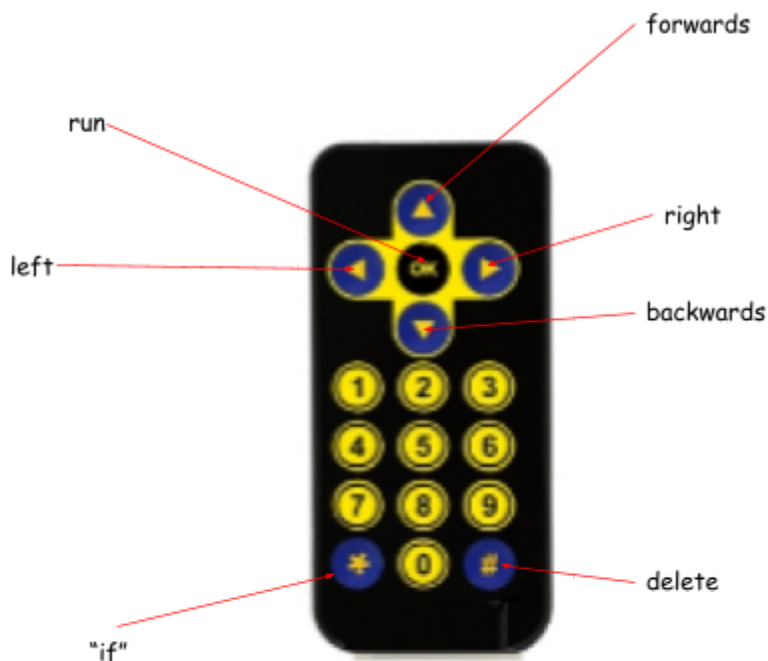


Programming Guide

Overview:

A “program” is a sentence of sorts, made up of letters and numbers. Once you tell the robot to start the program(called “running” the program), it will read the sentence and do whatever the sentence tells it to do. You write the program to do what you want the robot to do. To run this robot’s program, you will press the “OK” button on the remote after you use the remote to write the program.

In order to write the program, you press different buttons on the remote. The guide below tells you how to do this. A picture of the remote is shown below.



Simple Movements:

In order to move the robot, press one of the arrow buttons. For example, pressing the up-arrow moves the robot forwards; pressing the down-arrow moves it backwards; pressing the left-arrow makes the robot turn left 90 degrees, and pressing right makes the robot turn right 90 degrees. On the screen, forwards is represented by ‘F’; backwards by ‘B’, left by ‘L’, and right by ‘R’.

Examples:

F	Move forwards once
FL	Move forwards, then turn left 90 degrees
RBL	Turn right, move backwards, turn left
LF	Turn left, then move forwards

For-Loops:

A for-loop is where you tell the robot to do something a certain number of times. It is useful when you want it to do something a lot of times. For example, if you want it to move forwards 10 times, instead of pressing the up-arrow 10 times, you can tell the robot directly to move forwards 10 times.

To use for-loops, press the movement, and then use the digit buttons to input the number. For example, 'F10' will move forwards 10 times.

Examples:

L3	Turn left 3 times
F10	Move forwards 10 times
L3B	Turn left 3 times, then move backwards
F1	Move forwards 1 time
F0	Move forwards 0 times, i.e. just stay still

If-Statements:

Most robots are able to take sensor readings. That means that they can take in specific inputs from the environment such as temperature, sound, or distance. This robot in particular can see how much light there is directly below it; it does

this using something called a photoresistor. A photoresistor works like your eye, and can tell the robot how bright it is.

When robots are able to know about their surroundings, then they can react to those surroundings. They can do this using something called an “if-statement”. An if-statement tells the robot to do something only if the surroundings are a certain way. In this case, if the floor is very bright or very dark, then the robot will do what you tell it to. Whether or not it reacts to brightness or darkness depends on you.

To input an if-statement, first press the * button, which is represented by “I” on the display. Then press 0 or 1 — 0 represents darkness, 1 represents brightness. On the screen, an arrow will appear directly after the 0 or 1. Then put whatever you want the robot to do if the condition is true.

Examples:

I0->F	If the floor is dark, then move forwards
I1->F	If the floor is bright, then move fowards
I1->L3	If the floor is bright, then turn left 3 times

Debugging:

Sometimes you will make mistakes, and your code will not work as you expect it to. This is when you do debugging, which is when you find your mistakes. Often when doing debugging, we will be helped by a list of all possible errors that we could make. Below is a list of those errors for programming this robot.

I2->F	ERROR: 'I' must be followed by either 0 or 1	This particular robot can only understand two things — when the floor is bright(1), and when it is dark(0). Therefore, you can only have these two values.
I1->	ERROR: The '->' must be followed by something.	The robot must have something to do if the floor is bright or dark
I	ERROR: The 'I' must be followed by either 0 or 1	An if-statement tells the robot to move <i>if</i> something happens. The number after 'I' is that something.
3	ERROR: A program cannot begin with a number	A for-loop tells the robot to do something for how many times you tell it to. In this case, you've told how many times, but not what to do