

Arieh Nachum

APPLIC-37
Manual del Usuario



Scientific Educational Systems

Arieh Nachum

APPLIC-37
Manual del Usuario

2^a edición española

(c) Copyright 2002 a Scientific Educational Systems Ltd.

Contenido

Capítulo 1 – Familiarización con APPLIC-37.....	1
1.1 Descripción general.....	1
1.2 Mapa de memoria del APPLIC-37	2
1.3 Canales de Entrada/Salida del APPLIC-37.....	3
1.4 Usos del APPLIC-37	5
1.5 Instalación de la interfaz	5
Capítulo 2 - Programación en SESLOGO	7
2.1 Instalación del programa.....	7
2.2 Comandos Directos	11
2.3 Programas y Procedimientos	23
2.4 Procedimientos como Nuevas Instrucciones.....	39
2.5 – Condiciones y Decisiones	45
Capítulo 3 - Programación en Ensamblador	55
3.1 Introducción	55
3.2 Controlador SES 8051	58
3.3 Operando el controlador con la PC.....	59
3.4 Escribir en ensamblar.....	63
Capítulo 4 – Programación en Lenguaje C	73
4.1 Fases de la Escritura de Programas	74
4.2 Lenguaje Máquina	77
4.3 Lenguaje Ensamblador	78
4.4 Lenguajes de Alto Nivel.....	79
4.5 Operando el controlador con la PC.....	79
4.6 Escribir en Lenguaje C.....	84

Capítulo 1 – Familiarización con APPLIC-37

1.1 Descripción general

APPLIC-37 es un módulo de sistema para el control Múltiple de E/S. El módulo incluye:

- Microcontrolador 89C52.
- Monitor de 8K ROM
- EEPROM de 8K.
- Circuito conmutador y decodificador.
- Interfaz serie para PC.
- Monitor Multi-función.

El módulo se conecta directamente vía RS232 con una PC IBM o compatible para la operación con la facilidad de pantalla completa, cargando y descargando desde los medios de almacenamiento magnéticos de la PC, y para la operación con el paquete Editor/Compilador/Depurador de pantalla completa.

El módulo se basa en el 89C52, miembro de la familia 8051 de microcontroladores, con las ventajas siguientes:

- Incluye 4 puertos programables.
- Incluye una RAM interna.
- Incluye 3 temporizadores internos.
- Incluye un UART para las comunicaciones RS232.
- Una gran cantidad de registros internos.

Características de funcionamiento:

- Programación en código de máquina o Ensamblador de un paso.
- Exhibición del desmontaje.
- Examina, cambia, agrega o cancela el contenido de la memoria.
- Funcionamiento con punto de interrupción o gradualmente.
- Lectura desde y escritura a los puertos.
- Ejecución de interrupciones.
- Uso de temporizadores.
- Examina el contenido del registro.
- Comunicaciones serie asíncronas.
- Conexión vía RS232 a una PC para la transferencia directa y la operación a través de un Editor/Compilador en Ensamblador y en C/Depurador, de pantalla completa.

Además de las características de programación, el módulo incluye las aplicaciones siguientes para E/S:

- 12V/2A fuente de energía.
- 8 salidas digitales con controladores de 0-12V/2.5A con indicación de LEDs.
- Al reemplazar la fuente de energía de 12V CD con la fuente de 25V CD cambia la salida digital a 0-25V/2.5A.
- 8 entradas digitales con indicación de LEDs.
- 1 salida convertidoras Digital a Análoga de 0-10V/1A.
- 8 entradas convertidoras Análoga a Digital de 0-5V.
- Todas las entradas y salidas tienen tomas de tipo banana.
- Conector para la expansión de E/S.

1.2 Mapa de memoria del APPLIC-37

FFFF	Rango de E/S
FF00 FEFF	
4000 3FFF	EEPROM
2000 1FFF	
0000	Monitor de programa - ROM

1.3 Canales de Entrada/Salida del APPLIC-37

APPLIC-37 viene acompañada por una fuente de energía de 12V/2A.

Los canales de entrada y los canales de salida están conectados a tomas banana de 2 mm.

Salida Digital Q0 – Q7

El puerto de salida está en el mapa de memoria en la dirección FF00H.

El voltaje de salida es 0V o 12V.

Una línea de tomas tipo banana de +12V (+V) está situada contra los terminales de entrada y salida digitales.

Al reemplazar la fuente de energía de 12V CD con la fuente de 25V CD cambia la línea +V y la salida digital a 25 V.

El canal de salida puede CONECTAR y DESCONECTAR un consumidor (una lámpara o un motor) conectado entre el terminal de salida y la línea de +V.

El bit '0' tiene salidas de 0V y CONECTA al consumidor con la salida de LED rojo (porque preferimos disminuir la corriente a la fuente).

Conectando un motor entre dos terminales de salida permite su CONEXION y DESCONEXION y también controlar la dirección de su rotación.

Salida de Relé AO1

APPLIC-37 cuenta con salidas de relé para consumidores de CA.

Entrada Digital J0 – J7

El puerto de entrada está en el mapa de memoria en la dirección FF00H.

Se debe conectar un sensor ON/OFF o un interruptor entre la línea +V y el canal de entrada.

Cuando se CONECTA el sensor, el LED verde se ENCIENDE y el puerto de salida lo lee como '0' (para adaptarlo al puerto de salida).

Salida Análoga AO2

Los canales de salida análoga se basan en un canal de dos DAC en serie (Convertidor Digital Analógico).

El rango de salida de cada canal es de 0-10V/1A.

Una línea de tomas tipo banana de tierra (GND) está situada contra los terminales de entrada y salida análogas.

El DAC es controlado por los bits del puerto 1 del 8051.

El Apéndice A incluye una rutina de muestra de los datos de salida para los canales DAC de salida.

El DAC controla un consumidor (un motor o una lámpara) conectada entre la salida análoga y la toma de tierra GND.

Conectar un motor entre un canal de salida análoga y un canal de salida digital permite controlar la velocidad del motor y también su rotación.

Entrada Análoga AI0 – AI7

Los canales de entrada análoga se basan en un canal de ocho ADC en serie (Convertidor Analógico Digital).

El rango de entrada de cada canal es de 0-5V. Los canales AI6 y AI7 incluyen resistores desplegados para sensores pasivos tales como los NTC, fototransistores o LDR.

El ADC es controlado por los bits del puerto 1 del 8051.

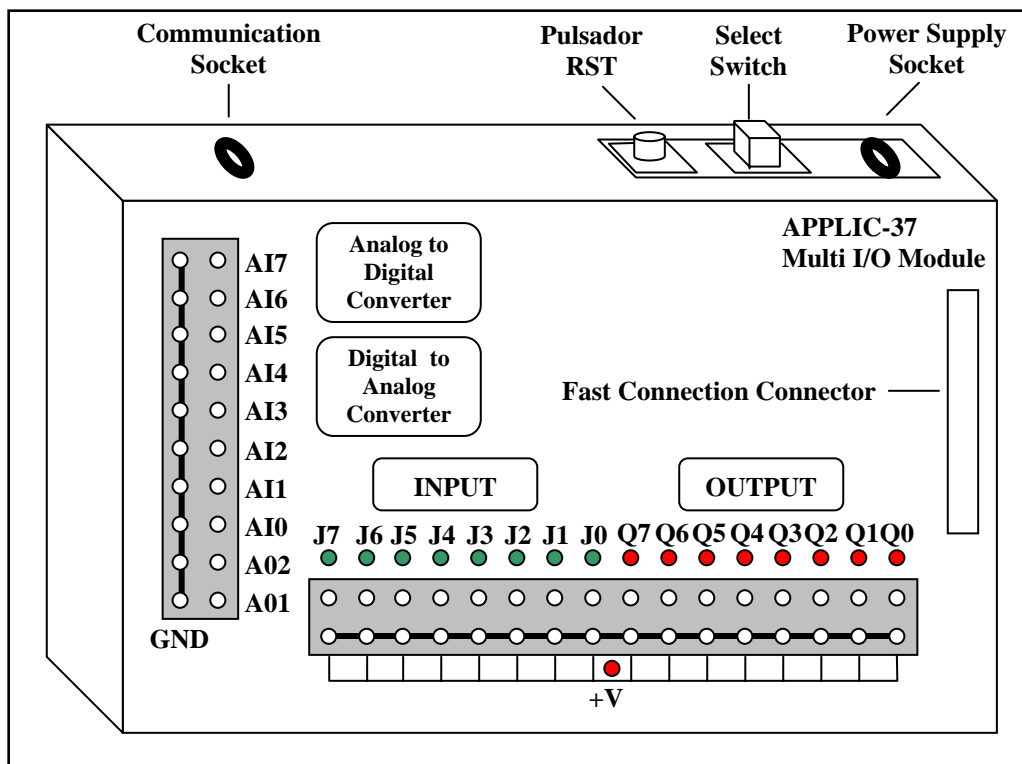
El Apéndice A incluye una rutina de muestra de lectura de datos de los canales de entrada ADC.

1.4 Usos del APPLIC-37

El APPLIC-37 es un sistema de entrenamiento en microcontrolador así como un controlador multi función. Puede usarse como:

- Sistema de control de automatización y Robótica.
- Sistema de adquisición de datos.
- Sistema de control de procesos.
- Sistema de entrenamiento del microcontrolador.
- Programación en lenguaje C de un controlador insertado.
- Puede usarse como un controlador independiente y también como una interfaz esclava de PC.
- El software en la PC se puede escribir en lenguajes C, Pascal, Basic, Delphi o Visual Basic.
- Viene acompañado con un paquete de desarrollo de software assembly, C y SESLOGO.

1.5 Instalación de la interfaz



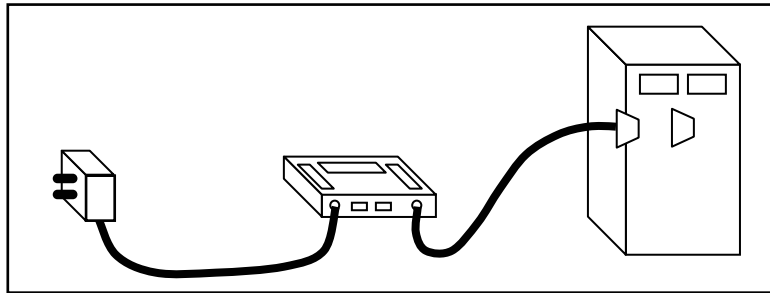
Paso 1: El cable de comunicación a la computadora tiene un enchufe auricular en un lado y un conector D () con nueve pines en el otro lado.

Inserte el enchufe del auricular en el zócalo de comunicación en la parte posterior de la APPLIC-37.

Paso 2: Localice los zócalos de comunicación en la parte posterior de su computadora (conectores D con nueve pines). Generalmente, hay dos canales de comunicación llamados COM1 y COM2.

Conecte el conector del cable al zócalo de comunicación en la computadora.

Paso 3: Conecte la fuente de poder al zócalo de energía en la parte posterior de la APPLIC-37 y luego a la red.



Verifique que la luz roja en la línea +V de la interfaz, está encendida.

Paso 4: Mueva el interruptor de selección hacia el lado más cercano al pulsador RST.

Paso 5: Presione el pulsador RST.

Todas las luces rojas deben ENCENDER.

Capítulo 2 - Programación en SESLOGO

Hay varios modos de programación: LOGO, BASIC, PASCAL, C y otros.

Hay también métodos de programación especiales para control – diagramas de flujo, diagramas en escalera o diagramas de control.

En este capítulo, trataremos con métodos de programación con la ayuda de un programa llamado SESLOGO. El lenguaje de programación es similar en principio al lenguaje LOGO (lo discutiremos más adelante), pero usa íconos como instrucciones.

2.1 Instalación del programa

La instalación del programa se hace sólo una vez. El software SESLOGO está diseñado para trabajar bajo WINDOWS.

Paso 1: Encienda la computadora y espere la pantalla de trabajo.

Paso 2: Haga doble clic en el ícono "My Computer".

Paso 3: Inserte el disquete con el software SESLOGO a la unidad de disco flexible A.

Paso 4: Copie la librería del SESLOGO del disco flexible A al disco duro C.

Paso 5: Entre a la librería de SESLOGO en C.

En esta librería usted encontrará el programa de instalación.

Paso 6: Haga doble clic sobre el ícono del archivo de instalación.

Ello abrirá los directorios y los archivos comprimidos de SESLOGO.

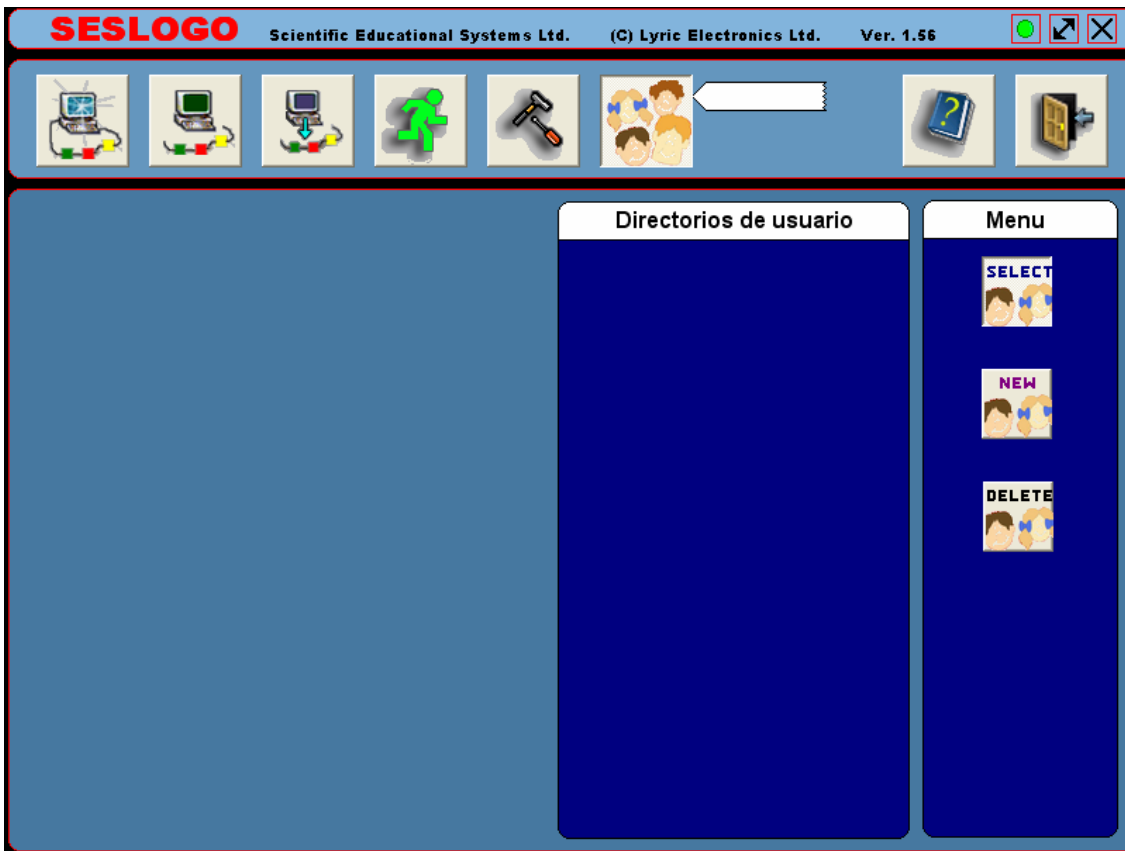
Paso 7: Uno de los archivos es SESLOGO.EXE.

Cree un ícono de acceso directo para este archivo y envíelo a su escritorio.

Paso 8: Verifique que APPLIC-37 está conectada a la computadora y a la red como fue explicado en la sección previa.

Paso 9: Haga doble clic sobre el ícono SESLOGO en el escritorio.

Aparecerá la pantalla siguiente:



Esta ventana le permite elegir su librería de usuario.

Las ventanas de software son ventanas dinámicas. Comprueban en todo momento la comunicación con la interfaz.

Si la comunicación está OK, aparece un círculo verde dentro del tercer recuadro en la parte superior derecha.

Identifique este recuadro.

Si aparece un círculo rojo, ello significa que no hay comunicación con la interfaz.

Paso 10: Si no hay comunicación con la interfaz, compruebe que el interruptor de selección está en el lado izquierdo (cerca del pulsador RST) y presione RST.

Paso 11: Si todavía no hay comunicación con la interfaz, pulse una vez (el botón izquierdo del mouse) en el ícono TC



Aparecerá la pantalla siguiente:




Paso 12: En el lado derecho (al lado del dibujo del cable de comunicación), aparece el número del canal de comunicación.

Si no hay comunicación con la interfaz, cambie el número del canal de comunicación según la salida de comunicación en la computadora que usted ha conectado. Después de esto pulse el botón OPEN (el botón con el dibujo del cable). Esta pulsación cerrará el canal de comunicaciones anterior y abre el nuevo.

Compruebe que aparece el círculo verde.

Abra la ventana TOOLS aún si la comunicación está OK.

Paso 13: En la ventana TOOLS (Herramientas) encontrará otro botón llamado INIT (Inicialización)  .

Este botón inicializa la interfaz, de esta manera puede responder a comandos directos (Modo Directo).

Pulse sobre este botón. El programa de inicialización será descargado en la interfaz.

Esta operación se debe hacer en la primera instalación de la interfaz o cuando la interfaz deja de responder a los comandos directos (explicado en el experimento 1.1), aunque haya comunicación.

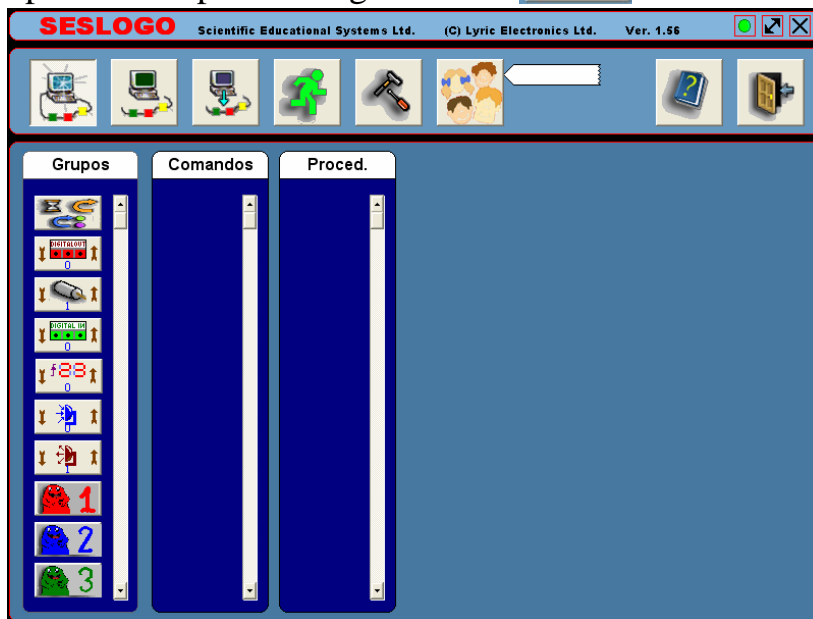
Paso 14: La programación abarca los botones de dibujo con el software paint (pintura).

La ventana TOOLS incluye un botón que busca en su computadora, el directorio de la aplicación PAINT.

Pulse sobre este botón  .

Este paso debe hacerse sólo una vez.

Paso 15: Pulse sobre el botón Direct Mode  (Modo Directo). Aparecerá la pantalla siguiente:



Esta ventana es la ventana de comando de modo directo y la explicaremos en el experimento siguiente.

2.2 Comandos Directos

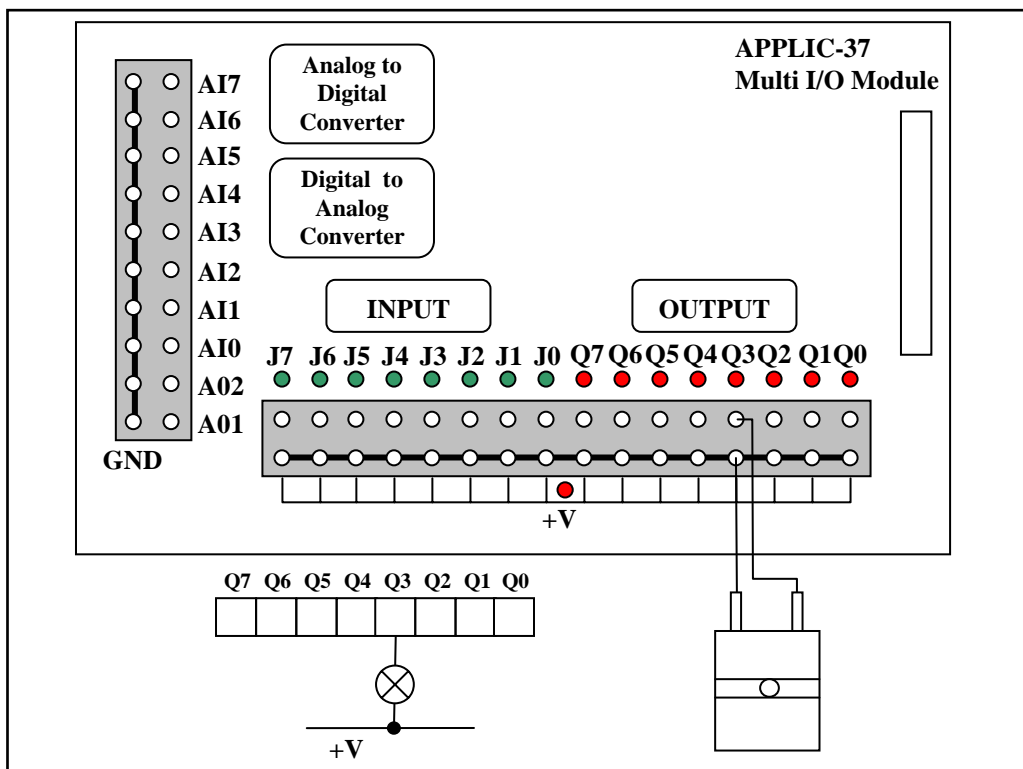
El software SESLOGO permite la operación de la interfaz APPLIC-37 directamente. Al pulsar sobre un botón de comando se ejecuta la interfaz.

