

Intel Galileo

Analía Faure¹, Mariano Lopez¹, Nicolás Ledesma¹, Leopoldo J. Rios²

¹Estudiantes, Licenciatura en Sistemas de Información, Universidad Nacional del Nordeste

Facultad de Ciencias Exactas y Naturales y Agrimensura, 9 de julio 1449 (3400) Corrientes
analia.f93@gmail.com, ledesma.nicolas.maximiliano@hotmail.com, m_villa@hotmail.com

²Profesor, Asignatura Redes de Datos, Licenciatura en Sistemas de Información, Universidad Nacional del Nordeste

Facultad de Ciencias Exactas y Naturales y Agrimensura, 9 de julio 1449 (3400) Corrientes
ljr@comunidad.unne.edu.ar

Abstract. El avance tecnológico permite la utilización de sensores los cuales captan magnitudes físicas, los mismos son incorporados a dispositivos digitales y el resultado puede ser transmitido por una red de datos para su manipulación. Se conoce como internet de las cosas a lo definido anteriormente, es decir a la interconexión de los objetos digitales con internet. En el presente trabajo utilizamos distintos sensores para obtener datos que pueden ser adquiridos por algún sistema informático y brindar mejor accesibilidad de los mismos al usuario.

Keywords: Internet de las cosas, placa microcontroladora

1 Introducción

El siguiente trabajo fue realizado como trabajo práctico de la materia “Redes de Datos” de la carrera Licenciatura en Sistemas de Información de la Universidad Nacional del Nordeste, durante el ciclo lectivo 2015. El mismo consiste en el desarrollo de un sistema web montado sobre un servidor web Apache, el cual debe interactuar con otro sistema de tiempo real desarrollado con Intel Galileo a través de una red de datos y poder almacenar información proveniente del sistema de tiempo real en una base de datos MySQL.

Es decir, lo que se logró es que no solo nos quedemos con las magnitudes físicas medidas por la placa Intel Galileo, sino también poder extraer esos datos aplicando conocimientos que hacen al área de redes y brindarle al usuario una manera accesible y útil de utilizarlos.

Hemos implementado un sistema web que recibe los datos de la placa, los almacena en una base de datos y a la vez lo imprime por pantalla, esto hace que estemos

hablando de un sistema dinámico. Consideramos relevante la funcionalidad que proporciona el sistema a la hora de analizar si el host está activo ya que esto hace puramente al ámbito de redes porque debe existir de por medio una red para captar ese dato.

El sistema desarrollado brinda accesibilidad hacia el usuario por que los datos que son visibles provienen de magnitudes físicas que fueron tomadas mediante los sensores conectados a la placa Intel, a la vez eso se vuelve muy útil para el usuario ya que desde su comodidad o desde donde tenga acceso a internet podrá visualizar a esas mediciones.

1.1 Internet of Things (IoT)

Generalmente al hablar de Internet de las cosas (también conocido como Internet de los Objetos), nos referimos a una red de interconexión diaria de objetos, los cuales están equipados con cierta inteligencia oblicua. IoT incrementa la ubicuidad de internet integrando cada objeto para su interacción vía sistemas embebidos, esto conlleva a una gran red distribuida de dispositivos comunicándose con humanos y también con otros dispositivos. Gracias al gran avance de la tecnología, IoT es una gran oportunidad para un gran número de aplicaciones que prometen mejorar la calidad de vida [1].

La visión de objeto inteligente es el corazón de IoT, lo cual es solo otro nombre para un sistema embebido que está conectado a Internet. Hay otras tecnologías que apuntan hacia la misma dirección – la tecnología RFID “Radio Frequency IDentification”. La tecnología RFID, es una extensión ubicua de una barra de códigos óptica que se encuentra en muchos productos hoy en día, en donde se incorpora al producto un componente electrónico inteligente de bajo costo “ID tag” que funciona como identificador y que puede ser decodificado a distancia. Agregando inteligencia dentro de este ID tag, el objeto enlazado se convierte en un objeto inteligente. La novedad de IoT no es la capacidad funcional de un objeto inteligente, al contrario, IoT se centra en otros aspectos como ser la forma de identificación de un objeto inteligente, la gestión automatizada y auto-organización de redes de objetos inteligentes, entre otros.

En la Fig.1 representa un ranking de las aplicaciones IoT realizado a fines del 2014, en donde vemos que las aplicaciones del tipo Smart Home “Hogar Inteligente” poseen 61 mil búsquedas en google por mes, 3 mil trecientos tweets por mes y 430 publicaciones por mes en LinkedIn, seguido por Wearables “Vestibles” con 33 mil búsquedas en google, 2 mil tweets y 320 publicaciones en linkedin.

