

TITLE: Interactive interface – RELP

LEARNING SCENARIO	
School:	Duration (minutes): 90
Teacher:	Students age: 14

Essential Idea:	Interactive interface – RELP
------------------------	-------------------------------------

Topics:

- Pupils deepen their understanding of the meaning, potential and risks of programming at a society level.
- Pupils learn to use artificial intelligence.

Aims:

- Pupils are able to design, create, document, and present programs and robots that solve a particular real-life problem. Created programs include search algorithms, tables and automatic functions. Several simultaneous events happen in these programs.

Outcomes:

- Pupils create more complex games, applications or mobile applications that simulate subject matters.
- Pupils learn about the potential and features of more advanced microcontrollers.

Work forms:

- individual work
- work in pairs
- group work

Methods:

- presentation
- discussion
- interactive exercise

ARTICULATION

Course of action (duration, minutes)

INTRODUCTION

Teacher starts discussion with pupils:
 We can use micro:bit interactive interface in Windows .

MAIN PART

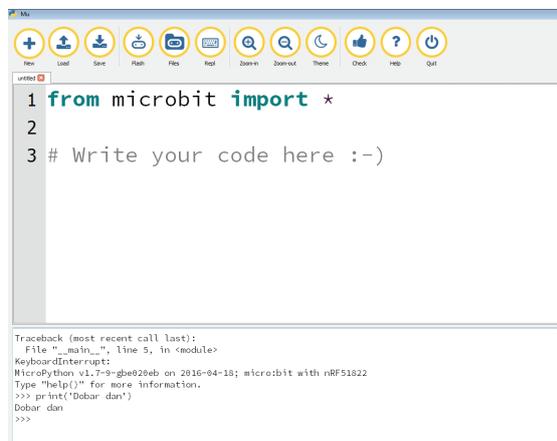
If we want to be able use micro:bit interactive interface in Windows we first need to install drivers for series communication.

They can be downloaded here: <https://codewith.mu/>



You can also just type „download the embeded windows serial port driver“ to goodle and follow the first result.

Interactive interface is started with the button Relp and at the bottom of the window we'll see a console in which we can write MicroPython commands. If you have some experience with „ordinary“ Python, try to write a program and you'll see that basic Python commands also work in MicroPython. It's important to understand that the program is executed on the micro:bit, and not on our computer.



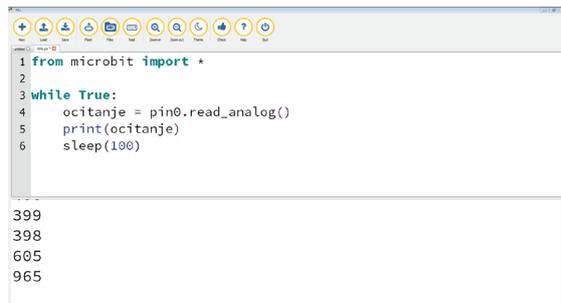
```

1 from microbit import *
2
3 # Write your code here :-)
```

Traceback (most recent call last):
 File "_main_", line 5, in <module>
 KeyboardInterrupt:
 MicroPython v1.7.9-gbe2beb on 2016-04-18; micro:bit with nRF51822
 Type "help()" for more information.
 >>> print("Dobar dan")
 Dobar dan
 >>>

If instead of using the command `print(„Hello world“)` you write `display.scroll(„Hello world!“)`, you'll see that the message „Hello world!“ is displayed not in the interface, but on the screen of the micro:bit.

We use the interactive interface to try out commands and print the data if we want to read it from the screen of our computer. In the example with the values from the photo sensor you can watch the results from the sensor on your computer screen.



```

1 from microbit import *
2
3 while True:
4     ocitanje = pin0.read_analog()
5     print(ocitanje)
6     sleep(100)
    
```

399
398
605
965

We just need to take the `display.scroll(str(ocitanje))` command and replace it with `print(ocitanje)`. Before you click on the Flash button the interface already needs to be running.

Now you can see the values from the photo sensor on your screen, and you can see how they change depending on the amount of light hitting the sensor. Put your hand over the sensor and watch the value change, or try pointing the sensor to a light source. Try to write a programme that displays a happy face on the micro:bit display if there's enough light hitting the sensor, and a sad face if the sensor is in the dark (you can set the border values yourself).

EXCERCISE

According to the previous example, pupils can design, create and test their own programs.

Examples:

<https://makecode.microbit.org>

CONCLUSION

Pupils and teacher discuss and evaluate the presented solutions.

Methods

presentation
discussion

Work forms

interview
demonstration

individual work
work in pairs

<i>work on the text</i> <i>graphic work</i> <i>interactive exercise /simulation on the computer</i>	<i>role playing</i>	<i>group work</i> <i>frontal work</i>
---	---------------------	--

Material:

- micro:bit

Literature

- <https://makecode.microbit.org>

PERSONAL OBSERVATIONS, COMMENTS AND NOTES