Injecting RDS-TMC Traffic Information Signals

Andrea Barisani
Chief Security Engineer
<andrea@inversepath.com>

Daniele Bianco
Hardware Hacker
<daniele@inversepath.com>

INVERSE PATH

http://www.inversepath.com

Introduction

DISCLAIMER:

All the scripts and/or commands and/or configurations and/or schematics provided in the presentation must be treated as examples, use the presented information at your own risk.

What's this all about ?

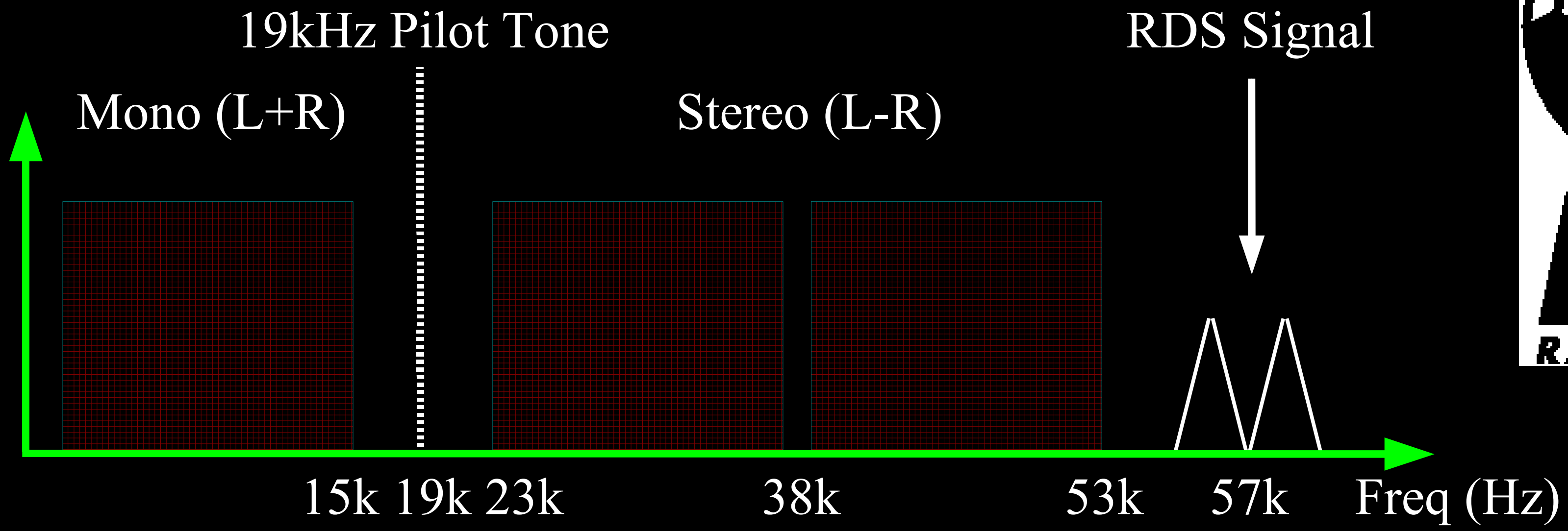
- Modern In-Car Satellite Navigation systems are capable of receiving dynamic traffic information
- One of the systems being used throughout Europe and North America is RDS-TMC (*Radio Data System – Traffic Message Channel*)
- One of the speakers bought a car featuring one of these SatNavs...he decided to play with it...just a little...
- We'll show how RDS-TMC information can be hijacked and falsified using homebrew hardware and software

Why bother ?

- First of all...hardware hacking is fun and 0wning a car is priceless ;-P
- it's so 80s
- ok seriously...Traffic Information displayed on SatNav is implicitly trusted by drivers, nasty things can be attempted
- more important: chicks will melt when you show this...

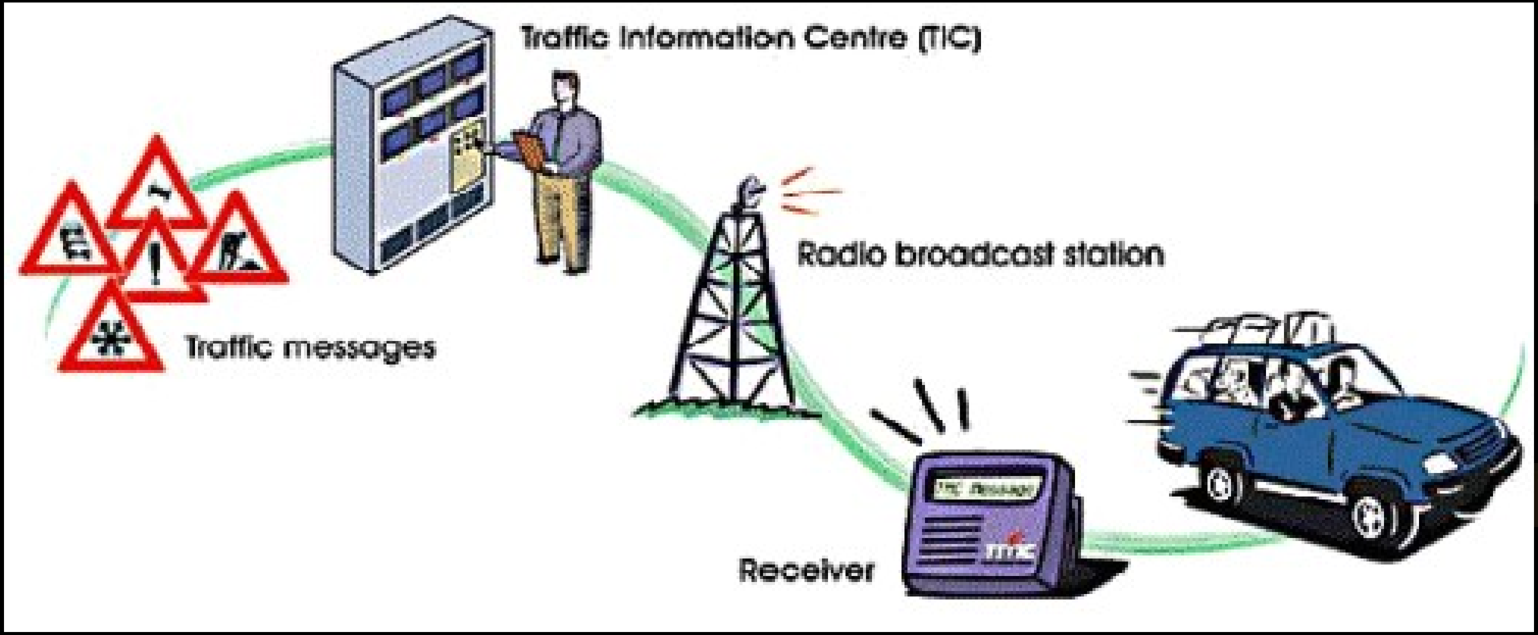
The Radio Data System

- RDS is used for transmitting data over FM (1187.5 bits/s)
- Described in European Standard EN50067 (April 1998)
- Its most prominent function is showing FM Channel Name on the radio display, also used for Alternate Frequencies, Programme Type, News override, etc.



RDS-TMC Introduction

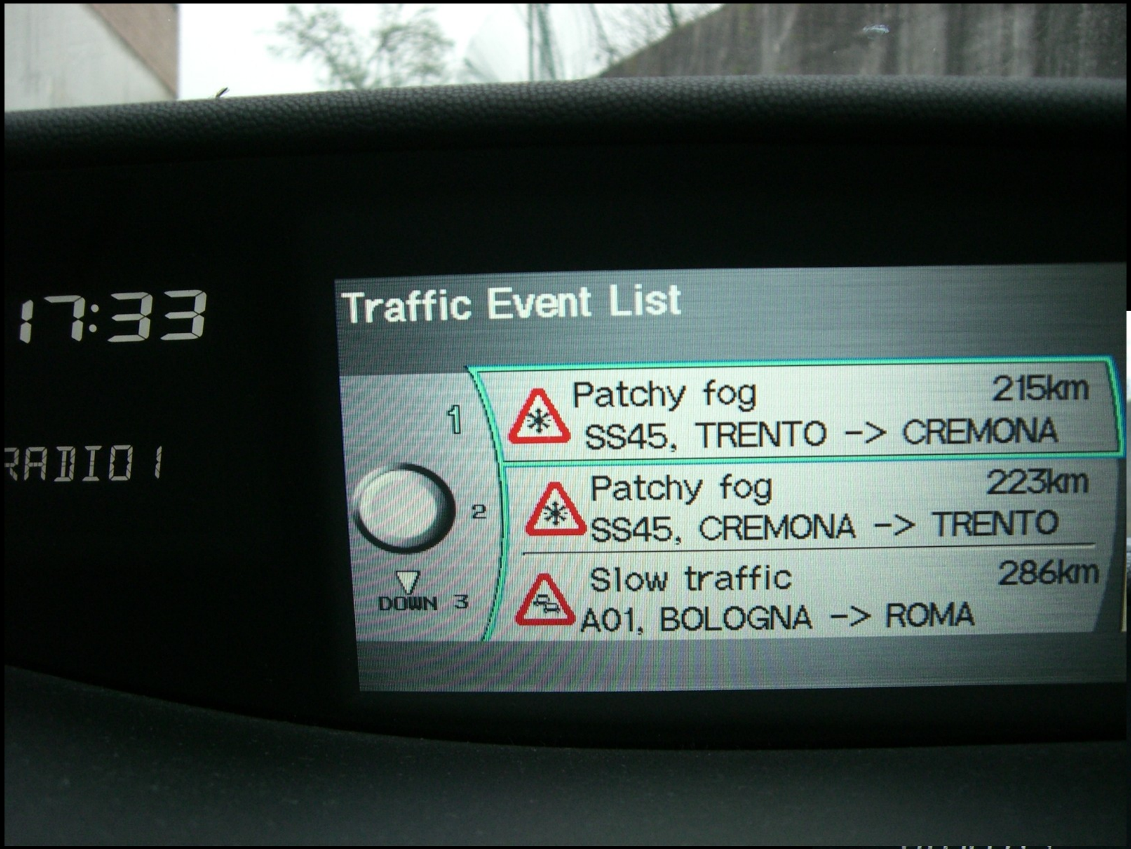
- First introduced around 1997 (Germany), implemented around Europe in the following years (Italy got it in 2004, Australia will get it in 2007)
- Described in ISO 14819-1
- TMC uses RDS for transmission over FM broadcasts



RDS-TMC Implementation

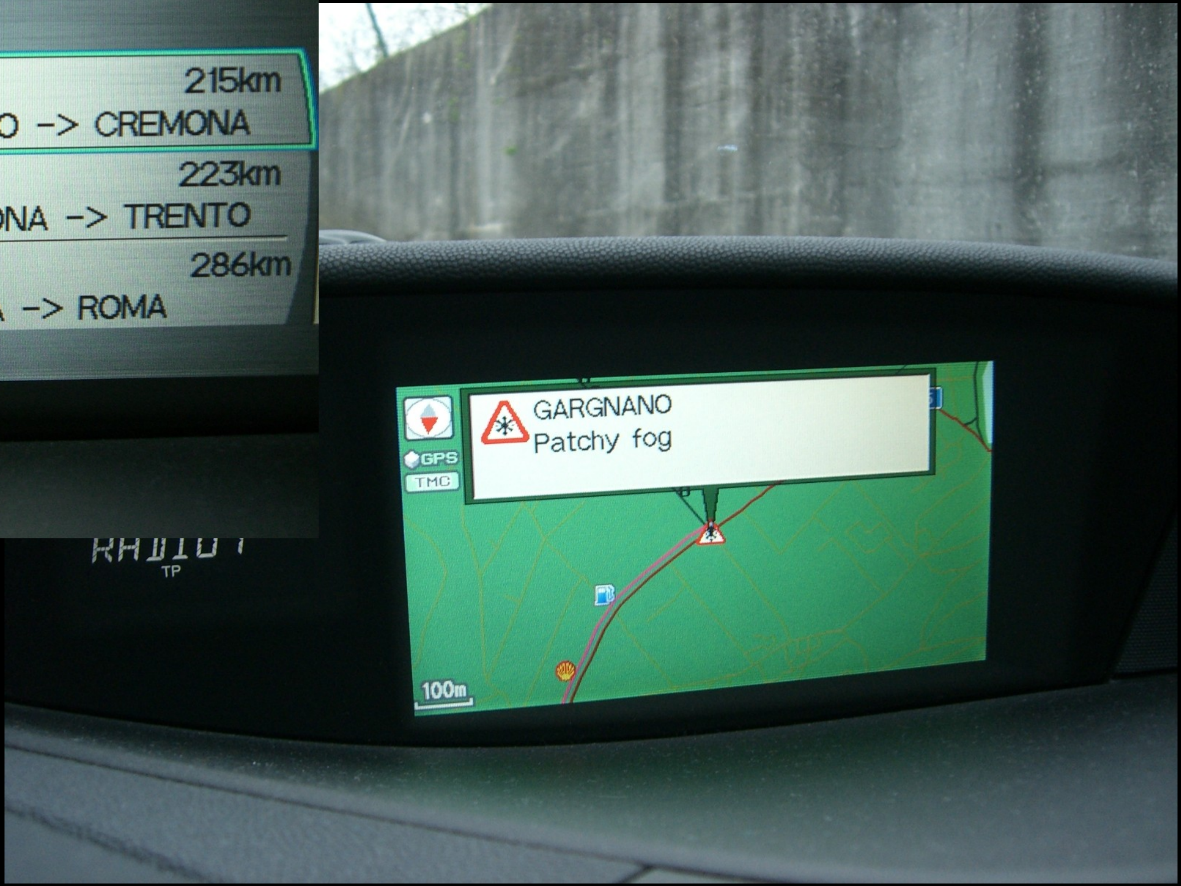
- Despite being a 10 year old protocol, implementation has been slow, SatNav systems have been fully supporting RDS-TMC only in the last few years
- implemented on most in-car SatNav shipped by the original manufacturer
- External and portable SatNav offer jacks for external FM receivers which add RDS-TMC capabilities
- RDS-TMC is available in both free and commercial services
- TMC can also be transmitted over DAB or satellite radio

