

DESPLIEGUE DE COORDENADAS EN EL TELESCOPIO DE 2 M.

**REPORTE TÉCNICO
OCTUBRE DE 1997**

**F. QUIROZ Y L. GUTIÉRREZ.
DEPARTAMENTO DE INSTRUMENTACIÓN, ENSENADA.**

ÍNDICE.

I. INTRODUCCIÓN.

II. JUSTIFICACIÓN.

III. OBJETIVO.

IV. SEÑALES DE CONTROL.

V. IMPLEMENTACIÓN DEL SISTEMA.

AGRADECIMIENTOS.

REFERENCIAS.

APÉNDICE A. DIAGRAMAS ESQUEMÁTICOS.

APÉNDICE B. ALAMBRADO DEL SISTEMA.

APÉNDICE C. PROGRAMA EN ENSAMBLADOR.

I. INTRODUCCIÓN.

El despliegue de coordenadas permite visualizar las coordenadas del telescopio en el piso del mismo.

El despliegue de coordenadas, consta de dos arreglos de seis dígitos, los cuales son totalmente independientes; cada uno cuenta con su electrónica y una fuente de alto voltaje (180 V) para alimentar a las lámparas de NEON que forman los dígitos.

En este reporte mencionaremos la electrónica de control, las modificaciones hechas, la filosofía de los manejadores de despliegue y la forma como materializamos la idea.

II. JUSTIFICACIÓN.

Durante el período activo de la anterior consola de control del telescopio de 2.12 m ésta contaba con un despliegue de NEON, ubicado en el mismo piso de telescopio; este despliegue era controlado por la electrónica de control, mediante una tarjeta conectada al ducto de la consola.

Mediante la implementación de un nuevo sistema de control para el telescopio este despliegue quedó inactivo, por lo que se pensó en hacer una nueva electrónica que controlara este despliegue de una de una manera más inteligente y autónoma, quedando como un periférico más de la nueva consola de control.

De esta manera se contaría nuevamente con el despliegue de las coordenadas del telescopio, visible desde cualquier parte del telescopio, lo que facilita el trabajo de los operadores del mismo.

III. OBJETIVO.

Para que el despliegue pudiera funcionar como un dispositivo de la nueva consola de control, la electrónica de control debe de cumplir con los siguientes puntos.

- **Tiene la capacidad de recibir mandos por un puerto serie.**
- **Descifrar la cadena de caracteres que manda la consola de control.**
- **Generar las fases que alimentan al despliegue de NEON.**

IV. SEÑALES DE CONTROL.

Para generar las fases de control para el despliegue utilizamos un microcontrolador, ya que cuenta con la potencia suficiente para recibir la cadena de caracteres, descifrarlas y, una vez con los datos de las coordenadas del telescopio, generar la secuencia necesaria para alimentar al despliegue de NEON.

Primero explicaremos la electrónica que alimenta directamente al despliegue de NEON, con la cual ya se contaba. También explicaremos las señales y las secuencias que dedujimos se debían generar. El despliegue completo consta de dos despliegues individuales e idénticos, por lo que solo nos referiremos a uno.

Con el fin de minimizar el número de líneas en el control de un despliegue de "n" número de dígitos, sólo se va prendiendo uno a la vez, con una frecuencia lo suficientemente alta para que el ojo humano no lo alcance a detectar y den la apariencia de estar todos los dígitos activos a la vez. Refiriéndonos a la electrónica que alimenta directamente a los dígitos de NEON (figura A1), se puede ver cómo se comparten los datos que se envían a cada uno de los dígitos a través del circuito integrado (DD700); para ir activando sólo un dígito a la vez, se utilizan las dos líneas de control RES y CLK que alimentan al CI 4017, que es el que va activando un dígito a la vez. Es importante entonces poner el dato correcto, uno a la vez, para que se despliegue en el dígito correcto. Por otra parte para indicar el signo de la coordenada se utiliza un LED, el cual es controlado directamente a través de un transistor.

Revisando los diagramas de tiempo para la generación de las señales en los circuitos CI 4017 y DD700, se obtiene el siguiente diagrama.

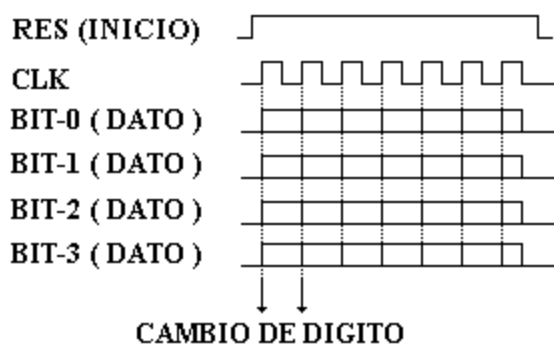


Figura 1. Señales de control del despliegue de NEON.

Las señales de control del despliegue se muestran en la figura 1, donde se muestran las dos líneas de control (RES y CLK) y las cuatro líneas de datos (Bit-0 - Bit-3). La línea RES es activa alto e indica el inicio del despliegue; con el flanco positivo de la línea CLK se va desplegando cada uno de los dígitos; en el primer flanco de la línea

CLK desplegamos el dígito 1, en el segundo flanco el dígito 2 y así sucesivamente. Pero, para esto cada vez que va a haber un flanco positivo de la línea CLK debemos poner previamente el dato correspondiente a cada dígito. Aprovechando las líneas RES y CLK, usamos las mismas señales para desplegar las dos coordenadas; de esta manera sólo utilizamos dos líneas de control, ocho líneas de datos y dos líneas para los signos de cada coordenada.

V. IMPLEMENTACIÓN DEL SISTEMA.

Una vez que tenemos la secuencia con la que alimentaremos al despliegue, nuestro siguiente paso es descifrar la cadena de caracteres que nos envía la consola de control del telescopio, que a fin de cuentas es la que lleva la posición real del mismo. Esta cadena la presentamos a continuación

:02 AR HH:MM:SS.S DEC GG°MM SS AH HH:MM:SS.S;

Por las limitantes del despliegue (sólo cuenta con dos renglones de seis dígitos), sólo desplegamos la posición en ascensión recta (AR) y en declinación (DEC); la información correspondiente al ángulo horario (AH) la deseamos, así como las décimas de segundo de ascensión recta.

Hablando en términos técnicos, se descifra la información de la cadena sobre la base de la posición de cada carácter en la cadena y, posteriormente, se cambia de su valor ASCII a su valor real en decimal. La comunicación se lleva a cabo mediante el estándar de comunicación RS-232, y los parámetros de comunicación se muestran en la tabla 1.

Parámetros de	Comunicación
Baud Rate	9600
Núm. de bits	8
Bits de paro	1
Paridad	No

Tabla 1. Parámetros de la comunicación serie.

Para materializar la idea, como ya lo habíamos mencionado, utilizamos un microcontrolador, el µC80C31, que es un microcontrolador de 8 bits de la familia INTEL. Para esto utilizamos una tarjeta comercial, manejada por el µC80C31 (Compañía DALLAS, modelo NMIY-0031).

Algunas características de esta tarjeta se muestran en los siguientes puntos

- 1 puerto de comunicación serie (RS-232 o RS-422 o RS-485).
- 1 puerto paralelo de 8 bits de salida.
- 1 puerto paralelo de 8 bits de entrada.
- 1 puerto bidireccional de 8 bits.
- 64 KB de memoria ROM.
- 64 KB de memoria RAM.
- 1 manejador de teclado.
- 1 manejador de despliegue LCD.
- 2 interrupciones y 2 contadores internos de 16 bits.

Los diagramas esquemáticos de la tarjeta NMIY-0031 se muestran en el apéndice A. Por la potencia y versatilidad de los microcontroladores la generación de las fases resultó ser la parte trivial del despliegue. Se utilizó el puerto de salida de 8 bits para la generación de los bits de datos de los dos despliegues y para la generación de las señales RES, CLK y los signos de coordenadas se usaron 4 bits del puerto bidireccional direccionable a bit. Los detalles sobre los bits que se usaron para la generación de las fases se muestran en la siguiente tabla.

Bit de salida	Función
Puerto A (Bit-0 - Bit-3)	Datos coord. AR
Puerto A (Bit-4 - Bit-7)	Datos coord. DEC
Signo DEC	Bit-0 del puerto 1
Signo AR	Bit-1 del puerto 1
RES	Bit-2 del puerto 1
CLK	Bit-3 del puerto 1

Tabla 2. Bits de salida del microcontrolador para la generación de las señales de control.

En la figura C1 del apéndice C se muestra un diagrama de flujo del programa en ensamblador, el cual siempre está refrescando el despliegue y revisando si llegó una nueva cadena de caracteres. El programa crea un buffer para ir recibiendo los datos del puerto serie para minimizar la pérdida de caracteres en la cadena de información.

AGRADECIMIENTOS.

Un sincero agradecimiento al departamento de mecánica de precisión, por las facilidades brindadas en la adaptación de la electrónica a la caja del despliegue.

También quiero agradecer a Salvador Zazueta por todos sus consejos referentes a electrónica.

REFERENCIAS.

The 8051 Microcontroller, Architecture, Programming and Applications .

Kenneth J. Ayala.

Western Carolina University.

Manual de la tarjeta NMIY-0031 .

Dallas Co.

Manuales de la consola del telescopio de 2.12 m, Versión anterior .

Luis Salas.

Instituto de Astronomía, UNAM.

Apuntes personales sobre microcontroladores .

Fernando Quiros.

Instituto de Astronomía, UNAM.

APÉNDICE A. DIAGRAMAS ESQUEMÁTICOS

APÉNDICE B. ALAMBRADO DEL SISTEMA.

Siguiendo la misma filosofía que se ha tomado en otros proyectos, se trató de modificar lo menos posible. La electrónica la introducimos en la caja donde se encontraba el despliegue y solo añadimos varios conectores e interruptores (encendido, cable de alimentación y un conector DB9 para la comunicación serie).

La tarjeta NMIY-0031 cuenta con dos conectores para cable plano (RIBBON), por donde se comunica con los dos despliegues. La distribución de la tarjeta de electrónica, los despliegues y fuentes de alimentación se muestran en el siguiente bosquejo.

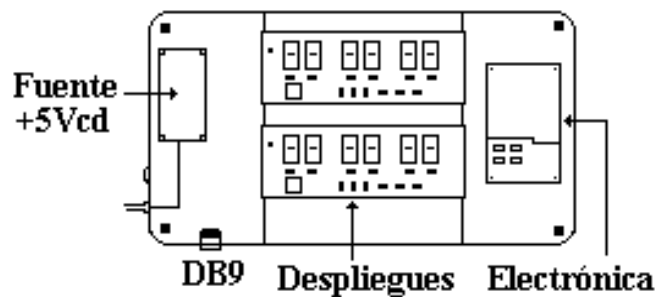


Figura B1. Distribución de las tarjetas en el despliegue de coordenadas.

APÉNDICE C. PROGRAMA EN ENSAMBLADOR

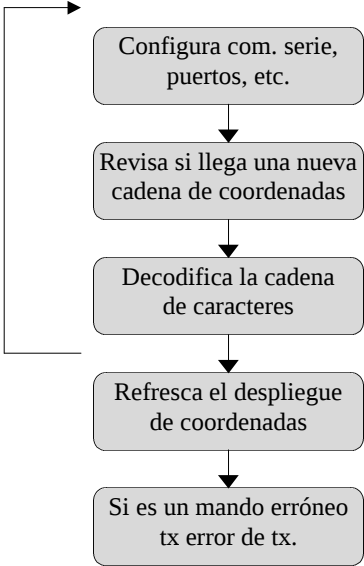


Figura C1. Diagrama a bloques de la estructura del programa en ensamblador.

A continuación se muestra el listado del programa en ensamblador.

_R	EQU	00H
_DIR	EQU	'0'
PTOB	EQU	0FFFFH
AR_HH	EQU	30H
AR_HL	EQU	31H
AR_MH	EQU	32H
AR_ML	EQU	33H
AR_SH	EQU	34H
AR_SL	EQU	35H
AR_DS	EQU	36H
DEC_GH	EQU	37H
DEC_GL	EQU	38H
DEC_MH	EQU	39H
DEC_ML	EQU	3AH
DEC_SH	EQU	3BH
DEC_SL	EQU	3CH
DEC_S	EQU	P1.0
AR_S	EQU	P1.1
RES	EQU	P1.2
CLK	EQU	P1.3
BUF	EQU	44H
ADR	EQU	43H

```

INI      EQU    42H
IND      EQU    41H
CTA      EQU    40H

RSEG APUN

RESET_P:
    LJMP  PROGRAMA_PRINCIPAL

    ORG   RESET_P+23H    ; MANEJADOR INT
SERIE
    JMP    MANEJADOR_INT_SERIE

    ORG   RESET_P+40H
MANEJADOR_INT_SERIE:
    PUSH  ACC
    PUSH  _R+0
    PUSH  _R+1
    PUSH  _R+2
    LCALL FUNCION_SERIE
    POP   _R+2
    POP   _R+1
    POP   _R+0

```

POP	ACC	MOV	A,AR_SL
RETI		SWAP	A
		ORL	A,DEC_SL
FUNCION_SERIE:		MOV	DPTR,#PTOB
JNB	RI,SALIR_INT_SERIE	MOVX	@DPTR,A
MOV	A,INI	SETB	RES
CJNE	A,#00H,CACHA	SETB	CLK
CLR	RI	CLR	RES
MOV	A,SBUF	LCALL	TIEMPITO
CJNE	A,#',SALIR_INT_SERIE	CLR	CLK
MOV	INI,#01H	LCALL	TIEMPITO
LJMP	SALIR_INT_SERIE	SETB	CLK
CACHA:		LCALL	TIEMPITO
MOV	A,ADR	CLR	CLK
CJNE	A,#00H,CACHA1	LCALL	TIEMPITO
CLR	RI		
MOV	A,SBUF	MOV	A,AR_SH
CJNE	A,#DIR,SALIR_INT	SWAP	A
MOV	ADR,#01H	ORL	A,DEC_SH
LJMP	SALIR_INT_SERIE	MOVX	@DPTR,A
CACHA1:		SETB	CLK
CLR	RI	LCALL	TIEMPITO
MOV	A,SBUF		
MOV	R2,A	MOV	A,AR_ML
CJNE	A,#',CACHA2	SWAP	A
LJMP	SALIR_INTF	ORL	A,DEC_ML
CACHA2:		CLR	CLK
MOV	A,IND	LCALL	TIEMPITO
MOV	R0,A	MOVX	@DPTR,A
MOV	A,#BUF	SETB	CLK
ADD	A,IND	LCALL	TIEMPITO
MOV	R1,A		
MOV	A,R2	MOV	A,AR_MH
MOV	@R1,A	SWAP	A
INC	IND	ORL	A,DEC_MH
LJMP	SALIR_INT_SERIE	CLR	CLK
SALIR_INTF:		LCALL	TIEMPITO
MOV	A,IND	MOVX	@DPTR,A
MOV	R0,A	SETB	CLK
MOV	A,BUF	LCALL	TIEMPITO
ADD	A,R0		
MOV	R1,A	MOV	A,AR_HL
MOV	@R1,#','	SWAP	A
MOV	CTA,#01H	ORL	A,DEC_GL
SALIR_INT:		CLR	CLK
MOV	INI,#00H	LCALL	TIEMPITO
MOV	ADR,#00H	MOVX	@DPTR,A
SALIR_INT_SERIE:		SETB	CLK
RET		LCALL	TIEMPITO
PROGRAMA_PRINCIPAL:			
LCALL	CONFIGURA	MOV	A,AR_HH
MOV	R5,#'@'	SWAP	A
LCALL	MANDA_CAR	ORL	A,DEC_GH
INICIO:		CLR	CLK
MOV	A,#00H	LCALL	TIEMPITO
MOV	CTA,A	MOVX	@DPTR,A
MOV	IND,A	SETB	CLK
MOV	INI,A	LCALL	TIEMPITO
MOV	ADR,A		
RECIBE_CAD:		MOV	A,#00H
LCALL	REFRESH	CLR	CLK
MOV	A,CTA	LCALL	TIEMPITO
JZ	RECIBE_CAD	MOVX	@DPTR,A
LCALL	DECODIFICA	SETB	CLK
LJMP	INICIO	LCALL	TIEMPITO
REFRESH:		SETB	RES
		FIN_REFRESH:	

```

    RET

TIEMPITO:
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    NOP
    RET

CONFIGURA:
    LCALL LIMPIA_VAR
    MOV    TL1,#0FDH
    MOV    TH1,#0FDH
    ORL    TMOD,#20H
    SETB   TCON.6

    MOV    SCON,#50H
    CLR    IP.4
    SETB   IE.4
    SETB   IE.7

RET

DECODIFICA:
    MOV    A,BUF
    CLR    C
    SUBB   A,#2'
    JNZ    FIN_DECOD
    SETB   AR_S
    MOV    A,BUF+3
    CJNE   A,#'-',DECOD_0
    CLR    AR_S

DECOD_0:
    MOV    A,BUF+4
    LCALL  CONV_A_H
    MOV    AR_HH,A
    MOV    A,BUF+5
    LCALL  CONV_A_H
    MOV    AR_HL,A
    MOV    A,BUF+7
    LCALL  CONV_A_H
    MOV    AR_MH,A
    MOV    A,BUF+8
    LCALL  CONV_A_H
    MOV    AR_ML,A
    MOV    A,BUF+0AH
    LCALL  CONV_A_H
    MOV    AR_SH,A
    MOV    A,BUF+0BH
    LCALL  CONV_A_H
    MOV    AR_SL,A
    MOV    A,BUF+0DH
    LCALL  CONV_A_H
    MOV    AR_DS,A

    SETB   DEC_S
    MOV    A,BUF+12H
    CJNE   A,#'-',DECOD_1
    CLR    DEC_S

DECOD_1:
    MOV    A,BUF+13H
    LCALL  CONV_A_H
    MOV    DEC_GH,A
    MOV    A,BUF+14H
    LCALL  CONV_A_H
    MOV    DEC_GL,A
    MOV    A,BUF+16H
    LCALL  CONV_A_H
    MOV    DEC_MH,A
    MOV    A,BUF+17H
    LCALL  CONV_A_H
    MOV    DEC_ML,A
    MOV    A,BUF+19H
    LCALL  CONV_A_H
    MOV    DEC_SH,A
    MOV    A,BUF+1AH
    LCALL  CONV_A_H
    MOV    DEC_SL,A

    MOV    R5,#'@'
    LCALL  MANDA_CAR

FIN_DECOD:
    RET

MANDA_CAR:
    CLR    IE.7
    MOV    A,R5
    MOV    SBUF,A
AQUI_M: JNB  TI,AQUI_M
    CLR    TI
    SETB   IE.7
    RET

CONV_A_H:
    CLR    C
    SUBB   A,#30H
    RET

LIMPIA_VAR:
    MOV    AR_HH,#01H
    MOV    AR_HL,#02H
    MOV    AR_MH,#03H
    MOV    AR_ML,#04H
    MOV    AR_SH,#05H
    MOV    AR_SL,#06H
    MOV    AR_DS,#07H
    MOV    DEC_GH,#08H
    MOV    DEC_GL,#09H
    MOV    DEC_MH,#00H
    MOV    DEC_ML,#01H
    MOV    DEC_SH,#02H
    MOV    DEC_SL,#03H
    SETB   AR_S
    CLR    DEC_S

RET

    ORG    RESET_P+1FFDH
    LJMP   RESET_P

END

```