Lesson 5. Appendix No. 1. Worksheets

Using the light and heat sensors

Task

- Using the Micro: bit programmable microchip create applications that respond to light and temperature changes.

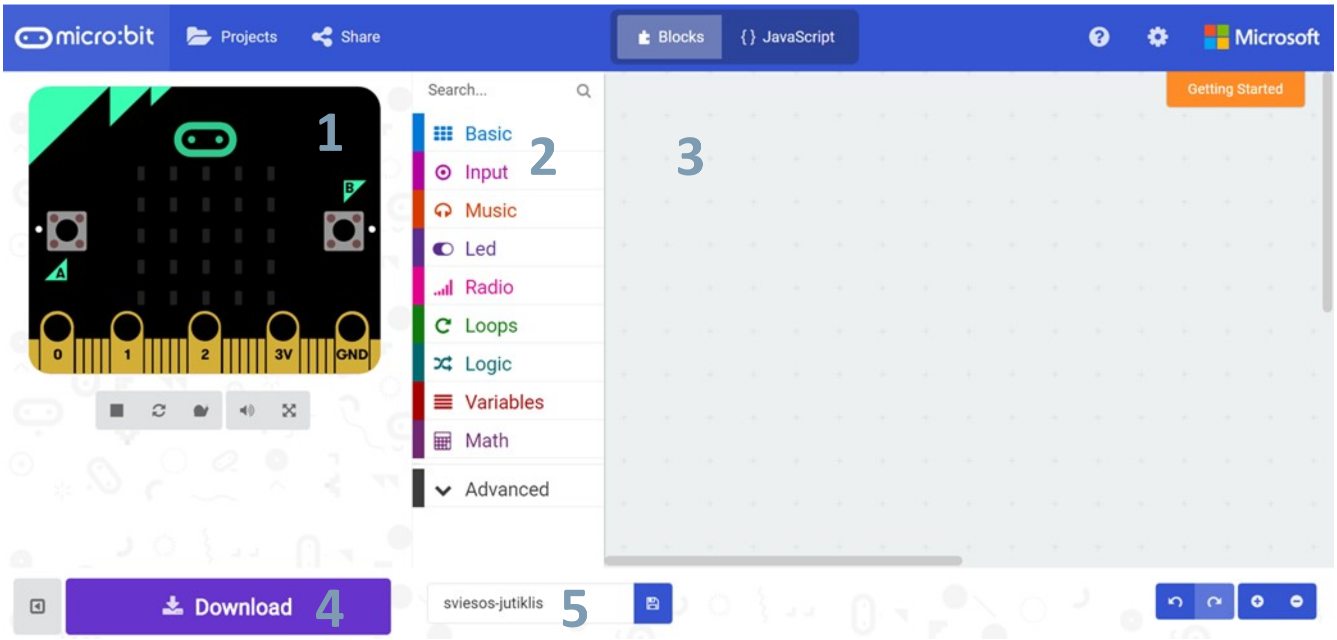
After completing this task, you will create a simple lighting and heat monitoring system.

- Lighting system principle: With sufficient lighting, red lights will turn off, and if the lighting is reduced, the Micro: bit chip will turn red LEDs on.
- Temperature monitoring principle: At low temperatures, the Micro: bit chip with LED lights indicates a warning, and if the temperature changes to the correct one, the warning is no longer displayed.

1. First of all, we have to create an **algorithm** so that Micro:bit could be applied:

Algorithm is a sequence of commands.

1.1. Micro:bit might be conveniently programmed by a computer using an internet browser. You can find the programming platform here: https://makecode.microbit.org/



1— results field. Here's how the Micro: bit program will work. In this area you can change the virtual temperature, light, press buttons to test the program completely;

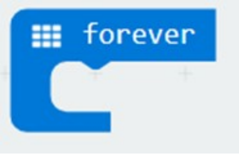
2— command field. All commands are in separate areas with titles ("basic", "input", etc.);

3—script field. Selected teams are uploaded to this field and merged into a common program;

4—program download button. By clicking on it, the created program is downloaded and can be loaded into the actual Micro: bit;

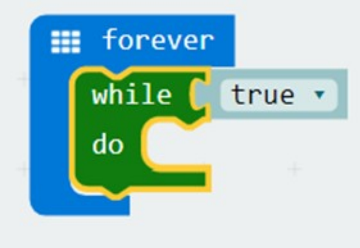
5— program title field. In this area you can write down the program title.

