

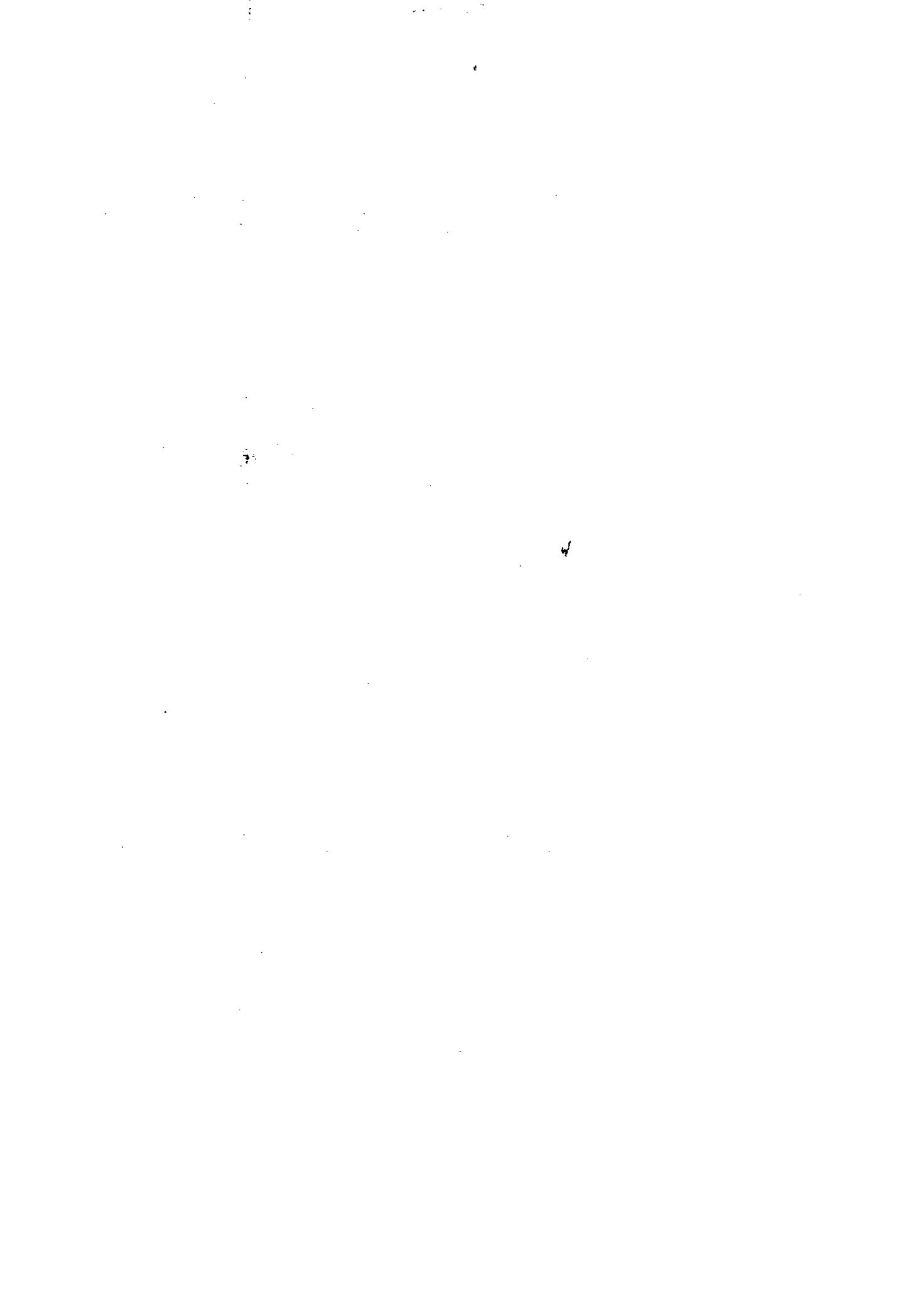
**ESCUELA POLITECNICA NACIONAL  
FACULTAD DE INGENIERIA ELECTRICA**

**DISEÑO Y CONSTRUCCION DE UNA TARJETA  
PARA MEDIR Y ALMACENAR VELOCIDAD DEL  
TROLEBUS – QUITO CON SOFTWARE PARA  
DESCARGAR Y ANALIZAR DATOS EN UN PC**

**ERNESTO V MONTALVO  
WALTER E NUÑEZ**

**TESIS PREVIA A LA OBTENCION DEL TITULO  
DE INGENIERO EN ELECTRONICA Y CONTROL**

**QUITO MARZO DE 1998**



***MANUAL DEL  
USUARIO.***



## INDICE

Introducción.....	1
Hardware.....	1
Software.....	2
Ventana principal .....	2
Ventana visualización de velocidad .....	4
Ventana recibir datos.....	5
Ventana reloj.....	5
Ventana datos de operación.....	6
Ventana abrir .....	7
Ventana guardar como .....	7
Ventana gráfico .....	8
Ventana descarrilamientos .....	9



## **INTRODUCCION.**

El presente folleto le ayudara en el manejo del módulo construido en la presente tesis, así como en el manejo del software de administración implementado.

## **HARDWARE.**

El hardware construido para operar necesita alimentación de -12V, tierra y +12V, la alimentación se la realiza con el conector de 3 polos, el cable color rojo corresponde a +12V, el cable color amarillo - verde corresponde a tierra y el cable color negro es el potencial -12V.

El otro conector del modulo (6 polos), corresponde a las señales de entrada, el cable color plomo corresponde a la entrada de tracción eléctrica, el cable color rosado es la entrada de descarrilamiento, el cable color azul corresponde a la señal de entrada de velocidad (onda cuadrada de frecuencia variable) y el cable color blanco es la entrada de la señal de tracción diesel.

El ancho de banda permitido de entrada de la señal de velocidad al módulo esta entre 0 Hz y 250 Hz, que es el valor máximo de frecuencia generada por los sensores de revoluciones en los ejes de transmisión del trolebús.

Los niveles de voltaje de entrada pueden ser desde 5V hasta 24V de corriente continua.

La conexión entre el módulo y el PC se lo hace por interface serial RS232, en el conector macho DB9 del módulo.

El cable de comunicación serial debe tener al menos 3 cables, y 2 conectores hembras DB9, su conexión debe ser entre los pines 2 y 3 cruzados, y el pin5 del un conector con el otro.

Una vez alimentado el módulo podrá observarse visualmente su correcto funcionamiento por cuanto hay un led indicador que se encuentra encendiéndose y apagándose secuencialmente a la frecuencia de 1Hz, por esto es apreciable a simple vista.

Si se ingresa un 1 lógico en cualquiera de las entradas de tipo de tracción se podrá observar que el led indicador de tracción eléctrica o diesel se enciende.

## **SOFTWARE.**

El programa implementado para la administración de datos, se ha construido en Visual BASIC, el programa se ha diseñado de tal forma que facilite al usuario su uso ya que cada ventana contiene botones de comando y con la acción del movimiento del ratón sobre cada botón se despliegan mensajes indicando la acción que cada uno de estos realiza.

El programa es un archivo ejecutable con nombre TROLE.EXE, el programa corre en ambiente WINDOWS, ya sea la versión 3.1 o superior, se recomienda copiarlo bajo un directorio propio, además se debe crear el directorio C:\DAT; ya que bajo este directorio se generarán los archivos de datos adquiridos por la tarjeta y enviados hacia el PC.

Los archivos generados por este programa se van guardando por defecto bajo este directorio.

A continuación se realizara una explicación breve de las pantallas principales que despliega el programa, y la función que cada comando ejecuta.

## **VENTANA PRINCIPAL.**

Cuando el programa se ejecuta despliega la pantalla principal que se muestra en la Figura 1





Figura 1  
Pantalla principal.

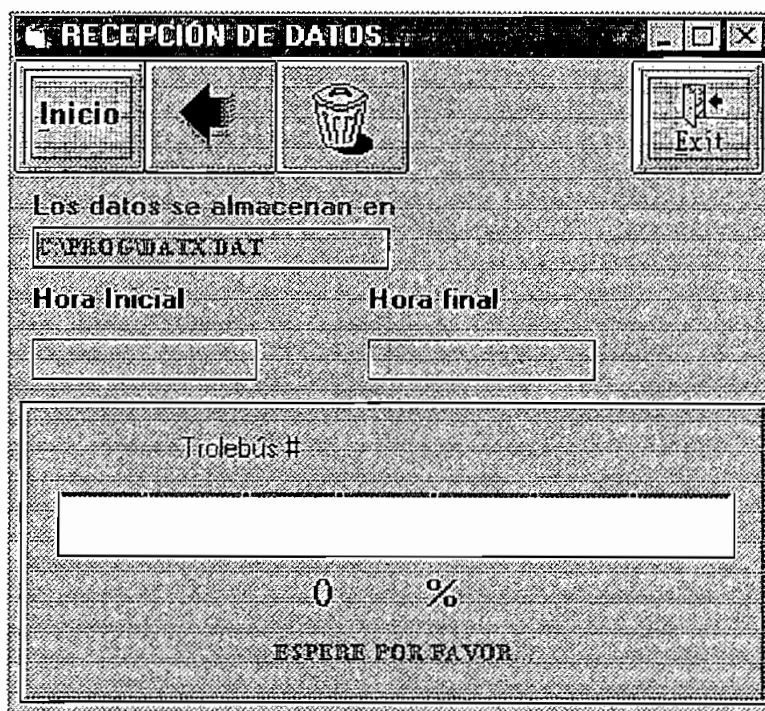
La pantalla principal posee tres menús (Figura 2).



Figura 2  
Despliegue de los menús de la pantalla principal.

## VENTANA RECIBIR DATOS.

El submenú Recibir Datos despliega una ventana (Figura 4), en la cuál por medio de botones de comando se ordena a la tarjeta descargar los datos adquiridos, a medida que los datos se van descargando la ventana visualiza el avance de la descarga de datos desde 0 hasta el 100%. La ventana también muestra la hora inicial y final de la descarga de datos hacia el PC.



*Figura 4*  
*Ventana recibir datos.*

## VENTANA RELOJ.

El submenú Igualar Reloj despliega una ventana (Figura 5) en cuál consta la fecha y hora actual (en el PC) la fecha y hora que tiene la tarjeta y casillas en las cuales se ponen los valores de horas/minutos/segundos y día/mes/año para cambiarlos a la tarjeta utilizando botones de comando, la ventana visualiza mensajes de la recepción en el módulo de los datos enviados.



*Figura 5*  
*Ventana reloj.*

## VENTANA DATOS DE OPERACIÓN.

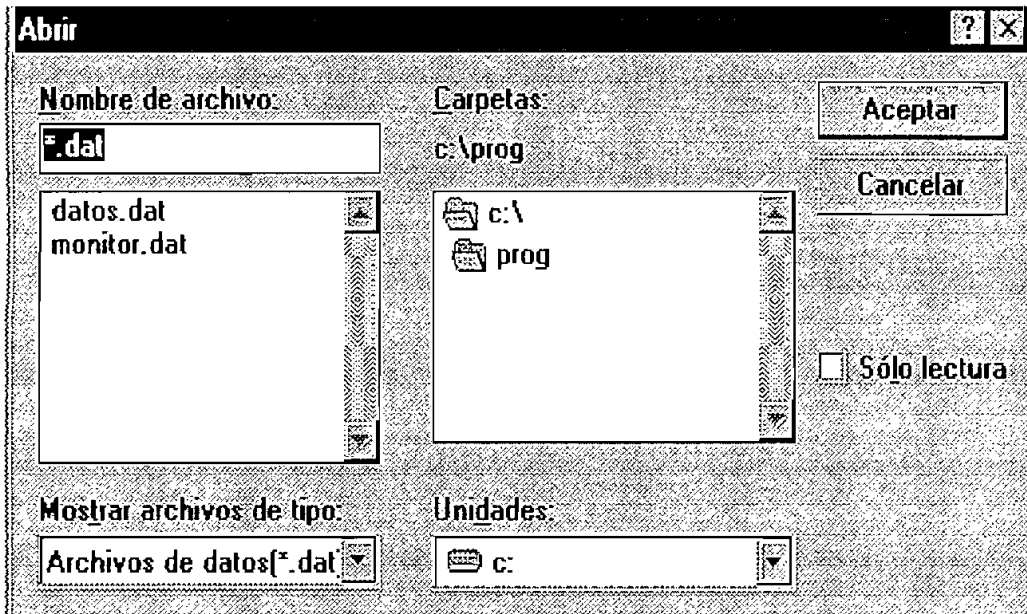
El submenú Datos de Operación despliega una ventana (Figura 6) en la cual se visualiza el número de trolebús, y el tiempo de muestreo, así como casilleros en los cuales se ponen el nuevo número y tiempo de muestreo si se desea cambiarlos.

*Figura 6*  
*Ventana Datos de operación*



## VENTANA ABRIR.

El submenú Abrir muestra una caja de dialogo (Figura 7) propia de Windows la misma que ayuda a buscar la localización del archivo de datos que se desea abrir, y luego lo despliega en una caja de texto.



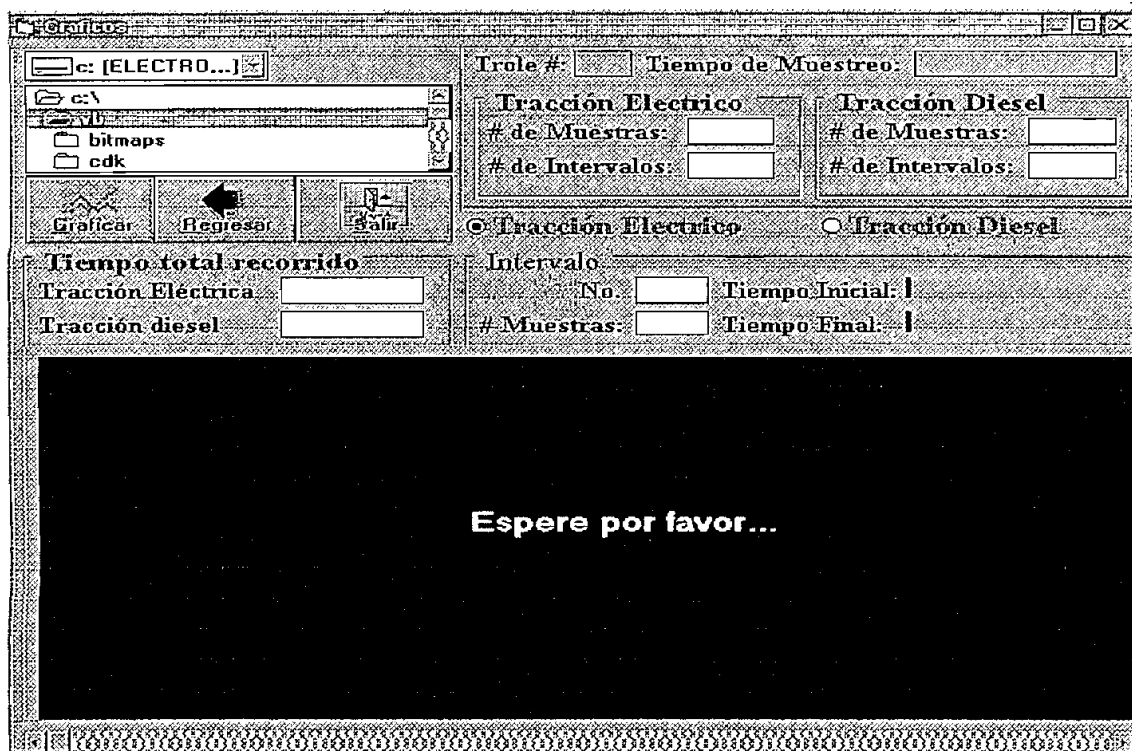
*Figura 7*  
*Caja de dialogo Abrir, archivos de datos.*

El submenú Cerrar cierra el archivo de datos que se encuentre abierto.

## VENTANA GUARDAR COMO

El submenú Guardar como despliega una caja de dialogo (Figura 8) propia de Windows que permite guardar el archivo de datos abierto con otro nombre y ubicación dentro del disco duro, u otro drive disponible en el PC.

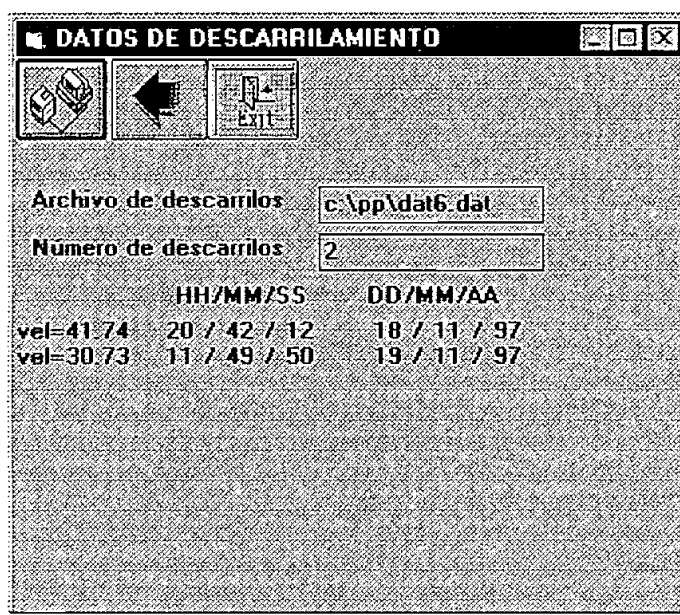




*Figura 9  
Ventana gráficos.*

**VENTANA DESCARRILAMIENTOS**

El sub menú descarrilo, despliega una ventana (Figura 10) en la cual se despliega los datos de descarrilamiento, los datos constan del dato de velocidad a la que se descarrilo y la fecha y hora en que ocurrió.



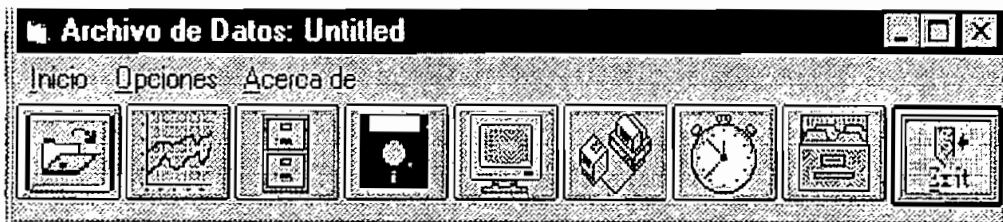
*Figura 10  
Ventana descarrilamientos.*





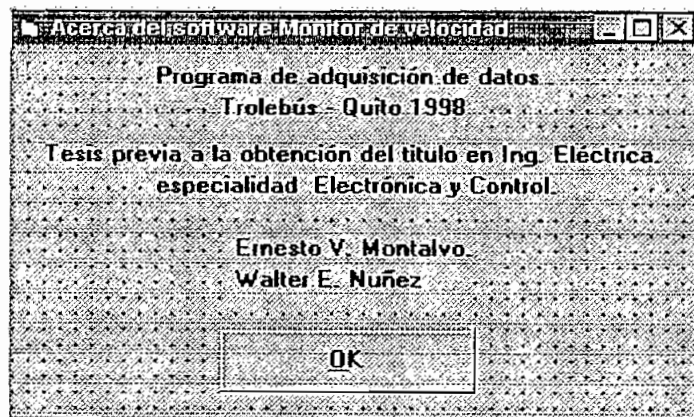
El submenú Salir finaliza todas las tareas que se estén realizando y sale del programa de administración de datos.

