

UNIVERSIDAD RICARDO PALMA
FACULTAD DE INGENIERÍA
PROGRAMA DE TITULACION POR TESIS
ESCUELA PROFESIONAL DE INGENIERÍA MECATRÓNICA



**DISEÑO DE UNA RED INALÁMBRICA PARA EL SERVICIO DE
PESAJE EN EL SECTOR AGROINDUSTRIAL UTILIZANDO
BALANZAS SUPER SS MEDIANTE PROTOCOLO SPI BASADO EN
MICROCONTROLADOR ESP32**

TESIS
PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERA MECATRÓNICA

PRESENTADA POR

Bach. BRYAN EMMANUEL CORPUS CHIRINOS

Bach. ANA MARÍA PORTUGAL AUCCASI

Asesor: Dr. Ing. SOTELO VALER FREEDY

LIMA – PERÚ

2021

DEDICATORIA

Le dedico el presente trabajo a Dios, mis padres por su apoyo y gran dedicación, a mis queridos hermanos por acompañarme siempre. A Mies y Zaha.

Ana María Portugal Auccasi

Le dedico este trabajo a Dios, a mis padres por su apoyo incondicional, a mi hermano y mis muy queridos abuelos, por su apoyo motivacional. Al futuro lector ingeniero que valore este humilde trabajo de tesis que aporta en su investigación.

Bryan Emmanuel Corpus Chirinos

AGRADECIMIENTO

Nuestro sincero agradecimiento a nuestra alma mater, a nuestros profesores por sus enseñanzas brindadas en cada clase y nuestros asesores por su constante apoyo y dirección en el desarrollo de nuestra investigación.

Ana Portugal y Bryan Corpus

ÍNDICE GENERAL

RESUMEN	i
ABSTRACT	ii
INTRODUCCIÓN	iii
CAPÍTULO I: PLANTEAMIENTO Y DELIMITACIÓN DEL PROBLEMA	1
1.1 Descripción y delimitación del problema	1
1.2 Formulación y delimitación del problema	3
1.2.1 Problema General	3
1.2.2 Problema Específicos	3
1.3 Importancia y justificación del estudio	3
1.3.1 Importancia del estudio	3
1.3.2 Justificación Tecnológica	3
1.3.3 Justificación Económica	4
1.4 Limitación del estudio	4
1.5 Objetivos	4
1.5.1 Objetivo General	4
1.5.2 Objetivo específico	4
CAPÍTULO II: MARCO TEÓRICO	5
2.1 Antecedentes del estudio de investigación	5
2.1.1 Antecedentes internacionales	5
2.1.2 Antecedentes Nacionales	6
2.2 Bases teóricas vinculadas a la variable o variables de estudio	7
2.2.1 Agroindustria	7
2.2.2 Balanzas	9
2.2.3 SolidWorks	15
2.2.4 Software Proteus	16
2.2.5 Sistema Electrónico	17
2.2.6 Microcontroladores	19
2.2.7 Microcontrolador ESP32	19
2.2.8 Protocolo SPI	20

2.2.9 Sistema HMI.....	25
2.2.10 Pantalla Nextion NX4832T035	26
2.2.11 Editor Nextion	27
2.2.12 Software de Aplicaciones	28
2.2.13 LABVIEW	29
2.2.14 SQL Server	31
2.2.15 Redes Inalámbricas.....	32
2.2.16 Modos de redes inalámbricas.....	33
2.2.17 Tecnología Wifi.....	34
2.2.18 Características resaltantes de las redes inalámbricas.....	34
CAPÍTULO III: DESARROLLO DEL DISEÑO DE LA RED INALÁMBRICA PARA LAS BALANZAS SUPER SS	36
3.1 Diagrama de conectividad general de la red inalámbrica de todo el proyecto	37
3.2 Diseño del sistema electrónico	38
3.2.1 Requerimiento para la selección del microcontrolador	38
3.2.2 Diseño del circuito de conexión con la tarjeta.....	39
3.2.3 Conexionado electrónico	43
3.2.4 Programación General Rutina de la tarjeta Electrónica.....	43
3.2.5 Diagrama de flujo del funcionamiento del sistema electrónico.....	43
3.2.6 Pantalla para Interfaz usuario máquina.....	44
3.2.7 Configuración de la pantalla Nextion	45
3.2.8 Desarrollo de la programación del microcontrolador ESP32	48
3.2.9 Costo del sistema electrónico	51
3.3 Desarrollo de software del sistema y base de datos.....	52
3.3.1 Selección de software de aplicativo.....	52
3.3.2 Desarrollo del sistema de pesaje usando LabVIEW	53
3.3.3 Desarrollo de la base de datos mediante SQL SERVER.....	59
3.4 Diseño de la carcasa.....	61
3.5 Implementación	62
CAPÍTULO IV: PRESENTACIÓN Y ANÁLISIS DE RESULTADOS	63
4.1 Conectividad inalámbrica entre los dispositivos	63

4.2 Sistema electrónico	64
4.3 Software de gestión y base de datos	67
CONCLUSIONES	69
RECOMENDACIONES	70
REFERENCIAS BIBLIOGRAFICAS	72
ANEXOS	75
Anexo 1: Manual Técnico de la balanza SUPER SS.....	75
Anexo 2: Manual Técnico Microcontrolador ESP32	80
Anexo 3: Manual Técnico de la pantalla Nextion	83
Anexo 4: Manual Técnico de chip FD628.....	86
Anexo 5: Código del Programa del microcontrolador ESP32.....	91
Anexo 6: Código del software de gestión elaborado en labview.....	118
Anexo 7: Código de base de datos SQL SERVER.....	120

ÍNDICE DE TABLAS

Tabla N°1: Características de las Galgas Metálicas y Semiconductoras.....	11
Tabla N°2: Lista de materiales galgas extensiométricas.	11
Tabla N°3: Característica de la balanza modelo SUPER SS.....	14
Tabla N°4: Cuadro Comparativo ESP32 vs Arduino UNO.....	20
Tabla N°5: Señales del SPI.....	21
Tabla N°6: Características de las versiones de pantalla Nextion.....	26
Tabla N°7: Características de SQL Server	31
Tabla N°8: Microcontroladores Posibles a usar	38
Tabla N°9: HMI Nextion posibles a utilizar.....	45
Tabla N°10: Tabla de costo	51
Tabla N°11: Tabla de comparación de costos entre balanza similares.....	52
Tabla N°12: Software posibles a usar.....	52

ÍNDICE DE FIGURAS

Figura N°1: Venta, mantenimiento y repuestos de balanzas industriales en Perú.....	2
Figura N°2: Crecimiento histórico en la agroexportación	8
Figura N°3: Evolución de exportaciones	8
Figura N°4: Celda de Carga.....	11
Figura N°5: Puente Wheatstone.....	12
Figura N°6: Balanza SUPER SS de 15 Kg.....	14
Figura N°7: Logo de SolidWorks	15
Figura N°8: Circuito de comunicación	16
Figura N°9: Software Proteus	17
Figura N°10: Sistema electrónico	17
Figura N°11: Sistema electrónico	18
Figura N°12: Diagrama que compone un microcontrolador.	19
Figura N°13: Protocolo SPI.....	21
Figura N°14: Presentación de transmisión SPI.....	22
Figura N°15: Opción 1 de transmisión SPI	24
Figura N°16: Opción 2 de transmisión SPI	24
Figura N°17: Opción 3 de transmisión SPI	25
Figura N°18: Opción 4 de transmisión SPI	25
Figura N°19: Software Nextion Editor.....	27
Figura N°20: Ejemplo de Pantalla Frontal de LabVIEW	30
Figura N°21: Redes inalámbricas	32
Figura N°22: Diagrama de flujo general	37
Figura N°23: Diagrama referencial de toda la red de las balanzas	38
Figura N°24: Esquema electrónico de la tarjeta desarrollado en Proteus.....	40
Figura N°25: Diseño de PCB de la tarjeta desarrollado en Proteus.....	41
Figura N°26: Vista 3D de la tarjeta realizado desarrollado en Proteus	41
Figura N°27: Tarjeta electrónica en físico.....	42
Figura N°28: Simulador Hyperterminal	42

Figura N°29: Diagrama electrónico.....	43
Figura N°30: Diagrama de flujo del programa ESP32	44
Figura N°31: Elaboración de la pantalla desarrollado en Nextion Editor	46
Figura N°32: Cargo de la pantalla Nextion desarrollado en Nextion Editor	47
Figura N°33: Pantalla Nextion Física	47
Figura N°34: Declaración de librerías	48
Figura N°36: Obtención de cada dígito en formato binario.....	49
Figura N°37: Conversión de dígito binario a decimal	50
Figura N°38: Habilidad de la antena wifi del ESP32	50
Figura N°39: obtención de peso y envío de información	51
Figura N°40: Aplicativo LabVIEW visualización de pesaje	54
Figura N°41: Configuraciones de la pantalla realizado en LabVIEW.....	55
Figura N°42: Modificación de las configuraciones de la pantalla en LabVIEW	55
Figura N°43: Envío de trama del software desarrollado en LabVIEW hacia simulador.....	56
Figura N°44: Modificación de las configuraciones de la pantalla realizado en LabVIEW.....	57
Figura N°45: Lectura de TCP de la trama de las tarjetas electrónicas en LabVIEW	57
Figura N°46: Modificación de las configuraciones de la pantalla en LabVIEW	58
Figura N°49: Ingreso del SQL Server.....	59
Figura N°50: Autenticación de SQL Server	60
Figura N°51: Habilidad del TCP/IP y puerto de comunicación	60
Figura N°52: Base de datos Super SS.....	61
Figura N°53: Diseño de la carcasa de la tarjeta electrónica realizado en SolidWorks	62
Figura N°54: Estructura interna de la balanza SUPER SS	62
Figura N°55: Conectividad de la tarjeta electrónica SUPERSS	63
Figura N°56: Ping de tarjeta electrónica a PC	64
Figura N°57: Captura de peso en tarjeta electrónica	64
Figura N°58: Pesaje automático vs pesaje manual	65
Figura N°59: Pantalla indicando que falta pesaje para llegar al objetivo.....	66
Figura N°60: Pantalla indicando que sobra pesaje para llegar al objetivo	66
Figura N°61: Pantalla indicando que sobra pesaje para llegar al objetivo	67
Figura N°62: Pantalla indicando que sobra pesaje para llegar al objetivo	67

Figura N°63: Visualización del software de gestión de pesaje realizado en LabVIEW	68
Figura N°64: Exportación en txt de la base de datos SQL Server	68
Figura N°65: Cuadro comparativo de velocidad de registro en computadora.....	69

RESUMEN

El desarrollo y uso de balanzas en el sector agroindustrial ha sido incrementado de manera exponencial y actualmente existen diferentes tipos de balanzas de acuerdo con la capacidad y cantidad que se desea pesar.

La presente investigación tuvo como objetivo diseñar una red inalámbrica para el servicio de pesaje para el sector agroindustrial utilizando balanzas SUPER SS mediante protocolo SPI basado en microcontrolador ESP32. Para visualizar una mejora y optimización en el proceso de productividad; en los productos alimenticios. Por lo cual se propone un sistema de automatización del proceso de pesaje para los productos alimenticios, con el fin de mejorar y brindar alta eficiencia, reducir el error humano, disminuir los tiempos de gestión. Esta investigación está dividida en diagrama de conectividad, etapa electrónica y desarrollo de aplicativo, basándose en el microcontrolador, softwares ya existentes en el mercado de marcas prestigiosas como ESP32 y LABVIEW.

Para el diseño electrónico se consideró un microcontrolador ESP32, ya que cuenta con tarjeta wifi-incorporada tanto para la reducción del costo, márgenes y espacios en la tarjeta. Se elaboró un programa desarrollado en LABVIEW para el HMI que permite al usuario inspeccionar el proceso de pesaje de forma más amigable, didáctica y de fácil interacción para el usuario.

Como resultado de la investigación, este sistema logró disminuir los tiempos excesivos en el proceso de conteo para los pesajes que se realizaban de manera manual, reduce los errores humanos al momento del ingreso de los datos obtenidos en el pesaje y facilita tener la información en tiempo real enviando los datos obtenidos al SQL server. Además, la mezcla de los diferentes aprendizajes obtenidos durante la carrera y la experiencia de ambos participantes.

Palabras claves: Pesaje, gestión, automatizar.

ABSTRACT

The development and use of scales in the agro-industrial sector has been increasing exponentially and currently there are different types of scales according to the capacity and quantity to be weighed.

The present investigation had as objective to design a wireless network for the weighing service for the agroindustrial sector using SUPER SS scales by means of SPI protocol based on ESP32 microcontroller. To visualize an improvement and optimization in the productivity process, in food products. Therefore, a system of automation of the weighing process for food products is proposed, in order to improve and provide high efficiency, reduce human error, reduce management times.

This research is divided into connectivity diagram, electronic stage, and application development, based on the microcontroller, software already existing in the market of prestigious brands such as ESP32 and LABVIEW.

For the electronic design, an ESP32 microcontroller was considered, since it has a built-in wifi card to reduce the cost, margins, and spaces on the card. A program developed in LABVIEW was elaborated for the HMI that allows the user to inspect the weighing process in a more user-friendly, didactic and easy-to-interact way.

As a result of the research, this system managed to reduce excessive time in the counting process for weighing that was done manually, reduces human errors when entering the data obtained in the weighing and facilitates having the information in real time by sending the data obtained to the SQL server. In addition, the mixture of the different learning obtained during the course and the experience of both participants.

Keywords: Weighing, management, automate.

INTRODUCCIÓN

El presente trabajo de investigación es desarrollado para mejorar los procesos de pesaje en el sector agroindustriales con la ayuda de una tarjeta electrónica para el reemplazo de la operación manual de los productos alimentarios, reducir los tiempos y tener un menor margen de error en el cálculo de su producción, con lo cual tiene como objetivo de diseñar una red inalámbrica para el servicio de pesaje en el sector agroindustrial utilizando balanzas SUPER SS mediante protocolo SPI basado en microcontrolador ESP32. La presente investigación está dividida en 4 capítulos, estructurados de la siguiente manera: En el capítulo I, se exponen los temas correspondientes al planteamiento del problema donde se considera la problematización, objetivo general y específicos, así como la importancia, justificación y límites de la investigación. En el capítulo II, se presenta el marco teórico, donde se consideran tanto los antecedentes nacionales como internacionales; adicionalmente las bases teóricas que brindan apoyo para esta investigación. Mencionando de manera descriptiva la balanza SUPERSS, LabVIEW, entre otros. En el capítulo III, se describe el desarrollo del diseño de la red inalámbrica para el servicio de pesaje utilizando las balanzas SUPERSS, donde se detalla el proceso de evaluación en los requerimientos para la selección del microcontrolador, software LabVIEW lo cual permite visualizar en la pantalla HMI Nextion los pesos obtenidos con rangos de tolerancia. En el capítulo IV, se presentan y analizan los resultados de la presente investigación. Finalmente se presentan las conclusiones en función a los objetivos propuestos y se redactan las recomendaciones que se deben tomar para futuras líneas de investigación.

CAPÍTULO I: PLANTEAMIENTO Y DELIMITACIÓN DEL PROBLEMA

En este capítulo se describe la problemática en la investigación, se presentan los objetivos generales y específicos, planteando los alcances y limitaciones que se dieron en el desarrollo del diseño, la importancia y justificación de la presente tesis.

1.1..... D descripción y delimitación del problema

En la actualidad en la agroindustria del Perú existe un déficit en el área de la investigación y tecnología con respecto a los sistemas de pesaje. Estas faltas tecnológicas producen que se generen retrasos, gastos y deficiencias en la producción de las empresas lo cual hace que la sostenibilidad de la producción y tiempos de entrega no sean los adecuados.

El sistema de pesaje más utilizado es el electromecánico; lo cual se ve reflejado en las empresas con mayor poder económico, que son las que pueden acceder a la tecnología avanzada que se encuentra en el mercado actual. Por otro lado, las pequeñas empresas y menos poder adquisitivo siguen recurriendo a la tecnología básica que tienen a su alcance.

Lo cual refleja que tenga que hacer un mayor esfuerzo en varios puntos de su producción en el sector como realizar cálculos y pesaje de modo manual, no tener un control en tiempo y baja seguridad en su sistema de conteo de producción. La mayoría de las empresas para poder acercarse a la tecnología de menor costo ha optado por adquirir equipos de procedencia asiática; ya que esta tecnología es más accesible que otras marcas reconocidas a nivel de pesaje industrial. En la Figura N°1 se puede visualizar el desarrollo por mes entre el 2020 y 2021 en venta, servicios de mantenimiento y repuestos de las balanzas industriales en el país.

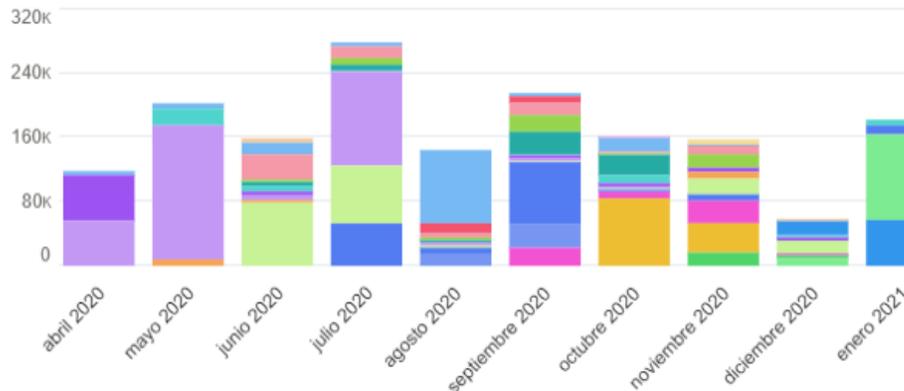


Figura N°1: Venta, mantenimiento y repuestos de balanzas industriales en Perú.

Fuente: PRECISIÓN PERÚ S.A. (2021)

El modelo de balanza asiática a utilizar para las empresas de la agroindustria son las SUPER SS; ya que tienen una confiabilidad de repetibilidad de peso, durabilidad y correcta calibración; no obstante, se tiene como desventaja el límite de comunicación que puede tener debido a que su arquitectura es de lazo cerrado y no cuenta con un puerto de comunicación externo hacia una PC o dispositivo electrónico. La falta de comunicación de estos equipos hace que los trabajadores digiten y guarden la información de cada pesaje de forma manual (hoja y lápiz) lo genera que todo su sistema se pueda hacer engorroso ya que se puede prestar a malas prácticas beneficiándose individualmente los trabajadores y generando pérdidas a las empresas. Por otra parte, el registro de datos al sistema se realiza al término de la faena de trabajo lo cual se efectúa de forma manual; generando retrasos y pérdidas de tiempo de los colaboradores. Al culminar la digitación de los pesajes se sube a sus servidores y en este punto recién se puede visualizar, gestionar y analizar de cómo va la producción, esta información ya no es en tiempo real ya que tiene una demora de 2 o 3 días.

Teniendo en cuenta lo mencionado anteriormente, se propone en la presente investigación el diseño de una red inalámbrica para el servicio de pesaje utilizando las balanzas SUPER SS, para lo cual se formula la siguiente interrogante:

1.2..... F

ormulación y delimitación del problema

1.2.1 Problema General

¿Cómo diseñar una red inalámbrica para el servicio de pesaje en el sector agroindustrial utilizando balanzas SUPER SS mediante protocolo SPI basado en microcontrolador ESP32?

1.2.2 Problema Específicos

- a) ¿Cómo diseñar la distribución de la red inalámbrica para el sistema de pesaje?
- b) ¿Cómo desarrollar la tarjeta electrónica con el microcontrolador ESP32 para establecer la comunicación entre la balanza y el software de gestión?
- c) ¿Cómo desarrollar el software de gestión del proceso LabVIEW para el monitoreo de la obtención del peso de las balanzas y SQL para manejo de datos en el servidor?

1.3..... I

Importancia y justificación del estudio

1.3.1 Importancia del estudio

Con el diseño se propone la automatización para el servicio de pesaje en el sector agroindustrial utilizando balanzas SUPER SS vinculada con una tarjeta ESP32 que se conectará vía Wifi a un software de PC para la recopilación y almacenamiento de datos que permitirá un proceso óptimo y confiable lo cual suprimirá drásticamente la operación manual.

1.3.2 Justificación Tecnológica.

La importancia de la tecnología de esta propuesta de diseño de una red inalámbrica para el servicio de pesaje con el propósito de que las empresas agroindustriales logren una mayor eficiencia; confiabilidad y seguridad al momento de la captura de información. Que al ser aplicado contribuirá a evitar el fraude al momento de realizar el pesaje. Viendo reflejada la obtención de mayores ganancias y rentabilidad; ya que va a tener mayor velocidad en la

captura de pesaje en tiempo real que ayudará a tener un mejor control y monitoreo de todo el proceso de producción.

1.3.3 Justificación Económica

Esta investigación al ser implementada ayudará a las empresas económicamente tanto en las ganancias y productividad de estas, a la vez permitirá un ahorro generando menos gastos en compras de equipos que podrían tener una solución similar, pero a un precio mucho más elevado.

1.4..... L

imitación del estudio

La energía de la batería de la balanza SUPER SS, se consumirá más rápido debido a que alimentará también a la tarjeta electrónica, esto hace que ambos equipos tengan que compartir la fuente de energía y genere esta menor duración, pero esto se podría superar cambiando la tecnología de la batería con otra de mayor capacidad.

El uso del proyecto solamente funcionará con las balanzas del modelo SUPER SS ya que la programación desarrollada solamente se enlaza con el protocolo de comunicación SPI y la trama de información que cuenta dicho modelo.

1.5..... O

objetivos

1.5.1 Objetivo General

Diseñar una red inalámbrica para el servicio de pesaje en el sector agroindustrial utilizando balanzas SUPER SS mediante protocolo SPI basado en microcontrolador ESP32.

1.5.2 Objetivo específico

- a) Diseñar la estructura de la red inalámbrica para el sistema de pesaje.
- b) Diseñar la tarjeta electrónica con el microcontrolador ESP32 para establecer la comunicación entre la balanza y el software de gestión.

- c) Desarrollar el software de gestión del proceso en LabVIEW para el monitoreo de la obtención del peso de las balanzas y SQL para manejo de datos en el servidor.

CAPÍTULO II: MARCO TEÓRICO

En el presente capítulo se describe el marco histórico de la investigación, donde se detalla los antecedentes nacionales e internacionales y las bases teóricas enlazadas que sustentan y complementan la tesis.

2.1..... A

ntecedentes del estudio de investigación.

A continuación, se presentan los estudios más importantes para esta investigación, basados en tesis referentes al desarrollo de sistemas de pesaje industrial.

2.1.1 Antecedentes internacionales.

Nieto y Quimi (2018), desarrollaron la investigación para obtener el título en Ingeniería Electrónica, que comprende el diseño e implementación de un sistema de pesaje con tecnología DFID. Universidad Politécnica Salesiana Sede Guayaquil. Ecuador, Guayaquil. Desarrollaron el proceso del sistema de pesaje con la finalidad de guardar en el proceso los pesos de los productos, para lograr disminuir el periodo en la transmisión de los datos obtenidos en la balanza. Este proceso se logra utilizando el PLC, módulo de pesaje y un HMI con el cual se puede visualizar los pesos de cada producto. En su investigación concluyeron que el diseño brinda mayor fiabilidad al almacenar los datos obtenidos.

Valdivieso (2018), realizó la investigación para obtener el título de Ingeniería Electrónica Digital y Telecomunicaciones, que comprende “Diseño de un sistema automático de pesaje de snacks para la empresa SOINTEC”. Universidad

Tecnológica Israel. Ecuador, Quito. Plantearon que el diseño de un sistema automático de pesaje para una máquina de empaque de snacks; el cual permite obtener un sistema de pesaje altamente preciso y automatizado, aumentando la capacidad productiva de la empresa donde se utilice el multicabezal que garantiza la medida exacta de varias presentaciones del producto y brinda seguridad en la verificación del producto final. Para el sistema desarrollado se utiliza un PLC XINJE XD3 que adquiere los datos de un HMI TouchWin OP320 de operación, pulsadores, entradas digitales, sensores, situaciones de emergencia y condiciones de falla humana. En su investigación resalta que el diseño del sistema de pesaje genera una mayor producción, optimiza recursos humanos y técnicos, reduce pérdidas o desperdicios de producto al momento de empacar.

Ledesma y Ortega (2017), realizaron la investigación para obtener el título de grado de ingeniería Mecatrónica, que comprende la automatización en el proceso de distribución, moltura y gestión de materias primas de alimentos balanceados. Universidad Autónoma del Occidente. Colombia, Santiago de Cali. Desarrollaron el diseño con el propósito de incrementar la productividad y reducir los gastos de fabricación de las líneas acuícola y pecuaria. Comienza con la adquisición de los datos, que es brindada por la empresa; al tener estos datos desarrolla el método de análisis QFD. El PLC brinda el control total del sistema, con el objetivo de mejorar la productividad, reduciendo el tiempo que está encendida la balanza. Concluyeron que el sistema logra el aumento de la producción en la fábrica y reduce el gasto de los kW/ t.

2.1.2 Antecedentes Nacionales

Sánchez (2019), realizó la investigación para obtener el título en ingeniería Industrial, donde se desarrolla la “Propuesta de mejora en las áreas de mantenimiento y logística para reducir costos operacionales en la empresa balanzas universal s.a.c”. Universidad Privada del Norte. Perú, Trujillo. Se realizó como objetivo el diagrama de Ishikawa donde se identificó los problemas que causan pérdidas; para luego realizar el diagrama de Pareto otorgando una

priorización dentro de la empresa. A la vez, el control de los costos operativos es fundamental para su crecimiento y para el incremento de su rentabilidad. Al lograr esto, las compañías mejoran continuamente sus operaciones y gestionar de mejor manera sus cadenas de suministro. Concluyó que se logró reducir los costos operativos que suelen estar asociados principalmente con la eficiencia y tiempo, siendo el resultado de la integración del sistema de administración dentro de las funciones que conforman la empresa.

Bayona (2020), realizó la investigación para obtener el título en ingeniería Informática, donde se desarrolla la “Implementación de un aplicativo móvil para la automatización de la toma de datos de pesaje para la empresa pesquera Terranova S.A.C.”. Universidad Nacional de Piura. Perú, Piura. Desarrolló el diseño de un sistema automatizado que tiene como objetivo principal implementar una aplicación móvil para visualizar la obtención de los datos en el pesaje. La empresa presenta dificultades en la obtención de datos porque se desarrolla de forma manual el cual genera errores. Para el desarrollo del aplicativo se utilizó la plataforma Android Studio. Se concluye que con el aplicativo desarrollado los colaboradores en la Empresa Pesquera Terranova S.A.C. realiza las actividades del pesaje con mayor precisión y agilidad, logrando reducir el tiempo de espera de los clientes.

2.2 Bases teóricas vinculadas a la variable o variables de estudio.

2.2.1 Agroindustria

La agroindustria en nuestro país representa a uno de los sectores con mayor crecimiento superará los US \$8 MM en exportaciones, a pesar de la pandemia. El portal Agronegocios - Perú informó que nuestro país es uno de los grandes exportadores a nivel mundial de frutas. La mayor exportación es de arándanos, espárragos y paltas en el mundo. En el caso de los mangos y uvas ocupamos el segundo puesto en exportación para el mercado latinoamericano. Como se muestra en la Figura N°2 se puede visualizar el desarrollo del crecimiento en la agroexportación. (Olivera, 2020).



Figura N°2: Crecimiento histórico en la agroexportación

Fuente: Sunat (2018)

Gracias al ingreso en el mercado asiático ha permitido este crecimiento ya que las exportaciones se han desarrollado de manera exponencial y se pronostica que continúe en el periodo 2021, como se muestra en la Figura N°3. Se espera abrir nuevos mercados con proyección a futuro como arándanos en India y Malasia, cítricos en India y Vietnam, granadas en China, Malasia, Corea del Sur y Taiwán, paltas Hass en Filipinas y Malasia, y uvas en Japón. (Rojas, 2020)

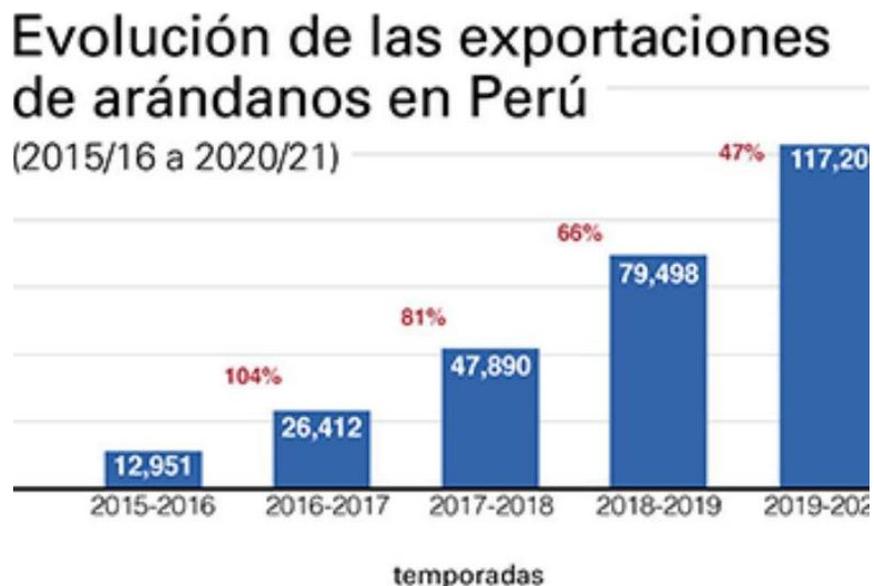


Figura N°3: Evolución de exportaciones

Fuente: Red Agrícola (2020)

Para seguir con este crecimiento se debe de acelerar los procesos de producción, por lo cual se requiere adquirir nuevas tecnologías para sustentar y desarrollar

estos nuevos negocios como por ejemplo la adquisición de equipos modernos y colaboradores con capacitación actualizada. Esto ayudará para abarcar toda la demanda que se desea cubrir en estos nuevos proyectos.

2.2.2 Balanzas

Hace más de 7 mil años las balanzas han sido uno de los instrumentos de medida usado con mayor frecuencia para el pesado de diversos materiales, desde elementos químicos hasta el pesaje de vehículos para minería.

