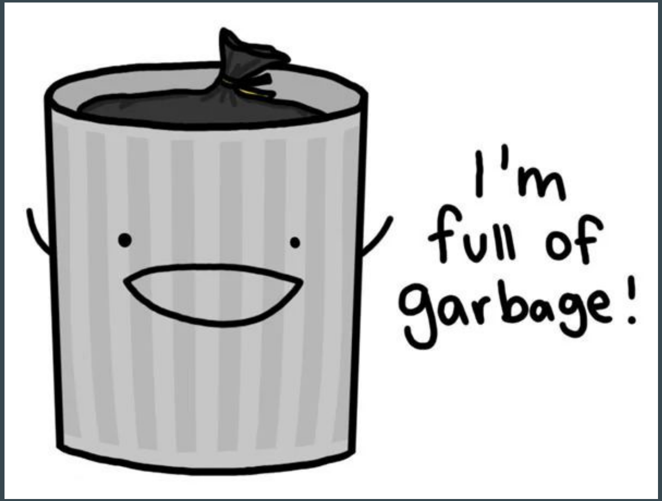
ECE 445 Spring 2017
Autonomous Trash Can

•••

Group #85: Eshwar Cheekati, Michael Gao, Aditya Sule

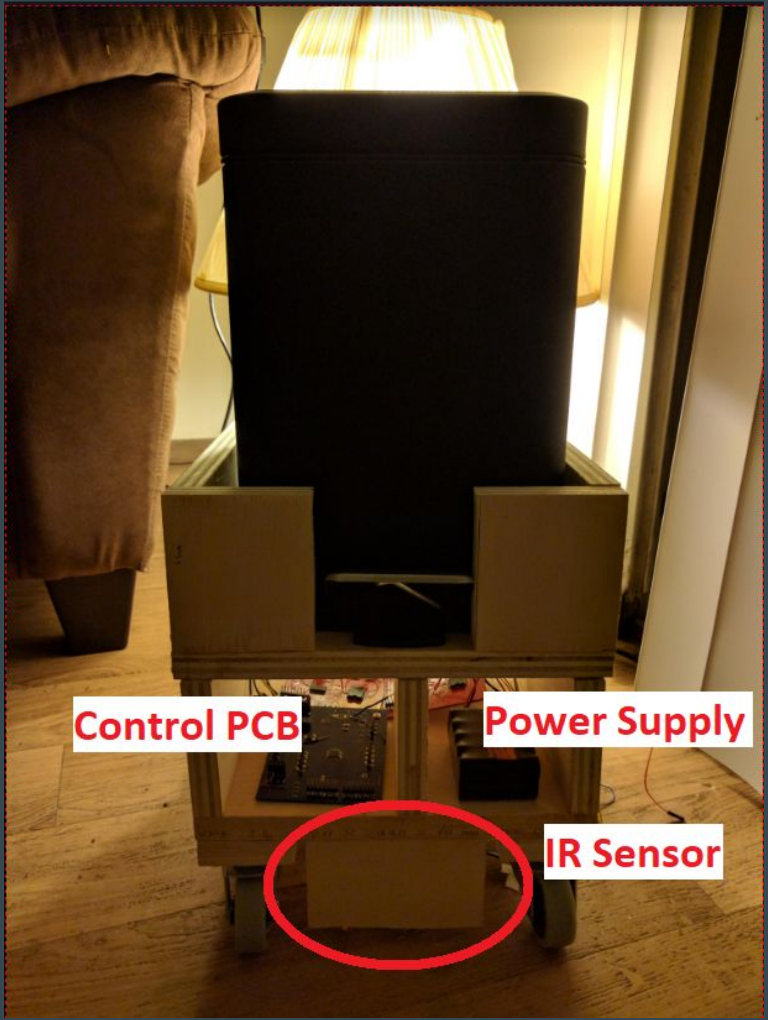
Introduction

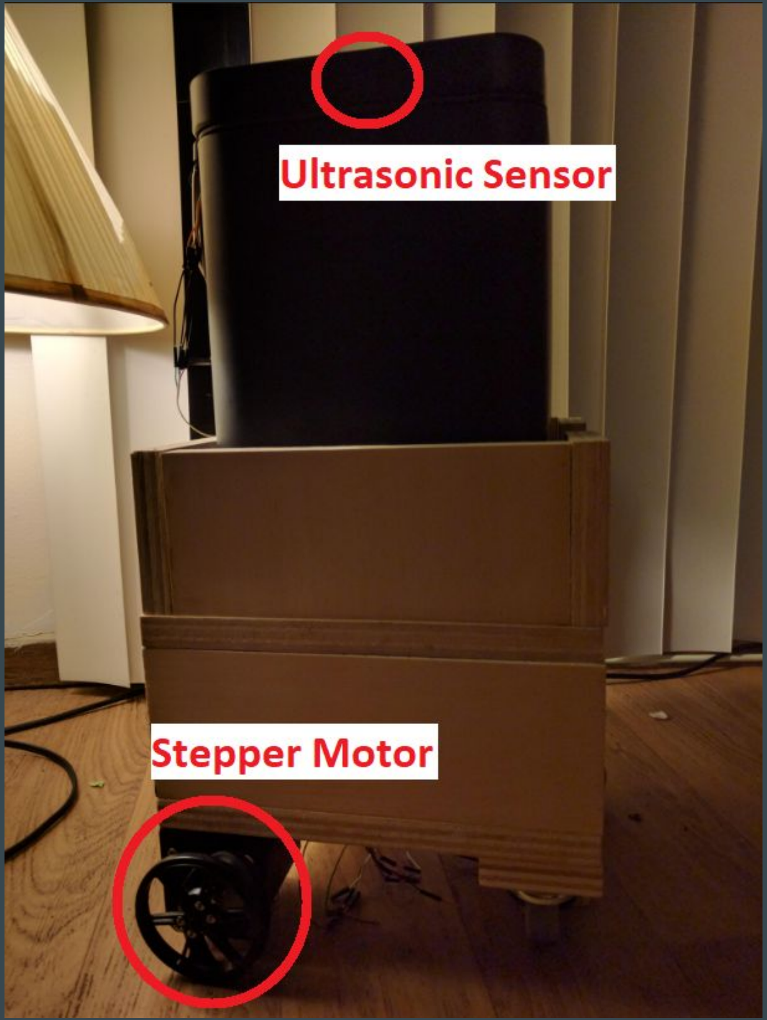
- High amount of waste generated
- Poor communication/trash management -> smelly odors
- Need for reminder to take trash out



