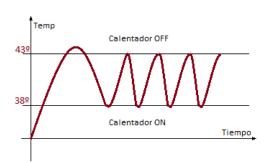
Total Hojas

Duración de todo el final 18:00 a 20:00

23/septiembre/2021 LPC1769

Nombre y Apellido	Nº Legajo	Calificación	Docente Evaluador	
			Nombre	Firma





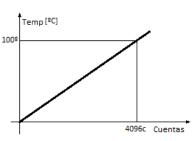


Figura 1

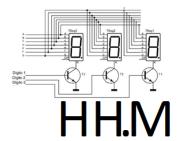
Figura 2

Figura 3

Se pide desarrollar una placa controladora para una yogurtera comercial IoT(Figura 1). El cultivo lácteo (yogurt) realiza el crecimiento dentro de los 38ºC a los 43ºC. Por encima de esa temperatura el cultivo se muere y por debajo no hay más crecimiento. Para controlar esta temperatura hay una resistencia térmica que eleva la temperatura al activarse y al apagarse se baja. El tiempo promedio para la fabricación de yogurt es de 8 horas aproximadamente.

El equipo contiene un sensor analógico para saber la temperatura en el interior. (Está conectado al conversor ADO canal 0). Además debe constar con 3 display's 7 segmentos (cátodo común) donde se mostrara el tiempo en horas y minutos (HH.M). El diseñador del hardware conecto los display sin decodificador, utilizando todos los pines del display incluyendo el punto decimal.

Esta yogurtera tiene un módulo WiFi para comunicarse con un celular de tal forma de poder visualizar el estado de la misma de forma online.



El equipo tiene dos pulsadores:

- Pulsador 1: Se presiona para encender o detener el proceso de control de temperatura.
- Pulsador 2: Cada vez que se presiona incrementa el tiempo 30 minutos, si el usuario se pasa de 15 horas se resetea a 6 horas que es el mínimo. (Mínimo=6 horas, Máximo=15 horas)

El proceso de funcionamiento consta de controlar la temperatura dentro del equipo durante el tiempo previamente configurado. Al finalizar el tiempo se debe apagar la resistencia y hacer sonar un buzzer por un segundo.

Una vez por segundo se debe enviar un reporte vía Wifi, utilizando la UARTO, con el siguiente formato:

I	CARÁCTER INICIO	TEMPERATURA	TIEMPO RESTANTE	CHECKSUM	CARÁCTER FIN
	<b>'&gt;'</b>	(unidad grados)	(unidad minutos)	(suma de los 4 bytes	<b>'</b> <'
	(1 BYTE)	(1 BYTE)	(2 BYTES)	anteriores)	(1 BYTE)
				(1 BYTE)	

Se pide realizar las siguientes funciones:

- Realizar la función "main".
- Realizar la función de Timertick (ya inicializada cada 1mseg)
- Realizar todas las funciones necesarias para el control del display
- Realizar todas las funciones que crea necesario para controlar la yogurtera y los pulsadores (máquina de estado).
- Realizar todas las funciones necesarias únicamente para el envió de la trama serial (No hace falta hacer la inicialización)
- Realizar todas las funciones para la lectura del conversor AD, usando interrupciones. (No hace falta hacer la inicialización) (La curva de conversión es la figura 3)

Pines	Puertos
Segmentos	P3.0 a P3.7
Digito 1	P4.6
Digito 2	P2.1
Digito 3	P0.24
Rele Temp	P0.23
Sensor Temp	AD0.0
Pulsador 1	P0.0
Pulsador 2	P0.1
Buzzer	P0.2

# NOTAS:

- Se pueden utilizar las funciones de librería de la cátedra tipo SetPIN(), GetPIN(), Push(), Pop(), etc.
- No se pide la inicialización de los pines.
- Utilice la siguiente tabla para los displays de 7 segmentos:

uchar code Disp7[] =  $\{0x3f, 0x06, 0x5B, 0x4f, 0x66, 0x6D, 0x7C, 0x07, 0x7f, 0x67\}$ ;

La resistencia se encienden con 1 y los sensores con 1 están activados

# FINAL INFORMATICAL

Total Hojas

Duración de todo el final 18:00 a 20:00

# 23/septiembre/2021 LPC1769

Nombre y Apellido	Nº Legajo	Calificación	Docente Evaluador	
			Nombre	Firma

## Información Extra

#### **ADC Registers**

Register	Description
ADCR	A/D COntrol Register: Used for Configuring the ADC
ADGDR	A/D Global Data Register: This register contains the ADC's DONE bit and the result of the most recent A/D conversion
ADINTEN	A/D Interrupt Enable Register
ADDR0 - ADDR7	A/D Channel Data Register: Contains the recent ADC value for respective channel
ADSTAT	A/D Status Register: Contains DONE & OVERRUN flag for all the ADC channels

## **ADC Register Configuration**

	ADCR							
	31:28	27	26:24	23:22	21	20:17	16	15:8
	Reserved	EDGE	START	Reserved	PDN	Reserved	BURST	CLCKDIV

#### Bit 7:0 - SEL: Channel Select

These bits are used to select a particular channel for ADC conversion. One bit is allotted for each channel. Setting the Bit-0 will make the ADC to sample AD0[0] for conversion. Similarly setting bit-7 will do the conversion for AD0[7].

## Bit 15:8 - CLCKDIV: Clock Divisor

The APB clock (PCLK\_ADCO) is divided by (this value plus one) to produce the clock for the A/D converter, which should be less than or equal to 13 MHz.

#### Bit 16 - BURST

This bit is used for BURST conversion. If this bit is set the ADC module will do the conversion for all the channels that are selected(SET) in SEL bits. CLearing this bit will disable the BURST conversion.

# Bit 21 - PDN: Power Down Mode

Setting this bit brings ADC out of power down mode and makes it operational.

Clearing this bit will power down the ADC.

#### Bit 24:26 - START

When the BURST bit is 0, these bits control whether and when an A/D conversion is started:

000 - Conversion Stopped

001- Start Conversion Now

The remaining cases (010 to 111) are about starting conversion on occurrence of edge on a particular CAP or MAT pin.

## Bit 27 - EDGE

This bit is significant only when the START field contains 010-111. It starts conversion on selected CAP or MAT input.

- 0 On Falling Edge
- 1 On Rising Edge

#### ADGDR or ADDRO/7 (ADC Global Data Register)

ADGDR					
31	27	26:24	23:16	15:4	3:0
DONE	OVERRUN	CHN	Reserved	RESULT	Reserved

#### Bit 15:4 - RESULT

This field contains the 12bit A/D conversion value for the selected channel in ADCR.SEL

The vale for this register should be read oncve the conversion is completed ie DONE bit is set.

#### Bit 26:24 - CHN: Channel

These bits contain the channel number for which the A/D conversion is done and the converted value is available in RESULT bits(e.g. 000 identifies channel 0, 011 channel 3...).