



# UNIVERSIDAD NACIONAL DE INGENIERÍA

---




## FACULTAD DE INGENIERIA MECÁNICA

*Procesadores digitales de señales (MT418 - C)*

### INFORME DE LABORATORIO N°1

*Elaborado por:*

GRUPO N°2

Apellidos y nombres	Código	Firma
Altamirano Lazo Olenka	20184078B	
De la Cruz Ventura, Alexander	20170266F	
Montes Chuñocca, Andy	20181032A	

*Revisado por:*

ING. DANIEL LEONARDO BARRERA ESPARTA

*Periodo Académico: 2022-1*

LIMA - PERÚ

# Índice

1. Introducción	3
2. Objetivos	3
3. Marco Teórico	3
4. Enunciado del problema	5
5. Diagrama de flujo	6
6. Código en Code Composer	7
7. Resultados	9
8. Conclusiones	10
9. Recomendaciones	10

## Índice de figuras

1.	Diagrama de bloques de un pin físico del GPIO F2833x . . . . .	4
2.	Agrupación GPIO F2833x . . . . .	5
3.	Diagrama del enunciado . . . . .	5
4.	Diagrama de Flujo General . . . . .	6
5.	Circuito de las entradas y salidas digitales con el DSP TMS320F28335 . . .	9
6.	Inputs del circuito etiquetados P,DL,DV,SP1,SP2 . . . . .	10
7.	Outputs del circuito etiquetados V1,V2,V3,VA,VB,M . . . . .	10

# 1. Introducción

En el presente informe de laboratorio se presenta la solución detallada de un problema real de control, analizaremos el proceso de mezcla de productos en un deposito de agua y propondremos una solución óptima.

En las primeras páginas se detallan los objetivos y el marco teórico necesario sobre los procesadores digitales de señales (DSP) para el desarrollo del laboratorio.

A continuación se muestra la descripción del problema y su diagrama de flujo con los pasos a seguir para la solución de este, analizando las variables a tomar en cuenta para la elaboración del código en el software Code Composer Studio.

Finalmente se presentan los resultados obtenidos en la experiencia, detallando las observaciones presentes en la implementación de nuestra solución.

## 2. Objetivos

- Identificar y analizar las variables de entrada y salida digitales para la solución a nuestro problema.
- Realizar el diagrama de flujo de la solución.
- Realizar un código de programa en Code Composer Studio (CCS).
- verificar los resultados de la programación en CCS.

## 3. Marco Teórico

Un DSP es un dispositivo similar a un microprocesador diseñado a partir de operaciones matemáticas típicas para manipular datos a alta velocidad que son medidos y otorgados por sensores. Debido a esto es muy útil para el procesamiento de señales en tiempo real. Un DSP puede trabajar con señales analógicas debido a eso tiene un conversor analógico/digital en la entrada y un conversor digital/análogo en la salida.

El DSP está diseñado para tareas que resultan ser mas habituales en el procesamiento digital como la suma, la multiplicación, sustracción, etc. Estos dejan de lado la arquitectura Von Neuman donde tanto programas y datos se encuentran en la misma zona de memoria y optan por una arquitectura Harvard donde físicamente los datos y programas están separados pero no pueden ser totalmente independientes así que para mejorar el procesamiento los DSP son un Harvard modificado. Se llega a seleccionar un DSP cuando hay sistemas donde la carga computacional es extremadamente intensa. Entre los principales fabricantes están Texas Instruments, con la serie TMS320; Motorola, con las series DSP56000, DSP56100, DSP56300, DSP56600 y DSP96000; Lucent Technologies con las series DSP1600 y DSP3200; y Analog Devices, con las series ADSP2100 y ADSP21000.

Dentro las aplicaciones de los DSP se pueden usar para sistemas radar, telefonía celular, disqueras, módems, exploración sonar y sísmicas. No hay procesador que satisfaga

todas las necesidades o aplicaciones así que se elegirá de acuerdo con el coste, integración, consumo, volumen de datos, facilidad de uso, soporte para configuración multiprocesador, etc.

