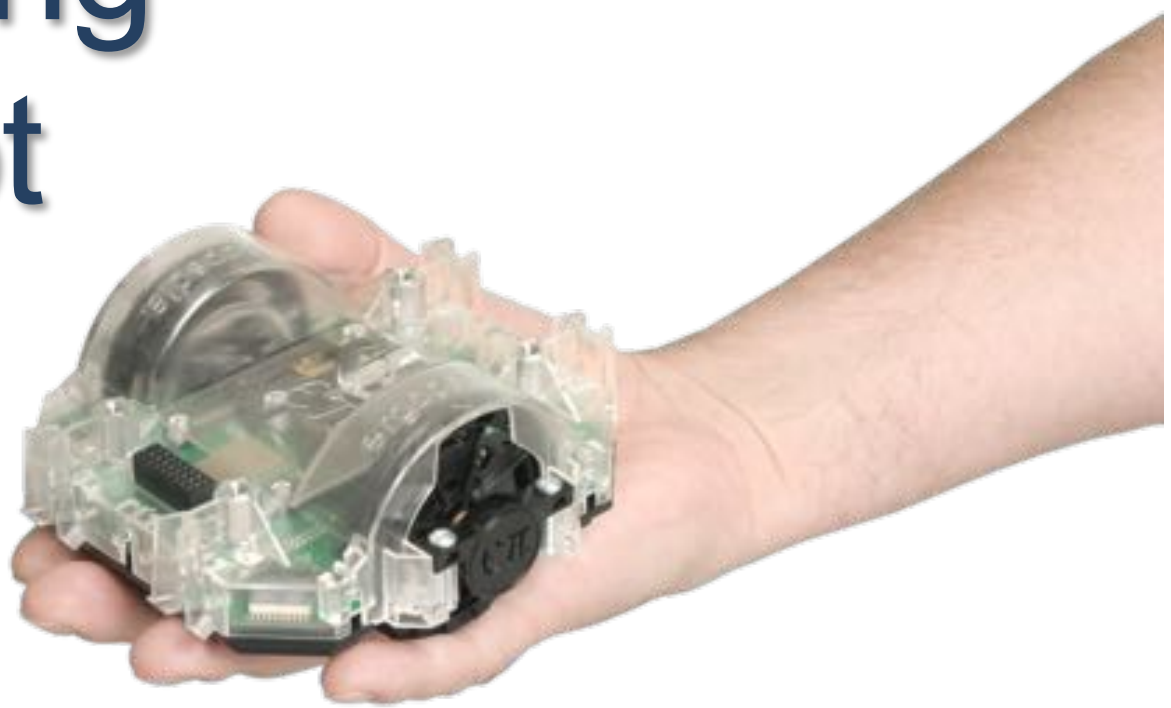
Programming the ErgoBot



Objectives

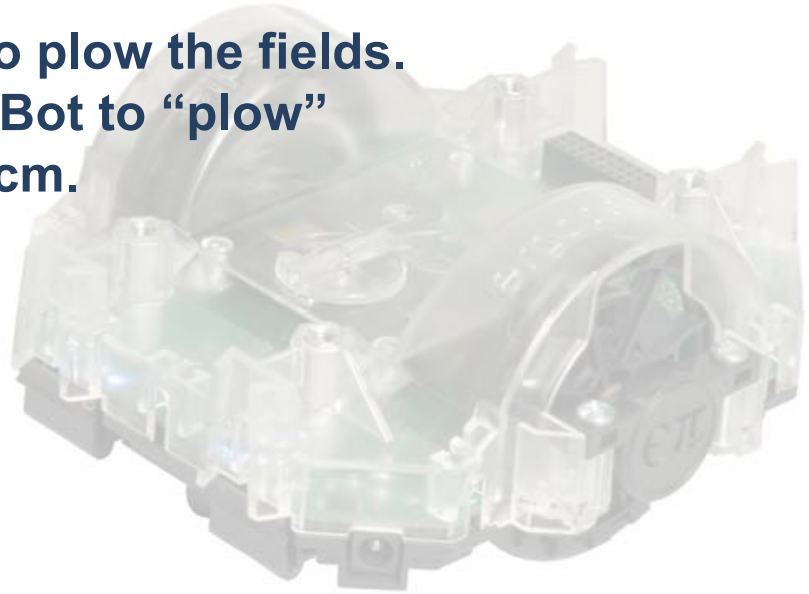
- **Demonstrate familiarity with several basic terms and elements of programming.**
- **Create and execute programs to operate a robot (the ErgoBot).**

Assessment

1. What are comment lines and why are they important in computer programs?

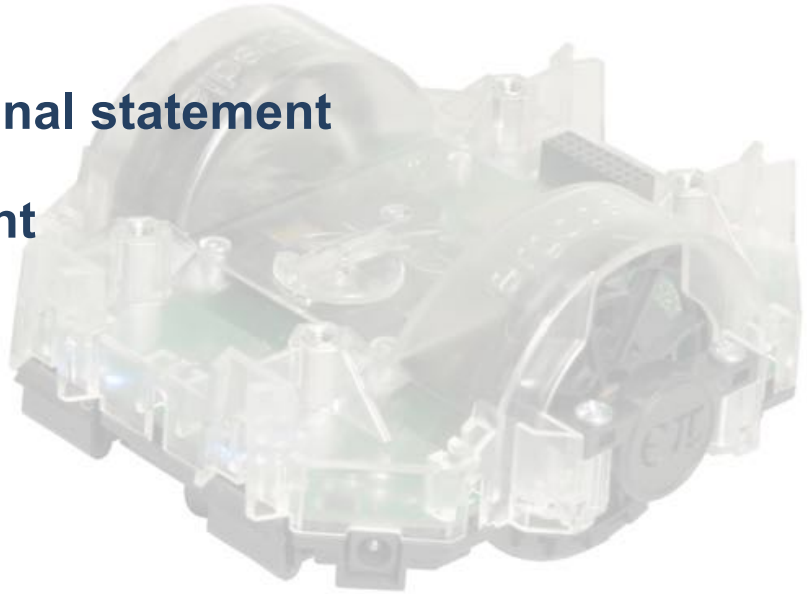
2. Modern farmers program their tractors to plow the fields. Write a program that will cause the ErgoBot to "plow" the edge of a "field" that is 50 cm by 50 cm.

