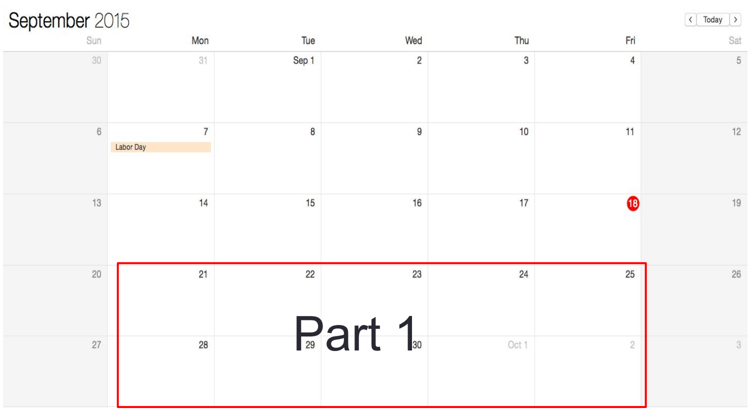
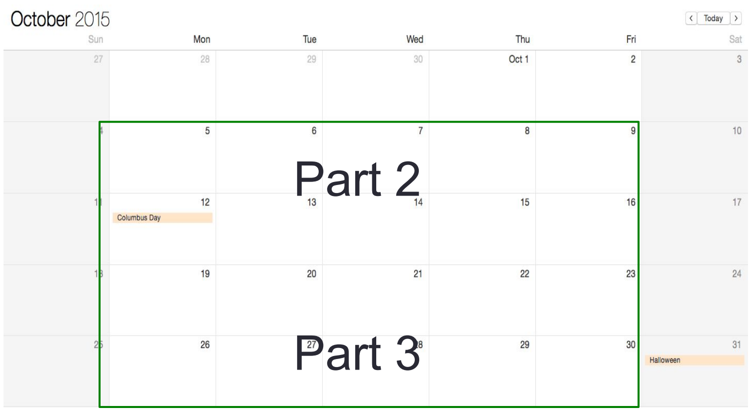
CS123 - I/O extensions

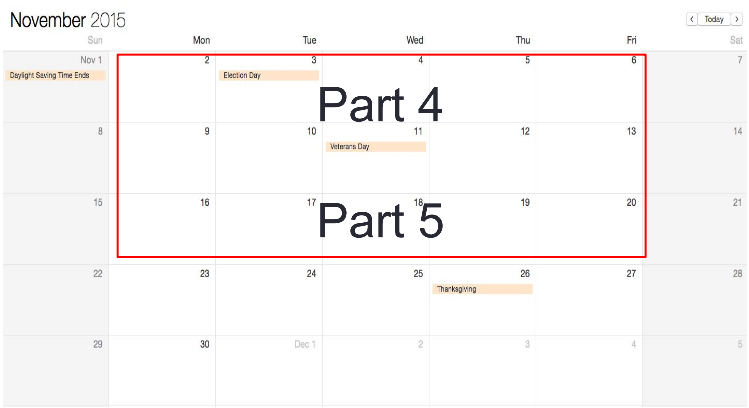
Programming Your Personal Robot

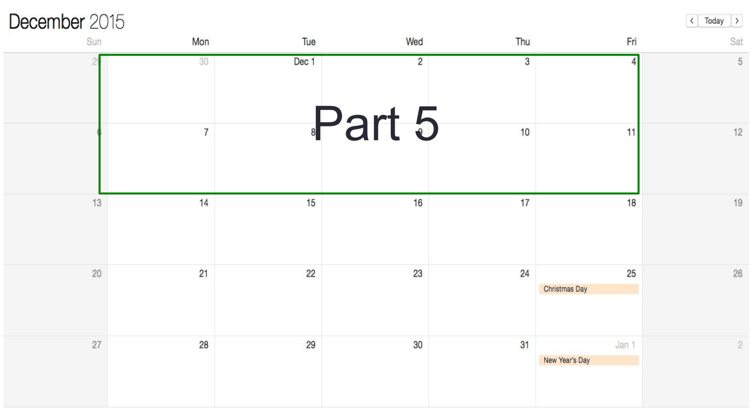
Kyong-Sok "KC" Chang, David Zhu
Fall 2015-16

Calendar









KC
Teaching

David
Teaching

Syllabus

- Part 1 - Communicating with robot (2 weeks)
  - BLE communication and robot API
- Part 2 - Event Driven Behavior (2 weeks)
  - Finite State Machine (Behavior Tree)
- Part 3 - Reasoning with Uncertainty (2 weeks)
  - Dealing with noisy data, uncertainty in sensing and control
- Part 4 - Extending the robot (1 weeks)
  - I/O extensions: digital, analog, servo, pwm, etc
- Part 5 – Putting it together (including UI/UX) (3 weeks)
  - Design and implement of final (group) project
  - Encourage you to go “above and beyond”

