



Proyecto de interfaz de usuario

ESPECIFICACIONES DE PROTOCOLO DE COMUNICACION

AREA DE DISEÑO Y DESARROLLO

Tabla de Contenidos

Lista de Figuras.....	2
Lista de Tablas.....	2
Generalidades	3
Componentes de Interfaz Gráfica	4
Teclado de membrana	4
Pantalla LCD	5
Interacción incubadora – interfaz de usuario	6
Comunicación Incubadora – Interfaz	7
Protocolo de Comunicación.....	7
Acciones entre Incubadora e Interfaz.....	7
Formato para comunicación	8
Envío de parámetros de sensores.....	9
Envío de Peso.....	10
Alarmas	11
Envío de parámetro de referencia	12
Movimiento de Trendelemburg.....	13
Reporte de Tendencias y Alarmas	14
Despliegue de Tendencias solicitado desde pantalla	14
Reporte de tendencias	15
Reporte de Alarmas	16
Anexos.....	17
Implementación de verificación de redundancia cíclica (CRC)	17

Lista de Figuras

FIGURA 1. DIAGRAMA DE BLOQUES DE INTERFAZ DE USUARIO.	4
FIGURA 2. FRONT END TECLADO DE MEMBRANA PARA INTERFAZ DE USUARIO.....	4
FIGURA 3. DISPLAYS PROPUESTOS PARA LA UI. A) QBASS40YG1; B) HDSP-433G	5
FIGURA 4. MÓDULO LCD ULCD-70DT (GEN4) POR 4D SYSTEMS.	5
FIGURA 5. DIAGRAMA DE BLOQUES DE INTERFAZ DE USUARIO CON INCUBADORA.	6
FIGURA 6. PINOUT DE CONECTOR DE INTERFAZ DE USUARIO.	6
FIGURA 7. ALGORITMO PARA ENVÍO DE TRAMA. (VÍA DIA DIAGRAM EDITOR).....	8
FIGURA 8. ALGORITMO PARA RECEPCIÓN DE TRAMA. (VÍA DIA DIAGRAM EDITOR)	9
FIGURA 9. ALGORITMO PARA MEDICIÓN DE PESO EN INCUBADORA.	11
FIGURA 10. ESQUEMA PARA SELECCIÓN DE PACIENTE.	15
FIGURA 11. EJEMPLO DE ALGORITMO PARA VERIFICACIÓN DE REDUNDANCIA CÍCLICA (CRC) DE MENSAJE.....	17

Lista de Tablas

TABLA 1. PINOUT DE CONECTOR PROPUESTO PARA UI.	6
TABLA 2. DESCRIPCIÓN DE ACCIONES ENTRE INCUBADORA E INTERFAZ.....	7
TABLA 3. COMPONENTES DE LA ESTRUCTURA GENERAL DE LAS TRAMAS PARA COMUNICAR INCUBADORA E INTERFAZ.	8
TABLA 4. FORMATO DE TRAMA PARA ENVÍO DE MEDICIONES DE SENSORES.....	9
TABLA 5. POSICIÓN DE SENSORES EN TRAMA ENVIADA.	10
TABLA 6. COMANDOS UTILIZADOS EN EL PROCESO DE MEDICIÓN DE PESO.	10
TABLA 7. COMPONENTES DE TRAMAS PARA ENVÍO DE ALARMAS.	11
TABLA 8. LISTA DE ALARMAS A DESPLEGAR EN UI.	12
TABLA 9. CARACTERÍSTICAS DE ALARMA SEGÚN NIVEL DE GRAVEDAD.	12
TABLA 10. FORMATO PARA ENVÍO DE PARÁMETRO DE CONTROL.	12
TABLA 11. LISTA DE COMANDOS ASOCIADOS A SU PARÁMETRO DE CONTROL	13
TABLA 12. FORMATO PARA ENVÍO DE COMANDOS DE MOVIMIENTO DE ACTUADORES.....	13
TABLA 13. FORMATO PROPUESTO PARA GUARDADO DE ELEMENTOS.....	14
TABLA 14. BITS ASOCIADOS A ALARMAS EN COMPLEMENTO DE TRAMA AL [0-6].....	16
TABLA 15. INTERPRETACIÓN DE VALORES DE BITS ASOCIADOS A ALARMA.	17

Generalidades

El presente documento tiene la finalidad de detallar el principio de funcionamiento esperado por la empresa Neonacare S.A. de C.V. proyecto de desarrollo Interfaz Gráfica de Usuario a cargo del proveedor

El proyecto a desarrollar por el proveedor tendrá su implementación final como parte del proyecto de desarrollo de Incubadora Neonatal de Cuidados Generales desarrollado por Neonacare S.A. de C.V.

Cualquier aspecto de diseño no abarcado por este documento que tenga impacto en el proyecto final o en alguno de los entregables debe de hacerse saber a cualquiera de los siguientes contactos:

Nombre: José Everardo Pérez Limón
Puesto: Jefe de Producción, D&D
T. Fijo: (55) 2624 0858
T. Celular: (351) 121 2708
E-mail: jever@neonacare.com

Nombre: Jorge Enrique Carrillo López
Puesto: Jefe de Calidad
T. Fijo: (55) 26 24 0858
T. Celular: (55) 7399 0848
E-mail: jcarrillo@neonacare.com

De igual manera, se establece que el presente documento es un conjunto de lineamientos sugeridos por Neonacare, considerando las necesidades contempladas para su implementación prevista. Si por alguna razón, el proveedor considera que alguno de estos lineamientos puede modificarse, omitirse o incluir algún otro aspecto no previsto la empresa, pueden hacer llegar sus comentarios a los contactos antes mencionados para tratarlos de manera acorde.

El proyecto de Interfaz de usuario (UI) consiste en un desarrollo electrónico que formará parte de una incubadora de cuidados generales, la principal función del UI es fungir como puente entre el usuario final y el controlador principal del sistema, esta interfaz le permitirá al usuario interactuar con el sistema de forma que el usuario pueda configurar modos de funcionamiento, solicitar datos, habilitar y deshabilitar funciones entre otras cosas.

Los principales componentes que incluirá la interfaz de usuario se muestran en la Figura 1. Dicha interfaz desplegará información al usuario haciendo uso de una pantalla LCD con pantalla táctil, así como de displays LED numéricos (7 segmentos) y alfanuméricos (14 segmentos); la interfaz deberá interactuar con el usuario utilizando los botones insertados en una membrana o haciendo uso de la pantalla. La UI también deberá contar con la capacidad de:

- Almacenar datos para despliegue de tendencias de sensores.
- Capacidad de configurar y guardar la fecha y hora.
- Generar 3 alarmas sonoras de distinto tono haciendo uso de una bocina de al menos 1 Watt.

