Final Design Project: Object Detection and Mapping

Sensing Objects with Sonar

ECE2031 Spring 2016

Motivation

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

Project Components

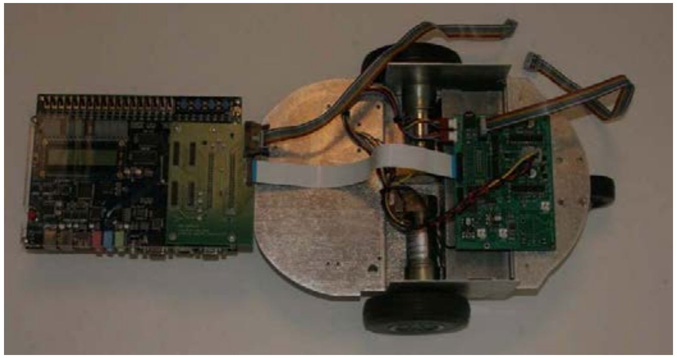
- Propose a solution to a problem
  - Problem specifics explained later in this presentation
- Implement the proposed design on the DE2Bot
- Demonstrate, present, and document your solution

Technical Communication Aspects

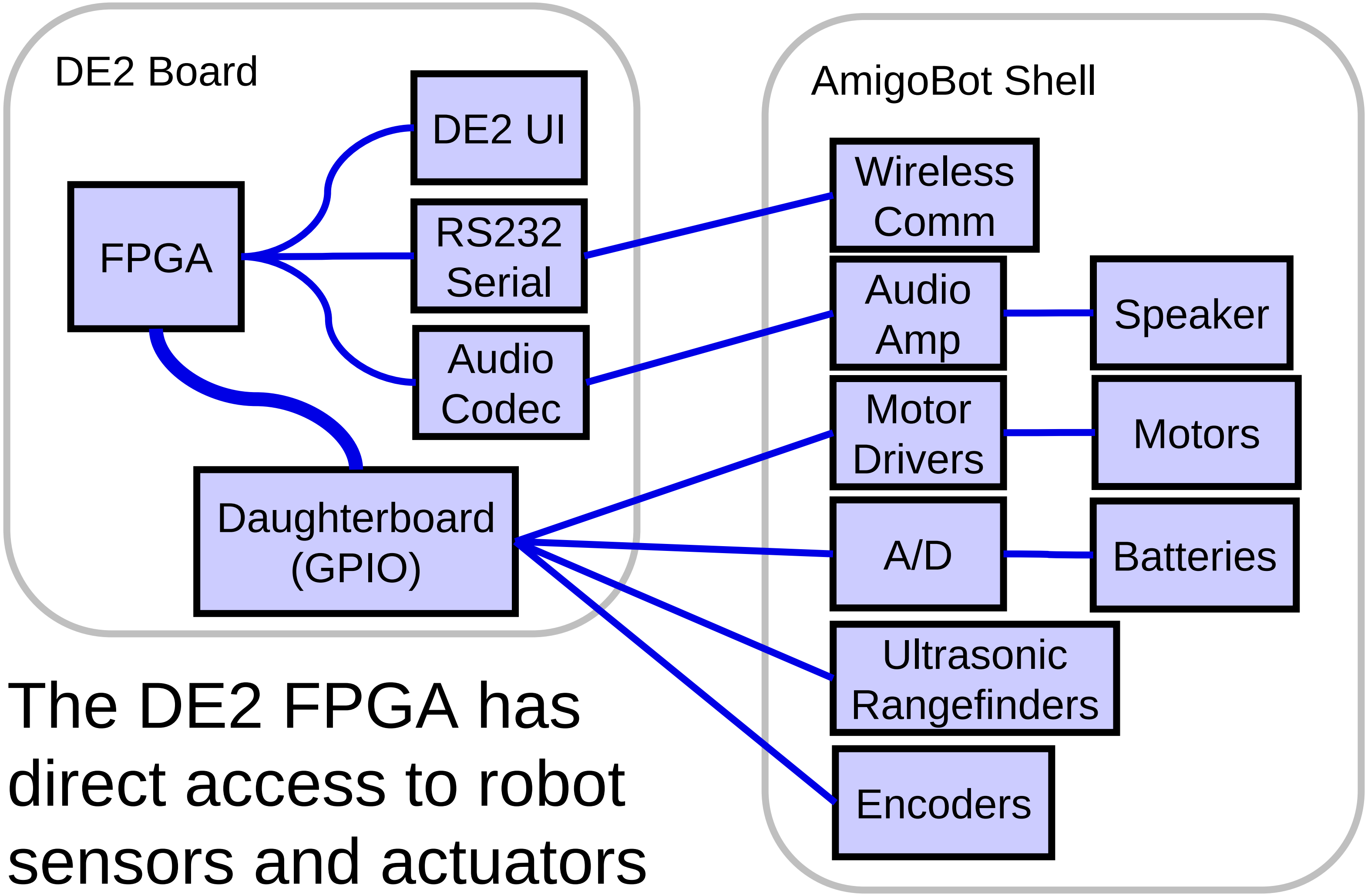
- Your project includes three major UPCP assignments:
  - A proposal outlining what you intend to develop
  - An oral presentation of your design and results
  - A final design report
- You will also maintain a design logbook using forms provided by the UPCP
  - Specific requirements will be detailed on Piazza

Background on DE2Bot

- In Summer 2010, older lab robots were gutted, adding a new internal controller board and a connected DE2 on top
- Beginning Fall 2010, each semester a new capability has been added, or a new application has been demonstrated

