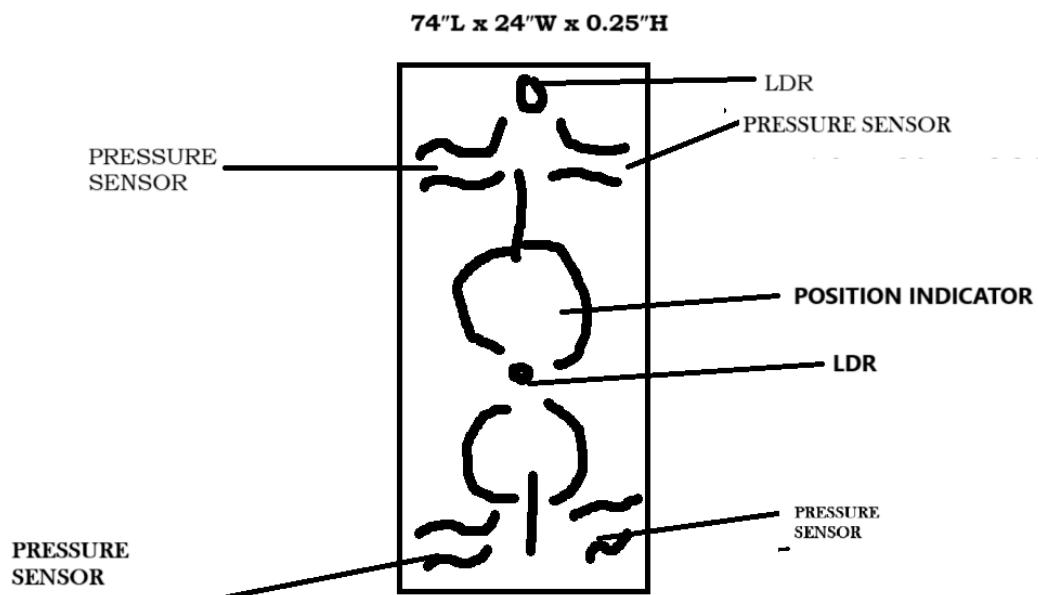


Balaji's hardware R&D

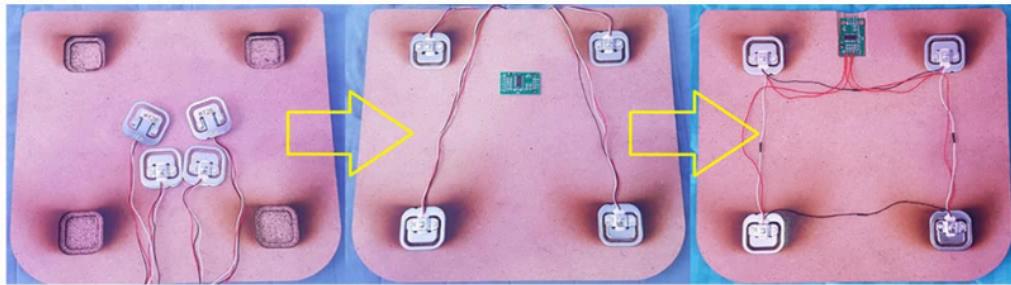
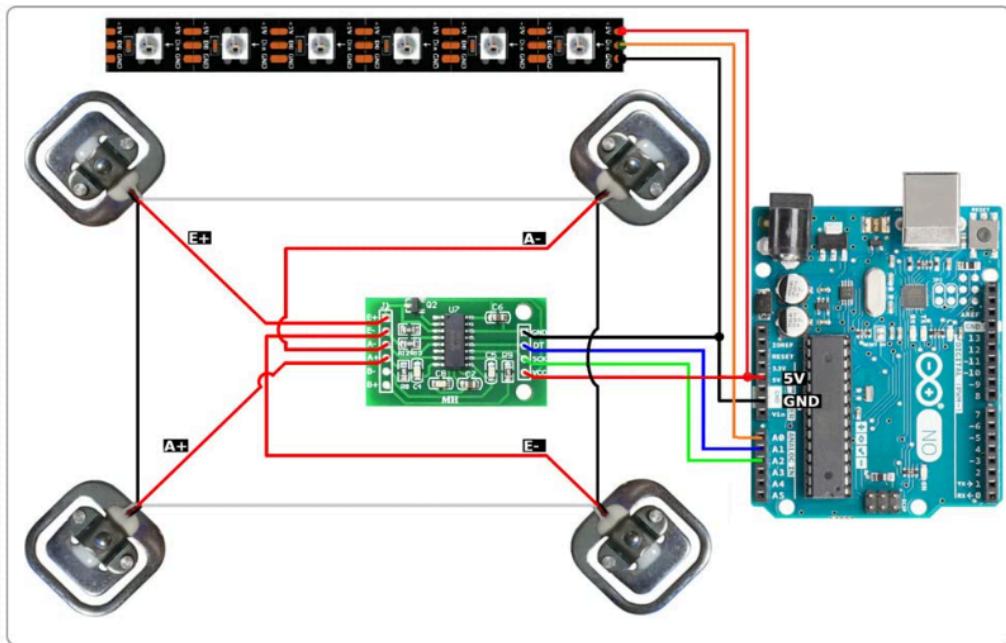
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>

