

# Informe del trabajo de laboratorio final

Bonserio Julia, Duarte Rodrigo Ivan, Mendoza Marcelo Daniel, Scholz Veronica

Lenguajes Formales y Automatas  
Universidad Nacional Arturo Jauretche  
Dirección Postal B1888.

[juliabonserio@live.com.ar](mailto:juliabonserio@live.com.ar), [rodrigo4@hotmail.com](mailto:rodrigo4@hotmail.com), [mendo.marce@gmail.com](mailto:mendo.marce@gmail.com), [vero.scholz@gmail.com](mailto:vero.scholz@gmail.com)

**Resumen-** En el presente informe se detalla el desarrollo de un sistema de control simulado que automatiza diferentes aspectos del hogar. Para esta tarea cuenta con sensores, actuadores y se diseñó un autómata para controlarlos. Finalmente, se presentan los resultados y dificultades encontradas durante el laboratorio y las conclusiones de la implementación.

**Palabras Clave-** domótica, arduino, proteus

## I. INTRODUCCIÓN

La domótica es un campo en crecimiento que utiliza la tecnología para automatizar ciertas tareas del hogar, lo cual aporta seguridad y comodidad a los hogares a la vez que gestiona de manera eficiente la energía.

En este trabajo de laboratorio se desarrolló un sistema que controla la iluminación, la activación de una alarma de movimiento codificada y la regulación de temperatura del hogar mediante la lectura de los estados de los sensores y la automatización de alguna acción predeterminada de los actuadores en base a la lectura recibida. Esto se logra mediante la utilización de una serie de componentes integrados: el simulador Proteus con un microcontrolador Arduino Uno, un teclado, sensores de luz, movimiento, temperatura y humedad; actuadores como luz led, un parlante, el display LCD y una consola serie.

## II. DISEÑO E IMPLEMENTACIÓN

El trabajo en general consistió en realizar la implementación en Proteus, diseñar el diagrama de estados del funcionamiento del sistema mediante un autómata de Moore y desarrollar un programa en Arduino IDE para ser interpretado por la placa arduino.

### A. Implementación en Proteus

Para el funcionamiento del sistema se requieren los siguientes componentes:

#### Placa Arduino Uno

La placa posee grupos de pines útiles para diferentes usos (ver Fig. 1):

- ANALOG IN: los pines A0, A1, A2, A3, A4, A5 permiten leer señales analógicas.
- DIGITAL (PWM~): son pines digitales que pueden funcionar como salidas con modulación por ancho de pulso (pulse-width modulation, PWM). PWM es una técnica que permite controlar la cantidad de energía que se envía a un dispositivo mediante una señal digital variando el ancho del pulso de una señal cuadrada con frecuencia constante.

- D0 y D1 permiten la comunicación serial con otros dispositivos (protocolos UART, SPI o I2C).

- 0: RX (Recepción): Recibe datos del dispositivo externo.
- 1: TX (Transmisión): Envía datos al dispositivo externo.

- POWER: proporciona alimentación de 5V a los componentes externos.
- GND: es la referencia de tierra para los circuitos.

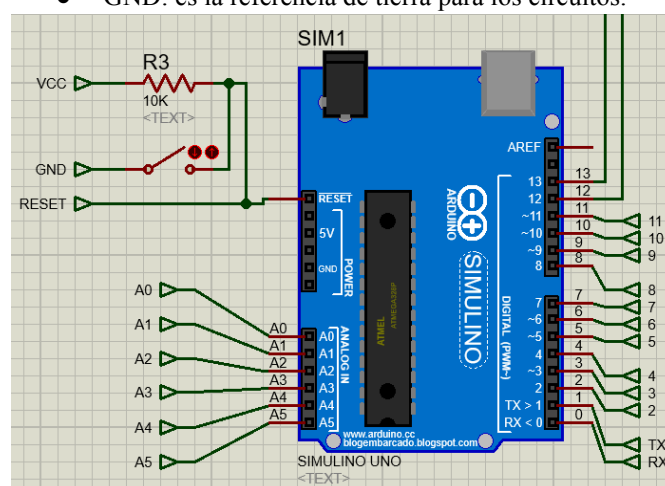


Fig. 2. SIMULINO (versión simulada de la Placa Arduino Uno)

#### Sensor de Luz (LDR)

El sensor detecta el nivel de luz ambiente. En el sistema, su función es activar la luz LED si la luz ambiental es menor a 100 lux. En la siguiente tabla se vuelcan los resultados de la medición de luz:

- S: son los valores que simula el sensor.
- Volts: valores leídos en el voltímetro.
- V: son los valores de tensión capturados por la placa.
- LUM: son los valores calculados sobre el valor V.