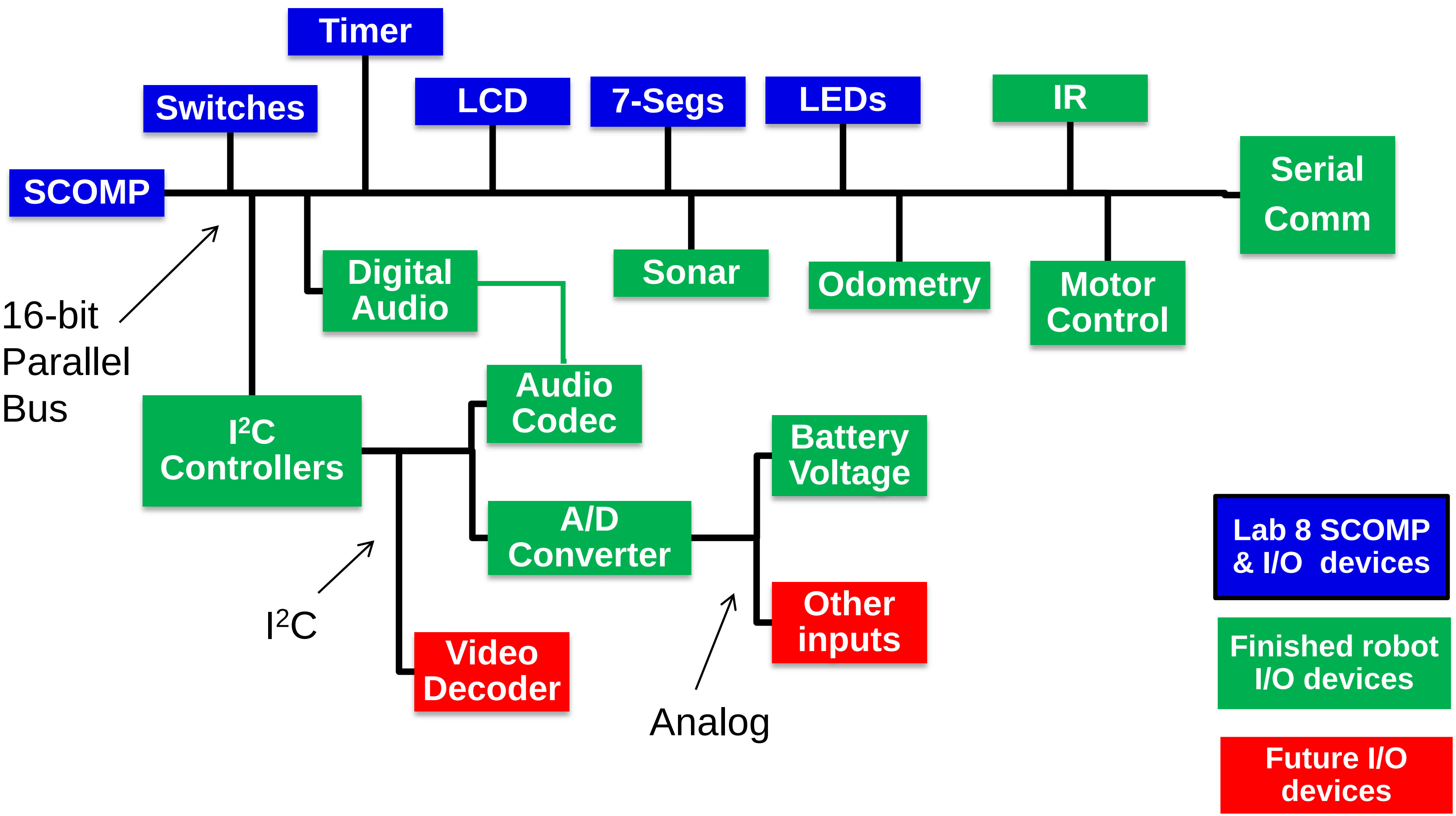


DE2Bot Hardware Architecture



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

DE2 and FPGA System Architecture



ECE2031 DE2Bot Past Projects

- Basic robot interface (3 projects related to motors, sensors, and I2C)
- Robot control (2 projects related to velocity control)
- Robot Self-test
- Odometry (position estimation from wheel rotation)
- Implementation of hardware interrupts for SCOMP
- Complex mathematical functions in software (ATAN)
- Communication (sound generation, infrared remote control, wireless data interface)
- Navigation (wall-following, area recognition, 2 projects in point-to-point movement)
- **This semester: Sensing objects using sonar**

Position Control in the DE2Bot

- You cannot control the robot's position directly
  - i.e., there is no goto(x, y) function
- You can read the cumulative rotation counter of the wheels
  - Has the benefit of giving you straight-line distance travelled, regardless of heading
- You can read the (X,Y,θ) estimate from the odometry module – discussed later
- The only things you can control are the velocities of the left and right wheels

“Open Loop” Movement

- Ideally, the robot would move in a straight line if both wheels receive the same velocity command
  - Ideally your car would drive in a straight line without you holding the steering wheel
- In reality, variations in motors, drive trains, and wheels will cause the robot to drift
- Most of the drift is detected by the odometry, and can thus be corrected
  - Using feedback from the odometry to correct drift would be “closed loop” control.