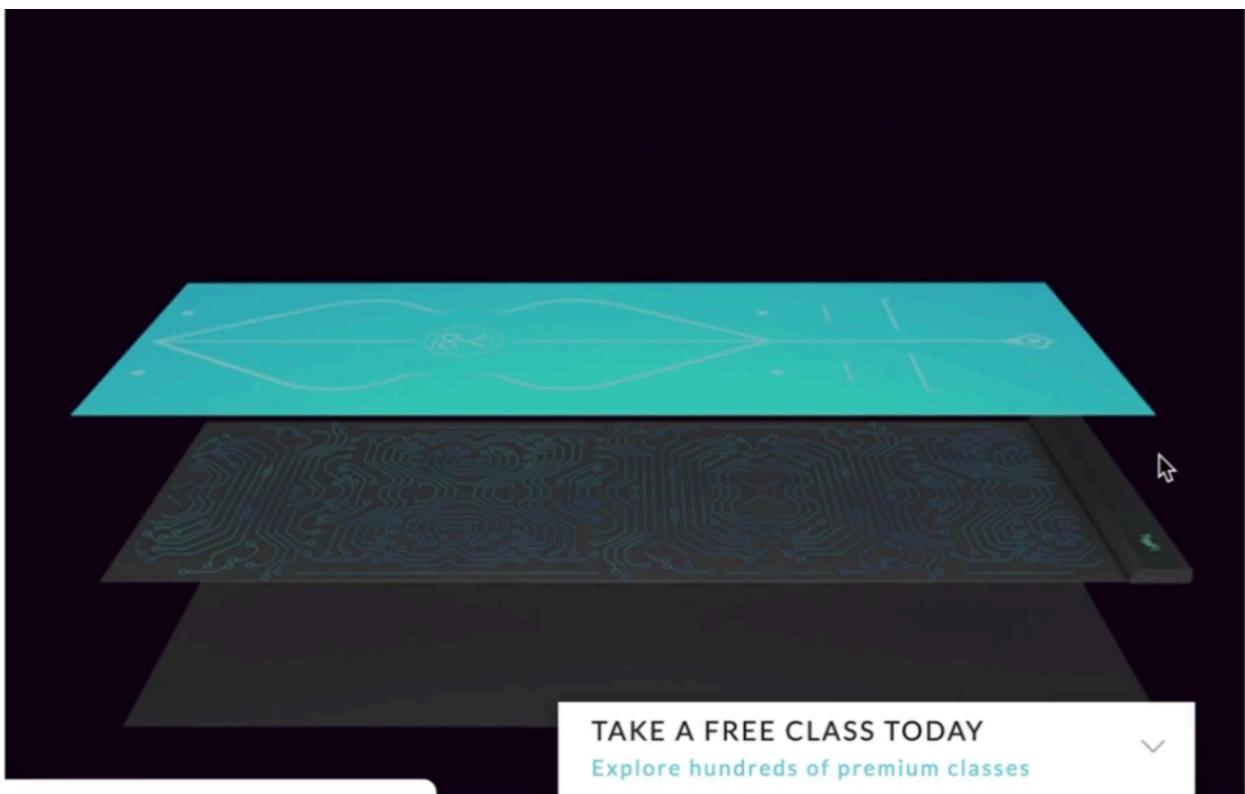
List of Hardware

Raspberry pi 4b * 1 and html cable for pi 4-B , sdcard32(GB), power supply
HX711 * 4
LDR Module * 5
connecting wires
microfiber plain glass
weight less cardboard sheet
plain yoga mat *2
position indicator print (for position indication)

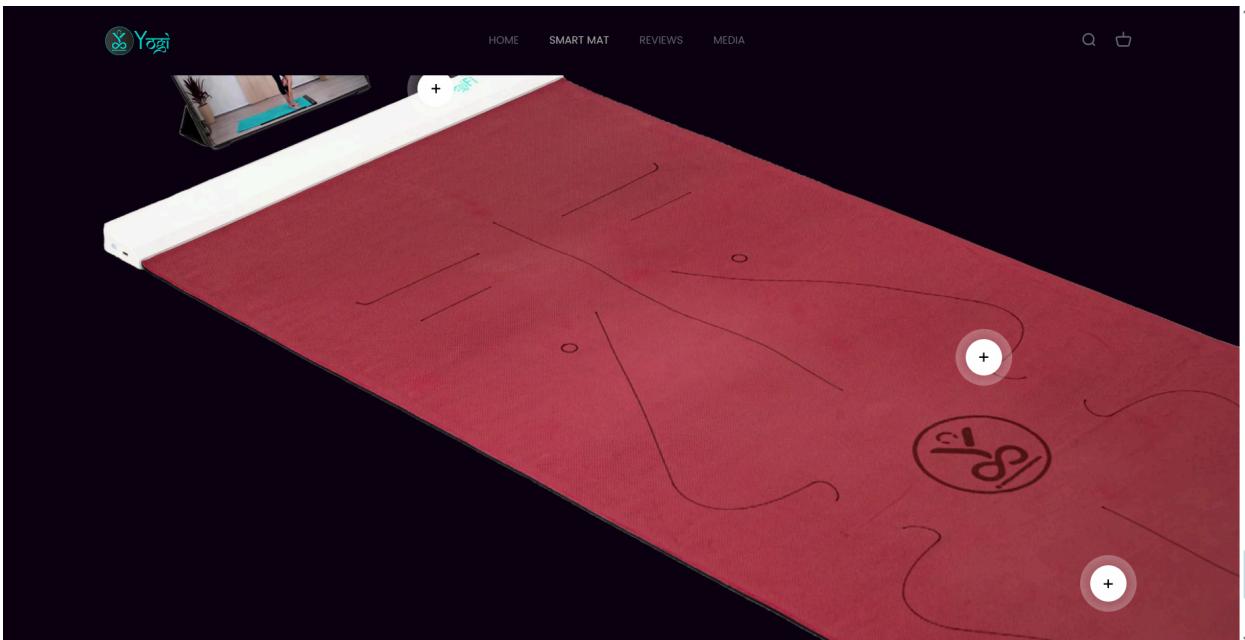


Architecture of yoga mat or design layout





existing Method



existing method for reference

Technical Specifications:

- Compatible With Phones and tablets having iOS v14.3+
- Plug Type: Flat Pin
- Voltage: 5V
- Cord Length: 8 ft
- Battery Run Time: 7 Hours
- Battery Charge Time: 4 Hours
- Compatible YogiFi App
- Bluetooth Range: 30 ft
- Max. Weight Limit: 350 lb
- For Indoor & Outdoor Use

^^ technical specification for reference from the existing method

The screenshot shows a product page for an LM393 Photosensitive Light-Dependent Control Sensor LDR Module. The page includes a navigation bar with links to Home, Shop, Forum, Bulk Enquiry, New Arrivals, ATL Kits Enquiry, Blogs, Careers, and Sell on Robu. The main content area shows a blue printed circuit board with a blue potentiometer and several wires. The product title is "LM393 Photosensitive Light-Dependent Control Sensor LDR Module". It has a 4-star rating from 7 reviews. The item is listed as "In stock". The SKU is 44118. The product description lists five features: 1. LDR module 4 PIN, 2. Operating voltage 3.3V-5V, 3. Signal output indicator light, 4. LDR module 4 PIN, and 5. Able to detect ambient brightness and light intensity Adjustable sensitivity (via blue digital potentiometer adjustment). The price is ₹ 31.00 (Incl. GST).

