

**MONITOREO DE PRESION DE REFRIGERANTE EN SISTEMA DE  
AIRE ACONDICIONADO Y CONTROL DE ACCESO A TRAVÉS DE  
ARDUINO**

**JORGE EDUARDO OSORIO CORREA  
PROYECTO DE GRADO**

**UNIVERSIDAD CATOLICA DE MANIZALES  
FACULTAD DE INGENIERIA  
PREGRADO INGENIERIA TELEMATICA  
MANIZALES  
DICIEMBRE DE 2016**

# Índice general

1. Introducción
2. Definición del problema
3. Justificación
4. Objetivos
  - 4.1 Objetivo general
  - 4.2 Objetivo específico
5. Marco teórico
6. Arduino y elementos utilizados
7. Programas individuales para proyecto con arduino
  - 7.1 Captura de foto con cámara serial a través de arduino
  - 7.2 Control remoto on-off a través de servidor web y Ethernet shield arduino
  - 7.3 Acceso imagen remotamente a través de servidor web y Ethernet shield arduino
  - 7.4 Envío de mensaje de texto a través de modem GPRS sim 900.
  - 7.5 Encender un led con sensor PIR
8. Aire acondicionado
9. Desarrollo
10. Resultados
11. Conclusiones
12. Referencias

## **1. Introducción**

El telecontrol e internet de las cosas cada día tiene un mayor campo de aplicación y necesidad de ser usado en diferentes áreas de la industria, automatización, domótica, control ambiental o telemedicina. Todos los días surgen nuevas necesidades de monitorear, medir o controlar variables, procesos, o situaciones, con el fin de ejercer una acción o para obtener diferentes tipos de información y datos del comportamiento de los mismos.

Estos datos o variables se pueden controlar en tiempo real, lo que hace que se pueda dar una respuesta o ejercer un control de forma inmediata, de igual forma las mediciones y datos se pueden almacenar para contar por ejemplo con un historial al cual se pueda acceder cuando se requiera, ejemplo de lo anterior una base de datos con el sistema de gestión Mysql de código abierto.

En el mercado podemos encontrar cientos de sensores para medir diferentes variables, obteniendo los datos e información para ejercer bien sea un control, una acción o un seguimiento, como por ejemplo la temperatura, la humedad, caudal, movimiento, nivel, humo, presión, todo esto se tiene aplicación bien sea a nivel industrial, comercial residencial o médico, cada día vemos más elementos son controlados remotamente inclusive las personas, no controladas pero sí monitoreadas.

El proyecto tiene como finalidad monitorear la presión del refrigerante en un sistema de aire acondicionado de un datacenter, un sitio de distribución de señal, nodo u otro sitio, también controla el acceso y apertura de la puerta de este. Con la combinación de un microcontrolador, sensores y otros elementos se pueden se puede controlar, medir, monitorear no solo de forma local si no remota diferentes variables a través de una red de datos como una red LAN, internet o también a través de otras redes como GSM, GPRS.

Los datacenter y sitios de distribución de señal cada vez más adquieren un mayor valor, la información y datos almacenados allí se convierten en el activo más valioso para cualquier empresa, es por eso que es imprescindible contar con las precauciones y cuidado que este requiere.

## **2. Definición del problema**

La necesidad de usar aires acondicionados es cada vez mayor en el mundo actual bien sea en los hogares, industria, comercio u otra área, incluso los aumentos de temperatura por el cambio climático han obligado utilizar estos sistemas en sitios donde décadas atrás no era necesario. Estos sistemas pueden fallar en su funcionamiento y desempeño normal por diferentes causas; este proyecto tiene como finalidad principal detectar la pérdida o escape del líquido o gas refrigerante, el cual es un componente muy importante en el sistema de refrigeración o aire acondicionado. La falta de gas puede ocurrir debido a una fuga en la tubería por una mala instalación del aire acondicionado o porque ha aparecido algún orificio por donde se escapa el refrigerante en la tubería o soldadura de la misma, esto puede presentarse por la vibración que se genera en la unidad condensadora o en la manejadora, o por la corrosión o por un golpe o manipulación en la tubería.

La ausencia o falta de este refrigerante perjudica el normal funcionamiento del aire, llegando a causar el daño del compresor, con lo cual el sistema de aire no trabajaría. Con la falta de presión de este líquido/gas y sin su circulación en la tubería, el aire caliente no vá poder extraerse del sitio, lo cual con lleva a la afectación en el normal funcionamiento de los equipos del sitio o sitios que deben ser refrigerados para permanecer a temperatura ambiente preferiblemente entre 20 y 23 grados centígrados. Con el aumento de temperatura global es necesario contar con un sistema de aire confiable en los centros de datos y de transmisión, ya que la falta de refrigeración en estos sitios puede hacer que la temperatura aumente a valores superiores a 40 grados centígrados.

### **3. Justificación**

El presente proyecto se pretende desarrollar principalmente con el fin de crear una forma de monitorear la presión del refrigerante en un sistema de aire acondicionado bien sea un datacenter, nodo, o sitio de distribución de señal de internet, televisión u otro sitio. Esta presión de refrigerante en los aires acondicionados se puede decir que ninguna empresa la monitorea en todo momento y en tiempo real, solo verifican el valor de dicha presión cuando se realiza un mantenimiento preventivo o correctivo al sistema de aire.

La falta o disminución en la presión de este refrigerante hace que el sistema no tenga un correcto funcionamiento o se presente un daño en el mismo. Los datacenter, nodos o centrales de comunicaciones deben contar con un aire acondicionado o refrigeración las 24 horas 365 días al año, con el fin de evitar elevadas temperaturas que puedan afectar los equipos y la normal prestación de los servicios. Un datacenter tiene el activo más valioso para una empresa el cual es la información, este activo puede estar por ejemplo almacenado en servidores, la ausencia o falla de un sistema acondicionado de aire, puede perjudicar estos servidores entorpeciendo la disponibilidad o causando la pérdida de este activo, lo cual puede ocasionar pérdidas muy grandes o incluso incalculables para la organización. De acuerdo a lo anterior es necesario monitorear en tiempo real y 7x24x365 días estos sistemas de aire,

### **4. Objetivos**

#### **4.1 Objetivo general**

Implementar un sistema de monitoreo de presión de refrigerante en un aire acondicionado y control de acceso a sitio remoto.

#### **4.2 Objetivos específicos**

- Implementar un código para capturar fotografías a través de una cámara serial TTL y arduino.

- Desarrollar un código para ejercer el control remoto on-off a través de servidor web y Ethernet shield arduino.
- Implementar un código para acceder a una imagen remotamente a través de un servidor web montado en la tarjeta SD en la Ethernet shield arduino
- Implementar un código para enviar un mensaje de texto a teléfono móvil como alerta por variación en presión de refrigerante en un sistema de aire acondicionado.

## **5. Marco teórico**

Para desarrollar un sistema de monitoreo de presión de refrigerante en un aire acondicionado y control de acceso, se debe tener conocimiento y conceptos claros de diversos temas como servicios web, peticiones cliente-servidor, estructuras de programas en java, diferencias entre de señales digitales y análogas con el fin de obtener la información adecuada con los datos suministrados por estas señales y por lo tanto poder aprovecharla de la mejor manera para nuestro objetivo. En el mercado existen diferentes soluciones para los controles de acceso pero no para el monitoreo remoto de la presión del refrigerante en un aire acondicionado. La tecnología y la informática se han convertido en parte fundamental de la evolución de la industria y en si del ser humano, posiblemente décadas atrás o incluso desde la creación del semiconductor o primer computador el ENIAC en 1946 nadie imaginaba hasta donde la evolución tecnológica influiría de manera tan drástica en nuestras vidas. De igual forma nadie tiene la certeza hasta donde evolucionará o tendrá un punto máximo en el cual no pueda avanzar.

Los conceptos más importantes a tener en cuenta para el desarrollo de este proyecto son los siguientes.

Un servidor web o servidor http es un programa que se ejecuta en un computador en todo momento y está a la espera de alguna petición por parte del cliente que puede ser otro computador. Estas peticiones pueden ser síncronas o asíncronas. Al responder a estas

peticiones por medio de un lenguaje como HTML, el computador cliente crea la página web con los datos requeridos.

Una señal digital tiene un valor único en el tiempo, es discontinua y solo puede estar en dos estados uno o un cero (bits) o high y low u on y off.

Una señal analógica es continua en el tiempo y puede tomar infinitos valores, siempre tiene valores intermedios para pasar de un valor a otro.

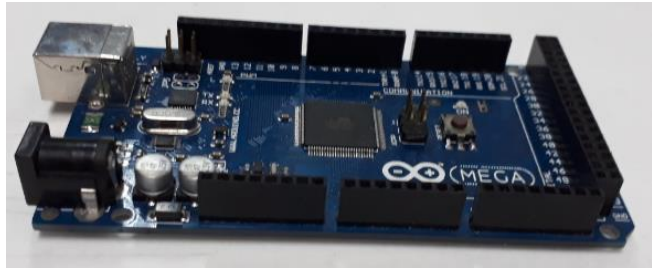
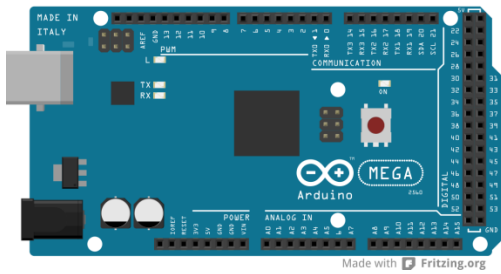
## **6. Arduino y Elementos utilizados**

Arduino es una plataforma de prototipos electrónica de código abierto basada en hardware y software fáciles de usar. La placa electrónica arduino tiene como componente principal un microcontrolador, además está compuesta por otros componentes y circuitos electrónicos, la placa tiene entradas y salidas analógicas, como entradas se pueden usar cientos de sensores con lo cual se puede monitorear y capturar múltiples señales y variables, cada vez más crean nuevos tipos de sensores con lo cual arduino se convierte en una herramienta de monitoreo y control muy completa y fácil de usar. Los proyectos de arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un pc. Los diseños de referencia, al ser código abierto se pueden utilizar y modificar para nuestro propio requerimiento. Arduino cuenta con shields o tarjetas de expansión con el fin de agregar más opciones, por ejemplo conectar la placa a internet a través wifi o un puerto LAN, o conexiones a redes GPRS o GSM. La placa arduino utiliza un convertidor USB-Serial para comunicarse con el computador, de esta forma la placa recibe los códigos o programas de forma serial.