RDS-TMC Terminal


17:33
RADIO 1

Traffic Event List

1	Patchy fog SS45, TRENTO -> CREMONA	215km
2	Patchy fog SS45, CREMONA -> TRENTO	223km
3	Slow traffic A01, BOLOGNA -> ROMA	286km


GARGNANO
Patchy fog

The Issue

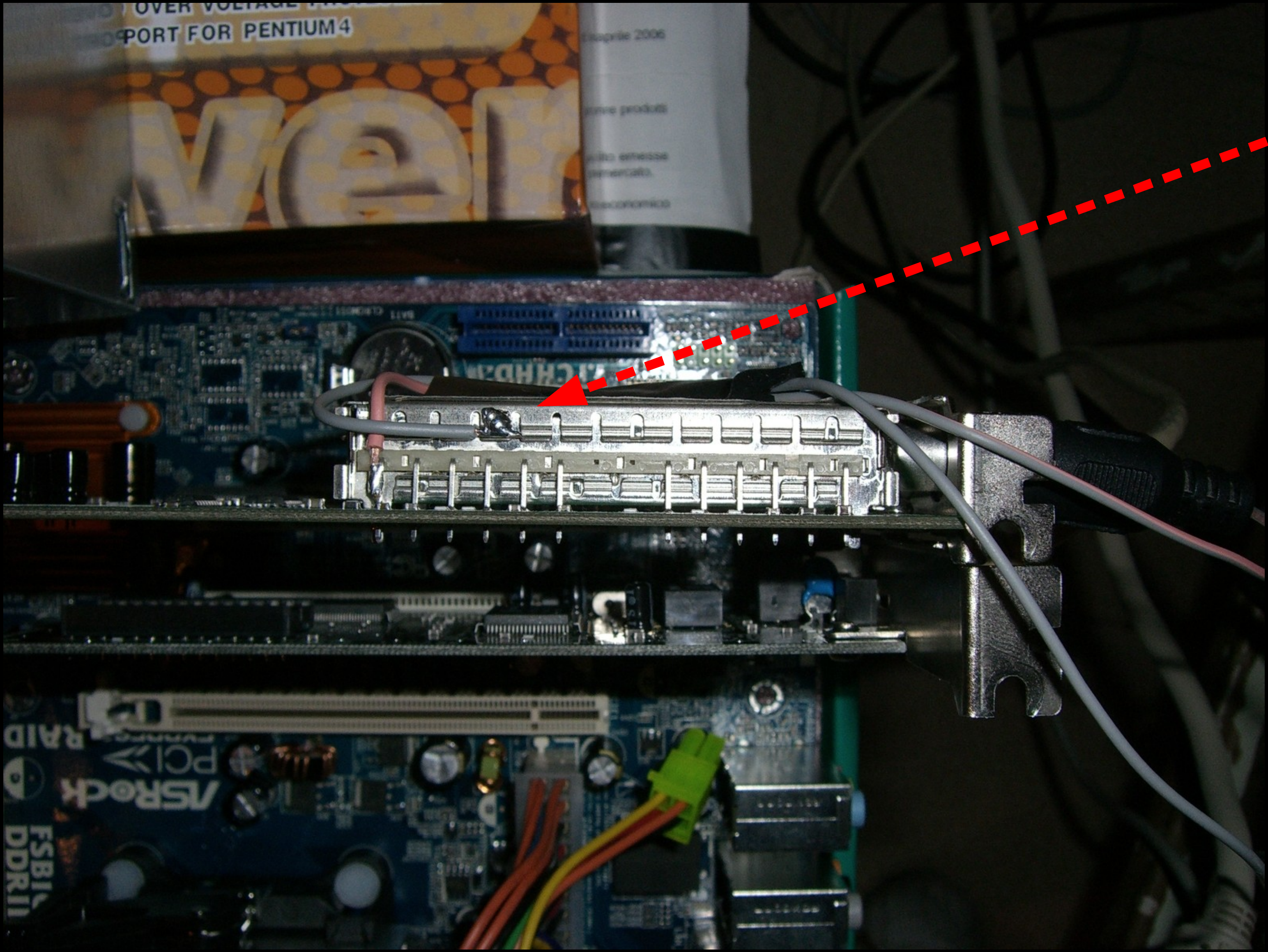
- there's no form of authentication of the data (encryption is supported for commercial services but irrelevant to our goals, more on that later)
- We tested the feasibility of decoding and injecting arbitrary TMC messages against our “victim”
- Off-the-shelf components and cheap electronics have been used
- ...you'll be the judge of our results...

The Victim

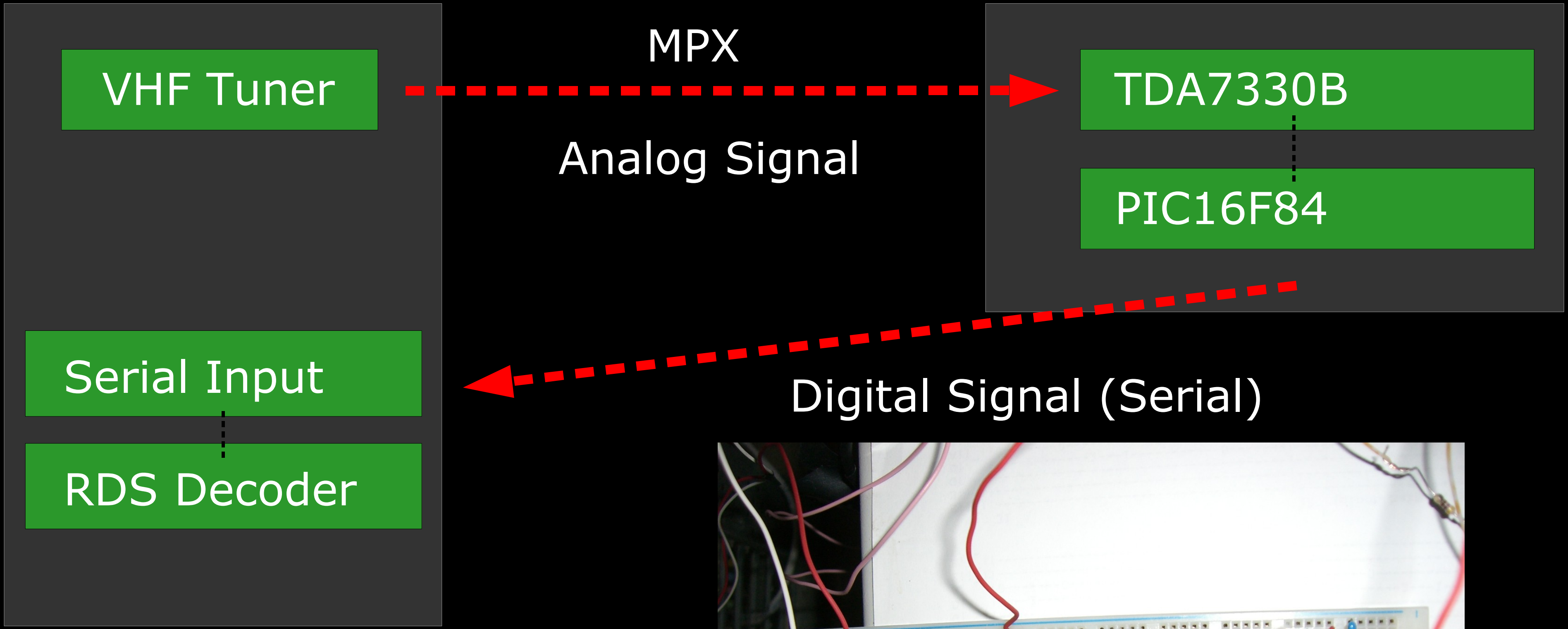
Sniffing RDS

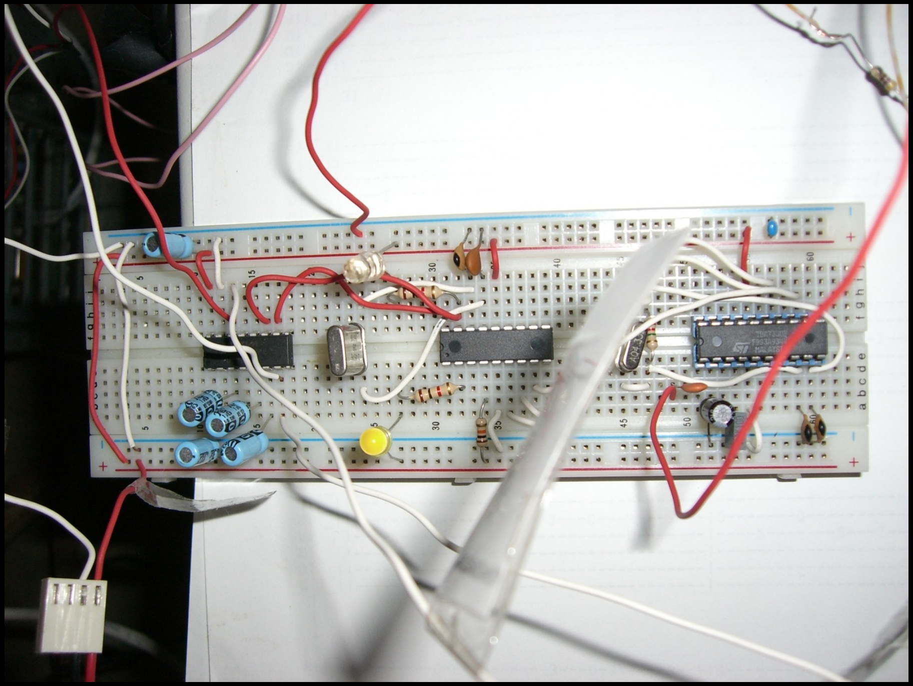
- We need to get a “raw” FM signal (MPX), there's a number of tuners that provide an accessible pin for that
- We use the FM1216 module from Philips available on many PCI TV cards (*http://pvrhw.goldfish.org*)
- Once we have the signal we decode the RDS sub-carrier using a TDA7330B RDS Demodulator (which samples the 1.11875 kHz signal), a PIC for serial conversion and decoding software (sRDSd)
- Using custom hardware and software allowed us to fully understand the protocol and decode TMC (alternatively *http://rdsd.berlios.de* looks like the most promising project)

Sniffing RDS



Sniffing RDS





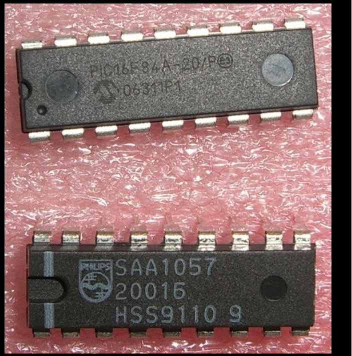
- Main components:

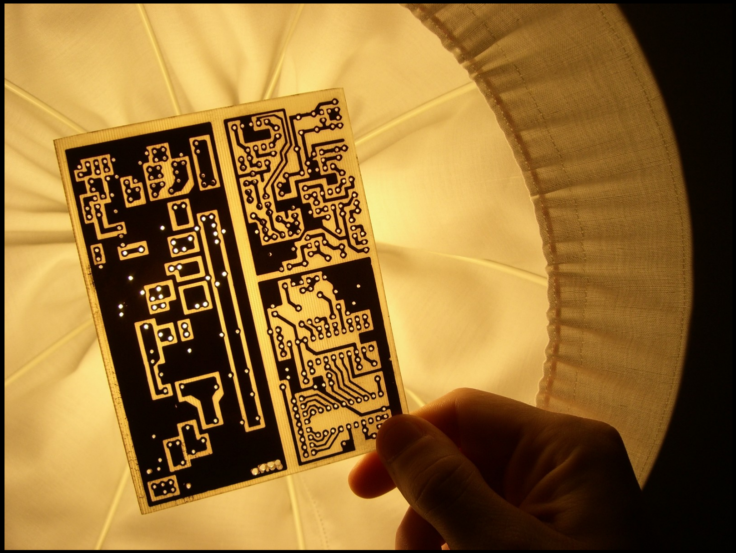
  1x TDA7330B

  1x PIC16F84

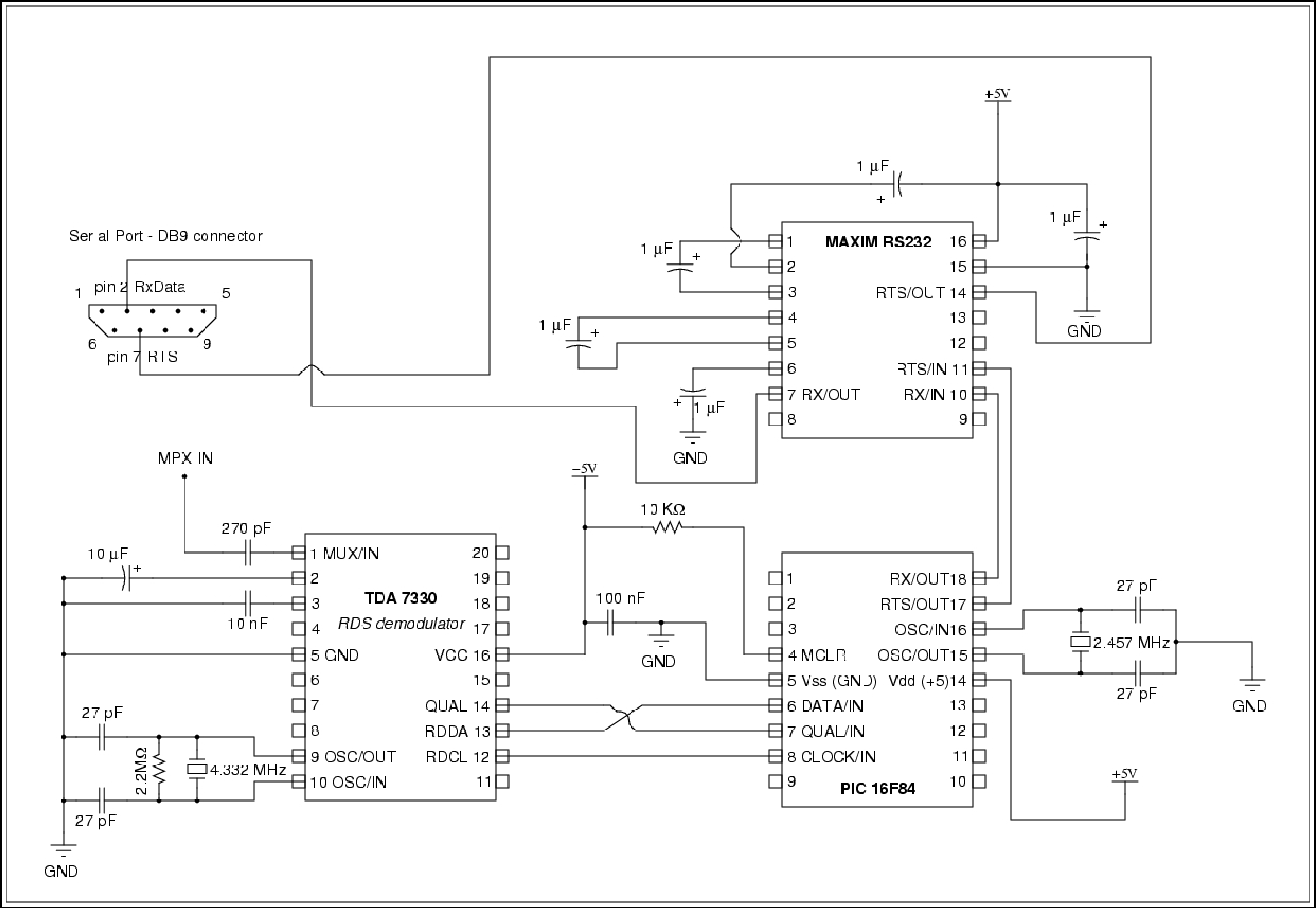
  1x MAX232

Assembly





Sniffing Circuit

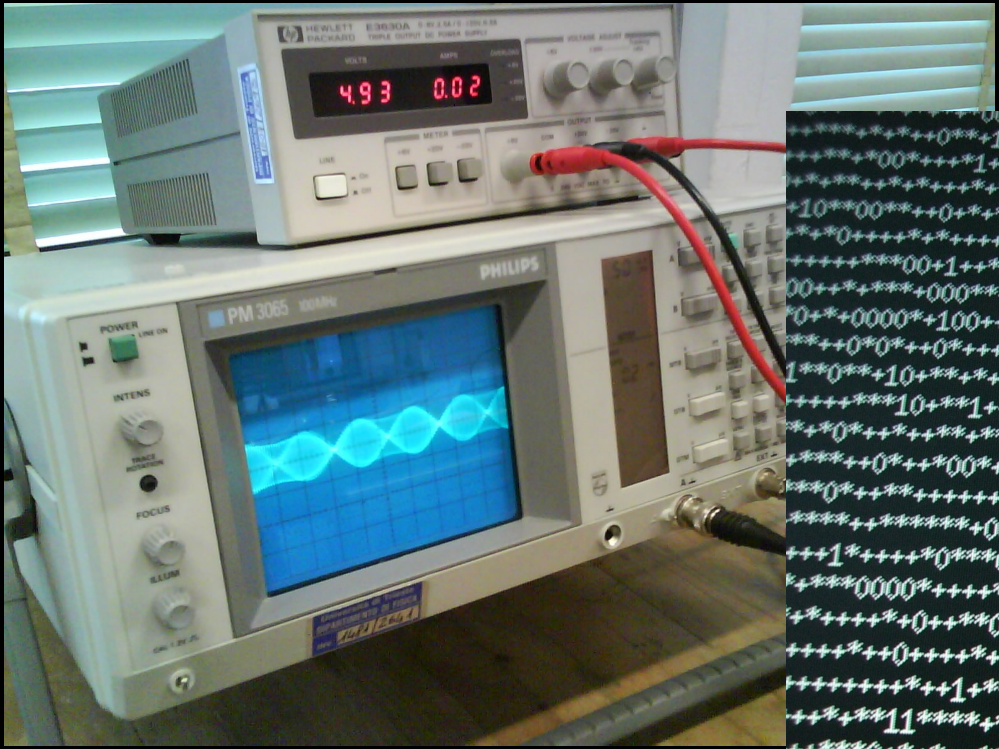


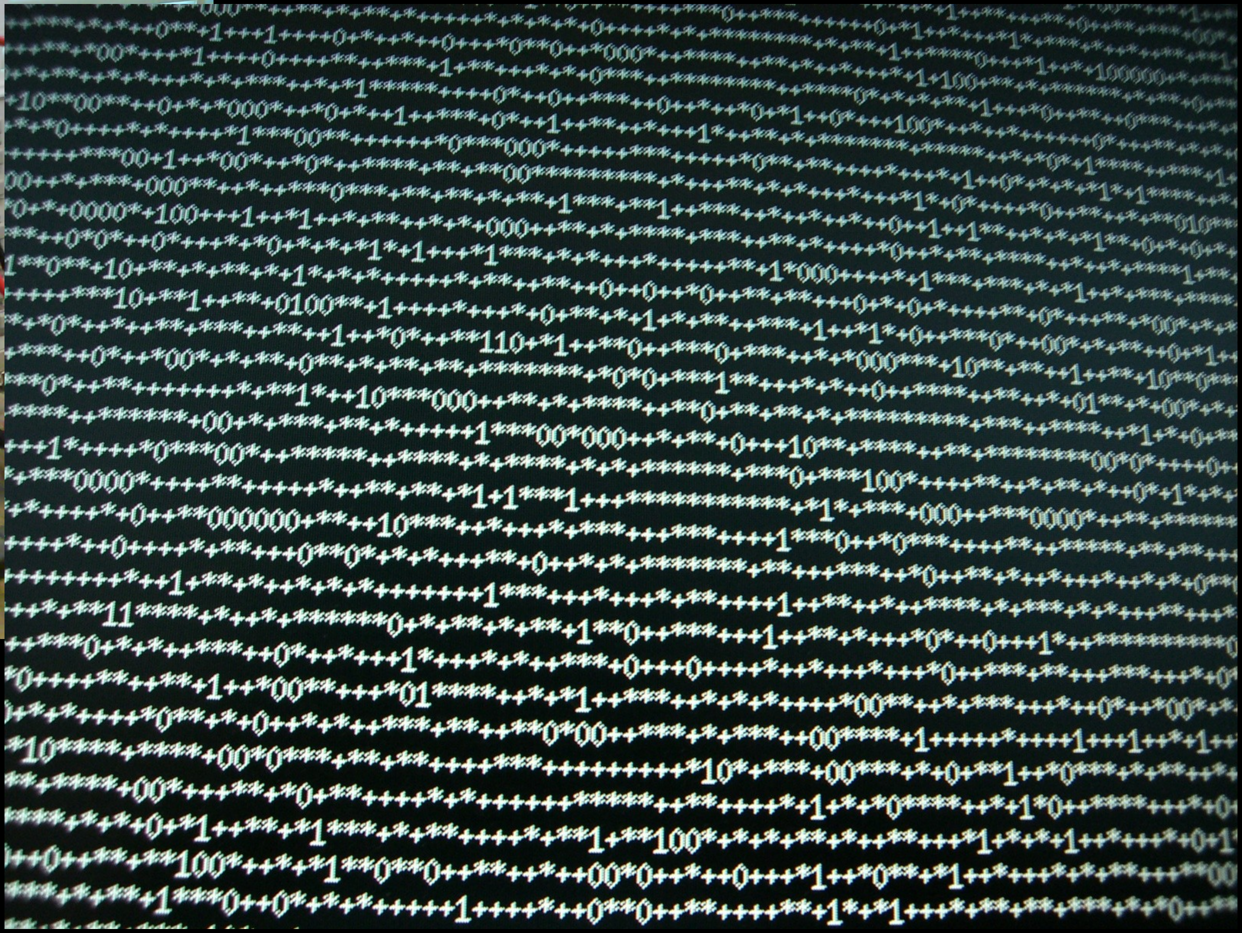
PIC Programming

- We program the PIC for converting RDS Demodulator data and send it to the serial port
- custom PIC programmer, a variation of the well known JDM one (*http://www.semis.demon.co.uk/uJDM/uJDMmain.htm*)
- output are 0 and 1, bad quality data is shown with * and + (either ignore sequences with bad data or replace them with 0 and 1 if you feel lucky)
- *http://dev.inversepath.com/rds/pic_code.asm*

The Output

```
# cat /dev/ttyS0
```





RDS Protocol

```
Group structure (104 bits):
   ---------------------------------------
  | Block 1 | Block 2 | Block 3 | Block 4 |
   ---------------------------------------
Block structure (26 bits):
   ---------------- ----------------------
  | Data (16 bits) | Checkword (10 bits) |
   ---------------- ----------------------
Block 1:
  --------------------
 | PI code | Checkword |
  --------------------
Block 2:
  --------------------------------------------------
 | Group code | B0 | TP | PTY | <5 bits> | Checkword |
  --------------------------------------------------
```

```
PI code    = 16 bits
Group code = 4 bits
B0         = 1 bit
TP         = 1 bit
PTY        = 5 bits
Checkword  = 10 bits
```

TMC / Alert-C Protocol

```
Block 1:
 ---------------------
| PI code | Checkword |
 ---------------------
Block 2:
 ------------------------------------------------------
| Group code | B0 | TP | PTY | T | F | DP | Checkword |
 ------------------------------------------------------
Block 3:
 -------------------------------------
| D | PN | Extent | Event | Checkword |
 -------------------------------------
Block 4:
 ----------------------
| Location | Checkword |
 ----------------------
```

T	= 1 bit
F	= 1 bit
DP	= 3 bits
D	= 1 bit
PN	= 1 bit
Extent	= 3 bits
Event	= 11 bits
Location	= 16 bits
Checkword	= 10 bits

TMC / Alert-C

PI code => Programme Identification

Group code => message type identification

B0 => version code

TP => Traffic Program

PTY => Programme Type

T, F, D => Multi Group messages

DP => Duration and Persistence

D => Diversion Advice

PN => +/- direction

Extent => event extension

Event => event code (see also TMDD – Traffic Management Data Dictionary)

Location => location code (DAT Location Table - TMCF-LT-EF-MFF-v06)

srdsd
Simple RDS Decoder

- Our custom tool for RDS decoding:
  - ISC-style licensed
  - performs nearly full RDS-TMC (and basic RDS) decoding
  - text and HTML output with Google Map links of GPS data
  - *http://dev.inversepath.com/rds/srdsd*

```
Simple RDS-TMC Decoder 0.1      || http://dev.inversepath.com/rds
Copyright 2007 Andrea Barisani || <andrea@inversepath.com>
Usage: ../srdsd/srdsd [-h|-H|-P|-t] [-d <location db path>] [-p
  <PI number>] <input file>
   -t display only tmc packets
   -H HTML output (outputs to /tmp/rds-<random>/rds-*.html)
   -p PI number
   -P PI search
   -d location db path
   -h this help

Note: -d option expects a DAT Location Table code according to
      TMCF-LT-EF-MFF-v06 standard (2005/05/11)
```

srdsd – PI Search

- We must “lock” parsing to the relevant PI
- Every FM Channel has its own code (google knows)
- You can guess the PI code by finding the most recurring 16-bit string:

```
# ./srdsd -P rds_dump.raw | tail

0010000110000000: 4140 (2180)
1000011000000001: 4146 (8601)
0001100000000101: 4158 (1805)
1001000011000000: 4160 (90c0)
0000110000000010: 4163 (0c02)
0110000000010100: 4163 (6014)
0011000000001010: 4164 (300a)
0100100001100000: 4167 (4860)
1010010000110000: 4172 (a430)
0101001000011000: 4185 (5218)

# ./srdsd -p 5218 -d ~/loc_db/ rds_dump.raw
```

srdsd output – 0A Group

```
Got RDS message (frame 75)
        Programme Identification: 0101001000011000 (5218)
        Group type code/version: 0000/0 (0A  - Tuning)
        Traffic Program: 1
        Programme Type: 01001 (9  - Varied Speech)
        Decoded 0A group:
                Traffic Announcement: 0
                Music Speech switch: 0
                Decoder Identification control: 100
                (Dynamic Switch / PS char 1,2)
                Alternative Frequencies: 10101010, 10101111
                (104.5, 105)
                Programme Service name: 0101001001010100 (RT)
                Collected PSN: RTL102.5

        Raw dump | Data             Checkword  Hex
        Block 1: | 0101001000011000 0000010100 5218
        Block 2: | 0000010100101100 0010101101 052c
        Block 3: | 1010101010101111 1010100110 aaaf
        Block 4: | 0101001001010100 0100110101 5254
```