Entonces la balanza o también conocida como báscula, es un instrumento que es utilizado con la finalidad de conocer la masa de un cuerpo u objeto. En la actualidad para las plantas industriales no es rentable utilizar básculas o balanzas domésticas, por ende, se tiene como opción las balanzas técnicas, donde la función principal es dar el verdadero peso de los productos. Los modelos de las balanzas son muy variados y se adaptan a las diferentes necesidades de cada usuario final, basados en mecanismos electrónicos que mide la masa de algún objeto, incluso los más ligeros con gran precisión y rapidez.

Las balanzas son instrumentos que miden la masa de algún objeto, mediante la comparación de la fuerza de gravedad que actúa sobre el cuerpo. El peso es la fuerza que el campo gravitacional ejerce sobre la masa de un cuerpo que son atraídos por la tierra, siendo la fuerza el producto de la masa por la aceleración de la gravedad.

$$F = M \cdot G \quad (1)$$

Donde:

F: Fuerza en N.

M: Masa en Kg.

G. Gravedad en m/s^2

a) Balanzas Electrónicas

Se caracterizan principalmente porque realizan el pesaje mediante procedimientos que implican el uso de sensores. Las mismas que se

presentan como una alternativa a las balanzas de tipo mecánica, que tiene el mismo cometido, pero se fundamentan en un juego de contrapesos (Mayorga, 2016).

Las balanzas electrónicas pueden llegar a tener una exactitud muy alta y un precio razonable, lo cual hace que sean competitivas en el mercado industrial. No obstante, requiere de mayor énfasis en el programa de calibración anual. La balanza electrónica utiliza un sensor para conocer el valor del peso que se deposita. El mismo envía distintas señales eléctricas en función del peso, señales que serán digitalizadas y decodificadas por un pequeño procesador. El valor resultante será mostrado en una pequeña pantalla LCD. Si la balanza está correctamente calibrada la exactitud logra ser muy alta; por ende, se ve reflejado la gran demanda en diversos ámbitos de trabajo.

- Celda de Carga.

Las celdas de carga son dispositivos que momentos aplicados en una trayectoria de carga y la fuerza, se da entre la superficie de contacto y la superficie de apoyo. Las celdas de carga son diseñadas con materiales elásticos, lo cual le ofrece características lineales que la convierten en una herramienta muy voluble. La relación entre las medidas de deformación tomadas y la carga aplicada será lineal mientras la superficie de contacto y la superficie de apoyo de la celda de carga permanezcan en la región de operación elástica.

En la siguiente Figura N°4 se muestra el modelo comercial en la marca PRECISION. El material usado para fabricar las principales partes de la celda de carga es el aluminio, con la finalidad de obtener el mejor rendimiento posible. Cabe agregar que el aluminio no es el único material utilizado; ya que se tiene actualmente las opciones con material semiconductor; en la Tala N°1 se describe las características de cada uno, con la ventaja de que por lo general son más pequeñas en comparación con

las metálicas. La clasificación de la celda de carga se divide principalmente en una parte mecánica y una parte eléctrica. (Platt 2011).



Figura N°4: Celda de Carga

Fuente: PRECISION (2021)

Tabla N°1: Características de las Galgas Metálicas y Semiconductoras

Características	Galgas Metálicas	Galgas de Semiconductor
Margen de Medida ($\mu\epsilon$)	0,1 – 40000	0,001 - 3000
FG	2 – 5	50 - 200
Resistencia (Ω)	120, 135, 600 ... ($\pm 0,1\%$)	1K – 5K ($\pm 1\%$)
Tamaño (mm)	0,4 – 150	1- 5

Fuente: Muller (2010)

- Galga extensiométrica

Es un transductor pasivo que convierte la deflexión, deformación o tensión en una señal eléctrica. También conocido como resistencia eléctrica de precisión, donde se tiene la relación de la deformación mecánica y la variación del valor de resistencia eléctrica. A continuación, en la Tabla N°2 se tiene la lista de materiales más usados en la fabricación de las extensiométricas.

Tabla N°2: Lista de materiales galgas extensiométricas.

Material	Factor de Galga
Platino (Pt 100%)	6.1
Platino – Iridio (Pt 95%, Ir 5%)	5.1
Platino – Tungsteno (Pt 92%, W 8%)	4.0
Isoelastic (Fe 55%, Ni 36% Cr 8%, Mn 0.5%)*	3.6
Advance / Copel (Ni 45%, Cu 55%)*	2.1
Nichrome V (Ni 80%, Cr 20%)*	2.1
Karma (Ni 74%, Cr 20%, Al 3%, Fe 3%)*	2.0
Armour D (Fe 70%, Cr 20%, Al 10%)*	2.0

Fuente: Omega (2021)

- Puente de Wheatstone

Esta configuración es el puente de medida más sencillo, como se puede visualizar en la siguiente Figura N°5, está conformada por 04 resistencias, el voltaje de alimentación y voltaje de salida. Es utilizado para lograr la medición de la resistencia eléctrica estática o dinámica; también es muy adecuado para lograr la compensación de temperatura. (Tipler, P. & Mosca, G., 2005)

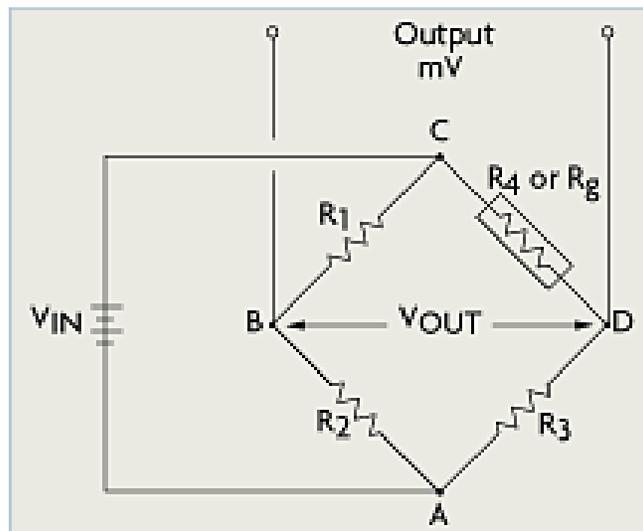


Figura N°5: Puente Wheatstone

Fuente: Omega (2021)

$$Vg = \frac{\frac{R_1 + R_4}{R_2 + R_3}}{\left(\frac{R_1}{R_2} + 1\right)\left(\frac{R_4}{R_3} + 1\right)} \quad (2)$$

Donde:

Vg: Voltaje de ganancia en Voltios

R: Resistencia en Ohmios

Condición de medición de resistencia en un puente de Wheatstone.

$$\frac{R_1}{R_4} = \frac{R_2}{R_3} \quad (3)$$

b) Balanza electrónica modelo SUPER SS

Diariamente en el proceso agroindustrial se requiere realizar rutinas de pesaje continuas, donde es indispensable que se realice de manera fácil y eficiente. Por ende, se utiliza las balanzas electrónicas que logran tener una exactitud relevante y un precio moderado, lo cual hace que sean muy competitivas en el mercado nacional. En la Figura N°6, se visualiza la balanza para el pesaje de productos alimenticios.

El modelo de balanza SUPER SS, cuenta con la certificación IP68 como sus siglas indican “Ingress Protection” 68, es el grado máximo de protección que en esta balanza tiene frente al acceso de partículas, como polvo, y líquidos (agua, por ejemplo). Lo cual hace que se vea reflejado el alto nivel de precisión en el pesado con facilidad de traslado y bajo costo. (Toledo 2021)



Figura N°6: Balanza SUPER SS de 15 Kg

Fuente: Mettler Toledo (2021)

En la siguiente Tabla N°3, se visualiza las principales características de la balanza modelo SUPER SS (Para mayor detalle, ver anexo N°1):

Tabla N°3: Característica de la balanza modelo SUPER SS.

Características	
Marca	PRECISION
Modelo	SUPER SS
Capacidad	06 kg
Precisión	1 gramo
Pantalla	Con retroiluminación / LCD
# de Pantallas	Doble
Indicador de carga	Si
Visualización	Peso bruto y neto

Certificación	IP68
Batería	Recargable
Material	Acero inoxidable
Función	Tara - Corrección de cero automático

Fuente: Precisión Perú S.A. (2021)

2.2.3 SolidWorks

El software SolidWorks presenta una excelente herramienta de software CAD para diseño y desarrollo mecánico en 2D y 3D de ensamblajes y piezas complejos, que ofrece la empresa SolidWorks Corp. El cual brinda la facilidad de crear un diseño de manera eficiente y precisa. Otra opción que brinda el software es evaluar los costos de diseño y fabricación utilizando herramientas de estimación de costes y realizando comprobaciones de la viabilidad de fabricación. (Dassault Systèmes 2002-2021).



Figura N°7: Logo de SolidWorks

Fuente: Dassault Systèmes SolidWorks Corporation (2002-2021)

Usando este programa podemos diseñar piezas, conjuntos y obtener de ellos planos específicos y sub-ensamblajes necesarios para la fabricación. El proceso se desarrolla de la siguiente manera; plasmar la idea mental del proyecto en el programa, desarrollando líneas de dibujo de la pieza o conjunto. Brindando las diferentes soluciones (planos, pesos del material, fichas de trabajo, etc.). En la Figura N°8, modelo de circuito de comunicación.

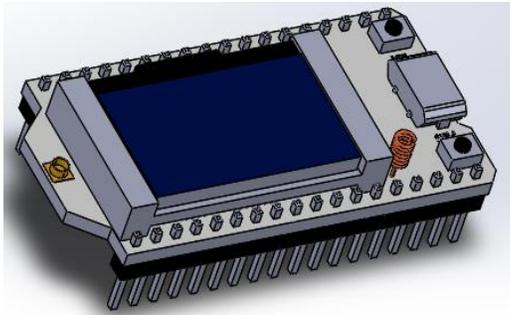


Figura N°8: Circuito de comunicación

Fuente: Propia.

2.2.4 Software Proteus

Proteus Design Suite es un programa de creación de diseño circuital, desarrollado por Labcenter Electronic, que está conformado de dos aplicativos principales: Ares e Isis. El Programa ISIS, Intelligent Schematic Input System el cual es para diseñar el plano eléctrico del circuito que se desea realizar, se puede utilizar desde simples equipos como resistencias, condensadores y fuentes de alimentación hasta microcontroladores. Los diseños realizados en Isis pueden ser simulados en tiempo real, mediante el módulo VSM, asociado directamente con ISIS.

Proteus tiene varias versiones de presentaciones como en la Figura N°9, integrada con ISIS, es VSM, el Virtual System Modeling, una extensión integrada con ISIS, la cual puede hacer simulación de tu circuito en tiempo real; la familia de todos los microcontroladores, introduciendo manualmente el programa que controlará el microcontrolador y cada una de sus salidas, y a la vez, simulando las tareas que queramos que lleve a cabo con el programa. Se puede simular circuitos electrónicos con microcontroladores conectados a diferentes equipos como leds, motores, pantallas LCD, entre otros.



Figura N°9: Software Proteus

Fuente: Labcenter Electronics 2021

El otro software para utilizar para el diseño de la tarjeta electrónica de tu circuito creado en ISI es el ARES, o Advanced Routing and Editing Software; esta herramienta se utiliza para la fabricación de placas de circuito impreso, permitiendo editar generalmente, las capas superficiales (Cobre Superior), y de soldadura (Cobre inferior). (Labcenter Electronics, 2021).

2.2.5 Sistema Electrónico

Está conformado por un grupo de componentes electrónicos que se relacionan obteniendo un resultado final. Se divide principalmente en tres partes, señal de entrada, procesamiento y señal de salida. Donde la señal de entrada es la parte que captura la información, ver Figura N°10, el procesamiento es la parte donde con la información obtenida de la primera parte se procesa para generar el resultado deseado y la señal de salida es donde la información transformada del proceso pueda ir al lugar correspondiente que se necesite.



Figura N°10: Sistema electrónico

Fuente: Programación y robótica (2014)

Usualmente funcionan usando señal en el tiempo-continuo (CT) o señales de tiempo discreto (DT). Un sistema de tiempo-continuo normalmente toman valores en un rango continuo siendo procesadas por sistemas analógicos (analizador de frecuencia, filtros) con la finalidad de modificar alguna característica o extraer la información requerida. En consecuencia, se tiene que las señales de ingreso como la de salida son de naturaleza analógica.

Para la otra opción que se tiene que es el método de una señal de tiempo discreto, en este caso se requiere realizar una interfaz de relación de la señal analógica y el procesador digital. Mayormente conocido como “Convertor análogo-digital (A/D)”

- Función de transferencia de sistemas

Cualquier subsistema puede representarse como un bloque simple con una entrada y salida como se muestra en la Figura N°11. Generalmente, la entrada se designa como: θ_i y la salida como: θ_o .

$$G = \frac{\theta_o}{\theta_i} \quad (4)$$

Donde:

G: Ganancia

θ_i : Entrada

θ_o : Salida

La relación de salida sobre entrada representa la ganancia, (G) del subsistema y, por lo tanto, se define como: $G = \theta_o / \theta_i$

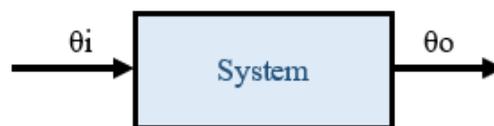


Figura N°11: Sistema electrónico

Fuente: Programación y robótica (2014)

2.2.6 Microcontroladores

Se trata de un circuito integrado como muestra la Figura N°12 que contiene en su interior una Unidad Central de Procesamiento (CPU), la cual cuenta con memoria tanto para códigos como datos y de recursos material necesario para el funcionamiento de aplicaciones. Estos equipos pueden ser tan pequeños para el tamaño de una moneda, aunque, depende del dispositivo que debe controlar. Un microcontrolador común contiene un generador de reloj integrado y una memoria RAM y ROM/EPROM/EEPROM, para su funcionamiento se necesitan principalmente cristal de sincronización su alimentación para su activación y el programa de control para el desarrollo de la secuencia a realizar. Los microcontroladores más usados son: AVR, ARM, MP430 y PIC39. (Max Embedde 2014)

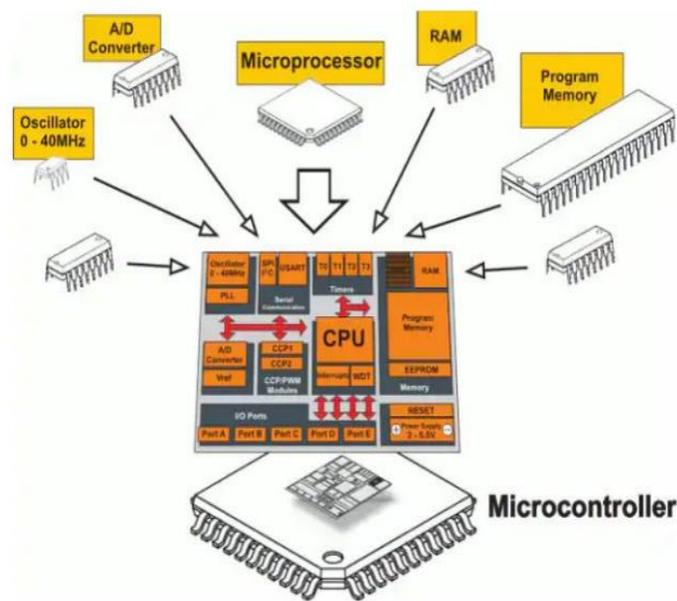


Figura N°12: Diagrama que compone un microcontrolador.

Fuente: Max Embedde (2014)

2.2.7 Microcontrolador ESP32

El ESP32 es una serie de microcontroladores de bajo costo y consumo, este dispositivo cuenta con su tarjeta Wifi y Bluetooth incorporado. A diferencia de otros microcontroladores populares como el Arduino este

tiene una mayor capacidad de almacenamiento como de velocidad en el proceso de información, periféricos integrados, doble núcleo incorporado, entre otras cosas. Este componente se utiliza mayormente en el desarrollo de sistemas IoT debido a su bajo consumo de energía y sus múltiples aplicaciones para vincularse a varios sistemas en tiempo real. Ya que es de código abierto este instrumento se puede programar en el IDE de Arduino. (Benito. 2019)

Como muestra en la Tabla N°4 donde se tiene el cuadro comparativo entre el ESP32 vs el Arduino UNO respecto a sus características, para ver la ficha técnica del microcontrolador ver Anexo N°2.

Tabla N°4: Cuadro Comparativo ESP32 vs Arduino UNO

Especificaciones / Tarjeta	ESP32	ARDUINO UNO
Numero de Núcleos	2	1
Arquitectura	32 bits	8 bits
Frecuencia CPU	160 MHz	16 MHz
WiFi	Si	No
Bluetooth	Si	No
Ram	512 KB	2KB
Flash	16 MB	32 KB
Pines i/O	36	14
Comunicaciones	SPI, i2C, UART, I ² S, CAN	SPI, I2C, UART
Pines Analógicos	18	6
Pines Digitales	2	0

Fuente: Elaboración propia.

2.2.8 Protocolo SPI

El protocolo SPI (Serial Peripheral Interface) es un modo de comunicación para la transmisión de información entre dispositivos electrónicos en el cual cuenta

con un dispositivo líder (MAESTRO) y uno o varios dispositivos como empleados (ESCLAVOS) ver Figura N°13.

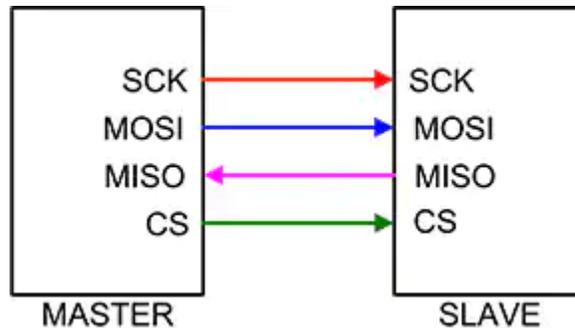


Figura N°13: Protocolo SPI

Fuente: DigiKey electronics 2016.

El Maestro es quien dirige y gobierna toda la red de información que se transmite con los cables. El SPI es un protocolo que transmite los datos por 4 señales como indica en la Tabla N°5:

Tabla N°5: Señales del SPI

SCLK	Es el punto que señala la sincronización con cada pulso de este reloj, se lee o se envía un bit.
MOSI (Máster Output Slave input)	Salida de datos del Máster y entrada de datos del esclavo.
MISO (Máster Input Slave input Output)	Salida de datos del esclavo y entrada de datos del Máster.
CS/select	Para seleccionar un Esclavo o el Máster le diga que se active

Fuente: DigiKey electronics 2016.

Las ventajas de esa comunicación es su gran rapidez en el envío de datos y gran facilidad por otro lado su desventaja es la corta distancia que puede comunicarse. (DigiKey electronics 2016). Para que el dispositivo maestro realice un envío de datos en modo esclavo, debe realizar ciertas pautas de modo correcto:

- Primer paso, se debe elegir el esclavo para realizar el intercambio de información. En este caso se debe seleccionar el nivel bajo para el selector de chip respectivo con el esclavo. Previo a la selección del esclavo, la salida de sus datos se realiza en alta impedancia lo cual permite a otros esclavos utilicen el bus de optima manera.
- Luego, el maestro opera el reloj de comunicación (SCLK) brindando una señal simétrica (igual tiempo a nivel alto y bajo) utilizando una frecuencia correcta para la velocidad de transmisión de datos.
- Al activarse el reloj, el maestro envía un bit en la MOSI en cada fase de reloj.
- Al terminar la función de transmisión de todos los bits, el maestro deja no disponible el reloj de comunicación.
- Para finalizar, el maestro elige el nivel alto la línea de selección de chip para colocar al esclavo que el envío ha finalizado. Por ende, el esclavo elige la salida de datos en alta impedancia y así facilita el envío de otros esclavos.

Como se muestra en la Figura N° 14, se tiene la transmisión entre los dispositivos maestro y esclavo.

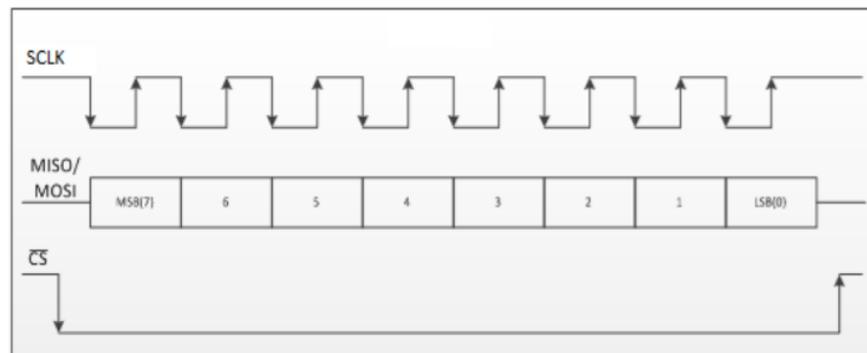


Figura N° 14: Presentación de transmisión SPI

Fuente: DigiKey electronics 2016.

Rapidez de transmisión (baudios)

Está definida en relación con la cantidad de bits enviados en un segundo por medio de bits/s. Entonces se tiene que el envío de cada bit se ejecuta durante un

ciclo del reloj SCLK, la rapidez de transmisión está sujeta con la frecuencia del reloj SCLK y se presenta en la siguiente fórmula, Velocidad de transmisión SPI:

$$Baudios = \frac{1}{F_{SCLK}^{(Hz)}} \quad (5)$$

Dónde:

Baudios: Velocidad de transmisión expresada en bits/s

F_{SCLK} : Frecuencia del reloj en Hercios.

Ya que el bus no posee una delimitación genuina, lo cual brinda valores estimados entre 12500 bit/s hasta 12500000 bits/s.

Formas de transferencia

Para el caso del bus SPI que ejecuta dos variables con la finalidad de presentar el instante en el que se redactan o leen las líneas de datos.

- CPHA: Este indicador presenta el retardo en referencia a la banda del reloj de sincronización.
- CPHA = 0: No presenta retardo.
- CPHA = 1: Se considera como una banda de retardo al momento de realizar la lectura.
- CPOL: Se entiende como la situación de stand by de la banda del reloj SCLK. Se presenta con dos estados 0 o 1.
- CPOL = 0: Nivel bajo cuando no se realicen envíos de datos.
- CPOL = 1: al realizar el envío de datos se requiere que el maestro y esclavo cuenten con los mismos parámetros.

Se tiene las siguientes opciones de combinar los estados anteriormente mencionados:

- Opción 1: CPHA = 0 y CPOL = 0

Para esta configuración se tiene que el reloj se mantiene a nivel bajo, por lo cual al iniciarse el envío de datos el maestro y el esclavo envían el primer bit, así se refleja en la Figura N°15:

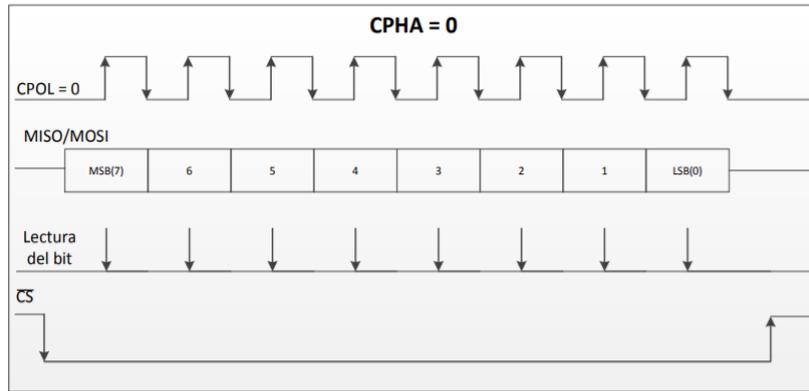


Figura N° 15: Opción 1 de transmisión SPI

Fuente: DigiKey electronics 2016.

- Opción 2: CPHA = 0 y CPOL = 1

Para esta configuración se tiene que el reloj se mantiene a nivel alto siempre y cuando no se realicen ninguna actividad, ver la Figura N°16:

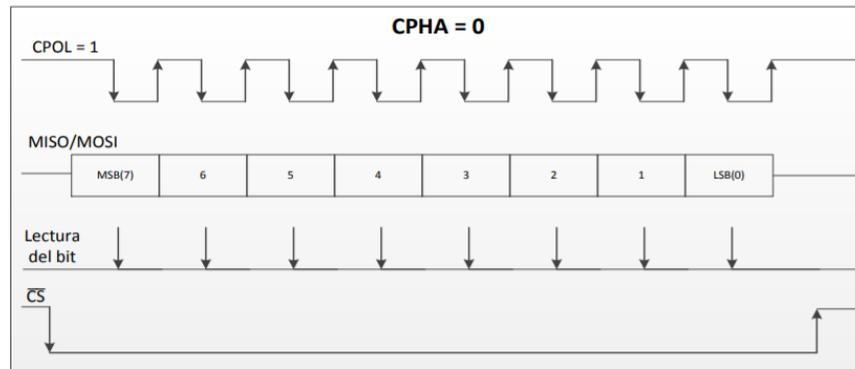


Figura N° 16: Opción 2 de transmisión SPI

Fuente: DigiKey electronics 2016.

- Opción 3: CPHA = 1 y CPOL = 0

En esta configuración se considera que existe el retardo con la línea del reloj SCLK, ver la Figura N°17:

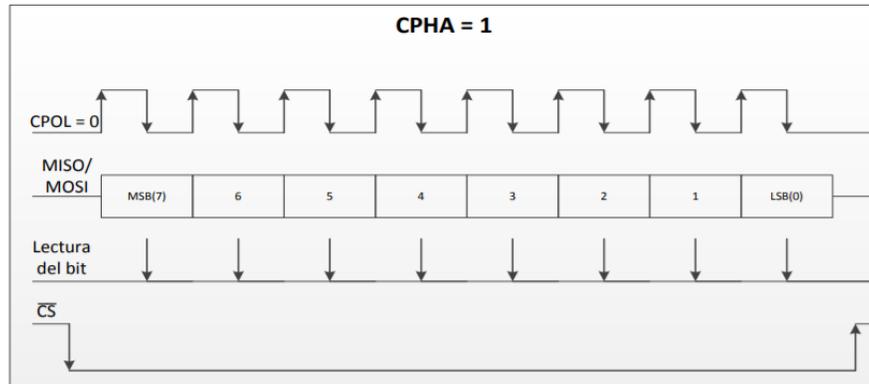


Figura N°17: Opción 3 de transmisión SPI

Fuente: DigiKey electronics 2016.

- Opción 4: CPHA = 1 y CPOL = 1

Para esta opción se desarrolla modificando el estado de reposo de la línea SCLK, quedando fijado en estado de reposo en el nivel alto, ver la Figura N°18:

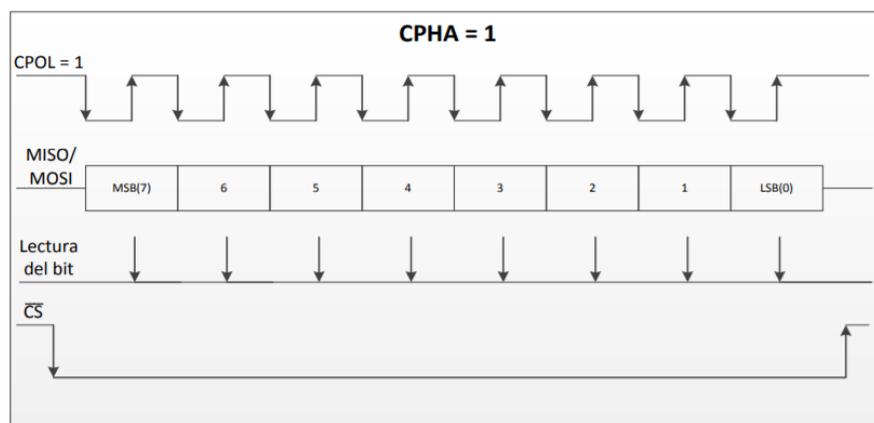


Figura N°18: Opción 4 de transmisión SPI

Fuente: DigiKey electronics 2016.

2.2.9 Sistema HMI

La relación Hombre-Máquina (HMI) es la combinación entre el proceso y los operadores; entendiéndose como el módulo de control del operador. Siendo la interfaz con mayor aplicación entre los operadores y jefes con la finalidad de

adecuar y manejar los sistemas industriales. El HMI transforma diversos datos o procesos de gran complejidad, en datos convenientes y de fácil manejo.

El trabajo del HMI está orientado en presentar datos operativos en tiempo real. Brindando los datos visuales de todos los parámetros como por ejemplo motores, tableros eléctricos, reductores, etc. Usualmente implementados en dispositivos complejos con el fin de supervisar, monitorear los parámetros más resaltantes dentro de un sistema industrial.