Odometry in the DE2Bot

- The wheel position encoders provide total distance moved by each wheel. They can not keep track of X, Y, and heading of the robot as a whole.
- A separate I/O device (from ECE2031 Summer 2012) added this capability:

$$x_i = x_{i-1} + \Delta U_i \cos \theta_i$$
$$y_i = y_{i-1} + \Delta U_i \sin \theta_i$$

where

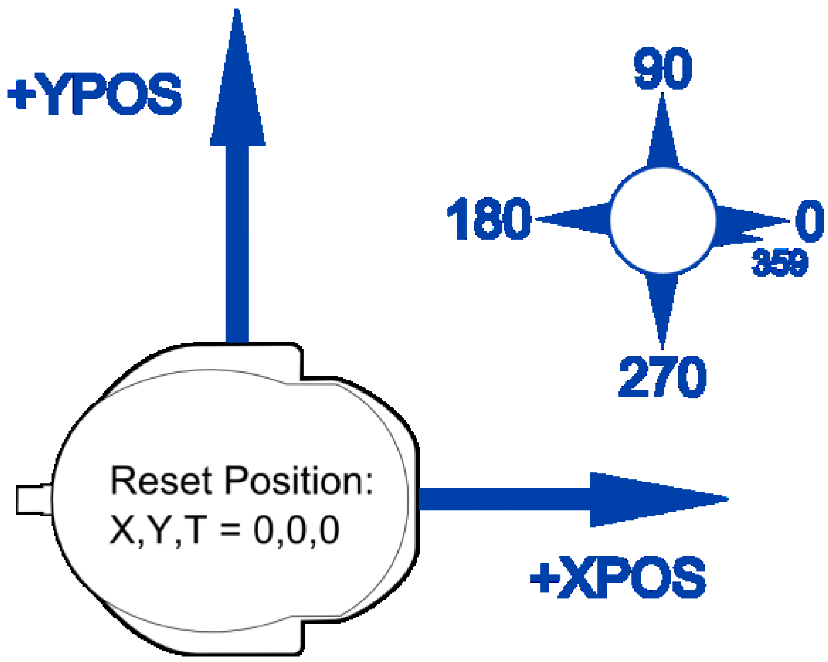
θ_i = heading of robot at instant i,

ΔU_i = distance travelled in i^{th} interval,

x_i, y_i = relative position of the robot's centerpoint c at instant i.

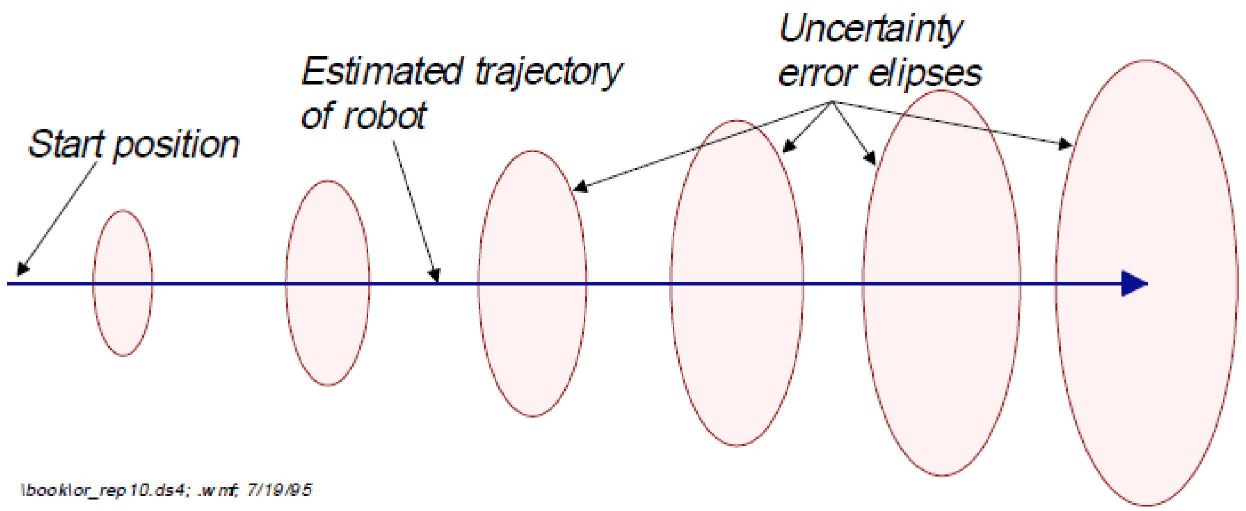
Odometry Coordinate System

- <u>At reset</u>, the front of the robot is defined as the positive X direction, and positive Y is to the robot's left.
- θ increases CCW, as would be the normal convention.
- For as long as you don't reset it, the odometry module will estimate movement relative to the reset position.



Limitations of Odometry

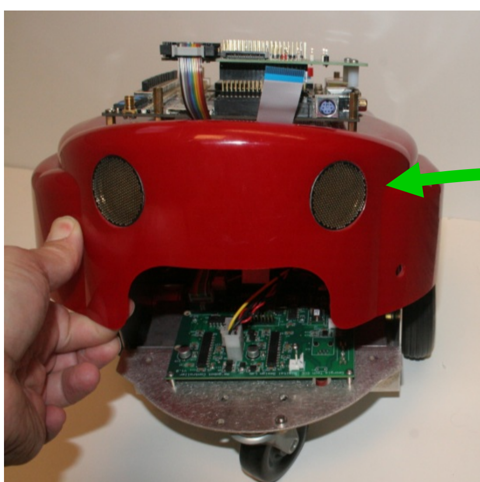
- Errors accumulate as robot moves
- Inaccurate heading (orientation) results in incorrect incrementing of X, Y components



