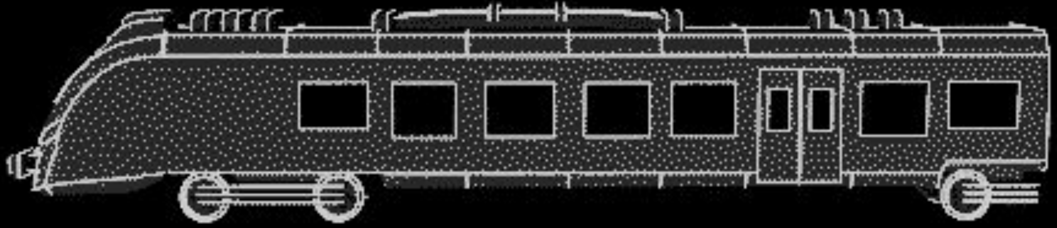
Breaking “DRM” in Polish Trains



37C3

whois

Redford, q3k, MrTick

Loosely affiliated with Dragon Sector.

Background

2016: Koleje Dolnośląskie buys eleven Impuls trains (45WE) from Newag.

Q2/3 2021: Koleje Dolnośląskie runs a tender to service trains. SPS Mieczkowski is selected.

Q1 2022: First 45WE arrives at SPS.

Author: Jon Worth, CC-BY

SPS/KD contract (public coverage)

2022/04: 45WE-024 serviced at SPS, but doesn’t start.

2022/06: Mysterious failures of other Impulses (Polregio).

2022/07: Newag blames SPS for messing with the ‘security system’ of the trains.

2022/08: SPS starts returning serviced trains.

SPS/KD contract

2022/04: 45WE-024 serviced at SPS, but doesn’t start.

2022/05: SPS gets in touch with us.

2022/06: Mysterious failures of other Impulses (Polregio).

2022/07: Newag blames SPS for messing with the ‘security system’ of the trains.

2022/08: We managed to unlock the first 45WE.

2022/08: SPS starts returning serviced trains.

2022/10: First report to authorities (UOKiK) by SPS.

How to unlock a train

What is a locked train?

HMI reports '*ready to move*'.

Throttle pushed forward to set speed.

Electro-pneumatic brakes release.

Train does not move.

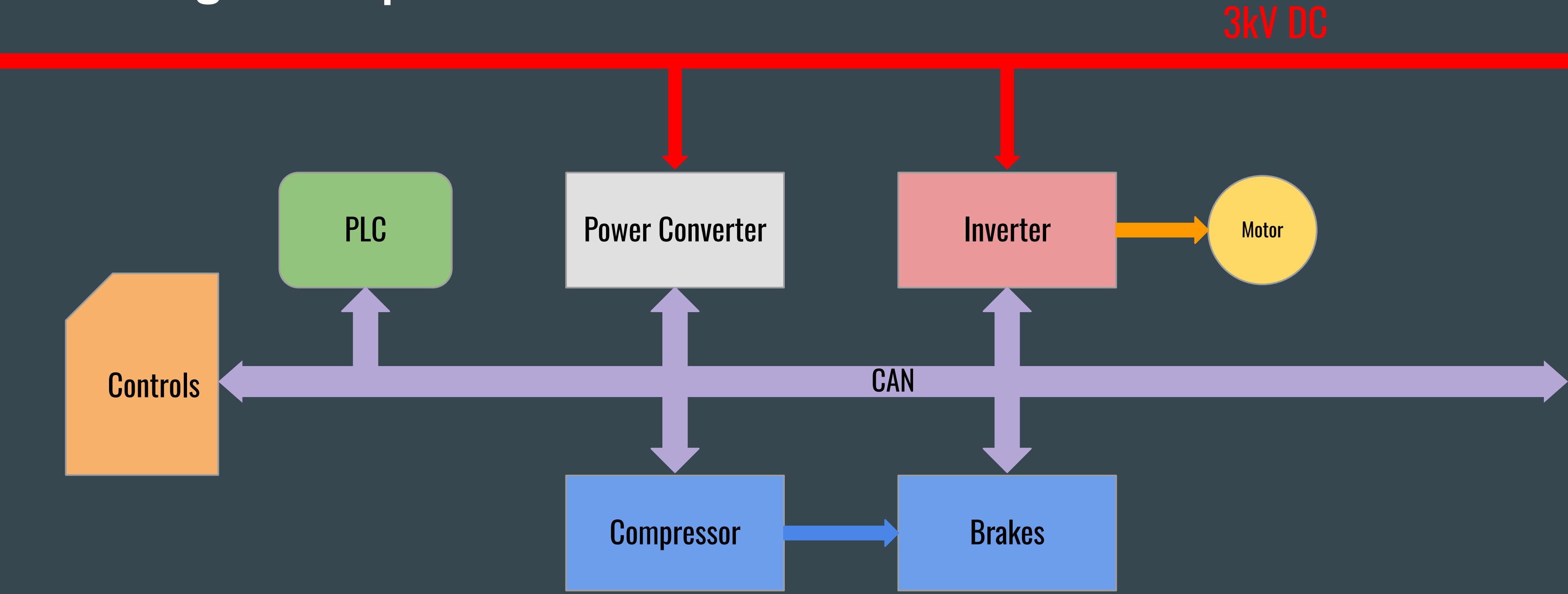
What do we have?

1x train (locked)

2x Train Controller with SDK

1x train service docs

Finding the culprit



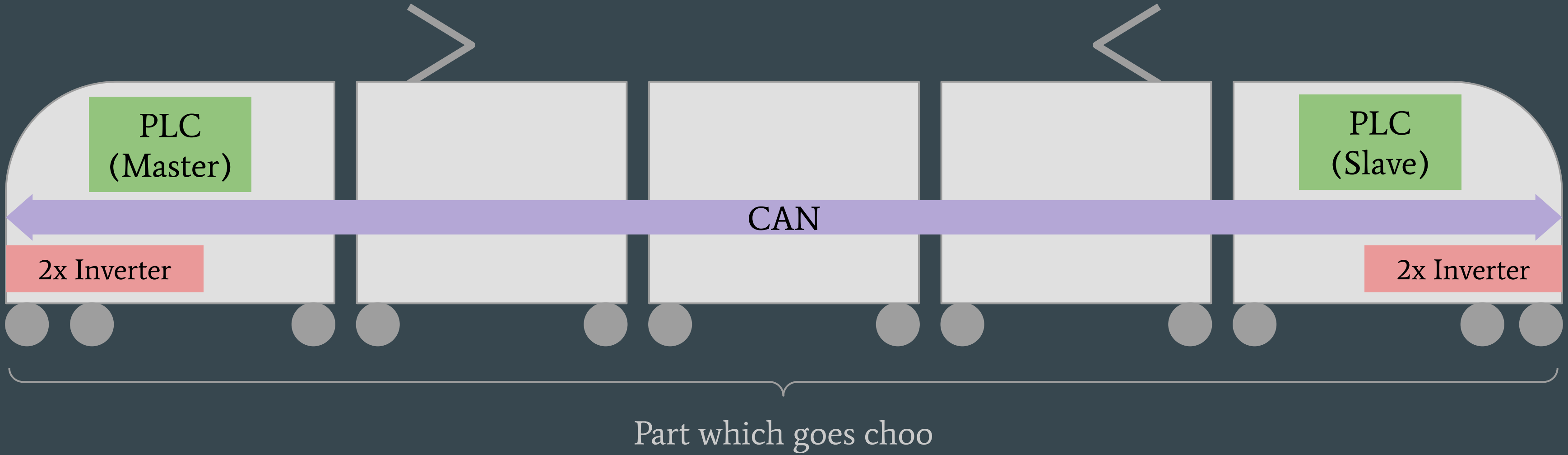
CAN all the way down

All communication performed over CAN/CANopen.

What’s the difference between a working and locked up train?

Train	**CANopen traffic**
45WE-022 *Working*	ID: `0x0212` Payload: `01` `1f` **`0b 0b 0b 0b`** `52 00` Sent by CPU on CAN1
45WE-023 *Locked up*	ID: `0x0212` Payload: `01` `0f` **`00 00 00 00`** `52 00` Sent by CPU on CAN1

Simplified scope



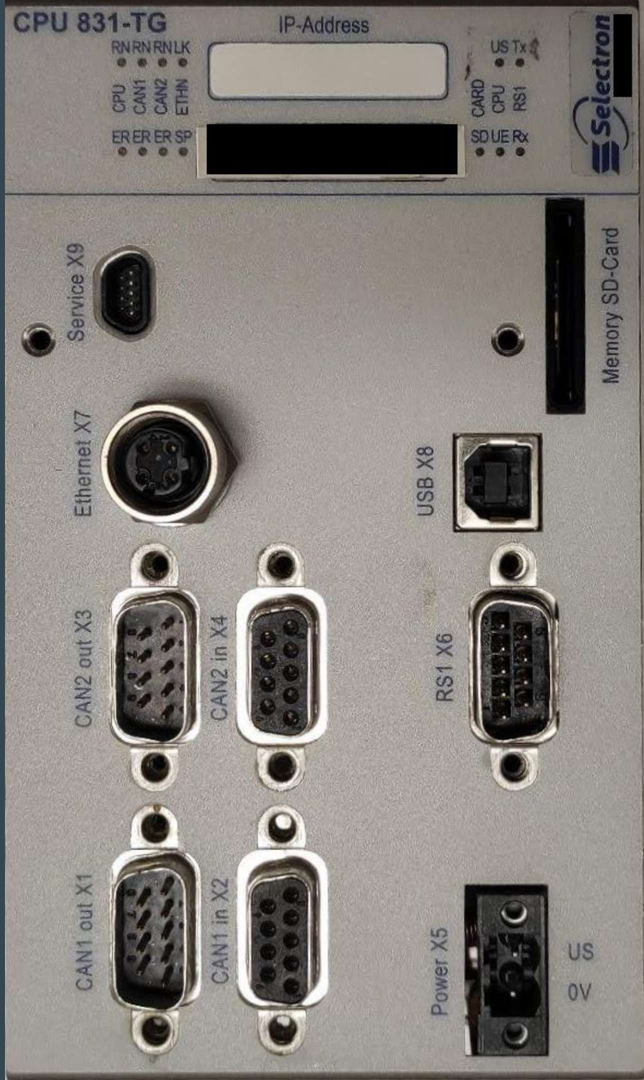
PLCs?

Part of the Train Control and Management System.

Selectron **MAS83x TCMS.**

CPU831-TG PLC.

