



Welcome to CoderDojo IRD Duhallow Intermediate Scratch and Beginner Arduino.

Over the next few sessions we will be working on a fun and useful project.

We will start with scratch to work through the steps we need to do, and later take the things we learn and use them.

We will be learning how to use a small computer called an Arduino UNO.

This little computer is really easy to use and can be programmed to do all sorts of cool things.

We will be using it to control a lot of LED lights.

We hope that you are looking forward to this as much as we are.

Remember, we are here to help, we will go through everything with you and it is okay to make mistakes - that's often the best way to learn.

Coder Dojo Team.

So, what are we up to?

Well, we are going to build a sequence of lights that can be used to help runners to train.

The Arduino will turn on and off the lights, and the runners need to match the position of the lights.

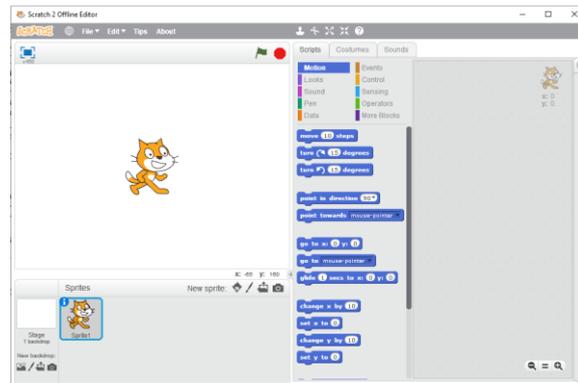
This will mean that the runners can train better.

Before we use the Arduino, and build the scale model, we need to figure out how we are going to do it.

For this we will use scratch, as it is easy to work with.

STEP ONE: Let's Go!

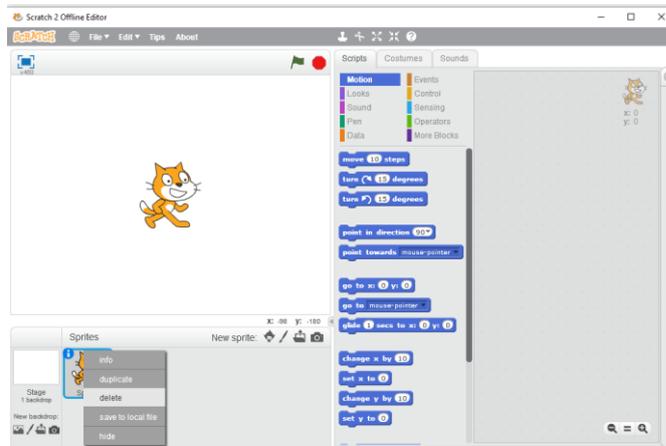
Open Scratch.



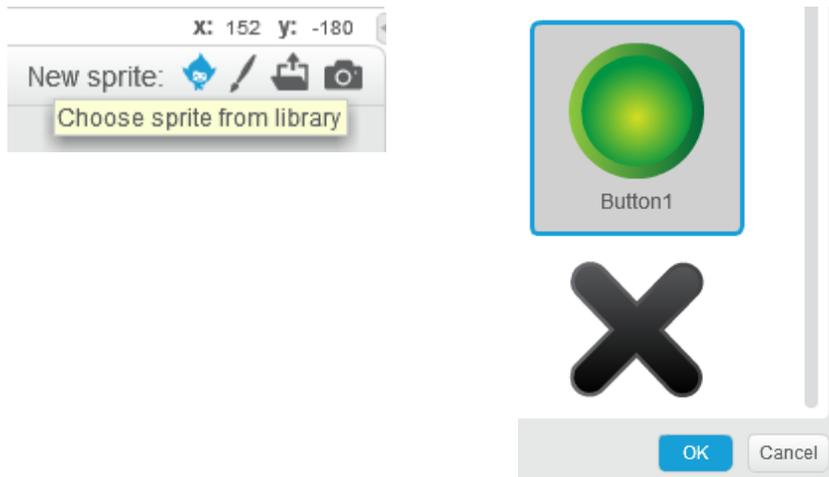
STEP TWO: Meow!

GET RID OF THE CAT!!

Right Click on the sprite and click "Delete".



