

**PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO – CHILE  
ESCUELA DE INGENIERÍA ELÉCTRICA**

**ESTUDIO PARA EL CONTROL DE ACTUADORES ELECTROMECAÑICOS  
VÍA ETHERNET POR SISTEMAS EMBEBIDOS Y TECNOLOGIA NFC**

**Maycol Fabián Arrué González**

**INFORME FINAL DEL PROYECTO  
PRESENTADO EN CUMPLIMIENTO DE  
LOS REQUISITOS PARA OPTAR AL  
TÍTULO PROFESIONAL DE  
INGENIERO ELECTRÓNICO**

**JUNIO DE 2017**

**ESTUDIO PARA EL CONTROL DE ACTUADORES  
ELECTROMECA'NICOS V'IA ETHERNET POR SISTEMAS EMBEBIDOS Y  
TECNOLOGIA NFC**

**INFORME FINAL**

Presentado en cumplimiento de los requisitos  
para optar al t'itulo profesional de  
**INGENIERO ELECTR'ONICO**  
otorgado por la  
**ESCUELA DE INGENIER'IA EL'ECTRICA**  
de la  
**PONTIFICIA UNIVERSIDAD CAT'OLICA DE VALPARA'ISO**

Maycol Fabi'an Arru' e Gonz'alez

Profesor Gu'ia:  
Profesor Correferente

Sr.Joaqu' in Astorga Cornejo  
Sr.David Velasco L'opez

JUNIO DE 2017

## ACTA DE APROBACIÓN

La Comisión Calificadora designada por la Escuela de Ingeniería Eléctrica ha aprobado el texto del Informe Final de Proyecto de Titulación, desarrollado entre el Primer Semestre de 2014 y el Primer Semestre de 2015 y denominado

### **ESTUDIO PARA EL CONTROL DE ACTUADORES ELECTROMECAÑICOS VÍA ETHERNET POR SISTEMAS EMBEBIDOS Y TECNOLOGÍA NFC**

Presentado por el Señor  
**Maycol Fabián Arrué González**

SR. Joaquín Astorga Cornejo  
**Profesor Guía**

SR. David Velasco López  
**Segundo Revisor**

**Valparaíso, JUNIO DE 2017**

*A mi QUERIDA y GRAN FAMILIA*

*13 de Junio de 2017*

## AGRADECIMIENTOS

A mis padres: Secos, nunca dudaron de sus hijos y siempre supieron apoyar en todo, costó mucho pero cumplieron su objetivo final, sus tres hijos Ingenieros. Grandes y únicos, María Angélica y Ramón algún día podré darles todo y más, se lo merecen.

A mis hermanos: el Enfermo y la Nerd, son lo máximo, secos y apañadores, siempre con consejos y las palabras precisas, gran parte de este proceso universitario se los debo a ustedes, se encargaron de recordarme una y otra vez sin cansancio que lo terminara, los odié infinitamente, pero valió la pena.

A mis familia, los GONZALEZ: lejos lo más grande!!!

A los amigos: todos locos por lo demás, no son muchos pero si los mejores ☺.

A DIOS

# ESTUDIO PARA EL CONTROL DE ACTUADORES ELECTROMECA'NICOS VÍA ETHERNET POR SISTEMAS EMBEBIDOS Y TECNOLOGIA NFC

MAYCOL FABIÁN ARRUE GONZÁLEZ

Profesor Guía Sr.Joaquín Astorga Cornejo

## RESUMEN

Este informe de proyecto tiene como fin el generar un control de actuadores comandados a través de una tarjeta de desarrollo ARDUINO, donde por medio de sistemas embebidos basados en ETHERNET y sistemas NFC se desea tener dicho control mediante tecnología de proximidad y de forma remota.

Para llevar a cabo este proyecto, se estudió distintas alternativas de sistemas embebidos basados en ETHERNET para interactuar con la tarjeta de desarrollo ARDUINO y permitirle a ésta ser comandada de forma remota, en la cual se analizó dos dispositivos embebidos: equipo MOXA Nport 5110 y SHIELD ETHERNET para ARDUINO. Este estudio nos permite determinar bajo un análisis detallado de características, prestaciones, beneficios y deficiencias aportadas, la mejor opción para este proyecto.

Al lograr la comunicación del sistema embebido junto a la tarjeta de desarrollo ARDUINO, ésta quedará integrada y con acceso a los beneficios que ofrece el estándar ETHERNET, lo que permite el acceso remoto y donde por medio de sketch de programación en esta tarjeta, se procede a generar los comandos para realizar el control de actuadores y desarrollar un sistema control Domótico.

Se incluye en este estudio nuevas tecnologías disponibles como el sistema de proximidad Near Field Communication (NFC), tecnología de proximidad que funciona a base de campos electromagnéticos y que hoy en día se encuentra presente en varios equipos móviles, la cual se utilizará a beneficio del proyecto, mejorando las prestaciones del sistema domótico a desarrollar.

# ÍNDICE

## INTRODUCCIÓN

CAPÍTULO 1	1
DESCRIPCIÓN DEL PROYECTO	1
1.1 Objetivos Generales	1
1.2 Objetivos Específicos	1
1.3 Descripción del Proyecto	2
CAPÍTULO 2	4
USO DE ARDUINO PARA CONTROL DE ACTUADORES ELECTROMECAÑICOS	4
2.1 ARDUINO	4
2.1.1 ARDUINO UNO	4
2.1.2 Hardware de ARDUINO UNO	4
2.1.3 Software de ARDUINO	6
2.2 Sistemas embebidos SHIELDS y ARDUINO	7
2.2.1 SHIELD ETHERNET	8
2.2.2 SHIELD de Motores	8
2.3 Actuadores electromecánicos y ARDUINO	9
2.3.1 Motores paso a paso	9
2.3.2 Servo-motor	10
2.3.3 Motores DC	10
2.3.4 Relé electromagnético	10
CAPÍTULO 3	11
ESTANDAR DE TRANSMISIÓN PARA REDES LOCALES: ETHERNET	11
3.1 Estándar ETHERNET IEEE 802.3	11
3.2 ETHERNET y ARDUINO UNO	11
3.2.1 MOXA Nport 5110	11
3.2.2 SHIELD ETHERNET para ARDUINO UNO	12
CAPÍTULO 4	14
COMUNICACIÓN DE DISPOSITIVOS BASADOS EN ETHERNET Y PLATAFORMA ARDUINO	14
4.1 MOXA NPort 5110 junto a ARDUINO UNO	14
4.1.1 Circuito adaptador para comunicación Serie entre dispositivos	14
4.2 Configuración de dispositivo MOXA NPort 5110	15
4.2.1 Modo Servidor TCP de MOXA NPort 5110	15
4.2.2 Modo servidor TCP en MOXA	16
4.2.3 Configuración de MOXA como servidor TCP	16
4.2.4 Configuración de ARDUINO UNO	19

4.3	SHIELD ETHERNET y ARDUINO UNO	20
CAPÍTULO 5		21
COMPARATIVAS ENTRE DISPOSITIVOS ETHERNET PARA EL CONTROL DE ACTUADORES A TRAVÉS DE ARDUINO		21
5.1	Configuración	21
5.2	Funcionalidad	21
5.3	Conexión	21
5.4	Simpleza	22
5.5	Resumen comparativa	22
5.6	Elección entre dispositivos comparados	23
CAPÍTULO 6		24
CONTROL DE ACTUADORES A TRAVÉS DE SHIELD ETHERNET JUNTO ARDUINO Y SISTEMA NFC		24
6.1	Diseño y programación de Sketch para control de actuadores electromecánicos	24
6.2	Sistema NEAR FIELD COMMUNICATION (NFC)	27
6.2.1	Funcionamiento	28
6.2.2	NFC Tags	29
6.3	Control NFC y ARDUINO	30
6.3.1	Funcionamiento	30
6.3.2	Aplicación para control por sistema NFC	32
6.3.3	Programación de la aplicación	32
CAPÍTULO 7		38
EVALUACIÓN ECONÓMICA		38
7.1	Definición del producto/servicio	38
7.2	Estudio de mercado	38
7.2.1	Mercado de Proveedores	38
7.2.2	Mercado de Competencia	39
7.2.3	Mercado de Consumidores	39
7.3	Análisis de Costos	39
7.3.1	Inversión inicial	40
7.3.2	Activos Fijos	40
7.3.3	Activos Nominales	40
7.3.4	Inversión de Capital de Trabajo	40
7.3.5	Financiamiento	41
7.3.6	Depreciación	42
7.3.7	Precio de Venta	43
7.4	Flujo de caja	43
CONCLUSIONES		45



APÉNDICE A

CÓDIGO DE PROGRAMACIÓN PARA CONTROL DEL ACTUADORES  
ELECTROMECAÑICOS

- A.1 Código para control de actuadores: Relés electromagnéticos A-2
- A.2 Código para control de actuadores: Motores paso a paso y Servo-motor A-4

APÉNDICE B

PRUEBAS PARA COMUNICACIÓN ENTRE DISPOSITIVOS MOXA – CIRCUITO  
ADAPTADOR – ARDUINO UNO

- B.1 Comprobación de funcionamiento del circuito adaptador B-2
- B.2 Prueba envío de datos vía TCP/IP por MOXA NPort 5110 hacia ARDUINO  
**¡Error! Marcador no definido.**

## ÍNDICE DE FIGURAS

Fig. 1-1: Esquema resumen del proyecto	3
Fig. 2-1: ARDUINO UNO.	5
Fig. 2-2: Ventana principal de software de ARDUINO.	6
Fig. 2-3: Imagen de SHIELD ETHERNET para ARDUINO.	8
Fig. 2-4: Imagen de SHIELD de Motores Adafruit para ARDUINO.	9
Fig. 3-1: Imagen de dispositivo RS232 universal MOXA NPort 5110.	12
Fig. 3-2: ARDUINO UNO y conexión con SHIELD ETHERNET.	13
Fig. 4-1: Circuito integrado MAX232N.	14
Fig. 4-2: Configuración de integrado MAX232N.	15
Fig. 4-3: Comunicación entre dispositivos.	16
Fig. 4-4: Asignación manual de IP y máscara de subred.	17
Fig. 4-5: Verificación de conexión en consola de Windows.	19
Fig. 4-6: Configuración para comunicación Serie en ARDUINO.	19
Fig. 4-7: SHIELD ETHERNET junto a ARDUINO UNO.	20
Fig. 6-1: Comunicación de Router inalámbrico y ARDUINO por cable CAT5.	24
Fig. 6-2: Comando ipconfig por medio de CMD de Windows.	25
Fig. 6-3: Código de programa para ARDUINO con SHIELD ETHERNET.	26
Fig. 6-4: Servidor de ARDUINO a través de navegador web desde Windows.	26
Fig. 6-5: Accionamiento de relé a través de ETHERNET.	27
Fig. 6-6: a) Comunicación entre dispositivos NFC Activos, b) Comunicación entre dispositivos NFC Activo y Pasivo (NFC Tags).	29
Fig. 6-7: NFC Tags tipo Pasivas.	30
Fig. 6-8: Acceso a interfaz gráfica de ARDUINO desde dispositivo móvil.	31
Fig. 6-9: Control de ARDUINO a través de sistema NFC.	32
Fig. 6-10: Configuración de Aplicación TRIGGER para control a través de sistema NFC.	33
Fig. 6-11: Configuración de Aplicación TRIGGER para acceso a interfaz gráfica de ARDUINO.	34
Fig. 6-12: Configuración para control de enchufe a través de sistema NFC.	35
Fig. 6-13: Configuración final de la aplicación TRIGGER para el control de actuadores desde ARDUINO por sistema NFC.	36
Fig. 6-14: Diagrama de bloques para el control de actuadores a través de ETHERNET.	37
Fig. 7-1: Simulación de crédito bancario.	42

## ÍNDICE DE TABLAS

Tabla 2-1: Características del hardware de ARDUINO UNO.	5
Tabla 3-1: Especificaciones Técnicas de dispositivo MOXA NPort 5110.	12
Tabla 4-1: Comandos dentro de consola de MOXA para su configuración.	18
Tabla 5-1: Comparativa resumen entre dispositivos basados en ETHERNET.	23
Tabla 7-1: Proveedores de materiales del proyecto.	39
Tabla 7-2: Activos Fijos.	40
Tabla 7-3: Activos nominales.	40
Tabla 7-4: Costos asociados.	41
Tabla 7-5: Inversión de capital.	41
Tabla 7-6: Depreciación de activos.	43
Tabla 7-7: Resumen cálculo VAN y TIR.	44

## GLOSARIO DE TÉRMINOS

<b>NFC:</b>	Near Field Communication, tecnología de proximidad por campo magnético.
<b>IEEE:</b>	Institute of Electrical and Electronics Engineers.
<b>SHIELD:</b>	Componentes diseñados para ampliar prestaciones de microcontroladores.
<b>ETHERNET:</b>	Estándar de transmisión para redes locales LAN.
<b>DOMÓTICA:</b>	Control automatizado de sistemas domésticos u oficina.
<b>SKETCH:</b>	Esquema para escribir programas en ARDUINO
<b>SRAM:</b>	Static Random Access Memory
<b>EEPROM:</b>	Electrically Erasable Programmable Read-Only Memory.
<b>TX:</b>	Transmisión de datos.
<b>RX:</b>	Recepción de datos.
<b>UART:</b>	Universal Asynchronous Receiver Transmitter.
<b>TTL:</b>	Transistor Transistor Logic.
<b>TCP:</b>	Transmission Control Protocol.
<b>IP:</b>	Internet Protocol.
<b>UDP:</b>	User Datagram Protocol
<b>DATASHEET:</b>	Hoja de datos del fabricante.

## INTRODUCCIÓN

En busca de satisfacer sus necesidades, el ser humano ha ido desarrollando nuevas tecnologías para realizar distintos procesos en su vida cotidiana, donde uno de ellos es el sistema de control, presente en gran parte de las grandes industrias para el monitoreo y control de procesos o accionamientos remotos de actuadores, donde parte fundamental de estos procesos es la integración entre los dispositivos que utilizan. De ahí viene la idea de integrar estos sistemas al uso doméstico y realizar control de acciones de forma remota por medio de la tecnología que nos ofrecen dispositivos de bajo costo como son los microcontroladores.

Las tarjetas de desarrollo ARDUINO [1] presentan una gran opción para diseñar e implementar sistemas de control a bajo costo, debido a las grandes prestaciones que poseen más su simpleza a la hora de programar, ARDUINO [1] será nuestro principal componente electrónico para el desarrollo de este proyecto, el cual consiste en realizar un control de actuadores por medio de esta tarjeta, a través de protocolos de comunicación ETHERNET [7] y llevarlo a Domótica con la inclusión de nuevas tecnologías como son los sistemas de proximidad Near Field Communication o NFC [13].

A lo largo de este proyecto, se hará un estudio detallado de dos tipos de sistemas embebidos basados en ETHERNET con el fin de que interactúen junto con la tarjeta de desarrollo ARDUINO, estos corresponden a MOXA Nport 5110 [8] y SHIELD ETHERNET [2] para ARDUINO, ambos capaces de aumentar las prestaciones de esta tarjeta y permitirle utilizar las ventajas de ETHERNET, permitiendo su acceso vía protocolos TCP/IP para su posterior control. El estudio realizado consta de características, configuración, funcionalidad, conexión y simpleza con que cuentan estos dispositivos donde se determinará la mejor opción para el proyecto. La toma de decisiones y elección de mejor opción para el correcto funcionamiento de este proyecto, el que integra la tarjeta de desarrollo ARDUINO junto con el sistema embebido basado en ETHERNET, se hará en base a una tabla comparativa donde se enfrenten los dispositivos antes mencionados.

El incluir nuevas tecnologías al proyecto para aumentar sus prestaciones, nos lleva a introducir a este estudio la integración de sistemas de proximidad NFC a través de campos electromagnéticos, tecnología disponible actualmente en varios dispositivos móviles de última generación, por medio de este estudio, nos permitirá hacer control de acciones de través de la lectura y reconocimiento de dispositivos pasivos llamados NFC Tags.

Lograr la total integración de esta tecnología por medio de su sistema de proximidad NFC permitirá realizar el control de actuadores en base a campos electromagnéticos para sistemas domóticos, demostrando que con esta tecnología podemos obtener bastantes beneficios que aún no son explotados en el mercado.

## CAPÍTULO 1

### DESCRIPCIÓN DEL PROYECTO

En el presente capítulo se presenta el tema de Proyecto “Estudio para Control de Actuadores Electromecánicos vía ETHERNET por Sistemas Embebidos y Tecnología NFC”. Se especificarán los aspectos a desarrollar para la creación del proyecto, además de las herramientas de las que disponemos para lograr el correcto funcionamiento de éste.

