Lesson Name: Cutebot Remote Control

School District:

Teacher Name:

Intended Grade Level: All

Differentiation: N/A

Accommodations:

Make accommodations appropriate for your classroom.

Material(s):

- Computer with internet access
- 2 Micro:bits with USB cable
- 1 cutebot with battery pack
- 1 battery pack for micro:bit

Time allotment (varies according to grade level):

- TBD

Alignment with CODERS module(s):

(Cutebot)

MLS Standards:

4.AP.M.01 – Decompose (breakdown) large problems into smaller, manageable subproblems to facilitate the program development process.

5.AP.M.01 – Decompose (breakdown) large problems into smaller, manageable subproblems and then into a precise sequence of instructions.

4.AP.A.01-Compare and simplify multiple algorithms (sets of step-by-step instructions) for accomplishing the same task verbally and kinesthetically, with robot devices or a programming language.

5.AP.A.01-Compare and simplify multiple algorithms (sets of step-by-step instructions) for accomplishing the same task verbally and kinesthetically, with robot devices or a programming language, then determine which is the most efficient.

4.AP.C.01-Create a program using control structures (e.g., sequence, conditionals, interactive-looping) to solve a problem or express ideas both independently and collaboratively.

5.AP.C.01-Create a program using control structures (e.g., sequence, conditionals, interactive-looping), event handlers and variables to solve a problem or express ideas both independently and collaboratively.

4.AP.M.02 – With grade appropriate complexity, modify, remix or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.

5.AP.M.02 – With grade appropriate complexity, modify, remix or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.

4.AP.PD.01 – Use an iterative and collaborative process to plan the development of a program that includes user preferences while solving simple problems.

5.AP.PD.01 – Use an iterative and collaborative process to plan the development of a program that includes other perspectives and user preferences while solving simple problems.

4.AP.PD.03-Analyze, create and debug a program that includes sequencing, repetition, conditionals and variables in a programming language.

5.AP.PD.03-Analyze, examine, create and debug a program that includes sequencing, repetition, conditionals and variables in a programming language.

6-8.CS.HS.01 – Design projects that combine hardware and software to collect and exchange data.

6-8.AP.V.01- Create clearly named variables to store and manipulate information.

6-8.AP.C.01- Design and develop combinations of control structures, nested loops and compound conditionals.

Objectives:

- Create a project to control a cutebot using another micro:bit.
- Implement the concept of motion, roll and pitch.

Background information / Activation of Prior Knowledge:

- Access the MakeCode classroom.
- Basic computational thinking skills.
- Basic knowledge of block programming.
- Knowledge of roll and pitch.

Introduction / Anticipatory Set:

Download the roll controller program or make them yourself by following the **"Guided Practice (We do)"** section. Look over the programs and develop familiarity with them.

ENTRANCE TICKET (pre-writing):

Teaching (I do):

Introduce what the goal of this project is controlling cutebot using micro:bit and understand the concept of roll and pitch. Then depending on students skill level/demeanor there are options:

- Assign students **"Writing Integration 1"** from the **"Group / Independent Practice (You do)"** section. After they have completed that assignment, you can quickly guide them through the **"Guided Practice (We do)"** section to ensure everyone is on the same page.