Then add in the green "Button 1" sprite. This looks like the LEDs we will use.



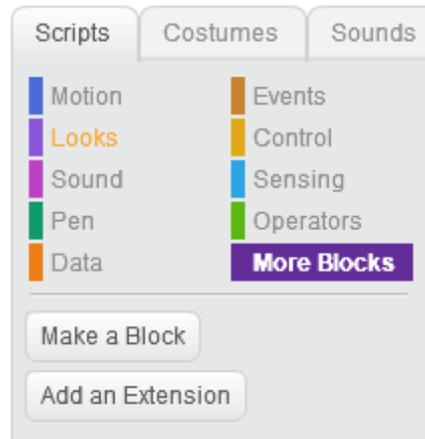
STEP THREE: Sequences

We are going to start with a very easy sequence.

A sequence is a group of programming blocks grouped together to do a specific action.

For this exercise we just want the Green “LED” to flash on and off every 1 second.

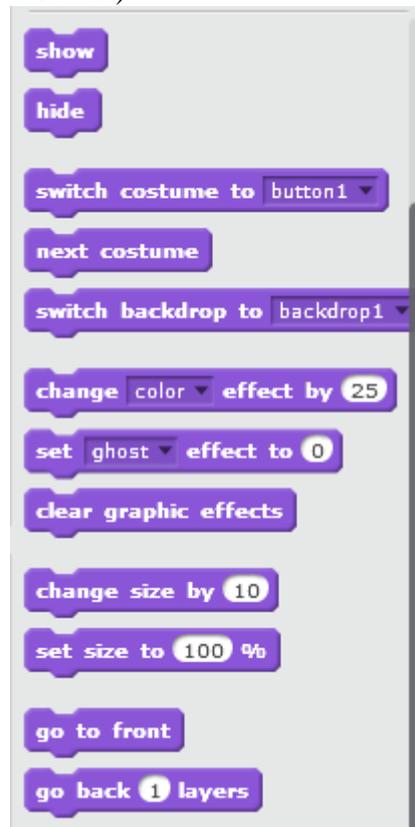
Can you remember or work out which group of scratch “Scripts” blocks we can use to make something appear and disappear?



YES! It's “Looks”.

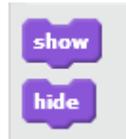
In this group can you see which blocks we could use?

(Hint: There is more than 1 way to do this.)

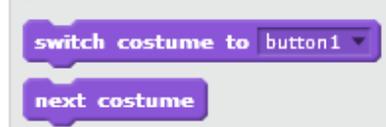


NO PEEKING!!

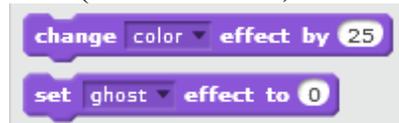
Solution 1: We can use **SHOW** and **HIDE**.



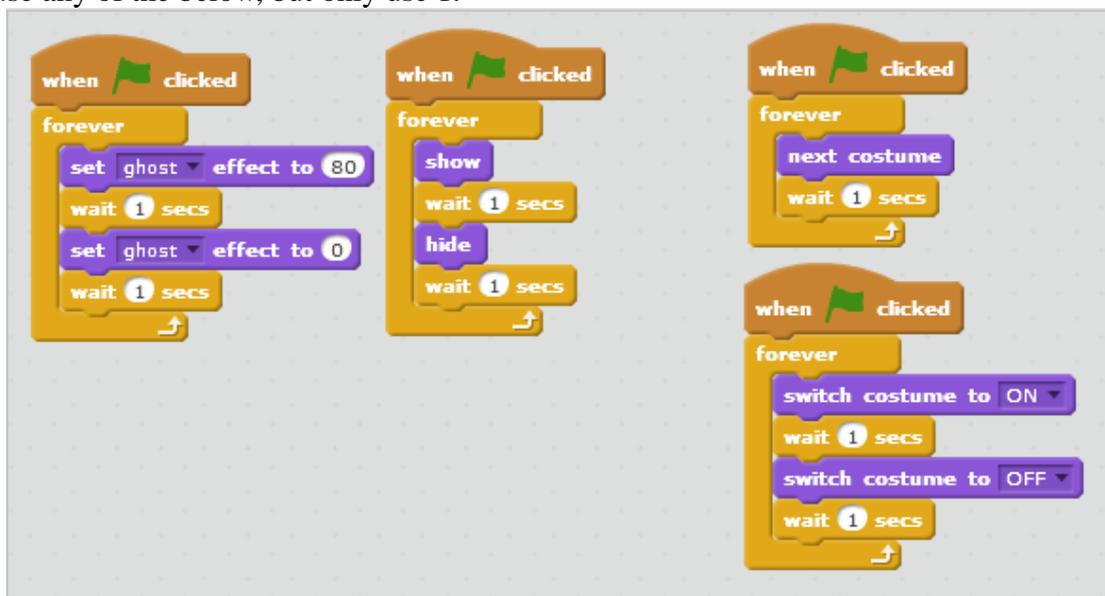
Solution 2: We can use a costume change – if we have a blank costume for “Button 1”.