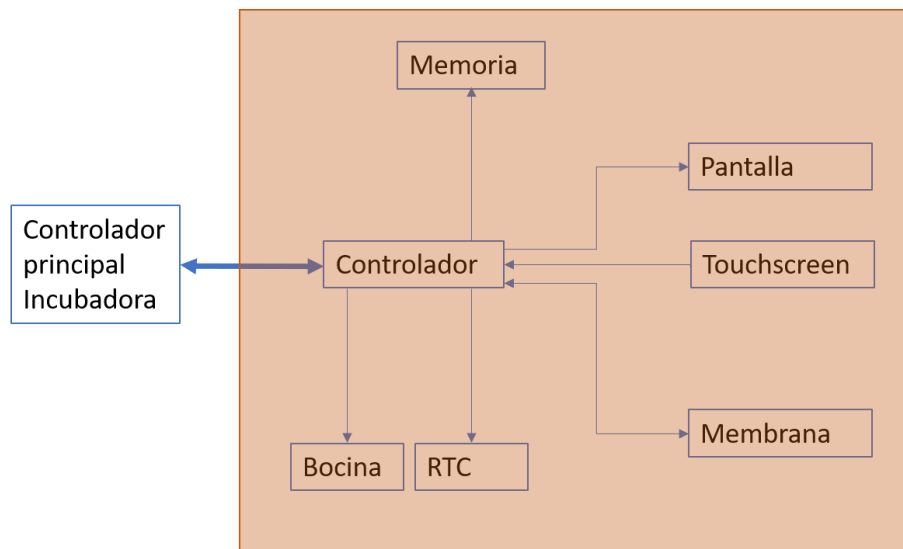


Figura 1. Diagrama de bloques de Interfaz de usuario.

Componentes de Interfaz Gráfica

Teclado de membrana

Uno de los métodos de entrada que tendrá a su disposición el usuario de la interfaz será un teclado de membrana (Figura 2). Este teclado está diseñado para ser usado como ‘front-end’ para el gabinete del control de la incubadora y cuenta con una abertura translúcida por la cual se puede visualizar e interactuar con la pantalla LCD. Cuenta con 12 botones NO (Normalmente abierto) unidos mediante un pin común, además la membrana contará con 11 indicadores LED de cátodo común.

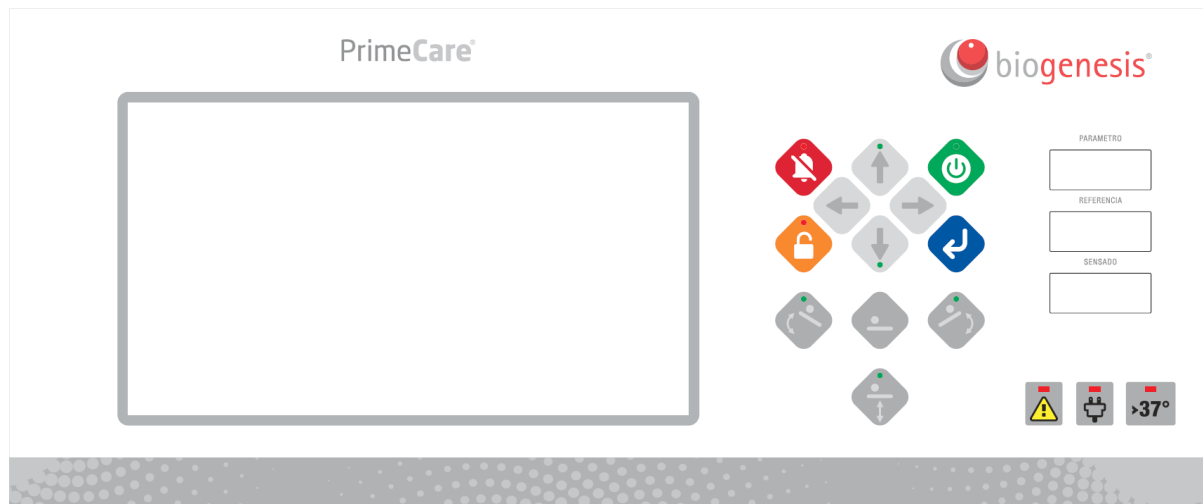


Figura 2. Front End teclado de membrana para interfaz de usuario.

Además, la membrana cuenta con ventanas adicionales en el extremo derecho para la visualización de displays numéricos y alfanuméricos tipo LED (Figura 3). Se espera que el sistema de la UI controle dichos displays como sigue:

- Información de parámetros (superior): Indica el parámetro que está siendo visualizado en los otros displays, estos pueden ser temperatura de aire (AIR), temperatura de piel 1 (P1), temperatura de piel 2 (P2), humedad relativa (%HR) y oxígeno (O2).
- Valor de referencia programado (medio): valor de referencia programado correspondiente al indicado por el display de información de parámetros.
- Valor real: Valor medido del parámetro indicado por los displays anteriores.

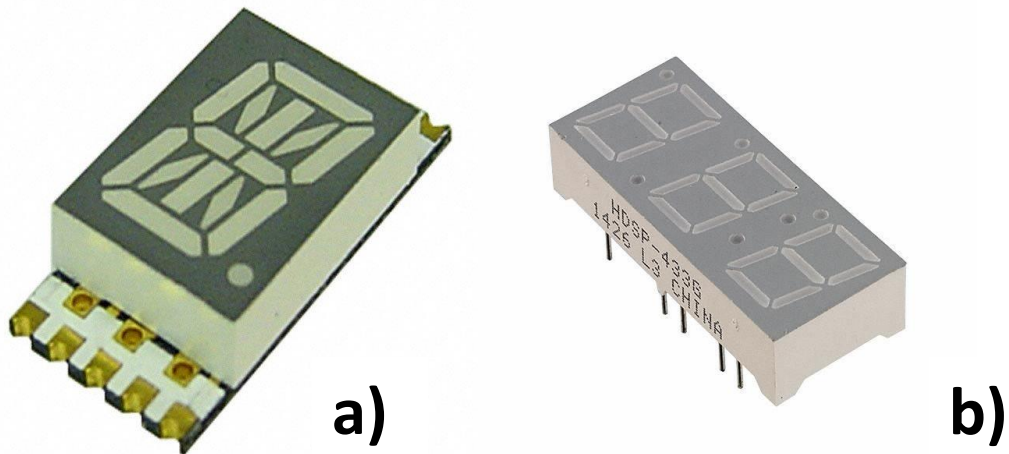


Figura 3. Displays propuestos para la UI. a) QBASS40YG1; b) HDSP-433G

Pantalla LCD

Otro método de entrada para el usuario será una pantalla LCD. Esta pantalla debe de tener capacidad touchscreen y debe tener una resolución mínima de 800x480 pixeles. La pantalla propuesta para utilizar es el módulo uLCD-70DT de la empresa 4D Systems (Figura 4).