2.2.10 Pantalla Nextion NX4832T035

Nextion son pantallas HMI (interfaz humano - Máquina) para la visualización de procesos industriales relacionados en el tema del internet de las cosas (IoT). Este dispositivo funciona a través de TTL serial (5V, GND, TX, RX). para poder proporcionar notificaciones de eventos sobre el dispositivo al cual se conectó. (Bento, 2017). Este cuenta con una parte de hardware y de software (Editor Nextion). La tarjeta de Nextion utiliza solo un puerto serie de comunicación, así evita a los usuarios inconvenientes con cableados más complejos. (Nextion, 2018). Para mayores características y especificaciones ver Anexo N°3

El modelo NX4832T035 es un modelo de la versión básica de las pantallas Nextion que cuenta con las siguientes características según indica la siguiente Tabla N°6:

Tabla N°6: Características de las versiones de pantalla Nextion

MODEL NAME	NX3224T024	NX3224T028	NX3224T032	NX3224T035
Size	2.4"	2.8"	3.2"	3.5"
Resolution	320*240	320*240	400*240	480*230
Touch Panel	RTP	RTP	RTP	RTP
Color	65536	65536	65536	65536
Flash	4 MB	4 MB	4 MB	16MB
RAM	3584 Byte	3584 Byte	3584 Byte	3584 Byte
MCU	48 MHZ	48 MHZ	48 MHZ	48 MHZ
EEPROM	-	-	-	-

Fuente: Nextion (2021)

2.2.11 Editor Nextion

Es el software para diseñar las pantallas en los dispositivos Nextion. Dependiendo del modelo Nextion a utilizar puede aumentar o disminuir las funciones que puede utilizar dentro del programa. (Buesaquillo,2019)

La funcionalidad de las pantallas Nextion es la comunicación mediante un protocolo serial rs232. Cada componente está identificado por un ID que contiene una serie de atributos que se modifican dependiendo a la programación. Estas instrucciones se escriben por el puerto serial con unas instrucciones definidas desde las que se ejecuta acciones, hasta obtener valores de variables dentro de la comunicación. Para trabajar con placas ESP32 el intercambio de estos datos terminan con tres saltos de línea. (Makerspace, s.f.). Se presenta a continuación la Figura N°19:

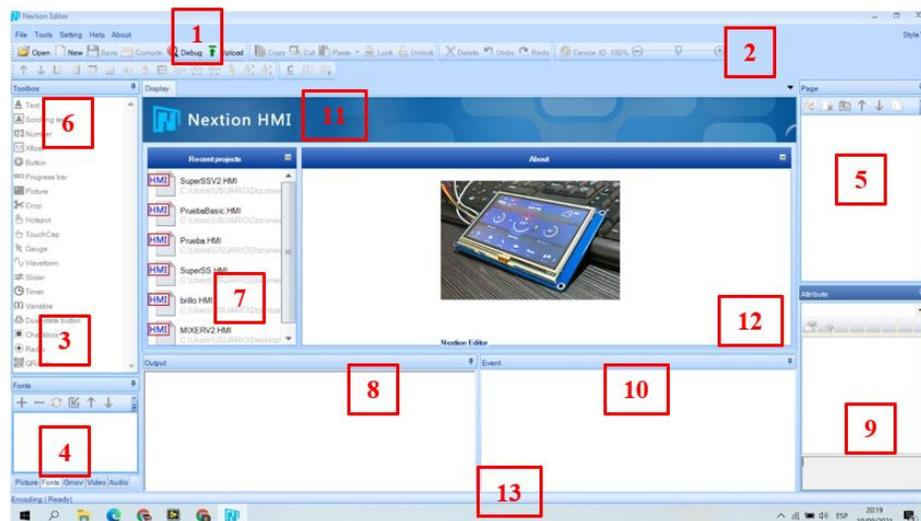


Figura N°19: Software Nextion Editor

Fuente: Makerspace 2021.

A continuación, se tiene las funciones que cuenta el Nextion Editor:

1. Menú principal
2. Componentes principales

3. Biblioteca de imágenes donde ingresan las imágenes que se utilizara en la interfaz
4. Área de visualización de las fuentes de los textos
5. Área de visualización de las páginas creadas en nextion editor.
6. Área de edición de componentes
7. Ventana visualizadora de la pantalla nextion que está utilizando.
8. Área del Evento donde se escribe el código simple.
9. Cuadro de parámetros del componente seleccionado
10. Visualización de la comunicación entre el nextion editor con la pantalla nextion.
11. Ventanas donde se puede abrir cambiar archivo nextion que estas editando
12. Visualización de la compilación del programa
13. Visualización de información básica de las herramientas que seleccionas

2.2.12 Software de Aplicaciones

Los softwares son los que permite a los usuarios realizar tareas puntuales, en cualquier campo tanto de gestión como de automatización de procesos industriales, los aplicativos sirve para brindar una mayor facilidad al usuario para poder ejercer tus tareas de una manera más rápida e eficiente en el entorno.

Los softwares servir para:

- Procesos de automatización industrial
- Procesos de gestión y control
- Software educativo
- Videojuegos
- Bases de datos
- Software médico
- Software de cálculo Numérico y simbólico.
- Proporcionar ayuda al usuario para desarrollar sus tareas y actividades de manera más rápida teniendo un mejor rendimiento y mayor eficiencia.

- Reducir el número de defectos y errores que pueden ser cometidos por el usuario.
- Mejorar la estabilidad del proceso de forma que se minimicen las reelaboraciones del producto.
- Acortar los tiempos de desarrollo.

2.2.13 LABVIEW

National Instruments (2021) refieren que Labview (Laboratory Virtual Instrumentation Engineering Workbench) está diseñado para implementación en el área de ingeniería de adquisición de datos, análisis de medidas y presentación de datos. Este programa tiene varias funcionalidades por su facilidad de lenguaje, rápido proceso de información y didáctico entorno. El lenguaje de programación que utiliza este software es denominado “G”. Este es considerado de alto nivel ya que es de tipo gráfico y orientado al uso en instrumentación. Pero como código, puede ser usado para desarrollar cualquier programa que se requiera como el análisis de datos, control y monitoreo, videojuegos, sistemas de gestión, entre otros. Los programas hechos en LabVIEW son llamados Instrumento Virtual (VI), el cual se puede crear instalador o hacerlo ejecutable para su uso. El software cuenta con dos pantallas como se aprecia en la Figura N°20:

Panel Frontal: Es donde el usuario final visualiza y manipula los datos que se desea tanto controlar cómo monitorear.

Diagrama de bloques: En este se encuentra el código fuente del Panel frontal (elaborado en lenguaje G) para la modificación o depuración del programa.

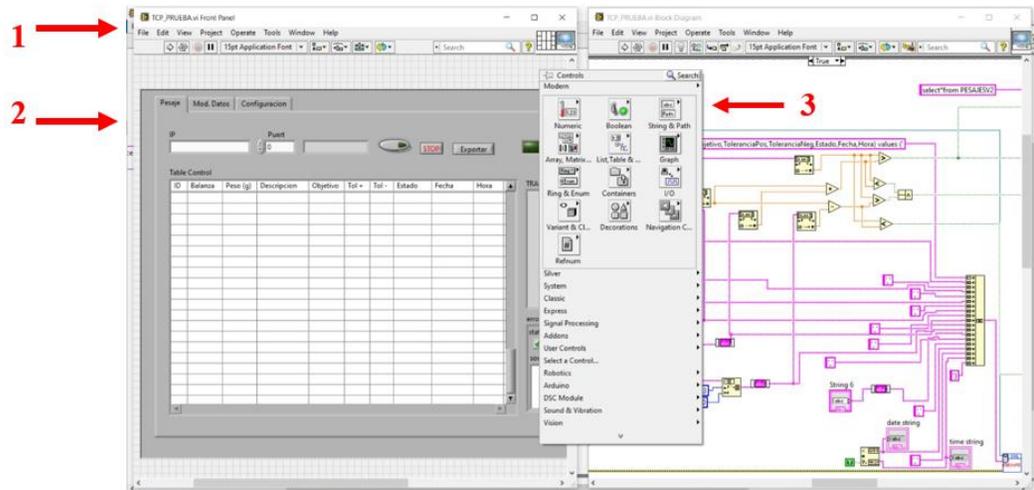


Figura N°20: Ejemplo de Pantalla Frontal de LabVIEW

Fuente: National instruments corp. (2021)

- Aplicaciones de LabVIEW

LabVIEW puede ser utilizado para varios tipos de actividades como aplicaciones de control de procesos industriales, sistemas de gestión para control de pase de personal, laboratorios educativos en el área de colegios. LabVIEW se utiliza generalmente en el manejo avanzado de señales (wavelets, FFT, Total Distorsion Harmonic TDH), aplicaciones biomédicas, procesamiento de imagen y sonido, robotización, entre otros.

- Programación gráfica en LabVIEW

El lenguaje de programación del software está basado en diagrama de bloque el cual está hecho para unir diferentes tipos de bloques que contienen una cierta tarea específica cada uno para así tener una lógica y haga cierta función. Este modo de lenguaje es más interactivo con el desarrollador para hacer su aplicativo de una manera más rápida y sencilla.

- Diseño la interfaz de usuario a partir de su código

El software tiene su pantalla frontal que interactúa con el usuario se llama VI (instrumento virtual), Este VI se puede utilizar como una subfunción dentro de un programa general. El instrumento virtual se define por la

pantalla de interfaz entre el usuario para poder hacer el proceso correcto para lo cual fue diseñado el software.

- Sistema de Comunicación

El software LabVIEW cuenta con todo tipo de sistemas de comunicación que se encuentra en el mercado incluyendo desde los antiguos hasta los modernos como, por ejemplo, Radio Frecuencia, Infrarrojo, Bluetooth, I2C, Ethernet, Serial, entre otros.

2.2.14 SQL Server

Es un software de administración de base de datos relacionado con la función principal de guardar y recuperar la información según requiera otras aplicaciones. Este programa soporta grandes cantidades de datos en el almacenamiento. Cuenta con una estabilidad y seguridad completa en toda tu información lo cual permite una confiabilidad con este sistema. Tiene la funcionalidad de trabajo modo Cliente-Servidor, esta modalidad hace que los datos sean almacenados en el servidor de la red y los clientes puedan tanto enviar como visualizar dicha información. En la Tabla N°7 se presenta características importantes que tiene el SQL Server.

Tabla N°7: Características de SQL Server

Características de SQL Server
- Soporte de Transacciones
- Escalabilidad, estabilidad y seguridad
- Soporta procedimientos almacenados
- Permite el uso de comandos DDL y DML gráficamente
- Además, permite administrar información de otros servidores de datos

Fuente: José Ramírez (s, f)

2.2.15 Redes Inalámbricas

Pipa. (2019), menciona que es una red en la que dos o más dispositivos electrónicos (computadoras, celulares, entre otros.) se pueden enlazar sin la obligación de una conexión física. Un cliente puede mantenerse conectado cuando se moviliza dentro de un determinado perímetro. Entonces, en varias ocasiones se hace uso de la frase movilidad al mencionar este punto. La red inalámbrica se basa en un enlace que utiliza ondas electromagnéticas en vez de usar conexión física. Hay diferentes tecnológicas que varían por la transferencia, frecuencia, alcance y velocidad de sus transmisiones.

Las redes inalámbricas como se muestra en la siguiente Figura N°21, donde conceden que los componentes remotos se conecten sin ningún inconveniente, así se encuentren tanto a distancia corta como a varios kilómetros. Por ende, las instalaciones de estas redes no necesitan de alguna modificación sumamente importante en la estructura existente como en las redes físicas cableadas. (MTM TELECOM 2020)

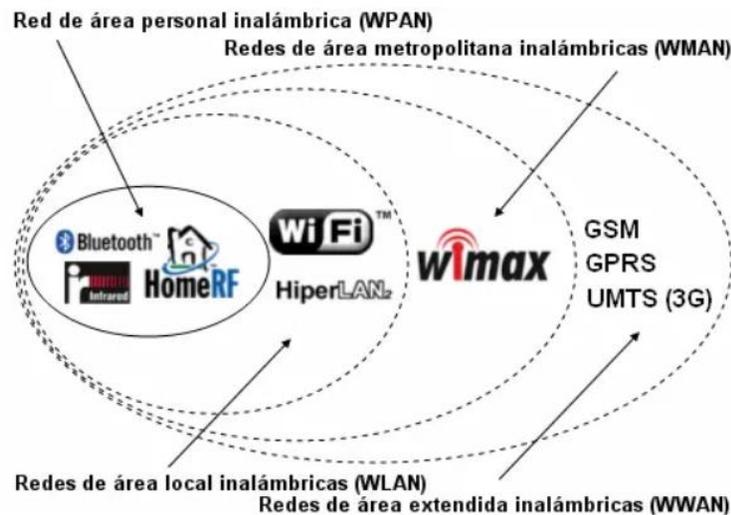


Figura N°21: Redes inalámbricas

Fuente: Ccm.net (2017)

Ventajas:

- Se basa en estándares y certificación Wi-Fi

- De fácil instalación.
- Alta confiabilidad.
- Administración es realizada por Servidor Web.
- Costo de implementación reducido.

Desventajas:

- Interferencias con infraestructuras.
- Reducida velocidad versus una red cableada.
- Representa menor seguridad.
- A mayor cantidad de usuarios conectado, la señal puede verse afectada.

2.2.16 Modos de redes inalámbricas

- WPAN (Wireless Personal Área Network)

Conocida como Red de área personal, normalmente conectado entre dispositivos electrónicos con un rango de menor alcance, en promedio de 10 metros. Usualmente usados en el entorno familiar como impresoras, teléfonos inteligentes y electrodomésticos. A través de la tecnología Bluetooth, comúnmente es reconocido como IEEE 802.15.1 y brinda una velocidad máxima de 10 Mbps.

- WLAN (Wireless Local Área)

Es la red de aérea local para una empresa, que permite compartir archivos, servicios de impresoras y demás recursos. Utilizando la tecnología Wifi (o IEEE 802.11) la cual brinda una velocidad promedio de 54 Mbps.

- WMAN (Wireless Metropolitan Área Network)

Aplicado para un espacio metropolitano como las ciudades, donde abarca un rango entre 4 a 10 k, comúnmente usados para las empresas de telecomunicaciones.

- WMAN (Wireless Wide Area Network)

Las redes inalámbricas con mayor alcance; permitiendo conectarse a diversas áreas geográficas como países y continentes a través de conexiones

satelitales o antenas de radio microondas. Usa tecnologías como: GPS, GPRS y UMTS.

2.2.17 Tecnología Wifi

Wifi es el mecanismo de comunicación inalámbrica, que en el principio fue utilizada en redes para área local y con el paso del tiempo se ha convertido en un medio para que muchos usuarios tengan acceso a internet de banda ancha. El estándar internacional con especificación IEE 802.11 (ISO/ IEC 8802-11). A continuación, se tiene las diferentes versiones de los estándares de Wifi:

- 802. Red ethernet inalámbrica.
- 802.11 b – Red ethernet inalámbrica de alta velocidad.
- 802.11 b+.
- 802.11g.
- 802. 11^a.
- 802.11n.
- HiperLAN2.

2.2.18 Características resaltantes de las redes inalámbricas.

Para el diseño de las redes inalámbricas se tienen las siguientes características:

- Facilidad de transferir información, aunque el usuario esté en constante movimiento siempre va a estar conectado a la red.
- El acceso a la información de datos debe poder realizarse en cualquier lugar y en cualquier momento.
- La información de los datos debe estar disponible siempre.
- A ser una red inalámbrica no se usa cableado lo cual hace fácil su instalación.
- Atravesar paredes para permitir alcanzar lugares donde las interconexiones no pueden.
- Llega a configurarse múltiples topologías satisfaciendo muchas aplicaciones e instalaciones.

- Los diseños son demasiado fáciles de intercambiar y también es fácil traer usuarios nuevos a la red.

Teniendo en cuenta las clases de frecuencias que usan para emitir las redes inalámbricas, pueden tener ciertas características:

Ondas de radio

- Transmisión insensible a la lluvia ya que trabaja a bajas frecuencias.
- El rango de frecuencias de operación va de 3 – 30 Hz, conocida como banda ELF
- (Extremely low frequency), y el rango 300 – 400MHz UHF (Ultra High Frequency)

Microondas terrestres:

- Usan frecuencias de uno a 300 GHz
- Usan antenas parabólicas de uno a tres metros y deben estar alineados
- La lluvia y otros fenómenos climatológicos pueden causar atenuación de la señal

Microondas satelital:

- Se utilizarán dos estaciones terrestres y satélites.
- El satélite recibe una señal (señal ascendente), la amplifica y la transmite (señal descendente).
- Cada satélite utiliza una banda en particular.

Infrarrojos:

- Transmisores y receptores regulan la luz infrarroja de forma incoherente.
- No pueden atravesar paredes y rango de operación 300 GHz – 384 THz.

CAPÍTULO III: DESARROLLO DEL DISEÑO DE LA RED INALÁMBRICA PARA LAS BALANZAS SUPER SS

En este capítulo se presenta el desarrollo del diseño de toda la red inalámbrica para el servicio de pesaje en el sector agroindustrial, utilizando la balanza SUPER SS mediante protocolo SPI basado en el microcontrolador ESP32, el cual está compuesto por las siguientes 03 etapas:

- Diagrama de conectividad.
- Electrónica de comunicación.
- Desarrollo del aplicativo.

En la Figura N°22, se muestra el diagrama de flujo general de todo el sistema:

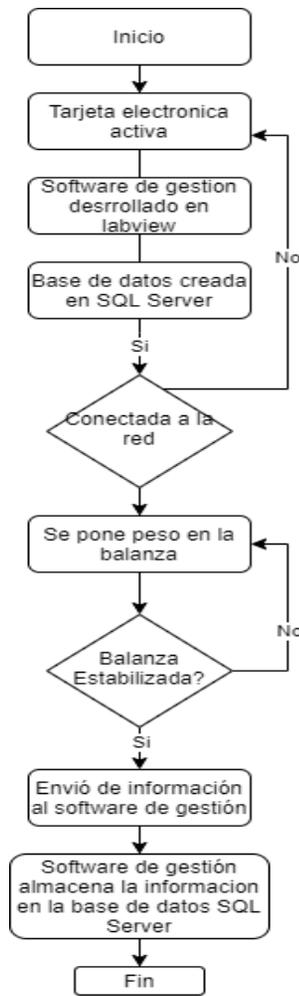


Figura N°22: Diagrama de flujo general

Fuente: Elaboración propia

3.1 Diagrama de conectividad general de la red inalámbrica de todo el proyecto

Se establece la conectividad de toda la red, donde se visualiza que para cada balanza SUPER SS se hace la conexión a una tarjeta electrónica para que esta pueda recibir su información. Con ello cada dispositivo se comunica con el SOFTWARE de PC vía WIFI lo cuales se conectan vía inalámbricamente al router el cual se vincula con la PC y el SERVIDOR.

En el diagrama mostrado en la Figura N°23, se observar que el cable azul es referencia al cable de red que conecta la PC y el servidor al router, al tenerlo conectado las balanzas SUPER SS están conectadas a cada tarjeta electrónica. El cable verde es la conexión Wifi de las balanzas hacia la red, con esto ya se tiene vinculado todo el sistema para el envío y recepción de información del pesaje.

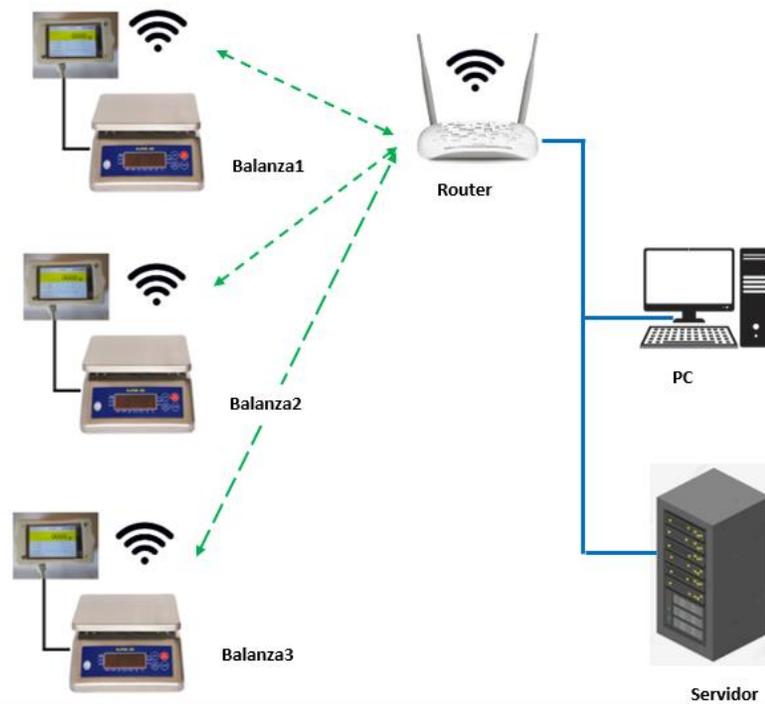


Figura N°23: Diagrama referencial de toda la red de las balanzas

Fuente: Elaboración propia

3.2 Diseño del sistema electrónico

En esta etapa se efectúa la selección de los equipos electrónicos para el procesamiento de la señal, se desarrolla el diagrama de bloques de la programación, diseño del entorno interfaz gráfica para el usuario y esquema electrónico de la tarjeta.

3.2.1 Requerimiento para la selección del microcontrolador

Teniendo en cuenta los parámetros que se necesita para poder tener toda la comunicación requerida se optó por utilizar microcontroladores. En la Tabla N°8, se muestran comparaciones de varios modelos basados en sus parámetros.

Tabla N°8: Microcontroladores Posibles a usar

Especificaciones/ Tarjeta	ESP32	ARDUINO UNO	ESP8266
# de Núcleos	2	1	2
Arquitectura	32 bits	8 bits	32 bits
Frecuencia CPU	160 MHz	16 MHz	160 MHz
WiFi	SI	No	SI

Bluetooth	SI	No	No
Ramc	512 KB	2 KB	80 KB
Flash	16 MB	32 KB	4 MB
Pines i/O	36	14	32
Comunicaciones	SPI,I2C,UART,I2S,CAN	SPI,I2C, UART	SPI,I2C,UART,I2S,CAN
Pines Analógicos	18	6	1
Pines Digitales	2	0	0

Fuente: Elaboración propia

Se escogió el ESP32, siendo este el modelo de la marca ESPRESSIF para sistemas de comunicación inalámbrica y sistemas IOT. Este dispositivo cuenta con doble procesador, cuenta con la memoria RAM necesaria para poder almacenar todo el tamaño del programa ya que se necesita una memoria superior de 100 KB, dimensiones pequeñas y una memoria flash que se necesita que debe ser superior de 12 MB para poder capturar el pesaje de la balanza SUPER SS, WiFi-incorporada en su tarjeta ideal para el funcionamiento de todo nuestro sistema, todas las características se encuentran en el Anexo N°5.

3.2.2 Diseño del circuito de conexión con la tarjeta.

Como se indica en la Figura N°24, el esquema electrónico de la tarjeta para poder hacer la comunicación con la balanza. Donde se conectan los cables de la balanza SUPER SS (VCC, GND, SCK, MISO, CS) con la bornera, la cual los de comunicación SPI van al regulador de voltaje hacia el ESP32 esto se hace para la protección del microcontrolador ya que los pines de I/O discreta funcionan a de 3.3v y el protocolo SPI de la balanza es a 5v.

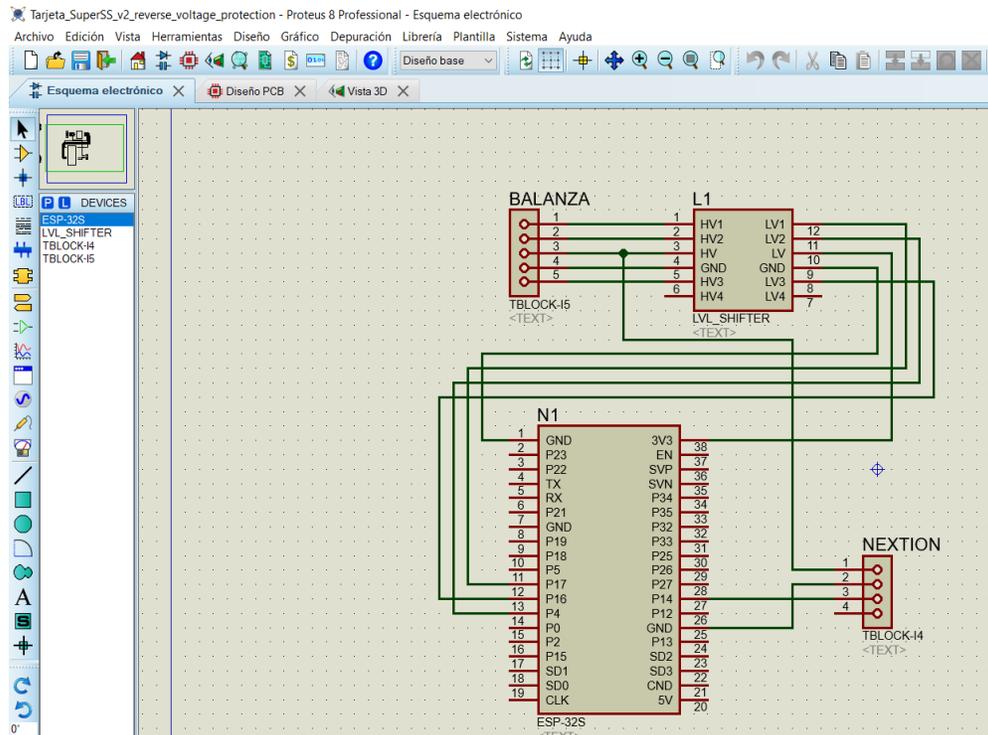


Figura N°24: Esquema electrónico de la tarjeta desarrollado en Proteus

Fuente: Elaboración propia

Al tener el desarrollo del esquema electrónico correctamente ejecutado se procede a continuar con el siguiente paso; el diseño digital PCB de la tarjeta con ello se desarrolla la implementación de dicha tarjeta. En la Figura N°25 visualizamos el desarrollo del diseño PCB.

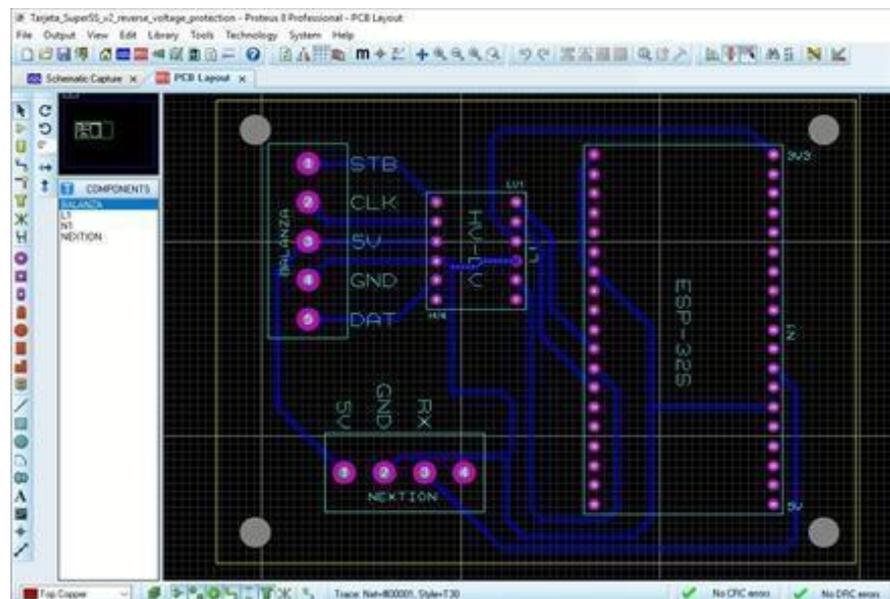


Figura N°25: Diseño de PCB de la tarjeta desarrollado en Proteus

Fuente: Elaboración propia

Con el diseño del PCB correctamente esquematizado se procede a visualizar en vista 3D, que permite el programa Proteus, esto permite verificar de una forma más detallada la tarjeta elaborada. Esta vista 3D se puede visualizar en la Figura N°26:

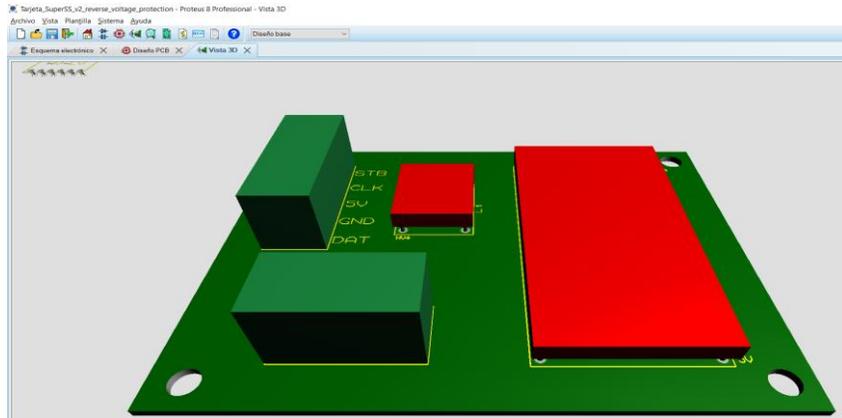


Figura N°26: Vista 3D de la tarjeta realizado desarrollado en Proteus

Fuente: Elaboración propia

En la Figura N°27, se visualiza la tarjeta hecha físicamente para la conexión con la balanza y la pantalla HMI. En donde se cuenta con la bornera para la conexión con la balanza SUPER SS, el regulador de voltaje, los pines para la vinculación con la pantalla Nextion y el microcontrolador ESP32.

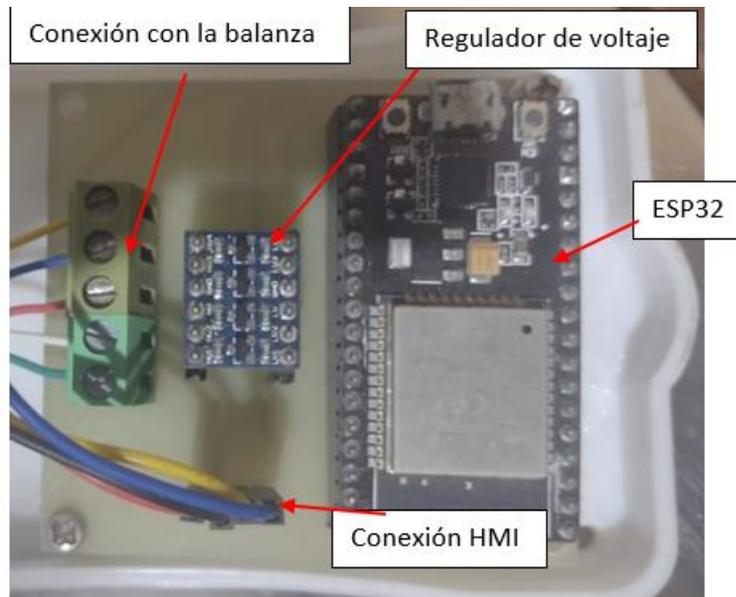


Figura N°27: Tarjeta electrónica en físico

Fuente: Elaboración propia

Al tener la tarjeta electrónica elaborada se puede hacer una prueba de funcionamiento de la recepción de la información por wifi, enviándola por un programa de captura llamado “Hyperterminal”. Este aplicativo sirve para poder comunicarse por TCP/IP con otros dispositivos y hacer la prueba de vinculación. En la Figura N°28 se visualiza la trama de la placa electrónica se envía correctamente.

