UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE TECNOLOGÍAS

DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN.

CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA

TRABAJO DE GRADUACIÓN PARA LA OBTENCIÓN DEL TÍTULO DE:

TECNÓLOGO EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA

TEMA: "IMPLEMENTACIÓN DE 4 MÓDULOS PARA PRÁCTICAS DE COMUNICACIÓN ETHERNET EMPLEANDO ARDUINO MEGA PARA EL LABORATORIO DE INSTRUMENTACIÓN VIRTUAL DE LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS".

AUTOR: ÑACATA OÑA, MAICOL DANIEL

DIRECTOR: Ing. Pilatasig, Pablo.

LATACUNGA

2015

CERTIFICADO

Certifico que el presente Trabajo de Graduación fue realizado en su totalidad por el Sr. **ÑACATA OÑA MAICOL DANIEL**, como requerimiento parcial para la obtención del título de **TECNÓLOGO EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y AVIONICA.**

SR. ING. PABLO PILATASIG
DIRECTOR DEL TRABAJO DE GRADO

Latacunga, Junio 2015.

AUTORÍA DE RESPONSABILIDAD

Yo, Ñacata Oña Maicol Daniel

DECLARO QUE:

El trabajo de grado denominado **"IMPLEMENTACIÓN DE 4 MÓDULOS PARA PRÁCTICAS DE COMUNICACIÓN ETHERNET EMPLEANDO ARDUINO MEGA PARA EL LABORATORIO DE INSTRUMENTACIÓN VIRTUAL DE LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS"**, ha sido desarrollado en base a una investigación científica exhaustiva, respetando derechos intelectuales de terceros conforme las citas constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente, este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico de trabajo de grado en mención.

Latacunga, Junio 2015.

_____

Maicol Daniel Ñacata Oña.

C.I: 1722719505

DEDICATORIA

A Dios, por permitirme llegar a este momento tan importante en mi vida. Por los triunfos y los momentos difíciles que me han enseñado a valorarlo cada día más, a mi madre por ser la persona que me ha acompañado durante todo mi trayecto estudiantil y de mi vida, a mis familiares quienes han velado por mi durante este arduo camino para convertirme en un profesional. A mi padre quien con sus consejos ha sabido guiarme para culminar mi carrera profesional. A mis amigos, que gracias al equipo que formamos logramos llegar hasta el final del camino. A mis profesores, gracias por su tiempo, por su apoyo así como por la sabiduría que me trasmitieron en el desarrollo de mi formación profesional

Maicol

AGRADECIMIENTO

Agradezco a Dios por protegerme durante todo mi camino y darme fuerzas para superar obstáculos y dificultades a lo largo de toda mi vida.

A mis padres, que con su demostración de unos padres ejemplares me han enseñado a no desfallecer ni rendirme ante nada y siempre perseverar a través de sus sabios consejos.

Al Ing. Pablo Pilatasig, director del trabajo de graduación, por su valiosa guía y asesoramiento a la realización de la misma

Maicol

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente trabajo de graduación realiza ejemplos donde explican cómo realizar la comunicación Ethernet utilizando la placa Arduino Mega 2560 y el shield Ethernet, el shield se comunica con la placa Arduino Mega 2560 a través de SPI y con el computador mediante un cable de red directo. Se crearon páginas web para el control de encendido/apagado de leds, una página para un led conectado en el pin 8 del Arduino Mega 2560, otra página para dos leds conectados en los pines 5 y 8 del Arduino Mega 2560 y otra página para adquirir señales analógicas de potenciómetros conectados a los pines de entradas analógicas de la placa Arduino Mega 2560 A0, A1 y A2. Las páginas web fueron creadas desde el software Arduino mediante código HTML básico para colocar títulos en la pestaña del navegador, títulos en la página, creación de botones, entre otras. Mediante el software Arduino se especificaron la dirección MAC y se asignaron la dirección IP de cada shield, además el puerto mediante el cual se realizará la comunicación Ethernet que es el 80. Con la ayuda de hyperterminal de Windows se realizó la comunicación Telnet pero TCP/IP, donde es necesario especificar la dirección IP del dispositivo con el que se va a comunicar y el puerto que para Telnet es el 23.

PALABRAS CLAVES

ARDUINO

SHIELD

IP

MAC

TELNET

ABSTRACT

This paper takes examples graduation where they explain how to perform Ethernet communication using the Mega 2560 Arduino and Ethernet shield, the shield is connected to the Arduino Mega 2560 via SPI and with the computer through a direct network cable. Web pages to control on / off LEDs, a page for a LED connected to pin 8 of the Arduino Mega 2560, another page for two LEDs connected to pins 5 and 8 of Arduino Mega 2560 and another page is created to acquire potentiometers analog signals connected to the analog input pin Arduino Mega 2560 A0, A1 and A2. The web pages were created from the Arduino software using basic HTML code to place titles in the browser tab, the page titles, creating buttons, among others. By Arduino software they specify the MAC address and IP address of each assigned shield, plus the port through which the Ethernet communication is 80. With the help of Windows hyperterminal perform the Telnet TCP communication but did / IP, which is necessary to specify the IP address of the device that will communicate and port for Telnet is 23.

KEYWORDS
ARDUINO
SHIELD
IP
MAC
TELNET

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA.

IMPLEMENTACIÓN DE 4 MÓDULOS PARA PRÁCTICAS DE COMUNICACIÓN ETHERNET EMPLEANDO ARDUINO MEGA PARA EL LABORATORIO DE INSTRUMENTACIÓN VIRTUAL DE LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS

1.1 ANTECEDENTES.

El laboratorio de Instrumentación Virtual localizado en las instalaciones de la Unidad de Tecnologías de la Universidad de las Fuerzas Armadas-ESPE es un centro académico de formación tecnológica superior situado en la provincia de Cotopaxi, cantón Latacunga; en la calle Xavier Espinoza y Av. Amazonas.

Se requiere incorporar dispositivos Arduino que es una plataforma de desarrollo de computación física, de código abierto, basada en una placa con un sencillo microcontrolador y un entorno de desarrollo para crear software. Arduino es hardware libre, debido a que no existen para el desarrollo de prácticas en la materia de microcontroladores II.

Arduino nació en el Instituto Italiano de Diseño Interactivo Ivrea, una escuela donde los estudiantes centraban sus experimentos en la interacción con dispositivos, muchos de ellos basados en microcontroladores.

Arduino surgió de una necesidad, de contar con un dispositivo de bajo coste, para utilizar en clase con los alumnos, y que funcionase bajo cualquier sistema operativo.

El lenguaje de programación de Arduino es una implementación de Wiring que a su vez se basa en Processing, Actualmente ya se han creado oficialmente más de 120 mil placas Arduino. En cuanto a ventas, Nathan Seidle, CEO de sparkfun.com dice que han vendido 40.000 unidades de la versión Arduino USB.

1.2 PLANTEAMIENTO DEL PROBLEMA.

El laboratorio de Instrumentación Virtual de la Unidad de Gestión de Tecnologías de la Universidad de Fuerzas Armadas – ESPE, no cuenta con

placas Arduino Mega y Shield de Comunicación Ethernet que permita a los alumnos de sexto Nivel de la carrera de Electrónica desarrollar proyectos relacionados con estos dos dispositivos electrónicos.

Al no contar con los expuestos se está limitando a los señores estudiantes a que desarrollen proyectos que de manera eficiente se pueden realizar utilizando las placas Arduino en aplicaciones como por ejemplo la domótica.

1.3 JUSTIFICACIÓN

En la actualidad las tarjetas Arduino son muy empleadas a Nivel Mundial para el desarrollo de un sinnúmero de proyectos electrónicos como por ejemplo domóticas, mandos a distancia, etc.

La implementación de dichos módulos permitirá mejorar la enseñanza aprendizaje de los estudiantes de sexto nivel de la carrera de Electrónica de la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas ESPE. A partir de los módulos implementados los estudiantes podrán desarrollar prácticas que permitan conectar Arduino a una Red LAN utilizando la librería Ethernet y de esta manera poder controlar y monitorear dispositivos electrónicos desde la red LAN e inclusive desde Internet.

1.4 OBJETIVOS.

1.4.1 Objetivo general.

Implementar 4 módulos para prácticas de comunicación Ethernet empleando Arduino Mega para el laboratorio de Instrumentación Virtual de la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas-ESPE**.**

1.4.2 Objetivos específicos.

- Investigar acerca de los Shield Ethernet para Arduino y el protocolo TCP/IP.
- Realizar pruebas de comunicación entre el Arduino Mega y el Shield Ethernet.
- Realizar el control señales digitales y el monitoreo de señales analógicas mediante la comunicación Ethernet

1.5 Alcance.

Este proyecto está dirigido a la carrera de Electrónica Mención Instrumentación y Aviónica de la Universidad de Fuerzas Armadas-ESPE, para la asignatura de Microcontroladores II, brindando a los estudiantes un módulo donde puedan desarrollar prácticas y aplicar los conocimientos adquiridos en el aula, lo que permitirá al estudiante obtener mayor experiencia en el campo práctico para posteriormente desempeñarse de mejor manera en el ámbito laboral, logrando contar con profesionales altamente capacitados y competitivos, capaces de contribuir con el desarrollo de nuestro país.

CAPÍTULO II
MARCO TEÓRICO.

2.1 Ethernet

Ethernet es una popular tecnología LAN (Red de Área Local) que utiliza el Acceso múltiple con portadora y detección de colisiones (Carrier Sense Múltiple Access with Collision Detection, CSMA/CD) entre estaciones con diversos tipos de cables. (Arauz, 2006)

2.1.1 Orígenes de Ethernet

Ethernet fue creado por Robert Metcalfe y otros en Xerox Parc, centro de investigación de Xerox para interconectar computadoras Alto. El diseño original funcionaba a 1 Mbps sobre cable coaxial grueso con conexiones vampiro (que "muerden" el cable) en 10Base5. Para la norma de 10 Mbps se añadieron las conexiones en coaxial fino categoría 5 (10Base2, también de 50 ohmios, pero más flexible), con tramos conectados entre sí mediante conectores BNC; par trenzado categoría 3 (10BaseT) con conectores RJ45, mediante el empleo de hubs y con una configuración física en estrella; e incluso una conexión de fibra óptica (10BaseF).

2.1.2 Características de Ethernet

- Es PASIVO, es decir, no requiere una fuente de alimentación propia, y por tanto, no falla a menos que el cable se corte físicamente o su terminación sea incorrecta.
- Se conecta utilizando una topología de bus en la que el cable está terminado en ambos extremos.
- Utiliza múltiples protocolos de comunicación y puede conectar entornos informáticos heterogéneos, incluyendo Netware, UNIX, Windows y Macintosh.

2.1.3 Método de Acceso

El método de acceso que usa Ethernet es el Acceso múltiple con portadora y detección de colisiones (Carrier Sense Múltiple Access with Collision Detection, CSMA/CD).

CSMA/CD es un conjunto de reglas que determina el modo de respuesta de los dispositivos de red cuando dos de ellos intentan enviar datos en la red simultáneamente. La transmisión de datos por múltiples equipos simultáneamente a través de la red produce una colisión.

Cada equipo de la red, incluyendo clientes y servidores, rastrea el cable en busca de tráfico de red. Únicamente cuando un equipo detecta que el cable está libre y que no hay tráfico envía los datos. Después de que el equipo haya transmitido los datos en el cable, ningún otro equipo puede transmitir datos hasta que los datos originales hayan llegado a su destino y el cable vuelva a estar libre. Tras detectar una colisión, un dispositivo espera un tiempo aleatorio y a continuación intenta retransmitir el mensaje.

Si el dispositivo detecta de nuevo una colisión, espera el doble antes de intentar retransmitir el mensaje.

2.1.4 Velocidad de Transferencia

Ethernet estándar, denominada 10BaseT, soporta velocidades de transferencia de datos de 10 Mbps sobre una amplia variedad de cableado. También están disponibles versiones de Ethernet de alta velocidad. Fast Ethernet (100BaseT) soporta velocidades de transferencia de datos de 100 Mbps y Gigabit Ethernet soporta velocidades de 1 Gbps (gigabyte por segundo) o 1,000 Mbps.

