6.270 Lecture 2

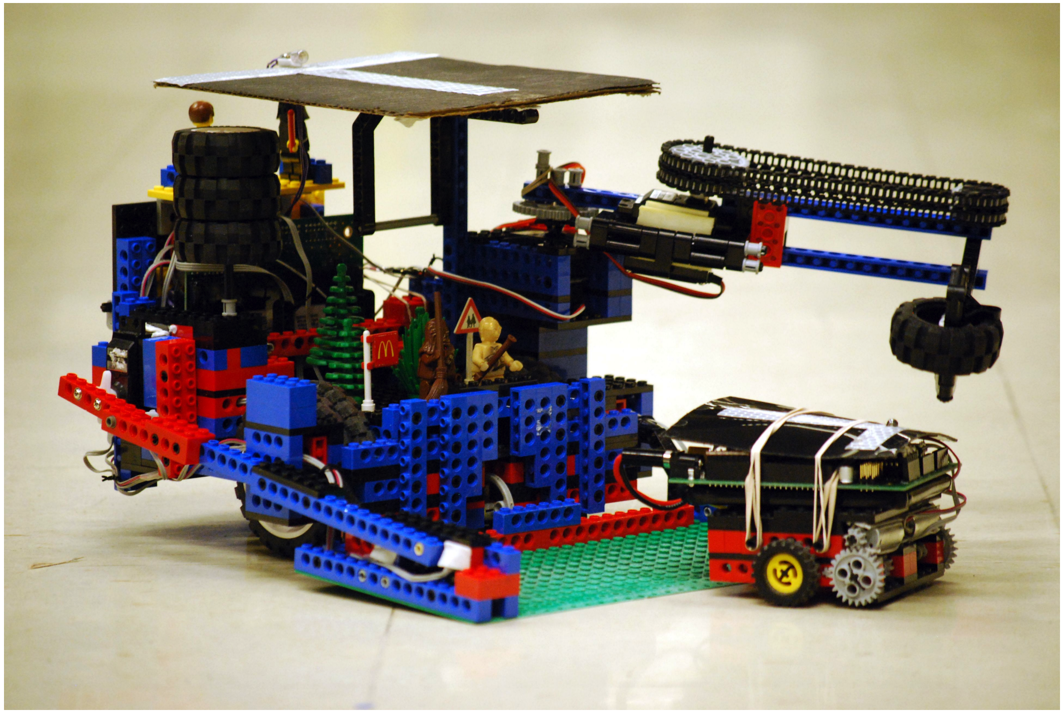
A basic robot

Scott Bezek
January 2011

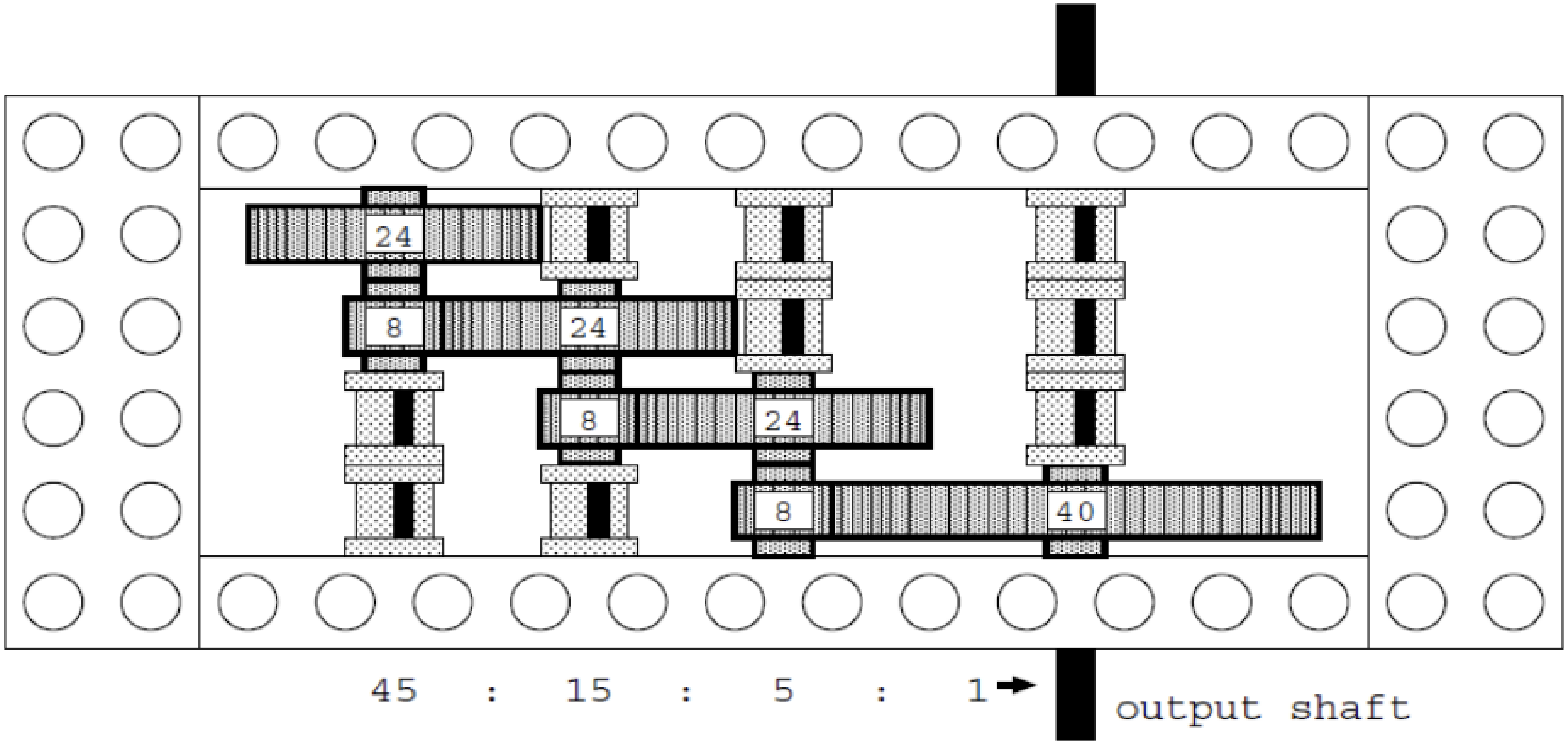
- Thanks for your patience and help during parts sorting!
- Still have some parts to distribute – look out for emails today
- HappyBoard Status
- Control Systems workshop moved to Friday
- Rules of lab

Overview

- Today:
  - Gearing/Bracing
  - Driving
  - Sensing/feedback
  - Basic software
- Later this week:
  - Control
  - Localization
  - Navigation
  - Fault tolerance