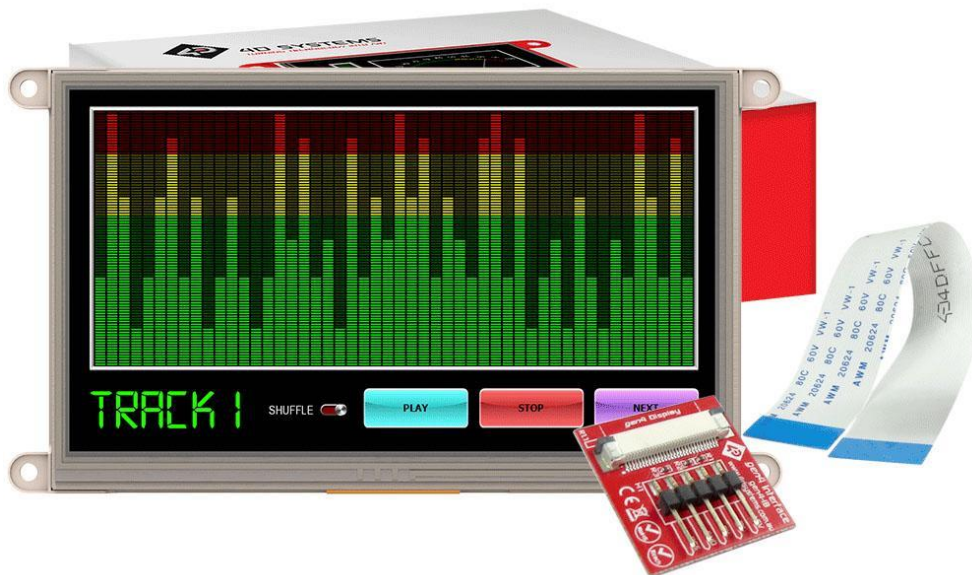


Figura 4. Módulo LCD uLCD-70DT (gen4) por 4D Systems.

Interacción incubadora – interfaz de usuario

El diseño del proveedor deberá contemplar su implementación en el proyecto de Neonacare considerando el diagrama de la Figura 5.

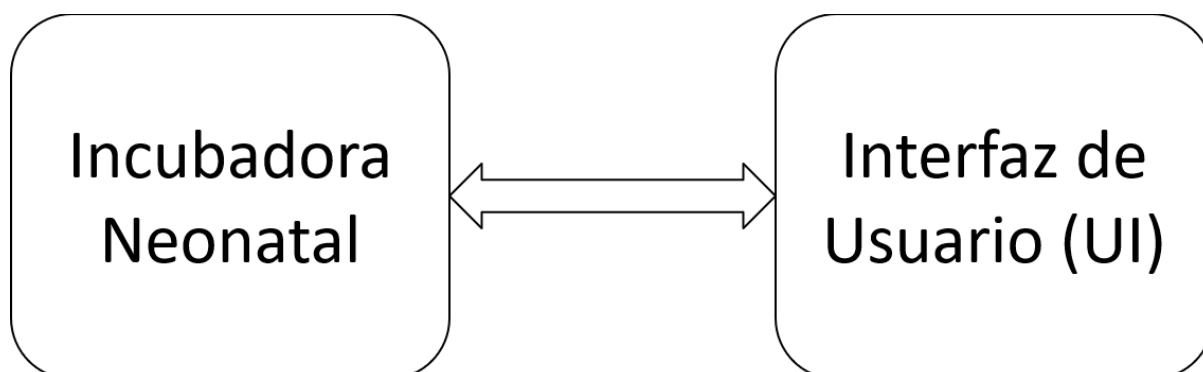


Figura 5. Diagrama de bloques de interfaz de usuario con incubadora.

El proyecto a desarrollar por el proveedor deberá trabajar de manera independiente al resto del sistema que contempla a la incubadora. La interfaz de usuario a desarrollar por el proveedor tendrá que interactuar con la incubadora a través de un conector y su respectivo arnés. Por medio conector se suministrará alimentación a la interfaz de usuario, se establecerá la vía de comunicación entre ambos sistemas, así como canales IO que comuniquen al controlador de la incubadora con algún subsistema específico de la interfaz.

El conector propuesto se observa en la Figura 6:

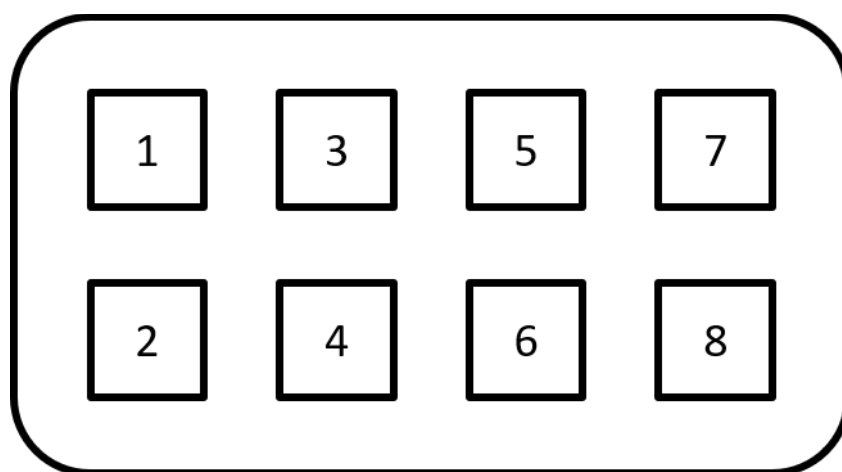


Figura 6. Pinout de conector de interfaz de usuario.

Se propone el uso de un conector cuadrado macho tipo header, ya sea tecnología THT o SMD, con separación de 100 mils entre pines y con el pinout descrito en la Tabla 1:

PIN	Nombre	Descripción.
2, 4	V _{DD}	Alimentación para sistema de UI.
6, 8	GND	Referencia.
1	F_LED	Conexión a ánodo de LED de falla eléctrica de membrana.
3	N.C.	Sin conexión eléctrica.
5	TX	Pin de transmisión de sistema de incubadora.
7	RX	Pin de recepción de sistema de incubadora.

Tabla 1. Pinout de conector propuesto para UI.

