

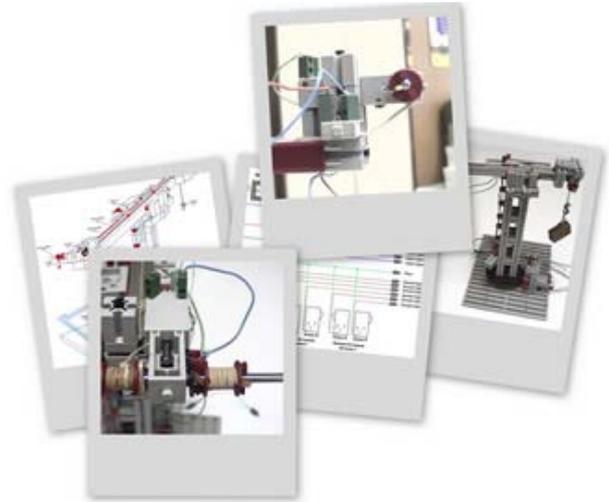
GRÚA

Introducción

Este dispositivo simula el funcionamiento de una grúa, que se caracteriza por tener tres tipos de movimientos diferentes:

1. Gira la base de la grúa de izquierda a derecha.
2. El rail superior de la grúa se mueve de delante a atrás.
3. La cuerda que tiene el gancho de la grúa se mueve de arriba abajo.

Por medio de la controladora realizaremos los movimientos típicos de una grúa (subir, bajar, desplazamientos laterales,...).



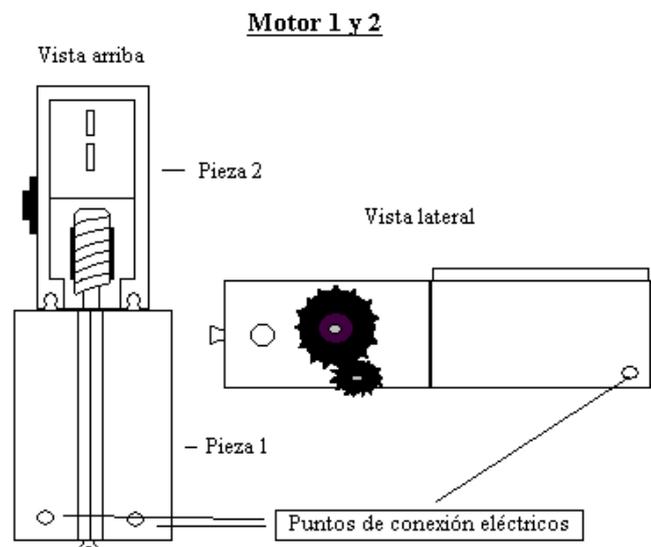
Construcción

El dispositivo consta de 3 motores que se utilizan para poder ejecutar las 3 diferentes clases de movimientos de la grúa.

Para controlar los movimientos de los distintos elementos de la grúa se pondrán unos sensores de contacto que marcarán los límites para los movimientos a realizar por la máquina.

Toda la estructura se apoya sobre una Plataforma (1) de 18´6 x 25´8 x 0´6 cm. Los elementos que la componen son:

- **4 columnas** (2): son los pilares de la grúa, cada columna esta formada por 8 bloques de plástico de 3 x 1´5 cm, y por 2 bloques de plástico de 1´5 x 1´5 cm, un bloque de estos últimos será el soporte de la rueda, sobre la cual se apoyará el resto de columnas.
- **6 sensores de contacto** (3): los cuales actuarán como toques para poder poner fin a cada uno de los distintos movimientos realizados por la máquina.
- **Motor 1** (4): este es el motor necesario para que la grúa efectúe movimientos rotativos. El motor está compuesto por 2 piezas: una es el motor propiamente dicho y la otra está compuesta por diferentes ruedas que son las que van a permitir mover los elementos conectados al motor.

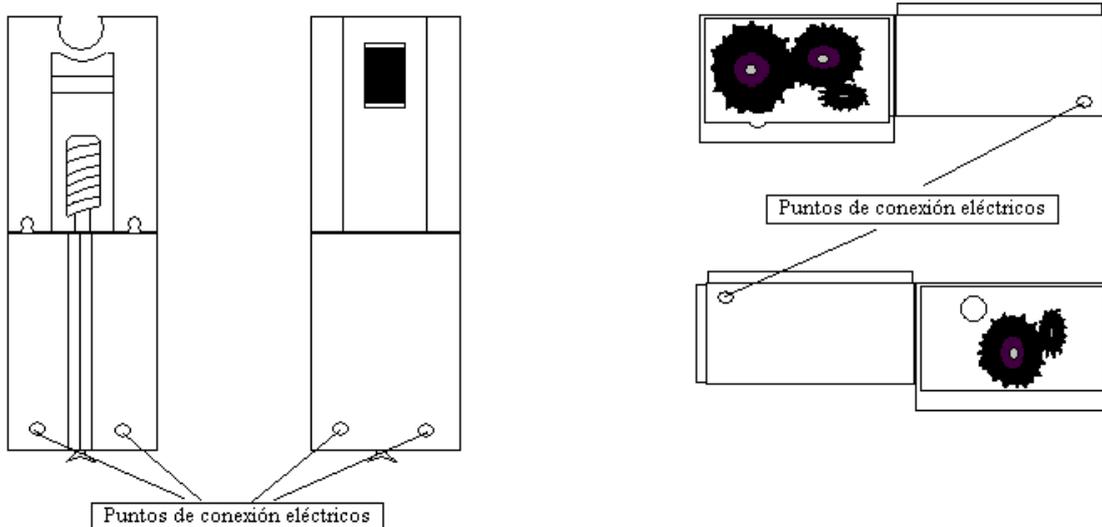


Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

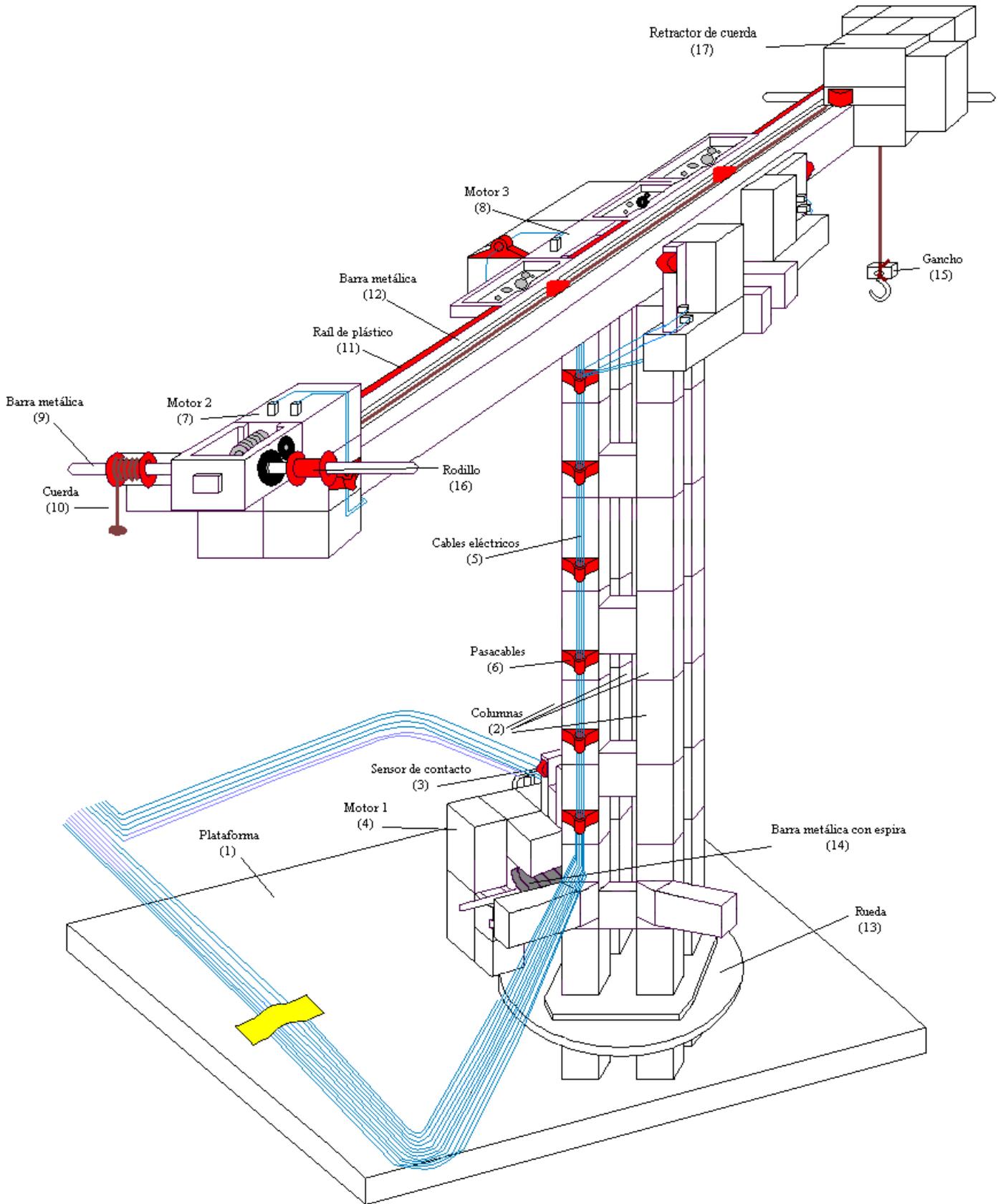
- **6 cables eléctricos** (5): los cuales nos permitirán llevar la corriente eléctrica hasta los 3 motores de los que consta la grúa, con lo que cada motor utilizará un par de cables.
- **8 pasacables** (6): se utilizarán como guías para los cables eléctricos, de tal manera que los cables queden bien fijados a los bloques de plástico que forman las columnas.
- **Motor 2** (7): este es el motor necesario para que la grúa estire o recoja la cuerda con el gancho. El motor está compuesto por 2 piezas: una es el motor propiamente dicho y la otra está compuesta por diferentes ruedas que son las que van a permitir mover los elementos conectados al motor.
- **Motor 3** (8): este es el motor necesario para que la grúa se mueva a través de la barra metálica. El motor está compuesto por 2 piezas: una es el motor propiamente dicho y la otra se integra por diferentes ruedas que son las que van a permitir mover los elementos conectados al motor. Este motor tendrá situado a cada extremo 2 piezas de plástico preparadas para encajar los railes.