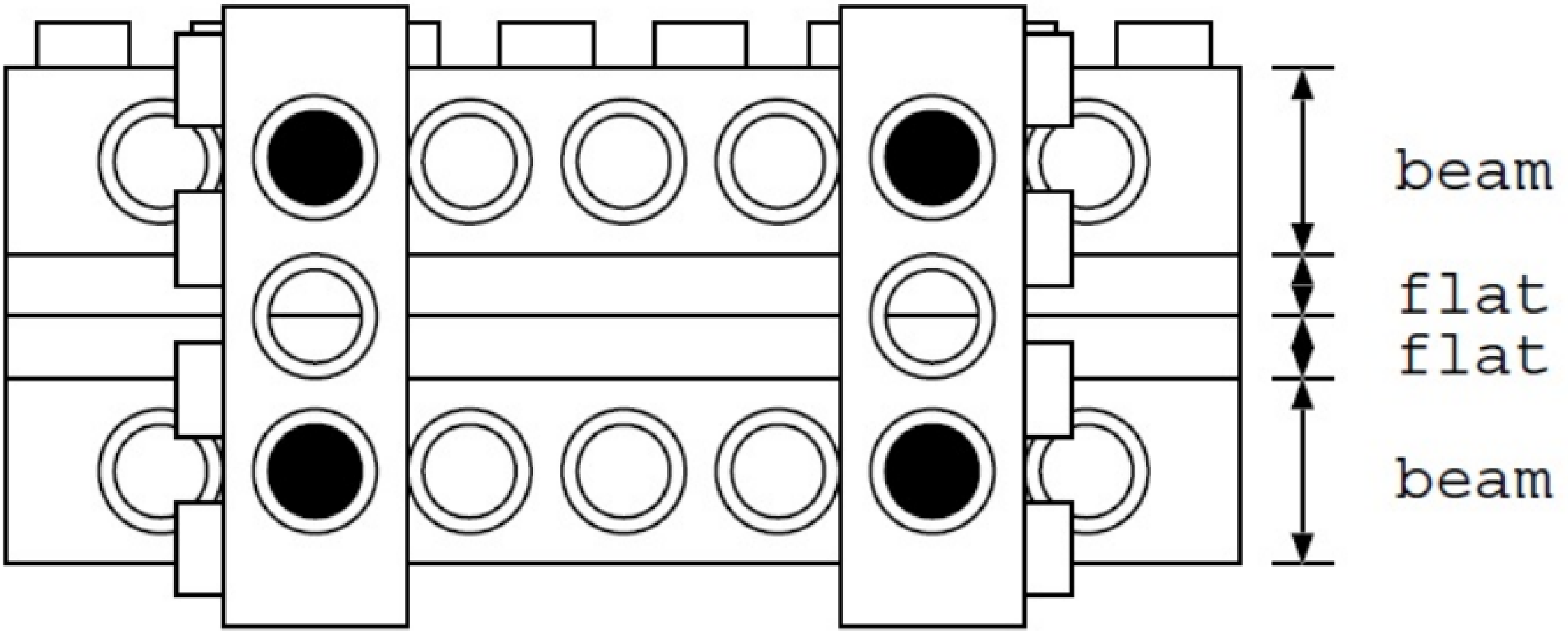


Gearing



- Doubly support shafts
- Start with a gear ratio between 50:1 and 125:1

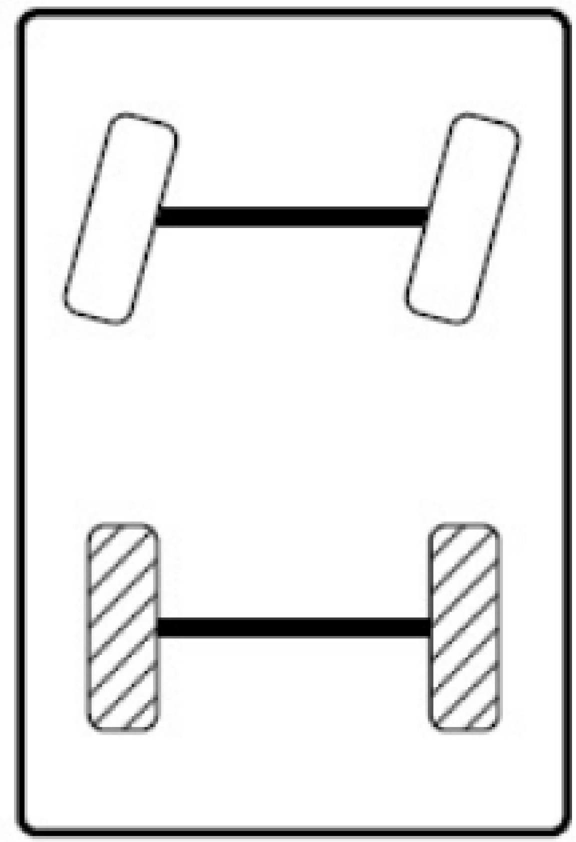
Bracing



- Use perpendicular beams for strength
- Must survive 3 foot drop test

