



UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO



**FACULTAD
DE INGENIERÍA**

**Licenciatura en Ciencias de la
Computación**

Sistemas Embebidos

Unidad 2

Comunicación con el exterior

GPIO (general-purpose input/output)

Puertos de entrada salida de propósito general

- Pines que pueden configurarse como entradas o salidas digitales.
 - Entrada:
 - Se lee como 1 lógico si se aplica un nivel alto de tensión en el pin.
 - Se lee como 0 lógico si se aplica un nivel bajo de tensión en el pin.
 - Salida:
 - Produce un nivel alto de tensión en el pin si se escribe un 1 lógico.
 - Produce un nivel bajo de tensión en el pin si se escribe un 0 lógico.
- Usualmente agrupados en grupos de 8 llamados “puertos”.
- Pueden tener resistencias pull-up o pull-down.
 - Pull-up: imponen un 1 lógico por defecto.
 - Pull-down: imponen un 0 lógico por defecto.

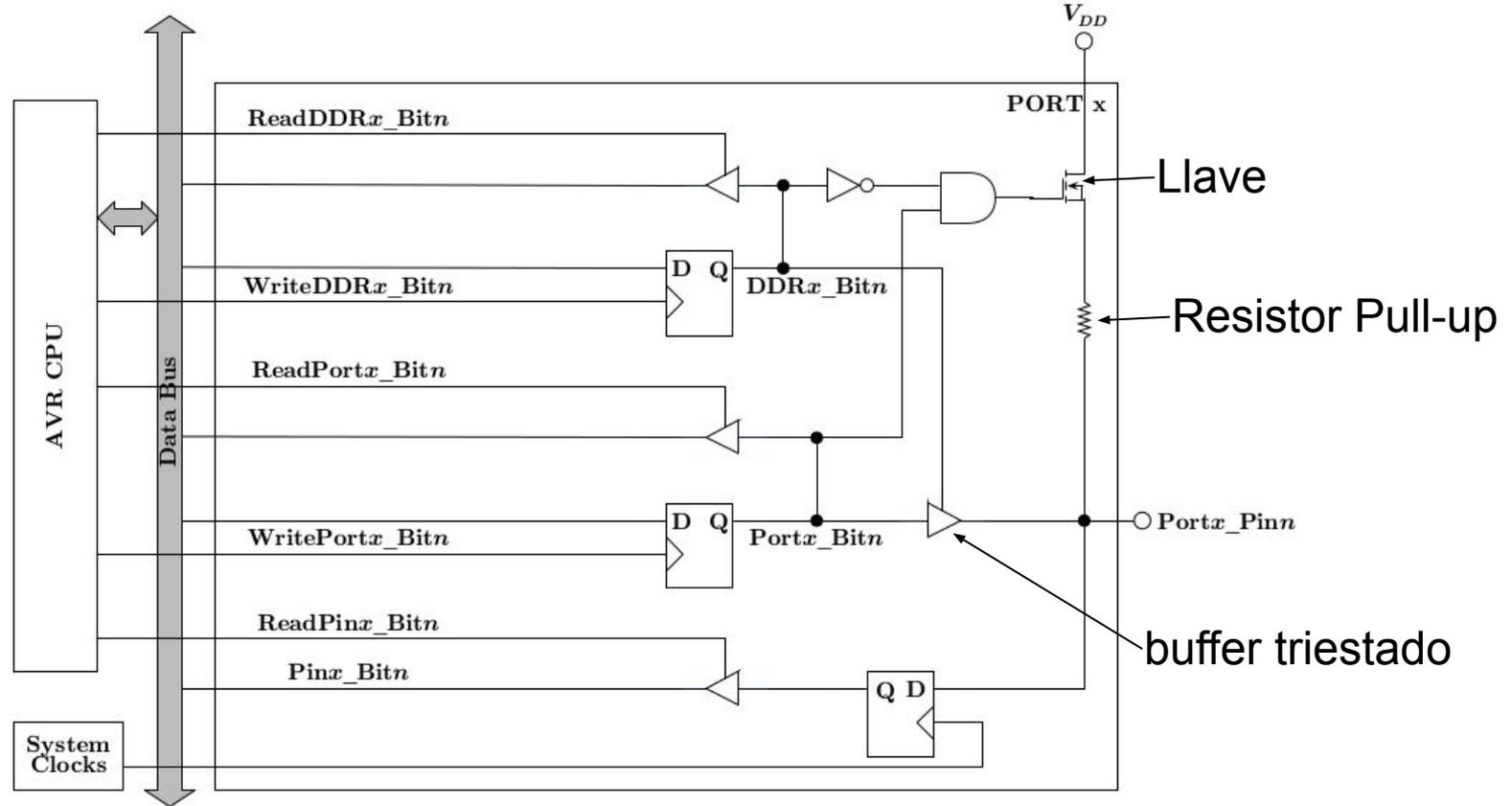


Figura obtenida de: David Russel, "Introduction to Embedded System Using ANSI C and the Arduino Development Environment", página 103



6.4.7 PORTD - THE PORT D DATA REGISTER

Salida

Bit	7	6	5	4	3	2	1	0
0x2B	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
Read/Write	R/W							
Default	0	0	0	0	0	0	0	0

- PORTD7-0: GPIO data value stored in bit n .

Registros para
manejar el
Puerto D del
ATmega 328

6.4.8 DDRD - THE PORT D DATA DIRECTION REGISTER

Dirección

Bit	7	6	5	4	3	2	1	0
0x2A	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
Read/Write	R/W							
Default	0	0	0	0	0	0	0	0

- DDRD7-0: selects the direction of pin n . If $DDRDn$ is written '1', then $PORTDn$ is configured as an output pin. If $DDRDn$ is written '0', then $PORTDn$ is configured as an input pin.

6.4.9 PIND - THE PORT D INPUT PINS ADDRESS

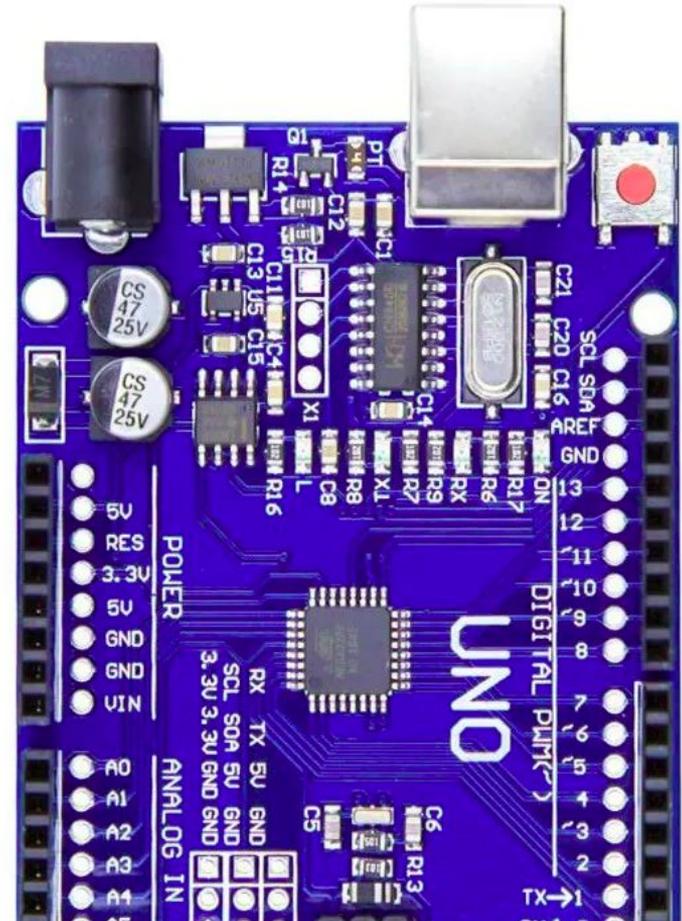
Entrada

Bit	7	6	5	4	3	2	1	0
0x29	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
Read/Write	R	R	R	R	R	R	R	R
Default	-	-	-	-	-	-	-	-

- PIND7-0: logic value present on external pin n .

Entrada Salida con el IDE de Arduino UNO

```
/*Configuración de pines como  
entrada o salida*/  
pinMode(pin, OUTPUT);  
pinMode(pin, INPUT);  
/*Leer el valor de un pin*/  
valor=digitalRead(pin);  
/*Escribe val en el pin indicado  
value puede valer HIGH o LOW. */  
digitalWrite(pin, value);
```



Ejemplo de especificaciones

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{IL}	Input Low Voltage, except XTAL1 and $\overline{\text{RESET}}$ pin	$V_{CC} = 1.8\text{V} - 2.4\text{V}$	-0.5		$0.2V_{CC}^{(1)}$	V
		$V_{CC} = 2.4\text{V} - 5.5\text{V}$	-0.5		$0.3V_{CC}^{(1)}$	
V_{IH}	Input High Voltage, except XTAL1 and $\overline{\text{RESET}}$ pins	$V_{CC} = 1.8\text{V} - 2.4\text{V}$	$0.7V_{CC}^{(2)}$		$V_{CC} + 0.5$	V
		$V_{CC} = 2.4\text{V} - 5.5\text{V}$	$0.6V_{CC}^{(2)}$		$V_{CC} + 0.5$	

Ejemplo: Si $V_{CC}=5.0\text{V}$

Un 0 será: de -0.5 V a 1.5V

Un 1 será: de 3.0 V a 5.5V

Valores máximos:

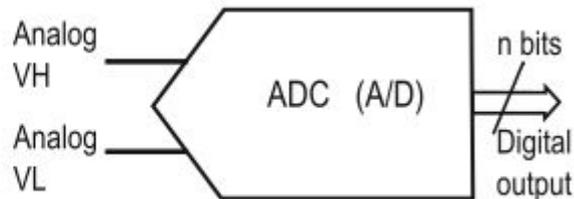
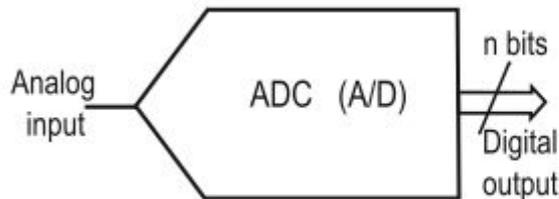
Maximum Operating Voltage	6.0V
DC Current per I/O Pin	40.0mA
DC Current V_{CC} and GND Pins	200.0mA

Entrada Salida en Raspbian

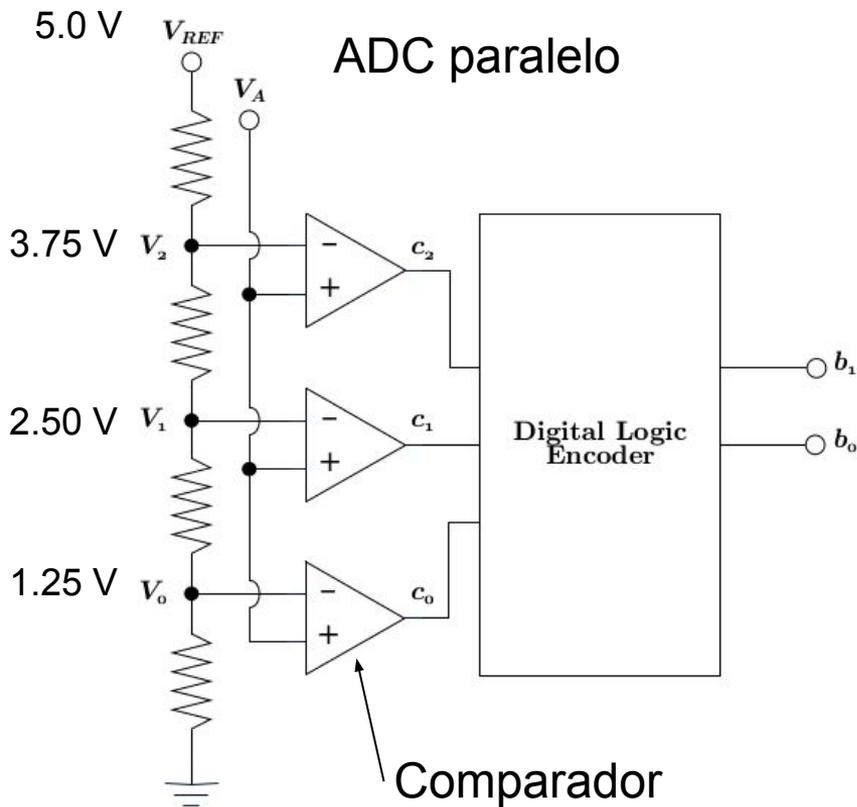
- La entrada/salida se controla mediante la escritura a archivos virtuales.
- Para habilitar un pin como GPIO, debemos escribir el número de pin en el archivo `"/sys/class/gpio/export"`.
 - Esto crea la carpeta `"/sys/class/gpio/gpioX"` siendo X el número de pin.
 - *Escribiendo in o out en `"/sys/class/gpio/gpioX/direction"` se configura como entrada o salida.*
 - *Leer o escribir el pin `"/sys/class/gpio/gpioX/value"`*
 - Para deshabilitar el GPIO usado, debe escribirse el número de pin en `"/sys/class/gpio/unexport"`.

Convertor analógico a digital

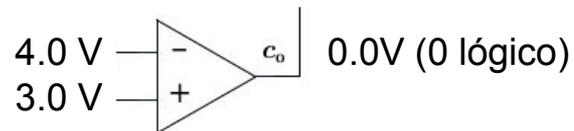
- Convertor Analógico a Digital, ADC o A/D: convierte una tensión (voltios) entre dos pines de analógico a un número digital de n bits.
- Puede tener:
 - Una entrada: mide diferencia de potencial entre la entrada y tierra (0 volts).
 - Dos entradas: mide diferencia de potencial entre esas dos entradas.



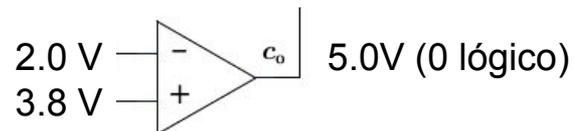
Convertor analógico a digital



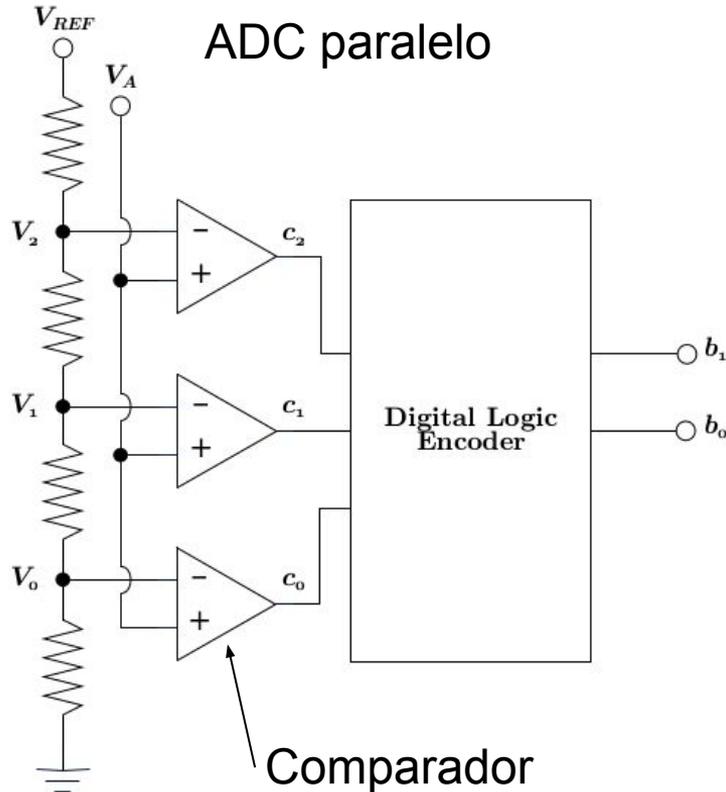
Si $V_+ < V_-$, Salida: 0



Si $V_+ > V_-$, Salida: 1



Convertor analógico a digital



- **Muy veloz**
- **Costoso (para 1024 niveles se necesitan 1023 comparadores).**

$$V_{REF} = 5.00V$$

$$\text{Resol.} = V_{REF} / 4 = 1.25V$$

$V_A = 0.7$	Salida: 00
$V_A = 1.2$	Salida: 01
$V_A = 2.3$	Salida: 01
$V_A = 3.1$	Salida: 10
$V_A = 3.9$	Salida: 11
$V_A = 4.9$	Salida: 11

Rango entrada	salida digital
0.00V a 1.25V	00
1.25V a 2.50V	01
2.50V a 3.75V	10
3.75V a 5.00V	11

Convertor analógico a digital

Cuantización y Resolución

$$V_{REF} = 5.00V$$

Resolución: 4 bits

$$\text{Resolución (LSB)} = V_{REF} / 2^4 = 0,3125V$$

(LSB: least significant bit)

