Poking the S in SD cards

Nicolas Oberli

Who am I ?

- Research team @KudelskiSec
  - Focusing on hardware / embedded devices security
- @BlackAlpsConf organization
  - Stickers !!
- @Hydrabus developer
  - Again, Stickers !!

How did it start ?

- SD stands for *Secure Digital*
  - What is the *Secure* for ?
- Keep the attacks as low cost as possible
  - You can replicate this at home
  - No physical attacks on the cards

Introduction to SD cards

What is an SD card ?

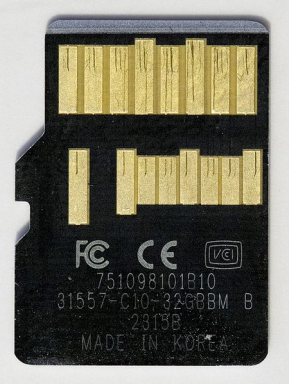
- Basically a microcontroller interfacing the SD interface with flash memory
- See bunnie and xobs talk @ 30C3 for details

Unlocked Locked

https://en.wikipedia.org/wiki/SD_card

Communication

- SD cards support 3 communication protocols
  - SPI Bus protocol
    - Classic SPI
  - SD / UHS-I Bus protocol
    - CLK, CMD, Up to 4 data lines
  - UHS-II Bus protocol
    - RCLK, 2 differential data lines

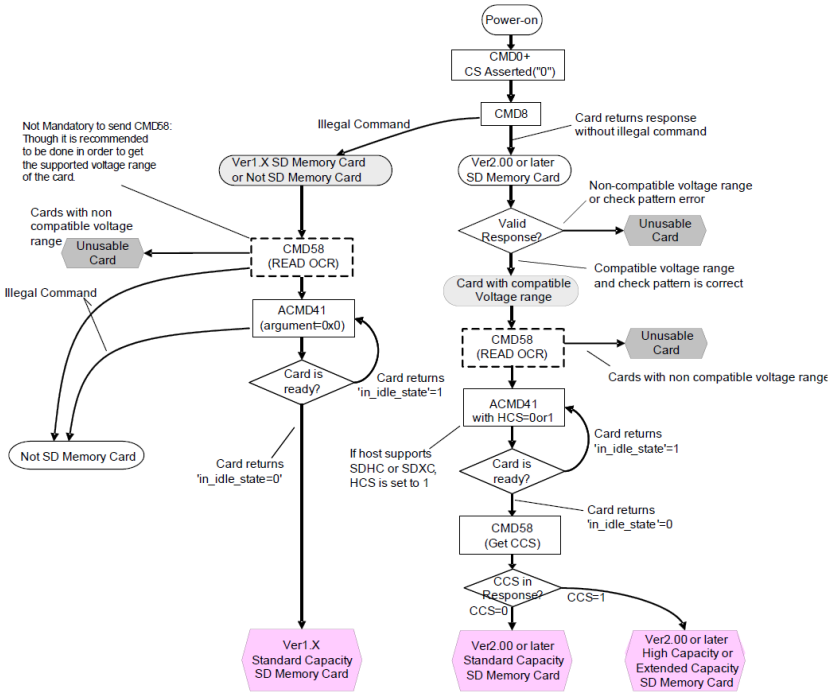


https://en.wikipedia.org/wiki/SD_card

Time to dig into the specs

- Specs are freely available in a simplified format on the SD association website
  - 262-pages document (general specs – part 1)
  - Presents the general description of the SD System

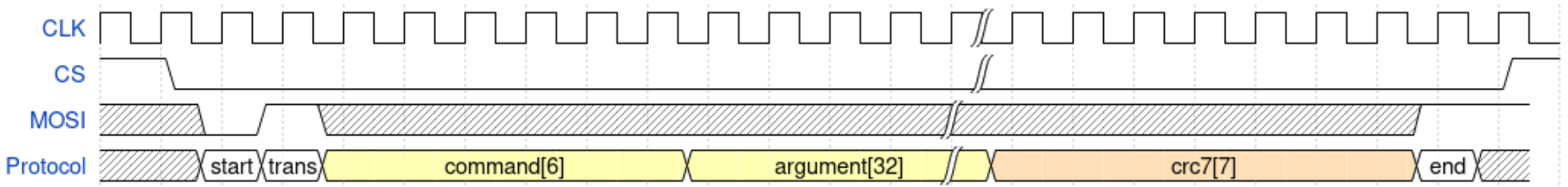
Initialization sequence



SD specs part 1, figure 7-2

Protocol

- Query/reply-based
- Each command has a number and is usually referenced with it
  - eg. CMD0 - GO_IDLE_STATE



Protocol – cont.

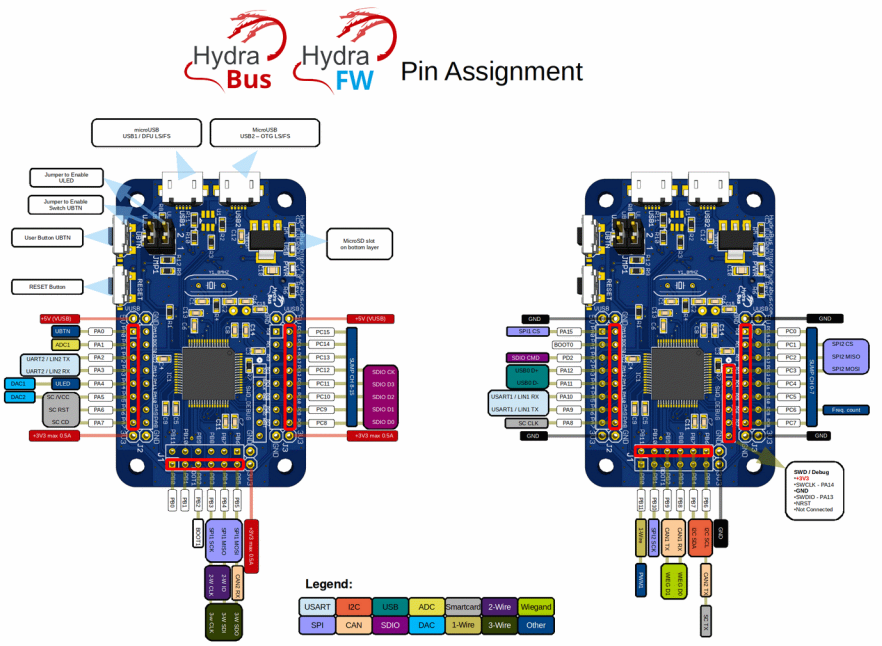
- 7 different response formats
  - Depends on the sent command
- Protocol implements a block transfer feature
  - Used to transfer more than 4 bytes
  - Block starts with *0xFE*
  - Length is defined by CMD16 (512 bytes by default)

Interfacing with SD card

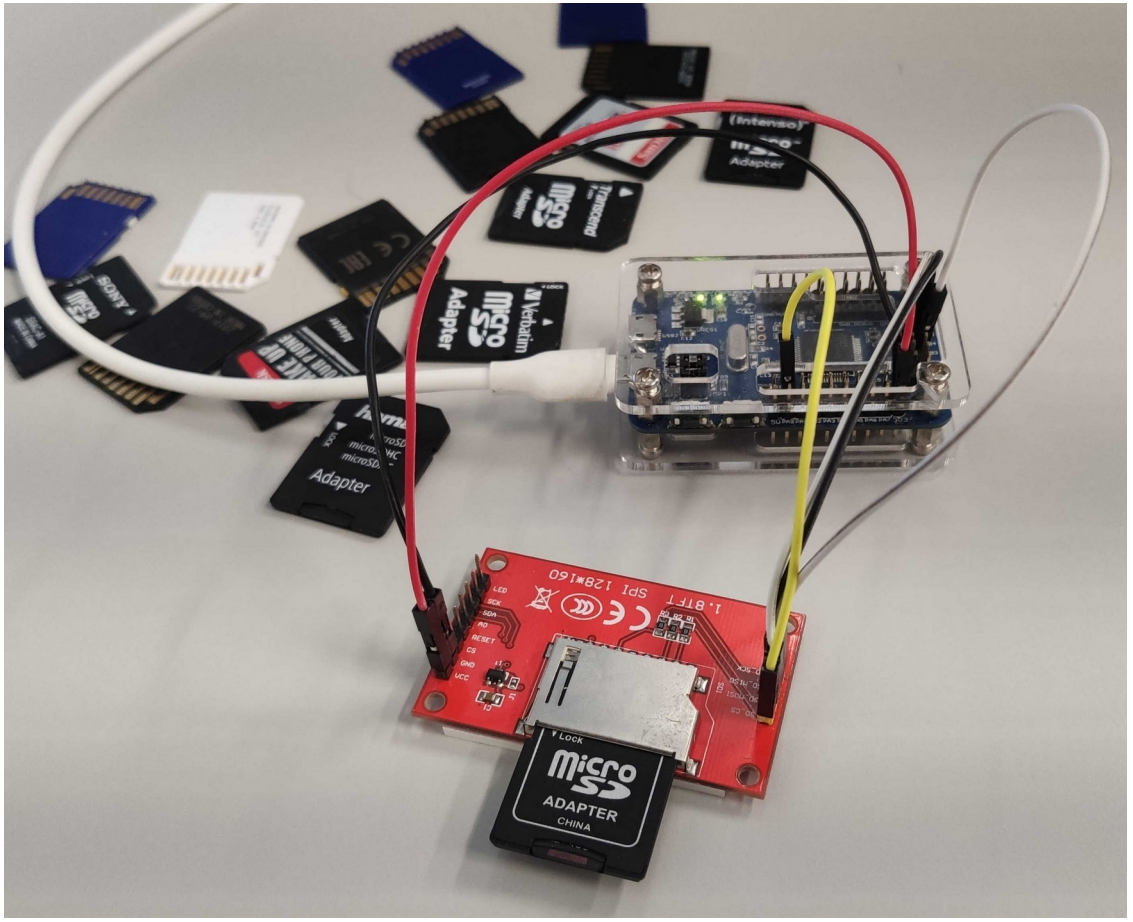
- First need to communicate correctly with the card
- SPI is used here
  - Lots of existing tools available to use SPI
  - Already supported by Hydrabus

Hydrabus

- *Bus Pirate on steroids*
  - More modern alternative
  - Many supported protocols
  - Open source



Setup



Tool

- Python CLI interface using pyHydrabus
- Drives SD card in SPI mode
  - Can send raw commands
  - Helper functions for specific commands
- CRC is optional in SPI mode, easier to play with
  - Except when some cards require a valid CRC no matter what

