Wireless image-sensor network application for population monitoring of lepidopterous insects pest (moths) in fruit crops

M.Gonzalez, J.Schandy, N.Wainstein, L.Barboni, A.Gomez
Dept. de Electrónica, Facultad de Ingeniería
Herrera y Reissig 565, CP 11300,Montevideo, Uruguay

C.Croce
Juventud Melilla Cooperativa Agraria R.Ltda,
Cno. de la Redención CP 10102 Montevideo, Uruguay

***Abstract*— This article shows the implementation of a wireless image-sensor node capable of taking images of plagues that attack fruits crops. Images can be transmitted over a wireless sensor network in order to build the plague population database and to take appropriate countermeasures in case of infection. Wireless sensor networks are suitable to implement such application, however the technology is certainly conditioned by economic constraints and the tendency is to use low-cost commercial nodes with constrained hardware resources. When trying to build an application that involves acquisition, processing and transmission of images, it is often argued that commercial available nodes can not fulfill the application specifications. Reported approaches to tackle the problem are to design ad-hoc high-performance nodes by using enhanced hardware resources, clearly superior to those of commercial nodes. It increases the cost against the WSN philosophy (i.e. low-cost nodes and constrained hardware resources usage). Contrary to the established, this article shows the achieved design of a low-cost wireless image-sensor nodes that incorporate image handling capabilities with constrained hardware resources. The implemented node and the WSN have achieved all the application requirements. Results and conclusions are offered.**

I. Introduction

Wireless sensor networks (WSN from herein) are comprised of small devices named nodes. Each programmable node is composed of a microcontroller, sensors and a radio often IEEE802.15.4 compliant to communicate wirelessly with neighbors nodes. Their increasingly widespread usage is due to the following reasons: *i)* the node's low-cost enables to build distributed deployments easily scalable with large spatial density of nodes per unit area with reduced cost of installation and maintenance, *ii)* these deployments enable to build maps describing scalar fields varying in time and space (e.g. temperature, humidity), *iii)* the nodes operate with low current consumption, so that they can achieve several months to a year of battery lifetime (e.g. by using AA alkaline, 2800mAh). WSNs receive considerable research and development effort in order to enhance their capabilities to be used for precision-agriculture and environmental monitoring. It enables to manage the farm productivity variability, taking into account their specific needs, getting product quality enhancement with reduced operational cost. Most reported applications are of this kind [1][2], as well as the application object of this work. However, literature emphasizes shortcomings; WSNs can be used for environmental monitoring applications whereas the technology enables the low cost implementation. Key applications are restricted to monitor environmental scalar magnitudes because the node design is conditioned by economic constraints and the tendency is to use cheap commercial off-the-shelf nodes that have their own sets of sensors. The problem arises if the application requires to incorporate novel sensing capabilities; such as applications to handle images over WSNs. In that case, literature shows that custom high performance nodes are always designed. For instance, in [4] is described a custom node that integrates a C328-7640 VGA resolution camera module that generates JPEG compressed images up to 640x480 pixels. It has not been used low cost commercial nodes, neither the available operating systems for WSNs. The proposed system remains as proof of concept (it is not a solution for large scale) due to its high cost. In [5] is proposed a wireless image-sensor node (we define WimSN from herein) based on a video camera from Logitech with adaptive compression factor, a PC-104 Embedded module from AxiomTek and the MaxStream XBee XB24B Zigbee Modem. The video camera features 8-bit gray scale resolution of 256x256 pixels. This proposed architecture is quite far from the trend of using reduced hardware for WSN applications and this would prevent its large scale implementation. In [6] is presented an ad-hoc WimSN composed by the microcontroller ATmega644P, the XBEE radio module from MaxStream and the C328R camera from CoMedia. This CMOS camera integrates the JPEG compression engine. The camera features VGA, QVGA and CIF resolution (manufacturer has discontinued camera support). In [7] is used a node with a slave digital signal processor TMS320F2812 (DPS, Texas Instrument 32bit, 150MHz and 1Mbyte of SRAM) to manage the Omnivision VGA 640x480 color CMOS sensor (OV7640). In [8] is implemented a WSN application to monitor a vineyard. It is used a Hercules Webcam Classic with a resolution of 1.3-Mpixel (1290x960 pixels) and a special electronic board to rotate the camera. Finally, in [9] has been designed a WimSN by using the camera C328R from Comedia (manufacturer has discontinued support) and the node has been programmed by using an operating system (i.e. TinyOS) that is not suitable to manage images over WSNs. Therefore, it is clear that there is an absence in literature of methodologies as well as suitable design criteria to implement low-cost WimSN. The article is aimed to provide them. The design approach has been holistic; it has been studied the overall hardware tradeoffs in order to select the devices that meet the requirements and it is proposed the best setting for the selected communication protocol in order to enhance the WSN throughput.

