Comité de Empresas de Software Embebido de Querétaro (CESEQ)

Diplomado en Software Embebido

Reporte del Proyecto Integrador

“Control de velocidad de motor de CD”

Que presentan:

Blancas Núñez Jorge

Martínez Rodríguez Ignacio

ASESOR

Luís Uriastegui

## Historial de cambios

|  |  |  |  |
| --- | --- | --- | --- |
| No Revisión | Descripción del cambio | Autor | Fecha |
| 1 | Documento inicial | Jorge Blancas | 25 enero 2019 |
| 2 | Metodología de desarrollo | Ignacio Martinez | 15 Marzo 2019 |
| 3 | Anexo de análisis de riesgos |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Alcance del proyecto

## Identificación del proyecto

SW de control para Dispositivo CESEQ-001

## Descripción del proyecto

Se controlará la velocidad de un motor de corriente directa mediante la aplicación de una señal cuadrada que varía en su ancho de pulso y cuya frecuencia de trabajo será fija fo=10KHz. Mediante el uso de un sensor de efecto hall acoplado al rotor del motor se deberá medir la velocidad del motor el cual proveerá 15 pulsos cada que se complete una vuelta completa (así, a mayor velocidad del motor, mayor será el número de pulsos leídos, mientras que a menor velocidad, menor será el número de pulsos). El voltaje de alimentación del motor será de 12 Vcd. El valor de referencia o “SetPoint” (velocidad deseada) deberá estar dado por el ajuste de una resistencia variable (potenciómetro). La pantalla LCD o interfaz gráfica deberá mostrar la velocidad del motor y SetPoint (ambos en RPM’s); así como el porcentaje de trabajo de la señal cuadrada.

## Requisitos específicos

|  |  |  |
| --- | --- | --- |
| ID Requisito | Descripción | Comentario |
| R1 | Alimentación del Motor de CD de 12 volts |  |
| R2 | Frecuencia del PWM 10kHz |  |
| R3 | Ajuste de la velocidad set point mediante potenciómetro |  |
| R4 | Uso de pantalla LCD para mostrar velocidad en RPM´s y porcentaje de PWM |  |
| R5 | Driver de manejo de potencia DRV8848-2A |  |
| R6 | Tipo de control PID, |  |
| R7 | Sistema operativo Máquina de estados finitos |  |
| R8 | Inicialización de memoria EEPROM |  |
| R9 | Temporizador para interrupción |  |
| R10 | Convertidor analógico digital |  |
| R10 | Salida PWM |  |
| R11 | Puertos digitales |  |
| R12 | Watchdog timer |  |
| R13 | Input Capture |  |
| R14 | Pantalla LCD | Puerto SPI |
| R15 | Detección de corto a batería | Análisis dinámico |
| R16 | Detección de corto a tierra | Análisis estático |
| R17 | Monitoreo de tareas 10X tic | Variables tipo entero |
| R18 | Throughput < 70% |  |
| R19 | Diagnóstico corrupción de memoria interna |  |
| R20 | Diagnóstico botón en corto |  |
| R21 | Complejidad ciclomática <19 |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Entregables con el cliente. Compromisos de entrega de los módulos del software

El compromiso es de entregar todos los módulos de software que implementen las funcionalidades que se describen en los requisitos específicos, estos módulos estarán documentados y probados, de manera individual y en su conjunto.

# Metodologías de desarrollo

## Metodología de desarrollo de software seleccionada.

Se utiliza la metodología Agile SCRUM la cual permite emplear entregables en lapsos cortos de tiempo con la finalidad de que el cliente o asesor permita ir visualizando el avance que se tiene en el proyecto.

## Plan de entregas de software al cliente

Se platea entregas por cada sprint el cual deberá tener una duración no máxima a tres semanas en

Común acuerdo con el asesor.

|  |  |  |
| --- | --- | --- |
| Entregable | Módulos del Proyecto | Sprint |
| 1 | Análisis de requisitos | 1 |
| 2 | Análisis y definición de módulos con los que contara el proyecto integrador | 2 |
| 2 | Implantación de control | 3 |
| 3 | Implementación módulo de Display | 4 |
| 4 | Pruebas de Integración | 5 |
| 5 | Liberación de sw de control. | 6 |
| 6 | Documentación | 7 |

## Procedimientos de control

Tabla de requisitos.