Motor 3

Vista de diferentes perspectivas



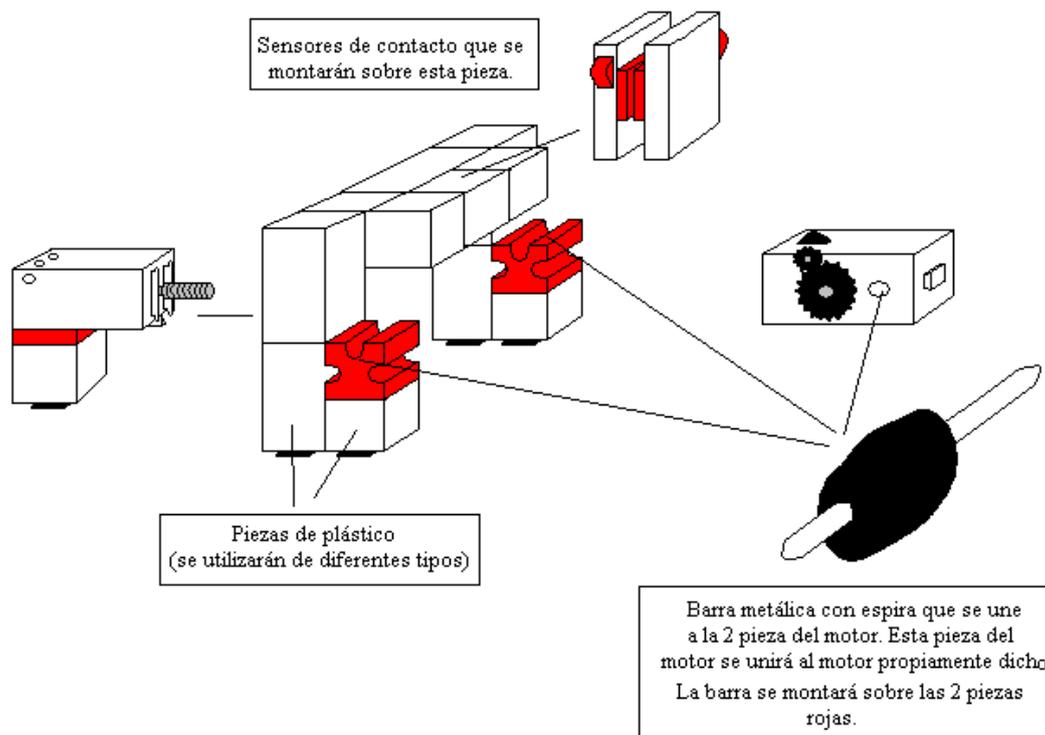
- **2 barras metálicas** (9): las barras tendrán unas dimensiones de 9 x 0´3 cm. Servirán para enrollar la cuerda que sujeta el gancho.
- **Cuerda** (10): es el elemento principal de la grúa. Se encarga de sujetar el gancho. El mejor material a utilizar es hilo de bramante.
- **Rail de plástico** (11): es el dispositivo a través del cual se mueve la barra metálica hacia delante y hacia atrás. Éste estará puesto sobre la barra metálica y ocupará prácticamente la totalidad de ésta, dejando 2 pequeños espacios en los extremos de la barra.
- **Barra metálica** (12): esta barra tiene unas dimensiones de 27 x 1´5 cm. En la barra se colocará el rail de plástico.
- **Rueda** (13): es el elemento imprescindible y necesario para que la grúa pueda efectuar movimientos rotativos. Ésta estará apoyada sobre 4 bloques de plástico de 1´5 x 1´5 cm, también estará en contacto con la espira de la barra metálica que es lo que va a permitir girar la rueda.
- **Barra metálica con espira** (14): la parte fundamental de la barra es la espira, que es la que va a permitir mover la rueda que hace girar la grúa. Esta barra se apoyará sobre 2 bloques preparados para el encaje de la barra.
- **Gancho** (15): es el que nos permite elevar objetos y estará situado en un extremo de la cuerda.
- **Rodillo** (16): sobre el rodillo quedará enrollada la cuerda al recogerse ésta.
- **Retractor de cuerda** (17): este elemento está en la barra metálica en el extremo contrario a donde está situado el motor 2.



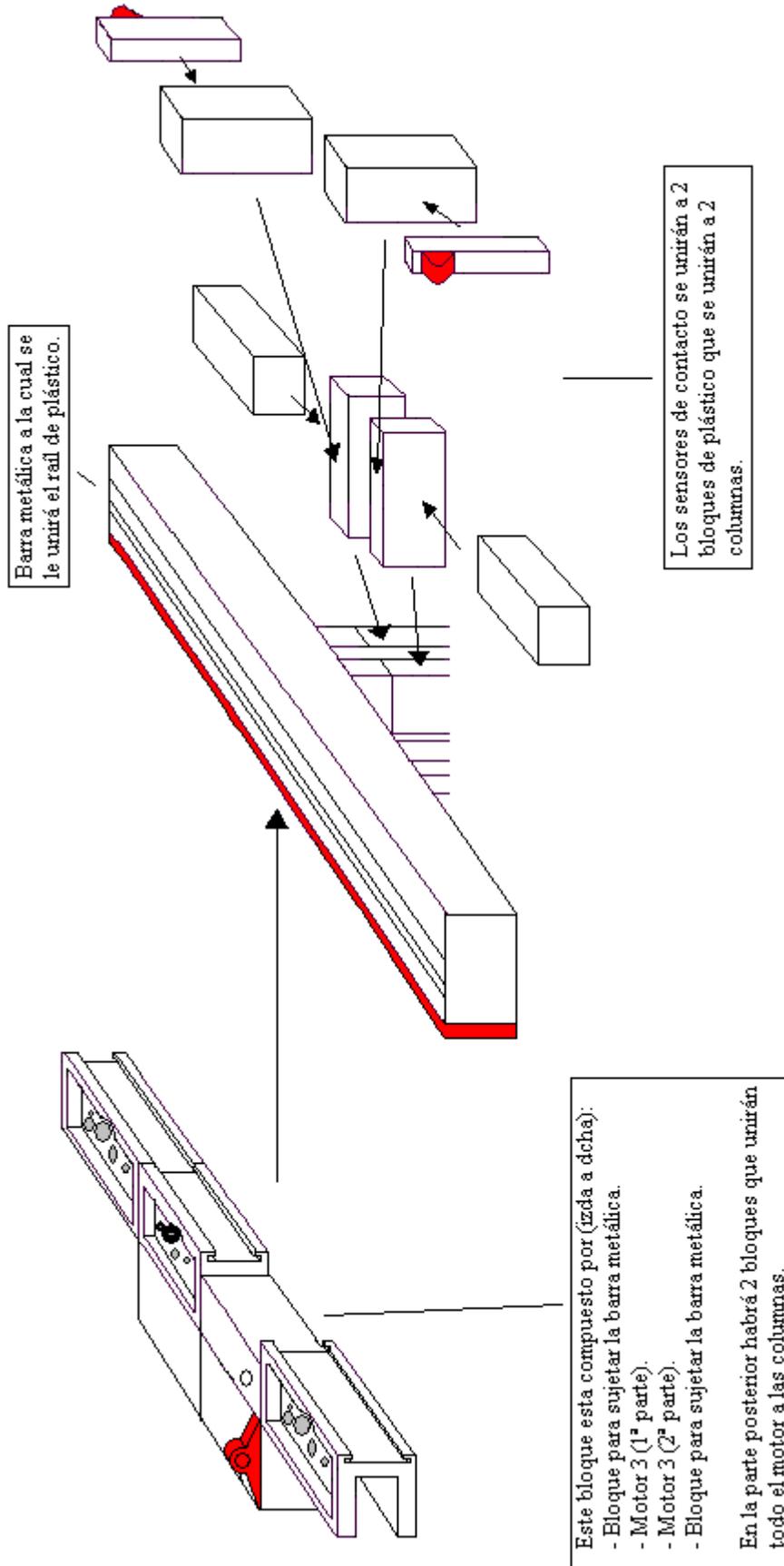
Para que resulte más sencillo el montaje de la grúa lo desglosaremos en diferentes pasos. En este proceso hemos utilizado piezas de Fischer, aunque se podría utilizar cualquier otro material.