   Use as few commands as possible.



New terms

- robot
- program
- execute
- initialization routine
- loop
- comment
- memory
- syntax
- conditional statement
- argument
- debug



What is a robot?

What is a robot? What makes it different from any other machine, such as a hammer, or a bicycle?

And how do robots differ from one another?

What is a robot?

A robot may not look like a human or an animal.
The ErgoBot is a robot that looks like a car.
Generally speaking, robots are “smart” machines.

What is a robot?

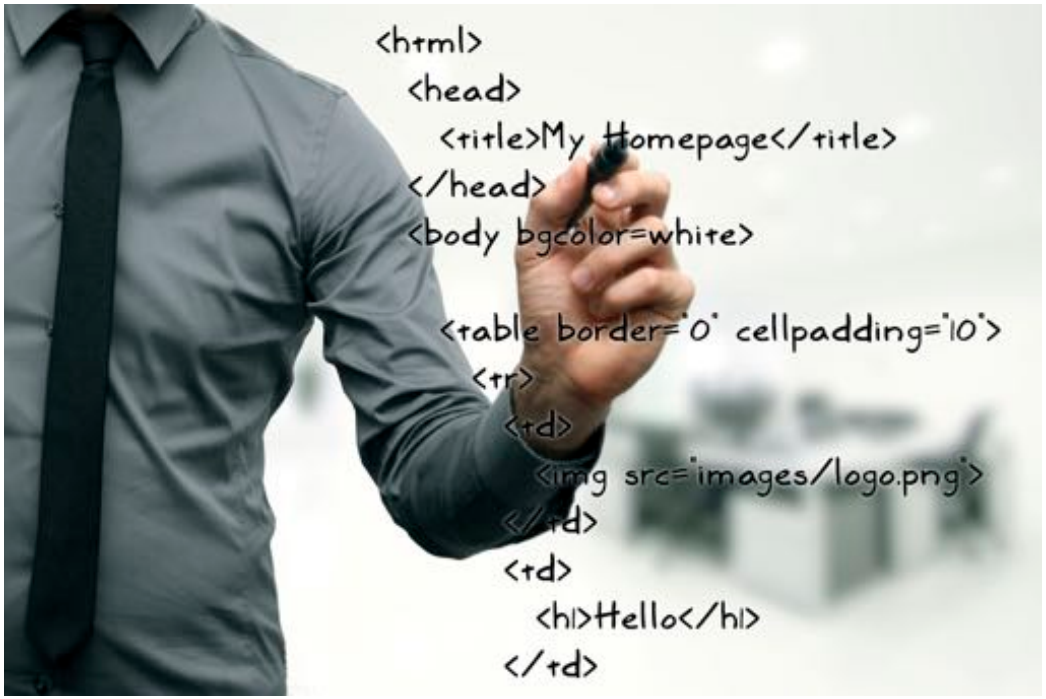
A robot may not look like a human or an animal. The ErgoBot is a robot that looks like a car. Generally speaking, robots are "smart" machines.

•Robots are electro-mechanical devices that contain microprocessors (computer chips)—and can be programmed for a variety of behaviors.

- **Many robots are able to receive, collect, and/or transmit data and information.**

What is a program?

A computer program is a list of commands written in a specific programming language to achieve a desired task.

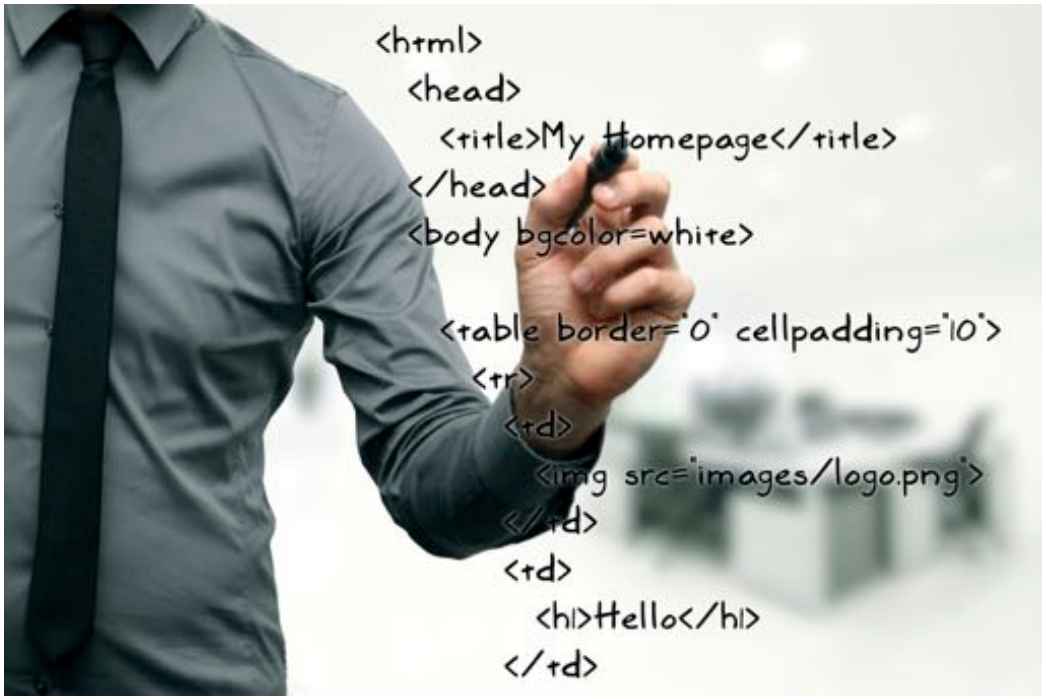


What is a program?

A computer program is a list of commands written in a specific programming language to achieve a desired task.

There are many different programming languages, but they all have certain features in common.

You will program the ErgoBot using a simple set of commands in its own language.



ErgoBot commands: start

Every ErgoBot program begins with the start command.

The start command executes an *initialization routine*: the ErgoBot "wakes up", beeps to signal that it is ready, and then executes the commands that follow start.