Objectives

- Two aspects - Detection and Navigation
- Detection focuses on accurate reading of when trash is full
- Navigation focuses on moving trash can to the door


Control PCB
Power Supply
IR Sensor

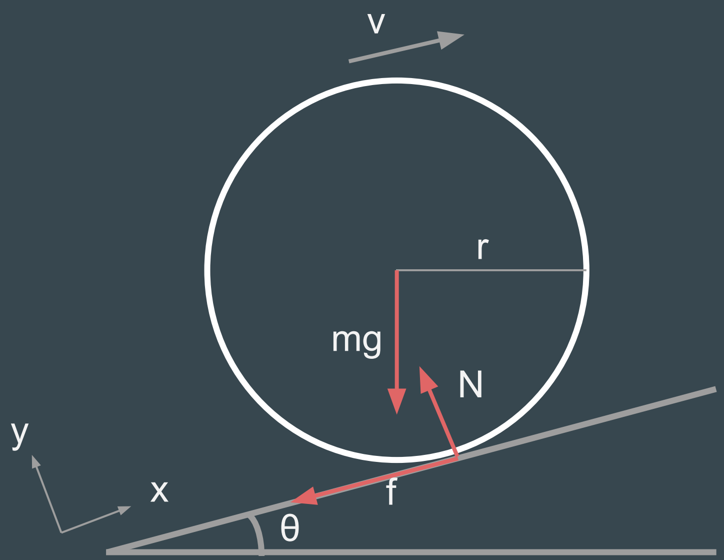

Ultrasonic Sensor
Stepper Motor

Power Calculations

- Important so we can select the following parts
    - Battery
    - Motor
- Required figures
    - What is the torque required to turn the wheels?
    - What is the power required to produce that torque?

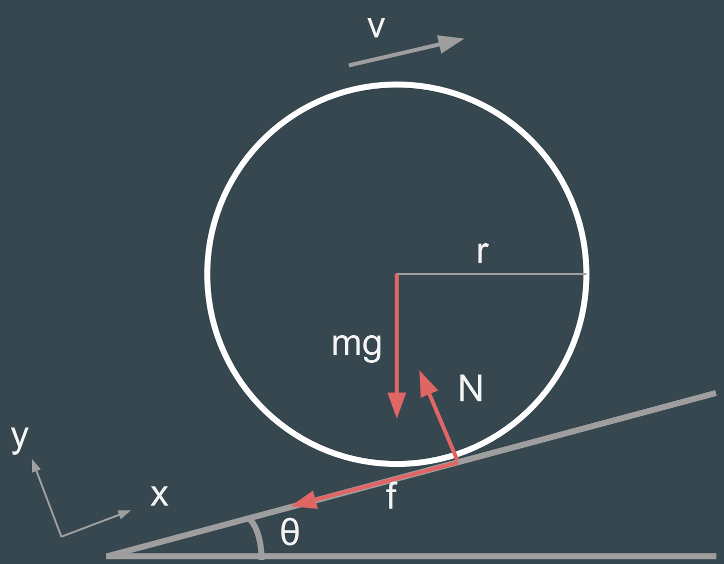
Power Calculations - Basic theory

- Simple system
- One wheel on a incline
- Forces acting on wheel
  - Friction (f)
  - Gravity (mg)
  - Normal (N)



Power Calculations - Assumptions

- Mass $m = 10kg$
- Coefficient of friction $\mu = 0.6$
- Incline $\theta = 3°$
- Radius $r = 5$ cm
- Efficiency $E = 60\%$
- Velocity $v = 5cm/s$



