ECE 100 - ITP
Lecture 7

Dr. Alexander J. Flueck
Electrical and Computer Engineering
Illinois Institute of Technology

flueck@iit.edu
http://www.ece.iit.edu/~flueck/ece100

Questions?

- HandyBoard/Interactive C/motors/sensors/LEGO/path
- How do I check and then fix IC syntax?
- Executive Summary - Due Oct 15 at the beginning of lecture.
- Presentation and Teamwork Scores
  - Were sent out via email. Contact me with any concerns.
- Mid-term Progress Grades
  - Will be assigned next week.

Executive Summary Assignment

- Due in **lecture** on October 15.
  - One page writing assignment to be reviewed by the Writing Lab Instructors in the Writing Lab (2nd floor Siegel Hall)
  - The "Executive Summary" is an argument, i.e., you will present a thesis, your analysis, and some evidence.
  - See links on ECE 100 web page
    - Technical Communication Resources; Argument Fundamentals; Exec Summary FAQ
- Topic: autonomous robot design proposal
  - You are the lead engineer for autonomous robots at a hi-tech automation and robotics firm specializing in hazardous search and rescue missions
  - Senior management has requested a design proposal for a robot that will meet a variety of customer needs with a better price/performance ratio than your competitor's current product
  - Use your LEGO robot prototype to demonstrate the strength of your design
  - DARPA Grand Challenge: http://archive.darpa.mil/grandchallenge/gallery.asp

Laboratory Reports - TA Feedback

- Design Portfolio format (pre-lab: PS,R/I,AS,OS,App; post-lab: complete)
  - Problem Statement: Navigate a maze in the shortest time possible.
    - Be concise and quantitative. List assumptions, constraints and criteria.
  - Research/Investigation: Present a theory for optimal performance.
    - Describe what you've learned from other sources & past efforts.
  - Alternative Solutions: Present various strategies (must be different!).
    - Include flowcharts to illustrate the strategies. Discuss differences & pros/cons.
  - Optimum Solution: Why was it chosen? How do you plan to get it done?

  - Construction/Implementation: Describe robot hardware/software as built
    - Explain/illustrate configuration details and any deviation from original plan.
  - Analysis/Testing: Quantify performance. Modify if necessary.
    - Perform repeated trials. Record your observations.
  - Final Evaluation/Conclusion: Comment on problems, adequacy of solution and future work.

  - Appendices: Attach IC codes

Teamwork Feedback

- Prepare before you come to lab.
- Participate in the design, improvement, re-work, etc.
- Listen to your teammates.
- Open communication is important.
- Don't try to control the entire project or your teammates, even if you're sure you know you're right. Uncertainty is everywhere; certainty is dangerous.
- Meet together as a team before lab, or at least exchange email plans.
- Divide and conquer.
- Be brutally honest with your performance.
- Be confident in your future ability.

Team Presentations - TA Feedback

- Focus on actual prototype. Don't waste time on fantasy application.
- Need large font and contrasting colors/brightness.
- Use bullets, not paragraphs. Should not be Post-Lab report on slides.
- Use diagrams - focus on illustrating the technical details.
  - Avoid overly complicated flowcharts
  - Avoid code - too hard to comprehend in real-time
  - Include picture of robot design modifications
- Must present results! And future plans!
- Establish eye contact with your audience.
- Practice your presentation in front of someone. Don't just read slides.
- Focus on persuasion. Not just a historical account.
  - Be specific.
  - Provide evidence.
  - Justify funding.

Lab 4 Observations

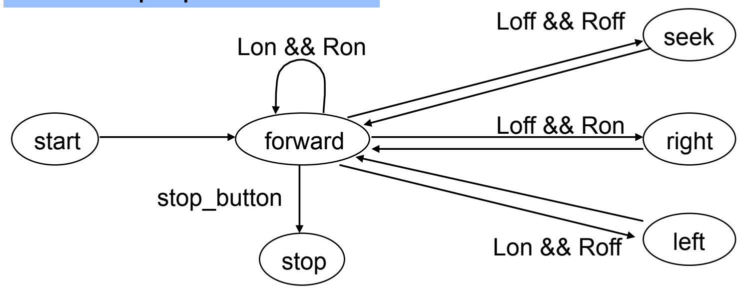
- Physical shakedown of robot chassis may be necessary.
  - Eliminate sources of friction, especially drive train.
  - Don't let motors stall! Motor windings can easily overheat.
  - Be careful with speed vs. torque tradeoff.
  - Don't force it! Pins and wires are fragile!
  - Beware of battery wires. Short circuit can cause severe damage!
  - Light sensor placement and shielding are important.
  - Document your designs in C/I (Design 1 features, Design 2 features, etc.) and your performance in A&T (Design 1 results, Design 2 results, etc.).
- Software
  - Need to handle possibly mismatched sensors.
  - "Simulate" robot performance with light_test.ic and your path-following code.
- Feel free to modify your robot and your code. Successful teams will take risks and learn from their experimentation.
  - How well did your robot follow the path? Why?

