

Sistema Embebido de Control, Monitoreo y Seguimiento Satelital Mediante SMS

Miguel Correia y Carlos Reynoso

Departamento de Ingeniería Eléctrica y de Computadoras,
Universidad Nacional del Sur,
8000 Bahía Blanca, Argentina

Director: Ing. Leonardo Ordinez
Cátedra: Computadoras Digitales e Interfases
Departamento de Ingeniería Eléctrica y Computadoras
Universidad Nacional del Sur

Abstract. Este artículo comprende el desarrollo de un sistema embebido capaz de sensar datos y accionar dispositivos remotamente mediante el uso de la red inalámbrica GSM.

Keywords: Monitoreo, Control, Seguimiento Satelital, GSM, GPS, SMS, PDU

1 Introducción

La evolución de la electrónica digital ha llevado consigo a la miniaturización de dispositivos y a la reducción de su consumo energético, lo que ha desencadenado una masiva popularización de los dispositivos móviles, como es el caso de los celulares, PDA, Palm, etc. A su vez, los avances en las comunicaciones inalámbricas han hecho posible que se establezcan redes de bajo costo y amplia cobertura para la transmisión de datos. Actualmente la red con mayor auge es el Servicio Global para la comunicación Móvil (GSM, de sus siglas en inglés), que está orientada principalmente a la telefonía digital, por lo que también ofrece la posibilidad de transmitir paquetes de datos, como es el caso del servicio de mensajes cortos (*Short Message Service*: SMS en inglés) que permite mandar pequeños textos, en tiempos relativamente cortos, a un destinatario particular localizado en cualquier lugar del planeta con cobertura GSM.

Lo anterior también ha contribuido a que las áreas de monitoreo y control adquieran nueva relevancia. Ya no sólo a nivel industrial, sino también a nivel doméstico para aplicaciones tales como sistemas de vigilancia, automatización hogareña (domótica) y/o seguimiento satelital.

Se pueden encontrar diversas áreas en las que el monitoreo y control a distancia se puede aplicar para mejorar aspectos de la actividad, por ejemplo: (1) en la agroindustria resulta de interés poder medir magnitudes como humedad del suelo,

nivel de agua de tanques, intensidad del viento, y demás medidas representativas de un sector, para luego tomar decisiones de control o en caso de un evento no deseado poder estar informado a distancia y actuar en consecuencia [1]. (2) Al sector transportista le interesa conocer la posición y el movimiento de un vehículo, como así también el estado de sensores vinculados a sus parámetros operativos [2]. (3) A las industrias que ceden su valiosa mercadería a empresas transportistas les es de gran ayuda tener un monitoreo constante de las cargas transportadas, y tomar ciertas medidas de control [3]. (4) A un usuario particular, le puede interesar localizar su vehículo, activar su alarma, o encender y apagar dispositivos electrónicos del hogar.

Para dar solución a estas problemáticas, se decidió diseñar un equipo genérico que utilice la tan extendida red inalámbrica móvil GSM, y el servicio gratuito de posicionamiento global GPS (*Global Positioning System*, en inglés), para posibilitar control y monitoreo a distancia mediante SMS. Inicialmente el sistema contará con un GPS, 4 salidas digitales, y una entrada analógica. Mediante mensajes de texto, un usuario podrá solicitar el estado de cualquiera de estas fuentes, y/o comandar las salidas. De esta forma, el sistema será capaz de comportarse como un mando a distancia, un sistema de seguimiento satelital, y un monitor de parámetros analógicos.

El software de control del sistema será implementado con un microcontrolador *Atmel 89c51*. Éste por su arquitectura y prestaciones, es muy utilizado en electrónica de control.

Siguiendo con el espíritu de que el sistema sea genérico, hardware y software serán diseñados en base a un criterio modular, para que una vez implementado, resulte flexible para ampliaciones o adaptaciones a casos específicos.

2 Desarrollo

Inicialmente se planteó un esquema de hardware y software modular, que permitió separar el proyecto en pequeños subproblemas a resolver para luego integrarlos en un todo que solucione el problema inicial.

El “cerebro” del sistema es el microcontrolador ATMEL 89C51ED2 (Fig. 2). Éste cuenta con una arquitectura tipo Harvard de 8 bits, y 64 Kbytes de memoria de programa, 4 puertos de 8 bits bidireccionales y multifuncionales, que ofrecen posibilidades de conexiones de datos paralelas y serie (sincrónica y asincrónica).

