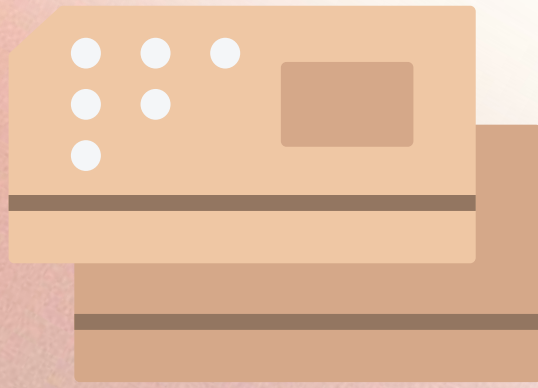


Circuits to Software: An Exploration of Computing Development in Three Stages.

Elizabeth Forrest Curry, department of
Mathematical and Computational sciences.

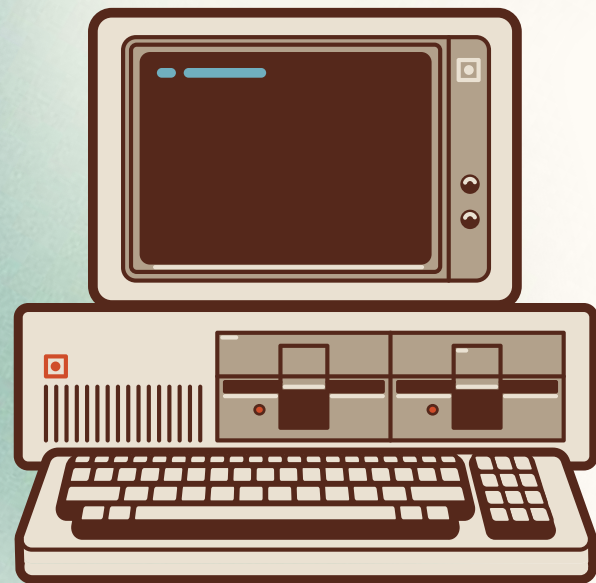
Advised by Dr. Dan Palmer



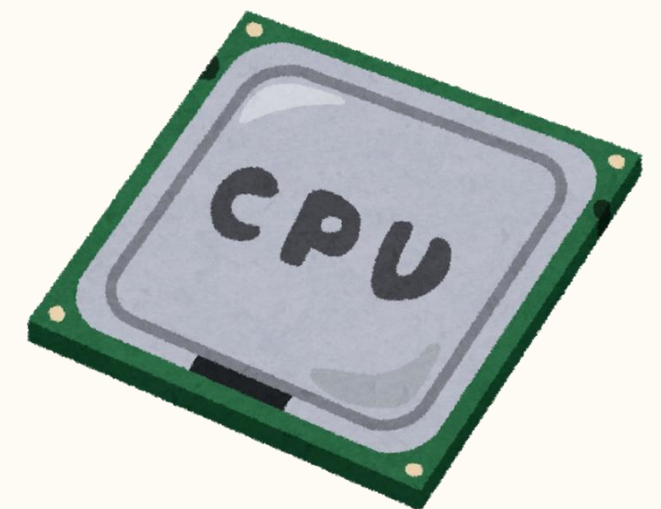
Motivations

Desire to work with hardware

How do computers really work?



How have they developed?



Project Approach

Create a puzzle game three ways and compare them

1. **Software:** implementation In Python
2. **Arduino Circuit:** physical components controlled by code
3. **Raw Circuit:** simple electrical components only