J. Borenstein, *Sensors and Methods for Mobile Robot Positioning*, 1996.

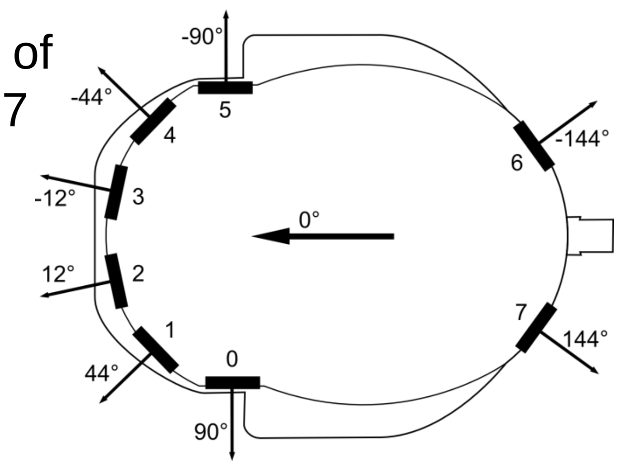
Sonar Transducers in the DE2Bot

- The sonar transducer emits a ping (a “chirp” at 50 kHz, actually) when a high voltage square wave is applied across its two terminals
- This is ultrasonic, but you will hear a click as a side-effect
  - “Ultrasonic ranging” and “sonar ranging” are often used synonymously
- Transducers produce a small signal when an echo is detected
- Time-of-flight is proportional to distance between the transducer and whatever reflected the ping



One of 8 transducers

Locations of Sonars 0-7



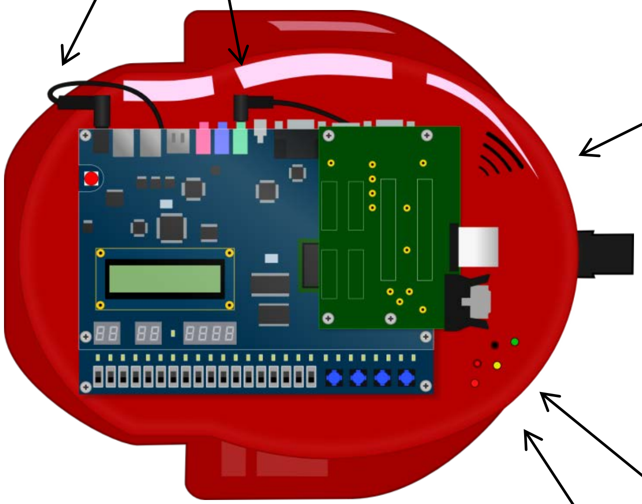
DE2Bot I/O Devices

- Your program interacts with the robot using IN and OUT instructions to peripheral devices
- The downloadable **DE2Bot Manual** includes detail about every I/O device

