

Trabajo de Diploma

Autor: Martín Ramón Ochoa Arboine

Tutor: Ing. Ángel Antonio Ravelo

Santiago de Cuba
2017



**UNIVERSIDAD
DE ORIENTE**

Facultad de Ingeniería Eléctrica

Departamento de Control Automático

Trabajo de Diploma

Título: Diseño e implementación de interfaz VGA/RCA
con Arduino para monitoreo domótico

Autor: Martín Ramón Ochoa Arboine

Tutor: Ing. Ángel Antonio Ravelo

Santiago de Cuba
2017

Resumen.

El presente Trabajo de Diploma trata el proceso de diseño y construcción de un dispositivo de monitoreo y visualización de variables para una instalación domótica en un televisor o monitor común, a partir de la placa Arduino. Se abordan generalidades sobre Arduino, su versatilidad, las ventajas que tiene y se explica el por qué se selecciona a Arduino Uno para la implementación. Se realiza una caracterización de los aspectos relacionados con los estándares de salida a TV por RCA y a monitor por VGA. Al finalizar se hace una valoración técnica de la instrumentación a emplear. El uso de la placa controladora Arduino supone una ventaja debido su bajo precio, versatilidad, fácil mantenimiento y compatibilidad. El algoritmo de programación será desarrollado en el lenguaje C, y seguirá en desarrollo para posteriores aplicaciones en otros sistemas de control que requieran monitorización de variables.

Abstract.

The present Project Work treats the design process and making of a device for monitoring and visualizing variables of an automated facility on a common TV or a common computer monitor using Arduino board. Here we boarded generalities about Arduino, its versatilities, its advantages and we explain why we choose Arduino Uno for implementation. We describe some aspects related to TV output standards like RCA and VGA output for computer monitors. At the end of this paper we present an economic evaluation for the instrumentation to employ.

Making use of Arduino controller board means an advantage for us, because it is cheap, versatile, easy maintenance and highly compatible. The code will be developed en C and will be continuously upgraded for later applications in other control systems that would require variable monitoring.

Contenidos.

INTRODUCCIÓN.	i
CAPITULO I.	- 1 -
1.1 Introducción.	- 1 -
1.2 Sistemas de Mediciones.	- 1 -
1.3 Sistema de Supervisión.	- 3 -
1.4 Domótica.	- 5 -
1.5 Métodos de supervisión para domótica.	- 6 -
1.6 Sensores y transductores.	- 8 -
1.7 Microcontroladores.	- 9 -
1.8 Arduino	- 12 -
1.8 Visualización por VGA y RCA.	- 14 -
Conclusiones parciales	- 17 -
CAPITULO II.	- 18 -
1.1 Introducción.	- 18 -
1.2 Descripción de la planta.	- 18 -
1.3 Sensores.	- 19 -
1.4 Arduino y sensores.	- 24 -
1.5 Comunicación Serie.	- 25 -
1.6 Recepción de comandos por infrarrojos	- 26 -
1.7 Archivo de registro en tarjeta SD	- 29 -
1.8 Protocolo de visualización VGA/RCA.	- 30 -
1.9 Valoración económica	- 34 -
Conclusiones generales	- 36 -
Bibliografía.	- 38 -
Anexos	- 40 -

INTRODUCCIÓN.

En la actualidad los procesos de monitoreo y supervisión en el campo de la domótica se basan en el empleo de métodos que se hacen cada vez más complejos y costosos. Estos métodos incluyen el uso de pantallas con tecnología LCD/LED, los ordenadores (PC), Smartphone, televisores inteligentes (Smart TV) u otros equipos de visualización que por lo general resultan caros para los usuarios de viviendas automatizadas. La mayoría de esos dispositivos no son fáciles de obtener en el mercado internacional por ser muy caros debido a la tecnología que emplean o por ser difíciles de instalar. Sin embargo es posible obtener funcionalidades similares a un menor precio con el uso de placas Arduino.

Arduino es una plataforma electrónica abierta para la creación de prototipos y aplicaciones basadas microcontroladores, donde tanto el software como el hardware son libres, flexibles y fáciles de usar. Está orientada a personas con perfiles creativos, sin experiencia previa en temas de hardware y software, interesadas en desarrollar proyectos electrónicos de forma fácil e intuitiva (Santa Emilia, 2015).

Una vez que el usuario es poseedor de una vivienda automatizada, utilizar los medios de visualización con que dispone es una ventaja a grandes rasgos. Es decir, partiendo de la idea de que en la actualidad prácticamente cualquier persona puede tener un televisor o quizá un monitor, lo que este proyecto propone es que los usuarios de viviendas automatizadas no tengan que comprar una nueva pantalla o implementos extra para visualizar las lecturas de sus sensores. Unido a ello se propone el empleo de la plataforma Arduino cuyo precio es ínfimo en los mercados alternativos, obteniendo como resultado una oferta muy económica competente con las alternativas convencionales.

Se define entonces como **problema de investigación**: la inexistencia de un sistema de monitoreo de bajo costo para el monitoreo de viviendas automatizadas.

Objeto de la investigación: los sistemas de supervisión y monitoreo.

Objetivo: Diseñar y construir una interfaz con Arduino que visualice en la pantalla de un televisor o de monitor común los valores de los sensores de una vivienda automatizada, registrando en un archivo (log) los datos históricos de estas lecturas.

Campo de acción: los sistemas de monitoreo para viviendas automatizadas.

Hipótesis: si se implementa una interfaz de monitoreo con Arduino, capaz de visualizar en pantalla y registrar las lecturas de los sensores de una vivienda automatizada, se contaría un sistema de monitoreo barato y rentable comparado con las alternativas convencionales.

Tareas de la investigación:

1. Caracterizar gnoseológicamente e históricamente los sistemas de monitoreo y supervisión.
2. Describir la planta que se monitorea.
3. Caracterizar a los microcontroladores y a la placa Arduino.
4. Programar el sistema de registro de lecturas de los sensores.
5. Programar la recepción de comandos por infrarrojos.
6. Programar la visualización y gestión de pantallas por VGA y RCA.
7. Analizar el comportamiento del prototipo en la maqueta

Organización del documento:

En el Capítulo I se describen conceptos necesarios relacionados con la actividad de supervisión de sistemas de control y métodos de llevarla a cabo, la domótica, sensores, microcontroladores y por último la visualización por VGA y RCA.

En el Capítulo II se realiza la descripción del proceso a monitorear o planta, los sensores que incorpora y como estos convierten las magnitudes en variables que se pueden monitorear. Más adelante se explica la comunicación que se establece entre las placas Arduino y el diseño e implementación del protocolo VGA y RCA que visualiza los datos leídos en pantalla.

CAPITULO I.

1.1 Introducción.

Este Capítulo hace mención de conceptos que son de interés en esta investigación, yendo desde lo general a lo particular: se describen conceptos relacionados con la actividad de supervisión de sistemas de control y métodos de llevarla a cabo, la domótica, sensores, microcontroladores y por último la visualización por VGA y RCA.

1.2 Sistemas de Mediciones.

La adquisición de datos (DAQ) es el proceso de medir con una PC un fenómeno eléctrico o físico como voltaje, corriente, temperatura, presión o sonido. Los sistemas de mediciones o sistemas de adquisición y procesamiento de datos surgieron en los años 60 para manejar grandes volúmenes de información. En aquella época era necesario crear un medio de medición que permitiera hacer todo lo que hacían los instrumentos digitales, pero que además tuviera la capacidad de almacenar y procesar la información, permitiendo que el operador pudiera leerla en una pantalla.

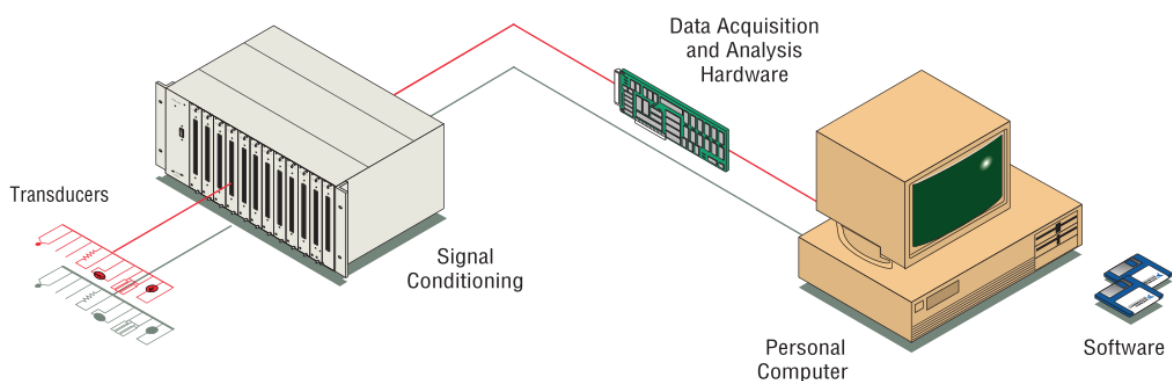


Figura 1 Sistema de mediciones genérico

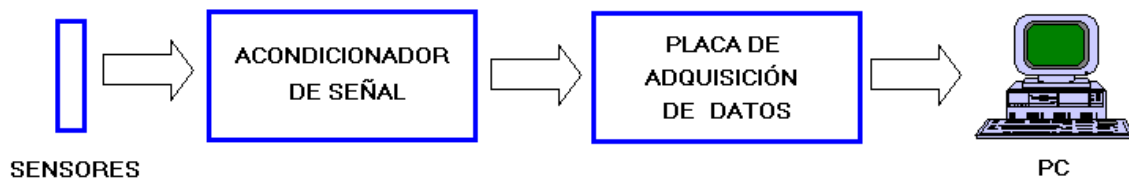
Imagen tomada de Data Acquisition (DAQ) Fundamentals – National Instruments

Básicamente, un sistema DAQ consta de sensores, hardware de medidas DAQ y una PC con software programable. Estos aprovechan la potencia de procesamiento, la productividad, la visualización y las habilidades de conectividad

de los ordenadores estándares en la industria proporcionando una solución de medidas más potente, flexible y rentable.

