

RESEARCH AND DEVELOPMENT OF INTERNET OF THINGS PART I: ARDUINO AND WORLD WIDE WEB

Ing. Román Gerardo
García García
www.raptors.net/esthelfania
roman.garcia@raptors.net

M. C. Luis Manuel Martínez
Ramírez
www.raptors.net/luis
luis.martinez@raptors.net

Ing. Brenda Esthelfania
Romero Díaz
www.raptors.net/esthelfania
esthelfania.romero@raptors.net

M. C. Omar Lucio Cabrera
Jiménez
www.raptors.net/omar
omar.cabrera@raptors.net

Dr. René Mac Kinney
Romero
www.raptors.net/mac
rene@xanum.uam.mx

Ing. Jazmin Villegas
Soriano
www.raptors.net/jazmin
jazmin.villegas@raptors.net

Ing. Christopher Arturo
Pacheco Pérez
www.raptors.net/vermox
christopher.pacheco@raptors.net

Universidad Tecnológica de Nezahualcóyotl

Circuito Universidad Tecnológica sin número, Colonia Benito Juárez, Nezahualcóyotl, Estado de México, C. P. 57000

Universidad Autónoma Metropolitana Iztapalapa

San Rafael Atlixco núm 186, Colonia Vicentina, Iztapalapa, Ciudad de México, C. P. 09340

Instituto Tecnológico de Iztapalapa

Avenida Telecomunicaciones sin número, Col. Chinampac, Iztapalapa, Ciudad de México, C. P. 09208

ABSTRACT

It describes the first lesson as an essential and a fundamental directive to build a Device which can be operated from World Wide Web, showing how to develop and assemble its systems based on the Technology of Control, Logical, Mechanical, Electrical and Electronic (LMEEC). The device built here consists in a simple led that can be turned on and turned off from a web site in Real-Time to the owner of the led, for the Internet of Things Technology. Also how this can be used on education to teach how to integrate a group of students to hard heavy duty when they are learning any Enginner University or Institute Career.

INTRODUCTION

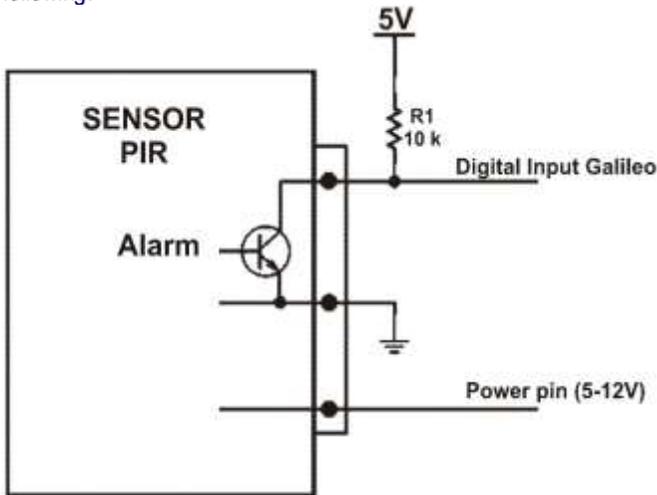
Communications World has been changing dramatically with the ocurrence of the world wide web in 1993 finally a good part of the population could access a global data network that for a long time was the exclusive use of Universities and of the Department of Defense of the United States of America [1]. This approach to this network of networks known as the Internet, brought many advantages, being one of the most important to shorten distances, allowed network users to access a large amount of information and also place the information within reach of the rest of users and this became a tool for everyday use in the office and at home. Internet enabled tasks that years ago would take days or hours in a few minutes giving rise to new ideas [2]. The possibility of remote control and maintenance operations was seen using the internet and this will be the focus of this project, which takes as a starting point the possibility and scope of using the internet as an easily accessible means to allow control and the maintenance of electronic devices [3]. With a simple browser it has a complete and real-time view of the state of an industrial plant, a farm or the security system of a home, and this can be controlled from anywhere in the world that has an internet connection available [4]. This is how the new concept called Internet of Things (IoT) was born, as we are entering an era in which the internet has the potential to dramatically improve the lives of everyone on our planet [5]. From curing diseases, to understanding climate change, improving the way of doing business, or making every day more enjoyable. As more things, people, and data connect, the power of the internet (which is essentially a network of networks) grows exponentially.

