UNLEASHING THE POWER

OF MY 20 YEARS OLD CAR

Stanislas Lejay (WhiteMotion)

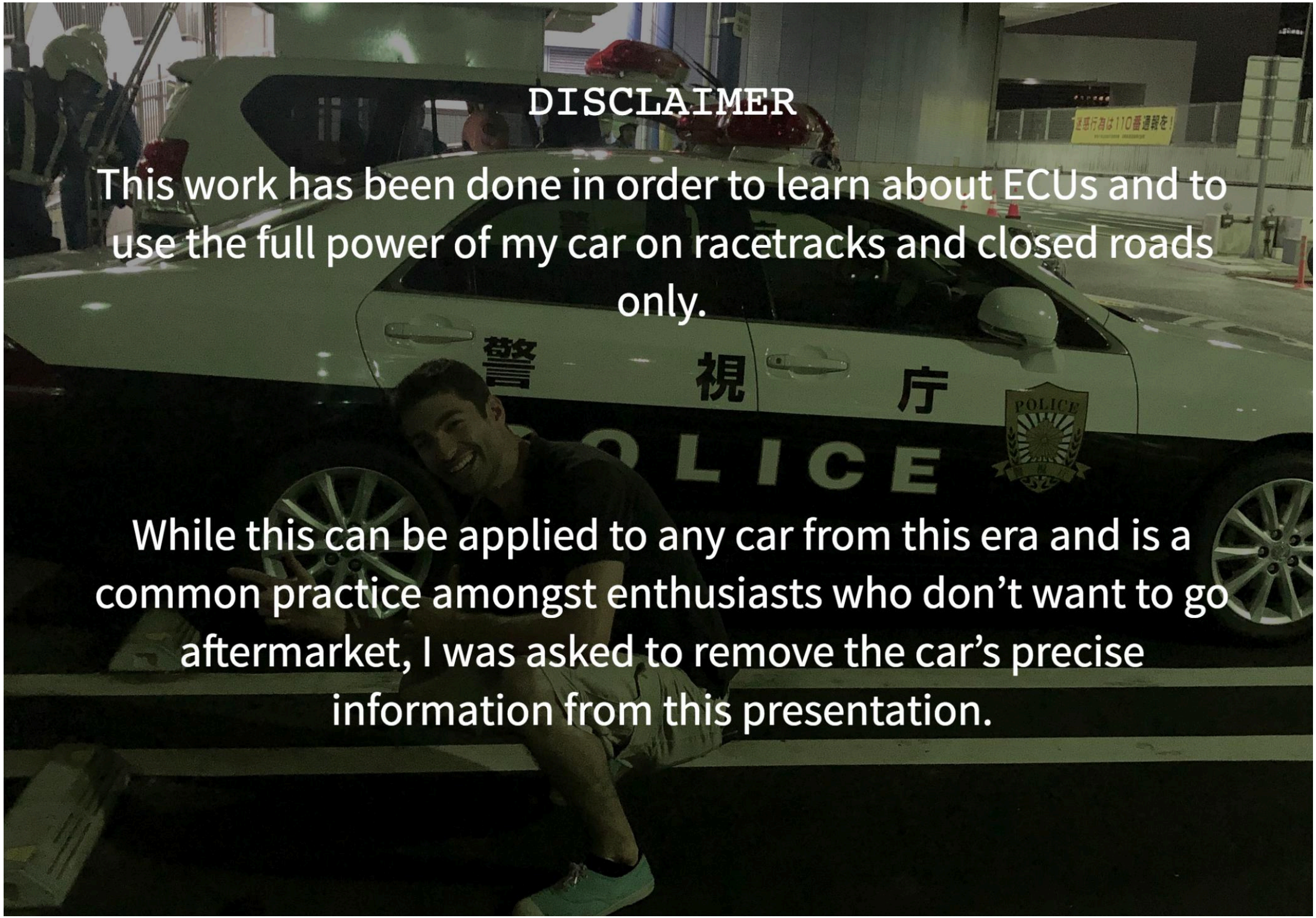
Black Hat Europe 2019

Who's that brat?

* Stanislas Lejay, french computer security engineer
* Love cars, and to fiddle with things
* Automotive vulnerability researcher at WhiteMotion (Tokyo, JP)

WHITEMOTION

Automotive Cyber Security



DISCLAIMER

This work has been done in order to learn about ECUs and to use the full power of my car on racetracks and closed roads only.

While this can be applied to any car from this era and is a common practice amongst enthusiasts who don’t want to go aftermarket, I was asked to remove the car’s precise information from this presentation.

