International Journal of Computer Networks and Applications (IJCNA)
Volume 2, Issue 6, November – December (2015)

RESEARCH ARTICLE

Wireless Sensor Network-Based Health Monitoring System for the Elderly and Disabled

Gurkan Tuna
Department of Computer Programming, Trakya University, 22020, Edirne, Turkey
gurkantuna@trakya.edu.tr

Resul Das
Department of Software Engineering, Firat University, 23119, Elazig, Turkey
rdas@firat.edu.tr

Ayse Tuna
School of Foreign Languages, Trakya University, 22030, Edirne, Turkey
aysetuna@trakya.edu.tr

Abstract –Even if the elderly and disabled need the assistance of their families, parents, and healthcare providers, they prefer to live in their homes instead of assisted-living centers. Therefore, their health and activities must be remotely monitored so that in case of an urgent unexpected situation, immediate help can be provided. In this respect, this paper proposes a wireless sensor network-based health monitoring system for the elderly and disabled, and focuses on its development steps. The proposed system is composed of low-cost off-the-shelf components and enables the monitoring of important health parameters of the elderly and disabled. Since it is a wireless and portable health monitoring solution, it can be a valuable remote monitoring tool for health care service providers by reducing the cost of their services. It can be combined with data mining solutions and/or machine learning techniques to offer novel features such as pattern extraction and behavior analysis.

Index Terms – Wireless sensor network, health monitoring, the elderly, the disabled, intelligent monitoring.

1. INTRODUCTION

Although the elderly and disabled generally prefer living in their homes, their health and activities must be continuously monitored so that in case of an urgent case, help can be provided immediately. Basically, smart homes are technologically advanced homes developed to enable domestic task automation, easier communication, and higher security. Since they have been geared to provide special needs of the elderly and disabled, they can enhance the lives of the elderly and disabled by allowing them to stay in their homes where they feel comfortable. At the same time, they can help caregivers to improve the quality of the provided services.

In parallel with the increase in the population of the elderly and disabled, the importance of smart home services has increased [1-3] and a number of wearable monitoring systems have been developed [4-6].

The ultimate goal of this study is to build a monitoring system capable of monitoring one or more person(s) during his/their daily activities at home. The proposed system consists of a main node and a group of distributed nodes and in this respect its nodes form a wireless sensor network (WSN) [7-11]. The remaining of the paper is as follows. Section 2 presents the details of the proposed WSN-based health monitoring system. Section 3 presents an evaluation study for the proposed system. Finally, the paper is concluded in Section 4.

2. WIRELESS SENSOR NETWORKS FOR HEALTH MONITORING

In this section, the details of our approach are presented.

2.1 Hardware description

Our wireless sensor network (WSN) health monitoring system consists of two kinds of wireless units: the “coach terminal” device and the “sensors’ node” devices. One sensors’ node is attached on each one of the monitored subjects. This is responsible for reading different health indicating sensors and sending wirelessly the accumulated data to the coach terminal. The coach terminal is the interface between the sensors’ nodes and the personal computer (PC) where the received data are stored and processed. The block diagram of the monitoring system is shown in Figure 1. The specific monitoring system is designed based on the Arduino Uno [12-14], a ubiquitous open-source hardware (HW) and software (SW) development platform. This single-board processor platform provides a low cost solution, however powerful enough to monitor the data required for this project.

RESEARCH ARTICLE

The processor platform: Arduino Uno is built around an ATmega328P microcontroller (MCU), an 8-bit RISC architecture processor available from Atmel. The main features that are of interest on this particular MCU are: (i) 32KB of flash memory, (ii) 2KB of SRAM memory, (iii) 16MHz clock speed, (iv) 6 channel 10bit analog to digital converter (ADC) at 76.9 kSPS, (v) 2-wire serial interface (TWI), (vi) programmable serial interface (USART). According to the first three characteristics, there are enough processing power and memory in order for one to design a real-time health monitoring system. The ADC peripheral is of key importance for this project, as most of the monitored health indicators we will be in the form of analog signals. As a result, these have to be digitized before being transmitted to the coordinator MCU for further processing. With its 10-bit resolution and high sampling rate we are able to acquire data reliably from the monitored athlete. The TWI peripheral is a serial interface compatible with the widely used I2C bus. This bus is used mainly for communication with sensors that have digital output. Finally, the programmable serial interface is used for communication between the nodes and the main server. The RF modules that are used have a serial communication interface. Additionally, the USART interface is used for data transmissions from the coordinator to the main PC. This is implemented by means of a secondary MCU dedicated to the USB to serial conversion.