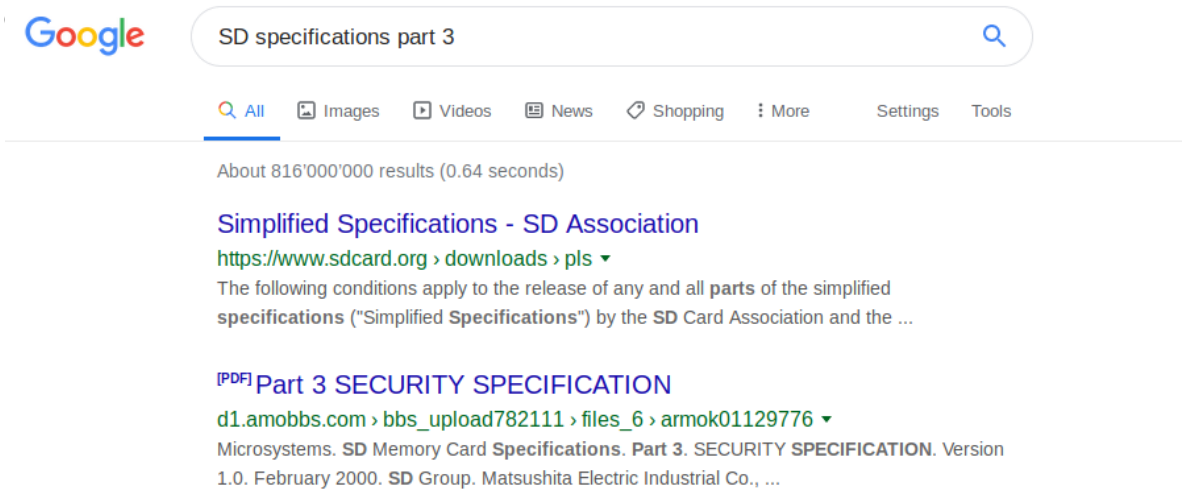
DEMO

```
sd > init
sd > cid

## CID ##
744a6055534420201042be9df6012a53
 Manufacturer ID : 0x74
 OEM ID :          b'J`'
 Product name :    b'USD  '
 Product revision :0x10
 Serial number :   42be9df6
 Manuf. date :     012a
sd > debug 20
sd > init
Response from CMD0 : 01
Response from CMD8 : 01000001aa
Response from CMD55 : 01
Response from CMD41 : 01
Response from ACMD41 : 01
Response from CMD55 : 01
Response from CMD41 : 00
Response from ACMD41 : 00
sd > csd

## CSD ##
Response from CMD9 : 00
Start of block
Block received from CMD9: 400e00325b59000076ed7f800a4000d5
400e00325b59000076ed7f800a4000d5
{'unused': 1, 'CRC': 106, 'Reserved': 0, 'FILE_FORMAT': 0, 'TMP_WRITE_PROTECT': 0, 'PE
RM_WRITE_PROTECT': 0, 'COPY': 0, 'FILE_FORMAT_GRP': 0, 'WRITE_BL_PARTIAL': 0, 'WRITE_B
L_LEN': 9, 'R2W_FACTOR': 2, 'WP_GRP_ENABLE': 0, 'WP_GRP_SIZE': 0, 'SECTOR_SIZE': 127,
'ERASE_BLK_EN': 1, 'C_SIZE': 30445, 'DSR_IMP': 0, 'READ_BLK_MISALIGN': 0, 'WRITE_BLK_M
ISALIGN': 0, 'READ_BL_PARTIAL': 0, 'READ_BL_LEN': 9, 'CCC': 1461, 'TRAN_SPEED': 50, 'N
SAC': 0, 'TAAC': 14, 'CSD_STRUCTURE': 1}
sd >
```

SD security features

Security features

- SDMI – Secure Digital Music Initiative
  - Detailed under specs part 3
    - Available only to SD members / NDA



Security features

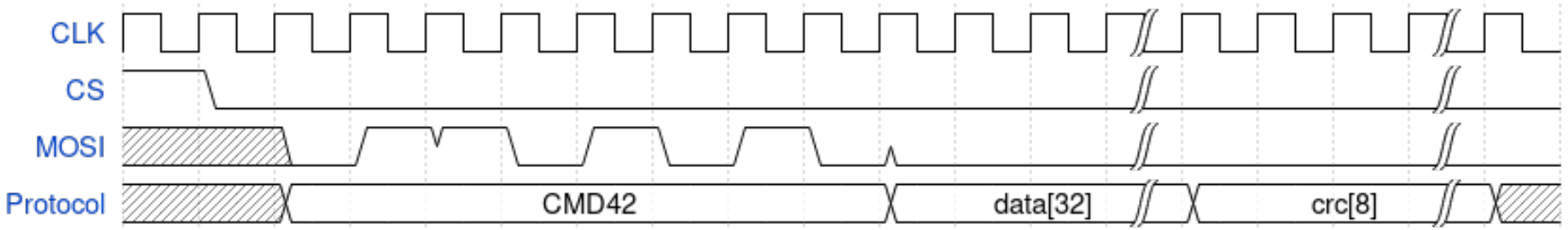
- Can be read- and/or write-protected
  - Available through several commands
    - CMD27 to set write protection bits
    - CMD42 to set read protection password
- These commands are **mandatory** to get SD label

CMD42 – LOCK_UNLOCK

- Used to control the password protection mechanism
  - Up to 16 bytes
  - Not limited to printable characters
  - Keyspace : 2^{128} – Same as an AES key
    - Bruteforce is unachievable

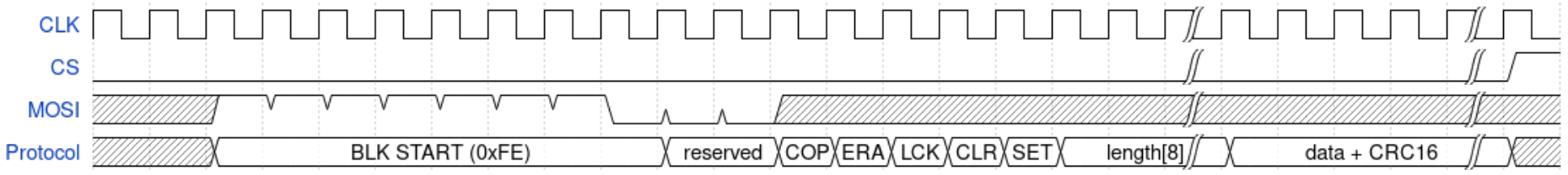
Locking the SD card

- The *CMD42* command controls the password locking functions
  - Takes no parameter, but card expects a following data block



CMD42 data block

- Contains the command options, length and the actual password



Locking SD card

- Send *CMD42*
- Send a data block, setting the *SET* bit, the password length and the password
  - Can optionally set the *LOCK* bit to lock the card in the process
- Lock status is available in the status bits (CMD13)

Unlocking SD card

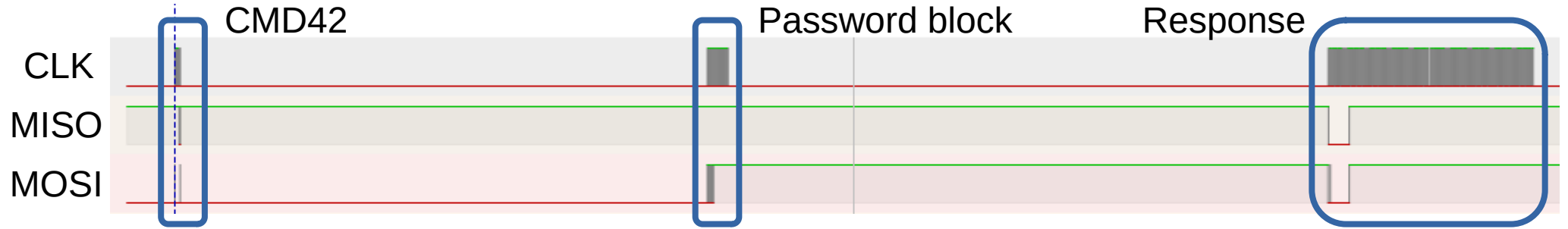
- Send *CMD42*
- Send a data block, unsetting the *LOCK* bit, setting the password length and the password
- Card will assert the MISO line, then send an ACK once the command has been processed
- Lock status is available in the status bits (CMD13)

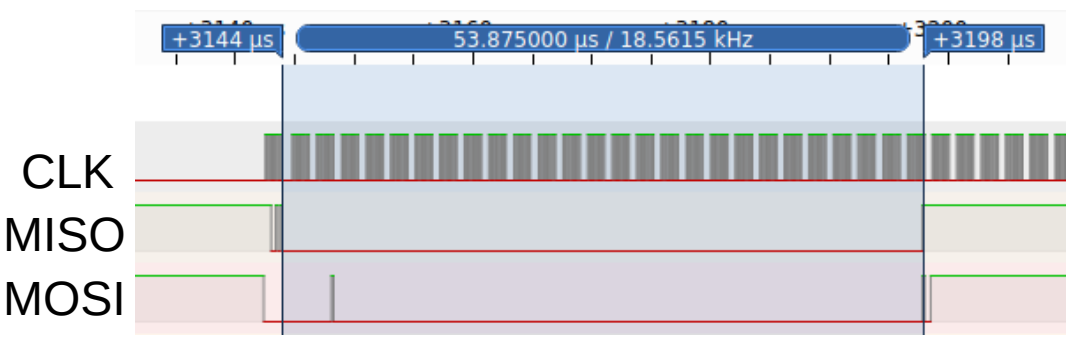
Attacking the password protection

Unlocking SD card

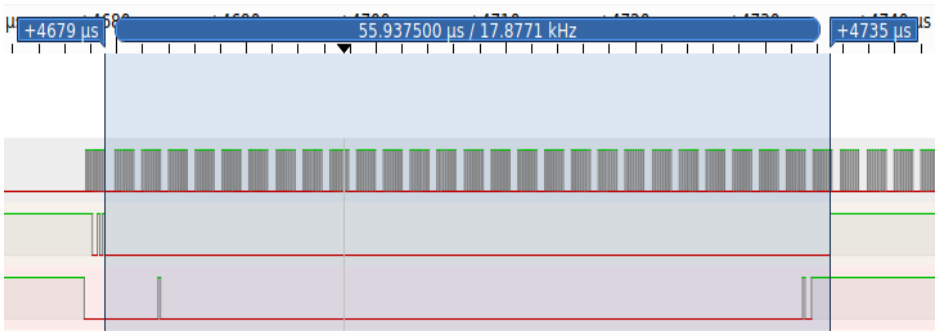
- Send *CMD42*
- Send a data block, unsetting the *LOCK* bit, setting the password length and the password
- **Card will assert the MISO line, then** send an ACK once the command has been processed
- Lock status is available in the status bits (CMD13)

Guess what happens ?





00000



000000

What’s happening ?

- SD controller checks for the length of the password, then compares each byte to the correct password
- Returns an error as soon as there is a discrepancy
- Possible to determine a correct byte by measuring processing time

Measuring time using SPI

- During processing time, read dummy bytes as fast as possible
- As long as we read zeroes, the password check is still ongoing
- Once we read a 1, count the number of zeroes

In practice :

```
sd > test_len
  00 : 122
  01 : 124
  02 : 124
  03 : 124
  04 : 124
  05 : 124
* 06 : 130
  07 : 124
  08 : 124
  09 : 124
  10 : 124
  11 : 124
  12 : 124
  13 : 124
  14 : 124
  15 : 124
  16 : 124
Length: 6
sd >
```

DEMO

So ?