Para la alimentación de la interfaz de usuario se propone una alimentación de 5V. El pin F_LED forma parte de un subsistema de la incubadora que se encarga de indicar falla de alimentación a la incubadora. Este subsistema debe indicar con una alarma visible y sonora el corte repentino de alimentación al sistema; para la parte visible de la alarma se pretende utilizar uno de los pines de la membrana (P10 de LEDS) para generar la alarma visual. Se debe de rutear un camino desde el conector hasta dicho pin.

Comunicación Incubadora – Interfaz

Protocolo de Comunicación

Se propone para la comunicación entre el controlador de la incubadora y el controlador de la UI el uso del protocolo UART (“*Universal Asynchronous Receiver-Transmitter*”) con las siguientes características:

- 115200 baud/seg
- Paridad: no
- Ancho: 8 bits
- Stop bit: 1

Acciones entre Incubadora e Interfaz

Las acciones que provocarán las acciones entre sistemas están mencionadas de manera general en la Tabla 2:

Incubadora		Interfaz de Usuario
Enviar parámetros de sensores	⇒	Guardar y desplegar parámetros
Enviar señales de alarmas	⇒	Activa bocina; Despliega Alarma en pantalla y membrana
Solicita fecha y hora actual	⇒	Envía la hora y fecha actual
Recibe parámetro de referencia	⇐	Usuario determina parámetro de referencia y se envía a incubadora
Recibe comando/Envía comando (RS-232) para generar reporte de tendencias	↔	Usuario solicita reporte de tendencias/ Envía tendencias guardadas
Recibe comando de mov. Trendelemburg y activa actuadores.	⇐	Lee botones de activación de Trendelemburg/ Envía comando correspondiente

Tabla 2. Descripción de acciones entre incubadora e Interfaz.

Formato para comunicación

Se propone, para el correcto procesamiento de las tramas de información entre el controlador de la incubadora, el siguiente formato de trama:

[Inicio|Comm|Data[0 – n]|CRC|Fin]

Componente de trama	Descripción	Valor
<i>Inicio</i>	Inicio de Trama. Se usa para identificar el inicio de la trama	1 byte (0x02h)
<i>Comm</i>	Comando. Se utiliza para identificar la acción a realizar	1 byte (0x00 – 0xFFh)
<i>Data[0 – n]</i>	Datos. Si aplica, $n - 1$ datos a ser procesados según el comando	≥ 1 byte
<i>CRC</i>	Cyclic Redundancy Check. Comprobación de integridad de mensaje	2 bytes
<i>Fin</i>	Fin de Trama. Se usa para identificar el fin de la trama	1 byte (0x03h)

Tabla 3. Componentes de la estructura general de las tramas para comunicar Incubadora e Interfaz.

Se propone el uso del algoritmo mencionados en la Figura 7 y Figura 8 para la correcta recepción y procesamiento de tramas:

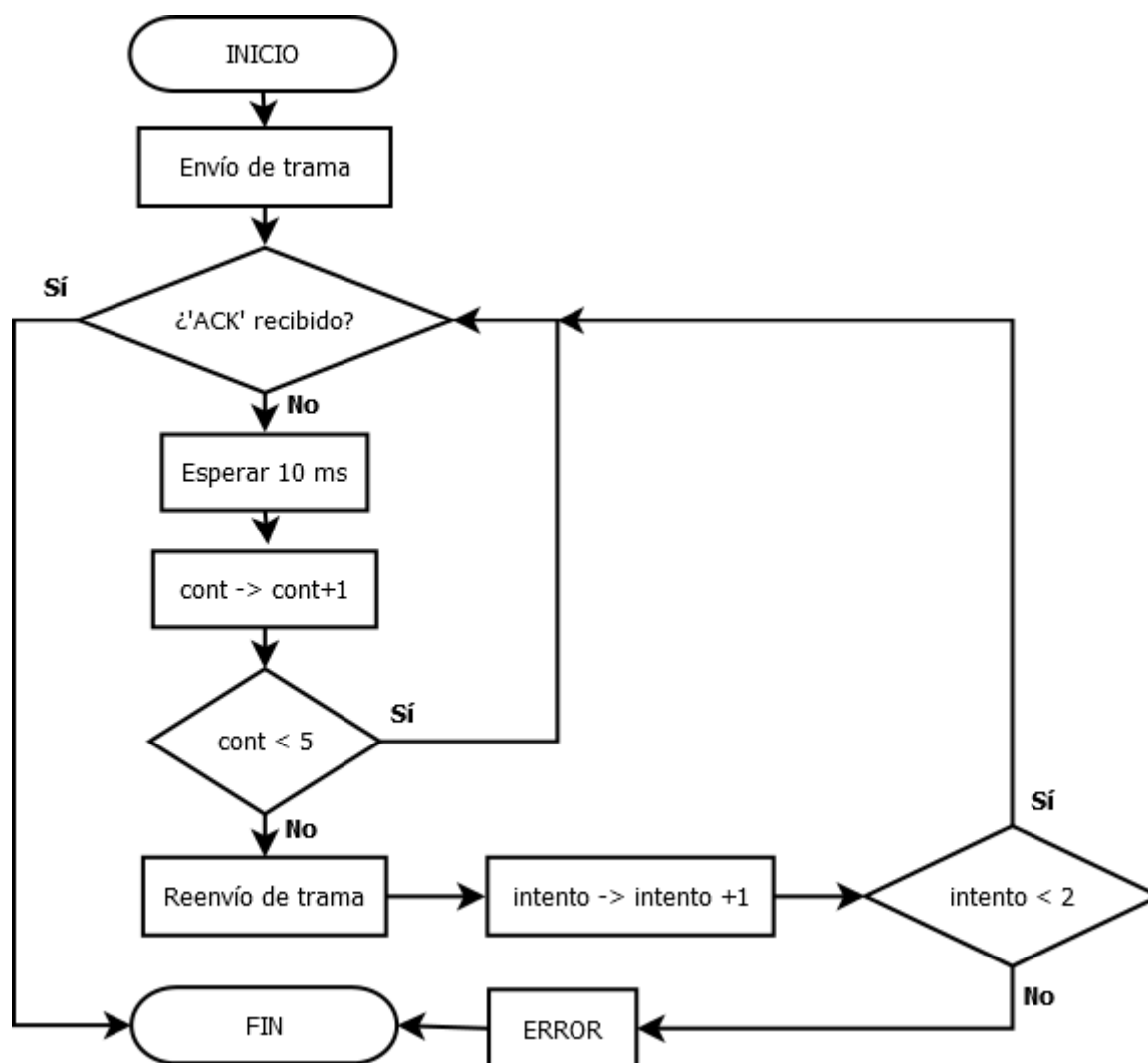


Figura 7. Algoritmo para envío de trama. (Vía Dia Diagram Editor)

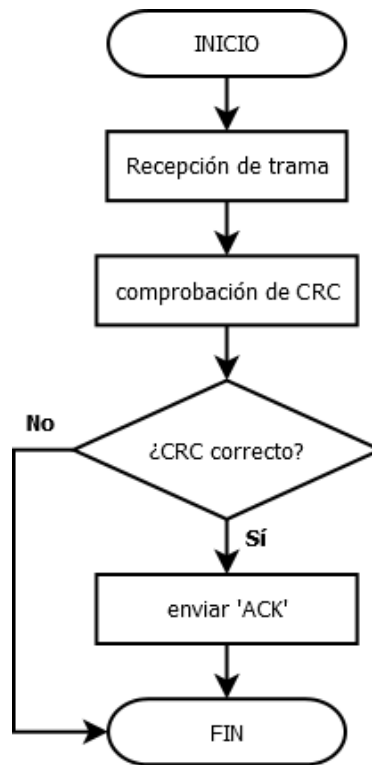


Figura 8. Algoritmo para recepción de trama. (Vía Dia Diagram Editor)

De acuerdo a los algoritmos mencionados, se requiere de una trama de confirmación a modo de “acknowledge” para indicar que la trama recibida es correcta.

También se hace mención de la comprobación de CRC para garantizar la integridad del paquete recibido. Para la generación del CRC se opta por utilizar el método de CRC-16/ARC con polinomio de la forma $x^{16} + x^{15} + x^2 + 1$ (0x8005h). Se anexará información ejemplo sobre cómo realizar dicha comprobación.

Envío de parámetros de sensores.

En esta sección se detallará la información sobre la forma propuesta de realizar el envío de datos provenientes de los sensores de la incubadora para su resguardo en memoria y despliegue por parte del controlador de la interfaz.

Se propone el envío de tramas de la siguiente manera:

$[Inicio|Comm|S_0|S_1|S_2|S_3|S_4|S_5|CRC|Fin]$

Componente de trama	Descripción	Valor
<i>Inicio</i>	Inicio de Trama. Se usa para identificar el inicio de la trama	1 byte (0x02h)
<i>Comm</i>	Comando. Se utiliza para identificar la acción a realizar	1 byte (0x01h)
$S_{[0-5]}$	Datos de sensores. Información referente al valor medido de cada parámetro.	6 × 3 bytes
<i>CRC</i>	Cyclic Redundancy Check. Comprobación de integridad de mensaje	2 bytes
<i>Fin</i>	Fin de Trama. Se usa para identificar el fin de la trama	1 byte (0x03h)

Tabla 4. Formato de trama para envío de mediciones de sensores.

Cada posición en la trama correspondiente a los sensores corresponderá siempre al mismo sensor. La correspondencia de sensores es como sigue:

Posición	Parámetro	Posición	Parámetro	Posición	Parámetro
S_0	Temperatura Aire	S_2	Temperatura Piel 2	S_4	Porcentaje Oxígeno
S_1	Temperatura Piel 1	S_3	Humedad Relativa	S_5	Porcentaje Potencia

Tabla 5. Posición de sensores en trama enviada.

A su vez, a cada valor de sensor le corresponde 3 bytes de información que se representan de la siguiente manera:

- Para sensores [0-2]:
 $S_n = [D_n|U_n|d_n]$
- Para sensores [3-5]:
 $S_n = [C_n|D_n|U_n]$

Para los primeros tres sensores, cada byte de información corresponderá a las decenas, unidades, y cifra decimal respectivamente, mientras que para los demás sensores los bytes de información corresponderán a los valores de centenas, decenas, y unidades correspondientes a cada sensor. Se propone que el sistema de la incubadora envíe lecturas de sensores cada 200 ms.

Envío de Peso

Se propone que la lectura de peso se realice a petición del usuario. Para una correcta lectura del peso se debe de realizar lecturas del peso del colchón junto con los accesorios u otros objetos dentro del capote que pudieran alterar la lectura de peso. Posteriormente se realiza la misma medición contemplando el peso del colchón más el peso del infante y finalmente se realiza una sustracción para determinar el peso del bebé. El algoritmo propuesto en la Figura 9 indica que el evento detonante de la medición será la solicitud por parte del usuario de la medición del peso desde la pantalla LCD. Una vez solicitada la medición, se enviará un comando al controlador de la incubadora para que realice la medición de peso (asumiendo que el infante se encuentra en la incubadora), una vez realizada la medición se enviará un comando a la interfaz para que le indique al usuario que debe levantar al infante.

Una vez levantado el bebé se le envía otro comando al controlador de la incubadora para que realice la segunda medición. Una vez obtenida la medición se realiza la sustracción para obtener el valor de peso del bebé y se envía a la interfaz para su posterior despliegue en la pantalla LCD.

Para el envío de comandos para la medición de peso se propone el envío de tramas con el siguiente formato:

$$[Inicio|Comm|CRC|Fin]$$

Donde la lista de comandos será la siguiente:

Comando	Descripción
0x02h	Inicio de medición de peso
0x03h	Confirmación de medición 1
0x04h	Confirmación de medición 2

Tabla 6. Comandos utilizados en el proceso de medición de peso.

Para el envío del peso se propone el siguiente formato, donde el comando es 0x05h:

$$[Inicio|Comm|U_w|d_w|c_w|CRC|Fin]$$

Donde los bytes para los datos corresponden a las unidades, decimas de unidad y centésimas de unidad (medición en kg).

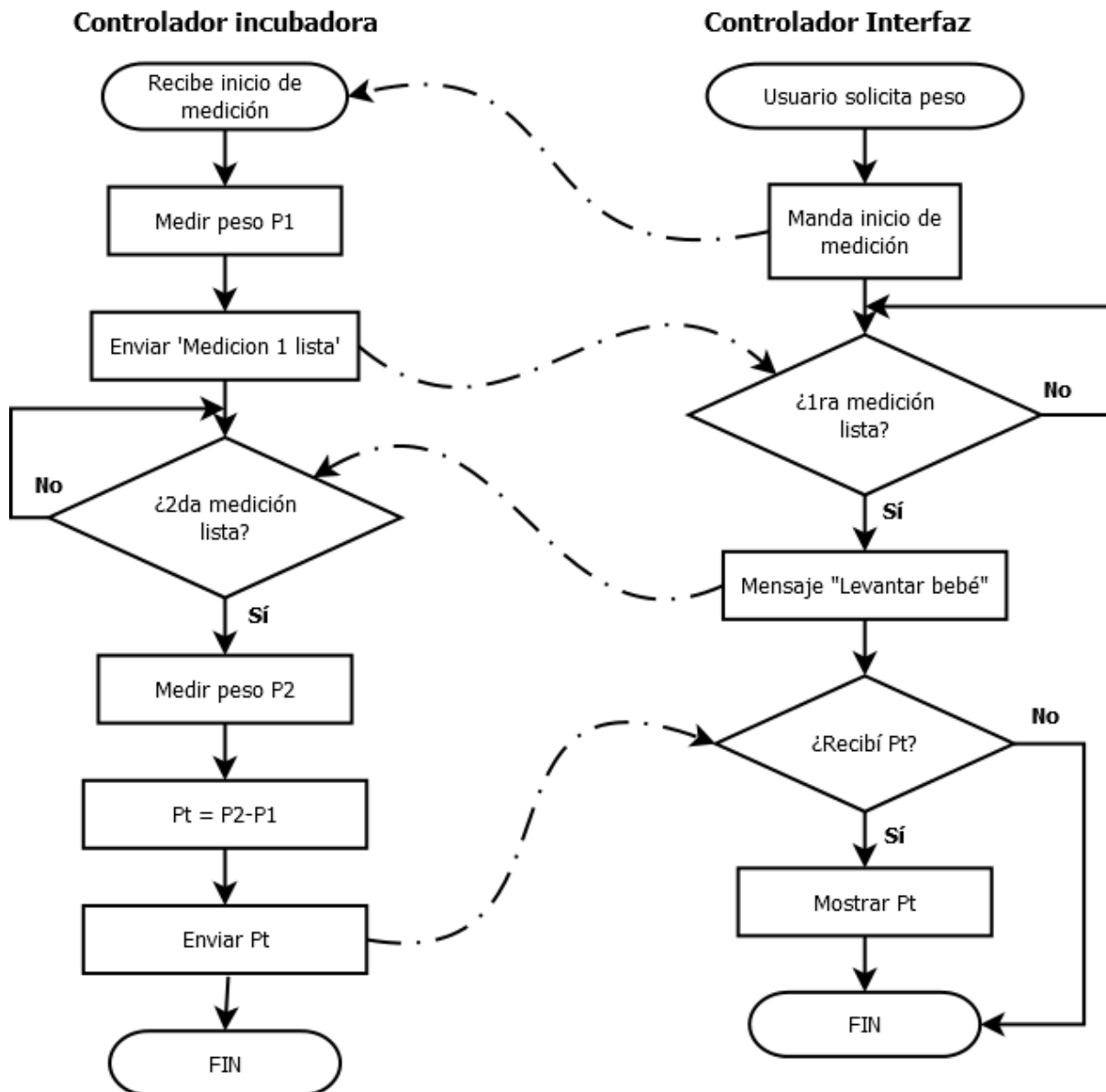


Figura 9. Algoritmo para medición de peso en incubadora.

Alarmas

Para el despliegue de las alarmas en la pantalla se propone el uso del siguiente formato de trama de la Tabla 7:

[Inicio|Comm|En|CRC|Fin]

Componente trama	Descripción	Valor
<i>Comm</i>	Comando. Comando asociado a una alarma específica	1 byte (0x06h – 0x01Ah)
<i>En</i>	Habilitador. Determina si la alarma correspondiente al comando está habilitada o deshabilitada.	1 byte (0x00h, 0x01h)

Tabla 7. Componentes de tramas para envío de alarmas.

Los comandos asociados a las alarmas son las siguientes:

Comando (Hex)	Descripción Alarma	Nivel
06	Falla del sistema	6
07	Falla de corriente	5
08	Temperatura del aire baja	3
09	Temperatura del aire alta	3
0A	Temperatura de paciente baja	3
0B	Temperatura de paciente alta	3
0C	Falla de sensor de piel	5
0D	Falla de circulación de aire	5
0E	Puerta abierta	2
0F	Filtro de aire expuesto	2
10	Falla de calefactor	6
11	Nivel de agua bajo	3
12	Humedad alta	3
13	Humedad baja	3
14	Oxígeno alto	3
15	Oxígeno bajo	3
16	Temperatura alta, calefactor deshabilitado	4
17	Cambio de celda de oxígeno	2
18	Sobrecalentamiento de calefactor	5
19	Falla en sensor de temperatura	3
1A	Falla de calefactor de humidificador	4

Tabla 8. Lista de alarmas a desplegar en UI.

A cada alarma de la Tabla 8 se le asocia un nivel, dicho nivel indica la gravedad de la alarma y cómo debe de ser el despliegue audiovisual (Tabla 9).

Nivel	Tonalidad	Volumen	Frecuencia	Silenciado en minutos
1	1	50 dB	600 Hz	No necesita silenciado
2	2	55 dB	600 Hz	15
3	3	60 dB	1500 Hz	10
4	4	70 dB	1500 Hz	5
5	5	80 dB	2500 Hz	2
6	Permanente	80 dB	2500 Hz	No se permite silenciar

Tabla 9. Características de alarma según nivel de gravedad.

Envío de parámetro de referencia

La UI envía al controlador de la incubadora el parámetro a controlar, así como el valor de control correspondiente. Para lo anterior se propone el uso del siguiente formato:

[Inicio|Comm|Val|CRC|Fin]

Componente trama	Descripción	Valor
Comm	Comando. Comando asociado a parámetro a controlar	1 byte (0x1Bh – 0x1Fh)
Val	Valor. Valor del parámetro de control	3 bytes

Tabla 10. Formato para envío de parámetro de control.

Comando (Hex)	Descripción
1B	Referencia de temperatura de aire
1C	Referencia de temperatura de piel 1
1D	Referencia de temperatura de piel 2
1E	Referencia de humedad relativa
1F	Referencia de porcentaje de oxígeno

Tabla 11. Lista de comandos asociados a su parámetro de control

Para la determinación de los bytes de la parte de valores se propone respetar el formato utilizado en la sección de “Envío de parámetros de sensores.

Movimiento de Trendelemburg

El sistema de la incubadora cuenta con 3 actuadores lineales, una columna y dos actuadores unidos a la cama de la incubadora para generar posiciones de trendelemburg y antitrendelemburg. Estas posiciones las determinará el usuario haciendo uso del teclado de membrana que forma parte de la UI.

Para lo anterior se propone el uso del formato de la Tabla 12:

[Inicio|Comm|CRC|Fin]

Donde la lista de comandos será la siguiente:

Comando (Hex)	Descripción
23	Subir trendelemburg
24	Bajar trendelemburg
25	Subir anti trendelemburg
26	Bajar anti trendelemburg
27	Subir Columna
28	Bajar Columna
29	Nivelar Trendelemburg a 0

Tabla 12. Formato para envío de comandos de movimiento de actuadores.

Reporte de Tendencias y Alarmas

Una característica que forma parte de los requerimientos para el funcionamiento de la incubadora, el sistema debe contar con capacidad para generar reportes de tendencias de los sensores medidos y reportes de alarmas, la solicitud de dichos reportes se puede generar desde la pantalla de tendencias o desde el puerto RS232 de la incubadora.

En esta sección se describirán de manera detallada las secciones propuestas para poder generar dichos reportes, sugerencias de como almacenar la información y cómo procesarla según el tipo y generación de reporte.

Las tendencias y alarmas deberán ser guardadas en un dispositivo de memoria seleccionado por el proveedor. Se propone el siguiente método de guardado tanto de alarmas como de parámetros enviados a la pantalla en un solo arreglo (Tabla 13):

$$[\#Medicion|Fecha|Hora|S_0|S_1|S_2|S_3|S_4|S_5|ID_{paciente}|ID_{usuario}|Nombre_{usuario}|Al_{[0-6]}]$$

Componente Trama	Descripción	Tamaño (byte)
$\#Medicion$	Numero de medición (Formato: 0h – 1517Fh)	3
$Fecha$	Fecha de la medición (Formato: ddmmaa; ASCII)	6
$Hora$	Hora de la medición (Formato: hhmmss; ASCII)	
$[S_0 - S_5]$	Valores de sensores (Formato como en sección de Envío de parámetros de sensores.)	6×3
$ID_{paciente}$	Identificador de paciente (sistema) (Formato 0h – 3Fh)	1
$ID_{usuario}$	Identificador de paciente (usuario) (ASCII)	4
$Nombre_{usuario}$	Nombre de paciente ingresado por usuario (ASCII)	16
$Al_{[0-3]}$	Byte para resguardo de habilitación de alarmas	7

Tabla 13. Formato propuesto para guardado de elementos.

El arreglo anterior se propone sea guardado, con los valores disponibles, cada 30 segundos. Se calcula que la cantidad de datos que pueda llegar a almacenar en 30 días a una tasa de 1 medición/30 segundos es de 86400 datos. Para un mejor orden de los arreglos guardados se propone una clasificación de los datos a través de un identificador de paciente designado por el sistema de la interfaz para su posterior despliegue durante la selección de la pantalla. Este método de clasificación se discutirá más adelante.

Despliegue de Tendencias solicitado desde pantalla

Este método de despliegue de tendencias se realizará cuando el usuario ingrese al menú de tendencias. En este menú el usuario verá una lista de los registros guardados en la memoria, cada elemento de la lista deberá indicar lo siguiente:

- Nombre de paciente (si se ingresó)
- Fecha de ingreso

Para asegurar la distinción entre pacientes se propone que al alimentar el sistema de la interfaz una de las primeras ventanas en desplegarse sea una ventana donde se le pregunta al usuario si quiere ingresar un nuevo paciente o continuar con el anterior, además de una opción para cancelar la operación. En caso de ingresar un paciente nuevo, incrementará el ID del paciente y el paciente indicará el nuevo nombre del paciente para posteriormente reanudar la operación normal de la interfaz. Si el usuario quiere continuar con un paciente anterior se mantiene el ID y se siguen registrando los datos con el paciente anterior (Figura 10).

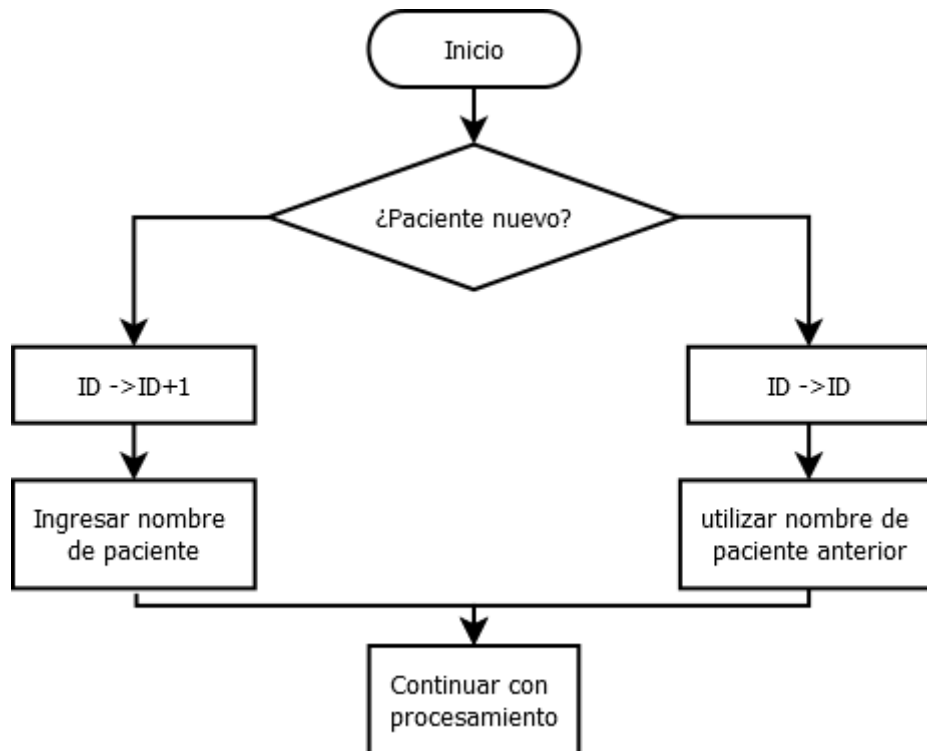


Figura 10. Esquema para selección de paciente.

Cuando el usuario selecciona las tendencias del paciente que quiere visualizar, el sistema de la interfaz deberá identificar los datos que contengan estos ID. Se propone también que del total de componentes se seleccionen una muestra de datos con un espacio uniforme de manera que se tenga una cantidad de muestras suficientes para poder generar un gráfico. Una vez seleccionados los datos, el usuario visualizará un gráfico vacío con botones estilo “toggle” relacionados a los parámetros de los sensores a visualizar de tal manera que al habilitar los botones toggle se superpondrán las gráficas de los parámetros obtenidos.

Reporte de tendencias

El reporte de tendencias de la incubadora regresará todos los datos almacenados en la memoria del sistema de la interfaz y la solicitud de dicho reporte puede venir de dos fuentes:

- Puerto RS-232 del sistema de la incubadora.
- Menú de reportes ubicado en la sección de servicio de la incubadora.

