INFRARED MONITORING SYSTEM USING MIWI TECHNOLOGY

Gheorghe TURCAN, Nicolai DOBROSTOMAT, Marius NEAG, Marina TOPA
Technical University of Cluj-Napoca, Cluj-Napoca, Faculty of Electronics, Telecommunications and IT
Str. Baritiu nr. 26, Cluj-Napoca, Tel. +40 264 401 224 Fax. +40 264 591 689, Email: Marina.Topa@bel.utcluj.ro

Abstract: This paper presents a wireless data acquisition system for monitoring thermal activity using single- or multiple-point infrared sensors. The wireless capability of the system is provided by Microchip's MIWI Point to Point protocol implemented by a PIC16LF1937 microcontroller which comprises the MRF24J40MA radio transceiver module from the same company. The received data are sent through the USB port to a PC, where they are processed and plotted by an application developed for this purpose. The application also allows the user to control all components of the systems through a simple GUI.

Keywords: *IR Sensors, Thermal monitors, Wireless, MIWI, Data Acquisition, Microcontrollers*

I. INTRODUCTION

Wireless data transfer is widely used in monitoring systems in order to reduce installation costs and inconveniences associated with wired communications. Monitoring systems have been used for decades for collecting data required by various applications such as security/surveillance and control of physical parameters (temperature, pressure, flow, etc.). However, having wireless communication capabilities is a relatively new feature, which makes them more field-usable and easier to install. Despite this, several different approaches to adding wireless to sensors have been reported in the literature, [1], [2].

This paper presents a data acquisition system for monitoring thermal activity using single- or multiple-point infrared sensors; its main components are shown in Fig.1. The infrared sensor (IR) on the left-hand side of Fig.1 provides thermal data in electronic format to a low-power transceiver, which transmits them wirelessly to a base station, represented here by a radio receiver and a PC. The base station can also transmit data to the IR sensor, in order to control its operation.

Numerous ready-to-use sensors are available today but most of them are not suited for the application envisaged here on two counts: cost and size. For example, Fluke provides a wide range of IR cameras, but their price start at 1.300$, with the ones having wireless data transfer capabilities going up from 4.000$ to 25.000$; also, Fluke TiX660 IR camera is rather big (210mm X 125mm X 155mm) and weight approximately 2Kg [3].The FLIR ONE IR cameras are far smaller (72 mm X 26mm X 18mm) but they need to be connected physically to a mobile device so the dimensions and cost are higher [4]. An interesting wireless thermal application is presented here [14] but the scale is much lower and the temperature sensors are like the ones on chip that do not provide visual information but only ambient temperature.

The system presented here employs a different approach in order to reduce both cost and size, while improving on the communication and ease-of-installation features of the products currently on the market.

Its key elements are:

- Microchip's MIWI Point to Point (P2P) protocol ported to PIC16LF1937 microcontroller that uses a Microchip MRF24J40MA radio transceiver module [5].
- the low-cost, small-size IR sensor MLX90620 [6]

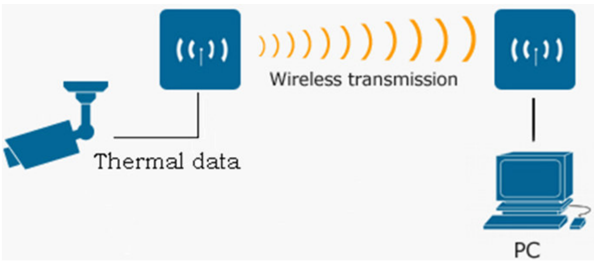

Figure 1: Block diagram of the system for monitoring thermal activity presented in this paper

MIWI and MIWI Point-2P are proprietary wireless Point -to-Point (P2P) protocols developed especially for wireless personal area networks (WPANs) built with low-power transceivers based on the IEEE 802.15.4 standard. They are designed for cost-constrained networks which require low data transmission rates over short distances, such as industrial monitoring and control, home and building automation, low-power wireless sensors and automated meter reading [5].