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

Robot Operational Details

Keep cables plugged in

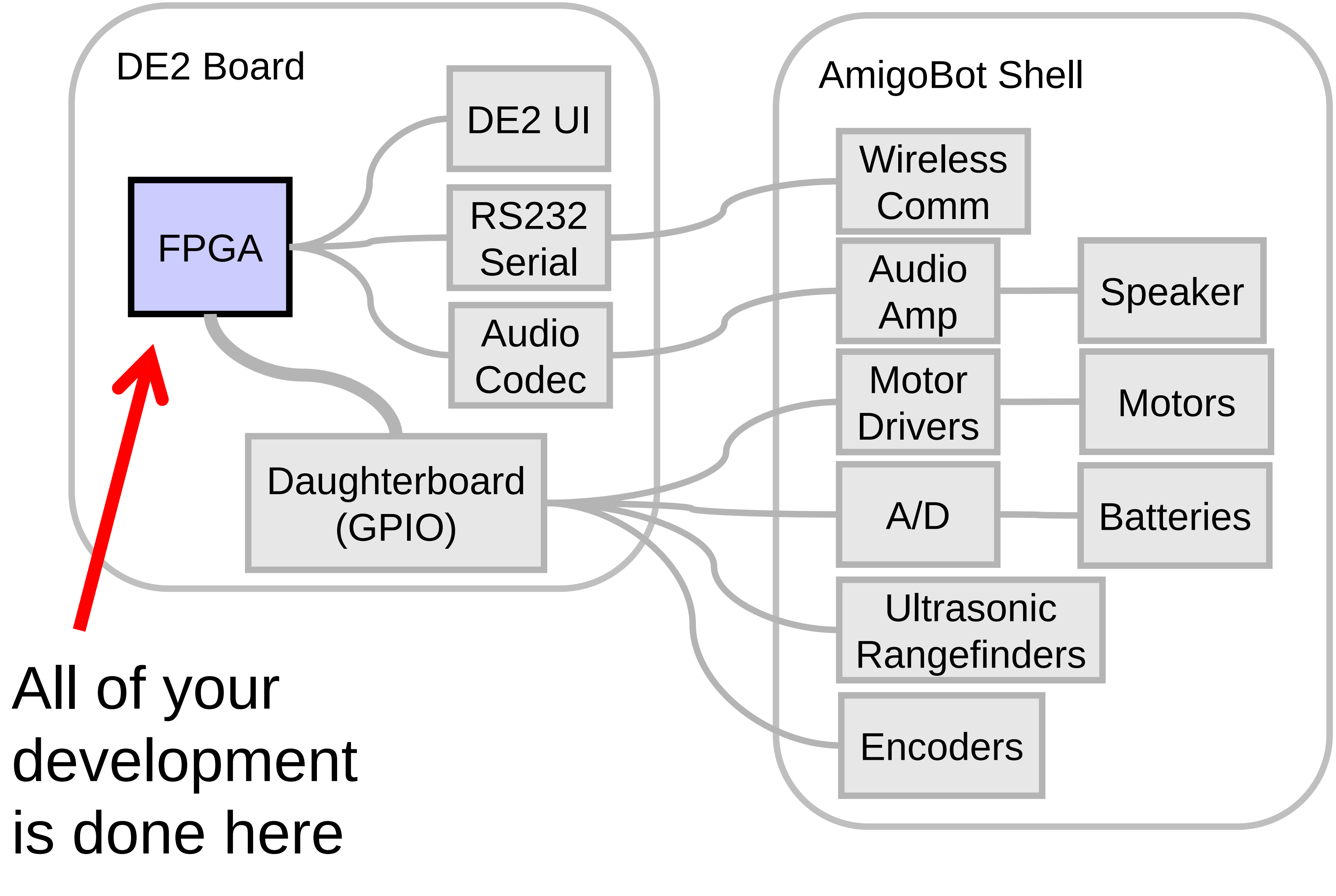


- ONE robot per team at a time
- Teams in the current section ALWAYS get a robot
- Power switch is on bottom (directly beneath speaker)
  - When robot has power, DE2 has power, too
- Turn robot off when not in use, to conserve battery
- LEDs indicate hardware status (see manual for details)
- Red and black pushbuttons are not used at this time

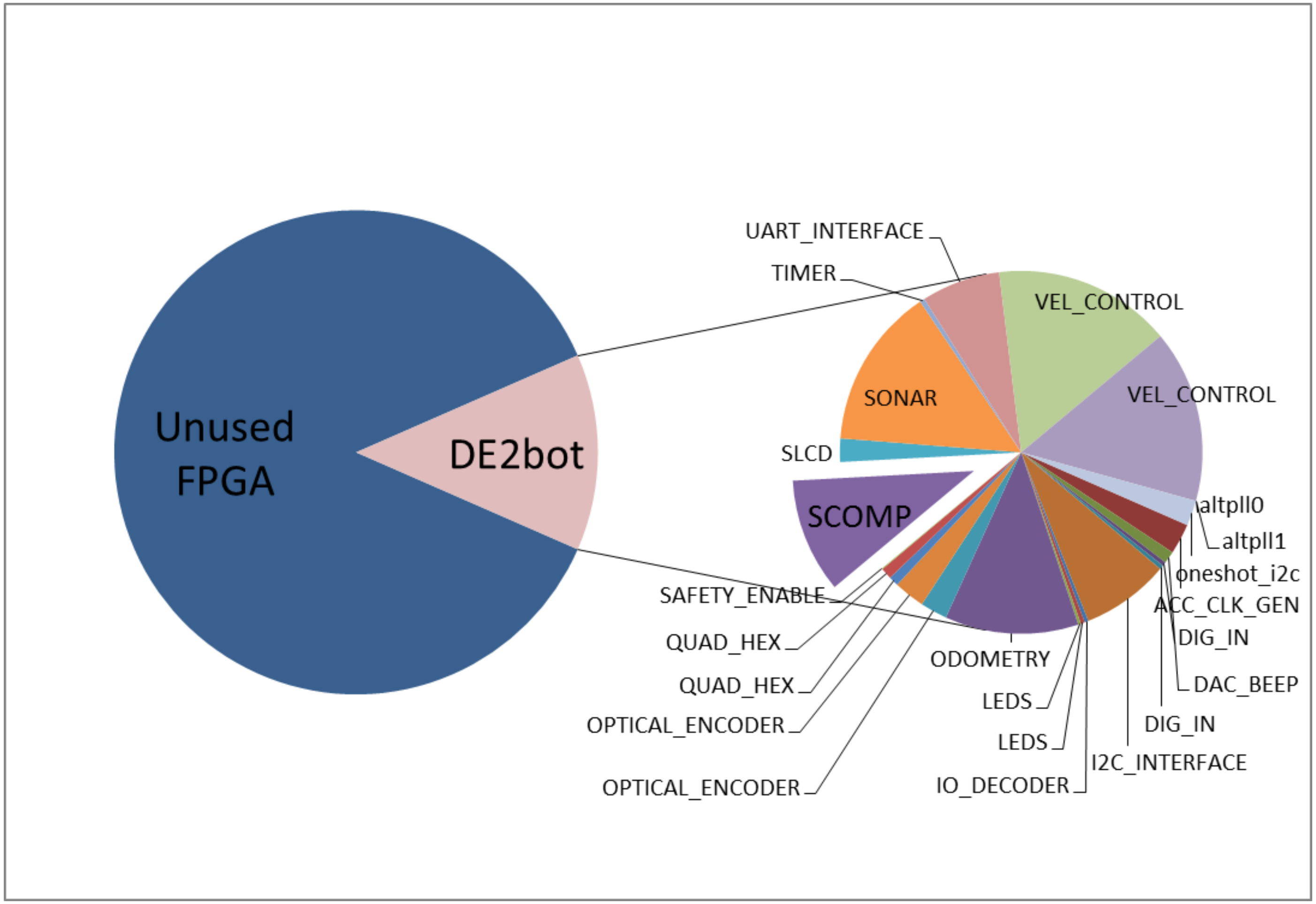
Working with Complex Systems

- When something doesn't work, the robot gets blamed
- It is true – equipment CAN fail, but almost all problems are user-related:
  - FPGA design (possibly in bdf, possibly in VHDL)
  - SCOMP code (assembly errors can be elusive)
  - Careless errors (code not assembled, some VHDL not compiled, variables not initialized, something not reset)
- Use the robot self-test to identify any hardware problems with a robot, and show this to a TA