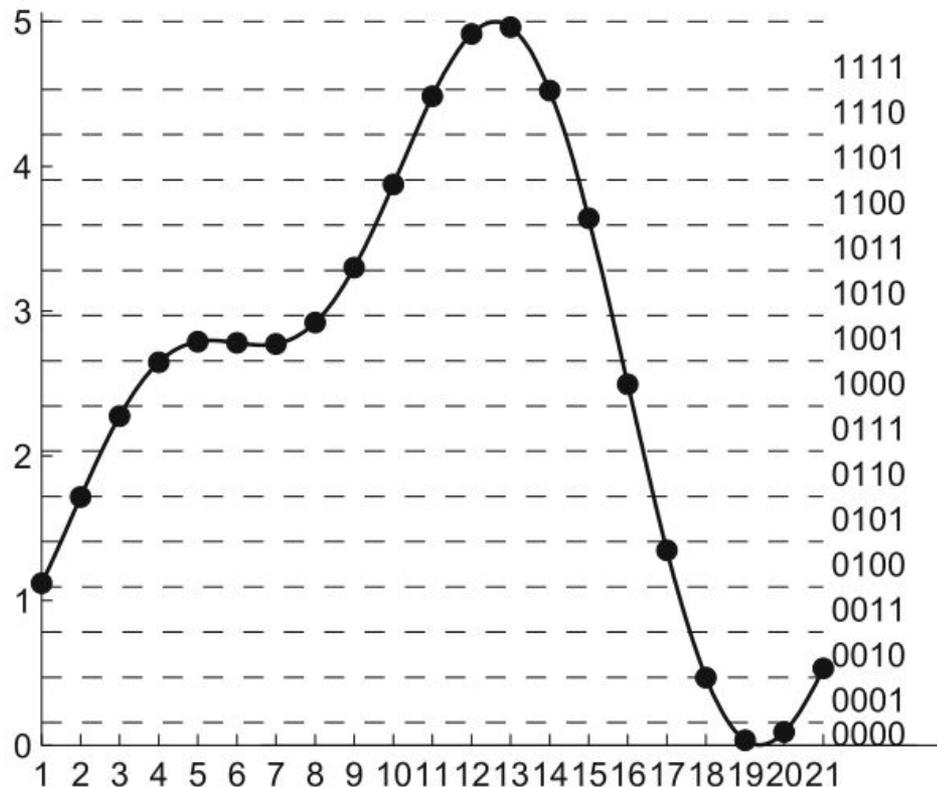
$$V_X = 1.0 \text{ Salida: } 0011$$

$$V_X = 2.0 \text{ Salida: } 0110$$

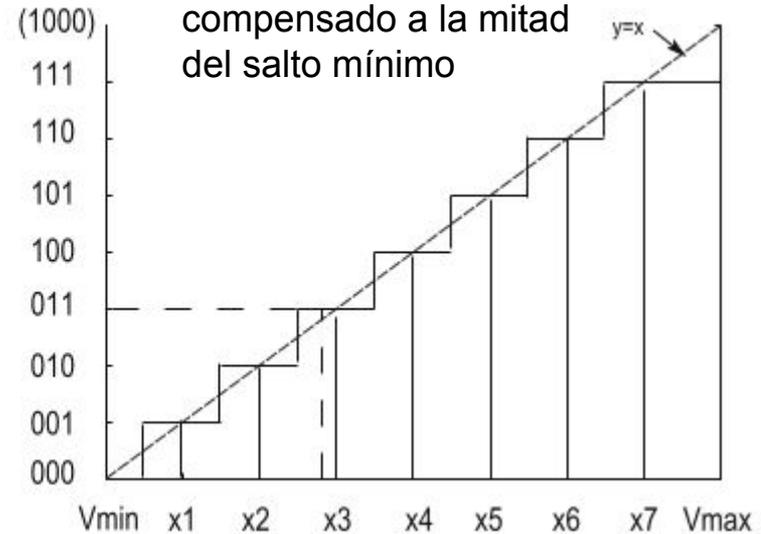
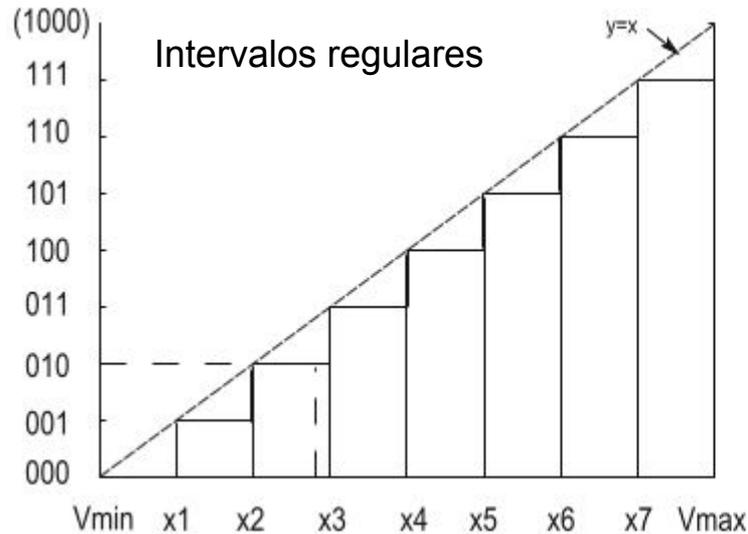
$$V_X = 1.88 \text{ Salida: } 0110 \text{ (} 1.88 / 0,3125 = 6.016 \text{)}$$

$$V_X = 2.18 \text{ Salida: } 0110$$

$$V_X = 2.19 \text{ Salida: } 0111$$



Convertor analógico a digital Diferentes formas de cuantización



Convertor analógico a digital

Resoluciones típicas
(suponiendo $V_{REF} = 5.00V$):

Resolución bits	Resolución Volts
8 bits	19.5 mV
10 bits	4.88 mV
12 bits	1.22 mV
16 bits	0.076 mV

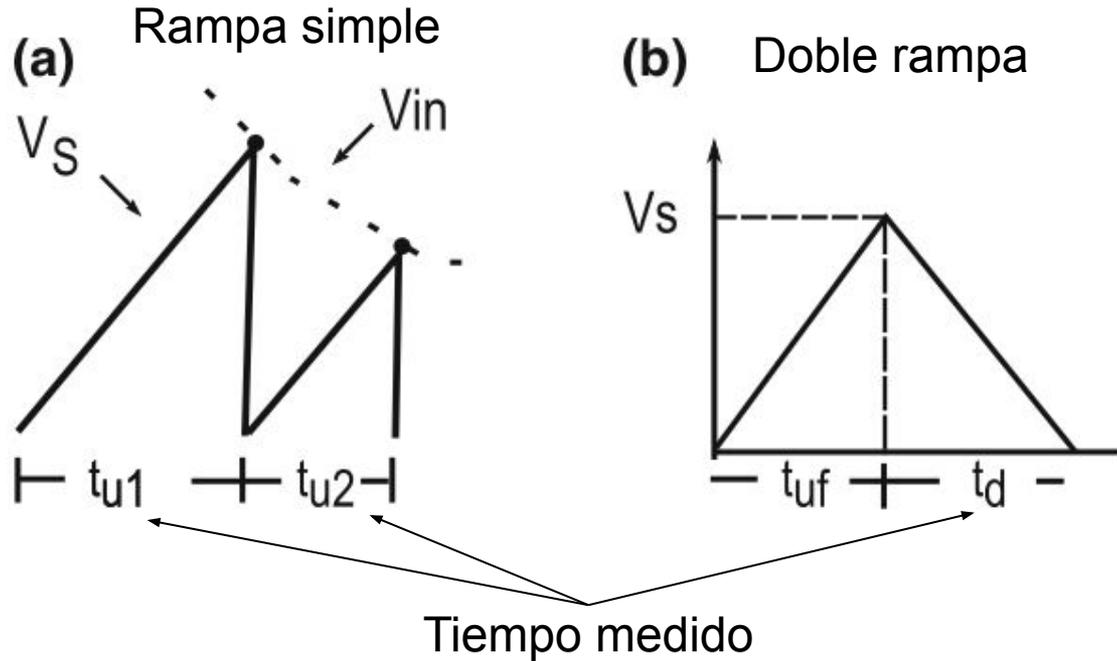
Más bits, más resolución, mayor tiempo de conversión

Exactitud

Errores producidos por diferentes factores:

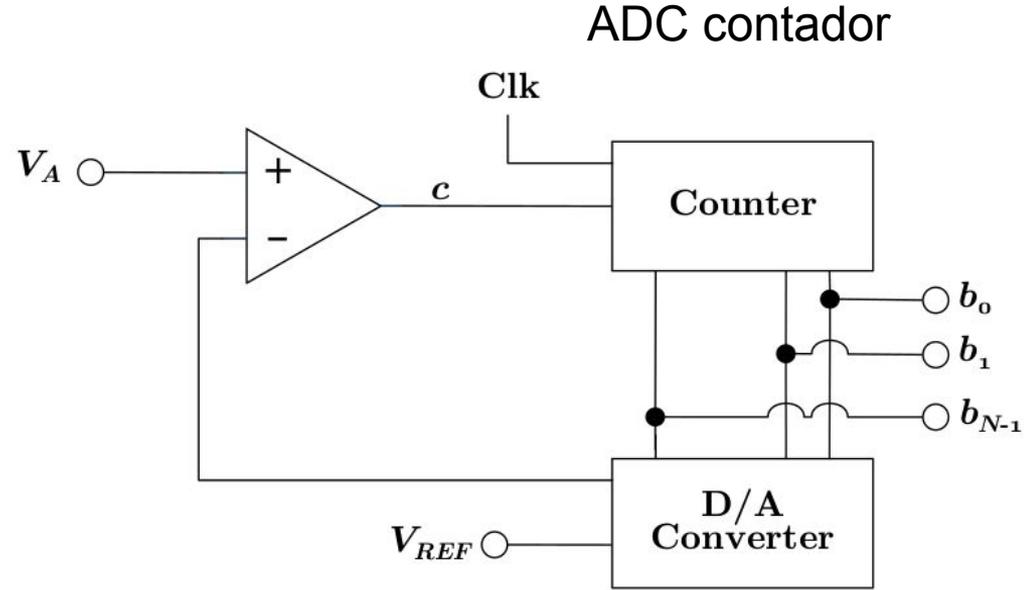
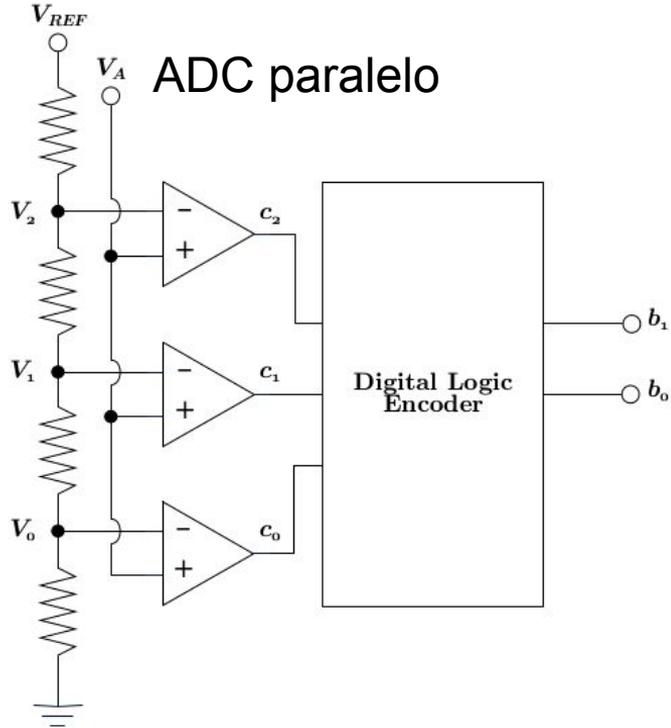
- Calidad de fabricación
- Distorsión introducida por los componentes.
- Ruido
- Usualmente especificada como un porcentaje de la resolución.