The sensor platform: The sensors that are employed for the purposes of this project are the following: (i) tri-axial accelerometer via the TWI interface, (ii) electrocardiography sensor (ECG) via the ADC, (iii) airflow sensor via the ADC, (iv) temperature sensor via the ADC, (v) galvanic skin response sensor (GSR) via the ADC.

The communication module: Every node of the system has to be able to transmit the acquired health indicating signals back to a main server for processing. As a result, every wireless node is equipped with a radio module which implements the wireless communication with the server. The module we are using is the Xbee PRO S2B [15]. This module implements the ZigBee protocol which is a communication protocol known for its high performance on multi-node sensor networks; it has low power consumption, high range (about 90 meters in indoor or urban areas) and high data throughput (up to 35kbps). The employed Xbee PRO S2B module has a serial communication interface which is supported by the USART peripheral on our MCU.The block diagram of a single sensors' node is shown in Figure 2, while Figure 3 presents a photograph of a fully wired functional sensors' node.

Considering the project's requirements, specifically in terms of processing power and main memory, Arduino Uno is one of the best low-cost solutions. Arduino Uno has enough processing power and memory in order for one to design a real-time health monitoring system. Basically, it is built around an ATmega328P microcontroller (MCU), an 8-bit RISC architecture processor available from Atmel. The main features that are of interest on this particular MCU are: (i) 32KB of flash memory, (ii) 2KB of SRAM memory, (iii) 16MHz clock speed, (iv) 6 channel 10bit analog to digital converter (ADC) at 76.9 kSPS, (v) 2-wire serial interface (TWI), (vi) programmable serial interface (USART).

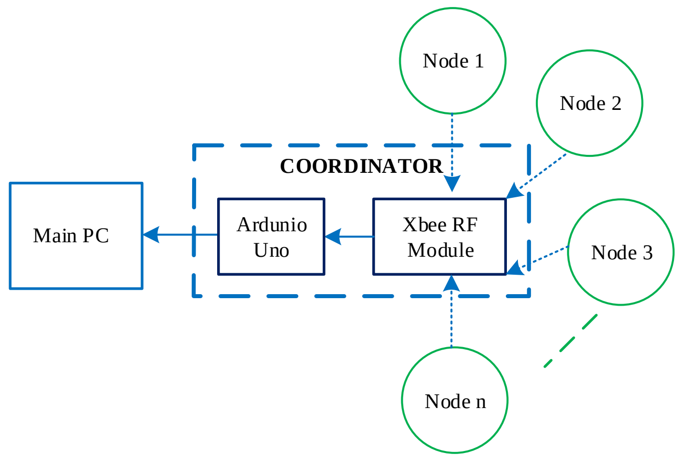

Figure 1: The monitoring system block diagram.

The ADC peripheral is of key importance for this project, as most of the monitored health indicators we will be in the form of analog signals. As a result, these have to be digitized before being transmitted to the coordinator MCU for further processing. With its 10-bit resolution and high sampling rate we are able to acquire data reliably from the monitored athlete. The TWI peripheral is a serial interface compatible with the widely used I2C bus. This bus is used mainly for communication with sensors that have digital output. Finally, the programmable serial interface is used for communication between the nodes and the main server. The RF modules that are used have a serial communication interface. Additionally, the USART interface is used for data transmissions from the coordinator to the main PC. This is implemented by means of a secondary MCU dedicated to the USB to serial conversion.

