

## PROGRAMACIÓN EN JAVA DE SISTEMAS EMBEBIDOS HACIENDO USO DE APPLETS Y SERVLETS PARA LA MEDICIÓN Y CONTROL DE VARIABLES DENTRO DE UN INVERNADERO

Omar Fabián Rivera Cenicerros, Luis Alberto Reyes Ibarra, Christian Ríos Chavarría

Universidad Politécnica de Durango  
Ingeniería en Telemática

Carretera Durango-México Km. 9.5, Rancho Valle Hermoso, Durango, Dgo.  
omar.rivera@unipolidgo.edu.mx, christian.rios@unipolidgo.edu.mx

### RESUMEN

En el presente trabajo se muestra el desarrollo de aplicaciones útiles para la medición, y control de procesos, empleando las tecnologías de la información y desarrollando sistemas embebidos bajo un entorno de desarrollo integrado libre creando componentes de aplicaciones que se ejecuta en el contexto de HTML, con interfaces más gráficas y amigables libres de hipervínculos dentro de un navegador web que soporte la maquina virtual Java, donde se corran los procesos de esta plataforma. Lo anterior permite que las aplicaciones sean independientes del sistema operativo o navegador, pudiendo realizar acciones libremente con los sistemas desarrollados en cualquier dispositivo fijo ordenador como laptops, desktops, PDA, móviles, tabletas u otros dispositivos. Teniendo aplicaciones dinámicas embebidas con programas prácticos e interactivos bajo el esquema de diferentes tecnologías maduras y estables basadas en Java, capaces de mantener un monitoreo continuo de variables monitoreadas a través de sistemas microcontrolados, tarjetas de adquisición de datos o módulos embebidos dentro de una aplicación que se comunica asincrónicamente con el servidor cargando un motor que ejecuta un programa en el servidor dentro de una página en HTML.

*Palabras clave: Java EE, Servlet, Applet, HTML, embebido, monitoreo, control.*

### Abstract.

This paper shows the development of useful applications for measurement and process control, using information technology (IT) and developing embedded systems under a free integrated development environment, creating application components running in the HTML context, with graphical interfaces free of hyperlinks within a web browser that supports Java, having installed the Java virtual machine. This allows applications to be independent of operating system or browser, we can perform

actions freely with systems developed in any computer fixed device such as laptops, desktops, PDAs, mobile phones, tablets or other devices. With embedded dynamic applications and interactive with practical programs under the scheme of different Frameworks based under Java, able to maintain a continuous monitoring of sensors, monitored through microcontrolled systems, data acquisition boards or modules embedded within a application that communicates with the server asynchronously loading a motor that runs a program on the server within a static HTML page.

### 1. INTRODUCCIÓN

Con el desarrollo acelerado de las tecnologías de la comunicación, y con el auge de Internet, la arquitectura *cliente-servidor* ha adquirido una mayor relevancia, ya que la misma es el principio básico de funcionamiento de la *World Wide Web* [1]. Los sistemas telemáticos y los avances en el telecontrol y teleproceso, se hace cada vez más necesario ofrecer alternativas tecnológicas que interactúen en sistemas embebidos donde se conjunte tanto el hardware como el software, trabajando en conjuntos los sistemas electrónicos, equipos de instrumentación, etcétera, junto a los sistemas computacionales, software embebido, redes de comunicación y bases de datos. En este trabajo se muestra el desarrollo de un sistema embebido, empleando diferentes tecnologías y Frameworks que ofrece Java, todo programado mediante un entorno de desarrollo integrado y de distribución libre.

Para el desarrollo de aplicaciones como se proponen en este trabajo, la herramienta de desarrollo NetBeans, la cual provee recursos más que necesarios para poder desarrollar un software que embeba diferentes aplicaciones y tecnologías para poder llevar a cabo la creación de páginas web estáticas ejecutándose en un servidor, capaces de correr una aplicación dinámica, haciendo peticiones al servidor y script capaces de ejecutarse sobre la aplicación

cliente sin hacer uso de actualizaciones de pagina web ni hipervínculos que muestren otra página web estática. Para realizar lo anterior encontramos varias alternativas:

- Aplicaciones CGI
- Java Applet
- Aplicaciones AJAX
- Java Servlets

Desde esa primera concepción del servidor HTTP como mero servidor de ficheros HTML el concepto ha ido evolucionando en dos direcciones complementarias:

1. Añadir más inteligencia en el servidor
2. Añadir más inteligencia en el cliente.

De cara a estos apuntes, se tiene mucho más interés los caminos seguidos para añadir aun más inteligencia en el servidor. Con el paso del tiempo esta forma ha de trabajar con formularios ha madurado, permitiendo usar diferentes herramientas alternativas más dinámicas para procesar, verificar y procesar los datos. Las formas más extendidas y aun vigentes de añadir inteligencia a los clientes han sido Javascript y las Applets de Java. Los programadores y desarrolladores de sistemas embebidos se encuentran a menudo con que las plataformas de desarrollo HTML no cumplen más allá de los requerimientos básicos como medición y control ON-OFF, mediante un diseño donde la interfaces con las que se trabaja, se construyen solamente con tareas mediante hipervínculos y cuadros de texto. Con este proyecto se trabajó con el desarrollo una interfaz embebida inteligente, con la intención de aplicar el sistema dentro de un invernadero para fines de desarrollo de tecnologías que brinden apoyo a la horticultura protegida.

### 1.1 Horticultura protegida

La horticultura protegida se define como el sistema de producción que permite modificar el ambiente natural en el que se desarrollan los cultivos hortícolas, con el propósito de alcanzar un crecimiento óptimo y con ello, un alto rendimiento. Este sistema permite ofrecer productos de alta calidad, con mejores precios de venta y con mayores niveles de inocuidad.

La horticultura protegida contribuye a sustentar y fomentar el desarrollo agroindustrial, a generar divisas y empleo para el país y una vida más digna entre la gente del medio rural. La productividad del cultivo de tomate en cierto grado suele estar limitada por luz, temperatura, nutrición y abastecimiento de agua. La producción de cultivos en casas sombra ha mostrado gran auge en el Noroeste del País, su ventaja fundamental es la reducción de la temperatura. Estas estructuras funcionan como

barrera física para los insectos y mejoran el ambiente en zonas con alta irradiación, alta temperatura y baja humedad relativa [31].

## 2. DESARROLLO

En el presente trabajo se muestra el desarrollo de aplicaciones Java útiles para la medición, y control de procesos, empleando las tecnologías de la información y desarrollando sistemas embebidos bajo un entorno de desarrollo integrado libre creando componentes de aplicaciones que se ejecuta en el contexto de HTML, con interfaces más gráficas y amigables libres de hipervínculos, dentro de un navegador web que soporte la maquina virtual Java, donde se corren los procesos de esta plataforma. Con esto se permite que las aplicaciones sean más dinámicas, con interfaces sencillas capaces de correrse en servidores remotos que permitan la administración de clientes remotos específicos que no requieran la instalación de una aplicación en sus ordenadores al hacer uso de Applets de Java montados en una clase servlet que permita la portabilidad y seguridad de las aplicaciones Java.

Lo anterior permitió el uso y conjugación de técnicas avanzadas de programación en java que logrando la perfecta integración e interacción con los dispositivos electrónicos con los cuales se desarrollaron diversos prototipos embebidos mediante el uso de tarjetas Arduino UNO, dando un nuevo enfoque de diseño a técnicas y tecnologías con las cuales se ha hecho desarrollo de aplicaciones por años [32]. Tarjetas bajo una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

### 2.1 Objetivo

Desarrollar un software para sistemas embebidos que permita la conexión asíncrona de interfaces seriales RS-232 en un servidor Web, empleando lenguajes y sistemas de hardware libre, mediante el desarrollo de aplicaciones con técnicas avanzadas de programación web en Java que permitan crear interfaces robustas cliente-servidor.

### 2.2 Diseño

Para el diseño de la interfaz Applet se dividió el diseño del algoritmo en 3 áreas importantes:

1. Interfaz gráfica de configuración y comunicación bidireccional RS-232 [5]
2. Servlet
3. Threads o Hilos

Optando por usar el servidor Apache Tomcat 7.0.26 para correr la aplicación.

Siguiendo los mismos esquemas open-source del software, se trabajó con una interfaz Arduino UNO. Arduino como ya se mencionó, es una plataforma de hardware libre, basada en

una placa con un microcontrolador Atmega328 [32].

En la figura 1 se muestra el diagrama a bloques a implementar para el desarrollo del proyecto y la prueba de la aplicación.



