

Every Computer has a tell. HARDHACK

ELECTRONICS & ROBOTICS CLUB | CYBERSECURITY COMMUNITY

ERC & CSeC PRESENT

HARDHACK

Introduction to Hardware Hacking

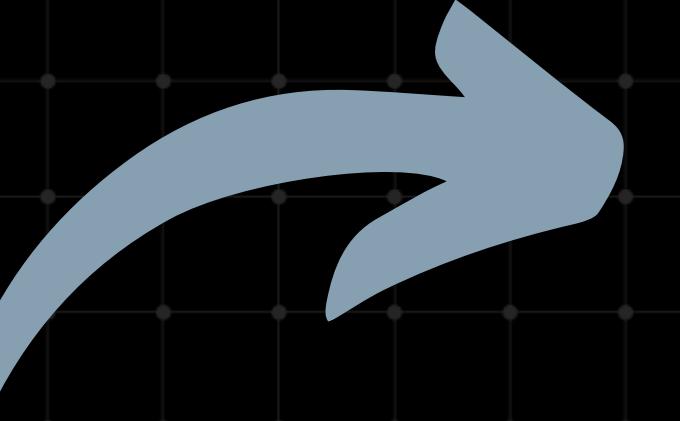


Well, Why do Hackers Hack?

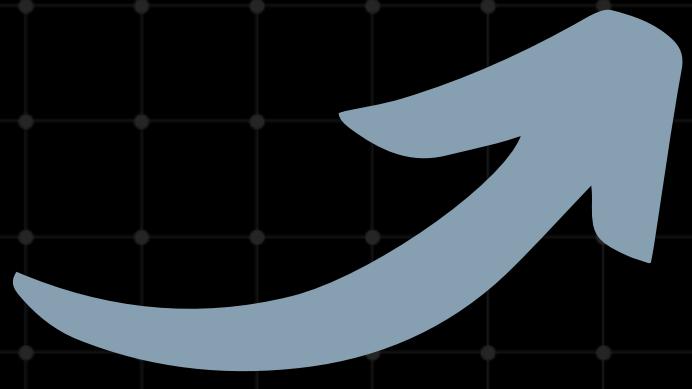
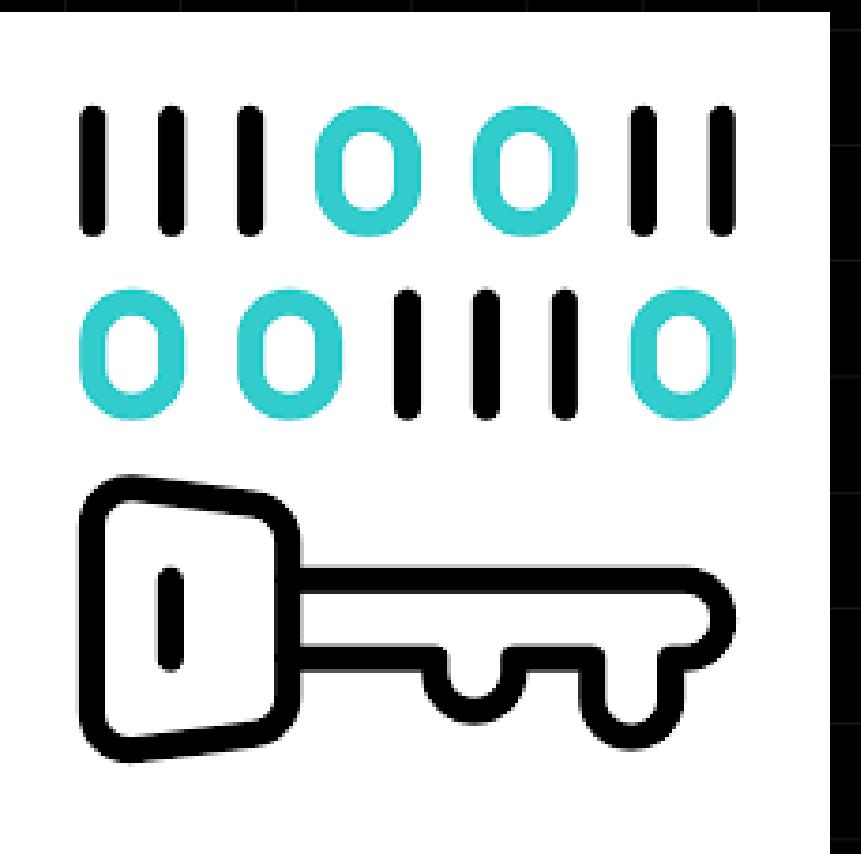
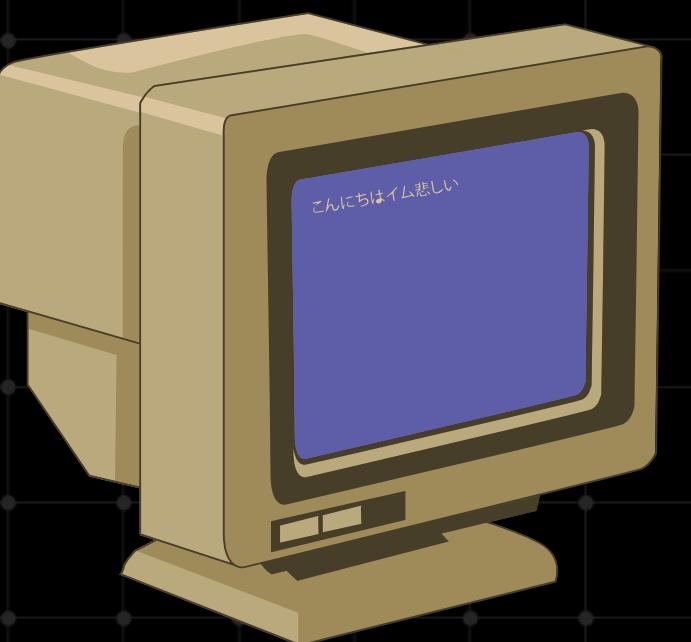


How do we send data-
SECURELY?

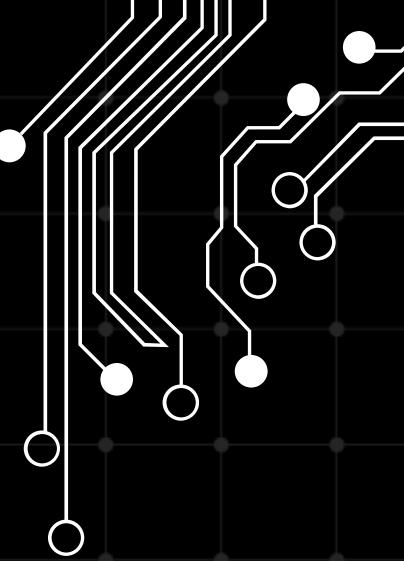
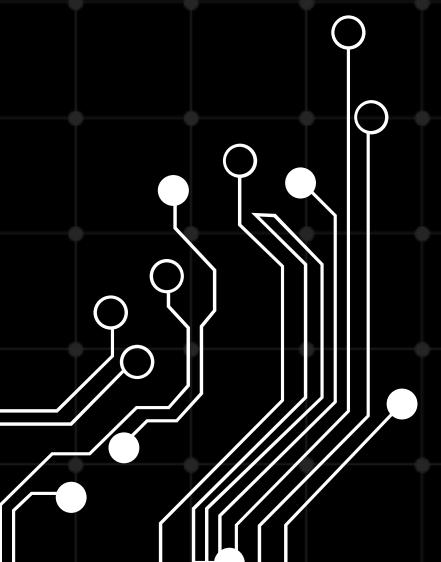
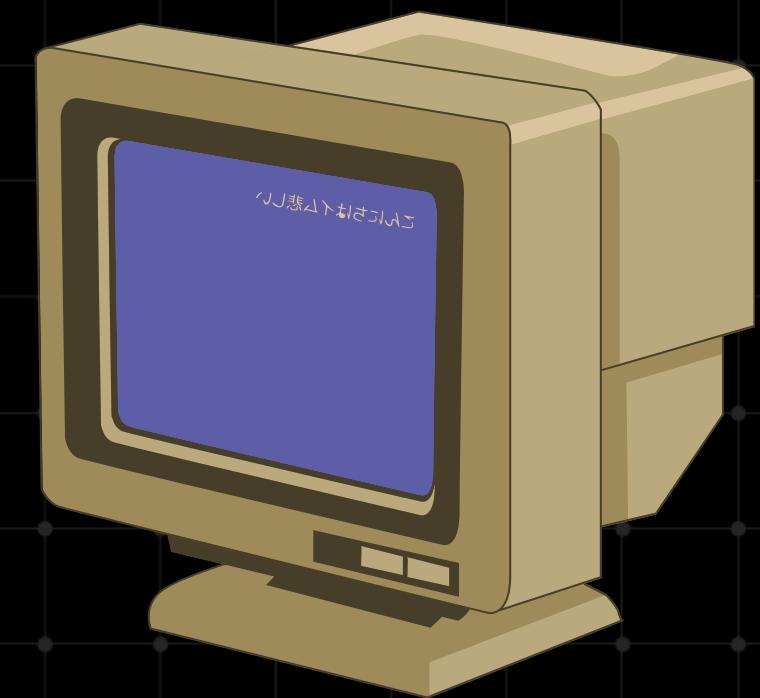


