# Goal 14 - Life under water – oil spill cleaner-upper (no making)

Getting started

Oil spills do untold damage to eco-systems.

A new material can absorb up to 90 times its own weight in spilled oil and then be squeezed out like a sponge and reused, raising hopes for easier clean-up of oil spill sites.

<https://www.newscientist.com/article/2123391-sponge-can-soak-up-and-release-spilled-oil-hundreds-of-times/>

A group of marine scientists have asked you to develop an algorithm that could be used on a boat drone to drag around a sheet of this smart material to clean up an oil spill.

Success criteria

* Build a prototype oil spill cleaner upper boat drone that starts with a button press
* The prototype should demonstrate an algorithm that moves a sprite autonomously over an area (5x5 LEDs)
* The prototype should be able to take in larger areas to cover larger spills

Breaking down the problem

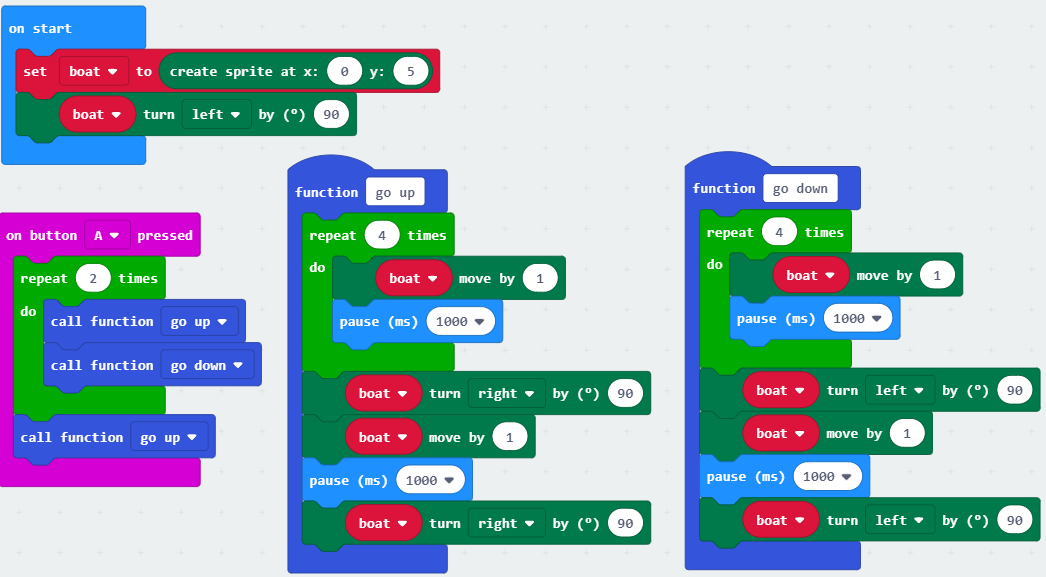
The input and output for this problem are simple as the drone boat should start with a button press and should follow a pre-programmed path. Creating the algorithm for the movement is the tricky part and will require some thought.

Input process output (IPO)

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Button press | Algorithm to control the movement of the boat drone in a path to clean up oil:    Increase or decrease the size of the grid | Show the sprite on the screen completing the algorithm  Larger LED screens could display the larger grid, smaller grids can be represented on the micro:bit |

|  |  |
| --- | --- |
|  | First of we all, we need to create a variable called ‘boat’ that will act as a container for our sprite.  By default, the sprite is facing right. We need to turn it 90 degrees to the left so that it is facing up. You cannot see the direction of the sprite on the LED screen so you will have to use your imagination! |
|  | We need to assign the sprite X and Y co-ordinates. The micro:bit’s screen is made up of 5x5 LEDs.  Each LED has a co-ordinate as shown here.  You can use these co-ordinates to control the sprite’s movements. |
|  | We are going to split the movement into two separate functions. Once called ‘go up’ and another called ‘go down’.  A function is like a mini programme within your programme that can be run by ‘calling’ it when needed.  In the ‘go up’ function we start with a loop that repeats for time. The loop moves the ‘boat’ sprite one LED forward, waits for a second and then loops though again. This will move the sprite from co-ordinate 0,4 to 0,0.  Next, the sprite is turned 90 degrees right and then moves one LED forward. The sprite is now facing right and is at co-ordinate 1,0.  We add a pause here to make the animation of the sprites easy to watch.  Next, we make the sprite face down, ready for the next function. |
|  | We now repeat the algorithm but in the opposite order so it goes down and then across. The ‘go down’ function has already left the sprite facing down so can start by moving the sprite down 4 times by using a ‘repeat 4 times’ loop.  The sprite should now be at co-ordinate 1,0.  Notice again how we add pauses to make the animation slower so we can watch it more easily.  Next, we turn the sprite 90 degrees to left and move it across to co-ordinate 2,0.  Finally, we turn the sprite to face upwards ready to start again. |
|  | We now need to sew the two functions together so we can go from co-ordinate 0,4 all the way to 4,0. The ‘go down’ and ‘go up’ functions only do half the job. We put all this in a ‘on button A pressed’ block so we can trigger the full programme from a button A.  We need to use a loop that ‘calls’ the two functions in the right order and go through them twice. This will take the sprite to co-ordinate 4,4.  Lastly, we call the ‘go up’ function one more time to take us to co-ordinate 4,0. |

The full program



Test time!