En estos sistemas la información viene en forma de señales analógicas y pasa por tres etapas antes de llegar al dispositivo visualizador:

1. Conversión de la magnitud medida a una señal eléctrica (sensor/transductor)
2. Adaptación de la señal eléctrica para su lectura digital (acondicionador de señal)
3. Hardware de adquisición de datos (DAQ) y visualización (generalmente un computador)



Esquema de Sistema de Adquisición de Datos

Figura 2 Proceso de Adquisición de datos

Imagen tomada de <http://sine.ni.com/>

En el último paso del proceso, la señal (ya acondicionada) se digitaliza a través de los convertidores analógico-digital (ADC). El hardware de adquisición de datos tiene como características:

- Número de canales
- Velocidad de muestreo
- Resolución
- Rango de entrada
- Comunicación con el computador

-Número de canales analógicos: Indica la cantidad de magnitudes distintas que se pueden adquirir en una misma unidad. Generalmente para lograr hacer varias mediciones se emplea un único convertidor analógico digital (ADC) y los diferentes canales se generan por medio de un multiplexor analógico.

-Velocidad de muestreo: Cuanto mayor sea esta velocidad, mejor será la representación de la señal analógica. Según el teorema de Nyquist la velocidad

de muestreo debe ser mayor que el doble de la frecuencia de la señal a muestrear.

-Resolución: está dada por el número de bits del ADC que se utilizan para representar cada muestra de la señal. Mientras mayor sea este número, mejor será la detección de cambios, más precisión en la lectura.

-Rango de entrada: Indica los márgenes entre los que debe estar la señal de entrada para que pueda ser convertida. Las tarjetas DAQ suelen dar varias posibilidades que se pueden seleccionar por hardware o por software.

Los sistemas de DAQ generalmente encuentran su aplicación en las grandes industrias, donde se tiene una enorme cantidad de sensores distribuidos por toda la instalación y para organizar todo el proceso se ubican dentro de un centro de mando. Pero también pueden encontrarse en casas, edificios, automóviles, aviones e incluso en la ropa (wearable y ropa inteligente) (Jones & Chin, 1983).

1.3 Sistema de Supervisión.

La palabra supervisión se ha empleado en varias ocasiones en este trabajo, para una mayor comprensión de la misma definiremos que se entiende por supervisión. Las exigencias que actualmente se imponen a los procesos productivos en cuestión del rendimiento, calidad y flexibilidad hacen necesario introducir las nuevas tecnologías en el control y vigilancia de éstos. Con este propósito nace la idea de supervisar los procesos. Esto permite la reducción del número de paradas innecesarias, la predicción de situaciones anómalas o la actuación rápida y eficaz de forma que asegure la continuidad y uniformidad de la producción (Colomer, Meléndez, & Ayza, 2010).

Concepto:

Un sistema de supervisión es a grandes rasgos una (o varias) aplicación(es) informática(s) que sacará(n) partido de toda información relativa al proceso para decidir sobre su correcto (o no) funcionamiento y proponiendo, a su vez, las acciones adecuadas para mantener los objetivos productivos fijados para el proceso. (Brennan, 2007)

Es decir, que los sistemas de supervisión se apoyan de los sistemas de mediciones, permitiendo adquirir datos y almacenarlos. Su principal objetivo es ofrecer estos datos al operario de manera amigable a través de una interfaz hombre maquina (HMI). Pero además deben ser capaces gestionar situaciones de alarma, pudiendo actuar por su cuenta u ofreciéndole al operario la posibilidad de decidir qué hacer ante tales situaciones. (Colomer, Meléndez, & Ayza, 2010).

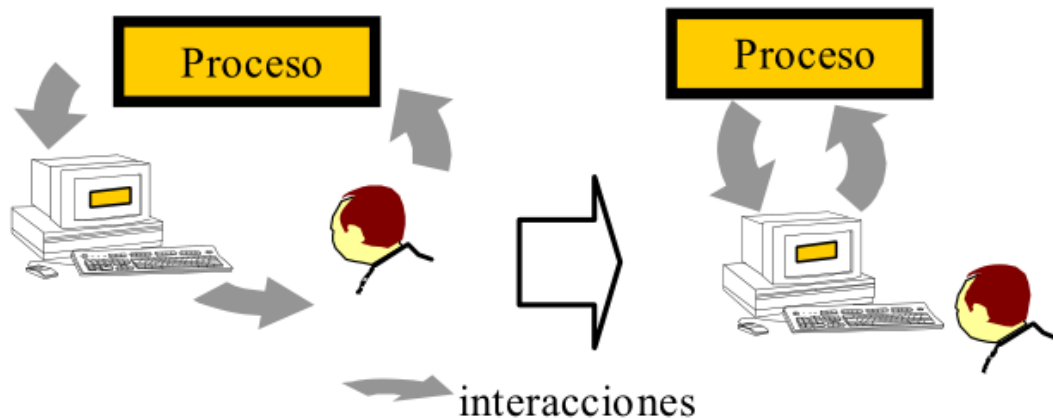


Figura 3 Monitorización y supervisión

Imagen tomada de Sistemas de Supervision.pdf

Para implementar la interfaz amigable se emplean los HMI que son generalmente aplicaciones informáticas, aunque se pueden venir en forma de dispositivos electrónicos.

Los sistemas de supervisión se pueden clasificar de acuerdo al modo en que el usuario accede a su interfaz como Interfaces locales o interfaces remotas, cuando el usuario accede a la información dentro del campo donde se ubican los sensores o fuera de este, respectivamente.

Existen numerosos tipos de sistemas supervisores aunque los más conocidos son los SCADA (Supervisory Control And Data Acquisition). También existen otros software orientados a la monitorización y control de procesos como son LabWindows/CVI, EROS, LabView, KNX (UNITEL, 2017). Desde el punto de vista de hardware de monitoreo se puede mencionar los sistemas de luces usados en los paneles industriales, paneles de visualización neumáticos de las viejas

fábricas y los más reciente que es la ropa inteligente, y dispositivos eléctricos de monitoreo en aviones, naves espaciales, autos y algunas casas.

1.4 Domótica.

La domótica automatiza las tareas repetitivas del hogar y su concepto está estrechamente vinculado con los sistemas de medición y de supervisión.

Se denomina domótica al conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar. Se podría definir como la integración de la tecnología en el diseño inteligente de un recinto cerrado. La domótica integra el control y supervisión de los elementos existentes en un edificio de oficinas, de viviendas o simplemente en cualquier hogar. (A. Datum Corporation, 2017).

Las instalaciones modernas requieren nuevos conceptos y exigencias que pueden ser caracterizadas por servicios como el confort, la seguridad, el ahorro de energía, las telecomunicaciones y los niveles mínimos de equipamiento, etc.

Los tres componentes fundamentales de una vivienda domótica son: sensores, controladores y actuadores. La secuencia básica de funcionamiento sería la siguiente: los sensores captan la información, datos de las magnitudes del hogar que son enviadas a los controladores, éstos en función de los valores de esas magnitudes deciden si es necesaria o no una actuación; en caso positivo, se envía la orden a los actuadores que activan el funcionamiento del dispositivo correspondiente de la vivienda.

En la aplicación de un sistema domótico existen necesidades que el usuario final debe tener en cuenta en el momento de la instalación. Los servicios o aplicaciones pueden abarcar también áreas de entretenimiento y ayuda a discapacitados. En este último campo se hacen indispensables los sistemas de seguridad y monitoreo, así como el confort en los sistemas de control para llevar a

cabo las órdenes por medio de interfaces humanas inalámbricas y evitar el desplazamiento dentro de un hogar (Hena Merchán, 2006).

A medida que aumenta el número de elementos automatizados de un sistema domótico se hace necesario organizar la forma en que se interconectan unos con otros y para ello se crean redes, como las utilizadas en los sistemas de medición industriales, solo que en este caso aparte de enviar información de los sensores y actuadores, también se transmite información de los electrodomésticos, alarmas y hasta información proveniente desde otras redes cercanas o Internet. Por lo cual el concepto de sistemas sistema de mediciones y sistemas de adquisición de datos cobra otro enfoque aún más amplio y más complejo.

1.5 Métodos de supervisión para domótica.

Existen varios métodos para supervisar el control de una vivienda domótica. Cada uno ofrece una determinada ventaja. El uso principal de estos sistemas está orientado al mantenimiento de la seguridad, el ahorro de energía y el confort.

