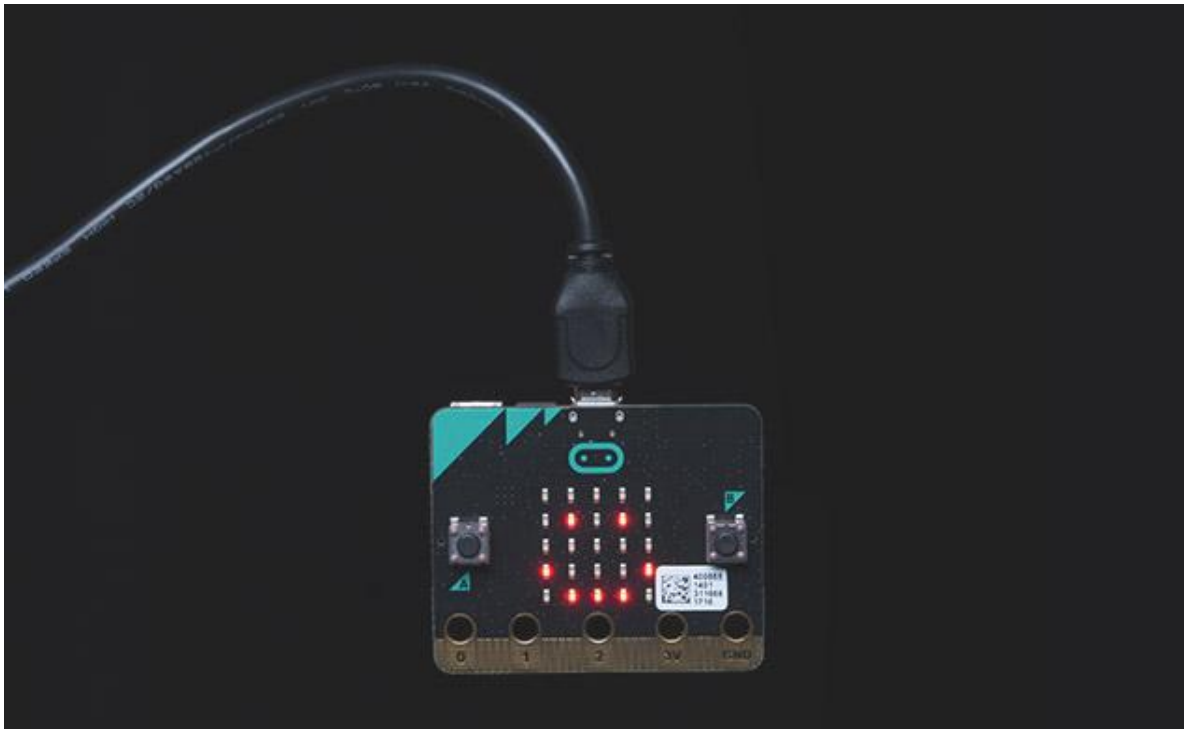


Micro:Bit v2 Virtual Pet (Tamagotchi)



By Joe Brown October 2022

Inspired by Microsoft's Pet Hamster workshop and
Stu Lowe's Tamagotchi workshop (Beacon Hill School)

Learning outcomes:

