

REC Unit 2: Introduction to Vex Programming

| | |
|-------------|--|
| Catalog No. | 77-8115-0020 |
| Category | Mobile Robotics |
| Duration | 15 Hours |
| Software | VEXCode or any software that can be downloaded to the V5 controller |

2.1 A (Core): Basic Motor Control

- Introduction to Process Control
- Control by Human
- Computer Program Defined
- Writing a Program
- Three Steps

2.1 B (Activity): Programming Components

- Programming Components
- Programming in VEXcode V5 Text
- The Commands
- Worksheet: Describe a Process

2.2 (Activity): Draw a Line

- Safety Rules Review
- Connecting the VEX Robot Brain
- Programming with VEXcode V5 Text
- Downloading the Program to the Robot
- Running the Program
- Observing the Robot
- Motor Direction

2.3 (Core): Pseudo-code and Turns

- Pseudo-code
- Right Turns
- Left Turns
- Worksheet

2.4 (Activity): Make a Square

- Make a Square
- Introduction
- Create Pseudo-code
- Convert to VEXcode
- Download and Test
- Analyze Behavior, Refine Code
- Challenge Activities

2.5 (Core): Variables, Constants and Comments

- Bits and More Bits
- Numbers, Constants, and Variables
- Naming Conventions
- Variable Types
- Creating Variables and Constants in C++
- Commenting
- Worksheet: Numbers, Constants, and Variables

2.6 (Activity): Apply Constants, Variables, and Comments

- Review Code from Activity 2.4
- Replace all Fixed Values with Constants
- Use a variable for the `sleep_for()` statement
- Modify the Program
- Challenge Activity

2.7 (Core): Dead Reckoning and User Functions

- Dead Reckoning
- Functions
- Example of a User-Defined Function
- Passing Parameters to a Function
- Receiving a Return Value from a Function
- Worksheet: User-Defined Functions

2.8 (Activity): Follow a Complex Path

- Program to Drive Specified Distance
- Determining DCONSTANT
- Create a Simple Function
- Create a Function with a Parameter
- Determining the Angular Constant
- Follow a complex Path

2.9 (Core): Conditional Statements

- Introduction
- The “if” Statement
- Expressions
- The “==” Comparison Operator
- The “if-else” Statement

2.10 (Activity): Modifying the GoForward Function

- Structure the Go Function
- Copy your Code
- Reverse

2.11 (Core): Loops

- “While” Loop
- “For” Loops
- Nesting

2.12 (Activity): Make Multiple Squares

- Using “while” Loops
- Infinite “while” Loop
- Conditional Loop
- Using “for” Loop
- Replacing the “while” Loop
- Create a Square Wave
- Create a Simulated Sin Wave

2.13 (Core): Simplified Symbols, Logical Operators, and Integer Math

Incrementing and Decrementing

C++ Shorthand Expressions

Integer Math

The MOD Operator

Wrapping

Signed and Unsigned Integers

Worksheet

2.14 (Project): Fine Motor Control

Introduction

Mack Your BaseBot Unstable

Write Your Program

Fine Motor Control - Conclusion