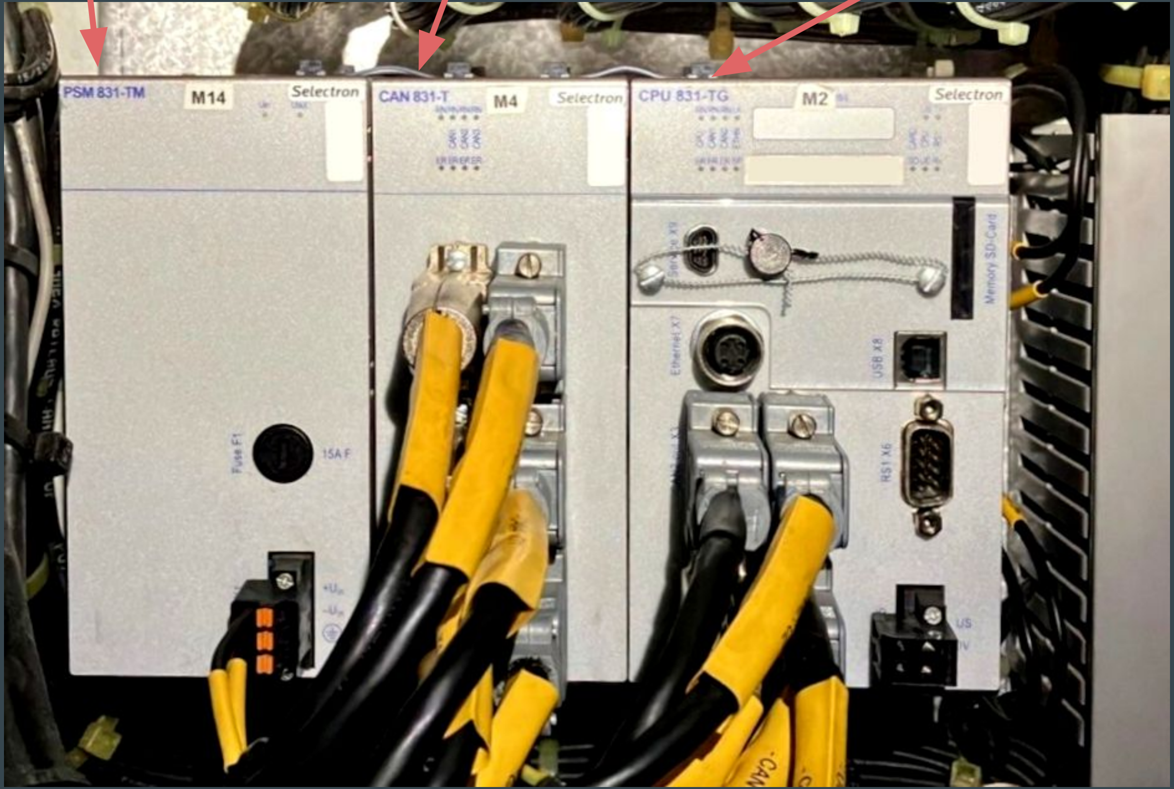
Programmed via the IEC 61131-3 standard.



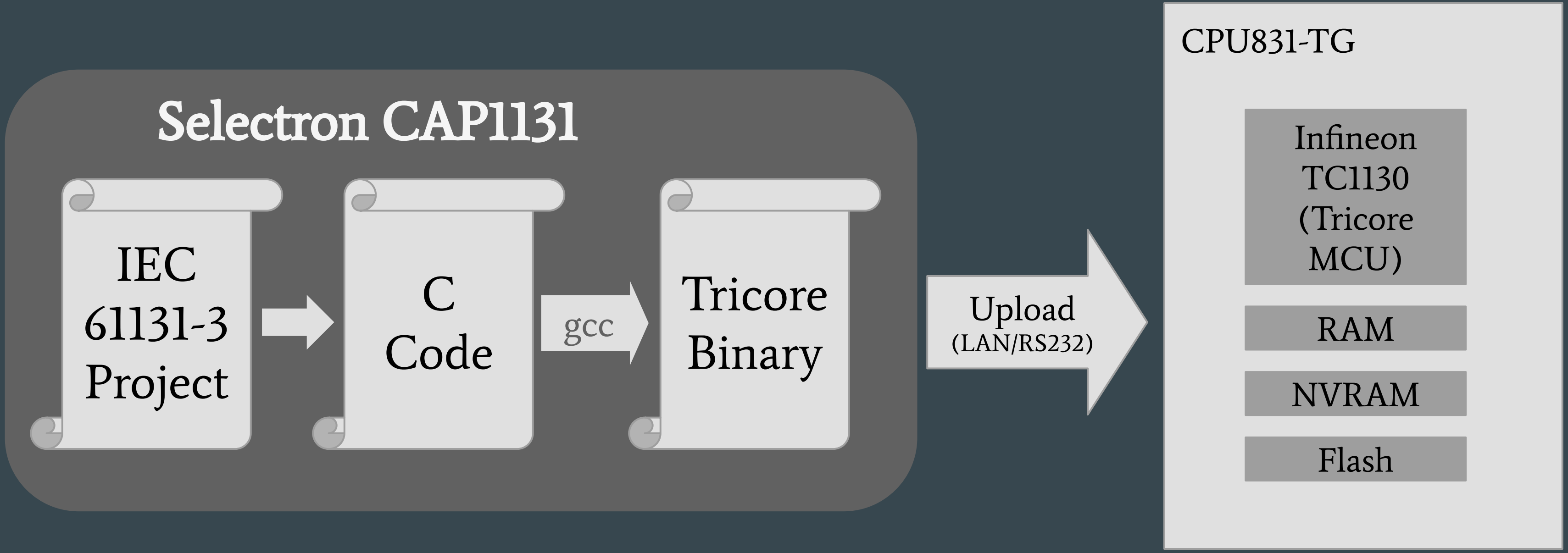
PSM831-TM (power)

CAN831-T (more CAN)

CPU831-TG (PLC)



CPU831-TG programming



CPU831-TG program acquisition

User Program not protected from readout.

... but also no ‘download from PLC’ button in CAP1131/WDLD1131.

Ended up using CAP1131 debugger to read memory.

SysCom protocol

Simple UDP-based protocol.

We reverse-engineered based on traffic dumps and a Windows DLL.

Flow:

Login (with a password!) (always user_0 and sele0).

Set a variable monitoring range by address.

Query the range’s state.

dumper.py

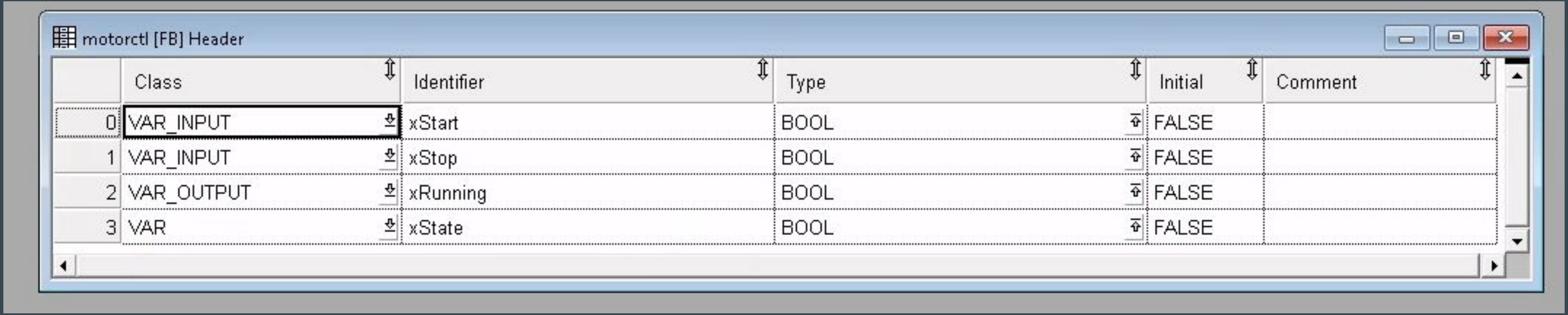
```
1. Hello PLC I am user_0
2. X = program_start
3. Monitor Range X to X+1000
4. Query Range
5. X += 1000
6. GOTO 3
```

```
$ strings flash.bin | tail -n 18 | head -n 10
[ProjInfo]
Name=c:\update\newag_45we_kds
Resource=Resource
compiled=28.10.2021 11:33:37
Version=326
Comment=
[ENVIRONMENT]
CAP1131=5.9.0110.3991
OS=MOS83x_831 V703.C96C
CPU=CPU831
```

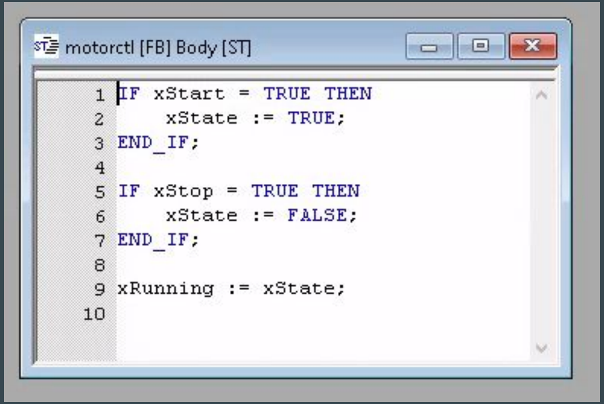
A quick IEC 61131-3 detour

Now is a good time to sleep.

CAP1131: A simple function block in ST


motorctl [FB] Header

	Class	Identifier	Type	Initial	Comment
0	VAR_INPUT	xStart	BOOL	FALSE	
1	VAR_INPUT	xStop	BOOL	FALSE	
2	VAR_OUTPUT	xRunning	BOOL	FALSE	
3	VAR	xState	BOOL	FALSE	


motorctl [FB] Body [ST]

```
IF xStart = TRUE THEN
    xState := TRUE;
END_IF;

IF xStop = TRUE THEN
    xState := FALSE;
END_IF;

xRunning := xState;
```

CAP1131: Codegen of a FBD in ST

```
typedef struct {
  SC_UDINT                            STRUCT__SIZE;
  SC_BOOL                             XSTART;
  SC_BOOL                             XSTOP;
  SC_BOOL                             XRUNNING;
  SC_BOOL                             XSTATE;
} MOTORCTL_TYP;
```

Variables from the FB.

CAP1131: Codegen of a FBD in ST

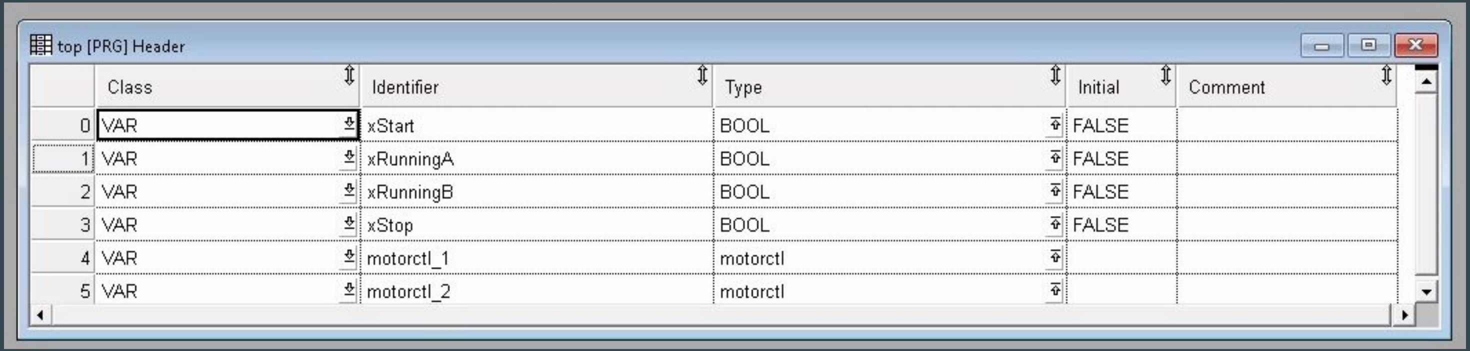
```
void SC_BODY__MOTORCTL_00012(MOTORCTL_TYP* ip) {
  FB_ENTRY(12, ip);

  if(((1 & ip->XSTART) == SC_C_BOOL(1))) {
     ip->XSTATE = SC_C_BOOL(1);
  }
  if(((1 & ip->XSTOP) == SC_C_BOOL(1))) {
     ip->XSTATE = SC_C_BOOL(0);
  }
  ip->XRUNNING = ip->XSTATE;

  FB_EXIT(12, ip);
}
```

Our ST code.