The sensors that are employed for the purposes of this project are the following: (i) tri-axial accelerometer via the TWI interface, (ii) electrocardiography sensor (ECG) via the ADC, (iii) airflow sensor via the ADC, (iv) temperature sensor via the ADC, (v) galvanic skin response sensor (GSR) via the ADC. Every node of the system has to be able to transmit the acquired health indicating signals back to a main server for processing. As a result, every wireless node is equipped with a radio module which implements the wireless communication with the server. The module we are using is the Xbee PRO S2B [15]. This module implements the ZigBee protocol which is a communication protocol known for its high performance on multi-node sensor networks; it has low power consumption, high range (about 90 meters in indoor or urban areas) and high data throughput (up to 35kbps). The employed Xbee PRO S2B

RESEARCH ARTICLE

module has a serial communication interface which is supported by the USART peripheral on our MCU. The block diagram of a single sensors' node is shown in Figure 2, while Figure 3 presents a photograph of a fully wired functional sensors' node.

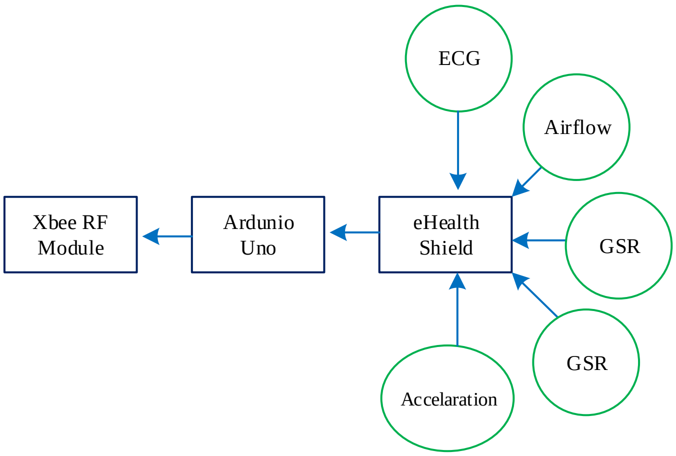

Figure 2: The block diagram of each sensor node.

Figure 3: Image of a fully wired node.

2.2 Software Description

The software developed for this project has been written in C programming language. The Atmel Studio IDE [16] which supports all MCUs produced by Atmel, was used. Although Arduino IDE is the most common IDE used for development with the Arduino hardware, and comes with a variety of peripheral drivers, we decided to use Atmel Studio in order to obtain a more accurate timing control, better code efficiency and easier debugging.

The main routine: The main function given in Table 1 is responsible (i) for setting up the peripherals, as it is explained later, and (ii) for transmitting the collected data to the monitor PC periodically. The data collected from the sensors are being buffered into arrays. When a buffer is full the *buffer_switched* flag is toggled. This event allows the code inside the loop to be executed and as a result, all the data from the full array are transmitted through the UART interface using the *printf()* function. After the transmission is complete, the flag is toggled off.

Table 1. Main function

```
int main(void)
{
        static FILE Filestream =
FDEV_SETUP_STREAM(UART_printf_wrap, NULL, _FDEV_SETUP_WRITE);
        stdout = &Filestream;
        Setup();
        _delay_ms(1000);
        sei();

  while(1)
  {
if (buffer_switched)
        {
                PORTD^=(1<<PIND4);
                uint8_t i;
                for (i=0;i<=(fast_buffer_capacity-1);i++)
                {
        printf("%d,%d,%d\r\n",Acc_X[column_to_send][i],
        Acc_Y[column_to_send][i],Acc_Z[column_to_send][i]);
                }
                buffer_switched=false;
                PORTD^=(1<<PIND4);
        }
  }
}
```

The "setup" functions: The "setup" function given in Table 2 is the first function called in the main routine. This function is used to enable and configure peripherals such as the serial interface, the ADC and the general purpose input-output pins (GPIO). It is also used to configure the accelerometer at the desired mode of operation. Finally, the Timer Interrupt is set up at the desired frequency.