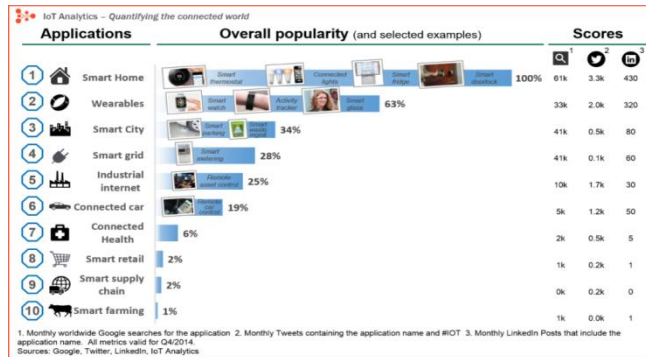


Fig. 1. Ranking de aplicaciones IoT [2]

- Smart Home: Este tipo de aplicaciones se concentra en el confort de los habitantes del hogar, en donde los mismos puedan programar o controlar dispositivos de la casa tales como televisores, heladeras, microondas, aires acondicionados, lámparas, termómetros, entre otros. Las comunicaciones no son solo Usuario → Dispositivo, en donde un ejemplo puede ser apagar algún electrodoméstico que nos olvidamos prendido en casa desde nuestro teléfono, sino también Dispositivo → Usuario, en donde un ejemplo podría ser una heladera avisando la falta de víveres.



Fig. 2. - Smart Home[3]

- Wearables: Este tipo de aplicaciones son diseñadas para objetos que puedan ser llevados por alguna persona, tales como ropa, reloj, auricular, etc. Actualmente es muy común encontrar dispositivos wearables que difundan nuestra posición GPS, o que informe nuestro ritmo cardiaco, entre otras soluciones que ofrece este tipo de aplicaciones.



Fig. 3. – Wearable[4]

1.2 Intel Galileo

Galileo es una placa micro controladora que posee características técnicas que hace que sea posible montar un sistema operativo Linux y poder tratarlo como la computadora que todos conocemos. Además, tiene una arquitectura Intel diseñada con hardware, software y pines compatibles con la placa de Arduino, especial para el uso de fabricantes, estudiantes, educadores y entusiastas del arte de crear productos electrónicos.

Intel Galileo está basada en Intel Quark SoC procesador 32-bits y posee varios puertos de E/S de PC estándares y funciones para expandir el uso nativo y las capacidades más allá del ecosistema de cubiertas Arduino. También incluye una ranura mini PCI Express de tamaño completo, puerto Ethernet de 100 Mb, ranura Micro-SD, cabezal 3.3V USB TTL UART de 6 clavijas, puerto de host USB, puerto de cliente USB y NOR Flash de 8 MB. Caché L1 de 16 KB. 512 KB de SRAM integrada en la matriz. Fácil de programar: subproceso único, un núcleo, velocidad constante. Compatible con el estado de suspensión de CPU y con ACPI. Reloj de tiempo real (RTC) integrado con batería "tipo botón" de 3V opcional para operar entre los ciclos de encendido. Velocidad de reloj de 400 MHz.

La programación se realiza mediante la herramienta IDE de Arduino que nos permite escribir programas a alto nivel "c/c++" y cuenta con un conjunto de librerías estandarizadas para lograr la comunicación con los sensores. Finalmente, a través de comunicación serial, el IDE graba el código binario en el chip de la placa, el cual ya se encuentra listo para usar.

Este microcontrolador nos permite medir la magnitud que se desea controlar mediante sensores que capturan dichas magnitudes del mundo real y las transforman en cantidades eléctricas equivalentes (EJ: 100km/h = 10v). También nos permite, mediante el IDE, comparar la medición realizada con una cantidad de referencia o salida deseada y actuar sobre el sistema según el resultado anterior.

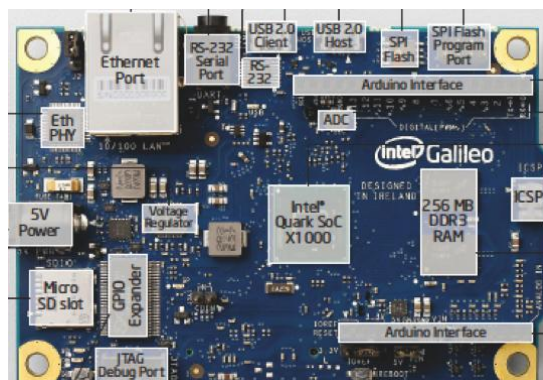


Fig. 4. - Especificaciones de Intel Galileo[5]