II. Application description

The lepidopterous insect pest (moths) produces diseases in trees. The moths lay eggs from which larvae are born and they produce lesions on the fruit. The control of the pest population is implemented by means of using plastic traps with a sticky bottom side and pheromone lures. The trap can capture male adult moths attracted by the female pheromone lures. A person, who periodically travels through crops, is in charge of performing the counting of insects caught in the trap. Interested readers can see in advance Figure 4(b) that shows the inner bottom side of the trap with some captured moths. This article proposes the design of a WimSN capable of taking images in the inside trap to transmit them via radio channel. The system usage benefits are: *i)* the person should not go through the crops for counting trapped moths, it only should go to change the floor of the trap, *ii)* it automates the pest population monitoring by establishing images databases and *iii)* it enables early alert in case of pest infection thus allowing performing localized fumigation in the crop with reduced pesticide usage and hence avoiding environmental and water pollution.

III. Wireless image-sensor node

A. Mechanical constraints and requirements

The effectiveness of the trap for attracting and capturing the male moths depends on various factors, such as pheromones concentration per trap, but most important are the trap size, shape and physical form factor and geometry, specifically, the trap entrance. The incorporation of the WimSN and its protective enclosure has not changed these trap's features. The camera is placed at the appropriate height with respect to the trap's floor, in order to capture a complete image of the sticky surface due to its view angle. Figure 1 shows the 3D trap model that is being currently used, and where the WimSN is placed. The designed WimSN protective enclosure is low cost and of easy construction and quick maintenance. It is waterproof and it provides protection against the fine droplets of pesticides that appear during crop spraying. Figure 2 shows a detailed 3D picture of the enclosure.

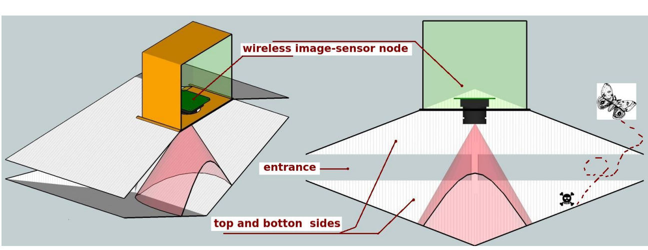

Fig. 1. A 3D trap schema with the camera and node on top. The camera takes images of the bottom side with trapped male moths.

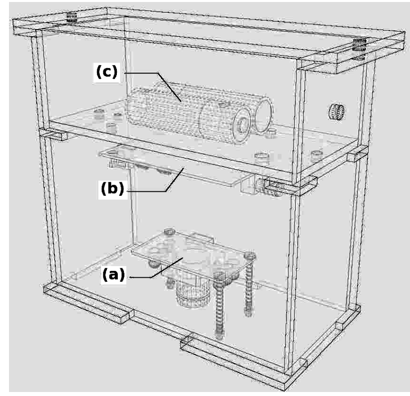

Fig. 2. A 3D schema of the enclosure with : (a) image sensor (camera), (b) CM3000 node telosb compatible, (c) batteries.

