UNIT 2
Introduction to Robotics

UNIT OVERVIEW

UNIT NUMBER: 2 **DURATION: 20 hours**

SUMMARY

One important skill that students will learn in this course is design thinking. They will use the design process to brainstorm, plan, and implement a robot design. Due to the complex nature of robotics, without prior experience it can be hard to imagine a robot and subsequently to create a design. This unit will walk students through creating the basic pieces of a robot to familiarize them with the various parts and processes. They will not be told exactly how to build the parts, but tutorials will be provided at each step to provide guidance. They will be graded weekly based on their engineering notebook and Self and Peer Evaluations. Then they will move on to design and implement their own robot in the following units.

INSTRUCTIONS

1. Introduce the Engineering Notebook
2. Divide the class into two groups:
    1. Mechanical/Electrical Engineers
    2. Programmers
3. Groups will work simultaneously (and collaboratively) to create a basic robot.
4. The groups should first test the individual components before doing any building or programming.
5. Then the groups will begin building the robot. The Engineering team will complete the Mechanical/Electrical steps, while the Programming team will

LEARNING RESOURCES

Below are general learning resources. Specific tutorials related to this unit are in production and will be added when ready.

Engineering Resources

Modern Robotics Hardware Tutorials
Rev Hardware Tutorials
PushBot Guide
https://www.firstinspires.org/resource-library/ftc/robot-building-resources
Modern Robotics Core Device

complete the Programming steps. Many steps will require collaboration between the two teams.

Engineering Notebook Resources

Engineering Notebook Guidelines and Self-Assessment
Engineering.com Engineering Notebook Explanation

Discovery
How to Solder
Modern Robotics Sensors

Programming Resources

FIRST Tech Challenge Specific Resources
Java Tutorials
Blocks Tutorials
FIRST Tech Challenge GitHub Code

Java Resources
Codecademy Java Course
Learn Java Online

ASSESSMENTS:

1. Weekly Engineering Notebook
2. Weekly Self & Peer Evaluation

TOOLS & MATERIALS

1. Tetrix Kit of Parts or Rev Kit of Parts
2. Electronics parts
3. Soldering iron
4. Saws
5. Wrenches
6. Drill
7. Allen wrenches
8. Any and all power tools available

STANDARDS ADDRESSED:

Full course standards alignments can be found here.

INTRODUCTION TO THE ENGINEERING NOTEBOOK

What is an Engineering Notebook?

An engineering notebook is formal documentation of the work of an engineer or engineering team on a specific design project. Engineers record ideas, inventions, experimentation records, observations and all work details in the engineering notebook. An engineering notebook can be used as a legal document to show the origin and evolution of an idea when seeking a patent.
One of the goals of *FIRST®* and *FIRST®* Tech Challenge is to recognize the engineering design process and the journey that a Team makes during the phases of the problem definition, concept design, system level design, detailed design, test and verification, and production of the robot. Throughout the process of designing and building a Robot, Teams will come across obstacles, lessons learned, and the need to draw things out on paper.

- What is the agenda for today?
- Why are you meeting?
- What are the goals for today?
- What decisions did your Team make in forming the Team, creating the robot, writing the program, the outreach projects, etc.?
- Why did you make that choice when building your robot, coded the software that way, chose that group of individuals to outreach to, etc.?
- What was the impact on your Team, robot, or community when you made that decision?
- What is the next step?

Anything and everything that documents a team's design process should be a part of the engineering notebook including:

- Written ideas
- Sketches (preferably annotated)
- Work session summaries
- Research findings
- Interview information: who was contacted, when, and what was discussed or learned
- CAD printouts
- Test Results
- Photographs
- Calculations

It doesn't matter if a line of investigation leads nowhere. Documented failures provide clues to success.

Students may choose to record their season with either handwritten, electronic, or online documents.

Electronic/Online: Teams may choose to use free website creation sites or Google Sites to create their Engineering Notebook. They may also choose to use Microsoft or Open Office Tools.

Handwritten: Teams can choose from spiral-bound, laboratory, or documentation notebooks available through their school or local stationary supply store. Teams can also use the binder supplied by Rockwell Collins delivered in the Registration and Welcome Kit.

For this course, you may decide to allow your students to choose their online format or your students may be limited to the handwritten option depending on their access to computers and internet. If your teams will be competing in official *FIRST* events, please make sure to review the Engineering Notebook guidelines.

The Engineering Notebook will be the primary source of grading in this course and should be reviewed by the teacher at regular intervals (preferably weekly) to assess each team's progress. Rubrics for grading the Engineering Notebook will be provided.

MECHANICAL/ELECTRICAL STEPS

Learn about the control system

1. Components of a Robot https://youtu.be/fU1GXteByJw

Set Up the Electronics

1. Connect the Driver Station to the Robot Controller https://youtu.be/LQqcX9rBX38
2. Power and Wire Your Robot https://youtu.be/DKwpEyKbTqM
3. Configure Your Robot https://youtu.be/VHyKE3B170k
4. Test your motors https://youtu.be/SOspcH58tZQ
5. Test your servos https://youtu.be/N6YulRyLmCQ
6. Test your sensors https://youtu.be/aZYBmU6ocec https://youtu.be/VAPxENrKRvo
7. Configure Your Gamepads https://youtu.be/mAivzV7jaRQ

Build a chassis

The chassis is the base or frame of the robot. Without the chassis, the rest of the robot has nothing to support it. The engineering team's first challenge is to: https://youtu.be/iv8qvimf-3A

Use beams, screws, and nuts to create a sturdy rectangle.
The rectangle should maintain it's shape and should not skew in any direction.
A rectangle is a good start to a chassis, but it will likely not be sufficient to hold all of the motors, servos, manipulators, and sensors that will go on the robot. Therefore, the team's next challenge is to:

Construct a frame that will hold all components of the robot securely and efficiently.

