



**Universidad Nacional Mayor de San Marcos**

**Universidad del Perú. Decana de América**

**Facultad de Ingeniería de Sistemas e Informática**

**Escuela Profesional de Ingeniería de Software**

**Implementación de una interfaz hardware-software y  
sus módulos de supervisión y análisis de datos para un  
sistema de pesaje electrónico**

**TRABAJO DE SUFICIENCIA PROFESIONAL**

**Para optar el Título Profesional de Ingeniero de Software**

**AUTOR**

**Lennin Paúl QUIRÓZ VILLALOBOS**

**ASESOR**

**Cayo Víctor LEÓN FERNÁNDEZ**

**Lima, Perú**

**2018**



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

## Referencia bibliográfica

---

Quiróz, L. (2018). *Implementación de una interfaz hardware-software y sus módulos de supervisión y análisis de datos para un sistema de pesaje electrónico*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Software]. Repositorio institucional Cybertesis UNMSM.

---



**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**  
**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**  
*Escuela Profesional de Ingeniería de Software*

**Acta de Sustentación del  
Trabajo de Suficiencia Profesional**

Siendo las 19:40 horas del día 20 de noviembre del año 2018, se reunieron los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Ing. Molina Neyra Cesar Alberto (Presidente), Lic. Gil Calvo Rubén Alexander (Miembro) y el Mg. León Fernandez Cayo Victor (Miembro Asesor) para la sustentación del Trabajo de Suficiencia Profesional Intitulado: **"IMPLEMENTACIÓN DE UNA INTERFAZ HARDWARE-SOFTWARE Y SUS MÓDULOS DE SUPERVISIÓN Y ANÁLISIS DE DATOS PARA UN SISTEMA DE PESAJE ELECTRÓNICO"**, por el Bachiller: **Quiróz Villalobos, Lennin Paúl**; para obtener el Título Profesional de Ingeniero de Software.

Acto seguido de la exposición del Trabajo de Suficiencia Profesional, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecida por los miembros del Jurado.

El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el Bachiller obtuvo la nota de 17. (En letras) Dieci Siete.....

A continuación el presidente de jurados el Ing. Molina Neyra Cesar Alberto, declara al Bachiller Ingeniero de Software.

Siendo las 20:15 horas, se levantó la sesión.

Presidente

**Ing. Molina Neyra Cesar Alberto**

  
Miembro

**Lic. Gil Calvo Rubén Alexander**

Miembro Asesor

**Mg. León Fernández Cayo Victor**

## **FICHA CATALOGRÁFICA**

**IMPLEMENTACIÓN DE UNA INTERFAZ HARDWARE-SOFTWARE Y SUS MÓDULOS DE SUPERVISIÓN Y ANÁLISIS DE DATOS PARA UN SISTEMA DE PESAJE ELECTRÓNICO**

**AUTOR: QUIRÓZ VILLALOBOS, LENNIN PAÚL**

**ASESOR: Mg. LEÓN FERNANDEZ, CAYO VICTOR**

**LIMA – PERÚ, 2018**

**Título Profesional/Grado Académico: Título Profesional de Ingeniero de Software**

**Área/Programa/Línea de Investigación: Ingenierías / Tecnología de Información y Comunicación / Ingeniería de Software**

**Pregrado: Universidad Nacional Mayor de San Marcos – Facultad de Ingeniería de Sistemas e Informática – Escuela Profesional de Ingeniería de Software**

**Formato 28 x 20 cm**

**Páginas: xi, 89**

## **DEDICATORIA**

*“Dedicado este trabajo a Dios, por iluminar siempre mi camino y a mis queridos padres por guiarme desde pequeño y apoyarme siempre para alcanzar mis sueños”*

## **AGRADECIMIENTOS**

*Mi más profundo agradecimiento a mis profesores de la FISI-UNMSM, ya que en este trabajo está plasmado sus enseñanzas y esfuerzo por brindarnos siempre lo mejor de sus conocimientos. Del mismo modo, quiero expresar mi más sincero agradecimiento a mi asesor, el Mg. Cayo Victor León Fernandez, por sus grandes cualidades personales y profesionales, sin las cuales este trabajo no hubiera sido posible.*

**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**  
**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE SOFTWARE**

**Implementación de una interfaz hardware-software y sus módulos de supervisión y  
análisis de datos para un sistema de pesaje electrónico**

**Autor** : Quiróz Villalobos, Lennin Paúl  
**Asesor** : Mg. León Fernandez, Cayo Victor  
**Título** : Informe de Trabajo de Suficiencia Profesional para optar el Título  
Profesional de Ingeniero de Software  
**Fecha** : Diciembre de 2018

---

## **RESUMEN**

Capturar, monitorear y analizar los datos provenientes de los instrumentos de medición, es un paso indispensable en cualquier proceso de producción. Los equipos de pesaje industrial proveídos por Miyake SCRL, incluyen por defecto un panel indicador que sólo permite la visualización instantánea del valor medido de manera local y una interfaz serial de comunicación mediante cable, de corto alcance (2 metros), sin embargo ningún software de gestión de datos es proporcionado por el fabricante. Para solucionar este problema, el presente proyecto propuso proveer a los instrumentos de medición, de capacidad de comunicación inalámbrica, así como implementar los módulos software de captura, análisis, supervisión y visualización de datos desde un computador remoto y una aplicación móvil. Para implementar la solución propuesta, se usó un módulo convertidor Serial-Wifi, para transmitir los datos hacia una PC remota, sockets cliente-servidor en lenguaje Java para el módulo de captura de datos, MongoDB para el módulo de almacenamiento y análisis de datos, LabVIEW para el módulo de supervisión y visualización, finalmente AppInventor para la aplicación móvil para el Sistema Operativo Android. Como resultado del presente proyecto, Miyake SCRL, teniendo la plataforma software como valor agregado, ha mejorado la demanda de sus equipos de medición y ha obtenido una ventaja competitiva al ser reconocida en su rubro como empresa innovadora y generadora de tecnología. Así mismo, las empresas clientes, también se han beneficiado, ya que la solución implementada permite alinear la tecnología con sus objetivos empresariales, disponiendo ahora de información útil y oportuna para sus procesos de control, auditoría y toma de decisiones.

**Palabras clave:** interfaz hardware-software, sistema de pesaje electrónico, comunicación serial, convertidor serial-wifi, sockets cliente-servidor, aplicación móvil.

**MAJOR NATIONAL UNIVERSITY OF SAN MARCOS**  
**FACULTY OF SYSTEMS AND COMPUTER ENGINEERING**  
**PROFESSIONAL SCHOOL OF SOFTWARE ENGINEERING**

**Implementation of a hardware-software interface and its monitoring and data analysis  
modules for an electronic weighing system**

**Author** : Quiróz Villalobos, Lennin Paúl  
**Advisor** : Mg. León Fernandez, Cayo Victor  
**Title** : Professional Sufficiency Work Report for opt for the Professional Title of  
Software Engineer  
**Date** : December 2018

---

## **ABSTRACT**

Capture, monitoring and analyze the data coming from the measuring instruments is a mandatory step in any production process. The industrial weigh equipments provided by Miyake SCRL, includes by default an indicator panel that only allows the instantaneous visualization of the measured value locally and a serial communication interface via cable, short range (2 meters), however any software for data management is provided by the manufacturer. To solve this problem, the present project proposed to provide measurement instruments with wireless communication capacity, as well as implement software modules for capturing, analyzing, monitoring and visualizing data from a remote computer and a mobile application. To implement the proposed solution, was used a Serial-Wifi converter module in order to transmit the data to a remote PC, client-server sockets in the Java language for the data capture module, MongoDB for the storage and data analysis module, LabVIEW for the monitoring and visualization module, finally AppInventor for the mobile application for the Android Operating System. As result of the present project, Miyake SCRL, having the software platform as an added value, has improved the demand of its measurement equipments and obtained a competitive advantage when it is recognized as an innovative and technology generator company. Likewise, the client companies also were benefited, since the implemented solution allows to them to align the technology with its business goals, having useful and timely information for its processes of control, audit and decisions making.

**Keywords:** hardware-software interface, electronic weighing system, serial communication, serial-wifi converter, client-server sockets, mobile application.

# ÍNDICE

|                                                                               |      |
|-------------------------------------------------------------------------------|------|
| CARÁTULA EXTERNA.....                                                         | i    |
| PÁGINA EN BLANCO.....                                                         | ii   |
| CARÁTULA INTERNA.....                                                         | iii  |
| FICHA CATALOGRÁFICA.....                                                      | iv   |
| DEDICATORIA.....                                                              | v    |
| AGRADECIMIENTOS.....                                                          | vi   |
| RESUMEN.....                                                                  | vii  |
| ABSTRACT.....                                                                 | viii |
| ÍNDICE.....                                                                   | ix   |
| ÍNDICE DE FIGURAS.....                                                        | x    |
| ÍNDICE DE TABLAS.....                                                         | xi   |
| INTRODUCCIÓN.....                                                             | 1    |
| CAPÍTULO I - TRAYECTORIA PROFESIONAL.....                                     | 2    |
| CAPÍTULO II - CONTEXTO EN EL QUE SE DESARROLLÓ LA EXPERIENCIA.....            | 5    |
| 2.1 EMPRESA - ACTIVIDAD QUE REALIZA.....                                      | 5    |
| 2.2 VISIÓN.....                                                               | 6    |
| 2.3 MISIÓN.....                                                               | 6    |
| 2.4 ORGANIZACIÓN DE LA EMPRESA.....                                           | 6    |
| 2.5 AREA, CARGO Y FUNCIONES DESEMPEÑADAS.....                                 | 7    |
| 2.6 EXPERIENCIA PROFESIONAL REALIZADA EN LA ORGANIZACIÓN.....                 | 7    |
| CAPÍTULO III - ACTIVIDADES DESARROLLADAS.....                                 | 8    |
| 3.1 SITUACIÓN PROBLEMÁTICA.....                                               | 8    |
| 3.1.1 DEFINICIÓN DEL PROBLEMA.....                                            | 8    |
| 3.2 SOLUCIÓN.....                                                             | 8    |
| 3.2.1 OBJETIVO GENERAL.....                                                   | 8    |
| 3.2.2 OBJETIVOS ESPECÍFICOS.....                                              | 9    |
| 3.2.3 ALCANCE.....                                                            | 9    |
| 3.2.4 FUNDAMENTOS UTILIZADOS.....                                             | 9    |
| 3.2.5 ETAPAS Y METODOLOGÍA.....                                               | 21   |
| 3.2.6 IMPLEMENTACION DE LAS AREAS, PROCESOS, SISTEMAS Y BUENAS PRÁCTICAS..... | 22   |
| ▪ MÓDULO DE CONVERSIÓN DE DATOS SERIAL A WIFI.....                            | 23   |
| ▪ MÓDULO DE CAPTURA DE DATOS.....                                             | 26   |
| ▪ MÓDULO DE ALMACENAMIENTO EN LA BASE DE DATOS NOSQL.....                     | 32   |
| ▪ MÓDULO DE ANÁLISIS DE DATOS.....                                            | 37   |
| ▪ MÓDULO DE VISUALIZACIÓN DE OPERACIONES EN TIEMPO REAL.....                  | 42   |
| ▪ MÓDULO DE APLICACIÓN MÓVIL DE VISUALIZACIÓN.....                            | 44   |
| 3.3 EVALUACIÓN.....                                                           | 45   |
| 3.3.1 EVALUACIÓN ECONÓMICA.....                                               | 46   |
| 3.3.2 INTERPRETACIÓN DEL VAN Y DEL TIR.....                                   | 47   |
| CAPÍTULO IV. REFLEXIÓN CRÍTICA DE LA EXPERIENCIA.....                         | 48   |
| CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES.....                               | 49   |
| 5.1 CONCLUSIONES.....                                                         | 49   |
| 5.2 RECOMENDACIONES.....                                                      | 50   |
| 5.3 FUENTES DE INFORMACIÓN.....                                               | 51   |
| 5.4. GLOSARIO.....                                                            | 52   |
| ANEXOS.....                                                                   | 52   |

# ÍNDICE DE FIGURAS

|                                                                                                         |    |
|---------------------------------------------------------------------------------------------------------|----|
| Figura 1: Marcas de Productos que representa Miyake SCRL .....                                          | 5  |
| Figura 2: Estructura Organizacional de Miyake SCRL .....                                                | 6  |
| Figura 3: Configuración de pin-out para la comunicación serial RS-232 .....                             | 11 |
| Figura 4: Diagrama de tiempos para la comunicación serial RS-232 .....                                  | 11 |
| Figura 5: Identificación de pines del conector DB-9 para la comunicación serial RS-232 .....            | 13 |
| Figura 6: Flujo de mensajes entre cliente y servidor en una comunicación basada en Sockets .....        | 14 |
| Figura 7: Jerarquía de elementos constitutivos en MongoDB .....                                         | 15 |
| Figura 8: Ejemplo de Diagrama de Frecuencias implementado con JfreeChart .....                          | 17 |
| Figura 9: Logotipo característico de LabVIEW .....                                                      | 18 |
| Figura 10: Ejemplo de un Bucle FOR implementado en LabVIEW .....                                        | 20 |
| Figura 11: Descripción de las etapas de la solución implementada. ....                                  | 21 |
| Figura 12: Diagrama de Bloques con la Arquitectura del Sistema Implementado .....                       | 22 |
| Figura 13: Tipos de conversión proporcionadas por el módulo convertidor. ....                           | 23 |
| Figura 14: Módulo convertidor HLK-RM04 .....                                                            | 24 |
| Figura 15: Diagrama de bloques del módulo convertidor .....                                             | 24 |
| Figura 16: Adecuación del módulo convertidor en el interior del instrumento de medición .....           | 26 |
| Figura 17: Diagrama esquemático del Módulo de Captura de Datos .....                                    | 27 |
| Figura 18: Panel de configuración del módulo convertidor HLK-RM04 .....                                 | 27 |
| Figura 19: Ejecución del módulo de captura de datos. ....                                               | 32 |
| Figura 20: Diagrama esquemático del módulo de persistencia de datos .....                               | 32 |
| Figura 21: Creación del archivo de configuración de la BD MongoDB .....                                 | 33 |
| Figura 22: Creación del archivo launcher para la BD MongoDB .....                                       | 33 |
| Figura 23: Servidor MongoDB en modo listening .....                                                     | 34 |
| Figura 24: Archivo de Configuración para iniciar el Cliente MongoDB .....                               | 34 |
| Figura 25: Cliente MongoDB en ejecución. ....                                                           | 35 |
| Figura 26: Vista utilizando la herramienta RoboMongo, de los datos almacenados en la BD de Miyake. .... | 36 |
| Figura 27: Diagrama esquemático del Módulo de Análisis de Datos .....                                   | 37 |
| Figura 28: Ejemplo de Histograma de Operaciones generado desde la BD Mongo DB. ....                     | 42 |
| Figura 29: Diagrama Esquemático del módulo de Visualización o Monitoreo .....                           | 43 |
| Figura 30: Código en lenguaje G, del módulo de Visualización. ....                                      | 43 |
| Figura 31: Panel frontal del Módulo de Visualización en LabVIEW .....                                   | 44 |
| Figura 32: Diagrama Esquemático de la App de visualización para dispositivos móviles .....              | 44 |
| Figura 33: Ejecución de la Aplicación Móvil sobre un dispositivo Android .....                          | 45 |
| Figura 34: Análisis de VAN y TIR del proyecto implementado. ....                                        | 47 |

## ÍNDICE DE TABLAS

|                                                                                                   |    |
|---------------------------------------------------------------------------------------------------|----|
| Tabla 1: Configuraciones de velocidad disponibles para comunicación serial y sus parámetros ..... | 12 |
| Tabla 2: Proceso para establecer una comunicación cliente-servidor basada en Sockets.....         | 14 |
| Tabla 3: Principales características de la herramienta LabVIEW .....                              | 19 |
| Tabla 4: Componentes o partes de un programa implementado con LabVIEW .....                       | 19 |
| Tabla 5: Características de la herramienta ApplInventor .....                                     | 20 |
| Tabla 6: Especificaciones Técnicas del Módulo Convertidor HLK-RM04 .....                          | 25 |
| Tabla 7: Código fuente para la implementación del Socket Server, usando lenguaje Java.....        | 28 |
| Tabla 8: Código fuente para implementar la persistencia en MongoDB, usando lenguaje Java.....     | 35 |
| Tabla 9: Código fuente para el módulo de Análisis de datos, usando lenguaje Java. ....            | 37 |

# INTRODUCCIÓN

La cuantificación y el registro de las operaciones que se realizan en una cadena de producción, es un paso obligatorio no sólo con fines de control o para mantener información histórica, sino también para ayudar a tomar decisiones acerca de dimensionamiento de recursos, materias primas, separar espacios de almacenamiento o realizar auditorías por parte de la alta gerencia o de las entidades reguladoras.

El proyecto implementado, soluciona la necesidad de captura, almacenamiento, visualización y análisis de datos de los sistemas de pesaje distribuidos por Miyake SCRL a sus diferentes clientes a nivel nacional. La arquitectura de software implementada, consta de varios módulos independientes, cada uno con una función bien definida, sin embargo, interrelacionados para que al integrarse se pueda lograr el objetivo final de ofrecer una solución integral.

El presente informe está dividido en cinco capítulos. A continuación se describirá el contenido temático de cada uno de ellos.

El Capítulo I contiene un resumen de la trayectoria profesional del autor desde sus inicios como practicante en la empresa Miyake SCRL hasta la actualidad, también se describe su formación académica en la UNMSM y su experiencia como docente e investigador.

El Capítulo II, describe el contexto en que se desarrolló el presente trabajo, la actividad que realiza la empresa, su organización, las funciones desempeñadas y mi experiencia en la organización.

El Capítulo III, describe las actividades desarrolladas para la implementación de la solución propuesta, iniciando desde la descripción de la situación problemática, la formulación del problema, la solución implementada y la evaluación de la misma. Respecto a la solución implementada se describen los objetivos, el alcance, los fundamentos teóricos utilizados y finalmente las etapas y módulos que conforman la solución.

El capítulo IV, contiene una reflexión crítica de la experiencia, el beneficio que ha significado para la empresa desarrolladora, para las empresas cliente y las implicancias profesionales para el especialista a cargo de la implementación.

Finalmente, el Capítulo V, contiene las conclusiones y recomendaciones que quedan como enseñanza después de finalizar el presente proyecto.

## **CAPÍTULO I - TRAYECTORIA PROFESIONAL**

Profesional egresado de la E.A.P. de Ing. de Software FISI-UNMSM en el año 2013, habiéndome graduado como parte del Tercio Superior de mis compañeros de promoción. Durante mi etapa profesional, he logrado consolidar lo aprendido en las aulas, habiendo adquirido sólidos conocimientos y amplia experiencia en el campo de programación de aplicaciones, base de datos, aplicaciones móviles, interfaces y protocolos de comunicación de datos.

He participado como ponente en temas de Inteligencia Artificial, Robótica y Procesamiento de Imágenes en diversos congresos (CONEISC, COREINSI) y eventos a nivel nacional. Así mismo he formado parte de diversos grupos y proyectos de investigación (Internet de las Cosas, Glucómetro) en la UNMSM.

He ejercido la docencia en diversas instituciones educativas de nivel superior (CIBERTEC, IPAE, UNMSM) en cursos referidos a informática básica, programación en lenguaje Java y Base de Datos Oracle.

Mi trayectoria profesional, me ha permitido obtener habilidades y acumular experiencia en las siguientes áreas:

- Analista Programador de Aplicaciones con diferentes lenguajes y motores de Bases de Datos.
- Formulación y Gestión de Proyectos Tecnológicos con adopción de nuevas tecnologías y modelos de negocio.
- Dirección y Gestión de equipos de desarrollo.
- Programación de Aplicaciones Móviles.
- Diseño de Interfaces y protocolos de comunicación.
- Proyectos de Investigación.
- Docencia Superior.

## Experiencia Profesional

| EMPRESA / INSTITUCIÓN                                                                                             | FUNCIONES DESEMPEÑADAS                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MIYAKE DISTRIBUCIONES S.C.R.L<br>Jefe de Proyectos y Soporte Técnico<br>(Enero 2014 – Actualidad)                 | Responsable de la implementación de los Proyectos Informáticos de la empresa, el mantenimiento de la infraestructura de TI, así como el desarrollo de soluciones a medida, asistencia técnica y capacitación a clientes. |
| MIYAKE DISTRIBUCIONES S.C.R.L<br>Practicante de Informática<br>(Agosto 2012 – Diciembre 2013)                     | Me desempeñé como Practicante del Área de Informática, habiendo desempeñado funciones como programador de aplicaciones y actividades de Soporte Técnico a la infraestructura informática de la empresa.                  |
| CIBERTEC<br>Sede Lima Norte – Los Olivos<br>(2014)                                                                | Me desempeñé como docente del Área de Tecnología, habiendo tenido a mi cargo el dictado del curso de Arquitectura de Computadoras.                                                                                       |
| UNMSM<br>Universidad Nacional Mayor de San Marcos<br>Facultad de Ing. de Sistemas e Informática<br>(2012-2013-II) | Me desempeñé como docente Jefe de Práctica, de la EAP de Ing. de Software, habiendo tenido a mi cargo los siguientes cursos:<br>-Programación I<br>-Estructura de Datos II                                               |

## Formación Académica

|                                                                                                                                                            |                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| UNMSM<br>Universidad Nacional Mayor de San Marcos<br>Facultad de Ing. de Sistemas e Informática<br>Escuela Académico Profesional de Ingeniería de Software | Grado Académico,<br>Bachiller en Ingeniería de Software,<br>Abril 2014. |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------|

|                                                                                                                                                            |                                                                                     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| UNMSM<br>Universidad Nacional Mayor de San Marcos<br>Facultad de Ing. de Sistemas e Informática<br>Escuela Académico Profesional de Ingeniería de Software | Egresado,<br>E.A.P. de Ingeniería de Software<br>Tercio Superior<br>Diciembre 2013. |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|

### Participación en Eventos y Congresos

|                            |                                                                                                           |
|----------------------------|-----------------------------------------------------------------------------------------------------------|
| <b>Evento:</b>             | CONEISC 2013 – Congreso Nacional de Estudiantes de Ingeniería de Sistemas y Computación.<br>(Agosto 2013) |
| <b>Participación como:</b> | <b>PONENTE</b>                                                                                            |
| <b>Tema:</b>               | “Diseño de un robot explorador con Inteligencia Artificial usando Lego NXT y Java Netbeans”               |

|                            |                                                                                             |
|----------------------------|---------------------------------------------------------------------------------------------|
| <b>Evento:</b>             | FISI UNMSM.– Semana de divulgación en Investigación.<br>(Diciembre 2013)                    |
| <b>Participación como:</b> | <b>PONENTE</b>                                                                              |
| <b>Tema:</b>               | “Implementación de un robot explorador con Inteligencia Artificial usando Java y Bluetooth” |

|                            |                                                                                        |
|----------------------------|----------------------------------------------------------------------------------------|
| <b>Evento:</b>             | XI COREINSI – Congreso Regional de Ingeniería de Sistemas e Informática<br>(Mayo 2018) |
| <b>Participación como:</b> | <b>PONENTE</b>                                                                         |
| <b>Tema:</b>               | “Técnicas de Detección de Humanos para aplicaciones de Robótica”                       |

## **CAPÍTULO II - CONTEXTO EN EL QUE SE DESARROLLÓ LA EXPERIENCIA**

### **2.1 EMPRESA - ACTIVIDAD QUE REALIZA**

Miyake Distribuciones SCRL, es una empresa peruana, con más de 20 años de experiencia, en el rubro de Balanzas Electrónicas, Cajas Registradoras, Empacadoras al vacío, Impresoras Térmicas, Scanner, siendo representante exclusivo en Perú de reconocidas marcas a nivel global como Uniwell, CAS, EPSON, lo que le permite, en su calidad de distribuidor oficial, brindar a sus clientes productos con las tecnologías más recientes en cada rubro y con el respaldo técnico directo del fabricante.

Cuenta con distintas líneas de productos, tales como:

- Cajas Registradoras Convencionales, Puntos de Venta Touch Screen, Impresoras Térmicas, Scanner, Gavetas, contómetros y accesorios.
- Balanzas Electrónicas para aplicaciones especializadas como explosión-proof, water-proof, con las más estrictas certificaciones de uso industrial a nivel mundial.
- Además, Balanzas Comerciales, Industriales, de Laboratorio y Sector Salud, así como también, toda una gama de soluciones de pesaje de alta capacidad para compañías mineras, transporte de mercancías.
- Cortadoras de Embutidos, Fileteadoras, Moledoras, Empacadoras Manuales, Empacadoras al Vacío.
- Equipos de refrigeración comercial, licuadoras industriales y accesorios.

La figura 1, muestra algunas de las principales marcas, distribuidas por Miyake S.C.R.L.



**Figura 1: Marcas de Productos que representa Miyake SCRL**

## 2.2 VISIÓN

Nuestro calificado y selecto personal está capacitado para brindarle asesoría y el mejor servicio, con los mejores productos, con la más alta calidad y eficiencia. Así mismo nuestros especialistas le brindarán el mejor soporte técnico, con todo el respaldo de nuestras marcas de fábrica.