B. Hardware platform design

The application requirements are the following : *i)* battery lifetime above 6 months for each node, *ii)* image capture and transmission twice a day, *iii)* network composed by 10 nodes in 10 hectares and *iv)* enough image resolution to identify the male moth. It has been required to analyze trade-offs between performance and energy saving in order to achieve an optimized design. The hardware selection turned out to be difficult because of limited market offer where most available cameras with suitable resolutions require fast processors and are energy demanding. In order to meet the best design, the application bandwidth requirement, processing speed as well as battery current consumption and node duty-cycle have been carefully analyzed and tackled.
Requirement of large memory spaces to store temporary data has been taken into account. After assessing different hardware configurations, the proposed wireless WimSN consists in the following. It has been selected the node CM3000 [10] that features the microprocessor TI MSP430F1611 16-bit RISC. The memory program flash is comprised of 48KB, data RAM 10KB and it also provides an external flash of 1MB. The microprocessor has a 12 bit resolution built-in ADC (8 channels). The wireless communication is implemented by means of the RF Chip TI CC2420 (IEEE 802.15.4 2.4GHz standard compliant), with output power ranges from -25dBm up to 20dBm (by using the additional 2.4GHz RF extender PA-LNA CC2591 software configurable). Figure 3 shows the block diagram of the implemented WimSN.

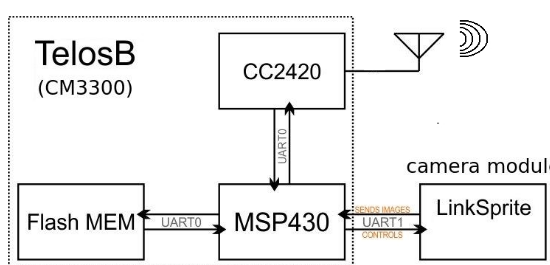

Fig. 3. Block diagram of the implemented wireless image-sensor node.

The radio coverage link ranges up to around 1000m (inside cultivation of crops) or 100-150m (indoor) with standard 5dBi antenna and RF extender of 20 dBm output power. The selected camera module is the LinkSprite JPEG Color Camera TTL Interface-Infrared SEN-11610 [11]. It needs DC 3.3V voltage power supply, the current consumption is around 80-100mA, VGA/QVGA/ 640x480 resolution, it integrates a JPEG compression engine, serial communication and 60 degrees viewing angle. The camera takes JPEG images with variable compression ratios at VGA resolution, and stores them into the memory buffer while data blocks can be transmitted on demand to the node through serial bus.

C. Software platform: criteria and implementation

The available operating systems for WSN are based on the abstraction layers software model, where each level gives services to upper layers through interfaces. By using this software model, the application is written by using customized interfaces and functions through which the application becomes platform-independent and consequently reusable. For this reason, there must be a balanced compromise between operating system features (e.g. kernel model, memory management, reduced code footprint) and application requirements. It has been performed an analysis of the available operating systems taking into account this considerations, particularly features as concurrency, flexibility, preemption computation, estimated energy consumption and efficiency with limited hardware resources. The options took into consideration were Contiki [12] and TinyOS [13] and the first one was selected. Both operating systems can run on platforms with constrained hardware resources, but Contiki fulfills concurrency requirements with the event-driven and multi-threading model while TinyOS offers only the event-driven model. Moreover, Contiki is a dynamic and modular operating system that facilitates allocation and deallocation of hardware resources at run time. Conversely, TinyOS is a static and monolithic system and the application must allocate all of the hardware resources at programming time, hence Contiki provides more flexibility than TinyOS. To be sure of selecting the suitable operating system, it has been assessed the performance of an algorithm for image processing, specifically a spatial filtering (mask). An image (size 150kB) stored in flash memory that features patterns has been filtered by a two-dimensional convolution algorithm implemented into the node for benchmarking. The resulting image was as expected and the algorithm execution time has been measured and it is approximately 2 minutes which is acceptable for our application. The main observed implementation problem was the lack of memory RAM (10kB), leading to the need of swapping data with the Flash memory when performing the convolution between the stored test image and the mask. The swapping is implemented by using the Contiki Coffee file system module [12] (the module availability is an advantage of Contiki with respect to TinyOS). The storage of both images of 150 kB each (the original and the filtered one) has been performed by means of typecasting sub-matrices of fixed size into an linear array. The convolution has been implemented by using masks with zero padding. It is clear that working inside the node with images in the pixel domain requires writing a very optimized firmware in order to reduce the algorithm execution time and to properly manage the constrained hardware resources. This can be done by using Contiki, but it has been decided to use a camera module that includes a JPEG compression chip. It decreases the image size and the memory space needed and it results in better network throughput. A driver for the selected camera module has been written in C to run under Contiki. It uses the UART library to implement the serial communication with the camera by using serial commands (e.g. take photo, change size, download image). The camera responds giving the data requested or communicates acknowledgments. The camera module transfers the image from its internal memory to the node by means of segmenting the image in fixed blocks which sizes must be integers multiple of 8. Each incoming block is stored in memory Flash by using the Contiki Coffe file system. A counter field keeps track of the number of delivered bytes in order to detect the JPEG string ends. Once the whole image is in Flash memory, the node can send the image through the network towards the node sink which holds a database. The JPEG compression rate can be set by programming the camera in any moment, enabling minimum or maximum compression ratio according to the requirements. With a resolution of 640x480 pixels and the average compression ratio, the image size is 50kB. By using higher compression ratios, image sizes of about 18kB can be achieved, without substantial quality loss. The transmission rate between the camera and the microcontroller can be set up to 115000 baud.

