

Plataforma de conexión de Redes Eléctricas Inteligentes a Internet de las Cosas

Canziani, Mónica; Gomez, Rodrigo; Lupi, Daniel; Nassipián, Verónica; Slawiski Javier; Turconi, Diego; Zaradnik, Ignacio

Laboratorio de Inteligencia Ambiental, Departamento de Ingeniería e Investigación Tecnológica
Universidad Nacional de la Matanza. Buenos Aires, Argentina.

izaradnik@gmail.com

Abstract—Es un hecho en permanente estudio y avance, la necesidad de administrar mejor los recursos energéticos, favorecer la protección del medio ambiente y responder a los requerimientos cada vez más exigentes de calidad de servicio y producto. Dadas las condiciones actuales del sector energético a nivel mundial, y en especial en la Argentina, las Redes Eléctricas Inteligentes e Internet de las Cosas constituyen poderosas herramientas para afrontar los futuros desafíos que impone el sector. En el presente trabajo se explica el desarrollo de una plataforma que permite la conexión de Redes Eléctricas Inteligentes a Internet de las Cosas. Se describe en forma general el funcionamiento de la plataforma, las etapas planteadas para la implementación del mismo, como la forma de llevarlas a cabo. A continuación se expone la selección del hardware asociado y el entorno de desarrollo para la implementación del firmware. La descripción del desarrollo del firmware fue dividida en dos partes, asociada cada una de ellas a uno de los tipos de comunicación que se implementaron. Finalmente se detallan los ensayos realizados y los resultados obtenidos del código desarrollado.

Keywords- *Redes Eléctricas Inteligentes; Internet de las Cosas; microcontroladores; sistemas operativos de tiempo; Ethernet; redes celulares.*

I. INTRODUCCIÓN

Según José Aga, Presidente de la Fundación Consejo para el Proyecto Argentino, el problema energético argentino no tiene relación con la disponibilidad de recursos naturales ni humanos, se trata de un fenómeno desencadenado por una política o la ausencia de ésta que ha llevado a una fuerte descapitalización del sector. Así como también que en lo eléctrico, abrir el camino de nuevas tecnologías como las Redes Eléctricas Inteligentes, los sistemas domésticos e industriales (solares y eólicos) conectados a la red es importante [1].

El concepto de Red Eléctrica Inteligente se basa en incorporar a la red eléctrica tradicional dispositivos electrónicos tales como medidores, sensores o mandos, vinculados a través de distintas tecnologías de comunicaciones, logrando la centralización y uso de la información para provecho de todos los actores involucrados y

así optimizar el sistema eléctrico en conjunto. Dicho concepto se puede asociar al de Internet de las Cosas (IoT), donde todas “las Cosas” conectadas a “Internet” pueden ser identificadas y gestionadas por equipos de igual forma que si lo fuesen por seres humanos.

La aplicación de estos conceptos se puede ver reflejada en el mantenimiento preventivo del sistema eléctrico de generación, distribución o transmisión, como se describe en el artículo “Los ensayos eléctricos: El principio de Pareto y la teoría del caos” [2]. En éste se plantea que la base del mantenimiento predictivo mediante técnicas ON-LINE, radica en que su aplicación no altere el funcionamiento del sistema, mientras se lo está aplicando, y se fundamente en generar un análisis de los parámetros funcionales, cuya evolución temporal permitirá detectar una falla antes de que ésta tenga consecuencias.

II. DESCRIPCION GENERAL

El trabajo que se presenta es parte de un sistema más complejo que involucra la lectura y/o medición de parámetros eléctricos, su transmisión a un nodo central, su concentración y posterior transmisión a un servidor para su análisis.

