

Open Design & Technology

Empowering Designers to Embrace Technology

Week 2 Review

Week 3

Basic Programming

Handwritten mathematical notes and diagrams:

- $P_0 = 60000$, $\sum x = 9475$, $\sum x^2 = 10000$, $b = 5.634432$, $c = 5.4329$
- $D(x) = 2 + 3 + 4.31447$
- $\sqrt{a^2 + b^2} = x^2 \rightarrow x = \sqrt{a^2 + b^2}$
- $c(x, y) \left\{ \begin{array}{l} xy = c \\ cx - cy = 35^2 \\ 2\pi = c \end{array} \right.$
- $T^B, \frac{2x}{y} + \frac{d^2 + s^2}{c} + \frac{x^2 + s^2}{a}$
- $men = 584. + n^{av} (x^2 + 34x + 4)$
- $\sum_{x=2}^{n=14} N_{30} \cdot x - \frac{1}{2} [964 + xg + g^2]$
- $\beta = 9 + x^2 + y^2$

A Basic Computer

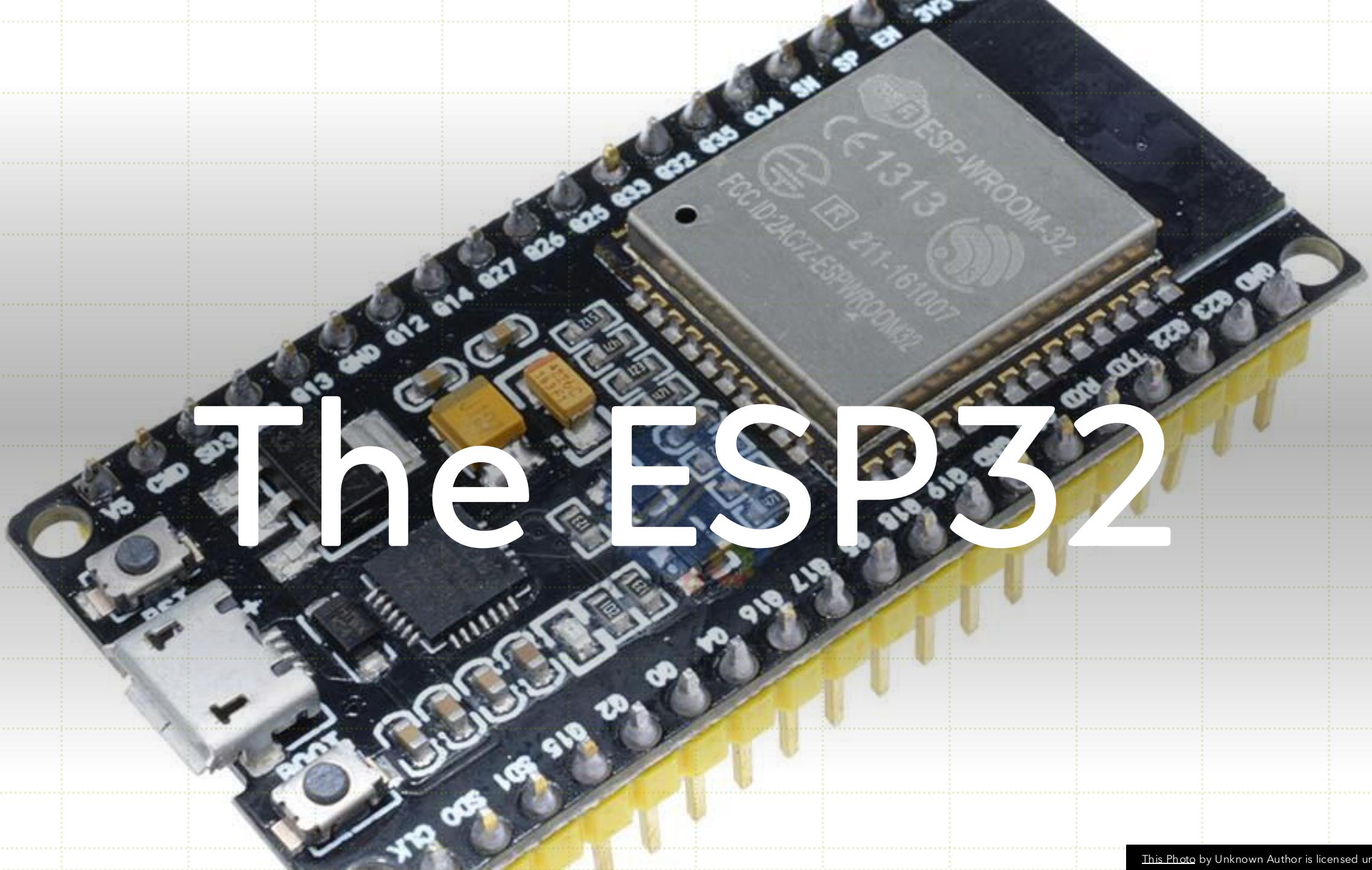


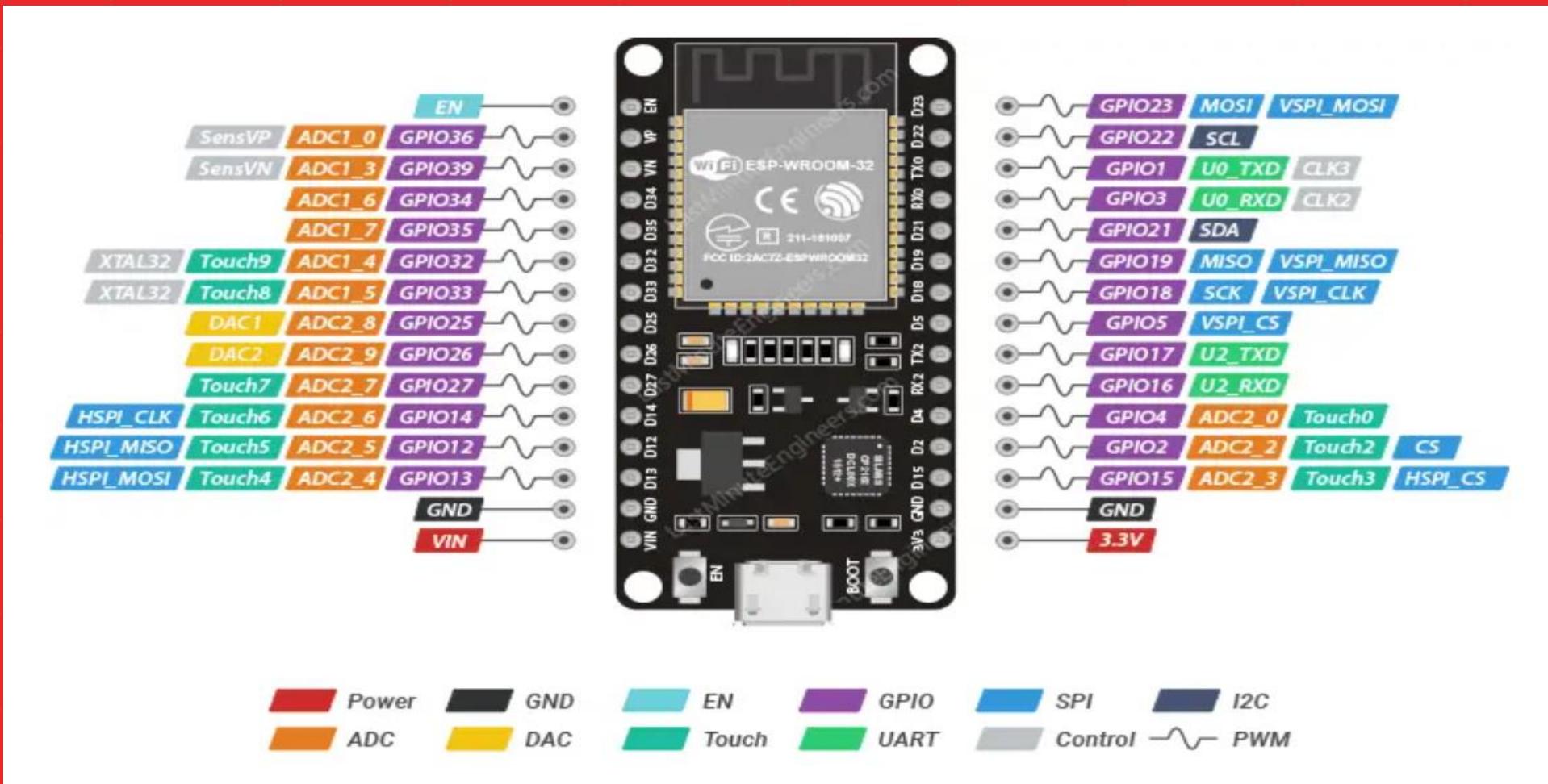
Computing Element

- Another Example
- Input | Output | Processing



The ESP32







Your first Python Code!

Coding Stage 1

LED

Delay Concept

Assignment Operator | Variable

2 or more external LEDs | Circuit Sketch

Circuit Debug : ESP32 Disconnect

```
    for object to mirror
    mirror_mod.mirror_object

    operation == "MIRROR_X":
        mirror_mod.use_x = True
        mirror_mod.use_y = False
        mirror_mod.use_z = False
    operation == "MIRROR_Y":
        mirror_mod.use_x = False
        mirror_mod.use_y = True
        mirror_mod.use_z = False
    operation == "MIRROR_Z":
        mirror_mod.use_x = False
        mirror_mod.use_y = False
        mirror_mod.use_z = True
```

```
selection at the end -add
    _ob.select= 1
    mirr_ob.select=1
    bpy.context.scene.objects.active = 
    ("Selected" + str(modifier))
    mirror_ob.select = 0
    bpy.context.selected_objects = 
    data.objects[one.name].select
```

```
int("please select exactly one object")
- OPERATOR CLASSES -
```

```
types.Operator):
    X mirror to the selected object.mirror_mirror_x"
    for X"
```

Operators in Python

Operators	Type
+ - * / %	Arithmetic
< <= > >= == !=	Relational
&& etc.	Logical
& ^ etc.	Bitwise
= += -= *= etc.	Assignment