Power Calculations - Working it out

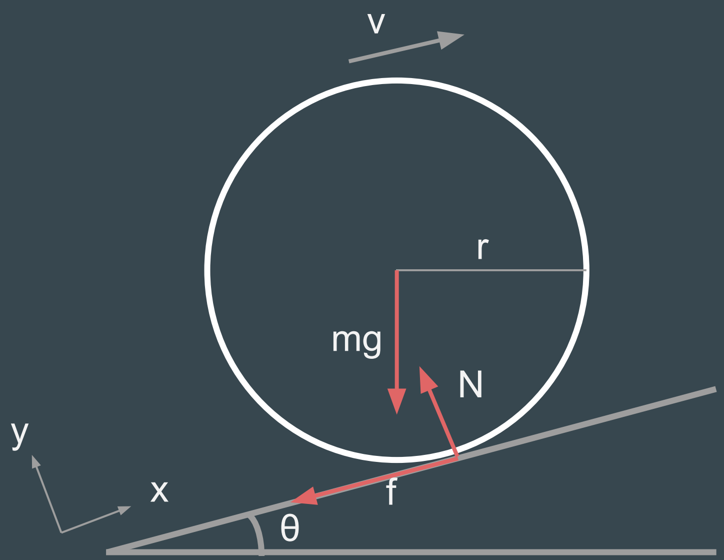
$$f + mg\sin(\theta) = F \quad (1)$$

$$N = mg\cos(\theta) \quad (2)$$

$$f = \mu N \quad (3)$$

$$\tau = Fr = mg\left(\sin(\theta) + \mu\cos(\theta)\right) r$$

$$\tau = 3.19\ \text{Nm}$$



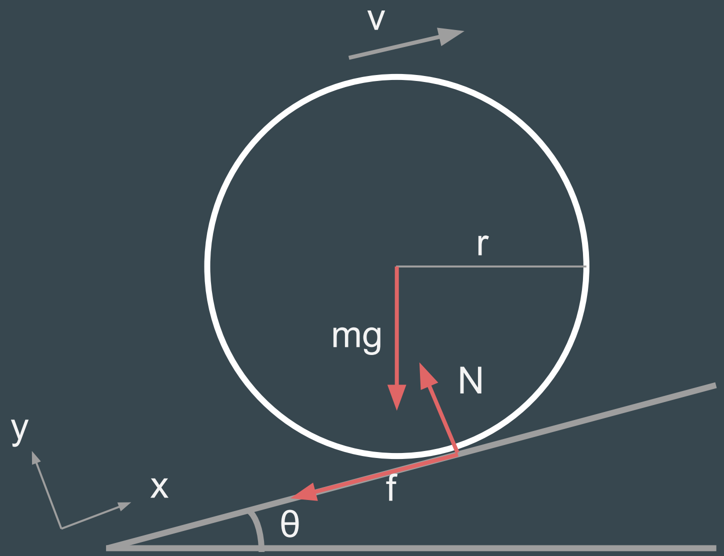
Power Calculations - Working it out

$P = \tau\omega/E \quad (1)$

$\omega = v/r \quad (2)$

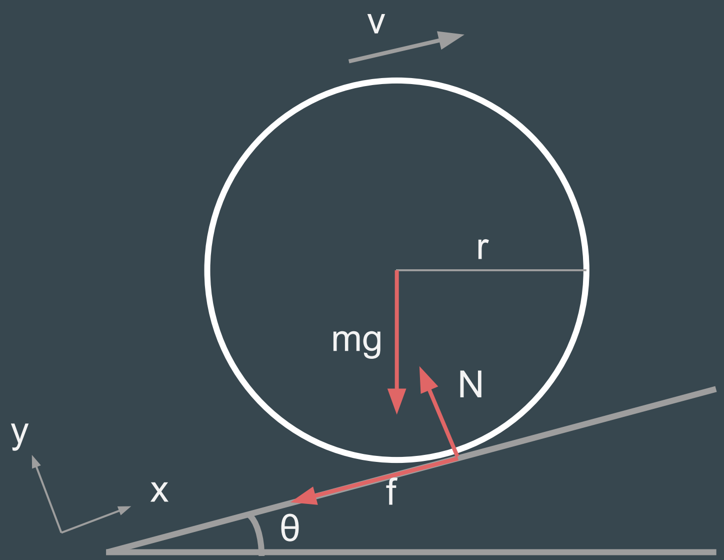
$\mathbf{P = \tau v/Er}$

P = 5.31 W



Power Calculations - Results

- We need about 5.5W to drive the motors
- We need a torque of 3.2Nm to able to keep in *steady* motion
- 1.6Nm per motor