The proposed system employs the hand shake function of the MIWI P2P protocol to establish a connection between the Full Function Device (FFD) which acts as the master and several Reduced Function Device (RFDs) working as slaves. This connection enables the master to control the slaves and gather data which is then conveyed to a PC via its USB port. The PC runs a dedicated application that processes the thermal data received from the IR sensor in order to present it in an understandable form to the user. Also, the application provides information about the slaves - such as number of slaves connected within the network and their status – and allows the user to control them. The wireless capability and low power consumption of the slave offers great mobility and the possibility to eliminate the grid

Manuscript received November 1, 2016; revised December 9, 2016

restrictions with the use of batteries.

This paper is organized in four Sections: the Introduction is followed by a detailed description of the proposed data acquisition system for monitoring thermal activity, including the main requirements and implementation details of its software and hardware elements. Experimental results are presented in Section III, followed by a brief comparison between the system proposed here and two similar systems currently on the market. The main results are summarized and the conclusions are drawn in the last Section.

II. DESCRIPTION OF THE PROPOSED SYSTEM

A. Main requirements of the envisaged application

The system presented here is tailored for low-cost IR sensors with small-to-medium resolution: 64 to 768 pixels. It ensures efficient monitoring of thermal activity based on fast data transfer via wireless communications. The system is battery-powered so low-power consumption is essential. The following resources were employed for implementing it:

1) Microchip PIC16LF1937: Microcontroller
2) Microchip MPLAB® X IDE: development environment
3) MRF24J40MA 2.4 GHz: radio transceiver module
4) 1 pixel sensor (MLX90615)
5) 16x4 pixel sensor (MLX90620).
6) Dip Trace Layout software

The PIC16LF1937 is used for data acquisition and transmission; the data can be stored on an 8192 words Flash memory and transmitted thanks to the 512 bytes of SRAM at disposal. This device is optimized for low-voltage and low-power operation: Vsupply between 1.8V to 3.6V and typical current consumption of 75 µA/MHz at1.8V.

B. Software Implementation

The MiWi protocol was chosen as it consumes less power than the WiFi protocol and is better suited for transferring large-size data than the Bluetooth Protocol.

MIWI P2P protocol on PIC16LF1937:

The MIWI Protocol takes more space than is usually available on an MCU but, different to other protocols, allows the user to disable unused functions. This way one can select only the most important features and run the resulting bare-bones protocol in a little more than 4K of flash memory.

The PIC16LF1937 is well suited for this approach: it is compliant with the minimum requirements of the MIWI P2P protocol, offers key peripheral features and the Nano Watt XLP Technology allows the current consumption of the board to drop as low as 31uAh during sleep mode.

Sensor Connection:

The selected sensors communicate with and provide data to the master through the SMBUS protocol. The version of the SMBUS protocol selected allows for a speed of 100 kHz. Data acquisition is set to be realized through pins C6 and C7 of the MCU.

The basic instructions are shown in Fig. 2; they are explained later in the text and detailed in Fig. 6.

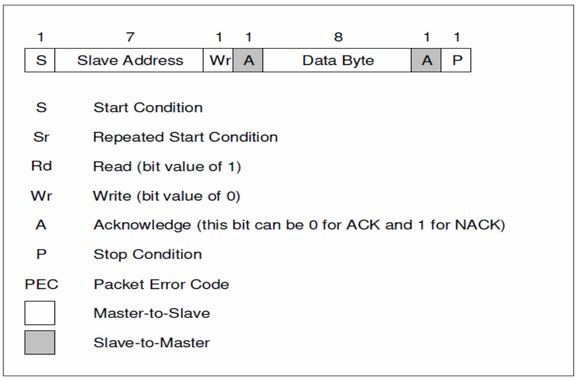

Figure 2: SMBUS packet element key

A bit-banging software function was created to implement the data acquisition and several special functions were created for handling transmission queues.

The basic functions were implemented in a way that allows the bit-banging protocol to be used universally in any 16bit PIC MCU without modifications, except for setting the acquisition pins and the core frequency. For this, the functions are implemented through voltage level manipulation on the SDA and SCL connections according to the protocol specifications.

