

Proyecto Final
Control de Invernadero

Objetivo:

El alumno aplique lo visto a lo largo del semestre para poder desarrollar un sistema embebido de manera física o simulada.

Materiales:

1-Tarjeta Raspberry pi

1-Sensor LM95

1 Triac

1 MOC

10 resistencias

1 sensor de agua

1 bomba de agua

1 ventilador fan

1 capacitor

1 MCP3008

Descripción del funcionamiento de los componentes electrónicos relevantes

PWM

Para el ventilador usamos como el PWM(Pulse Width Modulation) para transmitir una señal analógica mientras que la señal que queremos transmitir es digital, para transmitir una cantidad de energía. El ciclo de trabajo de una señal periódica es el ancho de su parte positiva, en relación con el tiempo y se expresa en porcentaje que va del 10 al 100%.

$\text{Ciclo trabajo} = t/T$

t = tiempo en parte positiva.

T =Periodo de tiempo total.

Para tener una buena señal necesita un filtro RC que consta de una resistencia y un capacitor que filtra la señal para tener una señal más autentica.

Modulación de potencia de carga resistiva en AC

Para la modulación de AC se necesitan cuatro parámetros:

- Voltaje de carga.
- Corriente que circula por la carga.
- impedancia de la carga.
- Angulo de fase.

$$p(t) = vi = V_m \sin(\omega t + \theta) I_m \sin(\omega t + \theta_i)$$

Donde:

ω velocidad angular donde $\omega = 2\pi f$

θ Angulo de fase

V_m voltaje máximo

I_m corriente máxima

El circuito es puramente resistivo la impedancia es constante, es proporcionalmente al voltaje a la entrada de carga y el voltaje pico para controlar la potencia. Se usa esta técnica para circuitos que convierten AC-AC como se muestra en la ilustración.

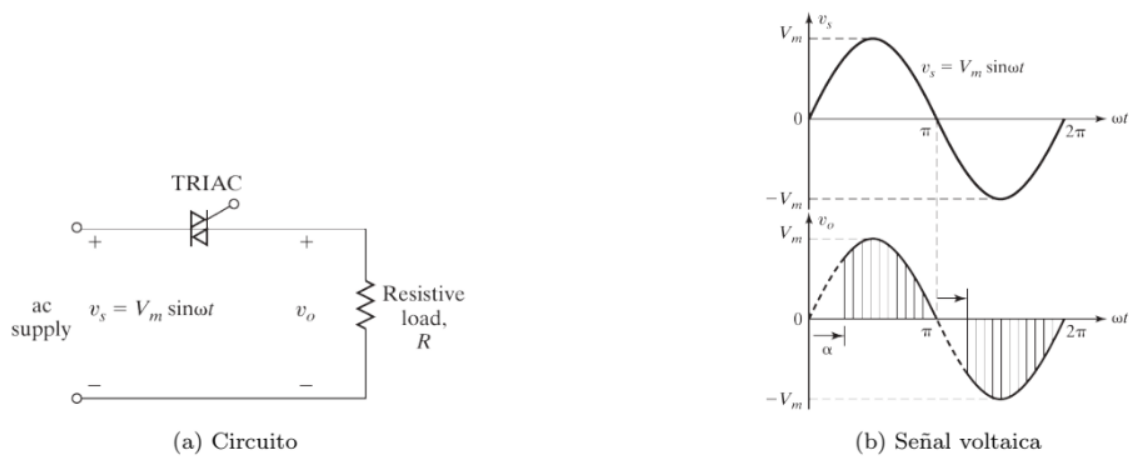


Ilustración 1 Convertidor AC-AC de una fase¹

¹ Fuente de la imagen: Rashid [2, pp. 33].