- Bought a bunch of SD cards (~20)
  - Different vendors
  - Different sizes
- Also asked colleagues / friends for SD cards
  - The only card I permanently locked was not mine (‘-’*)
- Locked them with “123456” as password

Special cases – Sony SD

- Card refuses to check the password after three failed attempts
- Need to remove and insert the card again to get 3 more attempts
  - In fact, doing a reset sequence (CMD0) is enough to get 3 more tries
  - Slightly makes the bruteforce slower

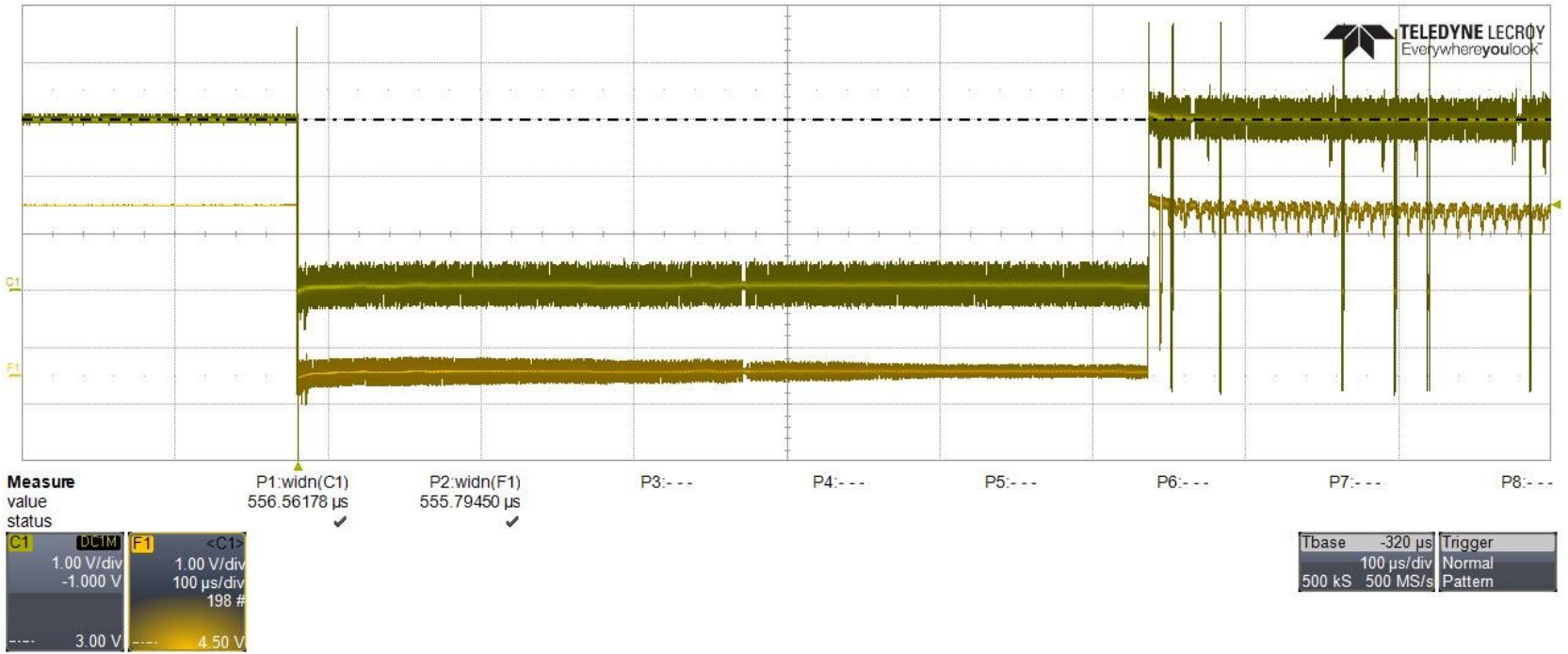
Special cases – Sony uSD

- Card seems to have a really fast checking time
  - Can get no or maybe one zero bit
- Sampling rate might be too slow
  - SPI interface is ~42MHz
  - Using logic analyzer (100MS/s) still does not show any usable results

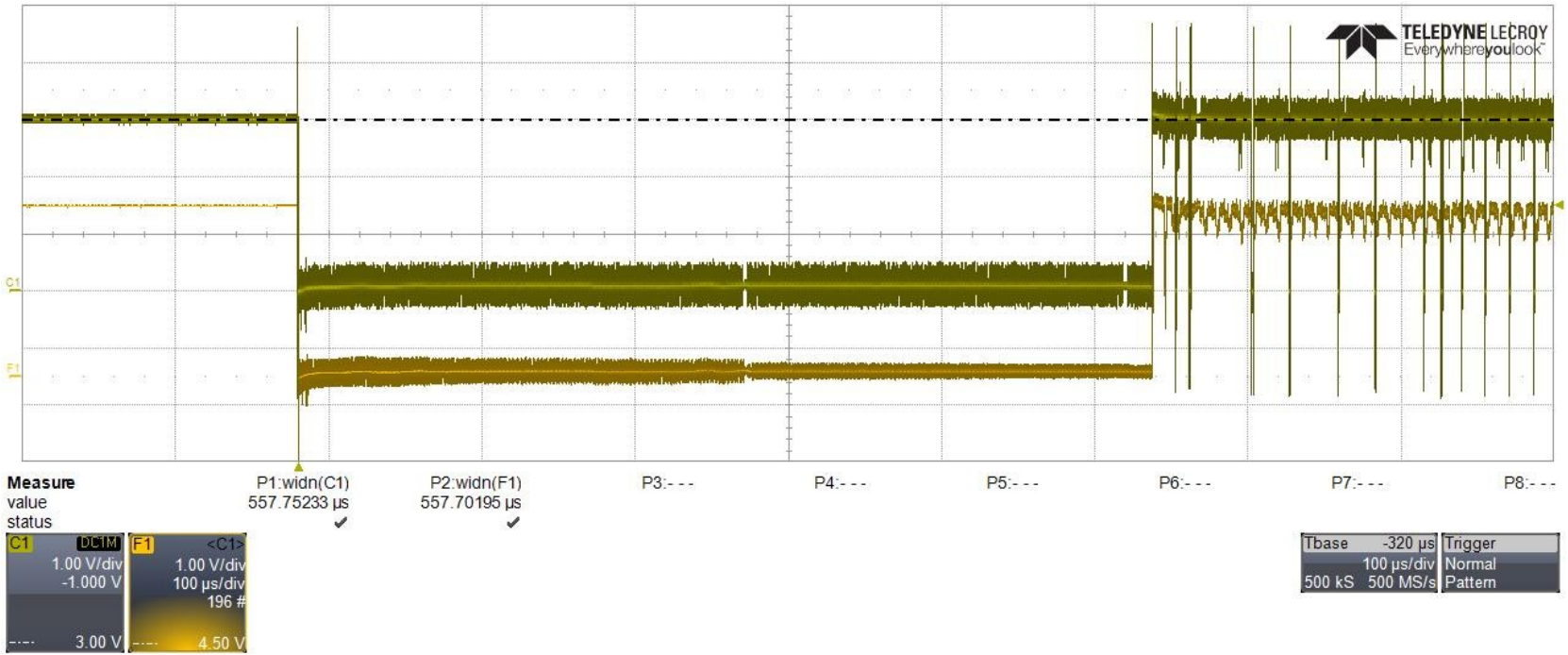
Faster !!

- Used lab oscilloscope
  - Up to 40GS/s, more than enough
- Had to setup a trigger for correct measurement

And...


TELEDYNE LECROY
Everywhereyoulook
C1
F1
Measure
value
status
P1:widn(C1)
556.56178 µs
P2:widn(F1)
555.79450 µs
P3:- - -
P4:- - -
P5:- - -
P6:- - -
P7:- - -
P8:- - -
C1 DC1M
1.00 V/div
-1.000 V
3.00 V
F1 <C1>
1.00 V/div
100 µs/div
198 #
4.50 V
Tbase -320 µs
100 µs/div
500 kS 500 MS/s
Trigger
Normal
Pattern

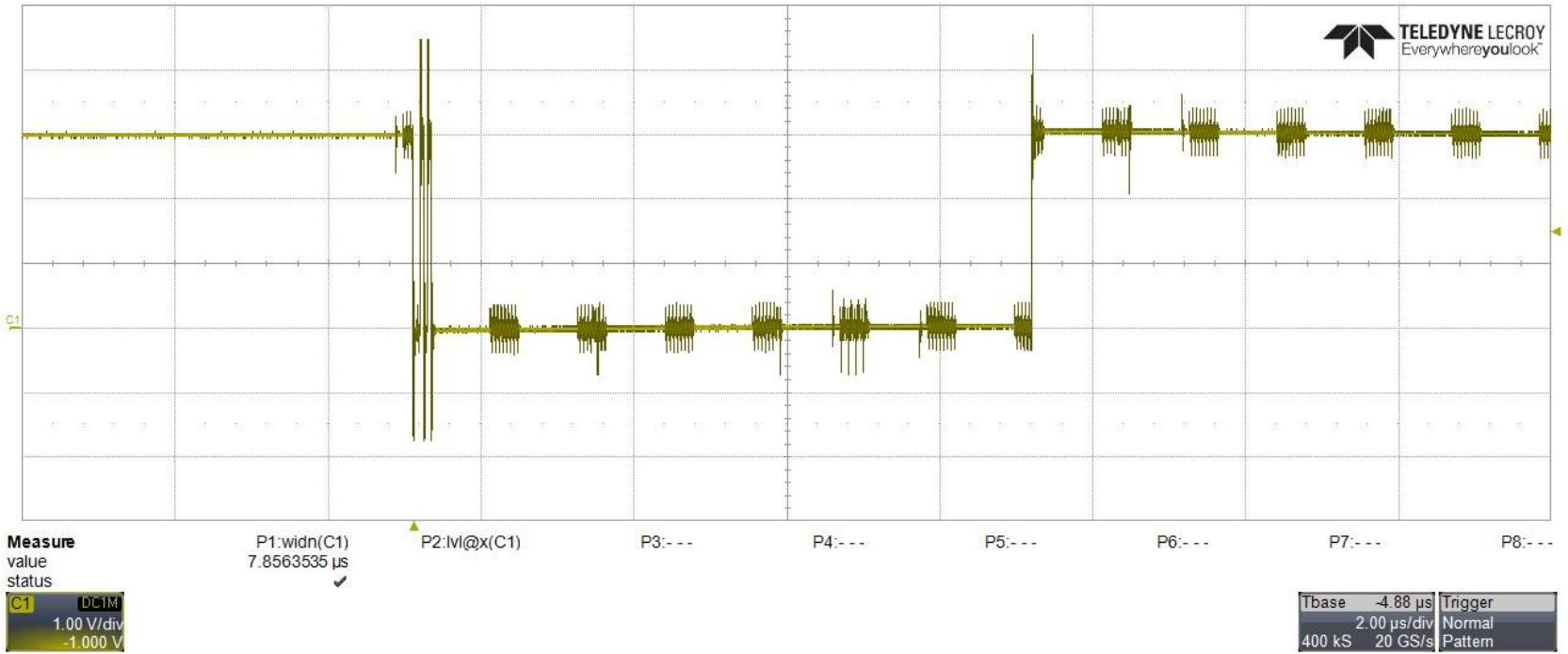
And...