- the “START” function is implemented in the following flow of manipulation: SDA/SCL – High; Delay; SDA – Low, SCL – High.
- the “STOP” function is implemented in the following flow of manipulation: SDA/SCL – Low; SDA – Low, SCL – High; Delay; SDA/SCL – High
- the “ACK” function is implemented in the following flow of manipulation: SDA – Low after 8bit data reception.
- more complex functions such as “DATA BYTE” are a combination mask of basic functions: 8 X “DATA BIT”.
- a simple read loop is used to receive data, which are memorized and transmitted to the interpretation application.

All communication functions are essentially combinations of same basic functions; thus, we can read words of 64 bytes by simply repeating the basic functions the required number of times.

The communication protocol specifies the way an error has to be handled; this was also implemented with basic functions. For example, receiving a negative acknowledgement “NACK” implies the need of a “STOP” function to be able to restart the request.

Typical read and write scenarios are described in Figs. 3 and 4. An important feature is that the number and/or the format of bytes the system reads or writes can be easily adapted to the sensor requirements. Fig. 5 depicts a 64 pixel sensor block diagram while Fig. 6 illustrates a 128 bytes (64 words) read function. This demonstrates that the complexity of the function is not important as long as the basic functions are correctly implemented.

Data Transfer:

The data acquired from the MLX90620 sensor has a dimension of 256 bits but maximum packet size the MiWi P2P connection can transfer is 96 bites. This problem was solved by transferring data in 4 packets of 64 bits plus the enumerator. This approach is useful in general for transferring large data packets, which cannot be handled directly by the MCU.

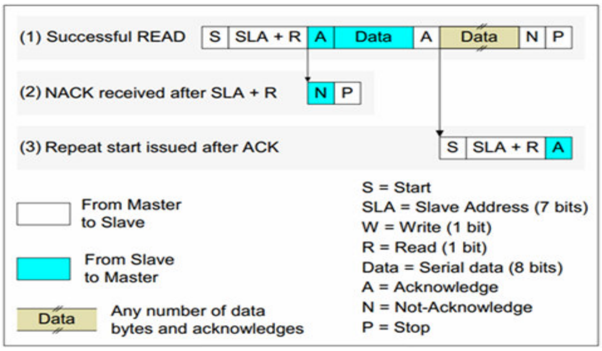

Figure 3: Typical READ Scenarios

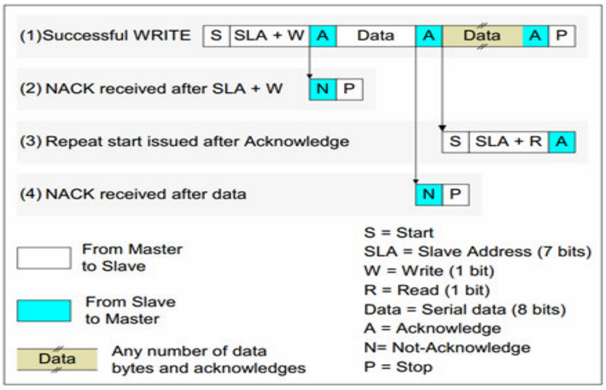

Figure 4: Typical WRITE Transfer Scenarios

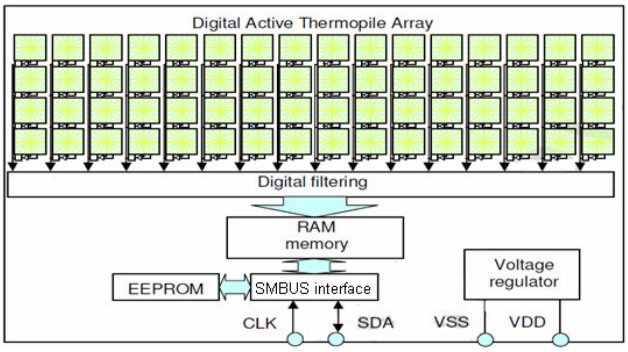

Figure 5: 64 Pixel IR sensor Block Diagram

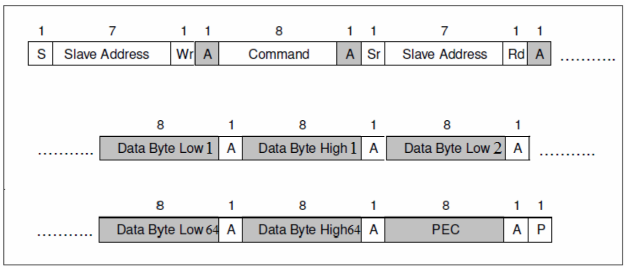

Figure 6: A "64 Words READ" function with SMBus.

