**More programming in EdPy**

|  |
| --- |
| **Waiting before each crosswalk intersection** |

**Let’s suppose that we want Edison to start from the starting point B and drive straight till the end of the road, stopping at each intersection, flashing its lights and waiting for pedestrians to cross the roadway. All we have to do is program Edison to:**

* Flash its left led for 1 millisecond and then its right led for 1 millisecond every time it stays motionless.
* Cover a specific distance (in cm) and stop at the first and then at the second intersection.
* Continue driving till the end of the road.

The program that you create should look like the following:

|  |  |
| --- | --- |
| 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 | a = ………… |
| 13 | x = 0 |
| 14 | while x < ……….. : |
| 15 | Ed.LeftLed( …………….. ) |
| 16 | Ed.TimeWait( ……………………………. , ………………………..…………………. ) |
| 17 | Ed.LeftLed( …………….. )  |
| 18 | Ed.RightLed( …………….. ) |
| 19 | Ed.TimeWait(……………………………. , ………………………..…………………. ) |
| 20 | Ed.RightLed( …………….. ) |
| 21 | Ed.Drive( ………………………… , ……………………………. , …………………………….. ) |
| 22 | x=x+a |

**Let’s complete it step by step!**



Starting point: **B**

Start

**1) Program the lights to flash**

The following incomplete part of the code is supposed to make the Edison robot flash its lights every time it stays motionless. Keeping in mind that it has to flash each led for 1 millisecond, complete the code with the appropriate commands:

|  |  |
| --- | --- |
| 15 | Ed.LeftLed( …………….. ) |
| 16 | Ed.TimeWait( ……………………………. , ………………………..…………………. ) |
| 17 | Ed.LeftLed( …………….. ) |
| 18 | Ed.RightLed( …………….. ) |
| 19 | Ed.TimeWait( ……………………………. , ………………………..…………………. ) |
| 20 | Ed.RightLed( …………….. ) |

**2) Add the drive function in the program**

Complete the following command, setting its parameters to make the Edison robot start from the starting point B and move straight till the first intersection. Keep in mind that Edison stays motionless at the starting point B, so it also has to flash its lights before it starts moving. This means that the following drive function has to be placed in the program **after** the flashing light’s code.

|  |  |
| --- | --- |
| 21 | Ed.Drive( ………………………… , ……………………………. , …………………………….. ) |

**Do you remember what parameters we have to set in the Ed.Drive command?**

**One of the parameters that has to be set in the Ed.Drive command is the distance. The unit value of this parameter is set by Ed.DistanceUnits in the setup code and the default unit for Edison V2.0 is centimeters (Ed.DistanceUnits = Ed.CM). However, you can change the unit value to inches whenever you want, by setting: Ed.DistanceUnits = Ed.INCH.**

Since the value of the distance is unknown, we have to measure the printed road with a ruler to find the distance between the two intersections. Measure the distance from the starting point B to the first intersection and then add the value in the Ed.Drive command.

Open EdPy and write the code that you’ve just completed (lines 15 – 21 of the example above). Then, download it to Edison and test it! Does it flash its lights and then move to the first intersection? If it doesn’t, make the appropriate changes in the code and test it again!

|  |
| --- |
| **Using variables in our program** |

**3) Adding variables**

As you can see, there is one more intersection till Edison reaches the end of the road. Measure the distance from the first intersection to the second one. Did you notice that it is distance is the same?

The fact that both distances are the same gives us the opportunity to simplify the whole process by creating a variable called “a” and putting it in the code instead of a number. Its value will be equal to the measured distance between two intersections and must be written at the beginning of our program (in line 12):

|  |  |
| --- | --- |
| 12 | A = ………… *#a = the measured distance between two intersections (in cm)* |

After setting the value of the variable “a” as the value of the measured distance make the appropriate changes in the program, so where there was previously the number showing the value of the distance, there is now the variable “a”. At this point, your code should look like the following:

|  |
| --- |
| a = …………Ed.LeftLed( …………….. ) |
| Ed.TimeWait( ……………………………. , ………………………..…………………. ) |
| Ed.LeftLed( …………….. ) |
| Ed.RightLed( …………….. ) |
| Ed.TimeWait( ……………………………. , ………………………..…………………. ) |
| Ed.RightLed( …………….. )Ed.Drive( ………………………… , ……………………………. , …………………………….. ) |

|  |
| --- |
| **Using loops** |

**4) Simplify the process using the “For loop”**

Now that the variable “a” is set, let’s take a look at our code. Is there any group of actions that is repeated?

If you look closely, you’ll observe that the actions of the drive motion and the flashing lights are repeated twice. After flashing its lights, the Edison robot moves from the starting point B, stops at the first intersection, flashes its lights again, then does the same movements till it reaches the second intersection and finally moves to the end of the road.

Use the “For loop” statement to make the robot repeat these two actions. After the “For loop” statement, you’ll have to make Edison drive forward till the end of the road.

Do you think that in this case we could use the “While loop” statement instead of the “For loop”?

**4) Make the process generally applicable using the “While loop”**

In the example of our printed road, where there are only 2 intersections and Edison has to cover a road of only 50 cm in length, it’s very convenient to use the “For loop” statement because we can easily count the number of intersections. What if the road’s length was 70 meters and we didn’t have the time to count how many intersections there were?

**The basic difference between the “For Loop” and “While loop” statements is that we use the first when we already know how many times the commands will be executed and the second when we don’t know the number of repetitions. The code that we’ll create using the “While loop” statement will work with every kind of road regardless of its length. Setting the condition: “while x < …..” means that the number of repetitions will be determined by how long or how short the road that will be covered is.**

Make the appropriate changes to your program to make it look like the example below:

|  |  |
| --- | --- |
| 11 | #--------Your code below----------- |
| 12 | a = ………… |
| 13 | x = 0 |
| 14 | while x < ……….. : |
| 15 | Ed.LeftLed( …………….. ) |
| 16 | Ed.TimeWait( ……………………………. , ………………………..…………………. ) |
| 17 | Ed.LeftLed( …………….. )  |
| 18 | Ed.RightLed( …………….. ) |
| 19 | Ed.TimeWait(……………………………. , ………………………..…………………. ) |
| 20 | Ed.RightLed( …………….. ) |
| 21 | Ed.Drive( ………………………… , ……………………………. , …………………………….. ) |
| 22 | x=x+a |

Measure the distance from the Starting point B to the end of the road and put this number on the “while” statement (in line 14):

|  |  |
| --- | --- |
| 14 | while x < ……….. : *#where x must be less than the measured length of the road* |

The “while x < ……” condition will cause the commands below it to be executed for as long as the distance that the Edison robot covers is less than the total length of the road. But, how exactly can Edison measure the distance that it’s covering?

We can program Edison to measure the distance that it covers using the following parameter and equation:

|  |  |
| --- | --- |
| 13 | x = 0 |

|  |  |
| --- | --- |
| 22 | x=x+a |

Setting the parameter “x” equal to 0 and the parameter “a” equal to the measured distance between two intersections at the beginning of the program and adding the equation x=x+a at the end of the loop statement means that each time Edison executes a loop and covers the distance between two intersections, the value of parameter “x” increases by the value of “a”. The “While” statement defines that the loop will be executed as long as the value of “x” is less than the total length of the road. Once the value of “x” becomes greater than the length of the road, the “While loop” stops being executed and Edison stops moving.

Write the program in EdPy, download it to Edison and test it! Does it stop when it reaches the end of the road? If it doesn’t, make the appropriate changes to the program and test it again!