Driving puzzle

Mystery Buttons Puzzle

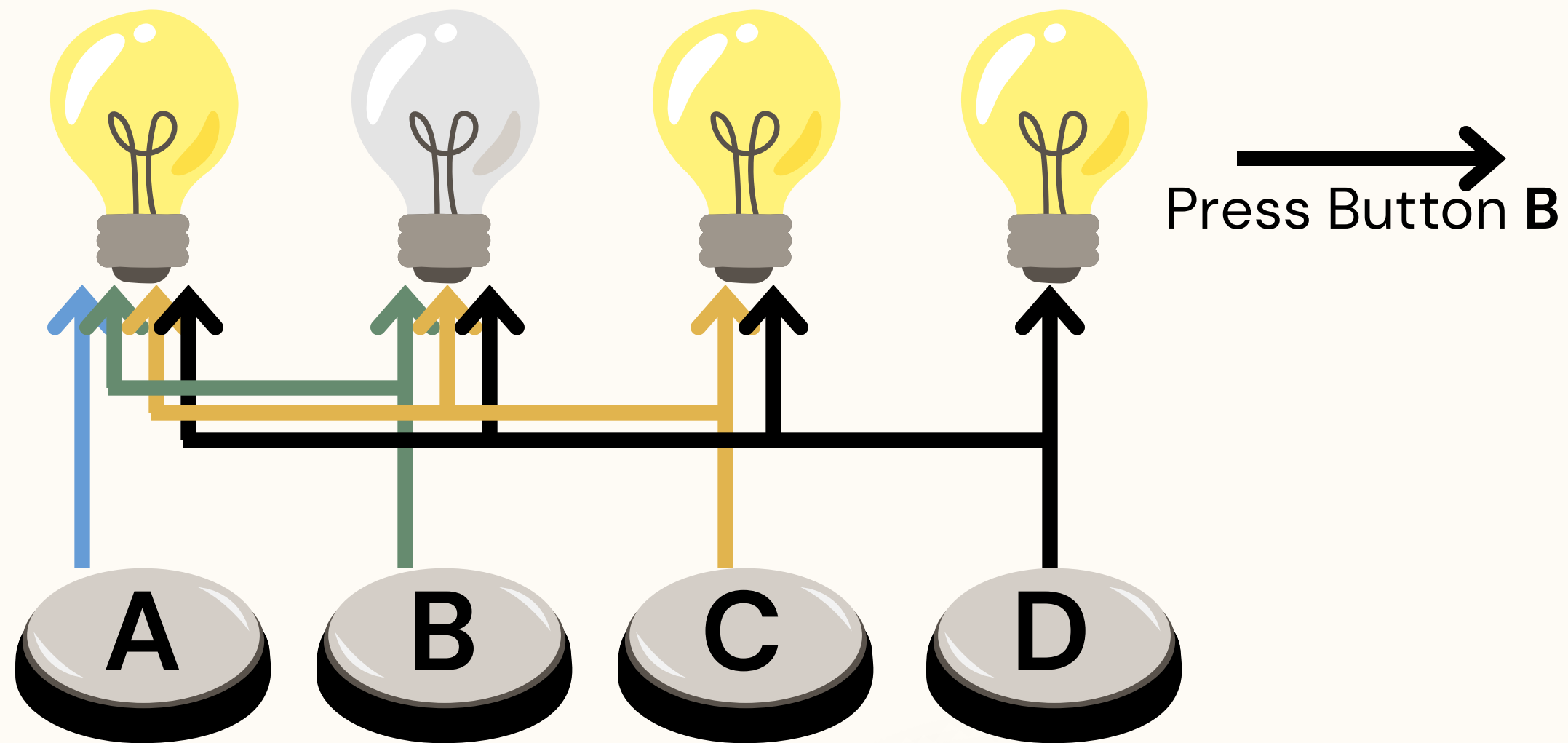
