Visitor to LOXTechnologies

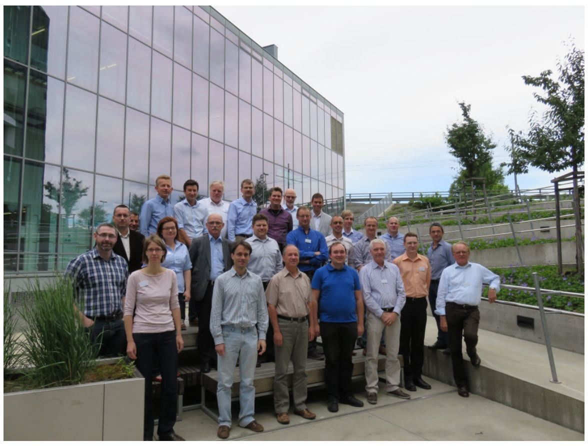
No. 3

Summer 2015

Fellowship of Sensors

LOX Technologies participates in fire safety consortium of European R&D entities

The main objective of the SAFESENS project is to obtain earlier and more reliable fire detection in buildings and support rescue worker brigades with vital information about occupants and fellow firemen. More specifically, the SAFESENS will enable early detection of fire by measuring multiple gases (e.g. CO_2, CO, volatile organic compounds) and increase in fire detection reliability by interpreting measured data with advanced sensor fusion algorithms. Integration of gas sensors with micro-bolometer-based presence detection technology will provide fire propagation and building occupancy maps for efficient evacuation. Wearable gas detectors and personal health monitors for rescue workers with inertial and radio frequency localization technologies should contribute to reduction in still alarming figure of fire-related life losses.

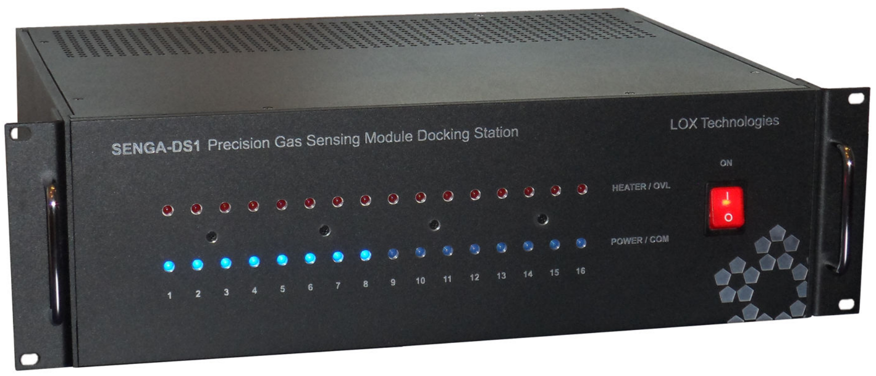


Following reception of Certificate of approval to conduct research and development from Slovak authority in 2013, LOX Technologies began collaboration within the SAFESENS consortium from its very beginnings in March 2014. Our primary domain is design and development of novel electronic circuitries for metal-oxide gas sensors, enabling a high sensor dynamic range (more than 6 orders of magnitude) and efficient operation, for targeting battery-operated sensors with low maintenance requirements. Within framework of Feasibility Study on Readout Approaches, we analyzed e.g. a novel circuitry based on composite constant voltage/constant current readout, which effectively doubles dynamic range of any of those methods. This approach have been then utilized in our design of precision gas sensor module, which was partially available as prototype in the late of 2014, so we were able to report practical properties of the CC/CV method at second consortium meeting in Cork in November of that year. Currently, the CC/CV method delivers over 7 orders of magnitude for the resistance sensing. An intelligent gas sensing module with digital interface is now readied to be available for utilization in first data collecting and demonstration setups.