1.2. When creating an automated system via Micro: bit, the main command "forever" is required. You can find it in the programming section. After clicking on a command, it will appear in the programming field.



```
forever
```

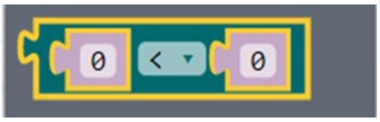
1.3. Now you need to program the condition that Micro: bit would respond to the lighting level. We take a cycle with the "while" from the Loops section and insert it into the command “forever”:



```
forever
  while true
  do
```

Cycle is the repetiion of certain commands.

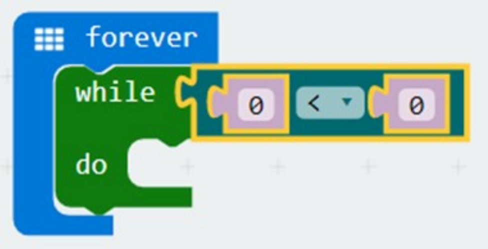
1.4. Find a command that will allow you to compare the light level with the selected one. Such a comparison command is in the “logic” section:



```
0 < 0
```

Variable are data (numbers, letters, etc.) that may vary according to the instructions or applied conditions.

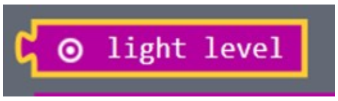
Once you have selected it, upload it to the appropriate location for the “while “cycle:



```
forever
  while 0 < 0
  do
```

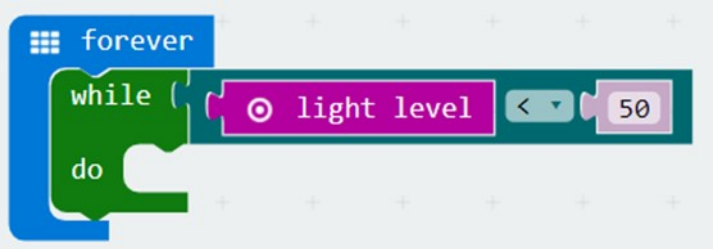
Micro: Bit evaluates illumination in numbers from 0 to 255. You need to compare the current illumination with a predetermined low light level (for example, 50).

Program the Micro: bit to turn on the lights if the lighting is less than set to 50. In this case, lighting is a variable. You'll find it in the “input” section:



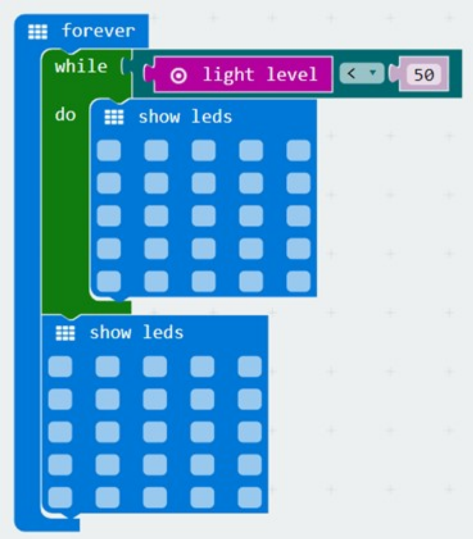
```
light level
```

Put the lighting variable on the left side of the comparison command and enter the number 50 on the right:



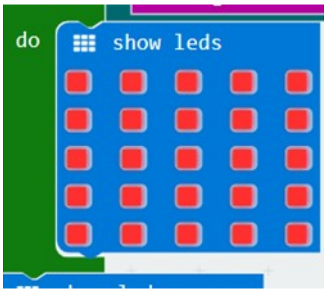
```
forever
  while light level < 50
  do
```

1.5. Finally, there are two commands left in the algorithm - turn on the LEDs and turn off the LEDs. Select the "show leds" command in the "Basic" section and upload it to two locations, in the “while” cycle and after the cycle:



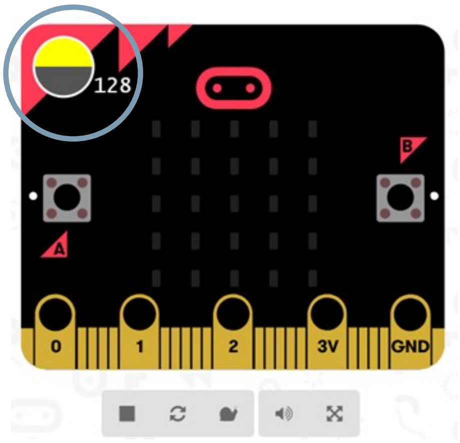
```
forever
  while ( light level < 50 )
  do  show leds
  show leds
```