#### 1.1 Objetivos Generales

- Estudiar los dispositivos involucrados en el desarrollo del proyecto, para el control de actuadores por medio de tecnología NFC vía estándares de transmisión de datos ETHERNET a través de tarjeta de desarrollo ARDUINO.
- Analizar distintas alternativas de sistemas basados en estándares de transmisión de datos ETHERNET IEEE 802.3 y sistemas embebidos (SHIELD) para operar junto con tarjeta de desarrollo ARDUINO.
- Utilizar los beneficios que ofrece ETHERNET para desarrollo del proyecto y lograr control de actuadores de forma remota.
- Estudiar la implementación del proyecto para automatización y sistemas domóticos.
- Hacer estudio de mercado y análisis estadísticos del proyecto para sistemas domóticos.

#### 1.2 Objetivos Específicos

- Estudiar estándar de transmisión de datos ETHERNET, características y su relación con el proyecto.
- Estudiar y determinar dispositivos basados en ETHERNET para trabajar junto a la tarjeta de desarrollo ARDUINO.
- Determinar características, beneficios y ventajas que presenta cada dispositivo basado en ETHERNET.
- Estudiar dispositivo servidor serial MOXA Nport 5110 basado en ETHERNET y su comunicación con ARDUINO.
- Programar y configurar servidor serial MOXA NPort 5110 para comunicación con ARDUINO

- Estudiar la comunicación Serie entre servidor serial MOXA Y ARDUINO
- Estudiar sistemas embebidos (SHIELD).
- Estudiar tipos de actuadores, características y sus aplicaciones.
- Establecer la comunicación entre los dispositivos involucrados del proyecto.
- Diseñar y generar Sketch para el control de actuadores a través del software de programación de ARDUINO.
- Utilizar los beneficios del estándar de transmisión de datos ETHERNET para ampliar las prestaciones de nuestro proyecto, permitiendo acceso y control de actuadores de forma remota.
- Estudiar e implementar proyecto para el desarrollo de sistemas domóticos
- Generar y programar interfaz gráfica para el control del actuadores, el cual permita un sistema de automatización (domótica) vía acceso remoto.
- Estudiar y utilizar a nuestro beneficio la tecnología NFC.
- Desarrollar sistema NFC para control de actuadores.

### 1.3 Descripción del Proyecto

Este proyecto enmarca la necesidad de control de actuadores por medio de órdenes y monitoreo a través de microcontroladores, desde la computadora o dispositivos móviles vía estándares de transmisión de datos ETHERNET.

Para el desarrollo de éste, el proyecto consiste de una tarjeta de desarrollo programable ARDUINO de código libre, el cual consta de un puerto serie TTL por el cual es posible configurarlo y programarlo para que ejecute diversas tareas que el usuario requiera. Para propósito nuestro, por medio de sistemas embebidos desarrollados para ARDUINO, también llamados SHIELD, lograr comandar las salidas de este microcontrolador para el control de diferentes actuadores (ya sea distintos tipos de motores como control de relés entre otros).

Se estudiarán distintos componentes basados en ETHERNET los cuales se interconectarán con el microcontrolador ARDUINO antes mencionado, para que este pueda ser comandado vía protocolos TCP/IP para realizar en control de actuadores. Estos dispositivos se estudiarán de forma detallada, en donde deben cumplir con los requerimientos que el proyecto necesita. Para ello se compararán las distintas prestaciones y beneficios que nos ofrecen y así poder determinar la mejor opción.

Los dispositivos a estudiar y posteriormente comparar que están basados en ETHERNET que utilizaremos, corresponden a MOXA NPort 5110 un servidor serial el cual es un convertidor de protocolo TCP/IP a serie (puerto por el cual se comunicará con nuestro microcontrolador ARDUINO) y un sistema embebido (SHIELD) para ARDUINO llamado SHIELD ETHERNET.

Se hará un estudio de los distintos tipos de actuadores y como estos interactuarán con el microcontrolador, su funcionamiento y beneficios posibles hacia el proyecto. Se pretende crear un sistema domótico a través de programación, por intermedio del software que utiliza el microcontrolador.

Este sistema domótico debe ser capaz de hacer control de distintas acciones dentro de un recinto, en donde el usuario sea capaz de comandarlos, ya sea desde su computadora de escritorio como de dispositivos móviles de forma remota. Para lograr esto, se utilizarán los beneficios que presenta ETHERNET donde se explicará el cómo utilizarlos.

Para complementar nuestro proyecto, se le incluirá el uso de tecnología NFC, de las siglas Near Field Communication, que son sensores de proximidad que funcionan por campo magnético presentes en dispositivos tecnológicos de última generación, para aumentar el control y automatización de nuestro sistema domótico. Se explicará y desarrollara un sistema para interactuar con el microcontrolador y así lograr el control de los actuadores por medio de sistema NFC.

A continuación en la figura Fig 1-1 se muestra un resumen del proyecto.

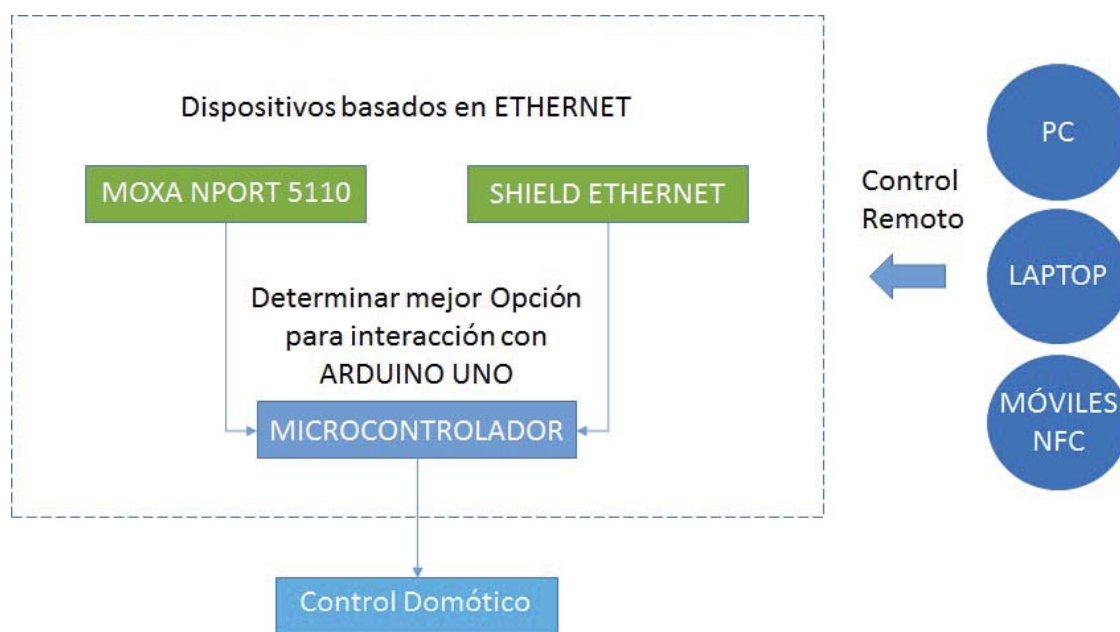


Fig. 1-1: Esquema resumen del proyecto



## CAPÍTULO 2

### USO DE ARDUINO PARA CONTROL DE ACTUADORES ELECTROMECAÑICOS

#### 2.1 ARDUINO

ARDUINO es una plataforma de desarrollo de código y hardware abierto, que permite iniciarnos en el mundo de la programación de manera amigable y sencilla. Esta plataforma permite realizar programación en un lenguaje C/C+ muy similar al lenguaje de programación Processing, por medio del software que provee esta plataforma.

Dentro de las características más relevantes que posee esta plataforma, destaca por la gran facilidad y simplicidad para la comunicación con los diferentes dispositivos con que se conecte, ya sea ordenadores, debido a que el software que posee es multi-plataforma donde es posible programarlo en los distintos sistemas operativos presentes actualmente, como la comunicación entre las mismas plataformas de ARDUINO u otros microcontroladores. [1]

También cuenta con la particularidad de tener un bajo costo en el mercado, lo que marca un punto importante debido a la gran cantidad de prestaciones que nos ofrece versus otros microcontroladores en el mercado que superan su mucho valor.

##### 2.1.1 ARDUINO UNO

Es una de las plataformas de ARDUINO más utilizadas, la cual presenta grandes prestaciones y a un bajo costo, en él es posible generar innumerables Sketch de programación debido al completo abanico de prestaciones que ofrece para realizar las distintas tareas que el usuario desee, el cual será nuestro dispositivo a utilizar para el desarrollo del proyecto.

##### 2.1.2 Hardware de ARDUINO UNO

ARDUINO UNO es un microcontrolador que está basado en ATmega328, el cual está provisto de 14 entradas/salidas programables, de 6 entradas análogas, un resonador de cerámico a 16 KHz, permite la conexión con ordenadores vía USB, jack para alimentación externa y botón de reset.

ATmega328 viene provisto de comunicación serial UART TTL (5V), con pines de entradas digitales 0 (RX) usada para recibir y 1 (TX) para enviar información serial, conectados a los correspondientes pines de microcontrolador serial ATmega8U2 USB a TTL que posee ARDUINO UNO.

A continuación en la figura Fig. 2-1 se presenta plataforma de entrenamiento ARDUINO UNO.

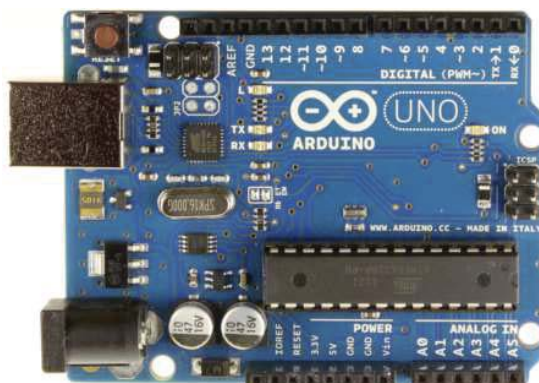


Fig. 2-1: ARDUINO UNO.

En la Tabla 2-1 a continuación se presentan las características del hardware de ARDUINO UNO

Tabla 2-1: Características del hardware de ARDUINO UNO.

Procesador	ATmega328
V de operación	5 V
V de entrada recomendada	7-12 V
Limites de Volt de entrada	6–20 V
Entradas-Salidas digitales	14 (con 6 salidas de control de ancho de pulso PWM)
Entradas Análogas	6
Corriente DC por entrada-salida	40mA
Corriente por pin de 3.3 V	50mA
Memoria Flash	32KB(ATmega328)
SRAM	2KB(ATmega328)
EEPROM	1KB(ATmega328)
Velocidad de reloj	16MHz

Este microcontrolador posee internamente 3 tipos de memorias, las cuales corresponden a Memoria Flash, SRAM y EEPROM donde:

- Memoria Flash: es donde se guarda el programa creado por medio del software que trae esta plataforma escrito en código. Esta memoria tiene la particularidad de no borrarse aun cuando ARDUINO es desenergizado. Su capacidad es de 32Kb.

- SRAM o Static Random Access Memory: es donde se crean y manejan las variables una vez que el código del programa está corriendo. Su memoria es volátil, es decir, pierde su información almacenada si ARDUINO es desenergizado. Su capacidad es de 2Kb.
- EEPROM: es una memoria ROM programable que puede ser borrada de forma electrónica, es no volátil. Su capacidad es de 1Kb.

### 2.1.3 Software de ARDUINO

El software que presenta ARDUINO para su programación es simple e intuitivo, en donde por comandos de C/C+ se puede programar el microcontrolador, el cual cuenta con una interfaz amigable con lo necesario para comenzar a programar.

A continuación en figura 2-2 se aprecia la ventana principal del software de ARDUINO.

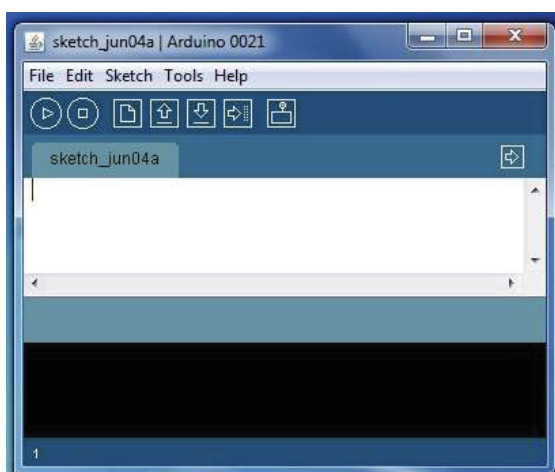


Fig. 2-2: Ventana principal de software de ARDUINO.

Dentro de lo que nos entrega este software, destacan de izquierda a derecha los siguientes controles:

- Verify/Compile: que se encarga de revisar el código y corroborar que no exista algún Error: en el área de color negro se indica posible error o correcto ingreso de códigos.
- Stop: para detener proceso de monitor serie.
- New: para crear nuevos códigos.
- Open: para abrir algún código de otro proyecto guardado.
- Save: para guardar códigos o cambios realizados a este.

- Upload to I/O Board: este permite compilar nuestro código hacia nuestro ARDUINO.
- Serial Monitor: este inicia la monitorización serie del proyecto en cuestión, esto significa que se muestran los datos de algún determinado parámetro enviados desde la placa ARDUINO hacia la interfaz o viceversa.

Con estas herramientas ya es posible comenzar a programar nuestra tarjeta de desarrollo ARDUINO (cada vez que se quiera compilar un Sketch de programación, el software automáticamente verifica y señala errores para su posterior corrección, evitando posibles errores que puedan dañar el dispositivo). Cabe destacar que dentro de la página web de ARDUINO se encuentran las librerías y los comandos que se disponen, la cual señala el cómo deben usarse correctamente, donde se exponen ejemplos simples y prácticos.

Dentro del software de ARDUINO vienen contenidas librerías que amplían la funcionalidad de esta tarjeta de desarrollo, permitiendo el control de mayor cantidad de dispositivos que se puedan conectar con él, estas librerías pueden ser modificadas o agregarse otras nuevas (principalmente de otros usuarios o desarrolladores de SHIELDS para ARDUINO) , debido a que como esta plataforma es de código y hardware libre, muchos diseñadores y programadores constantemente buscan ampliar sus prestaciones para satisfacer sus necesidades.

Se debe señalar también que ARDUINO desarrolla sistemas embebidos llamados SHIELDS los cuales, dependiendo de su diseño y forma, interactúan con esta plataforma para ampliar sus prestaciones.

## 2.2 Sistemas embebidos SHIELDS y ARDUINO

Los sistemas embebidos son extensiones de hardware que cumplen la función de ampliar las capacidades y prestaciones de la plataforma ARDUINO, diseñados principalmente para cubrir necesidades más específicas, estos sistemas embebidos pueden transformar proyectos de simples a otros más complejos y más completos. Por lo general los sistemas embebidos se pueden programar directamente en el lenguaje del mismo microcontrolador, lo que favorece en la adaptación de este a nuestro proyecto. La mayor parte de estos sistemas embebidos desarrollados para ARDUINO utilizan la misma fuente de poder que utiliza esta plataforma, lo que simplifica aún más su configuración.

Existe una cantidad de sistemas embebidos que se desarrollan para trabajar con la plataforma ARDUINO ya sean oficiales o no, como pueden ser SHIELD ETHERNET, de Motores, WIFI, DISPLAYS, GPS, Bluetooth entre varios otros y en donde se hace interminable la lista de estos sistemas embebidos.

Estos sistemas vienen provistos de librerías oficiales como no oficiales para hacer funcionar estos SHIELDS junto al software de ARDUINO, en donde también es posible hacer modificaciones o crear librerías uno mismo y comandar a gusto nuestro estos dispositivos.

### 2.2.1 SHIELD ETHERNET

Este sistema embebido está creado para interactuar junto con la placa de entrenamiento ARDUINO y cumple con la función de integrar al microcontrolador la capacidad de comunicarse a través de red local vía ETHERNET con otros dispositivos o computadores por medio de sus distintas funciones programables que posee. [2]

Este SHIELD ETHERNET posibilita a ARDUINO por medio de diferentes configuraciones operar en diferentes modos, como los son el modo Servidor, modo Cliente o modo UDP, lo que permite tanto el enviar como el recibir datos y actuar con cada configuración. [3]

A continuación en la figura 2-3 se presenta un SHIELD ETHERNET creado para funcionar junto a ARDUINO.

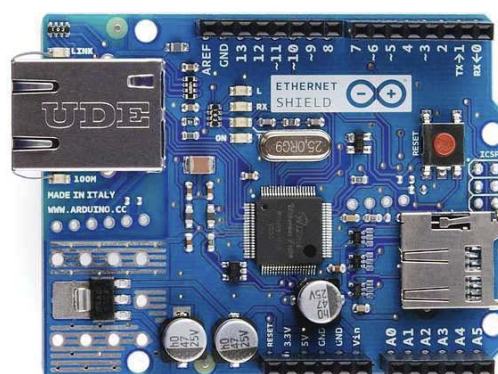


Fig. 2-3: Imagen de SHIELD ETHERNET para ARDUINO.

