Measurements on 4 sensors heterogenous SPC network

Roland Gamper, Lahniss Sebastien Deuve, Melexis

November 3rd 2020

Table of contents

List of Figures

Introduction

SPC is an extension of the SENT protocol, aimed at applications for which a master-to-many-slaves interrogation model is desired, as opposed to the slave-to-master continuous broadcast model of SENT.

In SPC, an ECU, acting as a master, interrogates individual sensors using a Master Trigger Pulse (MTP). Depending on the length of the MTP one and only one of the sensors will react by sending its data to the master as a SENT message. There are no structural differences between a SENT message broadcasted continuously by a SENT sensor and a SPC message emitted upon request by a SPC slave sensor.

The goal of this paper is to demonstrate the usage of the oscilloscope to verify all important aspects of a SPC network, as well as locating and understanding a typical error.

SENT and SPC specifications

SENT is formally specified by the Society of Automotive Engineers (SAE) under the denomination of J2716 201604. The current specification is available through www.sae.org in particular https://www.sae.org/standards/content/j2716_201604/ describes the inner workings of the Fast and Slow channels in detail, with various examples of applications with different parameters.

SPC is formally specified by Infineon. The current specification is available with Infineon.

SPC physical interface and frame description

In SPC, both the master and slaves have to drive the line, as opposed to standard SENT, with which only the sensor drives the lines, while the ECU only monitors the line.

Consequently, each element on the bus (master or slave) needs to have its output stage in high-impedance while it is not requesting (master) or transmitting (slave), to prevent interference between the master/slaves on the bus.

Physically, a SPC network can be represented by the schematic below: the master and slaves share the same line (sensor output). A pull-up ensures that the idle level is high when neither the master nor the slaves are active.

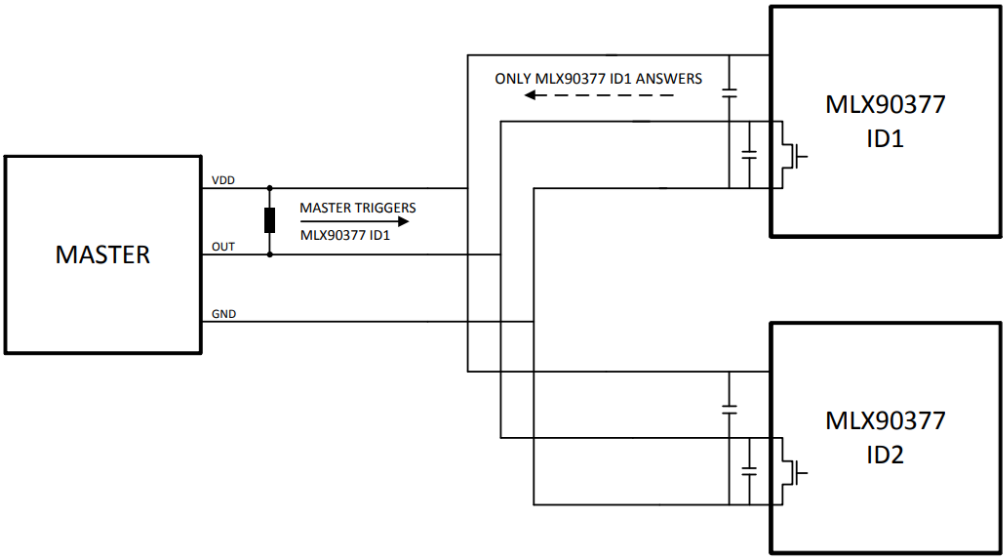


Figure 1: Schematic of a 2-sensor SPC bus

SPC frame architecture

In SPC, an additional step of triggering/requesting an answer from one of the slaves is required before the transmission of the frame. This step is highlighted in the waveform in Figure 2.

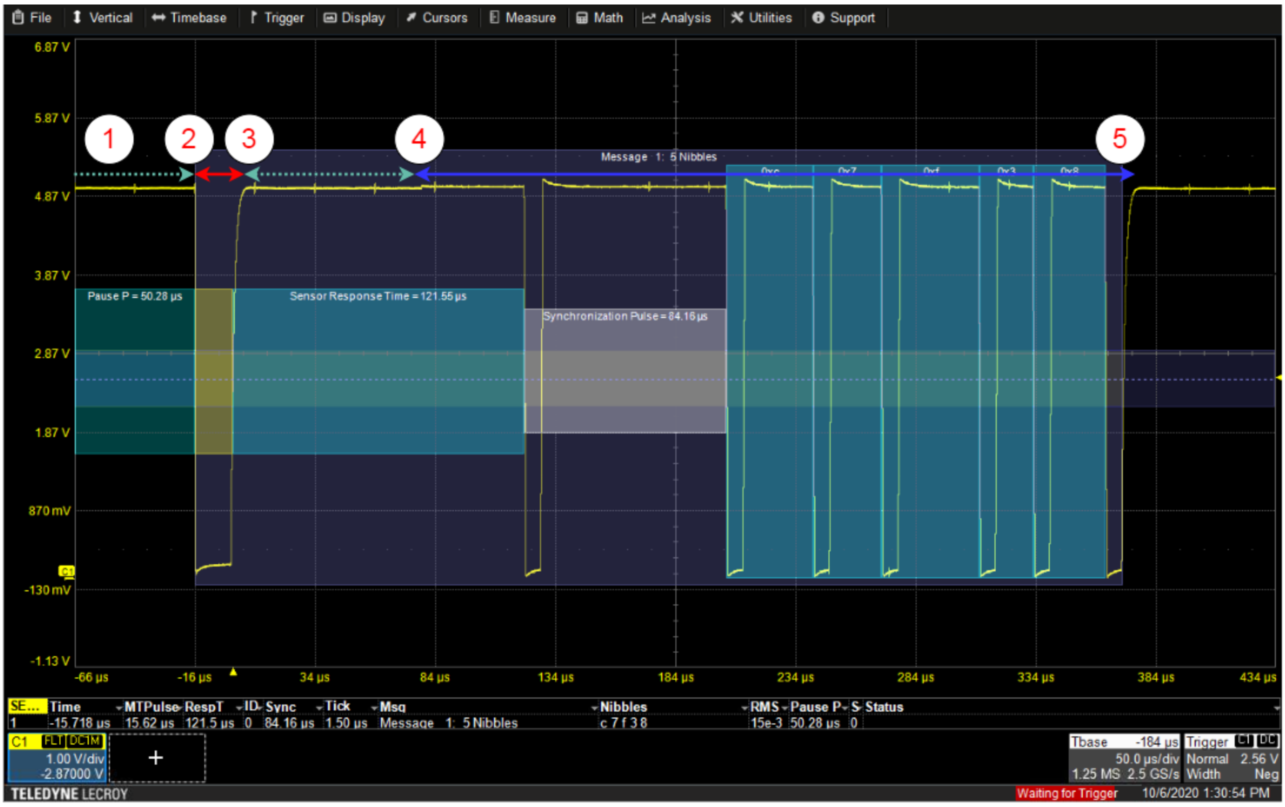


Figure 2: Triggering from master and answer by a slave with a digital push-pull output driver

The different steps are:

1. The line is in high impedance, waiting for the master to send a Master Trigger Pulse (MTP). Because of the pull-up, the line is pulled to the bus supply voltage.
2. The master pulls the line low.
3. The master releases the line after the duration of the MTP (determined by the sensor ID that needs to be triggered). The line is again in high impedance after.
4. The slave reacts to the MTP by starting to drive the line. The delay between the falling edge of the MTP and this transition is a configurable setting in MLX90377.
5. When the frame has been transmitted, the slave stops driving the line, which goes back to high-impedance. The moment when this transition happens is a configurable setting of MLX90377 (can be either after a fixed duration following the falling edge of the MTP, or a fixed duration after the last nibble). Once the line is in high-impedance again, the master is free to trigger again one of the slaves on the bus.