A command byte definition was implemented to allow control over the slave instruction directly from the PC application. This byte contains 8 instruction bits arranged as (b0 b1 b2 b3 b4 b5 b6 b7), each controlling an instruction for the slave, as follows:

-0x00: if the slave received all the configuration bits set to low, all the data transmission is stopped and the slave enters idle mode. Receiving 2 times in a row all the configuration bits set to low will start the sleep mode of the slave.

-b6: Transmission Mode = set to low will send the current measured data, set to high will send data continuously until stopped.

-b5: Temperature Type = if received low will read the ambient temperature (the temperature of the case), if set high will read the object temperature.

-b4: Sensor Select = 0 -> 1 pixel sensor; 1 -> 64 pixel sensor.

-b3: EEPROM Read = 0 -> ignore EEPROM; 1 -> read EEPROM.

The first 3 bits are for the master to address a selected slave, which allows for 8 slaves.

The sleep and wake functions of the slave are implemented to save battery power: after receiving this command the LEDs are turned off, the sensors are configured to enter their sleep mode, the MRF24J40MA enters the power state "SLEEP". Finally, the MCU itself goes into sleep mode. Once the slave begins the sleep sequence it will wake up every 4 seconds and will ask the master whether there are data to be received. If the answer is positive, the slave will proceed to implement the received command. Note that once the wake-up procedure begins, the sensors are reconfigured and they need 80ms delay to transmit valid data. If no data are available the slave will return to sleep for another 4 seconds, dropping its current consumption to 31uA This way a normal-size battery can supply a slave in sleep mode for up to 2 years. For this application the batteries selected are AA Duracell Alkaline MN1500 [7].

Software Implementation:

Fig. 7. presents the flowchart of the software the "SLAVE" MCU runs in order to implement the data acquisition, data storage, data transmission and the other behavior instructions. The flowchart showed in Fig. 8 details the "MASTER" flow of instructions that makes the bridge communication of the PC Application to the "SLAVE" sensor board.

C. Physical Implementation

A prototype of the proposed system was implemented on a 3cm x 4cm board than fits in a 3 cm tall plastic box. The board was implemented on a double layer FR4 board, minimum pin dimensions 0.3mm, minimum via dimension 0.6mm; the PCB layout is shown in Fig. 9 presents while a photo of the entire prototype is given in Fig. 10.

The prototype board includes measurement points and test areas which can be removed to reduce area. The board size can be further reduced by employing the QFN version of the MCU and the SMD version of the sensor and by using the internal timer instead of external crystal; also, one can eliminate the LED and Push Button. The overall dimensions can be brought down to be smaller than 20mm x 20mm.

The box shown in Fig. 10 accommodates the board and two AA batteries; its height can be reduced by using coin-size batteries.