### 2.2.2 SHIELD de Motores

Este SHIELD fue diseñado hacer control de distintos tipos de motores y en donde viene a suplir una deficiencia de la plataforma ARDUINO UNO por sí sola, que es el hacer control de motores y hacerlo de manera simultánea.

Dentro de las características de este SHIELD de motores, destacan el control para dos servo-motores de entre 4.5 y 25V DC con 0.6 A por cada uno, comandar hasta un máximo de cuatro motores DC bidireccionales con control de velocidad de 8 bits y el control de dos motores paso a paso tanto unipolar como bipolar para bobinado simple de poca fuerza y bajo consumo de energía, y el bobinado doble para mayor fuerza, con un mayor consumo de energía o una mezcla entre ambos.

Si bien este SHIELD es capaz de controlar servo-motores sin ningún problema, para hacer funcionar motores DC o motores paso a paso se requiere de energía por medio de una fuente externa de entre 5 y 24 Vcc. [4]

A continuación en la figura 2-4 se presenta el SHIELD de Motores Adafruit.



Fig. 2-4: Imagen de SHIELD de Motores Adafruit para ARDUINO.

## 2.3 Actuadores electromecánicos y ARDUINO

Existe una gran cantidad de actuadores del tipo electromecánico, se verán algunos de los más conocidos y utilizados, como lo son los Motores paso a paso, Servo-motor, Motores DC y Relé electromagnético para su control por medio de ARDUINO.

### 2.3.1 Motores paso a paso

Este dispositivo electromecánico es capaz de avanzar por grados discretos o pasos dependiendo de su entrada de control, corre con la ventaja de tener alta precisión donde es capaz de avanzar en ambas direcciones y modificar de forma sencilla su velocidad, es capaz de mantener su posición mientras se encuentra energizado.

Dentro de este tipo de motor, los más usados vienen a ser los motores pasos a paso unipolar y bipolar que se explican a continuación:

- Motor paso a paso unipolar: un motor simple de comandar el cual consta de 5 o 6 cables para su conexión, con uno en común de alimentación y los otros restantes para su control. Este tipo de motor es capaz de configurarse para controlar su fuerza como su velocidad y alternar entre estos.
- Motor paso a paso bipolar: algo más complejo su control el cual consta de 4 cables para su conexión, su funcionamiento requiere de cambios de flujo de corriente que recorre las bobinas internas del motor en una secuencia adecuada, para su correcto funcionamiento. [5]

### 2.3.2 Servo-motor

Los servo-motores son utilizados en las más variadas aplicaciones donde una elevada dinámica, precisión de velocidad y posicionamiento son factores decisivos para proyectos o industrias. Poseen características tales como un bajo costo, elevado desempeño y robustez. [6]

Estos motores trabajan por medio de pulsos, los cuales dependiendo de ellos, determinan cuanto avanzan, son capaces también de permanecer en su posición cuando están energizados. Existen servomotores que solo pueden avanzar hasta 180 grados como los que completan una vuelta completa de 360 grados. Se utilizan principalmente para controles industriales de alta precisión como también para control en modelismo de aviones, autos u otros dispositivos a control remoto entre otros.

### 2.3.3 Motores DC

Este tipo de motor es de los más conocidos, que es capaz de regular su velocidad desde vacío a carga completa, su fácil control de posición, par y velocidad lo convierte una de las mejores opciones en aplicaciones de control y automatización. Presenta en gran parte de equipos como ventiladores de equipos electrónicos, motores de autos a control remoto, entre otros.

### 2.3.4 Relé electromagnético

Dispositivo de accionamiento que funciona por medio de una bobina y un electroimán el cual actúa como un interruptor controlado eléctricamente. Dentro de las ventajas de estos dispositivos, es que cuentan con la capacidad de controlar sistemas de mayor potencia utilizando muy poca energía, donde puede dar control de uno o más interruptores simultáneamente. Existen relés que pueden ser accionados por unos pocos miliAmperes y controlar corrientes de más de 10 Amperes a 220 Volts alternos.

## CAPÍTULO 3

### ESTANDAR DE TRANSMISIÓN PARA REDES LOCALES: ETHERNET

#### 3.1 Estándar ETHERNET IEEE 802.3

Estándar de redes de área local, base del protocolo internacional IEEE 802.3, el cual define tipo de cableado y señalización de nivel físico y los formatos de tramas de datos del nivel de enlace de datos del modelo OSI.

Todos los equipos de una red ETHERNET están conectados a la misma línea de transmisión y la comunicación se lleva a cabo por medio de la utilización un protocolo denominado CSMA/CD (Carrier Sense Multiple Access with Collision Detect, que significa que es un protocolo de acceso múltiple que monitorea la portadora: detección de portadora y detección de colisiones). [7]

#### 3.2 ETHERNET y ARDUINO UNO

Para lograr que ARDUINO UNO pueda trabajar y operar bajo los estándares de transmisión para redes locales ETHERNET, se dispondrá de dos dispositivos, que al interconectarse junto a la plataforma ARDUINO UNO son capaces de lograr este cometido, estos dispositivos corresponden a NPort 5110 de MOXA y SHIELD ETHERNET diseñado para trabajar junto a ARDUINO, ambos dispositivos basados en este estándar.

##### 3.2.1 MOXA Nport 5110

Este es un dispositivo servidor serial diseñado para comunicaciones industriales, basado en ETHERNET, permite la conversión de protocolo TCP/IP para la comunicación con dispositivos serial RS-232. Su interfaz estándar TCP/IP permite distintos modos de operación como servidor TPC, cliente TCP, UDP cliente/servidor, modo Módem ETHERNET. [8]

La interfaz RS-232 que presenta se usa para conexiones punto a punto y a cortas distancias, por lo que no presenta inconvenientes para el proyecto, además de ser fabricado para comunicarse y ser compatible con la mayoría de los dispositivos con esta interfaz. La profundización sobre este dispositivo y su configuración se verá con posterioridad en capítulos posteriores.

A continuación en la figura 3-1 se presenta dispositivo MOXA NPort 5110.





Fig. 3-1: Imagen de dispositivo RS232 universal MOXA NPort 5110.

A continuación en Tabla 3 – 1 se presentan algunas especificaciones técnicas del dispositivo

Tabla 3-1: Especificaciones Técnicas de dispositivo MOXA NPort 5110.

LAN:	
ETHERNET	10/100 Mbps, RJ45, auto MDI/MDIX
SERIAL:	
Interface	RS-232
Puertos	1
Tipo de puerto	DB9 macho
Tipos de señales	TxD, RxD, RTS, CTS, DTR, DSR, DCD y GND

### 3.2.2 SHIELD ETHERNET para ARDUINO UNO

Este dispositivo mencionado en capítulos anteriores, posibilita a ARDUINO UNO por medio de diferentes configuraciones, el operar bajo el estándar de ETHERNET, generando el acceso a la plataforma desde la red local. Este dispositivo permite operar en distintos modos como son el modo Servidor, modo Cliente o modo UDP, lo que permite tanto el enviar como el recibir datos y actuar con respecto a éstos.

A continuación en la figura 3-2 se presenta la conexión entre ARDUINO UNO y SHIELD ETHERNET donde se especifica los pines utilizados para su comunicación.

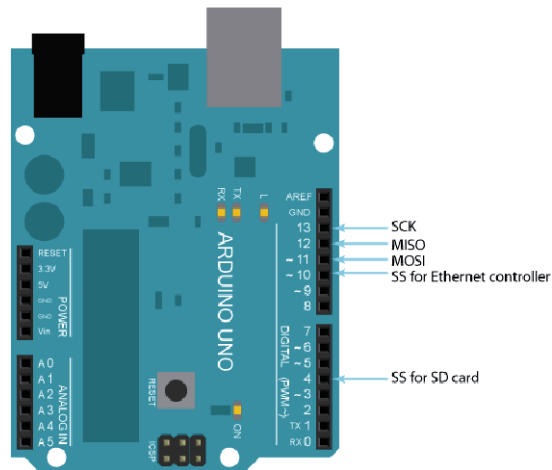


Fig. 3-2: ARDUINO UNO y conexión con SHIELD ETHERNET.

En capítulos posteriores se verá la conexión y comunicación de este SHIELD con ARDUINO UNO.

## CAPÍTULO 4

### COMUNICACIÓN DE DISPOSITIVOS BASADOS EN ETHERNET Y PLATAFORMA ARDUINO

#### 4.1 MOXA NPort 5110 junto a ARDUINO UNO

La comunicación que se requiere entre estos dispositivos debe ser del tipo Serial, y para ello, es necesario implementar un circuito adaptador el cual comunicará ambos dispositivos, el motivo se debe a que el puerto serie de ARDUINO (UART) presenta niveles TTL de 5V. [9]

##### 4.1.1 Circuito adaptador para comunicación Serie entre dispositivos

Para hacer efectiva la comunicación entre ambos dispositivos, es necesario implementar un circuito que adapte las señales que se envían entre los dispositivos, para ello se utilizará el integrado MAX232N, el cual es un transmisor/receptor dual (contiene 2 entrada Serie, 2 salidas Serie, 2 entradas TTL/CMOS y 2 salidas TTL/CMOS), diseñado para convertir entradas TIA/EIA-232-F (de tipo Serie) a niveles TTL/CMOS de 5V.

Este circuito integrado permite la conexión y comunicación entre el puerto Serie de ARDUINO (UART) y el puerto Serie del dispositivo MOXA.[10]

A continuación en la figura 4-1 se muestran algunas de las características que presenta la familia del circuito integrado MAX232N.

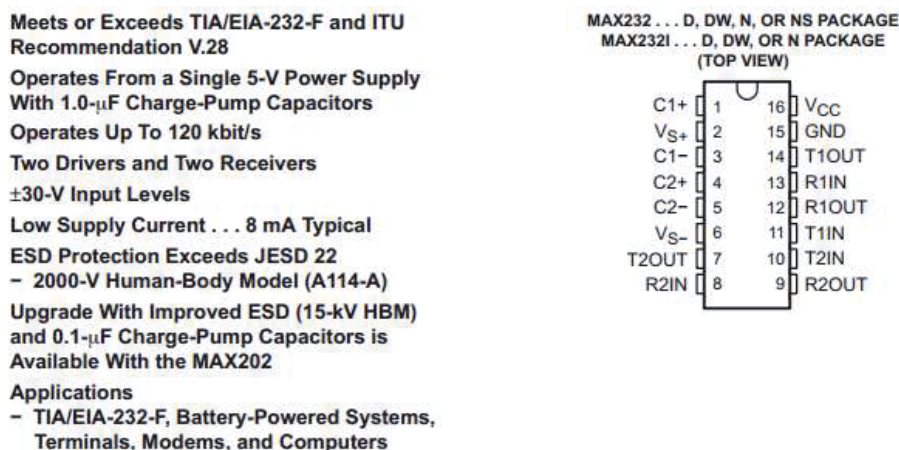


Fig. 4-1: Circuito integrado MAX232N.

La configuración de este circuito integrado para que pueda transmitir/recibir y permitir la comunicación entre los dispositivos se muestra en la figura 4-2, donde esta configuración proviene del mismo DATASHEET del circuito integrado. [10]

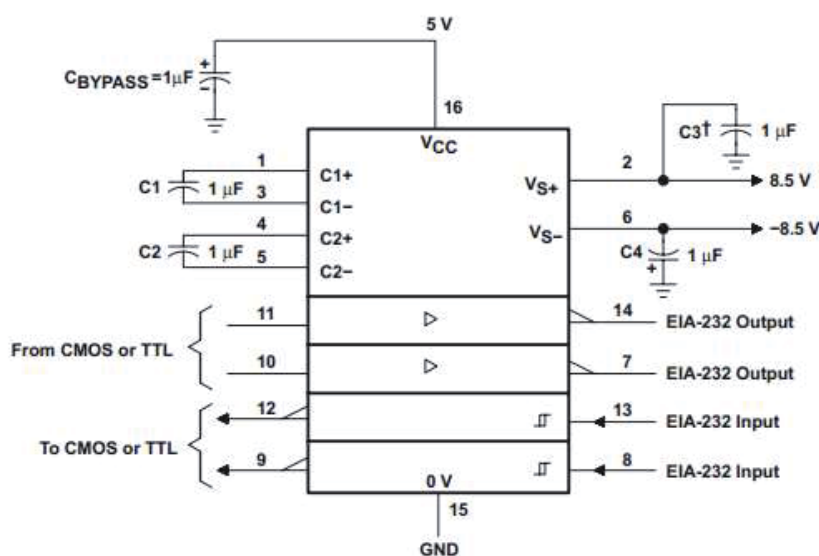


Fig. 4-2: Configuración de integrado MAX232N.

En donde los terminales del circuito integrado serán:

- 8 - 13 : Recibirán datos desde el terminal Serie del dispositivo MOXA
- 7 - 14 : Enviarán datos hacia el terminal Serie de dispositivo MOXA
- 9 - 12 : Recibirán datos desde el terminal Serie de ARDUINO UNO
- 10 – 11: Enviarán datos hacia el terminal Serie de ARDUINO UNO

## 4.2 Configuración de dispositivo MOXA NPort 5110

Para la comunicación entre PC y dispositivo MOXA (se referirá de esta forma al dispositivo MOXA NPort 5110 de este punto en adelante), se debe configurar ambas partes para la correcta comunicación entre estos, la idea de esta configuración trata en crear una comunicación entre cliente y servidor por medio de protocolos de comunicación TCP/IP, por lo que se requiere realizar una serie de pasos para lograr este propósito, que se especificarán en la subsección 5.2.1.

El protocolo de control de transmisión o TCP se utiliza para crear conexiones entre computadoras para realizar envío de flujo de datos entre ellas, donde este protocolo garantiza que los datos enviados serán entregados a destino sin errores, donde también utiliza el concepto de puerto para asignación de máquinas conectadas y así evitar posibles conflictos y pérdidas de datos. [11]

### 4.2.1 Modo Servidor TCP de MOXA NPort 5110

Como se mencionó anteriormente, se debe crear una comunicación entre los dispositivos involucrados, para dicho propósito, la computadora viene a ser nuestro

cliente TCP y Nport el servidor TCP, el cual irá conectado vía serie al dispositivo ARDUINO UNO.

A continuación en la figura 4-3 se muestra como debe ser la comunicación entre los dispositivos.

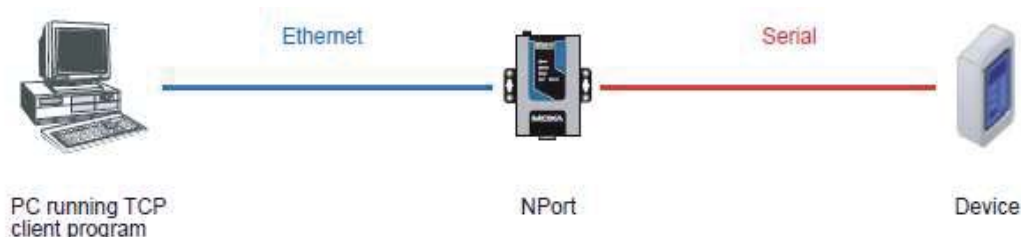


Fig. 4-3: Comunicación entre dispositivos.

Es esta la configuración en que nos enfocaremos que nos permitirá completar parte de nuestro objetivo final, el de comandar un sistema de motores vía ETHERNET.

#### 4.2.2 Modo servidor TCP en MOXA

En el modo servidor TCP, el puerto serie de nuestro Nport se le asigna un número de puerto, el cual no evita entrar en conflicto con algún otro puerto serie del Nport, para el caso de nuestro dispositivo, cuenta con 1 entrada serie por lo que se evita conflictos de este tipo. El host (u ordenador) establece conexión con el Nport y recibe información desde el dispositivo serie.

Este dispositivo cuenta con la ventaja de que múltiples host pueden acceder a la información del dispositivo serie conectado de forma simultánea. [8]

#### 4.2.3 Configuración de MOXA como servidor TCP

Como primera medida, el dispositivo MOXA debe estar con los valores que trae por defecto, para ello trae consigo un botón de reset para realizar esta acción, con esto aseguramos que no haya problemas a la hora de configurar nuestro dispositivo.