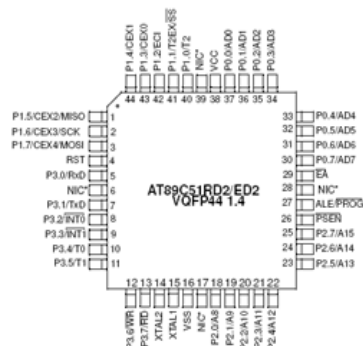


Fig. 2: Microcontrolador

Debido a que tanto el GPS como el teléfono se comunican con el controlador mediante una interfaz serie, y éste sólo cuenta con un puerto de este tipo, se debió expandir dicha interfaz mediante un multiplexor comandado por software.

La conversión analógica/digital se implementó con un MCP3208. Éste cuenta con 8 canales analógicos, y realiza una conversión de 12 bits (4096 valores posibles). Los datos convertidos los transmite por una interfaz de comunicación serie estandarizada en la industria: *Serial Peripheral Interface*, abreviado SPI [4]. El microcontrolador AT89C51ED2 incorpora un controlador para esta interfaz, lo cual simplificó notablemente la implementación del software.

El problema fue separado en partes, y resuelto según el siguiente orden:

1. Visor (Display)
2. Multiplexor serie
3. GPS
4. Teléfono
5. Relés
6. AD

Por cada componente de hardware se creó el módulo de software asociado.

2.1 Display

El LCD seleccionado para implementar el sistema consta de 2 líneas de 16 caracteres que son manejadas por un controlador interno que simplifica la conexión con otros dispositivos mediante una interfaz paralela.

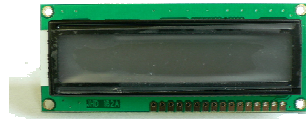


Fig. 3: Display 16 x 2

El dispositivo se comporta como un banco de memoria, que refleja el contenido de cada byte en el display imprimiendo el carácter ASCII correspondiente.

El manejo del display se realizó en modo estándar, requiriendo 10 líneas de control:

- *8 de datos*, mediante las cuales se transfieren códigos de caracteres a imprimir y/o comandos.
- *Línea RS*: indica si lo contenido en las líneas de datos es un comando o un carácter imprimible.
- *Línea Enable*: con un pulso en esta línea, el controlador del LCD toma el contenido de la línea de dato.

Lo anteriormente dicho se puede ver más claramente en el esquema temporal de la Fig. 4:

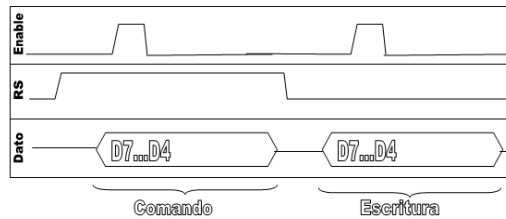


Fig. 4: Diagrama temporal del manejo del LCD

Las 8 líneas de datos fueron conectadas al puerto 1 del microcontrolador, RS y Enable a dos líneas del puerto 2. Por lo tanto, el esquema de conexionado es el mostrado en la figura 5.

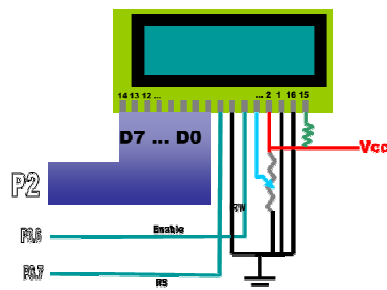


Fig. 5: Conexionado del display

Rutinas asociadas

```
void LCD_init()
```

Esta función es la encargada de enviar los comandos de inicialización necesarios al LCD. Configura el modo de comunicación con el microcontrolador, y parámetros de visualización tales como el tamaño de la letra.

```
void LCD_command(comando), LCD_putchar(carácter)
```

Estas funciones son muy similares, excepto por el nivel lógico del pin RS. Ambas envían un dato de 8 bits, y luego el display en función del estado de RS lo interpreta como comando o como carácter respectivamente.

```
void LCD_Escribir(columna, fila, cadena)
```

Permite escribir en la posición indicada por columna, fila una cadena de caracteres. Inicialmente manda los comandos de posicionamiento, y luego invocando repetitivamente la rutina LCD_putchar, envía uno a uno los caracteres contenidos en el arreglo "cadena".

