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

Piñón Raymundo

ASESOR

Luís Uriastegui

## Historial de cambios

|  |  |  |  |
| --- | --- | --- | --- |
| No Revisión | Descripción del cambio | Autor | Fecha |
| 1 | Documento inicial | Jorge Blancas | 25 enero 2019 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 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 y setpoint 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 decriben 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

## Plan de entregas de software al cliente

## Procedimientos de control

Tabla de requisitos.

Control de versiones

# 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
    2. Recursos humanos

1. Suposiciones/ Restricciones/ Riesgos/ Resolución de problemas
   1. Suposiciones
   2. Restricciones
   3. Riesgos
   4. Resolución de problemas
2. Planeación
   1. Roles y responsabilidades
   2. Trazabilidad y entregables
   3. Administración de riesgos
   4. Análisis derequisitos de software
   5. 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)
3. 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
      8. Normas aplicables
4. 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
5. Liberación de versiones de software
   1. GIT
6. Verificación y aseguramiento de la calidad
   1. Prueba funcional y validación en campo
   2. Calendario de revisión: documental y código