El menú Opciones visualiza una caja de dialogo (Figura 2) en la cuál se encuentra el submenú Ver barra de herramientas, este submenú visualiza una barra con botones de comando (Figura 11) que realizan las mismas acciones de los submenus del menú Inicio.



*Figura 11*  
*Visualización de la barra de herramientas.*

El menú Acerca de despliega una ventana (Figura 12) que contiene el submenu Acerca de, este submenú visualiza una forma que muestra información sobre los autores y fecha de implementación del programa.



*Figura 12*  
*Ventana acerca de.*







**ESCUELA POLITECNICA NACIONAL  
FACULTAD DE INGENIERIA ELECTRICA**

**DISEÑO Y CONSTRUCCION DE UNA TARJETA  
PARA MEDIR Y ALMACENAR VELOCIDAD DEL  
TROLEBUS – QUITO CON SOFTWARE PARA  
DESCARGAR Y ANALIZAR DATOS EN UN PC**

**ERNESTO V MONTALVO  
WALTER E NUÑEZ**

**TESIS PREVIA A LA OBTENCION DEL TITULO  
DE INGENIERO EN ELECTRONICA Y CONTROL**

**QUITO, MARZO 1998**



# ***ANEXO***

## ***3A***

***PROGRAMA DE LECTURA  
ALMACENAMIENTO Y  
TRANSMISIÓN DE DATOS EN  
LA TARJETA DE ADQUISICIÓN.***





```
;PROGRAMA DE ADQUISICIÒN DE DATOS DE VELOCIDAD Y DESCARRILO
;EN EL TROLEBÙS - QUITO.
;VINICIO MONTALVO
;WALTER NUÑEZ
;ENERO 1998
```

```
ORG      0000H          ;DIRECCIÒN DE INICIO DEL PROGRAMA
LJMP     INICIO        ;SALTO A ETIQUETA INICIO
ORG      0003H          ;VECTORIZACIÒN DE INTERRUPTIÒN CERO
LJMP     INTERO        ;ETIQUETA DE INTERRUPTIÒN CERO
ORG      30H           ;DIRECCIÒN DE LA ETIQUETA INICIO
```

```
;EN ESTA AREA DE PROGRAMA SE INICIALIZA TODOS LOS MODOS DE TRABAJO
;DEL MICROPROCESADOR, ASI COMO EL RELOJ Y DIRECCIONES DATOS EN
;LA MEMORIA RAM EXTERNA
```

INICIO:

```
MOV      SP,#32H        ;INICIO DEL PUNTERO DE DATOS
MOV      TMOD,#00100010B ;TIMER 0 Y 1 EN MODO 2 (MODO AUTORECARGA)
MOV      SCON,#01010000B
MOV      PCON,#00H
MOV      TH1,#0F8H     ;VALOR PARA COMUNICACION SERIAL A 2400 BAUDIOS
MOV      TL1,#0F8H     ;VALOR PARA LA RECARGA DEL TIMER 1
CLR      RI            ;LIMPIA LA BANDERA DE RECEPCIÒN SERIAL
CLR      TI            ;LIMPIA LA BANDERA DE TRANSMISIÒN SERIAL
CLR      TFO          ;LIMPIA LA BANDERA DE DESBORDAMIENTO DEL TIMER
0
SETB     TR1          ;HABILITA LA INTERRUPTIÒN DEL TIMER 1
CLR      P1.4         ;ENCIENDE LED DE CONDICIÒN NORMAL
LCALL    ENCERAR      ;BORRA LA MEMORIA RAM EXTERNA
LCALL    DESHAB       ;DESHABILITA TODAS LA INTERRUPTIÒNES
LCALL    RELOJO       ;INICIALIZA EL RELOJ
LCALL    DIRECO       ;INICIALIZA LAS DIRECCIONES DE MEMORIA RAM
EXTERNA
SETB     P1.6         ;APAGA LED DE TRACCIÒN DIESEL
SETB     P1.7         ;APAGA LED DE TRACCIÒN ELECTRICO
SETB     P1.5         ;APAGA LED INDICADOR DE DESCARRILAMIENTO
LCALL    LEDS         ;LLAMA A SUBROUTINA DE CONTROL DE ENCENDIDO DE
LEDS
LCALL    DIRECO       ;INICIALIZA LAS DIRECCIONES DEL MAPA DE
MEMORIA
```

```
;PROGRAMA PRINCIPAL, ESTA COMPUESTO DE LLAMADOS A SUBROUTINAS
;QUE SE REPITEN CICLICAMENTE
```

PRINCIPAL:

```
LCALL    BANCO0
MOV      DPTR,#06H     ;DIRECCIÒN DE RAM EXTERNA CONTIENE DATO TM
MOVX     A,@DPTR      ;LECTURA DEL TM
MOV      R7,A         ;TIEMPO DE MUESTREO EN SEGUNDOS
PRIN1:
LCALL    SEGUNDO      ;BASE DE TIEMPO DE 1 SEGUNDO
LCALL    GHORA        ;LLAMA SUBROUTINA QUE VERIFICA CAMBIOS DE
TRACCIÒN
LCALL    LEDS         ;LLAMA A SUBROUTINA DE CONTROL DE ENCENDIDO DE
LEDS
LCALL    TIEMPO       ;ACTUALIZA EL RELOJ EN TIEMPO REAL
LCALL    SERIAL       ;VERIFICA SI EXISTE COMUNICACIÒN
LCALL    BANCO0       ;SELECCIONA BANCO DE RAGISTROS CERO
DJNZ     R7,PRIN1    ;CUENTA N SEGUNDOS
```



```

LCALL  GUARDAR          ;GUARDA DATOS EN MEMORIA RAM
LJMP   PRINCIPAL       ;FIN PROGRAMA PRINCIPAL, SE REPITE

;SUBROUTINA QUE VERIFICA CAMBIOS DE TRACCIÀN Y GUARDA LA HORA
;FECHA Y EL NUMERO DE MUESTRAS EN CADA INTERVALO

GHORA:
LCALL  BANCO          ;SELECCIONA EL BANCO DE REGISTROS CERO
MOV    DPTR,#80       ;APUNTA DIRECCIÀN DEL CODIGO DE TRACCIÀN
MOVX   A,@DPTR        ;LEE EL CODIGO DE TRACCIÀN
MOV    R6,A           ;GUARDA EL CODIGO LEIDO EN EL REGISTRO 6
LCALL  DETECCION      ;LLAMA SUBROUTINA QUE DETECTA TIPO DE TRACCIÀN
ACTUAL
MOV    DPTR,#80       ;APUNTA DIRECCIÀN DEL CODIGO DE TRACCIÀN
MOVX   A,@DPTR        ;LEE EL CODIGO DE TRACCIÀN

CJNE   R6,#204,CEROE  ;VERIFICA CAMBIO DE TRACCIÀN DE ELECTRICO
CJNE   A,#200,CEROE   ;A CERO
LCALL  ELECTRIC011    ;SI HAY CAMBIO GUARDA LA FECHA Y HORA DE
CAMBIO
LJMP   GSALE

CEROE:
CJNE   R6,#200,ETIQ0  ;VERIFICA SI HAY CAMBIO DE CERO A ELECTRICO
CJNE   A,#204,ETIQ0  ;SI HAY CAMBIO GUARDA LA FECHA Y HORA DE
LCALL  ELECTRIC011
CAMBIO
LJMP   GSALE

ETIQ0:
CJNE   R6,#200,ETIQ1  ;VERIFICA CAMBIO DE CERO A DIESEL
CJNE   A,#209,ETIQ1  ;SI HAY CAMBIO GUARDA FECHHA Y HORA
LCALL  DIESEL11
LJMP   GSALE

ETIQ1:
CJNE   R6,#209,ETIQ2  ;VERIFICA CAMBIO DE DIESEL A CERO
CJNE   A,#200,ETIQ2  ;SI HAY CAMBIO GUARDA FECHHA Y HORA
LCALL  DIESEL11
LJMP   GSALE

ETIQ2:
CJNE   R6,#209,ETIQ3  ;VERIFICA SI HAY CAMBIO DE DIESEL A EELTRICO
CJNE   A,#204,ETIQ3  ;SI HAY CAMBIO GUARDA LA FECHA Y HORA DE
LCALL  ELECTRIC011
CAMBIO
LCALL  DIESEL11       ;SI HAY CAMBIO GUARDA FECHHA Y HORA
LJMP   GSALE

ETIQ3:
CJNE   R6,#204,GSALE  ;VERIFICA SI HAY CAMBIO DE ELECTRICO A DIESEL
CJNE   A,#209,GSALE  ;SI HAY CAMBIO GUARDA LA FECHA Y HORA DE
LCALL  ELECTRIC011
CAMBIO
LCALL  DIESEL11       ;SI HAY CAMBIO GUARDA FECHHA Y HORA
GSALE:
RET

```



```

;SUBROUTINA QUE GUARDA LA FECHA, HORA Y NUMERO DE MUESTRA
;DE TRACCIÒN DIESEL
DIESEL11:
LCALL   BANCO0           ;SELECCIONA BANCO DE REGISTROS CERO
MOV     DPTR,#36         ;APUNTA A DIRECCIÒN DE VALOR ACTUAL DE DPL
DIESEL
MOVX    A,@DPTR         ;LEE VALOR DPL
MOV     R4,A             ;GUARDA VALOR DE DPL EN R4
INC     DPTR            ;APUNTA DIRECCIÒN DE DPH DIESEL
MOVX    A,@DPTR         ;GUARDA VALOR DE DPH DIESEL
MOV     R5,A             ;GUARDA VALOR DE DPH EN R5
MOV     DPTR,#00H       ;APUNTA DIRECCIÒN QUE CONTIENE SEGUNDOS
MOVX    A,@DPTR         ;LEE EL VALOR DE SEGUNDOS
MOV     DPL,R4          ;RECUPERA VALOR DPL
MOV     DPH,R5          ;RECUPERA VALOR DPH
MOVX    @DPTR,A         ;GUARDA VALOR SEGUNDOS
INC     DPTR            ;INCREMENTA EL DPTR
MOV     R4,DPL          ;ACTUALIZA DPL EN R4
MOV     R5,DPH          ;ACTUALIZA DPH EN R5
LCALL   COMPARAD        ;LLAMA SUBROUTINA QUE VERIFICA SI SOBREPASA
AREA DE
MOV     DPTR,#01H       ;DATOS DESTINADA A TRACCIÒN DIESEL
MOVX    A,@DPTR         ;LEE EL VALOR DE MINUTOS ACTUALES
MOV     DPL,R4          ;RECUPERA VALOR DPL
MOV     DPH,R5          ;RECUPERA VALOR DPH
MOVX    @DPTR,A         ;GUARDA VALOR DE MINUTOS
INC     DPTR            ;INCREMENTA EL DPTR
MOV     R4,DPL          ;ACTUALIZA DPL EN R4
MOV     R5,DPH          ;ACTUALIZA DPH EN R5
LCALL   COMPARAD        ;LLAMA SUBROUTINA QUE VERIFICA SI SOBREPASA
AREA DE DATOS
MOV     DPTR,#02H       ;APUNTA DIRECCIÒN DE HORAS
MOVX    A,@DPTR         ;LEE EL VALOR DE HORAS
MOV     DPL,R4          ;RECUPERA VALOR DPL
MOV     DPH,R5          ;RECUPERA VALOR DPH
MOVX    @DPTR,A         ;GUARDA VALOR DE HORAS
INC     DPTR            ;INCREMENTA EL DPTR
MOV     R4,DPL          ;ACTUALIZA DPL EN R4
MOV     R5,DPH          ;ACTUALIZA DPH EN R5
LCALL   COMPARAD        ;LLAMA SUBROUTINA QUE VERIFICA SI SOBREPASA
AREA DE DATOS
MOV     DPTR,#03H       ;APUNTA DIRECCIÒN DE DIA
MOVX    A,@DPTR         ;LEE EL VALOR DIA
MOV     DPL,R4          ;RECUPERA VALOR DPL
MOV     DPH,R5          ;RECUPERA VALOR DPH
MOVX    @DPTR,A         ;GUARDA EL VALOR DE DIA
INC     DPTR            ;INCREMENTA EL DPTR
MOV     R4,DPL          ;ACTUALIZA DPL EN R4
MOV     R5,DPH          ;ACTUALIZA DPH EN R5
LCALL   COMPARAD        ;LLAMA SUBROUTINA QUE VERIFICA SI SOBREPASA
AREA DE DATOS
MOV     DPTR,#04H       ;APUNTA DIRECCIÒN DEL MES
MOVX    A,@DPTR         ;LEE EL VALOR DEL MES
MOV     DPL,R4          ;RECUPERA VALOR DPL
MOV     DPH,R5          ;RECUPERA VALOR DPH
MOVX    @DPTR,A         ;GUARDA EL VALOR DEL MES
INC     DPTR            ;INCREMENTA EL DPTR
MOV     R4,DPL          ;ACTUALIZA DPL EN R4
MOV     R5,DPH          ;ACTUALIZA DPH EN R5

```



```

LCALL  COMPARAD      ;LLAMA SUBROUTINA QUE VERIFICA SI SOBREPASA
AREA DE DATOS
MOV    DPTR,#05H    ;APUNTA DIRECCIÒN DEL AÑO
MOVX   A,@DPTR      ;LEE EL VALOR DEL AÑO
MOV    DPL,R4       ;RECUPERA VALOR DPL
MOV    DPH,R5       ;RECUPERA VALOR DPH
MOVX   @DPTR,A      ;GUARDA EL VALOR DEL AÑO
INC    DPTR         ;INCREMENTA EL DPTR
MOV    R4,DPL       ;ACTUALIZA DPL EN R4
MOV    R5,DPH       ;ACTUALIZA DPH EN R5
LCALL  COMPARAD      ;LLAMA SUBROUTINA QUE VERIFICA SI SOBREPASA
AREA DE DATOS
LCALL  BANCO2        ;LLAMA AL BANCO DE REGISTROS 2
MOV    A,R4         ;LEE EL NUEMRO DE MUESTRAS (UNIDADES)
MOV    R4,#00       ;INICIALIZA EL REGISTRO
LCALL  BANCO0        ;LLAMA BANCO DE REGISTROS CERO
MOV    DPL,R4       ;RECUPERA VALOR DPL
MOV    DPH,R5       ;RECUPERA VALOR DPH
MOVX   @DPTR,A      ;GUARDA EL NUMERO DE MUESTRAS
INC    DPTR         ;INCREMENTA EL DPTR
MOV    R4,DPL       ;ACTUALIZA DPL EN R4
MOV    R5,DPH       ;ACTUALIZA DPH EN R5
LCALL  COMPARAD      ;LLAMA SUBROUTINA QUE VERIFICA SI SOBREPASA
AREA DE DATOS
LCALL  BANCO2        ;LLAMA AL BANCO DE REGISTROS 2
MOV    A,R5         ;LEE EL NUMERO DE MUESTRAS (MULTIPLO DE 250)
MOV    R5,#00       ;INICIALIZA REGISTRO DECENAS
LCALL  BANCO0        ;LLAMA BANCO DE REGISTROS CERO
MOV    DPL,R4       ;RECUPERA VALOR DPL
MOV    DPH,R5       ;RECUPERA VALOR DPH
MOVX   @DPTR,A      ;GUARDA NUMERO DE MUESTRAS
INC    DPTR         ;INCREMENTA EL DPTR
MOV    R4,DPL       ;ACTUALIZA DPL EN R4
MOV    R5,DPH       ;ACTUALIZA DPH EN R5
LCALL  COMPARAD      ;LLAMA SUBROUTINA QUE VERIFICA SI SOBREPASA
AREA DE DATOS
MOV    DPTR,#36     ;APUNTA A DIRECCIÒN DE VALOR ACTUAL DE DPL
DIESEL
MOV    A,R4         ;LEE VALOR ACTUAL DEL DPL DIESEL
MOVX   @DPTR,A      ;GUARDA VALOR ACTUAL DPL DIESEL
INC    DPTR         ;APUNTA DIRECCIÒN DE VALOR DPH DIESEL
MOV    A,R5         ;LEE VALOR ACTUAL DPH DIESEL
MOVX   @DPTR,A      ;GUARDA VALOR ACTUAL DPH DIESEL
RET

;SUBROUTINA QUE REALIZA UNA REVISIÒN DE LOS VALORES DPL Y DPH
;CON LOS VALORES MAXIMOS DEL AREA DE DATOS DESTINADA A TRACCIÒN DIESEL

COMPARAD:
LCALL  BANCO0        ;LLAMA BANCO DE REGISTROS CERO
CJNE   R5,#0DH,SALD ;COMPARA EL VALOR DPH CON EL VALOR MAXIMO DEL
AREA
CJNE   R4,#0ACH,SALD ;COMPARA DPL CON EL VALOR MAXIMO DEL AREA
MOV    DPTR,#36     ;SI SOBREPASO APUNTA DIRECCIÒN DE DPL
MOV    A,#0DDH      ;INICIALIZA CON DPL BASE
MOV    R4,A         ;ACTUALIZA EL REGISTRO DESTINADO PARA DPL
MOVX   @DPTR,A      ;GUARDA EL VALOR BASE DE DPL
INC    DPTR         ;APUNTA DIRECCIÒN DE DPH
MOV    A,#05H       ;INICIALIZA CON DPH BASE
MOV    R5,A         ;ACTUALIZA EL REGISTRO DESTINADO PARA DPH

```





```

MOVX    @DPTR,A          ;GUARDA EL VALOR BASE DE DPH
SALD:
RET

```

```

;SUBROUTINA QUE GUARDA LA FECHA, HORA Y NUMERO DE MUESTRA
;DE TRACCIÀN ELECTRICO

```

```

ELECTRICO11:

```

```

LCALL   BANCO0          ;LLAMA BANCO DE REGISTROS CERO
MOV     DPTR,#32        ;APUNTA DIRECCIÀN DE DPL ELCTRICO
MOVX    A,@DPTR         ;LEE EL VALOR DE DPL ELECTRICO
MOV     R4,A           ;GUARDA VALOR DPL EN R4
INC     DPTR           ;APUNTA DIRECCIÀN DE DPH ELECTRICO
MOVX    A,@DPTR         ;LEE VALOR DE DPH ELECTRICO
MOV     R5,A           ;GUARDA DPH EN R5
MOV     DPTR,#00H      ;APUNTA DIRECCIÀN DE SEGUNDOS
MOVX    A,@DPTR         ;LEE EL VALOR DE SEGUNDOS
MOV     DPL,R4         ;RECUPERA VALOR DPL
MOV     DPH,R5         ;RECUPERA VALOR DPH
MOVX    @DPTR,A        ;GUARDA VALOR SEGUNDOS
INC     DPTR           ;INCREMENTA EL DPTR
MOV     R4,DPL         ;ACTUALIZA DPL EN R4
MOV     R5,DPH         ;ACTUALIZA DPH EN R5
LCALL   COMPAREAE      ;LLAMA A SUBROUTINA QUE VERIFICA SI PASO AREA
DE DATOS
MOV     DPTR,#01H      ;APUNTA DIRECCIÀN DE MINUTOS
MOVX    A,@DPTR         ;LEE EL VALOR DE MINUTOS
MOV     DPL,R4         ;RECUPERA VALOR DPL
MOV     DPH,R5         ;RECUPERA VALOR DPH
MOVX    @DPTR,A        ;GUARDA VALOR MINUTOS
INC     DPTR           ;INCREMENTA EL DPTR
MOV     R4,DPL         ;ACTUALIZA DPL EN R4
MOV     R5,DPH         ;ACTUALIZA DPH EN R5
LCALL   COMPAREAE      ;LLAMA A SUBROUTINA QUE VERIFICA SI PASO AREA
DE DATOS
MOV     DPTR,#02H      ;APUNTA DIRECCIÀN DE HORAS
MOVX    A,@DPTR         ;LEE EL VALOR DE HORAS
MOV     DPL,R4         ;RECUPERA VALOR DPL
MOV     DPH,R5         ;RECUPERA VALOR DPH
MOVX    @DPTR,A        ;GUARDA VALOR HORAS
INC     DPTR           ;INCREMENTA EL DPTR
MOV     R4,DPL         ;ACTUALIZA DPL EN R4
MOV     R5,DPH         ;ACTUALIZA DPH EN R5
LCALL   COMPAREAE      ;LLAMA A SUBROUTINA QUE VERIFICA SI PASO AREA
DE DATOS
MOV     DPTR,#03H      ;APUNTA DIRECCIÀN DE DIAS
MOVX    A,@DPTR         ;LEE EL VALOR DE DIAS
MOV     DPL,R4         ;RECUPERA VALOR DPL
MOV     DPH,R5         ;RECUPERA VALOR DPH
MOVX    @DPTR,A        ;GUARDA VALOR DIAS
INC     DPTR           ;INCREMENTA EL DPTR
MOV     R4,DPL         ;ACTUALIZA DPL EN R4
MOV     R5,DPH         ;ACTUALIZA DPH EN R5
LCALL   COMPAREAE      ;LLAMA A SUBROUTINA QUE VERIFICA SI PASO AREA
DE DATOS
MOV     DPTR,#04H      ;APUNTA DIRECCIÀN DE MES
MOVX    A,@DPTR         ;LEE EL VALOR DE MES
MOV     DPL,R4         ;RECUPERA VALOR DPL
MOV     DPH,R5         ;RECUPERA VALOR DPH
MOVX    @DPTR,A        ;GUARDA VALOR MES

```



```

INC      DPTR          ;INCREMENTA EL DPTR
MOV      R4,DPL       ;ACTUALIZA DPL EN R4
MOV      R5,DPH       ;ACTUALIZA DPH EN R5
LCALL   COMPAREAE    ;LLAMA A SUBROUTINA QUE VERIFICA SI PASO AREA
DE DATOS
MOV      DPTR,#05H    ;APUNTA DIRECCIÒN DE AÑO
MOVX    A,@DPTR      ;LEE EL VALOR DE AÑO
MOV      DPL,R4       ;RECUPERA VALOR DPL
MOV      DPH,R5       ;RECUPERA VALOR DPH
MOVX    @DPTR,A      ;GUARDA VALOR AÑO
INC      DPTR          ;INCREMENTA EL DPTR
MOV      R4,DPL       ;ACTUALIZA DPL EN R4
MOV      R5,DPH       ;ACTUALIZA DPH EN R5
LCALL   COMPAREAE    ;LLAMA A SUBROUTINA QUE VERIFICA SI PASO AREA
DE DATOS
LCALL   BANCO2       ;LLAMA BANCO DE REGISTROS 2
MOV      A,R0         ;LEE VALOR NUMERO DE MUESTRAS ELECTRICO
(UNIDADES)
MOV      R0,#00       ;INICIALIZ REGISTRO DE NUMERO DE MUESTRAS
LCALL   BANCO0       ;LLAMA BANCO DE REGISTROS CERO
MOV      DPL,R4       ;RECUPERA VALOR DE DPL
MOV      DPH,R5       ;RECUPERA VALOR DE DPH
MOVX    @DPTR,A      ;GUARDA VALOR DE NUMERO DE MUESTRAS
INC      DPTR          ;ACTUALIZA DPTR
MOV      R4,DPL       ;ACTUALIZA DPL EN R4
MOV      R5,DPH       ;ACTUALIZA DPH EN R5
LCALL   COMPAREAE    ;LLAMA A SUBROUTINA QUE VERIFICA SI PASO AREA
DE DATOS
LCALL   BANCO2       ;LLAMA BANCO DE REGISTROS 2
MOV      A,R1         ;LEE VALOR DE NUMERO DE MUESTRAS (MULTIPLoS DE
250)
MOV      R1,#00       ;INICIALIZA REGISTRO
LCALL   BANCO0       ;LLAMA BANCO DE REGISTROS CERO
MOV      DPL,R4       ;RECUPERA VALOR DE DPL
MOV      DPH,R5       ;RECUPERA VALOR DE DPH
MOVX    @DPTR,A      ;GUARDA VALOR DE NUMERO DE MUESTRAS
INC      DPTR          ;ACTUALIZA DPTR
MOV      R4,DPL       ;ACTUALIZA DPL EN R4
MOV      R5,DPH       ;ACTUALIZA DPH EN R5
LCALL   COMPAREAE    ;LLAMA A SUBROUTINA QUE VERIFICA SI PASO AREA
DE DATOS
MOV      DPTR,#32     ;APUNTA A DIRECCIÒN DE VALOR ACTUAL DE DPL
ELECTRICO
MOV      A,R4         ;LEE VALOR ACTUAL DEL DPL ELECTRICO
MOVX    @DPTR,A      ;GUARDA VALOR ACTUAL DPL ELECTRICO
INC      DPTR          ;APUNTA DIRECCIÒN DE VALOR DPH ELECTRICO
MOV      A,R5         ;LEE VALOR ACTUAL DPH ELECTRICO
MOVX    @DPTR,A      ;GUARDA VALOR ACTUAL DPH ELECTRICO
RET

```

;SUBROUTINA QUE REALIZA UNA REVISIÒN DE LOS VALORES DPL Y DPH  
;CON LOS VALORES MAXIMOS DEL AREA DE DATOS DESTINADA A TRACCIÒN  
ELECTRICO

COMPAREAE:

```

LCALL   BANCO0
CJNE    R5,#05H,SALE ;COMPARA EL VALOR DPH CON EL VALOR MAXIMO DEL
AREA
CJNE    R4,#0DCH,SALE ;COMPARA DPL CON EL VALOR MAXIMO DEL AREA
MOV      DPTR,#32     ;SI SOBREPASO APUNTA DIRECCIÒN DE DPL

```



```

MOV     A,#065H           ;INICIALIZA CON DPL BASE
MOV     R4,A              ;ACTUALIZA EL REGISTRO DESTINADO PARA DPL
MOVX    @DPTR,A          ;GUARDA EL VALOR BASE DE DPL
INC     DPTR              ;APUNTA DIRECCIOÀN DE DPH
MOV     A,#00H           ;INICIALIZA CON DPH BASE
MOV     R5,A              ;ACTUALIZA EL REGISTRO DESTINADO PARA DPH
MOVX    @DPTR,A          ;GUARDA EL VALOR BASE DE DPH
SALE:
RET

```

```

;SUBROUTINA DE ENCENDIDO DE LEDS INDICADORES
;PARA TRACCIÒN ELECTRICA, DIESEL, DESCARRILO , DESCARGA DE DATOS
;Y FUNCIONAMIENTO NORMAL

```

LEDS:

```

MOV     DPTR,#080         ;DIRECCIÒN DEL CODIGO TIPO DE TRACCIÒN
MOVX    A,@DPTR          ;LECTURA DEL CODIGO
CJNE    A,#204,LED1      ;ELECTRICO
CLR     P1.5              ;ILUMINA LED TRACCIÒN ELECTRICA
SETB    P1.6              ;APAGA LED TRACCIÒN DIESEL
SETB    P1.7              ;APAGA LED DESCARRILO

```

LED1:

```

CJNE    A,#209,LED2      ;DIESEL
CLR     P1.6              ;ILUMINA LED TRACCIÒN DIESEL
SETB    P1.5              ;
SETB    P1.7              ;

```

LED2:

```

CJNE    A,#203,LED3      ;DESCARRILO
SETB    P1.5              ;ILUMINA LED TRACCIÒN ELECTRICO
CLR     P1.7              ;ILUMINA LED DESCARRILO
SETB    P1.6              ;

```

LED3:

```

CJNE    A,#200,LED4      ;CERO
SETB    P1.6              ;APAGA LED TRACCIÒN DIESEL
SETB    P1.5              ;APAGA LED TRACCIÒN ELECTRICO
SETB    P1.7              ;APAGA LED DESCARRILO

```

LED4:

```

MOV     DPTR,#060H        ;APUNTA LA DIRECCIÒN DE CODIGO DE AVISO
MOV     A,#067H
MOVX    @DPTR,A          ;LEE DATO
RET

```

```

;SUBROUTINA QUE DETECTA EL TIPO DE TRACCIÒN EN QUE SE ENCUENTRA
;EL VEHICULO Y GUARDA EL CODIGO DEL MISMO EN LA DIRECCIÒN 80

```

DETECCION:

```

JB      P3.3,DETDIE      ;DETECTA SELECCIÒN ELECTRICO
JB      P3.4,DETELEC     ;DETECTA DESCARRILAMIENTO
MOV     DPTR,#80         ;SI LOS DOS SON NEGATIVOS
MOV     A,#203           ;ENTONCES HAY DESCARRILAMIENTO
MOVX    @DPTR,A          ;GUARDA CODIGO DE DESCARRILO IGUAL A 203
LJMP    DETSAL

```

DETELEC:

```

MOV     DPTR,#80         ;SI EL SELECTOR EN ELECTRICO
MOV     A,#204           ;GUARDA CODIGO IGUAL A 204
MOVX    @DPTR,A
LJMP    DETSAL

```

DETDIE:

```

JB      P3.5,DETCERO     ;SI EL SELECTORO EN DIESEL
MOV     DPTR,#80         ;GUARDA EL CODIGO IGUAL A 209

```



```

MOV     A, #209
MOVX    @DPTR, A
LJMP    DETSAL
DETCERO:
MOV     DPTR, #80           ;SI NIGUNA SELECCIÒN
MOV     A, #200           ;GUARDA CODIGO IGUAL A 200
MOVX    @DPTR, A
DETSAL:
RET

```

```

;SUBROUTINA PARA PONER NUMERO 255 EN CADA DIRECCIÒN
;DE LA MEMORIA RAM

```

```

ENCERAR:
MOV     DPTR, #060H       ;APUNTA LA DIRECCIÒN DE CODIGO DE AVISO
MOVX    A, @DPTR         ;LEE DATO
CJNE    A, #067H, EN CER ;SI EL DATO LEIDO NO ES IGUAL AL CODIGO DE
AVISO
LJMP    FINENC           ;INICIALIZA CON LOS VALORES BASES DE INICIO

```

```

ENCER:
LCALL   BANCO3          ;SELECCIONA EL BANCO DE REGISTROS 3
CLR     P1.4            ;ENCIENDE LED DE FUNCIÒN NORMAL
MOV     DPTR, #065H     ;APUNTA DIRECCION BASE DE DATOS
MOV     A, #0FFH        ;PONE EN EL ACUMULADOR EL VALOR 255
MOV     R3, #0FEH       ;PONE EN LOS CONTADORES R3 EL VALOR 254
MOV     R4, #081H       ;PONE EN EL CONTADOR R4 129 (129*254=32766)
PONER:
MOVX    @DPTR, A        ;PONE EN LA DIRECCIÒN APUNTADA EL VALOR 255
INC     DPTR            ;APUNTA DIRECCIÒN SIGUIENTE
DJNZ    R4, PONER       ;CUENTA 129
MOV     R4, #081H       ;INICIALIZA R4 CON 129
DJNZ    R3, PONER       ;CUENTA 254 VECES 129 (32766)
FINENC:
LCALL   GUARDAR         ;LLAMA SUBROUTINA DE GUARDAR DATOS
SETB    P1.4            ;APAGA LED INDICADOR
LCALL   BANCO0          ;LLAMA BANCO DE REGISTROS CERO
LCALL   DIRECO
RET

```

```

;SUBROUTINA PARA ALMACENAR DATOS, VERIFICA EL TIPO DE TRACCIÒN
;TRABAJA CON SUBROUTINAS DE ALMACENAMIENTO DE
;DATOS PARA TRACCIÒN DIESEL, ELECTRICO

```

```

GUARDAR:
JB      P3.3, DIESE     ;SUBROUTINA DE ALMACENAMIENTO DE DATOS
LCALL   ELECTRICO       ;DETECTA TIPO DE TRACCIÒN ELECTRICO
LJMP    FING            ;GUARDA DATOS DE TRACCIÒN ELECTRICO
DIESE:
JB      P3.5, FING1     ;SUBROUTINA DE ALMACENAMIENTO TRACCIÒN DIESEL
LCALL   DIESEL          ;DETECTA TIPO DE TRACCIÒN DIESEL
LJMP    FING            ;GUARDA DATOS DE TRACCIÒN DIESEL
FING1:
MOV     DPTR, #080     ;APUNTA DIRECCIÒN TIPO DE TRACCIÒN
MOV     A, #200        ;GUARDA CODIGO NINGUNA TRACCIÒN
MOVX    @DPTR, A       ;CODIGO CERO IGUAL 202
FING:
LCALL   BANCO0         ;SELECCIONA BANCO DE REGISTROS CERO
RET

```





```

MOV     R0,#00           ;INICIALIZA EL REGISTRO
INC     R1               ;CUENTA MULTIPLOS DE 250
CJNE   R1,#100,FINE1   ;VERIFICA SI REBASA NUMERO DE MUESTRAS
MOV     R1,#00           ;INICIALIZA EL REGISTRO
FINE1:
INC     R2               ;CUENTA EL NUMERO DE MUESTRAS TOTALES
CJNE   R2,#250,FINE1   ;VERIFICA SI SOBREPASO NUMERO MAXIMO DE
MUESTRAS
MOV     R2,#00           ;INICIALIZA EL REGISTRO
INC     R3               ;INCREMENTA MULTIPLO DE 250
CJNE   R3,#100,FINE1   ;VERIFICA SI SOBREPASA NUEMRO MAXIMO DE
MUESTRAS
MOV     R3,#00           ;INICIALIZA EL REGISTRO
FINE1:
LCALL  BANCO0           ;LLAMA AL BANCO DE REGISTRO CERO
MOV     DPTR,#080        ;APUNTA DIRECCIÒN TIPO DE TRACCIÒN
MOV     A,#204           ;GUARDA CODIGO TRACCIÒN ELECTRICO
MOVX   @DPTR,A          ;CODIGO ELECTRICO IGUAL 200
MOV     DPTR,#060H       ;GUARDA CODIGO QUE INDICA USO DE MEMORIA
MOV     A,#067H          ;RAM EXTERNA POR PRIMERA VEZ
MOVX   @DPTR,A
FINELEC:
RET

```

```

;SUBROUTINA PARA GUARDAR DATOS DE DESCARRILAMIENTOS, GAURDA LA
VELOCIDAD
;FECHA Y HORA DEL DESCARRILO, EN EL AREA DESTINADA PARA LOS DATOS
;DE DESCARRILO.
;ACTUALIZA DPL Y DPH, ALMACENANDO EN LAS DIRECCIÒNES 018 Y 019H

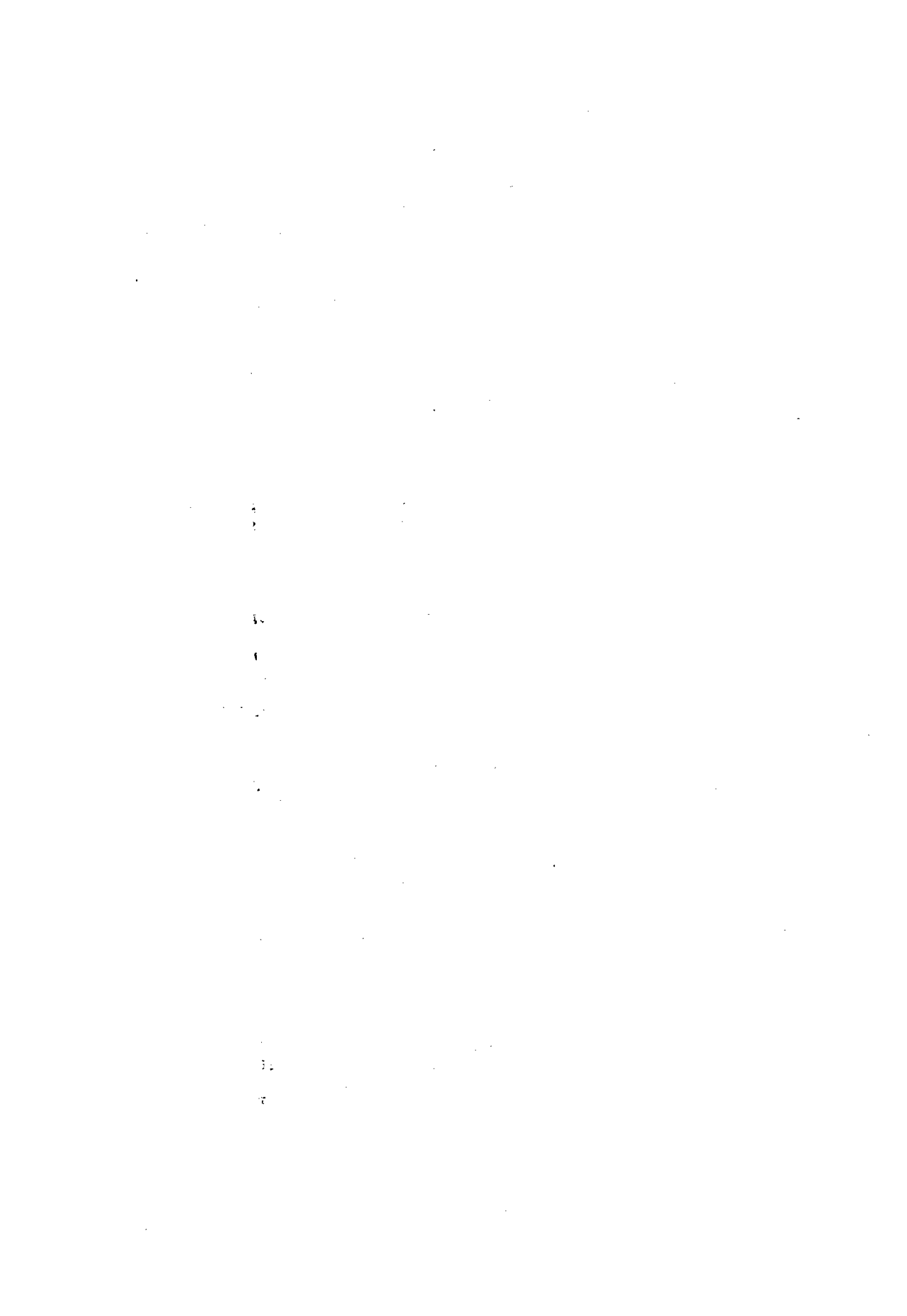
```