Project Development

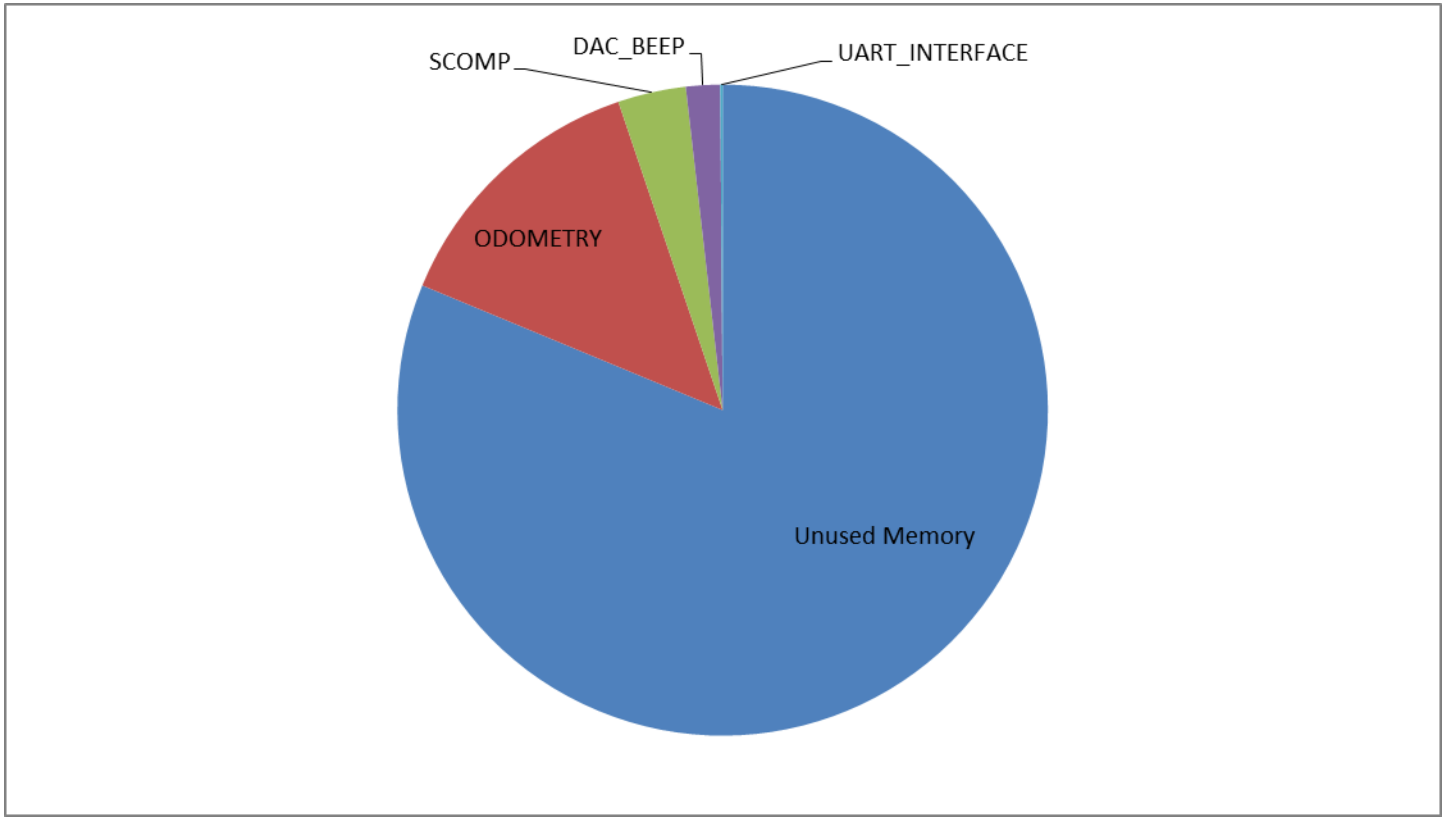


Inside the FPGA Hardware Block



FPGA logic cell usage by component

Memory resources



FPGA on-chip RAM usage by component

(Odometry uses a lot for big sine and cosine look-up tables)