Construiremos un dispositivo, que incluye una lámpara, un motor y un interruptor pulsador y aprenderemos cómo tratarlos directamente.

Paso 1: Conecte la interfaz a la PC y a la energía.

Paso 2: Prepare una lámpara, un motor y un interruptor con conectores de banana.

Paso 3: Conecte el circuito eléctrico como sigue:



Conecte la lámpara entre el voltaje +V y el canal de salida Q3 de la interfaz. Esta operación conecta Q3 a tierra (GND, el polo negativo de la fuente de poder) dentro de la interfaz y cierra el circuito.

Paso 4: Active el software SESLOGO como sigue:

Compruebe que la interfaz APPLIC-37 está adecuadamente conectada a la computadora y a la red eléctrica.

Compruebe que el interruptor selector esta para el lado de RST.

Presione el pulsador RST en la parte superior de la interfaz.

Todas las lámparas rojas deben ENCENDER.

Encienda la computadora.

Haga doble clic sobre el ícono SESLOGO.

Aparecerá la pantalla siguiente:



Paso 5: Verifique que aparece un círculo verde en la esquina derecha superior.

Paso 6: Si aparece un círculo rojo, realice las comprobaciones descritas en la sección, instalación del software.

Paso 7: Muévase con el cursor del ratón sobre los diferentes botones en la pantalla. Si se deja el cursor sobre uno de los botones, aparece una explicación corta sobre el botón.

Paso 8: Pulse sobre el botón NEW  (Nuevo).

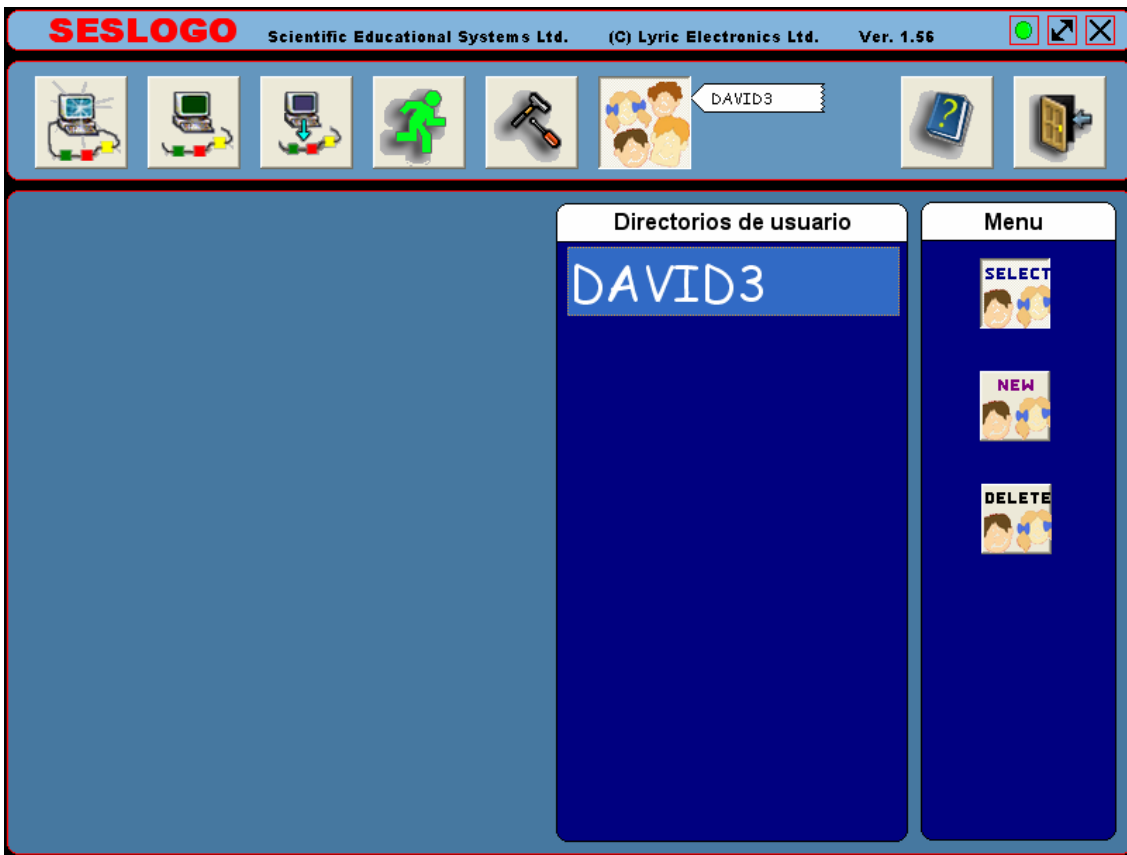
Aparecerá la pantalla siguiente:



Paso 9: Escriba un nombre diferente al de la lista de nombres que aparece en el campo "User Directories" / "Directorios de Usuario" (si hay algunos nombres). Por ejemplo, escriba el nombre DAVID3 y pulse OK.

Se cierra la ventana y el nombre DAVID3 aparece en la ventana "Directorios de Usuario".

Paso 10: Pulse sobre el botón SELECT (Seleccionar) y luego pulse sobre el nombre DAVID3.



Aparecerá el nombre DAVID3 cerca del botón USERS.

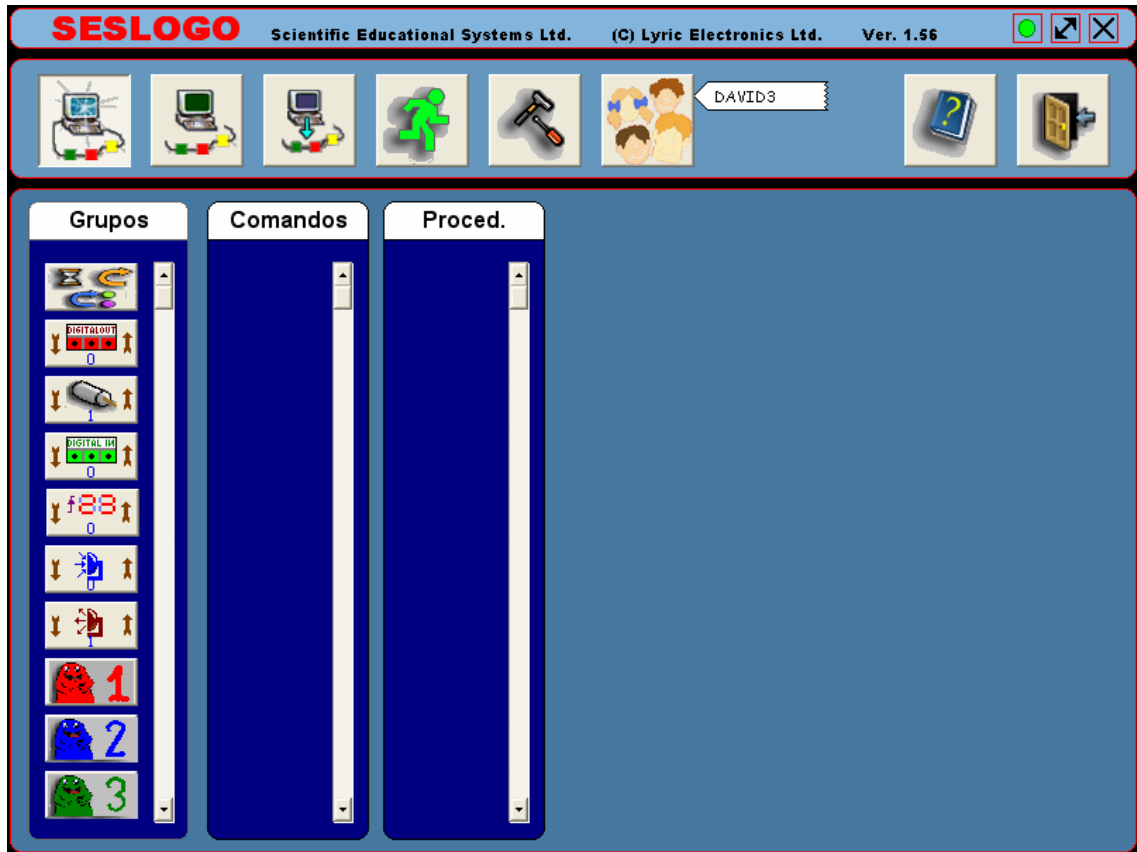
Todos los procedimientos que escribiremos serán almacenados en este directorio

Paso 11: Pulse sobre el botón Direct Mode



(Modo Directo).

Aparecerá la pantalla siguiente:



La pantalla incluye tres campos – Procedimientos, Comandos y Grupos.

Debido a que creamos una nueva librería, la columna de Procedimientos está vacía de procedimientos previos.

Cada botón en el campo de Grupos tiene un comando que le pertenece a él.

Paso 12: Pulse sobre el botón Digital Out  (Salida Digital).

Aparecerá la pantalla siguiente:



El botón Digital Out tiene tres comandos – ON, OFF y comando para determinar la energía. El comando de energía del sistema determina la cantidad de energía en el canal de salida cuando este se activa.

Paso 13: El botón Digital Out tiene dos flechas ↓ y ↑, lo que permite determinar el canal de salida deseado.

Fije el número 3 con las flechas (la lámpara se conecta a este canal).

Paso 14: Pulse sobre el botón ON en el campo de Comandos.

La lámpara se encenderá.

Paso 15: Pulse sobre el botón OFF.

La lámpara se apagará.

Paso 16: Encienda nuevamente la lámpara.

Paso 17: En el botón Set Power  (Ajustar Energía), aparece el número 150 (el valor máximo es 255).

Pulse sobre este botón. La intensidad de la luz de la lámpara descenderá.

Paso 18: Pulse sobre el número en el botón Set Power.

Aparecerá la pantalla siguiente:



Paso 19: Cambie el valor escrito a 255 y pulse OK.

Paso 20: Pulse sobre el botón Set Power.

La intensidad de la luz aumentará a su valor máximo.

Paso 21: Cambie la intensidad de la luz de la lámpara varias veces.

También observe el LED rojo conectado al canal de salida.

El resto de los comandos Set Power (energía del sistema) se relaciona con las variables (explicadas en el experimento 2.5).

Activación del motor:

Paso 22: El motor se puede conectar como la lámpara.

Conecte el motor entre el canal Q5 y +V.

Paso 23: Use los comandos Digital Out (aquellos que usó anteriormente) para activar el motor, desconéctelo y cambie su velocidad.

Paso 24: Para controlar la el sentido de giro del motor, conéctelo entre dos canales de salida.

Cada par de canales de salida tiene un número de motor:

MOTOR 1	-	Q1,Q0
MOTOR 2	-	Q3,Q2
MOTOR 3	-	Q5,Q4
MOTOR 4	-	Q7,Q6

Conecte los alambres del motor entre los canales Q4 y Q5.

Nota importante

Para motores con corriente superior a 0.5A no cambia la configuración de energía. Utilice valores de 0 y 255 solamente.

Paso 25: Pulse sobre el botón MOTOR.

Aparecerá la pantalla siguiente:



Paso 26: Cambie el número en el botón MOTOR a 3, para direccionar los canales de salida Q4 y Q5.

Paso 27: Pulse sobre el botón ON.

El motor comenzará a girar.

Paso 28: Cambie la velocidad del motor a su velocidad más baja con el botón Set Power



Paso 29: Pulse sobre el botón CW (Sentido Horario)  .

El motor girará en sentido horario.

Paso 30: Pulse sobre el botón CCW (Sentido Antihorario)  .

El motor girará en sentido antihorario.

Paso 31: Pulse sobre el botón "Change Direction"  (Cambio de Dirección).

El motor cambiará el sentido de su rotación.

Nota:

Se recomienda cambiar el sentido de giro del motor solamente cuando el motor está parado.

Paso 32: Pare el motor pulsando el botón OFF.

Paso 33: Juegue con los botones de varias maneras.

Cambio de lecturas:

Paso 34: El interruptor pulsador tiene puntos de conexión, marcados con los números 1,2,3.

Identifique estos puntos.

Cuando usamos los puntos 1 y 3, el interruptor sirve como un interruptor pulsador del tipo Normalmente Abierto.

Paso 35: Entre las entradas digitales (J7-J0) y la línea +V están conectados sensores digitales (con dos estados posibles) e interruptores.

Conecte los puntos 1 y 3 del interruptor a J4 y +V.

Paso 36: Pulse sobre el botón Digital Input (Entrada Digital) .



Aparecerá la pantalla siguiente:



Paso 37: Cambie el número del botón de la Entrada Digital a 4, para direccionar el canal de entrada J4.

Paso 38: El único comando directo para ejecutarse es el comando READ para leer el estado de la entrada.

Pulse sobre el botón del comando READ.

Aparecerá el número 0 en el botón READ.

Paso 39: Presione el pulsador.


Manténgalo presionado y pulse nuevamente sobre el comando READ.

En esta oportunidad aparecerá el número 1.

Paso 40: Repita los pasos 38 y 39 varias veces.

Lecturas en el Contador:

Paso 41: La interfaz cuenta los cambios de estado que ocurren en la entrada digital.

En la ventana Group ubique el botón Counter  (Contador).

Pulse sobre él.

Los comandos directos del contador son 0 (reajuste) y Lectura.

Paso 42: Cambie el número del contador a 4 (para J4).

Paso 43: Pulse sobre el botón del comando 0 (borrar).

Paso 44: Pulse sobre el botón READ.

Aparecerá el número 0 en el botón.

Paso 45: Presione el interruptor pulsador una vez y libérela.

Paso 46: Pulse nuevamente sobre el botón READ.

Aparecerá el número 1 en el botón.

Paso 47: Presione el interruptor pulsador 3 veces más.

Paso 48: Pulse nuevamente sobre el botón READ.

Aparecerá el número 4 en el botón.

Lectura analógica:

APPLIC-37 tiene 8 canales de entrada analógicos. El voltaje de entrada analógico (0-5V) es convertido a un número en el rango 0-255.

Los canales AI0-AI5 son para lectura de voltaje 0-5V
AI6 y AI7 son para sensores pasivos tales como LDR, fototransistor o termisores.

Paso 49: Conecte un LDR entre AI6 y la línea GND en contra del mismo.

Paso 50: Cubra el LDR y presione sobre el botón LEER READ.

2.3 Programas y Procedimientos

El lenguaje LOGO fue el primer lenguaje de programación en usar reglas de programación estructurada. El lenguaje incluye el sistema básico y primario de comandos. Con estos comandos, fueron creadas las series de instrucciones. A cada serie de instrucciones se le asignó un nombre. Cada serie de instrucciones se llama procedimiento.

La lectura del procedimiento por nombre causó la ejecución de una serie de instrucciones. De esta manera, el procedimiento se convierte en un nuevo comando.

De esta manera, el lenguaje LOGO se puede ampliar infinitamente. Es similar a usar un número limitado de letras para crear una gran cantidad de palabras y de oraciones.

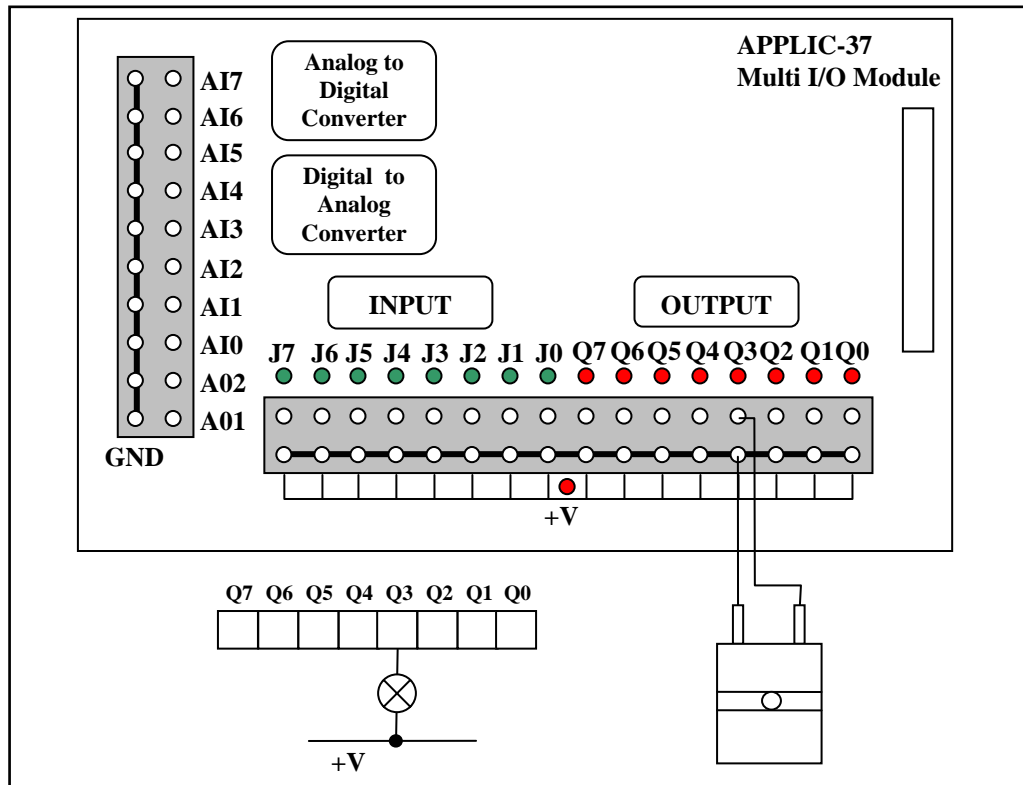
Hoy todos los lenguajes de programación son construidos de manera similar y usan reglas de programación estructuradas.

SESLOGO también usa reglas de programación estructurada. En el SESLOGO usamos íconos (botones) como comandos. Para crear un procedimiento, usado como un nuevo comando, necesitamos crear un botón nuevo para ello, él que será utilizado como el nombre del procedimiento.

Paso 1: Conecte la interfaz a la PC y a la energía.

Paso 2: Prepare una lámpara con conectores de banana.

Paso 3: Conecte el circuito eléctrico como sigue:



Conecte la lámpara entre el voltaje +V y el canal de salida Q3 de la interfaz.

Paso 4: Active el software SESLOGO como sigue:

Compruebe que la interfaz APPLIC-37 está conectada adecuadamente a la computadora y a la red.

Compruebe que el interruptor selector está hacia el lado de RST.

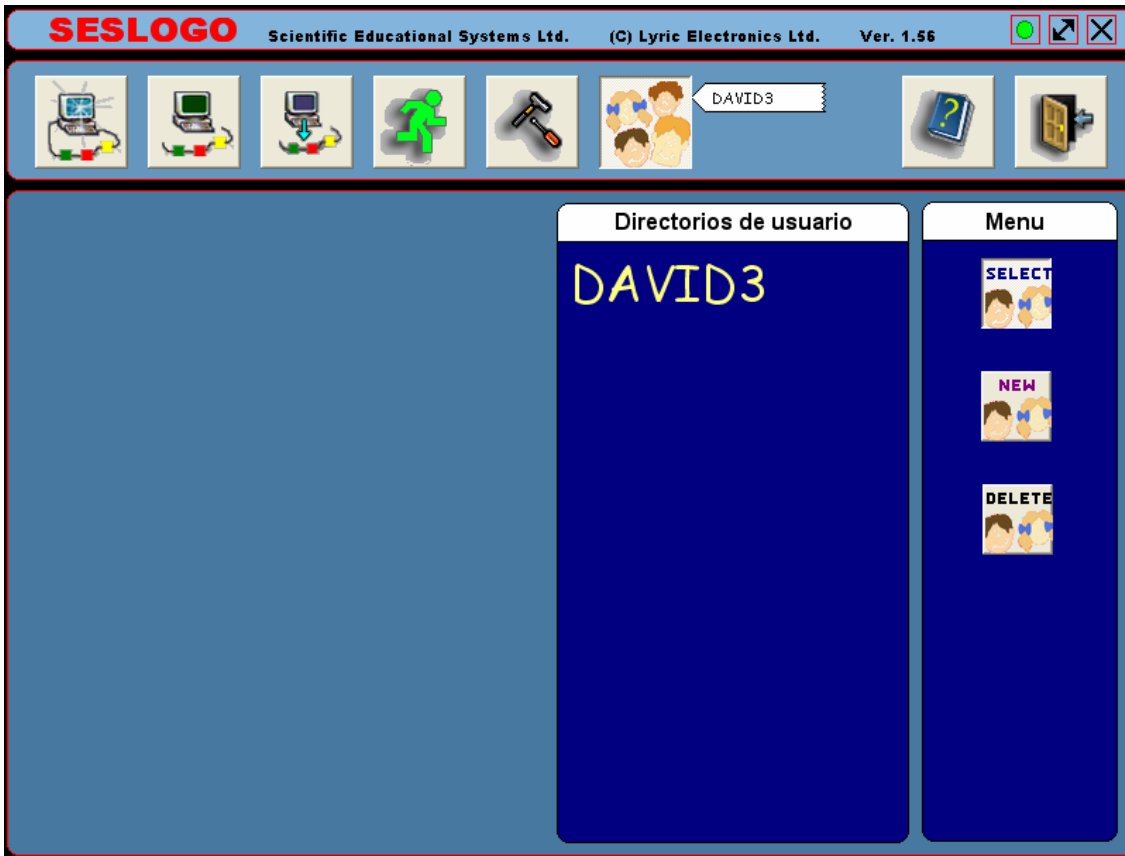
Presione el pulsador RST de la interfaz.