2.1.5 Importancia de Ethernet

Ethernet es popular porque permite un buen equilibrio entre velocidad, costo y facilidad de instalación. Estos puntos fuertes, combinados con la amplia aceptación en el mercado y la habilidad de soportar virtualmente todos los protocolos de red populares, hacen a Ethernet la tecnología ideal para la red de la mayoría de usuarios de la informática actual.

2.1.6 Hardware utilizados por Ethernet

NIC, o adaptador de red Ethernet, permite el acceso de una computadora a una red. Cada adaptador posee una dirección MAC que la identifica en la red y es única. Una computadora conectada a una red se denomina nodo.

Repetidor o repeater, aumenta el alcance de una conexión física, disminuyendo la degradación de la señal eléctrica en el medio físico.

Concentrador o hub, funciona como un repetidor, pero permite la interconexión de múltiples nodos, además cada mensaje que es enviado por un nodo, es repetido en cada boca el hub.

Puente o bridge, interconectan segmentos de red, haciendo el cambio de frames (tramas) entre las redes de acuerdo con una tabla de direcciones que dice en que segmento está ubicada una dirección MAC.

Enrutador o router, funciona en una capa de red más alta que los anteriores el nivel de red, como en el protocolo IP, por ejemplo- haciendo el enrutamiento de paquetes entre las redes interconectadas. A través de tablas y algoritmos de enrutamiento, un enrutador decide el mejor camino que debe tomar un paquete para llegar a una determinada dirección de destino.

Conmutador o Switch, funciona como el bridge, pero permite la interconexión de múltiples segmentos de red, funciona en velocidades más rápidas y es más sofisticado. Los switches pueden tener otras funcionalidades, como redes virtuales y permiten su configuración a través de la propia red.

2.2 Arduino

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador.

Las placas pueden ser hechas a mano o compradas montadas de fábrica; el software puede ser descargado de forma gratuita. Los ficheros de diseño de referencia (CAD) están disponibles bajo una licencia abierta. (Arduino, Arduino Genuino, s.f.)

2.3 Processing

Processing es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital. Fue iniciado por Ben Fry y Casey Reas a partir de reflexiones en el Aesthetics and Computation Group del MIT Media Lab dirigido por John Maeda.

2.4 Arduino Mega 2560

El Arduino Mega 2560 es una placa electrónica basada en el Atmega2560. Cuenta con 54 pines digitales de entrada / salida (de los cuales 15 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UARTs (hardware puertos serie), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un terminal ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar al microcontrolador; simplemente conectarlo a un ordenador con un cable USB o con un adaptador de CA o la batería a CC para empezar. (Arduino, Arduino Mega 2560, 2015)

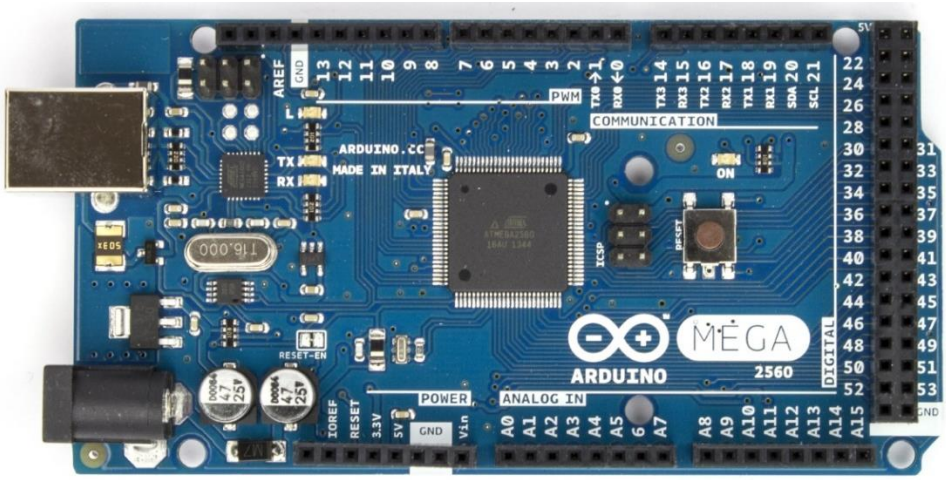


Figura 1. Arduino Mega 2560 vista frontal

Fuente: (Arduino, Arduino Mega 2560, 2015)

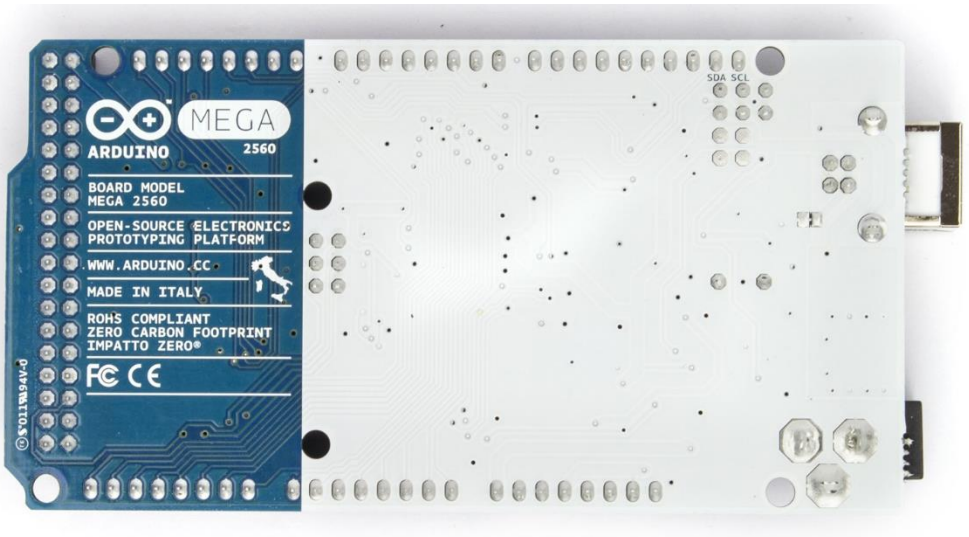


Figura 2. Arduino Mega 2560 vista posterior

Fuente: (Arduino, Arduino Mega 2560, 2015)

El Mega 2560 difiere de todas las placas anteriores en que no utiliza el chip controlador de USB a serial FTDI. En lugar de ello, se cuenta con el ATmega16U2 programado como un convertidor de USB a serie.

2.4.1 Resumen Arduino Mega 2560

Microntrolador	Atmega 2560
Voltaje de operación	5V
Voltaje de entrada (recomendado)	7 a 12 V
Voltaje de entrada (limites)	6 a 20 V
Pines de E/S digital	54 (de los cuales 15 para PWM)
Pines de entrada analógica	16
Corriente DC por pin de E/S	40 mA
Corriente Dc para pin 3.3 V	50 mA
Memoria Flash	256 KB de los cuales 8KB es usado por el bootloader
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz

2.4.2 Memoria Flash

La memoria flash derivada de la memoria EEPROM permite la lectura y escritura de múltiples posiciones de memoria en la misma operación. Gracias a ello, la tecnología flash, siempre mediante impulsos eléctricos, permite velocidades de funcionamiento muy superiores frente a la tecnología EEPROM primigenia, que sólo permitía actuar sobre una única celda de memoria en cada operación de programación. Se trata de la tecnología empleada en los dispositivos denominados memoria USB.

2.4.3 Gestor de arranque

Un gestor de arranque (en inglés «bootloader») es un programa sencillo que no tiene la totalidad de las funcionalidades de un sistema operativo, y que está diseñado exclusivamente para preparar todo lo que necesita el sistema operativo para funcionar. Normalmente se utilizan los cargadores de arranque multietapas, en los que varios programas pequeños se suman los unos a los otros, hasta que el último de ellos carga el sistema operativo.

2.4.4 SRAM

SRAM son las siglas de la voz inglesa Static Random Access Memory, que significa memoria estática de acceso aleatorio (o RAM estática), para denominar a un tipo de tecnología de memoria RAM basada en semiconductores, capaz de mantener los datos, mientras siga alimentada, sin necesidad de circuito de refresco. Este concepto surge en oposición al de memoria DRAM (RAM dinámica), con la que se denomina al tipo de tecnología RAM basada en condensadores, que sí necesita refresco dinámico de sus cargas.

2.4.5 EEPROM

EEPROM o E²PROM son las siglas de Electrically Erasable Programmable Read-Only Memory (ROM programable y borrable eléctricamente). Es un tipo de memoria ROM que puede ser programada, borrada y reprogramada eléctricamente, a diferencia de la EPROM que ha

de borrarse mediante un aparato que emite rayos ultravioleta. Son memorias no volátiles.

Las celdas de memoria de una EEPROM están constituidas por un transistor MOS, que tiene una compuerta flotante (estructura SAMOS), su estado normal está cortado y la salida proporciona un 1 lógico.

Aunque una EEPROM puede ser leída un número ilimitado de veces, sólo puede ser borrada y reprogramada entre 100.000 y un millón de veces.

Estos dispositivos suelen comunicarse mediante protocolos como I²C, SPI y Microwire. En otras ocasiones, se integra dentro de chips como microcontroladores y DSPs para lograr una mayor rapidez.

2.4.6 Alimentación

El Mega Arduino puede ser alimentado a través de la conexión USB o con una fuente de alimentación externa. La fuente de alimentación se selecciona automáticamente.

La fuente externa (no USB) puede ser un adaptador de CA a CC o una batería. El adaptador se puede conectar con un terminal de 2,1 mm de centro-positivo en el conector de alimentación de la placa. Los cables de una batería se pueden insertar en los terminales GND y Vin del conector POWER.

La placa puede funcionar con un suministro externo de 6 a 20 voltios. Si se suministra con menos de 7V, sin embargo, el pin de 5V puede suministrar menos de cinco voltios y la placa puede ser inestable. Si se utiliza más de 12V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son los siguientes

VIN. El voltaje de entrada a la placa Arduino cuando se utiliza una fuente de alimentación externa. Usted puede suministrar tensión a través de este pin, o, si el suministro de tensión a través de la toma de poder, acceder a él a través de este pin.

5V. Este pin es salida de 5V del regulador de la placa.

3V3. Entrega 3,3 V generados por el regulador de la placa. Entrega Corriente máxima es de 50 mA.

GND. Terminal de tierra.

Instrucción IOREF. Este pin de la placa Arduino proporciona la referencia de tensión con la que opera el microcontrolador.

2.4.7 Memoria

El Atmega 2560 tiene 256 KB de memoria flash para almacenar código (de los cuales 8 KB se utiliza para el gestor de arranque), 8 KB de SRAM y 4 KB de EEPROM (que puede ser leído y escrito con la libreria EEPROM).