Product Details:

- Materials Thermoplastic Elastomer (TPE)
- Measurements 74”L x 24”W x 0.25”H
- Weight 5.3lb
- Thickness:6mm

3v3 Power	1	2	5v Power
GPIO 2 (I2C1 SDA)	3	4	5v Power
GPIO 3 (I2C1 SCL)	5	6	Ground
GPIO 4 (GPCLK0)	7	8	GPIO 14 (UART TX)
Ground	9	10	GPIO 15 (UART RX)
GPIO 17	11	12	GPIO 18 (PCM CLK)
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3v3 Power	17	18	GPIO 24
GPIO 10 (SPI0 MOSI)	19	20	Ground
GPIO 9 (SPI0 MISO)	21	22	GPIO 25
GPIO 11 (SPI0 SCLK)	23	24	GPIO 8 (SPI0 CE0)
Ground	25	26	GPIO 7 (SPI0 CE1)
GPIO 0 (EEPROM SDA)	27	28	GPIO 1 (EEPROM SCL)
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12 (PWM0)
GPIO 13 (PWM1)	33	34	Ground
GPIO 19 (PCM FS)	35	36	GPIO 16
GPIO 26	37	38	GPIO 20 (PCM DIN)
Ground	39	40	GPIO 21 (PCM DOUT)

*list of materials and price

*

Raspberry pi 4b * 1

HX711 * 4

LDR Module * 5

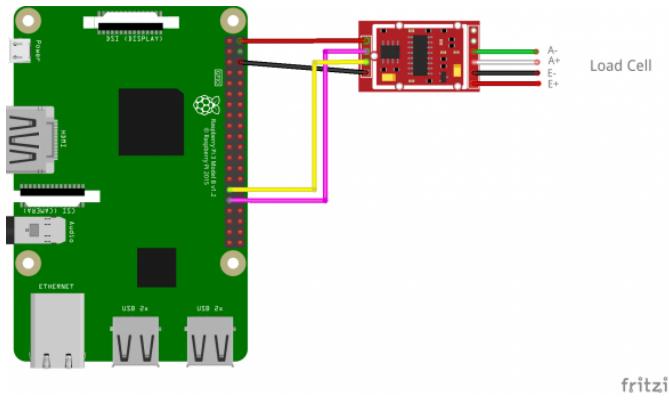
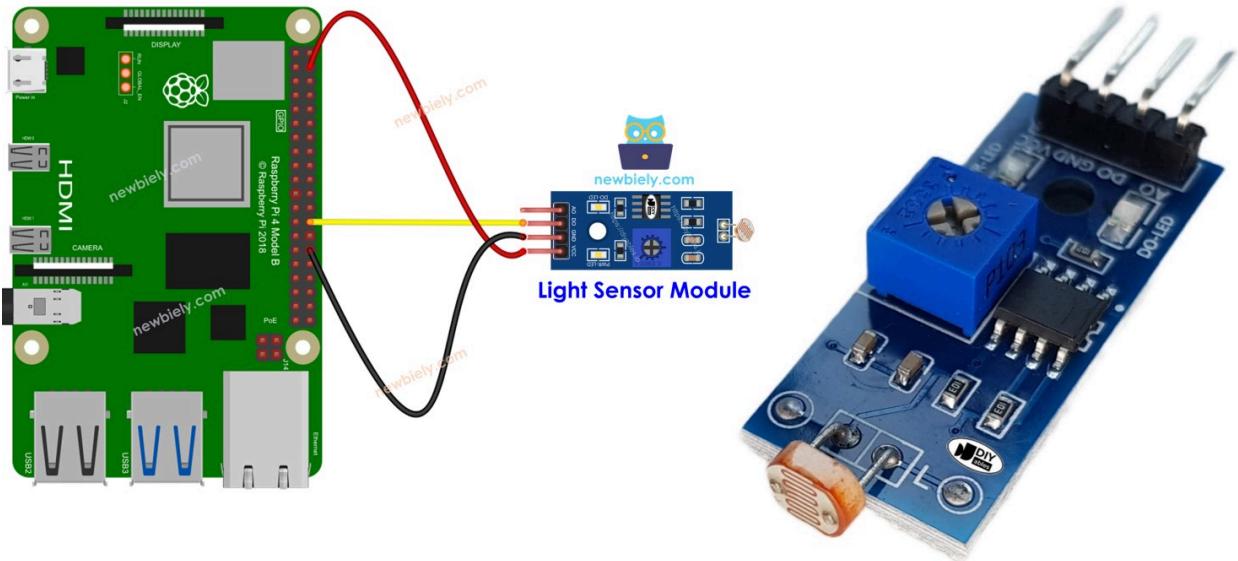
connecting wires

