Side-channel attacks on high-security electronic safe locks

DEF CON 24

plore@tuta.io

Agenda

- Background on electronic safe locks
- Cracking S&G 6120 safe lock
  - Recovering keycode using power analysis
- Cracking S&G Titan PivotBolt safe lock
  - Recovering keycode using timing attack
  - Defeating incorrect-code lockout

Background – Electronic safe locks

Image: ellenm1 on Flickr / CC BY-NC

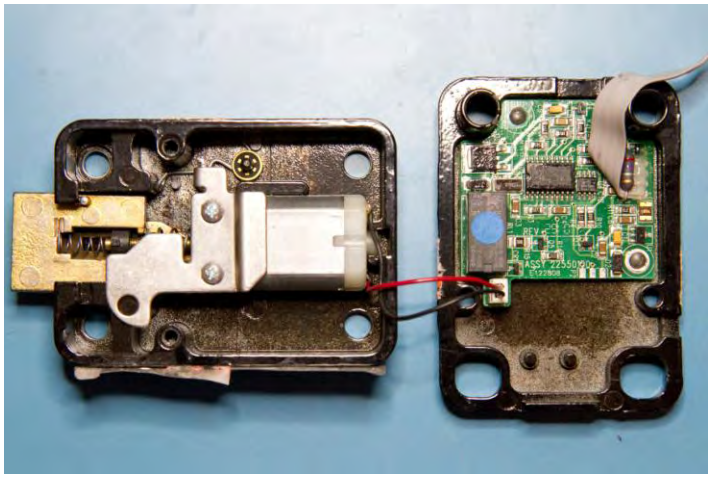
Background – Electronic safe locks

- Safe lock certification
  - UL Type 1 High-security electronic lock
  - Many others
- Out of scope: cheap, non-certified locks
  - Many of these can be easily brute-forced
  - Some can be “spiked” (bolt motor driven directly)
  - Some can be bypassed mechanically (see, e.g., [2] or [3])

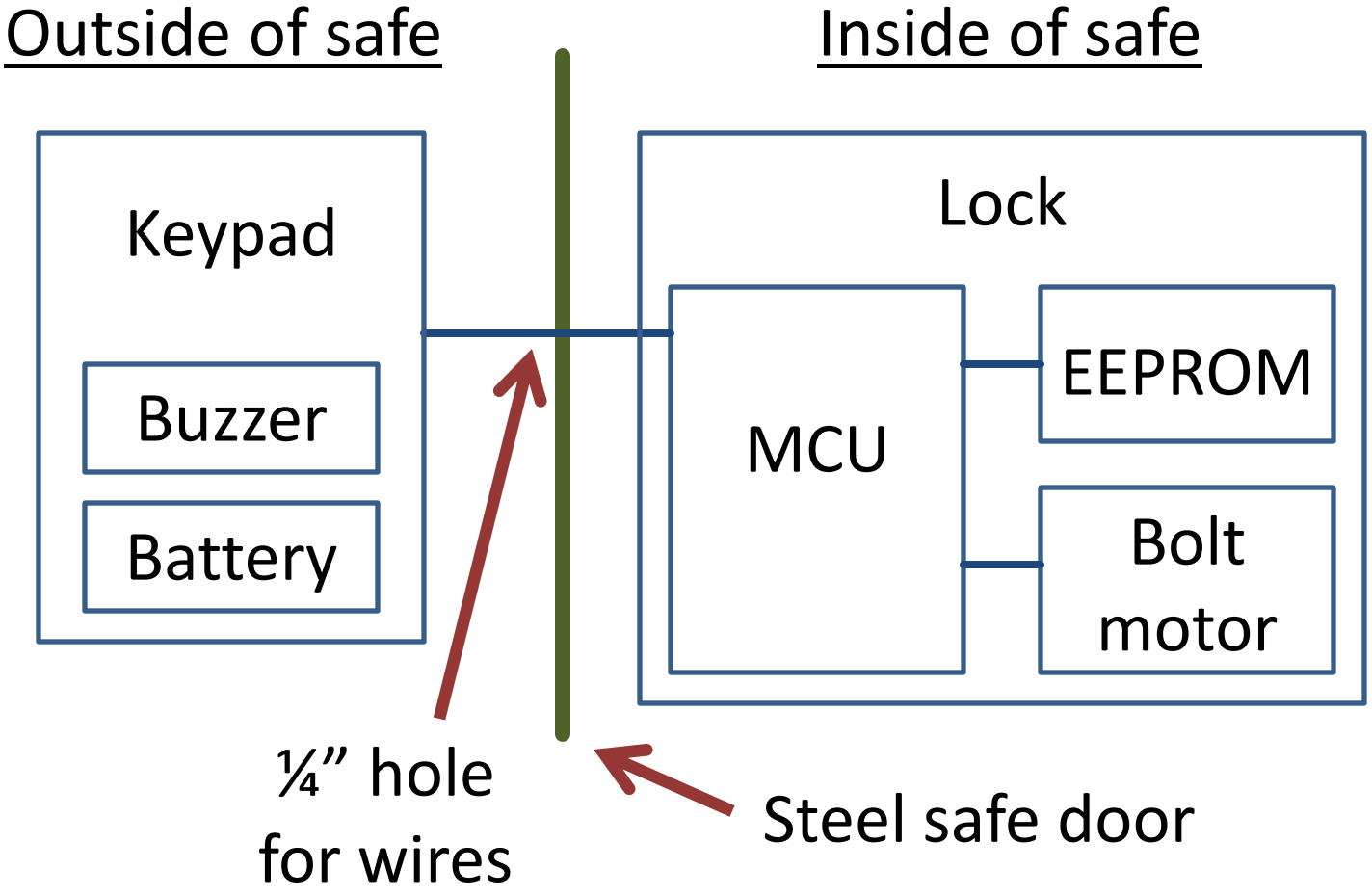
Agenda

- Background on electronic safe locks
- Cracking S&G 6120 safe lock
  - Recovering keycode using power analysis
- Cracking S&G Titan PivotBolt safe lock
  - Recovering keycode using timing attack
  - Defeating incorrect-code lockout

Sargent & Greenleaf 6120-332



6120 – System model



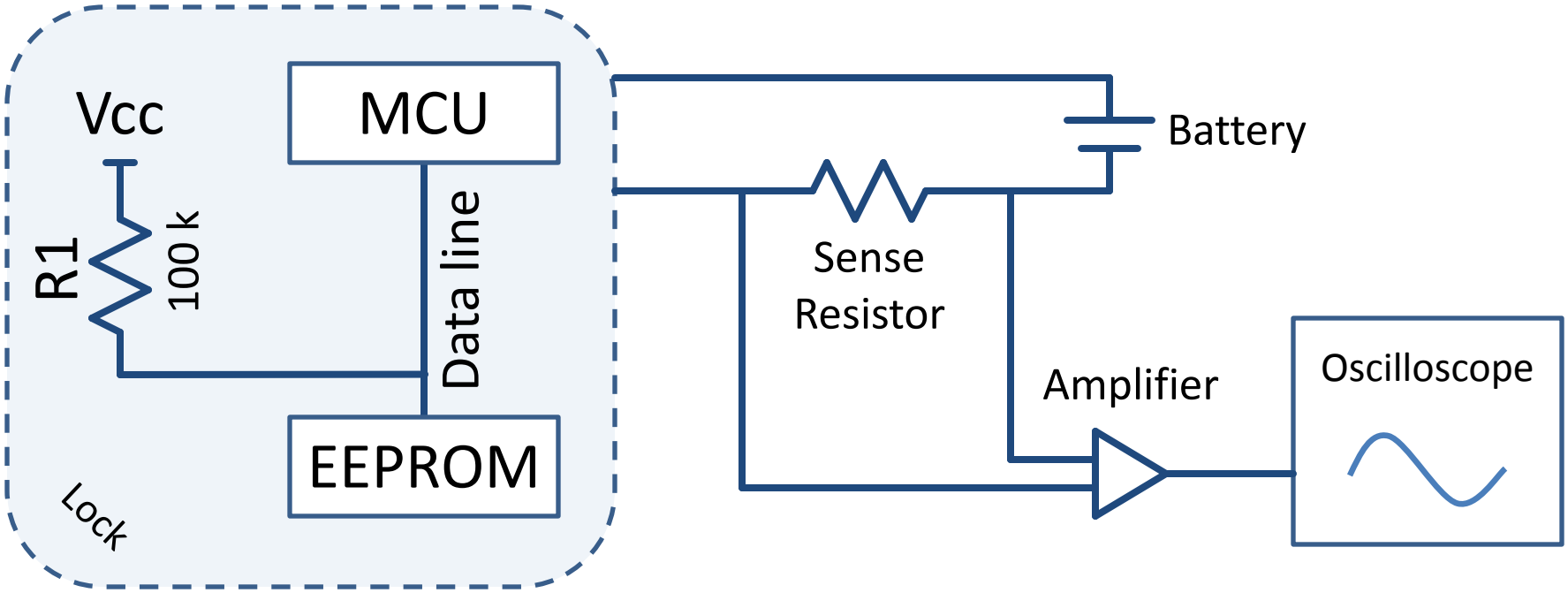
6120 – Design

- Keycodes stored in the clear in EEPROM
- MCU reads/writes EEPROM via 3-wire serial
  - "Microwire" interface (similar to SPI)
- Nice and slow
  - EEPROM to MCU ~1.5 kbit/s
  - Hundreds of milliseconds to read all data
- Lock reads all keycodes out of EEPROM on every attempt