Todas las E/S digitales se agrupan en “Puertos”, llamados GPIO-A, B y C. Aquí GPIO significa "salida de entrada de propósito general". El F2833x dentro de sus pines de E/S cuenta con un total de 88 pines llamados desde GPIO0 a GPIO87. El pin del dispositivo se puede utilizar hasta para 4 diferentes funciones y depende del programador decidir qué función hará.

Cada pin puede funcionar bien como entrada o como salida y eso dependerá del registro GPxDIR así que la dirección debe configurarse. En periférico hay dos grupos GPxMUX1 O GPxMUX2 si se requiere trabajar con GPIO el GPxMUX1 y GPxMUX2 deben ser 0. Otra configuración que debe realizarse es dar el valor del bit bien sea entrada o salida para eso se usa el registro GPxDAT también hay 3 registros más para darle valor como GPxCLEAR, GPxSET y GPxTOGGLE.

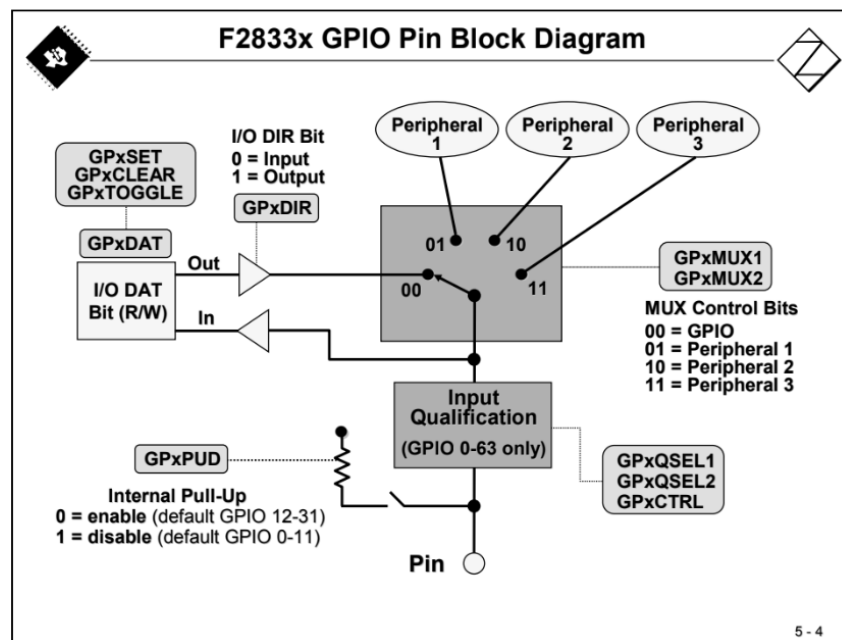


Figura 1: Diagrama de bloques de un pin físico del GPIO F2833x

La distribución de los puertos GPIO-A, B y C son: GPIO0 a GPIO31 para el bloque A, GPIO32 a GPIO63 para el bloque B y GPIO64 a GPIO87 para el bloque C, solo los puertos A y B tiene el input Qualification se pueden apreciar en la siguiente figura el conjunto de registros de control de E/S.

