**INFORME DE DESARROLLO Y ANÁLISIS DEL DESAFÍO**

Kewi Moya, Carlos Andrés Buelvas

Facultad de ingeniería, Universidad de Antioquia, Medellín

Informática II

Aníbal Guerra

17 de septiembre de 2024

Imagen que contiene Patrón de fondo

Descripción generada automáticamente

**INFORME DE DESARROLLO Y ANÁLISIS DEL DESAFÍO**

**Fecha: 11/09/2024**

En este día se llevó a cabo una reunión introductoria del proyecto para identificar los desafíos y problemas más significativos en el cumplimiento de los estándares mínimos. Durante la reunión, se destacaron tres problemas principales:

1. **Uso de la Memoria SRAM**:
   * La capacidad de memoria SRAM es de 2 Kbytes (2000 bytes). Esta memoria se dividirá en cinco segmentos, y un uso inadecuado podría llevar a la fragmentación o al colapso del programa si se alcanza su máxima capacidad. Es crucial diseñar el sistema de forma que se minimice la fragmentación, utilizando técnicas de gestión de memoria dinámica y asegurando que las variables se liberen adecuadamente después de su uso.
2. **Manejo de Recolección de Datos**:
   * Con un estándar de recolección de datos de 9600 BPS (1200 bytes por segundo), se genera una gran cantidad de datos en poco tiempo. Este flujo intenso de datos puede provocar que la memoria se llene rápidamente, lo que podría llevar a la pérdida de información o a errores en el procesamiento. Además, la cantidad de BPS es crucial para la precisión del programa, lo que implica que se deben implementar filtros y técnicas de compresión para optimizar el manejo de datos.
3. **Utilización de Datos**:
   * Existe una falta de conocimiento sobre Arduino y las características de las ondas, lo que dificulta cumplir con las funciones requeridas por el proyecto. Esta área requerirá investigación adicional, incluyendo el estudio de bibliotecas disponibles y la implementación de algoritmos que puedan procesar y analizar los datos de manera efectiva.

**Desafíos Identificados**

* **Limitaciones del Programa**: La capacidad del hardware y el software puede restringir la funcionalidad del proyecto, especialmente en términos de procesamiento de datos y velocidad de respuesta.
* **Limitaciones del Usuario**: Los usuarios finales pueden no tener el conocimiento técnico necesario para operar el sistema de manera efectiva, lo que puede llevar a malentendidos sobre las capacidades y limitaciones del programa.

**Acciones Decididas**

1. **Investigación**: Se llevará a cabo una investigación sobre fórmulas, conceptos y teoremas para abordar los problemas de recolección y utilización de datos. Esto incluirá el estudio de métodos de análisis de señales y técnicas de muestreo.
2. **Uso de Memoria Dinámica**: Se investigará y probará el uso de la memoria dinámica para manejar la gran cantidad de datos que recibirá el programa. Se explorarán estructuras de datos como listas enlazadas y arreglos dinámicos para mejorar la eficiencia en el uso de memoria.
3. **Pruebas de Limitaciones**: Se desarrollarán pruebas para identificar las limitaciones del proyecto y así crear un manual de usuario que aclare estas limitaciones. Esto incluirá la realización de pruebas de estrés para evaluar cómo el sistema maneja cargas de trabajo elevadas.

**Espacio de Reflexión de los Integrantes**

* **Kewi Moya (Integrante 1)**: Se destacó la necesidad de centrarse en la investigación antes de comenzar la codificación, debido a la falta de información útil. La investigación será clave para entender los conceptos fundamentales que guiarán el desarrollo del código.
* **Carlos Buelvas (Integrante 2)**: Se reconoció la complejidad del proyecto y la importancia de no iniciar la codificación sin un entendimiento claro de los conceptos necesarios. La discusión y el debate sobre los métodos a implementar serán esenciales para construir una base sólida para el código.

**Fecha: 12/09/2024**

En la segunda reunión, se discutieron métodos para la utilización de los datos. Se acordó que los datos se tomarían cada medio segundo, lo que podría requerir reescritura en la memoria si el usuario no los utiliza durante períodos prolongados. También se enfatizó la necesidad de investigar para escribir el código correctamente, dado que se requerirán nuevas declaraciones y acciones.

**Temas Abordados**

* **Frecuencia de Muestreo**: Se revisó la importancia de establecer un equilibrio entre la frecuencia de muestreo y la capacidad de procesamiento del sistema. Se considerará la posibilidad de ajustar la frecuencia según las necesidades del usuario.
* **Estrategias de Almacenamiento**: Se discutieron estrategias para almacenar datos en memoria, incluyendo el uso de buffers circulares para evitar la pérdida de datos y facilitar el manejo de la memoria.

**Espacio de Reflexión de los Integrantes**

* **Kewi Moya (Integrante 1)**: Después de esta reunión, solo puedo decir que hay vacíos que llenar en la información para poder completar el reto, así que la investigación será lo mejor. La implementación de técnicas de almacenamiento eficientes será crucial.
* **Carlos Buelvas (Integrante 2)**: En esta reunión me di cuenta de que aún falta mucho conocimiento por adquirir para que podamos avanzar de manera efectiva. Aunque discutimos varios métodos para procesar los datos, no logramos tomar una decisión concreta. La investigación será clave para entender mejor cómo implementar las nuevas declaraciones y acciones que ninguno de nosotros ha utilizado antes.

**Fecha: 14/09/2024**

Se decidió que la recolección de datos se realizará mediante dos listas dinámicas de 203 espacios. La recolección se llevará a cabo cada medio segundo, con un muestreo de 200 muestras por segundo. Este enfoque permitirá un manejo más flexible de los datos y minimizará el riesgo de saturar la memoria.

**Planificación de la Recolección de Datos**

* **Listas Dinámicas**: Se implementarán listas enlazadas para almacenar los datos de manera eficiente, permitiendo la adición y eliminación de elementos sin necesidad de redefinir el tamaño del arreglo.
* **Estrategias de Muestreo**: Se establecerán protocolos claros para la recolección de datos, asegurando que se tomen muestras representativas y se eviten errores de medición.

**Fechas: 15/09/2024 y 16/09/2024**

Después de la investigación, se tomó la decisión de comenzar a codificar. Aunque se reconocen las deficiencias en el conocimiento, esta decisión se considera necesaria para avanzar. La codificación se llevará a cabo en etapas, permitiendo iteraciones y mejoras continuas.

**Planteamiento del Código**

1. **Importación de Librerías**: Se comenzará importando las librerías necesarias para facilitar la escritura del código. Esto incluirá bibliotecas para la manipulación de datos y la comunicación serial.
2. **Declaración de Variables**: Se utilizarán variables en la memoria estática, incluyendo int, float y bool, con modificadores como short, unsigned y long para optimizar el uso de memoria. Se documentará cada variable para asegurar que su propósito sea claro.
3. **Función setup**: En esta sección se establecerá la conexión y la velocidad de transmisión (bps). Se configurarán los pines de entrada/salida y se inicializarán las variables necesarias.
4. **Función loop**:
   * **Pulsadores**: Se implementarán dos pulsadores. El primero comenzará la recolección de datos (200 datos por segundo) y almacenará los datos en listas semi-dinámicas de 103 posiciones. Esta elección se debe a la ineficiencia de usar memoria estática dada la alta frecuencia de cambio de datos.
   * **Tratamiento de Datos**: El segundo pulsador habilitará el tratamiento de datos, comenzando por identificar valores máximos y mínimos, seguido de la determinación de la amplitud y la frecuencia.

**Consideraciones Finales**

* **Documentación**: Se enfatizará la importancia de documentar el código a medida que se desarrolla, lo que facilitará futuras modificaciones y el mantenimiento del sistema.
* **Pruebas y Validación**: Se planificarán sesiones de prueba y validación para asegurar que el código funcione como se espera y cumpla con los requisitos del proyecto.
* **Revisión Continua**: Se establecerá un proceso de revisión continua para evaluar el progreso y realizar ajustes en el enfoque según sea necesario.

Este informe resume los desafíos y decisiones tomadas hasta ahora en el desarrollo del proyecto. La investigación y la planificación son fundamentales para abordar los problemas identificados y avanzar en el desarrollo del código. Se espera que el informe final se complete el 17/09/2024.