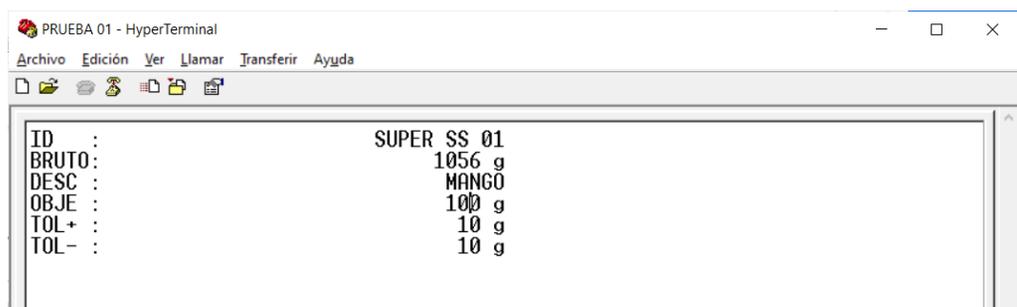


Figura N°28: Simulador Hyperterminal

Fuente: Elaboración propia

3.2.3 Conexión electrónica

En esta Figura N°29, se muestra como es la conexión física de la tarjeta desarrollada con la balanza SUPER SS y la pantalla Nextion:

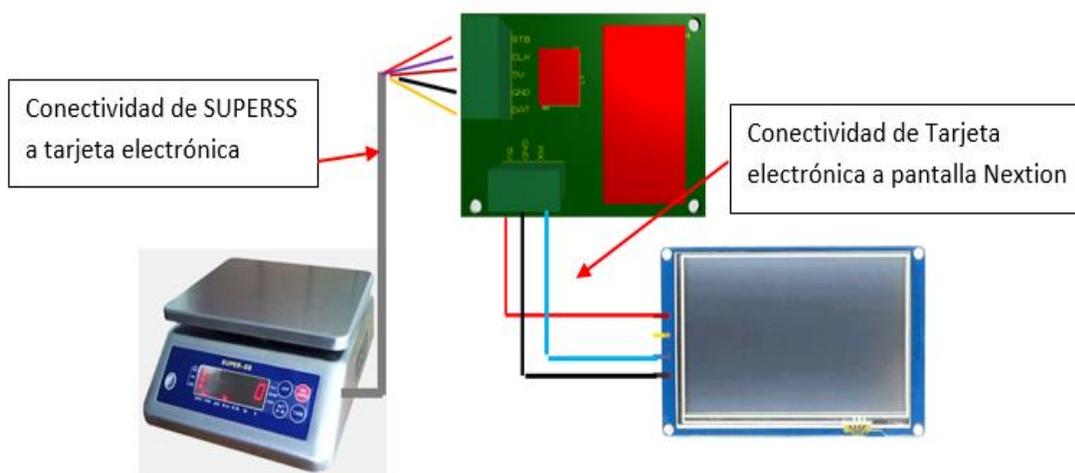


Figura N°29: Diagrama electrónico

Fuente: Elaboración propia

3.2.4 Programación General Rutina de la tarjeta Electrónica

Desarrollo del diagrama de flujo para la lógica de programación del microcontrolador, para la captura de información de la balanza SUPER SS y envío de esta información capturada hacia el software de gestión.

3.3.5 Diagrama de flujo del funcionamiento del sistema electrónico

En el esquema mostrado de la Figura N°30, indica primero la conectividad hacia la red donde está vinculado todo el sistema, esto se logra poniendo el nombre de la red y contraseña correspondiente en el programa del ESP32 y que el router esté funcionando correctamente. Al tener la primera parte correctamente se pasa a la segunda parte que es la consulta de nueva configuración. Esto significa sobre si el software de gestión quiere cambiar algún parámetro específico del sistema como la descripción del producto, objetivo, tolerancias, entre otros. A su vez el sistema está capturando la información de la balanza para su visualización al obtener dicha trama la tarjeta mandará el dato peso a la pantalla Nextion para su

visualización y para el programa de gestión los parámetros actuales de la balanza con el peso obtenido.

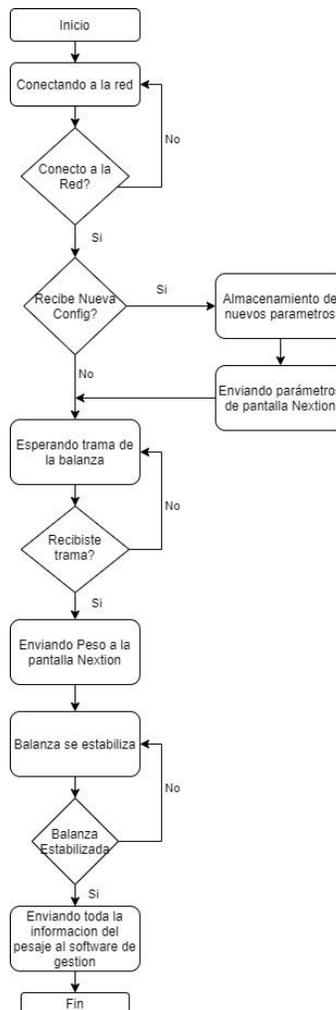


Figura N°30: Diagrama de flujo del programa ESP32

Fuente: Elaboración propia

3.2.6 Pantalla para Interfaz usuario máquina

Se optó por un HMI por el bajo costo, comunicación simple y rápida con el microcontrolador. En esta etapa se efectúa la selección de los equipos electrónicos para el procesamiento de la señal, se desarrolla el diagrama de bloques de la programación, diseño del entorno interfaz gráfico para el usuario y esquema electrónico de la tarjeta.

Para la selección de la pantalla se tiene presente una característica importante que es el precio debido a que el componente debe ser lo suficientemente económico para poder ser comercializado a gran escala para las empresas. Se escogió por la familia Nextion ya que cuenta con todo lo necesario para el proyecto ya que cuentan con un precio accesible a gran escala a diferencia de otras marcas. En la Tabla N°9 se escogió 3 HMI Nextion basándose en su costo.

Tabla N°9: HMI Nextion posibles a utilizar

Especificaciones/Tarjeta	NX3224F028	NX4832T035	NX4827P043
Marca	NEXTION	NEXTION	NEXTION
Tipo de comunicación	RS232	RS232	RS232
Flash	4MB	16 MB	128 MB
Ram	3584 Byte	3584 Byte	512 KB
Costo \$/.	20.4	39.9	50.5

Fuente: Elaboración propia

Se optó por el modelo NX4832T035 ya que es la gama estándar de la familia Nextion el cual cuenta con la comunicación serial, la memoria flash superior a los 12 MB que se requiere en el sistema para tener una comunicación fluida y sin complicaciones. Por último, su menor costo que es 39.9 dólares a comparación a una versión superior de pantalla Nextion NX4827P043.

3.2.7 Configuración de la pantalla Nextion

En la Figura N°31, se muestra el diseño para la visualización de la pantalla HMI. Donde lo principal de esta pantalla es brindar el dato del peso para que el operario pueda ver el objetivo al cual debería llegar el pesaje y sus tolerancias, con una descripción de lo que se está pesando. Además, la pantalla cambia de color dependiendo el pesaje.

- Si el peso es menor de la tolerancia mínima, la pantalla se presenta en amarillo.

- Si el peso es mayor de la tolerancia máxima, la pantalla se presenta en rojo.
- Si el peso está dentro del rango, la pantalla se presenta en verde.

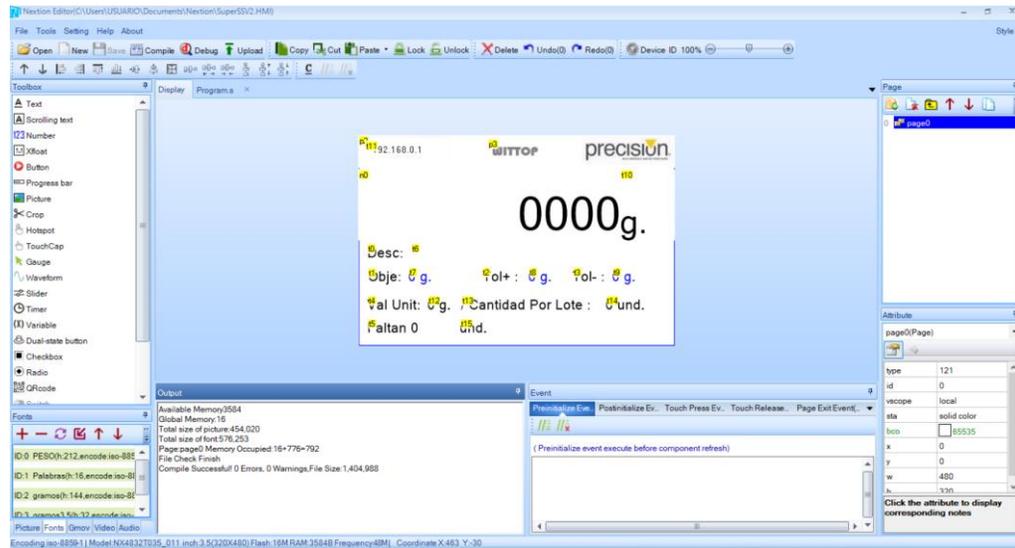


Figura N°31: Elaboración de la pantalla desarrollado en Nextion Editor

Fuente: Elaboración propia

Para carga el programa a la pantalla HMI física se puede hacer de dos maneras:

- Descarga por serial RS232
- Carga por tarjeta SD directo a la pantalla

En la Figura N°32, podemos visualizar la descarga por serial RS232 hacia la pantalla donde se debe encontrar el puerto COM del dispositivo para vincular con el programa Nextion Editor para poder cargar la configuración realizada.

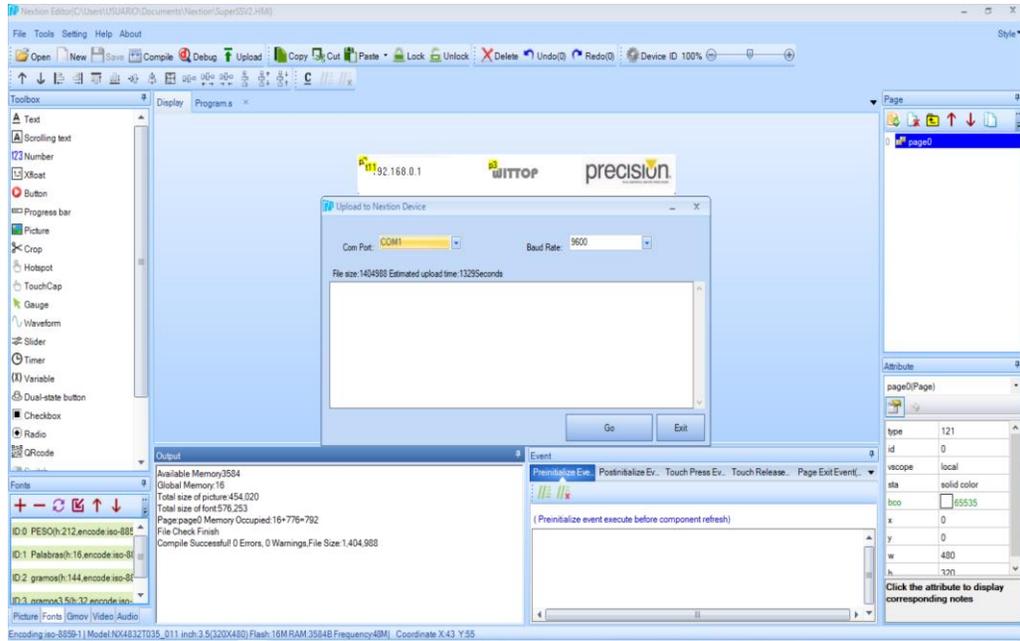


Figura N°32: Cargo de la pantalla Nextion desarrollado en Nextion Editor

Fuente: Elaboración propia

En la siguiente Figura N°33, podemos visualizar la pantalla cargada en el HMI físico de forma correcta donde se indica el pesaje de la balanza obtenida, la descripción del pesaje, el objetivo y las tolerancias.

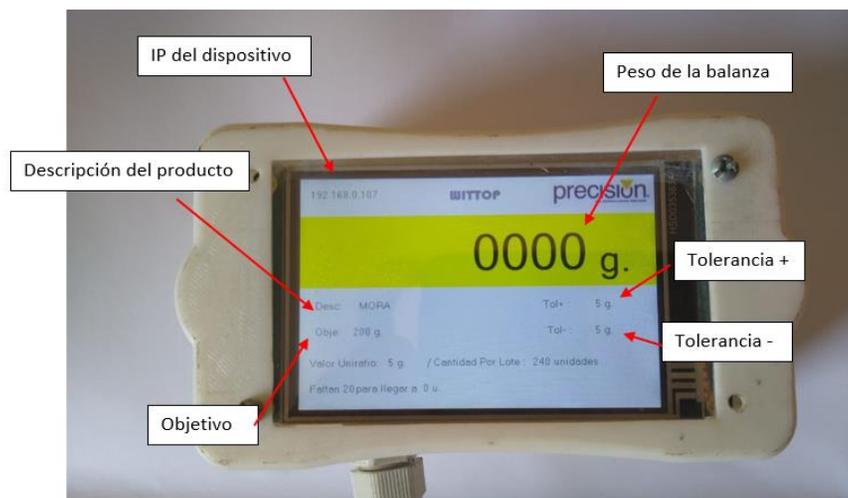
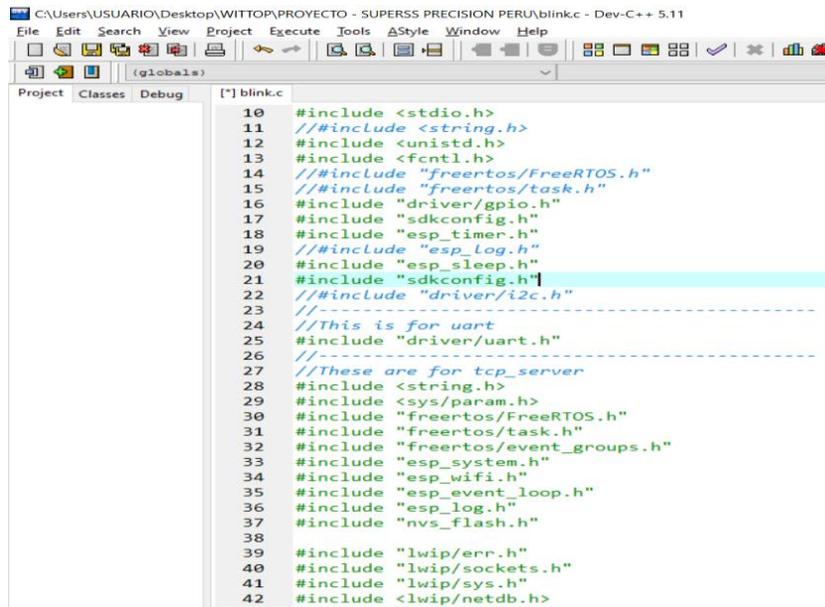


Figura N°33: Pantalla Nextion Física

Fuente: Elaboración propia

3.2.8 Desarrollo de la programación del microcontrolador ESP32

Se empieza utilizando las librerías correspondientes para la programación como el de esp, spi, wifi y comunicación tcp. Estas librerías nos facilitan a poder trabajar correctamente en la programación del microcontrolador. Todas estas librerías se pueden apreciar en la Figura N°34:



```
10 #include <stdio.h>
11 // #include <string.h>
12 #include <unistd.h>
13 #include <fcntl.h>
14 // #include "freertos/FreeRTOS.h"
15 // #include "freertos/task.h"
16 #include "driver/gpio.h"
17 #include "sdkconfig.h"
18 #include "esp_timer.h"
19 // #include "esp_log.h"
20 #include "esp_sleep.h"
21 #include "sdkconfig.h"
22 // #include "driver/i2c.h"
23 //-----
24 // This is for uart
25 #include "driver/uart.h"
26 //-----
27 // These are for tcp_server
28 #include <string.h>
29 #include <sys/param.h>
30 #include "freertos/FreeRTOS.h"
31 #include "freertos/task.h"
32 #include "freertos/event_groups.h"
33 #include "esp_system.h"
34 #include "esp_wifi.h"
35 #include "esp_event_loop.h"
36 #include "esp_log.h"
37 #include "nvs_flash.h"
38
39 #include "lwip/err.h"
40 #include "lwip/sockets.h"
41 #include "lwip/sys.h"
42 #include <lwip/netdb.h>
```

Figura N°34: Declaración de librerías

Fuente: Elaboración propia

En la Figura N°35, se muestra algunas de la variable a utilizar al programa como los del SPI “low_state”, “ligh_state”, “toma_datos”, “low_clock”, los dígitos de del pesaje “primera”, “segunda”, “tercera”, “cuarta”. la agrupación de los dígitos “cifra”.

```

106 bool low_state=false;
107 bool high_state=false;
108 long long int t_pasado;
109 long long int t_actual;
110 bool referencia_final=false;
111 int flancos_subida=0;
112 bool toma_datos=false;
113 bool low_clock=false;
114 int trama[100];
115 int i=0;
116 int count=0;
117
118 int digito;
119 int cuarta;
120 int tercera;
121 int segunda;
122 int primera;
123 int peso;
124 int cifra[7];
125 int pesos[2];
126 char num_c[5]; //Evaluar si num_c es necesario
127 int contador_filtro=0;
128 bool filtro_encontrado=false;
129 int camb=0;
130 int cambiopeso[2];
131 bool nuevo_peso=false;
132 bool hab_cuarta=false;
133 bool es_cuatro_cifras=false;
134

```

Figura N°35: Declaración de variables

Fuente: Elaboración propia

En la Figura N°36, se muestra la función para la obtención de cada dígito de la balanza en formato binarios por spi. Donde la “primeraiif” es la obtención del primer dígito, “segundacif” es la obtención del segundo dígito, “terceracif” es la obtención del tercer dígito y “cuartacif” es la obtención del cuarto dígito.

```

277     for(int k=0; k<7; k++){
278         cifra[k]=trama[j+k];
279     }
280     return true;
281 }else{
282     for(int k=0; k<7; k++){
283         cifra[k]=0;
284     }
285     return true;
286 }
287 }
288 }
289
290 bool primeracif(){
291     int j=84;
292     if(trama[j]==1){
293         return false;
294     }else{
295         do{
296             j++;
297         }while(trama[j]!=1);
298         for(int k=0; k<7; k++){
299             cifra[k]=trama[j+k];
300         }
301         return true;
302     }
303 }

```

Figura N°36: Obtención de cada dígito en formato binario

Fuente: Elaboración propia.

A continuación, se presenta en la Figura N°37, la función para convertir la información del dígito obtenido en binario a decimal haciendo sentencias IF.

```

302 }
303 }
304
305
306 int interpretacion(){
307
308 if(cifra[0]==0 && cifra[1]==0 && cifra[2]==1 && cifra[3]==0 && cifra[4]==1 && cifra[5]==0 && cifra[6]==0){
309     return 1;
310 } else if(cifra[0]==1 && cifra[1]==0 && cifra[2]==1 && cifra[3]==1 && cifra[4]==0 && cifra[5]==1 && cifra[6]==1){
311     return 2;
312 } else if(cifra[0]==1 && cifra[1]==0 && cifra[2]==1 && cifra[3]==1 && cifra[4]==1 && cifra[5]==1 && cifra[6]==0){
313     return 3;
314 } else if(cifra[0]==0 && cifra[1]==1 && cifra[2]==1 && cifra[3]==1 && cifra[4]==1 && cifra[5]==0 && cifra[6]==0){
315     return 4;
316 } else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==0 && cifra[3]==1 && cifra[4]==1 && cifra[5]==1 && cifra[6]==0){
317     return 5;
318 } else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==0 && cifra[3]==1 && cifra[4]==1 && cifra[5]==1 && cifra[6]==1){
319     return 6;
320 } else if(cifra[0]==1 && cifra[1]==0 && cifra[2]==1 && cifra[3]==0 && cifra[4]==1 && cifra[5]==0 && cifra[6]==0){
321     return 7;
322 } else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==1 && cifra[3]==1 && cifra[4]==1 && cifra[5]==1 && cifra[6]==1){
323     return 8;
324 } else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==1 && cifra[3]==1 && cifra[4]==1 && cifra[5]==1 && cifra[6]==0){
325     return 9;
326 } else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==1 && cifra[3]==0 && cifra[4]==1 && cifra[5]==1 && cifra[6]==1){
327     return 0;

```

Figura N°37: Conversión de dígito binario a decimal

Fuente: Elaboración propia.

En la Figura N°38, se muestra la función de la configuración y habilitación de la antena wifi del ESP32 para la conectividad de la red de la empresa.

```

398 }
399
400 static void initialise_wifi(void)
401 {
402     tcpip_adapter_init();
403     wifi_event_group = xEventGroupCreate();
404     ESP_ERROR_CHECK( esp_event_loop_init(event_handler, NULL) );
405     wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT();
406     ESP_ERROR_CHECK( esp_wifi_init(&cfg) );
407     ESP_ERROR_CHECK( esp_wifi_set_storage(WIFI_STORAGE_RAM) );
408     wifi_config_t wifi_config = {
409         .sta = {
410             .ssid = EXAMPLE_WIFI_SSID,
411             .password = EXAMPLE_WIFI_PASS,
412         },
413     };
414     ESP_LOGI(TAG, "Setting WiFi configuration SSID %s...", wifi_config.sta.ssid);
415     ESP_ERROR_CHECK( esp_wifi_set_mode(WIFI_MODE_STA) );
416     ESP_ERROR_CHECK( esp_wifi_set_config(ESP_IF_WIFI_STA, &wifi_config) );
417     ESP_ERROR_CHECK( esp_wifi_start() );
418 }
419
420 static void wait_for_ip()
421 {
422     uint32_t bits = IPV4_GOTIP_BIT | IPV6_GOTIP_BIT ;
423
424     ESP_LOGI(TAG, "Waiting for IP connection...");

```

Figura N°38: Habilitación de la antena wifi del ESP32

Fuente: Elaboración propia

En la Figura N°39, se muestra la vinculación de las cifras convertidas en decimales individuales en un solo número con el sistema decimal, al obtener el peso correspondiente manda la información a la pantalla nextion

automáticamente para que se visualice de forma instantánea y cuando encuentre estabilidad de peso lo manda por tcp/ip al software de PC. Ver Anexo N°5.

```

1250     primera=digito;
1251     //Serial.println(primer);
1252 }else{
1253     primera=0;
1254     //Serial.println("N");
1255 }
1256 }else{
1257     primera=0;
1258     //Serial.println("P");
1259 }
1260
1261 if(contador_filtro==0){
1262     pesos[0]=1000*cuarta+100*tercera+10*segunda+primera;
1263     contador_filtro++;
1264     printf("%d\n",pesos[0]);
1265 }else{
1266     pesos[1]=1000*cuarta+100*tercera+10*segunda+primera;
1267     itoa(pesos[1], peso_c, 10);
1268     //if(hab_cuarta){
1269     uart_write_bytes(UART_NUM_1, "n0.val=", 7);
1270     uart_write_bytes(UART_NUM_1, peso_c, strlen(peso_c));
1271     uart_write_bytes(UART_NUM_1, "\xff\xff\xff", 3);
1272     //}
1273 if(pesos[1]==pesos[0]){
1274     printf("%d\n",pesos[1]);
1275     //int j;
1276     filtro_encontrado=true;

```

Figura N°39: obtención de peso y envío de información

Fuente: Elaboración propia

3.2.9 Costo del sistema electrónico

La tarjeta electrónica tiene componentes fáciles de adquirir en el mercado lo cual son de un bajo costo para poder emplear esta solución por ello en la tabla N°10 se muestra la lista de precios, de los todos los componentes:

Tabla N°10: Tabla de costo

Especificaciones/Costo	Costo (\$/)
Convertidor de 5v a 3.3v	2
Microcontrolador ESP32	8.5
Pantalla Nextion	39.9
Placa	4.5
Borneras	1.5
Carcasa	25
Cables y Otros	15
Balanza Super SS	400

Fuente:

Elaboración propia

Con los precios obtenidos que se tiene se puede hacer una comparación de otros equipos de pesaje en el mercado que pueden contar con el mismo funcionamiento en la tabla N°11 podemos visualizar la comparación de costos.

Equipo de balanza	Costo (\$/)
IND 570	2300
ICS 685	1000
SUPER SS + Tarjeta electrónica	496.4

Tabla

N°11:

Tabla de

comparación de costos entre balanza similares

Fuente: Elaboración propia

Como se puede apreciar nuestro proyecto es un sistema de menor costo a comparación de otras balanzas que pueden hacer las mismas funciones.

3.3 Desarrollo de software del sistema y base de datos

En esta etapa se efectúa la selección de los programas para la aplicación y del almacenamiento de la base de datos.

3.3.1 Selección de software de aplicativo

Para la selección del software se tiene presente varios factores para su elección ya que tiene varios puntos que influyen para la creación y desarrollo del aplicativo. En la siguiente Tabla N°12 vemos dos opciones de las cuales se podría utilizar.

Tabla N°12: Software posibles a usar

Especificaciones/Software	LABVIEW	VISUAL STUDIO
Modo de lenguaje	Bloque G	C
Aplicación en Automatización Industrial	Si	No
Aplicación en TIC's y servicios. Tecnología de la información y Servicios	No	Si
Investigación	Si	No
Experiencia de los tesisistas	8 años	2 años
laboración propia		

Se optó el software LabVIEW debido a que es un software exclusivamente para adquisiciones de datos físicos, análisis de medidas, aplicaciones en automatización industrial y presentación de datos. También se escogió este aplicativo ya que contamos con una experiencia mucho mayor.

3.3.2 Desarrollo del sistema de pesaje usando LabVIEW

Para el desarrollo del aplicativo, se implementó en el programa LabVIEW para la obtención de la información de las balanzas, al tener estos datos se guardan en la base de datos SQL-server. El aplicativo también sirve para modificar los datos de los terminales del HMI y exportar la información por Excel.

En la Figura N°40, se muestra la pantalla principal del aplicativo para el usuario donde visualizamos los pesajes que llegan al sistema, luego la información recibida se guarda en la tabla que se encuentra en la imagen; estos registros son guardados en una base de datos SQL server. Al lado izquierdo tiene la visualización de la trama que está llegando al sistema y un registro de errores por si tiene algún inconveniente. Este aplicativo cuenta con un botón de exportar para poder enviar toda la información obtenida a un Excel para la manipulación del personal.

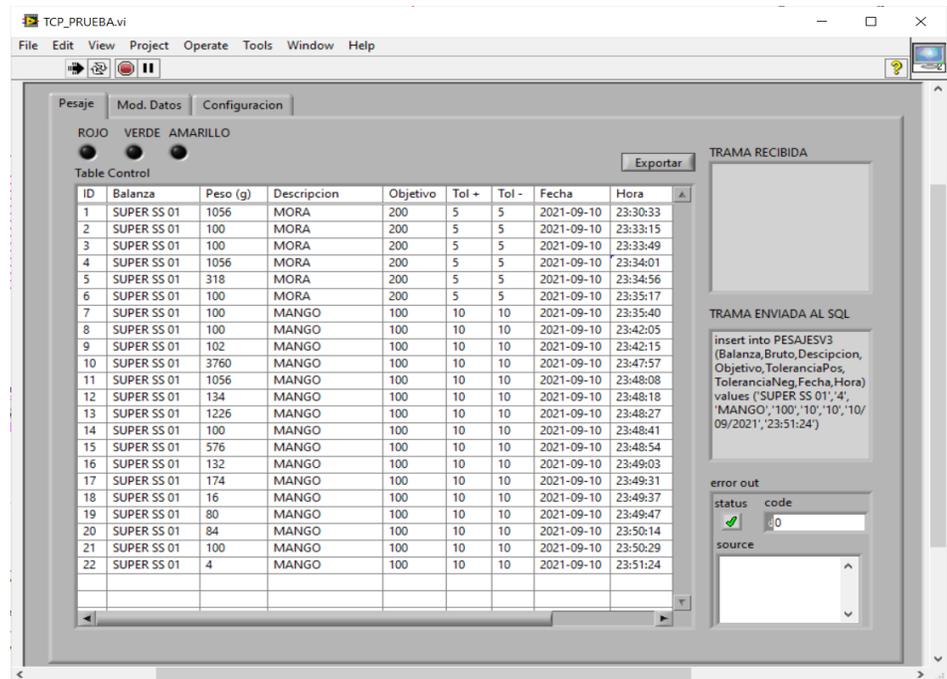


Figura N°40: Aplicativo LabVIEW visualización de pesaje

Fuente: Elaboración propia

En la Figura N°41, se muestra la ventana para la modificación de los datos de las pantallas HMI para que los operarios puedan saber el objetivo que debe alcanzar el pesaje, una descripción de los productos y las tolerancias máximas y mínimas de los pesos. Además, que en la pantalla se visualiza en tema del pesaje de colores indicando los siguiente:

Rojo → Pesaje Sobre la tolerancia Máxima

Amarillo → Pesaje debajo de la tolerancia Mínima

Verde → Pesaje dentro del rango de tolerancias.

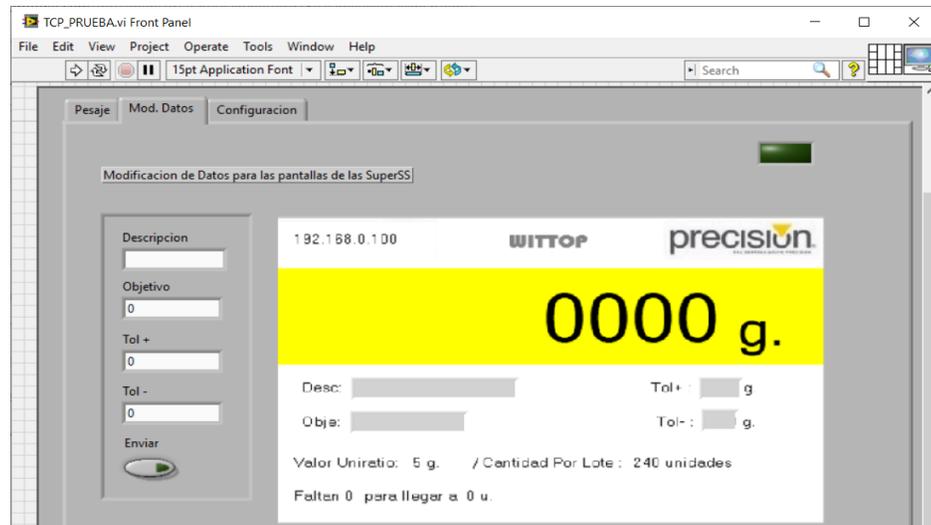


Figura N°41: Configuraciones de la pantalla realizado en LabVIEW

Fuente: Elaboración propia

En la Figura N°42, se muestra la ventana para la conexión de las balanzas donde debemos poner las IP y puertos de las balanzas que queremos conectar al sistema. Tiene el botón de conectar para vincularse y el botón de stop para detener la vinculación aparte de un visualizador led como indicador de texto para ver si está conectado o desconectado la balanza.