DESCARRILO:

```

LCALL  BANCO0
MOV     DPTR,#080        ;APUNTA DIRECCIÒN TIPO DE TRACCIÒN
MOV     A,#203           ;GUARDA CODIGO DESCARRILO
MOVX   @DPTR,A          ;CODIGO DESCARRILO IGUAL 203
LCALL  LEDS
MOV     DPTR,#09H        ;DIRECCIÒN DE RAM EXTERNA QUE CONTIENE DATO DE
VELOCIDAD
MOVX   A,@DPTR          ;LEE DATO DE VELOCIDAD
MOV     R3,A             ;Y LO RETIENE EN R3
MOV     DPTR,#060H       ;GUARDA CODIGO QUE INDICA USO DE MEMORIA
MOV     A,#067H          ;RAM EXTERNA POR PRIMERA VEZ
MOVX   @DPTR,A
MOV     DPTR,#09H        ;DIRECCIÒN DE RAM EXTERNA QUE CONTIENE DATO DE
VELOCIDAD
MOVX   A,@DPTR          ;LEE DATO DE VELOCIDAD
MOV     DPTR,#01CH       ;APUNTA DPL DE DESCARRILO
MOVX   A,@DPTR          ;LEE DPL
MOV     R0,A             ;GUARDA DPL EN R0
INC     DPTR             ;APUNTA DPH DESCARRILO
MOVX   A,@DPTR          ;LEE DPH
MOV     R1,A             ;GUARDA DPH EN R1
MOV     A,R3             ;RECUPERA VALOR DE VELOCIDAD
MOV     DPL,R0           ;ACTUALIZA DPL
MOV     DPH,R1           ;ACTUALIZA DPH
MOVX   @DPTR,A          ;GUARDA VALOR DE VELOCIDAD
INC     DPTR             ;ACTUALIZA NUEVA DIRECCIÒN DE
DESCARRILAMIENTOS
MOV     R4,DPL           ;RETIENE NUEVO DPL EN R4
MOV     R5,DPH           ;RETIENE NUEVO DPH EN R5

```



```

MOV     DPTR,#24      ;APUNTA DIRECCIÒN DE DPL DIESEL
MOVX   A,@DPTR      ;LEE VALOR DE DPL DIESEL
MOV     R0,A         ;RETIENE DPL EN R0
INC     DPTR         ;APUNTA DIRECCIÒN DE DPH
MOVX   A,@DPTR      ;LEE DPH DIESEL
MOV     R1,A         ;RETIENE VALOR DPH EN R1
MOV     A,R3         ;RECUPERA VELOCIDAD
MOV     DPL,R0       ;ACTUALIZA DPL
MOV     DPH,R1       ;ACTUALIZA DPH
MOVX   @DPTR,A      ;GUARDA VELOCIDAD EN AREA DE DATOS PARA
TRACCIÒN D
INC     DPTR         ;ACTUALIZA LA DIRECCIÒN DE DATOS DIESEL
MOV     R4,DPL       ;RETIENE VALOR DE DPL
MOV     R5,DPH       ;RETIENE EL VALOR DPH
MOV     DPTR,#24    ;APUNTA DIRECCIÒN DE DPL
MOV     A,R4         ;RECUPERA DPL
MOVX   @DPTR,A      ;GUARDA VALOR ACTUAL DE DPL DIESEL
INC     DPTR         ;APUNTA DIRECCIÒN DE DPH
MOV     A,R5         ;RECUPERA DPH
MOVX   @DPTR,A      ;GUARDA DPH ACTUAL DE DIESEL
CJNE   R5,#075H,FIN ;COMPARA SI SOBREPASO DPH DE AREA DE DATOS
DIESEL
CJNE   R4,#030H,FIN ;COMPARA SI SOBREPASO DPL DE AREA DE DATOS
DIESEL
MOV     DPTR,#018H  ;SI SOBREPESO EL AREA DE DATOS
MOV     A,#0A9H     ;INICIALIZA DPL Y DPH CON LOS
MOVX   @DPTR,A      ;VALORES DE LA DIRECCIÒN INICIO BASE
INC     DPTR         ;PARA LOS DATOS DE TRACCIÒN DIESEL
MOV     A,#061H
MOVX   @DPTR,A
FIND:
LCALL  BANCO2
INC     R4
CJNE   R4,#50,FINDI
MOV     R4,#00
INC     R5
CJNE   R5,#100,FINDI
MOV     R5,#00
FINDI:
INC     R6
CJNE   R6,#50,FINDI1
MOV     R6,#00
INC     R7
CJNE   R7,#100,FINDI1
MOV     R7,#00
FINDI1:
LCALL  BANCO0
MOV     DPTR,#080   ;APUNTA DIRECCIÒN TIPO DE TRACCIÒN
MOV     A,#209     ;GUARDA CODIGO TRACCIÒN DIESEL
MOVX   @DPTR,A     ;CODIGO DIESEL IGUAL 201
RET

```

```

;SUBROUTINA QUE MANTIENE IGUAL EL RELOJ IMPLEMENTADO EN EL
MICROPROCESADOR
;UTLIZA EL BANCO DE REGISTROS UNO

```

```

TIEMPO:
LCALL  BANCO1      ;LLAMA BANCO DE REGISTROS UNO
INC     R0         ;INCREMENTA EL REGISTRO QUE CUENTA SEGUNDOS
CJNE   R0,#03CH,SALI1 ;CUENTA 60 SEGUNDOS

```



```

LJMP     SALI2
SALI1:
LJMP     SALIDA
SALI2:
MOV      R0,#00H           ;INICIALIZA EN CERO SEGUNDOS
INC      R1                ;INCREMENTA EL REGISTRO QUE CUENTA MINUTOS
CJNE     R1,#03CH,SALIDA  ;CUENTA 60 MINUTOS
MOV      R1,#00H           ;INICIALIZA EN CERO MINUTOS
INC      R2                ;INCREMENTA EL REGISTRO QUE CUENTA HORAS
CJNE     R2,#018H,SALIDA ;CUENTA 24 HORAS
MOV      R2,#00H           ;INICIALIZA EN CERO HORAS
INC      R3                ;INCREMENTA REGISTRO QUE CUENTA DIAS
CJNE     R4,#01,ETQU0
CJNE     R3,#32,SALIDA    ;CUENTA 31 DIAS
MOV      R3,#01H          ;INICIALIZA EN DIA 1
INC      R4                ;INCREMENTA REGISTRO DE MESES
ETQU0:
CJNE     R4,#02,ETQU1
CJNE     R3,#29,SALIDA    ;CUENTA 28 DIAS
MOV      R3,#01H          ;INICIALIZA EN DIA 1
INC      R4                ;INCREMENTA REGISTRO DE MESES
ETQU1:
CJNE     R4,#03,ETQU2
CJNE     R3,#32,SALIDA    ;CUENTA 31 DIAS
MOV      R3,#01H          ;INICIALIZA EN DIA 1
INC      R4                ;INCREMENTA REGISTRO DE MESES
ETQU2:
CJNE     R4,#04,ETQU3
CJNE     R3,#31,SALIDA    ;CUENTA 30 DIAS
MOV      R3,#01H          ;INICIALIZA EN DIA 1
INC      R4                ;INCREMENTA REGISTRO DE MESES
ETQU3:
CJNE     R4,#05,ETQU4
CJNE     R3,#31,SALIDA    ;CUENTA 30 DIAS
MOV      R3,#01H          ;INICIALIZA EN DIA 1
INC      R4                ;INCREMENTA REGISTRO DE MESES
ETQU4:
CJNE     R4,#06,ETQU5
CJNE     R3,#31,SALIDA    ;CUENTA 30 DIAS
MOV      R3,#01H          ;INICIALIZA EN DIA 1
INC      R4                ;INCREMENTA REGISTRO DE MESES
ETQU5:
CJNE     R4,#07,ETQU6
CJNE     R3,#32,SALIDA    ;CUENTA 31 DIAS
MOV      R3,#01H          ;INICIALIZA EN DIA 1
INC      R4                ;INCREMENTA REGISTRO DE MESES
ETQU6:
CJNE     R4,#08,ETQU7
CJNE     R3,#30,SALIDA    ;CUENTA 29 DIAS
MOV      R3,#01H          ;INICIALIZA EN DIA 1
INC      R4                ;INCREMENTA REGISTRO DE MESES
ETQU7:
CJNE     R4,#09,ETQU8
CJNE     R3,#31,SALIDA    ;CUENTA 30 DIAS
MOV      R3,#01H          ;INICIALIZA EN DIA 1
INC      R4                ;INCREMENTA REGISTRO DE MESES
ETQU8:
CJNE     R4,#10,ETQU9
CJNE     R3,#32,SALIDA    ;CUENTA 31 DIAS
MOV      R3,#01H          ;INICIALIZA EN DIA 1

```



```

INC      R4                ;INCREMENTA REGISTRO DE MESES
ETQU9:
CJNE    R4,#11,ETQU10
CJNE    R3,#31,SALIDA    ;CUENTA 30 DIAS
MOV     R3,#01H          ;INICIALIZA EN DIA 1
INC     R4                ;INCREMENTA REGISTRO DE MESES
ETQU10:
CJNE    R4,#12,ETQU11
CJNE    R3,#32,SALIDA    ;CUENTA 31 DIAS
MOV     R3,#01H          ;INICIALIZA EN DIA 1
INC     R4                ;INCREMENTA REGISTRO DE MESES
ETQU11:
CJNE    R4,#0DH,SALIDA   ;CUENTA 12 MESES
MOV     R4,#01H          ;INICIALIZA EN MES 1
INC     R5                ;INCREMENTA REGISTRO QUE CUENTA AÑOS
SALIDA:
MOV     DPTR,#0BH        ;APUNTA DIRECCIÒN DE CODIGO DE INICIO
MOV     A,#065H          ;CARGA CODIGO
MOVX    @DPTR,A          ;GUARDA CODIGO
LCALL   RERAM            ;GUARDA VALORES DEL RELOJ EN RAM EXTERNA
LCALL   BANCOO
RET

;SUBROUTINA DE COMUNICACIÒN SERIAL, VERIFICA RI PARA SABER SI EL PC
;ESTA INTENTANDO COMUNICARSE, GUARDA EL VALOR DE VELOCIDAD.
;EN DIRECCIÒN 09H DE RAM EXTERNA
;RECIBE EL PRIMER DATO Y COMPARA CON LOS CODIGOS QUE LE INDICAN
;SI QUIERE MONITOREAR LA VELOCIDAD, IGUALAR EL RELOJ O
;DESCARGAR DATOS

SERIAL:
JNB     RI,TERRR          ;SI NO HAY BIT SALE
LJMP    COMS
TERRR:
LJMP    TERM11

COMS:
MOV     A,SBUF            ;LEE PUERTO SERIAL
CLR     RI                ;LIMPIA BANDERA DE RECEPCIÒN
MOV     R6,A              ;RETIENE CODIGO ENVIADO EN R6
MOV     DPTR,#0008H       ;APUNTA DIRECCIÒN DE CODIGO
MOVX    @DPTR,A           ;GUARDA DATO ENVIADO EN DIR 08H RAM EXT
MOV     A,R3              ;LEE DATO DE VELOCIDAD
INC     DPTR              ;APUNTA DIRECCIÒN DE VELOCIDAD EN MEMORIA
EXTERNA
MOVX    @DPTR,A           ;GUARDA DATO DE VELOCIDAD EN 09H RAM EXT
MOV     A,R6              ;RECUPERA CODIGO ENVIADO POR EL PC

CJNE    A,#096H,DATOS     ;VERIFICA SI ES CODIGO PARA MONITOREO DE
VELOCIDAD
LCALL   VELOCIDAD         ;LLAMA SUBROUTINA DE MONITOREO DE VELOCIDAD
LJMP    TERM11

DATOS:
CJNE    A,#097H,MINUT     ;VERIFICA SI ES CODIGO PARA DESCARGAR DATOS
LCALL   DESCARG           ;LLAMA SUBROUTINA PARA DESCARGAR DATOS
LJMP    TERM11

MINUT:
CJNE    A,#098H,HORA      ;VERIFICA SI ES CODIGO PARA IGUALAR MINUTOS

```





```

LCALL    MIN                ;IGUALA MINUTOS
MOV      DPTR,#060H        ;GUARDA CODIGO QUE INDICA USO DE MEMORIA
MOV      A,#067H           ;RAM EXTERNA POR PRIMERA VEZ
MOVX    @DPTR,A
LJMP    TERMI1

HORA:
CJNE    A,#099H,DIA        ;VERIFICA SI ES CODIGO DE IGUALAR HORAS
LCALL   HOR                ;IGUALA HORA
MOV      DPTR,#060H        ;GUARDA CODIGO QUE INDICA USO DE MEMORIA
MOV      A,#067H           ;RAM EXTERNA POR PRIMERA VEZ
MOVX    @DPTR,A
LJMP    TERMI1

DIA:
CJNE    A,#09AH,MES        ;VERIFICA SI ES CODIGO DE IGUALAR DIA
LCALL   DAY                ;IGUALA DIA
LJMP    TERMI1

MES:
CJNE    A,#09BH,ANO        ;VERIFICA SI ES CODIGO DE IGUALAR MES
LCALL   MESS               ;IGUALA MES
LJMP    TERMI1

ANO:
CJNE    A,#09CH, TMUE      ;VERIFICA SI ES CODIGO DE IGUALAR AÑO
LCALL   ANNO               ;IGUALA AÑO
LJMP    TERMI1

TMUE:
CJNE    A,#09DH, TROL      ;VERIFICA SI ES CODIGO DEL TIEMPO DE MUESTREO
LCALL   TMUES              ;LEE VALOR DE TIEMPO DE MUESTREO
LJMP    TERMI1

TROL:
CJNE    A,#09EH, MEMDIE    ;VERIFICA SI ES CODIGO DEL NUMERO DE TROLE
LCALL   TROLE              ;LEE EL VALOR DEL NUMERO DE TROLE
MOV      DPTR,#060H        ;GUARDA CODIGO QUE INDICA USO DE MEMORIA
MOV      A,#067H           ;RAM EXTERNA POR PRIMERA VEZ
MOVX    @DPTR,A
LJMP    TERMI1

MEMDIE:
CJNE    A,#208, MEMELE
LCALL   MEMINTD
LJMP    TERMI1

MEMELE:
CJNE    A,#209, MEM5
LCALL   MEMINTE
LJMP    TERMI1

MEM5:
CJNE    A,#210, MEM10
LCALL   MEM5K
LJMP    TERMI1

MEM10:
CJNE    A,#211, MEM15
LCALL   MEM10K

```



```

LJMP     TERMI1

MEM15:
CJNE    A, #212, MEM20
LCALL   MEM15K
LJMP    TERMI1

MEM20:
CJNE    A, #213, MEM30
LCALL   MEM20K
LJMP    TERMI1

MEM30:
CJNE    A, #215, MEM32
LCALL   MEM30K
LJMP    TERMI1

MEM32:
CJNE    A, #216, TOTAL
LCALL   MEM32K
LJMP    TERMI1

TOTAL:
CJNE    A, #217, BORRAR
LCALL   TOTAL1
LJMP    TERMI1

BORRAR:
CJNE    A, #214, TERMI1
MOV     DPTR, #060H
MOV     A, #070H
MOVX    @DPTR, A
LCALL   ENCERAR
LCALL   BANCO2
MOV     DPTR, #060H
MOV     A, #070H
MOVX    @DPTR, A
LCALL   DIREC0
LCALL   BANCO0
MOV     DPTR, #060H      ;APUNTA LA DIRECCIÒN DE CODIGO DE AVISO
MOV     A, #067H
MOVX    @DPTR, A        ;LEE DATO
TERMI1:
LCALL   BANCO0
MOV     DPTR, #06H      ;DIRECCIÒN DE RAM EXTERNA CONTIENE DATO TM
MOVX    A, @DPTR        ;LECTURA DEL TM
MOV     R7, A           ;TIEMPO DE MUESTREO EN SEGUNDOS
RET

TOTAL1:
LCALL   BANCO2
MOV     A, R2
MOV     SBUF, A         ;ENVIA EL DATO LEIDO HACIA EL BUFER
JNB     TI, $           ;DE COMUNICACIÒN SERIAL
CLR     TI
MOV     A, R3
MOV     SBUF, A         ;ENVIA EL DATO LEIDO HACIA EL BUFER
JNB     TI, $           ;DE COMUNICACIÒN SERIAL
CLR     TI
MOV     A, R6

```



```

MOV     SBUF,A           ;ENVIA EL DATO LEIDO HACIA EL BUFER
JNB     TI,$            ;DE COMUNICACIÀN SERIAL
CLR     TI
MOV     A,R7
MOV     SBUF,A           ;ENVIA EL DATO LEIDO HACIA EL BUFER
JNB     TI,$            ;DE COMUNICACIÀN SERIAL
CLR     TI
LCALL   BANCO0
MOV     DPTR,#06
MOVX    A,@DPTR
MOV     SBUF,A           ;ENVIA EL DATO LEIDO HACIA EL BUFER
JNB     TI,$            ;DE COMUNICACIÀN SERIAL
CLR     TI
INC     DPTR
MOVX    A,@DPTR
MOV     SBUF,A           ;ENVIA EL DATO LEIDO HACIA EL BUFER
JNB     TI,$            ;DE COMUNICACIÀN SERIAL
CLR     TI
RET

MEMINTE:
CLR     P1.4
LCALL   BANCO3           ;SELECCIONA EL BANCO DE REGISTROS 3
MOV     R2,#15           ;PONE EN EL REGISTRO 2 EL NUMERO 15
MOV     R7,#100          ;PONE EN R7 EL NUMERO 100 (15*100=1500)
MOV     DPTR,#0065H      ;DIRECCIÀN INICIAL DE MEMORIA RAM EXTERNA
DESME15:
MOVX    A,@DPTR          ;LEE MEMORIA RAM EXTERNA
MOV     SBUF,A           ;ENVIA EL DATO LEIDO HACIA EL BUFER
JNB     TI,$            ;DE COMUNICACIÀN SERIAL
CLR     TI
CJNE    A,#255,SEGUIR1
LJMP    FUERA
SEGUIR1:
JNB     RI,$
MOV     A,SBUF
CLR     RI
MOVX    @DPTR,A
INC     DPTR             ;APUNTA SIGUIENTE DIRECCIÀN
DJNZ    R2,DESME15       ;CUENTA 15
MOV     R2,#15           ;PONE 15 EN R2
DJNZ    R7,DESME15       ;CUENTA 100 VECES 15
MOV     DPTR,#060H
MOV     A,#070H
MOVX    @DPTR,A
FUERA:
LCALL   DIRECO
LCALL   BANCO0           ;LLAMA BANCO DE REGISTROS CERO
SETB    P1.4
RET

MEMINTD:
CLR     P1.4
LCALL   BANCO3           ;SELECCIONA EL BANCO DE REGISTROS 3
MOV     R2,#20           ;PONE EN EL REGISTRO 2 EL NUMERO 20
MOV     R7,#100          ;PONE EN R7 EL NUMERO 100 (20*100=2000)
MOV     DPTR,#05DDH      ;DIRECCIÀN INICIAL DE MEMORIA RAM EXTERNA
DESMD20:
MOVX    A,@DPTR          ;LEE MEMORIA RAM EXTERNA
MOV     SBUF,A           ;ENVIA EL DATO LEIDO HACIA EL BUFER