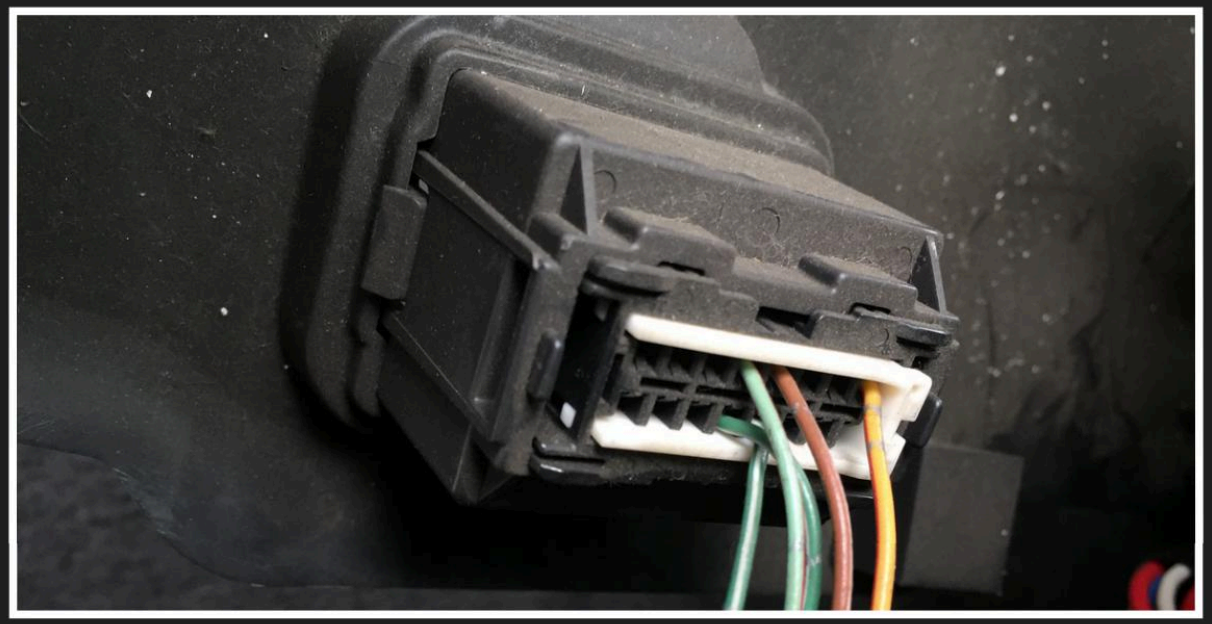
The test subject

* 1997 [MANUFACTURER] [MODEL]
* ~300hp from factory, mosty stock but for suspensions
* Nicknamed `Little Beast`
* 50 000km driven since I bought it, still rips

Playing around

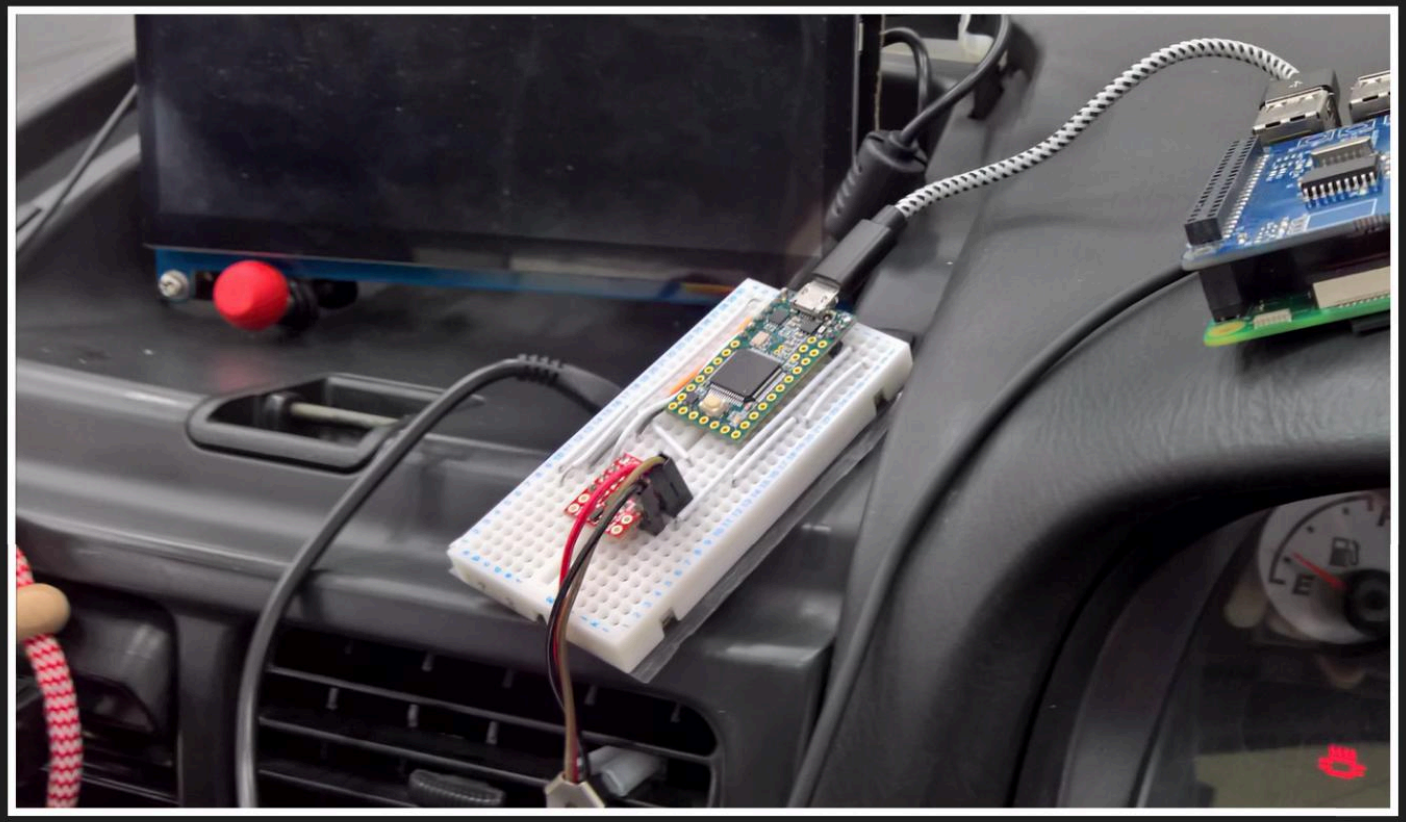
* I was developping my own IVI
* It has data logging options, alarms, music, etc
* Powerful tool when it can communicate with the ECU