To fully exploit this we can take advantage of the potential of the embedded systems that are available at each comment of our life which is a computer system designed to perform one or a few functions dedicated in real time. The car, the elevator, the audio equipment, the airplane are controlled by computers that normally do not have a screen, a keyboard or hard disk, and do not respond to what we commonly call a personal computer. Embedded systems despite not being named are in many parts, in fact, it is difficult to find any device whose operation is not based on any embedded system, from vehicles to cell phones and even in some common appliances such as refrigerators and microwave ovens. One of the great technological advances in the last 2 years is the creation of the intel galileo board. This works under the Intel architecture compatible with arduino being a complete hardware and software solution that helps to explore the space of the internet of things. And to create innovative projects. In order to program the intel galileo board, it is necessary to make the most of the microsd card to which the linux operating system can be installed to optimize the resources of said board (being an embedded system), thus working with the programming language of our pleasure but always bearing in mind that what is needed are the following attributes of quality: reliability, speed, availability and security since, as explained above, its operation is in real time.

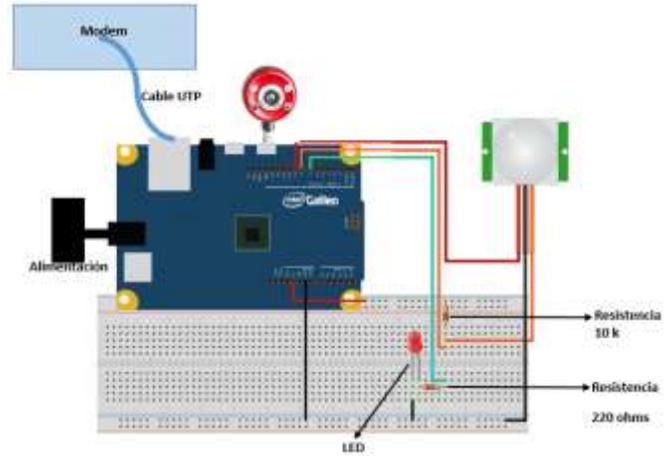
I. In this section we will see how to make the prototype application of control and monitoring using the intel galileo board applied to home automation which is an alarm system involving a pir sensor connected to the intel galileo board in addition to a webcam, when the sensor Detect a strange person. The webcam will take a picture each time at home which will be sent by email to the owner's email address. This photo taking will be done repetitively every 7 seconds while motion is detected. We need: **A.** intel galileo. **B.** 5v / 2a feeder (included in box). **C.** Sensor movement pir (fresnel lens). **D.** Ethernet Cable. **E.** micro sd memory, (minimum 8 gb). **F.** Resistors: 10k x1, 220 ohms x1. **G.** Led x1. **H.** Webcam 3.6 megapixel minimum. We have an Arduino Galileo Intel Board: **USB Client port to connect to the computer for programming the Galileo with Arduino.** **Port 2.0 USB Host serves as interface support with USB devices in this case here the webcam is connected.** **Ethernet port where the Galileo connects up to any speed of 10/100 Mb / s LAN.** **To connect to the internet and manage to send email.** **Power In This is a 2.1mm jack for a regulated 5V clean power supply.** **MSD card Galileo supports up to 32 GB microSD cards. In this case it will be at least 8 GB where the LINUX Yocto will be installed.** **Reset button Pressing this button starts the LINUX installed on the Intel Galileo Board.** **Here is used to store the main sketch which will start when you turn on the Intel Board.** **Galileo** is a microcontroller based on the iquark soc x1000 application processor. It is the first board based on Intel architecture and designed to be compatible with arduino. The digital inputs from 0 to 13 (and the pins aref and gnd), the analog inputs from 0 to 5, the power input.