## 2.3 MISIÓN

Miyake es un equipo de profesionales dedicado a brindarle soluciones a sus necesidades de equipamiento en su negocio como: Puntos de Venta, Balanzas Electrónicas, Cajas Registradoras Touch, periféricos y afines. Representamos en el Perú, marcas reconocidas a nivel mundial, con soporte directo del fabricante, sin intermediarios.

## 2.4 ORGANIZACIÓN DE LA EMPRESA

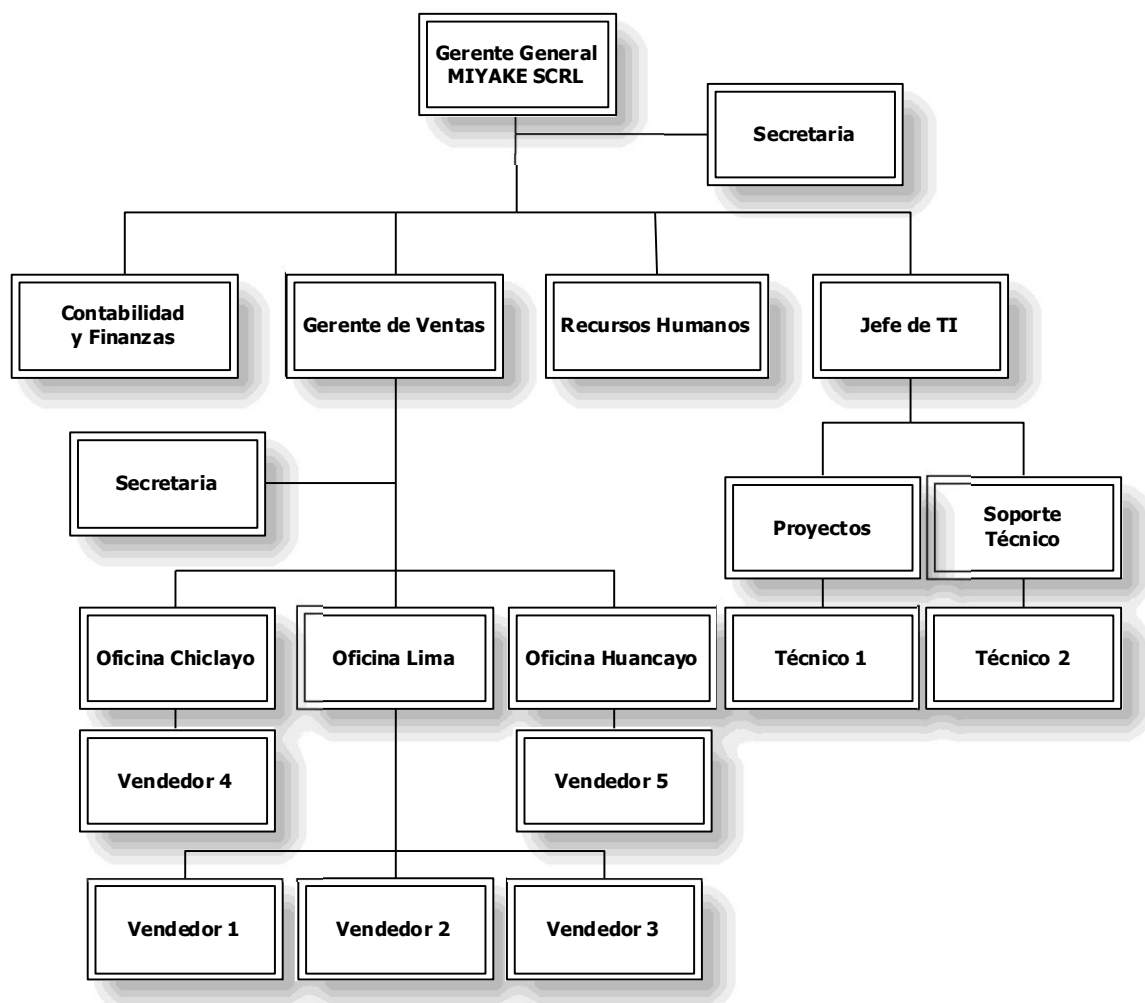


Figura 2: Estructura Organizacional de Miyake SCRL

## **2.5 AREA, CARGO Y FUNCIONES DESEMPEÑADAS**

Desde Agosto 2012, me desempeñé en MIYAKE DISTRIBUCIONES S.C.R.L como Practicante del Área de Informática, habiendo realizado funciones como programador de aplicaciones y actividades de Soporte Técnico a la infraestructura informática de la empresa.

Desde enero 2014, me desempeño como Jefe de Proyectos y Soporte Técnico, habiendo sido responsable de la implementación de los Proyectos Tecnológicos de la empresa, el mantenimiento de la infraestructura de TI, así como el desarrollo de soluciones a medida, asistencia técnica y capacitación a clientes.

Así mismo he tenido a mi cargo, la elaboración de propuestas de mejora en los procesos internos de la empresa, gestión de servicios de mantenimiento a clientes, evaluación de nuevos productos y tecnologías y capacitación a personal técnico de la empresa.

## **2.6 EXPERIENCIA PROFESIONAL REALIZADA EN LA ORGANIZACIÓN**

El haber laborado en Miyake SCRL, desde mis inicios como practicante y el haber realizado labores de mantenimiento como Técnico de Campo, me permitió conocer de cerca la problemática y las necesidades de los clientes, respecto al control y monitoreo de las operaciones de los instrumentos de medición.

Luego, cuando desde la Jefatura de TI, se propuso la implementación del presente proyecto, tuvo amplia aceptación, debido a los evidentes beneficios y oportunidad de negocio que significaba para la empresa, cubrir esa necesidad de los clientes.

Por esta razón, el proyecto, tuvo como objetivo implementar una solución al problema de comunicaciones, captura, almacenamiento, análisis y visualización de datos de los equipos de medición distribuidos por Miyake SCRL a sus diferentes clientes a nivel nacional.

.

## **CAPÍTULO III - ACTIVIDADES DESARROLLADAS**

### **3.1 SITUACIÓN PROBLEMÁTICA**

En este apartado se describirá cuál fue la situación problemática inicial que motivó la implementación del presente proyecto.

#### **3.1.1 DEFINICIÓN DEL PROBLEMA**

Los equipos industriales de medición de peso en sus diferentes modelos y capacidades, son distribuidos por el fabricante sin incluir la plataforma software que permite extraer, registrar y analizar los datos de las operaciones realizadas. Así mismo la interfaz de comunicaciones estándar incluida por defecto es bastante básica, limitándose únicamente a una interfaz serial cableada con unos pocos metros de alcance.

Sin embargo, para los propietarios de negocios y empresas de diferentes rubros, usuarios de los equipos de medición y clientes de MIYAKE SCRL, llevar un registro digital de las operaciones realizadas se hace imprescindible no sólo con el fin de controlar mercaderías, productos, ventas, sino también con fines de tributación, optimización de procesos y análisis de datos.

Así mismo en este tipo de dispositivos, al no contar con interfaz de comunicación inalámbrica, no era posible visualizar en tiempo real las operaciones realizadas de manera remota en un computador o en un dispositivo móvil, con fines de control, gestión o auditoría.

### **3.2 SOLUCIÓN**

A continuación, se detalla, la solución implementada desde los objetivos, el alcance hasta los módulos que la componen.

#### **3.2.1 OBJETIVO GENERAL**

Implementar la plataforma hardware-software para la adquisición, persistencia y análisis de datos de los sistemas de pesaje industrial distribuidos por MIYAKE S.C.R.L.

### **3.2.2 OBJETIVOS ESPECÍFICOS**

- Implementar la interfaz de comunicación RS-232 - Wifi, para enlazar los dispositivos de medición con un computador remoto de forma inalámbrica.
- Implementar el módulo de captura de datos provenientes del dispositivo de medición, bajo el modelo cliente-servidor.
- Diseñar e Implementar el modelo de persistencia de datos, para almacenar las operaciones realizadas en una Base de Datos no relacional.
- Implementar funcionalidades de análisis de datos para el apoyo a las decisiones de los clientes usuarios del sistema.
- Diseñar e implementar el sistema de visualización de operaciones, en tiempo real en un computador remoto.
- Diseñar la aplicación móvil, para el proceso de visualización y auditoria de operaciones de forma remota.

### **3.2.3 ALCANCE**

En la medida que MIYAKE SCRL es representante oficial de las marcas CAS, CAISUN, DIGI en Perú. El sistema únicamente garantiza la compatibilidad con la mayoría de los modelos de sistemas de pesaje de estas marcas.

El sistema usa un modelo Cliente-Servidor, para las comunicaciones, por lo tanto, la captura, almacenamiento y visualización de datos está supeditada a que ambos dispositivos de encuentren en la misma LAN corporativa. No está soportado la comunicación o envío de datos a través de internet o a servicios cloud.

### **3.2.4 FUNDAMENTOS UTILIZADOS**

A continuación, se describen los principios, técnicas, estándares y metodologías que han servido de base para implementar el presente trabajo.

#### ***Comunicación Serial***

Se denomina comunicación serie a aquella en la que se transmite la información por un solo canal de datos, en contraposición con la comunicación paralela, donde la transmisión es a través de múltiples líneas en simultáneo. Se puede notar la diferencia entre estas dos técnicas por la velocidad con que se transmiten los datos. Mientras en la comunicación serie se dispone de un solo cable por donde se envían los datos, en la

comunicación paralela aumenta notablemente la velocidad de transmisión al disponer de varios cables que entregan a cada instante un paquete de datos completo (por ejemplo, un paquete de 1 byte = 8 bits simultáneos). (Rs & Bihurriet, 2014)

A pesar de que en paralelo la comunicación es más eficiente respecto a velocidad (en bit/s = bps = baudio) a costo de más cables, si se considera el alcance y el largo del cable permitido por el paralelo, se encuentran más limitaciones que con la comunicación serie, esto se debe a que la diafonía o “crosstalk “ (una señal perturba otra señal cercana) produce interferencias entre las líneas paralelas, siendo cada vez más notoria cuanto más largo se hace el cable. (Rs & Bihurriet, 2014)

En la comunicación serie se presentan dos modos de transmisión, el síncrono y el asíncrono. Con éstos métodos podemos mantener los dispositivos cooperando en perfecto sincronismo, requerido para el envío y recepción de datos, evitando que un desfase en la toma de muestras del receptor altere la interpretación.

En el método sincrónico el reloj es compartido por ambos dispositivos, éste puede estar generado por uno de ellos o bien por una fuente externa. De este modo ambos dispositivos se encuentran sincronizados en todo momento. En cambio, el método asíncrono funciona con un reloj en cada dispositivo, sincronizados a la misma frecuencia, pero susceptibles a pequeñas diferencias que tarde o temprano conllevan a un desfase. Este problema es resuelto mediante la transmisión de paquetes de largo fijo, re-sincronizando entre la recepción de un paquete y otro. (Foreto Saboya, 2012) (Rs & Bihurriet, 2014)

### ***Estándar RS-232***

El protocolo RS-232, utiliza la técnica de comunicación serial. Tal como se aprecia en la Figura 3, bajo éste protocolo, los datos se transfieren por un único cable (más el cable común, tierra o GND"), por lo que la transmisión debe hacerse con un ratio de un bit, completando los datos conforme avanza el tiempo. (Rs & Bihurriet, 2014)

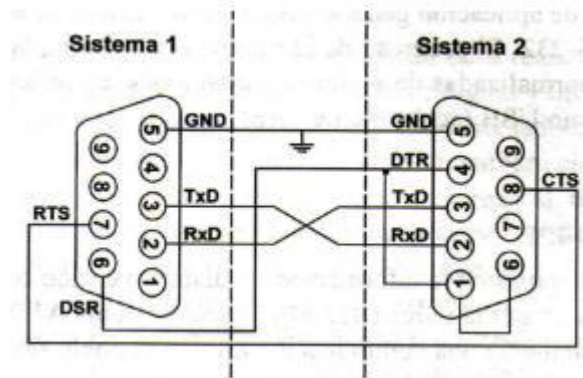


**Figura 3: Configuración de pin-out para la comunicación serial RS-232**

Fuente: (Foreto Saboya, 2012)

Cuando se transfieren valores digitales a un ratio de 1 bit por unidad de tiempo, se puede dividir los bits del flujo de datos tal como se aprecia en la Figura 4, esto es, un paquete de datos de 5 a 8 bits más un bit paridad y dos bits indicadores.

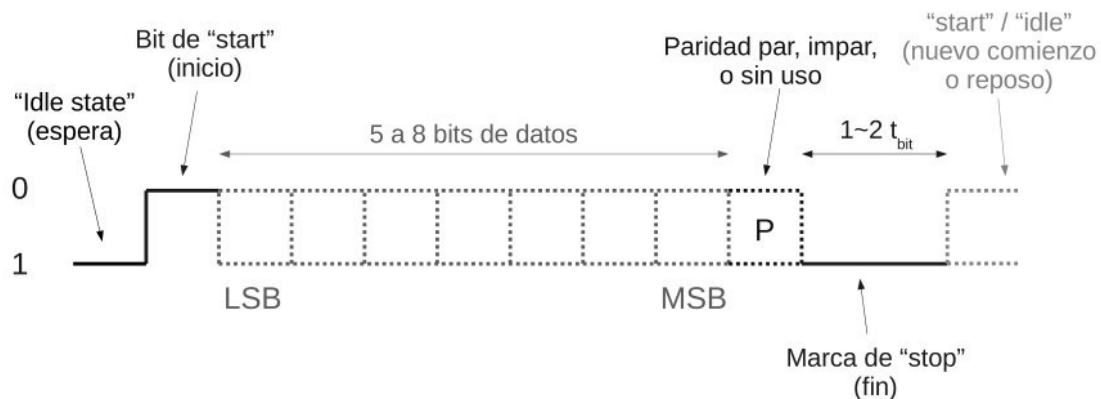


**Figura 4: Diagrama de tiempos para la comunicación serial RS-232**

Fuente: (Rs & Bihurriet, 2014)

Si por un tiempo indefinido la línea permanece en 1, no hay transmisión de datos y el receptor se mantiene en espera ("idle state"). Cuando se cambia a 0, este primer bit, conocido como bit de "start", indica que comenzará una nueva transmisión de datos (para un tamaño de datos y una velocidad previamente determinados). Este es el bit que permite que el receptor inicializa la sincronización, poniendo en fase su "clock" con el del transmisor.

Luego viene el paquete de datos (5 a 8 bits, comenzando el envío desde el bit menos significativo) seguido de un bit de paridad. Si la convención es paridad par, implica que el bloque total (incluido el bit de paridad) debe tener una cantidad par de unos.

Si en cambio se habla de paridad impar, el transmisor genera el bit de paridad de manera que el bloque total tenga un número impar de unos. Esto permite detectar errores simples de transmisión, aunque el receptor bien podrá ignorar este chequeo. El bit de paridad es opcional y esto se aclara previamente mediante una convención. (Rs & Bihurriet, 2014)

Para indicar los parámetros en la transmisión de datos, tales como la velocidad de bits, la cantidad de bits de datos, el tipo de paridad (o su ausencia) y la longitud del bit de "stop", existe una nomenclatura descrita en la Tabla 1.

**Tabla 1: Configuraciones de velocidad disponibles para comunicación serial y sus parámetros**

| Velocidad de bits | Bits de datos | Paridad                   | Tiempo de ' <i>stop bit</i> ' |
|-------------------|---------------|---------------------------|-------------------------------|
| 4800              | 5             | N: ' <i>None</i> ' (sin)  | 1                             |
| 9600              | 6             | E: ' <i>Even</i> ' (par)  | 1,5                           |
| 19200             | 7             | O: ' <i>Odd</i> ' (impar) | 2                             |
| etc..             | 8             |                           |                               |

Fuente: (Rs & Bihurriet, 2014)

Así, para una transmisión a 9600 bps, con 8 bits de datos, sin paridad, con bit de stop de longitud 1 tiempo de bit (la utilizada en este trabajo), la nomenclatura resultante es 9600/8N1.

El estándar considera uno lógico a todo valor de voltaje entre -5 y -12 volts, y un cero lógico a aquellos valores entre 5 y 12 volts positivos, con una zona de transición (zona de operación prohibida) de 5 a -5 volts. La interfaz se utiliza a una razón de menos de 20Kbps para una distancia menor de 15m.(David Luis la Red Martínez, 2009)

Cuando se transmite a través de un modem la norma define un conjunto de 22 señales divididas en señal de datos y señal de control distribuidas en un conector de tipo D de 25 terminales, sin embargo, no todas las señales de control son imprescindibles para establecer la comunicación entre dos equipos, es por eso que en muchas ocasiones se utiliza un conector macho tipo D de 9 terminales, tal como se muestra en la Figura 5. (Suarez, 2013)

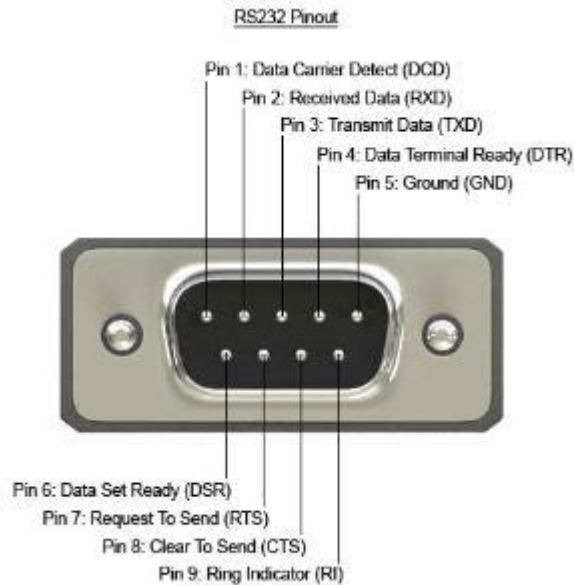


**Figura 5: Identificación de pines del conector DB-9 para la comunicación serial RS-232**

Fuente: (Suarez, 2013)

### *Sockets*

Los sockets son formas en las que podemos interconectar 2 (o más) programas que están ejecutándose en dispositivos diferentes de una red. Para crear una conexión mediante sockets se requiere básicamente conocer la dirección IP y el número de puerto por el que se establecerá la conexión. (Vlajic, 2016)

Cualquier conexión basada en sockets, requerirá de dos componentes:

**Servidor:** El programa servidor permanecerá “escuchando” en un número de puerto determinado. (Roffé, 2009)

**Cliente:** Debe saber la IP y el puerto por el que está escuchando el servidor. Para que sea posible la comunicación cliente-servidor, el cliente deberá establecer sockets y conectarlos a los sockets del servidor. (Roffé, 2009)

Los pasos que se siguen en una comunicación típica con sockets TCP, son mostrados en la Tabla 2.

**Tabla 2: Proceso para establecer una comunicación cliente-servidor basada en Sockets**

|    |                                                                                                              |
|----|--------------------------------------------------------------------------------------------------------------|
| 1° | Se crean los sockets en el cliente y el servidor.                                                            |
| 2° | El servidor establece el puerto por el que proporcionará el servicio.                                        |
| 3° | El servidor permanece a la escucha de peticiones de los clientes por el puerto definido en el paso anterior. |
| 4° | Un cliente conecta con el servidor.                                                                          |
| 5° | El servidor acepta la conexión.                                                                              |
| 6° | Se realiza el intercambio de datos.                                                                          |
| 7° | El cliente o el servidor, o ambos cierran la conexión.                                                       |

A continuación, La Figura 6, muestra de forma gráfica el flujo de mensajes entre Cliente-Servidor en una comunicación basada en Sockets.

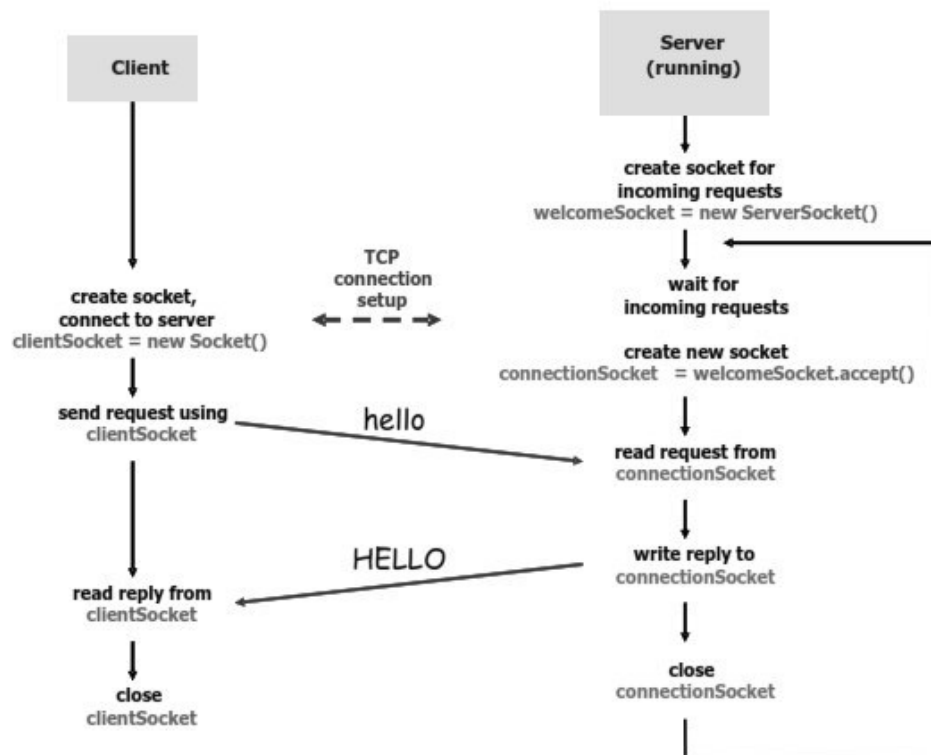


**Figura 6: Flujo de mensajes entre cliente y servidor en una comunicación basada en Sockets**

Fuente: (Vlajic, 2016)

Debemos tener en cuenta que para que la conexión mediante sockets sea posible, el servidor debe estar en ejecución antes que los clientes intenten conectarse a él y debe mantener activo un socket que permanezca a la espera de conexiones entrantes. (Albert, Rey, Carlos, & Castro, 2000)

Para establecer la conexión TCP, el cliente debe crear un socket con la dirección IP y el número de puerto de la aplicación que proporciona el servicio deseado. Sólo cuando se establece una conexión activa entre el socket del cliente y el socket del servidor, ambas aplicaciones pueden enviar o recibir datos.(Vila, 2008)(Vlajic, 2016)

## Bases de Datos NoSQL

MongoDB es un sistema de base de datos NoSQL multiplataforma de licencia libre. Está orientado a documentos de esquema libre, lo que implica que cada registro puede tener un esquema de datos distinto, (los atributos o campos no tienen que repetirse entre los diferentes registros). (Sánchez, 2012)

En MongoDB, cada registro o conjunto de datos se denomina documento, que pueden ser agrupados en colecciones, (equivalente a las tablas de las bases de datos relacionales, pero sin estar sometidos a un esquema fijo). Se pueden crear índices para algunos atributos de los documentos. (MongoDB.Inc, 2016)

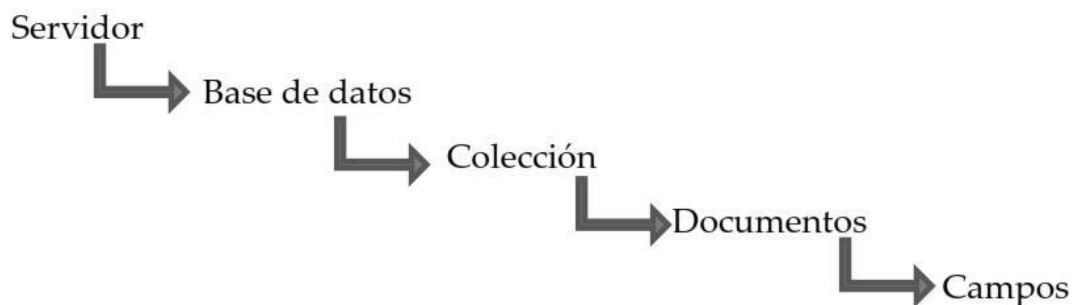


**Figura 7: Jerarquía de elementos constitutivos en MongoDB**  
Fuente: (Sánchez, 2012)

La jerarquía principal de MongoDB viene dada por los elementos presentados en la Figura 7. La diferencia con los SGBDR, (Sistemas Gestores de Bases de Datos Relacionales), es que no utilizamos tablas, filas ni columnas como allí lo hacíamos, sino que utilizamos documentos con distintas estructuras. Un conjunto de Campos formará un Documento, que en caso de asociarse con otros formará una Colección. Las bases de datos estará formada por Colecciones, y a su vez, cada servidor puede tener tantas bases de datos como el equipo lo permita.(Docs.mongodb.com, 2018) (Sánchez, 2012)(MongoDB.Inc, 2016)

Para el intercambio de datos para almacenamiento y transferencia de documentos en MongoDB usamos el formato BSON, (Binary JavaScript Object Notation). Se trata de una representación binaria de estructuras de datos y mapas, diseñada para ser más ligero y eficiente que JavaScript Object Notation (JSON).(Docs.mongodb.com, 2018)

Las bases de datos relacionales comerciales con capacidad distribuida suelen tener unas licencias realmente costosas, por lo que resulta interesante buscar alternativas eficientes y de bajo coste para nuestro sistema.(Sánchez, 2012)

Las bases de datos NoSQL hacen gala de una alta escalabilidad horizontal, que en resumidas cuentas implica la capacidad de adaptación de un sistema al rápido aumento de datos. La escalabilidad horizontal supone trabajar con varias máquinas de manera distribuida, almacenando en cada uno de los nodos cierta información que de una forma u otra debe estar comunicada con el resto de nodos que forman nuestro sistema. Esto dota de mayor flexibilidad al sistema, ya que facilita la agregación de equipos en función de las necesidades. (Amazon.com, 2018)

Entre las ventajas inherentes a las bases de datos NoSQL, podemos identificar:  
(Amazon.com, 2018)

- Debido a que soporta estructuras distribuidas presenta gran posibilidad de escalabilidad a bajo coste.
- Es una Base de datos abierta y flexible que se puede adaptar a las necesidades de diferentes proyectos de manera mucho más sencilla que los modelos de Entidad Relación.
- Se pueden hacer cambios de los esquemas sin tener que detener la Base de Datos.
- Escalabilidad horizontal: es capaz de crecer en número de máquinas, en lugar de tener que residir en una gran máquina.
- Se puede ejecutar en máquinas con pocos recursos.
- Proporciona consultas optimizadas para grandes cantidades de datos.

