black hat USA 2012

OPALE security

How the analysis of electrical current consumption of embedded systems could lead to code reversing ?

“Code extraction via Power analysis”

Focus on “Embedded systems”

Yann ALLAIN / Julien MOINARD

AGENDA

- Who we are
- Research context & goals
- Electronic 101 for Security Guys
- Proof of concept (soft, hard, …)
- Our experiments
- Results & Limits
- Further researches (Prospective)
- How to limit the risk
- Conclusion

AGENDA

- Who we are
- Research context & goals
- Electronic 101 for Security Guys
- Proof of concept (soft, hard, …)
- Our experiments
- Results & Limits
- Further researches (Prospective)
- How to limit the risk
- Conclusion

WHO WE ARE?

- From France
  - @OPALE SECURITY Company
  - IT Security & Embedded System Security
- Yann ALLAIN
  - 18 Years in IT security and electronic industry
  - Former CSO of application domain for an Hotel company
  - CEO and Owner of OPALE SECURTY
- Julien MOINARD
  - Electronic specialist
  - In charge of most technical implementation regarding this research

AGENDA

- Who we are
- Research context & goals
- Electronic 101 for Security Guys
- Proof of concept (soft, hard, …)
- Our experiments
- Results & Limits
- Further researches (Prospective)
- How to limit the risk
- Conclusion

Research context

- An another way to audit some Embedded system

Connecting
Consumer
Screen Phone
Mobile Phone
Set Top Box
Handheld Game
Game Console
Net TV
Auto PC
Smart Mobile Phone
DSC
Handheld PC
PDA
Thin Client
MP3
Palm PC
Computing

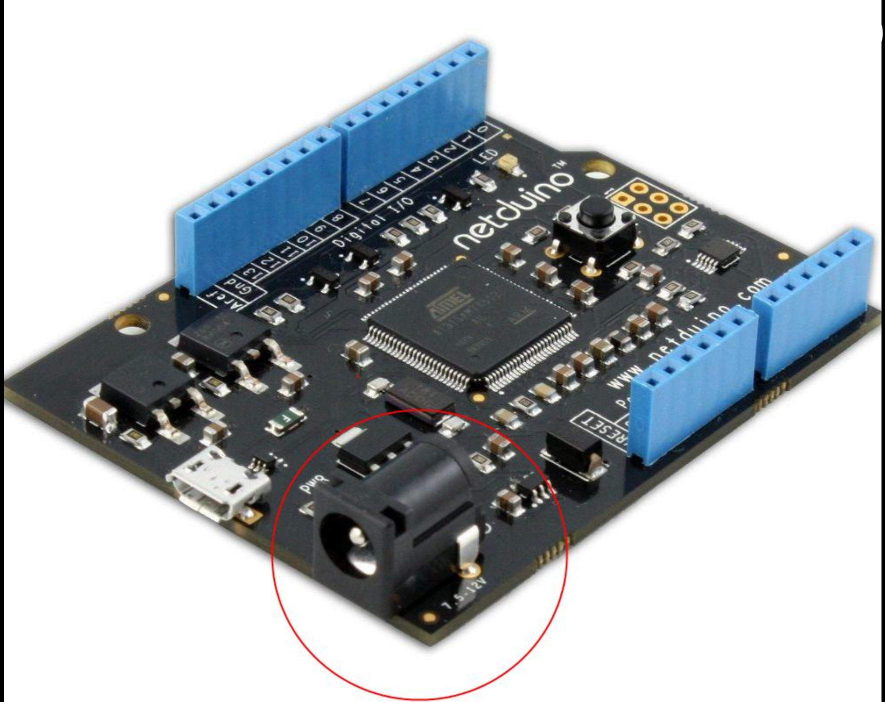
- Classical audit approach is done via
  - External pentest (IP Connexion, Web Interfaces…)
  - Hardware hacking stuff (Defeating anti tampering system, Opening the box
  - Etc…

- …but we want more…

Research context

- There always another access available on all Embedded system:
  - The electric power line !

rs is !



Research context

- As Security auditor, may we use this access to do something ?
- This our research & experimentation starting point
- Please remind that this is an ‘***in progress research project***’

So…

- As security guys, we wondered if

“Is there a way

to extract the code executed

on an embedded system

from its current/power consumption ?”

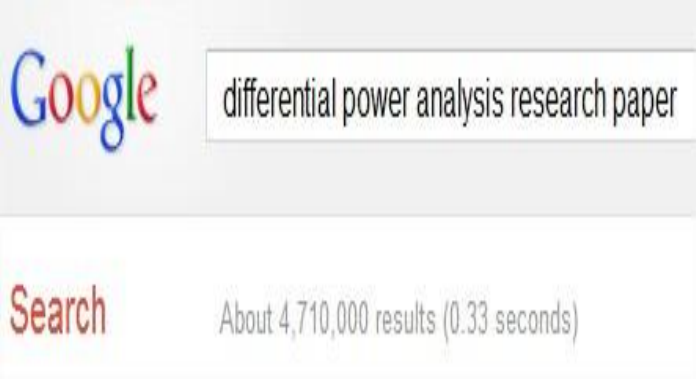
(≈ From the Power connector…)

Our wishlist

- Be pragmatic
- Keep it simple as possible
- No math and complex stuff
- Cheap approach (as much as possible)

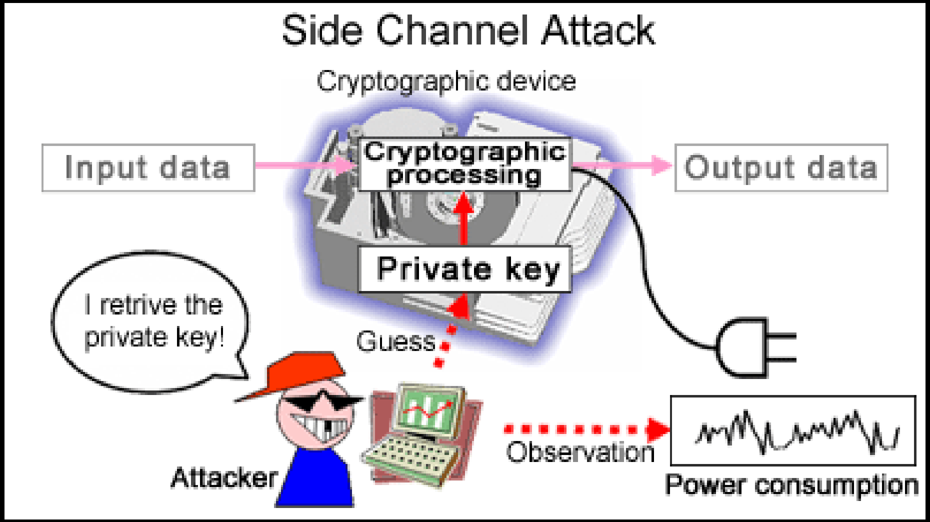
Existing research on this area?

- Yes…(many!) but with different goals
- Power analysis technics (DPA, SPA) and researchers seems to focus on extracting the cipher keys of sensitive device (Crypto system, Credit Card…)



Google differential power analysis research paper

Search About 4,710,000 results (0.33 seconds)



Existing research on thi

- But … Few papers related to cod
- We only find 3 available papers using the pow r consumption for finding instructi
  - –
  - barth)
  - Discovery of inform e encryption keys (Valette ,http://www.ssi.gouv.f rchive/fr/sciences/fichiers/lcr/dalemuva05.pdf)
  - Example adapted to JAVACARDS (Vermoen, http://ce.et.tudelft.nl/publicatio

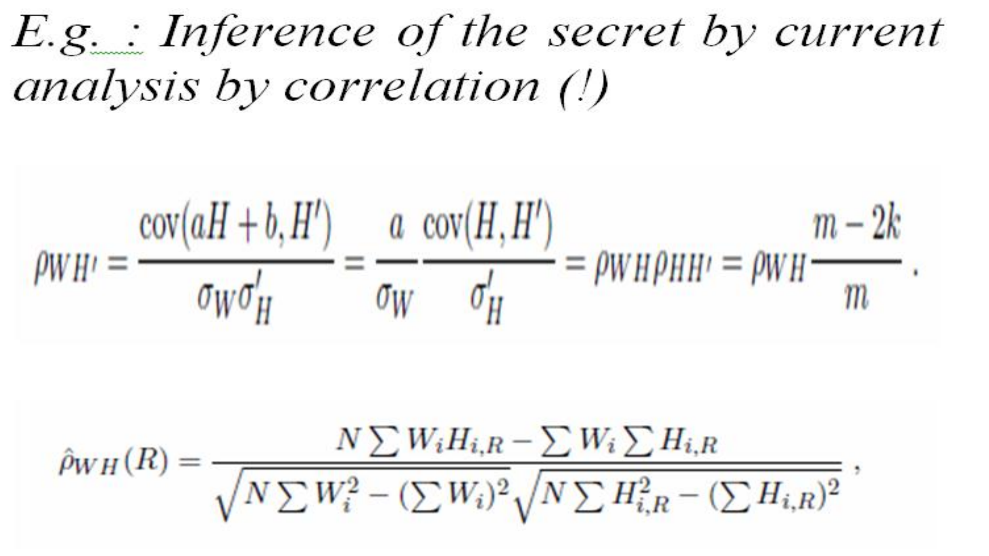
Cool ! . ..but researcher only focus on finding intructions…we need to access to Data also…(But great Paper!)

Too specific : Javacards

Some chapters dedicated to our goals but no so much information disclosed (Gouv.fr closed to ‘sort of’ military domain ?...)

Already existing research on this area?

- But these publications are full of mathematical formulae



E.g. : Inference of the secret by current analysis by correlation (!)

$$\rho_{WH'} = \frac{\operatorname{cov}(aH+b,H')}{\sigma_W\sigma'_H} = \frac{a}{\sigma_W}\frac{\operatorname{cov}(H,H')}{\sigma'_H} = \rho_{WH}\rho_{HH'} = \rho_{WH}\frac{m-2k}{m}.$$

$$\hat{\rho}_{WH}(R) = \frac{N\sum W_iH_{i,R} - \sum W_i\sum H_{i,R}}{\sqrt{N\sum W_i^2-(\sum W_i)^2}\sqrt{N\sum H_{i,R}^2-(\sum H_{i,R})^2}},$$

- which are more or less complex (***from our point of view***!)

- Not for us…. ;-)

Back to our goals…

Question

“What is the link between the power consumption

and instruction and data executed

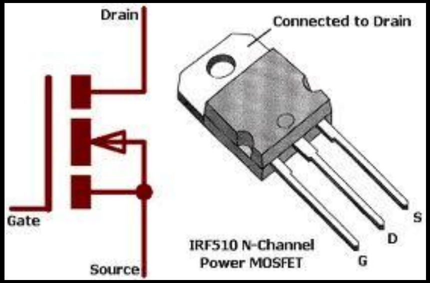
On most of embedded systems

based on microcontroller (or other stuff like that)?”