Los métodos de supervisión domótica se pueden dividir en dos grupos según la ubicación de la interfaz desde la cual se consulta la información:

- Interfaces locales: hace referencia a las interfaces de consulta de la información que se encuentran dentro de la misma zona que el (los) instrumento(s) que supervisa.
- Interfaces remotas: lo contrario de lo anterior.

Las interfaces locales son muy fáciles de identificar. Por ejemplo el sistema de alarma siguiente



Figura 4 Interfaz de supervisión local ubicada en el Edificio Administrativo de la Universidad de Oriente en Santiago de Cuba

Sin embargo las interfaces de supervisión remota pueden ser muy disímiles. Por ejemplo:

- Interfaces de supervisión web: son aplicaciones centralizadas por lo general en un servidor en Internet, al cual el usuario accede desde cualquier parte del mundo y consultar las lecturas de los sensores en tiempo real. Por ejemplo las aplicaciones que emplean DNA de Microsoft (Distributed InterNet Application).
- Interfaces de supervisión móviles: se basan en las anteriores, permite al usuario consultar información desde un dispositivo inalámbrico como un Tablet o un Smartphone. La diferencia principal es que estas están optimizadas para funcionar con este tipo de tecnología. Este tipo de interfaz tiene un área límite de funcionamiento y se podría decir que es una mezcla entre local y remoto.



Figura 5 Ejemplos de interfaces móviles de monitoreo y control

Imágenes tomadas de <http://www.jpm.pt/>

En general los sistemas de supervisión para domótica tienen como objetivo presentar la información que obtienen de los sensores de manera dinámica, intuitiva y amigable al usuario, apoyándose de las interfaces gráficas (GUI).

1.6 Sensores y transductores.

Antes se mencionó que los sistemas de monitoreo supervisan determinados procesos y permiten al usuario tomar decisiones acerca de su desempeño. Y ya que su misión es abstraer la información del medio, necesitan tomar toda esa información de alguna manera.

Los sensores censan los fenómenos físicos y proveen señales eléctricas que los sistemas de adquisición de datos pueden medir. Por ejemplo los termopares, RTD, termistores, y sensores en circuitos integrados convierten la temperatura en una señal analógica que luego puede ser medida por un convertidor analógico digital (DAC). Otros ejemplos incluyen galgas de tensión, transductores de flujo y transductores de presión, los cuales miden fuerza, frecuencia del flujo, y presión, respectivamente. En cada caso las señales eléctricas producidas son proporcionales a los parámetros físicos que se monitorean. Las señales eléctricas generadas por los sensores deben ser optimizadas para el rango de entrada de los sistemas de adquisición. (National Instruments, 1999)

Un sensor se diferencia de un transductor en que el sensor está siempre en contacto con la variable de instrumentación. En la actualidad la diferencia entre estos términos es transparente ya que los sensores se producen con circuitos para el acondicionamiento y transmisión de la señal.

Parámetros de un sensor

- Rango de medida: dominio en la magnitud medida en el que puede aplicarse el sensor.
- Precisión: es el error de medida máximo esperado.
- Offset o desviación de cero: valor de la variable de salida cuando la variable de entrada es nula. Si el rango de medida no llega a valores nulos de la variable de entrada, habitualmente se establece otro punto de referencia para definir el offset.

- Linealidad o correlación lineal.
- Sensibilidad de un sensor: suponiendo que es de entrada a salida y la variación de la magnitud de entrada.
- Resolución: mínima variación de la magnitud de entrada que puede detectarse a la salida.
- Rapidez de respuesta: puede ser un tiempo fijo o depender de cuánto varíe la magnitud a medir. Depende de la capacidad del sistema para seguir las variaciones de la magnitud de entrada.
- Derivas: son otras magnitudes, aparte de la medida como magnitud de entrada, que influyen en la variable de salida. Por ejemplo, pueden ser condiciones ambientales, como la humedad, la temperatura u otras como el envejecimiento (oxidación, desgaste, etc.) del sensor.
- Repetitividad: error esperado al repetir varias veces la misma medida.

(García, Antón, Rodríguez, Martín, & Ortega, 2004)

1.7 Microcontroladores.

Un microcontrolador (abreviado μC , UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.



Figura 6 El microcontrolador 8051 de Intel

Imagen tomada de <http://smithsonianchips.si.edu/>

Existen muchos tipos de microcontroladores en el mundo fabricados por distintas firmas:

- Analog Devices
- Atmel
- Dallas Semiconductor
- Fujitsu
- Intel (MCS8051)
- National Semiconductor
- NEC (Renesas)
- Parallax
- MicroChip (microcontroladores PIC)
- Motorola

Existen muchísimas firmas y cada una provee microcontroladores con distintas características. Uno de los más populares son los la familia Atmel, que junto con el proyecto de hardware libre Arduino produce las famosas placas.

No se puede dejar de mencionar en este ámbito a los controladores de la Fundación Raspberry PI, los cuales son muy potentes y de reducido tamaño y hoy por hoy han ganado tremendo auge en las comunidades de desarrolladores, sin embargo por ahora representan un solución muy cara para los objetivos de este proyecto, además de que no contamos con posibilidad de adquirirlas.

Características generales de los microcontroladores

Los microcontroladores están diseñados para reducir el costo económico y el consumo de energía de un sistema en particular. Por eso el tamaño de la unidad central de procesamiento, la cantidad de memoria y los periféricos incluidos dependerán de la aplicación.

En este caso la aplicación que se le dará al microcontrolador será como hardware de adquisición de datos.

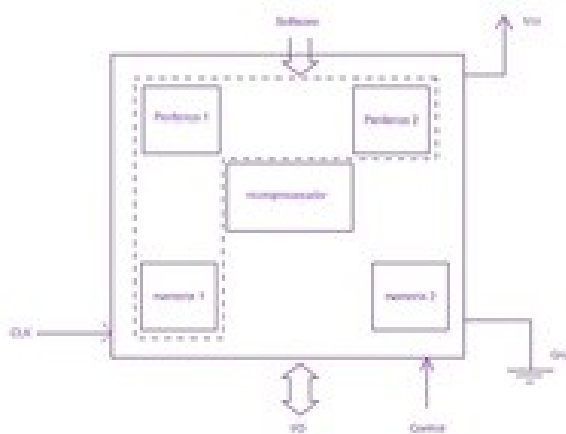


Figura 7 Esquema general de un microcontrolador

Imagen tomada de <http://smithsonianchips.si.edu/augarten/p38.htm>

Un microcontrolador típico tendrá un generador de reloj integrado y una pequeña cantidad de memoria de acceso aleatorio y/o ROM/EPROM/EEPROM/flash, con lo que para hacerlo funcionar todo lo que se necesita son unos pocos programas de control y un cristal de sincronización. Los microcontroladores disponen generalmente también de una gran variedad de dispositivos de entrada/salida, como convertidor analógico digital, temporizadores, UART y buses de interfaz serie especializados, como I2C y CAN.

Los modernos microcontroladores frecuentemente incluyen un lenguaje de programación integrado, como el lenguaje de programación BASIC que se utiliza bastante con este propósito.

La memoria en los microcontroladores debe estar ubicada dentro del mismo encapsulado, esto es así la mayoría de las veces, porque la idea fundamental es mantener el grueso de los circuitos del sistema dentro de un solo integrado.

En los microcontroladores la memoria no es abundante, aquí no encontrará Gigabytes de memoria como en las computadoras personales. Típicamente la memoria de programas no excederá de 16 K-localizaciones de memoria no volátil (flash o EPROM) para contener los programas.

La memoria RAM está destinada al almacenamiento de información temporal que será utilizada por el procesador para realizar cálculos u otro tipo de operaciones lógicas. En la memoria RAM se almacenan también los registros de trabajo y configuración del procesador y los distintos periféricos del microcontrolador.

El tipo de memoria utilizada en las memorias RAM de los microcontroladores es SRAM.

EEPROM (Electrical Erasable Programmable Read Only Memory). Fueron el sustituto natural de las memorias EPROM, la diferencia fundamental es que pueden ser borradas eléctricamente, por lo que la ventanilla de cristal de cuarzo y los encapsulados cerámicos no son necesarios.

Memoria flash. En el campo de las memorias reprogramables para microcontroladores, son el último avance tecnológico en uso a gran escala, y han sustituido a los microcontroladores con memoria EEPROM (Gridling & Weiss, 2007).

1.8 Arduino

Como se mencionó antes, para este proyecto es necesario un microcontrolador que actúe como hardware de adquisición de datos. Existen otros microcontroladores y plataformas con microcontroladores disponibles para el desarrollo de proyectos de ingeniería: Parallax Basic Stamp (Parallax Inc., 2017), BX-24 de Netmedia, Phidgets, Handyboard del MIT (Massachusetts Institute of Technology), PIC (Embedded Related, 2017) y muchos otros ofrecen funcionalidades similares.

Todas estas herramientas organizan el complicado trabajo de programar un microcontrolador en paquetes fáciles de usar.