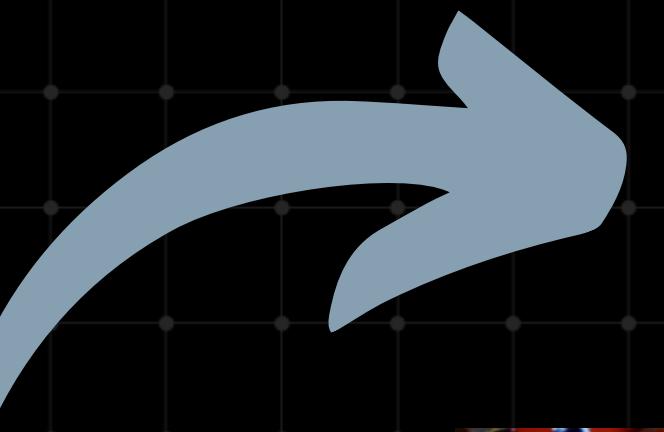


ENCRYPTION

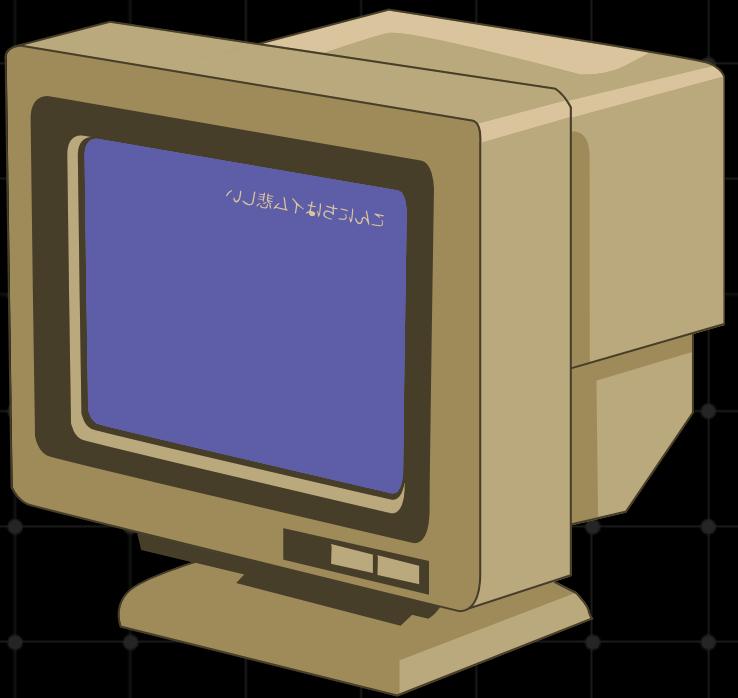
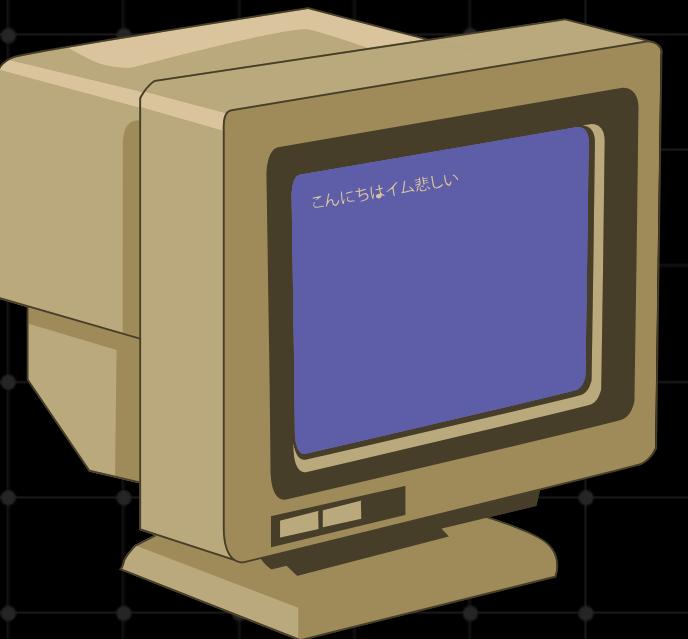


DECRIPTION



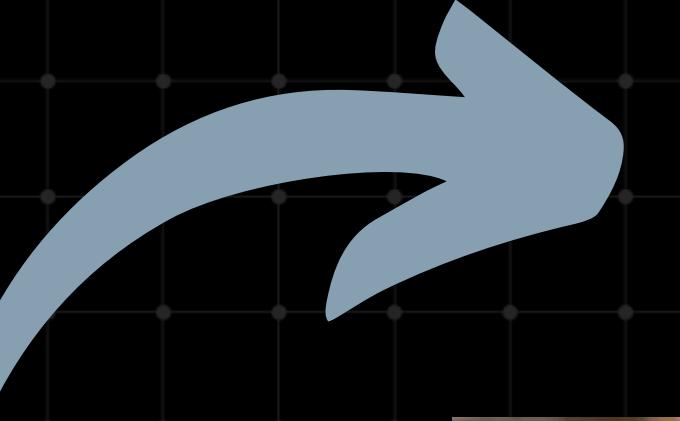


ENCRYPTION

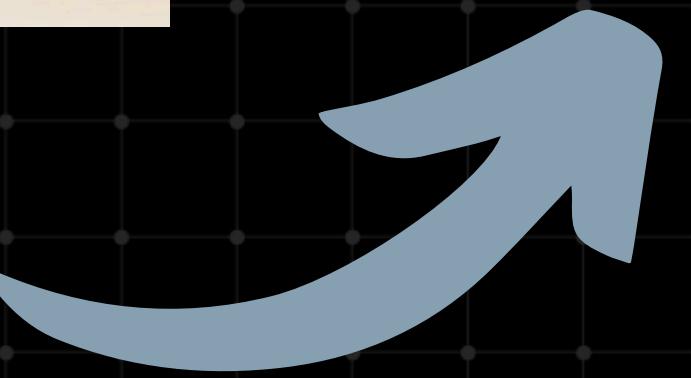
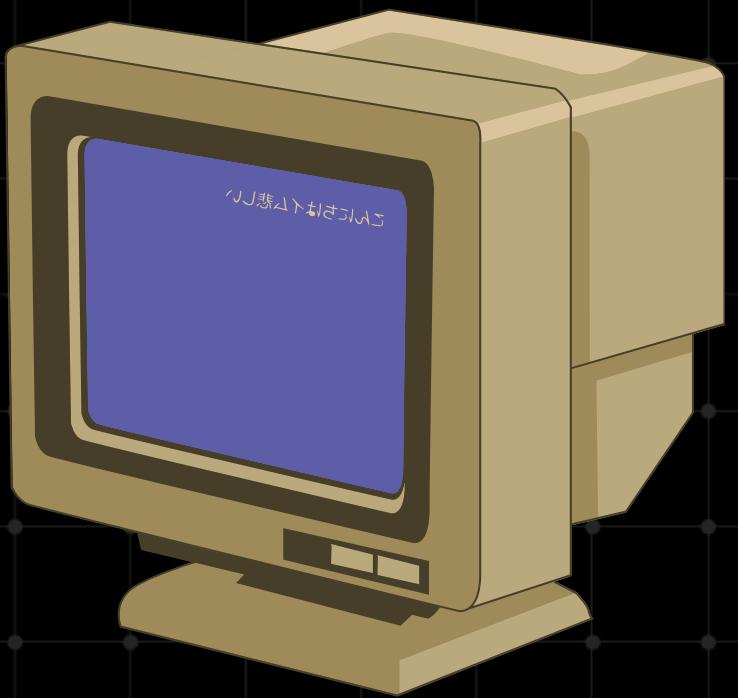
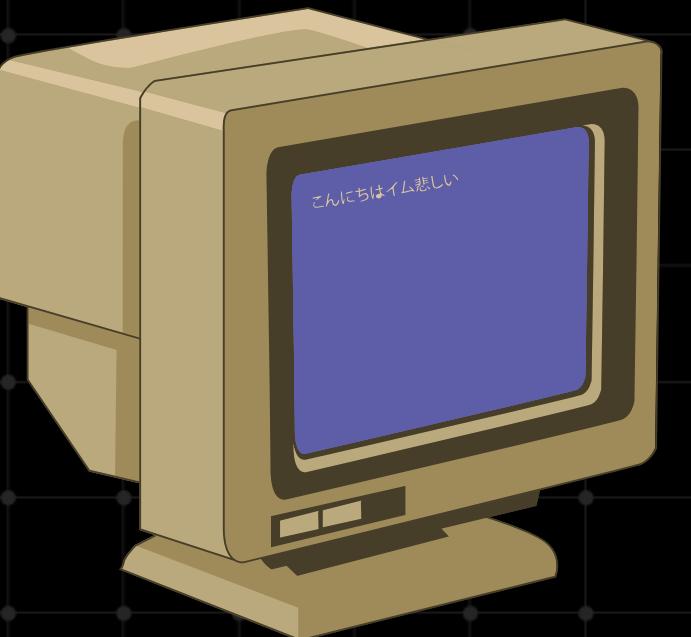


