

# **Sistemas embebidos 2.**

## **“Practica 1:** sistema comunicación cliente-servidor**.”**

# **Profesor: Rodrigo Aldana López.**

# **Alumnos:**

# Gustavo Araiza Obeso ie699209.

# Celis Tapia Daniel ie689705.

# Fecha:

# 04/12/2017

## INTRODUCCION:

Como parte de la materia de Sistemas Embebidos Basados en Microprocesadores que imparte el profesor José Luis Pizano Escalante, se llevó a cabo del desarrollo del proyecto final, que consistió en la implementación de un sistema de seguridad controlado basado en el módulo RFID. Este proyecto fue realizado por el Gustavo Araiza Obeso (IE699209).

En el siguiente reporte se analizarán todos los aspectos que fueron necesarios para la elaboración de este proyecto final, empezando por el planteamiento del problema, en otras palabras, las especificaciones de la misma. Así mismo se nombrarán los componentes y conceptos que se requirieron para la práctica. De ahí se pasará a hablar sobre el marco teórico, donde se explican todos los elementos teóricos que fueron necesarios conocer o investigar, elementos que se encuentran siendo aplicados en dicho proyecto.

Posteriormente se encuentra la sección de análisis y/o diseño teórico. también se incluirán los diagramas de flujo que se utilizaron para llegar al resultado teórico. Después se pasa a lo que es el desarrollo práctico, es decir, se describirán cada uno de los pasos seguidos en la elaboración de la parte teórica.

## Marco Teórico

**Resistencia:** (Garcia Alvarez, 2015)Resistencia eléctrica es toda oposición que encuentra la corriente a su paso por un circuito eléctrico cerrado, atenuando o frenando el libre flujo de circulación de las cargas eléctricas o electrones. Cualquier dispositivo o consumidor conectado a un circuito eléctrico representa en sí una carga, resistencia u obstáculo para la circulación de la corriente eléctrica.

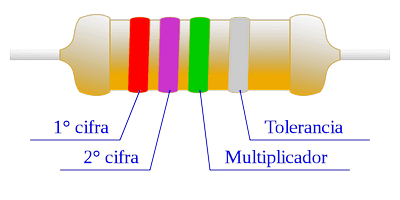


Figura 1. Resistencia

**RTC:** (Wikipedia, Wikipedia, 2017)Un Reloj en tiempo real (en inglés, *real-time clock*, **RTC**) es un reloj de un ordenador, incluido en un circuito integrado, que mantiene la hora actual. Aunque el término normalmente se refiere a dispositivos en ordenadores personales, servidores y sistemas embebidos, los RTC están presentes en la mayoría de los aparatos electrónicos que necesitan guardar el tiempo exacto.

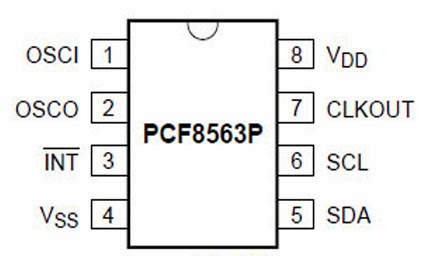


Figura 2. RTC

**EEPROM:** (Wikipedia, Wikipedia, 2017)Las memorias de tipo EEPROM es un derivado de la memoria sd. Es un dispositivo en forma de tarjeta, que se encuentra orientado a realizar el almacenamiento de grandes cantidades de datos en un espacio reducido, permitiendo la lectura y escritura de múltiples posiciones de memoria en la misma operación tienen como principal cualidad el permitir el almacenamiento y la sobre-escritura de datos por medio de los voltajes de operación norma de los circuitos electrónicos, además sostienen la información por muchos años sin fuente de alimentación.

