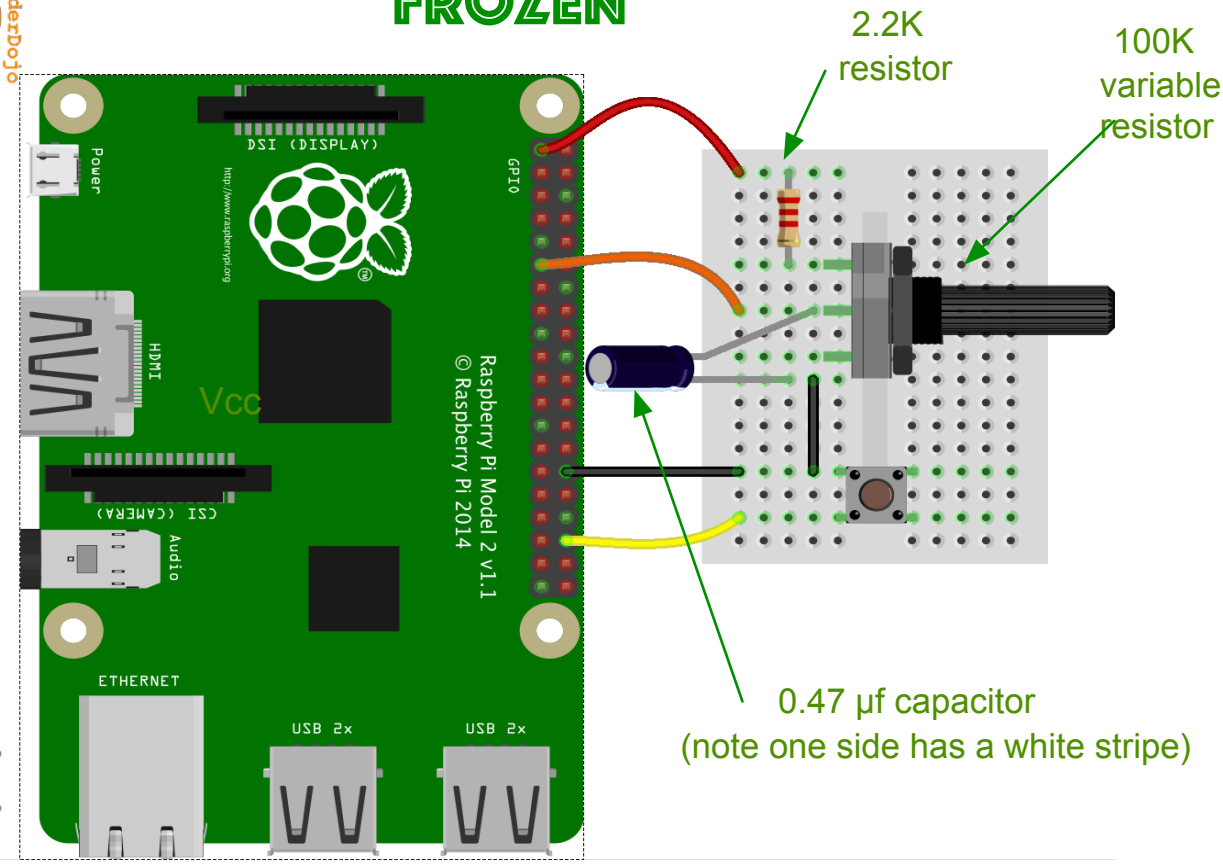
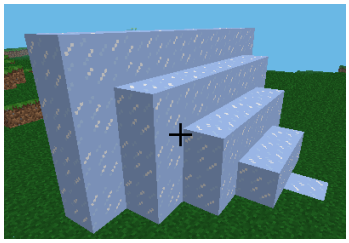
FROZEN



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.

```
from gpiozero import LightSensor, Button
import mcpi.minecraft as minecraft
import mcpi.block as block

button = Button(16) # Our button is on pin 16
# We're going to pretend our variable resistor is an LDR
pot = LightSensor(17, charge_time_limit=0.02)
max_spread = 10 # Adjust this value to set maximum range

mc = minecraft.Minecraft.create() # Connect to Minecraft

# Function to work our Freezing ray
# We will find every block in all three directions, up to the max range we've setBlock
# check that it is not AIR and then turn it to ICE.
# This will use 3 nested loops, one for each direction (x,y,z)

def freezeray(spread):
    pos = mc.player.getTilePos() # get current player's position

    for z_spread in range(0, spread): # First loop: Z direction
        print('Freezing distance = ' + str(z_spread))

        for x_spread in range (1- (z_spread+1), z_spread+1): # 2nd loop: X direction

            for y_spread in range(-1, z_spread): # Final loop: Y direction

                target_position = (pos.x + x_spread, pos.y + y_spread, pos.z + z_spread)
                target_block = mc.getBlock(target_position) # get the block type

                if target_block != block.AIR.id: # if block is not  AIR
                    mc.setBlock(target_position, block.ICE.id) # turn to ICE

try:
    while True:
        # Read the value of our variable resistor (it will be between 0 and 1)
        # and multiply by our spread
        value = int(pot._read() * max_spread)
        print(value)

        if button.is_pressed: # When the button is pressed
            freezeray(value) # Run the freeze ray function

except KeyboardInterrupt:
    exit()
```

Can you modify the code to increase the maximum range?

Can you make a lava-ray?