|  |  |
| --- | --- |
| ReqFun1 | El sistema deberá capturar el tren de pulsos del sensor de efecto hall  el cual proveerá 15 pulsos cada vez que se complete una vuelta |
| ReqFun2 | Definir el intervalo de actuación del setpoint para tener la velocidad deseada. |
| ReqFun3 | Asegurar la protección contra la corriente de descarga en la implementación del diodo de giro |
| ReqFun4 | El voltaje de alimentación del motor deberá ser de 12 Vcd y su frecuencia será fija fo=10KHz |
| ReqFun5 | Realizar el ajuste de valor de referencia el cual debe estar dada por un potenciómetro en RPMs. |
| ReqFun6 | El Sistema deberá Mostar en el display el porcentaje de operación puede ir desde 0%,25%… al 100% |
| ReqFun7 | Se debe realizar un control () |
| ReqFun8 | Realizar la configuración del sistema operativo |
| ReqFun9 | El sistema al iniciar deberá considerar la configuración previamente cargada. |
| ReqFun10 | Se deberá realizar un análisis de rendimiento para determinar el % si es mayor al 70% |
| ReqFun11 | del uso del CPU reconsiderar el diseño propuesto. |
| ReqFun12 | Las tareas podrán ser monitoreadas con una resolución de al menos 10x TIC's |
| ReqFun13 | Se deberán almacena las tareas en variables de tipo entero |
| ReqFun14 | El sistema deberá implementar diagnósticos de Potencia Dinámicos y estáticos. |
| ReqFun15 | El sistema debe de implementar Diagnósticos de corrupción de Memoria |
| ReqFun16 | El sistema debe de implementar un botón en corto. |
| ReqFun17 | Se debe de implementar diagnóstico de corto a tierra. |
| ReqFun18 | Se debe de implementar diagnóstico de corto a batería. |

|  |  |
| --- | --- |
| ReqNFun19 | Implementar un contraste adecuado en el display |
| ReqNFun20 | El display deberá contar con buena iluminación |
| ReqNFun21 | Implementar velocidad de refresco de display adecuado |
| ReqNFun22 | Implementar procedimiento de control de requisitos, versiones de sw y liberaciones |

Control de versiones

Se emplea el sistema de Control de versiones con la herramienta GIT.

# Estimación

## Recursos

* + 1. Recursos críticos (RAM, ROM, Throughput)

El sistema de desarrollo Renesas SK-S7G2 utilizado cuenta con:

Microcontrolador S7G2 ARM Cortex M4

Reloj a 240 MHz

RAM

ROM

Entradas digitales

Entradas analógicas

Puerto RS232

Puerto RS485

Puerto CAN

Puerto USB

2 X Interfase Pmod

Tres leds para el usuario

Pantalla Touchscreen LCD

* + 1. Infraestructura y Herramienta

Kit de desarrollo Renesas SK-S7G2.

1 Osciloscopio

2 puntas para osciloscopio

1 Generador de funciones

1 Potenciometro.

* + 1. Recursos humanos.

Se contempló que se asignara a tres recursos por equipo, por situaciones de índole laboral uno de los integrantes se retiró del curso por lo que solo se trabajó a lo largo del proyecto con solo dos recursos.

1. Suposiciones/ Restricciones/ Riesgos/ Resolución de problemas
   1. Suposiciones

Todas las actividades que se desconocen producen un retraso en la entrega de cada módulo del proyecto, así como cada nuevo requerimiento que se plantea.

Es preciso que se genere un buen análisis de toda la información para que se pueda consultar de forma oportuna con el asesor y descartar posibles retrasos.

