Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 134 (2018) 99–106

Procedia Computer Science

www.elsevier.com/locate/procedia

The 15th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2018)

A proposal of low-cost and low-power embedded wireless image sensor node for IoT applications

M. Tresanchez*, A. Pujol, T. Pallejà, D. Martínez, E. Clotet, J. Palacín

Departament of Computer Science and Industrial Engineering, University of Lleida, Jaume II, 69, Lleida, 25001, Spain

Abstract

This paper proposes the design of a low-cost, low-power and small size image acquisition device to be used in Internet of Things (IoT) applications which collects, processes and sends image data through a 2.4GHz Wi-Fi connection. The proposed processing unit is a low-cost high-performance STM32F7 microcontroller which is able to acquire and transfer images at full rate keeping the core idle to do image processing. The device is supported by a 1.3 Megapixel CMOS camera sensor, and a low consumption Wi-Fi module. This setup allows to create a very compact image sensor device with a built-in processing algorithm for a specific functionality in IoT. In this work, the device hardware architecture is described and discussed, and a comparison with alternative hardware solutions are presented.

Keywords: IoT sensor node; image sensor; video streaming; ARM; Cortex-M7;

1. Introduction

Image sensors are an important type of sensor in several emerging Internet of Things (IoT) applications. They provide detailed environment information by a large array of photodiodes and have a highly competitive market price due to their standardization. Their basic application as camera system for monitoring or supervising a location by video streaming can be extended when they are combined with image processing techniques [1-4]. In IoT, image sensors may be applied in many fields, some recent contributions use image sensors for surveillance video systems in smart

* Corresponding author. Tel.: +34-973-702-715; fax: +34-973-702-700.
E-mail address: mtresanchez@diei.udl.cat

1877-0509 © 2018 The Authors. Published by Elsevier Ltd.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/)
Peer-review under responsibility of the scientific committee of the 13th International Conference on Future Networks and Communications, FNC-2018 and the 15th International Conference on Mobile Systems and Pervasive Computing, MobiSPC 2018.
10.1016/j.procs.2018.07.149

cities [5], for personalized healthcare systems, for recognizing and understanding human activities [6, 7] or for video streaming systems in vehicular IoT [8].

Image acquisition systems require a high performance computation in order to handle the large amount of data. Furthermore, if they have to process and transmit image information through a wide network, more powerful and expensive Digital Signal Processors (DSP) are needed and, consequently, the current consumption increases. This is an important drawback in IoT sensors networking where it is crucial to have low-cost and low-power nodes because usually they are many and powered on all time. Additionally, another relevant factor to consider is the node size, weight and the hardware complexity for final product manufacturing. In [9] and [10] a more complex hardware is used based on System-On-Chip (SoC) which needs and operating system, makeing it less robust and requires a delicate assembly process that hinders its integration in a final IoT sensor product.

In this work, an embedded image acquisition system based on the new high-performance Cortex-M7 microcontroller and a compact CMOS camera module is proposed as a smart wireless IoT image sensor node. This proposed device has a very low power consumption, ideal in IoT environments, and requires very few electronic components which helps to achieve a small sized product at a competitive production cost. The hardware selected takes advantage of the internal hardware peripherals of the microcontroller to acquire and transfer images at the maximum wireless adapter throughput with a minimum computational load. This means that no software or operating system is required for the video streaming process which leaves the CPU free to execute digital image processing for specific applications and thus, having an embedded smart device.

This paper is structured as follows. Section 2 introduces the main electronic components of the proposed device. Section 3 defines the architecture configuration of the microcontroller for continuous image acquisition and transmission. Section 4 shows the performance results and discussion about the device's capabilities. Finally, Section 5 presents the conclusions of this work.

2. Materials

2.1. Digital processor unit

The main processor unit used in this work is the STM32F746NGH6 [11] Micro Controller Unit (MCU) from STMicroelectronics which is based on the high-performance ARM Cortex-M7 32-bit RISC (Reduced Instruction Set Computing) core operating at up to 216 MHz. This MCU incorporates high-speed embedded memories: 1 Mbyte of Flash memory, 320 Kbytes of SRAM, 16 Kbytes of instruction TCM RAM (for critical real-time routines), 4 Kbytes of backup SRAM available in the lowest power modes, and an extensive range of enhanced I/Os and peripherals connected to an Advanced High-performance Bus (AHB) matrix (multi-AHB). It also has two general-purpose 8-stream DMA controllers with FIFOs and burst support, a configurable 8 to 14-bit parallel camera interface that can operate at up to 54 Mbps, and a flexible memory controller to access external memories. These features are essential to make this approach possible.

