

Zigbee Image Transmission for Detecting and Tracking Multi Color Moving Object

M. Chakravarthi* and D. Aruna Kumari

Department of Electronics and Computer Engineering, KL University, Vaddeswaram, Guntur – 522502, Andhra Pradesh, India; chakri.39.3@gmail.com, aruna_d@kluniversity.in

Abstract

Objective: To design and implement a Zigbee Imaging system to track and detect the multi colored moving objects using Additive color model technique. **Analysis:** Digital image processing methods like Segmentation, background subtraction and point detectors are very popular and prominently used in the object detecting and tracking from live streaming frame sequences. These methods require high computing systems to process the data in real time. **Methodology:** To overcome such constraints, we implement the low cost IEEE 802.15.4 Zigbee based Image Processing System. This system can process and transmit the images efficiently even in Dynamic environment conditions. In this paper, we illustrate Zigbee Imaging system functionality at sensitive places of observation. **Finding/Improvements:** We implement our proposed system with a Low cost CMOS color camera and Advanced RISC processor, which are used for supporting the Zigbee imaging system in detecting and tracking the objects. The system operates at 30 fps with resolution 88x144. This system can detect and track 8 objects at a time in the camera field view and transmits the tracked information to the host system with wireless Technology 802.15.4 Zigbee. The Experimental results of Prototype system to image sequences are also figured.

Keywords: Additive Color Model, Multi Color Moving Object, Object Spying, Zigbee Imaging System

1. Introduction

We have an assortment of utilizations in implementing a basic system based on image processing algorithms for detecting and tracking the objects¹⁻³. Applications in the field of security and surveillance systems monitor the objects automatically based on their specific features. Object perception for tracking the object is done by using some image processing algorithms which recognizes and tracks the object based on the primary features like color and shape. But restrictions like Cost and Complexity to implement such algorithms on Hardware are needed to be considered. Out of these algorithms, recognition of the color can be achieved with the assistance of a Digital image processing algorithm, called additive Color Model scheme, which greatly reduces the complexity and also works in a cost effective manner⁴.

It is always a challenging task to achieve adequate performance in image processing with constrained amount of computing resources. Generally, such systems require high end processor, huge amounts of Random Access Memory (RAM) for storing and processing the images and consume enormous amounts of power while processing the images. Basic image processing systems conventionally comprise of a camera, a frame extractor, and a related computer to interface this frame extractor to complete the execution of the algorithm. But the subsequent result is quite difficult and extravagant to the developer, where he works on a system limited image functionalities. Now it is possible to make the necessary changes to scale down the price of those systems by adapting new technologies. Low Cost, high speed Microcontrollers and Cost effective CMOS color camera

*Author for correspondence

modules are two technology improvements which we've taken into this work. The main advantage of CMOS over CCD camera technology is that CMOS can integrate Analog to Digital converters and related pixel extracting circuitry on the same die on the sensor itself. So, we don't need to interface a separate frame extractor in the system. In recent times low cost and high capabilities of controller makes it possible to implement simple pixel processing as the pixel values of the camera are scanned and driven out, making an extra and an unused frame buffer in many situations⁵.

The complex computation methods which were implemented in image processing systems require a large number of computing elements. It is difficult to use the huge number of computing elements in real time image processing applications. Most of the image processing systems transmit the image to the host system using an extra communication medium such as cables. Subsequently, it will be added to the overall cost of the system.

In this paper, we propose a low cost and reliable Zigbee imaging system implementation and its performance evaluation in detecting and tracking multi colored moving objects in the ambience. Light illumination levels and noise levels in the background are the most common problems which arise during the tracking of these objects. These problems can be resolved in this Zigbee imaging system with latest advances in camera technologies where it can be adjusted as per the ambient lighting conditions. A Low Cost IEEE 802.15.4 Zigbee is employed as a communication medium for facilitating the transmission of the raw image data to the host system⁶.

2. System Design

Our Zigbee Imaging system is designed to track multi colored moving object by extracting high-level information from a camera image. This information is sent to a Host System Application for maintaining the tracked information of the multi color objects. The object color which is intended to track is selected from host system application, specially designed for imaging system. For instance, the particular bounded set of RGB (Red Green Blue) values is specified in the host system application and these values are sent to the imaging system via Zigbee module that we are implementing as a communication medium. The Zigbee Imaging system employs three different modules for achieving the efficient performance.