Reconnaissance



* How to communicate with the car ?
* Features an OBD-II like port, but no CAN
* xSM is a proprietary protocol from [THAT MANUFACTURER]
* Documented on `alcyone.org.uk`
* No K-Line connected, xSM1 it is

xSM1



* Simple serial protocol running at 1953 bauds at 5V TTL.
* Teensy + level converter is all I need

Reading from the ECU

```
byte simple_read_data_from_address(short addr) {

  // 78 msb lsb 00
  byte read_cmd[4] = {0x78, byte(addr >> 8), byte(addr & 0xff), 0x00}
  byte answer[3] = {0};

  HWSerial.clear();
  for (int i = 0; i < 4; ++i) {
          HWSerial.write(read_cmd[i]);
  }

  HWSerial.flush();
  HWSerial.readBytes(answer, 3);
  stop_read();
  return answer[2];
}
```

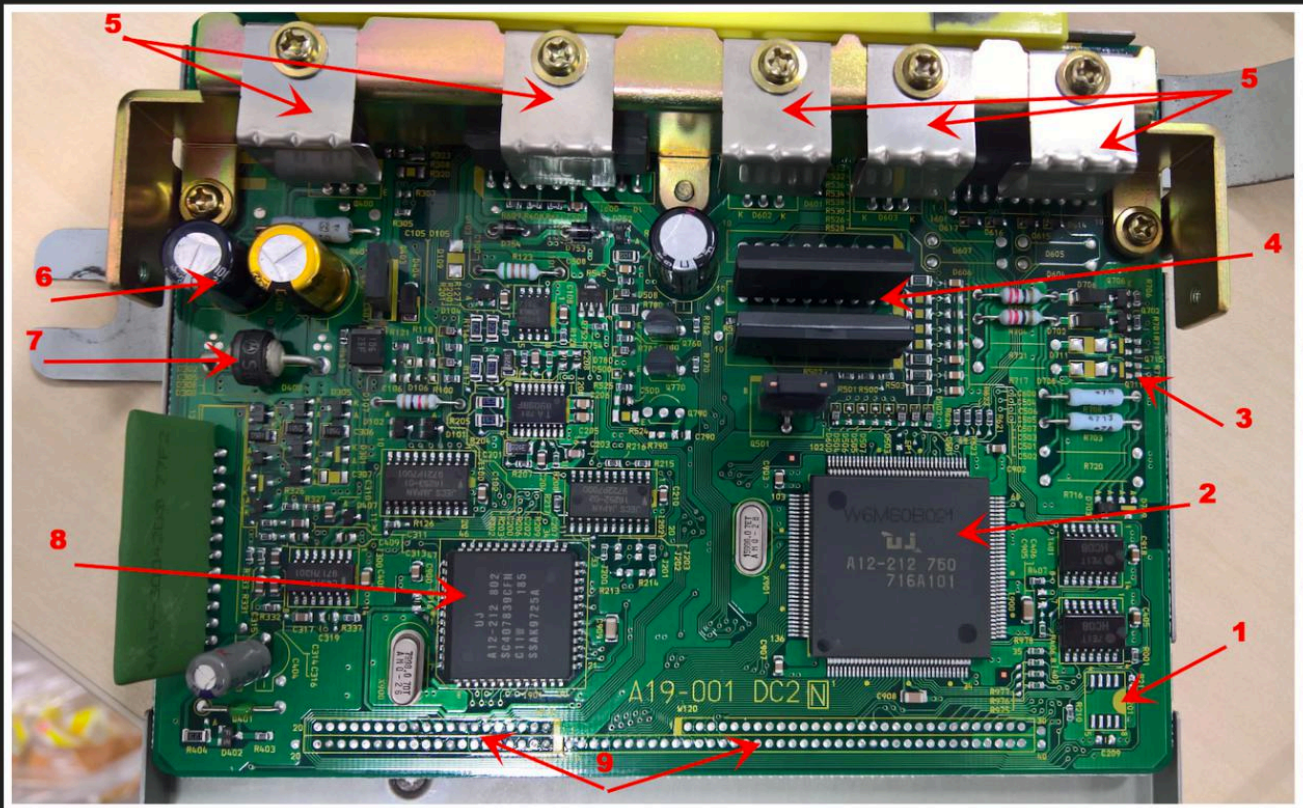
Dumping the ECU

* About 5 queries per second
* Plug the car's battery to a charger
* for loop from 0 to 0xffff, with a few checks
* Dump the whole address space in 9ish hours

Finding the ECU

* What architecture is it ? What ISA ?
* Need to check the CPU to determine that
* Take the ECU from under the passenger carpet (!!)

ECU and Processor



1- Immobiliser Chip (none)