En la mayoría de casos no interesa la potencia neta si no el porcentaje de la potencia máxima que la carga esta entregando. Al estar sincronizadas la corriente y el voltaje por ser resistivo el circuito y no haber mas de una fase, se simplifica a un voltaje RMS de la línea.

$$P_f = \frac{V_o}{V_{RMS}}$$

Tomando en cuenta que RMS es fijo tenemos

$$V_o = V_{RMS} \sqrt{\frac{1}{\pi} \left(\pi - \alpha + \frac{\sin(2\alpha)}{2} \right)}$$

Usamos el ángulo de disparo en el TRIAC la ecuación de potencia del foco nos queda

$$P_f = \frac{V_{RMS} \sqrt{\frac{1}{\pi} \left(\pi - \alpha + \frac{\sin(2\alpha)}{2} \right)}}{V_{RMS}}$$

$$P_f = \sqrt{\frac{1}{\pi} \left(\pi - \alpha + \frac{\sin(2\alpha)}{2} \right)}$$

El factor de potencia no depende de los valores de voltaje, sino sólo del ángulo de disparo del TRIAC α ¹. Y para conocer el tiempo de disparo se usa. ²

$$\tau = \frac{\alpha}{\omega}$$

Raspberry Pi

Es un ordenador de bajo costo y compacto el cual es muy utilizado para poder realizar prototipos y para la informática y electrónica en las escuelas. Habitualmente se usa el sistema operativo basado en GNU/Linux que estén basados en Debian.

LM95

² Practica 4[,pp.12]

Este es un sensor en el cual tenemos una entrada analógica en el cual nos da la información de la temperatura actual por medio de una señal. Así que usaremos un MCP3008 el cual sirve de puente el cual manda la señal serial SPI

El MCP3008 sirve para convertir señal analógico a digital de aproximación sucesiva de 10 bits con muestra integrada y circuitos retenidos.

El LM95 mide temperaturas de -55° a 150°C donde 1°C equivale a 10mV que soporta de 4V y 30V.

Advertencias y cuidados:

Este dispositivo puede si es mal usado puede que se le queme partes del si no es que se puede quemar todo el dispositivo.

Las bombas de agua si no tiene agua puede llegarse a quemar

El ventilador este libre, ya que puede romperse al no poder girar

No ponga el dispositivo a temperaturas mayores a 40° y menores a 30°

Que debe hacer

Control que permite monitorear y administrar remotamente un invernadero. El sistema embebido descrito deberá realizar las siguientes funciones:

Encendido y apagado de sistema de irrigación

Desplegado de gráfica con histórico de temperatura, irrigación y acciones tomadas

Control de temperatura del invernadero utilizando PID

Control de potencia del radiador (foco incandescente)

Control de potencia del ventilador (ventilador de CPU o cualquier otro motor DC)

Programado de ciclos de temperatura e irrigado

El sistema embebido incorporará un servidor web el control remoto (local vía IP) de las funciones descritas anteriormente.

Configuración:

1. Descargue el código de <https://github.com/JoseLuisLeyva/Proyecto-Final-FSEm-.git>
2. En la línea de comandos ponga `pip install tk`
3. En la línea de comandos ponga: `cd código`
4. En la línea de comandos ponga: `cd interfaz`
5. Para ejecutar: `python3 proyecto.py`
6. Mostrara una interfaz