Each individual module of the system is explained in the following sections.

2.1 System Hardware

The basic Hardware of the imaging system comprises of three components. The first two components, RISC integrated Atmega8 and CMOS C3088 Digital cameras are vital components of our system and an Xbee Zigbee module for supporting Zigbee Image Transmission. The data bus, configuration bus and synchronization pins are directly connected to At mega 8 controller without any complex logic for simple and reliable design. For processing the data in Real time, Atmega8 controller waits for the data coming from the camera continuously. Then the controller transfers the extracted high level data, i.e., image to the host system by using asynchronous serial interface which is implemented in embedded software. Operating voltage for imaging system is 5 volts and consumes about 57 mA of current. The image information to imaging system is given by a C3088 Digital Camera. It is a 1/4" Digital output Color camera module. Omnivision's CMOS image sensor 6620 is integrated into C3088 Camera⁷. Integrating this CMOS innovation with an Image processor is a simple to utilize computerized interface which makes C3088 camera integrated system a minimal effort solution for high quality video-image application. 8/16 bit-wide image data stream is continuously supplied by digital video port for tracking the object in real time. All camera functions, such as exposure, gamma, gain white balance, color matrix, windowing are programmable through I2C interface⁸. It also includes a 4.9 mm F2.8 lens for zooming purposes and 17 MHz on-chip Clock crystal for synchronization purpose. CMOS camera runs independently to transfer 8 bit RGB or YCrCb color Pixels through an 8 bit or a 16 bit sized data. To intimate the start of new frames and horizontal lines, Synchronization signals, Pixel clocks are being used. The Camera resolution of 352x288 at 60 fps refresh rate is the maximum attainable image frame size⁸.

To use video information from the CMOS C3088 Camera, One must legitimately instate the camera and afterwards stay synchronized with each of its output signals. The camera must be initialized properly to get the video data from CMOS C3088 and continue the synchronization signals for real time buffering. The controller, which is used to process the data, is Atmega8. It is RISC processor and instruction execution rate is 1 MIPS per MHz It has internal flash EEPROM of size 512 bytes,

programmable memory of size 8 kilo bytes and SRAM of size 1 Kilo bytes. It's Asynchronous serial port is connected to the Zigbee Xbee module which supports point-to-point and multi-point networks. Xbee module is the main module for the Zigbee imaging system which is configured in Transparent Mode (AT)⁹. Configuration details of Xbee module includes baud rate, Network address (PAN ID), Destination device short address. After configuring the device, it starts communicating with destination device. It can establish communication with destination device and transmit the data up to a range of 300 feet with Line of Sight¹⁰. The Real time performance in detecting and tracking can be achieved with our proposed hardware design. It is also possible to accommodate multiple host system with a single imaging system for broadcasting the tracking information. The hardware of the Zigbee Imaging system is given in the Figure 1.

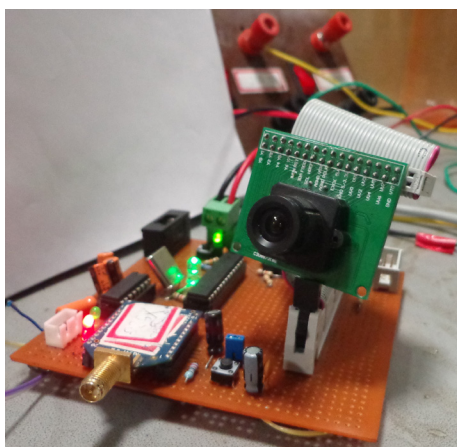


Figure 1. Hardware snapshot of zigbee imaging system.

2.2 Embedded Software

A set of functions in the source code of the system makes it easy to interact with CMOS image sensor at hardware level. Any system with some Imaging processing capabilities can make use of these functions to perform custom image processing. This embedded software utilizes around 4K of the 8K on-chip flash program memory, 700 bytes of 1K on-chip SRAM, and 48 bytes of the 512 bytes on-chip EEPROM. While developing the software to the system, the main thing that we have to consider here is the availability of SRAM in Microcontroller. SRAM availability after programming the software is only 324 bytes. It is beyond the SRAM size to buffer an entire color image at a time. To stream the image data from the camera without any buffering, we have to consider the limit in SRAM availability. We were able to perform basic image

processing during the time gap between pixel extractions by utilizing the data that streams from the camera. Image processing functions drop the trailing G component of the sensor's RGBG pixel group and skip every other G pixel to reduce processing time. This Procedure restricts the horizontal resolution to 176 RGB pixels, but sets limit on the maximum vertical resolution of 144 pixels. Additional post processing and transmission of data is done in the time gap between the end of each row and at the end of each frame. The imaging system is capable of implementing Additive color model scheme for attaining these required parameters of the system⁴.