Una vez bien conectados los dispositivos cliente-servidor, por medio del conector ETHERNET, se procede a configurar nuestro servidor TCP, para ello se procede a modificar las IP de los dispositivos, para que ambas se encuentren en la misma red, por defecto, nuestro MOXA que será nuestro servidor TCP trae consigo la dirección IP 192.168.127.254 con máscara de subred 255.255.255.0 por lo que procedemos a hacer las modificaciones para nuestra tarjeta de red, en propiedades de protocolo de internet (TCP/IPv4) donde modificamos y asignamos manualmente la nueva IP, la IP

192.168.127.253 con máscara de subred 255.255.255.0, así con ello lograr que ambos equipos se encuentren en la misma red.

A continuación en la figura 4-4 muestra la asignación manual de una IP.

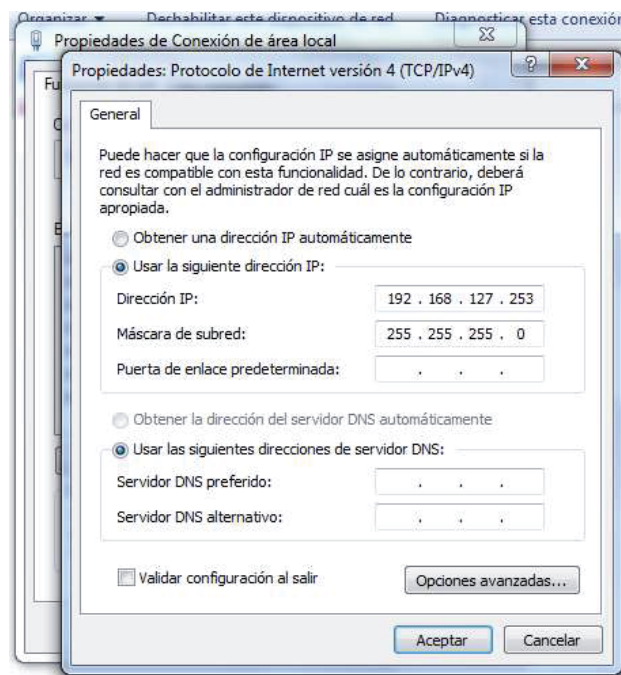


Fig. 4-4: Asignación manual de IP y máscara de subred.

Habiendo hecho esto, lo que sigue es configurar nuestro dispositivo MOXA para que trabaje como servidor TCP, para ello nos dirigimos a la consola web de nuestro dispositivo, escribiendo la dirección IP de este en el navegador de nuestra computadora y acceder a ella.

En esta consola web de MOXA se encuentran herramientas necesarias para configurar nuestro dispositivo a nuestro gusto, en donde podemos encontrar las siguientes funciones:

A continuación en la Tabla 4-1 se muestra los distintos comandos que presenta la consola del equipo MOXA para su configuración.

Tabla 4-1: Comandos dentro de consola de MOXA para su configuración.

Basic Settings	Permite modificar nombre del servidor, activar y desactivar opciones
Network Settings	Permite modificar opciones de internet como IP, si es estática o dinámica, etc.
Serial Settings	Permite modificar los parámetros para la comunicación serial
Operating Settings	Permite modificar tipo de operación, TCP cliente, servidor, UDP, etc.
Accesible IP Settings	Permite restringir acceso a dispositivo.
Auto Warning Settings	Da opciones de aviso de seguridad.
Monitor	Monitorea datos que procesa el dispositivo.
Change Password	Permite asignar o modificar contraseña para acceso a Consola Web.
Load Factory Default	Restablece valores de fábrica del dispositivo.
Save/Restart	Guarda cambios realizados en el dispositivo.

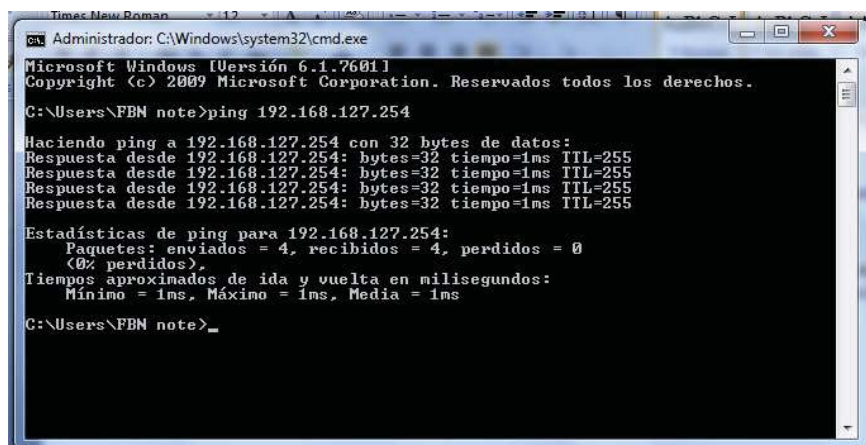
Dentro de estas funciones, nos dirigimos a configurar, para el propósito que necesitamos, en la función Serial Settings, seleccionamos las siguientes opciones:

- Baud Rate    9600
- Data bits     8
- Stop bits     1

Que son las configuraciones necesarias para comunicarnos con el microcontrolador ARDUINO UNO.

Luego en Operate mode, seleccionar la opción TCP Server, guardamos los cambios y ya tenemos nuestro dispositivo en modo servidor TCP configurado y listo para operar.

Algo importante es corroborar que exista comunicación entre MOXA y nuestra computadora, para ello, por medio de consola de Windows, se puede enviar un ping con la dirección IP de nuestro MOXA, como se muestra en la figura 4-5 a continuación:



```

Administrador: C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\FBN note>ping 192.168.127.254

Haciendo ping a 192.168.127.254 con 32 bytes de datos:
Respuesta desde 192.168.127.254: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.127.254: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.127.254: bytes=32 tiempo=1ms TTL=255
Respuesta desde 192.168.127.254: bytes=32 tiempo=1ms TTL=255

Estadísticas de ping para 192.168.127.254:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 1ms, Máximo = 1ms, Media = 1ms

C:\Users\FBN note>_

```

Fig. 4-5: Verificación de conexión en consola de Windows.

#### 4.2.4 Configuración de ARDUINO UNO

Esta etapa de configuración a través del software que nos entrega ARDUINO hace el trabajo más simple, ya que solo se necesita escribir comandos en el Sketch de programación del software para que este dispositivo quede escuchando por el puerto Serie y lograr la comunicación.

A continuación en la figura 4-6 se presenta la inclusión de los comandos para la comunicación serie en el software de ARDUINO.



```

ejemplo_datos_por_puerto_serie | Arduino 1.0.5
Archivo Editar Sketch Herramientas Ayuda

ejemplo_datos_por_puerto_serie

#include <SoftwareSerial.h>

SoftwareSerial mySerial(0, 1); // RX, TX

void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }
}

```

Fig. 4-6: Configuración para comunicación Serie en ARDUINO.

El comando para hacer posible la comunicación viene dentro de las librerías del software de ARDUINO o puede ser descargado desde la página web del proveedor del SHIELD adquirido. Estos códigos corresponden a <SoftwareSerial.h> para activar la comunicación Serie, SoftwareSerial mySerial (0, 1) que indica terminales del



microcontrolador ARDUINO UNO por donde se hará la comunicación Serie y Serial.begin (9600) que inicia el puerto serie a una velocidad de 9600 Baud (unidades de señal por segundo que deben coincidir con las configuraciones por parte del dispositivo MOXA) y que indica que permanecerá a la escucha de alguna comunicación por ese puerto Serie como se muestra en figura 4-6.

Las pruebas realizadas y el análisis de la comunicación entre dispositivo MOXA, circuito adaptador y ARDUINO se incluyen en el APENDICE B de este informe.

### 4.3 SHIELD ETHERNET y ARDUINO UNO

Este SHIELD simplemente se monta sobre ARDUINO y donde utiliza los pines 10, 11, 12 y 13 de la plataforma para su comunicación (por lo que estos pines quedan inhabilitados, disminuyendo la cantidad de salidas que dispongamos) el cual amplía las funciones del microcontrolador para realizar comunicación vía ETHERNET. [12]

Utiliza las librerías `#include <SPI.h>` e `#include <Ethernet.h>` para su funcionamiento dentro del software de ARDUINO.

A continuación en la figura 4-7 se aprecia un SHIELD ETHERNET junto a ARDUINO UNO.



Fig. 4-7: SHIELD ETHERNET junto a ARDUINO UNO.

## CAPÍTULO 5

### COMPARATIVAS ENTRE DISPOSITIVOS ETHERNET PARA EL CONTROL DE ACTUADORES A TRAVÉS DE ARDUINO

Para realizar esta comparativa, se evaluarán 4 puntos importantes, los cuales corresponden a configuración, funcionalidad, conexión y simpleza de ambos dispositivos, MOXA NPort 5110 y SHIELD ETHERNET para su funcionamiento junto a ARDUINO.

#### 5.1 Configuración

Comenzando por el servidor MOXA, este cuenta con un amplio conjunto de configuraciones por las cuales puede operar, debido a que es un dispositivo universal, creado para poder interactuar con la mayoría de los dispositivos serie existentes, el NPort 5110 está desarrollado para comunicación serie RS-232.

Para el caso de SHIELD ETHERNET, basta escribir unas pocas líneas de códigos dentro del Sketch de programación para configurarlo, debido a que está creado para trabajar con ARDUINO.

#### 5.2 Funcionalidad

El dispositivo servidor MOXA presenta una gran cantidad de funciones, su interfaz estándar TCP/IP permite distintos modos de operación como servidor TPC, cliente TCP, UDP cliente/servidor, modo Módem ETHERNET. Este dispositivo al interactuar con ARDUINO UNO solo por sus terminales serial (solo 2 terminales, Rx para recibir datos y Tx, para transmitir), deja a disposición la mayoría de los terminales que presenta el microcontrolador, permitiendo mayor número de control de salida/entrada.

SHIELD ETHERNET posee menos funciones que las que entrega MOXA, presentando el modo servidor, modo cliente y modo UDP.

Al estar montado sobre la plataforma ARDUINO UNO (es posible montarlo en casi todas las plataformas que ofrece ARDUINO), este SHIELD utiliza algunos de los terminales del microcontrolador para su comunicación, restando entradas/salidas y limitando un mayor número de control de actuadores.

#### 5.3 Conexión

Por parte del servidor MOXA, presenta problemas para ser conectado de forma directa con ARDUINO por medio de sus terminales serie, y para hacer efectiva la conexión entre ambos dispositivos, es necesario implementar un circuito que adapte las señales que se enviarán entre sí, viene provisto de fuente de alimentación por lo que no utiliza energía de la plataforma ARDUINO.

Por otra parte, el SHIELD ETHERNET solo basta montarlo sobre ARDUINO y se comunican sin problemas y sin necesidad de circuitos anexos. Utiliza la fuente de energía del mismo ARDUINO.

#### 5.4 Simpleza

Por parte de MOXA, se requiere tener un mayor manejo en protocolos de comunicación TCP/IP, en el proceso de configuración se ve la necesidad de tener que implementar un software dedicado para lograr obtener y enviar datos. Al tener fuente de alimentación propia, se debe tener en cuenta que el sistema necesita una toma de energía de la red eléctrica domiciliaria extra.

Para la comunicación con ARDUINO, necesariamente hay que implementar un sistema adaptador, lo que conlleva aumentar cantidad de componentes en el proyecto, como también energía extra para hacer funcionar el dispositivo adaptador.

Debido a utilizar muchos más componentes para la comunicación con la plataforma, se hace mayor el espacio que utilizará el proyecto como también la cantidad de cableados para su funcionamiento.

Por otro lado, SHIELD ETHERNET es simple, fácil de conectar y más amigable ya que trabaja con el mismo lenguaje de programación de ARDUINO. Al utilizar la misma fuente de alimentación de ARDUINO, se evita mayor concentración de conexiones para su funcionamiento y reduce considerablemente el espacio utilizado para el proyecto.

#### 5.5 Resumen comparativa

Podemos decir que aunque el dispositivo MOXA presenta una completa prestación de funciones, con una interfaz depurada y programable completamente, para comunicarse con ARDUINO, cae en problemas al requerir de circuitos anexos para poder comunicarse, al contrario de competir con un dispositivo creado para operar junto a ARDUINO como es el SHIELD ETHERNET, simple y rápido de configurar.

Al utilizar el SHIELD ETHERNET, éste nos permite crear una interfaz gráfica, simple y funcional, que nos favorece a la hora de generar algún proyecto.

A continuación en la Tabla 5-1 se presenta la comparativa que resume diferencias entre ambos dispositivos basados en ETHERNET para funcionar con ARDUINO:

Tabla 5-1: Comparativa resumen entre dispositivos basados en ETHERNET.

	MOXA NPort 5110	SHIELD ETHERNET
Configuración	Completa cantidad de funciones programables, requiere manejo en protocolos de comunicación, interfaz amigable	Simpleza para configurarlo, requiere manejo de lenguaje C y variaciones que utiliza ARDUINO
Funcionalidad	Completo ofrecimiento de servicios, servidor TPC, cliente TCP, UDP cliente/servidor, modo Módem ETHERNET, permite liberar entrada/salida del microcontrolador al sólo utilizar comunicación serial.	Limitadas funciones, modo servidor, modo cliente y modo UDP, utiliza terminales de entrada/salida de ARDUINO para su comunicación, limitando control de varios actuadores.
Conexión	Requiere circuito anexo para comunicación con ARDUINO (adaptación TTL-CMOS debido a ser un dispositivo de propósito Universal), tiene su propia fuente de alimentación	Solo basta montarlo sobre placa ARDUINO (Desarrollado para funcionar junto con ARDUINO), utiliza fuente de alimentación de ARDUINO
Simpleza	Requiere conocimientos de protocolos de comunicación, sistema más complicado, mayor cantidad de conexiones y aumento de utilización de espacio debido a componentes extra que se deben implementar	Trabaja directamente con ARDUINO y se programa con el mismo software de ARDUINO, no requiere fuente de alimentación externa y el uso de espacio es mínimo

## 5.6 Elección entre dispositivos comparados

Como se puede apreciar, ambos dispositivos presentan ventajas y desventajas a la hora de trabajar con la plataforma ARDUINO, en el resumen anteriormente visto, se ve claramente que si bien un dispositivo conversor universal como lo es MOXA NPort 5110 y lo completo que puede llegar a ser en lo que refiere a prestaciones, necesariamente necesita mayor dominio por parte del usuario en distintos temas de comunicación, ya sea del tipo de protocolos TCP/IP como manejo de comunicación serie con otros dispositivos, temas muy comunes cuando se trata de implementación de estos dispositivos en áreas industriales. Por otro lado, SHIELD ETHERNET es más simple, al estar diseñado especialmente para trabajar con ARDUINO, lo que simplifica significativamente todo el proceso de configuración e implementación, y debido a esta simple pero importante razón es que se utilizará este dispositivo para el desarrollo de nuestro proyecto.

## CAPÍTULO 6

### CONTROL DE ACTUADORES A TRAVÉS DE SHIELD ETHERNET JUNTO ARDUINO Y SISTEMA NFC

#### 6.1 Diseño y programación de Sketch para control de actuadores electromecánicos

Como debemos controlar actuadores vía ETHERNET se creará un programa el cual será capaz de controlar las salidas disponibles de nuestra plataforma de entrenamiento ARDUINO. Si bien el SHIELD ETHERNET para ARDUINO permite el acceso y control de forma directa desde el ordenador hacia el microcontrolador por medio del cable CAT5e y sus respectivo conector RJ45, éste también puede ser conectado a un Router y así utilizar los beneficios que nos ofrece este dispositivo para poder acceder al control de actuadores de forma remota, a través de redes inalámbricas locales, como varios equipos conectados en la misma red.

La forma de operar de este programa consiste en que ARDUINO y su SHIELD ETHERNET funcionen en modo servidor dentro de la misma red donde nos encontramos, el cual se mantendrá a la espera de algún cliente, que puede acceder de forma directa por algún ordenador o de forma remota vía WIFI por medio de dispositivos móviles.

La forma de operación será el control de las salidas de ARDUINO a través de la red local, donde la plataforma ARDUINO vendrá a operar como si fuese un pequeño PLC de baja potencia capaz de controlar varias salidas simultáneamente.

A continuación en la figura 6-1 se presenta la comunicación entre Router y ARDUINO vía cable CAT5.