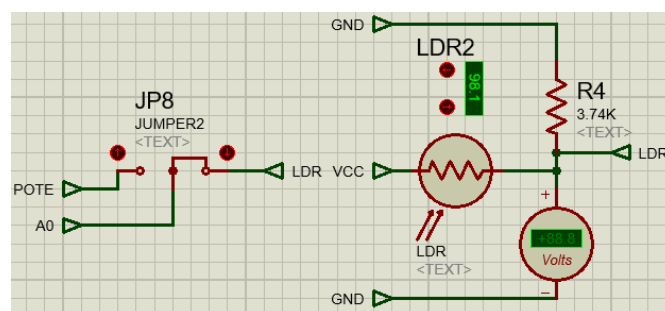


Fig. 3. LDR con switch

#### SENSOR DE LUZ: VALORES ENVIADOS Y LEÍDOS

S	VOLTS	V leída	LUM.
170	3,53	724	160
160	3,48	713	152
150	3,42	712	152
140	3,36	687	135
130	3,29	673	127
120	3,21	657	119
110	3,12	639	110
100	3,02	619	101
99	3,01	617	101
<b>98</b>	<b>3,00</b>	<b>615</b>	<b>100</b>
97	2,99	613	99
96	2,98	611	98
95	2,97	608	97
94	2,96	606	96
93	2,95	604	95
92	2,94	601	94
91	2,93	599	93
90	2,91	597	93

Tabla 1. Lectura en sensor de luz y cálculo de lúmenes

Se observa una diferencia entre el valor enviado y el valor medido. Esto podría interpretarse como el margen de error del sensor LDR o un error de calibración. De todos modos, en este sistema, estos valores podrían considerarse aceptables, ya que cuando el sensor envíe 98 lux, el sistema detectará los 100 lux necesarios para activar la luz led. Por esto se considera que  $\pm 2$  lux está dentro de un margen de error aceptable.

#### Sensor de Temperatura y Humedad (DHT11)

Este sensor mide la temperatura ambiental y la humedad relativa del aire.

En el sistema, tiene por objetivo, accionar frente a los cambios de temperatura según los siguientes requerimientos:

- Si la temperatura es mayor a 25°C → se activa la ventilación.
- Si la temperatura es menor a 18°C → se activa la calefacción.
- Si la temperatura es mayor a 50°C → se activa la alarma de incendio (sirena).

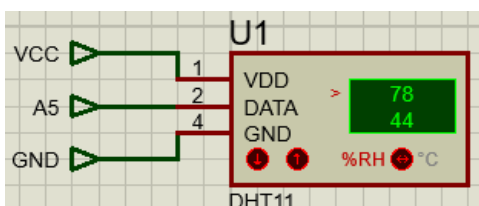


Fig. 5. DHT11

**Nota:** Tanto la ventilación como la calefacción son emuladas por un mensaje en la pantalla LCD.

#### Sensor de Movimiento

El sensor detecta movimiento en el área. Si se detecta movimiento y la alarma está activada, se activa una sirena que alerta posibles intrusos.

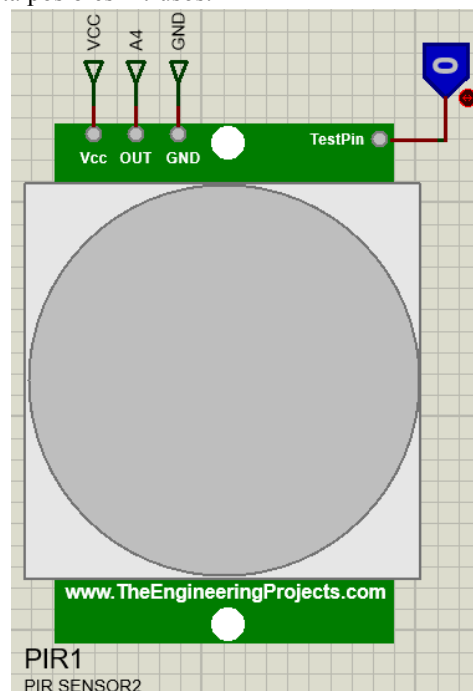


Fig. 5. Sensor PIR

#### Luz LED (actuador)

La luz led se activa cuando la placa detecta que el nivel de luz está por debajo de los 100 lux para iluminar el ambiente.