**Fecha: 17/09/2024**

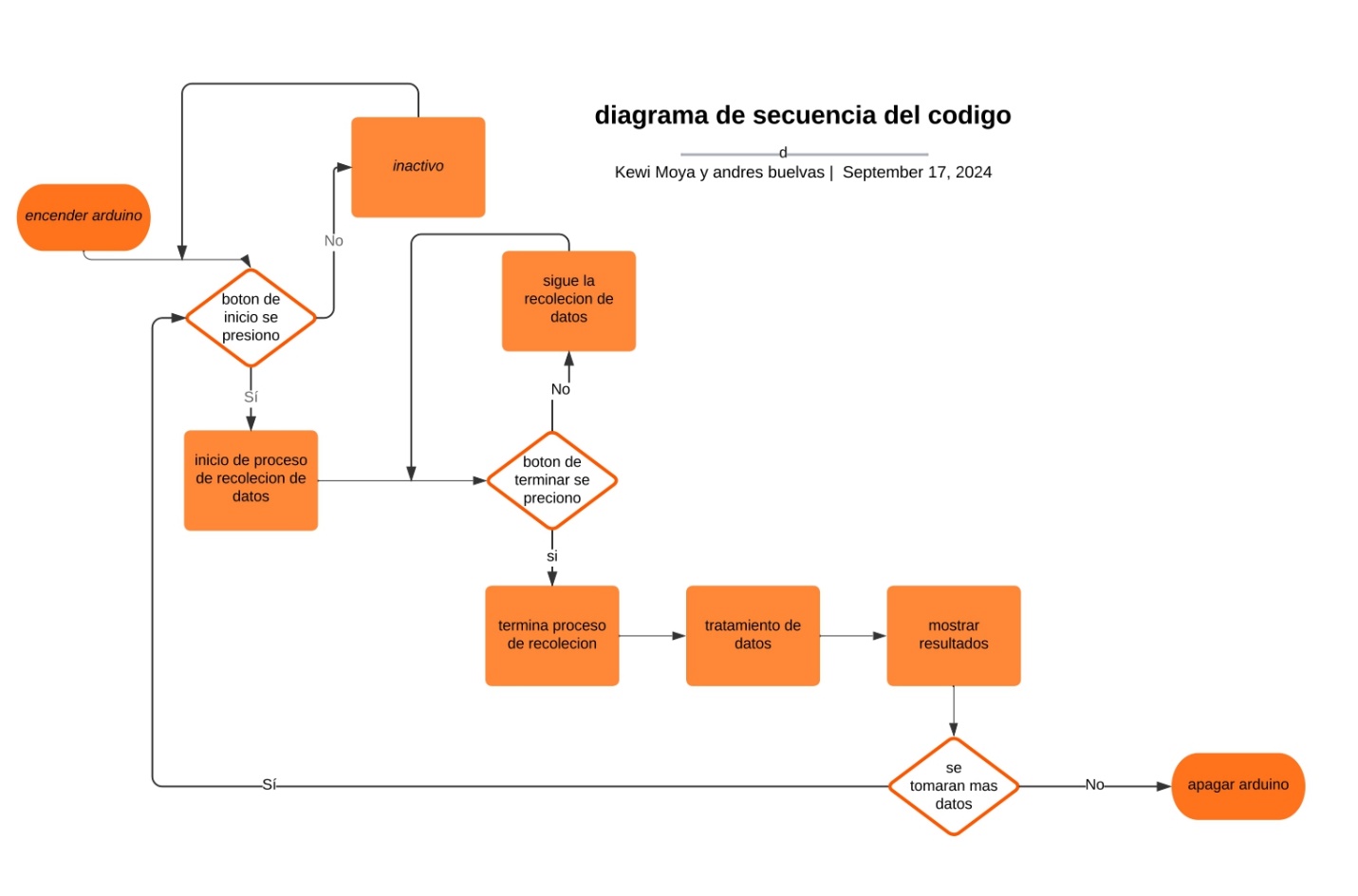
En la reunión de esta fecha se habló por las dificultades que se dio al codificar el programa del desafío, se pudo escribir el código, pero tiene errores semánticos relacionados al uso de memoria dinámica y le añadimos la inexperiencia con programación en Arduino, resulto que el código tiene problemas que no podemos descifrar o solucionar sin volver a escribir todo el código. Los errores más resaltantes fueron:

* Error de caracteres no legibles para el compilador y caracteres fantasmas, estos errores se repetían constantes en la parte de iniciaciones del código desde las líneas 24 a 188.
* Error de no cerrado de paréntesis y llaves, este error aparecía en las funciones que tenían arreglos dinámicos.
* Errores semánticos en el uso de la memoria dinámica, este error afectaba el código de una manera de herencia ya que el código utilizaba arreglos dinámicos en las funciones y en la toma de datos. Haciendo que la única solución seria solucionar el problema principal para que los demás se desbloqueara.

En conclusión, a lo largo del desafío en Tinkercad nos enfrentamos a múltiples obstáculos que nos impidieron completar la tarea de identificar el tipo de onda. Algunos de los inconvenientes más destacados fueron la falta de familiaridad con la programación en C++ y el uso del Arduino. Además, experimentamos dificultades técnicas en el manejo de los componentes y la simulación de circuitos en la plataforma.

Si bien no logramos culminar exitosamente el proyecto, el proceso nos permitió identificar las áreas en las que debemos mejorar, como el manejo de C++ y la correcta configuración de componentes en Arduino. Asimismo, adquirimos una mayor comprensión sobre la simulación de señales y la interacción con sensores.

En adelante, planeamos fortalecer nuestro conocimiento en estos aspectos mediante la práctica constante y la investigación sobre los fundamentos de Arduino y la programación en C++. Confiamos en que este enfoque nos permitirá enfrentar desafíos similares con mayor éxito en el futuro.

**Diagrama de flujo**