Answer

- A fondamental and basic electronic component….
- Used everywhere !
- Please gentlemen welcome to, our friends:

Transistors

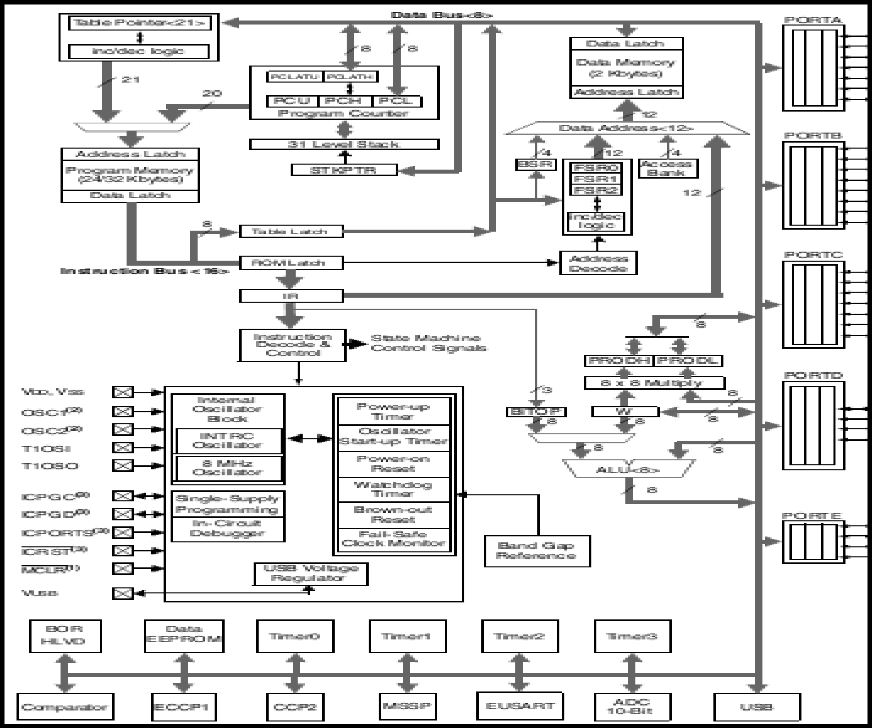


AGENDA

- Who we are
- Research context & goals
- **Electronic 101 for Security Guys**
- Proof of concept (soft, hard, …)
- Our experiments
- Results & Limits
- Further researches (Prospective)
- How to limit the risk
- Conclusion

Electronic 101

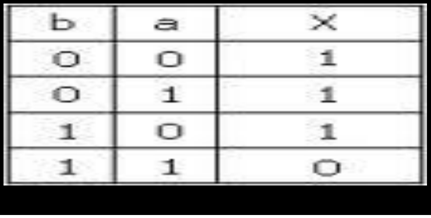
- Embedded systems are (could be) composed of microcontrollers (µC) that contain :
  - MEMORIES (Ram, Rom,..)
  - ALU (Arithmetic logic Unit)
  - TIMER (Counter)
  - SERIAL INTERFACES
  - I/O BUS (Latch)



Electronic 101

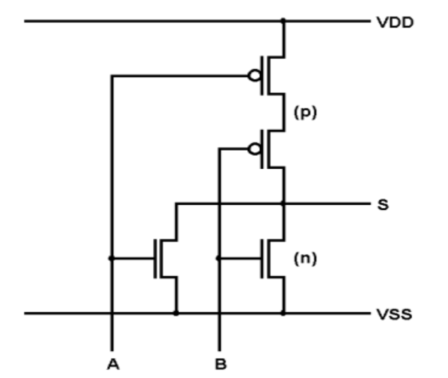
- Each basic functions included in µC are designed @electronic level with transistors
- For example , see how a "NAND" is designed @electronic level (simplification view of)

Logical view

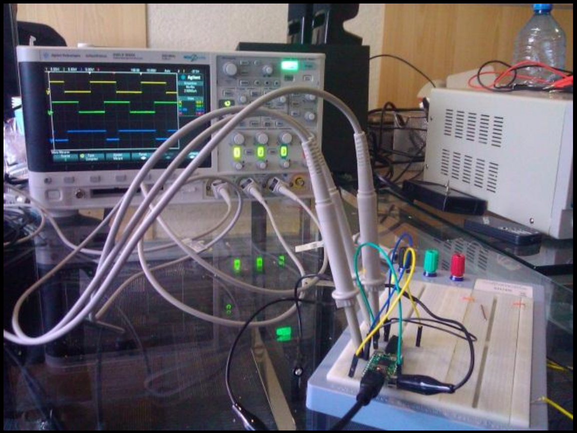


b	a	x
0	0	1
0	1	1
1	0	1
1	1	0

Electronic view (used only few transistors)

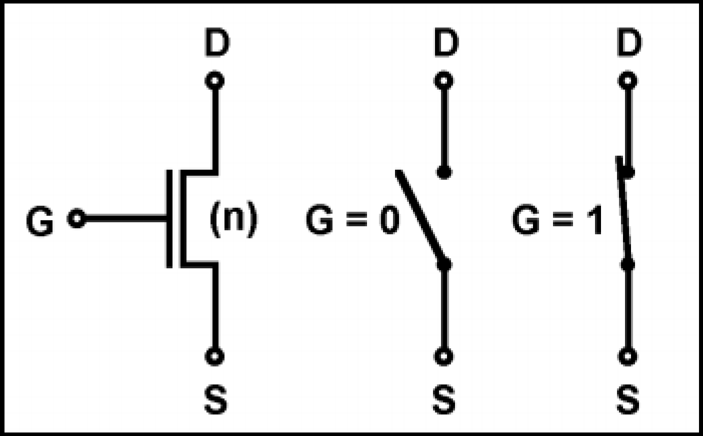


Physical Electric signal associated



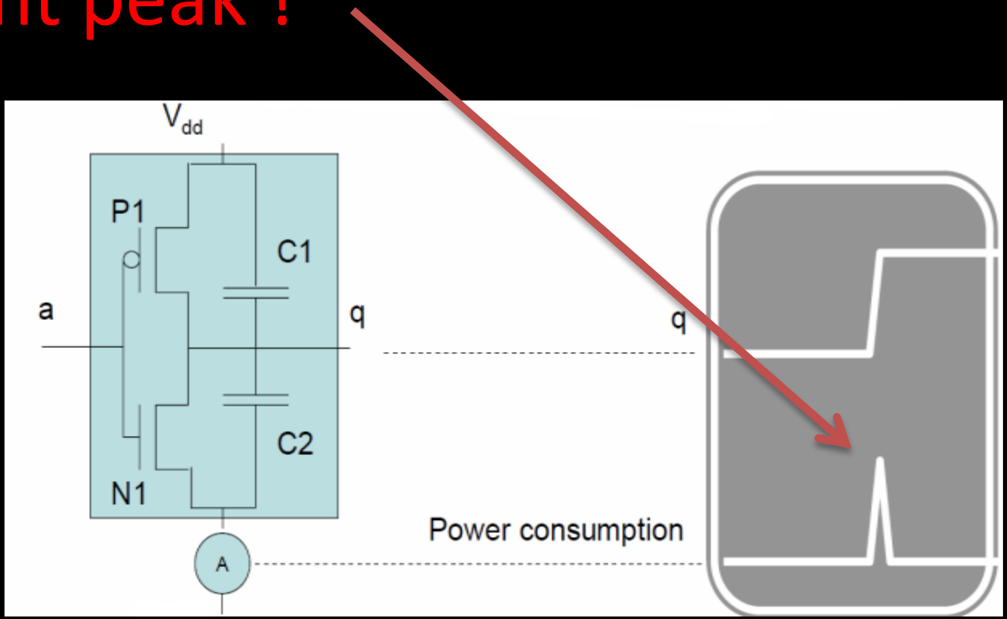
Electronic 101

- When a transistor "process" a bit @ physical level (Current, Voltage) , it "commutes"
- Transistor = sort of digital switch



Electronic 101

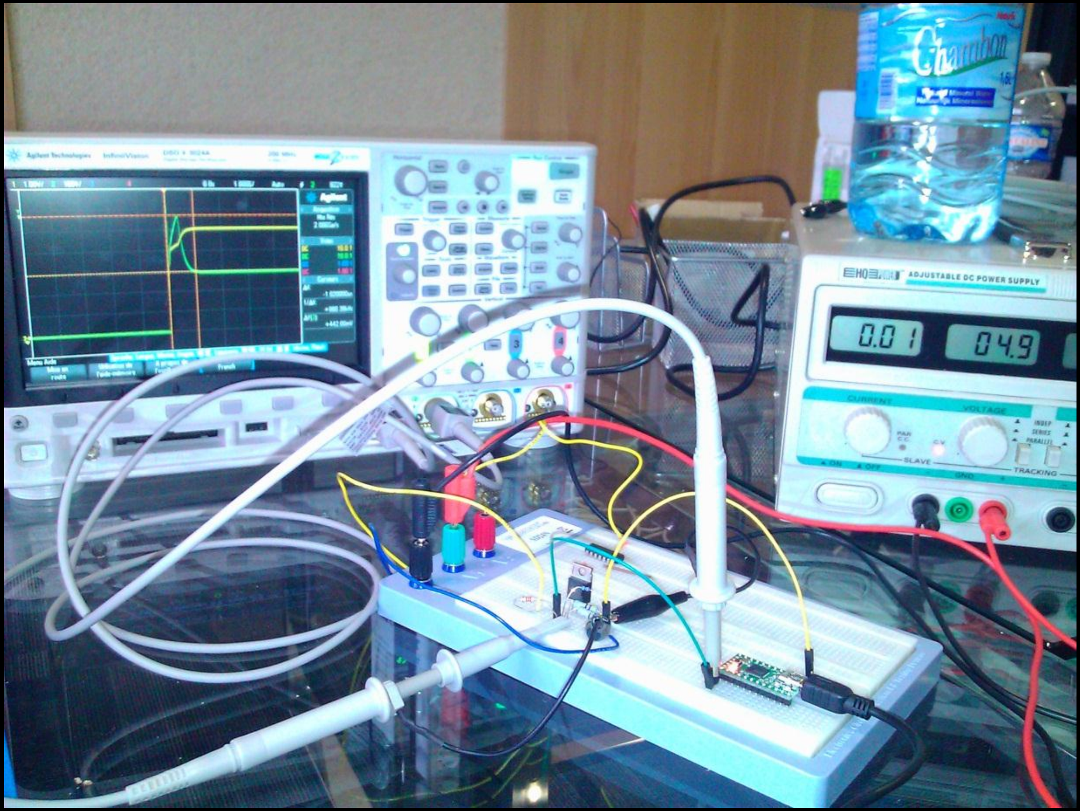
- When a Transistor "commutes", there is a current peak !



- Let see what going on in practice (Labs…)