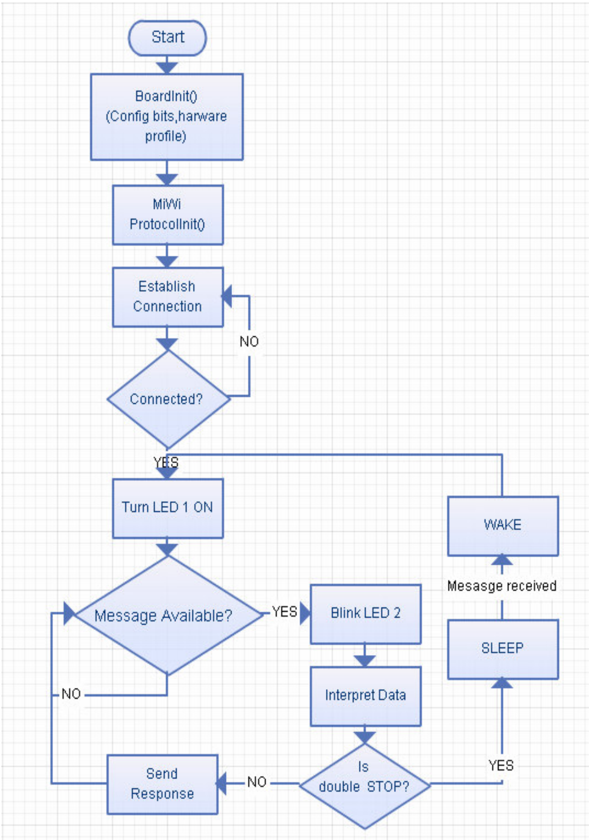

Figure 7: The flowchart of the "Slave MCU"

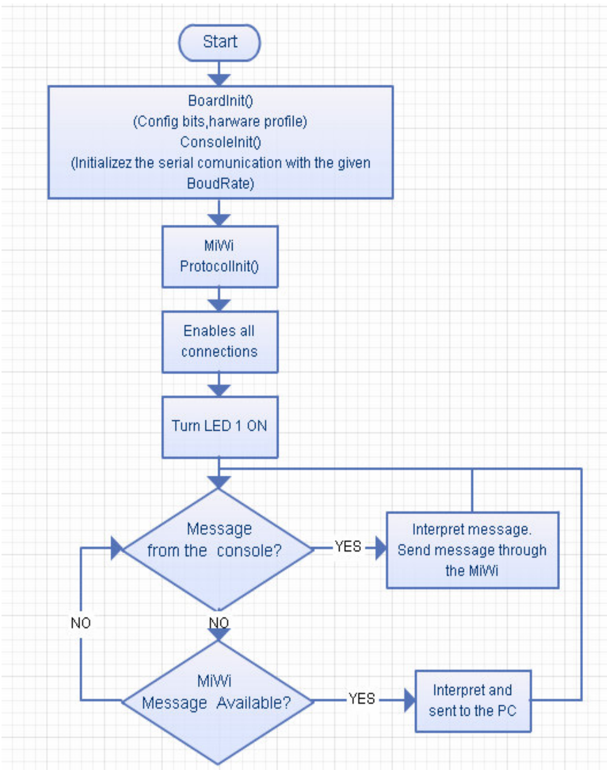

Figure 8: The flowchart of the bridge communication of the PC Application to the "SLAVE" sensor

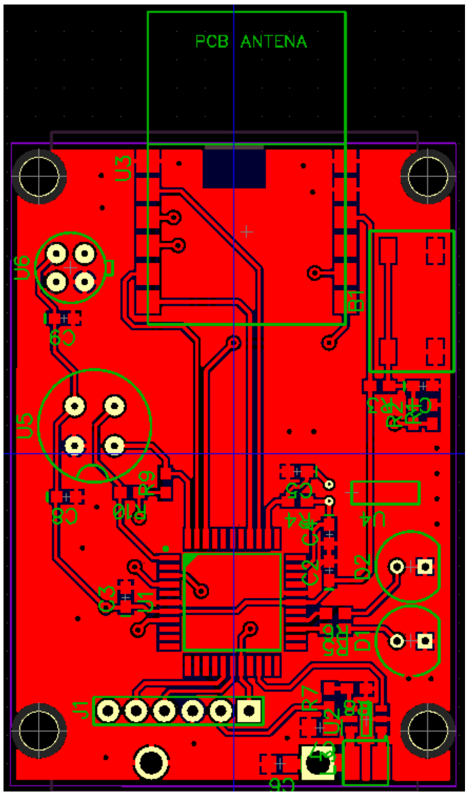

Figure 9: PCB layout design done in DipTrace

Figure 10: Physical implementation of the proposed system

The maximum power consumption achieved during both sensor communication and data transmission was just under 40mA peak. But this data burst takes only a few milliseconds/second, so the total power consumption of the board is under 10mAh when measuring continuously. The power consumption can be lowered even more by implementing the suggestions for improving the board size presented in the previous paragraph.