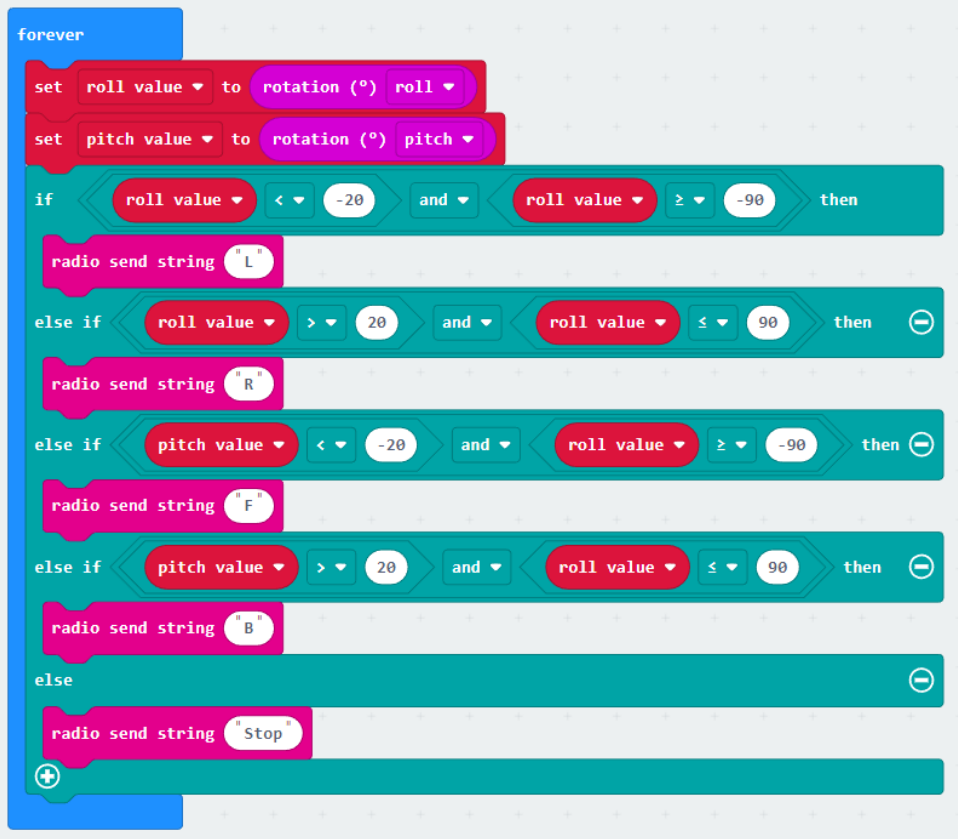
Guided Practice (We do):

1. **Plan and design cutebot programs.**
    a. **Controller/Sender micro:bit:**
        i. On start show icon
        ii. Set radio set group 1

        iii. Insert forever loop.
        iv. Create two variables: roll value and pitch value**.**
        v. Inside the loop set roll value to rotation roll.
        vi. Inside the loop set pitch value to rotation pitch .
        vii. If roll value is between -90 to -20 then send command to turn left by else if roll value<-20 and roll value>=-90 then radio send string L
        viii. If roll value is between 20 to 90 then send command to turn right by else if roll value>20 and roll value<=90 then radio send string R
        ix. If pitch value is between -90 to -20 then send command to move forward by else if pitch value<-20 and pitch value>=-90 then radio send string F

x. If pitch value is between 20 to 90 then send command to move backward/reverse by else if pitch value>20 and pitch value<=90 then radio send string B

xi. Otherwise, if roll and pitch value is between -20 to 20 means the controller is in a stable position so send command to stop the cutebot by else then radio send string Stop



b. Receiver micro:bit/cutebot:

i. On start show icon 😊

ii. Set radio set group 1

iii. Set left wheel speed 50% right wheel speed 50%

iv. Execute command according to received string from controller micro:bit. To receive string use on radio received receivedString