Table 2. Setup function

```
void Setup()
{
        ADC_Setup(Enable,div128,AREF,Disable,Disable);
        Timer_Setup(timer_div64,TimerBaudValue,Enable);
        UART_Setup();
        MMA_Setup();
        GPIO_Setup();
}
```

Most of the functions called inside the setup function are declared and implemented in separate source files given in Table 3 for better code management.

RESEARCH ARTICLE

Table 3. Declaration of the setup functions called in the main setup function

```
void UART_Init(uint8_t TXEnable,uint8_t RXenable,uint16_t Baudreg)
{
	UCSR0C|=(1<<UCSZ00)|(1<<UCSZ01);
	UBRR0|=Baudreg;
	if (TXEnable)
	{
		UCSR0B|=(1<<TXEN0);
	}
	if (RXenable)
	{
		UCSR0B|=(1<<RXEN0);
	}
}
void UART_Setup()
{
	UART_Init(Enable,Disable,8);
}
typedef enum {timer_div1,timer_div2,timer_div8,timer_div64,timer_div256,
timer_div1024} TimerPresc;
void Timer_Setup(TimerPresc presc,uint16_t CompareValue,State IntEnable)
{
	TCCR1B|=(1<<WGM12);
	TCCR1B&=~(0x07);
	TCCR1B|=presc;	OCR1A=CompareValue;
	if(IntEnable)
	{
	TIMSK1|=(1<<OCIE1A);
	}
	else
	{
	TIMSK1&=~(1<<OCIE1A);
	}
}
void MMA_Setup(void)
	{
		bool errorflag=false;
		errorflag|= MMA_TWI_Init(500);
		errorflag|=MMA_Mode(STANDBY);
		errorflag|= MMA_WakeModeCFG(_200H,NORMAL);
		errorflag|= MMA_WriteAddress(MMA8452,XYZ_DATA_CFG,0x02);
		errorflag|= MMA_GMode(_8G);
		errorflag|= MMA_AutoSleepEN(false);
		errorflag|=MMA_Mode(ACTIVE);
		if (errorflag)
		{
			printf("MMA setup Error!!!\r\n");
		}
		else
		{
			printf("MMA setup Complete\r\n");
		}
	}
```

The timer interrupt service routine: The "setup" function given in Table 4 is the first function called in the main routine. This function is used to enable and configure peripherals such as the serial interface, the ADC and the general purpose input-output pins (GPIO). It is also used to configure the accelerometer at the desired mode of operation. Finally, the Timer Interrupt is set up at the desired frequency. The Timer Interrupt Service Routine (ISR) executes the necessary code to get one sample from each one of the available sensors. The Timer is set to trigger at the highest necessary sampling rate. All other sampling rates are derived by appropriately subdividing the highest sampling rate. At an example, there is a 1/10 ratio between the sampling rates of the accelerometer and the sensors for temperature and galvanic skin response (GSR). For

RESEARCH ARTICLE

the higher frequency signals such as airflow and electrocardiography (ECG) we use a 1/2 sampling ratio. The ISR begins reading the six consecutive registers of the accelerometer starting from the register OUT_X_MSB. These registers contain the latest sampled data from the three axes of the sensor. These values are converted to signed integers and then are buffered. The same applies to all analog sensors that are sampled at different sampling frequencies. The data buffers consist of two columns. As soon as a column is full, a flag named *buffer switched*, indicates that this column is ready for transmitting. Additionally, the number of the column which is full is stored in the variable *column to send*. These two variables are used in the main thread which is responsible for transmitting them to the monitor PC. While the first column's data are being transmitted the other column becomes the active buffer and all new data are being stored there. This buffer switching procedure given in Table 5 continues endlessly in order to achieve continuous sampling at high sampling rate, and better control over the traffic congestion.