D. Communication protocol: criteria and implementation

One of the main application challenges is to reduce the lost frames of the segmented image conveyed by the network. For this reason, it has been assessed several network communications protocols in order to select the suitable one. In this application, the node wakes up just two times per day (at sunrise and sunset). The node takes an image each time, it activates the network protocol in order to send the image and it goes into sleep state after that. Therefore, the network protocol is enabled twice a day and the rest of the time the node sleeps with reduced battery current consumption. Based on this particular functionality, it has been selected the Rime Communication Stack [12]. It is a layered communication stack for sensor networks designed for the Contiki operating system. It provides the communication primitives that are required for this applications with the advantage of using much less code than other WSN communication protocols. Such primitives are: *broadcast*, *unicast*, *stunicast* and *runicast*. The *broadcast* primitive sends packets to all neighbors with a header that identifies the sender. The *unicast* primitive adds the single-hop receiver address attribute to the outgoing packets. For incoming packets, the *unicast* module inspects the single-hop receiver address attribute and discards the packet if the address does not match the expected one. The stubborn single-hop unicast primitive (named *stunicast*) repeatedly sends a packet to a single-hop neighbor using the *unicast* primitive. The *runicast* primitive uses acknowledgements and retransmissions in order to ensure that the neighbor successfully receives the packet. When the receiver has acknowledged the packet, the *runicast* module notifies to the sender application via callback. In this way, the *runicast* primitive becomes in *stunicast* to enable retransmissions in order to provide communication reliability. In most applications the routing protocol is permanently running in order to synchronize the nodes and to achieve reduced duty-cycle. In Contiki, the radio duty cycle is managed by the RDC layer (Radio Duty Cycling layer). As the application requires high throughput during very short periods of time, we do not use the RDC mechanism. To achieve this, it has been disabled the *ContikiMAC driver* (set by default) and it is used the *NullRDC driver*, thus setting the radio in awake mode during the whole transmission.

For the chosen compression ratio, the images have an average size of 18kB. The maximum payload size that can be sent using the *runicast* primitive is 100 bytes, resulting in approximately 180 segmented-frames. The assessment shows that the transmission time is of around four seconds by using *NullRDC driver* (throughput of 36.86kbps) and is of around 90 seconds by using *ContikiMAC* (throughput of 1.56kbps). Finally, the routing protocol is set as static routing protocol over a tree topology network. Each node sends the image during a time slot using *runicast* in approximately four seconds. The image data transmissions are performed by multihop towards the sink node. To prevent a possible failure at any node in the relay chain, each node has a second routing address.

IV. Wireless image-sensor node: image handling

To have better insight of the obtained results, Figure 4(a) shows the inner bottom side of a trap with a calibration image pattern. The selected low-cost camera produces curvature near the image boundary but it does not affect the performance of the application (male moths can be identified). Figure 4(b) shows the inner bottom side of a trap with some captured insects. The diffusive illumination comes from leds placed inside the trap. Such leds are managed by the node. Decoding the JPEG image to the pixel format is quite difficult to be implemented in the node due to the reduced computational node resources. Instead, a partial decoding and processing in the compressed domain was implemented within the node. As final goal we pretend to elaborate the images in the node, counting the number of moths, in order to send such information to a central computer. This prevents sending the whole image that can produce congested traffic due to multi-hopping communication among nodes.