El proyecto se debió adaptar a la red de medidores eléctricos instalados, la cual posee una gran variedad de modelos. En primera instancia, se planteó la lectura de la energía eléctrica consumida, es decir los kWh. Parámetro que puede ser obtenido del frente de todos los medidores a través de un indicador luminoso. Como segunda etapa, se planteó la lectura a través de la interfaz infrarroja, no disponible en todos los medidores, de parámetros tales como energía activa, reactiva, aparente, potencia, coseno ϕ , etc. Como última etapa, se planteó el desarrollo de un hardware que permita la medición de los parámetros antes detallados, así como también valores eficaces de la tensión, la corriente, distorsiones, etc.

La plataforma recibe los datos a través de un puerto serie, en función de su configuración transmite éstos a un puerto TCP/UDP de un servidor o envía un correo electrónico a una dirección de correo previamente configurada. Dichas operaciones las podrá realizar a través de una conexión Ethernet o a través de un enlace GSM/GPRS. En caso de no poderse realizar la transmisión, los datos serán almacenados en la memoria de la plataforma.

El desarrollo planteado se dividió en 5 etapas:

- Recepción de los datos a través del puerto serie.
- Transmisión de datos sobre Ethernet a un socket TCP/UDP.
- Transmisión de datos sobre Ethernet a un servidor de correo electrónico.
- Transmisión de datos sobre GPRS a un socket TCP/UDP.
- Transmisión de datos sobre GPRS a un servidor de correo electrónico.

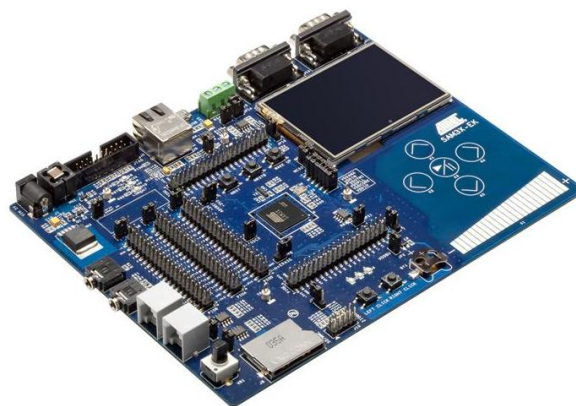


Figura N°1. Kit SAM3X-EK

Inicialmente se planteó que el firmware implementara las etapas detalladas previamente, pudiéndose seleccionar, en tiempo de ejecución, el tipo de interfaz a utilizar para la transmisión de los datos recibidos. Esto implicaba la implementación de una pila TCP/IP dentro del microcontrolador, lo cual permitiría la comunicación por Ethernet así como por GSM/GPRS. Dicha alternativa fue desestimada, ya que muchos módulos GSM/GPRS cuentan con una pila TCP/IP propia. Al utilizar dicho recurso, el código del microcontrolador se vuelve más sencillo, dando lugar al uso de uno de menores recursos. Con esto en mente se planteó el desarrollo de la comunicación por Ethernet utilizando un sistema operativo de tiempo real (RTOS) y la comunicación a través de GSM/GPRS implementando una máquina de estado.

III. HARDWARE Y ENTORNO

A. Hardware

Como base de nuestro desarrollo se tomó un microcontrolador basado en el núcleo Cortex-M3 de ARM. El producto seleccionado fue el SAM3X de ATMEL [3] y su kit de desarrollo el SAM3X-EK, el cual se puede ver en la Fig. 1. Si bien actualmente existe la familia ATSAM4E, la cual posee características superiores de procesamiento, al momento de iniciar el proyecto no estaba disponible. La elección se fundamentó en:

- La disponibilidad de conversores analógicos digitales para la medición de parámetros.
- Gran número de interfaces seriales (I2C, UART, SPI), que posibilita la comunicación con sensores y distintos módulos OEM.
- Interfaz Ethernet.
- Capacidad de procesamiento adecuada para trabajar con una pila TCP/IP.
- Sistema operativo de tiempo real portado para el microcontrolador.

