Real Time Monitoring of 3 Axis Accelerometer using an FPGA Zynq®-7000 and Embedded Linux through Ethernet

Alfredo Jesus Perez-Castillo
Roberto Morales-Caporal
Tecnológico Nacional de México
Instituto Tecnológico de Apizaco
Division de Estudios de Postgrado e Investigación
Email: alfredjpc@icloud.com, rmcaporal@hotmail.com

Jose de Jesus Rangel-Magdaleno
Carlos Javier Morales-Perez
Instituto Nacional de Astrofísica Óptica y Electrónica
Departamento de Electrónica
Luis Enrique Erro 1, Tonantzintla, Puebla, Mex. C.P. 72840
Email: {jrangel, carlosj.morales}@inaoep.mx

***Abstract*—This paper introduces the use of a Field Programmable Gate Array (FPGA) with Zynq-7000 architecture and embedded Linux, as a digital tool to perform real-time remote monitoring of Internet-connected devices, using sensors that support SPI communication, the advantage of implementing the SPI protocol in the FPGA is that it can use several sensors that support this protocol even joining them at the same time. The incorporation of an embedded Linux operating system opens the possibilities of communication and execution of software as if it were a personal computer, as an example in this work, a 3-axis accelerometer is used. By integrating the above elements, a digital tool capable of covering the need for remote monitoring in real time is accomplished with a web server and connecting the FPGA through the Ethernet port by the TCP / IP protocol.**

***Keywords*—FPGA, Xillinux, SPI, ADXL345, ZYNQ-7000, Zedboard™.**

I. INTRODUCTION

The human being is currently in constant movement, therefore, the local and remotely interaction with the electrical and electronic devices it is relevant to minimize response time. Besides that, through numerous technology advances, society is moving towards an always connected paradigm [1]. It requires an interface between the device and the user that allows monitoring and, according to the needs, obtains specific values. This is very important to the fourth industrial revolution, known as Industry 4.0, brings Cyber-physical systems and Internet of Things (IoT) to industrial manufacturing systems [2], [3]. Furthermore, the number of interconnected physical devices will increase drastically, and they will continuously interact with local cloud services in order to act intelligently and flexibly [4].

In previous works [5]–[8] the remote monitoring solution has been implemented. However, it has been done using micro-controllers and even FPGA but without the architecture capable of installing an operating system or without the incorporation of communication Serial Peripheral Interface Bus (SPI). SPI is a synchronous serial data link standard named by Motorola that operates in full duplex mode. Sometimes SPI is called a ”four wire” serial bus. There is one master and one or more slave devices in the communication [9]. In addition, the advantage of this bus compared to I2C communication is that it offers a higher transmission speed.

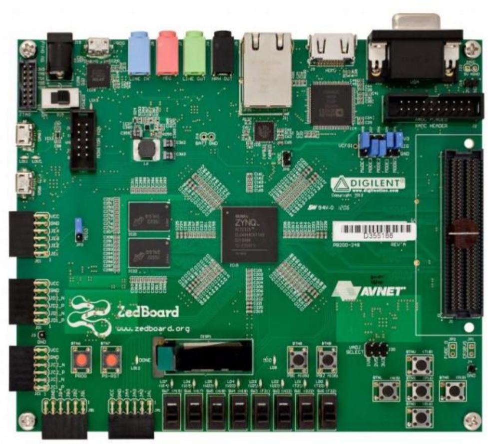


Fig. 1. ZedBoard™ FPGA [17].

Previous works have implemented the SPI communication protocol and have been done using microcontrollers such as LM358962 [12], ATmega32 [13], ESP32 [14], and also using FPGA for example a Spartan 2 [15], on the part of the microcontrollers are of fixed architecture and cannot be modify the hardware and in the cases of FPGA without architecture SoC are limited to a Bare Metal design, the contribution of this work is that by using the Zynq®-7000 chip offers the possibility of creating a own hardware design and install an embedded operating system (OS) and with the above create an environment similar to that of a personal computer, having an OS can create and execute programs made under high-level or low level programming languages, for example, in works

978-1-5386-7033-0/18/$31.00 ©2018 IEEE

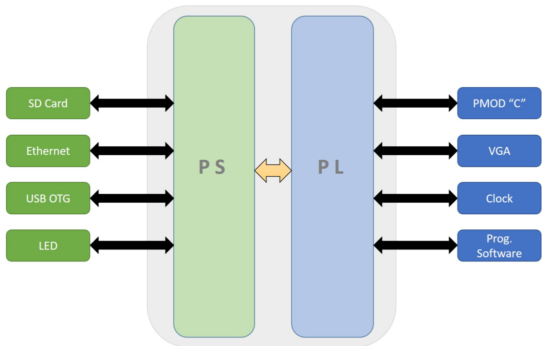


Fig. 2. PS and PL section used.