2- Main CPU + FW

3- Ignition circuitry

4- Low power transistors

5- High power transistors

6- Power management

7- Diode

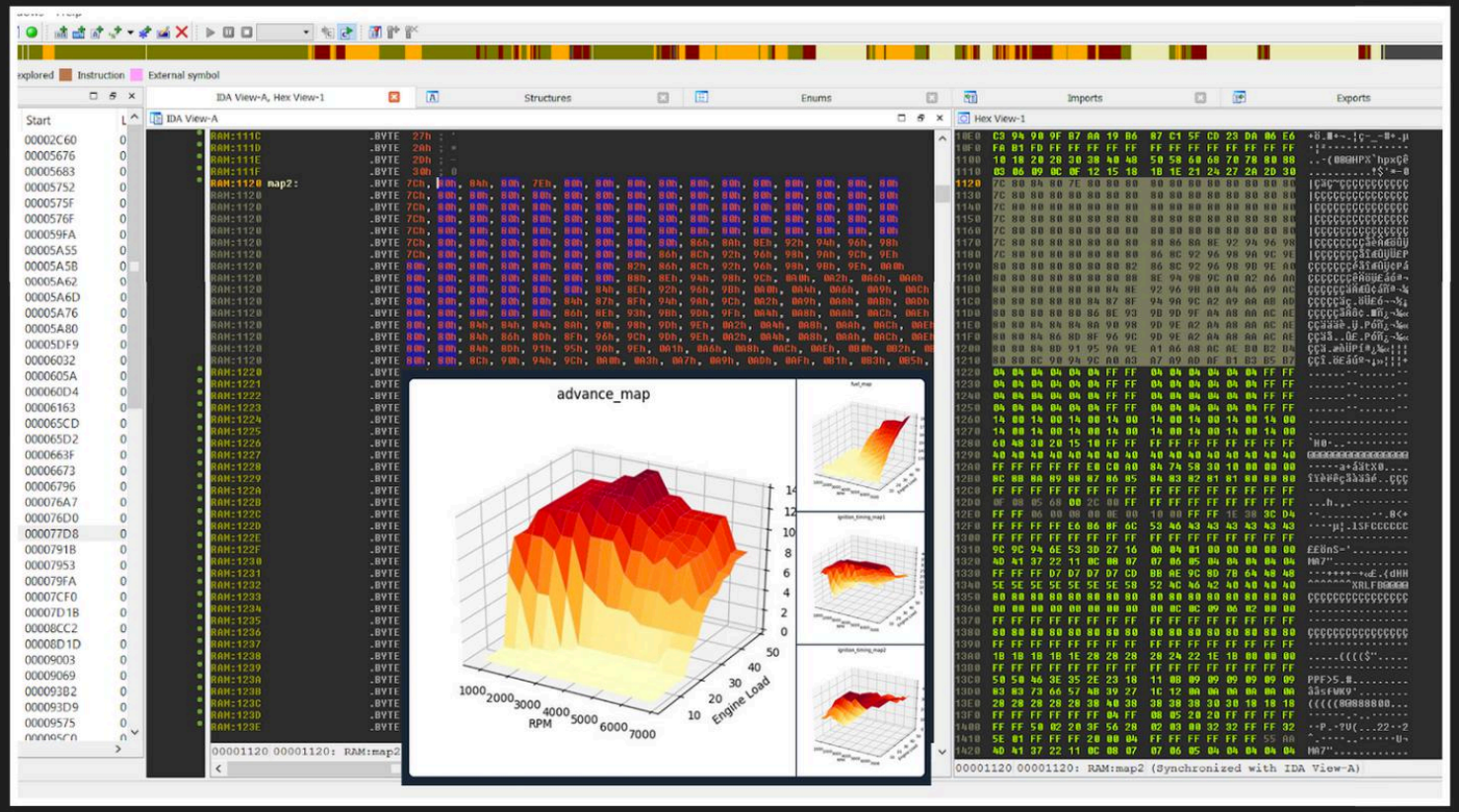
8- Secondary custom IC

9- Contact strip

* IC based on the Mitsubishi M37791 CPU (7700 family)
* `M flag` changes the instruction decoding at runtime
* At least IDA handles it

Reverse engineering

* 64kb blob, where to start ?
* Turn everything into code.