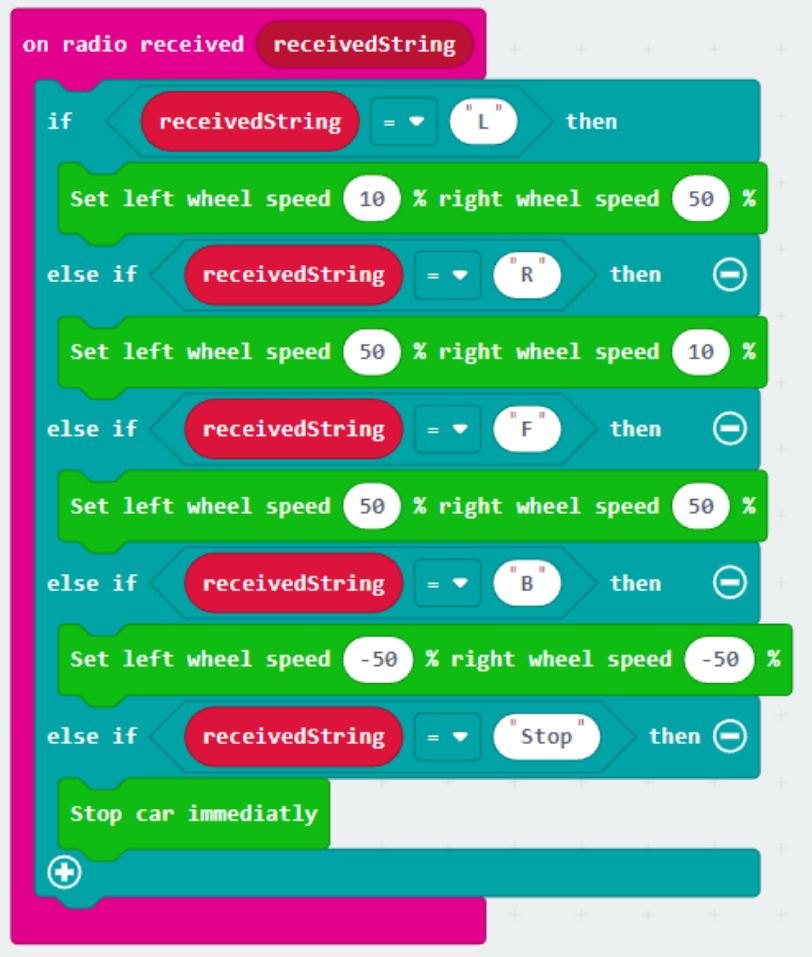
v. If receivedString=L turn left by Set left wheel speed 10% right wheel speed 50%

vi. Else if receivedString=R turn right by Set left wheel speed 50% right wheel speed 10%

vii. Else if receivedString=F move forward by Set left wheel speed 50% right wheel speed 50%

viii. Else if receivedString=B move backward by Set left wheel speed -50% right wheel speed -50%

ix. Else if receivedString=Stop stop car by stop car immediately



2. **Design experiments to analyze the motion of cutebot.**
    a. Here you can prompt students to come up with their own experiments or provide them with the ones below:

i. The first experiment would be to let the students try some other range for pitch and roll and analyze how it moves and turns.

ii. The next experiment would be to change the speed of wheel into other values and analyze its motion.

iii. The next experiment would be to control the speed of cutebot based on range of pitch and roll.

3. Perform the experiments designed in section #2.

I. Controller/Sender micro:bit

a. Plug micro:bit into computer and connect to MakeCode.

b. Download the sender code into micro:bit.

c. Connect with battery pack.

II. Receiver micro:bit/Cutebot

a. Plug micro:bit into computer and connect MakeCode.

b. Download the receiver/cutebot code into micro:bit.

c. Insert the micro:bit into cutebot.

d. Place the cutebot onto the floor and turn on the power button.

e. Analyze the motion of the cutebot.

4. Analyze the motion and rotation.

a. How the motion and rotation are analyzed can be customized to class curriculum. For example, when the cutebot turns left or right can be noted on variety of speed of wheel or on variety of range of pitch and roll.

b. Discuss the movement of cutebot as a class; comparing and contrasting the observations collected from the experiments. Or assign **"Writing Integration 3"** from the **"Group / Independent Practice (You do)"** section.

Group / Independent Practice (You do):

Writing Integration 1:

Prompt paired students to write out instructions to program the functionality of the sender and receiver programs (each student does one). Then have the student who wrote the instructions for the sender program read the instructions the other student wrote for the receiver program and proceed to code the receiver program based only on the written instructions. Then the students swap places, and the receiver program instruction writer reads the instructions for the sender program and programs it.

Writing Integration 2:

Prompt students to change the range of roll and pitch and analyze the change in motion of the cutebot. Provide reasoning for this deviation i.e. if the range of roll and pitch is changed how it impacts on motion.

Writing Integration 3:

Write about the results collected from each experiment. Ask students to compare and contract the results between experiments. Were the results what they expected they would be? If not, then what might be happening to the results that changed them so?

Notes, Reflections, Attachments