DECRYPTION



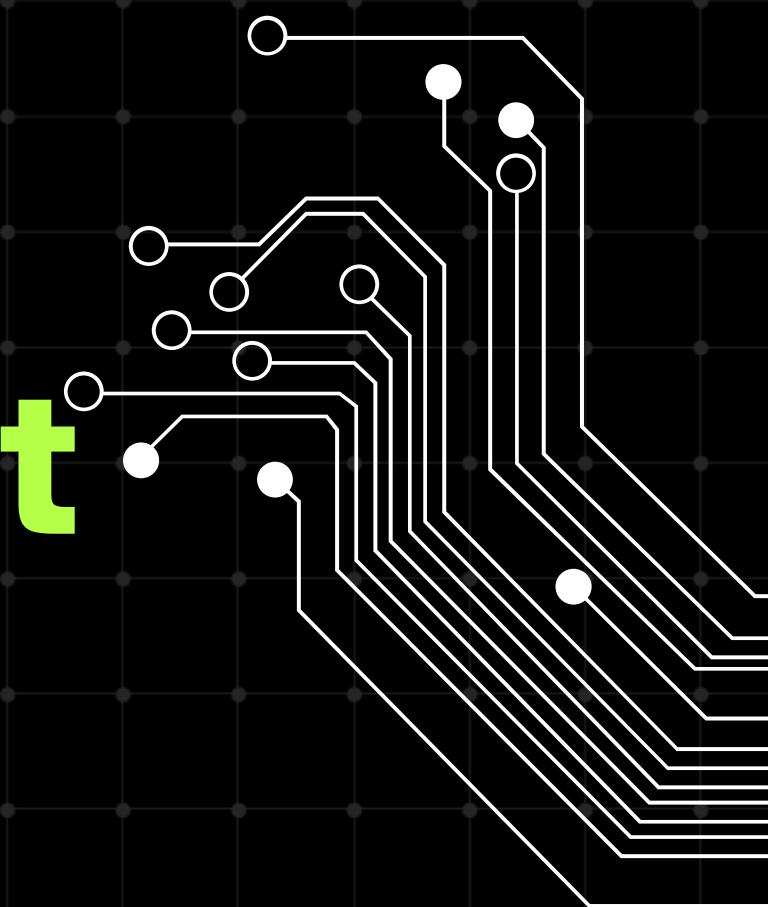


ENCRYPTION

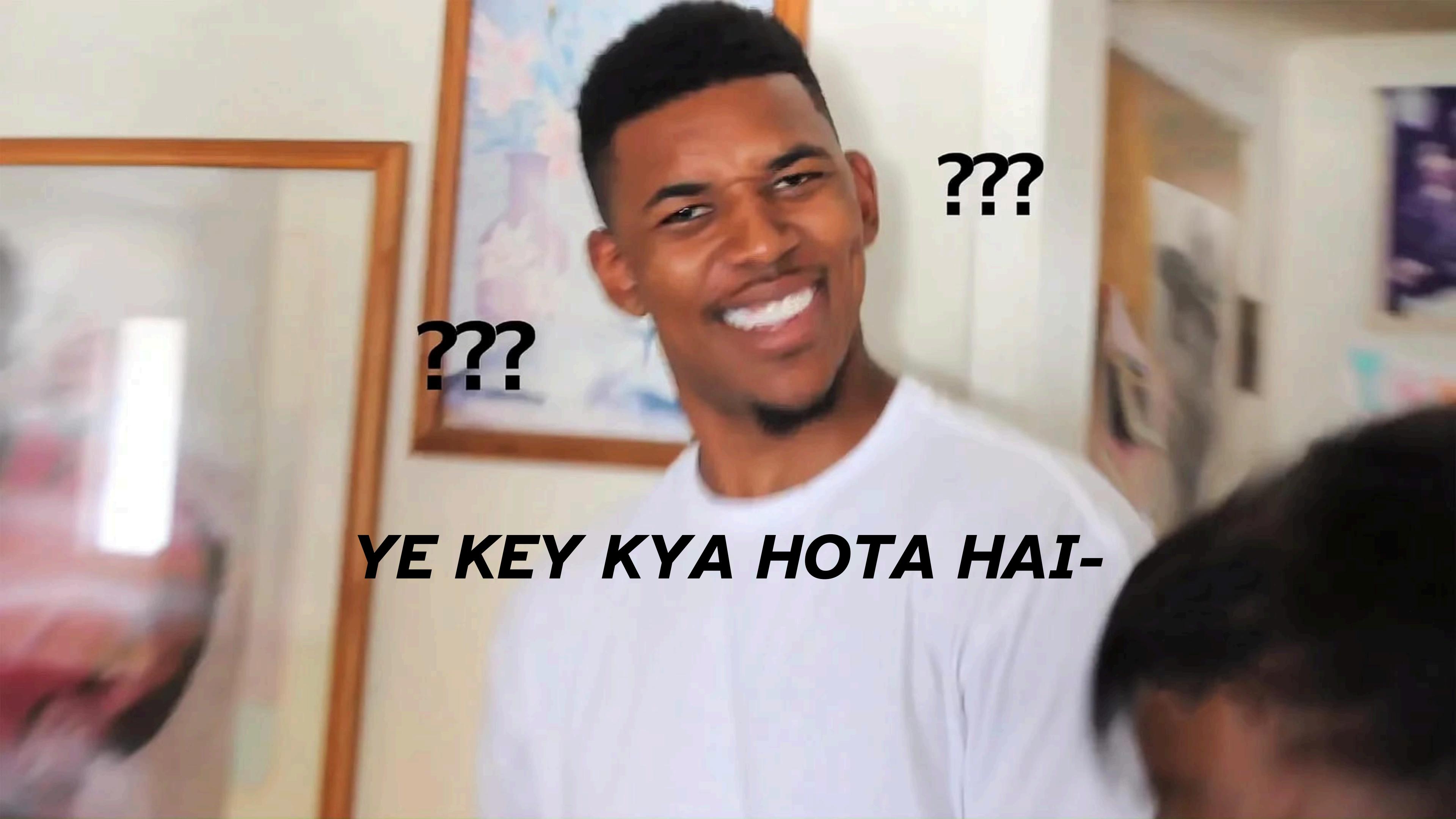


DECRIPTION

What do hackers do?
Try whatever they can to get
the key.



Par Didi...



???

???

YE KEY KYA HOTA HAI-

101101 DECRYPTION KEY

UCJAMKC RM FYPBUYPC FYAIGLE

Maps to: Y Z A B C D E F G H I J K L M N O P Q R S T U V W X

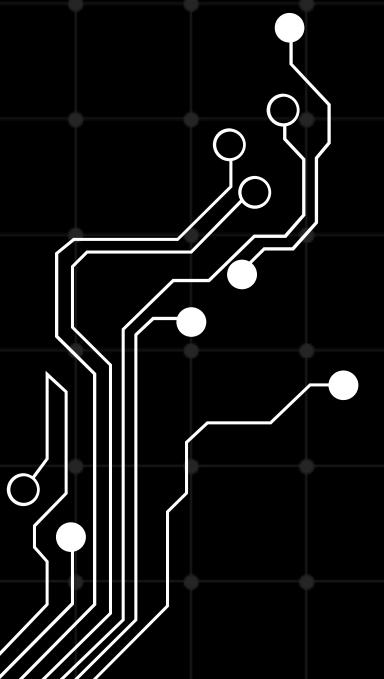
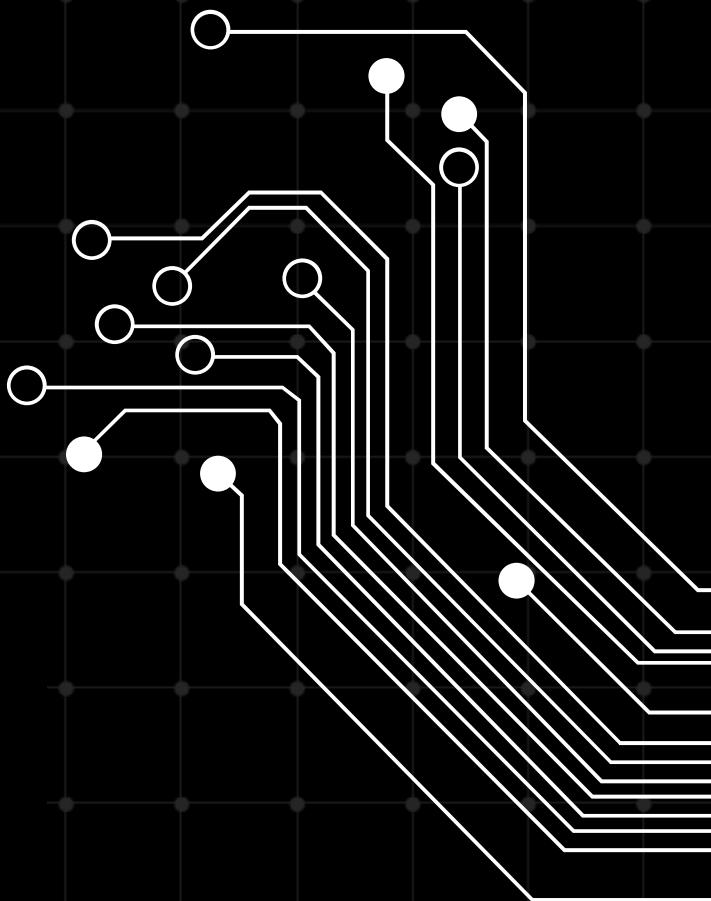
Original: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

"SHIFTING 2 LETTERS"

WELCOME TO HARDWARE HACKING

101101 **DECRYPTION**
KEY

&>ydd>yR &>nCEDQ Ex
GVXW HBCRy dEWF >
nBcHvwRy- Ew wRBbXg
LBv n>wnFi



A black background featuring a subtle grid pattern of small white dots. In the top-left and bottom-right corners, there are stylized white circuit board patterns with various lines, ovals, and small circles.

But, How exactly?

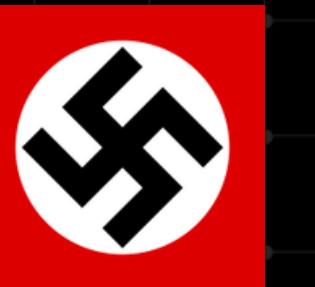
SIDE CHANNELS

GERMAN TANK PROBLEM

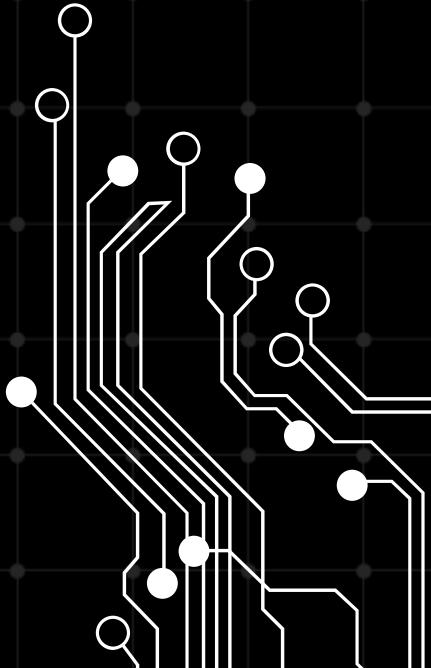
A decorative background consisting of a grid of small white dots on a black background, with white circuit board traces and component outlines branching out from the top left and bottom right corners.

But, How exactly?

SIDE CHANNELS

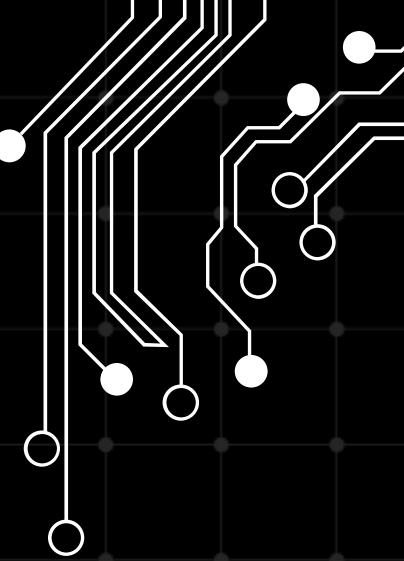
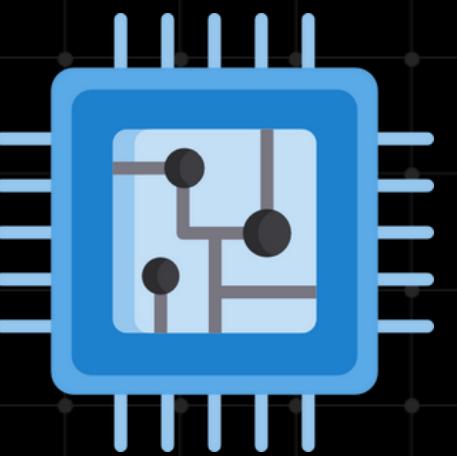


TANK PROBLEM





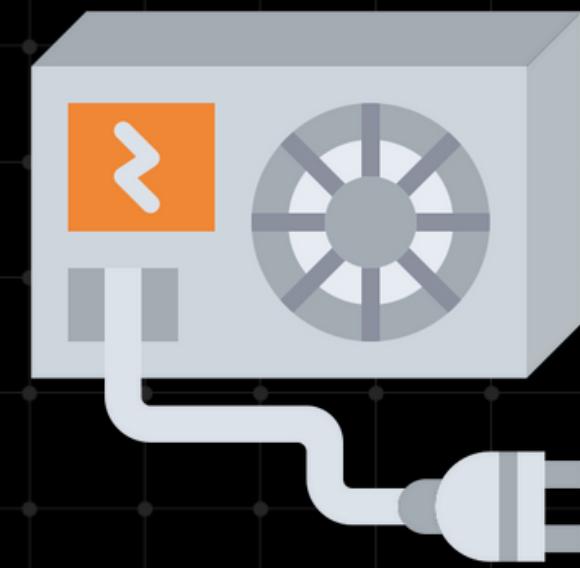
SIDE-CHANNEL ANALYSIS



TIMING ATTACKS



POWER ANALYSIS





TIMING ATTACKS

**Let's say,
Hypothetically,
You were to hack your TA's phone.**

This is completely hypothetical ~~for legal reasons~~



TIMING ATTACKS

Now, how to crack this?

