ECE 2031 Final Design Project

Introduction

ECE 2031 Spring 2019

Design Project Motivation

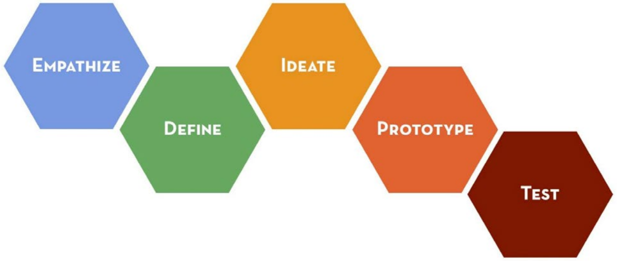
- ECE 2031 includes the sophomore-level team design experience
- You are developing a useful set of tools
  - eventually including an entire computer within the DE2 board
- Using tools creatively to solve problems is what engineers and computer scientists do

ECE 2031 Project Components

- Propose a solution to a problem:
  - What’s the best approach for completing a given task within the constraints of the project?
  - More details next week on proposal
- Implement the proposed design on the DE2Bot
- Demonstrate, present, and document your solution

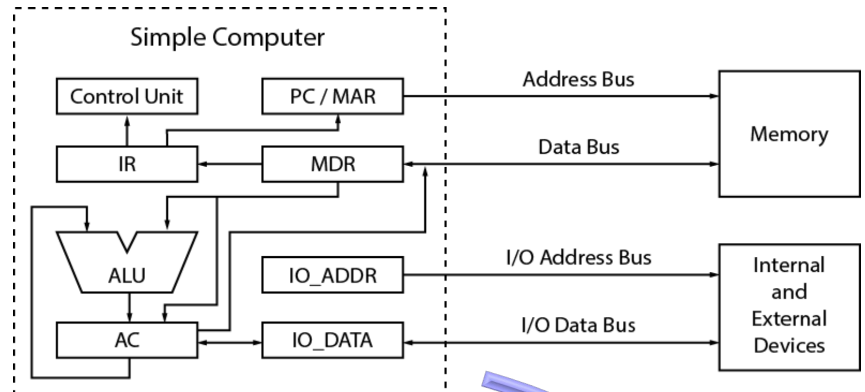
Open-ended Design

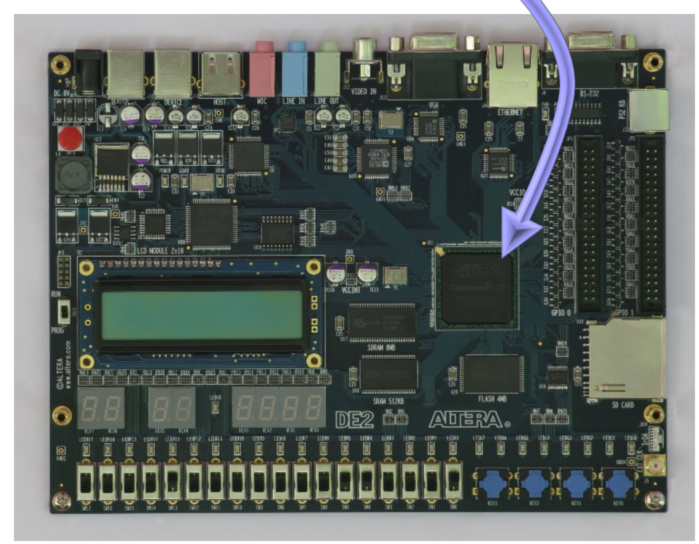
- Recall the very first lecture in 2031, about the design process
  - You will only need to do a little empathizing, but you will go through every other step
- You will also experience many other aspects of open-ended design
  - We will use some of today or next week to talk about limitations, debugging, and efficient design



The Simple Computer (SCOMP)

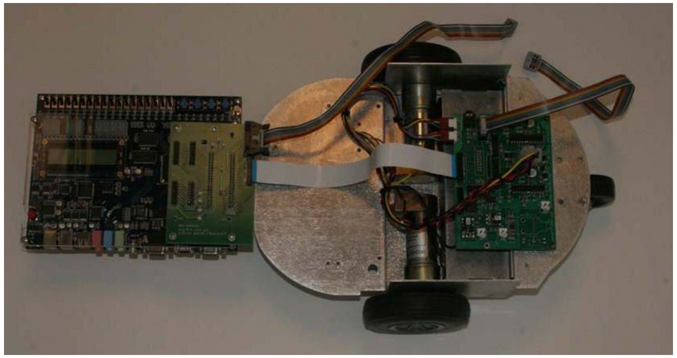
- In Labs 7 and 8, you have implemented a computer in the FPGA
  - Both hardware (a description of gates, flip-flops, etc. that form the computer)
  - And software (various programs that you load onto your computer, in its RAM)





