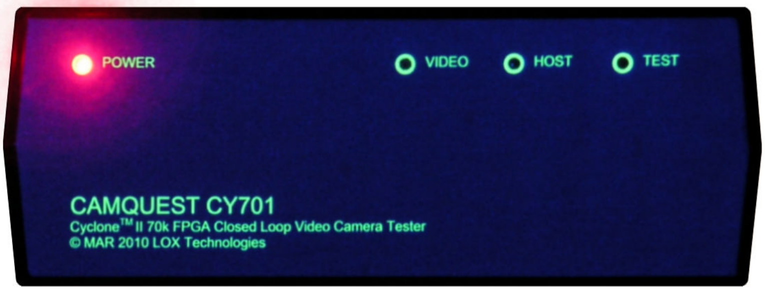
Visitor to LOXTechnologies

No. 2 Spring 2010

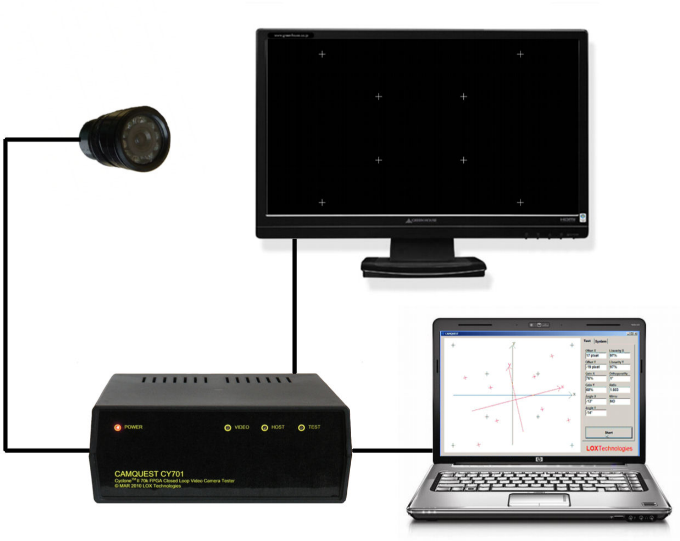
Quest for reliable vision

Camera tester for automotive industry incorporates powerful Cyclone II FPGA


POWER
VIDEO
HOST
TEST
CAMQUEST CY701
Cyclone™ II 70k FPGA Closed Loop Video Camera Tester
© MAR 2010 LOX Technologies

To address rising needs for testing equipment in automotive industry, LOX Technologies prototypes semi-autonomous closed loop camera tester. Christened CAMQUEST CY701, tester is equipped with ALTERA CycloneII EP2C70 FPGA chip with 70,000 cells, composite video input, VGA monitor output and various PC connectivity. The closed-loop testing system is based on generation of testing images onto LCD monitor in front of tested video camera, whose signal is then processed and evaluated in the CAMQUEST CY701 unit for possible camera skew or displacement. Results are then reported through RS232, USB or LAN interface to standard PC for graphic representation.

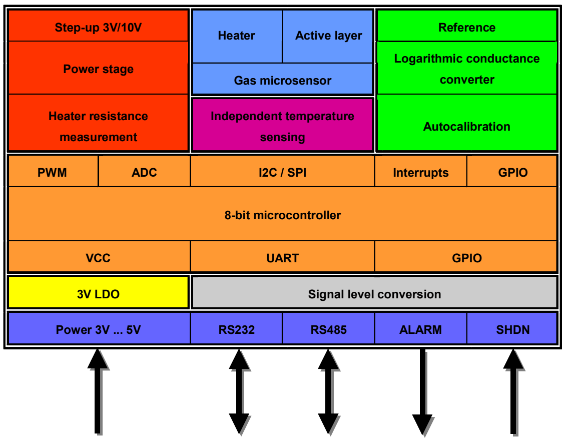
Because all image generation and processing takes place in the FPGA chip, the testing process is fast and there is no need for extensive processing resources on the PC site. In fact, the CAMQUEST CY701 unit can be utilized stand-alone, connected only to plant LAN network for testing process control and recording. ANALOG DEVICES ADV7180 in the video path autodetects and decodes composite analog $1V_{PP}$ NTSC/PAL/SECAM signals, which are then readied in 8-bit ITU656 format for further processing in the FPGA; however, FPGA can process digital LVDS video signals as camera input as well. In the output path, ADV7123 10-bit triple video DAC is utilized, with signals led to standard analog VGA monitor connector. Generated images and camera input are cross-correlated, and evaluated image offset, zoom, side ratio, linearity, arbitrary 360° skew, resulting from camera properties and placement, are graphically represented on the connected PC as alignment of two coordinate systems. Detection and evaluation of mirroring cameras, found in rear bumpers, is also provided.