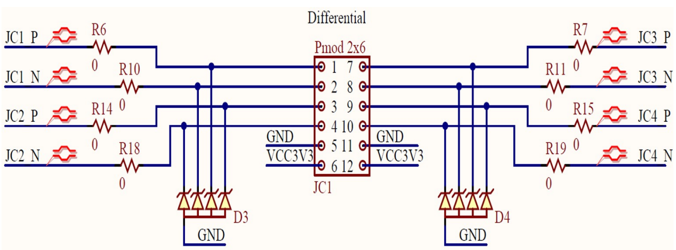


Fig. 3. PMOD "C" with LVDS [10]

related to the detection of motor failures [16], FPGA and adxl345 accelerometer have been used via SPI, however the processing of acquired signals is done offline using additional hardware and software, the main contribution of this work is that it can be performed the processing in the same FPGA and online with the operating system incorporated and SPI protocol. Therefore this work proposes the use of a Xilinx FPGA Zedboard™ development board with Zynq-7000 architecture, all programmable SoC [17] (see Fig.1), this architecture includes a Processing System (PS) and a Programmable Logic (PL) incorporating a Dual-Core ARM Cortex-A9 at 866 Mhz and an Artix-7 FPGA respectively, which allows installing an operating system (OS) and in the logical part the co-design of a specific hardware for the required interface. In this case is a 4-wire SPI communication to receive data from the triaxial accelerometer model ADXL345 [11]. The appearance of processors and FPGA of low cost and energy efficiency has made possible the realization of these applications through the joint design of hardware/software [18].

II. Hardware Co-design and OS

A. Hardware core Zedboard

The FPGA based on Zynq-7000 Zedboard model has two options of programming it, in a bare-metal way or using an operating system which uses the architecture that is defined in the design made in the proprietary software of Xilinx® (Vivado VHDL) (see Fig.2). By using the bare-metal type is limited to have only one specific application, however if an operating system is used it exists the possibility of install applications and run software as if it were a Personal Computer. In this case it was co-designed the base hardware already designed

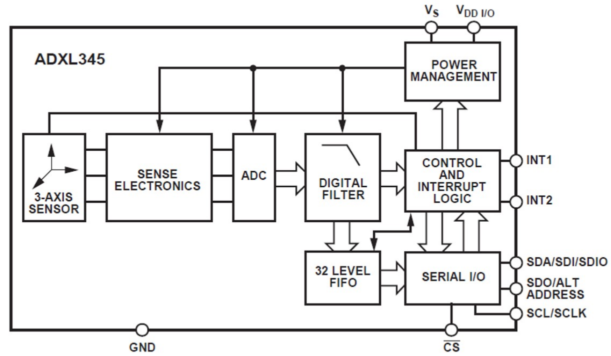


Fig. 4. Functional Block Diagram ADXL345 [11].

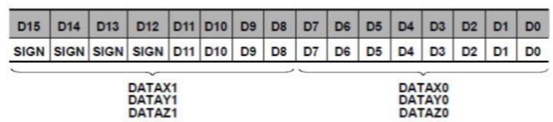


Fig. 5. Data Construction of ADXL345 [11].

by the company Xillybus and add the SPI interface to one of the 2 available PMOD™ (GPIO) of differential type in the part of the programmable logic that includes the FPGA [19]. The main feature of having an FPGA and a processor on the same chip is that the hardware can be changed and adapted to specific needs just replacing a file inside SD card using by Zedboard.