```



```

JNB     TI, $           ;DE COMUNICACIÀ SERIAL
CLR     TI
CJNE    A, #255, SEGUIR2
LJMP    FUERA2
SEGUIR2:
JNB     RI, $
MOV     A, SBUF
CLR     RI
MOVX    @DPTR, A
INC     DPTR           ;APUNTA SIGUIENTE DIRECCIÀ
DJNZ    R2, DESMD20   ;CUENTA 20
MOV     R2, #20       ;PONE 20 EN R2
DJNZ    R7, DESMD20   ;CUENTA 100 VECES 20
MOV     DPTR, #060H
MOV     A, #070H
MOVX    @DPTR, A
FUERA2:
LCALL   DIRECO
LCALL   BANCO0       ;LLAMA BANCO DE REGISTROS CERO
SETB    P1.4
RET

MEM5K:
CLR     P1.4
LCALL   BANCO3       ;SELECCIONA EL BANCO DE REGISTROS 3
MOV     R2, #25       ;PONE EN EL REGISTRO 2 EL NUMERO 25
MOV     R7, #215      ;PONE EN R7 EL NUMERO 215 (25*215=5375)
MOV     DPTR, #0DADH  ;DIRECCIÀ INICIAL DE MEMORIA RAM EXTERNA
DESM5:
MOVX    A, @DPTR     ;LEE MEMORIA RAM EXTERNA
MOV     SBUF, A      ;ENVIA EL DATO LEIDO HACIA EL BUFER
JNB     TI, $         ;DE COMUNICACIÀ SERIAL
CLR     TI
CJNE    A, #255, SEGUIR3
LJMP    FUERA3
SEGUIR3:
JNB     RI, $
MOV     A, SBUF
CLR     RI
MOVX    @DPTR, A
INC     DPTR         ;APUNTA SIGUIENTE DIRECCIÀ
DJNZ    R2, DESM5    ;CUENTA 25
MOV     R2, #025     ;PONE 25 EN R2
DJNZ    R7, DESM5    ;CUENTA 215 VECES 25
MOV     DPTR, #060H
MOV     A, #070H
MOVX    @DPTR, A
FUERA3:
LCALL   DIRECO
LCALL   BANCO0       ;LLAMA BANCO DE REGISTROS CERO
SETB    P1.4
RET

MEM10K:
CLR     P1.4
LCALL   BANCO3       ;SELECCIONA EL BANCO DE REGISTROS 3
MOV     R2, #25       ;PONE EN EL REGISTRO 2 EL NUMERO 253
MOV     R7, #215      ;PONE EN R7 EL NUMERO 129 (253*129=32637)
MOV     DPTR, #022ACH ;DIRECCIÀ INICIAL DE MEMORIA RAM EXTERNA

```





```

DESM10:
MOVX    A,@DPTR          ;LEE MEMORIA RAM EXTERNA
MOV     SBUF,A           ;ENVIA EL DATO LEIDO HACIA EL BUFER
JNB     TI,$             ;DE COMUNICACIÒN SERIAL
CLR     TI
CJNE    A,#255,SEGUIR4
LJMP    FUERA4
SEGUIR4:
JNB     RI,$
MOV     A,SBUF
CLR     RI
MOVX    @DPTR,A
INC     DPTR             ;APUNTA SIGUIENTE DIRECCIÒN
DJNZ    R2,DESM10        ;CUENTA 253
MOV     R2,#25           ;PONE 253 EN R2
DJNZ    R7,DESM10        ;CUENTA 129 VECES 253
MOV     DPTR,#060H
MOV     A,#070H
MOVX    @DPTR,A
FUERA4:
LCALL   DIRECO
LCALL   BANCO0           ;LLAMA BANCO DE REGISTROS CERO
SETB    P1.4
RET

MEM15K:
CLR     P1.4
LCALL   BANCO3           ;SELECCIONA EL BANCO DE REGISTROS 3
MOV     R2,#25           ;PONE EN EL REGISTRO 2 EL NUMERO 253
MOV     R7,#215          ;PONE EN R7 EL NUMERO 129 (253*129=32637)
MOV     DPTR,#037ABH     ;DIRECCIÒN INICIAL DE MEMORIA RAM EXTERNA
DESM15:
MOVX    A,@DPTR          ;LEE MEMORIA RAM EXTERNA
MOV     SBUF,A           ;ENVIA EL DATO LEIDO HACIA EL BUFER
JNB     TI,$             ;DE COMUNICACIÒN SERIAL
CLR     TI
CJNE    A,#255,SEGUIR5
LJMP    FUERA5
SEGUIR5:
JNB     RI,$
MOV     A,SBUF
CLR     RI
MOVX    @DPTR,A
INC     DPTR             ;APUNTA SIGUIENTE DIRECCIÒN
DJNZ    R2,DESM15        ;CUENTA 253
MOV     R2,#25           ;PONE 253 EN R2
DJNZ    R7,DESM15        ;CUENTA 129 VECES 253
MOV     DPTR,#060H
MOV     A,#070H
MOVX    @DPTR,A
FUERA5:
LCALL   DIRECO
LCALL   BANCO0           ;LLAMA BANCO DE REGISTROS CERO
SETB    P1.4
RET

MEM20K:
CLR     P1.4
LCALL   BANCO3           ;SELECCIONA EL BANCO DE REGISTROS 3
MOV     R2,#25           ;PONE EN EL REGISTRO 2 EL NUMERO 253

```



```

MOV     R7,#215           ;PONE EN R7 EL NUMERO 129 (253*129=32637)
MOV     DPTR,#04CAAH     ;DIRECCIÒN INICIAL DE MEMORIA RAM EXTERNA
DESM20:
MOVX    A,@DPTR         ;LEE MEMORIA RAM EXTERNA
MOV     SBUF,A          ;ENVIA EL DATO LEIDO HACIA EL BUFER
JNB     TI,$            ;DE COMUNICACIÒN SERIAL
CLR     TI
CJNE    A,#255,SEGUIR6
LJMP    FUERA6
SEGUIR6:
JNB     RI,$
MOV     A,SBUF
CLR     RI
MOVX    @DPTR,A
INC     DPTR            ;APUNTA SIGUIENTE DIRECCIÒN
DJNZ    R2,DESM20      ;CUENTA 253
MOV     R2,#25         ;PONE 253 EN R2
DJNZ    R7,DESM20      ;CUENTA 129 VECES 253
MOV     DPTR,#060H
MOV     A,#070H
MOVX    @DPTR,A
FUERA6:
LCALL   DIRECO
LCALL   BANCO0         ;LLAMA BANCO DE REGISTROS CERO
SETB    P1.4
RET

```

```

MEM30K:
CLR     P1.4
LCALL   BANCO3         ;SELECCIONA EL BANCO DE REGISTROS 3
MOV     R2,#064H      ;PONE EN EL REGISTRO 2 EL NUMERO 253
MOV     R7,#032H      ;PONE EN R7 EL NUMERO 129 (253*129=32637)
MOV     DPTR,#061A9H  ;DIRECCIÒN INICIAL DE MEMORIA RAM EXTERNA
DESM30:
MOVX    A,@DPTR       ;LEE MEMORIA RAM EXTERNA
MOV     SBUF,A        ;ENVIA EL DATO LEIDO HACIA EL BUFER
JNB     TI,$          ;DE COMUNICACIÒN SERIAL
CLR     TI
CJNE    A,#255,SEGUIR7
LJMP    FUERA7
SEGUIR7:
JNB     RI,$
MOV     A,SBUF
CLR     RI
MOVX    @DPTR,A
INC     DPTR          ;APUNTA SIGUIENTE DIRECCIÒN
DJNZ    R2,DESM30    ;CUENTA 253
MOV     R2,#064H     ;PONE 253 EN R2
DJNZ    R7,DESM30    ;CUENTA 129 VECES 253
MOV     DPTR,#060H
MOV     A,#070H
MOVX    @DPTR,A
FUERA7:
LCALL   DIRECO
LCALL   BANCO0         ;LLAMA BANCO DE REGISTROS CERO
SETB    P1.4
RET

```

```
MEM32K:
```



```

CLR      P1.4
LCALL   BANCO3          ;SELECCIONA EL BANCO DE REGISTROS 3
MOV     R2,#014H        ;PONE EN EL REGISTRO 2 EL NUMERO 253
MOV     R7,#08AH        ;PONE EN R7 EL NUMERO 129 (253*129=32637)
MOV     DPTR,#07531H    ;DIRECCIÒN INICIAL DE MEMORIA RAM EXTERNA
DESM32:
MOVX    A,@DPTR         ;LEE MEMORIA RAM EXTERNA
MOV     SBUF,A          ;ENVIA EL DATO LEIDO HACIA EL BUFER
JNB     TI,$            ;DE COMUNICACIÒN SERIAL
CLR     TI
CJNE    A,#255,SEGUIR8
LJMP    FUERA8
SEGUIR8:
JNB     RI,$
MOV     A,SBUF
CLR     RI
MOVX    @DPTR,A
INC     DPTR            ;APUNTA SIGUIENTE DIRECCIÒN
DJNZ    R2,DESM32      ;CUENTA 253
MOV     R2,#014H        ;PONE 253 EN R2
DJNZ    R7,DESM32      ;CUENTA 129 VECES 253
MOV     DPTR,#060H
MOV     A,#070H
MOVX    @DPTR,A
FUERA8:
LCALL   DIRECO
LCALL   BANCO0          ;LLAMA BANCO DE REGISTROS CERO
SETB    P1.4
RET

```

```

;SUBROUTINA PARA MONITOREO DE LA VELOCIDAD EN EL PC
;ENVIA DATOS DEL RELOJ (HORA Y FECHA)
;NUMERO DE TROLEBUS, TIEMPO DE MUESTREO Y LA VELOCIDAD
;DEL TROLEBÉS.

```

```

VELOCIDAD:
MOV     R5,#0AH         ;CONTADOR DE DATOS DE RAM EXTERNA 11
MOV     DPTR,#0000H     ;APUNTA DIRECCIÒN DE INICIO DE DATOS A ENVIAR
ENVIO1:
MOVX    A,@DPTR         ;LEE DATO A ENVIAR
MOV     SBUF,A          ;ENVIA DATO VIA SERIAL
JNB     TI,$            ;DETECTA SI EL DATO FUE RECIBIDO
CLR     TI              ;LIMPIA BANDERA DE TRASNMISSIÒN
INC     DPTR            ;APUNTA DIRECCIÒN DE DATO SIGUIENTE
DJNZ    R5,ENVIO1      ;CUENTA ONCE DATOS
MOV     DPTR,#080
MOVX    A,@DPTR
MOV     SBUF,A
JNB     TI,$
CLR     TI
RET

```

```

;SUBROUTINA PARA IGUALAR LOS MINUTOS, RECIBE EL DATO
;Y LO DEVUELVE AL PC PARA COMPROBAR SU CORRECTA RECEPCIÒN
;LUEGO ALMACENA EL DATO EN EMORIA RAM EXTERNA

```

```

MIN:
LCALL   BANCO1          ;LLAMA BANCO DE REGISTROS 1
MOV     SBUF,A          ;ENVIA EL CODIGO ENVIADO DESDE EL PC

```



```

JNB     TI,$           ;VERIFICA SI EL PC RECIBIO EL DATO
CLR     TI             ;LIMPIA BANDERA DE TRANSMISIÒN SERIAL
JNB     RI,$           ;ESPERA EL BIT DE INICIO DEL DATO MINUTO
MOV     A,SBUF         ;LEE BUFER DE RECEPCIÒN SERIAL
CLR     RI             ;LIMPIA BANDERA DE RECEPCIÒN
MOV     R1,A           ;RETIENE MINUTOS EN R1
LCALL   RERAM         ;GUARDA EL DATO EN RAM EXTERNA
RET

```

```

;SUBROUTINA PARA IGUALAR LAS HORAS, RECIBE EL DATO
;Y LO DEVUELVE AL PC PARA COMPROBAR SU CORRECTA RECEPCIÒN
;LUEGO ALMACENA EL DATO EN EMORIA RAM EXTERNA

```

```

HOR:
LCALL   BANCO1        ;LLAMA BANCO DE REGISTROS 1
MOV     SBUF,A         ;ENVIA EL CODIGO ENVIADO DESDE EL PC
JNB     TI,$           ;VERIFICA SI EL PC RECIBIO EL DATO
CLR     TI             ;LIMPIA BANDERA DE TRANSMISIÒN SERIAL
JNB     RI,$           ;ESPERA EL BIT DE INICIO DEL DATO HORAS
MOV     A,SBUF         ;LEE BUFER DE RECEPCIÒN SERIAL
CLR     RI             ;LIMPIA BANDERA DE RECEPCIÒN
MOV     R2,A           ;RETIENE HORAS EN R2
LCALL   RERAM         ;GUARDA EL DATO EN RAM EXTERNA
RET

```

```

;SUBROUTINA PARA IGUALAR EL DIA, RECIBE EL DATO
;Y LO DEVUELVE AL PC PARA COMPROBAR SU CORRECTA RECEPCIÒN
;LUEGO ALMACENA EL DATO EN EMORIA RAM EXTERNA

```

```

DAY:
LCALL   BANCO1        ;LLAMA BANCO DE REGISTROS 1
MOV     SBUF,A         ;ENVIA EL CODIGO ENVIADO DESDE EL PC
JNB     TI,$           ;VERIFICA SI EL PC RECIBIO EL DATO
CLR     TI             ;LIMPIA BANDERA DE TRANSMISIÒN SERIAL
JNB     RI,$           ;ESPERA EL BIT DE INICIO DEL DATO DIA
MOV     A,SBUF         ;LEE BUFER DE RECEPCIÒN SERIAL
CLR     RI             ;LIMPIA BANDERA DE RECEPCIÒN
MOV     R3,A           ;RETIENE DIA EN R3
LCALL   RERAM         ;GUARDA EL DATO EN RAM EXTERNA
RET

```

```

;SUBROUTINA PARA IGUALAR EL MES, RECIBE EL DATO
;Y LO DEVUELVE AL PC PARA COMPROBAR SU CORRECTA RECEPCIÒN
;LUEGO ALMACENA EL DATO EN EMORIA RAM EXTERNA

```

```

MESS:
LCALL   BANCO1        ;LLAMA BANCO DE REGISTROS 1
MOV     SBUF,A         ;ENVIA EL CODIGO ENVIADO DESDE EL PC
JNB     TI,$           ;VERIFICA SI EL PC RECIBIO EL DATO
CLR     TI             ;LIMPIA BANDERA DE TRANSMISIÒN SERIAL
JNB     RI,$           ;ESPERA EL BIT DE INICIO DEL DATO MES
MOV     A,SBUF         ;LEE BUFER DE RECEPCIÒN SERIAL
CLR     RI             ;LIMPIA BANDERA DE RECEPCIÒN
MOV     R4,A           ;RETIENE MES EN R4
LCALL   RERAM         ;GUARDA EL DATO EN RAM EXTERNA
RET

```

```

;SUBROUTINA PARA IGUALAR EL AÑO, RECIBE EL DATO
;Y LO DEVUELVE AL PC PARA COMPROBAR SU CORRECTA RECEPCIÒN
;LUEGO ALMACENA EL DATO EN EMORIA RAM EXTERNA

```





```

ANNO:
LCALL  BANCO1          ;LLAMA BANCO DE REGISTROS 1
MOV    SBUF,A          ;ENVIA EL CODIGO ENVIADO DESDE EL PC
JNB    TI,$            ;VERIFICA SI EL PC RECIBIO EL DATO
CLR    TI              ;LIMPIA BANDERA DE TRANSMISIÒN SERIAL
JNB    RI,$            ;ESPERA EL BIT DE INICIO DEL DATO AÑO
MOV    A,SBUF          ;LEE BUFER DE RECEPCIÒN SERIAL
CLR    RI              ;LIMPIA BANDERA DE RECEPCIÒN
MOV    R5,A            ;RETIENE AÑO EN R5
LCALL  RERAM           ;GUARDA EL DATO EN RAM EXTERNA
RET

```

```

;SUBROUTINA PARA RECIBIR EL TEIMPO DE MUESTREO EN SEGUNDOS
;EL DATO RECIBIDO SE DEVUELVE AL PC PARA COMPROBAR SU CORRECTA
; RECEPCIÒN LUEGO ALMACENA EL DATO EN EMORIA RAM EXTERNA

```

```

TMUES:
LCALL  BANCO1          ;LLAMA BANCO DE REGISTROS 1
MOV    SBUF,A          ;ENVIA EL CODIGO ENVIADO DESDE EL PC
JNB    TI,$            ;VERIFICA SI EL PC RECIBIO EL DATO
CLR    TI              ;LIMPIA BANDERA DE TRANSMISIÒN SERIAL
JNB    RI,$            ;ESPERA EL BIT DE INICIO DEL DATO TIEMPO DE
MUESTREO
MOV    A,SBUF          ;LEE BUFER DE RECEPCIÒN SERIAL
CLR    RI              ;LIMPIA BANDERA DE RECEPCIÒN
MOV    R6,A            ;RETIENE TIEMPO DE MUESTREO EN R6
LCALL  RERAM           ;GUARDA EL DATO EN RAM EXTERNA
RET

```

```

;SUBROUTINA PARA CAMBIAR EL NUMERO DEL TROLEBÉS , RECIBE EL DATO
;Y LO DEVUELVE A PC PARA COMPROBAR SU CORRECTA RECEPCIÒN
;LUEGO ALMACENA EL DATO EN EMORIA RAM EXTERNA

```

```

TROLE:
LCALL  BANCO1          ;LLAMA BANCO DE REGISTROS 1
MOV    SBUF,A          ;ENVIA EL CODIGO ENVIADO DESDE EL PC
JNB    TI,$            ;VERIFICA SI EL PC RECIBIO EL DATO
CLR    TI              ;LIMPIA BANDERA DE TRANSMISIÒN SERIAL
JNB    RI,$            ;ESPERA EL BIT DE INICIO DEL DATO # DE TROLE
MOV    A,SBUF          ;LEE BUFER DE RECEPCIÒN SERIAL
CLR    RI              ;LIMPIA BANDERA DE RECEPCIÒN
MOV    R7,A            ;RETIENE # DE TROLE EN R7
LCALL  RERAM           ;GUARDA EL DATO EN RAM EXTERNA
RET