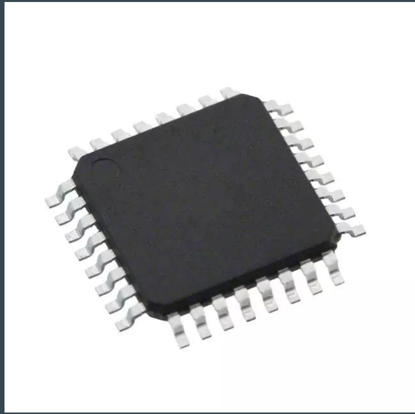


Power Supply

- 12 V, 2000mAh NiMH battery
- Rechargeable
- Safer than Li-ion batteries

Microcontroller - Atmega328pb-au

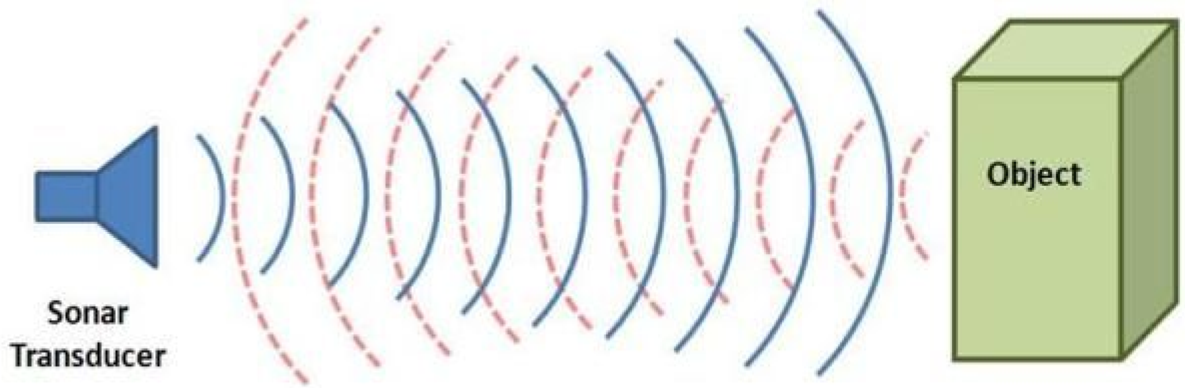
- 24 digital IO pins
- 16 MHz clock frequency
- Compatible with Arduino



Trash Level Detection:

- Weight vs Volume
- Ranging sensors
- HC-SR04 ultrasonic sensor

Ultrasonic Sensors



Basic sonar illustration – a transducer generates a sound pulse and then listens for the echo.

Trash Level Distance Calculation:

$$\text{Distance (cm)} = (T/2) \times (1\ \text{second}/34300\ \text{cm})$$

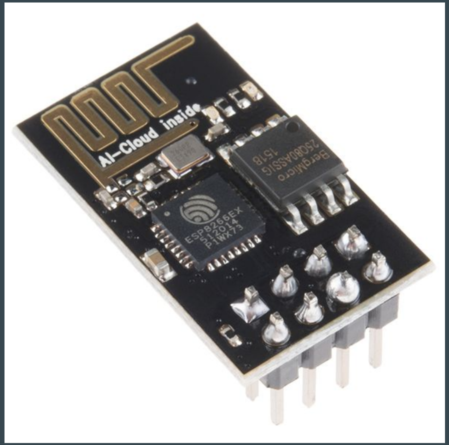
- T = Time between sending wave from trig pin to receiving wave from echo
- Trash level distance reading taken every 50ms

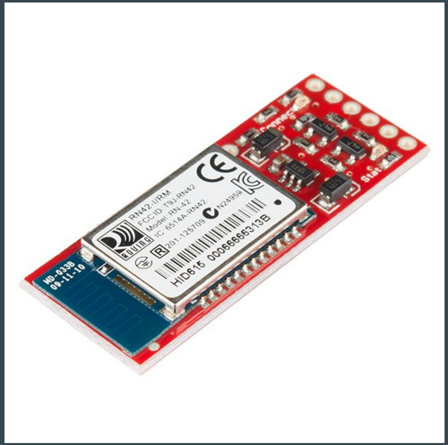
Trash Level Percentage Calculation:

- D = average of 50 ultrasonic sensor measurements
- Trash Percentage fullness: (24 - D)/24 x 100 %
- 24 cm is the height of the trash can

Data Collection

- Wifi vs Bluetooth





ESP8266

- Low cost
- Full TCP/IP Stack
- Substantial online documentation

Web Server:

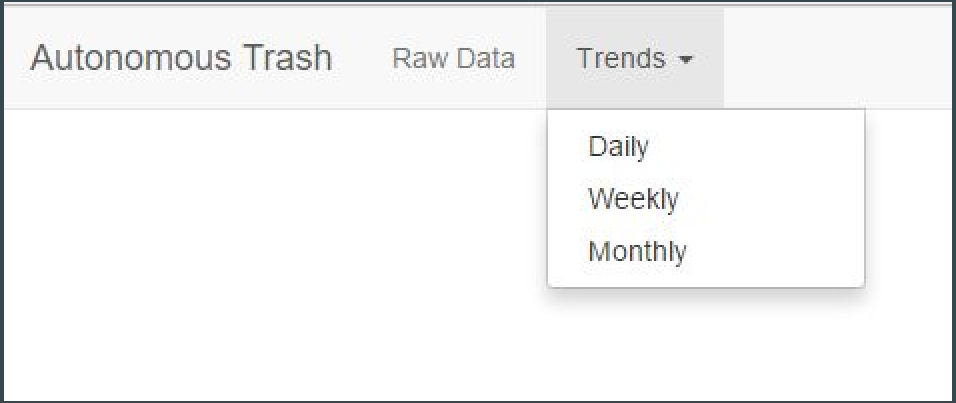
HTTP Request:

- POST: Sent to my web server URL with trash level attached as url-encoded value
- GET: Sent to my web server URL and returns all data samples in JSON


```
{"_id":"58ff00e7c42ff47e14b5f0e9","trash_level":"10","__v":0,"time_stamp":"2017-04-25T07:55:19.542Z"},
{"_id":"58ff00ecc42ff47e14b5f0ea","trash_level":"90","__v":0,"time_stamp":"2017-04-25T07:55:24.488Z"},
{"_id":"58ff00ecc42ff47e14b5f0eb","trash_level":"29","__v":0,"time_stamp":"2017-04-25T07:55:24.497Z"},
{"_id":"58ff00f1c42ff47e14b5f0ec","trash_level":"13","__v":0,"time_stamp":"2017-04-25T07:55:29.486Z"},
{"_id":"58ff00f1c42ff47e14b5f0ed","trash_level":"3","__v":0,"time_stamp":"2017-04-25T07:55:29.497Z"},
{"_id":"58ff00f6c42ff47e14b5f0ee","trash_level":"44","__v":0,"time_stamp":"2017-04-25T07:55:34.486Z"},
{"_id":"58ff00f6c42ff47e14b5f0ef","trash_level":"97","__v":0,"time_stamp":"2017-04-25T07:55:34.493Z"},
```