BLE: GATT Profile: Hamster

OTA DFU service(Over-The-Air Device Firmware Update)

- service used by host for wireless firmware update
- sending DFU packet: DFU Packet Characteristic and Control Point Characteristic
- BLE Service UUID: 00001530

```
PrimaryService, Value: 00-18, Generic Access (0x1800)
  CharacteristicDeclaration, Value: 0A-03-00-00-2A, Properties: Read, Write, Characteristic UUID: 0x2A00
    DeviceName, Value: 48-61-6D-73-74-65-72, DeviceName: Hamster
  CharacteristicDeclaration, Value: 02-05-00-01-2A, Properties: Read, Characteristic UUID: 0x2A01
    Appearance, Value: 02-04, Appearance: 0x0402
  CharacteristicDeclaration, Value: 02-07-00-04-2A, Properties: Read, Characteristic UUID: 0x2A04
    SlavePreferredConnectionParameters, Value: 10-00-10-00-00-00-32-00, MinConnInterval: 0x0010, MaxConnInterval: 0x0010, SlaveLatency: 0x0000, SupervisionTimeoutMultiplier: 0x0032
PrimaryService, Value: 01-18, Generic Attribute (0x1801)
PrimaryService, Value: 0A-18, Device Information (0x180A)
  CharacteristicDeclaration, Value: 02-0B-00-29-2A, Properties: Read, Characteristic UUID: 0x2A29
    Manufacturer Name String, Value: 52-6F-62-6F-6D-61-74-69-6F-6E
  CharacteristicDeclaration, Value: 02-0D-00-26-2A, Properties: Read, Characteristic UUID: 0x2A26
    Firmware Revision String, Value: 31-2E-32
PrimaryService, Value: 66-9A-0C-20-00-08-E2-A5-E3-11-80-9C-01-90-00-00, 0x00009001-9C80-11E3-A5E2-0800200C9A66
  CharacteristicDeclaration, Value: 14-10-00-66-9A-0C-20-00-08-E2-A5-E3-11-80-9C-0A-90-00-00, Properties: WriteWithoutResponse, Notify, Characteristic UUID: 0x0000900A-9C80-11E3-A5E2-0800200C9A66
    UUID: 0000900A-9C80-11E3-A5E2-0800200C9A66, Value: 00-00-10-00-1B-00-65-14-02-BF-12-BF-C7-7F-00-00-3C-F9-F8-00
    CharacteristicUserDescription, Value: 53-65-6E-73-6F-72-73, UserDescription: Sensors
    ClientCharacteristicConfiguration, Value: 00-00, CharacteristicConfigurationBits: None (0x0000)
PrimaryService, Value: 66-9A-0C-20-00-08-E2-A5-E3-11-80-9C-00-A0-00-00, 0x0000A000-9C80-11E3-A5E2-0800200C9A66
  CharacteristicDeclaration, Value: 04-15-00-66-9A-0C-20-00-08-E2-A5-E3-11-80-9C-06-A0-00-00, Properties: WriteWithoutResponse, Characteristic UUID: 0x0000A006-9C80-11E3-A5E2-0800200C9A66
    UUID: 0000A006-9C80-11E3-A5E2-0800200C9A66, Value: 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
    CharacteristicUserDescription, Value: 45-66-66-65-63-74-6F-72-73, UserDescription: Effectors
PrimaryService, Value: 23-D1-BC-EA-5F-78-23-15-DE-EF-12-12-30-15-00-00, DFU (0x00001530-1212-EFDE-1523-785FEABCD123)
  CharacteristicDeclaration, Value: 04-19-00-23-D1-BC-EA-5F-78-23-15-DE-EF-12-12-32-15-00-00, Properties: WriteWithoutResponse, Characteristic UUID: 0x00001532-1212-EFDE-1523-785FEABCD123
    DFU Packet, (No values read)
  CharacteristicDeclaration, Value: 18-1B-00-23-D1-BC-EA-5F-78-23-15-DE-EF-12-12-31-15-00-00, Properties: Write, Notify, Characteristic UUID: 0x00001531-1212-EFDE-1523-785FEABCD123
    DFU Control Point, (No values read)
    ClientCharacteristicConfiguration, Value: 00-00, CharacteristicConfigurationBits: None (0x0000)
  CharacteristicDeclaration, Value: 02-1E-00-23-D1-BC-EA-5F-78-23-15-DE-EF-12-12-34-15-00-00, Properties: Read, Characteristic UUID: 0x00001534-1212-EFDE-1523-785FEABCD123
    UUID: 00001534-1212-EFDE-1523-785FEABCD123, Value: 01-00
```