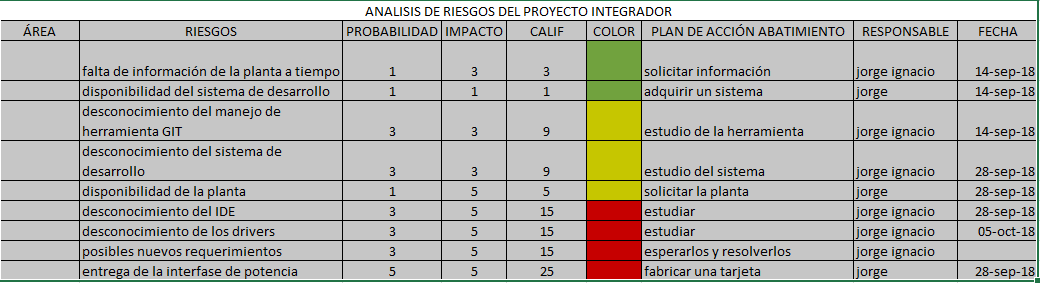
* 1. Restricciones

Falta de información de la plata a tiempo

Disponibilidad del sistema del desarrollo

Desconocimiento del IDE de desarrollo

* 1. Riesgos

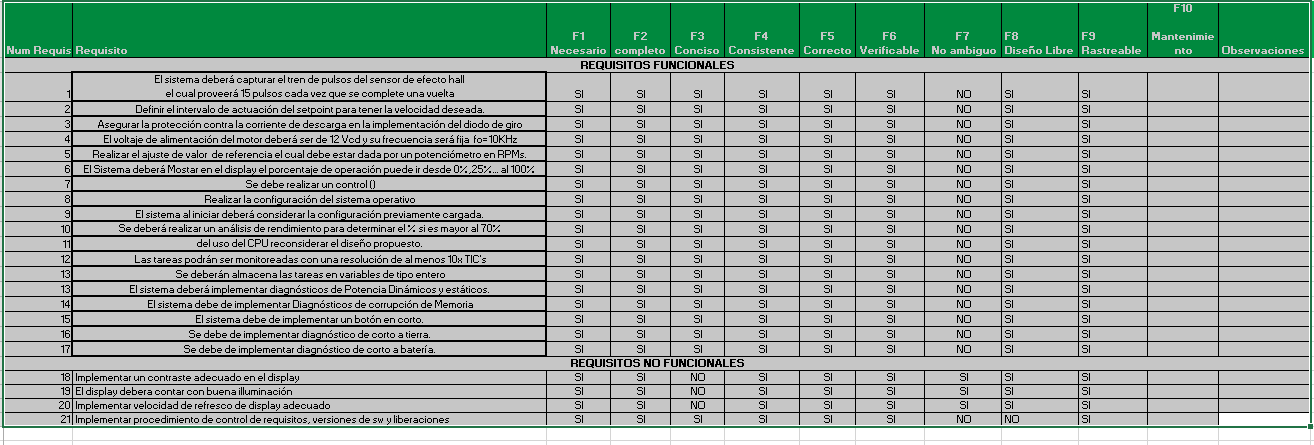


* 1. Resolución de problemas

1. Planeación
   1. Roles y responsabilidades

|  |  |  |
| --- | --- | --- |
| Rol | Responsable | Actividades |
| Investigación | Jorge e Ignacio | El uso del IDE de desarrollo.  Resolución de problemas con equipo de computo |
| Programador | Jorge e Ignacio | Implementación de cada módulo del control de velocidad para su desarrollo. |
| Documentador | Jorge e Ignacio | Generar la documentación del proyecto. |

* 1. Trazabilidad y entregables



Entregables.

|  |  |  |
| --- | --- | --- |
| Entregable | Módulos del Proyecto | Sprint |
| 1 | Análisis de requisitos | 1 |
| 2 | Análisis y definición de módulos con los que contara el proyecto integrador | 2 |
| 2 | Implantación de control | 3 |
| 3 | Implementación módulo de Display | 4 |
| 4 | Pruebas de Integración | 5 |
| 5 | Liberación de sw de control. | 6 |
| 6 | Documentación | 7 |

* 1. Administración de riesgos
  2. Análisis de requisitos de software

SWE.1 Análisis de requisitos

1. Implementar configuración de Display
2. Implementar formato o diseño para mensajes de Display
3. Implementar refresco del Display ¿preguntar a cliente cual es el periodo o frecuencia para realizar esta tarea?
4. Almacenar el diagnostico de corto. ¿preguntar a cliente cual formato de archivo para guardar?
5. El cambio en el tiempo On puede ir desde 0%,25%… al 100% ¿Preguntar a cliente si Deberán existir rangos intermedios?
6. El sistema debe detectar un corto a batería y a tierra. ¿Preguntar a cliente que acción deberá realizar en estos casos?
7. El sistema debe detectar corrupción de memoria
8. El sistema deberá de generar análisis estático y dinámico.
9. Apariencia del Display ¿Preguntar a cliente Cuál es el rango de contraste adecuado para el display?
   1. Diseño de arquitectura de software
      1. Diagrama de bloques de software
      2. Diagrama de control
         1. Entradas
         2. Salidas
         3. Ruido
         4. Sistema
         5. Flujo de datos
      3. UML
         1. Diagrama de estados
         2. Diagrama de tiempos/ recursos
      4. Árbol de llamadas
      5. Diagramas de flujo
      6. Final % uso de ROM, % uso de RAM, STACK, Mapa de memoria
      7. Throughput
      8. Complejidad de código (Índice Cyclomatic Complexity)