Existen varias placas arduino como Duemilanove, Uno, Leonardo, Mega, este último es el que se utilizará en este proyecto.

La placa arduino Mega 2560 posee un microcontrolador ATmega 2560, memoria flash de 128 kB, un oscilador de 256Mhz, voltaje de entrada de 7 a 12 VCC, 54 entradas/salidas (40ma) de los cuales 16 son entradas analógicas, 4 puertos UART (puertos seriales hardware), fuentes o salidas de voltaje de 5 y 3.3V.

## Arduino 2560



## Especificaciones técnicas

Microcontrolador	ATmega2560
Voltaje de operación	5V
Voltaje de entrada	7-12V
Pines Digitales I/O	54 (15 con salida PWM)
Entradas pines análogas	16
Corriente I/O Pines	20 mA
Corriente para 3.3V Pin	50 mA
Memoria Flash	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Velocidad reloj	16 MHz
LED_BUILTIN	13
Ancho	101.52 mm
Alto	53.3 mm
Peso	37 g

## Shield SIM 900 para arduino

GPRS Shield SIM900 es una tarjeta compacta de comunicación inalámbrica con el chip SIM900. La tarjeta es compatible con todos los modelos de Arduino, también es compatible con otros microcontroladores. La tarjeta está basada en el módulo SIM900 que se maneja con el protocolo serial.

La tarjeta GPRS Shield SIM900 se puede configurar y controlar por vía UART usando comandos AT. Ideal para sistemas remotos, puntos de control, mandar mensajes de texto y llamadas a celulares, entre otros. Esta tarjeta no se maneja con bibliotecas en particular debido a que los comandos son solamente caracteres que se envían por el puerto serial.



Esta tarjeta es Hardware Libre al igual que las placas arduino y se le puede dar el uso que se requiera.

Especificaciones

Quad-Band 850/ 900/ 1800/ 1900 MHz

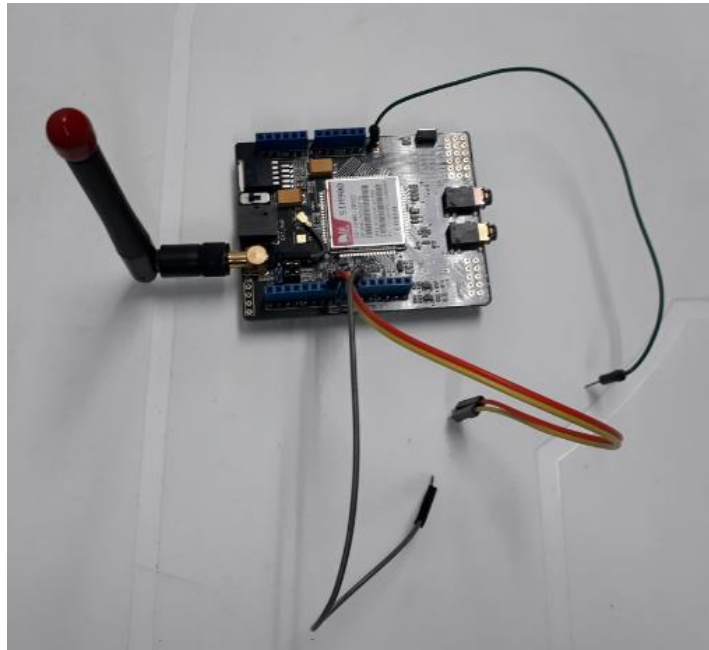
GPRS mobile station clase B

Compatible con GSM

Clase 4 (2W (AT) 850/900Mhz)

Clase 1 (1W (AT) 1800/1900Mhz)

TCP/UP embebido



## Shield Ethernet



Ethernet shield sobre el arduino mega 2560

## Cámara serial TTL para arduino

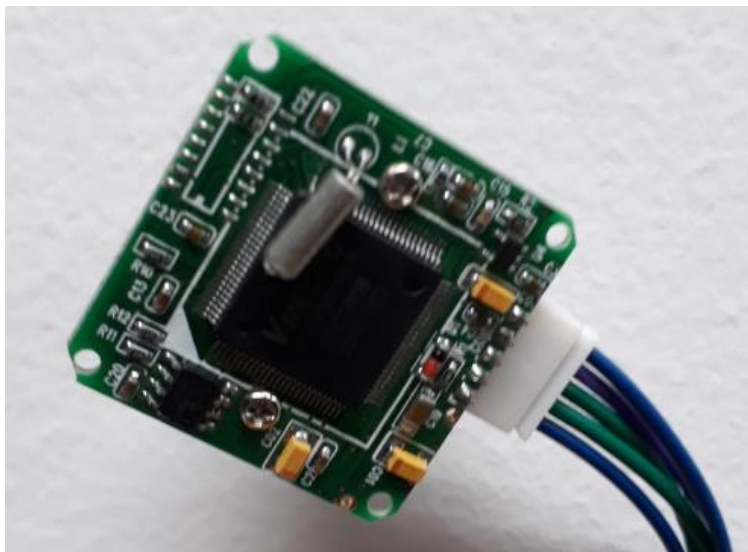
Es una cámara monocromática que captura imágenes y las transmite en forma serial, estas imágenes son en formato JPEG de 640x480, 320x240, 160x120, las cuales se pueden almacenar en una memoria sd o microsd, además cuenta con enfoque manual autoajustable balance de blancos, ajuste de brillo y contraste automáticos, cuenta con un detector de movimiento o sensor para capturar imágenes, transmite a 38400 baudios.

Para utilizar esta cámara con arduino es necesario habilitar una librería en el arduino, la cual es un conjunto de programas que al ejecutarse hace que la cámara pueda tomar la foto.

Esta cámara trabaja con un reloj interno a 16 Mhz el cual trae incorporado, existen otras cámaras más económicas que no traen este reloj y es necesario adaptarle uno externo, lo cual complica un poco la utilización de la misma y no garantiza su correcto funcionamiento.

Esta aunque es sencilla cámara cumple para ser usada en sistemas de seguridad y monitoreo u otra aplicación que se le quiera dar.

## Cámara serial TTL VC0706



Tamaño: 32mm x 32mm

Sensor de Imagen: CMOS 1/4 inch

CMOS Pixels: 30M

Tamaño de Pixel: 5.6um\*5.6um

Formato de Salida: Standard JPEG

Balance de Blancos: Automática

Exposición: Automática

Ganancia: Automática

Shutter: Electrónica rolling shutter

SNR: 45dB

Rango Dinamico: 60dB

Máxima Ganancia: 16dB

Frame Speed: 640\*480 30fps

Scan Mode: Scan progresivo

Angulo de Vision: 60°

Distancia: 10 metros, Maximo 15 metros (ajustable)

Tamaño de Imagen: VGA(640\*480), QVGA(320\*240), QQVGA(160\*120)

Baudios: Por defecto 38400, Maximo 115200

Consumo: 75mA

Voltaje Operativo: +5V DC

Comunicación: 3.3V TTL (3 cables TX, RX, GND)

### **Sensor de movimiento infrarrojo PIR**

Los sensores infrarrojos pasivos (PIR) son dispositivos para la detección de movimiento. Son baratos, pequeños, de baja potencia, y fáciles de usar. Por esta razón son frecuentemente usados en juguetes, aplicaciones domésticas o sistemas de seguridad.

Los sensores PIR se basan en la medición de la radiación infrarroja. Todos los cuerpos (vivos o no) emiten una cierta cantidad de energía infrarroja, mayor cuanto mayor es su temperatura. Los dispositivos PIR disponen de un sensor piezo eléctrico capaz de captar esta radiación y convertirla en una señal eléctrica.

En realidad cada sensor está dividido en dos campos y se dispone de un circuito eléctrico que compensa ambas mediciones. Si ambos campos reciben la misma cantidad de infrarrojos la señal eléctrica resultante es nula. Por el contrario, si los dos campos realizan una medición diferente, se genera una señal eléctrica.

De esta forma, si un objeto atraviesa uno de los campos se genera una señal eléctrica diferencial, que es captada por el sensor, y se emite una señal digital.



## Especificaciones

Voltaje de funcionamiento: DC 4.5V-20V-

Corriente en reposo: menos de 60uA

Salida de tensión: Alta / Baja de nivel de señal: 3,3 V de salida TTL

Distancia de detección: 3 - 7M (puede ser ajustado)

Ángulo de detección: <math><110^\circ</math>

Tiempo de retardo: 5-200S (se puede ajustar, por defecto 5s + -3%)

Tiempo de bloqueo: 2.5S (por defecto)

Método de disparo: L: irrepitable gatillo H: disparador repetible (por defecto)

Fotosensible ajuste: CDS usar (por defecto: incluido, no soldadas)

Temperatura de trabajo: -20 --- 80?