Coding to for integrating this hardware is written in C Language and a target hex file is created using the Atmel studio compiler. After compiling the firmware, it takes 4K words of ROM and at some point; it is utilized for storing image pixels. RAM availability becomes very less after executing the program. Hence, it is needless to say that the firmware has to be coded very carefully. We can see embedded software block diagram in the Figure 2.

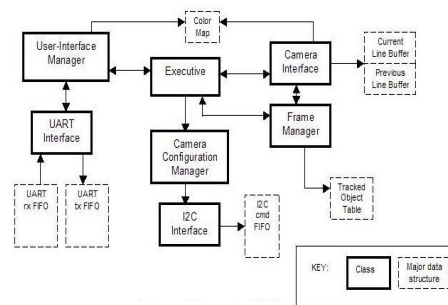


Figure 2. Embedded software architecture.

Commands for a typical Zigbee imaging system are formatted in ASCII operation-code (op code). Each command is framed with Two ASCII characters followed by an optional parameter and carriage-Return character. Whenever the command is given to imaging system, then it should respond with either a positive Acknowledgment (ACC) indicating the command was successful or a Negative Acknowledgment (NAC) indicating that the command is unsuccessful.

For instance, if the user wants to check the Zigbee imaging system to ensure that the system is functioning properly or not, PN\r (PING) command is sent to system, and then it responds with a Positive acknowledgment (Acc). The host system confirms that the imaging system is alive and operating properly, because it has responded with a positive acknowledgment (ACC). This system can

also give its name with a Command request GN\r (GET NAME). The response to this command gives the normal acknowledgment followed by the Name of the System. Likewise, the software of the system is coded with simple command requests and responses for interacting with the user to ensure its proper functioning.

2.3 Color Detection Method

The Additive color model, also known as RGB color model is implemented in embedded software as a special function. This function acts as color recovery function and maintains a running sum of the independent color channel components. Upon completion of the frame, the mean color returned by summing of all the pixels is divided by accumulated pixel values. It additionally gives back an estimation of the total deviation from the mean of every color. The color statistics can be used as a building block for determining the color of an object or detecting motion in a particular area in the field of view. Also allows the user to enter a range for either each of the three RGB or YCrCb color channel values depending on camera configurations. After receiving the color boundary values, each pixel is compared with received color range of values. For simplifying this task, Atmega8 controller stores the global information like Image raw data and bounding box coordinates of the Frame. The global data contains the upper left x1, y1 coordinates and the lower right x2, y2 coordinates which include pixels that satisfy the color bounds. The coordinates of the pixels close to the color bounds are compared against previously stored coordinates to generate the bounding box. Likewise, we also count how many pixels come in the boundary limits of actual color. Once the complete frame has been processed, further post processing operations are completed specifically. A scaled ratio between the calculated area and the total sum of pixels within the color boundaries is computed by considering the bounding box. This value can then be used as reference value which intimates whether there is only one object or multiple small objects being tracked, which occupies the bounding box. The x and y Coordinate positions of the each detected object is acquired by the system after an entire frame has been received. By using these values, system can return the number of detected pixels, coordinates of bounding box and reference values of the bounding box.

2.4 System Modes and Camera Functional Settings

Zigbee imaging system has three different modes apart from the normal command response protocol, which can be initiated by sending commands to the imaging system, then after it transmits some additional packets to the host system asynchronously. Each mode takes the system into different functionalities for distinctive cases. The default mode is idle mode and it is first mode for the imaging system. In this mode, Host system does not receive any packets from the imaging system. The Frame dump mode is second mode. The system is entered into this mode by sending DF\r (DUMP FRAME) command. In response to this command request, imaging system starts sending Dump packets to the host system until the entire image frame has been sent. Once all dump packets for an entire frame have been sent, the imaging system switches back to idle mode. The color tracking mode is the third mode. The system enters this mode by sending TS\r (TRACKING START) command. Then it starts the tracking and sends this tracking information to the host system. All these would happen only when requested command is received properly and positive acknowledgment is sent back to the host system. This tracking packet transmission takes place during the time that system in the Tracking mode. Tracking information packets are not generated when the system is not able to detect the object and stays in tracking mode till we send the command to disable the tracking mode. We should send TP\r (TRACKING STOP) command to the imaging system. This command switches the system back to Idle Mode. The hexadecimal 8-bit values range between 0x00 and 0xFF are transmitted to Host system in the well defined packet format. These tracking packets are not easy to comprehend with a normal serial terminal emulator. We can see the information in software specially designed for the Zigbee Imaging system.