CAP1131: Codegen of a FBD in ST

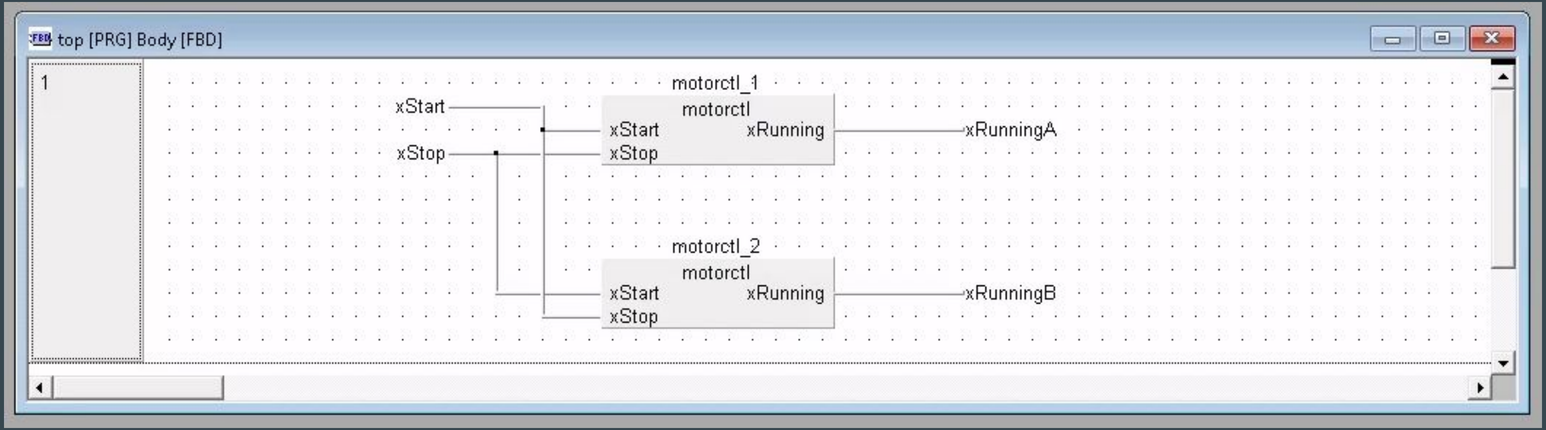
```
void SC_INIT__MOTORCTL_00016(MOTORCTL_TYP* ip, int iRetain) {
  ip->STRUCT__SIZE = OFFSET(MOTORCTL_TYP, XSTATE) + sizeof(ip->XSTATE);
  ip->XSTART = SC_C_BOOL(0);
  ip->XSTOP = SC_C_BOOL(0);
  ip->XRUNNING = SC_C_BOOL(0);
  ip->XSTATE = SC_C_BOOL(0);
  iRetain = iRetain;
}
```

CAP1131: A simple program in Functional Block Diagram



top [PRG] Header

	Class	Identifier	Type	Initial	Comment
0	VAR	xStart	BOOL	FALSE	
1	VAR	xRunningA	BOOL	FALSE	
2	VAR	xRunningB	BOOL	FALSE	
3	VAR	xStop	BOOL	FALSE	
4	VAR	motorctl_1	motorctl		
5	VAR	motorctl_2	motorctl		


top [PRG] Body [FBD]

CAP1131: Codegen of a program in FBD

```
typedef struct {
  SC_UDINT                   STRUCT__SIZE;
  SC_BOOL                    XRUNNINGA;
  SC_BOOL                    XRUNNINGB;
  SC_BOOL                    XSTOP;
  SC__INSTC                  MOTORCTL_1;
  SC__INSTC                  MOTORCTL_2;
} TOP_TYP;
```

Variables in “top”.

Instances in “top”.

CAP1131: Codegen of a program in FBD

SC_INSTC..?

```
typedef struct {
  SC_UDINT                        STRUCT__SIZE;
  SC_BOOL                         XRUNNINGA;
  SC_BOOL                         XRUNNINGB;
  SC_BOOL                         XSTOP;
  SC__INSTC                       MOTORCTL_1;
  SC__INSTC                       MOTORCTL_2;
} TOP_TYP;
```

Variables in “top”.

Instances in “top”.

CAP1131: Codegen of a program in FBD

```
void SC_INIT__TOP_00013(TOP_TYP* ip, int iRetain) {
  ip->STRUCT__SIZE = OFFSET(TOP_TYP, TON_1) + sizeof(ip->TON_1);
  ip->XRUNNINGA = SC_C_BOOL(0);
  ip->XRUNNINGB = SC_C_BOOL(0);
  ip->XSTOP = SC_C_BOOL(0);
  iRetain = iRetain;
}
```

Constructor… but it doesn’t initialize instances.

CAP1131: Codegen of a program in FBD

```c
void SC_BODY__TOP_00011(void) {
  TOP_TYP *ip = (TOP_TYP*)__OD(14);

  PRG_ENTRY(11, ip);
  {
     MOTORCTL_TYP* __lpt0 = (MOTORCTL_TYP*)__OD(ip->MOTORCTL_1);
     MOTORCTL_TYP* __lpt1 = (MOTORCTL_TYP*)__OD(ip->MOTORCTL_2);
     (*__lpt0).XSTART = ip->XSTART;
     (*__lpt0).XSTOP = ip->XSTOP;
     ((__BPFN_MOTORCTL)__OF(12))((__lpt0));
     ip->XRUNNINGA = (*__lpt0).XRUNNING;
     (*__lpt1).XSTART = __OG(XSTOP);
     (*__lpt1).XSTOP = __OG(XSTART);
     ((__BPFN_MOTORCTL)__OF(12))((__lpt1));
     ip->XRUNNINGB = (*__lpt1).XRUNNING;
  }
  PRG_EXIT(11, ip);
}
```

Evaluating the first motorctl...

Evaluating the second motorctl...

CAP1131: Codegen of a program in FBD

```
void SC_BODY__TOP_00011(void) {
  TOP_TYP *ip = (TOP_TYP*)__OD(14);

  PRG_ENTRY(11, ip);
  {
     MOTORCTL_TYP* __lpt0 = (MOTORCTL_TYP*)__OD(ip->MOTORCTL_1);
     MOTORCTL_TYP* __lpt1 = (MOTORCTL_TYP*)__OD(ip->MOTORCTL_2);
     (*__lpt0).XSTART = ip->XSTART;
     (*__lpt0).XSTOP = ip->XSTOP;
     ((__BPFN_MOTORCTL)__OF(12))((__lpt0));
     ip->XRUNNINGA = (*__lpt0).XRUNNING;
     (*__lpt1).XSTART = __OG(XSTOP);
     (*__lpt1).XSTOP = __OG(XSTART);
     ((__BPFN_MOTORCTL)__OF(12))((__lpt1));
     ip->XRUNNINGB = (*__lpt1).XRUNNING;
  }
  PRG_EXIT(11, ip);
}
```

Evaluating the first motorctl...

Evaluating the second motorctl...

__OD?
__OF?

CAP1131: __OD & __OF

`__OD(ix)`: access an object instance by index

`__OF(ix)`: access an evaluation function by index

In practice: #define __O{D,F}(ix) OCO__PT[ix]

Every embedding of an FB/P in another FB/P generates accesses solely over __OD and __OF.

CAP1131: OCO__PT

```c
void* OCO__PT[65536] = {
  /*[ 0]*/ &ProjInfoCodeV23,               //internal use
  /*[ 1]*/ 0,                              //not used
  /*[ 2]*/ 0,                              //not used
  /*[ 3]*/ 0,                              //not used
  /*[ 4]*/ &SC_DATA____GLOBALS_00004,      //<GVL>.<all simple types>/__OG(<glob var name>)
  /*[ 5]*/ SC_INIT____GLOBALS_00005,       //GVL init/__OF(5)
#if defined(__BACK_TRACE__)
  /*[ 6]*/ szObjectStringTable,            //internal use
#else
  /*[ 6]*/ 0,                              //not used
#endif
  /*[ 7]*/ SC_BODY__LOOP_00007,            //<tasks>.loop/__OF(7)
  /*[ 8]*/ 0,                              //not used
  /*[ 9]*/ 0,                              //not used
  /*[10]*/ 0,                              //not used
  /*[11]*/ SC_BODY__TOP_00011,             //PRG body/__OF(11)
  /*[12]*/ SC_BODY__MOTORCTL_00012,        //FB  body/__OF(12)
  /*[13]*/ &SC_DATA__TOP_00013,            //top/__OD(13)
  (...)
}
```

CPU831-TG: First RE

Alright, let’s load our earlier example into Ghidra.

ELF with DWARF - free symbols!

Shouldn’t be that hard, right?

CPU831-TG: First RE in Ghidra (with symbols)

```
void SC_BODY__TOP_00011(void) {
 [...]

 top = OCO__PT.TOP;
 motctl1 = (MOTORCTL_TYP *)(&OCO__PT.field0_0x0)[top->MOTORCTL_1];
 motctl2 = (MOTORCTL_TYP *)(&OCO__PT.field0_0x0)[top->MOTORCTL_2];
 motctl1->XSTART = top->XSTART;
 motctl1->XSTOP = top->XSTOP;
 (*(code *)((uint)OCO__PT.TOP_EVAL & 0xfffffffe))(motctl1);
 motctl2->XSTART = top->XSTOP;
 motctl2->XSTOP = top->XSTART;
 (*(code *)((uint)OCO__PT.TOP_EVAL & 0xfffffffe))(motctl2);
 top->XRUNNINGA = motctl1->XRUNNING;
 top->XRUNNINGB = motctl2->XRUNNING;
```

CPU831-TG: Original source for comparison

```
void SC_BODY__TOP_00011(void) {
  TOP_TYP *ip = (TOP_TYP*)__OD(14);
  PRG_ENTRY(11, ip);
  {
     MOTORCTL_TYP* __lpt0 = (MOTORCTL_TYP*)__OD(ip->MOTORCTL_1);
     MOTORCTL_TYP* __lpt1 = (MOTORCTL_TYP*)__OD(ip->MOTORCTL_2);
     (*__lpt0).XSTART = ip->XSTART;
     (*__lpt0).XSTOP = ip->XSTOP;
     ((__BPFN_MOTORCTL)__OF(12))((__lpt0));
     ip->XRUNNINGA = (*__lpt0).XRUNNING;
     (*__lpt1).XSTART = __OG(XSTOP);
     (*__lpt1).XSTOP = __OG(XSTART);
     ((__BPFN_MOTORCTL)__OF(12))((__lpt1));
     ip->XRUNNINGB = (*__lpt1).XRUNNING;
  }
  PRG_EXIT(11, ip);
}
```

CAP1131 RE pain: OCO__PT

Global array of all objects.

Numerically indexed as an array, but values are heterogenous.

Numerical indexes are often dynamic.

```
void *foo = OCO__PT[this->field_0x20];
```

So what’s the type of foo?

Tricore RE pain: Calling convention

Separate data/address registers!