Peso del producto: 6g

Tamaño del producto: 3.2 \* 2.4 \* 2.4cm

### **Switch Dlink**

Para la conexión a nivel local se utilizó un switch de 8 puertos, se usan 3 puertos, para el cable modem inalámbrico quien asigna por DHCP las ip para los equipos de nuestra red LAN, shield Ethernet para la comunicación del arduino y computador o celular para acceder al servidor web para ver la foto tomada y realizar la conexión http para abrir la puerta. En este caso como es a nivel local también se podría asignar una ip fija privada al pc y al shield Ethernet por ejemplo (192.168.1.110 y .120) para no utilizar el cable modem, pero no se contaría con conexión wifi para la conexión a través del celular.



El cable modem es el que provee el servicio de internet de cualquier operador.

## Microsd

Se utilizó una memoria microsd de 8 Gigas aunque se puede utilizar de menor capacidad, ya que la página web y la foto que captura la cámara ocupan menos de 300 kilobytes.



## 7. Programas individuales para proyecto con arduino

Para el desarrollo de este proyecto primero se realizó el diseño, código y montaje por partes o por cada componente, la captura de una foto con la cámara serial TTL para arduino, el control remoto on-off a través de servidor web y Ethernet shield arduino con el cual se puede controlar la apertura de una puerta u otro elemento, acceso a una imagen remotamente a través de servidor web en microsd y Ethernet shield arduino, envío de mensaje de texto a través del modem GPRS SIM 900.

Este proyecto se desarrolla a nivel local o LAN, para realizar las conexiones al servidor web y http desde afuera como internet, es necesario contratar una ip fija con el operador de internet o utilizar un servicio de dominio dinámico como DynDNS.

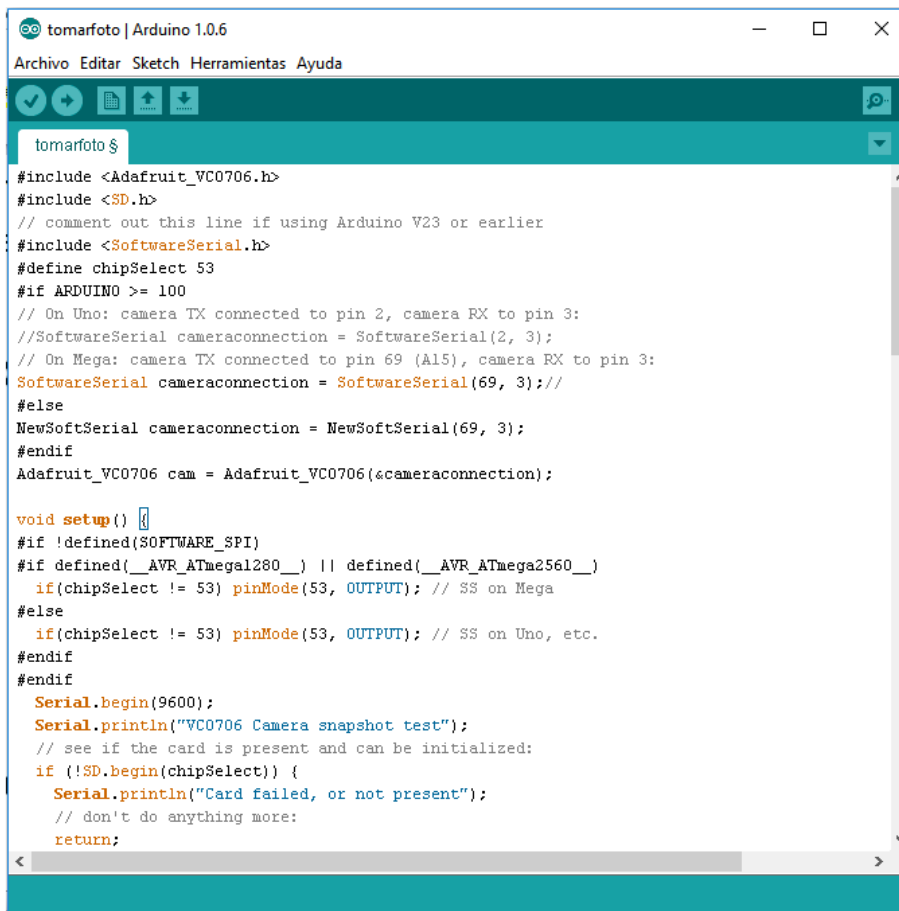
La facilidad que tiene arduino para adaptar diferentes dispositivos, además de la cantidad de elementos que se pueden encontrar en el mercado, hace que este proyecto se pueda aplicar o adaptar para otras necesidades de control y monitoreo.

Para el servidor web que aloja la foto, se utiliza el puerto 80 que es utilizado por defecto para páginas web, para la página web con la cual se abre la puerta se utiliza el puerto 2000.

## 7.1 Captura de foto con cámara serial a través de arduino

EL código al igual que la librería para la cámara se tomó del ejemplo que aparece en la página de adafruit y otros códigos libres que aparecen en internet y se citan en el ítem referencias del presente proyecto. Igualmente se realizaron adecuaciones a estos códigos para los requerimientos del proyecto.

Adafruit es una empresa que vende la cámara y otros elementos compatibles con arduino como el shield GPRS sim 900, además en su página comparte códigos y librerías que pueden ser descargados libremente para usar y adecuar según las necesidades de los usuarios.



```
tomarfoto | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda

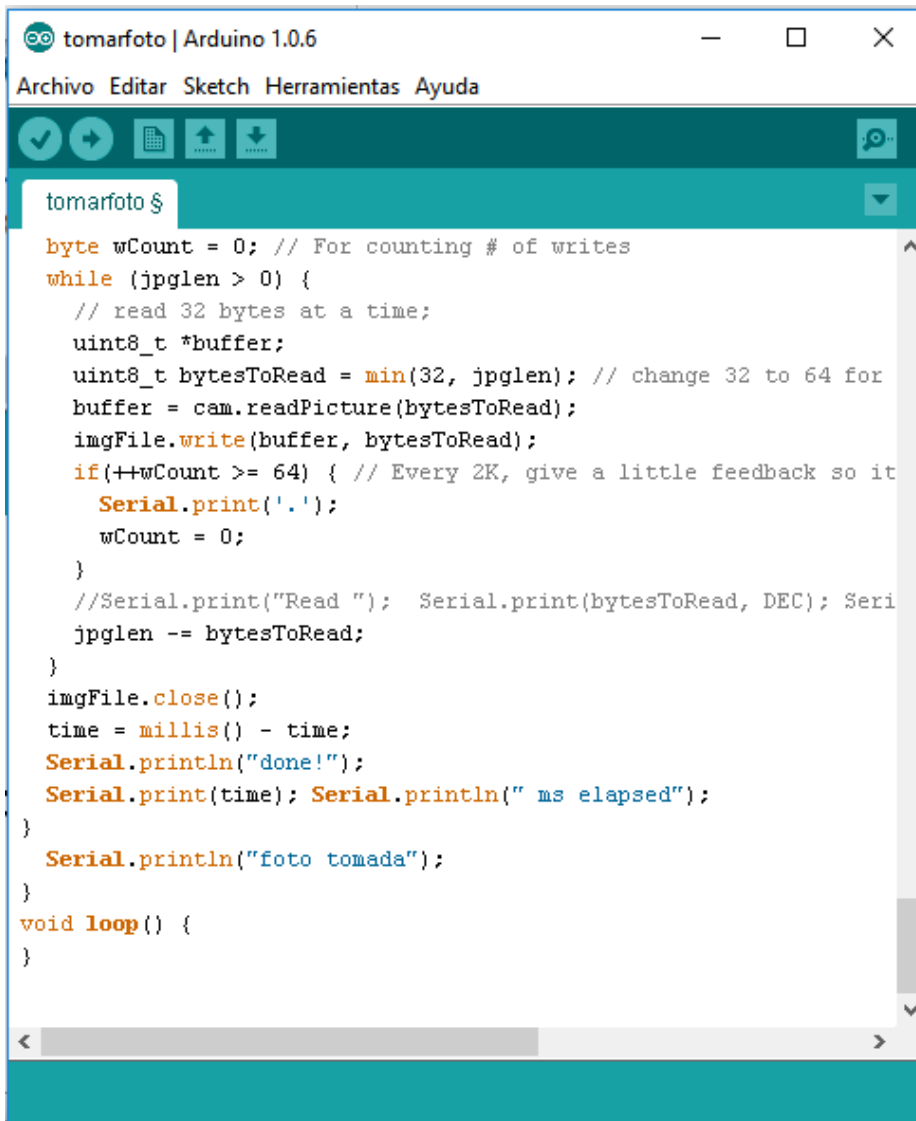
tomarfoto $
#include <Adafruit_VC0706.h>
#include <SD.h>
// comment out this line if using Arduino V23 or earlier
#include <SoftwareSerial.h>
#define chipSelect 53
#if ARDUINO >= 100
// On Uno: camera TX connected to pin 2, camera RX to pin 3:
//SoftwareSerial cameraconnection = SoftwareSerial(2, 3);
// On Mega: camera TX connected to pin 69 (A15), camera RX to pin 3:
SoftwareSerial cameraconnection = SoftwareSerial(69, 3);//
#else
NewSoftSerial cameraconnection = NewSoftSerial(69, 3);
#endif
Adafruit_VC0706 cam = Adafruit_VC0706(&cameraconnection);

void setup() {
#if !defined(SOFTWARE_SPI)
#if defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
  if(chipSelect != 53) pinMode(53, OUTPUT); // SS on Mega
#else
  if(chipSelect != 53) pinMode(53, OUTPUT); // SS on Uno, etc.
#endif
#endif
  Serial.begin(9600);
  Serial.println("VC0706 Camera snapshot test");
  // see if the card is present and can be initialized:
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    // don't do anything more:
    return;
  }
}
```



```
tomarfoto | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda
tomarfoto $
}
// Try to locate the camera
if (cam.begin()) {
  Serial.println("Camera Found:");
} else {
  Serial.println("No camera found?");
  return;
}
// Print out the camera version information (optional)
char *reply = cam.getVersion();
if (reply == 0) {
  Serial.print("Failed to get version");
} else {
  Serial.println("-----");
  Serial.print(reply);
  Serial.println("-----");
}
// Set the picture size - you can choose one of 640x480, 320x240 or 160x120
// Remember that bigger pictures take longer to transmit!
cam.setImageSize(VC0706_640x480); // biggest
//cam.setImageSize(VC0706_320x240); // medium
//cam.setImageSize(VC0706_160x120); // small
// You can read the size back from the camera (optional, but maybe useful?)
uint8_t imgsize = cam.getImageSize();
Serial.print("Image size: ");
if (imgsize == VC0706_640x480) Serial.println("640x480");
if (imgsize == VC0706_320x240) Serial.println("320x240");
if (imgsize == VC0706_160x120) Serial.println("160x120");
Serial.print("Ponga pin 8 en alto para tomar foto");
Serial.println("");
```

```
tomarfoto | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda
tomarfoto $
pinMode(8, INPUT);
tomar=digitalRead(8);
if (tomar==HIGH)
{ Serial.println("captura in 3 segundos...");
  delay(3000);
  if (! cam.takePicture())
    Serial.println("Failed to snap!");
  else
    Serial.println("Picture taken!");
  // Create an image with the name IMAGExx.JPG
  char filename[13];
  strcpy(filename, "IMAGE00.JPG");
  for (int i = 0; i < 100; i++) {
    filename[5] = '0' + i/10;
    filename[6] = '0' + i%10;
    // create if does not exist, do not open existing, write, sync
    if (! SD.exists(filename)) {
      break;
    }
  }
  // Open the file for writing
  File imgFile = SD.open(filename, FILE_WRITE);
  // Get the size of the image (frame) taken
  uint16_t jpglen = cam.frameLength();
  Serial.print("Storing ");
  Serial.print(jpglen, DEC);
  Serial.print(" byte image.");
  int32_t time = millis();
  pinMode(8, OUTPUT);
  // Read all the data up to # bytes!
```



```
tomarfoto | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda
tomarfoto $
byte wCount = 0; // For counting # of writes
while (jpglen > 0) {
  // read 32 bytes at a time;
  uint8_t *buffer;
  uint8_t bytesToRead = min(32, jpglen); // change 32 to 64 for
  buffer = cam.readPicture(bytesToRead);
  imgFile.write(buffer, bytesToRead);
  if(++wCount >= 64) { // Every 2K, give a little feedback so it
    Serial.print('.');
    wCount = 0;
  }
  //Serial.print("Read "); Serial.print(bytesToRead, DEC); Seri
  jpglen -= bytesToRead;
}
imgFile.close();
time = millis() - time;
Serial.println("done!");
Serial.print(time); Serial.println(" ms elapsed");
}
Serial.println("foto tomada");
}
void loop() {
}
```