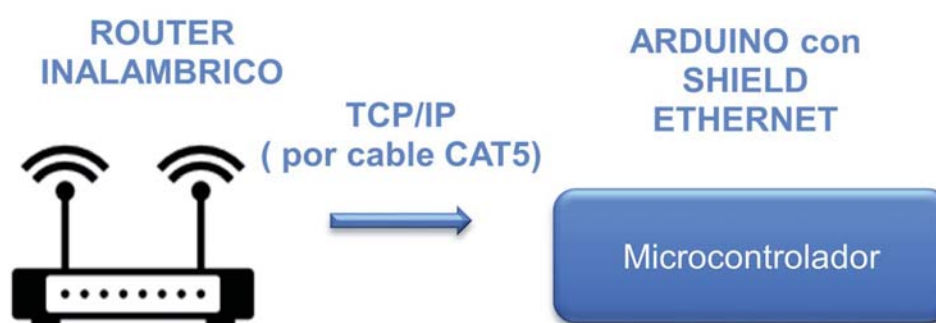
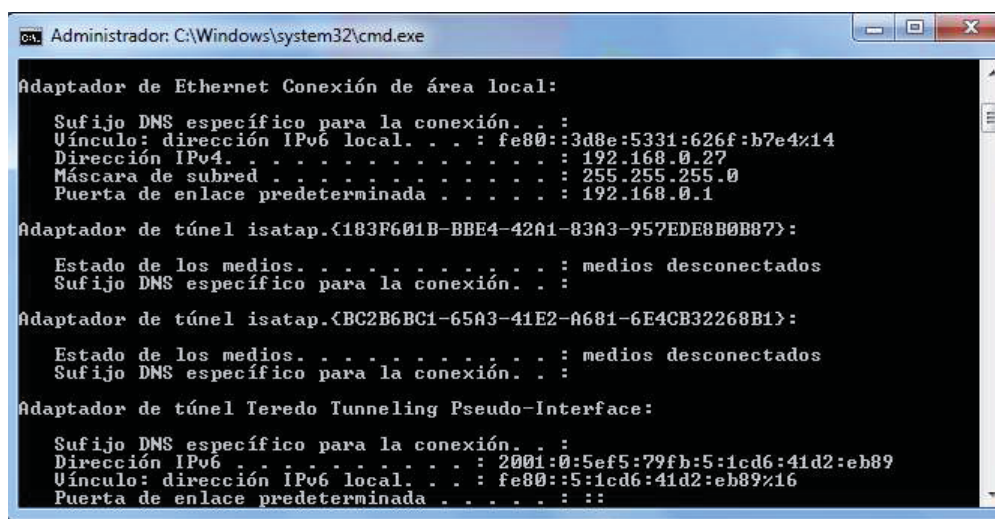


Fig. 6-1: Comunicación de Router inalámbrico y ARDUINO por cable CAT5.

Para el correcto funcionamiento del programa, es necesario que ARDUINO UNO se encuentre en la misma red en que se encuentran nuestros equipos, por lo que se debe comprobar la dirección IP en donde estamos trabajando, para ello, al igual que se hizo para la conexión del en red del dispositivo MOXA vista en capítulos anteriores, por

medio de CMD de Windows, y el comando ipconfig, determinaremos la red donde nos encontramos trabajando.

A continuación en Fig. 6-2 se muestra el comando ipconfig dentro de sistema operativo Windows para determinar parámetros importantes para que el Router pueda detectar y permitir la comunicación con ARDUINO y su SHIELD ETHERNET:



```

Administrador: C:\Windows\system32\cmd.exe

Adaptador de Ethernet Conexión de área local:
    Sufijo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . . . : fe80::3d8e:5331:626f:b7e4%14
    Dirección IPv4. . . . . : 192.168.0.27
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . : 192.168.0.1

Adaptador de túnel isatap.{183F601B-BBE4-42A1-83A3-957EDE8B0B87}:
    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

Adaptador de túnel isatap.{BC2B6BC1-65A3-41E2-A681-6E4CB32268B1}:
    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

Adaptador de túnel Teredo Tunneling Pseudo-Interface:
    Sufijo DNS específico para la conexión. . . :
    Dirección IPv6 . . . . . : 2001:0:5ef5:79fb:5:1cd6:41d2:eb89
    Vínculo: dirección IPv6 local. . . . . : fe80::5:1cd6:41d2:eb89%16
    Puerta de enlace predeterminada . . . . . :
  
```

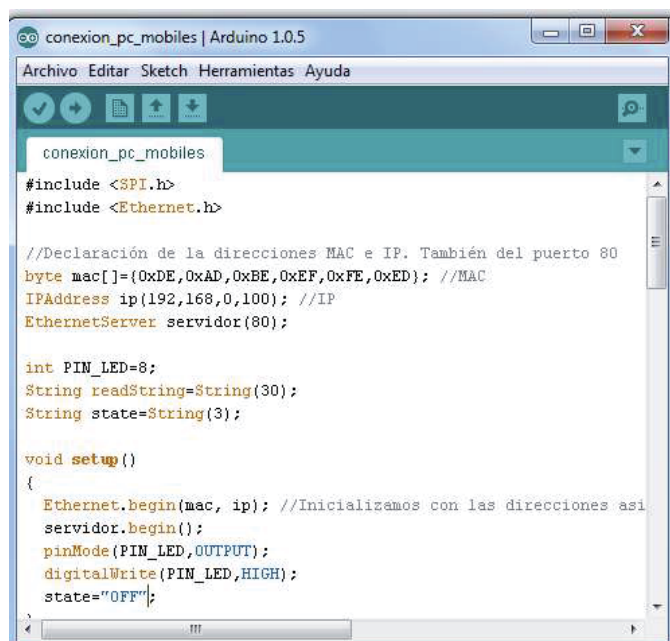
Fig. 6-2: Comando ipconfig por medio de CMD de Windows.

En ella se aprecia que la IP de la red de trabajo entregada por nuestro Router inalámbrico es la 192.168.0.1 por lo que debemos asignar a ARDUINO una IP dentro de ese dominio.

Dentro de la programación para el control de actuadores, se debe partir por la asignación de una IP y una MAC para el posterior acceso a nuestro ARDUINO, por lo que se debe asignar la IP 192.168.0.100 para que este dispositivo se encuentre dentro de la misma red que entrega el Router utilizado y una MAC DE-AD-BE-EF-FE-ED la cual es aleatoria, donde se debe tener presente que estos datos no deben ser la mismos que cualquiera de los dispositivos pertenecientes a la misma red, para evitar problemas posteriores, esto se puede apreciar más adelante en la figura 6-3 donde se muestra parte del código del programa.

El Sketch completo del programa se encuentra en la sección APENDICE A al final de este informe.

Una vez cargado el Sketch a nuestra plataforma de desarrollo ARDUINO (señalado en apéndice A), en conjunto con SHIELD ETHERNET ya podemos establecer una comunicación con el Router inalámbrico, y con ello tener acceso al control de estos dispositivos vía ETHERNET de forma remota a través de cable CAT5.



```

conexion_pc_mobiles | Arduino 1.0.5
Archivo Editar Sketch Herramientas Ayuda

conexion_pc_mobiles
#include <SPI.h>
#include <Ethernet.h>

//Declaración de la direcciones MAC e IP. También del puerto 80
byte mac[]={0xDE,0xAD,0xBE,0xEF,0xFE,0xED}; //MAC
IPAddress ip(192,168,0,100); //IP
EthernetServer servidor(80);

int PIN_LED=8;
String readString=String(30);
String state=String(3);

void setup()
{
  Ethernet.begin(mac, ip); //Inicializamos con las direcciones asi
  servidor.begin();
  pinMode(PIN_LED,OUTPUT);
  digitalWrite(PIN_LED,HIGH);
  state="OFF";
}

```

Fig. 6-3: Código de programa para ARDUINO con SHIELD ETHERNET.

A continuación en la figura 6-4 se muestra la interfaz gráfica programada (Sketch apéndice A) para el control de actuadores accedida desde un navegador WEB de dirección 192.168.0.100.

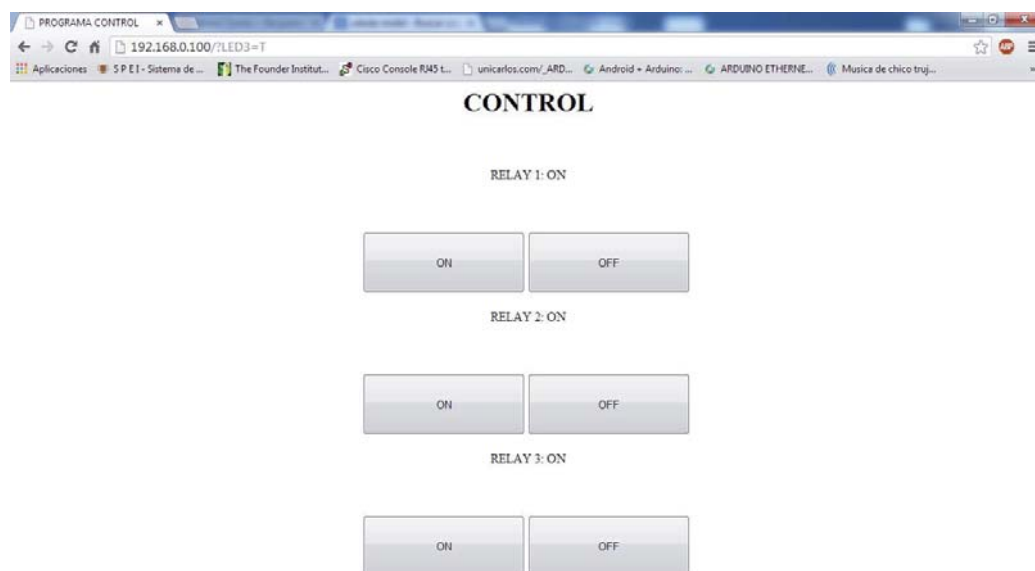


Fig. 6-4: Servidor de ARDUINO a través de navegador web desde Windows.

Este programa funciona de forma tal que cuando se active una salida por medio de la interfaz gráfica, ARDUINO envíe un pulso constante por la salida activa a un relé para accionarlo y realizar control mecánico ya sea accionar enchufes eléctricos, encendido u apagado de Iluminarias, etc.

A continuación la figura 6-5 muestra el accionamiento de los relés tras interactuar con los botones por medio de la interfaz gráfica desde el navegador WEB.

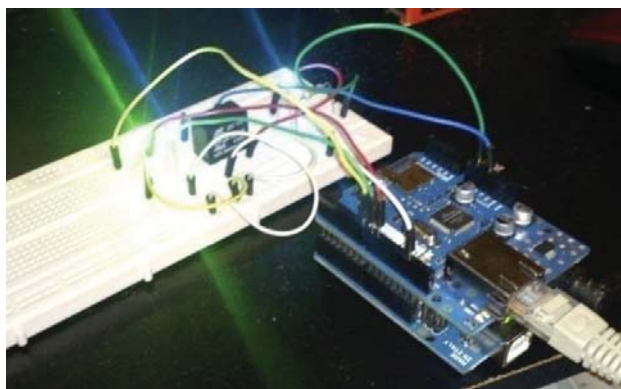


Fig. 6-5: Accionamiento de relé a través de ETHERNET.

Cabe destacar que este circuito puede usarse para fines Domóticos al poder controlar varias salidas de forma remota, donde se puede decir que este dispositivo funcionaría como un PLC de menor tamaño con la particularidad de ser comandado vía ETHERNET y de forma remota.

Para mejorar la experiencia, se puede reemplazar el SHIELD ETHERNET por otro SHIELD WIFI haciendo todo esto inalámbrico donde sólo se necesite energizar el ARDUINO y en donde se sigue el mismo procedimiento antes mencionado.

Un uso real que se le puede dar a este circuito es de energizar o desenergizar enchufes, con ello podemos apostar a reducir consumos extra de equipos conectados a la red, como lo son los televisores, equipos de sonido, decodificadores que normalmente quedan encendidos y que funcionan de forma constante, entre otros, como también tener el control de iluminarias de hogar u oficinas.

## 6.2 Sistema NEAR FIELD COMMUNICATION (NFC)

Este es un sistema de transmisión inalámbrico de corto alcance que funciona por proximidad el cual viene integrado en dispositivos móviles de última generación, en donde es posible la interconexión entre dispositivos y etiquetas NFC para el envío rápido de pocos bits de información, el cual a su vez permite la identificación y la validación del usuario. Funciona de forma similar a la transmisión de datos Bluetooth sin embargo ofrece mayores prestaciones al utilizar la capacidad de cómputo de los equipos móviles.



Los Estándares de NFC cubren protocolos de comunicación y formato de intercambio de datos basado en ISO 14443, RFID un identificador por radio frecuencia. Esta tecnología fue desarrollada en 2004 por las empresas Sony, Nokia y Philips.

### 6.2.1 Funcionamiento

El sistema NFC se comunica mediante la inducción en un capo magnético, donde dos antenas de espira son colocadas en los respectivos campos cercanos. Trabaja dentro de las bandas de 13.56 MHz, lo cual genera que no aplique ninguna restricción o requiera una licencia para su uso.

Existen dos modos de funcionamiento, en donde todos los dispositivos del estándar NFCIP-1 deben soportarlos, el modo Activo y modo Pasivo

- Modo Activo, en donde ambos dispositivos generan su propio campo magnético para la transmisión de datos.
- Modo Pasivo, solo uno de los dispositivos genera su campo magnético, y donde el otro dispositivo se aprovecha de la modulación para la comunicación y transferencia de datos, llamadas comúnmente NFC Tags, Tags, tarjetas entre otros.

Dependiendo del tipo de modo de funcionamiento, va a depender la distancia máxima a la que los dispositivos deben encontrarse para que exista la comunicación. En donde puede ir desde 0 a 20 cm de distancia, donde por su corto alcance nos proporciona mayor seguridad de los intercambios y conexiones accidentales, cabe destacar que existen dispositivos activos que puedes llegar a superar los 100 mt. de distancia. [13]

A continuación en la figura 6-6 se presentan distintos tipos de comunicación NFC.

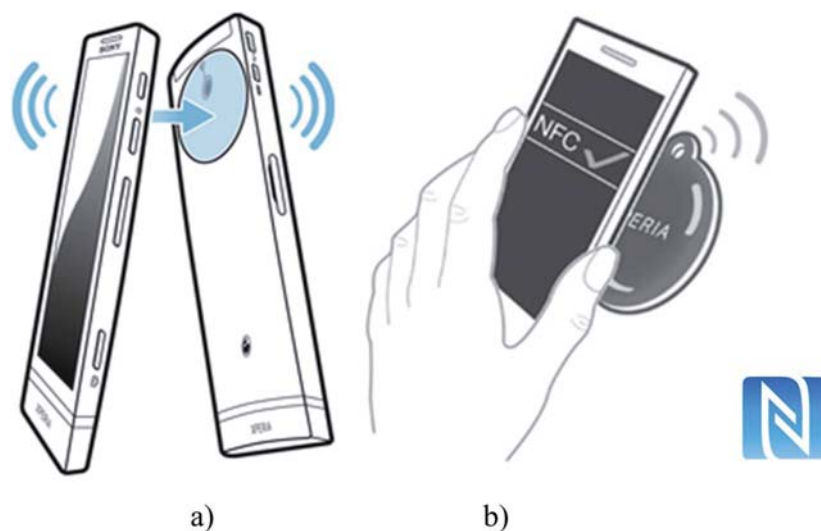


Fig. 6-6: a) Comunicación entre dispositivos NFC Activos, b) Comunicación entre dispositivos NFC Activo y Pasivo (NFC Tags).

El protocolo NFCIP-1 puede funcionar a diversas velocidades como 106, 212, 424 o 848 Kbits/s donde los equipos se pueden configurar para la velocidad deseada.

### 6.2.2 NFC Tags

Estos dispositivos del tipo pasivo, son los encargados de interactuar con los equipos móviles que traen incluida la tecnología NFC, los cuales son capaces de tener una identidad única, donde existen del tipo lectura que contienen información precargada no modificable y otros los cuales es posible cargarles información de contactos, direcciones, páginas web entre otros.

Las especificaciones técnicas de las NFC Tags en cuanto a almacenamiento van entre 4 y 32 Kbits, capacidad suficiente para almacenar una buena cantidad de datos, en donde por lo demás el equipo móvil es el encargado de procesar esta información y actuar, por lo que no es necesario tener capacidad mayor en estos dispositivos.

A continuación en la figura 6-7 se presentan dispositivos NFC Tags del tipo pasivo.



Fig. 6-7: NFC Tags tipo Pasivas.

Actualmente no es muy masivo el uso de estos dispositivos, donde se está investigando y mejorando esta tecnología, la cual a futuro se espera que reemplace a los actuales códigos de barra de productos, donde se puede cargar información extra en estos, ayudando el uso de inventarios, se está utilizando en la identificación de mascotas por medio de chips sub cutáneos, y los próximo que ya se habla es el pago por medio de este servicio NFC entre otros usos donde se ha ido instaurando. [14]

### 6.3 Control NFC y ARDUINO

Utilizaremos esta tecnología para el control de ARDUINO, para ello debemos contar con a lo menos dos dispositivos que contengan este sistema integrado, en este caso, un Smartphone de gama media - alta con tecnología NFC y NFC Tags (para el caso nuestro, dispositivo NFC tipo pasivo) y de la ayuda de algún software para realizar este propósito.