TELEDYNE LECROY
Everywhereyoulook
C1
F1
Measure
value
status
P1:widn(C1)
557.75233 µs
P2:widn(F1)
557.70195 µs
P3:- - -
P4:- - -
P5:- - -
P6:- - -
P7:- - -
P8:- - -
C1 DC1M
1.00 V/div
-1.000 V
3.00 V
F1 <C1>
1.00 V/div
100 µs/div
196 #
4.50 V
Tbase -320 µs
100 µs/div
500 kS 500 MS/s
Trigger
Normal
Pattern

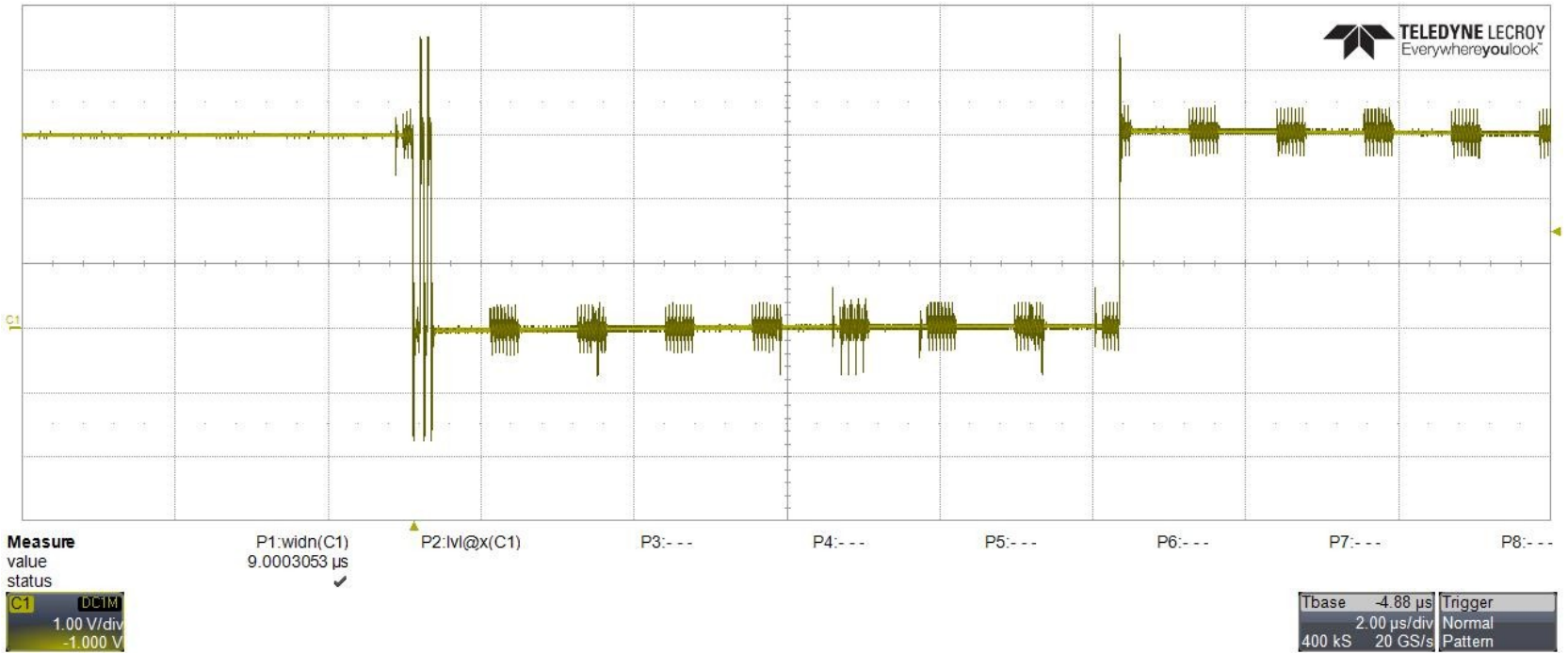
Special cases - Kingston

- It is possible to count the password length, but not the password chars
- Took a lot of measurements until I found this :

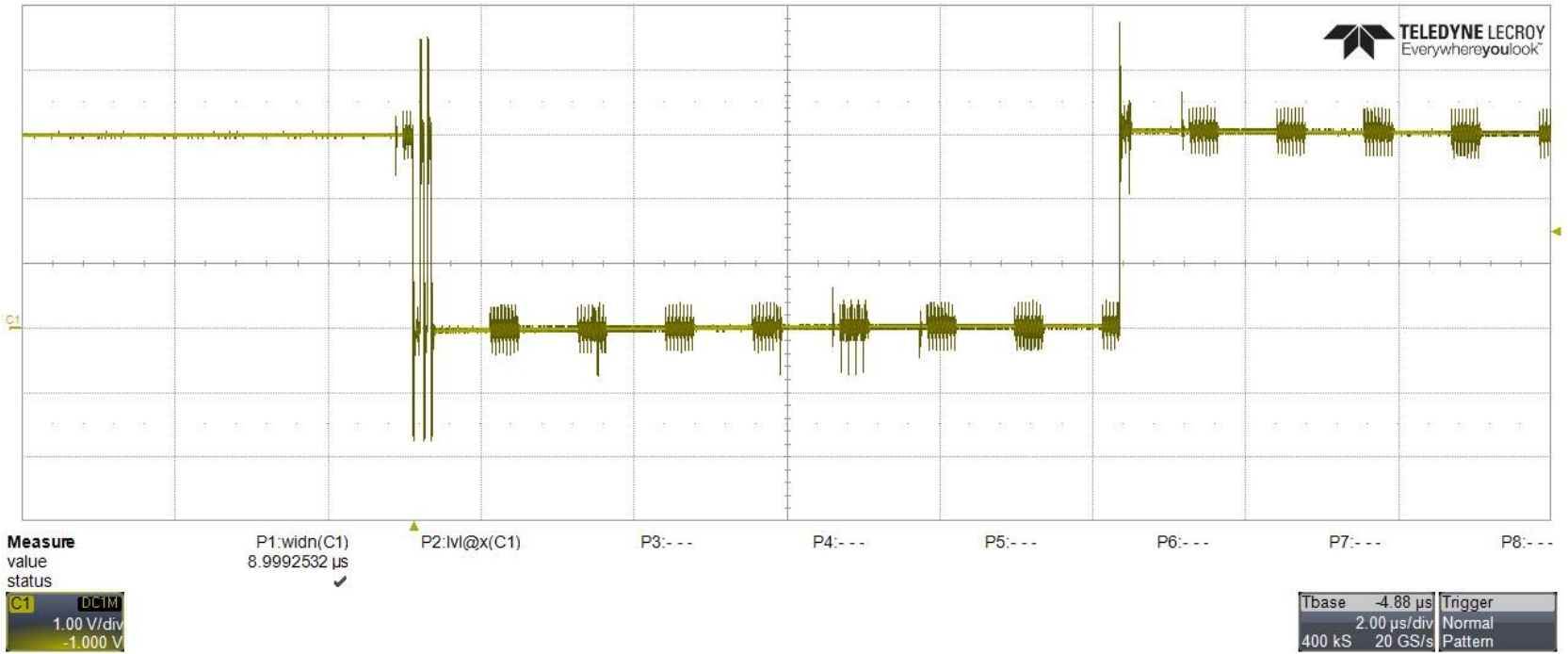
00000


TELEDYNE LECROY
Everywhereyoulook
C1
Measure
value
status
P1:widn(C1)
7.8563535 µs
P2:lvl@x(C1)
P3:- - -
P4:- - -
P5:- - -
P6:- - -
P7:- - -
P8:- - -
C1 DC1M
1.00 V/div
-1.000 V
Tbase -4.88 µs
2.00 µs/div
400 kS 20 GS/s
Trigger
Normal
Pattern

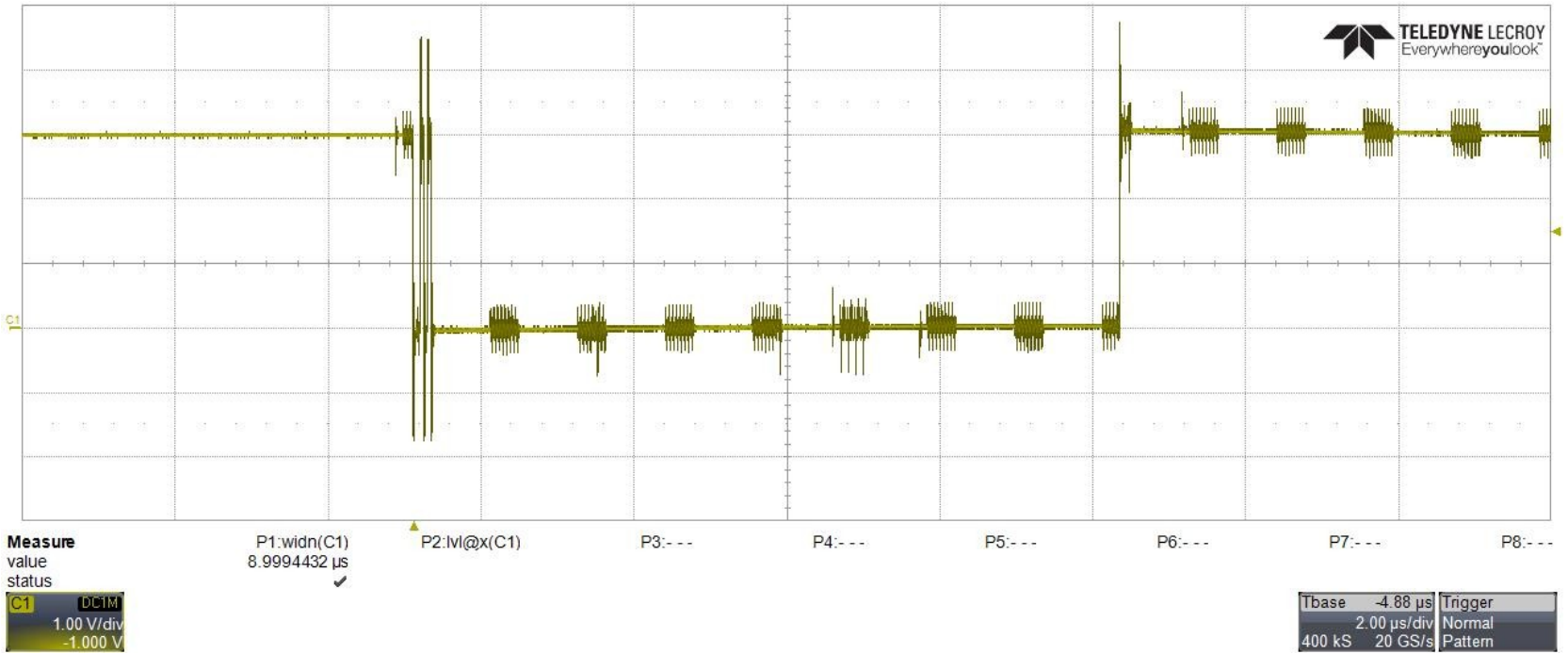
000000


TELEDYNE LECROY
Everywhereyoulook
C1
Measure
value
status
P1:widn(C1)
9.0003053 µs
P2:lvl@x(C1)
P3:- - -
P4:- - -
P5:- - -
P6:- - -
P7:- - -
P8:- - -
C1 DC1M
1.00 V/div
-1.000 V
Tbase -4.88 µs
2.00 µs/div
400 kS 20 GS/s
Trigger
Normal
Pattern

