

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

i



PROYECTO FINAL

*"Sistema Scada para monitorizar Heladeras para
Vacunas del Hospital y Centros de Salud"*

Universidad Tecnológica Nacional

Facultad Regional de Neuquén

Año 2016

Alumno: Chandia Pablo Lucas

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

Dedicatoria

A los forjadores de mis valores:

Mis padres y Mis Hermanos

A los amores de mi vida

Mi mujer Mabel y Mis Hijos

Juan Cruz y Julian

Con todo mi amor y cariño

Lucas

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

Resumen:

El proyecto plantea una solución por medio un Microcontrolador con capacidad de comunicación Ethernet, para el registro y monitorización temperatura interna de las heladeras en las que se almacenan vacunas, medicación especiales , reactivos del Hospital Zonal Cutral Co Plaza Huincul y sus 12 Centros de Salud de Atención Primaria, de forma de poder avisar en forma remota a través de Internet cualquier variación fuera de los parámetros establecidos para poder asegurar las condiciones almacenamiento y mantener un registro continuo, exigidos por los entes de salud.

Contenido

Dedicatoria.....	i
Resumen:	ii
Indice de Tablas	vii
Indice de Ilustraciones.....	viii
Introduccion	x
Introducción a las Vacunas	xi
Vacunas y Conservación de las Mismas	xii
Planteamiento del Problema.....	xiii
Justificación.....	xiv
Hipótesis	xiv
Objetivos General.....	xv
Objetivos Secundarios	xv
Contenido del documento de tesis	xv
Capitulo 1-Marco Teórico	1
1.1.1-Origen de las vacunas	2
1.1.2-Tipos de vacunas	4
1.1.3- Desarrollo de la inmunidad	5
1.2-Mantenimiento de las Vacunas	6
1.2.1-Características propias de la Vacuna	6
1.2.2-Mantenimiento de las Vacunas.....	7
1.2.3-Cadena de frío.....	8
1.2.4-Elementos de la cadena de frío.....	9
1.2.5-Rangos de temperaturas de conservación de las vacunas.....	10
1.3.1-Valores comerciales de las Vacunas.....	18

Capitulo 2-Sistemas de control de temperatura.....	20
2.1-Termometros	20
2.1.1-Termometros de Alcohol.....	20
2.1.3-Termometro de máxima y mínima.....	21
2.2-Sistemas Electronicos	22
2.2.1-Sistemas de alarma por temperaturas altas o bajas	22
2.2.3-Centinelas Alarma por temperatura.....	24
2.2.4-Datalogger Ethernet de Temperatura A205.....	25
2.2.5-Registrador de Temperatura con soporte Web TandD TR-71W	29
Capitulo 3-Desarrollo del sistema	33
3.1-Diseño del sistema propuesto.....	33
3.1.1-Requisitos	33
3.1.2-Alternativas al diseño:.....	35
3.3- Plataforma de desarrollo de hardware	38
3.3.1-WIZ220IO	39
3.3.2 PIC-MINI-WEB.....	42
3.3.3- Mbed - LPC1768 Development Board	45
3.3.4-Arduino	48
3.4-Selección de plataforma y experiencia en la compra.....	53
3.4.1-Modularidad del Sistema.....	54
3.4.2- Seleccion del Shield Ethernet	55
3.4.3- Ethernet Shield V1.0 basado en ENC28J60	55
3.4.3-Ethernet Shield Con Wiznet W5100.....	57
3.5- Sensores de Temperatura	59
3.5.1-Sensor de Temperatura LM35	61
3.5.2-Sensor de Temperatura DS18B20.....	64

3.6- Selecccion del Display LCD.....	67
3.6.1-LCD Keypad Shield para Arduino.....	69
3.6.2 Adaptador I2c para Display PCF8754.....	71
3.6.3-Elecccion del Display	74
3.7- Alarma Sonora	74
3.7.1- Interruptor de Apertura de Puerta de Heladera.....	76
3.8- Herramientas de software para el desarrollo	82
3.9- Introducci3n al concepto M2M	83
3.10-Internet de las cosas(Internet of things)	93
3.10.1- APIs para el Internet de las Cosas: ThingSpeak y Pachube	95
3.10.2-ThingSpeak API.....	96
3.10.3- IoT ThingSpeak Monitor Widget.....	99
3.10.4-.Pachube API (Xively)	101
Capitulo 4- Interacci3n e implementaci3n.....	103
4.1-Hardware.....	103
4.2-Software.....	109
4.2.1-Ethernet.h	109
4.2.2-OneWire.h	120
4.2.3 DallasTemperature.h	122
4.2.3- LiquidCrystal.h.....	125
4.3-Diseño software.....	130
4.3.1- Algoritmo de Inicializacion (Setup).....	131
4.3.2- Algoritmo de Ejecuci3n C3clica (Loop).....	131
4.3.4-Diagrama de Flujo	133
4.4-Codigo final de Proyecto	135
Capitulo 5 - Experiencia, pruebas y resultados	140

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

5.1-Experiencia en el desarrollo del Sistema	140
5.2-Mango M2M (Metodo Post y Get).....	142
5.3-Experiencia con LM35 como sensor de temperatura.....	143
5.4-Experiencia con el DS18B20 como sensor de temperatura	146
5.5-Experiencia con ThinkSpeak.....	147
5.6-Experiencia con IoT ThingSpeak Monitor Widget	149
Capitulo 6-Conclusiones.....	153
6.1-Trabajos Futuros.....	154
Bibliografia	155

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

Indice de Tablas

Tabla 1-Tabla orientativa de termoestabilidad de las vacunas	14
Tabla 2-Valores comerciales de las vacunas	19
Tabla 3-Especificaciones WIZ220IO	40
Tabla 4-Variables, Registros y Objetos en el Sketch del Programa	130

Indice de Ilustraciones

Ilustración 1-Termometro digital y registro de temperatura.....	16
Ilustración 2-Heladera de vacunas tipica.....	17
Ilustración 3-Termometro de Alcohol.....	20
Ilustración 4-Termometro Bimetalico.....	21
Ilustración 5-Termometro de Maxima y Minima.....	22
Ilustración 6-Registro de Temperatura.....	22
Ilustración 7-Sistemas de alarma por temperaturas altas o bajas.....	23
Ilustración 8-Sistema de alarma con marcador telefónico.....	23
Ilustración 9-Centinelas Alarma por temperatura.....	24
Ilustración 10-Datalogger Ethernet de Temperatura A205.....	25
Ilustración 11-Registrador de Temperatura con soporte Web.....	29
Ilustración 12-Interconexion Registrador de Temperatura con soporte Web.....	30
Ilustración 13-Canales de comunicacion del sistema scada propuesto.....	37
Ilustración 14-Canales de comunicacion del sistema propuesto sin scada.....	38
Ilustración 15-WIZ220IO.....	39
Ilustración 16-PIC-MINI-WEB.....	42
Ilustración 17-Diagrama Ciurcuital pic-Mini-web.....	44
Ilustración 18-Mbed - LPC1768 Development Board.....	45
Ilustración 19-Mbed - LPC1768 Distribucion de pines.....	46
Ilustración 20-Arduino Uno genuino.....	48
Ilustración 21-Ethernet Shield V1.0 para Arduino (Microchip's ENC28J60 SPI).....	50
Ilustración 22-Ethernet Shield w/microsd Card Reader (Wiznet W5100 Ethernet controller).....	50
Ilustración 23-Arduino Ethernet Shield V1.0.....	56
Ilustración 24-Bloques del ENC28j60.....	56
Ilustración 25-Circuito de Ethernet Shields v1.0.....	57
Ilustración 26-Ethernet Shield Con Wiznet W5100.....	59
Ilustración 27-pines de i/o Ethernet Shield Wiznet W5100.....	59
Ilustración 28-Sensor LM35.....	62
Ilustración 29-Arduino + LM35.....	63
Ilustración 30-Arduino + LM35 montado en proboard.....	63
Ilustración 31-Diagrama de bloques de DS18B20.....	64
Ilustración 32-Dallas ds18b20 Ilustración 33-DS18B20 Pines.....	65
Ilustración 34-Arduino + DS18B20.....	66
Ilustración 35-Arduino + DS18B20 montado en protoboard.....	66
Ilustración 36-lcd LCD1602A.....	67
Ilustración 37-Diagrama de conexion Arduino con LCD 16x2.....	68
Ilustración 38-ILUSTRACIÓN 37-CONEXION real ARDUINO CON LCD 16X2.....	68
Ilustración 39-LCD Keypad Shield.....	69
Ilustración 40-Circuito de Botones Shield LCD+Keyboard.....	71
Ilustración 41-Adaptador I2c para Display PCF8754.....	72
Ilustración 42-Conexion de pcf8754 a lcd.....	72
Ilustración 43-PCF8754 montado sobre lcd.....	72

Ilustración 44-Diagrama de conexión Arduino + Buzzer	76
Ilustración 45-conexión real Arduino + Buzzer.....	76
Ilustración 46-Magnetic reed	77
Ilustración 47-Magnetic reed utilizado como sensor de puerta	77
Ilustración 48-Apertura y Cierre de Magnetic Reed	79
Ilustración 49-Montaje sin resistencia Pull-Down.....	80
Ilustración 50-Montaje con resistencia Pull-Down.....	80
Ilustración 51-Montaje en arduino sin resistencia pull-down	81
Ilustración 52-MONTAJE EN ARDUINO con RESISTENCIA PULL-DOWN	81
Ilustración 53-Logo de Mango M2M	84
Ilustración 54-Dispositivo M2M	84
Ilustración 55-Descripción gráfica del mundo interconectado.	93
Ilustración 56-Captura de pantalla de canales de thingspeak.....	97
Ilustración 57-Graficos de canales thinkpeak	98
Ilustración 58-iot thingspeak Monitor Widget.....	99
Ilustración 59-pantalla de celular con la aplicación iot thingspeak Monitor Widget.....	100
Ilustración 60-Alarma IOT THINGSPEAK MONITOR WIDGET	100
Ilustración 61-Grafico generado por IOT THINGSPEAK MONITOR WIDGET	101
Ilustración 62-Pachube	102
Ilustración 63-Arduino Uno Genuino	103
Ilustración 64-Ethernet Shield Ws5100 Mega Compatible	104
Ilustración 65-Shield LCD Keypad 16x2	105
Ilustración 66-SMCB de X28	107
Ilustración 67-Circuito completo de conexión entre todos los componentes.....	108
Ilustración 68-diagrama de Flujo 1.....	133
Ilustración 69-DIAGRAMA DE FLUJO 2.....	134
Ilustración 70-Comunicación cliente servidor.....	143
Ilustración 71-Registro con Mango m2m	144
Ilustración 72-Grafico obtenido con M2M.....	144
Ilustración 73-Datos exportados a txt	145
Ilustración 74-Grafico obtenido con sensor Ds18B20.....	146
Ilustración 75-Archivo de datos con DS18B20 exportado a TXT	147
Ilustración 76-Captura thingspeak canal.....	148
Ilustración 77-Graficos del canal.....	149
Ilustración 78-Configuración Widget	150
Ilustración 79-pantalla principal con Widget.....	150
Ilustración 80-alarma en barra de notificaciones	151
Ilustración 81-reconociendo Alarma	151
Ilustración 82-Grafico de Temperatura.....	152
Ilustración 83-Grafico de Puerta.....	152

Introduccion

El sistema de almacenamiento de vacunas de nuestros hospitales y centros de atención primaria de salud, consisten en heladeras de uso domésticos, la cuales regulan la temperatura por medio del termostato. Estas heladeras no cuentan con ningún sistema de seguridad ni de aviso en caso de corte de suministro eléctrico o mal funcionamiento. Para poder asegurar que se sigan manteniendo el frío durante algunas horas, ante algunos inconvenientes, se colocan botellas de agua en la parte inferior y en el congelador bolsas de gel refrigerante.

La temperatura esta monitorizada por termómetros de alcohol que solo se puede tomar su lectura si uno abre la heladera, no poseen un sistema de lectura externa.

La hermeticidad de la heladera es también un punto clave, si en una heladera, algún burlete no cierra bien, se produce un gradiente térmico en esa zona, si por error la heladera queda mal cerrada o se tarda demasiado tiempo en volver a cerrar la puerta también ocurre lo mismo, con lo cual no tenemos certeza si se corto la cadena de frío. No se tiene ningún sistema de aviso ante estas situaciones en la cual interviene el factor humano en la manipulación del sistema de almacenamiento de vacunas.

Nuestro sistema surgió de estas inquietudes que recibimos en el Servicio de Electromedicina del Hospital Complejidad VI de Cutral-Có y Plaza Huincul, dado de que nosotros somos los encargado del mantenimiento y calibración de las heladeras del hospital y 12 centros de atención primaria de la salud.

La idea principal del proyecto es diseñar e implementar un sistema de monitorización local y remota de las heladera utilizadas en el almacenamiento de vacunas, las misma en el Hospital Zonal de Cutral-Co Plaza Huincul, esta distribuidas en dos almacenes principales de 6 heladeras , pero dentro del hospital hay 8 heladeras con menor cantidad pero no menor importancia.

Introducción a las Vacunas

Las vacunas se utilizan para reforzar el sistema inmunitario y prevenir enfermedades graves y potencialmente mortales.

El impacto de las acciones de vacunación ha contribuido a nivel mundial en la reducción de las enfermedades prevenibles por vacunas, por esta razón es que se le ha otorgado a esta estrategia un lugar central para la reducción de la mortalidad infantil y materna.

En este punto podríamos afirmar que el éxito o el fracaso de un sistema de inmunizaciones, dependerá básicamente de la calidad y potencia inmunológica de las vacunas utilizadas y por eso es fundamental el mantenimiento de condiciones adecuadas desde el laboratorio productor hasta su aplicación en el usuario final.

Las redes de distribución son complejas, lo que hace que en cada punto crítico se tenga especial cuidado en la detección y corrección de las excursiones, en las condiciones de conservación. El almacenamiento y conservación en condiciones térmicas adecuadas de las vacunas representan el aspecto más crítico de todo proceso de vacunación.

Los mayores inconvenientes derivados de la pérdida de la cadena de temperatura controlada se traducen en:

- Pérdida de eficacia
- Cambios en la seguridad
- Reducción del periodo de eficacia

En este punto todos los esfuerzos concurrentes pueden fracasar, con mayor razón si no se ha tenido la previsión de evaluar y asegurar la capacidad frigorífica de los centros de almacenamiento de vacunas.

Para realizar esta tarea, mayormente se utilizan termómetros, sean de mercurio y alcohol, o termómetros digitales. La lectura es in-situ, siempre y cuando se encuentre el personal en el Vacunatorio o Depósito. Cuando no se encuentra el personal, y ante una falla eléctrica o del sistema de refrigeración, el mantenimiento de las vacunas no están asegurado.

Dado el avance del uso de computadoras e internet en casi todos los centros de salud, se presenta la oportunidad de diseñar un sistema de monitoreo de temperatura remoto.

Vacunas y Conservación de las Mismas

La cadena de frío es el proceso que permite mantener la capacidad inmunizante de una vacuna desde su elaboración hasta su administración mediante su conservación a la temperatura idónea (entre +2° y +8° C) en todo momento.

El almacenamiento, mantenimiento, transporte y conservación de las vacunas deben reunir unas condiciones adecuadas.

El Frigorífico o Heladera es un elemento esencial en la cadena de frío, por lo que un mal funcionamiento del refrigerador puede destruir todo el programa de vacunación de un lugar.

Para asegurar su buen funcionamiento es imprescindible que el aparato reúna una serie de requisitos:

- Destinado exclusivamente a la conservación de vacunas. No debiendo almacenarse con las vacunas otros materiales: comida, material radioactivo, etc.
- Estar dotado de termostato (entre +2°C y +8°C).
- Verificar que mantiene una temperatura en su interior entre +2°C y +8°C.

Instalar un termómetro dentro del frigorífico, dejarlo permanentemente dentro y a la vista en la zona central (nunca en la puerta) y verificar todos los días al llegar y al finalizar la sesión de vacunación la temperatura que marca. Los termómetros de máxima-mínima son los más

recomendados porque indican las variaciones de temperatura producidas a lo largo del día en el interior de la Heladera

Respetar una serie de recomendaciones:

-La Heladera o Frigorífico debe estar situado a la sombra, alejado de toda fuente de calor y a unos 15 cm de distancia con respecto a la pared.

Estar conectados a la red general, no a derivaciones, para evitar desconexiones accidentales. Estar dotado de sistemas de alarma y generadores eléctricos de emergencia que se ponen en marcha si se desconecta accidentalmente el refrigerador. Cada centro deberá asegurarse aquel sistema que garantice un mejor control y vigilancia de la temperatura.

-Abrir la puerta solamente lo imprescindible.

-Se descongelará periódicamente, porque la acumulación de escarcha disminuye la capacidad frigorífica.

-Colocar en el congelador acumuladores de frío (bolsas "Ice-Pack") y en los últimos estantes de la Heladera botellas llenas de agua salada o de suero fisiológico, ya que ayudan a estabilizar la temperatura interna del frigorífico y en caso de corte de electricidad pueden mantener el frío durante 8 horas siempre y cuando el frigorífico en este lapso de emergencia no se abra. Deben guardar entre sí una distancia de 2,5 a 5 cm y estar situadas a idéntica distancia de las paredes del frigorífico para que el aire circule.

Planteamiento del Problema

Uno de los problemas principales en la institución es que los lugares asignados para instalar las heladera de vacunas, no estaban independizado de la instalación eléctrica, de forma que en algún desperfecto eléctrico producido fuera de horario de presencia del personal de vacunatorio, el desperfecto no era detectado, con lo cual al día siguiente si el suministro eléctrico

no se reponía durante las 8 horas siguientes, se encuentran con el problema de que se había perdido la cadena de frío.

También, no se dispone de un sistema de registro continuo de temperatura de las heladeras, de forma de poder realizar la trazabilidad del almacenamiento a la temperatura adecuada de 2 a 8 °C, solo se registra en un formulario a las 8 y 16 hs de cada día.

Si por alguna razón falla alguna heladera, no hay forma de saber hasta el día siguiente cuando se va a tomar nuevamente el registro.

Si se detecta algún inconveniente en alguna heladera, se debe convocar a la jefa del vacunatorio la cual evalúa si las vacunas siguen en condiciones óptimas y si es necesario reubicarlas en otra heladera.

Por todo estos problemas, se necesita un sistema de monitoreo y alarmas, que es la idea de proyecto propuesto.

Justificación

Ante varias pérdidas de vacunas en la institución, no se cambiaron los sistemas de medición de temperatura de heladeras ni los sistemas de alarma con lo cual queda expuesto a nuevamente pueda suceder. El sistema planteado, permitiría avisar para poder realizar acciones ante una inminente falla en la cadena de frío.

Hipótesis

Mediante el diseño de un sistema de monitorización local y remota de la temperatura en las heladeras será posible obtener los datos de temperatura instantáneas e históricas, así como poder utilizarlo como sistema de alarma ante valores máximo y mínimos críticos de mantenimiento de las vacunas.

Objetivos General

Diseñar y construir un sistema de monitorización de temperatura de heladeras dedicadas para vacunas, que permita ver los valores instantáneos en forma local y remota, de forma de avisar en caso de los parámetros se excedan de los valores normales configurados.

Objetivos Secundarios

- Implementar conexión Ethernet dentro de la red utilizada en el hospital
- Implementa conexión Ethernet externar por medio de Internet.
- Implementar la conexión Wifi para los CAPS.
- Evaluar algún sistema de diagnostico preventivo para prevenir fallas en heladera.
- Documentar el desarrollo del trabajo de investigación y pruebas

Contenido del documento de tesis

Este documento de tesis está estructurado de la siguiente manera:

El capítulo 1 presenta una breve introducción sobre las vacunas y su mantenimiento.

El capítulo 2 describe los sistemas de control de temperatura utilizados en las heladeras de vacunas.

El capítulo 3 presenta la metodología de desarrollo de sistemas

El capítulo 4 la interacción e implementación del hardware y software

El capítulo 5 plantea las conclusiones y los trabajos futuros de investigación.

Por último se presentan las referencias bibliográficas

Capítulo 1-Marco Teórico

Las vacunas son preparados que se realizan a partir de ciertos microorganismos que producen determinadas enfermedades. La administración de estas vacunas (antígenos) estimula al sistema inmunológico a producir anticuerpos (defensas) contra esa enfermedad.

En la actualidad disponemos de una amplia variedad de vacunas que son seguras y efectivas, propiedades que, sin embargo, no son suficientes para garantizar la eficiencia de los programas de vacunación. Es imprescindible que dichas vacunas, además de ser accesibles a la población, lleguen en perfecto estado de conservación, de forma que pueda garantizarse tanto su inmunogenicidad como su eficacia protectora.

La vacuna (del latín "vaccinus-a-um", "(vacuno)"; de "vacca-ae", "vaca") es un preparado de antígenos que una vez dentro del organismo provoca la producción de anticuerpos y con ello una respuesta de defensa ante microorganismos patógenos. Esta respuesta genera, en algunos casos, cierta memoria inmunitaria produciendo inmunidad transitoria frente al ataque patógeno correspondiente.

Las vacunas son el principal logro de la investigación biomédica y una de las principales causas de la mejora de la salud y la calidad de vida del ser humano. Desde el comienzo de las epidemias en China, la experiencia y la observación dieron lugar a los primeros métodos de profilaxis, la variolización.

La primera vacuna descubierta fue la usada para combatir la viruela por Edward Jenner en 1796,2 y debe su nombre al hecho de que las ordeñadoras de la época que estaban en contacto con la viruela de vaca (viruela "vacuna"), la cual era menos patógena, hacía que estas personas se inmunizaran y no contrajeran la viruela humana.

Clasificación

Las vacunas se clasifican en dos grandes grupos:

- Vacunas vivas atenuadas.
- Vacunas muertas o inactivadas.

Existen varios métodos de obtención:

- Vacunas avirulentas preparadas a partir de formas no peligrosas del microorganismo patógeno.
- Vacunas posificadas a partir de organismos muertos o inactivos.
- Antígenos purificados.
- Vacunas genéticas.

Las vacunas se administran por medio de una inyección, o por vía oral (tanto con líquidos como con pastillas).

1.1.1-Origen de las vacunas

La viruela fue la primera enfermedad que el ser humano intentó prevenir inoculándose a sí mismo con otro tipo de enfermedad. Se cree que la inoculación nació en la India o en China alrededor del 200 a. C. En China, a los pacientes que sufrían tipos leves de viruela se les recogían fragmentos de pústulas secas para molerlas hasta conseguir una mezcla con aspecto de polvo que luego se le introducía por la nariz, esperando que esto les inmunizara. En 1718, Lady Mary Wortley Montague informó que los turcos tenían la costumbre de inocularse con pus tomado de la viruela vacuna. Lady Montague inoculó a sus propios hijos de esta manera.

En 1796, durante el momento de mayor extensión del virus de la viruela en Europa, un médico rural de Inglaterra, Edward Jenner, observó que las recolectoras de leche adquirían

ocasionalmente una especie de «viruela de vaca» o «viruela vacuna» (cowpox) por el contacto continuado con estos animales, y que luego quedaban a salvo de enfermar de viruela común.

Efectivamente se ha comprobado que esta viruela vacuna es una variante leve de la mortífera viruela «humana». Trabajando sobre este caso de inoculación, Jenner tomó viruela vacuna de la mano de la granjera Sarah Nelmes. Insertó este fluido a través de inyección en el brazo de un niño de ocho años, James Phipps. El pequeño mostró síntomas de la infección de viruela vacuna. Cuarenta y ocho días más tarde, después de que Phipps se hubiera recuperado completamente de tal enfermedad, el doctor Jenner le inyectó al niño infección de viruela humana, pero esta vez no mostró ningún síntoma o signo de enfermedad.

En 1881 Louis Pasteur lleva a cabo su audaz y brillante experimento público en comprobación de la efectividad de la vacuna antiantráxica ideada por él, en la granja, hoy histórica, de Pouilly-le-Fort. El desarrollo del experimento fue como sigue[cita requerida]:

“...El 5 de mayo inyecta 24 carneros, 1 chivo y 6 vacas con 58 gotas de un cultivo atenuado de Bacillus anthracis. En mayo 17, estos mismos animales fueron inoculados nuevamente con la misma cantidad de un cultivo menos atenuado, o sea más virulento. En mayo 31 se realizó la prueba suprema. Se inyectaron con cultivos muy virulentos, todos los animales ya vacunados, y además, 24 carneros, 1 chivo y 4 vacas no vacunados, que sirvieron como grupo testigo a la prueba. En junio 2, una selecta y nutrida concurrencia apreció los resultados, que fueron los siguientes: Todos los carneros vacunados estaban bien. De los no vacunados, 21 habían muerto ya, 2 más murieron durante la exhibición ante la propia concurrencia y el último al caer de la tarde de ese día. De las vacas, las 6 vacunadas se encontraban bien, mientras que las 4 no vacunadas mostraban todos los síntomas de la enfermedad y una intensa reacción febril.

Louis Pasteur”

Al comunicar estos resultados, Pasteur introdujo los términos de vacuna y vacunación que provienen de la palabra latina vacca, fruto de los resultados obtenidos al inocular el virus de la vacuna (cow-pox); en la terminología médica como homenaje a Jenner, su ilustre predecesor.

1.1.2-Tipos de vacunas

Las vacunas pueden estar compuestas de bacterias o virus, ya sean vivos o debilitados, que han sido criados con tal fin. Las vacunas también pueden contener organismos inactivos o productos purificados provenientes de aquellos primeros. Hay cuatro tipos tradicionales de vacunas:

Inactivadas:

Microorganismos dañinos que han sido tratados con productos químicos o calor y han perdido su peligro. Este tipo de vacunas activa el sistema inmune pero es incapaz de reproducirse en el huésped. La inmunidad generada de esta forma es de menor intensidad y suele durar menos tiempo, por lo que este tipo de vacuna suele requerir más dosis. Dado que la respuesta inmune lograda es menor, se utilizan en estas vacunas unas sustancias denominadas adyuvantes. Estas sustancias están compuestas por aluminio y sirven a la vacuna a aumentar la respuesta inmunitaria del organismo. Los compuestos de aluminio deben inyectarse por vía intramuscular profunda ya que pueden producir irritación, inflamación y lesión de tejidos. Ejemplos de este tipo son: la gripe, cólera, peste bubónica y la hepatitis A.

Vivas atenuadas:

Microorganismos que han sido cultivados expresamente bajo condiciones en las cuales pierden sus propiedades nocivas. Suelen provocar una respuesta inmunológica más duradera, y son las más usuales en los adultos. Esto se debe a que el microorganismo no se encuentra inactivado y conserva su estructura. Por eso, en muchas ocasiones puede provocar la enfermedad en personas inmunodeprimidas. Por ejemplo: la fiebre amarilla, sarampión o rubéola (también llamada sarampión alemán) y paperas.

Toxoides:

Son componentes tóxicos inactivados procedentes de microorganismos, en casos donde esos componentes son los que de verdad provocan la enfermedad, en lugar del propio microorganismo. Estos componentes se podrían inactivar con formaldehído, por ejemplo. En este grupo se pueden encontrar el tétanos y la difteria.

Subunitarias:

Introduce un componente subcelular purificado de un microorganismo (por ejemplo, una proteína) dentro del sistema inmunitario, para crear una respuesta inmunitaria. Un ejemplo característico es la vacuna subunitaria contra la hepatitis B, que está compuesta solamente por la superficie del virus (superficie formada por proteínas).

1.1.3- Desarrollo de la inmunidad

El sistema inmunitario reconoce los agentes de la vacuna como extraños, destruyéndolos y «recordándolos». Cuando una versión realmente nociva de la infección llega al organismo, el sistema inmunitario está ya preparado para responder:

- 1º) Neutralizando al agente infeccioso antes de que pueda entrar en las células del organismo;
- 2º) Reconociendo y destruyendo las células que hayan sido infectadas, antes de que el agente se pueda multiplicar en gran número.

Las vacunas han contribuido a la erradicación de la viruela, una de las enfermedades más contagiosas y mortíferas que ha conocido la humanidad. Otras como la rubéola, la polio, el sarampión, las paperas, la varicela-zoster (virus que puede producir la varicela común y el herpes zóster) y la fiebre tifoidea no son tan comunes como hace un siglo. Dado que la gran mayoría de la gente está vacunada, es muy difícil que surja un brote y se extienda con facilidad. Este fenómeno es conocido como "inmunidad colectiva". La polio, que se transmite sólo entre humanos, ha sido el objetivo de una extensa campaña de erradicación que ha visto restringida la polio endémica, quedando reducida a ciertas partes de cuatro países (India, Nigeria, Pakistán y

Afganistán). La dificultad de hacer llegar la vacuna a los niños ha provocado que la fecha de la erradicación se haya prolongado hasta la actualidad.

1.1.4-Vacunas y economía

La economía es uno de los mayores retos de las vacunas. Muchas de las enfermedades que más demandan una vacuna (incluyendo el sida, la malaria o la tuberculosis) están presentes especialmente en países pobres. A pesar de que algunas empresas farmacéuticas y compañías de biotecnología han incentivado el desarrollo de vacunas para estas enfermedades limitadamente (dado que las expectativas de ingresos son bajas) el número de vacunas realmente administradas ha aumentado dramáticamente en las últimas décadas, especialmente aquellas suministradas a los niños en los primeros años de vida. Esto quizás se deba más a medidas gubernamentales que a incentivos económicos. La mayoría del desarrollo de vacunas hasta la fecha se ha debido a impulsos de gobiernos y ONG, agencias internacionales, universidades...

Muchos investigadores y políticos hacen un llamamiento para unir y motivar dicha industria, usando mecanismos de presión como los precios, impuestos o compromisos empresariales que puedan asegurar la retribución a las empresas que exitosamente consigan una vacuna contra el VIH (causante del sida).

(WIKIPEDIA) (LABORATORIO RUMINAL S.A) (ROMERO & ANDRADE) (CAMORRO)

1.2-Mantenimiento de las Vacunas

1.2.1-Características propias de la Vacuna

Las vacunas son productos biológicos termolábiles (sustancia inestable al calor) que se deben conservar entre +2° C y +8°C. Temperaturas superiores pueden producir pérdida de actividad (con efecto acumulativo), inactivación (irreversible), no cambios de aspecto (necesidad

de controles). Ciertas vacunas son más sensibles que otras al calor. Cuando la vacuna pierde su eficacia, el hecho de colocarla en la heladera o en el congelador no le devuelve su eficacia.

A una temperatura dada, la velocidad a la que cae la potencia depende en gran medida de factores tales como la naturaleza de los estabilizantes, las cepas de las vacunas, la humedad residual y la técnica de liofilización.

La pérdida de potencia es variable y depende de los umbrales de temperatura alcanzados así como el tiempo de permanencia de las vacunas expuestas a dichas temperaturas.

No hay datos concretos de estabilidad de las vacunas a temperatura ambiente y por periodos cortos de tiempo fuera de la heladera. En general, espacios cortos de tiempo, como puede requerir su transporte a los centros de vacunación no afecta sensiblemente al producto. Pero temperaturas inferiores provocan inactivación (congelación), floculación (DPT, antitetánica), sinmodificación del aspecto (antigripal, hepatitis B).

1.2.2-Mantenimiento de las Vacunas

Debemos ser conscientes de que el éxito de un programa de vacunación depende no sólo de que las vacunas hayan sido correctamente fabricadas sino de que se hayan respetado rigurosamente una serie de normas de almacenamiento, transporte y conservación, de forma que el producto llegue a su destino en condiciones óptimas. El inadecuado manejo de una partida de vacunas puede interferir gravemente o llevar al fracaso un programa de vacunación perfectamente bien diseñado, sea por la pérdida de eficacia de las vacunas administradas o incluso por un aumento de la reactogenicidad (Capacidad de los fármacos o vacunas para producir reacciones adversas) local producida por las vacunas conservadas en condiciones inadecuadas.