Logistics

- Getting new PSD Scanner
  - Update Hamster firmware
    - Over-The-Air Device Firmware Update (DFU)
    - nRF Toolbox App
  - Install the hardware
  - Sign-up sheet
- TA sessions (office hours): this week
  - Location: Gates B21 (Th: Huang basement)
  - Time: M:2~4pm, Tu:2~4pm, W:12:30-2:30pm, Th:2~4pm
- Lab reserved for CS123: this week
  - MTuW: 12~6pm, ThF: ?? @ Gates B21
- My office hours (KC)
  - Tues and Thurs: 12:30-2:00pm @ Gates B21

Humanoids 2015: Workshop

Seoul, Korea, November 3-5, 2015

Title	Organizers
Towards Intelligent Social Robots – Current Advances in Cognitive Robotics	Aly Amir, ENSTA ParisTech, France Griffiths Sascha, Queen Mary University, England Stramandinoli Francesca, IIT, Italy Tapus Adriana, ENSTA ParisTech, France Nori Francesco, IIT, Italy
Can we build Baymax?: Soft Robotics and Safe Human-Robot Interaction in Humanoids	Kim Joohyung, Disney Research, USA Yamane Katsu, Disney Research, USA Atkeson Christopher, Carnegie Mellon Univ., USA Park Yong-Lae, Carnegie Mellon University, USA Tsagarakis Nikos, Istituto Italiano di Tecnologia, Italy
Proprioceptive and Exteroceptive Data Fusion for State Estimation and Whole-Body Control of Humanoid Robots	Moro Federico Lorenzo, IIT, Italy Kanoulas Dimitrios, IIT, Italy Jaeheung Park, Seoul National University, Korea Sentis Luis, University of Texas at Austin, USA
Human locomotion understanding for the design and control of next generation of humanoids and assistive devices	Demircan Emel, California State Univ. Long Beach, USA Kuli ć Dana, University of Waterloo, Canada
Whole-Body Multi-Task Multi-Contact Humanoid Control	Khansari Mohammad, Stanford University, USA Menon Samir, Stanford University, USA Chung Shuyun, Stanford University, USA Khatib Oussama, Stanford University, USA
The 10th Workshop on Humanoid Soccer Robots	Behnke Sven, University of Bonn, Germany Lee Daniel D., University of Pennsylvania, USA Lau Nuno, University of Aveiro, Portugal Ramamoorthy Subramanian,Univ. of Edinburgh, UK
Reusable and Open-source Modules for Humanoid Robots	Kim Jong-Wook, Dong-A University, Korea Kuindersma Scott, Harvard University, U.S.A. von Stryk Oskar, Technische Univ. Darmstadt, Germany
What did we do for the Darpa Robotics Challenge?	Jun Ho Oh, KAIST, Korea

Outline

- Logistics
- Future robots: Humanoids
- Recap Part 3: Reasoning with Uncertainty
- Part 4: I/O extensions
  - Electricity
  - Analog vs. Digital
  - ADC
  - PWM
  - Hamster I/O Mode
- Final project

Recap: Reasoning with uncertainty

- Part 3-1: Robot Programming
  - Modeling
  - Localization
  - Planning
  - Execution
  - Reactive is not enough: better knowledge of environment
- Part 3-2:
  - Localization
    - Relative (Internal): dead reckoning
    - Absolute (External): distance sensors (Geometric feature detection), IR, Landmark
  - Modeling Environment
    - Least Square (Fit): minimization

Recap: Reasoning with uncertainty

- Part 3-3
  - Motion Planning
    - Configuration Space: C-Space
    - Discretization: Visibility Graph, Voronoi Diagrams, Cell Decomposition
    - Cell Decomposition: Exact, estimate
  - Motion Planning with Uncertainty
    - Landmarks
    - Preimage backchaining
- Part 3-4
  - Search
    - Uninformed (Blind): BFS, DFS
    - Informed (Heuristic): Evaluation function: Dijkstra's, A*
    - Potential Field
  - Motion Control: Motion Primitives