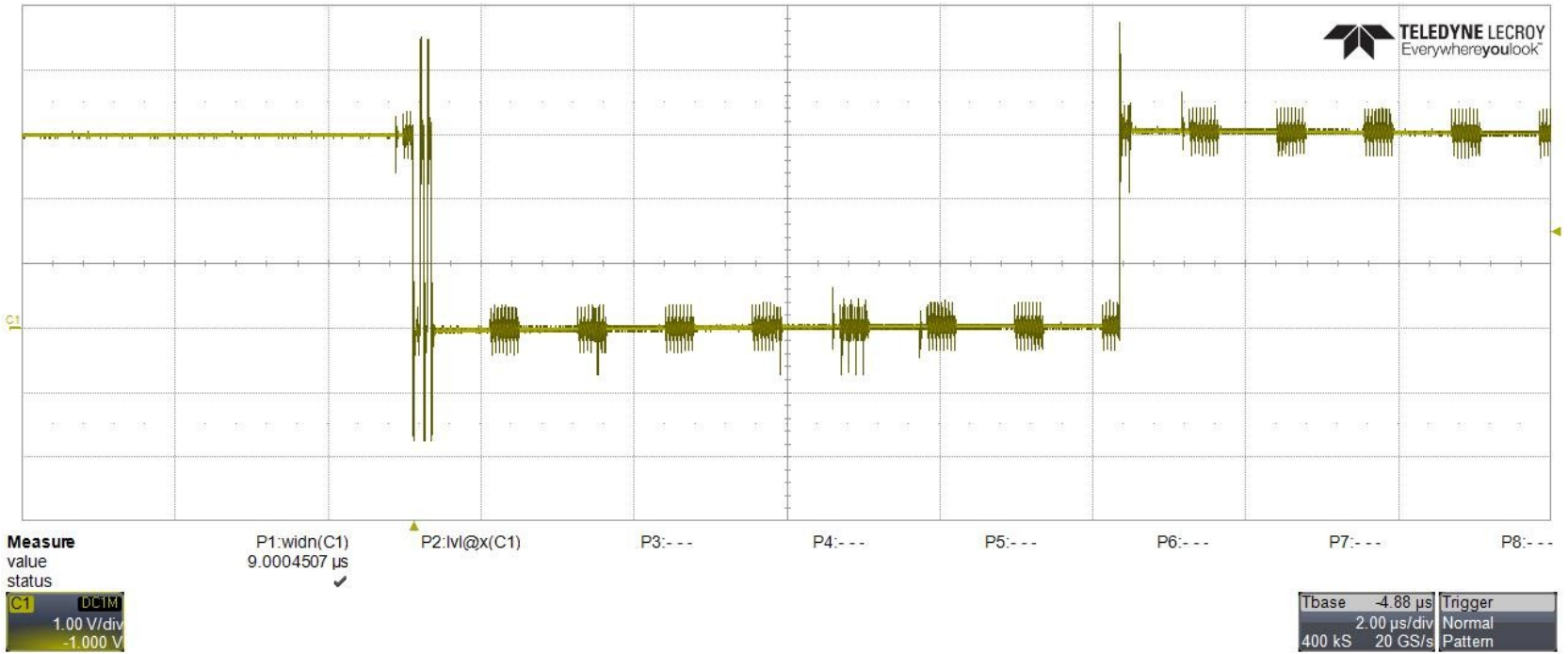
100000


TELEDYNE LECROY
Everywhereyoulook
C1
Measure
value
status
P1:widn(C1)
8.9992532 µs
P2:lvl@x(C1)
P3:- - -
P4:- - -
P5:- - -
P6:- - -
P7:- - -
P8:- - -
C1 DC1M
1.00 V/div
-1.000 V
Tbase -4.88 µs
2.00 µs/div
400 kS 20 GS/s
Trigger
Normal
Pattern

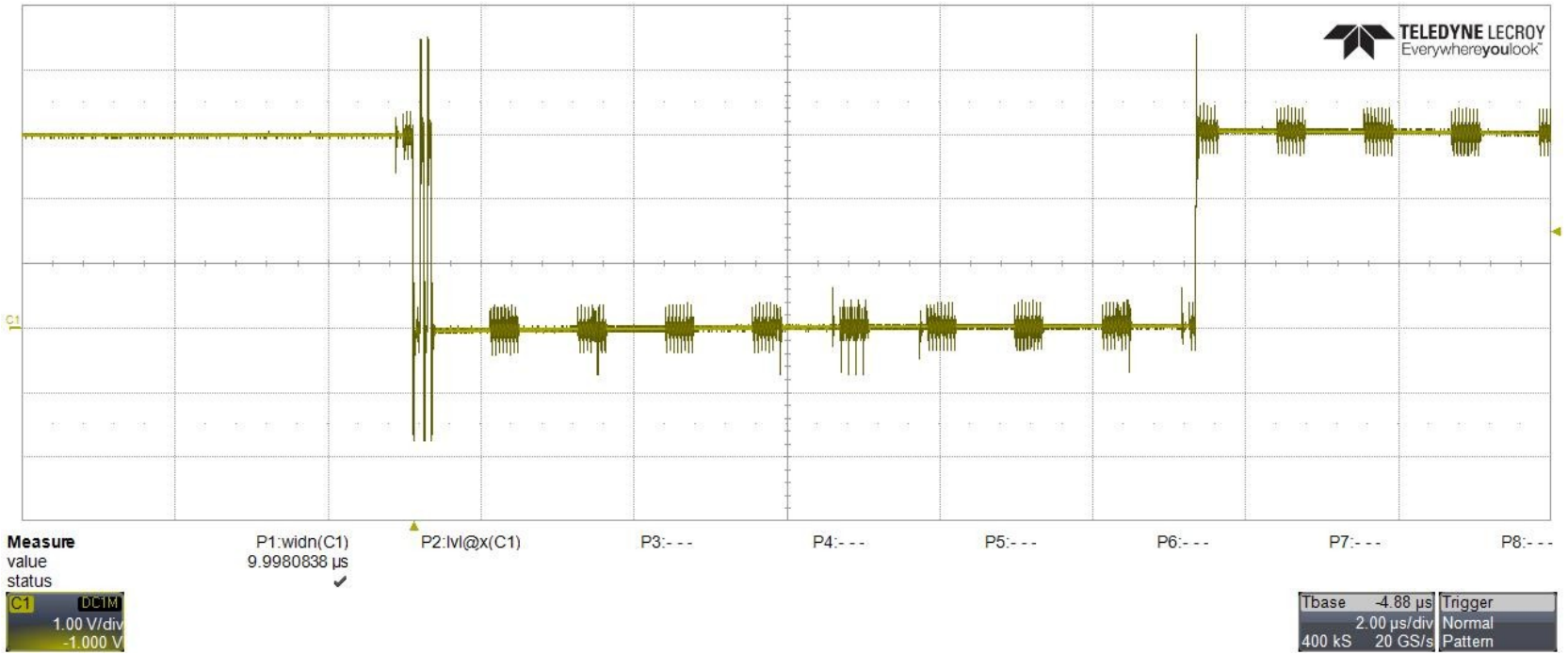
120000


TELEDYNE LECROY
Everywhereyoulook
C1
Measure
value
status
P1:widn(C1)
8.9994432 µs
P2:lvl@x(C1)
P3:- - -
P4:- - -
P5:- - -
P6:- - -
P7:- - -
P8:- - -
C1 DC1M
1.00 V/div
-1.000 V
Tbase -4.88 µs
2.00 µs/div
400 kS 20 GS/s
Trigger
Normal
Pattern

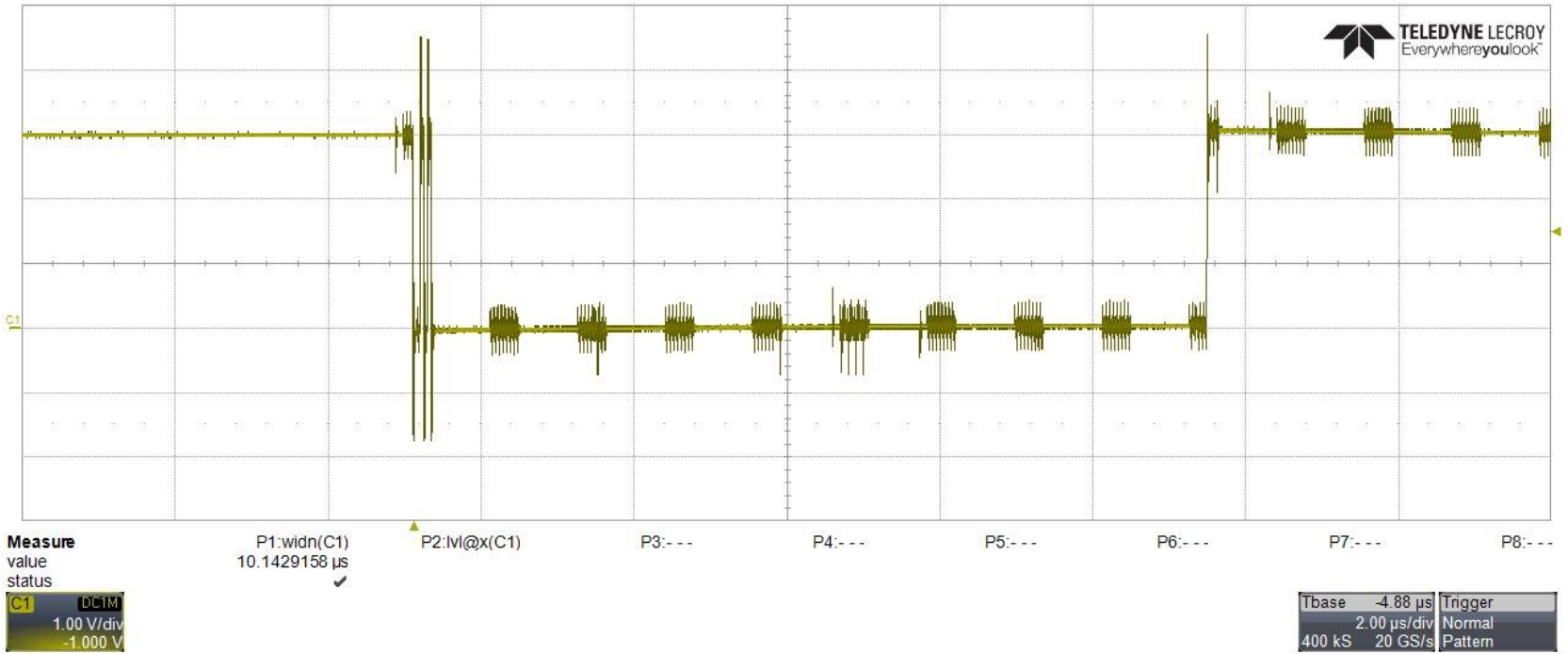
123000


TELEDYNE LECROY
Everywhereyoulook
C1
Measure
value
status
P1:widn(C1)
9.0004507 µs
P2:lvl@x(C1)
P3:- - -
P4:- - -
P5:- - -
P6:- - -
P7:- - -
P8:- - -
C1 DC1M
1.00 V/div
-1.000 V
Tbase -4.88 µs
2.00 µs/div
400 kS 20 GS/s
Trigger
Normal
Pattern