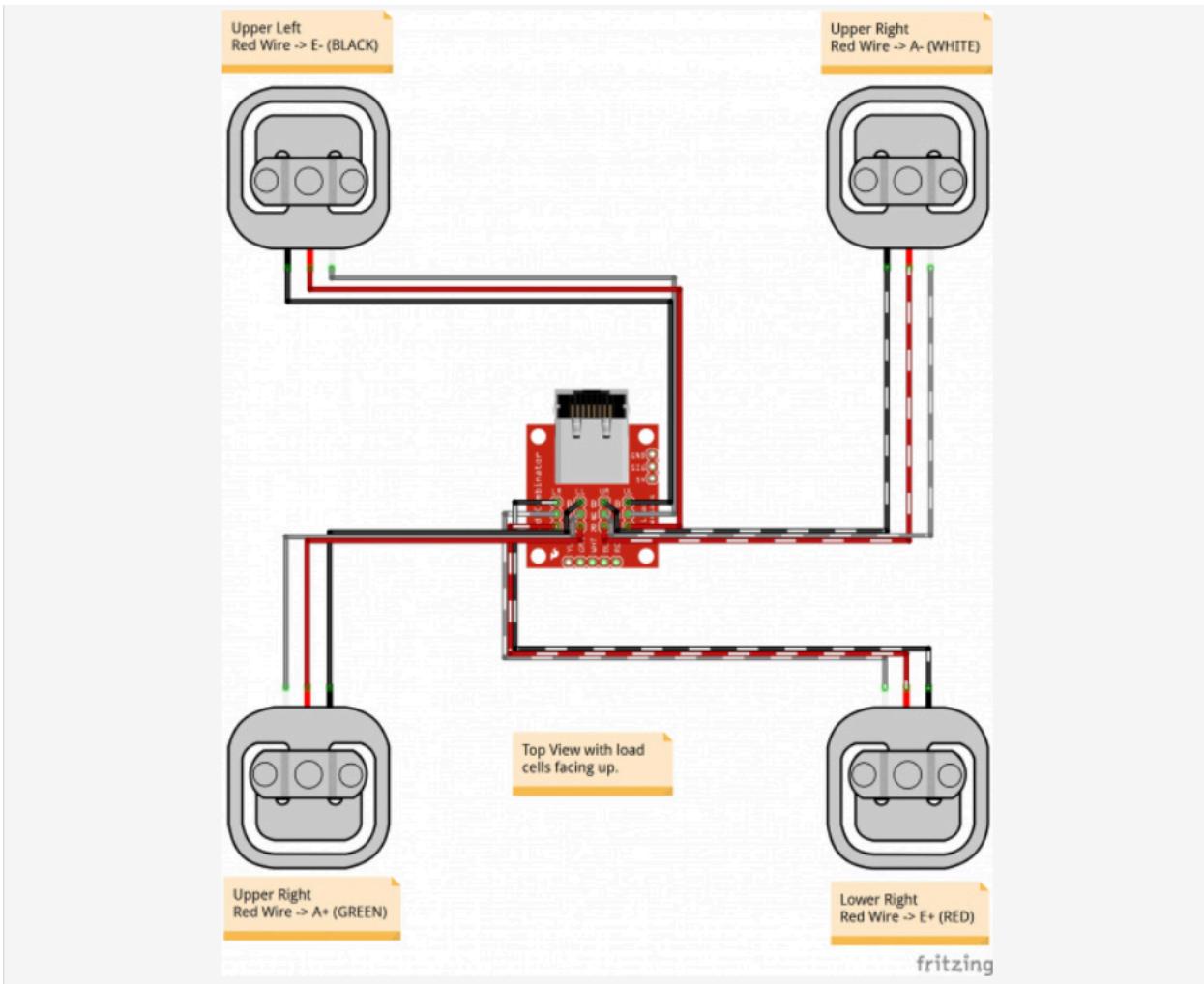
microfiber plain glasss

weight less cardboard sheet

plain yoga mat *2

position indicator print (for position indicator)





Con #1. They're Not All Mobile

Not every intelligent yoga mat is fully transportable. For example, the TERA mat may be more difficult to transport, as it is a full circle meant to act as a carpet when not in use.

Con #2. AI Isn't Perfect

An intelligent yoga mat is still not a person, and likely can't identify when a pose is simply not meeting your unique needs.

Con #3. They're Not Compatible With Your Favorite Apps

Smart yoga mats must be used with their respective companion apps, and can't be used with any other yoga app. So if you already have a favorite (like [YA Classes](#)), it's not a good option.

Love practicing online? Here are [7 Ways to Get the Most Out of Your Online Yoga Practice](#)

Con #4. They Need to Be Charged

Even with long-lasting charges, you still need to remember to charge your mat. You don't want to get to "class" only to remember you haven't charged your mat.

Con #5. They're Pricey

The mats are expensive. Most smart mats these days will run you at least \$400, if not more.

While this may eventually pay off, since you don't have to pay for in-person yoga classes, this is still a price many of us can't afford.

Con #6. They're Not Eco-Friendly

Compared to conventional mats, electronic mats are not environmentally-friendly. Many electronics are difficult to recycle properly, and may contain [harmful materials](#) like mercury or cadmium that require testing to identify.

Cons

Pro #1. Provide Instruction at Home

The mats offer personalized instruction in your own home. This makes it easier to incorporate feedback into your at-home sessions, and provides more privacy.

So if you're nervous about trying out Crow Pose or Headstands in a public studio, smart mats could be a good way to learn new poses, in the privacy of your own home.

Pro #2. They're Portable

Some smart yoga mats, like SmartMat, don't require internet, so you can do your yoga anywhere. Both the SmartMat and the YogiFi mat can be rolled up and transported.

Pro #3. They're Pretty

The TERA mat doubles as home decor! The mat is made from carpet-like material, so you can just leave it in your living room – no need to put it away. Other smart yoga mats have similar designs to more traditional yoga mats, so you're not missing out on the aesthetic component.

Love a pretty yoga mat? [Found: A Work of Art Yoga Mat ... That Actually Works! Meet Metta Mats](#)

Pro #4. They're Long-Lasting

Electronic yoga mats typically work for between four to six hours, which is hopefully more than enough time for you to get your session in.

Pros

21 lines (15 loc) · 418 Bytes

Code Blame ...

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BEAN)

resistorPin = 7

while True:
    GPIO.setup(resistorPin, GPIO.OUT)
    GPIO.output(resistorPin, GPIO.LOW)
    time.sleep(0.1)