Background on DE2Bot

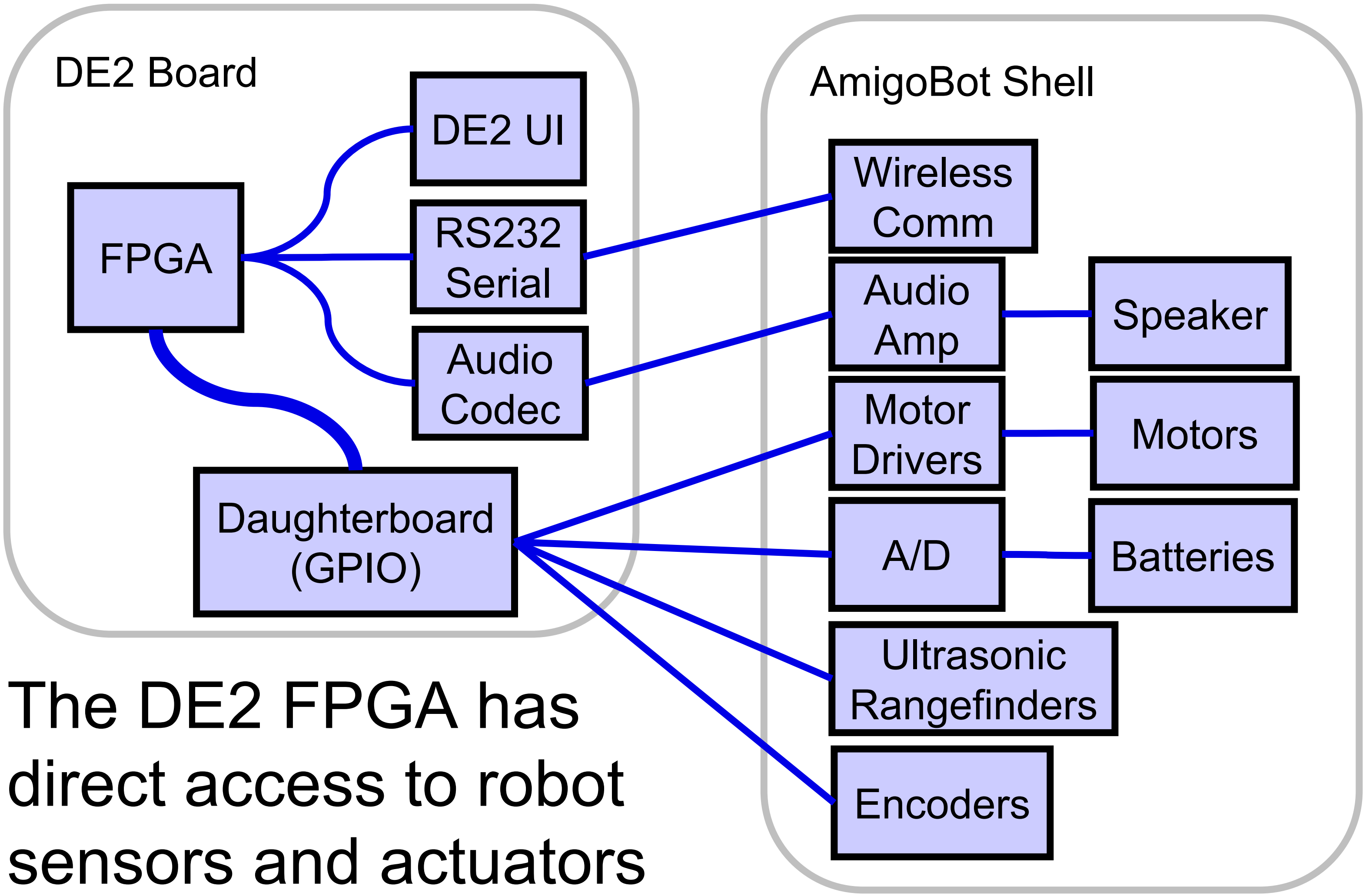
- Many semesters ago, older lab robots were gutted, adding a new internal controller board and a connected DE2 on top
- Then, each semester a new capability has been added, or a new application has been demonstrated



ECE2031 DE2Bot Past Projects

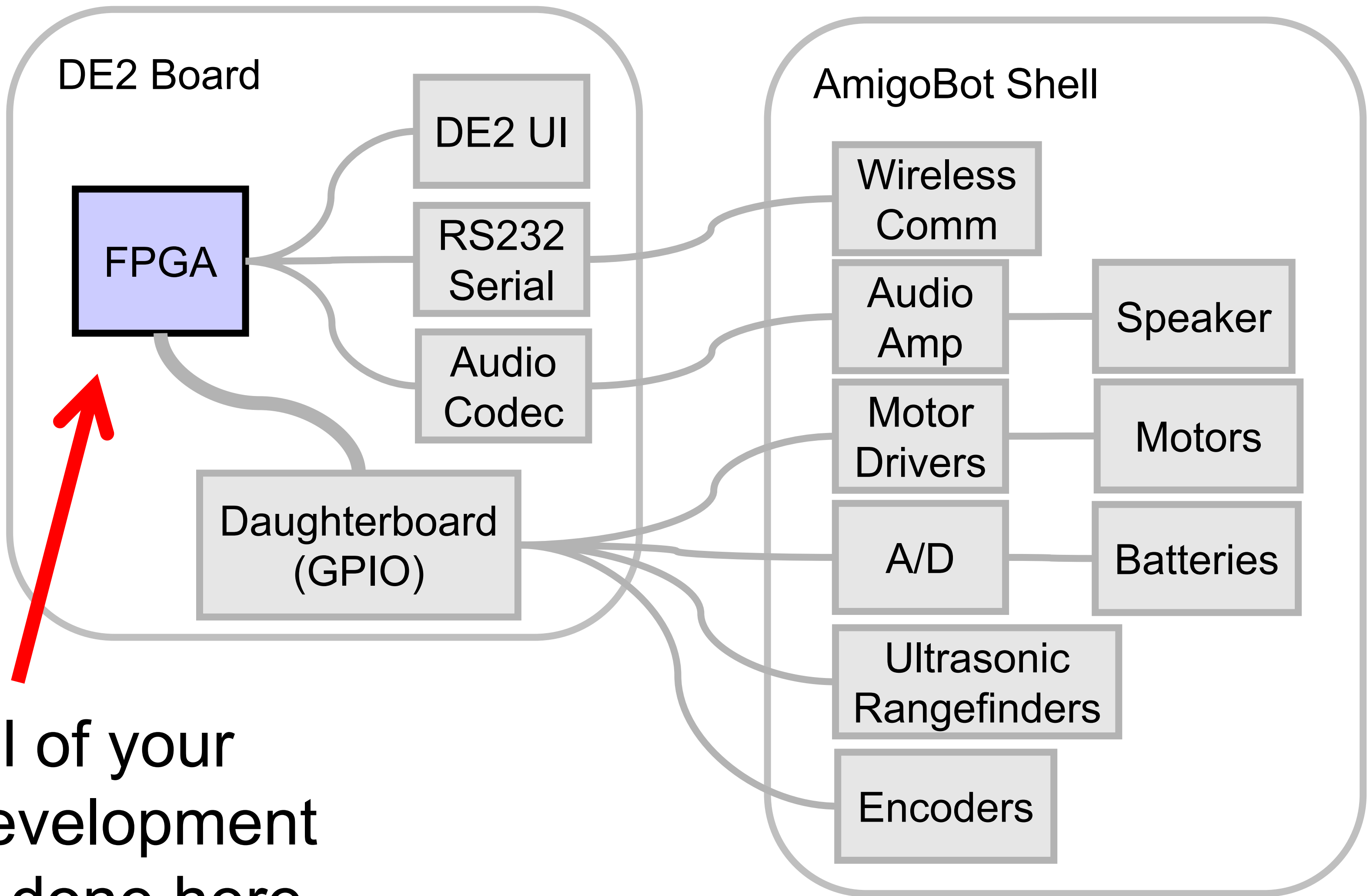
- Position/velocity feedback, and motor velocity control
- Processing of sonar transducers
- I^2C interface for battery monitoring and audio codec control
- Odometry (position estimation from wheel rotation)
- Audio codec interface and digital sound generation
- Robot Self-test
- Infrared signal detection and “remote control” demonstration
- UART for serial communication
- Implementation of hardware interrupts for SCOMP
- Complex mathematical functions in software (ATAN)
- Analyzing sonar data to locate objects and make contact
- Many sensing, localization, and navigation demonstrations