Outline: Part 4 & 5

- Part 4: I/O extensions
  - Electricity, Analog, Digital, PWM, Servo
  - More sensors and effectors: PSD Scanner
  - Better knowledge of environment
    - Filtering: low-pass
    - Modeling Environment
      - Least Square (Fit): minimization
  - Feedback control: line-tracing
- Part 5: Putting it together (Navigation)
  - Robot Programming
    - Modeling
    - Localization
    - Planning
    - Execution
    - UI / UX

I/O mode: Hamster

Sensors Service Packet format definition

	Details	Value from Robot	User converted value
0	**Version / Topology**	0 ~ 255	0 ~ 255
1	**Network ID**	0 ~ 255	0 ~ 255
2	**Command / Security**	0 ~ 255	0 ~ 255
3	**Signal Strength**	-128 ~ 0	-128 ~ 0 dBm
4	**Left Proximity**	0 ~ 255	0 ~ 255
5	**Right Proximity**	0 ~ 255	0 ~ 255
6	**Left Floor**	0 ~ 255	0 ~ 255
7	**Right Floor**	0 ~ 255	0 ~ 255
8	**Acc X High**	-32768 ~ 32767	-32768 ~ 32767
9	**Acc X Low**		
10	**Acc Y High**	-32768 ~ 32767	-32768 ~ 32767
11	**Acc Y Low**		
12	**Acc Z High**	-32768 ~ 32767	-32768 ~ 32767
13	**Acc Z Low**		
14	**Flag**		
15	**Light High or Temperature**	0 ~ 65535 -128 ~ 127	0 ~ 65535 Lux -40 ~ 88 °C
16	**Light Low or Battery**	0 ~ 255	0 ~ 100 %
17	**Input A**	0~255	0 ~ 255 (0 ~ 3.3 V)
18	**Input B**		
19	**Line Tracer State**	0 ~ 255	0 ~ 255

Effector Service Packet format Definition

	Data	Value to Robot	User input value
0	**Version / Topology**	0 ~ 255	0 ~ 255
1	**Network ID**	0 ~ 255	0 ~ 255
2	**Command / Security**	0 ~ 255	0 ~ 255
3	**Left Wheel**	-100 ~ +100 (+fwd, -bwd)	-100 ~ 100 %
4	**Right Wheel**		
5	**Left LED**	0 (off) ~ 7	0 (off) ~ 7
6	**Right LED**		
7	**Buzzer High**	0(off) 1 ~ 16777215	0(off) 1.00 Hz ~ 167.77215 KHz,
8	**Buzzer Middle**		
9	**Buzzer Low**		
10	**Musical Note**	1~88(piano key) 0(off)	1~88 0(off)
11	**Line Tracer Mode/Speed**	0x11 ~ 0x6A 0x0?(off)	0x11 ~ 0x6A 0x0?(off)
12	**Proximity IR Current**	0 ~ 7 (default 2)	0 ~ 7 (default 2)
13	**G-Range, Bandwidth**	0 ~ 3 (default 0), 0 ~ 8 (default 3)	0 ~ 3 (default 0), 0 ~ 8 (default 3)
14	**IO Mode(A, B)**	0 ~ 127	0 ~ 127
15	**Output A**	0 ~ 255	0 ~ 255
16	**Output B**		
17	**Wheel Balance**	-128 ~ 127	-128 ~ 127
18	**Input Pull**	0~16	
19			

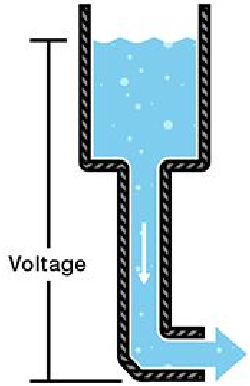
Ref. Kre8 Technology, Inc.