Como regla general, el rango ideal de temperatura para asegurar su perfecta conservación es entre 2° a 8°C, siendo por tanto imprescindible una correcta planificación logística de los

programas de inmunización, así como el adecuado mantenimiento de la cadena de frío durante todo el proceso. Dependiendo de su naturaleza pueden alterarse al ser expuestas a variaciones de la temperatura. La mayoría de las vacunas se inactivan por el calentamiento y otras por el contrario, se inactivan con la congelación (en general todas las que contengan adyuvantes) así como el cambio brusco de la temperatura. Las alteraciones y pérdidas de potencia inmunogénica de las vacunas por la exposición a temperaturas inadecuadas son acumulativas e irreversibles.

La potencia de una vacuna que se ha perdido por exposición al calor o al frío inadecuado no se recupera por almacenarla nuevamente a la temperatura correcta.

El único método que en la actualidad nos permite garantizar la inmunogenicidad y eficacia protectora de una vacuna desde su elaboración hasta su administración, es el mantenimiento de la cadena de frío.

(Gobierno Vasco) (Del Campo Alepuz)

1.2.3-Cadena de frío

La cadena de frío es el proceso de conservación, manejo y distribución de las vacunas. La finalidad de este proceso es asegurar que las vacunas sean conservadas debidamente dentro de rangos establecidos de temperatura, para que no pierdan su poder inmunogénico.

En el mantenimiento de la cadena de frío constituyen dos factores importantes la temperatura y el tiempo. A mayor aumento de uno y de otro, mayor deterioro del material biológico.

La potencia de una vacuna que se ha perdido por exposición al calor o al frío inadecuado no se recupera por almacenarla nuevamente a la temperatura correcta.

Asimismo, las características organolépticas de las vacunas no varían por los cambios de temperatura ni por colocarlas a temperaturas que no corresponden.

Lamentablemente no hay signos externos que adviertan sobre la pérdida de la potencia, solamente puede detectarse cuando ha existido congelación de los productos adsorbidos.

1.2.4-Elementos de la cadena de frío

Recursos humanos:

Todo individuo que interviene en el transporte, distribución, manipulación y administración.

Recursos materiales:

-Cadena fija: cámaras frigoríficas, frigoríficos y congeladores

-Cadena móvil: vehículos frigoríficos, heladeras portátiles, cajas isotérmicas....

En la actualidad disponemos en nuestro medio de los recursos necesarios para garantizar el adecuado mantenimiento de la cadena de frío. No obstante, la fase de almacenaje constituye el periodo más crítico por varios motivos, entre los que destaca, la escasez de espacio para el correcto almacenaje del pedido y la temperatura ambiente de la zona. Distintos estudios realizados recientemente en países industrializados evidencian serios problemas durante el almacenaje y manipulación en los puntos de vacunación, así como serias deficiencias en el nivel de formación del personal responsable de la cadena de frío.

Interrupción de la cadena del frío

No siempre resulta evidente la interrupción de la cadena del frío, por lo que es imprescindible:

“ Disponer de dispositivos automáticos electrónicos que por medios gráficos (u otros) acusen los accidentes sufridos por la instalación que puedan trascender en perjuicio de la calidad de las vacunas almacenadas.”

Medidas utilizadas para proteger las vacunas:

- Si la avería es de escasa duración, se mantendrá cerrada la puerta del heladera y se controlará posteriormente la temperatura y la estabilidad de dicha temperatura. En caso de corte de fluido eléctrico las heladeras están capacitadas para mantener su temperatura interna durante al menos 6 horas, siempre que se mantengan cerradas.
- Si se prevé una avería de larga duración, se trasladarán las vacunas con acumuladores de frío a una heladera bien controlada.

Caducidad

La fecha de caducidad será siempre el último día del mes indicado en el envase. Nunca utilizar vacunas caducadas.

1.2.5-Rangos de temperaturas de conservación de las vacunas

Las vacunas, como todas las sustancias biológicas, sufren deterioro y degradación acelerados por el calor, con la consiguiente destrucción del principio activo o antígeno inmunizante.

Esta degradación sucede como un proceso natural; sin embargo, existe la posibilidad de postergarla o detenerla, por un determinado tiempo, mediante la aplicación de frío.

Este es el motivo por el que las vacunas deben ser conservadas a temperaturas frías, estables y adecuadas al tipo de elementos constitutivos de cada una.

Existen dos rangos de temperaturas que interesan para las vacunas. Uno de estos rangos es el de refrigeración y el otro el de congelación.

Refrigeración: rango de temperatura entre 0° C y +8° C, preferentemente entre +2° C a +8° C, para evitar mantener la vacuna en el extremo inferior con peligro de congelación. Para control se prefiere mantener el refrigerador en +4° C.

Congelación: rango de temperatura entre 0° y -20° C. Cuando se desea mantener el biológico a esta temperatura durante el transporte, se utiliza hielo seco. Es el caso de una de las vacunas antivaricela. El hielo seco no debe ser utilizado para el transporte de ninguna otra vacuna.

En cada una de esos rangos de temperatura las vacunas podrán permanecer el tiempo especificado por el fabricante. El período de conservación depende de la composición de la vacuna y se basa en la curva de degradación del antígeno a través del tiempo.

a) Vacunas que pueden permanecer congeladas y refrigeradas

Todas las vacunas pueden ser refrigeradas, pero sólo algunas pueden ser, además, congeladas. Algunas vacunas sólo pueden mantenerse refrigeradas y nunca deben congelarse, porque cristalizan las sales de aluminio usadas como adyuvantes y destruyen el antígeno.

b) Vacunas que pueden permanecer congeladas y también refrigeradas

Las vacunas que pueden ser congeladas y refrigeradas son la antipolimiéltica oral, antisarampión, antiparotiditis, antirrubéola, triple viral y antiamarílica.

Las vacunas conservadas en congelación, una vez descongeladas, conviene mantenerlas a temperaturas de refrigeración y no volverlas nuevamente a -20° C.

Aunque es posible congelar y descongelar un número determinado de veces algunas vacunas, desde el punto de vista operativo se considera inconveniente la recongelación, porque el proceso destruye partículas de la sustancia biológica.

Un ejemplo es la vacuna antipolio oral, que puede ser congelada y descongelada hasta diez veces, siempre que la temperatura de descongelación no pase de los +8° C, ni los períodos de descongelación sumados superen las 24 horas. Esta medición es difícil de hacer en los niveles intermedios de la cadena, por lo que se aconseja no congelar una vez descongelada.

Las vacunas contra el sarampión, la parotiditis y la rubéola tienen una vida media de 2 años. El fabricante puede mantenerla a -20° C durante 1 año y, posteriormente a su entrega, los distribuidores pueden mantenerla 1 año más a temperatura de refrigeración.

c) Vacunas que sólo pueden permanecer refrigeradas

Las vacunas que sólo pueden ser refrigeradas son: DPT, antitetánica, antidiftérica (toxoides), antimeningocócica, antineumocócica, anti Hib, antihepatitis B, antihepatitis A, antipoliomielítica inyectable, antirrábicas, antitifoideas, anticoléricas, antiinfluenza.

En las vacunas antiinfluenza, antimeningocócica, antineumocócica y antirrábica la congelación desvirtúa el principio activo. En el caso de las adsorbidas a sales de aluminio o alumbre, el gel pierde su estructura coloidal y cristaliza, con lo cual pierde su actividad la vacuna y pueden producirse abscesos asépticos en el lugar de aplicación.

En los centros de vacunación las vacunas deben permanecer a temperatura estricta de refrigeración y no a temperaturas de congelación o superiores a los +8° C.

d) Conservación del diluyente

El diluyente debe ser colocado dentro de la heladera para mantenerlo frío; caso contrario, mantenerlo en lugar fresco y refrigerar antes de usar.

El diluyente no debe colocarse a temperatura de congelación porque estallan los frascos al aumentar el volumen del contenido en su paso de líquido a sólido.

Algunas consideraciones sobre Termoestabilidad

Las vacunas pueden diferenciarse por su termo estabilidad, teniendo en cuenta que no solo las temperaturas elevadas producen daño a las vacunas, también lo hacen las temperaturas inferiores a 0°C (punto de congelación).

Tolerancia de las vacunas a la temperatura

Vacunas atenuadas (virus vivos)

- toleran temperatura bajas.
- los virus atenuados se deterioran con gran rapidez cuando dejan de estar refrigerados.

Vacunas inactivadas o recombinantes y vacunas combinadas

- tolera períodos de tiempo con temperatura superior a 8 °C.
- el hidróxido aluminio como potenciador para producir la inmunidad protectora precipita a congelación.

Estas generalidades son de utilidad para la preparación de planes de contingencia, la organización interna de las vacunas dentro de los equipos frigoríficos y ante eventuales y breves excursiones térmicas.

Tabla 1-Tabla orientativa de termoestabilidad de las vacunas

Vacuna	Observaciones
Polio Oral (*)	Es una de las vacunas más sensible a temperaturas por sobre elevadas. Puede almacenarse congelada en niveles centrales.
Polio Inactivada (IPV) (*)	No debe exponerse a temperaturas inferiores de las recomendadas
BCG (*)	Su termoestabilidad depende mucho del laboratorio productor, en general es bastante estable-Liofilizada tolera temperaturas bajas. Usar hasta 6 hs después de reconstituida,
DTP/Hepatitis B	No debe exponerse a temperaturas inferiores de las recomendadas.
Haemophilus influenzae b	No debe exponerse a temperaturas inferiores de las recomendadas.
dTpa Td/T Hepatitis A (*)	No debe exponerse a temperaturas inferiores de las recomendadas.
Triple vírica (sarampión, rubeola , paperas) (*)	El liofilizado puede congelarse, NO el disolvente. Usar hasta 6 hs después de reconstituida,
Meningocócica conjugada (*)	No debe exponerse a temperaturas inferiores de las recomendadas.
Gripe	No debe exponerse a temperaturas inferiores de las recomendadas.
Varicela (*)	Liofilizada es estable a temperaturas bajas , si se ha almacenado congelada; No recongelar.
Neumocócica Polisacarida	No debe exponerse a temperaturas inferiores de las recomendadas.
Neumocócica Conjugada	No debe exponerse a temperaturas inferiores de las recomendadas.
Rabia (*)	Estable a temperaturas bajas
Fiebre amarilla (*)	Estable a temperaturas bajas. Usar hasta 6 hs después de reconstituida,
Rotavirus	Estable a temperaturas bajas
Papilomavirus (bivalente)	No debe exponerse a temperaturas inferiores de las recomendadas.

(Dra. Vizzotti & Lic. Aquino)

1.3-Recomendaciones para el almacenamiento de vacunas

-Las vacunas con fecha de caducidad más próxima, se colocarán de forma que estén más fácilmente accesibles que aquellas con fecha de caducidad posteriores. Retirar lo antes posible las vacunas caducadas para evitar su uso accidental.

-Las vacunas más sensibles al calor: polio oral, triple vírica, sarampión, rubéola, BCG y fiebre amarilla, deben estar situadas en la parte más fría de la heladera pero nunca en el congelador.

- Designar a una persona como responsable de las vacunas.

- Disponer de heladeras de 2°C a 8°C. El refrigerador o la heladera debe ser para uso exclusivo de las vacunas, estando prohibido guardar comidas, bebidas o muestras clínicas en el mismo.

- Colocar un termómetro de máximo y mínimo en la parte central de la heladera.

- El termómetro debe leerse muy fácilmente, debe ser de cuarzo líquido y que pueda ser adherido o colocado en la parte interna de la heladera y no en las puertas o laterales.

- La temperatura puede variar en las heladeras de 2 a 8 grados y debe ser controlada y registrada en planillas dos veces por día.

-Asegurarse que el suministro de electricidad del refrigerador y heladera no pueda ser interrumpido accidentalmente. Si se requiere descongelar para limpieza o se suspende el suministro de electricidad deben mantener los recaudos necesarios para no interrumpir la cadena de frío para las vacunas almacenadas.

-Para que el frío se mantenga en forma homogénea deben colocarse paquetes con hielo en el congelador y en la parte inferior botellas de agua para su enfriamiento, separadas por no menos de 2 cm para permitir la circulación de aire entre ellas y así mantener las vacunas a temperatura constante.

-Las heladeras deben estar ubicadas en lugares resguardados del sol y con separación de 15 cm de la pared.

-Los conservadores de vacunas o cajas isotérmicas para transporte deben ser sólidos, de cierre hermético, con lugar para colocar no sólo las vacunas para ser administradas durante el día, sino también para colocar paquetes de hielo en el piso, las paredes y la tapa.

-La ruptura de la cadena de frío reduce la potencia de las vacunas y contribuye al fracaso primario de la inmunización.

-El fracaso primario de una vacuna es la principal razón de padecer la enfermedad después de la vacunación.

-La potencia de una vacuna está fundamentalmente comprometida cuando ésta ha sido guardada por más de 8 meses.

-En la práctica, en general, es necesario realizar controles periódicos sobre el manejo de la cadena de frío para las vacunas.



ILUSTRACIÓN 1-TERMOMETRO DIGITAL Y REGISTRO DE TEMPERATURA

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN



ILUSTRACIÓN 2-HELADERA DE VACUNAS TIPICA

1.3.1-Valores comerciales de las Vacunas

Si bien las vacunas son provistas a los hospitales por las Subsecretaría de Salud de Neuquén y el Ministerio de Salud de la Nación, a través de diferentes planes, las misma no tiene un costo para los hospitales.

Sin embargo, las vacunas tienen un costo muy alto, a causa principalmente del costo de desarrollo de las mismas por la industria farmacéutica.

Se desarrollo un pequeña investigación de campo, consultado por el stock de vacunas que se encuentran en el vacunatorio del hospital, donde se almacenan y se aplican la mayor parte de la vacunas.

Solicitando la información del stock mensual, se tomaron los precios de venta de cada vacuna por separado, consultado a la revista electrónica KAIROS (<http://ar.kairosweb.com/>), se obtuvieron los siguientes resultados, mostrado en la siguiente Tabla 1

Según los cálculos obtenido, el costo total de las vacunas en las 3 heladeras del vacunatorio es aproximadamente en total \$ \$ 584,923.51 un valor mucho mas alto de lo que se esperaba, ya que se estimaba un valor de \$300.000.

Es un valor muy importante, con lo cual se justifica montar un sistema de monitoreo y alarma para las vacunas.

Tabla 2-Valores comerciales de las vacunas

PRECIO DE VACUNAS UTILIZADAS EN EL HOSPITAL, SI SE TUVIERAN QUE REPONER TODAS LAS VACUNAS EN CASO DE PERDIDA DE LA CADENA DE FRIO (*)							
CAJAS	DOSIS X CAJA	DOSIS	NOMBRE COMERCIAL	DETALLE	PRESENTACION	PRECIO POR CAJA	PRECIO
3	5	15	SYNAGIS	Palivizumab	100mg F.Amp. x 1	\$ 25,589.05	\$ 76,767.15
95	10	950	VACUNA BCG SSI	Vacuna antituberculosa	Caja Vial x 10 x 10 Dosis + disol	\$ 1,938.72	\$ 184,178.40
8	2	16	TUBERCULINA PPD RT23 SSI	Tuberculina	2 UT Caja Vial x 1 x 1.5ml	\$ 436.87	\$ 3,494.96
60	1	60	PREVENAR 13	Vacuna antineumocócica conjugada	Jer. Prell. x 1 x 0.5ml	\$ 456.86	\$ 27,411.60
50	1	50	BUSTRIX	Vacuna triple acelular	Jer. Prell	\$ 278.14	\$ 13,907.00
40	1	40	CERVARIX	Vacuna contra HPV tipos 16 y 18	Iny. Susp. x 0.5ml + Jer. Prell.x1	\$ 1,465.80	\$ 58,632.00
40	1	40	IMOVAXD DT ADULTO	Vacuna doble (DT)	Jer. Prell. x 1 Dosis x 0.5ml	\$ 46.96	\$ 1,878.40
45	1	45	ACTACEL	Vacuna cuádruple acelular	Vial x 1 Dosis x 0.5ml	\$ 599.17	\$ 26,962.65
45	1	45	TRIMOVAX	Vacuna trivalente viral	F.Amp. x 1 + Jer. Prell.	\$ 172.85	\$ 7,778.25
160	1	160	ISTIVAC JUNIOR	Inmunización antigripal	Niños Jer. Prell. x 0.25ml Trivalente	\$ 187.47	\$ 29,995.20
150	1	150	VAXIGRIP	Vacuna antigripal	Jer. Prell. x 1 x 0.5ml	\$ 177.28	\$ 26,592.00
80	10	800	PENTAXIM	Vacuna quintuple acelular	Fco.Vidrio Polvo + Susp. en Jer. Prell.	\$ 737.62	\$ 59,009.60
150	1	150	ENGERIX-B	Vacuna antihepatitis B	Jer. Prell. x 1ml	\$ 264.54	\$ 39,681.00
45	1	45	HAVRIX	Vacuna antihepatitis A	Jer. Prell. x 1	\$ 636.34	\$ 28,635.30
						TOTAL	\$ 584,923.51
						POR HELADERA TENIENDO EN CUENTA 3 HELADERAS	\$ 194,974.50

*- LA INFORMACION DE LOS PRECIOS INDIVIDUALES FUE OBTENIDA DE LA REVISTA ELECTRONICA "KAIROS" (<http://ar.kairosweb.com>)

Capítulo 2-Sistemas de control de temperatura

Sistemas de medición de temperatura

Para poder controlar la temperatura de las heladera utilizadas para conservación de vacunas, se utilizan una serie de elementos para evidencia la temperatura de la heladera, ya que ninguno de los elemetos controla el frio de las mismas

A continuación citamos los más utilizados:

2.1-Termómetros

2.1.1-Termómetros de Alcohol

Termómetro de alcohol: está construido de un pequeño tubo de vidrio en cuyo interior hay una columna de alcohol coloreado (rojo, azul, verde, etc.) sobre una escala, que permite ver la temperatura que marca el termómetro. Tiene la ventaja de ser de bajo costo. Se recomienda para el control de la temperatura del refrigerador.

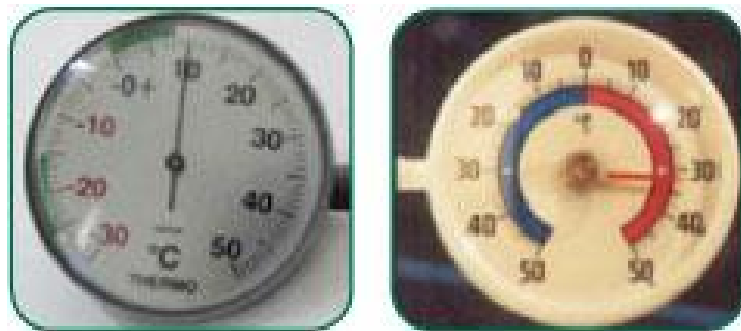
Barra negra significa “Zona de Seguridad” para vacuna bacteriana, es decir +2°C a +8°C (BCG).



ILUSTRACIÓN 3-TERMOMETRO DE ALCOHOL

2.1.2-Termómetro bimetálico

Es de forma circular y posee una aguja en el centro que se mueve a la izquierda o a la derecha dependiendo de la temperatura registrada por el sensor bimetálico que está en la parte posterior del termómetro. Su uso amerita calibración de manera que se asegure su funcionamiento adecuado. Se utiliza para monitorear la temperatura del refrigerador.



Termómetro bimetálico

ILUSTRACIÓN 4-TERMOMETRO BIMETALICO

2.1.3-Termómetro de máxima y mínima

Sirve para conocer el rango de temperatura en que trabaja el refrigerador, el cual debe mantenerse entre $+2^{\circ}\text{C}$ y $+8^{\circ}\text{C}$.

Debe colocarse en la pared posterior de primera bandeja del refrigerador, en un lugar seguro y accesible para su observación.

La lectura se realiza dos veces al día (mañana y tarde), sin sacarlo del interior del refrigerador.

Se lee donde marca el extremo inferior del testigo azul en las columnas de máxima y mínima. Posteriormente se juntan los testigos con la columna de mercurio, presionando el botón del termómetro o mediante un imán.

Registrar los valores en “Gráfica de Control de Temperatura”, con lápiz rojo la máxima y con azul la mínima.

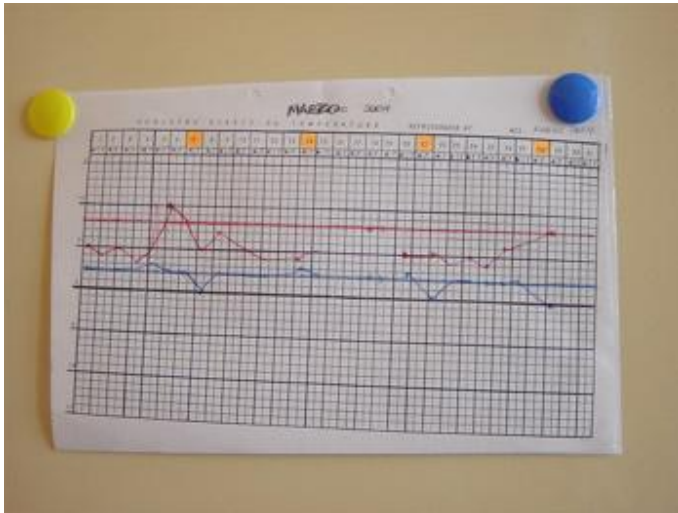


ILUSTRACIÓN 6-REGISTRO DE TEMPERATURA

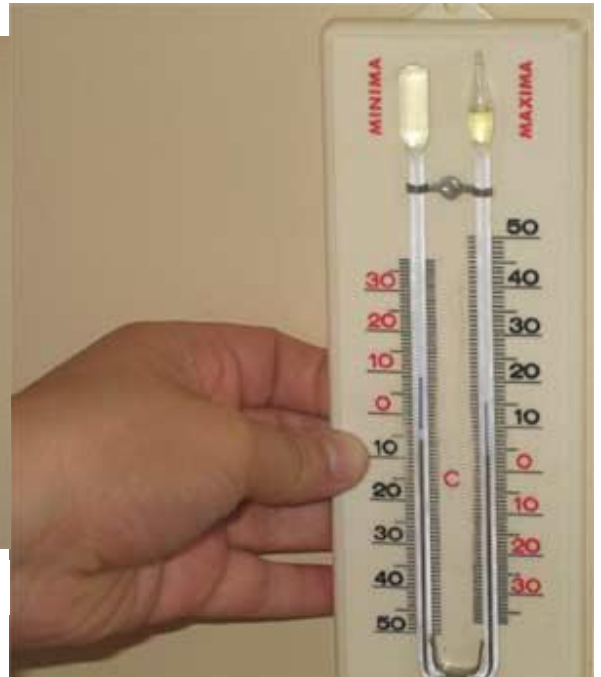


ILUSTRACIÓN 5-TERMOMETRO DE MAXIMA Y MINIMA

2.2-Sistemas Electronicos

2.2.1-Sistemas de alarma por temperaturas altas o bajas

Es un sistema que permite el monitoreo de la temperatura de un refrigerador o cuarto frío. Por medio de la alarma se pueden detectar tanto las temperaturas bajas como las altas, aunque haya energía eléctrica o no. El sistema cuenta con un retardo de tiempo para evitar falsas alarmas cuando se abre la puerta del refrigerador.

El sistema puede monitorear la temperatura hasta 72 horas después del corte de energía.



ILUSTRACIÓN 7-SISTEMAS DE ALARMA POR TEMPERATURAS ALTAS O BAJAS

2.2.2-Sistema de alarma con marcador telefónico

Este sistema tiene las mismas características que el anterior, pero su sistema de alarma es mediante un marcador telefónico que puede programarse hasta con cinco teléfonos. En el momento en que la alarma avisa por motivo de la temperatura, el marcador telefónico llama a cada uno de los cinco teléfonos programados, hasta que uno de ellos conteste. Tienen mensaje de voz y también pregrabado.

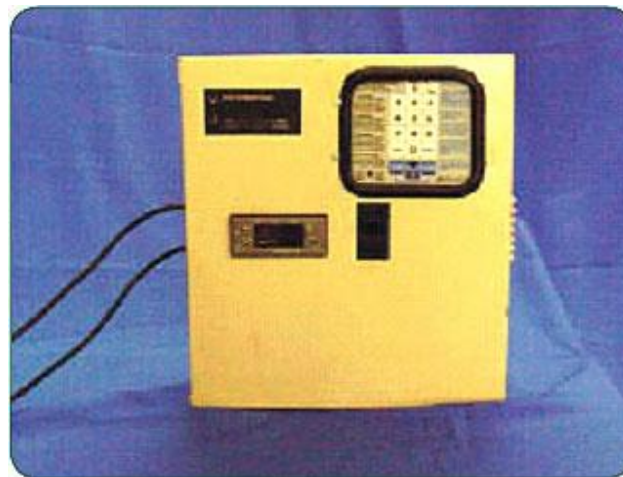


ILUSTRACIÓN 8-SISTEMA DE ALARMA CON MARCADOR TELEFÓNICO

Vemos a continuación un producto similar al sistema de alarma con marcador telefónico

((OPS))

2.2.3-Centinela Alarma por temperatura

Centinela es un equipo que permite monitorear en tiempo real la temperatura y marcación telefónica a fijo o celular activada por alarma.

En la actualidad los sistemas de refrigeración cuentan con alarmas sonoras que permiten avisar de fallas de forma local y presencial l corriéndose el riesgo del deterioro de los productos almacenados, cuando no se encuentre en ese lugar personal pendiente del sistema de refrigeración.



ILUSTRACIÓN 9-CENTINELA ALARMA POR TEMPERATURA

Centinela nace de la necesidad de resolver esta limitación, informando de forma instantánea, remota y durante las 24 horas del día de los cambios drásticos de temperatura, por vía telefónica sin importar ubicación y distancia del personal

Centinela es un sistema de control que permite alta confiabilidad en la refrigeración de cualquier producto (alimentos, vacunas, flores etc) que implique gran precisión en las temperaturas necesarias para su conservación, al posibilitar correcciones a tiempo por parte del personal especializado, encargado de su mantenimiento.

- Monitoreo y control del estado de los compresores, ventiladores, ciclo de deshielo y alarmas, en línea desde una consola digital fácil de utilizar.
- Consola digital de administración local, remota (computador conectado Internet /celular con plan de datos).
- Alarma: telefónica , visual y virtuales (correo electrónico)
- Generador de informes (Gráficos, texto)

Es indispensable tener un computador encendido conectado al Centinela, durante el tiempo que se desee monitorear el sistema de refrigeración. Si el sistema de refrigeración se desea monitorear remotamente desde un computador a través de internet o un celular con plan de datos con acceso a internet, el modulo local necesitara tener una conexión a internet para poder transmitir la información

(DARTICO)

En el mercado recientemente se encontró un sistema silmilar al propuesto, llamado

2.2.4-Datalogger Ethernet de Temperatura A205

Monitor de temperatura con conexión Lan



ILUSTRACIÓN 10-DATALOGGER ETHERNET DE TEMPERATURA A205

El Accsense A2-05 del monitor de temperatura con conexión de cable Lan está diseñado para medir la temperatura de los refrigeradores, congeladores, médicos de almacenamiento criogénico e Incubadoras.

Dispone de conexiones para dos sondas RTD externa y un sensor termopar. En el caso de una pérdida de la conexión de red que puede almacenar hasta 256 puntos de datos, o hasta que se restaura la conexión.

Todas las medidas adquiridas utilizando Accsensen están disponibles de inmediato en su cuenta de Accsense, online. Actualización en tiempo real, puede trazar la historia de la medición, análisis de datos tabulares, y establecer alarmas para ver sus datos y proporcionar correo electrónico, buscapersonas, teléfonos celulares o alertas

Características:

- Rápido y fácil de configurar.
- Cifra todos los datos, completamente seguro.
- Envía datos a servidores seguros Accsense.
- Mezcle y combine con monitores inalámbricos Accsense.
- Conexión Ethernet estándar a Internet.
- Varios monitores se pueden ver en una sola cuenta.
- Soporta Power over Ethernet

Características del software:

- Fácil de usar, prácticamente ninguna curva de aprendizaje.
- Basado en la Web, sin instalación de software o molestias configuración del firewall.

- Interfaz del usuario personalizable con el panel de control de la mayoría de las mediciones recientes de todos los sensores.
- Acceso instantáneo a los gráficos que representan la historia de medida, con rangos tan angosto como 5 minutos o tan ancho como 90 días.
- Escala vertical manual permite una consistencia a través de múltiples gráficos y ejemplos.
- Configuración de alarmas a través del teléfono celular de correo electrónico, buscapersonas o teléfono fijo. Incluso envía la misma alarma en varias ubicaciones!
- Tabla de datos de medición en línea
- Descarga de datos de medición para análisis fuera de línea
- Permite a través contraseña-protegida, el acceso limitado a ciertos usuarios
- Completamente seguro y una copia de seguridad en Rackspace™
- Actualizaciones software y de las funciones gratis

Capacidades de mediciones:

- Velocidad de datos de almacenamiento: 30 segundos a 24 horas (cambios en la tasa de verificación de la alarma durante una condición de alarma)
 - Comprobación de Alarma: 30 segundos a 24 horas (independientemente del tipo de almacenamiento)
 - Almacenamiento local (fuera de Internet): 256 puntos
- Sondas (2 x RTD) para / Heladera / Congelador Incubadora
- Tipo de 100 ohmios Platino alfa = 0,00385 (PT100)
 - Rango de temperatura: -100 A 100 °

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

-Precisión sobre Rango completo: ± 1 ° C (excluyendo Precisión IDT)

SONDA (1 x termopar) para criogénicos

-Tipo T

-Rango de temperatura: -200 A 400 °

- Precisión sobre Rango completo ± 1 ° C (excluyendo Precisión TC)

Condiciones de Operacion

-Rango de temperatura: -20 A +70 °

-Almacenamiento: -40 A +85 ° C

-Humedad (sin condensación) 5% a 95%

Bateria Interna

-Autonomía (se recarga automáticamente): 4 horas (carga completa)

Precio aproximado de un sistema completo:

-Datalogger Ethernet de Temperatura A2-05: USD 779.00

-Sonda de temperatura RTD PT100: USD 89

-Certificacion de Calibracion NIST: USD 45

-Servicio de monitoreo Anual: USD 69

-Suma total : USD982 convirtiendo a la moneda Pesos tomando el valor de dólar USD 1 = \$5, el costo total del sistema es \$ 4910, nos servirá como referencia en el análisis de costos

(CAS_Data_Logger) (Control_Solutions_Inc)

Otro sistema similar es el siguiente.

2.2.5-Registrador de Temperatura con soporte Web TandD TR-71W

El Registrador de Temperatura con soporte Web utiliza su red LAN para almacenar las mediciones de temperatura de las sondas de los sondas externos. Este registrador cuenta con múltiples ajustes de la alarma de alta y baja con las notificaciones enviadas a su correo electrónico, 8K de capacidad de almacenamiento de datos, pantalla LCD, 2 sondas de temperatura exterior y 15 opciones de intervalo de grabación. Estos registradores se pueden conectar a través de una red LAN por cable o una red LAN inalámbrica.