DE2Bot Hardware Architecture



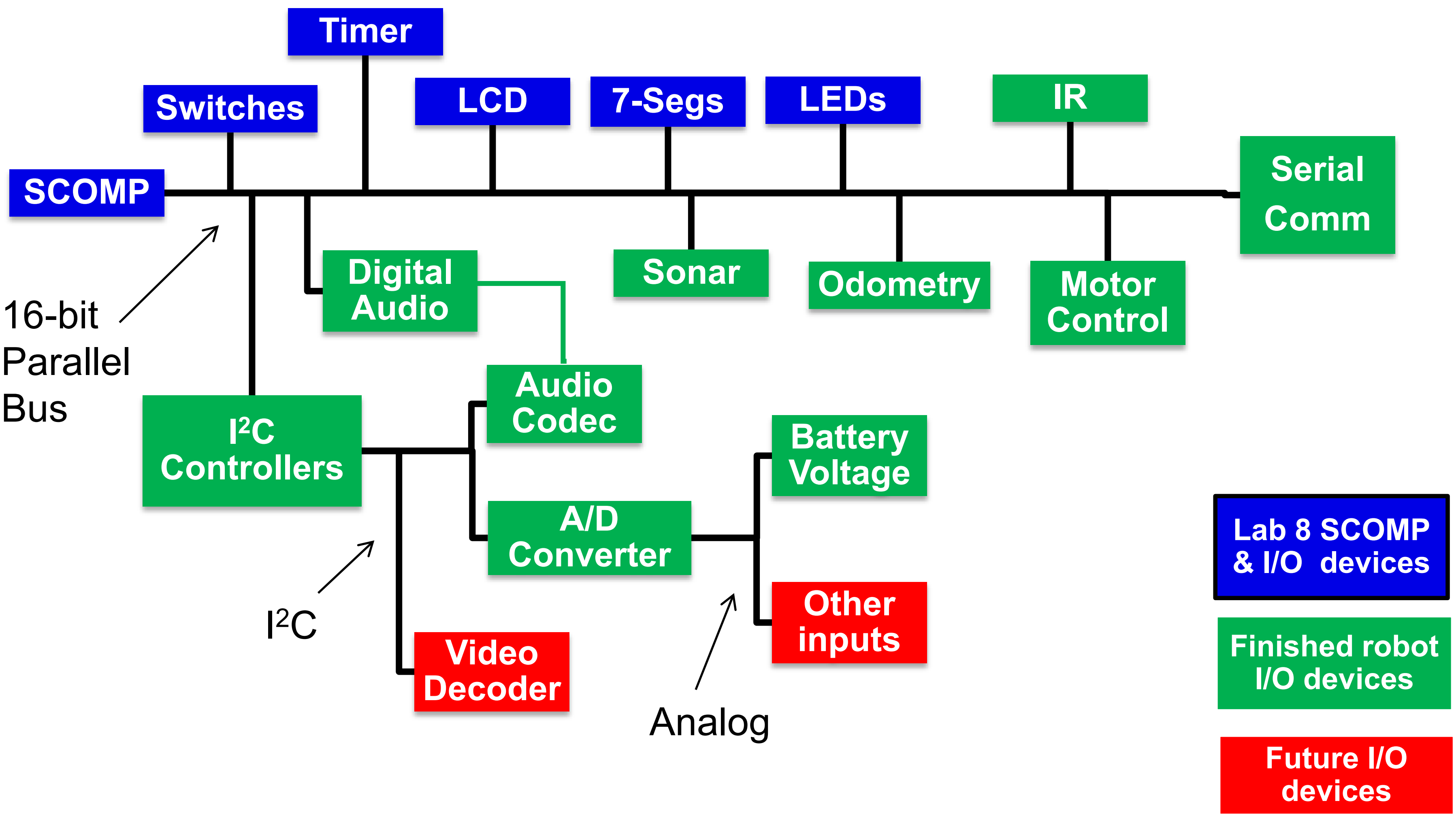
- The DE2 FPGA has direct access to robot sensors and actuators

Project Development



All of your development is done here

DE2 and FPGA System Architecture



Working with Real Systems

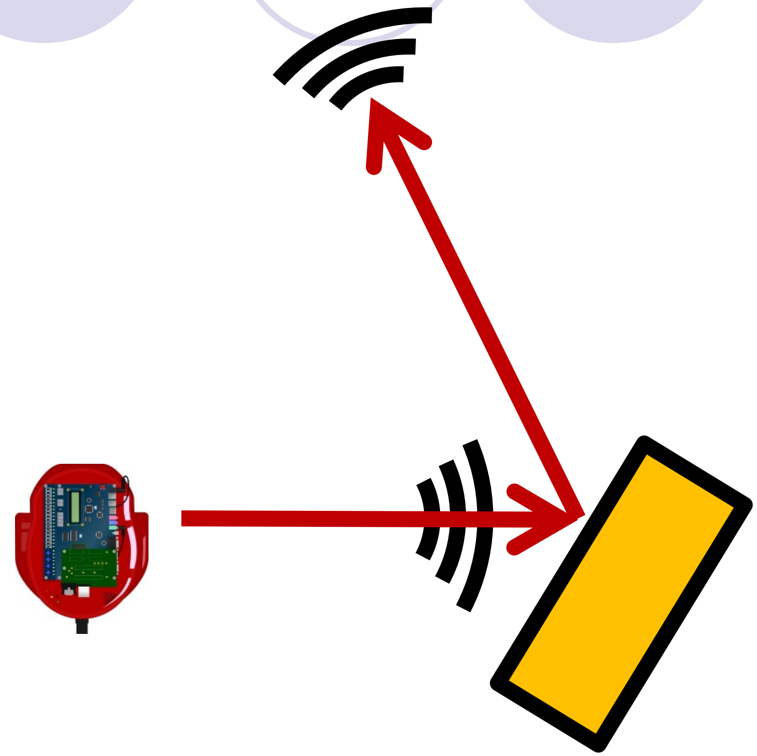
- Part of this project is interfacing with real systems and interpreting real data
- The robot is <u>imprecise</u> – turning 90° might actually be 92°, and moving 5 ft. might be 5.2 ft.
- The data will be <u>messy</u> – it is low-resolution and it will have noise and gaps
- That's part of what makes this project interesting! This isn't a contrived situation, and there is no perfect solution.

