Servos

Making a Sound Meter

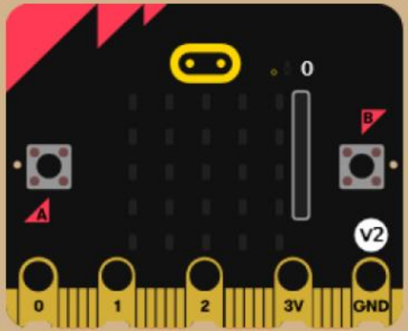
This week you be creating your own sound reactive program using a servo and its extension.

Whether programming in blocks, JavaScript or Python, please complete the bronze requirements before starting the silver, and complete the silver before starting the gold.

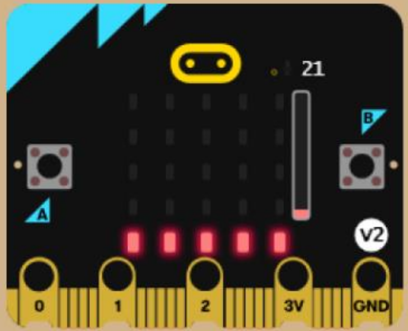
BRONZE Challenge:

- Create a start-up sequence for your program.
- Write a program that plays an alarm when the sound level is over 200.
- Have the Micro:Bit display the sound level value on its screen.
- Notice how the show string block slows down our alarm sound. Let us resolve this by replacing the number with the images and conditions shown on the next slide

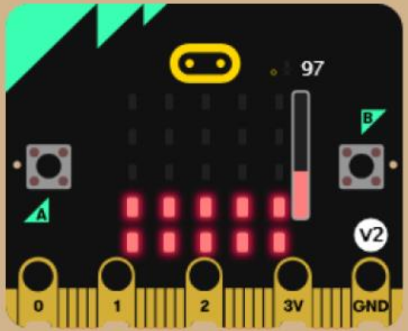
Sound level is 0.



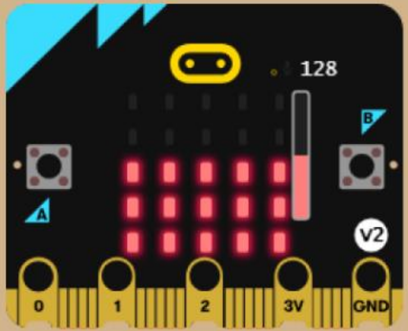
Sound level is over 0
and
less than or equal to 51.



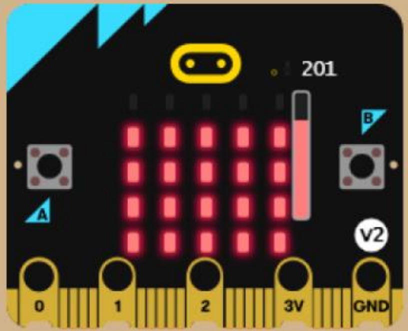
Sound level is over 51
and
less than or equal to 102.



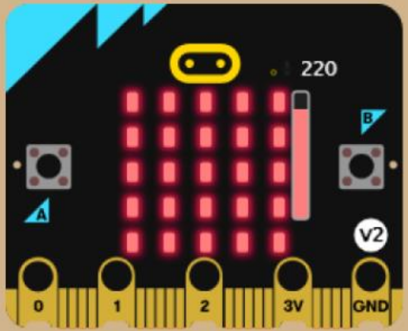
Sound level is over 102
and
less than or equal to 153.



Sound level is over 153
and
less than or equal to 204.



Sound level is over 204.



SILVER Challenge:

- Add a 180° positional servo to pin 1. Program the servo to rest at 0° whilst the sound level is less than or equal to 102. Otherwise, it needs to move to 180°
- To reduce stress on the servo, have it move only 10° every 300 milliseconds?
- If not done so already, create functions for the servo movements (one for increasing the angle, the other decreasing).