According to the manufacturer's datasheet, the MCU has a maximum current consumption of 124 mA running at 210MHz with all peripherals enabled whereas in the lowest power mode (standby mode), without losing the program execution, is less than 3.6µA.

This work is carried out using the STM32F746G-DISCO advanced development board (Fig. 1) based on this MCU. In addition, the board provides an external 128-Mbit SDRAM (64-Mbit accessible), used to store the temporal images during the data acquisitions, a 4.3-inch 480x272 LCD-TFT color display, used for camera feedback view, and a 30 Flexible Printed Circuit (FPC) connector, used to connect the external camera module.

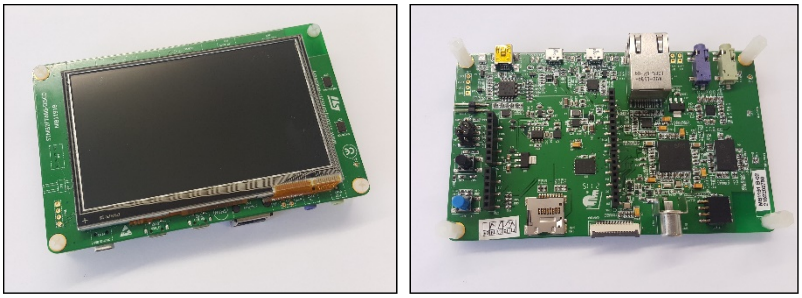

Fig. 1. STM32F746G-DISCO development board from STMicroelectronics [11].

2.2. Image sensor

The image sensor used is the Omnivision's OV9655 CMOS 1.3 Mega Pixel color sensor [12] which is integrated in a compact camera module (Fig. 2) along with optical lens and a 24 pin FPC connector for communication.

The OV9655 image sensor has two communication buses: 1) a serial camera control bus (SCCB) to initialize and configure the parameters of the camera and 2) a parallel output pixel data bus up to 10 lines, which is continuously transmitting images. The output format is settable to Raw RGB, RGB (GRB 4:2:2, RGB 565, RGB 555), YUV (4:2:2) and YCbCr (4:2:2) formats. The sensor has an active 1280 x 1024 pixel array but the image size can be selected between SXGA, VGA, CIF, and any size scaling down from CIF to 40x30 by subsampling. All configurations have a maximum transfer rate of 30 frames per second except for SXGA which is 15 frames per second. Exposure, gain, white balance, band filter and others image parameters can be configured through the SCCB bus. In the same way, the image quality parameters such as color saturation, gamma, sharpness, lens correction, white pixel cancelling, noise cancelling, and 50/60 Hz luminance detection, can be configured as well.

The camera sensor works with different power supply sources. Its core needs 1.8V if the internal regulator is disabled. Its analog lines need a source from 2.45 to 3V that can be used for the IO lines. When the camera is in normal working conditions its current consumption is around 20mA whereas in standby can reach 1 µA.

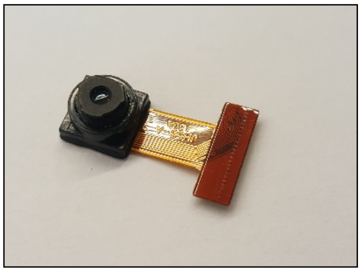

Fig. 2. OV9655 CMOS 1.3 Mega Pixel camera module from Omnivision [12].

2.3. Wi-Fi module

The selected wireless connectivity for the proposed device is Wi-Fi technology which allows to connect the device into a TCP/IP local area network (LAN) with many others common IoT devices. The used wireless module is the ATWINC1500-MR210PB low-power consumption 2.4GHz 802.11b/g/n FCC Certified Wi-Fi from Microchip [13]. It has an internal dedicate MCU to handle the TCP/IP stack that allows to transfer information through a simple 48MHz Serial Port Interface (SPI). It enables to transmit data at up to 48 Mbps in 802.11g/n mode. This throughput makes it perfect for streaming video in low-cost and low-power embedded devices. Fig. 3 shows the ATWINC1500-

MR210PB Wi-Fi module and the respective development extension board from the same manufacturer (ATWINC1500-XPRO).