The necessary OS to complement the interface should be capable of covering local and remote connection needs, beside a program that links the sensor with the FPGA. Therefore a Linux distribution is proposed as an operating system, thus allows to install a web server to be able to access remotely and makes requests to the server [20], also an HTML website was created that contains Hypertext Preprocessor code (PHP) [21]. The application which controls the sensor can be written in high-level language (C, C++, Java, python, etc.), it is proposed to program above application in "C" language due to its high portability. Finally the Ethernet port was used 10/100/1000 PHY of the FPGA to communicate through TCP/IP. With the aforementioned, a digital tool is obtained that incorporates the necessary interface for the interaction between the device and the user, creating the possibility of applying said tool to varied applications.

Figure 2 shows the elements of the processing part as well as programmable logic that were used mainly. The Zedboard has 5 PMOD available, nonetheless only 2 (PMOD "C" and PMOD "D") use the low-voltage differential signal system or LVDS. The LVDS is a technology that was developed [22], [23] to provide a low-power, low-voltage alternative for high-speed point-to-point data transmission. Figure 3 shows the "C" PMOD with LVDS signal.

B. Vibrations Sensor

Microelectromechanical systems (MEMS) digital sensor ADXL345 is a small, thin, ultra-low power, 3-axis accelerom-

TABLE I
G RANGE SETTING

Setting		G Range
D1	D2	
0	0	$\pm 2g$
0	1	$\pm 4g$
1	0	$\pm 8g$
1	1	$\pm 16g$

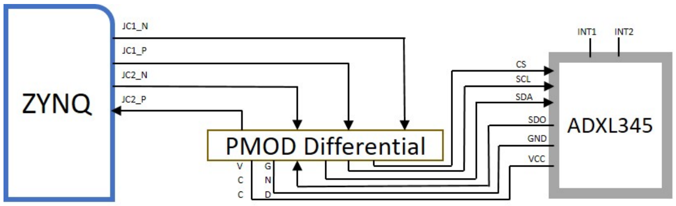


Fig. 6. SPI 4-wire to PMOD.

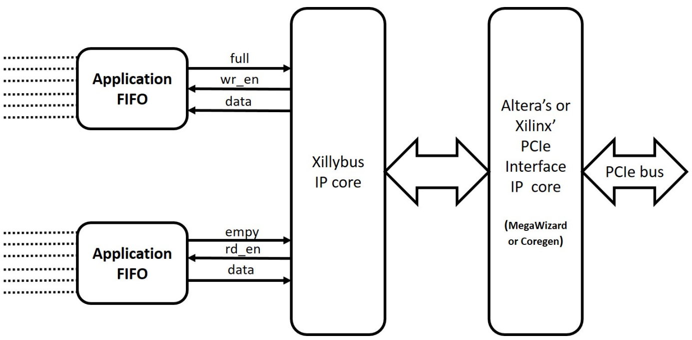


Fig. 7. Simplified FPGA block diagram of Xillybus using PCIe transport [23].

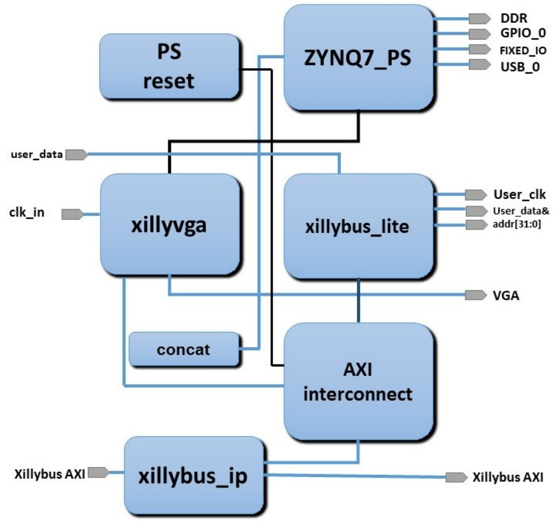
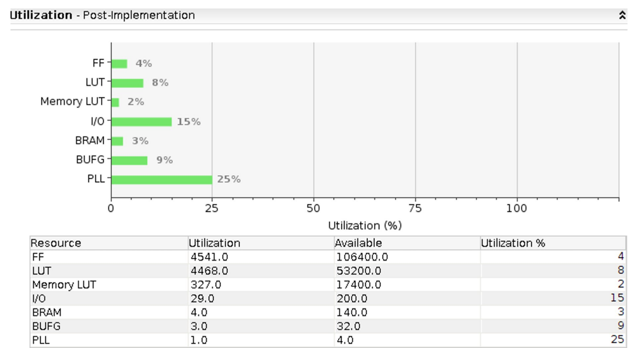


Fig. 8. Block diagram design in Vivado.