2.4.8 Diagramas del pines del Arduino Atmega 2560

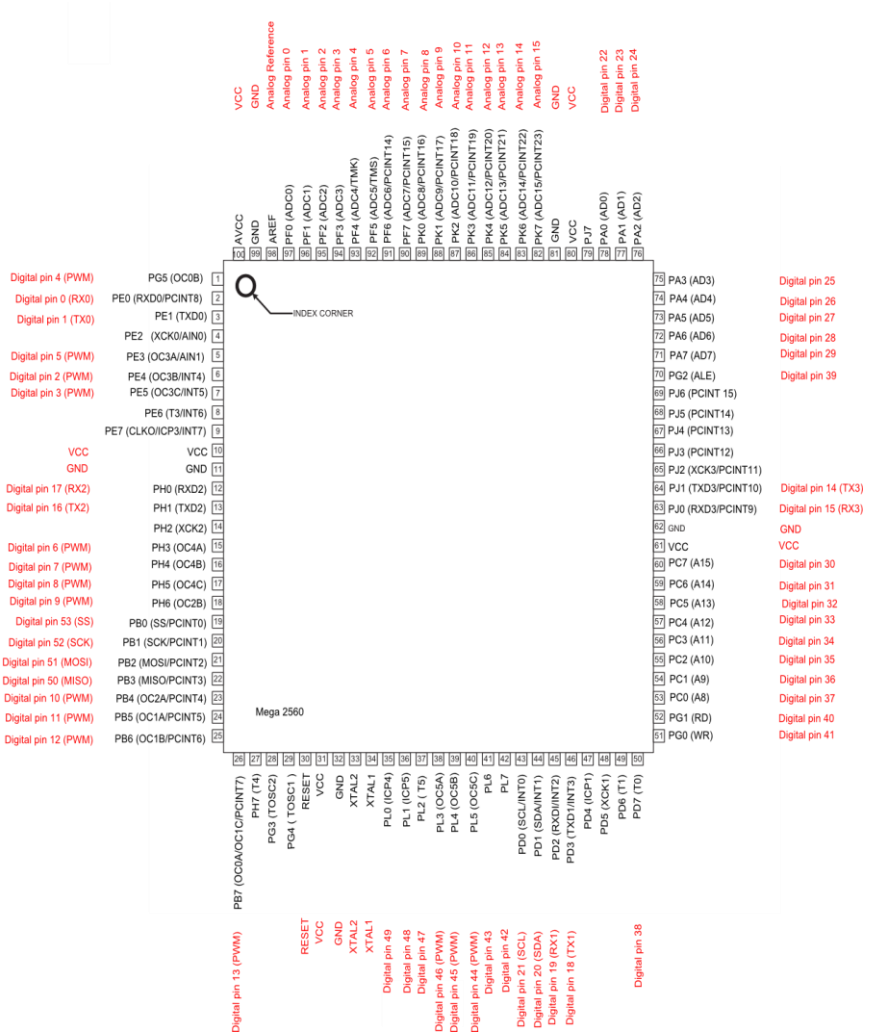


Figura 3. Pines del Arduino Mega 2560

Fuente: (Arduino, Arduino Mega 2560, 2015)

Número de Pin	Nombre del Pin	Descripción
1	PG5 (OC0B)	Digital pin 4 (PWM)
2	PE0 (RXD0/PCINT8)	Digital pin 0 (RX0)
3	PE1 (TXD0)	Digital pin 1 (TX0)
4	PE2 (XCK0/AIN0)	
5	PE3 (OC3A/AIN1)	D
6	PE4 (OC3B/INT4)	Digital pin 2 (PWM)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)
8	PE6 (T3/INT6)	
9	PE7 (CLKO/ICP3/INT7)	
10	VCC	VCC
11	GND	GND
12	PH0 (RXD2)	Digital pin 17 (RX2)
13	PH1 (TXD2)	Digital pin 16 (TX2)
14	PH2 (XCK2)	
15	PH3 (OC4A)	Digital pin 6 (PWM)
16	PH4 (OC4B)	Digital pin 7 (PWM)
17	PH5 (OC4C)	Digital pin 8 (PWM)
18	PH6 (OC2B)	Digital pin 9 (PWM)
19	PB0 (SS/PCINT0)	Digital pin 53 (SS)
20	PB1 (SCK/PCINT1)	Digital pin 52 (SCK)
21	PB2 (MOSI/PCINT2)	Digital pin 51 (MOSI)
22	PB3 (MISO/PCINT3)	Digital pin 50 (MISO)
23	PB4 (OC2A/PCINT4)	Digital pin 10 (PWM)
24	PB5 (OC1A/PCINT5)	Digital pin 11 (PWM)
25	PB6 (OC1B/PCINT6)	Digital pin 12 (PWM)
26	PB7 (OC0A/OC1C/PCINT7)	Digital pin 13 (PWM)
27	PH7 (T4)	
28	PG3 (TOSC2)	
29	PG4 (TOSC1)	

Continua

30	RESET	RESET
31	VCC	VCC
32	GND	GND
33	XTAL2	XTAL2
34	XTAL1	XTAL1
35	PL0 (ICP4)	
36	PL1 (ICP5)	Digital pin 48
37	PL2 (T5)	Digital pin 47
38	PL3 (OC5A)	Digital pin 46 (PWM)
39	PL4 (OC5B)	Digital pin 45 (PWM)
40	PL5 (OC5C)	Digital pin 44 (PWM)
41	PL6	Digital pin 43
42	PL7	Digital pin 42
43	PD0 (SCL/INT0)	Digital pin 21 (SCL)
44	PD1 (SDA/INT1)	Digital pin 20 (SDA)
45	PD2 (RXDI/INT2)	Digital pin 19 (RX1)
46	PD3 (TXD1/INT3)	Digital pin 18 (TX1)
47	PD4 (ICP1)	
48	PD5 (XCK1)	
49	PD6 (T1)	
50	PD7 (T0)	Digital pin 38
51	PG0 (WR)	Digital pin 41
52	PG1 (RD)	Digital pin 40
53	PC0 (A8)	Digital pin 37
54	PC1 (A9)	Digital pin 36
55	PC2 (A10)	Digital pin 35
56	PC3 (A11)	Digital pin 34
57	PC4 (A12)	Digital pin 33
58	PC5 (A13)	Digital pin 32
59	PC6 (A14)	Digital pin 31
60	PC7 (A15)	Digital pin 30

Continua

61	VCC	VCC
62	GND	GND
63	PJ0 (RXD3/PCINT9)	Digital pin 15 (RX3)
64	PJ1 (TXD3/PCINT10)	Digital pin 14 (TX3)
65	PJ2 (XCK3/PCINT11)	
66	PJ3 (PCINT12)	
67	PJ4 (PCINT13)	
68	PJ5 (PCINT14)	
69	PJ6 (PCINT 15)	
70	PG2 (ALE)	Digital pin 39
71	PA7 (AD7)	Digital pin 29
72	PA6 (AD6)	Digital pin 28
73	PA5 (AD5)	Digital pin 27
74	PA4 (AD4)	Digital pin 26
75	PA3 (AD3)	Digital pin 25
76	PA2 (AD2)	Digital pin 24
77	PA1 (AD1)	Digital pin 23
78	PA0 (AD0)	Digital pin 22
79	PJ7	
80	VCC	VCC
81	GND	GND
82	PK7 (ADC15/PCINT23)	Analog pin 15
83	PK6 (ADC14/PCINT22)	Analog pin 14
84	PK5 (ADC13/PCINT21)	Analog pin 13
85	PK4 (ADC12/PCINT20)	Analog pin 12
86	PK3 (ADC11/PCINT19)	Analog pin 11
87	PK2 (ADC10/PCINT18)	Analog pin 10
88	PK1 (ADC9/PCINT17)	Analog pin 9
89	PK0 (ADC8/PCINT16)	Analog pin 8
90	PF7 (ADC7)	Analog pin 7
91	PF6 (ADC6)	Analog pin 6

Continua

92	PF5 (ADC5/TMS)	Analog pin 5
93	PF4 (ADC4/TMK)	Analog pin 4
94	PF3 (ADC3)	Analog pin 3
95	PF2 (ADC2)	
96	PF1 (ADC1)	Analog pin 1
97	PF0 (ADC0)	Analog pin 0
98	AREF	Analog Reference
99	GND	GND
100	AVCC	VCC

Continua

Tabla 1. Descripción de pines del Arduino Mega 2560

Fuente: (Arduino, Arduino Mega 2560, 2015)

2.5 Shield Ethernet Arduino

2.5.1 Introducción

El Shield Ethernet Arduino permite conectar a Arduino a Internet o a una red LAN en cuestión de minutos. Sólo tiene que conectar este módulo en su placa Arduino, conectarlo a su red con un cable RJ45 y seguir algunas instrucciones sencillas para empezar a controlar su mundo a través de internet o aplicaciones LAN. Como siempre con Arduino, todos los elementos de la plataforma hardware, software y documentación son de libre acceso y de fuente abierta. Esto significa que usted puede aprender exactamente cómo se hace y utilice su diseño como punto de partida para sus propios circuitos. (Arduino Ethernet, 2015)

- Requiere unaTarjeta Arduino
- Voltaje de operación 5V (suministrado por la tarjeta Arduino)
- Controlador Ethernet W5100 con buffer interno de 16 K
- Velocidad de conexión 10/100 Mb
- Conexión con Arduino con el puerto SPI

2.5.2 Descripción

El Shield Ethernet Arduino permite que una placa Arduino se conecte a internet. Se basa en el chip ethernet Wiznet W5100. El Wiznet W5100 ofrece una red (IP) TCP y UDP. Soporta hasta cuatro conexiones de socket simultáneas. Utilice la librería de Ethernet para escribir programas que se conectan a Internet a través del shield. El shield de Ethernet se conecta a una placa Arduino usando terminales largos. Esto mantiene la disposición de las pines intactos y permite que otro shield se conecte en la parte superior.

Hay una ranura para tarjetas micro-SD en la tarjeta, que se puede utilizar para almacenar archivos para enviar a través de la red. Es compatible con el Arduino Uno y Mega (utilizando la librería Ethernet). El lector de tarjetas microSD a bordo es accesible a través de la librería SD. Cuando se trabaja con esta librería, SS es el Pin 4.

El shield también incluye un controlador de reajuste, para asegurar que el módulo Ethernet W5100 se restablece correctamente en el encendido.

El shield tiene un módulo de alimentación a través de Ethernet (Power over Ethernet PoE), diseñado para extraer energía de un cable Ethernet de par trenzado de categoría 5 convencional

- Compatible con IEEE802.3af
- Bajo rizo y ruido de salida (100 mVpp)
- Rango de voltaje de entrada de 36 a 57 V
- Protección contra corto circuito y sobrecarga
- Convertidor DC/DC de alta eficiencia
- Salida de 9V
- Aislamiento de 1500 V (entra a la salida)

Arduino se comunica tanto con el W5100 y la tarjeta SD usando el bus SPI (a través de la cabecera ICSP). Esto es en los pines digitales 10, 11, 12, y 13 en el Uno y los pines 50, 51, y 52 en los Mega. En ambas tarjetas, el pin 10 se utiliza para seleccionar el W5100 y el pin 4 de la tarjeta SD. Estos pines no se pueden utilizar como entradas o salidas.

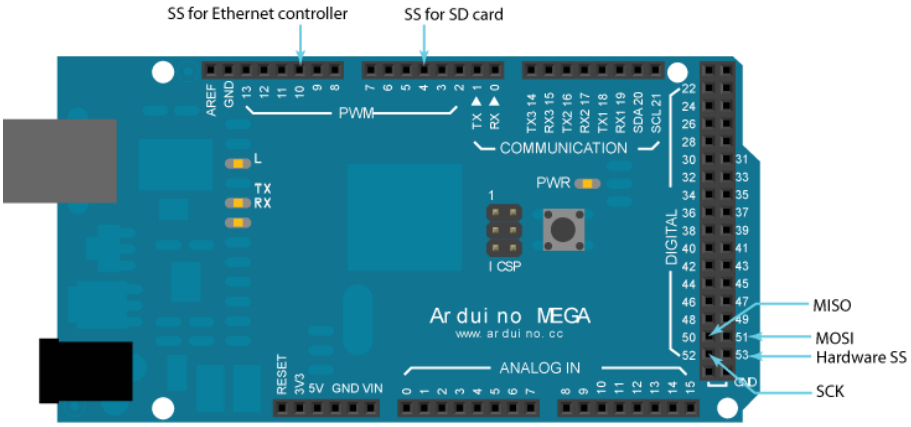


Figura 4. Terminales ocupados por el shield Ethernet

Fuente: (Arduino Ethernet, 2015)

El shield provee un conector ethernet RJ45 estándar.

El botón de reinicio en el shield restablece tanto el W5100 y la placa Arduino.