Como elemento para realizar el enlace GSM/GPRS se empleó un modem basado en el módulo G24- EDGE [4], actualmente de la firma Telit, en la Fig. 2 se puede ver el módulo detallado. La elección se fundamentó en:

- Poseer una pila TCP/IP disponible.
- Disponibilidad de un modem, lo cual nos redujo el tiempo de desarrollo del firmware.

B. Entorno de desarrollo integrado(IDE)

Para la realización del proyecto se utilizó el software **ATMEL STUDIO 6.1.2665 Service pack 1** y el **Atmel Software Framework 3.9.1.780**, como se puede ver en la Fig. 3. La elección del entorno de desarrollo se basó en:

- El mismo es un software gratis, que se descarga de la web del proveedor.
- Posee compilador y debugger incorporados sin límite en tamaño de código ni número de archivos.
- Gran número de ejemplos disponibles a través de Atmel Software Framework.
- Posibilidad de migración a futuras plataformas.

IV. DESARROLLO

A. Comunicación Ethernet

Como base del código de la comunicación Ethernet se empleó el ejemplo **FreeRTOS with lwIP Example – SAM3X-EK** y se modificó para adaptarlo a las necesidades del proyecto. El ejemplo se enfoca en la utilización de un RTOS para la atención de tareas, las cuales en este ejemplo son: un web-server donde se puede ver el consumo y las estadísticas de las tareas y un servidor FTP en donde se puede subir y descargar archivos.

Dicho ejemplo realiza las siguientes acciones:

- Selecciona el clock a utilizar y configura su frecuencia.
- Configura la consola de debug.

- Configura pines de entrada y salida para el manejo de los LEDs,
- Genera tarea de parpadeo de un LED.
- Dispara la tarea de Ethernet, la cual crea las tareas de Webserver y FTP y luego se dispara el scheduler (secuenciador) del RTOS.

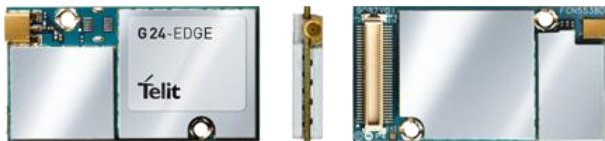


Figura N°2. Modulo G24-EDGE

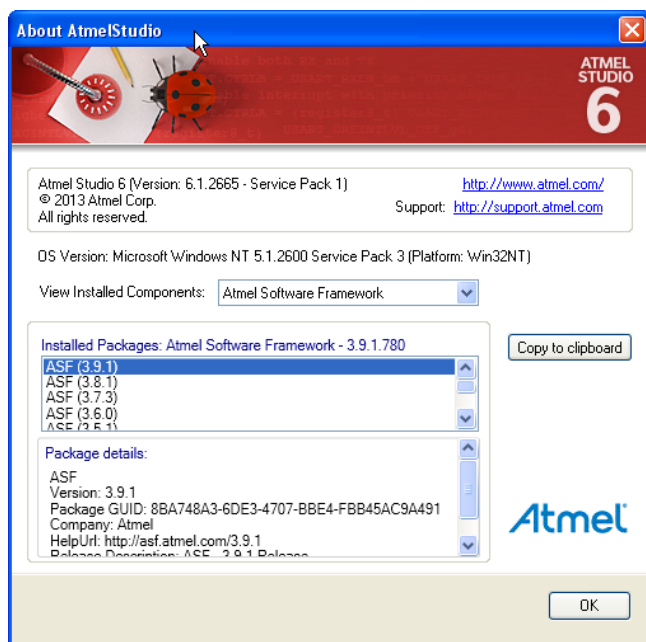


Figura N°3. Detalle de versión del IDE

Se agregó una nueva tarea de llamada Socket, encargada de loguearse por DHCP y realizar el enlace con el servidor para enviar los datos a un puerto TCP previamente configurado y/o a través de un correo electrónico. Las demás tareas se deshabilitaron, agregándose únicamente el manejo por interrupción de un puerto serie adicional por donde ingresarán los datos.