123400


TELEDYNE LECROY
Everywhereyoulook
C1
Measure
value
status
P1:widn(C1)
9.9980838 µs
P2:lvl@x(C1)
P3:- - -
P4:- - -
P5:- - -
P6:- - -
P7:- - -
P8:- - -
C1 DC1M
1.00 V/div
-1.000 V
Tbase -4.88 µs
2.00 µs/div
400 kS 20 GS/s
Trigger
Normal
Pattern

123450


TELEDYNE LECROY
Everywhereyoulook
C1
Measure
value
status
P1:widn(C1)
10.1429158 µs
P2:lvl@x(C1)
P3:- - -
P4:- - -
P5:- - -
P6:- - -
P7:- - -
P8:- - -
C1 DC1M
1.00 V/div
-1.000 V
Tbase -4.88 µs
2.00 µs/div
400 kS 20 GS/s
Trigger
Normal
Pattern

Still vulnerable

- Password checking works on groups of 4 bytes
- If remaining bytes to check is >= 4, test each byte individually
- Attack takes more time, but works anyways

Results

Card	Manufacturer	Prod. date*	Vulnerable ?
Transcend uSD 4GB	Transcend (0x74)	09/2011	Yes
Transcend uSD 16GB	Transcend (0x74)	10/2012	Yes
Hama 8GB	Phison (0x27)	06/2010	Yes
Maxell 32GB	Phison (0x27)	10/2011	Yes
Sony uSD 32GB	Sony (0x9c)	07/2012	Yes
Sony 32GB	Sony (0x9c)	12/2011	Yes
Kingston uSD 32GB	Unknown (0x9f)	10/2012	Yes
Sandisk Extreme 128GB	Sandisk (0x03)	03/2012	No
Sandisk mobile ultra 16GB	Sandisk (0x03)	12/2009	No
Samsung Evo+ uSD 32GB	Samsung (0x1b)	10/2012	Unsupported

* Production date format is not consistent

Ouch

- Sandisk only controller I tested not vulnerable to this attack
- Remember : SD vendor != Controller manufacturer
- Samsung cards respond with *invalid command* when sending CMD42

Write lock mechanism

Abusing Write lock mechanism

- Setting the *TMP_WRITE_PROTECT* bit in CSD register puts the card in read-only mode
- Hypothesis: This will prevent the flash memory content to be erased when a clear password is sent

Testing for vulnerability

- Write data on some pages
- Set write protection
- Set password
- Power cycle card
- Clear password
- Test for password presence, and if data is still present

Results

- All tested cards do correctly erase the *TMP_WRITE_PROTECT* flag and erase the data
- Did not test the permanent write protect yet

4.3.7.3.3 Force Erase Function to the Locked Card

Table 4-8 clarifies the relation between force erase and Write Protection. The force erase does not erase the secure area. The card shall keep its locked state during the erase execution and change to the unlocked state after the erase of all user area is completed. Similarly, the card shall keep Temporary and Group Write Protection during the erase execution and clear Write Protection after the erase of all user area is completed. In the case of an erase error occurs, the card can continue force erase if the data of error sectors are destroyed.

Abusing password clear feature

Password clearing

- By setting the CLR bit in CMD42, it is possible to remove the password protection on a card
- Card content is erased in the process
- Hypothesis : Does the card clear its password BEFORE erasing the flash memory ?

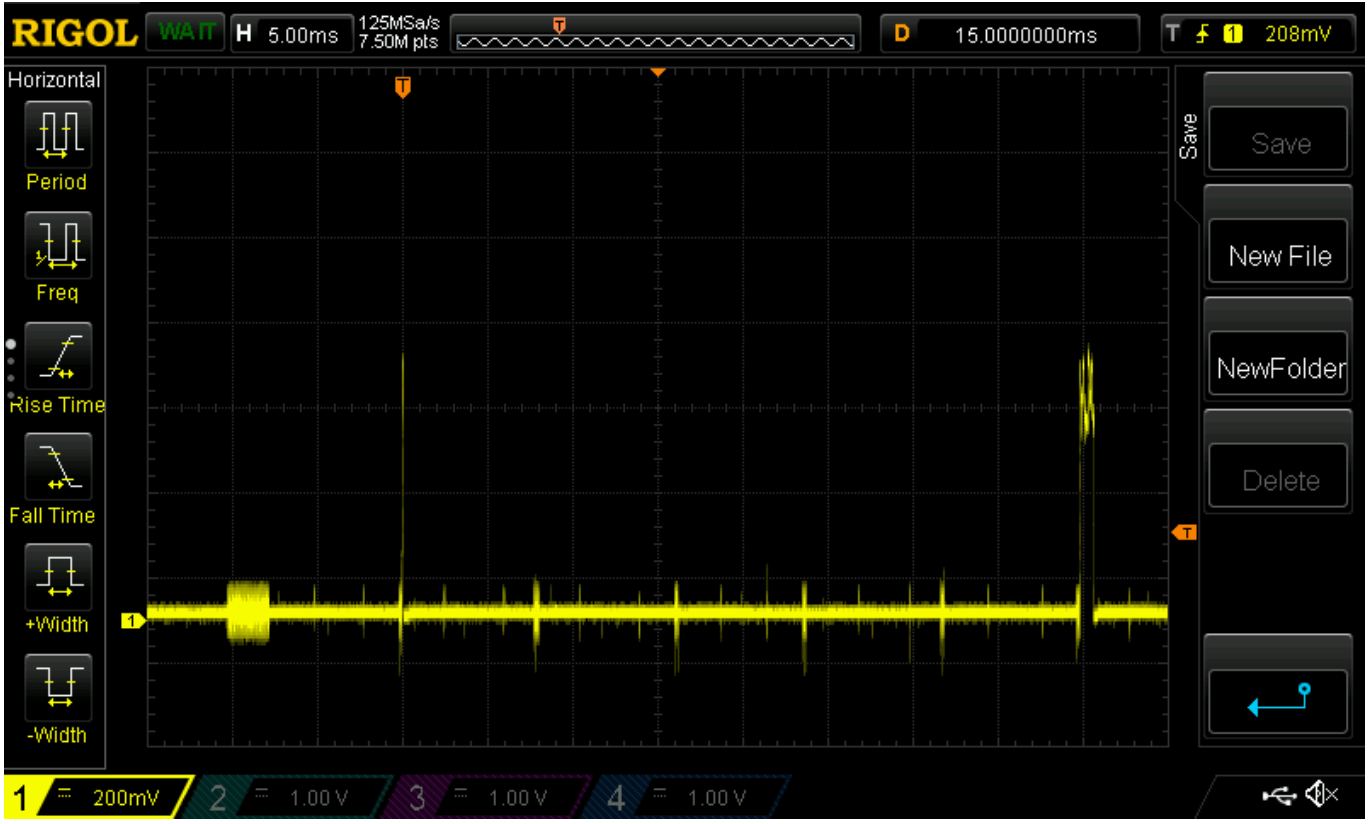
Detecting flash writes

- Flash memory uses charge-pump mechanism to provide enough current to change memory value
- Detecting an increase in power consumption would mean the flash will be written

Measuring current consumption

- Ohm’s law : increasing the current through a fixed resistance will increase voltage drop
- Add a small (<10Ω) resistance after the SD card and measure voltage using oscilloscope
  - Might need to slightly increase source voltage
- Budget-tip: If you don’t have small resistances, vape coils do work

Power consumption



By the way...

- Checking a password consumes power
- So the timing attack is also visible by looking at the card power consumption

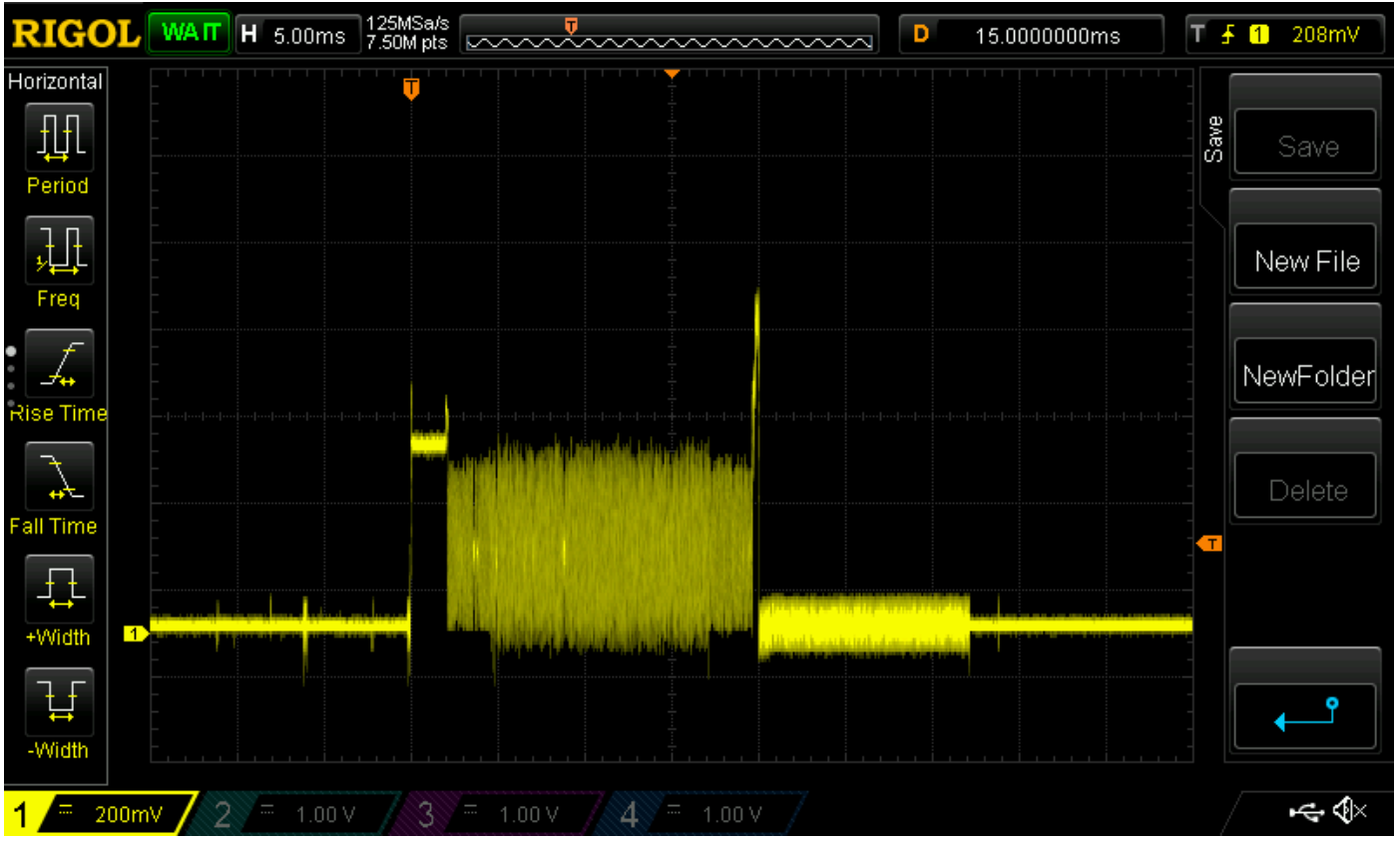
00000


RIGOL
WAIT
H 5.00ms
125MSa/s
7.50M pts
D 15.0000000ms
T 208mV
Horizontal
Period
Freq
Rise Time
Fall Time
+Width
-Width
Save
Save
New File
NewFolder
Delete
1 200mV
2 1.00 V
3 1.00 V
4 1.00 V

