Project 4: Line Following Robot

- Project 4 content
- Suggested Steps

Project 4 Content

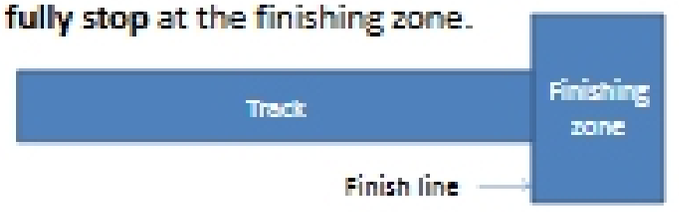
* Goals of this project:
  - Integrating all what you learned from the course together
* Minimal Hardware:
  - PIC32 Kit board, Robot Kit Frame, PmodSSD, Pmod microphone
  - Pmod IR sensor, IR sensor*4
  - R/C Servo Control pmod, R/C Servo*2
  - Oscilloscope (for debugging)
* Inputs:
  - Microphone, IR Sensor, one on-board button
* Outputs:
  - Four on-board LEDs
  - One external PmodSSD
  - Two R/C Servos

Robot functions

- Follow a black stripe on a white background.
- Make turns based on the black stripe path.
- Calculate how far it travels and display it on the SSD in inches.
- Make sure it is accurate, robust, and fast! (Yes, Robot Wars!)
- Any customization in robot functionality is welcome and may bring additional points! (Did we mention Robot Wars?)

Line following rules

- Two groups will compete on two identical, parallel tracks. The track assignment is random.
  - There will be both sharp and curve turns on the track. The smallest angle will be around 90 degree.
- Starting signal: a sound.
  - The robot stays in the start line. As soon as it detects the starting sound, it will start going forward.
- Ending signal: a finishing zone.
  - The robot should **fully stop** at the finishing zone.



Servo Motors

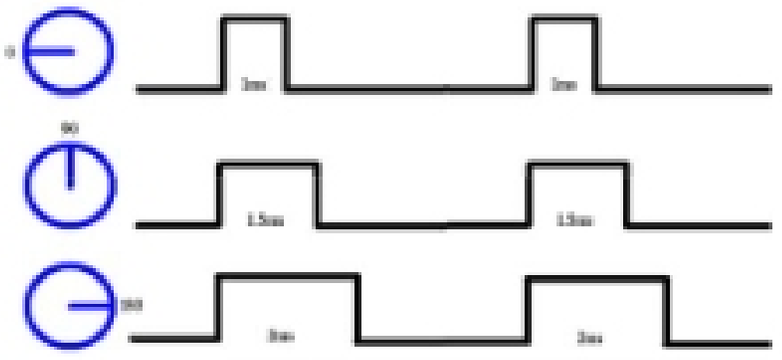
- A motor that has an on-board electric circuit for controlling the direction of rotation, as well as the position, of the output shaft.
- Typical rotation range: 90 degrees or 180 degrees

Usage of Servo Motors

- Used in a wide variety of application
  - Robotic arms
  - Toy-cars, air-planes and helicopters
  - Industrial machinery
- Easy to control
  - Only need PWM signal
- Low energy consumption
  - Works in closed loop
- Small Size
  - As small as several inches squared

Control robot speed/direction

- PWM period should be 16~23ms
  - the frequency should be 44Hz~62Hz (we recommend 50Hz)
- The position of the motor is not determined by duty cycle but the high time. Pulse width range is from 0.9ms to 2.1ms. Typically 1.5ms high corresponds to the neutral position.



Project 4 Checkpoints

- Configure the sensors to detect the black stripe and display it on the four LEDs
- Configure the (PWM). Hook it up to an oscilloscope and make sure the waveforms are generated correctly and precisely.
- Attach the Servos and get them working with PWM.
- Program your state machine algorithm to finish the path.
- Testing. You can use 4 - AAA batteries as external power (instead of USB supplied power).

Important dates

- Flowchart & PWM waveforms due Tuesday 05/06 at noon
  - We only specified the required functionalities. The exact number of states and transitions are left for you to decide.
- C Code due Friday 05/16 at 11am
- Competition time Friday 05/16 at 1:30
- Project Report Monday 05/19 at noon