Debugging in Robotics

General Advice:

1. Baby steps. Program and test small actions at a time. It’s much easier to debug your work when it’s fresh in your head.
2. Don’t make mistakes ☺

# Strategies

## Run the program while holding the robot in the air

Helps you:

1. Detect if there’s a mechanical issue preventing some motion.
2. See when certain parts of your program run when they didn’t on the mat.

e.g. You’re raising or lowering your arm before a moving forward, and it’s not going forward. If you hold the robot in the air and the arm raises, then the wheels move there’s likely a problem with the arm not being able to complete its motion.

e.g. Your robot isn’t turning because you have an axle jammed into a beam, you’d see this when you’re watching the wheels try to move.

## Add a sound

Helps you know you’ve reached a certain point in your program.

e.g. You expect your robot to turn after going forward, but it’s not turning. Put a sound in before the turn, after the move, so you know you completed the move.



## Write to the display

Helps you see what’s going on in your program.

e.g. You’re trying to debug a gyro, output the gyro degrees to the screen.

e.g. You’re trying to use math blocks to calculate rotations when you input inches, output the result of the calculation to the screen.



## Add a pause

Helps you stop the robot at a certain point in its mission to help you see if it’s getting ready to do the next thing right.

e.g. You’ve moved, turned, moved and getting ready to turn or lift or place something and you want to make sure you’re lined up. Adding a pause until you press a button gives you time to figure out what’s going on.



## Run it on another robot

If you’re lucky enough to have a second robot built identically as a backup, run your program on this robot to see if it’s a problem with the original robot.

e.g. You had to rebuild something but when you put the robot back together you swapped a wire and now one of the motors isn’t working. If the backup robot didn’t have their wires swapped you would at least know the program is fine and the problem is with the hardware.

## Slow down

This helps for a bunch of reasons:

1. Gives the robot time to react to certain events (e.g. gyro degrees).
2. Prevents drastic stops and shuttering of the robot as it transitions between steps of the program.
3. Often more accurate.
4. Like adding a pause, sort of, in that it lets you observe your robot more easily.

If you had a standard move speed as a constant, you could change it in one place and ripple everywhere. Or change it on individual move blocks. This pictures shows using move speed as a constant.



## Delete and re-download

Sort of a last resort, but if you’re convinced the program should be doing something and everything else you’ve tried isn’t working, maybe you didn’t download the version of the program you thought you did.