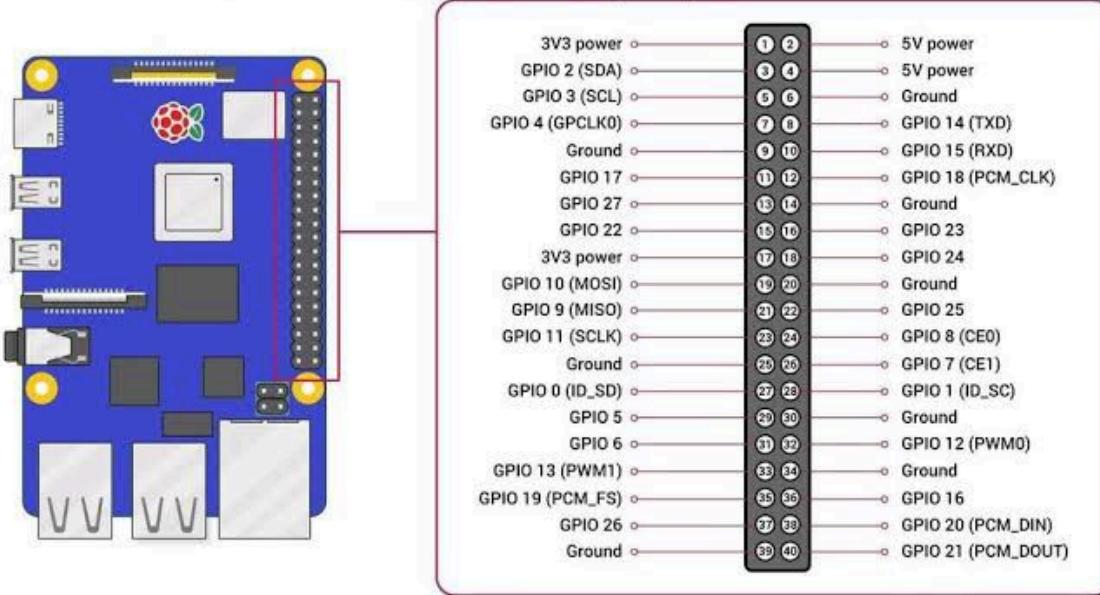
    GPIO.setup(resistorPin, GPIO.IN)
    currentTime = time.time()
    diff = 0

    while(GPIO.input(resistorPin) == GPIO.LOW):
        diff = time.time() - currentTime

    print(diff * 1000)
    time.sleep(1)
```

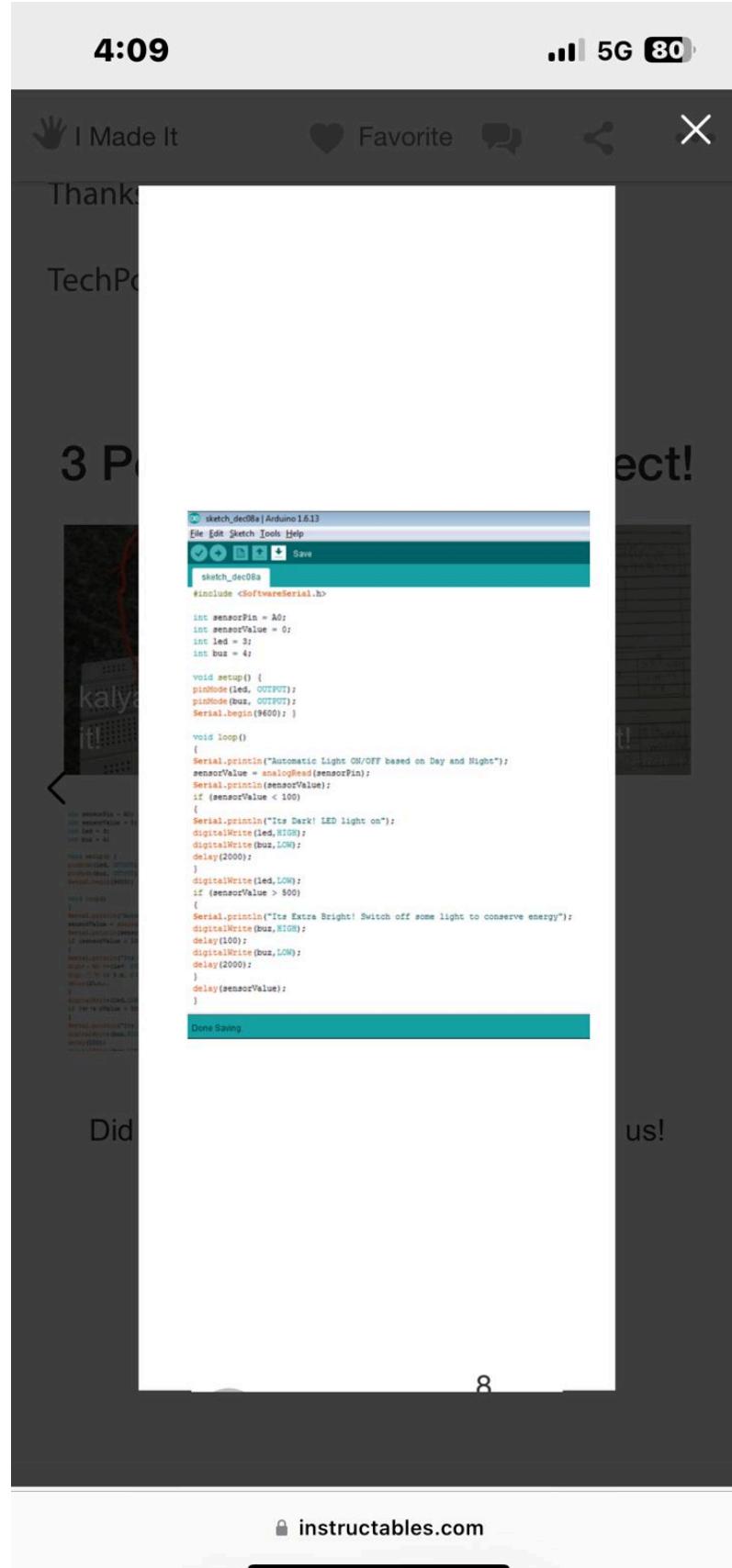
Base code

40 GPIO Pins Description of Raspberry Pi 4

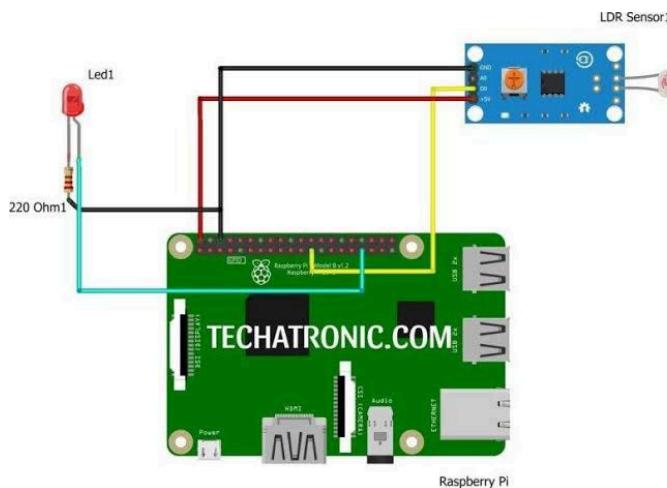


hackatronic.com

Pin Diagram



reference code



Reference diagram for the circuit

How to connect the raspberry pi 4 B to a mobile phone

Reference Link :

<https://bluedot.readthedocs.io/en/latest/pairpiandroid.html#:~:text=On%20your%20Raspberry%20Pi%3A,Click%20Bluetooth%20-%gt;%20Add%20Device>

Range check Code for the LDR Module :