* Look for maps, find the Xrefs to those maps, and go from here
* Struggle until you have something ok-ish (normal RE)



And my IVI works

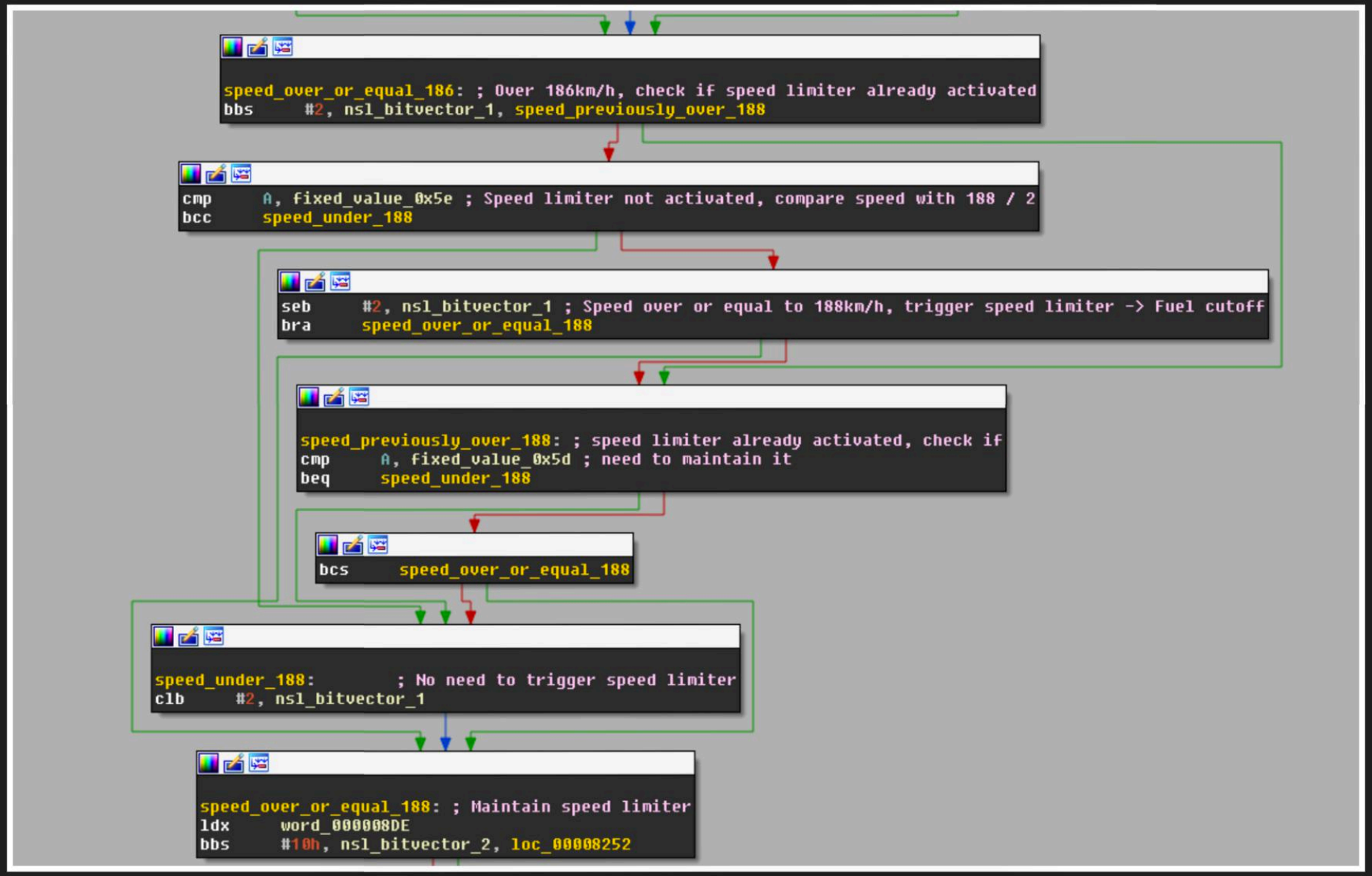
History: JDM speed limiters

* Cars sold in Japan used to have a ringing sound when going over 100km/h
* Nowadays, this became an ECU controlled speed limiter at around 180km/h
* Some cars have options for circuits