Flowcharts - TA Feedback

- Use the "standard shapes"
  - Ovals for start and stop (every flowchart must have start/stop)
  - Boxes for actions
  - Diamonds for decision points
- Package your routines such that each fits on a single page
  - Do not use arrows that point to the next/previous page
  - Use an action box, then create a separate flowchart for the action
- Double-check your flowcharts
  - Only diamonds are allowed to have two outputs
  - Flawed "metasens" flowchart in the textbook

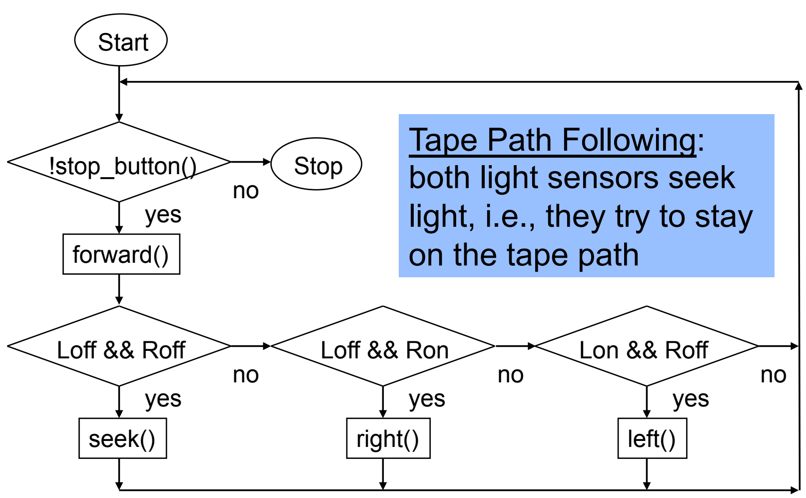
Converting “Solution into Code” - State Machine

<u>Tape Path Following</u>: both light sensors seek light, i.e., they try to stay on the tape path



Converting Solution into Code - The Flowchart

<u>Tape Path Following</u>: both light sensors seek light, i.e., they try to stay on the tape path



Braitenberg - extended

```
void main() {
   while (1) {
      motor(LEFT_MOTOR, normalize(analog(RIGHT_EYE)));
      motor(RIGHT_MOTOR, normalize(analog(LEFT_EYE)));
      if ((normalize(analog(LEFT_EYE)) == 0) &&
         (normalize(analog(RIGHT_EYE)) == 0)) {
         backward(); sleep(0.5);
         left(); sleep(1.05);
      }
      .
      .
      .
   }
}
```


Bad Code 1 - "recursive function loop"

```
void main() {
   while (1) {
      motor(LEFT_MOTOR, normalize(analog(RIGHT_EYE)));
      motor(RIGHT_MOTOR, normalize(analog(LEFT_EYE)));
      killer();
   }
}

void killer() {
   .
   .
   .
   main();
}
```

Bad Code 2 - right light sensor never recognizes dark surface

```
void main() {
   while (1) {
      if(analog(REYE)<95) {
         bk(RMOTOR);
         bk(LMOTOR);
         if(analog(REYE)>95) {
            .
            .
            .
         }
      }
   }
}
```


Bad Code 3 - robot always backs up! (conditional expression mistake)

```
void main() {
   while (1) {
      forward();
        .
        .
        .
      if( ( analog(REYE) && analog(LEYE) ) <95) {
         bk(RMOTOR);
         bk(LMOTOR);
            .
            .
            .
         }
      }
   }
}
```

Bad Code 4 - should use "while" loop

```
void main() {
   while (1) {
      if(analog(REYE)<95 && analog(LEYE)<95) {
         fd(RMOTOR);
         off(LMOTOR);
         sleep(1.25);
      }
      if(analog(REYE)<95 && analog(LEYE)<95) {
         fd(RMOTOR);
         off(LMOTOR);
         sleep(0.25);
      }
      if(analog(REYE)<95 && analog(LEYE)<95) {
         fd(RMOTOR);
         off(LMOTOR);
         sleep(0.25);
      } /* actually repeated three more times! */
   }
}
```


Post-Lab 5 Milestone Report - Due week of Oct 8

- Post-Lab Milestone report -- details, details, details!
  - Detailed summary of how things have gone so far. What? When?
  - Describe testing procedures and robot performance. Provide table with performance categories and corresponding evaluations for each trial.
  - Present your accomplishments alongside the detailed timeline in your proposal (complicated problem broken into smaller tasks).
    - Be sure to cover both hardware and software.
  - Is the project on schedule? If yes, what were the most critical tasks?
  - If not, why not? What remedies are available? New flowcharts?
  - You may need to modify your strategy and timeline as you go. Deviations must be explained in the Milestone reports.
  - Attachments: new optimum solution code, lab notes
- Key issue: Show how you have succeeded, even if you faced unforeseen challenges that forced you to adapt.
- Include laboratory teammates' names on post-lab report. Get the dates right. Staple your report! Bring your code! Submit SafeAssign!

Competition Round 1 - Look familiar?

- Follow the path in the shortest time possible.
  - The tape path will be continuous, i.e., no branches, no gaps.
  - The tape path will be between 2 and 3 inches wide.
  - The tape path will not intersect itself.
  - All tape path turns will be less than or equal to 135 degrees to the left or right of the robot's forward direction.
  - If your robot leaves the tape path before the end, then its position will be recorded.
  - If no robot completes the course, then ranking will be based on distance completed.
  - If two or more robots finish with the same time, a winner will be determined randomly.
  - The top four teams across all laboratory sections will compete in the class runoff during the following week's lecture period.
  - All students are required to observe the class runoff in lecture and predict the performance of the robots based on each team's proposed strategy.
  - Complete specifications online: http://www.ece.iit.edu/~flueck/ece100