El shield contiene un número de LEDs informativos:

ON: indica que la tarjeta arduino y shield están encendidos

LINK: indica la presencia de un enlace de red y parpadea cuando el shield transmite o recibe datos

100M: Indica la presencia de una conexión de red del 100 Mb/s

RX: parpadea cuando el shield recibe datos

TX: parpadea cuando el shield envía datos

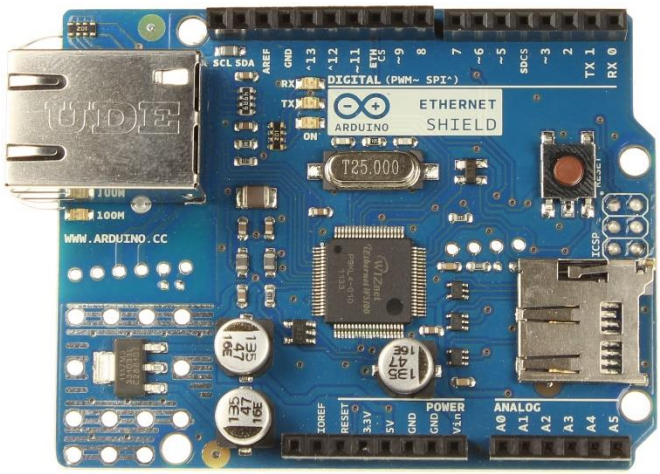


Figura 5. Arduino Ethernet Shield vista frontal

Fuente: (Arduino Ethernet, 2015)

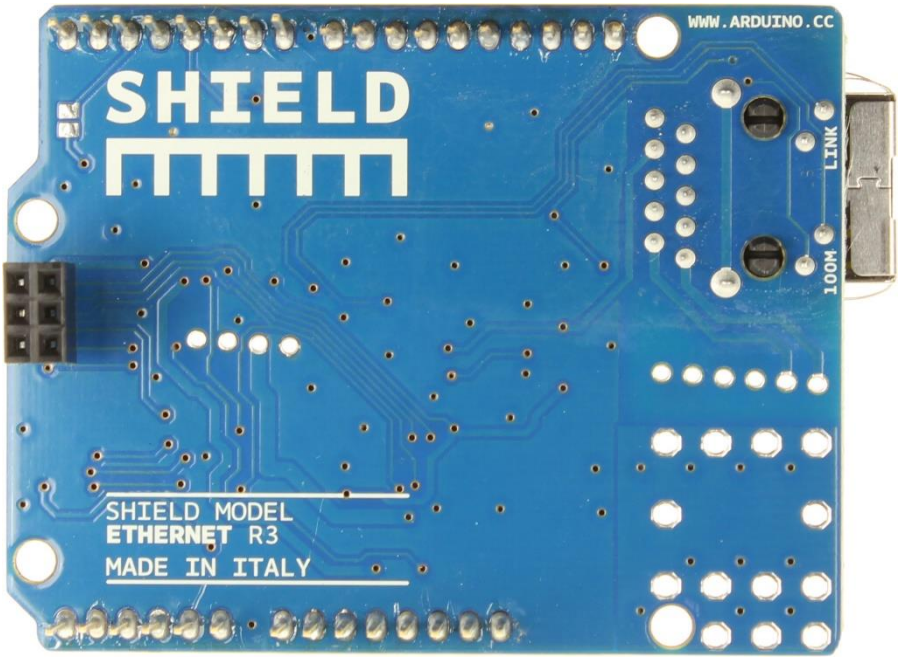


Figura 6. Arduino Ethernet Shield vista posterior

Fuente: (Arduino Ethernet, 2015)

2.6 Entorno de desarrollo Arduino

El entorno de desarrollo Arduino contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comunes, y una serie de menús. Se conecta al hardware Arduino para cargar programas y comunicarse con ellos.

2.6.1 Creación de programas (sketches)

El software escrito utilizando Arduino se llama sketch. Estos sketches se escriben en el editor de texto. Los sketches se guardan con la extensión de archivo .ino. Tiene características para cortar/pegar y para buscar / reemplazar texto. El área de mensajes proporciona información mientras se guarda o se compila programas y también muestra los errores. La consola muestra la salida de texto por el entorno Arduino incluyendo mensajes de error completas y otra información. La esquina derecha inferior de la ventana muestra la tarjeta empleada y el puerto serie (COM). Los botones de la barra de herramientas le permiten verificar y cargar programas, crear, abrir y guardar dibujos, y abrir el monitor de serie.

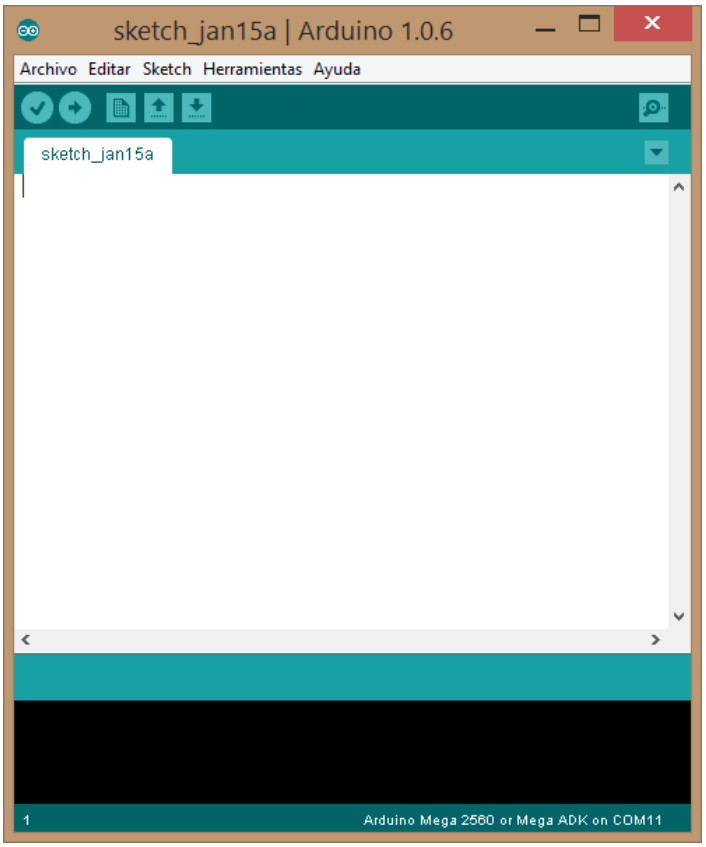


Figura 7. IDE de Arduino

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

2.6.2 Estructura de programación

La estructura de programación en Arduino está formada por la función setup() y la función loop().

Setup(): La función setup () se llama cuando se inicia un sketh. Se una para inicializar variables, asignación de dirección a los pines de E/S digitales, inicio de librerias, etc. La función de configuración sólo se ejecutará una vez, después de cada arranque o reinicio de la placa Arduino.

Loop(): Después de crear una función setup(), que inicializa y establece los valores iniciales, la función loop() hace exactamente lo que su nombre indica un bucle de forma consecutiva, lo que permite el programa para cambiar y responder. Lo utilizan para controlar activamente la placa Arduino.

2.6.3 Funciones de E/S digitales

2.6.3.1 pinMode()

Configura el pin especificado a comportarse ya sea como una entrada o una salida.

Sintaxi

pinMode(pin, mode);

Parámetros

pin: número de pin cuyo modo se desea fijar

mode: INPUT, OUTPUT, or INPUT_PULLUP

2.6.3.2 digitalWrite()

Escribe un valor de alto o bajo en el pin digital

Si el pin se ha configurado como una salida con pinMode (), su voltaje se establece en el valor correspondiente: 5V para alto, 0V (tierra) para bajo

El pin 13 tiene soldado un LEd y una resistencia en la placa Arduino y nunca podrá ser usado como entrada digital.

Sintaxi

digitalWrite(pin,valor);

Parámetros

pin: número de pin

valor: HIGH o LOW

2.6.3.3 digitalRead()

Lee el valor de un pin digital especificado, ya sea alto o bajo.

Sintaxi

digitalRead(pin);

Parámetro

pin: el número del pin digital que desea leer

2.6.4 Funciones de E/S analógicas

2.6.4.1 analogRead()

Lee el valor del pin analógico especificado. La placa Arduino Mega contiene 16 canales analógicos de 10 bits. Esto significa que va a asignar tensiones de entrada entre 0 y 5 voltios en valores enteros entre 0 y 1023.

Esto produce una resolución entre las lecturas de: 5 voltios / 1024 unidades o, 0,0049 voltios (4,9 mV) por unidad.

Sintaxi

analogRead(pin);

Parametro

pin: número del pin analógico de entrada que se desea leer

2.6.4.2 analogWrite()

Escribe un valor analógico (onda PWM) en un pin. Se puede utilizar para encender un LED en diferentes brillos o controlar un motor a distintas velocidades. Después de llamar a analogWrite(), el pin generará una onda cuadrada estable del ciclo de trabajo especificado hasta la siguiente llamada a analogWrite(). La frecuencia de la señal PWM es de aproximadamente 490 Hz.

Sintaxi

analogWrite(pin,valor);

Parámetros

pin: pin donde se escribe el avlor

valor: ciclo de trabajo entre 0 y 255

2.7 Telnet

Telnet (Teletype Network) es el nombre de un protocolo de red que nos permite viajar a otra máquina para manejarla remotamente como si estuviéramos sentados delante de ella. También es el nombre del programa informático que implementa el cliente. Para que la conexión funcione, como en todos los servicios de Internet, la máquina a la que se acceda debe tener un programa especial que reciba y gestione las conexiones. El puerto que se utiliza generalmente es el 23.

2.8 HTML creación de páginas web

2.8.1 Introducción

HTML es un lenguaje universal, que funciona en cualquier plataforma (Windows, Macintosh, Unix, OS/2, etc.) y con cualquier navegador o browser (Netscape, Internet Explorer, Mozilla Firefox, etc.). El precio que paga por su universalidad es su poca sofisticación, puesto que no es otra cosa que el viejo formato ASCII2 (y con 7 bits en lugar de ocho, por lo que ni tan siquiera tiene acentos ni otros caracteres especiales).

Por tanto, para que un documento HTML sea algo más que simples caracteres básicos, debe contener, además de dicho texto, una serie de instrucciones para el browser que lo va a reproducir: estas instrucciones se denominan etiquetas o tags y se distinguen del texto porque van entre guiones (< >)3. Estas etiquetas contienen todo el resto de la información de la página web.

Por tanto, si un documento HTML no es más que texto ASCII (parte sin guiones y parte entre guiones: las etiquetas), cualquier documento web, hasta el más sofisticado, puede escribirse directamente desde un sencillo programa básico de texto, como el Cuaderno de Notas (Notepad) de Windows, por ejemplo.

Sin embargo, escribir un documento complejo de esta manera exige un conocimiento exhaustivo de las numerosísimas etiquetas existentes y sus normas de utilización; para evitar esta dificultad, hay una serie de programas denominados comúnmente editores de HTML (desde el Composer al Dreamweaver, pasando por tantos otros), que permiten al usuario componer una página (es decir, generar etiquetas HTML) desde un interfaz más o menos parecido al de un procesador de textos.

A la hora de crear una página web, y aunque utilicemos un editor de HTML, es casi imprescindible entender la disposición del etiquetado de un documento y manejar siquiera las etiquetas más elementales. Son varias las razones que pueden aducirse para ello:

Los editores de HTML no son siempre herramientas perfectas, sobre todo cuando ha habido muchas modificaciones durante el proceso de creación. Con bastante frecuencia se hará necesario consultar el código completo, con las etiquetas HTML (lo que se suele denominar código fuente o código madre) para corregir las disfunciones.

Tal vez la mejor manera de aprender a diseñar páginas web es observar las páginas de los demás, cuyos códigos fuente están siempre disponibles para ser imitados (o incluso copiados, técnicamente, al menos). Pero, para ello, es necesario entenderlos.

Los mismos editores HTML manejan términos propios del lenguaje HTML (heading, cellspacing, etc.), que es preciso comprender.

Es necesario un buen conocimiento de HTML para insertar otros códigos más potentes, como los de Javascript, por ejemplo.

El objetivo de este documento es aprender a componer una sencilla página web desde el más básico de los programas de texto disponibles en Windows: el Cuaderno de Notas.4 De esta manera, seremos capaces de entender y manejar los códigos fuente de las páginas más complejas. (Iribar, s.f.)

2.8.2 Esquema mínimo de un documento HTML

Un documento HTML comienza siempre con la etiqueta <HTML>, que indica que el documento en cuestión está construido con dicho lenguaje.
La mayoría de las etiquetas son pareadas, es decir, <...> corresponde al principio de la acción y </...> indica el fin de dicha acción.
Por tanto, una página web estará siempre contenida entre las etiquetas <HTML> y </HTML>.
Por otra parte, todo documento HTML consta de dos partes: la cabecera (head) y el cuerpo del documento (body).

- La cabecera contiene básicamente información destinada al browser (o navegador), que queda oculta al usuario. Su etiqueta (pareada) es <HEAD>.
- El cuerpo es el documento que ve el usuario. Su etiqueta (pareada) es <BODY>.

