

Revisiones	Fecha	Comentarios
0	11/08/06	

Una vez más, nuestra conocida “pizarra remota” vuelve a evolucionar. Esta vez, agregamos la posibilidad de escribirla mediante mensajes de texto (SMS) empleando un módulo GSM SIM200. Modificamos además el código de la interfaz serie para poder, mediante un módulo Bluetooth, conectarnos desde (por ejemplo) una PDA. La descripción y desarrollo del código para Bluetooth se encuentra en el Comentario Técnico CTC-036. La descripción de cómo enviar y recibir mensajes de texto con un SIM200 se encuentra en el Comentario Técnico CTC-037. La Nota de Aplicación CAN-043 explica cómo recibir emails y hace una descripción de la pizarra en sí, brindando además numerosos punteros a notas de aplicación anteriores que cubren la recepción de mensajes por HTTP, el hardware de display, y conceptos más básicos de Rabbit.

En síntesis: el módulo Rabbit funciona como un servidor web y con cualquier navegador podemos escribir un simple mensaje en la pizarra; además, periódicamente revisa el correo en un servidor y si ve un mensaje muestra la primera línea en la pizarra. Además, periódicamente interroga al módulo SIM200 para ver si hay mensajes de texto SMS y si los hay muestra la primera línea en la pizarra. Por último, controla al módulo Bluetooth para que al recibir una conexión, permita al usuario ingresar una línea de texto a ser mostrada en la pizarra.

Más allá de su posible utilidad en la vida real, esta aplicación nos permite ejemplificar el uso de las capacidades de TCP/IP de Dynamic C, conectividad con módulos GSM, y módulos Bluetooth. No se darán demasiados detalles acerca del display y su software de control, dado que este tema se ha desarrollado ya en varias notas de aplicación. El lector interesado puede remitirse a dichas notas de aplicación para mayor información.

Descripción del funcionamiento

Nuestra pizarra será un display gráfico de 128x64, que mostrará texto en 8 filas de 20 caracteres. La fila superior mostrará la fecha y hora, y los mensajes ingresarán por la fila inferior, desplazando el texto anterior hacia arriba.

Para ingresar un mensaje, el usuario puede:

- ◆ Conectarse con su navegador preferido a nuestra dirección IP, donde verá una planilla en la que ingresará el texto a mostrar. El usuario puede ver además la fecha y hora y la cantidad de mensajes enviados. Esto se realiza mediante “server side includes” (SHTML). La planilla es un HTML FORM, en modo POST. Al remitirla (submit), el usuario hace que el módulo Rabbit ejecute una función (CGI), cuyos parámetros son provistos por el navegador y su resultado es la actualización del display; al mismo tiempo que se le mostrará en el navegador una segunda pantalla, confirmando la acción.
- ◆ Enviar un email a una dirección de mail válida. El módulo Rabbit verificará periódicamente la existencia de correo en esa dirección, y de encontrarlo lo traerá e imprimirá los primeros 20 caracteres de la primera línea del mensaje.
- ◆ Enviar un mensaje de texto SMS al número correspondiente a la tarjeta SIM utilizada en el módulo SIM200. El módulo Rabbit verificará periódicamente la existencia de mensajes, y de encontrarlos imprimirá los primeros 20 caracteres de la primera línea del mensaje.
- ◆ Mediante algún programa que permita utilizar el perfil SPP de Bluetooth¹, descubrir al módulo de KCWirefree y conectarse al mismo; observar el prompt e ingresar el mensaje. El usuario puede ver además la fecha y hora y la cantidad de mensajes enviados.

En todos los casos, la información de fecha y hora se toma directamente del timer provisto por Dynamic C, se asume que el usuario ya hizo la “puesta en hora” del sistema mediante alguno de los ejemplos provistos por

¹ Esto se explica más en detalle en CTC-036

Dynamic C. No es necesario reajustar fecha y hora a menos que se apague el módulo y no se disponga de battery backup.

Software

Desarrollaremos a continuación el software de la pizarra remota. Dado que, como dijéramos, se trata de una nueva encarnación de un viejo proyecto, comentaremos sólo aquellas partes que revistan alguna novedad o modificación. El lector interesado puede remitirse a la CAN-004, que contiene la descripción del código original, y CAN-043 que realiza modificaciones para otros displays y recibir emails.