## ***JfreeChart***

JFreeChart es un conjunto de librerías open source para el lenguaje de programación Java, las cuales permiten crear de forma simple gráficos complejos a partir de data estructurada. JfreeChart puede generar diferentes tipos de gráficas, incluyendo cuadros combinados. A continuación se detallan todos los tipos de gráficos que son soportados:(Gilbert, 2008) (JFreeChart.Documentation, 2018)

- Gráficos XY (línea, spline y dispersión) con eje de tiempo.
- Gráfico circular.
- Diagrama de Gantt.
- Gráficos de barras (horizontal, vertical, apilados o independientes).
- Graficador de histogramas.
- Gráficas específicas: gráfica polar, burbujas.
- Permite adicionar varios tipos de marcadores en el área de dibujo.

Una ventaja adicional de JFreeChart es que dibuja automáticamente las escalas de los ejes y leyendas. Incluso por medio del mouse se puede hacer zoom en la interfaz de la gráfica automáticamente y cambiar algunos ajustes a través del menú local. Las tablas existentes pueden actualizarse fácilmente a través de los escuchadores(o listeners) que la biblioteca tiene disponible.(Omar & Flores, 2014).

La Figura 8, muestra un ejemplo de un diagrama de Frecuencias, graficado usando JfreeChart.

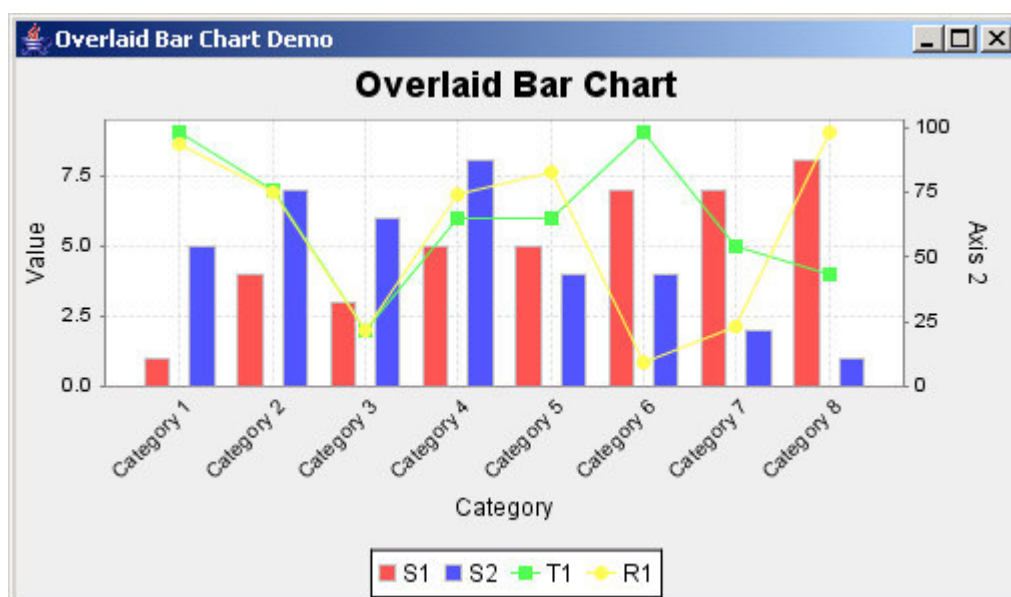


**Figura 8: Ejemplo de Diagrama de Frecuencias implementado con JfreeChart**

## ***LabView***

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) es una plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual gráfico. Recomendado para sistemas hardware y software de pruebas, control y diseño, simulado o real y embebido, pues acelera la productividad debido a su interfaz visual. El lenguaje que usa se llama lenguaje G, donde la G simboliza que es lenguaje Gráfico. (Roncancio & Cifuentes, 2000)



**Figura 9: Logotipo característico de LabVIEW**

Los programas desarrollados con LabVIEW se llaman Instrumentos Virtuales, o VIs, y su origen proviene del control de instrumentos industriales al que inicialmente estuvo orientado LabVIEW, aunque hoy en día esta funcionalidad se ha extendido grandemente con una gran cantidad de nuevas funciones y dispositivos. (Roncancio & Cifuentes, 2000)

Entre sus objetivos están el reducir el tiempo de desarrollo de aplicaciones de todo tipo, optimizar el uso de todos los núcleos disponibles del CPU y proporcionar una interfaz gráfica potente e intuitiva. Al mismo tiempo LabVIEW es compatible prácticamente con todo tipo de hardware (tarjetas de adquisición de datos, interfaces) y software de diferentes fabricantes. (National.Instruments.Inc, 2013)

En la Tabla 3, se presentan algunas características más resaltantes de LabVIEW. (National.Instruments.Inc, 2013)

**Tabla 3: Principales características de la herramienta LabVIEW**

|                                                    |                                                                                                                                                                                                                           |
|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Interfaces de comunicaciones:                      | <p>Puerto serie</p> <p>Puerto paralelo</p> <p>TCP/IP, UDP, DataSocket</p> <p>Bluetooth</p> <p>USB</p>                                                                                                                     |
| Compatibilidad con otros lenguajes y aplicaciones: | <p>DLL: librerías de funciones</p> <p>Matlab/Simulink</p> <p>AutoCAD, SolidWorks</p>                                                                                                                                      |
| Herramientas y funcionalidades avanzadas.          | <p>Herramientas gráficas y textuales para el procesado digital de señales.</p> <p>Visualización y manejo de gráficas con datos dinámicos.</p> <p>Adquisición y tratamiento de imágenes.</p> <p>Control de movimiento.</p> |

En LabVIEW cada VI consta de dos partes bien diferenciadas, tal como se muestra en la Tabla 4: (Roncancio & Cifuentes, 2000)

**Tabla 4: Componentes o partes de un programa implementado con LabVIEW**

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Panel Frontal:</i>       | <p>Es la interfaz gráfica, la utilizamos para interactuar con el usuario cuando el programa se está ejecutando. Los usuarios podrán observar los datos del programa actualizados en tiempo real.</p> <p>En esta interfaz se definen los <i>controles</i> (los usamos como entradas, pueden ser botones, marcadores, etc..) e <i>indicadores</i> (los usamos como salidas, pueden ser gráficas, paneles, etc).</p> |
| <i>Diagrama de Bloques:</i> | <p>Es el programa propiamente dicho, donde se define la funcionalidad, aquí se colocan íconos que realizan una determinada función y se corresponden uno a uno con los elementos gráficos del Panel Frontal.</p>                                                                                                                                                                                                  |

A continuación, en la Figura 10, se muestra un ejemplo del Diagrama de Bloques para un bucle FOR, elaborado con LabVIEW.

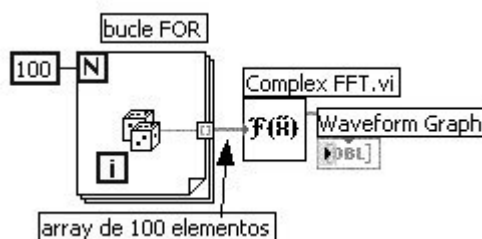


**Figura 10: Ejemplo de un Bucle FOR implementado en LabVIEW**

## App Inventor

App Inventor es un entorno de desarrollo de software soportado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android. El usuario puede, de forma visual y a partir de un conjunto de herramientas predefinidas, ir enlazando una serie de módulos para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web. Las aplicaciones creadas con App Inventor permiten cubrir un gran número de necesidades básicas en un dispositivo móvil. (Derek Walter, 2015)

Google App Inventor, tiene como objetivo democratizar el desarrollo de software y promover la investigación en tecnología entre los jóvenes. (AppInventor.MIT.OficialWeb, 2018)

En la Tabla 5, se destacan las principales características de AppInventor. (Gaddis & Halsey, 2015)

**Tabla 5: Características de la herramienta AppInventor**

|   |                                                                                                                                                                         |
|---|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Basado en [http Blockly] de JavaScript para crear un lenguaje visual. Estas librerías están distribuidas por Massachusetts Institute of Technology bajo licencia libre. |
| 2 | Permite programar aplicaciones muy complejas en mucho menos tiempo respecto a los lenguajes tradicionales, basados en texto.                                            |
| 3 | App Inventor se ejecuta como un servicio web para el desarrollo de aplicaciones móviles administrado por el MIT.                                                        |
| 4 | La interfaz gráfica, permite al usuario crear aplicaciones eficientes con muchas funcionalidades.                                                                       |

### 3.2.5 ETAPAS Y METODOLOGÍA

La Figura 11, muestra las etapas que fueron necesarias para la construcción de la solución y el orden de precedencia entre ellas.

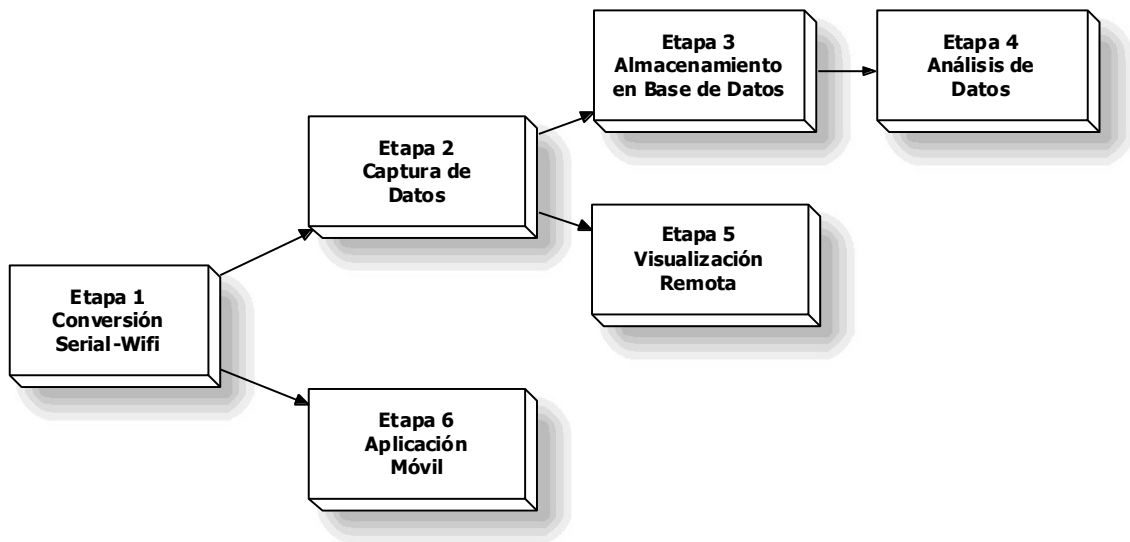


Figura 11: Descripción de las etapas de la solución implementada.

#### Metodología

**Etapa 1:** lo primero que fue necesario es realizar la conversión de serial a Wifi para poder transmitir los valores de las mediciones a una PC remota de manera inalámbrica. Esto se logró adecuando en el dispositivo de medición una tarjeta convertidora de Serial a Ethernet, de fabricación china.

**Etapa 2:** El siguiente paso, fue realizar la captura de los datos provenientes de forma inalámbrica del equipo de medición. Esta etapa fue tal vez la más crítica, ya que se tuvieron que realizar muchas pruebas e investigación al respecto. Finalmente se logró realizar la captura de datos implementando una aplicación en lenguaje Java basada en sockets.

**Etapa 3:** los datos resultantes de la captura, fue necesario almacenarlos con fines de control y análisis posterior. Esto se logró implementando una base de datos MongoDB.

**Etapa 4:** El módulo de análisis de datos usa la biblioteca JfreeChart, para generar información útil a partir de los datos almacenados en la base de datos.

**Etapa 5:** Con fines de supervisión, se implementó el módulo de visualización o supervisión remota, en tiempo real de las operaciones realizadas. Esto se logró mediante los elementos gráficos de LabView.

**Etapa 6:** con fines de auditoria y brindar soporte a las decisiones de la alta dirección, se desarrolló el módulo de visualización para dispositivos móviles Android. Esto se logró mediante la plataforma que AppInventor tiene disponible para el desarrollo de aplicaciones móviles.

### 3.2.6 IMPLEMENTACION DE LAS AREAS, PROCESOS, SISTEMAS Y BUENAS PRÁCTICAS

Los módulos que componen el sistema implementado, se describen a continuación:

- I. Módulo de conversión de datos Serial a Wifi
- II. Módulo de Captura de Datos
- III. Módulo de Almacenamiento en Base de Datos NoSQL
- IV. Módulo de Análisis de Datos.
- V. Módulo de Visualización de Operaciones en Tiempo Real
- VI. Aplicación móvil de visualización

En la Figura 12 se muestra la arquitectura de la solución implementada:

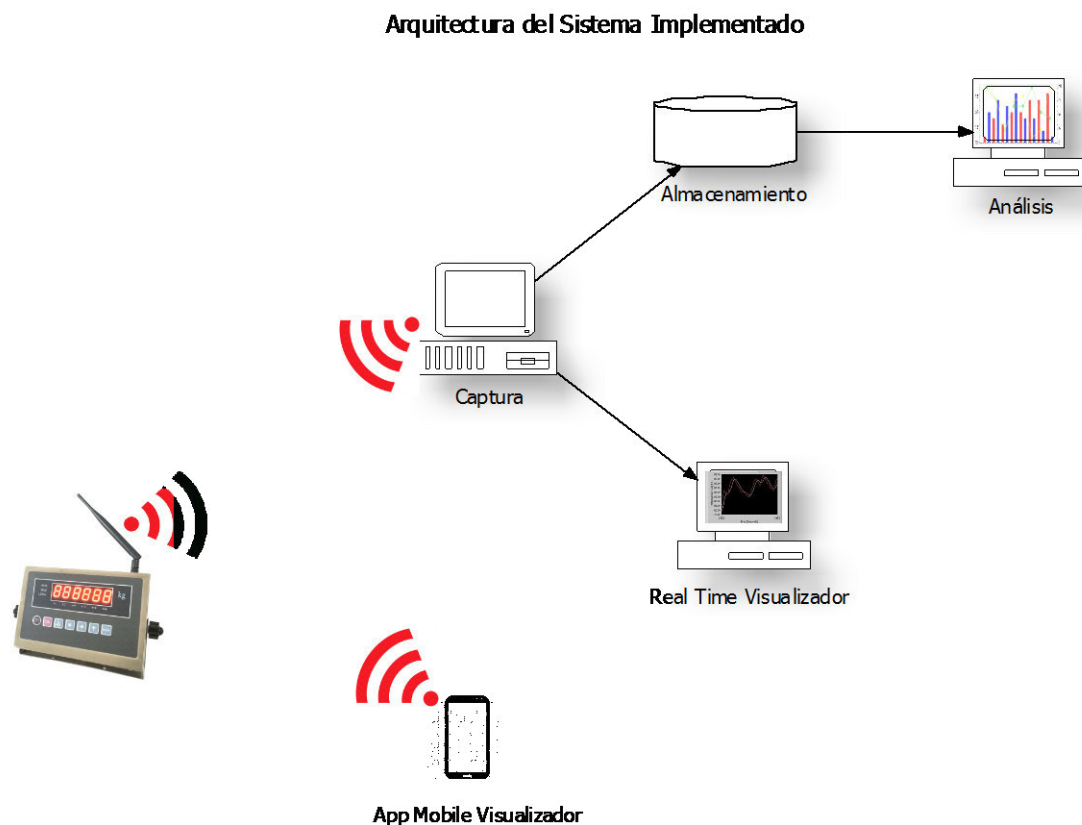


Figura 12: Diagrama de Bloques con la Arquitectura del Sistema Implementado

A continuación, se describe cada uno de los módulos que componen el sistema con sus principales características:

- **MÓDULO DE CONVERSIÓN DE DATOS SERIAL A WIFI**

Para la etapa inicial del proceso de captura de datos se utilizó un módulo convertidor de serial a wifi, HLK-RM04, producido por la compañía Shenzhen Hi-Link Electronic. El objetivo de utilizar este módulo es convertir la data que se recibe del dispositivo de medición en formato serial a formato Ethernet, de tal manera que pueda ser transmitida a un computador remoto usando una comunicación wifi standard.

El HLK-RM04 es un módulo embebido de bajo costo que realiza la conversión UART-ETH-WIFI. El módulo implementa una interfaz serial como entrada y luego la pila de protocolos TCP/IP, por lo tanto, permite realizar la conversión desde la interfaz serial hacia las interfaces Ethernet cableada y Wifi, tal como se muestra en la Figura 13.

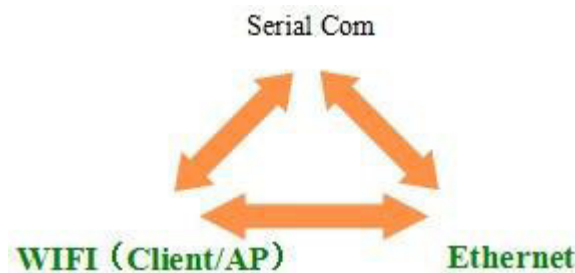


**Figura 13: Tipos de conversión proporcionadas por el módulo convertidor.**

La principal ventaja de usar este módulo es que no se requiere realizar ninguna modificación en el equipo emisor que entrega los datos, ya que la data es recibida por el módulo en formato serial, transformada y luego transmitida a través de la red por Ethernet o Wifi.

La Figura 14, muestra la vista física del módulo HLK-RM04. Acá podemos observar en ambos extremos los pines de conexión que son usados para UART, Power, Ethernet, el conector de la antena y los estándares soportados para la comunicación Wifi (b,g,n).



Figura 14: Módulo convertidor HLK-RM04

La Figura 15, describe la estructura interna del módulo convertidor, sus principales componentes y los pines de entrada y salida.

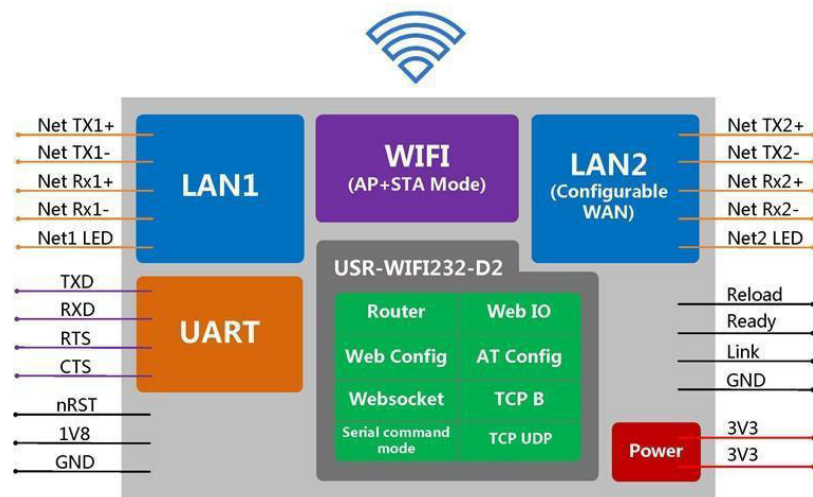


Figura 15: Diagrama de bloques del módulo convertidor

A continuación, en la Tabla 6, se presenta un Resumen de las principales especificaciones técnicas del módulo convertidor **HLK-RM0**.

### ***HLK-RM04 Technical Specifications***

**Tabla 6: Especificaciones Técnicas del Módulo Convertidor HLK-RM04**

(Fuente: [http://www.hlktech.net/product\\_detail.php?ProId=39](http://www.hlktech.net/product_detail.php?ProId=39))

|                              |                                                                                      |
|------------------------------|--------------------------------------------------------------------------------------|
| Network standard             | wireless : IEEE 802.11n、IEEE 802.11g、IEEE 802.11b<br>wired : IEEE 802.3, IEEE 802.3u |
| Wireless transmission rate   | 11n: maximum up to 150Mbps<br>11g: maximum up to 54Mbps<br>11b: maximum up to 11Mbps |
| Tracks number                | 1-14                                                                                 |
| Frequency range              | 2.4-2.4835G                                                                          |
| Emission power               | 12-15DBM                                                                             |
| Interface                    | 10/100Mbps LAN/WAN multiplex interface、 interface                                    |
| <b>Antenna</b>               |                                                                                      |
| Antenna type                 | Onboard antenna / External Antenna                                                   |
| <b>Functional Parameters</b> |                                                                                      |
| WIFI work mode               | Client/AP/Router                                                                     |
| WDS Function                 | Support WDS wireless bridge connection                                               |
| Wireless security            | Wireless MAC address filtering                                                       |
|                              | Wireless security function switch                                                    |
|                              | 64/128/152 bit WEP encryption                                                        |
|                              | WPA-PSK/WPA2-PSK、 WPA/WPA2 security mechanism                                        |
| Network management           | Remote Web management                                                                |
|                              | Configuration file import and export                                                 |
|                              | WEB software upgrade                                                                 |
| <b>Serial to Network</b>     |                                                                                      |
| Maximum transmission rate    | 230400bps                                                                            |
| TCP connection               | Max connection number>20                                                             |
| UDP connection               | Max connection number>20                                                             |
| Serial baud rate             | 50~230400bps                                                                         |
| <b>Other Parameters</b>      |                                                                                      |
| Status indicator             | Status indicator                                                                     |
| Environmental standard       | Operating temperature : -20-70°C                                                     |
|                              | Operating humidity : 10%-90%RH                                                       |
|                              | Storage temperature : -40-80°C                                                       |
|                              | Storage humidity : 5%-90%RH                                                          |
| Additional properties        | Frequency bandwidth optional : 20MHz、 40MHz ,Auto                                    |

En la Figura 16, se muestra una imagen del módulo convertidor instalado dentro del instrumento de medición.

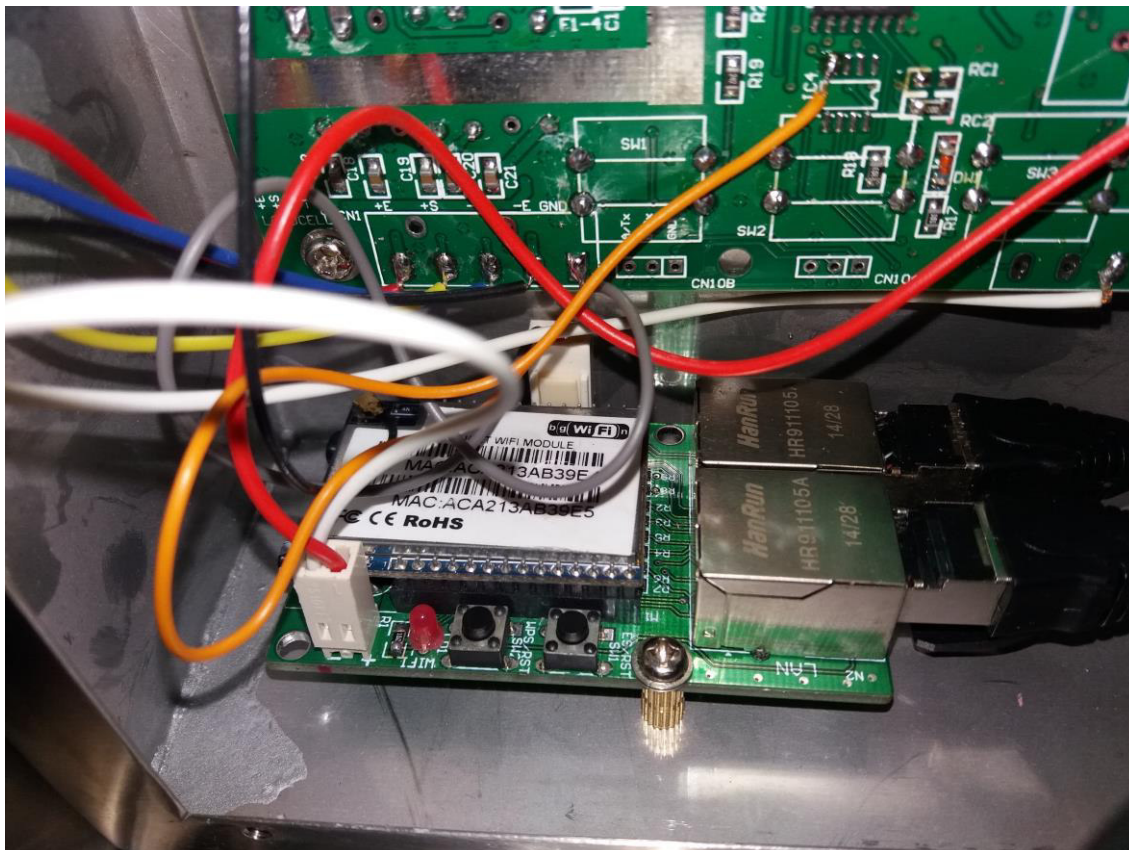


**Figura 16: Adecuación del módulo convertidor en el interior del instrumento de medición**

(Fuente: Toma fotográfica en las oficinas de Miyake SCRL, Lima-Perú)

Información adicional acerca de los modos de operación y los parámetros de configuración del módulo convertidor puede ser encontrada en el **Anexo 01** de este informe.