Del ejemplo original se modificaron 10 archivos y se agregaron 4 más al proyecto:

- **Archivos modificados:** Sam3x_ek.h, main.c, ethernet.c, opt.h, debug.h, cc.h, ethernetif.c, lwipopts.h, conf_uart_serial.h, conf_lwip_threads.h
- **Archivos agregados:** BasicSocket.c, BasicSocket.h, smtp.h, smtp.c

Los archivos agregados incorporaron las funciones necesarias para el manejo de los sockets y del protocolo SMTP para el envío de correos electrónicos.

Los archivos BasicSocket.c y BasicSocket.h fueron desarrollados para esta aplicación, mientras que los archivos smtp.c y smtp.h fueron descargados de un reservorio de código [5].

Durante las distintas pruebas que se fueron realizando se detectó un error asociado al debug, el cual presentaba un error de linker al habilitar el mismo en lwip. Dicho error fue solucionado, las correcciones fueron realizadas en los **debug.h**, **ethernetif.c** y **cc.h**.

B. Comunicación GSM/GPRS

Para el desarrollo del código de la comunicación GSM/GPRS no se utilizó ningún ejemplo; se partió desde la plantilla (template) que ofrece el ATMEL STUDIO para la placa de desarrollo seleccionada. Esto generó un proyecto con una estructura como la que se puede ver en la Fig. 4, donde existe un archivo **main.c** el cual sólo posee la inicialización de la placa. La misma configura los pines de las distintas interfaces disponibles en la placa, así como las entradas y salidas utilizadas para LEDs o pulsadores.

La configuración de la frecuencia de trabajo del microcontrolador se realiza en el archivo **conf_clock.h**, el que se encuentra ubicado en el directorio **config**. La configuración por defecto es una frecuencia de 84Mhz, la máxima del microcontrolador seleccionado.

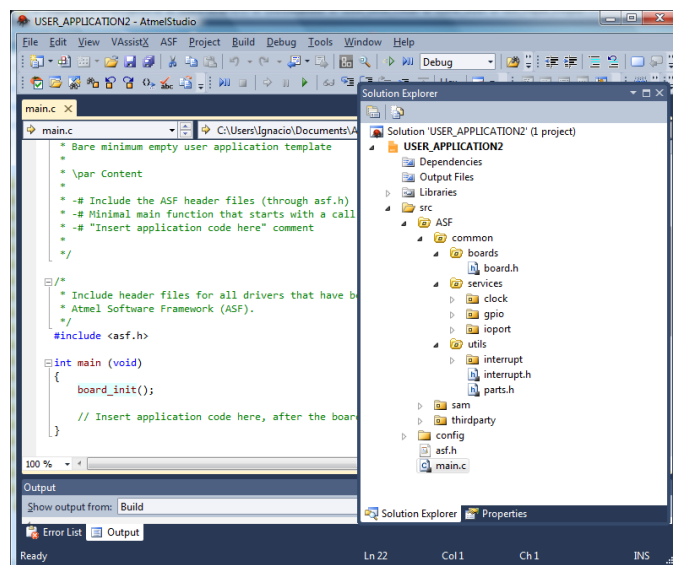


Figura N°4. Proyecto generado desde la plantilla.

A través del ATMEL Software Framework, herramienta integrada en el ATMEL STUDIO, se integraron al proyecto drivers para el manejo de los puertos serie e interrupciones, así como también rutinas de servicio para la generación de demoras y buffers circulares.

El código implementado es básicamente una máquina de estado, los cuales son definidos en una enumeración, Fig. 5.

```
typedef enum
{RESET,INIT,RECEP,SMTP,MESSAGE,PANIC
}Enum_Mod_State;
```

Figura N°5. Enumeración de estados.