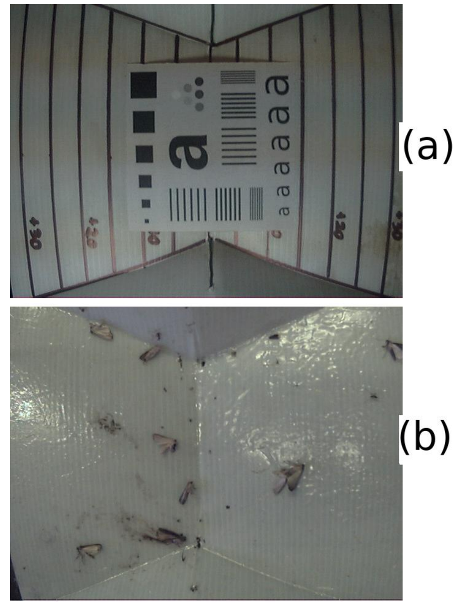

Fig. 4. **(a)** The inner bottom side of a trap with a calibration image pattern (image 640x480 pixels, 47kB). **(b)** The inner bottom side of a trap with some captured insects (image 640x480 pixels, 47kB). The image resolution is enough to count the pest specimen number.

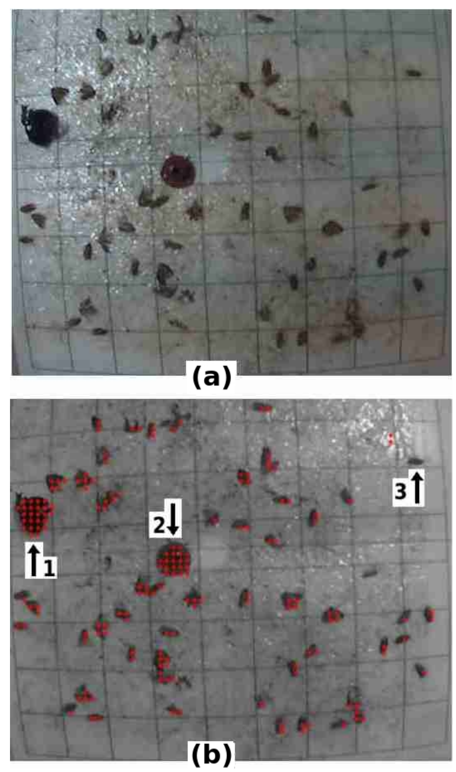

Fig. 5. **(a)** The inner bottom side of a trap with moths, **(b)** the algorithm can localize and count the pest population (moths). It is marked with red points to visualize them but into the node is implemented the automatic counting. There are elements that the algorithm classifies as moths as is marked by 1 and 2. Some moths are not well classified as is marked by 3.

A Fisher Linear Discriminant (FLD from here onwards) has been programmed into the node. The algorithm is explained in a companion article because the implementation is out of scope of this article. Nevertheless, an overview of the results are shown in Figure 5(a) and (b).

Figure 5(a) shows the inner bottom side of a trap with moths. It is possible to observe that the algorithm can localize and count the pest population as we can see in Figure 5(b), however it has been detected a minor drawback on the obtained classification capacity. For instance, it is counted as moths the following: *i)* a bee (marked with 1) and *ii)* the pheromone pill marked with 2. Moreover, few moths are not classified such as indicated by mark 3. The problem lies in to improve the statistical information used by the algorithm, thus enhancing its learning to classify image frames. In spite of this problem, the proof of concept as well as positive assessment of the WimSN have been well established.