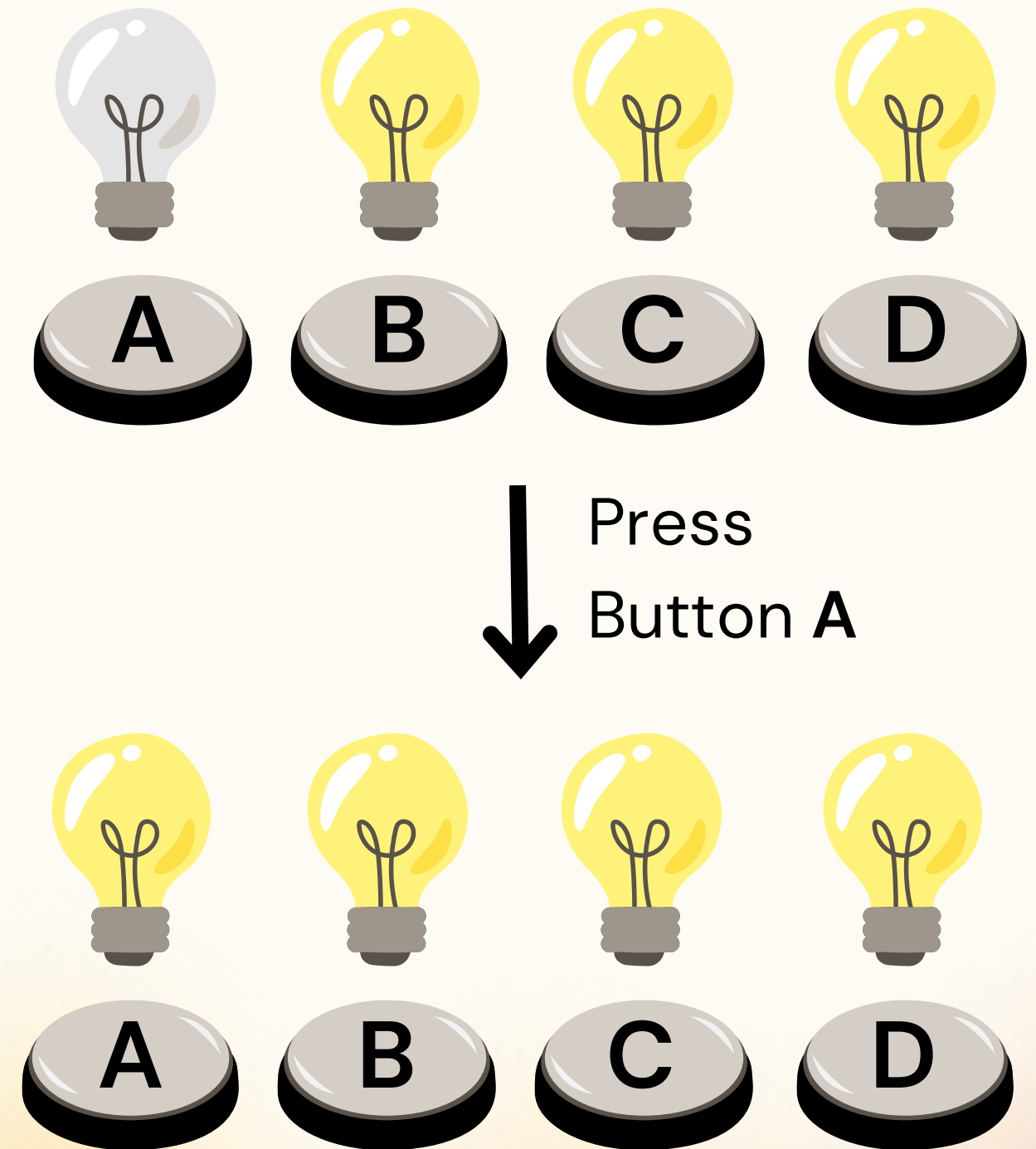