Luego de la inicialización del sistema: configuración del watchdog, clock, activación de los periféricos y habilitación de las interrupciones, se ejecuta la máquina de estados. El primero de ellos es INIT, en el cual se cancela el eco de la comunicación del módulo GSM/GPRS, se verifica el registro en la red GSM y GPRS y se establece la conexión con el Gateway de la prestadora de servicio. Luego se pasa al estado RECEP, donde se aguarda la recepción de un dato. Recibido éste, es almacenado y se pasa al estado MESSAGE o SMTP dependiendo de la configuración del sistema. En el estado MESSAGE el dato es enviado a un puerto TCP en un servidor remoto, mientras que en el estado SMTP el dato es enviado a través de un correo electrónico. Finalizado el envío del dato por cualquiera de los métodos se vuelve al estado RECEP. El estado RESET se ejecuta si se detecta algún error, como ser: no recibir respuesta del modem, no obtener registro, no obtener conexión, una falla en la transmisión, etc. El estado PANIC está reservado para futuras aplicaciones.

Se destaca que si bien el módulo posee una pila TCP/IP, que facilitó el desarrollo de la aplicación, para el envío del correo electrónico se debió implementar el protocolo SMTP [6]. En dicha implementación la primera acción que se realiza es la apertura del puerto 25 de un servidor SMTP. Abierto éste se envían una secuencia de mensajes y se aguarda su respuesta. Los mensajes enviados incluyen: el usuario y la contraseña de la cuenta, codificados en base 16, el remitente y el destinatario del correo electrónico, el asunto y el mensaje propiamente dicho.

En la Fig. 6 se puede ver el código implementado para el envío de dichos mensajes y el análisis de la respuesta del mismo.

```
do{usart_write_line(USART_SERIAL,Gsm_Msg_Write[0]);us
art_write_line(CONSOLE_UART,Gsm_Msg_Write[0]);}
while((MOD_State==INIT)&&(USART_GSM.Error)&&(Gs
m_Proc_Rec(3,0)==0));
if ((MOD_State!=INIT)/(!USART_GSM.Error))break;
```

Figura N°6. Envío de comando y análisis de respuesta.

Como se puede ver en el código asociado a la instrucción “do”, se utiliza un vector para detallar el mensaje a enviar, dicho vector se puede ver en la Fig.7.

Con la instrucción “while” se espera la respuesta al mensaje o al comando, las cuales se encuentran detalladas en un vector de igual forma que los mensajes enviados, Fig. 8.

```
uint8_t *Gsm_Msg_Write[]={
"ATE0\r\n",
"AT+CREG?\r\n",
"AT+CGATT?\r\n",
"AT+MIPCALL=1,\"internet.gprs.unifon.com.ar\", \"
wap\", \"wap\" \r\n",
"AT+MIPODM=1,6001,\"190.16.194.147\",6001,0\r\n",
"AT+MIPCLOSE=1\r\n",
"+++",
"ATO\r\n",
"AT+MIPCALL?\r\n"
};
```

Figura N°7. Comandos enviados.

```
uint8_t *Gsm_Msg_Read[]={
"OK\r\n",
"ERROR\r\n",
"+CREG: 000,001\r\n",
"+CGATT: 1\r\n",
"+MIPCALL: 1,",
"+MIPODM: 1,1\r\n",
"+MIPCLOSE: 1,0\r\n",
"220",
"250",
"334",
"235",
"354",
"221"
}
```

Figura N°8. Respuestas esperadas.

V. ENSAYOS

A. Comunicación Ethernet

Para las pruebas de la comunicación Ethernet se utilizó el software Hercules [7], el cual se puede descargar en forma gratuita. El mismo puede ser utilizado para crear un cliente o servidor TCP o UDP, así como también usarlo para el manejo del puerto serie. Como servidor de SMTP se empleó un servidor propio el cual no implementa el protocolo SSL.

Para los ensayos de la comunicación Ethernet se seleccionó la opción TCP SERVER, donde se definió el Port 6001, la Fig. 8 muestra la pantalla del software Hercules.