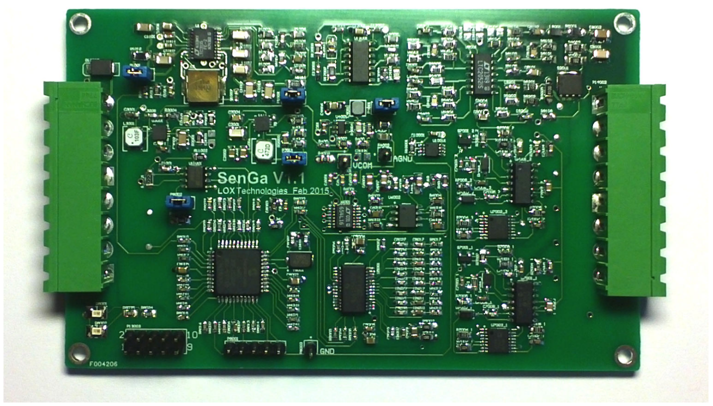
Boxed Precision

Delivery of first multi-channel precision gas sensing system for R&D purposes

In the late of 2014, LOX Technologies got an assignment to deliver laboratory equipment for multiple gas sensor readout for a research project run at Slovak University of Technology. Our approach resulted in development of scalable system comprising set of independent intelligent modules with digital interface, enabling PC-controlled measurement and data acquisition process. After six months of research, design and prototype fabrication, first SENGA Docking Station equipped with 8 gas sensor modules was delivered to the customer and demonstrated its superior features in precision gas sensor data acquisition.

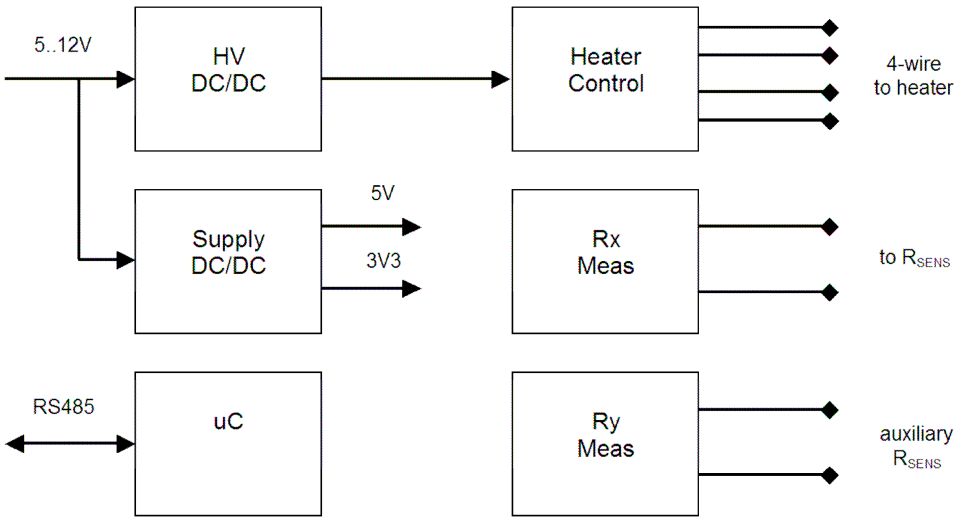


The intelligent SENGA modules come with state-of-the-art power-efficient boosting circuitry, enabling utilization of a wide range of gas sensors with up-to 25V, 10W power requirements at one side, or novel micro-heating elements with power requirements down to 10mW, on the other. Two independent resistance measurement inputs found at each module, which span more than 6 orders of magnitude, can be utilized for single or differential gas concentration measurements, as well as an independent feedback for heater or ambient temperature data. By default, the module is equipped with on-the-fly temperature measurement in the platinum heater circuitry utilizing DC current-voltage method. SENGA modules are equipped with RS485 industrial serial bus that enables connection of up-to 64 modules with the same host. To provide housing and docking for multiple modules, SENGA Docking Station in industrial 19" case, involving 16 module slots, integrated power supply and a USB/RS485 bus converter, has been readied.