III. EXPERIMENTAL RESULTS

A. Brief description of the GUI

Microsoft Visual Studio 2010 was used to implement the Graphic User Interface application, which is the bridge that connects the user with the hardware.

The GUI realizes a connection with the hardware through the USB to serial communicator. The application implemented allows the user to choose the connected port and the Baud Rate of the communication as depicted in Figure 11; it displays the connection status and the available modules. The user is able to select which sensor data to acquire, and either to get one capture or continuous data flow as depicted in Figure 12.

The temperature is received as a numerical value but a color representation is used for graphical display. Fig. 13 details the mapping of temperature ranges to colors.

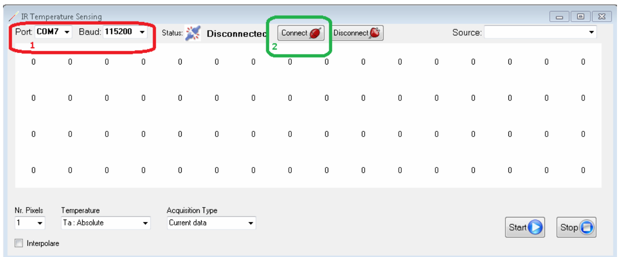

Figure 11: Port and Baud selection.

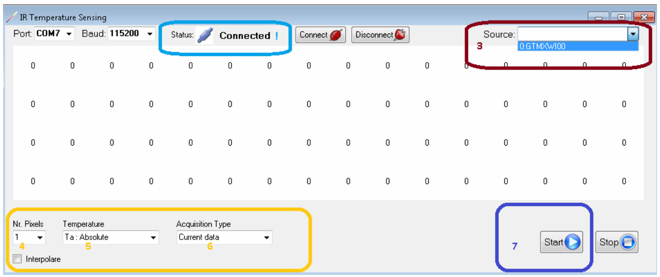

Figure 12: Module and Sensor selection.

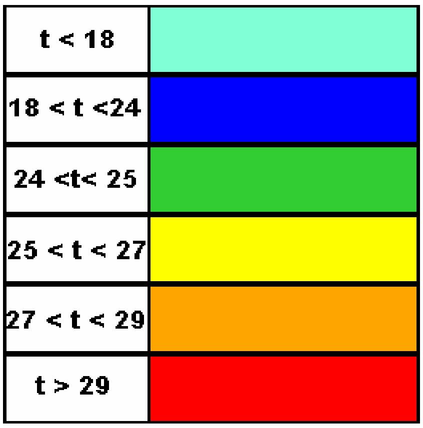

Figure 13: Mapping of temperature ranges to colors.

B. Three real-life tests

Test 1: In Figure 14 the application was used to read the

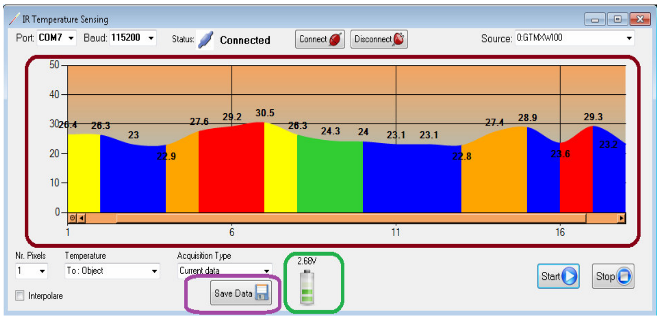

Figure 14: The data read from the 1 pixel MLX90615 sensor is represented into a chart.

1px sensor, both Ta and To: the application displays the received data using the color code described in Fig. 13 but the numerical values in degrees Celsius are also given. The battery status is also monitored as is showed in the green rectangle. A save data function is also available – marked by the purple rectangle. When activated it generates a text file that comprises the complete log of temperatures received since the beginning of the data acquisition session.