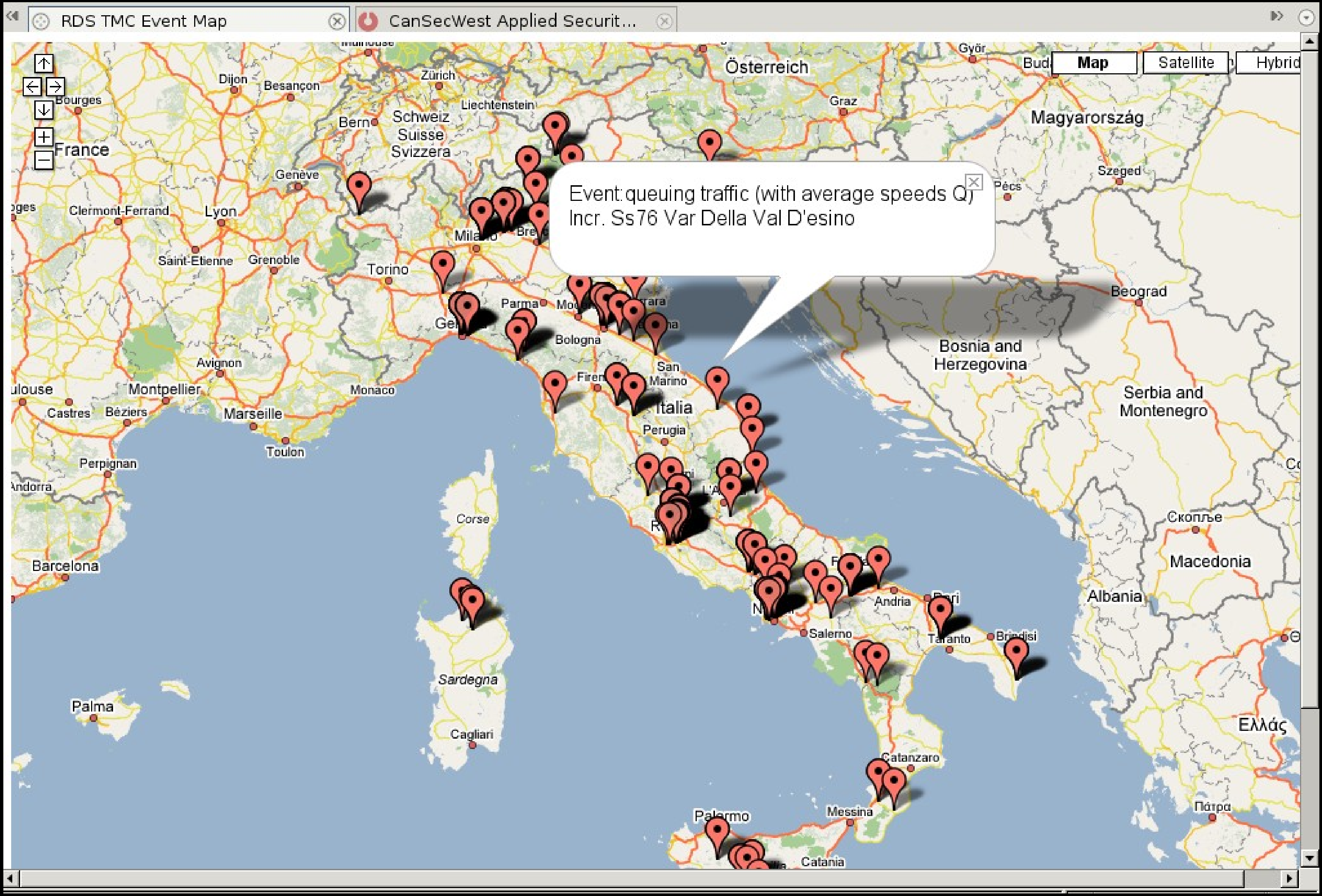
srdsd output – 8A Group

```
Got RDS message (frame 76)
        Programme Identification: 0101001000011000 (5218)
        Group type code/version: 1000/0 (8A  - TMC)
        Traffic Program: 1
        Programme Type: 01001 (9  - Varied Speech)
        Decoded 8A group:
                Bit X4: 0 (User message)
                Bit X3: 1 (Single-group message)
                Duration and Persistence: 000 (no explicit duration given)
                Diversion advice: 0
                Direction: 1 (-)
                Extent: 011 (3)
                Event: 00001110011 (115 - slow traffic (with average speeds Q))
                Location: 0000110000001100 (3084)
                Decoded Location:
                        Location code type: POINT
                        Name ID: 11013 (Sv. Grande Raccordo Anulare)
                        Road code: 266 (Roma-Ss16)
                        GPS: 41.98449 N 12.49321 E
                        Link:
  http://maps.google.com/maps?ll=41.98449,12.49321&spn=0.3,0.3&q=41.98449,12.49321

        Raw dump | Data             Checkword  Hex
        Block 1: | 0101001000011000 0000010100 5218
        Block 2: | 1000010100101000 1110000111 8528
        Block 3: | 0101100001110011 0001011001 5873
        Block 4: | 0000110000001100 0111000011 0c0c
```

srdsd output – 3A Group

```
Got RDS message (frame 181)
        Programme Identification: 0101001000011000 (5218)
        Group type code/version: 0011/0 (3A  - ODA ID)
        Traffic Program: 1
        Programme Type: 01001 (9  - Varied Speech)
        Decoded TMC Sys Info group (3A - AID 52550):
                Location Table Number: 000001 (1)
                Alternative Frequency bit: 1
                Mode of Transmission: 0
                International Scope: 1
                National Scope: 0
                Regional Scope: 0
                Urban Scope: 0
                AID: 1100110101000110 (52550)

        Raw dump | Data             Checkword  Hex
        Block 1: | 0101001000011000 0000010100 5218
        Block 2: | 0011010100110000 1111101000 3530
        Block 3: | 0000000001101000 0010011011 0068
        Block 4: | 1100110101000110 1111001001 cd46
```

srdsd + Google Maps



Event: queuing traffic (with average speeds Q)
Incr. Ss76 Var Della Val D'esino

Injecting RDS-TMC

Video Clip time!

Injecting RDS-TMC

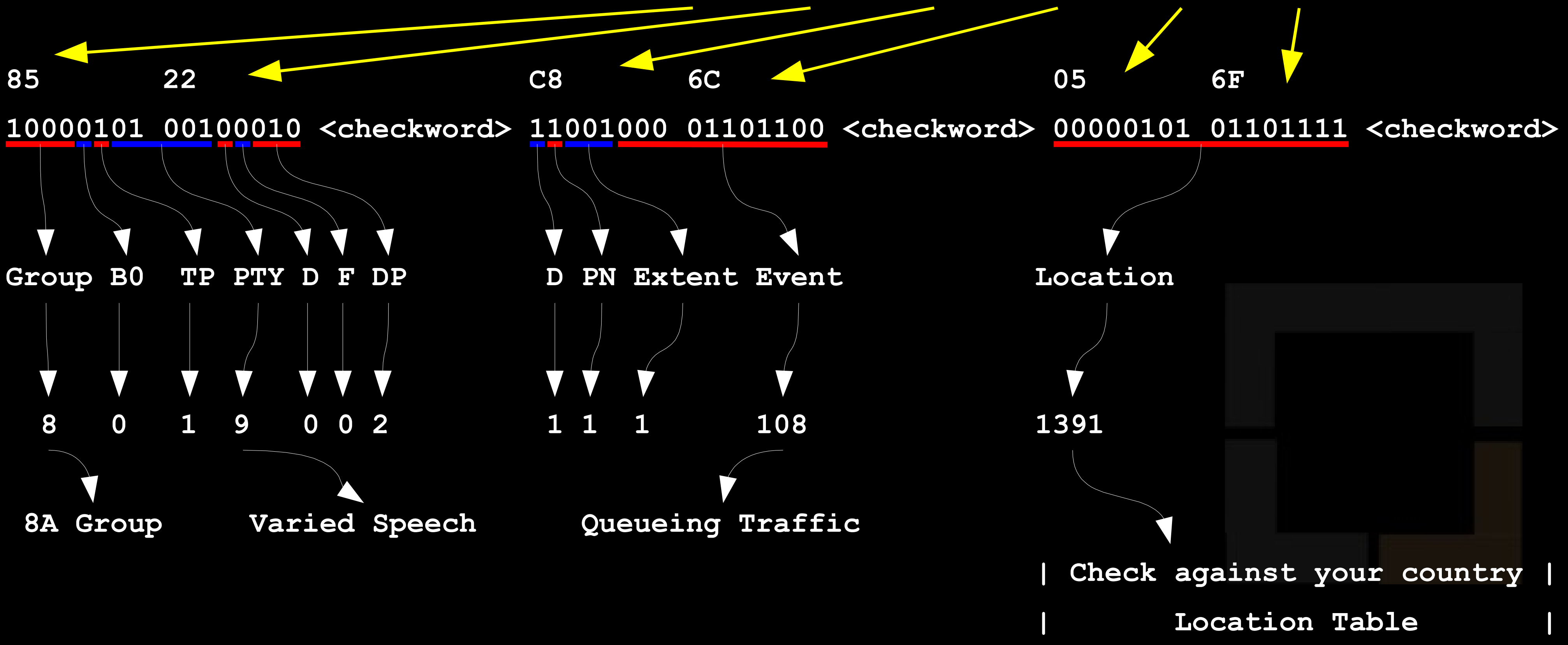
* WARNING: Your Experience May Differ

Injecting RDS-TMC

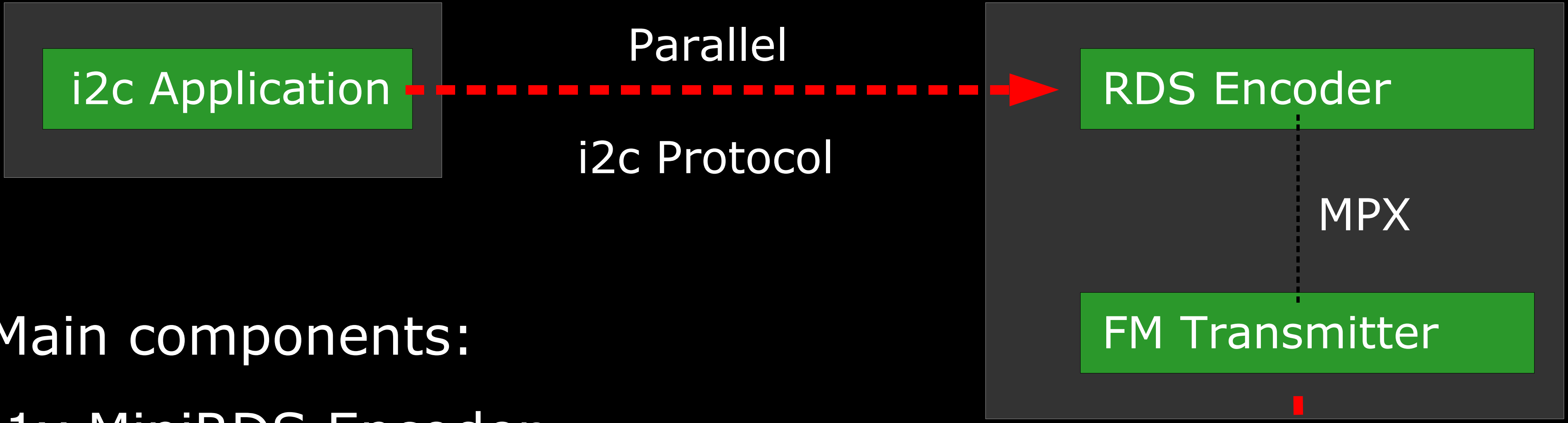
- We use a commercialy available RDS encoder (40$ USD), but it's reasonable to build your own (we are working on it)
- i2c is being used for communicating with its chipset, we use our custom C application over the supplied client for being able to send different Group Types
- We set all parameters (PI, PTY, etc) + the remaining data (last 3 RDS Blocks in Hexadecimal)
- The checkword is automatically computed by the chipset
- *http://dev.inversepath.com/rds/i2c_minirds.tar.gz*

Injecting RDS-TMC

```
unsigned char PI_buf[PI_BUF]     = { '\x52', '\x18' };                            /* PI */
unsigned char PS_buf[PS_BUF]     = { 'R', 'A', 'D', 'I', '0', '1', '0', '5' };   /* PS */
...
unsigned char UDG2_buf[UDG2_BUF] = {'\x35','\x30','\x00','\x66','\xCD','\x46'}; /* 3A */
unsigned char UDG1_buf[UDG1_BUF] = {'\x85','\x22','\xC8','\x6C','\x05','\x6F'}; /* 8A */
```



Injecting RDS-TMC



Main components:

1x MiniRDS Encoder

(*http://www.pira.cz*)