Galileo is also compatible with the arduino development environment (ide), which facilitates its usability. In addition to that compatibility with arduino, galileo has a series of ports and features that are standard in the pc industry and that take their capabilities beyond the ecosystem of arduino sites. Then, we have: **A.** The intel galileo board can be powered by the usb client port but intel does not recommend it since the board can be burned, instead it is recommended that the power socket be through the power in with a power supply of 5 v cd regulated. **B.** It is recommended that the contracted internet be reliable and the modem available is in good condition because otherwise the device functionality can not be guaranteed, being the responsibility of the user if a failure occurs. **C.** It must be disconnected as long as it is not needed. **D.** When connecting to the outlet the pir sensor requires a preparation time to start operating properly (from 10 s to 30 s), so the absence of people is recommended. **Passive infra red.** **A.** Measuring range $6m > x <= 10m$. **B.** Minimum calibration time. **C.** It has only three terminals: two are used for power and the remaining is the motion detection output. Operation. It contains special filters called fresnel lenses that focus the infrared signals on the sensor element. When the infrared signals of the environment where the sensor is changing rapidly, the amplifier activates the output to indicate movement, this output remains active a few seconds allowing the microcontroller if there was movement. **Webcam.** **A.** 360 ° rotating head. **B.** Compatible with any USB 2.0 port itself through which it is powered. **C.** 60 fps image sensor (frames per second). **D.** Maximum resolution 1024x680 (vga). **E.** 4 night vision leds. **F.** Compatible with linux with driver v4l2.

II. We will connect the pir sensor on the pins of the galileo. The distribution of the sensor pins may vary depending on the model. In our case, the output pin is in open collector mode so we will have to put a resistance of 10k. The circuit based for this purpose is the following:



Each time a PIR sensor is powered, using Edison's principle, as ramptors team showed at Part I of this Research Project, there is a small calibration time that varies in each sensor. This is why we connect the sensor power pin to a Galileo digital output to control this calibration. We will also put a led with a resistance that will turn on every time the PIR sensor detects movement. Each time a py sensor is powered by the direct current; there is a small calibration time that varies at each sensor. This is why we will connect the sensor power pin to a galileo digital output to control that calibration. We will also put a led with a resistance that will turn on every time the pir sensor detects movement. Here we have an image of how our circuit would be armed.



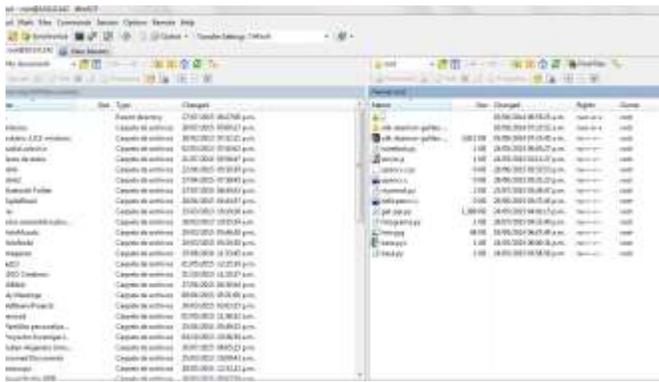
After the circuit has been assembled, using Python Language:

```
#program to get images
import numpy as np
import cv2
import sys
import cv
#constructor de videocapture para capturar imagen
Cap =cv2.VideoCapture(-1)
Cap.set(cv.CV_CAP_PROP_FRAME_WIDTH,640); #ancho de foto
Cap.set(cv.CV_CAP_PROP_FRAME_HEIGHT,480); #alto de foto
#función para saber si se ha abierto bien la cámara
if not cap.isOpened(): #si no se puede conectar a la cámara
o no se localiza
    print "la cámara no se encuentra"
    sys.exit(-1)
else:
    print "la cámara funciona"
Ret,frame = cap.read() #se crea un objeto donde se guarda la
imagen
cv2.imwrite('foto.jpg',frame) #se genera la foto
#para cerrar la cámara o la captura
Cap.release() #se termina el proceso de captura
```

In this case it was pasted in notebook on the pc and saved as [fotograma.py](#), in order to execute this code from the galileo said script had to be saved in a folder of the linux system integrated in this case in **/home/root**. To access the interior of these linux folders we used a program called winscp. It is by means of sftp its acronym means ssh file transfer protocol, it is completely different from the ftp protocol (file transfer protocol). Sftp was built from scratch and adds the characteristic of ftp to ssh. Only use a communication channel, send and receive messages in binary (and not in text format as ftp does). When it is opened, the following window appears:



III. Where we enter the ip assigned to the intel galileo board and the password. Prior to this, galileo must be connected via ethernet to a modem with internet so that it works and we can interact with it through another computer connected in the same network whether in a wired or wireless form. We click on login and we can navigate in the folders of linux of Galileo. In which on the left side are the folders of our personal computer and on the right is linux of our galileo transferring of the python script or other files to the galileo.