Table 4. Interrupt service routine

```
ISR(TIMER1_COMPA_vect)
{
        PORTD^=(1<<PIND2);
        slow_ref_ticks++;
        medium_ref_ticks++;
        uint16_t temp;
        uint8_t data[6];
        MMA_ReadFromAddress(MMA8452,OUT_X_MSB,data,5);
        temp=(data[0]<<8)+data[1];
        Acc_X[column][fast_index]=(int16_t)temp;
        temp=(data[2]<<8)+data[3];
        Acc_Y[column][fast_index]=(int16_t)temp;
        temp=(data[4]<<8)+data[5];
        Acc_Z[column][fast_index]=(int16_t)temp;
        fast_index++;
        if(medium_ref_ticks>=medium_sampling_ratio)
        {
                PORTD^=(1<<PIND3);
                ecg_buff[column][medium_index]=Read_ADC(ecg);
                airflow_buff[column][medium_index]=Read_ADC(airflow);
                medium_ref_ticks=0;
                medium_index++;
                PORTD^=(1<<PIND3);
        }
        if(slow_ref_ticks>=Slow_sampling_ratio)
        {
                PORTD^=(1<<PIND4);
                gsr_buff[column][slow_index]=Read_ADC(gsr);
                temp_buff[column][slow_index]=Read_ADC(temperature);
                slow_ref_ticks=0;
                slow_index++;
                PORTD^=(1<<PIND4);
        }
```

Table 5. Buffer management routine

```
if (fast_index>=fast_buffer_capacity)
{
        fast_index=0;
}
if(medium_index>=medium_buffer_capacity)
{
        medium_index=0;
}
if(slow_index>=slow_buffer_capacity)
{
        slow_index=0;
        column_to_send=column;
        column^=0x01;
        buffer_switched=true;
}
```

3. EVALUATION STUDY

An evaluation study was realized to prove the usability of the proposed system and some test data was as shown in Figures 4 and 5. All data have been logged with Realterm [17], serial communication terminal software, and then imported and processed using the MATLAB numerical computing environment [18]. An ECG graph taken for about fifteen seconds is depicted in Figure 4. It is easy to approximate the heart pulse rate by post processing the ECG data. In the specific example of Figure 4 it was estimated that heart pulse rate was approximately 84 pulses per minute.

International Journal of Computer Networks and Applications (IJCNA)
Volume 2, Issue 6, November – December (2015)

RESEARCH ARTICLE

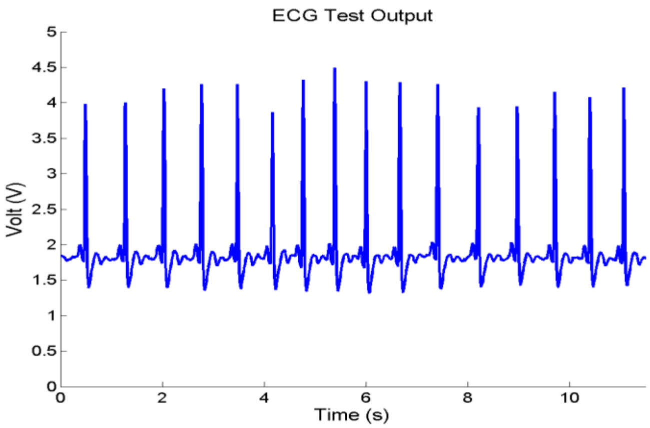

Figure 4: ECG test output data.

Test data acquired from the accelerometer are shown in Figure 5. Note that the accelerometer was configured so that the integrated high pass filter was to be disabled. The subject's motion that was recorded corresponds to 180 degrees turn around the Y axis, a free fall (at about 7sec) and finally an impact on a table.

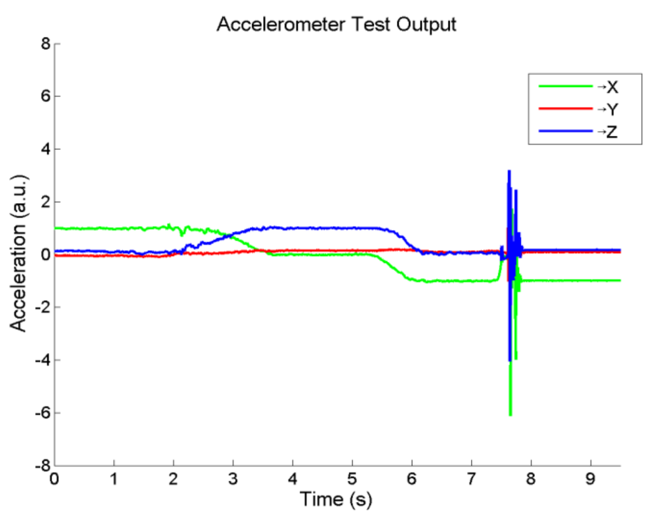

Figure 5: Accelerometer test output data.