1x FM transmitter

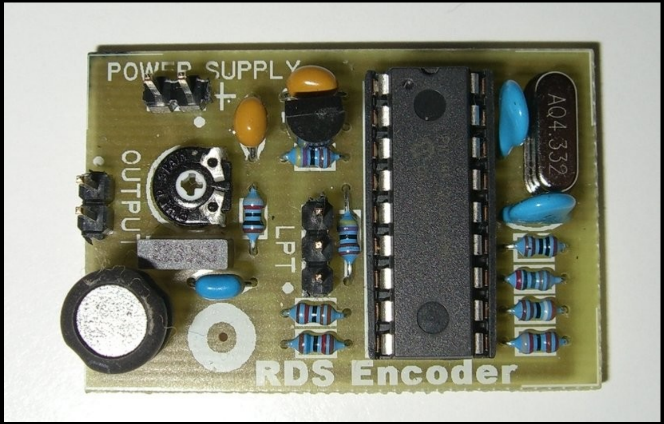
1x PIC16F84

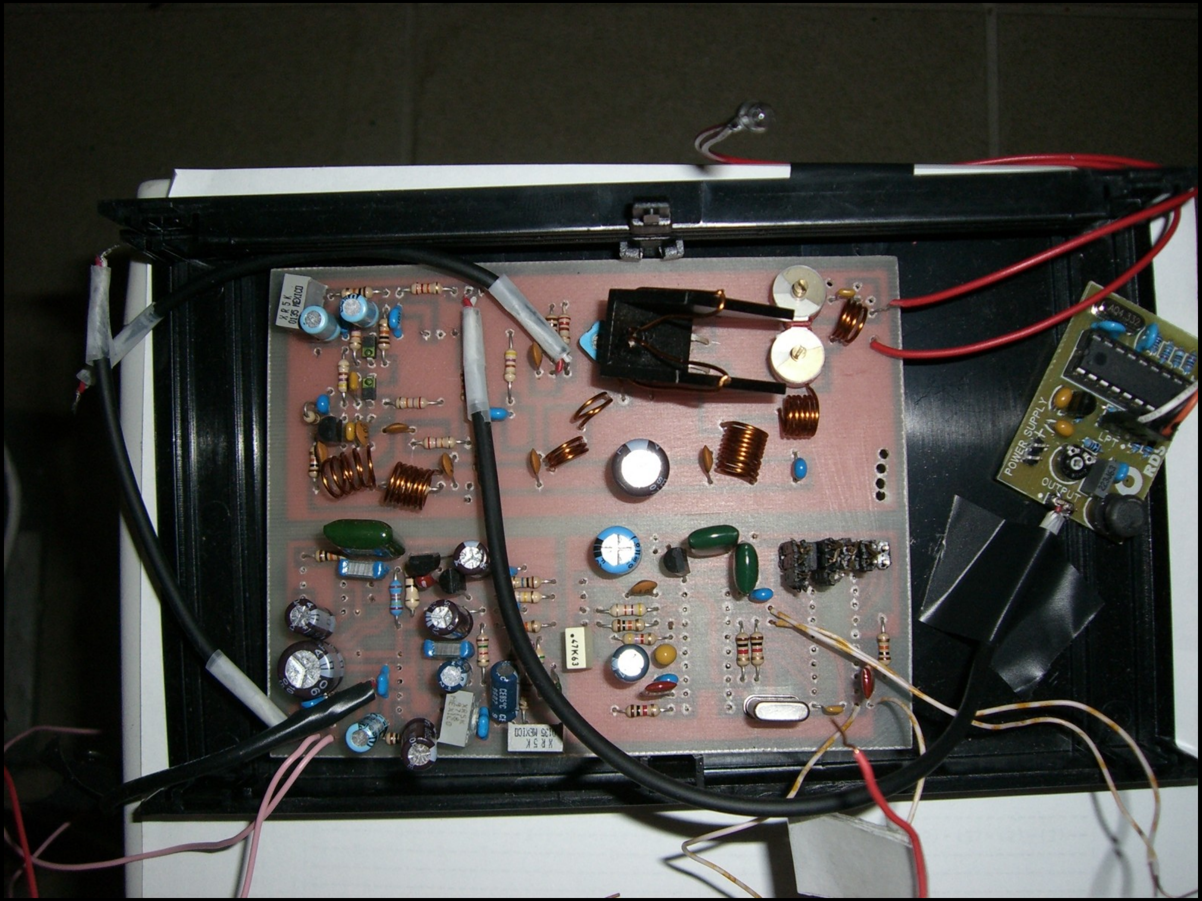
1x SAA1057 (digital PLL tuning)

1x closed dipole antenna

TX Antenna

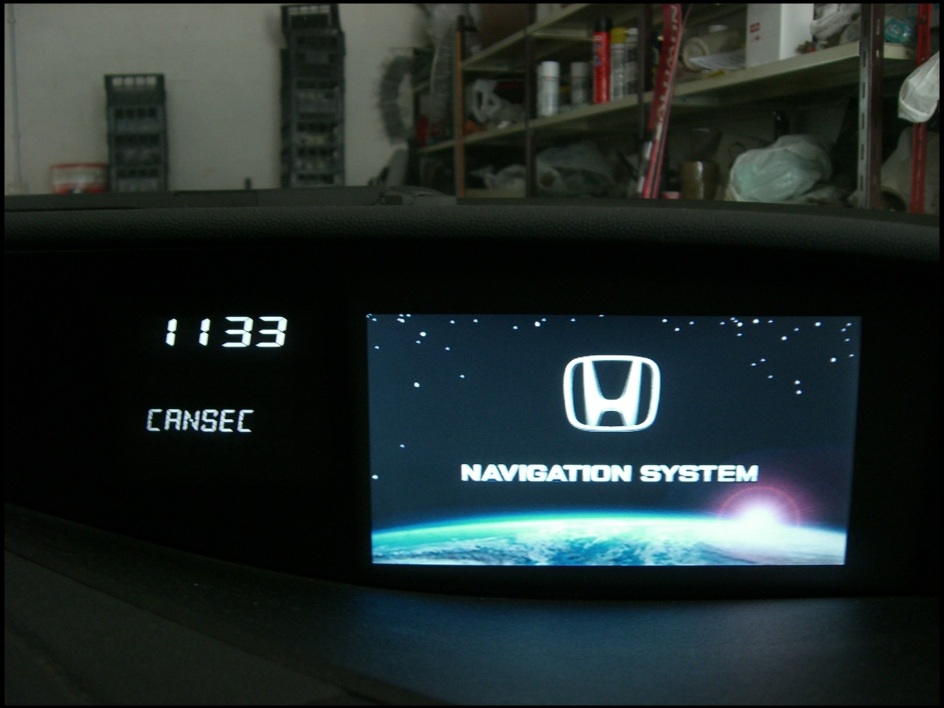
Injection Circuitry

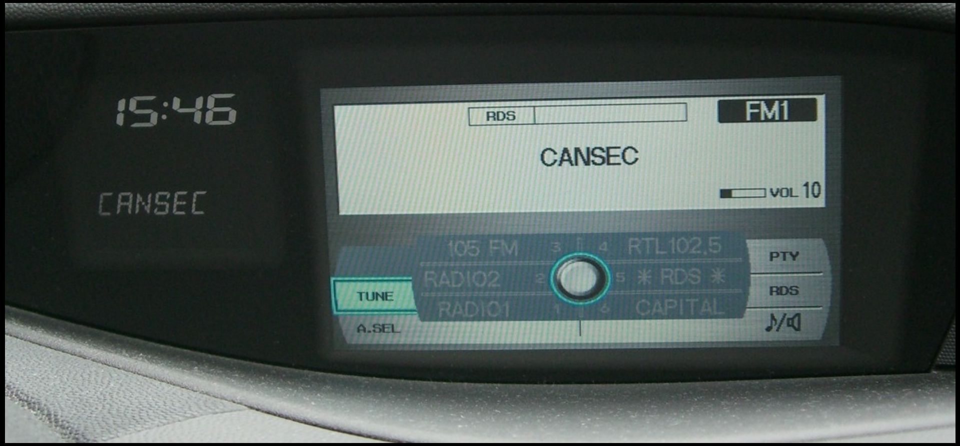




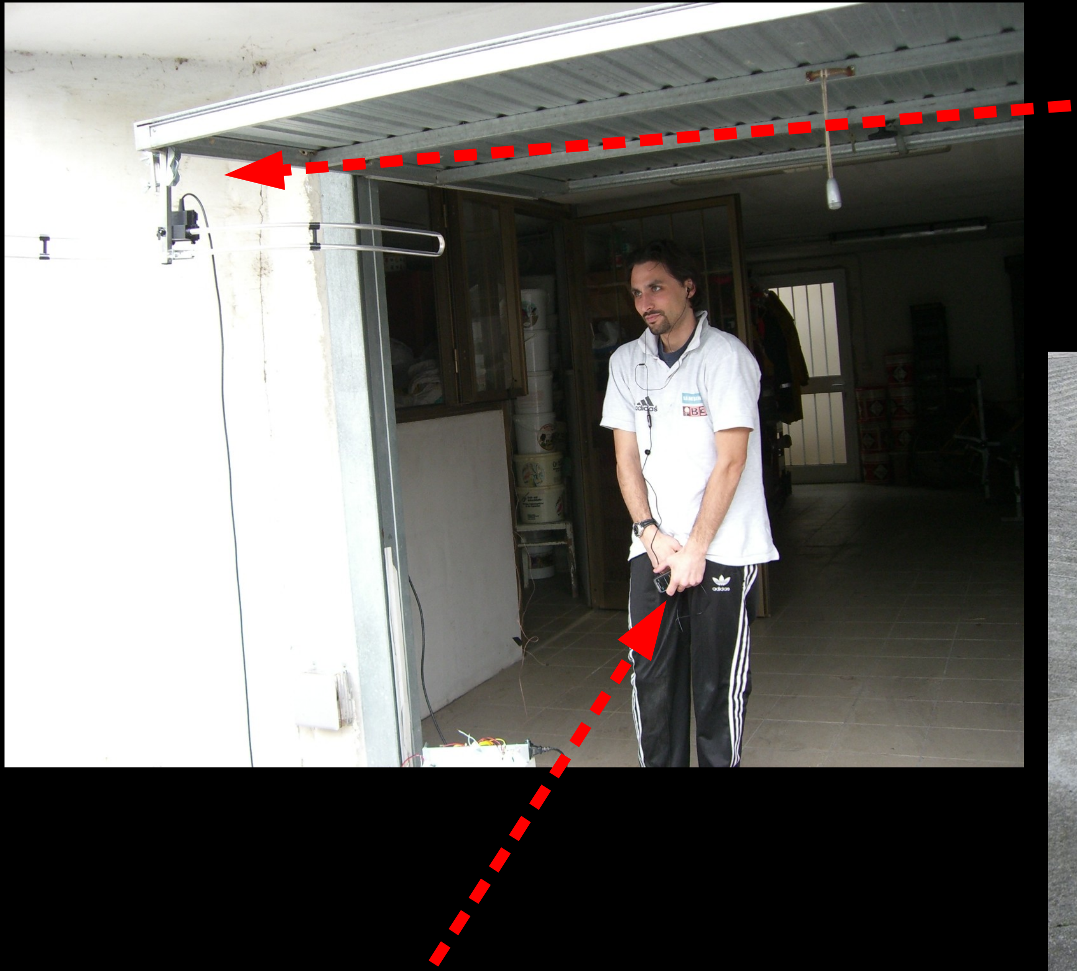
Transmitting FM

- The FM transmitter can be tuned to arbitrary frequencies
- It's important to have a stable transmitter for data injection
- Long distances can be easily covered (but it might be desirable to keep it short enough to reach only the victim)





Transmitting FM



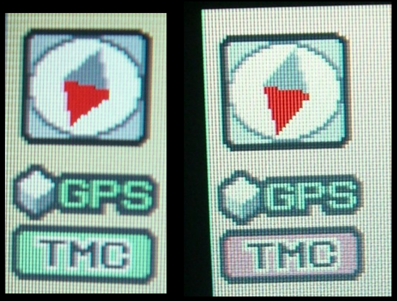
TX “*The Sterilizer*” Antenna

(Resistance is Futile)

Transmitting FM

Video Clip time!

Locking the SatNav Tuner

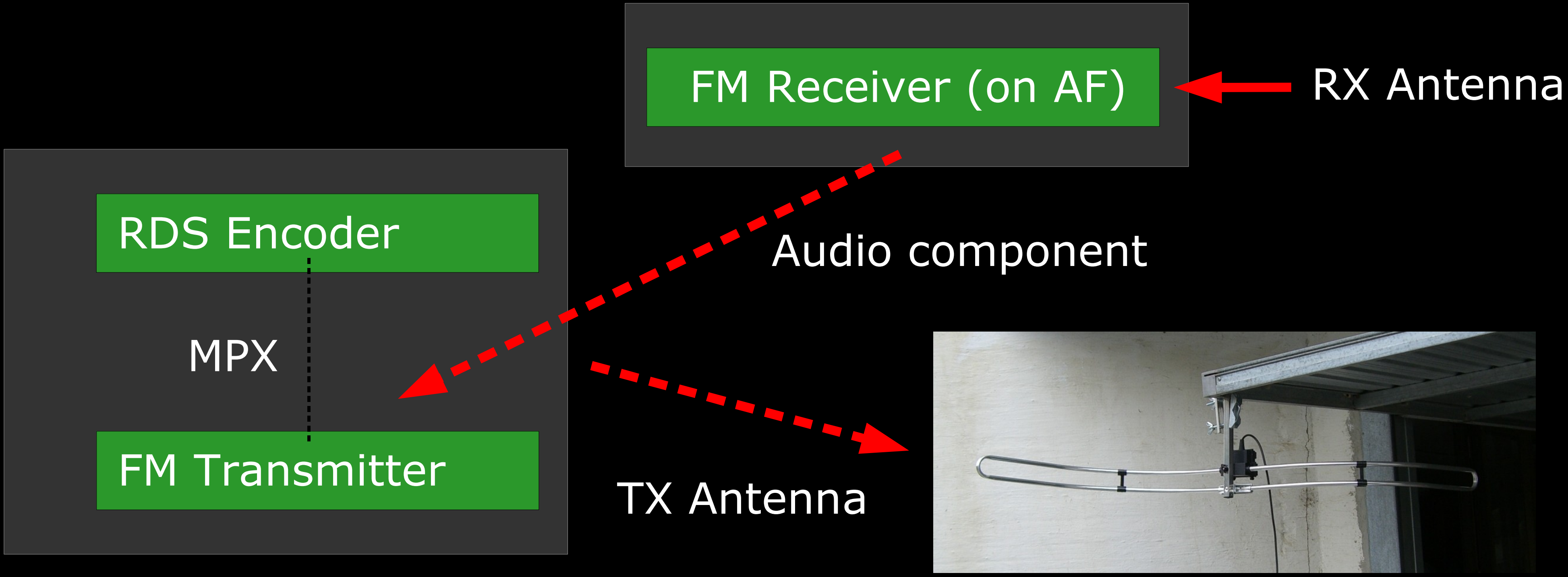


- RDS-TMC is detected using 3A Sys Info groups which specify the Location Table, the Scope of the service and timing settings
- Hijack existing channels:
  1. Find the frequency of a channel that provides RDS-TMC
  2. Obscure the channel and send 8A packets (3A not necessary) when SatNav locks on it (careful timing)
- Fake a FM broadcast using 3A groups:
  1. Find an unused frequency
  2. Transmit 3A groups continuosly + 8A packets

Being Stealthy

Option 1: Mix the audio component taken on the Alternate Frequency (AF) for the hijacked channel

Option 2: Fake a new channel on an unused frequency



Attack 1: Standard Traffic Msgs

- We can create:
  1. Queues
  2. Bad Weather (Rain, Smog, Fog, Fresh Snow,...)
  3. Full Car Parks
  4. Overcrowded Service Areas (OMG!)
  5. Accidents
  6. Roadworks

  ...and so on...

- Not particularly exciting but still nice...it gets better though...