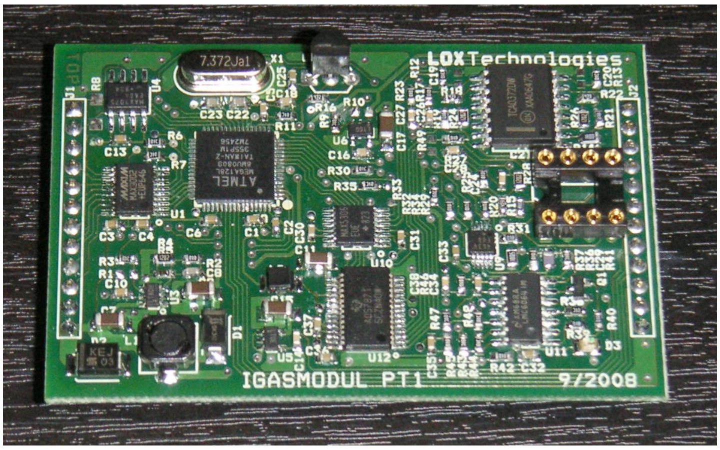
Sense and the infinity

Novel logarithmic converter debuts on an intelligent gas sensing module

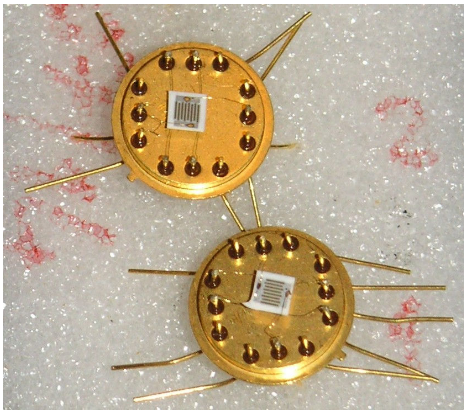
One of the essential part of LOX Technologies activity is research and development maintained by collaboration with leading scientific institutes in Slovakia. Since 1999, LOX Technologies has provided numbers of studies for Institute of Electrical Engineering of Slovak Academy of Sciences involving simulations and characterizations of novel MEMS (Micro-Electro-Mechanical System) devices. Then, starting in 2004, Faculty of Electrical Engineering of Slovak Technical University chose LOX Technologies as an associate company for R&D on novel gas sensing microsystem based on NiO depositions. The project evolved eventually in task to develop an Intelligent Gas Sensing Module with sensor supplied by the University and electronic module with its firmware supplied by our company. We are now proud to present the fruits of this collaboration.