Figure 1: Mystery Buttons game state example



Software

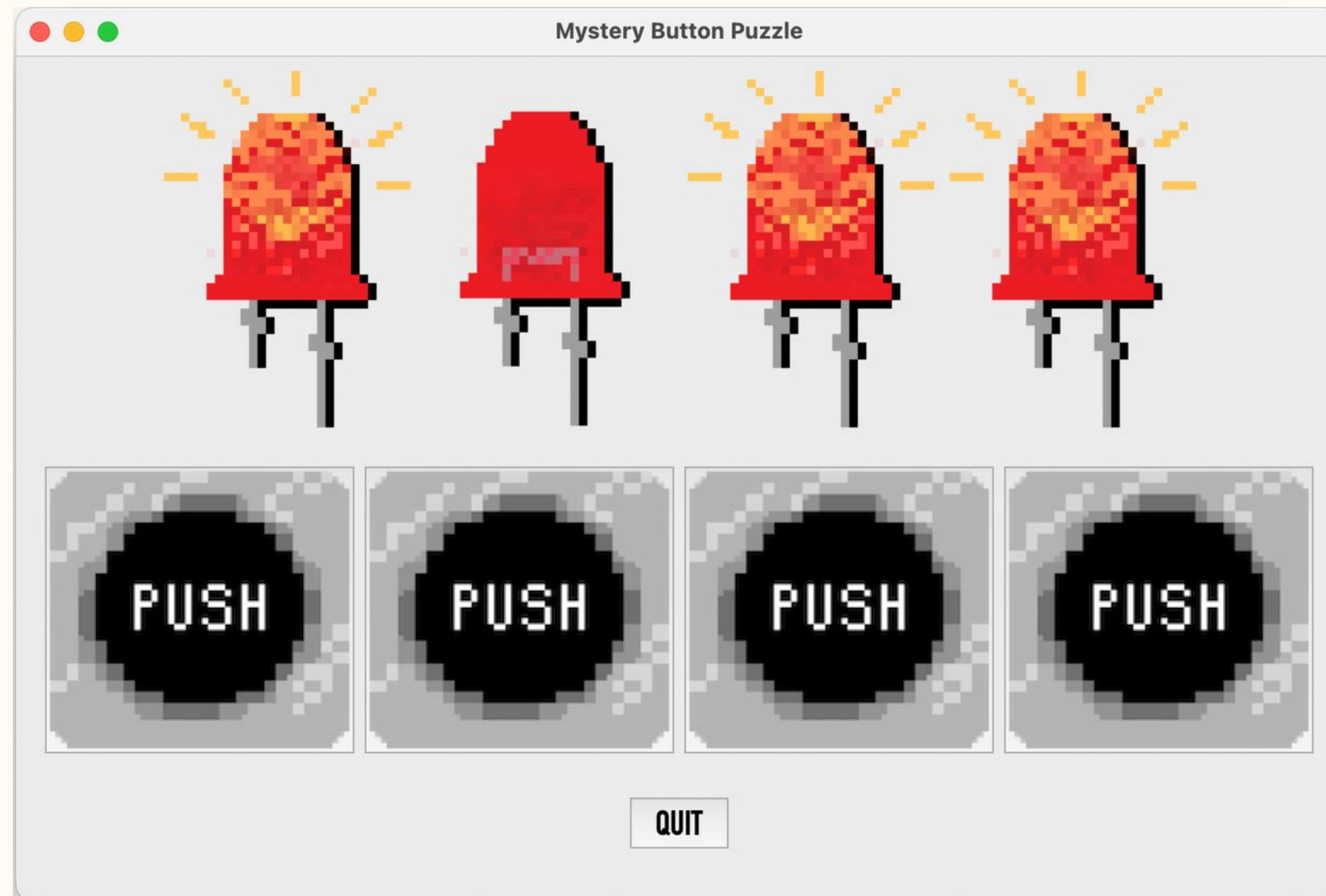


Figure 2: GUI for the software implementation

Software

```
def computeGameLogic(num):  
    global lights  
    for i in range(len(lights)):  
        lights[i] = lights[i] ^ buttons[num][i]  
    return
```

Figure 3: Game logic in software

↓ In plain
English

1. Check every light one by one
2. If the light is on and the button connects turn the light off
3. If the light is off and the button connects turn the light on
4. If the button doesn't connect change nothing

Arduino Circuit

Software

1. Wait for button press
2. Toggle light/s on or off if button is connected
3. Wait 80 milliseconds as to not read false button press