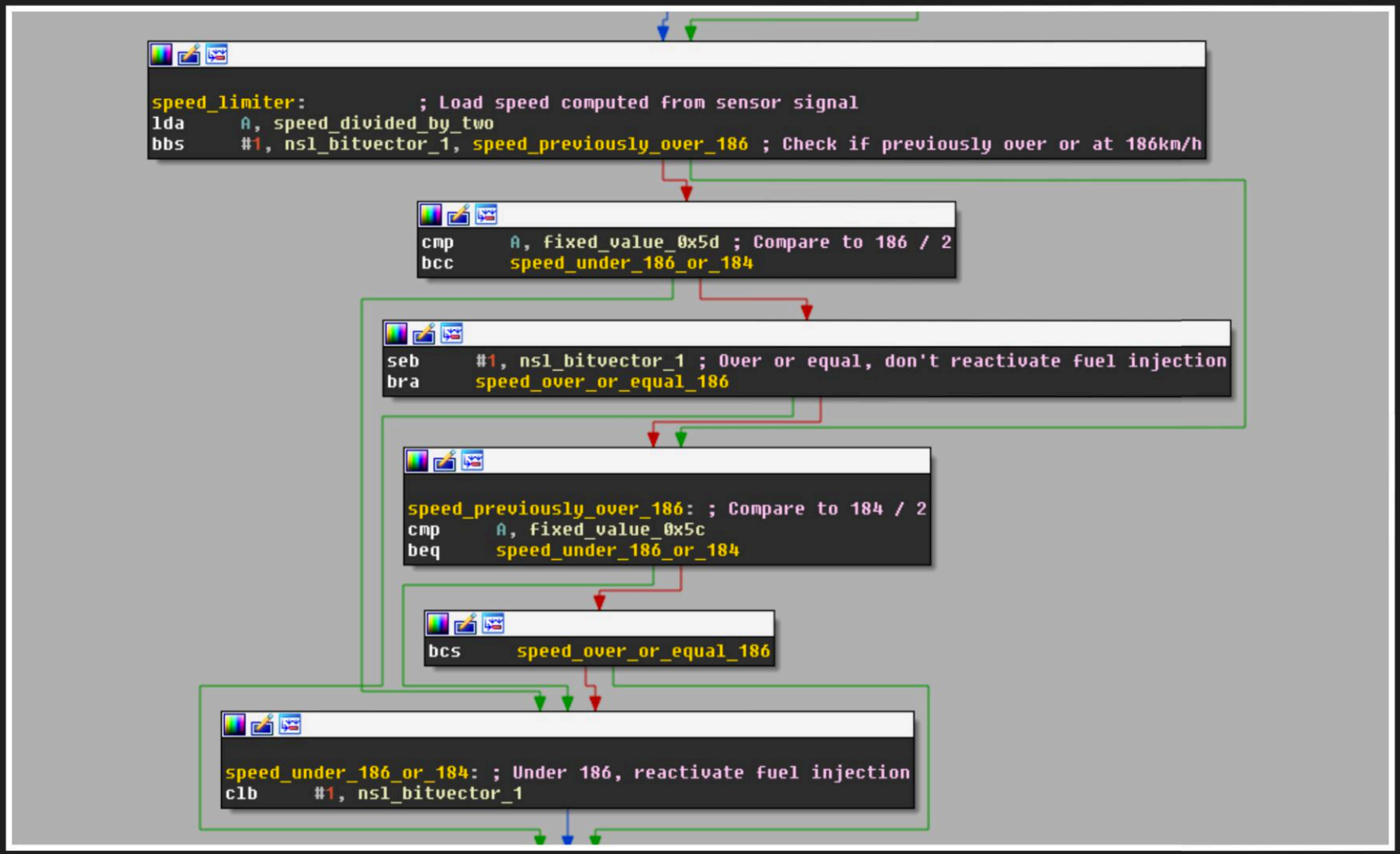
My car is no exception

* Fuel cut is pretty brutal
* Can I get rid of it ?
* Need to understand how it works