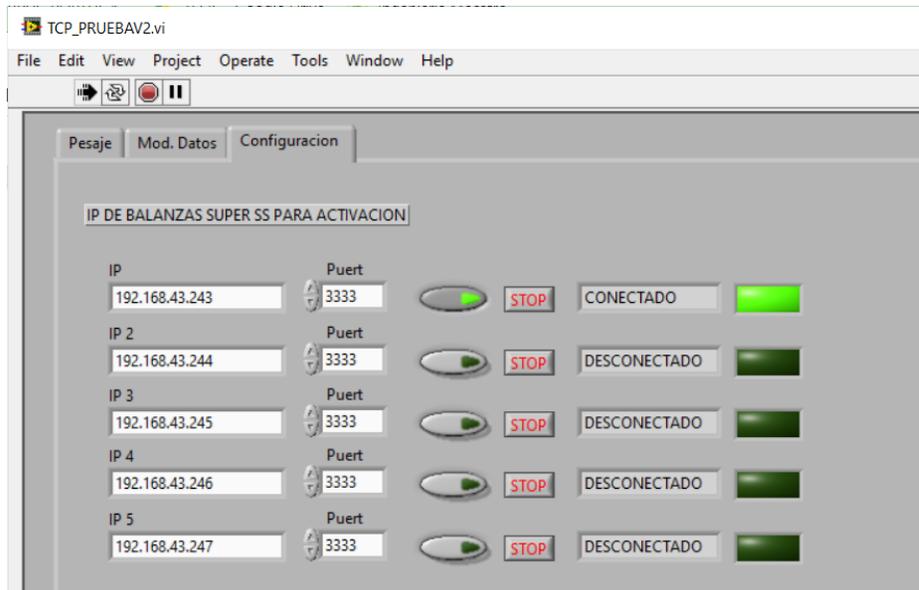


Figura N°42: Modificación de las configuraciones de la pantalla en LabVIEW

Fuente: Elaboración propia

En la Figura N°43, se observa el aplicativo de simulación “Hércules” para poder visualizar si el programa de envío de datos desarrollado en LabVIEW funciona de manera correcta. Este programa funciona como un simulador de balanza para poder capturar la trama que envía el software para la configuración de información de la pantalla Nextion.

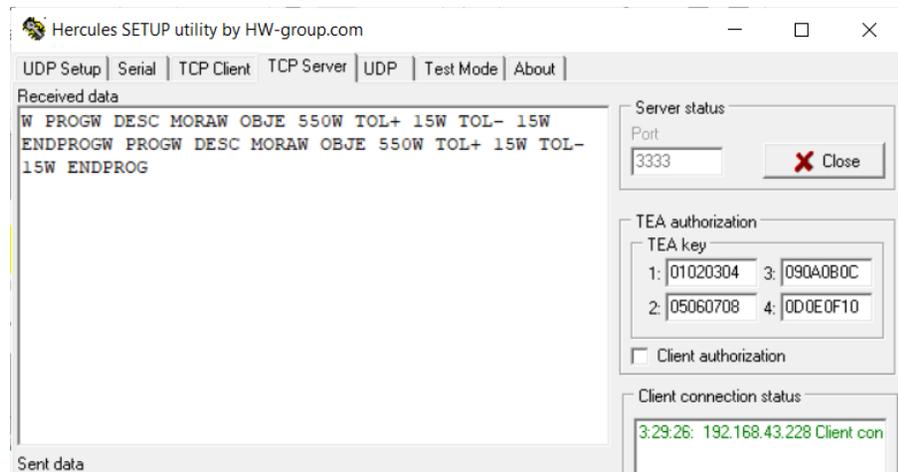


Figura N°43: Envío de trama del software desarrollado en LabVIEW hacia simulador

Fuente: Elaboración propia

Trama que recibía el simulador:

- W PROG ----> Inicio del envío de información
- W DESC MORA ----> Envío de la descripción
- W OBJE 550 ----> Envío del objetivo
- W TOL+ 15 ----> Envío de la tolerancia +
- W TOL- 15 ----> Envío de la tolerancia -
- W ENDPROG ----> Fin del envío de la información

En la Figura N°44, se muestra la vinculación del SQL Server hacia el software LabVIEW para poder enviar la información que registre el aplicativo para este enlace se necesita los parámetros de Nombre del servidor, nombre de la base de datos, nombre de usuario y contraseña.

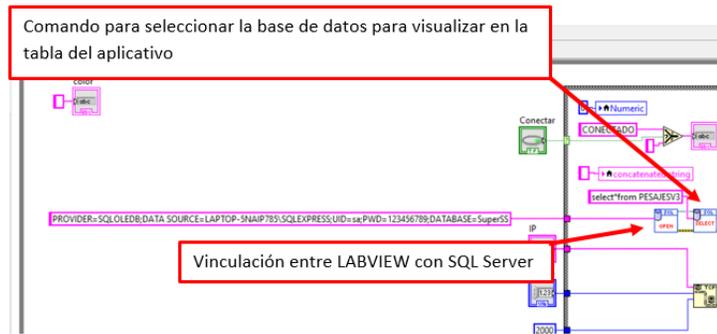


Figura N°44: Modificación de las configuraciones de la pantalla realizado en LabVIEW

Fuente: Elaboración propia

Para la comunicación entre las tarjetas electrónicas con el software se utilizan los bloques tcp/ip del programa LabVIEW donde se tiene que conectar la ip y el puerto de la balanza habilitada, con ello la capturar la trama. Como muestra la Figura N°45, se aplica doble bloque de “TCP Read” para obtener toda la información respectiva de la tarjeta ya que con una no es suficiente debido a la posible pérdida de información ya que la trama no llegue en partes, para evitar esto se utiliza el bloque “Type cast” ya que ayuda a calcular todos los caracteres que falta y se pueda tener adquirir toda la trama, para visualizar el código completo de LabVIEW ver anexo N°6

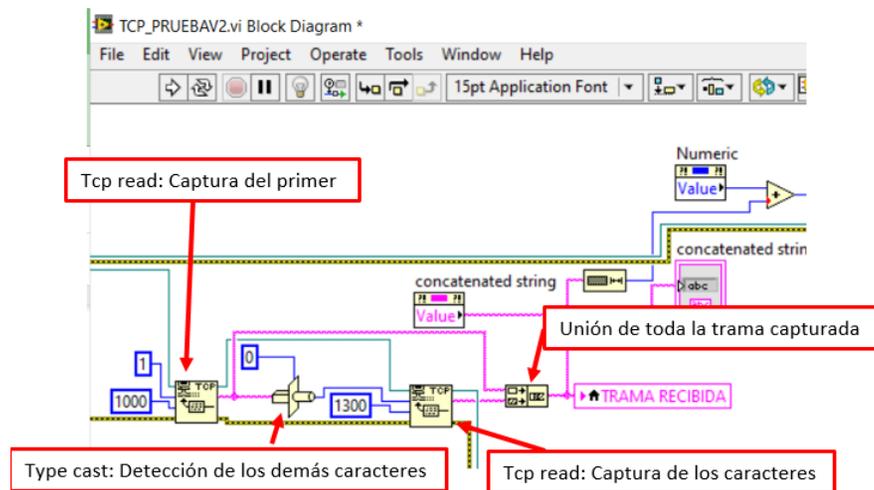


Figura N°45: Lectura de TCP de la trama de las tarjetas electrónicas en LabVIEW

Fuente: Elaboración propia

Para hacer la exportación de la base de datos a un archivo txt se utiliza el bloque de labview “write to spreadsheet file.vi” este se vincula con el comando del SQL para poder capturar la información de la base de datos. Se aprecia en la siguiente Figura N°46.

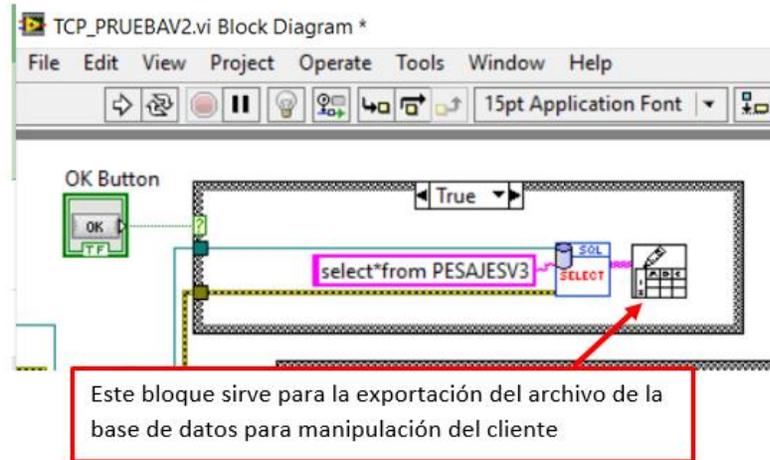


Figura N°46: Modificación de las configuraciones de la pantalla en LabVIEW

Fuente: Elaboración propia

En la Figura N°47, se muestra parte del programa donde fracciona la trama enviada de la balanza para poder guardarla en la base de datos para poder visualizarlo en el aplicativo

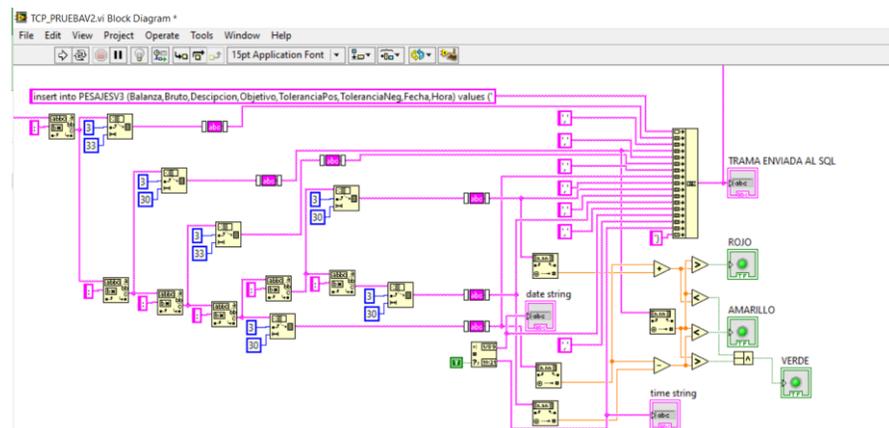


Figura N°47: Código para fraccionar trama realizado en LabVIEW

Fuente: Elaboración propia.

En la Figura N°48, se muestra parte del programa donde manda la modificación de las configuraciones de las balanzas donde se puede cambiar la descripción, objetivo y tolerancias.

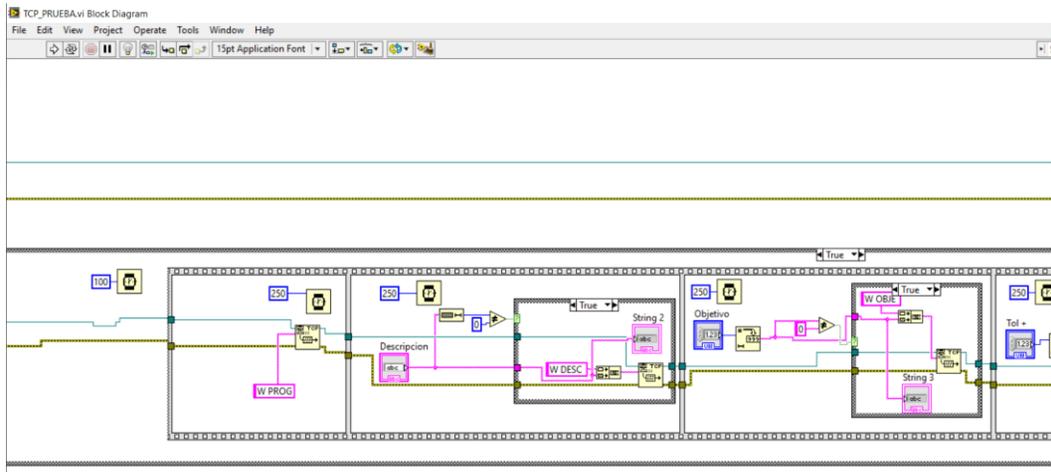


Figura N°48: Código para modificación de configuraciones realizado en LabVIEW

Fuente: Elaboración propia

3.3.3 Desarrollo de la base de datos mediante SQL SERVER

En la Figura N°49, se muestra el ingreso del SQL server para la creación de la base de datos. Se ingresa con autenticación de SQL Server con su inicio de sesión y contraseña.

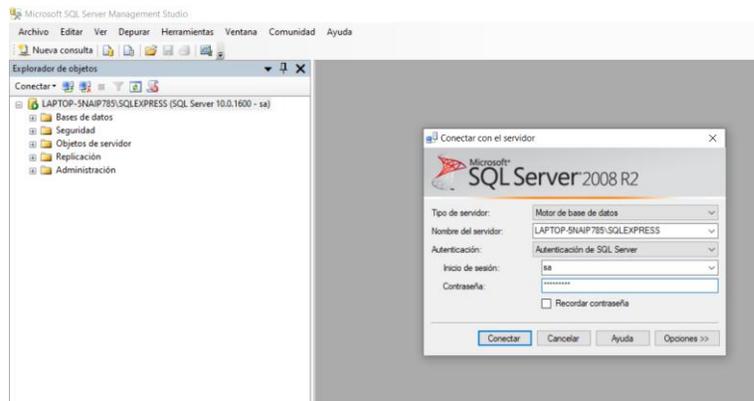


Figura N°49: Ingreso del SQL Server

Fuente: Elaboración propia

Para la vinculación del servidor con la PC se debe realizar dos pasos importantes. El primer paso de activar la autenticación de SQL con ello el aplicativo se puede conectar a la red, la Figura N°50, muestra dónde está la activación de la autenticación del SQL Server.

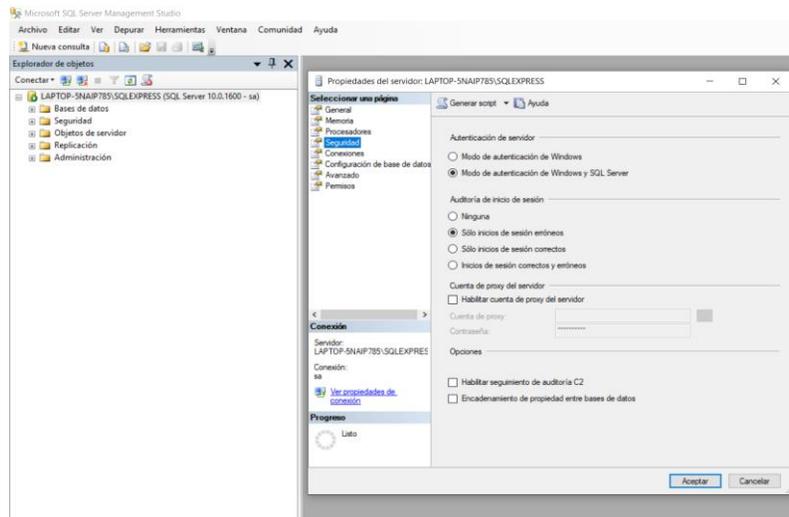


Figura N°50: Autenticación de SQL Server

Fuente: Elaboración propia

El segundo paso importante para la vinculación es activar el protocolo TCP/IP del SQL server con ello logra vincular cualquier computadora hacia la base de datos del servidor, el puerto de conexión es el 1433. En la Figura N°51, muestra la habilitación y el puerto TCP.

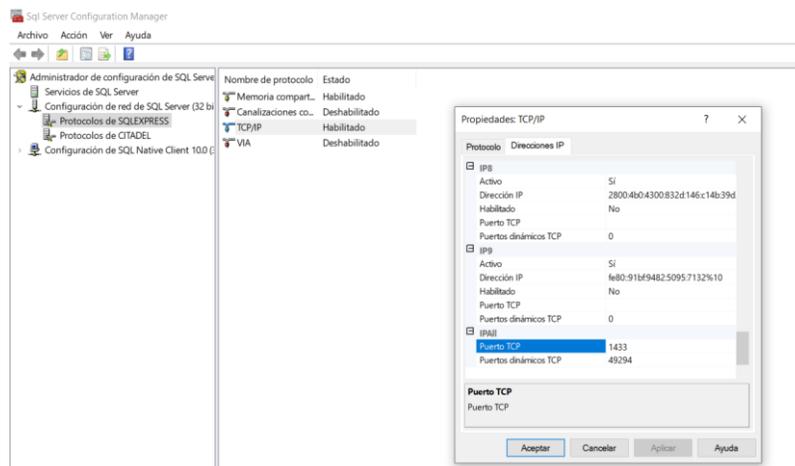


Figura N°51: Habilidad del TCP/IP y puerto de comunicación

Fuente: Elaboración propia

Se crea la base de datos con el nombre Super SS en el cual debe tener toda la información que se necesita. En la Figura N°52, se puede visualizar el diseño de la base de datos que creó.

D → int

Balanza → varchar (50)

Bruto → varchar (50)

Descripción → varchar (50)

Objetivo → varchar (50)

ToleranciaPos → varchar (50)

ToleranciaNeg → varchar (50)

Fecha → date

Hora → varchar (50)

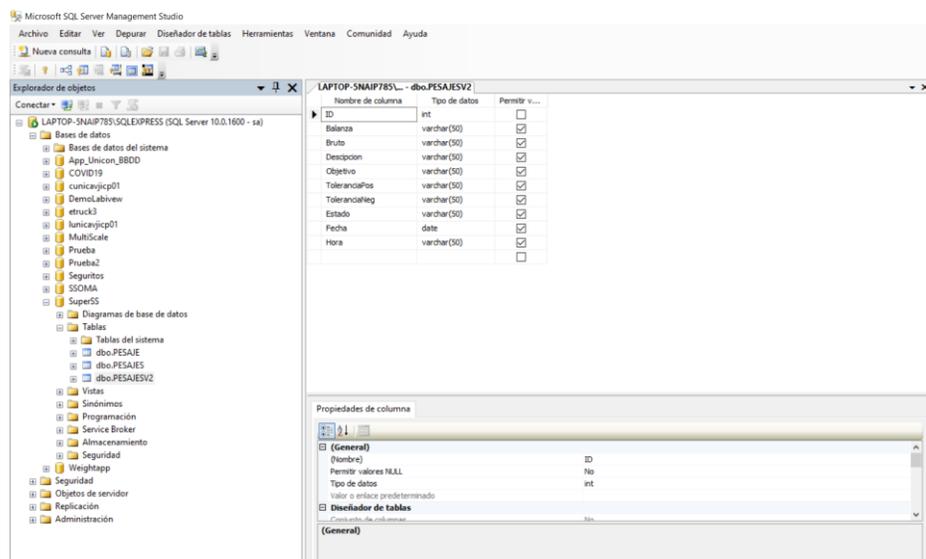


Figura N°52: Base de datos Super SS

Fuente: Elaboración propia

3.4 Diseño de la carcasa

Para el diseño de la protección de los componentes electrónicos, se necesita de material lo suficientemente resistente al agua ya que el modo de limpieza es a chorros constantemente. Como se ve en la imagen podemos apreciar el modelo de la carcasa

donde va ubicada la tarjeta electrónica y la pantalla nextion. En la Figura N°53, podemos observar el diseño de la parte mecánica.

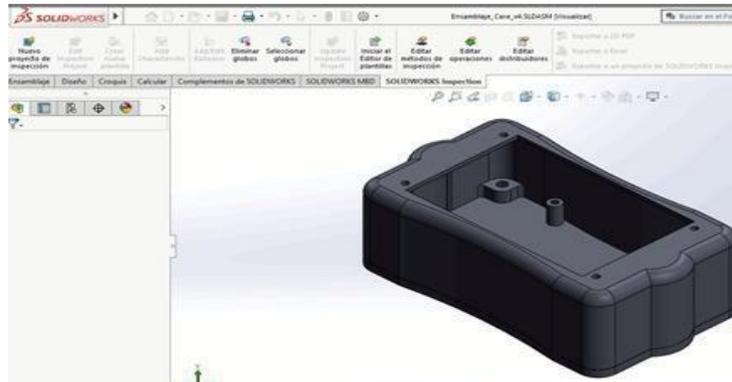


Figura N°53: Diseño de la carcasa de la tarjeta electrónica realizado en SolidWorks

Fuente: Elaboración propia

3.5 Implementación

En la Figura N°54, se visualiza la estructura interior de la balanza SUPERSS para poder hacer la conectividad de la tarjeta electrónica, para su enlace de comunicación se debe conectar los cables de la comunicación entre el display principal con el secundario.

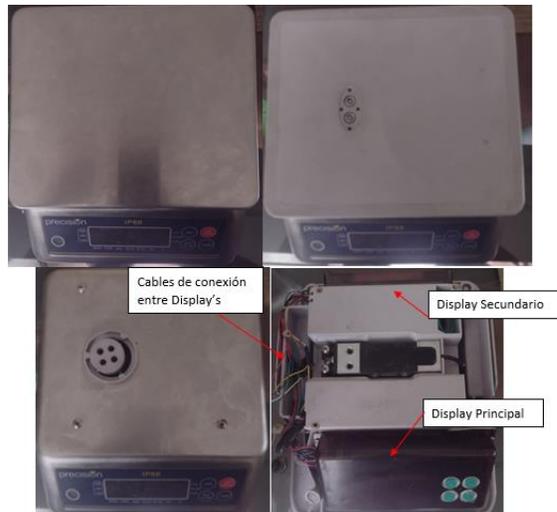


Figura N°54: Estructura interna de la balanza SUPER SS

Fuente: Elaboración propia

En la Figura N°55, se visualiza la conexión del cableado de la tarjeta electrónica para poder jalar su comunicación entre los Display's de la balanza SUPER SS. La conectividad del cableado se hace con la información de chip del display (FD628) donde encontramos que la conectividad que utiliza interfaz serial (CLK, STB, DATA). Para el Protocolo SPI se tiene (CLK, CS, MISO). En el anexo 6 se adjunta hoja técnica del chip FD628.



Figura N°55: Conectividad de la tarjeta electrónica SUPERSS

Fuente: Elaboración propia

CAPÍTULO IV: PRESENTACIÓN Y ANÁLISIS DE RESULTADOS

En este capítulo se detalla los resultados que se obtuvieron al realizar el diseño de la investigación en base a los objetivos planteados en el anterior capítulo. Con esto podremos evaluar qué tan factible y productivo es este proyecto de tesis.

4.1 Conectividad inalámbrica entre los dispositivos

Para la conectividad se hizo la prueba de vinculación por el programa Símbolo del sistema que cuenta la PC para poder visualizar conexión de la tarjeta. El sistema está configurado si detecta la red WiFi específica (Nombre de red: WITTOP, contraseña: WITTOP04) visualiza la IP en la parte superior izquierda de la pantalla Nextion, caso contrario aparece vacío. En la Figura N°56 se muestra el ping de la tarjeta electrónica con la computadora lo cual demuestra que hay una conexión.

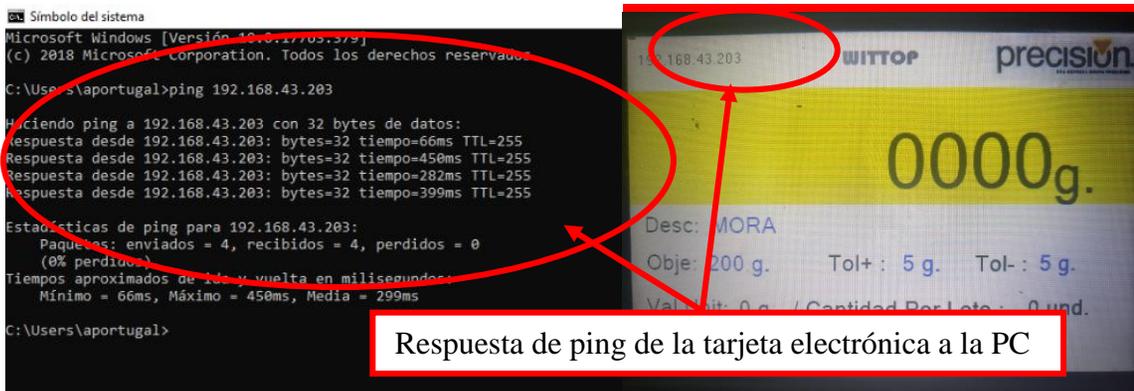


Figura N°56: Ping de tarjeta electrónica a PC

Fuente: Elaboración propia

4.2 Sistema electrónico

Para las pruebas de comunicación entre la balanza SUPER SS y tarjeta electrónica se hizo el pesaje de algunos productos para visualizar la captura de la trama de la balanza para poder visualizarlo en la pantalla Nextion. Como podemos visualizar en la Figura N°57 se obtiene el mismo pesaje.



Figura N°57: Captura de peso en tarjeta electrónica

Fuente: Elaboración propia

Obteniendo los pesajes de forma automática y registrándolo en el sistema de gestión se realizó un análisis estadístico del número de pesos realizados, la cual se compara con la forma manual que se hacían anteriormente. En la Figura N°58 se muestra el aumento de cantidad de pesaje registrado en 5 procesos de cinco minutos cada uno.

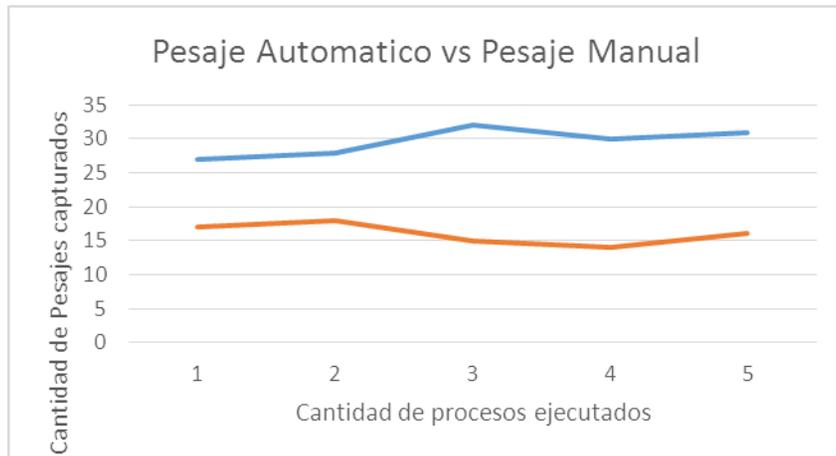


Figura N°58: Pesaje automático vs pesaje manual

Fuente: Elaboración propia

Adicionalmente el sistema electrónico cuenta con tres casos posibles casos a suceder en todo el proceso del funcionamiento del sistema. En las pruebas se utiliza los siguientes parámetros en la tarjeta:

- Descripción del alimento: Fresas
- Cantidad por lote: 10 unidades
- Valor Unitario de fresa: 17 gramos
- Objetivo: 175 gramos
- Tolerancia máxima: 10 gramos
- Tolerancia mínima: 10 gramos

En la Figura N°59, se muestra el primer caso cuando el pesaje no es el suficiente para alcanzar la tolerancia mínima. En este evento la pantalla HMI se visualiza de color amarillo indicando al operario que el pesaje actual de la balanza no es lo suficiente para poder alcanzar el objetivo. En la parte inferior de la visualización indica la cantidad de fresas que faltan colocar en el recipiente.

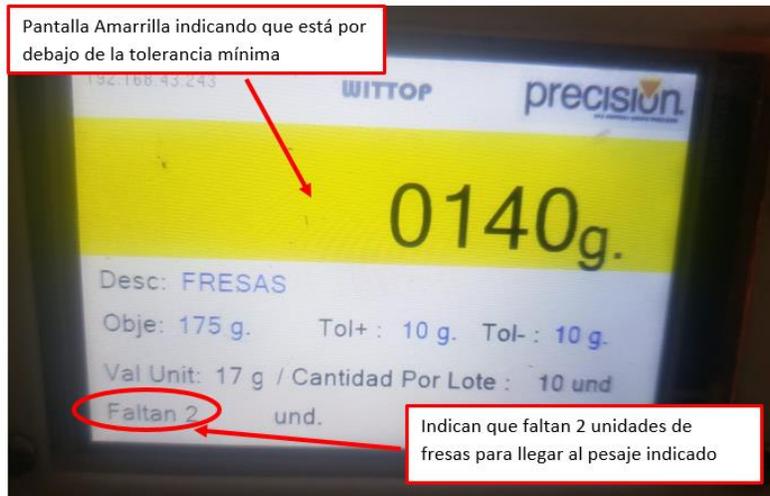


Figura N°59: Pantalla indicando que falta pesaje para llegar al objetivo

Fuente: Elaboración propia

En la Figura N°60, se muestra el segundo caso cuando el pesaje sobrepasa la tolerancia máxima. En este evento la pantalla HMI se visualiza de color rojo indicando al operario que el pesaje actual de la balanza es demasiado alto que el objetivo. En la parte inferior de la visualización indica la cantidad de fresas que debes retirar del recipiente.

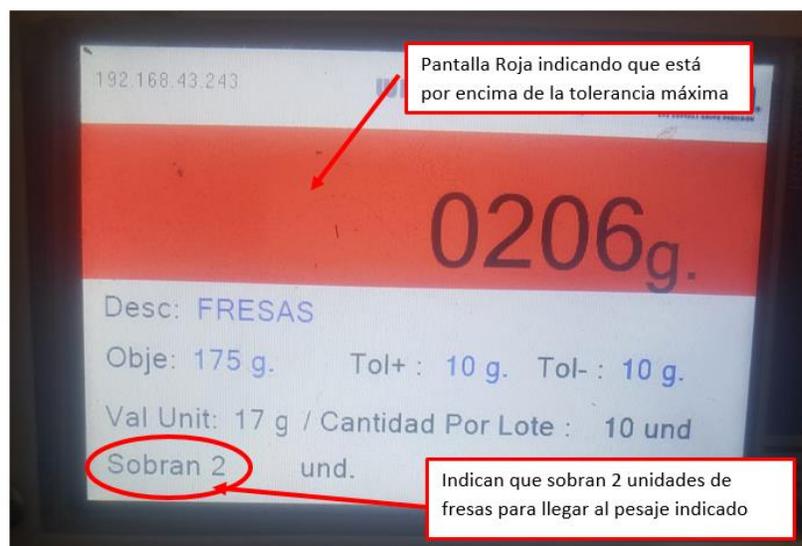


Figura N°60: Pantalla indicando que sobra pesaje para llegar al objetivo

Fuente: Elaboración propia

En la Figura N°61, se muestra el tercer caso cuando el pesaje está dentro de las tolerancias. En este evento la pantalla HMI se visualiza de color verde indicando al operario que el pesaje actual de la balanza es el correcto. En la parte inferior de la visualización indica que es el exacto.