Ilustración 2 Simulador python

7. Vamos a la carpeta de `interfaz_control`
8. hacemos doble click en `index`
9. Debe mostrar la ilustración

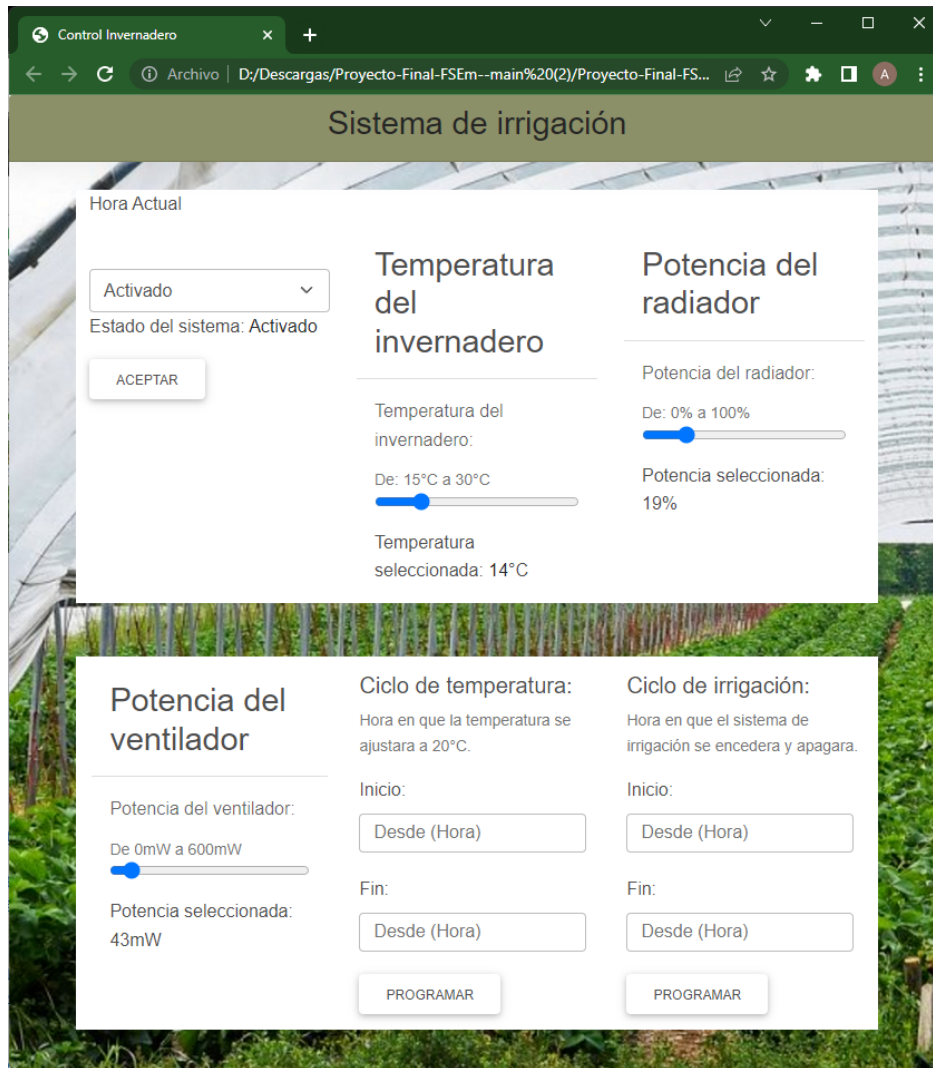


Ilustración 3 Menú página web

Desarrollo de los componentes de software

En el archivo de cálculos.py encontramos

Para el index, usamos las siguientes bibliotecas en el cual bdfuncion conectamos la base de datos que usamos es firebase en el cual declaramos los valores de la interfaz que deseamos pasar al simulador en python.