A pesar de que la cámara puede tomar fotografías a mayor resolución se configuró de 160x120, con esta es suficiente para tomar la foto en el sitio y proceder a dar apertura a la puerta, la fotografía siguiente fue tomada con esta cámara a dicha resolución.



## **7.2 Control remoto on-off a través de servidor web y Ethernet shield arduino.**

Con este programa se abre la puerta a través de una conexión http a través de un navegador web, en el arduino se crea la página web en formato HTML en la cual se muestran dos botones para abrir o cerrar la puerta, además muestra el estado de la misma, abierta o cerrada. Al seleccionar un botón en la página, esta orden es enviada por el método GET al arduino para colocar en HIGH en una salida digital y activar un relé abriendo o cerrado la puerta a través de un electroimán.

La apertura de la puerta se simula encendiendo o apagando un led, colocándole un HIGH o un LOW.

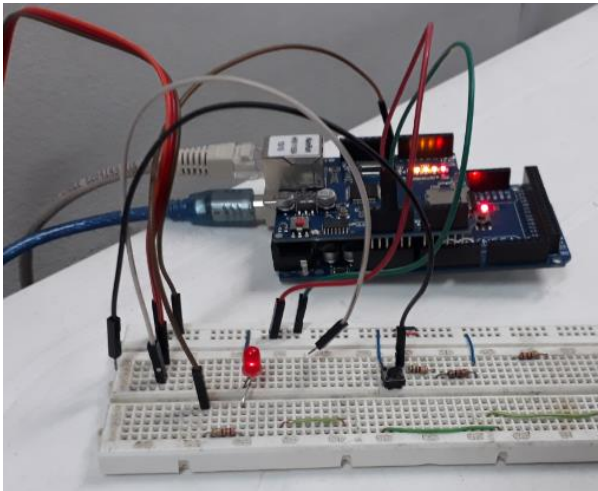
## Impresión de arduino a través del puerto serial



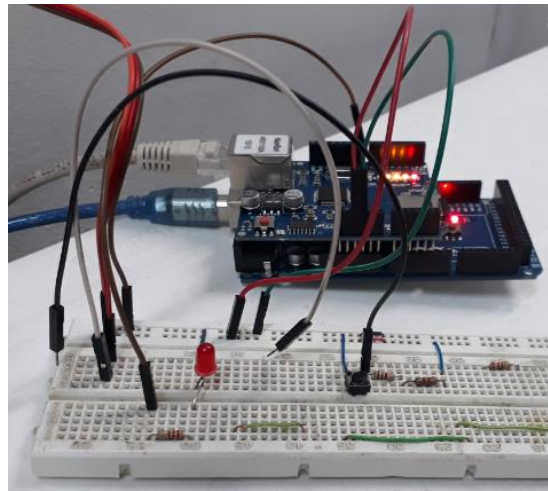
```
COM3
El servidor esta en :192.168.1.110
new client
GET / HTTP/1.1
Host: 192.168.1.110
Connnew client
GET /favicon.ico HTTP/1.1
Host: 192.168.1.110
Conn
```

La siguiente es la página web para encender o apagar el led remotamente desde una red local o internet. Al seleccionar un botón ON u OFF el led prende o apaga





Led encendido



Led apagado

```
controlar_in_out_desde_un_navegador_web | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda

controlar_in_out_desde_un_navegador_web $
#include <SPI.h>
#include <Ethernet.h>

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; //Ponemos la dirección MAC de la Ethernet Shield
IPAddress ip(192,168,1,110); //Asignamos la IP al Arduino
EthernetServer server(80); //Creamos un servidor Web con el puerto 80 que es el puerto HTTP por defecto

int LED=8; //Pin 8 salida, entrega un high al LED
String estado1="ON"; //Estado del led inicialmente "ON"

void setup()
{
  Serial.begin(9600);

  // Inicializamos la comunicación Ethernet y el servidor
  Ethernet.begin(mac, ip);
  server.begin();

  Serial.print("El servidor esta en :");
  Serial.println(Ethernet.localIP());

  pinMode(LED,OUTPUT);
  pinMode(5,INPUT); //Entrada a verificar
  digitalWrite(LED,HIGH);
}

void loop()
{
  <
  >
}

Carga terminada.
Tamaño binario del Sketch: 16.318 bytes (de un máximo de 256.048 bytes)

Activar Windows
Ve a Configuración para activar Windows.

Arduino Mega 2560 en Mega ADK en COM3
```

```
controlar_in_out_desde_un_navegador_web | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda

controlar_in_out_desde_un_navegador_web $
}

void loop()
{
  EthernetClient client = server.available(); //Creamos un cliente Web
  //Verificamos si se detecte un cliente a través de una petición HTTP
  if (client) {
    Serial.println("new client");
    boolean currentLineIsBlank = true; //Una petición HTTP acaba con una línea en blanco
    String cadena=""; //Creamos una cadena de caracteres vacía
    while (client.connected()) {
      if (client.available()) {
        char c = client.read(); //Leemos la petición HTTP carácter por carácter
        Serial.write(c); //Visualizamos la petición HTTP por el Monitor Serial
        if(cadena.length()<50)
        {
          cadena.concat(c); //concatenamos el String 'cadena' con la petición HTTP (c). De esta manera convertimos la petición HTTP a un String
          //Ya que hemos convertido la petición HTTP a una cadena de caracteres, ahora podemos buscar partes del texto.
          int posicion=cadena.indexOf("Data="); //Guardamos la posición de la Palabra "Data=" a la variable 'posicion'

          if(cadena.substring(posicion)=="Data=1")//Si en la posición hay "Data=1"
          {
            digitalWrite(LED,HIGH);
            estadol="ON";
          }
          else if(cadena.substring(posicion)=="Data=2")//Si en posición hay "Data=2"
          {

```

Carga terminada.

Tamaño binario del Sketch: 16.318 bytes (de un máximo de 258.048 bytes)

Activar Windows  
Ve a Configuración para activar Windows.