#### ▪ **MÓDULO DE CAPTURA DE DATOS**

Para la captura de datos provenientes del dispositivo de medición a través de la interfaz wifi, se ha utilizado una implementación Java basada en sockets.

Debido a que el dispositivo de medición es el que inicia el proceso de transmisión sólo cada vez que una operación se concreta (esto es, el indicador se pone en estado estable), éste actuará como cliente en el proceso de comunicación y el computador remoto que ejecuta el módulo de captura de datos deberá ejecutarse en modo servidor, para estar en modo escucha, atento a las solicitudes que pueden provenir

del cliente en cualquier momento. Este proceso se muestra en el diagrama esquemático de la Figura 17.



Figura 17: Diagrama esquemático del Módulo de Captura de Datos

En la Figura 18, se muestra el conjunto de parámetros que fueron necesarios setear en el módulo convertidor serial-wifi para que opere correctamente como socket cliente:

English 简体中文

HLK-RM04

- Serial2Net Settings
- Advance Settings
- Serial2Net UART 2 Settings
- Administration

NetMode: Default

SSID: CAISUN\_AP05

Password: 12345678

|                            | Current         | Updated                                   |
|----------------------------|-----------------|-------------------------------------------|
| Serial Configure:          | 9600,8,n,1      | 9600,8,n,1                                |
| Serial Framing Lenth:      | 64              | 64                                        |
| Serial Framing Timeout:    | 10 milliseconds | 10 milliseconds (< 256, 0 for no timeout) |
| Network Mode:              | client          | Client                                    |
| Remote Server Domain/IP:   | 192.168.16.234  | 192.168.16.234                            |
| Locale/Remote Port Number: | 8080            | 8080                                      |
| Network Protocol:          | tcp             | TCP                                       |
| Network Timeout:           | 0 seconds       | 0 seconds (< 256, 0 for no timeout)       |

Apply Cancel

Figura 18: Panel de configuración del módulo convertidor HLK-RM04

A continuación, en la Tabla 7, se explica de forma detallada el código en lenguaje Java, que fue necesario implementar en la PC remota para la puesta en marcha del socket server.

Acá podemos observar, que luego que el server acepta una conexión, comienza a leer los datos del stream de entrada línea por línea, hasta que el cliente nuevamente cierra la conexión, después de lo cual, los valores recibidos son almacenados en la base de datos.

**Tabla 7: Código fuente para la implementación del Socket Server, usando lenguaje Java**

```
/**
 *
 * @author Paul Quiroz
 */
public class SocketServer extends Thread {

    private static SocketServer socketServer = null;
    private Socket socket = null;
    private ServerSocket server = null;
    private SocketTestServer parent;
    private BufferedInputStream in;
    private boolean desconneted=false;
    private boolean stop = false;

    //disconnect client
    public synchronized void setDesonneted(boolean cr) {
        if(socket!=null && cr==true) {
            try.....
                socket.close();
            } catch (Exception e) {
                System.err.println("Error closing clinet : setDesonneted : "+e);
            }
        }
        desconneted=cr;
        //parent.setClientSocket(null);
    }
    //stop server
    public synchronized void setStop(boolean cr) {
        stop=cr;
        if(server!=null && cr==true) {
            try.....
                server.close();
            } catch (Exception e) {
                System.err.println("Error closing server : setStop : "+e);
            }
        }
    }

    public SocketServer(SocketTestServer parent, ServerSocket s) {
        super("SocketServer");
        this.parent = parent;
        server=s;
        setStop(false);
    }
}
```

```

        setDesonneted(false);
        start();
    }

    public static synchronized SocketServer handle(SocketTestServer parent,
        ServerSocket s) {
        if(socketServer==null)
            socketServer=new SocketServer(parent, s);
        else {
            if(socketServer.server!=null) {
                try .....
                socketServer.setDesonneted(true);
                socketServer.setStop(true);
                if(socketServer.socket!=null)
                    socketServer.socket.close();
                if(socketServer.server!=null)
                    socketServer.server.close();
            } catch (Exception e) .....
            // parent.error(e.getMessage());
        }
        socketServer.server = null;
        socketServer.socket = null;
        socketServer=new SocketServer(parent,s);
    }
    return socketServer;
}

    public void run() {
        while(!stop) {
            try.....
            socket = server.accept();
        } catch (Exception e) {
            if(!stop) {
                // parent.error(e.getMessage(),"Error acception connection");
                stop=true;
            }
            continue;
        }
        try {
            startServer();
        } catch (UnknownHostException ex) {
            Logger.getLogger(SocketServer.class.getName()).log(Level.SEVERE, null, ex);
        }
        if(socket!=null) {
            try .....
            socket.close();
        } catch (Exception e) {
            System.err.println("Erro closing client socket : "+e);
        }
        socket=null;
        // parent.setClientSocket(socket);
    }
    } //end of while
} //end of run

    private void startServer() throws UnknownHostException {
        // parent.setClientSocket(socket);
        InputStream is = null;

```

```

//    parent.append("> New Client: "+socket.getInetAddress().getHostAddress());
    try {
        is = socket.getInputStream();
        in = new BufferedInputStream(is);
    } catch (IOException e) {
//        parent.append("> Cound't open input stream on Clinet "+e.getMessage());
        setDesonneted(true);
        return;
    }

    String rec=null;
    while(true) {
        rec=null;
        try.....
            rec = readInputStream(in);//in.readLine();
        } catch (Exception e) {
            setDesonneted(true);
            if(!desonneted) {
//                parent.error(e.getMessage(),"Lost Client conection");
//                parent.append("> Server lost Client conection.");
            } else
//                parent.append("> Server closed Client conection.");
            break;
        }

        if (rec != null) {
            //rec = rec.replaceAll("\n","<LF>");
            //rec = rec.replaceAll("\r","<CR>");
            //parent.append("R: "+rec);
            appendnoNewLine(rec);
        } else {
            setDesonneted(true);
//            parent.append("> Client closed conection.");
            break;
        }
    } //end of while
} //end of startServer

private static String readInputStream(BufferedInputStream _in)
throws IOException {
    String data = "";
    int s = _in.read();
    if(s== -1)
        return null;
    data += ""+(char)s;
    int len = _in.available();
    //System.out.println("Len got : "+len);
    if(len > 0) {
        byte[] byteData = new byte[len];
        _in.read(byteData);
        data += new String(byteData);
    }
    return data;
}

public void appendnoNewLine(String msg) throws UnknownHostException {
//    messagesField.append(msg);
//    messagesField.setCaretPosition(messagesField.getText().length());
    StringBuilder builder=new StringBuilder(msg);
    String data=builder.reverse().toString();

```

```

//System.out.println(data);

//Guarda en la BD
if(data.length()<=9 && data.length()!=4){

    String dataN=data.substring(0, 8);

    float dataDB=Float.parseFloat(dataN);

    System.out.println(dataDB); //este valor se guarda en la BD

    ///////////
    FileWriter fichero=null;
    try {
        fichero=new FileWriter("E:/P/data3.txt");
    } catch (IOException ex) {
        Logger.getLogger(SocketServer.class.getName()).log(Level.SEVERE, null, ex);
    }
    PrintWriter pw=null;
    pw=new PrintWriter(fichero);
    pw.println(dataDB);
    try {
        fichero.close();
    } catch (IOException ex) {
        Logger.getLogger(SocketServer.class.getName()).log(Level.SEVERE, null, ex);
    }
    ///////////

    //////////////////////////////////////
    MongoDB mdb=new MongoDB(dataDB);
    mdb.getConection();
    mdb.saveData();
    //////////////////////////////////////

}

}
}

```

A continuación, en la Figura 19, se muestra la ejecución del módulo de captura de datos.

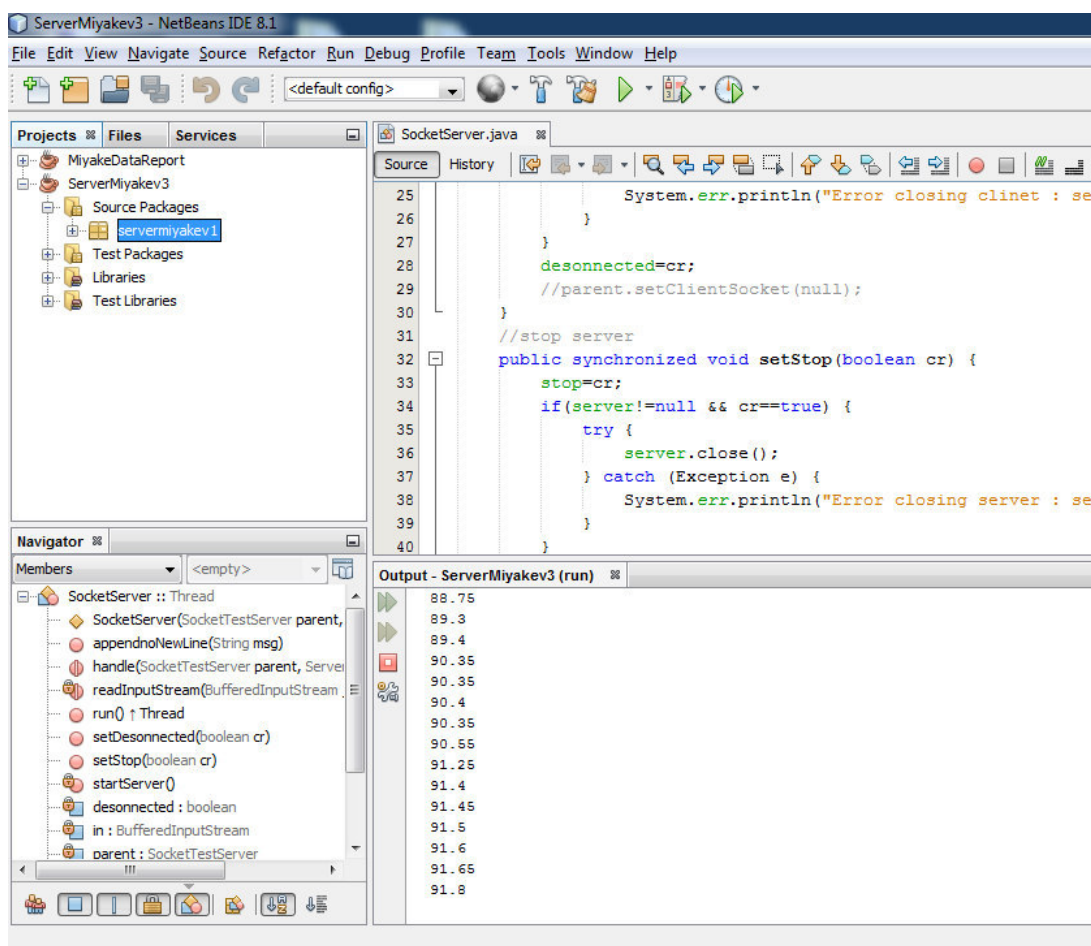


**Figura 19: Ejecución del módulo de captura de datos.**  
(Fuente: elaboración propia)

## ▪ MÓDULO DE ALMACENAMIENTO EN LA BASE DE DATOS NOSQL

Para el almacenamiento en la BD se ha utilizado la BD NoSQL MongoDB, fundamentalmente por su flexibilidad para implementar diferentes esquemas y formatos de los datos, además de su facilidad de escalamiento y alto rendimiento cuando la velocidad de transferencia de datos provenientes del emisor es muy alta. La arquitectura utilizada para implementar la Base de Datos, se muestra en la Figura 20.

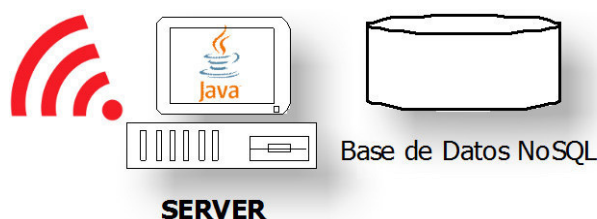
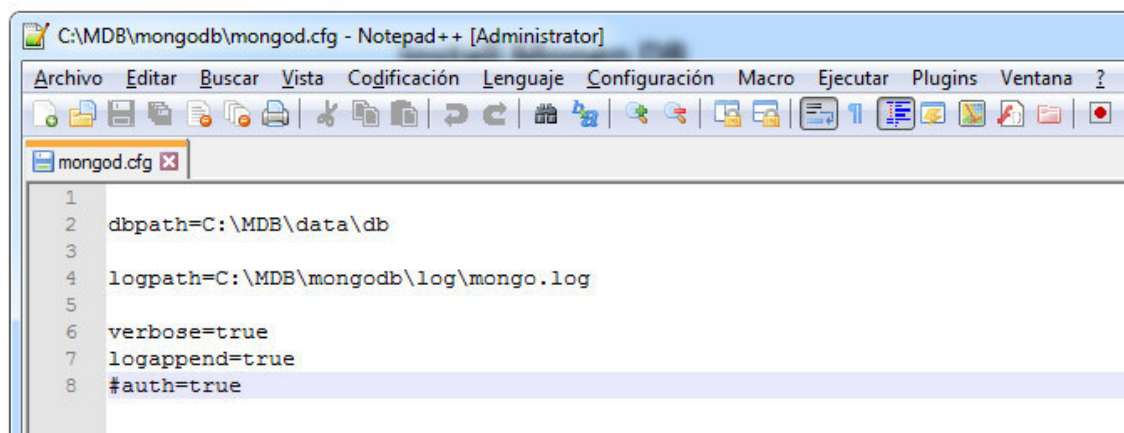


**Figura 20: Diagrama esquemático del módulo de persistencia de datos.**

Para poner en funcionamiento la Base de Datos MongoDB fue necesario realizar las siguientes configuraciones:

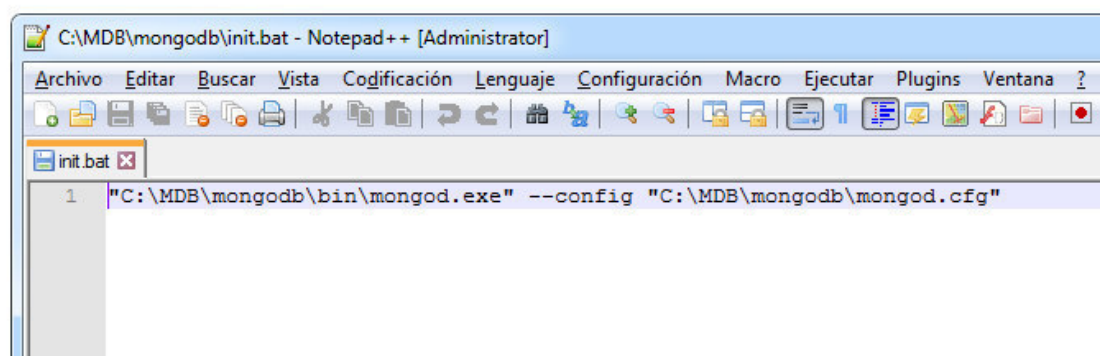
La Figura 21, muestra la creación del archivo de Configuración con los parámetros necesarios.



```
C:\MDB\mongodb\mongod.cfg - Notepad++ [Administrator]
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Macro  Ejecutar  Plugins  Ventana  ?
mongod.cfg
1
2  dbpath=C:\MDB\data\db
3
4  logpath=C:\MDB\mongodb\log\mongo.log
5
6  verbose=true
7  logappend=true
8  #auth=true
```

**Figura 21: Creación del archivo de configuración de la BD MongoDB**  
(Fuente: elaboración propia)

La Figura 22, muestra la creación del archivo \*.bat necesario para iniciar el servicio de MongoDB. En éste archivo se debe indicar la ruta completa del ejecutable de MongoDB y la ruta del archivo de configuración, después del modificador *config*.



```
C:\MDB\mongodb\init.bat - Notepad++ [Administrator]
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Macro  Ejecutar  Plugins  Ventana  ?
init.bat
1  "C:\MDB\mongodb\bin\mongod.exe" --config "C:\MDB\mongodb\mongod.cfg"
```

**Figura 22: Creación del archivo launcher para la BD MongoDB**  
(Fuente: elaboración propia)

Al hacer doble clic sobre el archivo de inicio *init.bat*, se lanza el servicio de MongoDB, tal como se muestra en la Figura 23. A partir de éste momento el servidor estará en modo “escucha” o “listening” a la espera de nuevas conexiones de los clientes.

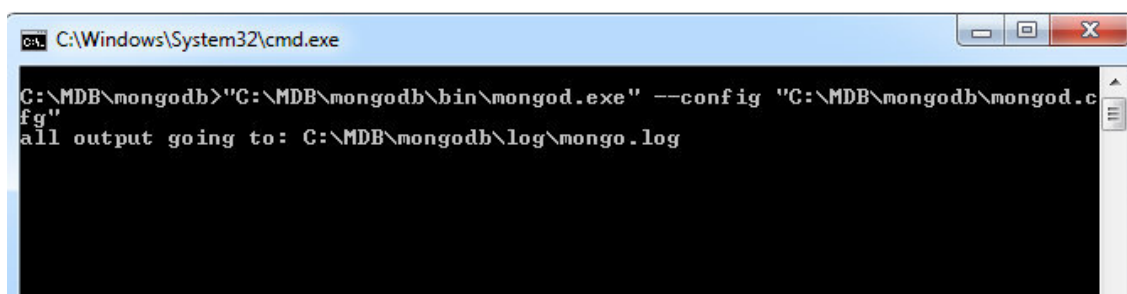


**Figura 23: Servidor MongoDB en modo listening**  
(Fuente: elaboración propia)

La Figura 24, muestra la creación del archivo de configuración *connect.bat*, necesario para iniciar un cliente MongoDB. Debemos tener en cuenta que el archivo de inicio del cliente sólo debe ser lanzado cuando el servidor (Figura 23), ya está en ejecución.

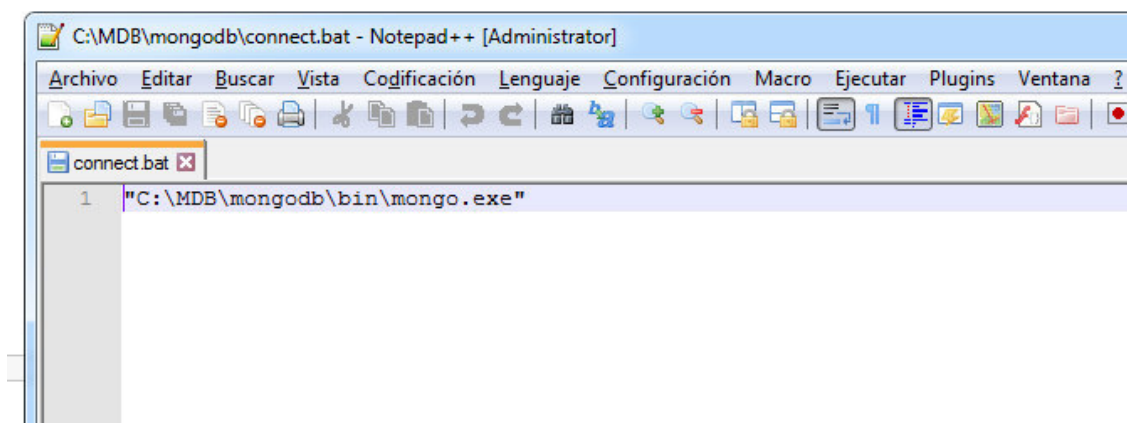
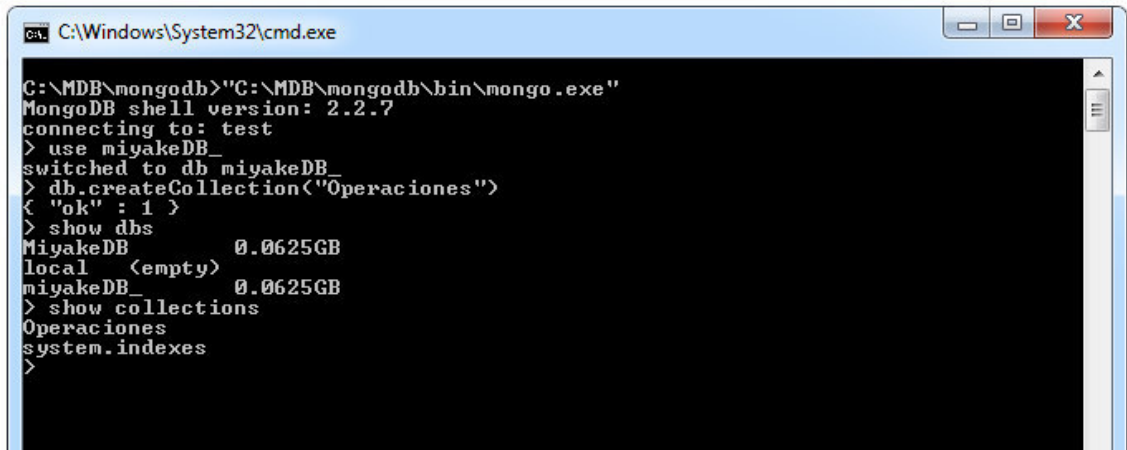


**Figura 24: Archivo de Configuración para iniciar el Cliente MongoDB**  
(Fuente: elaboración propia)

Al ejecutar el archivo *connect.bat* (Figura 24), se iniciará la consola del cliente MongoDB, tal como se muestra en la Figura 25, con lo cual podemos iniciar el proceso de creación de la BD.



```
C:\Windows\System32\cmd.exe
C:\MDB\mongodb>"C:\MDB\mongodb\bin\mongo.exe"
MongoDB shell version: 2.2.7
connecting to: test
> use miyakeDB_
switched to db miyakeDB_
> db.createCollection("Operaciones")
{ "ok" : 1 }
> show dbs
miyakeDB_      0.0625GB
local (empty)
> show collections
Operaciones
system.indexes
>
```

**Figura 25: Cliente MongoDB en ejecución.**

(Fuente: elaboración propia)

Para poblar la Base de datos fue necesario utilizar una interfaz Java, que almacene los datos provenientes de la interfaz wifi con el formato deseado de unidades, tiempo, id del dispositivo de medición.

A continuación, en la Tabla 8, se muestra el código en lenguaje Java de la interfaz necesaria para almacenar los datos en MongoDB. Aquí podemos observar, que el primer paso es indicar la ubicación del host, el puerto (en éste caso 27017), el nombre de la base de datos y la collection (tabla) a la que deseamos acceder. Luego se aplica el formato necesario a los datos y finalmente se realiza la operación de insert.