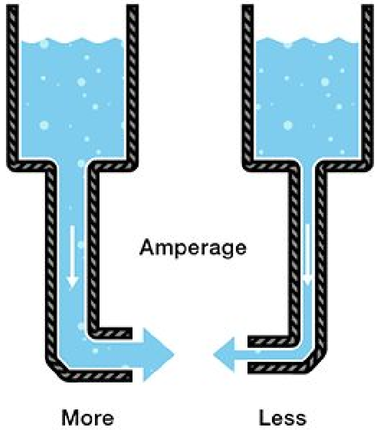
Electricity

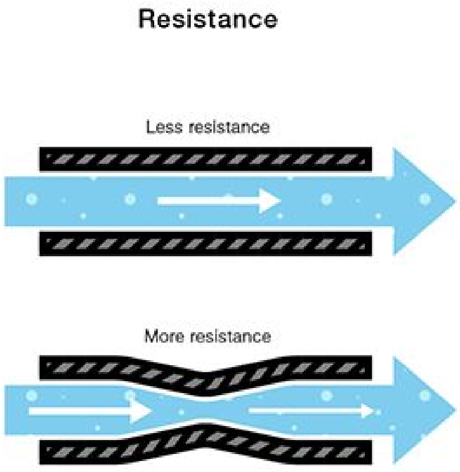
- Voltage, Current, Resistance
  - Electricity: movement of electrons (charge)
  - Voltage (V): difference in charge between two points.
  - Current (I): rate at which charge is flowing.
  - Resistance (R): material's tendency to resist the flow of c (current).
- Analogy: a water tank
  - Charge: Water amount
  - Voltage: Water pressure
  - Current: Water flow
  - Resistance: Water hose width
- Ohm's Law
  - $V = I * R$

Electricity: Analogy

- Analogy: a water tank
  - Charge: Water amount
  - Voltage (in volts): Water pressure
  - Current (in amperes): Water flow
  - Resistance (in ohms): Water hose width
- Ohm's Law: $V = I * R$





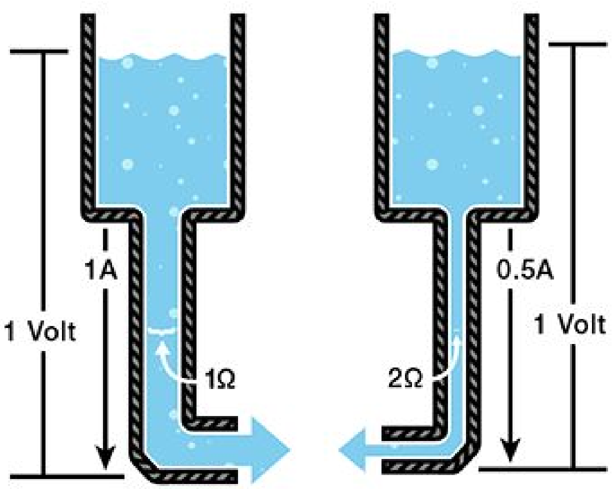


Ref. learn.sparkfun.com

Ohm's Law: *V = I * R*

1 V = 1 A * 1 Ω

= 0.5 A * 2 Ω

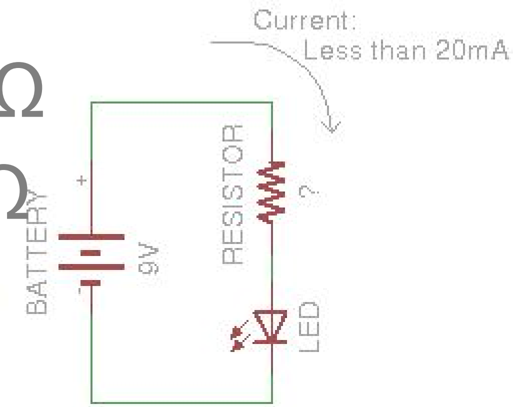


Ex) Resistance?

R = V / I = 9 V / 20 mA

= 450 Ω

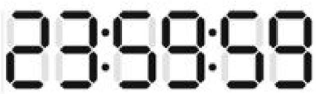
R > 450 Ω



Ref. learn.sparkfun.com

Analog vs. Digital

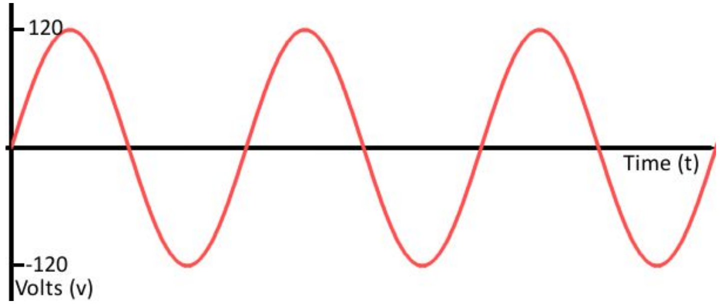
- We live in an analog world with both analog and digital devices (signals, Input/output).
- Analog: infinite set of values (infinite resolution)
- Digital: finite/discrete set of values