As it is shown in this evaluation study, the proposed low-cost system can provide data related to specific health conditions. Hence it can be a valuable tool for healthcare providers and the families of the elderly and disabled. However, considering the size related constraints which can limit its practical and comfortable use, some major improvements are needed to make the system more compact and easy-to-care. One of the best approaches for this aim is the use of Micro-Electromechanical Systems (MEMS) technology which combines microcomputers/controllers with a set of tiny mechanical devices such as sensors, valves, and actuators in semiconductor chips [19, 20]. In this direction, a useful complementary technique for MEMS-based wireless healthcare applications especially for smart home integrated applications is the above integrated circuit which enables the placement of a Radio Frequency (RF)-MEMS device directly on top of the integrated circuit by using a thick copper technology [21-23].

4. CONCLUSION

It is obvious that efficient care can enable the elderly and disabled to enjoy the comfort of living at home with full confidence. To achieve this, a combination of different technological solutions can be employed and personalized care and immediate help can be provided to the elderly and disabled. In this study, a wireless sensor network-based health monitoring system for the elderly and disabled has been investigated and a performance evaluation study to show the use of the proposed system has been carried out. In this paper, the proposed system's both hardware and software components have been presented. If connected to the Internet through a gateway, the proposed low-cost system enables health care providers to remotely monitor their patients' main health parameters. It is also a valuable tool for the relatives of the elderly and disabled. The future work of this study consists of an evaluation study in a distributed scenario involving a group of elderly people.

ACKNOWLEDGMENT

This work was supported by Research Fund of the Trakya University. Project Number: 2015/109.

REFERENCES

[1] Istepanian, R. S. H., Jovanov, E. and Zhang, Y. T. (2004). "Introduction to the Special Section on M-Health: Beyond Seamless Mobility and Global Wireless Health-Care Connectivity," IEEE Transactions on Information Technology in Biomedicine. 8(4), 405-414.

[2] Das, R., Tuna, G., (2015), "Machine-to-Machine Communications for Smart Homes", International Journal of Computer Networks and Applications (IJCNA), EverScience Publications, 2 (4), 196-202.

[3] Tuna, A., Das, R., and Tuna, G. "Integrated Smart Home Services and Smart Wearable Technology for the Disabled and Elderly," (2015). In Proceedings of 4th International Conference on Data Management Technologies and Applications, ISBN 978-989-758-103-8, pp. 173-177, 2015. DOI: 10.5220/0005552001730177

[4] Moutacalli, M. T., Marmen, V., Bouzouane, A. and Bouchard, B. (2013). "Activity Pattern Mining using Temporal Relationships in a Smart Home", IEEE Symposium on Computational Intelligence in Healthcare and e-health (CICARE), 83-87.

[5] Jovanov, E., Lords, A., Raskovic, D., Cox, P., Adhami, R. and Andrasik, F., (2003). "Stress monitoring using a distributed wireless intelligent sensor system", IEEE Engineering in Medicine and Biology Magazine, 22(3), 49-55.

[6] Son, Y. –S., Jo, J., Park, J. –H. and Pulkkinen, T. (2013). "Diabetic Patient Care using Home User Activity Recognition", ICTC 2013, 191-196.

[7] Alemdar, H. and Ersoy, C., (2010). "Wireless sensor networks for healthcare: A survey", Computer Networks, 54(10), 2688-2710.

RESEARCH ARTICLE

[8] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. and Cayirci, E., (2002). "Wireless sensor networks: a survey", Computer Networks, 38, 393-422.

[9] Buratti, C. and Verdone, R., (2009). "Performance analysis of IEEE 802.15.4 non beacon-enabled mode", IEEE Transactions on Vehicular Technology, 58(7), 3480-3493.