Todas las luces rojas deben encenderse.

Encienda la computadora.

Haga doble clic sobre el ícono SESLOGO.

Aparecerá la siguiente pantalla:



- Paso 5: Compruebe que el círculo verde aparece en la esquina derecha superior.
- Paso 6: Si aparece un círculo rojo, siga las instrucciones descritas en la sección instalación del software.
- Paso 7: Seleccione la nueva librería (DAVID 3) que creó en el experimento previo.

Paso 8: Pulse sobre el botón Off Line Mode  Modo Fuera de Línea.

Aparecerá la pantalla siguiente:



Paso 9: Dos ventanas adicionales son agregadas a las ventanas previas – Ventana de Edición y Menú.

Muévase con el cursor sobre los botones del campo del menú e identifique su significado.

Paso 10: El botón derecho superior en la ventana del Menú se usa para crear un nuevo botón de procedimiento.

Pulse sobre este botón



Aparecerá un nuevo botón en la ventana de Procedimientos.

Paso 11: Cada botón puede editarse como dibujo y como texto.

Pulse sobre el botón del ícono Edit Procedure (Procedimiento de Edición) en la ventana del Menú.



Aparecerá la pantalla Paint (Pintura) de la computadora con el recuadro del botón en la esquina izquierda superior.

Paso 12: Agrande el recuadro con la lupa.

Paso 13: Con colores y las herramientas de Paint dibuja un círculo rojo lleno en el centro del recuadro.

Paso 14: Salga del programa Paint pulsando en la X en la esquina derecha superior o pulsando la función FILE y EXIT.

El programa paint le preguntará: "Save changes?" (¿Guardamos los Cambios?).

Pulse OK para guardar el dibujo.

Paso 15: Otra ventana se abre con un campo para teclear el nombre del botón como texto.

Teclee la palabra RED en el campo.

Pulse OK para guardar.

El botón en la ventana de Procedimientos cambiará al botón siguiente.



Paso 16: Pulse sobre el botón Salida Digital



Paso 17: Cámbielo al número 3.

Paso 18: Pulse sobre el botón ON (Conectado).

Aparecerá la pantalla siguiente:



Paso 19: Pulse ahora sobre el botón General Group  Grupo General.

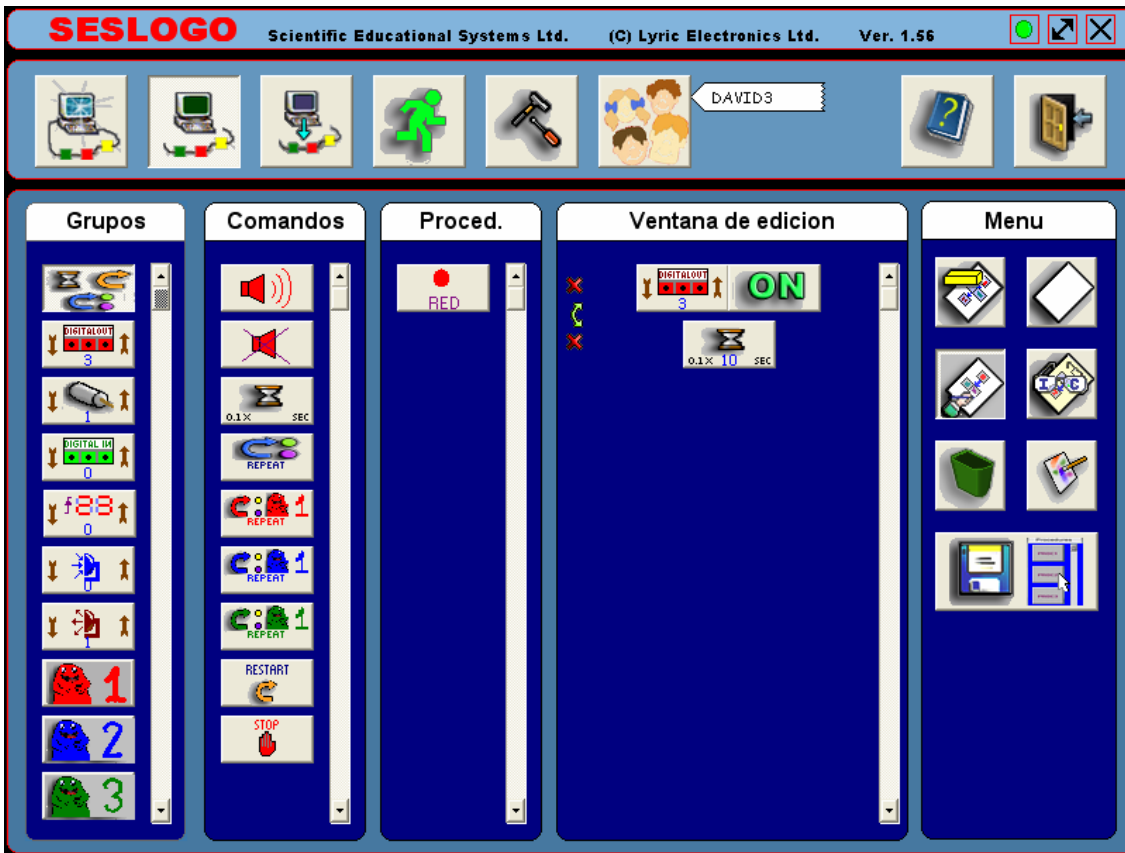
Aparecerá la pantalla siguiente:



Los comandos de este grupo son: retardo, sonido, repita el comando, regresar al principio, repetición por variable y parada.

Paso 20: Pulse sobre el botón delay  (retardo).

Aparecerá la pantalla siguiente:



Paso 21: El retardo se mide en décimas de segundo.

Pulse sobre el número que aparece en el botón.

Se abrirá una ventana con un campo para escribir el retardo deseado.

Paso 22: Teclee el número 20 para un retardo de 2 segundos.

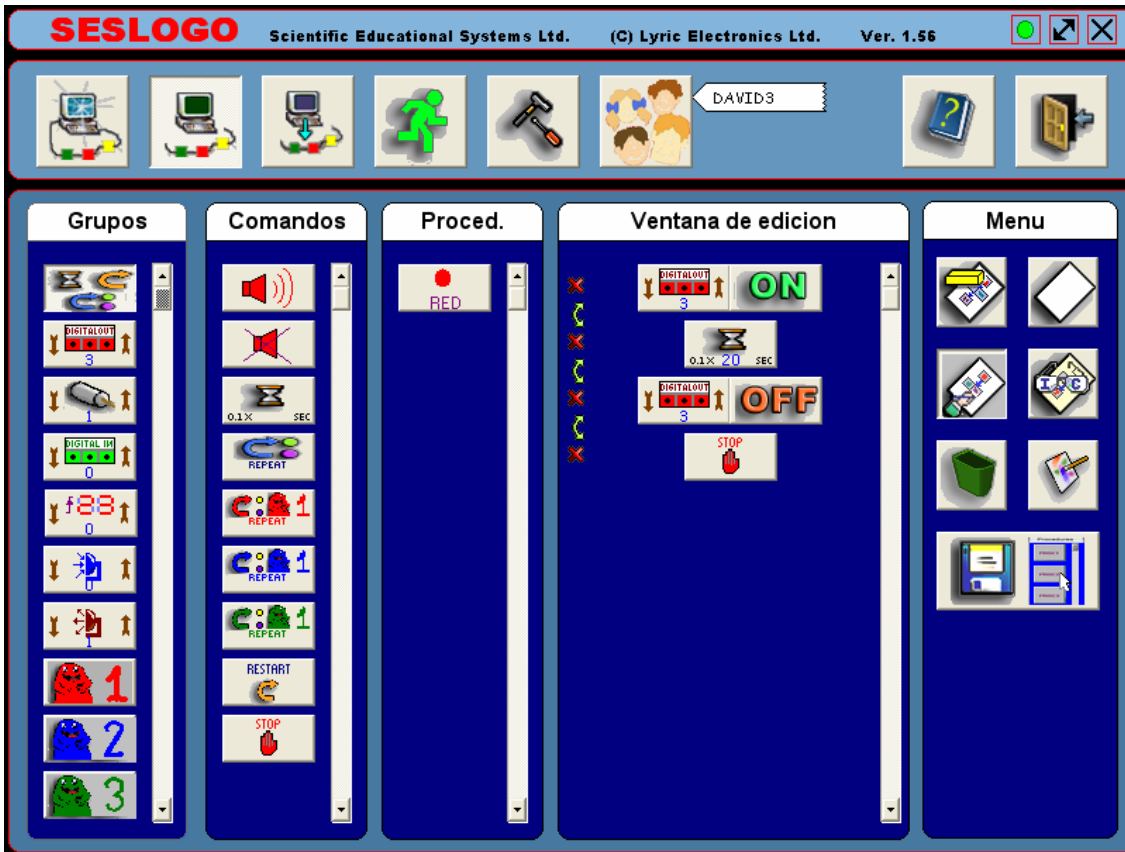
Pulse OK y la ventana se cerrará.

Paso 23: Pulse nuevamente sobre el botón de Salida Digital y seleccione el comando OFF (Desconectado).


Paso 24: Pulse nuevamente sobre el botón grupo general.

Paso 25: Seleccione el botón STOP (Parada)  para parar el procedimiento.

Compruebe que usted recibe el procedimiento siguiente:



Paso 26: El programa que aparece en la ventana de Edición no pertenece todavía al botón RED (ROJO) que usted creó.


Identifique el botón Save As (Guardar como)  en la ventana del Menú y pulse sobre él.

El botón será presionado hacia adentro.


El sistema espera que se pulse sobre cualquier botón de procedimiento.

Paso 27: Pulse sobre el botón del procedimiento RED.

El procedimiento es guardado bajo este botón.

Paso 28: Para cerciorarse de que el procedimiento es guardado bajo el botón RED, pulse sobre el botón izquierdo superior Clear Edit Window (Borrar Ventana de Edición)  en la ventana del Menú.

La ventana de Edición será borrada.

Paso 29: Pulse ahora sobre el botón View Procedure  (Ver Procedimiento), (el segundo botón derecho superior).

El software espera que se pulse sobre el botón del procedimiento adecuado.

Paso 30: Pulse sobre el botón RED.

Las instrucciones del procedimiento aparecerán en la ventana de editar procedimiento.

El botón del procedimiento RED ahora aparecerá en el botón más bajo (botón de Guardar) en la ventana del Menú.

Cada pulsación en el botón Guardar guardará de nuevo el programa en la ventana de Editar bajo el nombre RED.

Note que:

Distinguimos entre dos expresiones similares – instrucciones y comandos.

La instrucción es un comando escrito en el procedimiento y ejecutado solamente cuando se ejecuta el procedimiento.

El comando se ejecuta inmediatamente sobre la dirección de él. La mayoría de los comandos de grupo también se utilizan como instrucciones dentro de los procedimientos, pero no todos.

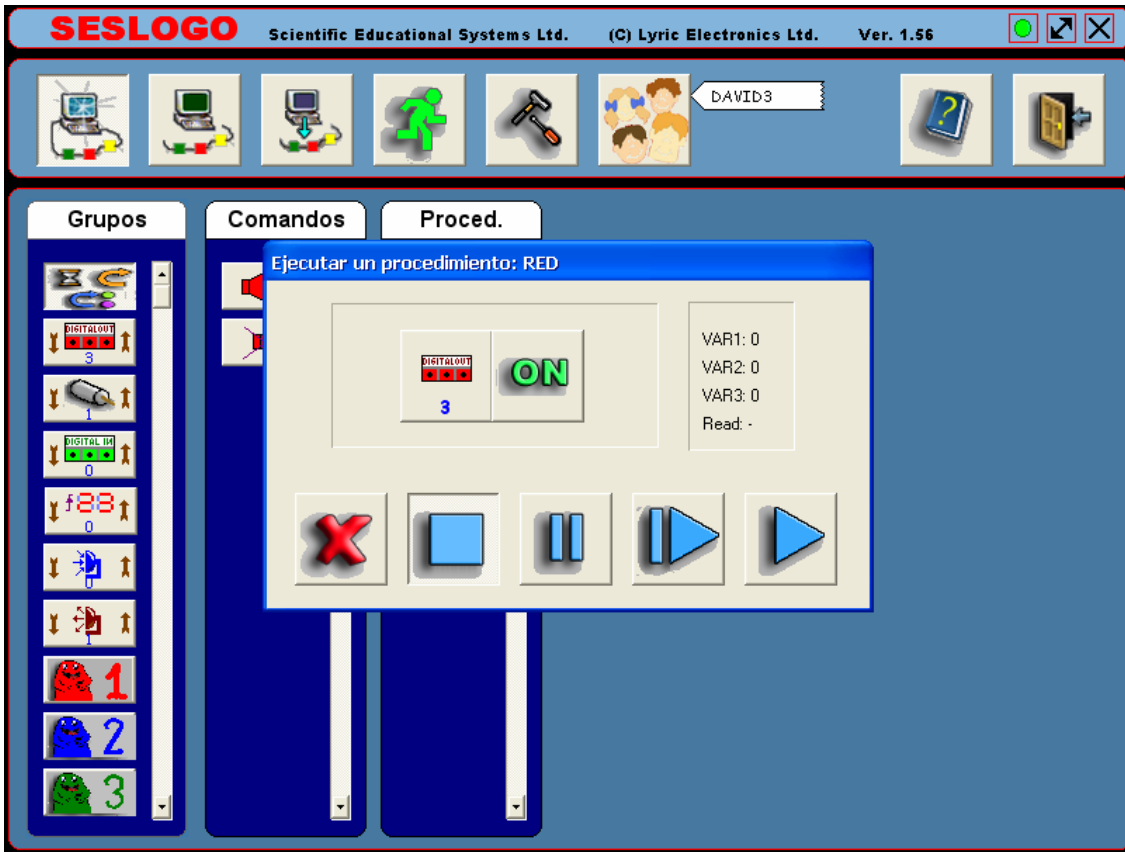
Paso 31: Ahora tenemos un botón nuevo, llamado RED con un dibujo sobre él, el cual maneja la iluminación de una lámpara, con dos segundos de retardo y se apaga.


Pulse sobre el botón Modo Directo



Paso 32: Pulse ahora sobre el botón ROJO.

Se abrirá una ventana de funcionamiento, presentando la primera instrucción del procedimiento.




Paso 33: Pulsando sobre el botón  se ejecutará el procedimiento, mientras se presentan las instrucciones de ejecución en la ventana.

Pulse sobre este botón.


La lámpara se ENCENDERÁ y se APAGARÁ después de dos segundos.

Paso 34: Pulse nuevamente sobre el botón ROJO.

Paso 35: El botón  se usa para la ejecución de una instrucción del procedimiento a la vez.

Pulse sobre este botón una vez.

La lámpara debe encenderse.

Paso 36: Pulse nuevamente sobre el botón .

Se producirá un retardo de dos segundos y después aparecerá el comando OFF (Desconectado).

Paso 37: Pulse nuevamente sobre el botón  .


La lámpara debe apagarse.

Paso 38: Pulse nuevamente sobre el botón  .

Se termina la ejecución del procedimiento.


Paso 39: Ahora cambiaremos el procedimiento al centelleo del bucle infinito.

Pulse sobre el botón Modo Fuera de Línea  .

Paso 40: Pulse sobre el botón “Ver Procedimiento”  y después sobre el botón ROJO.


En la ventana de Editar aparecerán las instrucciones del procedimiento.

Paso 41: Para realizar cambios en el procedimiento debemos movernos al estado editar.

Pulse sobre el botón Procedimiento de Editar  .

Paso 42: En el lado izquierdo de cada instrucción hay tres señales:

X en el medio, que significa: "borrar la instrucción".

Dos señales  (arriba y abajo), que significa: "cambie los lugares entre esta instrucción y la siguiente".

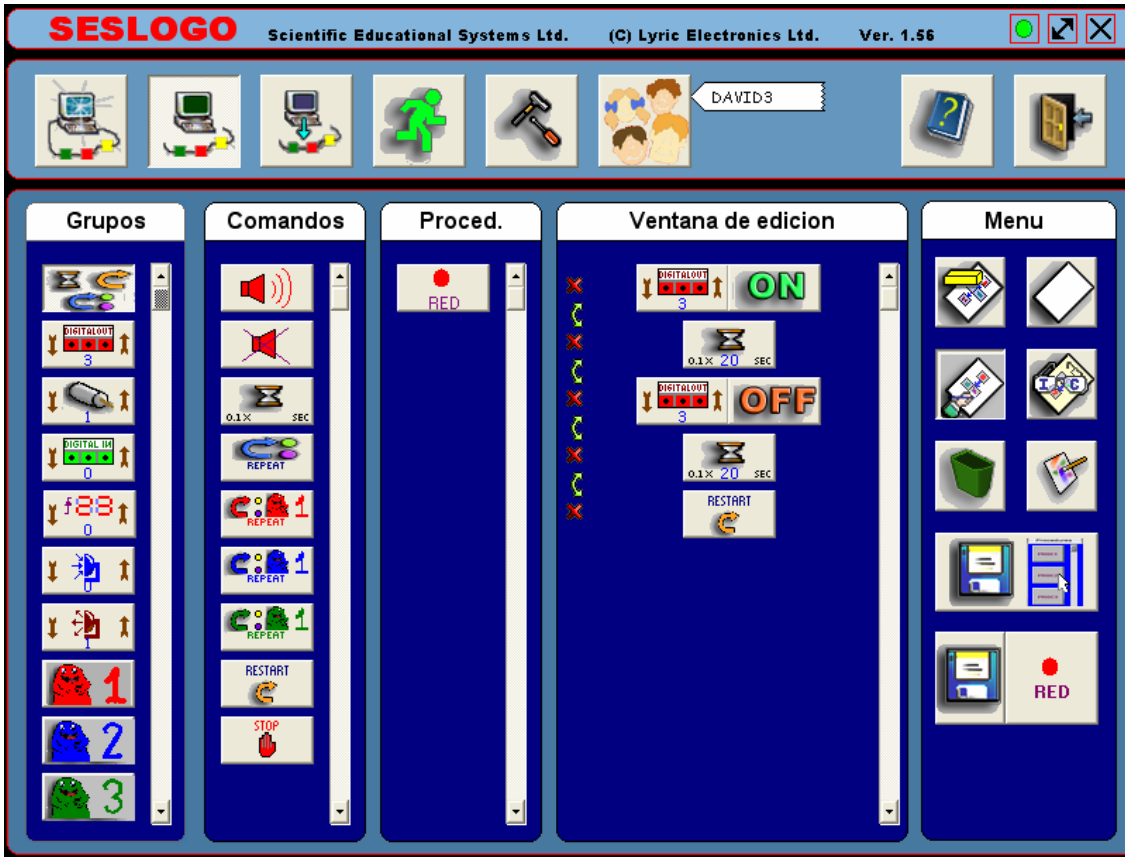
Borre la instrucción STOP al final del procedimiento.

Paso 43: Agregue otra instrucción de retardo de dos segundos a la continuación del procedimiento.

Paso 44: Agregue una instrucción de Restart  Reinicialización al final del procedimiento.

Cuando el procedimiento alcance esta instrucción, se moverá al principio y ejecutará las instrucciones del procedimiento otra vez.

Paso 45: Compruebe que usted recibe el procedimiento siguiente:



Pulse sobre el botón SAVE



Guardar (el de más abajo) en la ventana del menú.

Paso 46: Pulse sobre el botón Modo Directo



Paso 47: Pulse sobre el botón de procedimiento ROJO.

Se abrirá una ventana de ejecución.

Paso 48: Pulse sobre el botón de funcionamiento normal



El procedimiento funcionará en un bucle infinito y la lámpara se ENCENDERÁ y se APAGARÁ alternadamente.

Paso 49: Para parar el procedimiento, pulse el botón



Paso 50: Cierre la ventana Run Procedure (Ejecución de Procedimiento).


Paso 51: La interfaz es también una computadora. Ella tiene su propia CPU (Unidad Central de Procesamiento) y su memoria.

Podemos descargar el procedimiento en su memoria.

Identifique el botón Download  (Descargar) .

Paso 52: Compruebe que aparece el círculo verde en la esquina derecha superior de la pantalla (no el círculo rojo).

Si no, presione el pulsador RST en la interfaz.

Paso 53: Pulse sobre el botón Descargar  .

El software espera que se pulse el botón del procedimiento destinado para descargar.

Paso 54: Pulse sobre el botón del procedimiento RED.

Aparecerá una regla, lentamente llenando para arriba, simbolizando la transferencia directa del procedimiento de la computadora a la interfaz

Cuando desaparece la regla y no hay mensaje de error, indica que el procedimiento fue descargado totalmente en la interfaz.

Paso 55: En la fila superior de la pantalla aparece un botón con un hombre corriendo On



Este botón indica el funcionamiento de un procedimiento, que está localizado en la memoria de la interfaz.

Pulse sobre este botón.

La lámpara ahora centellea sin parar.

Paso 56: El círculo verde en la esquina derecha superior cambiará a rojo.

La interfaz está ocupada corriendo el procedimiento y no responde a la computadora.

Para parar el procedimiento y regresar al círculo verde, presione el pulsador RST en la parte superior de la interfaz.