Figura 1. Diagrama a bloques del sistema de Telecontrol

En el diagrama de bloques se puede ver la implementación de una red de área local y un servidor Web para realizar las pruebas correspondientes de control y medición del sistema embebido por medios telemáticos, al conectar varios host con su configuración de red correspondiente, con el objetivo de simular una red dentro de la nube de internet.

### 3. RESULTADOS

Se desarrolló el algoritmo y la aplicación de acuerdo con las especificaciones de requerimientos del sistema. El diseño arquitectónico debe permitir visualizar la interacción entre las entidades, cuya estructura está definida por el objetivo final del proyecto y las necesidades que se tienen para el control telemático de actuadores. Se creó una red de área local (LAN) para poder realizar las conexiones con el servidor DNS (Domain Name System), donde se debe instalar y configurar el servidor Tomcat para que la aplicación se pueda ejecutar en la dirección IP correspondiente al servidor. Por la parte electrónica, como ya se mencionó, se trabajó con una tarjeta de desarrollo Arduino UNO, la cual sirvió como interfaz embebida para adquisición de datos provenientes de 2 sensores (uno de temperatura y otro de nivel ultrasónico), además de llevar el control de los actuadores a través de una etapa de potencia construida a base de optoacopladores y Triacs para manipular una motobomba que suministra un cisterna y un calentador que se trabajan en un invernadero de experimentación de las propia universidad.

#### 3.1 Interfaz Applet

Un Applet es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web [2]. El Applet debe ejecutarse en un contenedor, que lo

proporciona un programa anfitrión, mediante un plug-in. A diferencia de un programa, un Applet no puede ejecutarse de manera independiente, ofrece información gráfica y a veces interactiva con el usuario. Un Applet normalmente lleva al cabo una función muy específica que carece de uso independiente. El Applet se utiliza principalmente para el trabajo de páginas web, ya que es un pequeño programa incrustado en una página HTML y representado por una pequeña pantalla gráfica dentro de ésta. Cuando un navegador carga una página web que contiene un Applet, este se descarga en el navegador web y comienza a ejecutarse. Esto permite crear programas que cualquier usuario puede ejecutar con tan solo cargar la página web en su navegador.

Sin embargo, el inconveniente principal de los Applet es que se descargan y se ejecutan en la máquina virtual del cliente, por lo tanto, aquellos ingenieros o estudiantes de ingeniería que estén trabajando con el desarrollo de sistemas embebidos, adquiriendo datos con tarjetas de adquisición, monitoreando, consultando o controlando variables como: *Temperatura, Humedad, Presión, Radiación Solar, Voltaje, Intensidad, Impedancia, etcétera*, hacer uso de Applet resulta de que no se pueda descargar un Applet que se encuentre conectado directamente a un servidor y este interactuando directamente con los puertos del ordenador, ya que el Applet al descargarse en la máquina cliente, pretenderá leer los puertos del mismo ordenador cliente. Sin embargo para resolver ese problema, se propuso hacer uso de un **servlet**, para que el Applet programado se ejecute en el mismo servidor y se comuniquen por lo tanto, con los mismos puertos de la máquina que provee el servicio.

#### 3.1.1 Diseño del Applet

Por lo general, un Applet que desarrolla la misma tarea una y otra vez, debería tener un hilo (thread) con un bucle while o do-while que desarrolle dicha tarea [2]. Un ejemplo típico es un Applet que realiza una animación temporizada, como un reproductor de películas o un juego. Sin embargo para la aplicación desarrollada, el uso de un hilo es orientado a la lectura constante de datos provenientes de la tarjeta de adquisición, haciendo uso del puerto de comunicaciones serial RS232, donde se realiza el muestreo del arribo de los datos. Para la apertura, configuración y cierre del puerto COM para entablar la comunicación serial, se empleó la librería Java *GiovynetDriver1.1*, la cual se debe de importar dentro de nuestro Applet, para permitir la comunicación con el puerto COM de la máquina servidora al configurarla con los parámetros correctos:

```
****Abrir puerto****/
Parameters parameters = new
Parameters();
parameters.setPort(jComboBox1.getSelected
Item().toString()); //Se selecciona el
puerto
parameters.setBaudRate(Baud._2400); //
System.out.println("Puertoabierto:"+jCom
boBox1.getSelectedItem().toString());
Com com = new Com(parameters);
****Envío de datos****/
String data = "A"; //Se envía la letra de
control A hacia el modulo Arduino
for (int i = 0; i <
data.toCharArray().length; i++) {
Thread.sleep(400);
com.sendSingleData(data.toCharArray()[i]
);
```