```

```

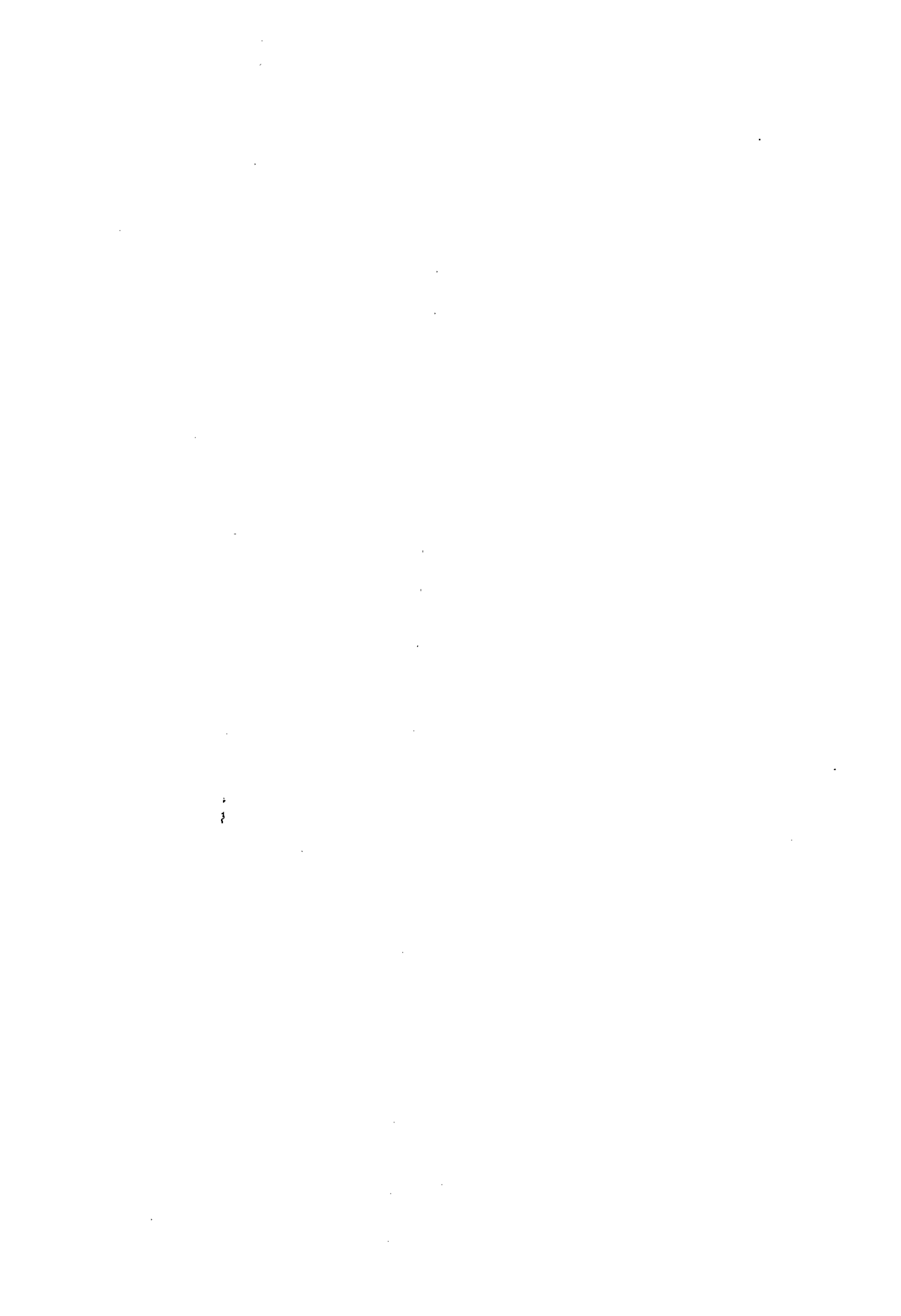
;SUBROUTINA PARA DESCARGAR DATOS DE MEMORIA RAM EXTERNA HACIA
;EL PC, DESCARGA 32766 DATOS, UTILIZA LOS REGISTROS R2 Y R7
;DEL BANCO DE REGISTROS 3 PARA CONTABILIZAR ESTOS DATOS

```

```

DESCARG:
CLR    P1.4
LCALL  BANCO3          ;SELECCIONA EL BANCO DE REGISTROS 3
MOV    R2,#0FDH        ;PONE EN EL REGISTRO 2 EL NUMERO 253
MOV    R7,#081H        ;PONE EN R7 EL NUMERO 129 (253*129=32637)
MOV    DPTR,#0065H     ;DIRECCIÒN INICIAL DE MEMORIA RAM EXTERNA
DESCAR1:
MOVX   A,@DPTR         ;LEE MEMORIA RAM EXTERNA
MOV    SBUF,A          ;ENVIA EL DATO LEIDO HACIA EL BUFER
JNB    TI,$            ;DE COMUNICACIÒN SERIAL
CLR    TI

```



```

JNB     RI, $
MOV     A, SBUF
CLR     RI
MOVX    @DPTR, A
INC     DPTR                ;APUNTA SIGUIENTE DIRECCIÒN
DJNZ   R2, DESCAR1         ;CUENTA 253
MOV     R2, #0FEH          ;PONE 253 EN R2
DJNZ   R7, DESCAR1         ;CUENTA 129 VECES 253
MOV     DPTR, #060H
MOV     A, #070H
MOVX    @DPTR, A
LCALL   DIREC0
LCALL   BANCO0             ;LLAMA BANCO DE REGISTROS CERO
SETB    P1.4
RET

```

```

;SUBROUTINA PARA RETARDO DE 1 SEGUNDO EN TIEMPO REAL
;HABILITA LA INTERRUPTIÒN CERO PARA MEDIR LA FRECUENCIA
;DE LA SEYAL DEL TACOGRIFO

```

SEGUNDO:

```

LCALL   HAB                ;HABILITA INTERRUPTIÒN EXTERNA CERO
CLR     TR0                ;DESHABILITA EL TIMER 0
CLR     P1.4              ;ACTIVA LED DE TIEMPO
SETB    TR0                ;ACTIVA TIMER 0
LCALL   MEDS              ;LLAMA RETARDO DE MEDIO SEGUNDO
CLR     TR0                ;DESHABILITA TIMER 0
SETB    P1.4              ;DESACTIVA LED E TIEMPO
SETB    TR0                ;ACTIVA TIMER CERO
LCALL   MEDS              ;LLAMA RETARDO DE MEDIO SEGUNDO
LCALL   DESHAB           ;DESHABILITA INTERRUPTIÒN CERO
LCALL   BANCO0
MOV     A, R3              ;LEE DATO DE VELOCIDAD
MOV     DPTR, #09H        ;APUNTA DIRECCIÒN DE VELOCIDAD EN MEMORIA
EXTERNA
MOVX    @DPTR, A          ;GUARDA DATO DE VELOCIDAD EN 09H RAM EXT
LCALL   LEDS
RET

```

;SUBROUTINA DE HABILITACIÒN DE INTERRUPTIÒN CERO

```

HAB:
MOV     R3, #00H          ;PONE EN CERO EL REGISTRO QUE MIDE LA
FRECUENCIA
SETB    IE                ;HABILITA INTERRUPTIÒNES
SETB    EA                ;PONE EN 1 EL BIT QUE HABILITA TODAS LA
INTERRUPTIÒNES
SETB    IT0              ;HABILITITA INETRUPCIÒN CERO
RET

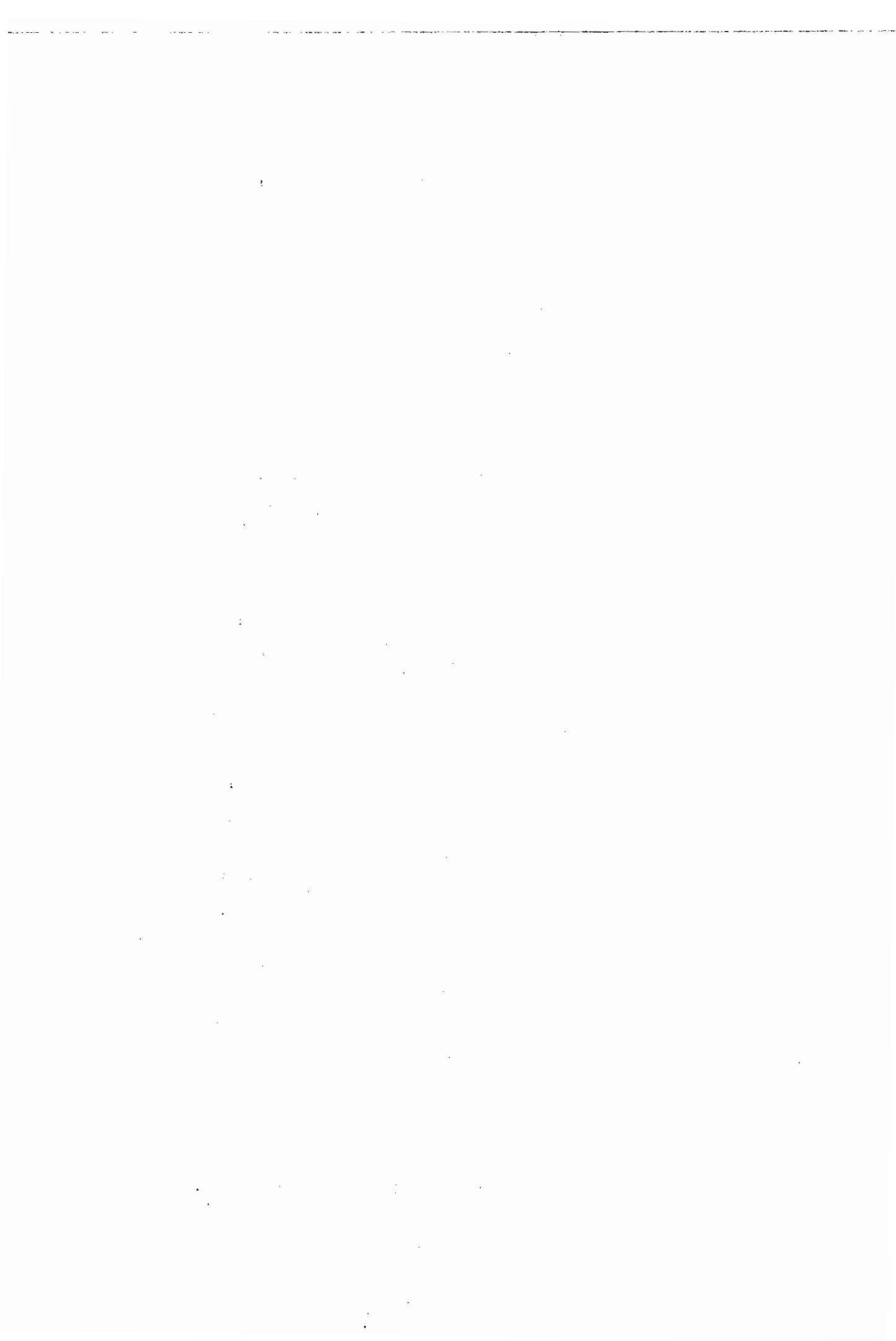
```

;SUBROUTINA PARA DESHABILITAR LA INTERRUPTIÒN EXTERNA CERO

```

DESHAB:
CLR     IE                ;DESHABILITA INTERRUPTIÒNES
CLR     EA                ;PONE EN 0 BIT QUE HABILITA TODAS LAS
INTERRUPTIÒNES
CLR     IT0              ;DESHABILITA INTERRUPTIÒN EXTERNA CERO
RET

```



```

;SUBROUTINA QUE INICIALIZA TODAS LAS DIRECCIONES DEL AREA DE DATOS
;DE TRACCIÀN ELECTRICA, DIESEL Y DESCARRILOS, SI NO ENCUENTRA EL
;CODIGO DE AVISO.
DIRECO:
MOV     DPTR,#060H      ;APUNTA LA DIRECCIÀN DE CODIGO DE AVISO
MOVX    A,@DPTR        ;LEE DATO
CJNE    A,#067H,DEFECD ;SI EL DATO LEIDO NO ES IGUAL AL CODIGO DE
AVISO
LJMP    FINDIR        ;INICIALIZA CON LOS VALORES BASES DE INICIO
DEFECD:                ;DE AREA DE DATOS
MOV     DPTR,#20       ;APUNTA DIRECCIÀN DE DPL ELECTRICO
MOV     A,#0ADH        ;CARGA DATO DPL BASE DE AREA DE DATOS TRACCIÀN
E
MOVX    @DPTR,A        ;GUARDA DATO BASE DE DPL E
INC     DPTR           ;APUNTA DIRECCIÀN DPH ELECTRICO
MOV     A,#0DH         ;CARGA DATO DPH BASE DE AREA DE DATOS TRACCIÀN
E
MOVX    @DPTR,A        ;GUARDA DATO BASE DPH EN RAM EXTERNA
INC     DPTR           ;APUNTA DIRECCIÀN DPL DIESEL
MOV     A,#0A8H        ;CARGA DATO DPL MAXIMO ELECTRICO
MOVX    @DPTR,A        ;GUARDA DATO DPL MAXIMO ELECTRICO
INC     DPTR           ;APUNTA DIRECCIÀN DPH MAXIMO
MOV     A,#061H        ;CARGA DATO DPH MAXIMO ELECTRICO
MOVX    @DPTR,A        ;GUARDA DATO DPH MAXIMO DE TRACCIÀN ELECTRICO
INC     DPTR           ;APUNTA DIRECCIÀN DPL BASE TRACCIÀN DIESEL
MOV     A,#0A9H        ;CARGA DATO DPL BASE TRACCIÀN DIESEL
MOVX    @DPTR,A        ;GUARAD DPL BASE TRACCIÀN DIESEL
INC     DPTR           ;APUNTA DIRECIÀN DPH BASE TRACCIÀN DIESEL
MOV     A,#061H        ;CARGA DATO BASE DPH TRACCIÀN DIESEL
MOVX    @DPTR,A        ;GUARDA DPH BASE TRACCIÀN DIESEL
INC     DPTR           ;APUNTA DIRECCIÀN DPL MAXIMO TRACCIÀN DIESEL
MOV     A,#030H        ;CARGA VALOR MAXIMO DPL TRACCIÀN DIESEL
MOVX    @DPTR,A        ;GUARDA VALOR MAXIMO DPL TRACCIÀN DIESEL
INC     DPTR           ;APUNTA DIRECCIÀN DPH MAXIMO TRACCIÀN DIESEL
MOV     A,#075H        ;CARGA VALOR MAXIMO DPH TRACCIÀN DIESEL
MOVX    @DPTR,A        ;GUARDA VALOR MAXIMO DPH TRACCIÀN DIESEL
INC     DPTR           ;APUNTA DIRECCIÀN DPL BASE DESCARRILO
MOV     A,#031H        ;CARGA VALOR BASE DPL DESCARRILO
MOVX    @DPTR,A        ;GUARDA VALOR BASE DPL DESCARRILO
INC     DPTR           ;APUNTA DIRECCIÀN BASE DPH DESCARRILO
MOV     A,#075H        ;CARGA VALOR BASE DPH DESCARRILO
MOVX    @DPTR,A        ;GUARADA VALOR BASE DPH DESCARRILO
INC     DPTR           ;APUNTA DIRECCIÀN DPL MAXIMO DESCARRILO
MOV     A,#0F8H        ;CARGA VALOR MAXIMO DPL DESCARRILO
MOVX    @DPTR,A        ;GUARDA VALOR DPL MAXIMO DESCARRILO
INC     DPTR           ;APUNTA VALOR MAXIMO DPH DESCARRILO
MOV     A,#07FH        ;CARGA VALOR MAXIMO DPH DESCARRILO
MOVX    @DPTR,A        ;GUARDA VALOR MAXIMO DPH DESCARRILO
INC     DPTR           ;APUNTA DIRECCIÀN DPL BASE INTERVALOS ELEC
MOV     A,#065H        ;CARGA VALOR BASE DPL INTERVALOS ELECTRICO
MOVX    @DPTR,A        ;GUARDA VALOR BASE DPLINTERVALOS ELECTRICO
INC     DPTR           ;APUNTA DIRECCIÀN BASE DPH INTERVALOS ELECTRICO
MOV     A,#00H         ;CARGA VALOR BASE DPH INTERVALOS ELECTRICO
MOVX    @DPTR,A        ;GUARADA VALOR BASE DPH INTERVALOS ELECTRICO
INC     DPTR           ;APUNTA DIRECCIÀN DPL MAXIMO INTERVALOS
ELECTRICO
MOV     A,#0DCH        ;CARGA VALOR MAXIMO DPL INTERVALOS ELECTRICO
MOVX    @DPTR,A        ;GUARDA VALOR DPL MAXIMO INTERVALOS ELECTRICO
INC     DPTR           ;APUNTA VALOR MAXIMO DPH INTERVALOS ELECTRICO
MOV     A,#05H        ;CARGA VALOR MAXIMO DPH INTERVALOS ELECTRICO

```



```

MOVX    @DPTR,A        ;GUARDA VALOR MAXIMO DPH INTERVALOS ELECTRICO
INC     DPTR           ;APUNTA DIRECCIÀN DPL BASE INTERVALOS DIESEL
MOV     A,#0DDH        ;CARGA VALOR BASE DPL INTERVALOS DIESEL
MOVX    @DPTR,A        ;GUARDA VALOR BASE DPLINTERVALOS DIESEL
INC     DPTR           ;APUNTA DIRECCIÀN BASE DPH INTERVALOS DIESEL
MOV     A,#05H        ;CARGA VALOR BASE DPH INTERVALOS DIESEL
MOVX    @DPTR,A        ;GUARADA VALOR BASE DPH INTERVALOS DIESEL
INC     DPTR           ;APUNTA DIRECCIÀN DPL MAXIMO INTERVALOS DIESEL
MOV     A,#0ACH        ;CARGA VALOR MAXIMO DPL INTERVALOS DIESEL
MOVX    @DPTR,A        ;GUARDA VALOR DPL MAXIMO INTERVALOS DIESEL
INC     DPTR           ;APUNTA VALOR MAXIMO DPH INTERVALOS DIESEL
MOV     A,#0DH        ;CARGA VALOR MAXIMO DPH INTERVALOS DIESEL
MOVX    @DPTR,A        ;GUARDA VALOR MAXIMO DPH INTERVALOS DIESEL
LCALL   BANCO2
MOV     R0,#00
MOV     R1,#00
MOV     R2,#00
MOV     R3,#00
MOV     R4,#00
MOV     R5,#00
MOV     R6,#00
MOV     R7,#00
FINDIR:
RET

```

;SUBROUTINA DE INICIALIZACIÀN DE VALORES POR DEFECTO DEL RELOJ

```

RELOJO:
MOV     DPTR,#060H      ;APUNTA DIRECCIÀN DE CODIGO DE AVISO
MOVX    A,@DPTR        ;LEE CODIGO
CJNE    A,#067H,DEFEC  ;COMPARA CODIGO DE AVISO
LJMP    SALRE          ;SI EL CODIGO ES IGUAL NO INICIALIZA EL RELOJ
DEFEC:
LCALL   BANCO1         ;SELECCIONA BANCO DE REGISTROS UNO
MOV     R0,#00H        ;SEGUNDOS IGUAL A CERO
MOV     R1,#00H        ;MINUTOS IGUAL A CERO
MOV     R2,#00H        ;HORAS GUAL A CERO
MOV     R3,#01H        ;DIA IGUAL A UNO
MOV     R4,#01H        ;MES IGUAL A UNO
MOV     R5,#062H       ;AÑO IGUAL A 1998
MOV     R6,#03H        ;TIEMPO DE MUESTREO EN SEGUNDOS IGUAL A 3
MOV     R7,#05H        ;NUMERO DE TROLEBUS
LCALL   RERAM          ;GUARDA ESTOS VALORES EN RAM EXTERNA
SALRE:
RET