Arduino, además de simplificar el proceso de trabajar con microcontroladores, ofrece algunas ventajas respecto a otros sistemas similares, entre estas tenemos:

- Económicas: Las placas de Arduino son más baratas comparadas con otras plataformas de microcontroladores. La versión más cara de un módulo de Arduino cuesta menos de 60€.
- Multi-Plataforma: El software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux. La mayoría de los demás entornos para microcontroladores están limitados a Windows.
- Entorno de programación simple y directo: El entorno de Arduino es fácil de usar para principiantes y flexible para los usuarios avanzados. Arduino está basado en el entorno de programación de Processing con lo que el

estudiante que use este entorno se sentirá familiarizado con el entorno de desarrollo Arduino.

- Software ampliable y de código abierto: El software Arduino está publicado bajo una licencia libre y puede ser ampliado por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++, y si se está interesado en profundizar en los detalles técnicos, se puede dar el salto a la programación en el lenguaje AVR C en el que está basado.
- Hardware ampliable y de Código abierto: Arduino está basado en los microcontroladores ATmega168, ATmega328y ATmega1280. Los planos de los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores de circuitos con experiencia pueden hacer su propia versión del módulo, ampliándolo u optimizándolo. Incluso usuarios relativamente inexpertos pueden construir la versión para placa de desarrollo para entender cómo funciona y ahorrar algo de dinero.

Para realizar este proyecto solamente se contaba con placas Arduino Uno. Por todas estas razones y enfatizando aún más en las razones económicas es que se escogió como hardware de trabajo.

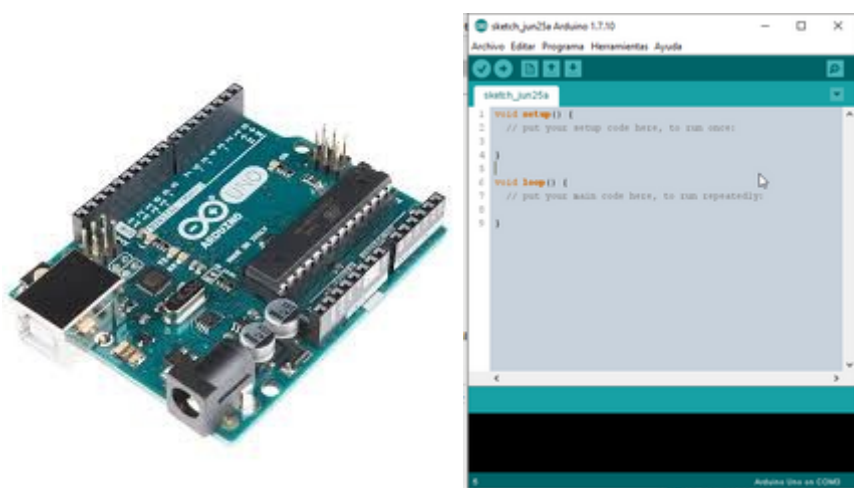


Figura 8 Arduino Uno y Arduino IDE

Imagen tomada de <http://arduino.cc>

En la tabla siguiente se citan algunas características de la placa Arduino escogida.

Familia del microcontrolador	ATmega328
------------------------------	-----------

Voltaje operativo	5V
Voltaje de entrada recomendado	7-12V
Voltaje de entrada limite	6-12V
Pines de E/S digitales	14 (6 salidas PWM)
Pines de entrada analógica	6
Corriente máxima para pines E/S	40mA
Corriente máxima para pines de 3.3V	50mA
Memoria Flash	32KB (Atmega328), de los cuales 0.5 KB son para el <i>bootloader</i> .
SRAM	2KB
EEPROM	1KB
Frecuencia del cristal	16MHz

La principal ventaja de escoger Arduino Uno es por su reducido precio y su popularidad. Al ser el producto más icónico de la compañía, sus unidades se producen más haciéndolo más barato que sus compañeros. Formalmente las unidades son producidas en Italia, pero existen muchos derivados de este, desarrollados por terceras compañías (los llamados clones) que son aún más baratos y por tanto representan una alternativa muy viable.



Figura 9 Elegoo UNO R3 es un clon de Arduino, una alternativa a precio reducido.

1.8 Visualización por VGA y RCA.

Hoy en día casi todos los televisores y/o monitores incluyen al menos uno de los conectores VGA o RCA. Estos son los conectores para transmisión de señal de video más populares (ahora están siendo remplazados por HDMI). Todos los

televisores y los monitores poseen al menos uno de estas entradas que permiten transmitir video como señales analógicas.

VGA

Es un estándar para la transmisión de video que es muy común en los monitores de las computadoras de escritorio. Transmite en canales separados información de color y sincronismos necesarios para formar una imagen en pantalla. El conector VGA está formado por 15 pines de conexión, pero de estos solo 5 señales son las importantes: HSYNC, VSYNC, R, G y B, los demás pines se conectan a tierra.

Cuando se habla de VGA no se puede dejar de mencionar la resolución que esta implica: 640x480 pixeles.

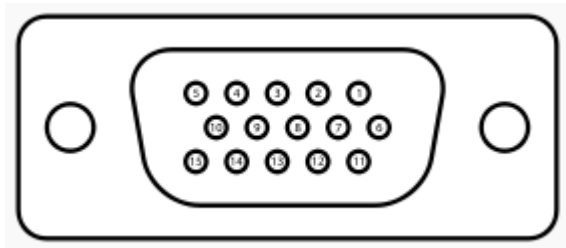


Figura 10 Conector VGA de 15 pines

Algunos datos técnicos:

- 256 KiB de VRAM
- Reloj maestro seleccionable de 25,2 MHz o 28,3
- Máximo de 800 píxeles horizontales
- Máximo de 600 líneas
- 0,7 V pico a pico
- 75 ohmios de impedancia de doble terminación (18,7 mA - 13 mW)

Un conector VGA como se le conoce comúnmente (otros nombres pueden ser conector RGBHV, D-sub 15, sub mini mini D15 y D15), consta de tres hileras de 15 pines DE-15. Con la revolución digital, a partir de 2009 se comienza a reemplazar estos conectores VGA por conectores HDMI que debido a sus características avanzadas en tarjetas gráficas, pantallas y monitores actuales (Area Urbana.net, 2017).

RCA

Es un estándar de transmisión de video que es más común en los televisores. La señal de video se transmite en un mismo canal, es decir que por un mismo cable pasan las señales de sincronismo y la información de color, por eso se le conoce como video compuesto. El conector RCA solo tiene dos canales, el de referencia a tierra (-) y el canal de la señal de video (+).

El nombre "RCA" deriva de la Radio Corporation of America, que introdujo el diseño en los 1940.



Figura 11 Conector RCA

Los colores usados suelen ser:

- Amarillo para el vídeo compuesto
- Rojo para el canal de sonido derecho
- Blanco o negro para el canal de sonido izquierdo (en sistemas estéreo)

Video por Componentes Analógico (YPbPr)

Es un estándar aparentemente muy parecido a RCA, pero se diferencia en que la información de video analógica se transmite mediante 2 o más señales. Es usado en la mayoría de los televisores porque ofrece mayor calidad que el antiguo video compuesto.

Las señales de video analógico (también llamadas componentes) proporcionan información acerca de la cantidad de rojo, verde y azul para crear la imagen en el televisor.



Figura 12 Conectores del video por componentes analógico

Imagen tomada de <http://areaurbana.net>

Las componentes son:

Y – componente de luminancia

Pb – componente de color azul

Pr – componente de color rojo

Sin embargo la señal de video compuesto (RCA) es compatible con la componente de luminancia Y, es decir que aún se puede ver la imagen conectando el conector amarillo (RCA) con el conector verde (YPbPr).

Actualmente, la conexión de video por componentes está siendo reemplazada por HDMI (Area Urbana.net, 2017).

Conclusiones parciales

En este capítulo se ha explicado la variedad de sistemas de supervisión para instalaciones domóticas y como ellos están relacionados con los sistemas de mediciones, a partir de esto se concluye que la interfaz que se desea implementar debe ser capaz de mostrar la información y también de almacenarla. Para ello debe cumplir ciertos requisitos propios del hardware de adquisición de datos.

Se explicó el concepto de sensor, de aquí se concluye que estos serán la unidad básica de adquisición de la información para nuestra interfaz.

Se explicó el concepto de microcontrolador, alegando que se selecciona Arduino sobre las demás plataformas por ser la más económica.

Se mencionaron los estándares de salida a monitor y televisor y se mencionaron sus características.

CAPITULO II.

1.1 Introducción.

En este capítulo se comentan los pasos seguidos para la implementación de la interfaz: se realiza la descripción del proceso a monitorear o planta, los sensores que incorpora y como estos convierten las magnitudes en variables que se pueden monitorear. Más adelante se explica la comunicación que se establece entre las placas Arduino y el diseño e implementación del protocolo VGA y RCA que visualiza los datos leídos en pantalla.

1.2 Descripción de la planta.

El sistema de monitoreo se implementa tomando lectura de los sensores de una maqueta de casa automatizada. Dicha maqueta está disponible en el laboratorio de procesos de la carrera de Ingeniería Automática de la Facultad de Ingeniería Eléctrica de la Universidad de Oriente, en Santiago de Cuba. (Perez, 2016)



Figura 13 Maqueta de una casa domotica

Esta maqueta representa un sistema domótico convencional. Integra sensores en cada una de sus habitaciones y también realiza el control domótico. La implementación que se realiza sobre la maqueta es con fines demostrativos.