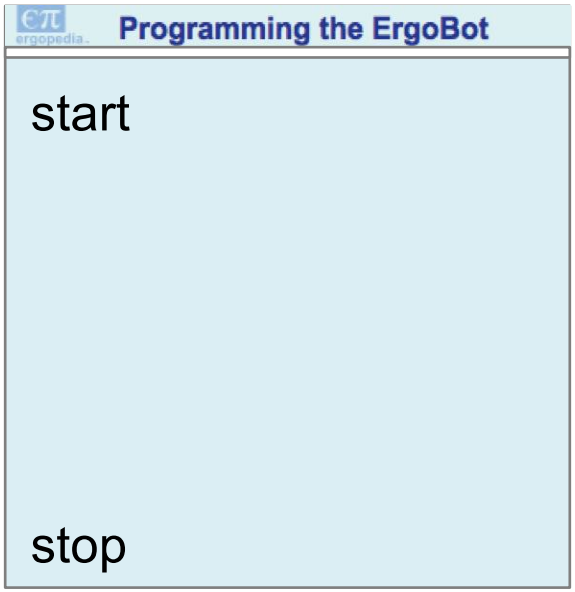
Commands appearing before the start are not executed.

ErgoBot commands: stop

Every ErgoBot program ends with the stop command.

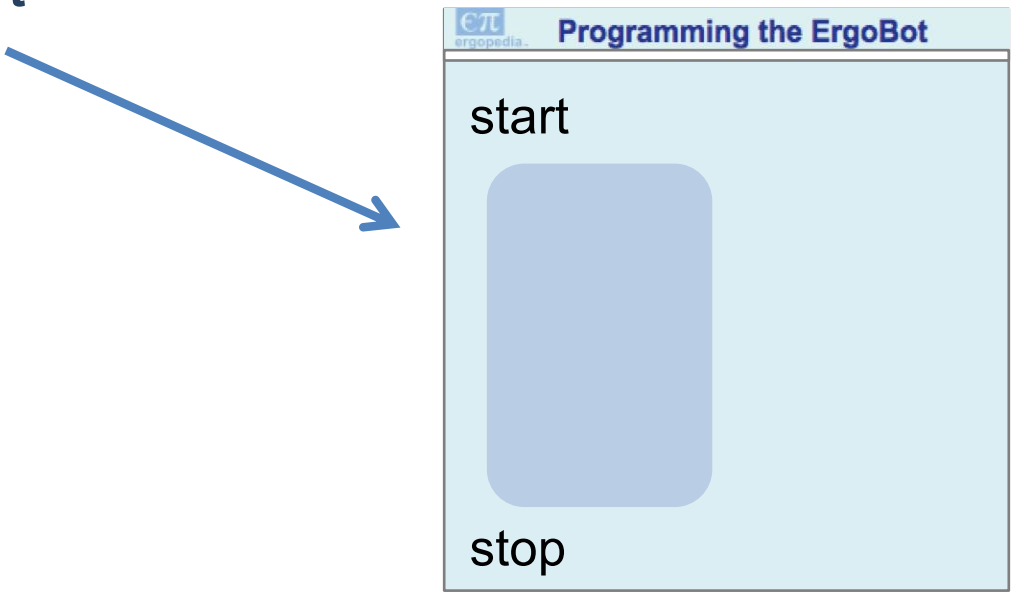
The stop command causes the ErgoBot to beep, signaling that the program is finished.

Commands after the stop command are not executed. This can be helpful for testing your code in sections.



The main body

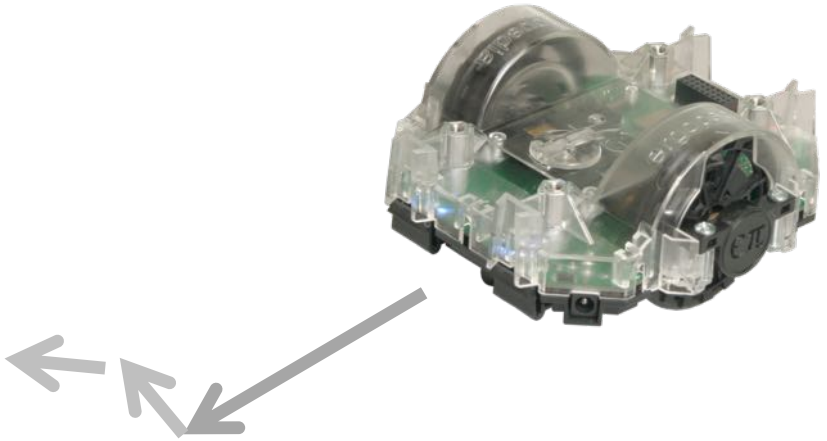
The *main body* of the program is written here, between the start and stop commands.



Navigational commands

You will use these commands to program the ErgoBot's movements.

drive (0.5)	Drive forward 0.5 meters.
right (90)	Turn right 90°.
left (60)	Turn left 60°.



Arguments

The numbers inside the commands are called *arguments*.
You can change these arguments to other values you need.

drive (-0.5)	Drive backwards 0.5 meters.
right (45)	Turn right 45˚ .
left (70)	Turn left 70˚ .
	The maximum turn angle is 360˚ .

Syntax

Programs must use proper *syntax*.

Commands must be written and arranged according to the rules of the programming language, or the program won't *execute*. Find the syntax errors in this sample:

```
Start

forward (2)

right (180)

left (500)
```

Syntax

Programs must use proper *syntax*.