SPC frame with pulse shaping

MLX90377 brings innovation by combining SPC with pulse shaping, in order to reduce generated electromagnetic emissions. While the structure of the frame is exactly the same as in push-pull mode, the transitions from high-impedance to pulse shaping and vice versa are more visible. Indeed, the amplitude of the signal in pulse-shaping mode is modulated in order to be as close as possible to the SPC physical layer level specifications (Vol<0.5V ; Voh>4.1V), aiming to reduce the generated electromagnetic emissions, while the high impedance level is equal to the supply voltage (because of the pull-up resistor on the signal line).

A standard pulse-shaped SPC frame is represented in the scope screenshot below, showing the different phases (high impedance, pulse shaping, high impedance) and their transitions:

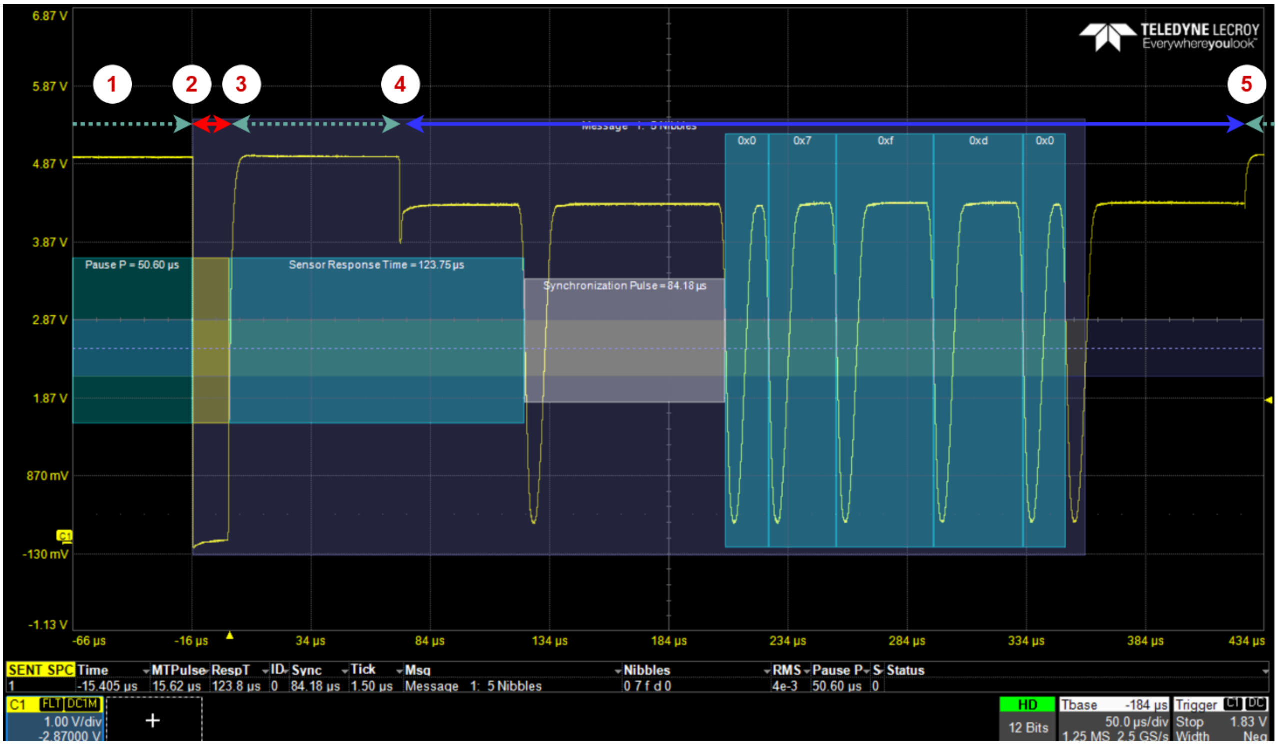


Figure 3: Triggering from master and answer by a slave with a analog “pulse shaping” output driver

Mini-Network implementation, equipment used

Network structure, schematic and topology

This image explains the general organization of the experiment. It is built in such a way that we expect very similar, or even quasi identical results on all 4 decoders. If this were not the case it would indicate that the concept is flawed, or some pathology is affecting one or more of the components chain.

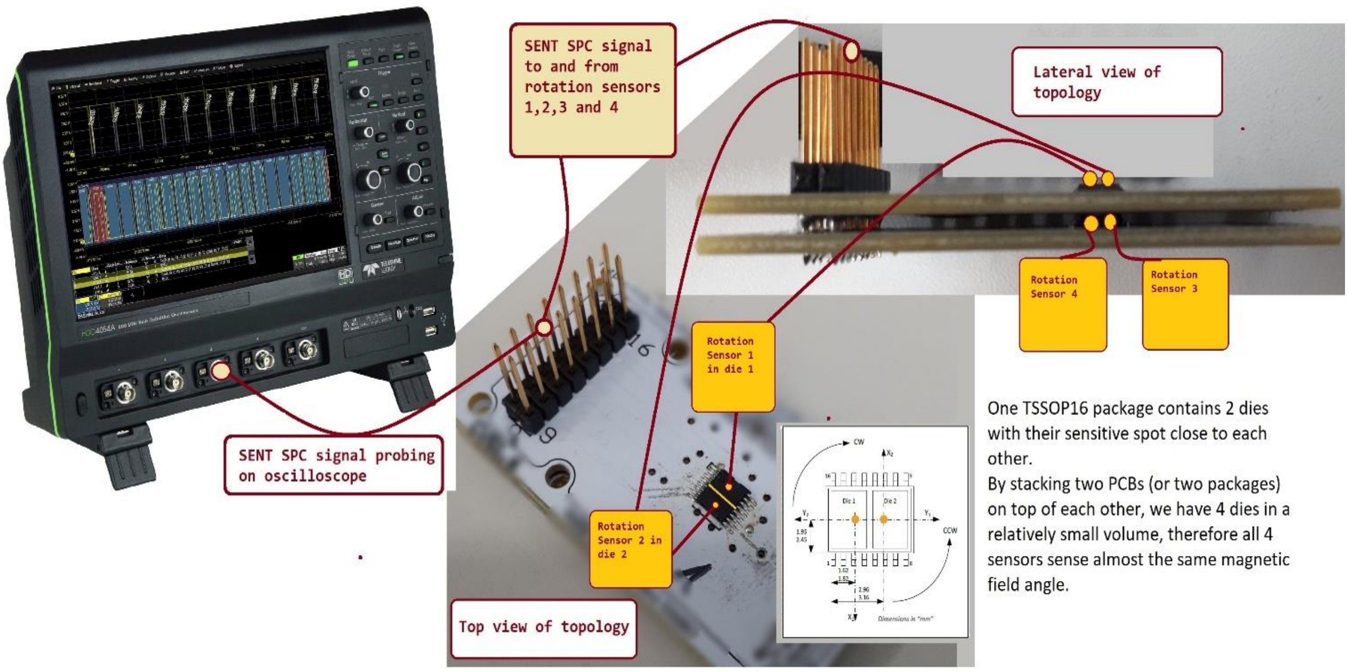


Figure 4 Complete setup used for the experiments, including sensors, oscilloscope and PTC-004

The position of the probing point yields insight about all the important aspects of the component chain: quality of the angle measurement by the sensors, transmission of the angle value from sensors to the microcontroller, correctness of the microcontroller queries to the sensors, adequacy of the probing method itself and pertinence of the oscilloscope settings. In this setup, the oscilloscope measures the signal close to the OUT pin of the sensors (see Figure 1).