```
from machine import Pin  
import time  
  
# Initialize LDR and LED pins  
ldr = Pin(10, Pin.IN) # LDR connected to GP10  
led = Pin(11, Pin.OUT) # LED connected to GP11
```

```

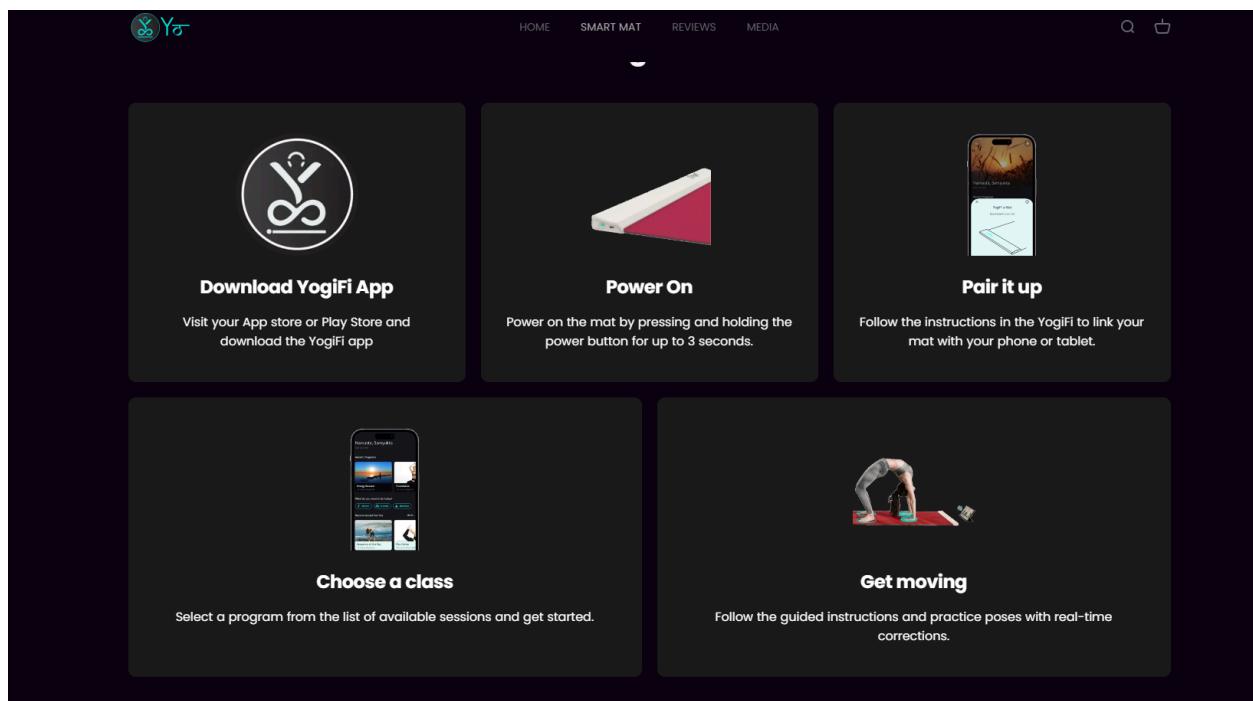
while True:
    # Read the digital input from LDR
    ldr_value = ldr.value()

    # Print the LDR value for our reference
    print("LDR Value:", ldr_value)

    # Control the LED based on the LDR input
    if ldr_value == 0:
        led.on() # Turn on the LED when LDR input is LOW (dark)
    else:
        led.off() # Turn off the LED when LDR input is HIGH (light)

