

MicroBit Beginner exercises

You are now familiar with the Micro:Bit and MakeCode. Congratulations – the world is at your fingertips!

Part 1 exercises are meant for you to discover the different possibilities in the Micro:Bit (some of them easy, some of them harder) using the different sensors and programming constructs in MakeCode. You can complete as many of Part 1 exercises as you wish, or you can move to the more advanced exercise in Part 2, if you are up for a challenge that can help save the world in its current state!

Part 1: Discovery!

1. Package Fail!

Make sure this thing never happens again. Program the Micro:Bit to let my delivery man know that he's being reckless with my expensive stuff! The Micro:Bit is far cheaper than buying a new chandelier for your castle.

2. Hide-and-seek!

Build a hide-and-seek game. The program should support:

One Micro:Bit must be the seeker and be able to call out for the hidden Micro:Bit

The hidden Micro:Bit should make a sound when requested by the seeker.

Think it is just for kids? Think again! TheTileApp has received more than \$100 million in funding! It's basically just a hide and seek game for adults who can't find their stuff (Fun fact: One of the EBS teachers owns more than 5 Tiles).

Try playing hide and seek somewhere on Campus.

Possible extension: Are there other ways to build a hide-and-seek game than with sound? Can you perhaps show a signal indicator as well to let even deaf users know they are close?

3. I'm sorry, boss, I can't come into work today! (Tribute to Chris og chokoladefabrikken)

When you are sick, you might have a fever. If you don't have a thermometer at hand, the Micro:Bit can help. But is it able to measure your surface body temperature correctly? Find out now! Where is the temperature sensor located? (To read more about surface body temperature, look here: https://en.wikipedia.org/wiki/Skin_temperature)

Possible extensions: What else would the temperature sensor be good for?

4. Geez Louise! I'm trying to sleep!

Your partner keeps turning the light on when you are trying to sleep (you prefer to sleep in the dark). Build a thing that lets them know how much light they can turn on – set your preferred threshold (experiment at home to find out what the ideal threshold is).

Possible extensions: What else is the light sensor useful for?

5. I'm better than you! (Perhaps the hardest of the part 1 exercises as it requires solving several problems)

Your classmates claims that they have faster acceleration than you when sprinting. You don't believe it for a second, but you can use your engineering skills to build a device that will prove who has it for good. Build it – and try it outside! Who wins? (Hint: Accelerations should be stored in an

array for a specific period of time – then determine the highest acceleration)

Possible extension: Who can get the furthest in a specific timespan (hint: Micro:Bit has a compass)

If you finish the exercises early, you may extend upon your programs and/or build your own thing.

Part 2: We are engineers – We build stuff and solve problems!

Fair warning: This is a difficult exercise very early in the semester. We do not expect you to build a fully fledged virus game. Try building just a little part of a full systems. Teachers have a “back-up virus” ready to play in the classroom.

There are so many ways to survive a pandemic. Some better than others... Complete lockdown has proved quite effective at stopping the spread of the virus... But it is expensive. Maybe there is a better way depending on the symptoms of the virus... Who knows, really?

Exercise:

Build a pandemic simulation game. Play the best version of the game with all your classmates.

You can now determine the best strategy to win by simulation.

If you find this exercise too difficult – perhaps you can build a social distancing device instead?

Possible specifications:

One or more Micro:Bit must be patient zero (infected). Perhaps you can designate patient status with a special type of variable? Those not infected are healthy. (Hint: Boolean)

When starting the Micro:Bit there is a small chance yours becomes the infected. (Hint: Math and randomization)

When a an infected Micro:Bit comes close to a non-infected Micro:Bit, the non-infected becomes infected. Maybe there is a likelihood for not catching it? (Hint: Radio Communication and Signal strength)

The Micro:Bit must let the person carrying the Micro:Bit know they are infected (Hint: Different output possibilities. Do you want the signal the virus using sound? Visual using LED?)

When done programming the game – try playing it. What is the best strategy for survival given your implementation?