

**Escuela Superior Politécnica del Litoral**

**Facultad de Ingeniería en Mecánica**

**“ PROGRAMA COMPUTACIONAL PARA EL CONTROL  
DE LOS ELEMENTOS DE UNA RED DE RIEGO ”**

**Proyecto de Grado**  
PREVIO A LA OBTENCION DEL TITULO DE:  
**Ingeniero Mecánico**

Realizado por :

**RODOLFO ANTONIO JACOME AGUIRRE**

Guayaquil - Ecuador  
1993

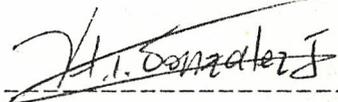
DECLARACION EXPRESA.

" La responsabilidad por los hechos, ideas y doctrinas expuestos en este Proyecto de Grado, me corresponden exclusivamente; y, el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITECNICA DEL LITORAL".

(Reglamento de tópicos de graduación).

A handwritten signature in black ink, reading "Rodolfo Jacome". The signature is stylized and written over a horizontal dashed line.

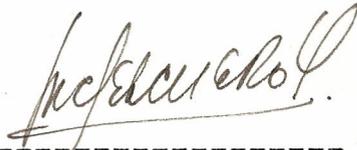
RODOLFO JACOME



---

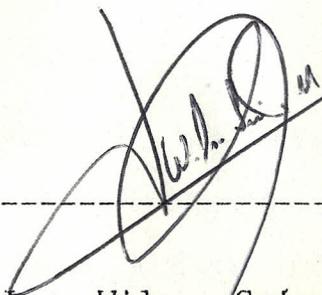
Dr. Alfredo Barriga  
Decano  
Facultad de Ingeniería  
en Mecánica

Ing. Víctor H. González  
Director de  
Proyecto de Grado



---

Ing. Manuel Helguero  
Miembro del Tribunal  
de Grado



---

Ing. Wilson Suárez  
Miembro del Tribunal  
de Grado

AGRADECIMIENTO.

AL ING. VICTOR H. GONZALEZ, POR  
SU INVALORABLE COLABORACION EN  
LA CULMINACION DE ESTE TRABAJO,  
Y A MIS AMIGOS POR SU APOYO EN  
LOS MOMENTOS DIFICILES.

DEDICATORIA

A MIS PADRES

A MI HERMANO

A MI HERMANA

A MIS FAMILIARES, LOS QUE ESTAN,  
LOS QUE SE FUERON Y LOS QUE VENDRAN

A MIS AMIGOS.

## RESUMEN

El desarrollo de la automatización en la agricultura ha hecho imprescindible la aparición de nuevas formas de control de los procesos agrícolas ( cosecha, postcosecha, riego), en el área de riego se ha tratado de minimizar la acción humana en el desarrollo de las plantas, disminuyendo costos y mejorando la calidad de los productos aplicando la cantidad correcta de riego en un tiempo dado. El presente proyecto:

" Programa computacional para el control de los elementos de una red de riego", tiene como objetivo el control de la red de riego de acuerdo a las necesidades hídricas del cultivo y los parámetros meteorológicos del medio.

Se ha escogido el control por computadora, ya que este instrumento se ha popularizado en nuestro medio siendo de fácil adquisición por los usuarios que deseen un sistema de riego controlado.

En el capítulo I procedemos a la introducción del trabajo, realizando la descripción de la red experimental y estableciendo los beneficios de la automatización en riego. En el capítulo II, se realiza una breve explicación de adquisición de datos. Luego en el capítulo III, se estudian los

fundamentos de control, la tarjeta utilizada, los parámetros a controlar y se presenta el programa de control de la red con su respectivo ejemplo de utilización.

Por último en el capítulo IV se realizará la evaluación experimental de la red, con sus respectivas pruebas parciales y el análisis global del proceso; en el capítulo V se efectúa las conclusiones y recomendaciones.

## INDICE GENERAL

Resumen.....	VI
Indice General.....	VIII
Indice de Figuras.....	X
Indice de Tablas.....	XI

### 1. Introducción

1.1 Descripción de la red experimental.....	1
1.2 Factibilidad económica y beneficios de la <i>automatización en riego</i> .....	3

### 2. Adquisición de datos

2.1 Explicación de la adquisición de datos.....	7
2.2 Descripción de los instrumentos utilizados (tarjeta computada) .....	11
2.3 Programa de adquisición de datos .....	17

### 3. Control automático de la red

3.1 Fundamentos de control automático .....	21
3.2 Descripción de los elementos de la red .....	28
3.3 Programa de control de la red .....	33

4. Evaluación experimental de la red	
4.1 Pruebas parciales .....	46
4.2 Análisis integral del proceso .....	46
5. Conclusiones y Recomendaciones.....	49

BIBLIOGRAFIA

INDICE DE FIGURAS

- FIGURA 1.1 DIVISION DE PARCELAS Y VALVULAS SOLENOIDES.
- FIGURA 2.1 LAZO DE ADQUISICION DE DATOS Y CONTROL DE PROCESOS.
- FIGURA 2.2 CONVERSION COMPLETA DE ADQUISICION DE DATOS.
- FIGURA 2.3 ORDEN DE EJECUCION DE LENGUAJE C.
- FIGURA 3.1 MALLA CERRADA. SIN RETROALIMENTACION.
- FIGURA 3.2 MALLA ABIERTA O DE REALIMENTACION.
- FIGURA 3.3 VALVULA SOLENOIDE.
- FIGURA 3.4 FILTRO DE MALLA.
- FIGURA 3.5 MANOMETRO.

INDICE DE TABLAS.

TABLA I. ANALISIS DE PROCESOS DE CONTROL.

TABLA II. PERDIDA DE PRESION EN VALVULAS SOLENOIDES.

## CAPITULO I

### INTRODUCCION

#### 1.1 Descripción de la red experimental.

La red de riego experimental consta de cuatro parcelas en una superficie total de 700 m<sup>2</sup>, ubicadas en el área destinada a la investigación agroindustrial en la Facultad de Ingeniería en Mecánica (ESPOL), tres parcelas son de ciclo corto (melón, tomate y pimiento) y otra parcela de mango cuyo período de crecimiento hasta su cosecha es de cuatro años. Los cultivos de ciclo corto constan de un sistema de riego por goteo mientras que el mango un sistema de riego por microaspersión, cada una de las parcelas encontramos dispuestas válvulas solenoides (Figura 1.1), que controlan el fluido de agua hacia los cultivos, tenemos además un filtro de malla en los cultivos de ciclo corto; se evita la obstrucción de los goteros, también tenemos un manómetro para conocer la presión de trabajo.

En el control automático se usa un computador que se encarga de recibir y enviar los datos en el puerto paralelo, la unidad central es el componente principal del sistema que usa un microprocesador

8088.

En el software utilizado es en lenguaje C que actúa en el momento que está sensando la humedad del terreno, para ordenar o activar las válvulas para que se inicie el riego.

## 1.2 Factibilidad económica y beneficios de la automatización en riego.

En la implantación para la automatización de cualquier proceso normalmente debe seguirse el siguiente esquema.

- Estudio del producto cuyo proceso de fabricación debe automatizarse.
- Estudio del proceso de fabricación.

De los estudios previos deben sacarse las siguientes conclusiones:

- Posibles modificaciones del producto y del proceso en vista a facilitar la automatización.
- Si se puede dar el caso, la nueva distribución de la planta.
- Determinar las operaciones independientes o puntos del proceso donde no es compatible la

automatización.

- Determinación de la tecnología más adecuada y la explotación máxima de las posibilidades de esta tecnología.
- Determinar la inversión necesaria, rentabilidad, amortización, volumen estimado de producción para lograr la máxima rentabilidad, costos del estudio e instalación del nuevo sistema, etc.
- Instalación y puesta a punto.

Debe convencerse al operador sobre los beneficios del nuevo sistema e instruirle en el correcto manejo de la nueva instalación. Aparte de las ventajas específicas que una automatización reporta en general, existe una serie de ventajas que se dan de manera particular en una instalación de riego.

- Reducción de los costo de mano de obra directos, es decir, al existir un riego controlado, el crecimiento de maleza disminuye dependiendo del tipo de cultivo y del sistema de riego.
- Uniformidad de la producción (cosecha) y consiguiente ahorro de mano de obra en el clasificado del producto.
- Aumenta capacidad de producción, al tener la

misma extensión del terreno y con menos mano de obra.

- Mejora relación cliente-productor al tener un producto de calidad.

Seguidamente se presenta una guía muy general, para realizar un estudio económico, que debe estar acompañado al estudio técnico de la instalación de riego, se debe tomar en cuenta lo siguiente.

- Costo del nuevo equipo o modificación del antiguo sistema de riego (red de tuberías, sistema de bombeo, tipo de aspersores, toma de agua, etc.).
- Costo del estudio e instalación del nuevo sistema.
- Costo del tiempo de parado de la instalación del sistema de riego, si es el caso de daño en las tuberías, aspersores, bomba, etc.
- Variación del costo de la cosecha.
- Consumo de energía en el nuevo sistema.
- Variabilidad del sistema de riego para diversos productos.
- Extensión del área de cultivo o el número de plantas para hacer rentable el nuevo sistema de riego.
- Vida útil del nuevo sistema de riego.

Dependiendo del estudio técnico como económico, se puede determinar la rentabilidad o no, de realizar una instalación de riego, además de los puntos anteriores se debe revisar los factores climatológicos de la zona en donde se quiere realizar la instalación.

## CAPITULO II

### ADQUISICION DE DATOS

#### 2.1 Explicación de la adquisición de datos.

Para realizar el trabajo de adquisición de datos y control de la red (riego), existen un gran número de medidores a los que se les denomina "sensores", estos no solamente sirven para la automatización de máquinas y plantas, también se los ha utilizado en la medición del medio ambiente (suelo, humedad, temperatura, etc).

La selección del sensor apropiado incide sobre el volumen de trabajo del diseño y también sobre el costo de la construcción del sistema. Las señales de los sensores son preparadas adecuadamente, pueden ser adquiridas por la computadora y entonces forman un sistema muy poderoso para controlar procesos, y fácilmente mejorar la capacidad de los controladores tradicionales; muchas veces a precios inferiores.

En la figura 2.1 se muestra el lazo de adquisición de datos y control de procesos.

Como se observa en la figura 2.1 muy rara vez

pueden los sensores ser usados directamente, es necesario realizar ciertas transformaciones.

En la figura 2.2 se muestra una conversión completa de adquisición de datos, este diagrama muestra los componentes requeridos para el sistema interconectado.

Los parámetros físicos a medir, son convertidos a una señal eléctrica por el transductor y luego alimentado a un amplificador (eleva la señal de 1 a 5 voltios), luego la señal es filtrada para eliminar frecuencias alternantes o ruidos de la señal (perturbaciones), esta señal se dirige a un multiflexor analógico que conmuta secuencialmente varios canales de entrada de señal con un convertidor analógico digital. La salida del multiflexor va al circuito de muestreo, el cual, muestrea la salida del multiflexor durante un tiempo específico y luego mantiene el nivel de voltaje hasta la operación de conversión analógico a digital.

La secuencia del programa, manda señales de control digitales a los actuadores para controlar el proceso.

El convertidor A/D para una entrada de voltaje

analógico dado en el comparador, todos los comparadores bajo este nivel de entrada se accionan y todos los comparadores arriba de este nivel de entrada se apagan instantáneamente. El método emplea un cuantificador que es completado en el tiempo de interrupción de un sólo comparador.

La salida del comparador, no está en código binario y tiene por lo tanto que ir a un decodificador.

Hay un número de parámetros importantes usados para caracterizar la precisión de un convertidor A/D y D/A.

Estos parámetros son: Resolución, linealidad, diferencial, monotonía, error, precisión relativa, precisión absoluta, error de compensación, error de ganancia, etc.

## 2.2 Descripción de los elementos de adquisición y control.

Para ejecutar la adquisición de datos y control en cualquier proceso se necesitan de tres accesorios que son imprescindibles los cuales son:

- La tarjeta de adquisición de datos.
- El micro computador.

- El programa de adquisición y control personal.

Estos procesos automáticos son gobernados desde el puerto paralelo, este recibe señales que se encargan de energizar o desenergizar las válvulas.

#### FUNCIONAMIENTO DE LA TARJETA DE CONTROL

En el diseño de la tarjeta se utilizó un transformador de 110 V (AC) a 26 V (AC). Esto tuvo que realizarse para poder tener a la entrada del regulador de voltaje AN7812, 15 V (DC) y a la salida 12 V (DC), el mismo que es regulado a 5 V (DC) a través de un transistor Q1, que es el voltaje que servirá para activar los relés. Usando el comando (out portb) del Lenguaje C se envían ceros (0) y unos (1) a través del puerto paralelo de la computadora, la tarjeta fue diseñada con el objetivo de que se activen los relés con una salida del computador de 0(V), ya que inicialmente al ingresar al lenguaje C, todas las salidas están a 5 voltios.

En el momento que mandamos un nivel lógico (bajo) a través de la micro este nivel va a la entrada de una puerta NAND que invertirá el nivel a alto y que

está conectado a la base de un transistor que va a trabajar en corte y saturación. Cuando el transistor esta en saturación van a caer los 5 voltios en la bobina del relé que hará cerrar sus contactos para activar las válvulas solenoides.

Un diodo va a estar en paralelo con la bobina para evitar que esta se quede cargada.

#### EL COMPUTADOR PERSONAL .

Este equipo es un sistema con un teclado y una disquetera. La unidad central es el componente principal del sistema que usa un microprocesador 8088 y soporta diversas opciones. El teclado se utiliza para enviar datos al sistema; y está conectado mediante un cable extensible. La unidad de diskette instalada en la unidad central, lee datos desde el diskette y graba los mismos.

El hardware por si mismo, no constituye un computador personal totalmente operativo, sino que deben estar presentes dos elementos más que son: el software y el usuario.

En el interior del computador se tiene una tarjeta MPG-CARD, la posee dos puertos, el uno es un adaptador de display y el otro un puerto paralelo.

Este última se lo usa para introducir los datos que sensa el computador y a su vez efectua el control sobre una de las válvulas solenoides. Este puerto es un corrector hembra DB25 cuyas especificaciones son las siguientes:

NUMERO DE LA SEÑAL	NUMERO DEL PIN
STROBE	1
DATA BIT 0	2
DATA BIT 1	3
DATA BIT 2	4
DATA BIT 3	5
DATA BIT 4	6
DATA BIT 5	7
DATA BIT 6	8
DATA BIT 7	9
ACK	10
BUSY	11
PAPER END	12
SELECT	13
AUTO FEED	14
ERROR	15
INITIALIZE PRINTER	16
SELECT INPUT	17
GROUND	18-25

La tarjeta MGP-CARD es seteada a través de dos jumpers para habilitar el puerto LPT1 que es el que se usa en el proyecto y por lo tanto queda habilitado el registro de impresión de datos,

controlado por el puerto de dirección 3BC.

Para ingresar los datos al micro se usa un registro de estado, el cual es gobernado por la dirección 3BD. Este registro de estado es un registro solo de lectura cuyas especificaciones son las siguientes:

BIT 7	BUSY
BIT 6	ACK
BIT 5	PE
BIT 4	SELECT STATUS
BIT 3	ERROR
BIT 2	RESERVES
BIT 1	RESERVES
BIT 0	RESERVES

El programa que se usa para realizar el proceso automático de adquisición central de datos está hecho en Lenguaje C, el cual posee básicamente dos instrucciones: de entrada y salida.

La instrucción envía un byte a un vía de acceso de salida a través de el puerto de dirección 8 H3 BC y la instrucción retoma el byte leído desde la vía de entrada y se almacena en un registro a través del puerto de dirección 8 H3 BD.

El programa que corre en la computadora está conjuntamente sensando que el terreno esté húmedo y lo hace recibiendo información a través de tarjetas sensoras de humedad que son las que

indican las condiciones en que se encuentra el terreno. De acuerdo a esta información, el computador manda a activar las válvulas solenoides.

#### LENGUAJE DE ADQUISICION Y CONTROL

Los lenguajes de programación para control de proceso tienen como objetivo, cuando ya se ha definido el problema a resolver: el control de un proceso, establecer la forma en que el equipo tiene que actuar para realizar su función indicando todas las actividades, cálculos y otras operaciones a efectuar y la forma de realizarlas, con todas las circunstancias que puedan darse: independencia, tratamiento de señales externas, informes, etc.

Los lenguajes clásicos son de tipo secuencial en los que las acciones tienen un estricto orden en el tiempo sin influir en el comportamiento del problema, la duración de su ejecución.

Las características de los lenguajes avanzados son por una parte, el carácter estructurado del control y la abstracción funcional (procedimientos); y por otra, la posibilidad de definir tipos de datos: simples o estructurados.

### 2.3 Programa de adquisición de datos.

El software utilizado para el control de la red de riego se realizó en lenguaje C que es un lenguaje de propósito general. Sus principales características son:

- Programa estructurado.
- Economía en las expresiones.
- Abundancia en operadores y tipos de datos.
- Codificación en alto y bajo nivel simultáneamente.
- Reemplaza ventajosamente la programación ensamblador.
- Utilización natural de las funciones primitivas del sistema.
- Facilidad de aprendizaje.
- Producción de código, objeto altamente optimizado.
- No está orientado a ningún área en especial.

Nació en los laboratorios de la Bell Telephone y ha sido estrechamente asociado con el sistema operativo UNIX como el propio compilador C y la casi totalidad de los programas y herramientas de UNIX, fueron escritos en C. El lenguaje C tiene una

gran cantidad de operadores, se puede decir que posee un operador para cada una de las posibles operaciones en código máquina.

Existen múltiples operaciones que se pueden hacer con el lenguaje C que realmente no están incluidas en el compilador propiamente dicho, sino que les realiza un preprocesador justo antes de cada compilación. Es un lenguaje altamente transportable y para programar lo improgramable, pero al igual que otros lenguajes no es perfecto, no tiene instrucciones de entrada/salida, carece de instrucciones para manejo de caracteres, por lo que este trabajo queda para la librería de rutinas, con la consiguiente pérdida de transportabilidad. La excesiva libertad en la escritura de los programas puede llevar a errores en la programación que, por ser correctos sintácticamente no se detectan a simple vista. Sin embargo las precedencias de los operadores convierten a veces las expresiones en pequeños rompecabezas.

A pesar de todo, el lenguaje C ha demostrado ser un lenguaje extremadamente eficaz y expresivo. Para realizar un programa en C se deben realizar los siguientes pasos (figura 2.3) en el orden de

ejecución.

Este ejemplo indica que a su vez editados los ficheros fuente A.C. y B.C, son compilados, obteniéndose los ficheros objeto A.OBJ y B.OBJ los cuales son enlazados con el fichero C.OBJ, con la librería D.LIB con las librerías del sistema LIB. dando lugar a un único fichero ejecutable A.EXE.

### CAPITULO III

#### CONTROL AUTOMATICO DE LA RED

##### 3.1 Fundamentos de control automático.

*Algunos principios de control automático ya eran utilizados en el siglo tercero antes de Cristo, Tesibio, inventor griego, diseñó en esa época una clepsidra que utilizaba un mecanismo similar al usado en los carburadores de los automóviles actuales. Los sistemas de control automático comenzaron su desarrollo intenso durante la revolución industrial.*

En esa época, James Watt adaptó el primer regulador automático de velocidad a la máquina de vapor.

Durante la segunda guerra mundial, el interés en las aplicaciones bélicas, hizo que se consideraran problemas de dirección y guía de los proyectiles balísticos, lo que tuvo como consecuencia el estudio de sistemas estocásticos y no lineales.

Con el advenimiento de las computadoras digitales se han redescubierto las variables de estado y se han estudiado más profundamente problemas tales como optimización y control bajo incertidumbre; además se ha incorporado el desarrollo del control numérico

y el control jerárquico con aplicaciones diversas.

Las aplicaciones de los sistemas de control cubren un amplio dominio, desde el diseño de dispositivos de control de extremada precisión, como delicados instrumentos empleados en la navegación inercial, hasta el diseño de equipos masivos, como los utilizados para el control de la manufactura del acero y otros procesos industriales.

En los últimos 15 años en especial países como Israel y los Estados Unidos se ha procedido a la automatización de la agricultura (riego, cosecha, postcosecha), dando una nueva área de aplicación del control automatizado.

El problema de control consiste en seleccionar, de un conjunto específico o arbitrario de elementos (parámetros, configuraciones, funciones del tiempo, etc.), aquellos que aplicados a un sistema fijo, hagan que este se comporte de una cierta manera deseada.

Por ejemplo, un problema de control es seleccionar el punto de apoyo de la palanca de un regulador de nivel para que la altura del líquido en el recipiente se mantenga constante a pesar de las variaciones de gasto de salida.

Otro ejemplo, la composición del torrente de salida de un reactor químico depende de la temperatura y composición de flujo de entrada.

Un tercer ejemplo es el que se plantea a continuación: el riego de una parcela se la debe realizar a una cierta hora y una determinada cantidad de agua.

Analizando los diversos problemas siempre se encontrarán tres elementos en común:

- 1) Uno que se puede modificar, llamado entrada.
- 2) Otro llamado salida, que se desea que tenga ciertas características, y
- 3) Un tercero, llamado planta o proceso, que relaciona la entrada con la salida y que no puede ser modificado.

Seguidamente se muestra una tabla donde se identifica las soluciones a los problemas planteados.

TABLA I. ANALISIS Y SOLUCION DE PROBLEMAS.

Problema	Entrada	Salida	Planta /Proceso
Control de nivel de apoyo	Localización del punto de líquido	Variación en el nivel de sistemas	Relaciones mecánicas del Reactor
Temperatura del flujo alimentación	Composición del torrente de salida	Relaciones de balance y cinética del reactor	Reactor
Control de riego	Necesidad humedad del suelo	Humedad del suelo a la hora y cantidad	Sistemas de aspersores

En particular, cuando la entrada y la salida son funciones del tiempo, puede resolverse el problema de dos maneras diferentes. Una llamada de "malla cerrada" (Fig. 3.1), consiste en seleccionar de antemano la entrada, en función del tiempo, con la cual se obtendrá la salida deseada. Esta entrada se aplica al sistema de una manera preprogramada.

La otra forma, llamada de "malla abierta" o de realimentación (Fig. 3.2). La variable controlada (salida) es comparada con la variable de referencia (entrada) y cualquier diferencia que exista entre ambos (el error) es usada para reducir esta última.

Las principales características de un sistema de

control es la estabilidad, exactitud y rapidez de respuesta. La estabilidad de una señal o una perturbación debe alcanzar y mantener un valor útil durante un período razonable. Al ser exacto dentro de ciertos límites especificados, es decir, debe reducir cualquier error a un valor aceptable; la exactitud es muy relativa y sus límites están basados en la aplicación particular que se habla del sistema de control. Un sistema de control debe completar su respuesta a cierta señal de entrada en un tiempo aceptable. Aunque un sistema sea estable y tenga la exactitud requerida, no tiene ningún valor si el tiempo para responder totalmente a una respuesta es mucho mayor que el tiempo entre las señales.

En la industria de procesos, manufactura, o en la agricultura, los controladores más usados han sido todo/nada (válvulas de dos posiciones) y los basados en las acciones: proporcional, integral y derivativa.

El más sencillo controlador es el digital o todo/nada.

Su utilización más corriente tiene lugar cuando su acción desencadena la evolución del proceso en un sentido y su inhibición lleva al proceso a evolucionar en sentido contrario o su término.

### 3.2 Descripción de los elementos de la red.

Como ya se dijo en el capítulo 1 el proyecto experimental consta de 4 parcelas, tres con riego por goteo y otra con riego por microaspersión.

A continuación se hará una descripción de los características más importantes.

1. Válvulas solenoides marca RAIN BIRD modelo 100V con conexión de entrada y salida de 1" (Fig. 3.3)

Especificaciones de operación:

- Flujo : 3 a 40 G.P.M.
- Presión : 15 a 150 psi
- Temperatura : 125°F a 50°C máximo

Características técnicas:

- Fácil mantenimiento.
- De manejo manual a falta de fluido eléctrico (1/4 de vuelta).
- Diafragma de balance de presión de larga vida.
- Opera a presiones bajas.
- Diseñada para trabajar con poca energía.
- Constitución durable, anticorrosiva, acero altamente aleado.

Características eléctricas:

Tensión : 24 V AC 50/60 Hz 0.30 Amn.

de salida 0.19 Amn.

Potencia: 2 W

Además se presenta en la tabla II presentamos las pérdidas de presión en las válvulas solenoides modelo 100V.

TABLA II. PERDIDA DE PRESION EN VALVULAS SOLENOIDES MOD. 100V

Flujo (GPM)	Pérdidas de Presión (psi)
3	0.5
5	1.0
10	2.6
20	3.6
30	6.5
40	9.0

## 2. Filtro de malla RB4-100 RAIN BIRD

El filtro de malla RB4-100 con conexión de entrada y salida de 1" (Fig. 3.4).

Especificaciones de operación:

Flujo : 5 a 30 GPM.

Presión : 10 a 120 psi

Características técnicas:

- Durable, es un filtro económico para sistemas de bajo flujo.
- Cuerpo compacto en forma de Y fabricado en polipropileno.
- Un filtro de rejilla de poliéster muy durable con *armazón de polipropileno.*
- Selección de 4 reemplazos de trampa reja.
- Tapa de filtro de polipropileno.
- Anillo para prevenir goteras.
- Fácil retiro de tapa y rejilla.
- De construcción durable , con componentes anticorrosivos.
- De múltiples usos en los diferentes sistemas de riego.
- La finalidad de estos filtros es la de retener las partículas finas que lleva el agua para evitar las obstrucciones en los goteros.

Especificaciones:

Rejilla (Malla) de color blanco Grado de filtración  
MES H 200 (75 )

### 3. Manómetro.

Medidor de presiones marca RAIN BIRD Modelo  
RBG-L 160DC (Fig. 3.5).

Características:

- Llenado de glicerina para prevenir vibraciones, pulsaciones y otras distorsiones que afecten la lectura.
- Diámetro de carátula de 2 1/2".
- Medidas de 0 -160 psi
- Componentes internos de bronce.
- De estructura acrílica para prevenir impactos, el tiempo y productos químicos

#### 3.3 Programa de control de la red.

El programa de control de la red se lo ha realizado lo más sencillo posible, su manejo no requiere de conocimientos previos de lenguaje C.

El programa consta de un cuerpo principal (riego.c) desde el cual se llama a subrutinas que realizan el *proceso de muestreo* de movimiento de válvulas y la toma de datos de humedad del suelo.

Entre las subrutinas tenemos graphis.h y tubería.h es

La parte que simula la red en la pantalla de la computadora, las subrutinas `stdlib.h` y `string.h` presentan los archivos de control del sistema de riego que se produce a conveniencia del cultivo.

A continuación se presenta el programa base:



```

#include <stdio.h>
#include <conio.h>
#include <conio.h>
#include <stdlib.h>
#include <tuberia.h>
#include <time.h>
#include <dos.h>
#include <unistd.h>
#include <string.h>

char *f1;
char *f2;
char *f3;
char *f4;
char *f5;
char *f6;
char *f7;
char *f8;
char *f9;
char *f10;
char *f11;
char *f12;
char *f13;
char *f14;
char *f15;
char *f16;
char *f17;
char *f18;
char *f19;
char *f20;
char *f21;
char *f22;
char *f23;
char *f24;
char *f25;
char *f26;
char *f27;
char *f28;
char *f29;
char *f30;
char *f31;
char *f32;
char *f33;
char *f34;
char *f35;
char *f36;
char *f37;
char *f38;
char *f39;
char *f40;
char *f41;
char *f42;
char *f43;
char *f44;
char *f45;
char *f46;
char *f47;
char *f48;
char *f49;
char *f50;
char *f51;
char *f52;
char *f53;
char *f54;
char *f55;
char *f56;
char *f57;
char *f58;
char *f59;
char *f60;
char *f61;
char *f62;
char *f63;
char *f64;
char *f65;
char *f66;
char *f67;
char *f68;
char *f69;
char *f70;
char *f71;
char *f72;
char *f73;
char *f74;
char *f75;
char *f76;
char *f77;
char *f78;
char *f79;
char *f80;
char *f81;
char *f82;
char *f83;
char *f84;
char *f85;
char *f86;
char *f87;
char *f88;
char *f89;
char *f90;
char *f91;
char *f92;
char *f93;
char *f94;
char *f95;
char *f96;
char *f97;
char *f98;
char *f99;
char *f100;

int main()
{
    int x,y;
    unsigned char resultado;
    time_t t;
    char *f1;
    char *f2;
    extern char *f3;
    int driver, mode;
    int lec(int locx, int locy);
    void *f4;
    void *f5;
    void *f6;
    void *f7;
    void *f8;
    void *f9;
    void *f10;
    void *f11;
    void *f12;
    void *f13;
    void *f14;
    void *f15;
    void *f16;
    void *f17;
    void *f18;
    void *f19;
    void *f20;
    void *f21;
    void *f22;
    void *f23;
    void *f24;
    void *f25;
    void *f26;
    void *f27;
    void *f28;
    void *f29;
    void *f30;
    void *f31;
    void *f32;
    void *f33;
    void *f34;
    void *f35;
    void *f36;
    void *f37;
    void *f38;
    void *f39;
    void *f40;
    void *f41;
    void *f42;
    void *f43;
    void *f44;
    void *f45;
    void *f46;
    void *f47;
    void *f48;
    void *f49;
    void *f50;
    void *f51;
    void *f52;
    void *f53;
    void *f54;
    void *f55;
    void *f56;
    void *f57;
    void *f58;
    void *f59;
    void *f60;
    void *f61;
    void *f62;
    void *f63;
    void *f64;
    void *f65;
    void *f66;
    void *f67;
    void *f68;
    void *f69;
    void *f70;
    void *f71;
    void *f72;
    void *f73;
    void *f74;
    void *f75;
    void *f76;
    void *f77;
    void *f78;
    void *f79;
    void *f80;
    void *f81;
    void *f82;
    void *f83;
    void *f84;
    void *f85;
    void *f86;
    void *f87;
    void *f88;
    void *f89;
    void *f90;
    void *f91;
    void *f92;
    void *f93;
    void *f94;
    void *f95;
    void *f96;
    void *f97;
    void *f98;
    void *f99;
    void *f100;

    menu(void);

    detectmaph($g_driver,$g_mode);
    initraph($g_driver,$g_mode,"");
    clearmaph();
    use_arch("tuberia.dat");
    mainmenu();
    closemaph();
}

void *mainmenu(void);

char *f1;

ymax = getmaxy();
ymax = getmaxy();
rectangle(0,0,xmax,ymax);
rectangle(0,0,xmax,25);
rectangle(0,0,xmax,ymax-25);
setbkcolor(0);

```

```

setcolor(15);
outtextxy(5,0," ESPOL");
outtextxy(5,max-15,"");
setcolor(6);
outtextxy(5,8," ESPOL");
outtextxy(5,max-15,"");
while(1)

```

```

Setear
150: hauraa o la oaa
2
(DEL)

```

```

    ch = getch();
    ch = tolower(ch);
    if (ch=='e')
        setear();
    if (ch=='r')
        reportar();
    }
    if (ch=='a')
        activar();
    if (ch==27)
        break;

```

```

void far humedad(void)

```

```

    int  entrada[15],salida[15],i,vloc,x1loc,x2loc;
    char  c;
    int  poly[8],sombra1[8],sombra2[8];

```

```

    poly[0] = 75;   sombra1[0] = 561;   sombra2[0] = 85;
    poly[1] = 50;   sombra1[1] = 50;   sombra2[1] = 251;
    poly[2] = 560;  sombra1[2] = 571;  sombra2[2] = 561;
    poly[3] = 50;   sombra1[3] = 60;   sombra2[3] = 251;
    poly[4] = 560;  sombra1[4] = 571;  sombra2[4] = 561;
    poly[5] = 250;  sombra1[5] = 261;  sombra2[5] = 251;
    poly[6] = 75;   sombra1[6] = 561;  sombra2[6] = 85;
    poly[7] = 250;  sombra1[7] = 261;  sombra2[7] = 251;
    ventana(poly,sombra1,sombra2,15,1);
    men_val();

```

```

    i = 1; x1loc = 320; x2loc = 495; vloc = 75;
    while(i<=15)

```

```

        entrada[i] = lee(x1loc,vloc);
        if (entrada[i] == -1)

```

```

            ventana(poly,sombra1,sombra2,3,3);
            break;
        }

```

```

        salida[i] = lee(x1loc,vloc);
        if (salida[i] == -1)

```

```

            ventana(poly,sombra1,sombra2,3,3);
            break;
        }

```

```

        vloc = vloc + 10;
        ***
    }

```

```

while (1)

```

```

    if (entrada[i]==-1 || salida[i]==-1)

```

```

        ventana(poly,sombra1,sombra2,3,3);
        break;
    }

```

```

outtextxy(5,8," ESPOL
outtextxy(5,ymax-15."
setcolor(6);
outtextxy(5,8," ESPOL
outtextxy(5,ymax-15."
while(1)
{
    ch = getch();
    ch = tolower(ch);
    if (ch=='s')
        setear();
    if (ch=='r')
        reporte();
    if (ch=='a')
        activar();
    if (ch==27)
        break;
}
}

void far humedad(void)
{
    int     entrada;15; ,salida;15; ,i ,yloc ,x1loc ,x2loc;
    char    c,n;
    int     poly;8; ,sombra1;8; ,sombra2;8;

    poly;0; = 75;      sombra1;0; = 561;      sombra2;0; = 85;
    poly;1; = 50;      sombra1;1; = 60;      sombra2;1; = 251
    poly;2; = 560;     sombra1;2; = 571;     sombra2;2; = 561
    poly;3; = 50;      sombra1;3; = 60;      sombra2;3; = 251
    poly;4; = 560;     sombra1;4; = 571;     sombra2;4; = 561
    poly;5; = 250;     sombra1;5; = 261;     sombra2;5; = 261
    poly;6; = 75;      sombra1;6; = 561;     sombra2;6; = 85;
    poly;7; = 250;     sombra1;7; = 261;     sombra2;7; = 261
    ventana(poly,sombra1,sombra2,15,1);
    men_val();

    i = 1; x1loc = 320; x2loc = 495; yloc = 75;
    while(i<=15)
    {
        entrada;i; = lee(x1loc,yloc);
        if (entrada;i; == -1)
        {
            ventana(poly,sombra1,sombra2,3,3);
            break;
        }
        salida;i; = lee(x2loc,yloc);
        if (salida;i; == -1)
        {
            ventana(poly,sombra1,sombra2,3,3);
            break;
        }
        yloc = yloc + 10;
        ++i;
    }
    while (1)
    {
        if (entrada;i;== -1 || salida;i;== -1)

```

```

        ventana(poly, sombra1, sombra2, 3, 3);
        break;
    }

    c = getch();
    if (c==0+':')
        graba_arch(entrada, salida, "humedad.dat");
        ventana(poly, sombra1, sombra2, 3, 3);
        break;
    }
    if (c==27)
        ventana(poly, sombra1, sombra2, 3, 3);
        break;
    }
}
}
}

```

```

void far graba_arch(int entrada;15;, int sale;15;, char far *nombre;
FILE *fp,*fopen();
int i;

fp = fopen(nombre,"w");
for (i=1;i<=15;i++)
    fprintf(fp,"%d %d\n",entrada;i,sale;i);
fclose(fp);
}

```

```

void far lee_arch(char far *nombre)
FILE *fp,*fopen();
int i,numero,c;
double resto,band;
char far *letra1="00",far *letra2="00";

fp = fopen(nombre,"r");
band = 1.0;
i = 0;
while (c!=EOF)
    c = fgetc(fp);
    if ((c!=10) && (c!=32))
        letra2 = &c;
        strcat(letra1,letra2,1);
        numero = atoi(letra1);
    }
    if ((c==10) && (c==32))
        resto = (mod(band,2.0));
        if (resto==1.0)
            ++i;
            val_ent;1; = numero;
        }
        if (resto!=1.0)
            val_sal;1; = numero;

```

```

        strcpy( letra2, "NO");
        strcpy( letra1, letra2);
        band = band + 1.0;
    }
}
fclose(fp);
}

void far activar(void)
{
    int color;
    char far *ficherol="/0";

    color = 15;
    setcolor(15);
    tuberia0(color);
    tuberial(color);
    tuberia2(color);
    tuberia3(color);
    ficherol = encabera();

    n = 0;
    c = ' ';
    while(1)
    {
        n = kbhit();
        if (n!=0)
        {
            c = getch();
            if (c==27)
                break;
            if (c=='a')
                resultado = val_enti1;
            if (c=='s')
                resultado = val_enti2;
            if (c=='d')
                resultado = val_enti3;
            if (c=='f')
                resultado = val_enti4;
            if (c=='g')
                resultado = val_enti5;
            if (c=='h')
                resultado = val_enti6;
            if (c=='i')
                resultado = val_enti7;
            if (c=='k')
                resultado = val_enti8;
            if (c=='l')
                resultado = val_enti9;
            if (c=='z')
                resultado = val_enti10;
            if (c=='x')
                resultado = val_enti11;
            if (c=='c')
                resultado = val_enti12;
            if (c=='v')
                resultado = val_enti13;
            if (c=='b')
                resultado = val_enti14;
            if (c=='n')
                resultado = val_enti15;
            n = 0; c = ' ';
        }
    }
}

```

```

time(&t);
setcolor(15);
outtextxy(430,75,ctime(&t));
setcolor(0);
outtextxy(430,75,ctime(&t));
/* resultado = inport(957);*/
decision(resultado.fichero1);
resultado = 0;
if (resultado==127)
    break;
}
}

```

```

void far decision(int resultado,char far *fichero1)

```

```

int          color;
struct date  d;

```

```

getdate(&d);
if (resultado==5255)
{
    mensaje_pan("Apertura de v1");
    color = 1;
    setcolor(1);
    tuberia0(color);
    pone_datos(fichero1,"Apertura de v1");
    outport(956, val_sal12);
}

```

```

if (resultado==63)
{
    mensaje_pan("Apertura de v2");
    color = 1;
    setcolor(1);
    tuberia1(color);
    pone_datos(fichero1,"Apertura de v2");
    outport(956, val_sal22);
}

```

```

if (resultado==95)
{
    mensaje_pan("Apertura de v3");
    color = 1;
    setcolor(1);
    tuberia2(color);
    pone_datos(fichero1,"Apertura de v3");
    outport(956, val_sal32);
}

```

```

if (resultado==111)
{
    mensaje_pan("Apertura de v4");
    color = 1;
    setcolor(1);
    tuberia3(color);
    pone_datos(fichero1,"Apertura de v4");
    outport(956, val_sal42);
}

```

```

if (resultado==131)

    mensaje_pan("Apertura de v1 v2");
    color = 1;
    setcolor(1);
    tuberia0(color);
    tuberia1(color);
    pone_datos(fichero1,"Apertura de v1 v2");
    outport(956, val_sal15);
}

if (resultado==223)

    mensaje_pan("Apertura de v1 v3");
    color = 1;
    setcolor(1);
    tuberia0(color);
    tuberia2(color);
    pone_datos(fichero1,"Apertura de v1 v3");
    outport(956, val_sal16);
}

if (resultado==339)

    mensaje_pan("Apertura de v1 v4");
    color = 1;
    setcolor(1);
    tuberia0(color);
    tuberia3(color);
    pone_datos(fichero1,"Apertura de v1 v4");
    outport(956, val_sal17);
}

if (resultado==31)

    mensaje_pan("Apertura de v2 v3");
    color = 1;
    setcolor(1);
    tuberia1(color);
    tuberia2(color);
    pone_datos(fichero1,"Apertura de v2 v3");
    outport(956, val_sal18);
}

if (resultado==47)

    mensaje_pan("Apertura de v2 v4");
    color = 1;
    setcolor(1);
    tuberia1(color);
    tuberia3(color);
    pone_datos(fichero1,"Apertura de v2 v4");
    outport(956, val_sal19);
}

if (resultado==79)

```

```

    mensaje_pan("Apertura de v3 v4");
    color = 1;
    setcolor(1);
    tuberias2(color);
    tuberias3(color);
    pone_datos(fichero1,"Apertura de v3 v4");
    outport(956, val_sal; 9);
}

if (resultado==159)

    mensaje_pan("Apertura de v1 v2 v3");
    color = 1;
    setcolor(1);
    tuberias0(color);
    tuberias1(color);
    tuberias2(color);
    pone_datos(fichero1,"Apertura de v1 v2 v3");
    outport(956, val_sal; 10);
}

if (resultado==175)

    mensaje_pan("Apertura de v1 v2 v4");
    color = 1;
    setcolor(1);
    tuberias0(color);
    tuberias1(color);
    tuberias3(color);
    pone_datos(fichero1,"Apertura de v1 v2 v4");
    outport(956, val_sal; 11);
}

if (resultado==207)

    mensaje_pan("Apertura de v1 v3 v4");
    color = 1;
    setcolor(1);
    tuberias0(color);
    tuberias2(color);
    tuberias3(color);
    pone_datos(fichero1,"Apertura de v1 v3 v4");
    outport(956, val_sal; 12);
}

if (resultado==15)

    mensaje_pan("Apertura de v2 v3 v4");
    color = 1;
    setcolor(1);
    tuberias1(color);
    tuberias2(color);
    tuberias3(color);
    pone_datos(fichero1,"Apertura de v2 v3 v4");
    outport(956, val_sal; 13);
}

if (resultado==143)

    mensaje_pan("Apertura de v1 v2 v3 v4");
    color = 1;
    setcolor(1);

```

```

    tuberia0(color);
    tuberia1(color);
    tuberia2(color);
    tuberia3(color);
    pone_datos(fichero1,"Apertura de v1 v2 v3 v4");
    outport(956, val_sal;14);
}

if (resultado==127)
{
    mensate_pan("fin");
    pone_datos(fichero1,"fin");
    outport(956, val_sal;15);
}
resultado = 0;
}

void far_setear(void)
{
    char    c;
    int     poly;8;,sombra1;8;,sombra2;8;

    poly;0; = 140;    sombra1;0; = 281;    sombra2;0; = 150;
    poly;1; = 27;    sombra1;1; = 35;    sombra2;1; = 131;
    poly;2; = 280;   sombra1;2; = 291;   sombra2;2; = 281;
    poly;3; = 27;    sombra1;3; = 35;    sombra2;3; = 131;
    poly;4; = 280;   sombra1;4; = 291;   sombra2;4; = 281;
    poly;5; = 130;   sombra1;5; = 141;   sombra2;5; = 141;
    poly;6; = 140;   sombra1;6; = 281;   sombra2;6; = 150;
    poly;7; = 130;   sombra1;7; = 141;   sombra2;7; = 131;
    while(1)
    {
        ventana(poly,sombra1,sombra2,15,1);
        men_ventana(145);
        c = getch();
        c = tolower(c);
        if (c==27)
        {
            ventana(poly,sombra1,sombra2,3,3);
            break;
        }
        if (c=='p')
        {
            break;
        }
        if (c=='l')
        {
            break;
        }
        if (c=='v')
        {
            break;
        }
        if (c=='a')
        {
            break;
        }
        if (c=='t')
        {
            break;
        }
    }
}

```

```

    if (c=='r')
        break;
    }
    if (c=='h')
        ventana(poly,sombra1,sombra2,3,3);
        humedad();
    }
}
}

```

```

void far_reporte(void)

```

```

    char    c;
    far #nombre="80";
    int     tam,x,y;
    int     poly[8],sombra1[8],sombra2[8];

```

```

poly[0] = 330;    sombra1[0] = 471;    sombra2[0] = 340;
poly[1] = 27;    sombra1[1] = 35;     sombra2[1] = 131;
poly[2] = 470;  sombra1[2] = 481;    sombra2[2] = 481;
poly[3] = 27;    sombra1[3] = 35;     sombra2[3] = 131;
poly[4] = 470;  sombra1[4] = 481;    sombra2[4] = 481;
poly[5] = 130;  sombra1[5] = 141;    sombra2[5] = 141;
poly[6] = 330;  sombra1[6] = 471;    sombra2[6] = 340;
poly[7] = 130;  sombra1[7] = 141;    sombra2[7] = 141;

```

```

ventana(poly,sombra1,sombra2,15,1);

```

```

men_ventana(335);

```

```

while(1)

```

```

    strcpy(nombre,"80");

```

```

    c = getch();

```

```

    c = tolower(c);

```

```

    if (c==27)

```

```

        ventana(poly,sombra1,sombra2,3,3);

```

```

        break;

```

```

    }

```

```

    if (c=='p')

```

```

        break;

```

```

    }

```

```

    if (c=='l')

```

```

        break;

```

```

    }

```

```

    if (c=='v')

```

```

        break;

```

```

    }

```

```

    if (c=='a')

```

```

        break;

```

```

    }

```

```

    if (c=='t')

```

```

        break;

```

```

    }

```

```

    if (c=='r')

```

```

        break;

```

```

    }

```

```

    if (c=='e')

```

```

        break;

```

```

    }

```

```

    if (c=='r')

```

```

        break;

```

```

    }

```

## CAPITULO IV

### EVALUACION EXPERIMENTAL DE LA RED

#### 4.1 Pruebas parciales.

Como se puede observar en el capítulo 3.4 en el que se presentó el programa de control de riego, las pruebas parciales consisten en determinar la efectividad del programa en el sistema de riego en las parcelas cultivadas de melón, tomate, pimiento y mango.

La simulación efectuada por computadora nos permite observar el desarrollo al introducir primeramente los parámetros de apertura de las válvulas.

#### 4.2 Análisis integral del proceso.

Con un sistema de control automatizado se busca la optimización y aprovechamiento controlado de los recursos hídricos, a la vez que se mejora la producción al tenerse el mayor control en las cosechas.

Aunque el proceso de automatización de la red de

riego en las parcelas trata de mantener un proceso físico-químico constante en la toma de agua por las plantas, esta absorción de agua por las raíces es muy pequeña o nula, a consecuencia precisamente del elevado grado de saturación de todos los tejidos de la planta.

Se sabe que la frecuencia y el volumen de los riegos se determinan por el consumo o evaporación de agua (evapotranspiración) del campo cultivado (parcelas), y esto naturalmente varía según el tamaño de las plantas, el número y tamaño de las hojas, la transpiración por unidad de área foliar, etc.

Un campo cubierto por un cultivo arbóreo (árboles) evapora naturalmente más agua que el campo con plantas como maíz, frijoles, tomate, pimiento, etc., cultivadas en la misma zona; los primeros por supuesto requieren de más agua.

Sin embargo se tiene que tomar en cuenta que el consumo de agua está subordinado principalmente a los factores meteorológicos.

Este proceso de riego consiste principalmente en un programa de control de la red de riego, la tarjeta

transductora, válvulas solenoides, los goteros y el área de cultivo (plantas), la interacción de cada una de estos elementos permitirá la correcta producción de las parcelas o cultivos.

Al ser determinada la cantidad de agua requerida por el cultivo y el tiempo de riego, en el programa de control se procederá a designar la secuencia de encendido de las válvulas solenoides, estas realizarán las aperturas y comenzarán el riego en el conjunto de parcelas designadas para el riego.

## CAPITULO V

### CONCLUSIONES Y RECOMENDACIONES

#### CONCLUSIONES.

- El sistema como está concebido presenta múltiples etapas de enseñanza didáctica, desde el mismo punto del programa de riego, las válvulas solenoides de control y la tarjeta de control.
- Al ser la computadora personal (PC) de fácil adquisición para cualquier persona en la actualidad que desee utilizar sus servicios, este instrumento se vuelve óptimo para este tipo de trabajo experimental.
- Aunque la inversión inicial de un proceso de riego automatizado es alto, esto no debe impedir su avance ya que esto permite obtener productos de mejor calidad y producción, lo cual mejora los precios en el mercado haciendo que la inversión inicial se pueda cubrir fácilmente.
- Con el control de riego de las parcelas se mejora el control de malezas e insectos que atacan los cultivos

y los vuelven improductivos.

- Aunque el diseño del programador de control es realizado por un programador especializado, este tipo de programas son de fácil manejo y comprensión.
- Al estar incorporado al programa una sección de reportes de riego, dicho control es automático con lo cual se puede observar fallas o correcciones que se deban hacer al sistema.

#### RECOMENDACIONES.

- El computador personal (PC), las válvulas solenoides deben ser de fácil manejo, utilización e instalación.
- Se requiere de mayor experimentación en parcelas de mayor tamaño para conocer las posibles limitaciones del programa o de la instalación del computador personal y todas las conexiones eléctricas.
- Debe existir una variación de cultivos para obtener diferenciaciones en los tiempo de riego y poder observar el comportamiento del programa.

- Existe la posibilidad de programar a medio o largo plazo, aprovechando las características de constancia y uniformidad de producción en los cultivos.
- Este tipo de programas de control deben de existir en otros lenguajes, para determinar cual es el mas óptimo en este tipo de trabajo.

BIBLOGRAFIA

1. Ramiro Aguirre Aguirre, Problemas de riego en el Ecuador y posibles soluciones. (Universidad Nacional de Loja, 1987)
2. Enrique Blair F., Manual de riegos y avenimientos. (Instituto Americano de Ciencias Agricolas).
3. Agricultural Irrigation Equipment RAIN-BIRD
4. Turf Irrigation Equipment RAIN-BIRD  
1991-1992.
5. Keith Shepersky, Landscape Irrigation Design Manual  
RAIN-BIRD.
6. Lenguaje C. Manual de bolsillo.
7. Harrison-Ballinger, Controles Automáticos.  
Editorial Trillas. 5da. Edición 1983