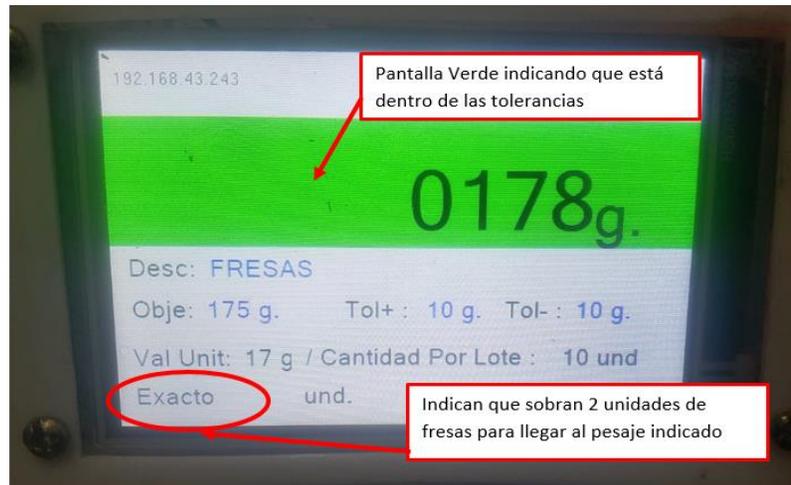


Figura N°61: Pantalla indicando que sobra pesaje para llegar al objetivo

Fuente: Elaboración propia

En la Figura N°62 se muestra un cuadro de las cantidades de los pesos realizados por el sistema; brindado los datos de estas pruebas 17 pesajes realizadas por el sistema.



Figura N°62: Pantalla indicando que sobra pesaje para llegar al objetivo

Fuente: Elaboración propia

4.3 Software de gestión y base de datos

En el aplicativo de gestión se lleva todos los registros de los pesajes enviados desde la tarjeta electrónica para poder direccionar a la base de datos SQL server y a su vez

visualizarlo en el software. El envío de información hacia el software es de manera que tiene un retardo de 250 milisegundos para su captura, entre más balanzas se conecte al dispositivo el retardo de captura se puede aumentar y tener un tiempo de demora para extenso. En la Figura N°63, se ve el almacenamiento del pesaje capturado de la balanza SUPERS SS con el aplicativo de gestión creado en LabVIEW.

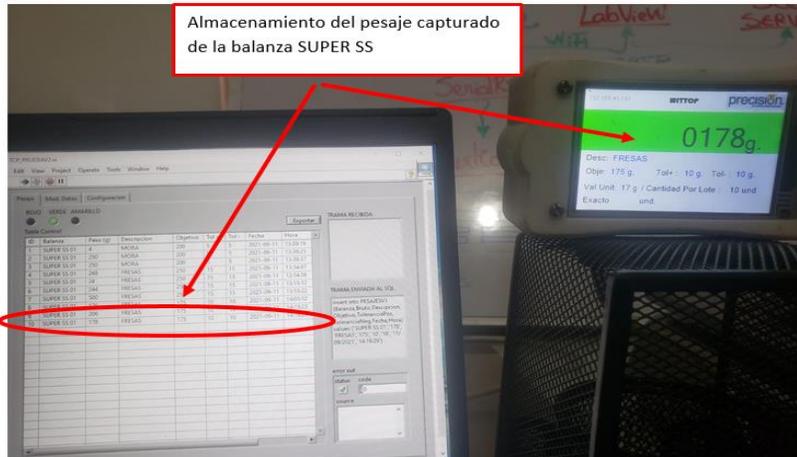


Figura N°63: Visualización del software de gestión de pesaje realizado en LabVIEW

Fuente: Elaboración propia

En la Figura N°64, se observa la exportación en txt del archivo del SQL server para manipulación del cliente.

Resgistro de datos: Bloc de notas

Archivo	Edición	Formato	Ver	Ayuda					
1	SUPER SS 01	4	MORA	200	5	5	2021-09-11	13:39:19	
2	SUPER SS 01	250	MORA	200	5	5	2021-09-11	13:39:25	
3	SUPER SS 01	250	MORA	200	5	5	2021-09-11	13:39:37	
4	SUPER SS 01	248	FRESAS	250	15	15	2021-09-11	13:54:07	
5	SUPER SS 01	24	FRESAS	250	15	15	2021-09-11	13:54:58	
6	SUPER SS 01	244	FRESAS	250	15	15	2021-09-11	13:55:12	
7	SUPER SS 01	500	FRESAS	250	15	15	2021-09-11	13:55:22	
8	SUPER SS 01	176	FRESAS	175	10	10	2021-09-11	14:05:52	
9	SUPER SS 01	206	FRESAS	175	10	10	2021-09-11	14:15:23	
10	SUPER SS 01	178	FRESAS	175	10	10	2021-09-11	14:16:29	

Figura N°64: Exportación en txt de la base de datos SQL Server

Fuente: Elaboración propia

Obteniendo los registros de forma automática en el sistema de gestión se realizó un análisis estadístico de la velocidad de captura entre la cual se compara con la forma manual que se hacían anteriormente. En la figura N°65 se muestra la disminución de

tiempo para el registro de los pesajes registrado en 5 procesos de cinco minutos cada uno.

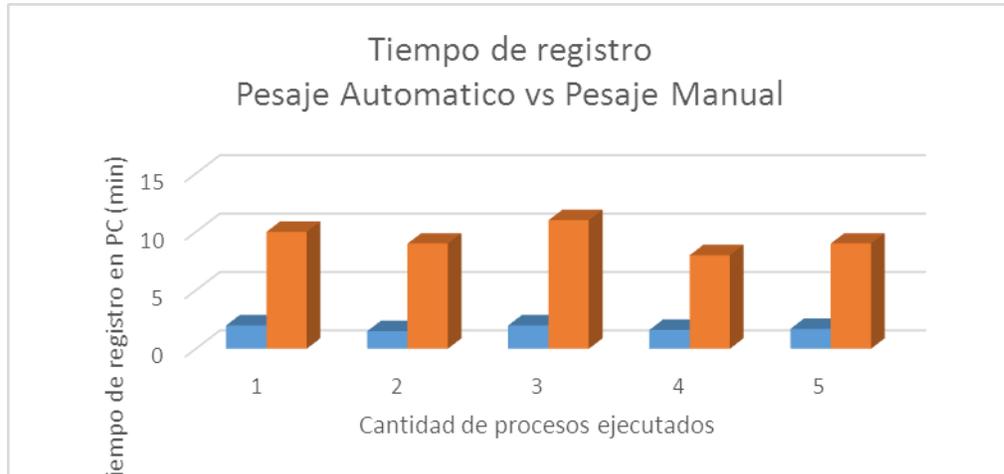


Figura N°65: Cuadro comparativo de velocidad de registro en computadora

Fuente: Elaboración propia

CONCLUSIONES

1. Se logró establecer una red en la cual la balanza SUPER SS con su tarjeta electrónica implementada pueda estar vinculada con el software de gestión y la base de datos obteniendo una conectividad entre dispositivos para su funcionamiento respectivo.
2. Ha sido optimizado el proceso de registro de modo manual como actualmente se hace con esta balanza al modo automático gracias al diseño de la tarjeta electrónica, lo cual hace que el registro de los pesos realizados en la balanza SUPER SS sea un 75% más eficiente versus una balanza convencional, con menor margen de error y una visualización en tiempo real de la información obtenida.

3. El software de gestión brinda una mayor administración de los pesajes obtenidos por las balanzas; lo cual brinda mayor facilidad a los colaboradores en contar con toda la información en su historial de registro.
4. La base de datos administra adecuadamente el almacenamiento de la información obtenida de las balanzas SUPER SS para tener un historial de todos los pesajes hechos por los usuarios, con esto tiene una mejor gestión.
5. Se logró implementar un diseño del sistema de pesaje de bajo costo a comparación de otras soluciones existentes en el mercado actual, lo cual hace que el sistema pueda ser comercializado de forma masiva en las empresas agroindustriales.

RECOMENDACIONES

1. Se recomienda para cambiar la conectividad inalámbrica del Wifi de la tarjeta electrónica a una conexión física se puede utilizar el módulo ethernet ENC28J60 ya que su programación es similar entre ambas en el microcontrolador ESP32.
2. Se recomienda para tener un mayor tiempo de uso de la balanza SUPER SS con la tarjeta electrónica utilizar una batería de mayor capacidad; ya que actualmente la balanza cuenta con una batería de menor capacidad lo cual hace que el tiempo de rendimiento de la misma menora, debido a que ahora también debe alimentar a la tarjeta electrónica.

3. Se recomienda para una mejor visualización de la información del HMI se cambie a una versión Nextion de mayores pulgadas. El tamaño de la pantalla no afecta a la programación realizada.
4. Es aconsejable que el encargado del aplicativo de gestión deba recibir una capacitación para la correcta manipulación; para evitar posibles errores e inconvenientes en el funcionamiento de este.
5. Se recomienda para el almacenamiento de la información de los pesajes registrados en SQL SERVER se puede utilizar tanto la versión gratuita como la versión pagada, ambas versiones no afectan en sistema desarrollado.
6. Se recomienda para mejorar el sistema de gestión agregar a la tarjeta electrónica un identificador de RFID, para conocer que operario está utilizando la balanza, con esto se tendría un mejor control y registro.
7. Se recomienda para mejorar el sistema de gestión agregar un desarrollo web para poder registrar la información hacia la nube y poder visualizarlo en cualquier parte ya no solo en la red implementada donde se encuentra las balanzas.

REFERENCIAS BIBLIOGRAFICAS

- Almidón A. y Julián E. (2018) *Manual de programación LABVIEW9.0 (Nivel Básico)* - Perú.https://www.researchgate.net/publication/328404013_Manual_de_programacion_LabVIEW_90
- Arevalo, G. (2018) “*Propuesta del diseño de red para la distribución de los dispositivos de conexión inalámbrica en la ciudad universitaria de la universidad nacional de san Martín - Tarapoto, año 2017*”. (Tesis de pregrado) Universidad Nacional de San Martín. Tarapoto, Perú.
- Bayona, V. (2020) *Implementación de un aplicativo móvil para la automatización de la toma de datos de pesaje para la empresa pesquera Terranova S.A.C. (Tesis de pregrado)* Universidad Nacional de Piura. Perú, Piura.
- Benito, A. (2019) *Desarrollo de aplicaciones para IoT con el módulo ESP32* - España.[Trabajo de Grado de Ingeniería en tecnologías de telecomunicaciones, Universidad de Alcalá Escuela Politécnica Superior]
https://ebuah.uah.es/dspace/bitstream/handle/10017/35420/TFG_Benito_Herranz_2019.pdf?sequence=1&isAllowed=y
- Bento, A. (2019) *An Experiment with Arduino Uno and Tft Nextion for Internet of Things* - Brazil [Paper de investigación, Universidade Nove de Julho]
file:///C:/Users/USUARIO/Downloads/12_IEEEICRIEECE_AnExperimentwithArduinoUnoandTftNextionforInternetofThings.pdf
- Buesaquillo L. (2019) *Sistema de control para mejorar el desempeño de una máquina tostadora de café* - Ecuador.
<http://repositorio.utn.edu.ec/bitstream/123456789/9546/2/04%20MEC%20273%20TRABAJO%20GRADO.pdf>
- Carrión, M. & Romero, C (2012) *Desarrollo de software hmi scada e implementación sobre un módulo didáctico autónomo para ventas de la empresa ecuaínsetec cía. ltda.* (Tesis de pregrado). Universidad Politécnica Salesiana Sede Quito. Ecuador.
- Degollado C, Tovar C, Garcia D, Tijerina F, Chavez J, Olivares J, Buendía K, Rodríguez M, Lizarraga M, Hernandez R, Vieira R y Galvan S. (2019) *Manual SQL Server Transact SQL Básico / Avanzado* - México.
http://www.utsc.edu.mx/vidaEstudiantil/pdf/pdf_pades/manual_sql_server_2019.pdf
- Drake J. (2008) *Programación orientada a objetos: Lenguajes, Metodologías y Herramientas. Máster de computación. Proceso de desarrollo de aplicaciones software.*
https://www.ctr.unican.es/asignaturas/MC_OO/Doc/OO_08_I2_Proceso.pdf

- Dassault Systèmes SolidWorks Corporation (2002-2021). SolidWorks ingeniería. Waltham, EE UU. Recuperado de <https://www.solidworks.com/es/contact-us>.
- García, S. (2014) *Implementación de analizadores de protocolos de comunicaciones spi, i2c*. (Tesis de pregrado). Universidad Carlos III de Madrid. https://e-archivo.uc3m.es/bitstream/handle/10016/26320/TFG_Salvador_Cristina_Garcia_2014.pdf?sequence=1&isAllowed=y
- Nieto, G. & Quimi, J (2018) *Diseño e implementación de un sistema de pesaje con acceso a base de datos con tecnología DFID y reportería por medio de una red local*. (Tesis de pregrado). Universidad Politécnica Salesiana Sede Guayaquil. Ecuador, Guayaquil.
- López E. (2018) *Ingeniería en Microcontroladores Protocolo SPI (Serial Peripheral Interface) - México*. <http://www.i-micro.com/pdf/articulos/spi.pdf>
- MTM TELECOM. *Enlaces inalámbricos punto a punto y punto multipunto*. [Online].; 2012 [cited 2020 febrero 12. Available from: <http://www.mtmtelecom.com/index.php/2012-07-04-19-05-27/enlaces-inalambricos-punto-apunto-y-punto-multipunto.html>.
- MAYORGA, J. (9 de 11 de 2016). Balanza Electrónica. Obtenido de <https://prezi.com/lryhbcnd1ky/una-balanza-es-un-dispositivo-utilizado-para-determinar-la-m/>
- Pelayo J. (2019) *Módulo inalámbrico de señalización luminosa en vehículos de emergencias - España*. <https://core.ac.uk/download/pdf/250406843.pdf>
- Pipa J. (2019) *Introducción a las redes inalámbricas, ventajas y desventajas del wifi, estándares inalámbricos, hardware inalámbrico, diseño de una red inalámbrica, instalación de una red inalámbrica, configuración de red inalámbrica, software para redes inalámbricas, aplicaciones - Perú*. <https://repositorio.une.edu.pe/bitstream/handle/UNE/5004/Redes%20inal%C3%A1mbricas.pdf?sequence=1&isAllowed=y>
- Tipler, P. & Mosca, G., 2005. *Física para la ciencia y la tecnología: electricidad y magnetismo*. Volumen II. Quinta ed. España. Editorial Reverté. Págs. 901.
- Solvo Vallés (2017) *Balanza de Peso SS*. Recuperado de https://solvo.es/WebRoot/Store19/Shops/fab964d6-f17c-4c7b-be1a-4645d1f9dad2/MediaGallery/ss_solvo.pdf

Valdivieso, C. (2018). *Diseño de un sistema automático de pesaje de snacks para la empresa SOINTEC*. (Tesis de pregrado). Universidad Tecnológica Israel. Ecuador, Quito.
<http://repositorio.unsm.edu.pe/bitstream/handle/11458/2914/SISTEMAS%20-%20Gilmer%20Anthony%20Ar%C3%A9valo%20Tuanama.pdf?sequence=1&isAllowed=y>

ANEXOS

Anexo 1: Manual Técnico de la balanza SUPER SS



SUPER-SS

ELECTRONIC SINGLE/DUAL DISPLAY

Thank you for using our ACS-3S/NEW SU serial single/double side displays product of electronic waterproof weighing/counting scale. New SU are plastic scales and 3S a stainless steel scale. The scale is manufactured with advances computer-controlled hi-techniques. It has the characteristics of satisfactory in performance accurate in weighing, convenient for carrying, cheap in price. It is sturdy and durable and widely used in not only retail stores and weighing of ration packing of commodities but also in high humidity surroundings because of it's well waterproof function.

I. MAIN TECHNICAL INDEX

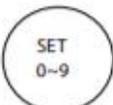
1. BASIC DATA

	3S/NEW SU-3	3S/NEW SU-6	3S/NEW SU-15	3S/NEW SU-30
Max. Capa.	3000g	6000g	15 kg	30 kg
Min. Capa.	20e			
Test Divi.	1g	2g	5g	10g
Display Divi.1	0.5g	1g	2g	5g
Display Divi.2	0.2g	0.5g	1g	2g
Max. Tare	100% Max. Capa.			
Accuracy	III			
Pan size	3S/NEW SU		190x230mm	

- 2. WORKING TEMPERATURE: -5~+35°C
- 3. STORAGE TEMPERATURE: -25~+50°C
- 4. POWER: 6V4.5Ah rechargeable battery
- 5. NET WEIGHT: 3.6kg/pc
- 6. DIMENSION: 4pcs/ctn, 580x350x300

II. KEYBOARD AND CHARACTER PROMPT

1. KEYBOARD

	Turns the scale on and zero
	Tare key
	Power Off
	Set/Fig. Key. Set parameter and input Fig.

2. CHARACTER PROMPT

dC x.xx: Mean the voltage of battery is x.xx V.

-----: Show this as the weight over 100%FS+9e, with the ding sounding, this means over load, reduce the weight.

-AdC-: As the A/D change overflowing, display shows this with the ding sounding

bATLo: The voltage is lower. You can use a short while and charge immediately.

c_End: Recharge is end.

Err-0: Overload zero of loadcell. Please re-adjust or change loadcell.

II. OPERATION GUIDE

1. Preparation

Adjust the base foot, watch the level, make the pan in horizontal.

2. Turn on or turn off

2.1 Turn on

Press the [ON/ZERO] key, the ding sound, it displays version No. and the battery voltage level and 9,8,7---1,0, the "ZERO" lamp lighting. Now the scale is in a working state.

2.2 Turn off Long- press [OFF] key.

2.3 Auto-off

After 10 minutes in zeroing, auto-off.

2.4 Low voltage auto-off

When the voltage lower than 5.6V, then power-off.

3. Zero

When without taring the weight display is $\leq 4\%$ Max. capa., press the [ON/ZERO] key, the display shows "0", the zero lamp lighting.

4. Weighing

Put weight to pan.

5. Tare weighing

Press the [TARE] key, it's being think the "skin" tare, the weight display is "0", the "ZERO" lamp off, the "TARE" lamp lighting. Pick up the weight and the "skin", the weight display is negative, the "ZERO" lamp lighting, press the [TARE] key, the "TARE" lamp off.

6. Save power function

Zero after 40 seconds, auto save power, shows "0". Put a weight start weighing

7. Battery indicate.

Battery lamp: "High" means voltage higher than 6.3V, "Mid" means voltage between 6V and 6.3V, "Low" means voltage lower than 6V.

8. Charge

Power supplied by a internal rechargeable lead acid battery (6V/4Ah). When it shows "-bAtLo-". Please turn off and charge immediately. The time of full charging is 12 hours.

9. Set parameter

Press [SET/0~9] key 3 seconds to set. Now show menu. Press [SET/0~9] continuous it displays :

rAngE(set limit)

UnItS(choose unit)

A-oFF(choose auto power off)

FILt(set display mode)

ZErO(set auto zero)

bUZZEr(set buzzle)

d(choose division)

on-rA(choose range of power on)

d-dP(choose single or dual display)

9.1 Set of upper and low limit

9.1.1 When it shows rAngE, press [TARE] and then [SET/0~9] to choose on or oFF, on is start the function,oFF is exit,press [TARE] to confir.

9.1.2 When choose on and press [TARE],now weighing with limit," under" lighting, max figure is blinking,press [SET/0~9] once to add 1,press [TARE] to confirm,when finishing auto exit and set "over".After setting into weighing.

9.1.3 When low limit > high limit, it displays off,then exit.

9.1.4 Alarming indicate

"m" is the weight

when $m \geq$ high limit, "over" lamp is lighting.

When low limit $< m <$ high limit, "accept" lamp is lighting.

When $m \leq$ low limit, "under" lamp is lighting.

Caution: When the display is not stabilization, no lamp lights.

9.2 Change unit

9.2.1 When it displays "units", press [TARE], then press [SET/0~9] to choose, then press [TARE] to confirm and exit.

Unit: H9(kg), 9(g), Lb, Lb.oz, pcs(counting). (We may not choose imperial units when calibrating).

9.2.2 counting: When it displays "pcs", press [TARE]. Now "pcs" lamp lighting.

Longtime press [ON/ZERO], it displays count and return to 0. then put the sample and press [TARE]. Press [SET/0~9] to add 1 and press [TARE] to choose. After finishing it display pcs and pcs lamp lighting.

Exit counting: choose a unit.

Caution: Pick quantities of Max less 30000

9.3 A-off(choose auto power off)

When menu displays A-off,press [TARE] to enter,press [SET/0~9] to choose n or y, n is not auto power off,y is after 10 minutes in zeroing, auto-off.

9.4 Change display mode

When it displays "FILtX" (FILt1 means one step to display, FILt2 means 3-4 steps to display, FILt3 means 6-8 steps to display, FILt4 means display is fast). Press [SET/0~9] to change, press [TARE] to confirm and exit.

9.5 Set range of auto-zero:

When it displays zero, press [TARE], now show zero x.x(x.x is the range:0.5d,1d,1.5d,2d,2.5d,3d,3.5d,4d,4.5d,5d. d is the division value when n=3000).

Press [SET/0~9] to choose and [TARE] to confirm.

9.6 bUZZEr(set buzze sound on or off)

When menu displays bUZZEr,press [TARE] to enter,press [SET/0~9] to choose on or off, press [TARE] to confirm.

9.7 Change division figure

When menu displays d,press [TARE] to enter,press [SET/0~9] to choose division.

9.8 On-rA(choose range of power on)

When menu displays On-rA,press [TARE] to enter,press [SET/0~9] to choose 100, 100 is 100% capacity. Press [TARE] to confirm.

9.9 d-dP(choose single or dual display)

When menu displays d-dP,press [TARE] to enter,press [SET/0~9] to choose n or y, n is single display is dual display, Press [TARE] to confirm.

10. Tune LED lighting

On powering on, long time press [zero/on], it shows LC-,press [SET/0~9] to choose LU-1 or LU-2,LU-1 is saving mode(commend),Press [TARE] to confirm.

IV. NOTE

1. Not overloading. Not throw heavy things on the scale violently. The damage for overloading is not within the range of guaranteed maintenance.
2. Don't use sharp thing to operation the keyboard. If the rubber is damaged it will influence the performance of waterproof.
3. Keep the scale clean in order to weighing well.
4. There is a lead seal on the bottom of the scale. Users are not allowed to remove the seal. If the seal is damaged the scale is not guaranteed for maintenance.
5.
 - a. You must charge the battery at least per 3 months when you don't use the scale for a long time.
 - b. Please correctly connect line between adapter and battery, black to black, red to red.
 - c. If after charging the used time becomes short than often, please change a new battery.

CALIBRATION OF SS SCALE (VERSION 3S-5.3OR5.4)

A. Multi-point adjust:

1. Push the adjust key to switch on, the switch is on the bottom of the scale, locked with a seal. Press "on/zero" key, after a while start to adjust.
2. Press "set/0~9" and "on/zero" key the same time, choose weighing units-kg, lb or counting function, press tare to confirm. Then the display shows "c". "c" is the Max. capacity, press "set/0~9" to choose c-600,c-1500,c-3000,c-6000,c-15H9(15kg),c-30H9(30kg) and press "tare" to confirm.
3. Now it displays 1/3 Max capacity, then display 0. (If is not 0, press "on/zero")
4. Put the displayed weights on the centre of the plat, after it shows stable, press the "tare", it shows 2/3 Max capacity. Then put the displayed weight it displays Max capacity, put the displayed weight. Press "tare".
5. Now it shows c-0, "c" is flashing. Now change gravity. Press "on/zero" to add 1 internal number, Press "set/0~9" to reduce 1 internal number. 60000 internal number means the max. capacity. Press "tare" to exit adjustment.
6. Push the adjust key to switch off.

B. One-point user ADJUST:

Push the adjust key to switch on. Press "tare" 3 seconds, display "CAL"0". Press "tare" to confirm zero. Put the Max capacity weight, after 5 seconds press "tare", it shows CAEnd and exit adjustment. Push the adjust key to switch off. NOTE: After changed loadcell or PCB, if you need adjust but it shows "Err-0", it means zero is not in the limited range. Then after power on please press "tare" immediately, it will go into adjusting.

Block Diagram

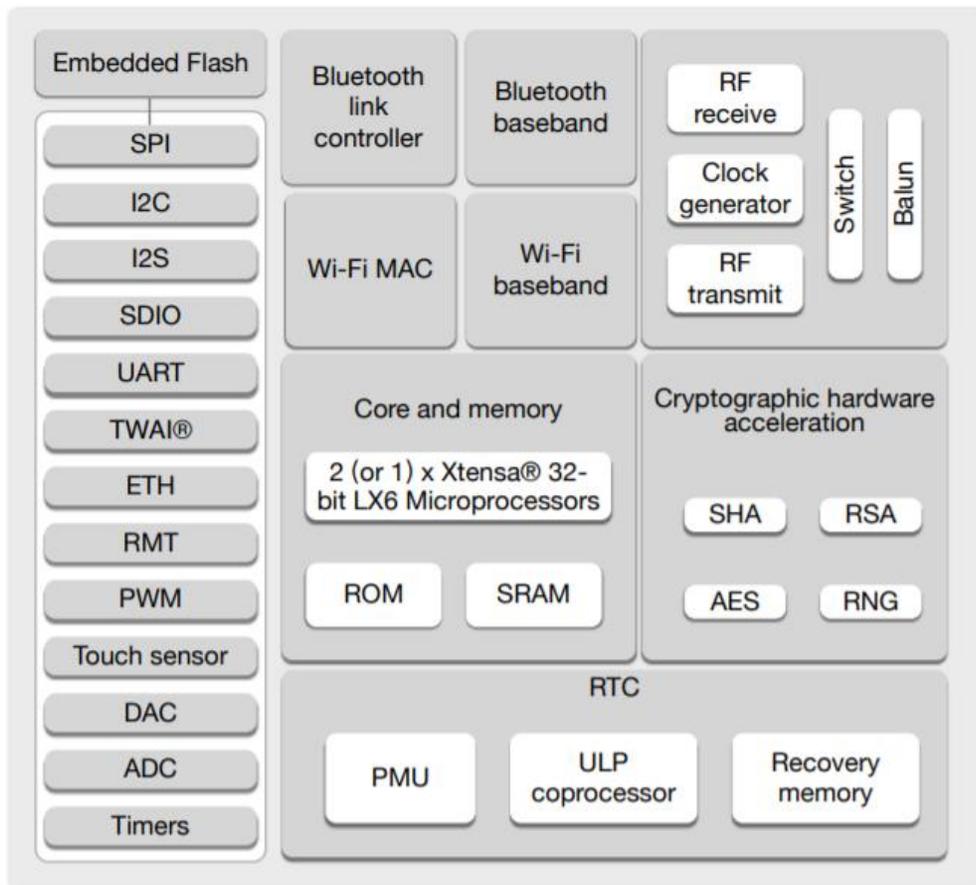


Figure 1: Functional Block Diagram

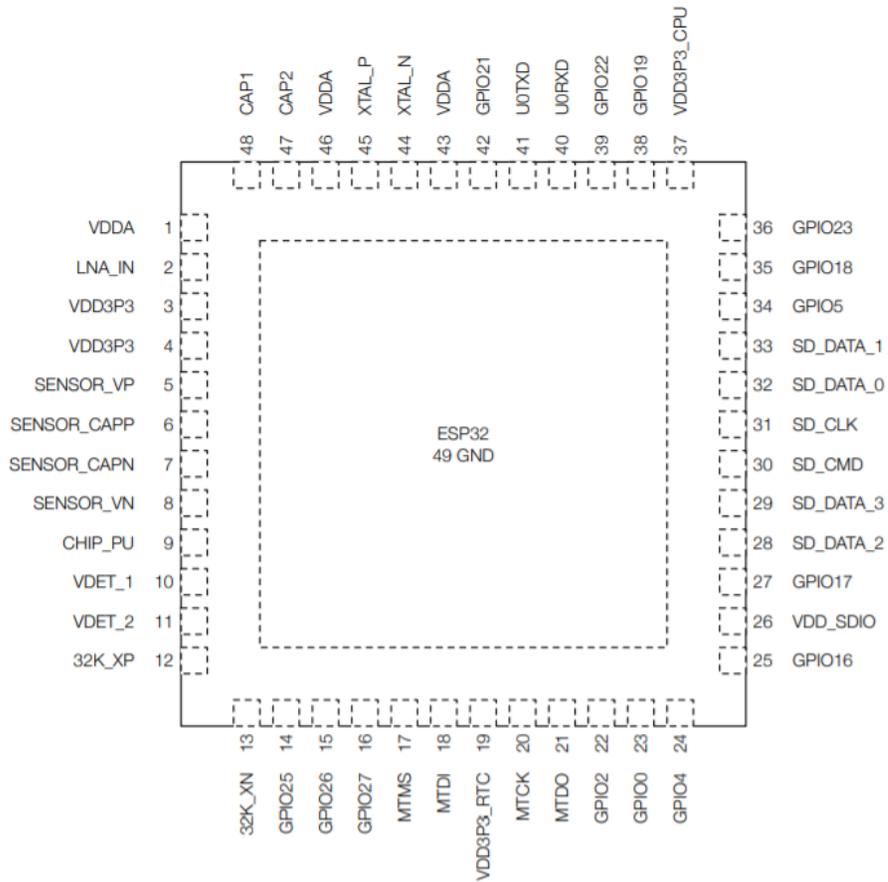


Figure 2: ESP32 Pin Layout (QFN 6*6, Top View)

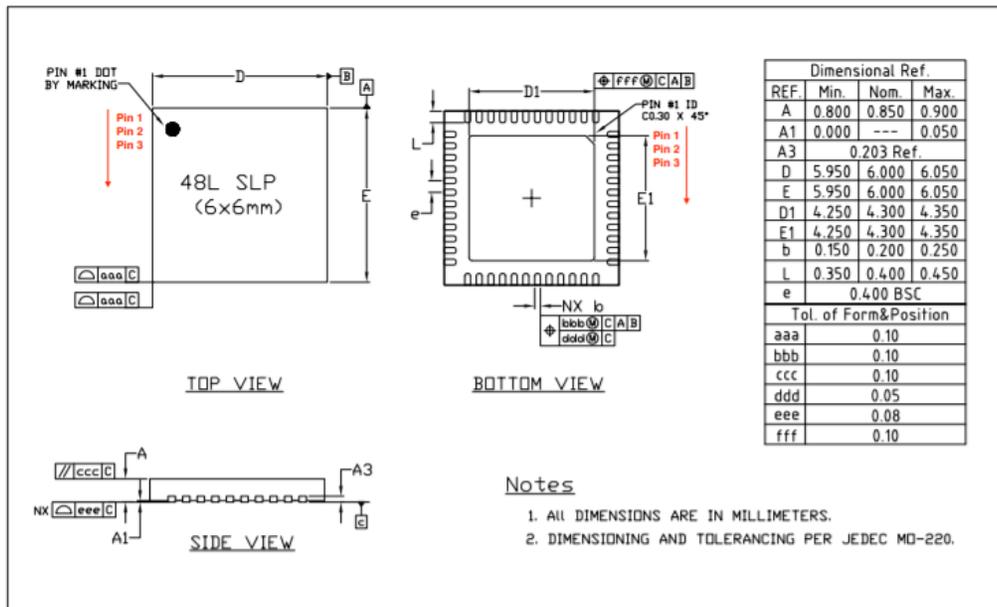


Figure 8: QFN48 (6x6 mm) Package

Anexo 3: Manual Técnico de la pantalla Nextion

NX4832T035

- Overview
- Nextion Models
- Specifications
- Electronic Characteristics
- Working Environment & Reliability Parameter
- Interfaces Performance
- Memory Features
- Product Dimensions



Overview

Nextion is a seamless Human Machine Interface (HMI) solution that provides a control and visualisation interface between a human and a process, machine, application or appliance. Nextion is mainly applied to IoT or consumer electronics field. It is the best solution to replace the traditional LCD and LED Nixie tube. With the Nextion Editor software ([Official Download](#)), users are able to create and design their own interfaces for Nextion display.