Las siguientes líneas escriben Ejemplo en la barra de título del explorador

```
<HTML>
<HEAD>
<TITLE>Ejemplo 2</TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

2.8.3 Texto básico de un documento HTML

El texto básico de un documento HTML puede escribirse sin etiquetas. Cada navegador lo visualiza entonces con el tipo y tamaño de caracteres escogidos en su configuración por defecto, e introduce un salto de línea cuando los caracteres alcanzan el borde de la ventana.

2.8.3.1 Etiqueta básica

La etiqueta básica, que controla el tipo de fuente utilizado, es <FONT>.

<FONT> es una etiqueta pareada. Por tanto, afecta a los caracteres introducidos entre <FONT> y </FONT>. Pero sin más especificaciones, la etiqueta aún no sirve para nada.

Se puede especificar el tipo de letra (es decir, la fuente de caracteres) añadiéndolo a la etiqueta, de la siguiente manera (en este caso, la fuente escogida es Arial): <FONT FACE="Arial">...</FONT>.

Se puede especificar el color de la fuente con la etiqueta <FONT COLOR="?">...</FONT>. Para especificar los colores, conviene saber al menos lo siguiente:

Cada color posee su correspondiente código.

Hay dos códigos: hexadecimal –lo más habitual– o RGB. En ambos casos se trata de expresar las proporciones de tres colores básicos: rojo, verde y azul.

Los valores hexadecimales de los tres colores se expresan de la siguiente manera:

Rojo: color:#ff0000;

Verde: color:#00ff00;

Azul: color:#0000ff;

Los mismos colores, expresados con los valores RGB (red, green, blue), son los siguientes:

Rojo: color:rgb(255,0,0);

Verde: color:rgb(0,255,0);

Azul: color:rgb(0,0,255);

Todos los colores se expresan mediante combinaciones de estos tres colores básicos. Existen muchos programas que proporcionan los valores de todos los colores que el usuario puede componer. También en internet existen muchas páginas con los Códigos de colores.

Los colores más básicos pueden especificarse sin código, simplemente escribiendo la palabra correspondiente (por supuesto en inglés). Por ejemplo: <FONT COLOR="blue">...</FONT>, <FONT COLOR="red">...</FONT>, etc.

Se puede especificar el tamaño de la fuente con la etiqueta <FONT SIZE="?">...</FONT>. El valor del tamaño ("?") deseado puede suministrarse de manera absoluta (un número del 1 al 7) o relativa (+1, +2, -1, -2, etc., teniendo en cuenta que el valor por defecto es 3).

Todas estas etiquetas se pueden combinar, de modo que con una sola se controla, por ejemplo, el tipo, el tamaño y el color de los caracteres: <FONT FACE="Arial" COLOR="#0000FF" SIZE="5">...</FONT>.

2.8.3.2 Etiquetas de forma de caracteres

A partir de este momento, se trata simplemente de añadir más etiquetas a nuestra aún pequeña colección, de modo que vayamos teniendo cada vez más control sobre el formato de los caracteres. La tabla siguiente muestra las más comunes:

Código	Función
<B>...</B>	Negrita
<I>...</I>	Cursiva
<U>...</U>	Subrayado
_{...}	Subíndice
^{...}	Superíndice

Tabla 2. Pines del Arduino Mega 2560

Fuente: (Arduino, Arduino Mega 2560, 2015)

Hay que realizar algunos comentarios sobre las etiquetas precedentes:
Existe otra etiqueta similar a <B>, menos utilizada porque depende de las opciones de configuración del navegador: <STRONG>...</STRONG>.
Existe otra etiqueta similar a <I>, menos utilizada porque depende de las opciones de configuración del navegador: <EM>...</EM>.

Conviene tener cuidado a la hora de manejar el subrayado, puesto que éste se utiliza convencionalmente para indicar los hiperenlaces, y podría por tanto resultar confuso para el usuario.

Es normal utilizar varias etiquetas para un mismo elemento de texto. En ese caso, hay que tener cuidado para encajarlas correctamente. Por ejemplo:

<FONT FACE="Courier"><B><I>...</I></B></FONT> es correcto, pero

<FONT FACE="Courier"><B><I>...</FONT><B></I> es incorrecto.

2.8.4 Párrafos de un documento HTML

2.8.4.1 Párrafos y su alineamiento

La etiqueta
 inserta un salto de línea, pero no un salto de párrafo.7 En cambio, la etiqueta <P>...</P> inserta una salto de línea y una línea en blanco, por lo que, en la práctica, equivale a un nuevo párrafo.

Por supuesto, tanto
 como <P>...</P> pueden repetirse varias veces seguidas para dejar más distancia entre las líneas y los párrafos.

De todas maneras, sólo podemos separar los párrafos con líneas completas (una, dos, etc.), pero no podemos controlar el espacio entre los

párrafos con la precisión que permite, por ejemplo, el programa Word (con las opciones de espaciado anterior y posterior).

Un párrafo puede alinearse introduciendo los siguientes atributos en la etiqueta habitual:

Código	Función
<P align="left">...</P>	Párrafo alineado a la izquierda
<P align="center">...</P>	Párrafo centrado
<P align="right">...</P>	Párrafo alienado a la derecha
<P align="justify">...</P>	Párrafo justificado

Tabla 3. Atributos de párrafo

Fuente: (Arduino, Arduino Mega 2560, 2015)

2.9 Hyperterminal

Hyperterminal es un cliente para hacer conexiones telnet por medio de los puertos serie (por ejemplo COM1) con dispositivos externos. Estos dispositivos pueden variar e incluyen opciones tales como equipos de radio comunicación, los robots, y los instrumentos utilizados para mediciones científicas y equipos de red.

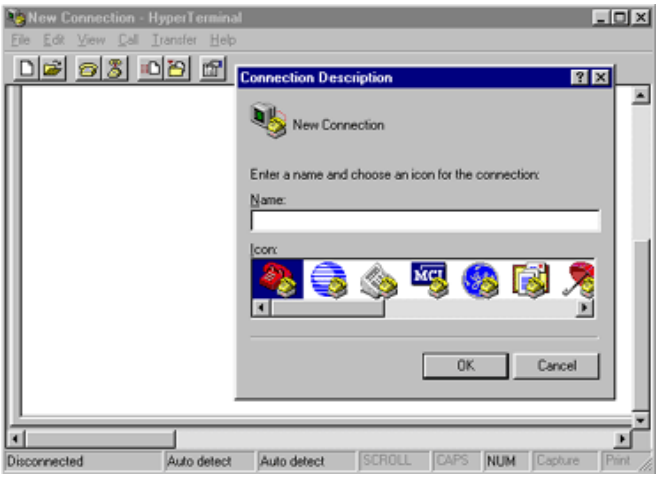


Figura 8. Hyperterminal

Fuente: (Arduino, Arduino Mega 2560, 2015)

CAPÍTULO III
IMPLEMENTACIÓN DE LA COMINICACIÓN ETHERNET

3.1 Preliminares

Para la implementación de la comunicación Ethernet fue necesario adquirir la tarjeta Arduino Mega 2560, el shield Ethernet para Arduino y dispositivos electrónicos varios.

Primero se explicará mediante ejercicios la manipulación del estado de las salidas digitales y analógicas así como el monitoreo de la entradas digitales y analógicas.

3.2 Parpadeo de pin 13 de la placa Arduino Mega 2560

Para el parpadeo del led SMD que esta soldado en la placa Arduino, se escribe en el software Arduino lo siguiente:

Figura 9. Parpadeo de un Led

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

En el parámetro void setup() se configura mediante pinMode(13,OUTPUT) el modo del pin en este caso el número 13 que corresponde al led como salida.

El void loop() que es el lazo indefinido, con digitalWrite(13,HIGH) se coloca en alto el pin13, con delay(400) se realiza un retardo de 400 ms, con digitalWrite(13,LOW), se coloca en bajo el pin 13 y finalmente con delay(400)

se realiza otro retardo de 400 ms. De esta forma el led que corresponde al pin 13 estará parpadeando cada 400 ms.

3.3 Configuración de entrada digital normal

El siguiente ejemplo configura al pin 2 como entrada digital normal; es decir, que se necesita conectar una resistencia del pin 2 a Vcc. Cuando se conecte el pin 2 a GND la salida 13 configurada como salida se encenderá.

Figura 10. Configuración del pin 2 como entrada

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

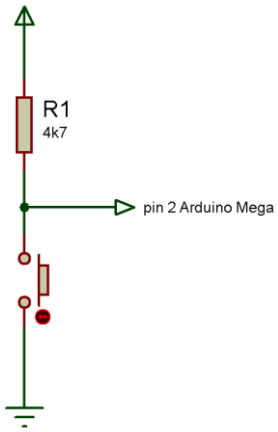


Figura 11. Entrada digital con resistencia pull up externa

3.4 Configuración de entrada digital pull up

El siguiente ejemplo configura al pin 2 como entrada digital pull up; es decir, que no necesita conectar una resistencia del pin 2 a Vcc porque se

utiliza las resistencias internas del microcontrolador. Cuando se conecte el pin 2 a GND la salida 13 configurada como salida se encenderá.

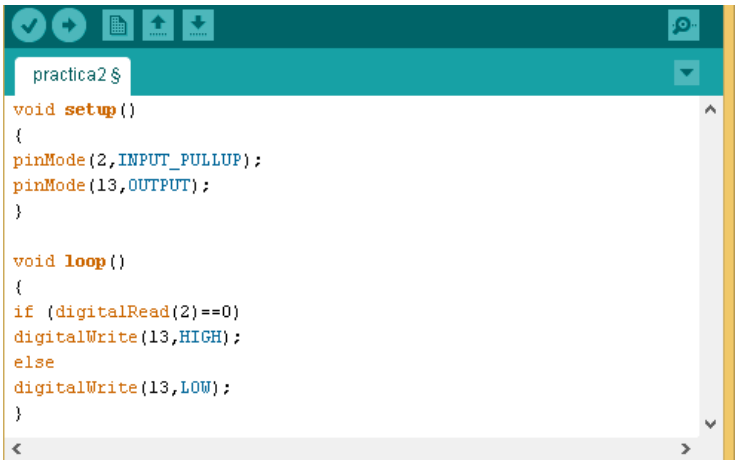


Figura 12. Configuración del pin 2 como entrada pull up

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

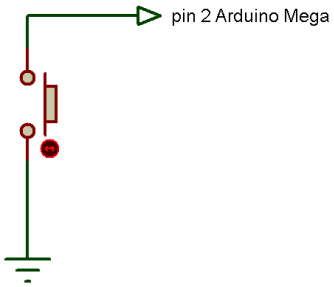


Figura 13. Entrada digital con resistencia pull up interna

3.5 Adquisición de señales analógicas

La tarjeta Arduino Mega 2560 tiene 16 entradas analógicas, etiquetas como A0 hasta la A15, para el ejemplo se utiliza la entrada A0. Cada entrada admite una variación de voltaje entre 0 y 5V, como la resolución del ADC del microcontrolador Atmega 2560 es de 10 bits, se tiene un valor del 1023 que corresponde a los 5V externos en el canal de entrada analógico; es decir, que para convertir ese valor a voltaje es necesario aplicar una regla de tres.

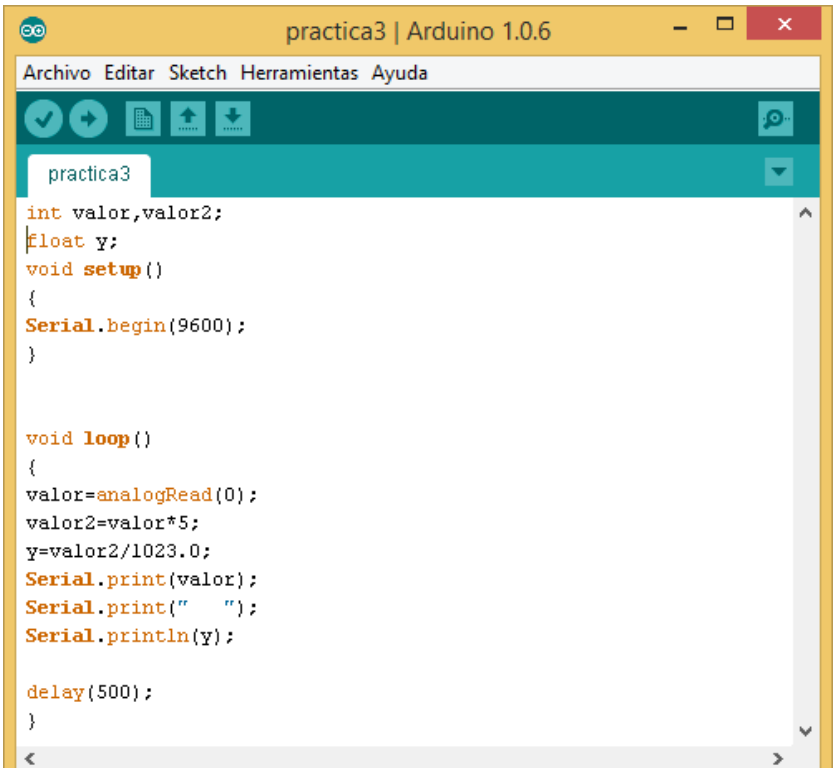


Figura 14. Adquisición de señales analógicas

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

Se declaran dos variables globales de tipo int, llamadas valor y valor2, almacena un valor de 2 bytes, que corresponde a un rango de -32768 a +32767.

La variable y es de tipo flotante, sirve para guardar el valor de la regla de tres simple que es el voltaje.

Para visualizar el valor entero entre 0 a 1023 y su correspondiente voltaje, se emplea el visor serial del software Arduino. Primero se configura la velocidad de transmisión a 9600 con la función Serial.begin(9600). Para mostrar los datos en el visor serial, se emplea la función Serial.Print.

Para comprobar el funcionamiento de la adquisición analógica se emplea un potenciómetro conectado el terminal variable a la entrada analógica A0.

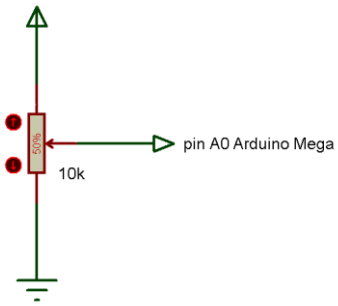


Figura 15. Conexión del potenciómetro para entrada analógica

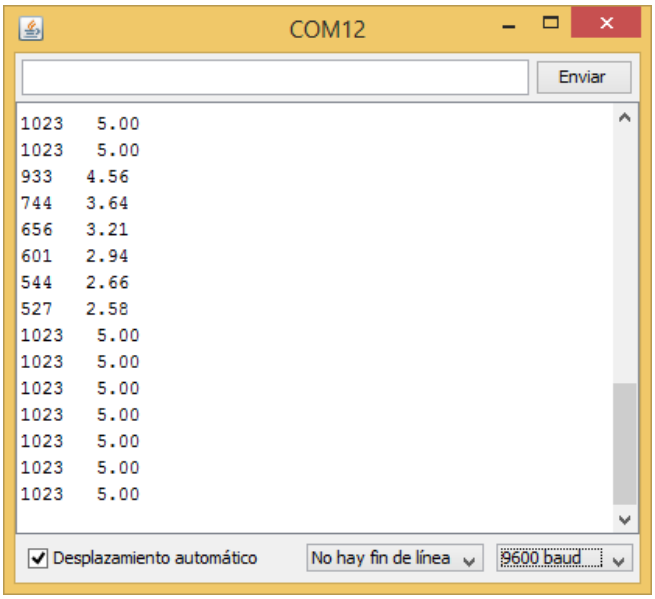


Figura 16. Visualización de la adquisición analógica

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

3.6 Encendido/apagado led desde página HTML

Los pasos para el control de un led conectado al pin 8 del Arduino Mega 2560 desde un navegador web son los siguientes:

1. Copie la dirección MAC del shield Ethernet para este ejemplo es 90-A2-DA-0D-D5-2D que necesario cuando se programa en Arduino.
2. Inserte el shield Ethernet en la tarjeta Arduino Mega 2560
3. Conecte un cable Ethernet desde el shiled hacia el puerto Ethernet del computador.
4. Energice la tarjeta Arduino mediante el puerto USB del computador
5. Asigne una dirección IP estática en el computador por ejemplo 192.168.0.5

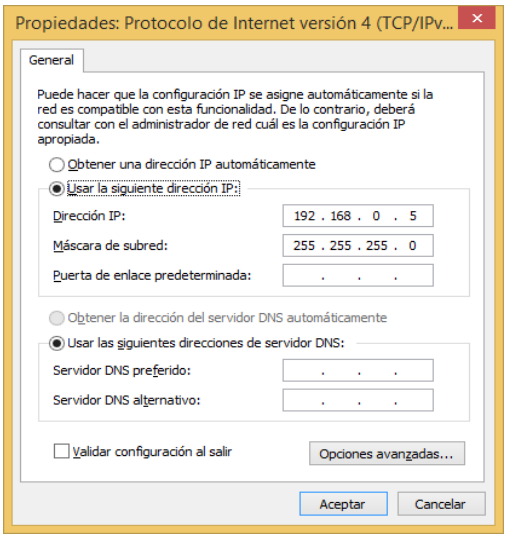


Figura 17. Dirección IP del computador

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

6. Abrir el software Arduino para llamar a las librerías necesarias y escribir la dirección MAC e IP que tendrá el shield. Cargue el programa en la tarjeta Arduino Mega 2560