23 Arduino Mega 2560 or Mega ADK on COM3

```
controlar_in_out_desde_un_navegador_web | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda

controlar_in_out_desde_un_navegador_web $
    estadol="ON";
  }
  else if(cadena.substring(posicion)=="Data=2")//Si en posición hay "Data=2"
  {
    digitalWrite(LED,LOW);
    estadol="OFF";
  }
}

//Cuando reciba una línea en blanco, quiere decir que la petición HTTP ha acabado y el servidor Web está listo para enviar una respuesta
if (c == '\n' && currentLineIsBlank) {

  // Enviamos al cliente una respuesta HTTP
  client.println("HTTP/1.1 200 OK");
  client.println("Content-Type: text/html");
  client.println();

  //Página web en formato HTML
  client.println("<html>");
  client.println("<head<title>Maylamp Mechatronics</title>");
  client.println("</head>");
  client.println("<body style='background-color: #40E0D0;'>");
  client.println("<br/><br/>");
  client.println("<div style='text-align:center;'>");
  client.println("<h1>APAGAR 0 ENCENDER UN LED Y VERIFICAR UNA ENTRADA CON SHIELD</h1>");
  client.println("<br /><br />");
  client.println("<h2>Entradas Digitales</h2>");
  client.println("<div style='display: flex; justify-content: space-around;'>");
  client.println("<div style='border: 1px solid black; padding: 5px;'>IN 5</div>");
  client.println("</div>");
  client.println(digitalRead(5));
}

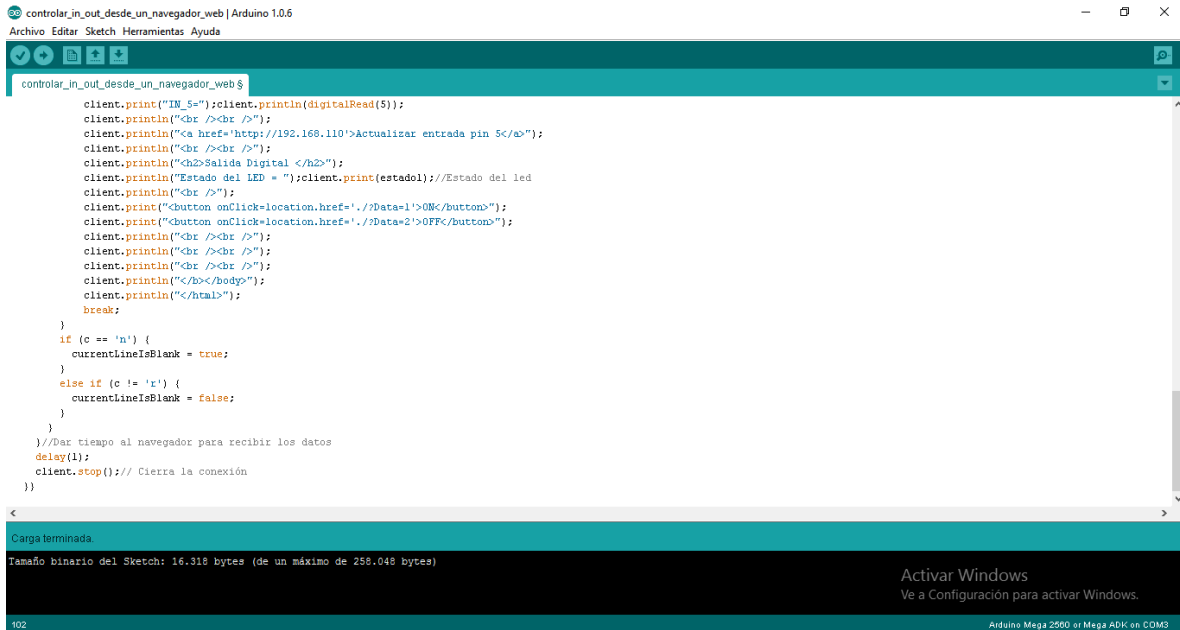

```

Carga terminada.

Tamaño binario del Sketch: 16.318 bytes (de un máximo de 258.048 bytes)

Activar Windows  
Ve a Configuración para activar Windows.

23 Arduino Mega 2560 or Mega ADK on COM3



### 7.3 Acceso a imagen remotamente a través de servidor web y Ethernet shield arduino.

Se crea el servidor web en la microsd, este servidor mostrará la imagen que también es guardada en la microsd de la persona que llegó al sitio. Una vez se abra la puerta la foto se elimina, esto porque siempre que se haga la solicitud al servidor, este va a buscar la fotografía con el nombre IMGE00.JPG , siempre que capture una foto se guardará en la microsd con este nombre. El acceso a la foto es mediante una petición GET.

```
C:\Users\jorge\Desktop\pagina web_foto arduino\index.htm - Notepad++
Archivo Editar Buscar Vista Codificación Lenguaje Configuración Macro Ejecutar Plugins Ventana ?
index.htm x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Fotografia en servidor web</title>
5 </head>
6 <body>
7 <h1>Persona que ha llegado a sitio remoto</h1>
8 
9 </body>
10 </html>
11
```

Hyper Text Markup Language file      length : 218    lines : 11      Ln : 11    Col : 1    Sel : 0 | 0      Unix (LF)    UTF-8    INS

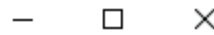


## Persona que ha llegado a sitio remoto





COM3



Enviar

```
Initializing SD card...
SUCCESS - SD card initialized.
SUCCESS - Found index.htm file.
GET / HTTP/1.1
Host: 192.168.1.110
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp
Accept-Encoding: gzip, deflate, sdch
Accept-Language: es-ES,es;q=0.8

GET /IMAGE00.jpg HTTP/1.1
Host: 192.168.1.110
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Accept: image/webp,image/*,*/*;q=0.8
Referer: http://192.168.1.110/
Accept-Encoding: gzip, deflate, sdch
Accept-Language: es-ES,es;q=0.8

GET /favicon.ico HTTP/1.1
Host: 192.168.1.110
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
Accept: image/webp,image/*,*/*;q=0.8
Referer: http://192.168.1.110/
Accept-Encoding: gzip, deflate, sdch
Accept-Language: es-ES,es;q=0.8
```

```
Servidor_que_muestra_imagende_SD | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda

Servidor_que_muestra_imagende_SD $
#include <SPI.h>
#include <Ethernet.h>
#include <SD.h>
// size of buffer used to capture HTTP requests
#define REQ_BUF_SZ 20
#define chipSelect 4
// MAC address from Ethernet shield sticker under board
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 0, 20); // IP address, may need to change depending on network
EthernetServer server(80); // create a server at port 80
File webFile;
char HTTP_req[REQ_BUF_SZ] = {0}; // buffered HTTP request stored as null terminated string
char req_index = 0; // index into HTTP_req buffer
void setup()
{
  // disable Ethernet chip
  pinMode(53, OUTPUT);
  digitalWrite(53, HIGH);
  Serial.begin(9600); // for debugging
  // initialize SD card
  Serial.println("Initializing SD card...");
  if (!SD.begin(4)) {
    Serial.println("ERROR - SD card initialization failed!");
    return; // init failed
  }
  Serial.println("SUCCESS - SD card initialized.");
  // check for index.htm file

```

```
Servidor_que_muestra_imagende_SD | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda

Servidor_que_muestra_imagende_SD $
// check for index.htm file
if (!SD.exists("index.htm")) {
  Serial.println("ERROR - Can't find index.htm file!");
  return; // can't find index file
}
Serial.println("SUCCESS - Found index.htm file.");
Ethernet.begin(mac, ip); // initialize Ethernet device
server.begin(); // start to listen for clients
}

void loop()
{
  EthernetClient client = server.available(); // try to get client
  if (client) { // got client?
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) { // client data available to read
        char c = client.read(); // read 1 byte (character) from client
        // buffer first part of HTTP request in HTTP_req array (string)
        // leave last element in array as 0 to null terminate string (REQ_BUF_SZ - 1)
        if (req_index < (REQ_BUF_SZ - 1)) {
          HTTP_req[req_index] = c; // save HTTP request character
          req_index++;
        }
        // print HTTP request character to serial monitor
        Serial.print(c);
        // last line of client request is blank and ends with \n

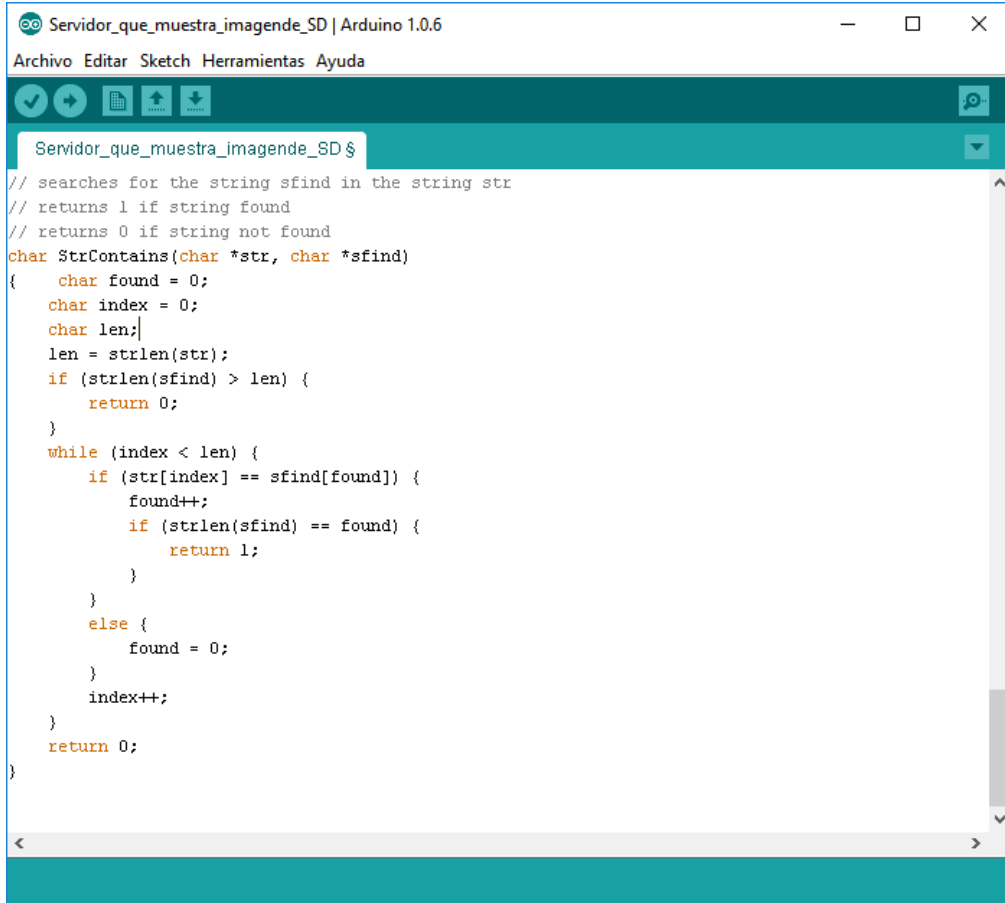
```

```
Servidor_que_muestra_imagende_SD | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda

Servidor_que_muestra_imagende_SD $
// last line of client request is blank and ends with \n
// respond to client only after last line received
if (c == '\n' && currentLineIsBlank) {
  // open requested web page file
  if (StrContains(HTTP_req, "GET /")
      || StrContains(HTTP_req, "GET /index.htm")) {
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("Connection: close");
    client.println();
    webFile = SD.open("index.htm"); // open web page file
  }
  else if (StrContains(HTTP_req, "GET /IMAGE00.jpg")) {
    webFile = SD.open("IMAGE00.jpg");
    if (webFile) {
      client.println("HTTP/1.1 200 OK");
      client.println();
    }
  }
  if (webFile) {
    while(webFile.available()) {
      client.write(webFile.read()); // send web page to client
    }
    webFile.close();
  }
  // reset buffer index and all buffer elements to 0
  req_index = 0;
}
```

```
Servidor_que_muestra_imagende_SD | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda

Servidor_que_muestra_imagende_SD $
req_index = 0;
StrClear(HTTP_req, REQ_BUF_SZ);
break;
}
// every line of text received from the client ends with \r\n
if (c == '\n') {
  // last character on line of received text
  // starting new line with next character read
  currentLineIsBlank = true;
}
else if (c != '\r') {
  // a text character was received from client
  currentLineIsBlank = false;
}
} // end if (client.available())
} // end while (client.connected())
delay(1); // give the web browser time to receive the data
client.stop(); // close the connection
} // end if (client)
}
// sets every element of str to 0 (clears array)
void StrClear(char *str, char length)
{
  for (int i = 0; i < length; i++) {
    str[i] = 0;
  }
}
}
```



```
Servidor_que_muestra_imagende_SD | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda
Servidor_que_muestra_imagende_SD $
// searches for the string sfind in the string str
// returns 1 if string found
// returns 0 if string not found
char StrContains(char *str, char *sfind)
{
  char found = 0;
  char index = 0;
  char len;
  len = strlen(str);
  if (strlen(sfind) > len) {
    return 0;
  }
  while (index < len) {
    if (str[index] == sfind[found]) {
      found++;
      if (strlen(sfind) == found) {
        return 1;
      }
    }
    else {
      found = 0;
    }
    index++;
  }
  return 0;
}
```