Resource	Utilization	Available	Utilization %
FF	4541.0	106400.0	4
LUT	4468.0	53200.0	8
Memory LUT	327.0	17400.0	2
I/O	29.0	200.0	15
BRAM	4.0	140.0	3
BUFG	3.0	32.0	9
PLL	1.0	4.0	25

Fig. 9. Utilization post-implementation report.

eter with high resolution (13-bit) measurement at up to 16 g. Digital output data is formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I2C digital interface. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (3.9 mg/LSB) enables measurement of inclination changes less than 1.0 degree, it has been used in works related to failure detection in induction motors [16], [24]. The range of the sensor is shown in table I, and functional block diagram in Figure 4.

The data format of the ADXL345 is 16 bits. Once acceleration data is acquired from data registers, the user must reconstruct the data. DATAX0 is the low byte register for X-axis acceleration and DATAX1 is the high byte register. In 13-bit mode, the upper 4 bits are sign bits (see Fig.5). When in 13-bit mode, 1 LSB represents about 3.9 mg.

Communication of the ADXL345 sensor with the PMOD ”C” of the Zedboard FPGA was through 4-wire SPI CS, SCL, SDA and SDO (see Fig.6) and that connects to ZYNQ by AB7, AB6, Y4 and AA4 General Purpose Input/Output (GPIO).

III. Installing OS and Creating Applications

A. Selecting OS

The FPGA Zedboard allows for its architecture installs an operating system, the selection will depend on the needs, there are several options such as Linux™, Windows® and even Android™. The distributions are specific depending on the board, are made to order, for the purpose Xillinux was used [25] which is a Linux distribution based on Ubuntu 12.04 version, the distribution includes a Zedboard base design (in Verilog or VHDL code) in addition to the opportunity to create a specific IP from Xillybus web portal, the IP created is appended to the IP core in Vivado, the main feature of Xillinux is that it works through FIFOS using the Peripheral Component Interconnect Express bus (PCIe), the developed applications are communicated through specific FIFO that can be of different sizes (8, 16 and 32 bits) the overall operation is shown in Figure 7.

B. Vivado Software Design

After hardware and OS selection, it proceeded to hardware design in Vivado [17] provided by Xillybus, which was attached to a shift register to achieve SPI communication and

```
//part1
Write(0x31, 0x00); // 2g
Write(0x2d, 0x08); // Start Measure

//part2
xdata0 = SpiRead(0x32); // Read X0
printf("%x", xdata0);
xdata1 = SpiRead(0x33); // Read X1
printf("%x ", xdata1);
ydata0 = SpiRead(0x34); // Read Y0
printf("%x", ydata0);
ydata1 = SpiRead(0x35); // Read Y1
printf("%x ", ydata1);
zdata0 = SpiRead(0x36); // Read Z0
printf("%x", zdata0);
zdata1 = SpiRead(0x37); // Read Z1
printf("%x\n ", zdata1);
```

Fig. 10. Extract program code.

TABLE II
ADXL345

Address	*Name*	*Features*	*Type*
0x00	DEV ID	bit[1:0]	R
0x2D	POWER CTL	bit[3]	RW
0x31	DATA FORMAT	bit[1:0]	RW
0x32	DATAX0	X0 value	R
0x33	DATAX1	X1 value	R
0x34	DATAY0	Y0 value	R
0x35	DATAY1	Y1 value	R
0x36	DATAZ0	Z0 value	R
0x37	DATAZ1	Z1 value	R

Register Map.

performed the synthesis and implementation thereof with the aim of generating the file necessary to start the FPGA with the Linux distribution. The block diagram design of the project in the Vivado software is shown in Figure 8. With the above, a file with a .bit extension has been created, which contains the design of the hardware for the programming of the FPGA when starting up. It should be noted that the SD card contains 2 partitions, one with Xillinux in EXT4 format and the other one in FAT format with the necessary files to start up the FPGA, the four necessary files are boot.bin, Devicetree.dtb, uImage and own design.bit, the use of the FPGA after the synthesis and implementation shown in Figure 9.

