RT meetings

Week 2

Electricity basics

https://youtu.be/mc979OhitAg?si=Lq8q9hTMTrw748iR

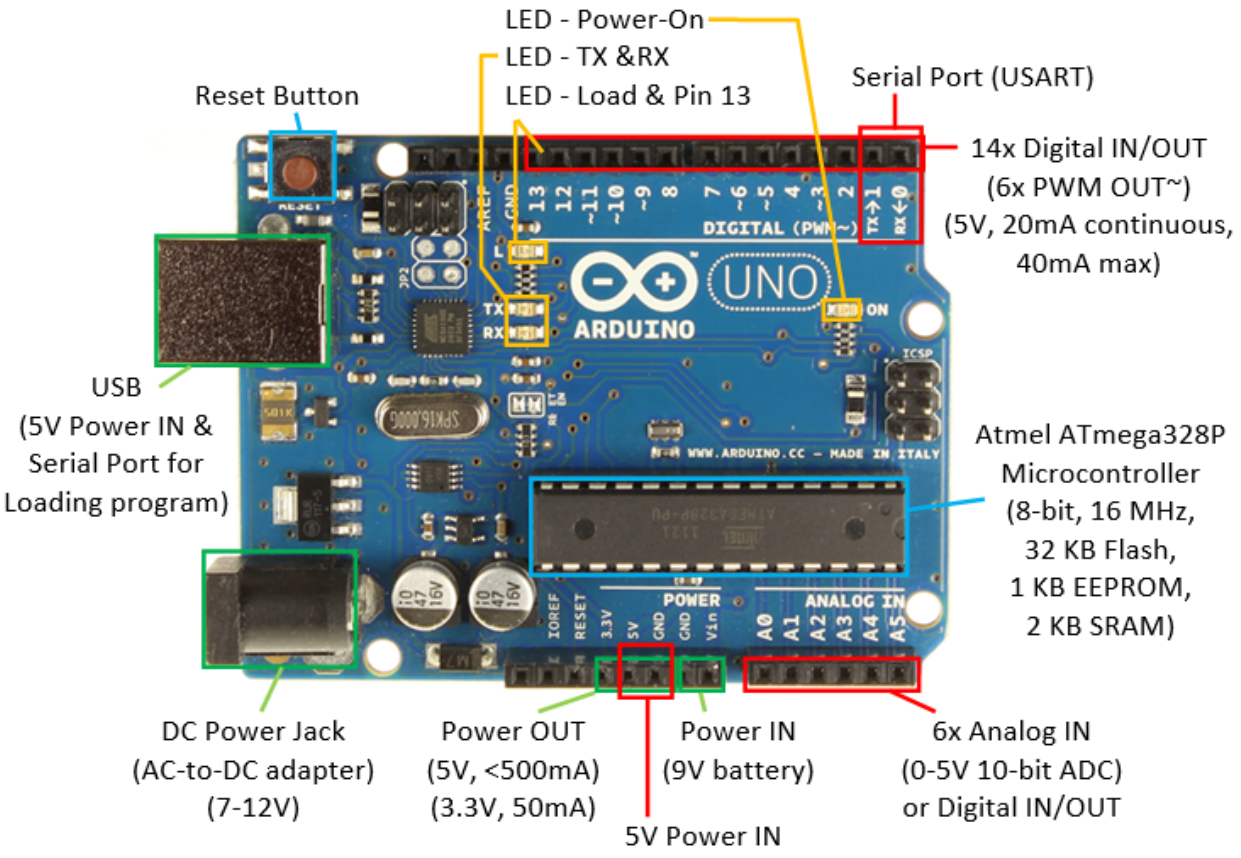
Arduino

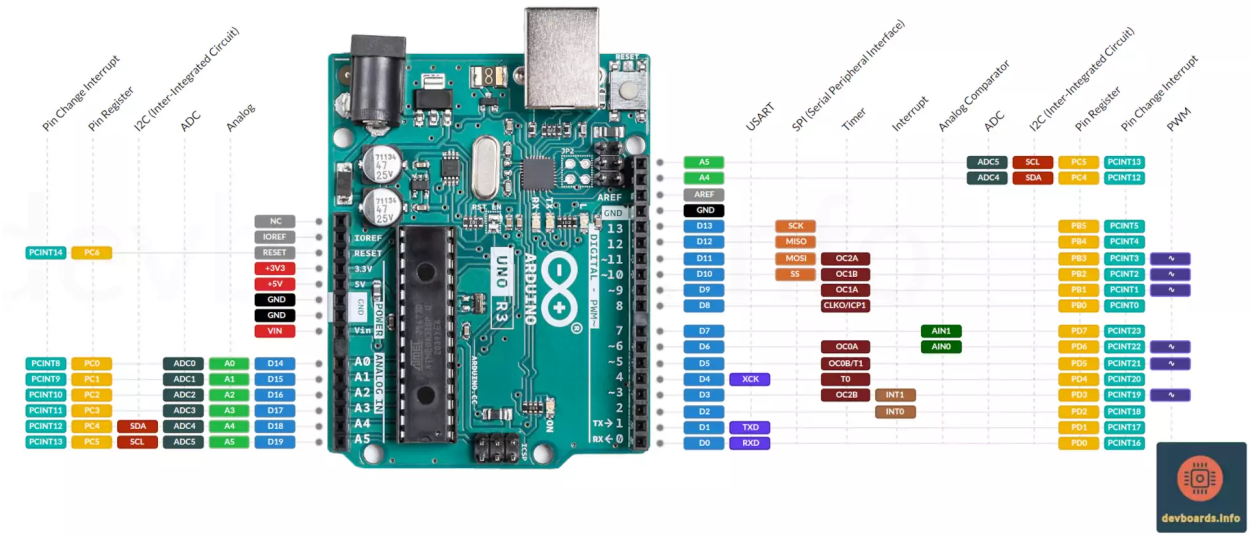
"Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It is intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments."

The official site for Arduino is http://arduino.cc.

https://www.arduino.cc/en/hardware

Arduino UNO


LED - Power-On
LED - TX &RX
LED - Load & Pin 13
Reset Button
Serial Port (USART)
14x Digital IN/OUT
(6x PWM OUT~)
(5V, 20mA continuous,
40mA max)
USB
(5V Power IN &
Serial Port for
Loading program)
Atmel ATmega328P
Microcontroller
(8-bit, 16 MHz,
32 KB Flash,
1 KB EEPROM,
2 KB SRAM)
DC Power Jack
(AC-to-DC adapter)
(7-12V)
Power OUT
(5V, <500mA)
(3.3V, 50mA)
Power IN
(9V battery)
6x Analog IN
(0-5V 10-bit ADC)
or Digital IN/OUT
5V Power IN


devboards.info

https://devboards.info/boards/arduino-uno-rev3

Digital I/O

- D0 - D13, A0-A5 can be used as digital pins(D14 - D18)
- LED on D13, toggled by sending digital HIGH/LOW pulse.

Serial communication/USART

- TX - D0
- RX - D1

I2C

- SDA - A4
- SCL - A5

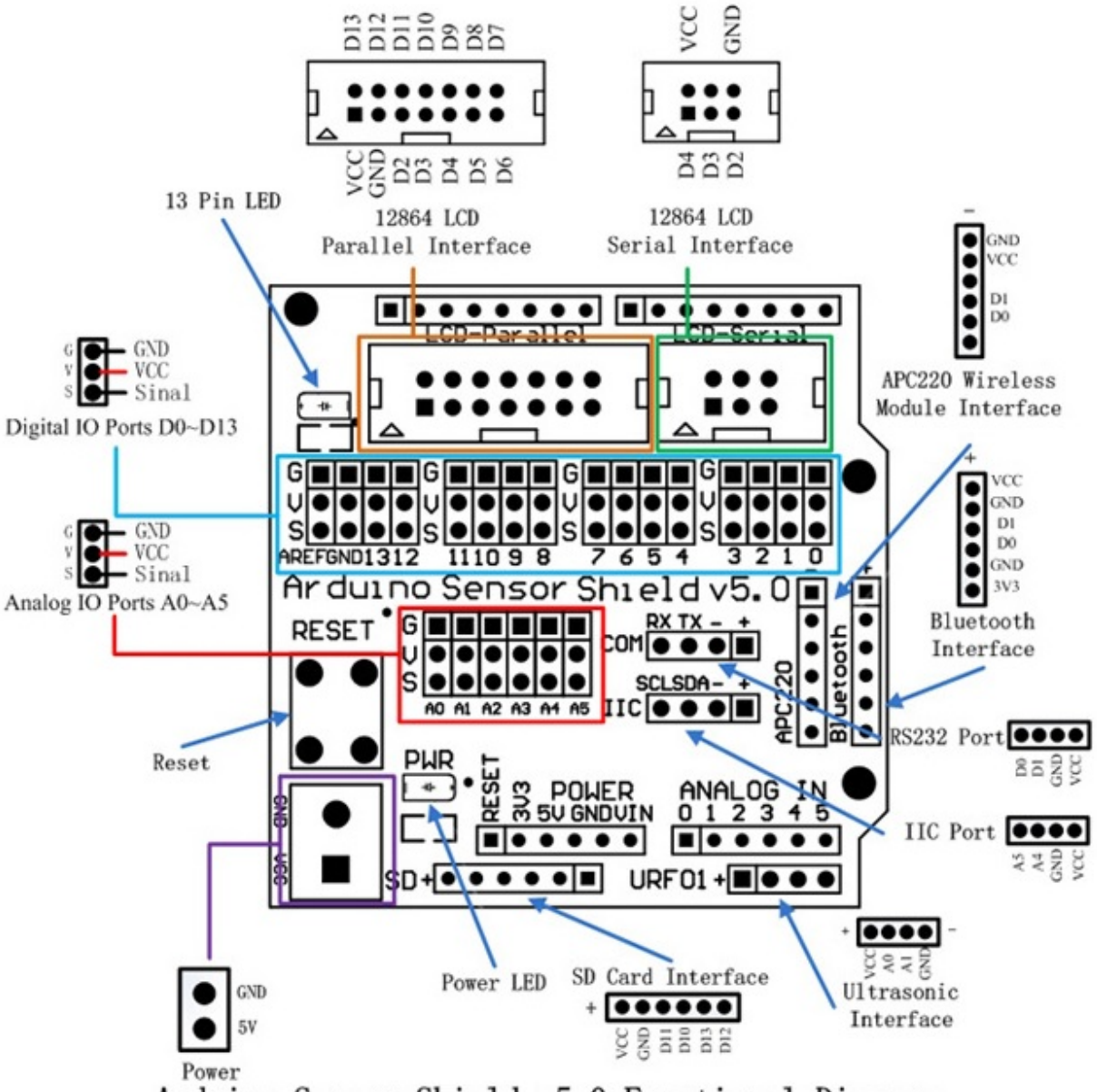
PWN - Pulse Width Modulation

- Digital signal → Analog Voltages
- PWM Pin `~` 3, 5, 6, 9, 10, 11. PWM frequency: 490 Hz (pins 5 and 6: 980 Hz)
- Built-in functions https://www.arduino.cc/reference/en/
  - `digitalWrite(pin, HIGH|LOW)`
  - `analogWrite(Pin, DutyCycle)`, DutyCycle - 0 - 255 (2^8)