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

1. Se montará la **rueda** (13) sobre 4 piezas de plástico pequeñas (tamaño 1´5 x 1´5), éstas a su vez se montarán sobre la **plataforma** (1), la cual será la base que soportará la grúa.
2. Sobre la rueda se montarán las 4 columnas formadas por piezas de plástico. En uno de los laterales de una de las 4 columnas se colocarán los pasacables (6) los cuales tienen una pequeña apertura que se tapaná con celofán o cinta aislante.
3. Al lado de la rueda se montará el **motor 1** (4), al cual se acoplará la **barra metálica con espira** (14) que entrará en contacto con la **rueda** (13) haciéndola girar. Los dos primeros elementos se encuentran sobre una plataforma formada por piezas de plástico. En la parte superior de la plataforma se colocarán **2 sensores de contacto** (3) apuntando siempre los pulsadores de cada sensor hacia fuera de la plataforma.



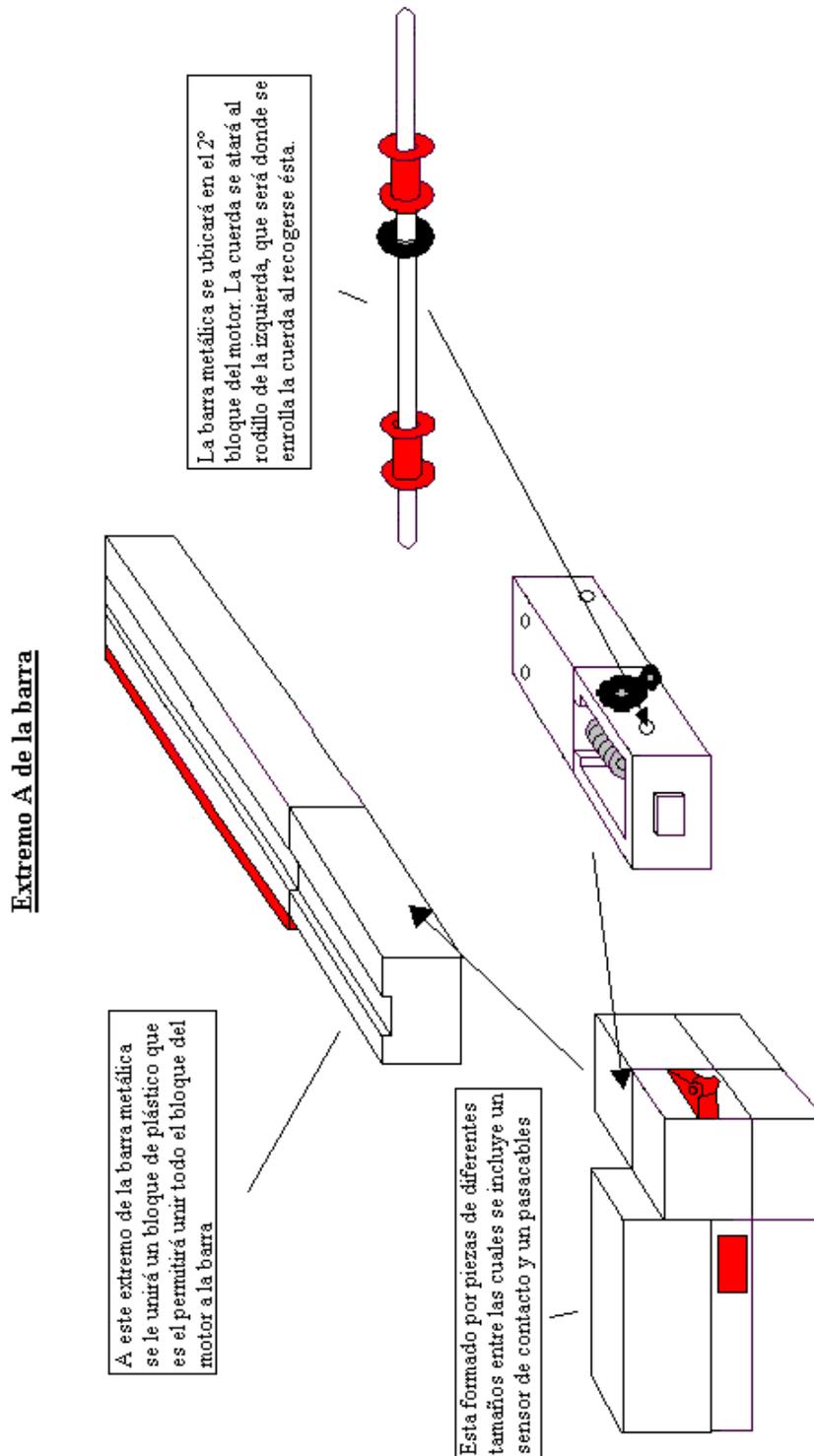
4. Se montará el **motor 3** (8) al que se acoplará la **barra metálica** (12). A lo largo de ésta se fijará un **raíl de plástico** (11) que está compuesto por 3 piezas.



Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

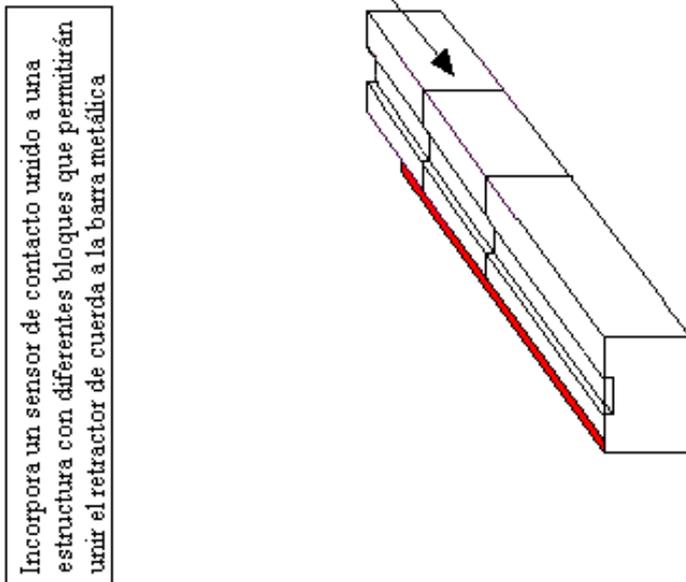
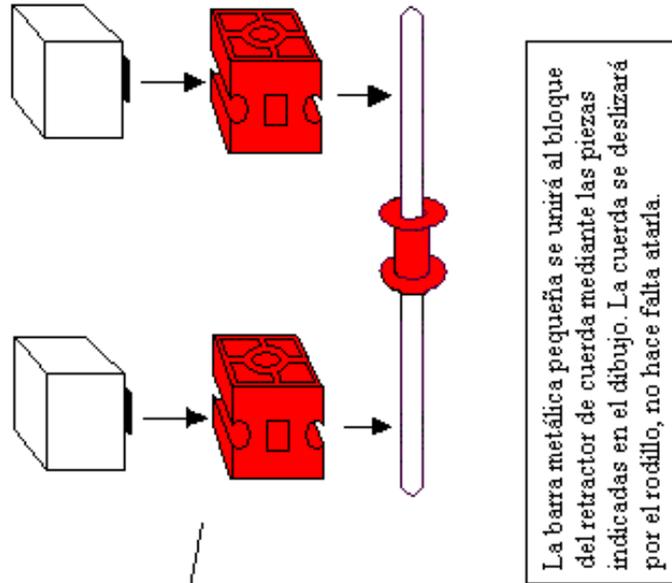
Todo esto a su vez irá montado en la parte superior de las columnas, para lo cual se utilizarán varias piezas de plástico. En cada extremo de la barra metálica se adjuntarán 1 (Extremo A) y 2 (Extremo B) piezas de plástico respectivamente.