2.2 Multiplexor Serie

Dado que tanto el GPS como el teléfono deberán escribir datos sobre la entrada del bus serie del microcontrolador, se debió multiplexar el canal. El multiplexor implementado se basa en dos buffers y un inversor comandados por uno de los pines del microcontrolador. Esto permite que desde el software con una simple primitiva, se conmute cualquiera de los dos dispositivos: `SetGPS()` y `SetTEL()` para conmutar el GPS y el Teléfono respectivamente.

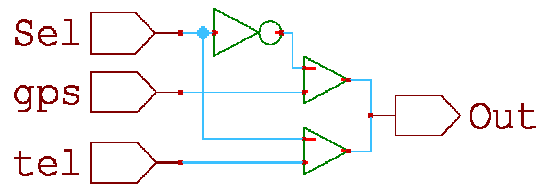


Fig. 6: Esquemático del multiplexor implementado

Como se verá mas adelante, el teléfono se controla enviando comandos por el puerto serie, por otra parte el GPS presenta continuamente sus datos en el puerto serie sin la necesidad de que el microcontrolador efectúe una consulta. En consecuencia, el canal de envío de datos del microcontrolador sólo debió vincularse al Teléfono, y no al GPS. Por lo tanto el Multiplexor serie sólo se utilizó en la línea de entrada del microcontrolador.

2.3 Modulo Receptor de GPS

Este dispositivo calcula por triangulación satelital su posición sobre la superficie terrestre. Luego procesa esta señal y la transmite por el puerto serie en forma de cadena de caracteres utilizando el protocolo estándar NMEA (del ingles *National Marine Electronic Assosiation*) [5]. El módulo utilizado transmite una gran variedad de datos, aunque sólo algunos son de interés para el sistema que se diseñó. Estos son: la posición -latitud, longitud y altura- y velocidad.

Cada sentencia del protocolo NMEA comienza con un signo "\$", luego le sigue un identificador de 5 caracteres, y por último valores alfa-numéricos separados por comas. Cuando el programa del controlador precisa un dato de posición, conmuta mediante el multiplexor la conexión serie con el GPS (el cual está constantemente enviando información independientemente de si alguien la recibe o no) y se mantiene a la espera de la sentencia \$GPRMC. Dicha sentencia contiene la información antes mencionada en formato de texto delimitada por comas. A continuación una rutina se encarga de hacer el 'parsing' y de organizar la información en variables globales de tipo *String* (llamadas *Latitud*, *Longitud* y *Velocidad*).

Rutina asociada:

```
int BuscarDatosGPS() :
```

Es la rutina que se encarga de hacer lo anteriormente dicho. Devuelve un valor de tipo entero el cual representa el estado del GPS: -1 cuando no se pudo establecer conexión, 0 si hay conexión, pero el GPS no está correctamente posicionado ó 1 si los datos son correctos

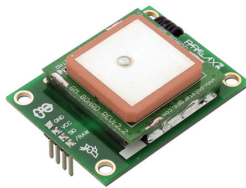


Fig. 7: Modulo GPS

2.4 Teléfono

El teléfono utilizado (Sony Ericsson K300a) admite su control mediante el puerto serie usando el estándar AT [6]. Éste es un protocolo de control para dispositivos de acceso telefónico que permite manejar la totalidad de sus funciones.

El microcontrolador hace uso de comandos que le permiten leer, borrar y escribir mensajes de texto. Para la lectura y escritura se utiliza el protocolo de encriptación PDU [7] que consiste en una forma particular de organizar la información de un mensaje de texto (número telefónico, configuración, y mensaje). Por lo tanto se requirieron desarrollar algoritmos de conversión de modo texto a modo PDU y viceversa: `ParsingMensaje()` y `CodificarMensaje()`.

Para la lectura de los mensajes entrantes se invoca la rutina `TEL_TraerMensaje()`, que devuelve un valor entero: -1 si el teléfono no responde, 0 si no hay mensajes nuevos ó 1 si hay mensajes nuevos

En caso de que haya un mensaje nuevo éste es guardado en el arreglo de caracteres `SMS_completo`. A continuación se debe hacer la decodificación de la información, separando el número telefónico del remitente y el mensaje encriptado para convertirlo a modo texto. Esto se realiza llamando a las rutinas `ParsingMensaje()` y luego `ExtraerTexto()`. Al finalizar, el texto y el número estarán en las cadenas de caracteres `'TXTascii'`, y `'Telefono'` respectivamente.