Web Server:

- Collected data from trash level readings and displayed results in daily, weekly, and monthly views

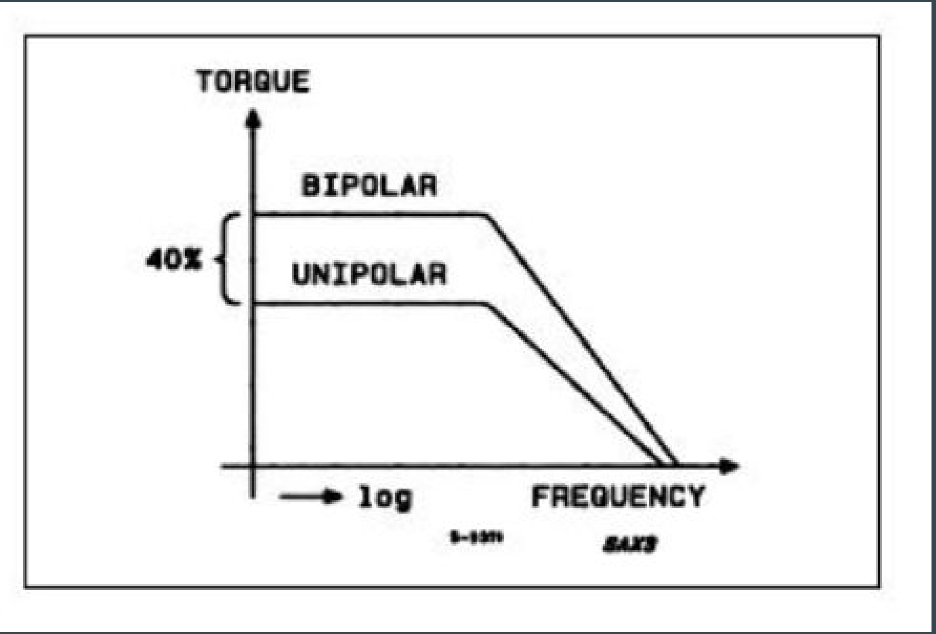


Web Server:


4/30
Trash taken out
Trash level
100.0
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
Trash Level (%)
20:00:15
Trash level 23.0
19:59:09
20:03:20
20:06:40
20:10:00
20:13:20
20:16:40
20:20:00
20:24:44
Date