Los objetivos de las pruebas fueron verificar:

- La obtención de una IP por parte del kit al conectarse a un router que asigna las direcciones a través de DHCP.
- La conexión contra el Server TCP corriendo en el software Hercules.
- Recepción de datos desde el Server.
- Recepción y transmisión al server de los datos que ingresen al puerto serie con el protocolo definido.

- Envío de un correo electrónico con los datos recibidos por el puerto serie.

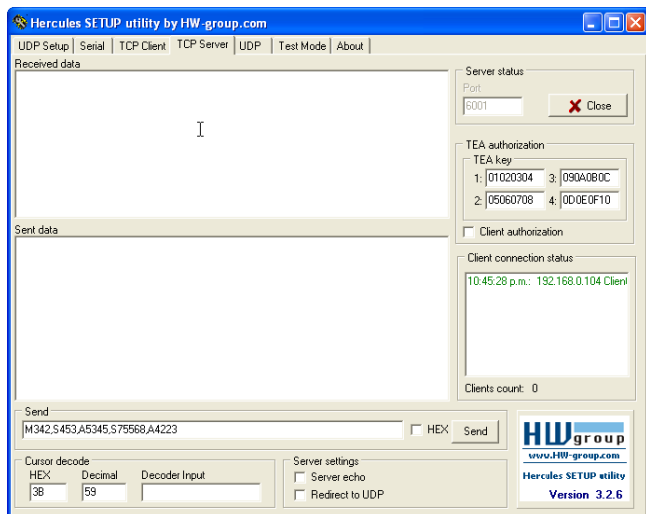


Figura N°8. Software Hercules.

Para realizar las pruebas se configuraron las direcciones IP del kit y del servidor en el archivo BasicSocket.c, Fig. 9.

```
// Definicion del puerto e IP del kit
#define SENDER_PORT_NUM 6000
#define SENDER_IP_ADDR "192.168.0.104"

// Definicion del puerto e IP del server
#define SERVER_PORT_NUM 6001
#define SERVER_IP_ADDRESS "192.168.0.106"
```

Figura N°9. Configuración de direcciones IP.

Así como también los datos asociados al servidor de SMTP y la cuenta de correo electrónico.

```
// first, set the server address
smtp_set_server_addr("mail.????-????.com");
// set the server port: only necessary if != default (which is 25)
smtp_set_server_port(25);
// set your authentication data (unless the server does not need
this, which is very rare)
smtp_set_auth(" ???@????-????.com", "???????");
// then, call one of the send functions, like:
smtp_send_mail(" ???@????-????.com",
"diego.turconi@gmail.com", buf_uart_rx_out, "Por Fin salio",
my_smtp_result_fn, NULL);
```

Figura N°10. Configuración SMTP.

Al conectarse el sistema, se empieza a reportar la actividad a través de la consola de debug. En la Fig. 11 podemos ver los mensajes en el momento de arranque del sistema.

Esto indica que el kit obtuvo la IP 192.168.0.104 y el socket 6000, que está intentado conectarse al servidor cuya IP fue configurada (192.168.0.103 al puerto 6001) y que dicha conexión se realizó con éxito. Esto se puede verificar en la

pantalla del Hercules, tal como se ve en la Fig. 8 en la ventana "Client connection status".

```
15/01/2014 22:45:19.607 [RX] - -- FreeRTOS with lwIP
Example --<LF><CR>
-- SAM3X-EK<LF><CR>
-- Compiled: Jan 15 2014 22:36:05 --<LF><CR>
15/01/2014 22:45:22.770 [RX] - LwIP: DHCP
Startedlwip_socket(PF_INET, SOCK_STREAM, 0) =
0<LF>
lwip_bind(0, addr=192.168.0.104 port=6000)<LF>
lwip_bind(0) succeeded<LF>
lwip_connect(0, addr=192.168.0.103 port=6001)<LF>
15/01/2014 22:45:23.761 [RX] - Network
upIP=192.168.0.104
15/01/2014 22:45:28.566 [RX] - lwip_connect(0)
succeeded<LF>
lwip_recvfrom(0, 0x2007fe80, 80, 0x0, ..)<LF>
lwip_recvfrom: top while sock->lastdata=0x0<LF>
```

Figura N°11. Mensaje de consola inicial.