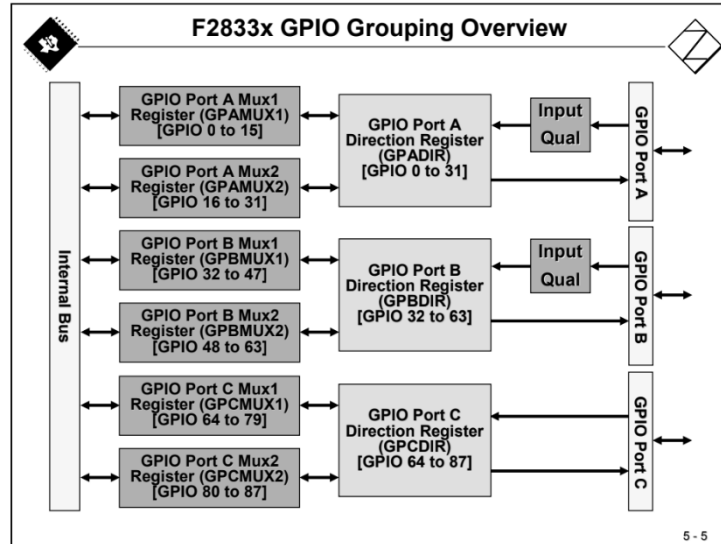


Figura 2: Agrupación GPIO F2833x

## 4. Enunciado del problema

Se requiere mezclar dos productos con agua. Se completa el depósito de agua realizando la apertura de la válvula V1. La dosificación de los productos se realiza con una tolva acumulativa, se coloca el producto A sobre la tolva hasta que se alcance un peso SP1 y luego el producto B hasta conseguir el peso total de los dos productos "SP2", tanto SP1 como SP2 se consideran como señales de entrada digital. Luego se apertura la válvula de la tolva durante 10 segundos para dejar caer el contenido. A continuación se realiza la mezcla de los productos durante 30 segundos accionando el agitador y se vacía el depósito para iniciar nuevamente el ciclo secuencial. El proceso se activa con un interruptor P.

Se requiere un programa en DSP basado en GPIOs que consideren las entradas digitales P, DL, DV, SP1 y SP2, y como salidas digitales V1, V2, V3, VA, VB y M. Considere el uso de diagramas de estado adecuados.