Para enviar un mensaje de texto, se debe indicar el número de destinatario y el texto a enviar en las cadenas `'Telefono'` y `'TXTascii'`, y luego llamar la rutina `EnviarMensaje()`, que se encarga de codificar la información en PDU (invocando a `CodificarMensaje()`) y transmitirla al teléfono.

Para eliminar los mensajes de texto se puede utilizar la rutina `BorrarMensaje(n)`, donde 'n' especifica el índice del mensaje a borrar.



Fig.8: Teléfono utilizado
(K300a)

2.5 Driver Relés

Se configuraron como salidas cuatro pines del puerto 3 del microcontrolador, y se conectaron mediante transistores configurados como conmutadores a 4 relés. Los transistores le permiten al microcontrolador manejar corrientes elevadas, lo cual es necesario para lograr encender los relés sin sobrecargar el microcontrolador. Los relés son llaves comandadas eléctricamente. Constan de una bobina que al excitarse genera un campo magnético capaz de mover un conjunto de piezas que cierran un circuito eléctrico totalmente aislado del que excitó la primera bobina. Esto le permite al controlador conmutar circuitos que no necesariamente tienen un vínculo eléctrico con él. De esta forma, un dispositivo que funciona con 12 Volts, puede manejar equipos de mayor tensión (motores, electrodomésticos, iluminaria, etc.).

Cada pin está vinculado a un bit dentro del byte que representa el puerto. A nivel software se ve como una posición de memoria. Para conmutar alguno de los relés se le debe asignar un 1 al bit asociado. Se eligió el valor lógico alto para encendido, por compatibilidad con el modelo de transistor utilizado en el driver.

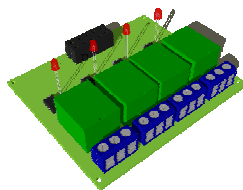


Fig. 9: Driver relés

2.6 Conversor Analógico-Digital (AD)

Los conversores analógicos-digitales permiten codificar tensiones analógicas en palabras binarias. Dichas tensiones generalmente provienen de sensores que poseen la capacidad de transducir magnitudes físicas en valores de tensión, para luego poder ser leídos, procesados y/o transmitidos por circuitos eléctricos.

Por ejemplo, un sensor de temperatura transforma la temperatura T de un recinto en una tensión V mediante la función $V(T)$. Luego un conversor AD transformaría dicha señal continua en una palabra digital, para que un procesador la interprete y

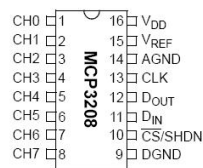


Fig. 10: Conversor AD

actúe en consecuencia, accionado actuadores (calentadores, ventiladores, etc.) que le permitan controlar la temperatura [8].

En este proyecto se utilizó un conversor MCP3208 (como el mostrado en la figura 10), que cuenta con 8 canales analógicos con una resolución de 12 bits. Ésto permite medir hasta 8 magnitudes analógicas pudiendo diferenciar 4096 valores (si la señal variara entre 0 y 5 Volts, se podrían ver cambios de 130 micro-volts). El valor convertido puede ser leído mediante una interfaz SPI.

Sólo uno de los canales fue utilizado, quedando los otros 7 disponibles para futuras ampliaciones. El microcontrolador consulta el estado de las entradas analógicas mediante una comunicación SPI, haciendo uso de la rutina `LeerCH0()`. Dado que el valor leído es de 12 bits, éste es devuelto en dos bytes, que son guardados en las variables globales `ADhigh` y `ADlow`. La primera contiene los 4 bits más altos, y la segunda los restantes 8.