El procedimiento no se borra de la memoria de la interfaz, aún si desconectamos la energía a la interfaz.

Paso 57: Compruebe esto.

Desconecte la energía de la interfaz.

El círculo verde cambiará a rojo.

Paso 58: Conecte la energía a la interfaz.

Si el círculo rojo no cambia a verde en dos segundos, presione el pulsador RST.

Paso 59: Ejecute nuevamente el procedimiento.

La lámpara nuevamente centellará.

Paso 60: Presione el pulsador RST en la interfaz para parar la ejecución.

Paso 61: La interfaz puede funcionar como un controlador independiente tan pronto como se almacena en ella cualquier procedimiento.

Desconecte el cable de comunicación de la interfaz.

Aparecerá un círculo rojo en la pantalla.

Paso 62: Mueva el interruptor selector a la derecha (hacia el zócalo de la fuente de poder).

Presione RST.

La lámpara comenzará a centellar.

Al presionar nuevamente RST no se detendrá el procedimiento.

Paso 63: Desconecte la energía y reconéctela.

La lámpara comenzará a centellar.

Paso 64: Conecte nuevamente el cable de comunicación a la interfaz.

Paso 65: Mueva el interruptor selector a la izquierda (hacia el pulsador RST) y presione RST.

La lámpara se apagará y aparecerá el círculo verde en la pantalla.

Ejercicio de desafío – luces centellantes:

Paso 66: En el dispositivo que usted tiene hay dos lámparas.

Ponga un casquillo rojo en la lámpara de la derecha, que está conectada con Q3.

Paso 67: Ponga un casquillo verde en la lámpara de la izquierda y conéctelo entre el canal de salida Q5 y +V.

Paso 68: Prepare un nuevo botón de procedimiento con círculos verde y rojo y llámelo BLINK (CENTELLEAR).

Paso 69: Escriba un procedimiento, que haga centellear a las lámparas. La lámpara derecha estará encendida durante dos segundos y la lámpara izquierda también por dos segundos en forma alternada.

Guarde este procedimiento bajo el botón BLINK.

Paso 70: Ejecute el procedimiento desde la computadora y compruebe su comportamiento.

Paso 71: Descargue el procedimiento en la interfaz y ejecútelo desde la computadora.

Paso 72: Ejecute el procedimiento independientemente desde la interfaz.

Paso 73: No olvidar al final del procedimiento, conectar el cable de comunicación a la interfaz y mover el interruptor selector hacia el pulsador RST.

2.4 Procedimientos como Nuevas Instrucciones

Para cada procedimiento que construimos, creamos un botón con un ícono y/o nombre para él.

Este botón puede usarse como un comando o como una nueva instrucción de procedimiento. De esta forma, se crea un rico lenguaje de computadora, el cual se adapta a la aplicación que queremos implementar.

En el experimento previo, aprendimos a crear procedimientos, que conectan y desconectan lámparas.

En este experimento, usaremos estos procedimientos para construir un semáforo.

De manera similar, operaremos el motor del sistema como un tambor de giro de una máquina de lavado.

Cada computadora incluye una Unidad Central de Procesamiento (CPU), que decodifica las instrucciones del procedimiento y las ejecuta.

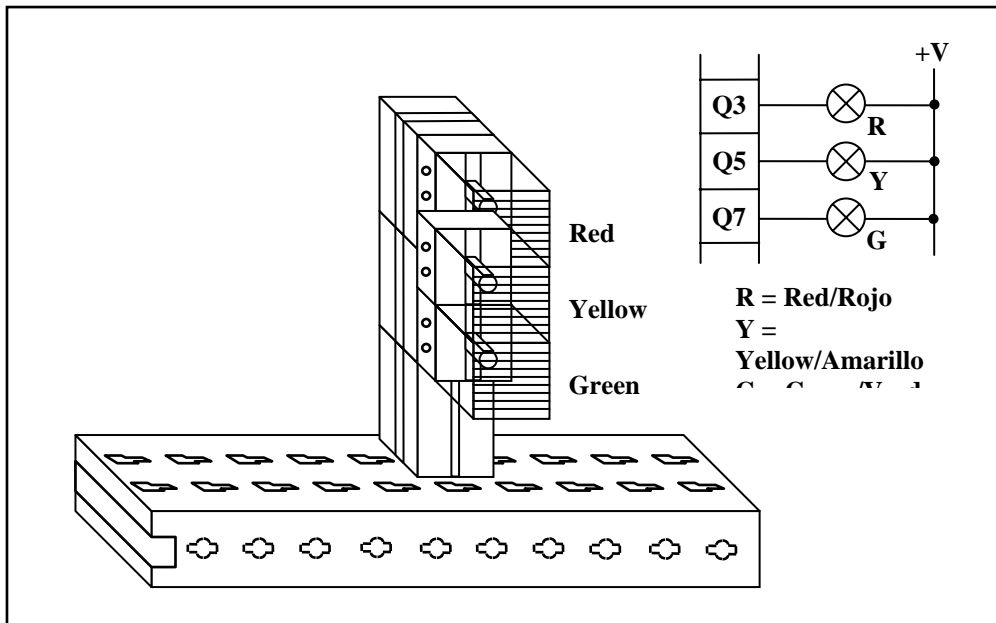
El software principal (SESLOGO en nuestro caso), el cual permite editar el procedimiento, guardarlo, ejecutarlo y operar los diferentes comandos, se llama sistema operativo.

El procedimiento principal que la computadora ejecuta al principio se llama programa de computadora. Al final del programa necesitamos agregar una instrucción, que vuelve el programa al principio (si quisiéramos que el programa fuera ejecutado repetidas veces), o parar el programa y regresar al sistema operativo.

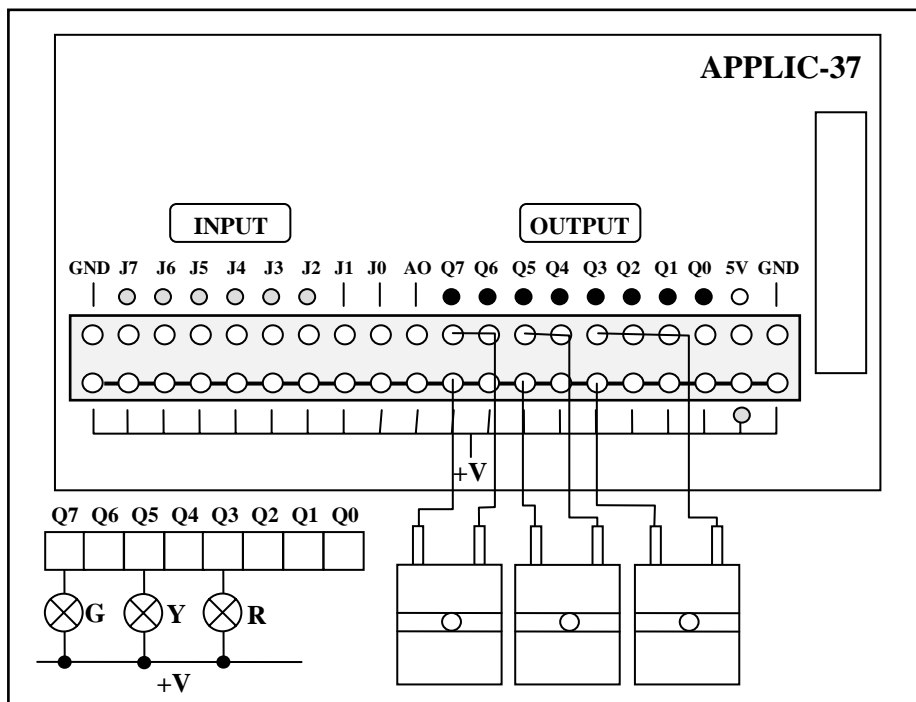
El programa puede llamar y usar procedimientos.

Paso 1: Conecte la interfaz a la PC y a la energía.

Paso 2: Construya el dispositivo descrito como sigue:



Paso 3: Conecte el circuito eléctrico como sigue:



Paso 4: Active el software SESLOGO como sigue:

Compruebe que la interfaz APPLIC-37 está conectada adecuadamente a la computadora y a la red.

Compruebe que el interruptor selector está en el lado RST.

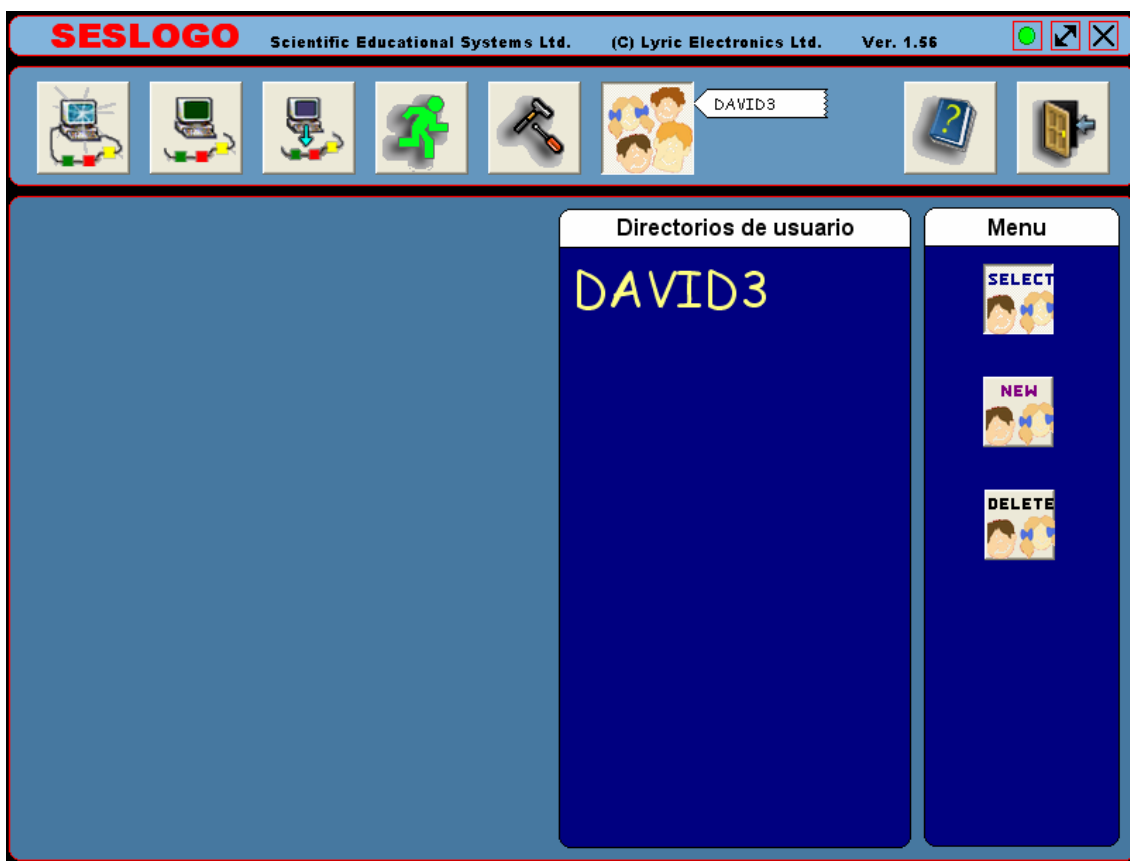
Presione el pulsador RST en la parte superior de la interfaz.

Todas las luces rojas deben estar encendidas.

Encienda la computadora.

Haga doble clic en el ícono de SESLOGO.

Aparecerá la pantalla siguiente:



Paso 5: Compruebe que aparece un círculo verde en la esquina derecha superior.

Paso 6: Si aparece un círculo rojo, siga las instrucciones descritas en la sección instalación del software.

Seleccione la nueva librería (DAVID 3) que usted creó en el experimento previo.

Paso 7: Desde este experimento y hacia adelante no describiremos la construcción del procedimiento. Si usted no recuerda cómo hacerlo, vaya de nuevo al experimento 1,2.

Cambie el procedimiento ROJO al procedimiento siguiente:

DIGITAL OUT 3	ON
------------------	----



DIGITAL OUT 3	OFF
------------------	-----

Paso 8: Ejecute y compruebe el comportamiento del procedimiento.

Paso 9: Cree un nuevo botón YELLOW (Amarillo) con un círculo lleno en amarillo.

Paso 10: Construya el procedimiento siguiente y guárdelo bajo el botón AMARILLO.

DIGITAL OUT 5	ON
------------------	----



DIGITAL OUT 5	OFF
------------------	-----

Paso 11: Ejecute y compruebe el comportamiento del procedimiento.

Paso 12: Cree un nuevo botón GREEN (Verde) con un círculo lleno en verde.

Paso 13: Construya el procedimiento siguiente y guárdelo bajo el botón VERDE.

DIGITAL OUT 7	ON
------------------	----

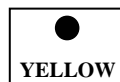
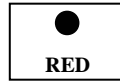


DIGITAL OUT 7	OFF
------------------	-----

Paso 14: Ejecute y compruebe el comportamiento del procedimiento.

Paso 15: Cree un nuevo botón TRAFFIC (Tráfico) con tres círculos – rojo, amarillo y verde.

Paso 16: Construya el siguiente procedimiento y guárdelo bajo el botón TRAFFIC.



La última instrucción (Restart/Reajuste) se puede hallar en el grupo de instrucciones generales.

Esta instrucción ejecuta nuevamente el procedimiento.

Paso 17: Ejecute y compruebe el comportamiento del procedimiento.

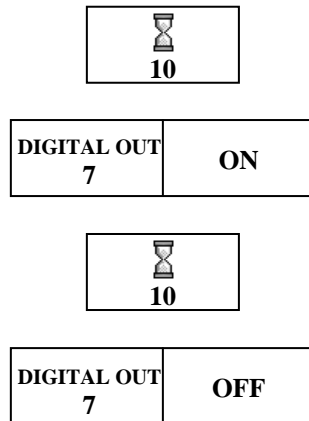
Paso 18: Descargue el procedimiento en la interfaz y ejecútelos independientemente (moviendo el interruptor selector a la derecha).

Paso 19: Mueva el interruptor selector a la izquierda y presione RST.

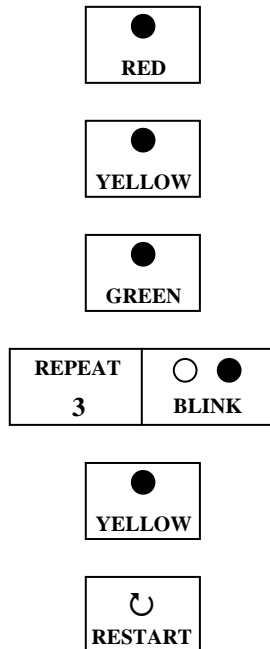
Paso 20: Agregaremos un verde centelleante después del verde en el procedimiento del semáforo.

Cree un nuevo botón BLINK con dos círculos verdes, uno lleno y el otro vacío.

Paso 21: Construya el procedimiento siguiente y guárdelo bajo BLINK.



Paso 22: Cambie el procedimiento del semáforo al siguiente:



A la instrucción REPEAT (Repetir) se le puede encontrar en el grupo de instrucciones generales.

Esta instrucción opera el procedimiento a la derecha varias veces según el número escrito en ella.

Paso 23: Ejecute y compruebe el procedimiento en la computadora y también descargándolo a la interfaz.

2.5 – Condiciones y Decisiones

Todos los procedimientos y programas que escribimos operaron hasta ahora el sistema controlado (lámparas y motores) solamente por programa.

Este sistema se llama sistema de Lazo Abierto. El controlador opera el sistema controlado sin la recepción de datos del sistema y relacionar a estos datos.

Un sistema de control real es un sistema, que funciona como lazo cerrado. Recopila datos del sistema controlado y según estos datos, toma decisiones y actúa por consiguiente.

Estos datos se reciben de interruptores y de sensores. Los interruptores y los sensores están conectados con las unidades de entrada de la interfaz. El software del controlador direcciona las unidades de entrada para conseguir el estado de los sensores y de los interruptores.

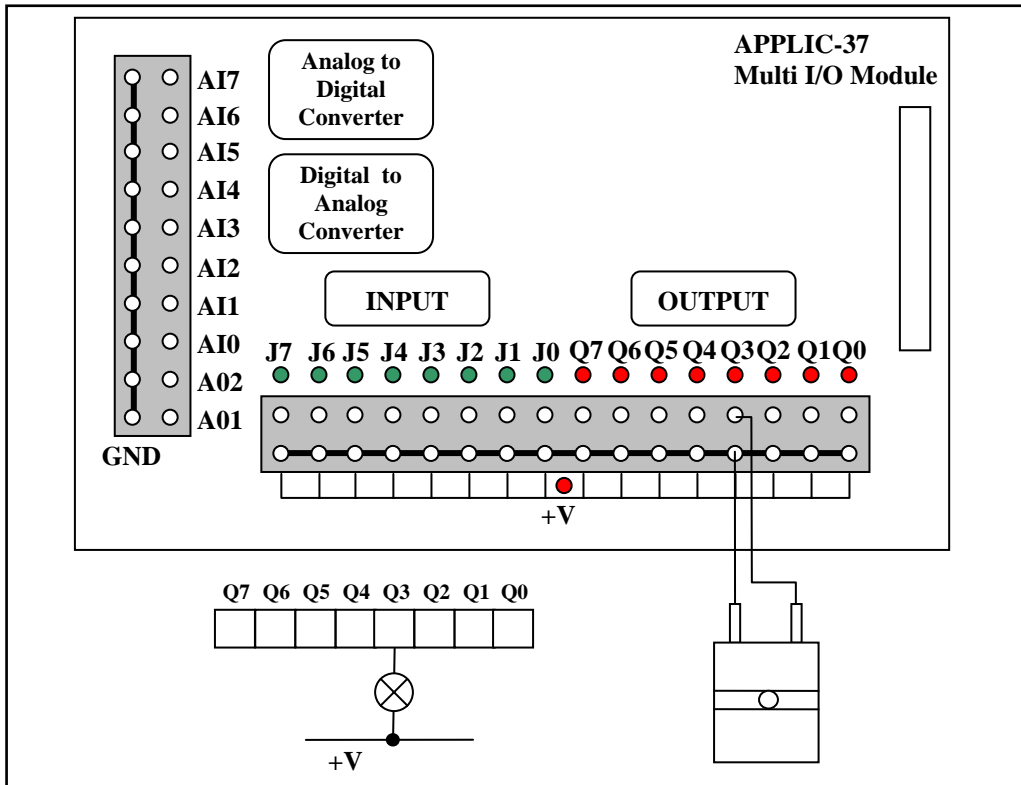
Por ejemplo, un sistema de iluminación automático. Este sistema incluirá un sensor de luz. El sistema de control encenderá la lámpara cuando está oscuro y la apagará cuando hay luz. El sistema se adapta automáticamente al horario de verano (cuando la noche es corta) y al horario de invierno (cuando la noche es larga y comienza temprano).

Por supuesto, necesitamos tener cuidado que el sensor de luz no sea influenciado por el encendido de la luz.

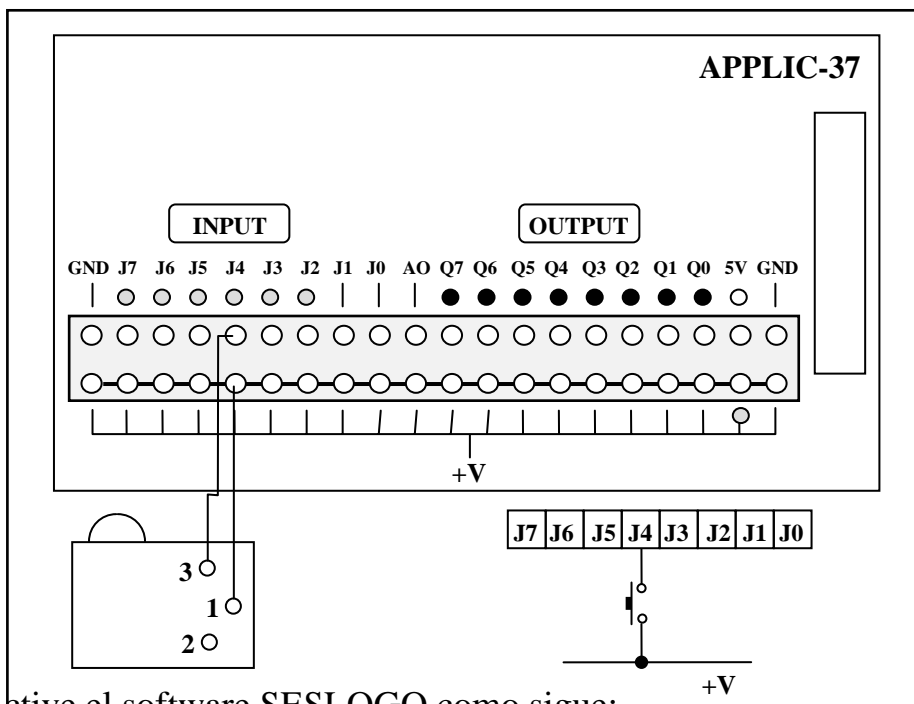
Paso 1: Conecte la interfaz a la PC y a la energía.

Paso 2: Prepare una lámpara y un interruptor con conectores de banana.

Paso 3: Conecte el circuito eléctrico como sigue:



Paso 4: Conecte un interruptor en la base del dispositivo y conéctelo con alambres a los canales J4 y GND como sigue:



Paso 5: Active el software SESLOGO como sigue:

Compruebe que la interfaz APPLIC-37 está conectada adecuadamente a la computadora y a la red.

Compruebe que el interruptor selector está en el lado RST.

Presione el pulsador RST en la parte superior de la interfaz.

Todas las luces rojas deben estar encendidas.