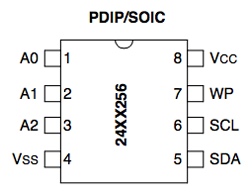


Figura 3. EEPROM

Los módulos de **bluetooth HC-05** y HC-06 son módulos muy populares para aplicaciones con microcontroladores PIC y Arduino. Se trata de dispositivos relativamente económicos y que habitualmente se venden en un formato que permite insertarlos en un protoboard y cablearlo directamente a cualquier microcontrolador, incluso sin realizar soldaduras. En esta entrada del blog vamos a explicar un poco del funcionamiento de estos módulos y como configurarlos. También abordaremos las diferencias entre el HC-05 y el HC-06.



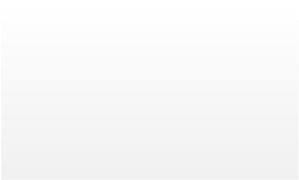
Figura 4. HC05

## Planteamiento del proyecto:

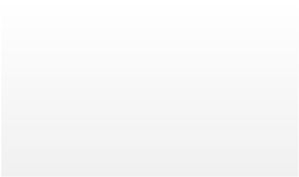
Consiste en la implementación de sistema comunicación cliente-servidor. Este sistema cuenta con 2 terminales de acceso al servidor, una a través de una terminal 1 en la PC (Tera Term) y la terminal 2 que se accede a través de un dispositivo móvil como un Smartphone.

El sistema a diseñar está compuesto por

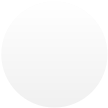
* Servidor (K64)
* Módulo de bluetooth HC05
* Un display Nokia 5110 (terminal del servidor)
* Botones B0-B5
* Una memoria EEPROM 24LC256
* Un Reloj de tiempo real MCP7940M



LCD Nokia

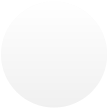


Freedom board



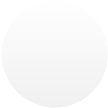
B

0



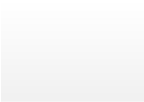
B

1



B

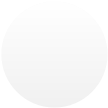
2



24

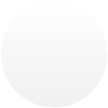
LC

256



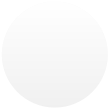
B

3



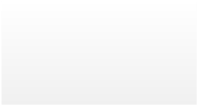
B

4

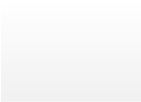


B

5

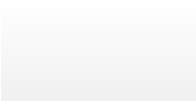


bluetooth



PCF

8583



Ter

a

Term

Termin

a

l

1

Terminal

2

Figura 5. diagrama

# Descripción funcional

En cada terminal del sistema de comunicaciones se desplegara el siguiente menú:

Opciones:

1. Leer Memoria I2C
2. Escribir memoria I2C
3. Establecer Hora
4. Establecer Fecha 5) Formato de hora
5. Leer hora
6. Leer fecha
7. Comunicación con terminal 2
8. Eco en LCD

***Convenciones***

En este documento el texto en color naranja y cursivo representa el texto que se debe visualizar en la terminal, por ejemplo

*Dirección de lectura: 0x0000*

*Longitud en bytes: 50*

Cuando en la descripción aparezca [ESC] o [enter] significa que se presionó la tecla escape o enter respectivamente.

## Leer Memoria I2C

Cuando se selecciona este menú la terminal debe pedir la dirección a leer (en hexadecimal) y la longitud de los datos a leer, por ejemplo:

*Dirección de lectura: 0x0000 Longitud en bytes: 50 Contenido:*

*Micros y DSP ITESO 2017*

*Presiona una tecla para continuar……*

## Escribir memoria I2C

Cuando se selecciona este menú la terminal debe pedir la dirección de escritura (en hexadecimal) y parara hasta que se presione la tecla enter, por ejemplo:

*Dirección de escritura: 0x0050 Texto a guardar:*

*Esta es una prueba de escritura [enter] Su texto ha sido guardado……..*

## Establecer hora

Este menú establece la hora del reloj de tiempo real al introducir la hora en formato hh/mm/ss y presionar la tecla enter

*Escribir hora en hh/mm/ss: 7:30:55 La hora ha sido cambiada….*

En caso de no poder comunicarse con el dispositivo I2C se debe mostrar un mensaje de error, en el LCD y la terminal que este accediendo a la lectura el RTC.