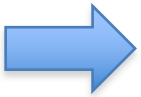
Commands must be written and arranged according to the rules of the programming language, or the program won't *execute*. Find the syntax errors in this sample:

~~**Start**~~

~~**forward**~~ **(2)**

right (180)

left ~~(500)~~



start

drive (2)

right (180)

left (360)

left (140)

Debugging **is the process of fixing any errors of syntax or logic in a program so that it runs correctly—an essential programming skill.**

Investigation

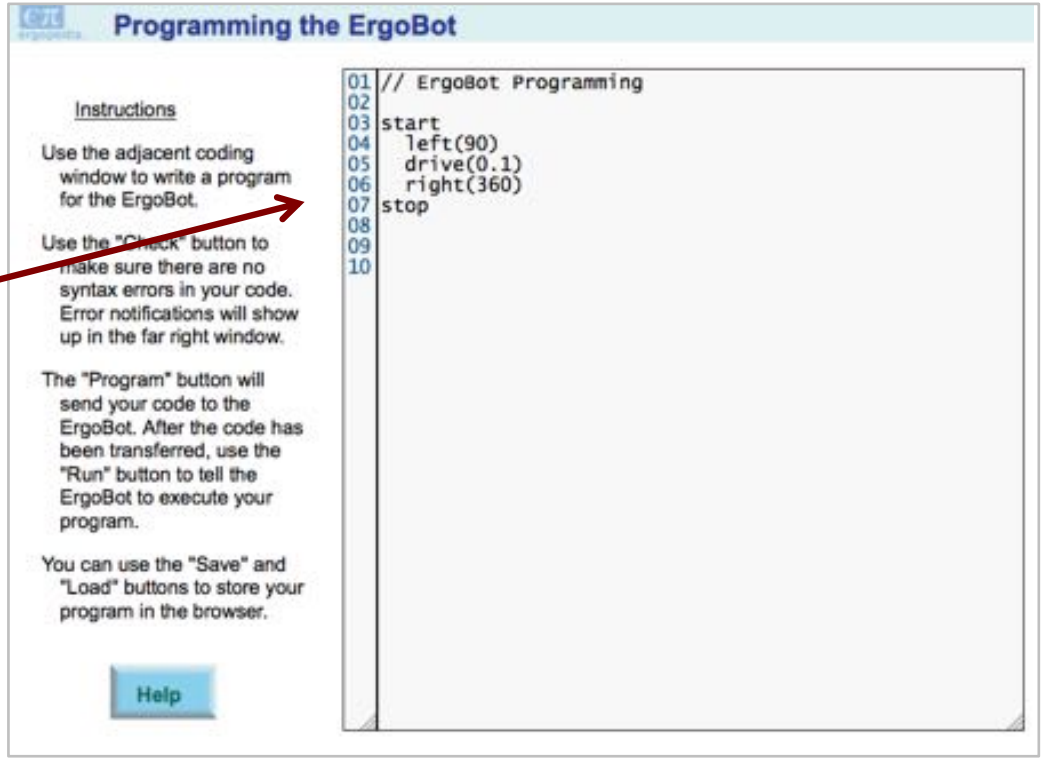
Click open the interactive simulation in your electronic resources:

Programming the ErgoBot.

Investigation

Part 1: Clear the road

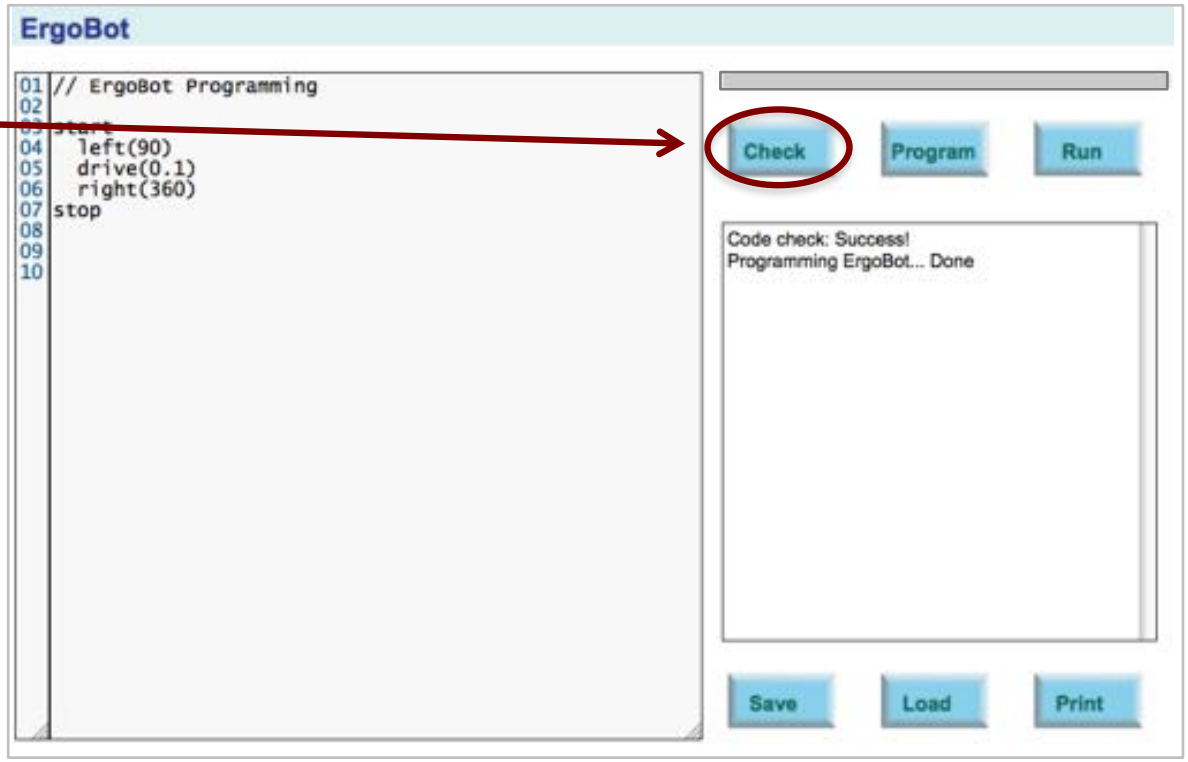
1. Pair the ErgoBot to your computer or tablet and set it to drive mode.
2. Type commands into the programming window.



Investigation

Part 1: Clear the road

3. [Check] finds any errors in the program and reports them to the Output window.