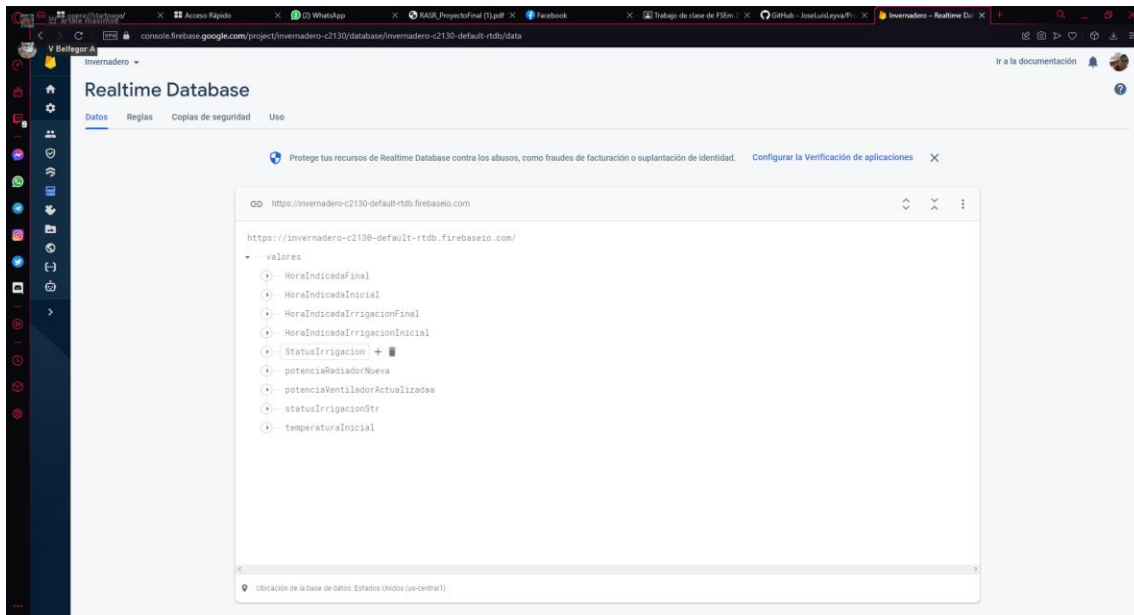


Ilustración 4 Base de datos

Con bdfuncion hace la conexión la página con la base de datos, con common mostramos los datos en el index para que se vea que se cambia, también se encuentra la programación de la irrigación y hace las modificaciones y la temperatura la ajusta. En la de irrigación, ventilador, temperatura, radiador, programacionTemperatura y programacionIrrigacion envía a la base de datos los datos de cada acción.

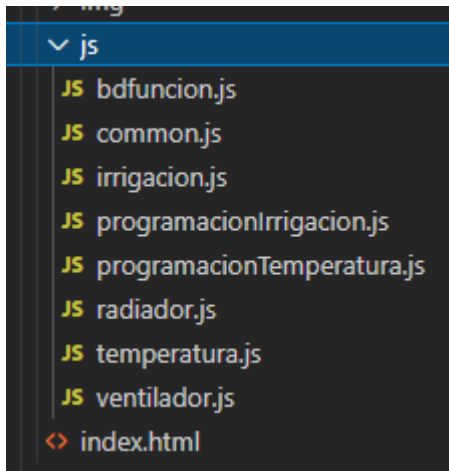


Ilustración 5 javascripts

Usamos la biblioteca firebase para obtener los datos de la base de datos, pero tuvimos que arreglar porque después de la versión 3.6 ya no se podía usar porque la palabra async se hizo reservada y lo arreglamos como debe cambiar el nombre del archivo async.py (elijo async_fb.py)

luego, en el archivo 'init .py', reemplace la línea 3 con: from .async_fb import process_pool

luego, en el archivo 'firebase.py', reemplace la línea 12 con: from .async_fb import process_pool

En calculos.py son las tres funciones principales donde PWM, la intensidad,pid y la temperatura en el cual en el ciclo del pwm calculamos el porcentaje de del ciclo con la fórmula de

$$ciclo = \frac{voltajeIntrducido * 100}{voltajeMax - valtajeMin}$$

Este nos dará la velocidad que se moverá el pwm.

```
def valor_dutycycle(potencia):
    potenciaCalcular = int(potencia)/1000

    #PWM
    Amperaje=0.05
    volatajeMax=12
    VoltajeMin=0

    volataje=potenciaCalcular/Amperaje
    dutycycle=(volataje*100)/(volatajeMax-VoltajeMin)

    #Frecuencia Corte usamos un condensador pwm con una R=10k ohms y un capacitor de 1uF
    Resistencia=20000
    Capacitor=0.1e-6
    Fc=1/(2*math.pi*Resistencia*Capacitor) #Hz

    return dutycycle,Fc
```