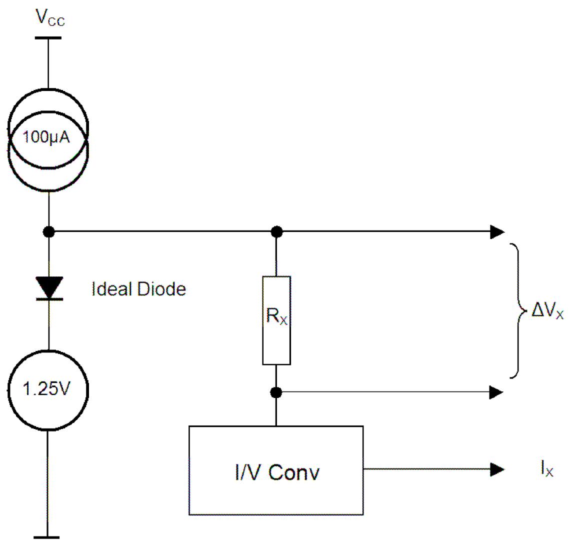
The module, schematically depicted in figure on next page, comprises controlled DC-DC converter and linear current controller as heater power source, two independent resistance measurement circuits, analog-digital frontend, 16-bit digital signal controller and diverse power management and communication circuits. The power unit that can supply up to 10W into heater located on the MOX gas sensor is equipped with 4-wire (Kelvin) heater resistance measurement, enabling heater temperature control with 1 degree Celsius precision. Analog acquisition is provided by fast-sampling, low noise 24-bit Delta-Sigma converter (ADS1256), while control voltages are generated in a 16-bit DAC with integrated low-drift voltage reference (LTC2654). Great attention is paid for proper grounding and decoupling of power supplies and digital part from sensitive analog circuitry.

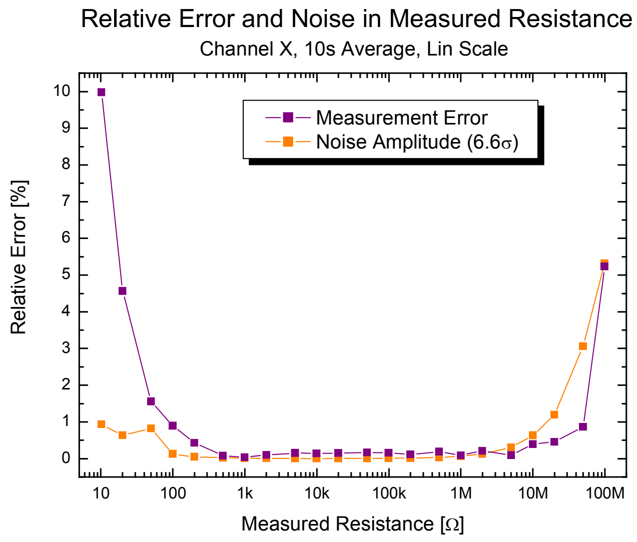
Digital signal controller from dsPIC family is responsible for data acquisition and filtration of ADC signals in digital filters down to 4Hz bandwidth, subsequent data processing, measurement process control and communication of the results. In the first 4-pole filter banks, which are operated at ADC acquisition speed of 400Hz per channel, signals are low-pass filtered to 16Hz corner frequency. The first stage filter is conceptualized as an effective 5-pole low-pass Butterworth with one 16Hz pole in analog RC filter circuitry



found at each of the eight ADC inputs. After the first-stage filtration, signals are re-sampled with 100Hz sampling clock and again filtered in the second stage 5-pole low-pass Butterworths (with coefficients equal to the first stage filter) down to 4Hz. The process ensures that the rest of noise over the Nyquist point is negligible in comparison to nominal circuit noise expected in the 4Hz range. Biquad architecture, i.e. a sequence of 1- and 2-pole units, has been utilized to implement the 4- and 5-pole filter banks. Signals in filters are treated as _Q32_16 (32 bit fractional format with LSB weight of 2-16) integer numbers, then converted to represent actual physical values into 32-bit float format.

SENGA large-span resistance measurement involves utilization of both constant current and constant voltage methods in a composite circuitry (figure right). For resistances lower as certain value, constant current source generates voltage difference across the unknown resistance, which is linearly proportional to its value. With increased resistance, at some point, voltage reaches clamping point of the constant voltage generator, which begins to sink current through the ideal diode. Thus, device switches into constant-voltage mode and useful information is obtained in measuring the resistance current, which is then inverse-proportional to the unknown resistance. In this way, span of measured resistance decades can be almost doubled.