Package includes: Nextion Display, connecting wire, a power supply test board.

Go Shopping: [NX4832T035 \(IM150918001\)](#)

Note: the small power supply test board and connecting wire inside the package allow you to test if the electrical supply is enough or not. See the image below on how to use it.

Caution:



Working under insufficient power supply condition will damage the Nextion model easily.

Blurred screen? Flashing? You may be suffering from power shortages. Power off at the first possible moment. No more repeated attempts to damage your Nextion model.

A small connector is included in the package. Please try to power Nextion with your phone charger through the connector to check if Nextion works well.

A high quality usb cable is required.



Nextion Models

Nextion Type	Basic Series
Nextion Models	NX4832T035_011N (N: No touch)
	NX4832T035_011R (R: Resistive touchscreen)

Specifications

	Data	Description
Color	64K 65536 colors	16 bit 565, 5R-6G-5B
Layout size	100.5 (L)×54.94 (W)×4.25 (H)	NX4832T035_011N
	100.5 (L)×54.94 (W)×5.45 (H)	NX4832T035_011R
Active Area (A.A.)	85.50mm(L)×54.94mm(W)	
Visual Area (V.A.)	73.44mm(L)×48.96mm(W)	
Resolution	480×320 pixel	Also can be set as 320×480
Touch type	Resistive	
Touches	> 1 million	
Backlight	LED	
Backlight lifetime (Average)	>30,000 Hours	
Brightness	200nit (NX4832T035_011N)	0% to 100%, the interval of adjustment is 1%
	180 nit (NX4832T035_011R)	0% to 100%, the interval of adjustment is 1%
Weight	38.2g (NX4832T035_011N)	
	48.2g (NX4832T035_011R)	

Electronic Characteristics

	Test Conditions	Min	Typical	Max	Unit
Operating Voltage		4.75	5	7	V
Operating Current	VCC=+5V, Brightness is 100%	-	145	-	mA
	SLEEP Mode	-	15	-	mA
Power supply recommend : 5V, 500mA, DC					

Working Environment & Reliability Parameter

	Test Conditions	Min	Typical	Max	Unit
Working Temperature	5V, Humidity 60%	-20	25	70	°C
Storage Temperature		-30	25	85	°C
Working Humidity	25°C	10%	60%	90%	RH

Interfaces Performance

	Test Conditions	Min	Typical	Max	Unit
Serial Port Baudrate	Standard	2400	9600	115200	bps
Output High Voltage	IOH=-1mA	3.0	3.2		V
Output Low Voltage	IOL=1mA		0.1	0.2	V
Input High Voltage		2.0	3.3	5.0	V
Input Low Voltage		-0.7	0.0	1.3	V
Serial Port Mode	TTL				
Serial Port	4Pin_2.54mm				
USB interface	NO				
SD card socket	Yes (FAT32 format), support maximum 32G Micro SD Card * microSD card socket is exclusively used to upgrade Nextion firmware / HMI design				

Memory Features

Memory Type	Test Conditions	Min	Typical	Max	Unit
FLASH Memory	Store fonts and images			16	MB
RAM Memory	Store variables			3584	BYTE

Product Dimensions

3.5"_Nextion_Dimension

Product Certificates

CE-EMC, RoHS

LED 驱动控制/键盘扫描专用集成电路 FD628

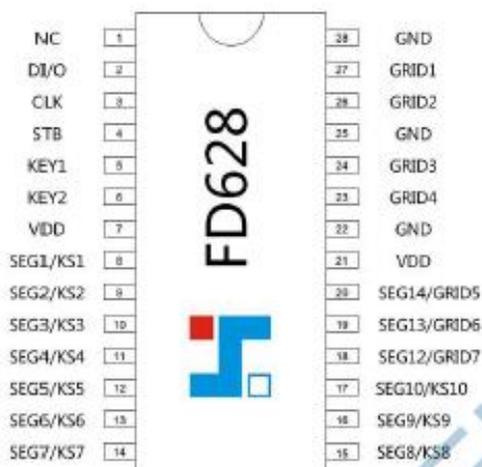
一、概述

FD628是一种带键盘扫描电路接口的LED驱动控制专用电路。内部集成有MCU输入输出控制数字接口、数据锁存器、LED驱动、键盘扫描、辉度调节等电路。本芯片性能稳定、质量可靠、抗干扰能力强，可适应于24小时长期连续工作的应用场合。

二、特性说明

- 显示模式：4位*13段或7位*10段
- 8级辉度控制
- 5V的条件下，位驱动电流可达到650mA
- 键盘扫描：10×2（支持组合键）
- 串行接口（CLK,STB,DAT）
- 内置时钟振荡电路
- 内置上电复位电路
- 工作电压：3.3V~5V
- 采用COMS工艺
- ESD HBM: >8KV（美标883模式）
- SOP28(FD628S)封装

三、管脚定义



四、管脚功能说明

符号	管脚名称	说明
NC		空脚
DI/O	数据输入/输出	串行接口的数据输入输出，内置上拉电阻。
CLK	时钟输入	串行接口的数据时钟输入，内置上拉电阻。
STB	数据传输控制	内置上拉电阻。在下降沿初始化串行接口。随后等待接收指令，STB为低后的第一份个字节作为指令。当STB为高时，CLK被忽略。
KEY1-KEY2	键扫信号输入	键扫信号在显示周期结束后被锁存
SEG1/KS1-SEG10/KS10	段输出	开漏输出，管脚也做键扫描
SEG12/GRID7-SEG14/GRID5	段/位输出	段/位驱动输出
GRID1-GRID4	位输出	开漏输出
VDD	逻辑电源	5V±10%
GND	逻辑地	接地

五、主要电气参数

极限参数 (Ta = 25℃)

参 数	符 号	范 围	单 位
逻辑电源电压	VDD	-0.5 ~ +7.0	V
逻辑输入电压	V _{I1}	-0.5 ~ VCC + 0.5	V
LED 段驱动输出电流	I _{SEG}	-50	mA
LED 位驱动输出电流	I _{GRID}	700	mA
工作温度	T _{OPT}	-40 ~ +80	℃
储存温度	T _{STG}	-65 ~ +150	℃

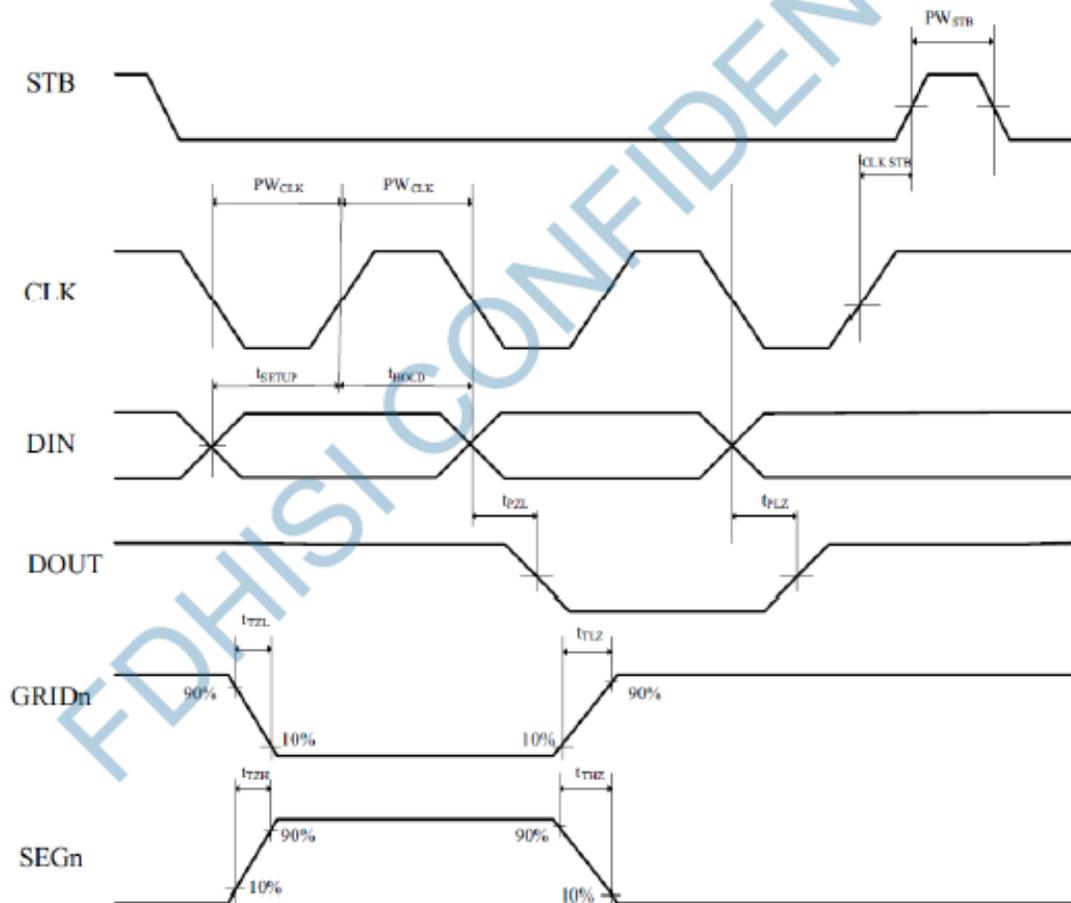
电气特性 (测试条件: Ta = 25℃, VCC = 5V)

参 数	符 号	测试条件	最 小	典 型	最 大	单 位
逻辑电源电压	VDD	-	3.3	5.0	5.5	V
高电平输入电压	V _{IH}	-	0.7VDD	-	VDD	V
低电平输入电压	V _{IL}	-	0	-	0.3VDD	mA
静态电流	I _{DD}	VDD=5.0V, 无负载, 显示关闭	-	-	1.2	mA
SEG驱动电流	I _{SEG}	VDD=5.0V, SEG接2Ω电阻对GND	-	-47	-	mA
GRID驱动电流	I _{GRID}	VDD=5.0V, GRID接2Ω电阻对GND	-	650	-	mA
KEY下拉电阻	R _{KEY}	-	10	-	25	KΩ
D/IO、CLK、 STB上拉电阻	R	-	-	15	-	KΩ
GRID震荡频率	f _{GRID}	VDD=5.0V	-	210	-	Hz

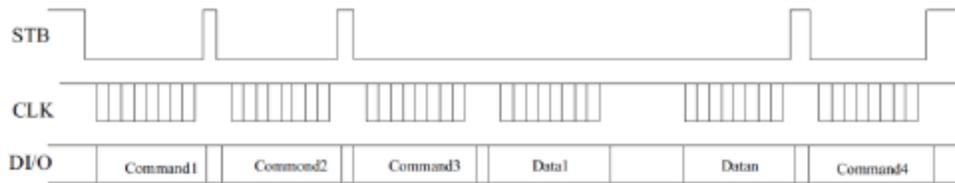
时序特性 (测试条件: $T_a = 25^\circ\text{C}$, $V_{CC} = 5\text{V}$)

参 数	符 号	测试条件	最 小	典 型	最 大	单 位
时钟脉冲宽度	PW_{CLK}	-	500	-	-	ns
选通脉冲宽度	PW_{STB}	-	1	-	-	μs
数据建立时间	t_{SETUP}	-	100	-	-	ns
数据保持时间	t_{HOLD}	-	100	-	-	ns
CLK→STB时间	t_{CLKSTB}	CLK↑→STB↑	1	-	-	μs

时序波形图



*注: DIN 和 DOUT 为芯片内部 PAD 位, 通过封装合并成 DIO。



Command1: 设置显示模式;
 Command2: 设置数据;
 Command3: 设置地址;
 Data1~n : 传输显示数据 (最多 14bytes);
 Command4: 控制显示;

注: 显示寄存器地址范围为: 00H-0DH, 总共 14 个字节。连续地址模式写时, 如果地址为 00H 时, 后面紧跟 14 个显示数据字节; 地址为 01H 时, 后面紧跟 13 个; 以此类推, 如果地址为 n, 那么紧跟后面的显示字节为 14-n 个。

读按键模式



Command1: 设置读按键模式;
 Data1~n : 按键数据;

FDHISI CONFIDENTIAL

Anexo 5: Código del Programa del microcontrolador ESP32

```
#include "driver/uart.h"
#include <string.h>
#include <sys/param.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "freertos/event_groups.h"
#include "esp_system.h"
#include "esp_wifi.h"
#include "esp_event_loop.h"
#include "esp_log.h"
#include "nvs_flash.h"

#include "lwip/err.h"
#include "lwip/sockets.h"
#include "lwip/sys.h"
#include <lwip/netdb.h>

//-----
//These are for UART
#define ECHO_TEST_TXD (GPIO_NUM_14)
#define ECHO_TEST_RXD (GPIO_NUM_12)
#define ECHO_TEST_RTS (UART_PIN_NO_CHANGE)
#define ECHO_TEST_CTS (UART_PIN_NO_CHANGE)
#define BUF_SIZE (1024) //Creo que no es necesario

//-----
//These are for tcp_server
#define EXAMPLE_WIFI_SSID "WITTOP"
#define EXAMPLE_WIFI_PASS "WITTOP04"
#define PORT 3333
/* FreeRTOS event group to signal when we are connected & ready to make a request */
static EventGroupHandle_t wifi_event_group;
const int IPV4_GOTIP_BIT = BIT0;
const int IPV6_GOTIP_BIT = BIT1;
static const char *TAG = "example";
//-----

#define DATA_GPIO 16
#define CLOCK_GPIO 4
#define STB_GPIO 17

bool busqueda=true;
bool low_state=false;
bool high_state=false;
long long int t_pasado;
long long int t_actual;
bool referencia_final=false;
int flancos_subida=0;
```

```

bool toma_datos=false;
bool low_clock=false;
int trama[100];
int i=0;
int count=0;

int digito;
int cuarta;
int tercera;
int segunda;
int primera;
int peso;
int cifra[7];
int pesos[2];
char num_c[5]; //Evaluar si num_c es necesario
int contador_filtro=0;
bool filtro_encontrado=false;
int camb=0;
int cambiopeso[2];
bool nuevo_peso=false;
bool hab_cuarta=false;
bool es_cuatro_cifras=false;

char desc[20]="MORA";
char objed[8]="200";
char tolhd[8]="5";
char tolld[8]="5";
char valu[8]="5";
char cant[8]="20";
int reseteo=0;
char *descf=desc;
char *objef=objed;
char *tolh=tolhd;
char *toll=toll;
char *valuf=valu;
char *cantf=cant;
char *descf_f=desc;

bool enviar_habilitado=false;
int obje=200;
int tolema=5;
int tolema=5;
int valunit=5;
int cantid=20;
char peso_c[5];
char trama_c[12];
char espacios[12];

```

```

int sock;
int medida=1;
char medida c[5]="1";

//These are for SuperSS
bool cuartacif(){
    int j=0;
    int r=0;
    if(trama[j]==0){
        do{
            ++j;
            if(trama[j]==1){
                ++r;
            }
        }while(r!=2);
    }else{
        do{
            ++j;
            if(trama[j]==1){
                ++r;
            }
        }while(r!=2);
    }
    for(int k=0; k<7 ; k++){
        cifra[k]=trama[j+k+1];
    }
    return true;
}

```

```

bool terceracif(){
    int j=52;
    int r=0;
    bool vacio=false;
    if(trama[j]==1){
        return false;
    }else{
        do{
            j++;
            r++;
            if(r>13){
                vacio=true;
                break;
            }
        }while(trama[j]!=1);
    }
    if(vacio==false){
        for(int k=0; k<7; k++){
            cifra[k]=trama[j+k];
        }
    }
}

```

```

    }
    return true;
} else {
    for(int k=0; k<7; k++){
        cifra[k]=0;
    }
    return true;
}
}
}
}

```

```

bool secundacif(){
    int j=68;
    int r=0;
    bool vacio=false;
    if(trama[j]==1){
        return false;
    } else {
        do {
            j++;
            r++;
            if(r>13){
                vacio=true;
                break;
            }
        } while(trama[j]!=1);
        if(vacio==false){
            for(int k=0; k<7; k++){
                cifra[k]=trama[j+k];
            }
            return true;
        } else {
            for(int k=0; k<7; k++){
                cifra[k]=0;
            }
            return true;
        }
    }
}
}
}
}

```

```

bool primeracif(){
    int j=84;
    if(trama[j]==1){
        return false;
    } else {
        do {
            j++;

```

```

| }while(trama[j]!=1);
  for(int k=0; k<7; k++){
    cifra[k]=trama[j+k];
  }
  return true;
}
}

```

```

int interpretacion(){

```

```

if(cifra[0]==0 && cifra[1]==0 && cifra[2]==1 && cifra[3]==0 && cifra[4]==1 && cifra[5]==0
&& cifra[6]==0){
  return 1;
}else if(cifra[0]==1 && cifra[1]==0 && cifra[2]==1 && cifra[3]==1 && cifra[4]==0 &&
cifra[5]==1 && cifra[6]==1){
  return 2;
}else if(cifra[0]==1 && cifra[1]==0 && cifra[2]==1 && cifra[3]==1 && cifra[4]==1 &&
cifra[5]==1 && cifra[6]==0){
  return 3;
}else if(cifra[0]==0 && cifra[1]==1 && cifra[2]==1 && cifra[3]==1 && cifra[4]==1 &&
cifra[5]==0 && cifra[6]==0){
  return 4;
}else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==0 && cifra[3]==1 && cifra[4]==1 &&
cifra[5]==1 && cifra[6]==0){
  return 5;
}else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==0 && cifra[3]==1 && cifra[4]==1 &&
cifra[5]==1 && cifra[6]==1){
  return 6;
}else if(cifra[0]==1 && cifra[1]==0 && cifra[2]==1 && cifra[3]==0 && cifra[4]==1 &&
cifra[5]==0 && cifra[6]==0){
  return 7;
}else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==1 && cifra[3]==1 && cifra[4]==1 &&
cifra[5]==1 && cifra[6]==1){
  return 8;
}else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==1 && cifra[3]==1 && cifra[4]==1 &&
cifra[5]==1 && cifra[6]==0){
  return 9;
}else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==1 && cifra[3]==0 && cifra[4]==1 &&
cifra[5]==1 && cifra[6]==1){
  return 0;
}else{
  return -1;
}
}

```

```

int interpretacionb(){

```

```

if(cifra[0]==1 && cifra[1]==0 && cifra[2]==1 && cifra[3]==0 && cifra[4]==0 &&
cifra[5]==0 && (cifra[6]==0 || cifra[6]==1)){
    return 1;
}else if(cifra[0]==1 && cifra[1]==0 && cifra[2]==1 && cifra[3]==1 && cifra[4]==0 &&
cifra[5]==1 && cifra[6]==1){
    return 2;
}else if(cifra[0]==1 && cifra[1]==0 && cifra[2]==1 && cifra[3]==1 && cifra[4]==1 &&
cifra[5]==1 && cifra[6]==0){
    return 3;
}else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==1 && cifra[3]==1 && cifra[4]==0 &&
cifra[5]==0 && cifra[6]==0){
    return 4;
}else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==0 && cifra[3]==1 && cifra[4]==1 &&
cifra[5]==1 && cifra[6]==0){
    return 5;
}else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==0 && cifra[3]==1 && cifra[4]==1 &&
cifra[5]==1 && cifra[6]==1){
    return 6;
}else if(cifra[0]==1 && cifra[1]==0 && cifra[2]==1 && cifra[3]==0 && cifra[4]==1 &&
cifra[5]==0 && cifra[6]==0){
    return 7;
}else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==1 && cifra[3]==1 && cifra[4]==1 &&
cifra[5]==1 && cifra[6]==1){
    return 8;
}else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==1 && cifra[3]==1 && cifra[4]==1 &&
cifra[5]==1 && cifra[6]==0){
    return 9;
}else if(cifra[0]==1 && cifra[1]==1 && cifra[2]==1 && cifra[3]==0 && cifra[4]==1 &&
cifra[5]==1 && cifra[6]==1){
    return 0;
}else{
    return -1;
}
}

```

```

//These are for tcp_server
static esp_err_t event_handler(void *ctx, system_event_t *event)
{
    switch (event->event_id) {
        case SYSTEM_EVENT_STA_START:
            esp_wifi_connect();
            ESP_LOGI(TAG, "SYSTEM_EVENT_STA_START");
            break;
        case SYSTEM_EVENT_STA_CONNECTED:
            /* enable ipv6 */

```

```

    tcpip_adapter_create_ip6_linklocal(TCPIP_ADAPTER_IF_STA);
    break;
case SYSTEM_EVENT_STA_GOT_IP:
    xEventGroupSetBits(wifi_event_group, IPV4_GOTIP_BIT);
    ESP_LOGI(TAG, "SYSTEM_EVENT_STA_GOT_IP");
    char *ip4 = ip4addr_ntoa(&event->event_info.got_ip.ip_info.ip);
    printf("\nMi ip: %s\n", ip4);
    uart_write_bytes(UART_NUM_1, "t11.txt=\'", 9);
    uart_write_bytes(UART_NUM_1, ip4, strlen(ip4));
    uart_write_bytes(UART_NUM_1, "\"\xff\xff\xff", 4);

    break;
case SYSTEM_EVENT_STA_DISCONNECTED:
    /* This is a workaround as ESP32 WiFi libs don't currently auto-reassociate. */
    esp_wifi_connect();
    xEventGroupClearBits(wifi_event_group, IPV4_GOTIP_BIT);
    xEventGroupClearBits(wifi_event_group, IPV6_GOTIP_BIT);
    break;
case SYSTEM_EVENT_AP_STA_GOT_IP6:
    xEventGroupSetBits(wifi_event_group, IPV6_GOTIP_BIT);
    ESP_LOGI(TAG, "SYSTEM_EVENT_STA_GOT_IP6");

    char *ip6 = ip6addr_ntoa(&event->event_info.got_ip.ip6_info.ip);
    ESP_LOGI(TAG, "IPv6: %s", ip6);
default:
    break;
}
return ESP_OK;
}

static void initialise_wifi(void)
{
    tcpip_adapter_init();
    wifi_event_group = xEventGroupCreate();
    ESP_ERROR_CHECK( esp_event_loop_init(event_handler, NULL) );
    wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT();
    ESP_ERROR_CHECK( esp_wifi_init(&cfg) );
    ESP_ERROR_CHECK( esp_wifi_set_storage(WIFI_STORAGE_RAM) );
    wifi_config_t wifi_config = {
        .sta = {
            .ssid = EXAMPLE_WIFI_SSID,
            .password = EXAMPLE_WIFI_PASS,
        },
    };
    ESP_LOGI(TAG, "Setting WiFi configuration SSID %s...", wifi_config.sta.ssid);
    ESP_ERROR_CHECK( esp_wifi_set_mode(WIFI_MODE_STA) );
    ESP_ERROR_CHECK( esp_wifi_set_config(ESP_IF_WIFI_STA, &wifi_config) );
}

```

```

    ESP_ERROR_CHECK( esp_wifi_start() );
}

static void wait_for_ip()
{
    uint32_t bits = IPV4_GOTIP_BIT | IPV6_GOTIP_BIT ;

    ESP_LOGI(TAG, "Waiting for AP connection...");
    xEventGroupWaitBits(wifi_event_group, bits, false, true, portMAX_DELAY);
    ESP_LOGI(TAG, "Connected to AP");
}

static void tcp_server_task(void *pvParameters)
{
    while(1){

        char rx_buffer[128];
        //char tx_prueba[3];
        //int peso = 1234;
        char addr_str[128];
        int addr_family;
        int ip_protocol;
        int listen_sock=0;

        while (1) {

#ifdef CONFIG_EXAMPLE_IPV4
            struct sockaddr_in destAddr;
            destAddr.sin_addr.s_addr = htonl(INADDR_ANY);
            destAddr.sin_family = AF_INET;
            destAddr.sin_port = htons(PORT);
            addr_family = AF_INET;
            ip_protocol = IPPROTO_IP;
            inet_ntoa_r(destAddr.sin_addr, addr_str, sizeof(addr_str) - 1);

            int listen_sock = socket(addr_family, SOCK_STREAM, ip_protocol);
            if (listen_sock < 0) {
                ESP_LOGE(TAG, "Unable to create socket: errno %d", errno);
                break;
            }
            ESP_LOGI(TAG, "Socket created");

            int err = bind(listen_sock, (struct sockaddr *)&destAddr, sizeof(destAddr));
            if (err != 0) {
                ESP_LOGE(TAG, "Socket unable to bind: errno %d", errno);
                break;
            }
}
}
}

```

```

ESP_LOGI(TAG, "Socket binded");

err = listen(listen_sock, 1);
if (err != 0) {
    ESP_LOGE(TAG, "Error occurred during listen: errno %d", errno);
    break;
}
ESP_LOGI(TAG, "Socket listening");
    reseteo=0;
struct sockaddr_in6 sourceAddr; // Large enough for both IPv4 or IPv6
uint addrLen = sizeof(sourceAddr);
int sock = accept(listen_sock, (struct sockaddr *)&sourceAddr, &addrLen);
if (sock < 0) {
    ESP_LOGE(TAG, "Unable to accept connection: errno %d", errno);
    break;
}
ESP_LOGI(TAG, "Socket accepted");

int flage = fcntl(sock,F_GETFL);
fcntl(sock, F_SETFL, flage|O_NONBLOCK);

while (1) {
    int len = recv(sock, rx_buffer, sizeof(rx_buffer) - 1, 0);
    if (len < 0) {

        vTaskDelay(50 / portTICK_PERIOD_MS);
        if(nuevo_peso && pesos[1]!=0){
            nuevo_peso=false;

            trama_c[0]='B';
            trama_c[1]='R';
            trama_c[2]='U';
            trama_c[3]='T';
            trama_c[4]='O';
            trama_c[5]='.';
            trama_c[6]=' ';
            trama_c[7]=0;
            int err = send(sock, trama_c, 8, 0);
            if (err < 0) {
                ESP_LOGE(TAG, "Error occurred during sending PESO: errno %d", errno);
                break;
            }
            vTaskDelay(50 / portTICK_PERIOD_MS);

            itoa(pesos[1], peso_c,10);
            for(int j=0; j<(11-strlen(peso_c));j++){
                espacios[j]=' ';
            }

```

```

    }
    //espacios[11-strlen(peso_c)]=0;

    err = send(sock, espacios, 11-strlen(peso_c), 0);
    if (err < 0) {
    ESP_LOGE(TAG, "Error ocurred during sending peso_espacios: errno %d",
    errno);
    break;
    }
    vTaskDelay(50 / portTICK_PERIOD_MS);
    err = send(sock, peso_c, strlen(peso_c), 0);

    if (err < 0) {
    ESP_LOGE(TAG, "Error ocurred during sending valor_peso: errno %d", errno);
    break;
    }
    vTaskDelay(50 / portTICK_PERIOD_MS);

    err = send(sock, " g", 2, 0);
    if (err < 0) {
    ESP_LOGE(TAG, "Error ocurred during sending gramos: errno %d", errno);
    break;
    }
    vTaskDelay(50 / portTICK_PERIOD_MS);

    trama_c[0]='\n';
    trama_c[1]='\r';
    trama_c[2]='D';
    trama_c[3]='E';
    trama_c[4]='S';
    trama_c[5]='C';
    trama_c[6]=' ';
    trama_c[7]=':';
    trama_c[8]=' ';
    trama_c[9]=0;
    err = send(sock, trama_c, 10, 0);
    if (err < 0) {
    ESP_LOGE(TAG, "Error ocurred during sending DESC: errno %d", errno);
    break;
    }
    vTaskDelay(50 / portTICK_PERIOD_MS);

    for(int k=0; k<(13-strlen(desc));k++){
        espacios[k]=' ';
    }