A continuación se envió desde el software Hercules el siguiente mensaje M342,S453,A5345,S75568,A4223. La recepción del mismo pudo observarse a través de la consola, Fig. 12.

```
15/01/2014 22:59:20.266 [RX] - lwip_recvfrom:
netconn_recv err=0, netbuf=0x20075e98<LF>
lwip_recvfrom: buflen=28 len=80 off=0 sock-
>lastoffset=0<LF>
lwip_recvfrom(0): addr=192.168.0.103 port=6001
len=28<LF>
lwip_recvfrom: deleting netbuf=0x20075e98<LF>
M342,S453,A5345,S75568,A4223lwip_recvfrom(0,
0x2007fe80, 80, 0x0, ..)<LF>
lwip_recvfrom: top while sock->lastdata=0x0<LF>
```

Figura N°12. Recepción de mensaje en consola.

La siguiente prueba fue recibir un paquete a través del puerto serie y enviarlo al servidor de datos como al servidor de correo electrónico. En este punto se detectó un problema, ya que para que el mensaje que ingresó a través del puerto serie sea enviado al server, un mensaje del mismo debe llegar primero. Dicho inconveniente se encuentra en estudio. Por la extensión del log de esta operación, la misma no es incluida.

B. Comunicación GSM/GPRS

Para los ensayos de la comunicación GSM/GPRS se emplearon las mismas herramientas que en los ensayos de la comunicación Ethernet.

Los objetivos de las pruebas fueron verificar los mismos puntos que en el ensayo anterior con la salvedad del primer punto, el cual fue la obtención de una IP por parte del sistema al conectarse al Gateway de la prestadora de servicio.

La Fig. 13 nos muestra la operación de verificación de conexión a la red GPRS, su respuesta, la solicitud de IP y finalmente la obtención de la misma.

```
22/04/2014 22:04:51.479 [RX] - AT+CGATT?<CR><LF>
<CR><LF>
+CGATT: 1<CR><LF>
<CR><LF>
OK<CR><LF>
22/04/2014 22:04:52.503 [RX] -
AT+MIPCALL=1,"internet.gprs.unifon.com.ar","wap","wap"
<CR><LF>
<CR><LF>
OK<CR><LF>
22/04/2014 22:04:53.606 [RX] - <CR><LF>
+MIPCALL: 10.182.46.72<CR><LF>
```

Figura N°13. Verificación de conexión GPRS.