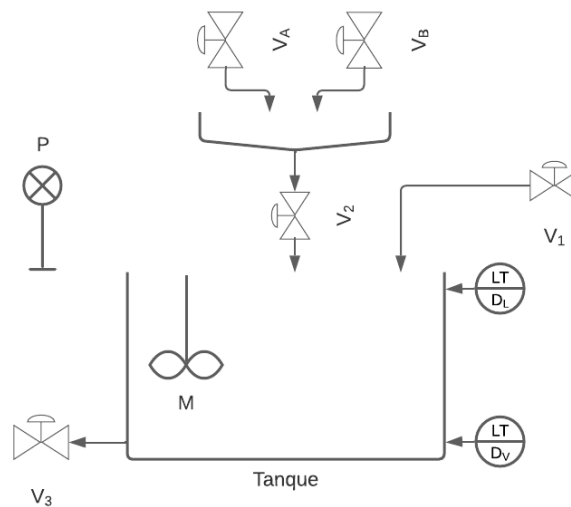


Figura 3: Diagrama del enunciado

## 5. Diagrama de flujo

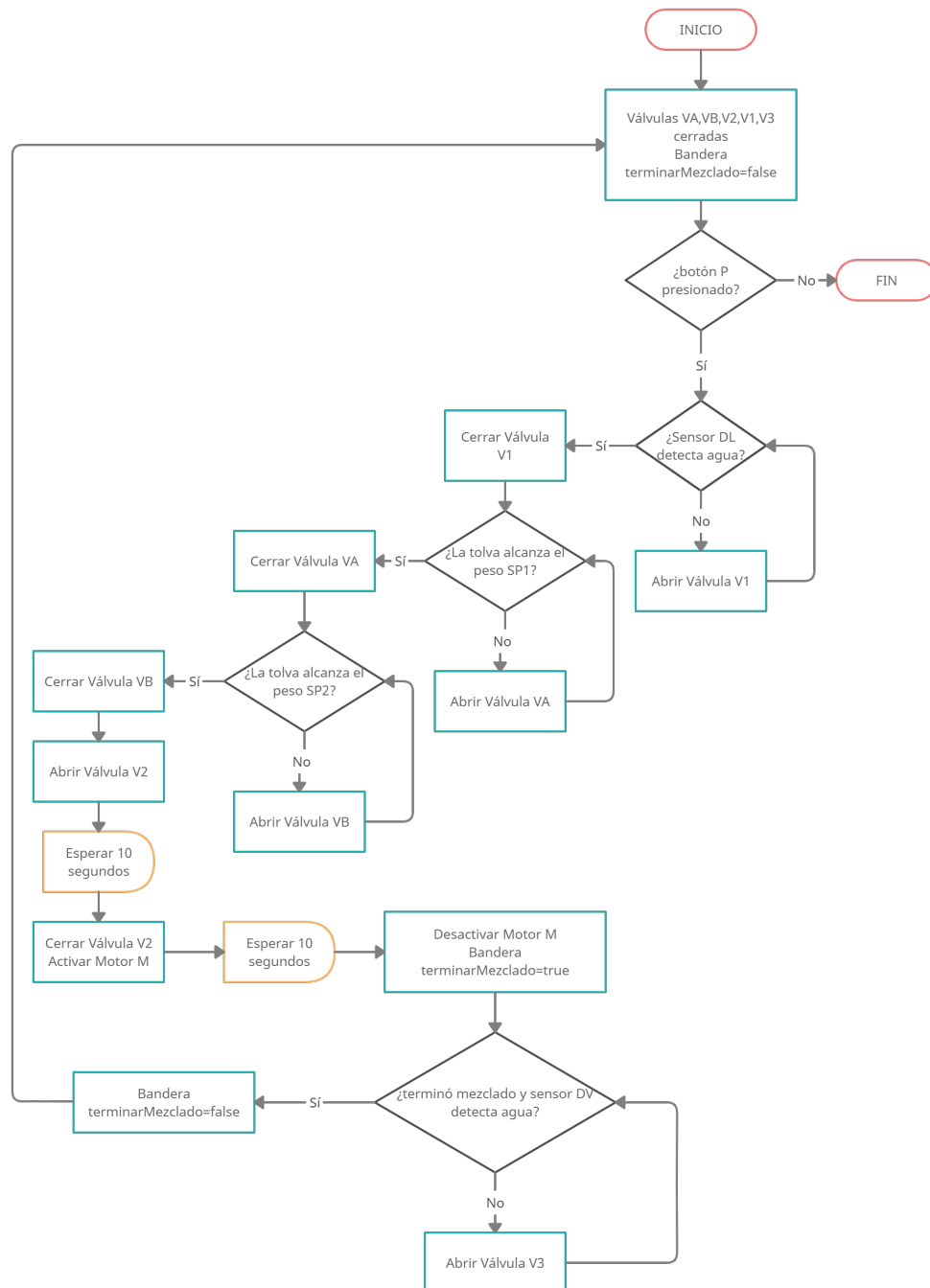


Figura 4: Diagrama de Flujo General

## 6. Código en Code Composer

```
#include "DSP2833x_Device.h"
#include "DSP2833x_Examples.h"

/***** DEF DE FUNCIONES *****/
void Gpio_select(void); // definiendo una funcion auxiliar de configuracion de
    puertos
void proceso(void); // definiendo una funcion auxiliar de proceso
void delay_1s(void); // definiendo delay de 60 segundos
void abrirV2(void);
void activarMezclado(void);
/***** VARIABLES GLOBALES *****/
int terminarMezclado=0;
/***** MAIN *****/
void main(){
    InitSysCtrl(); // es una funcion de configuracion inicial DEL REGISTRO DE
        CONTROL ubicada en el programa DSP2833x_SysCtrl.c
    Gpio_select(); // ejecutando la funcion auxiliar de configuracion de puertos
        DINT;
    InitPieCtrl(); // es una funcion de configuracion inicial DEL REGISTRO PIE
        ubicada en el programa DSP2833x_PieCtrl.c
    // Disable CPU interrupts and clear all CPU interrupt flags:
    IER = 0x0000;
    IFR = 0x0000;
    InitPieVectTable(); // es una funcion de configuracion inicial ubicada en el
        programa DSP2833x_PieVect.c
    proceso();
}

/***** FUNCIONES *****/
void Gpio_select(void){ //define entradas y salidas
    EALLOW;
    GpioCtrlRegs.GPAMUX1.all = 0x00000000; // All GPIO
    GpioCtrlRegs.GPBMUX1.all = 0x00000000; // All GPIO

    GpioCtrlRegs.GPADIR.all = 0x00000000; // All Port A inputs
    GpioCtrlRegs.GPBDIR.all = 0xFFFFFFFF; // All Port B outputs
    EDIS;
}

void proceso(void){
    //DEFINIENDO MACROS
    //ENTRADA A: 0->31
    #define P      GpioDataRegs.GPADAT.bit.GPIO2
    #define DL      GpioDataRegs.GPADAT.bit.GPIO4
    #define DV      GpioDataRegs.GPADAT.bit.GPIO6
    #define SP1     GpioDataRegs.GPADAT.bit.GPIO8
    #define SP2     GpioDataRegs.GPADAT.bit.GPIO10
    //SALIDA B: 32->63
    #define v1      GpioDataRegs.GPBDAT.bit.GPIO32
```



```

#define v2      GpioDataRegs.GPBDAT.bit.GPIO33