Speed limiter: Activation



Speed limiter: Deactivation



The different options

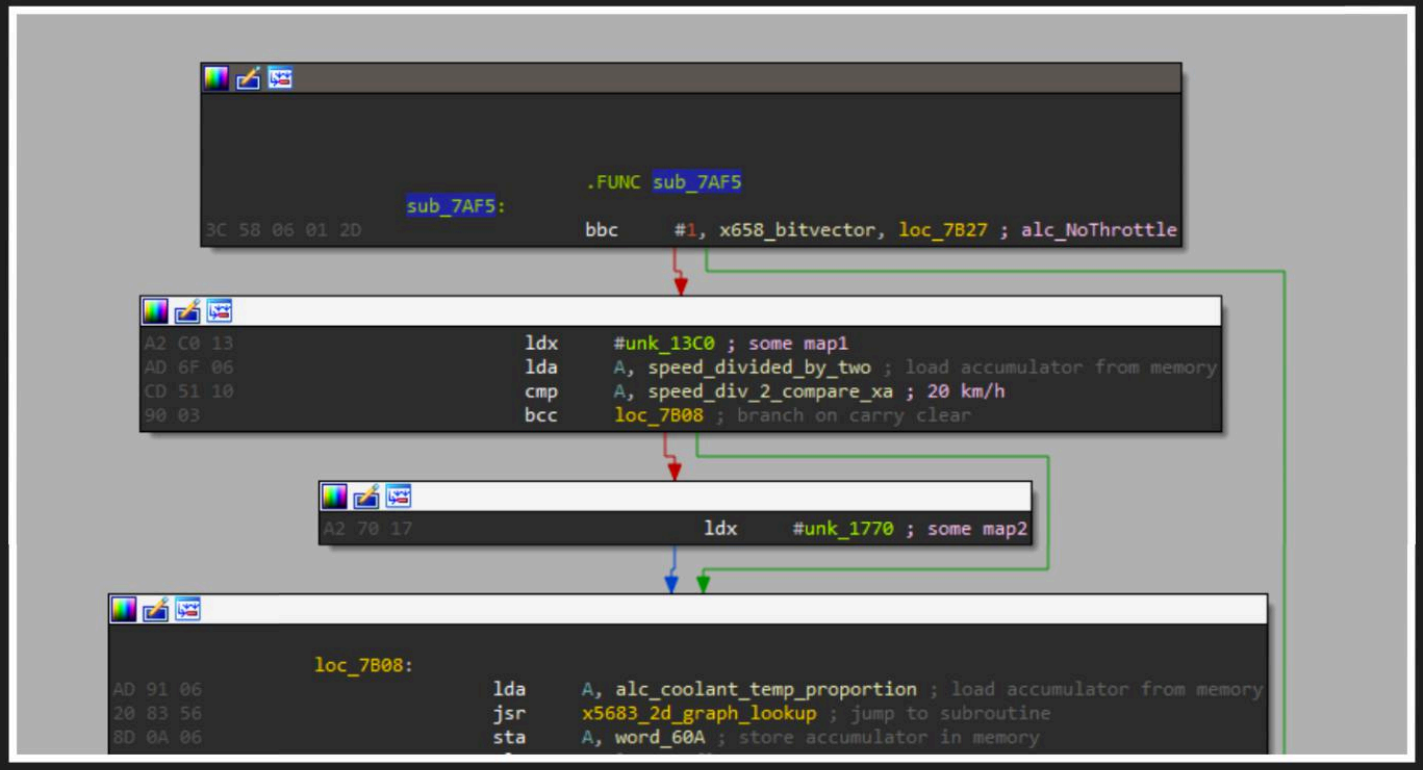
Bypassing the limiter: Aftermarket ECU

* Easiest and most customizable solution but
  - Not cheap
  - Needs a retune
  - I'm losing all the work done so far

Bypassing the limiter: Daughterboard

* Piggybacks on the stock ECU but
  - Not cheap
  - Needs a retune

Bypassing the limiter: Cut VSS

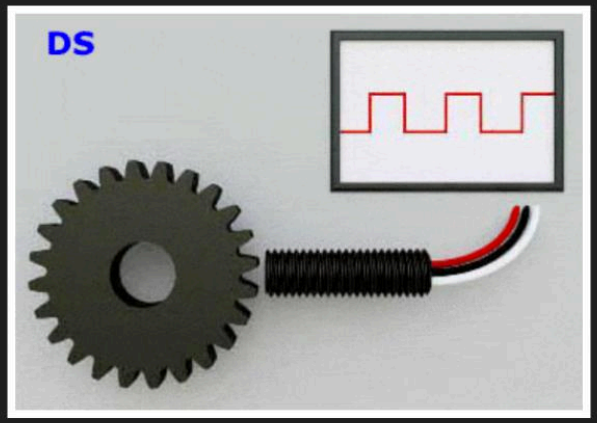


* Easiest, cheapest, easiest to understand solution but...
* Transition Maps
* Verification in code leading to limp mode

Bypassing the limiter: Faking the VSS signal

* HKS sells the "Speed Limit Defencer" (SLD)
* Not cheap (100-200 euros)

Faking the VSS signal for cheap*



* SLD's principle is fairly easy
* Intercept the signal and send a dummy one if needed
* The ECU still receives a high speed signal, but not too high