Project Motivation

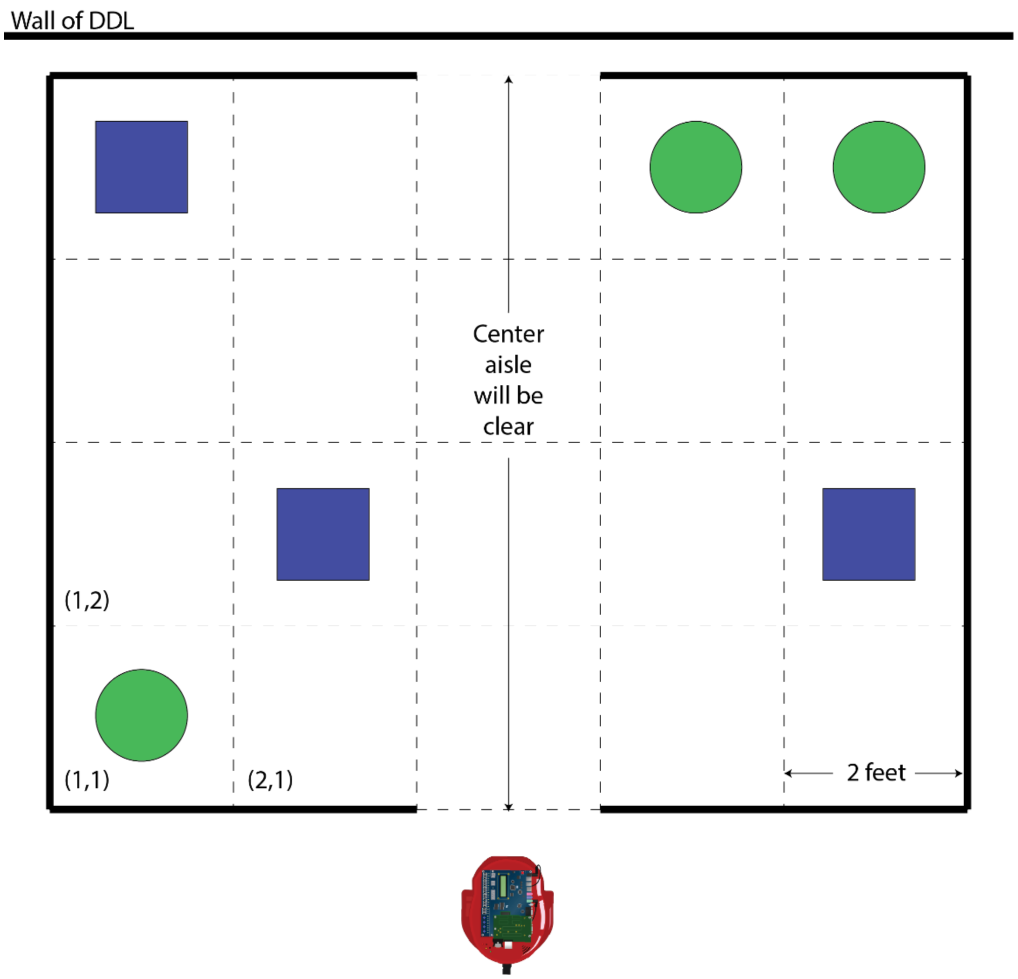
- Previously, sonar has been used only in a limited sense
- The few related projects have explored:
  - Wall following (maintaining a near-constant sonar reading on one side of the robot as it moves)
  - Recognizing large areas as being “different”
  - Avoiding collision (in multiple projects)
- Now, it’s time to use sonar to find objects
- A combination of intelligent sensing strategy and efficient movement will be required to maximize your performance

Your Design Task for Spring 2016

- Given an arena with certain locations that may be occupied by objects, use sonar to find all the present objects in a limited time
  - Counting them is sufficient for significant points
  - Describing their grid location results in greater points
- Demo score will be based on number of objects identified, location accuracy, speed, and lack of collisions

Sonar sensing arena

- 8 ft. by 10 ft. (approx.) arena bounded by short walls with two gaps
- 3-6 objects placed near center of grid cells
- Two different types of objects
  - Cube
  - Cylinder
- Robot starts outside arena

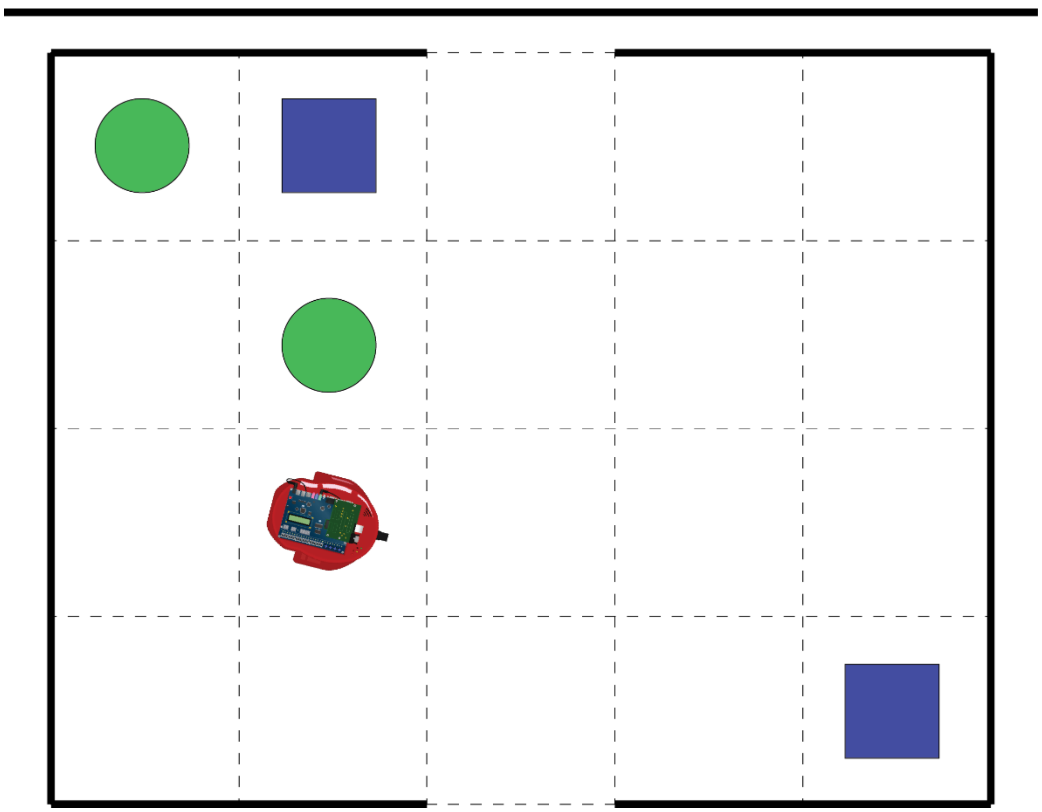


Characteristics of sensed objects

- Cube and cylinder will soon be defined
  - Cylinder will probably be 5-gallon plastic bucket
  - Cube will probably be about 1 ft x 1 ft. x 1 ft. (or slightly smaller), in a material suitable for sonar sensing
- Arena walls are particle board
  - Easily sensed, except perhaps from shallow angles