Investigation

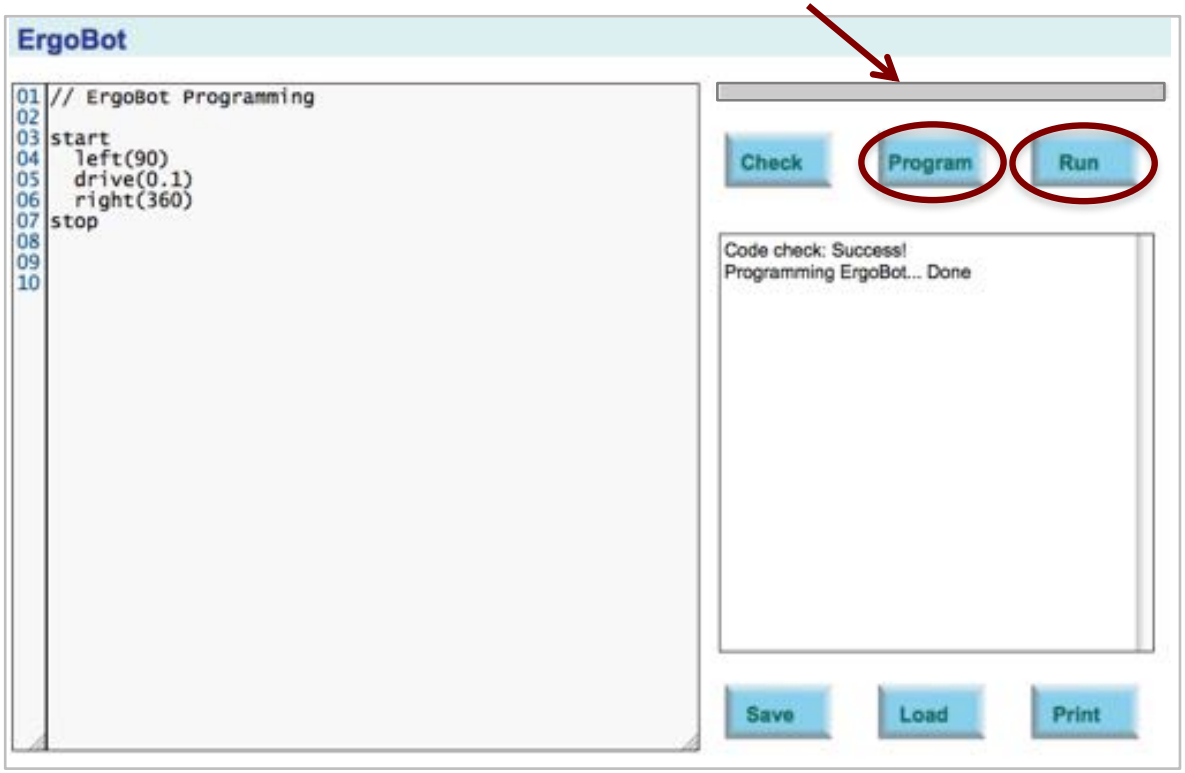
Part 1: Clear the road

3. **[Check] finds any errors in the program and reports them to the Output window.**

4. **[Program] uploads the program to the ErgoBot.**

5. **[Run] executes the program.**

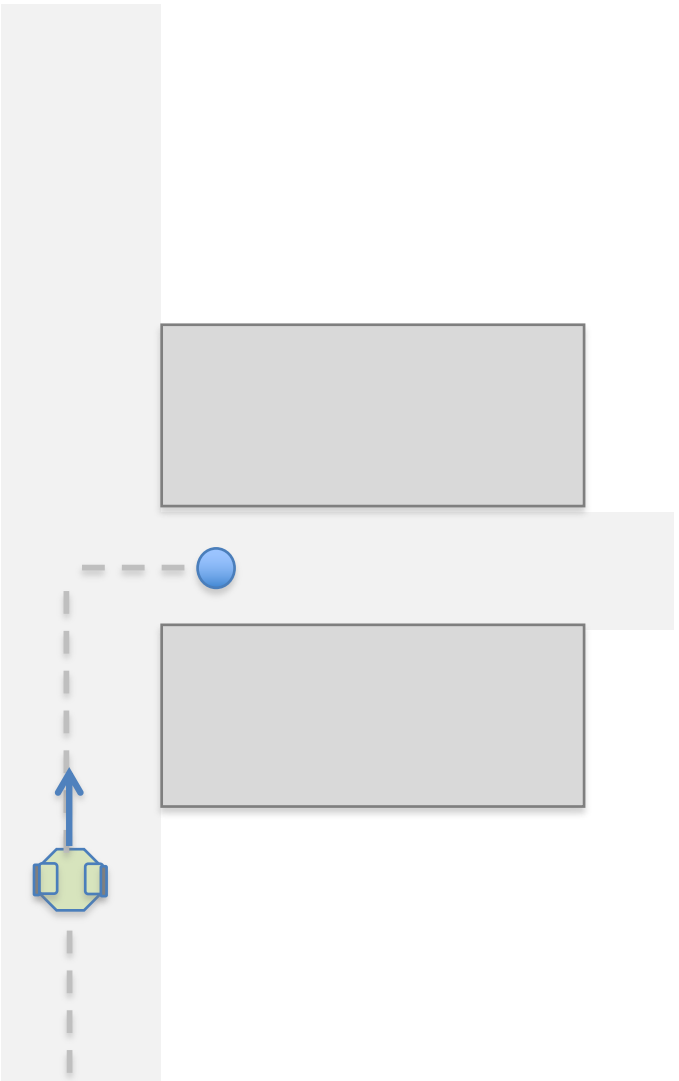
The progress bar indicates when the download is complete.