## 7.4 Envío de mensaje de texto a través de modem GPRS sim 900.

Si la presión del refrigerante en el aire acondicionado ha disminuido a 25 psi o menos, el presostato abrirá el contacto enviando una señal de LOW al arduino y este a su vez enviará un mensaje de texto SMS a un número celular alertando de la disminución de la presión del refrigerante en el aire acondicionado, el presostato cierra el contacto nuevamente cuando la presión llegue a 80 psi o más, esto sucederá luego de que el sistema de aire, las fugas haya sido reparado y llenado de refrigerante, pero esto no viene al caso y es otro tema que no abarca este proyecto.

```
enviar_sms_desde_arduino | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda

enviar_sms_desde_arduino
void setup()
{
  Serial.begin(9600);
  if (Serial.available())
  {
    Serial.println("pto serial ok");
  }
  delay(4000); // give time to log on to network.
}

void enviar_sms()
{
  Serial.println("AT+CMGF=1");//activamos el modo SMS
  delay(1500);
  Serial.print("AT+CMGS="); // comando AT para enviar SMS
  Serial.write( byte(34)); // enviar caracter "
  Serial.print("3138892905"); // numero de movil receptor del SM
  Serial.write( byte(34)); // enviar caracter "
  Serial.println();
  delay(1500);
  Serial.print("Ha llegado alguien"); //texto del SMS
  delay(1000);
  //END OF SMS
  Serial.write(0x1A); //Manda un Ctrl+z
  Serial.write(0x0D); //Manda un CR
  Serial.write(0x0A); //Manda un LF
}

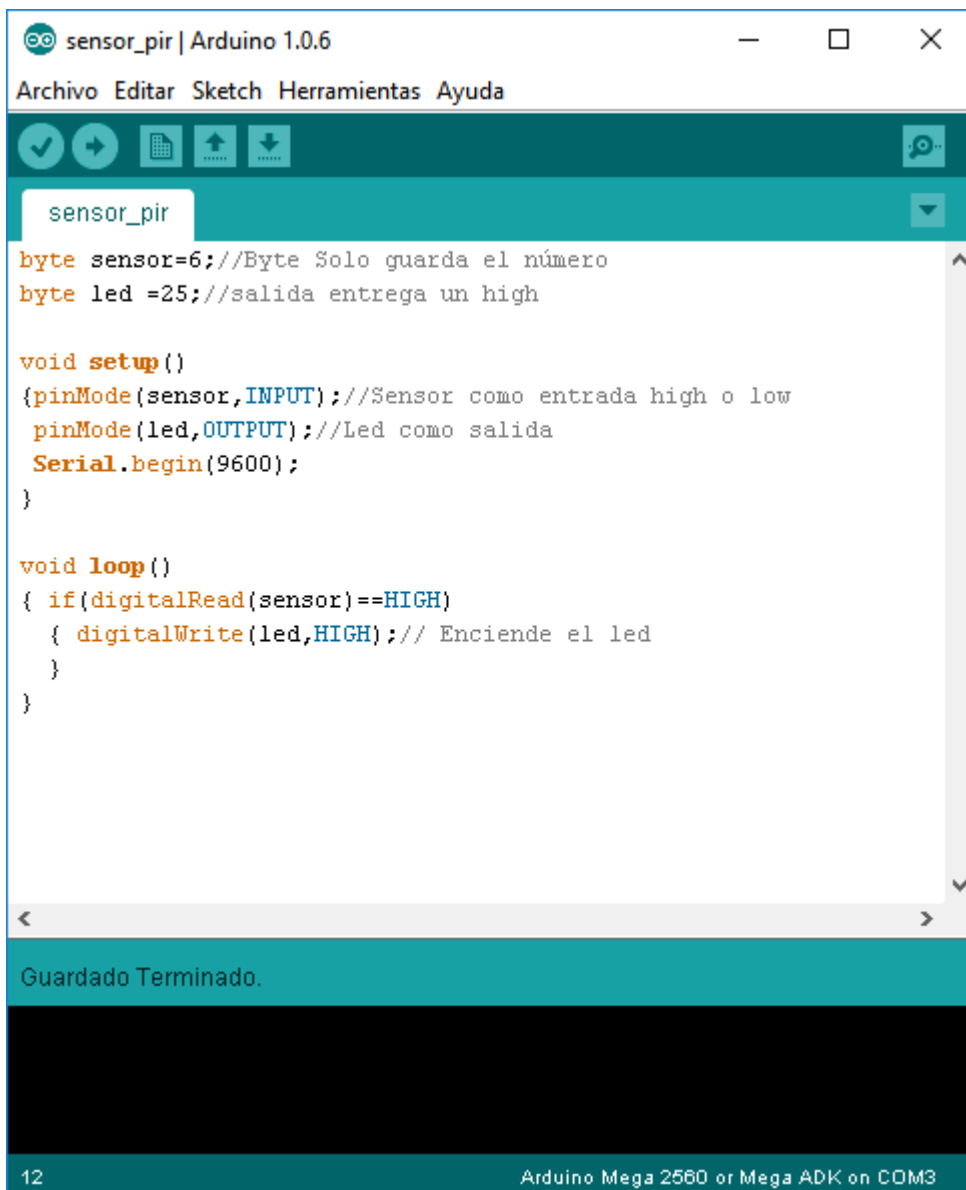
void loop ()
{
  enviar_sms (); // call someone
  do { } while (1); // do nothing
}

1 Arduino Mega 2560 or Mega ADK on COM3
```

## 7.5 Encender un led con sensor PIR

Esta parte del programa captura el movimiento en el sitio a través de un sensor PIR o infrarrojo, el cual entrega un HIGH en la salida al detectar movimiento. La señal emitida por el PIR hace que la cámara se active y se tome la foto de la persona. También se

puede enviar un mensaje de texto o realizar una llamada para avisar a quien sea necesario, pero para este caso solo se toma la foto.



```
sensor_pir | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda

byte sensor=6;//Byte Solo guarda el número
byte led =25;//salida entrega un high

void setup()
{pinMode(sensor,INPUT);//Sensor como entrada high o low
 pinMode(led,OUTPUT);//Led como salida
 Serial.begin(9600);
}

void loop()
{ if(digitalRead(sensor)==HIGH)
  { digitalWrite(led,HIGH);// Enciende el led
  }
}
```

Guardado Terminado.

12 Arduino Mega 2560 or Mega ADK on COM3

## 8. Aire acondicionado

Un aire acondicionado extrae el calor del sitio que se quiere acondicionar su temperatura, el refrigerante es una mezcla de hidrofluoruros carbonos que tiene como principal característica pasar de líquido a gas y viceversa. El refrigerante al cambiar de líquido a estado gaseoso absorbe calor, cuando vuelve a estado líquido libera el calor absorbido. En estado líquido su temperatura es de 2° centígrados (menor que la temperatura ambiente) por lo cual absorbe calor del sitio, después su temperatura aumenta y se convierte en gas, a este proceso se le llama ciclo de refrigeración mediante el cual suceden los siguientes procesos.