Encienda la computadora.

Haga doble clic en el ícono de SESLOGO.

Paso 6: Compruebe que un círculo verde aparece en la esquina derecha superior.

Paso 7: Si aparece un círculo rojo, siga las instrucciones descritas en la sección instalación del software.

Paso 8: Muévase a la pantalla Modo Directo con el botón



Paso 9: Cambie el número del canal en el botón Entrada Digital número 4.



Paso 10: Pulse sobre el botón Entrada Digital.

Paso 11: Pulse sobre el botón READ (LEER) que aparece en la ventana de Comandos.

El número 0 aparecerá en el botón.

Paso 12: Presione el pulsador, que está conectado al canal de entrada J4.

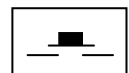
Mantenga presionado el pulsador y pulse nuevamente el botón LEER.

Esta vez aparecerá el número 1 en el botón.

Paso 13: Muévase al estado de Modo Fuera de Línea.

Las instrucciones relacionadas con el canal de entrada digital aparecerán en la ventana de los comandos, y ellas son:

Espera para 0, espera para 1, si es igual a 0 ir a ..., si es igual a 1 ir a ...




Paso 14: Cree un nuevo botón llamado SWITCH (Interruptor) con el dibujo .


Paso 15: Construya el procedimiento siguiente, que espera para que el pulsador sea presionado y luego CONECTE y DESCONECTE la lámpara.

DIGITAL IN 4	Wait Until 0
-----------------	-----------------

DIGITAL OUT 3	ON
------------------	----

 10

DIGITAL OUT 3	OFF
------------------	-----

 10

 RESTART
--

Paso 16: Guarde el procedimiento bajo el nombre INTERRUPTOR.

Paso 17: Ejecute el procedimiento.

Nada sucederá a la lámpara.

Paso 18: Presione el pulsador.

La lámpara centelleará tan pronto como presione el pulsador.


Paso 19: Pare la ejecución del procedimiento.

Paso 20: Cambie el procedimiento de modo que cada vez que presiona y suelta el pulsador haga que la lámpara centellee una vez.


DIGITAL IN 4	Wait Until 0
-----------------	-----------------

DIGITAL IN 4	Wait Until 1
-----------------	-----------------

DIGITAL OUT 3	ON
------------------	----

 10

DIGITAL OUT 3	OFF
------------------	-----

 10

 RESTART
--

Paso 21: Guarde el procedimiento.

Paso 22: Ejecute el procedimiento.


La lámpara centelleará solo una vez después de presionar y soltar el pulsador.

Cada vez que se presiona y se suelta se producirá un centelleo de la lámpara.

Compruebe esto.

Contador:

El cambio de estado a 1 y su caída a 0 se llama pulso. La interfaz tiene un mecanismo propio para contar los pulsos recibidos en sus entradas. Cada entrada tiene un contador interno, que cuenta independientemente los pulsos recibidos en la entrada.

Paso 23: Pulse sobre el botón counter  (contador).


Los siguientes comandos del contador aparecerán en la ventana de Comandos.

0 – reset (poner en cero), Wait Until (Esperar Hasta) >, Wait Until <, If bigger than...(Si es mayor que)....., If smaller than...(Si es menor que).....

En los sistemas de control tratamos de evitar comprobar la igualdad para saber si hay valores que puedan pasar el valor de la igualación.

Necesitamos recordar que la comprobación de la condición es hecha solamente cuando la computadora alcanza la instrucción de la condición. Puede presentarse una situación cuando presionamos el pulsador por varias veces y el estado del contador pasó la igualdad.

Paso 24: Cambie el procedimiento de tal manera que solamente después de 3 pulsaciones, la lámpara centelleará una vez.

COUNTER 4	
--------------	---

COUNTER 4	Wait Until > 2
--------------	-------------------

DIGITAL OUT 3	ON
------------------	----

 10

DIGITAL OUT 3	OFF
------------------	-----

 10

 RESTART
--

Paso 25: Guarde el procedimiento bajo el nombre SWITCH (Interruptor).

Paso 26: Ejecute el procedimiento.

Paso 27: Presione el pulsador tres veces.

La lámpara centelleará una vez.

Paso 28: Compruebe si después de cada tres pulsaciones, la lámpara centellea una vez.

Paso 29: Para el procedimiento.

Condiciones:

Las instrucciones de la condición son instrucciones condicionales de ejecución. Se ejecuta la instrucción si existe la condición escrita en la instrucción.

Las instrucciones en la instrucción condicional de SESLOGO deben ser procedimientos.

Paso 30: Cree un nuevo botón llamado Q3 ON con un círculo rojo lleno.

Paso 31: Guarde el procedimiento de la instrucción bajo este botón.

DIGITAL OUT 3	ON
------------------	----

Paso 32: Cambie el procedimiento SWITCH al siguiente:

DIGITAL IN 4	? = 1	● Q3 ON
-----------------	----------	------------

↻ RESTART

Paso 33: Ejecute el procedimiento SWITCH.

Nada sucederá.

Paso 34: Presione el pulsador.

La lámpara se ENCENDERÁ y no se APAGARÁ nuevamente.

Paso 35: Pare el procedimiento.

Para apagar la lámpara, necesitamos dar la instrucción de apagado.

Paso 36: Cree un nuevo procedimiento llamado Q3 OFF con un círculo rojo vacío.

Paso 37: Guarde la instrucción siguiente bajo el procedimiento Q3 OFF:

DIGITAL OUT 3	OFF
------------------	-----

Paso 38: Cambie el procedimiento SWITCH por el siguiente:

DIGITAL IN 4	? = 1	● Q3 ON
-----------------	----------	------------

DIGITAL IN 4	? = 0	○ Q3 OFF
-----------------	----------	-------------

↻ RESTART

Paso 39: Ejecute el procedimiento SWITCH.

Esta vez, presionando el pulsador se encenderá la lámpara y soltando el pulsador la lámpara se apagará.

Paso 40: Compruebe la reacción del sistema cuando el procedimiento se ejecuta desde la computadora.

Paso 41: Descargue el procedimiento a la interfaz y ejecútelo.

Ahora compruebe la reacción del sistema.

El sistema reacciona como si el pulsador está conectado a la lámpara.

Variables:

Todos los números que escribimos en las diferentes instrucciones tales como: el valor para la comparación, regresar un número de veces etc., son números, que se fijan mientras se escribe el programa. Estos números se llaman constantes.

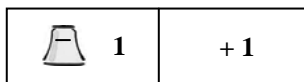
A veces necesitamos que el programa se adapte a cierta situación o que aprenda cierto valor. En este caso, utilizamos la célula de memoria donde los valores tales como éstos se almacenan.

El SESLOGO tiene tres botones  , que representan tres celdas de memoria.

Paso 43: Escribiremos un procedimiento donde el sistema comprueba cuánto tiempo tomará hasta que un botón se presiona y enciende una lámpara, por consiguiente, después de presionar el botón.

Cree un botón llamado COUNT  (Contar).

Paso 44: Escriba el procedimiento siguiente:

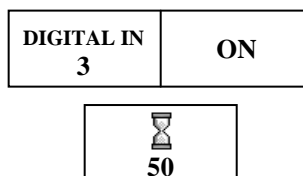


Guarde el procedimiento bajo el botón COUNT.

Este procedimiento agrega 1 a la variable 1.

Paso 45: Cree un nuevo botón llamado DELAY .



Paso 46: Escriba el procedimiento siguiente:



Guarde el procedimiento bajo el botón DELAY.

Paso 47: Cree un nuevo botón llamado LAMP  (lámpara).

Paso 48: Escriba el procedimiento siguiente:



 1 REPEAT	= DELAY
DIGITAL OUT 3	OFF
 1	0

Guarde el procedimiento bajo el botón LAMP.

Paso 50: Cree un nuevo botón llamado CNT&LAMP



Paso 51: Escriba el procedimiento siguiente:

DIGITAL IN 4	? = 0	<input type="checkbox"/> COUNT
DIGITAL IN 4	? = 1	<input type="radio"/> LAMP
 5		
 RESTART		

Guarde el procedimiento bajo el botón CNT&LAMP.

Este procedimiento comprueba al pulsador. Mientras el pulsador no se presiona, llama al procedimiento CONTAR, que incrementa la variable 1.

Cuando se presiona el pulsador, se llama al procedimiento de la LÁMPARA, que enciende la lámpara por una duración igual al tiempo que pasó hasta que el pulsador fue presionado. El procedimiento de la LÁMPARA reajusta la variable.

Paso 51: Ejecute el procedimiento CNT&LAMP desde la computadora y desde la interfaz después de la descarga, por varias veces. Cada vez que, presione el pulsador después de esperar diferentes períodos de tiempo y compruebe la reacción del sistema.

Capítulo 3 - Programación en Ensamblador

3.1 Introducción

El 8051 es un microprocesador algo complicado. Siga las explicaciones a su propio ritmo, puesto que los ejercicios harán más fácil comprender los aspectos teóricos. Se recomienda repasar de nuevo éstos después de haber realizado aquéllos .

La CPU - el corazón del sistema- funciona siguiendo un programa. Su "plan" de trabajo consta de instrucciones codificadas en números binarios de 8 bits contenidos en celdas de memoria secuenciales. La CPU extrae un número binario de una celda, lo decodifica y ejecuta la instrucción que representa.

Cada celda de memoria tiene una dirección exclusiva, que es un número binario de 16 bits. La CPU está conectada a las celdas de memoria por medio de tres BUSES: un bus de direcciones, un bus de datos y un bus de control. Cada bus es un conjunto de líneas que llevan números binarios.

Todas las celdas de memoria están conectadas a los buses.

Cuando la CPU quiere escribir un dato en una celda de memoria, hace lo siguiente:

Registra en el bus de direcciones la dirección de la celda de memoria de que se trate.

Declara que se encuentra (la CPU) en modo de ESCRITURA, y selecciona el tipo de memoria por medio del bus de control.

Escribe el dato en el bus de datos.

La celda de memoria que identifica como propios el tipo de memoria del bus de control y la dirección del bus de direcciones recoge el dato (después del paso c), que pasa así a reemplazar al contenido anterior de la celda.

El contenido de las celdas de memoria no son borradas.

A continuación se dan los pasos tomados por el CPU, cuando quiere anotar información en la memoria:

- a) La dirección de la celda requerida es anotada por el CPU en el bus de direcciones.
- b) Escribe la información en el bus de información.
- c) El CPU declara que está en estado de escritura, y selecciona el tipo de memoria por medio del bus de control.

La celda de memoria que identificó la dirección en el bus de direcciones su propia dirección, así como su tipo de memoria en el bus de control, coleccionará la información.

La nueva información reemplazará el contenido de la celda de memoria.

Como se ve, un microordenador es un sistema sincronizado, provisto de un reloj que establece la velocidad y marca el paso de los distintos componentes de la máquina.

Como ya se ha dicho, un programa de ordenador es una serie de números binarios almacenados en celdas de memoria consecutivas. El orden de los números define lo que hace el programa y las operaciones que debe ejecutar la CPU.

Todos los programas tienen una dirección de partida, que es aquella en la cual se encuentra la primera instrucción.

Para que la CPU ejecute un programa, debe dirigirse a la dirección correspondiente a la primera celda de dicho programa, a partir de la cual avanzará siguiendo las demás instrucciones.

Algunas de las celdas de la memoria del microordenador admiten operaciones de lectura y escritura, y constituyen la llamada memoria RAM (memoria de LECTURA/ESCRITURA).

El contenido de estas celdas desaparece cuando se corta la alimentación eléctrica. Al conectarla de nuevo, el contenido de las celdas es aleatorio.

Hay otras celdas que conservan su contenido incluso después de desconectar la alimentación eléctrica.

En ellas no puede escribirse nada, y conforman la llamada memoria ROM o de sólo lectura.

En ellas se graban los programas y datos permanentes del sistema.

El microprocesador 8051 tiene tres áreas de memoria distintas: área de PROGRAMAS, área de RAM interna y área de DATOS.

El programa del ordenador se almacena en el área de PROGRAMA, que por lo general es de tipo ROM; comprende las direcciones 0000 - FFFF, con una capacidad de 64 Kbytes.

El 8051 tiene una ROM de 4 Kbytes. El software que contiene debe grabarse en el chip de la CPU durante la fabricación.

El microcontrolador 8031 es idéntico al 8051, con la sola diferencia de que carece de ROM interna, aunque trabaja con otra externa.

Mientras ejecuta el programa, la CPU necesita un área de RAM para almacenar datos variables. Los creadores del 8051 colocaron una RAM de 256 bytes en la CPU, 128 de los cuales sirven como área de almacenamiento de datos o MEMORIA DIRECTA, como también se llama.

Los primeros 32 bytes de este área están organizados en 4 bloques, formado cada uno de ellos por ocho registros de 8 bits llamados R0-R7. Naturalmente, el direccionamiento de esta área de la memoria no se hace por medio de los buses externos del microprocesador.

Los otros 128 bytes de RAM interna son REGISTROS ESPECIALIZADOS.

Ocasionalmente, el microprocesador necesita un área de RAM de más de 128 bytes, situación típica de un microordenador utilizado para procesar grandes cantidades de datos. Cuando esto ocurre, pueden añadirse al 8051 unidades de RAM externas, que se conectan en paralelo a los buses de direcciones, datos y control.

Estas unidades pueden también llenar el área 0000-FFFF o área de DATOS del 8051. La CPU dispone de líneas para indicar si está direccionando el área de programa o el área de datos. En consecuencia, cada unidad de memoria dispone de un circuito que decodifica las líneas y las activa.

La activación del sistema o la pulsación de la tecla <RST> remiten a la CPU a la dirección 0000 del área de PROGRAMA, donde se encuentra la ROM del sistema. Esta dirección contiene una instrucción que remite a la CPU a un programa llamado PROGRAMA DE CONTROL.

La CPU tiene instrucciones especiales que le permiten direccionar tanto el área externa de DATOS como el área DIRECTA. La unidad de control de la CPU decodifica la instrucción recibida y la ejecuta en consecuencia.

Para realizar los ejercicios, se utilizan números hexadecimales en lugar de binarios. Este sistema de numeración de base 16 es una forma abreviada de presentación o codificación de números binarios.

3.2 Controlador SES 8051

APPLIC-37 (la tarjeta de robótica) es una tarjeta de la CPU 89C52 (miembro de la familia 8051), que fue diseñada como controlador para robótica automatizada y sistemas de máquinas. La tarjeta incluye todas las unidades de E/S y controladores, que son necesarios para controlar los sistemas. La tarjeta viene con un sistema operativo incorporado en la ROM interna de su microcontrolador. La tarjeta está conectada directamente con la PC en comunicación serie.

En este capítulo, describiremos la forma del funcionamiento del Applic-37 trabajando con una computadora y el paquete de software de apoyo, mientras practica y aprende Lenguaje Ensamblador y de Máquina. Este paquete de software incluye un Depurador del Ensamblador del Editor de Pantalla Completa.

Instalación del software de SES51C:

El SES51C es un programa de uso bajo WINDOWS. No tiene ninguna protección y solamente su copiado hace la instalación.

- 1) Encienda la PC.
- 2) Copie los subdirectorios de SESS51, USR y TMP del CD al disco duro.
- 3) Cree un acceso directo con SESS51C Spanish.exe.
- 4) Coloque este acceso directo en la ventana de escritorio.

Para hacer funcionar el software de SES51C usted tiene que hacer clic en el icono de este acceso directo.

Nota:

Los subdirectorios USR y TMP son para el compilador C y deben ser subdirectorios principales.

El subdirectorio USR incluye programas de muestra y archivos doc para el compilador C.

3.3 Operando el controlador con la PC

3.3.1 El cable de comunicación serial (CBL-1) es un cable de 3 alambres con un conector de tipo-D en un lado y un enchufe de audífono en el otro lado.

Conecte el conector de tipo-D al puerto de comunicación serial del PC (COM1 o COM2) y el enchufe de audífono a la entrada del audífono.

3.3.2 Conecte la fuente de alimentación con la entrada de energía de Applic-37.

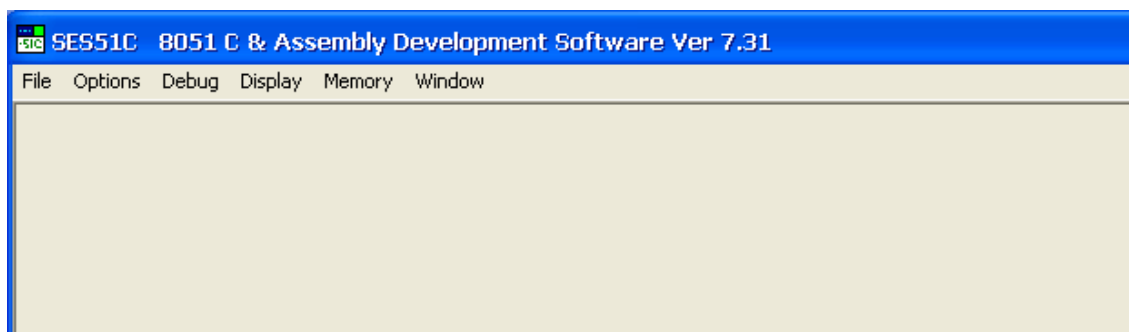
3.3.3 Conecte la fuente de alimentación a la red.

3.3.4 Conecte el cable a la toma de corriente.

3.3.5 Encienda la unidad y pulse la tecla <RST> en el controlador.

3.3.6 Haga clic con el ratón en el icono de SES51C en su escritorio.

Una pantalla introductoria aparecerá en la pantalla:



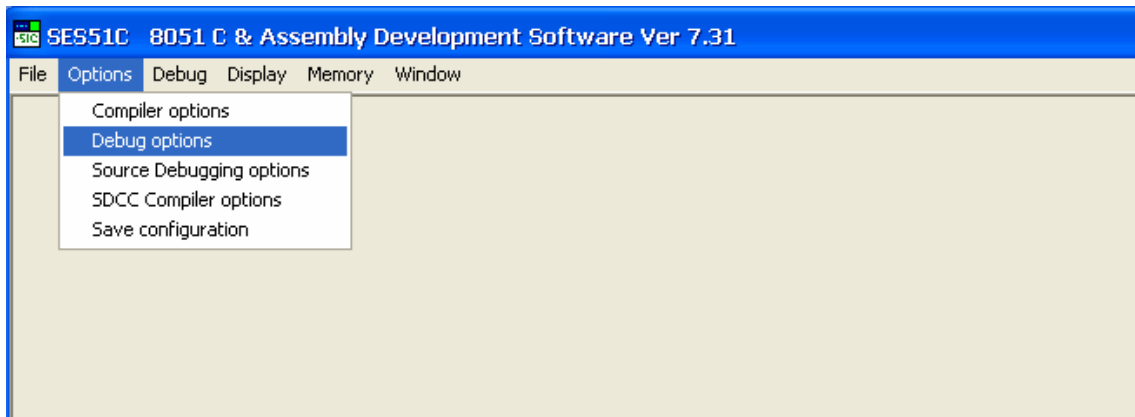
3.3.7 Use el ratón o pulse [Alt + A] en ARCHIVO y observe las funciones de este.

Mueva el ratón hacia la derecha y a la izquierda y observe las otras funciones.

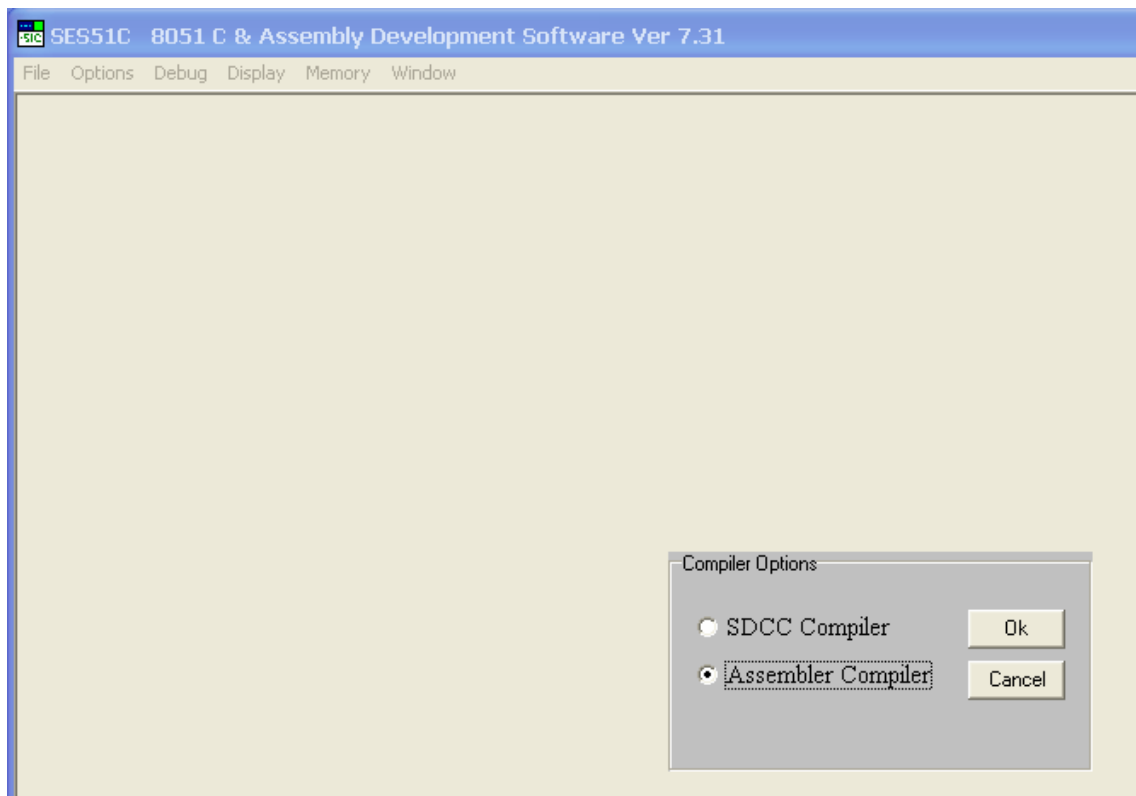
Para salir pulse <ESC>.