The operation voltage of this wireless module is from 3.0V to 4.2V and the current consumption at 3.3V in 802.11n is 244mA for transmitting, 58.5mA for receiving and 380µA when the device is in Doze mode (the device is on but the communication is disabled). The module also includes a Power Amplifier (PA), Low-Noise Amplifier (LNA) and a Printed Circuit Board (PCB) antenna, giving a sensitivity up to -89dBm for receiving and 17.0dBm for transmitting in 802.11n mode. This fully integration helps to reduce the developing time and final manufacturing costs. In simple systems applications, the internal CPU of this wireless module is capable to act as main processor unit running a custom firmware. However, the external I/O pins and the processing capabilities are quite limited.

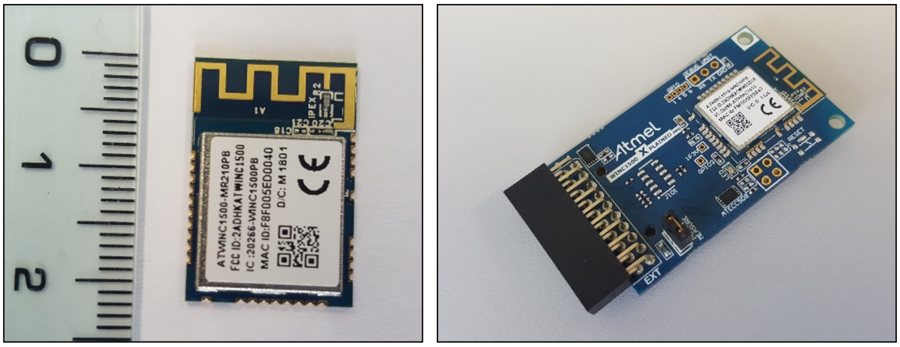

Fig. 3. ATWINC1500-MR210PB Wi-Fi module (left) and the ATWINC1500-XPRO extension board (right).

3. Architecture configuration

3.1. Image acquisition

One of the main objectives of this setup is to achieve a high speed data acquisition without using any CPU processing time. In order to do this, the MCU is designed with several hardware peripherals that can be configured to control the different parts of the process, keeping the CPU free to do image processing. The internal Digital Camera Interface (DCMI) is configured to capture images from the camera and store each pixel on a temporal peripheral register which is mapped by hardware into a memory region of the external SDRAM. It is done using the Direct Memory Access (DMA) that provides high speed block data transfer by hardware between peripherals and memories, and the Flexible Memory Controller (FMC) to translate bus transactions of an external memory to internal memory registers. Fig. 4 shows the hardware architecture diagram used for image acquisition.

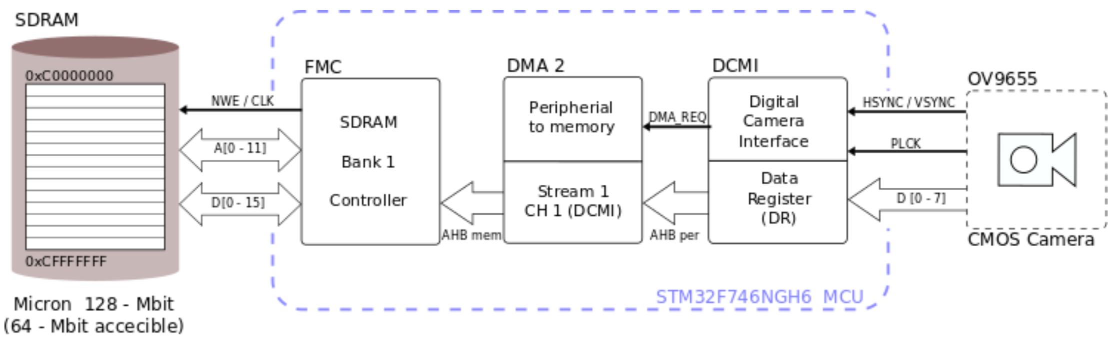

Fig. 4. Proposed image acquisition hardware architecture.