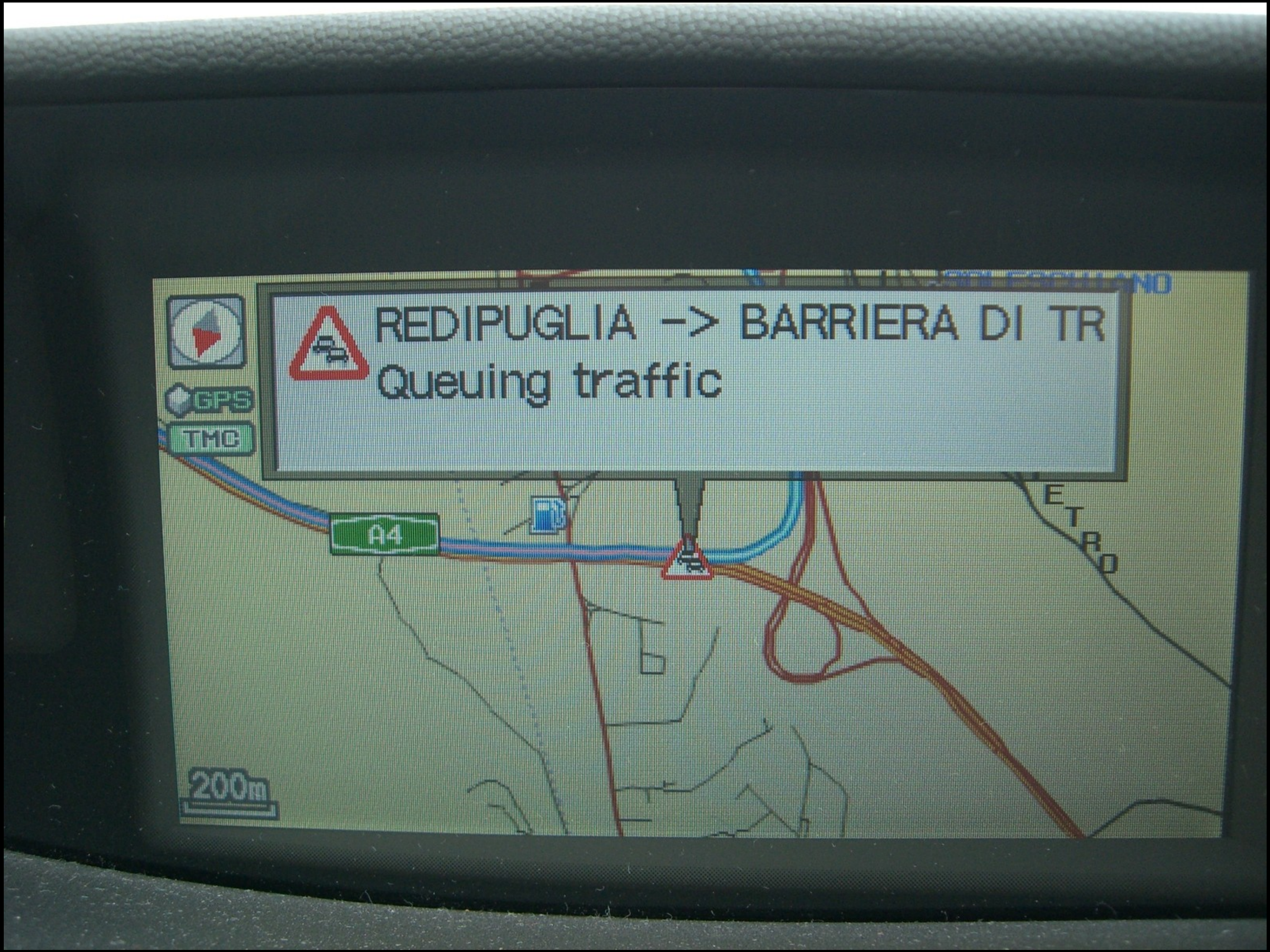
Attack 1: Standard Traffic Msgs

Code 108

-

Queueing

Traffic

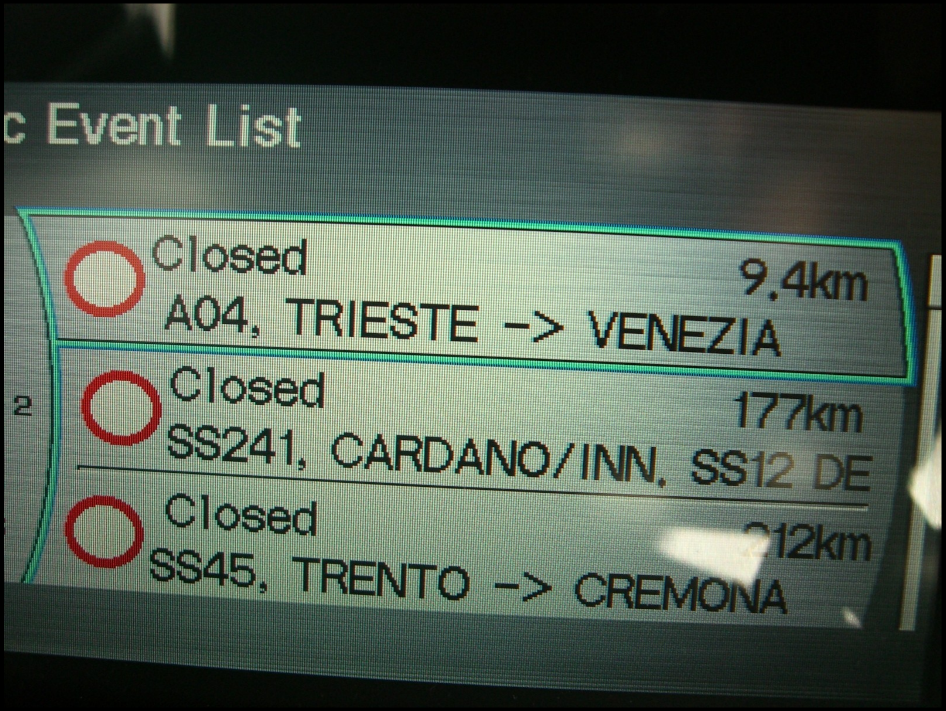

REDIPUGLIA -> BARRIERA DI TR
Queuing traffic

Attack 2: Closing Roads

- We can close arbitrary roads, bridges and tunnels with a number of Events: Closed, No through traffic, Accidents
- The SatNav will pop-up the event (even if no diversion is specified on our model) and ask the user for a detour
- If the closed road is encountered during re-calculation of the route (which is a very common thing) it will be *silently* avoided

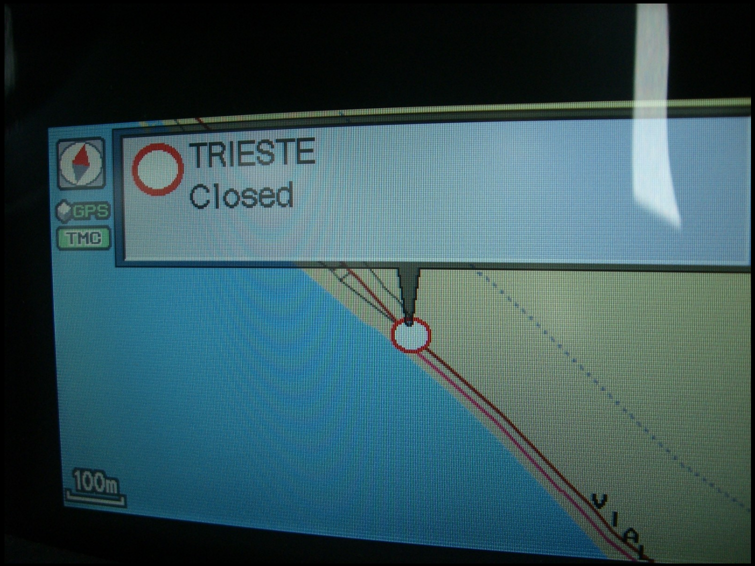
- this attack is also known as “keep your parents from reaching home”...

Attack 2: Closing Roads

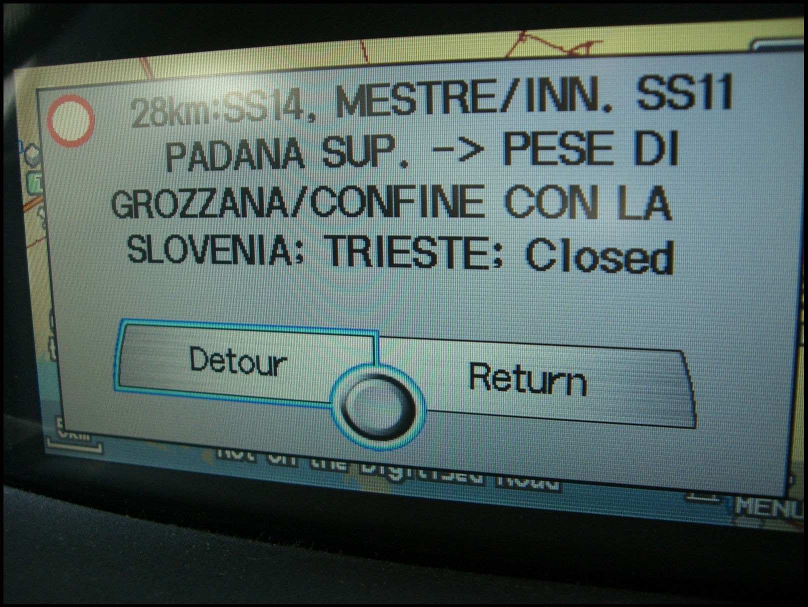


c Event List

Closed	9.4km
A04, TRIESTE -> VENEZIA	
Closed	177km
SS241, CARDANO/INN. SS12 DE	
Closed	212km
SS45, TRENTO -> CREMONA	



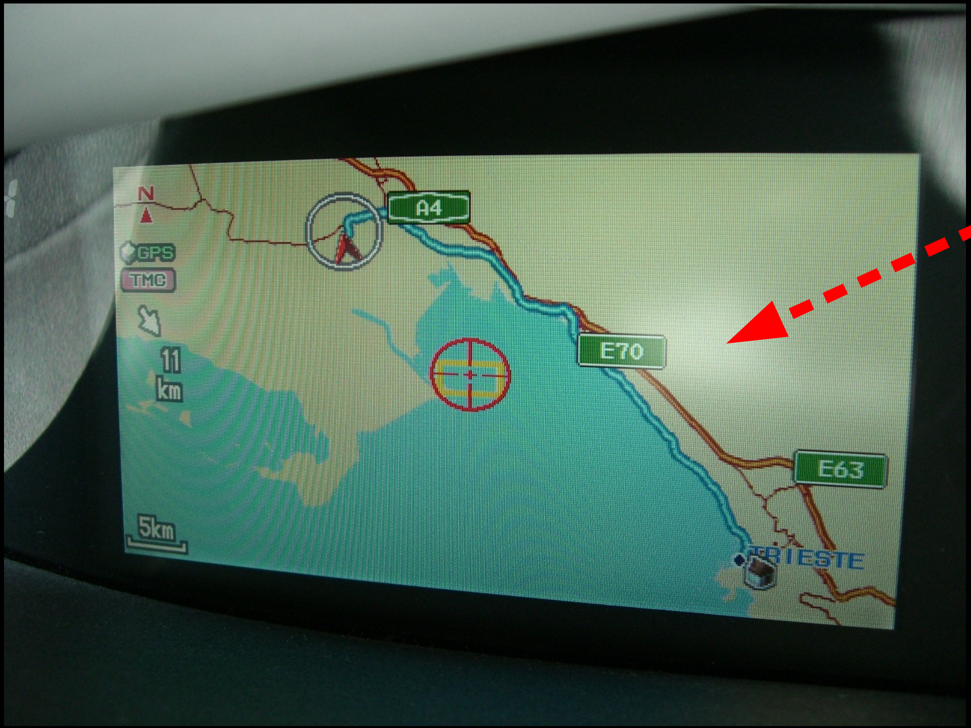
TRIESTE
Closed



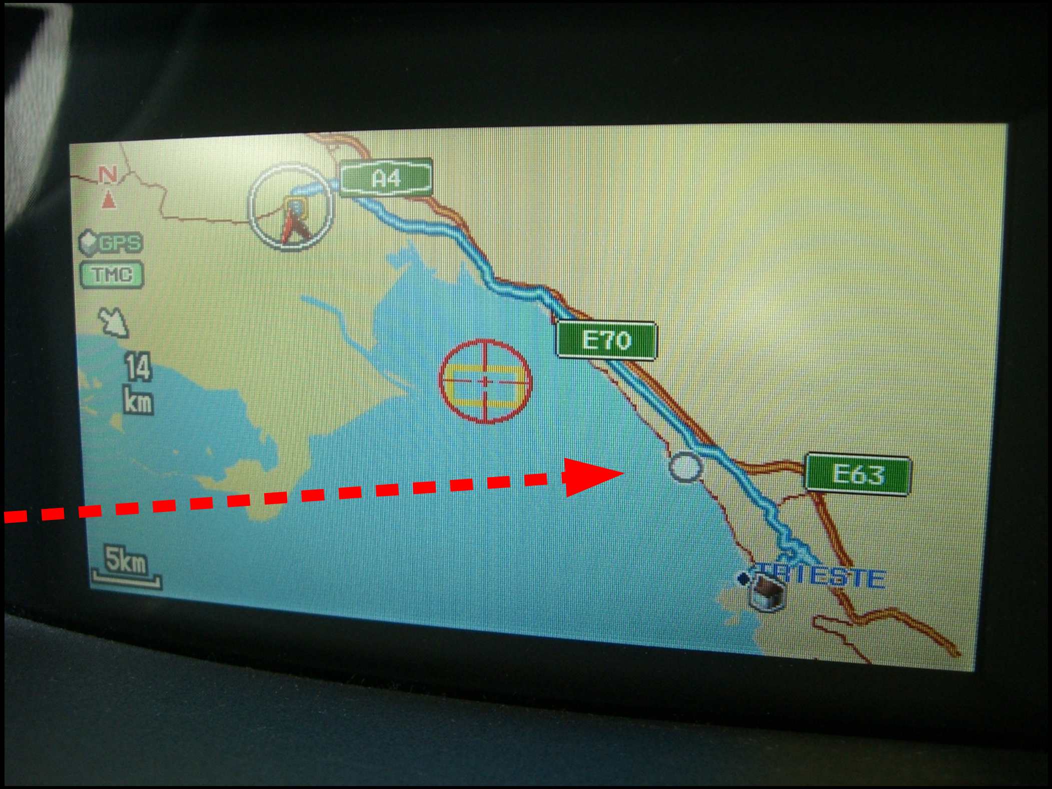
28km:SS14, MESTRE/INN. SS11 PADANA SUP. -> PESE DI GROZZANA/CONFINE CON LA SLOVENIA; TRIESTE; Closed

