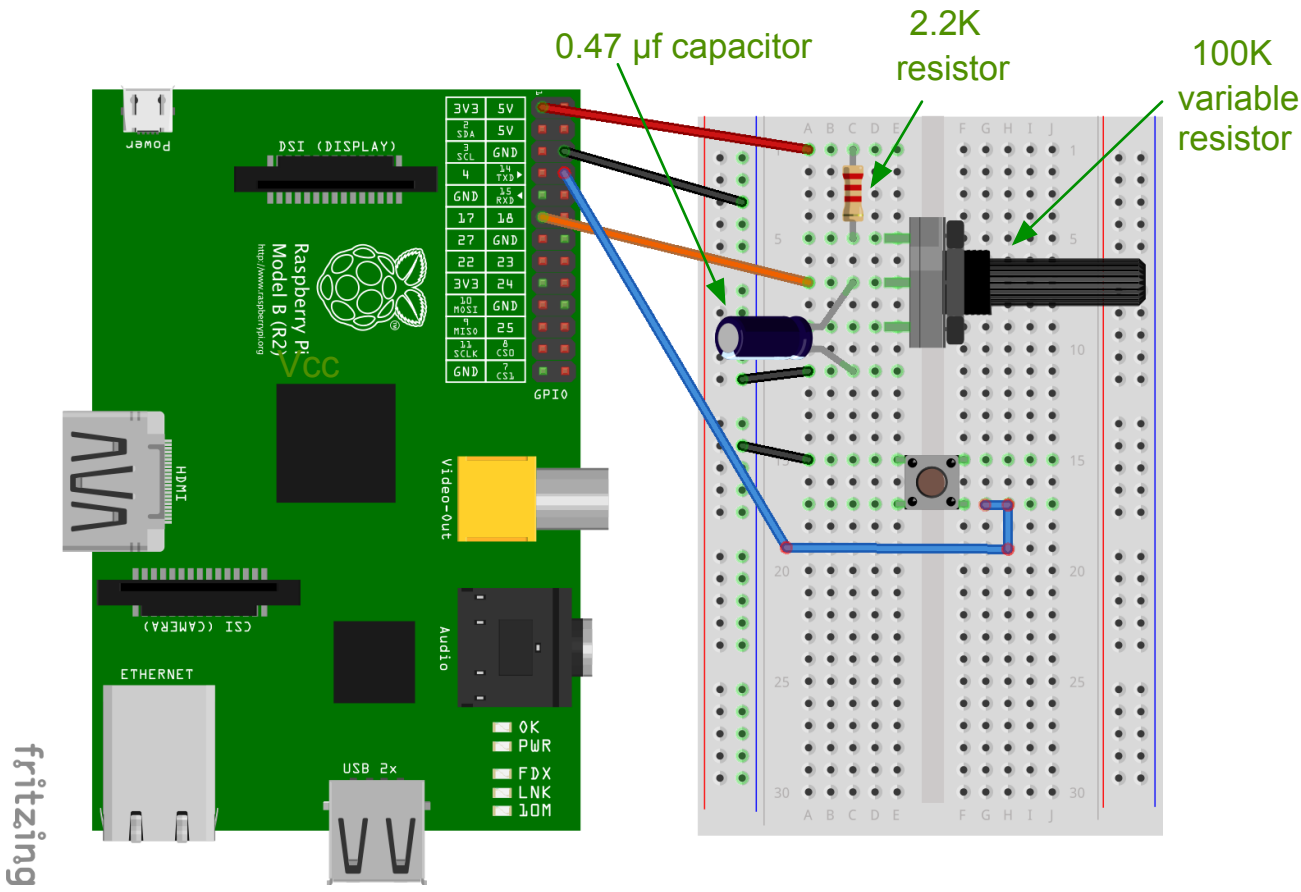
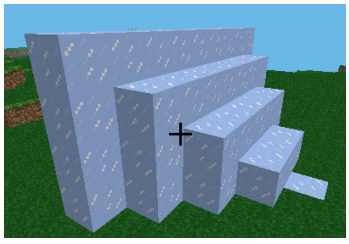
FROZEN



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.

```
import mcpi.minecraft as minecraft
import mcpi.block as block
import time
import RPi.GPIO as GPIO

BUTTON=8
GPIO.setmode(GPIO.BOARD)
GPIO.setup(BUTTON,GPIO.IN, pull_up_down=GPIO.PUD_UP)
PIN =11
ConvFactor = 50  # you may need to change this depending on the values you get

mc = minecraft.Minecraft.create()

# create a function to measure the charging time

def PotRead():
    # set the reading to 0 each time
    reading = 0

    # set the GPIO pin to be an output and turn it off
    # this discharges the capacitor
    GPIO.setup(PIN,GPIO.OUT)
    GPIO.output(PIN, GPIO.LOW)

    time.sleep(0.1)

    #now make the pin an input
    GPIO.setup(PIN, GPIO.IN)

    # start counting and keep counting until the the pin reads high
    # which means the capacitor is charged again
    while (GPIO.input(PIN) == GPIO.LOW):
        reading +=1

    return reading/ConvFactor

def freezeRay(spread):
    pos = mc.player.getTilePos()
    #spread = 15
    for j in range(0,spread):                # z direction (forward)
        print 'Freezing level ' + str(j)
        for i in range(1-(j+1),j+1):     # x direction (left and right)
            for k in range(-1,j):    # y direction (up)
                target_block = mc.getBlock(pos.x+i,pos.y+k,pos.z+j)
                if target_block != block.AIR.id:
                        mc.setBlock(pos.x+i, pos.y+k, pos.z+j,block.ICE.id)
# run the function forever
try:
    while True:
        value = PotRead()
        print 'setting is ' + str(value)
        if GPIO.input(BUTTON) == False:
            print 'Getting chilly...'
            freezeRay(value)

# end the program if we type Ctrl + c
except KeyboardInterrupt:
    print 'exiting'

# tidy up the GPIO as the program ends
finally:
    GPIO.cleanup()
```

Use the dotted lines to help you get each block of code aligned correctly.

This part is the same code used in the LDR project