<?xml version="1.0" encoding="UTF-8"?>

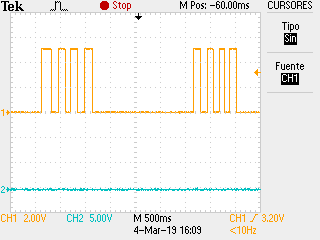
<!--Report on software metrics-->

[<CCCC\_Project>](file:///D:\Diplomado\CCCC\cccc.xml)<timestamp>Fri Mar 15 12:19:50 2019 </timestamp>[<project\_summary>](file:///D:\Diplomado\CCCC\cccc.xml)<number\_of\_modules level="**0**" value="**1**"/><lines\_of\_code level="**0**" value="**108**"/><lines\_of\_code\_per\_module level="**0**" value="**108.000**"/><McCabes\_cyclomatic\_complexity level="**0**" value="**6**"/><McCabes\_cyclomatic\_complexity\_per\_module level="**0**" value="**6.000**"/><lines\_of\_comment level="**0**" value="**36**"/><lines\_of\_comment\_per\_module level="**0**" value="**36.000**"/><lines\_of\_code\_per\_line\_of\_comment level="**0**" value="**3.000**"/><McCabes\_cyclomatic\_complexity\_per\_line\_of\_comment level="**0**" value="**0.167**"/><IF4 level="**0**" value="**0**"/><IF4\_per\_module level="**0**" value="**0.000**"/><IF4\_visible level="**0**" value="**0**"/><IF4\_visible\_per\_module level="**0**" value="**0.000**"/><IF4\_concrete level="**0**" value="**0**"/><IF4\_concrete level="**0**" value="**0.000**"/><rejected\_lines\_of\_code level="**0**" value="**35**"/></project\_summary>[<procedural\_summary><module>](file:///D:\Diplomado\CCCC\cccc.xml)<name>anonymous</name><lines\_of\_code level="**0**" value="**73**"/><McCabes\_cyclomatic\_complexity level="**0**" value="**6**"/><lines\_of\_comment level="**0**" value="**26**"/><lines\_of\_code\_per\_line\_of\_comment level="**0**" value="**2.808**"/><McCabes\_cyclomatic\_complexity\_per\_line\_of\_comment level="**0**" value="**0.231**"/></module></procedural\_summary>[<oo\_design><module>](file:///D:\Diplomado\CCCC\cccc.xml)<name>anonymous</name><weighted\_methods\_per\_class\_unity level="**0**" value="**2**"/><weighted\_methods\_per\_class\_visibility level="**0**" value="**0**"/><depth\_of\_inheritance\_tree level="**0**" value="**0**"/><number\_of\_children level="**0**" value="**0**"/><coupling\_between\_objects level="**0**" value="**0**"/></module></oo\_design>[<structural\_summary><module>](file:///D:\Diplomado\CCCC\cccc.xml)<name>anonymous</name><fan\_out\_visible level="**0**" value="**0**"/><fan\_out\_concrete level="**0**" value="**0**"/><fan\_out level="**0**" value="**0**"/><fan\_in\_visible level="**0**" value="**0**"/><fan\_in\_concrete level="**0**" value="**0**"/><fan\_in level="**0**" value="**0**"/><IF4\_visible level="**0**" value="**0**"/><IF4\_concrete level="**0**" value="**0**"/><IF4 level="**0**" value="**0**"/></module></structural\_summary>[<other\_extents><rejected\_extent>](file:///D:\Diplomado\CCCC\cccc.xml)<name><file scope items></name><source\_reference line="**1**" file="**c:/hal\_entry.c**"/><lines\_of\_code level="**0**" value="**35**"/><lines\_of\_comment level="**0**" value="**10**"/><McCabes\_cyclomatic\_complexity level="**0**" value="**0**"/></rejected\_extent></other\_extents></CCCC\_Project>