GIF SOURCE: HTTPS://SENSORSO.COM/GEAR-DETECTION-SENSORS.HTML

Teensy for the win

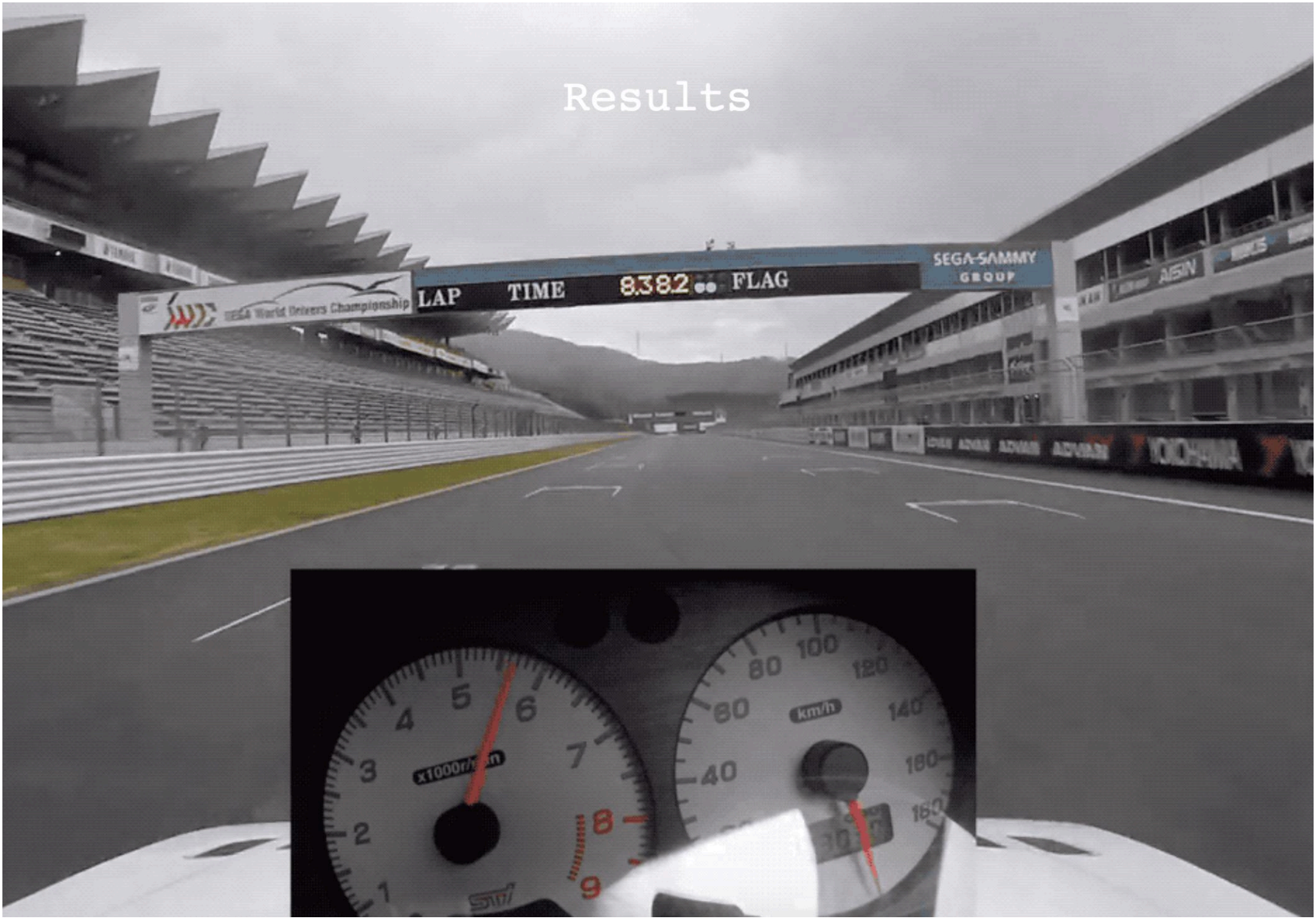
```
// Initial version of the of the limiter bypass
while (speed >= 180) {
        if (previous_signal_value) {
                signal_value = 0;
        } else {
                signal_value = 1023;
        }
        analogWrite(VSS_TX, signal_value);

        // Remove the delay to get 354km/h and stall the engine
        delay(5); // Speed recorded is around 140km/h

        previous_signal_value = signal_value;
}
```

and then...

Results


LAP TIME 8.382 FLAG
SEGA SAMMY GROUP
SEGA World Drivers Championship
AISIN
YOKOHAMA
km/h

Black Hat Sound Bytes

* Most aftermarket tools are not witchcraft
* ECUs are getting complicated, but the basics stay the same
* Go simple, but go safer

Thank you!

Twitter: 0xP1kachu