#define v3      GpioDataRegs.GPBDAT.bit.GPIO34
#define VA      GpioDataRegs.GPBDAT.bit.GPIO48
#define VB      GpioDataRegs.GPBDAT.bit.GPIO59
#define M      GpioDataRegs.GPBDAT.bit.GPIO61
GpioDataRegs.GPBDAT.all = 0x00000000;

while(1){ //loop
    if (P==1){
        //llenadoAguaTanque-----
        while(DL!=1){
            v1=1;//v1=1 - llenado del tanque de agua
            //esperar llenado
        }
        if(DL==1){
            v1=0;//se apaga el llenado porque alcanzo el limite DL
        }
        //-----
        //llenadoTolvaAcumulativa
        while(SP1==0){//mientras que el peso no sea sp1
            VA=1;//lenar del contenido a
        }
        if(SP1==1){
            VA=0;//apagar A
        }
        while(SP2==0){//MIENTRAS QUE EL PESO NO SEA SP2
            VB=1;//llenar del contenido B
        }
        if(SP2==1){
            VB=0;//apagar B
            abrirV2();//dejar caer el contenido por "10 segundos".
            activarMezclado();//activar el mezclado por "30 segundos"
        }
        while(terminarMezclado==1&&DV==1){
            v3=1;
        }
        if(DV==0){//si el deposito esta vacio
            terminarMezclado=0;
            v3=0;
        }
    }
}

void abrirV2(){
    v2=1;
    delay_1s();
    delay_1s();
    v2=0;
}

void activarMezclado(){
    M=1;

```

```

    delay_1s();
    delay_1s();
    M=0;
    terminarMezclado=1;
}
void delay_1s(){
    long i;
    for (i = 0; i < 20000000 ; i++) {}
}

```

## 7. Resultados

Se armó el circuito con switchs en configuración pull up conectados a los pines GPIO A del dsp y leds conectados a los pines del GPIO B del dsp. Se trabajó con un regulador de voltaje para protoboard de 5V y 3.3V cuya máxima corriente de alimentación es de 700mA. Considerando que todos los pines tienen una potencia de accionamiento de 4mA típica. Y las entradas no son tolerantes a 5V. Se requieren de 5 entradas por lo que se necesita 20mA provenientes de la fuente. Esto sin considerar las pérdidas de potencia en los cables o pérdida de la señal original por diafonía.