Common Stimulation of all sensors on network

The stimulation of the Hall effect sensors is provided by a small magnet rotated by a hand-held drill above the tightly arranged 4 sensors.

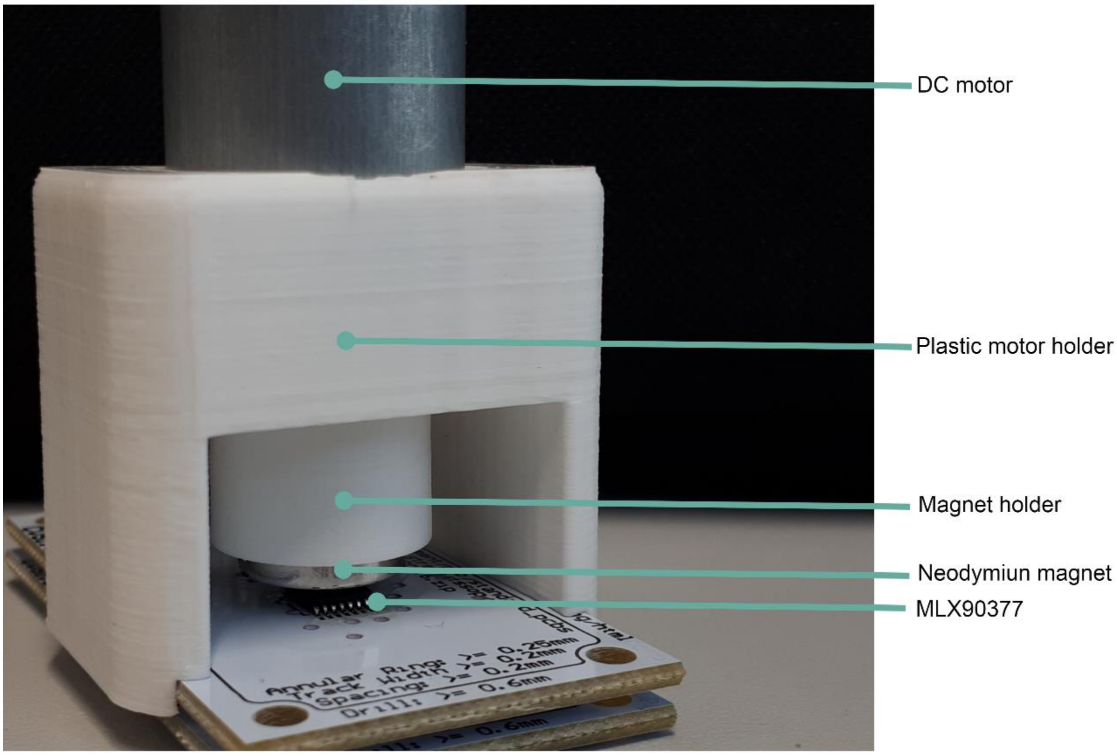


Figure 5 Rotating magnet used to stimulate MLX90377 angle sensors

This simple method warrants similar data on all 4 SPC channels, therefore simplifying the comparison of the 4 decoded results on the oscilloscope.

Equipment used for the experiments

Melexis magnetic position sensor:

https://www.melexis.com/en/products/sense/position-sensors

Oscilloscope and decoder:

https://teledynelecroy.com/options/productseries.aspx?mseries=617&groupid=88

Probe PP026:

https://teledynelecroy.com/probes/passive-probes/pp026-1

Sensor configuration tool:

https://www.melexis.com/en/software-tools/ptc04

Sensors parametrization using the PTC-04 programming tool

Typical SPC sensors support a wide range of settings through their internal registers. This configuration is performed at the end of the manufacturing process by Melexis or by the customer, by using the programming tool Melexis PTC-04. These settings are stored in a non-volatile memory (NVRAM) and never modified after initial programming.

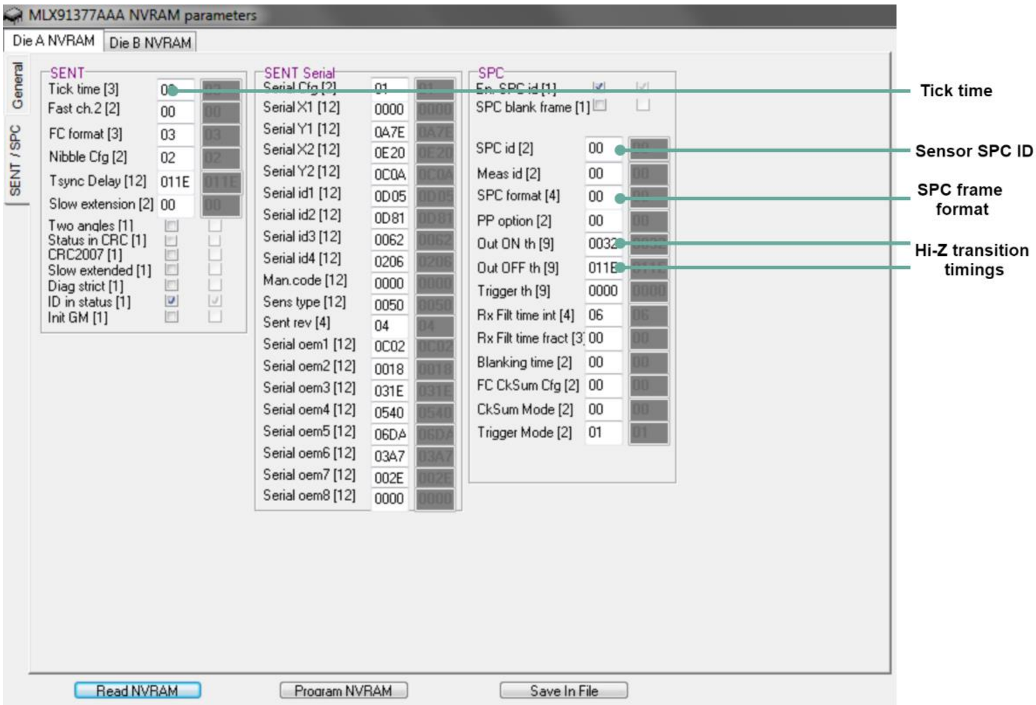


Figure 6: SPC configuration interface for PTC-04 and MLX90377

Each sensor can be programmed by Melexis with a different communication identifier to allow programming in bus mode up to 4 sensors. The SPC protocol also allows sensor interrogation in bus mode up to 4 sensors, therefore the devices can be assembled in a module and programmed or calibrated on a bus.

For this experiment, the sensors have been re-programmed several times in order to allow a variety of measurements, including the generation of controlled errors.

The oscilloscope is also used to verify that the sensors parametrization has succeeded. Most of the key elements of the SPC transmission can be observed using the oscilloscope.

Case 1 Observing a healthy homogenous network, all sensors identically parametrized

In this first experiment we observe the traffic on the SPC bus **interrogating 4 sensors configured in the same manner**. More precisely all 4 sensors use the same Tick Time (1.5 us), Nibble Number (5), Polarity Idle High) and CRC method, as well as nibble structure (angle encoded on 12 bits = 3 nibbles). However, each sensor is interrogated using a different Master Trigger Pulse (MTP), which is reflected in the decoder's dialog by entering the adequate values in micro-seconds, under ID 0,1,2 and 3.