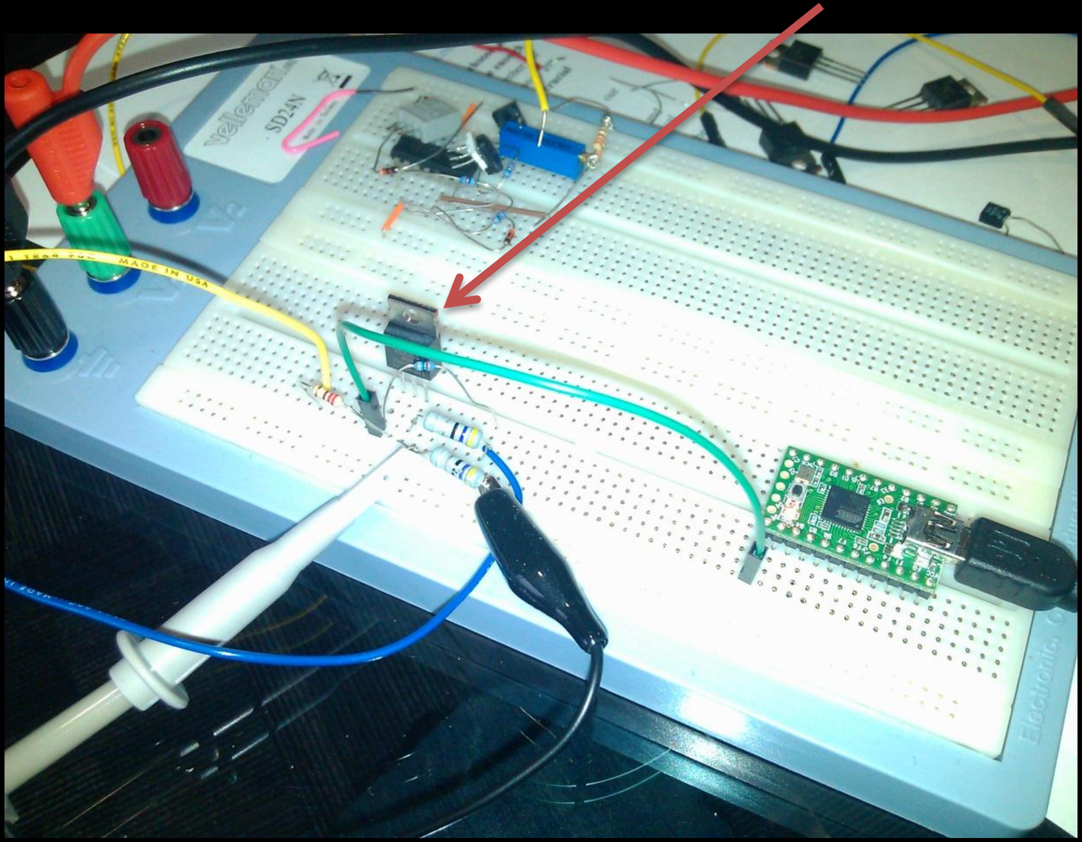
Electronic 101

- Labs #1 – Screenshot 1 – Hardware stuff



Electronic 101

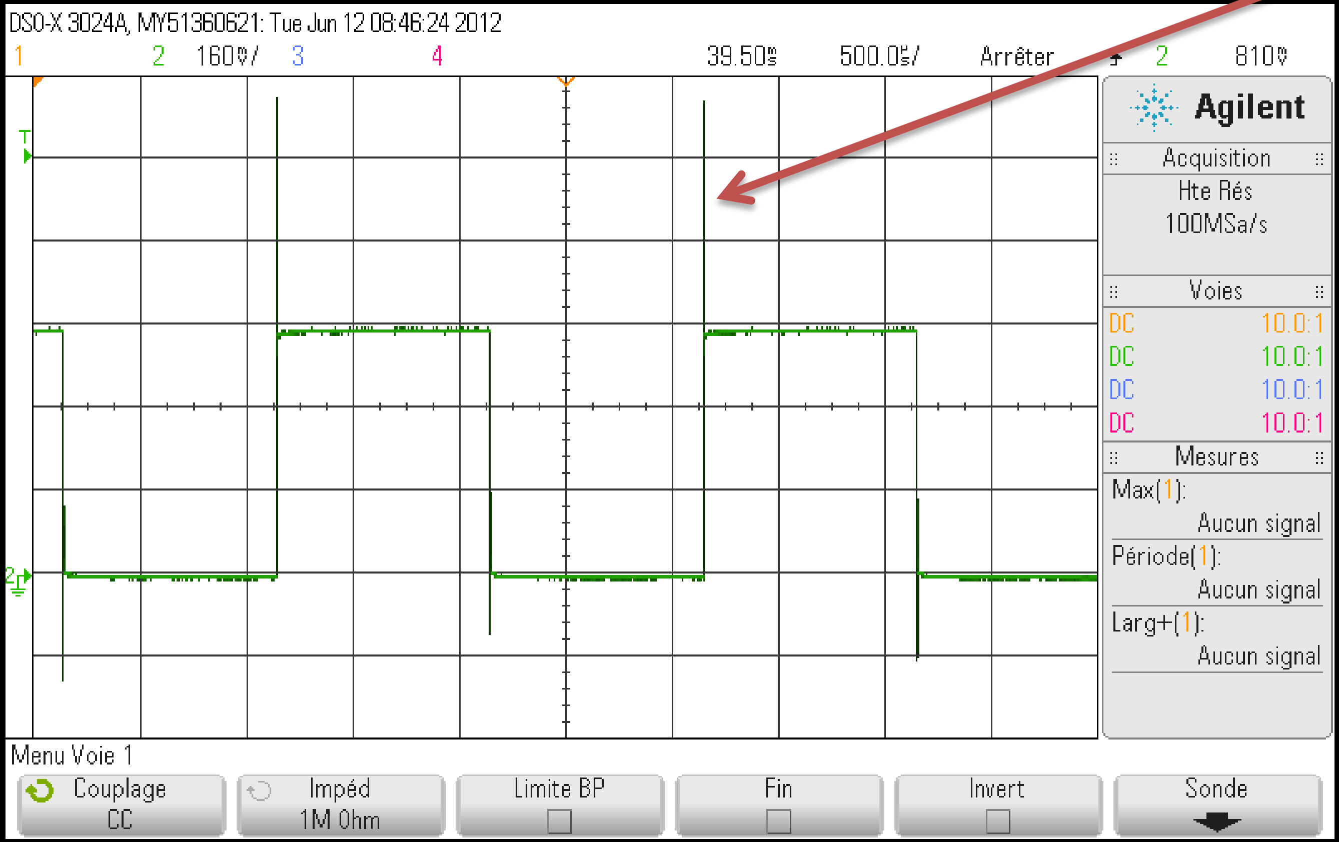
- Labs #1 – Screenshot 2 – One Transistor !



Electronic 101

- Labs #1 – Screenshot 3

current peak !



DSO-X 3024A, MY51360621: Tue Jun 12 08:46:24 2012

1 2 160mV/ 3 4 39.50µs 500.0µs/ Arrêter 2 810mV

Acquisition
Hte Rés
100MSa/s

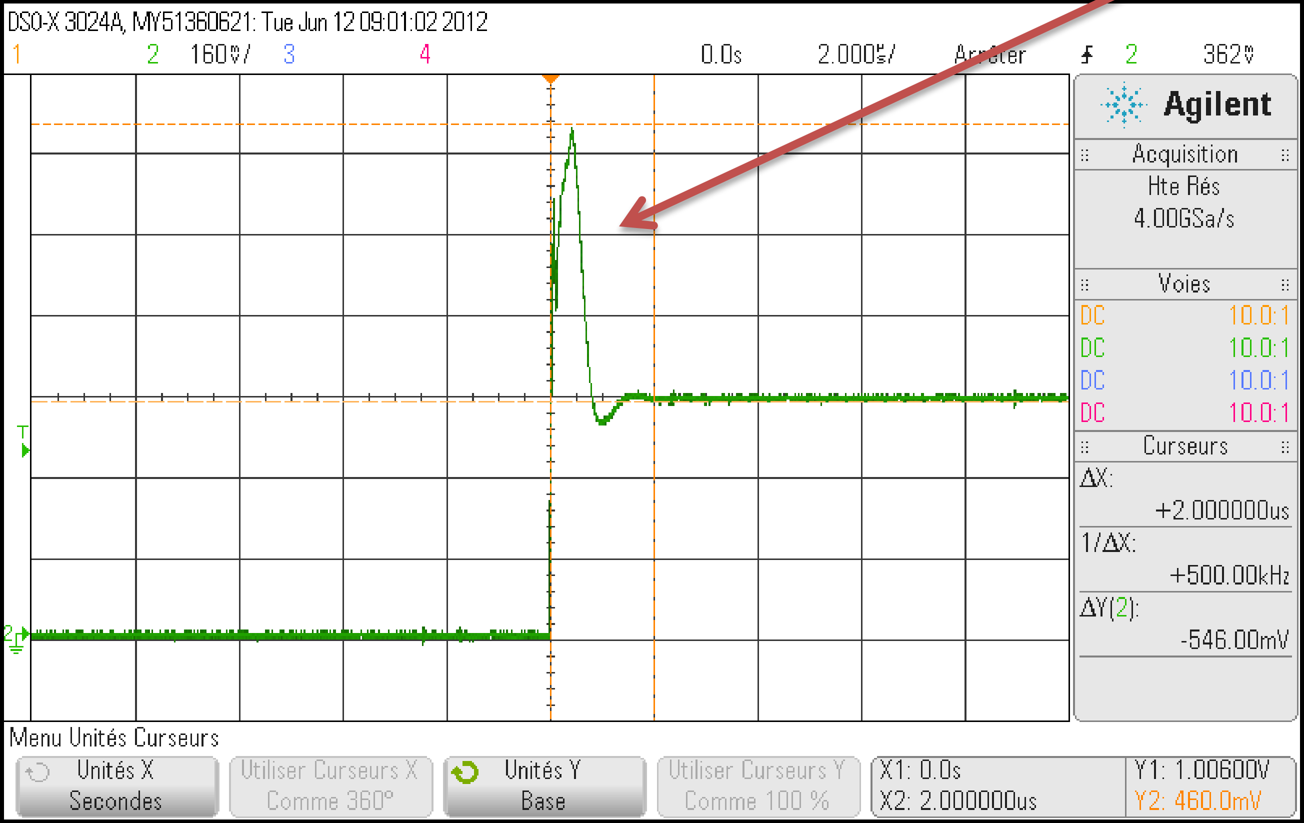
Voies
DC 10.0:1
DC 10.0:1
DC 10.0:1
DC 10.0:1

Mesures
Max(1):
Aucun signal
Période(1):
Aucun signal
Larg+(1):
Aucun signal

Menu Voie 1

Electronic 101

- Labs #1 – Screenshot 4



Zoom of current peak !

DSO-X 3024A, MY51360621: Tue Jun 12 09:01:02 2012

1 2 160mV/ 3 4 0.0s 2.000µs/ Arrêter ƒ 2 362mV

Acquisition
Hte Rés
4.00GSa/s

Voies
DC 10.0:1
DC 10.0:1
DC 10.0:1
DC 10.0:1

Curseurs
ΔX:
+2.000000us
1/ΔX:
+500.00kHz
ΔY(2):
-546.00mV

Menu Unités Curseurs

X1: 0.0s
X2: 2.000000us
Y1: 1.00600V
Y2: 460.0mV

Brief

- Transistors everywhere in µC
- When a transistor “process” a bit, there is a current peak

“We just find the link between the power consumption and bits processed”

- *Information leakage from power consumption validated !*

☺

AGENDA

- Who We Are
- Research context & goals
- Electronic 101 for Security Guys
- **Proof of concept (soft, hard, …)**
- Our experiments
- Results & Limits
- Further researches (Prospective)
- How to limit the risk
- Conclusion

Proof of concept

- How to move from one bit grabbed (step1) to a set of data & instructions code (step2) with our approach ?

- We have designed a proof of concept tool to analyze the electrical current consumption of embedded systems to extract the code it executes

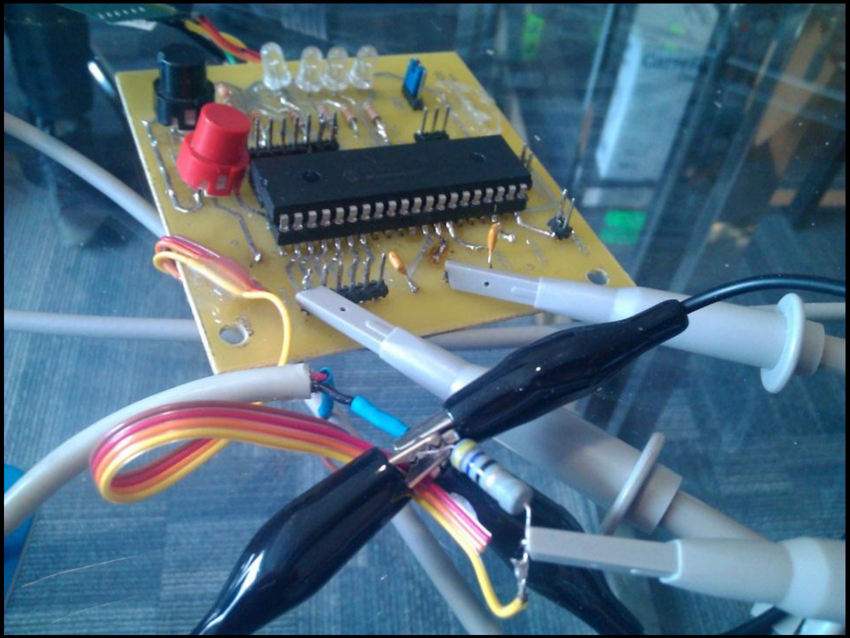
Proof of concept

- We need to acquire more bits…via a current consumption analysis

- “Acquiring current consumption” : How?

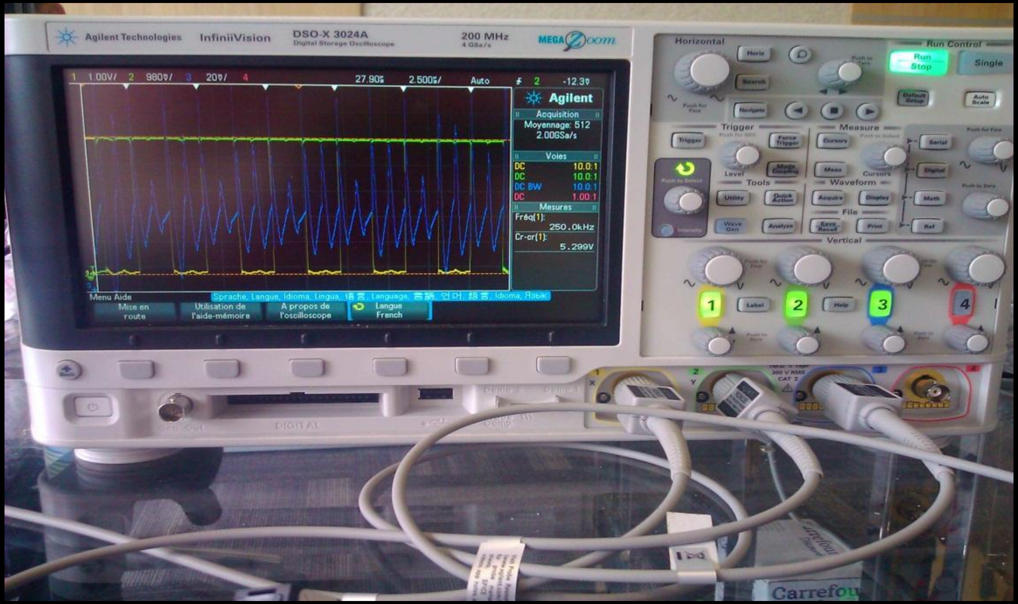
Proof of concept

- What we need : A “homemade” embedded system (the target…)
    - Based on PIC18F4620 µC



Proof of concept

– What we need : An Agilent oscilloscope for acquiring current consumption
  - AGILENT Dso3024a



Proof of concept

- What we need : A programmer /Debugger (Microchip Real Ice)

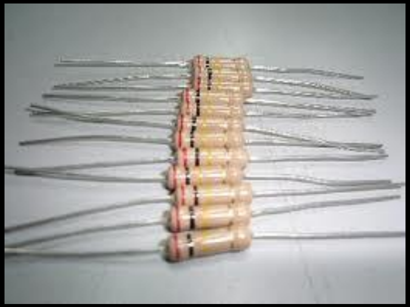
Proof of concept

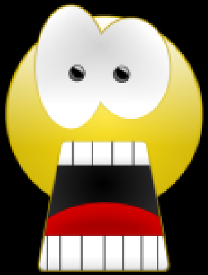
- What we need : A current probe
  - Very expensive Professional tools (magnetic or electromagnetic current probe) > 400$ each

  Or

  - a simple resistor which cost less than 1 $

  - We choose the resistor !