Relational

`==`

Assignment

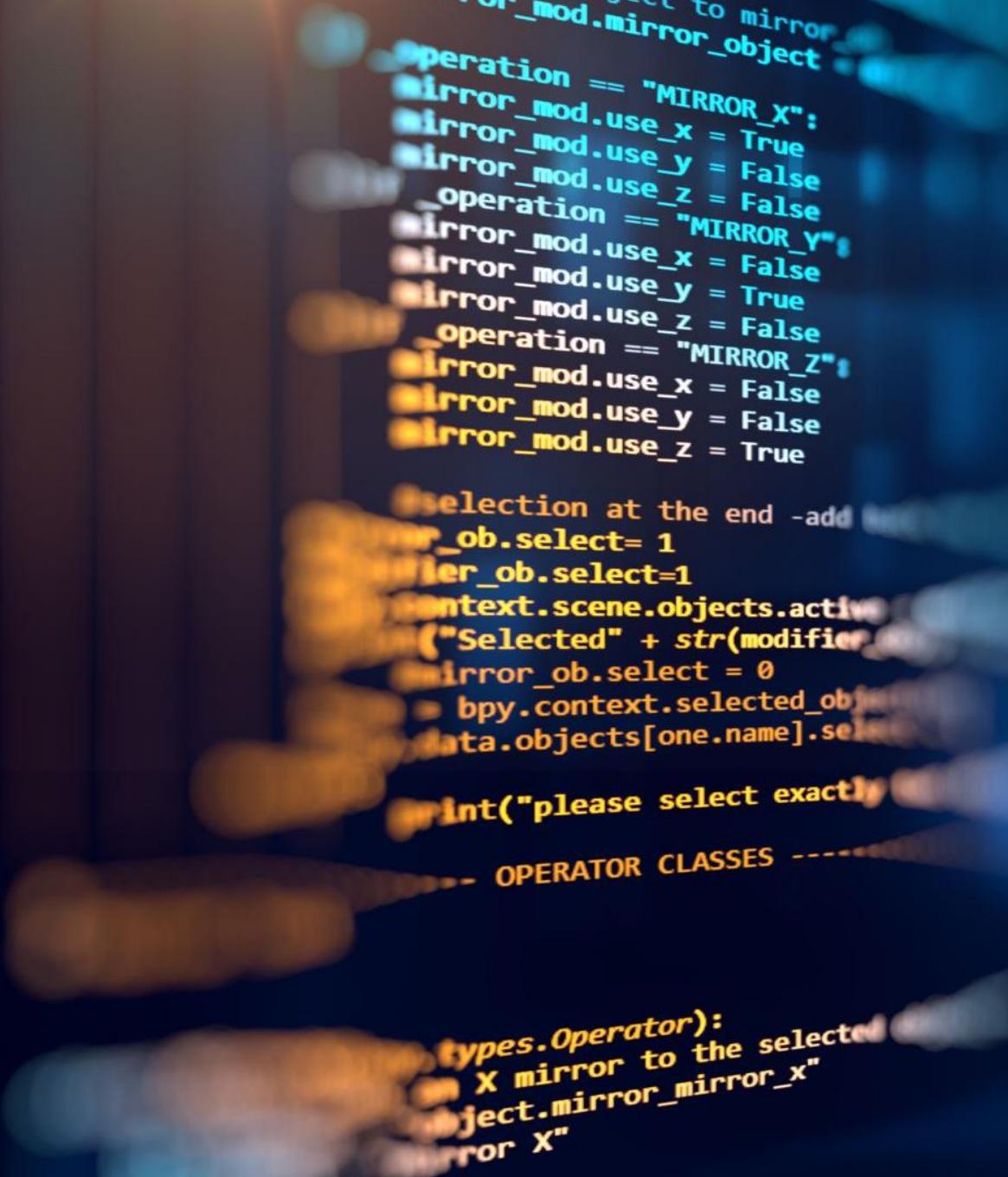
`=`

Coding Stage 1

Activity :

2 / more LEDs with different patterns

Circuit Debug : ESP32 Disconnect



The image shows a person's hand pointing at a computer screen. The screen displays a block of Python code. The code is part of a class definition, likely for a Blender operator. It includes logic for mirroring objects across different axes (X, Y, Z) and handling selection. The code uses variables like `mirror_mod`, `operation`, `mirror_use_x`, `mirror_use_y`, `mirror_use_z`, and `mirror_ob`. It also interacts with the Blender context and scene to select objects. A yellow box highlights the word `operator` in the code.

```
    for object in self.selected_objects:  
        if object == self.mirror_object:  
            continue  
        if self.operation == "MIRROR_X":  
            mirror_mod.mirror_object = object  
            mirror_mod.mirror_axis = "X"  
            mirror_mod.mirror_use_x = True  
            mirror_mod.mirror_use_y = False  
            mirror_mod.mirror_use_z = False  
        elif self.operation == "MIRROR_Y":  
            mirror_mod.mirror_object = object  
            mirror_mod.mirror_axis = "Y"  
            mirror_mod.mirror_use_x = False  
            mirror_mod.mirror_use_y = True  
            mirror_mod.mirror_use_z = False  
        elif self.operation == "MIRROR_Z":  
            mirror_mod.mirror_object = object  
            mirror_mod.mirror_axis = "Z"  
            mirror_mod.mirror_use_x = False  
            mirror_mod.mirror_use_y = False  
            mirror_mod.mirror_use_z = True  
  
    #selection at the end -add this  
    mirror_ob.select= 1  
    mirror_ob.select=1  
    bpy.context.scene.objects.active = mirror_ob  
    print("Selected" + str(modifier))  
    mirror_ob.select = 0  
    bpy.context.selected_objects.clear()  
    data.objects[one.name].select = 1  
  
    print("please select exactly one object")  
  
- OPERATOR CLASSES -  
  
types.Operator:  
    X mirror to the selected object.mirror_mirror_x"  
    or X"
```