Code is in 'brain'
of Arduino

Hardware

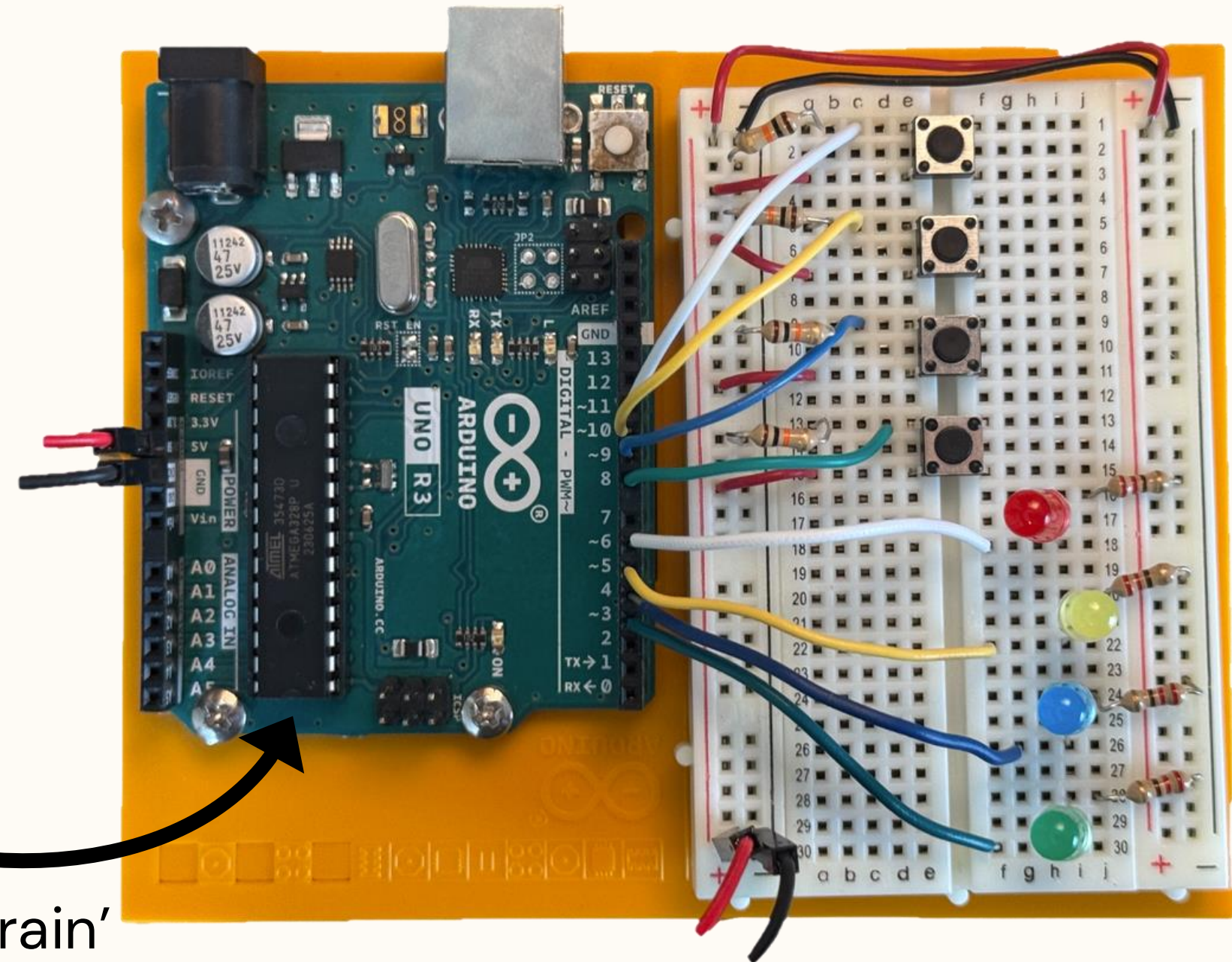


Figure 4: Arduino implementation

Button Bouncing