Code 401 - Closed

Attack 2: Closing Roads



Normal route to home

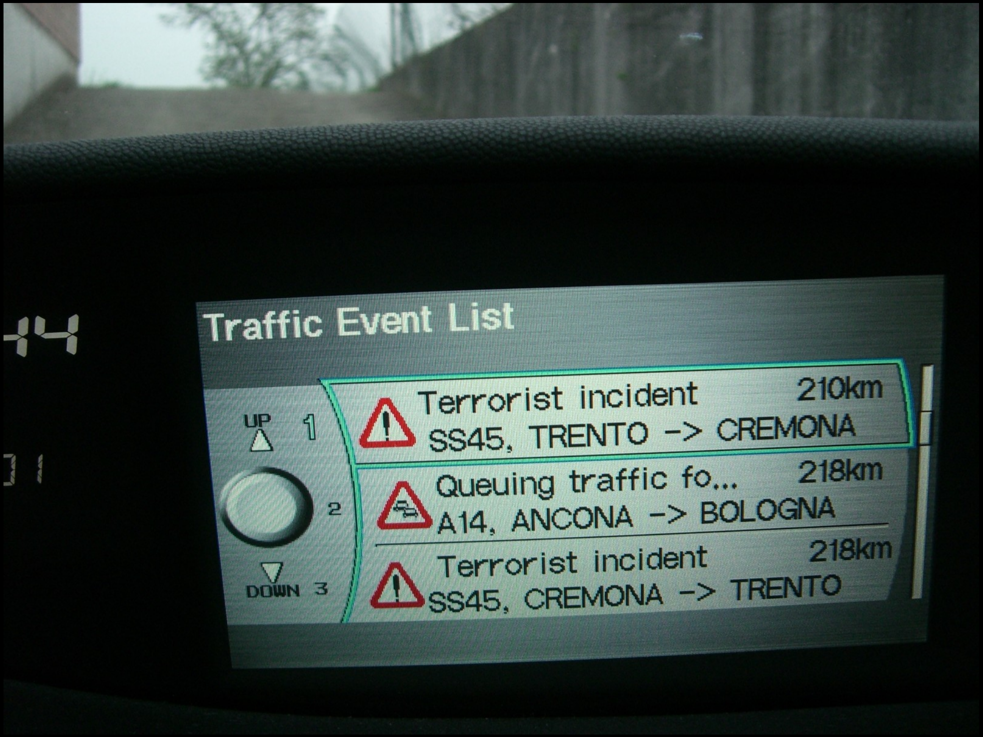


Route avoiding the "Closed" Event

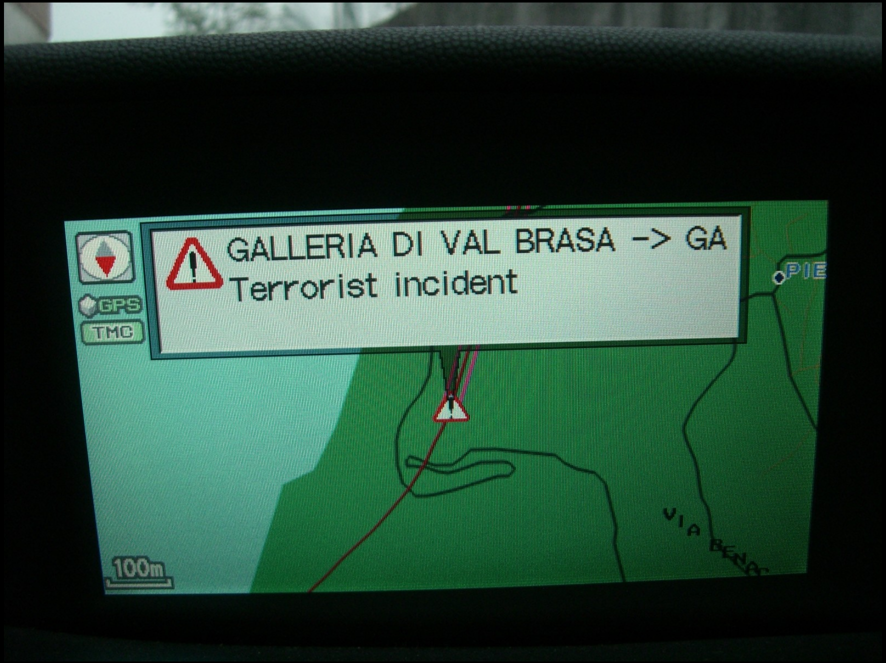
Attack 3:
Security Messages

- The Event table supports a number of security related messages
- We doubt anyone ever used them so far
- They pose a very interesting target for social engineering purposes (Homeland Security would freak out)

Attack 3: Security Messages

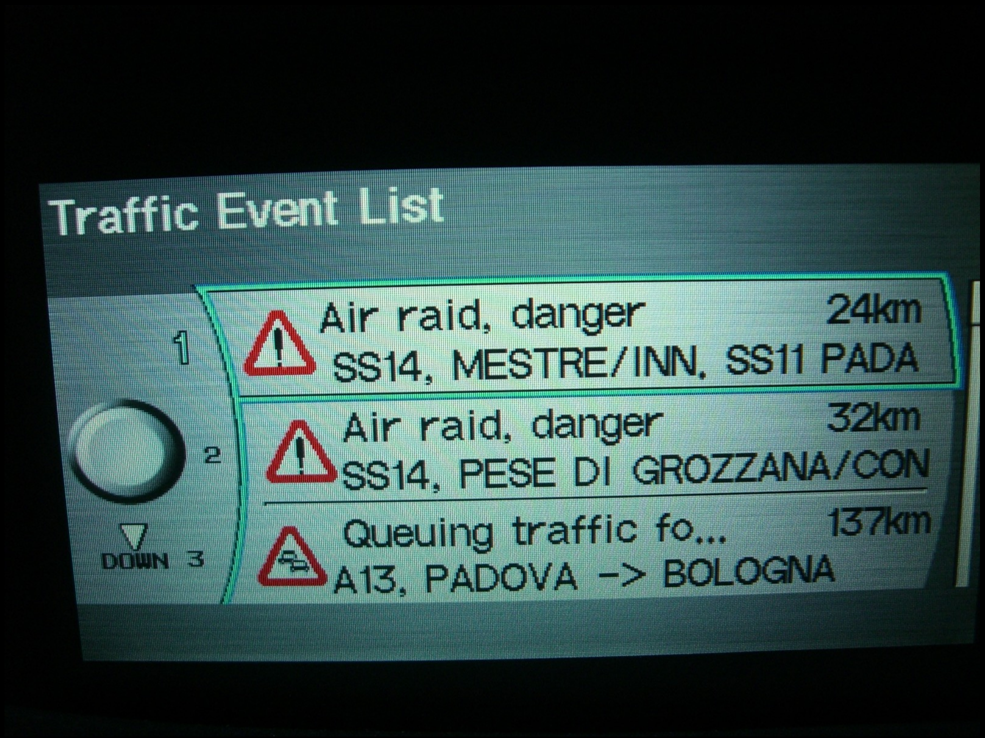

Traffic Event List

1	Terrorist incident SS45, TRENTO -> CREMONA	210km
2	Queuing traffic fo... A14, ANCONA -> BOLOGNA	218km
3	Terrorist incident SS45, CREMONA -> TRENTO	218km


GALLERIA DI VAL BRASA -> GA
Terrorist incident

Code 1518 – Terrorist Incident

Attack 3: Security Messages

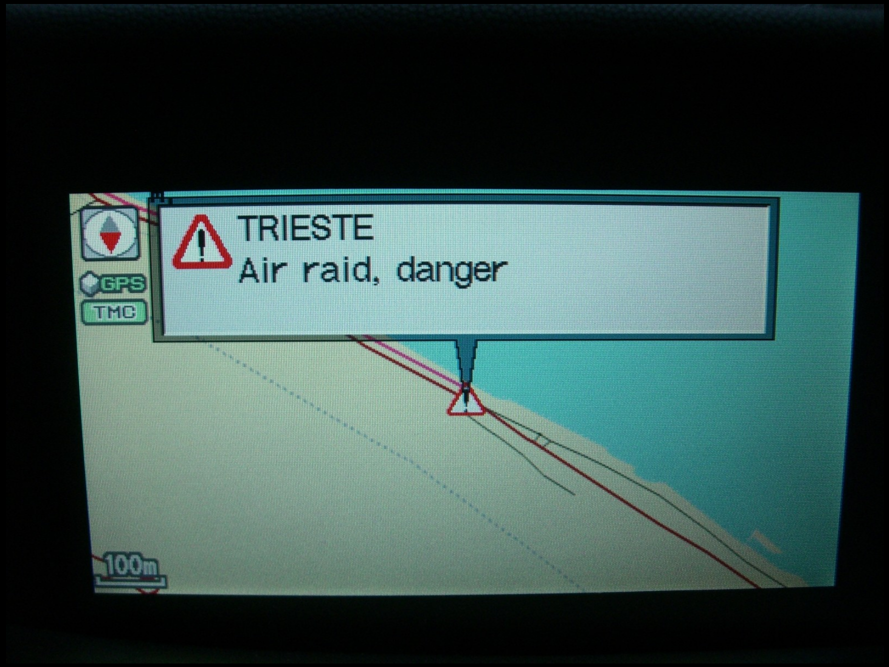


Traffic Event List

1 Air raid, danger 24km
SS14, MESTRE/INN, SS11 PADA

2 Air raid, danger 32km
SS14, PESE DI GROZZANA/CON

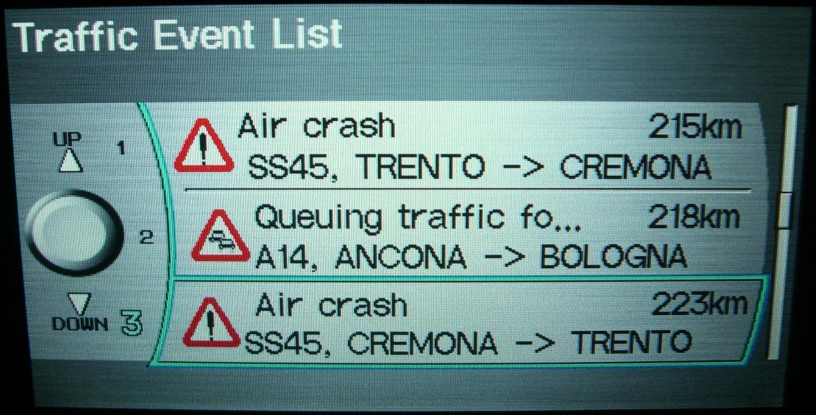
3 Queuing traffic fo... 137km
A13, PADOVA -> BOLOGNA