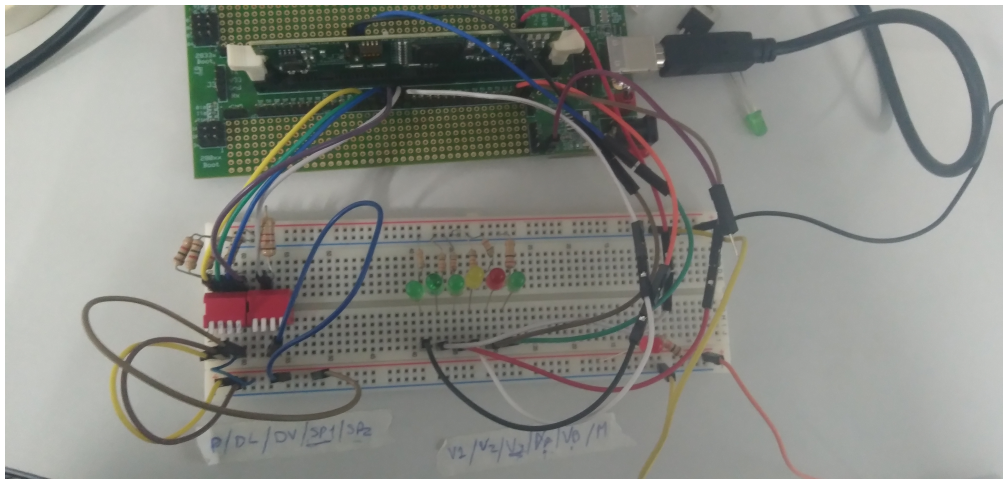


Figura 5: Circuito de las entradas y salidas digitales con el DSP TMS320F28335

El circuito armado considera entradas digitales al boton de inicio P, el sensor de nivel de agua del tanque DL y DV y a los pesos SP1 y SP2 que debe cumplir la tolva para el vertido de contenido A y B. Estos inputs representados como switchs pull up.

El circuito tiene como salidas digitales del sistema a las válvulas V1, V2, V3, VA, VB y el motor M. Estos outputs son representados por leds.

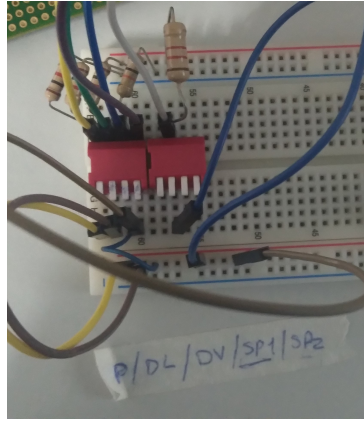


Figura 6: Inputs del circuito etiquetados P,DL,DV,SP1,SP2

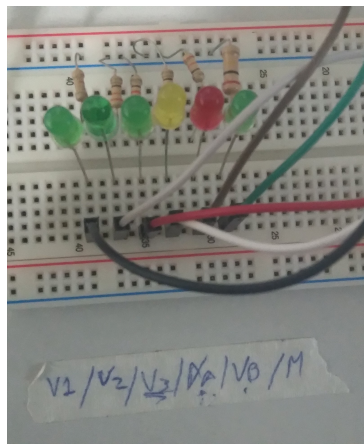


Figura 7: Outputs del circuito etiquetados V1,V2,V3,VA,VB,M

## 8. Conclusiones

Nuestra solución cumple satisfactoriamente con el control del proceso de mezcla de productos en un deposito de agua usando diferentes funciones y los pines digitales para representar las entradas y salidas del proceso. Además este es un proceso secuencial por lo cual no fue necesario el uso de interrupciones.

Para la correcta implementación de este tipo de aplicación de automatización se concluye que primero se deben verificar el circuito electrónico implementado, verificar si no hay corto y que exista continuidad. También, tomar en cuenta la potencia suministrada para los inputs del sistema que van conectados al dsp, tomando en cuenta una entrada de 3.3V.

## 9. Recomendaciones

- Realizar primero diagrama de flujo para que se nos facilite la explicacion en el funcionamiento del programa.
- Tener en cuenta los puertos de los cuales se hace uso y cuales no.
- Tener clara la configuracion de los pinens tanto de entradas como de salidas asi como los puertos a los que pertenecen para evitar futuros errores.

- Verificar la correcta polaridad del circuito armado antes de conectarlo al dsp.
- Verificar la lectura de los gpios de nuestro circuito con el dsp uno por uno, antes de conectarlo al circuito general, esto se puede hacer con un solo boton o switch y un led.
- En la implentacion hay que etiquetar los jumper con las respectivas entradas y salidas para evitar confusiones.

## Referencias

- [1] Salazar. J. Procesadores Digital de señal (DSP) Arquitectura y criterios de seleccion.  
<https://www.um.es/documents/4874468/19345367/ssee-da-t04-01.pdf/20ed49bb-90d7-4d1b-a5d5-bc5cad34fef>
- [2] Phil Lapsley, Jeff Bier, Amit Shoham and Edward A. Lee, “DSP Processor Fundamentals: Architectures and Features”, Berkeley, California: Berkeley Design Technology, Inc., 1996.