```

```

err = send(sock, espacios, 13-strlen(descf), 0);
if (err < 0) {
ESP_LOGE(TAG, "Error ocured during sending desc_espacios: erro %d",
errmo);
break;
}
vTaskDelay(50 / portTICK_PERIOD_MS);

err = send(sock, descf, strlen(descf), 0);
if (err < 0) {
ESP_LOGE(TAG, "Error ocured during sending desc_string: erro %d", erro);
break;
}
vTaskDelay(50 / portTICK_PERIOD_MS);

trama_c[0]='\n';
trama_c[1]='\r';
trama_c[2]='O';
trama_c[3]='B';
trama_c[4]='J';
trama_c[5]='E';
trama_c[6]=' ';
trama_c[7]='.';
trama_c[8]=' ';
trama_c[9]=0;
err = send(sock, trama_c, 10, 0);
if (err < 0) {
ESP_LOGE(TAG, "Error ocured during sending OBJE: erro %d", erro);
break;
}
vTaskDelay(50 / portTICK_PERIOD_MS);

char obje_c[10];
itoa(obje, obje_c, 10);

for(int k=0; k<(11-strlen(objef));k++){
    espacios[k]=' ';
}

err = send(sock, espacios, 11-strlen(objef), 0);
if (err < 0) {
ESP_LOGE(TAG, "Error ocured during sending espacios_OBJE: erro %d",
errmo);
break;
}

```

```

vTaskDelay(50 / portTICK_PERIOD_MS);

err = send(sock, objef, strlen(objef), 0);
if (err < 0) {
ESP_LOGE(TAG, "Error ocured during sending valor_obje: errno %d", en
break;
}
vTaskDelay(50 / portTICK_PERIOD_MS);

err = send(sock, " g", 2, 0);
if (err < 0) {
ESP_LOGE(TAG, "Error ocured during sending gramos_obje: errno %d",
ermo);
break;
}
vTaskDelay(50 / portTICK_PERIOD_MS);

trama_c[0]='\n';
trama_c[1]='r';
trama_c[2]='T';
trama_c[3]='O';
trama_c[4]='L';
trama_c[5]='+';
trama_c[6]=' ';
trama_c[7]='.';
trama_c[8]=' ';
trama_c[9]=0;
err = send(sock, trama_c, 10, 0);
if (err < 0) {
ESP_LOGE(TAG, "Error ocured during sending TOL+: errno %d", errno);
break;
}
vTaskDelay(50 / portTICK_PERIOD_MS);

char tolemas_c[10];
itoa(tolemas, tolemas_c, 10);

for(int k=0; k<(11-strlen(tolh));k++){
    espacios[k]=' ';
}

err = send(sock, espacios, 11-strlen(tolh), 0);
if (err < 0) {
ESP_LOGE(TAG, "Error ocured during sending espacios_TOL+: errno %d",
ermo);

```

```

break;
}
vTaskDelay(50 / portTICK_PERIOD_MS);

err = send(sock, tolh, strlen(tolh), 0);
if (err < 0) {
ESP_LOGE(TAG, "Error ocurred during sending valor_tolmas: errno %d",
errno);
break;
}
vTaskDelay(50 / portTICK_PERIOD_MS);

err = send(sock, " g", 2, 0);
if (err < 0) {
ESP_LOGE(TAG, "Error ocurred during sending gramos_tolmas: errno %d",
errno);
break;
}
vTaskDelay(50 / portTICK_PERIOD_MS);

trama_c[0]='\n';
trama_c[1]='\r';
trama_c[2]='T';
trama_c[3]='O';
trama_c[4]='L';
trama_c[5]='.';
trama_c[6]='.';
trama_c[7]='.';
trama_c[8]='.';
trama_c[9]=0;
err = send(sock, trama_c, 10, 0);
if (err < 0) {
ESP_LOGE(TAG, "Error ocurred during sending TOL-: errno %d", errno);
break;
}
vTaskDelay(50 / portTICK_PERIOD_MS);

char tolemenos_c[10];
itoa(tolemenos, tolemenos_c, 10);

for(int k=0; k<(11-strlen(toll));k++){
    espacios[k]=' ';
}

err = send(sock, espacios, 11-strlen(toll), 0);

```

```

        if (err < 0) {
            ESP_LOGE(TAG, "Error occurred during sending espacios_TOL-: erro %d",
ermo);
            break;
        }
        vTaskDelay(50 / portTICK_PERIOD_MS);

        err = send(sock, toll, strlen(toll), 0);
        if (err < 0) {
            ESP_LOGE(TAG, "Error occurred during sending valor_tolemenos: erro %d",
ermo);
            break;
        }
        vTaskDelay(50 / portTICK_PERIOD_MS);

        err = send(sock, " g", 2, 0);
        if (err < 0) {
            ESP_LOGE(TAG, "Error occurred during sending gramos_tolemenos: erro %d",
ermo);
            break;
        }
        vTaskDelay(50 / portTICK_PERIOD_MS);

        trama_c[0]=' ';
                                trama_c[1]='\n';

        trama_c[2]='\r';
        trama_c[3]='\0';
        trama_c[4]='\0';
        trama_c[5]='\0';
        trama_c[6]='\0';
        trama_c[7]='\0'; //antes '\n'
        trama_c[8]='\0'; //antes '\r'
        trama_c[9]=0;
        err = send(sock, trama_c, 10, 0);
        if (err < 0) {
            ESP_LOGE(TAG, "Error occurred during sending espacios_y_saltos: erro %d",
ermo);
            break;
        }
        vTaskDelay(50 / portTICK_PERIOD_MS);
    }
    //break;
}
// Connection closed, si cierro mi cliente entonces se cierra la conexion y sale del while
else if (len == 0) {

```

```

        ESP_LOGI(TAG, "Connection closed");
        //enviar_habilitado=false;
        break;
    }
    // Data received-----
-----
    else {
        // Get the sender's ip address as string
        if (sourceAddr.sin6_family == PF_INET) {
            inet_ntoa_r(((struct sockaddr_in *)&sourceAddr)->sin_addr.s_addr, addr_str,
sizeof(addr_str) - 1);
        } else if (sourceAddr.sin6_family == PF_INET6) {
            inet6_ntoa_r(sourceAddr.sin6_addr, addr_str, sizeof(addr_str) - 1);
        }

            rx_buffer[len] = 0; // Null-terminate whatever we received and
treat like a string
        ESP_LOGI(TAG, "Received %d bytes from %s:", len, addr_str);
        ESP_LOGI(TAG, "%s", rx_buffer);

        const char ch=' ';

        if(!strncmp(rx_buffer,"W PROG",6)){

            //send(sock, "Modo Programacion\n\r", 19, 0);
            memset(rx_buffer, 0, sizeof(rx_buffer));

            while(1){

                len = recv(sock, rx_buffer, sizeof(rx_buffer) - 1, 0);
                if (len == 0) {
                    ESP_LOGI(TAG, "Connection closed");
                    break;
                }else if(len>3){ //Este numero 3 es para evitar el rebote del \n\r y algun espacio
vacio porseaca

                    rx_buffer[len] = 0;

                    if(!strncmp(rx_buffer,"W DESC ",7)){
                        if(len<19){

                            strcpy(descf, &rx_buffer[7]);

                            printf("\nDESCRIPCION, retorna: |%s|\n", descf);
                            fflush(stdout);
                            printf("\nElemento1: %c , Elemento2: %c\n",
*(descf+strlen(descf)-1), *(descf+strlen(descf)-2));

```

```

return\n");
if(*(descf+strlen(descf)-1)=='\n'){
    printf("Se ha detectado el salto de linea y carriage
//--*(descf_f+strlen(descf_f)-1)=='\0';
*(descf+strlen(descf)-2)=='\0';
}
printf("len = %d\n", len);
descf, strlen(descf));
printf("Manda por i2c: |%s| y el numero de bytes es: %d\n",

uart_write_bytes(UART_NUM_1, "t6.txt=\\"", 8);
uart_write_bytes(UART_NUM_1, descf, strlen(descf));
uart_write_bytes(UART_NUM_1, "\"\xff\xff\xff", 4);

}else{
    send(sock, "Codigo Incorrecto\n\r", 19, 0);
    send(sock, "Ha ingresado muchas letras\n\r", 28, 0);
}
}else if(!strncmp(rx_buffer,"W OBJE ",7)){
    if(len<13){

//objef=strchr(rx_buffer, ch)+1;
strcpy(objef, strchr(rx_buffer, ch)+1);
if(atoi(objef)!=0){

printf("\nOBJETIVO, retorna: |%s|\n", objef);
fflush(stdout);

if(*(objef+strlen(objef)-1)=='\n'){
    printf("Se ha detectado el salto de linea y carriage
return\n");
*(objef+strlen(objef)-2)=='\0';
}
printf("len = %d\n", len);
obje=atoi(objef);
printf("El valor decimal de objetivo es: %d\n", obje);
uart_write_bytes(UART_NUM_1, "t7.txt=\\"", 8);
uart_write_bytes(UART_NUM_1, objef, strlen(objef));
uart_write_bytes(UART_NUM_1, " g.\\"
}
}else{
    send(sock, "Codigo Incorrecto\n\r", 19, 0);
    send(sock, "Ingrese solo numeros\n\r", 22, 0);
}
}else{
    send(sock, "Codigo Incorrecto\n\r", 19, 0);

```



```

printf("len = %d\n", len);
tolemenos=atoi(toll);
printf("El valor decimal de tolerancia- es: %d\n",
tolemenos);

uart_write_bytes(UART_NUM_1, "t9.txt=", 8);
uart_write_bytes(UART_NUM_1, toll, strlen(toll));
uart_write_bytes(UART_NUM_1, " g.\\"x\xff\xff\xff", 7);

}else{
send(sock, "Codigo Incorrecto\n\r", 19, 0);
send(sock, "Ingrese solo numeros\n\r", 22, 0);
}
}else{
send(sock, "Codigo Incorrecto\n\r", 19, 0);
send(sock, "Ha ingresado muchos terminos\n\r", 30,
0);
}
}else if(!strcmp(rx_buffer,"W VALU ",7)){
strcpy(valuf, strchr(rx_buffer, ch)+1);
if(atoi(valuf)!=0){

if(*(valuf+strlen(valuf)-1)=='\n'){
printf("Se ha detectado el salto de linea y carriage
return\n");
*(valuf+strlen(valuf)-2)=='\0';
}
valunit=atoi(valuf);
uart_write_bytes(UART_NUM_1, "t12.txt=", 9);
uart_write_bytes(UART_NUM_1, valuf, strlen(valuf));
uart_write_bytes(UART_NUM_1, " g.\\"x\xff\xff\xff", 7);

//send(sock, "Valor Unit. Actualizado\n\r", 25, 0);
}else{
send(sock, "Codigo Incorrecto\n\r", 19, 0);
send(sock, "Ingrese solo numeros\n\r", 22, 0);
}
}else if(!strcmp(rx_buffer,"W CANT ",7)){
if(len<13){

strcpy(cantf, strchr(rx_buffer, ch)+1);
if(atoi(cantf)!=0){

if(*(cantf+strlen(cantf)-1)=='\n'){
printf("Se ha detectado el salto de linea y carriage
return\n");
*(cantf+strlen(cantf)-2)=='\0';
}
}
}
}

```



```

    if(reseteo>=10){
        esp_restart();
    }
    //vTaskDelete(NULL);
}

static void imprimir(void *pvParameters){
    while(1){

        //printf("\n*****\n");
        //vTaskDelay(500 / portTICK_PERIOD_MS);
        //-----
        if(busqueda){
            if(gpio_get_level(STB_GPIO)==0){
                low_state=true;
            }

            if(low_state==true){ //Esto es un flanco de subida, deja de ejecutarse cuando lo
detecta
                if(gpio_get_level(STB_GPIO)==1){ //Graba un millis t pasado
                    low_state=false;
                    t_pasado=esp_timer_get_time();
                    high_state=true;
                }
            }

            if(high_state==true){ //Estado alto, esto es flanco de bajada, se ejecuta solo
cuando lo detecta
                if(gpio_get_level(STB_GPIO)==0){ //Graba millis t_actual
                    high_state=false;
                    t_actual=esp_timer_get_time();
                    low_state=true;
                }
            }

            if((t_actual-t_pasado)>=2000){ //Si el tiempo encontrado es mayor a 2ms =
2000us entonces entra en la zona
                busqueda=false; //en donde el CLK y DATA estan activos cuya
duracion es de 1ms
                referencia_final=true;
                t_actual=0;
                t_pasado=0;
            }
        }
    }
}

```

```

if(referencia_final){
if(gpio_get_level(STB_GPIO)==0){
    low_state=true;
}
}

del STB
if(low_state==true){ //Inicialmente en LOW, aqui cuenta los flancos de subida

    if(gpio_get_level(STB_GPIO)==1){
        low_state=false;
        ++flancos subida;
    }
}

datos
if(flancos_subida==3){ //Si llega a 3 entonces comienza la toma de

    toma_datos=true;
}

if(flancos_subida==4){ //Si llega a 4 acaba la toma de datos
    toma_datos=false;
    i=0;
    flancos_subida=0;
    //-----
}

if(cuartacif()){
    //es_cuatro_cifras=true;
    digito=interpretacion();
    if(digito>=0){
        cuarta=digito;
        //hab cuarta=true;
        //Serial.print(cuarta);
    }else{//Esto es cuando no encuentra nada o fallido (-1)
        //cuarta=0;
        //hab_cuarta=false;
        //Serial.print("N");
        ;
    }
}

//else{

    //cuarta=0;
    //hab_cuarta=false;
    // Serial.print("C");
//}

if(terceracif()){
    digito=interpretacionb();
    if(digito>=0){

```

```

        tercera=digito;
        //Serial.print(tercera);
    }else{
        tercera=0;
        //Serial.print("N");
    }
}else{
    tercera=0;
    //Serial.print("T");
}

if(segundacif()){
    //Serial.print(" - ");
    //for(int f=0; f<7; f++){
        //Serial.print(cifra[f]);
    //}
    //Serial.print(" - ");
    digito=interpretacionb();
    if(digito>=0){
        segunda=digito;
        //Serial.print(segunda);
    }else{
        segunda=0;
        //Serial.print("N");
    }
}else{
    segunda=0;
    //Serial.print("S");
}

if(primeracif()){
    digito=interpretacionb();
    if(digito>=0){
        primera=digito;
        //Serial.println(primer);
    }else{
        primera=0;
        //Serial.println("N");
    }
}else{
    primera=0;
    //Serial.println("P");
}

if(contador_filtro==0){
    pesos[0]=1000*cuarta+100*tercera+10*segunda+primera;
    contador_filtro++;
}

```



```

        }else{
            uart_write_bytes(UART_NUM_1,
"t5.txt=\"Exacto\"\\xff\\xff\\xff", 18);
        }

        }else{
            filtro_encontrado=false;
        }
        pesos[0]=pesos[1];
    }

    fflush(stdout);

    if(filtro_encontrado){
        if(camb<1){
            cambiopeso[camb]=pesos[1];
            ++camb;
        }else{
            cambiopeso[camb]=pesos[1];
            if(cambiopeso[0]==0){
                if(cambiopeso[1]!=0){
                    nuevo_peso=true;
                }
            }

            if(cambiopeso[0]!=cambiopeso[1]){
                if( pesos[1]<(obje-tolemenos) ){
                    uart_write_bytes(UART_NUM_1,
//vTaskDelay(50 / portTICK_PERIOD_MS);
                    uart_write_bytes(UART_NUM_1,
"p0.pic=3\\xff\\xff\\xff", 11);

//vTaskDelay(50 / portTICK_PERIOD_MS);
                    uart_write_bytes(UART_NUM_1,
"n0.bco=65504\\xff\\xff\\xff", 15);

//vTaskDelay(50 / portTICK_PERIOD_MS);
                    uart_write_bytes(UART_NUM_1,
"t10.bco=65504\\xff\\xff\\xff", 16);

//vTaskDelay(50 / portTICK_PERIOD_MS);
                    //uart_write_bytes(UART_NUM_1,
"n0.bco=65504\\xff\\xff\\xff", 15);

//vTaskDelay(50 / portTICK_PERIOD_MS);
                    }else if( pesos[1]>(obje+tolemas) ){
                        uart_write_bytes(UART_NUM_1,
"p0.pic=4\\xff\\xff\\xff", 11);

//vTaskDelay(50 /
portTICK_PERIOD_MS);
                        uart_write_bytes(UART_NUM_1,
"n0.bco=63488\\xff\\xff\\xff", 15);

```



```

        if(gpio_get_level(CLOCK_GPIO)==0){ //Detecta un LOW del CLK
            low_clock=true;
        }

        if(low_clock==true){
            if(gpio_get_level(CLOCK_GPIO)==1){ //Entra al flanco de subida del
CLK
                if(gpio_get_level(DATA_GPIO)==1){
                    trama[i]=1;
                }else if(gpio_get_level(DATA_GPIO)==0){
                    trama[i]=0;
                }
                ++i;
                low_clock=false;
            }
        }
        //-----
    }
}

void app_main()
{
    uart_config_t uart_config = {
        .baud_rate = 9600,
        .data_bits = UART_DATA_8_BITS,
        .parity = UART_PARITY_DISABLE,
        .stop_bits = UART_STOP_BITS_1,
        .flow_ctrl = UART_HW_FLOWCTRL_DISABLE
    };
    uart_param_config(UART_NUM_1, &uart_config);
    uart_set_pin(UART_NUM_1, ECHO_TEST_TXD, ECHO_TEST_RXD, ECHO_TEST_RTS,
ECHO_TEST_CTS);
    uart_driver_install(UART_NUM_1, BUF_SIZE * 2, 0, 0, NULL, 0);

    uart_write_bytes(UART_NUM_1, "t6.txt=", 8);
    uart_write_bytes(UART_NUM_1, descf, strlen(descf));
    uart_write_bytes(UART_NUM_1, "\n\xff\xff\xff", 4);

    uart_write_bytes(UART_NUM_1, "t7.txt=", 8);
    uart_write_bytes(UART_NUM_1, objef, strlen(objef));
    uart_write_bytes(UART_NUM_1, " g.\n\xff\xff\xff", 7);

    uart_write_bytes(UART_NUM_1, "t8.txt=", 8);
    uart_write_bytes(UART_NUM_1, tolh, strlen(tolh));
}

```

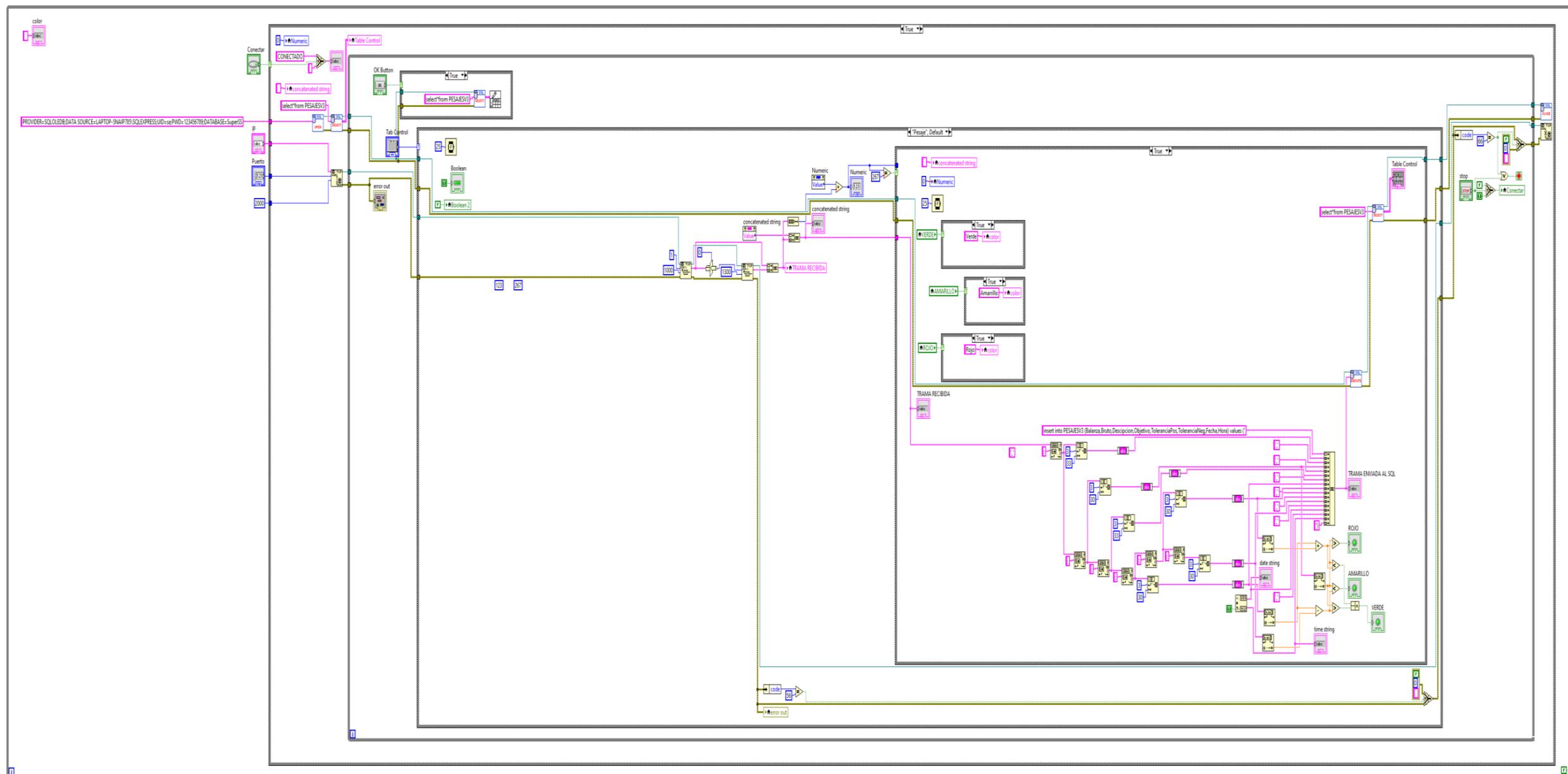
```
    uart_write_bytes(UART_NUM_1, " g.\\"\\xff\\xff\\xff", 7);

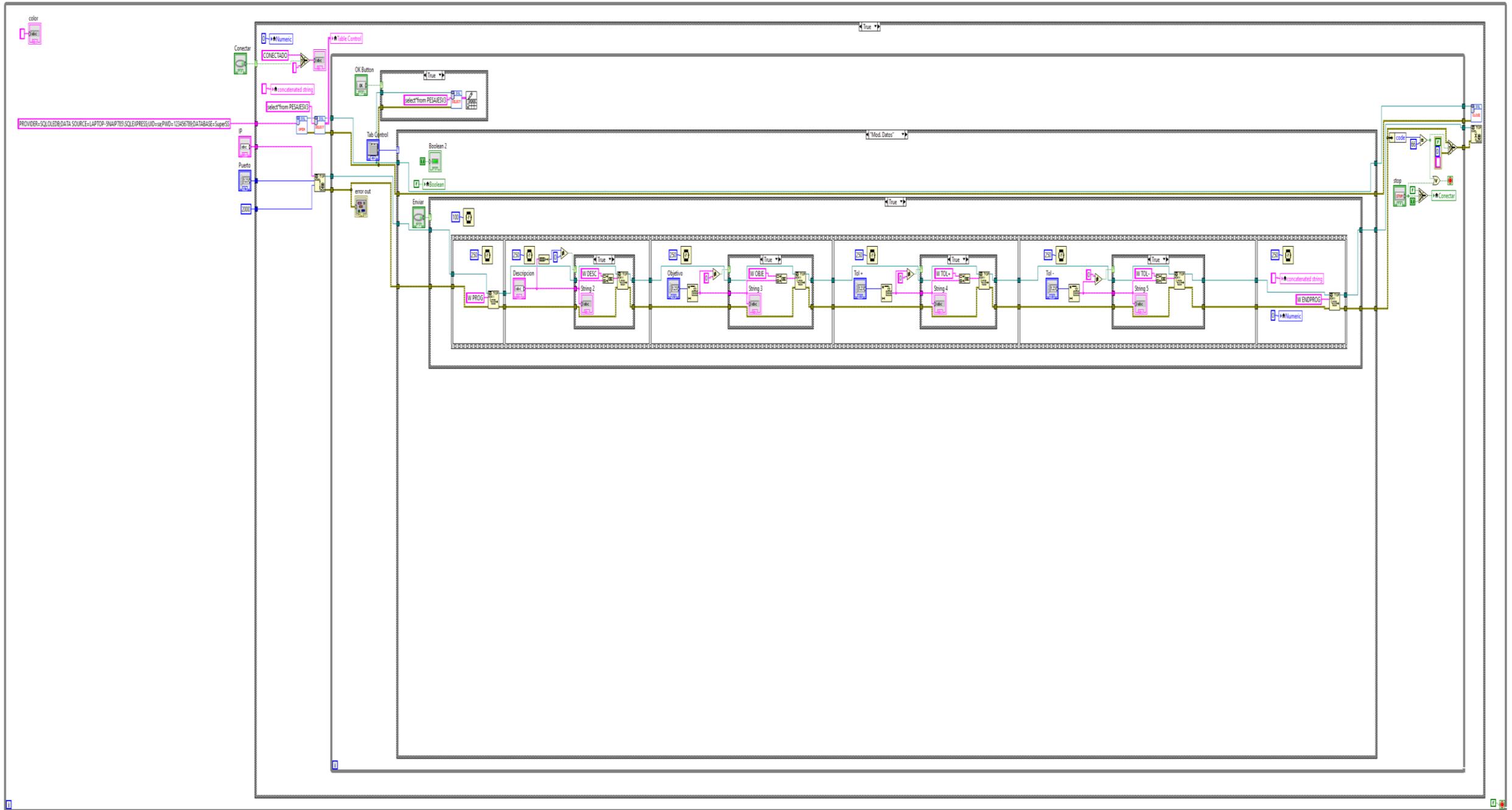
    uart_write_bytes(UART_NUM_1, "t9.txt=\\", 8);
    uart_write_bytes(UART_NUM_1, toll, strlen(toll));
    uart_write_bytes(UART_NUM_1, " g.\\"\\xff\\xff\\xff", 7);

    ESP_ERROR_CHECK( nvs_flash_init() );
    initialise_wifi();
    wait_for_ip();

    xTaskCreate(tcp_server_task, "tcp_server", 4096, NULL, 5, NULL);
    xTaskCreate(imprimir, "Imprimiendo", 4096, NULL, 6, NULL);
}
```

Anexo 6: Código del software de gestión elaborado en labview





Anexo 7: Código de base de datos SQL SERVER

```
USE [master]
GO
/***** Object: Database [SuperSS] Script Date: 08/15/2021 04:30:33 *****/
CREATE DATABASE [SuperSS] ON PRIMARY
(NAME = N'SuperSS', FILENAME = N'c:\Program Files (x86)\Microsoft SQL
Server\MSSQL10.SQLEXPRESS\MSSQL\DATA\SuperSS.mdf', SIZE = 2304KB, MAXSIZE
= UNLIMITED, FILEGROWTH = 1024KB)
LOG ON
(NAME = N'SuperSS_log', FILENAME = N'c:\Program Files (x86)\Microsoft SQL
Server\MSSQL10.SQLEXPRESS\MSSQL\DATA\SuperSS_log.LDF', SIZE = 576KB,
MAXSIZE = 2048GB, FILEGROWTH = 10%)
GO
ALTER DATABASE [SuperSS] SET COMPATIBILITY_LEVEL = 100
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [SuperSS].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [SuperSS] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [SuperSS] SET ANSI_NULLS OFF
GO
ALTER DATABASE [SuperSS] SET ANSI_PADDING OFF
GO
ALTER DATABASE [SuperSS] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [SuperSS] SET ARITHABORT OFF
GO
ALTER DATABASE [SuperSS] SET AUTO_CLOSE ON
GO
ALTER DATABASE [SuperSS] SET AUTO_CREATE_STATISTICS ON
GO
ALTER DATABASE [SuperSS] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [SuperSS] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [SuperSS] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [SuperSS] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [SuperSS] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [SuperSS] SET NUMERIC_ROUNDABORT OFF
GO
```

```

ALTER DATABASE [SuperSS] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [SuperSS] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [SuperSS] SET ENABLE_BROKER
GO
ALTER DATABASE [SuperSS] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [SuperSS] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [SuperSS] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [SuperSS] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [SuperSS] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [SuperSS] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [SuperSS] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [SuperSS] SET READ_WRITE
GO
ALTER DATABASE [SuperSS] SET RECOVERY SIMPLE
GO
ALTER DATABASE [SuperSS] SET MULTI_USER
GO
ALTER DATABASE [SuperSS] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [SuperSS] SET DB_CHAINING OFF
GO
USE [SuperSS]
GO
/***** Object: Table [dbo].[PESAJESV2]  Script Date: 08/15/2021 04:30:33 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[PESAJESV2](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Balanza] [varchar](50) NULL,
    [Bruto] [varchar](50) NULL,
    [Descpcion] [varchar](50) NULL,
    [Objetivo] [varchar](50) NULL,
    [ToleranciaPos] [varchar](50) NULL,
    [ToleranciaNeg] [varchar](50) NULL,

```

```

        [Estado] [varchar](50) NULL,
        [Fecha] [date] NULL,
        [Hora] [varchar](50) NULL
    ) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[PESAJES]   Script Date: 08/15/2021 04:30:33 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[PESAJES](
    [ID] [int] NOT NULL,
    [Peso_G] [varchar](20) NOT NULL,
    [Descripcion] [varchar](20) NOT NULL,
    [Objetivo] [varchar](12) NOT NULL,
    [ToleranciaPos] [varchar](12) NOT NULL,
    [ToleranciaNeg] [varchar](12) NOT NULL,
    [Fecha] [date] NOT NULL,
    [Hora] [time](7) NOT NULL
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
/***** Object: Table [dbo].[PESAJE]   Script Date: 08/15/2021 04:30:33 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[PESAJE](
    [ID] [int] IDENTITY(1,1) NOT NULL,
    [Pesaje_G] [varchar](50) NULL,
    [Descripcion] [varchar](50) NULL,
    [Objetivo] [varchar](50) NULL,
    [ToleranciaPos] [varchar](50) NULL,
    [ToleranciaNeg] [varchar](50) NULL,
    [Fecha] [varchar](50) NULL,
    [Hora] [varchar](50) NULL,
    CONSTRAINT [PK_PESAJE] PRIMARY KEY CLUSTERED
(
    [ID] ASC

```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY  
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]  
GO  
SET ANSI_PADDING OFF  
GO
```