1. Construcción del software
   1. Aplicación de estándares de codificación y convenciones
      1. Nomenclatura de archivos
      2. Nomenclatura de funciones
      3. Nomenclatura de clases
      4. Nomenclatura de variables
         1. Variables locales
         2. Variables globales
      5. Macros
      6. Tipos de datos
         1. Estándares/ primitivos
         2. Definidos por el usuario
      7. Estructura del proyecto
         1. **Análisis de la planta a controlar**
            1. El motor es un motor de CD y después de mediciones en laboratorio, de acuerdo al diagrama de la figura 1, se presenta la siguiente tabla (tabla 1), de su comportamiento, la consideración que tenemos en cuenta es que el sensor de RPM genera 4 pulsos por revolución, el cual es un dato que es independiente de cualquier otra variable, por esto se toma como base para determinar la velocidad del motor en RPM ( revoluciones Por minuto).

**Determinación de los pulsos por revolución:**

Se hizo girar una revolución completa al motor y se obtuvo la siguiente gráfica donde se muestra que por cada revolución se generan cuatro pulsos provenientes del sensor del efecto Hall, con esta señal podremos determinar la velocidad del motor en revoluciones por minuto RPM.



Figura, Pulsos por revolución que genera el sensor de efecto hall, aquí se observa la respuesta a dos revoluciones generadas por el motor.



Figura 1 Esquema de caracterización del motor,

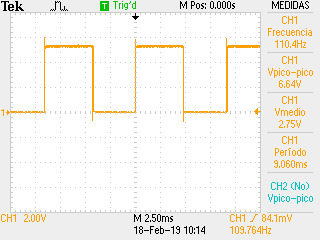


Figura 2 Respuesta del sensor Hall, la velocidad del motor se controló mediante la varicación de la tensión de trabajo

* + - * 1. Fórmulas utilizadas:

Periodo T=1/F [segundos] [1]

Velocidad V [rev/min]= F[pulsos/seg] \* 60[seg/min] \* 1/4 [rev/pulsos] [2]

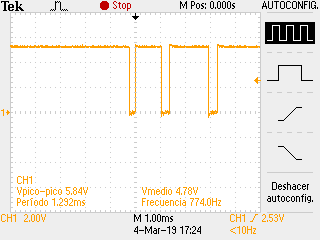
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| V Tensión de control (Volts) | I Corriente de consumo (mA) | F Frecuencia medida (Hz) | T Periodo (ms) | V Velocidad calculada ( RPM) |
| 2 | 50 | 13.5 | 74.074 | 202 |
| 4 | 70 | 62.8 | 15.923 | 942 |
| 6 | 120 | 107 | 9.345 | 1605 |
| 8 | 190 | 148 | 6.756 | 2220 |
| 10 | 270 | 182 | 5.494 | 2730 |
| 12 | 350 | 215 | 4.651 | 3225 |

Tabla 1 caracterización del motor

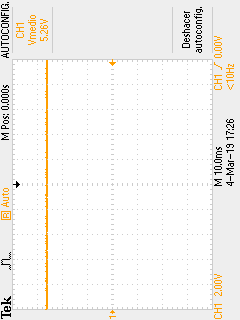
**Problema:** al tratar de repetir la caracterización del motor utilizando el generador de funciones como fuente de señal PWM a 10 kHz, observamos que el sensor de efecto hall , no responde como se esperaba, se utilizó un montaje de acuerdo al siguiente esquema, figura 2



Circuito original para conocer la respuesta del sensor de efecta Hall



Respuesta errática del sensor de efecto Hall, PWM @10kHz, DC=50%



Respuesta errática del sensor de efecta Hall, PWM @10kHz, DC=70%

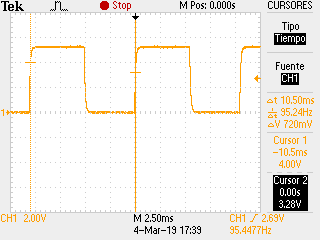
**Solución al problema**, después de analizar la respuesta del sensor de efecto hall, controlando la velocidad del motor mediante la variación de la tensión de control y observando que cuando se utiliza la señal de PWM a 10 kHz y otras frecuencias inferiores, se determinó que:

Debido a que el sensor de efecto Hall es un sensor que detecta la presencia del campo magnético, entonces se ve afectado por el campo magnético propio de la frecuencia del PWM, así que se optó por poner un filtro pasabajos de segundo órden formado por un inductor de L1 de22uH y un condensador electrolítico C1 de 1000uF, este arreglo presenta una atenuación cercana a los 40 dB a una frecuencia de 10kHz, aún asi la señal del sensor Hall se veía con un poco de ruído por lo que se utilizó un segundo filtro de primer orden formado por un resistor R1 de 1kohm y un condensador cerámico de 100nF, este arreglo se observa en el siguiente diagrama



Circuito para acondicionar la señal de PWM y la señal del sensor de efecto Hall, mediante filtros pasivos L1 C1 y R1 C2

A continuación se muestra la respuesta del sensor de efeto Hall utilizando los filtros antes señalados.



Respuesta del sensor Hall después de utilizar un filtro LC en la salida de PWM y un filtro RC en la salida de pulsos del motor, se observa como se recupera la respuesta del sensor de efecto Hall.

PWM @10kHz, DC=50% filtro L1=22uH C1=1000uF R1=1kohm C2=100nF

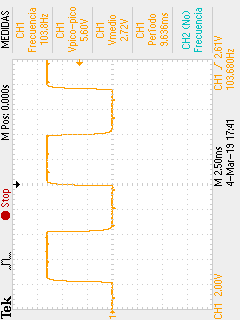


Fig F0034TEK.BMP Respuesta del sensor Hall después de utilizar un filtro LC en la salida de PWM y un filtro RC en la salida de pulsos del motor, se observa como se recupera la respuesta del sensor de efecto Hall.

PWM @10kHz, DC=70%, filtro L1=22uH C1=1000uF R1=1kohm C2=100nF

Evaluación de la respuesta dinámica del motor

Para controlar adecuadamente cualquier sistema, se requiere conocer su comportamiento dinámico, en el ambiente de control también se utiliza el término identificación de la planta, esto se logra aplicando un estímulo conocido y se evalua el comportamiento del sistema, en nuestro caso se aplicó una señal escalón, el cual es un estimulo (ancho de pulso conocido) y se midió la respuesta de velocidad del motor, la foma de medir la velocidad del motor se logró al medir los anchos de pulso generados por el sensor de velocidad de efecto hall, dado que el ancho de pulso corresponde a la frecuencia de los pulsos y esta frecuencia corresponde directamente a la velocidad del motor; utiizamos un módulo propio del sistema de desarrollo SKS7G2 y su SSP, el cual es capaz de medir anchos de pulso, posteriormente se calculó la velocidad en RPM y se pasó al convertidor Digital-Analógico del SKS7G2, de esta manera la señal de velocidad se pudo graficar en el osciloscopio. El montaje experimental se iustra en la siguiente imagen.



Arreglo para encontrar la respuesta temporal del motor a una entrada escalón

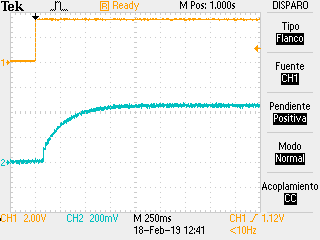


Fig. F0021TEK.BMP Respuesta temporal del motor a una entrada escalón, FPWM=10kHz, DC=75%.

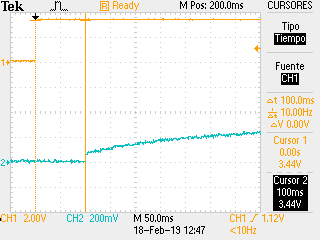


Fig F0023TEK.BMP En esta imagen podemos encontrar un timpo muerto de 100 ms DC=75%

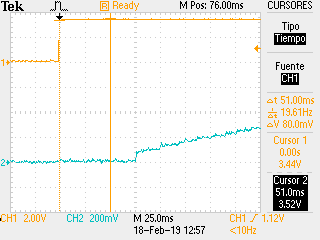


Fig. F0028TEK.BMP En el caso extremo de DC=100%, el tiempo muerto se reduce a 51 ms

Ahora analizando las imágenes obtenemos los parámetros que nos permiten obtener las constantes PID para el control de velocidad, de acuerdo al método sugerido por Ziegler/Nichols