## Establecer fecha

Este menú establece la fecha que tendrá el reloj de tiempo real en formato dd/mm/aa y presionar la tecla enter, por ejemplo:

*Escribir fecha: en dd/mm/aa: 24:12:2014 La hora ha sido cambiada….*

En caso de no poder comunicarse con el dispositivo I2C se debe mostrar un mensaje de error.

## Formato de hora

Este menú cambia el formato que representa la hora, por ejemplo:

*El formato actual es 12h*

*Desea cambiar el formato a 24h si/no? Si El formato ha sido cambiado….*

En caso de no poder comunicarse con el dispositivo I2C se debe mostrar un mensaje de error.

## Eco en LCD

Cuando se selecciona esta opción todo lo que se escriba por la terminal aparecerá en el LCD del servidor hasta que se presione la tecla esc, por ejemplo

*Escribir texto:*

*Esta una práctica de micros y DSP [ESC] Termino la conexión….*

## Leer hora

Cuando se accede a este menú la hora se está actualizado constantemente, por ejemplo:

*La hora actual es:*

*5:30:54 am*

Nota, la hora que se presenta depende del formato hora activo. En caso de no poder comunicarse con el dispositivo I2C se debe mostrar un mensaje de error.

## Leer fecha

Cuando se accede a este menú se muestra la fecha actual, por ejemplo

*La fecha actual es*

*20/10/2014*

En caso de no poder comunicarse con el dispositivo I2C se debe mostrar un mensaje de error.

## Comunicación con terminal 2

Este menú permite la comunicación simultánea de la terminal 1 y 2. Él envió un mensaje se envía cuando se presiona la tecla enter y la comunicación termina cuando alguna de las dos terminales presiona la tecla ESC, por ejemplo

*Terminal 1:*

*Como esta?*

*Terminal 2:*

*Bien, aquí sufriendo con la práctica y tú? [ESC] Terminal 2 se desconectó…..*

## LCD Nokia

En todo momento se debe estar mostrando la hora y la fecha, en particular la hora se debe estar actualizando cada segundo. La hora no se mostrará mientras se esté en modo ECO LCD. Cuando la terminal finalice la comunicación en modo eco se volverá a mostrar la hora y fecha.

Además de configurar la hora y fecha a través de las distintas terminales, también se debe configurar con los botones B0-B5. La cantidad de botones y el menú de configuración lo establece el diseñador de la aplicación

## Análisis y/o diseño teórico:

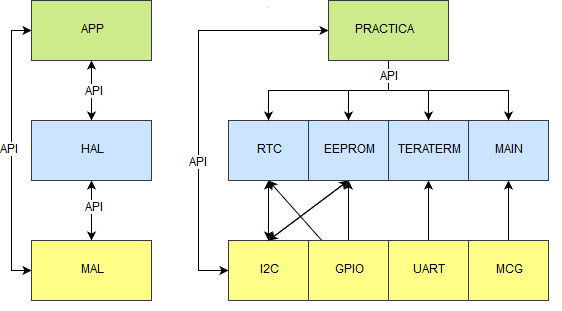
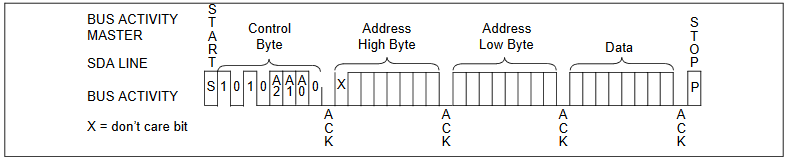
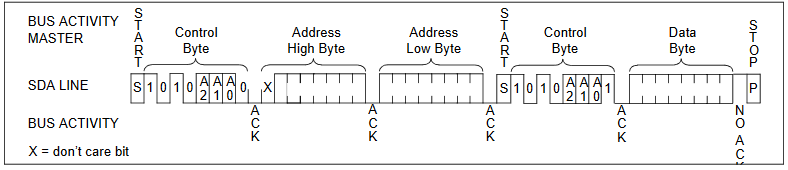


