITB IEEE lab
- All peripheral devices moved to standard 4x circuit board form a solderless board after finalization.
- TIP Transistors, Resistors, Switches and low power voltage convertor.
  - (TIP-120 Transistors)
  - (4.7k Ω, 10k Ω)

FISHMO


DS18B20
Digital Thermometer Sensor
GND
GPIO
+ 3.3V
4.7kΩ
10kΩ
RGB LED Strip
R G B
+12 V
Voltage Regulators
TIP 120 Transistors
DC Power Supply
+12 V
GND
5.0 V
GND
GPIO 24
3.3 V
GPIO 2
GPIO 3
GPIO 4
GND
GPIO 17
GPIO 27
GPIO 22
USB 1
USB 2
USB Cable
802.11 N Wifi
Camera
Raspberry PI Model B+
Ethernet Port
HDMI
Power Source
SEN02141B Water Flow Sensor
Aquarium Water Heater
Micro USB Adapter 5.0 V
GND
+7.5 V
DC Power Supply
GND
+5.0 V
DC Power Supply
Aquarium Air Pump

Challenges

Frying the Pi

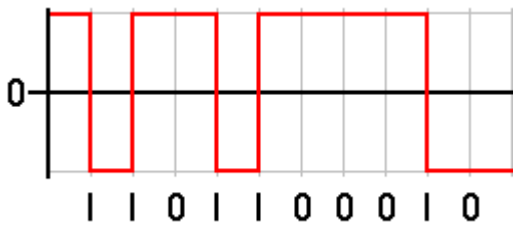
-Using correct regulators

Power Shortages

-Heater(5V), LED’s,(12V),Air Pump(5V),Temp-sensor(3.3V), Flow-Sensor(5V)

Converting Values to be displayed

-Signal edge graph



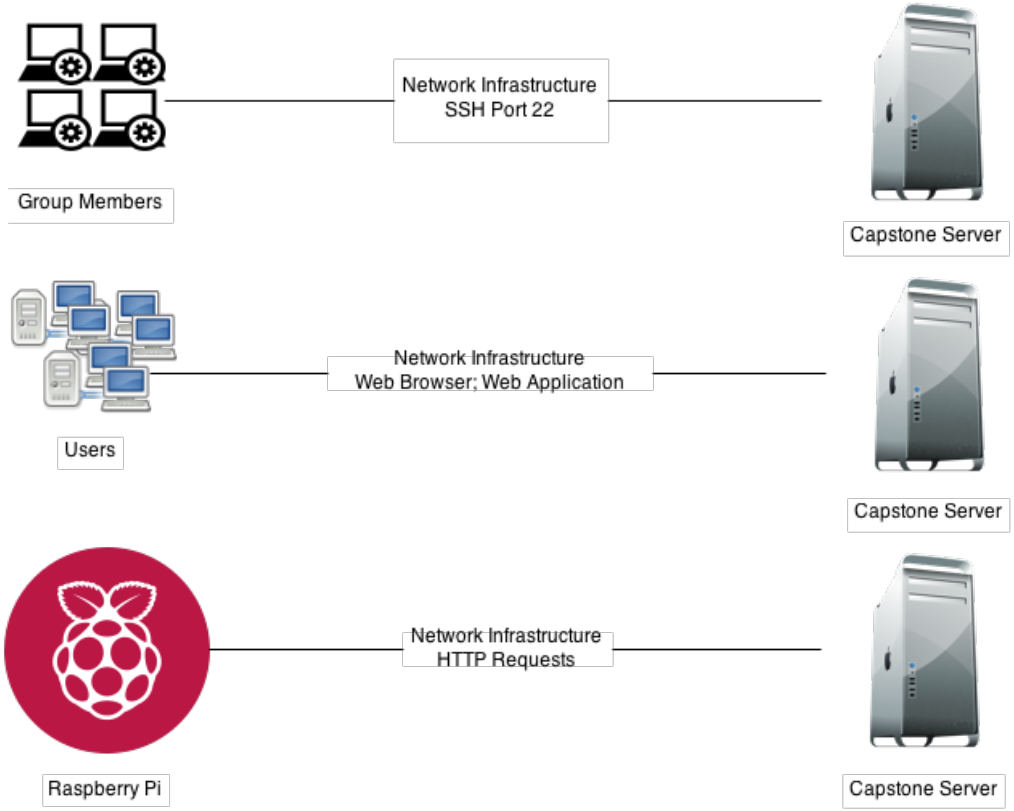
Backend

Network

Airport Extreme

Capstone Server

Backend


Network Infrastructure
SSH Port 22
Group Members
Capstone Server
Network Infrastructure
Web Browser; Web Application
Users
Capstone Server
Network Infrastructure
HTTP Requests
Raspberry Pi
Capstone Server

Challenges

- Keeping the server up and running consistently and stable
  - Dealing with errors, timeouts, sleep
- User permissions
- Crashing

Software

- The Web app is the control interface between the user and the Raspberry Pi.
- Provides the user the ability to update the way the Pi controls the tank.
- Built using HTML, CSS, JS, and PHP
  - The Materializecss framework was used to assist with the CSS styling.

Security

- We ensure our servers and the users data is secure by various methods:
  - We redirect users if they try to access any page without logging in.
  - We prevent SQL injection attacks via php script.
  - Encrypted the server via 256bit - AES
  - Hash User details as we push to the server

Challenges

- Design Challenges
  - Layout
  - Colour Scheme
  - Visually Appealing
  - Providing feedback on changes

Integration

- Connecting the back-end to the software (i.e. the website)

- Connecting the hardware to the web application

Video

Challenges

- Integration
  - Connecting Pi and Server
    - SSL Handshake Protocol issues (remote connections)
  - Sending data from Pi to the Server

Registration Process

- There are a set list of tanks in the database for potential users to register.
- The users specify their FISHMO ID from the FISHMO unit when registering.
  - This links their user account with their specific hardware unit.

Web App Changes

- Settings changed on the web application send updates to the database via Ajax, PHP and MySQL and via PHP SSH to the hardware
- Values updated on the FISHMO unit updates the database via HTTP POST

DEMO