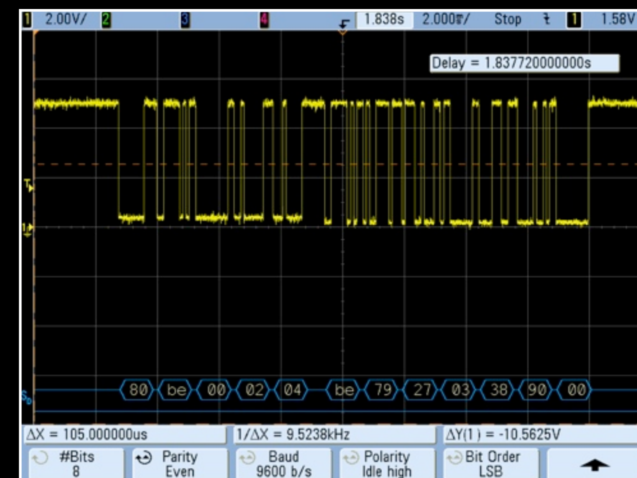
"Smart" Parking Meter Implementations, Globalism, and You

Black Hat USA 2009



Joe Grand aka Kingpin, Grand Idea Studio, Inc.
Jacob Appelbaum, Noisebridge
Chris Tarnovsky, Flylogic Engineering

Joe Grand

- Electrical engineer
- Hardware hacker
- Grand Idea Studio: Product development & licensing, consumer devices and electronics modules for hobbyists
- Member of the L0pht hacker think-tank in 1990s
- Prior security work includes numerous USB authentication token & PDA vulnerabilities/ forensics

Jacob Appelbaum

- Developer for The Tor Project
- cDc member
- Founding member of Noisebridge hacker space in San Francisco
- Notable work includes Cold Boot Attacks, Rogue CA Certificate creation, Reversing File Vault

Chris Tarnovsky

- Flylogic Engineering: Security analysis of silicon die and semiconductor devices
- Early satellite TV hacking of smartcard-based systems
- Recent work includes glitching attacks on smartcards using sewing needles

Why Parking Meters?

- We take these systems for granted and rely heavily on them, so they deserve a review
- Many U.S. cities are spending millions of dollars deploying "smart" electronic systems
  - Ex.: San Francisco, 2003, $35 million pilot program to replace 23,000 mechanical meters
  - Others include Atlanta, Boston, Chicago, Los Angeles, New York, Philadelphia, Portland, San Diego
- Is proper security due diligence really being done by parking meter vendors before implementation?

Why Parking Meters? 2

- Parking industry generates $28 billion annually
- Where there's money, there's risk for fraud and abuse
- Attacks/breaches can have serious implications
  - Fiscal
  - Legal
  - Social

Our Goals

- Understand the current state of (un)fare collection infrastructure
- Demonstrate attacks, explain potential weaknesses, present fixes
- Educate attendees on the hardware hacking process
- Case study: San Francisco Municipal Transportation Agency (MTA)

Fare Collection Infrastructure

- Parking meters
  - Single space
  - Multiple space
- Audit log retrieval
- Coin/payment retrieval
- Maintenance/repair
- Intentional role separation/distribution of trust

Parking Meter Technology

- Pure mechanical replaced with hybrid electromechanical in early 1990s
  - Mechanical coin slot
  - Minimal electronics used for timekeeping and administrator access (audit, debug, programming?)
- Now, we're seeing pure electronic "smart" systems
  - Microprocessor, memory, user interface
  - Has potential for problems like any other hardware-based embedded system

Parking Meter Technology 2

- User Interfaces
  - Coin
  - Smartcard
  - Credit card
- Administrator Interfaces
  - Coin
  - Smartcard
  - Infrared
  - Wireless (RF, GPRS)
  - Other (Serial via key, etc.)

Austin, TX


Meter Number: 1031S3
Notice
Parking in this space longer than the time limit posted on the meter is a parking violation.
Meter Holidays
New Year's Day
Independence Day
Labor Day
Thanksgiving Day
Christmas Day
Report malfunctions to 463-3476
Note meter Number
INSERT COIN OR ParkCard TO DISPLAY TIME PURCHASED

Chicago, IL

Vancouver, BC, Canada

Jerusalem, Israel

Prior Problems and/or Failures

- New York City reset via infrared (universal remote control), 2001, `http://tinyurl.com/mae3g8`
- San Diego stored value card by H1kari, 2004, `www.uninformed.org/?v=1&a=6&t=txt`
- Chicago multi-space failures, June 2009
  - Firmware bug or intentional social disobedience?
  - `http://tinyurl.com/nt7gl9`
  - `http://theexpiredmeter.com/?p=3081`

General Process

- Attack postulation
- Information gathering
- Hardware analysis
- Firmware reverse engineering
- Smartcard analysis

Attack Postulations

- Covert channels/message passing via LCD
- Meter-to-meter virus propagation via RF
- Denial-of-service
  - Set meter to "Out of Order"
  - Destruction of smartcard or coin processing circuitry/fuses via ESD
  - Cause a legitimate user to be added to fraud blocklist (if used)

Attack Postulations 2

- Immediate deduction of credit
  - Ex.: Cause a targeted law-abiding citizen to receive a ticket
- Audit log retrieval/modification
- Changing time/date
  - Ex.: Every day is Sunday, Sunday, Sunday!
- Unlimited payment via smartcard

Information Gathering

- Social engineering
- Crawling the Internet for specific information
  - Product specifications, design documents, etc.
  - What is the core business competency?
  - Do they have technical troubles?
- Dumpster diving
- Acquire target hardware
  - Purchase, borrow, or ask the vendor