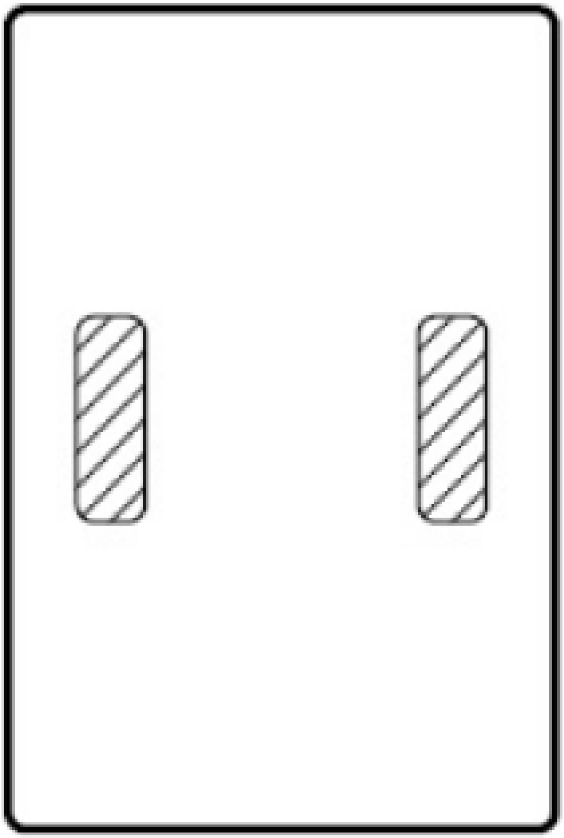
Driving - Steering

- Steering wheel + Drive wheel
- Great for cars: only need one engine
- Pros: easy to drive straight
- Cons: wide turns, hard to navigate (parallel parking!), must move forward or backward to rotate

