

## Engineering with EV3 Workshop

### Baker's Dozen Car / Half Dozen Car

Simplicity can be a wonderful thing.

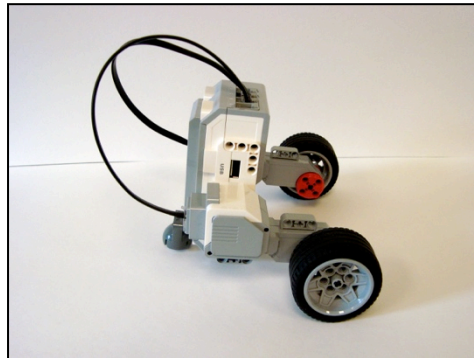
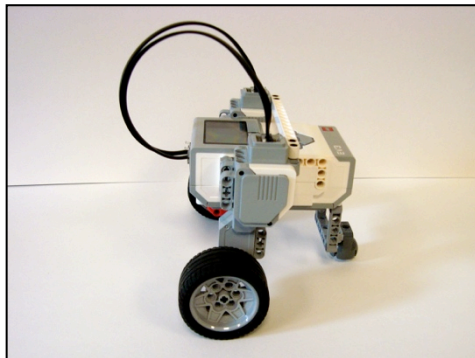
Baker's Dozen Car:

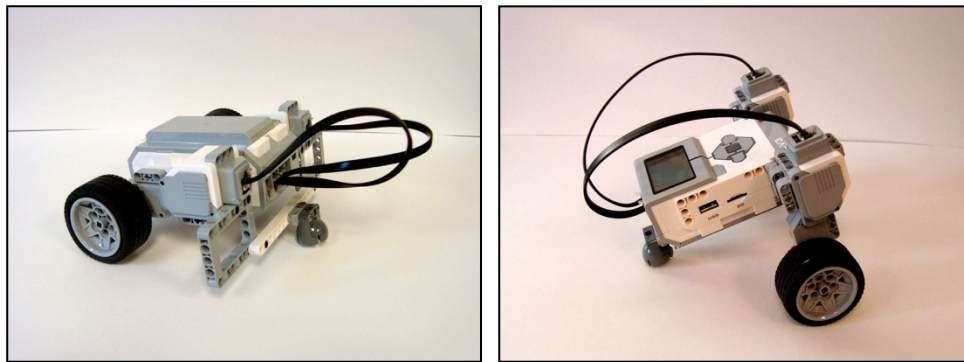
Build a simple EV3 car using any or all of the pieces shown below—plus no more than thirteen additional pieces.



Your car must be able to go forward, go backward, spin right, and spin left—all without falling apart. There are many possible solutions to this challenge—there is no one “correct” answer. Feel free to tinker.

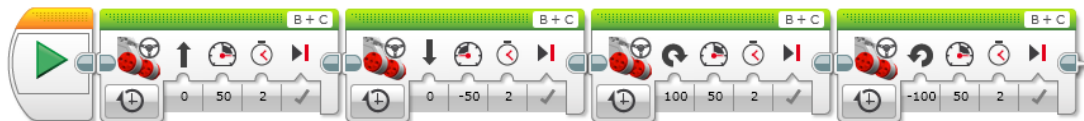
To help get you started, here are some possible solutions. Each car shown uses no more than thirteen additional parts.





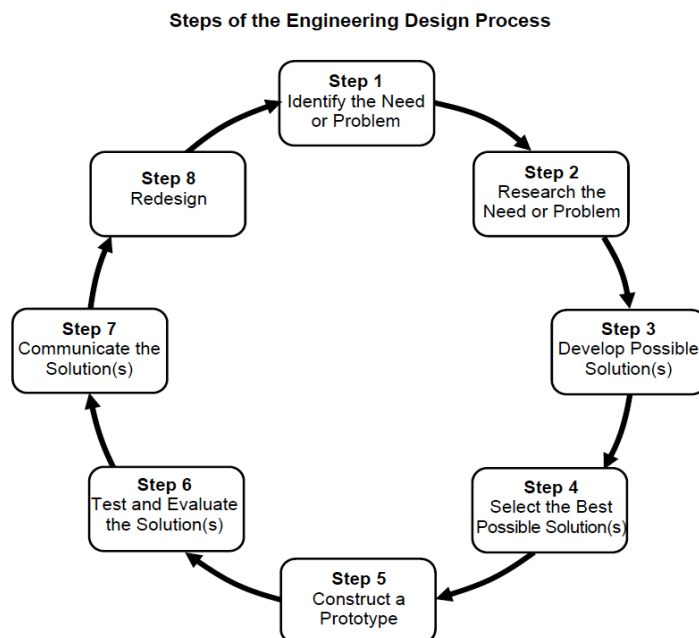
For an extra challenge, try a Half Dozen Car:  
 Build a simple EV3 car using any or all of the pieces shown above—plus no more than six additional pieces.