Tomando de las figuras anteriores los tiempos t0, t1, t2 y los valores del escalón y la respuesta en RPM, tenemos:







segundos



RPM



RPM





















Constante proporcional





Constante integrativa





Constante derivativa



tiempo de muestreo













Convertidor analógico-digital ADC

Este convertidor nos servirá para seleccionar la velocidad a la que girará el motor, cuenta con 12 bits por lo que sus lecturas podrán tomar cualquier valor entre 0 y 4095, y que podremos escalar para una velocidad entre 0 y 3300 RPM. Así que el SETPOINT se calculará con la siguiente ecuación tomando inicialmente el valor de convertidor ADC

SETPOINT= ADC()\*3300/4095;

Medición de la velocidad con el sistema SK S7G2

Como ya se comentó el motor cuenta con un sensor de fecto hall el cual es capaz de entregar 4 pulsos por revolución , así que la frecuencia de estos pulsos es representativa de la velocidad, el periodo de los pulsos como es sabido depende de la frecuencia de los pulsos y el microcontrolador es capaz de medir el tiempo de duración de los pulsos, la ecuación del cálculo de la velocidad en RPM es la siguiente.



pulsos por revolución



minutos por segundo

cálculo de la velocidad conociendo la frecuencia de los pulsos:



Como T=1/f entonces



Ahora si medimos el semiperiodo es decir el ancho de los pulsos tenemos





Si hacemos



La ecuación de cálculo de la velocidad conociendo el ancho de pulso será:



Entonces el algoritmo deberá medir el ancho del pulso proveniente del sensor Hall y utilizar la fórmula anterior para conocer la velocidad del motor.

**Problema.** Debido al método utilizado para medir la velocidad del motor, el cual se basa en la medición del ancho de los pulsos provenientes del sensor Hall, entonces el tiempo de muestreo es variable y depende de la misma velocidad del motor, para llegar a tener un tiempo de muestreo uniforme, se propone utilizar una rutina de retardo cuyo retardo sea variable y se ajuste a la velocidad del motor.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **useg** | **tiempo** |  |
| **f** | **RPM** | **delay** | **ciclo ms** | **ts** |
| 13.5 | 206 | 840 | 74.16 | 75 |
| 14.4 | 220 | 5480 | 69.52 | 75 |
| 16.4 | 250 | 13880 | 61.12 | 75 |
| 17.5 | 270 | 18440 | 56.56 | 75 |
| 20.1 | 306 | 25160 | 49.84 | 75 |
| 26.63 | 406 | 37400 | 37.6 | 75 |
| 33.2 | 506 | 44840 | 30.16 | 75 |
| 39.8 | 607 | 49840 | 25.16 | 75 |
| 46.48 | 709 | 53460 | 21.54 | 75 |
| 53.1 | 810 | 56140 | 18.86 | 75 |
| 59 | 900 | 58040 | 16.96 | 75 |
| 62.1 | 947 | 58880 | 16.12 | 75 |
| 66 | 1006 | 59830 | 15.17 | 75 |
| 72.32 | 1100 | 61150 | 13.85 | 75 |
| 78.9 | 1200 | 62300 | 12.7 | 75 |
| 85.5 | 1300 | 63300 | 11.7 | 75 |
| 92 | 1400 | 64110 | 10.89 | 75 |
| 98.57 | 1500 | 64840 | 10.16 | 75 |
| 105.06 | 1600 | 65460 | 9.54 | 75 |
| 111.56 | 1700 | 66032 | 8.968 | 75 |
| 131.22 | 2000 | 67376 | 7.624 | 75 |
| 145 | 2213 | 68096 | 6.904 | 75 |
| 158 | 2412 | 68664 | 6.336 | 75 |
| 171 | 2612 | 69144 | 5.856 | 75 |
| 184 | 2810 | 69560 | 5.44 | 75 |
| 197 | 3009 | 69912 | 5.088 | 75 |
| 203 | 3103 | 70063 | 4.937 | 75 |
| 215 | 3284 | 70348 | 4.652 | 75 |

Cálculo del controlador PID, el cual deberá de utilizar un tiempo de muestreo de la velocidad de Ts= 0.02 segundos que es un tiempo de muestreo Ts < t1 / 4 de acuerdo a los criterios de Ziegler/Nichols.

1. Metodología de revisión de software
   * 1. Pruebas de software
        1. Pruebas de integración
        2. White box test
        3. Gray Box (opcional)
        4. Black Box Test
        5. ABA test
2. Liberación de versiones de software
   1. GIT
3. Verificación y aseguramiento de la calidad
   1. Prueba funcional y validación en campo
   2. Calendario de revisión: documental y código