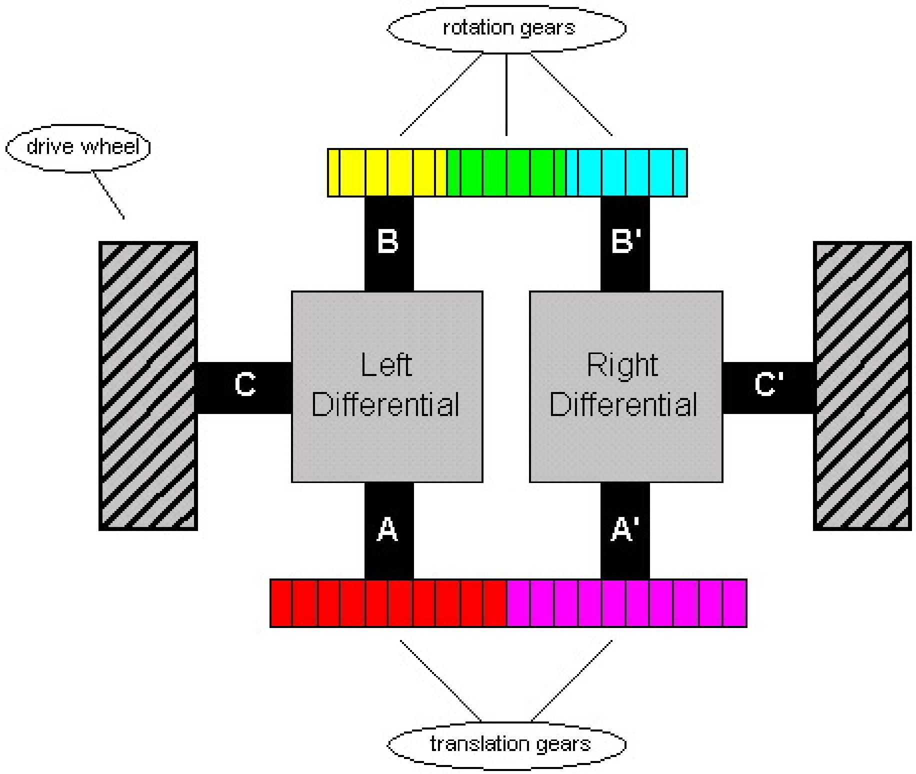


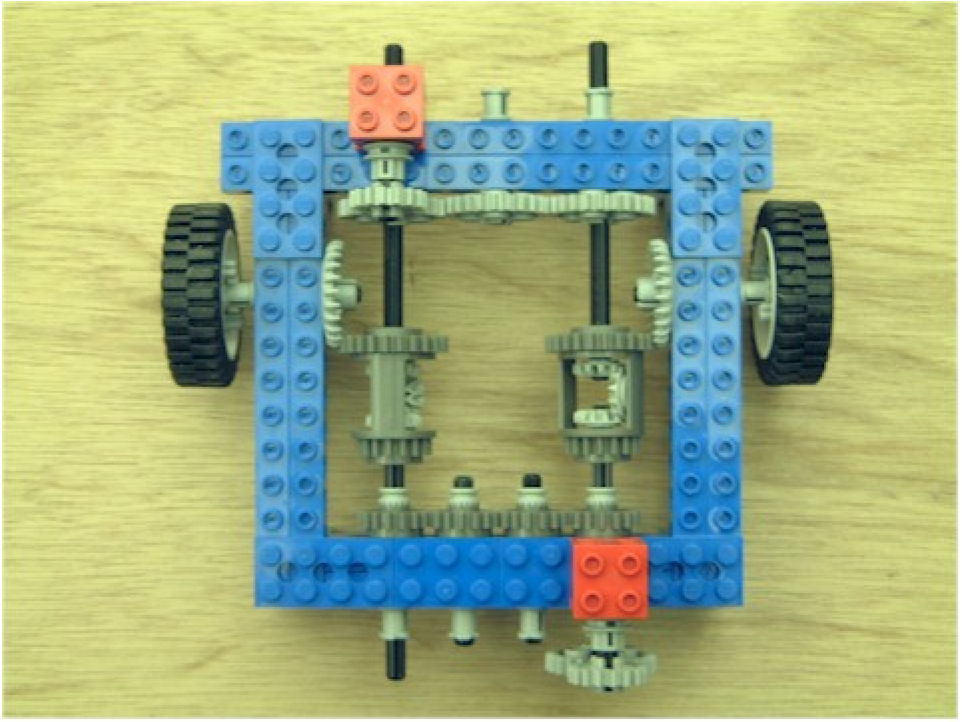
Driving - Differential

- Two independent drive wheels + free wheel or skid
- Most common for robots
- Pros: simple, rotate in place, easy control system
- Cons: hard to drive straight, still kind of hard to navigate



Driving – Dual Differential



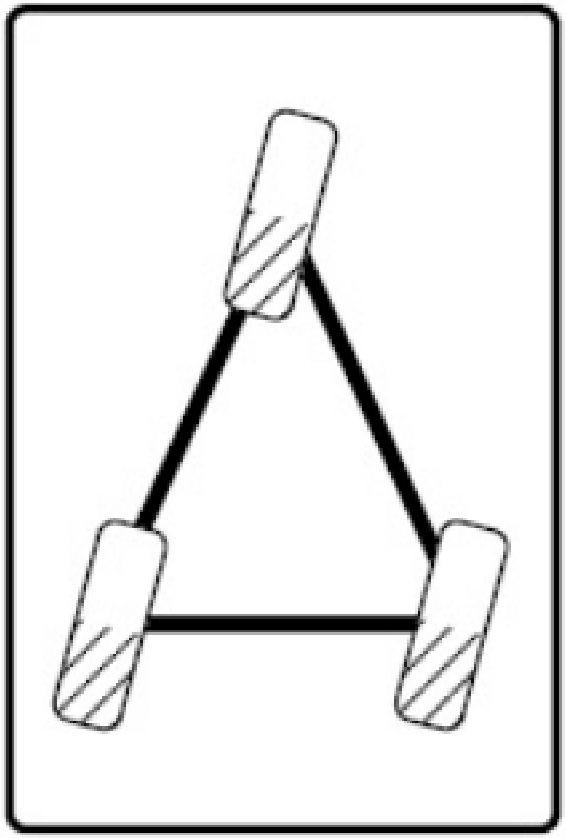


- Like differential drive, but separates drive motor from rotation motor using gearing
- Pros: Fixes issue of driving straight
- Cons: inefficient frictional losses from many gears, somewhat complex

Photos from http://groups.csail.mit.edu/drl/courses/cs54-2001s/dualdiff.html

Driving - Synchro

- 3 drive+steering wheels
- Wheels steer in sync when driving straight
- Pros: No need to turn before translating
- Cons: Complex LEGO structure – leads to tall awkwardly balanced robot



Driving - Omnidirectional

- Drive wheels have embedded smaller wheels that allow for sideways movement
- Pros: can translate and rotate simultaneously, easy control system
- Cons: impractical to build with LEGO
- http://www.youtube.com/watch?v=NPGeqwEW8Mo

Driving

- Questions?

- Mount motors and build chassis by Friday, in time for the Control Systems Workshop

Basic Software

- Let's make the robot move...
- Start simple:
- Turn on a motor
  - int umain(){
  - motor_set_vel(0, 150);
  - }

Basic Software 2

- Turn on motor for 3 seconds
  - int umain(){
  -     motor_set_vel(0, 150);
  -     pause(3000);
  -     motor_set_vel(0, 0);
  - }

Basic Software 3

- Basic control loop: read sensors, act, repeat
  - int umain() {
  -     while(1) {   //loop forever
  -         if (digital_read(0) == true) {
  -             motor_set_vel(0, 0);
  -         } else {
  -             motor_set_vel(0, 150);
  -         }
  -     }
  - }