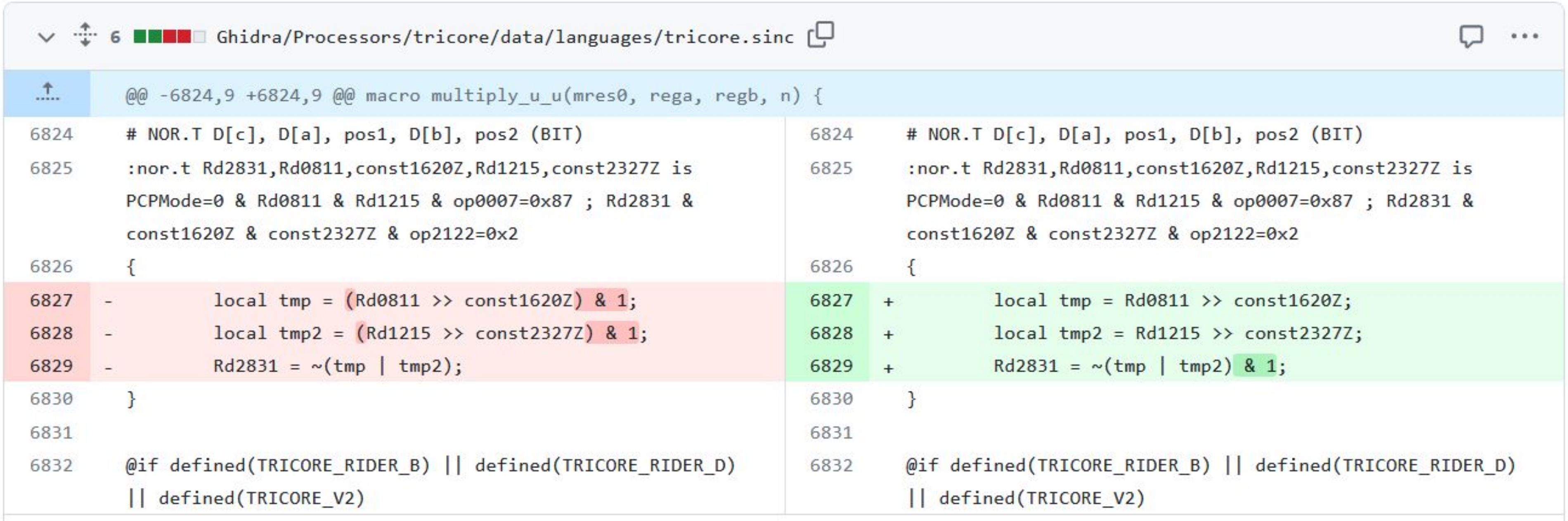
Calling convention? Arguments via registers (d4, d5... and a4, a5...)

```
int foo(int m, int *n, int *o, int p);
//        d4 ^    a4 ^    a5 ^   d5 ^
```

Ghidra doesn’t implement this convention correctly.

Tricore RE pain: Ghidra bugs

A few bugs in Tricore semantics!



```
6 Ghidra/Processors/tricore/data/languages/tricore.sinc
```

```
@@ -6824,9 +6824,9 @@ macro multiply_u_u(mres0, rega, regb, n) {
```

```
  # NOR.T D[c], D[a], pos1, D[b], pos2 (BIT)
  :nor.t Rd2831,Rd0811,const1620Z,Rd1215,const2327Z is PCPMode=0 & Rd0811 & Rd1215 & op0007=0x87 ; Rd2831 & const1620Z & const2327Z & op2122=0x2
  {
-         local tmp = (Rd0811 >> const1620Z) & 1;
-         local tmp2 = (Rd1215 >> const2327Z) & 1;
-         Rd2831 = ~(tmp | tmp2);
+         local tmp = Rd0811 >> const1620Z;
+         local tmp2 = Rd1215 >> const2327Z;
+         Rd2831 = ~(tmp | tmp2) & 1;
  }

  @if defined(TRICORE_RIDER_B) || defined(TRICORE_RIDER_D) || defined(TRICORE_V2)
```

CAP1131 RE: Finding the lockup

Too much code, what to do?

- Scripting
- Dynamic analysis
- Diffs between trains
- Matching CAN traffic to code
- Get motivated by the deadline

CAP1131 RE: Finding the lockup

Diffing exposed a few interesting places in the code!

Some checks, then NVRAM modification. RAM dump shows values around 1 mln.

NVRAM bits set to 0 only in locked-up trains!

Force to 1 on the test controller...

2
2

Train unlocked!

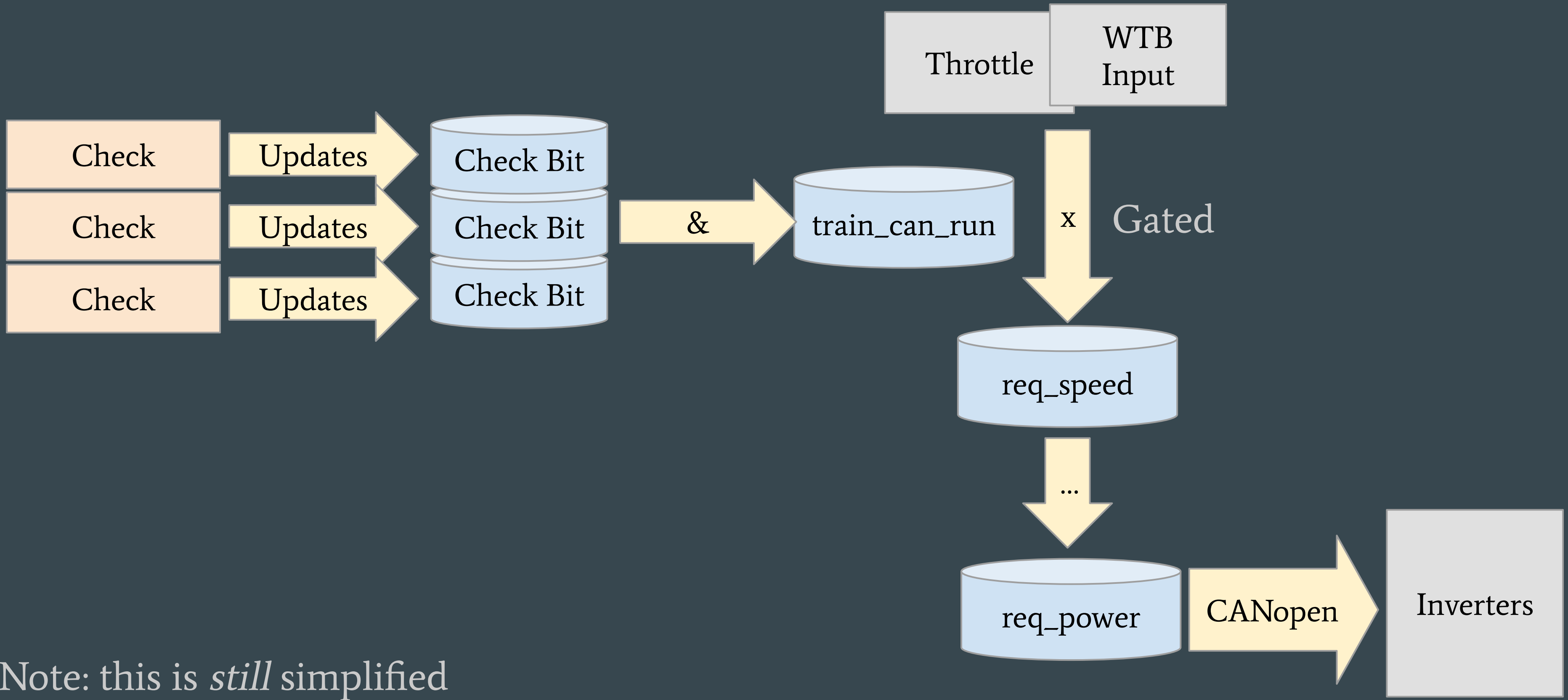
Before we ship this, we have to analyze things further to understand what’s actually going on...

What lock mechanisms are there?

Lockup mechanisms: Force power to 0

Train	**CANopen traffic**
45WE-022 *Working*	ID: `0x0212` Payload: `01 1f` **`0b 0b 0b 0b`** `52 00` Sent by CPU on CAN1
45WE-023 *Locked up*	ID: `0x0212` Payload: `01 0f` **`00 00 00 00`** `52 00` Sent by CPU on CAN1

Lockup mechanisms: Force power to 0



Note: this is *still* simplified

Lockup mechanisms: ‘Emergency stop’ signal

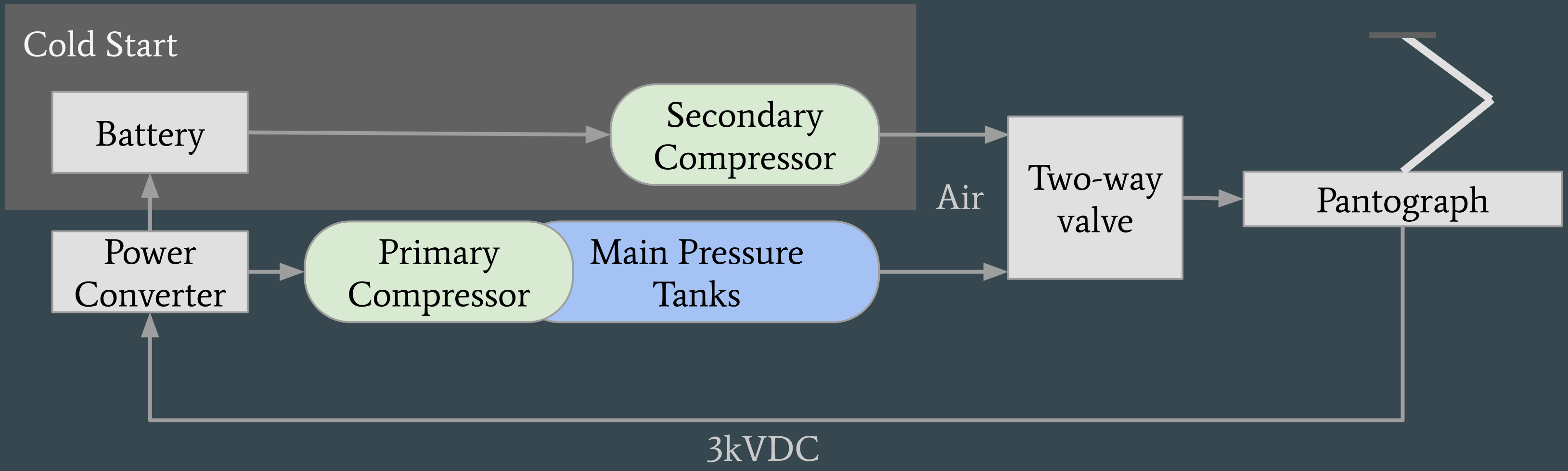
Train	**CANopen traffic**
45WE-022 *Working*	ID: `0x0212` Payload: `01 1f` **`0b 0b 0b 0b`** `52 00` Sent by CPU on CAN1
45WE-023 *Locked up*	ID: `0x0212` Payload: `01 0f` **`00 00 00 00`** `52 00` Sent by CPU on CAN1

Lockup mechanisms: ‘Emergency stop’ signal

```
G_221_SNEAKY_CHECKS(sneaky_obj);
NVRAM_inverter_reserved4 = sneaky_obj->is_train_ready_to_start;

[...]

car_A_inv_output.flags = pack_bits_lsb_first(invD_on, invC_on, invB_on, invA_on,
                                              NVRAM_inverter_reserved4, 0, 0, 0);
```