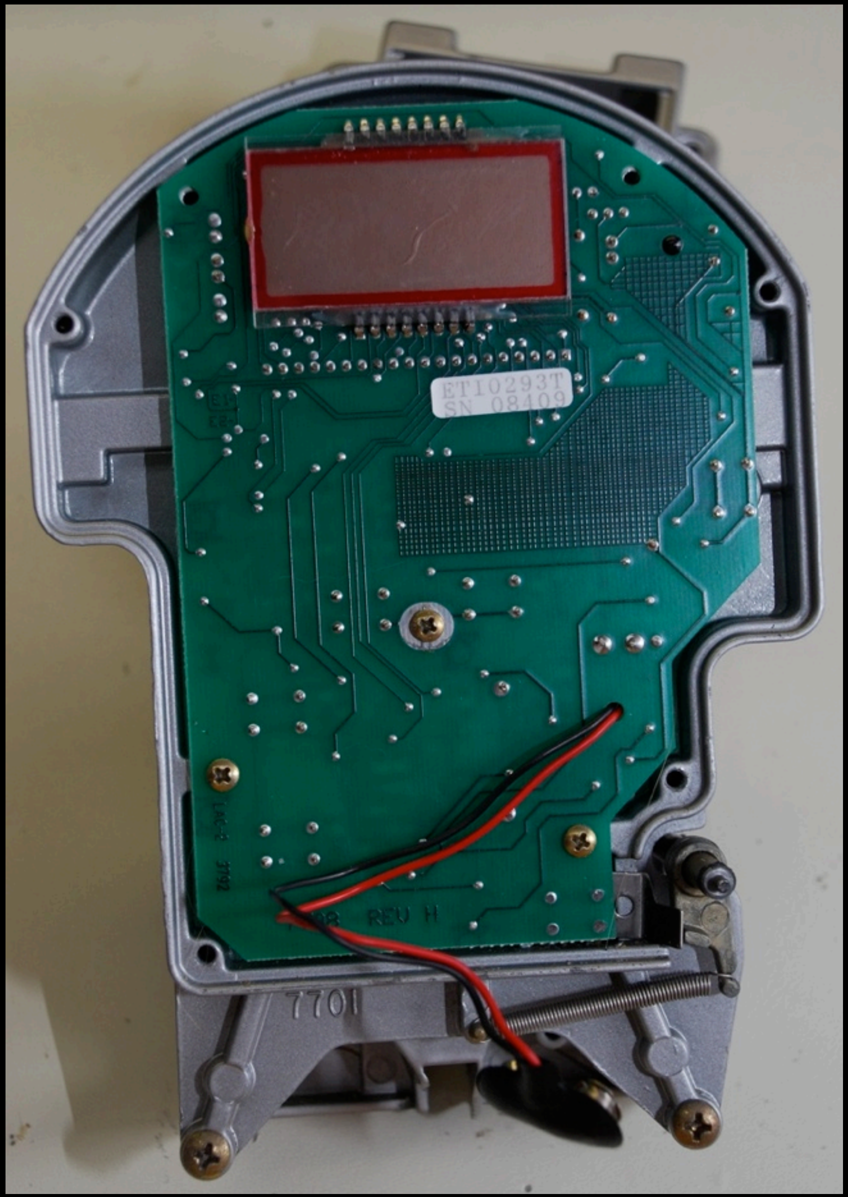
Hardware Analysis

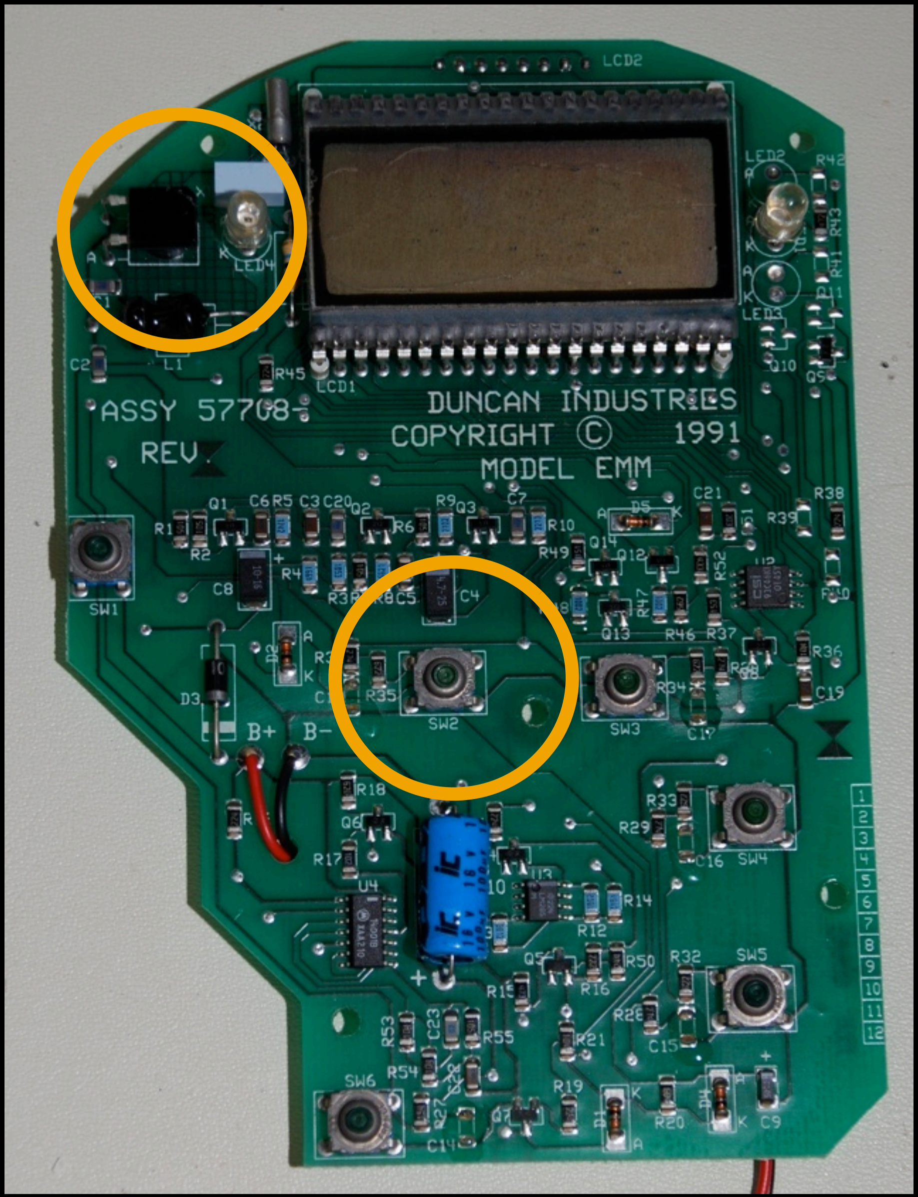
- Meter hardware and electronics disassembly
- Component and subsystem identification
- Gives us clues about design techniques, potential attacks, and meter functionality
- Typically there are similarities between older and newer designs
  - Even between competing products
- Explored a selection of single space meters
  - All purchased on eBay, prices range from $0.99 to $500
    - Duncan EMM 7700
    - POM APM
    - MacKay Guardian