HINT:
You may also use the Pushbot Build Guide for assistance.

Construct a drivetrain

The drivetrain is what allows a vehicle or a robot to move. In order to move, a robot needs motors, wheels, and axels. The team's next challenge is to: https://youtu.be/28-VLKM3Wy0

Use set screws to mount a wheel on a motor.

Now it's a little hard to move or balance with only one wheel. Repeat this challenge for as many motors or wheels the team chooses. Some robots have 2, 3, 4, or even 6 wheels! Even if you only have 2 motors, you can use omniwheels that aren't attached to motors to help the robot maneuver. It also helps if the wheels and motors are attached to the rest of the robot. The team's next challenge is to:

Mount the motor/wheel combinations onto the robot.

HINT:
You may also use the Ranger Bot Movement Building Guide for assistance.

Mount electronics

Once the team knows how to connect all of the electronics, the parts need to be attached to the robot.

Mount the electronics onto the robot.

https://youtu.be/t1hK50_66zs

Wire management is important. Parts may get damaged if the wires are loose, pinched, or twisted. Now the team should:

Check the wiring to ensure safety and secure connections.

https://youtu.be/9u3Z0XrrXKQ

Build a Manipulator

The team should now have a good feel for how to connect beams, mount and connect parts, and design a basic robot. The last piece for this module will be to add a manipulator to the robot. A manipulator is simply an addition to the robot that is used to manipulate other objects. This could be an arm, a claw, a sweeper, or anything else the team can think of to help them complete the game's challenges! Typically manipulators are controlled using servos.

Build and attach a servo-controlled arm to the robot.

HINT:
The TETRIX Arm and Gripper Build Guide provides a good example of how to construct a manipulator.

PROGRAMMING STEPS

Getting started

There are a few things the team must do before they begin programming the robot. This will keep the programming team occupied until the engineering team can build enough of the robot to allow the programming team to test their code.

Select a programming language https://youtu.be/yGOIJG6d3g0

Install the software

In order to begin programming, the appropriate software must be installed on the team's computer(s). Which software to use depends on which programming language the team is using. The instructions can be found here:

- Java Installation Instructions https://youtu.be/mMrC5kDcUN4
- Blocks Installation Instructions https://youtu.be/AKoqwqPo-Vc https://youtu.be/BL0kvoQcjxU

Learn basic programming skills
If the team is new to programming and is using Java, it may help to learn basic programming skills before jumping into robot programming. This course from Codecademy.com will give the students a great introduction.

Learn about *FIRST* Tech Challenge programming
If the team is new to *FIRST* Tech Challenge, it may be helpful to learn how the technology works.

What is a robot? https://youtu.be/yDvZHnH5t-4 **What is an op mode?** https://youtu.be/PvCTwPDyen8

Op Modes

How do I write an op mode?
https://youtu.be/7CDK5-m-vQw https://youtu.be/o-7z9WE6JAI

How do I install an op mode onto my robot?
https://youtu.be/cRcdGPaZ70Y https://youtu.be/uItHYCxT9eY

Moving

Now the team is ready to start programming the robot! Here are the first challenges the team should overcome:

Program the robot to move forward.

Program the robot to move backward.

Java https://youtu.be/dNiGzPqEF9I **Blocks https://youtu.be/f8-NQIS3IEM**

Turning

Moving forward and backward is great, but the playing field requires the robot to move on more than one dimension. Next the team should:

Program the robot to turn 90 degrees to the left.

Program the robot to turn 90 degrees to the right.

Java https://youtu.be/Xpgs3UKDiW8 **Blocks** https://youtu.be/PuOKjimrJ0k

---

Implement Gamepads

When using a gamepad, it is very useful to be able to move the robot at multiple speeds. Bring on the next challenge!

Program the robot to adjust its speed based on the position of the remote control's joysticks.

Java https://youtu.be/-w_d508l7P8

Blocks **https://youtu.be/8KXQketfarE** **Blocks** **https://youtu.be/75ndVZzZJjU** **Blocks** **https://youtu.be/tQW8ZG6B_bQ**

Rotate Servos

The gamepad must control every aspect of the robot, not just the movement. Anything controlled by a motor or servo should have a corresponding gamepad control.

Program the gamepad to rotate any and all arms, levers, pulleys, or other servo or motor controlled mechanisms on the robot.

Java https://youtu.be/-J7RNUxm_p8 **Blocks** https://youtu.be/Qd8iMiAkFGI

Drive in a square

The *FIRST* Tech Challenge game begins with a 30 second autonomous period. This means the robot must act completely on its own for 30 seconds. How is the robot supposed to know where it is or where to go? Good question! There are many different ways, but the tutorial with the challenge below shows the most basic way for beginners.

Autonomously drive along the perimeter of a 5'x5' square.

Autonomously drive along the perimeter of a 10'x10' square.

Java https://youtu.be/BLYqh_svu8Q

Helpful Tips

https://youtu.be/YDEkcunllYI https://youtu.be/fgGMq7ulXP4