Testing this program is important as you need to ensure the functions act as you expect, and that the sprite behaves as you have planned. This is particularly important when adding in the functionality that allows the user to change the size of the grid. Think about this carefully!

Stretch tasks

* Now you have completed the algorithm prototype , now use the ‘with build’ worksheet and create the boat drone using servos. This will involve some additional peripherals and materials. You will have to re-write the algorithm to control the servos rather than a sprite, but the steps are the same.

Final thoughts

Combining smart material and autonomous drones is just one way that technology can help contribute to the global goals. Can you think of other ways that technology can help?

# Goal 14 - Life under water – oil spill cleaner-upper (with making)

Getting started

Oil spills do untold damage to eco-systems.

A new material can absorb up to 90 times its own weight in spilled oil and then be squeezed out like a sponge and reused, raising hopes for easier clean-up of oil spill sites.

<https://www.newscientist.com/article/2123391-sponge-can-soak-up-and-release-spilled-oil-hundreds-of-times/>

A group of marine scientists have asked you to develop an algorithm that could be used on a boat drone to drag around a sheet of this smart material to clean up an oil spill.

Success criteria

* Build a floating oil spill cleaner upper boat drone that starts with a button press
* The product should be able to autonomously navigate over an area
* The product should be made to clean up an oil spill by dragging a ‘smart material’

Breaking down the problem

The input and output for this problem are simple as the drone boat should start with a button press and should follow a pre-programmed path. Creating the algorithm for the movement is the tricky part and will require some thought.

Input process output (IPO)

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| Button press | Algorithm to control the movement of the boat drone in a path to clean up oil: | Servo motor control |

Pro-tip

Don’t worry about distances at this point. Oil spills can be small or large, and the product just needs to be able autonomously clean an area. Think about how the size of the area can be changed.

We won’t have access to any smart material but we can simulate it using a normal sponge.

Building the product

For this project we need to build a simple boat. You can use anything that is waterproof.

|  |  |
| --- | --- |
|  | **Kit required:**   * A micro:bit * Header wires * Battery pack * Boat building materials * A foam sponge * A mini screwdriver * A servo driver board   There are many types of servo controller boards for micro:bit, in this example an ‘automation bit’ was used. |
| A close up of a camera  Description automatically generatedA circuit board  Description automatically generated | Here you can see how the servo motors are wired to the servo controller and micro:bit.  The + cable from **both** the servos need to go into the 3v opening on the servo control board.  The – cable needs to go into output 1 and 2 respectively.  Pay attention to which side you put them on. In this image:  Output 1 = Right  Output 2 = Left  You may have a third cable for the servo which is the ground (GND), attach this to the GND terminal on the board if you have this. |
| A picture containing weapon, gun  Description automatically generated | Here you can see the two + cables from the servos going into the same 3v terminal.  Other servo control boards may have more than one 3v terminal and so these should be separated if you can. |

Example code

|  |  |
| --- | --- |
| In this example some custom extension blocks were used. To find these blocks you need to click the ‘Extensions’ | |
|  | tab and then search for the control board that you are using. In this example an ‘automation bit’ was used, but other servo control boards will also work. |

To start with we will create the first part of the algorithm that turns the servos on when a button is pressed:

|  |  |
| --- | --- |
|  | This simply sets each of the outputs to 1 (on) once button A is pressed. |
|  | To get the servos to stay on for different amounts of time we need to create a timer. In this example we use a ‘forever’ block to change the value of a variable by -1 every second. We can then count down from any number we like by setting the variable to however many seconds we need and then doing something when it reaches 0. |
|  | In this example we set the button B to set the outputs to 0 so that we can use it to turn of the robot drone.  We could however use the A/B reset button to do this, but we may need this button for something else later. |
|  | In this example we use the button A as the trigger to set the timer variable to 10.  We then use a **while loop** to check if the timer variable is greater than 0; if it is, it turns both outputs on. This would give us 10 seconds of forward motion for the boat.  Once the timer variable reaches 0 we set the timer variable to 2 (to give us 2 seconds) and then use another **while loop** to check if the timer variable is greater than 0. If it is, then it sets only output 1 to 1 (on) so the drone boat will turn to the right.  You will need to experiment with how many seconds it takes to turn 90 degrees. |
|  | This set of blocks shows the first few steps of the algorithm that automates the oil spill clean-up.  You can use this as a starting point and adapt it to meet the success criteria. |

Test time!

**BEWARE**! **Do not test this in water, micro:bits and peripherals are not waterproof!** Electricity and water do not mix well, and you can permanently damage the electronics by getting them wet.

You can test the boat drone by timing how long and which servos are turned on to see if your programme behaves as you would expect.

Stretch tasks

* Adapt the program so that the navigation is done using the micro:bit’s compass so that is turns precisely 90 degree and can stay on course more accurately
* Add a moisture sensor so that the boat drone only starts cleaning when in the water
* In large oil spills, many drones would be used at one. Adapt your programme so that the boat drones don’t collide with each other (you could the radio blocks for this)
* Adapt your program so that the ‘smart material’ is dragged by two drone boats and the smart material is in a long thin sheet. This helps the oil to be squeezed out of it more easily
* Adapt your program so that the boat drones can return to a ‘base’ where the smart material can be wrung out and then re-used
* Adapt your program so that you can control the boat drones direction of travel remotely using another micro:bit

Final thoughts

Combining smart material and autonomous drones is just one way that technology can help contribute to the global goals. Can you think of other ways that technology can help?