Figure 7 Homogenous network, 4 sensors using single decoder, Column D0 aggregates the angle data from all sensors.

One can observe that sensor ID 1 is interrogated using an MTP of approximately 32 microseconds. The exact measured value of the pulse is 32.051 us and appears for example on line 427 of the decode table. The red frames and connecting lines in the image help understanding the user set value, the measured value and the actual visualization of the pulse in the signal for sensor ID 1.

For the sake of clarity, the traffic has been configured in **bursts of 4 messages** separated by an interval of approximately 2 milliseconds, while the interval between the messages (within the burst) is shorter, 700 microseconds. The following images documents the cadence of the messages, visually and numerically:

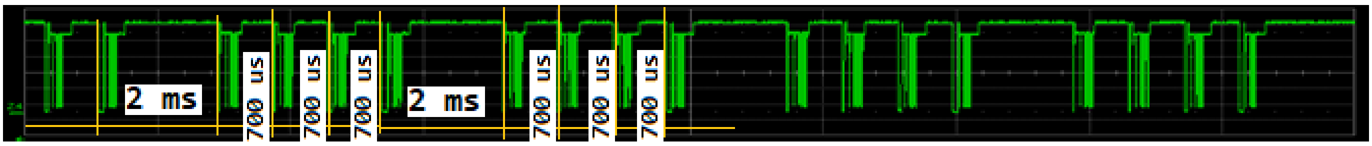


Figure 8 Cadence of messages and bursts viewed on a raw signal, at 2.6 ms/division

The tabular listing of the Pause Pulse shows the exact values of the intervals between the messages and the bursts

Pause P
704.8 µs
704.6 µs
2.005 ms
704.3 µs
704.8 µs
704.3 µs
2.006 ms
704.2 µs
704.7 µs
704.7 µs
2.006 ms
704.2 µs
704.7 µs
704.2 µs

Figure 9 Tabular listing of time intervals between messages and bursts

While the method above has the benefit of simplicity and ease of setup, it has limited value when it comes to evaluate the data emitted by each sensor. More precisely, the caveat of the experiment above, when monitoring sensors measuring different environment variables, is that the Data 0 column of the common decoder shows unrelated results from several sensors. Henceforth there is little value in graphing the Data 0 column of multiple sensors measuring different parameters (i.e. angle, torque, temperature, pressure, etc.) and aggregating them.

The next experiment will explain how to setup for decoding results from distinct sensors, (possibly configured differently in terms of nibble number and CRC method) to allow graphing of the output values of each sensor.

This next experiment, using the very same signals as in the first experiment, is a baseline, for which identical results are expected since all 4 ICs are identical both in their **hardware and firmware settings**.

We use 4 decoders with identical settings (except for the MTP length) to selectively decode and graph one sensor only per decoder. We also use 4 ColumnToValue parameters to convert the D0 column of each table into a list of data that is the source of the 4 Tracks & Rescale. As opposed to the first experiment, we need to track the emitted values from each sensor to observe its behavior over time.

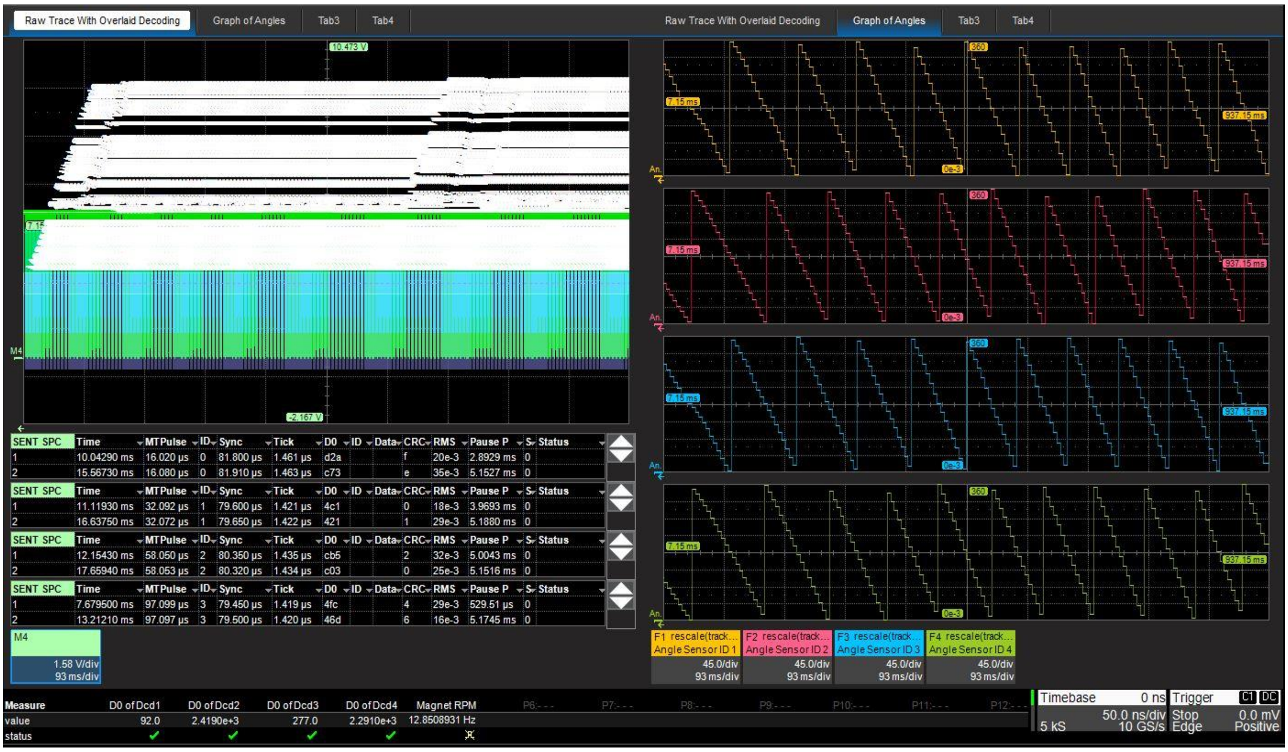


Figure 10 Measurement on 4 identical sensors, 1 and 3 shifted by 180 degrees, with magnet at 12.85 rps

The decoded trace in the left window shows the raw trace measured by the oscilloscope. The 4 decoders act on the same source trace M4 and superimpose their annotations on the trace, resulting in the heavily annotated signal. The 4 tables, below the raw trace, are the outputs of the 4 decoders, with column D0 dedicated to each sensor (and not interspersed as in the first experiment).

The 4 stacked graphs in the right windows show the angular measurement, decreasing from 360° degrees to 0, and wrapping back to 360°. The 180° phase shift between odd traces (F1 and F3) and even traces (F2 and F4) is due to the die's orientation in the packages (180° phase shift between two dies of the same package).

The graph has time in X and degrees in Y. Using the grid labels and the rescale functions, the values are plotted in the Degree vs Time graph.

The Frequency parameter P5, calculated on graph F1, exhibits a rotational speed of 12.85 revolutions per second, induced by the DC motor shown in Figure 5.

Case 2: Observing a healthy heterogeneous network, sensors with different CRC methods

In this case the difference with case 1 is the CRC computational method, which is **different for each one of the 4 sensors**. The measurement contents are the same, as in the case above, only the CRC computation differs as shown below. This forces the use of 4 independent decoders, so that the CRC methods can be defined specifically for each sensor.