Basic Software 4

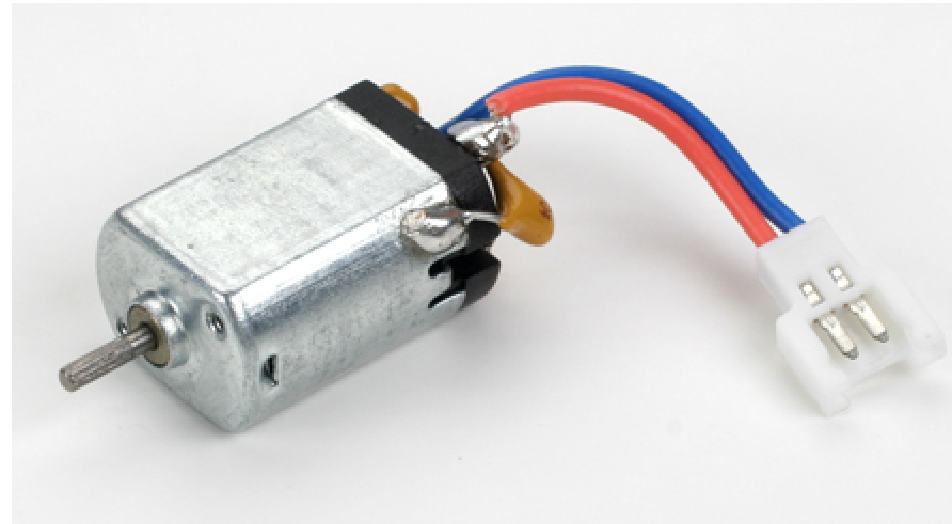
- Drive “straight”:
  - int umain() {
  - while(1) { //loop forever
  - if (gyro_get_degrees() > 45) {
  - motor_set_vel(0, 150);
  - motor_set_vel(1, 50);
  - } else {
  - motor_set_vel(0, 50);
  - motor_set_vel(1, 150);
  - }
  - }
  - }

Hardware Toolbox

- Output:
  - High speed motors
  - Servos
- Input:
  - A whole bunch of standard sensors:
    - Switches
    - Encoders
    - Gyro
    - Distance
  - Buy your own sensors!

High speed motors

- Need to gear down to reduce speed and increase torque
- Gear ratios between 75:1 and 125:1 work well



- motor_set_vel(PORT, VELOCITY)
  - VELOCITY = -256 to 255
- Quick changes may cause brownout

Servo Motors

- Limited range of motion: +/- 90 degrees
  - Can be modified for continuous rotation
- “black box”: tell servo to rotate to specified position
- Useful for arms, lifts, platforms
- servo_set_pos(PORT, POSITION)
  - POSITION = 0 to 511

Digital Inputs

- Two possible values – on or off
- Provide 1 bit of information (0/1)
- HappyBoard has 8 digital inputs, **WITH PULLUP RESISTORS**
- Read value with digital_read(PORT) → 0 or 1

Analog Inputs

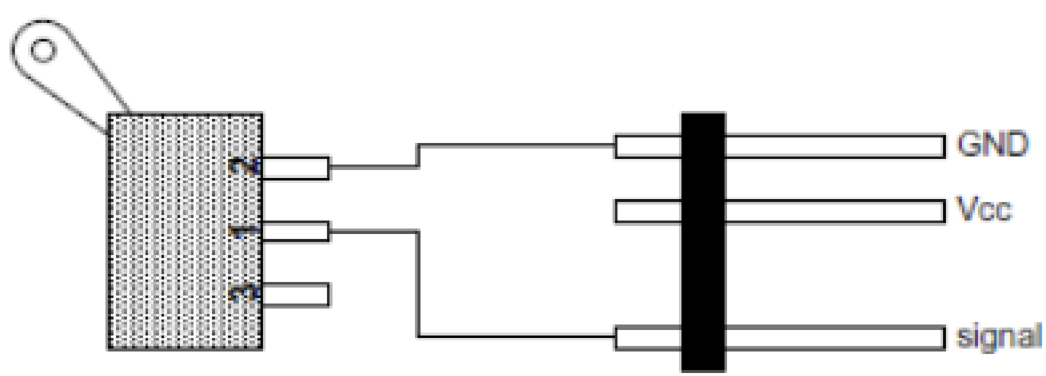
- Continuous voltage range (0-5V)
- Infinite intermediate voltages, so must be quantized
- HappyBoard has 10-bit ADCs, so voltages from 0-5V map to values 0-1023
- 16 analog inputs, **with pullups**
- Read value with analog_read(PORT) → 0 to 1023

Encoder (Digital)

- HappyBoard has 4 “Encoder” inputs
- Specialized digital input
- Designed to count the **# of transitions** (from 0→1) extremely quickly (no need to continuously poll the input)
- Useful for counting rotations of an axle at high speed
- Read the count with encoder_read(PORT) → 0 to 65535
- Reset with encoder_reset(PORT)