Running the website script. It was accessed through ssh through the putty program:



A. IP Address is entered, the ssh option is chosen and give it in load and ready. B. Enter the root password to start working on the linux console. C. To execute the python script for the camera we moved to the folder.

```
login as: root
root@quark00f424:~# cd /home/root
root@quark00f424:~# ls
base.py  helloopencv.c  tweetbot.py
base.pyc  myemail.py  xdk-daemon-galileo-precompiled-0.0.31
foto.jpg  opencv.c  xdk-daemon-galileo-precompiled-0.0.31.tar
fotograma.py  opencv.ccp
get-pip.py  server.js
root@quark00f424:~#
root@quark00f424:~#python fotograma.py
```

A first difficulty was found when sending an error message and this was due to the lack of the python cv2 library, so a research process was started to solve this problem and the solution was the following: It was necessary to install the opencv library, which is a library for capturing video and images with cameras originally developed by intel. In this case it was needed for the calibration of the camera. This has been achieved, by programming in optimized c and c++ code, also taking advantage of the capabilities provided by multi-core processors. The allocated space of 1gb originally for the image of the installed linux was not enough to support this library so it was necessary to expand the partition to 6 gb to make this possible. When the partition was expanded and the library was installed, the code was rerun and without any problem I took the photograph. The second difficulty was when the image was reviewed and it was dark. We started to investigate similar cases in the network and the solutions that appeared were not enough until it was found that a controller called v412 already existed, which had properties of brightness, contrast and saturation. Some screenshots are shown where they are appreciated. The properties of this controller that influenced the generated picture. Contrast of 64, saturation 64 and brightness 0;

So they started moving the values until the perfect fit was found as we see in the following screenshot. Contrast of 100, saturation 100 and brightness 50. The test photo was in a file. Once the photograph was taken successfully, we proceeded to program the code to send the photo by email.

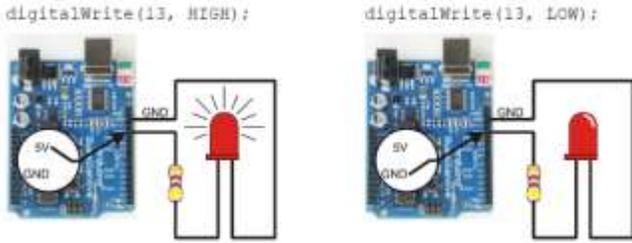
```
# importamos librerias
Import smtplib # para lograr conexión hacia el correo electrónico
Import mimetypes # para los tipos de objetos que contienen texto
Import os # para lectura html
# Importamos los módulos necesarios
From email.mime.multipart import mimemultipart # para el mensaje
From email.mime.image import mimeimage # para la imagen
From email.encoders import encode_base64 # para tipo de codificación ( símbolos)
From email.mime.text import mimetext # para almacenar temporalmente los textos de los correos
# creamos objeto multipart, quien sera el recipiente que enviaremos
Msg = mimemultipart() # creamos objeto de este tipo para el mensaje
Msg['from'] = "rubenalejandrol989@gmail.com" # para quien es el correo
Msg['to'] = "rubenalejandrol989@gmail.com" # desde donde se envia
Msg['subject'] = "¡Alerta! En tu hogar" # asunto del mensaje
Html = "" # código html básico solo para darle formato al texto en la bandeja de entrada del usuario
<html>
<head></head>
<body bgcolor="red">
<center><b><font color="red" size="5" face="engravers mt">alguien ha entrado a tu
habitacion aqui se te envian las fotos:</font></b></center>
</body>
</html>
"""
Part2 = mimetext(html, 'html') # asignamos el mensaje escrito en código html
# ajustamos imagen
File = open("foto.jpg", "rb") # para asignar nombre a la fotografia tomada
Attach_image = mimeimage(file.read()) # para cargar la fotografia
Attach_image.add_header('content-disposition', 'attachment; filename = "foto2.jpg") #
para ajustar la fotografia al correo electrónico
Msg.attach(part2) # para asignar al contenido del mensaje al correo
Msg.attach(attach_image) # para enviar la imagen al correo deseado
# autenticamos
Mailserver = smtplib.smtplib('smtp.gmail.com',587) # para hacer conexión con el servidor gmail
Mailserver.ehlo() # confirmación de conexión
Mailserver.starttls() # se ejecuta la conexión
Mailserver.ehlo() # nuevamente se confirma conexión
Mailserver.login("rubenalejandrol989@gmail.com", "*****") # se ingresa contraseña al correo
# enviamos
Mailserver.sendmail("rubenalejandrol989@gmail.com", "rubenalejandrol989@gmail.com",
msg.as_string()) # se manda el correo
# cerramos
Mailserver.close() # se cierra sesión
```