#### 6.3.1 Funcionamiento

Para el control de ARDUINO por medio de sistema NFC, como primer paso, este debe estar configurado y estar conectado vía cable de red al Router inalámbrico como se describió en el Capítulo 6 de este informe, posterior a esto, para que exista comunicación entre el dispositivo móvil y el microcontrolador, solo bastaba el pertenecer a la misma red que entrega el Router inalámbrico. Una vez ingresado el dispositivo móvil a la misma red que el microcontrolador, este tiene acceso por intermedio del Router inalámbrico al microcontrolador vía WIFI, en la imagen a continuación podemos ver el acceso al programa de ARDUINO a través de la dirección 192.168.0.100 por medio del dispositivo móvil el cual utiliza el sistema operativo Android.

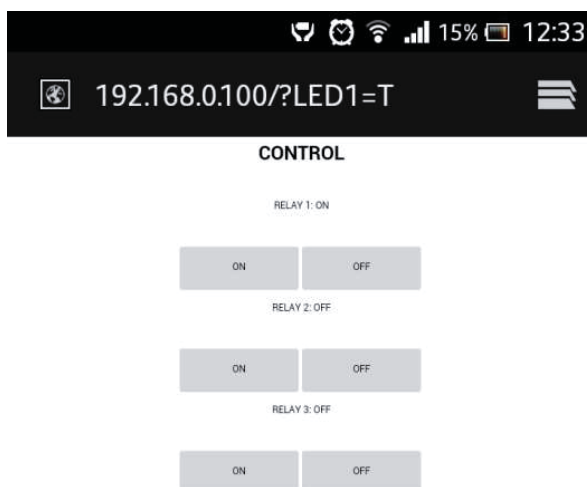


Fig. 6-8: Acceso a interfaz gráfica de ARDUINO desde dispositivo móvil.

Ya logrando esta conexión, podemos seguir con la programación de equipo móvil para interactuar con las NFC Tags, y controlar ARDUINO.

El principio de funcionamiento consiste en que al acercar nuestro dispositivo móvil a las NFC Tags, este reciba la información necesaria y automáticamente envíe datos directamente hacia ARDUINO y que este actúe, logrando comandarlo por medio de este sensor de proximidad. A su vez que en pantalla del Smartphone se cargue el programa de ARDUINO para el control total de este.

A continuación en la figura 6-9 se muestra la comunicación entre los dispositivos involucrados para el control de actuadores a través de sistema NFC.

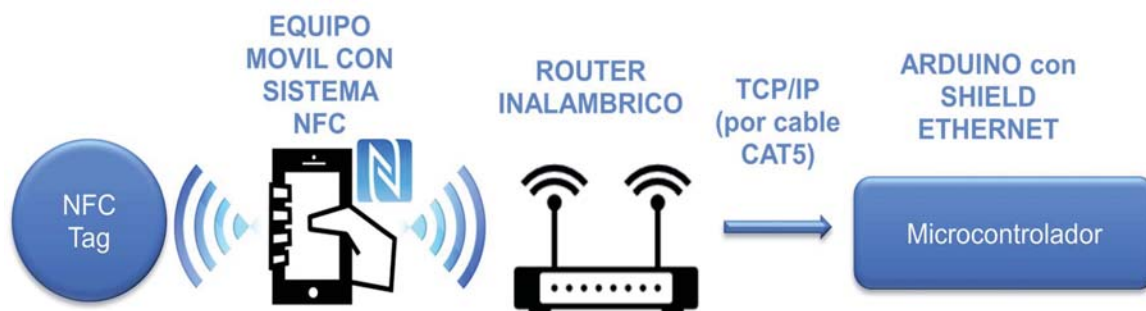


Fig. 6-9: Control de ARDUINO a través de sistema NFC.

Para hacer esto posible, por parte del sistema operativo Android, existe gran cantidad de programas gratuitos disponibles en su tienda de aplicaciones Play Store, que permiten el configurar las acciones de los dispositivos móviles al interactuar con otros dispositivos u accesorios que se le conecten, donde utilizaremos TRIGGER para lograr el propósito antes mencionado, el de comandar ARDUINO por sistema NFC.

### 6.3.2 Aplicación para control por sistema NFC

En este punto, para concluir con nuestro objetivo, utilizaremos la aplicación TRIGGER para Android, la cual permite que nuestro dispositivo móvil interactúe con su entorno, donde utiliza los recursos que presenta el equipo, ya sea Bluetooth, WIFI o para el caso nuestro el uso de NFC para actuar, la cual aparece de forma gratuita en la tienda de aplicaciones para Android.

Este programa permite ser configurado de forma tal que cada vez que acerquemos el sensor NFC de nuestro dispositivo móvil a alguna NFC Tag, este es capaz de leerla y realizar alguna acción programada, esto puede ser el activar los contenidos multimedia, reproducir música, activar alarmas, o a acceder a alguna dirección web, entre otras varias funciones que presenta la aplicación. [15]

Para términos legales, esta aplicación a utilizar es solo para fines de estudio con el fin de optar al Título Universitario, donde queda pendiente para futuros proyectos el desarrollo de una aplicación por intermedio de programación JAVA o algún otro tipo, para el control de acciones por tecnología NFC.

### 6.3.3 Programación de la aplicación

Para la configuración de esta aplicación, como primer paso es agregar el tipo de activador para que el programa actúe a través de las tareas que se le configuren, en la

figura 6-10 a continuación se puede apreciar los tipos de activadores posibles que ofrece la aplicación.

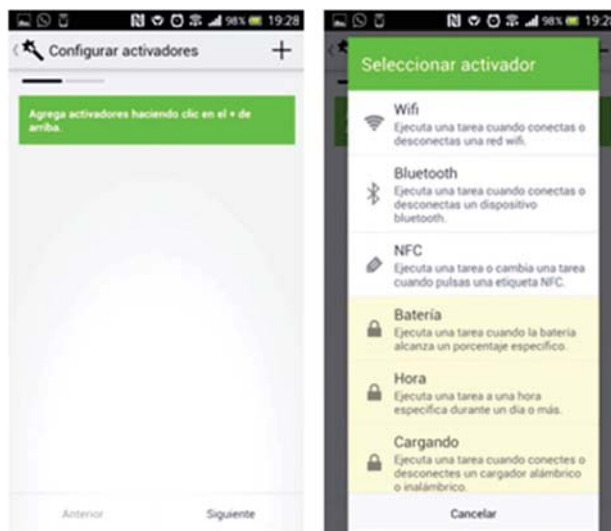


Fig. 6-10: Configuración de Aplicación TRIGGER para control a través de sistema NFC.

En ella se aprecia que la aplicación permite la activación de tareas cuando se activa WIFI, Bluetooth o sistema NFC los cuales vienen activos en la versión gratuita del software, y otros 3 activadores pertenecientes a la versión de pago, para el caso nuestro utilizaremos el activador NFC.

Una vez seleccionado el tipo de activación se debe seleccionar la acción a realizar, para el caso nuestro la de cargar la dirección IP de nuestro ARDUINO conectado al Router Inalámbrico, dirección IP 192.168.0.100, como se muestra en la figura 6-11 a continuación:

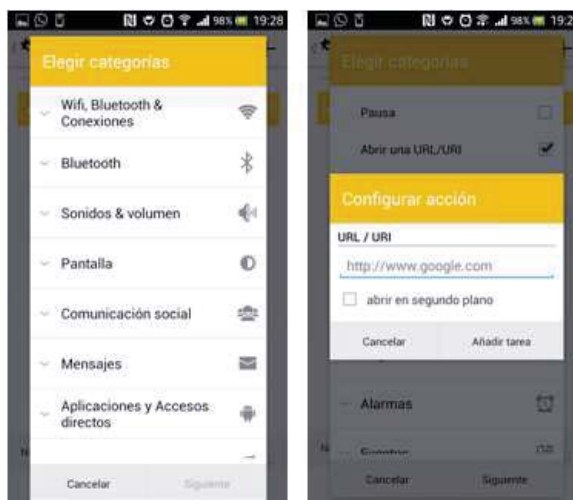


Fig. 6-11: Configuración de Aplicación TRIGGER para acceso a interfaz gráfica de ARDUINO.

Donde se selecciona el menú Aplicaciones y Accesos directos y luego acceder a una URL/URI donde se debe ingresar la dirección IP antes mencionada, con esto cada vez que aproximemos nuestro equipo móvil a una NFC Tag, este cargará automáticamente la dirección ingresada, accediendo por intermedio del navegador Web del equipo móvil a la interfaz gráfica de ARDUINO para el control de actuadores.

Para hacer más eficiente y automatizado este programa, a la dirección IP mencionada anteriormente se le agregará información complementaria, la que permitirá la activación y desactivación inmediata de las salidas que controla ARDUINO, vale decir, la nueva dirección será `192.168.0.1/?LED1=T`, donde el comando `?LED1=T` significa que LED1 es igual a verdadero, encendiendo este LED1 implicando la activación de esa salida (se está actuando directamente en el código de programación de ARDUINO, código de programa adjunto en el Anexo de este informe) para el control de actuadores.

Esta aplicación también cuenta con la capacidad de hacer ciclos de trabajo, donde al acercar nuestro dispositivo móvil a un NFC Tags más de una vez, este realice otras acciones. Se utilizará este beneficio para configurar las acciones de la aplicación en donde al aproximar nuestro equipo móvil una primera vez a un NFC Tags, este active la salida de ARDUINO, y al aproximarse una segunda vez esta salida sea desactivada, como se muestra en la figura 6-12 a continuación:

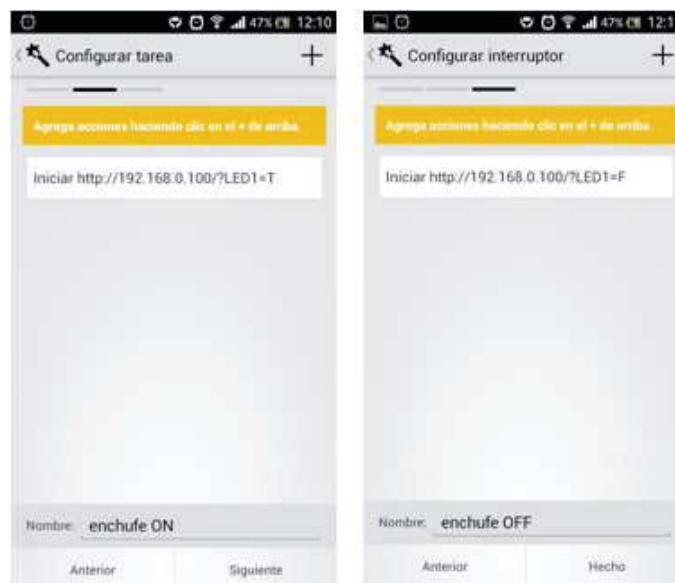


Fig. 6-12: Configuración para control de enchufe a través de sistema NFC.

Esta aplicación permite que dentro de un activador se puedan configurar varias acciones, por lo que si disponemos de más NFC Tags, podemos ingresar otras funciones y así comandar varios actuadores a través del microcontrolador ARDUINO. Ya que cada una de las NFC Tags contiene autentificaciones diferentes, permiten que el sensor NFC del equipo móvil lea cada una de ellas y realice acciones distintas.

Para nuestro proyecto se pretende hacer control Domótico, esto enfocado a controlar Enchufes eléctrico domiciliario y hacer control de iluminación, el programa final dentro de la aplicación se muestra en la figura 6-13 a continuación:





Fig. 6-13: Configuración final de la aplicación TRIGGER para el control de actuadores desde ARDUINO por sistema NFC.

En la Figura 6-13 podemos ver el estado en que se encuentran cada actuador que se configuró a través de la aplicación móvil, la cual además nos permite cambiar los estados de los actuadores al utilizar el sistema de proximidad NFC, con la ventaja de que esta aplicación nos permite ejecutar acciones en segundo plano, sin necesidad de mantener tanto pantalla como aplicación abierta.

A continuación en la figura 6-14 se presenta diagrama de bloque resumen para el control de actuadores a través de SHIELD ETHERNET para ARDUINO.

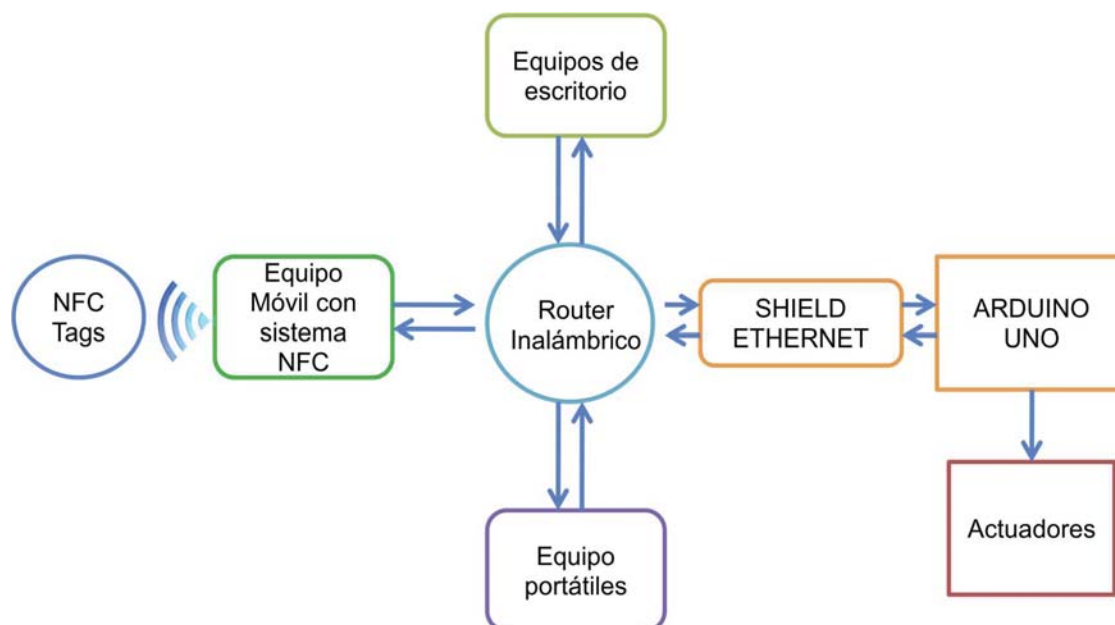


Fig. 6-14: Diagrama de bloques para el control de actuadores a través de ETHERNET.

De la figura 6-14, tenemos conectado ARDUINO a nuestro Router a través de ETHERNET, podemos acceder a él y controlarlo de forma directa y sencilla, ya sea desde un equipo de escritorio, a través de dispositivos portátiles o móviles por medio de configuración de redes donde además podemos ampliar las prestaciones de este sistema utilizando comunicación por VPN de forma remota para hacer control de los actuadores (desde cualquier lugar siempre y cuando tengamos salida a Internet).

## CAPÍTULO 7

### EVALUACIÓN ECONÓMICA

La evaluación económica es empleada con el fin de determinar la rentabilidad de un proyecto por medio de uno o más indicadores, que nos ayudan a la toma de decisiones. Estos indicadores son los que se utilizan como criterio de resolución, para ello se determinará el indicador VAN (Valor Actual Neto) el cual nos indicará si es o no rentable nuestro proyecto.

#### 7.1 Definición del producto/servicio

Nuestro producto/servicio consiste en un sistema domótico para el control remoto de acciones donde se necesite controlar a distancia apertura o cierre de ventanas así como control de persianas entre otros usos dedicados posibles, utilizando la misma red del hogar u oficina, donde se utilizará tecnología NFC, tecnología de proximidad para mayores prestaciones del servicio.

El servicio que se ofrece presenta como sistema básico el control de 3 actuadores, que pueden ser comandados por cualquier dispositivo presente en la red que se genere tras la instalación del servicio, ya sea computadores, notebook, equipos móviles y donde se entregará un equipo móvil con tecnología NFC para hacer el control por medio de su sistema de proximidad.

#### 7.2 Estudio de mercado

El estudio de mercado se realiza para determinar la viabilidad del proyecto, donde es de total importancia poder determinar y analizar los distintos mercados presentes que influirán de forma directa en el proyecto.

Se analizarán 3 mercados: Proveedor, Competencia y Consumidores.

##### 7.2.1 Mercado de Proveedores

Se encargan de acercarnos las materias primas necesarias para el desarrollo de nuestro proyecto, los cuales van a ser determinantes debido al fuerte impacto que pueden llegar a generar, ya sea por valores que ofrecen al mercado, como la calidad de sus productos, posibles importaciones que se deban realizar y los tiempos de envío, entre otros factores.

En Tabla 7-1 se aprecia proveedores de materiales utilizados para el proyecto.