V. Battery current consumption and lifetime

We now turn attention to the battery lifetime estimate. The measure of average battery current consumption when the microcontroller operates in LPM3 mode is $147\mu A$ (the LPM3 mode is the idle microntroller operating mode with minimum functionality). In such case the radio transceiver, the external FLASH memory and the image sensor are switched off. Moreover, the measure of average battery current consumption to set the camera and to take an image is $63.5mA$. On the other hand, time measurements are the following: *i)* the image sensor (camera) initialization followed by the time it takes to acquire an image is $19s$, *ii)* required time to send the image to the neighbor node is $5.5s$. In consequence the average battery charge consumption to set the camera and to take an image is $0.327mAh$. If a couple of batteries with effective charge $2800mAh$ provide the voltage power supply to the node, the obtained lifetimes are: *i)* in case of taking one image per day, 723 days and *ii)* 663 days if two images are taken per day.

It is important to underline that this estimation for the battery depletion time has been performed at single node level. Complete measurements have to be performed on the deployment because the routing protocol (node-hopping communication) introduces non deterministic current consumption behavior that can modify such estimate.

VI. Conclusion and final remarks

This article shows and discusses the implementation of a WimSN capable of taking images of plagues that attack fruits crops. It enables to implement early alerts systems in case of pest infection thus allowing performing localized fumigation in the crop with reduced pesticide usage and hence avoiding environmental and water pollution. It allows assessing pest populations variability due to climate change and to identify the crops genetically more resistant to the lepidopterous insect pest (moths). The achieved design is low cost (around U$S 140 all included), it is of easy construction and quick maintenance and it enables to change the trap's floor while keeping the superior structure. It fulfills all requirements asked by the entomologists partners of this project. The implemented WimSN achieves all application requirements outperforming solutions given nowadays in literature.

Acknowledgment

The authors are grateful for the support of CSIC - Universidad de la República and INIA - FPTA.

References

[1] Pierce, F.J., Elliott, T.V; *Regional and on-farm wireless sensor networks for agricultural systems in Eastern Washington*, in Computers and Electronics in Agriculture (Elsevier) Volume 61, Issue 1, April 2008, Page(s): 32-43, ISSN: 01681699

[2] López, J.A. et.al; *Development of a Sensor Node for Precision Horticulture* in Sensors 2009, 9, 3240-3255; doi:10.3390/s90503240 (www.mdpi.com/journal/sensors)

[3] I.F. Akyildiz, T. Melodia, K.R. Chowdhury; *A Survey on wireless multimedia sensor networks* Computer Networks (Elsevier), vol.51, no. 4, pp. 921-960, Mar. 2007.

[4] A.A.Shahidan, N.Fisal, A.H.A.H.Fikri, NS.N.Ismail, F.Yunus; *Image Transfer in Wireless Sensor Networks* in International Conference on Communication Engineering and Networks IPCSIT vol.19 (2011) IACSIT Press, Singapore

[5] G.Nikolakopoulos, D.Kandris, A.Tzes; *Adaptive Compression of Slowly Varying Images Transmitted over Wireless Sensor Networks* Sensors 2010, 10, 7170-7191; doi:10.3390/s100807170 . www.mdpi.com/journal/sensors

[6] A.H.F.B.A Hamid, et.al.; *Development of IEEE 802.15.4 based Wireless Sensor Network Platform for Image Transmission* International Journals of Engineering and Sciences- 98210-5656 IJET-IJENS

[7] J. Karlsson; *Image Compression for Wireless Sensor Networks* Master Thesis in Computing Science, Umea University, Department of Computing Science SE-901 87 UMEA SWEDEN

[8] J.Lloret, I.Bosch, S.Sendra, A.Serrano *A Wireless Sensor Network for Vineyard Monitoring That Uses Image Processing* in Sensors 2011, 11, 6165-6196; doi:10.3390/s110606165, www.mdpi.com/journal/sensors

[9] P.Tirelli,N.A. Borghese, F.Pedersini,G. Galassi, R. Oberti; *Automatic monitoring of pest insects traps by Zigbee-based wireless networking of image sensors* in IEEE Instrumentation and Measurement Technology Conference (I2MTC), 2011 ISSN : 1091-5281

[10] [Online]:http://www.advanticsys.com/

[11] [Online]:https://www.sparkfun.com/products/11610

[12] [Online]:http://www.contiki-os.org/

[13] [Online]:http://www.tinyos.network