It was executed successfully, managing to send the test mail. Later a code was programmed in arduino which allows to coordinate the hardware function (galileo intel board, camera, pir sensor) with the software or the embedded system (configuration of the camera in the linux and python codes).

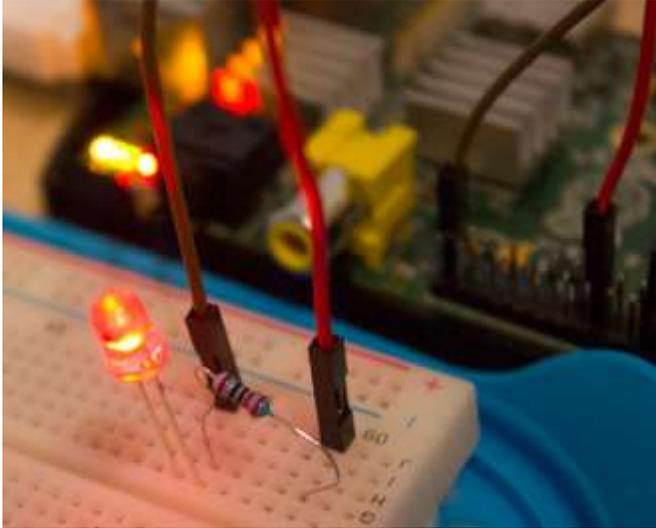
```
int led = 7; int pir = 8; int power_pir = 9;
void setup() {
serial.begin ( 9600 );
pinmode ( led, output );
pinmode ( pir, input );
pinmode ( power_pir, output );
digitalwrite ( power_pir, high );
serial.println ( "calibrating..." );
delay ( 10000 );
serial.println ( "pir ok" );
}
void loop() {
delay(2000);
if(digitalread(pir)==low)
{
digitalwrite(led,low);
system("v412-ctl --set-ctrl contrast=100");
system("v412-ctl --set-ctrl
saturation=100");
system("v412-ctl --set-ctrl
brightness=50");
system("python /home/root/fotograma.py");
system("python /home/root/myemail.py");
serial.println("mail sent");
}
else
{
digitalwrite(led,low);
}
}
}
```

It is upload the sketch and when it is verified, it says "I detect movement", "I take the photo and sent it" to the desired mail, we just press the reset button of the galileo and the sketch will be saved on the board so the next time it is want to run the prototype you just have to connect to the modem and then to the power outlet and you're done. The computer is no longer needed to put it into use since the same Galileo board executes it in embedded mode. Finally, it is ready to be used as many times as necessary. And that show that works OK.