Tabla 7-1: Proveedores de materiales del proyecto.

Ítem	Proveedores
SHIELDS, ARDUINO y Relés	Olimex Ingeniería MCI Ltda.
Componentes adicionales	Globalchile Electrónica . Valparaíso.
Herramientas varias	Globalchile Electrónica Valparaíso.
SmartPhone con tecnología NFC	Compañías telefónicas –Grandes tiendas.
NFC Tags	SonyChile, DealExtreme.

### 7.2.2 Mercado de Competencia

Hoy en día existen varias empresas que han empezado a abarcar estos mercados, como son Domotika o Smarthomechile ofreciendo servicios similares a elevados costos, muchos de ellos sobrepasando el \$1.000.000 de pesos. Si bien ofrecen el control de distintos dispositivos no presentan la tecnología NFC.

Debemos ser capaces de ser una buena competencia, aprovechando que varios de nuestros materiales utilizados son de fácil acceso y económico, disminuyendo el costo total para la producción de nuestros productos, dándole énfasis a la tecnología NFC y a su seguridad y fácil acceso.

### 7.2.3 Mercado de Consumidores

Este mercado de consumidores está compuesto por personas, empresas u organizaciones que requieran servicios especializados para satisfacer sus distintas necesidades propias, y que cuentan con los recursos para adquirir estos servicios.

Este proyecto apunta a facilitar tareas que realizan personas comunes debido a la conexión remota que presenta nuestro sistema, por lo que es adaptable a oficinas como el propio hogar, al poseer un sistema seguro, se puede controlar accesos a recintos o el mismo hogar.

## 7.3 Análisis de Costos

Este análisis se compone de inversión inicial, costos de operación y los flujos de caja, con los cuales se determinarán los indicadores de rentabilidad de proyecto, Valor Actual Neto (VAN) y Tasa Interna de Retorno (TIR).

### 7.3.1 Inversión inicial

Este es el capital con que se debe disponer para el desarrollo del proyecto, para la obtención de las materias primas e insumos que se utilizarán. Se presentará en tres puntos, Activos Fijos, Activos Nominales y Capital de Trabajo.

### 7.3.2 Activos Fijos

Estos permiten el completo desarrollo del proyecto. En la Tabla 7-2 se muestra los activos fijos utilizados:

Tabla 7-2: Activos Fijos.

Ítem	Cantidad	Costo
Computador Escritorio	1	350.000
Multitester	1	15.000
Kit Herramientas	1	7.000
Cautin 25W	1	6.000
Total	\$	378.000

### 7.3.3 Activos Nominales

Estos son bienes intangibles que están en estrecha relación con el desarrollo del proyecto, para nuestro caso se tomara el sueldo del Ingeniero a cargo del desarrollo del producto. A continuación en la Tabla 7-3 se muestra el activo nominal utilizado:

Tabla 7-3: Activos nominales.

Ítem	Costo
Sueldo Ingeniero a cargo	\$ 700.000

### 7.3.4 Inversión de Capital de Trabajo

En la Tabla 7-4 se muestran los ítems y los costos asociados para el desarrollo del producto final.

Tabla 7-4: Costos asociados.

Ítem	Cantidad	Costo
SmartPhone con tecnología NFC	1	150000
NFC Tags	3	3.000
ARDUINO UNO	1	15.000
SHIELD ETHERNET	1	30.000
Relés	3	4.500
Transformador AC-DC 1.5 - 12 V	1	3.800
cable USB	1	800
ProtoBoard	1	4.000
Router Inalámbrico	1	15.000
<b>Total</b>	<b>\$</b>	<b>226.100</b>

Para poder desarrollar este servicio, se debe contar con capital de trabajo, el cual debe permitirnos fabricar el producto para su posterior comercialización, donde para el primer año se espera vender 20 servicios, dándonos un capital de trabajo de \$.4.222.000

A continuación, en la Tabla 7-5 se entrega un resumen del costo total de inversión para el inicio del proyecto, donde se integra Activos Fijos, Nominales y Capital de Trabajo:

Tabla 7-5: Inversión de capital.

Ítem	Costo
Activos Fijos	378.000
Activos Nominales	700.000
Capital de Trabajo	4.222.000
<b>Total</b>	<b>\$ 5.300.000</b>

### 7.3.5 Financiamiento

Para el financiamiento de la inversión inicial de este proyecto, se utilizará la solicitud de un crédito bancario, donde se hará la simulación del crédito, donde nos indicará tasa y montos mensuales a cancelar como se indica en la figura 7-1 a continuación.

<b>Resultado de simulación</b>				
Esta simulación no constituye una aprobación formal, por lo tanto es sólo referencial.				
✦ Características de Crédito:				
<b>Valor Cuota:</b>		<b>\$157.290</b>		
Tasa interés Mensual:	1,22% (14,64% anual)			
Monto Solicitado:	<b>\$5.500.000</b>			
Monto Bruto del Crédito <sup>1</sup> :	<b>\$5.669.923</b>			
Plazo del Crédito:	<b>48 Meses</b>			
✦ GASTOS				
	Valor Total	✦ Seguros Voluntarios	Valor Mensual <sup>2</sup>	Valor total
Notario:	\$3.000	Desgravamen:	\$3.012	\$144.563
Impuestos:	\$22.453	<b>Total:</b>	<b>\$3.012</b>	<b>\$144.563</b>
<b>Total:</b>	<b>\$25.453</b>			
<b>Carga Anual Equivalente:</b>		<b>16,5%</b>		
Carga Anual Equivalente: Indicador que incluye los intereses, gastos y seguros asociados al crédito expresados en forma anual.				

Fig. 7-1: Simulación de crédito bancario.

### 7.3.6 Depreciación

Es la desvalorización de un activo, a una vida útil dada, para su posterior renovación o venta de estos activos, para el caso nuestro la depreciación de activos fijos se muestra en la tabla presentada más abajo.

$$\text{Depreciación Lineal} = \frac{ci - vr}{n} [16]$$

Donde:

$ci$  = Costo inicial en pesos

$vr$  = Valor residual

$n$  = Vida útil en años

Cabe destacar que para el cálculo de depreciación, se da un valor residual 0, por no percibir ingresos por los activos al final de su vida útil. A continuación en la Tabla 7-6 se muestra la depreciación de los activos fijos:

Tabla 7-6: Depreciación de activos.

Ítem	Vida Útil	Costo	Depreciación anual
Computador Escritorio	4	350000	87500

### 7.3.7 Precio de Venta

Para el cálculo de precio de venta, este viene dado por la siguiente fórmula:

$$p = \frac{cv \times cf}{q} \quad [17]$$

Donde:

$p$  = Precio unitario

$cv$  = Costo variable

$cf$  = Costo fijo

$q$  = Cantidad a producir

Reemplazando con los datos obtenidos anteriormente, tenemos un valor unitario de \$226.100 pesos como mínimo para que proyecto sea rentable, por lo que se asigna un valor de venta de \$750.000 pesos por el servicio, el cual incluye equipo móvil con tecnología NFC, 3 SmartTags, ARDUINO UNO, SHIELD ETHERNET para ARDUINO, 3 Relés, Router Inalámbrico (ver Tabla 7-4) más la instalación, recordar que si el cliente desea aumentar el número de actuadores se debe agregar costos extra por servicio.

## 7.4 Flujo de caja

En este punto, se sabe si el proyecto es viable o no por medio de los indicadores económicos VAN y TIR. Para el cálculo de estos, primero es necesario determinar la tasa de descuento para el proyecto, esto es, sumarle a la tasa anual del préstamo (14.64%) un porcentaje de interés por parte del inversionista (que tomaremos un 10%), por lo que se tiene una tasa de descuento de un 25% aproximado.

Para efectos de cálculo de VAN se tiene:

$$VAN = -I + \sum_{j=1}^n \frac{Fc}{(1+r)^n} \quad [18]$$

Donde:

$I$  = Inversión inicial

$Fc$  = Flujos de caja

$r$  = Tasa de descuento



$n$  = Número de años

Criterio decisión de indicador VAN

$VAN > 0$	Proyecto es rentable
$VAN = 0$	Solo se recupera la inversión inicial, criterio de decisión indiferente
$VAN < 0$	Proyecto no rentable

En la Tabla 8 -7 a continuación se determina el VAN y TIR del proyecto, en la que se determina que el VAN es \$ 11.861.034,78 y el TIR es 44% lo que nos indica que el proyecto es económicamente rentable y realizable.

Tabla 7-7: Resumen cálculo VAN y TIR.

	AÑOS										
	0	1	2	3	4						
CANTIDAD DE VENTAS		50	70	90	120						
INGRESOS POR VENTAS		14000000	19600000	25200000	33600000						
COSTOS FIJOS		8400000	8400000	8400000	8400000						
COSTOS VARIABLES		4250000	4600000	4860000	5160000						
DEPRECIACION		87500	87500	87500	87500						
UTILIDAD BRUTA		1262500	6512500	11852500	19952500						
IMPUESTO		201576	1039810	1892416	3185694						
UTILIDAD NETA		1060924	5472690	9960084	16766806						
AMORTIZACION DE LA INVERSION		1887480	1887480	1887480	1887480						
INVERSION INICIAL	7253000										
FLUJO DE CAJA	- 7253000	-826556	3585210	8072604	14879326						
<table border="1"> <tr> <td>Tasa de descuento</td> <td>25%</td> </tr> <tr> <td>VAN</td> <td>\$ 11.861.034,78</td> </tr> <tr> <td>TIR</td> <td>44%</td> </tr> </table>						Tasa de descuento	25%	VAN	\$ 11.861.034,78	TIR	44%
Tasa de descuento	25%										
VAN	\$ 11.861.034,78										
TIR	44%										

## CONCLUSIONES

De los resultados obtenidos con ambos dispositivos, se llega a la conclusión de que nos conviene trabajar con un dispositivo creado especialmente para operar con ARDUINO como lo es el SHIELD ETHERNET, no así el dispositivo MOXA, el cual es de tipo Universal, en donde se requiere bastantes configuraciones para comunicarse con ARDUINO y a esto sumarle que se debe agregar un circuito que adapte y establezca la comunicación TTL/CMOS entre estos, lo que ocasionó más de algún problema. Al utilizar el SHIELD ETHERNET, la comunicación entre los dispositivos es instantánea, trabajando como una sola máquina, por lo que es prácticamente seguro su buen funcionamiento y nos hace inclinarnos a trabajar con este dispositivo.

Hoy en día podemos decir que la tecnología apunta a una sola cosa, la automatización, el control total de acciones en el entorno que nos rodea. Un sistema que sea capaz de esto a través de las prestaciones que nos entregan las tarjetas de desarrollo como lo son ARDUINO y sus SHIELDS a un bajo costo se puede concretar, donde se puede tener el control de varias acciones con una sola placa con acceso rápido y remoto.

Debido al déficit energético en que nos encontramos actualmente, tener un sistema que ayude en este punto es fundamental, por lo que el sistema domótico presentado en este proyecto es interesante, ya que se puede llegar a tener un buen ahorro de energía al administrarla correctamente por medio de control de enchufes, luces de hogares y oficinas, utilizando dispositivos pasivos como son las NFC Tags como también estas tarjetas de desarrollo como ARDUINO, de bajo consumo energético.

El estudio de nuevos desarrollos tecnológicos avanza día a día, por lo que se hace interesante el trabajar con tecnologías que aún no llegan a mostrar todo su potencial en la actualidad, como es la tecnología NFC, que probablemente en un futuro no muy lejano la veamos de forma más habitual en más dispositivos de uso tanto doméstico como de la vida cotidiana. El haber logrado un control de actuadores con esta tecnología es solo una pequeña muestra de las posibles prestaciones que puede llegar a tener.

Podemos decir que el proyecto, por medio de análisis estadísticos, en términos generales es totalmente viable, ya que los indicadores VAN y TIR demostraron rentabilidad.

Al hacer un estudio de mercado simple, podemos determinar cómo dirigimos y captar la atención de los clientes, en donde podemos darnos cuenta de los valores elevados que se cobran por servicios de domótica ya sea casera como para oficinas, donde se puede aprovechar de entrar al mercado con precios más accesibles.

Para un mejor análisis de mercado se pueden utilizar herramientas como “FODA” u “OCEANO AZUL” entre otras, que nos ayudan para determinar las reales ventajas y posibilidades de este proyecto y su inclusión en el mercado.

## BIBLIOGRAFIA

- [1] ARDUINO, « ARDUINO Board UNO » [en línea] Available:  
<http://arduino.cc/en/Main/ArduinoBoardUno#.UxXsSv15PrM>
- [2] ARDUINO, « ARDUINO Ethernet Shield » [en línea] Available:  
<http://arduino.cc/en/Main/ArduinoEthernetShield>
- [3] J. Ruiz Gutierrez, « ARDUINO + Ethernet Shield » [en línea] Available:  
[http://unicarlos.com/\\_ARDUINO/Arduino%20+%20Ethernet%20Shield%20\(1\).pdf](http://unicarlos.com/_ARDUINO/Arduino%20+%20Ethernet%20Shield%20(1).pdf)
- [4] Adafruit Industries, « Adafruit motor/Stepper/Servo Shield » [en línea] Available:  
<http://adafruit.com/products/81#Description>
- [5] Wikipedia, « Motor paso a paso » [en línea] Available:  
[http://es.wikipedia.org/wiki/Motor\\_paso\\_a\\_paso](http://es.wikipedia.org/wiki/Motor_paso_a_paso)
- [6] Wikipedia, « Servo Motor » [en línea] Available:  
<http://es.wikipedia.org/wiki/Servomotor>
- [7] Kioskea, « Ethernet » [en línea] Available:  
<http://es.kioskea.net/contents/672-ethernet>
- [8] MOXA, « NPort 5110 » [en línea] Available:  
[http://www.moxa.com/product/nport\\_5110.htm](http://www.moxa.com/product/nport_5110.htm)
- [9] ARDUINO, « Serial » [en línea] Available:  
<http://arduino.cc/es/Reference/serial>
- [10] Alldatasheet, « circuito integrado MAX232N » [en línea] Available:  
<http://pdf1.alldatasheet.es/datasheet-pdf/view/27238/TI/MAX232DE4.html>
- [11] Wikipedi, « Transmission Control Potocol » [en línea] Available:  
[http://en.wikipedia.org/wiki/Transmission\\_Control\\_Protocol](http://en.wikipedia.org/wiki/Transmission_Control_Protocol)
- [12] ARDUINO, « Ethernet » [en línea] Available:  
<http://arduino.cc/en/reference/ethernet>
- [13] Revia UNAM, « NFC » [en línea] Available:  
<http://www.revista.unam.mx/vol.11/num3/art33/NFC.swf>
- [14] Wikipedia, « RFID » [en línea] Available:  
<http://es.wikipedia.org/wiki/FeliCa>
- [15] Google Play, « Trigger App » [en línea] Available:  
<https://play.google.com/store/apps/details?id=com.jwsoft.nfcactionlauncher&hl=es>
- [16] Meigs R. & Williams J. & Haka S. & Bettr M., 2000. Contabilidad. La base para decisiones gerenciales. Un décima edición. D Vinni editorial. Colombia. Capítulo 9, páginas 370-415.
- [17] Meigs R. & Williams J. & Haka S. & Bettr M., 2000. Contabilidad. La base para decisiones gerenciales. Un décima edición. D Vinni editorial. Colombia. Capítulo 13, páginas 547-662.
- [18] Meigs R. & Williams J. & Haka S. & Bettr M., 2000. Contabilidad. La base para decisiones gerenciales. Un décima edición. D Vinni editorial. Colombia. Capítulo 13, páginas 547-662.