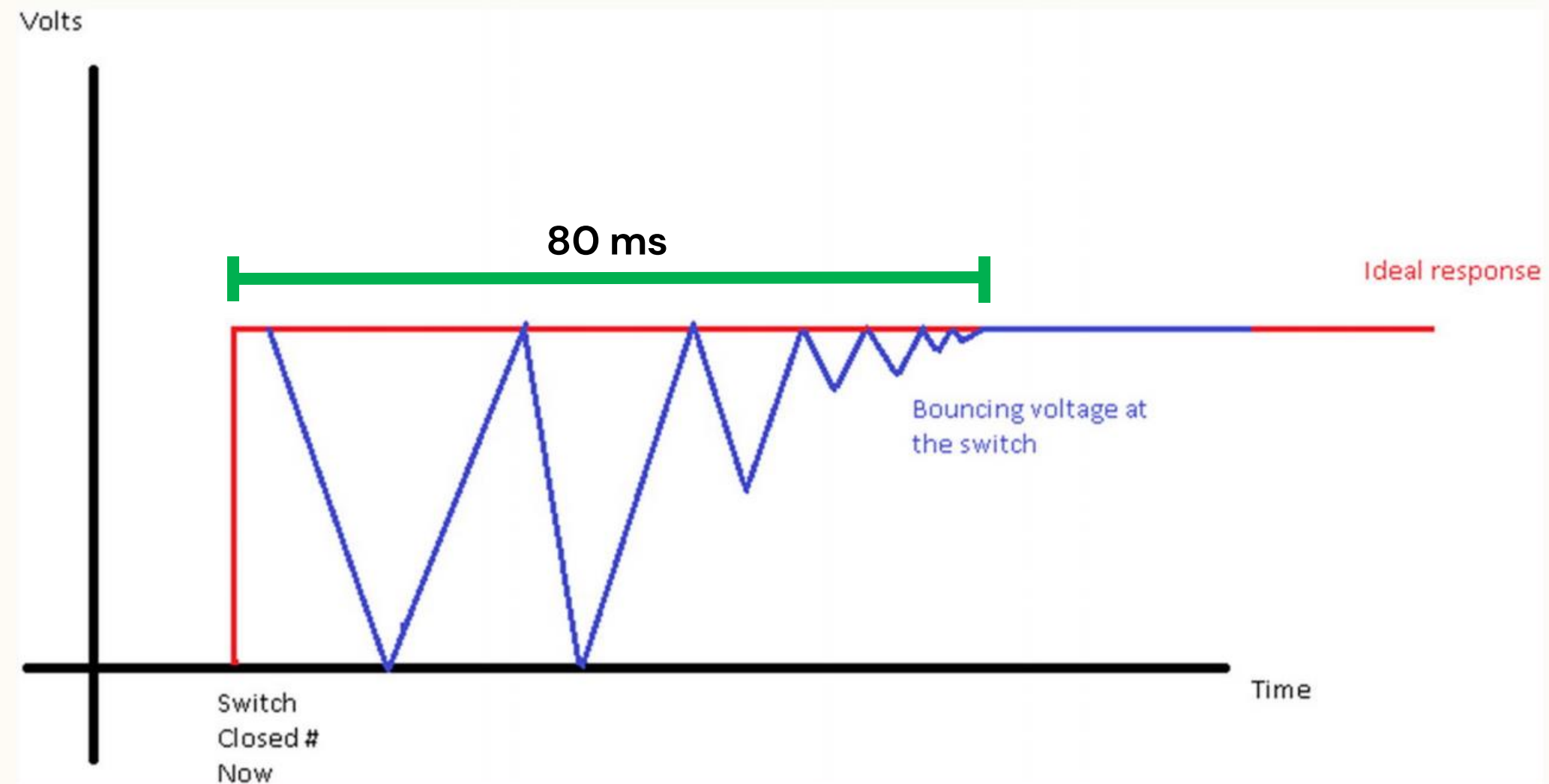


Figure 5: Visualization of bouncing voltage from physical button input [4]