Meter Disassembly: Duncan EMM 7700

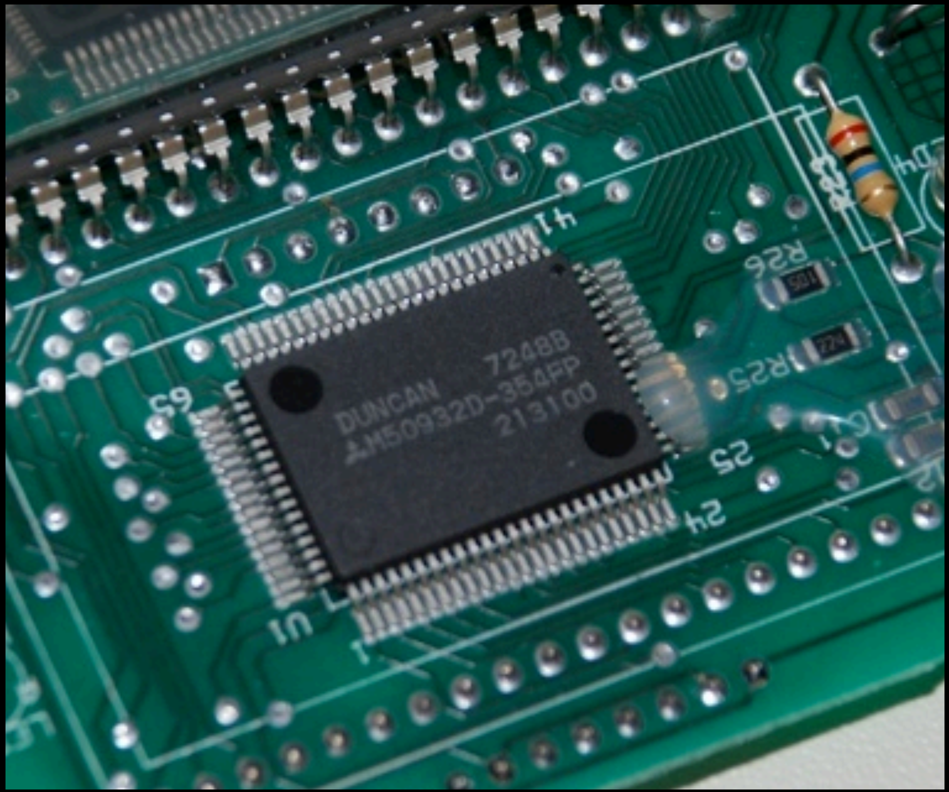
Meter Disassembly: Duncan EMM 7700 2

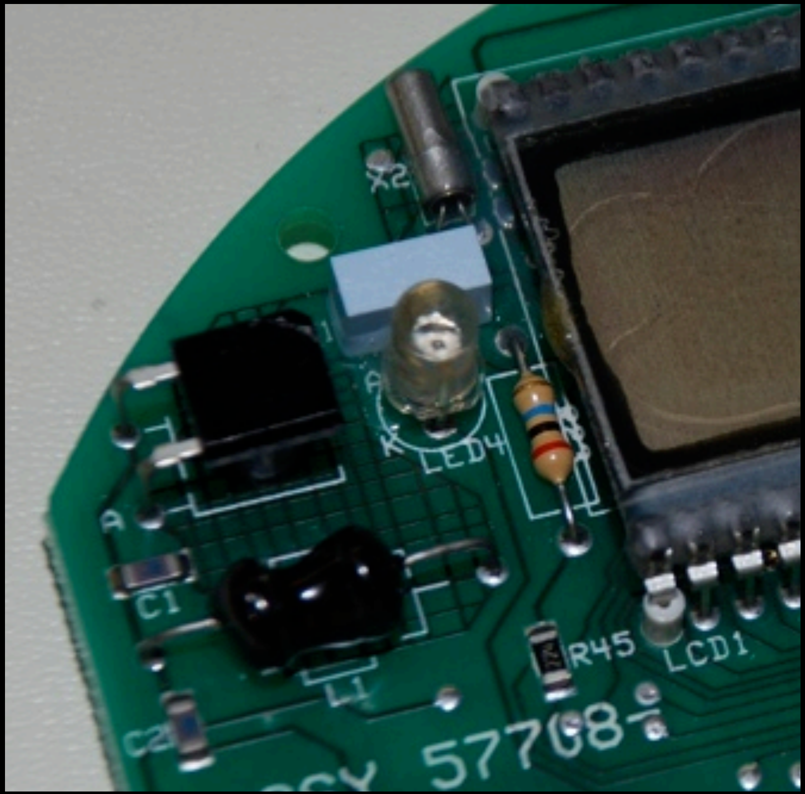
Meter Disassembly: Duncan EMM 7700 3



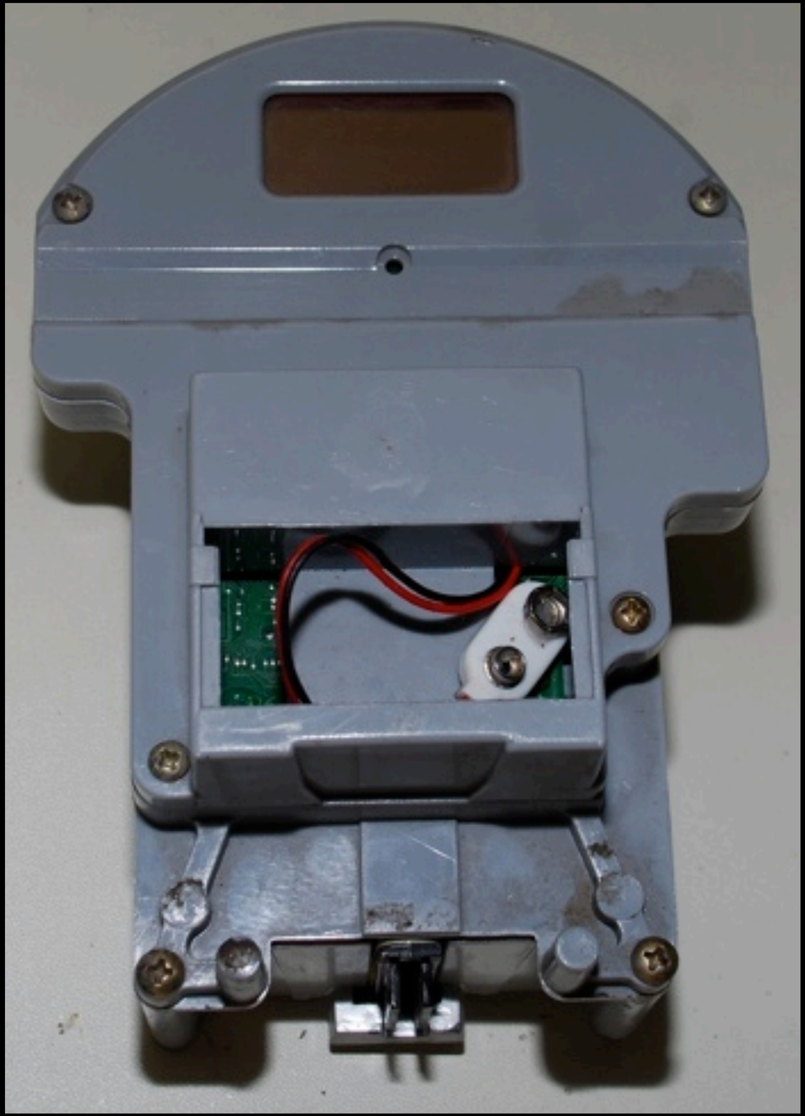


Meter Disassembly: Duncan EMM 7700 4

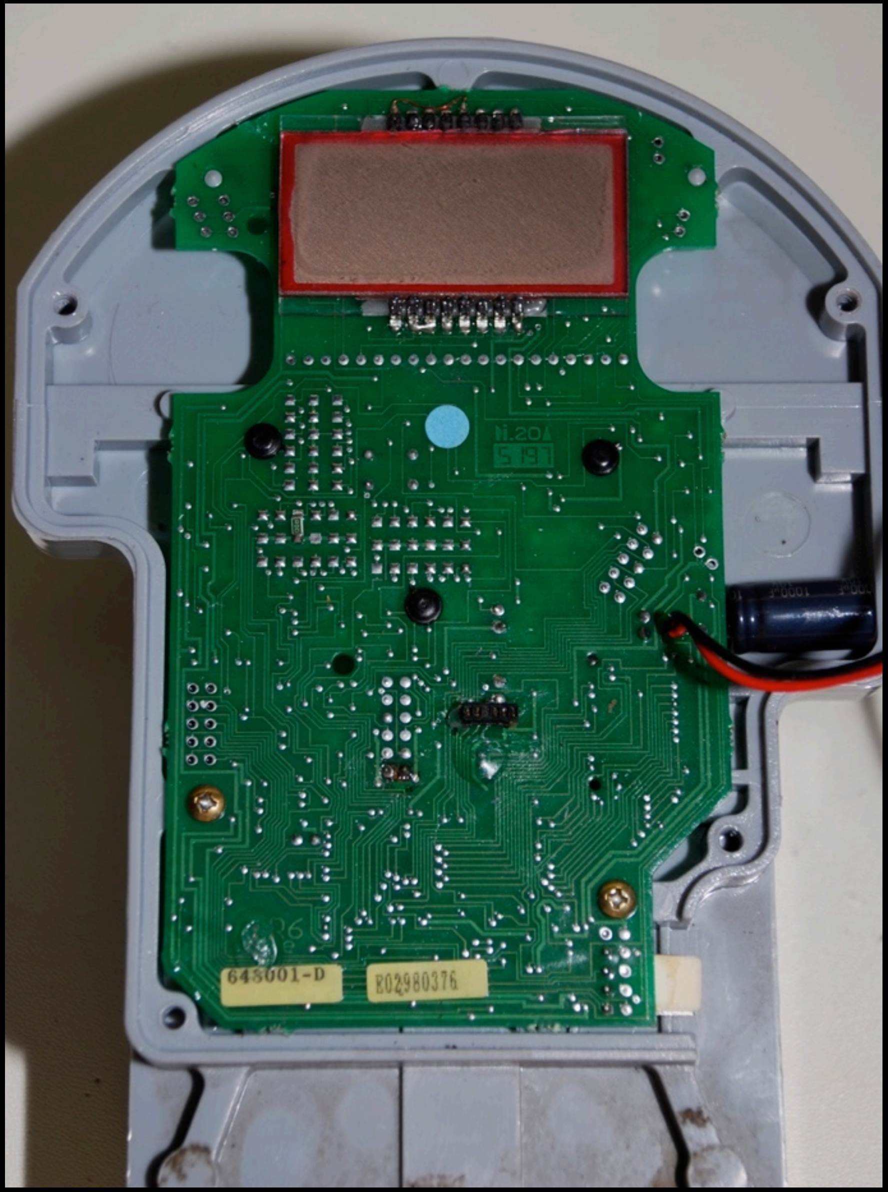




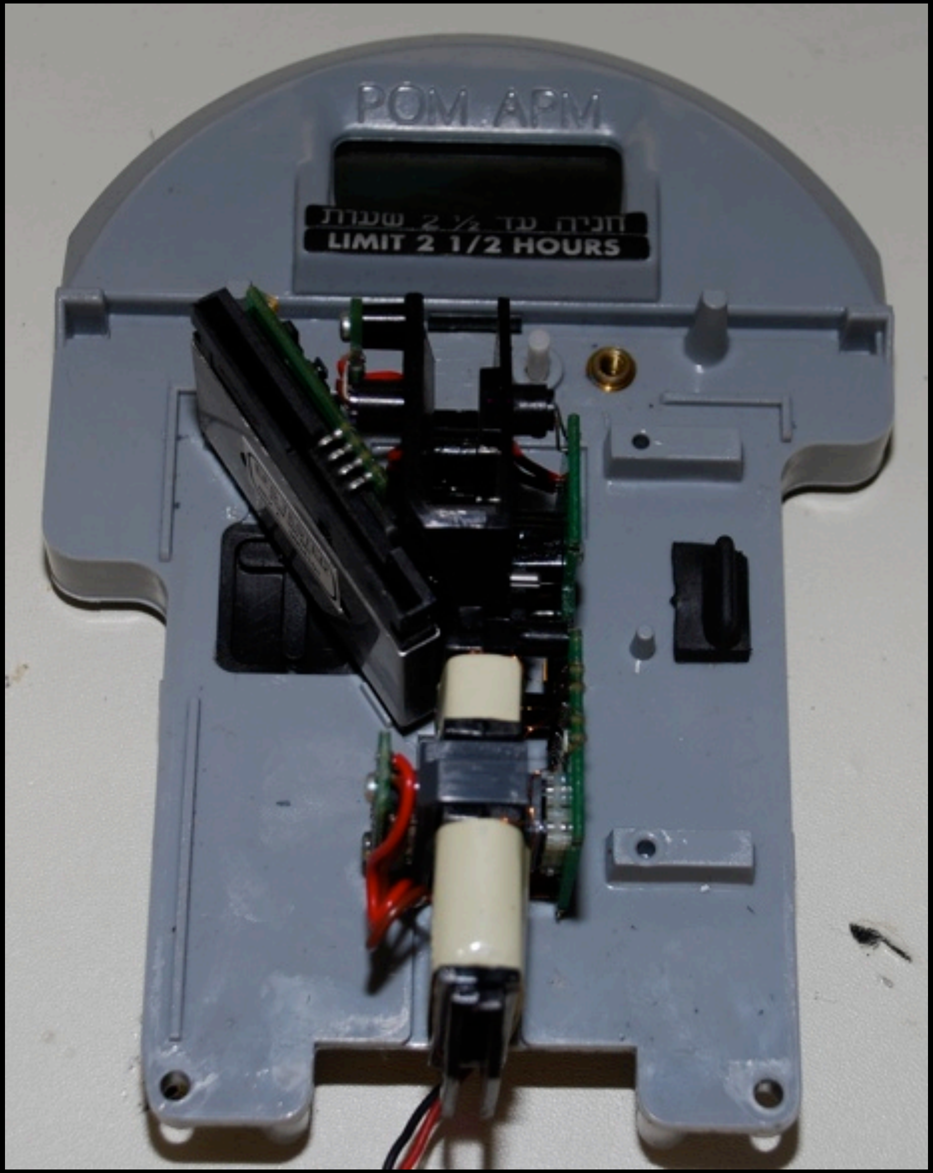
Meter Disassembly: POM APM

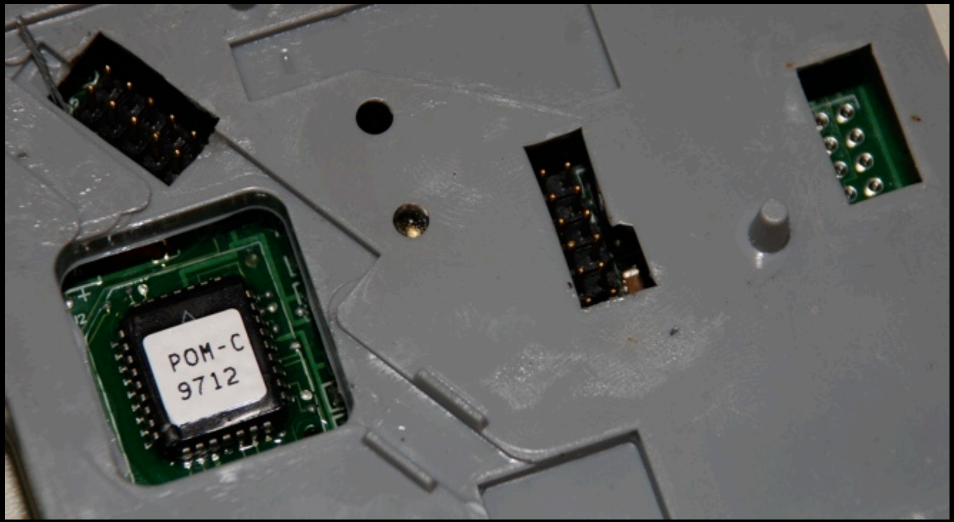


Meter Disassembly: POM APM 2