*Expansión* inicialmente el refrigerante está en estado líquido en el condensador a una alta presión, al pasarlo por una válvula de expansión se reduce la presión y la temperatura del líquido para continuar hacia la unidad interior.

*Evaporación* En la unidad interior o manejadora está el evaporador, el líquido se evapora convirtiéndose en gas, con lo cual libera el frío al sitio mediante un ventilador.

*Compresión* El refrigerante en estado gaseoso regresa a la unidad condensadora o exterior para cambiar a líquido, este gas se comprime en el compresor con lo cual aumenta su presión y temperatura a 65° centígrados y continúa hacia el condensador.

*Condensación* El gas al pasar por el condensador libera el calor al medio a través del serpentín, con lo cual se convierte en líquido e inicia nuevamente el ciclo.

El compresor realiza el trabajo contrario a la válvula de expansión, el compresor es el componente más importante y consume más energía, cuando en el sitio se consigue la temperatura preestablecida el compresor baja su ritmo de trabajo y se apaga si es necesario.

El aire acondicionado además de reducir la temperatura también reduce la humedad, estas son sus principales funciones, aunque también logra mantener el aire del sitio limpio al filtrarlo, se puede decir que mantiene el aire puro en el cuarto.

El refrigerante al contener hidrofluoruros puede ser contaminante para el medio ambiente como el refrigerante R22, el cual hoy en día se usa muy poco y se restringe su uso. El más utilizado en la actualidad es el R410A por ser poco contaminante y muy buen refrigerante. Este proyecto se desarrolla en base a las características de presión del refrigerante R410A.

## **PROPIEDADES FISICAS R410A**

### **Mezcla binaria R32 / R125**

Composición (%) 50 / 50  
Peso molecular (Kg/Kmol) 72.6  
Temperatura ebullición (°C) 52.7  
Deslizamiento temperatura (Glide) (°C) 0.5  
Temperatura crítica (°C) 72.5  
Presión crítica (bar) 49.5  
Densidad del líquido (25°C) (Kg/l) 1.07  
Densidad del líquido (-25°C) (Kg/l) 1.27  
Densidad del vapor (kg/m<sup>3</sup>) 4.2  
Presión del vapor (25°C) (bar) 16.5  
Presión del vapor (-25°C) (bar) 3.34  
Calor de vaporización (KJ/Kg) 257  
Capacidad calor del líquido (25°C) (KJ/Kg K) 1.71  
Capacidad calor del vapor (25°C) (KJ/Kg K) 1.28  
Conductibilidad térmica del líquido (25°C) (W/mK) 0.091  
Conductibilidad térmica del vapor (25°C) (W/mK) 0.013  
Solubilidad con el agua (25°C) ppm 2500  
Límite de inflamabilidad (% vol.) Ninguno  
Toxicidad (AEL) ppm 1000  
ODP – 0

### **Sensor de presión**

Un sensor de presión es un dispositivo que entrega en la salida un voltaje de acuerdo al valor o variación de presión para que este voltaje se pueda convertir mediante una fórmula a la unidad de presión. La variación del voltaje depende del cambio en la resistencia interna según la presión.

Existe un sensor 4-20 ma el cual se puede utilizar para medir en tiempo real la presión del refrigerante.

*Este sensor no se utilizó en el presente proyecto por su alto costo, además se debe encargar o comprar a través de internet porque no se consigue fácilmente en el mercado local. Se puede comprar en el exterior a través de internet y su envío puede tardarse un mes.*



Sensor De Presión 0-10 Bar (150psi) 4-20 ma Conexión 1/4npt.



### **Presostato**

El funcionamiento del presostato es similar al sensor de presión, solo que de acuerdo a la presión detectada se abre o cierra un contacto, según esto opera como un switch on-off. Para el presostato utilizado para este proyecto, el contacto se abre cuando la presión disminuye a 25 PSI o menos y se cierra cuando la presión es de 80 PSI o más, esto quiere decir que la presión inicial para que se cierre el contacto debe ser igual o mayor a 80 PSI.

PSI (Pounds per square Inch) libra por pulgada cuadrada es la unidad de presión en el sistema de medida inglesa, un PSI equivale a 0,0689476 Bares. De acuerdo a lo anterior, el presostato abre su contacto en 1,72 Bares o PSI y cierra el contacto en 5,51 bares u 80 PSI.



## 9. Desarrollo

Después de verificar el correcto funcionamiento de cada programa y componente de forma independiente, se procede a realizar el programa que abarca todo el proyecto, el cual se compone de varias pestañas o funciones que son llamadas en la pestaña principal, de acuerdo a las condiciones requeridas y secuencia del mismo. Estas funciones son los programas que se realizaron de forma independiente y son llamados en la pestaña principal "proyectogradoshieldEthernet".

```
ProjectogradoshieldEthernet | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda

ProjectogradoshieldEthernet$ abrir_puerta confirmar_sensores encender_modern enviar_sms sensor_movil sensor_prestion tomar_foto

#include <SPI.h> //Ethernet shield utiliza comunicacion SPI con arduino
#include <Ethernet.h>

#include <SD.h>
// size of buffer used to capture HTTP requests
#define REQ_BUF_SZ 20
#define chipSelect 4

int eliminarfoto=0;

int sensoromovimiento=0;
int sensorespresion=0;
char estadosensor=0;
// int foto=0;
|

// Declaración de la direcciones MAC,IP
byte mac[]={0xDE,0xAD,0xBE,0xEF,0xFE,0xED};
IPAddress ip(192,168,0,20); // 192.168.1.XX donde XX es una dirección que no esté utilizada
//IPAddress gateway(192, 168,1,1);
//IPAddress subnet(255,255,255,0);
// Creamos un servidor Web con el puerto 80 que es el puerto HTTP por defecto

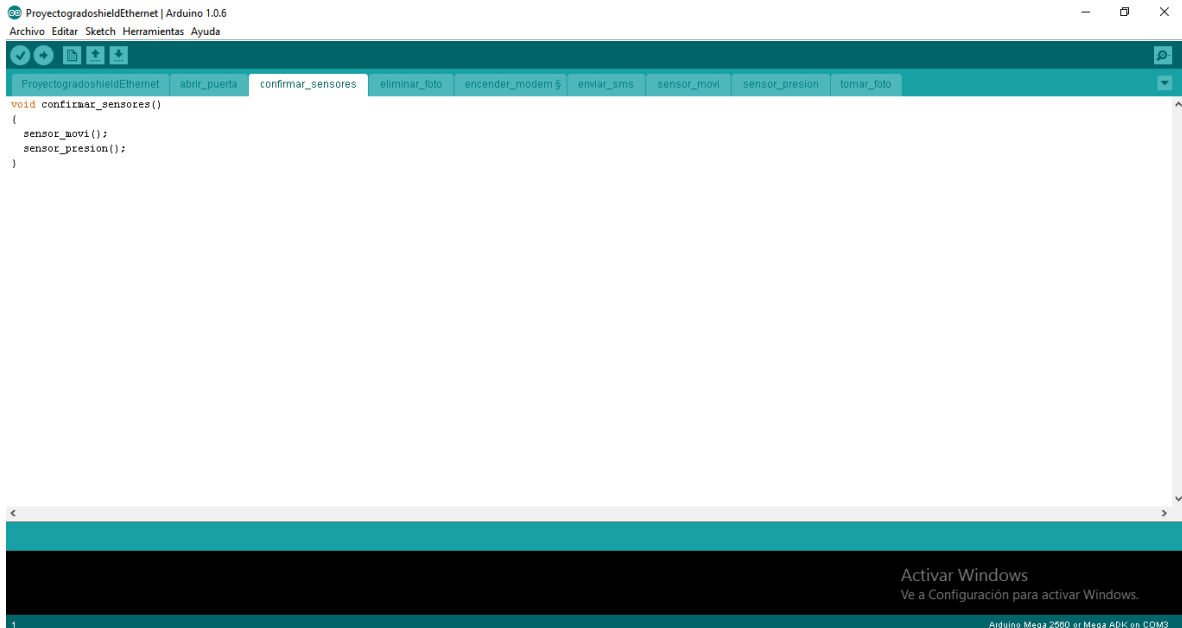
EthernetClient client;

EthernetServer server(2000); // SERVIDOR PARA ABRIR O CERRAR PUERTA
EthernetServer serverfoto(80); //SERVIDOR PARA VER IMAGEN DESDE SERVIDOR WEB

10 Arduino Mega 2560 or Mega ADK en COM3
```

Activar Windows  
Ve a Configuración para activar Windows.

Cada función es de tipo void y son llamadas desde el código principal cuando se requieran, por lo cual no se están ejecutando siempre en el programa. Esto ayuda a optimizar el recurso del hardware haciéndolo más rápido y eficaz para que no se ejecute instrucciones que no se requieren.