Raw Circuit

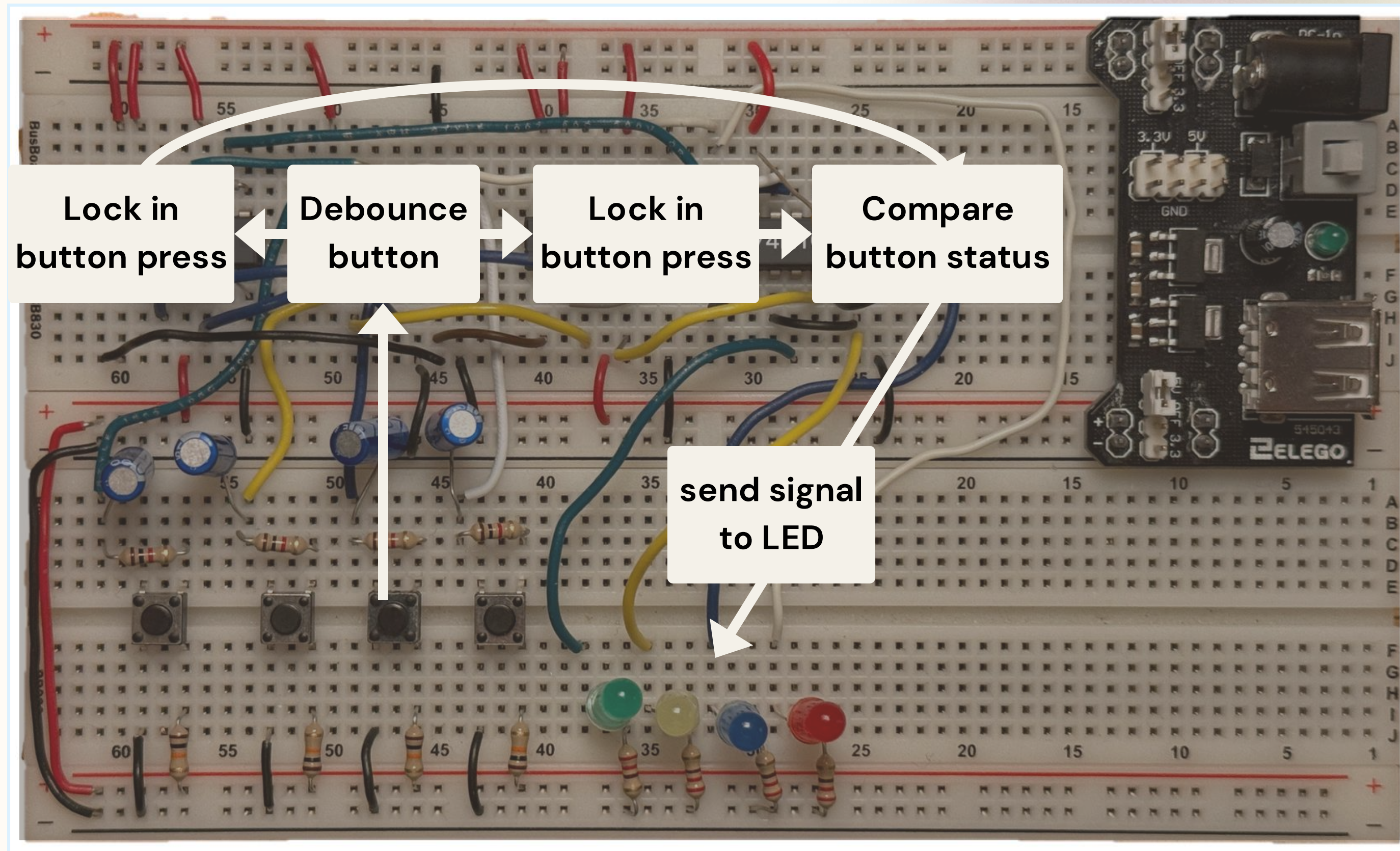


Figure 6: Raw circuit implementation, 74HC74N: D flip flop, 74HCT14N: Schmitt Trigger, 74HTC86N: XOR gates