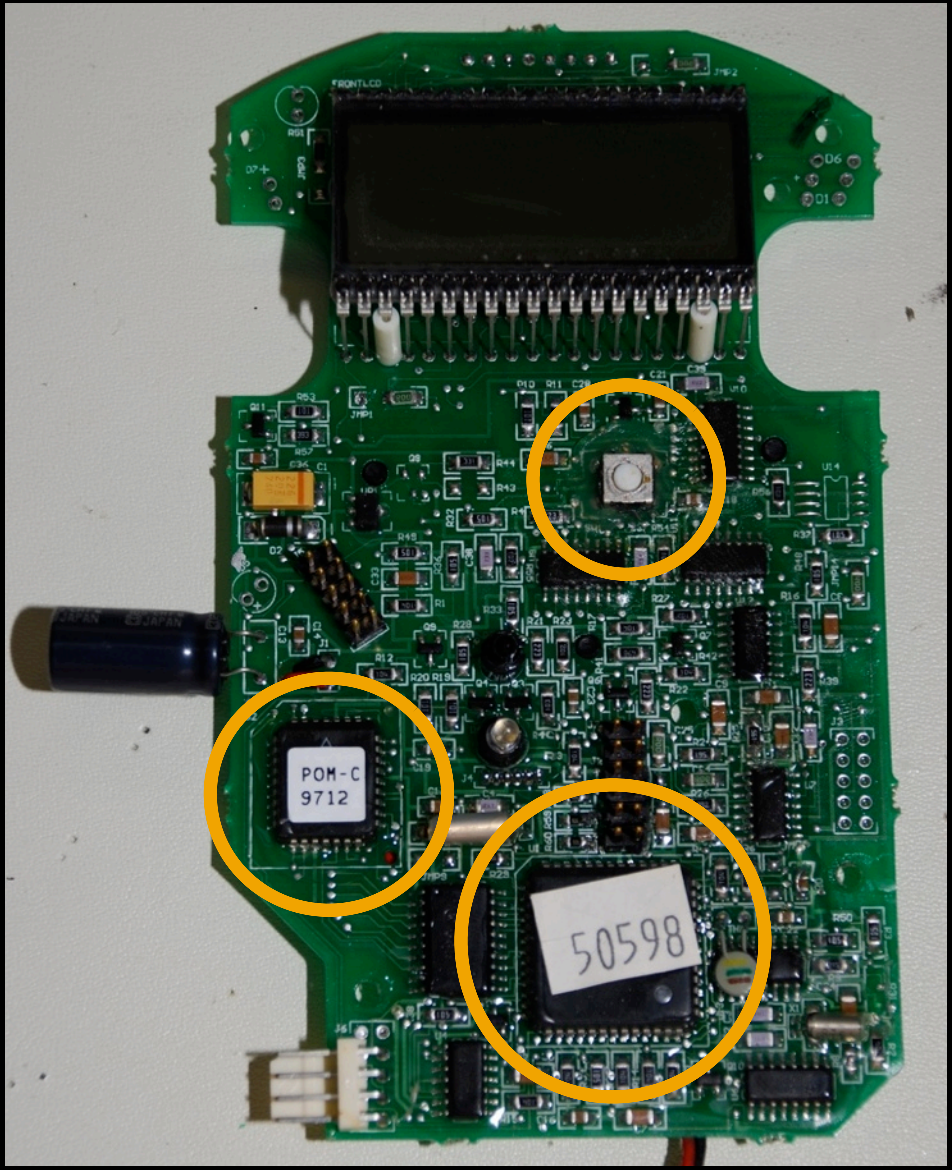


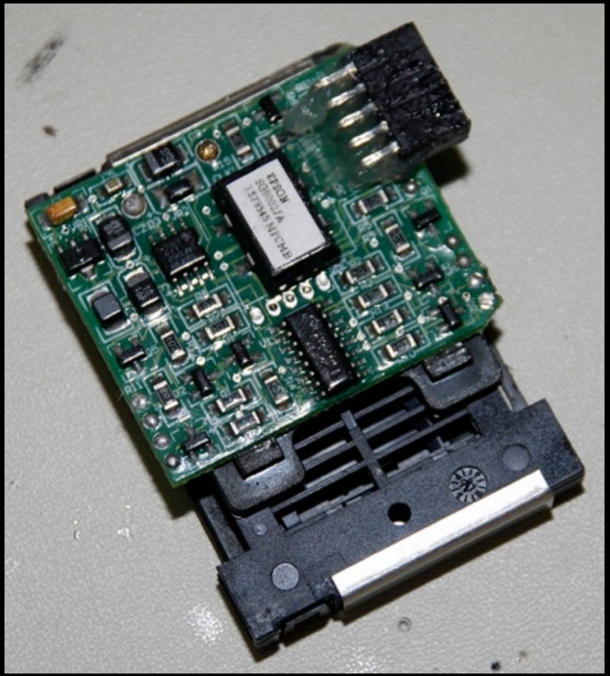
Meter Disassembly: POM APM 3

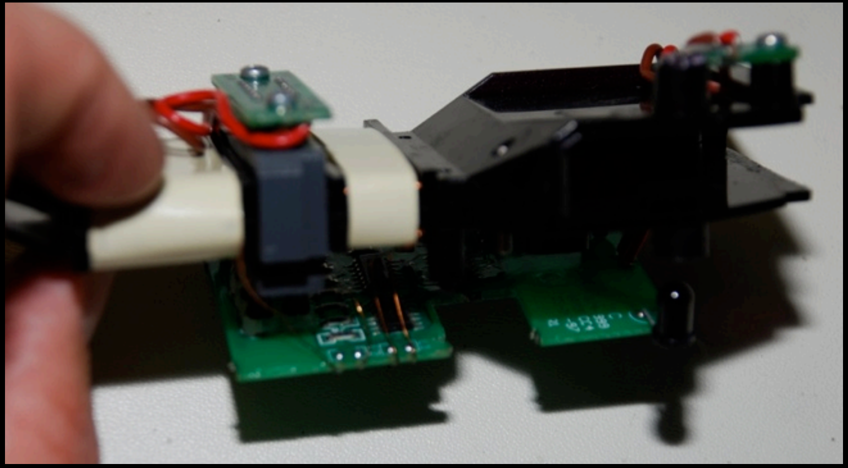




Meter Disassembly: POM APM 4

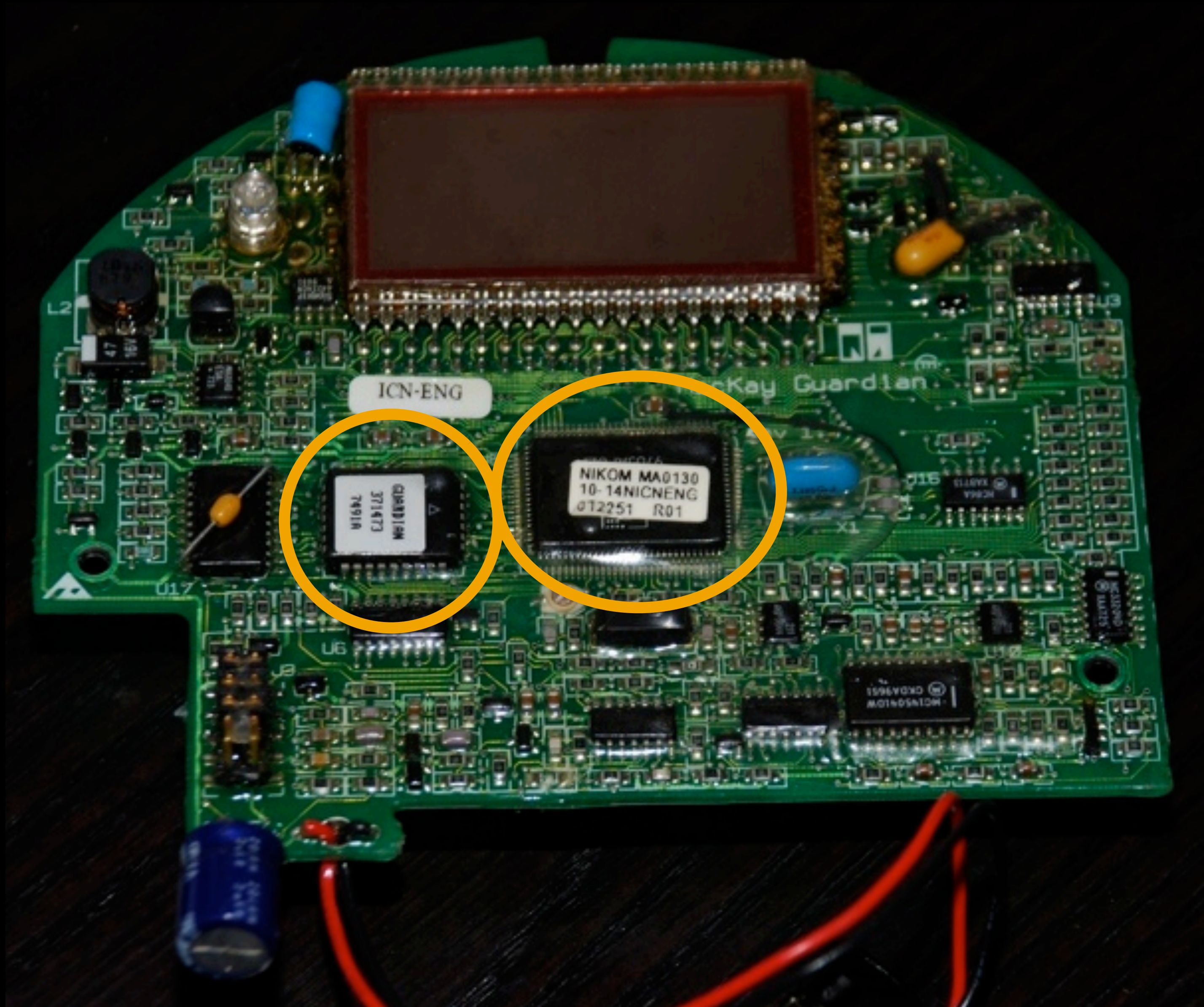




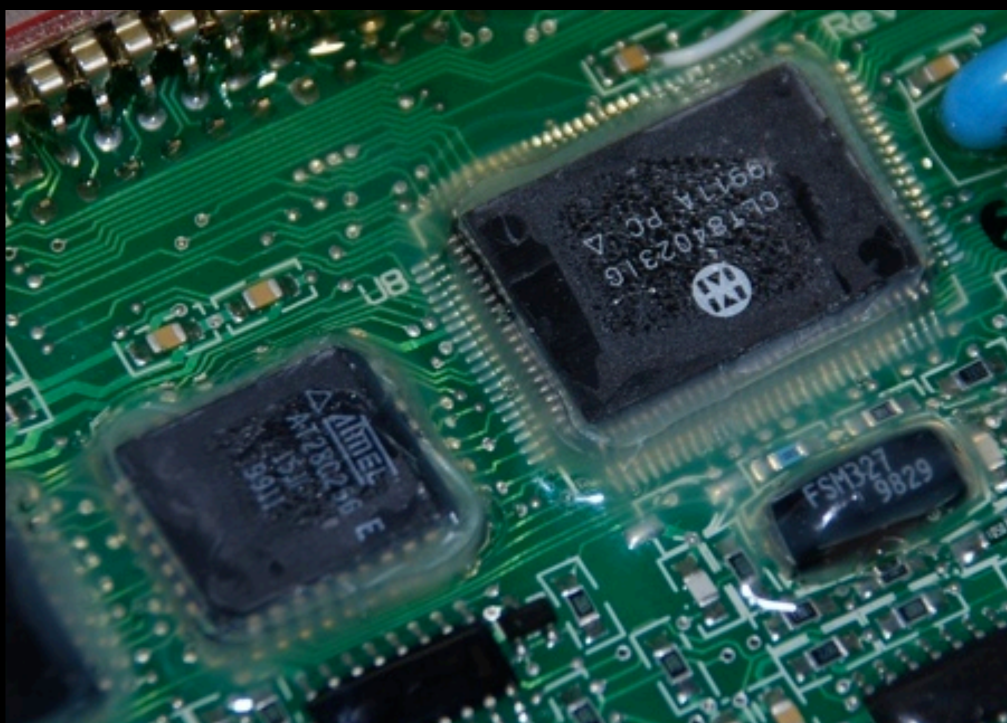


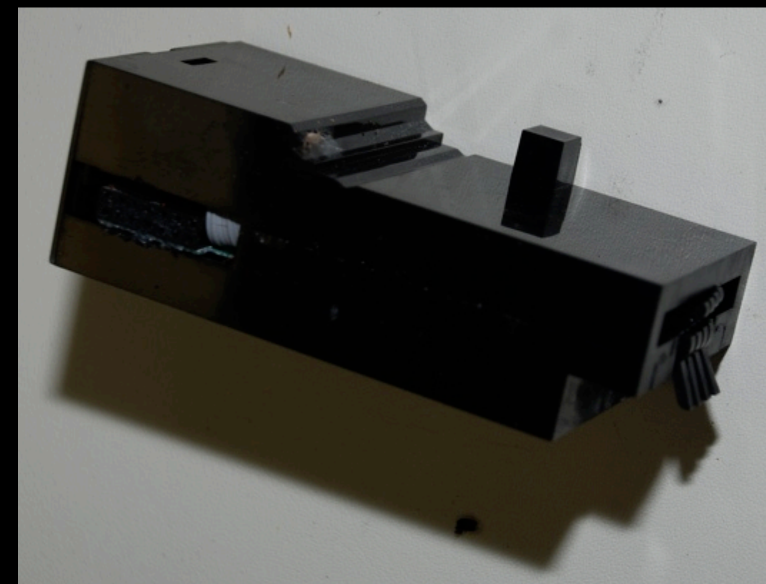
Meter Disassembly: MacKay Guardian

Meter Disassembly: MacKay Guardian 2

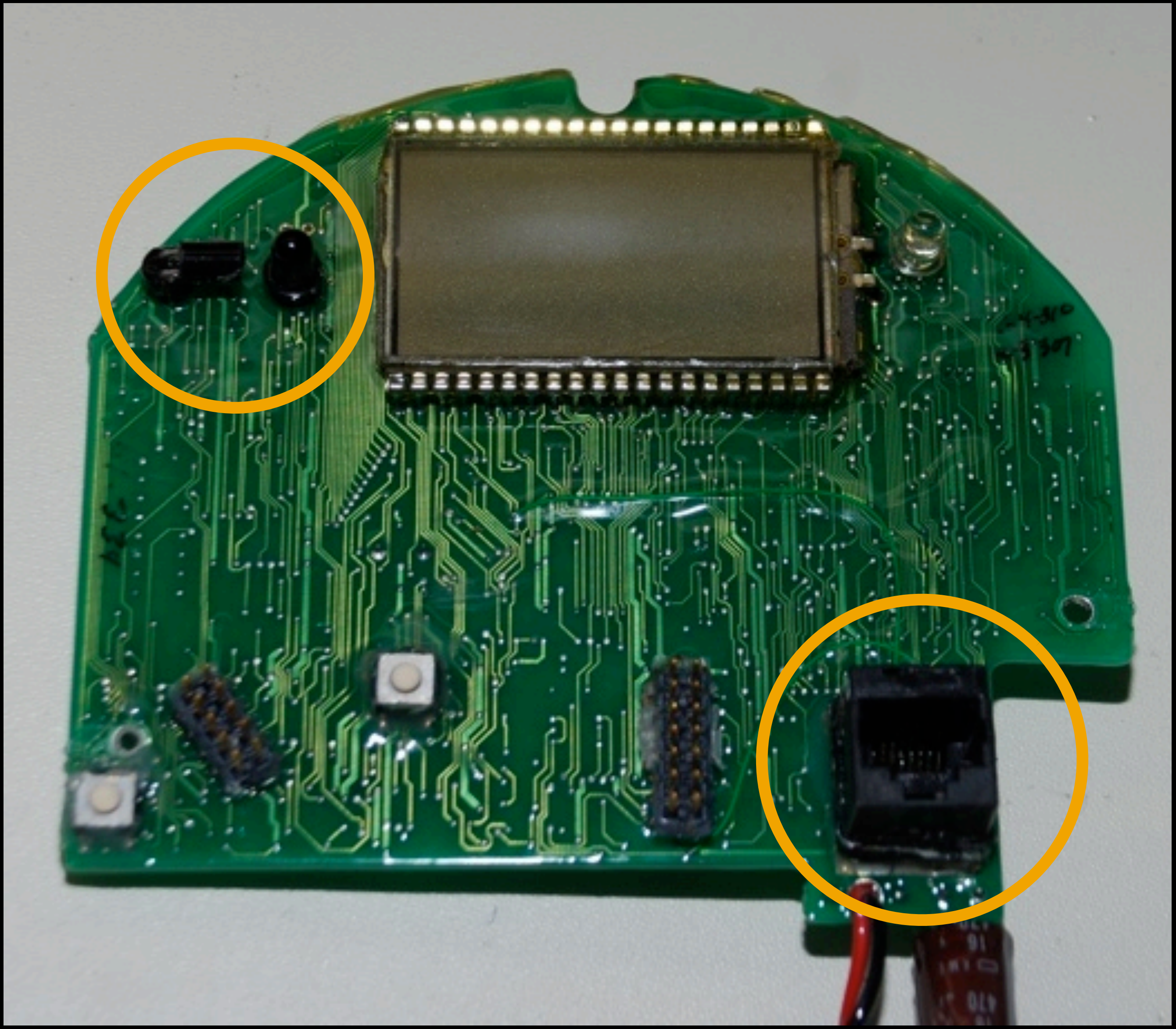


Meter Disassembly: MacKay Guardian 3