Test 2: Figure 15 presents the image of a cup filled with hot water (left hand side of the picture) sitting opposite a cup filled with cold water, as seen by the MLX90620 sensor. The 64 pixels were arranged in a 16x4 image. Note that the temperature of each of the pixel is displayed both numerically and in color-code. The minimum and maximum temperature values are given at the bottom of the screen – see the area marked by the grey rectangle. The very same data are presented in Figure 16, but this time the interpolation function was activated.

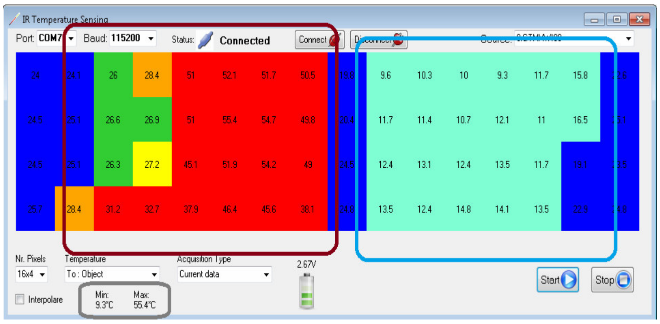

Figure 15: The data read from the 64 pixel MLX90620 sensor presented as a 16x4 pixel image.

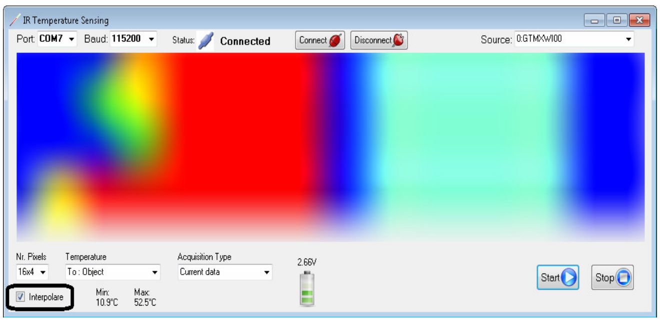

Figure 16: Same data as those shown in Fig.5 by presented here after bi-cubic interpolation

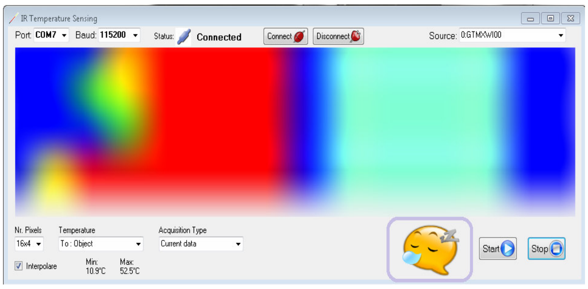

Figure 17: The application notifies the user that the board is in "SLEEP MODE"

Several versions of three main types of interpolation were implemented and tested: Nearest-neighbor, Bilinear and Bi-cubic [8]. The "**High bi-cubic interpolation**" [9] yielded the best results for several real-life applications we considered. The numerical values are not displayed when the *"Interpolation"* function is activated.

The user can put the system in sleep mode by pressing the Stop button two times in quick sequence; the GUI notifies the user that the selected source has indeed gone in sleep mode as shown in Figure 17.

Test 3: Figure 18 presents the image obtained by placing a human hand in front of the camera. The *"Interpolation"* function was activated in order to help increase the definition of the four fingers in this image. This experiment suggests that the proposed system, although relatively simple, can be used in object recognition applications.

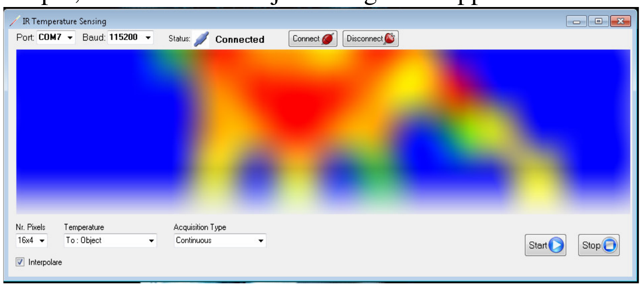

Figure 18: Finger distinction realized with interpolation

C. Comparison with similar systems

To allow for a direct comparison the main features of the data acquisition system for monitoring thermal activity presented in this paper are listed in Table 1, alongside the performance of two commercially-available systems.