Esto verificó el primer punto. La Fig. 14 nos muestra el envío del dato recibido a través de un correo electrónico. Lo que permite la verificación del resto de objetivos, ya que el envío de un correo electrónico implementando el protocolo SMTP implica el envío de datos a un puerto TCP.

```
22/04/2014 22:05:03.533 [RX] -
AT+MIPODM=1,25,"mail.????-????.com",25,0<CR><LF>
<CR><LF>
OK<CR><LF>
22/04/2014 22:05:30.214 [RX] - <CR><LF>
+MIPODM: 1,1<CR><LF>
EHLO<CR><LF>
22/04/2014 22:05:37.264 [RX] - 220-ci1.toservers.com
ESMTP Exim 4.82 #2 Tue, 22 Apr 2014 22:05:28 -0300
<CR><LF>
220-We do not authorize the use of this system to transport
unsolicited, <CR><LF>
220 and/or bulk e-mail. <CR><LF>
250-ci1.toservers.com Hello 200-081-038-
043.wireless.movistar.net.ar [200.81.38.43]<CR><LF>
250-SIZE 52428800<CR><LF>
250-8BITMIME<CR><LF>
250-PIPELINING<CR><LF>
250-AUTH PLAIN LOGIN<CR><LF>
250-STARTTLS<CR><LF>
250 HELP<CR><LF>
auth login<CR><LF>
***** En base 64 *****
22/04/2014 22:05:38.526 [RX] - 334
VXNlcm5hbWU6<CR><LF> Username:
U??5????IG1hew==<CR><LF>
```

```
22/04/2014 22:05:39.469 [RX] - 334
UGFzc3dvcmQ6<CR><LF> Password:
B?????hdGU=<CR><LF>
22/04/2014 22:05:40.349 [RX] - 235 Authentication
succeeded<CR><LF>
MAIL FROM:???@ ???- ????.com<CR><LF>
22/04/2014 22:05:41.692 [RX] - 250 OK<CR><LF>
22/04/2014 22:05:42.619 [RX] - RCPT
TO:diego.turconi@gmail.com<CR><LF>
22/04/2014 22:05:43.786 [RX] - 250 Accepted<CR><LF>
```

```
22/04/2014 22:05:44.633 [RX] - DATA<CR><LF>
22/04/2014 22:05:45.432 [RX] - 354 Enter message, ending
with "." on a line by itself<CR><LF>
Subject: PRUEBA TUNEL<CR><LF>
From: ???@????-????.com<CR><LF>
To: diego.turconi@gmail.com.ar<CR><LF>
M342,S453,A5345,S75568,A4223<CR><LF>
.<CR><LF>
22/04/2014 22:05:46.615 [RX] - 250 OK id=1Wcld8-001eei-
QZ<CR><LF>
QUIT<CR><LF>
22/04/2014 22:05:47.431 [RX] - 221 ci1.toservers.com
closing connection<CR><LF>
<CR><LF>
+MIPSTAT: 1,1<CR><LF>
```

Figura N°14. Envío de correo electrónico.

VI. CONCLUSIONES

Se ha logrado cumplir parcialmente con los objetivos planteados, quedando por resolver el problema presentado en el envío de los datos a través de la interfaz Ethernet.

Sobre la base del firmware actual se contemplan las siguientes mejoras:

- Implementar un buffer circular que permita almacenar al menos 10 mensajes en la plataforma.
- Configuración de parámetros de operación por puerto serie. Parámetros tales como la dirección IP y el puerto del servidor, la dirección de correo electrónico destino, la configuración del servidor de SMTP, etc.
- Reconocimiento automático de la IP asignada por el servidor de DHCP.
- Mejorar la implementación del protocolo SMTP de forma que pueda ser utilizado para trabajar con cuentas con seguridad SSL.

REFERENCIAS

- [1] Martin J.Garcia, "La situación energética en primera fila," revista Megavattios vol.391, pp 22-23, noviembre 2013.
- [2] Martin J.Garcia, "Los ensayos electricos: Principio de Pareto y la Teoría del Caos," revista Megavattios vol.391, pp 46-52, noviembre 2013.
- [3] Atmel Corp., " AT91SAM ARM-based Flash MCU, SAM3X-SAM3A Series" Fecha de consulta: julio 6, 2014. Disponible en: <http://www.atmel.com/Images/doc11057.pdf>
- [4] Telit Communications PLC, "G24-EDGE," Fecha de consulta: julio 6, 2014. Disponible en: http://www.telit.com/telit/Pulsar/en_US.Store.display.1029/g24-edge
- [5] Goldsimon, "smtp clien using the lwIP raw API," Fecha de consulta: julio 6, 2014. Disponible en: <https://github.com/yiannisylwip-allnetworks/tree/master/apps/smtp>
- [6] HW Group, "Hercules SETUP utility", Fecha de consulta: julio 6, 2014. Disponible en: http://www.hw-group.com/products/hercules/index_en.html
- [7] J.Klensin, "Simple Mail Transfer Protocol," Fecha de consulta: julio 6, 2014. Disponible en: <http://tools.ietf.org/html/rfc5321>