Engineering skills - trial and error, problem solving, creativity

Electronics hardware - LEDs, buttons, inputs and outputs, USB & storage

Computer programming - loops, variables, if statements, delays

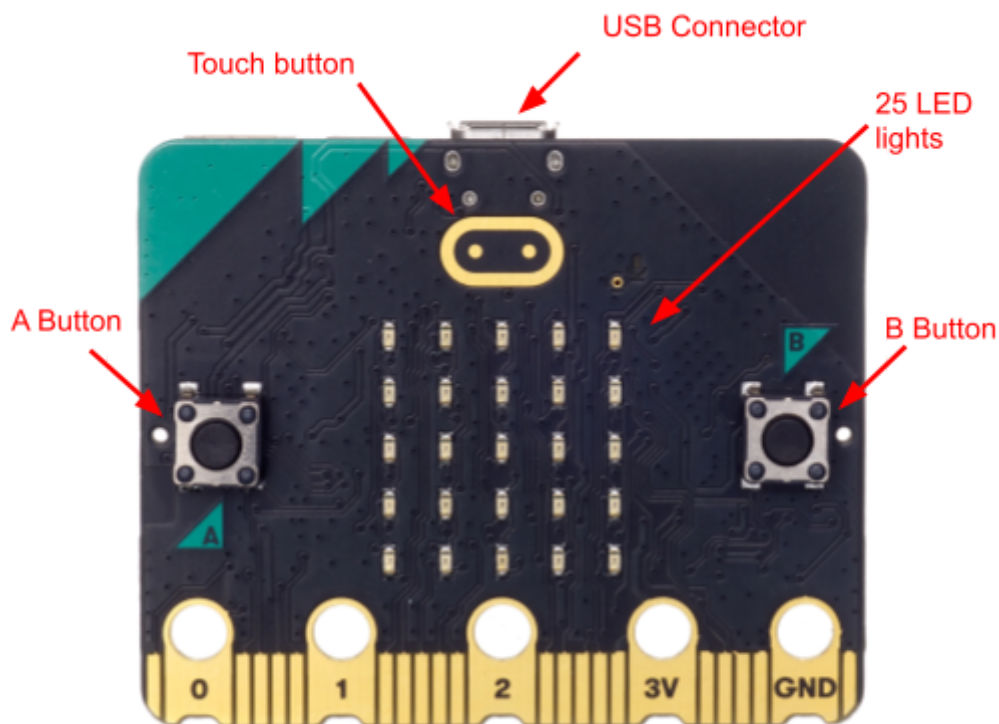
1 Introduction

1.1 Introduction

This workshop is based on a virtual pet toy from the 1990s known as the Tamagotchi:



This is a worksheet to teach you some coding concepts using 'block based editing' with microbit - this is very similar to scratch coding but the difference is your code will physically run on the microbit device rather than just on your computer! This is a microbit v2:



On the back: microphone, speaker, compass, accelerometer

For this workshop you'll use various **inputs** of the microbit to interact with your pet
Inputs: buttons, microphone, accelerometer (detects movement)

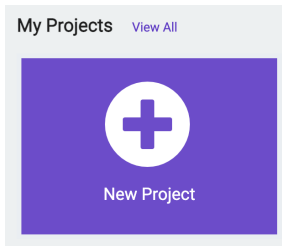
You'll use various **outputs** to display your pet's responses

Outputs: LED lights ('screen'), speaker

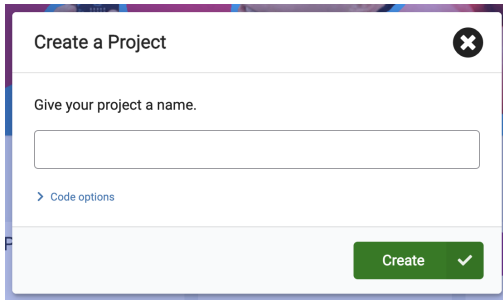
And you'll upload your code to your microbit using the USB connector.

1.2 Setup

Open a web browser, go to makecode.microbit.org and click New Project:



Give your project the name of your pet! Then click create.

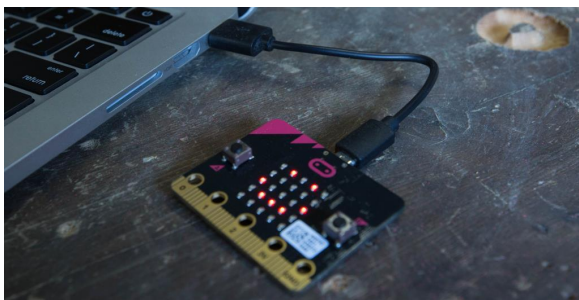


A new project will load. This is the layout of the editor, and all the buttons you'll need:



You'll be using blocks from the block library and dragging them into the code area. To test your code on your microbit, you'll be downloading the code with the download button.

You'll want to plug the micro:bit into your computer like so:

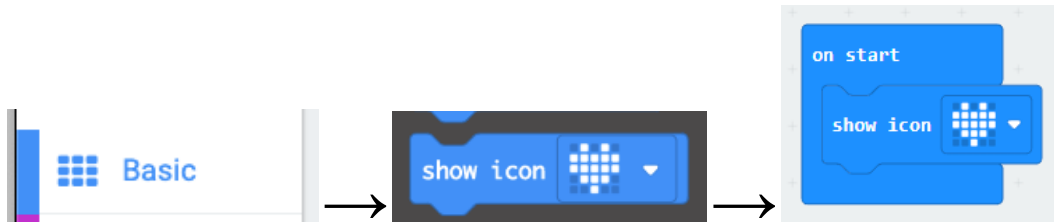


Don't worry if no lights are showing on the micro:bit yet!

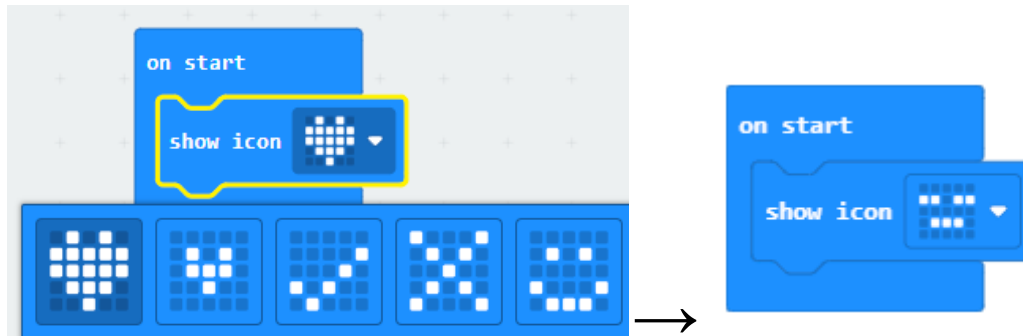
2 Pet Interactions

2.1 Showing your pet's face - Screen Output

We'll be showing your pet on the micro:bit 'screen' which is a grid of 25 LED lights which will represent your pet's face. Your pet is very sleepy! Let's show that on the microbit. Click on the **Basic** tab, click and drag the **show icon[]** block into the **on start** container:



Now change the icon by using the drop down to an asleep emoji -_- using the drop down:



2.2 Testing the code

Great now let's test our code! Make sure your microbit is plugged into your computer

 **Download**

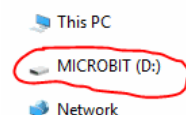
Click the **download button** and the bottom of the screen:

If you're on Edge browser on a Windows computer then follow the below instructions, otherwise ask one of our helpers how to download code onto microbit for your laptop and browser and go to the next page.

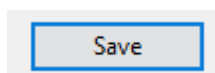
Click the **little ^ arrow button** next to the save button at the bottom of the screen:



click **Save as:**  then **MICROBIT** from the side bar:

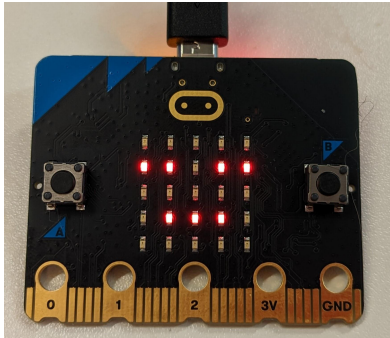


then click the **Save** button:



Ask for help if you're stuck.

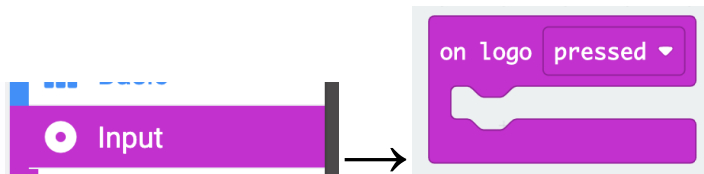
When you click save, you should see the back of the microbit flashing yellow. After it's finished downloading, you should see your sleepy face on the physical microbit:



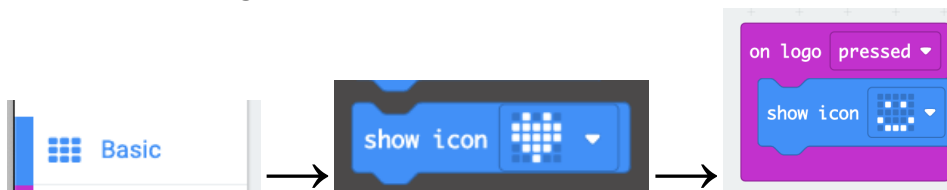
2.3 Tickling Your Pet - Touch Input

Now let's add an input - tickling your pet.

Click on the **Input** tab, click and drag the **on logo [pressed]** container to the code area - it's near the bottom of the inputs list.



Now add a **smiling face** from the **Basic** tab > **show icon[]** block:



Download the code to and test it!

If you touch this part of the microbit  does your pet smile?

2.4 Pet giggle - Sound Output

We've got input from the touch button, output on the screen, let's add some sound.

Grab a **play sound [giggle] until done** block from the **Music** tab

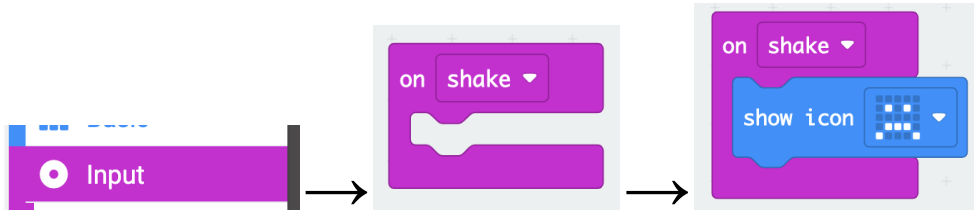
Then add it to the **on logo pressed** block:



Download the code to and test it, can you hear your pet giggle when it smiles?

2.5 Shaking your pet - Accelerometer Input

Let's add another input - the accelerometer is a sensor that detects when the microbit is being moved! That's in the **input** tab, called **on [shake]** container and add a frowning face by the **Basic** tab **show icon []** block:



Add a **play sound [giggle] until done** block from the **Music** tab and change **giggle** to **sad**.

Download your code to the microbit and test it out - when you shake your microbit you should see a sad face and hear a sad sound from your pet!