6120 – Vulnerability

- Susceptible to power analysis
- Keycode bit values change amount of current consumed during EEPROM read-out
- Translate current changes into key values
- Enter key values on keypad
- Zero modification required
- Zero evidence of tampering left behind
  - Covert entry

6120 – Circuit model

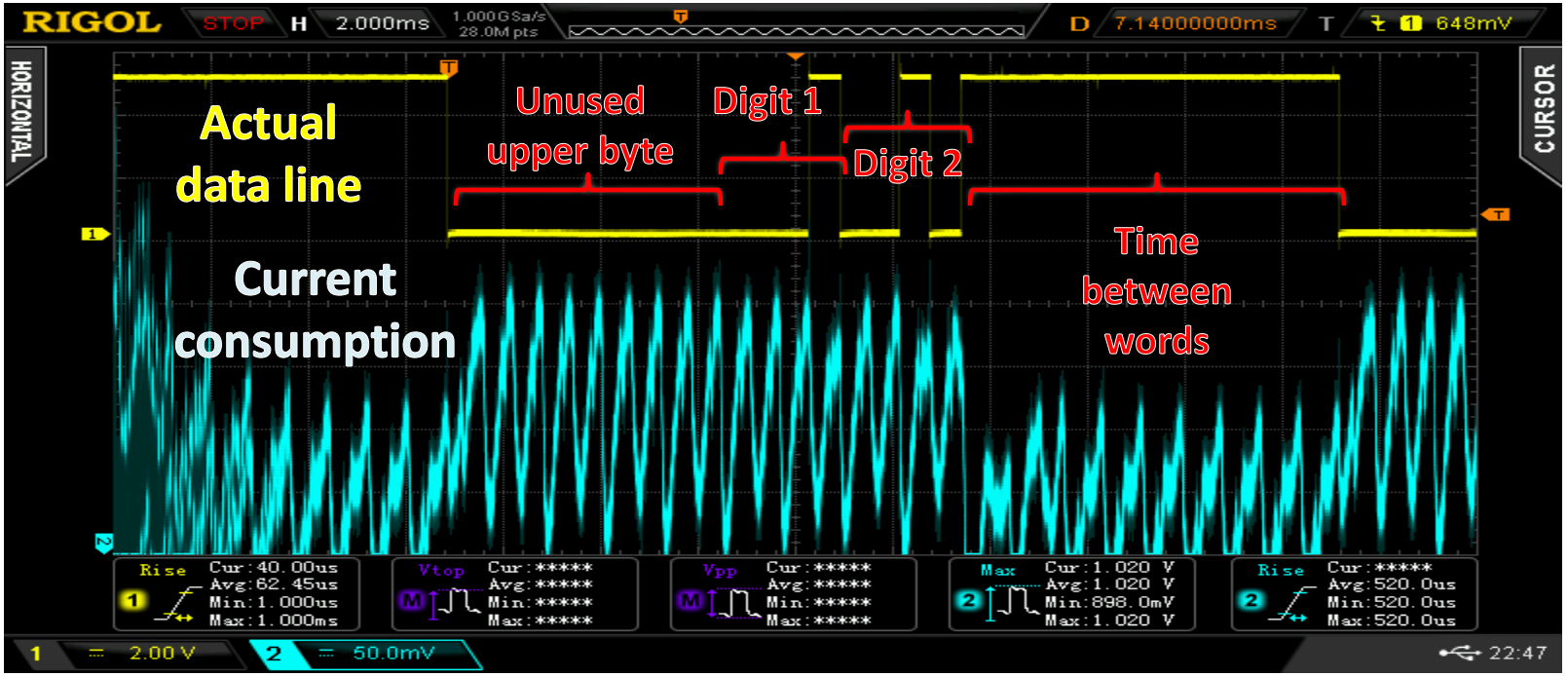


Bit value	Data line volts	Volts *across* R1	Current through R1
1	5 volts	0 volts	0 μA
0	0 volts	5 volts	50 μA

∴ Higher current consumption means the bit being read from EEPROM is a 0, and a lower current means the bit is 1

6120 – Full scope trace

- 1 nibble per keycode digit
- Only lower byte in each EEPROM word is used
- Upper byte always 0x00

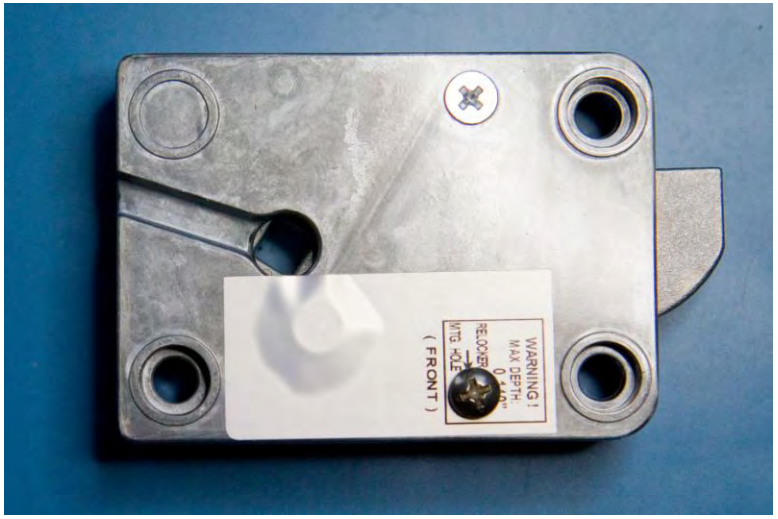


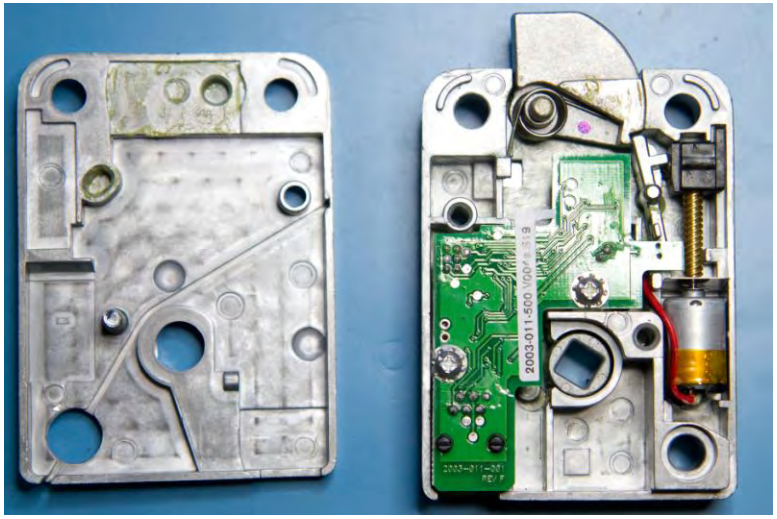
6120 – Demo

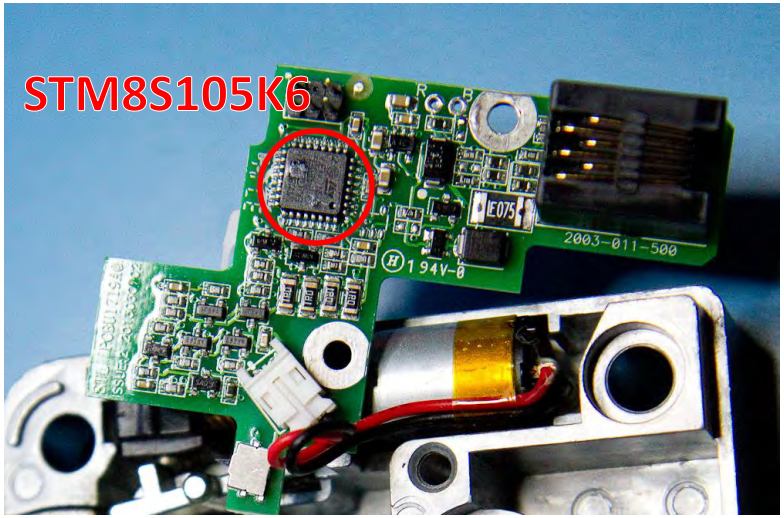
Agenda

- Background on electronic safe locks
- Cracking S&G 6120 safe lock
  - Recovering keycode using power analysis
- Cracking S&G Titan PivotBolt safe lock
  - Recovering keycode using timing attack
  - Defeating incorrect-code lockout

S&G Titan PivotBolt







Titan – Software design

- Keycodes stored in EEPROM within MCU
- Supports 10 keycodes
- 10-minute lockout after 5 incorrect codes in a row
  - Persists across power removal
  - Failed-attempt count stored in EEPROM

Titan – Timing attack

- Entire six-digit keypad sequence is captured before starting comparison to key from EEPROM
- Pseudocode of lock FW keycode comparison:

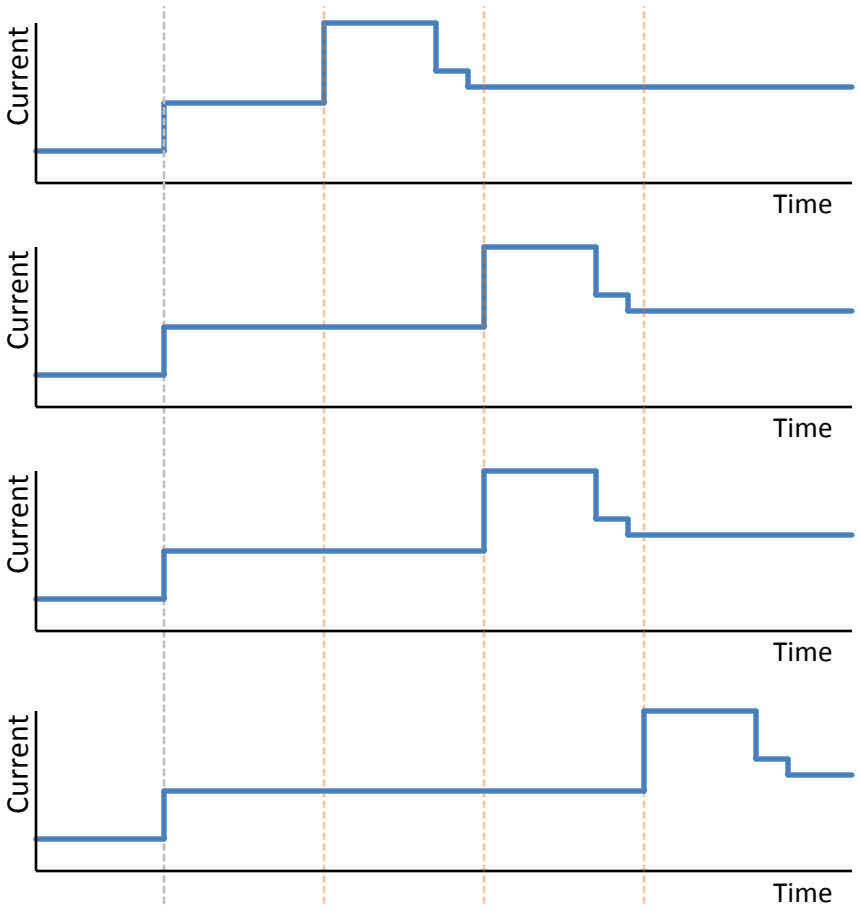
```
bool check_code(int enteredCode[6], int actualCode[6])
{
    for (int digit = 0; digit < 6; digit++)
        if (enteredCode[digit] != actualCode[digit])
            return false;

    return true;
}
```

Each iteration takes another 28 μs

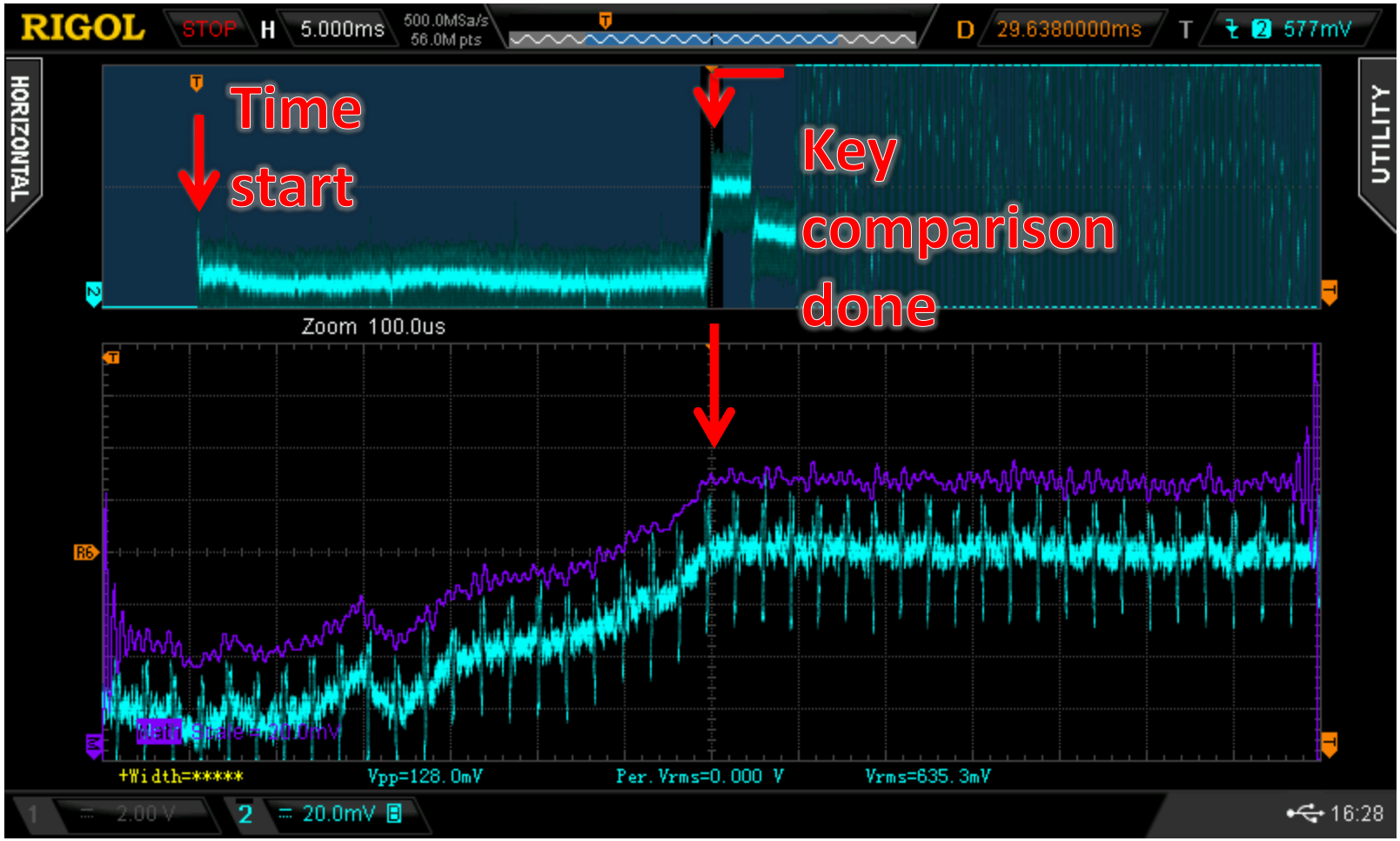
Titan – Timing attack

Suppose that the actual code is 908437

Code tried	Correct run length	Current trace
123456 ↑ Wrong	0	
923456 ↑ Wrong	1	
913456 ↑ Wrong	1	
903456 ↑ Wrong	2	