```
#include <SPI.h>
#include <Ethernet.h>
//Declaración de la direcciones MAC e IP.
byte mac[]={0x90,0xA2,0xDA,0x0D,0xD5,0x2D}; //MAC
IPAddress ip(192,168,0,4); //IP

void setup()
{
 Ethernet.begin(mac, ip); //Inicializamos con las direcciones asignadas
}
void loop()
{
}
```

Como se observa en las líneas anteriores la dirección IP que tiene el shiled es la 192.168.0.4

7. Empleando la consola de Windows ejecute el comando ping para verificar la comunicación entre el computador y el shield Ethernet.

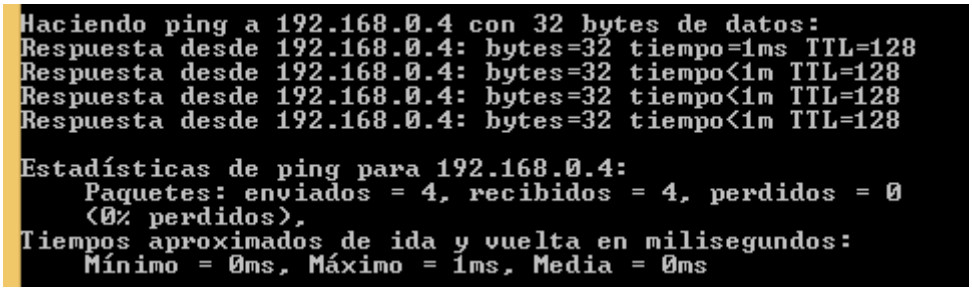


Figura 18. Comprobación de la comunicación Ethernet

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

8. Escriba las siguientes líneas en el software Arduino, el siguiente programa configura para que el pin 8 de la tarjeta Arduino Mega 2560 como salida, realiza la programación mediante el código HTML de la página web para el control del encendido y apagado de un led. Cargue el programa en la tarjeta Arduino.

```
#include <SPI.h>
#include <Ethernet.h>
//Declaración de la direcciones MAC e IP. También del puerto 80
byte mac[]={0x90,0xA2,0xDA,0x0D,0xD5,0x2D}; //MAC
IPAddress ip(192,168,0,4); //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 asignadas
  servidor.begin();
```

```
  pinMode(PIN_LED,OUTPUT);
  digitalWrite(PIN_LED,HIGH);
  state="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(readString.length()<30)//Leemos petición HTTP caracter a caracter
        {
        readString.concat(c); //Almacenar los caracteres en la variable readString
        }

        if(c=='\n' && lineaenblanco)//Si la petición HTTP ha finalizado
        {
          int LED = readString.indexOf("LED=");
          if(readString.substring(LED,LED+5)=="LED=T")
          {
          digitalWrite(PIN_LED,HIGH);
          state="ON";
          }
          else if (readString.substring(LED,LED+5)=="LED=F")
          {
          digitalWrite(PIN_LED,LOW);
          state="OFF";
          }
```

```
          //Cabecera HTTP estándar
          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>LED ON/OFF</title>");
          cliente.println("</head>");
          cliente.println("<body width=100% height=100%>");
          cliente.println("<center>");
          cliente.println("<h1>Led ON/OFF</h1>");
          cliente.print("<br><br>");
          cliente.print("Estado del LED: ");
          cliente.print(state);
          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.println("</center>");
          cliente.println("</body>");
          cliente.println("</html>");
          cliente.stop();//Cierro conexión con el cliente
          readString="";
        }
      }
    }
  }
}
```

9. Conecte un led en el pin 8 de la tarjeta Arduino Mega 2560 en serie con una resistencia de 330 Ω.
10. Abra un navegador web y escriba la dirección IP 192.168.0.4, que es la dirección del shiled Ethernet, pulse enter,

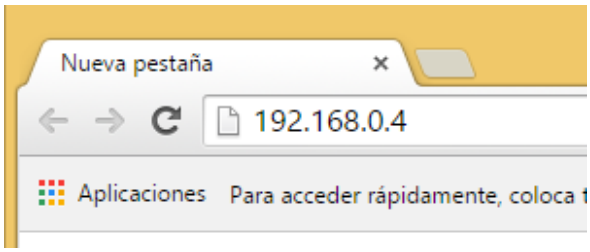


Figura 19. Dirección IP del shield Ethernet

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

11. Aparecerá la página web creada desde Arduino

Figura 20. Página web para el control de un led

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

12. Pulse el botón ON para que se encienda el led y el botón OFF para que el led se apague.

3.7 Encendido/apagado 2 leds desde página web

Los 2 leds se conectan a los pines 5 y 8 del Arduino Mega 2560, el programa creado en el software Arduino es el siguiente:

```
#include <SPI.h>
#include <Ethernet.h>
//Declaración de la direcciones MAC e IP. También del puerto 80
byte mac[]={0x90,0xA2,0xDA,0x0D,0xD5,0x2D}; //MAC
IPAddress ip(192,168,0,4); //IP

EthernetServer servidor(80);
```

```
int PIN_LED=5;
int pin_led2=8;
String readString=String(30);
String state=String(3);
String state1=String(3);

void setup()
{
  Ethernet.begin(mac, ip); //Inicializamos con las direcciones asignadas
  servidor.begin();
  pinMode(PIN_LED,OUTPUT);
  digitalWrite(PIN_LED,LOW);
   pinMode(pin_led2,OUTPUT);
  digitalWrite(pin_led2,LOW);
  state="OFF";
  state1="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(readString.length()<30)//Leemos petición HTTP caracter a caracter
        {
        readString.concat(c); //Almacenar los caracteres en la variable
readString
        }
```