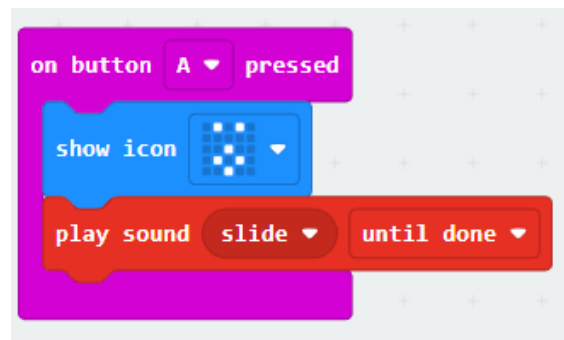
Tip: you can get blocks quicker by right clicking > duplicate from a block you've already used! - it's like copy pasting - you can use this to get **show icon** and **play sound** for 2.6:

2.6 Food

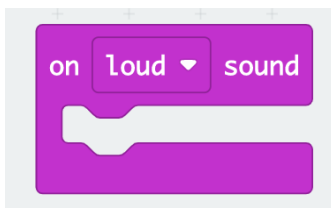
Don't forget you need to feed your pet! Let's set that up now. It's similar to the previous few blocks so see if you can find **on button [A] pressed** container.

And a **show icon []**, **play sound []** block like before - but change their emoji and sound to ones suitable for eating food!

Once you've added all the blocks - *download your code to the microbit and test it.*



2.7 Optional Extension: Microphone



If the room isn't too loud you could add another interaction with your pet - using the microbit's microphone to detect a loud sound such as a clap - using the **on [loud] sound** container.

Add an **show icon []** & **play sound []** block inside the **on [loud] sound** container. Choose an emoji and sound of your choice to create the reaction to the loud sound from your pet!

3 Variables and if statements

3.1 Sleepy Variable

Now we're going to create a 'variable' called sleepy.

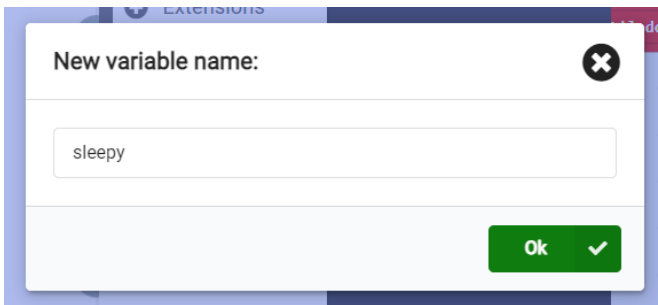
A **variable** is a block with a value that is remembered by the micro:bit - we're going to use it to remember if the pet is sleepy or fully awake.

If sleepy = 1, your pet is sleepy. If sleepy = 0 your pet is fully awake!

Click the **Variables** tab and **Make a Variable...** button:

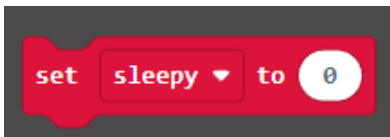


Name it 'sleepy' and click ok:

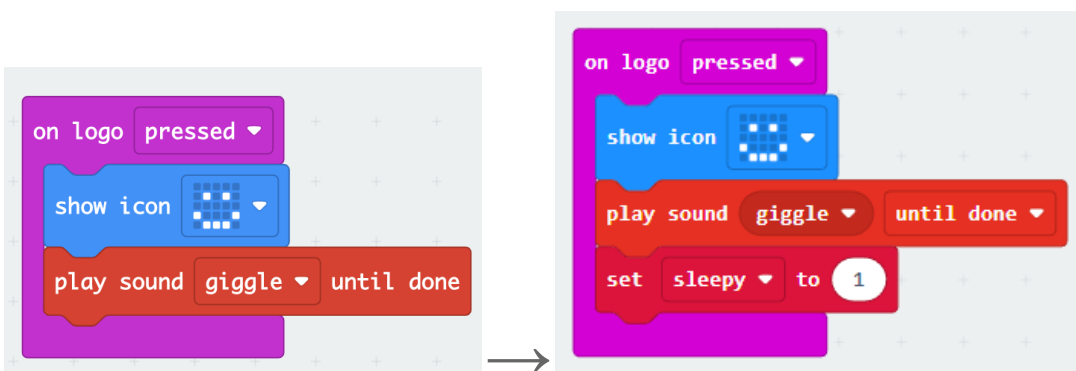


Let's make it so if you tickle the pet (touch the logo) it gets sleepy afterwards!

Go to the **Variables** tab and **grab a set sleepy to 0** block



And let's add it to **on logo pressed** and set it to 1:



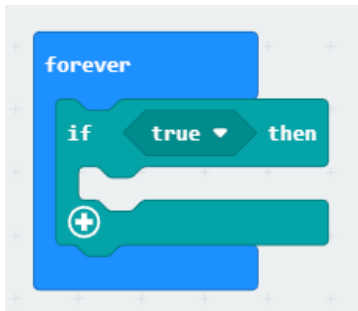
3.2 If statements

Now we need to show that the pet is sleepy on the screen!

