

# Taller de Microcontroladores y Placas de Desarrollo

Profesor: Kalun José Lau Gan

Semestre 2023-2

Sesión 1

1

## Agenda:

- Lectura del sílabo
- Lista de materiales
- Requerimientos de software y documentación
- Repaso de conocimientos previos

2

## Lectura del sílabo

Disponible en el Campus Virtual

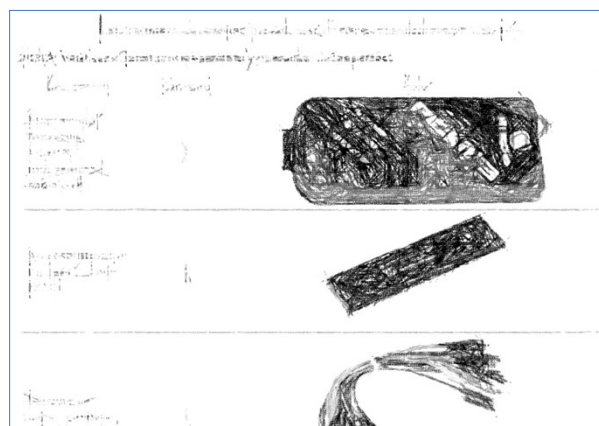
- Objetivos
- Competencias
- Contenido
- Evaluaciones
- Fórmula de evaluación

SÍLABO SEMESTRE ACADÉMICO 202320	
<b>DATOS GENERALES</b>	
1. FACULTAD(ES)	INGENIERÍA
2. PROGRAMA(S) DE ESTUDIOS	INGENIERÍA ELECTRÓNICA
3. ASIGNATURA	TALLER DE MICROCONTROLADORES Y PLACAS DE DESARROLLO
4. MODALIDAD	PRESENCIAL
5. CÓDIGO	ELEC-225
6. CICLO DE ESTUDIOS	05
7. CRÉDITOS	2
8. NIVEL	PREGRADO
9. TOTAL DE HORAS SEMESTRALES	68
10. TOTAL DE HORAS SEMANALES	4
11. FECHA DE INICIO	28/08/2023
12. FECHA DE CULMINACIÓN	23/12/2023
13. DURACIÓN EN SEMANAS	17
14. PRERREQUISITOS	ELEC-221 O ELEC-184 O ELEC-144
15. DOCENTE(S)	LAU GAN, KALUN JOSE;
16. CORREO ELECTRÓNICO INSTITUCIONAL	klaug1@upao.edu.pe
<b>SUMILLA</b>	
La asignatura pertenece al área de formación básica, de naturaleza teórica-práctica experimental. Su propósito es desarrollar en los estudiantes conocimientos y habilidades para la solución de problemas reales de ingeniería. Comprende como ejes temáticos: diseño electrónico de alto nivel, configuración de sistemas operativos en tiempo real, diseño CAD/CAM para el desarrollo de placas impresas (PCB).	

3

## Lista de materiales

- Disponible en el campus virtual.
- Mínimo de materiales para el desarrollo del curso.
- Se emplearán tanto para las sesiones de laboratorio como para las asignaciones y prácticas.



4

## Requerimientos de software y documentación :

- Software:

- Microchip MPLAB X v6.15 (solo soporte a PICKIT4 en adelante, no soporta MPASM)
- ~~Microchip MPLAB X v6.05 (última en dar soporte a PICKIT3, no soporta MPASM)~~
- ~~Microchip MPLAB X v5.35 (para los que usan PICKIT3, soporta MPASM)~~
- ~~Microchip MPLAB X v5.30 (para los que usan PICKIT2, soporta MPASM)~~

Archivo histórico de versiones antiguas de Microchip:

<https://www.microchip.com/development-tools/pic-and-dspic-downloads-archive>

- Microchip XC8 v2.41 (la mas actual)

- Documentación inicial:

- Hoja técnica del PIC18F57Q43
- PIC18F57Q43 Curiosity Nano Hardware User Guide
- MPLAB\_XC8\_C\_Compiler\_Legacy\_User\_Guide
- MPLAB\_XC8\_C\_Compiler\_User\_Guide\_for\_PIC
- MPLAB\_XC8\_C\_Compiler\_Users\_Guide\_for\_Embedded\_Engineers\_PIC

5

## ¿Qué es un microcontrolador?

- Es un dispositivo microelectrónico (basado en tecnología de semiconductores) programable que posee casi todos los componentes para un funcionamiento autónomo.
- Componentes: CPU, memorias de programa y de datos (RAM y ROM), periféricos (E/S, temporizadores, A/D, etc), gestión energética, fuentes de reloj
- Se requiere de un programa (hecho con un lenguaje de programación desde un entorno de desarrollo en una PC) y de un programador para que transporte el código compilado hacia la memoria de programa del microcontrolador antes de iniciar su operación
- Su funcionamiento es de manera secuencial (necesita de una fuente de reloj).
- Para dar soluciones “compactas” (embedded) a determinado problema.
  - Portátil (autonomía, tamaño, consumo, etc)