C. Server and Application to manage sensor

To achieve the remote monitoring of the sensor, it is proposed to install a Web Server in the FPGA to attend the remote requests of the clients, for this Apache Web Server was installed in the Ubuntu distribution making possible a web portal in HTML that executes scrips in PHP and these at the same time execute applications. The advantage of embedded operating system is that it can develop applications in the high or low level language that is required externally or directly in FPGA, in this case the application was made in C language because it is a transportable and transparent language, Shell Script to execute commands automatically and PHP that is included within the HTML markup language. The application to activate the ADXL345 sensor must send, according to the needs, the hexadecimal values that are defined in the ADXL345 sensor shown in table II. On the other hand, it can see in Figure 10 in the code extract in the language "C" in part 1 the instructions are sent for the ADXL345 sensor to start the measurement and also to be set to 2G (it can be another range), in part 2 of the code we observe where the values 0 and 1 of XYZ are requested according to Table II.

The overall interaction of the ADXL345 sensor with Xillybus and the FPGA is described in the diagram of Figure 11.

D. Web page executing PHP

The main web page hosted on the server contains PHP code within its structure, which has the purpose of executing specific applications (see Fig.12). The above is very useful because the user can executes applications or even scripts by clicking on the button inside the web page.

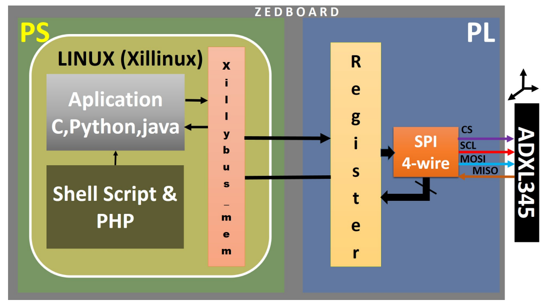


Fig. 11. Interaction between Zedboard and Xillybus.

IV. Measurements

A. Local

Locally to the Zedboard card the user can connect a monitor and keyboard/mouse devices through the USB On-The-Go port (OTG) to interact as if it were a Personal Computer, in this way the user can run the scheduled applications and visualize in real time how the sensor values go printed on the screen. On the other hand, if it is required, the user could use graphic libraries such as Opencv (Open Computer Vision) [26] (see Fig.13).

B. Remote

Access remotely is the most useful since there is a possibility of accessing the system from any site through the Internet, as mentioned in the previous section a web portal was built that hosted on a web server that offers the possibility of interact remotely performing the measurements and obtaining them in RAW form or if the user prefers also download it in a comma separated file format (CSV) to graph or manipulate it more easily. The diagram in Figure 14 shows graphically the operation and interaction of the FPGA card with the installed server and interacting with the clients, and in the Figure 15 shown the measurement of SPI signals in oscilloscope. The integration of the system and its operation is shown in the Figure 16, where it is observed that the FPGA is making measurements locally with directly connected peripherals and remotely through the TCP / IP protocol, following the requests of the web clients.

```
HTML
< a href= "code.php"
class="button">button</a>
        PHP
        <?php
        $output = shell_exec("./");
        ?>
```

Fig. 12. HTML code with PHP inside.