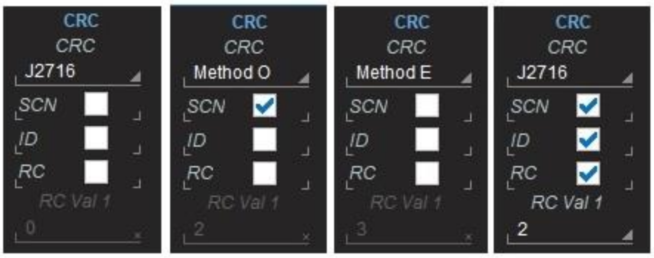


Figure 11 CRC interpretation parameters of each sensor, (decoder setup)

The resulting decoded values are identical, as shown by the quadruple plot of the 4 sensors' outputs.

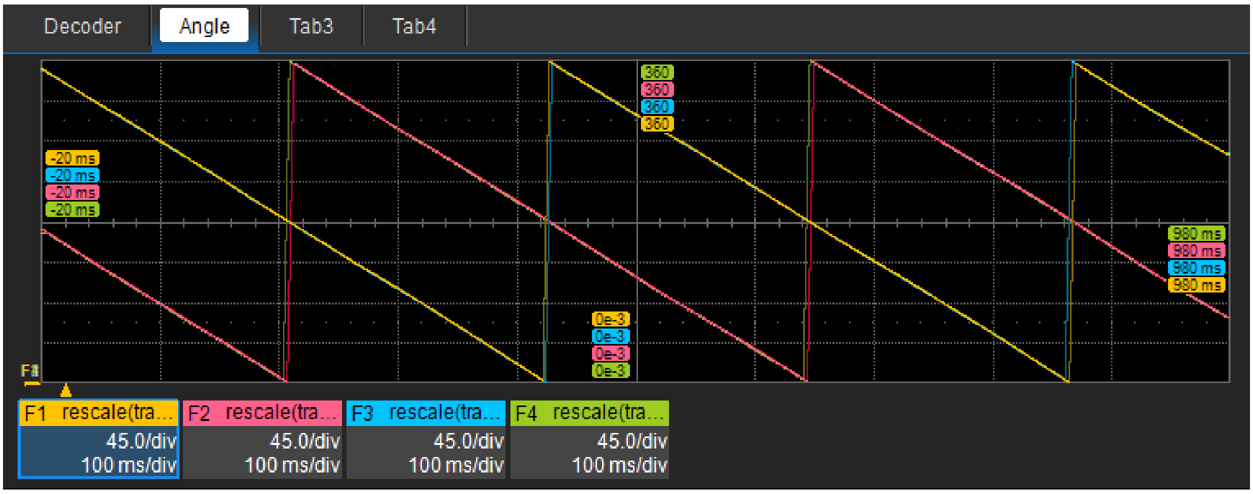


Figure 12 Angle graphs of each sensor, with 180° phase shift between odd and even traces F1-F4

Figure 13 Heterogeneous network, with staggered angle graphs and 4 independent decoders

Case 3: Detecting and correcting a parametrization problem on a sensor or on the ECU

The same setup is used as for previous experiment; however, the interrogation rate of the sensors was increased to reflect the growing need in industry for real time monitoring of physical behaviors. Each sensor transmits 361 messages/second. Using a round robin interrogation cadence, the total bus load is 1444 msg/sec. The following screen dump for the decode table shows the last lines of the decode table with the message count over 1 second

SENT SPC	Time	MTPulse	RespT	ID	Sync	Tick	Msg	Stat	b0	b1	b2	b3	D0	ID	Data	[illegible]
357	987.397 ms	97.42 µs	34.17 µs	3	79.61 µs	1.421 µs	Message 357: 9 Nibbles 1 Words	3	1	1	0	0	4e4			8
358	990.190 ms	97.37 µs	34.27 µs	3	79.50 µs	1.420 µs	Message 358: 9 Nibbles 1 Words	b	1	1	0	1	593			b
359	992.964 ms	97.37 µs	34.27 µs	3	79.50 µs	1.420 µs	Message 359: 9 Nibbles 1 Words	3	1	1	0	0	641			7
360	995.714 ms	97.32 µs	34.27 µs	3	79.61 µs	1.421 µs	Message 360: 9 Nibbles 1 Words	7	1	1	1	0	6f1			e
361	998.470 ms	97.35 µs	34.27 µs	3	79.50 µs	1.420 µs	Message 361: 9 Nibbles 1 Words	f	1	1	1	1	79f			0

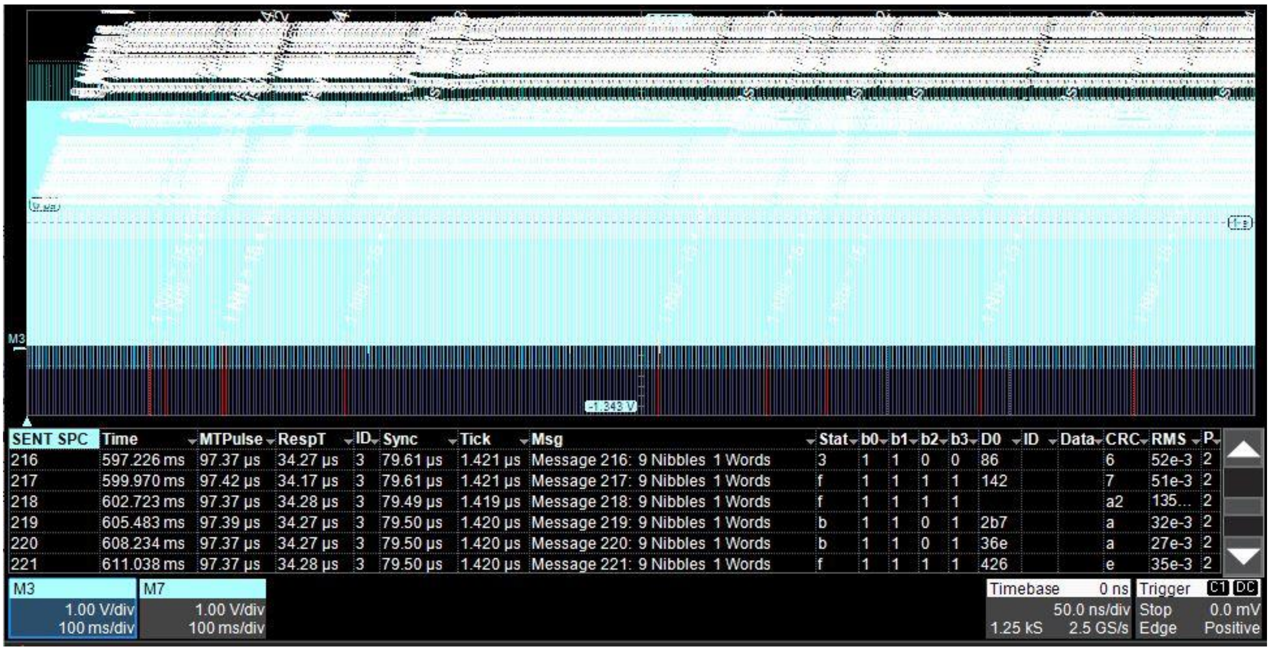
Figure 14 Total message count (for ID 3) over one second acquisition window

It stands to reason that when pushing the transmission speed on the system some undesired side effects could appear. It is the purpose of this section to analyze such a case.