Tests carried using a resistance decade showed the converter 1% error interval spans more than 5 orders of magnitude and performs well with 5% error in the 100MΩ limit. Average measurement error in tested range is 1.12%; maximum error in 5-decades range of 100Ω to 10MΩ is 0.9%. Accounting for error in testing equipment, the maximum measurement error in this range can be limited to 1.5%.

Firmware, Know Thyself

Descriptors enable embedded data reflection and self-generated user interfaces

Descriptors is technology for semi-automated generation of meta information structures for diverse data within a C/C++ application. While targeted into embedded domain, descriptors can be fruitfully utilized in any generic application, where data self-awareness is requested. In fact, they mimic some of self-descriptive properties found in modern higher-level languages, like Java of C#, while preserving standard C/C++ source code structure with no, or minimal, interference upon the original code functionality.

In descriptor-enabled application, primitive or structured data with static address get, when requested, a constant descriptor structure with all relevant information, including data type, name, format of its text interface, its structural relations etc. The descriptor can provide these meta information for any entity that requests it, e.g. for text or graphical input/output routines, for distant monitoring, for serialization of data, for logging etc. And, it provides it coherently, i.e. data properties utilized in different situations are controlled from singular point in source code.

This "point" is descriptor directive placed right before data declaration in code, presumably in header file of a C/C++ module. The scope of descriptor directive is up to the end of the current declaration, where information about type and name of the target data can be found. Commands can be assigned to the directive within its argument list, providing way to set meta-information not found in C declaration code, like format of text interface.

From the compiler point-of-view, the descriptor directive

__descriptor__(())

is fully transparent, i.e. the application, leaving descriptor structures aside, behaves exactly in the same way, as application without descriptor directives. This is achieved elegantly by providing a macro for standard C/C++ preprocessor which defines descriptor directive as an empty text.

Example of descriptorized code

```
__descriptor__(())
typedef enum EValues
{
    EVALUES_IDLE,
    EVALUES_RUN,
    EVALUES_STOP,
    EVALUES_GO,
    EVALUES_ERROR = -1
} TValues;

__descriptor__((  Format="%.2f",
                  Unit="[liter]"))
typedef float TVolume;

__descriptor__((  Unit="[EUR]",
                  Decimal=1,
                  DecWeight=-2))
typedef int TPrice;

TValues State = EVALUES_GO;
TVolume Volume = 1.23;
TPrice = 234;
```

Example of interface interrogation

```
State?
State = EVALUES_GO;
Volume?
Volume = 1.23 [liter]
Price?
Price = 2.34 [EUR]
```

Descriptor directives are processed before compilation, and even before standard preprocessing, by a special descriptor preprocessor. The preprocessor application takes all relevant project source code and generates C module with definition of descriptor structures. This module, together with additional interfacing module(s), can be attached into the project and bulk compiled into a target descriptor-enabled application which provides monitoring or text-interface functionality to all descriptorized data.

The key feature of a descriptor-enabled application is, that any change within data organization and properties is automatically aware for all methods which access these data thru descriptors, without need to adapt several parts of code. For example, without descriptors, when changing data type "int" to "float", any text interface within the project has to be adapted to this change, while for descriptor-enabled application, the change will propagate without need of programmer attention into descriptor structures and ultimately to the interface methods, as soon as descriptor preprocessing and consequent project building takes place. This feature, with access to extensive data monitoring in live application through a descriptor-based GUI, can lead to much shorter development times, compared with standard programming patterns which often create a non-transparent mesh of project-wide dependencies.

Because descriptor structures are constant data, the requirements for additional data RAM is minimal in the descriptorized version. This is crucial for embedded applications run on microcontrollers. Moreover, descriptors in non-volatile flash memories of microcontrollers cannot be corrupted by software malfunction and thus provide solid and reliable basis for elevated dependability of application key tasks.