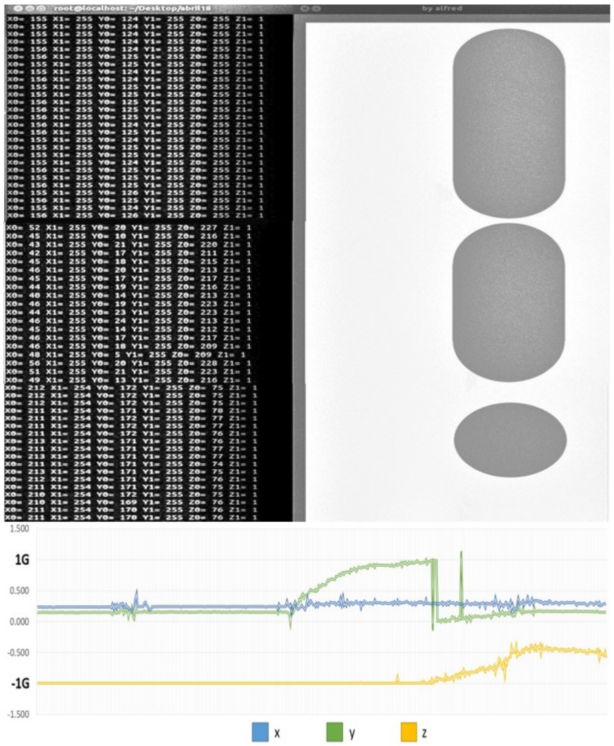


Fig. 13. Sensor values on the left and on the right side of a graph generated by the OpenCV library reacted to X values, and the corresponding graph

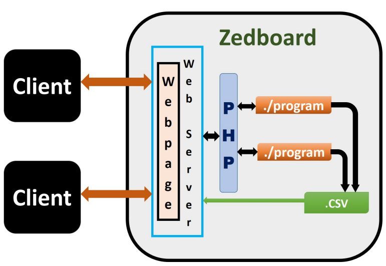


Fig. 14. Client-server model implement.

V. Conclusion

The integration of the Zynq 7000 FPGA with an embedded Linux-based system, as well as the enabling of a PMOD with LVDS and SPI communication, resulted in a powerful digital tool of great relevance due to besides of ADXL345 sensor used in this work. It can be used more sensors that accept SPI communication even having two or more sensors connected at the same time, plus the advantage of an embedded operating system is that the user can creates custom software

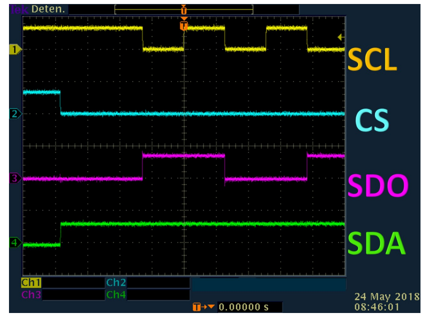


Fig. 15. Measurement of CS, SCL, SDO and SDA signals of the SPI in an oscilloscope.

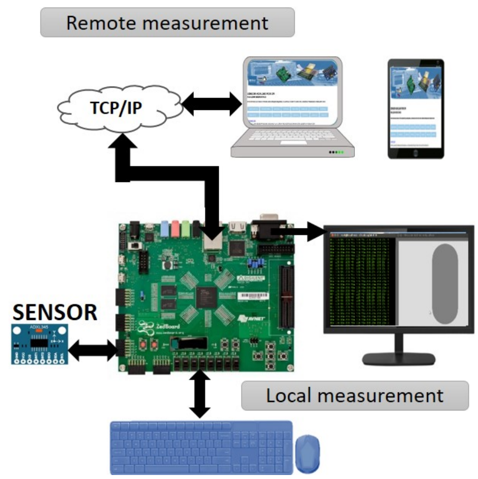


Fig. 16. Integrated system and running.

to manipulate the data received through the PMOD containing the SPI communication using applications made in high level languages. Furthermore it can be installed a web server and be able to perform remote monitoring, thus, it can reuse all the hardware design or even add additional hardware made in Vivado, as well as the web portal already installed in the FPGA, and it would only be necessary a new application that has the necessary parameters to activate another type of sensor and manipulate the data according to the needs. In future work this digital tool will be integrated to the monitoring of vibration signals in induction motors.

Acknowledgment

The first author wants to thanks to Consejo Nacional de Ciencia y Tecnología (CONACYT, México) for the support to the master degree studies in the Technological Institute of Apizaco and to National Institute for Astrophysics, Optics and Electronics (INAOE, México) for allow the technical residence and for all the support provided.

References

[1] L. Coetzee and J. Eksteen, "The internet of things - promise for the future? an introduction," in *2011 IST-Africa Conference Proceedings*, May 2011, pp. 1–9.

[2] R. Drath and A. Horch, “Industrie 4.0: Hit or hype? [industry forum],” *IEEE Industrial Electronics Magazine*, vol. 8, no. 2, pp. 56–58, June 2014.

[3] N. Jazdi, “Cyber physical systems in the context of industry 4.0,” in *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*, May 2014, pp. 1–4.