Funcionamiento interno: Rampa



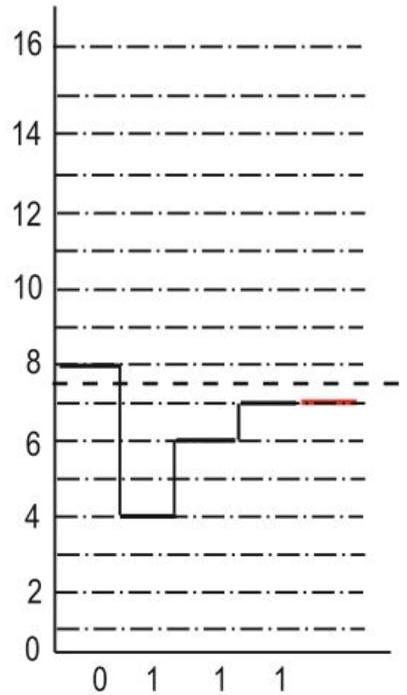
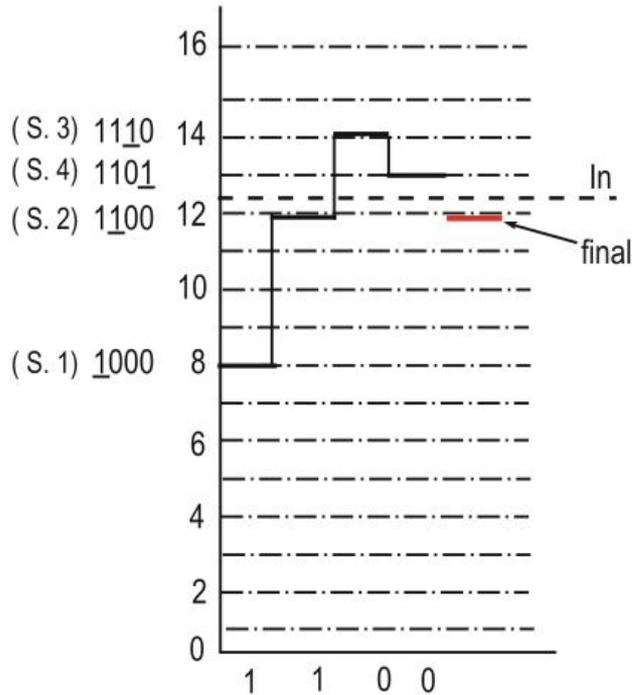
- Convierte una medición de voltaje en una medición de tiempo (necesidad de contadores).
- Ventaja: bajo costo.
- Desventaja: demora en tomar la lectura

Funcionamiento interno: Contador





Funcionamiento interno: Contador con aproximaciones sucesivas

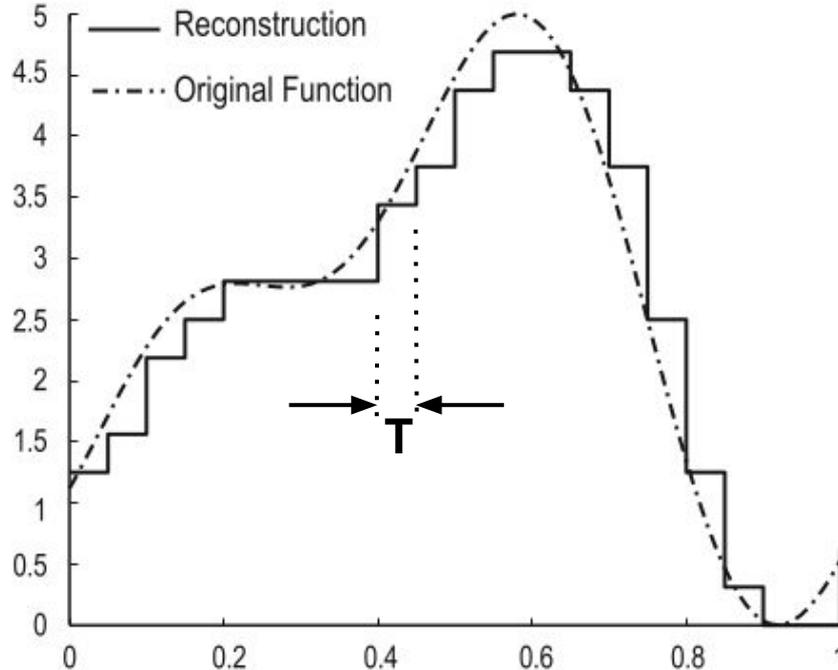


- Requiere un conversor de digital a analógico.
- Ventaja: más rápido
- Desventaja: más costoso.



Frecuencia de muestreo y principio de Nyquist

Frecuencia de muestreo



frecuencia de muestreo: $f_s = 1/T$

principio de Nyquist

Frecuencia de muestreo para no
perder información o para poder
reconstruir la señal:

$$f_s \geq 2f_{\text{MAXIMA}}$$

**Ejemplo de conversor A/D del
microcontrolador MSP430**

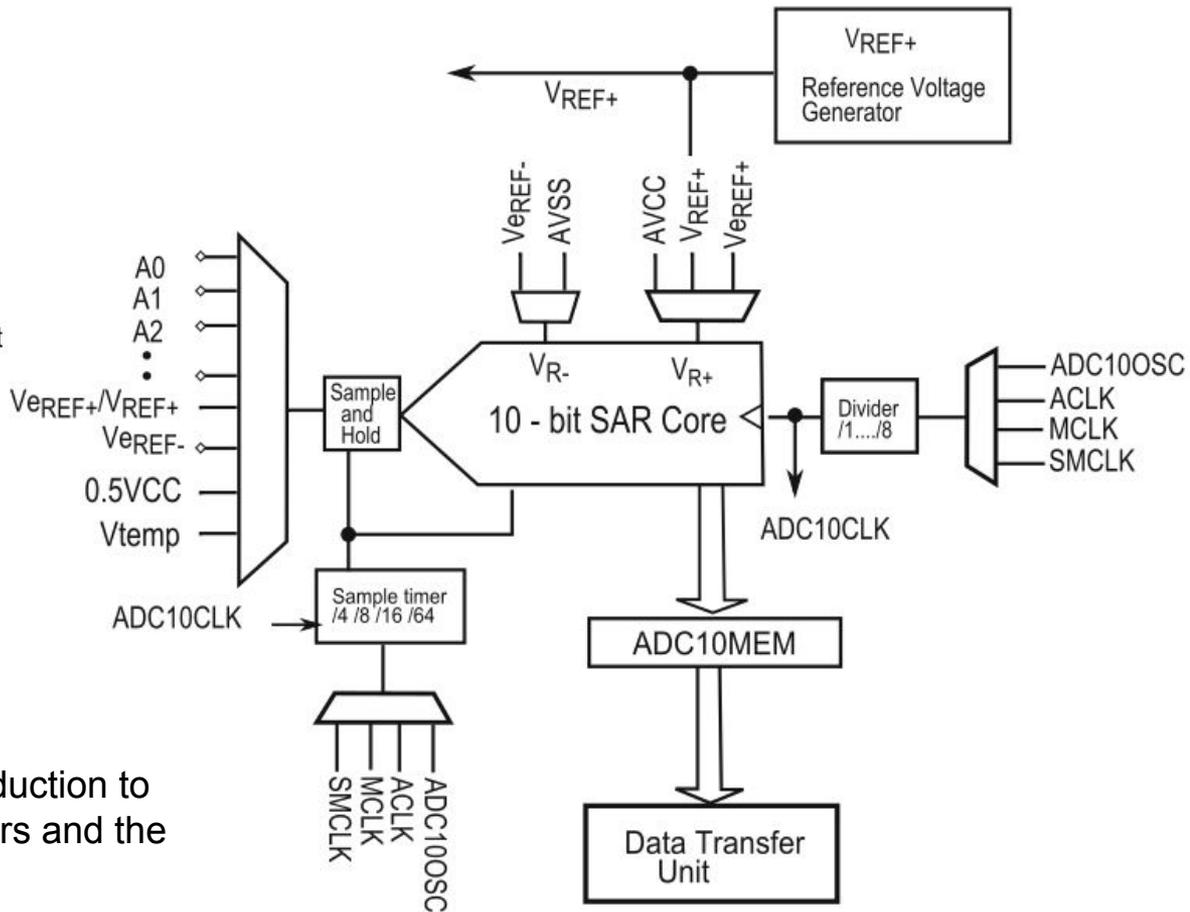
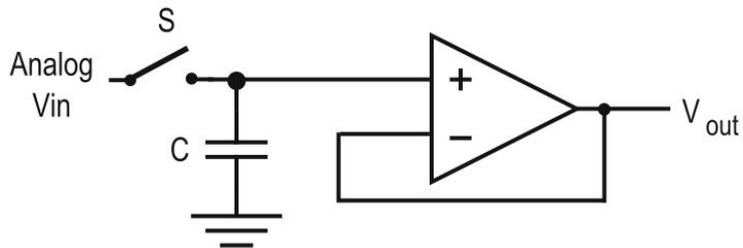


Figura obtenida de: Jiménez et. al. "Introduction to Embedded Systems Using Microcontrollers and the MSP430", página 586