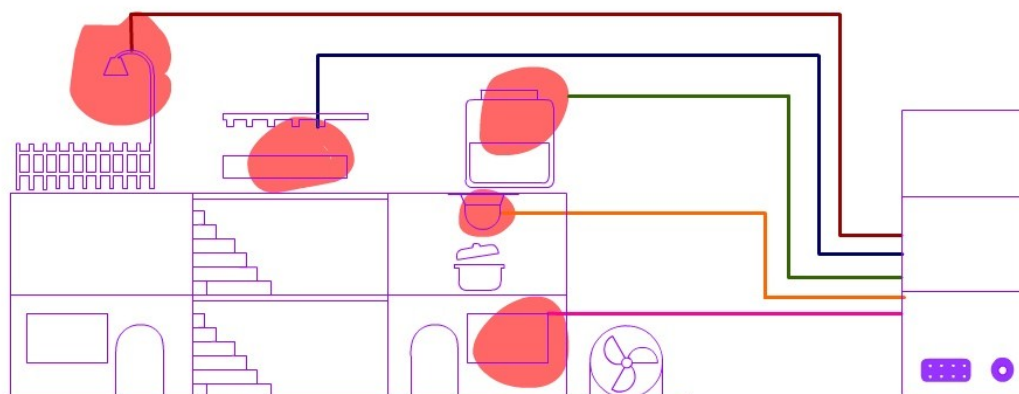


Figura 14 Diagrama básico de operación y ubicación de los sensores

En la figura se puede apreciar la disposición de los sensores escogidos en la maqueta. Las líneas de colores representan el cableado: rojo para el sensor de luz, azul para el sensor de humedad, verde para el sensor de nivel, naranja para el sensor de presencia de gas y por ultimo rosa para el sensor de temperatura.

En esta sección se propone realizar la adquisición de los datos de los sensores que ya están instalados en la maqueta. La interfaz propuesta deberá hacer la lectura de los sensores (a una frecuencia de muestreo predeterminada), almacenar cada una de estas lecturas en un archivo de registro y luego convertir estos datos en información de video para visualizar en un TV o monitor. Para ello la interfaz contará con conectores hembra de VGA y RCA.

1.3 Sensores.

Para este proyecto se escogieron convenientemente algunos sensores que están dispuestos en distintas ubicaciones de la maqueta que ofrecen sus lecturas de forma analógica:

Sensor de temperatura: En esta maqueta se emplea el sensor resistivo de temperatura CyL (SEN23292P). Este sensor tiene la característica que no

necesita acondicionamiento de señal debido a que fue diseñado específicamente para conectarse con Arduino. De hecho el acondicionamiento de la señal se realiza dentro del mismo sensor por medio de un circuito integrado (por tanto es un transductor). Internamente está constituido por un NTC, que devuelve un valor de resistencia que aumenta a medida que disminuye la temperatura. Tiene tres pines de entrada/salida.

- VCC: entrada de voltaje +5V
- GND: tierra
- SIG (señal): por este pin el sensor envía a Arduino una señal en voltaje correspondiente con el valor de la temperatura medida.

El sensor de temperatura está ubicado en el local del cuarto.



Figura 15 Módulo Sensor de Temperatura Cyl

Imagen tomada de <http://www.agelectronica.com>

Más datos:

Este módulo usa un termistor TTC3A103_34D que retorna la temperatura ambiente por medio de la variación de la resistencia que luego se utiliza para alterar Vcc (5V en el Arduino). Luego en Arduino se convierte ese valor de voltaje medido por el puerto analógico, en temperatura. Su rango de operación es de -40 a 125°C con una precisión de +/-1.5%.

Tabla 1 Especificaciones del sensor de temperatura Grove

Artículo	Mínimo	Típico	Máximo	Unidad
Dimensiones	2.0 x 2.0			cm
VCC	3.3	5	30	V

Corriente	0.5	-	5	mA
-----------	-----	---	---	----

Sensor de humedad del suelo: el sensor de humedad del suelo está ubicado en el jardín o cantero que está en la zona del techo de la casa. De igual manera que el sensor de temperatura, este sensor esta específicamente diseñado para Arduino e integra un circuito que realiza el acondicionamiento de señal en el propio instrumento, por lo cual se puede conectar directamente con Arduino sin necesidad de alguna interfaz externa. El funcionamiento de este sensor se basa en la medición de la resistencia del suelo para determinar el porcentaje de humedad.

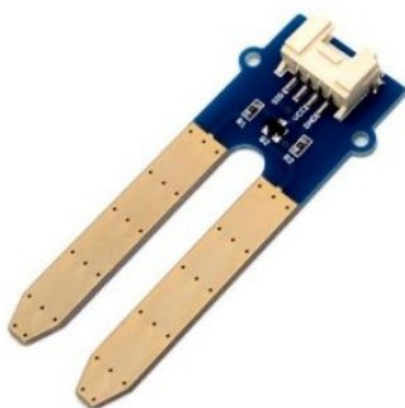


Figura 16 Módulo Sensor de Humedad del suelo CyL

Imagen tomada de <http://www.agelectronica.com>

Este sensor ofrece los siguientes valores de salida:

Tabla 2 Especificaciones del sensor de humedad del suelo

Artículo	Parámetros	Mínimo	Típico	Máximo	Unidades
VCC	-	3.3	-	5	V
Corriente	-	0	-	35	mA
Valor de salida	Seco	0	-	300	-
	Húmedo	300	-	700	-
	Sensor sumergido en agua	700	-	950	-

Sensor de nivel: El sensor de nivel usado en la maqueta es un Módulo Sensor de Agua CyL (SEN11304P) de la empresa Grove que también está diseñado específicamente para Arduino. Está ubicado en el tanque de agua que está en el techo de la casa. Es un sensor del tipo conductivo y ofrece valores analógicos mayores a medida que se sumerge más en el agua.



Figura 17 Módulo Sensor de Agua CyL.

Imagen tomada de <http://www.agelectronica.com>

Más datos:

El módulo sensor de agua es parte del sistema Grove e indica si el sensor está seco, húmedo o completamente sumergido en agua mediante la medición de la conductividad. Este sensor pondrá en alto (Vcc) su salida hasta que una gota de agua ponga en corto a dicho sensor, poniendo tierra a su salida (Gnd). Este circuito trabaja con los pines digitales de la placa Arduino o con los pines analógicos para detectar la cantidad de contacto entre las pistas de dicho sensor y tierra.

Tabla 3 Especificaciones del sensor detector de agua Grove

Artículo	Mínimo	Típico	Máximo	Unidad
Voltaje de trabajo	4.75	5	5.25	V
Corriente	<20			mA
Temperatura de trabajo	10	-	30	°C
Humedad de trabajo	10	-	90	%

Sensor de presencia de gas: Ubicado en la zona de la maqueta que representa la cocina se encuentra un sensor de presencia da gas. Este sensor emplea la conductividad térmica para la detección de gas. Es un Módulo Sensor de Gas y Humo MQ2 CyL (SEN90512P).



Figura 18 Módulo Sensor de Gas y Humo MQ2 CyL (SEN90512P).

Imagen tomada de <http://www.agelectronica.com>

El sensor de gas analógico (MQ2) se utiliza en la detección de fugas de gas de equipos en los mercados de consumo y la industria, este sensor es adecuado para la detección de gas LP, i-butano, propano, metano, alcohol, hidrógeno, tiene una alta sensibilidad, un tiempo de respuesta rápido. Y dicha sensibilidad puede ser ajustada por el potenciómetro.

Condiciones de trabajo:

- Voltaje de circuito: 5V.
- Voltaje de calentamiento: 5V.
- Resistencia de carga: puede ser ajustable.
- Resistencia del calentador: $33\Omega \pm 5\%$.
- Consumo: menos de 800mW.

Sensor de intensidad de luz: Los sensores de luz están ubicados en la zona del techo y sirven para controlar el iluminado general de la instalación domótica. Detectan si es de día o de noche y para ello emplean una fotorresistencia.



Figura 19 Módulo Sensor de luz CyL (SEN11302).

Imagen tomada de <http://www.agelectronica.com>

Es un sensor de la gama conectar y listo diseñado específicamente para Arduino. Este módulo consta de una fotorresistencia GL5528 para detectar la intensidad de la luz en el ambiente, cuyo valor depende de la intensidad de luz que recibe, a mayor cantidad de luz, menor resistencia. También posee un amplificador operacional LM358 configurado como seguidor de tensión y un conector de 4pines.

1.4 Arduino y sensores.

Como se explicó en epígrafes anteriores, la conexión de Arduino con los sensores es tan simple como conectar y listo. Estos sensores fueron diseñados por la empresa Grove que produce instrumentación a pedido para el proyecto Arduino. Para estos sensores no se necesita acondicionamiento de señal externo, ya que el propio sensor integra un circuito interno que lo hace.