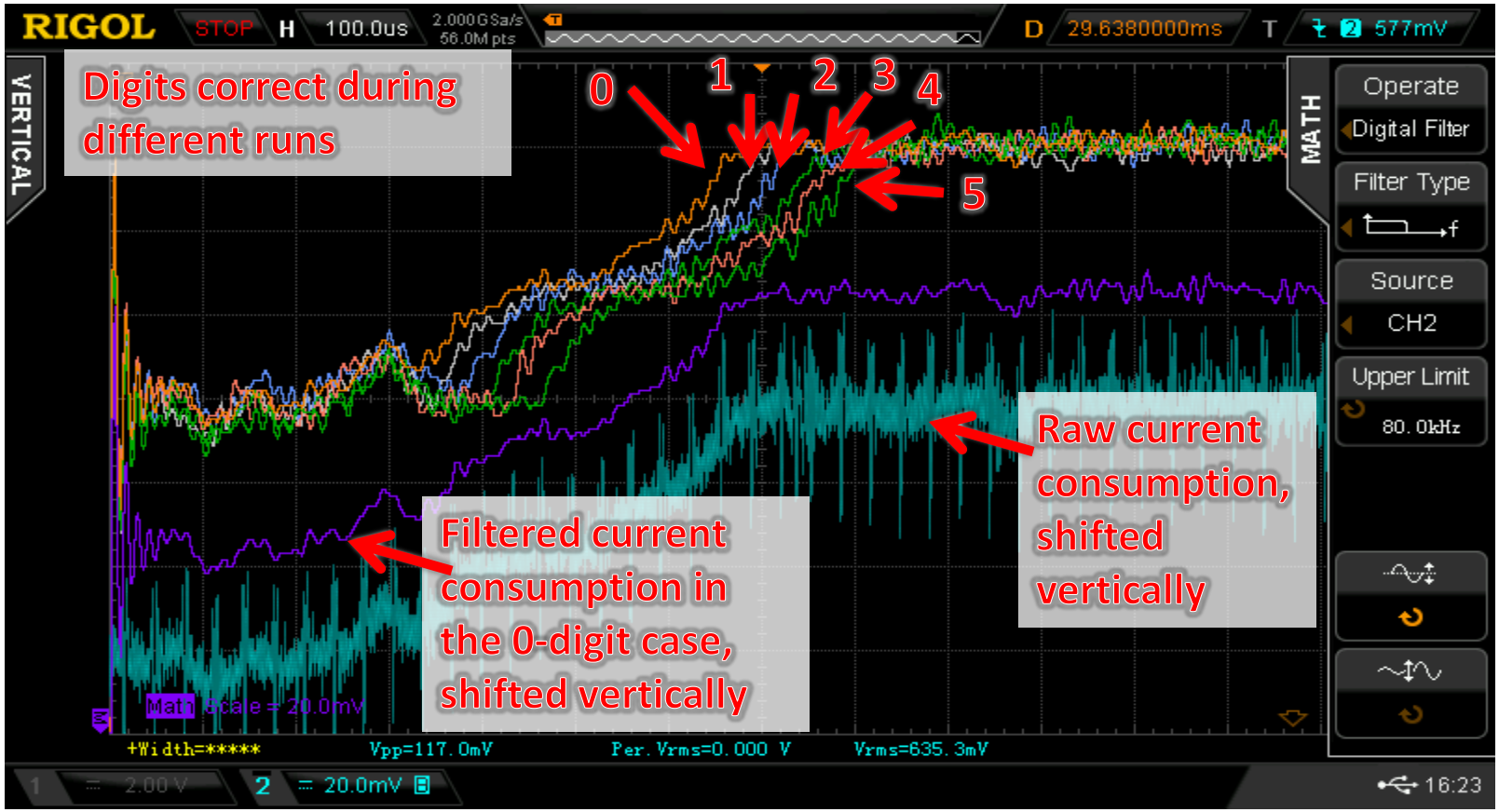
Titan – Timing attack

- Current consumption markers for timing delta



Titan – Timing attack

- The more digits you have correct, the more delayed the current-consumption rise



Titan – Timing attack

- Attack algorithm:
  - Try keycode starting with 0
    - Remaining five key digits don't-care
  - Watch for timing signs showing trial digit match/mismatch
  - If mismatch, try again with keycode starting with 1
    - Retry with increasingly high digit values (2, 3, 4, etc.) until "match" signature encountered (i.e., 28 μs longer delay)
  - Once first digit in keycode discovered, repeat for second, third, fourth, fifth digit
  - Sixth digit is a special case (brute force the 10 possibilities)
- Reduces worst-case attempt count from 1,000,000 to as few as 60

Titan – Lockout defeat

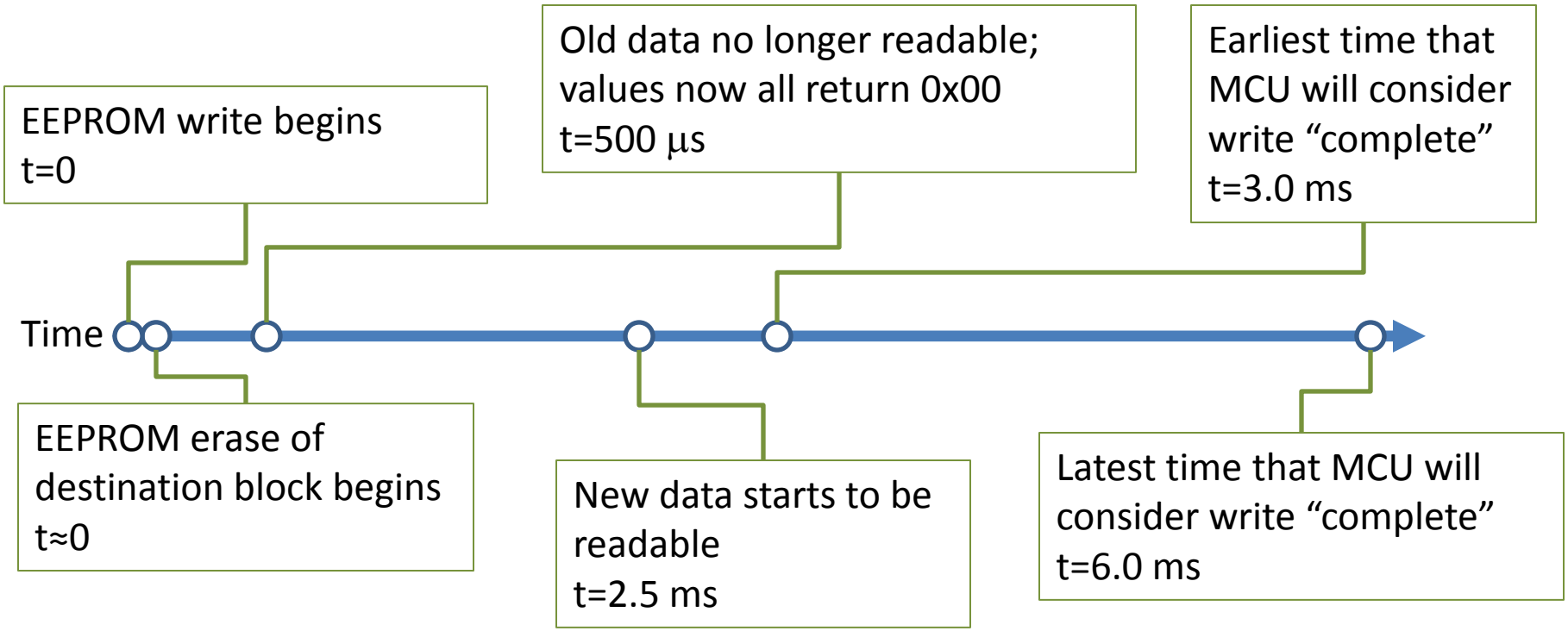
- Normally, 5 incorrect codes in a row leads to a 10-minute penalty lockout period
  - New attempts during lockout are refused
  - Penalty goes back to 10 minutes if power removed
- Incorrect code count tracked in EEPROM
- One of two goals:
  - Prevent increment of failure counter, or:
  - Be able to reset failure counter

Titan – EEPROM write timeline

How EEPROM in STM8 behaves after starting a byte-size write

Initial conditions:

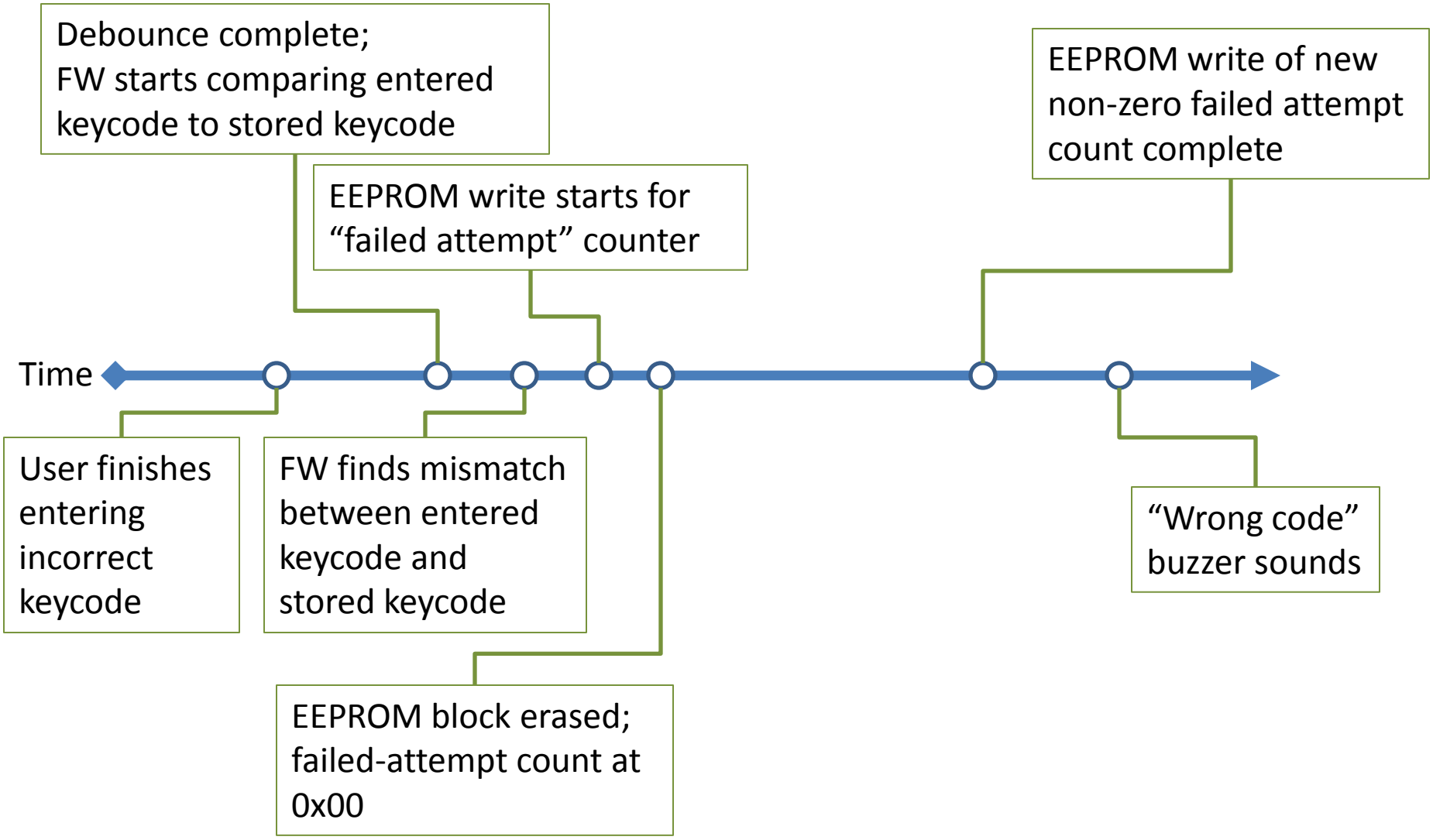
MCU V_{dd} = 5v
MCU clock = 2 MHz
Destination in EEPROM has existing data (i.e., not 0x00)