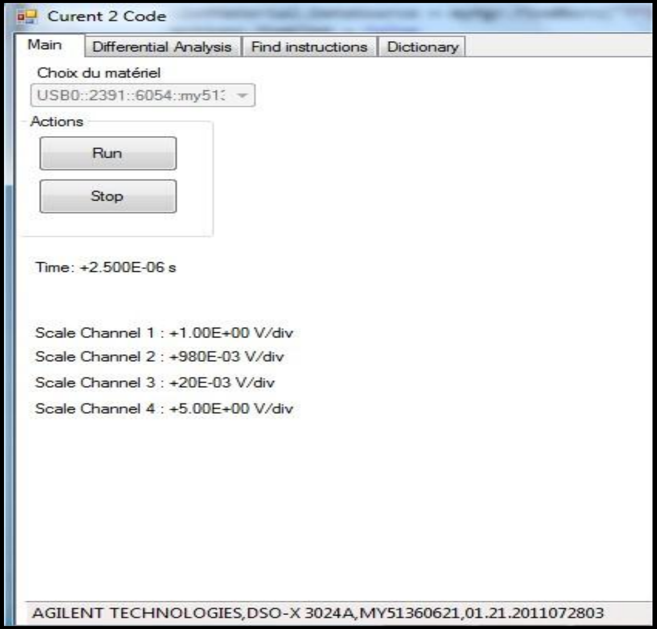


Proof of concept

- What we need : A bit of software
  - Homemade code (VB.NET…sorry ☺) used to control and pilot the oscilloscope
  - The code used the Standard protocol: VISA COM 3.0
  - It’s a Free Library that let us communicate with agilent oscilloscope with simple set of commands
    - Get datum measurement, Launch voltage or current acquisition process, Send numerical value of current acquired,…

Proof of concept

- What we need : A GUI



Choix du matériel

USB0::2391::6054::my51:

Actions

Time: +2.500E-06 s

Scale Channel 1 : +1.00E+00 V/div

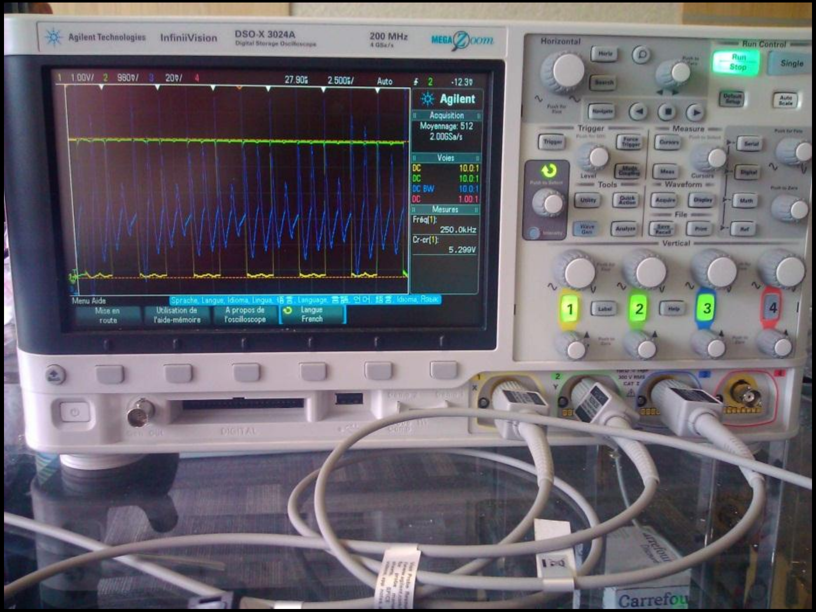
Scale Channel 2 : +980E-03 V/div

Scale Channel 3 : +20E-03 V/div

Scale Channel 4 : +5.00E+00 V/div

AGILENT TECHNOLOGIES,DSO-X 3024A,MY51360621,01.21.2011072803

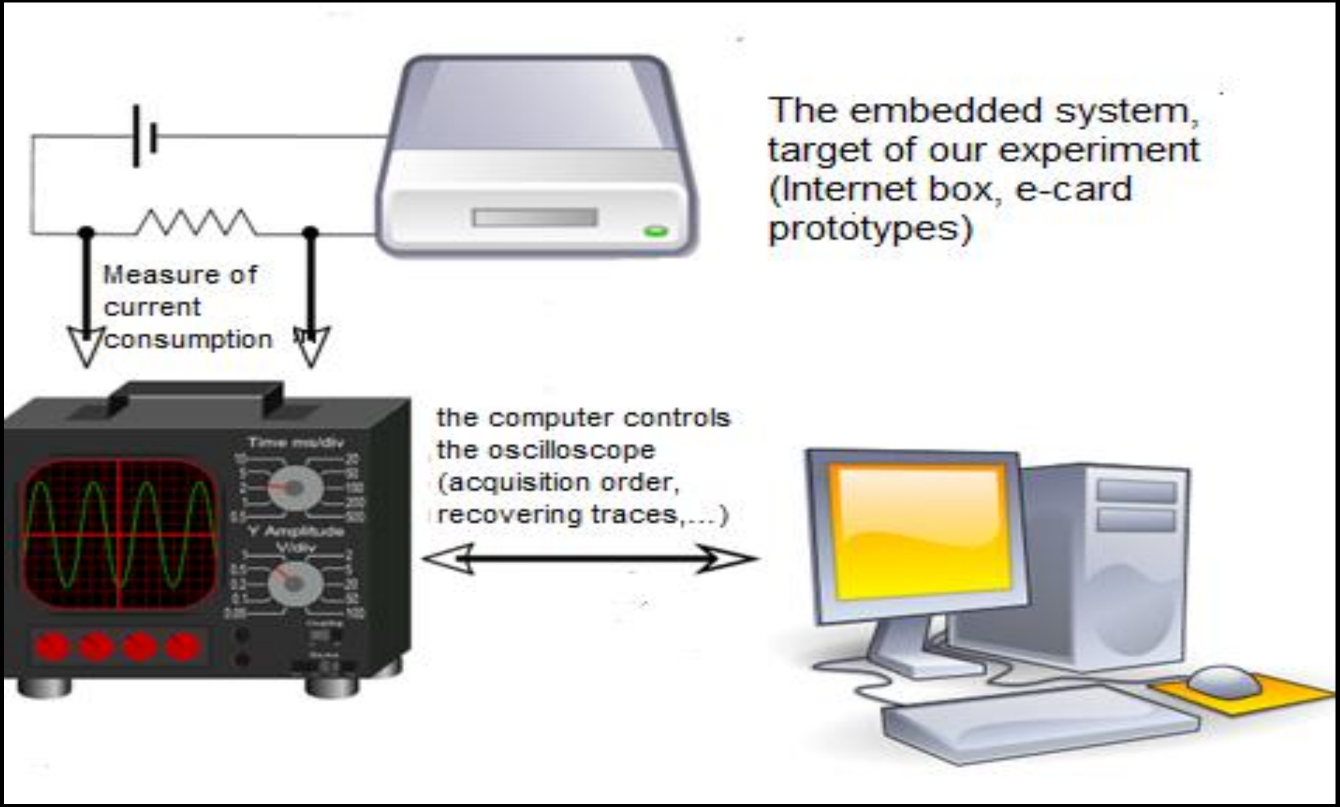
Command/Data



GUI of our Proof
of concept tool

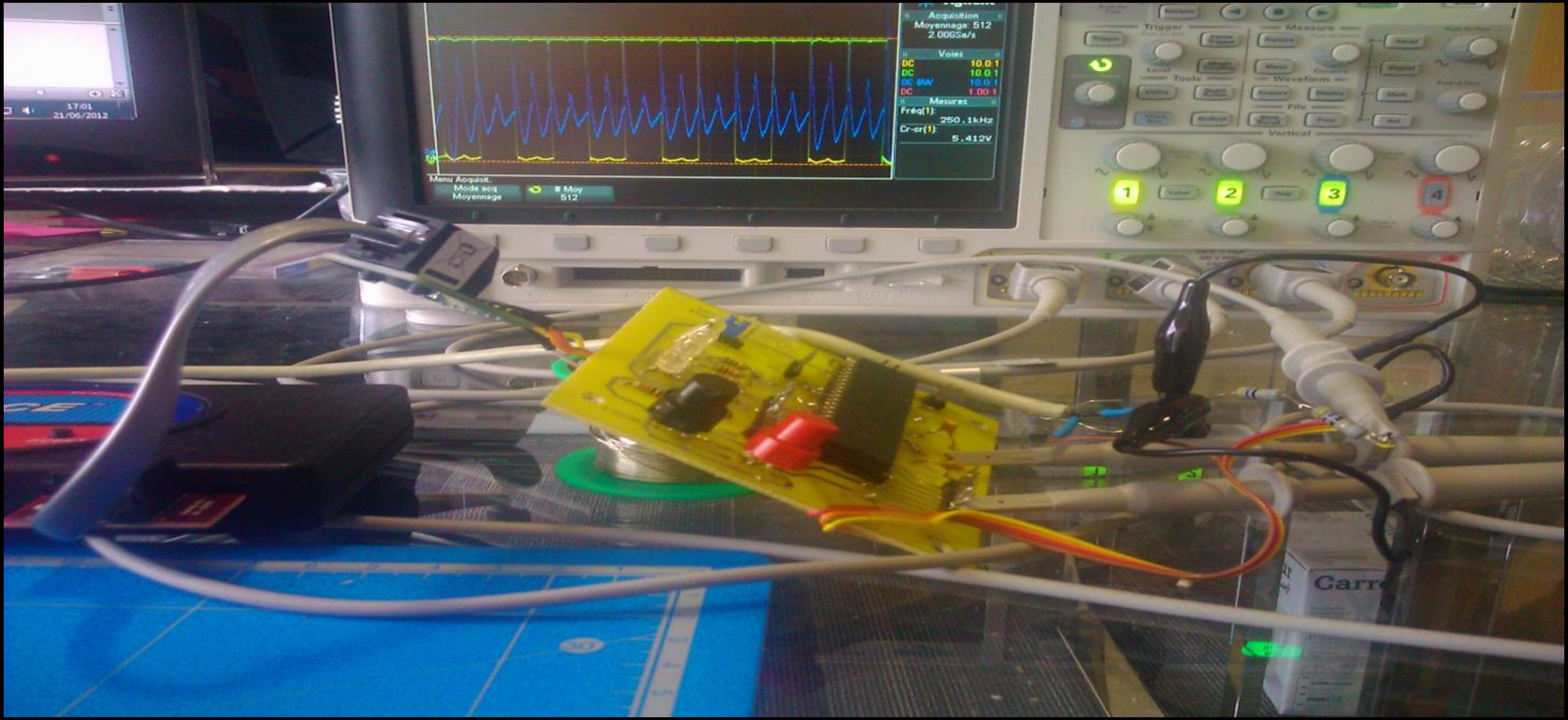
Proof of concept

- Our acquisition chain looks like that :


The embedded system, target of our experiment (Internet box, e-card prototypes)
Measure of current consumption
the computer controls the oscilloscope (acquisition order, recovering traces,....)
Time ms/div
Y Amplitude V/div