'If sleepy is 1 then show a sleepy face' is our aim. We can do that with blocks of code:

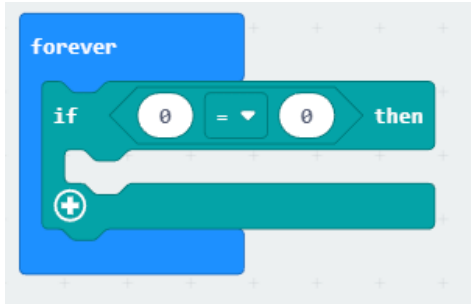
Go to **Basic** tab and get a **forever** container (or use an empty one that's already there!)

go to **Logic** tab and get a **if <true> then** block and put it in the **forever** container:



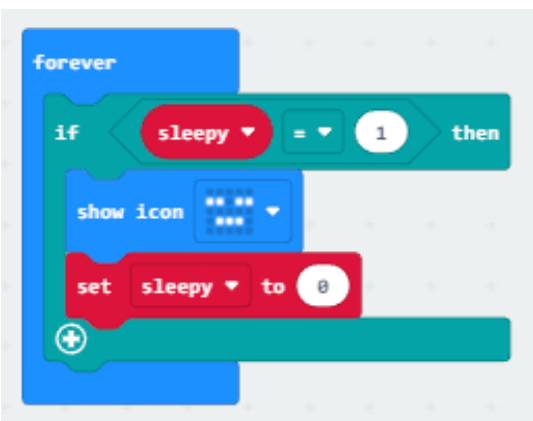
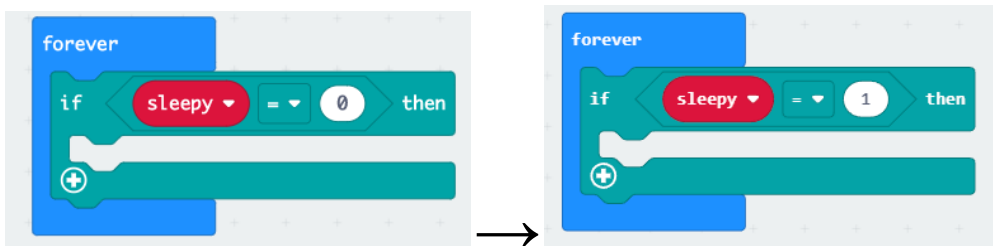
Forever blocks keep looping around and around forever! When it finishes running the blocks, it repeats again and again from the top.

Next Logic tab and get a **<0 [=] 0>** block. Drag it where <true> is:



Go to **Variables** tab and get a **(sleepy)** variable.

Drop it in the left hand side of the **<0 [=] 0>** block. Change **sleepy = 0** to **sleepy = 1**



Now add the **show icon** block from **Basic** tab and set it to a sleepy face.

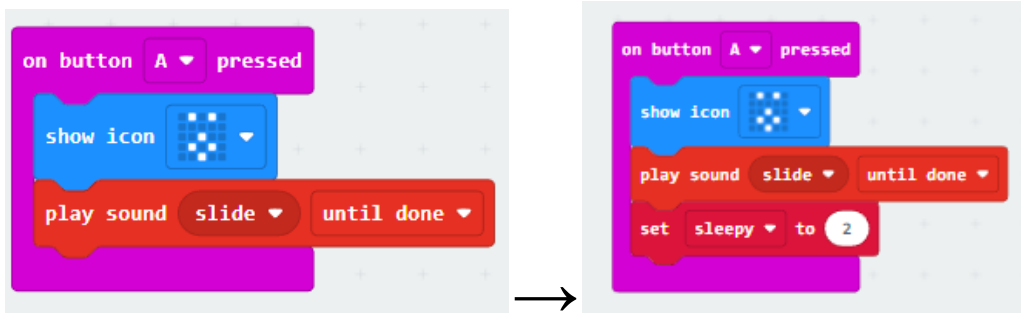
And so your pet isn't sleepy forever, let's reset the sequence by **Variables** tab **set [sleepy] to 0** block

Download your code and try it! If you touch the icon does your pet get sleepy after smiling?