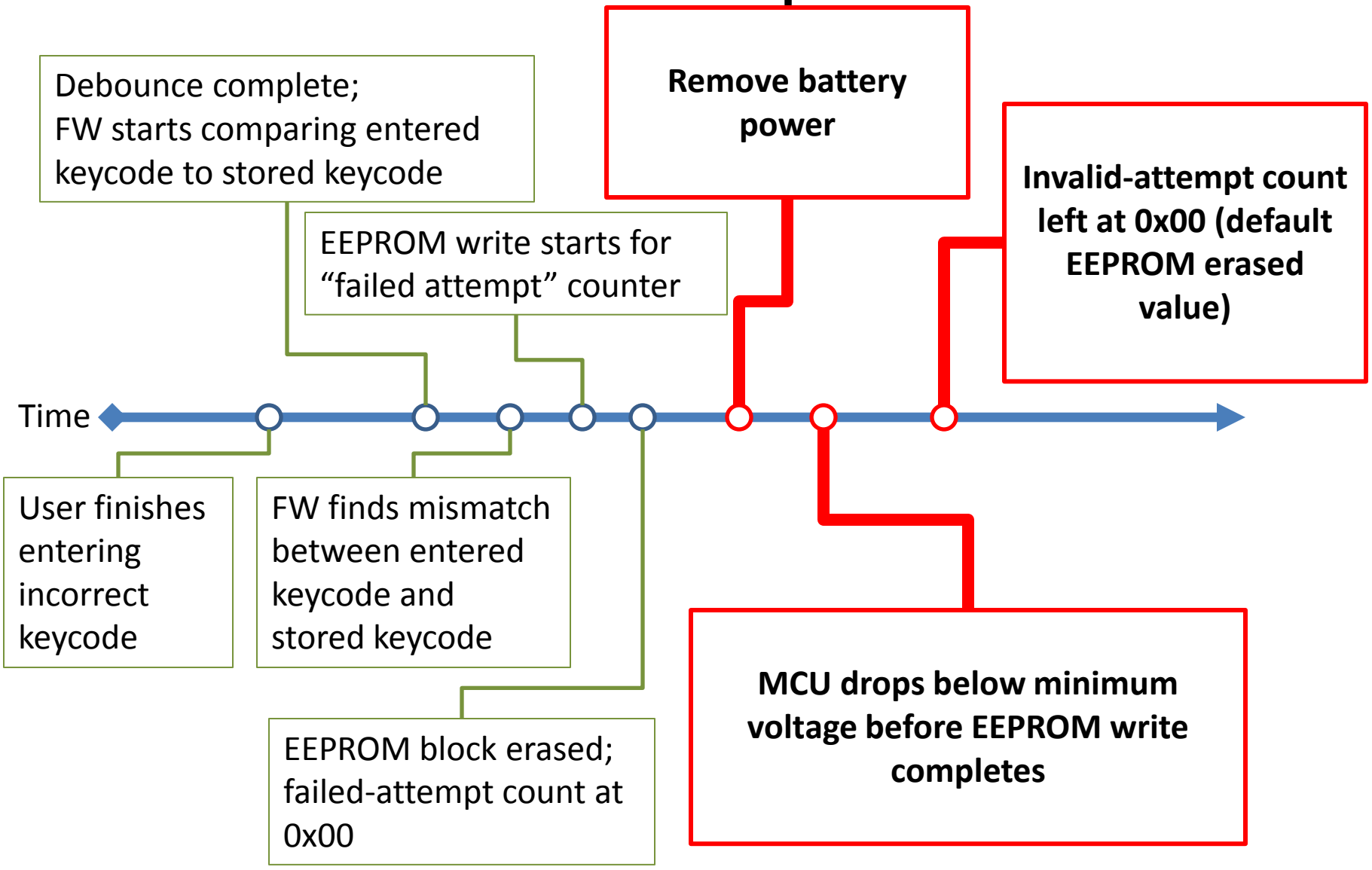
Titan – Lockout defeat

- Measured EEPROM behavior when power cut
  - Block already erased
    - 500 μs (or less) to commit new data
  - Existing data in block
    - About 500 μs from start of cycle until old data no longer readable and bytes return 0x00
    - About 3 ms from start of cycle until new data becomes persistent

Titan – Normal wrong code



Titan – Lockout prevented



Support hardware – Custom PCB

- Microammeter
  - Low-side current sense for simplicity
  - Gain: 40 dB
  - Low-pass filter (second-order, f_c=25 kHz)
- Power control
  - Quickly apply or remove power to/from lock
- Keypress simulator
  - Use DAC and buffer to provide voltages that simulate keys being pressed on the keypad

Titan – Automated code recovery

- Runs on external MCU (STM32L476G)
- Uses functionality from the custom PCB
- Sends keycodes in sequence during search
- Measures time deltas to infer correct values
- Modulates lock power to avoid lockout
- Outputs results

Titan – Demo

Conclusions

- Would I still buy/use an electronic safe lock?
  - Yes! (But probably not the 6120)
- Burglars aren't going to bother with this
  - They'll use the saw or crowbar from your garage

Image: HomeSpotHQ on Flickr / CC BY

Feel free to email me:

plore@tuta.io

Backup slides

Background – Electronic safe locks

- Opening a lock
  - User enters code on keypad
  - Microcontroller (MCU) checks code
  - MCU drives motor to free bolt if correct

Background – Electronic safe locks

- All logic resides inside safe
- Only keypad and battery are outside safe
- Connection is via wires through a small hole in the door metal
- Hardened steel plate in lock
- No direct access to the lock PCB possible

Background – Side channel attack

- Side channel attack
  - Gaining knowledge about the state of a device through unintentional information leakage
- Attacks used in this talk
  - Power analysis
  - Timing attack
- And, a related concept
  - Forcing a system into a particular state using unexpected inputs (in this case, removing power)

S&G 6120

- Sargent & Greenleaf 6120-332 safe lock
  - UL listed Type 1 high-security electronic lock
  - Still being produced (as of at least late 2015)
  - Designed and certified ca. 1994
  - ST62T25C microcontroller (ST)
  - 93LC46B serial EEPROM (Microchip)
  - 9v alkaline battery located in external keypad
  - S&G is a large, well-respected lock manufacturer

6120 – MCU

ST6215C/ST6225C

8-BIT MCUs WITH A/D CONVERTER,
TWO TIMERS, OSCILLATOR SAFEGUARD & SAFE RESET

- **Memories**
  - 2K or 4K bytes Program memory (OTP, EPROM, FASTROM or ROM) with read-out protection
  - 64 bytes RAM
- **Clock, Reset and Supply Management**
  - Enhanced reset system
  - Low Voltage Detector (LVD) for Safe Reset
  - Clock sources: crystal/ceramic resonator or RC network, external clock, backup oscillator (LFAO)
  - Oscillator Safeguard (OSG)
  - 2 Power Saving Modes: Wait and Stop
- **Interrupt Management**
  - 4 interrupt vectors plus NMI and RESET
  - 20 external interrupt lines (on 2 vectors)
  - 1 external non-interrupt line
- **20 I/O Ports**
  - 20 multifunctional bidirectional I/O lines
  - 16 alternate function lines
  - 4 high sink outputs (20mA)
- **2 Timers**
  - Configurable watchdog timer
  - 8-bit timer/counter with a 7-bit prescaler
- **Analog Peripheral**
  - 8-bit ADC with 16 input channels
- **Instruction Set**
  - 8-bit data manipulation
  - 40 basic instructions
  - 9 addressing modes
  - Bit manipulation

PDIP28

S028

SS0P28

CDIP28W

(See Section 12.5 for Ordering Information)

- **Development Tools**
  - Full hardware/software development package

6120 – EEPROM

MICROCHIP

93AA46A/B/C, 93LC46A/B/C, 93C46A/B/C

1K Microwire Compatible Serial EEPROM

Device Selection Table

Part Number	Vcc Range	ORG Pin	Word Size	Temp Ranges	Packages
93AA46A	1.8-5.5	No	8-bit	I	P, SN, ST, MS, OT
93AA46B	1.8-5-5	No	16-bit	I	P, SN, ST, MS, OT
93LC46A	2.5-5.5	No	8-bit	I, E	P, SN, ST, MS, OT
93LC46B	2.5-5.5	No	16-bit	I, E	P, SN, ST, MS, OT
93C46A	4.5-5.5	No	8-bit	I, E	P, SN, ST, MS, OT
93C46B	4.5-5.5	No	16-bit	I, E	P, SN, ST, MS, OT
93AA46C	1.8-5.5	Yes	8 or 16-bit	I	P, SN, ST, MS
93LC46C	2.5-5.5	Yes	8 or 16-bit	I, E	P, SN, ST, MS
93C46C	4.5-5.5	Yes	8 or 16-bit	I, E	P, SN, ST, MS