3 Integración del Sistema

Una vez resueltos todos los módulos planteados, se comenzó a trabajar en la integración de los mismos. Con este fin se desarrolló el programa principal que hace uso de todas las rutinas. Éste cuenta con un ciclo que repetitivamente comprueba el arribo de nuevos mensajes y actúa en función de ello. El diagrama de flujo de la Fig.11, esquematiza el funcionamiento del programa principal:

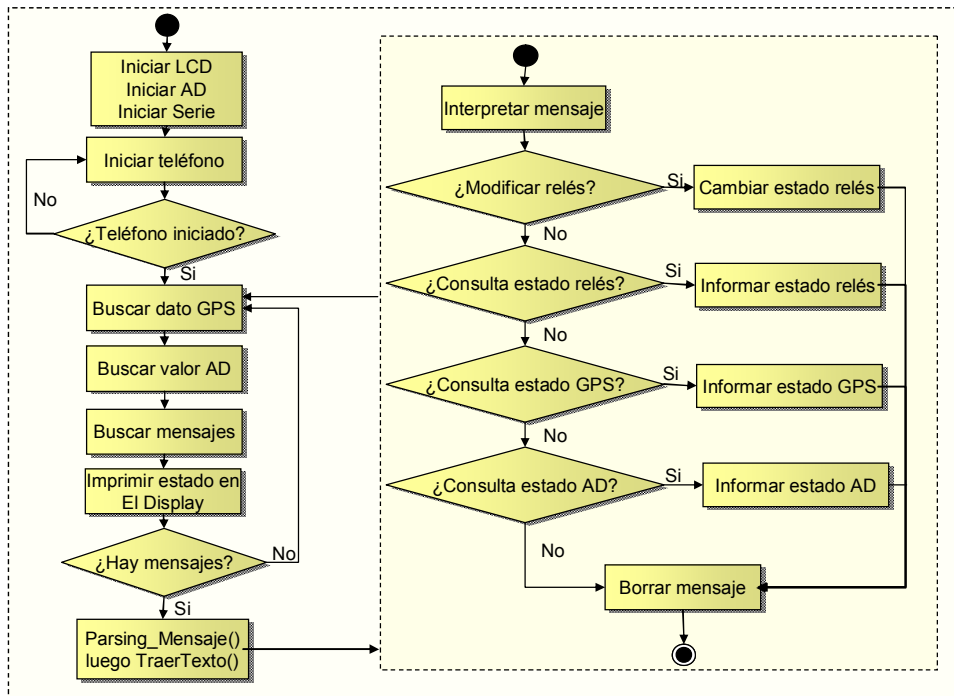


Fig. 11: Diagrama en bloques del programa

El programa comienza inicializando los dispositivos conectados y las variables asociadas a estos. Tanto el LCD, como el AD y el puerto serie se inician en forma directa (no se espera respuesta), en cambio al iniciar el teléfono se espera una respuesta para discriminar si está conectado o no, en caso de estarlo continua la ejecución, y en caso contrario muestra un mensaje en el LCD y se detiene hasta que el usuario conecte el teléfono. Luego entra en el ciclo principal donde se mantiene a la espera del arribo de nuevos mensajes. Al recibir un nuevo mensaje hace un ‘parsing’ separando el mensaje y el número telefónico (para luego, en caso de ser necesario, responder con un mensaje de texto al teléfono que efectuó la consulta). En caso de reconocer el mensaje como una instrucción válida la ejecuta. Las instrucciones válidas son las siguientes:

- *Rele?*
 - Devuelve un mensaje de texto con el estado de los cuatro relés con la estructura *ReleXXXX*, donde *X* representa el estado lógico de la salida con un 0 o un 1 simbolizando el apagado y encendido respectivamente.
- *ReleXXXX*
 - Asigna el estado de las salidas lógicas. *XXXX* es una palabra de 4 caracteres donde cada dígito representa un relé, y puede tomar los

valores 0 o 1 si se quiere apagar o encender, o X si no se desea modificar el estado.

- *Ana?*
 - Devuelve un SMS con el valor de la entrada analógica con la forma *ADhigh.ADlow*, ambas magnitudes expresadas en base 10.
- *Gps?*
 - Devuelve los datos de posición y velocidad con la siguiente sintaxis: “Lat.: (latitud) Lon.: (longitud) Alt.: (altura) Vel.: (velocidad)”

Luego borra el mensaje recibido, y retoma el ciclo repetitivo.

Si durante la ejecución del programa se desconecta el teléfono, el sistema reconoce el evento y vuelve al inicio, esperando la reconexión del mismo y lo informa en el display con el mensaje “Teléfono desconectado”.