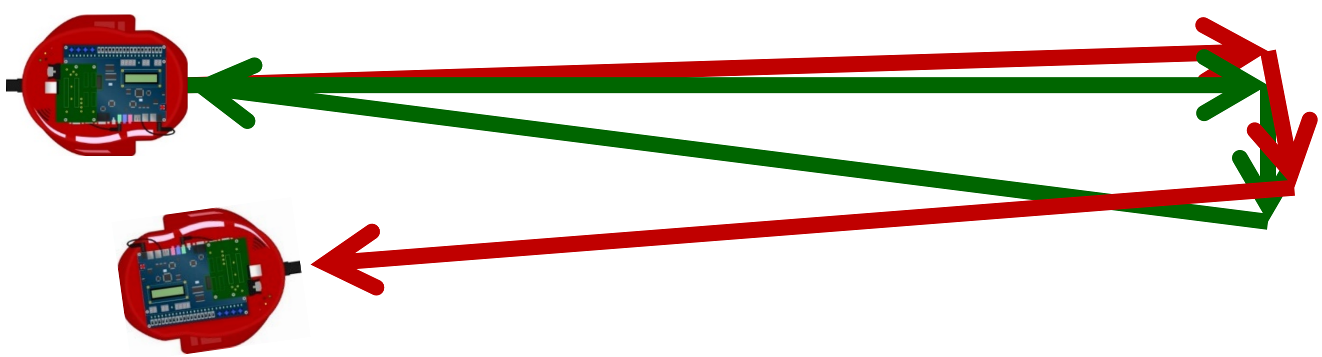
Acknowledging Limitations

- Engineering is a process that involves tradeoffs and limitations
- You never have unlimited speed, silicon, space, memory, time, money, data, people…
- In this project, wishing for different sensors, a smoother floor, more time – is not useful
- The limitations are *the same for everyone*, and it's your job to implement the best solution you can within those confines

DE2Bot considerations

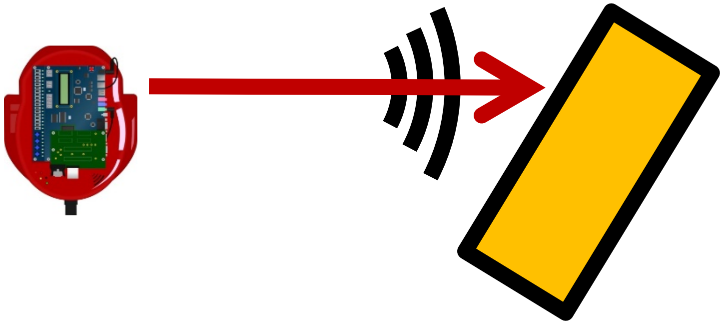
- The recent online activity described odometry and sonar
- Both are likely useful
- Both have their limits





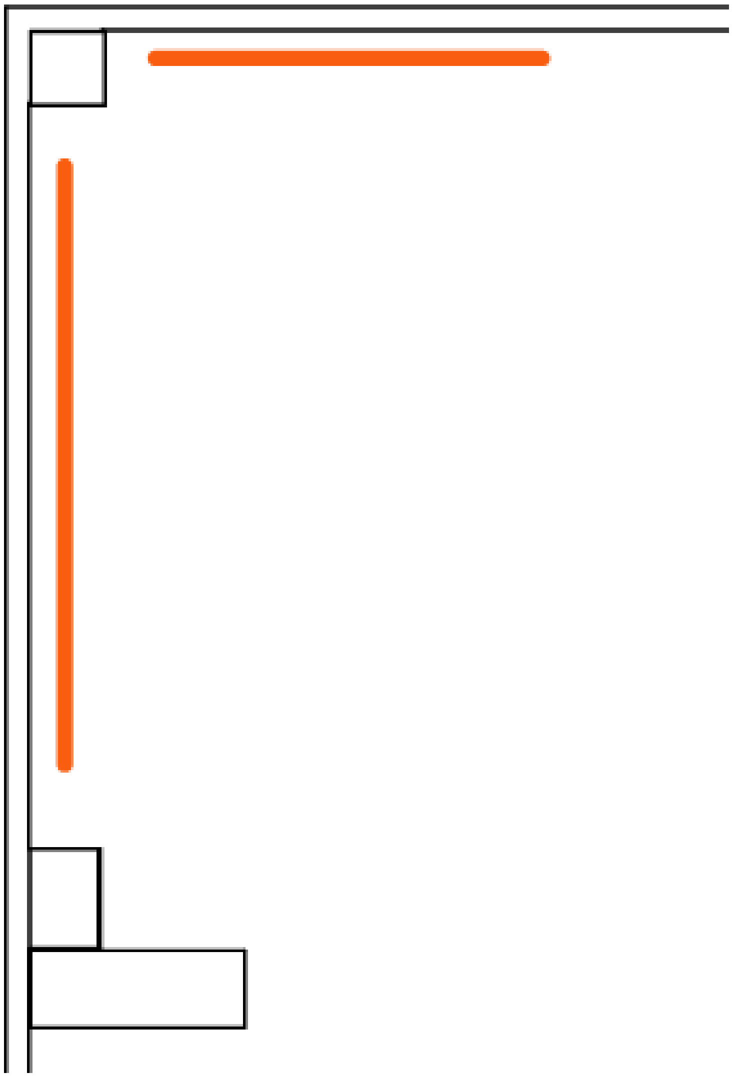
Pros and Cons of Sonar

- Interface is simple
  - 8 sensors, with a single numerical reading from each
  - A reading is just a distance in the direction of the sensor (it is not an x/y position in the robot's coordinate system)
- Objects may be missed
  - Usually because of reflection from oblique surfaces
- But sonar is the ONLY way to get information about the surroundings