[4] A. E. Kalor, R. Guillaume, J. J. Nielsen, A. Mueller, and P. Popovski, “Network slicing in industry 4.0 applications: Abstraction methods and end-to-end analysis,” *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2018.

[5] Y. Zeng, L. Liu, and Y. Yu, “Design of security positioning and remote network server monitoring terminal,” in *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, Oct 2017, pp. 2026–2030.

[6] J. Saha, A. K. Saha, A. Chatterjee, S. Agrawal, A. Saha, A. Kar, and H. N. Saha, “Advanced iot based combined remote health monitoring, home automation and alarm system,” in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan 2018, pp. 602–606.

[7] Z. Jinxiang and S. Chuanwei, “Research and design of remote data acquisition system based on cc3200,” in *2017 International Conference on Robotics and Automation Sciences (ICRAS)*, Aug 2017, pp. 158–161.

[8] Z. Siyu, M. Jun, Z. Jianyong, and M. Fei, “Research on a multilayer distributed online monitoring system of 10kv high voltage switchgear,” in *2013 Fourth International Conference on Digital Manufacturing Automation*, June 2013, pp. 1025–1028.

[9] A. Szekacs, T. Szakaill, and Z. Hegykozi, “Realising the spi communication in a multiprocessor system,” in *2007 5th International Symposium on Intelligent Systems and Informatics*, Aug 2007, pp. 213–216.

[10] Zedboard, “Schematic prints(”all documents”, logical),” urlhttp://zedboard.org, 2018.

[11] A. Devices, “3-axis, 2 g/4 g/8 g/16 g digital accelerometer data sheet adxl345.” urlhttp://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf, 2018.

[12] Z. Ruimei and W. Mei, “Design of arm-based embedded ethernet interface,” in *2010 2nd International Conference on Computer Engineering and Technology*, vol. 4, April 2010, pp. V4–268–V4–270.

[13] W. Zhang, “The design of atmega32 mcu spi and max191 serial a/d communication,” in *2011 International Conference on Business Management and Electronic Information*, vol. 2, May 2011, pp. 526–528.

[14] I. Allafi and T. Iqbal, “Design and implementation of a low cost web server using esp32 for real-time photovoltaic system monitoring,” in *2017 IEEE Electrical Power and Energy Conference (EPEC)*, Oct 2017, pp. 1–5.

[15] S. Saha, M. A. Rahman, and A. Thakur, “Design and implementation of spi bus protocol with built-in-self-test capability over fpga,” in *2014 International Conference on Electrical Engineering and Information Communication Technology*, April 2014, pp. 1–6.

[16] C. Morales-Perez, J. Rangel-Magdaleno, H. Peregrina-Barreto, J. Ramirez-Cortes, and I. Cruz-Vega, “Fpga-based broken bar detection on im using omp algorithm,” in *2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, May 2017, pp. 1–6.

[17] X. inc., “Xilinx - adaptable, inteligent.” urlhttp://www.xilinx.com, 2018.

[18] G. D. Michell and R. K. Gupta, “Hardware/software co-design,” *Proceedings of the IEEE*, vol. 85, no. 3, pp. 349–365, Mar 1997.

[19] AVNET, “Hardware users guide,” urlhttp://zedboard.org /sites/default/files/documentations/ZedBoard_HW_UG_v2_2.pdf, 2014.

[20] Apache, “The apache software fundation,” urlhttps://www.apache.org/, 2018.

[21] T. P. Group, “The php group,” urlhttp://www.php.net/, 2018.

[22] Q. Cabanes and B. Senouci, “Objects detection and recognition in smart vehicle applications: Point cloud based approach,” in *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, July 2017, pp. 287–289.

[23] Xilinx, “Zynq-7000 soc technical reference manual,” urlhttps://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf, 2018.

[24] C. Morales-Perez, J. Rangel-Magdaleno, H. Peregrina-Barreto, and J. Ramirez-Cortes, “Half-broken rotor bar detection on im by using sparse representation under different load conditions,” in *2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, Nov 2017, pp. 1–5.

[25] Xillybus, “Xillybus. ip cores and desing service,” urlhttp://xillybus.com/, 2018.

[26] O. team, “Opencv library,” urlhttps://opencv.org/, 2018.