TiltBit: Description

Your task is to design a Makecode software library that lets you use the micro:bit as a virtual joystick that will drive a vehicle. You will need to simulate the signals the controller will send to the vehicle and show that they are working as intended.

The micro:bit joystick should produce the following outputs:

A y-axis which has values that range from -100 (fully tilting the micro:bit toward the user) to +100 (fully tilting the micro:bit away from the user)

An x-axis which has values that range from -100 (fully tilting the micro:bit to the left) to +100 (fully tilting the micro:bit to the right

Pressing the B button should set a hornIsOn variable to 1 while the button is pressed, and 0 when it is not pressed.

Pressing the A button should set a boostModeEnabled variable to 1 if it is equal to 0, and 0 if it is equal to 1.

A serial string variable called outputString that has a string value of the format "yValue,xValue,hornIsOn,boostModeEnabled" all separated by commas.

A sample outputString value would be "75,-25,1,0" . This corresponds to the following:

- xAxis = 75
- yAxis = -25
- hornIsOn = 1
- boostModeEnabled = 0

You should use an LCD to show the output values in the string are working as expected as you move the micro:bit around.

As you think about the functions you will use here, follow these guidelines:

- Use one function to read a sensor and then another process its value
- Use constants wherever possible to define important values for your program
- The Math.map() function will be your friend here.

TiltCar - Description

Your task is to design a Makecode software library to read the output string from a TiltBit micro:bit program and use them to drive a four-wheel-drive rover. Here's the idea:

The TiltController (when correctly implemented) outputs a string that looks like "75,-25,1,0" . This corresponds to the following:
xAxis = 75

yAxis = -25

hornIsOn = 1

boostModeEnabled = 0

You will receive this string over the radio. You will need to use the provided function to get the values out of the received string.

Starter code is here: https://makecode.microbit.org/_9F7LMW26P1jJ

You will then need to process them to apply an arcade drive system to control the rover.

Arcade drive for a robot means that the x and y values of the joystick are blended together to give a signal for the left and right motors. Here are some example values:

An input of (0,0) should stop the rover.

An input of (50,50) means the rover moves forward in a straight line.

An input of (100,50) will result in a left forward turn.

(-50,-100) means a backward turn to the left.

The page linked here has a good explanation of this.

You need to write a micro:bit program that does the following:
Reads the latest message received over the radio to get the TiltControlBit string.

Process this string using the provided code into variable values.

Implement an arcade drive function that maps the x and y values from the joystick to leftMotor and rightMotor. This should be done using two separate functions:

If the boostModeEnabled value is 1, the rover should run with maximum power.

If the boostModeEnabled value is 0, the rover should run with 50% maximum power.

For both motors you will need to use the servoWritePulse command within a function to output the correct value. A servoWritePin command of 2000 is full-speed in one direction, 1000 full speed in the other direction, and a value of 1500 representing a stopped motor.

Write a function that sets a digital output pin on the micro:bit to 1 when the hornIsOn variable is equal to 1. This will ultimately be used to control a light or a buzzer.

You should use an LCD to show that the output values are working as expected.

As you think about the functions you will use here, follow these guidelines:
Use one function to process the values from the radio and then another to process its value for the outputs. This means an arcadeDrive function that takes in the joystick values and outputs power values for the left and right sides. A separate function should then process this value for sending a servo pulse to the motor.

Use constants wherever possible to define important values for your program

The Math.map() function will be your friend here.