**Tabla 8: Código fuente para implementar la persistencia en MongoDB, usando lenguaje Java.**

```
/**
 *
 * @author Paul Quiroz
 */
public class MongoDB {

    float current_weigh;
    DB db;
    DBCollection table;
    BasicDBObject c;

    public MongoDB(float current_weigh){
        this.current_weigh=current_weigh;
    }

    public void getConection() throws UnknownHostException{
        MongoClient mongo = new MongoClient("localhost", 27017);
```

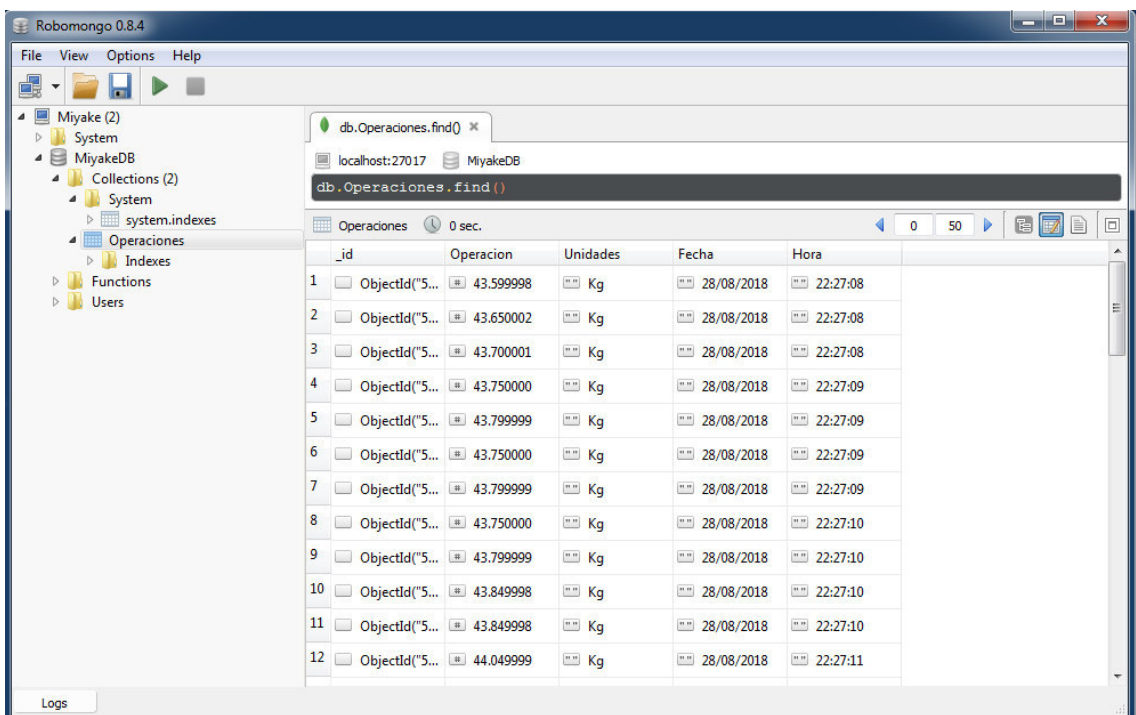
```

db = mongo.getDB("MiyakeDB");
table = db.getCollection("Operaciones");
}

public void saveData(){
    Date date=new Date();
    c = new BasicDBObject();
    c.put("Operacion", current_weigh);
    c.put("Unidades", "Kg");
    DateFormat dateFormat1=new SimpleDateFormat("dd/MM/yyyy");
    c.put("Fecha", dateFormat1.format(date));
    DateFormat dateFormat2=new SimpleDateFormat("HH:mm:ss");
    c.put("Hora", dateFormat2.format(date));
    table.insert(c);
}
}

```

A continuación, en la Figura 26, se muestran los resultados de una consulta realizada a la BD MongoDB utilizando la aplicación RoboMongo.



|    | _id              | Operacion | Unidades | Fecha      | Hora     |
|----|------------------|-----------|----------|------------|----------|
| 1  | ObjectId("5...") | 43.599998 | Kg       | 28/08/2018 | 22:27:08 |
| 2  | ObjectId("5...") | 43.650002 | Kg       | 28/08/2018 | 22:27:08 |
| 3  | ObjectId("5...") | 43.700001 | Kg       | 28/08/2018 | 22:27:08 |
| 4  | ObjectId("5...") | 43.750000 | Kg       | 28/08/2018 | 22:27:09 |
| 5  | ObjectId("5...") | 43.799999 | Kg       | 28/08/2018 | 22:27:09 |
| 6  | ObjectId("5...") | 43.750000 | Kg       | 28/08/2018 | 22:27:09 |
| 7  | ObjectId("5...") | 43.799999 | Kg       | 28/08/2018 | 22:27:09 |
| 8  | ObjectId("5...") | 43.750000 | Kg       | 28/08/2018 | 22:27:10 |
| 9  | ObjectId("5...") | 43.799999 | Kg       | 28/08/2018 | 22:27:10 |
| 10 | ObjectId("5...") | 43.849998 | Kg       | 28/08/2018 | 22:27:10 |
| 11 | ObjectId("5...") | 43.849998 | Kg       | 28/08/2018 | 22:27:10 |
| 12 | ObjectId("5...") | 44.049999 | Kg       | 28/08/2018 | 22:27:11 |

**Figura 26: Vista utilizando la herramienta RoboMongo, de los datos almacenados en la BD de Miyake.**  
(Fuente: elaboración propia)

## ▪ MÓDULO DE ANÁLISIS DE DATOS.

Para el análisis de datos en forma gráfica se utilizó la librería JfreeChart la cual nos permite generar un histograma con las frecuencias acumuladas a partir de un rango de fechas indicado. El Diagrama Esquemático del Módulo de Análisis de Datos, se muestra en la Figura 27:

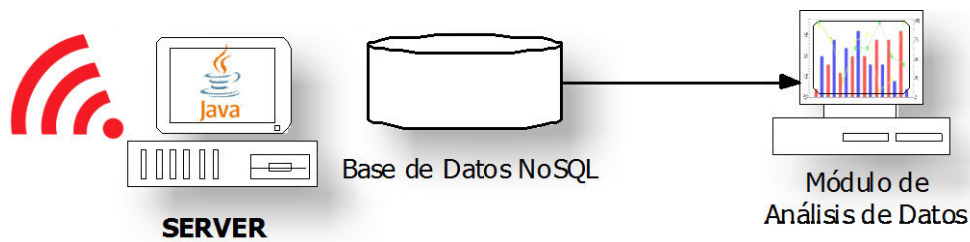


**Figura 27: Diagrama esquemático del Módulo de Análisis de Datos**

La Tabla 9, contiene el código en lenguaje Java necesario para implementar el módulo de análisis de datos. Acá podemos observar que para la elección del rango de fechas se está usando dos objetos de tipo *jDateChooser* con formato “dd/MM/yyyy” (día, mes, año), luego se realiza la consulta en la base de datos la cual devuelve un objeto collection con los resultados, se le da un formato adecuado y finalmente los objetos son pasados a una instancia de *JfreeChart* para generar el histograma de frecuencias.

**Tabla 9: Código fuente para el módulo de Análisis de datos, usando lenguaje Java.**

```
/**
 *
 * @author Paul Quiroz
 */
public class NewJFrame extends javax.swing.JFrame {

    public NewJFrame() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jPanel2 = new javax.swing.JPanel();
        jDateChooser2 = new com.toedter.calendar.JDateChooser();
        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
        jButton2 = new javax.swing.JButton();
        jDateChooser1 = new com.toedter.calendar.JDateChooser();
    }
}
```

```

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jPanel1.setBackground(new java.awt.Color(0, 0, 204));
jPanel1.setLayout(new java.awt.BorderLayout());

jPanel2.setBorder(javax.swing.BorderFactory.createEtchedBorder());

jDateChooser2.setDateFormatString("dd/MM/yyyy");

jLabel1.setText("Desde");

jLabel2.setText("Hasta");

jButton2.setText("Generar Estadísticas");
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jDateChooser1.setDateFormatString("dd/MM/yyyy");

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(18, 18, 18)
            .addComponent(jLabel1)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jDateChooser1, javax.swing.GroupLayout.PREFERRED_SIZE, 133, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jLabel2)
            .addGap(12, 12, 12)
            .addComponent(jDateChooser2, javax.swing.GroupLayout.PREFERRED_SIZE, 136, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jButton2)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addComponent(jDateChooser1)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jDateChooser2)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
            .addComponent(jButton2)
            .addContainerGap())
);
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(18, 18, 18)
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel1)
                .addComponent(jDateChooser1)
                .addComponent(jLabel2)
                .addComponent(jDateChooser2))
            .addGap(18, 18, 18)
            .addComponent(jButton2)
            .addContainerGap())
);

```



```

/////DATE
Date jdc3;
jdc3=jdc1;
DateFormat dateFormat3=new SimpleDateFormat("dd/MM/yyyy");
String f3=dateFormat3.format(jdc3);

DefaultCategoryDataset bcd=new DefaultCategoryDataset();

int k=0;
while(f3.equals(f2)!=true){
double suma=0;
BasicDBObject q = new BasicDBObject();
q.put("Fecha", f3);
DBCursor cur=table.find(q);
while(cur.hasNext()){
double op=(double) cur.next().get("Operacion");
suma=suma+op;
}

bcd.setValue(suma,"Peso "+k,f3);

jdc3.setDate((jdc3.getDate()+1);
f3=dateFormat3.format(jdc3);
k++;
}

/////////

JFreeChart bc=ChartFactory.createBarChart("Consolidado de Operaciones",
"Fecha", "Total Kilogramos", bcd, PlotOrientation.VERTICAL, false, true, false);

CategoryPlot bcp =bc.getCategoryPlot();

bcp.setRangeGridlinePaint(Color.ORANGE);

ChartPanel barPanel=new ChartPanel(bc);

////////////////////////////////////

jPanel1.removeAll();
jPanel1.add(barPanel, BorderLayout.CENTER);
jPanel1.validate();
} catch (UnknownHostException ex) {
Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);
}

}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
/* Set the Nimbus look and feel */
//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional)

```

```

">
    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
    *
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new NewJFrame().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton jButton2;
public static com.toedter.calendar.JDateChooser jDateChooser1;
public static com.toedter.calendar.JDateChooser jDateChooser2;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
public static javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
// End of variables declaration
}

```

A continuación, la Figura 28, muestra un ejemplo de un gráfico generado por el módulo de análisis de datos, para las operaciones realizadas en un periodo de tres días.

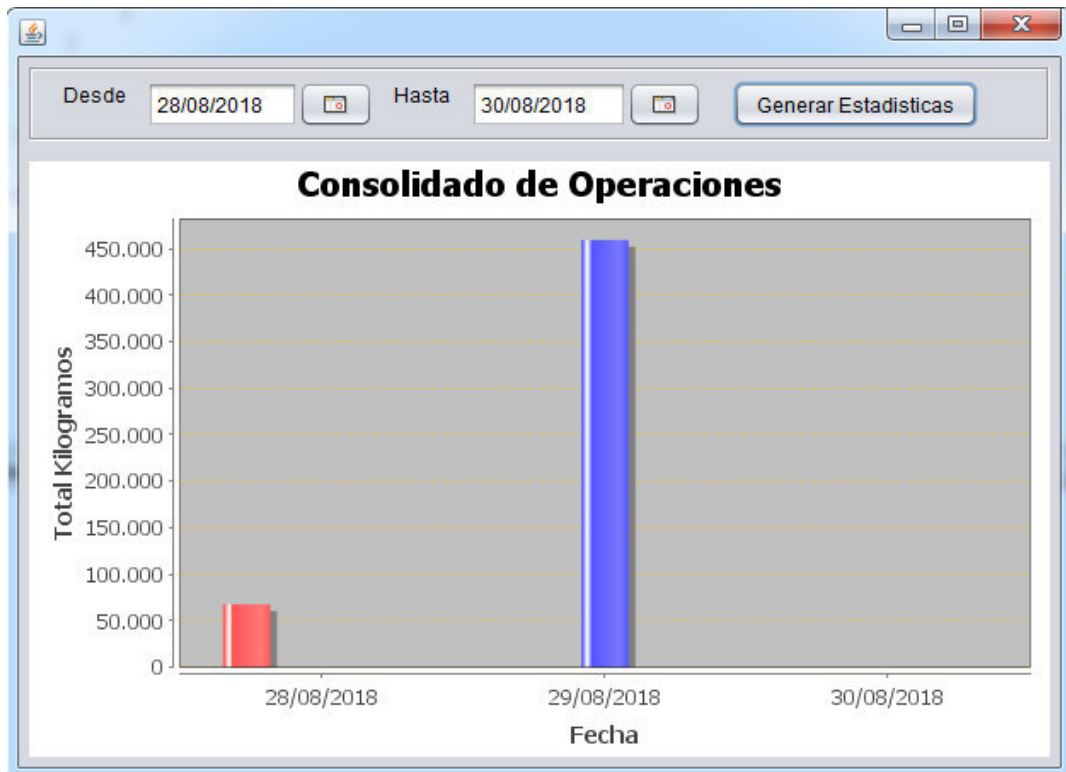


**Figura 28: Ejemplo de Histograma de Operaciones generado desde la BD Mongo DB.**

Tal como se muestra en el consolidado de operaciones de tres días consecutivos, el día 29 se alcanzó el valor pico de operaciones con un total de 450 Kg de producto, sin embargo, el día anterior, el 28, el valor estuvo muy por debajo de éste valor pico y el día 30 no se realizó ninguna operación.

#### ▪ **MÓDULO DE VISUALIZACIÓN DE OPERACIONES EN TIEMPO REAL**

El módulo de visualización de operaciones en tiempo real, tiene como objetivo mostrar el valor instantáneo de cada operación realizada y el valor histórico de las operaciones previas. El Diagrama Esquemático de éste módulo se muestra en la Figura 29.



**Figura 29: Diagrama Esquemático del módulo de Visualización o Monitoreo**

Para la implementación de éste módulo se utilizó LabView, por ser una plataforma muy potente para la visualización de datos en tiempo real. El valor instantáneo de las operaciones se muestra en formato digital y analógico, con el fin de facilitar la lectura de una manera más intuitiva. Así mismo se muestra la evolución de los valores de las operaciones realizadas a través del día con el objetivo de proporcionar al cliente una herramienta que ayude en la planificación de recursos, proveedores de materias primas, programación de mantenimientos.

La Figura 30, muestra una captura del programa en lenguaje G, que implementa esta funcionalidad.

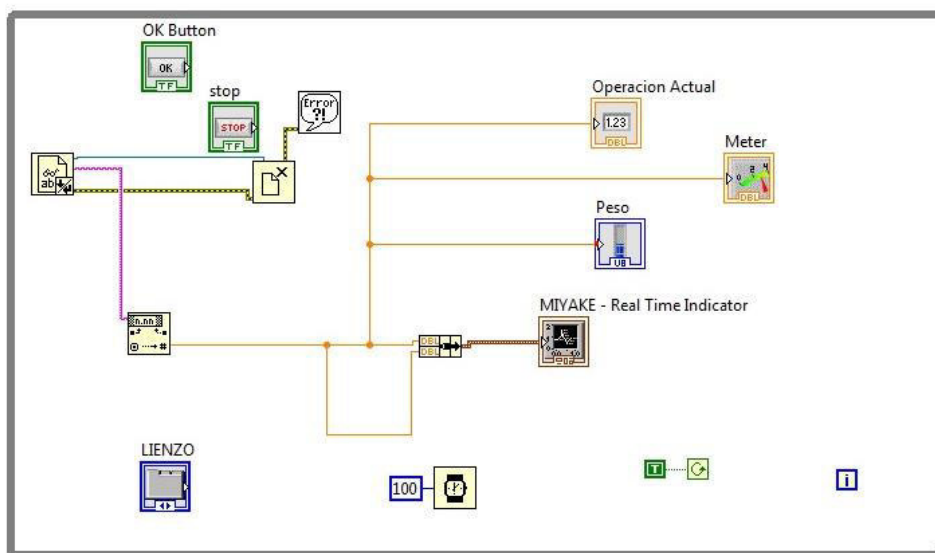


**Figura 30: Código en lenguaje G, del módulo de Visualización.**

La Figura 31, muestra el Panel Frontal del módulo de visualización en funcionamiento. Dentro de los elementos de la interfaz gráfica podemos observar un indicador numérico digital y un indicador analógico de aguja para mostrar los valores instantáneos de las mediciones, luego un fillbar, para indicar el peso actual

respecto al máximo soportado por el instrumento y un elemento time register, para monitorear la evolución histórica de las operaciones.



**Figura 31: Panel frontal del Módulo de Visualización en LabVIEW**

#### ▪ **MÓDULO DE APLICACIÓN MÓVIL DE VISUALIZACIÓN**

Además del módulo de visualización de operaciones en una PC remota, también es posible visualizar los valores instantáneos de las operaciones en un dispositivo móvil. El Diagrama Esquemático del Módulo de la Aplicación Móvil de visualización se muestra en la Figura 32:



**Figura 32: Diagrama Esquemático de la App de visualización para dispositivos móviles**

La aplicación móvil de visualización fue programada con AppInventor y tiene como objetivo además de la visualización, facilitar la auditoria de las operaciones por la alta dirección desde cualquier punto de la compañía, donde esté instalado el sistema.

A continuación, la Figura 33, muestra una captura de pantalla de la aplicación móvil en funcionamiento. Dentro de los elementos de la interfaz gráfica, podemos observar el indicador de estado de la conexión y del valor instantáneo de la medición.



**Figura 33: Ejecución de la Aplicación Móvil sobre un dispositivo Android**

Para que la aplicación móvil pueda mostrar los valores de las operaciones, es requisito que previamente la conexión wifi entre el dispositivo móvil y el instrumento de medición haya sido establecida, es decir que se muestre el mensaje de “Conexión OK”.

### **3.3 EVALUACIÓN**

Desde el punto de vista estratégico para las empresas usuarias del sistema, la solución implementada se alinea con el modelo de COBIT (Objetivos de Control para Información y Tecnologías Relacionadas), asegurando que los procesos y recursos de información y tecnología contribuyen al logro de los objetivos del negocio en un mercado cada vez más exigente, complejo y diversificado.

La implementación de la arquitectura del presente trabajo, ha sido definida por los requisitos y las restricciones empresariales identificadas en una etapa temprana por los clientes de Miyake SCRL.

Para la captura de los requisitos empresariales, fue necesario tener en cuenta:

- Objetivos de negocio, fue muy útil para decidir qué funcionalidades se deberían incluir en la solución.
- Tipo de negocio, si es un cliente final que atiende personas, o si sus clientes son otras empresas o es un proveedor de servicios.
- Identificar stakeholders, fue importante identificar para quiénes es necesario registrar, analizar o sólo visualizar la data.
- Cualidades críticas, así mismo fue necesario priorizar cuáles eran las cualidades críticas de la solución, como la transmisión inalámbrica, visualización en tiempo real, la posibilidad de ejecutarse sobre un dispositivo móvil.
- Usuarios impactados, quienes se verían influenciados de manera positiva o negativa, por la puesta en marcha de la solución.
- Niveles de Servicio, disponibilidad, fiabilidad de la aplicación, se convirtieron en elementos críticos en ciertos rubros de clientes.
- Seguridad, quiénes pueden acceder a la información, quiénes pueden generar reportes.
- Limitaciones del negocio, respecto a infraestructura de comunicaciones, equipos donde se ejecutaría la aplicación, capacitación de los operarios, etc.
- Restricciones de presupuesto, teniendo en cuenta que la solución se ofrece como un complemento a los instrumentos de medición, el costo de desarrollo, instalación y adecuación en cada cliente no debería sobrepasar el costo del equipo principal.
- Restricciones de costo de apropiación: manejo de licencias, necesidad de adecuaciones eléctricas y/o comunicaciones.
- Reglamentaciones, para efectos de auditoría, control.

### **3.3.1 EVALUACIÓN ECONÓMICA**

El desarrollo del proyecto requirió una inversión inicial de diez mil soles (S/. 10,000) para la adquisición de un equipo de cómputo (laptop), módulos de comunicación, módulo convertidor y software de desarrollo. Este costo, no incluye el panel indicador ni la plataforma de medición, los cuales fueron facilitados en calidad de préstamo del almacén de la empresa.

El costo de la instalación inicial del sistema es aproximadamente \$1200 (más IGV) por cada estación de medición en el cliente, incluye mantenimiento y soporte por un año.

Después del primer año, el cliente opcionalmente puede pagar \$450 anual, para tener derecho a mantenimientos de su Base de datos, actualizaciones y soporte.

### 3.3.2 INTERPRETACIÓN DEL VAN Y DEL TIR

La Figura 34, muestra el cálculo del VAN y TIR para el proyecto implementado.

Posteriormente se realiza el análisis de los resultados obtenidos.

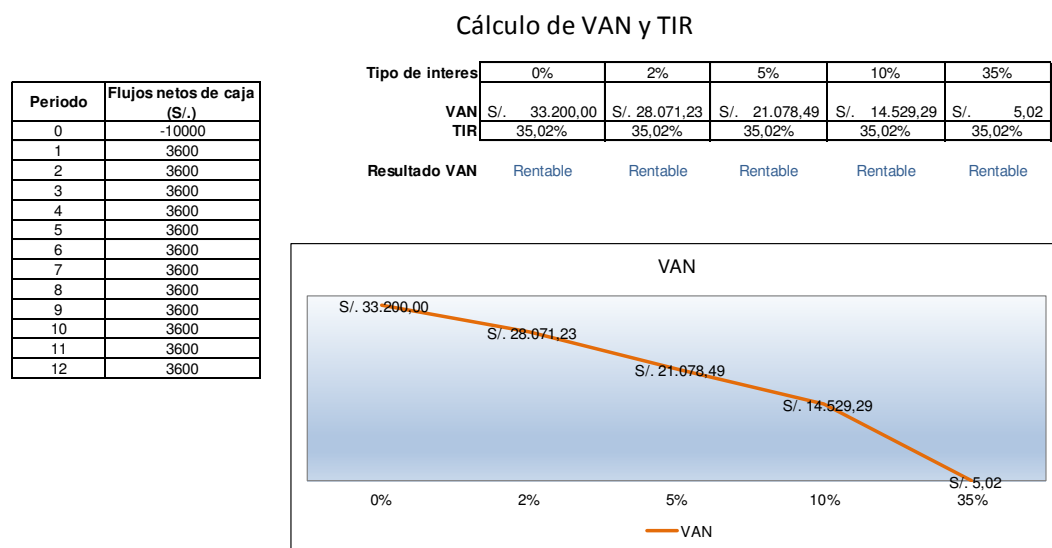


**Figura 34: Análisis de VAN y TIR del proyecto implementado.**

El Valor Actual Neto (VAN) es el valor actual de los flujos futuros de efectivo descontados a determinada tasa de interés.

Para nuestro proyecto, tal como podemos observar, para todos los tipos de interés analizados, el Valor Actual Neto (VAN), nos da resultados positivos, lo cual significa que el proyecto es viable hasta una rentabilidad esperada límite de 35%.

El TIR (**Tasa de Rendimiento Interno**) es la tasa de descuento en la que se iguala el valor de la inversión (egresos) con el valor de los flujos de tesorería futuros (ingresos).

Para nuestro proyecto podemos observar, que este punto de equilibrio se logra a un porcentaje de 35,02%, lo cual significa que nuestro proyecto será rentable hasta una tasa máxima de 35,02%, lo cual coincide con el valor máximo de 35%, obtenido para el VAN.

## **CAPÍTULO IV. REFLEXIÓN CRÍTICA DE LA EXPERIENCIA**

Este proyecto representó uno de los más grandes retos profesionales asumidos hasta ahora, ya que se tuvieron que tomar muchas decisiones de diseño críticas (lenguaje de programación, arquitectura, base de datos), aplicar conceptos y técnicas de ingeniería, investigar y aprender sobre nuevos temas, hacer muchas pruebas con equipos de alto costo, seleccionar y organizar mucha información, documentar y gestionar el proyecto durante toda su etapa de desarrollo. Esto me permitió complementar mi formación académica de la universidad y adquirir experiencia valiosa para futuros proyectos.

Del mismo modo, desarrollar y distribuir esta solución, también crea un compromiso de Miyake SCRL con sus clientes, para brindar servicios de mantenimiento periódico al sistema, lanzar actualizaciones para nuevos sistemas operativos o incorporar nuevos dispositivos.

Adicionalmente, con las herramientas disponibles actualmente, el sistema podría incorporar análisis de big data en la nube, técnicas de Inteligencia Artificial para poder hacer predicciones de operaciones, detectar patrones de actividades, programar de manera óptima los mantenimientos preventivos, detectar requerimientos de calibración para los instrumentos de medición.

## **CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES**

### **5.1 CONCLUSIONES**

El presente proyecto implementó una plataforma hardware-software para la adquisición, persistencia, visualización y análisis de datos provenientes de los sistemas de pesaje industrial que la empresa Miyake SCRL tiene instalado en sus diferentes clientes a nivel nacional.

Inicialmente se implementó la interfaz convertidora de comunicación RS-232 a Wifi, con lo cual se logró enlazar el dispositivo de medición con un computador remoto de forma inalámbrica.

Posteriormente, utilizando sockets, se implementó el módulo de captura de los datos provenientes del dispositivo de medición, bajo el modelo cliente-servidor.

Para garantizar la persistencia y disponibilidad de los datos capturados, la Base de Datos se diseñó e implementó sobre una Base de Datos Mongo DB.

Para proveer un sistema eficiente de apoyo a las decisiones de alto nivel, se implementó un sistema de análisis de datos y generación de reportes de las operaciones realizadas por rangos de fechas, utilizando JfreeChart.

El sistema de monitoreo y visualización remoto en tiempo real, se implementó mediante un indicador digital, un indicador analógico y un graficador de curvas de tendencia, programados en Labview.

Finalmente, para apoyo en los procesos de auditoría y control, se implementó una aplicación móvil para visualizar en tiempo real, desde cualquier lugar, las operaciones realizadas utilizando un teléfono móvil con Sistema Operativo Android.

El desarrollo de ésta solución, ha contribuido a que la empresa Miyake SCRL, obtenga una ventaja competitiva en su sector y a que sea reconocida como una empresa innovadora y generadora de tecnología, a tal punto que se han recibido solicitudes de desarrollo a medida de otros sistemas relacionados.