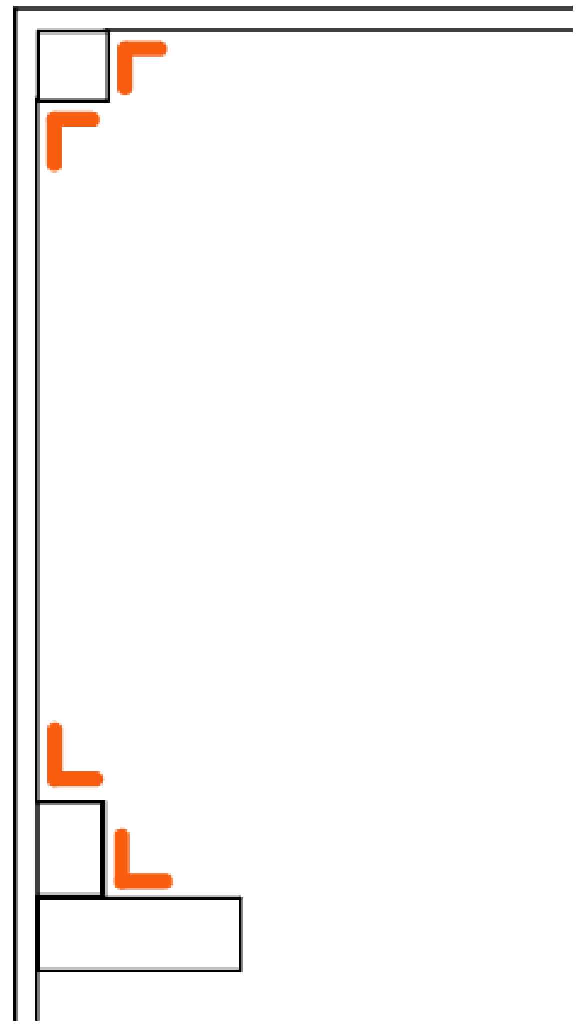
Using sonar

- There are some stretches of flat, orthogonal surfaces
- These may show up clearly in sonar data
  - But only if a sonar transducer is roughly perpendicular
  - And at a distance greater than the minimum sonar reading (~15 cm)



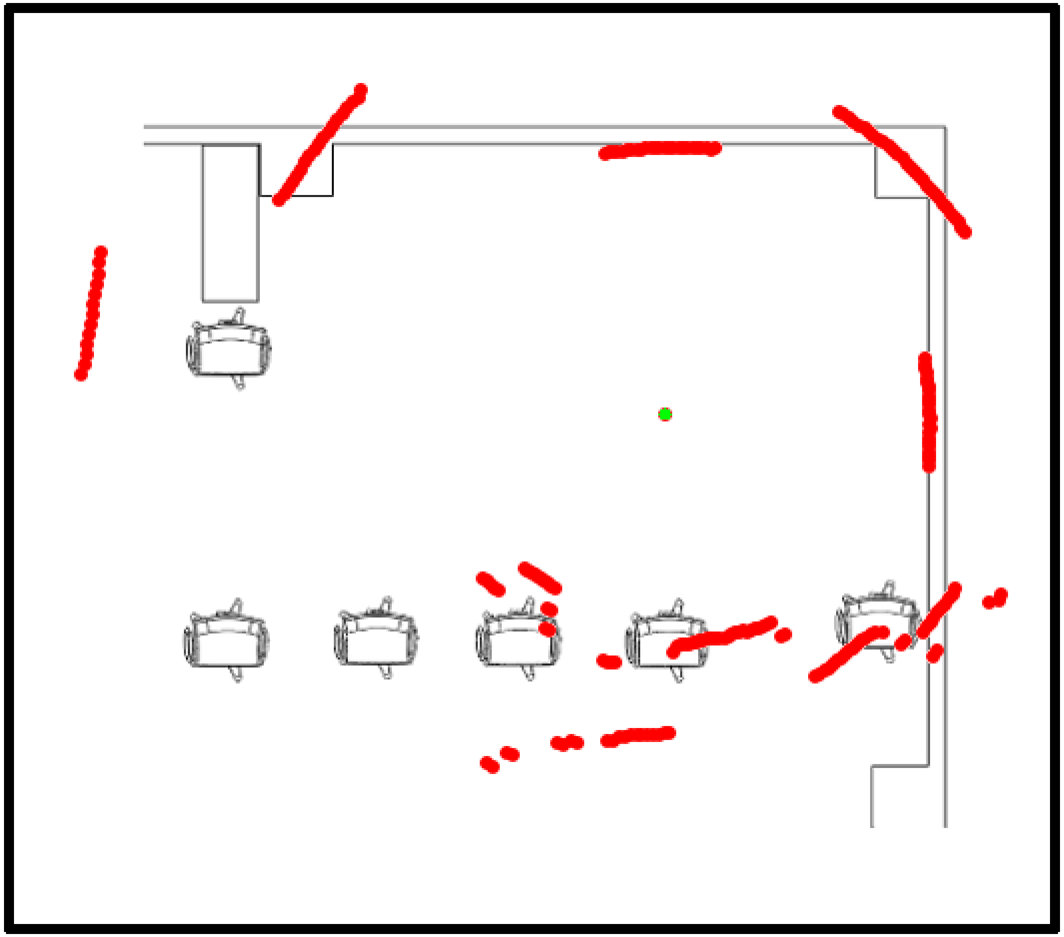
Sonar – corners

- Inside corners make great reflectors (see: corner reflector)
- But they do not look like “corners”, due to sonar beam spread