Se incluye con cada Registrador de Temperatura Web:

(2) TPE-Resina Sensores blindados (# TR-0106)

(1) Batería de litio CR-2032 (# HRB-TEMP-5)

2 metros de cable LAN

Adaptador AC (# AD0605)

TR-7W de software para PC en CD



ILUSTRACIÓN 11-REGISTRADOR DE TEMPERATURA CON SOPORTE WEB

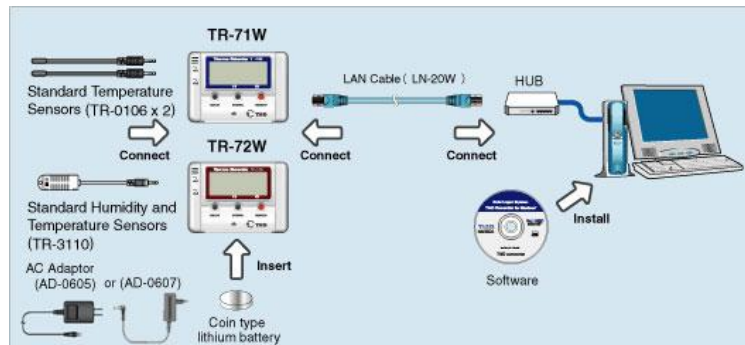


ILUSTRACIÓN 12-INTERCONEXION REGISTRADOR DE TEMPERATURA CON SOPORTE WEB

Registrador de Temperatura con soporte Web

Características

- Navegador Web basado en gestión de registrador
- Notificaciones de alarma enviadas al correo electrónico o teléfono móvil
- Compatible con redes LAN inalámbricas
- Incluye manual del software, la batería y el usuario

Especificaciones

- Canales de Medición: 2 canales de temperatura (sensor externo)
- Sensor externo: -40°C a 110°C
- Sensor opcional: -60°C a 155°C * 1

Precisión de las mediciones

- Promedio de $\pm 0,3^{\circ}\text{C}$ (-20°C a 80°C)
- Media $\pm 0,5^{\circ}\text{C}$ (-40°C a $-20 / 80$ a 110°C) * 2
- Resolución: $0,1^{\circ}\text{C}$

-Sensores conectados: TR-0106: TPE blindados en resina sensores x 2

Intervalo de grabación : 1.2.5.10.15.20.30 Segundos, minutos 1.2.5.10.15.20.30.60 / total de 15 opciones

Capacidad de almacenamiento de datos: 8.000 lecturas x 2 canales

Método de grabación: Endless (GrabacionCircular): Sobrescribir de los datos más antiguos cuando se graba

One-Time Method: Detiene la grabación cuando la capacidad está llena

Pantalla LCD: Las mediciones, (CH1, CH2, o alternado), estado de grabación, advertencia de la batería, cantidad de datos registrados, unidad de medida

Fuente de alimentación: Adaptador de AC especial (DC5V)

Los datos de respaldo: Se almacenan por cerca de 3 meses con una batería de litio (CR-2032)

Interfaz: Wired LAN : 10/100 BASE-TX conector RJ45 LAN inalámbrica : IEEE 802.11b CF de tipo Wireless LAN Card

Dimensiones : 55mm x 78mm x 39mm

Peso : Cerca de 101g (incluyendo una batería)

Entorno de funcionamiento: Temperatura : 0 a 50 ° C Humedad : 20 a 80% HR (sin condensación)

Funciones del navegador de datos:

Ver datos de temperatura y humedad en forma de gráfico, inicie la grabación, descarga de datos, monitorear Lecturas actuales, Check back-up de la batería

Función de advertencia E-mail: Registra hasta 5 direcciones de correo electrónico para recibir

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

Consumo de energía: Cerca de 290mA (cuando se utiliza con cable LAN)

Seguridad Acceso controlado Dirección IP, autenticación de usuario Contraseña

Compatible con OS Windows 2000/XP/Vista

Precio Aproximado:

TandD Web Enabled 2 Channel Temperature Logger TR-71W: U\$D 539

TandD TPE Resin-Shielded Temperature Sensor (0.6 Meter Cable Length and 15mm Probe Length) TR-0106 U\$D 26

Calibración NIST: u\$49

(Loggershop)

Capítulo 3-Desarrollo del sistema

En este capítulo se presenta el desarrollo del sistema Scada para monitorizar la temperatura de las heladeras de vacunas, utilizando una metodología de desarrollo para sistemas empotrados (embedded systems).

Esta metodología plantea el diseño del software y del hardware en las siguientes fases:

3.1-Diseño del sistema propuesto

De la información antes expuesta, el sistema de propuesto se desarrollara integrando los proyectos desarrollados en la Cátedra de Técnicas Digitales III “Control de Temperatura Microprocesador para Heladeras de Vacunas” del año 2010 y de Cátedra Tratamiento de la información en Sistemas de Control y Supervisión “Tutorial de instalación y manejo de sistema Mango M2M”

Establecemos los requisitos que deberá tener el sistema, a partir de la experiencia recabada en el Hospital Zona Cutral-Có Plaza Huincol y de los proyectos anteriores

3.1.1-Requisitos

El sistema propuesto tendrá las siguientes características:

- Utilizar un Microcontrolador con posibilidad de conexión Ethernet.
- Medir la temperatura entre 2 y 8 °C, con 4 sensores distribuidos estratégicamente en la bandeja utilizada para el almacenamiento de las vacunas.
- Mostrar en un display la temperatura y por medio de unos Led indicadores el estado de la cadena de frío.

- Realizar la monitorización de la temperatura de la heladera de vacunas, en forma local y mostrar los valores por medio de un página web, por medio del SCADA Mango M2M.
- Ser de fácil uso para el personal técnico y usuario.
- Debe ser un dispositivo que se puede adaptar a diferentes tipos de heladeras, las cuales en su mayoría son heladeras domesticas sin requisitos especiales.
- Posibilidad de utilizar la red ethernet que está disponible en el hospital para intercambiar la información. En los lugares donde no se dispone de bocas adicionales se evaluara si es necesario agregar hub switch en algunos lugares, ya que la cantidad de bocas de red esta limitada.
- Posibilidad de registrar la temperatura exterior de la sala donde se encuentra la heladera, dado que una calefacción elevada, influye en el rendimiento de la misma.
- Ser independientes del servidor de SCADA, en caso de falla, continua registrando.
- Detectar una puerta mal cerrada.
- Avisar con una alarma sonora en caso disminución o aumento de de temperatura excesivo, o puerta de heladera mal cerrada y corte de suministro eléctrico.
- Sistema de batería, para una autonomía superior a 8hr, dado que ante un corte de luz, las heladeras no soportan más de ese tiempo, la temperatura de las vacunas.
- Utilizar protocolo de comunicación modbus tcp,
- El tipo de protección del gabinete IP (IP54, etc), dado que en le proceso de desinfección, puede entrar humedad al sistema.

3.1.2-Alternativas al diseño:

-Independizarnos del SCADA, de forma que el Microcontrolador sea también servidor de una pagina web.

-Conexión Ethernet por WIFI.

-Evaluar un diseño paralelo con comunicación sobre red telefónica existente

-Analizar las vibraciones del motor de la heladera y consumo eléctrico de la misma, de forma de utilizar estos datos para realizar un diagnostico preventivo en caso de falla.(ver si es viable y si es de ayuda)

-Detectar un gradiente temperatura anormal de la calibración de la heladera, para detectar un posible fallo antes que ocurra.

3.2- Funcionamiento propuesto del sistema

El funcionamiento del “Sistema Scada para monitorizar Heladera de vacunas del Hospital y los centros de salud” será el siguiente.

La Heladera en la cual están las vacunas, se colocaran 2 sensores de temperatura (LM35 inicialmente y luego de opto por DS18B20), 1 en cada bandeja de en la que se colocan las vacunas, estos sensores medirán la temperatura y la transforma en valor de tensión.

Estas tensiones son medidas por puerto seleccionado como entrada analógica del Arduino(LM35) o por medio de un puerto digital(DS18B20) que se ubica en un gabinete fuera del la Heladera. El sistema dispone de un interruptor Reed Switch normar cerrado, ubicado en la en el exterior de puerta, que indica si la puerta de la heladera quedo mal cerrada.

El Arduino toma los datos de temperatura y tiempo, los prepara y los enviar por medio de puerto Ethernet, que conecta a la Red LAN del hospital

Los datos son recibidos por el Servidor SCADA Mango, el cual almacena es su base de datos, preparados para mostrarlos cuando algún usuario lo requiera, o cuando los valores produzcan un evento de alarma.

Las temperatura pueden ser supervisadas en la PC del Vacunatorio, simultáneamente de todas la heladeras.

Los eventos de alarmar inicialmente serán mostradas en el LCD, y en forma sonora por medio de Buzzer del Arduino y notificadas a la PC de Vacunatorio en caso de que no sea reconocida localmente esta serán enviados por e-mail o sms desde el servidor al Smartphone de la Supervisora de Enfermería.

En el caso de las Heladeras de los Centros de Salud, la comunicación entre el Arduino y el Servidor Servidor SCADA Mango se realiza Via Internet.

Los datos pueden presentarse en forma histórica, para evaluar las temperaturas entre las cuales están almacenadas la vacunas. También se puede presentar un informe de las alarmas que se generaron.

En el grafico siguiente vemos los canales de comunicación entre los elementos del Sistema.

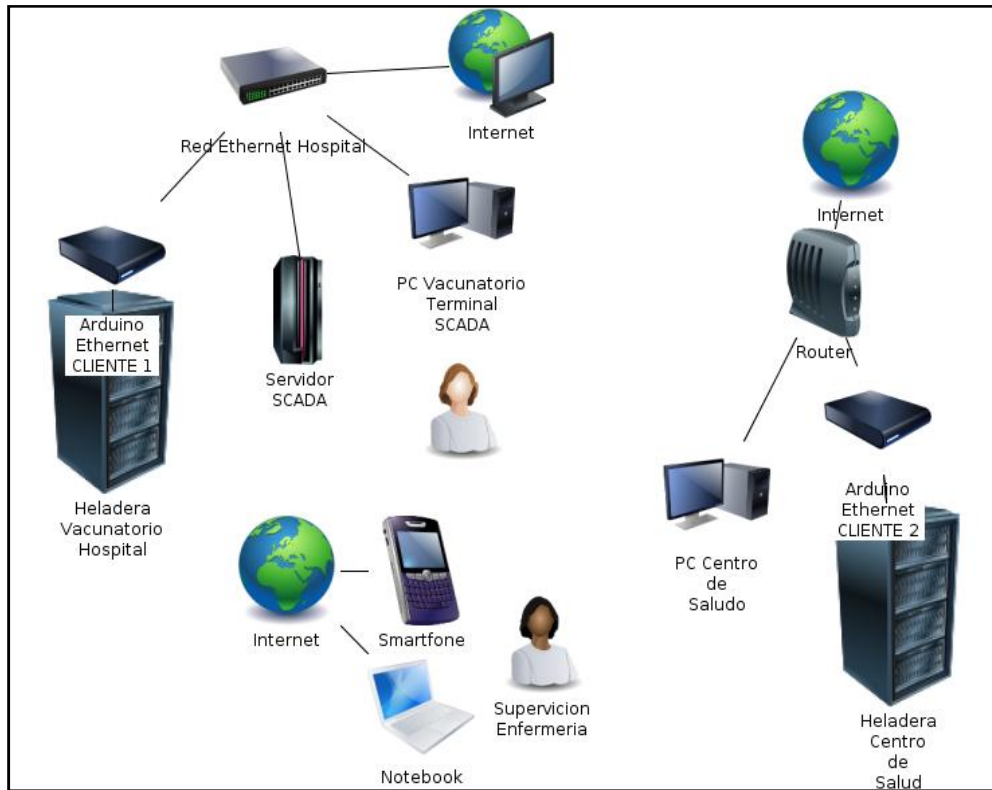


ILUSTRACIÓN 13-CANALES DE COMUNICACION DEL SISTEMA SCADA PROPUESTO

Otra variante al diseño del sistema es utilizar el mismo Arduino como servidor WEB sin necesidad de una Pc, utilizando un memoria Micro SD para almacenar los datos de temperatura , tiempo y alarmas.

Permitirá graficar y mostrar los datos, para ello utiliza el Arduinio para mandara una pagina con los datos, ya que el renderizado de la pagina Web sera realizada en la PC.

Si bien la funcionalidad se verá limitada, es una variante que deber ser desarrollada, para poder evaluar el limite del Arduino UNO.

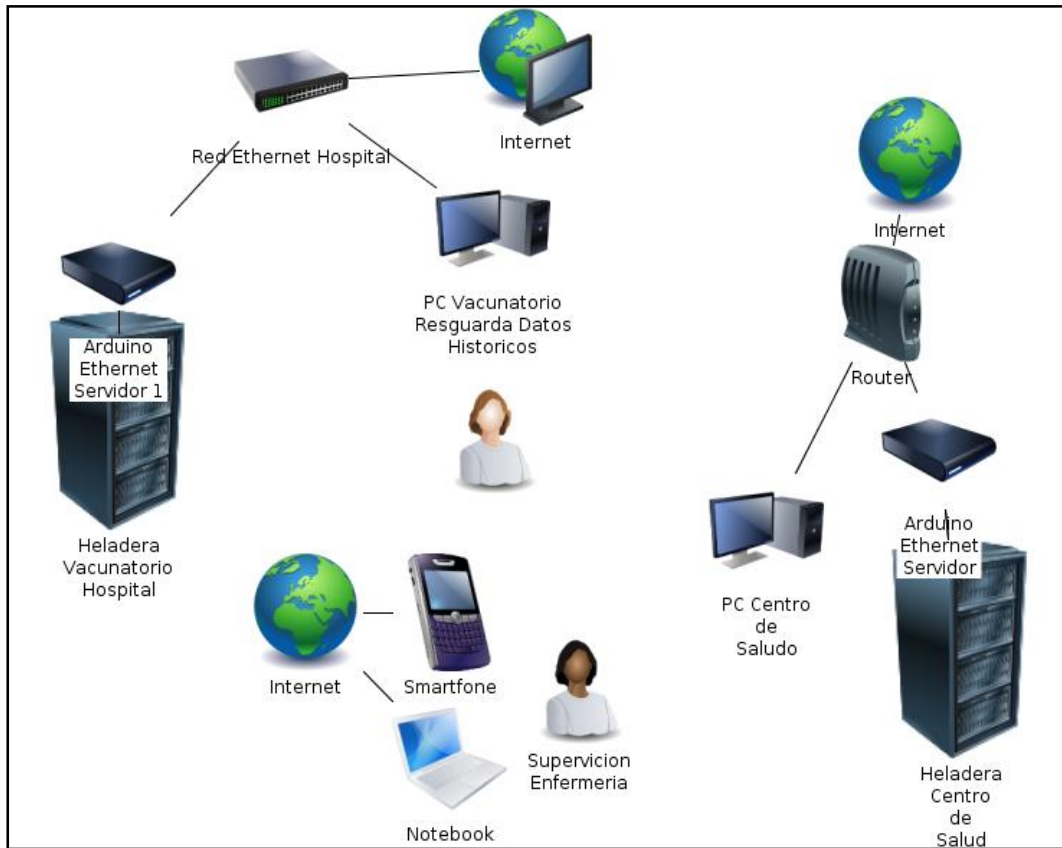


ILUSTRACIÓN 14-CANALES DE COMUNICACION DEL SISTEMA PROPUESTO SIN SCADA

3.3- Plataforma de desarrollo de hardware

Al plantear el diseño del sistema, en el proyecto anterior de “Control de Temperatura Microprocesador para Heladeras de Vacunas”, se armó el sistema alrededor de un Microcontrolador 16F878A de Microchip, al sumarse ahora al sistema la posibilidad de conexión Ethernet, sumo una complejidad al desarrollo, con lo cual se optó por utilizar plataformas de hardware ya montadas, las cuales son utilizadas para realizar desarrollo de aplicaciones electrónica en forma rápida, con una determinada cantidad de entradas salidas, y puertos de comunicación.

Dentro de las plataformas desarrollo se busco un microcontrolador con conexion ethernet, por lo menos con 4 entradas analógicas y 4 digitales, de fácil adquisición, y de un valor menor a USD 100, y de fácil adquisición en Argentina

Selección de las plataformas:

3.3.1-WIZ220IO

Del fabricante WIZnet, un módulo de control y monitorización I/O con Ethernet basado en servidor web. Ideal para aplicaciones con contadores ('metering'), el WIZ220IO se caracteriza por ocho puertos de entrada digital, ocho puertos de salida digital, dos puertos de entrada analógica (resolución de 12 bit) y dos puertos de salida analógica.



ILUSTRACIÓN 15-WIZ220IO

Este módulo compacto (63 x 45 mm) de monitorización remota, que se puede aplicar sin la necesidad de desarrollos adicionales, ofrece soporte para programa de aplicación y servidor web.

Las características se completa con un interface de red 10BaseT/100BaseTX con 'auto-negociación' (full-dúplex y half-dúplex) y auto MDIX, compatibilidad con los protocolos ARP, ICMP, TCP, DHCP, PPPoE y HTTP, y alimentación DC de 5 V.

Webserver based IO control module

Key Features

- Remote I/O Monitoring and Control with Ethernet
- 8 Digital Input Ports
- 8 Digital Output Ports
- 2 Analog Input Ports (12 bit resolution)
- 2 Analog Output Ports (12 bit resolution)
- Supports Application Program
- Supports WebServer

Tabla 3-Especificaciones WIZ220IO

Network Interface	10BaseT/100BaseTX Auto Negotiation (Full-duplex and half-duplex) Auto MDIX
Digital Input	Input Port 8 ports Input Type Voltage Low level input voltage -0.5V ~ 0.8V High level input voltage 2.0V ~ 5.5V
Digital Output	Output Port 8 ports Output Type Voltage Low level output voltage MAX 0.4V High level output voltage MIN 2.4V
Analog Input	Input Port 2 ports Resolution 12-bit Input Type Voltage

	Sampling Rate 100ksps Input Range 0 ~ 5V
Analog Output	Output Port 2 ports Resolution 12-bit Output Type Voltage Setting Time 4.5us Output Range 0 ~ 5V
Protocol	ARP, ICMP, TCP, DHCP, PPPoE, HTTP
Dimensions	63 x 45 (mm)
Power	DC 5V

(CASADOMO)

Distribuidor en Argentina:

<http://www.semak.com.ar/>

Cuando se lo consulto, no tenían stock y no sabían cuando iban a recibir de nuevo otra alternativa era comprarlo en Chile por medio de:

<http://www.olimex.cl>

a un precio de USD\$47.25 + IVA

Ventajas:

-No se programa, solo recibe y responde datos por medio paquetes TCP socket, casi similar a la trama de datos modbus.

Desventajas

- No hay muchos desarrollos bien documentados.

3.3.2 PIC-MINI-WEB

Un servidor de red que cabe en la palma de tu mano. Esta placa de desarrollo tiene todo lo que necesitas para servir páginas web con un megabit de almacenamiento en la placa. Recuerda que necesitarás la placa adaptador ICSP nombrada en la lista de abajo para unir un programador olimex

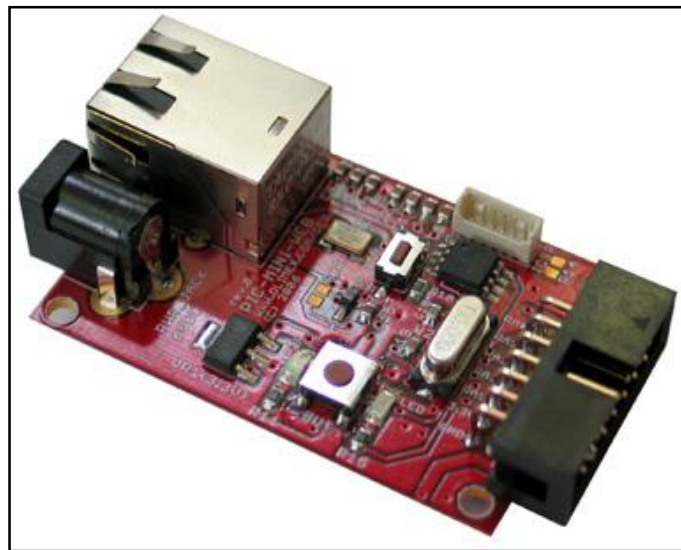


ILUSTRACIÓN 16-PIC-MINI-WEB

Especificaciones:

- Microcontrolador PIC18F25J10
- Controlador Ethernet ENC28J60
- 1 Mbit de memoria flash(serial) de almacenamiento para páginas web
- Conector mini ICSP/ICD para programación y depuración
- Botón de reset
- Boton de evento de usuario (User event)

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

- servidor web completo y soporte para stack TCP-IP de acuerdo al stack TCP-IP de código abierto de Microchip's
- Conector jack hembra para fuente de poder de + 5VDC
- Regulador de voltaje +3.3V y condensadores de filtrado
- Leds de estado
- conector header de extensión para conexión con otras placas
- Dimensiones 55x36mm (2.16x1.42")

Distribuidor en Argentina:

No se encontro ninguno, otra alternativa era comprarlo en chile por medio de:

<http://www.olimex.cl>

a un precio de USD\$42.04 + IVA

Ventajas:

- Tecnología pic, la cual ya estamos acostumbrados.
- Se carga la pagina en un formato propio del stack Microchip TCP/IP Stack Application Note AN833

Desventajas:

-Si queremos modificar el stack, es muy complejo.

-El chip tiene limitaciones en escritura, ya que es un chip de producción y no de desarrollo

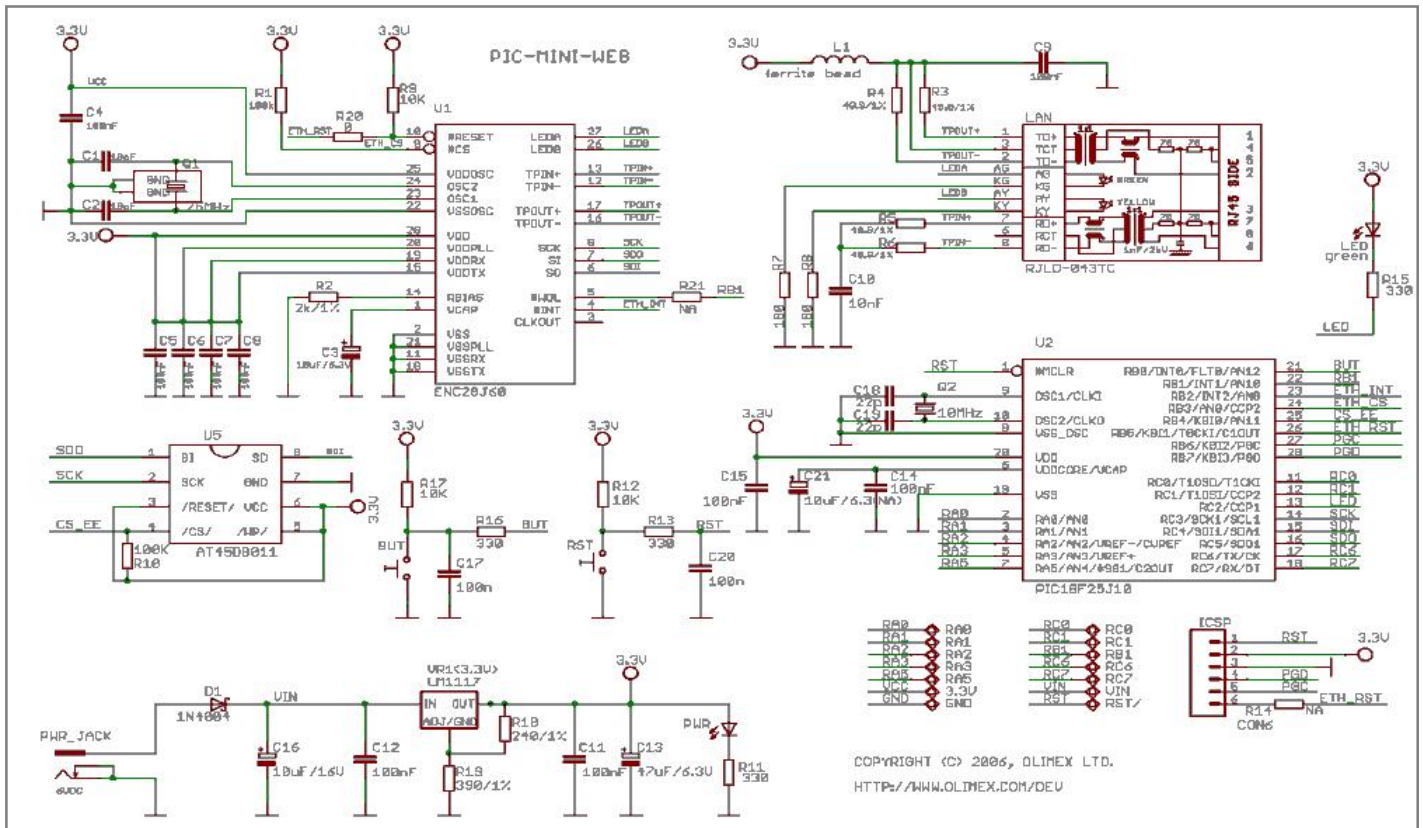


ILUSTRACIÓN 17-DIAGRAMA CIURCUITAL PIC-MINI-WEB

(Electronics)

3.3.3- Mbed - LPC1768 Development Board

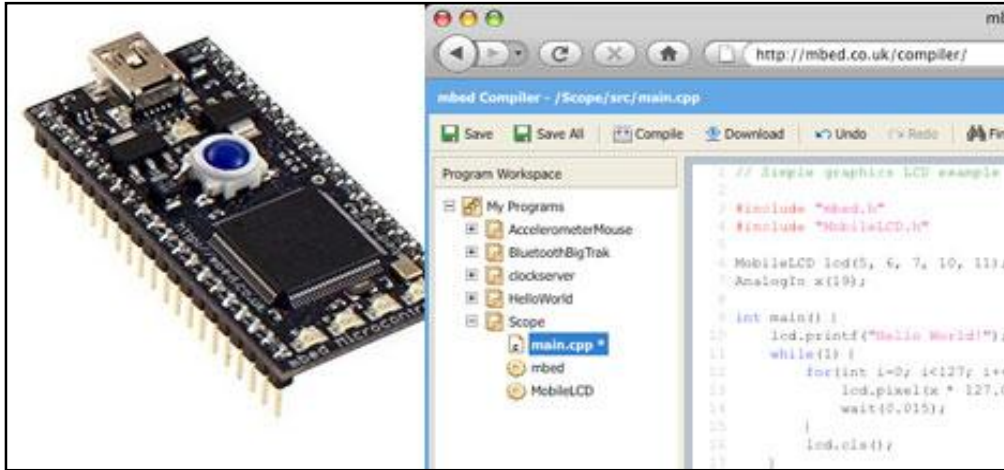


ILUSTRACIÓN 18-MBED - LPC1768 DEVELOPMENT BOARD

El Microcontrolador mbed es un procesador ARM, con una completa serie de periféricos, programación USB e interfaz de comunicación que viene en un paquete DIP práctico y pequeño. El mbed es una herramienta para la creación rápida de prototipos y es súper fácil de usar.

El Microcontrolador mbed fue diseñado para la creación de prototipos rápidos, y viene en un formato DIP 40-pin con paso de 2.54 mm, por lo que es ideal para experimentar con protoboards, stripboards y PCB. Soporta muchos tipos de interfaz, entre ellas Ethernet, puerto serial, USB, SPI y I2C CAN.

La descarga de programas es tan simple como conectar el dispositivo a tu computador, este lo reconocerá como memoria USB, copias el programa y el mismo queda cargado

Funciona en Mac, Windows y Linux gracias a que posee un compilador online, el cual permite realizar programas en C++. Existen también librerías de mbed disponibles para sacar el máximo provecho al microcontrolador.

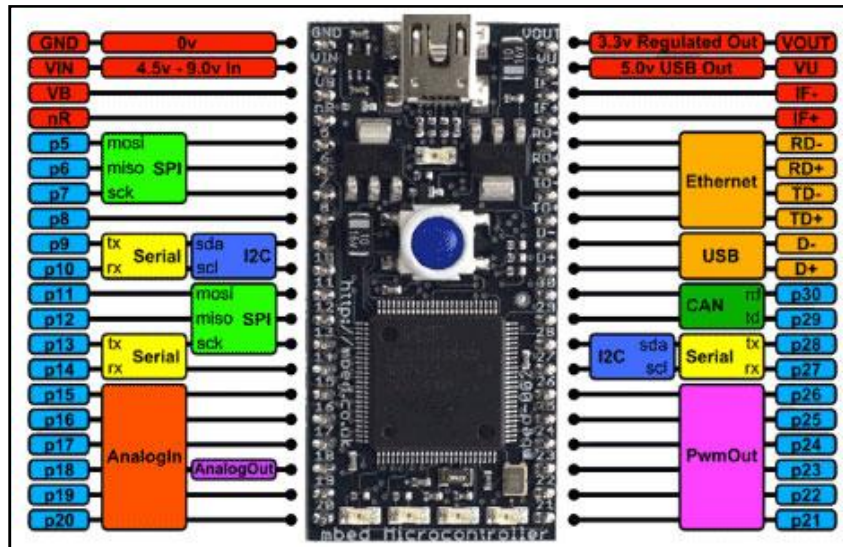


ILUSTRACIÓN 19-MBED - LPC1768 DISTRIBUCION DE PINES

Mbed ARM Cortex-M3

Características

- Microcontrolador NXP LPC1768
- ARM® Cortex™-M3 de alto desempeño
- 6MHz, 32KB RAM, 512KB FLASH
- Ethernet, USB Host/Device, 2xSPI, 2xI2C, 3xUART, CAN, 6xPWM, 6xADC, E/S de proposito general (GPIO)
- Factor de forma p/prototipado
- Empaque DIP 40-pin 0.1" (breadboard), 54x26mm
- Alimentación voltaje 5V USB o 4.5-9V
- Programador flash incorporado tipo “arrastra y suelta” USB (como un pen drive USB)

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

-Sitio web para el desarrollador: mbed.org

-Compilador Online “liviano”

-Kit de desarrollo de software (SDK) de alto nivel en C/C++

-Librerías y proyectos publicados como referencia inicial

Ventajas

-Ideal para iniciarse con microcontroladores ARM de 32 bits y lenguajes C/C++

-No se necesita un programador adicional, el hardware está listo para usarse

-EDI y nuestros proyectos disponibles online en cualquier parte (acceso a internet)

-Bueno para prototipos y para experimentar

Desventajas

Costo

Mbed Cortex-M3: \$us120.- (aprox.) en Latinoamérica

Mbed Cortex-M0: \$us90.- (aprox.) en Latinoamérica

No cuenta con interfaz JTAG para depuración

Se programa únicamente en forma ONLINE.

Distribuidor en Argentina:

SLAB (<http://www.slabs.com.ar>)

Cuando se les consulto no tenían stock, estaban esperando para hacer un pedido gran, pero no nos pudieron determina la fecha.

Precio en el exterior: €40

3.3.4-Arduino

Es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.2 3

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque (boot loader) que corre en la placa.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software del ordenador (por ejemplo: Macromedia Flash, Processing, Max/MSP, Pure Data). Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente.



ILUSTRACIÓN 20-ARDUINO UNO GENUINO

Características técnicas del Duemilanove ATmega328

(el más popular y flexible hoy en día)

- Microcontrolador: ATmega328
- Tensión de operación: 5V
- Tensión de entrada (recomendado): 7-12V
- Tensión de entrada (límite): 6-20V
- Digital I/O Pins: 14 (de los cuales 6 proveen salida PWM)
- Pins de entrada analógica: 6
- Corriente continua por pin I/O: 40 mA
- Corriente continua para el pin en 3.3V: 50 mA
- Memoria flash: 32 KB de los cuales 2 KB son usados por el bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Frecuencia de reloj: 16 MHz

Shields Ethernet

Los Shields son placas que se colocan encima de la placa Arduino y que amplían una nueva función para que sea controlada desde Arduino, para controlar diferentes aparatos, adquirir datos, etc.