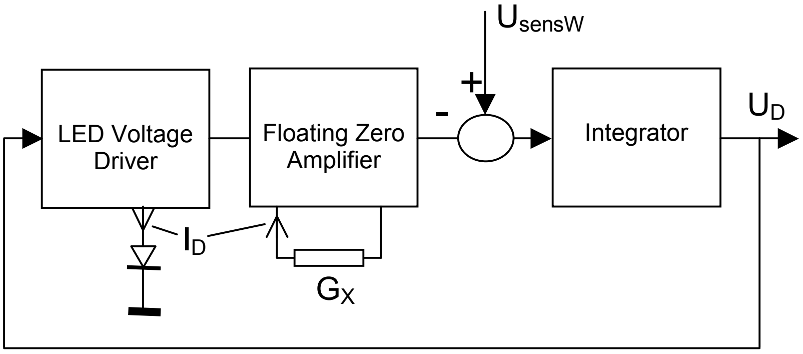
The heart of our IGASMODUL is gas microsensor based on NiO layers deposited on ceramic substrate with a platinum heater. In active state, the sensor is heated to temperatures as high as 300°C, when conductivity of the active NiO layer depends strongly on presence of target gasses. While the platinum heating element changes its conductivity by around 50% in the 20-300°C temperature span, which is utilized for measuring its actual temperature, the active sensing layer drops its resistivity from gigaohm-range in inactive state to several tens or hundreds ohms when heated and exposed to gas, i.e. the drop spans in more than 6 orders of magnitude. Clearly, a careful approach has to be taken in designing the sensor signal processing circuits, presumably a non-conventional one. We found a solution in utilizing an analog logarithmic converter connected to standard ADC, and with microcontroller-supported calibration in the analog path.



The digital and power subsystems on the IGASMODUL are also worth of mention. As seen on the schematic picture, the module is designed for 3,3V to 5V operation and involves step-up converter for supplying 10V, 0.1A for the sensor heating element. Heater power stage in bridge configuration is equipped with dual power operation amplifier TCA0372 with temperature and overcurrent protection, driven by complementary voltage values generated in MAX5306 12-bit serial DAC. Such circuitry enables various modes of operation: it can operate in linear mode as well as a PWM stage. Even in worst case, the heat dissipation in linear mode can be maintained by the power amplifier, thus linear operation can be sustained. However, we aim to set rough heater temperature in PWM mode and then, when measurements on sensor will take place, switch to linear mode in order not to disturb the sensible measurements. In such way, both power efficiency and converter precision can be preserved. Heating current is sensed at a resistor serial with the heater and is utilized in evaluating of both heating power and the platinum meander resistance, and hence the actual heater temperature. For calibrating the temperature dependence, an independent digital thermometer with one-wire bus is present on the module. The module, equipped with RS232 and RS485 communication interfaces as well as alarm and shut-down signals working with actual used-supplied digital levels, has a footprint of 32cm^2, which makes it ideal for integrating within deliberate target device.



Now look on the logarithmic converter found in IGASMODUL. As a non-linear element, a standard red LED is utilized, with well-known Shockley exponential dependence between current and applied voltage. When all the LED current flows through the unknown conductance, by applying constant voltage at the conductance, the LED current is proportional to the conductance and LED voltage is proportional to its logarithm, so the LED voltage is output of the logarithmic converter. However, there is no solution for direct sensing the LED voltage without loading it (note that Gx can go up to gigaohms). To cope with that, the converter operates rather in reverse mode and controls the LED voltage so that the voltage applied to unknown conductance is constant. LMC6064 CMOS precision quad opamp is utilized for the closed loop logarithmic converter with MAX4634 analog multiplexer switching between unknown conductance of the sensor active layer and three known resistors 1k, 100k and 10M for converter calibration.



Sharper sight

Effect of whitening on image cross-correlation performance

Often, multiple maxima can be found in cross-correlations of images with strong repetitive patterns, which can lead to ambiguities and deterioration of the image recognition algorithm performance. Suppose we have two images U, V with equal content that is only translated in the second image

$$U = C$$
$$V = T * C$$

where C denotes the image content, T is translation core and the operation in second line denotes convolution of image content with the translation. Note that autocorrelation function of second image is the same as for the first image independent of T

$$\mathrm{autocor}(V) = V^* * V = (T * C)^* * (T * C) = C^* * T^* * T * C = C^* * C = \mathrm{autocor}(U)$$