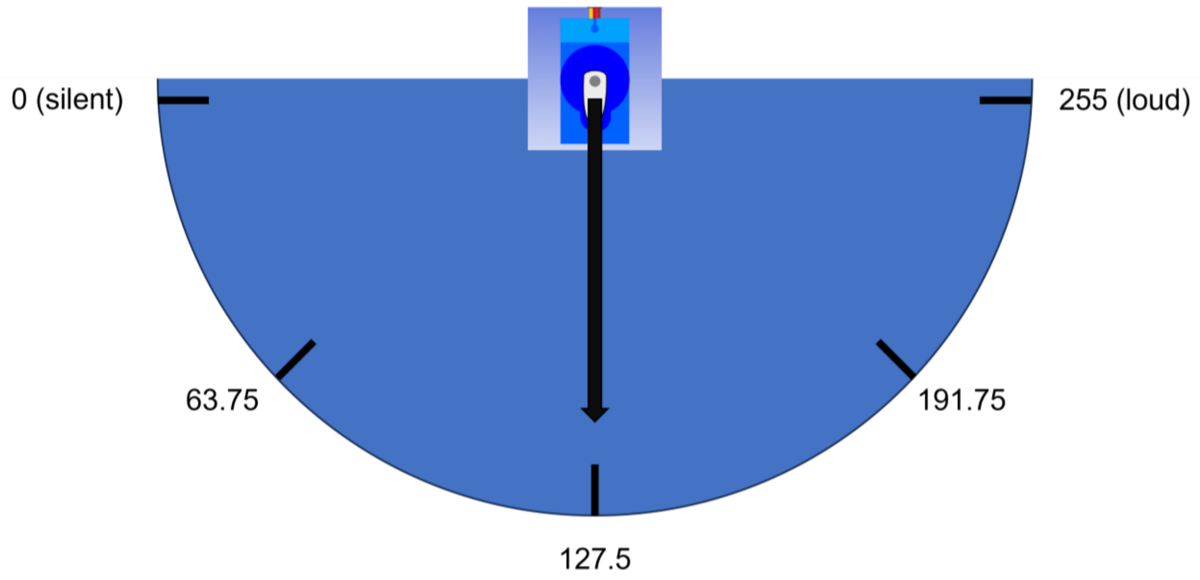
STOP!

If you are doing these challenges as part of our Wednesday Zoom session, let a member of staff know that you have completed the work so far.

Do not continue until we confirm that we have saved a copy of your program as the next stage involves re-writing it.

GOLD Challenge:

- Instead of having a single value determining the movement of the servo, let us look at having the servo act like the arm on a sound meter (as shown below), traveling to the correct angle for the current sound level.

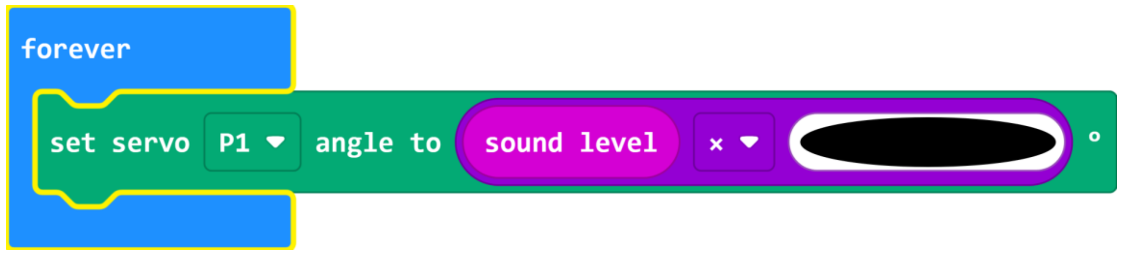


Important:

We have a range of 0° to 180° on the servo but the sound level has a range of 0 to 255. There are two ways to approach this:

1: Do the maths ourselves:

Work out how far the meter needs to move for one unit of sound (180 ÷ 255). Use this value with a Maths block inside a 'set servo' block



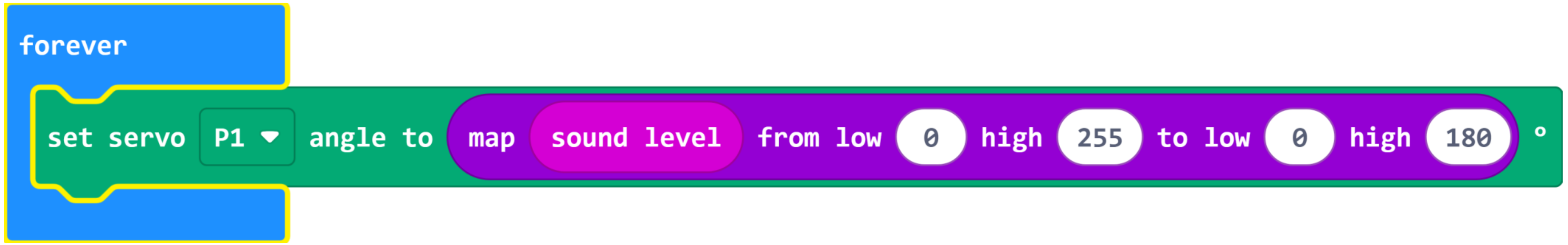
```
forever
  set servo P1 ▾ angle to ( sound level × ▾ [redacted] ) °
```

1872 PRIFYSGOL ABERYSTWYTH UNIVERSITY

Important:

2: Use a mapping block:

The map block allows us to enter the range we're measuring (sound level) which is 0 to 255 and convert it to the servo's range of 0 to 180. The block then does all the calculations for us.



```
forever
    set servo P1 ▾ angle to map sound level from low 0 high 255 to low 0 high 180 °
```

Extension Challenge:

Can you have a second servo acting as a signal strength meter for a radio channel of your choice.

Thank You

1872 PRIFYSGOL ABERYSTWYTH UNIVERSITY