Desde el punto de vista de la estrategia empresarial de los clientes de Miyake SCRL, la solución implementada, contribuye a lograr los objetivos estratégicos del negocio; ya que permite mejorar procesos, planificar recursos y tomar mejores decisiones.

A nivel personal, el presente proyecto me ha permitido afianzar y aplicar los conocimientos teóricos adquiridos hasta ahora, así como lograr crecimiento y realización profesional en el desarrollo y gestión de proyectos innovadores de tecnología.

A nivel social y gubernamental, la implementación del presente proyecto, permite brindar una mejor calidad de atención a los usuarios finales, clientes de los negocios a los que Miyake SCRL les proporciona equipos. Así como habilita una vía para cumplir con normativas relacionadas a tributación y regulaciones ambientales, de una manera más efectiva.

## **5.2 RECOMENDACIONES**

Una posibilidad de mejora del sistema implementado, incluye la generación de nuevas versiones con integración al Sistema de Factura Electrónica, que actualmente muchos negocios están empezando a implementar por ley.

El módulo de análisis de datos, podría extender su funcionalidad de generar reportar desde datos históricos, incluyendo técnicas avanzadas de minería de datos, big data para realizar predicciones de volúmenes de producción, predicción de fallas de los equipos, identificación de patrones de comportamiento de los clientes.

### 5.3 FUENTES DE INFORMACIÓN

- Albert, M. J., Rey, U., Carlos, J., & Castro, E. B. (2000). Programación de aplicaciones distribuidas usando sockets.
- Amazon.com. (2018). BD\_NoSQL. Retrieved from <https://aws.amazon.com/es/nosql/>
- AppInventor.MIT.OficialWeb. (2018). AppInventor\_Oficial. Retrieved from <http://appinventor.mit.edu/explore/about-us.html>
- David Luis la Red Martínez. (2009). Transmisión asíncrona y síncrona., 47.
- Derek Walter, M. S. (2015). *Learning MIT App Inventor*.
- Docs.mongodb.com. (2018). Manual MongoDB. Retrieved from <https://docs.mongodb.com/>
- Foreto Saboya, N. G. (2012). Normas de Comunicación en Serie, (2), 86–94.
- Gaddis, T., & Halsey, R. (2015). *Starting out with App Inventor for Android*.
- Gilbert, D. (2008). The JFreeChart Class Library Reference Document. *Chart*, 159.
- JFreeChart.Documentation. (2018). OficialWebJFreeChart. Retrieved from <http://www.jfree.org/jfreechart/>
- MongoDB.Inc. (2016). MongoDB Documentation, 1–687.
- National.Instruments.Inc. (2013). LabVIEW, (June).
- Omar, D., & Flores, A. (2014). *Java Para Aplicaciones*.
- Roffé, L. V. S. (2009). La Interfaz Socket.
- Roncancio, H., & Cifuentes, H. (2000). Tutorial de LABVIEW. *Labview Tutorial, 1*, 20.
- Rs, P., & Bihurriet, F. (2014). Comunicación Serie.
- Sánchez, S. B. (2012). Base de datos NoSQL: MongoDB, 42–57.
- Suarez, A. A. R. (2013). Redes Industriales.
- Vila, J. (2008). Sockets en Java.
- Vlajic, N. (2016). Socket Programming in Java.

## 5.4. GLOSARIO

**Socket:** Los sockets son una tecnología que permite comunicar 2 (o más) programas ejecutándose en PCs distintas pero que están interconectadas a través de una red cableada o inalámbrica. En Java para poder crear conexiones se requiere de una dirección IP y un número de puerto para establecer la conexión. Los sockets trabajan bajo el modelo de Cliente-Servidor.

**Lenguaje G:** Es un lenguaje de programación, que representa de forma iconográfica la lógica de un programa. Es usado en programas como LabVIEW. El lenguaje G incluye todas las construcciones de un lenguaje de programación tradicional tales como tipos de datos, ciclos, eventos, variables, recursividad y programación orientada a objetos. Su principal ventaja es que la representación gráfica de un programa se traduce directamente a código de máquina para su ejecución, lo que le da gran rapidez.

**Plataforma de Pesaje:** La estructura de un equipo de medición, dónde se colocan los objetos, materias primas a evaluar.

**Indicador:** El módulo de un equipo de medición, que contiene el display indicador del valor medido y el teclado. Éste se comunica con la Plataforma de Pesaje a través de un cable apantallado.

**Celda de Carga:** Es una galga extensiométrica utilizada como elemento sensor de los equipos de medición. Usualmente un equipo de medición, incluye 1,2,4 celdas de carga dependiendo de su capacidad.

**Peso Bruto:** Es el valor indicado por el equipo de medición incluyendo el peso de la cubierta de la plataforma metálica.

**Tara:** Peso máximo que puede ser censado por las celdas de carga del instrumento de medición.

**Calibración:** Procedimiento que consiste en colocar a cero el nivel de referencia del instrumento de medición, una vez que se ha instalado todas las celdas de carga, la plataforma y se ha seteado la sensibilidad mínima.

**Terminal:** Cualquier dispositivo que sirve como receptor y transmisor de datos en una topología de comunicaciones.

## ANEXOS

Anexo 01: **HLK-RM04 Module User Manual.**



深圳市海凌科电子有限公司 **Shenzhen Hi-Link Electronic Co.,Ltd**  
[Http://www.hlktech.com](http://www.hlktech.com) Tel:0755-83575155 Fax:0755-83575189



Shenzhen Hi-Link ElectronicTechnology co., Ltd

---

# HLK-RM04 User Manual

ETHERNET

WIFI

Full Function Serial Network/Wireless Module

|          |                                |          |
|----------|--------------------------------|----------|
| <b>1</b> | <b>BRIEF INTRODUCTION.....</b> | <b>5</b> |
| <b>2</b> | <b>SUMMARIZE.....</b>          | <b>5</b> |

|            |                                               |           |
|------------|-----------------------------------------------|-----------|
| 2.1        | TECHNICAL SPECIFICATIONS .....                | 5         |
| 2.2        | HARDWARE EXPLANATION .....                    | 6         |
| 2.2.1      | <i>Mechanical Dimensions</i> .....            | 6         |
| <b>3</b>   | <b>QUICK START.....</b>                       | <b>8</b>  |
| 3.1        | RESTORE FACTORY SETTINGS .....                | 8         |
| 3.2        | CONFIGURATE NETWORK PARAMETER .....           | 8         |
| 3.3        | CONFIGURATE SERIAL NETWORK PARAMETER .....    | 9         |
| <b>4</b>   | <b>FUNCTION DESCRIPTION .....</b>             | <b>9</b>  |
| 4.1        | SERIAL TO ETHERNET .....                      | 9         |
| 4.2        | SERIAL TO WIFI CLIENT.....                    | 10        |
| 4.3        | SERIAL TO WIFI AP .....                       | 10        |
| 4.4        | DEFAULT MODE .....                            | 11        |
| <b>4.5</b> | <b>PARAMETER CONFIGURATION DIRECTION.....</b> | <b>11</b> |
| <b>5</b>   | <b>WEB CONFIGURATION .....</b>                | <b>12</b> |
| 5.1        | WEB NETWORK CONFIGURATION.....                | 12        |
| 5.1.1      | <i>Serial to Ethernet-dynamic ip</i> .....    | 13        |
| 5.1.2      | <i>Serial to Ethernet-static ip</i> .....     | 13        |
| 5.1.3      | <i>Serial to WIFI CLIENT-dynamic ip</i> ..... | 13        |
| 5.1.4      | <i>Serial to WIFI CLIENT-static ip</i> .....  | 14        |
| 5.1.5      | <i>Serial to WIFI AP</i> .....                | 14        |
| 5.2        | WEB SERIAL CONFIGURATION .....                | 14        |
| <b>6</b>   | <b>SERIAL AT COMMAND CONFIGURATION.....</b>   | <b>16</b> |
| 6.1        | ACCESS TO AT COMMAND MODE.....                | 16        |
| 6.2        | AT COMMAND .....                              | 16        |
| 6.2.1      | <i>Net mode</i> .....                         | 17        |
| 6.2.2      | <i>wifi_conf</i> .....                        | 18        |
| 6.2.3      | <i>dhcpc</i> .....                            | 18        |
| 6.2.4      | <i>net_ip</i> .....                           | 19        |
| 6.2.5      | <i>net_dns</i> .....                          | 19        |
| 6.2.6      | <i>dhcpd</i> .....                            | 19        |
| 6.2.7      | <i>dhcpd_ip</i> .....                         | 20        |
| 6.2.8      | <i>dhcpd_dns</i> .....                        | 20        |

|           |                                                        |           |
|-----------|--------------------------------------------------------|-----------|
| 6.2.9     | <i>dhcpcd_time</i> .....                               | 20        |
| 6.2.10    | <i>net_commit</i> .....                                | 20        |
| 6.2.11    | <i>out_trans</i> .....                                 | 21        |
| 6.2.12    | <i>remoteip</i> .....                                  | 21        |
| 6.2.13    | <i>remoteport</i> .....                                | 21        |
| 6.2.14    | <i>remotepro</i> .....                                 | 22        |
| 6.2.15    | <i>timeout</i> .....                                   | 22        |
| 6.2.16    | <i>mode</i> .....                                      | 22        |
| 6.2.17    | <i>uart</i> .....                                      | 23        |
| 6.2.18    | <i>uartpacklen</i> .....                               | 23        |
| 6.2.19    | <i>uartpacktimeout</i> .....                           | 23        |
| 6.2.20    | <i>save</i> .....                                      | 23        |
| 6.2.21    | <i>reconn</i> .....                                    | 24        |
| 6.2.22    | <i>ver</i> .....                                       | 24        |
| 6.3       | AT COMMAND CONTROL CODE ROUTION .....                  | 24        |
| 6.3.1     | <i>Inquiry configuration information</i> .....         | 24        |
| 6.3.2     | <i>Serial to Ethernet(Dynamic ip address)</i> .....    | 25        |
| 6.3.3     | <i>Serial to Ethernet(static ip address)</i> .....     | 26        |
| 6.3.4     | <i>Serial to wifi client(dynamic IP address)</i> ..... | 27        |
| 6.3.5     | <i>Serial to wifi client(static IP address)</i> .....  | 28        |
| 6.3.6     | <i>Serial to wifi AP</i> .....                         | 29        |
| 6.3.7     | <i>Restore factory value</i> .....                     | 30        |
| <b>7</b>  | <b>SERIAL CONFIGURATION TOOLS .....</b>                | <b>30</b> |
| 7.1       | SEARCHING THE MODULE.....                              | 32        |
| 7.2       | SET EACH PARAMETERS.....                               | 32        |
| 7.3       | SUBMIT THE CONFIGURATION.....                          | 33        |
| 7.4       | USER DATA RETENTION .....                              | 34        |
| 7.5       | INQUIRY CONFIGURATION .....                            | 34        |
| 7.6       | ACCESS TO TRANSPARENT TRANSMISSION MODE.....           | 34        |
| 7.7       | RESTORE FACTORY FACTORY VALUE SETTING.....             | 34        |
| <b>8</b>  | <b>DEVICE SEARCH TOOLS.....</b>                        | <b>35</b> |
| <b>9</b>  | <b>RESTORE FACTORY SETTINGS .....</b>                  | <b>36</b> |
| <b>10</b> | <b>FIRMWARE UPGRADE .....</b>                          | <b>37</b> |



深圳市海凌科电子有限公司 **Shenzhen Hi-Link Electronic Co.,Ltd**  
[Http://www.hlktech.com](http://www.hlktech.com) Tel:0755-83575155 Fax:0755-83575189

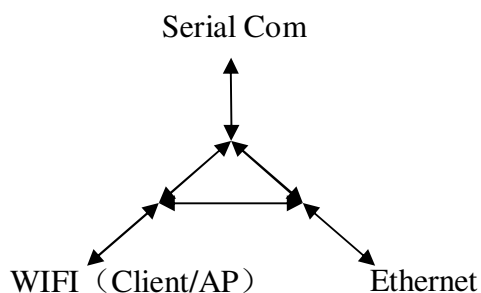
|                                                 |           |
|-------------------------------------------------|-----------|
| <b>APPENDIX A DOCUMENT REVISION RECORD.....</b> | <b>37</b> |
|-------------------------------------------------|-----------|

## 1 Brief Introduction

HLK-RM04 is a new low-cost embedded UART-ETH-WIFI module (serial port - Ethernet - Wireless network) developed by Shenzhen Hi-Link ElectronicTechnology co., Ltd

This product is an embedded module based on the universal serial interface network standard, built-in TCP / IP protocol stack, enabling the user serial port, Ethernet, wireless network (wifi) interface between the conversions.

Through the HLK-RM04 module, the traditional serial devices do not need to change any configuration; data can be transmitted through the Internet network. Provide a quick solution for the user's serial devices to transfer data via Ethernet.



Picture1.F-structure

## 2 Summarize

### 2.1 Technical Specifications

Table2-1Technical Specifications

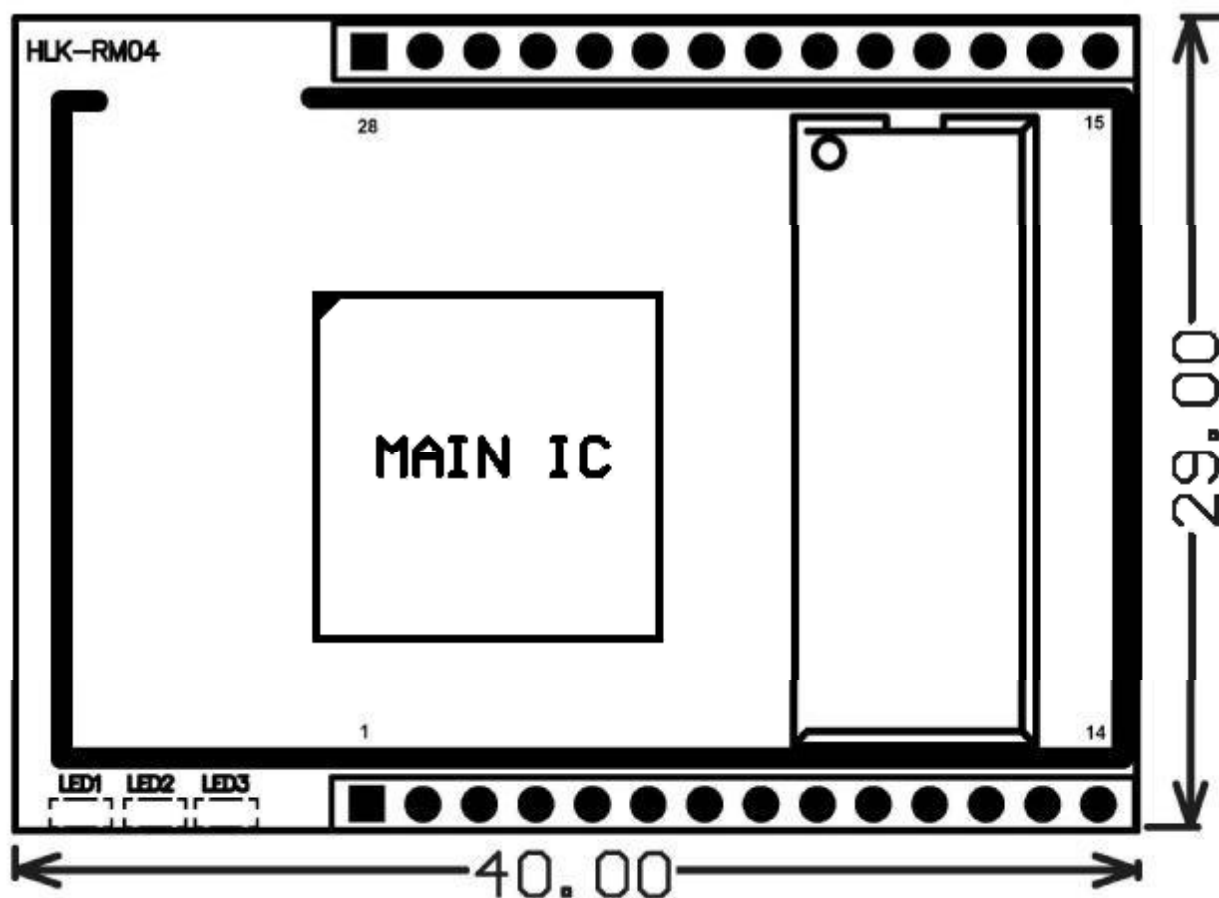
|                            |                                                       |
|----------------------------|-------------------------------------------------------|
| Network standard           | wireless: IEEE 802.11n、IEEE 802.11g、IEEE 802.11b      |
|                            | wired: IEEE 802.3、IEEE 802.3u                         |
| Wireless transmission rate | 11n: maximum up to 150Mbps                            |
|                            | 11g: maximum up to 54Mbps                             |
|                            | 11b: maximum up to 11Mbps                             |
| Tracks number              | 1-14                                                  |
| Frequency range            | 2.4-2.4835G                                           |
| Emission power             | 12-15DBM                                              |
| Interface                  | 1 个 10/100Mbps LAN/WAN multiplex interface、 interface |
| <b>Antenna</b>             |                                                       |
| Antenna type               | Onboard antenna / External Antenna                    |

|                              |                                                      |
|------------------------------|------------------------------------------------------|
| <b>Functional Parameters</b> |                                                      |
| WIFI work mode               | Client/AP/Router                                     |
| WDS Function                 | Support WDS wireless bridge connection               |
| Wireless security            | Wireless MAC address filtering                       |
|                              | Wireless security function switch                    |
|                              | 64/128/152 bit WEP encryption                        |
|                              | WPA-PSK/WPA2-PSK、WPA/WPA2 security mechanism         |
| Network management           | Remote Web management                                |
|                              | Configuration file import and export                 |
|                              | WEB software upgrade                                 |
| <b>Serial to Ethernet</b>    |                                                      |
| Maximum transmission rate    | 230400bps                                            |
| TCP connection               | Max connection number>20                             |
| UDP connection               | Max connection number>20                             |
| Serial baud rate             | 50~230400bps                                         |
| <b>Other Parameters</b>      |                                                      |
| Status indicator             | Status indicator                                     |
| Environmental standard       | Operating temperature: -20-70℃                       |
|                              | Operating humidity: 10%-90%RH (noncondensing)        |
|                              | Storage temperature: -40-80℃                         |
|                              | Storage humidity: 5%-90%RH (noncondensing)           |
| Additional properties        | Frequency bandwidth optional: 20MHz、40MHz, Automatic |

## 2.2 Hardware Explanation

### 2.2.1 Mechanical Dimensions

HLK-RM04 *Mechanical* Dimensions is shown in the following picture:



Picture2.Dimensions Unit:mm

## 2.2.2 Pins Interface

The Pin of this product as shown above is defined as follows:

Table2-2 module pin interface

| No. | Function | Direction | Explanation            |
|-----|----------|-----------|------------------------|
| 1   | VDD5V    | A         | 5 Power input          |
| 2   | GND      | GND       | Power ground           |
| 3   | GND      | G         | Serial sending         |
| 4   | 3.3V     | I         | 3.3V power output      |
| 5   | LINK1    | I/O       | Net gape 1 LED indicte |
| 6   | USB_P    |           | USB signal             |
| 7   | USB_M    |           | USB signal             |
| 8   | I2S_SD   |           | I2C DATA/GPIO          |
| 9   | I2S_CLK  |           | I2C CLK/GPIO           |
| 10  | GIOP0    | I/O       | Universal GPIO         |
| 11  | TXOP1    | I/O       | Net gape 1 TX-P        |

|    |         |           |                             |
|----|---------|-----------|-----------------------------|
| 12 | TXON1   | I/O       | Net gape 1 TX-N             |
| 13 | RXIP2   | I/O       | Net gape 2 RX-P             |
| 14 | RXIN2   | I/O       | Net gape 2 RX-N             |
| 15 | RXIN1   | I/O       | Net gape 1 RX-P             |
| 16 | RXIP1   | I/O       | Net gape 1 RX-P             |
| 17 | TXON2   | I/O       | Net gape 2 TX-N             |
| 18 | TXOP2   | I/O       | Net gape 2 TX-P             |
| 19 | RTS_N   | I         | All function serial RTS     |
| 20 | UART_RX | I         | Simple serial RX            |
| 21 | UART_TX | O         | Simple serial TX            |
| 22 | RXD     | I         | All function serial RX      |
| 23 | LINK2   | I/O       | Net gape 2 LED I/O indictor |
| 24 | CTS_N   | O         | All function serial CTS     |
| 25 | RIN     | I         | GPIO                        |
| 26 | TXD     | O         | All function serial TX      |
| 27 | 1.8V    | Power Out | Net gape 1.8V output        |
| 28 | VDD5V   | Power In  | 5V input                    |

### 3 Quick Start

#### 3.1 Restore factory settings

In order to ensure that all of configuration process is correct, bringing the module to restore the factory settings firstly. Factory mode, the module can skip this step. Above 5V (500mA) to power the module on the power, wait about 2.5 minutes for the system to start, after the start completion, pulled ES / RST pin down and make it surpass Trst, release ES / RST pin, the system will automatically restart. After rebooting, the system is already in Factory mode.

#### 3.2 Configure network parameter

Set the PC to static IP mode and then connect it with the module via Ethernet or wifi. The IP address is set to 192.168.16.100/255.255.255.0, gateway 192.168.16.254. The (wifi default ssid and the default password, see this document.) open the browser <http://192.168.16.254>, enter the web configuration page, default user name and password is admin / admin. Modify the network parameters through the web. Now, the module's IP address is 192.168.16.254. Configuration details can be seen in 5.1.

### 3.3 Configure serial network parameter

Opens the browser <http://192.168.16.254/ser2net.asp>, enter the serial-to-network web configuration page. Configure the serial-to-network parameters as needed through a web page. Configuration details can be seen in 5.2.

## 4 Function Description

The module can be divided into four major modes: default mode, serial to Ethernet, serial to WIFI CLIENT and serial to WIFI AP.

### 4.1 Serial to Ethernet

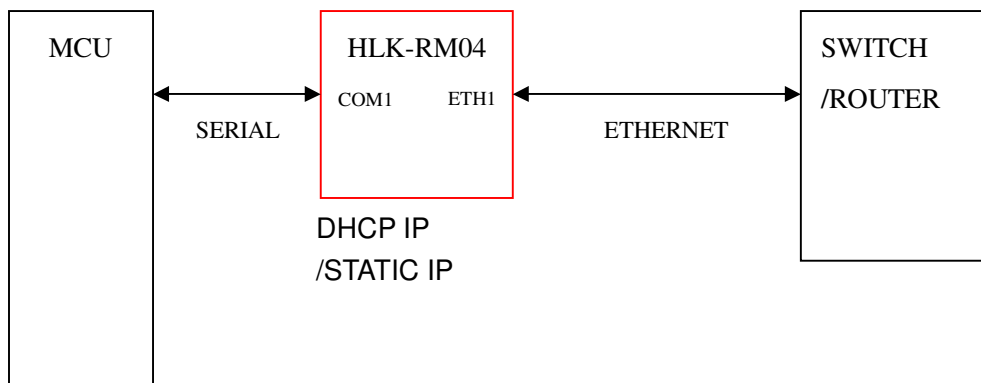


Chart3.serial to Ethernet model

In this mode, ETH1 enable, WIFI, ETH2 function close. Through the appropriate settings, the data between COM1 and ETH1 network can achieve mutual conversion.

Ethernet can be configured as dynamic IP address (DHCP), can also be configured as static IP address (STATIC).

## 4.2 Serial to WIFI CLIENT

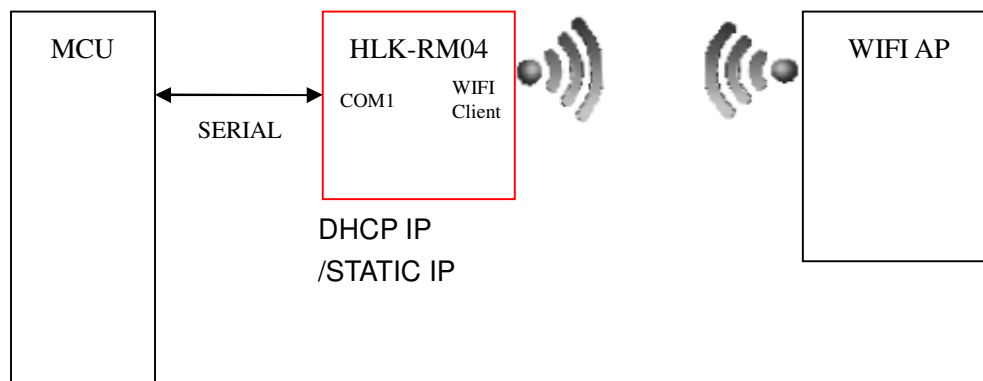


Chart 4.Serial to WIFI CLIENT model

In this mode, WIFI enable, module works in the client mode, ETH1, ETH2 function close. Through the appropriate settings, the data between COM1 and WIFI network can achieve mutual conversion.

