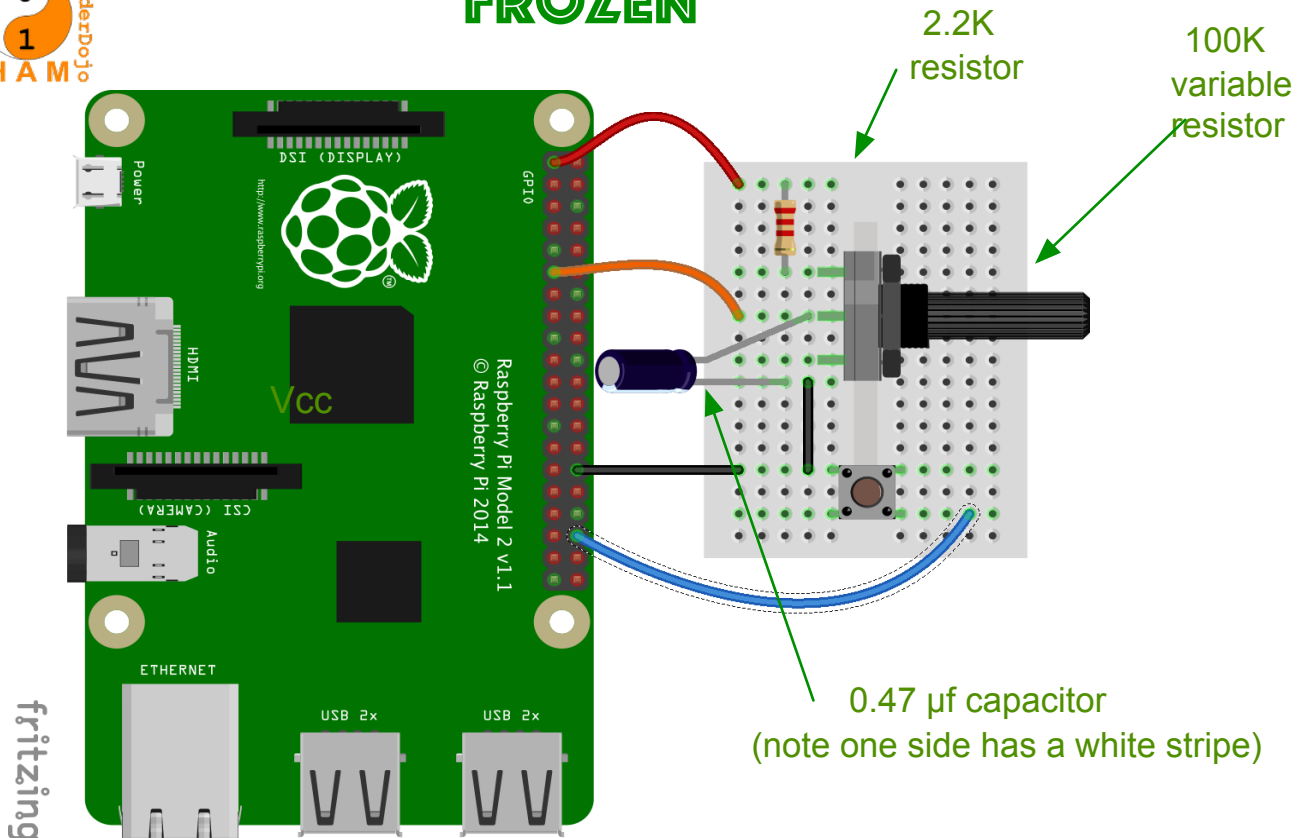
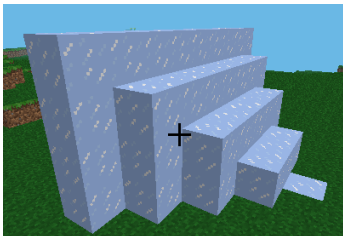


FROZEN



We want to be able to freeze Minecraft blocks, but have control over how far our magic powers can reach.

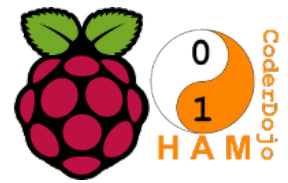


Let's have the coldness spreading out from us in this shape, but only freezing blocks that are NOT air.



We can adjust the variable resistor to set the range of our freezing. Just like an LDR, it is an analogue component so we use a capacitor to make a timing circuit.

We're pretending the variable resistor is an LDR (LightSensor) and hacking the gpiozero class so that it does what we want.



```
1 from gpiozero import LightSensor, Button
2 import mcpi.minecraft as minecraft
3 import mcpi.block as block
4
5 button = Button(16) # Our button is on pin 16
6 # We're going to pretend our variable resistor is an LDR
7 pot = LightSensor(17, charge_time_limit=0.02)
8 max_spread = 10 # Adjust this value to set maximum range
9
10 mc = minecraft.Minecraft.create() # Connect to Minecraft
11
12 # Function to work our Freezing ray
13 # We will find every block in all three directions, up to the max range we've setBlock
14 # check that it is not AIR and then turn it to ICE.
15 # This will use 3 nested loops, one for each direction (x,y,z)
16
17 def freezeray(spread):
18     pos = mc.player.getTilePos() # get current player's position
19
20     for z_spread in range(0, spread): # First loop: Z direction
21         print('Freezing distance = ' + str(z_spread))
22
23         for x_spread in range(1- (z_spread+1), z_spread+1): # 2nd loop: X direction
24
25             for y_spread in range(-1, z_spread): # Final loop: Y direction
26
27                 target_position = (pos.x + x_spread, pos.y + y_spread, pos.z + z_spread)
28                 target_block = mc.getBlock(target_position) # get the block type
29
30                 if target_block != block.AIR.id: # if block is not AIR
31                     mc.setBlock(target_position, block.ICE.id) # turn to ICE
32
33     try:
34         while True:
35             # Read the value of our variable resistor (it will be between 0 and 1)
36             # and multiply by our spread
37             value = int(pot._read() * max_spread)
38             print(value)
39
40             if button.is_pressed: # When the button is pressed
41                 freezeray(value) # Run the freeze ray function
42
43     except KeyboardInterrupt:
44         exit()
45
```

Can you modify the code to increase the maximum range?

Can you make a lava-ray?