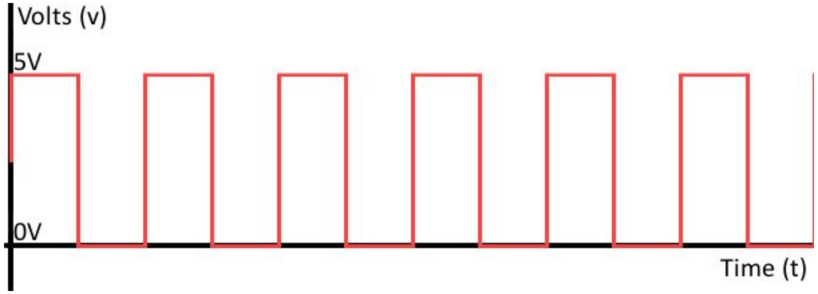


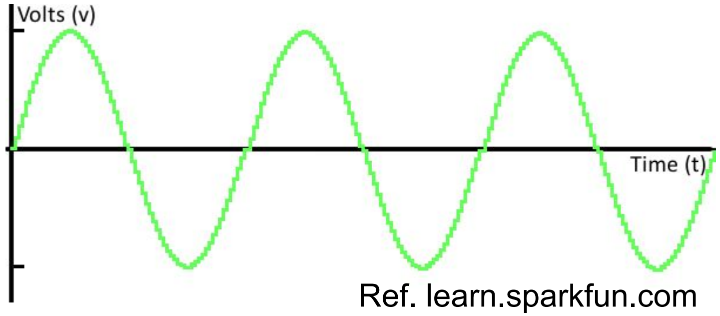
Ref. learn.sparkfun.com

Analog vs. Digital: Signals

- Signals
  - time-varying “quantities” which convey some sort of information
  - for EE: a voltage that’s changing over time
  - passed between devices in order to send and receive information
  - Analog: smooth, continuous
    - composite video (RCA)
    - volume knob (variable resistor)
  - Digital: discrete steps, square
    - HDMI, Serial







Ref. learn.sparkfun.com

Analog to Digital Converter (ADC)

- ADC
  - converts an analog voltage on a pin to a digital number
  - converting from the analog world to the digital world
  - to use electronics to interface to the analog world
- Relating ADC Value to Voltage
  - The ADC reports a *ratiometric value*

$$\frac{Resolution\ of\ the\ ADC}{System\ Voltage} = \frac{ADC\ Reading}{Analog\ Voltage\ Measured}$$

- Hamster
  - System Voltage = 3.7 V
  - Resolution of the ADC = 8 bit = 255 (= 0xFF)
  - Input (Analog): Voltage measured
  - Output (Digital): ADC Reading = Input * 255 / 3.7

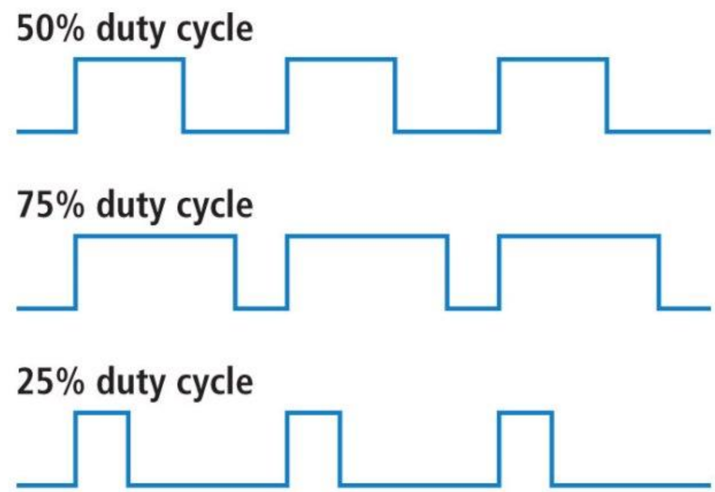
Ref. learn.sparkfun.com

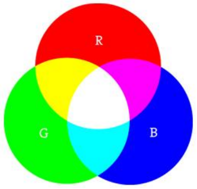
Pulse Width Modulation (PWM)

- Analog Circuit and Control
  - Most of the fundamental electronic components
    - resistors, capacitors, inductors, diodes, transistors, and operational amplifiers – inherently analog
  - Analog Circuit: built with a combination of solely analog components
  - Direct control of voltage/current: intuitive and simple
  - Sensitive to noise: infinite resolution
  - Drift over time: difficult to tune
  - Large, heavy, expensive: old home stereo equipment
  - Inefficient power consumption: physically hot
- Digital Control: PWM
  - Controlling analog circuits digitally: cheaper and consumes less power
  - allows to vary how much time the signal is high in an analog fashion
  - controlling analog circuits with a microprocessor's digital outputs.
  - In a nutshell, PWM is a way of digitally encoding analog signal level

Pulse Width Modulation (PWM)