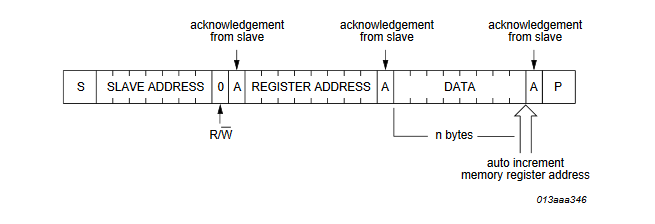
Figura 6. Diagrama de capas



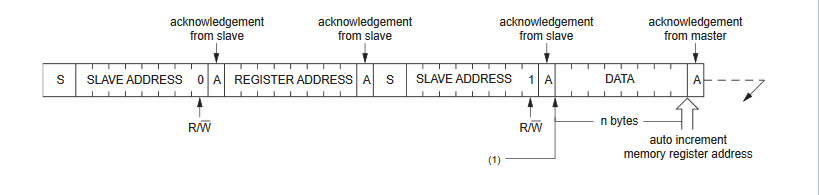
Escritura de un dato en una dirección



Lectura de un dato en una dirección



Escritura de un registro del RTC.



Lectura de un registro del RTC.

## Desarrollo práctico:

Para el desarrollo de la práctica, necesitamos crear una serie de tareas que nos permitan ejecutar correctamente las secuencias. Básicamente, necesitamos tareas que comuniquen lo que tenemos que hacer en el lado del hardware, y una serie de secuencias que llaman a las tareas y esperan o su finalización para ejecutar los pasos necesarios para que funcionen correctamente.

En el diagrama, podemos ver como el punto clave de nuestra secuencia sería el RTC seguido de escape, ya que es el que tiene la más alta prioridad. Hay algunas secuencias de inicialización que crean las tareas correspondientes para cada terminal, pero se eliminan tan pronto como se inician los menús.

## Desarrollo:

Los recursos están protegidos por un mutex para evitar problemas prioritarios. La mayoría de las llamadas se realizan a través de grupos de eventos, que a su vez son los que permiten que las secuencias funcionen correctamente, ya que realizan un seguimiento de las tareas habilitadas y el correspondiente indicador de finalización. Las principales secuencias utilizadas para esta práctica son:

-Secuencia ESC: la secuencia de escape se invoca cada vez que los terminales reciben el carácter de escape (# debido a las limitaciones del terminal Bluetooth). Desglosa la tarea actual y recuerda el menú para permitir la entrada de nuevos usuarios.

- Secuencia de menú: secuencia de menú es la que llama a las otras secuencias correspondientes. Una vez que terminan, regresan a la secuencia de menú ya que es la base para la aplicación.

- Secuencia de memoria de escritura: Espera una dirección y los datos para colocar en la memoria externa.

- Leer secuencia de memoria: espera una dirección y una longitud de datos para devolver los datos guardados en la memoria externa.

- Tarea I2C: tarea que recibe un identificador de transferencia que indica los parámetros para operar los componentes de hardware I2C.

- Tarea de recepción: recibe la entrada del usuario al sistema, la pone en cola o se rompe si recibe un carácter de retorno (fin del mensaje) o un carácter de escape (que invoca la secuencia de escape).

- Tarea de transmisión: vacía su cola y la envía al terminal correspondiente.