The final test is here:



We have the picture of de the development:



CONCLUSIONS

The prototype is considered as a reliable system to guarantee the safety of the user's home, in addition to the fact that the home automation features were fulfilled because the security of the home is automated, it is integrated with an internal network (prototype-modem)) as an external network (internet). Due to its small volume is ideal for the room, the bedroom, a small office and unlike a traditional security camera this device is easily portable since it can be moved without any problem, as long as there is a modem available for Internet connection regardless of the service provider that provides the internet. Galileo is cheap and funny way to get into building any student. It can also be daunting to get started. Here has shown how to get a start-to-finish with a killer project: building a safe security system. In this paper, it has been introduce Galileo using the robot project as a guide to a variety of skills. The Internet of things (IoT) is becoming an increasingly growing topic in the workplace and outside of it. It is a concept that not only has the potential to impact how we live but also how we work. But what exactly is the Internet of things and what impact is it going to have on any people. There are a lot of complexities around the Internet of things but the basics were shown in this set of papers. Lots of technical and policy-related conversations are being had but many people are still just trying to grasp the foundation of what the heck these conversations are about. This is the concept of basically connecting any device with an on and off switch to the Internet (and/or to each other). This also applies to components of machines, for example a jet engine of an airplane or the drill of an oil rig. In 2020 there will be over 26 billion of connected devices some even estimate this number to be much higher, over 100 billion. The IoT is a giant network of connected things (which also includes people). The relationship will be between people-people, people-things, and things-things.

REFERENCES

- [1] Martin Ford. Rise of the Robots: Technology and the Threat of a Jobless Future. IEEE Robotics and Automation Society. (The Institute of Electrical and Electronics Engineers, Inc., 2014).
- [2] Leon Clay. Robotia: the perfection of a robot and the alarming consequences. IEEE Robotics and Automation Society. (The Institute of Electrical and Electronics Engineers, Inc., 2015).
- [3] Paul J. Springer. Military Robots and Drones: A Reference Handbook (Contemporary World Issues). IEEE Robotics and Automation Society. (The Institute of Electrical and Electronics Engineers, Inc., 2015).
- [4] Simon Monk. Programming Arduino Getting Started with Sketches. (New York: Addison Weasley, 2014).
- [5] Luis M. Martinez. Omar Cabrera. Veronica Pacheco. A Robot made of Computer Trash. IEEE Robotics and Automation Society. (The Institute of Electrical and Electronics Engineers, Inc., 2015).

CURRICULUM VITAE



Ing. Román García García. He studied at Universidad Nacional Autónoma de México – Facultad de Estudios Superiores Cuautitlán Campus Four, Mechanical Electrical Electronics Engineer. Now he is the last member of Nezahualcóyotl PHP Ramptors. He has worked at Universidad Tecnológica de Nezahualcóyotl as Systems Teacher since 2011.



M. C. Luis Manuel Martínez. He studied at Universidad Autónoma Metropolitana Mathematics Applied to Computer Science. Master in Mathematics at Centro de Investigación y Estudios Avanzados. Education Master at Universidad del Valle de México. He had worked at Universidad Tecnológica de Nezahualcóyotl as Researcher Teacher since 1993. IEEE member 4150968.



Ing. Brenda Esthefania Romero Díaz. She studies at Universidad Tecnológica de Nezahualcóyotl, Information Technology Engineer. Now, she is learning the fundamentals of Programming and Data Bases and Applications Development World Wide Web. Works at Artificial Nerds as QA Engineer. She is the IEEE Student Branch Nezahualcoyotl. IEEE member 97873418.



M. C. Omar Lucio Cabrera Jiménez. He studied at Universidad Autónoma Metropolitana Electronics Engineer. Master in Computer Science at Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas. He is the Systems Computer Career Dean at Universidad Autónoma Metropolitana Iztapalapa. He has worked at Universidad Autónoma Metropolitana as Researcher Teacher since 1996.



Dr. René Mac Kinney Romero. He studied at Universidad Autónoma Metropolitana Computer Systems Career. Master in Computer Science at Universidad Autónoma Metropolitana Iztapalapa. Doctorate in Computer Science at Universidad Autónoma Metropolitana Iztapalapa. He has worked at Universidad Autónoma Metropolitana as Researcher Teacher since 1990.



Ing. Jazmin Villegas Soriano. She studied at Universidad Tecnológica de Nezahualcóyotl, Information Technology Engineer. She is the Systems Computer Career Dean at Instituto Tecnológico de Iztapalapa. Now, she is researching Networks and Protocols. IEEE member 97937973.



Ing. Cristopher Arturo Pacheco Pérez. He studies at Instituto Tecnológico de Iztapalapa, Systems Engineer. Now, he is learning the fundamentals of Programming and Servers and Applications Development World Wide Web.