La Arduino Ethernet Shield permite a una placa Arduino conectarse a internet usando la librería Ethernet. Para usar la Ethernet Shield solo hay que montarla sobre la placa Arduino (p. e. una Diecimila). Para cargar los sketches a la placa conectarla al ordenador mediante el cable

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno:Chandia Pablo Lucas

UTN-FRN

USB como se hace normalmente. Una vez que el sketch ha sido cargado se puede desconectar la placa del ordenador y alimentarla desde una fuente externa.

Conectar la Ethernet Shield a un ordenador, a un switch o a un enrutador utilizando un cable ethernet standard (CAT5 o CAT6 con conectores RJ45). La conexión al ordenador puede requerir el uso de un cable cruzado (aunque muchos ordenadores actuales, incluyendo los últimos modelos Mac pueden hacer el cruce de forma interna).



ILUSTRACIÓN 21-ETHERNET SHIELD V1.0 PARA ARDUINO (MICROCHIP'S ENC28J60 SPI)



ILUSTRACIÓN 22-ETHERNET SHIELD W/MICROSD CARD READER (WIZNET W5100 ETHERNET CONTROLLER)

Ventajas

-Asequible - Las placas Arduino son más asequibles comparadas con otras plataformas de microcontroladores.

-Multi-Plataforma - El software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux. La mayoría de los entornos para microcontroladores están limitados a Windows.

-Entorno de programación simple y directo - El entorno de programación de Arduino es fácil de usar para principiantes y lo suficientemente flexible para los usuarios avanzados. Pensando en los profesores, Arduino está basado en el entorno de programación de Processing con lo que el estudiante que aprenda a programar en este entorno se sentirá familiarizado con el entorno de desarrollo Arduino.

-Software ampliable y de código abierto- El software Arduino esta publicado bajo una licencia libre y preparado para 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. De igual modo se puede añadir directamente código en AVR C en tus programas si así lo deseas.

-Hardware ampliable y de Código abierto - Arduino está basado en los microcontroladores ATMEGA168, ATMEGA328 y 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

Desventajas

-Dependiendo de la cantidad de entrada y salidas, al colocar shield, pude limitar la cantidad de las misma, ya que las usa para comunicarse con las mismas.

-Ya que es una plataforma de desarrollo, el consumo puede ser mayor a otro tipo de placas similares, se complica si el desarrollo implica bajo consumo.

-La administración de las memorias.

La Memorias Flash, donde se almacena los skeches (programa), para el ATmega168 es de 16k y con el bootloader se pierden 2k y con el ATmega328 es de 32k se pierden 0,5 k, Por lo que debemos desarrollar los programas de forma muy optimizada, usando los tipos de variables que menos memoria requieran, en la medida de lo posible. También debemos optimizar el código fuente de la aplicación para no repetir líneas de código.

La memoria SRAM (Static Random Access Memory ó memoria estática de acceso aleatorio) es de tipo volátil, es el espacio donde los sketches (programas) almacenan y manipulan variables al ejecutarse. La información guardada en esta memoria será eliminada cuando Arduino pierda la alimentación. Esta memoria es de uso exclusivo para el programa en ejecución.

La memoria SRAM de Arduino es muy pequeña, por lo que debemos optimizar nuestros programas al máximo y no abusar de variables de tipo char muy grandes. Hay que tener en cuenta que cada carácter de una variable char utiliza un byte. En el microcontrolador ATmega 168 el tamaño de la memoria SRAM es de 1024 bytes, para el caso de un chip ATmega328 (como el que incorpora Arduino UNO) el tamaño es de 2KB (2048 bytes).

Si la SRAM se queda sin espacio, el programa de Arduino fallará de forma imprevista, aunque se compile y se suba a Arduino correctamente la aplicación no se ejecutará o se ejecutara de manera extraña.

Distribuidores en Argentina.

ARGEDUINO (<http://www.argeduino.com.ar/>)

DITENTEC(<http://www.ditentec.com.ar/>)

ELECTROSHIP(<http://electroship.com/>)

El Arduino es de mas fácil adquisición anteriores plataforma,

Se pudo adquirir por medio de ELECTROSHIP(2012):

1 Arduino UNO	\$145 actualmente \$ 270
---------------	--------------------------

1 Shield Ethernet Arduino V1.0	\$120 actualmente \$ 170
--------------------------------	--------------------------

Por medio de ARGEDUINO (2012)

1 Arduino Uno + Cable:	\$ 160 actualmente \$300
------------------------	--------------------------

1 Ethernet Shield Micro-sd W 5100 :	\$ 195 actualmente \$320
-------------------------------------	--------------------------

Todos los valores estan expresados en pesos

3.4-Selección de plataforma y experiencia en la compra

Del análisis de las diferentes plataformas, el principal problema no fue las limitaciones de hardware ni las del entorno de desarrollo en si, sino la adquisición de la misma, principalmente por problemas de importación y la aduana.

Se opto por la plataforma Arduino, principalmente por dos razones:

- Mucha información de librerías y proyectos de ejemplo.
- Mas facil de adquirir.

En la primera adquisición de la placa Arduino UNO y Ethernet Arduino V1.0, se adquirió por medio de ELECTROSHIP, que en realidad era un gestor de compra, no maneja stock, realizaba la compra por EBAY en Hong Kong, y se encargaban de enviarlo a domicilio como regalo para no pagar impuesto, el problema surgió en la Aduana de Neuquen, el envío fue retenido, por lo cual tuve que viajar a Neuquen, para ver porque razón fue retenido.

En la Aduana me pidieron la explicación de que era lo que estaba importando y que tenía que demostrar el costo del mismo. Por lo cual fui a cyber he imprimí un comprobante de la compra, por lo cual me liberaron el envío y lo entregaron.

En la segunda adquisición por medio de ARGEDUINO, fue más fácil, ya que el proveedor tenía en stock.

3.4.1-Modularidad del Sistema

El hecho de utilizar un Shield Ethernet y un Arduino, nos abre la posibilidad de realizar un sistema modular, que es una ventaja de la Plataforma Arduino, a tal punto muchas otras plataforma se han adaptados para utilizar el mismo formato de placa (chipKIT, PICAXE-28X2, PIC.Duino Net, Bambino 210, FRDM-KL02Z) compartiendo puertos i2c, alimentación, entradas analógicas y digitales, salidas digitales, entre el Arduino y las diferentes Shields.

Al tener un sistema modular, en caso de falla de alguna, no se tiene que cambiar una placa entera, sino el Shield o Arduino que se daño, con lo cual se puede dar una solución rápida en caso de falla.. El diseño propuesto es a partir de un Arduino Uno y un Shield Ethernet, al cual se le sumara un Shield LCD con Pulsadores.

Al utilizar Shields, se reducen los tiempos de desarrollo de los proyectos, si bien son plataformas de desarrollo de hardware, según se planteo en el proyecto, los mismos se utilizaran para realizar el producto final, de forma de ahorra tiempo en el desarrollo del hardware.

La cantidad de Shield que se puede ir sumando depende principalmente de varios factores:

- Que los Shield no compartan pines de comunicación o salida y entradas, ya que por hardware no pueden ser modificadas, solo algunos por software.
- Que por la disposición de elemento no se produzcan cortocircuitos entre las placas.
- La cantidad de corriente que puede manejar el regulador de tensión del Arduino, puesto que la tensión de los Shield es provista por este regulador.

3.4.2- Selección del Shield Ethernet

Se propuso para realizar la conexión a la red Ethernet utilizar un shield, para poder dotar la capacidad modular al sistema, de forma de que ante un fallo, se puedan cambiar los módulos en forma rápida y fácil.

Dentro de las opciones de analizaron los dos siguientes módulos shields:

3.4.3- Ethernet Shield V1.0 basado en ENC28J60

El módulo de interfaz Ethernet con ENC28J60 permite conectar a internet cualquier microcontrolador con interfaz SPI y con la capacidad de memoria RAM y Flash para ejecutar un stack TCP/IP embebido (Microchip TCP/IP, uIP, etc). La tarjeta incluye todo lo necesario para realizar la interfaz con una red ethernet, incluido el conector ethernet con transformadores de aislamiento. Solamente hace falta conectarla a los pines SPI del microcontrolador y a la alimentación.

Esta tarjeta se puede utilizar con un microcontrolador PIC que sea capaz de ejecutar el Stack TCP/IP de Microchip o con tarjetas Arduino.

Características del ENC28J60 Módulo Interfaz Ethernet

- Basado en el controlador Ethernet ENC28J60 de Microchip.
- Compatible con redes 10/100 Mbps, velocidad de operación 10 Mbps
- Regulador de 3.3 volts incluido en la placa
- Se puede usar fácilmente con Arduino, AVR, PIC, MSP430, etc.
- Compatible con microcontroladores de 3.3 (también se puede comunicar con CPUs de 5 volts).
- Interfaz Ethernet 802.3 10 Mbps Half o Full Duplex (Manual).
- Interfaz con el microcontrolador host por SPI.
- Soportado por el Stack TCP/IP de Microchip Technology.

-Cristal de 25 Mhz, Conector Ethernet y todo el hardware de soporte incluido

-Aplicaciones en: Servidores web embebidos, control de iluminación, convertidores de protocolos, demótica y automatización, proyectos escolares.

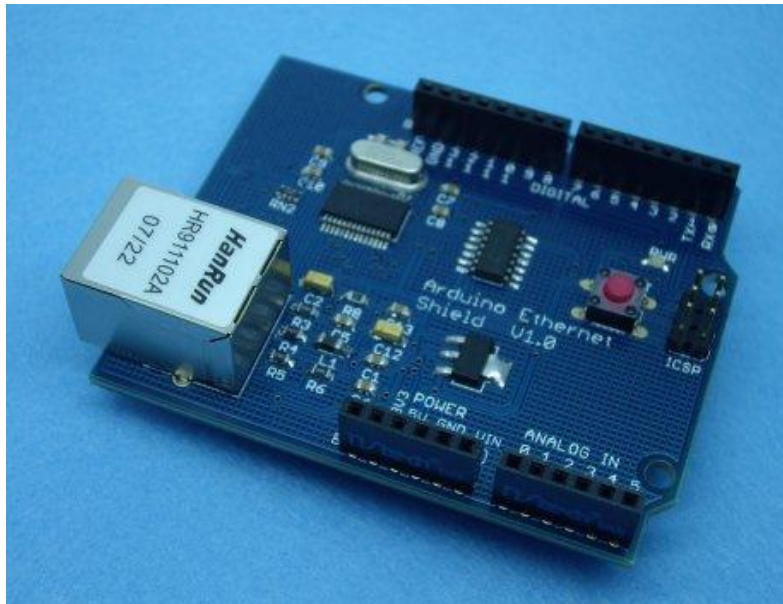


ILUSTRACIÓN 23-ARDUINO ETHERNET SHIELD V1.0

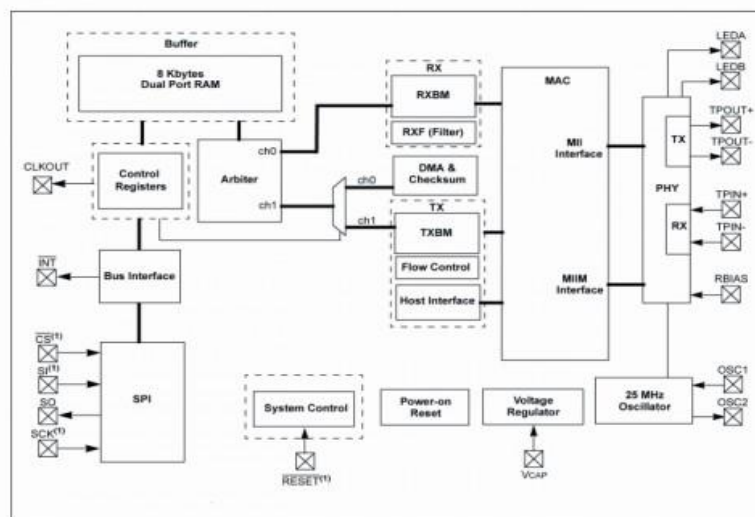


ILUSTRACIÓN 24-BLOQUES DEL ENC28J60

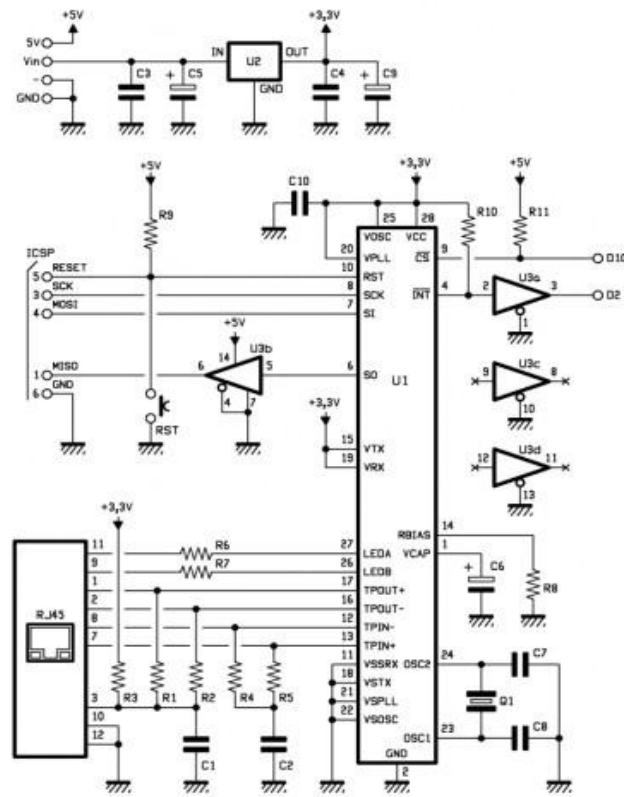


ILUSTRACIÓN 25-CIRCUITO DE ETHERNET SHIELDS V1.0

3.4.3-Ethernet Shield Con Wiznet W5100

Este shield permite conectar Arduino a Internet. Está basado en el chip de ethernet Wiznet W5100 con funcionalidades de IP tanto para TCP como UDP. El Arduino Ethernet Shield soporta hasta 4 conexiones simultáneas. Utiliza la librería Ethernet para escribir rápidamente programas que se conecten a Internet empleando este shield.

Características:

- Shield para Arduino basada en el controlador de Ethernet W5100 de Wiznet
- Buffer de memoria interna de 16 kB
- 10 baseT/100BaseTX

- Stack de red (IP) con capacidad para TCP y UDP
- Soporta hasta 4 sockets independientes y simultáneos
- Posee jack RJ-45 estándar y transformador de línea integrado
- Conector para memoria microSD (Memoria no incluida) para almacenar y servir archivos en la red
- Habilitado para PoE (Power Over Ethernet = Alimentación del cable de red) (Módulo no incluido)
- Botón de reset que resetea tanto la shield como la board Arduino
- Estándar Arduino Pinout 1.0

Una gran ventaja de este shield es que es apilable por lo que podrás disponer de todos sus pines en otros shields.

Esta última versión incluye un slot para tarjetas micro-SD, el cual puede ser empleado para almacenar archivos que podrás poner disponibles a través de la red. Es compatible con el Arduino UNO y el Mega (empleando la librería que viene en la versión de software Arduino 0019).

El Arduino UNO utiliza los pines digitales 11, 12 y 13 (SPI) para comunicarse con este shield. El Mega emplea los pines 50,51 y 52. En ambas tarjetas (UNO y Mega) el pin 10 es empleado para seleccionar el W5100 y el pin 4 para la tarjeta SD. Mientras emplees las funcionalidades de ethernet estos pines no estarán disponibles. Así mismo toma en cuenta que en el Arduino Mega a pesar de que el pin 53 (SS pin) no es empleado para seleccionar ni el W5100 ni la tarjeta debe ser configurado como una salida ya de otra manera la interfaz SPI no funcionará.

Debido a que el W5100 y la tarjeta SD comparten el mismo bus SPI solo uno podrá estar activo a la vez. Si se estás usando ambos periféricos en el programa, se debe tomar en cuenta

esto. Si no estas empleando alguno de estos periféricos debes “deseleccionarlo” explícitamente. Para deshabilitar la tarjeta SD, configura el pin 4 como salida y escríbele HIGH. Para deshabilitar el W5100 configura el pin 10 como una salida en HIGH.

El shield posee un conector RJ45 estándar para ethernet. El botón de reset inicializa tanto el shield como el Arduino.

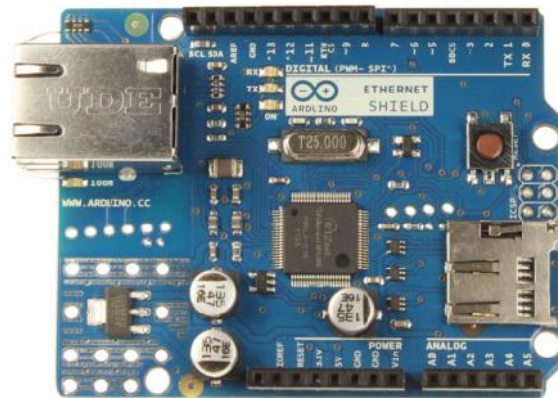


ILUSTRACIÓN 26-ETHERNET SHIELD CON WIZNET W5100

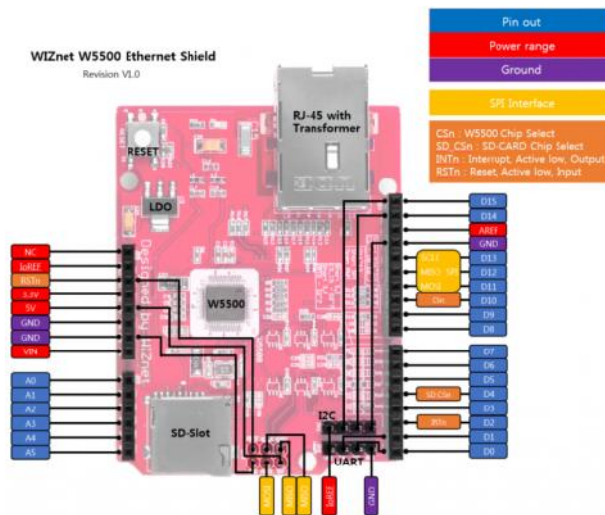


ILUSTRACIÓN 27-PINES DE I/O ETHERNET SHIELD WIZNET W5100

3.5- Sensores de Temperatura

Elección del Sensor de Temperatura:

Se planteo utilizar unos sensores de temperatura, que tengan un rango de temperatura de 0 a 40°C con una precisión de +/-0,5°C, la cual es suficiente para nuestra aplicación, dentro de la gamma de sensores que se puede adquirir fácilmente encontramos los siguientes:

-LM35, Circuito Integrado de National Semiconductor, el cual entrega una tensión proporcional con la temperatura en °C, con un rango de -55 a +150 °C con una precisión +/- 0,25°C , utilizando un puerto analógico del Arduino para leer esta tensión y por soft convertirlo en el valor correspondiente.

-DS18B20, Termómetro Digital de Maxim, el cual entrega una lectura digital de temperatura °C, con un rango de -10 a +85°C con una precisión de +/- 0,5°C, un puerto digital del Arduino para leer el valor digital.

3.5.1-Sensor de Temperatura LM35

Información Ampliada obtenida de Data Sheet LM35

Sensores de Precisión de Temperatura en Centígrados

Descripción general

La serie LM35 son sensores de temperatura de circuitos integrados de precisión, cuya tensión de salida es linealmente proporcional a la temperatura Celsius (centígrados). Así pues, el LM35 tiene una ventaja sobre los sensores de temperatura lineales calibradas en ° Kelvin, ya que el usuario no está obligado a restar una gran tensión constante de su producción para obtener escalamiento centígrados conveniente. El LM35 no requiere ninguna calibración externa o recorte para proporcionar precisiones típicas de $\pm 1/4^{\circ}\text{C}$ a temperatura ambiente y $\pm 3/4^{\circ}\text{C}$ sobre un -55 a $+150^{\circ}\text{C}$ rango de temperatura. De bajo coste está asegurada por el recorte y la calibración a nivel del mar. La impedancia de salida baja del LM35, salida lineal, y la calibración precisa inherente hacen interfaz con lectura de salida o circuito de control especialmente fácil. Se puede usar con fuentes de alimentación individuales, o con más y menos suministros. Ya que consume $60\ \mu\text{A}$ de su fuente, que tiene muy baja auto-calefacción, menos de $0,1^{\circ}\text{C}$ en aire en reposo. El LM35 está calificada para operar en un rango de temperatura de -55 a $+150^{\circ}\text{C}$ mientras que el LM35C está pensado para una -40° a $+110^{\circ}\text{C}$ rango (-10° con una mayor precisión). La serie LM35 está disponible encapsulado hermetico TO-46 de transistores, mientras que el LM35C, LM35CA, y LM35D también están disponibles en el plástico TO-92 encapsulado de transistor.

Características

-Calibrado directamente en ° Celsius (centígrados)

-Factor de escala Lineal $+10,0\ \text{mV} / ^{\circ}\text{C}$

$-0,5^{\circ}\text{C}$ exactitud garantizable (a $+25^{\circ}\text{C}$)

- Nominal para la gama plena -55° a $+150^{\circ}$ C
- Adecuado para aplicaciones remotas
- Bajo coste debido a la oblea de nivel de recorte
- Opera desde 4 a 30 voltios
- Menos de 60 uA de fuente actual
- Baja auto-calefacción, 0.08° C el aire en calma
- Linealidad típica solamente $\pm 1/4^{\circ}$ C
- Salida de baja impedancia, 0,1 de carga de 1 mA

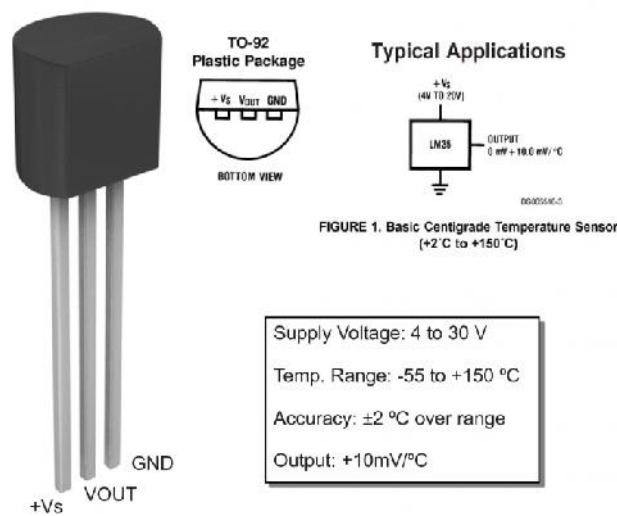


ILUSTRACIÓN 28-SENSOR LM35

3.5.2-Sensor de Temperatura DS18B20

Información Ampliada obtenida de Data Sheet DS18B20

El termómetro digital DS18B20 proporciona "9 bits a 12 bits" mediciones de temperatura Celsius y tiene una función de alarma con puntos disparo no volátiles superior e inferior programables por el usuario. El DS18B20 comunica a través de un bus 1-Wire, que por definición requiere una sola línea de datos (y tierra) para la comunicación con un microprocesador central. Además, el DS18B20 puede derivar energía directamente de la línea de datos ("alimentación parásita"), eliminando la necesidad de una fuente de alimentación externa.

Cada DS18B20 tiene un código de 64 bits de serie único, que permite que múltiples DS18B20s funcionen en el mismo bus 1-Wire. Por lo tanto, es fácil de usar un microprocesador para controlar muchos DS18B20 distribuidos sobre un área grande. Las aplicaciones que pueden beneficiarse de esta característica incluyen controles de HVAC ambientales, sistemas de monitoreo de temperatura dentro de los edificios, equipos o maquinaria, y la supervisión de procesos y sistemas de control.

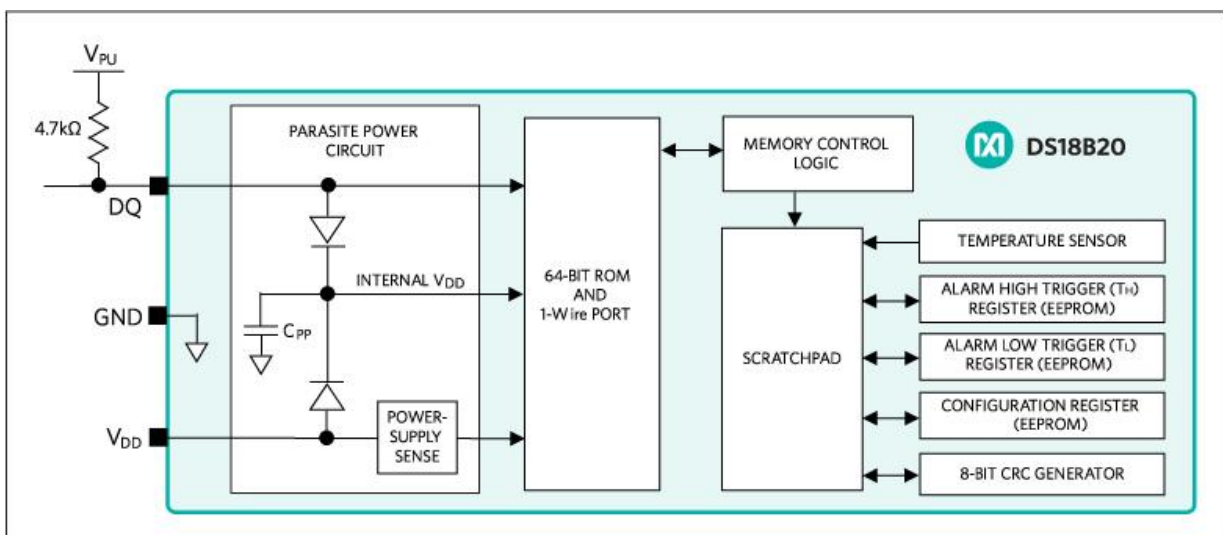


ILUSTRACIÓN 31-DIAGRAMA DE BLOQUES DE DS18B20

Características principales

- Interfaz única de 1 Wire®, requiere sólo un puerto Pin para la Comunicación
- Componente reducir el conde con sensor de temperatura integrado y EEPROM
- Mide temperaturas desde -55°C a $+125^{\circ}\text{C}$ (-67°F a $+257^{\circ}\text{F}$)
- $\pm 0,5^{\circ}\text{C}$ Precisión de -10°C a $+85^{\circ}\text{C}$
- Resolución de 9 bits programable de 12 bits
- Sin componentes externos requeridos
- Modo de energía parasitaria requiere sólo 2 botones para el funcionamiento (DQ y GND)
- Simplifica las aplicaciones distribuidas de detección de temperatura con capacidad multipunto
- Cada dispositivo tiene un código de serie único de 64 bits almacenado en la ROM a Bordo
- Flexible Configuración de alarma con alarma comando de búsqueda definible por el usuario no volátil (NV) identifica los dispositivos con temperaturas fuera de los límites programados
- Disponible en encapsulados 8-pin SO (150 milésimas de pulgada), 8-Pin μSOP , y 3-Pin TO-92

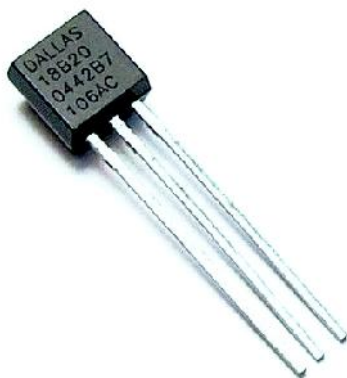


ILUSTRACIÓN 32-DALLAS DS18B20

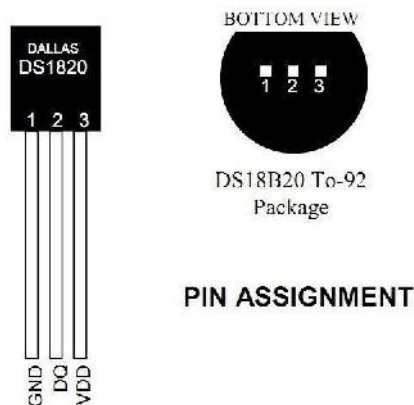


ILUSTRACIÓN 33-DS18B20 PINES

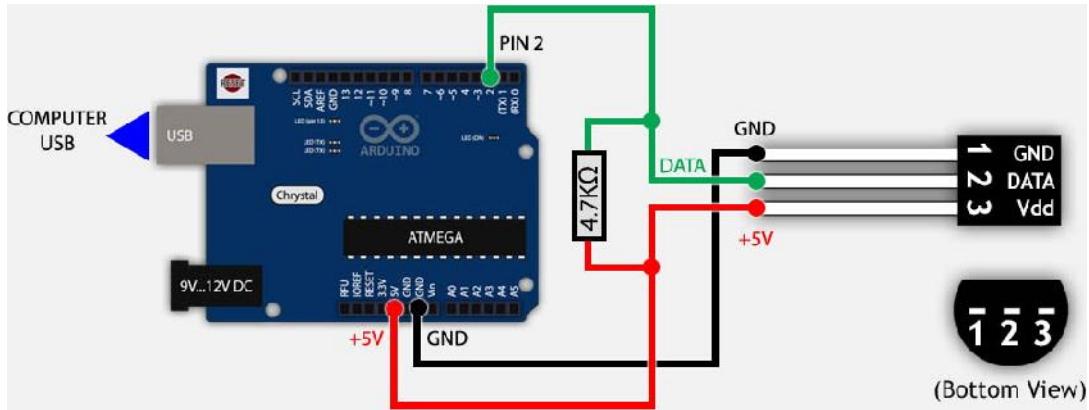


ILUSTRACIÓN 34-ARDUINO + DS18B20

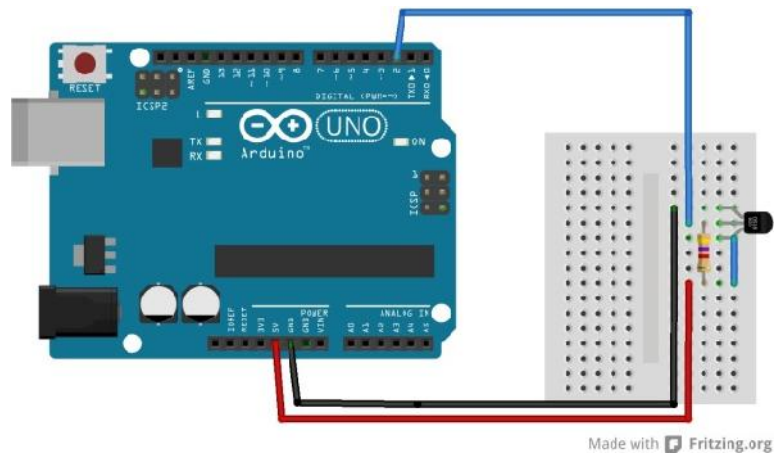


ILUSTRACIÓN 35-ARDUINO + DS18B20 MONTADO EN PROTOBOARD

3.6- Selección del Display LCD

El display a utilizar será un LCD de 16 x 2 caracteres compatible con Hitachi HD44780, dado que suficiente para mostrar la información necesaria para el proyecto.