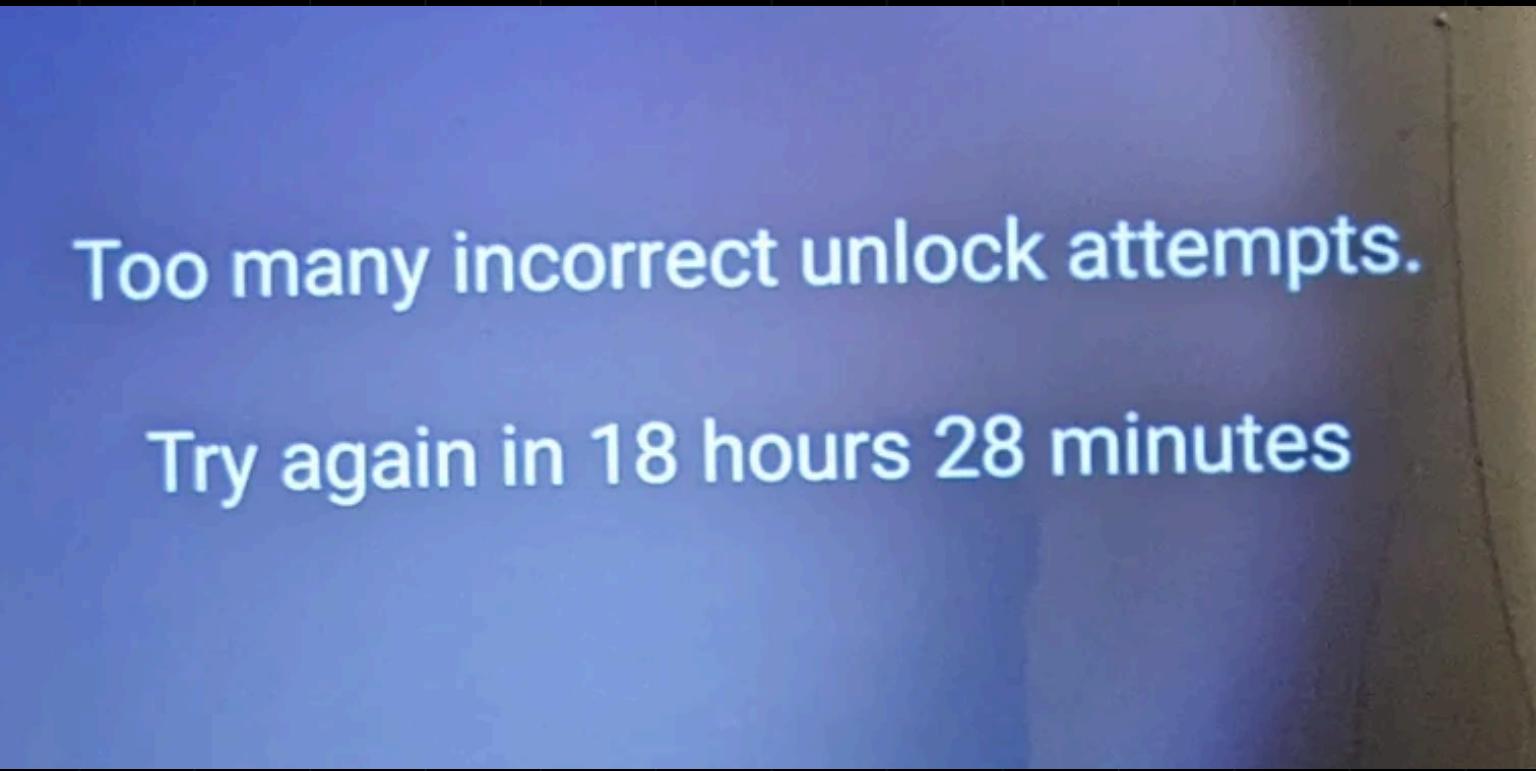
Well... You try all the
combinations.

10000

DIFFERENT COMBINATIONS



But this guy pops up



Hypothetically, for legal reasons

A SIMPLER EXAMPLE!

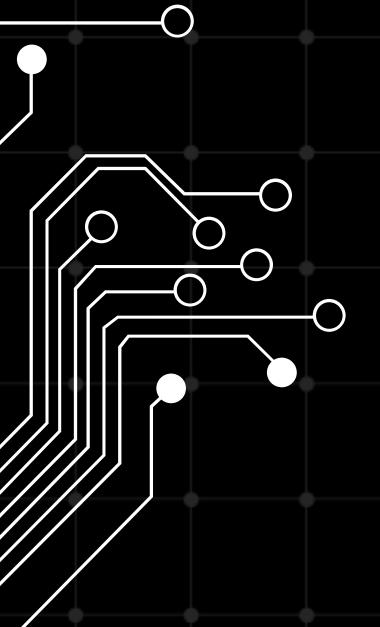
The screenshot shows the Arduino IDE interface with a project for a keypad. The code editor contains the following C++ code:

```
1 #include <Keypad.h>
2
3 #define ROWS 4
4 #define COLS 4
5 #define PASSWORD_LENGTH 4
6 int LED_RED=13;
7 int LED_GREEN=12;
8
9 // Keypad button layout
10 char keys[ROWS][COLS] = {
11     {'1', '2', '3', 'A'},
12     {'4', '5', '6', 'B'},
13     {'7', '8', '9', 'C'},
14     {'*', '0', '#', 'D'}
15 };
16
17 byte rowPins[ROWS] = {9,8,7,6}; // Keypad row pins
18 byte colPins[COLS] = {5,4,3,2}; // Keypad column pins
19
20 Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS,
21
22
```

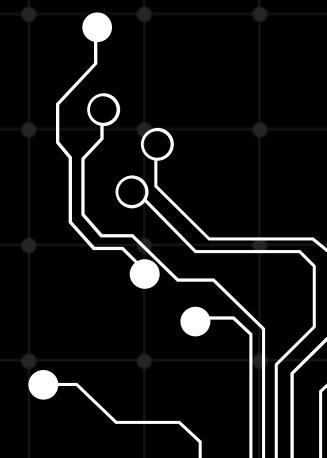
The Serial Monitor window is currently empty.

The hardware setup shown in the IDE includes an Arduino Uno connected to a breadboard. A 4x4 keypad is connected to the breadboard, which is then connected to the Arduino Uno. A red LED is connected to digital pin 13 (labeled LED_RED) and a green LED is connected to digital pin 12 (labeled LED_GREEN). A 10k pull-down resistor is also connected between the keypad ground and digital pin 9.