Solution 3: We can use the “Ghost” effect. (100 = invisible)



The choice is up to you, remember this is only to work out what we will do later. I'm going to use “Ghost”, because it will be easier to see the “off” lamps later. You can use any of the below, but only use 1.



When you are done, save your scratch program.

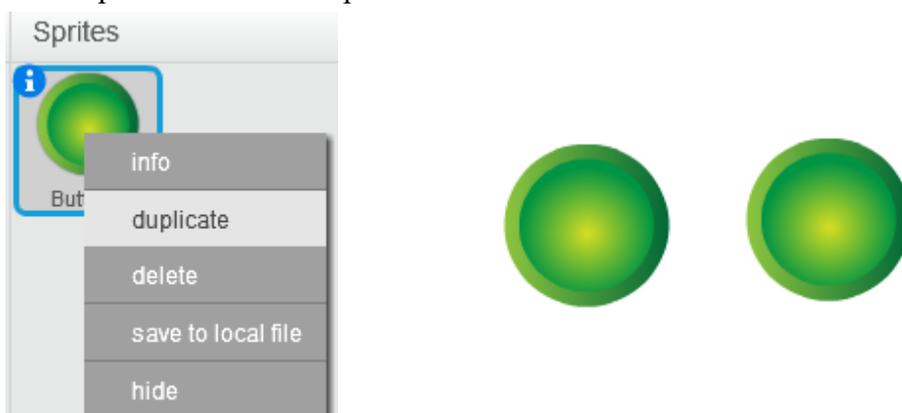
Then click the green Flag.

Does it work? Is the Green “LED” flashing on and off every 1 second?

If it isn't can you figure out why not, if you are stuck ask someone else to check your code.

When you are ready we will move on and add in a 2nd “LED”.

Right Click on the Sprite and click “Duplicate”



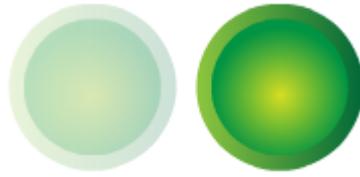
You should now have 2 identical Green “LED”.

When you press the green flag what do you expect to happen?

“**Duplicate**” copies everything – so it takes all the code as well. This is a great shortcut so you don't have to redo all your code - **MAKE A NOTE OF THIS, IT WILL SAVE YOU LOTS OF TIME**

BUT, this isn't really what we want to happen.

We would like if the 2nd LED comes on after the 1st LED goes off. How could we now do this?



What would happen if you reversed the order of your code for the 2nd Sprite?

If you used “Show” first and then “Hide” switch the order to “Hide” and then “Show” for sprite2.

Now what happens?

Is this right? **Remember we want both lights to be 1 second apart.**

ANY QUESTIONS?

What about changing the “Wait” times?

If we turn on LED1 at the start for 1 second, then we turn it off and 1 second later turn on LED2 and then 1 second later we turn this off and 1 second later we turn on LED 1 again... and repeat forever. How would this look.

Take a notepad and write it out. This is call a program flow, or an execution path.

```
START
LED1 ON
Wait
LED1 OFF
Wait
LED2 ON
Wait
LED2 OFF
Wait
LED1 ON.....
```

Can this be programmed into these 2 LED Sprites?