Raw Circuit

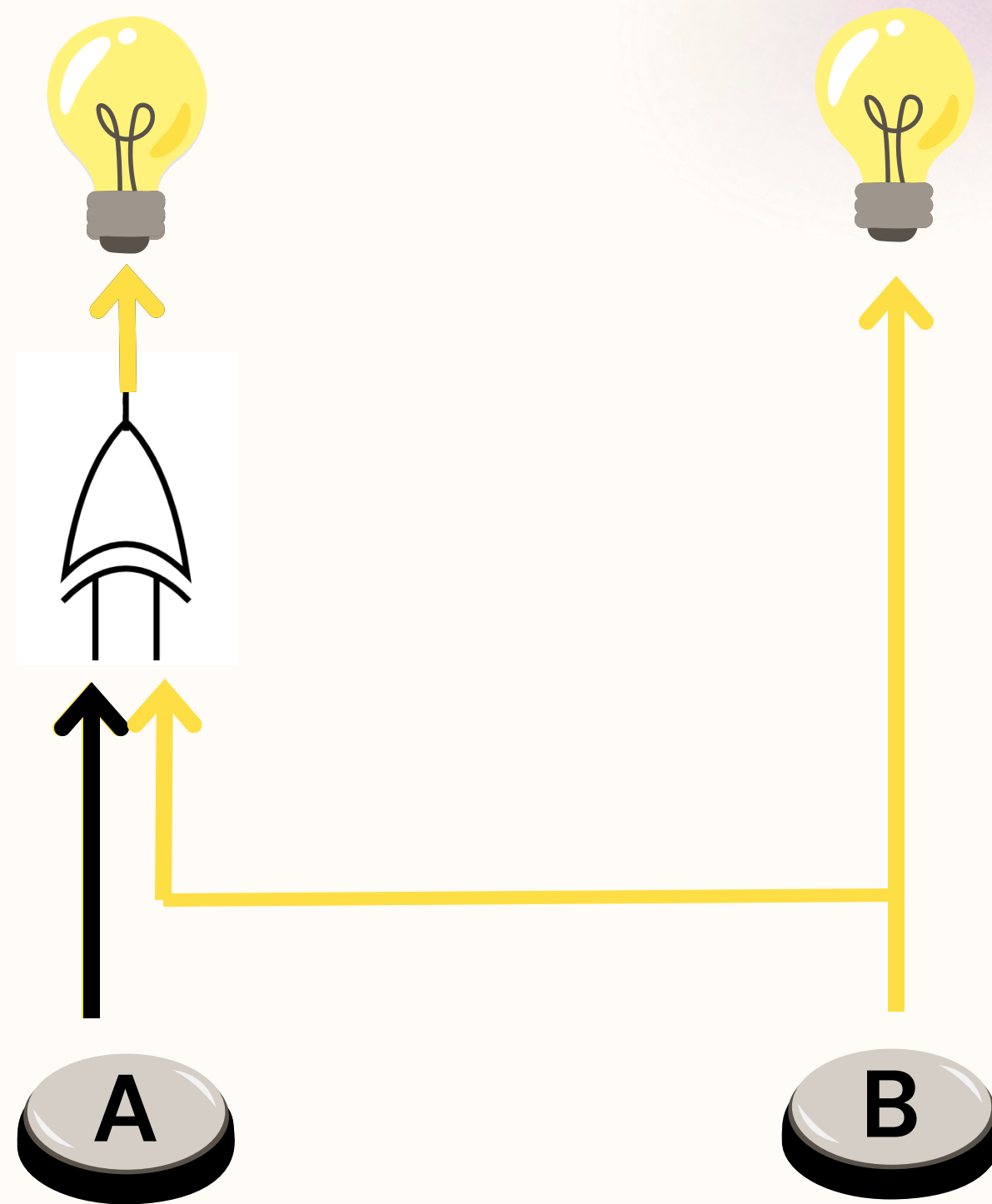


Figure 6: Example button, light toggle logic

Conclusion

Implementation	Effort	Flexibility
Software	1	10
Arduino	4	5
Raw Circuit	8	2

Figure 7: Example button, light toggle logic

- Underlying logic remains the same
- Logic develops into more accessible ways uses
- Complexities are abstracted away

References

- [1] Mouser.com. 74HC14 Hex SchmittTrigger Inverter. 2007. url:<https://www.mouser.com/datasheet/2/308/74HC14.REV1-34947.pdf?srsId=AfmBOor9JuffsKYoQTlohvuCoSkJdckpnRjtW6FJQyN7XIZ2uFUrcfVk> (page 44).
- [2] Plantz, Robert G. "Chapter 7: Sequential Logic Circuits". In: Introduction to Computer Organization. No Starch Press, Jan. 2022. isbn: 9781098130183. url: <https://learning.oreilly.com/library/view/introduction-to-computer/9781098130183/> (pages 26, 29).
- [3] Texas Instruments. SNx4HC86 Quadruple 2-Input XOR Gates. Dec. 1982. url: <https://www.ti.com/lit/ds/symlink/sn74hc86.pdf> (page 32).
- [4] Ward, Hubert Henry. "Chapter 4: Moving On from the NAND Gate". In: Mastering Digital Electronics. Maker Innovations Series. Berkeley, CA: Apress, 2024, pp. 139–176. isbn: 978-1-4842-9880-0. doi: 10.1007/978-1-4842-9878-7_4. url: https://link.springer.com/10.1007/978-1-4842-9878-7_4 (pages 29–30).