000000


RIGOL
WAIT
H 5.00ms
125MSa/s
7.50M pts
D 15.0000000ms
T 208mV
Horizontal
Period
Freq
Rise Time
Fall Time
+Width
-Width
Save
Save
New File
NewFolder
Delete
1 200mV
2 1.00 V
3 1.00 V
4 1.00 V

123450


RIGOL
WAIT
H 5.00ms
125MSa/s
7.50M pts
D 15.0000000ms
T 208mV
Horizontal
Period
Freq
Rise Time
Fall Time
+Width
-Width
Save
Save
New File
NewFolder
Delete
1 200mV
2 1.00 V
3 1.00 V
4 1.00 V

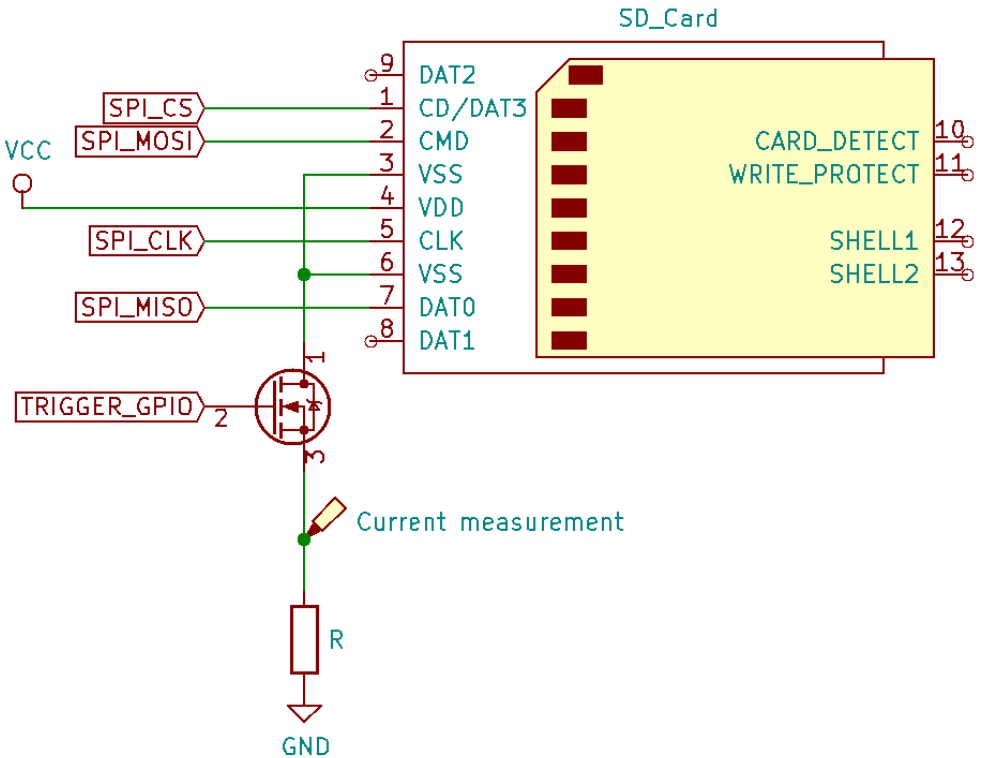
Triggering on consumption

- STM32 ADC offers a watchdog feature
  - Watchdog triggered when voltage goes above or below thresholds
- Added feature to Hydrabus
  - Programmable thresholds
  - Programmable delay (1µs minimum delay)

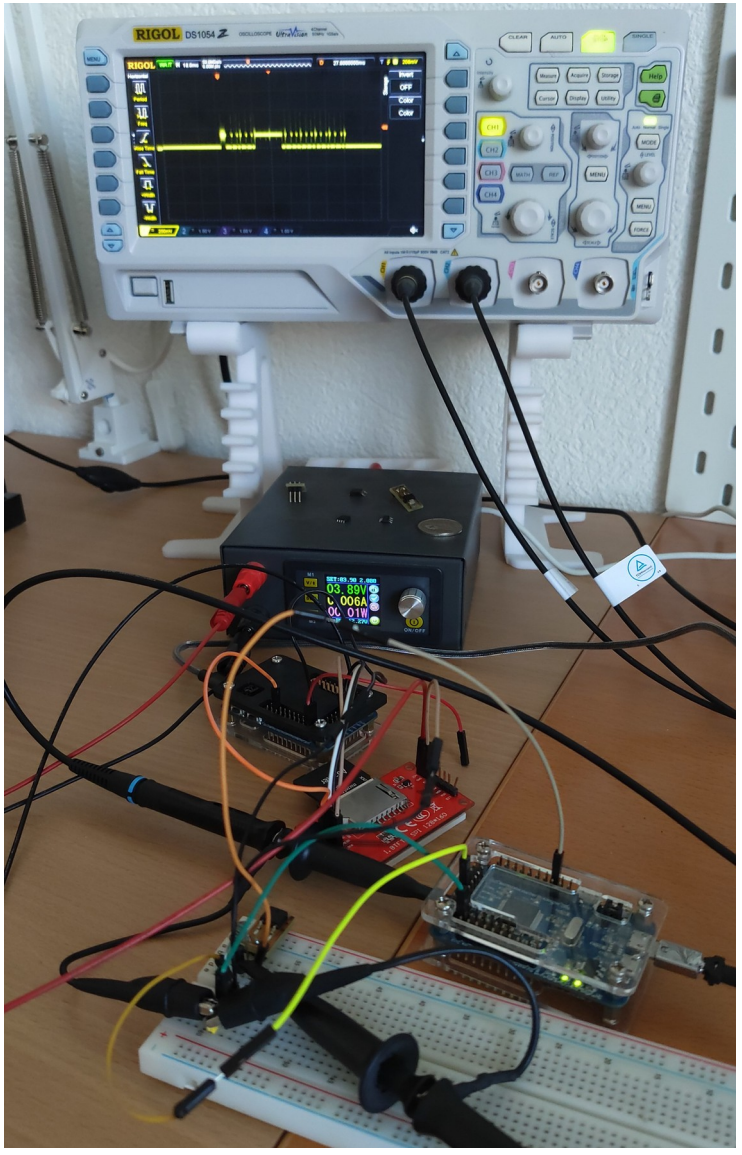
Cutting power

- Used a MOSFET to drive the SD card current
  - Easy to use as a digital switch
  - Can be operated by a GPIO
- Budget-tip: Motherboards have a lot of MOSFETs that can be used.
  - Recycle your old stuff !