Features

- Low-power CMOS technology
- ORG pin to select word size for '46C version
- 128 x 8-bit organization 'A' ver. devices (no ORG)
- 64 x 16-bit organization 'B' ver. devices (no ORG)
- Self-timed ERASE/WRITE cycles (including auto-erase)
- Automatic ERAL before WRAL
- Power-on/off data protection circuitry
- Industry standard 3-wire serial I/O
- Device Status signal (READY/BUSY)
- Sequential READ function
- 1,000,000 E/W cycles
- Data retention > 200 years
- Temperature ranges supported
  - Industrial (I) -40°C to +85°C
  - Automotive (E) -40°C to +125°C

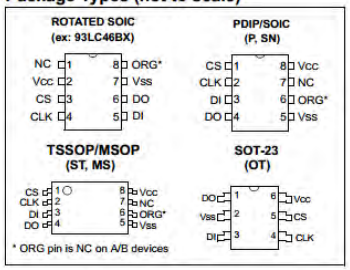
Pin Function Table

Name	Function
CS	Chip Select
CLK	Serial Data Clock
DI	Serial Data Input
DO	Serial Data Output
Vss	Ground
NC	No internal connection
ORG	Memory Configuration
Vcc	Power Supply

Description

The Microchip Technology Inc. 93XX46A/B/C devices are 1K bit low voltage serial Electrically Erasable PROMs (EEPROM). Word-selectable devices such as the 93AA46C, 93LC46C or 93C46C are dependent upon external logic levels driving the ORG pin to set word size. For dedicated 8-bit communication, the 93AA46A, 93LC46A or 93C46A devices are available, while the 93AA46B, 93LC46B and 93C46B devices provide dedicated 16-bit communication. Advanced CMOS technology makes these devices ideal for low power, nonvolatile memory applications. The entire 93XX Series is available in standard packages including 8-lead PDIP and SOIC, and advanced packaging including 8-lead MSOP, 6-lead SOT-23, and 8-lead TSSOP. Pb-free (Pure Matte Sn) finish is also available.

Package Types (not to scale)

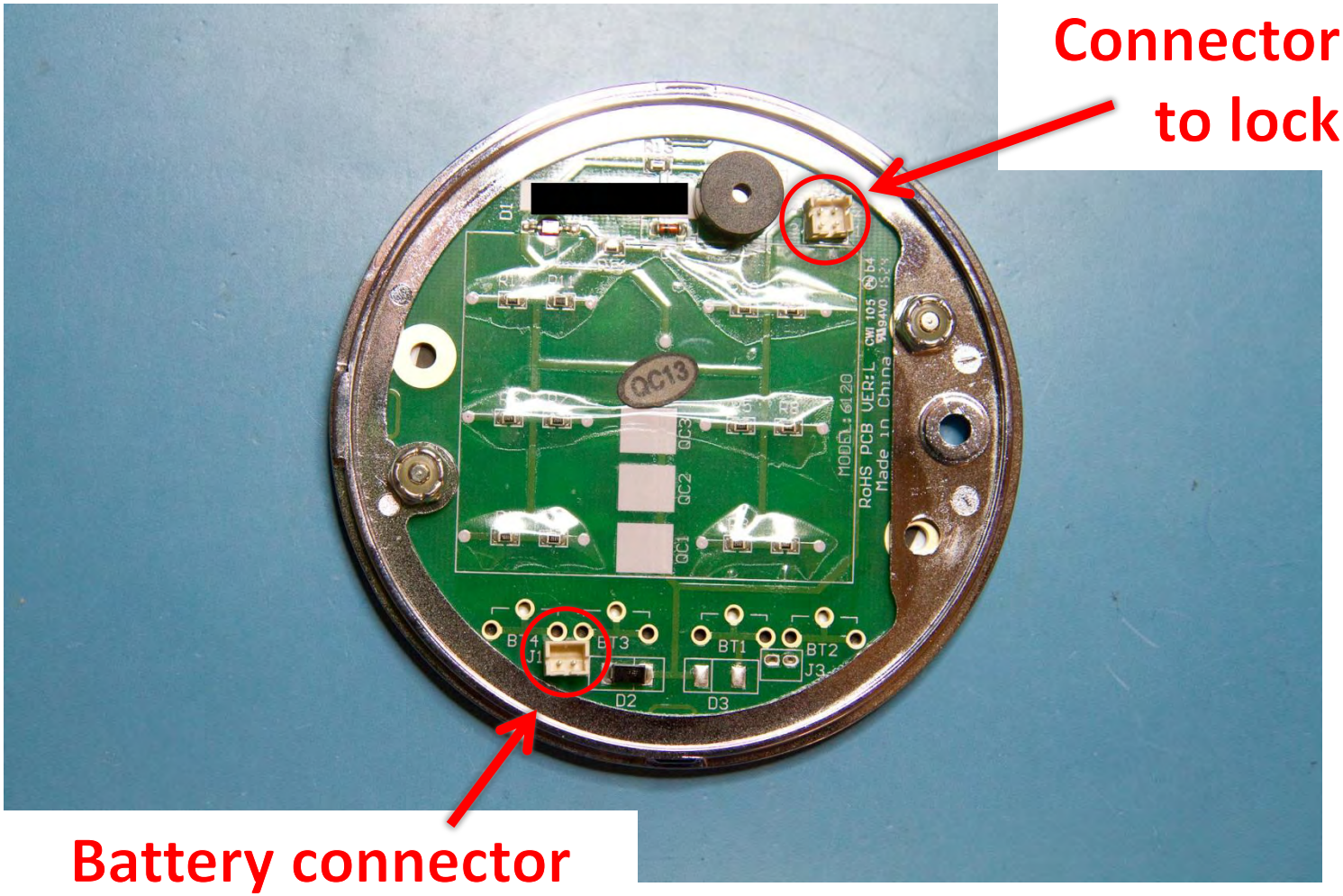


6120 – Keycode storage

Suppose that the actual code is 908437

	EEPROM address	Stored word value	Keycode digits
Keycode 1	0x00	0x 0 0 9 0	9, 0
	0x01	0x 0 0 8 4	8, 4
	0x02	0x 0 0 3 7	3, 7
Keycode 2	0x03	← Start of next keycode	
	.		
	.		
	.		

S&G 6120 (and Titan) Keypad Interior



6120 – Wires from keypad

There are four wires from the keypad to the lock inside the safe:

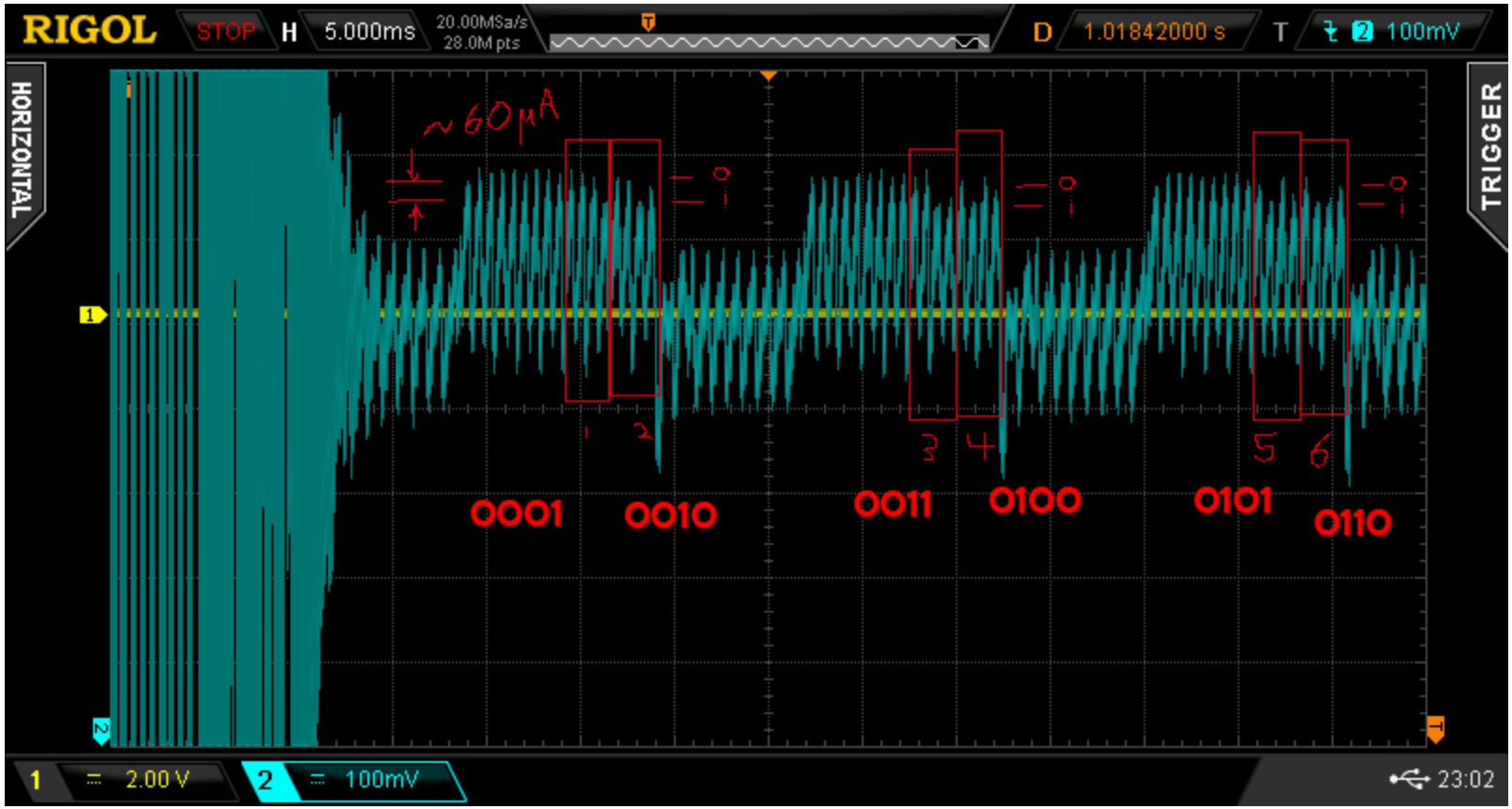
Line	Description
Battery	9v nominal
Ground	Complete circuits are good, right?
Keypress	5v when idle, less depending on key being pressed
Buzzer	Hi-Z when idle, pulled to ground for buzzer/LED

