

Laboratorio III: Diseño de un sistema de adquisición de datos. Parte II

Prof. Kaleb Alfaro Badilla

1. Objetivo General

Elaborar software para hacer uso de un sistema de adquisición de datos de dos canales de tensión.

2. Objetivos Específicos

1. Elaborar una visualización de datos adquiridos en tiempo de real.
2. Crear un diseño de software que permita ajustar las unidades de magnitud física sobre los datos adquiridos.
3. Construir mecanismos para almacenar de manera segura grabaciones realizadas con el DAQ.

3. Investigación Previa

Con base en las siguientes preguntas, elabore un documento en PDF o en MARKDOWN con sus respuestas y subalo en su repositorio de Github

1. Investigue ¿Qué es una base de datos SQL o NoSQL? Elabore un listado de comandos básicos para una base de datos SQL

R/ Base de Datos SQL:

Definición: Las bases de datos SQL son bases de datos relacionales que utilizan tablas para almacenar datos y requieren un esquema predefinido.

Ejemplos de Sistemas de Gestión de Bases de Datos SQL (SGBD SQL): MySQL, PostgreSQL, SQLite, SQL Server, Oracle, etc.

Características:

Datos estructurados en tablas con esquema fijo.

Lenguaje SQL para consultas y manipulación de datos.

Transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).

Soporte para relaciones entre tablas.

Comandos Básicos en SQL:

Crear una tabla:

```
CREATE TABLE nombre_tabla (  
columna1 tipo_dato,  
columna2 tipo_dato,  
...  
);
```

Insertar datos:

```
INSERT INTO nombre_tabla (columna1, columna2, ...) VALUES (valor1, valor2, ...);
```

Consultar datos:

```
SELECT columnas FROM nombre_tabla WHERE condición;
```

Actualizar datos:

```
UPDATE nombre_tabla SET columna = nuevo_valor WHERE condición;
```

Eliminar datos:

```
DELETE FROM nombre_tabla WHERE condición;
```

Crear índices:

```
CREATE INDEX nombre_indice ON nombre_tabla (columna);
```

Modificar estructura de tabla:

```
ALTER TABLE nombre_tabla ADD columna_nueva tipo_dato;
```

Eliminar tabla:

```
DROP TABLE nombre_tabla;
```

Unir tablas (JOIN):

```
SELECT columnas FROM tabla1 JOIN tabla2 ON tabla1.columna = tabla2.columna;
```

Transacciones:

```
BEGIN; -- Iniciar transacción
```

```
-- ... Operaciones SQL ...
```

```
COMMIT; -- Confirmar cambios
```

```
ROLLBACK; -- Deshacer cambios
```

Base de Datos NoSQL:

Definición: Las bases de datos NoSQL son bases de datos no relacionales que almacenan datos en formatos flexibles y no requieren un esquema fijo.

Ejemplos de Bases de Datos NoSQL: MongoDB, Cassandra, Redis, CouchDB, Neo4j, etc.

Características:

Datos no estructurados o semiestructurados.

No requieren un lenguaje SQL para consultas.

Escalabilidad horizontal.

Flexibilidad en el esquema de datos.

La elección entre SQL y NoSQL depende de los requisitos específicos de un proyecto. SQL es ideal para aplicaciones con datos estructurados y relaciones complejas, mientras que NoSQL es adecuado para aplicaciones con datos no estructurados o semiestructurados y necesidades de escalabilidad.

2. Investigue y practique de manera local la creación y uso de una base de datos SQL por medio de Python/labVIEW. Link de referencia

4. Enunciado

Para este laboratorio se integrará el módulo de hardware entregado en el laboratorio pasado por medio de software para elevar su uso práctico. Entre las especificaciones del software, tiene que cumplir con las siguientes:

1. Tiene que utilizar el mismo *diseño* entregado en el laboratorio pasado para el hardware (incluyendo los dos canales y el ADC). Sin embargo se reconstruir dicho diseño ya sea en una placa perforada o impresa para mejorar la durabilidad del mismo.
2. Cada escala de tensión debe ser seleccionable por el usuario por medio del controlador embebido (Arduino/MyDAQ).
3. El controlador debe tener algún mecanismo de comunicación bidireccional a una computadora de uso personal. Algunos de los posibles métodos podrían ser: UART/Ethernet/WIFI.

4. La computadora debe ser capaz de indicarle al microcontrolador cual canal y/o escala se encuentra seleccionado por el usuario. En cualquier momento el usuario puede desactivar el uso de uno o ambos canales del sistema de adquisición de datos.
5. Se considera un canal apagado si este no se encuentra enviando muestras a la computadora/host.
6. El usuario debe tener una GUI en dónde se pueda asignar por canal: magnitudes físicas para los datos adquiridos por canal, conversión de tensión eléctrica a magnitud física, canal encendido y nivel de escala.
7. La GUI debe tener un modo de visualización (gráfico señal vs tiempo) para cuando un canal se encuentra encendido este debe estar actualizándose con las señales medidas. Debe ser posible visualizar ambos canales de manera simultánea y en el mismo gráfico.
8. El usuario debe ser capaz iniciar y almacenar en una base de datos SQL una grabación de 1024/2048 muestras con los siguientes atributos: autor, fecha, número de canal, unidades físicas y vector de datos.

4.1. Presentación de los resultados

1. Se deberá almacenar el código fuente del software que elaboré (microcontrolador y software del ordenador) en el repositorio del proyecto.
2. Debe elaborar una demostración del proyecto en dónde se pueda medir la temperatura (termocupla) y algún otro sensor de manera simultánea. Esta demostración se documentará por medio de un vídeo dónde se pueda observar el uso de la GUI, visualización de los datos y grabación.

5. Evaluación (100 %)

Rubro	Fecha de Entrega Límite	Valor (%)	Detalles
Defensa programada	9-10 de octubre	100	Documentación en repositorio, mediciones experimentales, tablas, fotografías, etc(40 %). Evaluación de cumplimiento de los requisitos indicados (30 %). Funcionalidad lograda por medio del vídeo (30 %). Extra: circuito construido en placa perforada/ circuito impreso (30 %)