where factor $T^* * T$ means autocorrelation for simple translation that is equal to delta function, and thus could be omitted from the convolution. Performing simple image correlation, we get

$$\mathrm{crosscor}(U,V) = U^* * V = C^* * C * T\ ,$$

which is autocorrelation of image convoluted by T. So we can try to obtain the pure translation by deconvolution of the cross-correlation and autocorrelation

$$\tilde{T} = \mathrm{deconvol}[\mathrm{crosscor}(U,V), \mathrm{autocor}(U)],$$

which in frequency representation is

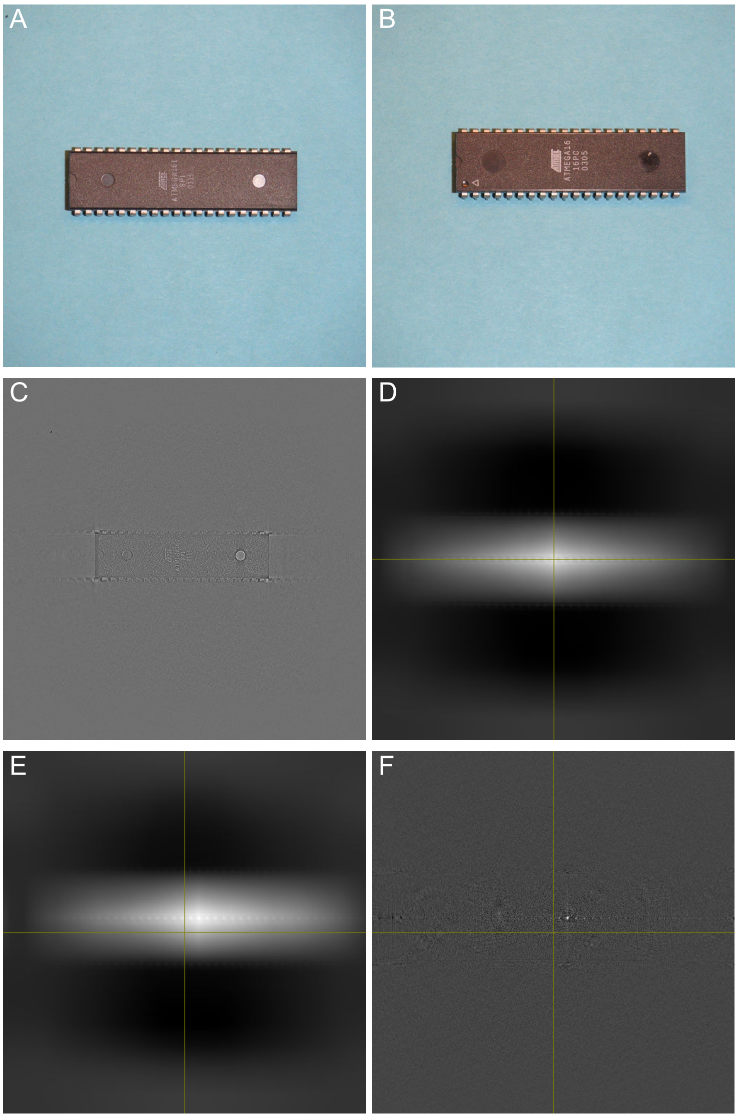
$$\tilde{T}(p,q) = \frac{U^*(p,q)V(p,q)}{U^*(p,q)U(p,q)}$$

with p,q denoting discrete spatial frequencies. Considering autocorrelations of both images are the same, we can write

$$\tilde{T}(p,q)=\frac{U^{*}(p,q)}{\sqrt{U^{*}(p,q)U(p,q)}}\,\frac{V(p,q)}{\sqrt{V^{*}(p,q)V(p,q)}}=\mathrm{crosscor}(U_{wh},V_{wh}),$$

where U_{wh},V_{wh} are "whitened" images, i.e. images with normalized magnitudes of complex coefficients in frequency domain.