WIFI CLIENT can be configured as dynamic IP address (DHCP), can also be configured as static IP address (STATIC).

WIFI safety: support all encryption methods at present.

## 4.3 Serial to WIFI AP

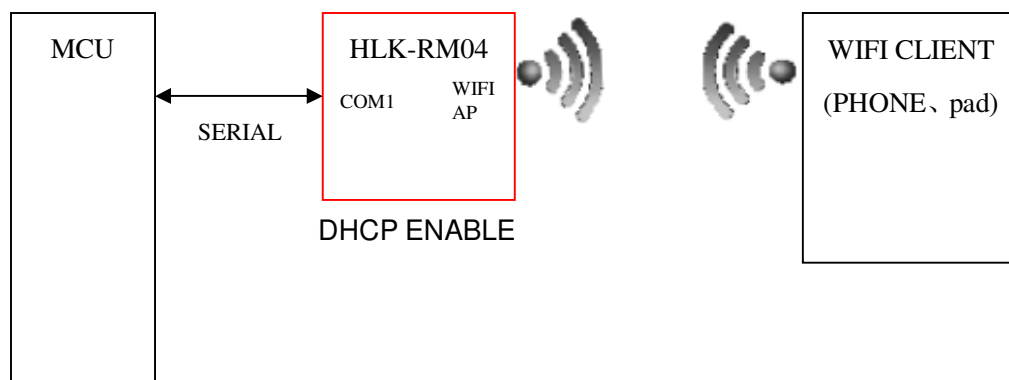


Chart 5. Serial to WIFI AP model

In this mode, WIFI enable, module works in the AP mode, ETH1, ETH2 function close.

Through the appropriate settings, the data between COM1 and WIFI network can achieve mutual conversion.

WIFI safety: support all encryption methods at present.

In this mode, WIFI device can connect with the module and become the device under WIFI LAN.

#### 4.4 Default mode

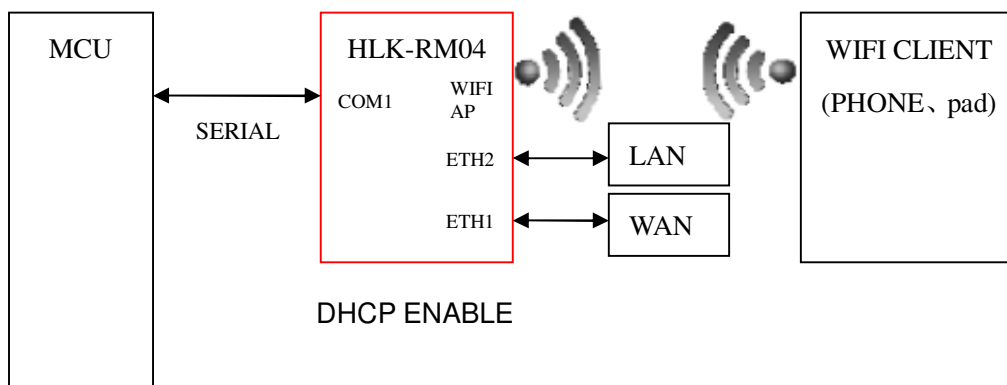


Chart 6.Default mode model

In this mode, WIFI enable, module works in the AP mode, ETH1, ETH2 function enable. ETH1 works as WAN, ETH2 works as LAN. Through the appropriate settings, the data between COM1 and network can achieve mutual conversion.

WIFI safety: support all encryption methods at present.

In this mode, WIFI device can connect with the module and become the device under WIFI LAN.

WAN default IP is dynamic IP address. LAN, WIFI for the same local area network, enabled by default DHCP server.

#### 4.5 Parameter configuration direction

The module provides two ways for the configuration parameters:

- 1.Web page;
2. Serial AT command.

Access to WEB configuration page requires the confirmation of the module's IP addresses, as well as the user name and password that authenticated by WEB.

Configuring parameters through the serial port AT command needs to make the module into the AT command mode first.

Serial configuration tool HLK-RM04\_CONFIG: Configure the module through AT command, provide a easier and convenient configuration process through the configuration combination of each parameter.

## 5 WEB configuration

|                            | Current         | Updated                                                                |
|----------------------------|-----------------|------------------------------------------------------------------------|
| Serial Configure:          | 115200,8,n,1    | <input type="text" value="115200,8,n,1"/>                              |
| Serial Framing Lenth:      | 64              | <input type="text" value="64"/>                                        |
| Serial Framing Timeout:    | 10 milliseconds | <input type="text" value="10"/> milliseconds (< 256, 0 for no timeout) |
| Network Mode:              | none            | <input type="text" value="None"/>                                      |
| Remote Server Domain/IP:   | 192.168.11.245  | <input type="text" value="192.168.11.245"/>                            |
| Locale/Remote Port Number: | 8080            | <input type="text" value="8080"/>                                      |
| Network Protocol:          | tcp             | <input type="text" value="TCP"/>                                       |
| Network Timeout:           | 0 seconds       | <input type="text" value="0"/> seconds (< 256, 0 for no timeout)       |

Chart 7.WEB configuration page

Through the correct module address (default address:<http://192.168.16.254/ser2net.asp>),you can access to the WEB configuration page.

The page can be divided into 3 areas:

- 1 Network configuration area
- 2 Serial function configuration areas
- 3 Configuration submit area

### 5.1 WEB network configuration

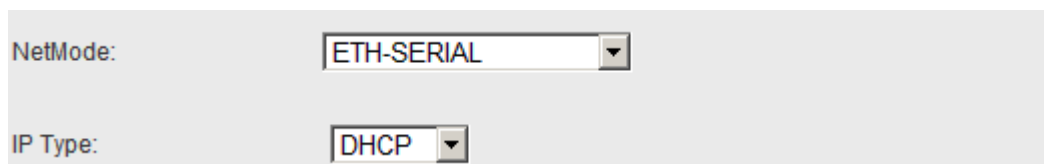
#### Net mode selection:

- Default – default work mode
- ETH-SERIAL – Serial to Ethernet
- WIFI (CLIENT)-SERIAL – serial to WIFI CLIENT
- WIFI (AP)-SERIAL) – Serial to WIFI AP

Choose different work mode, the web will show you different page.Mode configuration page

is as follows:

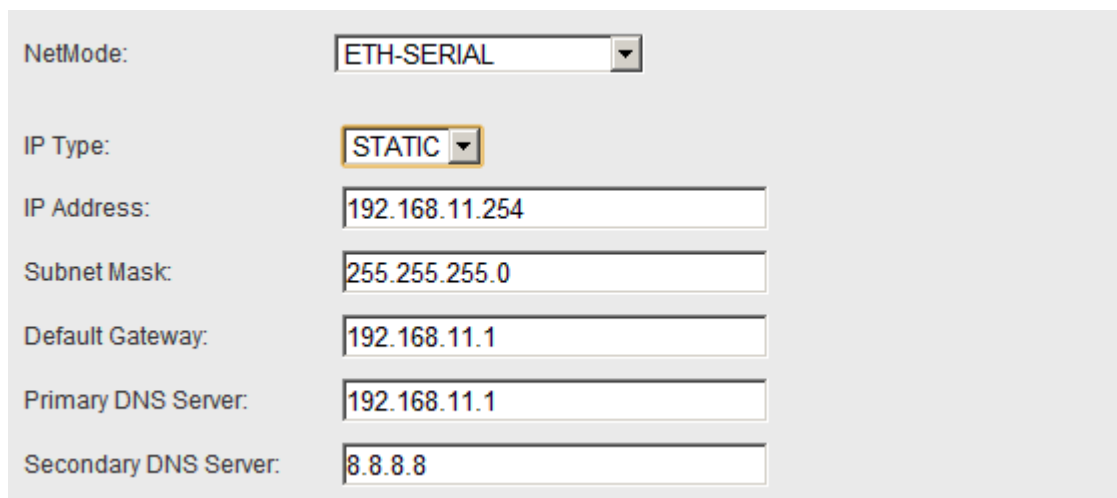
#### 5.1.1 Serial to Ethernet-dynamic ip



|          |            |
|----------|------------|
| NetMode: | ETH-SERIAL |
| IP Type: | DHCP       |

Chart 8. Serial to Ethernet-dynamic

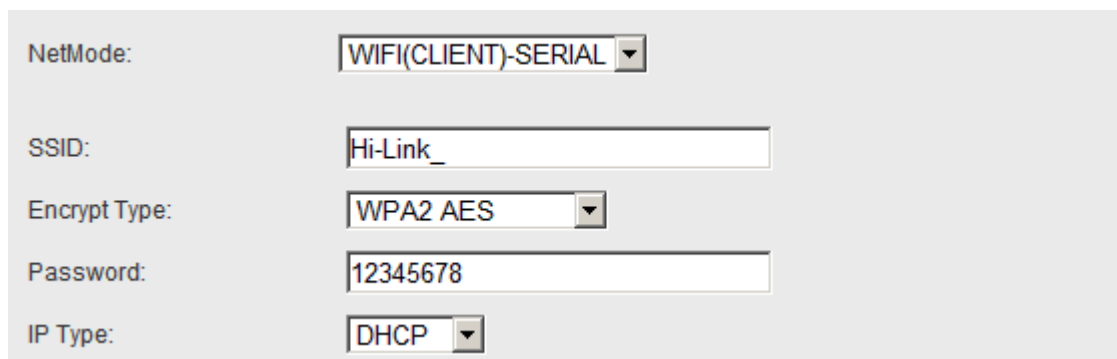
#### 5.1.2 Serial to Ethernet-static ip



|                       |                |
|-----------------------|----------------|
| NetMode:              | ETH-SERIAL     |
| IP Type:              | STATIC         |
| IP Address:           | 192.168.11.254 |
| Subnet Mask:          | 255.255.255.0  |
| Default Gateway:      | 192.168.11.1   |
| Primary DNS Server:   | 192.168.11.1   |
| Secondary DNS Server: | 8.8.8.8        |

Chart 9. Serial to Ethernet-static

#### 5.1.3 Serial to WIFI CLIENT-dynamic ip



|               |                     |
|---------------|---------------------|
| NetMode:      | WIFI(CLIENT)-SERIAL |
| SSID:         | Hi-Link_            |
| Encrypt Type: | WPA2 AES            |
| Password:     | 12345678            |
| IP Type:      | DHCP                |

Chart 10. serial to WIFI CLIENT dynamic

#### 5.1.4 Serial to WIFI CLIENT-static ip

|                       |                                                  |
|-----------------------|--------------------------------------------------|
| NetMode:              | <input type="text" value="WIFI(CLIENT)-SERIAL"/> |
| SSID:                 | <input type="text" value="Hi-Link_"/>            |
| Encrypt Type:         | <input type="text" value="WPA2 AES"/>            |
| Password:             | <input type="text" value="12345678"/>            |
| IP Type:              | <input type="text" value="STATIC"/>              |
| IP Address:           | <input type="text" value="192.168.11.254"/>      |
| Subnet Mask:          | <input type="text" value="255.255.255.0"/>       |
| Default Gateway:      | <input type="text" value="192.168.11.1"/>        |
| Primary DNS Server:   | <input type="text" value="192.168.11.1"/>        |
| Secondary DNS Server: | <input type="text" value="8.8.8.8"/>             |

Chart 11. Serial to WIFI CLIENT-static

#### 5.1.5 Serial to WIFI AP

|               |                                              |
|---------------|----------------------------------------------|
| NetMode:      | <input type="text" value="WIFI(AP)-SERIAL"/> |
| SSID:         | <input type="text" value="Hi-Link_"/>        |
| Encrypt Type: | <input type="text" value="WPA2 AES"/>        |
| Password:     | <input type="text" value="12345678"/>        |
| IP Address:   | <input type="text" value="192.168.11.254"/>  |
| Subnet Mask:  | <input type="text" value="255.255.255.0"/>   |

Chart12. Serial to WIFI AP

### 5.2 WEB serial configuration

Serial Web configuration page (ser2net.asp) is as follows:

## Serial Settings

|                                       | Current         | Updated                                                                |
|---------------------------------------|-----------------|------------------------------------------------------------------------|
| Serial Configure:                     | 115200,8,n,1    | <input type="text" value="115200,8,n,1"/>                              |
| Serial Framing Lenth:                 | 64              | <input type="text" value="64"/>                                        |
| Serial Framing Timeout:               | 10 milliseconds | <input type="text" value="10"/> milliseconds (< 256, 0 for no timeout) |
| Network Mode:                         | client          | <input type="text" value="Client"/>                                    |
| Remote Server Domain/IP:              | 192.168.11.245  | <input type="text" value="192.168.11.245"/>                            |
| Locale/Remote Port Number:            | 8080            | <input type="text" value="8080"/>                                      |
| Network Protocol:                     | udp             | <input type="text" value="UDP"/>                                       |
| Network Timeout:                      | 0 seconds       | <input type="text" value="0"/> seconds (< 256, 0 for no timeout)       |
| <input type="button" value="Submit"/> |                 |                                                                        |

**Current** shows the current configuration , **Updated** shows the current revision parameters。  
**Submit** submit the revision.

**Serial Configure:** Serial configuration.fomat: Baud rate, data bits, parity bit, stop bit.  
For example: “115200,8,n,1”.

**Serial Framing Lenth:** The Lenth of Serial Framing

**Serial Framing Timeout:** The time of Serial Framing

**Network Mode:** choose Client、Server or none。

**Remote Server Domain/IP:** Remote Server Domain/IP address

For exmpale: 192.168.11.245 or [www.hlktech.com](http://www.hlktech.com) .

**Locale/Remote Port Number:** The specified parameter is not the same under the different network modes. Client specifies the port number on the remote, Server specified local port number.

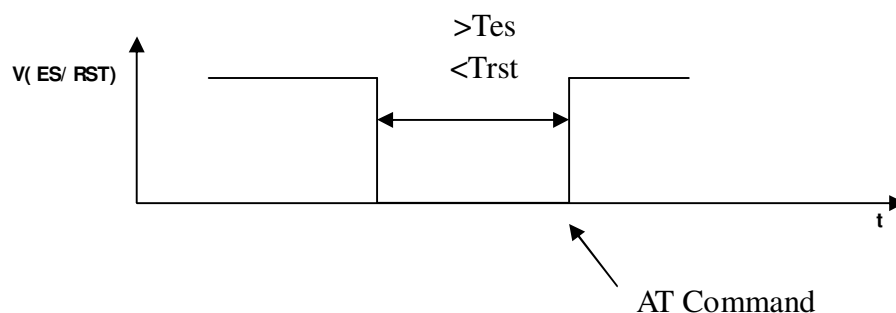
**Network Protocol:** Use tcp or udp Protocol

**Network Timeout:** Under the server network mode, no data transmission within the timeout period, the connection will be disconnected. 0 specifies never disconnected.

## 6 Serial AT command configuration

### 6.1 Access to AT command mode

Module in network fault, such as fault allocation situation will automatically exit the transparent transmission mode, enter AT instruction mode. In any condition, keep ES/RST feet low level of time but more than  $T_{es}$  and less than  $T_{rst}$ , the module will enter AT instruction mode immediately.



### 6.2 AT Command

In AT mode, you can configure the system parameters through the serial port AT instruction. Instruction format is as follows:

`At+[command]=[value]\r`

According to the different command, module will return a different return value.

For example: "at+remoteip=192.168.11.133\n" set remote ip address as 192.168.11.133.

For example: "at+remoteip=? \n" Inquiry remote ip address.

At command is as follows:

|                  |                           |
|------------------|---------------------------|
| <b>netmode</b>   | Network mode              |
| <b>wifi_conf</b> | Wifi configuration        |
| <b>dhcpc</b>     | Dhcp client configuration |
| <b>net_ip</b>    | Network ip address        |

|                        |                                          |
|------------------------|------------------------------------------|
| <b>net_dns</b>         | Network dns address                      |
| <b>dhcpd</b>           | Dhcp server configuration                |
| <b>dhcpd_ip</b>        | Dhcp server ip address                   |
| <b>dhcpd_dns</b>       | Dhcp server dns address                  |
| <b>dhcpd_time</b>      | Dhcp sever time allocation               |
| <b>net_commit</b>      | Submit network configuration             |
| <b>out_trans</b>       | Exit transparent transmission mode       |
| <b>remoteip</b>        | Remote server domain name or IP address  |
| <b>remoteport</b>      | The local or distal port number          |
| <b>remotepro</b>       | Network Protocol type                    |
| <b>timeout</b>         | Network timeout                          |
| <b>mode</b>            | Network mode                             |
| <b>uart</b>            | Serial port configuration                |
| <b>uartpacklen</b>     | Serial group frame length                |
| <b>uartpacktimeout</b> | Serial framing time                      |
| <b>save</b>            | Save the configuration and start service |
| <b>reconn</b>          | Restart services                         |
| <b>default</b>         | Restore factory value settings           |
| <b>reboot</b>          | Restart the module                       |
| <b>ver</b>             | The version of module                    |

### 6.2.1 Net mode

#### Function:

Network mode setting

#### Format:

At+netmode=<netmode>\r

#### Parameters:

Table 6-3 network mode

| value | meaning       |
|-------|---------------|
| 0     | Default setup |
| 1     | Ethernet      |
| 2     | Wifi client   |
| 3     | Wifi ap       |

### 6.2.2 wifi\_conf

**Function:**

Wireless parameter setting

**Format:**

At+wifi\_conf=<ssid>, <encrypt type>, <password> \r

**Parameters:**

ssid: Network SSID

Encrypt type: Encryption mode

Table 6-4 Encryption mode

| value        | meaning                                    |
|--------------|--------------------------------------------|
| none         | Open network                               |
| wep_open     | Wep encryption, open authentication method |
| wep          | Wep encryption, encryption authentication  |
| wpa_tkip     | wpa tkip                                   |
| wpa_aes      | wpa aes                                    |
| wpa2_tkip    | wpa2 tkip                                  |
| wpa2_aes     | wpa2 aes                                   |
| wpawpa2_tkip | wpa/wpa2 tkip                              |
| wpawpa2_aes  | wpa/wpa2 aes                               |

### 6.2.3 dhcpc

**Function:**

Dhcp client enable

**Format:**

At+dhcpc=<dhcpc>\r

**Parameters:**

Table 6-5 Dhcp client enable

| value | meaning            |
|-------|--------------------|
| 0     | Static ip address  |
| 1     | Dynamic ip address |

#### 6.2.4 net\_ip

**Function:**

Network mode setting

This parameter is not valid when Dhcp client feature is turned on.

**Format:**

At+Net\_ip=<ip>, <mask>, <gateway>\r

**Parameters:**

Ip: Ip address

Mask: Subnet mask

Gateway: Gateway Network Element

#### 6.2.5 net\_dns

**Function:**

Network mode setting

This parameter is not valid when Dhcp client feature is turned on

**Format:**

At+Net\_dns=<dns1>, <dns2>\r

**parameters:**

dns1: Major DNS address

dns2: Minor DNS address

#### 6.2.6 dhcpd

**Function:**

Dhcp server enable

This parameter is not valid when the network mode is AP.

**Format:**

At+dhcpd=<dhcpd>\r

**Parameters:**

Table 6-6 Dhcp servers enable

| value | meaning |
|-------|---------|
| 0     | close   |
| 1     | open    |

#### 6.2.7 dhcpd\_ip

**Function:**

Dhcp server IP setting

**Format:**

At+Dhcpd\_ip=<ip start>, <ip end>, <mask>, <gateway>\r

**parameters:**

Ip start: Ip started address

Ip end: Ip ended address

Mask: Subnet mask

Gateway: Gateway Network Element

#### 6.2.8 dhcpd\_dns

**Function:**

Dhcp server dns setting

**Format:**

At+Dhcpd\_dns=<dns1>, <dns2>\r

**Parameters:**

dns1: Major dns address

dns2: Minor dns address

#### 6.2.9 dhcpd\_time

**Function:**

Dhcp server time setting

**Format:**

At+Dhcpd\_time=<time >\r

**Parameters:**

time: Dhcp effective time assigned to device.

#### 6.2.10 net\_commit

**Function:**

Submit to network setting

Network configuration parameters set to be submitted by this parameter to save the entry into force.

**Format:**

At+ Net\_commit=< Net\_commit >\r

**Parameters:**

Table 6-7 submit to network setting

| value | meaning |
|-------|---------|
| 0     | invalid |
| 1     | submit  |

## 6.2.11 out\_trans

**Function:**

Exit the transparent transmission mode

**Format:**

At+out\_trans=<out\_trans>\r

**Parameters:**

Table 6-8 Exit the transparent transmission mode

| value       | meaning                                |
|-------------|----------------------------------------|
| Arbitrarily | Exit the transparent transmission mode |

## 6.2.12 remoteip

**Function:**

Remote ip or domain name setting

**Format:**

At+remoteip=< remoteip >\r

**Parameters:**

Remote server domain name or IP address

## 6.2.13 remoteport

**Function:**

Remote port setting

**Format:**

At+ remoteport=<remoteport>\r

**Parameters:**

**Remoteport:** Remote port

## 6.2.14 remotepro

**Function:**

Protocol Type setting

**Format:**

At+ remotepro=&lt;remotepro&gt;\r

**Parameters:**

Table 6-9 remotepro parameters setting

| value | meaning      |
|-------|--------------|
| None  | No protocol  |
| Tcp   | Tcp protocol |
| Udp   | Udp protocol |

## 6.2.15 timeout

**Function**

Network time-out

**Format:**

At+timeout=&lt;timeout&gt;\r

**Parameters:**

Network time-out server

Network mode, when there is not any data transfer during the time-out, the connection will be disconnected. 0 specifies never disconnected.

## 6.2.16 mode

**Function:**

The conversion mode setting

**Format:**

At+mode=&lt;mode&gt;\r

**Parameters:**

Table 6-10 mode setting

| value  | meaning      |
|--------|--------------|
| None   | No protocol  |
| Client | Tcp protocol |
| Server | Udp protocol |

#### 6.2.17 uart

**Function:**

Serial configuration setting

**Format:**

At+uart=<baud>, <data>, <parity>, <stop>\r

**parameters:**

Baud: Baud rate

Data: Data bits

Parity: Parity bit

Stop: length of stop bit

#### 6.2.18 uartpacklen

**Function:**

Serial framing length setting

**Format:**

At+uartpacklen =<uartpacklen>\r

**Parameters:**

**uartpacklen:** Serial framing length (Unit: bit) .Default value: 64.

#### 6.2.19 uartpacktimeout

**Function:**

Serial framing time setting

**Format:**

At+ uartpacktimeout=<uartpacktimeout>\r

**Parameters:**

**uartpacktimeout:** Serial framing time (unit: ms)。 Default value:10

#### 6.2.20 save

**Function:**

Submitted to serial converter configuration and restart the service.