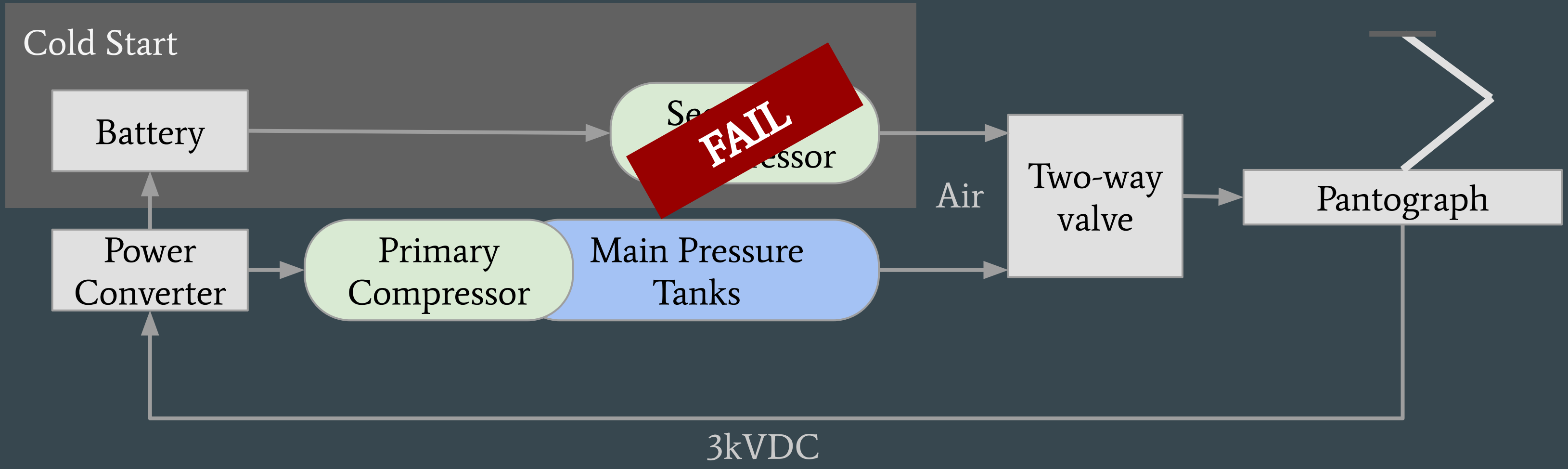
Lockup mechanisms: ‘Compressor failure’

Secondary kind. Found only in a single Polregio train.



Lockup mechanisms: ‘Compressor failure’

Secondary kind. Found only in a single Polregio train.



What triggers them?

Lockup triggers: Lack of movement

The most popular mechanism.

10 days without reaching 60km/h for 3 minutes (and some more checks).

Lockup triggers: Lack of movement

The most popular mechanism.

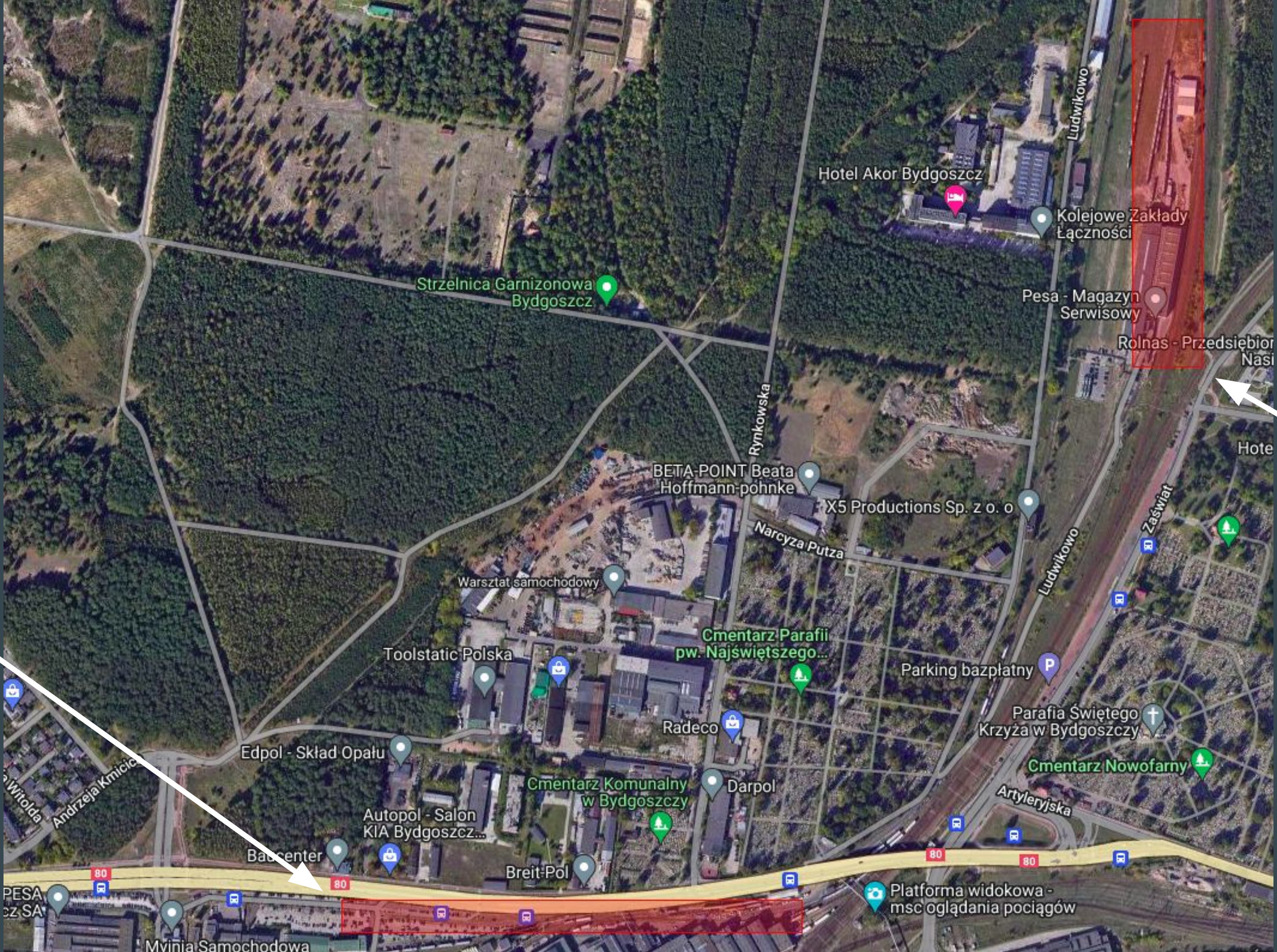
10 days without reaching 60km/h for 3 minutes (and some more checks).

False positives: needs fixes!

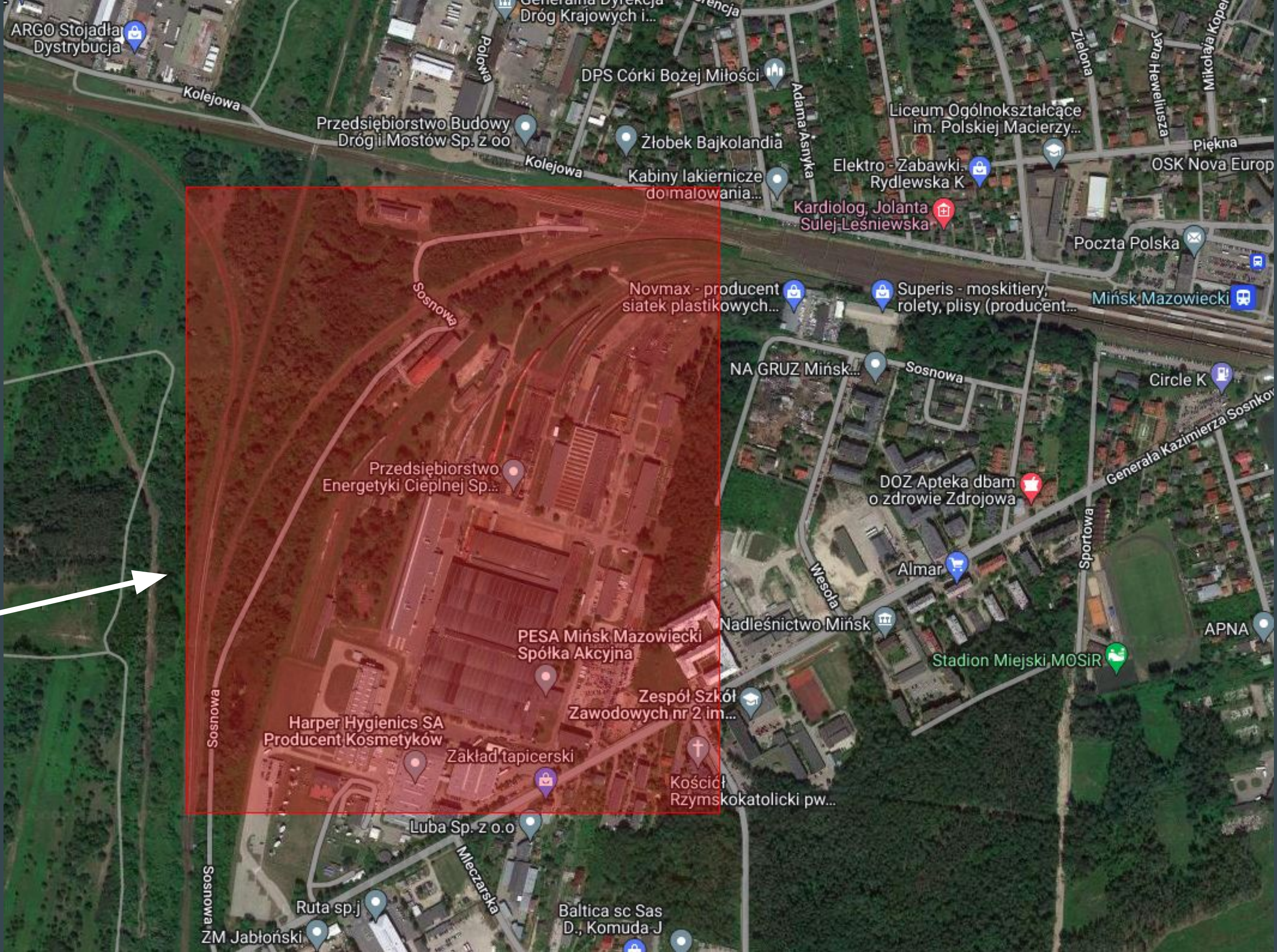
Extended to 20/21 days, added geofencing.