To detect and track the object at super sensitive places, it is important to adjust the camera functional settings as per the lighting conditions. It will have a major effect on the real capabilities of the system. For instance, reflective objects appear as high intensity white spots as light fall on the reflective objects irrespective of color of the object. We can adjust the camera lens and its parameters as per lighting conditions. The parameters that are possible to adjust by sending commands to the imaging system from serial terminal or GUI are auto-adjust mode, Auto-white

balance, and fluorescent light filtering as shown in Figure 3. Then the camera is able to detect the object even in the poor lighting conditions.

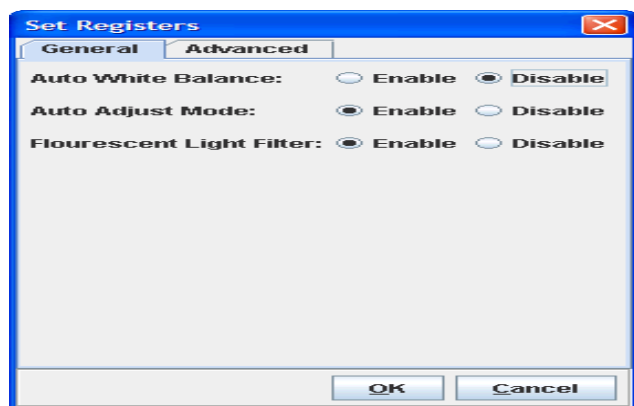


Figure 3. Camera registers window

3. Implementation

A user can interact with imaging system through serial protocol. This imaging system uses a Human understandable ASCII code format. The system is configured to communicate with the Zigbee imaging system at 115.2 kilo baud rate. This system can also dump the raw image data while communicating with host system through serial port interfaced with Zigbee module.

Frame dump does not occur in real time due to the high data rate required. Instead, the imaging system will send each row of image data per frame to the host system. The maximum frame size of 176x144 is the default resolution and dumping this entire frame takes about 4 seconds.

3.1 Serial Terminal

The most straightforward approach to begin with the Zigbee imaging system user interface is to utilize a serial terminal emulator running on a Host System (such as HyperTerminal, or Real Term). The serial terminal emulator ought to be set up to keep running at a baud rate of 115.2 kbps, 8 data bits, 1 stop bit, and no flow control. Observe that the terminal is just sending a carriage return/line-feed when the enter key is pressed. This permits the user to send basic commands to the imaging system and see the command responses sent by the system to the host system serial terminal.

For instance, if we give the PN\r command to the imaging system from the terminal then the response obtained would be as shown in the Figure 4. Likewise,

Name of the system can also be requested from a terminal window. To get the image Raw data, we should send DF\r command. This Raw data ranges from 0x00 to 0xFF hexadecimal 8-bit values are transmitted to the system in a well defined packet format as shown in the Figure 4

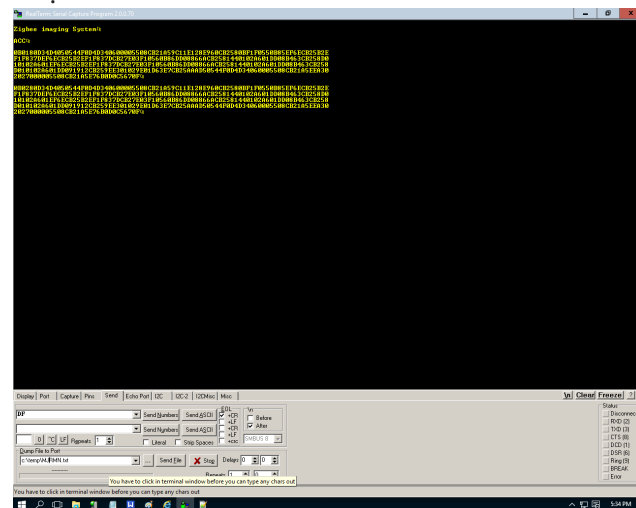


Figure 4. Serial terminal window with command responses.