6

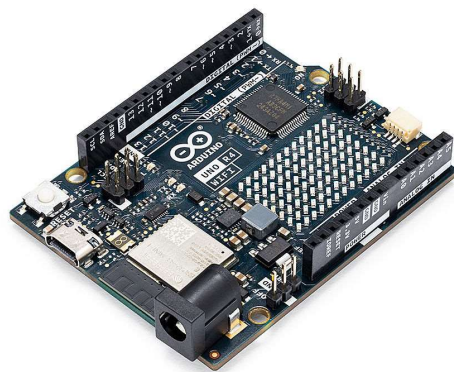
¿Esto es un microcontrolador?



- Es una plataforma de desarrollo electrónico basado en un microcontrolador (ATMEL ATmega 328P)

7

¿Esto es un microcontrolador?



- Es una plataforma de desarrollo electrónico basado en un microcontrolador (Renesas RA4M1 ARM Cortex-M4)

8

## ¿Por qué no enseña Arduino en lugar de PIC? ¿Arduino no es mas fácil?

- Arduino consume mas energía
- Arduino se programa a un nivel mas alto y por ende consume mas recursos de procesamiento.
- Arduino lo trabajas como si fuera una caja negra.
- Trabajar con PIC puedes alcanzar mayores niveles de eficiencia en términos de desempeño, costo, consumo energético, uso de memoria.
- Arduino es un entorno de desarrollo open-source el cuál el microcontrolador destino posee un firmware inicial para la interacción con el software IDE en la PC. Como consecuencia de esto el microcontrolador tendrá menor desempeño frente a usar lenguaje Assembler.
- Cuando se tiene que atender aplicaciones o procesos críticos, en Arduino no tenemos velocidad de respuesta a menos que se emplee microcontroladores de mayor desempeño. Esto no representaría problema alguno si se desarrolla en Assembler.

9

## Entonces si usar Arduino presenta tantas desventajas. ¿Por qué se usa extensivamente?

- Tendencia open source, open hardware.
- Por el poco tiempo que requieres para hacer una solución electrónica, sin tener prioridad en la optimización de recursos.
- Mucha experiencia y muchos usuarios (técnicos y no técnicos) usando esta plataforma.
- Bastante documentación de ejemplos y notas de aplicación basado en esta plataforma.
- Fabricantes OEM desarrollan variedad de sensores y actuadores en forma de módulos plug-in para ser usados en esta plataforma.

10

## Repaso de conocimientos previos

- Álgebra de Boole, circuitos digitales (Fund. Sist. Digit. Thomas Floyd)
- Algoritmos, diagramación en diagrama de flujo
- Arquitectura de computadoras (Org. Y Arq de PCs de William Stallings)
- Circuitos eléctricos (interfaces de potencia, sensores)
  - Transistores en corte y saturación, diodos rectificadores, LEDs.
  - Optoacopladores
  - Relés
- Señales analógicas y digitales (señales y sistemas)
  - Op-Amp: Modos de trabajo (amplificador (noinv, inv), oscilador, comparador, sumador, integrador, diferencial, compresor, filtrado, etc)
  - Conversión A/D y D/A, Teorema de muestreo

11

## Bases numéricas



Decimal	Binario	Octal	Hexa
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F


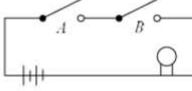
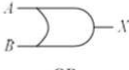
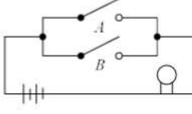



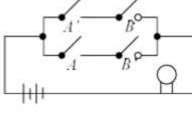
BIN = 11010100  
 HEX = D4H

HEX = A5H  
 BIN = 10100101

HEX = FBH  
 BIN = 11111011

12

# Álgebra de Boole

Expresión	Compuerta Lógica	Tabla de Verdad	Circuito de Interruptores															
$X = AB$	 AND	<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1	
A	B	X																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
$X = A + B$	 OR	<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1	
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
$X = A'$	 NOT	<table><tr><th>A</th><th>X</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	X	0	1	1	0										
A	X																	
0	1																	
1	0																	
$X = A \oplus B$ $\Rightarrow$ $X = A'B + AB'$	 XOR (OR exclusivo)	<table><tr><th>A</th><th>B</th><th>X</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	0	
A	B	X																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