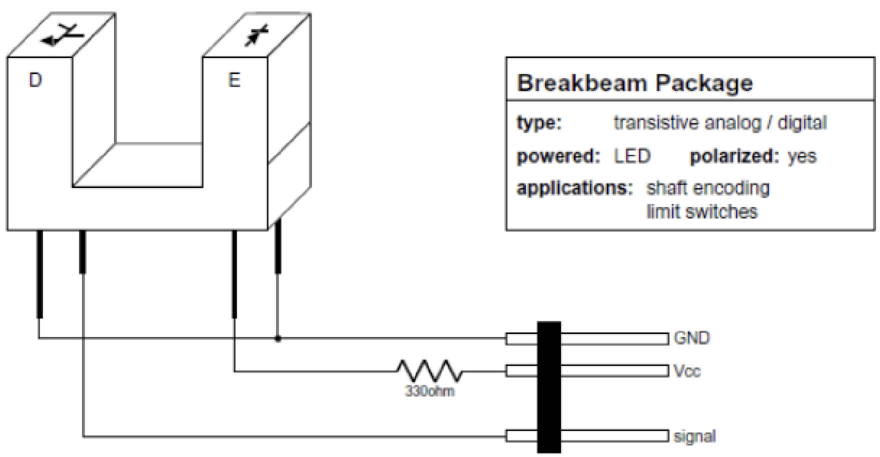
Switch

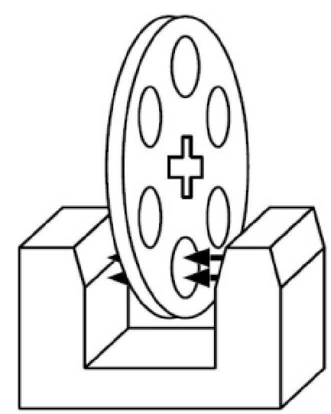
- Simplest digital sensor
- SPDT – connect with COM/NC, or COM/NO, but NOT NC/NO
- Often used for collision detection
- Other uses: lift switch, object detection (breakbeam perhaps better)