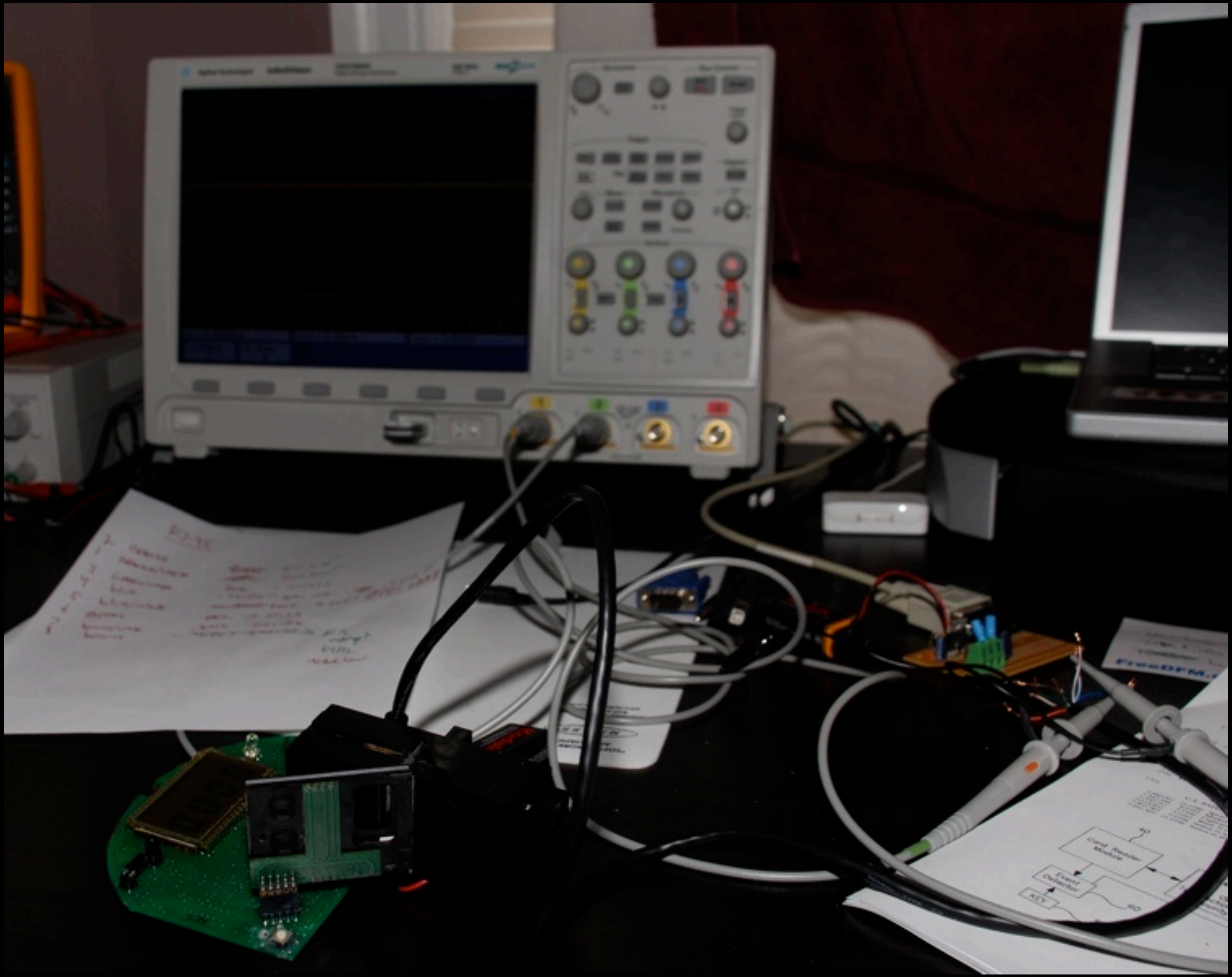




Meter Disassembly: MacKay Guardian 4



Meter Disassembly: MacKay Guardian 5



Firmware Analysis

- Extract program code/data from on-board memory devices (Flash or ROM)
- Quick run through w/ *strings* and hex editor to pick most interesting area to begin with
- Disassembly and reverse engineering
- Gives clues to possible entry/access points to administrative menus or ideas of further attacks

