

Lesson 6: Worksheet 6.1 - Flash the LED in response to a clap

In this activity, you need to write a program using Edison's clap-detecting sensor to get the robot to flash one LED light whenever it detects a clap.

The first thing to do is to plan out the program.





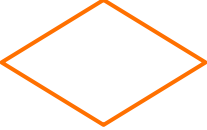
Flowcharts in programming

Professional programmers usually plan out their program before they start writing their code. Using flowchart diagrams is a way that programmers can organise and plan out their programs.

The idea of a flowchart is to graphically summarise what happens in the code without needing to go into all of the detail. Flowcharts allow a programmer to visualise and communicate how the 'flow' of the program will work.

In a flowchart, a program is represented using different shapes and arrows. Each shape represents a different element in the program, and the arrows show how the elements work together.

There are five main symbols used in flowcharts:

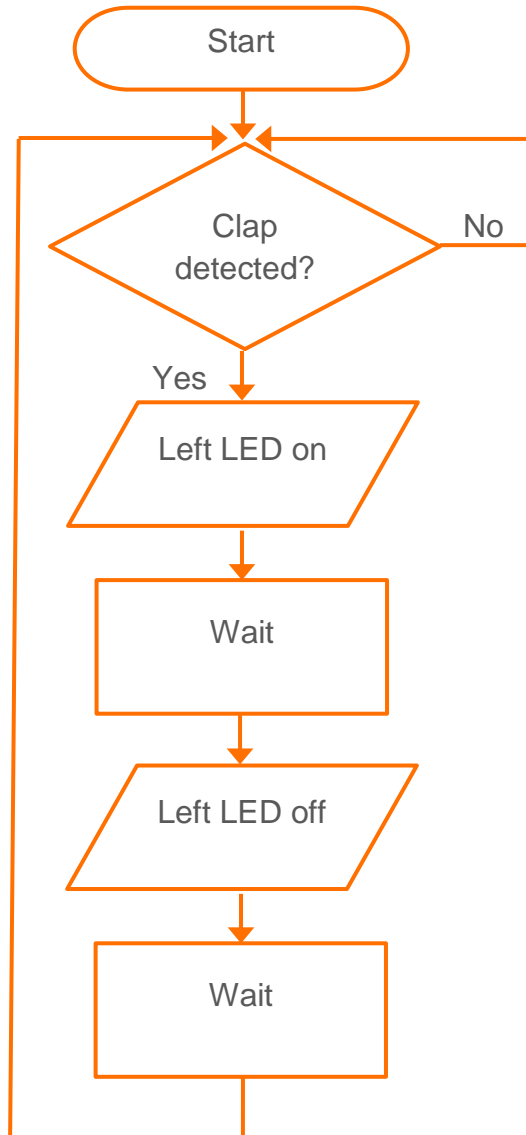
Symbol	Name	Function
	Terminator (start/end)	An oval represents a start or end point
	Arrow	A line which acts as a connector, showing the relationships between representative shapes
	Input/output	A parallelogram represents an input or an output
	Process	A rectangle represents a process or action
	Decision	A diamond represents a decision

More complicated flowcharts may also use additional shapes with different meanings.

When making a flowchart to plan a program, words are often added inside the shapes or next to the arrows. These words are short summaries of the process or decision.

Let's look at an example flowchart summarising the program we want to make.

Here is a flowchart for a program that will tell your Edison robot to wait for a clap, then flash the left LED:



This program will use Edison's sound-detecting sensor to determine whether or not a clap has occurred. The result determines how the code flows next.

When you look at this flowchart, you may notice that it doesn't have an 'end' terminator. That is because this program uses a 'while' loop set up in a way to make the program continue indefinitely.

Making an infinite loop

Sometimes you may want to write a program that doesn't have an end, but loops forever. In programming, this is often referred to as an infinite loop.

You can use an infinite loop to make the program planned out in the flowchart.

Look at the following program:

```
1
2 #-----Setup-----
3
4 import Ed
5
6 Ed.EdisonVersion = Ed.V2
7
8 Ed.DistanceUnits = Ed.CM
9 Ed.Tempo = Ed.TEMPO_MEDIUM
10
11 #-----Your code below-----
12 while True:
13     Ed.ReadClapSensor()
14     while Ed.ReadClapSensor() != Ed.CLAP_DETECTED:
15         pass
16     Ed.LeftLed(Ed.ON)
17     Ed.TimeWait(50, Ed.TIME_MILLISECONDS)
18     Ed.LeftLed(Ed.OFF)
19     Ed.TimeWait(50, Ed.TIME_MILLISECONDS)
20
```

This program is represented by our flowchart and includes an infinite loop.

Look at line 12 of the program, which uses a 'while' loop.

'While' loops always need a condition. The loop will repeat any indented code while that condition resolves to 'true'.

If we want the 'while' loop to repeat infinitely, instead of giving a condition the program must evaluate, we replace the condition with 'True'.

'True' always resolves to 'true'. By setting the condition to 'True', we have hardcoded the condition of our 'while' loop to be 'true'.

In programming, hardcoding is where you force something to be a specific way by explicitly typing it out.

By using 'True' as the condition for the 'while' loop in our program, the condition of the loop can never be false and will repeat infinitely.

Your turn:

Write the above code to program your Edison robot to make the left LED flash when you clap. Download it and test to see how it works.

1. What is the furthest distance away from your Edison that you can be and still have the robot sense when you clap?