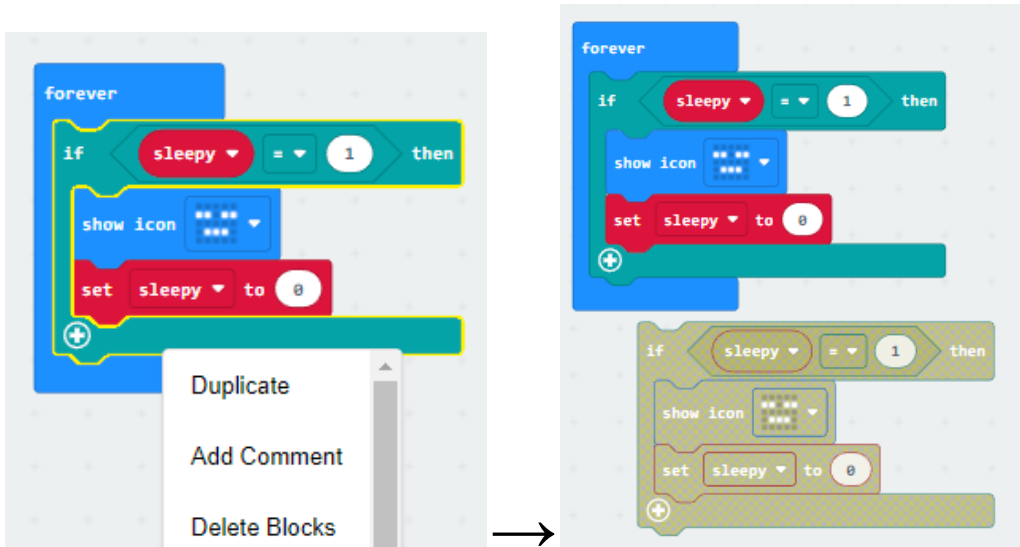
3.3 Getting sleepier!

How about when you feed your pet it gets even more sleepy?

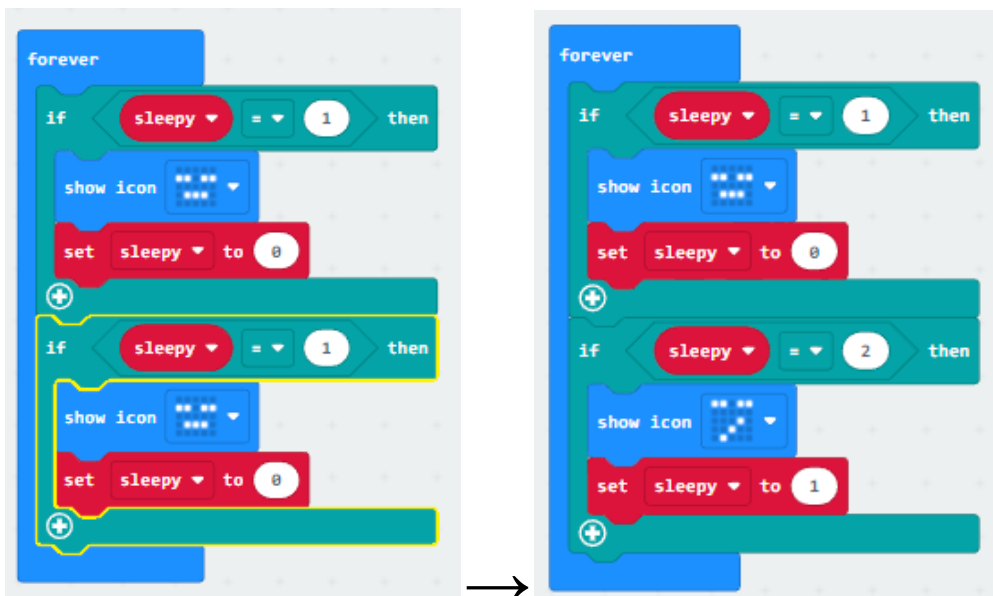
Let's try that with another **set sleepy to 0** block but this time set it to 2!



We need another if statement in the **forever** container for **if (sleepy) = 2** - remember we can use right click > duplicate the **if (sleepy) = 1** to save us finding all the blocks manually:



Add the block below the first if, setting **if sleepy = 2**, a sleepier face, and **set sleepy to 1**



What do you think this code will do when you press the A button?