Smartcard Analysis

- Communications monitoring
- Protocol decoding and emulation
- Silicon die analysis (if resources allow)

Case Study: San Francisco MTA

Case Study: San Francisco MTA

Case Study: San Francisco MTA Introduction

- Part of a $35 million pilot program to replace 23,000 mechanical meters in 2003
- City is considering adding more meters to fill every available parking spot
  - 320,000 of them!
  - `http://tinyurl.com/nhpgzm`
- Infrastructure currently comprised of MacKay Guardian XLE meters

Case Study: San Francisco MTA Introduction 2

- Stored value smart card
  - $20 or $50 quantities
  - Can purchase online with credit card or in cash from selected locations
- Easy to replay transaction w/ modified data to obtain unlimited parking
  - Determined solely by looking at oscilloscope captures of smartcard transactions
  - Succeeded in three days

Case Study: San Francisco MTA Process

- Information Gathering
- Smartcard & Silicon Die Analysis
  - Treated as a black box attack, no meter required

Case Study: San Francisco MTA Caveats

- Released code is solely for educational purposes
  - Commands/data will be changed to prevent fraud against SFMTA
  - The goal is to show how attack was successful without putting any company at risk
  - Get it from `www.grandideastudio.com/portfolio/smart-parking-meters/`

Case Study: San Francisco MTA Information Gathering

- A chance encounter w/ Department of Parking & Transportation technician on the streets of SF
  - Ask smart, but technically awkward questions to elicit corrections
- Crawling the Internet for specific information
  - Product specifications, design documents, etc.
  - What is the core business competency?
  - Do they have technical troubles?

Case Study: San Francisco MTA They Do Have Technical Troubles!

```
# From: xxx <xxx at jjmackay dot ca>
# Date: Wed, 14 Mar 2001 10:27:29 -0400

I am learning how to use CVS and as part of this process I set up a test
repository to 'play' with.

D:\src\working\epurse\cvstest>cygcheck -s -v -r -h

Cygnus Win95/NT Configuration Diagnostics
Current System Time: Wed Mar 14 09:39:50 2001

Win9X Ver 4.10 build 67766446  A

Path:   /cygdrive/c/NOVELL/CLIENT32
        /cygdrive/c/WINDOWS
        /cygdrive/c/WINDOWS/COMMAND
        /usr/bin
        /cygdrive/c/JJMACKAY/MET_TALK
        /cygdrive/c/JJMACKAY/UTILITY

GEMPLUS_LIB_PATH = `C:\WINDOWS\GEMPLUS'

Found: C:\cygwin\bin\gcc.exe
Found: C:\cygwin\bin\gdb.exe

xxx, Sr. Software Designer
```

http://www.cygwin.com/ml/cygwin/2001-03/msg00842.html

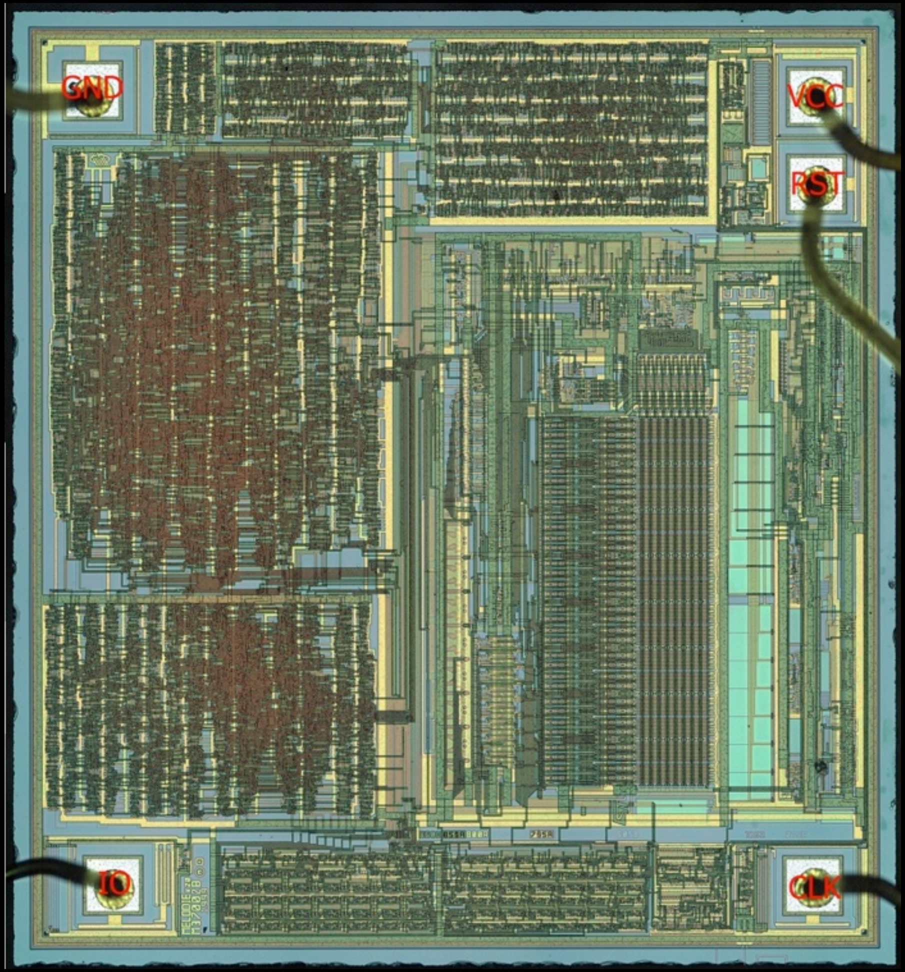
Case Study: San Francisco MTA Silicon Die Analysis

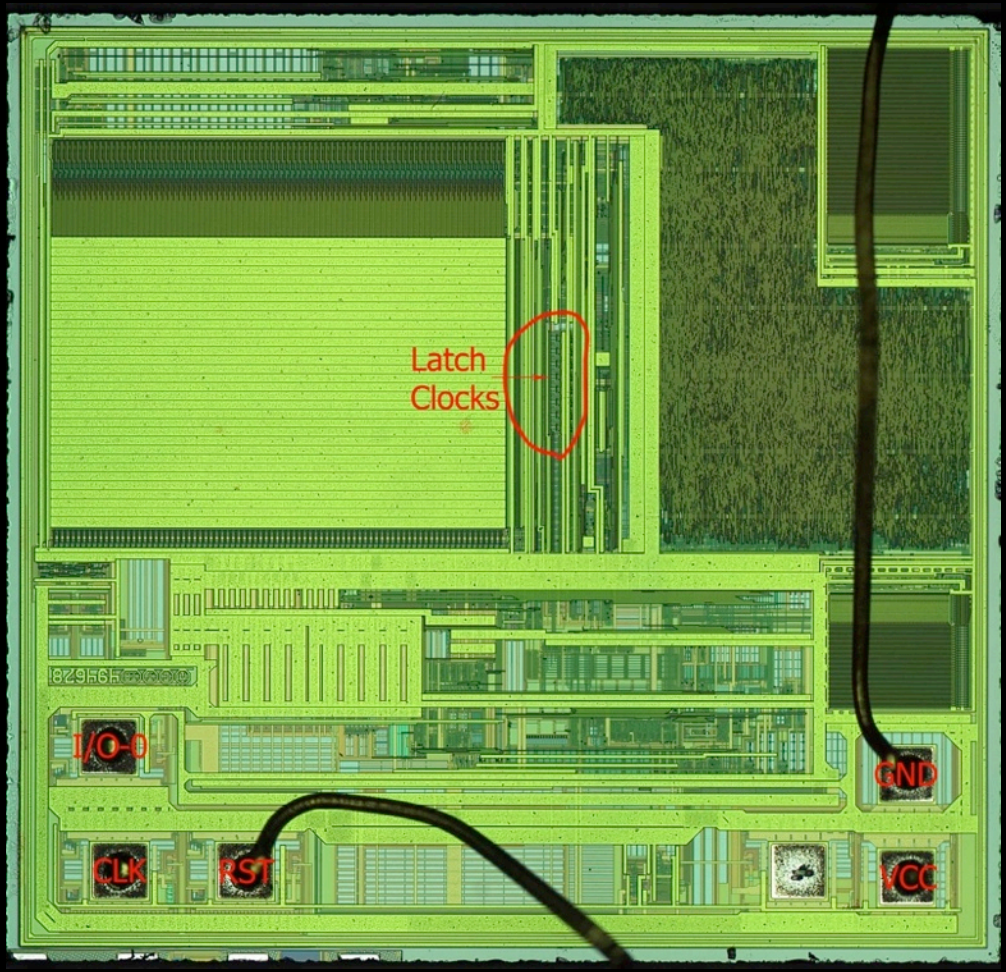
- Purchased and decapsulated multiple cards to look for clues of manufacturer and functionality
- Decapsulation process for smartcards
  1. Remove plastic surrounding the die (usually w/ acetone)
  2. Throw die into small Pyrex of heated Fuming Nitric Acid (HNO_3)
  3. Rinse in acetone
  4. Glue die into a ceramic DIP package (for probing)
  5. If part is for analysis, prevent scratching!