3.3.8 Seleccione OPCIONES.

Seleccione la función OPCIONES DEL COMPILADOR.



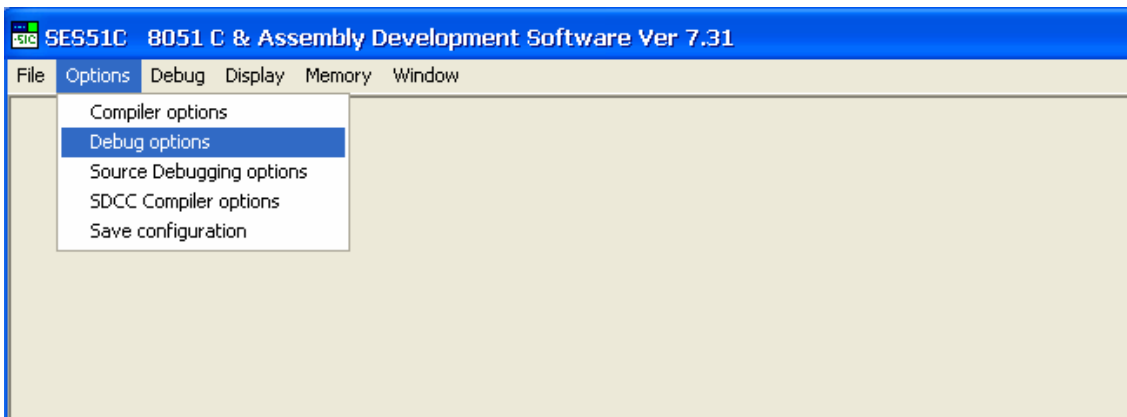
Aparecerá la siguiente pantalla.



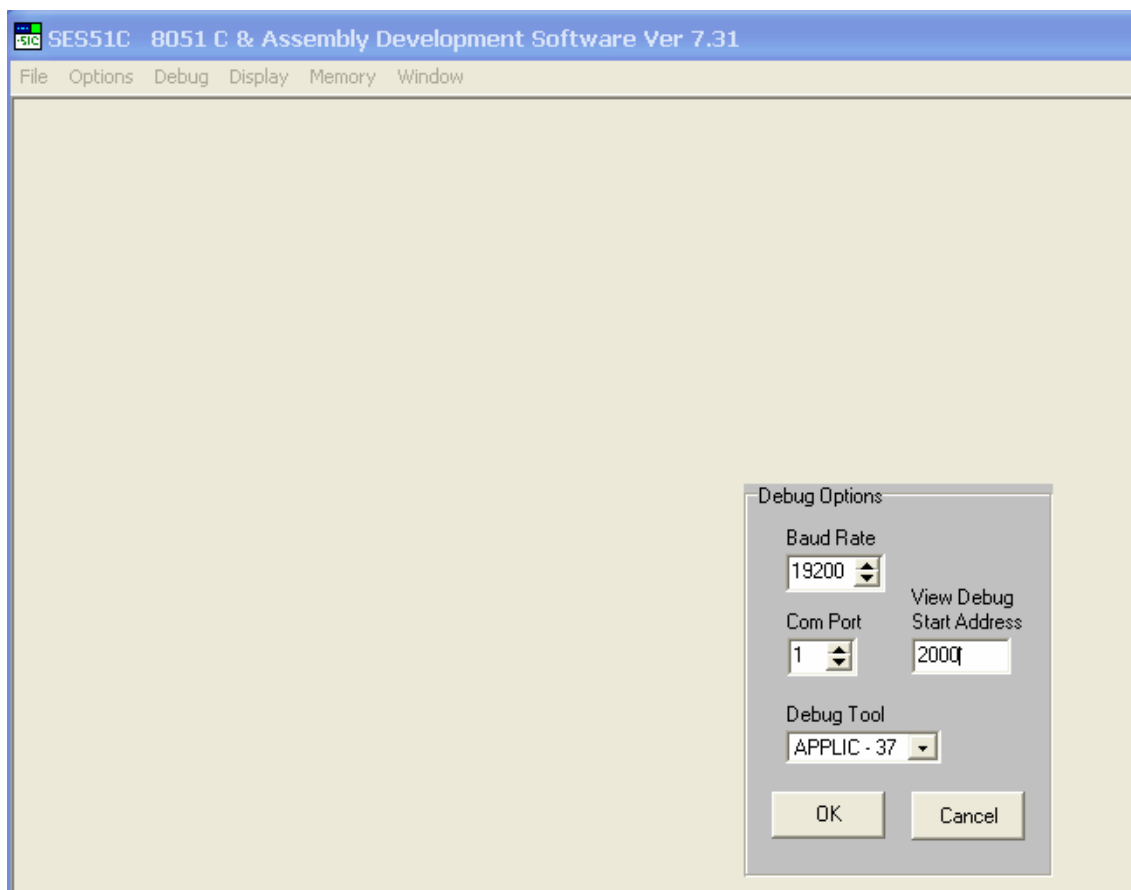
3.3.9 Seleccione "Compilador Ensamblador" y Confirme.

3.3.10 Seleccione OPCIONES.

Seleccione la función OPCIONES DEL DEPURADOR.



En esta pantalla podemos definir el puerto serial de la PC, la Velocidad de Transmision y Direccion de Inicio.



Para moverse de una celda a otra utilice <TAB> o <Shift TAB> o el ratón.

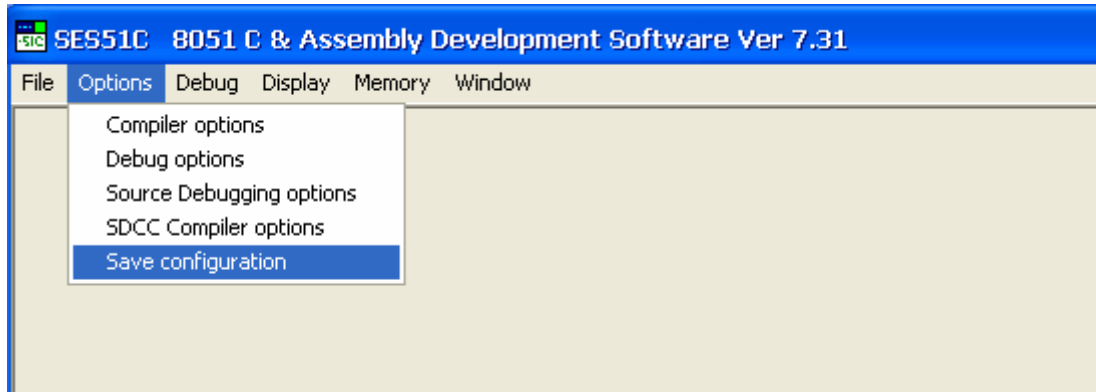
Para moverse dentro de la celda utilice las flechas

APPLIC-37 opera en 19,200 baud.

Seleccione el "Puerto Com" y pulse <ENTER> o seleccione <OK> con el ratón.

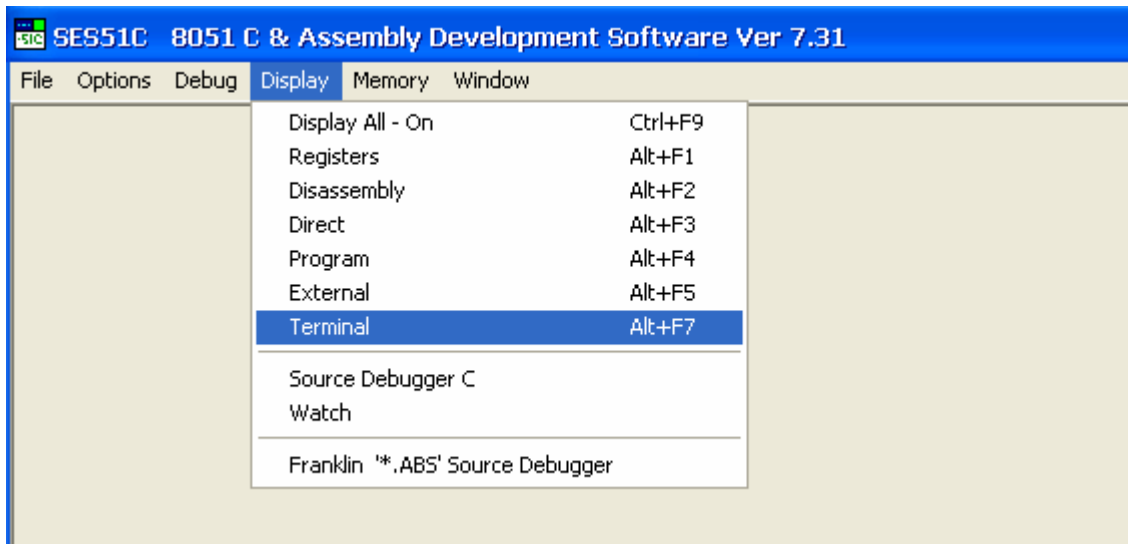
3.3.11 Seleccione OPCIONES nuevamente.

Seleccione GUARDAR CONDIGURACIONES.



Las nuevas definiciones han sido salvadas en el archivo de configuración y serán las mismas para todas las sesiones hasta que se cambien nuevamente.

3.3.12 Seleccione la sub-función TERMINAL en la función VISUALIZACION.



Esta función abre una pantalla terminal de ventana.

Usted puede agrandar esta ventana y ubicarla en cualquier lugar en la pantalla.

- 3.3.13 Presione el botón <RST> en el APPLIC-37. Después de un segundo, se enviará al terminal un mensaje de abertura.

Teclee sobre el terminal de pantalla.

- 3.3.14 Teclear el siguiente comando:

***OUFF00,01<ENTER>**

Esto significa el número 01 en el puerto de salida en la dirección FF00.

El LED más significativo de la derecha se ENCENDERÁ.

- 3.13.14 Intente hacer salir diversos números en el puerto de salida.

3.4 Escribir en ensamblar

Cuando escribimos rutinas cortas para la práctica o programas cortos de prueba, podemos escribirlos con la ayuda del DEPURADOR.

Podemos también almacenar estas rutinas en un disquete y cargarlas del disquete.

Pero cuando estamos interesados en escribir un programa que deseamos almacenar en forma ordenada y que podamos actualizarlo fácilmente, necesitamos hacerlo en forma diferente.

El software se debe escribir en el lenguaje del microprocesador. Puede ser escrito también en un lenguaje de alto nivel, pero nos relacionaremos con esto más adelante.

El lenguaje assembly contiene, además de la instrucción mnemónica, el uso de etiquetas, símbolos, directivas, observaciones, etc. Esto hace al programa escrito en lenguaje assembly absolutamente flexible y fácil de leer y entender.

El programa en assembly se escribe usando un editor de texto. Este texto se llama **PROGRAMA FUENTE**.

Después de que el programa fuente se escriba y se almacene en el disquete, ejecutaremos el **PROGRAMA DEL ENSAMBLADOR** sobre él.

El programa del ensamblador lee el programa fuente, línea tras línea, y lo traduce a lenguaje de máquina.

El proceso de traducir requiere generalmente un ensamblador de dos pasos.

En el primer paso, el software del ensamblador forma una tabla con los valores para todas las etiquetas, símbolos y variables.

En el segundo paso, forma el programa en lenguaje de máquina, conteniendo los valores de la tabla.

Se utilizan las etiquetas como nombres de líneas que ramifican y los símbolos se utilizan como nombres para las diversas variables y parámetros.

El ensamblador calcula las direcciones y las planta en los lugares correctos.

El programa en lenguaje de máquina se llama **PROGRAMA OBJETO**.

Además de él también recibimos un **PROGRAMA LISTA** consistiendo en el programa objeto y en el programa fuente.

Lo que sigue es un programa fuente en assembly que transfiere el estado de los conmutadores a los LED en un bucle infinito.

```
IOPOR EQU 0FF00H
T
ORG 2100H
SWLD: MOV DPTR,#IOPORT
SWLD1 MOV A,@DPTR
: X
MOV @DPTR,A
X
SJMP SWLD1
END
```

La regla de distinción entre un número y un símbolo es:

Cada número comenzará con un dígito y cada etiqueta o símbolo comenzará con una letra.

A un número, que comienza con una letra, le agregamos 0 al principio. La letra H al final del número indica que es un número hexadecimal.

En el programa utilizamos tres directivas del ensamblador: EQU, ORG y EXTREMO.

Las directivas también se llaman pseudo operaciones.

Aparecen en el programa en assembly como instrucciones, pero no pertenecen al sistema de instrucción del microprocesador.

Nos ayudan al escribir el programa, para darle flexibilidad y una forma generalizada.

EQU (igual) - definir un símbolo para un valor numérico:

Esta directiva asigna un valor numérico fijo al símbolo.

Esto permite el uso de símbolos en el programa en vez de valores numéricos fijos.

Por ejemplo:

```
PORTA EQU 3401H
```

En el programa, escribiremos solamente el símbolo (en nuestro ejemplo PORTA) y no el valor numérico.

Si deseamos cambiar el valor de PORTA necesitamos cambiar solamente la directiva y el ensamblador realizará el cambio en el programa objeto dondequiera que el símbolo PORTA aparezca.

Usted puede también utilizar la frase siguiente:

```
PORTB EQU PORT1+1
```

ORG - directiva del origen:

Esta directiva es acompañada por una dirección.

Le indica al ensamblador en que dirección de inicio localizar la sección del programa que sigue.

Los Bytes traducidos serán puestos en las direcciones consecutivas a partir de esta dirección.

Un número de instrucciones marcando origen pueden aparecer en un programa.

Por ejemplo:

ORG 0100H

.

Sección 1 del programa

.

.

ORG 300H

.

Sección 2 del programa

.

.

El ensamblador sabe la posición de cada instrucción referente a la directiva relevante de ORG, y calcula su dirección. De esta manera, sabe la dirección de una etiqueta, y sabe calcular desplazamientos de ramificación.

El ensamblador crea un programa de dos secciones del programa. El ensamblador crea el programa de destino en un formato especial llamado: formato HEX.

Este formato permite al archivo ser transmitido vía comunicación de una computadora a otra. La computadora de recepción sabe dónde poner las diversas secciones del programa.

END - directiva de final del programa:

Esta directiva indica el final del programa fuente al ensamblador. Esta directiva es vital para el ensamblador, que realiza dos pasos sobre el programa fuente. Cuando se alcanza la directiva de FINAL, comienza el segundo paso.

Si un programa no es limitado por una declaración de FINAL, el ensamblador indicará un error y abortará la traducción del programa.

Un programa, que fue creado en lenguaje de máquina se llama un programa objeto. Diversos ensambladores forman programas objeto en formas levemente diversas, según sus programas de carga. Debemos recordar que el programa objeto es destinado para ser cargado en la memoria del sistema, donde debe funcionar. Por esa razón contiene, además de códigos en lenguaje de máquina, los parámetros conectados con la localización de las diversas partes del programa.

Haremos referencia al paquete de software SES51W. Es obligatorio que el nombre del programa fuente tenga la extensión .A51. Por ejemplo, si escribimos el programa fuente antedicho bajo el nombre de:

SWLD.A51

Cuando ejecutamos el ensamblador, forma un programa objeto, que recibe el nombre del programa fuente con la extensión, HEX.

Por ejemplo, para el archivo SWLD.A51 recibiremos un archivo SWLD.HEX.

Además del programa objeto, el ensamblador también crea un archivo con tabla de símbolos (con todos los símbolos y etiquetas y sus valores) con la extensión. SYM.

Por ejemplo:

SWLD. SYM

El ensamblador también forma otro archivo llamado archivo de LISTA, que consta del programa fuente y del programa objeto. Este archivo consigue la extensión. LST.

Por ejemplo:

SWLD. LST

Si hay errores en el programa, el ensamblador los formará en un archivo especial con la extensión .ERR.

Por ejemplo:

SWLD. ERR

En la sección siguiente, ejercitaremos todas las etapas de desarrollo del software en assembly.

3.4.1 Seleccione la función FILE y la sub-función NEW.

Se abrirá una ventana sin nombre (UNTITLED) con editor de pantalla completa.

El editor es nuestro procesador de textos para escribir nuestro programa fuente.

3.4.2 Teclee el programa siguiente:

```

PLED    EQU    0FF00H
        ORG    2000H
        LJMP   BLINK
        ORG    2100H
BLINK:  MOV    DPTR,#PLE
        D
BLINK1  MOV    A,#0FFH
:
        MOVX   @DPTR,A
        LCALL  DEAY
        MOV    A,#0
        MOVX   @DPTR,A
        LCALL  DELAY
        SJMP  BLINK1
;
DEAY:   MOV    R2,#0FFH
LOOP1:  MOV    R1,#0FFH
LOOP2:  DJNZ   R1,LOOP2
        DJNZ   R2,LOOP1
        RET
        END

```

Nota:

Cada etiqueta se debe colocar al principio de la línea (sin ningún espacio antes de ella).

Cada instrucción debe estar por lo menos después de un espacio.

3.4.3 Después de teclear el programa entero juegue un poco con las otras funciones del editor, EDIT y SEARCH.

Observe las sub-funciones.

3.4.4 **Funciones del Editor (EDIT):**

El editor tiene un área en el fondo llamada sujetapapeles (clipboard).

Podemos cortar o copiar una sección del programa al sujetapapeles.

Para marcar una sección antes de cortar o de copiar, traiga el cursor al primer carácter de la sección deseada. Pulse la tecla <Shift> y manténgala presionada, mientras que usted mueve el cursor (con las teclas de flechas, <End>, <PgUp>, <PgDn> o <Home>) al final de la sección.

La sección será destacada.

Para copiar la sección destacada de la ventana del editor al sujetapapeles, elija la sub-función COPIAR o pulse <Ctrl + C>.

Para cortar la sección destacada de la ventana del editor sujetapapeles, elija la sub-función CORTAR o pulse <Ctrl + X>.

La sección destacada desaparecerá y será ahorrada en el fondo.

Cuando una sección está en el sujetapapeles, podemos agregarla y pegarla dondequiera en el texto sin ninguna limitación.

Mueva el cursor al lugar que usted desea agregar la sección y seleccione la sub-función PEGUR o pulse <Ctrl + V>.

El contenido del sujetapapeles será copiado y agregado al texto.

El contenido del sujetapapeles no cambiará.

El copiado o cortando otra sección al sujetapapeles substituirá la vieja sección en ella.

La sub-función DESHACER o <Ctrl + Z> deshará los cambios desde el último movimiento del cursor.

3.4.5 **Funciones de BÚSQUEDA (SEARCH):**

Para encontrar una secuencia en el texto utilizamos la función BUSCAR y la sub-función ENCONTRAR.

Se abrirá una ventana de la entrada de datos pidiendo una secuencia a buscar.

En el lado derecho podemos ver una flecha de "historia", que significa que si presionamos la tecla de la flecha hacia abajo, se abrirá otra ventana mostrando todas las secuencias de búsqueda anteriores.

Se puede elegir a uno de ellas en lugar de teclearla nuevamente.

Ingrese una secuencia (teclearemos generalmente una etiqueta, un símbolo o una directiva) y pulse <ENTER>. El editor buscará la secuencia de la posición del cursor hacia abajo.

Si la secuencia se encuentra será iluminada y el editor pondrá el cursor en el final de la secuencia.

Para buscar la misma secuencia de nuevo, usted puede elegir la sub-función BUSCAR OTRA VEZ o presionar F3.

Para sustituir una secuencia por otra, utilizamos la sub-función REEMPLAZAR.

Esta función abre dos campos de la ventana de la entrada de datos.

El primero pide una secuencia para encontrar y la segunda pide la secuencia de reemplazo.

El movimiento de un campo al otro se hace tecleando <Tab> o <Shift Tab>.

Antes de sustituir, el editor ilumina la secuencia que se sustituirá y pide la confirmación.

Para buscar la misma secuencia de nuevo, usted puede elegir la sub-sub-función BUSCAR OTRA VEZ o presionar F3.

3.4.6 Compruebe que el texto del programa siga siendo igual que el programa en el paso 3.4.2.

3.4.7 Para compilar un archivo a lenguaje de máquina, y crear un programa objeto, ejecutaremos el ensamblador.

El ensamblador puede trabajar solamente en un archivo de disquete, así que tenemos que guardar el archivo primero.

Seleccione la función ARCHIVO y la sub-función GUARDAR COMO.

Anote el nombre:

SWLD. A51

El texto será guardado bajo este nombre.

3.4.8 Seleccione la función COMPILAR y la sub-función COMPILAR.

El ensamblador compilará el archivo SWLD.A51 y creará un archivo HEX, también creará un archivo SYM y un archivo LST.

Anuncia el número de los errores que encontró, el número de líneas en el programa y el número de Bytes en el programa objeto de lenguaje de máquina.

3.4.9 Si se encuentran errores, podemos encontrarlos en un archivo con el mismo nombre y con la extensión: SWLD. ERR.

Seleccione la función VER y la sub-función MENSAJES DE ERROR.

Una ventana nueva mostrará el archivo SWLD.ERR.

Podemos movernos dentro de las ventanas usando <Ctrl + Tab>.

Use <Ctrl + Tab> para entrar al editor de pantalla y fijar los errores.