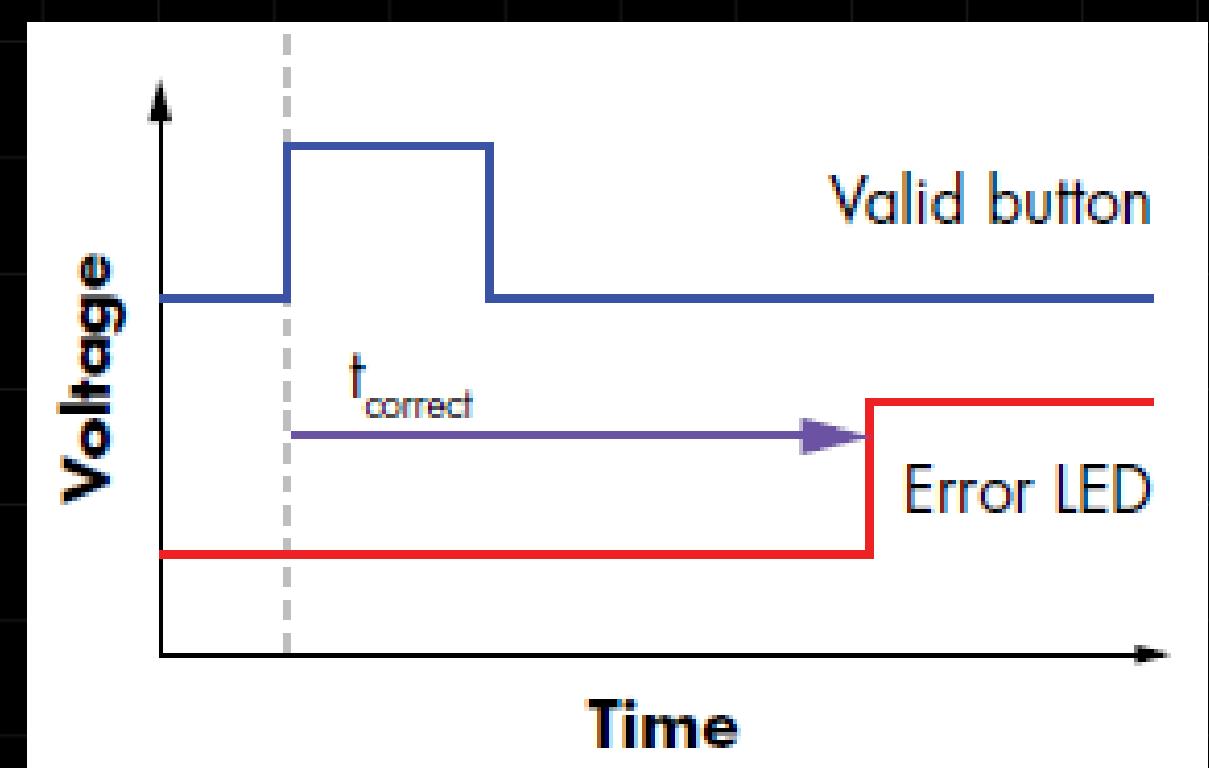
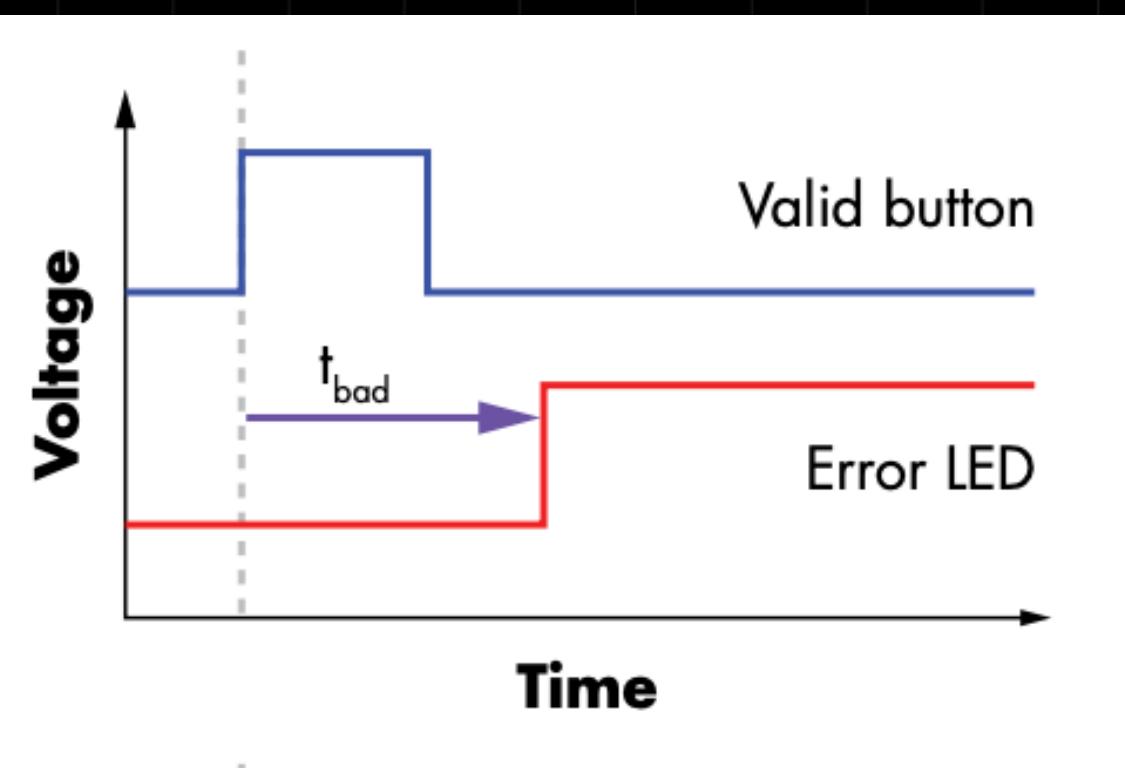
What does this code do?



```
int checkPassword() {  
    int user_pin[] = {1, 1, 1, 1};  
    int correct_pin[] = {5, 9, 8, 2};  
  
    error_led_off();  
  
    for(int i = 0; i < 4; i++) {  
        user_pin[i] = read_button();  
    }  
  
    while(done_pressed() == 0);  
  
    for(int i = 0; i < 4; i++) {  
        if(user_pin[i] != correct_pin[i]) {  
            error_led_on();  
            return 0;  
        }  
    }  
    return 1;  
}
```



TIMING GRAPHS



GUESS:

0000

first digit itself is wrong

5000

your luck was restricted to 1st digit :)

“Timing is Everything.”

3691



```
1 #include <stdlib.h>
2
3 int checkPassword() {
4     int user_pin[] = {1, 1, 1, 1};
5     int correct_pin[] = {5, 9, 8, 2};
6
7     error_led_off();
8
9     int delay_index = rand() % 4; // Returns a random integer from {0, 1, 2, 3}
10
11    for(int i = 0; i < 4; i++) {
12        user_pin[i] = read_button();
13    }
14
15    while(done_pressed() == 0);
16
17    for(int i = 0; i < 4; i++) {
18
19        if (i == delay_index) { // Adding delay at the chosen index
20            delay_ms(100); // Delay of 100 ms
21        }
22
23        if(user_pin[i] != correct_pin[i]) {
24            error_led_on();
25            return 0;
26        }
27    }
28    return 1;
29 }
```

Random Delay

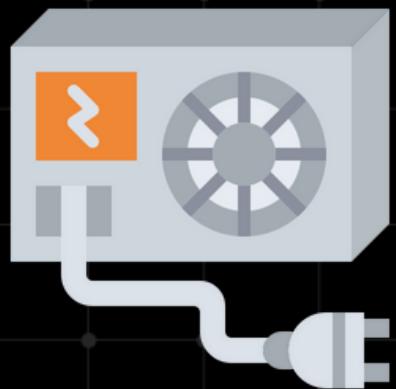
Oh,
Crap.



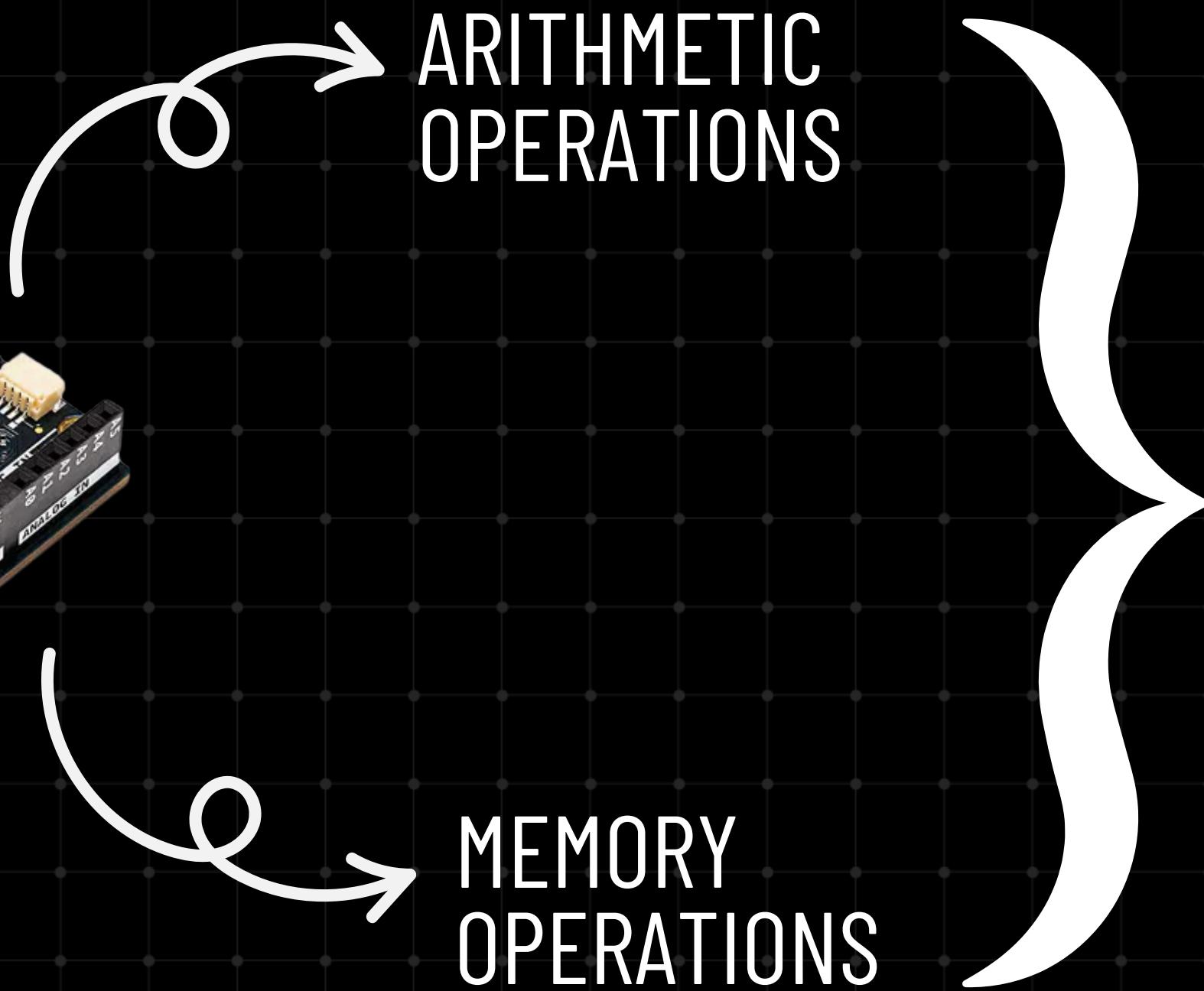
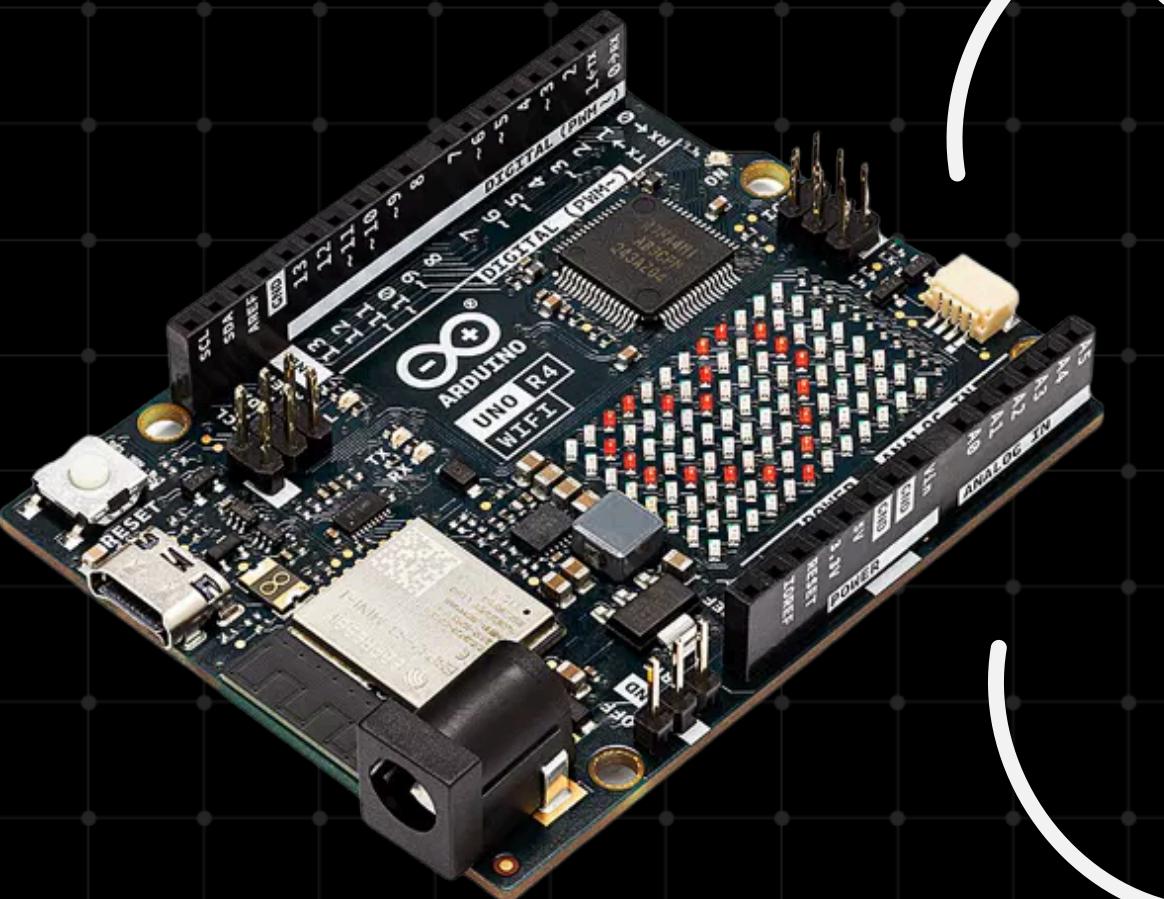
Well, it appears timing isn't good enough.