The multi color moving object tracking is enabled by giving the command TS\r. Then system will send the tracking packets in hexadecimal format. Each packet contains tracking information of the color object. We should send TS\r before changing its mode of operation. The Zigbee imaging system is capable of tracking up to eight objects per frame. It may be zero, if the objects are not detected. All the traceable objects information in the camera field view will be framed as packet and returned to host system as color-tracking packet. We can also adjust the camera functions like white balance, gamma and exposure from serial terminal with CR\r (CHANGE CAMERA REGISTER) command and the corresponding values. We can observe the changes in Tracking information and the raw image data which is sent by the imaging system.

3.2 GUI

A Java based Graphical User Interface (GUI) helps in system integration, which makes easy for the user to communicate with the imaging system from a Windows based PC. This GUI enables you to explore all the functions of the camera from user familiar environment. The GUI clearly shows real-time information from the camera in a common way. For instance, the color pixel

to be tracked is picked from captured image by clicking on the enable track button. Then GUI frames user command request and sends it to the camera via serial port connected to it. The camera output data is determined and viewed in a GUI window that displays the actual bounding box along with tracking color. The color position depends on the reference value that is returned. Depending on the camera configuration, binary image of tracking mode and centroid may also be superimposed on the bounding box. When user calls the image recovery function, then returned color image data is mixed and displayed. The standout component among the most essential components of the GUI is its “capacity” to show dumped image.

To test the GUI Host system application, we first send the PN\r command by clicking on the Radio button of GUI user interface. Message viewer of the GUI shows the command request and its response. After getting the Positive acknowledgment from the system, we press the capture button in the command bar to get the entire frame of the camera field view and display it in the frame window as shown in the Figure 5. It takes 4 seconds to complete the image acquisition. The camera module lens needs to be adjusted to get the clear image without any blur if the image is not clear.

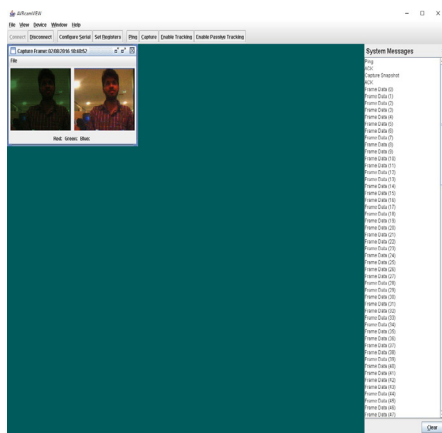


Figure 5. Captured image window.

After capturing the image, we can pick the color of the object that we want to track by moving the mouse pointer over the color of interest in the image window. We can observe the variation in RGB values in the image window whenever the mouse pointer moves over the image of different colors. To track the color of our interest, we add the colors in GUI color map as shown in Figure 6. This color map allows us to efficiently map up

to 8 different colors. In this way we can add 8 different colors of our interest.

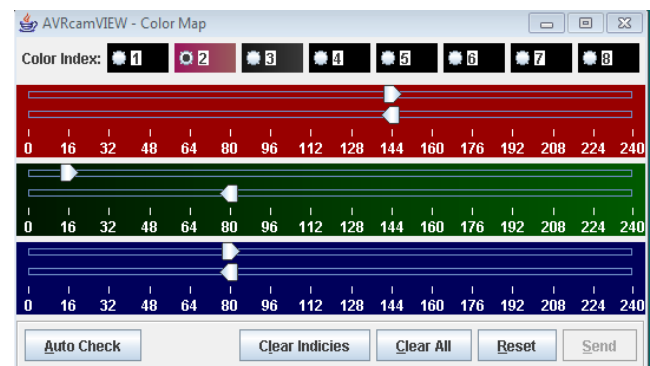


Figure 6. Color map window.

We can start the object tracking by clicking on “Enable Tracking” button, after setting color to the color map. All command bar options are disabled while system is in tracking mode. We can see the detected and tracked objects along with the color in the center of the bounding box of tracking window. Based on colors in the color palette, as tracking objects move through the imaging system field view, then real time tracking windows updates to show that the object is being tracked and it is depicted in the Figure 7.