Los display LCD1602A, dispone de un bus de datos de 8 y 4 bits, y de alimentan con 5V, a continuación describimos la función de cada pin de conexión:

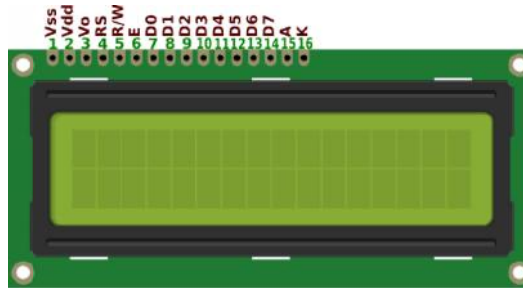


ILUSTRACIÓN 36-LCD LCD1602A

VSS que es el pin de negativo o masa o 0 volts o GND.

VDD es la alimentación principal de la pantalla y el chip, lleva 5 voltios.

VO es el contraste de la pantalla, debe conectarse con un potenciómetro de unos 10k ohms o una resistencia fija una vez que encontremos el valor deseado de contraste.

RS es el selector de registro (el microcontrolador le comunica a la LCD si quiere mostrar caracteres o si lo que quiere es enviar comandos de control, como cambiar posición del cursor o borrar la pantalla, por ejemplo).

RW es el pin que comanda la lectura/escritura. En nuestro caso siempre estará en 0 (conectado a GND) para que escriba en todo momento.

E es enable, habilita la pantalla para recibir información.

D0 a D7 bus de datos de 8 bits sirven para establecer las líneas de comunicación por donde se transfieren los datos.

D4 a D7 bus de datos de 4 bits.

A y K son los pines del led de la luz de fondo de la pantalla. A se conectará a 4 o 5 volts y K a gnd.

El costo del mismo es de alrededor de \$100.

Al conectar el display LCD con un bus de datos de 4 bit al Arduino UNO la conexión puede ser la siguiente

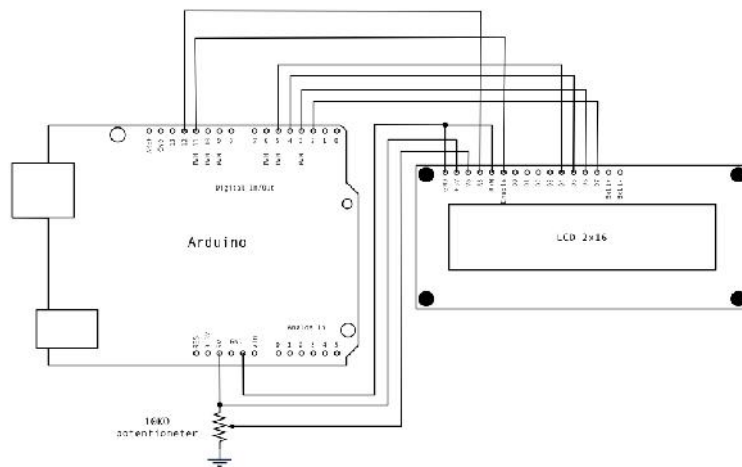


ILUSTRACIÓN 37-DIAGRAMA DE CONEXION ARDUINO CON LCD 16X2

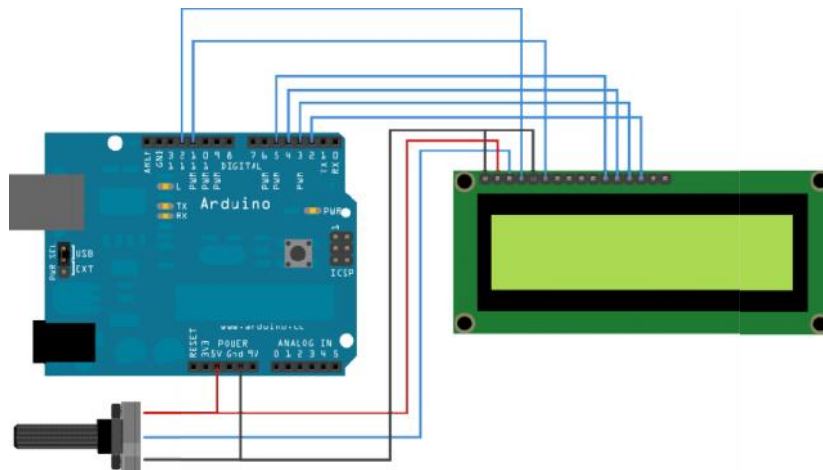


ILUSTRACIÓN 38-ILUSTRACIÓN 37-CONEXION REAL ARDUINO CON LCD 16X2

Podemos observar que utiliza 6 salidas digitales del Arduino Uno.

Para poder manejar este display de puede utilizar la librería “LiquidCrystal.h”, en el entorno de programación de Arduino.

Investigando se encontró que este Display también está disponible en formato de Shield para Arduino, con el agregado de que dispone de un teclado de 6 botones. El mismo tiene el nombre de LCD Keypad Shield, se adquirió uno por un valor de \$170.

3.6.1-LCD Keypad Shield para Arduino

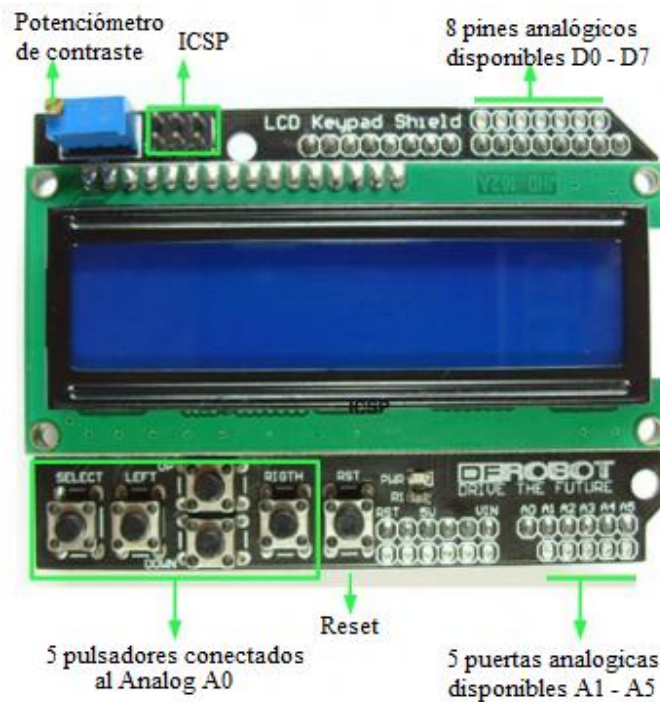


ILUSTRACIÓN 39-LCD KEYPAD SHIELD

Esta es una Shield que dispone de una pantalla LCD de 16x2 y un teclado para placas Arduino Uno, Diecimila, Duemilanove y Freeduino.

Características:

- Luz de fondo azul con las palabras blancas
- Utiliza 4 bits Arduino Biblioteca LCD

-Botones de Izquierda, Derecha, Arriba, Abajo y Select

-Ajuste de contraste de la pantalla

-Botón de reinicio Arduino

-La placa utiliza diferentes pines al ejemplo de Arduino, por lo que para hacer el trabajo de visualización, utilice la siguiente secuencia de seteo cuando se inicia la biblioteca:

lcd LiquidCrystal (8,9,4,5,6,7);

Los botones están conectados a un solo pin de entrada analógica a través de resistencias para dar una tensión diferente para cada botón, con el consiguiente ahorro en los pines de entrada / salida. La lectura de los botones es fácil.

Pines de Conexion

Pin Funcion

Analógico 0- Botones (Select, arriba, derecha, abajo e izquierda)

Digital 4-DB4

Digital 5 -DB5

Digital 6 -DB6

Digital 7 -DB7

Digital 8 -RS (visualización de datos o señal de selección)

Digital 9 -Habilitar(Enable)

Nota: No usar el pin digital 10 cuando este conectado este shield

Detalle del Hardware de los Botones.

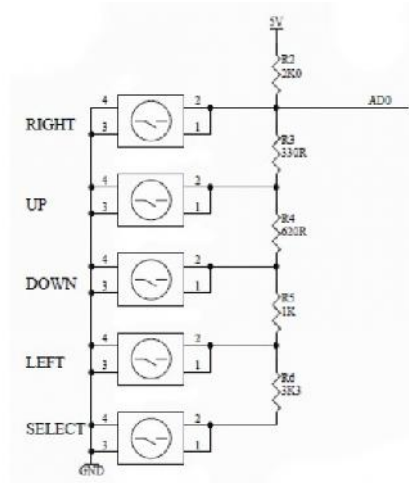


ILUSTRACIÓN 40-CIRCUITO DE BOTONES SHIELD LCD+KEYBOARD

Este Shield nos permite darle una mejor presentación del proyecto y minimizar los problemas de fallas por falso contactos, formando un solo bloque de placas, sin conexiones externas, pero se siguen utilizando 6 salidas digitales de arduino, lo cual a futuro puede limitar la capacidad de ampliación a futuro del proyecto.

Por esta razón se investigo y se encontraron los adaptadores Adaptador I2c para Display Lcd , basados en PCF8754 de Philips.

3.6.2 Adaptador I2c para Display PCF8754

Descripción General

El adaptador de LCD I2C convierte un LCD paralelo con interface HD44780 en un LCD I2C serie que puede ser controlado a través de sólo 2 hilos. El adaptador utiliza el chip de PCF8574A que se comunica con tu Arduino o cualquier otro microcontrolador utilizando el protocolo I2C. Un total de 8 pantallas LCD puede ser conectado al mismo bus I2C eligiendo una dirección diferente. La dirección I2C predeterminado es 0x3F y se puede cambiar a cualquiera de los siguientes 0x38 ~ 0x3F mediante soldadura pines A0 A1 A2.

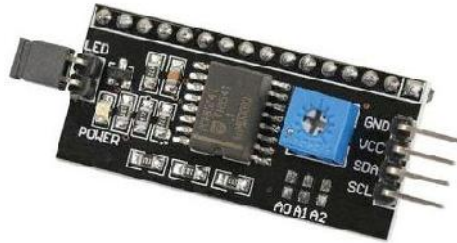


ILUSTRACIÓN 41-ADAPTADOR I2C PARA DISPLAY PCF8754

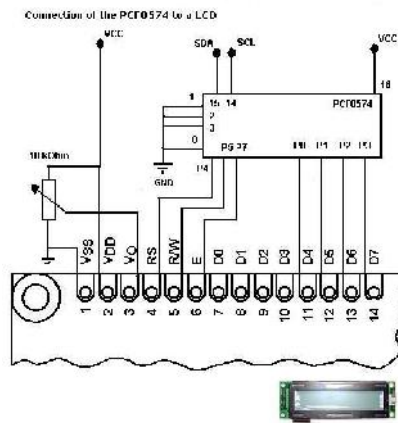


ILUSTRACIÓN 42-CONEXION DE PCF8754 A LCD



ILUSTRACIÓN 43-PCF8754 MONTADO SOBRE LCD

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

Características

- Facil de usar con cualquier display de caracteres
- Compatible con interface HD44780
- Hasta 8 controladores en el mismo bus
- Ajuste de contraste con preset
- Control de encendido de backlight con jumper
- Alimentacion de 5V

Este debe ser soldado a un display LCD 16x2, con un costo de \$76 (LCD 1602A \$100 + PCF8754 \$76).

Nos permite pasar de utilizar 6 salidas a solo 2 salidas i2c, mejorando la capacidad de ampliación a futuro.

3.6.3-Eleccion del Display

En la selección de display se optó por mantener el display Shield LCD +Keyboard varias razones:

- Fisicamente le da al proyecto mejor presentación y robustez, ya que no necesita un anclaje separado del Arduino y la Ethernet Shield.
- No necesita cableado auxiliar con lo que se minimiza las posibles fallas.
- Le suma la posibilidad de que el usuario interactúe con el sistema por medio de botones físicos.
- Los costos son los mismos que un display Lcd 1602A + Adaptador I2C.
- Al probar Lcd 1602A + Adaptador I2C, este luego de 10 min fallaba, no pudimos determinar si era a causa del adaptador o el lcd estaba fallado.

3.7- Alarma Sonora

Para realizar esta función se propuso utilizar un Buzzer, que utiliza el elemento Piezoeléctrico.

La Piezoelectricidad es un fenómeno que ocurre en determinados cristales que, al ser sometidos a tensiones mecánicas, adquieren una polarización eléctrica y aparece una diferencia de potencial y cargas eléctricas en su superficie que generan una tensión eléctrica.

Este fenómeno también ocurre a la inversa: se deforman bajo la acción de fuerzas internas al ser sometidos a un campo eléctrico. El efecto piezoeléctrico es normalmente reversible: al dejar de someter los cristales a un voltaje exterior o campo eléctrico, recuperan su forma.

Es decir, que son materiales (el cuarzo es el más conocido) que si los sometemos a una tensión eléctrica variable (como una señal PWM, que ya nos son familiares) vibran.

Es un fenómeno bastante conocido y muchos encendedores domésticos de gas funcionan bajo este principio. Un resorte golpea un cuarzo y como resultado tenemos la chispa que enciende el gas o el calentador de agua con un característico click).

En otro orden de cosas, los circuitos electrónicos digitales, suelen disponer de un reloj interno que vibra a una velocidad patrón, basados en cristales de cuarzo piezoeléctrico. El cristal de Arduino vibra a 16Mhz por segundo y la flecha indica su posición.

Si conectamos un piezo con una señal digital, vibran a una frecuencia sigue bastante fielmente la variación eléctrica con que los excita, y si vibran a la frecuencia audible, oiremos el sonido que producen. A un componente que hace esto, le llamamos Buzzer o zumbador.

Naturalmente, la calidad del sonido que producen dista bastante de lo que podríamos denominar alta fidelidad. Pero es suficiente para generar tonos audibles (como la típica alarma de los despertadores digitales) e incluso tonos musicales reconocibles que podemos secuenciar, hasta en piezas musicales (por más que uno quisiera estar en otro lugar cuando las oyes).

Con la plataforma de desarrollo de Arduinio, la conexión es tan simple como conectar negativo a GND y positivo al pin 9. De todas maneras hay que tener cuidado. Los piezos tienen polaridad y hay que asegurarse de conectarlos correctamente. Si se conecta al revés, simplemente no sonará, y tendrá que dar la vuelta la conexión.

Además necesitamos usar un pin PWM (como el 9) porque es la alternancia entre HIGH y LOW lo que produce el efecto piezoeléctrico (recordad que una señal PWM envía un tren de ondas cuadradas de amplitud variable), por lo que lo más cómodo es usar un pin PWM en lugar de programar un efecto equivalente en un pin normal.

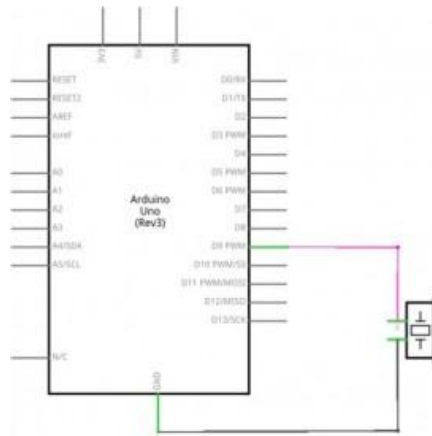


ILUSTRACIÓN 44-DIAGRAMA DE CONEXION ARDUINO + BUZZER

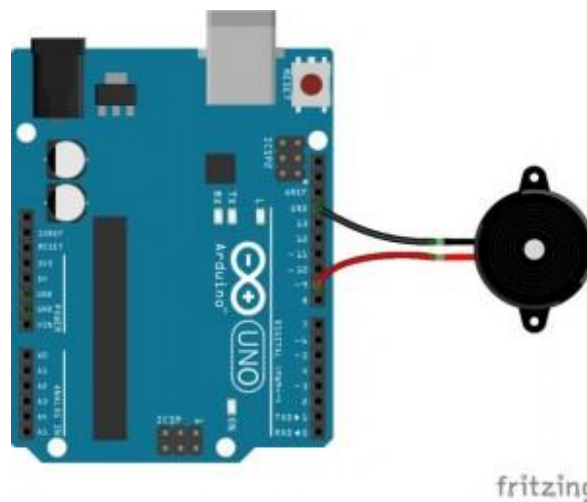


ILUSTRACIÓN 45-CONEXION REAL ARDUINO + BUZZER

3.7.1- Interruptor de Apertura de Puerta de Heladera

Se propuso utilizar un Interruptor Reed Switch como los utilizados en los sistemas de alarma, por su tipo de montaje externo, lo que nos permite adaptarlo a las puertas de las heladeras, sin realizar modificaciones en interior de estas.

Un magnetic reed es un dispositivo electromecánico que se comporta como un interruptor que se activa ante la presencia de un imán.



ILUSTRACIÓN 46-MAGNETIC REED



ILUSTRACIÓN 47-MAGNETIC REED UTILIZADO COMO SENSOR DE PUERTA

Los sensores magnetic reed son ampliamente utilizados. Por ejemplo, muchas alarmas de puertas y ventanas funcionan ubicando un imán en el elemento, y detectando la apertura con un magnetic reed. También podemos ubicar el imán en una puerta, o vitrina, para encender una luz, etc.

Como un magnetic reed no deja de ser un interruptor, podemos emplear este dispositivo para encender o apagar un dispositivo directamente. Por supuesto, también podemos emplear las entradas digitales de Arduino para leer el estado del magnetic reed, de forma similar a como vimos en la entrada leer un pulsador con Arduino.

Los interruptores magnéticos tienen la ventaja de ser baratos, sencillos y no requieren corriente para su funcionamiento. Además, funcionan tanto en corriente alterna como continua.

Por el contrario, al ser dispositivos electro mecánicos, el tiempo de conmutación es relativamente alto, del orden de 1-5 ms.

Además, los magnetic reed tienen una vida útil limitada, es decir, hay un número de conmutaciones que puede realizar antes de dañarse. No obstante, en algunos dispositivos puede ser hasta 100 millones de conmutaciones, lo cual es suficiente para la mayoría de aplicaciones. Por último, la presencia de vibraciones o movimientos bruscos puede alterar su funcionamiento, y dar lugar a mediciones erróneas.

Costo

Los interruptores magnetic reed son dispositivos muy baratos el precio es desde \$70 a \$120.

El precio depende de la intensidad y tensión máxima, la potencia máxima que pueden gestionar, y de la velocidad de conmutación y vida útil.

Funcionamiento de Magnetic Reed

Físicamente un magnetic reed está constituido por dos elementos ferromagnéticos de níquel, ubicados en el interior de una ampolla de vidrio sellada.

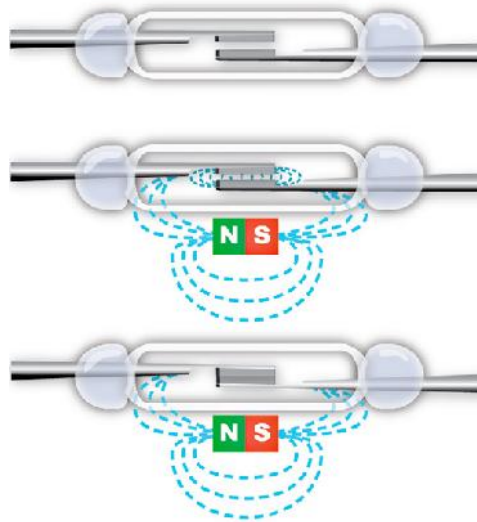


ILUSTRACIÓN 48-APERTURA Y CIERRE DE MAGNETIC REED

Al acercar un campo magnético la fuerza generada provoca que ambos elementos entren en contacto, cerrando el circuito eléctrico.

También existen magnetic reed cuyo estado es normalmente cerrado, y abren el circuito eléctrico en presencia del campo magnético. Incluso encontramos magnetic reed que disponen ambas salidas, una normalmente abierta y otra normalmente cerrada, en el mismo dispositivo.

El campo magnético requerido para activar el interruptor es típicamente del orden de 50 Gauss.

Esquema Electrico

El esquema eléctrico es sencillo. Vamos a usar las resistencias internas de pull-up de Arduino, por lo que simplemente conectamos el magnetic reed entre GND y la entrada digital que queramos emplear.

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

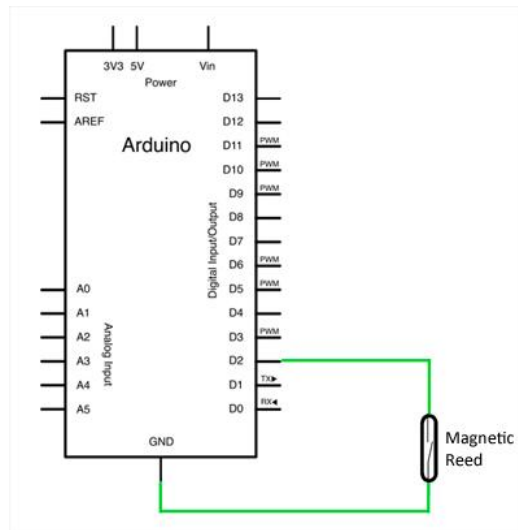


ILUSTRACIÓN 49-MONTAJE SIN RESISTENCIA PULL-DOWN

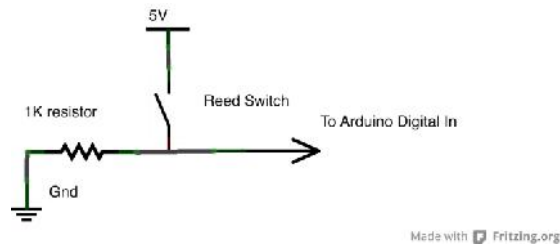


ILUSTRACIÓN 50-MONTAJE CON RESISTENCIA PULL-DOWN

Esquema de Montaje

Mientras que el montaje en una protoboard sería el siguiente.

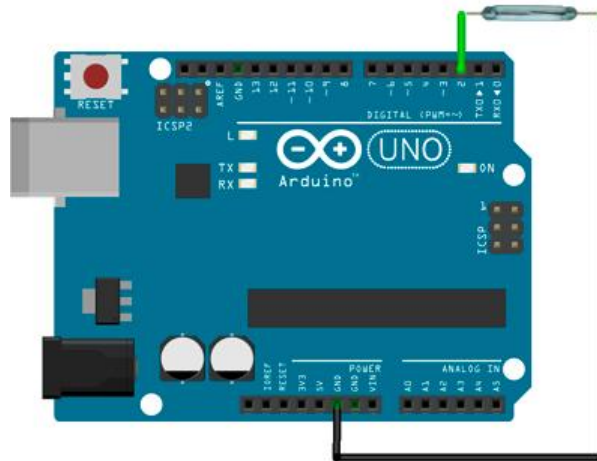


ILUSTRACIÓN 51-MONTAJE EN ARDUINO SIN RESISTENCIA PULL-DOWN

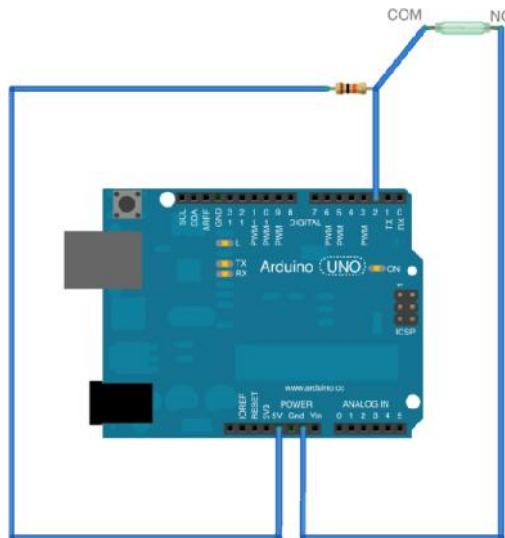


ILUSTRACIÓN 52-MONTAJE EN ARDUINO CON RESISTENCIA PULL-DOWN

3.8- Herramientas de software para el desarrollo

Para el desarrollo del sistema de alimentación se utilizan las siguientes herramientas software y hardware:

-Arduino Software 1.6.9

Entorno de desarrollo (IDE) basado en el entorno de Processing y lenguaje de programación basado en Wiring, así como en el cargador de arranque (bootloader) que es ejecutado en la placa.⁴ El microcontrolador de la placa se programa a través de un computador, haciendo uso de comunicación serial mediante un convertidor de niveles RS-232 a TTL serial.

-Mango M2M

Es uno de los software M2M (Machine-to-Machine) más populares del mundo, también conocido como software de control industrial o SCADA. Mango es una aplicación web, software M2M desarrollado en Ajax, que permite a los usuarios acceder y controlar sensores electrónicos, dispositivos y máquinas a través de múltiples protocolos simultáneamente. Mango ofrece una interfaz con la que diversas fuentes de datos se pueden crear y configurar al mismo tiempo que permite la gestión del acceso de los usuarios, alertas, registro de datos y automatización.

- Turnkey Mango 1.8

Es una biblioteca de dispositivos virtuales basada en Debian, que integra soluciones del tipo ready-to-use con el mejor software de código abierto.

Es una distribución sin entorno de escritorio ya que su principal objetivo es ser usada como servidor, ofrece una inmensa cantidad de ediciones. Estas ediciones son soluciones o “appliance” que son imágenes de disco con servicios pre instalados. De esta forma, instalando una imagen determinada obtenemos nuestro servidor sin tener que hacer nada más.

En nuestro caso disponemos de un sistema servidor debían con un servidor web en el cual esta instalado Mango M2M

-Vmware Player 12 utilizado para ejecutar TurnKey Mango.

Es una aplicación de virtualización de escritorios que ejecuta uno o varios sistemas operativos en el mismo ordenador sin necesidad de reiniciar.

-Fritzing

Fritzing es un programa de automatización de diseño electrónico libre que busca ayudar a diseñadores y artistas para que puedan pasar de prototipos (usando, por ejemplo, placas de pruebas) a productos finales.

Fritzing fue creado bajo los principios de Processing y Arduino, y permite a los diseñadores, artistas, investigadores y aficionados documentar sus prototipos basados en Arduino y crear esquemas de circuitos impresos para su posterior fabricación. Además, cuenta con un sitio web complementario que ayuda a compartir y discutir bosquejos y experiencias y a reducir los costos de fabricación.

- ClickCharts

ClickCharts es un programa para la creación de diagramas de flujo gratuito.

Para crear de todo, desde diagramas de flujo de organización y procesos hasta organigramas, mapas conceptuales y otras secuencias con una fácil edición y herramientas personalizadas.

3.9- Introducción al concepto M2M

M2M (Machine to Machine o Máquina a Máquina) es un concepto genérico que indica el intercambio de información en formato de datos entre dos máquinas remotas.



ILUSTRACIÓN 53-LOGO DE MANGO M2M

Elementos Fundamentales

Los elementos fundamentales que aparecen en todos los entornos M2M son los siguientes:

- **Máquinas que gestionar:** Gestión de flotas, Alarmas domésticas, TPV (Terminal Punto de Venta), Contadores de agua/gas/ electricidad, paneles informativos en carreteras, máquinas vending, telemantenimiento de ascensores, estaciones meteorológicas.

- **Dispositivo M2M:** módulo conectado a la máquina y que provee de comunicación con el servidor. Usualmente, el dispositivo M2M también consta de capacidad de proceso donde se ejecuta la aplicación de negocio. Por una parte implementa el protocolo para poder comunicarse con la máquina y por otra parte implementa el protocolo de comunicación para el envío de información.



ILUSTRACIÓN 54-DISPOSITIVO M2M

- **Servidor:** Ordenador que gestiona el envío y recepción de información de las máquinas que gestiona. Habitualmente está integrado además con el core business de la empresa (ERP, MapasGIS de trazabilidad de flotas de camiones, Sistema de pedidos, Centrales receptoras de alarmas, Helpdesk, ...) de modo que la información recibida pasa a ser parte crítica del negocio.

- **Red de comunicación:** pueden ser de dos naturalezas principalmente, a través de cable: PLC,Ethernet, RTC, RDSI, ADSL ... o bien a través de redes inalámbricas: GSM/UMTS/HSDPA, Wifi,Bluetooth, RFID, Zigbee, UWB,

APLICACIONES

En una publicidad de Telefónica España decía "El ser humano nunca podrá competir con las máquinas que inventa ni en eficiencia ni fiabilidad". Y es que la multinacional ha creado una unidad internacional específica que se llamará Global M2M con el objetivo de ofrecer soluciones M2M, es decir, sistemas informatizados que se comunican entre sí y controlan procesos como leer un contador, chequear la salud de un paciente, desarrollar programas para aseguradoras que controlen la agresividad o tranquilidad de sus asegurados, etc.

El sistema M2M (Máquina a Máquina) funciona con un tarjeta SIM que informa y se comunica con otros mecanismos para detectar, analizar y ejecutar procesos. Un ejemplo de ello es el de la directiva europea, que obligará a partir de 2012 a que todos los coches que se fabriquen lleven una tarjeta SIM que avise automáticamente al número 112 en caso de accidente.

Respecto a los contadores de agua, gas, luz, etc. las compañías respectivas ya están desarrollando planes en los que la M2M es el pilar de funcionamiento. Estos planes buscan informatizar todas las lecturas y diagnósticos de averías para poder ahorrar la mano de obra humana, es decir, eliminar los puestos de trabajo de personas que actualmente van casa por casa haciendo las lecturas del consumo de los usuarios.

Es seguro que los millones de máquinas que existen en el mundo establecerán comunicación un día muy próximo. De esto se trata el concepto Máquina a Máquina (M2M), el

cual se utiliza en plataformas como GSM. Las posibilidades son ilimitadas, y por supuesto, generará un gran negocio.

Pensamos en la cantidad de máquinas que existen en el mundo? Hablamos de millones, desde equipos electrodomésticos hasta maquinaria industrial. A cada una se le debe proporcionar comunicación y un canal seguro de bajo precio, lo que se obtiene con un concepto idóneo: M2M (Machine to Machine).

La compañía que se encuentra a la cabeza de la comunidad M2M es Nokia, pues emplea la plataforma GSM, la misma que se utiliza para la comunicación celular entre personas.

El nuevo concepto

Esta modalidad abarca todos los sistemas que permitan comunicar máquinas con otras máquinas (máquina-a-máquina), personas (móvil-a-máquina), sistemas de información de empresas (máquina-a-sistema) y proveer información en tiempo real.

Hoy en día existen máquinas que desearían comunicarse: una tragamonedas de refrescos quisiera avisarle al camión repartidor que le surta algún producto cuando está por agotarse. Este aviso le ayudaría al repartidor a detectar dicha máquina, llevar el refresco necesario y prever siempre que esté bien surtida sin importar la temporada o el lugar donde se encuentre; con ahorros sustanciales en gasolina y tiempo al no tener que realizar viajes innecesarios.

Una aplicación interesante sería que los medidores de luz enviaran el total de consumo realizado en el mes a la central e hicieran más eficiente la lectura mediante la disminución de tiempos en la facturación y cobranza; además eliminaría errores de captura y recapturas.

Otro ejemplo sería que un turista, a través de su agenda personal conectada a Internet, pudiera verificar que las cerraduras estén puestas en su casa, que la alarma de incendio o seguridad se encuentre funcionando, y activara los equipos de riego del jardín.

Las soluciones M2M están diseñadas para incrementar la productividad, las utilidades y la competencia al reducir costos y tiempos, agilizar procesos, mejorar el servicio al cliente, y sobre todo, nuevas formas de hacer las cosas.

Un gran campo de acción

Estas aplicaciones están desarrolladas para enviar indicaciones de situaciones inusuales, recolectar información o configurar parámetros de acuerdo a las necesidades del negocio de manera remota.