6120 – Actual vs Power Zoomed

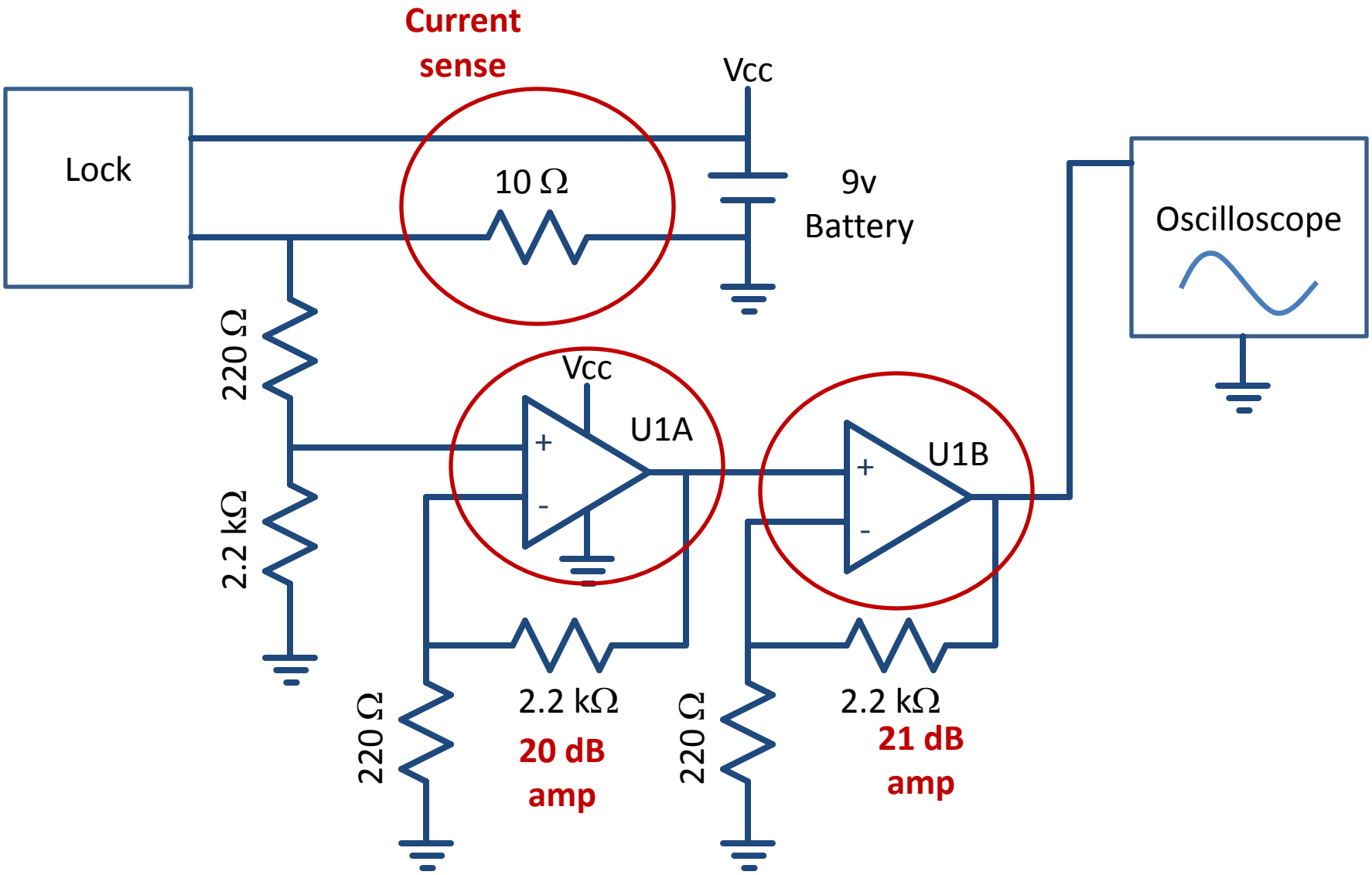
- Yellow: Actual data line between MCU and EEPROM
- Blue: Current into lock (2 μA per mV)

6120 – Annotated trace

- In this case, the keycode is “123456”



6120 – Demo circuit



6120 – Demo

- Only basic equipment required to read code
  - Cheap oscilloscope (e.g., the $60 DDS120)
  - Seven resistors (precision not critical)
  - Basic op-amp (e.g., the 10 MHz GBW LM6132)
  - Breadboard/wires/etc.

6120 – Notes

- Final bit in each word (i.e., LSB for every even keycode digit) is shifted lower in amplitude by about 20 μA regardless of value
- Reading first three words is enough for master keycode
- Remaining words are for additional keycodes
- Failure count written after all codes read out

6120 – Lessons

- Don’t store data in the clear
  - I mean, good lord…

6120 – Lessons

- Store critical data on-chip if possible
  - Harder to probe when analyzing lock
  - Less EM radiation
  - Faster access
  - Possibly smaller current swing

6120 – Lessons

- Use a fast serial bus
  - Simple power analysis is harder at higher speeds due to capacitive and inductive effects
  - Higher speeds could make attack inaccessible to the simple tools shown in the demo

Titan – Hardware

- Motor-driven acme screw to unblock bolt
- STM8S105K6 MCU runs at 2 MHz
- Keypad identical to one with S&G 6120
  - Resistor ladder
  - 9v alkaline battery
- Designed c.a. 2010, currently in production
- UL listed Type 1 high-security electronic lock

Titan – MCU

life.augmented

STM8S105C4/6 STM8S105K4/6 STM8S105S4/6

Access line, 16 MHz STM8S 8-bit MCU, up to 32 Kbyte Flash, integrated EEPROM, 10-bit ADC, timers, UART, SPI, I²C

Datasheet - production data

Features

Core

- 16 MHz advanced STM8 core with Harvard architecture and 3-stage pipeline
- Extended instruction set

Memories

- Program memory: up to 32 Kbyte Flash; data retention 20 years at 55 °C after 10 kcycle
- Data memory: up to 1 Kbyte true data EEPROM; endurance 300 kcycle
- RAM: up to 2 Kbyte

Clock, reset and supply management

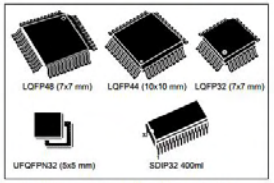
- 2.95 to 5.5 V operating voltage
- Flexible clock control, 4 master clock sources
  - Low power crystal resonator oscillator
  - External clock input
  - Internal, user-trimmable 16 MHz RC
  - Internal low-power 128 kHz RC
- Clock security system with clock monitor
- Power management:
  - Low-power modes (wait, active-halt, halt)
  - Switch-off peripheral clocks individually
- Permanently active, low-consumption power-on and power-down reset

Interrupt management

- Nested interrupt controller with 32 interrupts
- Up to 37 external interrupts on 6 vectors

Timers

- Advanced control timer: 16-bit, 4 CAPCOM channels, 3 complementary outputs, dead-time insertion and flexible synchronization



- 2x16-bit general purpose timer, with 2+3 CAPCOM channels (IC, OC or PWM)
- 8-bit basic timer with 8-bit prescaler
- Auto wake-up timer
- Window watchdog and independent watchdog timers

Communication interfaces

- UART with clock output for synchronous operation, SmartCard, IrDA, LIN master mode
- SPI interface up to 8 Mbit/s
- I2C interface up to 400 kbit/s

Analog to digital converter (ADC)

- 10-bit, ±1 LSB ADC with up to 10 multiplexed channels, scan mode and analog watchdog