Lockup triggers: Lack of movement + geofencing

```
check1 = 53.13845 < var1 && var1 < 53.13882 && 17.99011 < var2 && var2 < 17.99837;
check2 = 53.14453 < var1 && var1 < 53.14828 && 18.00428 < var2 && var2 < 18.00555;
check3 = 52.17048 < var1 && var1 < 52.17736 && 21.53480 < var2 && var2 < 21.54437;
check4 = 49.60336 < var1 && var1 < 49.60686 && 20.70073 < var2 && var2 < 20.70840
         && (var3 & 1);
check5 = 53.10244 < var1 && var1 < 53.10406 && 18.07817 < var2 && var2 < 18.08243;
check6 = 50.12608 < var1 && var1 < 50.12830 && 19.38411 < var2 && var2 < 19.38872;
check7 = 52.77292 < var1 && var1 < 52.77551 && 18.22117 < var2 && var2 < 18.22724;
```

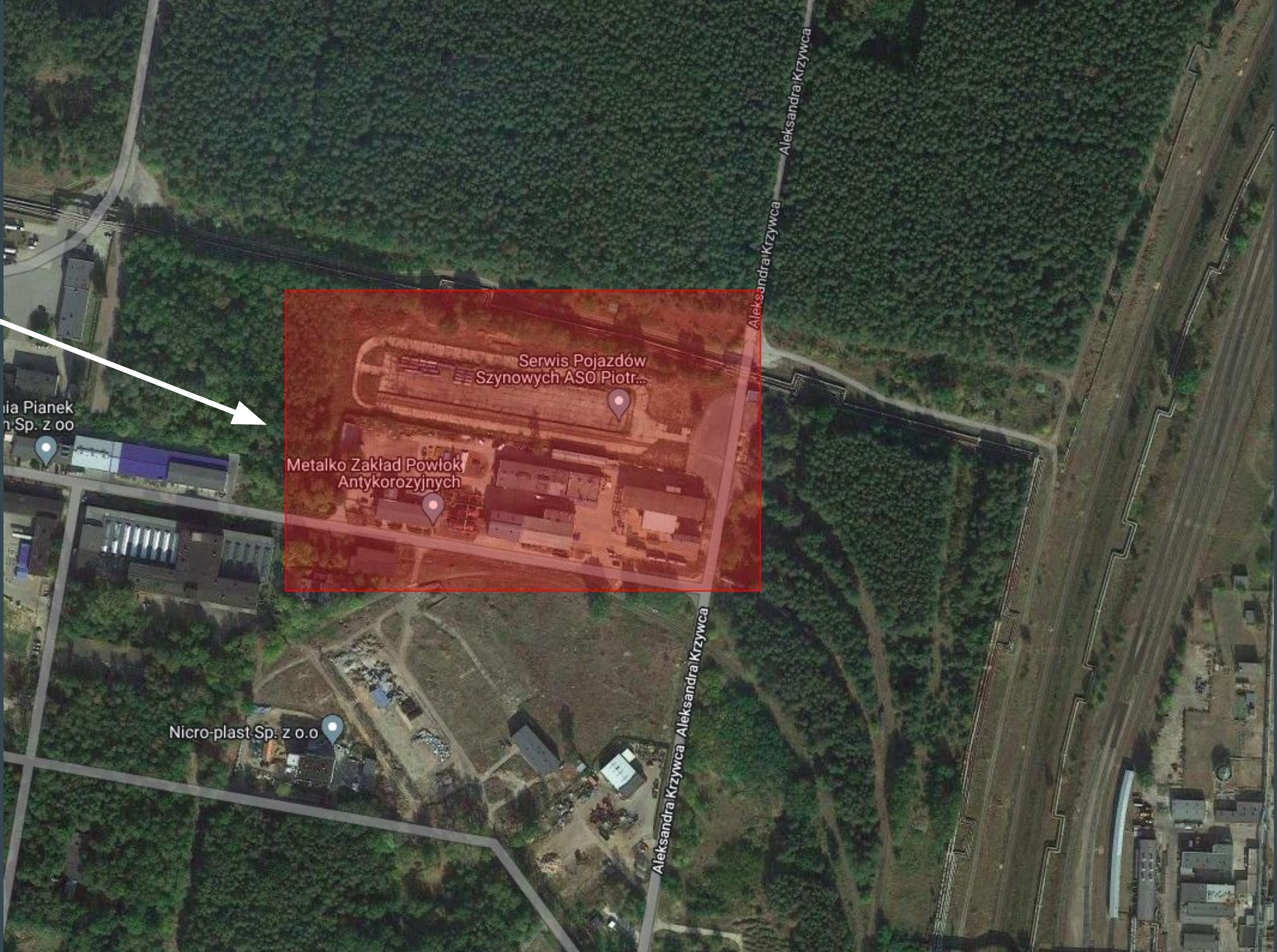


check1
check2

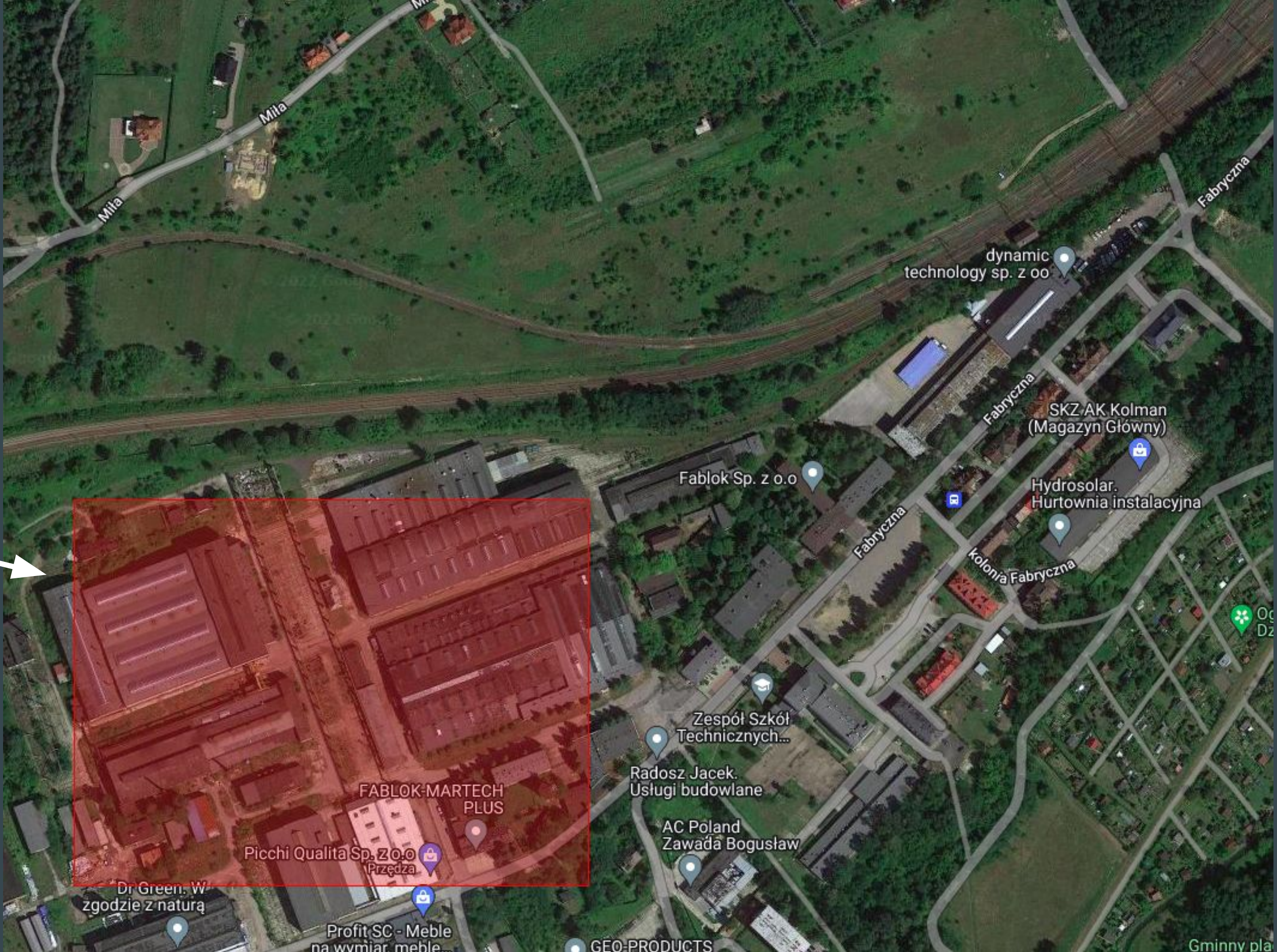

ARGO Stojadła Dystrybucja
Generalna Dyrekcja Dróg Krajowych i...
Polowa
Kolejowa
DPS Córki Bożej Miłości
Przedsiębiorstwo Budowy Dróg i Mostów Sp. z oo
Żłobek Bajkolandia
Adama Asnyka
Zielona
Jana Heweliusza
Mikołaja Kopernika
Liceum Ogólnokształcące im. Polskiej Macierzy...
Piękna
OSK Nova Europ
Elektro - Zabawki. Rydlewska K
Kabiny lakiernicze do malowania...
Kardiolog, Jolanta Sulej-Leśniewska
Poczta Polska
Sosnowa
Novmax - producent siatek plastikowych...
Superis - moskitiery, rolety, plisy (producent...
Mińsk Mazowiecki
NA GRUZ Mińsk...
Circle K
Generała Kazimierza Sosnkowskiego
Przedsiębiorstwo Energetyki Cieplnej Sp...
DOZ Apteka dbam o zdrowie Zdrojowa
Almar
Wesoła
Sportowa
PESA Mińsk Mazowiecki Spółka Akcyjna
Nadleśnictwo Mińsk
APNA
Stadion Miejski MOSiR
Zespół Szkół Zawodowych nr 2 im...
Harper Hygienics SA Producent Kosmetyków
Zakład tapicerski
Kościół Rzymskokatolicki pw...
Luba Sp. z o.o
Mleczarska
Ruta sp.j
Baltica sc Sas D., Komuda J
ZM Jabłoński