The image sensor OV9655 is configured to gather the data in RGB565 mode (each pixel consists of 16-bits) and send it through a 30 pin FPC as a constant stream running at 30 frames per second. The data is sent one byte at a time in an 8-bit parallel line together with the Horizontal Synchronization (HSYNC), Vertical Synchronization (VSYNC) and the Pixel Clock (PCLK) signals, and is received by the DCMI, where it is packed into a 32-bit register called Data Register (DR). A DMA request is generated each time the camera interface receives a complete 32-bit data block in its data register and it is sent to the DMA2, which initiates a transfer from the DR to an internal FIFO through the AHB peripheral bus. When the threshold level of the FIFO is reached the contents are drained and stored into the destination through the AHB memory bus. The data is handled by the FMC in order to translate AHB transactions in the appropriate external SDRAM protocol, then it is transferred in 16-bit blocks and stored into a 12-bit memory address.

3.2. Wi-Fi transmission

The proposed device is designed to be capable to process and transmit through the Wi-Fi each frame as fast as possible in order to have a streaming monitoring in real time. The Wi-Fi module is able to transmit up to 48Mbps. This is enough to transmit images with basic VGA resolutions as long as the MCU hardware guarantees the continuous transmission of data. Fig. 5 shows the hardware architecture configuration to transmit an entire image from the external SDRAM to the SPI bus, using the internal DMA. The goal of this scheme is to reach a maximum SPI throughput rate keeping the CPU idle for possible image processing tasks.

In the case of receiver mode, due to the small amount of received data, all the incoming information is just handled with an interrupt byte by byte. In the case of transmitting (Fig.5), it is used the reverse method as the image gather: the image data is taken from the external SDRAM using a FMC, and the DMA1 creates a memory map from the AHB memory to the DR of the SPI2 peripheral in order to transfer the entire image data to the Wi-Fi module in one DMA transaction. The SPI2 peripheral of the MCU acts as a master whereas the dedicated MCU of the Wi-Fi module act as a slave which translate the data received to TCP/IP frames using its internal software stack.

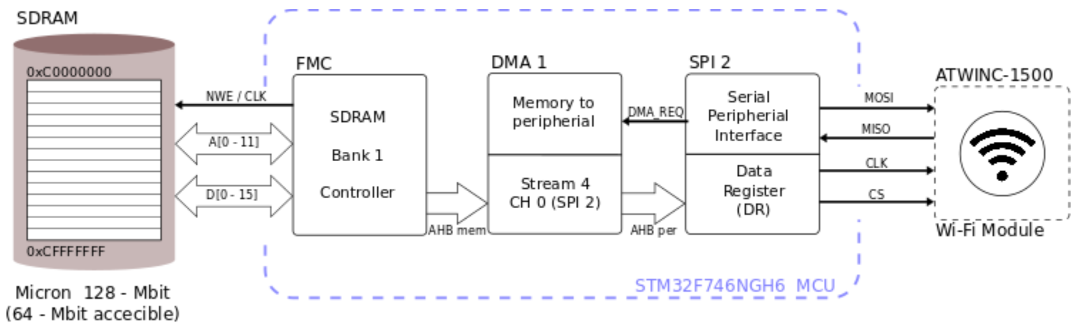

Fig. 5. Hardware architecture proposed for the wireless communication.

4. Results and discussion

The proposed device and hardware architecture has been tested and four performance experiments has been carried out with three different camera resolutions, 160x120, 320x40 and 640x480. Table 1 shows the device performance acquiring and storing images in real-time at these resolutions. The PCLK value depends on the resolution to maintain the maximum frame rate of the camera and the capture time of a horizontal line, HSYNC, also it is proportional to the resolution. A time gap of 1.33 ms between frames is generated. This time is considered an idle time for the acquiring system, which has already finished transmitting the current frame and has to wait for the next one. This timing could be used to run image processing algorithms. Considering these acquisition times, the measured frame rate matches the manufacturer's datasheet, which specifies a maximum of 30 fps for all tested resolutions.

Table 1. Capturing performance of the proposed hardware architecture.