time.sleep(1)2406

```



Flow of App

it should be theme based

related to problem statement

it should impact full, realistic , capable enough to develop the idea , business value , practicality and technical complexity

CV REFERENCE :

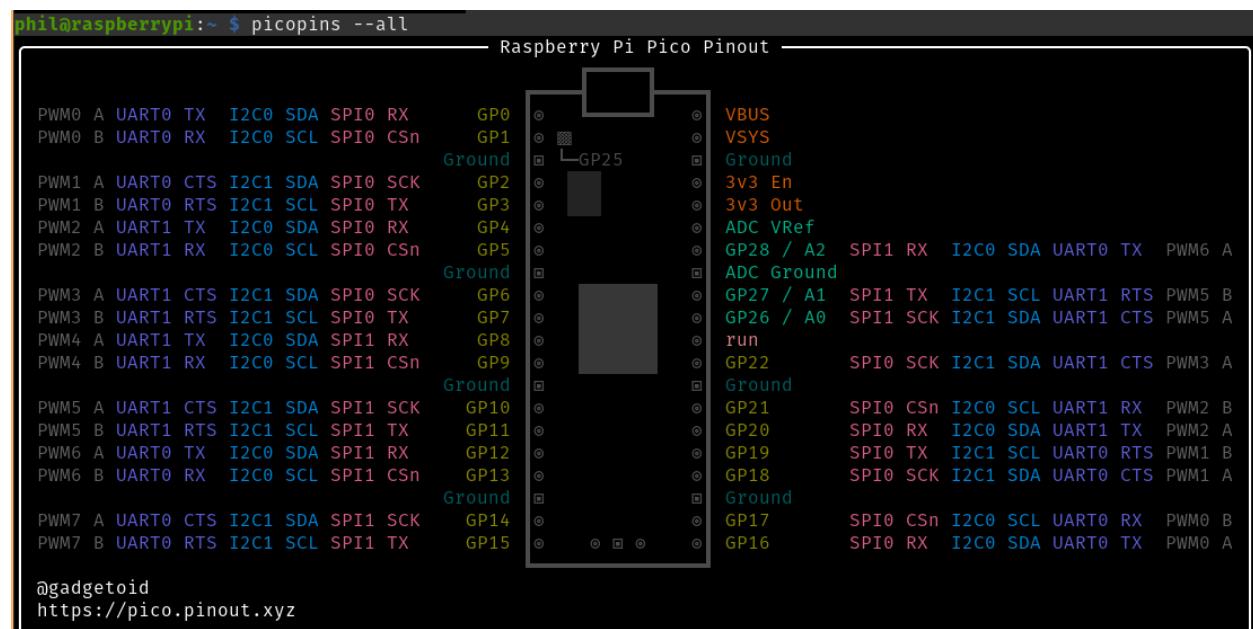
https://youtu.be/fiDaAc7z_kQ?si=eYjFBtH2m-6laRTe

SOURCE CODE :

REFERENCE APP :

<https://github.com/harshbhatt7585/YogaIntelliJ>

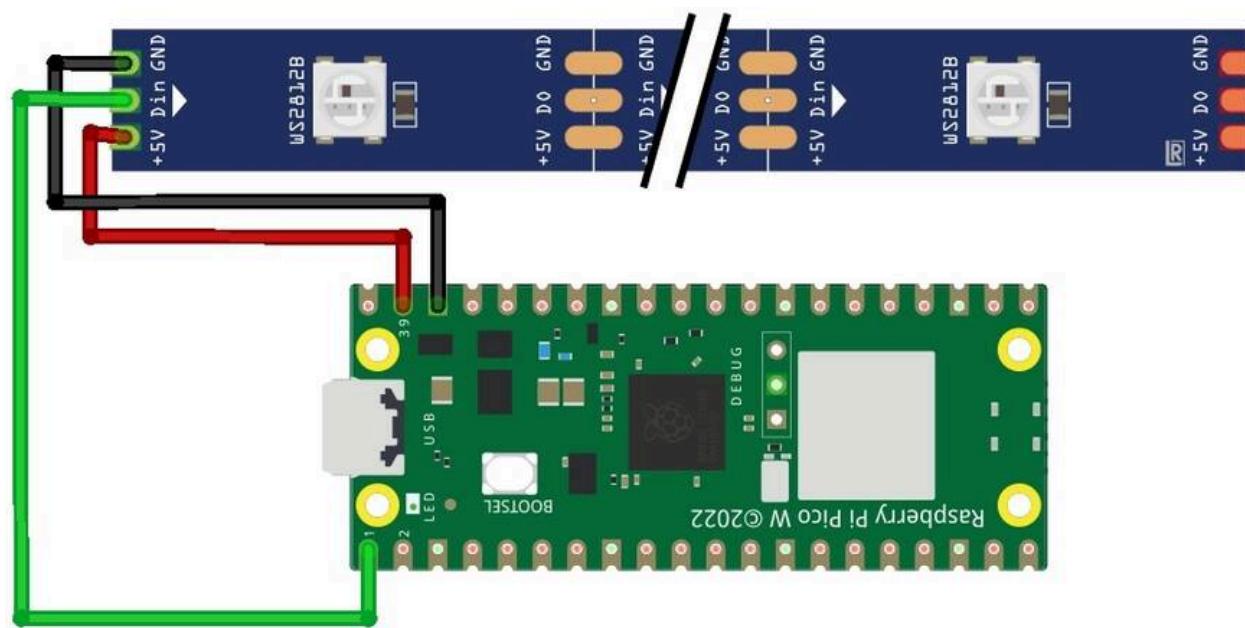
<https://eager-bardeen-e9f94f.netlify.app>



<https://pico.pinout.xyz>

I2c Communication between two or more boards

https://shillehtek.com/blogs/news/sending-data-between-multiple-raspberry-pi-pico-ws-in-thonny?utm_source=youtube&utm_medium=product_shelf



Code for the 2LDR connection for checking purpose

```
from machine import Pin  
import time
```

```
led = Pin(3, Pin.OUT)  
led.value(1)
```

```
ldr1 = Pin(0, Pin.IN)  
ldr2 = Pin(1, Pin.IN)  
ldr3 = Pin(2, Pin.IN)
```

```

while True:

    ldr1_value = ldr1.value()
    ldr2_value = ldr2.value()
    ldr3_value = ldr3.value()

    if ldr1_value == 1:
        #print("User detected by LDR 1, ignoring LDR 2")
        led.off()

    elif ldr3_value == 1:
        #print("User detected by LDR 2, no user detected by LDR 1")
        led.off()

    else:
        #print("No user detected")
        led.on()

    time.sleep(1)

```

Code with or statement

```

from machine import Pin
import time

```

```

ldr1 = Pin(0, Pin.IN)
ldr2 = Pin(1, Pin.IN)
ldr3 = Pin(2, Pin.IN)
ldr4 = Pin(3, Pin.IN)
ldr5 = Pin(4, Pin.IN)

```

```
led = Pin(3, Pin.OUT)
led.value(1)

while True:

    ldr1_value = ldr1.value()
    ldr2_value = ldr2.value()
    ldr3_value = ldr3.value()
    ldr4_value = ldr4.value()
    ldr5_value = ldr5.value()