El trabajo con los sensores de Arduino pasa entonces a un plano meramente programático, donde prácticamente todo se hace con código. Como se dijo antes, el valor de salida de los sensores es un voltaje que puede fácilmente interpretarse con Arduino como un valor de 0 a 1023 que se “lee” a través de los pines de entrada/salida analógicos. Esta operación se realiza por medio de la función *analogRead()*, que admite como parámetro el pin analógico que se escoja para conectar con el sensor.

Para el ajuste o calibración del rango leído de los pines, con el valor real medido por el sensor se puede realizar manualmente. Por ejemplo para ajustar el valor de temperatura ambiente en Arduino se puede utilizar un termómetro común y corriente: se hace la lectura con el termómetro y luego con el sensor. Luego por una simple relación de tres se saca cual es la constante que relaciona el valor leído con el valor real. También se puede utilizar alguna librería de código o las tablas que provee el fabricante de rango de entrada contra rango de salida del sensor. Para este proyecto se usó el primer método para el ajuste de la temperatura, ya que se consideró que era más preciso y porque había disponibilidad de un termómetro físico para realizar la comparación. Para los demás sensores se empleó el método de tablas y constantes.

```
314 values[TEMP] = map(analogRead(A0),0,1023,0, 35);
315 log_data_2_SD(TEMP);
316 values[HUMED] = map(analogRead(A1),0,1023,0,100);
317 log_data_2_SD(HUMED);
318 values[NIVEL] = map(analogRead(A2),0,1023,0,30);
319 log_data_2_SD(NIVEL);
320 values[PRESEN] = digitalRead(A3);
321 values[LUZ] = map(analogRead(A4),0,1023,0,9);
322 log_data_2_SD(LUZ);
```

Figura 20 mapeo de los valores de los sensores

En lo concerniente a Arduino se le llama mapeado al proceso de convertir los valores leídos de los pines en información útil. En la imagen anterior se mapean los valores de cada uno de los sensores para que coincidan con los valores de interés, para ello se utiliza la función *map*.

La variable *values* es de tipo arreglo y contiene los valores (mapeados) de cada uno de los sensores.

1.5 Comunicación Serie.

Debido a las características físicas de Arduino Uno, la placa que se ha utilizado en este proyecto (véase epígrafe Microcontroladores, Capítulo I), no es posible obtener toda la funcionalidad en una sola placa.

La generación de señales de video es muy costosa y se consume todos los recursos de la placa, haciendo imposible la lectura de los sensores en la misma. Por eso se separó la funcionalidad de lectura de los datos de los sensores y generación de señal de video en dos placas Arduino y estas a su vez se comunican a través del puerto serie asíncrono (UART).

Antes se explicó cómo se realizaba la lectura de los sensores. Pues bien, después de almacenar los datos en variables internas, estos se transmiten por el puerto UART empleando una codificación determinada. Esta codificación le indica al segundo Arduino la manera en que debe mostrar los datos en pantalla. Se puede decir que el segundo Arduino realiza la misma función que una tarjeta gráfica.

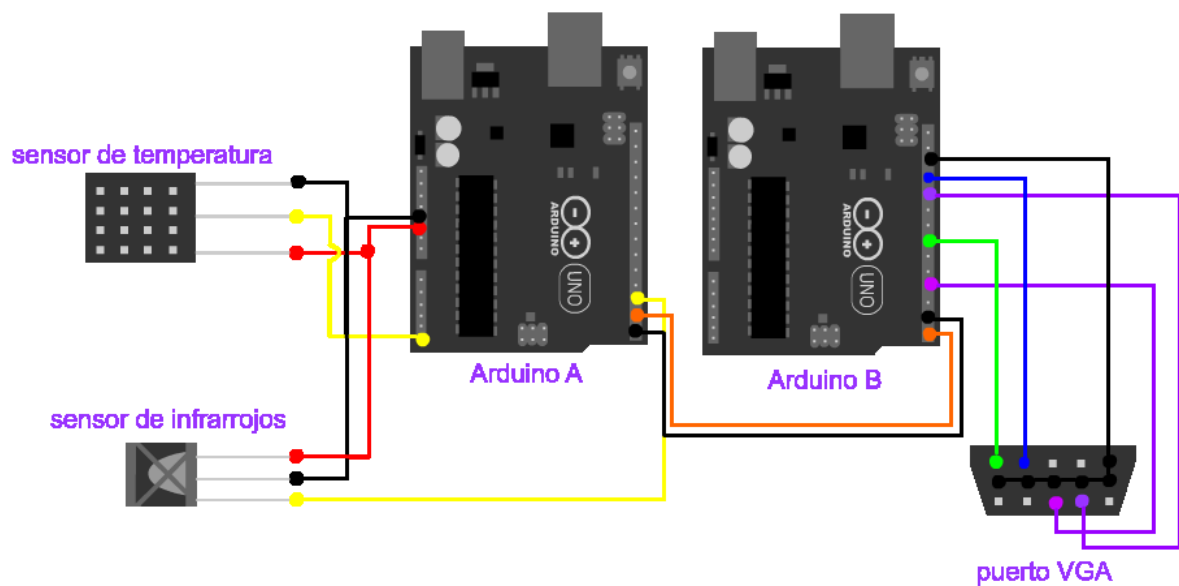


Figura 21 Conexión de ambas placas, los cables Naranja y Negro son RX y TX

1.6 Recepción de comandos por infrarrojos

Para interactuar con el menú se añadió la funcionalidad de recibir comandos desde un control remoto común y corriente, complementando así con el uso del televisor o monitor del usuario. El usuario no necesita comprar nuevos artefactos como un control remoto externo para poder usar este dispositivo.

```

7  #include "IRNEC1838.h"
8  #include <SD.h>
9  #include <SPI.h>
10 #include <avr/pgmspace.h>

```

Figura 22 la librería <IRNEC1838.h>

La recepción por infrarrojos se logra empleando la librería de código <IRNEC1838.h> que provee los métodos necesarios para detectar cada comando de tecla presionada en el control remoto. El único hardware externo que se

requiere para usar esta librería es un fotodiodo, un fototransistor o cualquier componente que pueda responder ante cambios en la intensidad luz, propiamente dicho, cambios de alta frecuencia en la intensidad de luz. (Ventura, 2017)

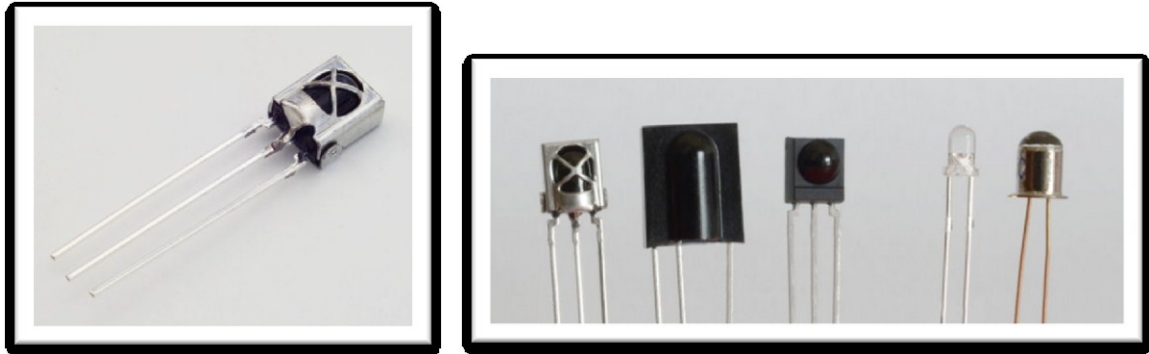


Figura 23 a) Sensor de infrarrojos TL1738HS b) Otros sensores de la misma gama.

Imágenes tomadas de <http://polaridad.es>

Para este proyecto se emplea un fototransistor modelo TL1738HS. El “38” del nombre hace referencia a que trabaja con una señal modulada a 38 KHz (Ventura, 2017).

```
106 void getOption(byte ir_opt){
107     switch(ir_opt){
108         case 2: //up
109             up();
110             break;
111         case 130: //down
112             down();
113             break;
114         case 224: //left
115             left();
116             break;
117         case 96: //right
118             right();
119             break;
120     }
```

Figura 24 Identificación del comando del sensor infrarrojo

En el código de la imagen, la función *getOption()* permite diferenciar el comando enviado desde el control remoto y ejecutar la instrucción correspondiente para cada orden.



Figura 25 Dispositivo para convertir cualquier Smartphone en un control remoto universal

Se probó con distintas configuraciones de controles remotos, y para ello se confeccionó un dispositivo de funcionalidad similar (Canvas, 2017). Este dispositivo permite convertir cualquier Smartphone Android o iOS en un mando universal empleando la aplicación ZaZaRemote (CNXSoft, 2017). Este artefacto solo necesita conectarse al conector del auricular del móvil.