check3

check5

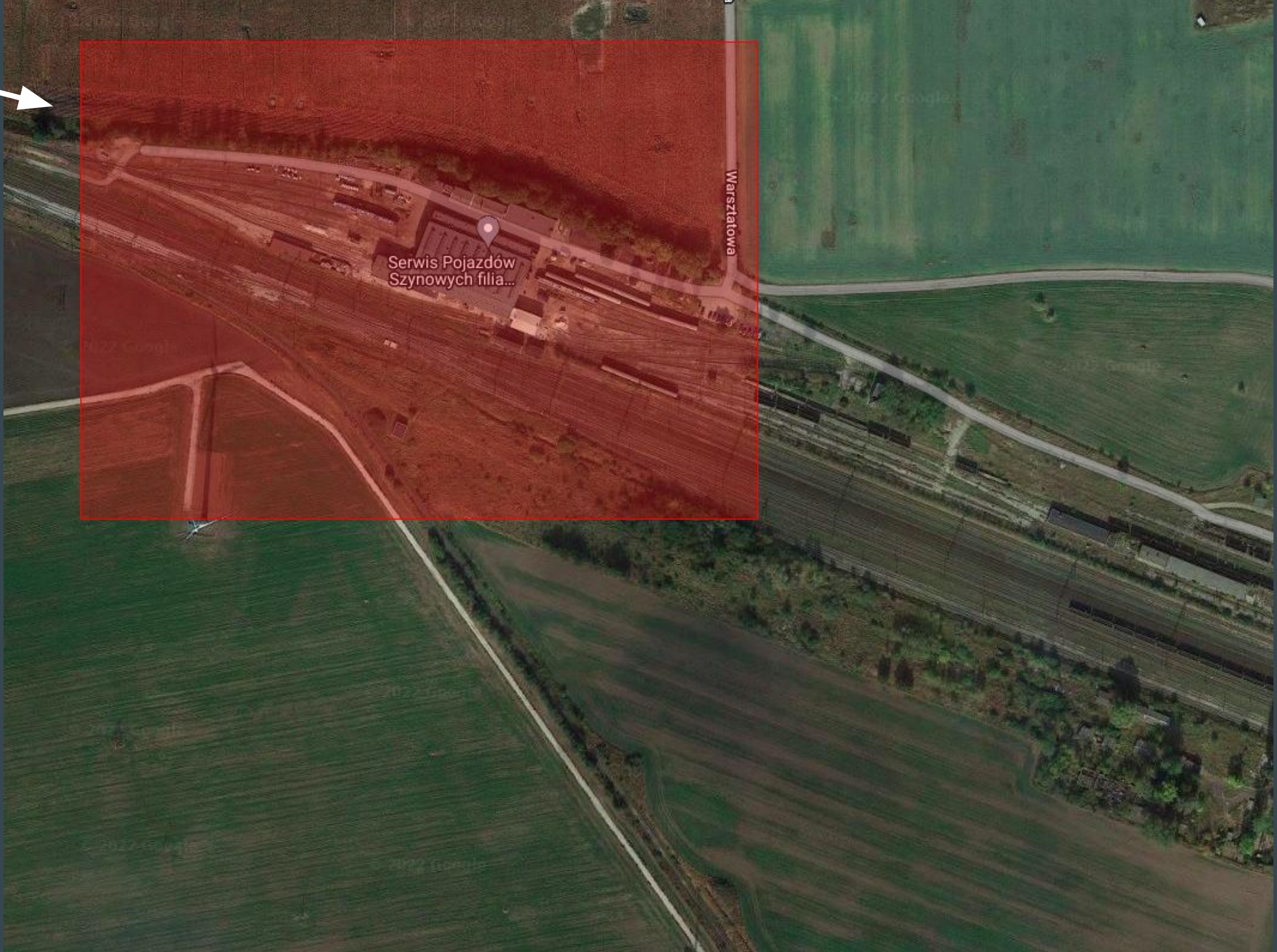

Aleksandra Krzywca
Aleksandra Krzywca
Serwis Pojazdów Szynowych ASO Piotr...
Metalko Zakład Powłok Antykorozyjnych
ia Pianek
n Sp. z oo
Aleksandra Krzywca
Aleksandra Krzywca
Aleksandra Krzywca
Nicro-plast Sp. z o.o

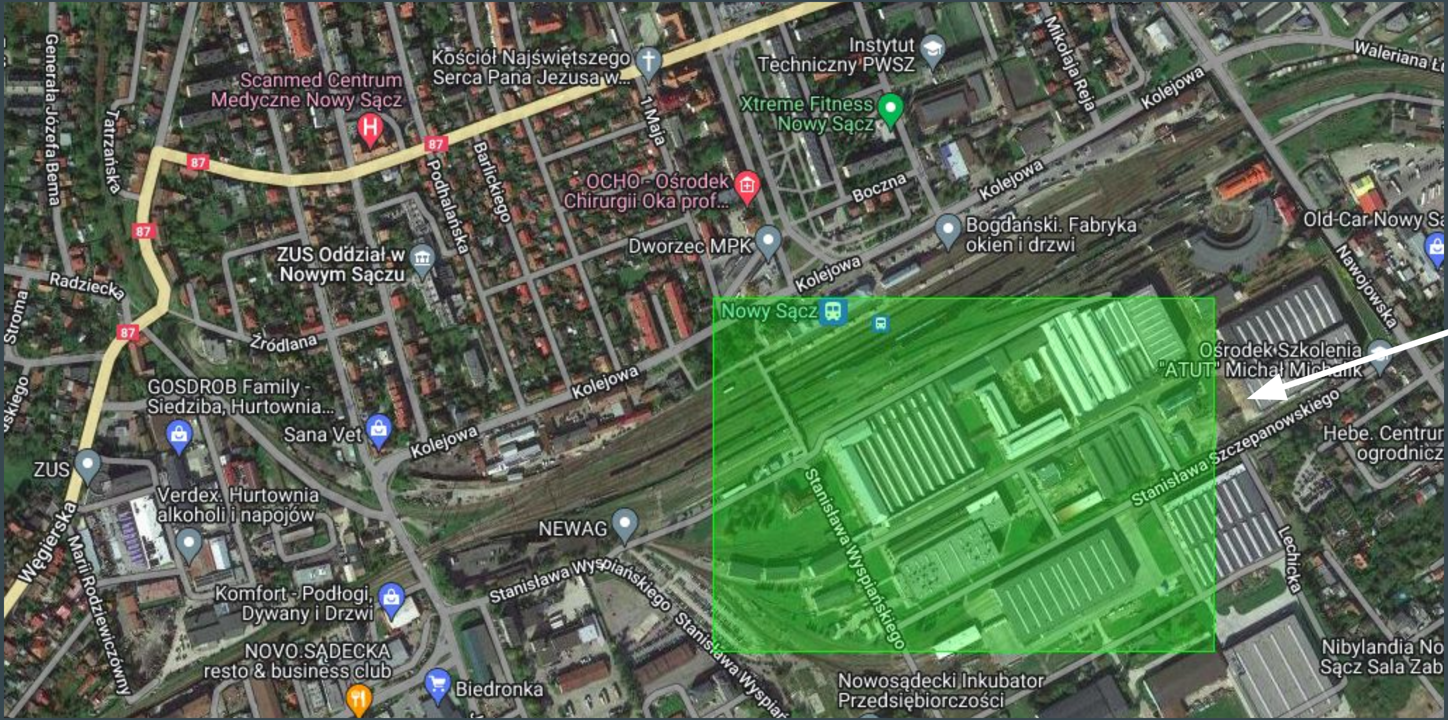
check6



Miła
Miła
Fabryczna
dynamic technology sp. z oo
Fabryczna
SKZ AK Kolman (Magazyn Główny)
Fablok Sp. z o.o
Hydrosolar. Hurtownia instalacyjna
Fabryczna
Kolonia Fabryczna
Zespół Szkół Technicznych...
Radosz Jacek. Usługi budowlane
AC Poland Zawada Bogusław
FABLOK-MARTECH PLUS
Picchi Qualita Sp. z o.o Przędza
Dr Green. W zgodzie z naturą
Profit SC - Meble na wymiar, meble...
GEO-PRODUCTS
Gminny pla

check7




check4
Nowy Sącz
NEWAG
Dworzec MPK
Kolejowa
Stanisława Wyspiańskiego
Stanisława Szczepanowskiego
Ośrodek Szkolenia "ATUT" Michał Michalik
Nowosądecki Inkubator Przedsiębiorczości

```
&& (var3 & 1)
```

Lockup triggers: Miscellaneous

- CAN831 serial change
- WTB module serial change
- Inverters firmware version
- 1.000.000 km
- Odometer inconsistency

Lockup triggers: Date

Found only in the compressor-failure-lock:

```
day >= 21 && month >= 11 && year_lo >= 21
```

Intent: Fail compressor after 21st November 2021?

Reality: Compressor breaks November 21st-30th and December 21st-31st.

Code sample: Compressor failure

Polregio ED78-010 / 31WE-015
SHA256: 0411b9af468331e0593674b03acd1a06
ede2aef90f553704405da1806d231638

```
[...]
G_181_get_current_time(cur_time);
sr->SET1 = 21 <= cur_time->day
        && 11 <= cur_time->month
        && 21 <= cur_time->year_lo;
sr->RESET = ton_reset->Q;
SR(sr);
me->is_date_after = sr->Q1;
[...]
```

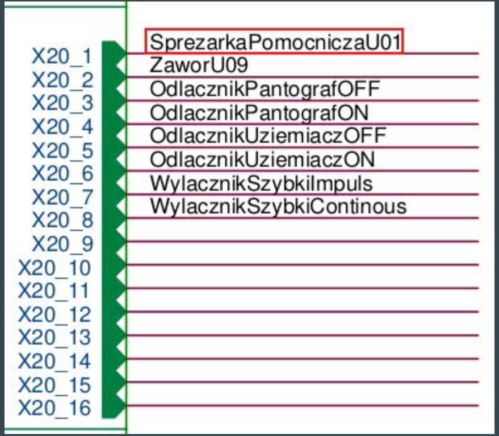
```
[...]
r_trig_odo->CLK = 1000000 < reg_odometer_km;
R_TRIG(r_trig_odo);
trigger_lock = r_trig_odo->Q
            || (car_?_udpcan_input.lock_train & 1)
            || me->is_date_after;
rs_lock->SET = trigger_lock;
RS(rs_lock);
me->is_train_locked = rs_lock->Q1;
// [...]
rs_lock2->SET = trigger_lock;
rs_lock2->RESET1 = ton_reset->Q;
RS(rs_lock2);
NVRAM_lock_enabled = rs_lock2->Q1;
[...]
```

Code sample: Compressor failure

Polregio ED78-010 / 31WE-015
SHA256: 0411b9af468331e0593674b03acd1a06
ede2aef90f553704405da1806d231638

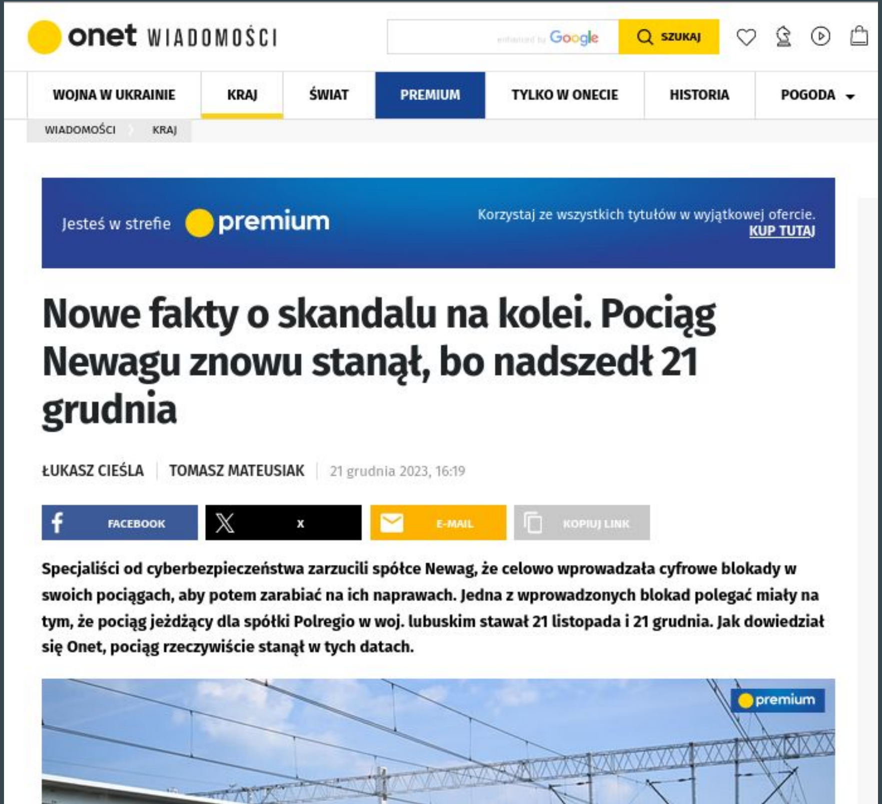
```
compressor_on &= (car_B_dc_input.presostatU05 & 1)
              && !me->used_more_than_10_min
              && !NVRAM_lock_enabled;
car_B_dc_output.sprezarkaPomocniczaU01 =
    car_B_dc_output.SprezarkaPomocniczaU01 & 0xfe
    | compressor_on;
// [...]
timer->IN = compressor_on;
timer->PT = 600000; // 10 minutes
TON(timer);
rs->SET = timer->Q;
rs->RESET1 = NVRAM_ac00207b;
RS(rs);
me->used_more_than_10_min = rs->Q1;
```

car_B_dc_output



Real life compressor failure

Polregio ED78-010 / 31WE-015
SHA256: 0411b9af468331e0593674b03acd1a06 ede2aef90f553704405da1806d231638



Nowe fakty o skandalu na kolei. Pociąg Newagu znowu stanął, bo nadszedł 21 grudnia

ŁUKASZ CIEŚLA | TOMASZ MATEUSIAK | 21 grudnia 2023, 16:19

Specjaliści od cyberbezpieczeństwa zarzucili spółce Newag, że celowo wprowadzała cyfrowe blokady w swoich pociągach, aby potem zarabiać na ich naprawach. Jedna z wprowadzonych blokad polegać miały na tym, że pociąg jeżdżący dla spółki Polregio w woj. lubuskim stawał 21 listopada i 21 grudnia. Jak dowiedział się Onet, pociąg rzeczywiście stanął w tych datach.