I/Os

- Up to 38 I/Os on a 48-pin package including 16 high sink outputs
- Highly robust I/O design, immune against current injection

Unique ID

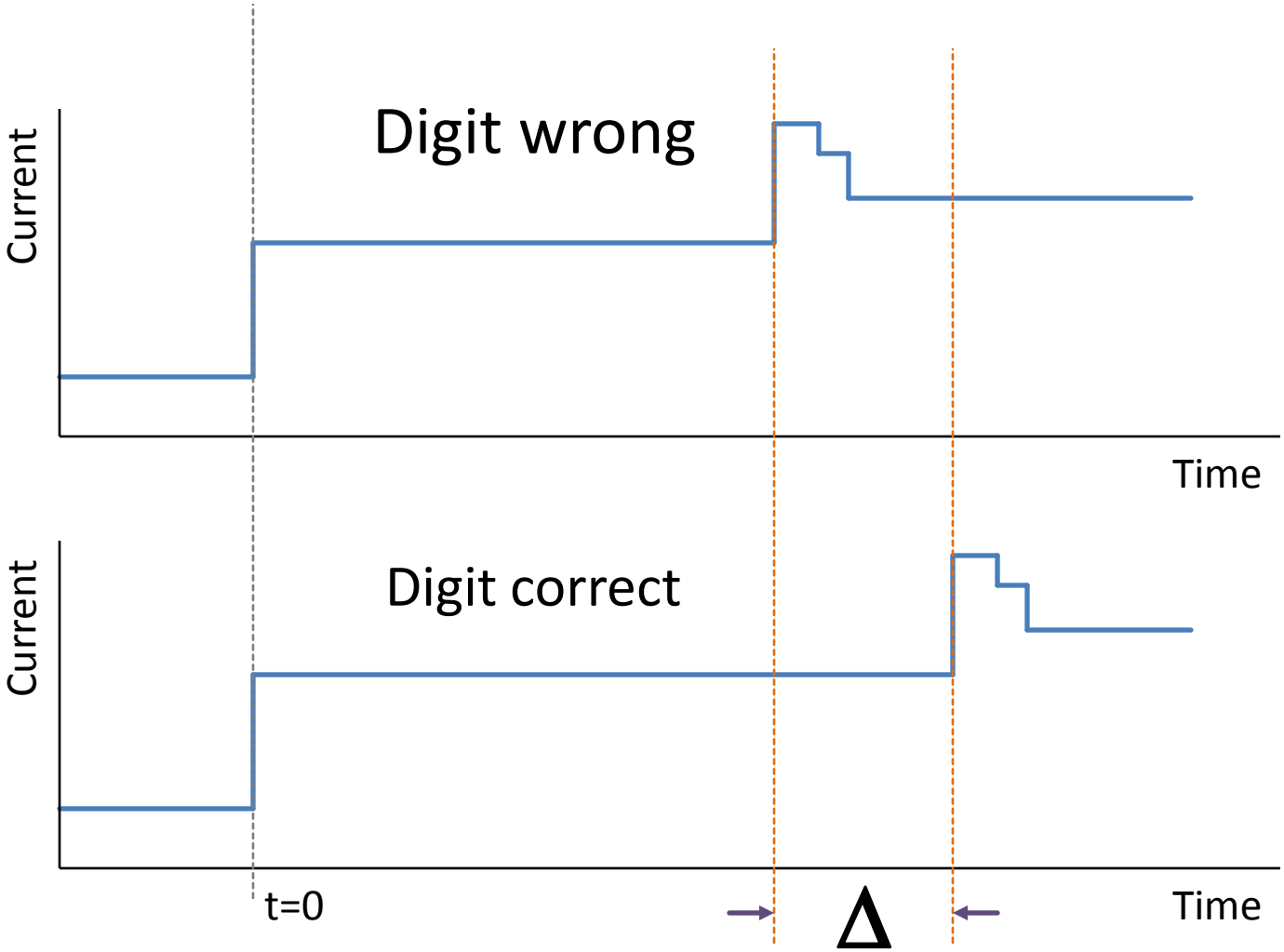
- 96-bit unique key for each device

This is information on a product in full production. www.st.com

Titan – Keypad emulation

- Keypad is resistor ladder hooked to voltage divider with a 20.0 kΩ source leg
  - e.g., “3” is 7.68 kΩ
- Simulate by sending the voltage that the divider would produce for a given key
  - e.g., 7.68 kΩ is 1.40 V
- Lock tolerates voltage error of ±0.10 V
- Debounce time ~30 ms
- Key interval ~120 ms

Titan – Timing attack



Titan – Timing attack

- Power analysis for timing markers
  - Watch current drawn
- Current consumption jumps about 29.6 ms before keycode comparison completes
  - Use this rise as a reference point for timing
  - Reasonably stable time reference (jitter about ±10 μs)
- Keycode comparison takes about 200-300 μs
  - Depends on how many digits before mismatch
- At end of keycode comparison, current rises another 275 μA
  - Determine success/failure based on delay of this rise relative to reference point ≈29.6 ms earlier
  - 28 μs more delay per additional correct digit

Titan – Timing attack

- It’s like in the movies where they get one digit of the electronic lock’s code at a time
  - …and the others are all changing rapidly

Image: The Thomas Crown Affair (1999)

Titan – Timing attack

- Noise
  - Jitter in ADC sampling times
  - Jitter in lock clock
  - Noise from the ADC itself
  - Noise of unknown origin in current consumption
  - Timing is very tight and amplitude difference between noise and signal is very small
- Oversample
  - Sample each time delay for each digit multiple times
  - 10x oversampling seems fine
  - Adds significantly to recovery time
  - Will work with lower oversampling multiplier but less reliable
- Detect errors
  - If average times aren't the expected amount longer (28 μs) during testing for the next digit, the previous digit's value is probably wrong, so go back
  - If the time for a digit is way too early or too late, retry it

Titan – Timing attack

- Entire six-digit keypad sequence is captured before starting comparison
- Entered code is compared one digit at a time to the keycode stored in EEPROM
- If digit in entered keycode sequence doesn't match, exit loop immediately

Titan – Lockout defeat

- Goal is to get V_{dd} below STM8 brownout voltage (2.7v) before the EEPROM write has completed
- If STM8 is running (not halted), and the battery voltage (V_{batt}) is 9.0v, roughly 2.7 ms elapse between floating V_{batt} and V_{dd} going below the STM8 brownout voltage
- Can be reduced to 1.0 ms if V_{batt} starts at 4.3v and a key on keypad is held down (to increase current drain)
- To defeat the FW battery check, voltage must be reduced only *after* the STM8 has been woken up

Titan – Lockout defeat

- Failure count stored in EEPROM
- EEPROM writes on STM8 are asynchronous
  - 500 μs to complete if EEPROM block already blank
  - 3 ms to complete if block has existing data
  - EEPROM writes become blocking if second write attempted before first finishes
- If we can cut power to the STM8 after it has revealed if a digit in the keycode is valid but before the failure has been recorded…
  - …we get as many attempts as we want!

Titan – Lockout defeat

- Either:
  - Kill power before the erase-write cycle starts, or
  - Kill power after the erase part of the cycle starts but before the new value is written
- Usually, erased values in EEPROM are 0xFF
  - Not in the STM8
  - In the STM8, EEPROM erased value is 0x00
  - Thus, erased value is a valid count: “zero failures”

Titan – Automated code recovery

- First five digits via timing attack
- Sixth digit through brute force (10 attempts)
  - Try keycode ending with each possible value
  - Check if buzzer line indicates error beep sequence
    - Two long beeps = Wrong code
    - One short beep = Correct code
  - Every fourth attempt, try a known-wrong keycode and kill the power during the invalid-attempt count EEPROM update to reset the count to 0x00
  - Go through all ten possibilities this way

Titan – Lessons

- Use constant-time comparisons
  - Would defend against timing attack

Titan – Lessons

- Assume failure first
  - Increment “failed attempt” counter before key comparison begins, not after
  - Then, clear “failed attempt” count only if the correct code was actually entered
- However…
  - Don’t make the erased value of the EEPROM/flash a valid value for the counter

Titan – Lessons

- Run MCU clock faster
  - Less margin for timing attacks
  - Not a total solution, but could increase the difficulty of the attack
  - Be careful that a faster MCU doesn't lead to other stronger signals

Are there better locks? Yup!

- FF-L-2740B federal specification
  - GSA-approved locks
  - For securing material classified up to Top Secret
- Mandates significantly better design
  - Power source internal (no power analysis)
  - Resistance to various attacks for at least 20 man-hours
  - Approval revoked if design found vulnerable

References

[1] Gun safe analysis http://gunsafereviewsguy.com/

[2] “Safes and Containers: Insecurity Design Excellence” Tobias, Fiddler, and Bluzmanis. DEF CON 20

[3] “Safe to Armed in Seconds: A Study of Epic Fails of Popular Gun Safes” Deviant Ollam Cluebat Quartermaster. DEF CON 19

[4] “Hacking smart safes” Salazar and Petro. DEF CON 23

[5] DoD Lock Program http://www.navfac.navy.mil/navfac_worldwide/specialty_centers/exwc/products_and_services/capital_improvements/dod_lock.html