For this experiment, a parametrization error was purposively introduced, and its effect was observed using the SPC decoder on the scope. The error is an early switch-back to high impedance state for ID1 (see SPC physical interface and frame description), hence occasionally truncating the last nibble (CRC) of the message and therefore causing a CRC error.

When working on such network the advanced indicators of a pathology are the occasional CRC errors. The image below shows these errors with subtle red vertical marks against the pale blue of the trace at the 100 ms/div time scale because the annotations are very compressed. As soon as the scale is decreased the vertical lines morph into rectangles with more details about the error. (not shown here)

In this case 6 nibbles (N1 through N6) are transmitted, but in some cases the last pulse delimiting the CRC nibble does not have enough time to be emitted, and hence causes the CRC error.



SENT SPC	Time	MTPulse	RespT	ID	Sync	Tick	Msg	Stat	b0	b1	b2	b3	D0	ID	Data	CRC	RMS	P
216	597.226 ms	97.37 µs	34.27 µs	3	79.61 µs	1.421 µs	Message 216: 9 Nibbles 1 Words	3	1	1	0	0	86			6	52e-3	2
217	599.970 ms	97.42 µs	34.17 µs	3	79.61 µs	1.421 µs	Message 217: 9 Nibbles 1 Words	f	1	1	1	1	142			7	51e-3	2
218	602.723 ms	97.37 µs	34.28 µs	3	79.49 µs	1.419 µs	Message 218: 9 Nibbles 1 Words	f	1	1	1	1				a2	135...	2
219	605.483 ms	97.39 µs	34.27 µs	3	79.50 µs	1.420 µs	Message 219: 9 Nibbles 1 Words	b	1	1	0	1	2b7			a	32e-3	2
220	608.234 ms	97.37 µs	34.27 µs	3	79.50 µs	1.420 µs	Message 220: 9 Nibbles 1 Words	b	1	1	0	1	36e			a	27e-3	2
221	611.038 ms	97.37 µs	34.28 µs	3	79.50 µs	1.420 µs	Message 221: 9 Nibbles 1 Words	f	1	1	1	1	426			e	35e-3	2

Figure 15 The decoder for Sensor ID 1 exhibits errors (vertical red lines in M3) at 100ms / division

The following screen dump shows time aligned good and bad SENT messages with the nibbles numbered and the missing nibble sketched over the annotation. The “good” messages have shorter nibbles then the “bad” messages, therefore leaving more time for the last nibble (CRC) to be transmitted.

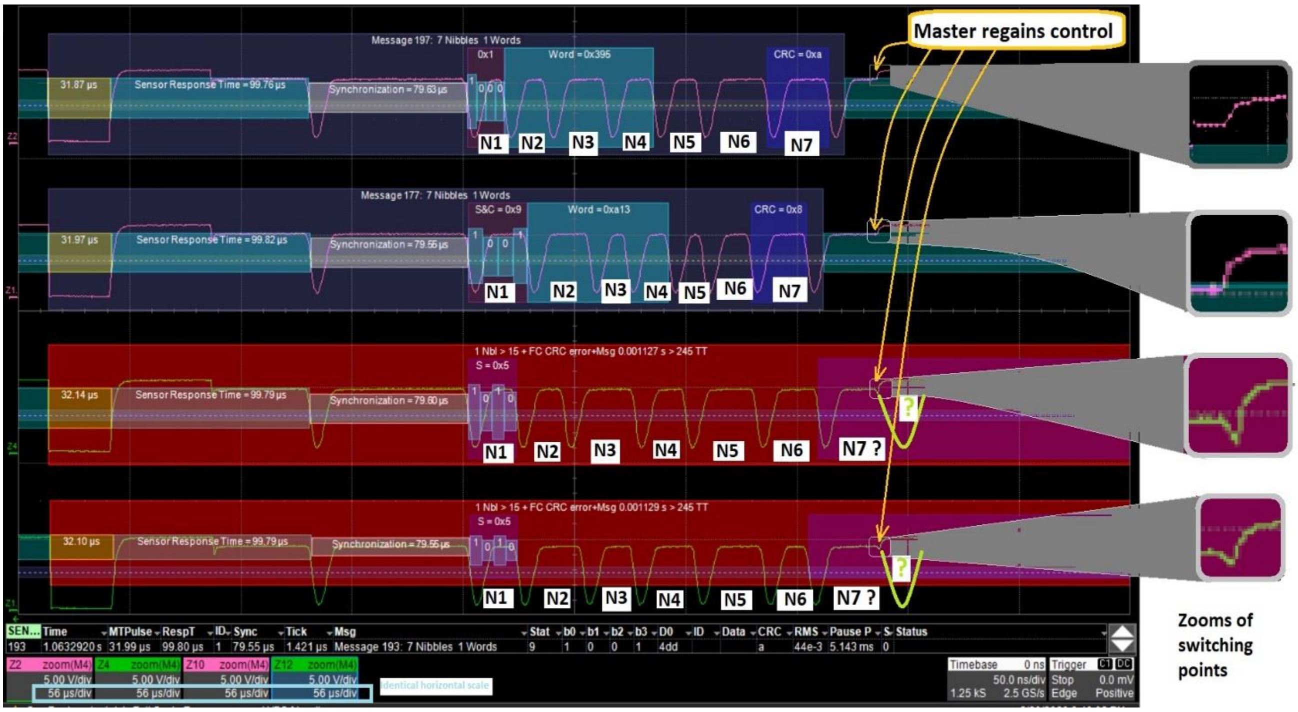


Figure 16 2 complete messages and 2 truncated messages within the 420 us window

Note that the detection of the end of the time slot is not automated and requires a careful observation of the signal. This is shown in the image below, in which every sample point appears. It is also important to note that this transition from pulse shaping to high impedance is specific to the pulse shaping mode (see chapter “SPC frame with pulse shaping”), and would not be present in “classic” open-drain mode.

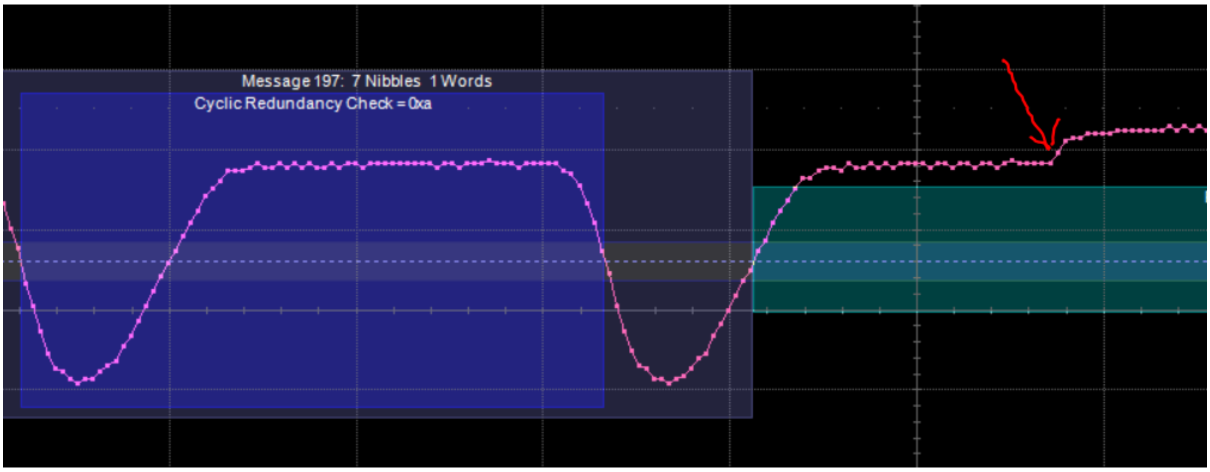


Figure 17 Detailed view of the switching point, when ECU regains control of the line

Once the above observations are confirmed, the observation setup can be simplified and made more efficient by focusing on the error cause. Since the errors only occur on sensor ID 1, the decoding can be limited to this sensor, while the decoders on ID 0,2,3 are turned off. This decreases the clutter on the screen and does not hide, skip or miss information since the other sensors are behaving correctly. The table view also becomes more compact and easier to examine, with more emphasis on the lines exhibiting the CRC error.