Investigation

Part 1: Clear the road

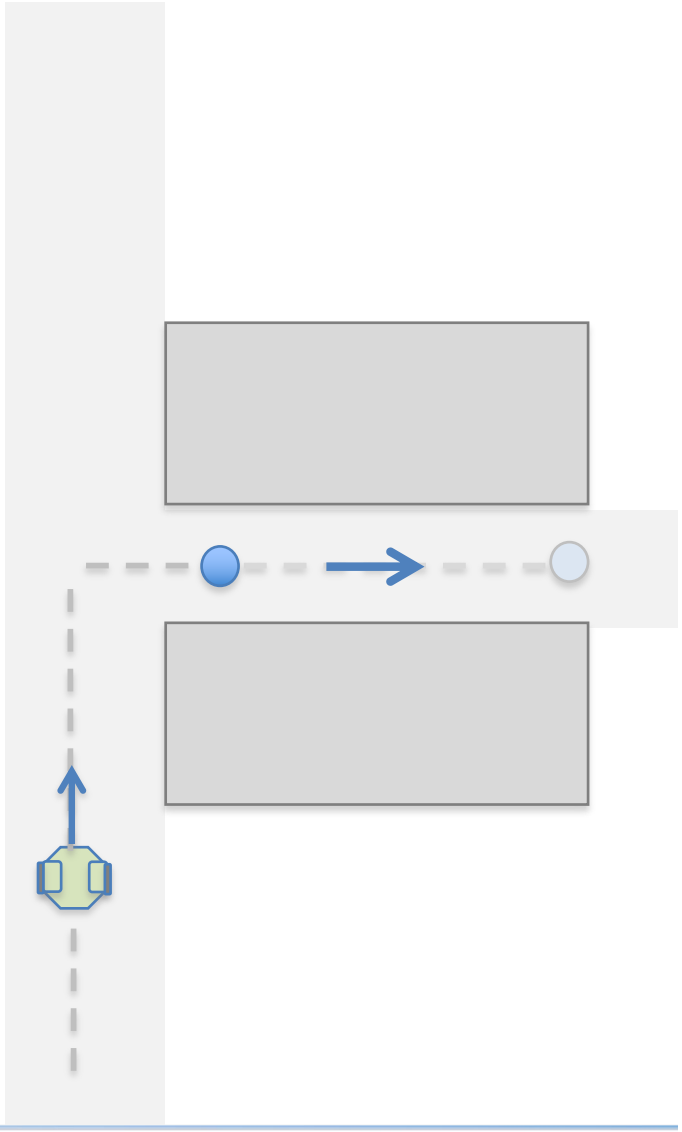
6. **Program the ErgoBot to complete this task:**
   - **Drive down a straight "road".**
   - **Turn into a "side street".**



Investigation

Part 1: Clear the road

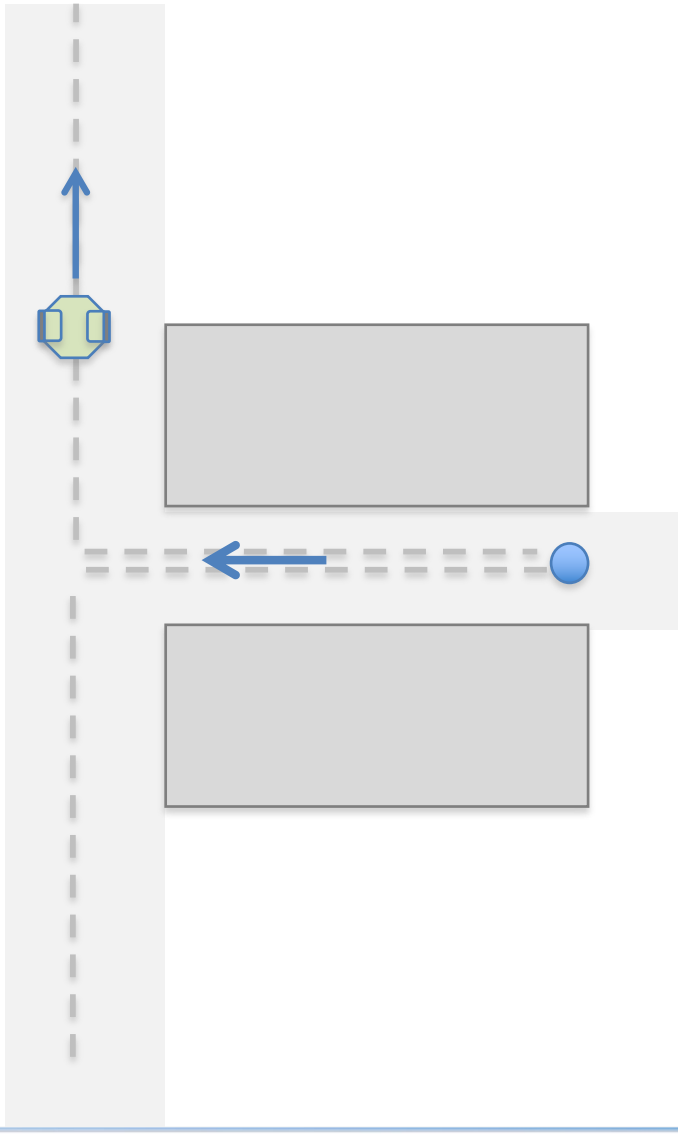
6. **Program the ErgoBot to complete this task:**
   - **Drive down a straight "road".**
   - **Turn into a "side street".**
   - **Plow an obstacle (a paper cup) to the end of the side street.**



Investigation

Part 1: Clear the road

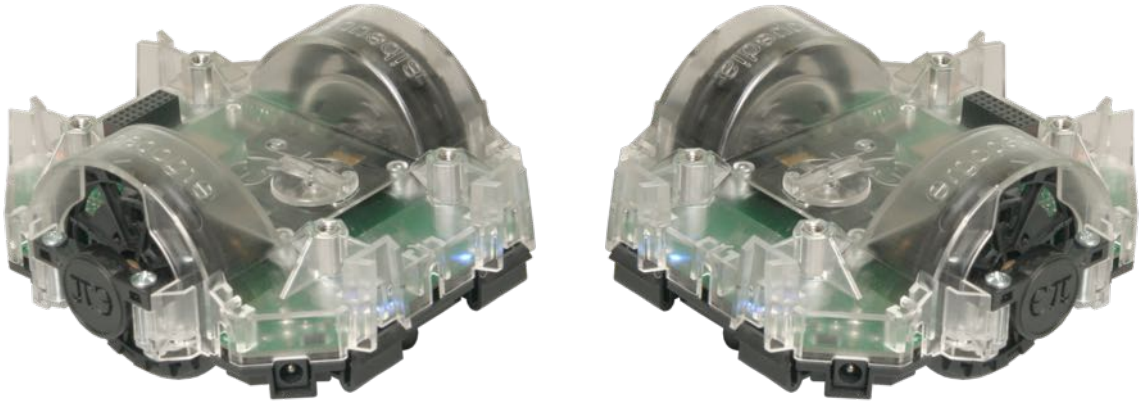
6. **Program the ErgoBot to complete this task:**
   - **Drive down a straight "road".**
   - **Turn into a "side street".**
   - **Plow an obstacle (a paper cup) to the end of the side street.**
   - **Return to the main road and continue on.**



Investigation

Part 2: Join the dance

In the second part of this investigation, you will incorporate *comment lines* and *loops* in your program.