Ejemplo: Conversor AD Atmega 328

28.1. Features

- 10-bit Resolution
- 0.5 LSB Integral Non-Linearity
- ± 2 LSB Absolute Accuracy
- 13 - 260 μ s Conversion Time
- Up to 76.9kSPS (Up to 15kSPS at Maximum Resolution) SPS: samples per second
- Six Multiplexed Single Ended Input Channels
- Two Additional Multiplexed Single Ended Input Channels (TQFP and VFQFN Package only)
- Temperature Sensor Input Channel
- Optional Left Adjustment for ADC Result Readout
- 0 - V_{CC} ADC Input Voltage Range
- Selectable 1.1V ADC Reference Voltage
- Free Running or Single Conversion Mode
- Interrupt on ADC Conversion Complete
- Sleep Mode Noise Canceler

$$V_{REF}=5.0V$$

$$LSB=5.0V/(2^{10})=0,00488V$$

$$Lectura=X\pm 0,00977 V$$

Programación del ADC en ensamblador (ATmega 328)

Name: ADMUX

Offset: 0x7C

Reset: 0x00

Property: -

Eligen el pin

Bit	7	6	5	4	3	2	1	0
	REFS1	REFS0	ADLAR		MUX3	MUX2	MUX1	MUX0
Access	R/W	R/W	R/W		R/W	R/W	R/W	R/W
Reset	0	0	0		0	0	0	0

REFS[1:0]	Voltage Reference Selection
00	AREF, Internal V_{ref} turned off
01	AV_{CC} with external capacitor at AREF pin
10	Reserved
11	Internal 1.1V Voltage Reference with external capacitor at AREF pin



Programación del ADC en ensamblador (ATmega 328)

Name: ADCSRA

Offset: 0x7A

Reset: 0x00

Property: -

Bit	7	6	5	4	3	2	1	0
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

ADEN: encender/apagar el AD

ADSC: comenzar la conversión. Mientras la conversión está en proceso, se lee como 1. Cuando la conversión termina, se lee como 0.

ADATE: Autoconversión con flanco positivo.

ADIF y ADIE: Interrupt Flag e Interrupt Enable, respectivamente.

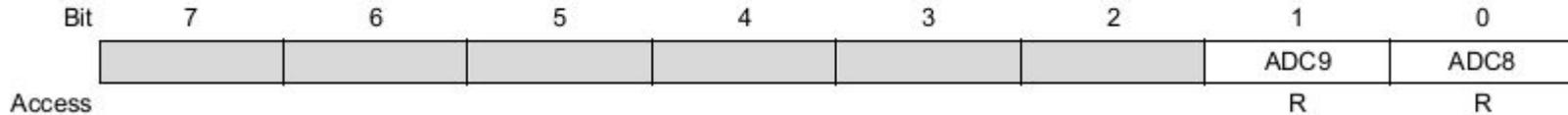
ADPS2 - ADPS0: Prescaler del reloj



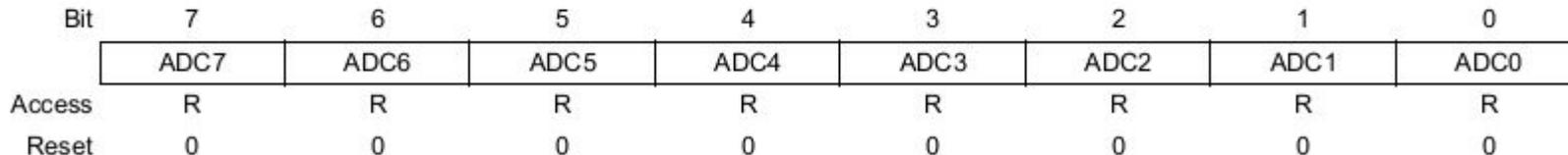
Programación del ADC en ensamblador (ATmega 328)

Name: ADCH
Offset: 0x79
Reset: 0x00
Property: ADLAR = 0

Resultado de la conversión



Name: ADCL
Offset: 0x78
Reset: 0x00
Property: ADLAR = 0





Conversor A/D en Arduino

```
analogReference(DEFAULT);
```

DEFAULT: 5 Volts
INTERNAL: 1.1 V
EXTERNAL: Voltaje aplicado al pin AREF
otras opciones dependiendo del modelo...

```
pinMode(A0, INPUT);
```

Sin resistencia pullup (sin valor por defecto)

```
pinMode(A0, INPUT_PULLUP);
```

Con resistencia pullup (valor por defecto = V_{REF})

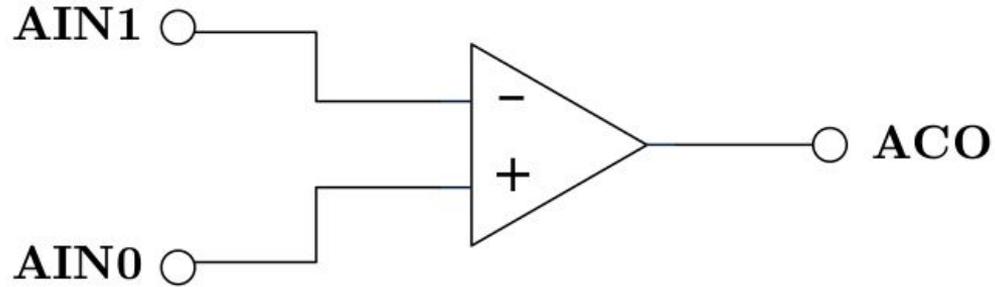
```
analogReadResolution(bits);
```

Solo algunos modelos

```
valor=analogRead(A0);
```



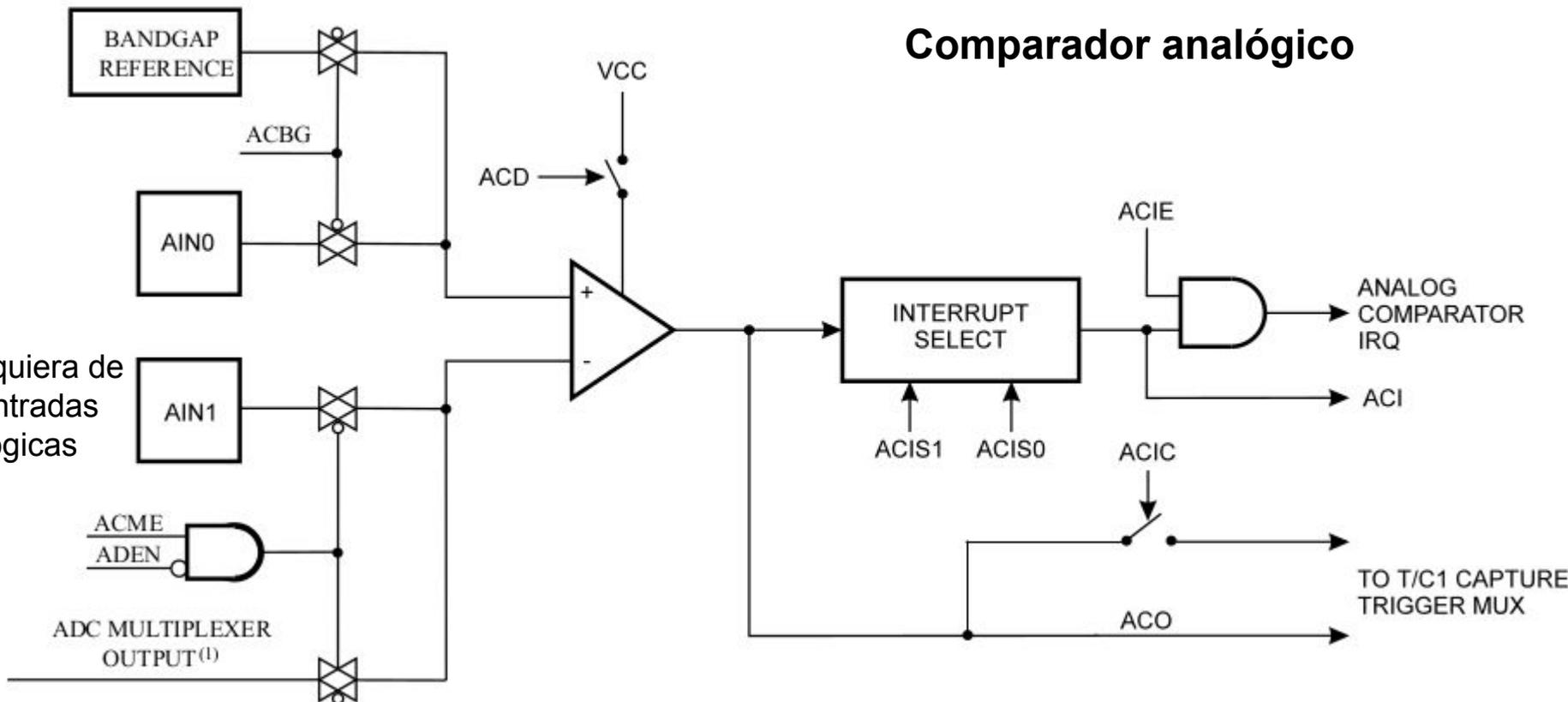
Comparador analógico



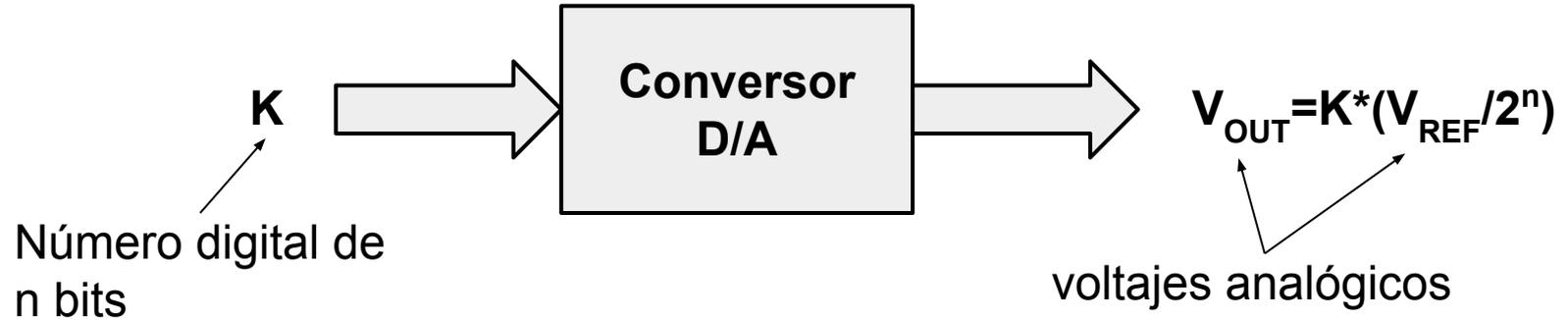


Comparador analógico

Cualquiera de las entradas analógicas



Conversor Digital a analógico



Ejemplo: $V_{REF} = 5.00V$ y tenemos 8 bits
Resolución = $V_{REF} / 2^n = 5.00V / (2^8) = 0.01953V$

Si $K = \text{bx}10010010$ (146); $V_{OUT} = 2.85V$
Si $K = \text{bx}00110000$ (48); $V_{OUT} = 0.937V$

V_{REF} puede ser interno o externo

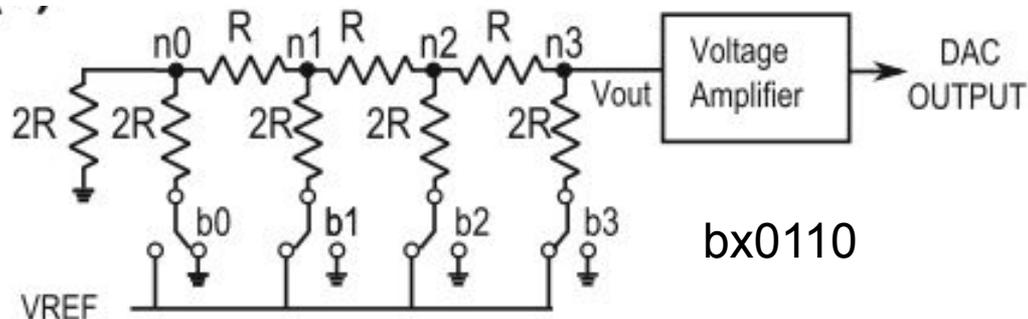
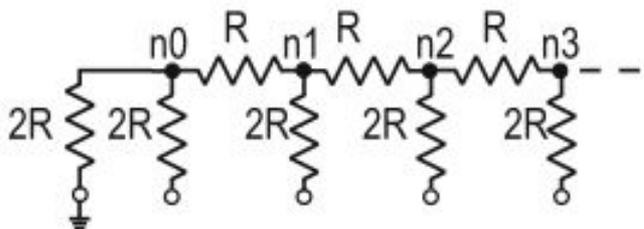
Convertor Digital a analógico

- Control de velocidad de motores
- Control de calefacción
- Control del nivel de iluminación
- Reconstruir una señal (por ejemplo: audio)

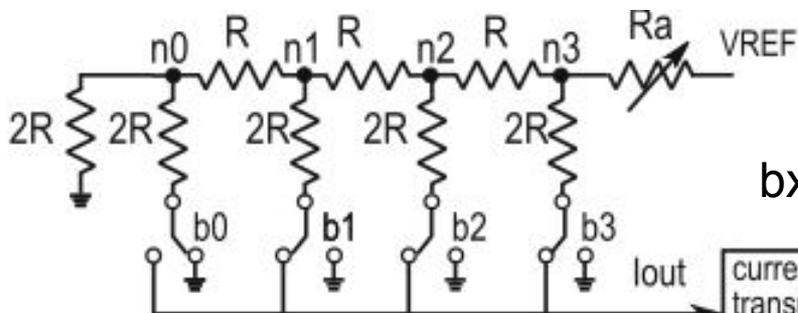




Conversor Digital a analógico: Conversor R-2R



bx0110

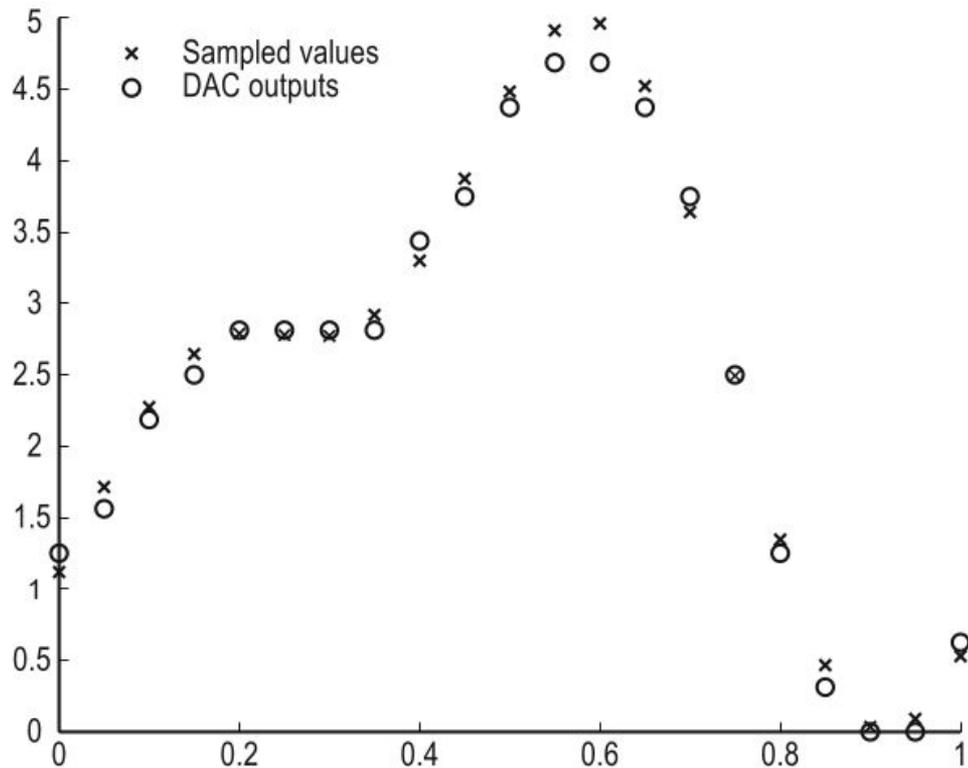
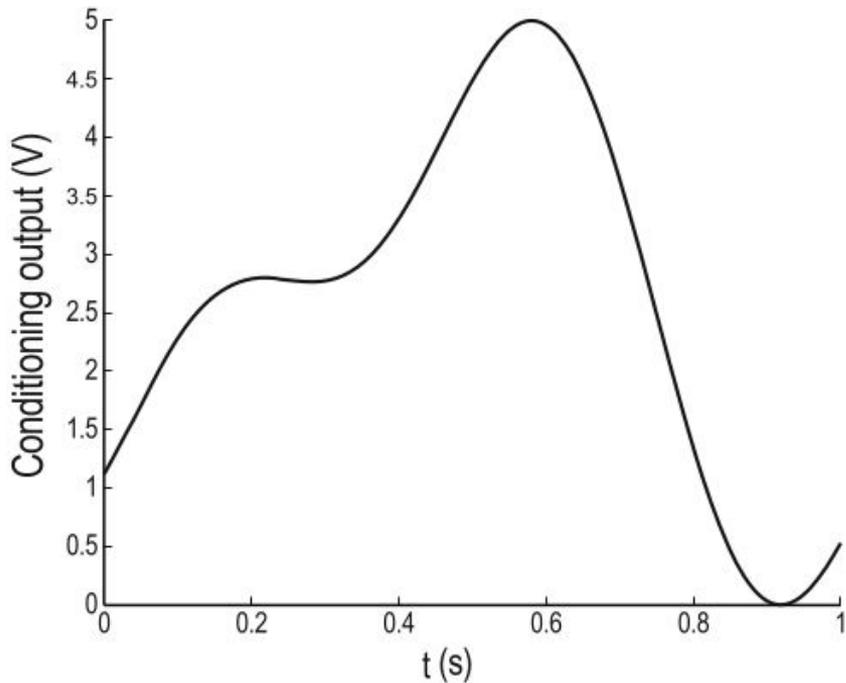


$$V_{th} = \frac{V_{REF}}{2^n} \left(2^{n-1} b_{n-1} + 2^{n-2} b_{n-2} + \dots + 2 b_1 + b_0 \right)$$

bx1110

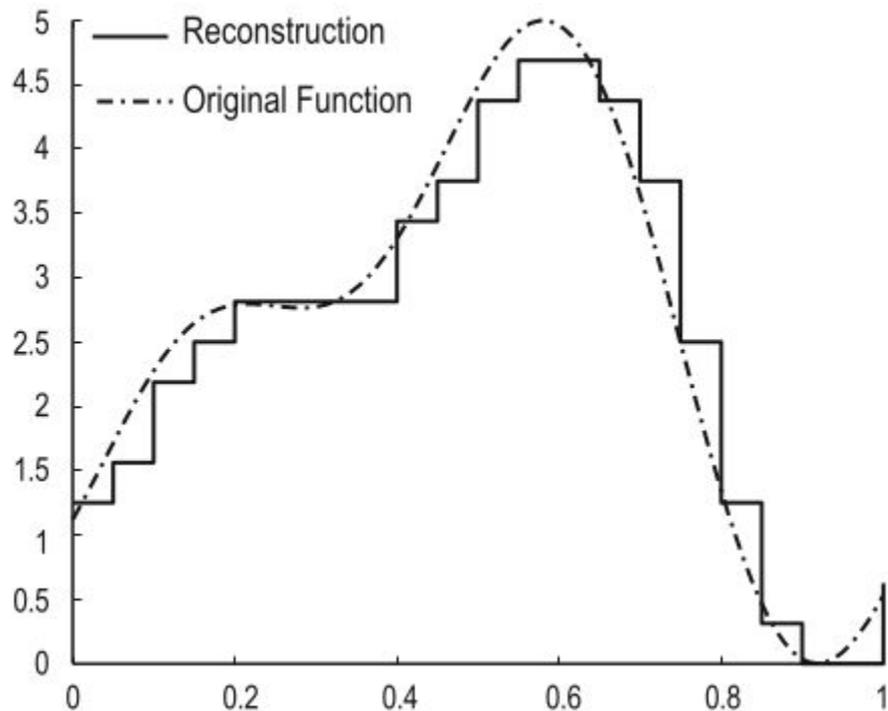
$R_a \ll R$; Output at virtual ground.