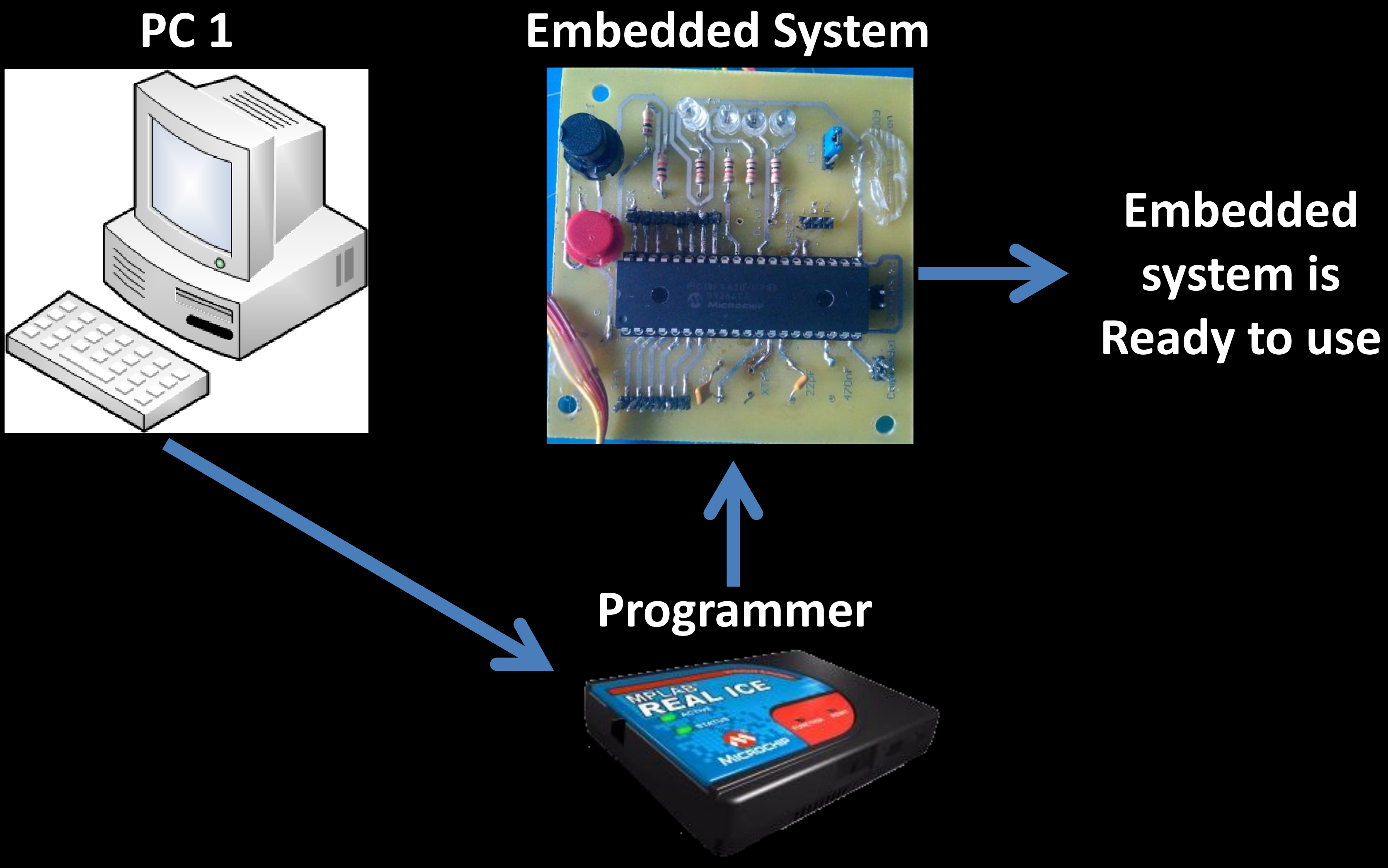
Proof of concept

- In practice, it looks like that…



How we proceed to grab the current and extract the code?

<u>Step 1 send a dummy code to µC</u>



PC 1

Embedded System

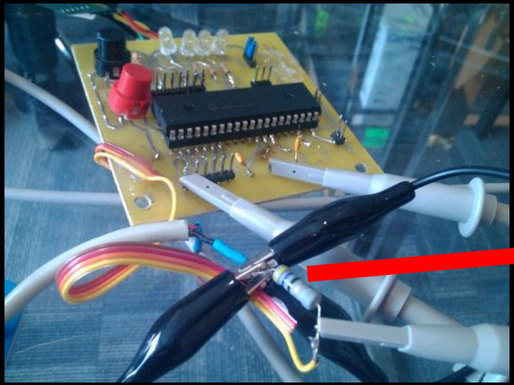
Embedded system is Ready to use

Programmer

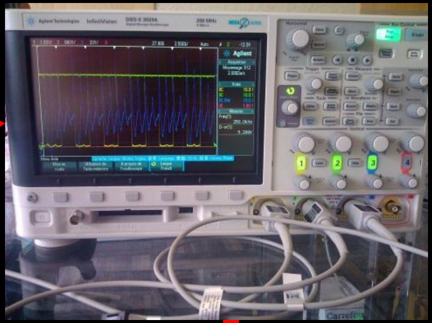
Proof of concept

Step 2 , In lab

Embedded System with probes

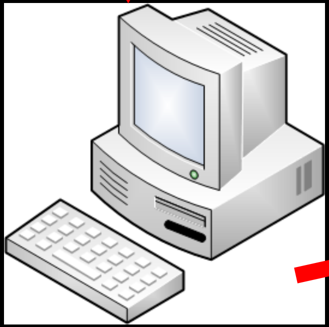


Oscilloscope (Measure)

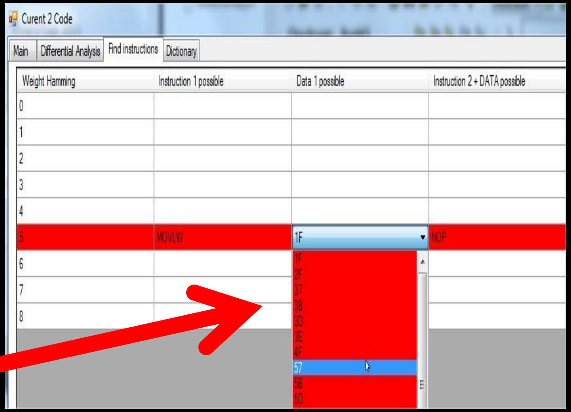


Current Consumption

PC 2 (Lab machine)



Our tool try to find instruction & data executed from the current consumption



Weight Hamming	Instruction 1 possible	Data 1 possible	Instruction 2 + DATA possible
0			
1			
2			
3			
4			
5	MOVLW	1F	NOP
6			
7			
8			

AGENDA

- Who We Are
- Research context & goals
- Electronic 101 for Security Guys
- Proof of concept (soft, hard, …)
- **Our experiments**
- Results & Limits
- Further researches (Prospective)
- How to limit the risk
- Conclusion

Our Experiments

#1: Does the code really impacts the power consumption?

#2: Do a MOVLW 0xFF & a MOVLW 0x00 lead to measurable differences in power analysis?

#3: Why µC’s instructions Pipeline impact current consumption?

#4: How to overcome Pipeline issues for our goals?

#5: Could we create a (sort of) ‘disassembler’ over electricity?

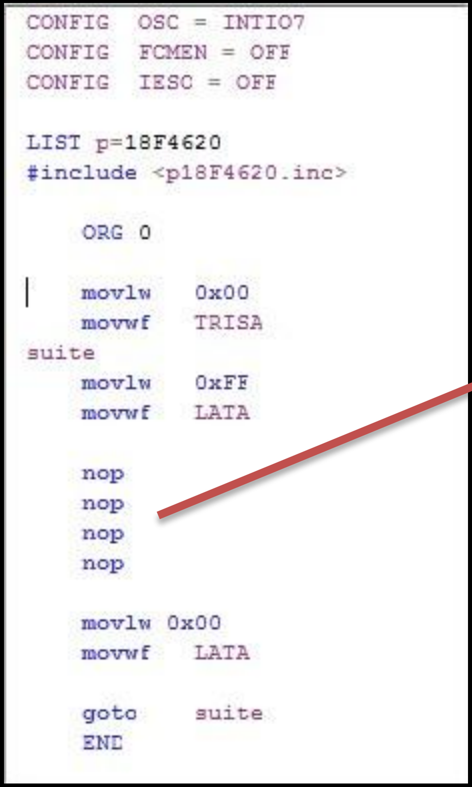
Does the code really impacts the power consumption?

(Experiment #1)

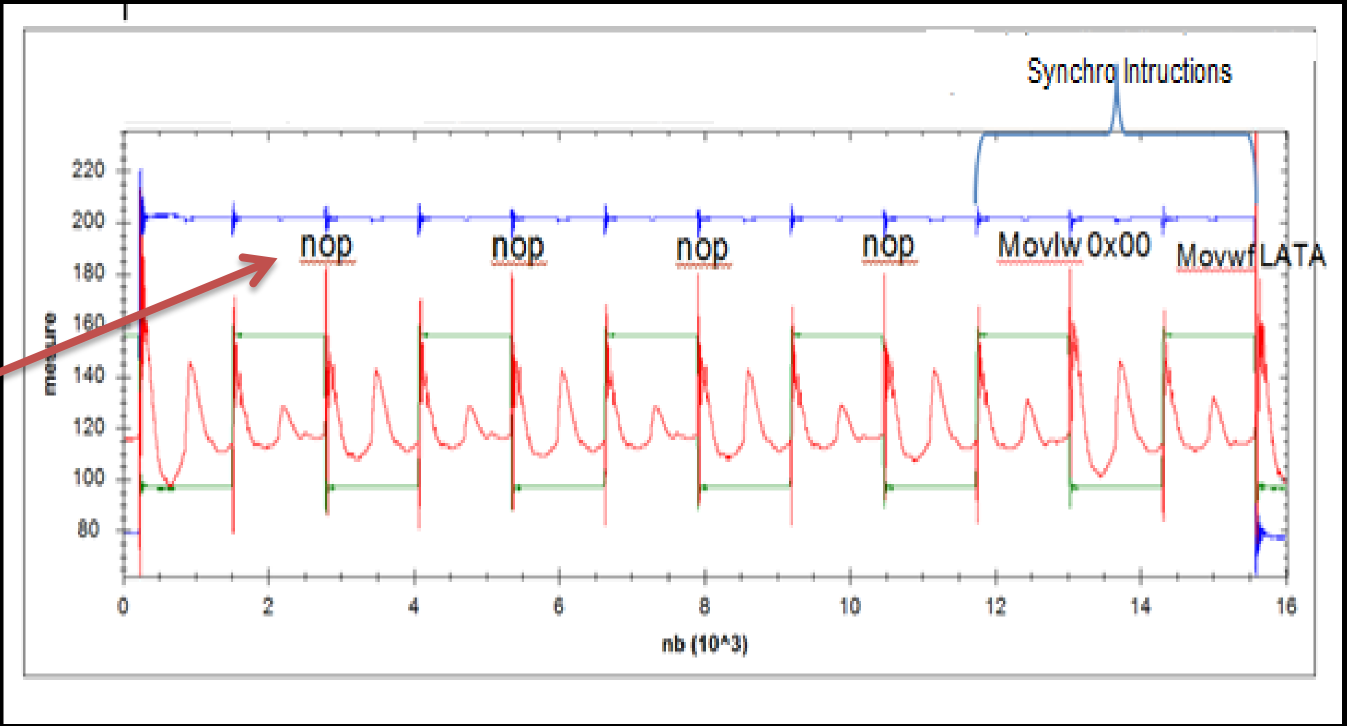
Does the code really impacts the power consumption?

(Experiment #1)

- Result #1 : We have a current consumption related with nop instructions



```
CONFIG   OSC = INTIO7
CONFIG   FCMEN = OFF
CONFIG   IESC = OFF

LIST p=18F4620
#include <p18F4620.inc>

    ORG 0

    movlw    0x00
    movwf    TRISA
suite
    movlw    0xFF
    movwf    LATA

    nop
    nop
    nop
    nop

    movlw 0x00
    movwf    LATA

    goto     suite
    END
```



In Red → Current during the execution
In Blue → Synchronization signal
In Green → Clock embedded system

Do a MOVLW 0xFF & a MOVLW 0x00
lead to measurable differences
in power analysis?