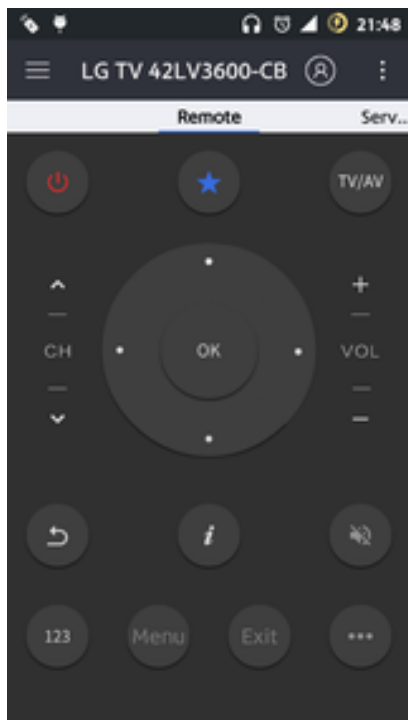


Figura 26 ZazaRemote en Android

Para este caso particular los comandos enviados fueron para un televisor LG, pero la interfaz puede configurarse para que funcione con cualquier control remoto.

1.7 Archivo de registro en tarjeta SD

Los sistemas de supervisión convencionales incorporan la característica de poder almacenar datos históricos del sistema que supervisan. A medida que un usuario incorpora cada vez más sensores en su sistema domótico se hace necesario llevar un registro de estos datos. Esto permitirá en el futuro conocer cómo se ha estado comportando el sistema e incluso detectar fallos. Para el caso particular de los sistemas domóticos tiene gran utilidad debido a que si una alarma no es manejada, o es desactivada sin el consentimiento del usuario, los datos generados durante la excepción quedan almacenados permanentemente en un archivo.

```
7  #include "IPNEC1838.h"
8  #include <SD.h>
9  #include <SPI.h>
10 #include <avr/pgmspace.h>
```

Figura 27 Librería <SD.h>

La interfaz que se lleva a cabo en este proyecto también tiene la funcionalidad de guardar un archivo de registro en una tarjeta microSD. Para ello Arduino cuenta con la librería <SD.h> que provee los métodos necesarios para crear, eliminar, leer y escribir en archivos. Después de realizar la medición de cada sensor, una variable contiene sus valores. Entonces se implementó una función que recibe como parámetro el sensor cuyo valor desea archivar.

```
326 void log_data_2_SD(byte s){
327     //log data string structure
328     //dd.mmm.aaaa hh:mm- xxxx:zzuu
329     TXbuffer = "";
330     if(day/10==0)
331         TXbuffer.concat("0" + String(day));
332     else
333         TXbuffer.concat(String(day));
334     TXbuffer.concat(".");
335     strcpy_P(buffer, (char*)pgm_read_word(&(months[month])));
336     TXbuffer.concat(buffer);
337     TXbuffer.concat(".");
338     TXbuffer.concat(String(year));
339     TXbuffer.concat(" ");
340     if(hour/10==0)
341         TXbuffer.concat("0" + String(hour));
342     else
343         TXbuffer.concat(String(hour));
```

Figura 28 Formato de la trama del archivo de registro

Dentro de la función se establece el formato (en ASCII) de la trama de datos que se guarda en el archivo de registro:

- Fecha: dd.mm.aaaa
- Separador: carácter de espacio
- Hora: hh.mm
- Separador carácter guion (-)
- Sensor: una cadena de cuatro caracteres en mayúsculas, por ejemplo TEMP
- Valor: el valor medido

Luego la función *println* de la librería SD hace el trabajo de actualizar el archivo de registro con los nuevos datos. El archivo se crea la primera vez en la ruta “/LOG/logdata.txt”. Desde entonces se actualiza constantemente hasta que no quede espacio libre tarjeta microSD.

El proceso de actualización del archivo de registro se realiza en intervalos de 5 segundos. Se escogió este valor debido a que las magnitudes que se registran no varían muy rápido, sin embargo este número podrá ajustarse luego, dependiendo de la utilización que se le dará.



Figura 29 Shield Ethernet y tarjeta microSD

El hardware que se empleó fue el Shield Ethernet de Arduino, que tiene funcionalidad para almacenar información en una tarjeta microSD. La tarjeta microSD usada tiene una capacidad de 4GB.

1.8 Protocolo de visualización VGA/RCA.

Normalmente, generar una señal de video empleando Arduino implica conocer a fondo los protocolos de tiempo de dicha señal. Esto envuelve el dominio de temas avanzados de programación como el uso de interrupciones, programación de registros del microcontrolador a bajo nivel y cálculos exhaustivos que tomarían mucho tiempo.

Para dibujar los elementos en pantalla se utilizaron algunas librerías para la generación de video. Por ejemplo la librería <VGAX.h> provee métodos para la generación de señal de video por VGA en Arduino Uno (Maffiodo, 2017) y la librería <TVout.h> provee métodos para la generación de señal de video por RCA (Avamander, 2017). Se optó por estas librerías porque no requieren el uso de hardware externo, la generación de video se lleva a cabo usando al propio Arduino como procesador de video.

La librería VGAX permite generar señales de video en el estándar VGA compatible con los monitores de las computadoras. La salida de la señal se obtiene por los siguientes pines:

- PIN3 – HSYNC
- PIN6 – color verde
- Pin7 – color azul
- PIN9 – VSYNC
- PIN14 – GND

Estos pines se conectan al conector hembra VGA utilizando soldadura de estaño.

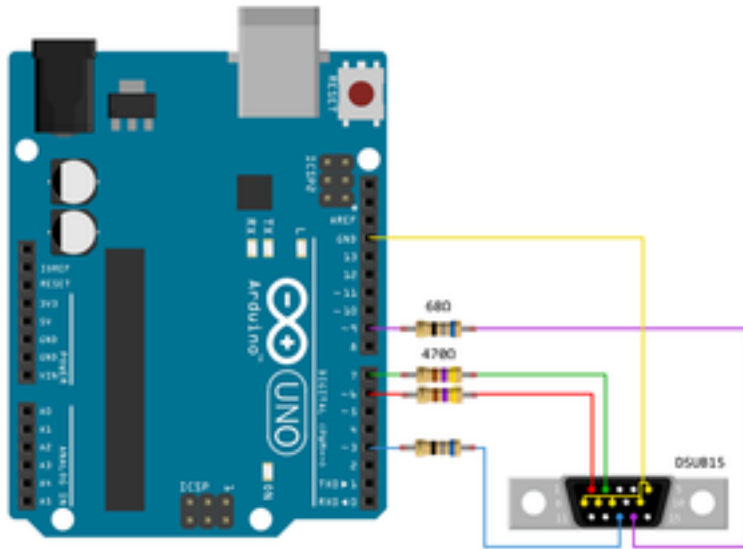


Figura 30 Diagrama de conexión con conector VGA

Imagen tomada de <https://github.com/smaffer/vgax>.

La librería *TVout* permite generar señales de video en el estándar RCA compatible con los televisores comunes. La salida de la señal se obtiene por los siguientes pines:

PIN7 – VIDEO

PIN9 – SYNC

PIN14 – GND

En los pines 7 y 9 se conecta una resistencia de 470Ω y de $1K\Omega$ respectivamente y estas a su vez se conectan con el polo central del conector RCA.

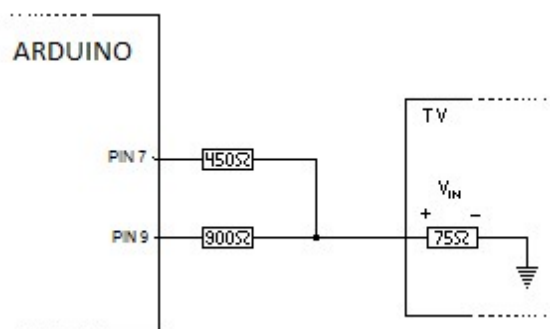


Figura 31 Diagrama de conexión con conector RCA

Imagen tomada de <http://www.eyetap.org/ece385/lab5.htm>

Ambas librerías proveen métodos que permiten dibujar texto y elementos simples en pantalla, como rectángulos y círculos. La librería VGAX soporta hasta 4 colores.

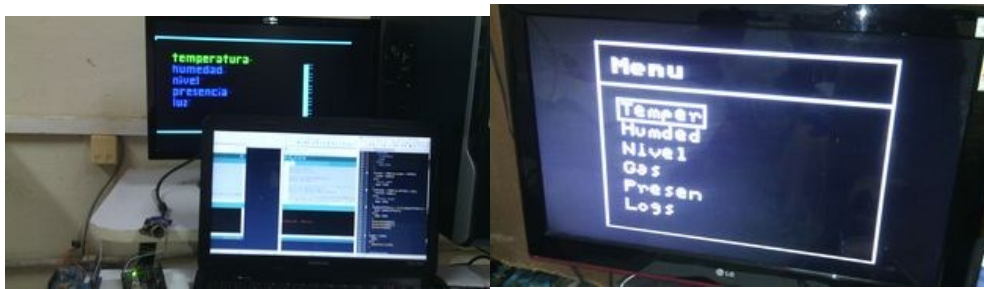


Figura 32 Diseño del menú gráfico en monitor y en televisor

Se diseñó un menú gráfico compuesto por tres pantallas:

- Pantalla principal: el usuario puede consultar aquí los valores censados



Figura 33 Pantalla principal en monitor y televisor

- Pantalla del archivo de registro (LOGS): el usuario puede consultar aquí las trazas del archivo.