[10] Zhu, C., Zheng, C. and Han, G., (2012). "A survey on coverage and connectivity issues in wireless sensor networks", Journal of Network and Computer Applications, 35, 619-632.

[11] Francesco, M. D., Anastasi, G., Conti, M., Das, S. K. and Neri, V., (2011). "Reliability and energy-efficiency in IEEE 802.15.4/ZigBee sensor networks: an adaptive and cross-layer approach", IEEE Journal on Selected Areas in Communications, 29(8), 1508-1524.

[12] *Internet:* Arduino 2015, what is Arduino, http://www.arduino.cc/ [Accessed 15 March 15]

[13] *Internet:* Cooking Hacks 2015 e-Health Sensor Platform V2.0 for Arduino and Raspberry Pi [Biometric / Medical Applications], http://www.cooking-hacks.com/documentation/tutorials/ehealth-biometric-sensor-platform-arduino-raspberry-pi-medical [Accessed 15 March 15]

[14] Buyukpatpat, H., Das, R., (2015). "Akıllı Ev Sistemleri", Fırat Üniversitesi, Teknoloji Fak., Yazilim Muhendisliği, Bitirme Projesi.

[15] *Internet:* Digi 2015 XBee / XBee-PRO ZB (S2B) Modules, http://www.digi.com/support/productdetail?pid=4549&type=documentation [Accessed 15 March 15]

[16] *Internet:* Atmel 2015 Atmel Studio 6, http://www.atmel.com/microsite/atmel_studio6/ [Accessed 15 March 15]

[17] *Internet:* Realterm 2015 Realterm: Serial Terminal, http://realterm.sourceforge.net/ [Accessed 15 March 15]

[18] *Internet:* MathWorks 2015 MATLAB, http://www.mathworks.com/products/matlab/ [Accessed 15 March 15]

[19] Madou, M. J., (2002), "Fundamentals of Microfabrication: The Science of Miniaturization", (Second Edition), Bosa Roca: CRC Press.

[20] Madou, M. J., (2011). "From MEMS to Bio-MEMS and Bio-NEMS: Manufacturing Techniques and Applications (Fundamentals of Microfabrication and Nanotechnology)", Bosa Roca: CRC Press.

[21] Nguyen, C. T.-C., (2006) MEMS Technologies and Devices for Single-Chip RF Front-Ends. Journal of Microelectronics and Electronic Packaging: October 2006, Vol. 3, No. 4, pp. 160-168. DOI: 10.4071/1551-4897-3.4.160.

[22] Nabki, F., Dusatko, T. A. and El-Gamal, M. N., (2008). "Frequency tunable silicon carbide resonators for MEMS above IC", IEEE Custom Integrated Circuits Conference, 185-188.

[23] Enz,C. C. and Kaiser,A.(Eds). (2013). "MEMS-based Circuits and Systems for Wireless Communication", Springer, New York.

Authors

Gurkan Tuna is an Assoc. Professor at the Department of Computer Programming of Trakya University, Turkey (2006….). He has authored several papers in international conference proceedings and refereed journals, and has been actively serving as a reviewer for international journals and conferences. His current research interests include smart grid, ad hoc and sensor networks, and robotic sensor networks.

Resul Das received his BS and MSc. in Computer Science from Firat University in 1999, 2002 respectively. He received PhD degree from Electrical and Electronics Engineering Department in same university in 2008. He is an Assoc. Professor at the Department of Software Engineering of Firat University, Turkey. He has authored several paper in international conference proceedings and refereed journals, and has been actively serving as a reviewer for international journals and conferences. His current research interests include Knowledge Discovery, Web Mining, Complex Networks, Computer Networks, Information and Network Security.

Ayse Tuna is a lecturer at School of Foreign Languages, Trakya University, Turkey since 2005. She has authored papers in international conference proceedings, and has been actively serving as a reviewer for international conferences and journals. Her research interests are support services for the elderly and disabled, preservation of cultural heritage, and data management methodologies.