The screenshot shows the Arduino IDE interface with the 'confirmar\_sensores' function selected in the tab bar. The code is as follows:

```
void confirmar_sensores()
{
  sensor_movi();
  sensor_presion();
}
```



The screenshot shows the Arduino IDE interface with the 'eliminar\_foto' function selected in the tab bar. The code is as follows:

```
#include <SD.h>
File myFile;
void eliminar_foto()
{
  // 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("Initializing SD card eliminar foto");
  pinMode(53, OUTPUT);

  pinMode(10, OUTPUT);
  digitalWrite(10,HIGH); //PARA HABILITAR COMUNICACION CON LA SD A TRAVES DE LA SHIELD
  // COMUNICARSE CON LA SD POR EL PIN 4 (SS)
  if (!SD.begin(4)) { //SS PARA COMUNICACION CON SD
    Serial.println("initialization failed de eliminar foto!");
    return;
  }
  Serial.println("initialization done.");
  //myFile = SD.open("jorge.txt", FILE_WRITE);
  //myFile.close();

  // delete the file:
  if (SD.exists("IMAGE03.JPG")) {
    Serial.println("se borrara la IMAGEN.");
    delay(2000);
    SD.remove("IMAGE03.JPG");
  }
}
```

ProyectogradoshieldEthernet | Arduino 1.0.6  
Archivo Editar Sketch Herramientas Ayuda

ProyectogradoshieldEthernet | abrir\_puerta | confirmar\_sensores | eliminar\_foto | encender\_modem\$ | enviar\_sms | sensor\_mov | sensor\_presion | tomar\_foto

```
void encender_modem()
{
  digitalWrite(9,HIGH);
  delay(2000);
  digitalWrite(9,LOW);
  for (int i=0;i < 5;i++)//Tiempo para que el modem establezca conexion estable a la red
  {
    delay(4000);
  }
  Serial.println("ATE0");//Eco, los comandos introducidos vuelven al pc, elimina el eco del modulo
  delay(4000);
  Serial.print("Modem GPRS encendido");
}
```

Activar Windows  
Ve a Configuración para activar Windows.

Arduino Mega 2560 or Mega ADK on COM3

ProyectogradoshieldEthernet | Arduino 1.0.6  
Archivo Editar Sketch Herramientas Ayuda

ProyectogradoshieldEthernet | abrir\_puerta | confirmar\_sensores | eliminar\_foto | encender\_modem\$ | enviar\_sms | sensor\_mov | sensor\_presion | tomar\_foto

```
void enviar_sms()
{
  Serial.println("La presion ha bajado a menos de 25 psi");//Mensaje a enviar
  Serial.println("Se enviara mensaje de texto a: 3138892905");
  Serial.println("AT+CMGF=1");//activamos el modo SMS, 1 para texto
  delay(2000);
  Serial.print("AT+CMGS="); // comando AT para enviar SMS
  Serial.write(byte(34)); // enviar caracter
  Serial.print("3138892905");// numero de movil receptor del SMS
  Serial.write(byte(34)); // enviar caracter "
  Serial.println();
  delay(2000);
  Serial.print("Ha llegado alguien");//texto del SMS
  delay(1000);
  //END OF SMS
  Serial.write(0x1A); //Manda un Ctrl+z
  Serial.write(0x0D); //Manda un CR
  Serial.write(0x0A); //Manda un LF
}
```

Activar Windows  
Ve a Configuración para activar Windows.

Arduino Mega 2560 or Mega ADK on COM3

ProyectogradoshieldEthernet | Arduino 1.0.6  
Archivo Editar Sketch Herramientas Ayuda

```
void sensor_mov1()
{
  sensormovimiento = digitalRead(6);

  if (sensormovimiento == HIGH)
  {
    //Serial.print("Alguien ha llegado*****");
    //Serial.println(sensormovimiento);
    Serial.println();
    delay(1000);
    //toto=L;
  }
}
```

Activar Windows  
Ve a Configuración para activar Windows.

Arduino Mega 2560 or Mega ADK on COM3

ProyectogradoshieldEthernet | Arduino 1.0.6  
Archivo Editar Sketch Herramientas Ayuda

```
void sensor_presion()
{
  sensorpresion= digitalRead(5);

  if (sensorpresion==LOW)
  { Serial.print("La presion ha disminuido");
    estadosensor='L';
    Serial.println();

    // delay(1000);
  }
}
```

Activar Windows  
Ve a Configuración para activar Windows.

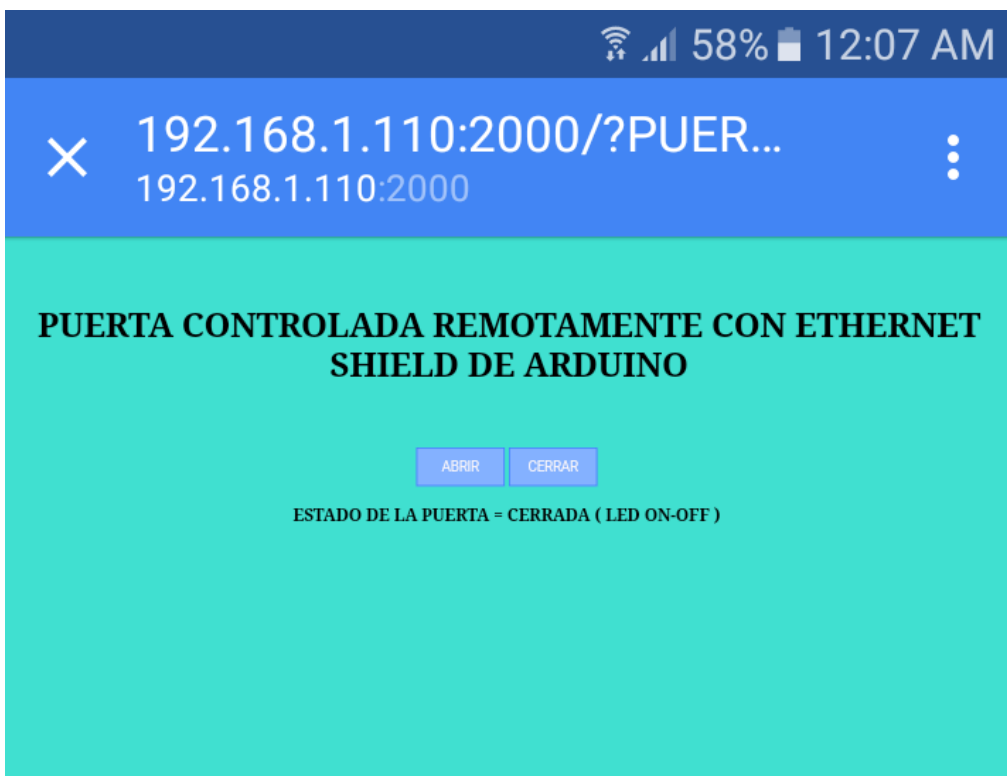
1

Arduino Mega 2560 or Mega ADK on COM3

Página web para abrir o cerrar la puerta, esta página se abre a través del puerto 2000, el cual fue asignado a gusto, se puede utilizar otro.



Ingrso desde el celular a través de la red wifi



Página web que muestra la foto del sitio y de la persona que llegó, para este caso se hace a través del puerto 80 que es por defecto el utilizado por las páginas web. Se utiliza la resolución más baja de la cámara y se puede observar que es una foto aceptable.



### Persona que ha llegado a sitio remoto



## 10. Conclusiones

Los resultados que se obtuvieron al finalizar el presente proyecto, hacen que el objetivo general y los objetivos específicos se hayan conseguido. Se logró implementar un sistema de control del nivel de presión de refrigerante y control de acceso a través del protocolo HTTP o página web en tiempo real. Adicional a esto se puede afirmar que con la consecución de este trabajo lo podríamos aplicar en diversos campos donde necesitemos realizar mediciones, telemetrías o control de variables en tiempo real a través de una página web.

Con la facilidad y disponibilidad de internet que hace que podamos conectarnos desde un sitio remoto en el mundo las 24 horas del día; este proyecto nos ayuda a que se puedan monitorear gran cantidad de cosas, aquí estaríamos entrando en el mundo del internet de las cosas IoT Internet of Things, el cual afirma que en el futuro a mediano plazo todo o casi todo estará monitoreado a través de la internet.

Se pudo comprobar también la capacidad de arduino de integrar más de una solución en control y transmisión, en este caso se utilizaron dos formas de transmitir a través de la shield Ethernet y el modem GPRS/GSM SIM 900, es una plataforma de control muy completa, hasta el punto de ser compatible con una cámara VGA.

## **11. Referencias**

<https://www.arduino.cc/>

<https://es.wikipedia.org/wiki/Arduino>

<https://www.arduino.cc/en/Main/arduinoBoardMega2560>

<https://www.arduino.cc/en/Tutorial/WebServer>

<http://hetpro-store.com/TUTORIALES/sim900-gsm-shieldarduino/>

<http://www.cetronic.es/sqlcommerce/disenos/plantilla1/seccion/producto/DetalleProducto.jsp?idIdioma=&idTienda=93&codProducto=999334029&cPath=1342>

<http://www.luisllamas.es/2015/07/detector-de-movimiento-con-arduino-y-sensor-pir/>

<https://github.com/adafruit/Adafruit-VC0706-Serial-Camera-Library>

<http://www.bloginstrumentacion.com/blog/2015/01/16/cmo-funciona-presostato/>

<http://www.electrobiomedical.com.co/download/datasheet/SEN0013.pdf>

[https://es.wikipedia.org/wiki/Acondicionamiento\\_de\\_aire](https://es.wikipedia.org/wiki/Acondicionamiento_de_aire)

<http://comofuncionaque.com/como-funciona-el-aire-acondicionado/>

<https://www.caloryfrio.com/aire-acondicionado/como-funciona-el-aire-acondicionado-infografia.html>



[http://www.construnario.com/ebooks/6730/fluidos%20frigor%C3%ADficos/gases%20refrigerantes%20\(hfcs\)/r410%20a/ficha%20t%C3%A9cnica.pdf](http://www.construnario.com/ebooks/6730/fluidos%20frigor%C3%ADficos/gases%20refrigerantes%20(hfcs)/r410%20a/ficha%20t%C3%A9cnica.pdf)

<https://startingelectronics.org/tutorials/arduino/ethernet-shield-web-server-tutorial/SD-card-web-server/>

<https://www.rinconingenieril.es/2-paginas-web-en-1-arduino/#>