Test program:  
 In order to test your car, write an EV3 program that runs the car forward, backward, spin right, and spin left, each for two seconds.

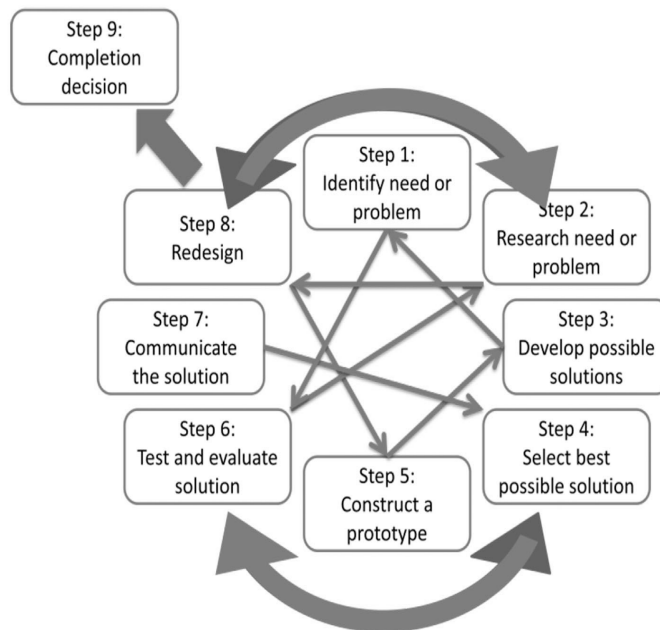


### The Engineering Design Process

The design process in theory--Massachusetts STE Frameworks, 2012



In practice, the design process often looks more like this...



In approaching a design problem, engineers must consider a number of things including their client's needs and wishes, the design criteria (requirements) and constraints (limitations), and how to conduct meaningful tests of their solution.

### Drawing the Line

Design a penholder for your car, one that will draw a clearly visible line as the car drives. You may use any combination of pieces from your EV3 kit, but no non-LEGO materials (except the pen, of course). Test your penholder using the forward-back-spin-spin test program.

What are the criteria and constraints for this problem?

Is the suggested test a meaningful one?

### Project Rules

*We will be doing projects, both large and small, throughout the year. Sometimes your attempt will be a success. Other times, it will be a miserable failure. Either result is okay—failure is an integral part of the design process.*

*What is not okay: laughing at someone else's project.*

*So, some rules for working in the class:*

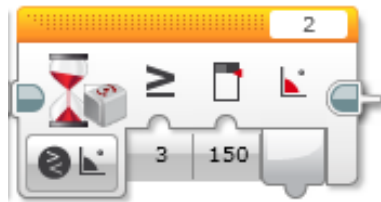
- 1. You may not criticize or make fun of another student's work. This includes laughing, teasing, or comparisons ("My car is so much better than your car....").*

2. *However, you may provide CONSTRUCTIVE criticism. To be constructive, the comment must be specific and offer a possible solution. For example, “Hey your car doesn’t go straight” is not acceptable, but “I noticed that your car veers to the left. It looks like the back wheel is rubbing against the frame” is welcomed.*
3. *No designs are private property. Anyone may get ideas from any other design. If someone copies a piece of your design, the proper reaction is to be flattered—clearly, the other person has recognized your brilliance.*
4. *If you get stuck, feel free to look at other people’s designs to see how they have solved similar problems.*
5. *And finally, relax! Things will go wrong—but you will have plenty of time and assistance to fix the problems.*

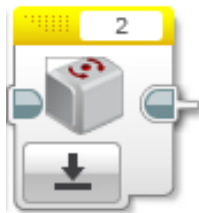
## Spirographer

Spirograph is a children’s toy that uses gears and rings with teeth to produce geometric designs. Use an EV3 robot with a gyro sensor to make your own spirographic designs.