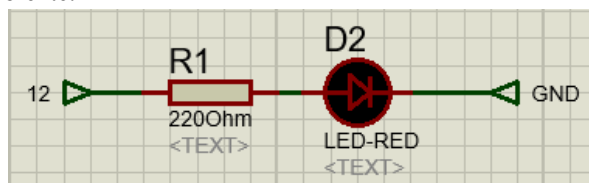


Fig. 5. Luz LED

#### Parlante (actuador)

El parlante emite una señal de alarma. En el sistema, se activa cuando la temperatura supera los 50°C para alarmar sobre un posible incendio y cuando, estando la alarma activada, se detecta movimiento, lo cual indica la presencia de posibles intrusos.

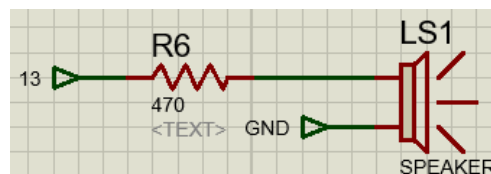


Fig. 5. Speaker

#### Visor LCD

El visor muestra información relevante de temperatura, encendido de calefacción, ventilación y del estado del sistema de alarma en tiempo real. Esto facilita la interacción del usuario con el sistema ya que proporciona una retroalimentación sobre las acciones y el estado del sistema. Por ejemplo, al presionar '\*' para ingresar la clave, el visor le indica claramente al usuario que "INGRESE CLAVE".

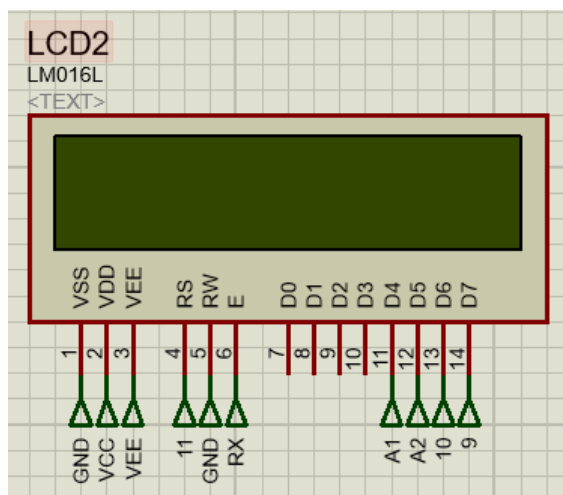


Fig. 5. Visor LCD

### Teclado numérico

El teclado permite ingresar la clave de activación/desactivación de la alarma del sistema, lo cual proporciona una capa de seguridad.

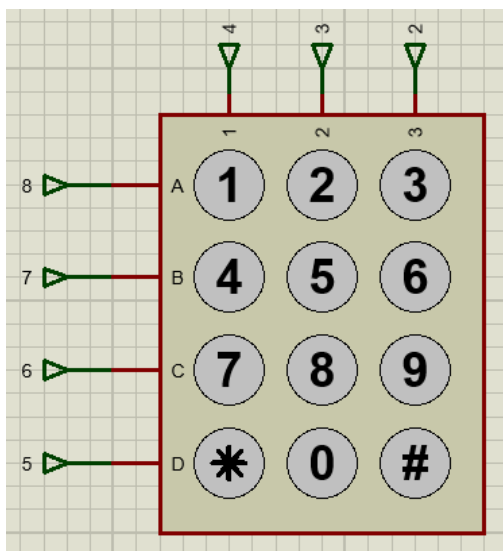


Fig. 5. Teclado numérico

### Monitor Serie

El monitor proporciona una interfaz de visualización remota a través del puerto serie (UART) de la placa Arduino. Se utiliza para hacer depuraciones y monitorear la actividad del sistema en tiempo real. Esto permite la visualización de datos de sensores, actuadores y del historial de la actividad del sistema que puede servir para el análisis y solución de problemas.

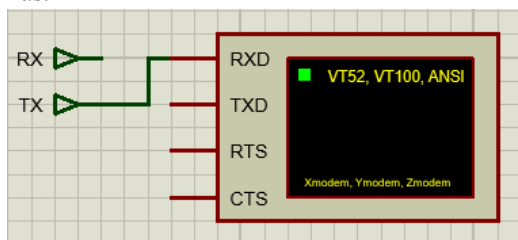


Fig. 5. Monitor Serie

### Conexión de componentes

A continuación se muestra el esquema completo y la TABLA de conexiones donde se puede ver a qué pin de la placa Arduino se conecta cada componente.

CONEXIONES		
Componente	Tipo	PINES ARDUINO
LDR	Analógico	A0, GND, VCC
DHT11	Analógico	A5, GND, VCC
PIR	Analógico	A4, GND, VCC
Luz led	Digital	D12, GND
Parlante	Digital	D13, GND
LCD		A1, A2, D9, D10, D11, RX, GND, VCC
Teclado numérico	Digital	COL: D2, D3, D4 ROW: D5, D6, D7, D8
Monitor	Comunicación Serial	TX

Tabla 1. Conexiones a la placa Arduino Uno

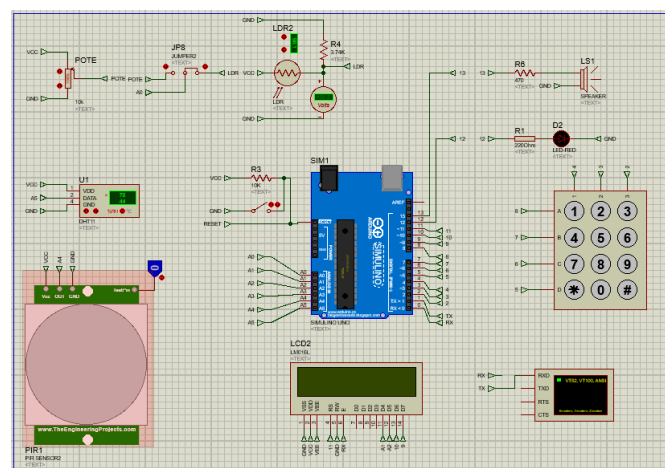


Fig. 1. Sistema Completo

### B. Desarrollo en Arduino IDE.

En el programa se capturan los estados de los sensores, se toman decisiones de acuerdo a los valores obtenidos y se envía una señal a los actuadores.

La activación de la alarma se realiza mediante una clave codificada en memoria usando la función de cifrado definida:

$$f(x) = (3x + 5) \bmod 10000 \quad (1)$$

esta función,

$$f: x \rightarrow y; \text{ con } x \in \mathbb{N}^+ \text{ e } y \in [0; 9999] \quad (2)$$

Es decir, la función *validarClave(code)* recibe el código *code* ingresado por el usuario, lo multiplica por 3, le suma 5 y luego calcula el módulo 10000 del resultado. De este modo,

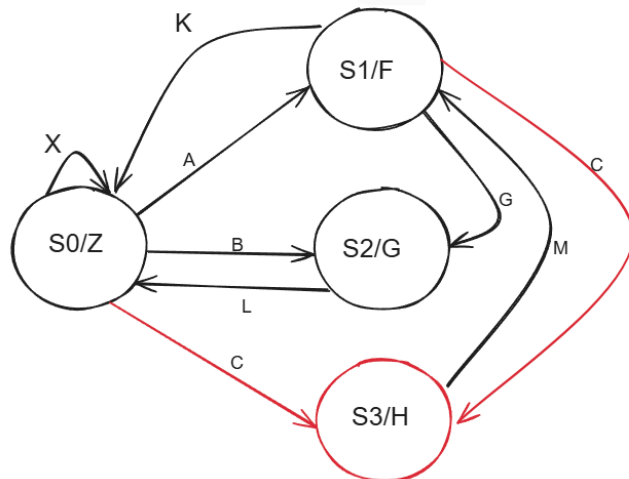
al ingresar el usuario la contraseña correcta del sistema (1234), la función calcula la imagen y se obtiene que:

$$f(1234) = 3707 \quad (3)$$

compara que el valor cifrado sea igual al SECRETO (3707) y devuelve un booleano que ACTIVA/DESACTIVA la alarma. En caso de que la contraseña sea incorrecta, devuelve un mensaje al usuario y continua la ejecución.

### C. Autómata de Mealy y diagrama de estados.

En esta parte se definieron 3 autómatas con el diagrama de Mealy, una máquina de estados finitos que dado su estado actual y una entrada, genera una salida. En el primero se analizaron las temperaturas, en el segundo la luz y en el tercero el movimiento. Se definieron los estados y se consideraron los sensores como entradas y los actuadores como salidas del sistema. Esto permitió verificar la comprensión de la consigna y el posterior desarrollo de la funcionalidad del programa.

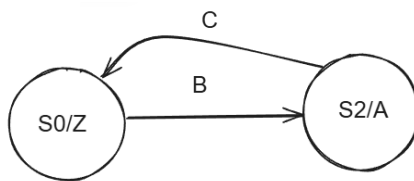


**S1** A - temperatura supera el umbral de los 25  
F - activar ventilación  
K - temp menor a 25 y mayor a 18

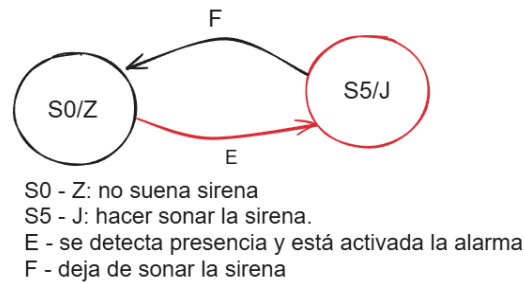
**S2** B - temperatura es menor a 18° encender la calefacción  
G - encender calefacción  
L - temp menor a 18