Comments

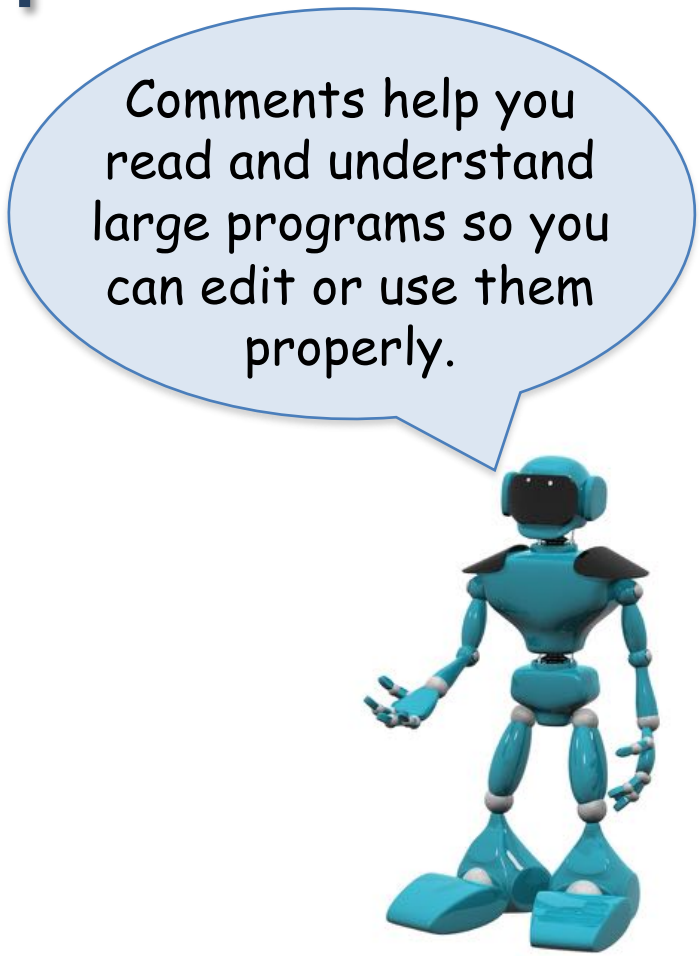
Good programmers provide internal documentation in the form of *comment lines*.

A comment is not a command. It is an unexecuted line of a program that explains what the program is doing.

```
//  This double slash tells you that this is a comment.
```

Comments: an example

```
// Title: First Program        Date: 12/10/20
start
    // Drive forward, spin around, and repeat.
    do
        drive (0.2)
        right (360)
    loop 2
stop
```



A programming tip

```
// Title:  First Program        Date: 12/10/20
start
    // Drive forward, spin around, and repeat.
    do
     // drive (0.2)
        right (360)
    loop 2
stop
```

Commenting is often used to temporarily switch off a command without erasing it. This can help you debug your program.

Loops

All programming languages allow you to repeat a set of commands. You can choose the number of times the ErgoBot repeats the commands inside a "do loop".

```
do
  command A
  command B
loop 4
```

Loops

All programming languages allow you to repeat a set of commands. You can choose the number of times the ErgoBot repeats the commands inside a "do loop".

```
do
  command A
  command B
loop 4
```

do: Execute the set of commands below, until you reach **loop**.

loop: Jump back to **do**. After 4 repeats, continue on with the rest of the program.

Test your knowledge

What does this program do? And what is missing?

```
start

    do

    drive (0.2)

    right (360)

    loop 3
```

Test your knowledge

What does this program do? And what is missing?

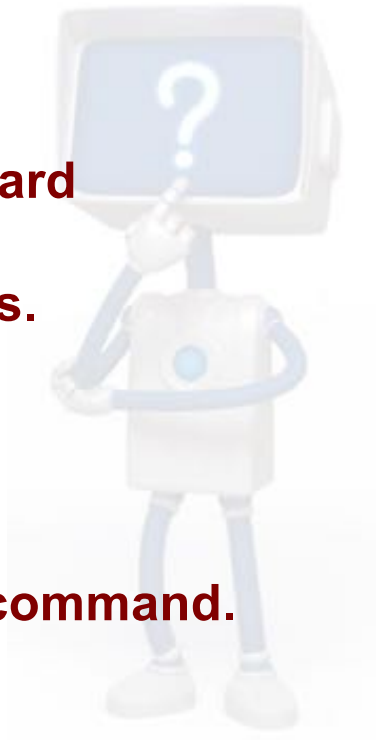
```
start

    do
    drive (0.2)
    right (360)
    loop 3

stop
```

The ErgoBot drives 20 cm forward and spins completely around. It repeats these motions 3 times.

The program is missing a stop command.



Memory

Inside the ErgoBot, your program is stored on a memory chip as a series of 0's and 1's, called bits.

Your ErgoBot has limited memory space for storing your program.

In Part 2 of the investigation, you will program your own motion routine—but don't make it too long, or you will exceed the memory space available.

Investigation

Part 2: Join the dance

1. **With a partner, program a brief motion routine for a pair of ErgoBots.**

2. **The routine must have two sections:**
   - **a section in which the ErgoBots execute the exact same moves; and**
   - **a section in which they perform opposite moves.**

Investigation

Part 2: Join the dance

3. **Include at least one loop.**
4. **Include a comment with the name and date of your program, and comment lines for each section of the dance.**
5. **Perform your routine for the rest of the class.**

The "dancers" may begin in any position.

Investigation

You can "comment out" an entire section of a program by surrounding it with these symbols: /* ~~entire section~~ */

```
start
    command xxx   // part 1 of dance
    command xxx
    command xxx
/*  command xxx   // part 2 of dance
    command xxx
    command xxx  */
stop
```

Use this method to test each section of your dance program separately.

In this example, Part 1 is being tested. Part 2 will not execute.

Controlling a robot

There are several methods for controlling robots:

- **Commands can be pre-programmed to produce specific behaviors.**

  You used this method in parts 1 and 2 of the investigation.

  Example: a robot used in automobile manufacturing may be programmed to repeat a specific task over and over.

Controlling a robot

There are several methods for controlling robots:

- **A robot may be actively controlled via a tether or a wireless device.**

  Example: a surgical robot may be controlled by the motions of the surgeon's arms and hands, allowing for increased precision.