Diferencia Intel Galileo - Arduino

La diferencia entre la Galileo y una placa Arduino normal la marca el hecho de poder combinar la estructura de hardware y software del Arduino con el sistema operativo Linux. Gracias a esto, podemos controlar hardware como sensores o motores con otros lenguajes de programación como Python o Node.js y, además también, es posible ejecutar comandos durante la ejecución del programa. Esta tarea es de mucha utilidad desde el punto de vista del programador al evitarle que tenga que programar funciones o procedimientos que realicen lo mismo y dejar que el sistema operativo se haga cargo mediante una llamada al sistema. Aparte, tendrá un directorio como para ir almacenando información de control o de cualquier índole que podrá verse desde un acceso remoto o con un sistema que sea capaz de leer los datos y organizarlos de manera conveniente.

Sensores, Módulos y Shields

Sensores: son los que nos permiten medir las magnitudes de mundo real y las transforma en señales eléctricas equivalentes. Son los componentes que hacen que los proyectos de IoT se hagan realidad. Dentro de esta la gama podemos encontrar sensores de movimiento, temperatura, humedad, gas, proximidad, etc.

Cada uno de ellos, viene acompañado de un documento “datasheet” que resume el funcionamiento y otras características como ser:

- Datos del fabricante.
- Esquema de conexiones.
- Tensión de alimentación.
- Condiciones de operación recomendadas.
- Tabla de especificaciones.
- Normas de seguridad y uso.

Módulos: Son componentes electrónicos que permiten en conectar la placa con otros dispositivos mediante algún medio. Entre ellos, podemos encontrar Ethernet, bluetooth, GPRS, WiFi, etc.

Shields: Cuando hablamos de shield estamos haciendo referencia a algún módulo fabricado de forma acoplable a la placa de manera tal de dejar libre la mayor cantidad de pines posibles.

2 Desarrollo

Nuestro proyecto consiste en:

- Una aplicación web desarrollada en PHP que permite al usuario consumir los servicios ofrecidos por la placa, para ello PHP posee funciones específicas para efectuar peticiones http y manipular objetos del tipo json.
- Un Web Service (WS) montado en la placa que fue desarrollado en el lenguaje C el cual puede responder a distintas peticiones, que se muestran en la tabla 1.

Tabla 1. –Comportamiento del Web Service

Método HTTP	Petición	Descripción	Entrada JSON	Salida JSON
GET	Temperatura	Obtiene la temperatura ambiental actual captada por el sensor.		{temperatura:valor}
GET	Intensidad luminosa	Obtiene la intensidad luminosa actual captada por el sensor		{luz:valor}
POST	Ping	Realiza ping a algún host indicado y devuelve el resultado del mismo	{ip_host:valor}	{status:valor,mensaje:valor,enviados:valor,recibidos:valor,perdidos:valor}
POST	LED	Prende o apaga el LED que se encuentra en la placa, con la posibilidad de hacer PWM “modulación por amplitud de pulsos”.	{encender:valor,pwm:valor}	{mensaje:valor}

Estas repuestas son del tipo JSON “JavaScript ObjectNotation” y se obtienen mediante el número IP: puerto/petición correspondiente”.

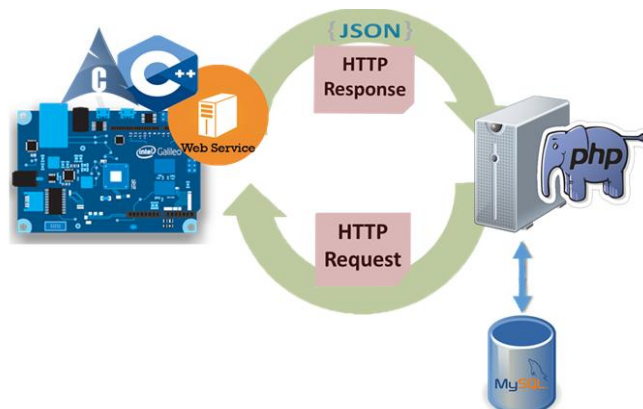


Fig. 5. – Esquema previsto - autoría propia.

Para establecer comunicación entre el servidor web Apache y la placa Intel Galileo es necesario conocer la dirección IP de la placa debido que al conectarla a la red la misma recibirá una dirección IP en caso de haber algún servidor DHCP en la red, caso contrario la misma no tendrá dirección IP o mantendrá alguna dirección estática establecida anteriormente.

Wireshark es un analizador de paquetes de red. Tratará de capturar los paquetes de red y trata de mostrar los datos de paquete lo más detallado posible.

Se podría pensar en un analizador de paquetes de red como instrumento de medida utilizado para examinar lo que está pasando en el interior de un cable de red.