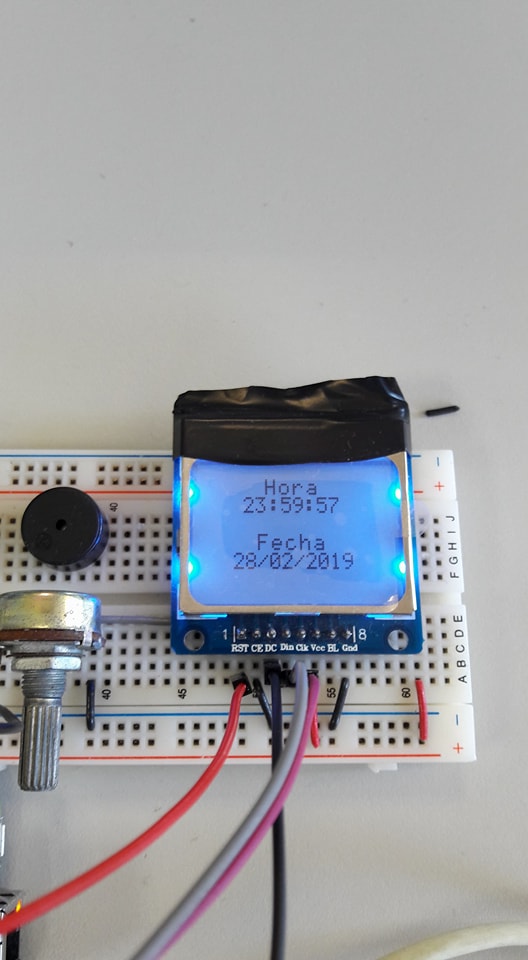
- Tarea para establecer fecha y hora: consta de 2 tareas en independientes en las cuales se ingresa de manera manual a través de la terminal la hora y fecha deseada.

- Tarea para leer hora y fecha: al ingresar en estas tareas se despliega en la terminal la hora y fecha actuales.

- Tarea para cambio de formato: actualiza el formato ya sea a 24 o 12 horas.

- Tarea para la ECO en la LCD.

Y por último una funcion en la cual se estará llamando constantemente para la lectura del RTC, convertir los datos y mostrarlos en una pantalla LCD como se ve a continuación:



## Conclusiones:

Esta práctica nos dio una idea de la implementación de los protocolos de comunicación y su interacción para permitir que un usuario obtenga y establezca información de diferentes periféricos. El mayor desafío fue comprender las funcionalidades básicas de la plataforma FreeRTOS y cómo usar su estructura para colocar las secuencias correspondientes para que funcionen correctamente y sin problemas entre dos terminales.

***Daniel***: Personalmente, me pareció desafiante comprender realmente cuáles fueron los problemas que surgieron durante el desarrollo de la práctica. Algunos de ellos no fueron capturados explícitamente por el IDE, por lo que se requiere un enfoque cuidadoso para ver dónde se originó el problema

***Gustavo:*** Durante el desarrollo del proyecto, surgieron increíblemente grandes cantidades de problemas y motivos que hacían que me trabara perdiendo así gran tiempo, pero con mucha dedicación y ánimos logre salir adelante y concluir con esta práctica. En especial en la implementación de las tareas en comunicación con el RTC, ya que, con un pequeño error, el sistema se rompía, y otro gran problema fue el tipo de integrado de RTC que se utilizó, ya que este no guardaba como tal un año y se pasaba ciertas banderas de conteo.

# Referencias

Garcia Alvarez, J. A. (Septiembre de 2015). *Asi funciona*. Obtenido de http://www.asifunciona.com/electrotecnia/ke\_resistencia/ke\_resistencia\_1.htm

Wikipedia. (17 de Marzo de 2017). *Wikipedia*. Obtenido de https://es.wikipedia.org/wiki/Reloj\_en\_tiempo\_real

Wikipedia. (20 de Octubre de 2017). *Wikipedia*. Obtenido de https://es.wikipedia.org/wiki/EEPROM