  Doctors are putting this robot into position for surgery.

Photo credit: Department of Defense home page

Controlling a robot

There are several methods for controlling robots:

- **A robot may make decisions autonomously—based on information it receives from sensors. This can make a robot seem "alive".**

Example: a vacuum cleaner robot may use its sensors to avoid stairs and maneuver around obstacles.

Assessment

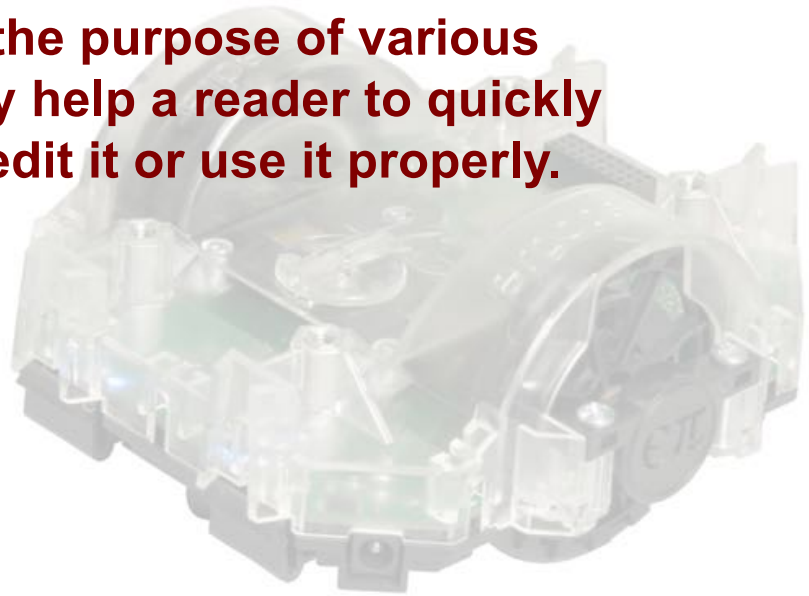
1. **What are comment lines and why are they important in computer programs?**

Assessment

1. **What are comment lines and why are they important in computer programs?**

   Comment lines are used for internal documentation.

   They describe the program and explain the purpose of various subsections of the computer code. They help a reader to quickly understand a large program in order to edit it or use it properly.



Assessment

2. **Modern farmers program their tractors to plow the fields. Write a program that will cause the ErgoBot to "plow" the edge of a "field" that is 50 cm by 50 cm. Use as few commands as possible.**

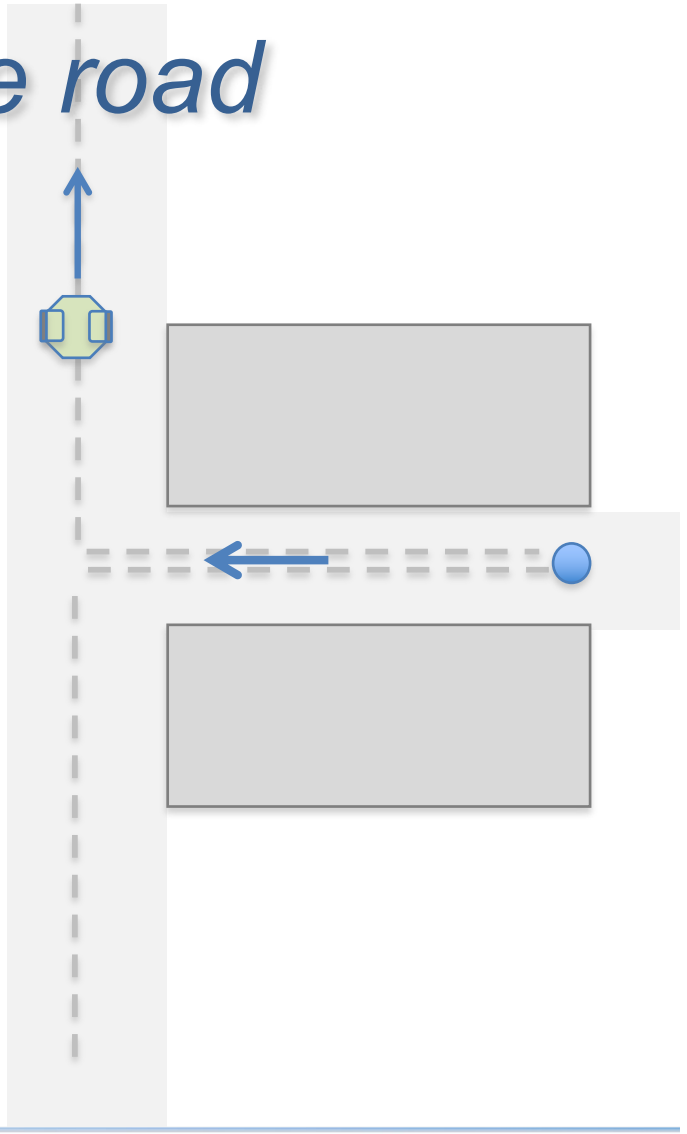
Assessment

2. Modern farmers program their tractors to plow the fields. Write a program that will cause the ErgoBot to "plow" the edge of a "field" that is 50 cm by 50 cm. Use as few commands as possible.

```
start
    do
      drive (0.5)
      right (90)
    loop 4
stop
```

Example program: *Clear the road*

```
start
  drive (1)
  left (90)
  drive (0.8)
  left (180)
  drive (0.8)
  left (90)
 drive (1)
stop
```



Example program: *Join the dance*

```
// Dance, ErgoBot #1:  4/10
start
do            // same moves
    left (30)
    right (30)
  loop 3
  drive (0.1)
  right (180)
  drive (0.1)
  left (180)

left (90)     // opposite moves
  drive (0.1)
  right(180)
  drive (0.1)
  left (90)
  drive (0.1)
  right (360)    // final spin
  stop
```

```
// Dance, ErgoBot #2:  4/10
start
do            // same moves
    left (30)
    right (30)
  loop 3
  drive (0.1)
  right (180)
  drive (0.1)
  left (180)

right (90)     // opposite moves
  drive (0.1)
  left (180)
  drive (0.1)
  right (90)
  drive (0.1)
  left (360)  // final spin
stop
```