Camera resolution	**PCLK (MHz)**	**HSYNC capturing time (µs)**	**Idle time between frames (µs)**	**Reached frame rate (fps)**
160 x 120 (QQVGA)	6.00	266.68	1331.68	29.88
320 x 240 (QVGA)	12.00	133.32	1342.82	29.99
640 x 480 (VGA)	24.00	66.67	1336.70	30.00

Moreover, the capturing tests corroborate that the MCU allows to perform all the data gathering, storing and transmitting at full rate leaving the main CPU with almost no load. It is possible thanks to the DMAs interfaces, used in almost all parts of the proposed architecture, which are designed to move blocks of data very fast by hardware. Thus, the CPU is free during the images transmissions and it can be used to execute image processing tasks, keeping the same frame rate. This proposal requires a double buffer algorithm to swap the memory area at each frame in order to have access (and process) an entire image while the next one is being acquired. In this case, the idle time for image processing tasks is increased from 1.33 ms to 33.3 ms (full time of image acquiring).

In this paper, an experimental test of this processing capability has been carried out. The CPU executes an algorithm to perform a basic red segmentation in order to detect objects while the images (the original and the result) are showed in the TFT-LCD at a maximum frame rate. Due to the limitations of the TFT-LCD size, the resolution used is the 160x120. Fig. 6 shows a picture of the proposed image acquisition system while it is executing the image progressing algorithm detecting a red piece and monitoring at full rate.

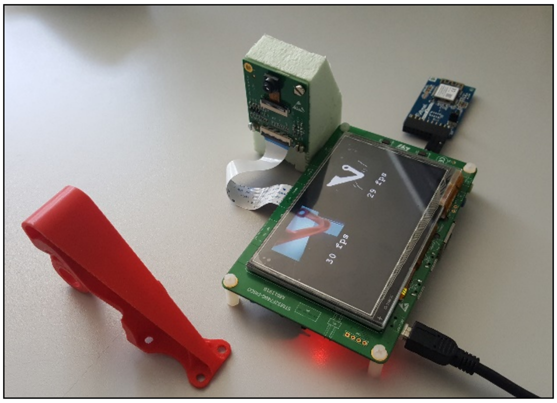

Fig. 6. The image acquisition system proposed detecting and monitoring a red piece in real-time.

Finally, the proposed wireless video system is compared in terms of cost and power performance against of other two shelf hardware solutions. The first one is the Raspberry Pi Zero W [12], which is based on a SoC Broadcom BCM2835 with an ARM11 CPU running at 1GHz and an internal 512 Mbytes of RAM, along with the Raspberry Pi camera module V2.1 that has a Sony IMX219 8 Mega Pixel sensor which supports up to 1080p (1920×1080) at 30 fps. This system is configured to work as a 2.4GHz 802.11n wireless Real Time Messaging Protocol (RTMP) streaming server using the FFmpeg GNU Software [15]. The second solution, is an off shelf wireless IP-surveillance commercial camera, the Edimax IC-3115W [16]. This camera has an integrated image sensor of 1.3 Mega Pixel with a maximum resolution of SXVGA (1280 x 960) at 30fps and a 2.4GHz 802.11n wireless network connection that allows to get the sensor data through streaming Motion JPEG (MJPEG).

Table 2 shows a comparison of these three types of solutions in terms of cost and power consumption at a maximum speed of wireless streaming. The Edimax IC-3115W is the most economical because it is an end-product and has a competitive price. In contrast, this solution has the worst power consumption which reaches up to 2100 mW. The Raspberry Pi has better image sensor with better power consumption regarding to our selected sensor, however this affects its price which is higher than the processor itself. In the case of the proposed device, the cost only considers the three main electronic components (MCU, image sensor and Wi-Fi module) because the development platform has many unused components. The approximated cost would not be much different when manufacturing a custom design and it is feasible due to the few required electronic components.

Table 2. Price and power consumption of three types of wireless IP camera solutions.

	Approximated cost (€)	**Power at full rate streaming (mW)**		
		Minimum	*Maximum*	*Average*
Our proposed device	**26.30**	**686.44**	**710.02**	**703.65**
STM32F746NGH6 MCU	*12.55*	*286.89*	*289.52*	*288.20*
OV9655 camera	*6.25*	*129.89*	*130.54*	*130.22*
ATWINC1500 Wi-Fi module	*7.50*	*269.66*	*289.95*	*285.23*
Raspberry Pi SoC solution	**43.90**	**1022.95**	**1257.48**	**1087.82**
RPi Zero WH board	*16.95*	*934.35*	*1168.24*	*997.94*
RPi Camera V2.1	*26.95*	*88.60*	*89.24*	*89.88*
Edimax IC-3115W IP camera	**22.40**	**1781.43**	**2100.79**	**1966.06**

Results show that the proposed device is the best solution in terms of energy consumption but does not differ much compared to the Raspberry Pi Zero W solution, which is more powerful. The goal of our proposal is that the MCU has many possibilities to work in low-power modes. Thus, if the final application not requires to be all the time doing streaming it would be possible to reduce its consumption significantly.