- Duty Cycle
  - on-time: when signal is high
  - duty cycle: amount of on-time
  - measured in % over a period
  - Ex) 5V
    - 50% duty cycle: 2.5V
- Examples
  - RGB LED
    - all equal duty cycle: white
  - Servo motors
    - frequency: 50 Hz waveform
    - duty cycle: 5~10%
    - 1.0 ms pulse: 0 deg
    - 1.5 ms pulse: 90 deg
    - 2.0 ms pulse: 180 deg





Ref. learn.sparkfun.com

I/O mode: Hamster

Sensors Service Packet format definition

	Details	Value from Robot	User converted value
0	**Version / Topology**	0 ~ 255	0 ~ 255
1	**Network ID**	0 ~ 255	0 ~ 255
2	**Command / Security**	0 ~ 255	0 ~ 255
3	**Signal Strength**	-128 ~ 0	-128 ~ 0 dBm
4	**Left Proximity**	0 ~ 255	0 ~ 255
5	**Right Proximity**	0 ~ 255	0 ~ 255
6	**Left Floor**	0 ~ 255	0 ~ 255
7	**Right Floor**	0 ~ 255	0 ~ 255
8	**Acc X High**	-32768 ~ 32767	-32768 ~ 32767
9	**Acc X Low**		
10	**Acc Y High**	-32768 ~ 32767	-32768 ~ 32767
11	**Acc Y Low**		
12	**Acc Z High**	-32768 ~ 32767	-32768 ~ 32767
13	**Acc Z Low**		
14	**Flag**		
15	**Light High or Temperature**	0 ~ 65535 -128 ~ 127	0 ~ 65535 Lux -40 ~ 88 °C
16	**Light Low or Battery**	0 ~ 255	0 ~ 100 %
17	**Input A**	0~255	0 ~ 255 (0 ~ 3.3 V)
18	**Input B**		
19	**Line Tracer State**	0 ~ 255	0 ~ 255

Effector Service Packet format Definition

	Data	Value to Robot	User input value
0	**Version / Topology**	0 ~ 255	0 ~ 255
1	**Network ID**	0 ~ 255	0 ~ 255
2	**Command / Security**	0 ~ 255	0 ~ 255
3	**Left Wheel**	-100 ~ +100 (+fwd, -bwd)	-100 ~ 100 %
4	**Right Wheel**		
5	**Left LED**	0 (off) ~ 7	0 (off) ~ 7
6	**Right LED**		
7	**Buzzer High**	0(off) 1 ~ 16777215	0(off) 1.00 Hz ~ 167.77215 KHz,
8	**Buzzer Middle**		
9	**Buzzer Low**		
10	**Musical Note**	1~88(piano key) 0(off)	1~88 0(off)
11	**Line Tracer Mode/Speed**	0x11 ~ 0x6A 0x0?(off)	0x11 ~ 0x6A 0x0?(off)
12	**Proximity IR Current**	0 ~ 7 (default 2)	0 ~ 7 (default 2)
13	**G-Range, Bandwidth**	0 ~ 3 (default 0), 0 ~ 8 (default 3)	0 ~ 3 (default 0), 0 ~ 8 (default 3)
14	**IO Mode(A, B)**	0 ~ 127	0 ~ 127
15	**Output A**	0 ~ 255	0 ~ 255
16	**Output B**		
17	**Wheel Balance**	-128 ~ 127	-128 ~ 127
18	**Input Pull**	0~16	
19			

Ref. Kre8 Technology, Inc.

PWM: Hamster

Sensors Service Packet format definition

	Details	Value from Robot	User converted value
0	**Version / Topology**	0 ~ 255	0 ~ 255
1	**Network ID**	0 ~ 255	0 ~ 255
2	**Command / Security**	0 ~ 255	0 ~ 255
3	**Signal Strength**	-128 ~ 0	-128 ~ 0 dBm
4	**Left Proximity**	0 ~ 255	0 ~ 255
5	**Right Proximity**	0 ~ 255	0 ~ 255
6	**Left Floor**	0 ~ 255	0 ~ 255
7	**Right Floor**	0 ~ 255	0 ~ 255
8	**Acc X High**	-32768 ~ 32767	-32768 ~ 32767
9	**Acc X Low**		
10	**Acc Y High**	-32768 ~ 32767	-32768 ~ 32767
11	**Acc Y Low**		
12	**Acc Z High**	-32768 ~ 32767	-32768 ~ 32767
13	**Acc Z Low**		
14	**Flag**		
15	**Light High or Temperature**	0 ~ 65535 -128 ~ 127	0 ~ 65535 Lux -40 ~ 88 ºC
16	**Light Low or Battery**	0 ~ 255	0 ~ 100 %
17	**Input A**	0~255	0 ~ 255 (0 ~ 3.3 V)
18	**Input B**		
19	**Line Tracer State**	0 ~ 255	0 ~ 255