Case Study: San Francisco MTA Silicon Die Analysis 2

- Visually identified that two different smartcard types exist
  - Gemplus GemClub-Memo (ASIC)
  - 8051 microcontroller *emulating* GemClub-Memo
- Dependent on card serial number
  - Older cards are ASIC, newer cards are MCU
- Microcontroller has potential for hidden/ undocumented commands
  - One could retrieve the code from the card and reverse engineer (we didn't)

Case Study: San Francisco MTA Silicon Die Analysis 3





Case Study: San Francisco MTA ISO7816 Overview

- International specification for smartcards
- Multiple sections
  - ISO7816-1: Physical Characteristics
  - ISO7816-2: Dimensions and Locations of Contacts
  - ISO7816-3: Electronic Signals and Transmissions Protocols
  - ...and many more!
- `http://en.wikipedia.org/wiki/ISO/IEC_7816`

Case Study: San Francisco MTA ISO7816 Overview 2

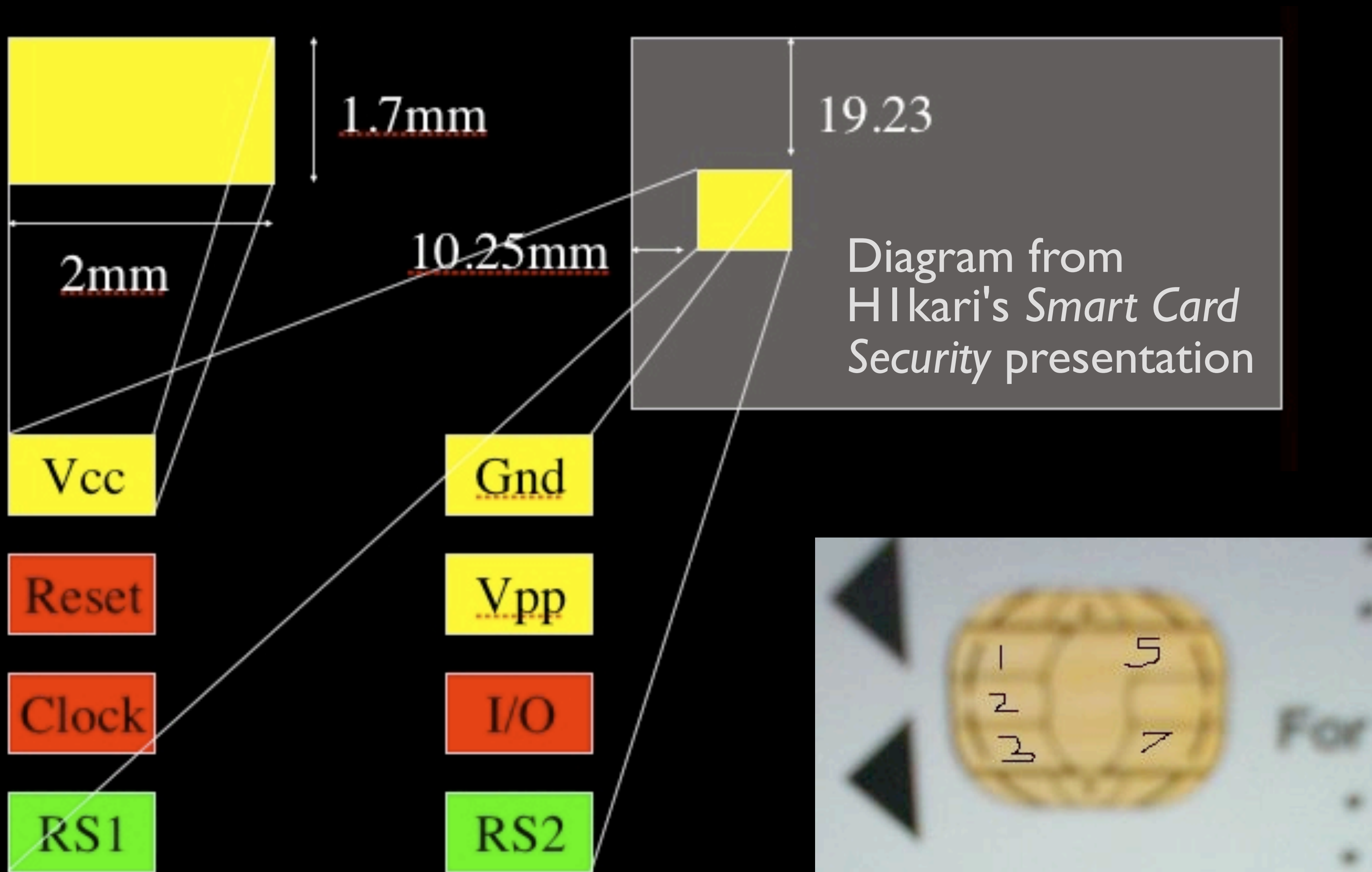


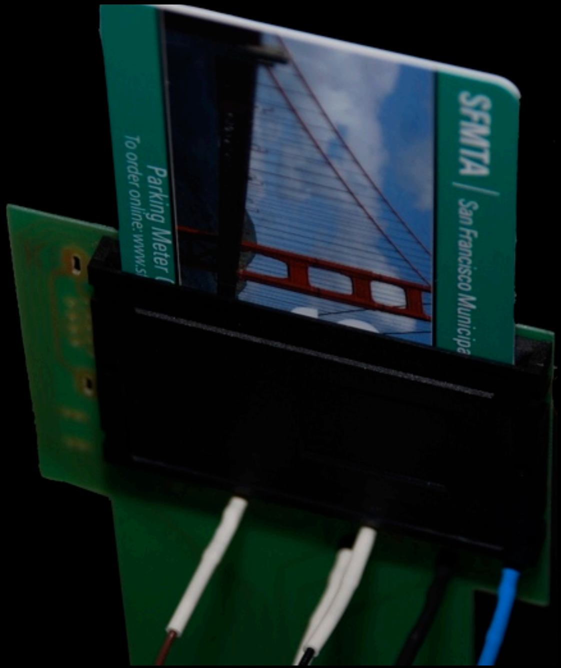
Diagram from H1kari's *Smart Card Security* presentation

Case Study: San Francisco MTA ISO7816 Overview 3

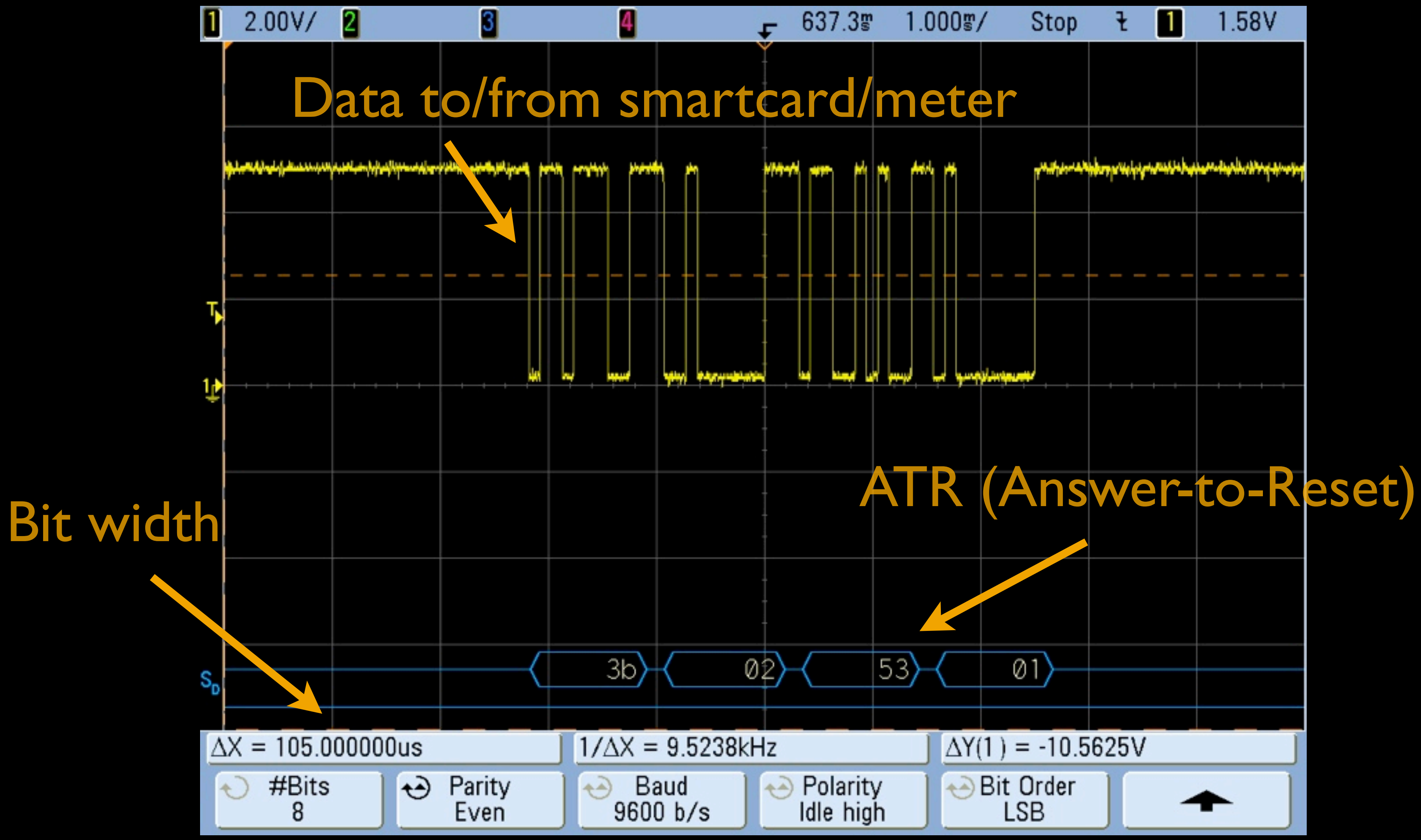
- Transmission Protocols
  - Asynchronous
    - No external clock needed ala RS232
    - T=0: Half-duplex character transmission
    - T=1: Half-duplex block transmission
    - Operates at a set baud rate (ex.: 9600bps)
    - Uses APDU (Application Protocol Data Unit) protocol
    - Ex.: Processor-based, Java, PKI, SIM cards
  - Synchronous
    - Data shifted in/out in relation to CLK ala I2C/SPI
    - Ex.: "Dumb" stored value/memory cards

Case Study: San Francisco MTA Communications Monitoring

- Used "shim" between smartcard and meter
  - Unpopulated Season 2 Interface
- Monitored I/O transaction w/ digital oscilloscope
- Asynchronous serial data @ 9600, 8E1 captured and decoded
  - Correct baud rate determined by measuring bit width on scope



Case Study: San Francisco MTA Communications Monitoring 2



Case Study: San Francisco MTA Protocol Decoding

- Captured multiple transactions to gather clues on operation
  - Different valued cards
  - Different serial numbers
- Based on what values changed per transaction & per card, could narrow down what data meant what
- Decoded transaction functionality by hand, no computer needed!