(Experiment #2)

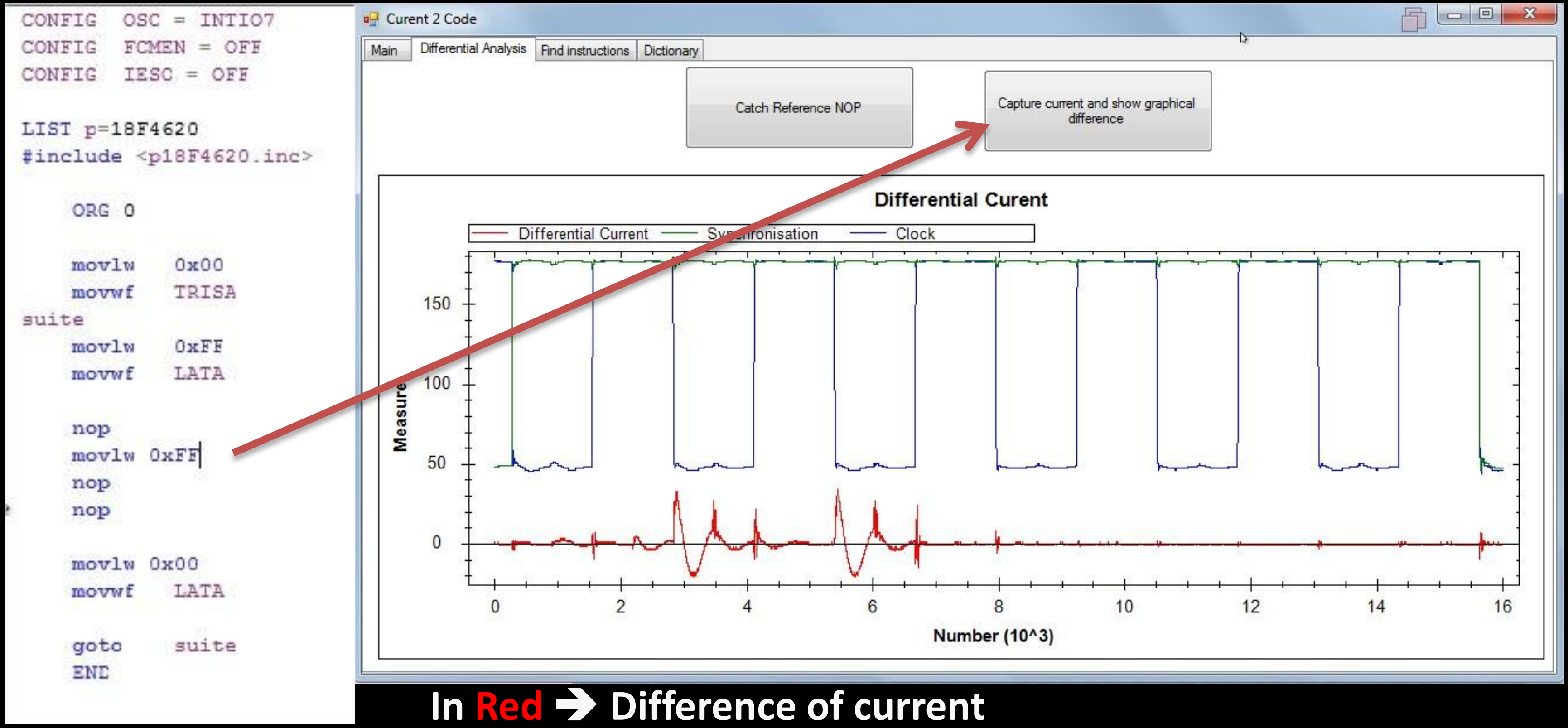
Do a MOVLW 0xFF & a MOVLW 0x00 lead to measurable differences in power analysis? (Experiment #2)

- Note : to limit impacts of parasites, our system take differential analysis

- @First time, we measured the difference between
  - Current consumption of 4 nop instructions
  - Current consumption of **movlw 0xFF** with 3 nop
- @Second time, we measured the difference between
  - Current consumption of 4 nop instructions
  - Current consumption of **movlw 0x00** with 3 nop

Do a MOVLW 0xFF & a MOVLW 0x00 lead to measurable differences in power analysis? (Experiment #2)

- Result #2 : Current Trace related to Movlw 0xFF

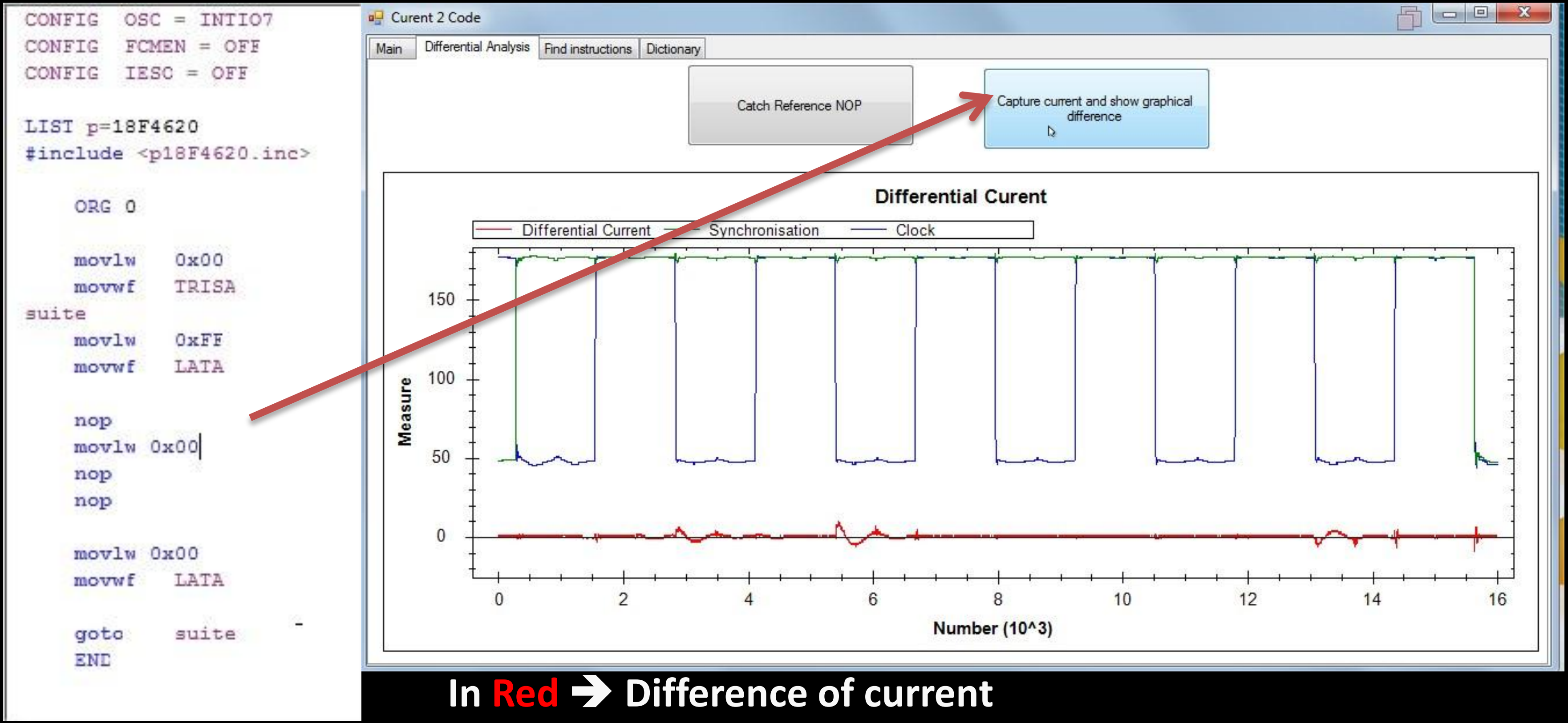


```
CONFIG   OSC = INTIO7
CONFIG   FCMEN = OFF
CONFIG   IESO = OFF

LIST p=18F4620
#include <p18F4620.inc>

    ORG 0

    movlw    0x00
    movwf    TRISA
suite
    movlw    0xFF
    movwf    LATA

    nop
    movlw 0xFF
    nop
    nop

    movlw 0x00
    movwf    LATA

    goto     suite
    END
```

In Red → Difference of current
In Blue → Synchronization signal
In Green → Clock embedded system

Do a MOVLW 0xFF & a MOVLW 0x00 lead to measurable differences in power analysis?
(Experiment #2)

- Result #2 : Current Trace related to Movlw 0x00



```
CONFIG   OSC = INTIO7
CONFIG   FCMEN = OFF
CONFIG   IESC = OFF

LIST p=18F4620
#include <p18F4620.inc>

     ORG  0

     movlw    0x00
     movwf    TRISA
suite
     movlw    0xFF
     movwf    LATA

     nop
     movlw  0x00
     nop
     nop

     movlw  0x00
     movwf    LATA

     goto     suite
     END
```

In Red → Difference of current

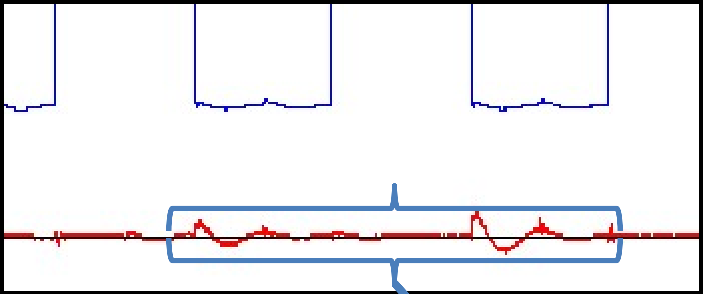
In Blue → Synchronization signal

In Green →Clock embedded system

Do a MOVLW 0xFF & a MOVLW 0x00 lead to measurable differences in power analysis? (Experiment #2)

- Result #2 : We have a correlation between different value of data and amplitude of current consumption

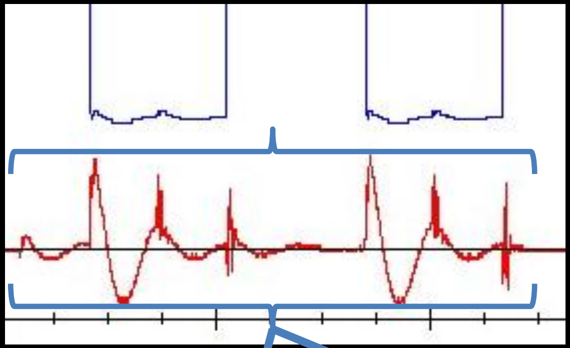
MOVLW 0x00



Encoding of the ***movlw 0x00*** instruction

➔ 0000 1110 0000 0000

MOVLW 0xFF



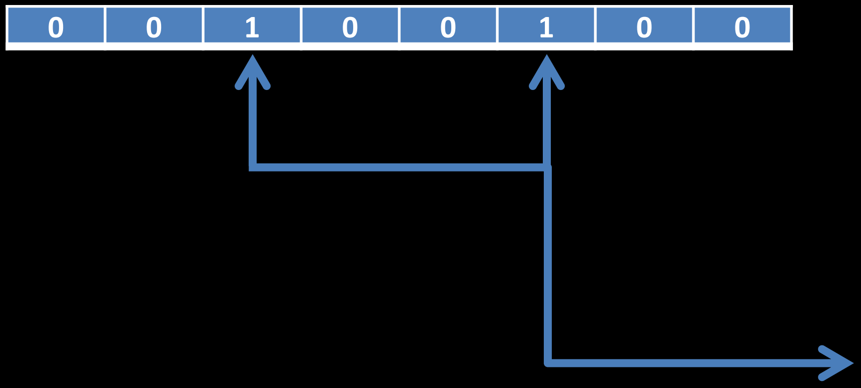
Encoding of the ***movlw 0xFF*** instruction

➔ 0000 1110 1111 1111

More bits = 1 -> More current consumption !

Do a MOVLW 0xFF & a MOVLW 0x00 lead to measurable differences in power analysis? (Experiment #2)

- The current value measured depend on the hamming weight groups of the data & instruction processed
- Example below (0x24 is in a hamming group of 2)



0	0	1	0	0	1	0	0

Hamming Group	Number of instruction or data value by hamming groups
0	1
1	8
2	28
3	56
4	70
5	56
6	28
7	8
8	1

Do a MOVLW 0xFF & a MOVLW 0x00 lead to measurable differences in power analysis? (Experiment #2)

- The hamming weight groups limits!

Description	Instruction	Coding instruction	Instruction Hamming Weight
No Operation	NOP	0000 0000	0
Multiply W with f	MULWF	0000 0010	1
Subtract W from Literal	SUBLW	0000 1000	1
Negate f	NEGF	0110 1100	4
Move W to f	MOVWF	0110 1110	5
Move Literal to W	MOVLW	0000 1110	3
Set f	SETF	0110 1000	3

Some instructions have the same Hamming weight (Collision) so we don’t able to differentiate MOVLW and SETF for example. It’s a limit of our analyze.