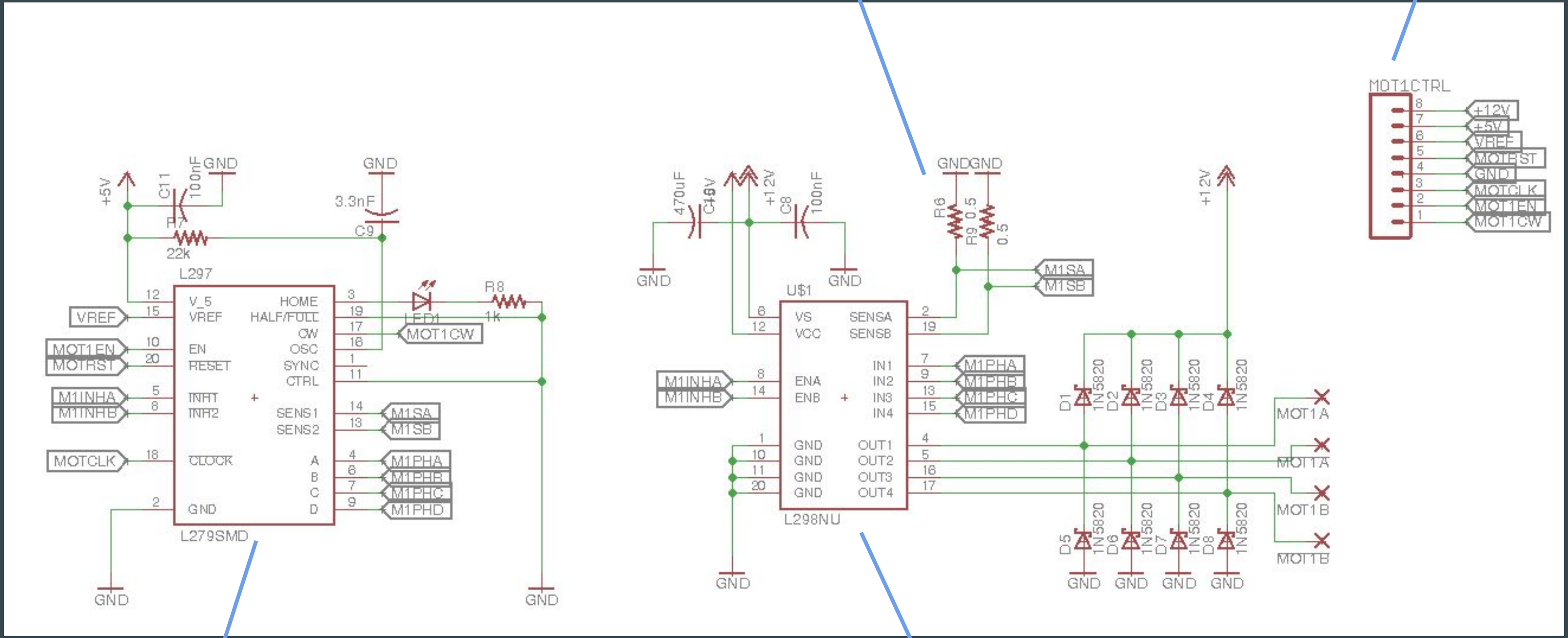
Motor Selection

- Stepper Motor vs DC motor
- Torque vs RPM
- Bipolar vs Unipolar
- Winding resistance -> B -> T
- 8.6V, 2A, 2.35Nm



Current sense resistors

Control signals



L297

L298

Motor Clock Frequency Performance

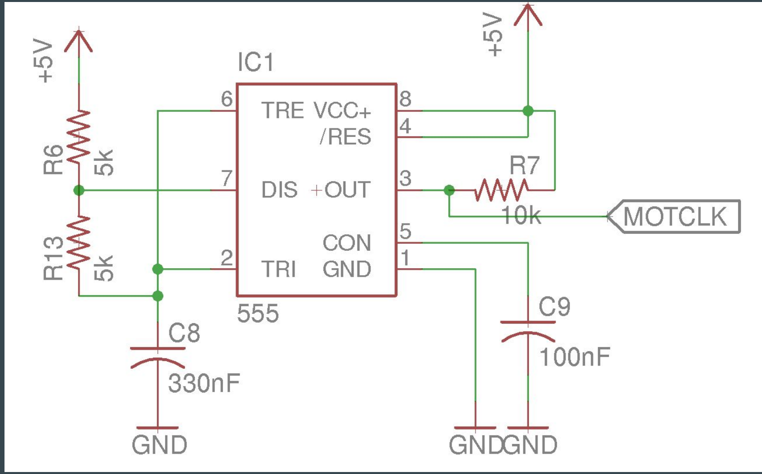
Frequency	Revolutions	Time	RPS
500	50	19.88	2.52
400	50	25.91	1.93
350	50	28.55	1.75
300	50	33.33	1.5
250	50	33.45	1.49
200	10	10.10	0.99

Hardware vs Software Generated Clock

- Atmega328 PWM timing issues
- NE555 timer

$$\text{frequency} \approx \frac{1.44}{(R_A + 2R_B)C}$$

$$\text{Output waveform duty cycle} = \frac{t_H}{t_H + t_L} = 1 - \frac{R_B}{R_A + 2R_B}$$



- Frequency = 291 Hz
- Duty cycle = 66.6%

Linear Velocity Calculation and Desired Wheel Radius

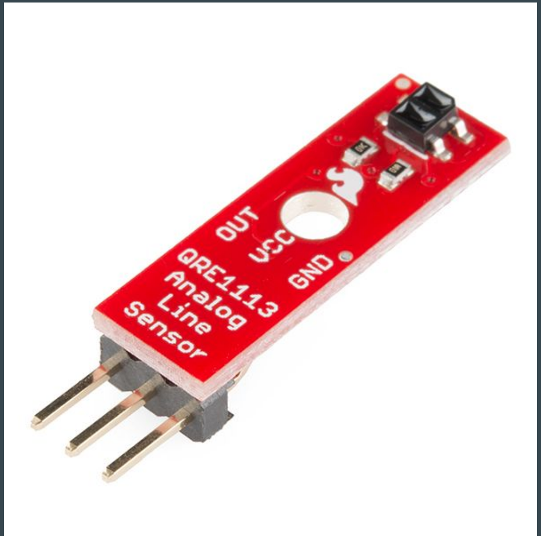
- v = rw
- w = 2π x rps
- v = r x (2π x rps)
- v = r x (2π x 1.5)
- With 40mm radius wheel, speed is 37.7 cm/sec
- No Load

Navigation Options

Left Motor Velocity	Right Motor Velocity	Motion
0	0	Not moving
0	1	Moving left
1	0	Moving right
1	1	Moving straight

IR Sensor and Tape-following

- # of sensors
- Sensor positioning
- Color threshold for IR reading



Navigation Decisions

S1	S2	S3	S4	Result
0	0	0	0	Stop b/c error
0	0	0	1	Right
0	0	1	0	Right
0	0	1	1	Right
0	1	0	0	Left
0	1	0	1	Don’t care
0	1	1	0	Go straight
0	1	1	1	Sharp right

Navigation Decisions contd

S1	S2	S3	S4	Result
1	0	0	0	Left
1	0	0	1	Don’t care
1	0	1	0	Don’t care
1	0	1	1	Don’t care
1	1	0	0	Left
1	1	0	1	Don’t care
1	1	1	0	Sharp Left
1	1	1	1	Stop reached