Figura 34 Pantalla de LOGS en monitor y televisor

- Pantalla de configuración: en esta pantalla el usuario puede consultar y ajustar la fecha y la hora, idioma de la visualización, puede escoger si desea ver todos los valores al mismo tiempo o ver uno a la vez, en el caso de los monitores se podrá cambiar el esquema de color usado y por ultimo borrar el archivo de registro, en caso de que la tarjeta microSD no admita más datos. Nota: esta pantalla está en desarrollo, ya que todavía no se ha

logrado vincular las funcionalidades con la visualización en la tarjeta gráfica.



Figura 35 Pantalla de configuración en monitor y en televisor

1.9 Valoración económica

Este proyecto se concibió para ajustarse a las limitaciones monetarias de cualquier persona, siempre pensando en el mínimo precio posible sin sacrificar la calidad.

Debido al auge del proyecto Arduino y su popularidad, se ha convertido en una alternativa para pequeñas empresas por su reducido precio. Los precios en el mercado internacional de las tarjetas, kits, módulos y accesorios que conforman la plataforma hacen que el sistema de control domótico implementado sea costeable y pueda ponerse en práctica en países en vías de desarrollo, como el nuestro.

En la siguiente tabla se relaciona lo que se considera como presupuesto mínimo del proyecto.

Elemento	precio unitario	cantidad	total
Arduino Uno (manufacturado en China)	\$3	2	\$6
Shield Ethernet (manufacturado en China)	\$6	1	\$6
conector VGA hembra	\$0.25	1	\$0.25
conector RCA hembra	-	1	-
Sensor de infrarrojos	-	1	-
tarjeta micro SD	\$2	1	\$2
			\$14.25

También se tuvo en cuenta el empleo de la mínima instrumentación, es decir, cuando el usuario decida instalar la interfaz en su casa, no necesita adquirir nada más para hacerlo funcionar. Aprovechamos la disponibilidad del televisor y del control remoto además de los sensores, que se supone ya deben estar instalados. Los precios fueron tomados del mercado alternativo de Aliexpress (Aliexpres.com, 2017).

Conclusiones generales

- Se presentó un estudio de las características de los sensores que se usaron en la maqueta demostrativa
- Se caracterizó la plataforma de Arduino Uno y se explicó la razón por la cual se escoge sobre las demás plataformas.
- Se explicó en detalle el diseño del sistema de visualización, presentándose su esquema de montaje.
- Se construyó un dispositivo de emisión por infrarrojo que replica la función de un control remoto universal y que permite interactuar con el menú gráfico de la interfaz.
- Se implementó un menú gráfico como interfaz principal de visualización.
- Se evidenció la rentabilidad de la interfaz de monitoreo domótico.

Recomendaciones

- Emplear mejor hardware, Arduino Mega, esto posibilitará reducir en espacio el esquema de montaje y será mucho más simple. Pueden emplearse otras plataformas también.
- Aumentar el número de variables monitoreadas, o dar la posibilidad de aumentarlas.
- Añadir información de color para la salida RCA.
- Incrementar la profundidad de color de salida VGA.
- Optimización general del código, para lograr escalabilidad y robustez.
- Agregar la posibilidad de conexión inalámbrica con los sensores para permitir la movilidad del sistema.

Bibliografía

- A. Datum Corporation. (Enero de 2017). *SlideShare.net*. Obtenido de SlideShare.net: <http://www.slideshare.net/luigidr/domotics-an-open-approach>
- Aliexpress.com. (Enero de 2017). *Ventas de productos en Aliexpress*. Obtenido de Aliexpress: <http://www.aliexpress.com>
- Area Urbana.net. (2017). *Conexiones de Video (Analógicas y Digitales)*. Lima: Publicaciones Adventure Works.
- Avamander. (Enero de 2017). *GitHub.com - Arduino-TVout*. Obtenido de Arduino-TVout: <https://github.com/Avamander/arduino-tvout/>
- Brennan, R. W. (2007). *IEEE Transactions on systems, Man and cybernetics—Part C: Applications and reviews* (Vol. 37).
- Canvas, A. (Enero de 2017). *Instructables.com*. Obtenido de Instructables.com: <http://www.instructables.com/id/DIY-Infrared-transmitter-for-iPhone-iPod/?ALLSTEPS>
- CNXSoft. (2017, Enero). *CNXSoft – Embedded Systems News*. Retrieved from Transform Your Smartphone into a Universal IR Remote Control with ZaZaRemote: <http://www.cnx-software.com/2014/07/21/transform-your-smartphone-into-a-universal-ir-remote-control-with-zazaremate/>
- Colomer, J., Meléndez, J., & Ayza, J. (2010). *Sistemas de Supervisión*. Andalucía: CEA-IFAC.
- Embedded Related. (Enero de 2017). *PIC List*. Obtenido de PIC List: <http://www.embeddedrelated.com/groups/piclist/1.php>
- García, M. A., Antón, J. C., Rodríguez, J. C., Martín, F. J., & Ortega, G. J. (2004). *Instrumentacion Electronica*. Madrid: Thomson Editores Spain.
- Gridling, G., & Weiss, B. (2007). *Introduction To Microcontrollers*. Viena: Vienna University of Technology.
- Henao Merchán, Ó. D. (2006). *Hardware Y Software Domótico*. Universidad Pontificia Bolivariana, Facultad de Ingeniería Eléctrica y Electrónica. Medellín, Colombia: Universidad Pontificia Bolivariana.
- Jones, L., & Chin, A. F. (1983). *Electronic Instruments and Measurements*. Oklahoma : Oklahoma State university and Tulsa Junior College. Obtenido de Sistemas de Adquisición de Datos: <http://www.monografias.com/trabajos17/sistemas-adquisicion-dato/>
- Maffiodo, S. (Enero de 2017). *GitHub - VGAX Library for Arduino UNO*. Obtenido de VGAX Library for Arduino UNO: <https://github.com/smaffer/vgax>
- National Instruments. (1999). *Data Acquisition (DAQ) Fundamentals Application Note 007* (Vol. 007). Austin, Texas, USA: National Instruments Editorial.
- Parallax Inc. (Enero de 2017). *Parallax Inc. Home Page*. Obtenido de Parallax Inc. Home Page: <http://www.parallax.com/>
- Perez, A. P. (2016). *Diseño e implementación de un sistema de control domótico utilizando la plataforma Arduino*. Santiago de Cuba: Universidad De Oriente.
- Santa Emilia, C. (13 de Diciembre de 2015). *Aprenda Robótica con Arduino. Curso introductorio*. Obtenido de <http://www.viaspositronicas.blogspot.com>
- UNITEL. (Enero de 2017). *Domótica KNX – Más que ventajas*. Obtenido de KNX: <http://www.knx.org/es/>

Ventura, V. (Enero de 2017). *Polaridad.es*. Obtenido de Polaridad.es:
<https://polaridad.es/libreria-manejar-controlar-arduino-mando-distancia-infrarrojos/>

Anexos

Índice de Ilustraciones.

Figura 1 Sistema de mediciones genérico	- 1 -
Figura 2 Proceso de Adquisición de datos	- 2 -
Figura 3 Monitorización y supervisión	- 4 -
Figura 4 Interfaz de supervisión local ubicada en el Edificio Administrativo de la Universidad de Oriente en Santiago de Cuba	- 7 -
Figura 5 Ejemplos de interfaces móviles de monitoreo y control	- 7 -
Figura 6 El microcontrolador 8051 de Intel	- 9 -
Figura 7 Esquema general de un microcontrolador	- 11 -
Figura 8 Arduino Uno y Arduino IDE	- 13 -
Figura 9 Elegoo UNO R3 es un clon de Arduino, una alternativa a precio reducido.	- 14 -
Figura 10 Conector VGA de 15 pines	- 15 -
Figura 11 Conector RCA	- 16 -
Figura 12 Conectores del video por componentes analógico	- 17 -
Figura 13 Maqueta de una casa domotica	- 18 -
Figura 14 Diagrama básico de operación y ubicación de los sensores	- 19 -
Figura 15 Módulo Sensor de Temperatura CyL	- 20 -
Figura 16 Módulo Sensor de Humedad del suelo CyL	- 21 -
Figura 17 Módulo Sensor de Agua CyL.	- 22 -
Figura 18 Módulo Sensor de Gas y Humo MQ2 CyL (SEN90512P).	- 23 -
Figura 19 Módulo Sensor de luz CyL (SEN11302).	- 24 -
Figura 20 mapeo de los valores de los sensores	- 25 -
Figura 21 Conexión de ambas placas, los cables Naranja y Negro son RX y TX	- 26 -
Figura 22 la librería <IRNEC1838.h>	- 26 -
Figura 23 a) Sensor de infrarrojos TL1738HS b) Otros sensores de la misma gama.	- 27 -
Figura 24 Identificación del comando del sensor infrarrojo	- 27 -
Figura 25 convertir cualquier Smartphone en un control remoto universal	- 28 -
Figura 26 ZazaRemote en Android	- 28 -
Figura 27 Librería <SD.h>	- 29 -
Figura 28 Formato de la trama del archivo de registro	- 29 -
Figura 29 Shield Ethernet y tarjeta microSD	- 30 -
Figura 30 Diagrama de conexión con conector VGA	- 32 -
Figura 31 Diagrama de conexión con conector RCA	- 32 -
Figura 32 Diseño del menú grafico en monitor y en televisor	- 33 -