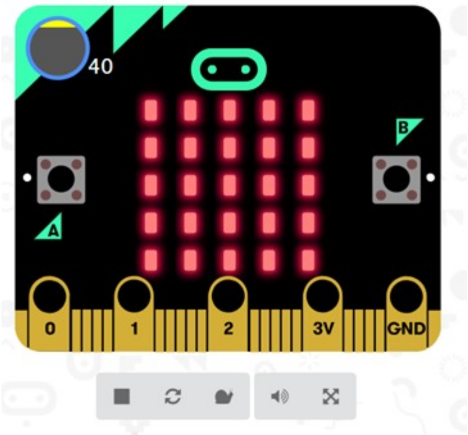
Mark all 25 mygtukus buttons „show leds“ which is in the „while“ command in order to turn on the lights:



```
do  show leds
```

2. Check if the program is working. On the left, in the results field, you must see the Micro: bit simulation. You can manually adjust the illumination (round button on the top left). When less than 50 lights are selected, red lights should light up. Try it out.



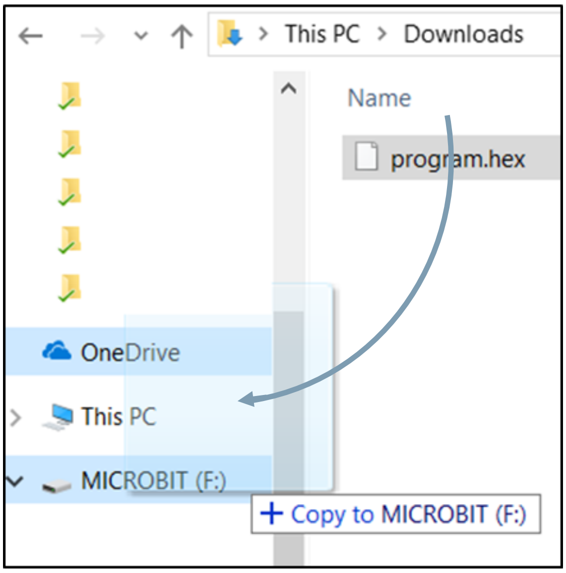


3. If you have a Micro: bit microcontroller, you can test it. To upload a project to the Micro: bit gadget, you need 3 steps:

3.1. Connect the Micro: bit microcontroller to your computer with a USB connection.

3.2. Name the project with the name you want and click on the download button. The project will have the abbreviation ".hex" at the end of the title.

3.3. Move the project from the download location to Micro: bit: Take the document and drag it to the Micro: bit folder (here is an example with Windows operating system):

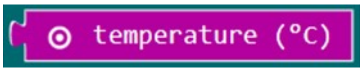


3.4. Watch the Micro: bit microcontroller. It should start working. The program will run until you upload a new program. To load another program, simply repeat the following steps (the old Micro: bit will no longer be in memory).

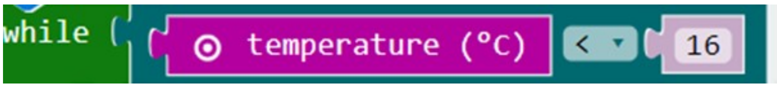
To save the program, download it and upload to your computer or external storage (as with Micro: bit).

4. Try the last program change. Program a Micro: bit microcontroller so that it does not respond to light but to heat. For example, the best temperature to grow tomatoes in a greenhouse is around 25˚C, but if the temperature drops below 16˚C, future yields can drop significantly.

Create a program that displays an alert if the temperature falls below 16˚C.

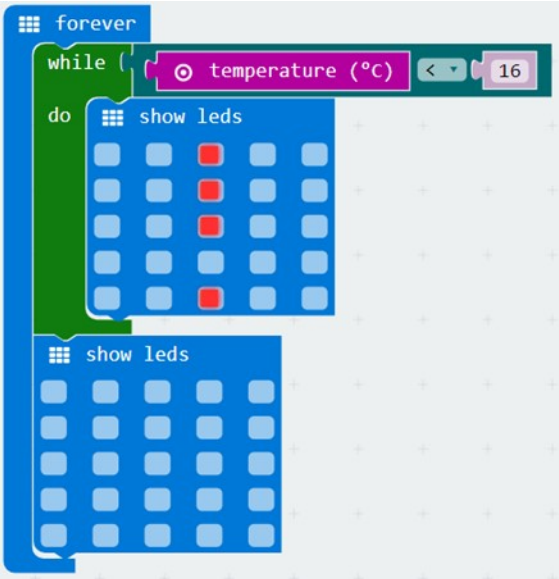
You can try to create the program by yourself or you can use the instructions. The only new required command is "temperature", which you can  find in the Input section:

4.1. The program being created remains similar to that used for light testing - a command “forever” and a “while” cycle are used. Only this time insert the temperature command to the right of the comparison team and type 16 on the left side:



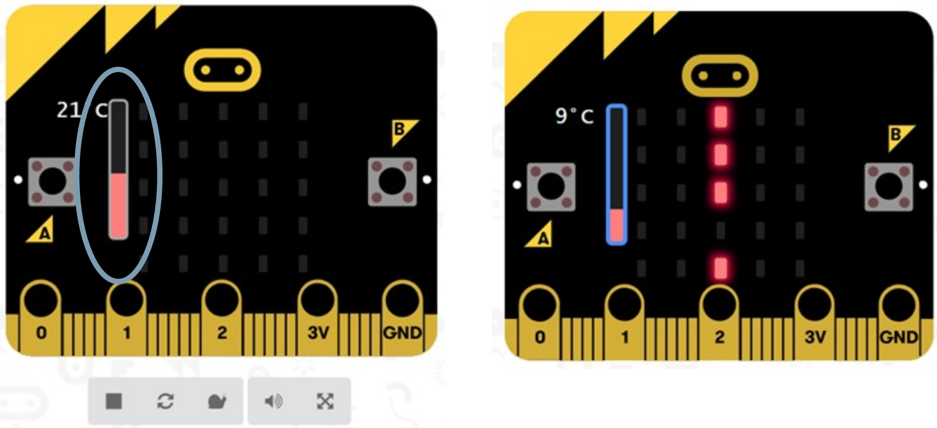
```
while ( temperature (°C) < 16 )
```

4.2. Then upload the “show leds” command to the “while” cycle and select 4 bulbs in it (draw an exclamation mark). Upload an empty “show leds” command after the “while” cycle:



```
forever
    while ( temperature (°C) < 16 )
    do
        show leds
    show leds
```

4.3. The program is finished. Now test it. First of all, test it virtually - change the temperature so that it would be lower than 16:



If the Micro:bit microcontroller works, you completed the task successfully.

Lesson 5. Appendix No. 2. Additional task. Using micro:bit buttons

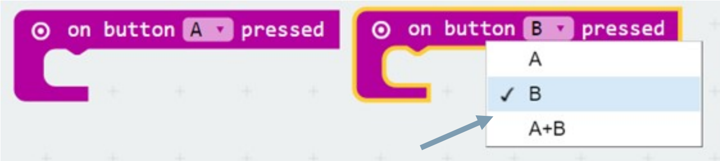
Using micro:bit buttons

Task

•Create a new program which would make Micro:bit shine only when you press the button "A" and which would stop it from shining when you press the button "B".

1. Plants are often grown in closed dark spaces. Natural light is often absent in such premises, so they are usually equipped with lighting that operates on a 24/7 basis. This solution is not always suitable for lighting plants, because they need a resting time when there is no light.

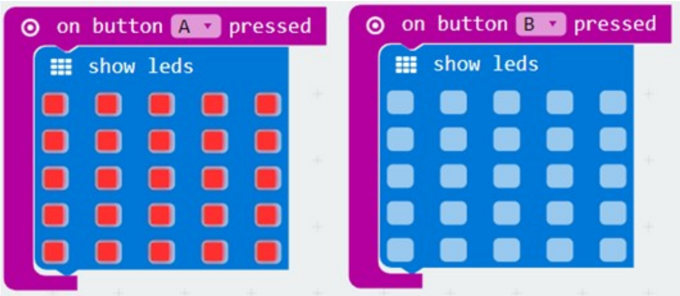
1.1. Select the command „On button A pressed". You will find it in the "Input" field. Select the same command again and choose the button "B":



on button A pressed

on button B pressed

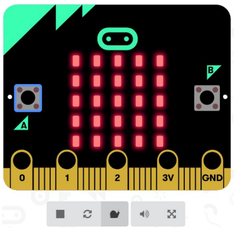
1.2. Insert the "show leds" commands to the former commands. Now the lighting will turn on only when you press the button "A". Try it.

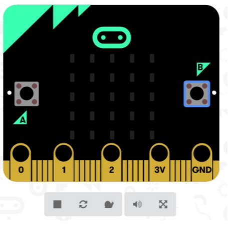


on button A pressed
show leds

on button B pressed
show leds

2. Test the program again. After pressing the button "A" (in the results field), the lights have to turn on and when you press "B" – they should be turned off.

After pressing "A": 

And after pressing "B": 

Upload the program to the Micro:bit microcontroller.