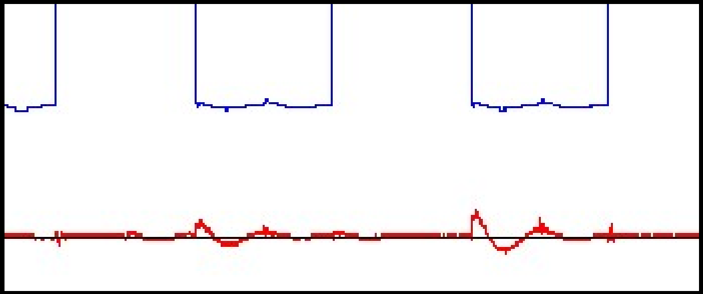
Why µC's instructions Pipeline impact current consumption?

(Experiment #3)

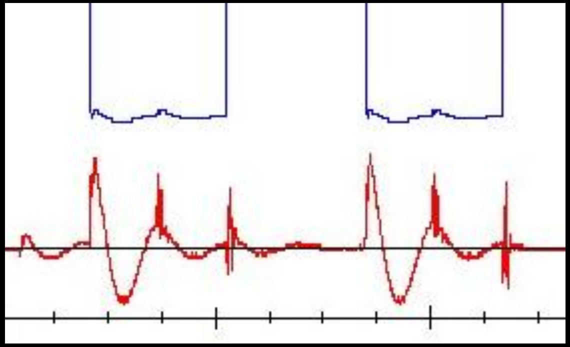
Why µC's instructions Pipeline impact current consumption? (Experiment #3)

- Result of our 3rd experimentation

MOVLW 0x00



MOVLW 0xFF



- But why we have two overshoots of current when the code only have one instruction that has been changed ?

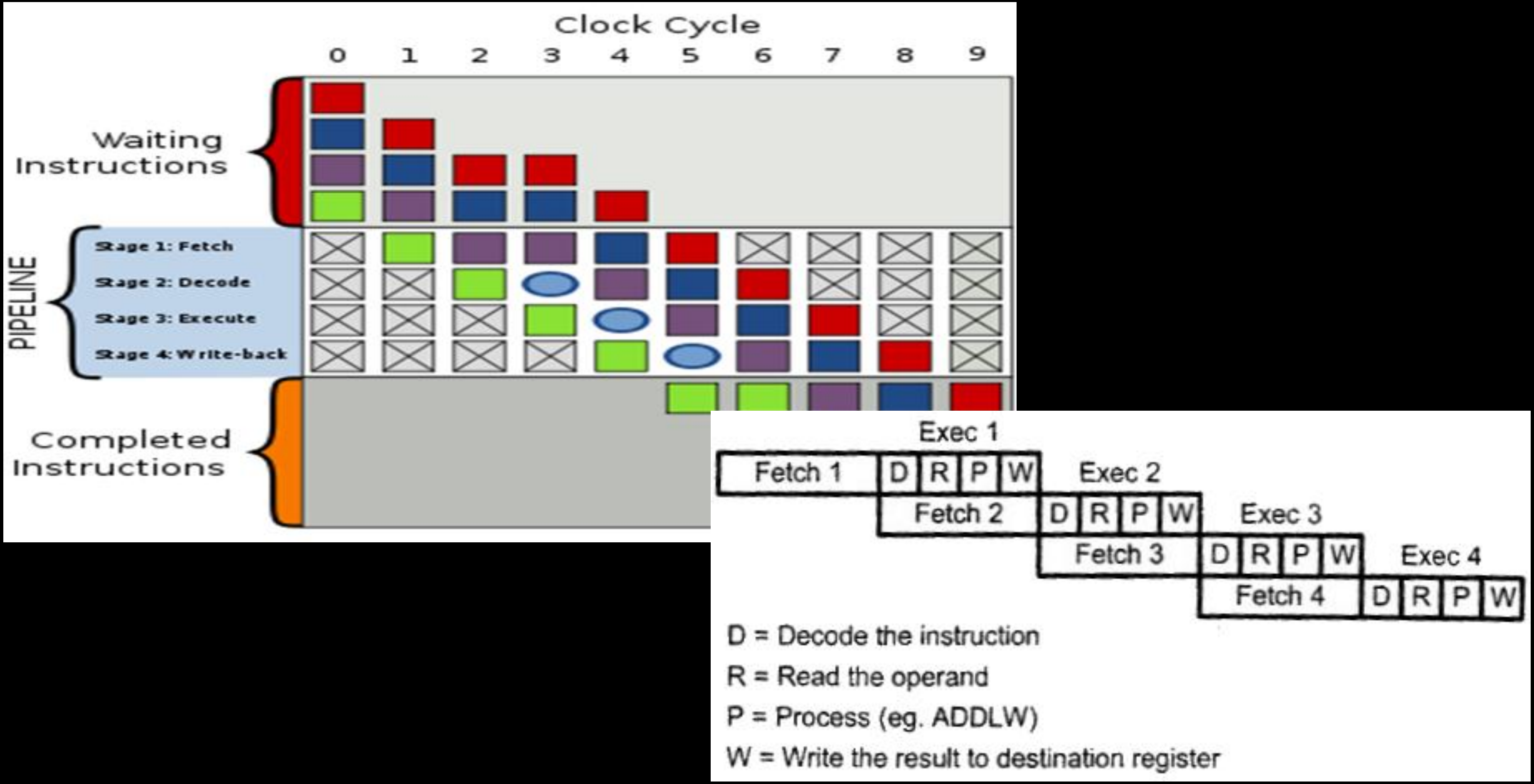
Why µC’s instructions Pipeline impact current consumption? (Experiment #3)

- Influence of Pipeline

C1	C2	C3	C4
Decoding	Read data here 0x00 (movlw 0x00)	ALU Calculation	ALU write the word in registers

Why µC's instructions Pipeline impact current consumption? (Experiment #3)

- Influence of Pipeline



D = Decode the instruction

R = Read the operand

P = Process (eg. ADDLW)

W = Write the result to destination register

Why µC's instructions Pipeline impact current consumption? (Experiment #3)

- Influence of Pipeline

Pipeline is not our friend because the current consumption of next instruction depend of previous instructions.

How to overcome Pipeline issues for our goals?

(Experiment #4)

How to overcome Pipeline issues for our goals? (Experiment #4)

- The main idea is use the principal of pre-calculated hash table

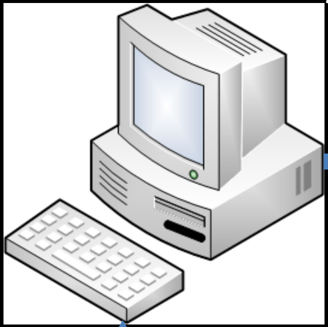
- The idea is to memorize a signature of electricity consumption for each pair of consecutive instructions in an exhaustive way. The idea is to create a sort of dictionary.

- We can now compare the current consumption of any (uncontrolled) executed code with the dictionary

How to overcome Pipeline issues for our goals? (Experiment #4)

- Generation of the dictionary

PC 2 (Lab machine)

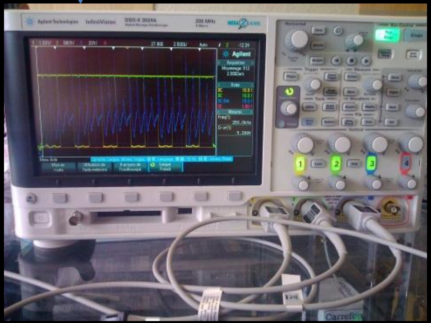


Send code with hamming code

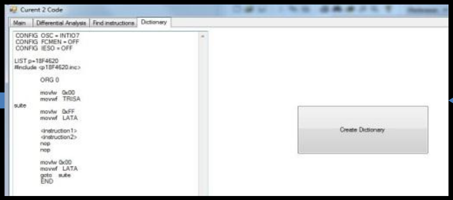
Programmer

Embedded System

Current Consumption

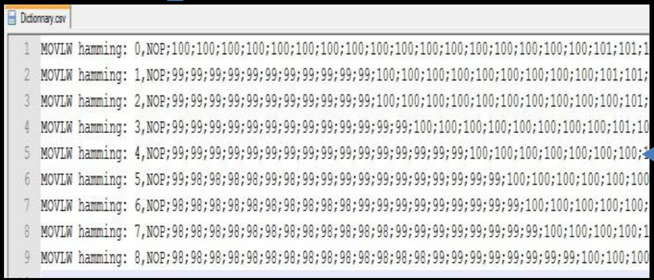


Oscilloscope (Measure)



```
 CONFIG  OSC = INTIO7
 CONFIG  FCMEN = OFF
 CONFIG  IESO = OFF

LIST p=18F4620
#include <p18F4620.inc>

         ORG 0

         movlw   0x00
         movwf   TRISA
suite
         movlw   0xFF
         movwf   LATA

         <instruction1>
         <instruction2>
         nop
         nop

         movlw 0x00
         movwf   LATA
         goto    suite
         END
```

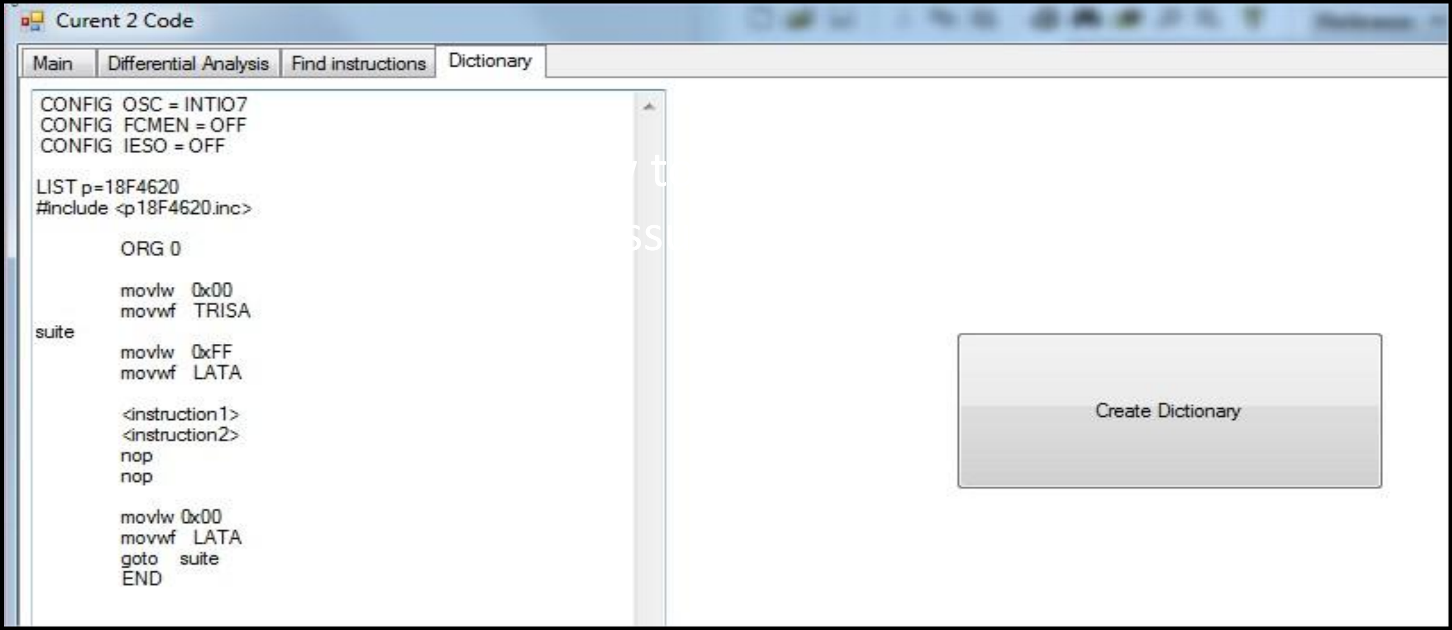
Save a dictionary



```
MOVLW hamming: 0,NOP;100;100;100;100;100;100;100;100;100;100;100;100;100;100;100;100;100;101;101;1
MOVLW hamming: 1,NOP;99;99;99;99;99;99;99;99;99;99;99;100;100;100;100;100;100;100;100;100;101;101;
MOVLW hamming: 2,NOP;99;99;99;99;99;99;99;99;99;99;99;100;100;100;100;100;100;100;100;100;100;101;
MOVLW hamming: 3,NOP;99;99;99;99;99;99;99;99;99;99;99;99;100;100;100;100;100;100;100;101;10
MOVLW hamming: 4,NOP;99;99;99;99;99;99;99;99;99;99;99;99;99;99;99;100;100;100;100;100;100;100;1
MOVLW hamming: 5,NOP;99;98;98;98;98;99;98;99;99;99;99;99;99;99;99;99;99;99;100;100;100;100;100;100
MOVLW hamming: 6,NOP;98;98;98;98;98;98;98;98;98;98;99;99;99;99;99;99;99;99;99;100;100;100;100;100;
MOVLW hamming: 7,NOP;98;98;98;98;98;98;98;98;98;98;98;98;99;99;99;99;99;99;99;99;100;100;100;100;1
MOVLW hamming: 8,NOP;98;98;98;98;98;98;98;98;98;98;98;98;98;98;99;99;99;99;99;99;99;99;100;100;100
```