Conversor Digital a Analógico - Efecto de la resolución en la Reconstrucción





Convertor Digital a Analógico - Reconstrucción



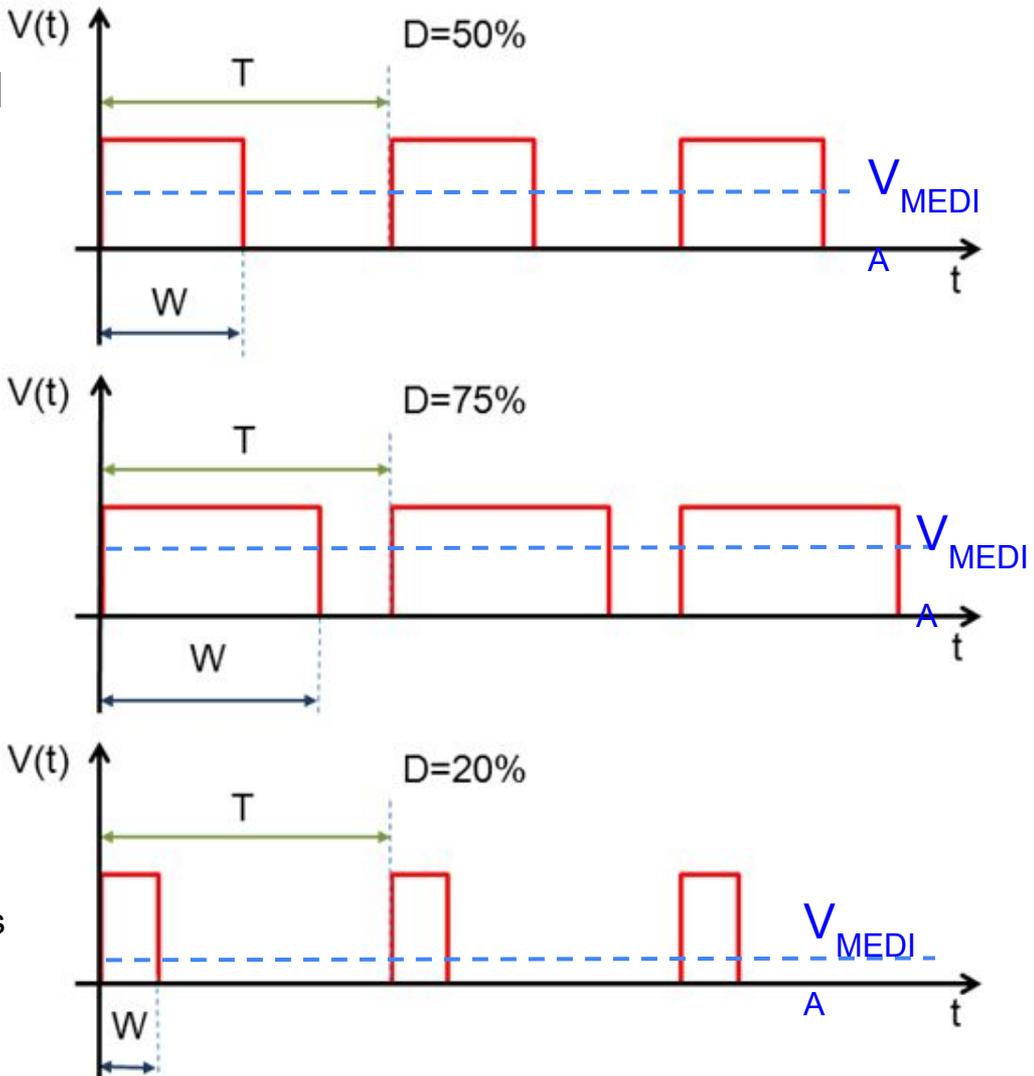


Convertor Digital a analógico Convertor basado en PWM (Pulse Width Modulation)

Duty cycle: fracción del tiempo total que el pulso está en alto.

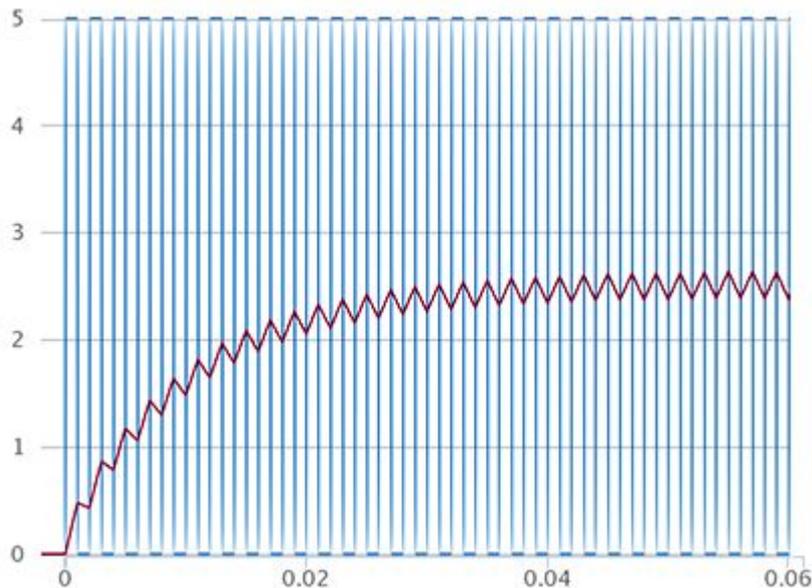
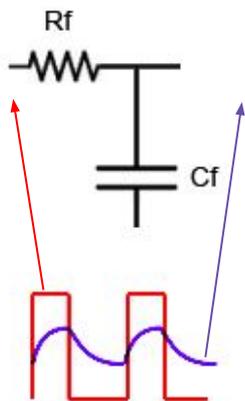
$$V_{MEDI} = V_{REF} * Duty_cycle$$

Figura basada en: Chaves Osorio, José & Quintero, Edwin & Cortes, Jimy. (2011). Generación de señales senoidales mediante PWM y filtros activos de segundo orden.

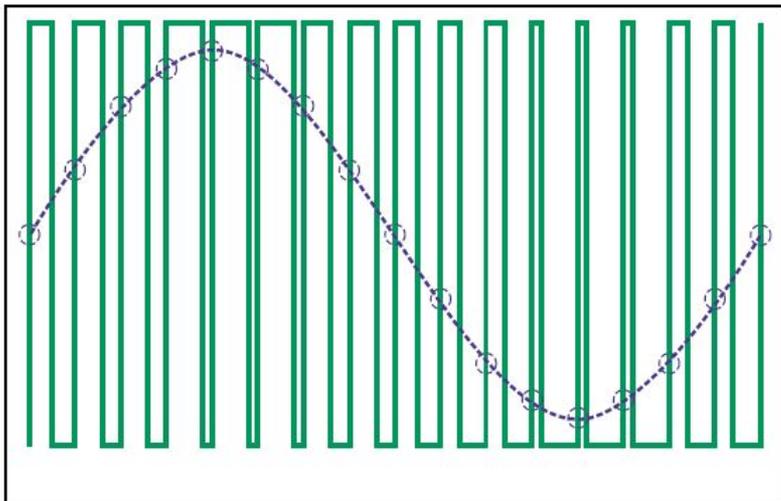




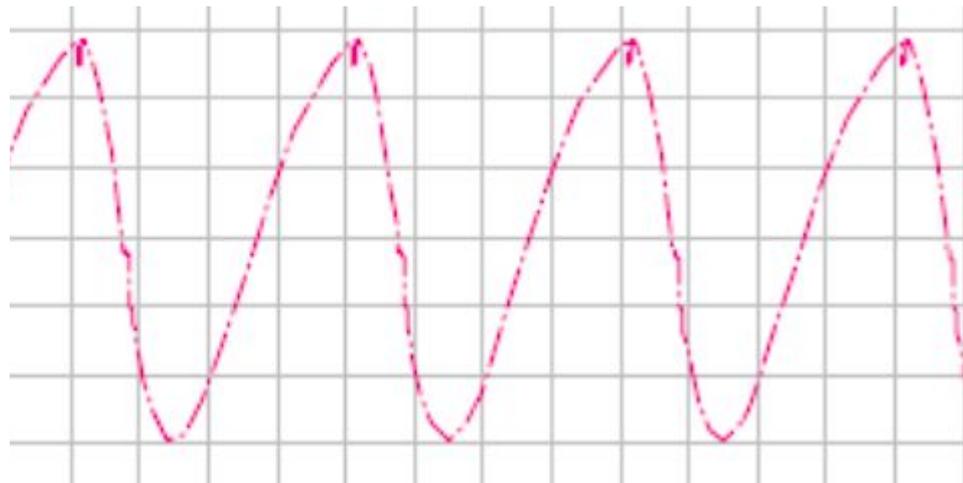
Reconstrucción de una señal analógica a partir de un PWM