El siguiente costate es la tarea que atiende al módulo KC-111 de KCWirefree, acorde a lo analizado en CTC-036. Detectamos conexión para enviar el prompt y desconexión para descartar los mensajes de control del módulo. Una vez establecida la conexión tomamos los datos que éste nos entrega y los mostramos mediante la función putmsg() que hemos desarrollado en CAN-043:

```

costate {
  if(!string_pos)
    serErdFlush();
  do {
    wfd string_pos=cof_serEgets(sentence,sizeof(sentence)-1,IDLE_TMOU);
  } while(string_pos == 0);
  if(strstr(sentence,"ConnectionUp")) // detecta conexión
    wfd putprompt(replydata); // envía prompt
  else {
    if((!strstr(sentence,"AT-ZV")) && (!strstr(sentence,"###NO CARRIER")))
      && strlen(sentence)){ // ignora mensajes de control del módulo
      sentence[MAX_SENTENCE]=0; // prepara mensaje para impresión
      sprintf(replydata,"\r\nOK\r\nMSG%d: %s\r\n",msg,sentence);
      wfd cof_serEwrite(replydata, strlen(replydata)); // confirma
      putmsg(sentence); // muestra mensaje en display LCD
    }
  }
}

```

El siguiente costate es la tarea que atiende al módulo SIM200, acorde a lo analizado en CTC-037. Interrogamos al módulo por la presencia de mensajes SMS y si los hay los extraemos de la respuesta

```

costate{
  wfd cof_serDputs("AT+CMGF=1\r\n"); // modo de trabajo: texto
  do {
    wfd count=cof_serDread(buffer,sizeof(buffer)-1,300); // ignora respuesta
  } while(!count);
  wfd cof_serDputs("AT+CNMI=2,0,0,0,0\r\n"); // no informes si no te lo pido
  do {
    wfd count=cof_serDread(buffer,sizeof(buffer)-1,300); // ignora respuesta
  } while(!count);
  wfd cof_serDputs("AT+CMGL=\"REC UNREAD\"\r\n"); // quiero los mensajes no leídos
  do {
    wfd count=cof_serDread(buffer,sizeof(buffer)-1,300);
  } while(!count);
  buffer[count]=0;
  ptr=strstr(buffer,"\n+CMGL:"); // busca inicio de respuesta
  while(ptr){ // repite mientras haya mensajes
    count=atoi(ptr+=7); // obtiene número de mensaje
    if(count){
      if(strptr=strchr(ptr,'\n')){ // busca fin de línea
        if(ptr=strstr(strptr+1,"\n+CMGL:")) // línea siguiente
          *ptr=0; // marca fin
        else if(auxptr=strstr(strptr,"\nOK")) // era la última línea
          *auxptr=0; // marca fin
        strptr[MAX_SENTENCE]=0; // prepara para impresión
        putmsg(strptr); // muestra en LCD
      }
      sprintf(buffer,"AT+CMGD=%d\r\n",count); // borra mensaje
      wfd cof_serDputs(buffer);
      do{ // ignora respuesta
        wfd count=cof_serDread(buffer,sizeof(buffer)-1,300);
      } while(!count);
    }
  }
}

```

CAN-053, Pizarra remota con SMS y Bluetooth con SIM200 y KC-111

```
    }  
    waitfor(DelaySec(30)); // espera 30 segundos  
}
```

Resumido y genérico, el programa principal, donde se integra todo, sería algo más o menos así:

```
main()  
{  
int i;  
unsigned long t;  
int input_char, string_pos;  
char replydata[128], sentence[MAX_SENTENCE+2];  
  
    msg=1;  
    string_pos = 0;  
    FORMSpec[0].name = "texto";  
  
    pop3_init(mailmsg);  
    brdInit();  
    sock_init();  
    http_init();  
    tcp_reserveport(80);  
  
    serDopen(115200);  
    serEopen(115200);  
  
    loopinit();  
  
    LCD_init();  
    LCD_clear();  
  
    while(1){  
        http_handler(); // TCP/IP + página web  
        loophead();  
        costate{  
            // Bluetooth, recién visto  
        }  
        costate{  
            // SMS, recién visto  
        }  
        costate{  
            // obtención de email, CAN-043  
        }  
        costate {  
            waitfor(DelaySec(1)); // espera 1 segundo  
            get_datetime(); // actualiza fecha y hora  
            LCD_printat(0,0,date); // lo muestra  
            LCD_printat(0,12,time);  
        }  
    }  
}
```