Sin embargo, en su mayoría se utilizan para servir a todas las áreas de los negocios: monitoreo de elevadores en centros comerciales, bajar juegos en máquinas tragamonedas, monitorear temperaturas en maquinarias, calderas, albercas e incluso monitorear el uso de las fotocopiadoras, sólo por listar algunas funcionalidades.

Mango M2M, código abierto

Mango es la alternativa de código abierto para software máquina a máquina (M2M) (también conocido como SCADA, HMI, o domótica). Mango esta basada en navegador, software Ajax de M2M que permite a los usuarios el acceso y el control de sensores electrónicos, dispositivos y máquinas de múltiples protocolos simultáneamente. Fácil de descargar e instalar en muchas plataformas, Mango ofrece una interfaz con la que diversas fuentes de datos se pueden crear y configurar la vez que proporciona la gestión de acceso de los usuarios, alertas, registro de datos y la automatización.

La misión del proyecto Mango es ofrecer una alternativa de código abierto dentro de la casa / edificio / campo de la automatización industrial.

La versión actual es una versión de trabajo del software. Algunas funciones disponibles en su versión comercial del producto no están disponibles. Múltiples protocolos estándar de aplicación incluyendo BACnet I / P, Modbus (RTU, ASCII, TCP y UDP), SNMP, NMEA 0183, y protocolos propietarios, así como varias fuentes de datos normalmente no se ofrece en software

de automatización como el receptor de HTTP, HTTP analizador, y acceso a bases de datos de SQL.

Aplicaciones Mango

Las aplicaciones típicas de Mango son las siguientes:

- Edificios / instalaciones
- Equipos de oficina, HVAC / sistemas ambientales y controles
- Domotica
- Control de las luces, la calefacción, aire acondicionado, puertas de garaje, todo lo que se deja conectado. Monitor de temperatura, humedad, puerta y ventana de contactos, el tanque de la acidez de pescado, etc Seguimiento de su consumo de electricidad, gas y agua.
- Consumidores y profesionales
- Electrodomésticos / electrodomésticos, electrónica de consumo
- Energía
- medidores de servicios, conservación, distribución generadores, red eléctrica y tuberías
- Industrial
- Equipo de automatización y control, equipos de seguimiento de activos de capital
- Medicina / Salud
- Dispositivos médicos
- Venta al por menor
- Escáneres y registros, la iluminación y los sistemas de refrigeración
- Transporte

- Flota de vehículos de la gestión, aviones y sistemas de transporte

Características

- **Velocidad**

El mango es un sistema de registros de datos, control y supervisión del sistema todo en uno con una interfaz basada en navegador. Sin embargo, en una máquina Linux a 800 MHz con 500 MB de RAM Mango puede albergar cientos de puntos de los datos obtenidos de múltiples fuentes de datos usando protocolos múltiples, incluso mientras se utiliza la base de datos incorporada.

- **Múltiples protocolo**

Mango pueden recibir datos desde cualquier dispositivo para el que hay un controlador de protocolo. protocolos soportados actualmente incluyen BACnet I / P, Modbus (ASCII, RTU, TCP y UDP), 1-alambre, SNMP, SQL, HTTP, POP3, NMEA 0183, MBUS, DNP3, OpenV, webcams, vmstat, y muchos protocolos propietarios desarrollado por los fabricantes de hardware. Mango también es compatible con datos de origen "virtual" que pueden generar datos para la evaluación comparativa o propósitos de prueba.

- **Múltiples bases de datos**

Mango tienen base de datos integrada de modo que usted no que tiene que gestionar una base de datos aparte. Pero, Mango también puede usar MySQL para usuarios exigentes que necesitan un rendimiento.

- **Meta puntos**

Utilizar secuencias de comandos para crear nuevos puntos basados en los valores de otros puntos. Sobre la base de Javascript, el lenguaje de scripting más populares en el mundo, los

puntos de meta datos permiten combinaciones potentes de los valores de punto, así como información de los puntos históricos.

- **Eventos definidos por el usuario**

Se puede indicar a Mango que eventos que les interesa a los usuarios, pueden definir criterios ilimitados en caso de detectar puntos de condiciones tales como alta y límites bajos, cambios de valores, cambio de cuenta del Estado, y tiempos de ejecución.

- **Importación y Exportación**

Exporte su configuración en un archivo de texto. Guardar este archivo como respaldo y recuperación, o utilizarlo para la importación en otros casos de Mango para hacer copias idénticas. O bien, utilizar el archivo para gestionar grandes configuraciones con mucha facilidad.

- **Soporte para Internet Explorer 7, Firefox, y Chrome**

Soporte completo para los navegadores más populares en el mundo.

- **Gestión de eventos**

Todos los eventos que ocurren, ya sea del sistema o definidos por el usuario, puede ser manejado arbitrariamente usando controladores de eventos definidos del usuario. Estos controladores pueden enviar mensajes de correo electrónico y la progresividad a listas de correo, o establecer valores en puntos de Mango.

- **Seguridad**

Todos los datos, quedan en forma residente desde que instala Mango, por lo que está en control. Permisos de usuario se definen por los administradores del sistema, y todas las comunicaciones con Mango se puede asegurar con SSL (Secure Socket Layer), garantizando la privacidad de su información.

- **Registro de datos**

Cada punto se puede configurar con sus propios datos de registro y registro de depuración características. los horarios de registro se pueden hacer para ser independiente de la lectura horarios.

- **Informes**

Crear y programar informes para ver en línea o por correo electrónico. Descargar datos en formato CSV para carga rápida en hojas de cálculo u otros programas de análisis de datos.

- **Publicación de los datos**

Reenviar la información recogida por Mango para otros sistemas de la arquitectura de M2M en evento de moda en la próximas actualización en tiempo real.

- **Revisar listas**

Crea tu propia lista personalizada de los puntos que usted desea ver. Fácilmente añadir y eliminar puntos de la lista para mantener un ojo en los valores de punto en particular y alarmas. Crear listas con chequeo de puntos con sus grupos favoritos de puntos. Vea al instante gráficos de múltiples puntos de los datos de puntos en su lista.

- **Punto de jerarquías**

Crea tus propias jerarquías arbitrarias de los puntos y el punto carpetas para organizar la información de la manera deseada para verla.

- **Puntos de vista gráfico**

Crear representaciones gráficas de los puntos a sus propios fondos, con funcionalidad de arrastrar y soltar. Todos los dispositivos con controles gráficos y gráficos funcionan de la misma en los gráficos como lo hacen en todas partes.

- **Punto de información**

Punto de información, incluyendo detalles Ver valor actual, detallada y gráfica de gráficos de tabla, alarmas, detectores de eventos y permisos de usuario.Establecer controles de punto están también disponibles.

- **Activacion de las alarmas.**

Todas las páginas de la aplicación incluyen un indicador del nivel más alto de alarma activa. Utilícelo para vincular a la lista de las alarmas activas donde se pueden ver todas las alarmas activas de un vistazo. Lea y añadir comentarios y punto de enlace a los detalles páginas donde se pueden tomar medidas

- **Notas de usuario**

Los usuarios pueden comentar sobre los eventos y puntos de manera que el conocimiento valioso no se pierde. comentarios de sucesos se envían las notificaciones de correo electrónico para que todos los usuarios se mantengan al día sobre el estado del sistema.

- **Evento de programación**

Definir eventos basados en los horarios. Eventos planteadas por los horarios tienen acceso a toda la funcionalidad de manipulación que los tipos de eventos otros.

- **Datos de auditoría**

Los cambios en el procesamiento de todos los objetos de información hacen que los eventos de auditoría que se plantearon, incluyendo objetos nuevos, cambios y tachaduras. Estos eventos pasan por el sistema de gestión de eventos para que todos los usuarios de forma independiente puede reconocer el hecho.

- **Las actualizaciones automáticas de software**

Recepción de notificaciones cuando las nuevas versiones están disponibles de Mango. Basta con copiar la nueva versión de Mango en su directorio de instalación y todos sus datos se actualizará automáticamente.

- **I18N (internacionalización)**

Mango es totalmente internationalizable. Nuevos idiomas pueden ser apoyadas simplemente traducir el archivo de etiqueta (y archivos de documentación contextual). En la actualidad se traduce idiomas Inglés, alemán, portugués, holandes, y el chino.

- **Alarma Sonora**

Escuchar sonidos cuando las alarmas se generan en Mango.

3.10-Internet de las cosas(Internet of things)

Dentro de la investigación al basarse en la plataforma Arduino, se encontraron muchas aplicaciones similares, basadas en el concepto de Internet de las cosas

(en inglés, Internet of things, abreviado IoT), lo cual nos permitió ver otra alternativa a la monitorización de los parámetros a controlar

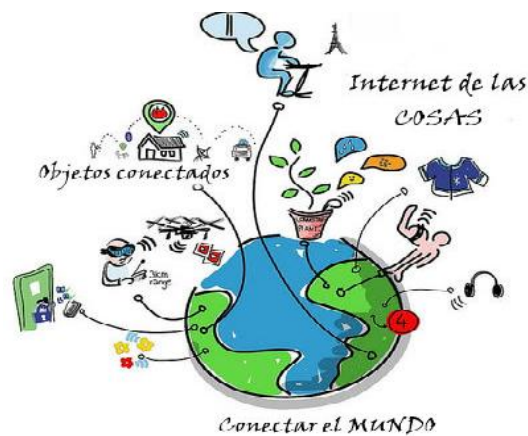


ILUSTRACIÓN 55-DESCRIPCIÓN GRÁFICA DEL MUNDO INTERCONECTADO.

Internet de las cosas (en inglés, Internet of things, abreviado IoT)^{1 2} es un concepto que se refiere a la interconexión digital de objetos cotidianos con internet. Alternativamente, Internet de las cosas es el punto en el tiempo en el que se conectarían a internet más “cosas u objetos” que personas. También suele referirse como el internet de todas las cosas o internet en las cosas. Si los objetos de la vida cotidiana tuvieran incorporadas etiquetas de radio, podrían ser identificados y gestionados por otros equipos, de la misma manera que si lo fuesen por seres humanos.

El concepto de internet de las cosas lo propuso Kevin Ashton en el Auto-ID Center del MIT en 1999, donde se realizaban investigaciones en el campo de la identificación por radiofrecuencia en red (RFID) y tecnologías de sensores.

Por ejemplo, si los libros, termostatos, refrigeradores, la paquetería, lámparas, botiquines, partes automotrices, etc. estuvieran conectados a Internet y equipados con dispositivos de identificación, no existirían, en teoría, artículos fuera de stock o medicinas caducadas; sabríamos exactamente la ubicación, cómo se consumen y se compran productos en todo el mundo; el extravío sería cosa del pasado y sabríamos qué está encendido o apagado en todo momento.

El internet de las cosas debería codificar de 50 a 100 000 billones de objetos y seguir el movimiento de estos; se calcula que todo ser humano está rodeado de por lo menos 1000 a 5000 objetos. Según la empresa Gartner, en 2012 habrá en el mundo aproximadamente 26 mil millones de dispositivos con un sistema de adaptación al internet de las cosas. Abi Research, por otro lado, asegura que para el mismo año existirán 30 mil millones de dispositivos inalámbricos conectados al Internet. Con la próxima generación de aplicaciones de Internet (protocolo IPv6) se podrían identificar todos los objetos, algo que no se podía hacer con IPv4. Este sistema sería capaz de identificar instantáneamente por medio de un código a cualquier tipo de objeto.

La empresa estadounidense Cisco, que está desarrollando en gran medida la iniciativa del internet de las cosas, ha creado un “contador de conexiones” dinámico que le permite estimar el número de “cosas” conectadas desde julio de 2013 hasta el 2020. El concepto de que los dispositivos se conectan a la red a través de señales de radio de baja potencia es el campo de

estudio más activo del internet de las cosas. Este hecho se explica porque las señales de este tipo no necesitan ni Wi-Fi ni Bluetooth. Sin embargo, se están investigando distintas alternativas que necesitan menos energía y que resultan más baratas, bajo el nombre de “Chirp Networks”.

Actualmente, el término internet de las cosas se usa con una denotación de conexión avanzada de dispositivos, sistemas y servicios que va más allá del tradicional M2M (máquina a máquina) y cubre una amplia variedad de protocolos, dominios y aplicaciones.¹⁸ El servicio touchatag de Alcatel-Lucent y el gadget Violeta Mirror pueden proporcionar un enfoque de orientación pragmática a los consumidores del internet de las cosas, por el que cualquiera puede enlazar elementos del mundo real al mundo en línea utilizando las etiquetas RFID (y códigos QR en el caso de touchatag).

3.10.1- APIs para el Internet de las Cosas: ThingSpeak y Pachube

Antes a hablar de las API para internet de las Cosas, explicaremos que es una API.

Una API es una interfaz de programación de aplicaciones (del inglés API: Application Programming Interface). Es un conjunto de rutinas que provee acceso a funciones de un determinado software.

Son publicadas por los constructores de software para permitir acceso a características de bajo nivel o propietarias, detallando solamente la forma en que cada rutina debe ser llevada a cabo y la funcionalidad que brinda, sin otorgar información acerca de cómo se lleva a cabo la tarea. Son utilizadas por los programadores para construir sus aplicaciones sin necesidad de volver a programar funciones ya hechas por otros, reutilizando código que se sabe que está probado y que funciona correctamente.

En la web, las API's son publicadas por sitios para brindar la posibilidad de realizar alguna acción o acceder a alguna característica o contenido que el sitio provee.

Las APIs se han convertido en la sangre del software. En el gran negocio del Internet de las Cosas, la interfaces de programación de aplicaciones cumplen un papel esencial en la recogida, análisis, visualización y manipulación de los datos. Aquí tres buenos ejemplos.

Son estas APIs las que permiten a los desarrolladores conectarnos al mundo de los objetos y relacionar entre sí los dispositivos que nos rodean: ‘enchufar’ nuestro cuerpo a medidores de constantes vitales, domotizar nuestra casa, convertir nuestro vehículo casi en un ser vivo... Las APIs se han convertido en la sangre del software y el llamado Internet de las Cosas no existiría sin ellas. Las APIs incorporan y trabajan con los datos que facilitan la conexión entre objetos.

Hoy en día la mayoría de soluciones del Internet de las Cosas se mueve alrededor de cuatro pilares básicos que son movilidad, economía social, análisis de grandes volúmenes de datos y la nube. Según un estudio de Cisco, el mercado global estimado en torno al IoT se elevará hasta los 14,4 trillones de dólares en 2022, una cifra que no ha dejado de crecer en los últimos años. Por tanto, hay negocio para los desarrolladores y las APIs son una de las grandes claves.

3.10.2-ThingSpeak API

La API de ThingSpeak siempre trabaja con datos, esa es su gran especialidad. Es una API abierta para el Internet de las Cosas que permite recopilar, almacenar, analizar, visualizar y actuar sobre la información recogida en sensores y dispositivos como aplicaciones web y móviles, redes sociales como Twitter, soluciones de mensajería, VoIP y nube como Twilio, hardware de código abierto como Arduino, Raspberry Pi o BeagleBone (los reyes del Internet de las Cosas y la robótica) o con lenguajes de cálculo computacional como MATLAB... ThingSpeak es una API conocida entre los desarrolladores y dispone ya de una gran comunidad.

ThingSpeak API funciona siempre con canales, los cuales contienen los campos de datos, ubicación y estado. Para empezar a trabajar con esta interfaz es necesario crear un canal, donde se recopilará la información de dispositivos y aplicaciones, datos que posteriormente se pueden analizar y visualizar en gráficos (este es un tutorial bastante completo de cómo crear gráficos con

ThingSpeak) y el paso final es operar sobre esa documentación. El proceso con la API siempre es el mismo. Al final, la ruta del proyecto con la API será parecida a lo siguiente, sustituyendo los campos CHANNEL_ID y FIELD_ID por los datos del canal recién abierto.

http://api.thingspeak.com/channels/CHANNEL_ID/charts/FIELD_ID

The screenshot shows the 'New Channel' form on the ThingSpeak website. The form is titled 'New Channel' and has a blue header with the ThingSpeak logo and navigation links: 'Channels', 'Apps', 'Blog', and 'Support'. The form fields are as follows:

- Name:** Dew Point Measurement
- Description:** (empty text area)
- Field 1:** Temperature [F] (checked)
- Field 2:** Humidity (checked)
- Field 3:** Dew Point (checked)
- Field 4:** (empty, partially obscured by a decorative border)
- Show Location:** (unchecked)
- Latitude:** 0.0
- Longitude:** 0.0
- Show Video:** (unchecked)
- Video ID:** (empty)
- Show Status:** (unchecked)

A green 'Save Channel' button is located at the bottom of the form.

ILUSTRACIÓN 56-CAPTURA DE PANTALLA DE CANALES DE THINGSPEAK

Un ejemplo del tipo de gráficos que se pueden crear fruto de la recogida, análisis y visualización con ThingSpeak son estas mediciones de temperatura:

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

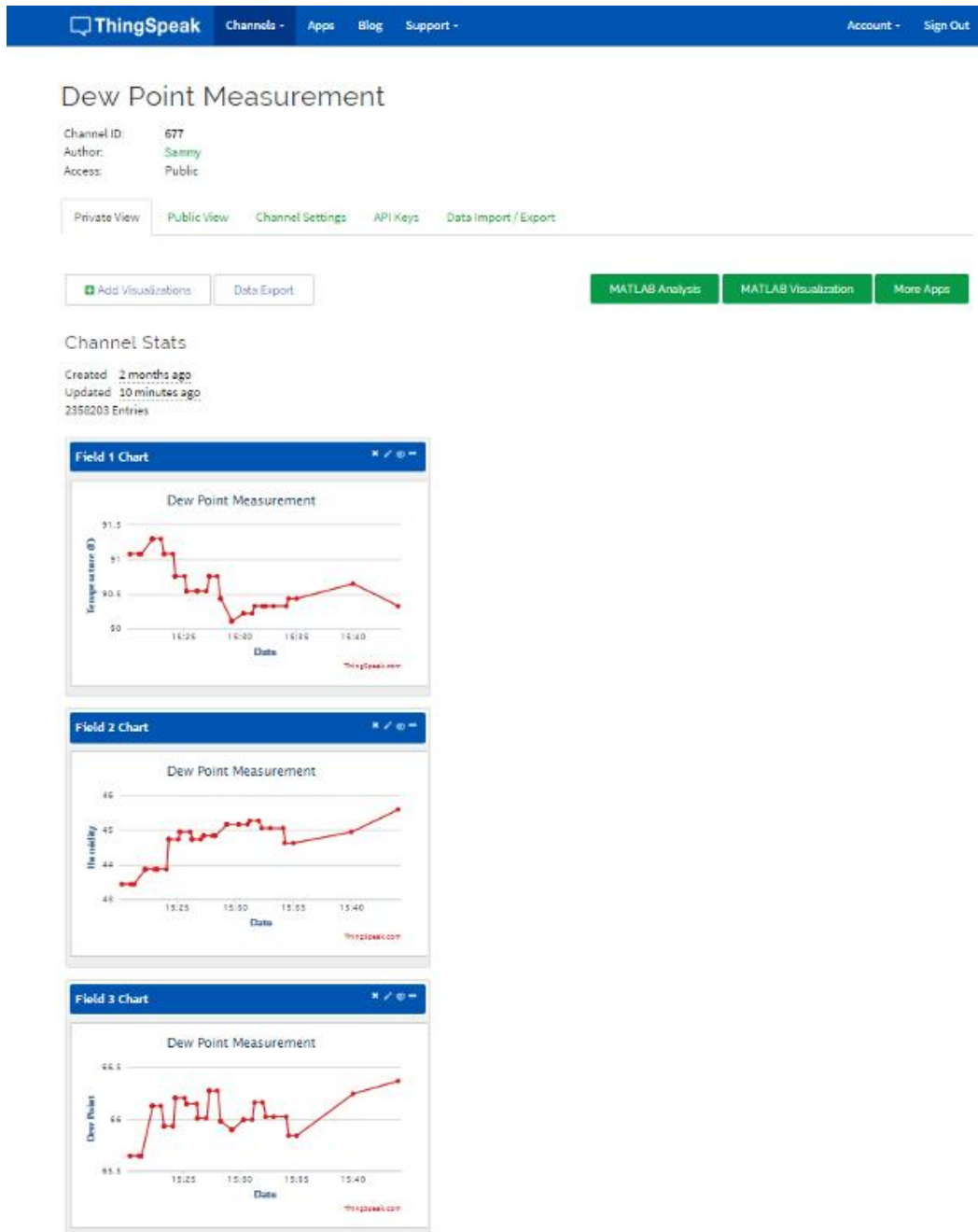


ILUSTRACIÓN 57-GRAFICOS DE CANALES THINKPEAK

Dentro de la investigación se utilizo un widget para Android para Thingspeak

3.10.3- IoT ThingSpeak Monitor Widget

IoT ThingSpeak Monitor Widget



ILUSTRACIÓN 58-IOT THINGSPEAK MONITOR WIDGET

Este widget permite estar siempre al día con el estado de los dispositivos conectados a la IO Thingspeak!

No es necesario para iniciar la aplicación cada vez que quiera conocer sus lecturas reales, ya se puede ubicar en la pantalla principal del dispositivo.

Los datos se actualiza por la conexión de datos, ya sea por Wifi o 3G y se puede setear cada cuanto tiempo se actualizar los datos, de forma de ahorrar en la conexión de datos.

Las características del widget:

- Crear widgets para monitorear los valores reales de los campos en su canal - uno o dos en cada dispositivo en forma simultánea.
- Supervisar muchos campos de diferentes canales que crean varios widgets en una pantalla.
- Monitorizar los canales privados que utilizan Leer claves de la API.
- Establecer los umbrales de alerta superior e inferior para recibir alertas si el valor del campo programado sobrepasa estos límites.
- Ver y personalizar gráficos, establecieron período o los resultados del recuento, media, la suma o la mediana de los valores del canal.

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

-Configurar la dirección URL de su propia instancia del servidor Thingspeak para monitorizar los datos de ella.

-Configurar el tiempo de actualización necesaria para todos los widgets.

Personalizar la interfaz de usuario del widget, valor de redondeo y tamaño de fuente, color de fondo y la transparencia del fondo.



ILUSTRACIÓN 59-PANTALLA DE CELULAR CON LA APLICACION IOT THINGSPEAK MONITOR WIDGET



ILUSTRACIÓN 60-ALARMA IOT THINGSPEAK MONITOR WIDGET



ILUSTRACIÓN 61-GRAFICO GENERADO POR IOT THINGSPEAK MONITOR WIDGET

3.10.4.-Pachube API (Xively)

La API de Pachube permite almacenar, compartir y analizar en tiempo real los datos de energía o medioambientales recogidos por sensores en edificios y otros dispositivos. Toda la funcionalidad del sistema de Pachube viene dada por la API, que es la que facilita que la información generada por edificios, contadores de energía o dispositivos móviles con sensores sea recogida y analizada y que todos esos objetos estén conectados entre sí. Es el verdadero Internet de las Cosas.

La API de Pachube funciona en el entorno EEMML (Extended Markup Language Environments), un protocolo para el intercambio de datos de sensores en ambientes a distancia, ya sean físicos o también virtuales (un ejemplo típico es la conexión con objetos de Second Life). Que exista un protocolo universal es lo que posibilita que las personas puedan compartir con la comunidad millones de datos en tiempo real de objetos, dispositivos o espacios de todo el mundo.



ILUSTRACIÓN 62-PACHUBE

Los pasos a seguir para comenzar a trabajar con Pachube son bastante sencillos:

Agregar un dispositivo: el desarrollador da un nombre, una descripción y permisos de privacidad al dispositivo que quiere agregar y automáticamente se le asigna un ID y la clave de la API necesaria para conectar ese terminal.

Conectar el dispositivo: es necesario copiar el ID y la clave de la API en el código del objeto para establecer una relación bidireccional entre ambos. Se pueden conectar dispositivos, pero también apps o servicios. Para realizar esa conexión, aquí hay todo tipo de librerías para dispositivos Android, objetos programados en Java, C, Objective-C, Python, PHP, Ruby y JavaScript, y tutoriales para hardware abierto como Arduino o Raspberry Pi.

Prueba de funcionamiento: una vez agregado y conectado, el desarrollador puede empezar a ver valores de datos actualizados en tiempo real en su canal y depurar las alteraciones irregulares que vea en esa información.

Capitulo 4- Interacción e implementación

4.1-Hardware

Para realizar el diseño final de proyecto, se utilizo el siguiente hardware:

-Microcontrolador

Arduino Uno R3 Genuino

Funcion principal: Ejecutar las órdenes grabadas en su memoria, tomar los datos de temperatura del sensor, mostrar en el display y enviar al servidor Mango, avisar en caso de que la puerta quede abierta.



ILUSTRACIÓN 63-ARDUINO UNO GENUINO

Placa para acceso a Red

Ethernet Shield Ws5100 Mega Compatible, conexión directa sobre Arduino Uno, comunicación SPI.

Función principal: Interfaz de comunicación entre Arduino y Red Ethernet (Mango M2M)

Utiliza los siguiente Pines:

-A0/WP SD

- A1/Detección de SD

- D4/I/O SD

- D10 / SS ~

- D11 / MOSI ~

- D12 / MISO

- D13 / SCK



ILUSTRACIÓN 64-ETHERNET SHIELD W5100 MEGA COMPATIBLE

Display

Shield LCD Keypad 16x2 caracteres y teclado de 5 teclas + reset, montado sobre el Shield Ethernet Ws 5100 se tuvo que modificar, ya que entraba en conflicto con el Shield Ethernet .

Utiliza los siguiente Pines:

-A0/Entrada Analógica de Botones Select, Up, Down, Right y Left - se trata de derivar por software y hardware, a A2 ya que se superpone con Shield Ethernet.

-D4/DB4- se trata de derivar por software y hardware, a D3 ya que se superpone con Shield Ethernet.

-D5/DB5

-D6 /DB6

-D7/DB7

-D8 /RS (Data o Signal Display Select)

-D9 /Enable

-D10/Backlit Control- se trata de derivar por software y hardware, a D2 ya que se superpone con Shield Ethernet.



ILUSTRACIÓN 65-SHIELD LCD KEYPAD 16X2

Sensor de Temperatura

DS18B20 sensor de temperatura digital, con comunicación OneWire

Utiliza los siguiente Pines:

-5V/Vcc

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

-Gnd/Gnd

-A2/Data

-Se agrega una resistencia 4,7Kohms entre Vcc y Data

Al tomar la temperatura dentro de la heladera, se conecta mediante un cable mallado de 120 cm y se asegura por medio de vaina termo contraíble para asegurar la estanqueidad.

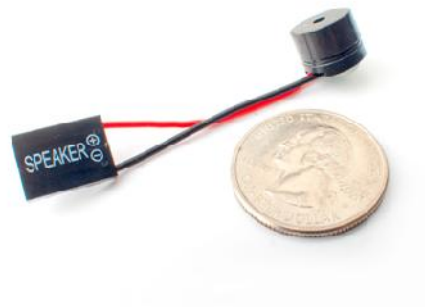
Buzzer

TDPB-1240PE Buzzer sin oscilador interno

. Utiliza los siguiente Pines:

-A5/+Positivo

-Gnd2/-Negativo



Sensor de Apertura de Puerta

SMCB de X28, Sensor micro magnético, miniatura, autoadhesivo, cableado

. Utiliza los siguiente Pines:

-5V/Cable Sensor Magnético

-A4/Cable Sensor Magnético + Resistencia 10K

-Gnd2/Resistencia 10 K

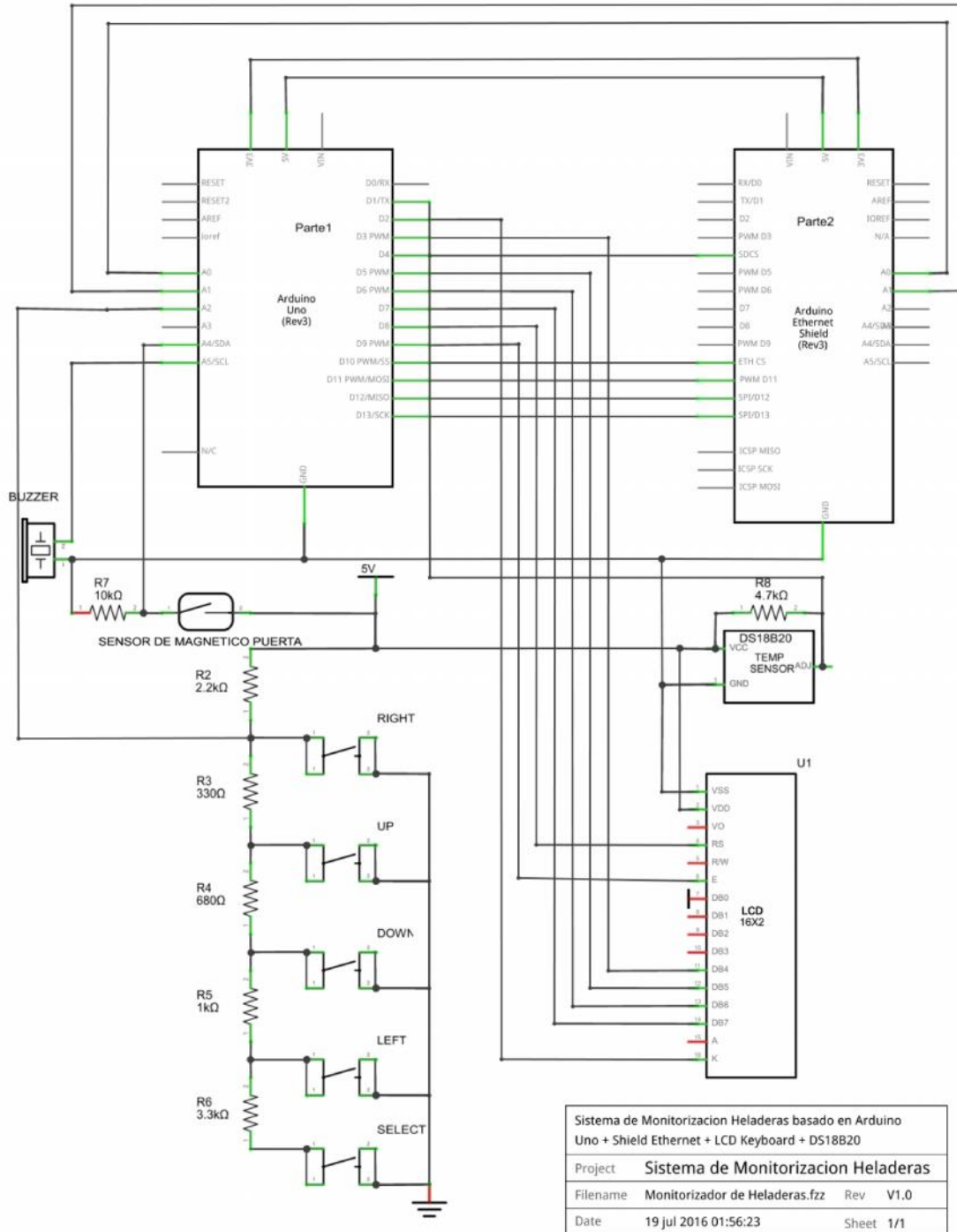


ILUSTRACIÓN 66-SMCB DE X28

Por la característica Modular en el diseño, no necesita de un circuito impreso para ubicar los componente, los modulos son apilables, sobre el Arduino Uno se monta el Shield Ethernet y sobre este el Shield LCD Keyboard.