Wireshark es quizás uno de los mejores analizadores de paquetes de código abierto disponibles en la actualidad [6].

Como no se conocía a priori la dirección IP de la placa, utilizamos el software WIRESHARK para obtenerla.

Web Service en Intel Galileo.

```
function traerTemperatura() {
    $url = 'http://169.254.212.42:8080/';
    $valor = array(
        'query' => 'temp');
    $get = http_build_query($valor);
    $conex = curl_init();
    curl_setopt($conex, CURLOPT_URL, $url.$get);
    curl_setopt($conex, CURLOPT_HTTPGET, 1);
    curl_setopt($conex, CURLOPT_HTTPHEADER, array(
        'Content-Type: application/json',
        'Accept: application/json')
    );
    curl_setopt($conex, CURLOPT_RETURNTRANSFER, true);
}
```

```

$result = curl_exec($conex);
curl_close($conex);
$obj = json_decode($result,true);
$this->load->model('basegalileo');
$date= date ('Y-m-d H:i:s');
$this->basegalileo-
>cargarTemperatura($obj['Celsius'],$date);
return ($obj['Celsius']);
}
public function getTemperatura(){
echo 'LA TEMPERATURA ACTUAL ES : <u>'.$this-
>traerTemperatura().' </u>';
}

```

Web Client en PHP.

“\$conex = curl_init()”, con esa línea se establece una conexión tcp “previamente configurada”.

Las respuestas del WS son almacenadas en una base de datos MySQL, a su vez las mismas son visualizadas por usuario, informando un historial de temperaturas (temperatura actual y temperaturas anteriores junto a su fecha y hora) que ha detectado la placa, como así también los pings que realizó a los distintos hosts.

```

function recuperarPing(){
    $ip = '169.254.212.41';
    $date= date ('Y-m-d H:i:s');
    $url = 'http://169.254.212.42:8080/';
    $campo = array(
        'ping' => $ip,);
    $post = http_build_query($campo);
    $conexion = curl_init();
    curl_setopt($conexion,CURLOPT_URL, $url);
    curl_setopt($conexion,CURLOPT_POST, count($campo));
    curl_setopt($conexion,CURLOPT_POSTFIELDS, $post);
    curl_setopt($conexion, CURLOPT_HTTPHEADER, array(
        'Content-Type: application/json',
        'Accept: application/json',
        'Content-Length: ' . strlen($post)
    ));
    curl_setopt($conexion, CURLOPT_RETURNTRANSFER,
true);
    $result = curl_exec($conexion);
    curl_close($conexion);
    $obj = json_decode($result,true);
    $a = ($obj['transmitidos']);
}

```



```
$b = ($obj['recibidos']);  
$c = ($obj['perdidos']);  
$this->basegalileo->cargarPing($a, $b, $c, $ip, $date);  
return ($obj['perdidos']);  
}
```

3 Conclusión

Con el desarrollo de la aplicación se logró profundizar los conocimientos de conceptos de Open Hardware, protocolo http y servicios REST, lo cual nos permitió hacer una interconexión entre un sistema de tiempo real encargado de recuperar las muestras de los sensores e informarlos a través de servicios web del tipo REST, y un sistema web el cual logra consumir los servicios brindados por la placa a través del protocolo http.

Queda demostrado que gracias al Open Hardware y Open Software hoy en día es muy fácil incluir manipulación de sensores de magnitudes físicas en nuestros desarrollos, ya que los mismos poseen sus propias hojas de especificaciones las cuales se encuentran estandarizadas y libres, lo cual dio pie a un amplio desarrollo de librerías para el tratamiento de los mismos, lo que se resume a que con pocas líneas de código podamos recuperar el valor numérico de alguna magnitud física. La utilización del protocolo http fue clave para poder establecer la comunicación entre ambos sistemas, el lenguaje de programación PHP posee en forma nativa librerías para poder realizar conexiones del tipo TCP/IP los cuales fueron requeridos para poder hacer uso de los servicios REST a través de http.

4 Bibliografía

- [1] L. Yang, L. Wang, A. Vinel y F. Xia, «Internet of Things,» International Journal Of Communication Systems, n° 25, pp. 1101-1102, 2012.
- [2] Fuente: <http://iot-analytics.com/10-internet-of-things-applications/>
- [3] Fuente: <http://smarthome.city/2015/07/15/what-can-an-australian-smart-home-do-this-2015/?v=5b61a1b298a0>
- [4] Fuente: <http://raconteur.net/wearable-technology>
- [5] Fuente: <http://www.neoteo.com/intel-galileo-ordenador-open-source-de-70-dolares>
- [6] Fuente: <https://www.wireshark.org/download/docs/user-guide-a4.pdf>