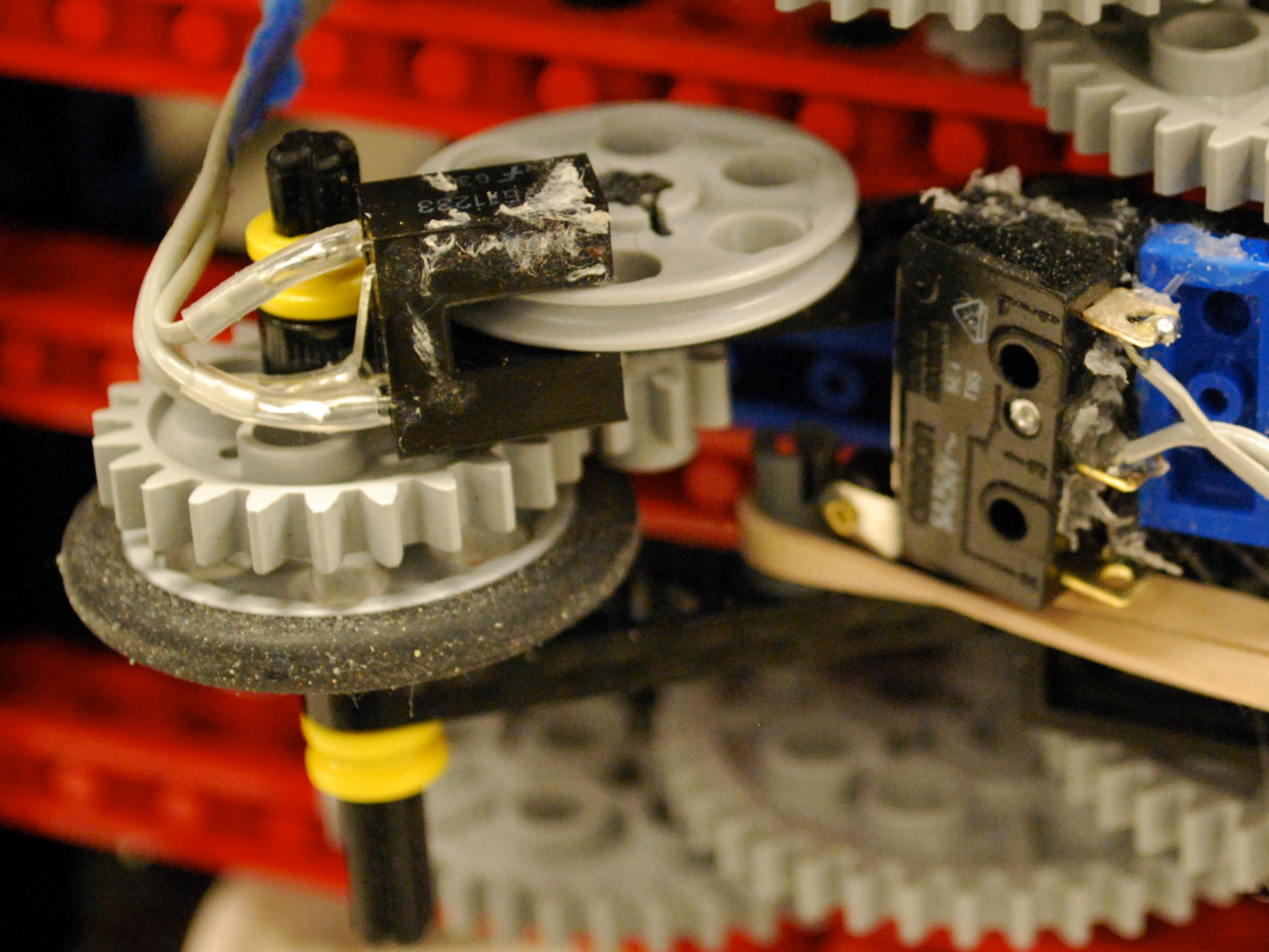


Optical Encoder/Breakbeam

- Connect to encoder inputs to count how many times IR beam is disrupted (broken)
- Use with 8-hole pulley to count axle revolutions
- Place “high” on the gear train for highest resolution
- Encoder on free wheel vs. drive wheel





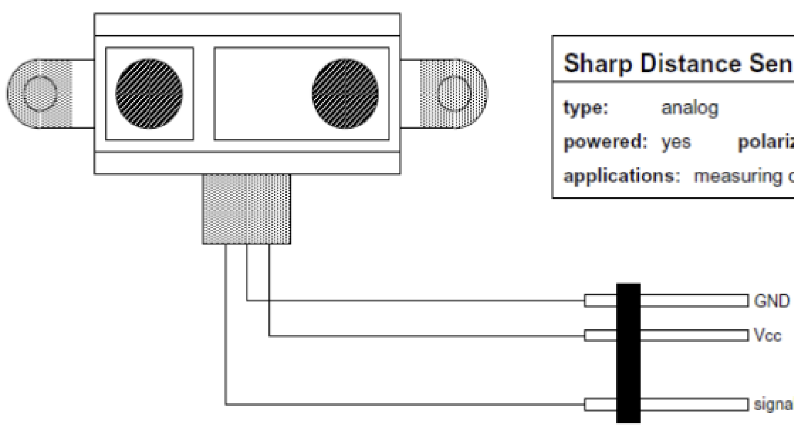


IR LED/Phototransistor pair

- Essentially a “big” version of the encoder/breakbeam sensor
- Decent range (~12in)
- Can be used for object detection
  - Set up beam across area of interest (e.g. inside claw)
  - Or, check for IR reflections from nearby objects
- Analog signal from phototransistor, but generally set a threshold in software to make it digital

IR Distance Sensor

- Range: ***<u>6in</u>*** to 3ft
- Nonlinear response – requires calibration for accurate distance measurement
- Useful for detecting things around the robot
- Must cut trace on HappyBoard



Sharp Distance Sensor

type: analog
powered: yes **polarized:** yes
applications: measuring distance

Gyroscope

- Provides single-axis angular velocity
- Built-in functions to integrate to get angle (theta)
- Must be calibrated
- Accuracy: +/- 5 deg. over 2 minutes
- Error compounded by integration, builds up over time
- Be wary of spinning too fast (e.g. hitting walls hard) – can saturate voltage, causing angle to be thrown off
- Any slight tilt changes perceived rate-of-rotation
- gyro_get_degrees()

Upcoming Events

- Soldering Workshop: 3pm in lab
- C Crash Couse Part 1: 7pm in 34-101

- Mock Competition 1 coming up next Monday!

Appendix: Disconnecting pullup resistors