TRIESTE
Air raid, danger

Code 1481 – Air raid, danger

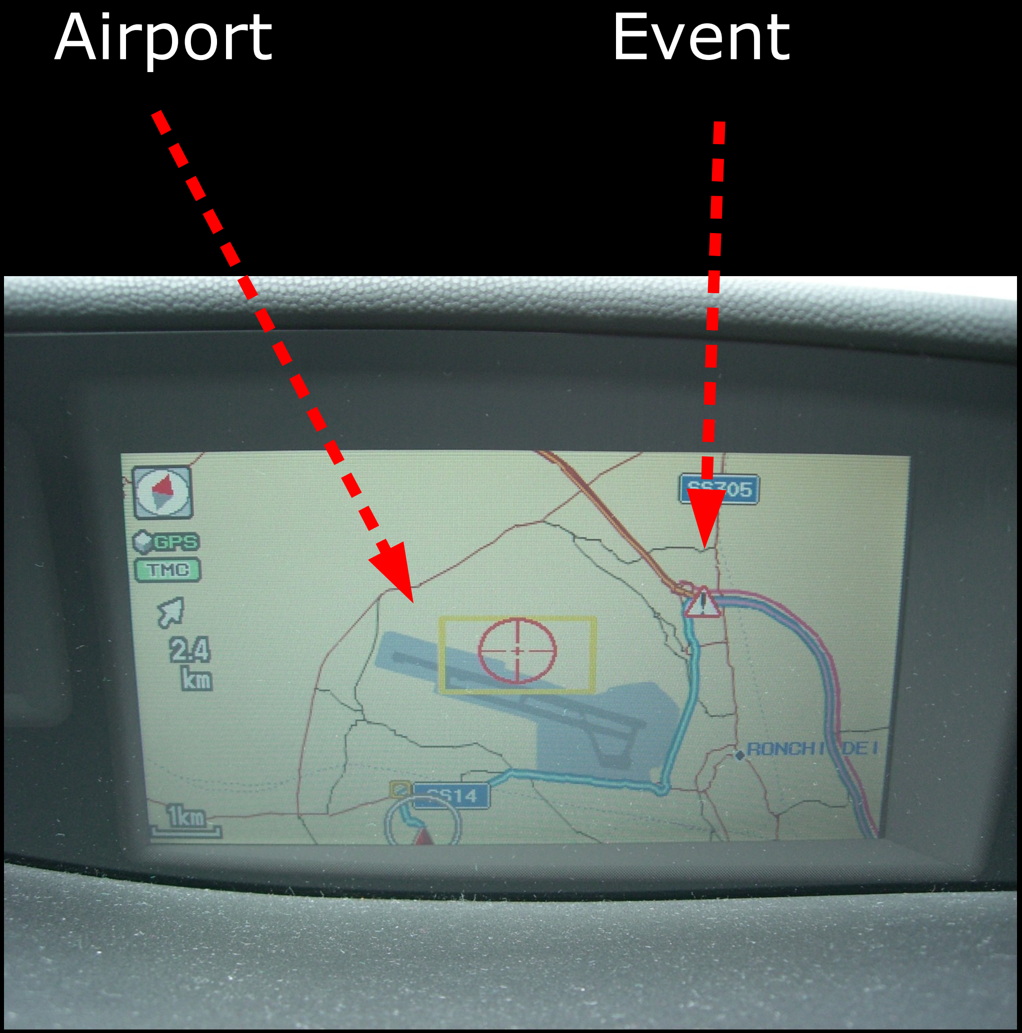
Attack 3: Security Messages



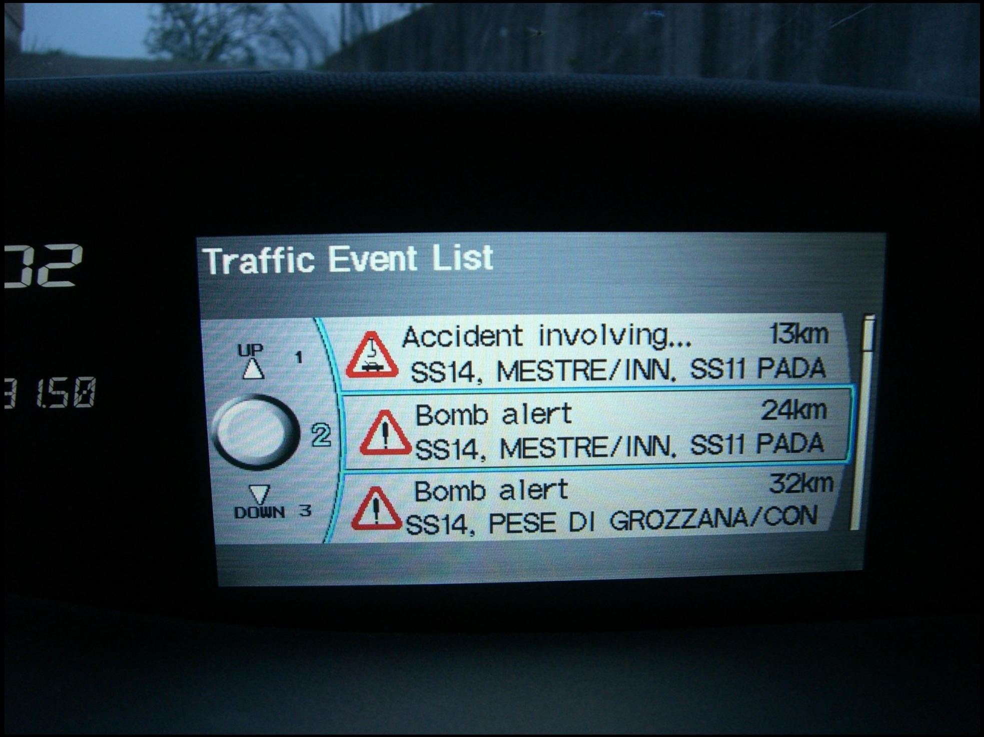
Traffic Event List

1	Air crash SS45, TRENTO -> CREMONA	215km
2	Queuing traffic fo... A14, ANCONA -> BOLOGNA	218km
3	Air crash SS45, CREMONA -> TRENTO	223km

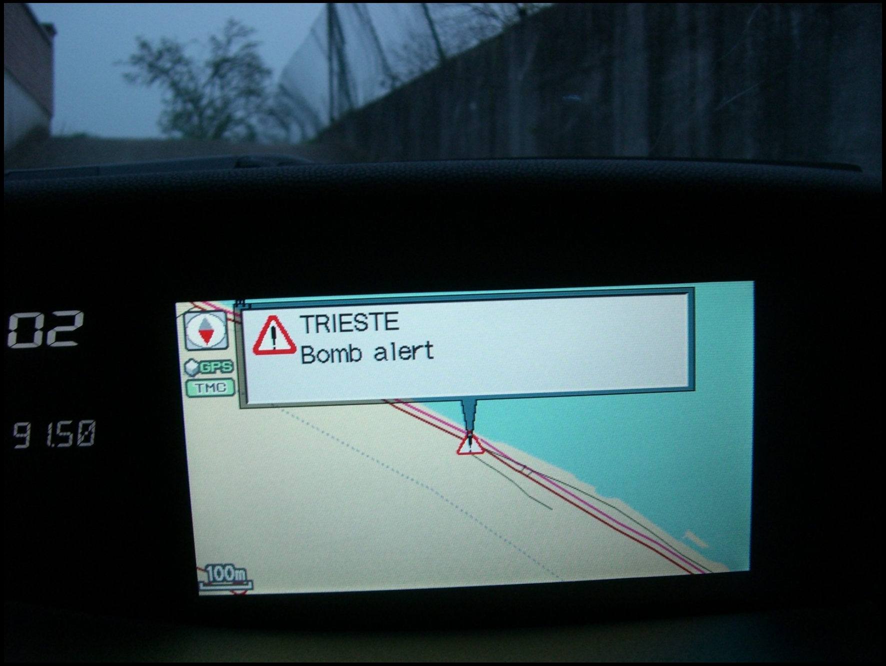
Code 978 – Air crash



Attack 3: Security Messages


Traffic Event List

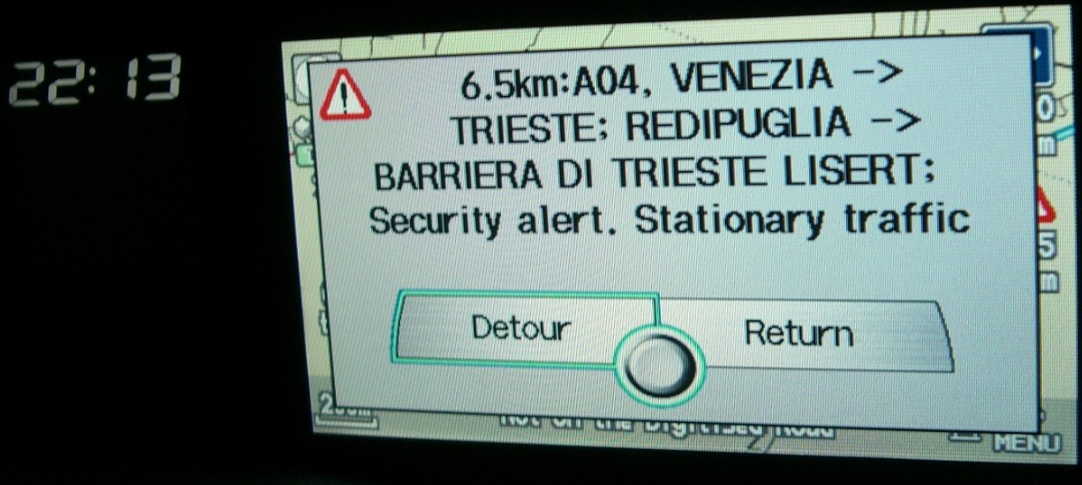
Accident involving... 13km
SS14, MESTRE/INN, SS11 PADA
Bomb alert 24km
SS14, MESTRE/INN, SS11 PADA
Bomb alert 32km
SS14, PESE DI GROZZANA/CON


TRIESTE
Bomb alert

Code 1516 – Bomb alert

Attack 3: Security Messages

- Security messages can be pop-up, if they affect current route
- Video Clip time!

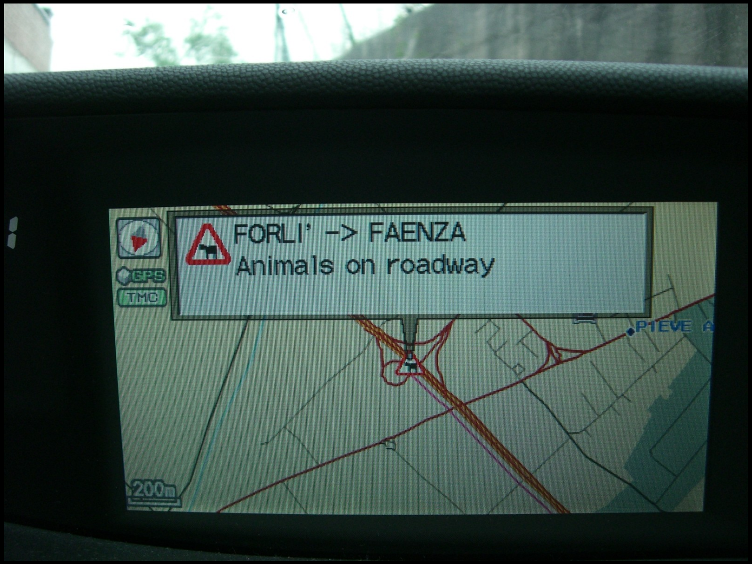

22:13

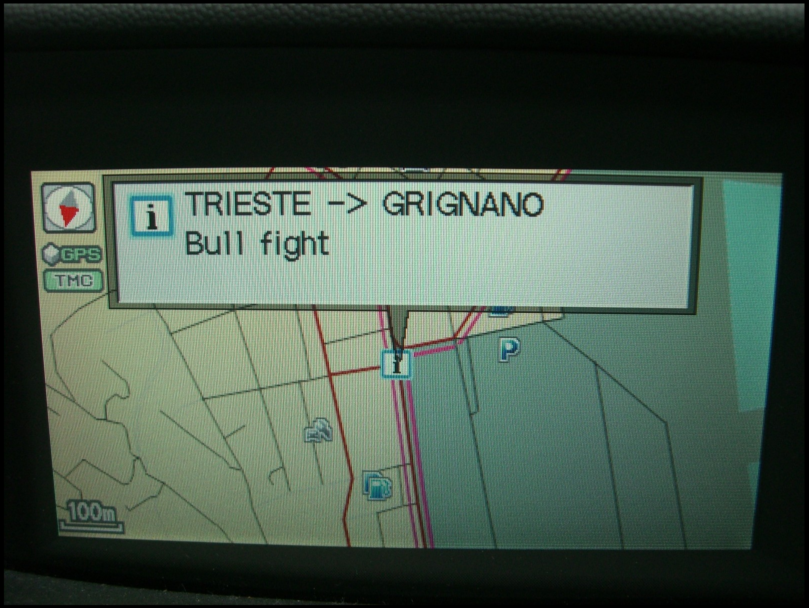
6.5km:A04, VENEZIA -> TRIESTE: REDIPUGLIA -> BARRIERA DI TRIESTE LISERT: Security alert. Stationary traffic

Code 1571

Security alert. Stationary traffic

Other funny messages


FORLI' -> FAENZA
Animals on roadway


TRIESTE -> GRIGNANO
Bull fight

Code 1456 – Bull Fight (you never know...)

Code 1560 – Delays due to parade

...and many more...(no you can't have a pony)

Implementation Issues

- On our Honda integrated SatNav we've seen that:
  - The PI is not associated to the frequency, any PI can be used on any frequency for hijacking
  - Total cancellation (Event: 2047, Location: 65535) is not honoured
  - Broadcast message (Location: 65535) is not honoured
  - Diversion bit is ignored for some categories and always assumed = 1
- We expect other SatNav systems to have similar or even more interesting issues

RDS-TMC Encryption

- TMC supports a very lightweight encryption for commercial services
- Described in ISO 14819-6
- It's used for signal discrimination rather than authentication
- Only the Location Code is encrypted
- It involves bitwise operations against a key
- The key can be trivially broken by sampling some data
- Terminals that support encryption are also expected to accept un-encrypted data, so injection is still possible

Security Considerations

- RDS-TMC can be trivially injected
- Drivers don't tend to have any security awareness towards their SatNav, social engineering, forced detours and panic attacks are possible
- We don't think it's "*The End Of The World As We Know It*" but these systems should be authenticated considering their increased usage and expansion
- These technologies have a very long life span and "patching" is not easy
- We hope to increase awareness about these kind of problems

TMC Forum Official Response

- "Hacking TMC – Unsuccessfully" (...not really)

- "The first and overriding statement that should be made is that transmissions of this type are directly analogous to "pirate" radio broadcasts and certainly will, in the case of Europe and the U.S., contravene each countries respective broadcasting legislation and laws."
- "...there is a chance that the false message could be decoded, but a degree of knowledge would have to be gained on parameters of the message being coded..."
- "...the random use of any location code would result in a randomly located event... Also random choices of Event codes may not cause the terminal to react..."

TMC Forum Official Response

- "In the case of (b), i.e. if the transmission is on a different frequency, it is very unlikely that a terminal will even tune to the false service. This is because this frequency will not be either in the main AF list or the secondary AF list broadcast in any of the tuning variants of the TMC data."
- "Service Providers and Broadcasters, *I am sure*, have many protection mechanisms and processes in place to prevent any illegitimate access to their services within their infrastructure."

- read the full response at:

  http://www.tmcforum.com/en/about_tmc/tmc_news/hacking_tmc_-_unsuccessfully.htm

  our reply:

  http://dev.inversepath.com/rds/our_response_to_TMC_Forum_statement.txt

The Future

- TMC is also supported over DAB and satellite radio, it's harder to inject compared to FM but still possible
- TPEG (Transport Protocol Experts Group) is the new standard designed for replacing TMC. It supports encryption but it's still optional. (*http://tpeg.org*)
- GST (Global System for Telematics) is an impressive new architecture for delivering a number of services. It's backed up by many manufacturers and it will support PKI for billing and transport purposes. Adoption is many years away from now. (*http://gstforum.org*)

Similar Systems

- Microsoft DirectBand (*http://www.directband.com*), used for MSN Direct, is another FM subcarrier channel for data transmission
- It has a larger bandwidth (15 times that of RDS) and full encryption
- Other than special wristwatches it's also been used on SatNav systems for traffic information (*http://garmin.msndirect.com*)
- Closed standard, not available in Europe, looks very promising...we'd love to play with that too ;)

The End

Thanks for listening! - Questions?

(shameless plug)

http://www.inversepath.com

Traffic Sign Images used with permission from
http://gettingaroundgermany.home.att.net
Thanks to Brian Purcell