Obstacle Avoidance

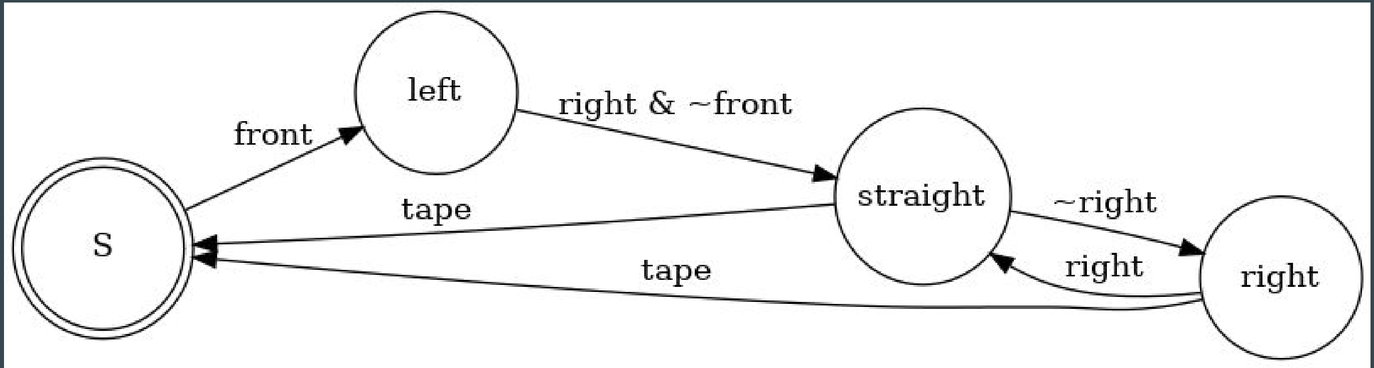
- Use three ultrasound sensors: front, left and right.
- Takes control when front sensor detects obstacle.
- Exact distance measurements by ultrasound sensors necessary.
- Returns control to navigation system when complete.
- Make an initial turn and then follow the obstacle.