We can observe the effect of image whitening on next illustrations. Pictures A and B are images of two integrated circuits with minor differences which show strong repetitive pattern due to inline signal pins ordered in regular distances. The repetitive pattern can be identified also on autocorrelation image D. Picture C depicts the whitened image of the first circuit. Simple cross-correlation of input images with suppressed DC bias can be found on picture E, and as anticipated, there are several poor maxima. The cross-correlation of whitened images is on picture F, where one sharp maximum clearly indicates sought translation offset. Example code in the inset utilizes VModules multivector manipulation library developed by LOX Technologies.



```
// init fourier transform
FFTX.init(NFFT);

// load images into real part of matrices
VMatrix<VComplexAtom> IM1A(NFFT,NFFT);
VMatrix<VComplexAtom> IM2A(NFFT,NFFT);

((VComplex<VMatrixAtom>)IM1A).real() = VBWImage("ic1c.bmp");
((VComplex<VMatrixAtom>)IM2A).real() = VBWImage("ic2c.bmp");

// fourier transform
VMatrix<VComplexAtom> IM1B(NFFT,NFFT);
VMatrix<VComplexAtom> IM2B(NFFT,NFFT);

IM1B = FFTX.direct2(IM1A);
IM2B = FFTX.direct2(IM2A);

// kill DC values
IM1B[0][0] = 0;
IM2B[0][0] = 0;

// power in frequency image
VMatrix<VComplexAtom> IM1C(IM1B,1);
VMatrix<VComplexAtom> IM2C(IM2B,1);

IM1C.inLine() << conj(IM1B.inLine()) * IM1B.inLine();
IM2C.inLine() << conj(IM2B.inLine()) * IM2B.inLine();

// whitening
VMatrix<VComplexAtom> IM1D(IM1B,1);
VMatrix<VComplexAtom> IM2D(IM2B,1);
VATOM eps = 1e-6;

IM1D.inLine() << IM1B.inLine() / sqrt(IM1C.inLine() + eps);
IM2D.inLine() << IM2B.inLine() / sqrt(IM2C.inLine() + eps);

// auto-correlation and cross-correlation
VMatrix<VComplexAtom> IM3D(NFFT,NFFT);
VMatrix<VComplexAtom> IM4D(NFFT,NFFT);

IM3D.inLine() << conj(IM1D.inLine()) * IM1D.inLine();
IM4D.inLine() << conj(IM1D.inLine()) * IM2D.inLine();

// inverse fourier transform
VMatrix<VComplexAtom> IM1E(NFFT,NFFT);
VMatrix<VComplexAtom> IM2E(NFFT,NFFT);
VMatrix<VComplexAtom> IM3E(NFFT,NFFT);
VMatrix<VComplexAtom> IM4E(NFFT,NFFT);

IM1E = FFTX.inverse2(IM1D);
IM2E = FFTX.inverse2(IM2D);
IM3E = FFTX.inverse2(IM3D);
IM4E = FFTX.inverse2(IM4D);

// convert real part to images
VBWImage IM1(((VComplex<VMatrixAtom>)IM1E).real(),1);
VBWImage IM2(((VComplex<VMatrixAtom>)IM2E).real(),1);
VBWImage IM3(((VComplex<VMatrixAtom>)IM3E).real(),1);
VBWImage IM4(((VComplex<VMatrixAtom>)IM4E).real(),1);

// normalize and save as bitmaps
IM1.inLine().setRange(0,1);
IM2.inLine().setRange(0,1);
IM3.inLine().setRange(0,1);
IM4.inLine().setRange(0,1);

IM1.saveAsBMP("im1.bmp");
IM2.saveAsBMP("im2.bmp");
IM3.saveAsBMP("im3.bmp");
IM4.saveAsBMP("im4.bmp");
```