Case Study: San Francisco MTA Initialization

Meter	*Card*
	[4 byte responses unless noted]
Reset	
	ATR
Read Address 0	
	Manufacturer ID
Read Address 1	
	Serial #
Read Address 2	
	Constant
Read Address 3	
	Unknown (8) [Used for meter to calculate CSC1 password]
Read Address 4	

Case Study: San Francisco MTA
Initialization 2

Meter		*Card*
		[4 byte responses unless noted]
Read CSC1 Ratification Counter	→ ←	0
CSC1 Password [Password calculated by meter and sent to card for authentication]	→ ←	Password OK (2)
Read Address 14	→ ←	0
Read CTC1 Card Transaction Counter	→ ←	CTC1 [value varies]

Case Study: San Francisco MTA Initialization 3

Meter		*Card*
Read Balance 2	→	[4 byte responses unless noted]
	←	Maximum Card Value Ex.: 0xFF FF F0 AF = $20 Ex.: 0xFF FF F1 27 = $50
Read CTC1 Card Transaction Counter	→	
	←	CTC1 [value varies]

Case Study: San Francisco MTA Deduction of Single Unit ($0.25)

Meter	*Card*
	[4 byte responses unless noted]
Update Balance 1 Current Value A1	OK (2)
Update Balance 1 Current Value A2	OK (2)

- By updating the Balance 1 Value (8 bytes), CTC1 automatically increments
- CTC1 is the only value that changes during the entire transaction!

Case Study: San Francisco MTA Computation of Card Value

- Maximum card value = (Balance 2 - 95d)
  - Ex.: $0AF (175d) - 95d = 80 units
    - 80 * 0.25 = $20
  - Ex.: $127 (295d) - 95d = 200 units
    - 200 * 0.25 = $50

Case Study: San Francisco MTA Protocol Emulation

- First attempt to replay exact transaction captured w/ scope
  - Microchip PIC16F648A
  - Written in C using MPLAB + CCS PIC-C
  - Challenge for code to be fast enough and incorporate required short delays while still be readable/useful C

Case Study: San Francisco MTA Protocol Emulation 2

card.c

card.c:5 <No selected symbol>

Code snippet

```c
#include "card.h"

void main (void)
{
    port_b_pullups(FALSE); // disable port B pull-ups

    atr();
    manufacturer();
    issuer();
    current_value();

    while(1)
    {
        issuer();
        deposit_coin();
    }
}

void atr(void)
{
    delay_ms(1);

    putc(0x3B);delay_us(170); // guard time
    putc(0x02);delay_us(170);
    putc(0x53);delay_us(170);
    putc(0x01);
}

void manufacturer(void)
{
    output_float(SIO);
    while (getc() != 0x80);
    while (getc() != 0xBE);
    while (getc() != 0x00);
    while (getc() != 0x00);
    while (getc() != 0x04);
    delay_us(500);
    putc(0xBE);delay_us(170); // guard time
    putc(0x7A);delay_us(170);
    putc(0x11);delay_us(170);
    putc(0x11);delay_us(170);
    putc(0xFF);delay_us(170);
    putc(0x90);delay_us(170);
    putc(0x00);
}
```

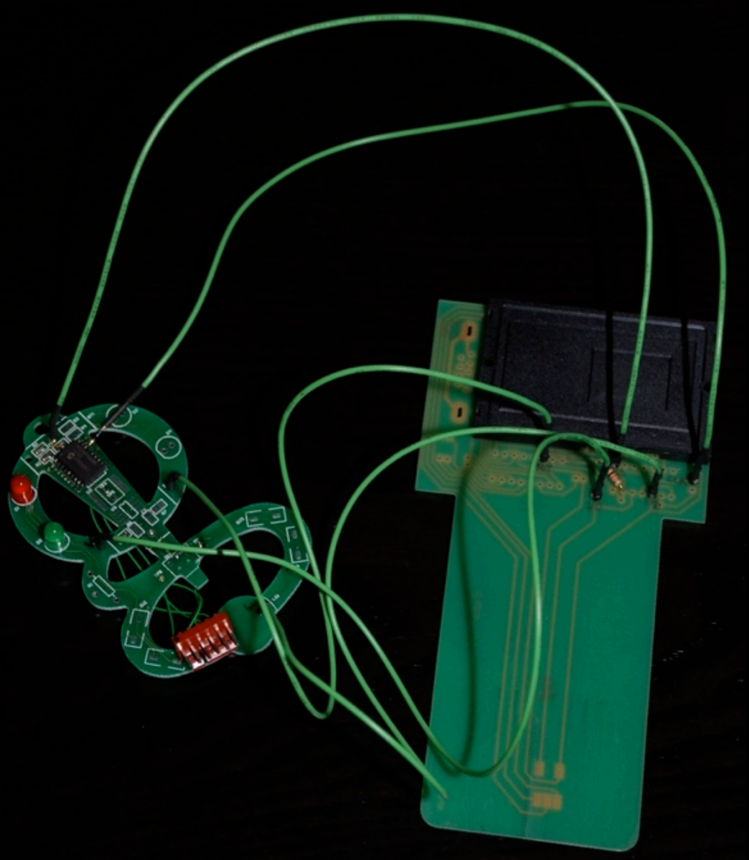
Case Study: San Francisco MTA Protocol Emulation 3

- Then, modified code to change various values until success
  - Knowing how "remaining value" is computed, what happens if we change Balance 2 to $FFF?
    - Ex.: $FFF (4095d) - 95d = 4000 units?
  - Meter believes card has the maximum possible value
  - Could also have the code never increment CTC1 so stored value never decreases

Case Study: San Francisco MTA Protocol Emulation 4

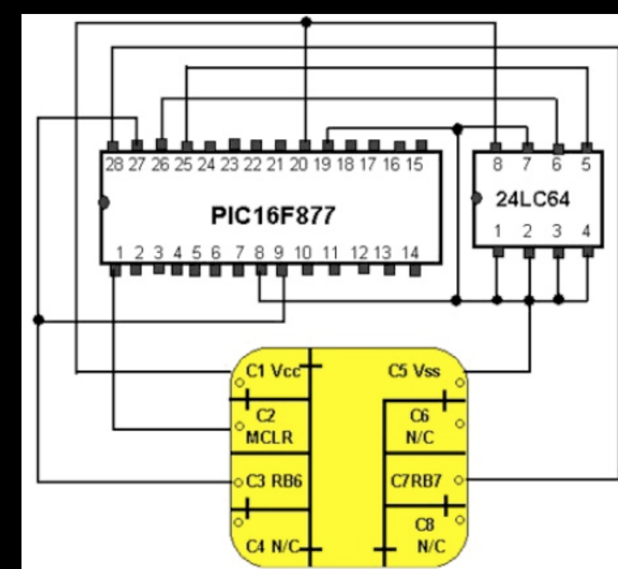
- Ported code to Silver Card (PIC16F877-based smart card)
  - PIC-based smartcards have been popular for satellite TV hackers for years, so required equipment is readily available
    - Ex.: `http://tinyurl.com/mqphcj`

Case Study: San Francisco MTA Hardware Evolution



1) Custom PCB + shim

2) MM2 card w/ external PIC



3) Silver Card PIC16F877 smartcard

Case Study: San Francisco MTA Results

Case Study: San Francisco MTA Recommended Fixes

- Daily audit log/serial number correlation/ blocklisting
  - There are serious privacy implications with this...
- Reduce number of access methods
  - Every access point is an avenue of attack
  - Ex.: MacKay Guardian XLE specification requires no fewer than *five*
- Incorporate anti-tamper mechanisms into parking meter circuitry
  - Will prevent easy access to firmware and other internals

Case Study: San Francisco MTA Recommended Fixes 2

- Abandon the use of an offline system
  - An isolated meter is no match for a dedicated attacker
- Meters could communicate with a mothership
  - Incorporate digital signatures for all transactions
  - New attacks may present themselves...
- See David Chaum's work on anonymous ecash
  - `http://en.wikipedia.org/wiki/Ecash`
  - Trust and verify: Don't contribute to counterfeiting

Final Thoughts

- Systems need to be fully tested before deployment
  - Why is hardware always inherently trusted?
- We are barely scratching the surface of what can be done against parking meters
  - Different cities have different implementations
  - Different vendors have different designs and exploitable features
- Parking meters are like real-world DRM
  - Good luck with that.
- Consider a world without parking meters
  - Ride a bicycle!

Thank You

- Jennifer Granick
  - Electronic Frontier Foundation, **`www.eff.org`**

Q & A