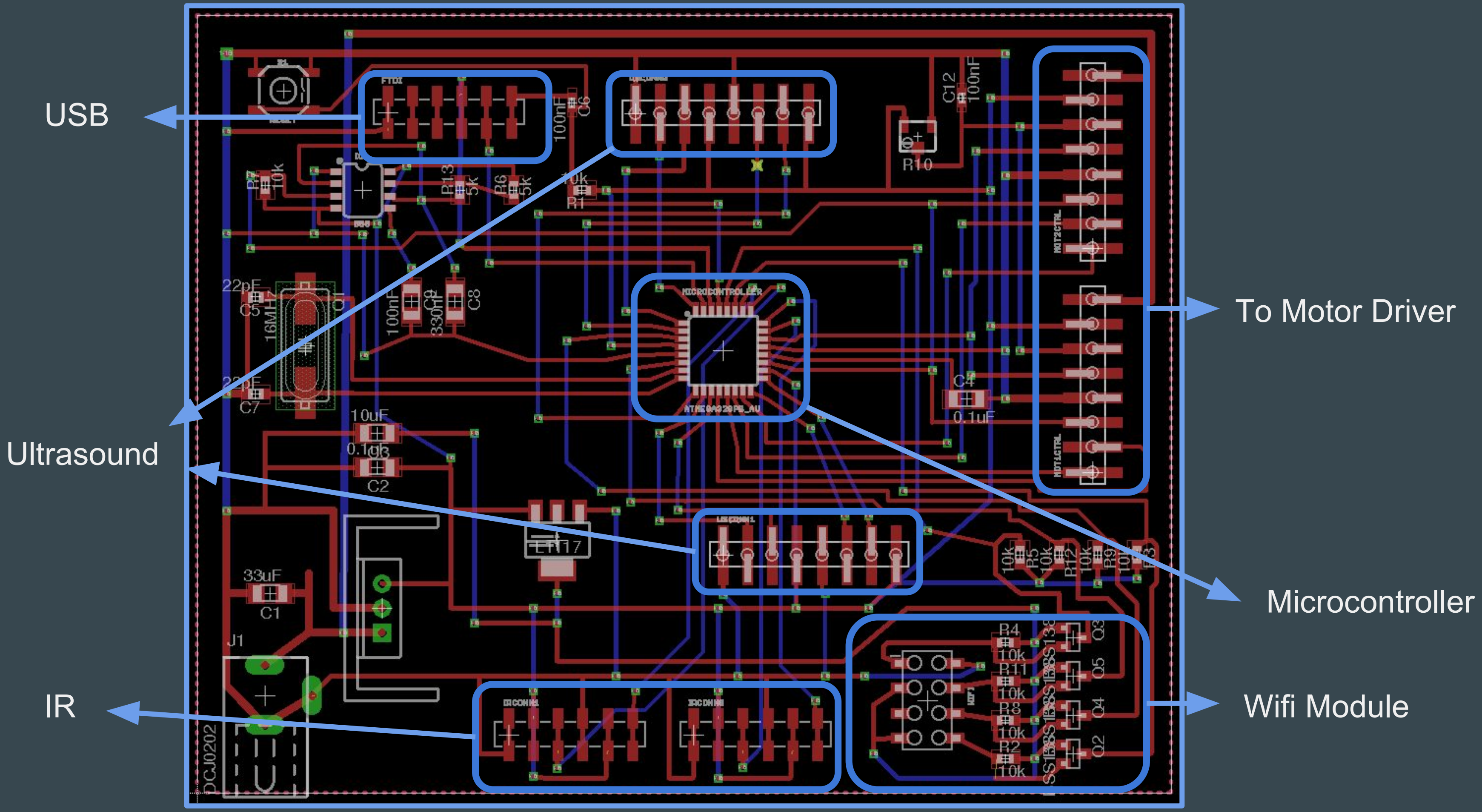
Obstacle Avoidance - Initial turn

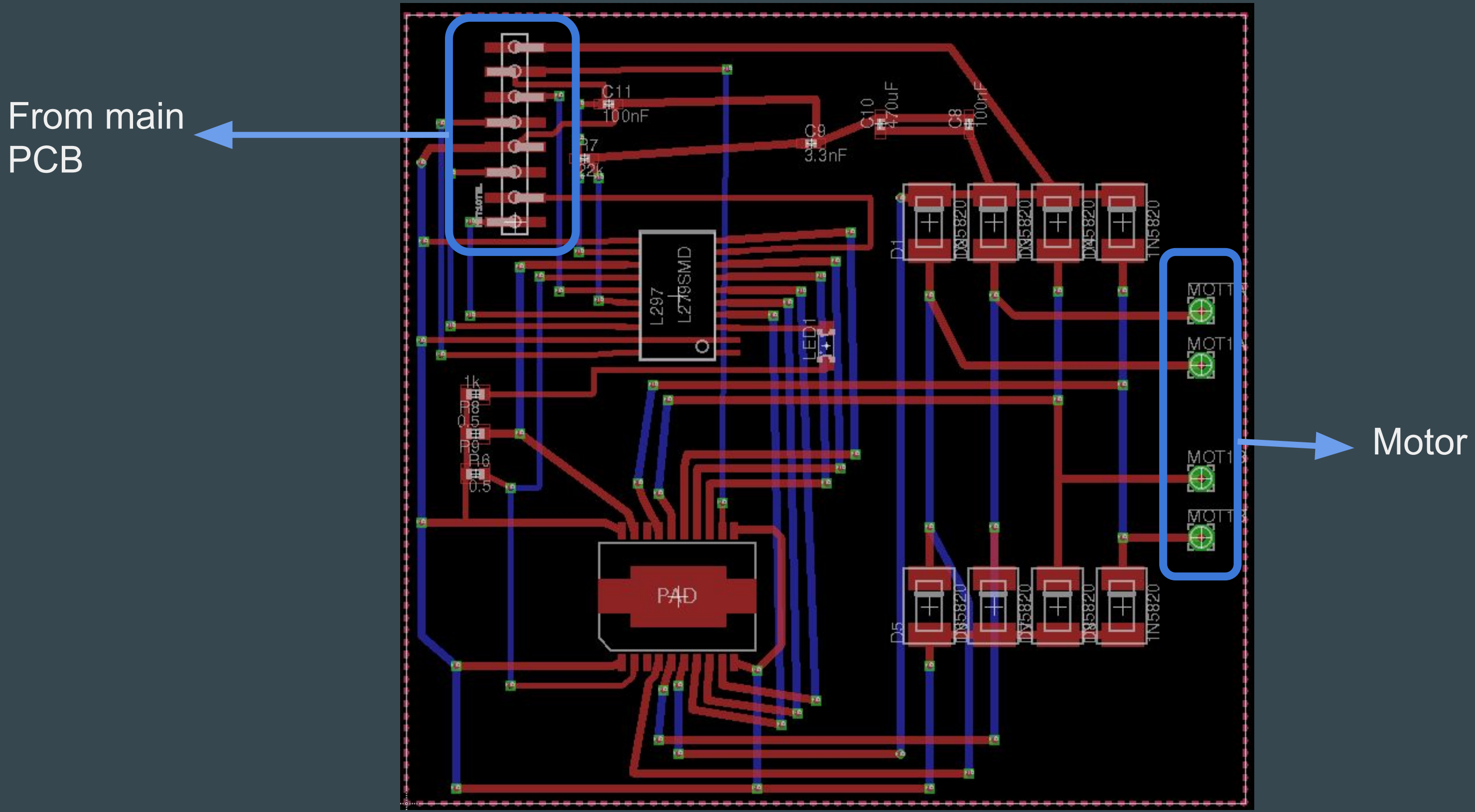
- If neither sensor detects an obstacle, turn left.
- If the left sensor also detects an obstacle turn right.
- If the right sensor also detects an obstacle turn left.
- If both sensors detect an obstacle, reverse.
- Follow state machine for going around the obstacle (left/right).

Going around the obstacle

- Say we choose to go left first.
- Controlled by simple state machine.
  - States represent directions we are moving in.
  - Transitions represent sensor inputs.
  - Exit when tape is detected again




USB
Ultrasound
IR
To Motor Driver
Microcontroller
Wifi Module


From main PCB
Motor
C11
100nF
C10
470uF
C8
100nF
C9
3.3nF
22k
L297
LED1
D1
D5
1N5820
1k
0.5
R6
0.5
PAD

What Went Wrong - PCB

- Clock too far from microcontroller.
- Ordered too few of the same parts.
- Microcontroller bootloader was hard to find.

Navigation Errors

- PCBs not functional after plugging in battery
- Incorrectly plugged in battery
- Vref set for peak load current to be 2A
- L298 cannot sustain 2A without huge voltage drop
- High heat -> chip fried
- Bench vs battery
- Need for current regulator and recalculation of Vref

Conclusion and What went wrong?

- Level-detection aspect successful.
- Problems with navigation.
- Problems with PCB design.
- Problems with power unit.

Further Work

- Support bigger trash cans
- Have smarter navigation
- More web-app features

Questions?