New facts in rail scandal. Newag train immobilized again, because it's December 21st.
Onet.pl, 2023/12/21

Unlock combo from HMI

Cheat codes, but for trains!

Different codes for different lock triggers.

Function got removed later, but we can still trigger the logic.

The real codes are more boring though :(

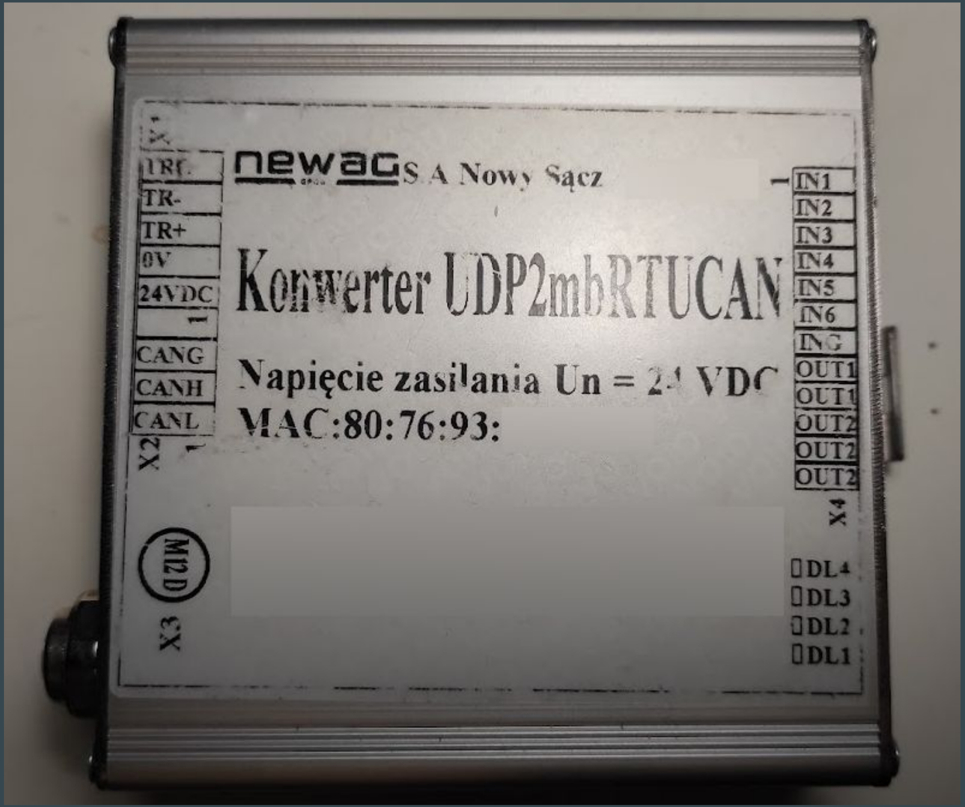
UDP/CAN converter

Found in some trains.

Not documented. Based on AT90CAN.

Connects to local telematics network (‘SIP’ / Passenger Information Services).

PLC code sends it lock state and in one case could be locked by it.

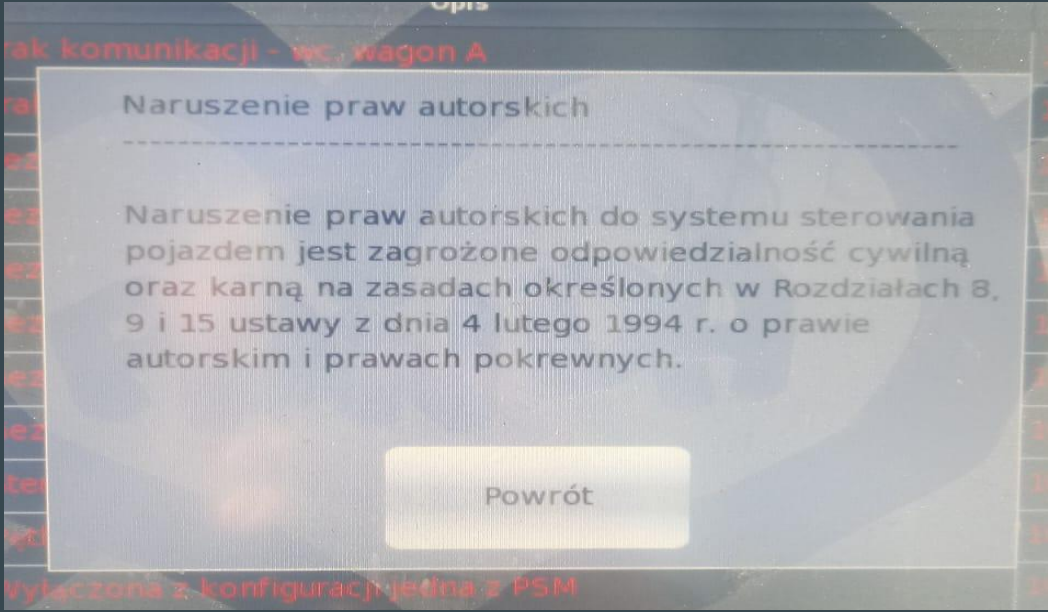


HMI secondary checks

New feature in some trains after being serviced by Newag.

‘Copyright infringement’

Violating the train control system’s copyright can have consequences blahblahblahblah


ak komunikacji - wc. wagon A

Naruszenie praw autorskich

Naruszenie praw autorskich do systemu sterowania pojazdem jest zagrożone odpowiedzialność cywilną oraz karną na zasadach określonych w Rozdziałach 8, 9 i 15 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych.

HMI secondary checks

The HMI is Linux + Qt. Not difficult to analyze?

So what’s violating Newag’s “Intellectual Propery Rights”?

Not moving the train for 21 days and then successful starting :)

Case study: POLREGIO Kraków / Sucha Beskidzka

4 locked up trains stuck in ‘lockup’.

Locks up again after powercycle. Programming bug?

You cannot fix them without modifying code. And we’re not doing that.

Newag got hired to fix this.

Lockup code got fixed, deadline got extended to 21 days, key unlock removed.

Case study: POLREGIO Szczecin

Date reported by blackbox: 22.04.2037

Aaaand locked. (15 years >= 10 days)

soruce: www.deuta.com

Train summary

Analyzed trains: 30

Trains with locks: 24

Most popular triggers:

1. Long stop (23)
2. CAN831 serial (20+)

Rarest triggers:

1. Long stop + GPS geofencing (2)
2. 1MKm / Date (1, compressor failure)
3. UDP/CAN converter (1+)

PLC User Code Metadata

KD 45WE-024

SHA256: 9eca6884b5ec136cdc9f4e89fa728d0a
2b6d5b4ff9bc9c94538496dcb7d18111

Contains project name, filesystem path, compilation date.

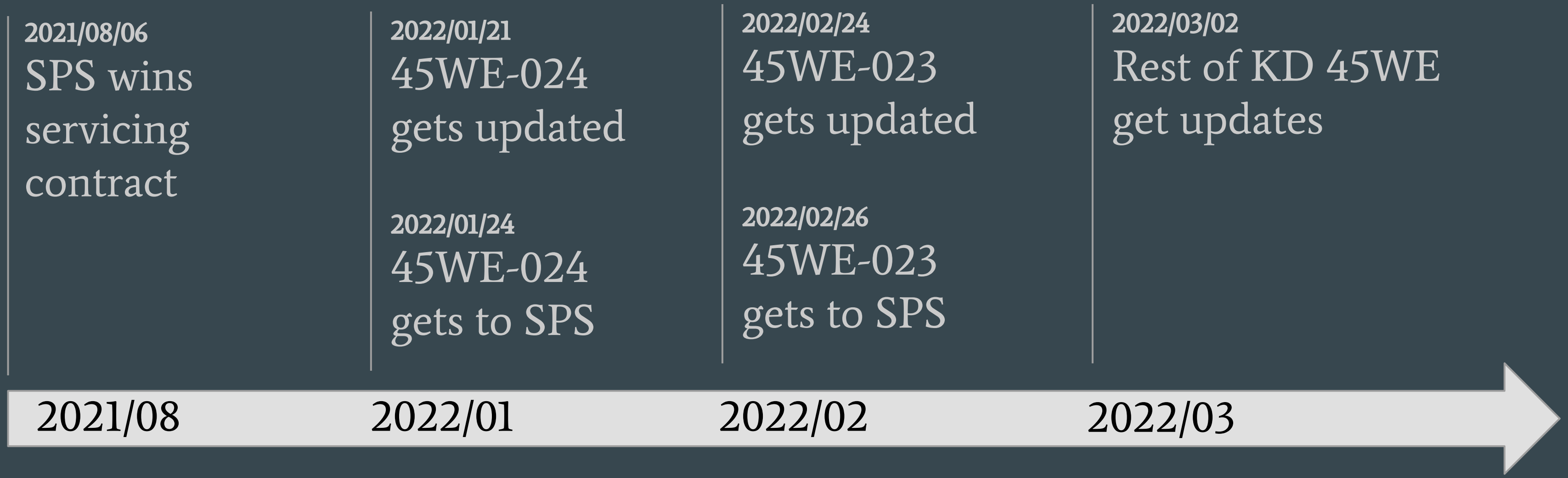
Can be trusted until some point: we found some weird stuff.

PLC logs an entry if a different software version gets uploaded.

```
[ProjInfo]
Name=c:\programycap\45we_kds\kopia\2017_newag_45we_kds\2017_newag_45we_kds
Resource=Resource
compiled=28.10.2021 12:11:02
Version=321
Comment=
[ENVIRONMENT]
CAP1131=5.9.0101.1827
OS=MOS83x_831 V703.C96C
CPU=CPU831
```

PLC User Code Metadata

A software update timeline can be reconstructed for KD trains:



2021/08/06
SPS wins servicing contract

2022/01/21
45WE-024 gets updated

2022/01/24
45WE-024 gets to SPS

2022/02/24
45WE-023 gets updated

2022/02/26
45WE-023 gets to SPS

2022/03/02
Rest of KD 45WE get updates

2021/08 · 2022/01 · 2022/02 · 2022/03

Newag’s response(s)

‘The hackers / someone else did it’

‘This is slander/defamation’

‘There is no proof’

‘We haven’t touched this train since 2018’

‘This is a violation of IP laws’

‘The software was interfered with and these trains are now unsafe’

‘SPS needed special software/docs’

‘SPS is incompetent and shouldn’t service Newag’s trains’

Next Steps

We are working on a combined technical report in English. Sorry, it’s a lot of work.

Newag still hasn’t sued us.

Some stuff happening at UOKiK (Competition / Customer Protection Office).

Some stuff happening at a few Public Prosecutor’s Offices.

Likely to be brought up in a Parliamentary Commission.

Questions?

37C3

UNLOCKED

@redford@infosec.exchange
@mrtick@infosec.exchange
@q3k@hackerspace.pl