5. Conclusions

In this work a new low-cost and low-power hardware architecture for an embedded wireless image sensor node for IoT applications was proposed. The configuration of all hardware parts involved to the image acquisition and transmission was defined in order to achieve the lowest CPU use and the maximum camera frame rate. The hardware proposed allows to handle images at the maximum camera frame rate. During the acquisition and wireless transmission, all CPU time between frames (33.3 ms) can be used for real-time image processing without dropping any frame.

The architecture configuration of the proposed image acquisition system shows that the new ARM Cortex-M7 MCUs fits perfectly for low-cost image processing systems. The proposed hardware architecture is as good as a complex vision hardware system based on SoC, as long as the CPU performance limitation for a secondary process is not a problem. The proposed hardware system is more economical and consumes less power than the standard ARM-based SoC systems, like the Raspberry Pi Zero W, and also, can be more robust since it works without operation system.

References

[1] Mathe K, Busoniu L. Vision and Control for UAVs: A Survey of General Methods and of Inexpensive Platforms for Infrastructure Inspection. *Sensors* 2015; 5:14887-14916.

[2] Liu H, Zhou Q, Yang J, Jiang T, Liu Z, Li J. Intelligent Luminance Control of Lighting Systems Based on Imaging Sensor Feedback. *Sensors* 2017; 17(2): 321.

[3] Sarikana SS, Ozbayoglub AM, Zilci O. Automated Vehicle Classification with Image Processing and Computational Intelligence. *Procedia Computer Science* 2017; 114:515-522.

[4] Font D, Moreno J, Tresanchez M, Teixido M, Palleja T, Palacin J. Compact and low cost embedded vision system for fruit detection and tracking. *9th European Conference on Precision Agriculture (ECPA 2013)*; 2013 Lleida, Spain, 2013.

[5] Tian L, Wang H, Zhou Y, Peng C. Video big data in smart city: Background construction and optimization for surveillance video processing. *Future Generation Computer Systems* 2018.

[6] Qia J, Yang P, Min G, Amft O, Dong F, Xu L. Advanced internet of things for personalised healthcare systems: A survey. *Pervasive and Mobile Computing* 2017; 41:131-149.

[7] Joshi NB, Nalbalwar SL. A fall detection and alert system for an elderly using computer vision and Internet of Things. *IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*; 2017, pp. 1276-1281, Italy.

[8] Aliyu A, Abdullah AH, Kaiwartya O, Cao Y, Lloret J, Aslam M, Joda UM. Towards video streaming in IoT Environments: Vehicular communication perspective. *Computer Communications* 2018; 118:93-119.

[9] Obuchi Y,Yamasaki T, Aizawa K, Toriumi S, Hayashi M. Measurement and evaluation of comfort levels of apartments using IoT sensors. *IEEE International Conference on Consumer Electronics (ICCE)*; 2018, Las Vegas, USA.

[10] Othman NA, Aydin I. A new IoT combined body detection of people by using computer vision for security application. *International Conference on Computational Intelligence and Communication Networks (CzCN);* 2017, Girne, Cyprus.

[11] STM32F746NGH6 STMicroelectronics ARM Cortex-M7 Microcontrollers. http://www.st.com/en/microcontrollers/stm32f7-series.html (accessed April 13, 2018).

[12] OV9655 Omnivision 1.3 Mpx CMOS color sensor. http://www.ovt.com (accessed April 13, 2018).

[13] ATWINC1500-MR210PB 2.4GHz 802.11b/g/n Microchip Wi-Fi adapter. http://www.microchip.com/wwwproducts/en/ATwinc1500 (accessed April 13, 2018).

[14] Raspberry Pi Zero WH single-board computer. http://www.raspberrypi.org/ (accessed April 13, 2018).

[15] FFmpeg free software cross-platform solution. http://www.ffmpeg.org/ (accessed April 13, 2018).

[16] Edimax IC-3115W wireless network IP-surveillance camera. http://www.edimax.com/ (accessed April 13, 2018).