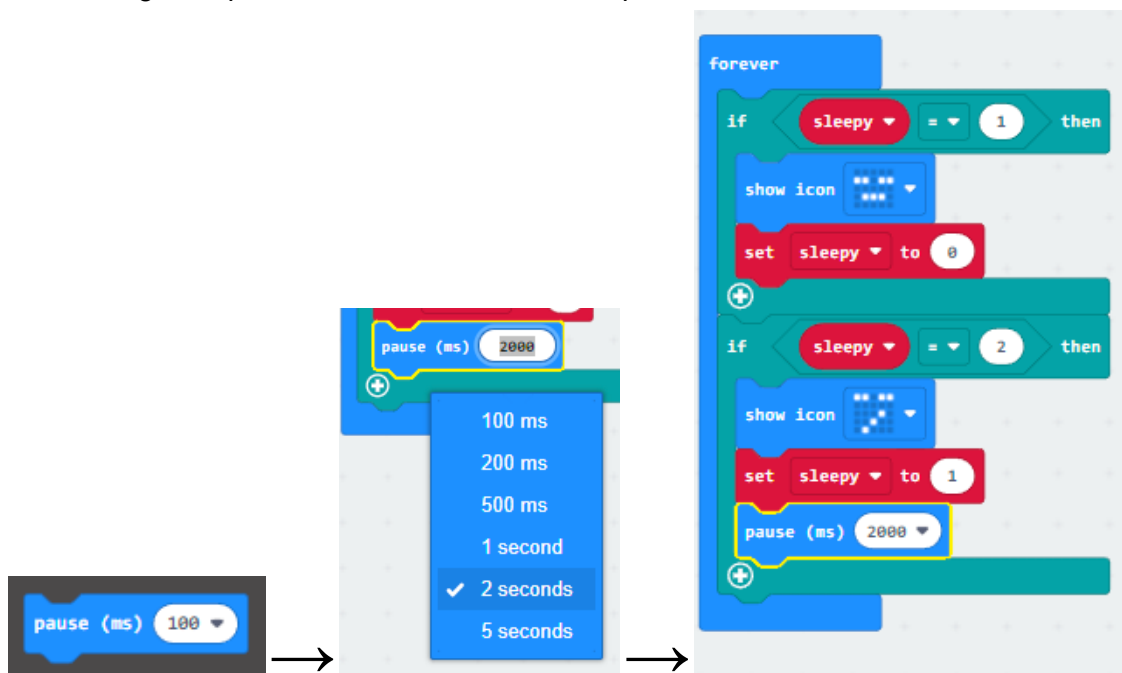
What will the screen do when sleepy = 2 then gets changed to sleepy = 1 then sleepy = 0?

Now download it to the microbit and test it out!

3.4 Pause

Pause are useful in code for slowing things down - computers work at very fast speeds and we want things to be visible to us humans!

Let's use a pause for our new sleepier if statement, in the **Basic** tab as **pause (ms) 100**:
Let's change the pause to 2 seconds on the drop down



So now we should see when you feed the pet - it's happy then very sleepy.
2 seconds later it will go back to normal sleepy!

Download this and test it out.

What happens if you change the size of the pause?

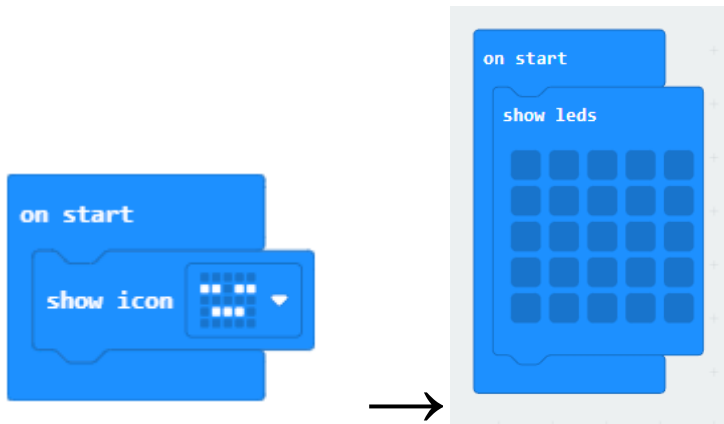
This is the end of the main workshop, but if you're finished early we have lots of extensions for you to try:

Section 4 & 5 is about customising the appearance of and animating your pet!

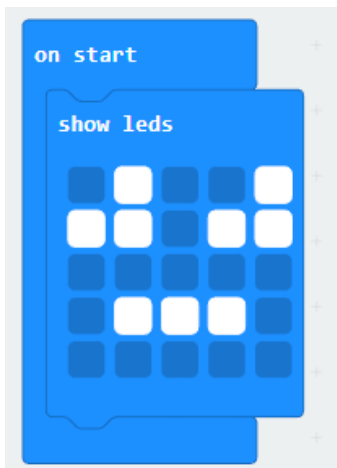
Section 6 is adding more game code so that your pet acts more like a tamagotchi - don't let it get bored for too long!

4. Optional Extension: Customising Faces

You can create your own custom faces for your pet. Go to the **Basic** tab and grab a **show leds** block - swap your **show icon** blocks with in the **on start** container.



'LEDs' stands for 'light emitting diodes' and it means the lights acting as a screen on your microbit - you can click on various squares within the **show leds** block to create your own faces for example:

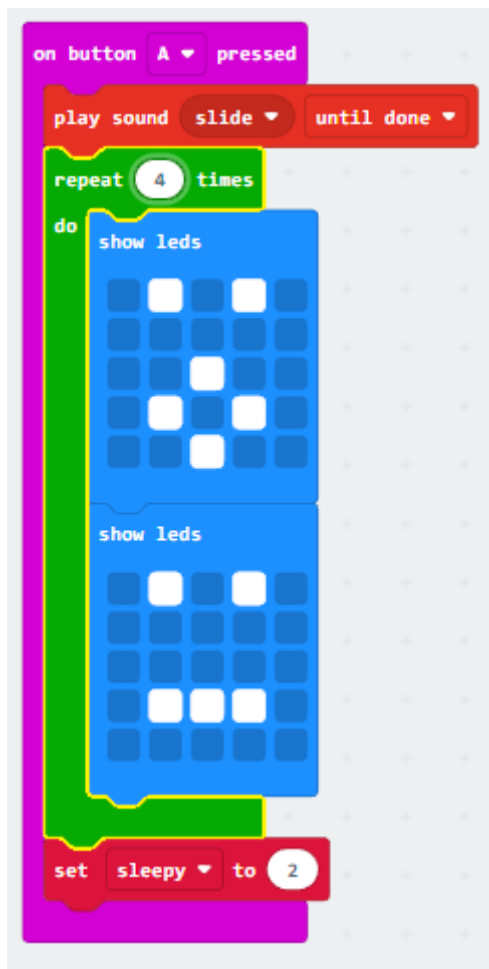


Replace your **show icon** blocks with your custom **show leds** faces for your:

- Sleepy - on start and in forever block
- More sleepy - in forever block
- Happy face- on logo pressed
- Sad face - on shake
- Eating face - on button A pressed

5. Optional Extension: Animated Faces

You can use the **repeat 4 times** block in the **Loops** tab to create animations for different emotions For example: eating when you press the A button:



This will loop these two **show leds** blocks 4 times making it look animated - in this case eating! You can choose different amounts of repeats - keep it less than 6!

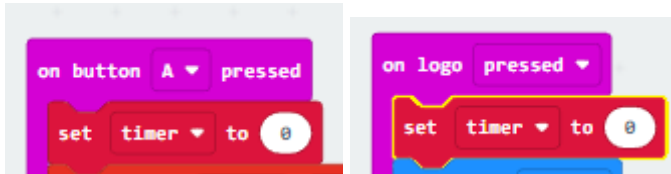
Why not add animations for happy, sad and sleeping too?

6. Optional Extension: Keep your pet busy!

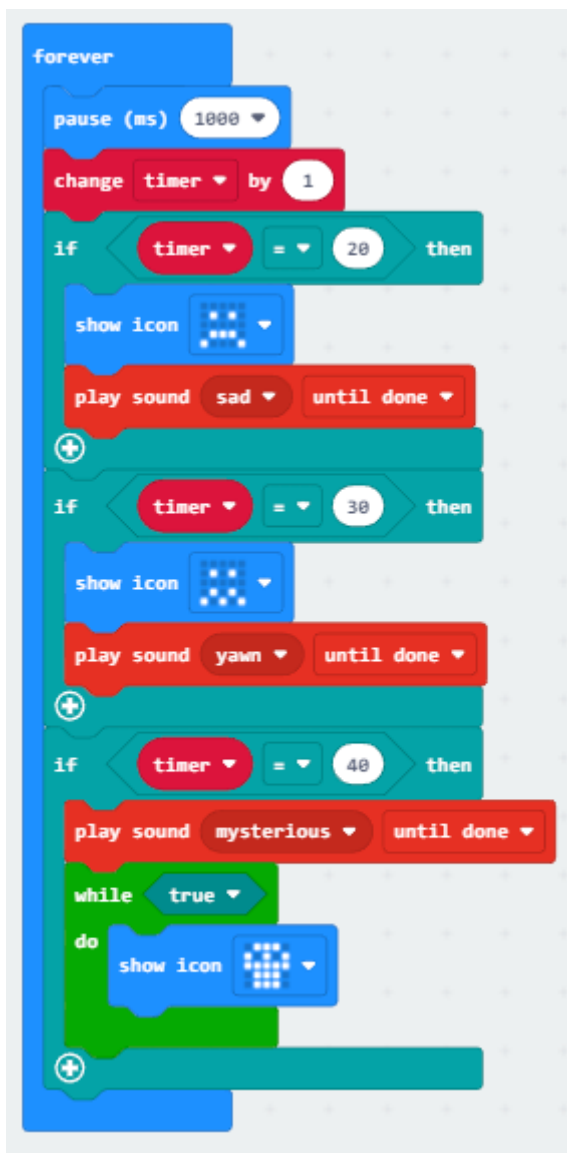
Add this code to set up more actions which happen when you stop interacting with your pet. Be careful not to leave your pet alone for too long! - this is just like a tamagotchi.

Make a **variable** called 'timer' (in the **Variables** tab)

add a **set timer to 0** block to the start of on **button A pressed** & on **logo pressed** blocks.



Now let's create some logic for if you pet gets bored - we'll count how many seconds since you have pressed the A button or the logo button:



The **forever**, **pause 1000ms**, **change timer by 1** is counting up each second.

If you press A or the logo you're resetting the timer variable to 0.

We then use **if statements** to show the pet getting sadder at 20 seconds,

30 seconds we're showing it even sadder!

and then at 40 seconds we're using a **while** block from the **loops** tab to show the skull icon forever past this point - until you reset the microbit and start a new pet!

Tip: there's a reset button on the back of the microbit if you want to respawn your pet!