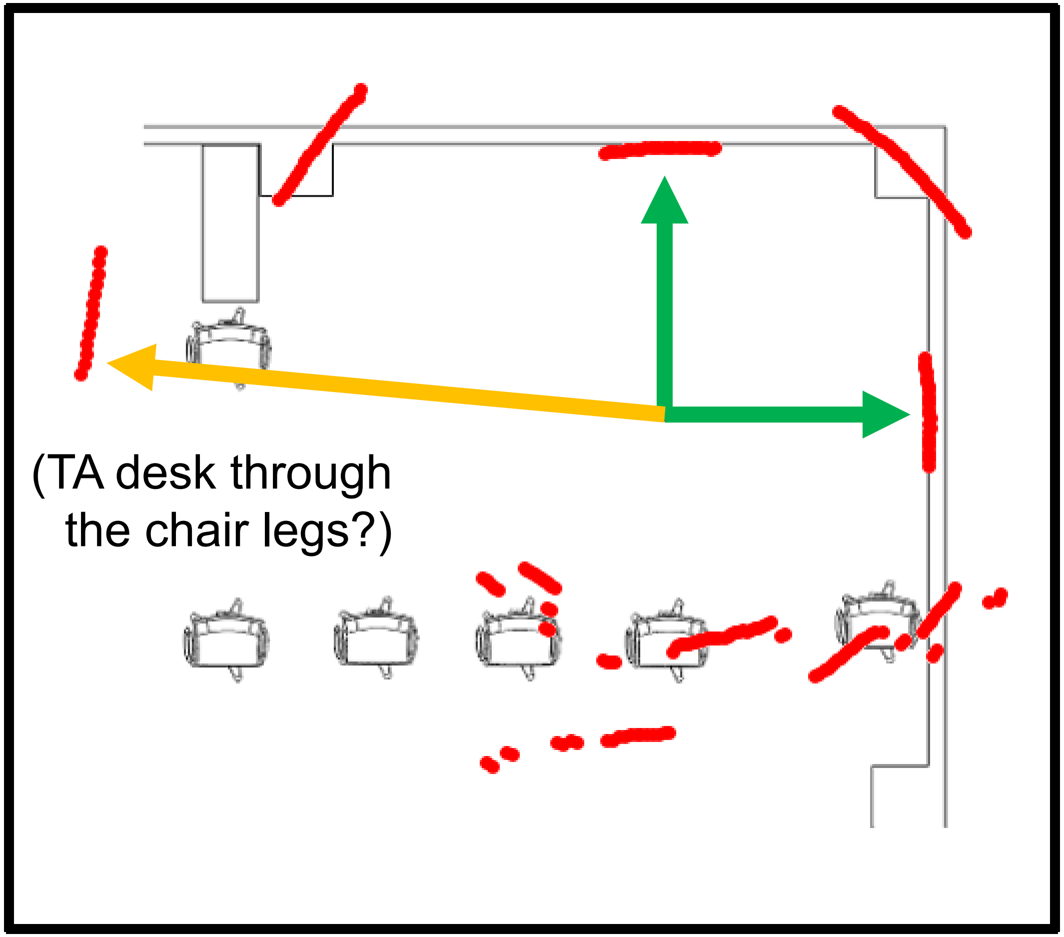


Example Sonar Data

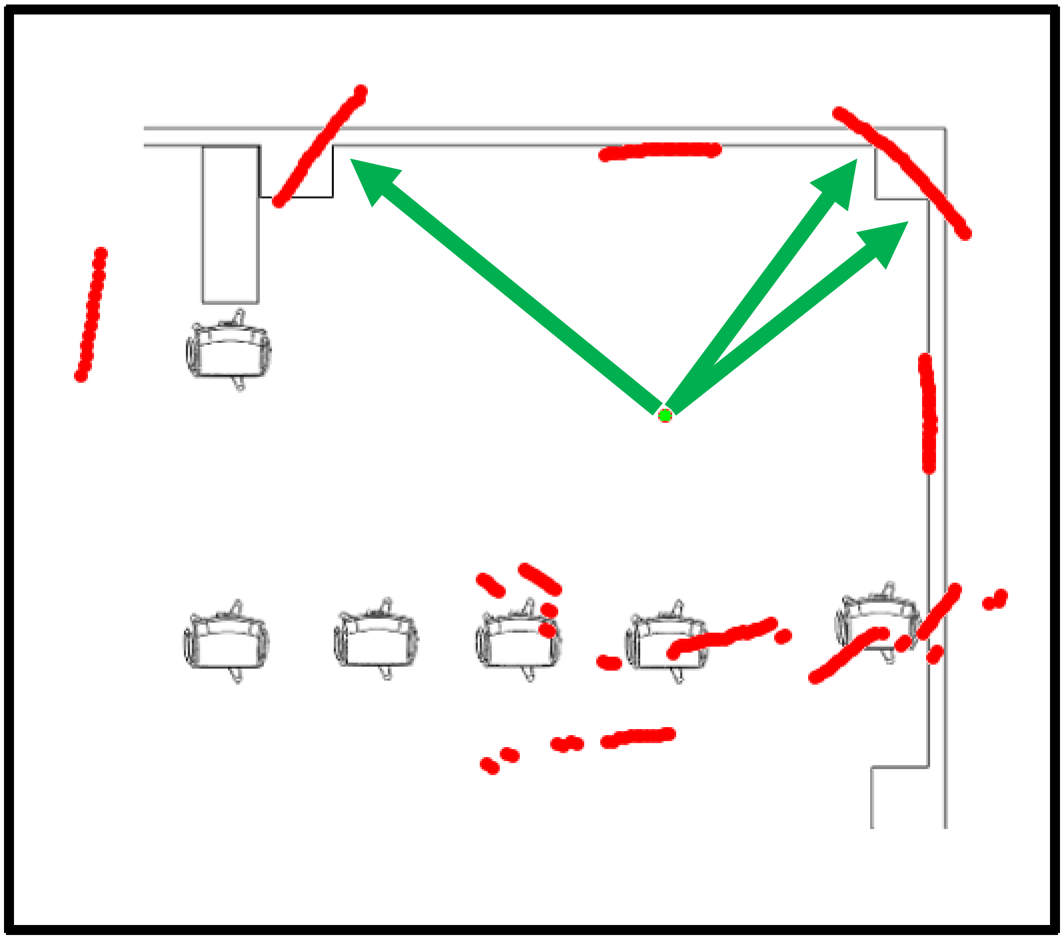
- One sonar value stored for each of 360 degrees, as the robot spins in-place
- Green dot is robot location (farther from wall than you will be)