Use <Ctrl + Tab> nuevamente para volver a la ventana de los mensajes de error.

Cierre la ventana de los mensajes de error pulsando la X en la esquina de la ventana.

3.4.10 Podemos también los archivos LST, SYM y HEX, con la función VIEW.

Nota:

El compilador se puede ejecutar solamente en el archivo en el disco.

Si sus cambios no fueron guardados, no serán compilados por el compilador.

Para guardar los cambios, utilice la función ARCHIVO y la sub- función GUARDAR o <Ctrl + S>.

Llamando al ensamblador crea un guardado automático y la actualización del archivo.

Cuando se cambia y no se guarda el archivo, aparecerá la palabra "modificado" en la esquina inferior del fondo izquierdo de la ventana del editor.

Ejecutando el compilador guarda los cambios y la palabra "modificado" desaparece.

- 3.4.11 Para descargar un programa al controlador, debe estar en formato ASCII.

Compruebe que el controlador esté conectado con el puerto serial de la PC.

Pulse <RST> en el controlador.

- 3.4.12 Seleccione la función **DEPURAR** y la sub-función **DESCARGA Y PROGRAMA**.

El archivo será enviado al controlador y situado según su **ORG**. El archivo será visto en la ventana **TERMINAL**.

- 3.4.13 Teclar sobre el terminal de pantalla.

- 3.4.14 Tipear el comando siguiente:

***GO2000<ENTER>**

Los LEDs deben centellar.

- 3.4.15 Para parar el programa presione <RST>.

Capítulo 4 – Programación en Lenguaje C

Todos los ordenadores constan de dos elementos básicos: el HARDWARE - los elementos físicos, que se ven y se tocan - y el SOFTWARE.

Antiguamente, el diseño y la construcción del hardware se consideraban los aspectos más importante del desarrollo de un microprocesador; pero actualmente, con la normalización de los componentes electrónicos y la disminución de precios, dichos aspectos son relativamente sencillos.

Por el contrario, el software cobra cada vez más importancia en el diseño de sistemas, debido a la necesidad de incrementar y diversificar cada vez más las aplicaciones del microprocesador.

En el capítulo 1 estudiamos la arquitectura y los componentes del microordenador, es decir, el hardware. En este hablaremos del software.

La CPU procesa datos recibidos de la memoria y de las unidades de entrada con arreglo a un programa almacenado en la memoria.

El software es un conjunto de instrucciones que obligan al microprocesador a trabajar de una forma determinada.

Esta colección de instrucciones la escriben el diseñador del sistema y el usuario. La CPU es necesaria para adoptar decisiones cuando se procesan datos, pero no es ella la que toma las decisiones, sino que le son impuestas por el programa.

Para ello se verifican determinadas condiciones de acuerdo con las instrucciones, y se suscita una reacción que depende del resultado de la mencionada verificación.

Todos los microprocesadores tienen entradas y salidas.

Las salidas están en función de las entradas y vienen definidas por los programas almacenados en el ordenador.

Aunque la estructura física de éste sea relativamente sencilla, puede desempeñar funciones complejas.

Un microordenador puede sustituir a cualquier sistema digital, por complejo que sea.

Las ventajas de solucionar los problemas por medio del software son:

- a) Empleo de sistemas normalizados y, por tanto, baratos.
- b) Posibilidad de hacer modificaciones o ampliaciones fácilmente (con sólo cambiar el software).
- c) Utilización del mismo hardware para llevar a cabo numerosas funciones en serie.

4.1 Fases de la Escritura de Programas

Las fases de la escritura de programas son:

- a) Definición del problema.
- b) Algoritmo y diagramas de flujo.
- c) Escritura del programa.
- d) Ejecución y comprobación del programa.

Aunque recorrer todas las fases parezca lento y pesado, en realidad ahorra muchísimo tiempo.

Cuanta más atención se preste a las primeras fases, tantas menos sorpresas se cosecharán en las últimas.

a) **Definición del problema:**

En esta fase se recoge la información relativa al problema y a las condiciones impuestas por el ordenador. Se hace una lista de los datos útiles para la entrada y de los necesarios para la salida.

Todavía no se detallan las fases de ejecución del programa, dado que es normal que se produzcan cambios en el programa o en los datos de entrada o salida. Para que la solución del problema sea acertada, el programador ha de conocer bien las limitaciones del sistema.

b) **Algoritmo y diagrama de flujo:**

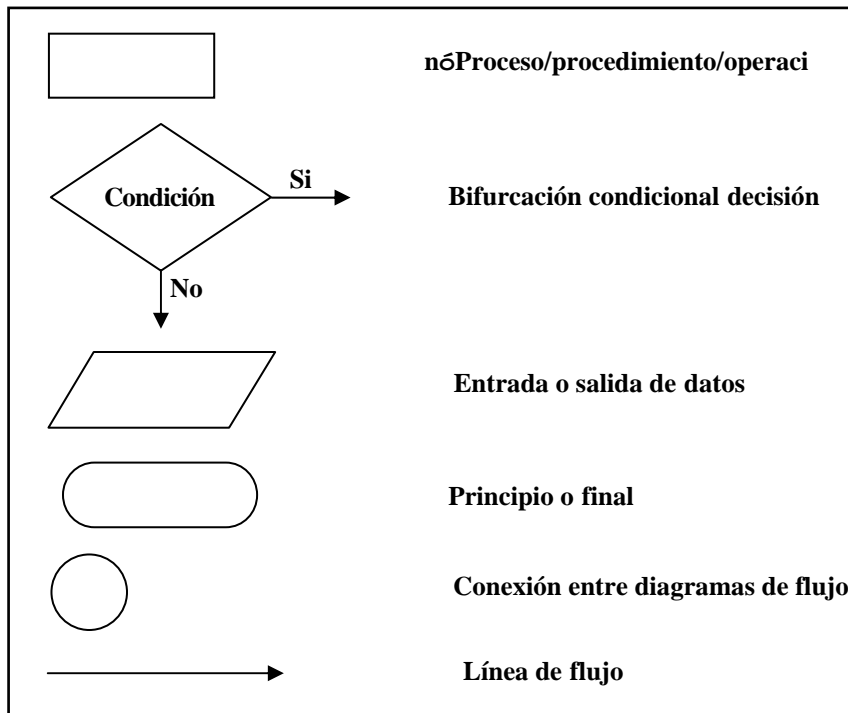
Un algoritmo es una especificación de las fases de ejecución de un problema, y se utiliza para definir una forma de resolver dicho problema y de recorrer las etapas que lo componen. El algoritmo se escribe en lenguaje corriente y en términos muy generales.

Un diagrama de flujo es una descripción gráfica de las fases del algoritmo. Las fases se representan mediante símbolos y se pasa de unas a otra con flechas.

Cuanto más detallada sea la especificación y más próxima esté a las instrucciones del programa, tanto más fácil será escribir éste.

El diagrama de flujo es la forma más fácil de entender un programa, sus condiciones y la forma en que se ejecuta; con su ayuda es también sencillo modificar o mejorar el programa original.

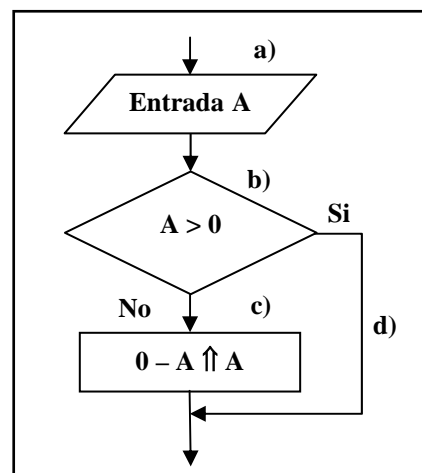
Los símbolos utilizados se corresponden con distintos tipos de instrucciones:



Veamos, por ejemplo, el esquema de un programa para determinar el valor absoluto de un número.

El algoritmo sería:

- Recibir el número.
- Comprobar si es positivo.
- En caso contrario, cambiarlo a positivo.
- En caso afirmativo, dejarlo sin cambiar.



Cuando se da la instrucción para ejecutar una operación, hay que indicar el destino al que debe transferirse el resultado.

El significado de la expresión $0-A \rightarrow A$ es: realizar la operación $0-A$ (cambiar de signo) y almacenar el resultado en A .

Una ventaja importante del ordenador es su capacidad para ejecutar una serie de operaciones varias veces seguidas (bucles) y para operar teniendo en cuenta condiciones muy numerosas.

Es difícil seguir o escribir un programa complejo sin ayuda del diagrama de flujo.

El programador tiene que conocer bien el lenguaje del microordenador y sus instrucciones para sacarle el máximo partido.

Hay bibliotecas de programas y subrutinas, y es aconsejable conocerlas y manejarlas para no tener que descubrir la pólvora.

c) **Escritura del programa:**

El programa se escribe en el lenguaje del microprocesador.

La CPU entiende sólo números binarios, que son los únicos que pueden entrar en la memoria. Pero sería muy complicado y laborioso escribir programas íntegramente en números binarios, y la solución de este inconveniente depende del nivel del microordenador.

La forma más inmediata es construir un sistema que convierta el código hexadecimal a binario; el programador escribiría las instrucciones y datos en código hexadecimal con ayuda de un teclado, y la unidad de entrada del microprocesador traduciría todo a números binarios (se trata de una operación de conversión muy sencilla).

Los programas escritos en números binarios o en código hexadecimal se dice que están escritos en lenguaje de máquina.

Pero hay otros lenguajes de programación, a saber:

- a) Lenguaje ENSAMBLADOR.
- b) Lenguajes de alto nivel.

d) Ejecución y comprobación del programa:

En esta fase se pone en marcha el programa y se comprueba su funcionamiento con varios datos de entrada para descubrir y corregir los errores.

Cuanto más se haya reflexionado durante las etapas de diseño, menos errores se observarán en esta .

Los errores de programación observados en la fase de pruebas se llaman "bichos" o "bugs" y se descubren y eliminan con ayuda de otros programas llamados depuradores o "debuggers".

4.2 Lenguaje Máquina

Para escribir programas en lenguaje máquina, hay que conocer dos cosas:

- a) Las instrucciones de la CPU y sus códigos.
- b) La dirección de memoria en la que se empieza a cargar el programa.

Junto a cada instrucción se escribe la dirección de la celda en la que debe introducirse.

Como ya se ha dicho, hay instrucciones que exigen la inserción de otro byte justo a continuación para que la CPU las ejecute; otras exigen la inserción de dos bytes más de información inmediata (y, por supuesto, hay otras que no necesitan ningún dato).

Estos valores necesarios se cargan en las celdas que ocupan las direcciones contiguas a la que alberga el código de la instrucción.

En el capítulo 2 practicamos con varias instrucciones y escribimos algunos programas en lenguaje máquina. Vamos aquí a resumir aquellos ejercicios.

Para que los programas sean fáciles de escribir y leer, se añaden a las instrucciones ETIQUETAS, CODIGOS MNEMOTECNICOS y COMENTARIOS.

En la práctica, los programas se disponen en forma de tablas como esta:

Dirección	Código	Etiqueta	Mnemotecnico	Notas(*)
5000	90 FF 00	START:	MOV DPTR,#0FF00H	;Cargar DPTR con las direcciones de las luces.
5003	74 0F		MOV A,#0FH	;Cargar A con 0F para encender las 4 luces.

La ETIQUETA sirve para poner nombres a las filas y facilitar su identificación.

No es preciso asignar etiquetas a todas las filas.

El CODIGO MNEMOTECNICO describe la esencia de la instrucción.

Consta de tres partes:

- a) OPERACION: suele ser una abreviatura de la palabra o las palabras que describen la operación de que se trate.
- b) OPERANDOS: son los datos o elementos con los que se ejecuta la operación.
- c) MODO DE DIRECCIONAMIENTO.

Notas:

En esta parte de la tabla se describe con el mayor detalle posible la naturaleza de la operación, para facilitar la interpretación del programa a quienes no lo han escrito o a su autor algunos días después de haberlo escrito.

4.3 Lenguaje Ensamblador

Es el más cercano al lenguaje máquina de todos los lenguajes de programación. Las instrucciones en ensamblador se escriben de forma muy similar a las instrucciones en código mnemotécnico, y se numeran a voluntad, sin que la numeración deba tener necesariamente en cuenta las direcciones reales.

En el programa hay que indicar la dirección inicial. En comparación con el lenguaje máquina, el ensamblador presenta muchas ventajas: no obliga a calcular desplazamientos (el desplazamiento es el número de posiciones de memoria sobre las que desea saltarse) en las bifurcaciones; es fácil añadir o quitar instrucciones; la ubicación del programa en la memoria del ordenador es flexible; es fácil localizar los errores.

En comparación con los lenguajes de alto nivel, el ensamblador tiene el inconveniente de que obliga a descomponer el algoritmo del problema en pasos elementales compatibles con el juego de instrucciones de la CPU.

La traducción de las instrucciones escritas en lenguaje ensamblador a código máquina (números binarios) se hace con un programa llamado ENSAMBLADOR.

4.4 Lenguajes de Alto Nivel

Se llama así a los lenguajes de programación que se parecen más al habla natural que al lenguaje máquina.

Con estos lenguajes es muy sencillo escribir programas para microprocesador. BASIC, FORTRAN, PL/M y PASCAL pertenecen a este tipo de lenguajes.

Cada una de las instrucciones escritas en lenguaje de alto nivel se convierte en varias en lenguaje máquina.

De la traducción se encarga un programa especializado llamado COMPILADOR, un software muy complejo que necesita grandes cantidades de memoria.

4.5 Operando el controlador con la PC

4.5.1 El cable de comunicación serial (CBL-1) es un cable de 3 alambres con un conector de tipo-D en un lado y un enchufe de audífono en el otro lado.

Conecte el conector de tipo-D al puerto de comunicación serial del PC (COM1 o COM2) y el enchufe de audífono a la entrada del audífono.

4.5.2 Conecte la fuente de alimentación con la entrada de energía del APPLIC-37.

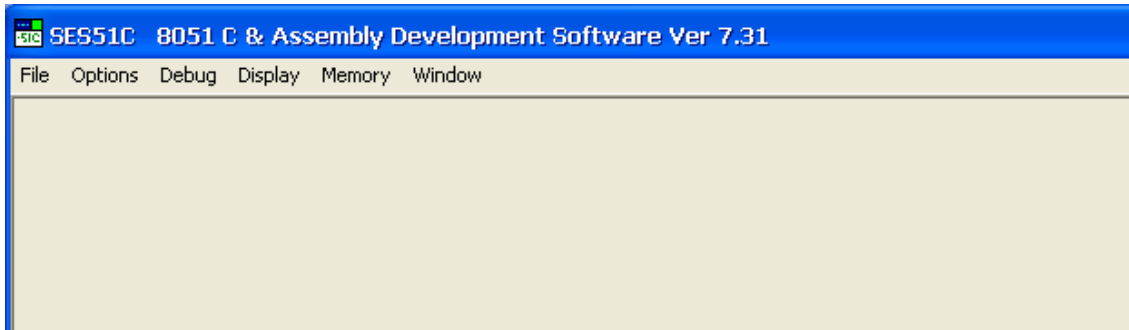
4.5.3 Conecte la fuente de alimentación a la red.

4.5.4 Conecte el cable a la toma de corriente.

4.5.5 Encienda la unidad y pulse la tecla <RST> en el controlador.

4.5.6 Haga clic con el ratón en el icono de SES51C en su escritorio.

Una pantalla introductoria aparecerá en la pantalla:



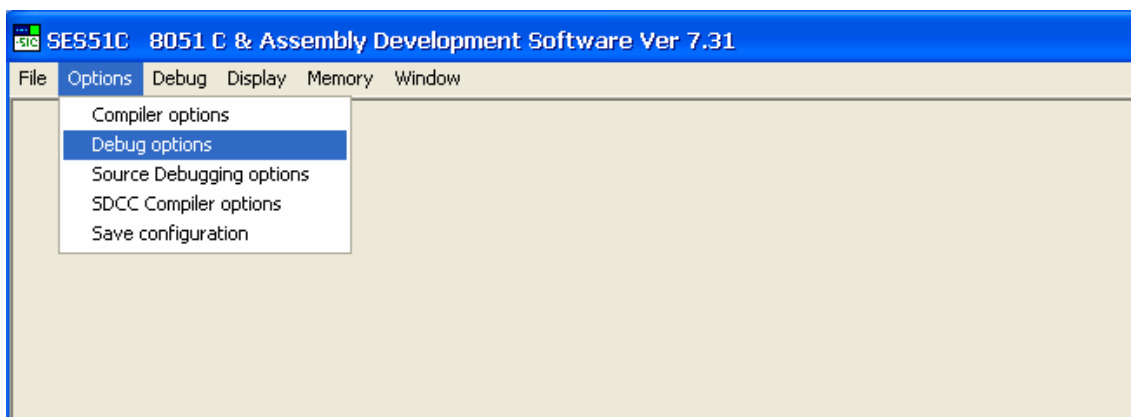
4.5.7 Use el ratón o pulse [Alt + A] en ARCHIVO y e observe las funciones de este.

Mueva el ratón hacia la derecha y a la izquierda y observe las otras funciones.

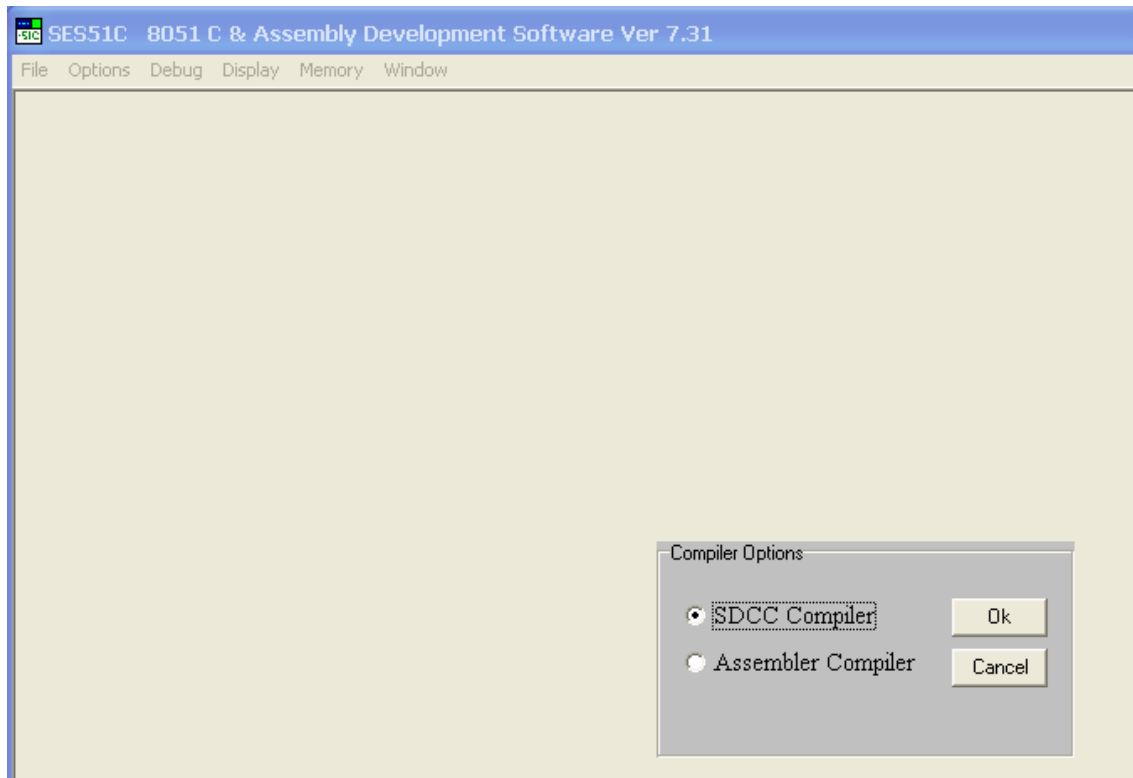
Para salir pulse <ESC>.

4.5.8 Seleccione OPCIONES.

Seleccione la función OPCIONES DEL COMPILADOR.



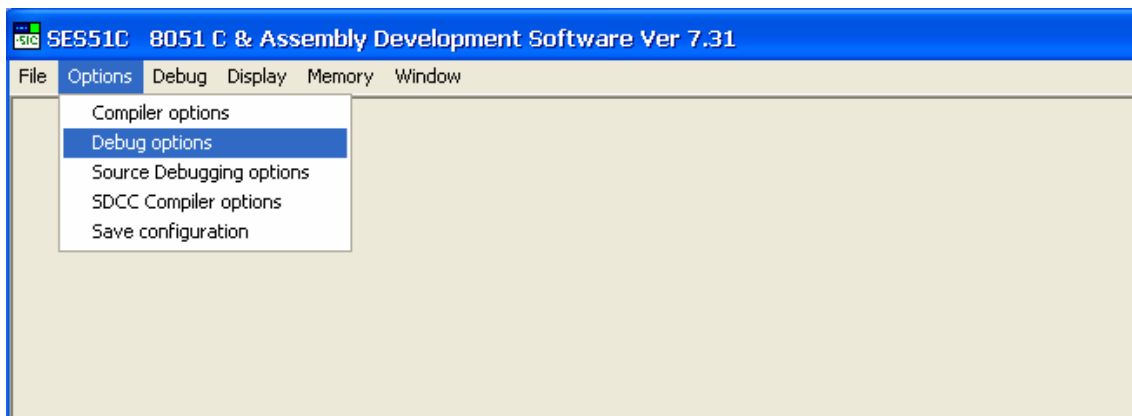
Aparecerá la ventana siguiente.