TRY IT – Can you work out how many seconds to use, and where to use them?

***If you draw a time-line under your note it might be easier to see.**

This note is called”Pseudo-Code” - It isn't actual code, but is allows you to see what you might need to do.*

So, can we see that this isn't a good solution to a timing problem?

Even when we only have 2 LED lamps it is difficult.

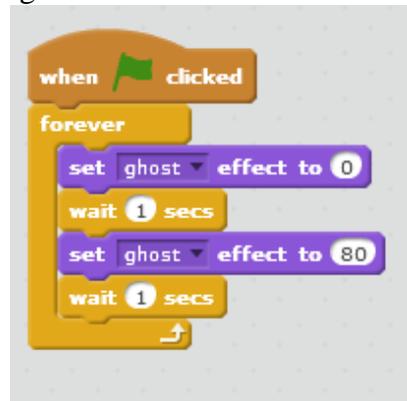
How much more difficult will it be when there is 6-8-10... 16!! LEDs?

Does everyone see why this would cause problems?

QUESTION? Is there a better way to control multiple LED?

STEP FOUR: Messages.

Messages are program blocks that can be used to control sprites.
We can use them to do things with a single sprite, or with many sprites.
Go back to the 2 sprites that you have.
This is my code – remember yours might be different.

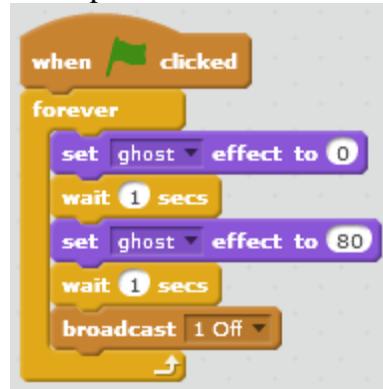


Go to the “Events” group and look for “broadcast [message1]”



If you click the black arrow you can create a new message – call this something that that will be easy to figure out. How about “1 Off”?

Add this to the end of your code blocks for sprite 1.



Next go to sprite 2. Can you see anything in the “Events” group that you can now use to tell this sprite what to do? **HINT: It is to do with the message we just made in Sprite 1.**



If you click the “Green Flag” now, does it work **EXACTLY** as you want it to?

If not, check your code – you might need to change some of the sequence.

At this point can we decide if “FOREVER” is a good choice to use in this part of the program?

STEP FIVE: Endless Chatter.

Messages, continued.

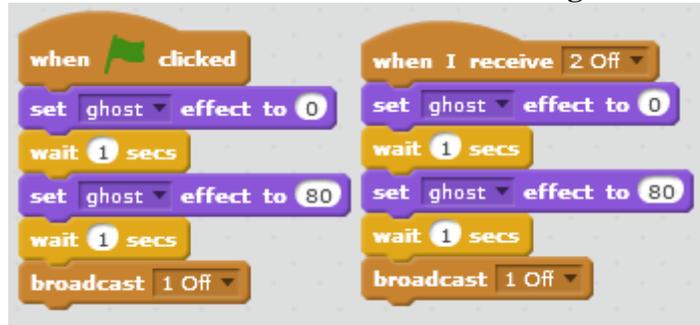
If we remove “Forever” from the program the sequence only happens once.

Can you explain the sequence?

Start-->1 ON-->wait--> 1 OFF-->wait-->broadcast 1 Off--> 2 ON-->wait-->2 OFF--> wait-->???

If we want Sprite 1 to come on again we have to tell it.

Can you work out how we could do this? **HINT: Make a new message somewhere?**



Now, we are using the green flag to start the program, but using the message from sprite 2 to loop it?

Try it now, does it do what we wanted?

Again, if your program doesn't seem to be working well, see if you can figure it out. Ask for help if needed.

WELL DONE Take a break and see how others in the room have done their programs, can you see if others are doing it differently?

REMEMBER – SOMETIMES THERE CAN BE A FEW DIFFERENT WAYS TO SOLVE A PROBLEM.

Today we covered:

How to use delays in a program using “wait” command.

Why delays don't always do what we would like.

How much **thought has to go into writing** a program using delays.

How **a notebook and a pen** is great for helping you figure things out.