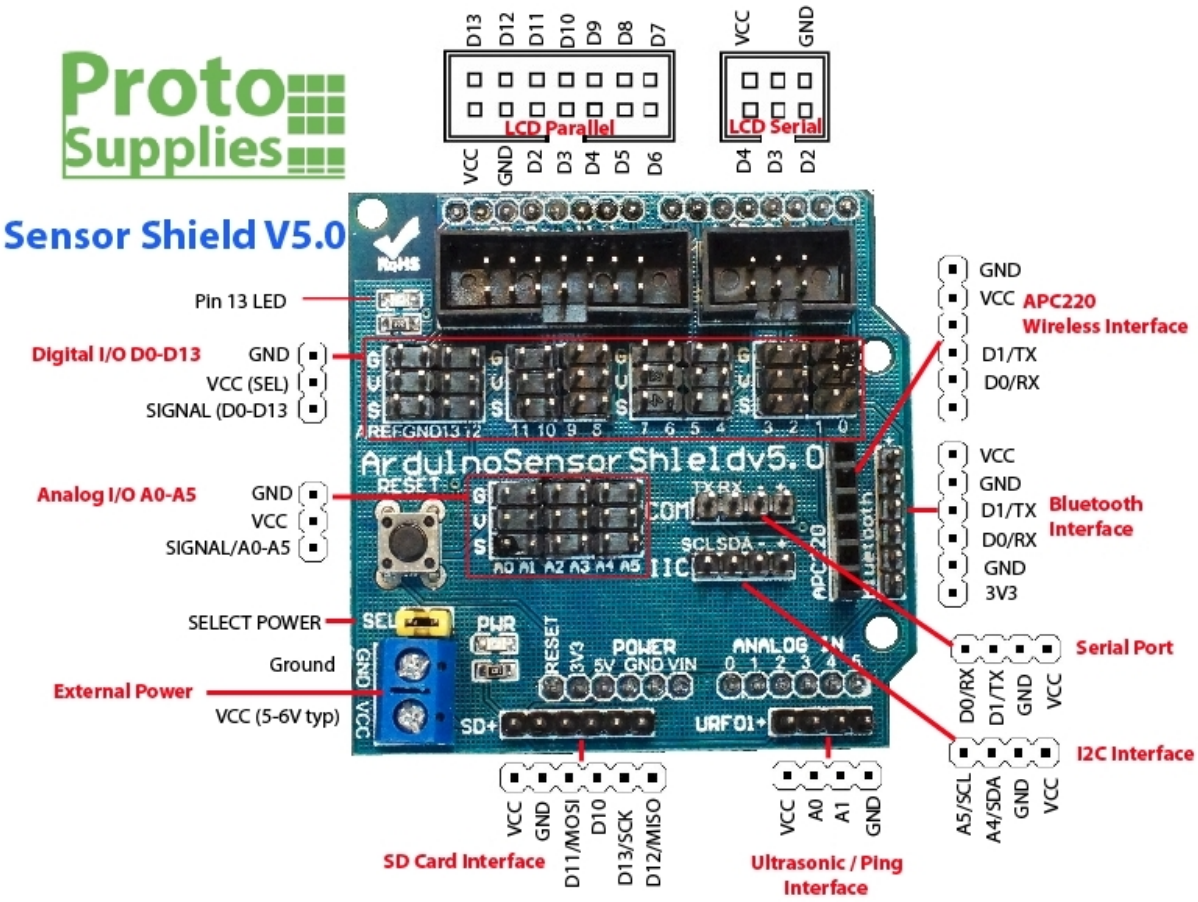
Interrupts

- Hardware interrupt pins: int0 - D2 and int1 - D3
- Timer
- Pin change

Sensor shield v5



Arduino Sensor Shield v5.0 Functional Diagram

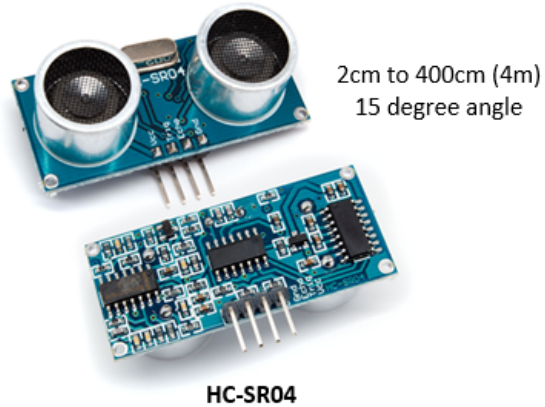


Spec: https://protosupplies.com/product/sensor-shield-v5-0/

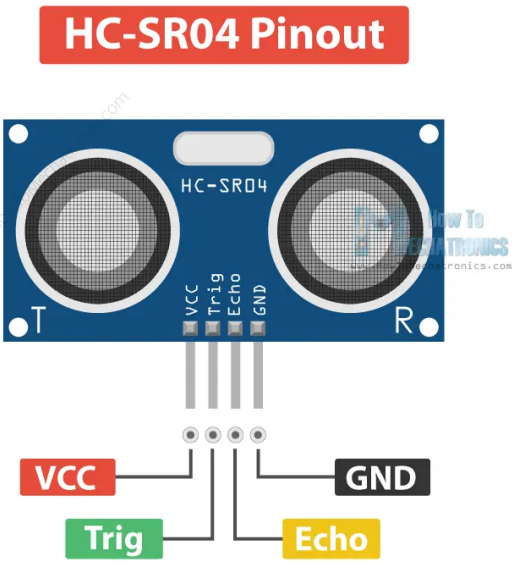
Arduino IDE(Integrated Development Environment)

https://www.arduino.cc/en/software

Ultrasonic sensor HC-SR04



HC-SR04
Ultrasonic Distance Measuring Sensor



1. "Ultrasonic Sensor HC-SR04 and Arduino Tutorial" @ http://howtomechatronics.com/tutorials/arduino/ultrasonic-sensor-hc-sr04/.
2. Arduino-HC-SR04-library @ https://github.com/bbkbarbar/Arduino-HC-SR04-library.

```
speed = distance / time = ?
distance = ?
```

Pins

- VCC
- TRIG - trigger
- ECHO
- GND

```
/*
 * HC-SR04 Pins:
```

```
 * Trig: 12
 * Echo: 13
*/

#define TRIG 12
#define ECHO 13


#define SPEED_OF_SOUND_IN_CM 0.0343 // CM per microsecond 343m/s

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);

  pinMode(TRIG, OUTPUT);
  pinMode(ECHO, INPUT);

}

void loop() {
  // clear the trig port
  digitalWrite(TRIG, LOW);
  delayMicroseconds(5);
  // Trigger sensor by a HIGH pulse of 10 micro seconds
  digitalWrite(TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG, LOW);

  // read the signal from sensor,  time in microseconds
  unsigned long duration = pulseIn(ECHO, HIGH);
  float distance = duration * SPEED_OF_SOUND_IN_CM / 2;
  Serial.print("Distance(cm): ");
  Serial.println(distance);

  // a bit delay
  delay(250);
```

```
}
```

Assembly instruction

https://youtu.be/5xWTDKHbu8w

Week 3

Electricity basics review

- Atom is made up of what particles?
- Which one can move? How do we force them to move
- What is voltage? current ?
- AC and DC? difference?

Electrical energy and power

- energy - total amount of produced or consumed
  - unit: Joule, Wh, kWh
    - `1 Joule = 1 W * s`
    - `1kWh = ? Joules` = 1000 * 3600 J
  - Electrical to mechanical
    - Work - force applied over distance
    - `J =  N * m = (kg * m / (s^2)) * m = m^2 * kg / (s^2)` One joule equals the work done (or energy expended) by a force of one newton (N) is applied over a distance of one meter (m)

    - Electrical energy to heat - `1 Calorie = 4.186 J`
    - 1 Joule can ? lift 98g object up 1 meter.
    - 1 kWh can ?
    - Electricity - 0.1 - 0.40 $ per kWh
- Power - rate of how much energy is transferred over time
    - unit - Watt, Horsepower
    - `1 Watt = 1 Joule per second`
    - `1 HP ~= 745.7 Watts`
    - Power = U * I , I - current, U - voltage
- Battery
    - Voltage - rechargeable - 1.2v
    - Capacity - Amp per hour - mAh, ~ 2000mAh

Motors

How does an electric motor work

https://youtu.be/CWuIQ1ZSE3c?si=d_qT-msUy6985tGU

3 factors

https://youtu.be/4PIhvPTONug?si=a94GHrFSjf5d6DAz

Torque

- Special type of work - produce rotation
- measurement of the force which causes something to rotate around a point
- `Torque = Force * Radius`

- unit: `N * m`
- https://youtu.be/T99yH_gw3p8?si=IFoTM0zer_gmPBNW

Speed

- how fast the rotation
- unit: `RPM` revolutions per minute

Power

- `Power(W) =  Torque(N.m) * Speed(RPM) * (2 * pi) / 60 = Torque(N.m) * Speed(RPM) / 9.549`

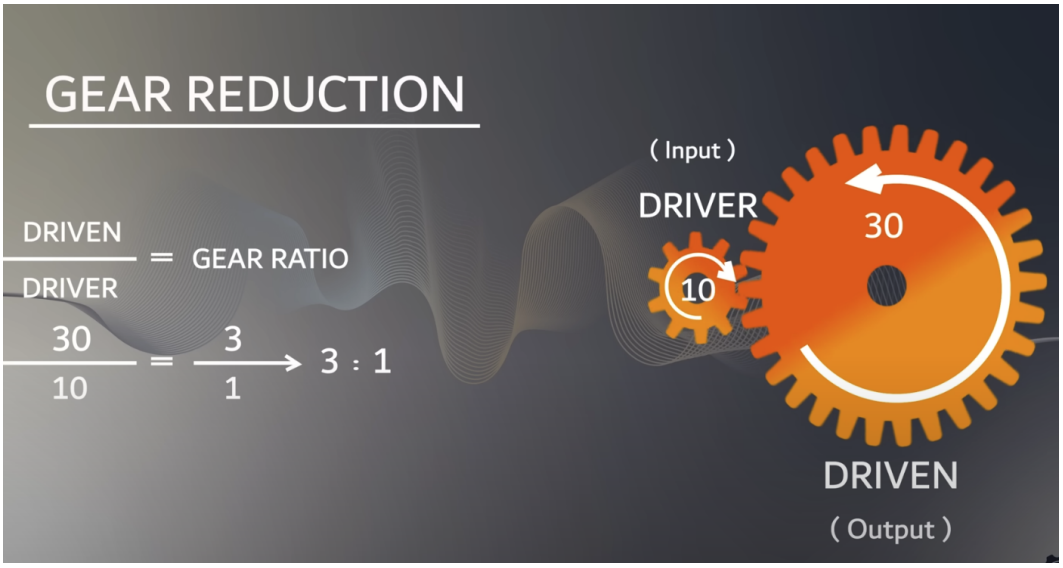
```
N = motor speed(RPM)
R = Radius(m)
F = force(N)
V = speed(m/s)
T = Torque(Nm)

Work = Force * distance
Power = Force * distance / time =  F * V

T = FR
V = (2*pi*R)N/60
P = FV = F * (2 * pi * T/F) * N / 60 = TN/9.549
```

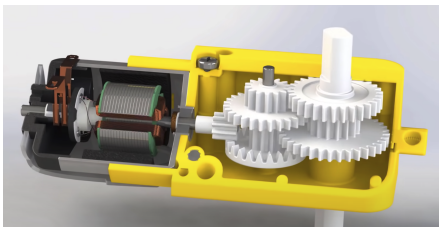
Gear ratio

- reduction in speed

```
driver - in
driven - out

P(in) ~= P(out)
T(out) / T(in) = R(in) / R(out) = N(num of teeth out) / N(in)
Gear ratio = RPM(in) / RPM(out) =  N(num of teeth out) / N(in) =
```

```
P = NT
T = F * R
```





TT motor

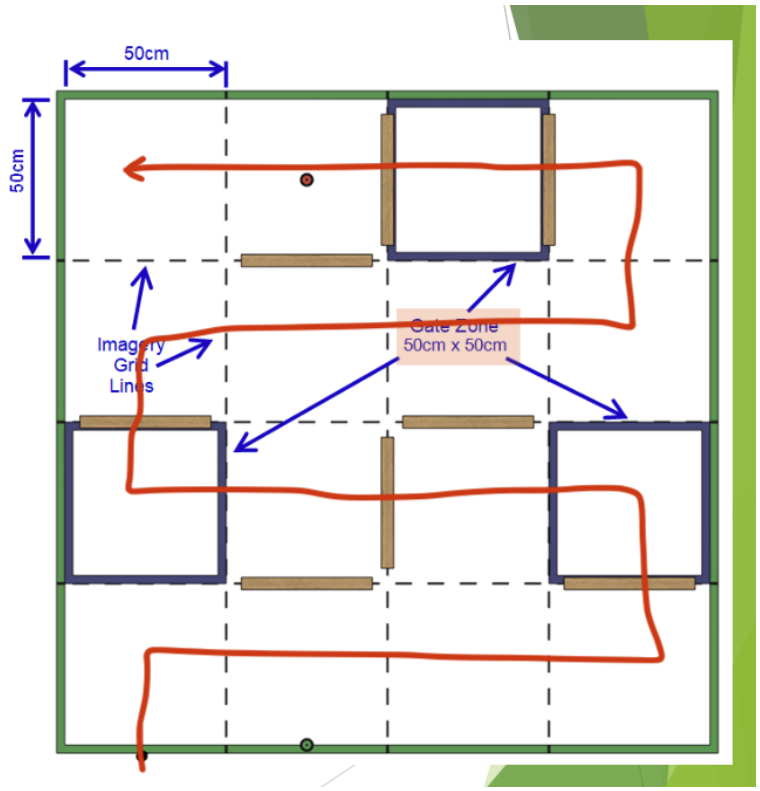
- Voltage: 3V - 12v, recommended 3V - 9V.

- Min operating speed:(6V): 200 RPM
- No load current at 6V: ~200mA
- Torque: 0.15Nm - 0.60Nm
- Gear Ratio: 1:48
- 1: 100?

https://youtu.be/GPVC84D5ULw?si=X0bctaHsO9KigFGh

Math

- If the robot moves along the path in red, what's the motor speed(RPM) we should set? Given that diameter of the wheels is 6.5cm, target time is 1 minute.



```
Diameter of the wheel
D = 6.5 cm

Circumference of the wheel
C = pi * D = 20.41 cm

distance L = 150 * 4 + 175 = 775 cm

num of revolutions
N = L / C = 775 cm / 20.41 cm = 38 r

Motor speed
target time: 60 seconds:  RPM = 38 r/min
target time : 50 seconds?  RPM = 38 r/min * 60 / 50 = 45.6 r/
target time : 70 seconds?  RPM 38 r/min * 60 / 70 = 32.6 r/mi
```

Arduino built-in functions

Arduino programming reference doc https://www.arduino.cc/reference/en/

Digital I/O input output

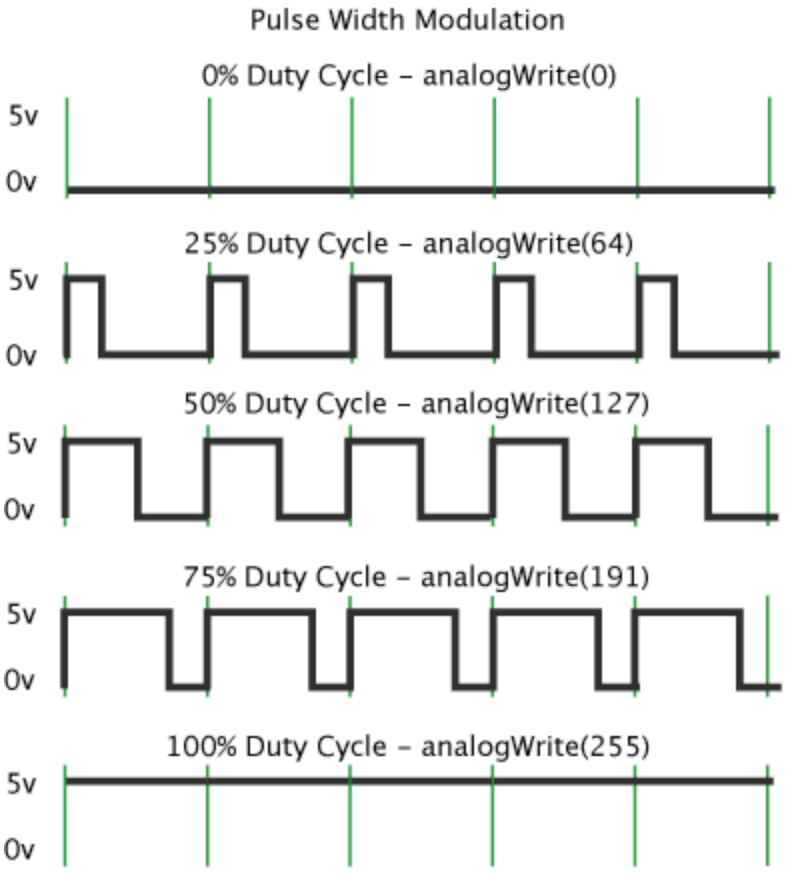
- `pinMode(INPUT|OUTPUT)`
- `digitalWrite(pin, HIGH|LOW)` - apply high or low voltage to a specified pin
- `digitalRead(pin)`

Advanced I/O

- `pulseIn(pin, HIGH|LOW)` return duration of the pulse

PWM - Pulse Width Modulation

- https://docs.arduino.cc/learn/microcontrollers/analog-output
- Digital signal → Analog Voltages
- Arduino UNO: PWM Pin `~` 3, 5, 6, 9, 10, 11 , PWM frequency: 490 Hz (pins 5 and 6: 980 Hz)
- `analogWrite(Pin, int DutyCycle)`, DutyCycle - 0 - 255 (2^8), 0 - always off, 255 - always on.

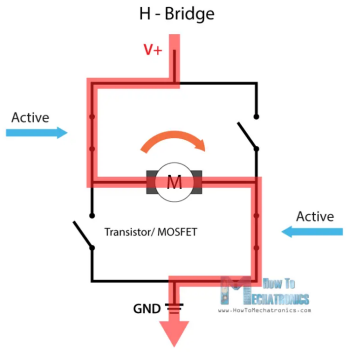


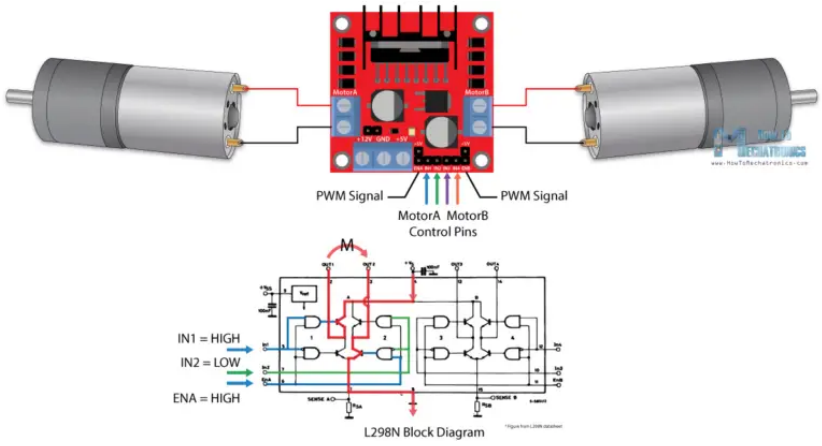
L298N - Motor driver

https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/

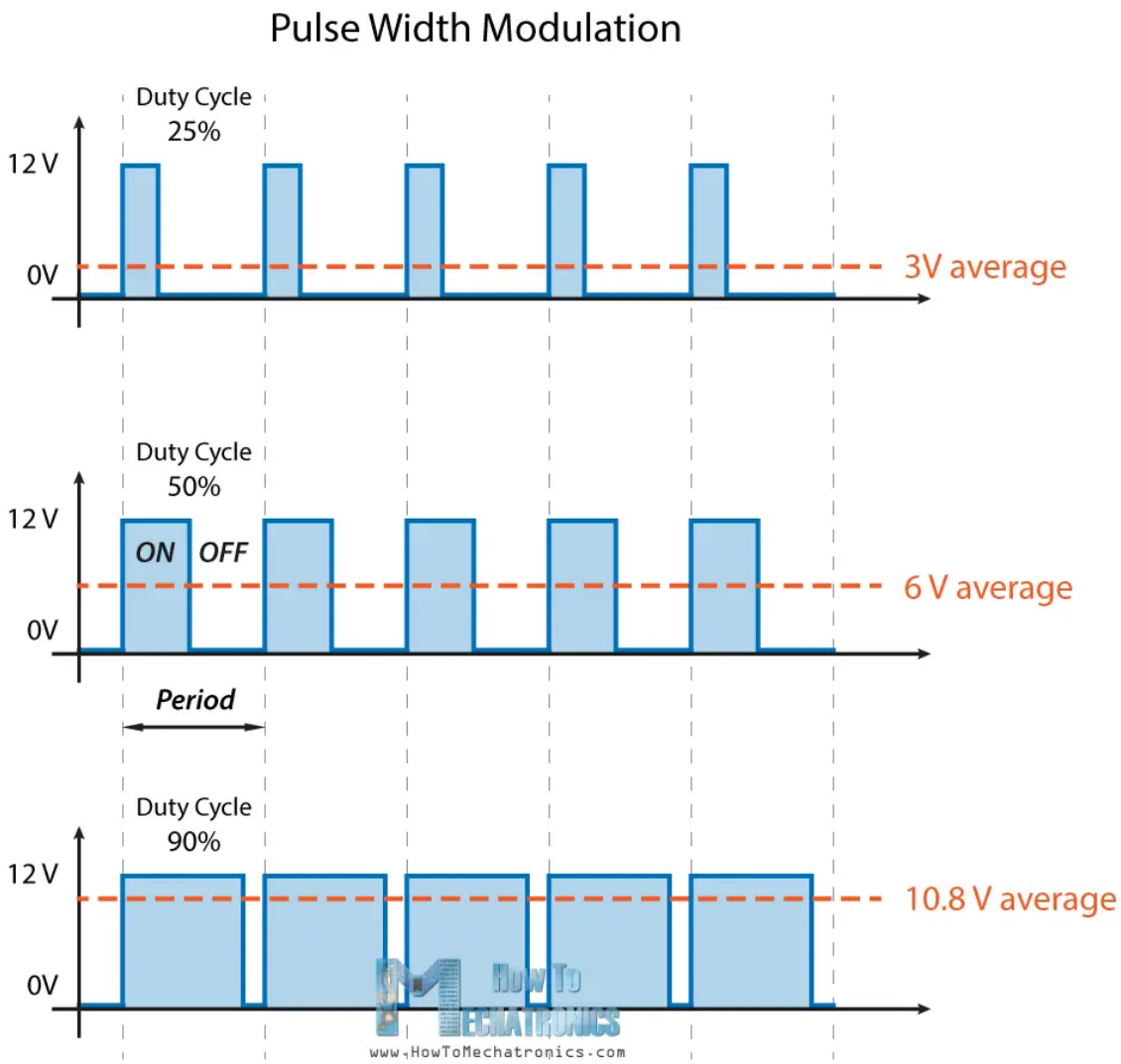
L298N Module Pinout Configuration

Pin Name	Description	
IN1 & IN2	Motor A input pins. Used to control the spinning direction of Motor A	
IN3 & IN4	Motor B input pins. Used to control the spinning direction of Motor B	
ENA	Enables PWM signal for Motor A	
ENB	Enables PWM signal for Motor B	
OUT1 & OUT2	Output pins of Motor A	
OUT3 & OUT4	Output pins of Motor B	
12V	12V input from DC power Source	
5V	Supplies power for the switching logic circuitry inside L298N IC	
GND	Ground pin	





ENA, ENB PWM



Homework

Programming

- How do we turn left or right?
- How do we stop when ultrasonic sensor detects that an object is 10cm away?

Brainstorm

- How do we measure the motor speed, distance traveled, robot orientation?

Week 4

Review

- Motor - Torque and speed
  - Power = Torque * RPM * constant
  - Input power ~= output power
    - Power efficiency **η = 100% · Pout / P***in*
    - Pin - electric power
    - Pout = ?

```
Pin = I * V

Pout = ?

N - motor speed(RPM)
R - Radius(m)
F - force(N)
V - speed(m/s)
T - Torque(Nm)
C - Circumference = 2 * pi * R

E(energy/work) = Force * distance
Power = Force * distance / time =  F * V

T = FR -> R = T/F
V = N * C / 60 = (2*pi*R)N/60
P = FV = F * (2 * pi * T/F) * N / 60 = TN/9.549
```

- Driver(input) gear and Driven(output) gear

```
P(in) ~= P(out)
T(out) / T(in) = R(in) / R(out) = N(num of teeth out) / N(in
Gear ratio = RPM(in) / RPM(out) =  N(num of teeth out) / N(i
```

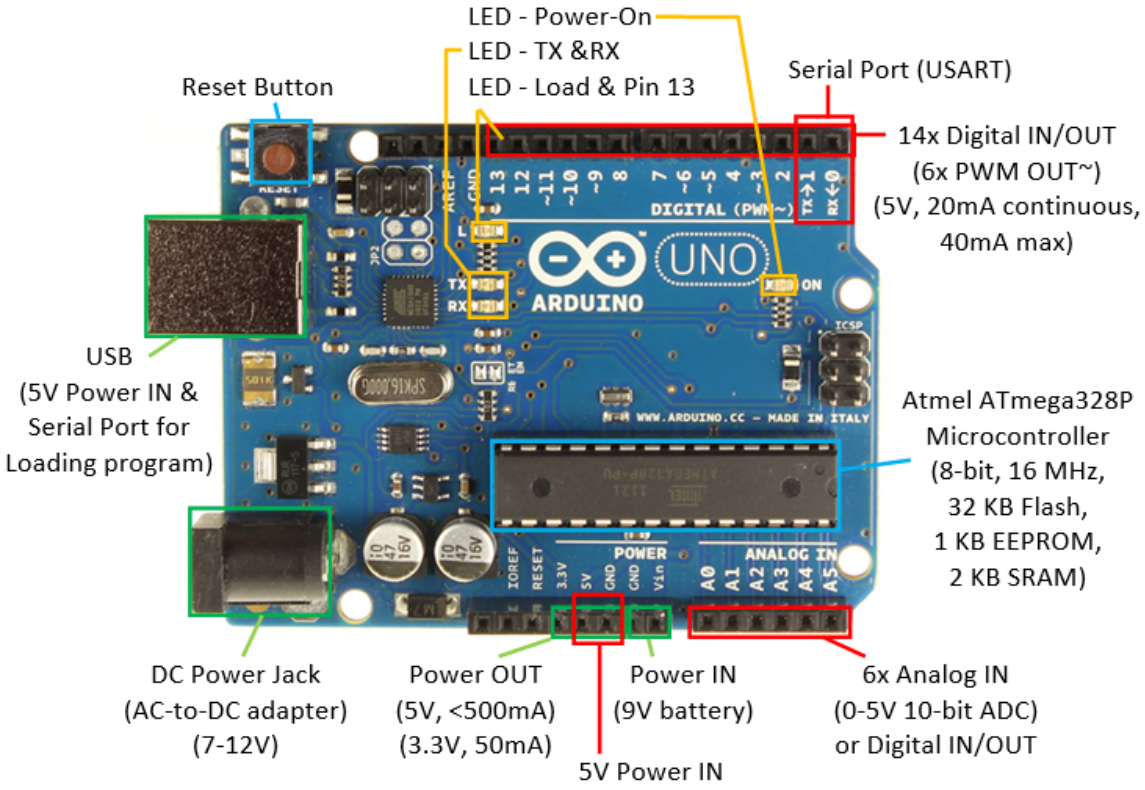
- What is `L298N`, why do we need it?
- How do we control the wheel rotation direction?
- How do we control the motor speed?
- When we apply 100% duty cycle, what's the voltage on the L298N motor output?

Arduino UNO and microcontroller Atmega328P

Arduino UNO R3 data sheet , wikipedia

ATMega328P data sheet

- CPU speed - 20MHz
- Flash memory - 32KB
- EPROM(erasable programmable read-only memory) - 1KB
- RAM(random access memory) - 2KB
- I/O ports
  - 14 digital I/O pins - D0 - D13
    - Built-in LED on pin D13
    - Serial communication through pin D0(Rx) and D1(Tx)
    - 6 PWM pins - D3, D5, D6, D9, D10, D11, PWM frequency: 490 Hz (pins 5 and 6: 980 Hz)
      https://docs.arduino.cc/learn/microcontrollers/analog-output
    - Interrupts - pin D2, D3
  - 6 analog pins - A0 - A5



Programming (c++) basics

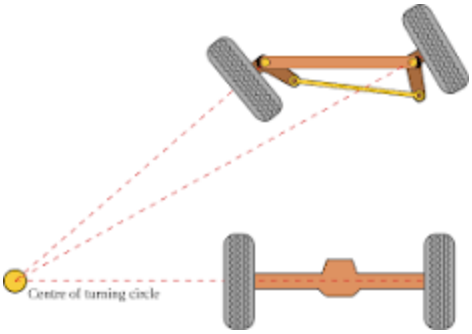
- Pre-defined macros - `#define` directive
  - syntax `#define identifier token-string`
  - compiler to substitute `token-string` for each occurrence of `identifier` in source code
- Built-in data types
  - character - `char`
  - Integer - `int`, with `unsigned long int`
  - Floating point - `float`, `double`
  - Boolean - `bool`
  - empty - `void`
- functions

Programming

- How do we control the robot to drive straight?
- How do we turn left or right?
    - Differential steering
- How do we stop when ultrasonic sensor detects that an obstacle

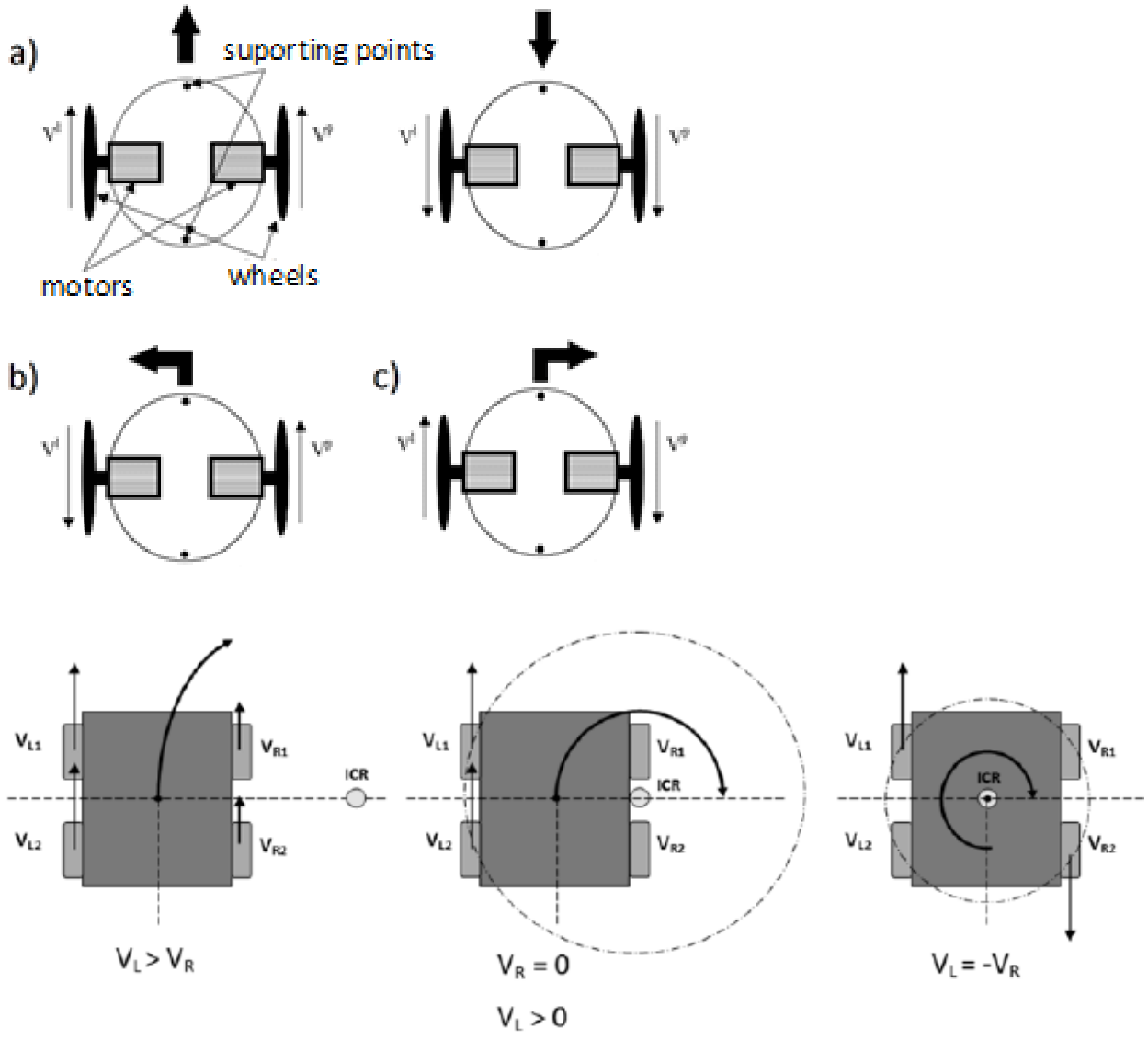
Steering

Ackermann steering



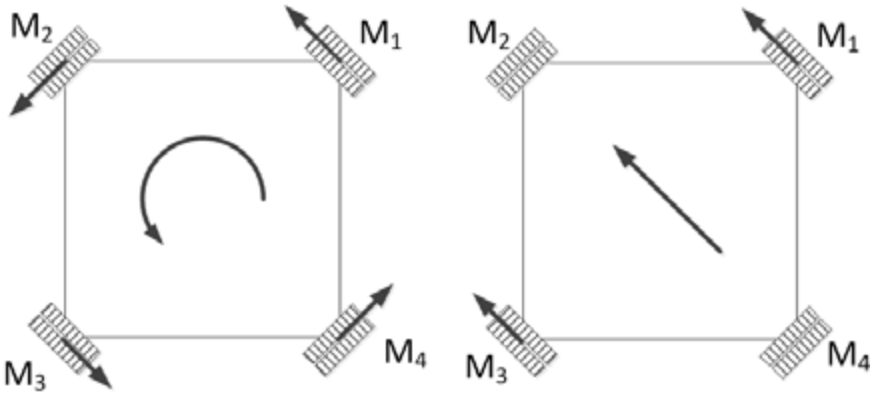
Rear-wheel drive(RWD)

Differential steering

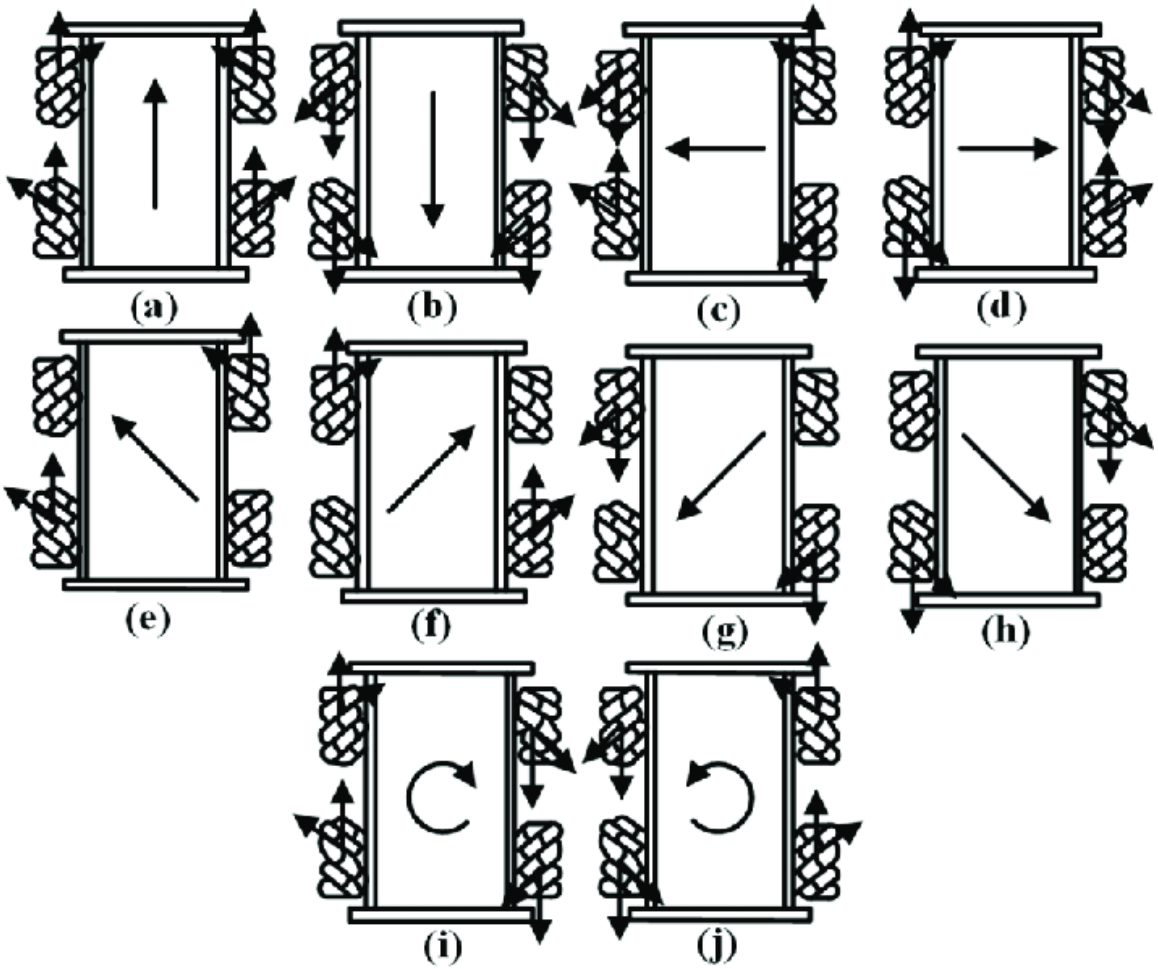


Omnidirectional steering

4-wheel omnidirectional wheel robot



4-wheel Mecanum wheel robot



Homework

Programming

- Write and test a program: `move 150 cm, stop for 1 second,  turn left, stop for 1 second, move 50 cm, stop`. Don't worry about that the robot doesn't drive straight or turn exactly 90 degree, adjust the code as much as you can with this open loop control
- If you have a multimeter, measure the motor driver output voltage(for both motors) when we send `127` and `255` duty cycle PWM signals.
- Think about how do we program to stop the robot when the ultrasonic sensor detects that an obstacle is 10 cm away?

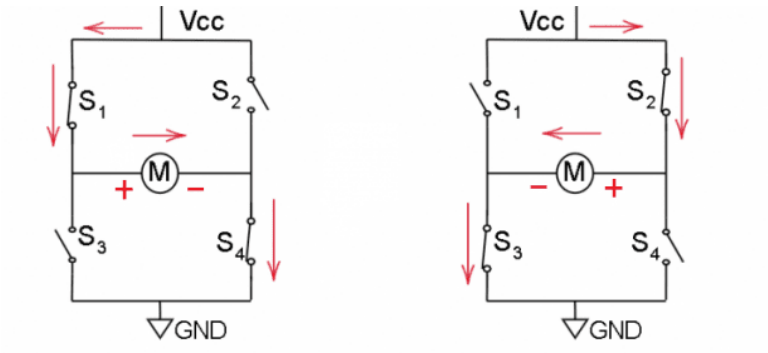
Preparing for closed loop control

- To under stand what is MPU-6050? https://components101.com/sensors/mpu6050-module
- Nathan and Jordan to solder the pins for MPU-6050 module.

Week 5

Motor basics

- Why do we need motor driver module?
  - speed and direction control
  - drive dc motors with high voltage and current.
- Motor spinning direction
  - Switch the direction of the current flow - By inverting the polarity of the applied voltage
  - Motor driver - L298N , H-bridge circuit.

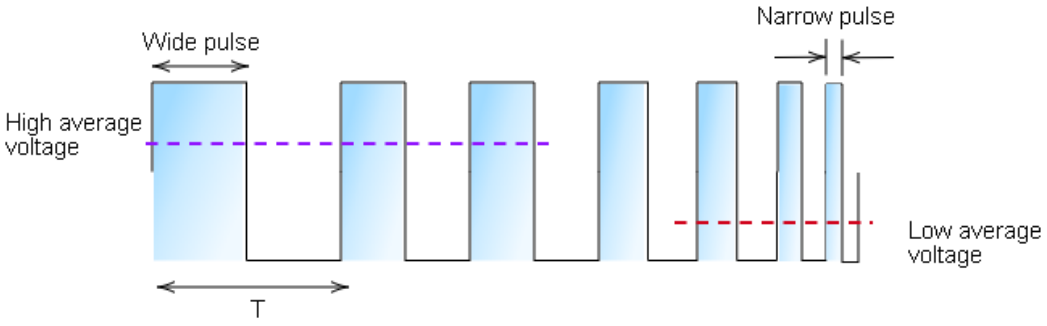


  - L298N - motor A - INT1, INT2, motor B: INT3, INT4.

INT1	INT2	Spinning direction
LOW	LOW	Power off
LOW	HIGH	Forward(CCW)

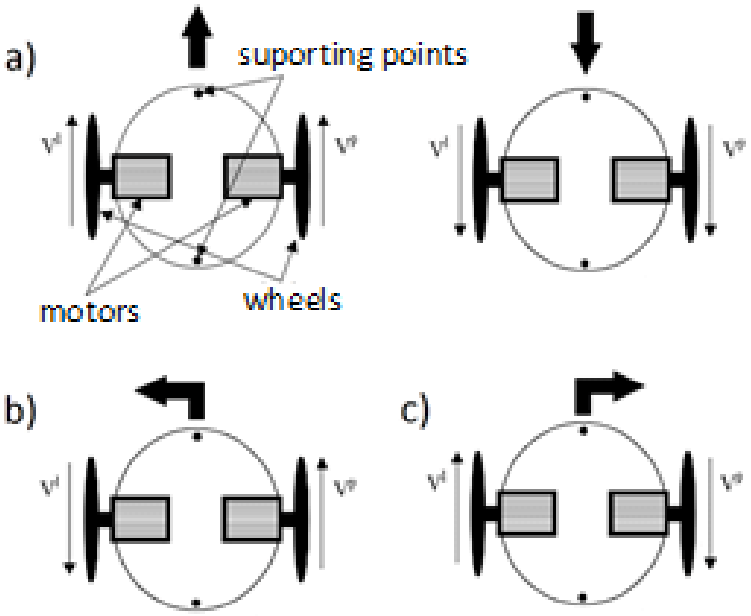
INT1	INT2	Spinning direction
HIGH	LOW	Backward(CW)
HIGH	HIGH	Power off

- Motor speed
  - PWM
    - By switching the supply voltage on and off very quickly, adjusting the length of the on/off pulses, we can set the voltage between 0V to maximum voltage.
    - Arduino UNO: 0 - 255, 0 - 100% voltage.
    - L298N - motor A: ENA, motor B: ENB



- gear reduction
  - reduce output RPM
  - increase output torque
- Robot velocity in (m/s)
  - Velocity = Circumference * RPM = PI * Diameter * motor speed(RPM) / 60
- Measure distance traveled? in (m/s)
  - Robot speed * time = PI * D * motor speed(RPM) * time / 60

Differential Steering



- Move forward or backward: V left = V right
- turn right (rotate CW) = V left = - V right
- turn left (rotate CCW) = V right = - V left

Homework review

Programming

- Write and test a program: `move 150 cm, stop for 1 second,  turn left, stop for 1 second, move 50 cm, stop`. Don't worry about that the robot doesn't drive straight or turn exactly 90 degree, adjust the code as much as you can with this open loop control
- If you have a multimeter, measure the motor driver output voltage(for both motors) when we send `127` and `255` duty cycle PWM signals.
- Think about how do we program to stop the robot when the ultrasonic sensor detects that an obstacle is 10 cm away?

Preparing for closed loop control

- To understand what is MPU-6050? https://components101.com/sensors/mpu6050-module

- Nathan and Jordan to solder the pins for MPU-6050 module.

C++ Programming

Coding style - naming

Naming https://google.github.io/styleguide/cppguide.html#Naming

- variable names(including parameters) - `snake_case`
- macros - `#define UPPER_CASE_SNAKE_CASE`
- Class and class member method names - `PascalCase`
- function names `camelCase`, e.g `digitalWrite`

Class - OOP - Object-oriented programming

- class - a blueprint for create objects that contain both data and functions, a particular data structure.
- member attributes - initial values for state
- member methods - implementations of behavior

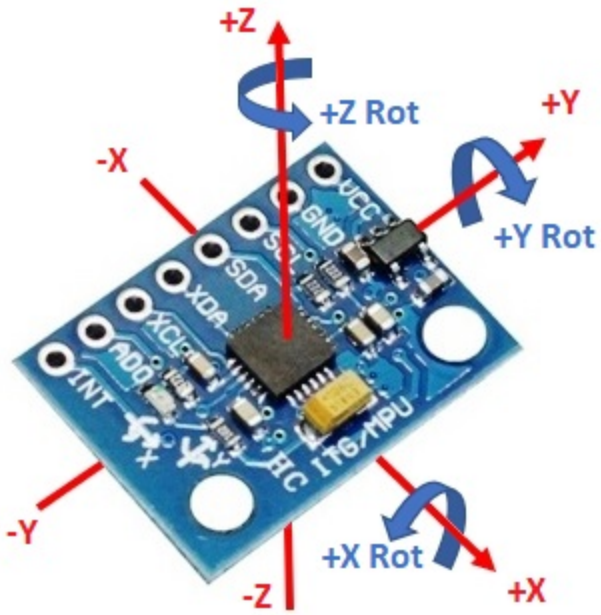
example class

```
// PIN definitions - motor A - left motor
#define ENA 5 // PWM
#define INT1 4
#define INT2 7

class Robot {
    public:
        // member attributes
        int num_driver_wheels; // 2, 4
        char steering_method; // 'd' - differential,  'a' - Acke
```

```
        // constructor
        Robot(int num_driver_wheels, char steering_type);

        // member methods
        void Move(int pwm, unsigned long duration);

};

Robot::Robot(int num_driver_wheels, char steering_type) {
    num_driver_wheels = num_driver_wheels;
    steering_method = steering_method;
    pinMode(ENA, OUTPUT);
    pinMode(INT1, OUTPUT);
    pinMode(INT2, OUTPUT);
}

void Robot::Move(int pwm, unsigned long duration) {
    digitalWrite(INT1, LOW);
  digitalWrite(INT2, HIGH);
    analogWrite(ENA, pwm);
    delay(duration);
    digitalWrite(INT2, LOW);
}

void setup() {
    Robot my_robot = Robot(2, 'd'); // instantiate an object of
    delay(3000);
    my_robot.Move(127, 3000); // call the member method Robot::M
}

void loop() {
}
```

MPU-6050 Accelerometer and Gyroscope sensor IMU

- https://howtomechatronics.com/tutorials/arduino/arduino-and-mpu6050-accelerometer-and-gyroscope-tutorial/
- https://youtu.be/XCyRXMvVSCw?si=aGis4m1eSYKDvu9i
- IMU - Inertial measurement unit
- 3-Axis Gyroscope - rotational velocity



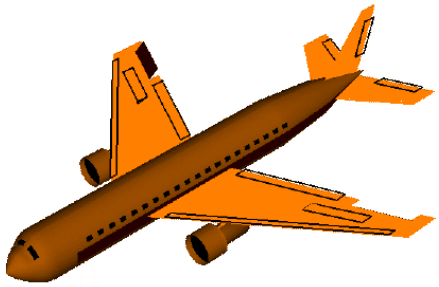
The MPU-6050 can measure rotation using its on-chip gyroscope with four programmable full scale ranges of ±250°/s, ±500°/s, ±1000°/s and ±2000°/s that can be set by the user.

- 3-Axis accelerometer - measures acceleration

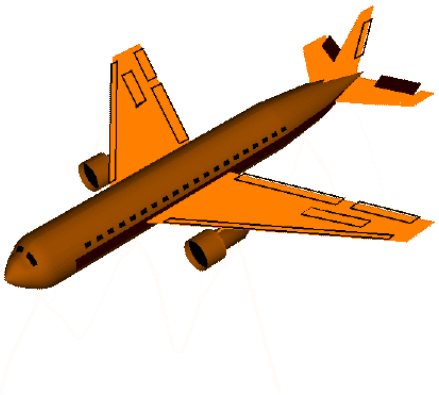
The MPU-6050 can measure acceleration using its on-chip accelerometer with four programmable full scale ranges of ±2g, ±4g, ±8g and ±16g that can be set by the user.

Dimensions of movement

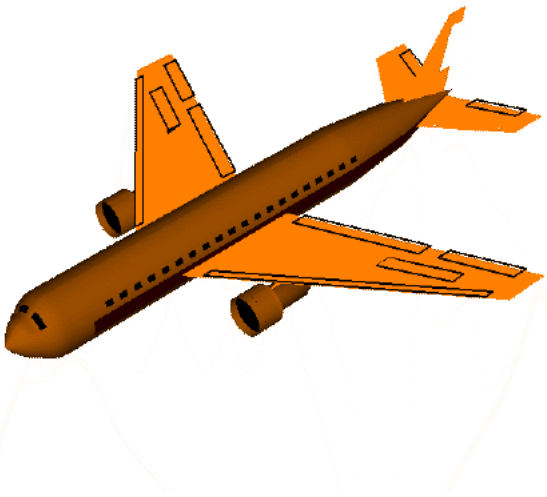
- Roll - a circular movement of the body



- Pitch - nose up or tail up



- Yaw - nose moves from side to side



Wiring

- VCC
- GND
- SCL(Serial Clock) - used for providing clock pulse for I2C communication
  - to Arduino pin A5
- SDA(Serial Data) - used for transferring data through I2C communication

- to Arduino pin A4

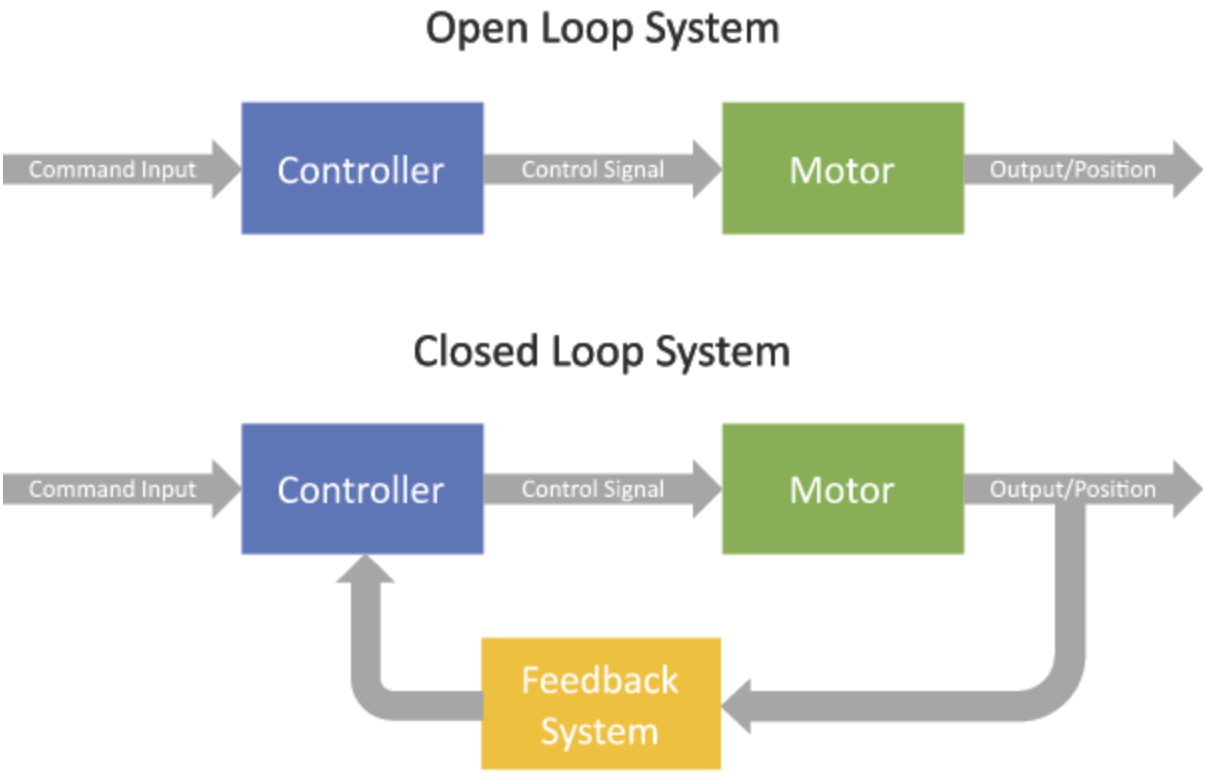
Read Arduino I2C protocol https://docs.arduino.cc/learn/communication/wire

Arduino MP-6050 libraries

- https://www.arduino.cc/reference/en/libraries/mpu6050/
- https://github.com/rfetick/MPU6050_light

Open loop and closed loop control

- Open loop - no feedback system. The system does not measure and act upon the output of the system.
- Closed loop - feedback system. The system measures the output of the system compared to the desired input, and takes corrective action to achieve the desired result.



Closed loop speed controller

- controller input: target motor speed(RPM), measured RPM, `error = target RPM - measured RPM`
- motor output: measured RPM
- controller output: PWM

Closed loop position controller - shaft position - N (number of revolutions)

- controller input: target shaft position, measured shaft position from encoder, `error = target shaft position - measured position`
- motor output: measured shaft position
- controller output: PWM

Closed loop drive straight controller

- feedback: MPU-6050
- input: target yaw movement angle: x, measured yaw movement angle, error = 0 - measured yaw
- output: PWM

Homework

- Test sensors
  - Optical flow sensor `PAA5100JE` https://shop.pimoroni.com/products/paa5100je-optical-tracking-spi-breakout?variant=39315330170963. No Arduino library or code example.
  - Laser Time-of-Flight ranging sensor `VL53L0X` https://learn.adafruit.com/adafruit-vl53l0x-micro-lidar-distance-sensor-breakout/arduino-code
  - Infrared Obstacle avoidance sensor `KY-032` https://arduinomodules.info/ky-032-infrared-obstacle-avoidance-sensor-module/

    - LCD 2004 - I2C connection https://youtu.be/cbL_7I9enqU?si=2DddlVfHxPDfXL1E

- Prep for next week
    - PID control
    - Control motors with encoders https://youtu.be/dTGITLnYAY0?si=p_hgK2rOmlGla0kl

Week 6

Prep for competitions

- Robot
    - Wooden dowel, and prepare the dowel - 1/4 to 3/8 (inch), 10cm long
        - Nathan to design and 3D print the `dowel rod holder` Example https://topfinishkits.com/index.php/models/robot
    - USB flash drive
    - 6 AA Batteries
    - Button to start the run,
    - Tools
        - rules: `Teams may bring tools, which includes a stand-alone non-programmable, non-graphing calculator (Class II), which do not need to be impounded..`
        - Non-programmable, non-graphing calculator(Class II)
        - other tools? screwdrivers(flat and cross), wrench, multimeter, triangle scale ruler, stopwatch
    - Practice log
        - Notes
- Track and Run

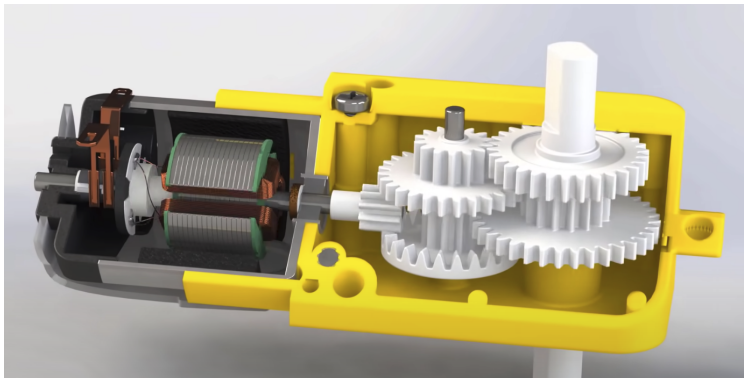
- obstacles - 8 1.5 × 3.5 × 16 (inches) for gate zones. 3 gate zones for regionals.
    - Nathan, get them ready by next week
- Test on different types or floor, tile. How the robots perform with different frictions?
    - rules: `smooth, level, and hard surface`
    - Booked Canyon MPR room where we run GullSO RT event, 12/1 and 12/8 evening.

Reminders

- Do not share with anyone how we are building our robots, what sensors, motors we are using.
- Do not share photos, videos
- At competitions, parents can only take photos, video for their own teams.

520/513 Brushed DC electric motors

vs



	MC520P60_12V MG513P60_12v	JGB37-520 110 RPM	JGB37-520 76 RPM	
Voltage(V)	12V	12V	12V	
Ratio	60:1	90:1	131:1	
Speed(No load) (RPM)	~180	~110	~76	
Rated current(A)	0.36	0.65	0.65	
Stall current(A)	3.2	2.4	2.4	
Rated torque(kg.cm)	2	3	3.9	
Stall torque(kg.cm)	9	12	15.6	
Encoder PPR(Pulses per revolution)	13	11	11	
CPR(Count per revolution)	3120	3860	5764	

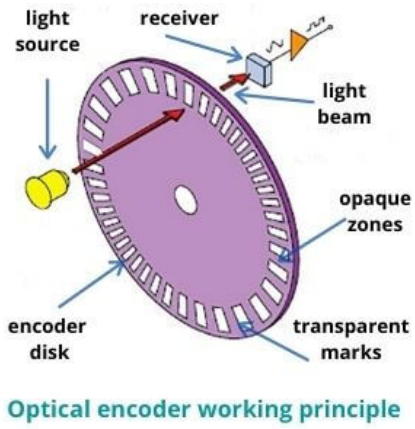
PINs

PIN	Wire	connection
Motor VCC(3-12V)	red	?
Encoder GND	black	
Encoder channel A	yellow	
Encoder channel B	green	
Encoder VCC(5V)	blue	
Motor GND	white	

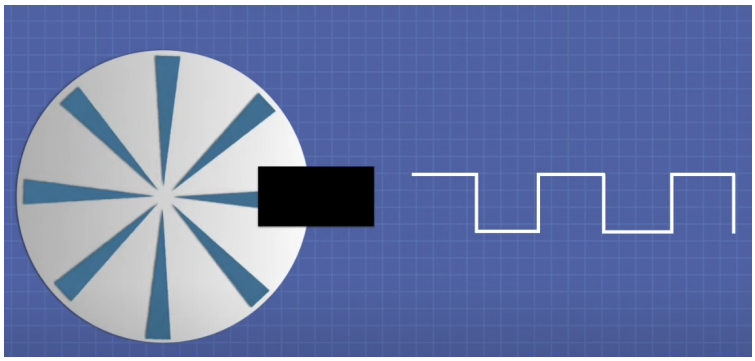
Encoders

Optical encoder

https://www.encoder.com/article-what-is-an-optical-encoder



Optical encoder working principle

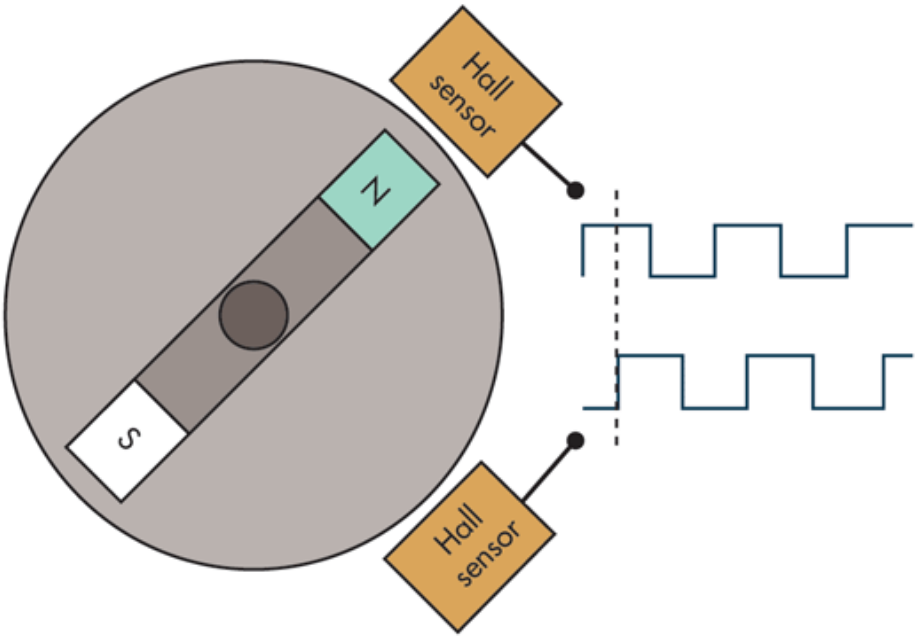


LM393 IR(Infrared) speed sensor

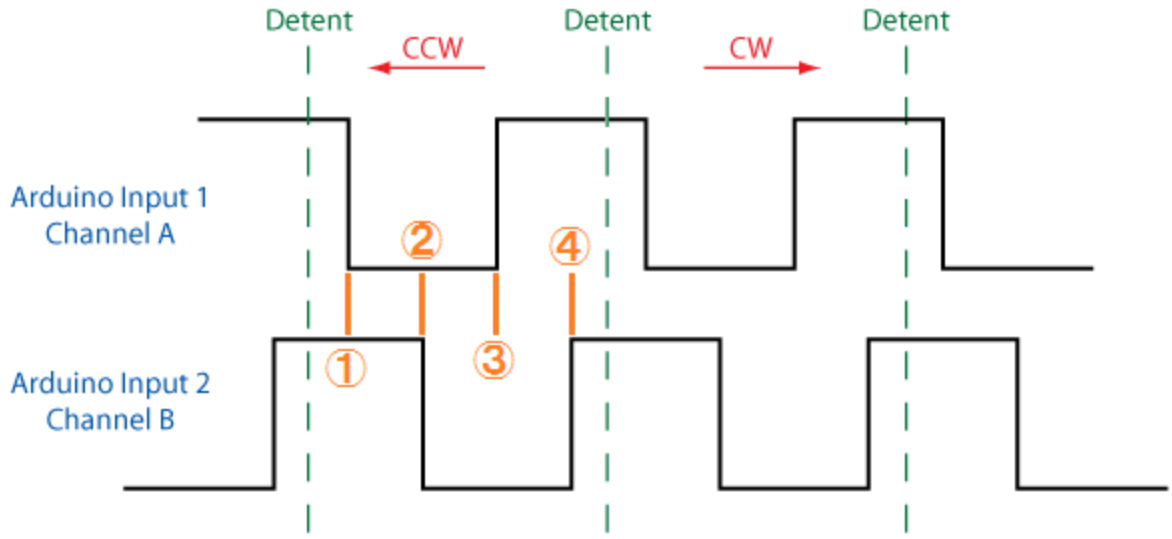
https://youtu.be/oQQpAACa3ac?si=4Ap46d3R4djw2fxB

Magnetic encoder

Hall-effect sensors



- Hall effect sensors measure the presence of the presence of magnetic fields.
  - 2 channels
- North pole - output a HIGH digital signal
- South pole - output a LOW digital signal
- PPR - pulses per revolution
  - 520 DC motors - 13 or 11
  - gear ratio - 60:1 , CPR = 13 * 60 = 780
  - C = 20cm
  - how many pulses we will see if we need to travel 100 cm?
    - num of pulses = 100/20 * 780 = 3900.
- CPR - counts per revolution - 4 X PPR



https://curiores.com/positioncontrol

https://www.electroniclinic.com/arduino-dc-motor-speed-control-with-encoder-arduino-dc-motor-encoder/

Questions

- Why are there 2 Hall-effect sensors? Directions.
- How many encoder pulses per channel are generated when the wheel spins a full rotation?

Program to measure the pulse count and RPM

Polling model

- keep refreshing, frequently check
- data could be missed

Hardware interrupts

- monitor the changes. Notification, triggers. The hardware signal triggers a function inside the Arduino code, it stops the main execution of your program, after the triggered function is done, your main execution resumes.

https://roboticsbackend.com/arduino-interrupts/

https://youtu.be/wIcC8-g9Lnw?si=-yDfqr2-0e2IOuCO

Other interrupts https://dronebotworkshop.com/interrupts/

- Pin change
- Timer

Arduino interrupt pins

- Use digital pins - UNO: D2 and D3
- Attach a function(ISR) to be triggered whenever a change on the pin is made.
- `attachInterrupt(digitalPinToInterrupt(pin), ISR, mode)` `(recommended)`
- https://www.arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/
- declare as `volatile` for the variables that are modified within the function. https://www.arduino.cc/reference/en/language/variables/variable-scope-qualifiers/volatile/

PID control

Closed loop control https://youtu.be/O-OqgFE9SD4?si=XBJIw0jzVmg6srZf

PID control - introduction

- https://youtu.be/UR0hOmjaHp0?si=Boe4_-_5tubR7wCi
- https://youtu.be/wkfEZmsQqiA?si=b-Cr5zFazEXZaWRj

Simple example of PID control https://youtu.be/XfAt6hNV8XM?si=h4hEJJ386-bB6QPU

From Nathan:

The first document is an introduction to basic integrals and derivatives:

https://docs.google.com/document/d/1raXDfm_P-H5dPyIwTI89mRtF18nLAoRa9gCDm7OU9ck/edit

And the second document is an introduction to PID loops

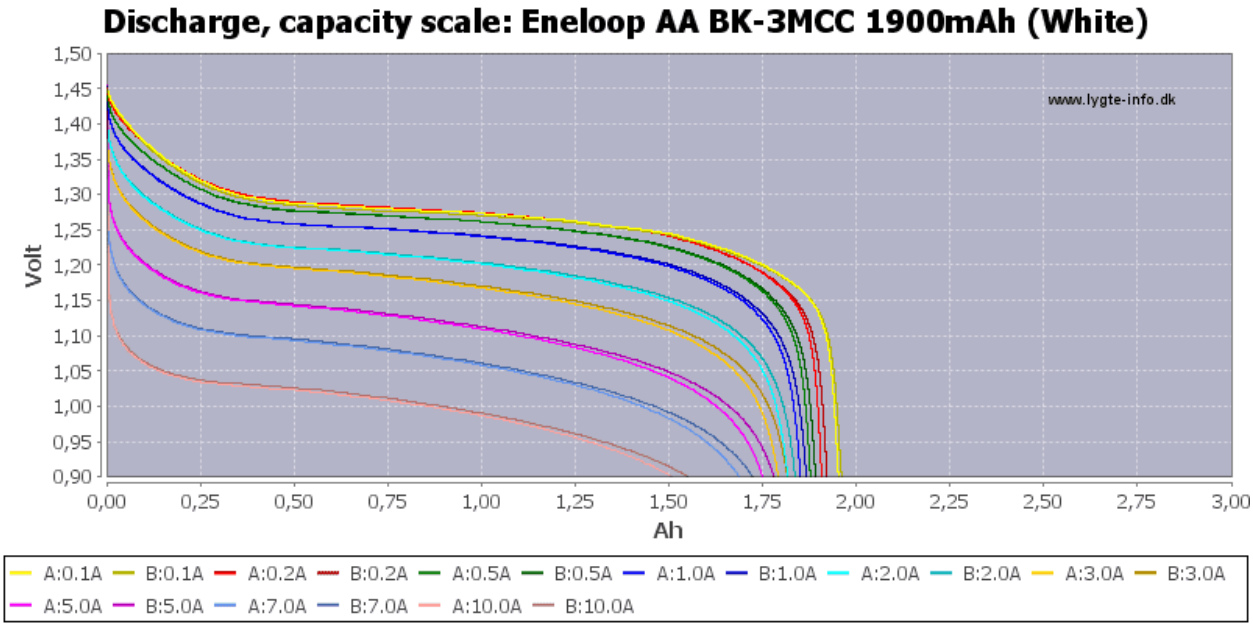
https://docs.google.com/document/d/1D61dinGqP__CHzRfc9yNugnS9K-JTa395_r0qP7L5W8/edit

You will need to read the first document to understand what is going on in the second one. Take your time, this decently complex stuff, ask if you have any questions, and good luck!

Week 9

AA rechargeable batteries

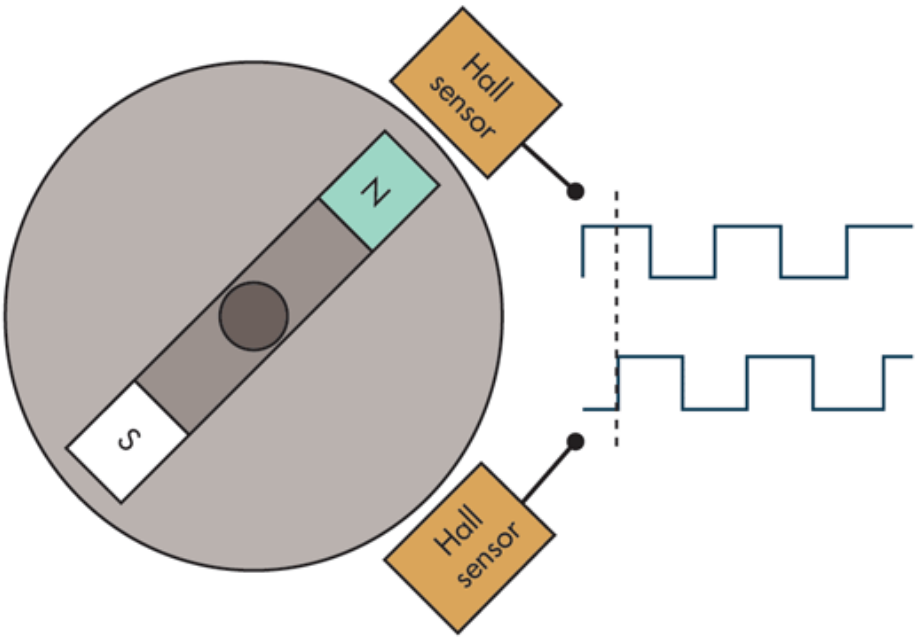
Discharge, capacity scale



- Fully charged: ~1.4V, quickly drops to 1.25V
- Stable voltage - 1.25V - 1.275V, total: 7.5V - 7.65V

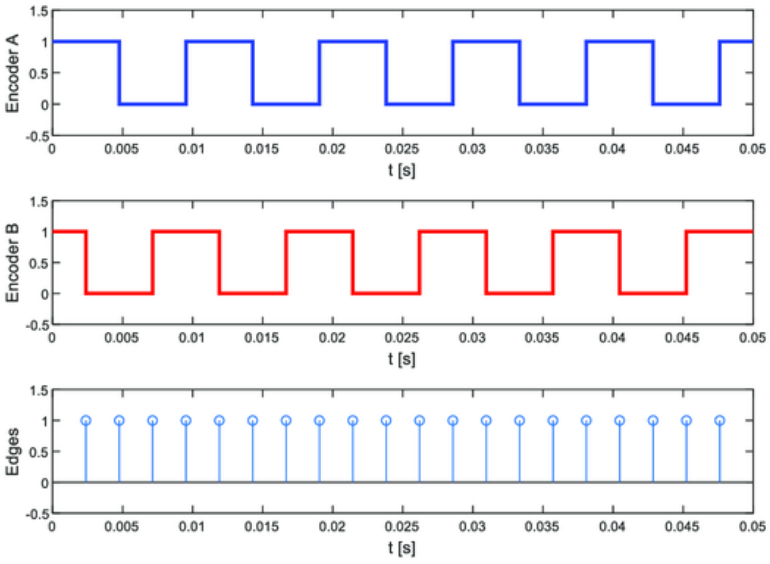
Magnetic Encoders

Hall-effect sensors

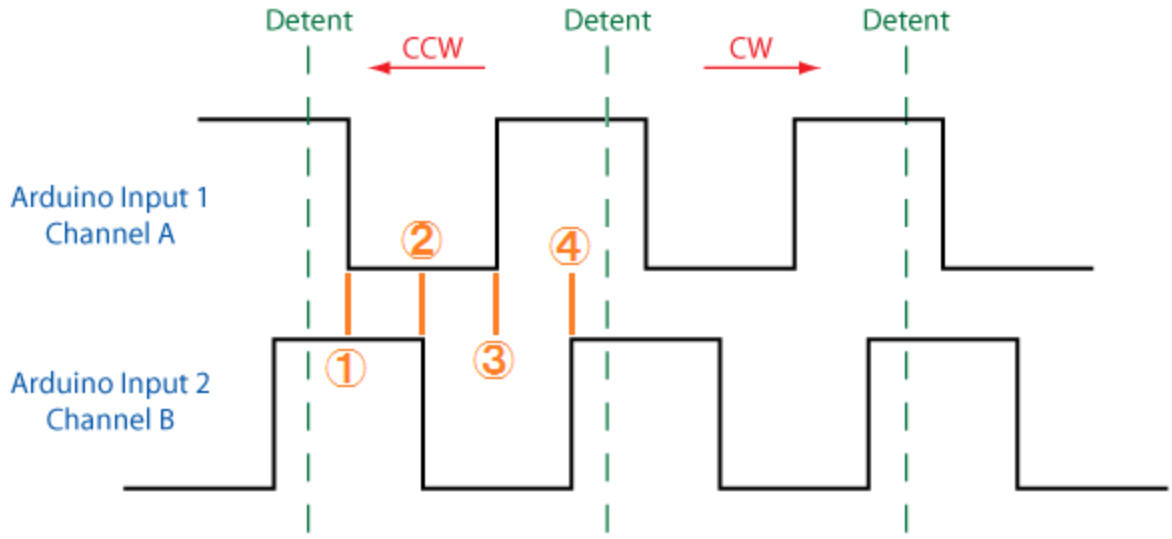

Hall sensor
N
S
Hall sensor


S
N
N
S
S
N
H A
H B

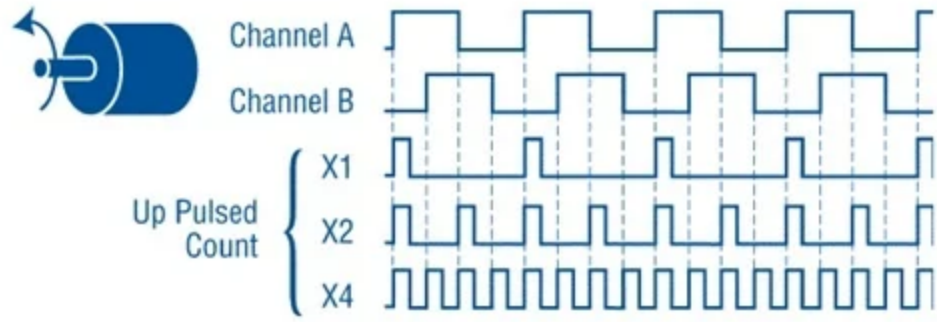
(a)


Encoder A
Encoder B
Edges
t [s]
1.5
1
0.5
0
-0.5
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0.045
0.05

(b)



1. A: 1→0, B: 1
2. A: 0, B1→0
3. A: 0→1, B: 0
4. A: 1, B: 0→1



- Hall effect sensors measure the presence of the presence of magnetic fields.
  - 2 hall-effect sensors, placed at 90″ phase offset
  - North pole - output a HIGH digital signal
  - South pole - output a LOW digital signal

    - Logical quadrature output signals.
- PPR - pulses per revolution(encoder magnetic disk)
    - Multi-pole magnetic disk
    - 520/513 DC motors - 13 or 11 poles
- CPR - counts per revolution - 4 X PPR
    - 4 distinct states, quadrature output
    - Logical counts - CPR = PPR * 4
- Gear reduction DC motors - Counts per wheel revolution
    - CPR * Gear ratio

https://www.cuidevices.com/blog/what-is-encoder-ppr-cpr-and-lpr

https://curiores.com/positioncontrol

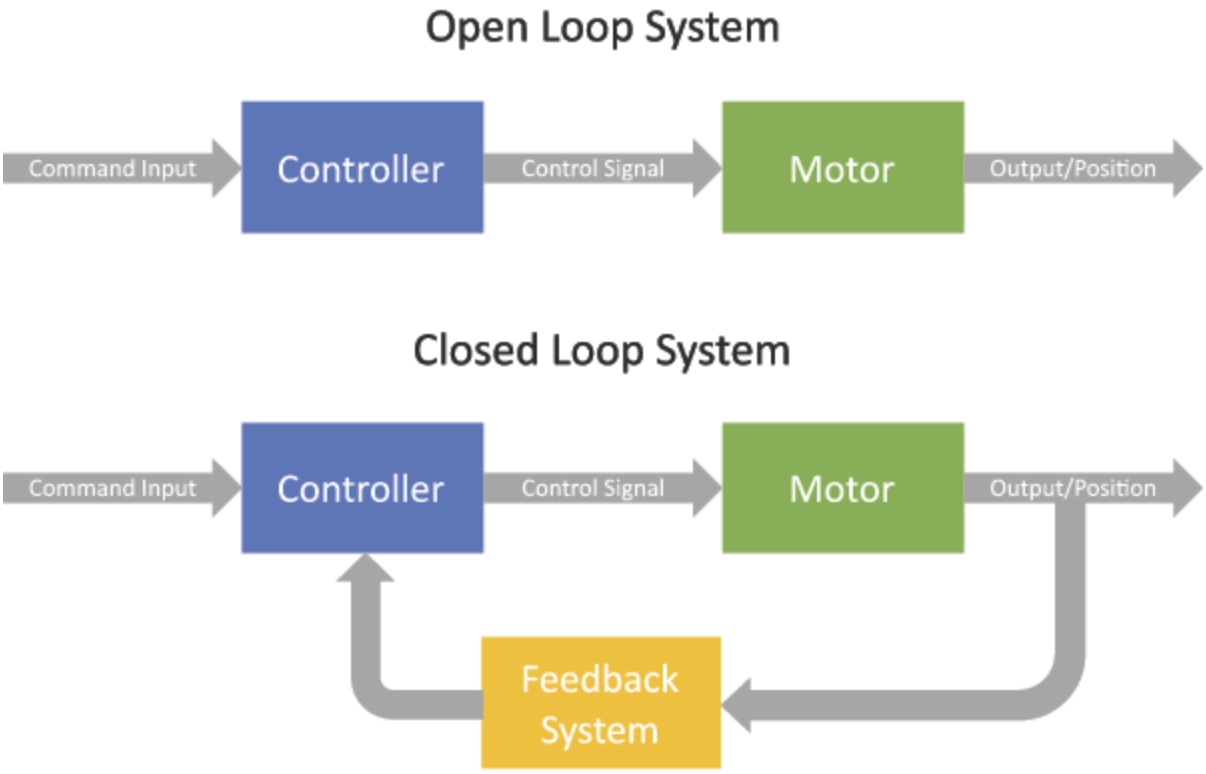
https://www.electroniclinic.com/arduino-dc-motor-speed-control-with-encoder-arduino-dc-motor-encoder/

520/513 Brushed DC electric motors

	MC520P60_12V MG513P60_12v	JGB37-520 110 RPM	JGB37-520 76 RPM	N20 6v 75 RPM
Voltage(V)	12V	12V	12V	6V
Ratio	60:1	90:1	131:1	210:1
Speed(No load) (RPM)	~180	~110	~76	
Rated current(A)	0.36	0.65	0.65	
Stall current(A)	3.2	2.4	2.4	
Rated torque(kg.cm)	2	3	3.9	
Stall torque(kg.cm)	9	12	15.6	

Encoder PPR(Pulses per revolution)	13	11	11	7
CPR(Counts per revolution)	52	44	44	28
CPWR(Counts per wheel revolution)	3120	3860	5764	5880

Controllers



Left motor - controller - travel to target location

- controller input: target pulse count, measured pulse count from encoder, `error = target pulse count - measured pulse count`
- motor output: measured pulse count
- controller output: PWM

Right motor - synchronize with left motor, drive straight or rotate in place

- controller input: target pulse count - 0, measured pulse count difference from left motor encoder, `error = 0 - measure pulse count difference`
- motor output: measured pulse count
- controller output: PWM that needs to be added to motor B

Arduino PID library

https://github.com/br3ttb/Arduino-PID-Library/tree/master

$$\text{Output} = K_P e(t) + K_I \int e(t)\,dt + K_D \frac{d}{dt} e(t)$$

$$\text{Where : } e = \text{Setpoint - Input}$$

Left motor(A) PID controller

```
/*working variables*/
unsigned long lastTime;
double Input, Output, Setpoint;
double errSum, lastErr;
double kp, ki, kd;
void Compute()
{
   /*How long since we last calculated*/
   unsigned long now = millis();
   double timeChange = (double)(now - lastTime);

   /*Compute all the working error variables*/
   double error = Setpoint - Input;
   errSum += (error * timeChange);
   double dErr = (error - lastErr) / timeChange;

   /*Compute PID Output*/
   Output = kp * error + ki * errSum + kd * dErr;
```

```
   /*Remember some variables for next time*/
   lastErr = error;
   lastTime = now;
}

void SetTunings(double Kp, double Ki, double Kd)
{
   kp = Kp;
   ki = Ki;
   kd = Kd;
}
```

Left motor(A) PID controller

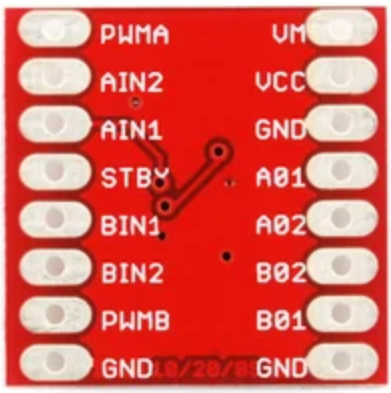
```
KP, KI, KD are constants

set point = target pulse count

error = target pulse count - current pulse count

integral = integral + error

derivative = error - last error

output = (error * KP) + (integral * KI) + (derivative * KD)
```

Right motor(B) PID controller

TB6612FNB motor driver

https://www.sparkfun.com/datasheets/Robotics/TB6612FNG.pdf

https://www.instructables.com/Using-the-Sparkfun-Motor-Driver-1A-Dual-TB6612FNG-/



- Logic supply voltage (VCC) of 2.7-5.5 VDC
- Motor supply voltage (VM) up tp 15VDC
- Current of 1.2A constant per channel (3.2A peak)
- Drives up to 2 motors