Ilustración 6 Función de porcentaje de ciclo

Con la intensidad del foco usamos un triac pero usamos el ángulo de disparo, este es el que nos ayuda a sacar el factor potencia y también sacamos alfa.

```
def focoIncandesente(potencia):
    #Se calculo con la potencia del foco con la ecuacion de angulo de disparo en el TRIAC
    #Pf=sqrt((1/pi)*(pi-alfa+(Sin(2*alfa)/2)))
    #alfa=algulo de disparo
    potencia=int(potencia)

    powerFactor={
        #PorcetanjeDisparo:alfa()
        0 :0,
        1 :0.01,
```

Ilustración 7 intensidad foco parte 1


```
    99: 3.19,  
    100: 3.199999  
  }  
  
  #Usamos a 60Hz  
  f=60  
  tiempoDisparo=(powerFactor[potencia]/(2*math.pi*f))*1000  
  
  return powerFactor[potencia],tiempoDisparo  
  
#Control de temperatura  
def tempPID(temp):  
    tempAnt = temp  
    tempAux = (firebase.get('/valores/temperaturaInicial',''))  
    temp = int(tempAux["value"])  
    while tempAnt != temp:  
        tempAnt = controlPID(temp,tempAnt)  
    return tempAnt
```

Ilustración 8 Intensidad foco parte 2

Tomando los datos hace el control de la temperatura

```
#Control de temperatura  
def tempPID(temp):  
    tempAnt = temp  
    tempAux = (firebase.get('/valores/temperaturaInicial',''))  
    temp = int(tempAux["value"])  
    while tempAnt != temp:  
        tempAnt = controlPID(temp,tempAnt)  
    return tempAnt
```

Ilustración 9 función del temp pid

Realiza el algoritmo del pid en el cual tiene

```
#controlador PID
def controlPID(tempP, tempAnt):
    global errorp,error,temp
    error_control = tempP - tempAnt
    tempAnt = (errorp * kd ) + (error_control * kp ) + (error * ki )
    temp = max(min(150,temp),-55)
    errorp = error_control
    error = error + error_control
    print ("teperatura",tempAnt)
    return int(tempAnt)
```

Ilustración 10 Control de pid

Es la interfaz en la cual muestra los datos y la animación de como se debe comportar.

```
# /usr/bin/python3
from tkinter import *
from calculos import *
from firebase import firebase

root = Tk()
root.title('Control invernadero')
root.geometry("1080x720")

firebase = firebase.FirebaseApplication("https://invernadero-c2130-default-rtdb.firebaseio.com/",None)
background = PhotoImage(file="invernadero.png")
imagen_label = Label(root, image=background)
imagen_label.place(x=0,y=0, relwidth=1, relheight=1)

var = StringVar()
label = Label( root, text="Control invernadero", anchor = CENTER, font=('Times 36'), fg="white" ,bg="#41aa5e")
label.config(width=200)

temp = firebase.get('/valores/temperaturaInicial','')
temp = temp["value"]

#textVariable
#var.set("Control invernadero")

#Labels de temperatura
temperatura_label = Label( root, text="Temperatura:", font=('Sans 12'), bg="white")
temperatura_label.place(x=50, y=100)
temperatura_label = Label( root, text="Rango: 15C - 30C", font=('Sans 8'), bg="white")
temperatura_label.place(x=50, y=150)

#Labels de potencia
radiador_label = Label( root, text="Potencia del radiador:", font=('Sans 12'), bg="white")
radiador_label.place(x=50, y=250)
radiador_label = Label( root, text="Rango 0 % a 100%", font=('Sans 8'), bg="white")
radiador_label.place(x=50, y=300)
factorF_label = Label( root, text="Factor de potencia ", font=('Sans 8'), bg="white")
factorF_label.place(x=50, y=325)
tiempo_label = Label( root, text="Tiempo de disparo (ms) ", font=('Sans 8'), bg="white")
tiempo_label.place(x=50, y=370)

#Hora irrigacion inicial
irrigacion_inicial_label = Label( root, text="Hora inicial irrigacion:", font=('Sans 12'), bg="white")
irrigacion_inicial_label.place(x=50, y=425)
```

Ilustración 11 Codigo de interfaz



Video Expliacion: <https://youtu.be/QsdbriDBtas>

Conclusión:

En este proyecto desarrollamos lo visto en clase y en el laboratorio, pero no se pudo realizar en forma física, pero desarrollamos un simulador el cual nos ayudo a mostrar como se debe comportar en forma física. Se llevo a los resultados que esperados con un análisis profundo de cada dispositivo.