Final schema



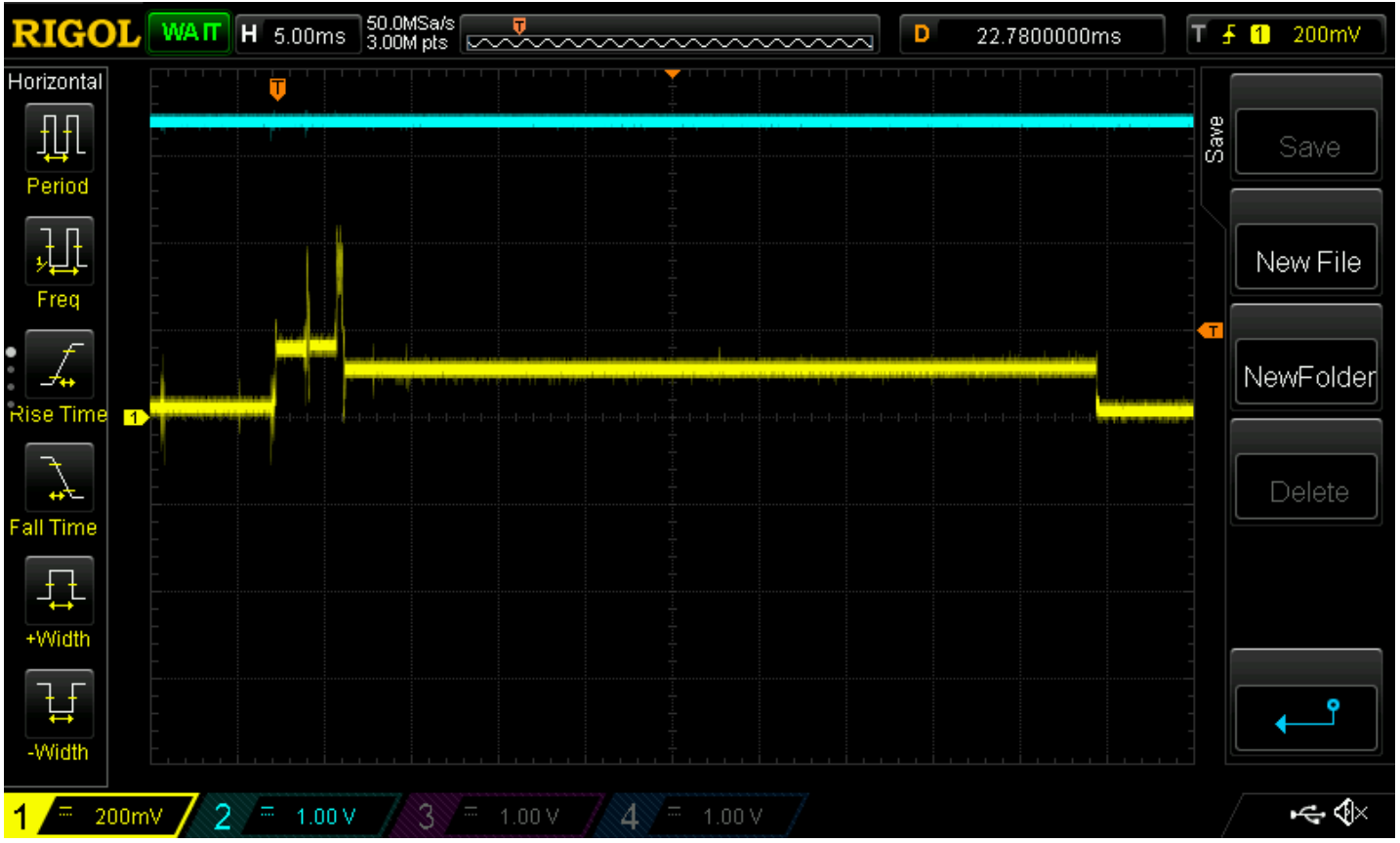
In practice



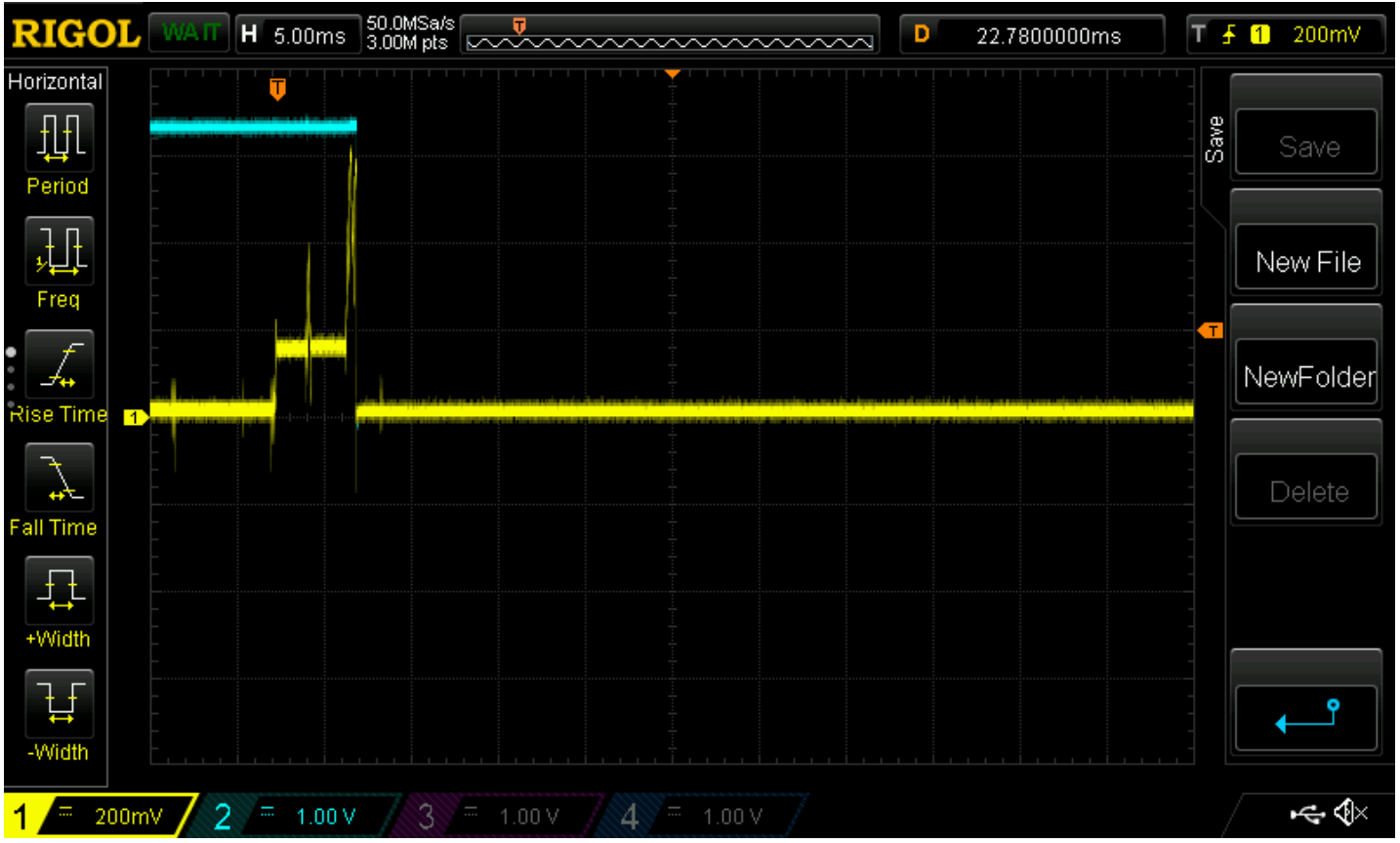
Testing for vulnerability

- Write data on some pages
- Set password
- Power cycle card
- Clear password with trigger
- Test for password presence, and if data is still present

Example – Flash erase



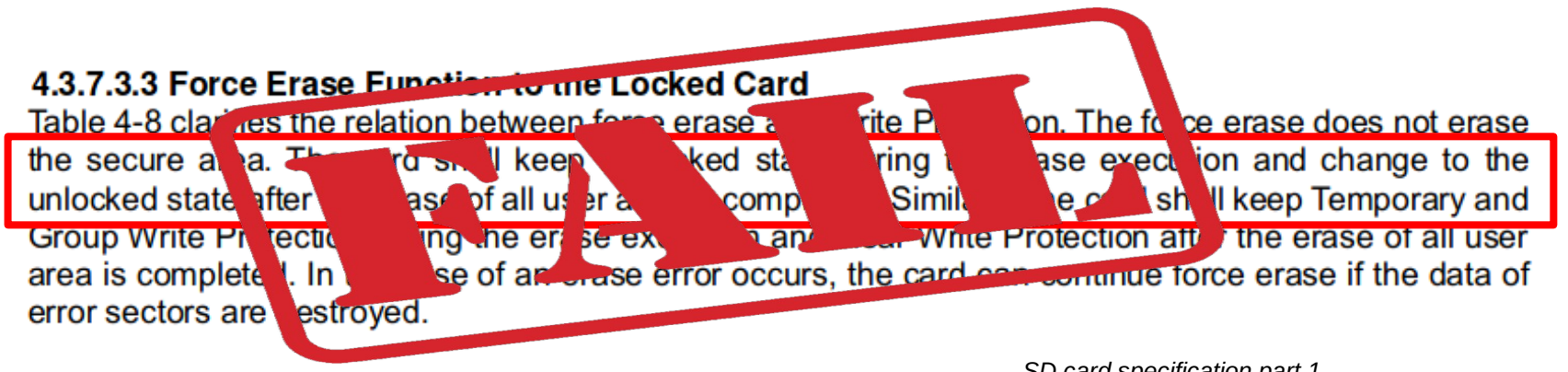
Example – Flash erase glitched



Example – Different card

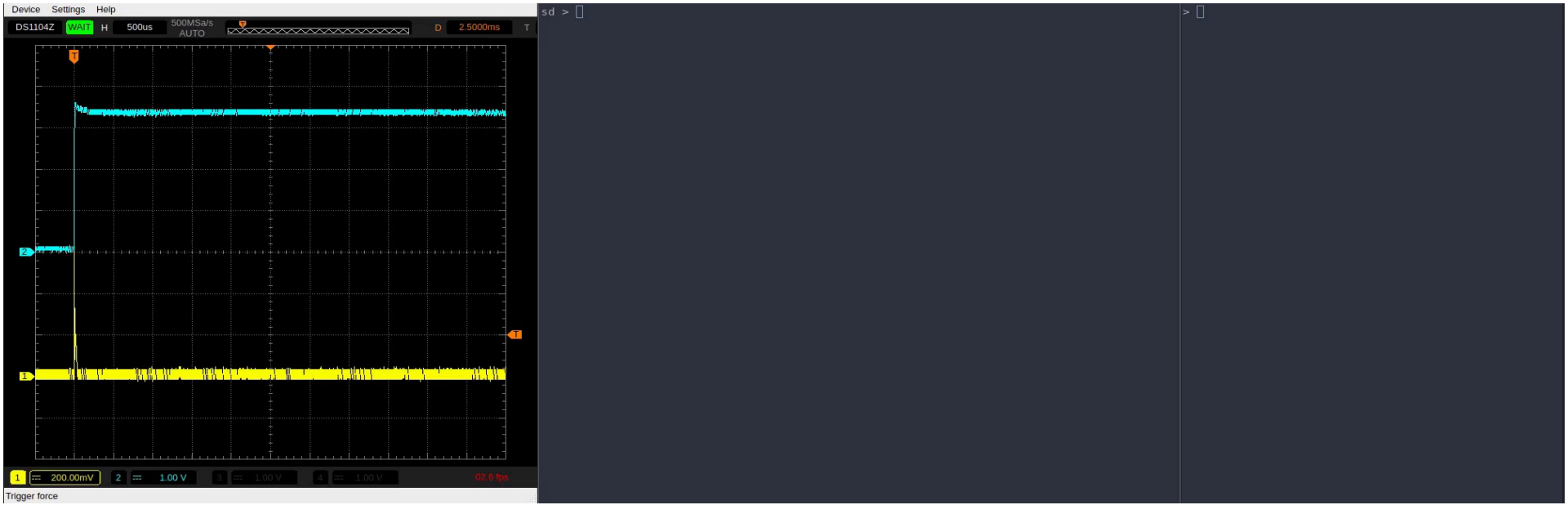
Vulnerable ?

- Some cards were successfully unlocked using this technique
- No data page lost in the process \o/



SD card specification part 1

Demo


Device Settings Help
DS1104Z WAIT H 500us 500MSa/s AUTO D 2.5000ms T
1 200.00mV 2 1.00 V 3 1.00 V 4 1.00 V 02.6 fps
Trigger force
sd >
>

Results

Card	Manufacturer	Prod. date*	Vulnerable ?
Transcend uSD 4GB	Transcend (0x74)	09/2011	Died :(
Transcend uSD 16GB	Transcend (0x74)	10/2012	Yes
Hama 8GB	Phison (0x27)	06/2010	Yes
Maxell 32GB	Phison (0x27)	10/2011	Yes
Sony uSD 32GB	Sony (0x9c)	07/2012	Yes
Sony 32GB	Sony (0x9c)	12/2011	No
Kingston uSD 32GB	Unknown (0x9f)	10/2012	Yes
Sandisk Extreme 128GB	Sandisk (0x03)	03/2012	No
Sandisk mobile ultra 16GB	Sandisk (0x03)	12/2009	No
Samsung Evo+ uSD 32GB	Samsung (0x1b)	10/2012	Unsupported

* Production date format is not consistent

Conclusions

Conclusions

- Useless vulnerabilities ?
  - Feature not supported by any OS
- Affects a lot of manufacturers
- Reading specs is fun
  - Don’t take them as granted though

Conclusions – Cont

- Simple side-channel analysis is not that hard
  - Does not require a lot of expensive tools to get things done if you are creative
- Tools are evolving, their price get lower, getting more accessible

Conclusions – Cont

- Automation is key
  - When you need hundreds of samples, better not have to stay around while it’s working

Based on https://xkcd.com/303/

Future work

- COP protection
  - Added in specs v5.00 (2016)
  - Adds a password to protect the clear password feature
  - Couldn’t find a card that supports it

Thank you !

Nicolas Oberli
@baldanos