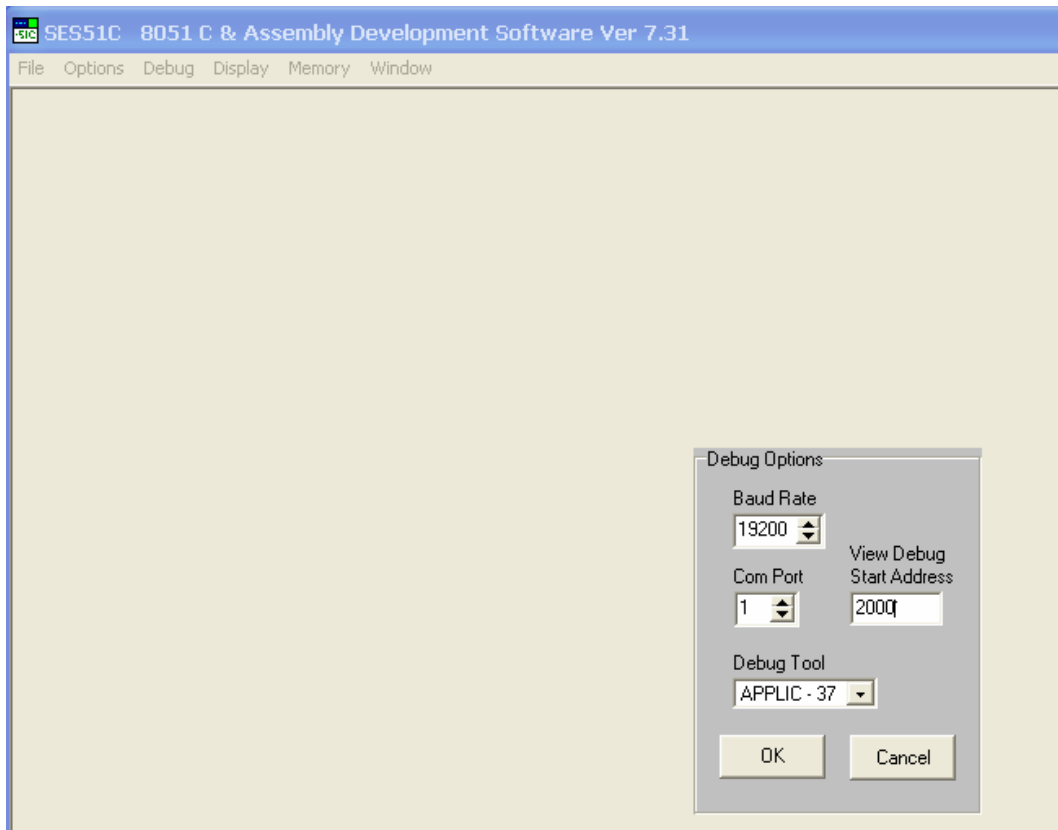
4.5.9 Seleccione 'Compilador SDCC ' y Confirme.

4.5.10 Seleccione OPCIONES.

Seleccione la función OPCIONES DEL DEPURADOR.



En esta pantalla podemos definir el puerto serial de la PC, el "baud rate" y el tipo de controlador que planeamos usar.



Para moverse de una celda a otra utilice <TAB> o <Shift TAB> o el ratón.

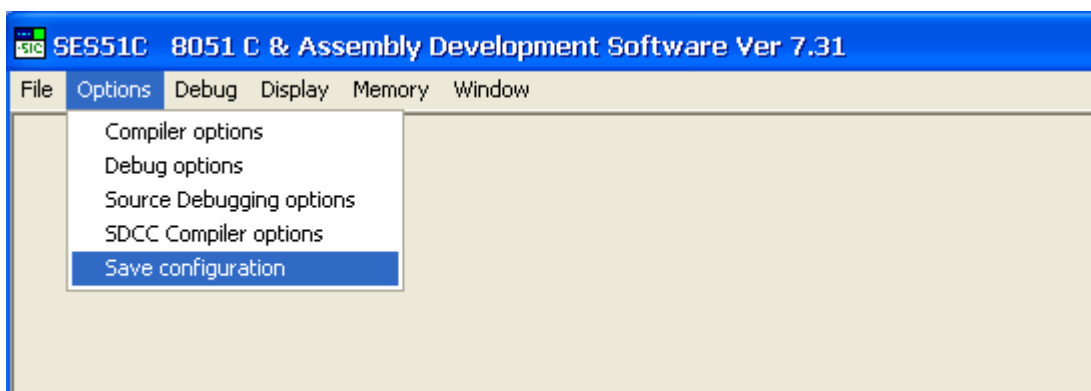
Para moverse dentro de la celda utilice las flechas

APPLIC-37 opera en 19,200 baud.

Seleccione el "Puerto Com" y pulse <ENTER> o seleccione <OK> con el ratón.

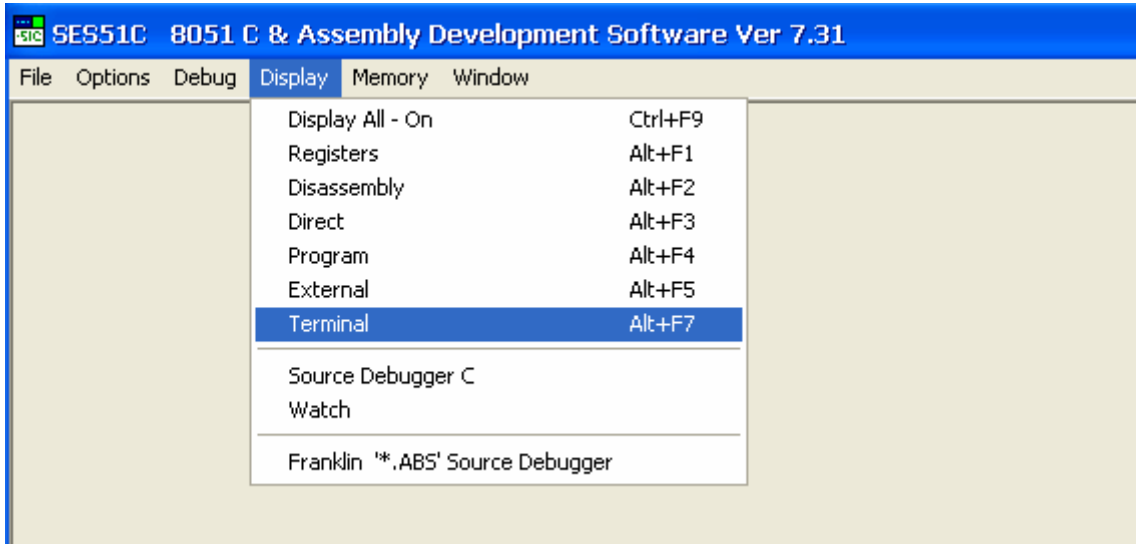
4.5.11 Seleccione OPCIONES nuevamente.

Seleccione GUARDAR CONFIGURACIONES.



Las nuevas definiciones han sido salvadas en el archivo de configuración y serán las mismas para todas las sesiones hasta que se cambien nuevamente.

4.5.12 Seleccione la sub-función TERMINAL en la función VISUALIZACION.



Esta función abre una ventana terminal de pantalla.

Usted puede agrandar esta ventana y ubicarla en cualquier lugar en la pantalla.

4.5.13 Presione el botón <RST> en el APPLIC-37. Después de un segundo, un mensaje de la abertura será enviado al terminal.

Teclee sobre el terminal de pantalla.

4.5.14 Tipear el comando siguiente:

***OUFF00,01<ENTER>**

Esto significa el número 01 en el puerto de salida en la dirección FF00.

El LED más significativo de la derecha se ENCENDERÁ.

4.5.15 Intente hacer salir diversos números en el puerto de salida.

4.6 Escribir en Lenguaje C

Cuando escribimos rutinas cortas para la práctica o programas cortos de prueba, podemos escribirlos con la ayuda del DEBUGGER.

Podemos también almacenar estas rutinas en un disquete y cargarlas del disquete.

Pero cuando estamos interesados en escribir un programa que deseamos almacenar en forma ordenada y que podamos actualizarlo fácilmente, necesitamos hacerlo en forma diferente.

El software se debe escribir en el lenguaje del microprocesador. Puede ser escrito también en un lenguaje de alto nivel, pero nos relacionaremos con esto más adelante.

El lenguaje assembly contiene, además de la instrucción mnemónica, el uso de etiquetas, símbolos, directivas, observaciones, etc. Esto hace al programa escrito en lenguaje assembly absolutamente flexible y fácil de leer y entender.

El programa en assembly se escribe usando un editor de texto. Este texto se llama **PROGRAMA FUENTE**.

Después de que el programa fuente se escriba y se almacene en el disquete, ejecutaremos el **PROGRAMA DEL ENSAMBLADOR** sobre él.

El programa del ensamblador lee el programa fuente, línea tras línea, y lo traduce a lenguaje de máquina.

El proceso de traducir requiere generalmente un ensamblador de dos pasos.

En el primer paso, el software del ensamblador forma una tabla con los valores para todas las etiquetas, símbolos y variables.

En el segundo paso, forma el programa en lenguaje de máquina, conteniendo los valores de la tabla.

Se utilizan las etiquetas como nombres de líneas que ramifican y los símbolos se utilizan como nombres para las diversas variables y parámetros.

El ensamblador calcula las direcciones y las planta en los lugares correctos.

El programa en lenguaje de máquina se llama **PROGRAMA OBJETO**.

Además de él también recibimos un **PROGRAMA LISTA** consistiendo en el programa objeto y en el programa fuente.

Lo que sigue es un programa fuente en assembly que transfiere el estado de las entradas a los LED en un bucle infinito.

```
unsigned char at 0xFF00 xdata PORT0;
unsigned char TEMP;
void main (void)
{
  while (1)
  {
    TEMP = PORT0;
    PORT0 = TEMP;
  }
}
```

4.6.1 Seleccione la función FILE y la sub-función NEW.

Se abrirá una ventana sin nombre (UNTITLED) con editor de pantalla completa.

El editor es nuestro procesador de textos para escribir nuestro programa fuente.

4.6.2 Teclee el programa siguiente:

```
unsigned char at 0xFF00 xdata PORT;
int i;
void delay ( )
{
  for (i = 0 ; i <= 32000; i++;)
}
void main (void)
{
  while (1)
  {
    PORT = 0xFF;
    delay ( );
    PORT = 0;
    delay ( );
  }
}
```


4.6.3 Para compilar un archivo a lenguaje de máquina, y crear un programa objeto, ejecutaremos el ensamblador.

El ensamblador puede trabajar solamente en un archivo de disquete, así que tenemos que guardar el archivo primero.

Seleccione la función ARCHIVO y la sub-función GUARDAR COMO.

Anote el nombre:

CBLINK.C

El texto será guardado bajo este nombre.

4.6.4 Seleccione la función COMPILAR y la sub-función COMPILAR.

4.6.5 Para descargar un programa al controlador, debe estar en formato ASCII.

Compruebe que el controlador esté conectado con el puerto serial de la PC.

Pulse <RST> en el controlador.

4.6.6 Seleccione la función DEOYRAR y la sub-función DESCARGA Y PROGRAMA.

El archivo será enviado al controlador y situado según su Dirección de Inicio. El archivo será visto en la ventana TERMINAL.

4.6.7 Teclar sobre la pantalla terminal.

4.6.8 Tipear el comando siguiente:

***GO2000<ENTER>**

Los LEDs deben centellar.

4.6.9 Para parar el programa presionar <RST>.

Appendix A

```

;APPLIC37 analog A51 sample routines
;
;DAC1 routine outputs the accumulator contents to the analog output channel 1
;DAC2 routine outputs the accumulator contents to the analog output channel 2
;
;ADC routine reads one of the analog channels (according to the accumulator
contents(
;and returns the analog channel value in the accumulator
;
-----;
;Hardware Handling
;
DAC_SYNC    EQU    P1.0
SER_CLK     EQU    P1.1
SER_DI      EQU    P1.2
SER_DO      EQU    P1.3
ADC_CS     EQU    P1.4
;SARS      EQU    P1.5
;
;
-----;
;    DAC1 ROUTINE
;    INPUT :      A = Value for DAC1
;    OUTPUT :     none
;    CALLS:
SDAC
;    USES:        A
-----;
DAC1:  PUSH  ACC
      CLR   SER_CLK;
      CLR   DAC_SYNC    ;clear SYNC
      ;
      MOV   A,#03H      ;setup DAC 1
      LCALL SDAC        ;send to the DAC
      ;

      POP   ACC
      LCALL SDAC        ;send to the DAC
      ;
      SETB  DAC_SYNC    ;STROBE DATA OUT
      RET;

;
-----;
;    DAC2 ROUTINE
;    INPUT :      A = Value for DAC2
;    OUTPUT :     none
;    CALLS:       SDAC
;    USES:        A
-----;
DAC2:  PUSH  ACC
      CLR   SER_CLK;
      CLR   DAC_SYNC    ;clear SYNC
      ;
      MOV   A,#07H      ;setup DAC 2

```

```

        LCALL SDAC          ;send to the DAC
        ;
        POP ACC
        LCALL SDAC          ;send to the DAC
        ;
        SETB DAC_SYNC      ;STROBE DATA OUT
        RET;
;
-----;
; SDAC ROUTINE
;SEND 8 bits of the ACC to the DAC
-----;

SDAC:
PUSH B
MOV B,#8
DACL: RLC A
MOV SER_DI,C
SETB SER_CLK
CLR SER_CLK
DJNZ B,DACL
POP B
RET
;
;
-----;
; ADC ROUTINE
; INPUT: A = MUX ADDRESS (0-7(
; OUTPUT: A = ACTUAL ADC VALUE
; NOTE: Stop timer (if is on) while using this routine
-----;
ADC: PUSH B ;we use B as a counter
SETB SER_DI ;for START bit
SETB ADC_CS
CLR SER_CLK
;
ANL A,#7 ;clear ACC.3-ACC.7 (leave 0-7(
PUSH DPH
PUSH DPL
MOV DPTR,#MUX_TABLE
MOVC A,@A+DPTR
POP DPL
POP DPH
SETB ACC.7 ;for single mode (not differential(
;
CLR ADC_CS
SETB SER_CLK ;latch and shift START BIT to the ADC
NOP
CLR SER_CLK
;
MOV B,#4 ;for sending 4 bits of mode and address
MUXL: RLC A
MOV SER_DI,C
SETB SER_CLK ;latch and shift one BIT to the ADC
NOP
CLR SER_CLK
DJNZ B,MUXL
;
MOV A,#0
MOV B,#8 ;for reading 8 bits of the analog channel value

```

```
;
ADCL:
    SETB  SER_CLK
    NOP
    CLR   SER_CLK
    MOV   C,SER_DO
    RLC   A
    DJNZ  B,ADCL
    ;
    SETB  ADC_CS
    POP   B
    RET                               ;with analog channel value in A
    ;
MUX_TABLE:
    DB    0           ;CAHNEL 0
    DB    40H        ;CAHNEL 1
    DB    10H        ;CAHNEL 2
    DB    50H        ;CAHNEL 3
    DB    20H        ;CAHNEL 4
    DB    60H        ;CAHNEL 5
    DB    30H        ;CAHNEL 6
    DB    70H        ;CAHNEL 7

    END
```

```

//APPLIC37 analog C51 sample routines
//
//DAC1 routine outputs the DAC_VALUE contents to the DAC channel 1
//DAC2 routine outputs the DAC_VALUE contents to the DAC channel 2
//ADC routine reads one of the analog channels (according to the ADC_CHANNEL
contents and returns the value in the ADC_VALUE

// Hardware Handling
sbit at 0x90 dac_sync ; //p1.0
sbit at 0x91 ser_clk ; //p1.1
sbit at 0x92 ser_di ; //p1.2
sbit at 0x93 ser_d0 ; //p1.3
sbit at 0x94 adc_cs ; //p1.4

unsigned char DAC_VALUE;
unsigned char ADC_CHANNEL;
unsigned char ADC_VALUE;
char B;

sdac(char A)
{
char J;
for (J = 1; J < 9; J++)
{
if ((A & 0x80) == 0x80)
{
ser_di = 1;
}
else
{
ser_di = 0;
}
ser_clk = 1;
ser_clk = 0;
A = A<<1;
}
}

dac1()
{
ser_clk = 0;
dac_sync = 0; //clear SYNC
sdac(3);
sdac(DAC_VALUE);
dac_sync = 1;
}

dac2()
{
ser_clk = 0;
dac_sync = 0; //clear SYNC
sdac(7);
sdac(DAC_VALUE);
dac_sync = 1;
}

adc()
{
char CH[8]= {0x0,0xc0,0x90,0xd0,0xa0,0xe0,0xb0,0xf0};
char A,J;
ser_di = 1; //for START bit
adc_cs = 1;
ser_clk = 0;

```

```
A = CH[ADC_CHANNEL];
adc_cs = 0;
ser_clk = 1;           //latch and shift START BIT to the ADC
ser_clk = 0;
for (J = 1; J < 5; J++)
{
  if ((A & 0x80) == 0x80)
  {
    ser_di = 1;
  }
  else
  {
    ser_di = 0;
  }
  ser_clk = 1;
  ser_clk = 0;
  A = A<<1;
}
A = 0;
for (J = 1; J < 9; J++)
{
  ser_clk = 1;
  ser_clk = 0;
  if (ser_d0 = 1)
  {
    A = A | 1;
  }
  else
  {
    A = A & 0xfe;
  }
  A = A<<1;
}
adc_cs = 1;
ADC_VALUE = A;
}

main()
{
}
}
```

Appendix B

TPS-3732 Mobile Robot

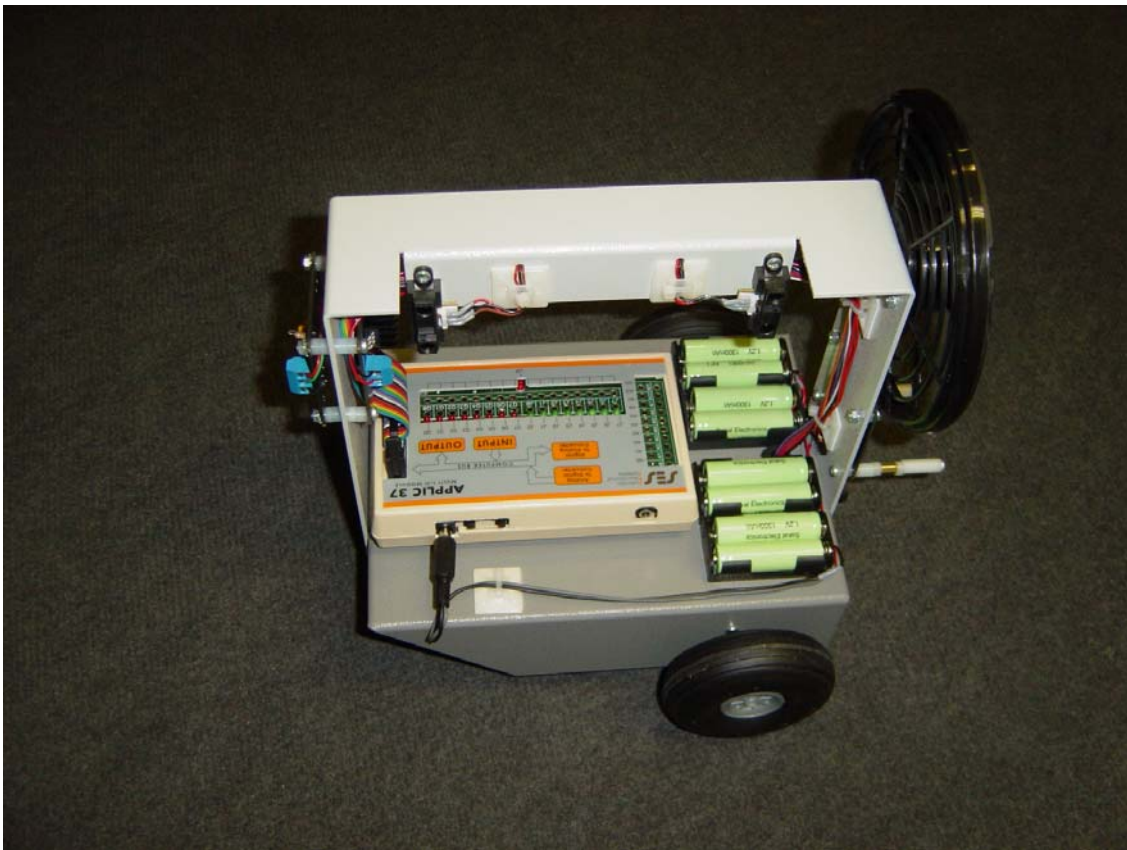
The TPS-3732 mobile robot is an AGV (Automatic Guided Vehicle) robot. It includes two wheels with motors and a support wheel.

It also includes a range sensor on its front and two range sensors on its right side.

A white line sensor is located on its bottom.

A sound sensor is located on its back.

A light sensor is located on the front side of the robot.



The disk includes a demo program called robot.a51. This program operates the robot as follows:

- 1) The robot waits for a buzzer sound near the sound sensor.
- 2) After the buzzer beeps, the robot goes along a right wall. If it does not sense a wall on the right, it goes straight.
- 3) When the robot is against a wall, it turns to the left and continues along the wall on the right.
- 4) When the robot senses a flame light in front of it, it stops, beeps and turn the fan ON until the flame is OFF.

Compile the robot37.a51 file and download it to the APPLIC-37 according to chapter 2.

Move the rear switch to the right (towards the power socket) and press RST.

Check the robot behavior according to the above description.