En esta aplicación el display muestra constantemente el estado del teléfono (desconectado, en espera, recibiendo mensaje), del GPS (Conectado, desconectado, sin posición), y de la entrada analógica (valor leído).

La edición y compilación del código de programa del sistema se desarrollo con el software de desarrollo Keil uVision3. Este consta de un IDE (Entorno de desarrollo integrado) y herramientas para trabajar con diversos microcontroladores, entre ellos el ATMEL 89C52ED2.

La interfaz de conexión entre la PC y la tarjeta programadora se realizo mediante el puerto serie R232. El control de datos se realizó a través del software FLIP, éste es un programa que permite la escritura y lectura sobre la memoria del microcontrolador.

4 Escalabilidad del sistema

El sistema prevé que una aplicación particular requiera más salidas y entradas por lo que pequeños cambios en hardware y software ampliarían el sistema a 16 salidas y/o entradas digitales y 8 entradas analógicas. Además dado que el software solo utiliza un 25% de la memoria interna del microcontrolador, se podría utilizar el 75% restante para ampliaciones. Por ejemplo, hacer que el sistema no sólo responda órdenes sino que también tome decisiones de control propias o envíe mensajes anunciando eventos (relacionados con variables internas, o algún reloj de tiempo real).

En ciertos casos resultaría útil tener control desde el mismo dispositivo, sin necesidad del uso de la mensajería instantánea. Para estas aplicaciones se podría agregar sin mucha complejidad extra un teclado y un sistema de menús basado en el display.

5 Conclusiones

En el presente trabajo se expuso el desarrollo modular de un sistema embebido de control, monitoreo y seguimiento satelital mediante SMS, controlado por un

microcontrolador Atmel 89c51ed2. El sistema responde a comandos enviados por mensaje de texto, siendo capaz de controlar salidas digitales, e informar el estado de una entrada analógica, y sus coordenadas geográficas y velocidad.

La modularidad del sistema simplificó significativamente el trabajo. Se estudió y reconoció el hardware para un mejor manejo a nivel de código de programa, lo que permitió asignar un módulo a cada componente de hardware en forma separada e independiente. La utilización del programa Keil uVision3 facilitó esta tarea mediante el uso del lenguaje C.

Se presentaron dificultades al encriptar y desencriptar los mensajes de textos ya que el teléfono utilizado manejaba sólo modo PDU. Si se hubiese utilizado un teléfono que maneje modo texto, la implementación hubiese resultado más sencilla y requerido menos memoria de programa para llevar a cabo el parsing de los mensajes, con esto, se contaría con más memoria para ampliaciones.

El no utilizar todos los puertos, ni toda la memoria del microcontrolador permite que el proyecto pueda ser ampliado o adaptarse a otras necesidades sin mayores complicaciones, con lo cual se ve claramente que el desarrollo de sistemas basados en microcontroladores presentan gran versatilidad para la adaptación de aplicaciones a casos particulares.

La implementación final del sistema fue exitosa, alcanzando y sobrepasando los objetivos planteados inicialmente.

6 Referencias

1. J.M. McKinion, J.L. Willers and J.N. Jenkins, Wireless local area networking for farm management. ASAE Paper: 04-3012, The American Society of Agriculture Engineers, St. Joseph, Michigan, USA (2004).
2. L.S. Guo and Q. Zhang, A wireless LAN for collaborative off-road vehicle automation, *Proceedings of Automation Technology for Off-Road Equipment Conference* Chicago, Illinois, USA, July 26–27 (2002), pp. 51–58.
3. G. Gebresenbet, D. Ljungberg, G. Van de Water and R. Geers, Information monitoring system for surveillance of animal welfare during transport, *Proceedings of the 4th European Conference in Precision Agriculture* Berlin, Germany, June 14–19 (2003).
4. Microchip: “SPI™: Overview and Use of the PICmicro Serial Peripheral Interface”, <http://ww1.microchip.com/downloads/en/devicedoc/spi.pdf> - (2002).
5. NMEA: “NMEA data”, <http://www.gpsinformation.org/dale/nmea.htm>
6. Sony Ericsson: “AT Commands Online Reference” (October 2004)
7. Developershome: “Short Message Service / SMS Tutorial”, <http://www.developershome.com/sms/>
8. Bertereix German, Tellarini Andres: Desarrollo de un Control de Temperatura Embebido (2009)