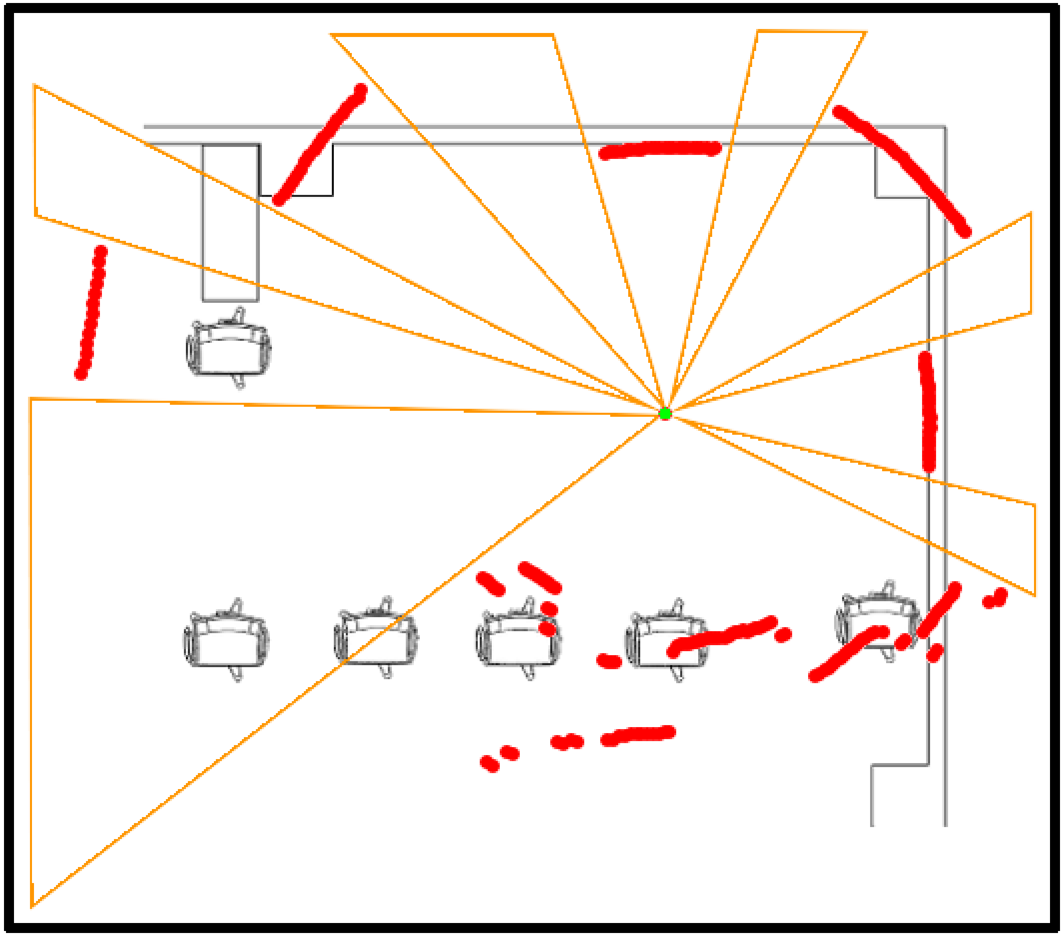
Perpendicular Flat Walls



Corner Reflectors

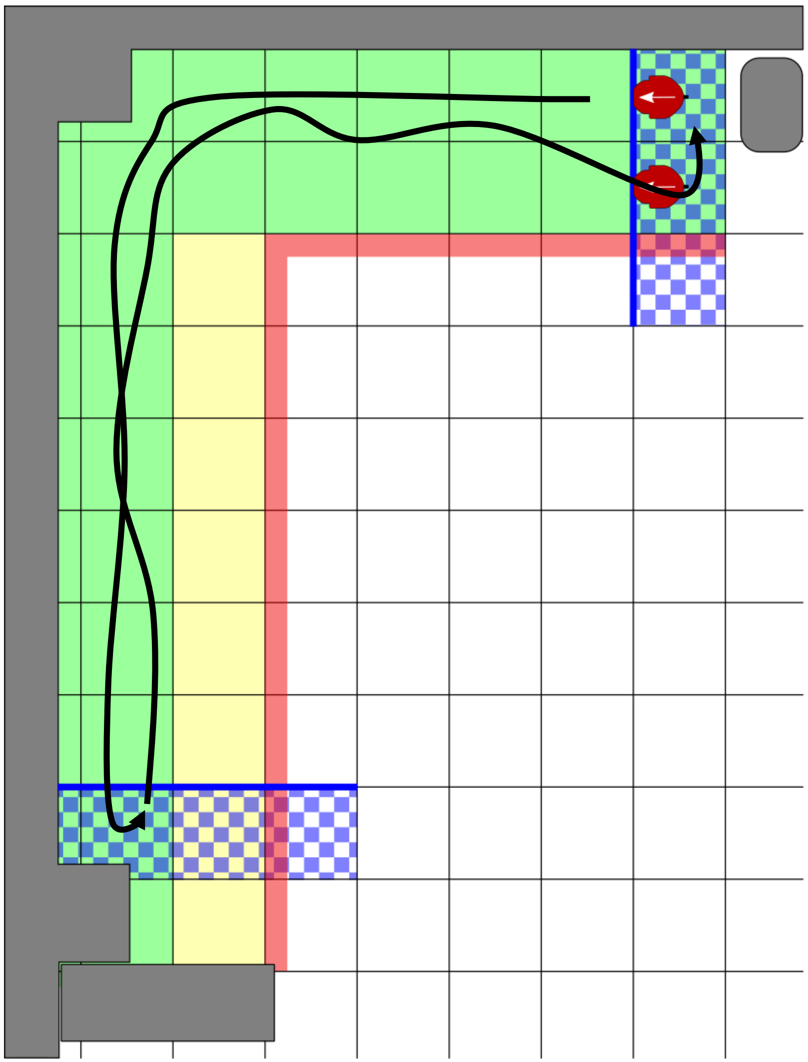


No Data



The robot <u>did</u> scan in these directions, but it got no return ping

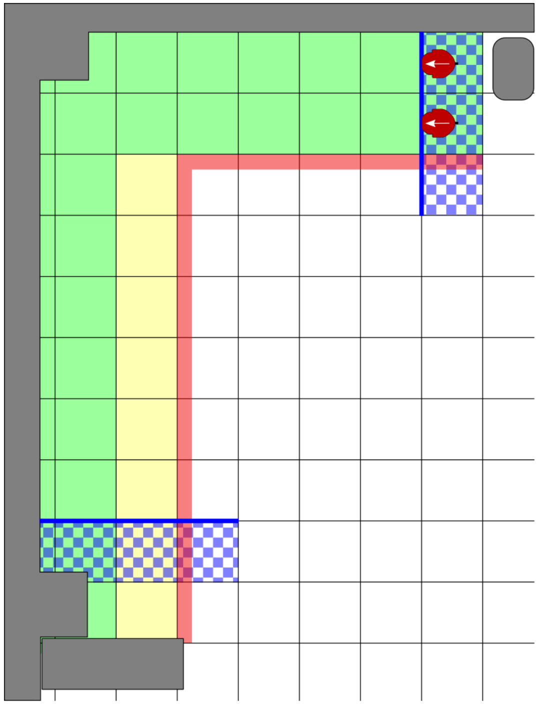
Your Design Task for Spring 2019



- Start in the top checkered area
- Go back and forth between the checkered areas as many times as possible within two minutes
- Grid represents carpet tiles
- Robot should remain in green (or checkered green) squares for maximum score
- The green/yellow/checkered squares will be clear of people, chairs, etc. The rest of the area is undefined.

Scoring and Demo “runs”

- A “leg” is traversal from one checkered area to the other
- Positive points for each completed leg (either direction)
- Reduced points or no points for a leg where robot crosses red line (TBD)
- Reduced points for entering yellow
- Reduced points for wall collisions
- A single two-minute attempt is a “run”
- Each team gets three runs
  - Probably will drop the lowest run



Design Space (factors that drive design choices)

<u>Reliance on odometry</u>:

- How long can you rely on odometry?
- Is odometry useful at all?

<u>Integration of sonar and odometry</u>:

- Can sonar inform/update odometry?
- Can odometry assist with optimal use of sonar?