## **APÉNDICE A**

### **CÓDIGO DE PROGRAMACIÓN PARA CONTROL DEL ACTUADORES ELECTROMECAÓNICOS**

## APÉNDICE A

### CÓDIGO DE PROGRAMACIÓN PARA CONTROL DE ACTUADORES ELECTROMECÁNICOS

#### A.1 Código para control de actuadores: Relés electromagnéticos

Este código permite el accionamiento de las salidas que dispone ARDUINO junto al SHIELD ETHERNET, generando niveles lógicos 1 y 0 en cada salida programada, estos pulsos son capaces de accionar las bobinas internas de relés generando el control del actuador. Por otra parte, este código está dividido en 2 partes, la del control de los actuadores y otra de diseño de una interfaz gráfica la cual se puede acceder a través de ETHERNET

```
#include <SPI.h>
#include <Ethernet.h>

//Declaración de la direcciones MAC e IP. También del puerto 80

byte mac[]={0xDE,0xAD,0xBE,0xEF,0xFE,0xED}; //MAC
IPAddress ip(192,168,0,100); //Dirección IP
EthernetServer servidor(80);
int PIN_LED1=8;
String readString1=String(30);
String state1=String(3);
int PIN_LED2=9;
String state2=String(3);
int PIN_LED3=7;
String state3=String(3);

void setup()
{
  Ethernet.begin(mac, ip); //Iniciamos con direcciones declaradas
  servidor.begin();
  pinMode(PIN_LED1,OUTPUT);
  digitalWrite(PIN_LED1,HIGH);
  state1="OFF";
  pinMode(PIN_LED2,OUTPUT);
  digitalWrite(PIN_LED2,HIGH);
  state2="OFF";
  pinMode(PIN_LED3,OUTPUT);
  digitalWrite(PIN_LED3,HIGH);
  state3="OFF";
}

void loop()
{
  EthernetClient cliente= servidor.available();
  if(cliente)
  {
    boolean lineaenblanco=true;
    while(cliente.connected()) //Cliente conectado
    {
      if(cliente.available())
      {
        char c=cliente.read();
        if(readString1.length()<30) //Leemos petición HTTP caracter a caracter
        {
```

```

readString1.concat@;          //Almacenar los caracteres en la variable readString
    }
if(c=='\n' && lineaenblanco) //Si la petición HTTP ha finalizado
{
int LED1 = readString1.indexOf("LED1=");
int LED2 = readString1.indexOf("LED=");
int LED3 = readString1.indexOf("LED3=");
if(readString1.substring(LED1,LED1+6)=="LED1=T")
{
digitalWrite(PIN_LED1,HIGH);
state1="ON";
} else if (readString1.substring(LED1,LED1+6)=="LED1=F")
{
digitalWrite(PIN_LED1,LOW);
state1="OFF";
}
if(readString1.substring(LED2,LED2+5)=="LED=T")
{
digitalWrite(PIN_LED2,HIGH);
state2="ON";
} else if (readString1.substring(LED2,LED2+5)=="LED=F")
{
digitalWrite(PIN_LED2,LOW);
state2="OFF";
}
if(readString1.substring(LED3,LED3+6)=="LED3=T")
{
digitalWrite(PIN_LED3,HIGH);
state3="ON";
} else if (readString1.substring(LED3,LED3+6)=="LED3=F")
{
digitalWrite(PIN_LED3,LOW);
state3="OFF";
}

//Cabecera HTTP estándar, diseño de interfaz gráfica

cliente.println("HTTP/1.1 200 OK");
cliente.println("Content-Type: text/html");
cliente.println();
//Página Web en HTML
cliente.println("<html>");
cliente.println("<head>");
cliente.println("<title>PROGRAMA CONTROL</title>");
cliente.println("</head>");
cliente.println("<body width=100% height=100%>");
cliente.println("<center>");
cliente.println("<h1>CONTROL</h1>");
cliente.print("<br><br>");
cliente.print("RELAY 1: ");
cliente.print(state1);
cliente.print("<br><br><br><br>");
cliente.println("<input      type=submit      value=ON      style=width:200px;height:75px
onClick=location.href='./?LED1=T\'>");
cliente.println("<input      type=submit      value=OFF     style=width:200px;height:75px
onClick=location.href='./?LED1=F\'>");
cliente.print("<br><br>");
cliente.print("RELAY 2: ");
cliente.print(state2);
cliente.print("<br><br><br><br>");
cliente.println("<input      type=submit      value=ON      style=width:200px;height:75px
onClick=location.href='./?LED=T\'>");
cliente.println("<input      type=submit      value=OFF     style=width:200px;height:75px
onClick=location.href='./?LED=F\'>");
cliente.print("<br><br>");
cliente.print("RELAY 3: ");
cliente.print(state3);
cliente.print("<br><br><br><br>");

```

```

cliente.println("<input      type=submit      value=ON      style=width:200px;height:75px
onClick=location.href='./?LED3=T\'>");
cliente.println("<input      type=submit      value=OFF     style=width:200px;height:75px
onClick=location.href='./?LED3=F\'>");
cliente.print("<br><br>");
cliente.println("</center>");
cliente.println("</body>");
cliente.println("</html>");
cliente.stop();                                //Cierre de conexión con el cliente
readString1="";
    }
  }
}
}
}

```

## A.2 Código para control de actuadores: Motores paso a paso y Servo-motor

Este código es un ejemplo de control para estos tipos de motores, el cual cuenta con las librerías <AFMotor.h> y <Servo.h> para controlarlas por medio de SHIELD de Motores para ARDUINO.

```

#include <AFMotor.h>;
#include <Servo.h>;
AF_Stepper motor1(45, 1);
AF_Stepper motor2(45, 2);
Servo servol;
int input=0;

void setup() {
Serial.begin(9600);
Serial.println("secuencia proyecto");
motor1.setSpeed(500);
motor2.setSpeed(500);
servol.attach(10);
}
void loop() {
input=Serial.read();
int i;
if(input=='a'){                                //compara entradas Serie en código
motor1.step(500, FORWARD, DOUBLE);           ASCII
delay(1000);
motor2.step(500, BACKWARD, DOUBLE);
delay(1000);
for (i=0; i<255; i++) {
servol.write(i);
delay(3);
}
for (i=255; i!=0; i--) {
servol.write(i-255);
delay(3);
}
motor2.step(500, FORWARD, DOUBLE);
motor1.step(500, BACKWARD, DOUBLE);
Serial.write("direccion 1-1 finalizada");
}
if(input=='b'){
motor1.step(1000, FORWARD, DOUBLE);
delay(1000);
motor2.step(500, BACKWARD, DOUBLE);
delay(1000);
for (i=0; i<255; i++) {
servol.write(i);
delay(3);
}
}
}
}

```

```
for (i=255; i!=0; i--) {
  servol.write(i-255);
  delay(3);
}
motor2.step(500, FORWARD, DOUBLE);
motor1.step(1000, BACKWARD, DOUBLE);
Serial.write("direccion 2-1 finalizada");
}
}
```



## **APÉNDICE B**

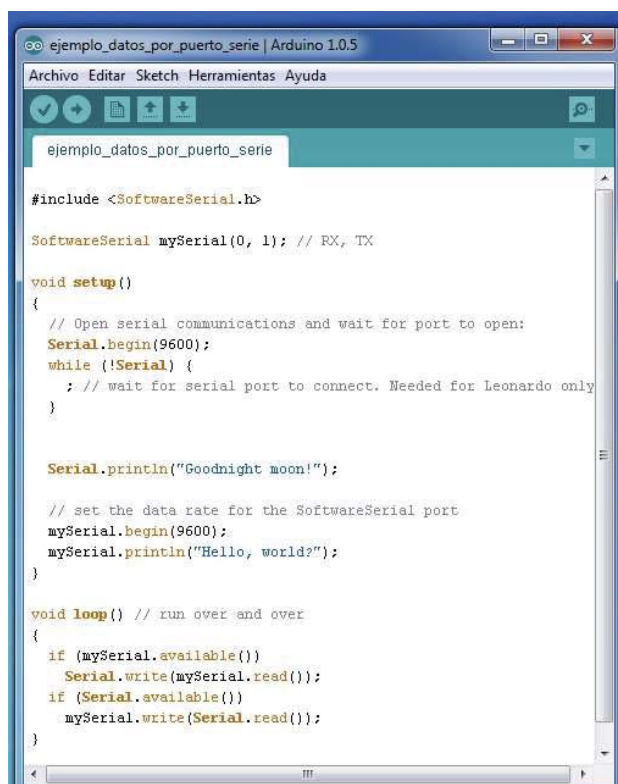
### **PRUEBAS PARA COMUNICACIÓN ENTRE DISPOSITIVOS MOXA – CIRCUITO ADAPTADOR – ARDUINO UNO**

## APÉNDICE B

### PRUEBAS PARA COMUNICACIÓN ENTRE DISPOSITIVOS MOXA – CIRCUITO ADAPTADOR – ARDUINO UNO

#### B.1 Comprobación de funcionamiento del circuito adaptador

Para verificar el correcto funcionamiento de este circuito, primero que todo se debe mencionar que ARDUINO UNO, dentro de las librerías de su software, cuenta con “SoftwareSerial.h” que contiene ejemplos para hacer estos chequeos, donde cargando el programa al micro-controlador, este al arrancar el programa, envía un texto por el puerto Serie.



```
ejemplo_datos_por_puerto_serie | Arduino 1.0.5
Archivo Editar Sketch Herramientas Ayuda
ejemplo_datos_por_puerto_serie

#include <SoftwareSerial.h>

SoftwareSerial mySerial(0, 1); // RX, TX

void setup()
{
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }

  Serial.println("Goodnight moon!");

  // set the data rate for the SoftwareSerial port
  mySerial.begin(9600);
  mySerial.println("Hello, world?");
}

void loop() // run over and over
{
  if (mySerial.available())
    Serial.write(mySerial.read());
  if (Serial.available())
    mySerial.write(Serial.read());
}
```

Fig. B 1: Ejemplo de envío de datos por puerto serie.

Para realizar esto, debemos tener un cable Serie y un software para hacer las pruebas de conexión entre dispositivos, las pruebas que se ejecutaran serán las siguientes:

Envío/recibo de datos Serie entre Ordenador – circuito convertidor – ARDUINO UNO

Para el envío de datos Serie se utilizará HYPERTERMINAL que viene por defecto en Windows XP, y que se puede descargar gratuitamente desde internet para sistemas operativos más actuales como son Windows 7 y 8.

Este programa debe ser configurado para trabajar con los mismos parámetros de trabaja nuestro ARDUINO, por lo que se debe asignar los valores como se muestra en la siguiente imagen:

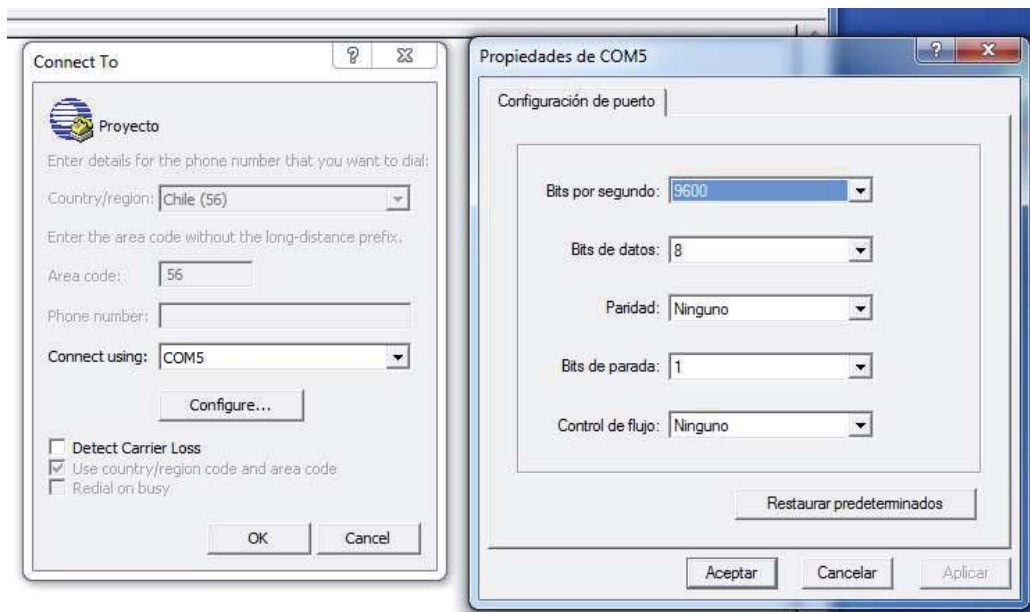


Fig. B 2: Configuración Serie de software HYPERTERMINAL.

De no contar con un puerto serie por parte del ordenador, existen dispositivos trabajan como puertos virtuales serie (puertos COM ), convirtiendo USB a puerto serie.



Fig. B 3: Conversor USB a RS232.

Una vez configurado en programa HYPERTERMINAL se enciende nuestro Arduino, el cual parte enviando información hacia nuestro ordenador que deberá aparecer en pantalla, lo cual nos demuestra que existe una correcta comunicación serie con Arduino

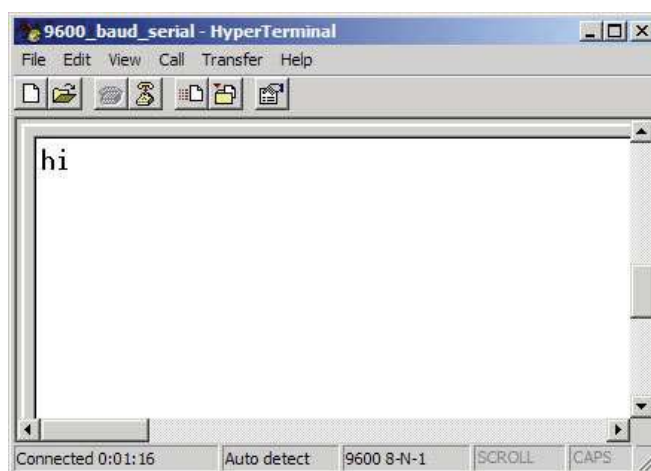


Fig. B 4: Envío de datos desde Arduino a HYPERTERMINAL.

## B.2 Prueba envío de datos vía TCP/IP por MOXA NPort 5110 hacia ARDUINO

Ya realizada la comunicación Serie, sólo nos queda hacer las conexiones entre dispositivo MOXA, el circuito adaptador y ARDUINO UNO, haciendo que la plataforma ARDUINO UNO realice control vía ETHERNET de actuadores electromecánicos.

Para hacer la correspondiente prueba de comunicación, al micro-controlador se le cargo previamente el Sketch para el control de Motores señalado en el APENDICE A, sección A.2 y el dispositivo MOXA se configuró en modo Servidor TCP/IP con IP: 192.168.127.254 y puerto de comunicación 4001.

Se utilizó el programa Comm Operator versión trial, el cual nos permite hacer envío de datos a través de protocolos TCP/IP hacia nuestro dispositivo MOXA el cual a su vez logra comunicarse con ARDUINO UNO, comandar los actuadores conectados al SHIELD de Motores y este entregándonos información de vuelta, completando la comunicación entre los dispositivos.

La figura a continuación nos muestra la comunicación entre los dispositivos, donde es posible apreciar que se envía un carácter en código ASCII (Send: (01:12:15) b) a través del software Comm Operator (configurado en modo Cliente) hacia dispositivo MOXA y que a su vez muestra que recibe información desde el micro-controlador ARDUINO señalando: “direccion 2-1 finalizada”

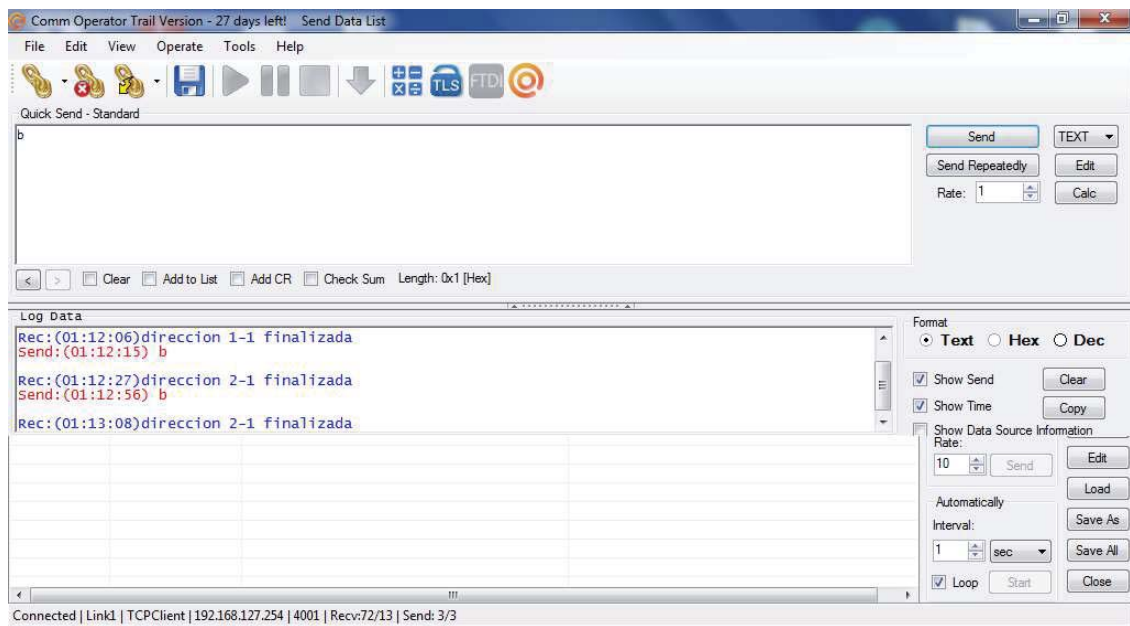


Fig. B 5: Comunicación entre Moxa y ARDUINO desde programa Comm Operator