Cuando la solicitud del reporte de tendencias proviene del puerto RS-232 del sistema de la incubadora, esta enviará un comando de inicio de reporte al sistema de interfaz con el formato descrito a continuación:

[Inicio|Comm|CRC|Fin]

Donde el comando de inicio/envío de tendencias es 1Fh. La pantalla, al recibir este comando, interrumpirá su operación normal (se propone utilizar alguna pantalla que con una leyenda de “cargando”) y empieza a mandar tramas con los datos de sensores guardados con el formato mostrado a continuación:

[Inicio|Comm|#Medicion|Fecha|Hora|S₀|S₁|S₂|S₃|S₄|S₅|S₆|ID_{paciente}|CRC|Fin]

Cada componte de la trama se procesa de la misma manera que al inicio de esta sección (Reporte de Tendencias y Alarmas).

Aun se espera definir y acordar con el proveedor si se enviarán las tramas a una velocidad fija o si se implementará otro tipo algoritmo de comunicación.

Cuando la pantalla finalice de enviar datos, esta enviará el comando de finalización de reporte de la siguiente manera:

[Inicio|Comm|CRC|Fin]

Siendo el comando de finalización 21h. Una vez enviado el comando, la UI regresa a su funcionamiento normal. De la misma manera, el sistema de la incubadora reanudará operaciones normales.

Cuando la solicitud de reporte de tendencias se realiza desde la pantalla, la pantalla enviará el comando de inicio de reporte de tendencias y posteriormente suspenderá actividades. El sistema de la incubadora, al recibir el comando, suspenderá actividades normales y recibirá las tramas de reporte de manera similar a la descrita anteriormente.

Reporte de Alarmas

Otra función con la que deberá contar el sistema de la incubadora será la capacidad de generar un reporte de alarmas. Dicho reporte de alarmas enviará al sistema de la incubadora los eventos guardados donde una alarma haya sido activada, para su posterior envío a través de puerto RS-232

. Con la intención de simplificar el procesamiento del reporte de alarmas se propone utilizar un esquema similar al tratado en la sección anterior (Reporte de tendencias).

La solicitud de reporte de alarmas se podrá generar como se menciona antes. Puede ser generado desde el puerto RS-232 del sistema de la incubadora o desde el menú correspondiente de reporte de alarma.

El formato para el envío de comando de inicio es similar al de reporte de alarmas, que sería el siguiente:

[Inicio|Comm|CRC|Fin]

Donde el comando de inicio/envío de reporte de alarmas es 20h. A su vez, el formato correspondiente en el que se enviarán las tramas de reporte será como sigue:

[Inicio|Comm|#Medicion|Fecha|Hora|Al_[0-6]|CRC|Fin]

El componente de trama Al_[0-3] está compuesto por 7 bytes que se pueden interpretar como 56 bits. Cada 2 bits estarán asociados a una de las alarmas mencionadas en la sección Alarmas (Tabla 8). Se asocia la alarma del numero de comando más bajo con el par de bits menos significativo como se muestra en la Tabla 14.

Byte 6								Byte 5				Byte 4			
X	X	X	X	X	X	X	X	AI 24	AI 23	AI 22	AI 21	AI 20	AI 19	AI 18	AI 17
B55	B54	B53	B52	B51	B50	B49	B48	B47	B46	B45	B44	B43	B42	B41	B40
Byte 3				Byte 2				Byte 1				Byte 0			
AI 15	AI 14	AI 13	AI 12	AI 11	AI 10	AI 09	AI 08	AI 07	AI 06	AI 05	AI 04	AI 03	AI 02	AI 01	AI 00
B31	B30	B29	B28	B27	B26	B25	B24	B23	B22	B21	B20	B19	B18	B17	B16
B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0

Tabla 14. Bits asociados a alarmas en complemento de trama AI [0-6].

Donde se tendrá los siguientes valores:

B0	B1	Estado de alarma
1	X	ERROR
0	1	Alarma Habilitada
0	0	Alarma Deshabilitada

Tabla 15. Interpretación de valores de bits asociados a alarma.

El sistema de la interfaz, al momento de procesar las tramas guardadas, detectará las mediciones donde no se ha habilitado ninguna alarma y las descarta para su envío. El resto de consideraciones para el envío de reportes se seguirá como se menciona en la sección Reporte de tendencias.

Anexos

Implementación de verificación de redundancia cíclica (CRC)

Se utilizó el documento [“A PAINLESS GUIDE TO CRC ERROR DETECTION ALGORITHMS”](#) por Ross N. Williams como referencia para la generación de un algoritmo de comprobación de CRC. Para el algoritmo presentado en la se puede comprobar sus resultados en el siguiente [calculador de crc en línea](#).

```

41
42 #define CRC16 0x8005
43 #include "crc.h"
44
45 byte message[] = { "que onda raza!" };
46 byte *p;
47 uint16_t crc_ex;
48 /* ==uint16_t gen_crc16(byte *data, uint16_t size)==
49 * Método que realiza verificación de redundancia cíclica del mensaje asociado al puntero '*data' de tamaño 'size'
50 * -PARAMETROS-
51 * byte *data. Puntero hacia el primer byte del mensaje a revisar.
52 * uint16_t size. Tamaño del mensaje a leer.
53 */
54 uint16_t gen_crc16(byte *data, uint16_t size) {
55     uint16_t out = 0, crc;
56     int bits_read = 0, bit_flag, i, j;
57     while (size > 0) { // Recorrido de todos los bits del mensaje. (Empezando por LSB)
58         bit_flag = out >> 15;
59         out <<= 1;
60         out |= (*data >> bits_read) & 1;
61         bits_read++;
62         if (bits_read > 7) {
63             bits_read = 0;
64             data++;
65             size--;
66         }
67         if (bit_flag) { // Cuando al inicio del buffer de CRC hay un uno, se aplica
68             out ^= CRC16; // operación XOR a buffer.
69         }
70     }
71     for (i = 0; i < 16; i++) { // Se repite operación con 16 bits extra (correspondientes a
72         bit_flag = out >> 15; // los bits agregados que se mencionan en la sección 6 del
73         out <<= 1; // documento 'A PAINLESS GUIDE TO CRC ERROR DETECTION ALGORITHMS'
74         if (bit_flag) // por Ross N. Williams.
75             out ^= CRC16;
76     }
77     i = 0x8000;
78     j = 0x0001;
79     for (; i != 0; i >>= 1, j <<= 1) { // Para ajustarse al formato del CRC/ARC se debe cambiar la posición
80         if (i & out) // de los bits del resultados de CRC.
81             crc |= j;
82     }
83     return crc;
84 }
85
86 void crc_implementation_test() {
87     p = &message[0];
88     crc_ex = gen_crc16(p, 14);
89
90 }
91

```

Figura 11. Ejemplo de algoritmo para verificación de redundancia cíclica (crc) de mensaje.