

clase-02

repaso clase anterior y programa hoy (15 min)

la clase pasada aprendimos:

- presentaciones
- contexto artes mediales y Arduino
- Git, GitHub y Markdown

hoy aprenderemos:

- señales analógicas y digitales
- computadores y microcontroladores
- programar semáforo usando Processing

señales analógicas y digitales (45 min)

supondremos que nuestras señales son del mundo real:

- señales unidimensionales
- señales en función del tiempo t

más (demasiada) información:

- señales y sistemas, Alan V. Oppenheim y Alan S. Willsky.

señales analógicas

las señales analógicas tienen **valores continuos** en **todo momento**:

- notación $y(t)$
- t es continuo
- $y(t)$ es continuo

ejemplos de señales analógicas:

- presión atmosférica
- fotografía polaroid
- temperatura de un lugar
- sonido en un disco de vinilo
- salida de un micrófono

análogo significa similar, porque las señales analógicas también se parecen a otras.

una señal analógica se puede obtener desde un sensor,

ejemplos de transductores / sensores análogos:

sensor	entrada	salida
--------	---------	--------

sensor	entrada	salida
micrófono	presión atmosférica	voltaje
cápsula de guitarra	vibración de cuerda	voltaje
fotoresistor	intensidad de luz	resistencia
perilla	posición (ángulo)	resistencia

la transducción no es perfecta, siempre introduce errores y distorsión.

esto no necesariamente es malo, de hecho en música estas diferencias resultan en sabores musicales distintos. más info sobre distintos tipos de compresores <https://reverb.com/news/what-are-the-types-of-compressor-effects-the-basics>

señales digitales

las señales digitales tienen **valores discretos** en **momentos discretos**:

- notación $y[n]$
- n es discreto
- $y(t)$ es discreto

ejemplos de señales digitales:

- sonido en un disco compacto
- imagen en un computador
- tiempo en un reloj digital

una señal analógica se puede obtener desde un sensor,

ejemplos de señales digitales:

sensor	resolución	salida
integer 8 bits	presión atmosférica	voltaje
cápsula de guitarra	vibración de cuerda	voltaje
fotoresistor	intensidad de luz	resistencia
perilla	posición (ángulo)	resistencia

pausa: materiales (15 min)

los materiales necesarios para esta clase están descritos en la página principal de este repositorio.

se recomienda adquirir el kit de MCI electronics, disponible en [Mercado Libre](#) a 13.990 CLP.

coordinar con profesor si quieren comprar en grupo.

computadores y microcontroladores (45 min)

conjuntos de números

- números naturales: los que contamos con los dedos. (supuesto: cada unidad es equivalente) por qué contamos con diez dedos? 1,2,3,...
- números enteros: números que pueden ser escritos sin fracciones ..., -3, -2, -1, 0, 1, 2, 3,
- números racionales: números que pueden ser escritos como fracciones de dos números enteros.
- números reales: números con parte decimal, sirven para medir distancias.
- números irracionales: números que son reales, pero no racionales, como $\sqrt{2}$ y π .
- números complejos: números que poseen coordenadas reales e imaginarias.

sistemas de números

- sistema decimal: sistema de contar con base 10.
- sistema binario: sistema de contar con base 2.
- sistema hexadecimal: sistema de contar con base 16.

actividad: escribir 20 números

escribamos los primeros 20 números, empezando desde 0, usando los sistemas decimal, binario y hexadecimal.

actividad: qué es un computador?

la clase construye una definición de qué es un computador.

Introducción a microcontroladores

Diferencias entre computadores y microcontroladores.

Actividad:

Hacer una lista sobre qué actividades los computadores y las personas realizan de forma fácil y difícil

<https://www.random.org/>

Código y comentarios

<https://www.wolframalpha.com/input?i=2+GB+%2F+8+bit>

Qué es código

Explicar diferencia entre Microsoft Word y editores de texto.

Diferencias entre espacios y tabulaciones.

Diferencias entre UTF-8 y emojis y sistemas de Strings.

Actividad: programar semáforo en Processing (60 min)

Materiales

Comprar materiales próxima clase

Próxima clase

- Programar semáforo en Arduino