**Format:**

At+ save=<save>\r

**Parameters:**

Table 6-11 submit to network setting

| value | meaning |
|-------|---------|
|-------|---------|

|   |         |
|---|---------|
| 0 | invalid |
| 1 | submit  |

### 6.2.21 reconn

**Function:**

Restart serial transformation service

**Format:**

At+ reconn =< reconn >\r

**Parameters:**

Table 6-12 reconn

| value       | meaning                               |
|-------------|---------------------------------------|
| arbitrarily | Restart serial transformation service |

### 6.2.22 ver

**Function:**

Inquiry the firmware version

**Format:**

At+ ver =? \r

**Parameters:**

None

## 6.3 AT command control code roution

### 6.3.1 Inquiry configuration information

Code:

```
char *query="\
at+netmode=?\r\n\
at+wifi_conf=?\r\n\
at+dhcpd=?\r\n\
at+dhcpd_ip=?\r\n\
at+dhcpd_dns=?\r\n\
at+dhcpd_time=?\r\n\
at+dhcpc=?\r\n\
at+net_ip=?\r\n\
at+net_dns=?\r\n\
at+net_wanip=?\r\n\
\
```

```
at+remoteip=?\r\n\  
at+remoteport=?\r\n\  
at+remoteport=?\r\n\  
at+timeout=?\r\n\  
at+mode=?\r\n\  
at+uart=?\r\n\  
at+uartpacklen=?\r\n\  
at+uartpacktimeout=?\r\n\  
at+ver=?\r\n\  
";
```

```
Com_send(query);
```

Run,return:

```
at+netmode=? 0  
at+wifi_conf=? Hi-Link,wpa2_aes,12345678  
at+dhcpcd=? 0  
at+dhcpcd_ip=? 192.168.14.1,192.168.15.254,255.255.254.0,192.168.15.254  
at+dhcpcd_dns=? 192.168.15.254,0.0.0.0  
at+dhcpcd_time=? 86400  
at+dhcpc=? 1  
at+net_ip=? 192.168.15.254,255.255.254.0,192.168.11.1  
at+net_dns=? 192.168.11.1,0.0.0.0  
at+net_wanip=? ,,  
at+remoteip=? 192.168.11.245  
at+remoteport=? 8080  
at+remoteport=? tcp  
at+timeout=? 0  
at+mode=? server  
at+uart=? 115200,8,n,1  
at+uartpacklen=? 64  
at+uartpacktimeout=? 10  
at+ver=? V1.39(Dec 6 2012)
```

### 6.3.2 Serial to Ethernet(Dynamic ip address)

Code:

```
char *commands_eth="\n\  
at+netmode=1\r\n\  
at+dhcpc=1\r\n\  
at+remoteip=192.168.11.245\r\n\  
at+remoteport=8080\r\n\  

```

```
at+remotepro=tcp\r\n\  
at+timeout=0\r\n\  
at+mode=server\r\n\  
at+uart=115200,8,n,1\r\n\  
at+uartpacklen=64\r\n\  
at+uartpacktimeout=10\r\n\  
at+net_commit=1\r\n\  
at+reconn=1\r\n\  
";
```

```
Com_send(commands_eth);
```

Run and return:

```
at+netmode=1 ok  
at+dhcpc=1  
at+remoteip=192.168.11.245 ok  
at+remoteport=8080 ok  
at+remotepro=tcp  
at+timeout=0 ok  
at+mode=server  
at+uart=115200,8,n,1 ok  
at+uartpacklen=64 ok  
at+uartpacktimeout=10 ok  
at+net_commit=1
```

### 6.3.3 Serial to Ethernet(static ip address)

Code:

```
char *commands_eth_static="\n\  
at+netmode=1\r\n\  
at+dhcpc=0\r\n\  
at+net_ip=192.168.11.254,255.255.255.0,192.168.11.1\r\n\  
at+net_dns=192.168.11.1,8.8.8.8\r\n\  
at+remoteip=192.168.11.245\r\n\  
at+remoteport=8080\r\n\  
at+remotepro=tcp\r\n\  
at+timeout=0\r\n\  
at+mode=server\r\n\  
at+uart=115200,8,n,1\r\n\  
at+uartpacklen=64\r\n\  
at+uartpacktimeout=10\r\n\  
at+net_commit=1\r\n\  

```

```
at+reconn=1\r\n\
";
```

```
Com_send(commands_eth_static);
```

Run and return:

```
at+netmode=1 ok
at+dhcpc=0
at+net_ip=192.168.11.254,255.255.255.0,192.168.11.1 ok
at+net_dns=192.168.11.1,8.8.8.8 ok
at+remoteip=192.168.11.245 ok
at+remoteport=8080 ok
at+remotepro=tcp
at+timeout=0 ok
at+mode=server
at+uart=115200,8,n,1 ok
at+uartpacklen=64 ok
at+uartpacktimeout=10 ok
at+net_commit=1
```

#### 6.3.4 Serial to wifi client(dynamic IP address)

Code:

```
char *commands_wifi_client="\
\
at+netmode=2\r\n\
at+wifi_conf=HI-LINK,wpa2_aes,12345678\r\n\
at+dhcpc=1\r\n\
\
at+remoteip=192.168.11.245\r\n\
at+remoteport=8080\r\n\
at+remotepro=tcp\r\n\
at+timeout=0\r\n\
at+mode=server\r\n\
at+uart=115200,8,n,1\r\n\
at+uartpacklen=64\r\n\
at+uartpacktimeout=10\r\n\
at+net_commit=1\r\n\
at+reconn=1\r\n\
";
```

```
Com_send(commands_wifi_client);
```

Run and return:

```
at+netmode=2 ok
at+wifi_conf=HI-LINK,wpa2_aes,12345678 ok
at+dhcpc=1
at+remoteip=192.168.11.245 ok
at+remoteport=8080 ok
at+remoteport=tcp
at+timeout=0 ok
at+mode=server
at+uart=115200,8,n,1 ok
at+uartpacklen=64 ok
at+uartpacktimeout=10 ok
at+net_commit=1
```

### 6.3.5 Serial to wifi client(static IP address)

Code:

```
char *commands_wifi_client_static="\
at+netmode=2\r\n\
at+wifi_conf=HI-LINK,wpa2_aes,12345678\r\n\
at+dhcpc=0\r\n\
at+net_ip=192.168.11.254,255.255.255.0,192.168.11.1\r\n\
at+net_dns=192.168.11.1,8.8.8.8\r\n\
\
at+remoteip=192.168.11.245\r\n\
at+remoteport=8080\r\n\
at+remoteport=tcp\r\n\
at+timeout=0\r\n\
at+mode=server\r\n\
at+uart=115200,8,n,1\r\n\
at+uartpacklen=64\r\n\
at+uartpacktimeout=10\r\n\
at+net_commit=1\r\n\
at+reconn=1\r\n\
";
```

```
Com_send(commands_wifi_client_static);
```

Run and return:

```
at+netmode=2 ok
at+wifi_conf=HI-LINK,wpa2_aes,12345678 ok
at+dhcpc=0
at+net_ip=192.168.11.254,255.255.255.0,192.168.11.1 ok
at+net_dns=192.168.11.1,8.8.8.8 ok
```

```
at+remoteip=192.168.11.245 ok
at+remoteport=8080 ok
at+remoteport=tcp
at+timeout=0 ok
at+mode=server
at+uart=115200,8,n,1 ok
at+uartpacklen=64 ok
at+uartpacktimeout=10 ok
at+net_commit=1
```

### 6.3.6 Serial to wifi AP

Code:

```
char *commands_wifi_ap="\
at+netmode=3\r\n\
at+wifi_conf=Hi-Link_,wpa2_aes,0000000000\r\n\
at+dhcpd=1\r\n\
at+dhcpd_ip=192.168.16.100,192.168.16.200,255.255.255.0,192.168.16.254\r\n\
at+dhcpd_dns=192.168.16.254,8.8.8.8\r\n\
at+dhcpd_time=86400\r\n\
at+net_ip=192.168.16.254,255.255.255.0,192.168.16.254\r\n\
at+net_dns=192.168.16.254,8.8.8.8\r\n\
\
at+remoteip=192.168.11.245\r\n\
at+remoteport=8080\r\n\
at+remoteport=tcp\r\n\
at+timeout=0\r\n\
at+mode=server\r\n\
at+uart=115200,8,n,1\r\n\
at+uartpacklen=64\r\n\
at+uartpacktimeout=10\r\n\
at+net_commit=1\r\n\
at+reconn=1\r\n\
";
```

```
Com_send(commands_wifi_ap);
```

Run and return:

```
at+netmode=3 ok
at+wifi_conf=Hi-Link_,wpa2_aes,0000000000 ok
at+dhcpd=1 ok
at+dhcpd_ip=192.168.16.100,192.168.16.200,255.255.255.0,192.168.16.254 ok
at+dhcpd_dns=192.168.16.254,8.8.8.8 ok
```

```
at+dhcpcd_time=86400 ok
at+net_ip=192.168.16.254,255.255.255.0,192.168.16.254 ok
at+net_dns=192.168.16.254,8.8.8.8 ok
at+remoteip=192.168.11.245 ok
at+remoteport=8080 ok
at+remoteproto=tcp
at+timeout=0 ok
at+mode=server
at+uart=115200,8,n,1 ok
at+uartpacklen=64 ok
at+uartpacktimeout=10 ok
at+net_commit=1
```

### 6.3.7 Restore factory value

Code:

```
char *commands_device_default="\
\
at+default=1\r\n\
at+reboot=1\r\n\
";

Com_send(commands_device_default);
```

Run and return:

```
at+default=1
```

After 30s, the modules start normally, all configuration parameters change to the factory configuration.

## 7 Serial configuration tools

HLK-RM04 CONFIG is a configuration tools that configure the module through the serial port. Tool interface is as follows:

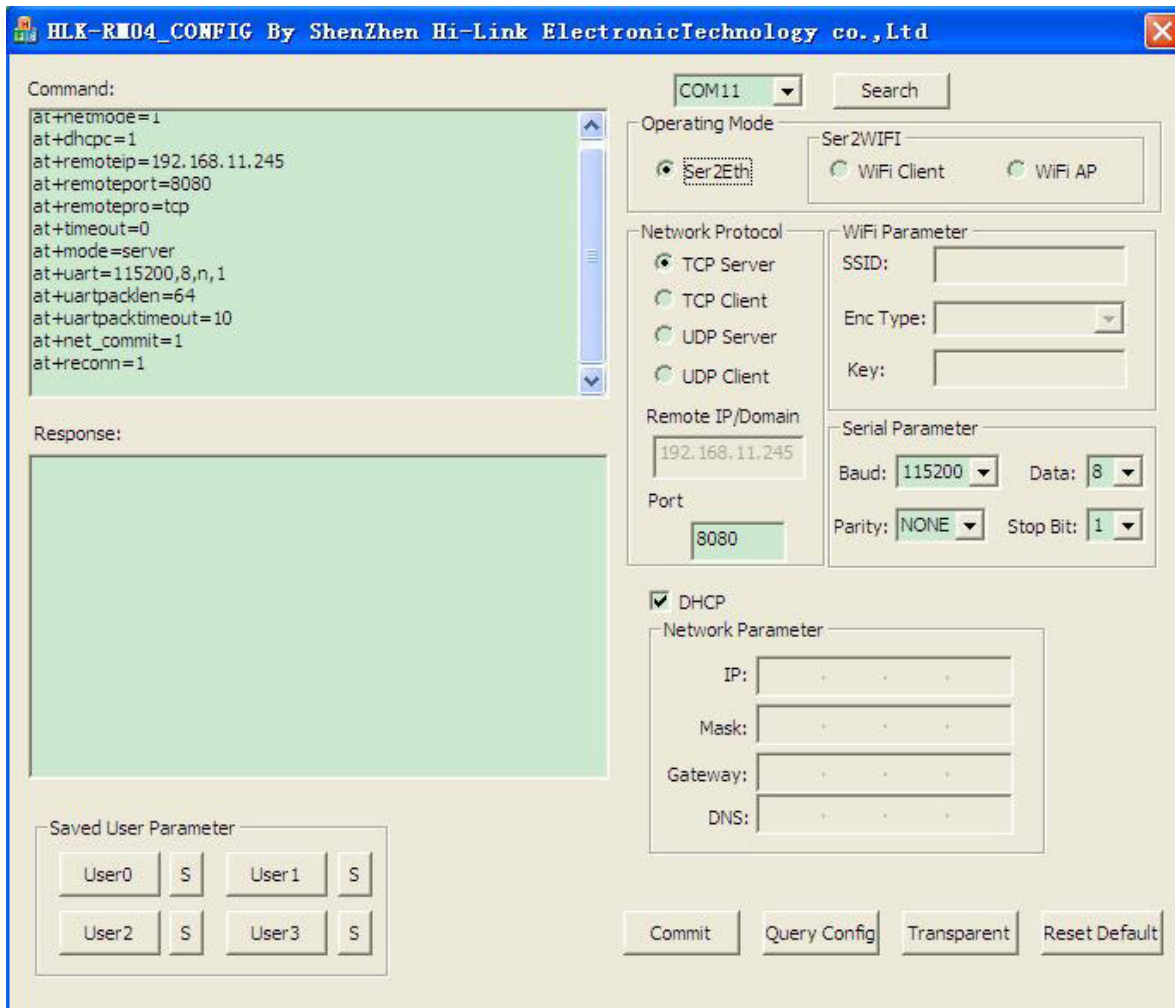


Chart 13 Configuration tools interface

**Description:**

1. 'Com 11' stands for configuration serial option
2. Search: module searching button
3. Operation mode: work mode selection
4. Wifi Parameter: wireless parameter configuration
5. Network Protocol: Network protocol selection
6. Serial Parameter: Serial parameter configuration
7. IP: Network IP address configuration
8. Commit: submit the configuration
9. Query config: Inquiry the configuration
10. Transparent: Access to transparent transmission mode
11. Reset Default: Restore the factory value setting
12. Saved User Parameter: User parameter holding area
13. Command: Ready for sent AT instruction area

#### 14. Response: AT instruction return information area

##### 7.1 Searching the Module

Through the "configure serial port choice" choose PC serial number and click on the "search module" button, the tool will use the specified serial search the module HLK - RM04, the module will be searched if it has been connected and in AT instruction mode . The module information will be found in the AT instruction return information area. Shown as below:

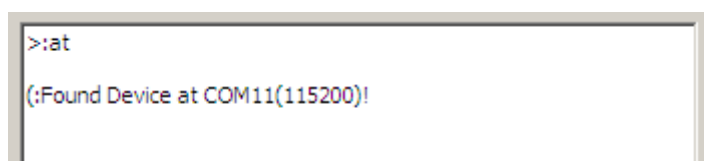


Chart 16 searching the module

At this time, The PC and module have been able to establish the normal AT command communication. All the AT interactive orders need to process based on the normal AT instruction communication.

##### 7.2 Set each Parameters

Configure the required function through the configuration items 3, 4, 5, 6, 7. The Configuration and modification information would immediately create the matching AT command in the ready for sent AT instruction area. The generated AT instruction will not send to the module at once. Shown as below:

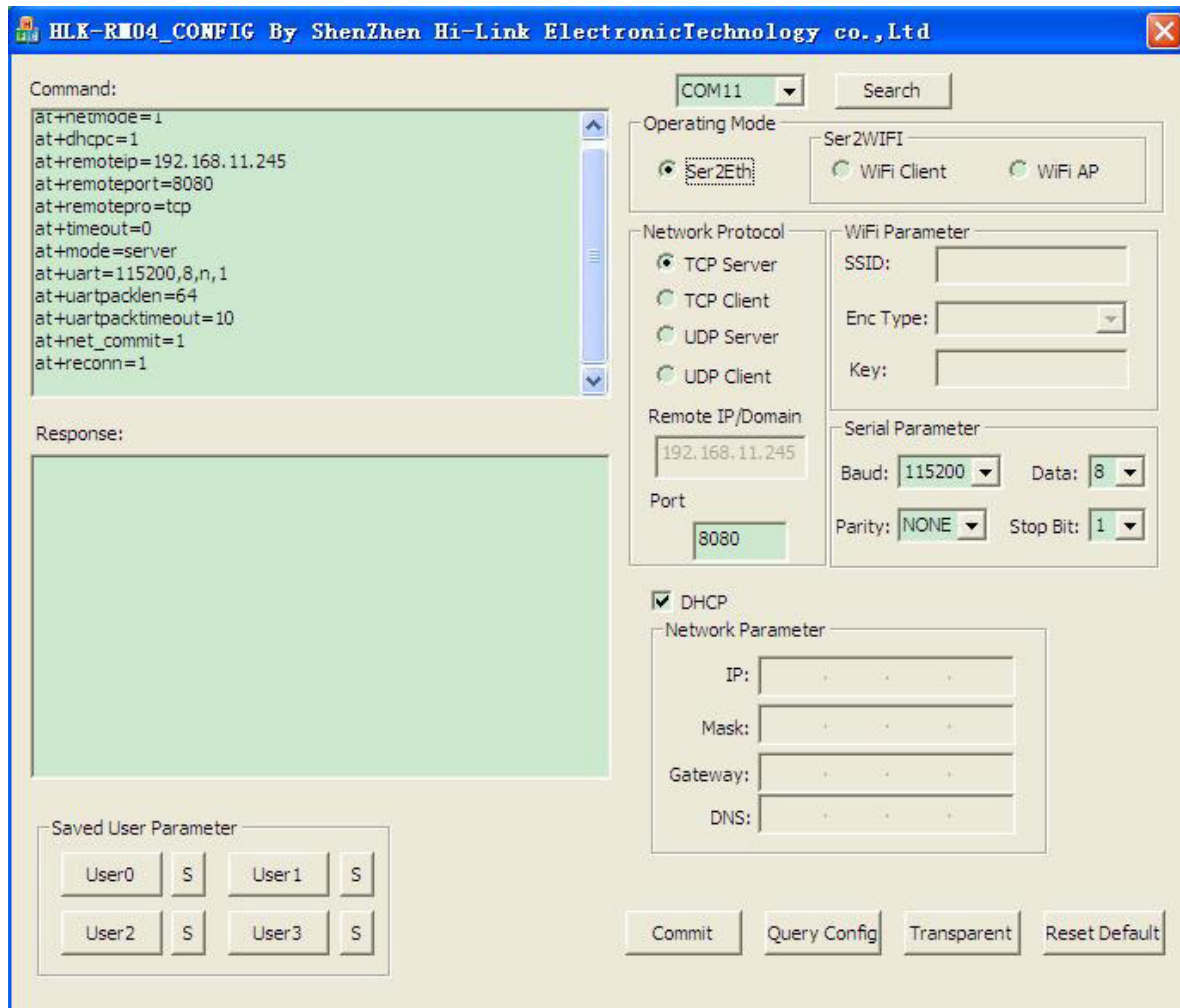


Chart 17 Serial configuration tool generates AT command

### 7.3 Submit the configuration

Click the submit configuration button, the tool will send the AT instruction in the ready for sent area to the module immediately. The information of command execution results will be shown in AT instruction return information area.



## Chart18. Serial configuration tools instruction execution

## 7.4 User Data Retention

The user parameter holding area provides parameter saving function. Through this function you can save up four sets of parameters at most, respectively, user0 user1, user2, user3. Click "S" button, it will pop up a confirmation dialog shown as below:

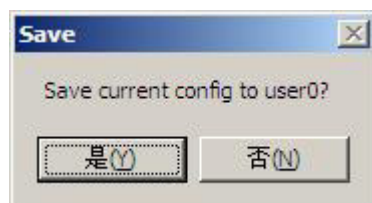


Chart 19. Pop-up Dialog of parameter saving

Click on the button "yes", the instruction in the ready for sent AT instruction area will save for user0 parameter group. After this step, when you click "user0" anytime, this parameter group can be called immediately, and covered to the ready for sent AT instruction area.

The stored user parameter will save as text file in the tools contents, file name, respectively, user0, user1, user2, user3.

## 7.5 Inquiry configuration

Click on the button 'inquiry', The tool will send a series of AT instructions immediately to the module to inquire the current configuration of the module, the result of execution will show in AT instruction return information area at once, each configuration items will make corresponding change with the return information.

## 7.6 Access to transparent transmission mode

If the module has already in the AT instruction mode, click on the button 'T/T', you can access to the transparent transmission mode at once.

## 7.7 Restore factory factory value setting

Click on the button 'reset', the tool will pop up a confirmed box shown as below:

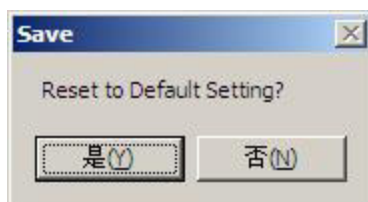


Chart 20 Pop-up box for reset default setting

Click on the button 'yes', The tools will send AT instruction immediately, after about 30 seconds, the module will access to the default state.

## 8 Device Search tools

HLK-RM04\_Discover is a search tool of network end used to search the module HLK-RM04. The interface is as follows:

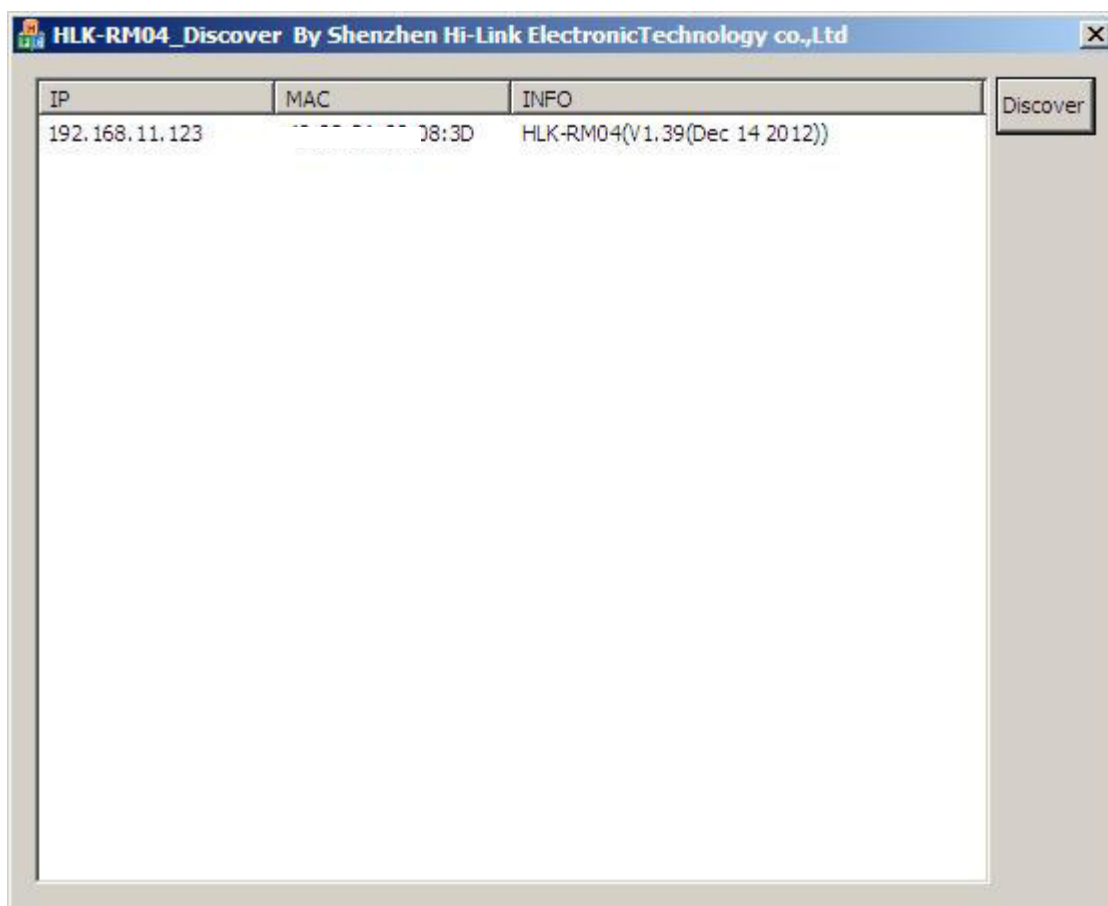


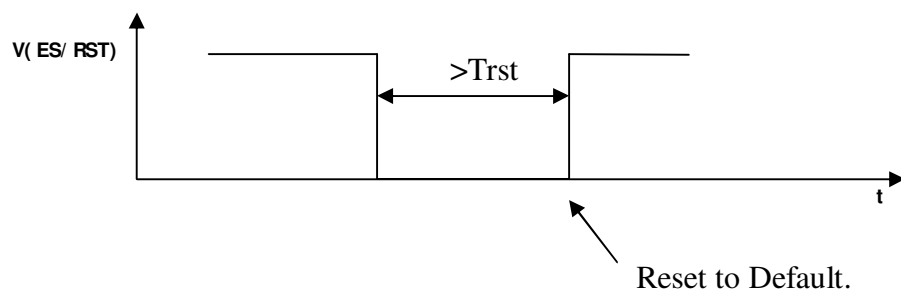
Chart 21. Device search tools

Click on the button "Discover", the tools will search all the HLK-RM04 module connected with PC in the LAN. The module being searched will show in the information box soon. The module information including: IP address, MAC address and version of it.

## 9 Restore factory Settings

Support the following ways to restore the factory settings

1. Through the Web page.
- 2 By keeping the ES/RST pin low level time greater than Trst.



Factory setting parameter values, see the following list:

|                              |                                                          |
|------------------------------|----------------------------------------------------------|
| <b>netmode</b>               | 0                                                        |
| <b>wifi_conf</b>             | Hi-Link_,wpa2_aes,12345678                               |
| <b>dhcpc</b>                 | 1                                                        |
| <b>net_ip</b>                | 192.168.11.254,255.255.255.0,192.168.11.1                |
| <b>net_dns</b>               | 192.168.11.1,8.8.8.8                                     |
| <b>dhcpcd</b>                | 1                                                        |
| <b>dhcpcd_ip</b>             | 192.168.16.100,192.168.16.200,255.255.255.0,192.168.16.1 |
| <b>dhcpcd_dns</b>            | 192.168.16.1,8.8.8.8                                     |
| <b>dhcpcd_time</b>           | 86400                                                    |
| <b>remoteip</b>              | 192.168.11.245                                           |
| <b>remoteport</b>            | 8080                                                     |
| <b>remotepro</b>             | tcp                                                      |
| <b>timeout</b>               | 0                                                        |
| <b>mode</b>                  | none                                                     |
| <b>uart</b>                  | 115200,8,n,1                                             |
| <b>uartpacklen</b>           | 64                                                       |
| <b>uartpacktimeout</b>       | 10                                                       |
| <b>IP address</b>            | 192.168.16.254                                           |
| <b>Wifi password</b>         | 12345678                                                 |
| <b>Web username/password</b> | admin/admin                                              |

|      |       |
|------|-------|
| Tes  | 100ms |
| Trst | 6s    |

## 10 Firmware upgrade

1. Restore the factory value.
2. Pc can connect with module through Ethernet, ip: 192.168.16.123/255.255.255.0. Browser visits 192.168.16.254. Username / password: admin / admin.
3. Open the following page. Select the appropriate firmware, click apply upgrades. Wait about 3 minutes. Can not cut out the upgrade process, otherwise it may cause damage to the module.



## Appendix A document revision record

| Version number | Revision range | Date      |
|----------------|----------------|-----------|
| 1.00           | Draft version  | 2012-9-10 |
|                |                |           |
|                |                |           |