    if ldr1_value == 1 or ldr2_value == 1 or ldr3_value == 1 or ldr4_value == 1 or ldr5_value == 1:
        print("Obstacle detected by one of the LDRs")
        led.value(0)
    else:
        print("No obstacle detected")
        led.value(1)
```

Battery Watt Hour (Here we are going to give 10000mah battery which is 37 WH)

mAh to Wh Conversion Chart

Milliamp Hours (mAh)	Voltage (V)	Watt Hours (Wh)
10,000mAh	3.7V	37Wh
15,000mAh	3.7V	55.5Wh
20,000mAh	3.7V	74Wh
25,000mAh	3.7V	92.5Wh
13 more rows		

Pico communication File

[easy_comms-main.zip](#)

Technical specification of LDR Module (Operating Current - 15 ma)

Specifications:

Operating Voltage	3.3V to 5V DC
Operating Current	15ma
Dimensions	5 x 4 x 3cms
Weight	50 grams

UART Communication Detailed explanation:

<https://youtu.be/kN2jCOF8zYc?si=84QS5QjPl-4QeDN->

Reference 2:

<https://github.com/JMartinUSN/Pi-PICO/commit/3ae8d4de11fbb7d85cc71dfa16a7fd1aa5ede5a8>

Arduino Nano UART Transmission :

SIR INPUTS

what is ur nature of work (posture) - which part of the body strains the most
food habits (to get to know abt their metabolism better)

spresence board buy sony

benefits from yoga aasana

validation

blockchain for data privacy

TIDE grant Meityfor patents

Scientific Writing for initial idea patent / proof

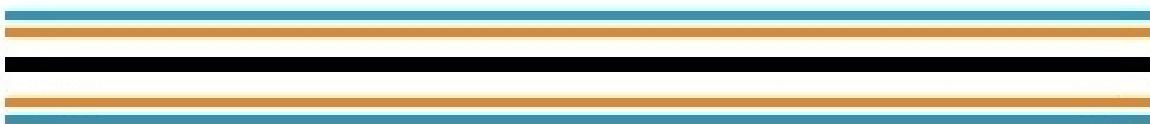
Opponent Sensor Material Creation :

<https://youtu.be/zUN2ZYdYAUo?si=Kdwvb0uQotMpCypf>

 **Plastic substrate**

 **Copper foil**

 **Velostat**



<http://multiwingspan.co.uk/pico.php?page=multiplex>

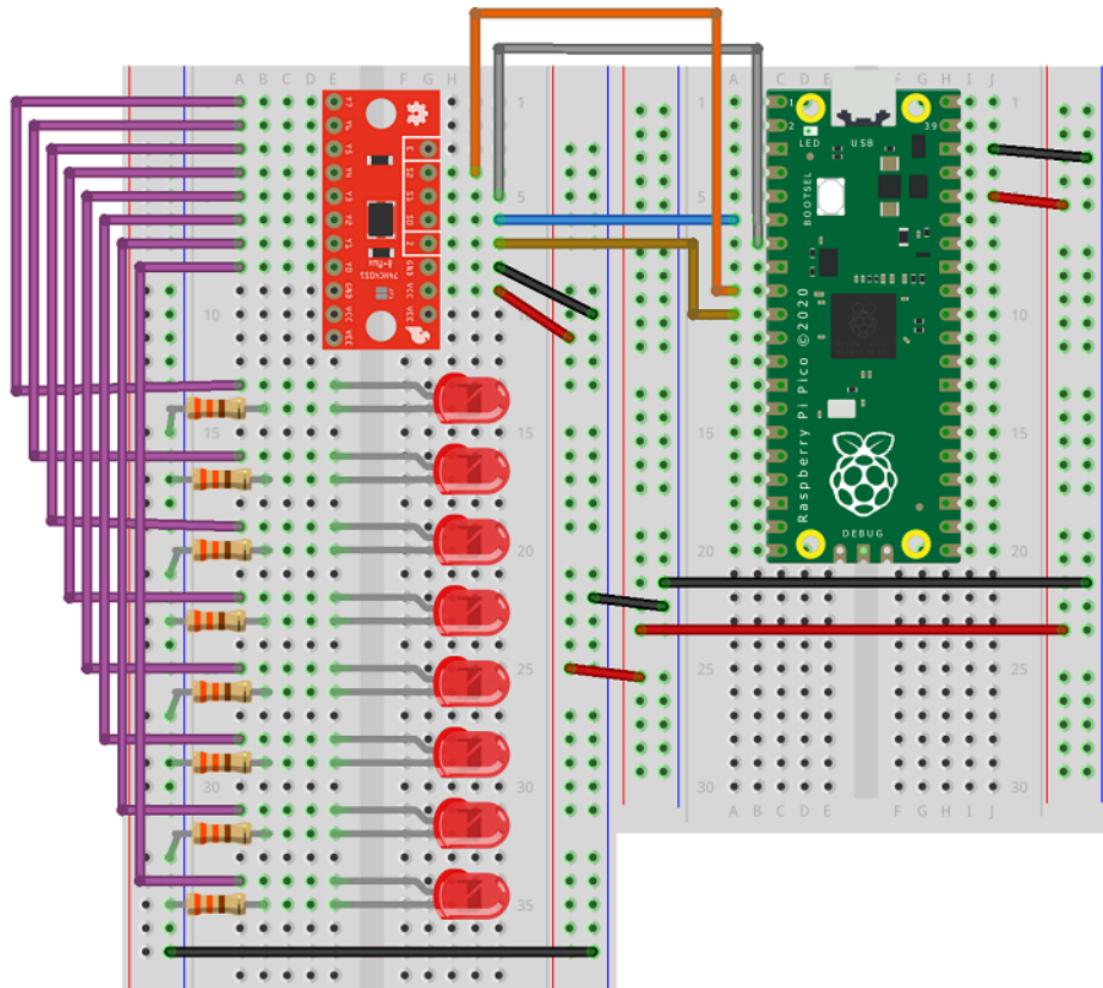
goes to the correct output.

```
from machine import Pin
from time import sleep

select_pin_nums = [4,5,6]
s_pins = [Pin(i, Pin.OUT) for i in select_pin_nums]
z_pin = Pin(7, Pin.OUT)

def select_pin(p, pins):
    for i in range(3):
        pins[i].value((p>>i)&1)

while True:
    for i in range(8):
        select_pin(i, s_pins)
        z_pin.value(1)
        sleep(0.25)
        z_pin.value(0)
        sleep(0.25)
```



Final code :

```
from machine import Pin
from time import sleep

select_pin_nums = [4, 5, 6]
s_pins = [Pin(i, Pin.OUT) for i in select_pin_nums]
z_pin = Pin(7, Pin.IN)

led = Pin(16, Pin.OUT)
led.value(1)
```

```
def select_channel(channel, s_pins):
    for i in range(3):
        s_pins[i].value((channel >> i) & 1)

while True:
    for i in range(8):
        select_channel(i, s_pins)
        if z_pin.value() == 1:
            led.value(0)
        else:
            led.value(1)

    time.sleep(1)
```

Editing Project

1