WOMP WOMP

That brings us to...



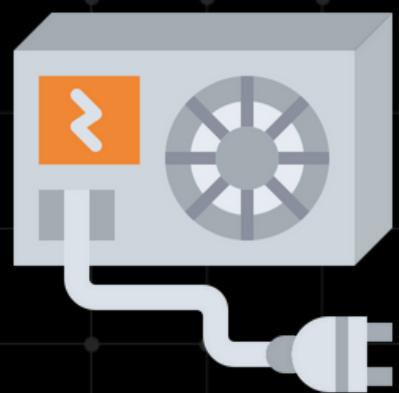
POWER ANALYSIS



ARITHMETIC
OPERATIONS

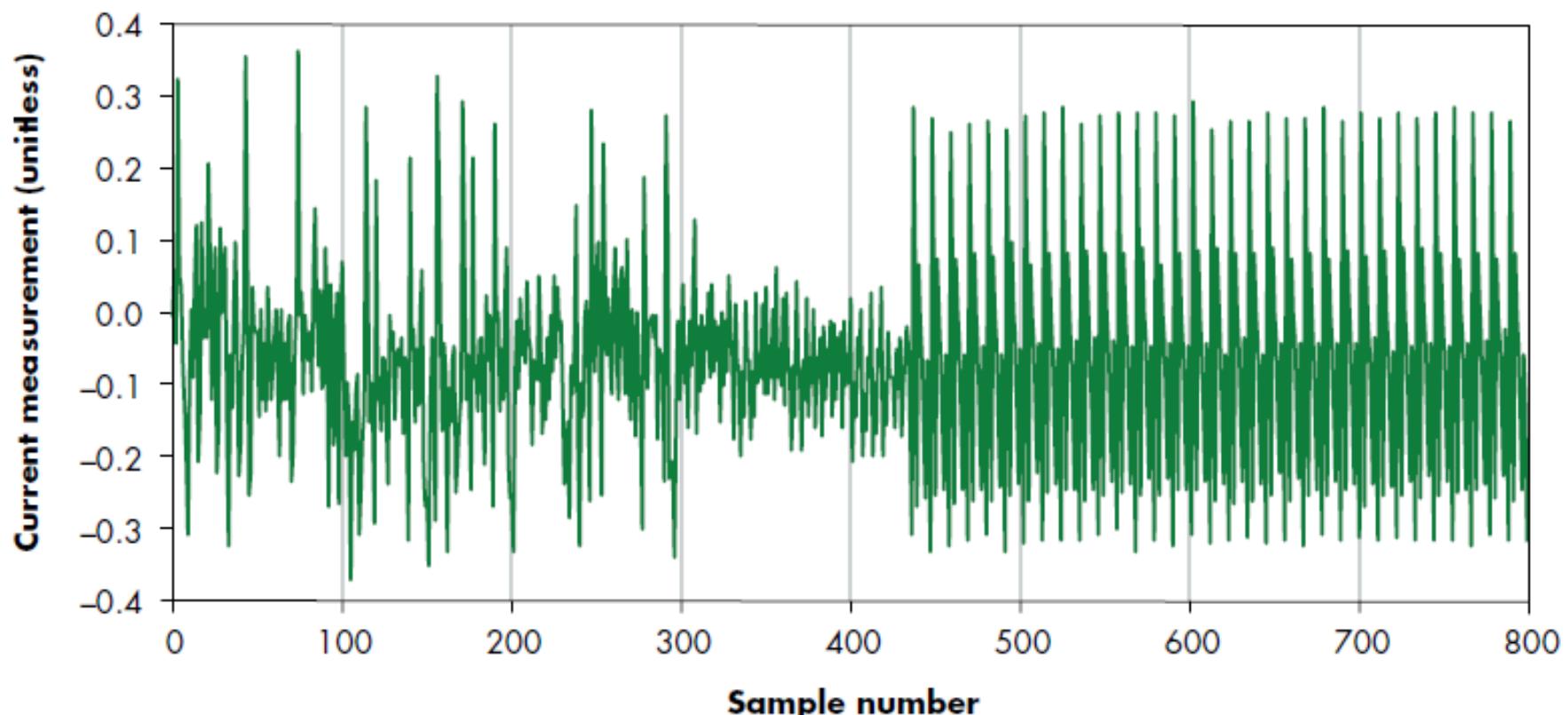
MEMORY
OPERATIONS

DIFFERENT
POWER
CONSUMPTION



POWER ANALYSIS

Capturing power traces



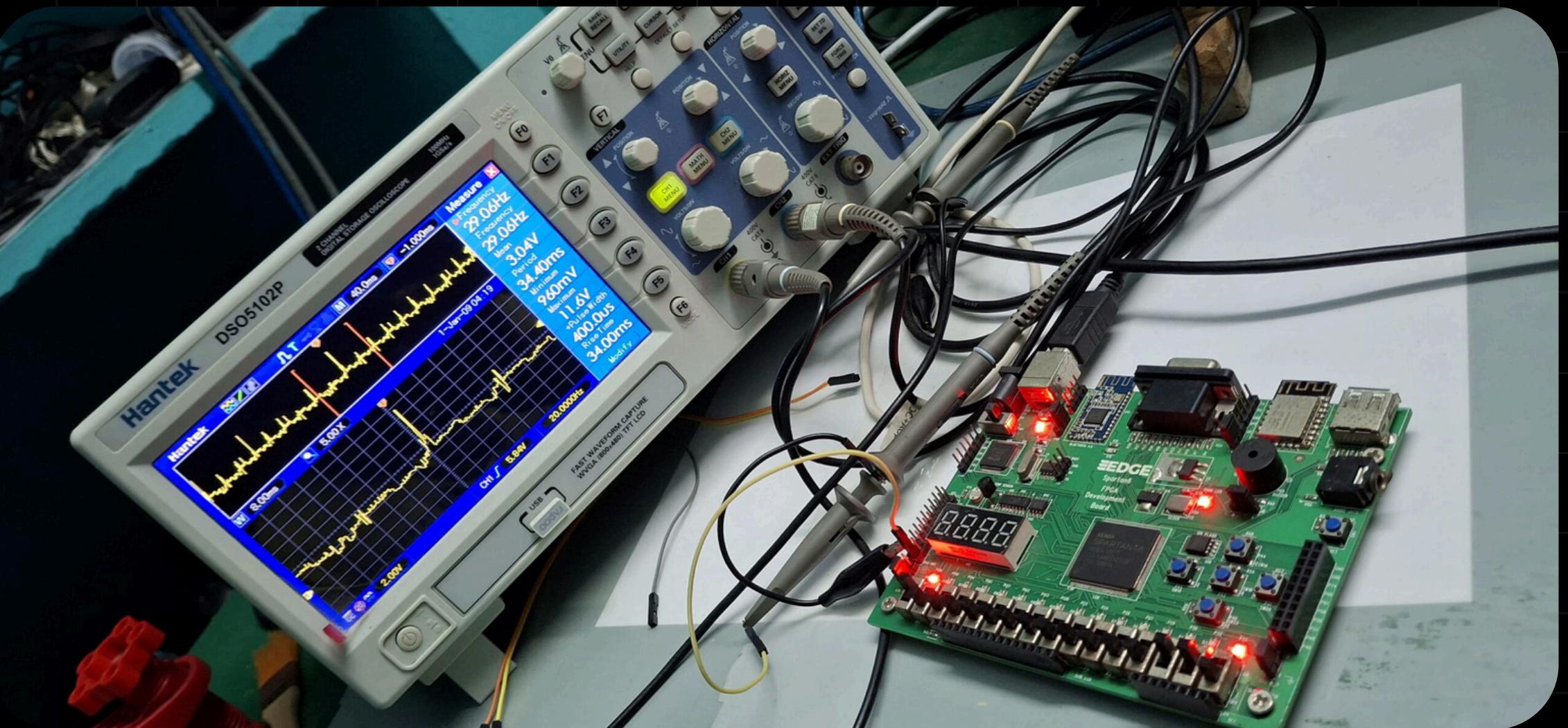
But How do we capture the power traces?