Para leer los datos se requiere abrir el puerto, el siguiente código hace la tarea:

```
public void AbrirPuerto() throws
Exception{
caracterIniciaLectura = 0; //Determinar
el caracter la lectura
if(formatoCadena!=null){
if(subCadena==null)
subCadena = formatoCadena;
caracterIniciaLectura =
formatoCadena.indexOf(subCadena);
}else{
formatoCadena = "";
bRegresaCadenaCompleta = true;}
try { //
if (getFormatoCadena().length()>0){
puerto = formatoCadena.length(); }
catch (Exception ex) {
throw ex;}}
```

Para finalizar el ciclo de vida Hilo se ejecuta el siguiente modulo:

```
public void CerrarPuerto() throws
Exception{
try {
puerto.close();
} catch (Exception ex){
throw ex;}}
```

### 3.2 Servlet

Los Applets y servlets son componentes diferenciados, pero que pueden ser coordinables entre sí [7]. De esa manera se ejecuta una aplicación que tenga un interfaz de lectura serial mediante un Applet que usen la librería Swing, y que además se conecte a un servlet. El servlet contendrá la lógica del dominio y también acceder a los puertos de la máquina servidora. Primeramente se debe de enviar y recibir datos a través del servlet, para poder entablar la comunicación entre el Applet y el servidor.

```
String input = puerto();
// Envío de datos al Servlet
URLConnection con =
getServletConnection();
OutputStream outstream =
con.getOutputStream();
ObjectOutputStream oos = new
ObjectOutputStream(outstream);
oos.writeObject(input);
oos.flush();
oos.close();
// Recepción de datos desde el Servlet
InputStream instr =
con.getInputStream();
ObjectInputStream inputFromServlet = new
ObjectInputStream(instr);
String Cadena = (String)
inputFromServlet.readObject();
inputFromServlet.close();
instr.close();
```

Para leer y escribir cadenas de caracteres al Applet desde el Servlet, esta implementación se ejemplifica con el siguiente código

```
InputStream in =
request.getInputStream();
ObjectInputStream inputFromApplet = new
ObjectInputStream(in);
String echo = (String)
inputFromApplet.readObject();
// Echo hacia el Applet
OutputStream outstr =
response.getOutputStream();
ObjectOutputStream oos = new
ObjectOutputStream(outstr);
oos.writeObject(echo);
oos.flush();
oos.close();
```

Solo queda llevar a cabo la visualización a través de una página web con el servidor Apache GlassFish v3.0.1 haciendo una llamada al "localhost" dando como resultado la interfaz gráfica de la figura 2 donde se muestra la interfaz de control y la interfaz de configuración de puerto.

The image displays two screenshots of the 'Config' window in the software. The left screenshot shows the 'Temperatura' field set to 27 °C, 'Distancia' set to 8 in, and 'Control de Calentador' set to 'ON'. The right screenshot shows the 'COM1' dropdown menu and the 'Conectar' button.

Figura 2. Interfaz Grafica de usuario creada mediante el uso de un Applet

### 3.3 Interfaz de desarrollo Arduino

```
void loop() {
int    sensorValue    =    analogRead(A0);
//activamos la lectura del sensor
ultrasonica
Serial.println(sensorValue,    DEC); //
enviamos la medición del sensor vía
serial
int    sensorValue    =    analogRead(A0);
//activamos la lectura del sensor
ultrasonica
```

```
Serial.println(sensorValue, DEC,); //
enviamos la medición del sensor vía
serial
if (Serial.available() > 0) { // inicia
prendido y apagado serial
char serIn=' ';
serIn = (Serial.read());
if (serIn == 65){ //prende bomba con "A"
digitalWrite(9,HIGH);}
if (serIn == 66){ // apaga bomba con "B"
digitalWrite(9,LOW);} }
if(serIn==67){ //prende calenton con "C"
digitalWrite(8,HIGH);}
if(serIn==68){ //apaga calenton con "D"
digitalWrite(8,LOW);} } } //termina
prendido apagado serial
```