Normally, the timeslot available for any sensor should be calculated based on the sensor’s response time and the frame structure (how many nibbles), giving the maximum duration of the data transmission, after which the line can be released by the slave.

Based on this duration, the offending sensor is re-parametrized and the measurement re-done. The following image shows the measurements, before (Memory 3) and after (Memory 7) correction of the wrong sensor configuration.

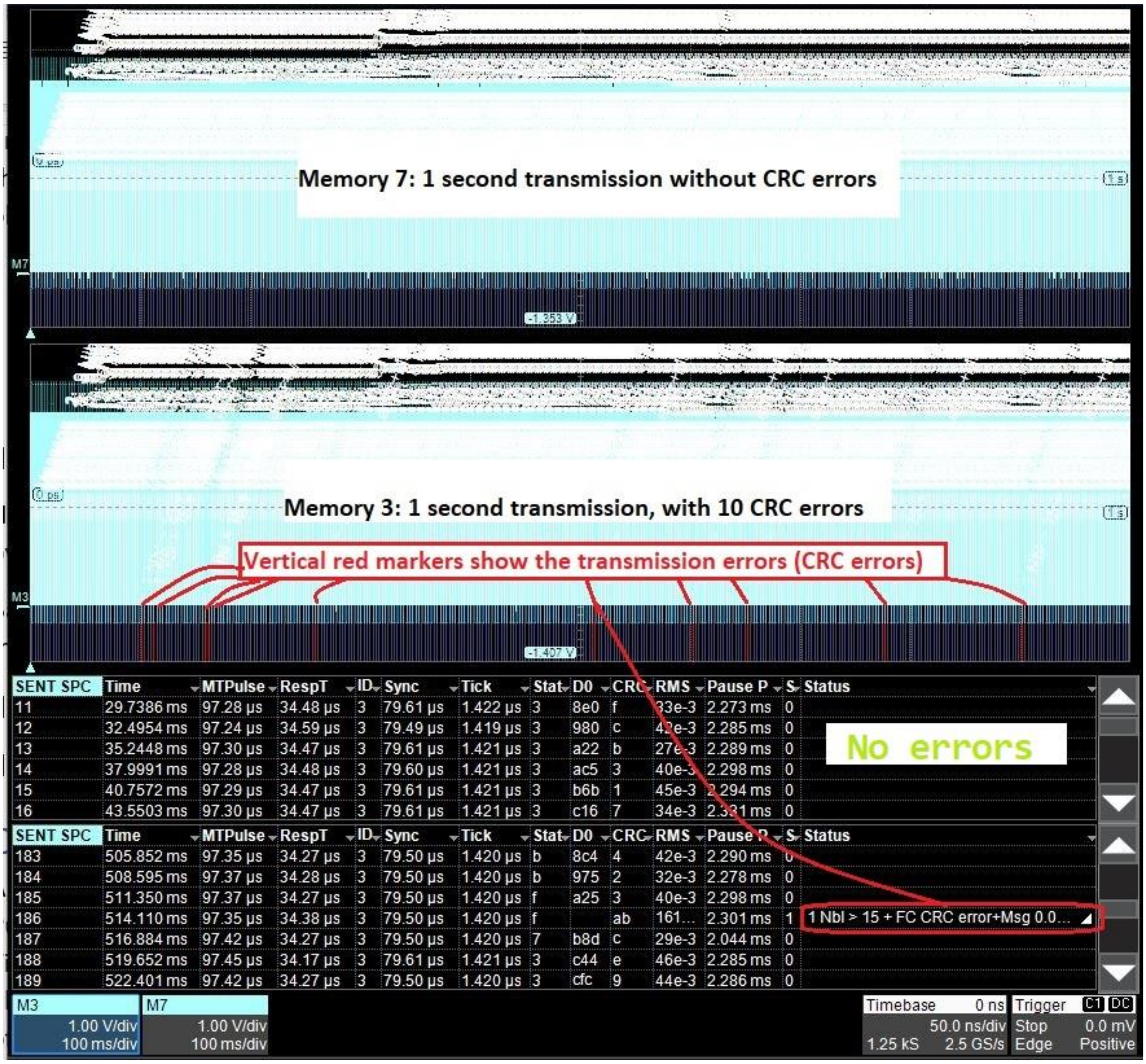


Figure 18 Comparison of transmission before and after increase of the time slot

The corrected transmission is now flawless, and no error appears at any point in time over the acquisition window of a full second.

Case 4: Verifying the response latency of the sensors on the bus

A SPC network expects the sensors to respond within a specified timeframe to the Master Trigger Pulse issued by the ECU, regardless of the length of the MTP. This latency is specified as a multiple of the Tick Time, typically between 80 and 100 Tick Time. The sensor's response latency is defined as the time elapsed between the falling edge of the MTP and the falling edge of the Sync Pulse. For example, for a TT of 3 us, using a factor of 90, the response latency of the sensors on the bus should be around 3 us x 90 = 270 us.

In this section we show how to verify the multiplication factor.

The concept relies on the direct measurement by the decoder on the signal of the:

- MTP length
- Time Between the Rising edge of the MTP to the falling edge of the Sync (RMFS)
- Tick Time

henceforth the desired ratio is expressed by Ratio = (MTP length + RMFS) / Tick Time

The following image shows the corresponding times and computations for one decoded SPC transaction

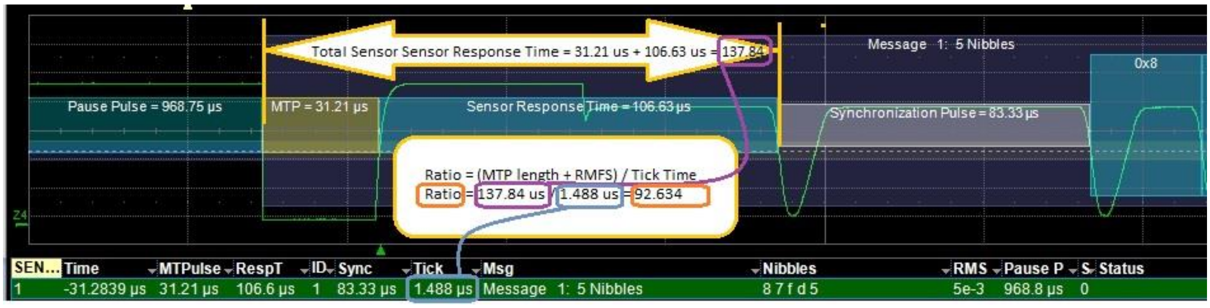


Figure 19 Constitution of sensor response time and ratio

The ratio needs to be computed for every transaction in the record, regardless if the record spans transactions to one or more sensors. The following image explains the computational flow.