POWER CONSUMED BY THE MICROCONTROLLER OVER TIME

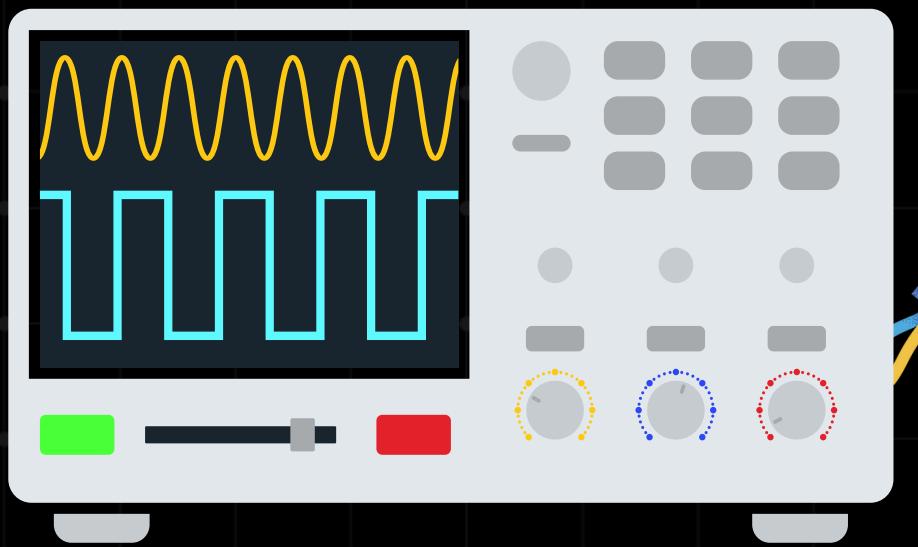
FLASHBACK

nick

OSCILLOSCOPE



FULL SETUP



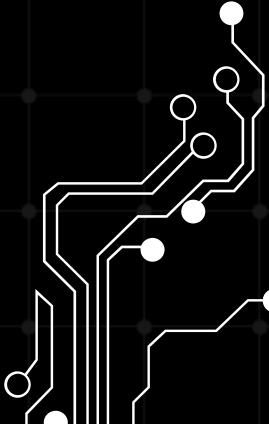
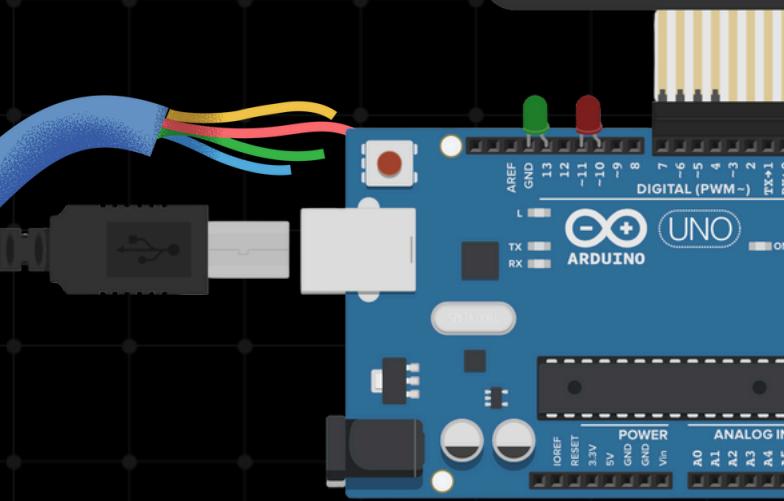
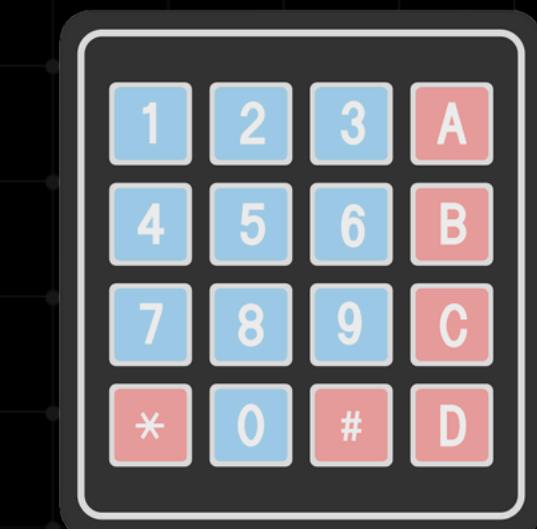
oscilloscope

+

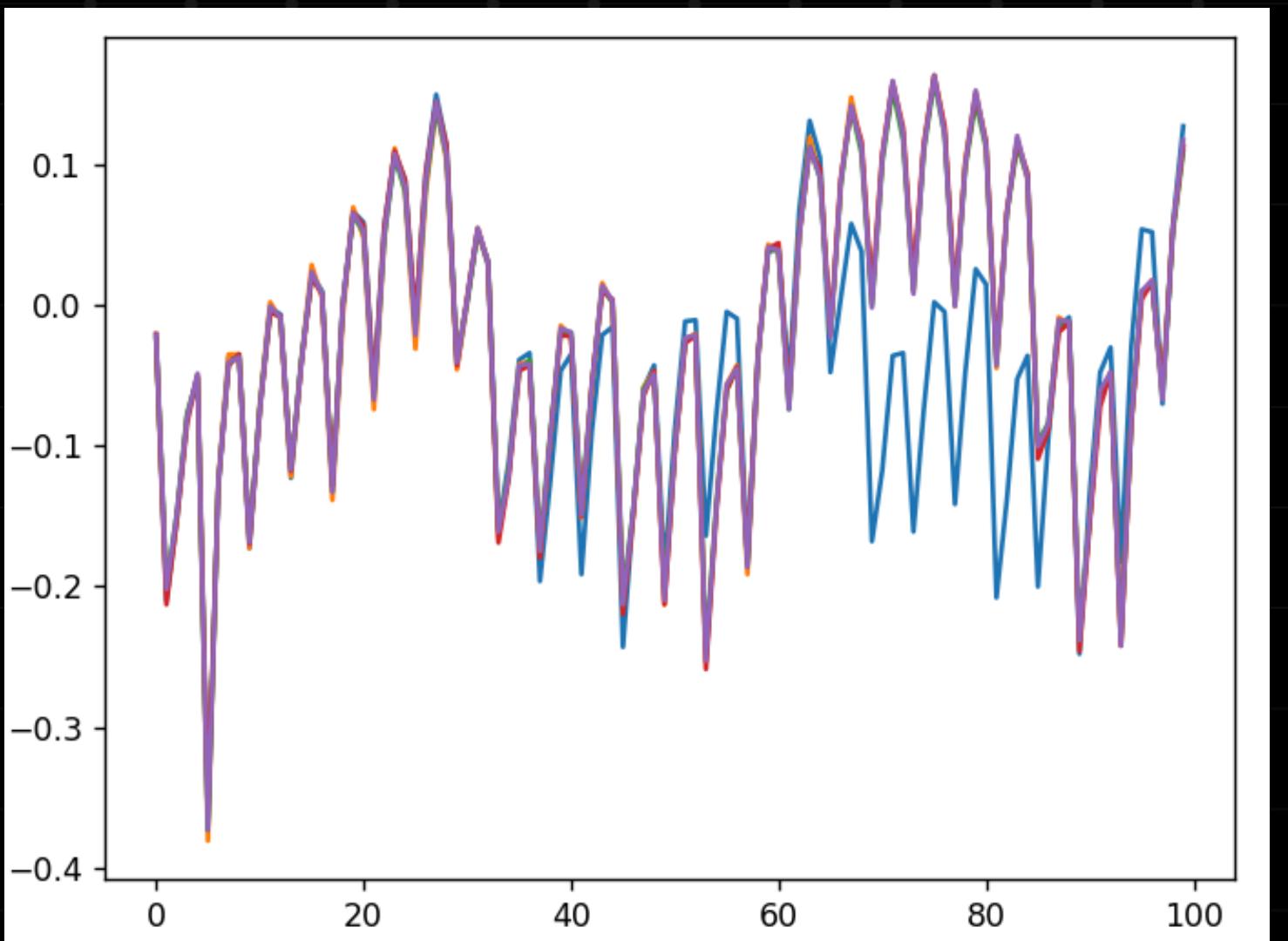
Probes

+

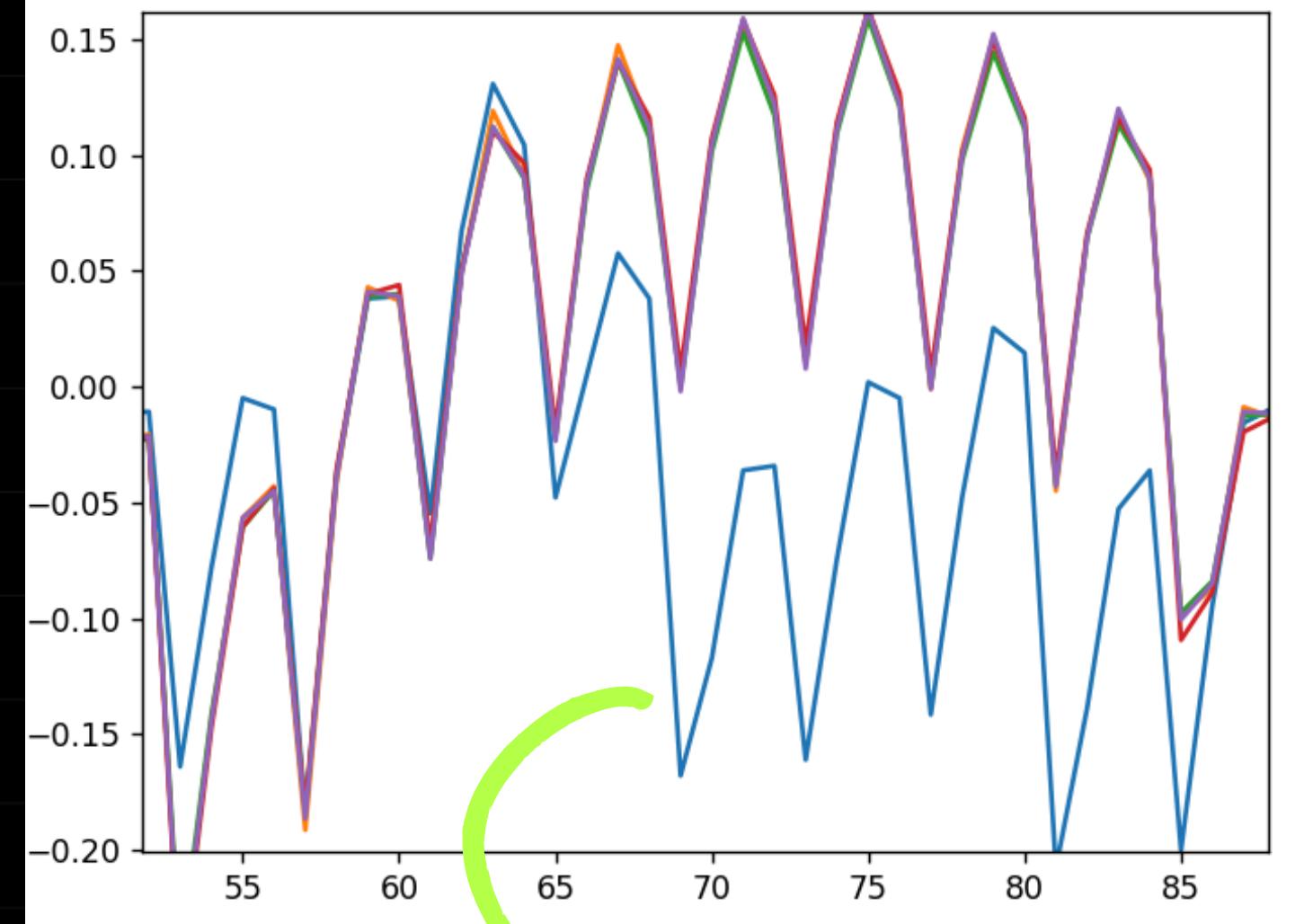
Target device



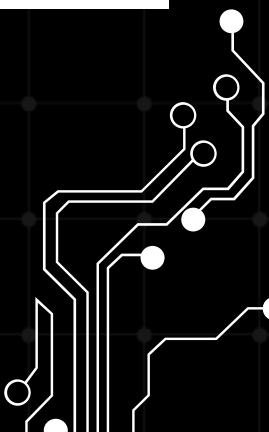
How are they useful?