El sistema es capaz de poder realizar la medición de la temperatura ambiente dentro de una sección del invernadero, instalando el sensor en una zona localizada en las proximidades de una de las esquinas, para las pruebas se utilizó un sensor básico de temperatura LM35CH de precisión en encapsulado metálico, el cual tiene una resolución de 10mv/°C. Además se empleó un sensor ultrasónico de alto desempeño LV-MaxSonar-EZ3, para la medición de llenado de la cisterna principal, lo que permite calcular el porcentaje de capacidad de agua presente dentro del cilindro. En la figura 4 se muestran tanto el sensor de temperatura localizado dentro en un panel dentro del invernadero en la zona anteriormente citada, tanto como la localización del sensor ultrasónico, instalado en el interior de la cisterna.



Figura 3. Sensores de Temperatura LM35CH, Ultrasónico EZ3.

El manejo de potencia, es decir la manipulación de altas corrientes, de hasta varios amperios, implica el tener consideraciones de seguridad eléctrica para los operarios y de protección para el sistema digital. Es deseable que la interconexión entre ambas etapas (la digital y la de potencia) se haga por un medio de un optoacoplador MOC3011 que permite aislar eléctricamente los dos sistemas. El triac de control seleccionado fue el BTA41A que soporta hasta 400vca y 40 amperes a plena carga, lo que nos hace que sea una opción

Figura 4. Etapa de potencia optoacoplada.



### 3.6 Diseño de la red

Para realizar la transmisión de los datos con el servidor, se diseñó una red LAN a través de un router Cisco 2621 y un switch Cisco 2960, en conjunto con un Access Point Teletronics compatible con el protocolo IEEE 802.11b para realizar la comunicación con dispositivos inalámbricos como Tablet, IPADs, Celulares, Laptops, etc.

La figura 6 muestra la conexión física de la red LAN, y la tabla presenta la tabla reducida de ruteo de la red diseñada.

Una vez que se realiza la red, se procede a comprobar la comunicación con el servidor, y la puesta en marcha de la página web que alberga el Applet, con la finalidad de comprobar el correcto funcionamiento de las llamadas al servlet a través de máquinas cliente.

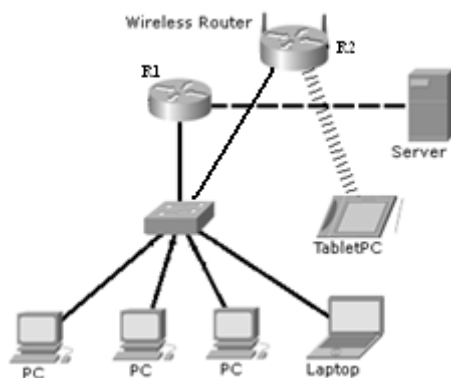


Figura 6. Conexión de la Red LAN

### 5. CONCLUSIONES

El software OpenSource se define por la licencia que lo acompaña, que garantiza a cualquier persona el derecho de usar, modificar y redistribuir el código libremente, y los desarrolladores que diseñan software para ser compartido, mejorado y distribuido libremente, pueden usar la marca. El desarrollo y uso de utilidades de código abierto fomentan el desarrollo de aplicaciones y herramientas pensando en los beneficios prácticos de compartir el código, permitiendo crear y mejorar la producción de las aplicaciones y sistemas. Lo anterior debido a que los programadores y desarrolladores de sistemas electrónicos o sistemas embebidos, pueden corregir, mejorar o adaptar las aplicaciones a las necesidades de cada estudio o diseño. EL proyecto aplica el desarrollo y uso de utilidades embebidas bajo un esquema de licencia GNU que garanticen el derecho de usar y modificar el sistema.

El uso de Applets para esta aplicación muestra la amplia gama de utilidades que aun se pueden aplicar en el desarrollo de nuevos sistemas, explotando y “sacándole jugo” para la creación de tecnologías embebidas, demostrando las capacidades poco explotadas de las aplicaciones Java que permiten la asimilación de distintas técnicas. Explotación de recursos de una manera poco convencional que demuestra que no por ser una herramienta vieja y en desuso, no se le pueda aun sacar ventajas y desarrollar de una manera inventiva, aplicaciones de gran impacto.