How **messages** can be used to control a program.

Questions?

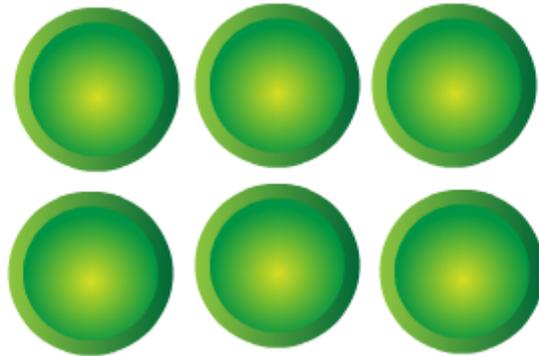
Welcome back!

Now that you have mastered the basics, can you add in more LED sprites?

Add 4 more to give you a total of 6.

Can you get them to flash in order, and 1 second apart?

Remember, ask for help if you are stuck, can others at your table help?

**Excellent!**

Now that you have your 6 LEDs, all working the way you want them to...**they are, aren't they?**
We can talk some more about “wait”.

What would you need to do if you wanted to speed up, or slow down the time between flashes?



Answer: You would need to go to every place, in every sprite where you have a number in the “wait” command and change it to a smaller or larger number.

You only have 6 sprites, think how many changes you would need to make if you had lots more!

So, can any of you think of a better way to control the “wait” value?

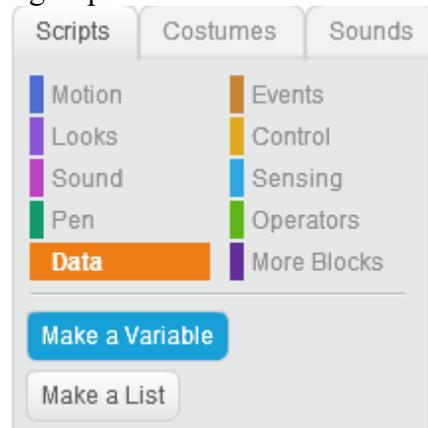
Step Six: Time for a change.

Variables:

Question: What is a variable?

Answer: A variable is a value in a program that can / might change.

In Scratch variables are in the “Data” group.



Make a “**Variable**”, call it “**time**”. (Make sure “**for all sprites**” is on.)



Now you see a new menu has appeared – just for your variable “**time**”.

Go to your sprites and drag in the “**time**” block into the white gap in the “wait” lines.



Now that you have told the program to wait “**time**” seconds, you need to give a value to “**time**” that the program can use.

You can add this to Sprite 1; add it immediately after the “**Green Flag**” event.



Now, try the program.

Again, if it doesn't work try to figure out why and ask for help if you get stuck.

Did it work?

All that is happening now is that the program now uses the Variable called “**time**” instead of a fixed value of 1 second. Change to number to see what happens.



Step Seven: More changes?**Changing variables:**

Now that that is working how can you change the value of “**time**” without stopping and restarting the program?

Remember in the “Data” group when we made the “**time**” variable there was a new range of commands appeared?

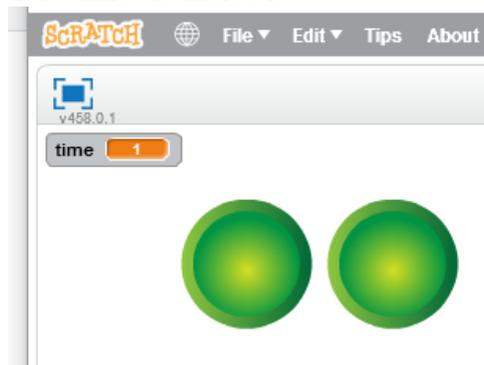


We now have the block to change the value in “time”.

Add the following code to your program for sprite 1.



You should now also see that a block has appeared in the top left corner of the stage. The “Starting Value” for “time” is inside it.



Now, Run your program and use the up & down arrow keys. This should make the LEDs flash faster or slower.

Does it work?

Well Done!

Now we know about **messages**, **variables**, **controlling** variables and **sequences** – more than enough to keep you busy.

In the next session we are going to take all of this information and use it to work out how to write the Arduino program.

NOTES: