

ÍNDICE

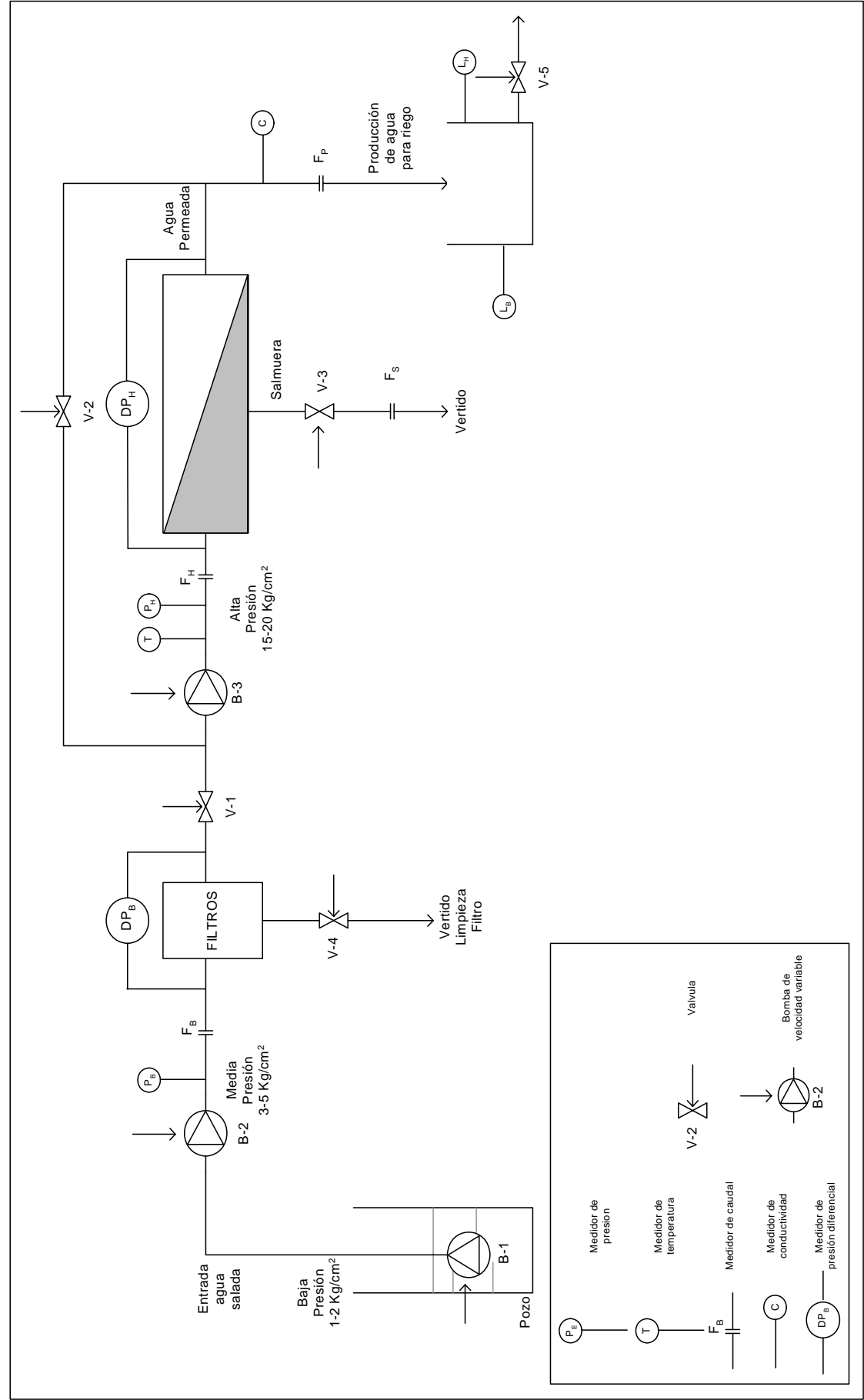
I.- Descripción de la planta elegida.....	2
1.1.- Modo producción de agua.....	4
1.2.- Modo limpieza de filtros.....	5
1.3.- Modo limpieza de membrana.....	5
II.- Modelado de la planta elegida.....	6
III.- Programación del microcontrolador.....	11
IV.- Carga del Código.....	18
V.- Pruebas y Ensayos.....	18

I.- DESCRIPCIÓN DE LA PLANTA ELEGIDA

Una técnica de desalación bien conocida es la ósmosis inversa. La ósmosis inversa se fundamenta en la inversión del proceso natural y espontáneo conocido como ósmosis. Este proceso natural se basa en la redistribución de dos disoluciones de distinta concentración y separadas por una membrana con el objetivo de igualarlas. De esta manera, el agua atraviesa la membrana y las dos disoluciones alcanzan el equilibrio osmótico. A la diferencia de presión de las dos disoluciones se le denomina presión osmótica. Al aplicar una presión mayor a la osmótica en la disolución más concentrada (agua salada) se consigue invertir el proceso obteniéndose un permeado (agua prácticamente pura) y salmuera.

Para nuestro caso hemos cogido una planta convencional con dos etapas de filtrado, control de conductividad de agua a la salida y una serie de bombas que regulan la presión en las distintas etapas de la instalación.

La figura muestra el diagrama de la instalación. Como se puede observar el agua salada, extraída de un pozo, es desalada por medio de ósmosis inversa obteniéndose, por un lado la producción de agua que es almacena de un deposito y por otro la salmuera que constituye el residuo de la instalación.



Modo producción de agua.

El modo de funcionamiento *producción de agua* consiste en la obtención de agua desalada.

La condición de arranque y parada de la instalación bien determinada por la medición del nivel en el depósito de almacenamiento, de manera que si el nivel es menor que un determinado nivel (L_B) la instalación comienza a funcionar, cuando se alcanza un determinado nivel (L_H) la instalación se detiene.

El agua salada es extraída de un pozo por medio de la bomba B-1 la cual la inyecta el agua en la instalación a baja presión ($1-2 \text{ Kg/cm}^2$) esta presión es medida por medio de P_E .

Por medio de la bomba B-2 el se eleva la presión del agua ($3-5 \text{ Kg/cm}^2$) con el fin de poder ser filtrada, esta presión es medida por P_B , si la presión media defiere de necesaria se actuará sobre la bomba B-2 para corregir las desviación

Por último la presión del agua es elevada por medio de la bomba B-3 ($15-20 \text{ Kg/cm}^2$), esta presión se mide por P_H , para poder se introducida en la membrana de ósmosis, de igual manera que anteriormente la presión medida en P_H , determinará las acciones de control sobre la bomba B-3. En dicha membrana se obtiene por una parte el vertido de salmuera, que es desechado; y por otra agua permeada dicha producción de agua es mezclada con cierta agua salada, con el fin de obtener una cierta concentración de sales en el agua, este agua ya tratada es almacenada en un deposito para su posterior uso.

Como ya se ha indicado la producción consiste en obtener agua que presente una cierta concentración de sales. Este agua se obtiene mezclando agua salada con agua permeada.

Para la obtención de agua permeada es necesario mantener los valores de presión y caudal a la entrada de la membrana (P_H y F_H respectivamente) en unos determinados valores, calculados previamente; estos valores dependen de la temperatura (que puede ser media por medio de sensor T), por lo tanto cuando la temperatura del agua varíe habrá que cambiar los valores de referencia para la P_H y F_H .

La concentración de sales de agua se mide por medio de un sensor de conductividad (C), el control de la cantidad de agua salada que se añade a

la producción se realiza con la válvula V-2, dependiendo del valor medido en C, se corregirá la apertura de la válvula.

Por último, por medio de F_P y F_S se controlan los caudales de salida de la producción de agua y de vertido de salmuera respectivamente.

Modo limpieza de filtros

El modo de funcionamiento *limpieza de filtros* pretende limpiar los filtros cuando se detecte que están atorados.

La condición de entrada en este modo viene determinada por la diferencia de presión entre la entrada y la salida del filtro, medida por DP_B , así si el valor medido esta por debajo de valor permitido se entra en este modo de trabajo.

Para limpiar los filtros se procede de la siguiente manera:

1. Parada de la bomba B-3.
2. Apertura de la válvula V-4.
3. Cierre la válvula V-1.

De esta manera se inyecta agua en los filtros y es expulsada junto con las impurezas que estuvieran obstruyendo los filtros.

Pasado un cierto tiempo se vuelve al modo *producción de agua*, realizando los pasos anteriores en orden inverso.

Modo limpieza de membrana

El modo de funcionamiento *limpieza de membrana* se procede a la limpieza de la membrana de ósmosis.

La condición de entrada en este modo viene determinada por la diferencia de presión entre la entrada y la salida de la membrana, medida por DP_H , así si el valor medido esta por debajo de valor permitido se entra en este modo de trabajo.

Para limpiar la membrana se abre la válvula V-3, de manera que el agua es expulsada al vertido, limpiando la membrana.

Pasado un cierto tiempo se vuelve al modo *producción de agua*, cerrando la válvula V-3.

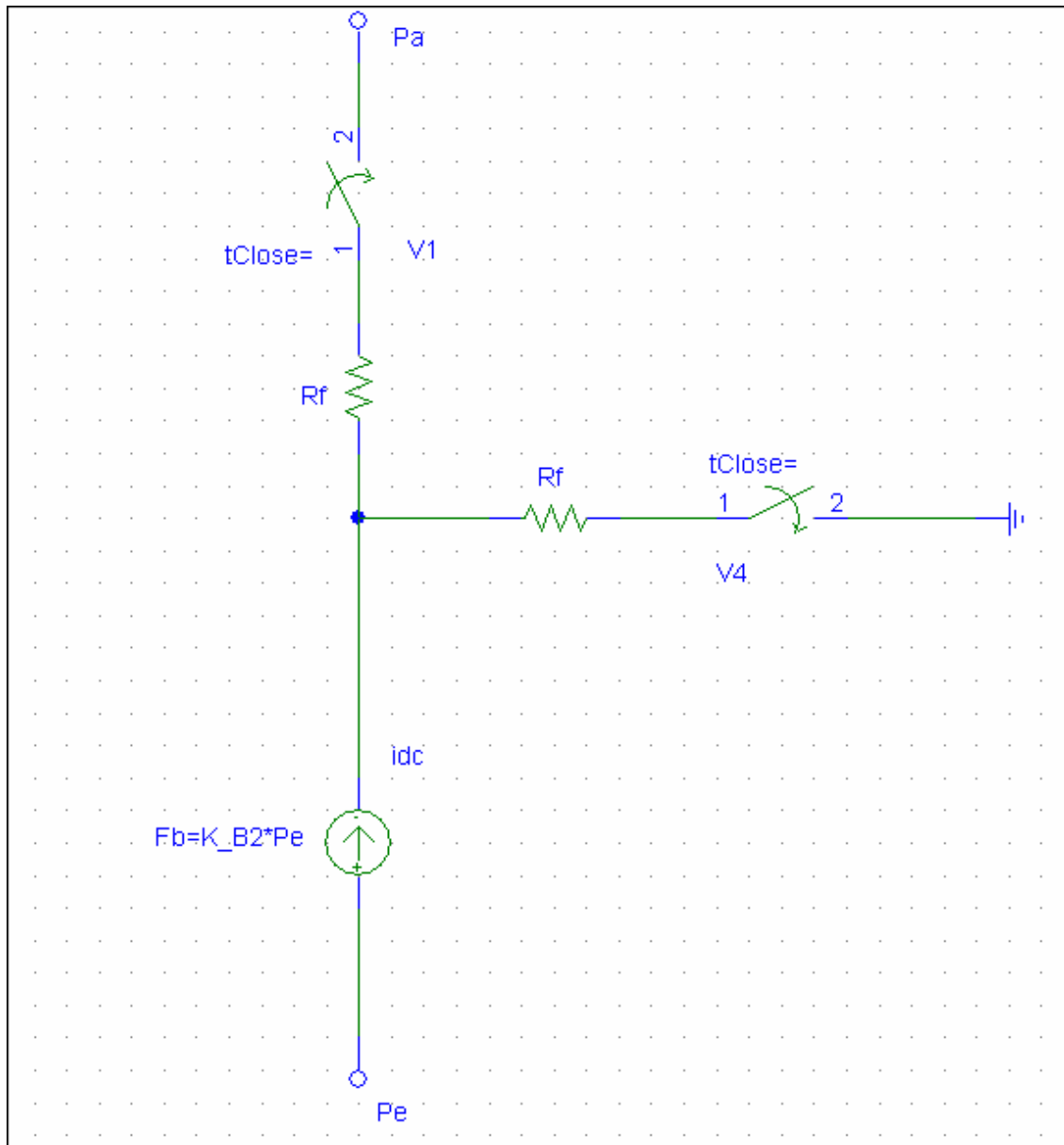
II.- MODELADO DE LA PLANTA

Para realizar el análisis de la planta usaremos un símil eléctrico, para resolver el modelo eléctrico, se realizará un análisis multietapa, para la primera etapa consideraremos que a su entrada tendremos una presión constante P_e .

B2 se considerará una bomba de caudal variable, dependiente de la presión de su entrada, para el modelo eléctrico lo consideraré como una fuente de intensidad controlada por tensión.

V1 y V4 los consideraré como interruptores enlazados (cuando uno está abierto el otro estará cerrado) controlados por DPb.

El filtro lo consideraré como una resistencia.



K_{B2} , $V4$ y $V1$ son las variables que se recibirán de los reguladores para gobernar la planta.

Una vez que se alcance el régimen permanente la presión P_a se mantendrá constante.

Con esto las ecuaciones de la primera etapa en modo normal serán:

$$F_b = K_{B2} * P_e \quad \text{con } P_e = cte$$

$$P_b = F_b * R_f + P_a \quad \text{con } P_a = cte$$

$$DPB = F_b * R_f$$

$R_f = R_f + 1$ que aumente la resistencia simula que se ensucian los filtros.

Las ecuaciones de la primera etapa en modo limpieza de filtros serán:

$R_f = R_{f-1}$ Esto simula la limpieza del filtro.

$P_a = 0$

$F_b = K_{B2} * P_e$ con $P_e = cte$

$P_b = F_b * R_f + P_a$ con $P_a = cte$

$DPB = F_b * R_f$

En modo limpieza de membrana la primera etapa estará en modo normal, pero para la simulación no es necesario considerarla porque el análisis multietapa nos permite usar una presión constante P_a de forma que no influye nada de lo que haya antes de ese punto.

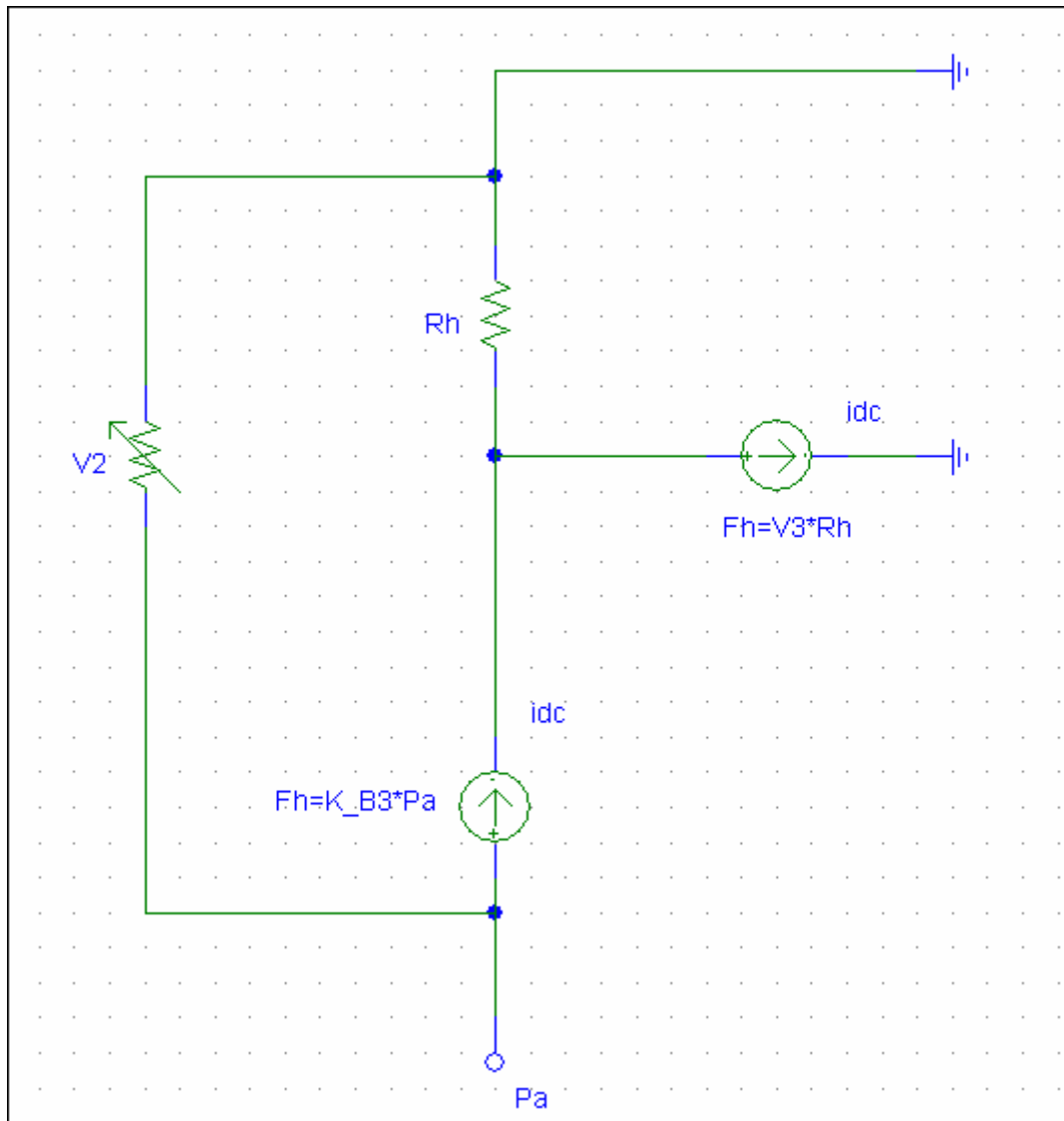
En la segunda parte de la planta haremos un análisis similar al anterior.

B3 se analizará como B2

V3 será una válvula gradual que modelaré como una fuente de intensidad.

V2 será una resistencia variable con la conductividad.

La membrana será una resistencia



La bomba B3 se simulará como una fuente de intensidad controlada por la variable K_{B3} que es la que da el regulador pertinente a su salida.

En la válvula $v3$ también es necesario considerar la resistencia debida a la membrana y $V2$ es una válvula que depende de la conductividad del agua a la salida.

Con esto las ecuaciones en modo normal de esta etapa serán:

$$Fh = K_{B3} * Pa$$

$$Ph = Fh * Rh$$

$$Fs = V3 * Rh$$

$$Fc = V2$$

$$Rh = Rh + 1 \text{ esto simula que se ensucia la membrana.}$$

$$Fp = Fc + Fh - Fs$$

En modo limpieza de membrana las ecuaciones de funcionamiento serán:

$Rh = Rh - 1$ simula la limpieza de la membrana.

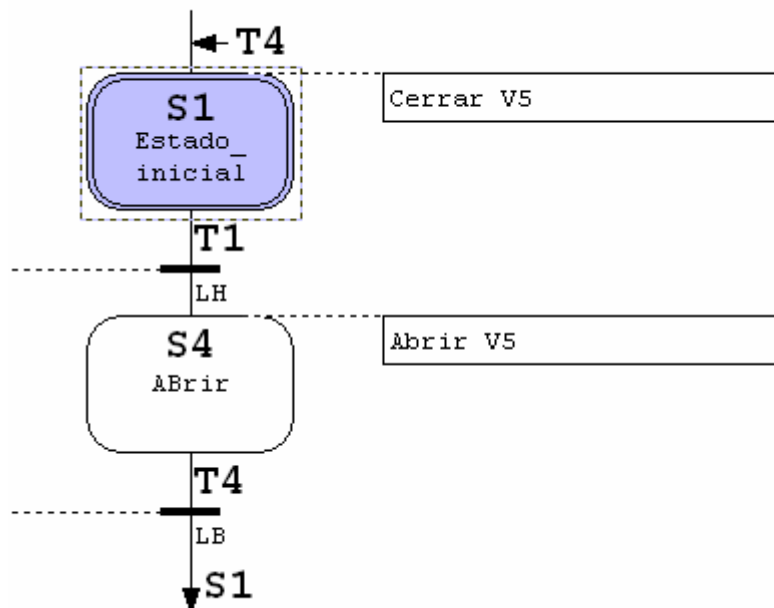
$Fh = K_{B3} * Pa$

$Ph = Fh * Rh$

En modo limpieza de filtros no influye la segunda parte del modelo porque se hace $Pa = 0$ esto implica que esta parte de la planta está completamente desactiva.

En cualquier caso los reguladores darán una serie de variables a su salida que al capturarlos la planta esta elige el modo de funcionamiento.

Como el funcionamiento del depósito de agua permeada es cuasi-independiente podremos modelarlo como un SFC (Sequential Function Chart) un caudal variable de entrada y una señal de salida (V5) que dependerá del nivel del depósito.



Para el correcto funcionamiento de la planta debemos suponer que V5 es capaz de evacuar más agua que la que suministra la planta.

No obstante esto es mas un modelo de control que de simulado porque los reguladores serán quienes nos digan cuando debemos abrir o cerrar las válvulas y en función de esto actuará la planta.

Con esto la planta quedará totalmente modelada para a partir de aquí realizar un programa que se cargará en el microcontrolador para simular su funcionamiento.

III.- PROGRAMACIÓN DEL MICROCONTROLADOR

En la programación del controlador lo primero que hay que tener en cuenta es que tenemos que configurar los puertos y el reloj de este, después el programa principal será un bucle infinito que en primer lugar leerá los datos que le darán los reguladores mediante el protocolo HandShake, luego en función de esa lectura entrará en un modo u otro de funcionamiento y por último se le mandarán los datos a los reguladores con el mismo protocolo.

Con esto el programa que resulta quedará de la siguiente manera:

```
#include <c8051f320.h>
#define SYS_int_OSC 0

int K_B2 = 1; //Salida regulador 1
int K_B3 = 1; //Salida regulador 2

/* Variables de control para el protocolo HandShake para la recepcion de datos
*/

sbit udata = P2^0; //entrada
sbit rdata = P2^1; //salida

/* Variables de control para el protocolo HandShake para el envio de datos */

sbit dato_leido = P2^2; //entrada
sbit dato_listo = P2^3; //salida
sbit led1 = P2^4;
sbit led2 = P2^5;

/* Variables de control de las válvulas */

int V1;
int V2;
int V3;
int V4;
int V5;

/* Variables de las resistencias de los filtros y de la valvula de conductividad
inicializadas a 100 por defecto */
```

```
int Rf = 10;
int Rh = 10;
```

```
/* Resto de variables de la planta para calcular */
```

```
int Pa, Pe; //deben ser inicializadas
int Pb, Fb, Ph, Fs, Fc, Fp, C, Fh, DPB, Lh, Lb;
```

```
/* Declaración de funciones */
```

```
void port_init (void);
void SYS_CLK_init (void);
void modo_normal(void);
void modo_limpieza_filtro(void);
void modo_limpieza_membrana(void);
void leer_datos (void);
void enviar_datos (void);
```

```
/* Función principal */
```

```
void main (void)
{
    PCA0MD &= ~0x40; //Se desactiva el Watch dog
    port_init();
    SYS_CLK_init();

    Pe = 5;
    Pa = 5;
    C=2;
    while (1)
    {
        leer_datos();
        led1 = 1;
        led2 = 1;
        if ( V1 == 0xFF && V4 == 0 )
        {
            modo_limpieza_filtro();
        }

        else if ( V3 == 0xFF && V2 == 0 )
        {
            modo_limpieza_membrana();
        }

        else
        {
            modo_normal();
        }
    }
}
```

```
        enviar_datos ();
    }

}

/* Funcion para leer los datos provenientes de los reguladores */

void leer_datos (void)
{
    led1= 1;
    led2= 0;
    //Leer K_B2
    while(udata != 1);
    if (udata == 1)
    {
        K_B2 = P0;
        rdata = 1;
    }
    while(udata != 0);

    if (udata == 0)
        rdata = 0;

    //Leer K_B3
    while(udata != 1);
    if (udata == 1)
    {
        K_B3 = P0;
        rdata = 1;
    }
    while(udata != 0);

    if (udata == 0)
        rdata = 0;

    //Leer V3
    while(udata != 1);
    if (udata == 1)
    {
        V3 = P0;
        rdata = 1;
    }
    while(udata != 0);

    if (udata == 0)
        rdata = 0;

    //Leer V2
    while(udata != 1);
```

```
    if (udata == 1)
    {
        V2 = P0;
        rdata = 1;
    }
    while(udata != 0);

    if (udata == 0)
        rdata = 0;

    //Leer V1
    while(udata != 1);
    if (udata == 1)
    {
        V1 = P0;
        rdata = 1;
    }
    while(udata != 0);

    if (udata == 0)
        rdata = 0;

    //Leer V4
    while(udata != 1);
    if (udata == 1)
    {
        V4 = P0;
        rdata = 1;
    }
    while(udata != 0);

    if (udata == 0)
        rdata = 0;

    //Leer V5
    while(udata != 1);
    if (udata == 1)
    {
        V5 = P0;
        rdata = 1;
    }
    while(udata != 0);

    if (udata == 0)
        rdata = 0;

}
```

```
/* Funcion que simula la limpieza de los filtros */
```

```
void modo_limpieza_filtro (void)
```

```
{
    Rf = Rf - 1;
    Pa=0;
    Fb = K_B2*Pe; // Pe = cte
    Pb = Fb*Rf + Pa; // Pa = cte
    DPB = Fb*Rf;

}
```

```
/* Funcion que simula la limpieza de la membrana */
```

```
void modo_limpieza_membrana (void)
```

```
{
    Rh = Rh - 1;
    Fh = K_B3*Pa;
    Ph = Fh*Rh;
}
```

```
/* Funcion que simula el modo normal de funcionamiento de la planta */
```

```
void modo_normal (void)
```

```
{
    Fb = K_B2*Pe; // Pe = cte
    Pb = Fb*Rf + Pa; // Pa = cte
    DPB = Fb*Rf;
    Rf = Rf + 1;
    Fh = K_B3*Pa;
    Ph = Fh*Rh;
    Fs = V3*Rh;
    Fc = V2;
    Rh = Rh + 1;
    Fp = Fc+Fh-Fs;
    //Como C no es facil modelarlo electricamente
    //lo tomare como una señal aleatoria
    if (C != 1)
    {
        C++;
    }
    if (C == 100)
    {
        C = 2;
    }
}
```

```
/* Funcion para enviar los datos que calculo */
```

```
void enviar_datos (void)
{
    led1 = 0;
    led2 = 1;

    //Envio de Pb
    P1 = Pb;
    dato_listo = 1;
    while (dato_leido == 0);
    dato_listo = 0;
    while (dato_leido != 0);

    //Envio de Fb
    P1 = Fb;
    dato_listo = 1;
    while (dato_leido == 0);
    dato_listo = 0;
    while (dato_leido != 0);

    //Envio de DPB
    P1 = DPB;
    dato_listo = 1;
    while (dato_leido != 1);
    dato_listo = 0;
    while (dato_leido == 1);

    //Envio de Ph
    P1 = Ph;
    dato_listo = 1;
    while (dato_leido != 1);
    dato_listo = 0;
    while (dato_leido == 1);

    //Envio de Fh
    P1 = Fh;
    dato_listo = 1;
    while (dato_leido != 1);
    dato_listo = 0;
    while (dato_leido == 1);

    //Envio de Fs
    P1 = Fs;
    dato_listo = 1;
    while (dato_leido != 1);
    dato_listo = 0;
    while (dato_leido == 1);

    //Envio de Fp
```



```
P1 = Fp;
dato_listo = 1;
while (dato_leido != 1);
dato_listo = 0;
while (dato_leido == 1);

//Envio de Lh
P1 = Lh;
dato_listo = 1;
while (dato_leido != 1);
dato_listo = 0;
while (dato_leido == 1);

//Envio de Lb
P1 = Lb;
dato_listo = 1;
while (dato_leido != 1);
dato_listo = 0;
while (dato_leido == 1);

//Envio de C
P1 = C;
dato_listo = 1;
while (dato_leido != 1);
dato_listo = 0;
while (dato_leido == 1);

}

/* Funcion para configurar los puertos */

void port_init ()
{
    P0MDIN = 0xFF; // entradas digitales
    P2MDIN = 0xFF; // entradas digitales
    P1MDOUT = 0xFF; //Salidas push pull
    P2MDOUT = 0xFF; // Se configura en push Pull
    P0SKIP = 0xFF; // Dejo libre en el crossbar los pines que necesito
    P1SKIP = 0xFF; // Dejo libre en el crossbar los pines que necesito
    P2SKIP = 0xFF; // Dejo libre en el crossbar los pines que necesito
    XBR0 = 0x00;
    XBR1 = 0x40; // Inicializamos solo el crossbar
}

/* Funcion para configurar el reloj */

void SYS_CLK_init (void)
{
```

```
OSCICN |= 0x83; // configura el reloj a 12 Mhz  
CLKSEL |= SYS_int_OSC;  
}
```

Con esto quedaría el programa funcionando, este debe de ser el código que se cargue en el microcontrolador para que comience a andar.

Al realizar esta carga el micro estará listo para comportarse a ojos de los reguladores como la planta desalinizadora, de forma que estos interactúen con ella como si fuera la planta, así se podrán probar nuevas técnicas de control sin poner en riesgo la planta.

IV.- CARGA DEL CÓDIGO

Para cargar el código en primer lugar instalaremos los drivers del cable USB-RS232 esto sólo tendremos que hacerlo la primera vez, tras esto ya podremos conectar el cable.

El compilador que usaremos será el propio del microcontrolador, el Cygnal IDE v1.85.

El siguiente paso es conectar la placa, este paso se realizará con el software del compilador abierto, para conectar la placa primero hay que poner el conversor de RS232 a JTAG estas líneas JTAG son las que se conectarán a la placa, posteriormente se alimentará la placa con un cable USB de impresora estándar, si el PC no reconoce la placa en el compilador deberemos darle a “conectar” esto solucionará los problemas.

V.- PRUEBAS Y ENSAYOS

Una vez realizado todo el proceso de diseño y puesta en marcha se procede a la verificación del producto.

Para observar el correcto funcionamiento hay que hacer incapié en dos conceptos fundamentales, por un lado se debe comprobar que el protocolo de comunicación actúa correctamente. El otro punto clave es la parte de cálculo, siendo esta la parte más importante del control de la planta, ya que de ella dependerá en gran medida el correcto funcionamiento de los controladores que en la placa se prueben en un futuro.

Para abarcar estos dos conceptos de verificación se ha ideado una forma de verificación consistente en establecer unos valores constantes a la entrada de forma que se fuerce la planta a estar en un modo de funcionamiento concreto y a partir de ahí leer las señales que genera la planta.

Con esta filosofía se procede a realizar los pertinentes ensayos, el primero de ellos será colocar la planta en modo limpieza de filtros, para ello introducimos en el programa los correspondientes valores de las válvulas.

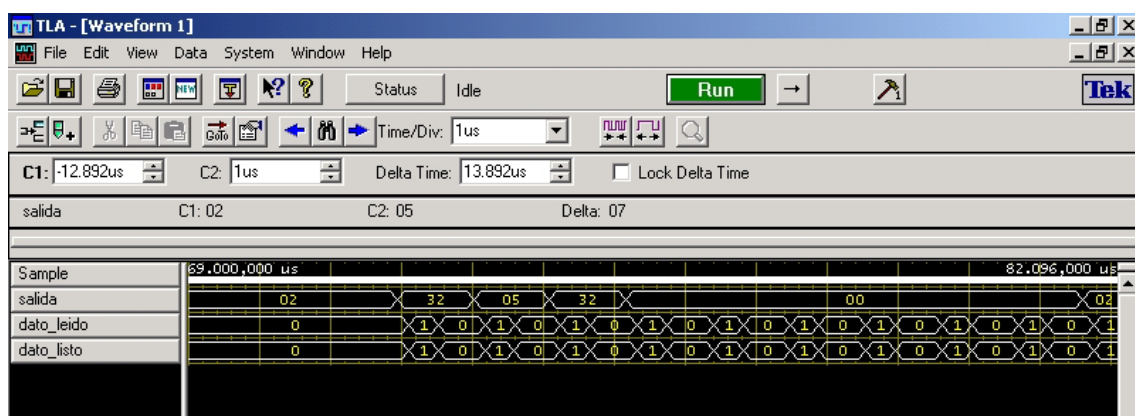
Cabe mencionar que se eliminan las líneas que simulan la suciedad de los filtros, ya que lo que se intenta demostrar es que funciona el modelo y el protocolo.

Con esto la parte de cálculo quedaría:

```
void modo_limpieza_filtro (void)
{
    //Rf = Rf - 1;
    Pa=0;
    Fb = K_B2*Pe; // Pe = cte
    Pb = Fb*Rf + Pa; // Pa = cte
    DPB = Fb*Rf;
}
```

Como $K_{B2} = 0$ y $Pe = 5 \Rightarrow Fb = 5$, además como $Rf = 50$ se comprueba que $Pb = 50$ y $DPB = 50$.

Al visualizar las señales con el analizador lógico:



Como podemos comprobar en la variable de salida primero representa en hexadecimal Pb, después DPB y por último Fb.

Los demás valores se mantienen a cero porque esa parte de la planta no tiene sentido en el modo limpieza de filtros.

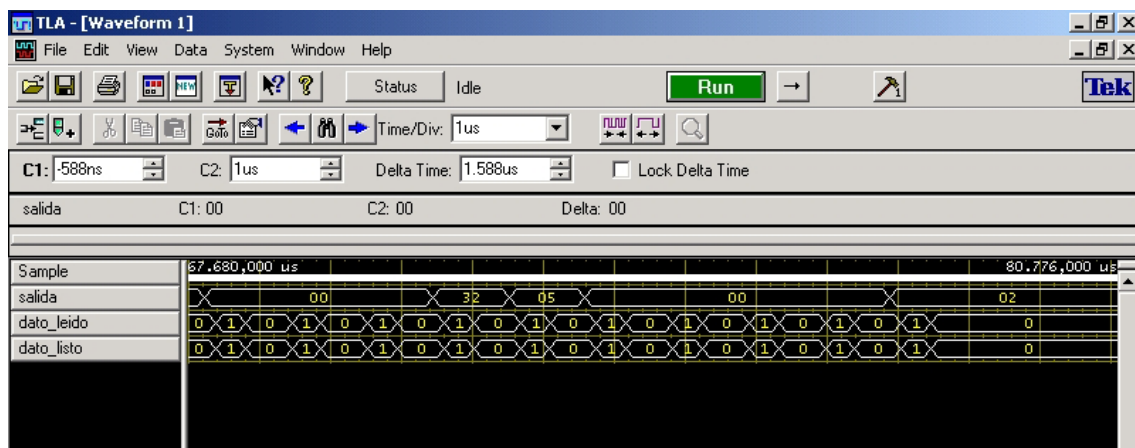
La siguiente comprobación se realiza forzando a la planta para que se sitúe en modo limpieza de membranas, para ello se colocan las válvulas como es necesario, igualmente se elimina la línea que simula la suciedad de los filtros.

El código de esta parte quedará:

```
void modo_limpieza_membrana (void)
{
    //Rh = Rh - 1;
    Fh = K_B3*Pa;
    Ph = Fh*Rh;
}
```

Como $Pa = 5$ y $K_{B3} = 1 \Rightarrow Fh = 5$, igualmente $Ph = 50$

Como comprobamos en el analizador lógico se pueden leer en hexadecimal estos valores, igualmente podemos comprobar que las variables que no influyen a este modo de trabajo se ignoran al enviar este paquete de datos.



Tras esta comprobación se procede a realizar la última comprobación consistente en colocar la planta en modo normal bajo las mismas condiciones de las anteriores pruebas, con esto el código de esa parte quedaría:

```
void modo_normal (void)
{
```

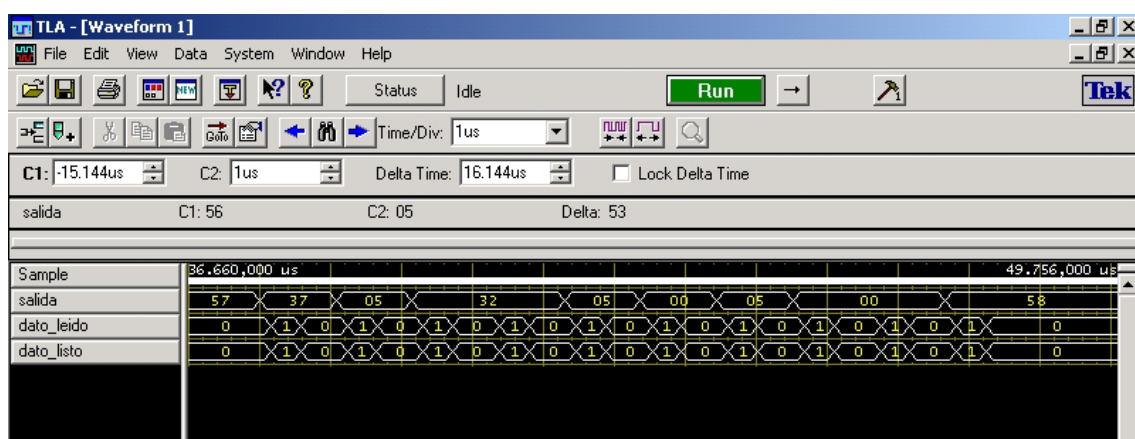
```

Fb = K_B2*Pe; // Pe = cte
Pb = Fb*Rf + Pa; // Pa = cte
DPB = Fb*Rf;
//Rf = Rf + 1;
Fh = K_B3*Pa;
Ph = Fh*Rh;
Fs = V3*Rh;
Fc = V2;
//Rh = Rh + 1;
Fp = Fc+Fh-Fs;
//Como C no es facil modelarlo electricamente
//lo tomare como una señal aleatoria
if (C != 1)
{
    C++;
}
if (C == 100)
{
    C = 2;
}
}

```

Con los valores propuestos quedan los siguientes resultados: $P_b=55$, $F_b=5$, $DPB=50$, $F_h=5$, $P_h=50$, $F_s=0$, $F_c=0$, $F_p=5$, y por último el parámetro C variará aumentando progresivamente para después colocarse en el valor inicial.

Como podemos comprobar el analizador lógico recoge perfectamente estos valores expresados en hexadecimal:



Con estos gráficos queda completamente demostrado que la planta funciona correctamente, ante este panorama lo único que restaría sería colocar los verdaderos controladores y establecer su punto de funcionamiento sin desperdiciar ni una sóla gota de agua.