```

;SUBROUTINA PARA GUARDAR LOS DATOS DEL RELOJ EN MEMORIA RAM EXTERNA  
;EN CASO DE UN RESET DEL MICROPROCESADOR EL TIEMPO DEL RELOJ NO SE VE  
;AFECTADO

```

RERAM:
LCALL   BANCO1         ;LLAMA BANCO DE REGISTROS 1
MOV     DPTR,#0000H    ;APUNTA DIRECCIÀN SEGUNDOS
MOV     A,R0           ;LEE SEGUNDOS
MOVX    @DPTR,A        ;GUARDA SEGUNDOS
INC     DPTR           ;APUNTA DIRECCIÀN MINUTOS
MOV     A,R1           ;LEE MINUTOS
MOVX    @DPTR,A        ;GUARDA MINUTOS
INC     DPTR           ;APUNTA DIRECCIÀN HORAS
MOV     A,R2           ;LEE HORAS

```





```

MOVX   @DPTR,A           ;GUARDA HORAS
INC    DPTR              ;APUNTA DIRECCIÒN DIA
MOV    A,R3              ;LEE DIA
MOVX   @DPTR,A           ;GUARDA DIA
INC    DPTR              ;APUNTA DIRECCIÒN MES
MOV    A,R4              ;LEE MES
MOVX   @DPTR,A           ;GUARDA MES
INC    DPTR              ;APUNTA DIRECCIÒN AÑO
MOV    A,R5              ;LEE AÑO
MOVX   @DPTR,A           ;GUARDA AÑO
INC    DPTR              ;APUNTA DIRECCIÒN TIEMPO DE MUESTREO
MOV    A,R6              ;LEE TIEMPO DE MUESTREO
MOVX   @DPTR,A           ;GUARDA TIEMPO DE MUESTREO
INC    DPTR              ;APUNTA DIRECCIÒN NÚMRO DE TROLEBÉS
MOV    A,R7              ;LEE NÚMRO DE TROLEBÉS
MOVX   @DPTR,A           ;GUARDA NÚMRO DE TROLEBÉS
LCALL  BANCO0            ;LLAMA BANCO DE REGISTRO CERO
RET

```

```

;SUBROUTINA PARA RETARDAR MEDIO SEGUNDO EN TIEMPO REAL
;UTILIZA EL TIMER CERO EN MODO DE AUTORECARGA

```

```

MEDS:
MOV    TL0,#012H         ;VALOR PARA RETARDO DE 0.14 MICROSEGUNDOS
MOV    TH0,#012H         ;VALOR DE AUTORECARGA
MOV    R0,#010H         ;CONTADOR 16
MOV    R1,#0A0H         ;CONTADOR 160
MOV    R2,#07H          ;CONTADOR 7
ETQ1:
JNB    TF0,$            ;DETECTA DESBORDAMIENTO TIMER CERO
CLR    TF0              ;LIMPIA BANDERA DE DESBORADMIENTO
DJNZ   R1,ETQ1          ;CUENTA 160 VECES 0.14 MICROSEGUNDOS
MOV    R1,#0AH          ;RECARGA CONTADOR
DJNZ   R0,ETQ1          ;CUENTA 16 VECES 22.4 MICROSEGUNDOS
MOV    R0,#010H         ;RECARGA CONTADOR
DJNZ   R2,ETQ1          ;CUENTA 7 VECES 358.4 MICROSEGUNDOS
MOV    R2,#07H          ;RECARGA CONTADOR
RET

```

```

;INTERRUPCIÒN CERO, CUENTA EL NÚMRO DE PULSOS NEGATIVOS QUE SE
;PRODUCEN (MIDE FRECUENCIA)

```

```

INTER0:
INC    R3                ;INCREMENTA EN CADA PULSO EL REGISTRO 3
RETI

```

```

;SUBROUTINA PARA SELECCIÒNAR EL BANCO DE REGISTROS CERO

```

```

BANCO0:
CLR    RS0
CLR    RS1
RET

```

```

;SUBROUTINA PARA SELECCIÒNAR EL BANCO DE REGISTROS UNO

```

```

BANCO1:
SETB   RS0
CLR    RS1
RET

```



; SUBROUTINA PARA SELECCIONAR EL BANCO DE REGISTROS DOS

BANCO2:

CLR RS0

SETB RS1

RET

; SUBROUTINA PARA SELECCIONAR EL BANCO DE REGISTROS TRES

BANCO3:

SETB RS0

SETB RS1

RET

END□



***ANEXO***

***3B***

***PROGRAMA DE  
ADMINISTRACIÓN EN EL PC.***



' PROGRAMA DE ADMINISTRACIÓN DE DATOS PARA EL PC.  
VENTANA PRINCIPAL

COMANDOS DEL SUBMENU O BOTON ABRIR .

```
Sub abrir_Click ()
portada.Visible = False
cmdialog1.Filter = "Todos los archivos(*.*)|*.*|Archivos de datos(*.dat)|*.dat"
cmdialog1.FilterIndex = 2
cmdialog1.Action = 1
Filename = cmdialog1.FileName
openfile (Filename)
txtEdit.Visible = True
cerrar.Enabled = True
guardar.Enabled = True
grafico.Enabled = True
  mnugrafico.Enabled = True
  mnucerrar.Enabled = True
  mnuguardar.Enabled = True
End Sub
```

COMANDOS DEL SUBMENU O BOTON CERRAR .

```
Sub cerrar_Click ()
closeFile (Filename)
txtEdit.Visible = False
cerrar.Enabled = False
guardar.Enabled = False
  mnucerrar.Enabled = False
  mnuguardar.Enabled = False
  mnugrafico.Enabled = False
portada.Visible = True
End Sub
```

COMANDOS DEL SUBMENU O BOTON RECIBIR DATOS .

```
Sub datos_Click ()
DATO.Visible = True
Unload Me
End Sub
```

COMANDOS DEL SUBMENU O BOTON DESCARRILAMIENTO.

```
Sub desc_Click ()
DESCA.Visible = True
Unload Me
End Sub
```

COMANDOS DE LA VENTANA PRINCIPAL .

```
Sub Form_Load ()
  ChDir app.Path
  ChDrive app.Path
  txtEdit.Move 0, 750
  Top = Screen.Height / 2 - Height / 2
  Left = Screen.Width / 2 - Width / 2
End Sub
```





## COMANDOS DEL SUBMENU O BOTON GRAFICO.

```
Sub grafico_Click ()
graficos.Visible = True
Unload Me
End Sub
```

## COMANDOS DEL SUBMENU O BOTON GUARDAR COMO.

```
Sub guardar_Click ()
NIVE.Caption = "Archivo de Datos: Untitled"
cmdialog1.Filter = "Todos los archivos(*.*)|*. *|Archivos de datos(*.dat)|*.dat"
cmdialog1.FilterIndex = 2
cmdialog1.Action = 2
Filename = cmdialog1.FileName
closeFile (Filename)
End Sub
```

## COMANDOS PARA MOVIMIENTO DEL RATON SOBRA CADA BOTON.

```
Sub mens1_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
mens1.Visible = False
mens2.Visible = False
mens3.Visible = False
mens4.Visible = False
mens5.Visible = False
mens6.Visible = False
mens7.Visible = False
mens8.Visible = False
End Sub
```

## COMANDOS PARA MOVIMIENTO DEL RATON SOBRA CADA BOTON.

```
Sub mens10_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
mens1.Visible = False
mens2.Visible = False
mens3.Visible = False
mens4.Visible = False
mens5.Visible = False
mens6.Visible = False
mens7.Visible = False
mens8.Visible = False
mens9.Visible = False
mens10.Visible = False
End Sub
```

## COMANDOS PARA MOVIMIENTO DEL RATON SOBRA CADA BOTON.

```
Sub mens2_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
mens1.Visible = False
mens2.Visible = False
mens3.Visible = False
mens4.Visible = False
mens5.Visible = False
mens6.Visible = False
mens7.Visible = False
mens8.Visible = False
End Sub
```



## COMANDOS PARA MOVIMIENTO DEL RATON SOBRA CADA BOTON.

```

Sub mens3_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
mens1.Visible = False
mens2.Visible = False
mens3.Visible = False
mens4.Visible = False
mens5.Visible = False
mens6.Visible = False
mens7.Visible = False
mens8.Visible = False
End Sub

```

## COMANDOS PARA EL MENU O BOTON DE COMANDO ABRIR.

```

Sub mnuabrir_Click ()
portada.Visible = False
txtEdit.Visible = True
cmdialog1.Filter = "Todos los archivos (*.*)|*.*)|Archivos de datos (*.dat)|*.dat"
cmdialog1.FilterIndex = 2
cmdialog1.Action = 1
Filename = cmdialog1.Filename
openfile (Filename)
cerrar.Enabled = True
guardar.Enabled = True
  mnucerrar.Enabled = True
  mnuguardar.Enabled = True
grafico.Enabled = True
  mnugrafico.Enabled = True
End Sub

```

## COMANDOS PARA EL SUBMENU ACERCA DE.

```

Sub mnuacerca_Click ()
acerca.Visible = True
End Sub

```

## COMANDOS PARA EL SUBMENU O BOTON SALIR

```

Sub salir_Click ()
End
End Sub

```

`VENTANA DE RECEPCIÓN DE DATOS DESDE EL MODULO AL PC.  
COMANDO DEL BOTON ADQUIRIR.

```

Sub adquirir_Click ()
timer1.Enabled = True
If ADQUI1.Value = True Then
Open archivo.Text For Output As #1
Print #1, "FECHA Y HORA DE INICIO DE LA ADQUISICIÓN DE DATOS"
Print #1, Now
Print #1, "VELOCIDAD EN Km/h"
Else
End If
End Sub

```



## COMANDO DEL BOTON ATRÁS.

```

Sub atras_Click ()
Unload Me
nive.Visible = True
End Sub

```

## COMANDOS DEL BOTON SALIR

```

Sub exit_Click ()
End
End Sub

```

## COMANDOS PARA EL MOVIMIENTO DEL RATO SOBRE LOS BOTONES DE COMANDO.

```

Sub Form_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
mens1.Visible = False
mens2.Visible = False
mens3.Visible = False
mens4.Visible = False
End Sub

```

## COMANDOS PARA EL MOVIMIENTO DEL RATO SOBRE LOS BOTONES DE COMANDO.

```

Sub mens1_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
mens1.Visible = False
mens2.Visible = False
mens3.Visible = False
mens4.Visible = False
End Sub

```

## COMANDOS PARA EL MOVIMIENTO DEL RATO SOBRE LOS BOTONES DE COMANDO.

```

Sub stop_Click ()
mens1.Visible = False
mens2.Visible = False
mens3.Visible = False
mens4.Visible = False
timer1.Enabled = False
End Sub

```

## COMANDO DEL TIMER 1 QUE REALIZA LA COMUNICACIÓN CON EL MODULO

```

Sub Timer1_Timer ()
mens1.Visible = False
mens2.Visible = False
mens3.Visible = False
mens4.Visible = False
I = 0
K = 150
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(K)
If ADQUI1.Value = True Then
Print #1, "FECHA Y HORA FINAL DE LA ADQUISICIÓN DE DATOS"
Print #1, ; Now
Close #1
Else

```



```

End If
Do
Loop While COMM1.InBufferCount = 0
M1 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M2 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M3 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M4 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M5 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M6 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M7 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M8 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M9 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M10 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M11 = Asc(COMM1.Input)
If M11 = 204 Then
TEXT11.Text = "TRACCIÓN ELECTRICO"
Else
End If
If M11 = 209 Then
TEXT11.Text = "TRACCIÓN DIESEL"
Else
End If
If M11 = 200 Then
TEXT11.Text = "CERO"
Else
End If
If M11 = 203 Then
TEXT11.Text = "DESCARRILAMIENTO!"
Else
End If
TEXT1.Text = M1
TEXT2.Text = M2
TEXT3.Text = M3
TEXT4.Text = M4
TEXT5.Text = M5
TEXT6.Text = M6
TEXT7.Text = M8
TEXT9.Text = M7
ve = M10 / 2.18
taco.Text = ve

```





```

vel1.Value = ve
If ADQUI1.Value = True Then
Print #1, ; ve
Else
End If
COMM1.PortOpen = False
End Sub

```

COMANDOS PARA EL BOTON BORRAR DATOS DEL MODULO.

```

Sub adquirir_Click ()
BORRA.Enabled = False
BORRA.Visible = False
timer14.Enabled = True
TEXT1.Visible = True
text5.Text = "ESPERE POR FAVOR..."
End Sub

```

COMANDO DEL BOTON ATRÁS.

```

Sub atras_Click ()
NIVE.Visible = True
Unload Me
End Sub

```

COMANDOS PARA EL MOVIMIENTO DEL RATO SOBRE LOS BOTONES DE COMANDO.

```

Sub Form_MouseMove (Button As Integer, Shift As Integer, X As Single, Y As Single)
mens4.Visible = False
mens3.Visible = False
mens1.Visible = False
TEXT6.Visible = False
End Sub

```

COMANDO DEL BOTON SALIR

```

Sub salir_Click ()
End
End Sub

```

COMANDO DEL TIMER 2 QUE REALIZA LA COMUNICACIÓN CON EL MODULO

```

Sub Timer6_Timer ()
k = 216
Open "c:\dat\valor.dat" For Input As #2
Input #2, kon
Close #2
mmm = 2760
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(k)
ARCHIVO.Text = "C:\dat\DAT6.DAT"
Open ARCHIVO.Text For Output As #1
For l = 1 To mmm
Do
Loop While COMM1.InBufferCount = 0
des = Asc(COMM1.Input)
If des = 255 Then
GoTo sale8

```



```

Else
cal = cal + 1
ve = cal * 100 / kon
Print #1, des
taco.Text = ve
vel1.Value = ve
COMM1.Output = Chr$(des)
End If
Next I
sale8:
BORRAR.Visible = True
COMM1.PortOpen = False
Close #1
text5.Text = "LISTO!"
TEXT4.Text = Now
BORRA.Enabled = True
BORRA.Visible = True
kon = kon - cal
Open "c:\dat\valor.dat" For Output As #3
Print #3, kon
Close #3
timer6.Enabled = False
End Sub

```

#### COMANDO DEL TIMER 3 QUE REALIZA LA COMUNICACIÓN CON EL MODULO

```

Sub Timer7_Timer ()
k = 215
Open "c:\dat\valor.dat" For Input As #2
Input #2, kon
Close #2
mmm = 5000
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(k)
ARCHIVO.Text = "C:\dat\DAT5.DAT"
Open ARCHIVO.Text For Output As #1
For I = 1 To mmm
Do
Loop While COMM1.InBufferCount = 0
des = Asc(COMM1.Input)
If des = 255 Then
GoTo sale7
Else
cal = cal + 1
ve = cal * 100 / kon
Print #1, Format$(des / 2.18, "#,##0.00")
taco.Text = ve
vel1.Value = ve
COMM1.Output = Chr$(des)
End If
Next I
sale7:
Close #1
COMM1.PortOpen = False
Close #1
kon = kon - cal
Open "c:\dat\valor.dat" For Output As #3
Print #3, kon

```



```

Close #3
timer7.Enabled = False
timer6.Enabled = True
End Sub

```

#### COMANDO DEL TIMER 4 QUE REALIZA LA COMUNICACIÓN CON EL MODULO

```

Sub Timer10_Timer ()
k = 211
Open "c:\dat\valor.dat" For Input As #2
Input #2, kon
Close #2
mmm = 5375
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(k)
ARCHIVO.Text = "C:\dat\DAT2.DAT"
Open ARCHIVO.Text For Output As #1
For l = 1 To mmm
Do
Loop While COMM1.InBufferCount = 0
des = Asc(COMM1.Input)
If des = 255 Then
GoTo sale4
Else
cal = cal + 1
ve = cal * 100 / kon
Print #1, Format$(des / 2.18, "#,##0.00")
taco.Text = ve
vel1.Value = ve
COMM1.Output = Chr$(des)
End If
Next l
sale4:
COMM1.PortOpen = False
Close #1
kon = kon - cal
Open "c:\dat\valor.dat" For Output As #3
Print #3, kon
Close #3
timer10.Enabled = False
timer9.Enabled = True
End Sub

```

#### COMANDO DEL TIMER 5 QUE REALIZA LA COMUNICACIÓN CON EL MODULO

```

Sub Timer11_Timer ()
k = 210
Open "c:\dat\valor.dat" For Input As #2
Input #2, kon
Close #2
mmm = 5375
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(k)
ARCHIVO.Text = "C:\dat\DAT1.DAT"
Open ARCHIVO.Text For Output As #1
For l = 1 To mmm

```

```

Do
Loop While COMM1.InBufferCount = 0
des = Asc(COMM1.Input)
If des = 255 Then
GoTo sale3
Else
cal = cal + 1
ve = cal * 100 / kon
Print #1, Format$(des / 2.18, "#,##0.00")
taco.Text = ve
vel1.Value = ve
COMM1.Output = Chr$(des)
End If
Next I
sale3:
COMM1.PortOpen = False
Close #1
kon = kon - cal
Open "c:\dat\valor.dat" For Output As #3
Print #3, kon
Close #3
timer11.Enabled = False
timer10.Enabled = True
End Sub

```

#### COMANDO DEL TIMER 6 QUE REALIZA LA COMUNICACIÓN CON EL MODULO

```

Sub Timer12_Timer ()
k = 208
mmm = 2000
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(k)
ARCHIVO.Text = "C:\dat\DIESEL.DAT"
Open ARCHIVO.Text For Output As #1
For I = 1 To mmm
Do
Loop While COMM1.InBufferCount = 0
des = Asc(COMM1.Input)
If des = 255 Then
GoTo sale2
Else
Print #1, des
COMM1.Output = Chr$(des)
End If
Next I
sale2:
COMM1.PortOpen = False
Close #1
timer12.Enabled = False
timer11.Enabled = True
End Sub

```

#### COMANDO DEL TIMER 7 QUE REALIZA LA COMUNICACIÓN CON EL MODULO

```

Sub Timer13_Timer ()
k = 209
mmm = 1500
COMM1.CommPort = 1

```

```

COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(k)
ARCHIVO.Text = "C:\dat\ELECT.DAT"
Open ARCHIVO.Text For Output As #1
For l = 1 To mmm
Do
Loop While COMM1.InBufferCount = 0
des = Asc(COMM1.Input)
If des = 255 Then
GoTo sale1
Else
Print #1, des
COMM1.Output = Chr$(des)
End If
Next l
sale1:
COMM1.PortOpen = False
Close #1
timer13.Enabled = False
timer12.Enabled = True
End Sub

```

#### COMANDO DEL TIMER 8 QUE REALIZA LA COMUNICACIÓN CON EL MODULO

```

Sub Timer14_Timer ()
text5.Text = "ESPERE POR FAVOR..."
k = 217
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(k)
ARCHIVO.Text = "C:\dat\total.DAT"
Open ARCHIVO.Text For Output As #1
Do
Loop While COMM1.InBufferCount = 0
DES1 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
DES2 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
DES3 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
DES4 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
des5 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
des6 = Asc(COMM1.Input)
Print #1, DES1
Print #1, DES2
Print #1, DES3
Print #1, DES4
Print #1, des5
Print #1, des6
COMM1.PortOpen = False
Close #1

```



```

kon = DES1 + DES2 * 250 + DES3 + DES4 * 50
cal = 0
Open "c:\dat\valor.dat" For Output As #2
Print #2, kon
Close #2
timer14.Enabled = False
timer13.Enabled = True
End Sub