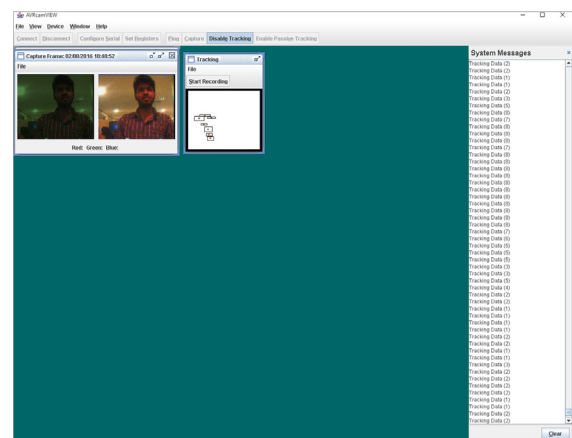


Figure 7. Object tracking window

The communication between the Zigbee imaging system and PC can be seen in the Log maintained by the GUI host system application. It shows the time stamp to know when the object is identified. This tracking information file can be in a text file for future reference. Disable tracking button in command bar needs to be clicked to disable the tracking and access entire command bar again.

4. Performance Evaluation

The System is designed to track the multi color moving object based on Zigbee image transmission and it is working as per requirements and conditions we've defined in the previous sections. Once, the system receives the color values needs to be tracked then it starts tracking the objects. This System covers the field area up to 35 feet away in the camera field view. It is also able to receive and transmit image information from both Serial terminal and GUI. Tracking Image transmission takes place at a rate of 30 frames per second with a maximum resolution of 88x144 is attained as per the defined requirements. The System is designed and implemented as per the expectations and is successfully tested for the functionality.

5. Conclusion

The objective of this work is to design and implement Low cost Zigbee Imaging system for tracking the multi color moving objects at super sensitive places in dynamic environment conditions. We've carried out checking the integrity of building a Low cost basic Imaging system comprising exclusively of a microcontroller and a CMOS camera chip. It should actualize straightforward algorithms at a reasonable frame rate. Tracking multiple objects at various conditions at 30 fps with Zigbee image transmission is achieved. The proposed system is able to show robust performance even when environment conditions are changed. Still some advancement can be included to the system to improve the overall performance. The size of the image buffer can be increased and the functionality of the system can be improved considerably by using various advanced processors with higher RAM Sizes.

6. Acknowledgements

We are grateful to KL University for their constant support and encouragement throughout this project. We

acknowledge the generous inputs and suggestions from the Embedded Systems and Sensor Networks (ESSN) research group of KL University.

7. References

1. Baghyasree T, Janakiraman K, Parkavi D. Efficient remote video monitoring for industrial environment based on LPC2148 and GSM. *Indian Journal of science and Technology*. 2014 Sep; 7(9):1333–41.
2. Krithika N, Seethalakshmi R. Safety scheme for mining industry using zigbee module. *Indian Journal of Science and Technology*. 2014 Aug; 7(8):1222–7.
3. Nourbakhsh I, Andre D, Tomasi C, Genesereth M. Mobile robot obstacle avoidance via depth from focus. *Robotics and Autonomous Systems*. 1997 Nov 20; 22(2):151–8. Crossref.
4. RGB color model [Internet]. 2017 [cited 2017 Feb 28]. Available from: Crossref.
5. Sugandi B, Kim HS, Tan JK, Ishikawa S. Real time object tracking and identification using camera. *The 23rd International Technical conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*; 2008. p. 1585–8.
6. Pekhteryev G, Sahinoglu Z, Orlik P, Bhatti G. Image transmission over IEEE 802.15.4 and zigbee networks. *Institute of Electrical and Electronics Engineers (IEEE) International Symposium on Circuits and Systems*. 2005 May 23–26; 4:3539–42.
7. Omnivision. OV6620 single chip CMOS CIF color digital camera OV6120 single-chip CMOS CIF B and W digital camera [Internet]. 1999 [cited 1999 Feb 6]. Available from: Crossref.
8. C3088 1/4 color camera module with digital output datasheet [Internet]. 2010 [cited 2010 Jul 28]. Available from: Crossref.
9. Wireless connectivity kit [Internet]. 2016 [cited 2016 Nov 30]. Available from: Crossref.
10. Microcontrollershop. XBee-PRO ZB S2B zigbee module, 63 mW, RP-SMA connector [Internet]. 2011 [cited 2011 Oct 17]. Available from: Crossref.