To draw a design, your robot needs to drive forward, turn using the gyro sensor, and then repeat these two steps over and over. To execute the turn, use the Wait for Gyro block, under Flow, setting it to Compare, then Angle.

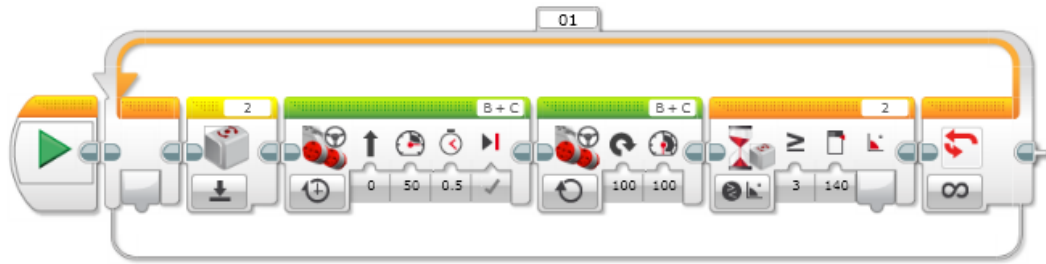


Since your program will call the gyro sensor repeatedly, you will need to initialize the sensor before it is used each time. To do this, go to Sensor, then Gyro Sensor. Choose Reset Gyro.

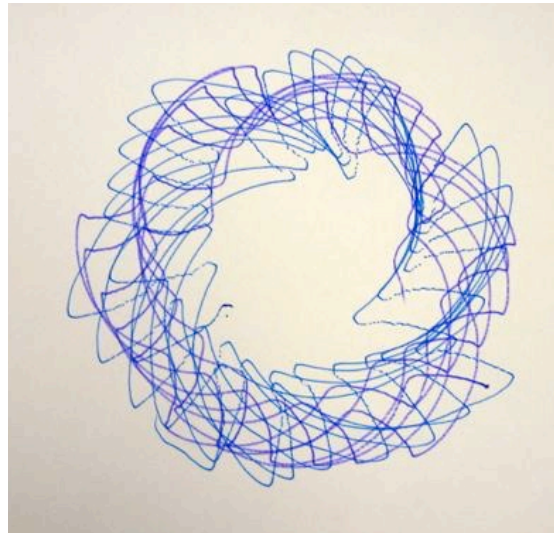
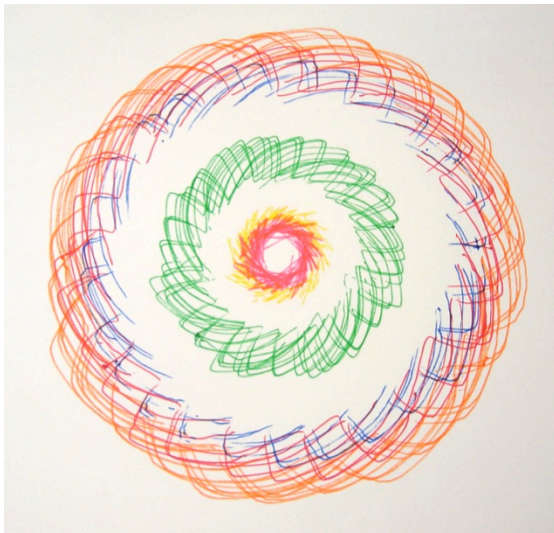


Once your basic spirographer is working, try modifying some of the parameters. What happens if you change the angle of the turn? The sharpness of the turn? The radius of the circle? The placement of the pens?

### Sample Program for Spirographer



### Sample Designs for Spirographer



### Gotta Dash

Create a robot that draws a dashed line. Your solution can involve building, programming, or both.