Table 1 Comparison of three thermal acquisition systems

Feature	FLIR E40	FLIR E4	This work
Resolution	19,200 pix	4,800 pix	up to 768 pix
Thermal Image Format	JPEG	JPEG	JPEG/BMP/ Excel data
Sensitivity	<0.07 °C	<0.15 °C	<0.2 °C
Accuracy	2°C	2°C	2°C
Wireless Connectivity	Wi-Fi	No	Mi-Wi
Software Application	Yes	No	Yes
Battery Life	4h	4h	32h
Price	**4000$**	**1100$**	**60$ - 90$**

While the maximum sensor resolution supported by the proposed system is up to 25 times lower than the competitors considered in Table 1 one notices its longer battery life and extremely low price.

IV. CONCLUSIONS

The design and implementation of a low-cost wireless data acquisition system for monitoring thermal activity using single- or multiple-point infrared sensors was demonstrated. It is based on Microchip's PIC16LF1937 microcontroller which comprises the MRF24J40MA radio transceiver module from the same company. The wireless capability relies on Microchip's MIWI Point to Point protocol, which allows data transfer over distances of up to 100m.

The selected MCU could not run the entire MIWI protocol so a minimum-feature version was employed. This way, 4K of flash memory was enough to run the bare-bones protocol that ensured the required functionality.

This connection enables the master to control the slaves and gather data which is then conveyed to a PC via its USB port. The PC runs a data-processing application developed for this purpose, which includes an easy-to-use GUI. The application provides mapping of numerical data to color and several types of data interpolation. It also enables the user to control the slaves and provides various information such as number of slaves connected and their current status.

The proposed system was optimized for low-voltage and low-power operation: Vsupply between 1.8V to 3.6V and typical current consumption of 75 µA/MHz at 1.8V; in sleep mode the slave consumes less than 31uA, enabling it to operate for up to two years without changing its batteries.

A direct comparison with two commercially-available systems showed that our solution implementation has important advantages which, for a host of applications, more than compensate for its poorer resolution: eight times longer battery life and six to twenty-five times lower cost.

Future developments of this system include:

- an application can be implemented for the gateway so that the acquired data can be accessed from the internet.
- the addition of more functions of the stack, to increase, for example, the number of possible connections and for supporting higher-performance sensors, or even webcams.

REFERENCES

[1] A. Topa, A. Contiu, V. Dădârlat, "A Mule Based Strategy for Wireless Sensor Motes to Reduce Power Consumption", Acta Technica Napocensis – Electronics and Communications, vol 49, Number 1/2008, pp.26-29.

[2] A. Contiu, A. Topa, V. Dădârlat, "A Virtuallab Approach for Wireless Sensor Motes and Wireless Sensor Networks", Acta Technica Napocensis, vol 49, no 1/2008, pp. 20-25

[3] http://en-us.fluke.com/products/infrared-cameras, 05/2016

[4] http://www.flir.eu/flirone/ios-android, 05/2016

[5] AN1204 ''Microchip MiWi™ P2P'', Application Note

[6] Datasheet:"MLX90615ssg-daa", Device information

[7] http://www.farnell.com/datasheets/578005.pdf, 09/2016

[8] http://www.cambridgeincolour.com/tutorials/image-interpolation.htm, 09/2014

[9] R. Keys, "Cubic convolution interpolation for digital image processing". IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. **29** (6), pp. 1153–1160

[10] Datasheet:" SMBUS Bit Banger",

[11] Datasheet:"PIC16LF1937 Data Sheet",

[12] Application Note AN1283 "MiMAC'' (DS01283)

[13] AN1284 ''Microchip MiApp'' (DS01284)

[14] M.Kohvakka, M.Hannikainen, T.D.Hamalainen, "Ultra low energy wireless temperature sensor network implementation", Proc. of PIMRC 2005 Int. Symposium, available on IEEE Xplore