```
if(c=='\n' && lineaenblanco)//Si la petición HTTP ha finalizado
{
  int LED = readString.indexOf("LED=");
  if(readString.substring(LED,LED+5)=="LED=1")
  {
  digitalWrite(PIN_LED,HIGH);
  state="ON";
  }
  else if (readString.substring(LED,LED+5)=="LED=0")
  {
  digitalWrite(PIN_LED,LOW);
  state="OFF";
  }
   if(c=='\n' && lineaenblanco)//Si la petición HTTP ha finalizado
{
  int LED = readString.indexOf("LED=");
  if(readString.substring(LED,LED+5)=="LED=2")
  {
  digitalWrite(pin_led2,HIGH);
  state1="ON";
  }
  else if (readString.substring(LED,LED+5)=="LED=3")
  {
  digitalWrite(pin_led2,LOW);
  state1="OFF";
  }
}
  //Cabecera HTTP estándar
  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>Control 2 leds</title>");
            cliente.println("</head>");
            cliente.println("<body width=100% height=100%>");
            cliente.println("<center>");
            cliente.println("<h1>Control ON/OFF</h1>");
            cliente.print("<br><br>");
            cliente.print("Estado del led1: ");
            cliente.print(state);
            cliente.print("<br><br>");
            cliente.println("<input type=submit value=ON
style=width:200px;height:75px onClick=location.href='./?LED=1\'>");
            cliente.println("<input type=submit value=OFF
style=width:200px;height:75px onClick=location.href='./?LED=0\'>");

            cliente.print("<br><br><br><br>");
            cliente.print("Estado del led2: ");
            cliente.print(state1);
            cliente.print("<br><br>");
            cliente.println("<input type=submit value=Encendido
style=width:200px;height:75px onClick=location.href='./?LED=2\'>");
            cliente.println("<input type=submit value=Apagado
style=width:200px;height:75px onClick=location.href='./?LED=3\'>");
            cliente.println("</center>");
            cliente.println("</body>");
            cliente.println("</html>");
            cliente.stop();//Cierro conexión con el cliente
            readString="";
          }
        }
      }
    }
}
```

La página web creada con Arduino se muestra en la figura

Figura 21. Página web para el control de 2 leds

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

3.8 Visualización de 3 señales analógicas en una página web

El siguiente programa adquiere 3 señales analógicas de potenciómetros conectados a los pines A0, A1 y A2 de la tarjeta Arduino Mega 2560. La página actualiza los datos cada 2 segundos. El programa realizado es el siguiente:

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

byte mac[]={0x90,0xA2,0xDA,0x0D,0xD5,0x2D}; //MAC
IPAddress ip(192,168,0,4); //IP
float vol;
EthernetServer server(80); //Es el puerto HTML para conexion a Internet

void setup() {
  Ethernet.begin(mac, ip); //Inicializa libreria y configuraciones de red
  server.begin();//Inicia comunicacion a traves del puerto
 }
```

```
void loop() {
  EthernetClient cliente = server.available();
  if (cliente) { //Una peticion http termina con una linea en blanco

    boolean lineaenblanco = true;
    while (cliente.connected()) {
      if (cliente.available()) {
        char c = cliente.read();

        if (c == '\n' && lineaenblanco) {
          //Envio encabezado estandar de respuesta HTTP
          cliente.println("HTTP/1.1 200 OK");
          cliente.println("Content-Type: text/html");

            cliente.println("Refresh: 2");   // refresca la pagina automaticamente
cada 2 sec
          cliente.println();
          cliente.println("<html>");
          cliente.println("<head>");
          cliente.println("<title>Adq Analog</title>");
          cliente.println("</head>");
          cliente.println("<body width=100% height=100%>");
          //cliente.println("<center>");
          cliente.println("<h1>Seniales Analogicas</h1>");
          cliente.print("<br><br>");
           //Imprime el valor de cada entrada analogica
          for (int canal = 0; canal < 3; canal++) {
            int sensores = analogRead(canal);
            vol=(float)sensores/1023*5;
            cliente.print("Entrada Analogica ");
            cliente.print(canal);
            cliente.print(" es ");
            cliente.print(sensores);
            cliente.println("<br />");
```

```
            cliente.print("Voltaje = ");
            cliente.print(vol);
            cliente.println("<br><br><br>");
          }

          cliente.println("</html>");
          break;
        }
        if (c == '\n') {
          lineaenblanco = true; //Comenzaremos una nueva linea
        }
        else if (c != '\r') {

          lineaenblanco = false; //Obtenemos un caracter en la linea actual
        }
      }
    }
    //Damos un tiempo al servidor web para recibir los datos
    delay(1);
    // cierra la conexión
    cliente.stop();
  }
}
```

La página web creada es la siguiente:

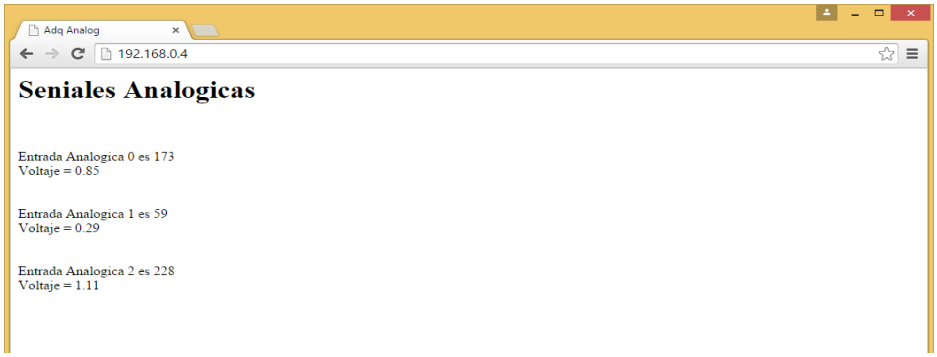


Figura 22. Página web para visualizar señales analógicas

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

3.9 Encendido/apagado led mediante Hyperterminal

Para encender un led que está conectado en el pin 13 de la tarjeta Arduino Mega 2560 mediante el hyperterminal de Windows, se utiliza el protocolo telnet. Cuando se escriba 1 en hyperterminal, el led se enciende y también Arduino envía un mensaje de encendido y se nuevamente se escribe 1 el led se apaga. El programa escrito en Arduino es el siguiente:

```
#include <SPI.h>
#include <Ethernet.h>
int a;
// Configuración de direccion MAC e IP.
byte mac[]={0x90,0xA2,0xDA,0x0D,0xD5,0x2D}; //MAC
IPAddress ip(192,168,0,4); //IP

// El puerto usado para Telnet es el 23
EthernetServer server(23);
boolean haymensaje=true;
boolean estadoled1=false, estadoled2=false,estadoled3=false;

void setup()
{
  pinMode(13,OUTPUT);

  Ethernet.begin(mac, ip);
  server.begin();
}

void loop()
{
    //nos mantenemos esperando un cliente nuevo
EthernetClient cliente=server.available();
if(cliente)
{
//leemos byte por byte
```

```
char data=cliente.read();
switch(data)
{
case '1':
digitalWrite(13,estadoled1);
cliente.print("led1");
cliente.println(estadoled1 ? "prendido":"apagado");
//si vuelve a mandar char '1' invertimos la operacion anterior
estadoled1=!estadoled1;
break;
}
}
}
```

Los pasos para crear una nueva conexión en hyperterminal son los siguientes:

1. Abrir hyperterminal y escriba un nombre para la conexión.

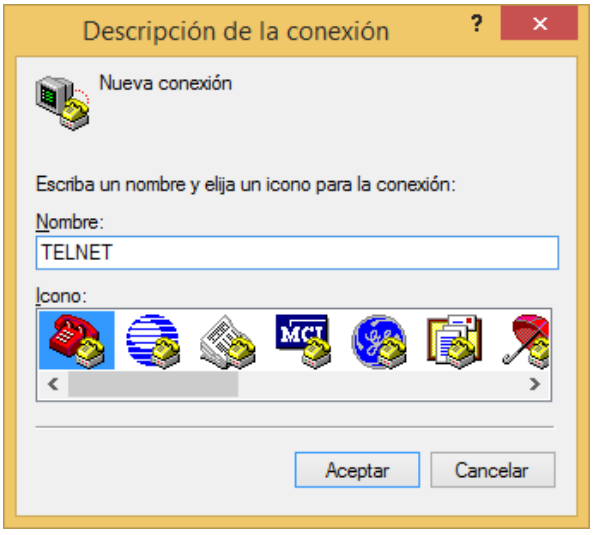


Figura 23. Nueva conexión en hyperterminal

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

2. Pulse en Aceptar, en la nueva conexión seleccione TCP/IP

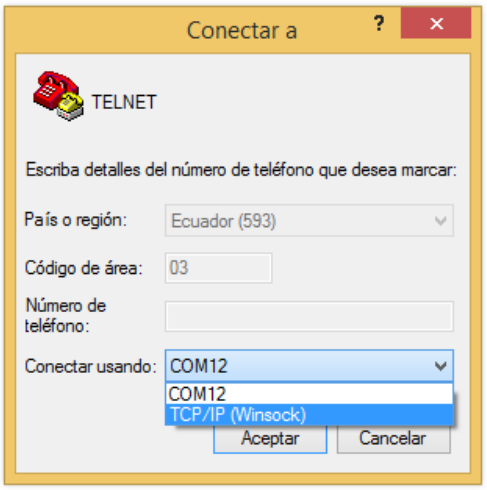


Figura 24. Telnet mediante TCP/IP

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

3. Escriba la dirección IP del shield Ethernet, pulse Aceptar

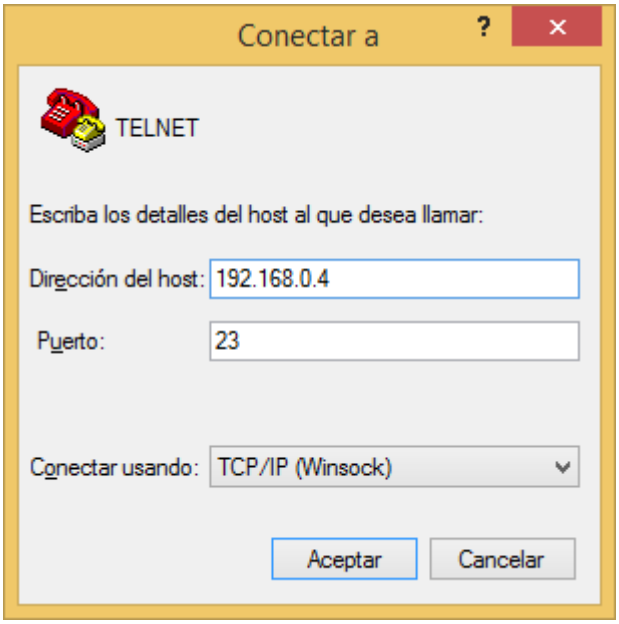


Figura 25. Asignación de IP en hyperterminal

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

4. Escriba 1 varias veces y observe los resultados

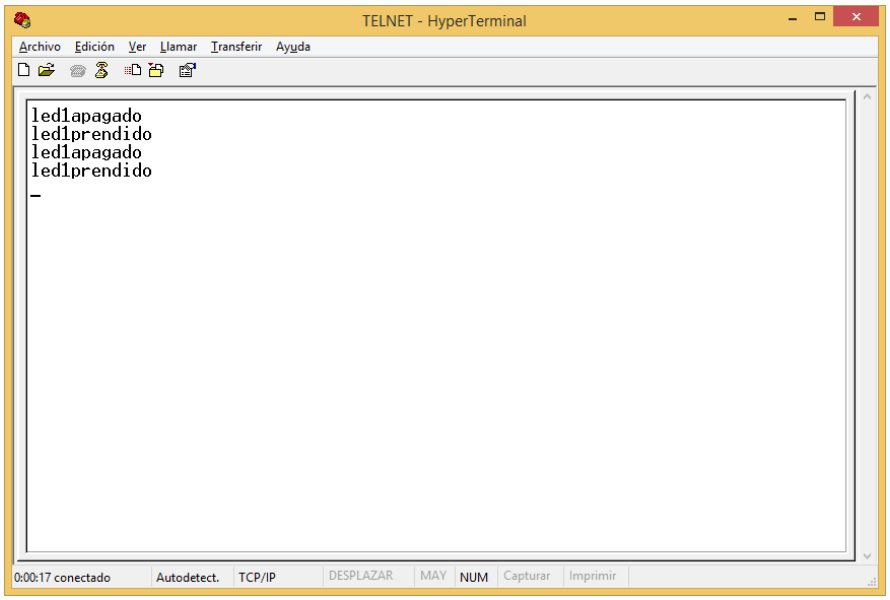


Figura 26. Conexión hyperterminal para control de led

Fuente: (Arduino, Entorno de Desarrollo de Arduino, 2014)

CAPÍTULO IV
CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Se implementó en el laboratorio de Instrumentación Virtual módulos para prácticas de comunicaciones Ethernet utilizando placas Arduino Mega 2560 y shields Ethernet.
- Las páginas web se crearon en el software Arduino utilizando lenguaje HTLM básico y se abren utilizando cualquier navegador web.
- La dirección IP asignada al shield es la 192.168.0.4 por lo que el computador puede tener un valor en el último octeto de 1 a 254 diferente de 4 por ejemplo 192.168.0.5.
- El puerto para conexiones web es el 80 y para telnet es el 23
- La distancia máxima entre el shiled Ethernet y el computador utilizando cable de red directo es 100 metros máximo de acuerdo al estándar Ethernet.
- En el presente trabajo de graduación se realizó el control de encendido y apagado de leds y la adquisición de señales de potenciómetros, pero pueden ser acondicionados para el control de dispositivos eléctricos y la adquisición de sensores.