Cuestionario:

¿El raspberry pi como se configura para tener el pwm?

Con la funcion del duty se puede configurar de la biblioteca machine.

¿Cómo transforma el MCP3008 de analogico a digital?

Se puede con cuatro pares de entradas pseudodiferenciales u ocho entradas de un solo extremo. La no linealidad diferencial (DNL) y la no linealidad integral (INL) se especifican en ± 1 LSB. La comunicación con los dispositivos se lleva a cabo mediante una sencilla interfaz en serie compatible con protocolo SPI.

¿Por qué la Pf como se puede obtener?

Pueden utilizarse varios métodos de aproximación numérica para cada uno de los segmentos de la curva y utilizar el más adecuado para en cada uno de los rangos de interés. Otros métodos que requieren un número mucho menor de cálculos hacen uso de una tabla de valores discretos (por ejemplo, incrementos de 1 % en el factor de potencia) e interpolan linealmente entre estos, dejando el error como una perturbación a corregir por el controlador

Referencias:

MEDIR TEMPERATURA CON ARDUINO Y SENSOR LM35. (2015,15 Julio).LUIS LLAMAS.

Recuperado 26 Mayo 2022 <https://www.luisllamas.es/medir-temperatura-con-arduino-y-sensor-lm35/>

¿Qué es PWM y cómo usarlo?(2020,26,Agosto) .Solectro. Recuperado 24 mayo 2022, de

<https://solectroshop.com/es/blog/que-es-pwm-y-como-usarlo--n38#:~:text=PWM%20son%20las%20siglas%20de,o%20sufrir%20distorsión%20por%20interferencias.>

MCP3008-I/P(2014). Newark. Recuperado 26 mayo 2022, de

<https://mexico.newark.com/microchip/mcp3008-i-p/analog-to-digital-converter-adc/dp/19C7200#:~:text=El%20dispositivo%20MCP3008%20es%20un,entradas%20de%20un%20solo%20extremo.>

Raspberry Pi: Crea proyectos DIY por muy poco dinero(2021,18,Julio). Profesional review,Recuperado 26 mayo 2022,de <https://www.profesionalreview.com/2021/07/18/que-es-raspberry-pi/>

MCP3008-I/P(2014). Newark. Recuperado 26 mayo 2022, de

<https://mexico.newark.com/microchip/mcp3008-i-p/analog-to-digital-converter-adc/dp/19C7200>

PWM con Micropython y Raspberry Pi PICO / ESP(2021,10,Abril). Control automático educación.

Recuperado 22 mayo 2022, de <https://controlautomaticoeducacion.com/micropython/pwm-pico-esp/>

How to install Tkinter in Python?(2021,11,Marzo).Tutorials point. Recuperado 25 mayo 2022, de

<https://www.tutorialspoint.com/how-to-install-tkinter-in-python>

How to Get Started with Firebase Using Python(2021,2, Febrero).freeCodeCamp.Recuperado 23 mayo 2022, de

<https://www.freecodecamp.org/news/how-to-get-started-with-firebase-using-python/>

Ventilador icono gratis(2021).Flaticon. Recuperado 27 mayo 2022, de https://www.flaticon.es/icono-gratis/ventilador_4617892

Termometro(2016,20, Diciembre).Pixabay. Recuperado 27 mayo 2022, de

<https://pixabay.com/es/illustrations/term%C3%B3metro-la-temperatura-la-medida-1917500/>

LAMPARAS LED VS LAMPARAS INCANDESCENTES(2020,6, Diciembre).Mi mundo

sustentable.Recuperado 27 mayo 2022, de <https://mimundosustentable.com/lamparas-led-vs-lamparas-incandescentes/>