<u>General</u>:

- Speed and type of robot movement (e.g., continuous vs. stop-and-turn)
- Use of one sonar vs. multiple sonars
- Control method (next slide)
- Close to wall? (risking collision) or far from wall? (risking exit of green squares)
- ?? (That’s why you brainstorm)

Control for wall following

- You don’t need to be a control theory expert to develop a valid technique
- For similar tasks, 2031 teams often choose a fixed desired distance and measure using the side sonar
  - If the measured distance is too large, turn that way
  - If the measured distance is too small, turn the other way
- This results in a swerving movement
  - Swerve too far and the sonar no longer points toward the wall, often resulting in crashes
- It's a valid approach to this problem, but our hope is that by focusing on wall following, we'll see some faster and more robust solutions

What is reasonable?

- This is deliberately open-ended
- There is no “perfect result,” and no “this score will earn this grade”
- Your peers are in the same situation
- Do not overreach (and over-propose)
  - Proposing a “perfect solution” is doomed
- DO propose a <u>progressive</u> path with incremental performance improvements

Time management

- Focus on how you can build towards a certain result by completing smaller tasks.
- If you spend "normal" 2031 time and use that time wisely, you'll end up with an acceptable project
  - Typically, most 2031 projects are split A/B, with only a very few C grades
  - A conscientious effort is what we expect, and no more time in lab than you would normally spend (splitting effort among the team)

Project tasks vs. tasks in Labs 1-8

- Replace time spent on prelab work with preparation
  - Adjust your overall plan according to what you've finished and how much time you have left
  - Plan how you will use robot time in lab
- Use lab time to make incremental progress
- The time previously spent assembling lab results can be used to assess current progress

Lab activity

- Do not all work on one piece of code
  - One or two team members can code
  - One or two can run tests
  - Some can work on deliverables like the proposal
- Open hours are still available
  - Offered for convenience, not because we require you to use them
  - Maintain a balance between this class, other classes, and personal time

Robot Logistical Details

- ONE robot per team
- Check out robots with a BuzzCard
  - You will get a long USB cable as well
- During your lab section, you will always have a robot available to you
- During open hours, robots are first-come first-served
  - With time limits imposed if robots are in high demand

Information on DE2Bot

- The downloadable **DE2Bot Manual** includes many details about the robot and how to use it.
- If you want to know something more, ask on Piazza

Name	IO Address	IN/OUT	Description
SWITCHES	0x00	IN	Read DE2 switch
LEDS	0x01	OUT	Write to DE2 LE
TIMER	0x02	IN/OUT	Read 10Hz timer
XIO*	0x03	IN	Read PB3-PB1,
SSEG1	0x04	OUT	Write to left 4-di
SSEG2	0x05	OUT	Write to right 4
LCD	0x06	OUT	Write to LCD (16
XLEDS	0x07	OUT	Write to DE2 LED
BEEP	0x0A	OUT	Write 1-7 for be
CTIMER	0x0C	OUT	Configurable tir
LPOS*	0x80	IN	Read the current
LVEL*	0x82	IN	Read the curren
LVELCMD*	0x83	OUT	Write the desir
RPOS*	0x88	IN	Read the current
RVEL*	0x8A	IN	Read the curren
RVELCMD*	0x8B	OUT	Write the desire
I2C_CMD*	0x90	OUT	Write configurati
I2C_DATA*	0x91	IN/OUT	Read or write d

Project Starting Point

- You will have a complete SCOMP
  - Implements all instructions in Table 7.1 of lab manual
  - Implements additional instructions detailed in robot manual
  - Implements a 10-level subroutine call stack
  - Has twice as much program memory (2048 words)
  - Supports hardware interrupts from four sources
- You will have a complete DE2Bot Quartus project
  - Has working interfaces with all DE2 I/O (switches, LEDs, etc.)
  - Has an additional DE2 I/O device working (LCD)
  - Has the full complement of robot I/O devices
- You will have some example ASM code and useful subroutines (Arctangent, Pythagorean distance, Multiplication, Division, Modulus)

Next Week in Lab

- First, there will be the second practical exercise
- There will be one last pre-lab quiz
  - Covering these slides, and any other project material on Canvas
- You will have some specific things to do
  - Form groups and share your brainstorming ideas
  - Learn to use the robot self-test
  - Implement some basic robot movement
- If you complete the exercises before your lab period is over, don’t waste that extra time
  - You have four lab sessions (including next week) and some open hours to complete this project
  - The lab will be busy in the last few days
    - Robots will be rationed – not guaranteed outside of your section
    - Don’t count on completing significant work during that time

Project Phases and Key Dates

- Introductory exercises (next week in lab)
  - Form project groups
  - Complete guided tasks (previous slide)
- Proposal presentations April 1st – 5th
  - Incorporate brainstorming ideas into a polished presentation
- Work on project in your lab section and as needed in open hours
  - After next week's guided tasks, you decide how to spend your time
  - Keep a design logbook, which will be used for the design summary
- Complete your design by April 14th
  - You <u>will not be able to work in the lab after this day</u>
- Final demonstrations in lab April 15th – 18th
  - Demonstrate your solution in your section
- Turn in final design summaries by the following Tuesday, April 23rd at 3:00 PM (on Canvas)

Project Schedule

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
	10-Mar	**11-Mar**	**12-Mar**	**13-Mar**	**14-Mar**	**15-Mar**	**16-Mar**
Lab Activity	OPEN HRS	Project Initial Exercises				CLOSED	CLOSED
Lecture Topic					Project and Proposal		
	17-Mar	**← SPRING BREAK →**					**23-Mar**
	24-Mar	**25-Mar**	**26-Mar**	**27-Mar**	**28-Mar**	**29-Mar**	**30-Mar**
Lab Activity	**CLOSED**	Project Work				CLOSED	CLOSED
Lecture Topic					In-class exam		
	31-Mar	**1-Apr**	**2-Apr**	**3-Apr**	**4-Apr**	**5-Apr**	**6-Apr**
Lab Activity	OPEN HRS	Project Work				CLOSED	CLOSED
Lecture Topic					Project Summary Tips		
	7-Apr	**8-Apr**	**9-Apr**	**10-Apr**	**11-Apr**	**12-Apr**	**13-Apr**
Lab Activity	OPEN HRS	Project Work				CLOSED	CLOSED
Lecture Topic					No Lecture		
	14-Mar	**15-Apr**	**16-Apr**	**17-Apr**	**18-Apr**	**19-Apr**	**20-Apr**
Lab Activity	OPEN HRS	Project Demos				CLOSED	CLOSED
Lecture Topic					No Lecture		