5. En cada extremo de la barra metálica se montarán el **motor 2** (7) y el **retractor de cuerda** (17) respectivamente.
 1. **Extremo A:** este es el extremo que se encuentra próximo a la columna donde se están los pasacables. Aquí se montará el motor 2 al que se le incorporará una barra metálica con 2 rodillos en cada extremo, uno de los cuales será donde se enrolle la cuerda al ser recogida. Se utilizarán piezas de plástico para fijar el motor a la barra metálica. Se le acoplará un sensor de contacto. (ver imagen: Extremo A de la barra).



2. **Extremo B:** en este extremo se montará el retractor de cuerda que recogerá la cuerda de la grúa. Se le acoplará un sensor de contacto. (ver imagen: Extremo B de la barra).

Extremo B de la barra



Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

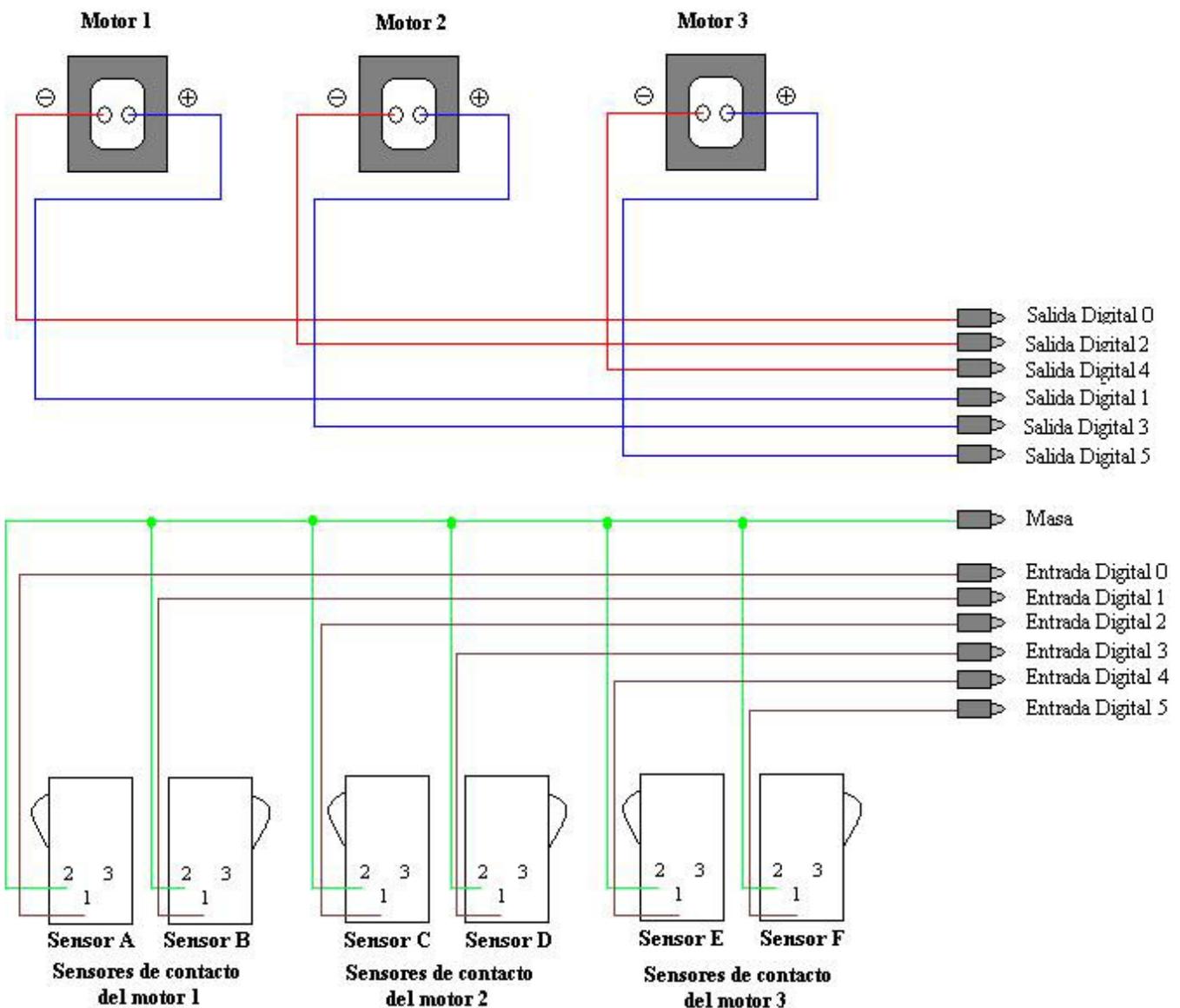
- Por último se colocará la **cuerda** (10) por el hueco central de la barra metálica, llegando un extremo de ésta al rodillo del extremo A, donde se la atará. El otro extremo de la cuerda se dejará deslizar por el rodillo del extremo B, en este extremo de la cuerda se atará un **gancho** (15). A la cuerda se le pegarán 2 pequeños trozos de plástico que serán los que pulsen los sensores de contacto. La distancia entre los nudos será de 10⁻⁵ cm.
- Se conectará a cada motor un par de hilos **eléctricos** (5). También se conectarán un par de hilos eléctricos a cada sensor de contacto. Todos los hilos se pasarán por los pasacables.

Esquemas eléctricos

Para poder alimentar los motores de la grúa son necesarios 3 pares de hilos eléctricos para cada uno de ellos que se conectarán a las salidas digitales. Los 3 motores harán que la grúa realice 3 clases diferentes de movimientos y para cada movimiento se van a utilizar 2 sensores. Uno de los conectores de los sensores se conectará a una de las entradas digitales y el otro conector de los sensores se conectará a la masa de la controladora.

En los sensores de contacto se conectará un hilo eléctrico a las entradas digitales y el otro hilo eléctrico a masa. En total se utilizarán 12 hilos, 2 por cada sensor de contacto.

La forma en la que se conectan los motores y los sensores de la grúa a la controladora se muestra en el siguiente esquema:



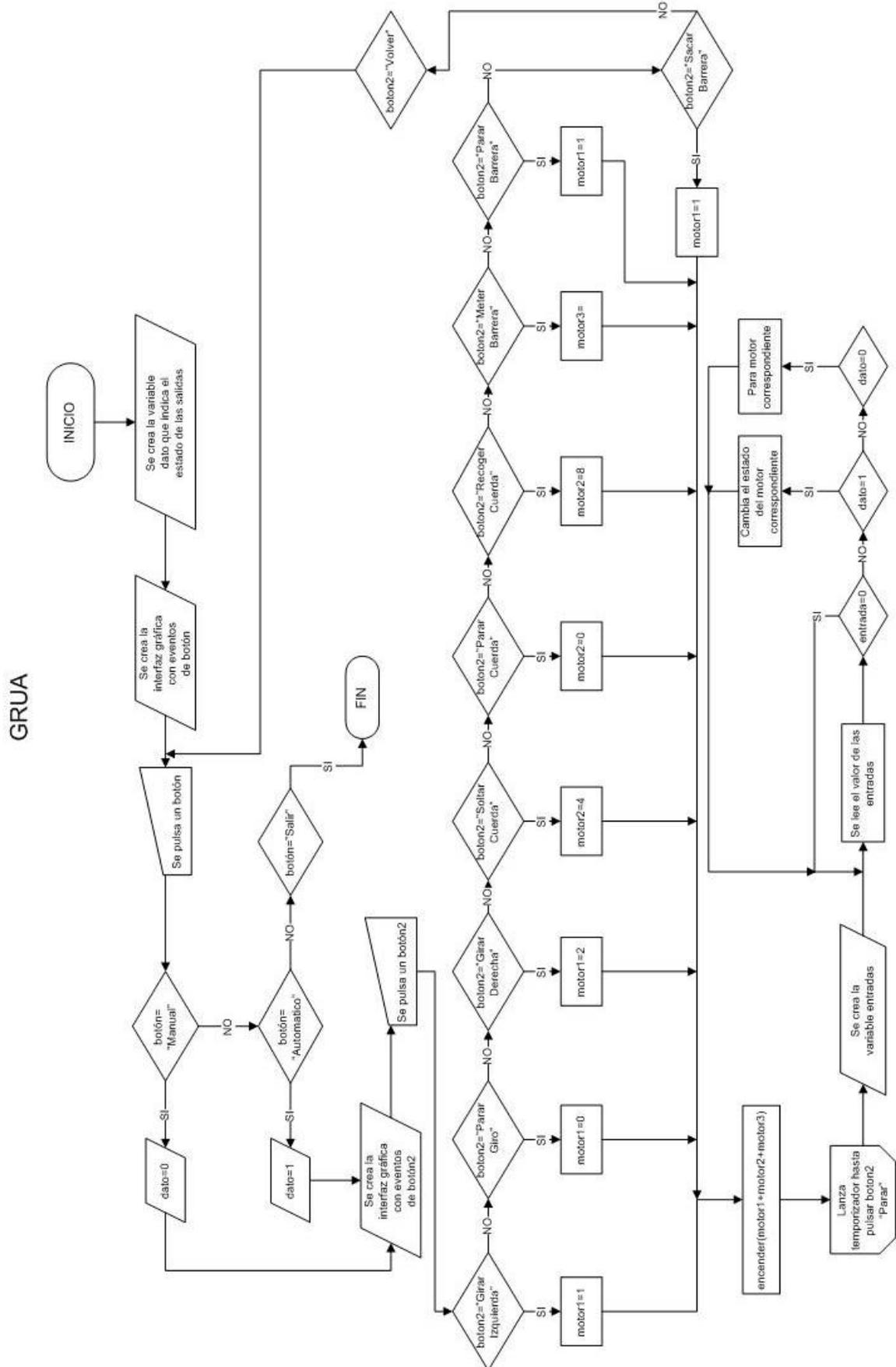
Conexiones

Las conexiones que se establecerán entre la grúa y la tarjeta controladora CNICE se muestran en la siguiente tabla:

Grúa	Controladora	
	Salida digital	Entrada digital
Motor 1 (+)	2	
Motor 1 (-)	1	
Motor 2 (+)	4	
Motor 2 (-)	3	
Motor 3 (+)	6	
Motor 3 (-)	5	
Sensor A		1
Sensor B		2
Sensor C		3
Sensor D		4
Sensor E		5
Sensor F		6
Masa		Masa (GND)

PROGRAMACIÓN

Diagrama de flujo



MSWLOGO

La programación de este caso se estructura en los siguientes pasos:

1. Se crea un procedimiento para crear la ventana gráfica principal del programa. Para ello se utiliza la función **creaventana**. Dentro de la ventana se crean los botones con la función **creaboton**. Dentro de cada botón se establecerán entre los corchetes las funciones que se han de ejecutar una vez presionado el botón. Uno de los botones creados será el que se utiliza para salir de la aplicación, para lo cual se utiliza la orden **adios**. También se añade la imagen de la grúa con la función **cargadib**.

```
creaventana "Principal "grua [Grua] 100 42 150 100 []
creaboton "grua "Manual "Manual 18 20 50 20 [proc_manual]
creaboton "grua "Automatico "Automatico 80 20 50 20 [proc_automatico]
creaboton "grua "Salir "Salir 50 50 50 20 [proc_salir adios]
cargadib ("grua.bmp)
```

En este procedimiento se crea y se inicializa a 0 una variable que llamaremos **bucle** que nos servirá para poder leer las entradas digitales de manera continuada.

2. Se crea un procedimiento para crear la ventana grafica para cada uno de los tipos de funcionamiento de la grúa, manual o automático. Estas dos ventanas van a ser de igual aspecto, la única diferencia que existe es la actuación de la grúa. Se crean los botones, uno para cada movimiento de la grúa, donde se establecerán entre los corchetes las funciones que se han de ejecutar una vez presionados.

```
creaventana " "Manual [GRUA-Manual] 100 42 275 200 []
creaboton "Manual "Girar_izquierda "Girar_izquierda 20 20 75 20
[proc_girarizquierda control1]
creaboton "Manual "Parar_Giro "Parar_Giro 100 20 75 20 [M1 "P
cargadib("grua.bmp) control1]
creaboton "Manual "Girar_derecha "Girar_derecha 180 20 75 20
[proc_girarderecha control1]
creaboton "Manual "Soltar_cuerda "Soltar_cuerda 20 60 75 20
[proc_soltarcuerda control1]
creaboton "Manual "Parar_Cuerda "Parar_Cuerda 100 60 75 20 [M2 "P
cargadib("grua.bmp) control1]
creaboton "Manual "Rocoger_cuerda "Recoger_cuerda 180 60 75 20
[proc_recogercuerda control1]
creaboton "Manual "Rail_Atras "Rail_Atras 20 100 75 20 [proc_railatras
control1]
creaboton "Manual "Parar_Rail "Parar_Rail 100 100 75 20 [M3 "P
cargadib("grua.bmp) control1]
creaboton "Manual "Rail_Adelante "Rail_Adelante 180 100 75 20
[proc_railadelante control1]
creaboton "Manual "Volver "Volver 100 140 75 20 [proc_parar graficos]
```

3. Se crean procedimientos para activar las salidas digitales que activan los 3 motores en uno u otro sentido, para desactivar los motores y salir de la aplicación creada. En estos procedimientos se va a utilizar la función **M** seguida del número que indique el par de salidas digitales a activar, y tras esto se pondrán unas comillas con la letra D o I que indicará que el motor gire a la derecha (activa la salida 1 y desactiva la salida 0) o que el motor gire a izquierda (activa la salida 0 y desactiva la salida 1), con la letra P se le indica al motor que se detenga (desactiva las 2 salidas digitales).

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

```
para proc_girarizquierda
cargadib ("gruagiro1.bmp)
M1 "I
Fin
```

4. Para poder leer las entradas digitales se necesita crear un bucle para que esté continuamente leyendo el valor de las entradas digitales. Para crear un bucle utilizamos la orden mientras y asignamos a una variable un valor, el cual se cambiará al dar al botón salir para poder salir del bucle y poder terminar así de la ejecución del programa.

Para leer las entradas digitales se utiliza la función **ve?** la cual nos devuelve el valor que hay en ese momento en las entradas, este valor lo guardaremos en una variable a la que llamamos **entrada**.

Luego, según el valor que tengan las entradas se activará un motor u otro en un sentido o en el otro, para lo cual se utilizarán condicionales.

A continuación se muestra el procedimiento

```
para control
mientras [:bucle=0] [
haz "entradas ve?

;Ahora se comprueba si se ha pulsado algún sensor#####
si (:entradas=1) [M1 "I]
si (:entradas=2) [M1 "D]
si (:entradas=4) [M2 "D]
si (:entradas=8) [M2 "I]
si (:entradas=16) [M3 "I]
.....
]
```

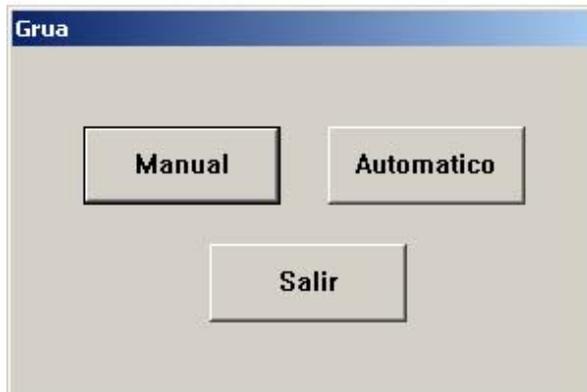
5. Se llama al procedimiento que crea la ventana gráfica fuera de cualquier procedimiento para que se cargue la aplicación gráfica nada más cargar el fichero de logo.

Descargar el archivo programado en **MSWLogo**. Descomprímalo y guárdelo en un directorio aparte. Contiene el fichero de código en MSWLogo (grua.lgo).

Ejecute el compilador **MSWLogo versión 6.5a** en castellano.

Vaya al menú del programa, Archivo/Abrir y seleccione el fichero **grua.lgo** que se descargó previamente.

Se visualizará la siguiente pantalla:



En ella se pulsa el botón correspondiente al funcionamiento que queremos. Aparecerá la siguiente pantalla, igual para los dos modos de funcionamiento:



En ella bastará con pulsar en los botones para que la grúa realice el movimiento deseado:

PROGRAMACIÓN C

La programación de este caso se estructura en los siguientes pasos:

- 1.- Se crea un nuevo proyecto.
- 2.- Se añaden al proyecto los archivos io.h, io.cpp, Primitivas_CNICE.CPP y Primitivas_CNICE.HPP
- 3.- Se crea el archivo main.c donde se incluirán las funciones necesarias para crear las ventanas.