```

#### COMANDOS DE LA VENTANA DE IGUALAR RELOJ

```

Sub Command3D1_Click ()
k = 214
mmm = 2760
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(k)
COMM1.Output = Chr$(k)
COMM1.Output = Chr$(k)
COMM1.Output = Chr$(k)
COMM1.Output = Chr$(k)
COMM1.PortOpen = False
Unload Me
End Sub
Sub Command3D2_Click ()
Unload Me
End Sub
Sub atras_Click ()
Unload Me
nive.Visible = True
End Sub

```

#### COMANDO DEL BOTON CAMBIAR

```

Sub cambiar_Click ()
timer1.Enabled = True
cambiar.Enabled = False
atras.Enabled = False
salir.Enabled = False
text16.Text = "ESPERE POR FAVOR"
End Sub

```

#### COMANDO DEL BOTON SALIR

```

Sub salir_Click ()
End
End Sub

```

#### COMANDOS DEL TIMER QUE REALIZA LA COMUNICACIÓN CON EL MODULO.

```

Sub Timer1_Timer ()
KV = 152
TEXT1.Text = "MINUTOS"
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(KV)
Do

```

```

Loop While COMM1.InBufferCount = 0
M1 = Asc(COMM1.Input)
If M1 = KV Then
TEXT2.Text = "CORRECTA"
Else
TEXT2.Text = "INCORRECTA"
End If
COMM1.PortOpen = False
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
MINU = Val(TEXT4.Text)
COMM1.Output = Chr$(MINU)
COMM1.PortOpen = False
TEXT2.Text = ""
KV = 153
TEXT1.Text = "HORAS"
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(KV)
Do
Loop While COMM1.InBufferCount = 0
M1 = Asc(COMM1.Input)
If M1 = KV Then
TEXT2.Text = "CORRECTA"
Else
TEXT2.Text = "INCORRECTA"
End If
COMM1.PortOpen = False
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
HORA = Val(TEXT5.Text)
COMM1.Output = Chr$(HORA)
COMM1.PortOpen = False
TEXT2.Text = ""
KV = 154
TEXT1.Text = "DIA"
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(KV)
Do
Loop While COMM1.InBufferCount = 0
M1 = Asc(COMM1.Input)
If M1 = KV Then
TEXT2.Text = "CORRECTA"
Else
TEXT2.Text = "INCORRECTA"
End If
COMM1.PortOpen = False
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
DIA = Val(TEXT8.Text)
COMM1.Output = Chr$(DIA)
COMM1.PortOpen = False
TEXT2.Text = ""
KV = 155

```

```

TEXT1.Text = "MES"
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(KV)
Do
Loop While COMM1.InBufferCount = 0
M1 = Asc(COMM1.Input)
If M1 = KV Then
TEXT2.Text = "CORRECTA"
Else
TEXT2.Text = "INCORRECTA"
End If
COMM1.PortOpen = False
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
MES = Val(TEXT7.Text)
COMM1.Output = Chr$(MES)
COMM1.PortOpen = False
TEXT2.Text = ""
KV = 156
TEXT1.Text = "AÑO"
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(KV)
Do
Loop While COMM1.InBufferCount = 0
M1 = Asc(COMM1.Input)
If M1 = KV Then
TEXT2.Text = "CORRECTA"
Else
TEXT2.Text = "INCORRECTA"
End If
COMM1.PortOpen = False
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
ANO = Val(TEXT6.Text)
COMM1.Output = Chr$(ANO)
COMM1.PortOpen = False
timer1.Enabled = False
TIMER2.Enabled = True
cambiar.Enabled = True
atras.Enabled = True
salir.Enabled = True
text16.Text = " LISTO"
End Sub

```

#### COMANDOS DE COMUNICACIÓN CON EL MODULO

```

Sub Timer2_Timer ()
k = 150
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(k)
Do

```

```

Loop While COMM1.InBufferCount = 0
M1 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M2 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M3 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M4 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M5 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M6 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M7 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M8 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M9 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M10 = Asc(COMM1.Input)
TEXT10.Text = M1
TEXT11.Text = M2
TEXT12.Text = M3
TEXT13.Text = M4
TEXT14.Text = M5
TEXT15.Text = M6
TEXT3.Text = M1
TEXT4.Text = M2
TEXT5.Text = M3
TEXT8.Text = M4
TEXT7.Text = M5
TEXT6.Text = M6
TEXT2.Text = "LISTO"
TIMER2.Enabled = False
fin1:
COMM1.PortOpen = False
cambiar.Enabled = True
atras.Enabled = True
salir.Enabled = True
text16.Text = " LISTO"
End Sub

```

#### COMANDOS DEL BOTON Atrás

```

Sub atras_Click ()
Unload Me
nive.Visible = True
End Sub

```

#### COMANDOS BOTON CAMBIAR

```

Sub cambiar_Click ()
timer1.Enabled = True
TEXT7.Text = "ESPERE POR FAVOR"
End Sub

```

#### COMANDO DEL BOTON SALIR

```

Sub salir_Click ()
End
End Sub

```

#### COMANDOS DE COMUNICACIÓN CON EL MODULO

```

Sub Timer1_Timer ()
Kv = 157
TEXT5.Text = "TIEMPO DE MUESTREO"
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(Kv)
Do
Loop While COMM1.InBufferCount = 0
H$ = COMM1.Input
M1 = Asc(H$)
If M1 = Kv Then
TEXT6.Text = "CORRECTA"
Else
TEXT6.Text = "INCORRECTA"
End If
COMM1.PortOpen = False
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
TM = Val(TEXT4.Text)
COMM1.Output = Chr$(TM)
COMM1.PortOpen = False
Kv = 158
TEXT6.Text = ""
TEXT5.Text = "NUMERO DE TROLE"
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
I = 0
COMM1.Output = Chr$(Kv)
Do
Loop While COMM1.InBufferCount = 0
H$ = COMM1.Input
M1 = Asc(H$)
If M1 = Kv Then
TEXT6.Text = "CORRECTA"
Else
TEXT6.Text = "INCORRECTA"
End If
COMM1.PortOpen = False
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
TM = Val(TEXT3.Text)
COMM1.Output = Chr$(TM)
timer2.Enabled = True

```

```

timer1.Enabled = False
fin:
COMM1.PortOpen = False
End Sub

```

#### COMANDOS DE COMUNICACIÓN CON EL MODULO

```

Sub Timer2_Timer ()
k = 150
COMM1.CommPort = 1
COMM1.Settings = "2400,N,8,1"
COMM1.PortOpen = True
COMM1.Output = Chr$(k)
Do
Loop While COMM1.InBufferCount = 0
M1 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M2 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M3 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M4 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M5 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M6 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M7 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M8 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M9 = Asc(COMM1.Input)
Do
Loop While COMM1.InBufferCount = 0
M10 = Asc(COMM1.Input)
TEXT1.Text = M8
TEXT2.Text = M7
TEXT3.Text = M8
TEXT4.Text = M7
TEXT6.Text = "LISTO"
timer2.Enabled = False
fin1:
COMM1.PortOpen = False
cambiar.Enabled = True
atras.Enabled = True
salir.Enabled = True
TEXT7.Text = " LISTO"
End Sub

```

VENTANA DE GRAFICOS.

COMANDOS DE DESPLAZAMIENTO DEL GRAFICO HACIA LA IZQUIERDA O DERECHA.

```
Sub Barra_Change ()
On Error Resume Next
    Fondo.Left = -Barra.Value
End Sub
```

COMANDO BOTON Atrás

```
Sub Command3D1_Click ()
Unload Me
nive.Show
End Sub
```

COMANDO BOTON SALIR

```
Sub Command3D2_Click ()
End
End Sub
```

COMANDO CAMBIO DE DRIVE

```
Sub Drive_Change ()
    Lista.Path = Drive.Drive
End Sub
```

SUBROUTINA DE ERROR EN RECONOCIMIENTO DE ARCHIVOS.

```
Sub Form_Load ()
On Error Resume Next
    Lista.Path = app.Path
    Ubicacion = ""
    Barra.Value = 0
    Diferencia = Fondo.Width - Recuadro.Width
    If (Diferencia < .033 * Recuadro.Width) Then
        Barra.LargeChange = Diferencia
    Else
        Barra.LargeChange = Int(.33 * Recuadro.Width)
    End If
    Barra.SmallChange = Barra.LargeChange \ 10
    Barra.Max = Diferencia
End Sub
```

SUBROUTINA PARA GRAFICAR.

```
Sub grafica_Click ()
On Error GoTo Graf_Err
    Msg.Visible = False
    Err_Gr = False
    If (Ubicacion = "") Then
        MsgBox "Debe seleccionar el directorio donde se hallan los datos"
        Exit Sub
    End If
    'Leo archivo de Totales
    'Busco primer multiplo de 15
    Inix = 500
    Iniy = Fondo.Height - 500
```

```

Min = Val(Format(HIni, "nn"))
Hor = Val(Format(HIni, "hh"))
TInicio = Min
If (Min <= 15) Then
    Min = 15
Else
    If (Min <= 30) Then
        Min = 30
    Else
        If (Min <= 45) Then
            Min = 45
        Else
            If (Min > 45) Then
                Min = 60
            End If
        End If
    End If
End If
Msg.Visible = True
DoEvents
Fondo.BackColor = QBColor(1)
Fondo.ForeColor = QBColor(8)
Fondo.FontName = "Courier New"
Fondo.FontSize = 7
Fondo.Line (Inix, Iniy)-(Fondo.Width - 100, Iniy), QBColor(4)
Fondo.Line (Inix, 100)-(Inix, Iniy + 100), QBColor(4)
Txt = HIni
Fondo.CurrentX = Inix
k = 0
MeFin = CVDate(HFin)
'Escala del eje x
Escala = 1
If (TInicio > 0) Then
    TInicio = Min - TInicio
    TInicio = 15 - TInicio
End If
i = 1
Do While CVDate(Txt) <= MeFin
    If (k = 0) Then
        Fondo.CurrentY = Iniy + 200
        Fondo.Print Txt
        Min = Min - 1
        MinTmp = "00:01"
        Txt = Hor & ":" & Trim(Str(Min))
        Txt = (CVDate(MinTmp) + CVDate(Txt)) - CVDate("00:15")
    Else
        Fondo.CurrentY = Iniy + 100
        Fondo.Print Txt
    End If
    'If (i = 1) Then
    ' Fondo.Line (Inix + (TInicio * 60) * Escala, Iniy + 100)-(Inix + (TInicio * 60) * Escala, Iniy -
100)
    'Fondo.CurrentX = Inix + (TInicio * 60) * Escala
    'Else
    Fondo.Line (Inix + (i * 900) - (TInicio * 60) * Escala, Iniy + 100)-(Inix + (i * 900) - (TInicio
* 60) * Escala, Iniy - 100)
    Fondo.CurrentX = Inix + (i * 900) - (TInicio * 60) * Escala
    'End If
    Diff = "00:15"
    Sumt = CVDate(Txt) + CVDate(Diff)

```



```

    Txt = Sumt
    k = 1
    i = i + 1
    Loop
Fondo.CurrentY = Iniy + 100
Fondo.Print Txt
MeY = Iniy - 140
MeX = 50
k = -1
For i = 1 To 25
    Fondo.Line (Inix - MeX, MeY)-(Inix + MeX, MeY)
    If (MeX = 50) Then
        MeX = 100
        k = k + 1
        Fondo.CurrentX = Inix - 500
        Fondo.Print k * 10
    Else
        MeX = 50
    End If
    MeY = MeY - 140
Next i
'NumInt = Intervalo a graficar
'TieMue = Tiempo de Muestreo
If (TDie.Value) Then
    'Para diesel
    Dat5 = Ubicacion & "\dat5.dat"
    Open Dat5 For Input As #1
    If (Err_Gr) Then
        MsgBox "Error al abrir los archivos de trabajo"
        Msg.Visible = False
        Exit Sub
    End If
    'Registro desde el cual inicio la lectura de datos
    RegPos = Acumulador + 1
    k = 0
    Fila = ""
    i = 0
    Posx = 0
    NumMue = Val(Muestras.Caption)# de Muestras
    Escalay = (Iniy - 100) / 255
    Fondo.Line (Inix, Iniy)-(Inix, Iniy)
    Do While Not EOF(1)
        Char = Input(1, #1)
        If (Char <> Chr(10)) Then
            'Si no es fin de linea
            Fila = Fila & Char
        Else
            'Si es fin de linea
            i = i + 1
            If (i >= RegPos And k < NumMue) Then
                k = k + 1
                'Registros a graficarse
                Posy = Val(Fila)
                Fondo.Line -(Inix + Posx, Iniy - Posy * Escalay)
                Posx = Posx + TieMue * Escala
            End If
            Fila = ""
        End If
    Loop
Close #1

```

```

Else
'Para electrico
Dat1 = Ubicacion & "\dat1.dat"
Dat2 = Ubicacion & "\dat2.dat"
Dat3 = Ubicacion & "\dat3.dat"
Dat4 = Ubicacion & "\dat4.dat"
Open Dat1 For Input As #1
Open Dat2 For Input As #2
Open Dat3 For Input As #3
Open Dat4 For Input As #4
If (Err_Gr) Then
    MsgBox "Error al abrir los archivos de trabajo"
    Msg.Visible = False
    Exit Sub
End If
'Registro desde el cual inicio la lectura de datos
RegPos = Acumulador + 1
k = 0
Fila = ""
i = 0
Posx = 0
NumMue = Val(Muestras.Caption) '# de Muestras
*****
Escalay = (Iniy - 100) / 255
Fondo.Line (Inix, Iniy)-(Inix, Iniy)
Do While Not EOF(1)
    Char = Input(1, #1)
    If (Char <> Chr(10)) Then
        'Si no es fin de linea
        Fila = Fila & Char
    Else
        'Si es fin de linea
        i = i + 1
        If (i >= RegPos And k < NumMue) Then
            k = k + 1
            'Registros a graficarse
            Posy = Val(Fila)
            Fondo.Line -(Inix + Posx, Iniy - Posy * Escalay)
            Posx = Posx + TieMue * Escala
        End If
        Fila = ""
    End If
End While
Loop
Close #1
If (k < NumMue) Then
    Do While Not EOF(2)
        Char = Input(1, #2)
        If (Char <> Chr(10)) Then
            'Si no es fin de linea
            Fila = Fila & Char
        Else
            'Si es fin de linea
            i = i + 1
            If (i >= RegPos And k < NumMue) Then
                k = k + 1
                'Registros a graficarse
                Posy = Val(Fila)
                Fondo.Line -(Inix + Posx, Iniy - Posy * Escalay)
                Posx = Posx + TieMue * Escala
            End If
        End If
    End While
End If

```

```

        Fila = ""
    End If
    Loop
End If
Close #2
If (k < NumMue) Then
    Do While Not EOF(3)
        Char = Input(1, #3)
        If (Char <> Chr(10)) Then
            'Si no es fin de linea
            Fila = Fila & Char
        Else
            'Si es fin de linea
            i = i + 1
            If (i >= RegPos And k < NumMue) Then
                k = k + 1
                'Registros a graficarse
                Posy = Val(Fila)
                Fondo.Line -(Inix + Posx, Iniy - Posy * Escalay)
                Posx = Posx + TieMue * Escala
            End If
            Fila = ""
        End If
    Loop
End If
Close #3
If (k < NumMue) Then
    Do While Not EOF(4)
        Char = Input(1, #4)
        If (Char <> Chr(10)) Then
            'Si no es fin de linea
            Fila = Fila & Char
        Else
            'Si es fin de linea
            i = i + 1
            If (i >= RegPos And k < NumMue) Then
                k = k + 1
                'Registros a graficarse
                Posy = Val(Fila)
                Fondo.Line -(Inix + Posx, Iniy - Posy * Escalay)
                Posx = Posx + TieMue * Escala
            End If
            Fila = ""
        End If
    Loop
End If
Close #4
End If
Msg.Visible = False
Exit Sub
Graf_Err:
    Err_Gr = True
    Resume Next
End Sub
Sub Lista_Change ()
On Error GoTo Salir
    Error_Lectura = False
    Fondo.BackColor = QBColor(1)
    Fondo.ForeColor = QBColor(8)
    Msg.Visible = False

```

```

Ubicacion = Lista.Path
Nm.Text = ""
Nmd.Text = ""
Ni.Text = ""
Nid.Text = ""
TM.Text = ""
nt.Text = ""
Nro.Text = ""
Muestras.Caption = ""
Inicio.Caption = ""
Inicio.Visible = False
Fin.Caption = ""
Fin.Visible = False
If (Ubicacion = "") Then
    Exit Sub
End If
Total = Ubicacion & "\total.dat"
Elect = Ubicacion & "\elect.dat"
Diesel = Ubicacion & "\diesel.dat"
Open Total For Input As #1
If (Error_Lectura) Then
    Exit Sub
End If
Open Elect For Input As #2
If (Error_Lectura) Then
    Exit Sub
End If
Open Diesel For Input As #3
If (Error_Lectura) Then
    Exit Sub
End If
'Proceso Totales
Fila = ""
i = 0
Tempo = 0
Do While Not EOF(1)
    Char = Input(1, #1)
    if (Char <> Chr(10)) Then
        'Si no es fin de linea
        Fila = Fila & Char
    Else
        'Si es fin de linea
        i = i + 1
        Select Case i
            Case 1:
                Tempo = Val(Fila)
            Case 2:
                Nm.Text = Val(Fila) * 250 + Tempo
            Case 3:
                Tempo = Val(Fila)
            Case 4:
                Nmd.Text = Val(Fila) * 50 + Tempo
            Case 5:
                TM.Text = Val(Fila) & " seg."
                TieMue = Val(Fila)
            Case 6:
                nt.Text = Val(Fila)
        End Select
        Fila = ""
    End If
End If

```

```

Loop
Close #1
'Leer intervalos electricos
i = 0
Do While Not EOF(2)
  Char = Input(1, #2)
  If (Char = Chr(10)) Then
    'Si es fin de linea
    i = i + 1
  End If
Loop
Ni.Text = Int(i / 16)
NumE = Int(i / 16)
Close #2
If (i Mod 16 <> 0) Then
  MsgBox "Ultimo intervalo eléctrico incompleto"
End If
'Leer intervalos diesel
i = 0
Do While Not EOF(3)
  Char = Input(1, #3)
  If (Char = Chr(10)) Then
    'Si es fin de linea
    i = i + 1
  End If
Loop
Nid.Text = Int(i / 16)
NumD = Int(i / 16)
Close #3
If (i Mod 16 <> 0) Then
  MsgBox "Ultimo intervalo diesel incompleto"
End If
Exit Sub
Salir:
  Error_Lectura = True
  Resume Next
End Sub

```

COMANDOS DE DETECCIÓN MOVIMIENTO DEL CURSOR SOBRE CAJA DE TEXTO.

```

Sub Nro_KeyDown (KeyCode As Integer, Shift As Integer)
On Error Resume Next
  If (KeyCode = KEY_RETURN) Then
    Nro_LostFocus
  End If
End Sub

```

SUBROUTINA DE SELECCIÓN DE INTERVALO AGRAFICAR.

```

Sub TDie_Click (Value As Integer)
  Nro.Text = ""
  Muestras.Caption = ""
  Nro.SetFocus
End Sub

```

LECTURA DE VALOR DE INTERVALO

```

Sub TElec_Click (Value As Integer)
  Nro.Text = ""
  Muestras.Caption = ""

```



```
text1.Text = J  
timer1.Enabled = False  
End Sub
```