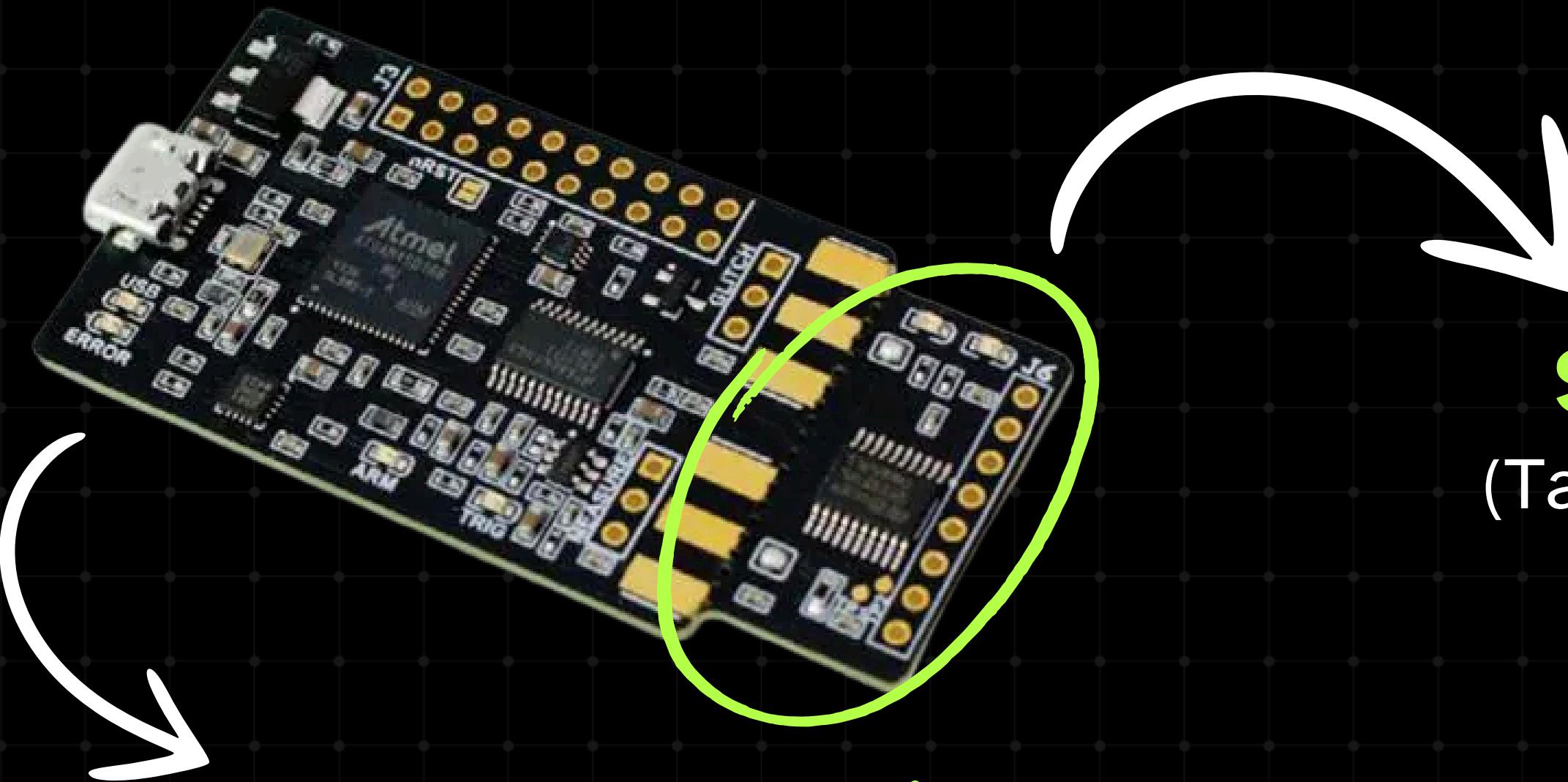
zoom



outlier



CHIPWHISPERER-NANO

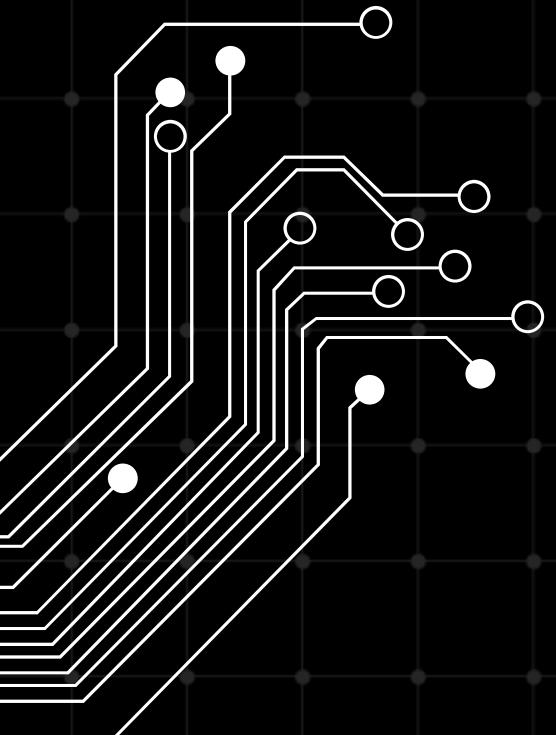


STM32
(Target Device)

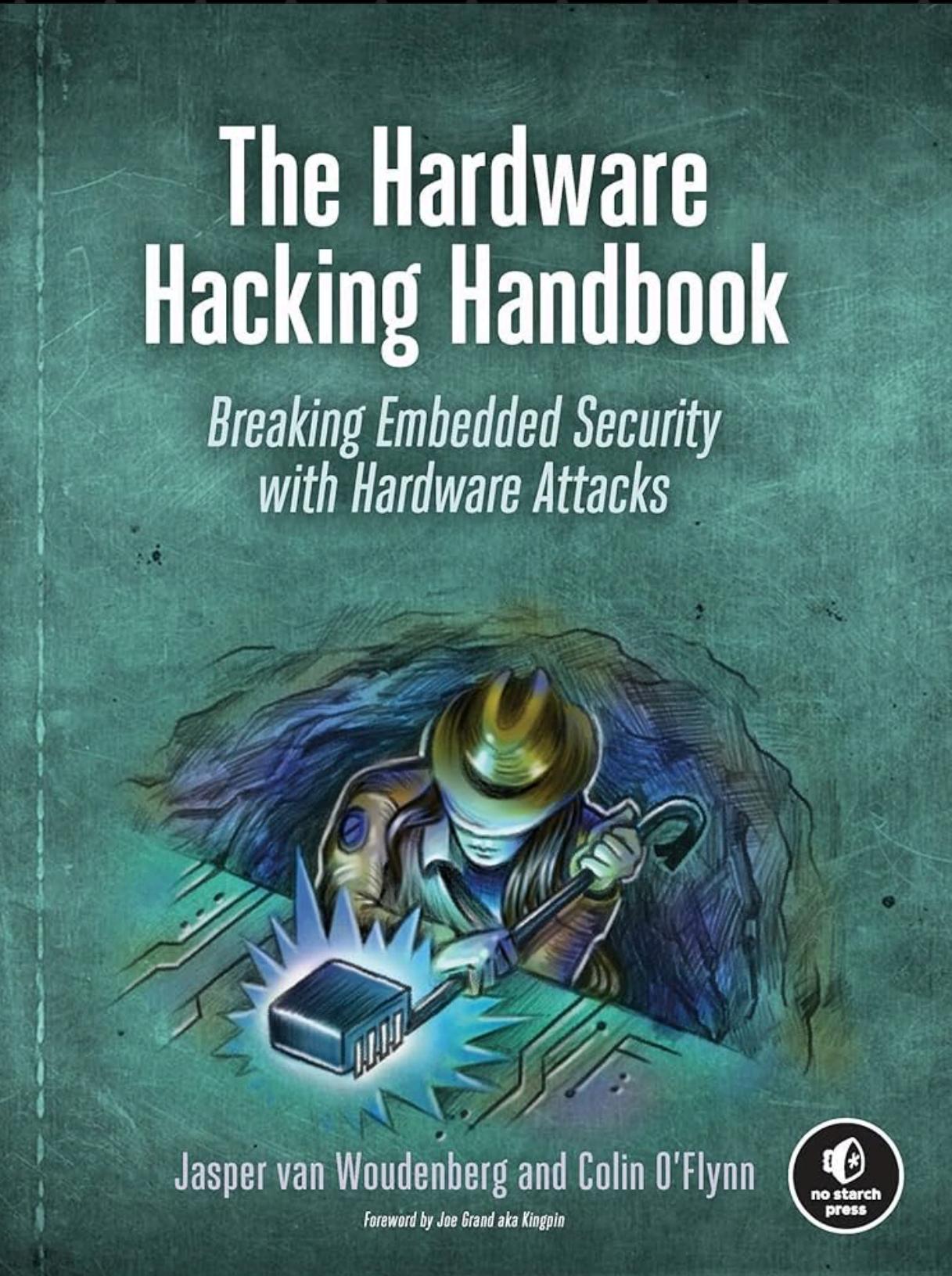
Measurement Section, which has a
microcontroller for sampling and **ADC** for
observing the power traces

NOTEBOOK LINK

tinyurl.com/2x5xe5j2



REFERENCES





HardHack - Day2

WhatsApp group