Sensor packet: 17th and 18th bytes: Input

Ref.12) Input A/B

ADC mode) Analog to Digital Converter mode (Measuring analog voltage)

Active only if the Effectors' IO Mode value == 0

Formula) Volt = 3.3 * ADC level / 255 (volt)

DI mode) Digital Input mode (Measuring digital input)

Active only if the Effectors' IO Mode value == 1

Formula) 1 if input voltage >= 0.5, 0 otherwise

Effector packet: 14th byte: External IO Mode

Port A and Port B are independent of each other.

bit	7	6	5	4	3	2	1	0
	Port A (0~127)				Port B (0~127)			
	ADC mode, 0x0				ADC mode, 0x0			

0x00 ADC (Analog-to-Digital)
0x01 DI (Digital Input)
0x08 SERVO (Analog Servo Control)
0x09 PWM (Digital-to-Analog)
0x0A DO (Digital Output)

Ref. Kre8 Technology, Inc.

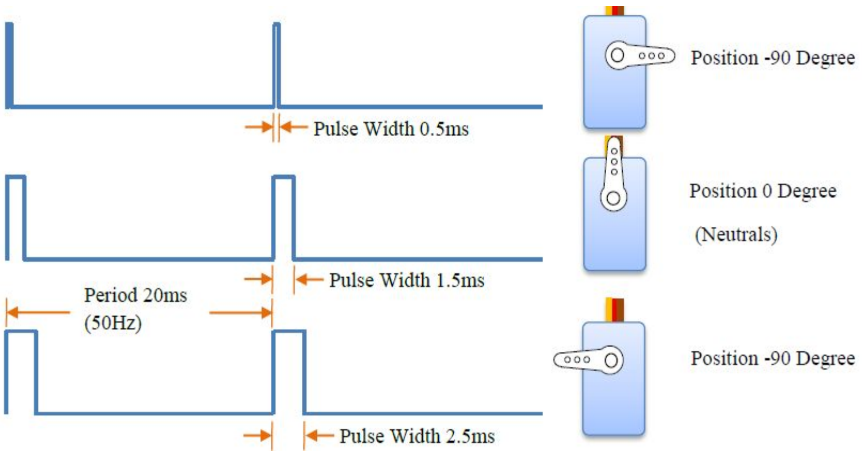
PWM: Hamster

Effector packet: 14th byte: External IO Mode

Port A and Port B are independent of each other.

bit	7	6	5	4	3	2	1	0
	Port A (0~127)				Port B (0~127)			
	ADC mode, 0x0				ADC mode, 0x0			

0x00 ADC (Analog-to-Digital)
0x01 DI (Digital Input)
0x08 SERVO (Analog Servo Control)
0x09 PWM (Digital-to-Analog)
0x0A DO (Digital Output)



ADC (Analog-to-Digital) Mode: 0x00
Measures input voltage with 8-bit ADC.
Max input voltage is ~3.7volt → 255(0xFF)

DI (Digital Input) Mode: 0x01
Detect input voltage to either 0 or 1.
1 if input voltage > 3.7/2 (~1.8 volt)
0 otherwise

SERVO (Analog servo) Output Mode: 0x08
Generating PWM signal(mode = 8) for external Servo control
* If value == 0(off) → no pulse
* If value > 180, pulse width limits to 2.5 ms

Port A	1 ~ 180	0(off), 90(center)	1deg=1.0ms, 90deg=1.5ms, 180deg=2.0ms
Port B	1 ~ 180	0(off), 90(center)	1deg=1.0ms, 90deg=1.5ms, 180deg=2.0ms

PWM (Digital-to-Analog) Output Mode: 0x09
Output: PWM signal's Duty value
If value > 100(0x64), output is 1 and PWM pulse period is 20 msec.
Therefore, if Duty value is 50%(50, 0x32), output is 0 for 10 msec, then output is 1 for the next 10msec.

DO (Digital Output) Mode: 0x0A
If value is not 0, output is 'high'.

Ref. Kre8 Technology, Inc.

Assignment#4

Make the scanning sensor work.

Analyze and model the sensor values.

-- PSD Sensor: IR

-- Motor: Servo

Reference and Reading

- "Voltage, Current, Resistance, and Ohm's Law" by learn.sparkfun.com
- "Analog vs. Digital" by learn.sparkfun.com
- "Analog to Digital Conversion" by learn.sparkfun.com
- "Pulse-width Modulation" by learn.sparkfun.com
- "Hamster Manual" by Kre8 Technology, Inc.