### 6. REFERENCIAS

- [1] García, J.; Rodríguez, I.; Imaz, A. *Aprenda Servlets De Java Como Si Estuviera En Primero*. Escuela Superior de Ingenieros. Campus Tecnológico de la Universidad de Navarra. San Sebastián, abril de 1999.
- [2] González, I.; Sánchez, A.; Hernández, D.; *Programación Orientada a Objetos, Java Threads*. Departamento de Informática y Automática. Universidad de Salamanca. Mayo de 2002.
- [3] Hunter, J.; Crawford W.; *Java Servlet Programming. Chapter 10: Applet-Servlet Communication*. Copyright 2001 O'Reilly & Associates. ISBN 1-56592-391-XE
- [4] Ozer, J.; Blemings, H.; *Practical Arduino*. Ed. Apress. 2009. ISBN: 978-1-4302-2478-5
- [5] *Programming Serial and Parallel Ports*. <http://java.sun.com/developer/Books/javaprogramming/cookbook/11.pdf>
- [7] Torrijos. R.; *Introducción a los Servlets*. [http://www.programacion.com/articulo/servlets\\_basico\\_108](http://www.programacion.com/articulo/servlets_basico_108)
- [8] *Escribiendo un Servlet simple*. <http://manuales.dgsca.unam.mx/webdina/escribiendo.htm>
- [9] *Servlet Central*. <http://www.servletcentral.com/>
- [10] *Java Servlet*. [http://es.wikipedia.org/wiki/Java\\_Servlet](http://es.wikipedia.org/wiki/Java_Servlet)
- [11] Sharathk60. *how to call applet from servlet class*. <http://www.daniweb.com/web-development/jsp/threads/281591/how-to-call-applet-from-servlet-class>. 2011.
- [12] *GNU General Public License*. [http://es.wikipedia.org/wiki/GNU\\_General\\_Public\\_License](http://es.wikipedia.org/wiki/GNU_General_Public_License)
- [13] Ferrari, A. *Avance del software libre en la administración argentina*. 29 de septiembre de 2003. <http://www.rebelion.org/hemeroteca/cibercensura/030929sl.htm>
- [14] Garrett, J. *Ajax: Un nuevo acercamiento a las aplicaciones web*. Maestros del Web. 2006 <http://www.maestrosdelweb.com/editorial/ajax/>

- [15] **Software libre.**  
[http://es.wikipedia.org/wiki/Software\\_libre](http://es.wikipedia.org/wiki/Software_libre)
- [16] **Open Source Initiative.**  
<http://www.opensource.org/licenses>
- [17] **NetBeans.**  
<http://es.wikipedia.org/wiki/NetBeans>
- [18] **Common Development and Distribution License.** <http://es.wikipedia.org/wiki/CDDL>
- [19] **AJAX.** <http://es.wikipedia.org/wiki/AJAX>
- [20] **Applet.** <http://es.wikipedia.org/wiki/Applet>
- [21] **Applet Java.**  
[http://es.wikipedia.org/wiki/Java\\_applet](http://es.wikipedia.org/wiki/Java_applet)
- [22] **Concepto de página Web dinámica.**  
<http://manuales.dgsca.unam.mx/webdina/concepto.htm>
- [23] **Página web.**  
[http://es.wikipedia.org/wiki/Página\\_web](http://es.wikipedia.org/wiki/Página_web)
- [24] **HTML5.**  
<http://es.wikipedia.org/wiki/HTML5>
- [25] **Web Applications using Spring.**  
<http://netbeans.org/kb/docs/web/quickstart-webapps-spring.html>
- [26] **AJAX Java.**  
<http://netbeans.org/kb/docs/web/ajax-quickstart.html>
- [27] **Aplicación para rastrear un gps, AJAX, PHP & Mysql.**  
<http://www.forosdelweb.com/f127/aplicacion-para-rastrear-gps-ajax-php-mysql-627655/>
- [28] **PHP, puerto rs232 y PIC.**  
<http://www.todopic.com.ar/foros/index.php?topicc=15846.0>
- [29] **PHP Serial Port API.**  
[http://www.easyvtools.com/phpserial/php\\_ser\\_SMS.html](http://www.easyvtools.com/phpserial/php_ser_SMS.html)
- [30] **Embedded Targets Development Guide.**  
Arduino/Embedded\_Targets\_Development\_Guide.htm
- [31] Castellanos, J. **Manual de Producción de tomate en invernadero.** Ed. Intagri, S. C. 2009  
ISBN:978-607-95302-0-4