**S3** C - temperatura es mayor a 50°,  
H - activar Alarma de incendio  
M - temp menor a 25 y mayor a 18

**S0** X - temperatura entre 18 y 25 o iluminacion mayor a 100  
Z0 - nada



S0 - Z: luz apagada  
S2 - A : encender luz artificial  
B: iluminación menor a 100  
C: iluminación mayor a 100



S0 - Z: no suena sirena

S5 - J: hacer sonar la sirena.

E - se detecta presencia y está activada la alarma

F - deja de sonar la sirena

### III. RESULTADOS OBTENIDOS

El sistema demostró su eficacia al automatizar aspectos clave del hogar, como la iluminación, ventilación, calefacción y seguridad, cumpliendo con las condiciones de funcionamiento especificadas.

- **Lectura de Sensores:** el sistema fue capaz de leer correctamente los valores de los sensores de luz, temperatura, humedad y movimiento, y mostrarlos en el visor LCD y el monitor serial.
- **Activación de Actuadores:** los actuadores se activaron correctamente según las condiciones predefinidas:
  - El LED se encendió cuando el nivel de luz ambiental cayó por debajo de 100 lux.
  - El motor de ventilación se activó cuando la temperatura superó los 25°C.
  - La calefacción se encendió cuando la temperatura bajó de 18°C.
  - La sirena se activó por alta temperatura (incendio) y por detección de movimiento con alarma activada.
- **Interfaz de Usuario:** El teclado permitía activar y desactivar la alarma, mientras que la pantalla LCD proporciona información en tiempo real. El monitor serie facilitaba la depuración y el monitoreo. La integración de estos elementos mejoró significativamente la interacción y optimizó la experiencia y funcionalidad del sistema.
- **El diagrama de estados** (autómata de Mealy) modeló correctamente el comportamiento del sistema, facilitando el diseño e implementación del código.

### IV. CONCLUSIONES

La experiencia de laboratorio permitió aplicar de manera práctica los conceptos estudiados en la materia de Lenguajes Formales y Autómatas, especialmente a través de la implementación de un sistema domótico que evidenció la utilidad de los modelos teóricos en la resolución de problemas reales.

El desarrollo del autómata facilitó la comprensión y la transición de los conceptos teóricos a aplicaciones prácticas, reforzando el aprendizaje y la capacidad de aplicar el conocimiento en contextos reales.

Además, el diagrama de estados permitió comprender mejor el sistema a implementar, facilitó la comunicación del equipo y el control de las funcionalidades a implementar.

### V. DIFICULTADES

Durante la implementación del sistema encontramos algunas dificultades:

- **Compartir Archivos por Google Drive o GitHub:** intentamos compartir los archivos del proyecto utilizando Google Drive. Sin embargo, actualizar los cambios desde el Arduino IDE resultaba incómodo y decidimos usar GitHub, pero encontramos problemas al hacer merge. Probablemente porque subimos también el proyecto de Proteus a la misma carpeta, lo que causó conflictos.
- **Pines en placa Arduino:** al utilizar el ejemplo y modificar los pines, algunos quedaron incorrectamente asignados, lo cual causó problemas de funcionamiento y tuvimos que corregir las asignaciones de los pines para que coincidieran.

Name	Type	Number	Net
SIM2_PB2/...	I/O	10	10
SIM2_PB3/...	I/O	11	11
SIM2_PB4/...	I/O	12	12
SIM2_PB5/...	I/O	13	13
SIM2_PC0/...	I/O	A0	A0
SIM2_PC1/...	I/O	A1	A1
SIM2_PC2/...	I/O	A2	A1
SIM2_PC3/...	I/O	A3	A3
SIM2_PC4/...	I/O	A4	A4
SIM2_PC5/...	I/O	A5	A5

- **Sensor PIR:** la placa Arduino no detectaba el sensor PIR correctamente porque intentamos cargar el archivo \*.ino sin la librería adecuada para el sensor. Una vez que incluimos la librería correcta, el sensor funcionó como se esperaba.
- **Delay:** tuvimos que acostumbrarnos a que los tiempos de la simulación son lentos en comparación con el tiempo real y para verificar algunas reacciones a cambios del sensor debimos esperar más tiempo.

#### REFERENCIAS

- [1] Apuntes y videos de la cátedra LFyA.