Esta disposición le da al conjunto una robustez que permite a los periféricos(sensor de temperatura ,buzzer y sensor de puerta) directamente soldar en los conectores del Shield LCD Keyboard

A continuación se muestra el circuito completo de conexión entre todos los componentes



fritzing

ILUSTRACIÓN 67-CIRCUITO COMPLETO DE CONEXIÓN ENTRE TODOS LOS COMPONENTES

4.2-Software

Para poder programar el Arduino Uno, se utiliza el lenguaje de nivel medio basado en C++ sumado a la utilización de librerías específicas para los Shields y Perifericos:

SPI.h y Ethernet.h – Ethernet Shield

OneWire.h y DallasTemperature.h - Sensor de temperatura DS18B20

LiquidCrystal.h – Lcd Keyboard Shield

A continuación explicaremos los parámetros que se deben configurar para el correcto funcionamiento de las mismas en nuestro proyecto

4.2.1-Ethernet.h

Si bien en nuestro proyecto se utiliza como Cliente Ethernet, para futuras aplicaciones también es necesario conocer cómo utilizarlo como Servidor Ethernet. Es una librería compleja, pero ahorra mucho tiempo de programación.

La librería Ethernet es la usada para manejar el Ethernet Shield que implementa la pila de protocolos TCP/IP y dentro de Arduino se implementan los protocolos en la capa de aplicación. La librería se usa entre otras cosas para mandar por Ethernet el protocolo programado en Arduino.

Para manejar el Ethernet Shield deberemos conocer todos los métodos que nos ofrece la librería Ethernet y así poder usarla.

(ARDUINO)

La librería ethernet se compone de 5 clases, cada una con sus métodos

Ethernet Class

Inicializa la librería ethernet y las configuraciones de red.

begin() – Inicializa la librería Ethernet (Constructor)

Sintaxis:

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
```

```
byte ip[] = { 10, 0, 0, 177 };
```

```
byte dns[] = { 8, 8, 8, 8 };
```

```
byte gateway[] = { 10, 0, 0, 1 };
```

```
byte subnet[] = { 255, 255, 255, 0 };
```

```
Ethernet.begin(mac);
```

Respuesta

La versión de esta función DHCP, *Ethernet.begin (mac)*, devuelve un int: 1 en una conexión DHCP éxito, 0 en caso de fallo. Las otras versiones no devuelven nada.

Ejemplo

```
if (Ethernet.begin(mac) == 0) { // inicia Etherner usando DHCP y devuelve 1  
“exito” y 0 “fallo”
```

```
Serial.println("Fallo en configurar Ethernet usando DHCP");
```

```
}
```

```
Ethernet.begin(mac, ip);
```

```
Ethernet.begin(mac, ip, dns);
```

```
Ethernet.begin(mac, ip, dns, gateway);
```

```
Ethernet.begin(mac, ip, dns, gateway, subnet);
```

localIP() – Obtiene la dirección IP. Útil al usar DHCP

Sintaxis y Ejemplo:

// imprime en la consola la direccion IP:

Serial.println(Ethernet.localIP());

maintain() – Solicita una renovación al servidor DHCP

Sintaxis y Ejemplo:

Ethernet.maintain();

Respuesta:

byte:

0: no paso nada

1: renovacion fallo

2: renovacion exitosa

3: rebind fallo

4: rebind exitoso

IPAddress Class

Trabaja con IPs locales y remotas. Facilita el trabajo con direcciones IPs.

IPAddress() – Define una dirección IP

Sintaxis y Ejemplo:

```
Serial.print("IP = ");
```

```
Serial.println(Ethernet.localIP());
```

Server Class

Creará un servidor que puede mandar y recibir datos de los clientes conectados.

Server() – Constructor de la clase server. No se usa directamente

EthernetServer() – Crea un servidor que escucha por las conexiones entrantes del puerto definido.

Syntaxis

```
Server(port);
```

Parametros

port: puerto a escuchar (int)

Ejemplo

```
// telnet defaults to port 23

EthernetServer server = EthernetServer(23);

void setup()

{

    // initialize the ethernet device

    Ethernet.begin(mac, ip, gateway, subnet);

    // start listening for clients

    server.begin();
```



```
}
```

begin() – Le dice al servidor que comience a escuchar.

Sintaxis

```
server.begin()
```

Ejemplo

```
void setup()
{
    // initialize the ethernet device
    Ethernet.begin(mac, ip, gateway, subnet);
    // start listening for clients
    server.begin();
}
```

available() – Devuelve el cliente que está conectado al servidor y tiene datos disponibles a leer.

Sintaxis

```
server.available()
```

Respuesta

un objeto de cliente; Si no hay cliente tiene datos disponibles para su lectura, este objeto se evaluará como falsa en una sentencia if

Ejemplo

```
void loop()
{
    // Si un cliente entrante se conecta, habrá bytes disponibles para leer:
    EthernetClient client = server.available();

    if (client) { // Leer los bytes desde el cliente entrante y escribe de nuevo
        // Para todos los clientes conectados al servidor:
        server.write(client.read()); }
}
```

write() – Escribe datos a todos los cliente conectados al servidor.

Sintaxis

`server.write(val)`

`server.write(buf, len)`

Parametros

val: un valor a enviar como un solo byte (byte o char)

buf: un array para enviar como una serie de bytes (byte o char)

len: la longitud del array

Respuesta

Byte

write () devuelve el número de bytes escritos. No es necesario para leer esto.

Ver Ejemplo anterior

print() – Escribe datos a todos los cliente conectados al servidor.

Sintaxis:

server.print(data)

server.print(data, BASE)

Parámetros

data: los datos a imprimir (char, byte, int, long o de cadena)

BASE (opcional): la base en la que imprimir números: BIN para binario (base 2), DEC para decimal (base 10), OCT para octal (base 8), HEX para hexadecimal (base 16).

Respuesta

byte

print () devolverá el número de bytes escritos, aunque la lectura de ese número es opcional

println() – Escribe datos a todos los cliente conectados al servidor seguido de una nueva línea.

Similar a *print()*.

Client Class

Crea un cliente que se conecta a un servidor y puede mandar y recibir datos.

Client – Constructor de la clase client. No se usa directamente

EthernetClient() – Crea un cliente que se conecta a una determinada IP y puerto

Syntax

EthernetClient()

Ejemplo

```
EthernetClient client;

void setup()
{
    Ethernet.begin(mac, ip);

    Serial.begin(9600);

    Serial.println("connecting...");

    if (client.connect(server, 80)) {

        Serial.println("connected");

        client.println("GET /search?q=arduino HTTP/1.0");

        client.println();

    } else {

        Serial.println("connection failed");

    }

}
```

if (EthernetClient) – Indica si el cliente Ethernet está preparado

Sintaxis

```
if (client)
```

Respuesta

boolean: devuelve verdadero si el cliente especificado está disponible.

connected() – Devuelve si el cliente está o no conectado

Sintaxis

```
client.connected ()
```

Respuesta

Devuelve true si el cliente está conectado, falso en caso contrario

Ejemplo

```
if (client.connected()) {  
  
    Serial.println("connected");  
  
    client.println("GET /search?q=arduino HTTP/1.0");  
  
    client.println();  
  
} else { Serial.println("connection failed");}
```

connect() – Conecta a una IP y puerto especificado. Soporta DNS lookup. Devuelve unos códigos en función del éxito o fallo de la conexión.

Sintaxis

```
client.connect ()
```

client.connect (IP, puerto)

client.connect (URL, puerto)

Parametros

IP: la dirección IP que el cliente se conectará al (serie de 4 bytes)

URL: el nombre de dominio del cliente se conectará al (cadena, ex.:"arduino.cc ")

Puerto: el puerto que el cliente se conectará al (int)

Respuesta

Devuelve un entero (1, -1, -2, -3, -4) que indica el estado de conexión:

ÉXITO 1

TIMED_OUT -1

INVALID_SERVER -2

TRUNCATED -3

INVALID_RESPONSE -4

write() – Escribe datos al servidor al que está conectado.

Sintaxis

client.write (val)

client.write (buf, lon)

Parámetros

val: un valor a enviar como un solo byte (byte o char)

buf: una matriz para enviar como una serie de bytes (byte o char)

len: la longitud de la memoria intermedia

Respuesta

byte

write () devuelve el número de bytes escritos. No es necesario para leer este valor.

print() – Escribe datos al servidor al que está conectado. Escribe los números como una secuencia de dígitos, cada uno un carácter ASCII (por ejemplo, el número 123 es enviada como los tres caracteres '1', '2', '3').

Sintaxis

client.print (datos)

client.print (datos, BASE)

Parámetros

los datos: los datos a imprimir (char, byte, int, long o de cadena)

BASE (opcional): la base en la que imprimir números: DEC durante decimal (base 10), OCT para octal (base 8), HEX para hexadecimal (base 16).

println() – Escribe datos al servidor al que está conectado, seguido de una nueva línea, similar a print()

available() – Devuelve el número de bytes disponibles para leer.

read() – Lee el siguiente byte recibido desde el servidor.

flush() – Borrar todos los bytes que han sido escritos en el cliente pero no leídos

stop() – Desconecta el cliente del servidor

EthernetUDP Class

Habilita el envío y recepción de mensajes UDP

begin() – Inicializar la librería UDP

read() – Lee datos UDP

write() – Escribe datos UDP a la conexión remota.

beginPacket() – Comienza una conexión para escribir paquetes UDP

endPacket() – Finaliza una conexión UDP después de escribir

parsePacket() – Comprueba la presencia de un paquete UDP

available() – Devuelve el nº de bytes disponible para leer en el buffer

stop() – Desconecta del servidor

remoteIP() – Obtiene la IP de la conexión remota

remotePort() – Obtiene el puerto de la conexión remota

4.2.2-OneWire.h

Biblioteca OneWire

OneWire le permite acceder a los dispositivos 1-Wire de Maxim / Dallas, tales como sensores de temperatura y Ibutton. Para los sensores de temperatura, la biblioteca DallasTemperature se puede utilizar con esta biblioteca.

Uso básico

OneWire myWire (pin)

Crear el objeto OneWire, utilizando un pin específico. A pesar de que se puede conectar muchos dispositivos 1-Wire al mismo pin, si usted tiene un gran número, grupos más pequeños cada uno en su propio pin puede ayudar a aislar los problemas de cableado. Se pueden crear varios objetos

OneWire, uno para cada pin.

myWire. search (addrArray)

Buscar el siguiente dispositivo. El addrArray es una matriz de 8 bytes. Si se encuentra un dispositivo, addrArray se llena con la dirección del dispositivo y se devuelve cierto. Si no se encuentran más dispositivos, se devuelve false.

myWire. reset_search ()

Comenzar una nueva búsqueda. El siguiente uso de la búsqueda comenzará en el primer dispositivo.

myWire. reset ()

Restablecer el bus 1-wire. Por lo general, esto es necesario antes de comunicarse con cualquier dispositivo.

myWire. select(addrArray)

Seleccione un dispositivo basado en su dirección. Después de un reinicio, esto es necesario para elegir el dispositivo que va a utilizar, y luego toda la comunicación será con ese dispositivo, hasta que otro reajuste.

myWire. skip ()

Omita la selección del dispositivo. Esto sólo funciona si tiene un único dispositivo, pero se puede evitar la búsqueda y usar esto para avisar inmediatamente acceder a su dispositivo.

myWire. write(num.)

Escribe un byte.

myWire write (num, 1)

Escribe un byte, y dejar potencia aplicada al bus 1 de alambre.

myWire. read ()

Leer un byte.

myWire. CRC8 (dataArray, longitud)

Calcular una comprobación CRC en una matriz de datos.

4.2.3 DallasTemperature.h

Esta biblioteca se deriva de un buen trabajo por la comunidad Arduino. Esta biblioteca es, en su mayor parte, una versión puramente readecuada de la biblioteca existente adaptada para nuestra aplicacion.

Dispositivos soportados

DS18B20

DS1822

DS18S20

DS1820

No se pudo encontrar información sobre los métodos y funciones de esta libreria, pero a continuación los listamos:

Methods and Functions

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

#####

setResolution

getResolution

getTempC

toFahrenheit

getTempF

getTempCByIndex

getTempFByIndex

setWaitForConversion

getWaitForConversion

requestTemperatures

requestTemperaturesByAddress

requestTemperaturesByIndex

isParasitePowerMode

begin

getDeviceCount

getAddress

validAddress

isConnected

readScratchPad

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

writeScratchPad

readPowerSupply

setHighAlarmTemp

setLowAlarmTemp

getHighAlarmTemp

getLowAlarmTemp

resetAlarmSearch

alarmSearch

hasAlarm

toCelsius

processAlarms

setAlarmHandlers

defaultAlarmHandler

calculateTemperature

En el proyecto se utilizo:

-Begin//inicia sensor

-requestTemperatures()//prepara sensor para la lectura

-getTempCByIndex()//lee la temperatura y devuelve el valor

4.2.3- LiquidCrystal.h

Esta biblioteca permite a una placa Arduino para controlar las pantallas LiquidCrystal (LCD) en base a la Hitachi HD44780 (o compatible) chipset, que se encuentra en la mayoría de las pantallas LCD basados en texto. La biblioteca trabaja con tanto en el modo de 4 u 8 bits (es decir, utilizando 4 u 8 líneas de datos, además de los rs, enable y, opcionalmente, las líneas de control rw).

Estas son las funciones de la librería LiquidCrystal:

LiquidCrystal ()

Descripción

Crea una variable de tipo LiquidCrystal. La pantalla se puede controlar por medio de 4 u 8 líneas de datos. En el primer caso, omitir los números de patas D0 a D3 y dejar esas líneas sin conectar. El pin RW puede estar vinculada a tierra en lugar de conectarse a un pin en el Arduino; si es así, omite de los parámetros de esta función.

Sintaxis

LiquidCrystal (RS, enable, d4, d5, d6, d7)

LiquidCrystal (RS, RW, enable, d4, d5, d6, d7)

LiquidCrystal (RS, enable, d0, d1, d2, d3, d4, d5, d6, d7)

LiquidCrystal (RS, RW, enable, d0, d1, d2, d3, d4, d5, d6, d7)

Parámetros

rs: el número de la clavija de Arduino que está conectado a la clavija de RS en la pantalla LCD

rw: el número de la clavija de Arduino que está conectado a la clavija de RW en la pantalla LCD (opcional)

enable: el número de la clavija de Arduino que está conectado a la clavija de habilitar la pantalla LCD

d0, d1, d2, d3, d4, d5, d6, d7: los números de los pasadores Arduino que están conectados a los correspondientes pasadores de datos en la pantalla LCD. *d0, d1, d2, d3* y son opcionales; Si se omite, la pantalla LCD se puede controlar utilizando sólo las cuatro líneas de datos (*d4, d5, d6, d7*).

begin()

Descripción

Inicializa la interfaz de la pantalla LCD, y especifica las dimensiones (anchura y altura) de la pantalla. *begin()* debe ser llamada antes de cualquier otro comando de la biblioteca LCD.

Sintaxis

lcd.begin(cols, filas)

Parámetros

lcd: una variable de tipo LiquidCrystal

cols: el número de columnas que tiene la pantalla

filas: el número de filas que la pantalla tiene

clear()

Descripción

Borra la pantalla LCD y posiciona el cursor en la esquina superior izquierda.

Sintaxis

lcd .clear()

Parámetros

lcd: una variable de tipo LiquidCrystal

home()

Descripción

Pone el cursor en la esquina superior izquierda de la pantalla LCD. Es decir, utilizar esa ubicación en la salida de texto subsiguiente a la pantalla. Para borrar también la pantalla, utilice la clara () función en lugar.

Sintaxis

lcd .home ()

Parámetros

lcd: una variable de tipo LiquidCrystal

setCursor ()

Descripción

Coloque el cursor del LCD; es decir, establecer la ubicación en la que se mostrará el texto escrita en el LCD.

Sintaxis

lcd .setCursor (col, fila)

Parámetros

lcd: una variable de tipo LiquidCrystal

col: la columna en la que para posicionar el cursor (siendo 0 la primera columna)

fila: la fila en la que para posicionar el cursor (siendo 0 la primera fila)

write()

Descripción

Escribir un personaje para la pantalla LCD.

Sintaxis

lcd.write(data)

Parámetros

lcd: una variable de tipo LiquidCrystal

datos: el personaje va a escribir en la pantalla

print ()

Descripción

Imprime el texto de la pantalla LCD.

Sintaxis

lcd.print (datos)

lcd.print (datos, BASE)

Parámetros

lcd: una variable de tipo LiquidCrystal

datos: los datos a imprimir (char, byte, int, long o de cadena)

BASE (opcional): la base en la que imprimir números: BIN para binario (base 2), DEC para decimal (base 10), OCT para octal (base 8), HEX para hexadecimal (base 16).

El resto de las funciones no son utilizadas en el proyecto, por lo que las nombraremos:

cursor()

noCursor()

blink()

noBlink()

display()

noDisplay()

scrollDisplayLeft()

scrollDisplayRight()

autoscroll()

noAutoscroll()

leftToRight()

rightToLeft()

createChar()

4.3-Diseño software

Las variables, registros y objetos utilizados en el programa son las siguientes:

Tabla 4-Variables, Registros y Objetos en el Sketch del Programa

Variable u Objeto	Tipo	Valor Inicial	Función
<i>Pin</i>	set I/O	2	Pin de conexión datos de DS18B20
<i>previousMillis</i>	unsigned long	0	Almacenar el tiempo cuando se dispara la condición de alarma
<i>alarmaMillis</i>	unsigned long	60000	Retardo para que suene nuevamente si se da la condición de alarma
<i>currentMillis</i>	unsigned long	-	Variable en la que se almacena el tiempo desde que esta encendido el procesador
<i>mudo</i>	boolean	false	Variable para deshabilitar alarma
<i>ourWire</i>	1-Wire	Pin	Objeto 1-Wire
<i>sensors</i>	DallasTemperature	ourWire	Objeto DallasTemperature
<i>temperatura</i>	float	-	Almacena el valor de temperatura
<i>tempmin</i>	float	2	Seteo de temperatura minima
<i>tempmax</i>	float	8	Seteo de temperatura minima
<i>var</i>	int	-	Variable utilizada en la rutina de buzzer
<i>lcd</i>	LiquidCrystal	8,9,4,5,6,7	Seteo de Objeto LCD

<i>mac</i>	byte	0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED	Dirección de MAC
<i>ip</i>	byte	192.168.1.113	Dirección IP
<i>server</i>	byte	192.168.1. 103	Dirección del Servidor Mango M2M
<i>client</i>	EthernetClient	-	Prepara la Placa Ethernet como Cliente

4.3.1- Algoritmo de Inicialización (Setup)

Define el orden en que se establecen los valores iniciales del sistema y la configuración de periféricos; se ejecuta cada vez que se inicia el sistema y cuando ocurra un reset de usuario mediante el botón destinado a este uso.

El algoritmo es el siguiente:

- Configura la cantidad de líneas y columna del display LCD.
- Muestra la leyenda “INICIO.....” y suena el Buzzer para comprobar funcionamiento.
- Configura Puerto Serie.
- Inicia el Sensor de temperatura.
- Configura la dirección IP.
- Muestra la leyenda “TEMPERAT.” en la fila superior y “T1:” en la fila inferior.

4.3.2- Algoritmo de Ejecución Cíclica (Loop)

Se ejecuta de forma cíclica en forma infinita, lo que posibilita que el programa esté respondiendo continuamente ante los eventos que se produzcan en la placa.

El algoritmo es el siguiente:

-Lee la temperatura y la muestra en el display.

-Intenta conectarse al Servidor Mango M2M.

-Si la conexión es posible mostrara los caracteres “~~” en la esquina inferior.

Enviara la cadena de caracteres ”POST <http://192.168.1.103/https?testKey=>”+
Valor de temperatura.

Finaliza el envio y muestra los caractetes “—“.

-Si la conexión no se pudo realiza mostrara los caracteres “XX” en la esquina inferior.

-Si la temperatura es menor a la temperatura minima, y si la variable mudo es falsa, suena la buzzer, indica en el lcd “BAJA”, sigue la rutina de programa.

-Si la temperatura es mayor a la temperatura maxima, y si la variable mudo es falsa, suena la buzzer, indica en el lcd “ALTA”, sigue la rutina de programa.

-Si no se cumple estas dos condiciones el display indica “NORMAL”

-Si el botón “SELECT” esta presionado, la variable mudo cambia a verdadera, por lo cual anula la condición de ALTA y BAJA por 1 minuto.

4.3.4-Diagrama de Flujo

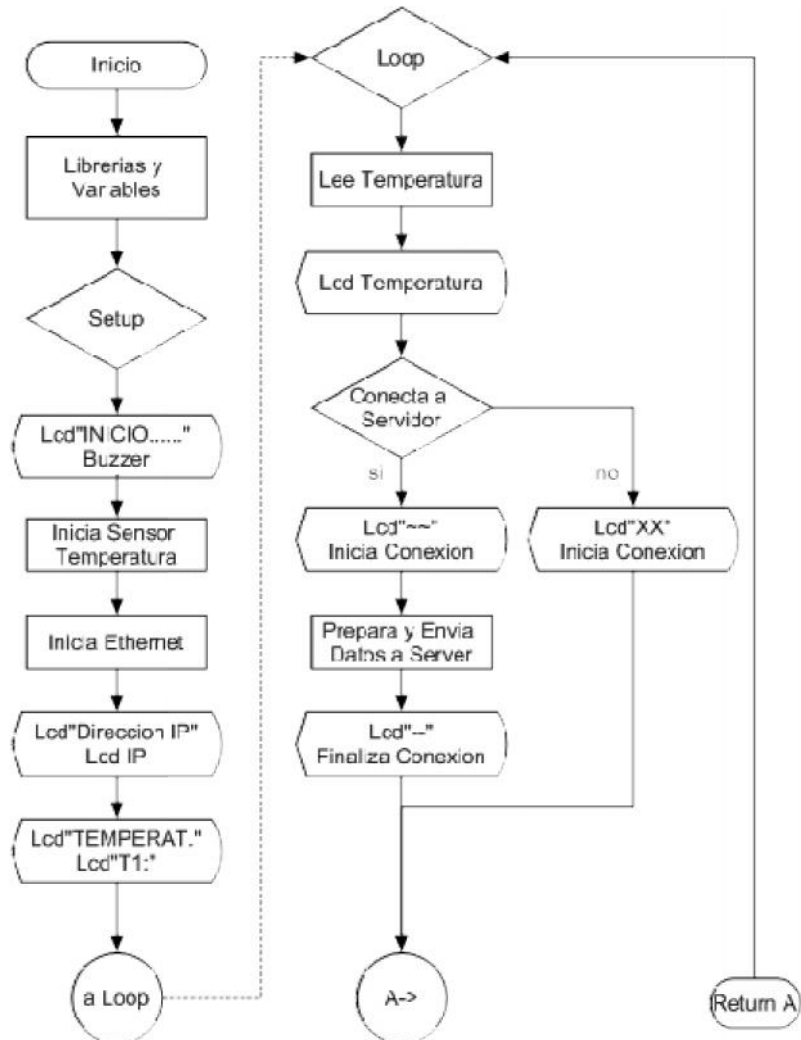


ILUSTRACIÓN 68-DIAGRAMA DE FLUJO 1

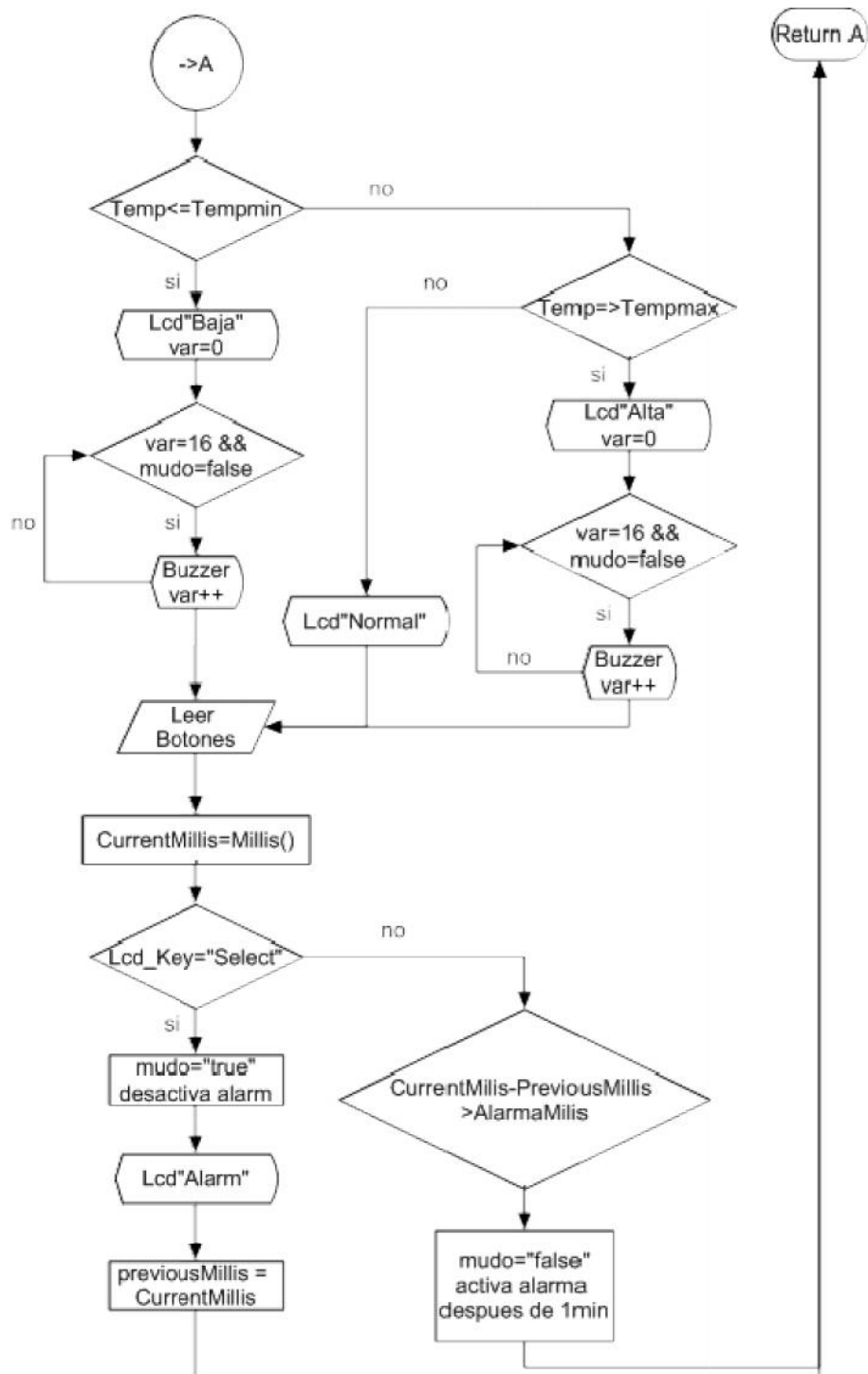


ILUSTRACIÓN 69-DIAGRAMA DE FLUJO 2

4.4-Codigo final de Proyecto

```

#include <SPI.h>
#include <Ethernet.h>
#include <OneWire.h> //Se importan las librerías
#include <DallasTemperature.h>
#include <LiquidCrystal.h>

#define Pin 17 //Se declara el pin donde se conectará la DATA
const int sensorPin = 18; // Nuestro sensor en la puerta
int estadoBoton = 1,ultimoBoton;
int Puerta=0;

unsigned long previousMillis = 0, alarmamillis = 60000, currentMillis,
previusPuerta=0,alarmaPuerta=10000,currentPuerta;
boolean mudo=false, bandera=true;
OneWire ourWire(Pin); //Se establece el pin declarado como bus para la comunicación OneWire
DallasTemperature sensors(&ourWire); //Se instancia la librería DallasTemperature
float temperatura, tempmin=2, tempmax=8;
int var;
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192,168,1,113 };
char server[]="192.168.1.116"; // Direccion IP del servidor
char thingSpeakAddress[] = "api.thingspeak.com";
String writeAPIKey = "1MWZSW295Y8N9TA9";

EthernetClient client ;

String tsData, medicion;
// define some values used by the panel and buttons
int lcd_key = 2;
int key = 0;
#define btnRIGHT 0
#define btnUP 1
#define btnDOWN 2
#define btnLEFT 3
#define btnSELECT 4
#define btnNONE 5

//Sirena
int duracion=50; //Duración del sonido
int fMin=2000; //Frecuencia más baja que queremos emitir
int fMed=3000; //Frecuencia más alta que queremos emitir
int fMax=4000; //Frecuencia más alta que queremos emitir
int i=0;
int j=0;

void setup()
{
  lcd.begin(16, 2);
  lcd.print("INICIO");
  var = 0;
  lcd.setCursor(0,1);
  while(var < 16){
    // do something repetitive 200 times
    lcd.print(".");
    var++;
    tone(19,800,250); // utilizo entrada A5 como salida digital
    delay(250);
  }
  sensors.begin(); //Se inician los sensores
  Serial.begin(9600);
  while (!Serial) ; // Retraso para el Leonardo
  lcd.setCursor(0,0);
  lcd.print("OBTENIENDO IP:");

```

4

```

//Ethernet.begin(mac, ip);
Ethernet.begin(mac);
//server.begin();
Serial.print("Servidor Web en la direccion: ");
Serial.println(Ethernet.localIP());
  lcd.setCursor(0,0);
  lcd.print("DIRECCION IP:  ");
  lcd.setCursor(0,1);
  lcd.print(Ethernet.localIP());
  delay(2000);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("TEMPERAT.");
  lcd.setCursor(0,1);
  lcd.print("T1:");
  //pinMode(18, INPUT);
}

void loop()
{
  estadoBoton = digitalRead(sensorPin);

  // compara el estado del pulsador con el estado previo del mismo
  if ( estadoBoton != ultimoBoton)
  {
    // si el estado cambia, el contador pasa a 1
    if ( estadoBoton == LOW)
    {
      Serial.println("Puerta Abierta");
      lcd.setCursor(0,0);
      lcd.print("Puerta Abierta!!");
      tone(19,2000,1000); // utilizo entrada A5 como salida digital
      delay(1000);
      delay(1000);
      Puerta=1;
    }
    else
      Puerta=0;
      Serial.println("Puerta OK");
      bandera=true;
  }
  else
    currentPuerta = millis();
    if ( (estadoBoton == LOW) && (bandera== true))
    {
      previusPuerta = currentPuerta;
      bandera=false;
    }
  if ((estadoBoton == LOW) &&(currentPuerta - previusPuerta > alarmaPuerta))
  {
    Puerta++;
    bandera=false;
    Serial.print("Puerta Abierta ");
    Serial.println(Puerta);
    lcd.setCursor(0,0);
    lcd.print("Puerta Abierta!!");
    lcd.setCursor(11,1);
    lcd.print(Puerta);
    lcd.setCursor(13,1);
    lcd.print("Min");
    tone(19,2000,1000); // utilizo entrada A5 como salida digital
    previusPuerta = currentPuerta;
  }

  // guarda el estado actual como el ultimo estado,
  ultimoBoton = estadoBoton;
}

```