Axiomas del Álgebra de Boole	
<b>Leyes Conmutativas</b>	
$a + b = b + a$	$a * b = b * a$
<b>Leyes Distributivas</b>	
$a + (b * c) = (a + b) * (a + c)$	$a * (b + c) = (a * b) + (a * c)$
<b>Leyes de Identidad</b>	
$a + 0 = a$	$a * 1 = a$
<b>Leyes de Complemento</b>	
$a + a' = 1$	$a * a' = 0$
<b>Leyes de Idempotencia</b>	
$a + a = a$	$a * a = a$
<b>Leyes de Acotamiento</b>	
$a + 1 = 1$	$a * 0 = 0$
<b>Leyes de Absorción</b>	
$a + (a * b) = a$	$a * (a + b) = a$
<b>Leyes Asociativas</b>	
$(a + b) + c = a + (b + c)$	$(a * b) * c = a * (b * c)$
<b>Unicidad del Complemento</b>	
Si $a + x = 1$ y $a * x = 0$ , entonces $x = a'$	
<b>Ley de Involución</b>	
$(a')' = a$	
<b>Teoremas</b>	
$0' = 1$	$1' = 0$
<b>Leyes de DeMorgan</b>	
$(a + b)' = a' * b'$	$(a * b)' = a' + b'$

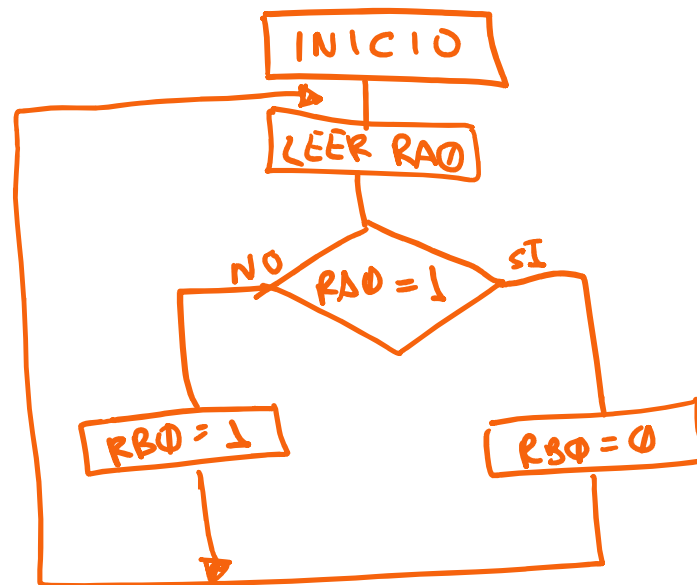
13

## Importancia del algoritmo

- Los algoritmos son representaciones gráficas de una tarea que va a hacer el microcontrolador.
- Pueden ser representados en diagramas de flujo, NS, pseudocódigo.
- En el presente curso se hará uso de diagramas de flujo (flowchart)

14

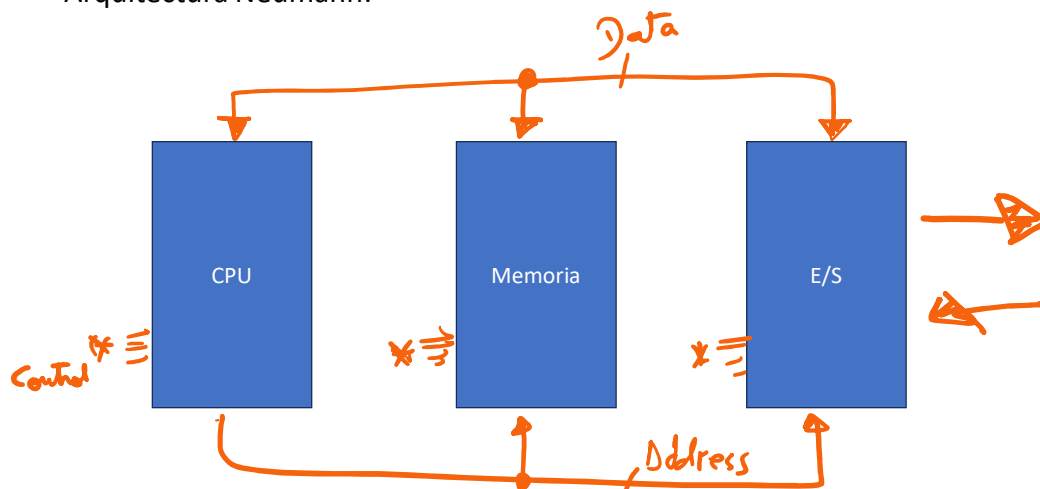
## Diagramas de flujo



15

## Arquitectura de Computadores

- Estructura de una computadora
  - Arquitectura Neumann:



16



## Workflow para el desarrollo de aplicaciones con microcontroladores:

1. Análisis de los requerimientos de la aplicación (prestaciones, consumo energético, puertos de E/S, funcionalidades, expandibilidad, actualizaciones a futuro, etc).
2. Desarrollo del hardware
  - a) Prototipado en físico usando protoboard
  - b) Prototipado en simulador (Proteus)
3. Desarrollo del algoritmo en diagrama de flujo
4. Codificación del algoritmo en un lenguaje de programación (XC8)
5. Pruebas en físico como en simulación
6. Elaboración de PCB (Autodesk Eagle)
7. Elaboración de carcasa (Autodesk Fusion 360)

17

Fin de sesión

18