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

4.- Dentro del archivo main.c creando anteriormente hay que crear un hilo para que compruebe los sensores en todo momento y se añade la declaración de las funciones de la biblioteca io.dd de la siguiente manera:

```
#include "io.h"
```

También se añade la declaración a las funciones de la biblioteca SDL.dll de la siguiente manera:

```
#include <SDL.h>
```

5.- En nuestro archivo main.c se define una función que permite activar o desactivar las entradas digitales de la controladora y otra que permite leer el estado de las entradas digitales. Las funciones son las siguientes:

```
void encender(int led)  
{  
    LoadIODLL(); //Carga la libreria io.dll  
    PortOut(0x37A,0x7);  
    PortOut(0x378,led);  
}  
int leedigital()  
{  
    int bajo=0;  
    int alto=0;  
    int d=0;  
  
    LoadIODLL();  
    PortOut(0x37A, 0x3);  
    bajo = PortIn(0x379);  
    bajo = (bajo & 0x78) / 8;  
    bajo = ~bajo;  
    bajo = bajo & 15;  
  
    PortOut(0x37A, 0x1);  
    alto = PortIn(0x379);  
    alto = (alto & 0x78) / 8;  
    alto = ~alto;  
    alto = alto & 15;  
    d = (alto * 16) | bajo;  
  
    return d;  
}
```

6.- Se crean dos botones en nuestra ventana, uno para la ejecución "Manual" y otro para la ejecución "Automática", y en cada uno de ellos se crea la ventana para manejar la barrera. Se crean 10 botones: Girar Barrera Izquierda, Parar Barrera, Girar Barrera Derecha, Recoger Cuerda, Parar Cuerda, Soltar Cuerda, Sacar Barrera, Parar Barrera, Meter Barrera. En cada uno de los botones se incluye la llamada a la función anterior para cambiar el estado de las salidas y así realizar el movimiento correspondiente de la siguiente manera:

```
switch(LOWORD(wParam))  
{  
    case 1: // botón Girar Izquierda  
        motor1=1;  
        dato=0;  
        izquierda = SDL_LoadBMP("gruagiro1.bmp");  
        screen = SDL_SetVideoMode( 300, 300, 0, SDL_NOFRAME );  
        if( screen == NULL ) {
```

```

        printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );
        SDL_Quit();
    }
    rect = (SDL_Rect) {0, 0, 200, 200};
    SDL_BlitSurface(izquierda, NULL,screen,&rect);
    SDL_Flip(screen);
    encender(motor1 + motor2 + motor3);
    break;
case 2: // botón Parar
    motor1 = 0;
    dato=0;
    encender(motor1 + motor2 + motor3);
    break;
case 3: // botón Girar Derecha
    motor1=2;
    dato=0;
    derecha = SDL_LoadBMP("gruagiro2.bmp");
    screen = SDL_SetVideoMode( 300, 300, 0, SDL_NOFRAME );
    if( screen == NULL ) {
        printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );
        SDL_Quit();
    }
    rect = (SDL_Rect) {0, 0, 200, 200};
    SDL_BlitSurface(derecha, NULL,screen,&rect);
    SDL_Flip(screen);
    encender(motor1 + motor2 + motor3);
    break;
case 4: // botón Soltar Cuerda
    motor2=4;
    dato=0;
    bajar = SDL_LoadBMP("gruacuerdaabajo.bmp");
    screen = SDL_SetVideoMode( 300, 300, 0, SDL_NOFRAME );
    if( screen == NULL ) {
        printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );
        SDL_Quit();
    }
    rect = (SDL_Rect) {0, 0, 200, 200};
    SDL_BlitSurface(bajar, NULL,screen,&rect);
    SDL_Flip(screen);
    encender(motor1 + motor2 + motor3);
    break;
case 5: // botón Parar
    motor2 = 0;
    dato=0;
    encender(motor1 + motor2 + motor3);
    break;
case 6: // botón Recoger Cuerda
    motor2=8;
    dato=0;
    subir = SDL_LoadBMP("gruacuerdaarriba.bmp");
    screen = SDL_SetVideoMode( 300, 300, 0, SDL_NOFRAME );
    if( screen == NULL ) {
        printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );
        SDL_Quit();
    }
    rect = (SDL_Rect) {0, 0, 200, 200};
    SDL_BlitSurface(subir, NULL,screen,&rect);
    SDL_Flip(screen);
    encender(motor1 + motor2 + motor3);
    break;
case 7: // botón Rail atras
    motor3=16;
    dato=0;
    atras = SDL_LoadBMP("gruabarradentro.bmp");
    screen = SDL_SetVideoMode( 300, 300, 0, SDL_NOFRAME );
    if( screen == NULL ) {
        printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );
        SDL_Quit();
    }

```

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

```
    }
    rect = (SDL_Rect) {0, 0, 200, 200};
    SDL_BlitSurface(atras, NULL,screen,&rect);
    SDL_Flip(screen);
    encender(motor1 + motor2 + motor3);
    break;

case 8: // botón Parar
    motor3 = 0;
    dato=0;
    encender(motor1 + motor2 + motor3);
    break;
case 9: // botón Rail adelante
    motor3=32;
    dato=0;
    adelante = SDL_LoadBMP("gruabarrafuera.bmp");
    screen = SDL_SetVideoMode( 300, 300, 0, SDL_NOFRAME );
    if( screen == NULL ) {
        printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );
        SDL_Quit();
    }
    rect = (SDL_Rect) {0, 0, 200, 200};
    SDL_BlitSurface(adelante, NULL,screen,&rect);
    SDL_Flip(screen);
    encender(motor1 + motor2 + motor3);
    break;
case 10: // botón Salir
    dato=0;
    encender(0);
    SendMessage(hwnd, WM_CLOSE, 0, 0);
    break;
default:
    break;
}
```

En este caso es la ventana de la ejecución "Manual". Se puede ver que se ha incluido lo siguiente:

```
    dato=0;
```

Esto determinara si la ejecución es la Automática o la Manual a la hora de hacer la comprobación de las entradas de los sensores que veremos a continuación.

Explicación de las funciones de la librería SDL.

Cada vez que se quiera mostrar una imagen se indicara en nuestro proyecto lo siguiente:

```
    foto = SDL_LoadBMP("Imagen.bmp");
    screen = SDL_SetVideoMode(200, 308, 0, SDL_NOFRAME );
    if( screen == NULL ) {
        printf( "Error al entrar a modo grafico: %s\n", SDL_GetError() );
        SDL_Quit();
        return -1;
    }
    rect.x=0;
    rect.y=0;
    rect.w=primera1->w;
    rect.h=primera1->h;
    destino.x=0;
    destino.y=0;
    SDL_BlitSurface(primera1, &rect, screen, &destino);
    SDL_Flip(screen);
```

donde **foto** y **screen** son del tipo `SDL_Surface` y **rect** y **destino** es del tipo `SDL_Rect`.

- `SDL_LoadBMP`: carga la imagen .bmp que queramos
- `SDL_SetVideoMode` (int width, int height, int bpp, Uint32 flags): configure un modo de video con una anchura (width), una altura (height) y unos bits-por-pixeles. El parámetro

flags indica el tipo de ventana que se quiere. En nuestro caso una ventana sin titulo no borde.

- SDL_BlitSurface(imagen, &rect, screen, &destino): pega desde la imagen, la porción seleccionada por rect sobre la superficie screen en el destino indicado por destino.
- SDL_Flip(screen): muestra la imagen que se ha seleccionado.

7.- Se crea una función que realice la comprobación de los sensores. Esta función es la que ejecuta el hilo que hemos creado al principio. La función es la siguiente:

```
DWORD WINAPI Comprobar_Sensor(LPVOID parametro)
{
    int E;
    for(;;){
        Sleep(500);
        E=leedigital();
        fflush(stdout);
        switch (E)
        {
            case 1:
                if(dato==1){
                    motor1=2;
                    encender(motor1+motor2+motor3);
                    fflush(stdin);
                }
                else if(dato==0){
                    motor1=0;
                    encender(motor1+motor2+motor3);
                    fflush(stdin);
                }
                break;
            case 2:
                if(dato==1){
                    motor1=1;
                    encender(motor1+motor2+motor3);
                    fflush(stdin);
                }
                else if(dato==0){
                    motor1=0;
                    encender(motor1+motor2+motor3);
                    fflush(stdin);
                }
                break;
            case 4:
                if(dato==1){
                    motor2=8;
                    encender(motor1+motor2+motor3);
                    fflush(stdin);
                } else if(dato==0){
                    motor2=0;
                    encender(motor1+motor2+motor3);
                    fflush(stdin);
                }
                break;
        }
    }
}
```

Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

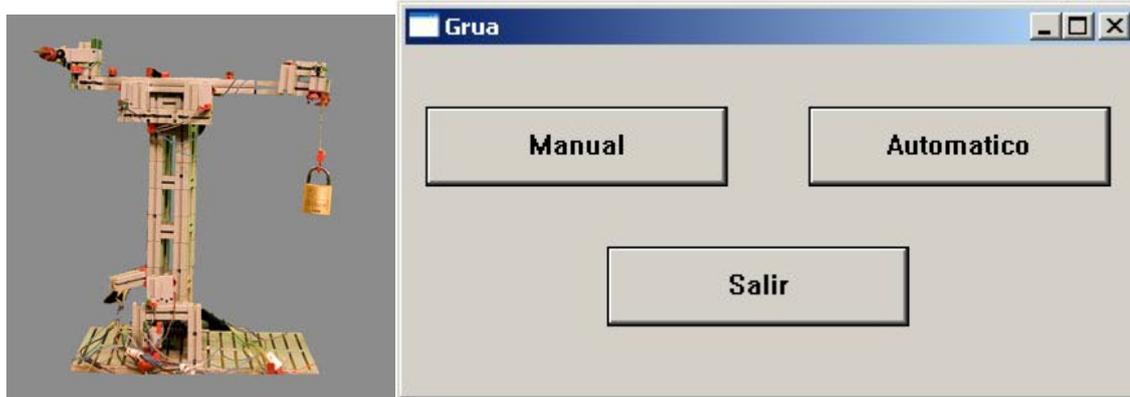
```
case 8:
    if(dato==1){
        motor2=4;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }else if(dato==0){
        motor2=0;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }
    break;
case 16:
    if(dato==1){
        motor3=16;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }else if(dato==0){
        motor3=0;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }
    break;
case 32:
    if(dato==1){
        motor3=32;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }else if(dato==0){
        motor3=0;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }
    break;
case 5:
    if(dato==1){
        motor1=2;
        motor2=8;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }else if(dato==0){
        motor1=0;
        motor2=0;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }
    break;
case 9:
    if(dato==1){
        motor1=2;
        motor2=4;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }else if(dato==0){
```

```
        motor1=0;
        motor2=0;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }
    break;
case 17:
    if(dato==1){
        motor1=2;
        motor3=32;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }else if(dato==0){
        motor1=0;
        motor3=0;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }
    break;
case 33:
    if(dato==1){
        motor1=2;
        motor3=16;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }else if(dato==0){
        motor1=0;
        motor3=0;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }
    break;
case 6:
    if(dato==1){
        motor1=1;
        motor2=8;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }else if(dato==0){
        motor1=0;
        motor2=0;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }
    break;
case 10:
    if(dato==1){
        motor1=1;
        motor2=4;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }else if(dato==0){
```

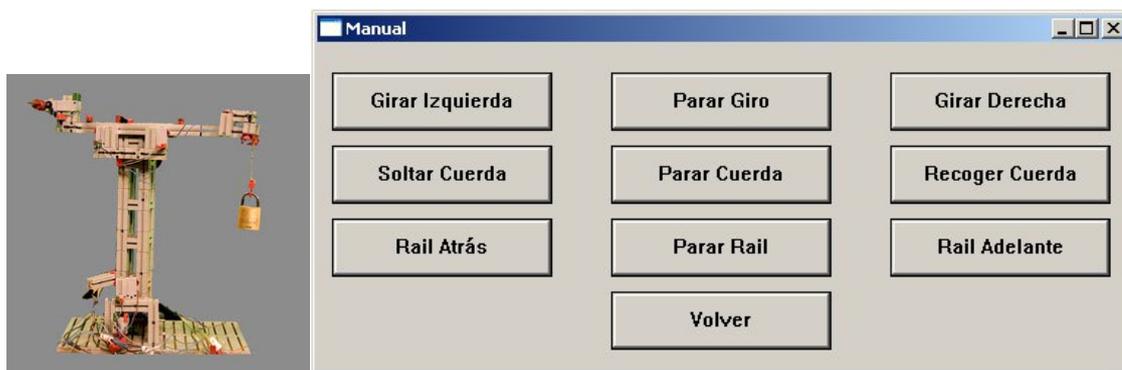
Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo

```
        motor1=0;
        motor2=0;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }
    break;
case 18:
    if(dato==1){
        motor1=2;
        motor3=32;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }else if(dato==0){
        motor1=0;
        motor3=0;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }
    break;
case 34:
    if(dato==1){
        motor1=2;
        motor3=16;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }else if(dato==0){
        motor1=0;
        motor3=0;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }
    break;
case 20:
    if(dato==1){
        motor2=8;
        motor3=32;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }else if(dato==0){
        motor2=0;
        motor3=0;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }
    break;
case 24:
    if(dato==1){
        motor2=4;
        motor3=32;
        encender(motor1+motor2+motor3);
        fflush(stdin);
    }else if(dato==0){
        motor2=0;
```


Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo



Seleccionando una de las opciones de ejecución de la grúa, se mostrara la siguiente pantalla:

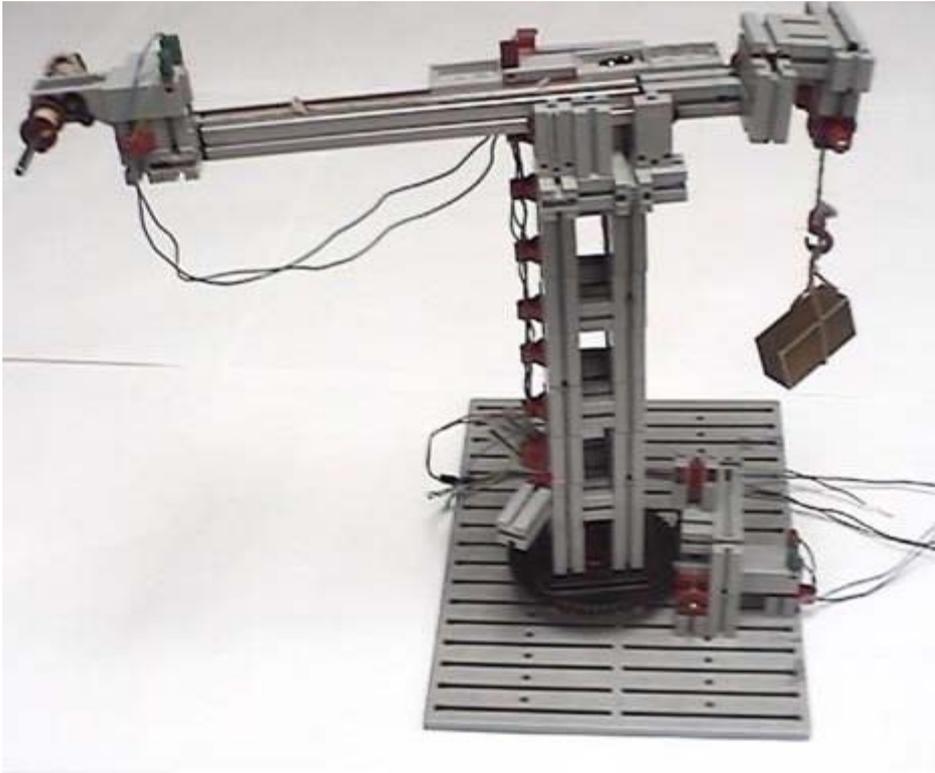


En ella bastará con pulsar los botones para realizar el movimiento que desee.

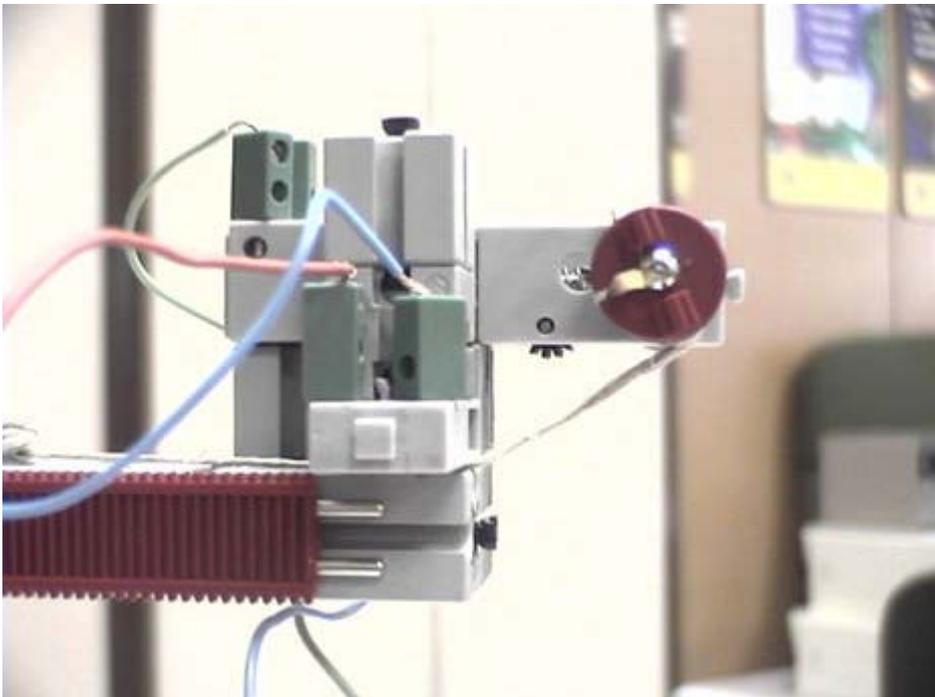
Nota:

En la aplicación programada con C, la imagen puede no aparecer al lado de la ventana. En este caso basta con mover nuestra ventana y se verá correctamente la imagen.

Imágenes

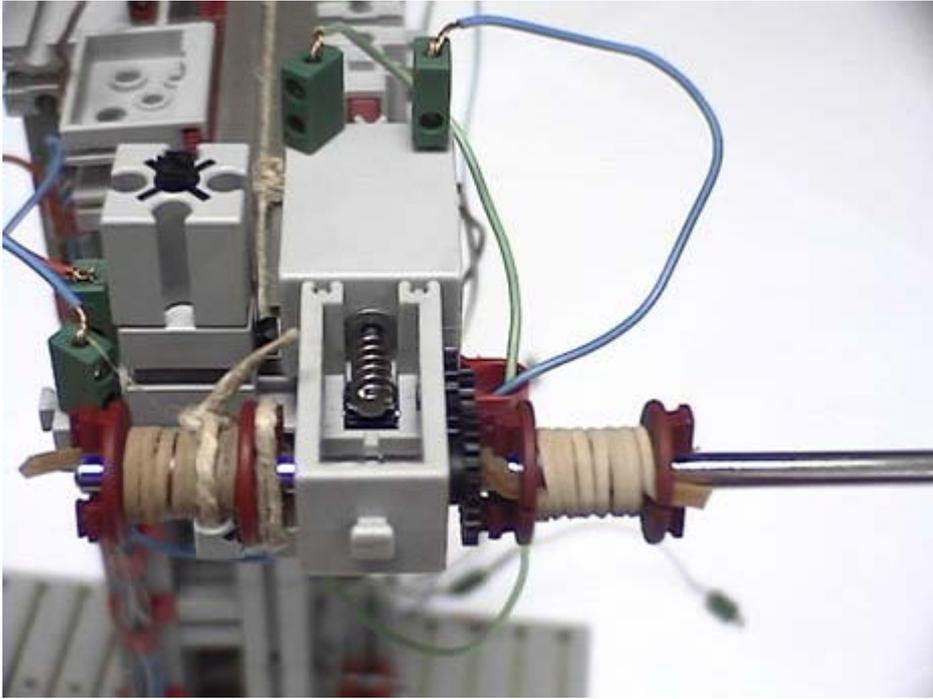


Vista completa de la grúa

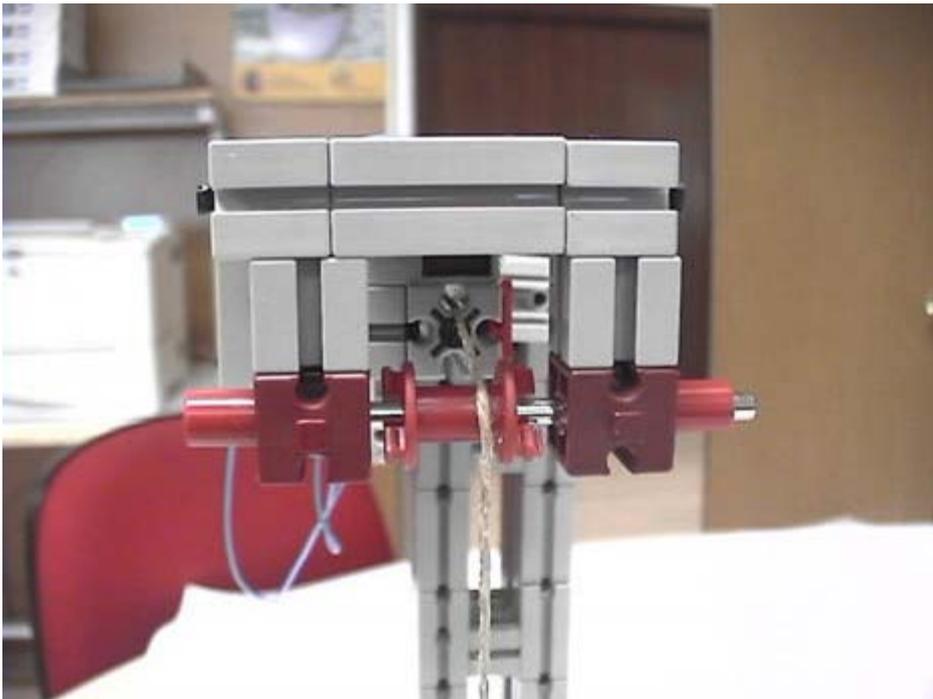


Vista lateral del motor

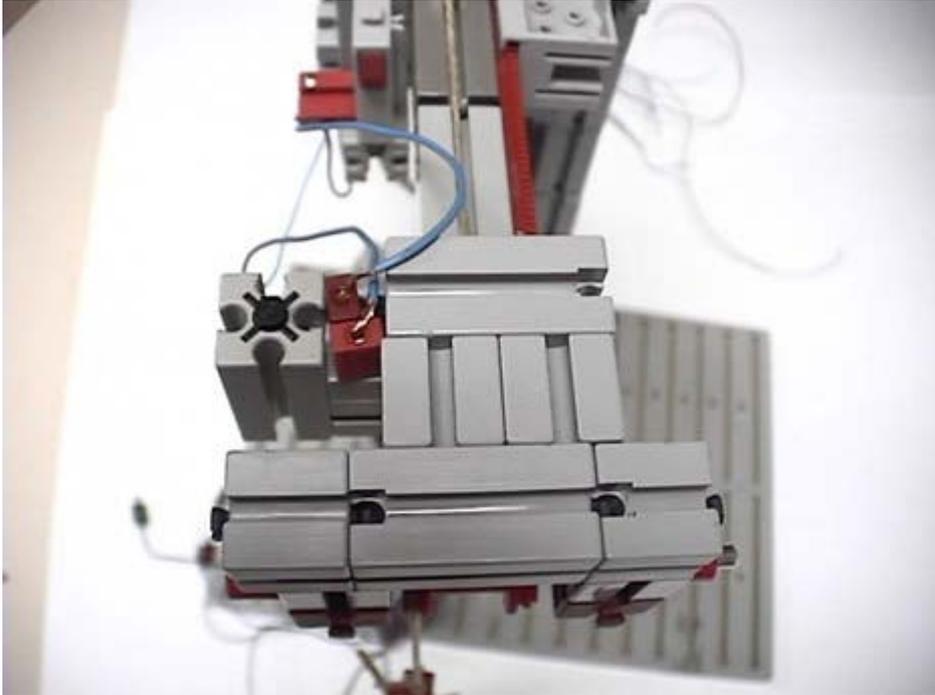
Interfaz de control de dispositivos externos por ordenador a través de puerto paralelo



Vista superior del motor



Vista trasera del retractor de la grúa



Vista superior del retrator de la grúa

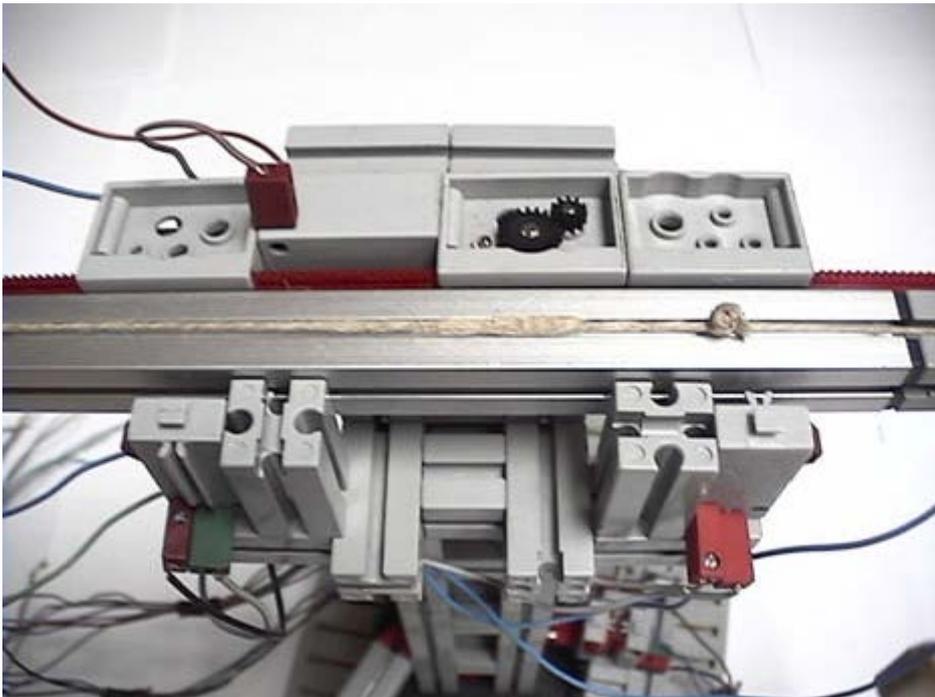


Imagen del motor de la barra