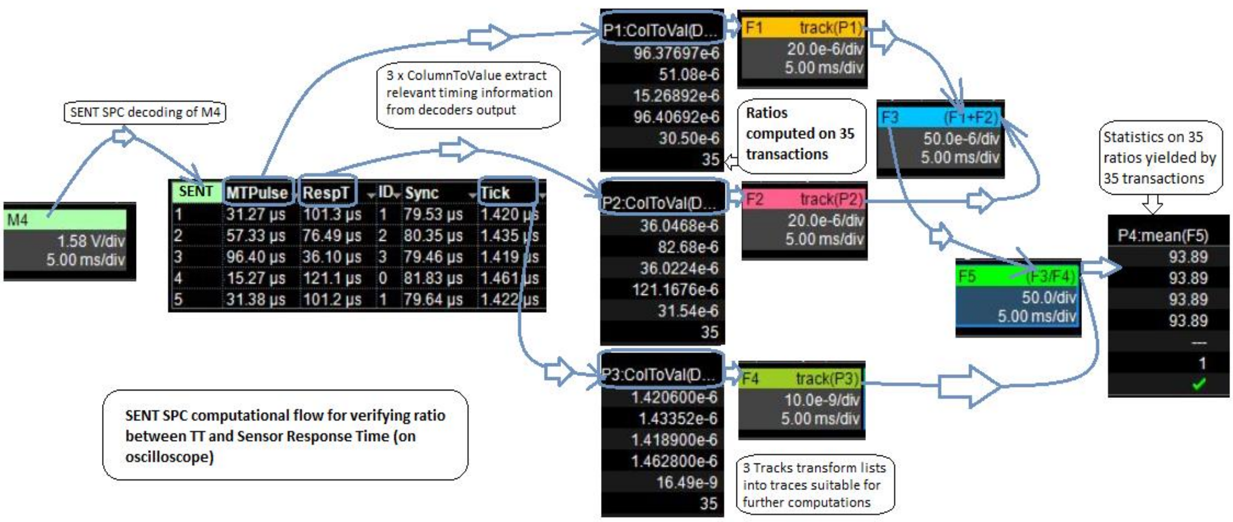


Figure 20 Computational flow needed to calculate Ratio

Case 5: Observing a network with uneven sensor interrogation rates

In this section we setup a system in which **Sensor 0 is interrogated more frequently** then sensors 1,2,3. This system architecture provides more data from sensor 0. It could be used when rotational speeds of sensor 0 is higher, or varying faster than those observed by sensors 1,2,3.

The zoomed image below visually captures the difference in interrogation rhythm. The bright annotated messages stand out against the dimmed traces M7 and M3. M7 is a measurement on a system operating at rhythm 012301230123 (as in previous cases) while M3 shows a rhythm of 010203010203.

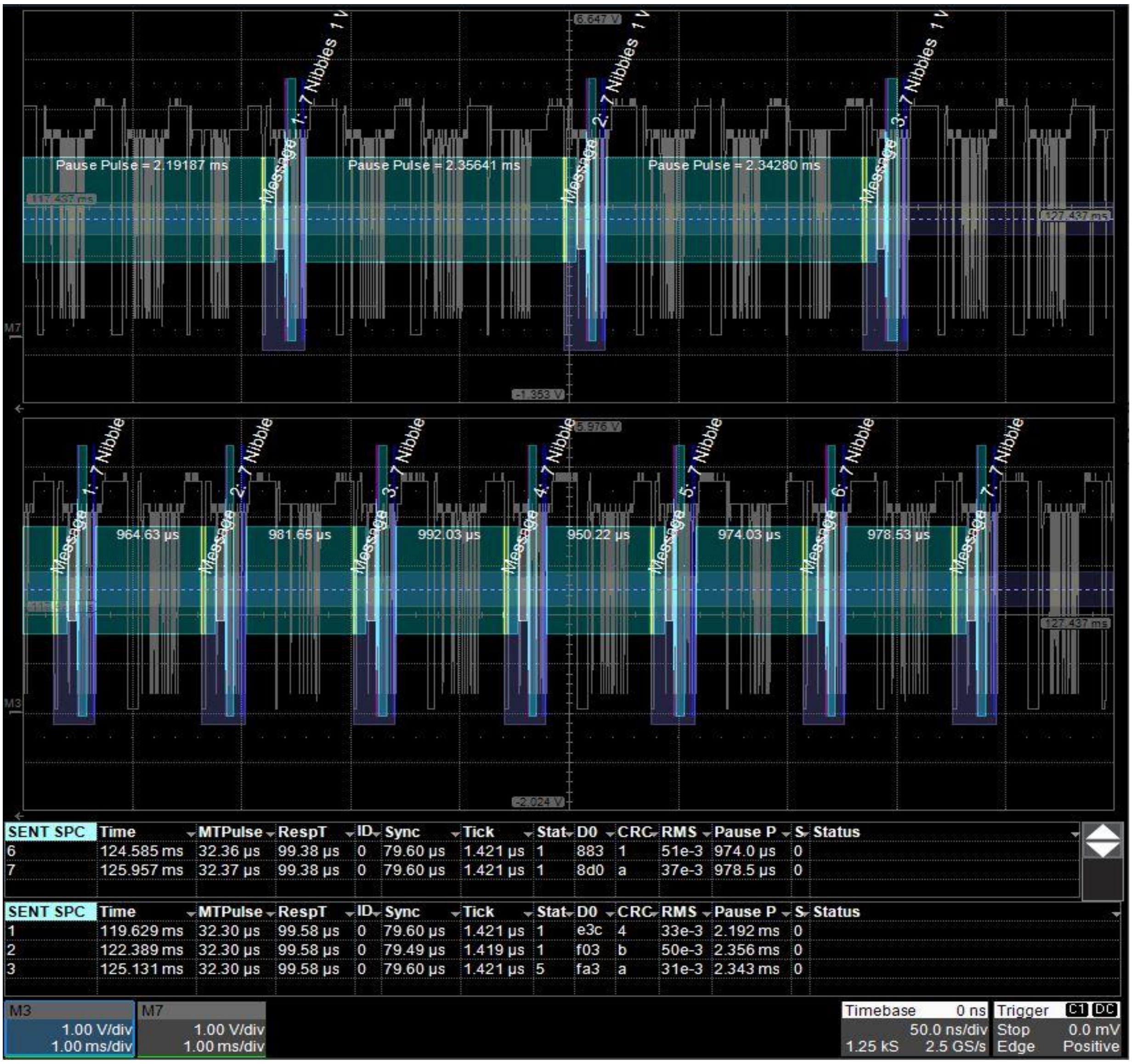


Figure 21 Visual observation of rhythm 01230123 vs 010203010203

In numerical terms, the following values result from the measurement:

Trace	Position	Average time between messages	Messages within 1 second
M7	Upper	2.77 ms	361
M3	Lower	1.38 ms	725

Conclusions

Thanks to the Coronavirus general slowdown, both authors had time to setup interesting experiments with a compact SPC network. The Melexis tool PTC-04 was used to configure the MLX90377 angle sensors connected to the control line, while a LeCroy oscilloscope with a SPC decoder is used to monitor the same network.

The experiments are based on stacked MLX90377 angle sensors, stimulated by a rotating magnet attached to a DC motor.

This paper shows how sensor transmission parameters (setup using the PTC-04) can easily be **verified** using the SPC enabled oscilloscope.

It is also demonstrated how transmission errors can be **detected** using the oscilloscope decoder and subsequently swiftly **repaired** by adjusting the sensor's configuration and finally **verified** using the decoder.

V15