Sensing strategy

- Any scenario is likely to have partially occluded objects
- Exploration by the robot may be necessary
- Every object is visible from at least one side
  - None are surrounded by other objects



Robot runs

- Each team gets three runs
- Number and location of objects varies in each run
- Each run lasts 60-90 seconds (still TBD)

Scoring generalities

- Accurately counting number of objects increases score
  - No extra credit for identifying cube vs. cylinder
- Being able to send coordinates results in higher score
  - Assembly code to send (x, y) data over wireless communication will be provided for you
- If not all time is used, additional points are given
- Collisions will result in point penalties
  - Small penalty for hitting objects
  - Greater penalty for hitting boundaries

Project Demo Score

- You will demonstrate your results on the last day of lab
  - This is separate from your oral (PowerPoint) presentation, which is also on that day
- The demo provides a **raw** score for each group
  - Scoring details provided next week
- All raw scores get compiled to rank teams for a 500-point **demo** score (i.e. “it gets curved”)
  - This is the 500-point “project demo” grade you see on the syllabus

Design Space (factors that drive design choices)

Sensing technique:

- Stop and turn 360 degrees while scanning?
- Scan while exploring?

Movement:

- How to enter arena safely?
- Attempt to do everything from center aisle? Or move throughout arena?

Hardware modifications:

- Is there anything useful to add to SCOMP or the peripherals?

General procedure:

- Will you change anything between runs?
  - You will not be allowed to interact with the robot once object locations are known.

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 an additional DE2 I/O device working: the LCD
  - Has the full complement of robot I/O devices
- You will have two example ASM programs
  - An introduction to the robot, for your exercises next week
  - A basic project starting point

Project Phases and Key Dates

- Introductory exercises (next week in lab)
  - Focus on understanding DE2Bot and its capabilities.
- Get more details, including proposal tips, in lecture next week
- Turn in proposal on April 7/8 at start of either lecture
  - That's three in-session weeks away, but you should have 3/4 of your design finished by then.
- Complete your design by April 18
  - You will not be able to work in the lab after this day.
- Final demonstration April 19-21
  - Demonstrate your solution in your section. Points for your demo will factor into your grade.
  - Make a PowerPoint presentation, explaining your design, what worked, & what didn't.
- Turn in final report by the following Monday, April 25

Project Schedule

	13-Mar	**14-Mar**	**15-Mar**	**16-Mar**	**17-Mar**	**18-Mar**	**19-Mar**
Assignments Due				Lab 8 Results			
Lab Activity	OPEN HRS	OPEN HRS		Project Initial Exercises*		OPEN HRS	CLOSED
Lecture Topic						Proposals & Exam Review	
	20-Mar	**21-Mar**	**22-Mar**	**23-Mar**	**24-Mar**	**25-Mar**	**26-Mar**
Assignments Due				NONE			
Lab Activity	CLOSED			Spring Break			CLOSED
Lecture Topic							
	27-Mar	**28-Mar**	**29-Mar**	**30-Mar**	**31-Mar**	**1-Apr**	**2-Apr**
Assignments Due				Logbook Checks*			
Lab Activity	CLOSED	OPEN HRS		Practical Exercise #2 & Project Work*		OPEN HRS	OPEN HRS
Lecture Topic						Written Exam	
	3-Apr	**4-Apr**	**5-Apr**	**6-Apr**	**7-Apr**	**8-Apr**	**9-Apr**
Assignments Due				Logbook Checks*		Proposal	
Lab Activity	OPEN HRS	OPEN HRS		Project Work*		OPEN HRS	OPEN HRS
Lecture Topic						Oral Presentation Tips	
	10-Apr	**11-Apr**	**12-Apr**	**13-Apr**	**14-Apr**	**15-Apr**	**16-Apr**
Assignments Due				Logbook Checks*			
Lab Activity	OPEN HRS	OPEN HRS		Finish Project		OPEN HRS	OPEN HRS
Lecture Topic						Report Tips	
	17-Apr	**18-Apr**	**19-Apr**	**20-Apr**	**21-Apr**	**22-Apr**	**23-Apr**
Assignments Due				Logbook Checks*			
Lab Activity	OPEN HRS	OPEN HRS		Project Demos		OPEN HRS	OPEN HRS
Lecture Topic						No Lecture	

Next Week in Lab

- You will have some guided exercises to perform
  - Robot self-test
  - Basic robot movement
  - Sonar reading
  - Associated check-offs count towards your Logbook grade
- If you complete the exercises before your lab period is over, **don’t waste that extra time**
  - You have four lab sessions (<u>including</u> next week) and a handful of open hours to complete this project
  - The lab will be **full** in those last few days
  - Robots will have to be rationed
  - Don’t be part of the herd

Clarifications

- Additional announcements and clarifications and will be posted **on Piazza**
  - You are responsible for information posted there
  - Could include changes to rules or assignments
  - Make sure you are monitoring it!
- Use Piazza to ask questions
  - If a general question is asked, everyone can benefit from the answer
  - If your question contains details specific to your design, you can limit the visibility to only instructors.
  - Grey areas of rules will not necessarily be interpreted in your favor at the last minute. Ask for clarification.