```

    sensors.requestTemperatures(); //Prepara el sensor para la lectura
    lcd.setCursor(4,1);
    temperatura = sensors.getTempCByIndex(0);
    lcd.print(temperatura);
    boolean enviado= false;
    if ((client.connect(server, 8080)))
    {
        Serial.println("connected");
        lcd.setCursor(14,1);
        lcd.print("~");
        delay(250);
        String servidor= server;
        String stringUno = "POST http://" + servidor + "/https?temp=";
        String stringMango = stringUno + temperatura + "&puerta="+Puerta;
        Serial.println(stringMango);
        client.println(stringMango);
        client.println();
        Serial.println("Conectado");
        delay(250);
        client.stop();
        enviado=true;
        Serial.println("finalizo envio");
        lcd.setCursor(14,1);
        lcd.print("-");
        delay(250);
        lcd.setCursor(14,1);
        lcd.print(" ");
    }
    else
    {
        Serial.println("Fallo en la conexion Mango M2M");
        client.stop();
        enviado=false;
        lcd.setCursor(14,1);

        lcd.print("X");
        delay(250);
    }
    if (!client.connected()) {
        Serial.println("Desconectado!");
    }
    client.stop();
    client.flush();
    medicion=String(temperatura);
    tsData=String("1=" + medicion + "2=" + Puerta);
    if (client.connect(thingSpeakAddress, 80))
    { // Conexion con el servidor
        //Display in Serial Monitor
        Serial.print(tsData+"\n"); //Return temperature to Monitor
        // Proceso de envio de muestras al servidor
        Serial.println("Conectando...");
        Serial.println("Enviando a Thingspeak...");
        lcd.setCursor(15,1);
        lcd.print("~");
        delay(250);
        client.print("POST /update HTTP/1.1\n");
        client.print("Host: api.thingspeak.com\n");
        client.print("Connection: close\n");
        client.print("X-THINGSPEAKAPIKEY: "+writeAPIKey+"\n");
        client.print("Content-Type: application/x-www-form-urlencoded\n");
        client.print("Content-Length: ");
        client.print(tsData.length());
        client.print("\n\n");
        client.print(tsData);
        Serial.println("Conectado");
        client.stop();
    }

```

```
        enviado=true;
        Serial.println("finalizo envio");
        lcd.setCursor(15,1);
        lcd.print("-");
        delay(250);
        lcd.setCursor(15,1);
        lcd.print(" ");
    }
    else
    {
        Serial.println("Fallo en la conexion Thingspeak");
        client.stop();
        enviado=false;
        lcd.setCursor(15,1);
        lcd.print("X");
        delay(250);
    }
    if (!client.connected()) {
        Serial.println("Desconectado!");
    }
    client.stop();
    client.flush();

    if (temperatura <= tempmin)
    {
        lcd.setCursor(0,0);
        lcd.print("TEMPERAT. BAJA ");
        var=0;
        while(var < 10 && mudo == false){
            // do something repetitive 200 times
            var++;
            for (i=fMed;i<=fMax; i++)
                tone(19, i, duracion);

            for (i=fMed;i>=fMin; i--)
                tone(19, i, duracion);
        }
    }
    else if (temperatura >= tempmax)
    {
        lcd.setCursor(0,0);
        lcd.print("TEMPERAT. ALTA ");
        var=0;
        while(var < 10 && mudo == false){
            // do something repetitive 200 times
            var++;
            for (i=fMin;i<=fMax; i++)
                tone(19, i, duracion);
            //sonido más grave
            for (i=fMax;i>=fMin; i--)
                tone(19, i, duracion);
        }
    }
    else
    {
        lcd.setCursor(0,0);
        lcd.print("TEMPERAT. NORMAL");
        lcd.setCursor(11,1);
        lcd.print(" ");
    }
    lcd_key = read_LCD_buttons();
    currentMillis = millis(); // Se toma el tiempo actual
    if( lcd_key == btnSELECT)
    {
        mudo = true;
    }
}
```

```
    lcd.setCursor(0,0);
    lcd.print("ALARM RECONOCIDA");

    tone(19,2000,1000); // utilizo entrada A5 como salida digital

    previousMillis = currentMillis;
    // lcd.setCursor(0,0);
    // lcd.print("TEMPERAT.    ");
}
else if (currentMillis - previousMillis > alarmamillis)
{
    mucdo = false;
}

/*
  if( lcd_key == btnRIGHT)
    lcd.print("RIGHT ");
  else if ( lcd_key == btnLEFT )
    lcd.print("LEFT  ");
  else if ( lcd_key == btnJE)
    lcd.print("UP   ");
  else if ( lcd_key == btnDCWN)
    lcd.print("DOWN ");
  else if ( lcd_key == btnSELECT)
    lcd.print("SELECT");
  else if ( lcd_key == btnNCNE)
    lcd.print("NONE  ");
  */
}

int read_LCD_buttons() // Lee los botones y devuelve le valor pulsado
{
  key = analogRead(0); // Leemos la puerta A0
  // Mis botones dan: 0, 145, 329,507,743 ; Comprueba que tu shield devuelve valores parecidos

  // Y ahora los comparamos con un margen comodo

  if (key > 1000) return btnNONE; //Si no se pulsa nada salimos (el 99% de las veces)
  if (key < 50)   return btnRIGHT;
  if (key < 250)  return btnUP;
  if (key < 450)  return btnDOWN;
  if (key < 650)  return btnLEFT;
  if (key < 850)  return btnSELECT;

  return btnNONE; // when all others fail, return this...
}
```

Capitulo 5 - Experiencia, pruebas y resultados

5.1-Experiencia en el desarrollo del Sistema

Desarrollo del Sistema

Inicialmente se implementaron pequeños programas con Arduino Uno y su plataforma de programación tales como:

- Encender y apagar Led.
- Seteo y prueba de puertos I/O.
- Interaccion con el controlador por medio de Consola Serie(Comunicacion entre PC y Puerto Serie de Arduino)
- Manejo de cadenas de caracteres.
- Operaciones Matematicas.
- Manejo de Display LCD.
- Librerias I/O y Shields.

Una vez que nos afianzamos en el manejo del entorno de programación, los puertos I/O y Shield, se probó implementar el protocolo de comunicación MODBUS en el Arduino con el Ethernet Shield con ENC28J60, para poder comunicarnos por este protocolo con el SCADA Mango M2M.

Para realizar la comunicación, se deben mandar paquetes de datos según el protocolo MODBUS TCP/IP, para realizar esta tarea de mandar paquetes se puede utilizar la librería "etherShield.h" y "ETHER_28J60.h" que son las librerías para utilizar Ethernet Shield con ENC28J60. Para utilizar correctamente esta librería, debemos asignar una MAC, IP y la máscara de Subred al Arduino, para que se puede conectar a la red.

Con esta librería podemos generar un pagina web o mandar datos, alojados en el mismo programa cargado en el Arduino.

Con los ejemplos que trae la libreria realizamos la siguientes experiencias:

-EtherShield_ping.pde : pudimos realizar la comprobación de comunicación con el Arduino, el mismo consta del envio de un paquetes de Protocolo de Mensajes de Control de Internet o ICMP, de solicitud y de respuesta, para identificar el numero de Ip del Arduino.

-EtherShield_web_temperature.pde: el ejemplo viene preparado para utilizar el sensor de temperatura 1-wire DS18B20, se modifiko para utilizar el sensor LM35 y el convesor A/D.

Se encontraron la librería para comunicación MOBUS TCP en Arduino, “Mudbus.h”, pero al implementarla, nos encontramos con el problema de que estaba preparada para el Ethernet Shield con W5100, el cual en ese momento no disponíamos del mismo.

Con el shield ENC28J60 la implementación de TCP/IP debe llevarse a cabo en el microcontrolador, ya que el ENC28J60 no tiene stack TCP/IP por hardware. Por esta razón se recomienda un microcontrolador host con suficiente memoria de programa (Flash) y RAM para albergar el stack y la aplicación de usuario.

Si se está utilizando Arduino,es recomendable utilizar el ethernet shield en lugar de este módulo, ya que en este la implementación de TCP/IP se lleva a cabo en el hardware del Shield, ahorrando valioso espacio de memoria RAM y Flash en el microcontrolador principal.

La diferencia de las librerías Ethernet con el ENC28J60 y el W5100,son muy notorias, no tienen equivalencias entre si, por lo cual se opto por conseguir Ethernet Shield con W5100, más moderna que la anterior, sumado a que integrado un lector de memoria MicroSD, la cual podemos intento utilizar para almacenar datos, pero fallo el modulo Micro SD y quedo inutilizado, ya que se superpuso con modulo LCD que también compartía una salida digital.

Pudimos adquirir Arduino Uno junto con Ethernet Shield Micro-Sd W 5100, con lo cual iniciamos la pruebas de comunicaciones que habíamos realizado con Ethernet Shield con ENC28J60.

5.2-Mango M2M (Metodo Post y Get)

En el medio del desarrollo se encontró una alternativa al protocolo de comunicación, al MODBUS TCP/IP.

El Servidor MANGO M2M puede aceptar los siguientes protocolos de comunicación:

- BACnet I/P

- Modbus (ASCII, RTU, TCP, and UDP),

- OPC DA

-1-wire

- SNMP

-SQL

- HTTP

- POP3

- NMEA 0183

- MBus

- DNP3

- OpenV

- webcams

- vmstat

Dentro de estos protocolos admite HTTP, por medio de los métodos GET y POST

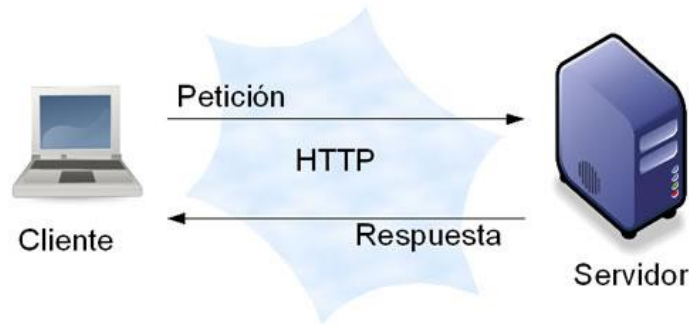


ILUSTRACIÓN 70-COMUNICACION CLIENTE SERVIDOR

Podemos pasar los datos directamente al Servidor, indicando el método, la variable de destino, y el valor de la misma. Y establecer que reciba datos de un cliente en particular.

Utilizando este método se realizó el primer registro con el sistema.

5.3-Experiencia con LM35 como sensor de temperatura.

La experiencia se desarrolló la noche del 12 de Abril 2012, desde la 1:50am a 7:20 am, registrando la temperatura ambiente, cerca de una ventana, para apreciar mejor la variación de la temperatura. Se registraba la temperatura con un sensor LM35 conectado al Arduino Uno +Ethernet Shield W5100, el mismo registraba cada 10 segundos la temperatura ambiente y la enviaba al Servidor Mango por medio del protocolo HTTP y el método GET, este servidor lo almacena en la variable "testKey".

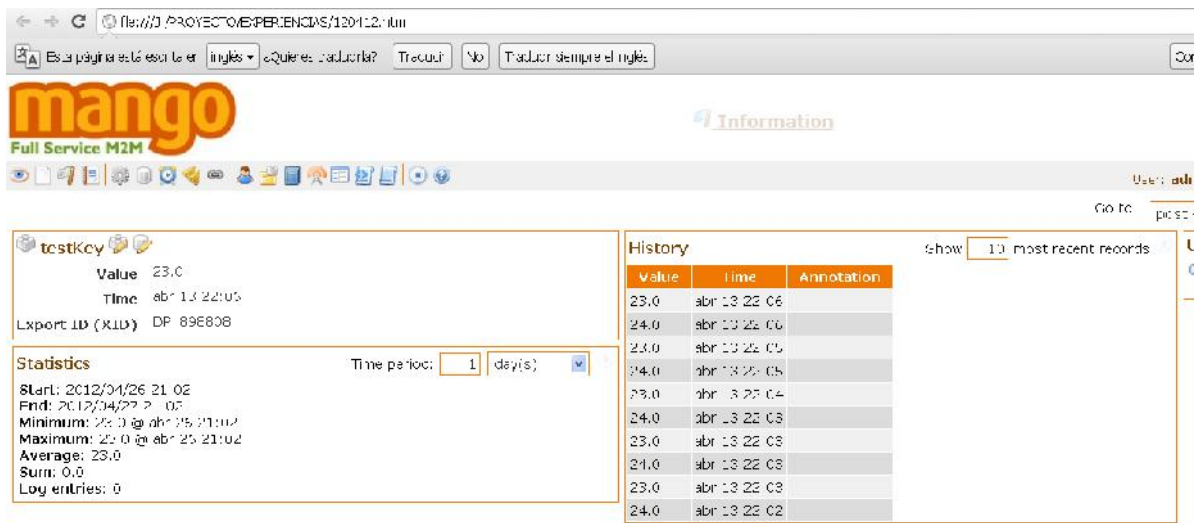


ILUSTRACIÓN 71-REGISTRO CON MANGO M2M

Luego de 5h y 35m registrando en forma continua, se obtuvieron 1175 datos de temperatura y hora, en los cuales se observa la disminución de la temperatura de 17° a 14° y algunos saltos de temperatura, que suponemos que se deben al intercambio entre las masa de aire caliente del interior y la masa de aire frio del exterior, a causa de la corriente de aire, ya que la ventana donde estaba el sensor estaba abierta y el sensor no tenia ningún elemento que sirviera para tener una inercia térmica mayor.



ILUSTRACIÓN 72-GRAFICO OBTENDIDO CON M2M

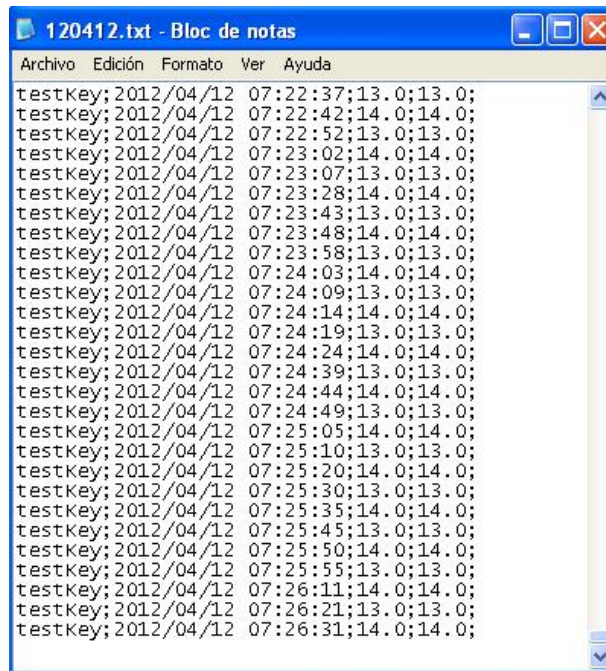


ILUSTRACIÓN 73-DATOS EXPORTADOS A TXT

Al realizar experiencia con temperaturas del rango de las de almacenamiento de vacunas nos encontramos con el problema de que si utilizamos la tensión de referencia de 5 V del Arduino, teniendo en cuenta que el LM35, por cada °C corresponde 10mV, al leer 2°C, el conversor A/D del Arduino tendría que leer 20mV, con una resolución de 10 bit ($5V/1024=5mV$).

Al ser tensiones muy bajas, puede interferir el ruido eléctrico propio de motor del compresor de la heladera, sumando alrededor de 20mV, con lo cual nuestro sistema tenía un error de +2°C.

Este problema se solución agregando un Amplificador Operacional, de forma de aumentar la tensión entregada por el sensor de temperatura.

Inicialmente el diseño funciono, mejoraba el funcionamiento, pero también amplificaba el ruido del sistema.

Por lo tanto se opto por utilizar un sensor de temperatura digital, el DS18B20, con el bus de datos 1-Wire.

5.4-Experiencia con el DS18B20 como sensor de temperatura

La experiencia se desarrollo la noche del 4 de Abril 2016, desde la 2:08am a 9:38 am , registrando la temperatura ambiente, cerca de una ventana, de igual forma que la experiencia anterior, para apreciar mejor la variación de la temperatura. Se registraba la temperatura con un sensor DS18B20 conectado al Arduino Uno +Ethenet Shield W5100, el mismo registraba cada 5 segundo la temperatura ambiente y la enviaba al Servidor Mango por medio del protocolo HTTP y el método GET, este servidor lo almacena en la variable TEMPERATURA”.



ILUSTRACIÓN 74-GRAFICO OBTENIDO CON SENSOR DS18B20

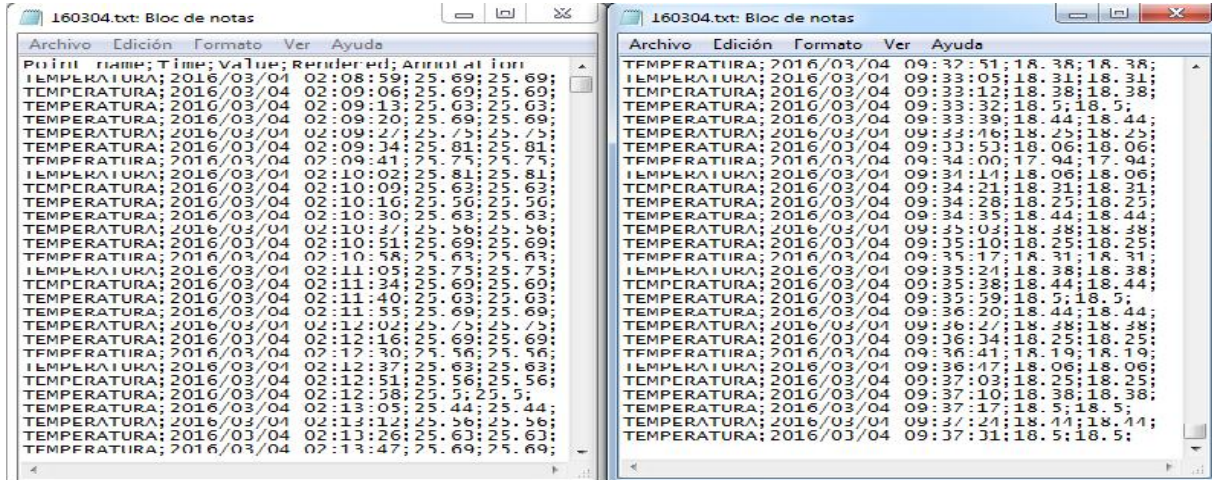


ILUSTRACIÓN 75-ARCHIVO DE DATOS CON DS18B20 EXPORTADO A TXT

Comparando los dos registros, observamos que el sensor DS18B20 tiene un mejor desempeño que el LM35, para nuestra aplicación, ya que el primero no tiene tantas lecturas con saltos de temperatura, en el caso del segundo, se pueden observar saltos de +/- 2°C.

La diferencia está dada por el hecho que el LM35 al ser de lectura analógica está expuesto al ruido ambiente, en cambio al DS18B20, tiene el convertor A/D dentro del circuito integrado.

En el diseño final se optó por el DS18B20, ya que es mucho más inmune al ruido eléctrico y con el valor agregado de que se pueden conectar muchos sensores, con lo cual se pueden monitorizar más puntos con un mismo hardware.

5.5-Experiencia con ThinkSpeak

Para utilizarlo, se creó una cuenta y se creó un canal, identificado como id:141159 (número que genera automáticamente la página Web) con el nombre "Temperatura Vacunas" y dos campos, Field 1: Temperatura y Field 2: Puerta.

En estos campos se almacenan los datos de Temperatura y las aperturas de Puerta.

El canal generado nos proporciona un clave de escritura de API en nuestro caso

“1MWZSW295Y8N9TA9”

Para poder escribir los datos, se debe enviar por Http la siguiente cadena de caracteres :

PUT https://api.thingspeak.com/channels/141159

api_key=1MWZSW295Y8N9TA9

name=Updated Channel

La pagina queda de esta manera

The screenshot shows the ThingSpeak interface. At the top is a blue navigation bar with the ThingSpeak logo and menu items: Channels, Apps, Blog, and Support. Below the navigation bar, the main heading is 'TEMPERATURA VACUNAS'. Underneath, there are details for the channel: Channel ID: 141159, Author: LUKBEQUE, and Access: Private. There are four tabs: Private View, Public View, Channel Settings (which is selected), API Keys, and Data Import / Export. The 'Channel Settings' section is expanded, showing a progress indicator for 'Percentage complete' at 50%. Below this, there are input fields for 'Channel ID' (141159), 'Name' (TEMPERATURA VACUNAS), and 'Description' (TEMPERATURA VACUNAS). There are also two 'Field' settings: 'Field 1' with the value 'TEMPERATURA' and a checked checkbox, and 'Field 2' with the value 'PUERTA' and a checked checkbox. On the right side, there is a 'Help' section with the heading 'ThingSpeak' and a paragraph of text, and a 'Channel Settings' section with a list of items: Channel Name, Description, and Field#: Check.

ILUSTRACIÓN 76-CAPTURA THINKSPEAK CANAL

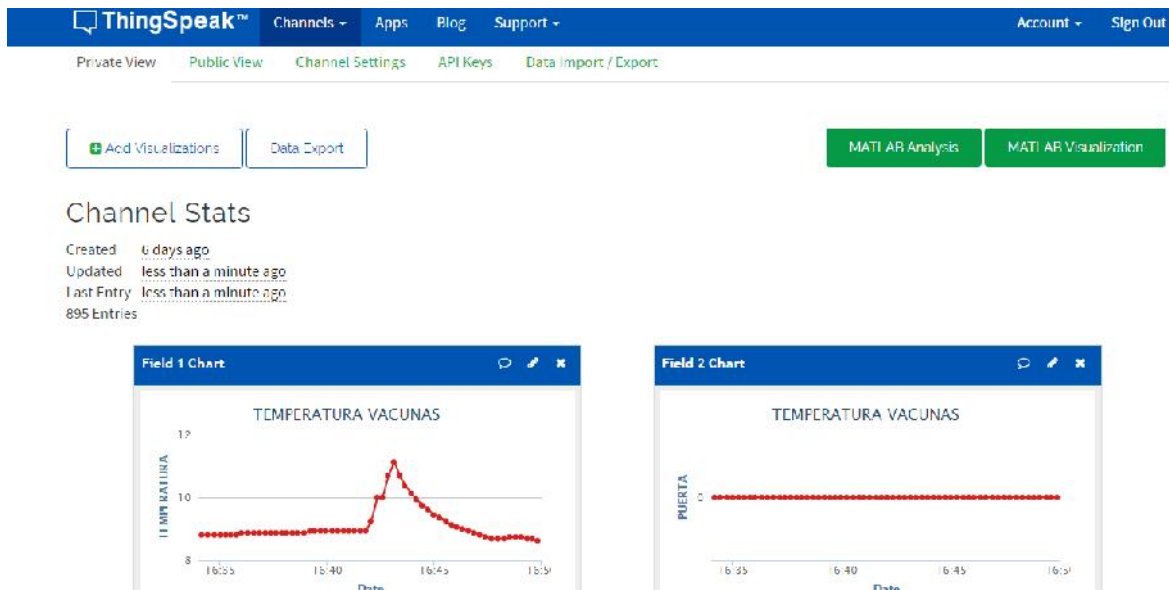


ILUSTRACIÓN 77-GRAFICOS DEL CANAL

De esta forma tan simple se pasan los datos a ThinkSpeak

5.6-Experiencia con IoT ThingSpeak Monitor Widget

Se instalo en un celular con Android y se configuro:

- Colocando en “Channel Id:”141159 y “Read API Key:” 1MWZSW295Y8N9TA9,
- Indicamos que el primer campo a a actualizar en “First field and alert setting “
“Field ID(1):”1 –Temperatura .
- Seteamos los valores de alarma máxima (8°C) tildando en Upper threshold exceeded Alert y Upper threshold value con el valor :8
- Seteamos los valores de alarma mínima (2°C) tildando en Lower threshold exceeded Alert y Lower threshold value con el valor :2
- Habilitamos el sonido insistente de notificación “Insistent sound notification”
- Indicamos que el segundo campo a a actualizar en “Second field and alert setting “
“Field ID(2):”2 –Puerta .
- Seteamos los valores de alarma máxima (mayor 2 implica que la puerta permanece abierta por mas de 2 minutos) tildando en Upper threshold exceeded Alert y Upper threshold value con el valor :2

-Habilitamos el sonido insistente de notificación “Insistent sound notification”

A continuación se muestran las capturas de pantalla del este widget funcionando

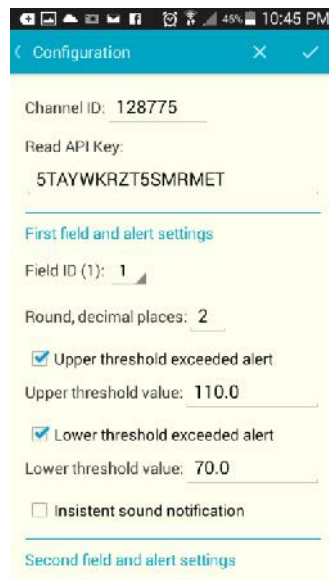


ILUSTRACIÓN 78-CONFIGURACION WIDGET



ILUSTRACIÓN 79-PANTALLA PRINCIPAL CON WIDGET

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN



ILUSTRACIÓN 80-ALARMA EN BARRA DE NOTIFICACIONES

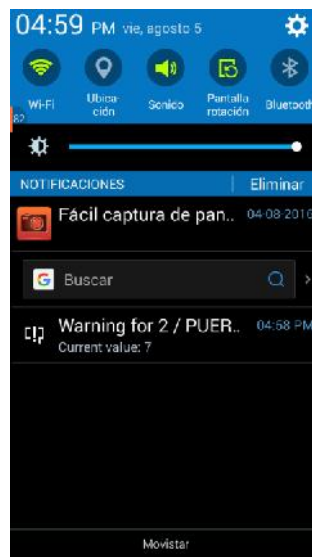


ILUSTRACIÓN 81-RECONOCIENDO ALARMA

Proyecto Final:

Sistema Scada para monitorizar Heladeras para Vacunas del Hospital y Centros de Salud

Alumno: Chandia Pablo Lucas

UTN-FRN

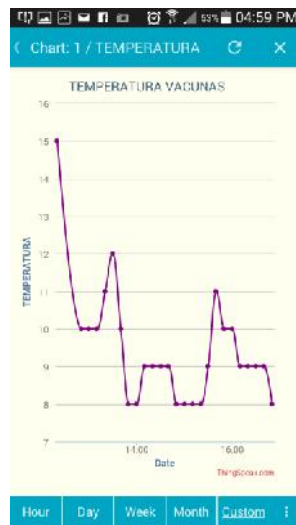


ILUSTRACIÓN 82-GRAFICO DE TEMPERATURA

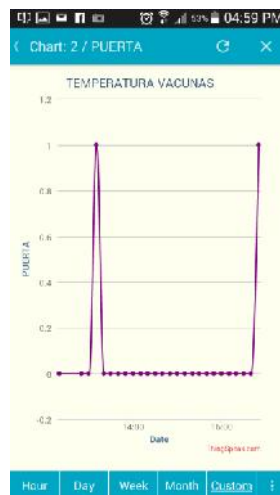


ILUSTRACIÓN 83-GRAFICO DE PUERTA

Capítulo 6-Conclusiones

El presente trabajo de investigación describe el diseño y construcción de un Sistema Scada para monitorizar Heladeras para Vacunas, se logro desarrollar un prototipo funcional, que cumple con los objetivos de monitorización remota, ya sea por medio de una Pc o mediante un Smartphone.

Del desarrollo de este trabajo se obtienen las siguientes conclusiones:

-El sistema de monitorización sirve para monitorizar la temperatura de heladeras y avisa si la puerta quedo mal cerrada o se alcanzaron temperaturas máximas y minimas.

-Con los datos obtenidos por medio del sistema, se puede realizar registros de temperatura e indirectamente la cantidad aproximadas de vacunas realiza, ya que cada apertura corresponde a una vacuna aplicada.

-El sistema puede ser utilizado para registrar temperaturas de -10 a 85°C, por lo cual puede ser aplicado para monitorizar procesos tales como pueden ser sistemas de calefacción o calderas.

-La plataforma puede servir para registrar con otro tipos de sensores otros fenómenos, sin demasiadas modificaciones, por ejemplo niveles de tanques, vibraciones, caudales, etc.

-El costo de materiales del sistema no superan los \$1000, por lo cual es un sistema muy accesible.

-Puede ofrecerse como producto con la opción de vender el servicio de monitorización en forma anual o mensual.

-Por medio de Thinkspeak, puede servir para monitorizar productos domesticos tales como lavarropas, Niveles de Tanque de Agua, Temperatura Ambiente y obtener los datos en un Celular.

-Se pudo conocer la importancia de la Vacunas así como el costo oculto que significa una falla en el sistema de matenimiento de las mismas.

-El proyecto integro el conocimiento adquirido en la Facultad y permitió adquirir nuevos conocimientos.

-Permitio conocer un campo nuevo en el cual todavía no esta insertada la Facultad, con lo cual es una oportunidad para desarrollar nuevas aplicaciones

6.1-Trabajos Futuros

Con base en el desarrollo del presente trabajo de tesis se proponen los siguientes trabajos futuros

-Desarrollar nuestra propia aplicación para Android para poder monitorizar los parámetros.

-Agregar la posibilidad de comunicación Via WiFi.

-Utilizar plataformas desarrollo que ya tengan Wifi integrado.

-Convertir el sistema para que pueda monitorizar niveles de tanque.

-Agregar sensores de corriente AC y vibraciones para poder realizar análisis preventivos ante una falla inminente de la heladera.

-Agregar al sistema un sistema de alimentación a baterías, para poder utilizarlo en los sistemas de almacenamiento de frío cuando se realizan campañas de vacunación en la zona rural.

Bibliografía

(OPS), O. P. (s.f.). *Curso de Gerencia para el Manejo Efectivo del Programa Ampliado de Inmunización*. Obtenido de http://www1.paho.org/english/ad/fch/im/isis/epi_mod/spanish/3/coge_termometros.asp

Camorro, P. (s.f.). *La Historia de las Vacunas*. Obtenido de Biomedical Capacities Support Programme: http://biocaps.webs.uvigo.es/sites/default/files/k._la_historia_de_las_vacunas-pedro_camorro_.pdf

CAS_Data_Logger. (s.f.). *ACCSENSE A2-05 ETHERNET WIRED TEMPERATURE MEASUREMENT POD*. Obtenido de http://www.dataloggerinc.com/products/A205_Ethernet_Temperature_Data_Logger/88/

Control_Solutions_Inc. (s.f.). *Accsense A2-05 LAN-Wired*. Obtenido de <http://www.vfcdataloggers.com/accsensea2-05lan-wiredmonintoringpod.aspx>

DARTICO, S. (s.f.). *DARTICO S.A.S*. Obtenido de <http://www.dartico.com/PRODUCTOS-CENTINELA.htm>

Del Campo Alepuz, G. (s.f.). *La cadena de frío de las vacunas en los Centros de Salud del Área 16, en la provincia de Alicante*. Obtenido de Página personal sobre la Formación Postgrado en Enfermería NICANOR ANIORTE HERNÁNDEZ: http://www.aniorte-nic.net/trabaj_vac_cad_frio.htm

Gobierno Vasco. (s.f.). Obtenido de http://www.euskadi.eus: www.euskadi.net/r33-2288/es/contenidos/informacion/vacunas_epidem/es_4330/adjuntos/conservacion_c.pdf

Laboratorio Ruminal S.A. (s.f.). *MANEJO CORRECTO DE VACUNAS Y BACTERINAS*. Obtenido de <http://www.ruminal.com.ar/sites/default/files/TRANSPORTE%20Y%20CADENA%20DE%20FRIO.pdf>

Loggershop. (s.f.). *TR-71NW LAN and USB 2 Channel Temperature Data Logger*. Obtenido de <https://www.loggershop.co.uk/product/tr-71nw-lan-two-channel-temperature-data-logger/>

Romero, L., & Andrade, D. (s.f.). *Colegio Robert F.Kennedy*. Obtenido de Las Vacunas: <http://es.calameo.com/read/0015351531cffe77e8d44>

Wikipedia. (s.f.). *Vacunas*. Obtenido de <http://es.wikipedia.org/wiki/Vacuna>