How to overcome Pipeline issues for our goals? (Experiment #4)

- One button in our GUI ☺



```
CONFIG  OSC = INTIO7
CONFIG  FCMEN = OFF
CONFIG  IESO = OFF

LIST p=18F4620
#include <p18F4620.inc>

        ORG 0

        movlw   0x00
        movwf   TRISA
suite
        movlw   0xFF
        movwf   LATA

        <instruction1>
        <instruction2>
        nop
        nop

        movlw 0x00
        movwf   LATA
        goto    suite
        END
```

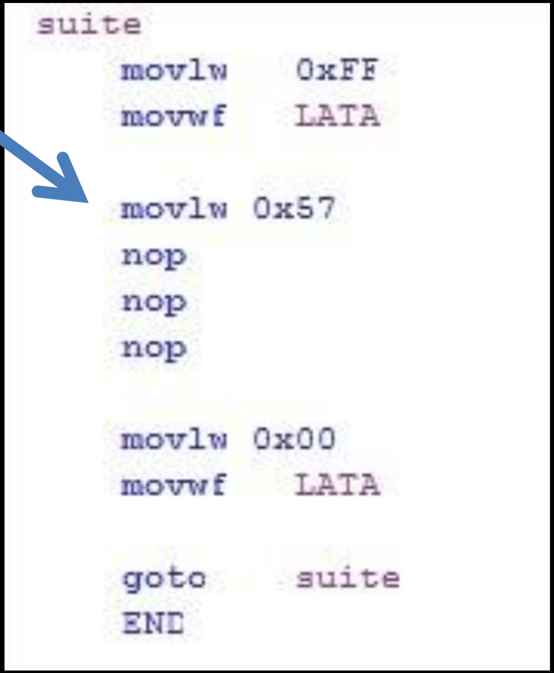
Could we create a (sort of) ‘disassembler’ over electricity?

(Experiment #5)

Could we create a (sort of) ‘disassembler’ over electricity? (Experiment #5)

Trying to find an instruction

- On PC 1, We sent to microcontroller the program with movlw 0x57 for example

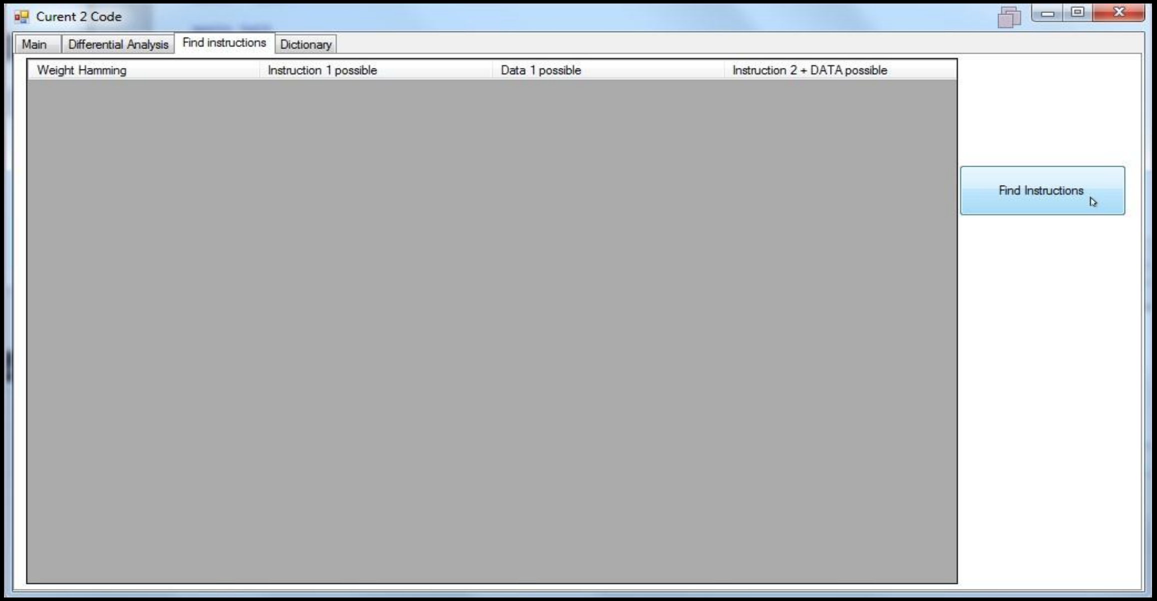


```
suite
    movlw   0xFF
    movwf   LATA

    movlw 0x57
    nop
    nop
    nop

    movlw 0x00
    movwf   LATA

    goto    suite
    END
```

Could we create a (sort of) 'disassembler' over electricity? (Experiment #5)

Trying to find an instruction

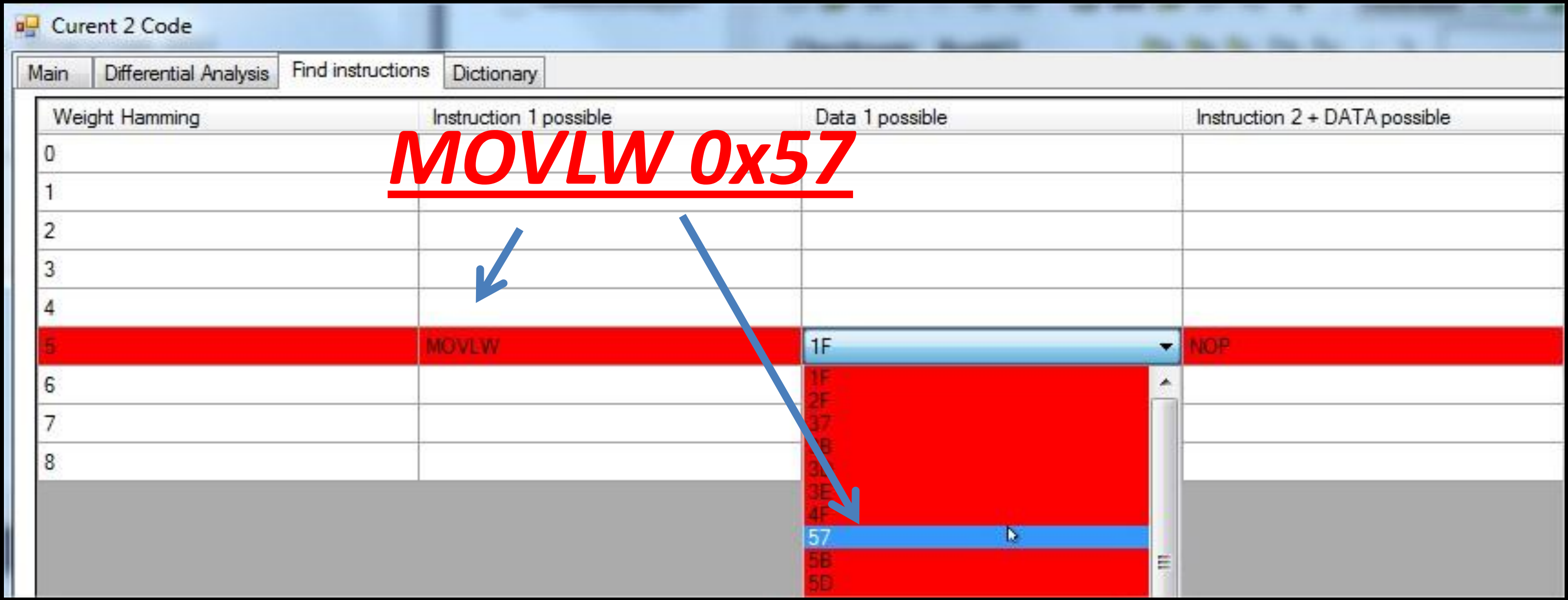
On PC2, We use the software to find instruction & data



Could we create a (sort of) ‘disassembler’ over electricity? (Experiment #5)

Trying to find an instruction

– Perfect, the instruction was found !



Curent 2 Code

Weight Hamming	Instruction 1 possible	Data 1 possible	Instruction 2 + DATA possible
0			
1			
2			
3			
4			
5	MOVLW	1F	NOP
6			
7			
8			

MOVLW 0x57

AGENDA

- Who we are
- Research context & goals
- Electronic 101 for Security Guys
- Proof of concept (soft, hard, …)
- Results & Limits
- Further researches (Prospective)
- How to limit the risk
- Conclusion

Results & Limits

- Extracting part of the code with current consumption seems to be a validated approach ☺

- But limits exist !

- Limited by hamming group / Collision of instructions
- Some issues regarding several specific set of instructions:
  - Branch and Jump instructions, I/O manipulation instruction,
  - more than 1 cycle instruction.
  - The influence on current consumption for those later would be different for sure (further investigation need to be scheduled!)
- Dictionary imply that our method could only be adapted to reverse the code of embedded system based on well know board or ready to use system (FGPA based board, Development board, Pre designed embedded system board…).

AGENDA

- Who we are
- Research context & goals
- Electronic 101 for Security Guys
- Proof of concept (soft, hard, …)
- Results & Limits
- Further researches (Prospective)
- How to limit the risk
- Conclusion

Prospective

- We based our approach on current amplitude measurement

- May be , we could add a temporal dimension to our measure to extract more information from the current consumption
  - Spot when the transistors commute
  - to be able to make a distinction of what bits is set to 1 (To be tested soon!)

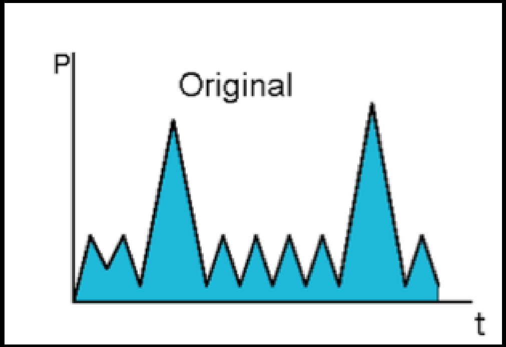
- We may also measure the electromagnetism waves create by the µC when code is executed

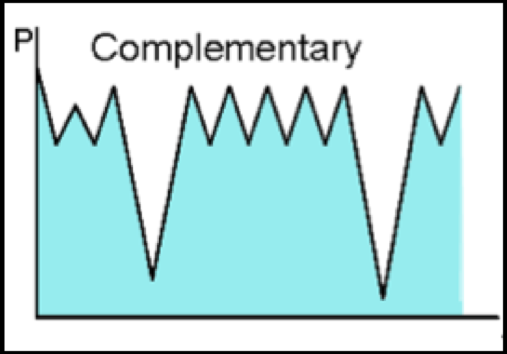
AGENDA

- Who we are
- Research context & goals
- Electronic 101 for Security Guys
- Proof of concept (soft, hard, …)
- Results & Limits
- Further researches (Prospective)
- How to limit the risk
- Conclusion

How to limit the risk

- Create a complementary current consumption (via soft or hardware) to hide the true power consumption





(source : http://scholar.lib.vt.edu/theses/available/etd-04302007-134556/unrestricted/Thesis.pdf)

- The µC manufacturers must be careful when designing the microcontroller instructions encoding table

AGENDA

- Who we are
- Research context & goals
- Electronic 101 for Security Guys
- Proof of concept (soft, hard, …)
- Results & Limits
- Further researches (Prospective)
- How to limit the risk
- Conclusion

Conclusion

- #1: Does the code really impacts the power consumption? -> YES
- #2: Do different instructions & Data could be retrieved via power analysis? -> YES
- #3: Could we create a (sort of) 'disassembler' over electricity? -> YES but with limits…

- A Hardware IDA plugins …Blackhat USA 2013 ? ☺ (#teasing)
  - Don't hesitate to donate… ;-p

Conclusion

- Cheap approach
  - 4500$ → oscilloscope
  - 10$ → Programmer / Debugger
  - 2$ → Embedded system
  - 1$ → Resistor

- Our code is open source … Download it ! Use it ! Improve it (and send us an update ;-p)

Q/A?

- To contact us :
  - *research@opale-security.com*