Coding Stage 2

"while" Loop

Infinite "while" Loop

Breadboard + LED + Resistor (CC)

Circuit Debug : ESP32 Disconnect



The image shows a close-up of a person's hands interacting with a computer screen. The screen displays a block of Python code. One hand is pointing at the code, specifically at the line 'mirror_mod = modifier_ob'. The code is related to Blender's operator classes, specifically for mirroring objects.

```
    mirror_mod = modifier_ob
    # mirror object to mirror
    mirror_mod.mirror_object
    operation = "MIRROR_X":
        mirror_mod.use_x = True
        mirror_mod.use_y = False
        mirror_mod.use_z = False
    operation == "MIRROR_Y":
        mirror_mod.use_x = False
        mirror_mod.use_y = True
        mirror_mod.use_z = False
    operation == "MIRROR_Z":
        mirror_mod.use_x = False
        mirror_mod.use_y = False
        mirror_mod.use_z = True

    #selection at the end -add
    mirror_ob.select= 1
    mirror_ob.select=1
    context.scene.objects.active
    ("Selected" + str(modifier))
    mirror_ob.select = 0
    bpy.context.selected_objects
    data.objects[one.name].sele
    print("please select exactly one object")
    - OPERATOR CLASSES -
types.Operator:
    X mirror to the selected object.mirror_mirror_x"
    "mirror X"
context):
    context.active_object is not
```

Coding Stage 2

Activity :

LED(s) + Resistor + Buzzer
+ Loops + More

Circuit Debug : ESP32 Disconnect

```
mirror_mod = modifier_obj
# mirror object to mirror
mirror_mod.mirror_object

operation = "MIRROR_X":
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False

operation == "MIRROR_Y":
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False

operation == "MIRROR_Z":
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True
```

```
#selection at the end -add
mirror_ob.select= 1
modifier_ob.select=1
context.scene.objects.active = modifier
("Selected" + str(modifier))
mirror_ob.select = 0
bpy.context.selected_objects = []
data.objects[one.name].select = 1
print("please select exactly one object")
```

- OPERATOR CLASSES ---

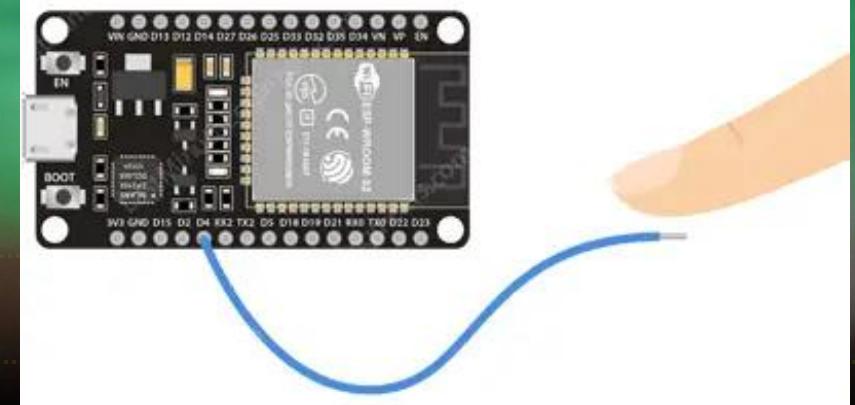
```
types.Operator):
    X mirror to the selected object
    object.mirror_mirror_x"
    "mirror X"
```

```
context):
    context.active_object is not None
```

Capacitive Touch

Specific Pins

Anything Conductive



Coding Stage 3

"IF" Loop

And everything together!