4.2 Recomendaciones

- Si se alimenta la placa Arduino con una fuente externa el valor máximo de voltaje debe ser 12 V de acuerdo a la recomendación del fabricante.
- Los tres primeros octetos de la dirección IP se debe mantener para asegurar que se esté trabajando en la misma red.
- Observe que el led LINK y el led 100M estén encendidos, este estado indica que existe comunicación Ethernet.
- Realice con los estudiantes prácticas utilizando más de una tarjeta Arduino con shiled Ethernet.
- En el hyperterminal se debe seleccionar la forma de conexión TCP/IP para que exista la comunicación correcta.

GLOSARIO DE TÉRMINOS.

BNC: Bayonet Neil-Concelman, o a veces British Naval Connector; se usa para la interconexión de equipos y/o dispositivos en redes locales 10BASE2 Ethernet con cable coaxial

CSMA/CD: Carrier Sense Multiple Access / Collision Detect (por sus siglas en inglés) - Acceso múltiple con detección de portadora y detección de colisiones (en español)

Gbps: Giga bytes por segundo

IP: Internet Protocol [Protocolo de Internet]

LAN: Local Area Network (por sus siglas en inglés); Red de Área Local (en español)

MAC: Media Access Control [Control de acceso al medio]

Mbps: Mega bytes por segundo

SPI: Interface Periférica Serial

UTP: Unshielded Twisted Pair; Par trenzado sin apantallar

REFERENCIA BIBLIOGRÁFICA.

Arauz, H. (Abril de 2006). *Ethernet*. Obtenido de http://www.angelfire.com/planet/netstechnology/ethernet.htm

Arduino. (Septiembre de 2014). Entorno de Desarrollo de Arduino.

Arduino. (2015). *Arduino Mega 2560*. Obtenido de http://www.arduino.cc/en/Main/ArduinoBoardMega2560

Arduino. (s.f.). *Arduino Genuino*. Obtenido de http://www.arduino.cc/es/pmwiki.php?n=

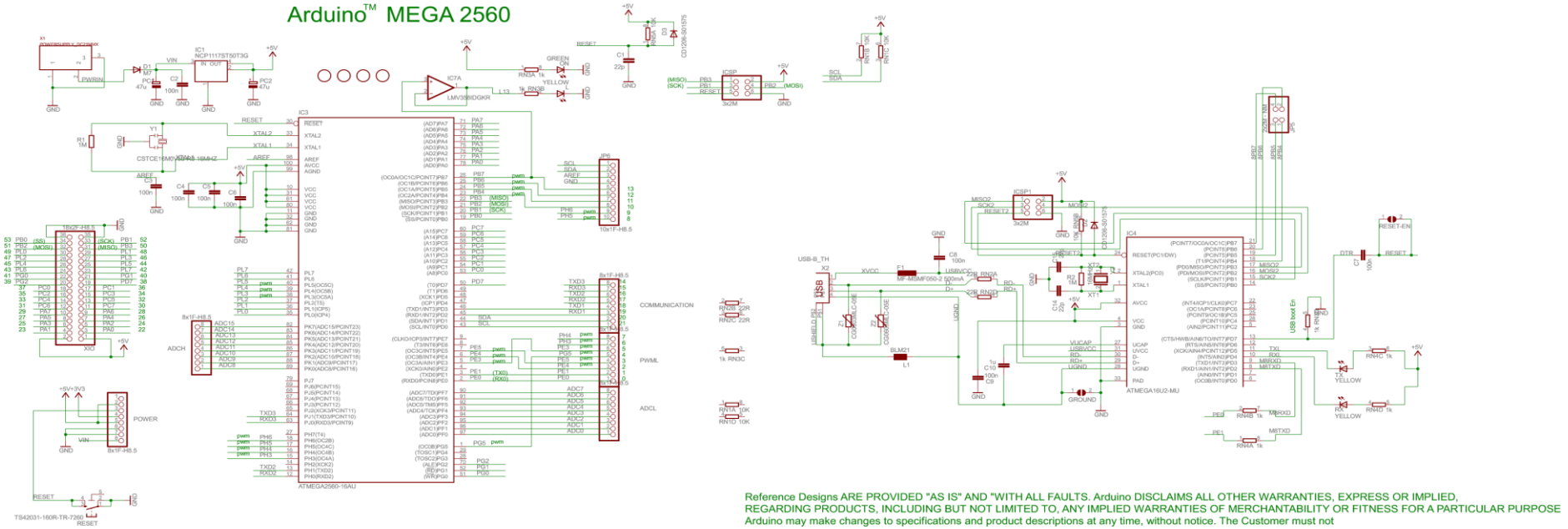
Arduino Ethernet, S. (2015). *Arduino*. Obtenido de http://www.arduino.cc/en/Main/ArduinoEthernetShield

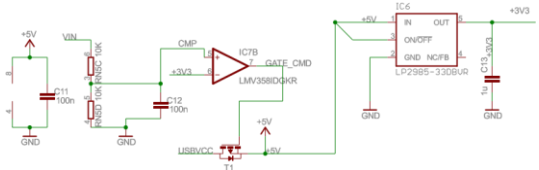
Iribar, A. (s.f.). *HTML Basico*. Obtenido de http://paginaspersonales.deusto.es/airibar/Ed_digital/HTML/HTML_1.html

ANEXOS.

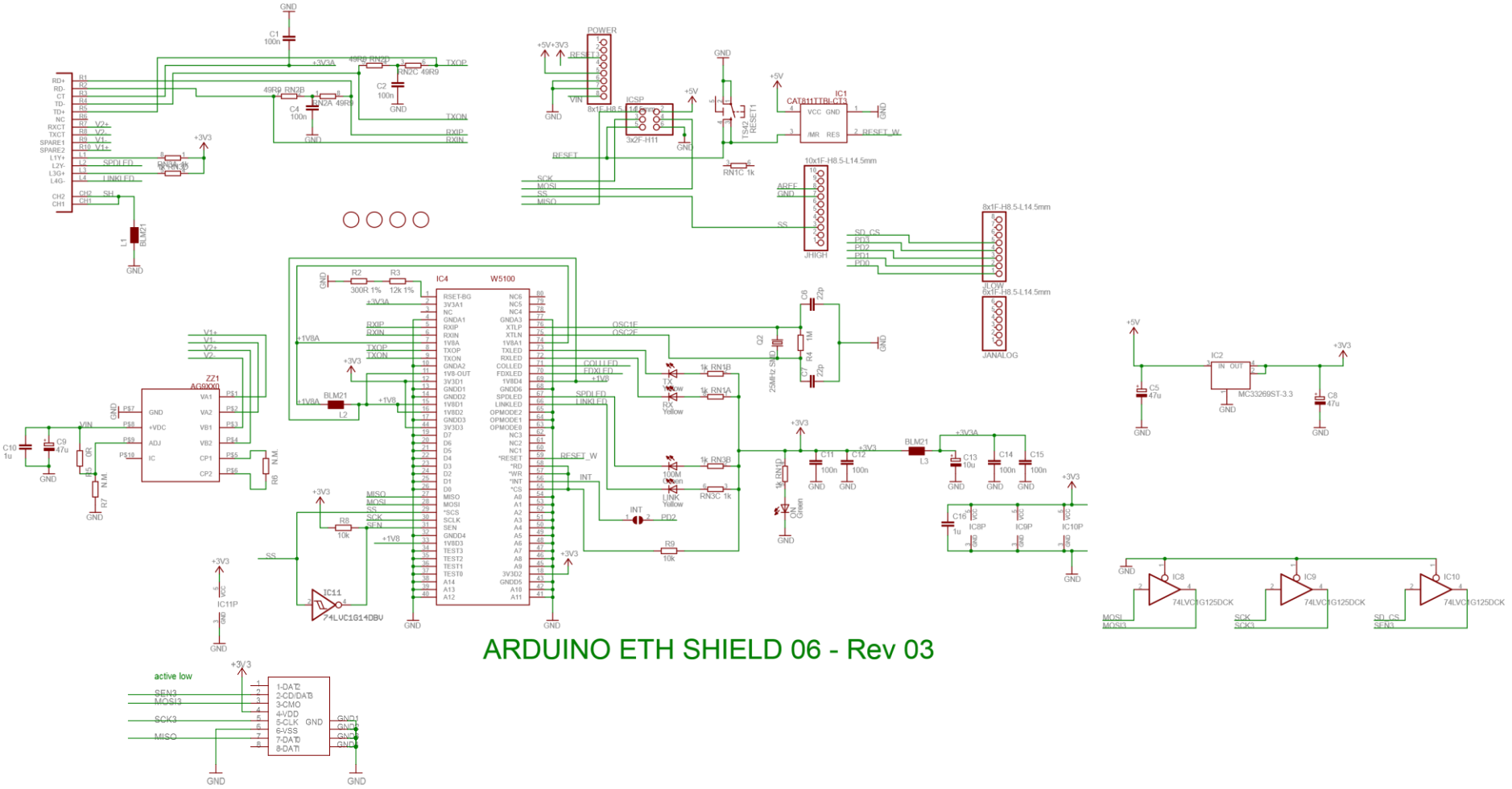
ÍNDICE DE ANEXOS

ANEXO A. Diagrama Arduino Mega 2560





ANEXO B. Diagrama shiled Ethernet



HOJA DE VIDA

DATOS PERSONALES:

NOMBRE: Maicol Daniel Ñacata Oña

NACIONALIDAD: Ecuatoriana

FECHA DE NACIMIENTO: 11/08/90

CÉDULA DE CIUDADANÍA: 1722719505

TELÉFONOS: 233520650/0987067121

CORREO ELECTRÓNICO: maicoldaniel_md1108@hotmail.es

DIRECCIÓN: Amaguaña Barrio Chaupitena

ESTUDIOS REALIZADOS:

PRIMARIA Escuela Rosario Del Alcázar N.-1

SECUNDARIA Colegio Técnico Experimental Salesiano Don Bosco

SUPERIOR Universidad de las Fuerzas Armadas ESPE

TÍTULOS OBTENIDOS:

SECUNDARIA Bachiller Industrial Electricidad-Electrónica

SUPERIOR Tecnólogo Electrónica Mención Instrumentación y Aviónica

FUERZA AEREA ECUTORIANA Sldo. Esp. Avc. Ayudante mantenimiento generadores

CURSOS Y SEMINARIOS.

Suficiencia idioma Ingles en Universidad de las FFAA ESPE

ACEPTACIÓN DEL USUARIO

Latacunga, Junio del 2015

Yo, ING PABLO PILATÁSIG en calidad de encargado del Laboratorio de Instrumentación Virtual de la Unidad de Gestión de Tecnologías, me permito informar lo siguiente:

El proyecto de graduación elaborado por el Sr. **ÑACATA OÑA MAICOL DANIEL,** con el tema: **"IMPLEMENTACIÓN DE 4 MÓDULOS PARA PRÁCTICAS DE COMUNICACIÓN ETHERNET EMPLEANDO ARDUINO MEGA PARA EL LABORATORIO DE INSTRUMENTACIÓN VIRTUAL DE LA UNIDAD DE GESTIÓN DE TECNOLOGÍAS",** ha sido efectuado de forma satisfactoria en las dependencias de mi cargo y que la misma cuenta con todas las garantías de funcionamiento, por lo cual extiendo este aval que respalda el trabajo realizado por el mencionado estudiante.

Por tanto me hago cargo de todas las instalaciones realizadas por el Sr. estudiante.

Atentamente,

_____

ING. PABLO PILATÁSIG
ENCARGADO DEL LABORATORIO DE INSTRUMENTACIÓN VIRTUAL

HOJA DE LEGALIZACIÓN DE FIRMAS

DEL CONTENIDO DE LA PRESENTE INVESTIGACIÓN SE RESPONSABILIZA EL AUTOR

ÑACATA OÑA MAICOL DANIEL

C.I: 1722719505

DIRECTOR DE LA CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA

Ing. Pablo Pilatásig Director Carrera de Electrónica Mención Instrumentación & Aviónica

Latacunga, Junio del 2015