Reconstrucción de una señal analógica a partir de un PWM



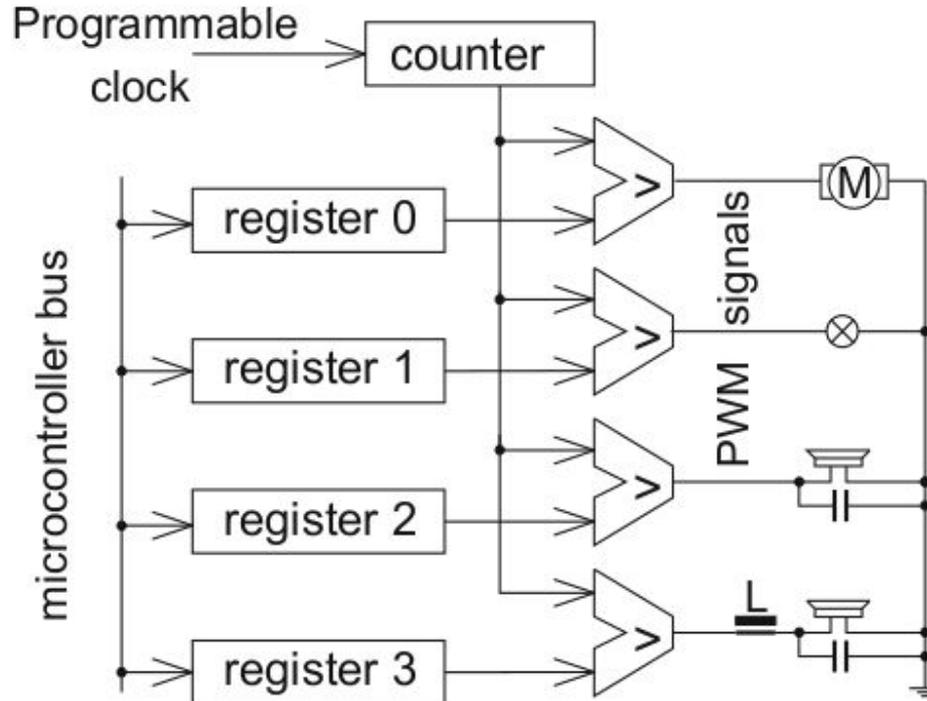
PWM y señal senoidal (ideal)



Señal senoidal resultante de un
convertor D/A tipo PWM más filtro

Figura basada en: Chaves Osorio, José & Quintero, Edwin & Cortes, Jimy. (2011). Generación de señales senoidales mediante PWM y filtros activos de segundo orden.

Convertor Digital a analógico Convertor basado en PWM (Pulse Width Modulation)



Convertor D/A en Arduino

- Convertor tipo PWM sin filtro pasa bajos (si es necesario, debe agregarse externamente) de 8 bits (256 niveles) o algunos modelos 12 bits (4096 niveles).

BOARD	PWM PINS	PWM FREQUENCY
Uno, Nano, Mini	3, 5, 6, 9, 10, 11	490 Hz (pins 5 and 6: 980 Hz)
Mega	2 - 13, 44 - 46	490 Hz (pins 4 and 13: 980 Hz)
Leonardo, Micro, Yún	3, 5, 6, 9, 10, 11, 13	490 Hz (pins 3 and 11: 980 Hz)

```
pinMode(pin, OUTPUT);
```

```
analogWriteResolution(resolución);
```

 Algunos modelos permiten elegir entre 8 y 12 bits.

```
analogWrite(pin, valor);
```

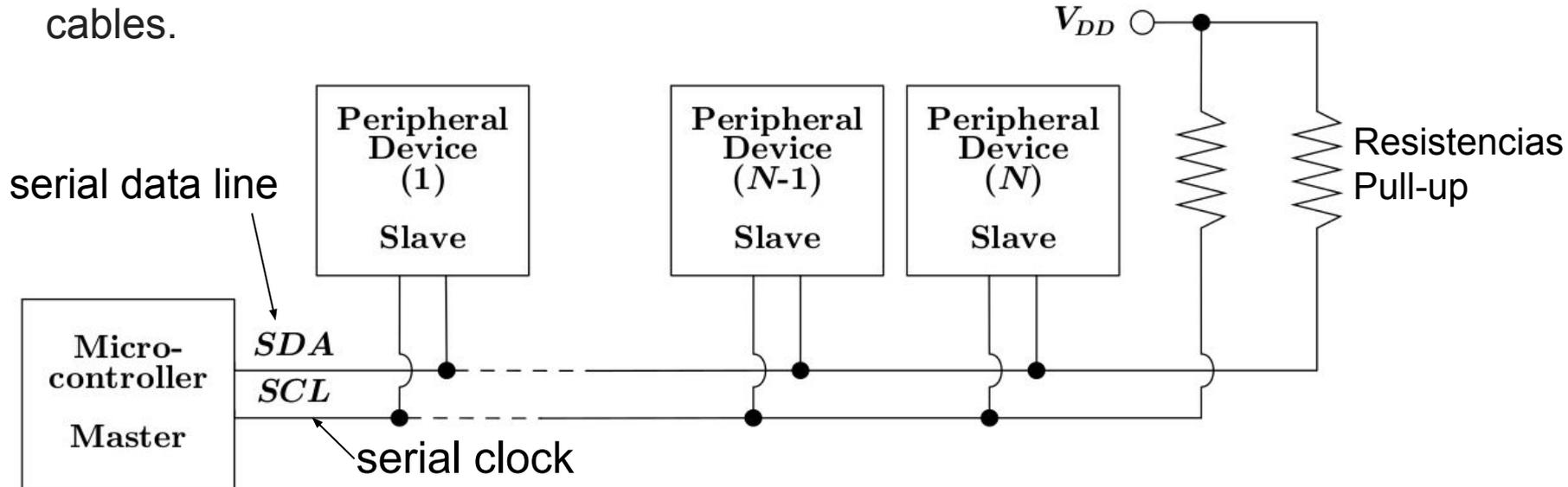
 valor: entre 0 y 256 o 4096 (algunos modelos)

Protocolos de comunicación

- Algunos microcontroladores tienen poder de procesamiento suficiente para utilizar IP, TCP y UDP, poseen Ethernet o WiFi, incluso permiten instalar servidores web (por ejemplo: Apache). Ejemplo: Raspberry.
 - **Ventaja: El sistema puede comunicarse directamente con otras computadoras por medios usuales (servicio web, ssh, etc.).**
 - **Desventaja: Requiere un microcontrolador con elevado poder de procesamiento, memoria y puertos Ethernet o Wifi.**
- Sistemas embebidos de menor poder de procesamiento o sensores específicos: Se requieren protocolos que requieran menores requisitos
 - SPI, I²C, CAN, RFID y NFC, ZigBee, LoRa.

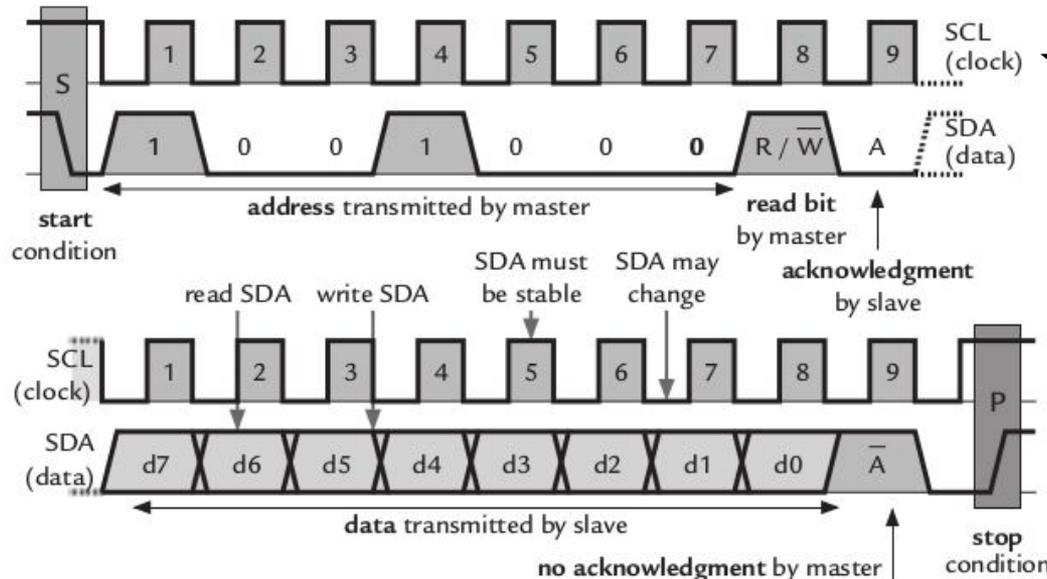
Protocolo I²C

- I²C o IIC (Inter-Integrated Circuit)
- Protocolo serial creado por NXP Semiconductors.
- Objetivo: Interconectar periféricos de baja velocidad a través de un bus de dos cables.



Protocolo I²C

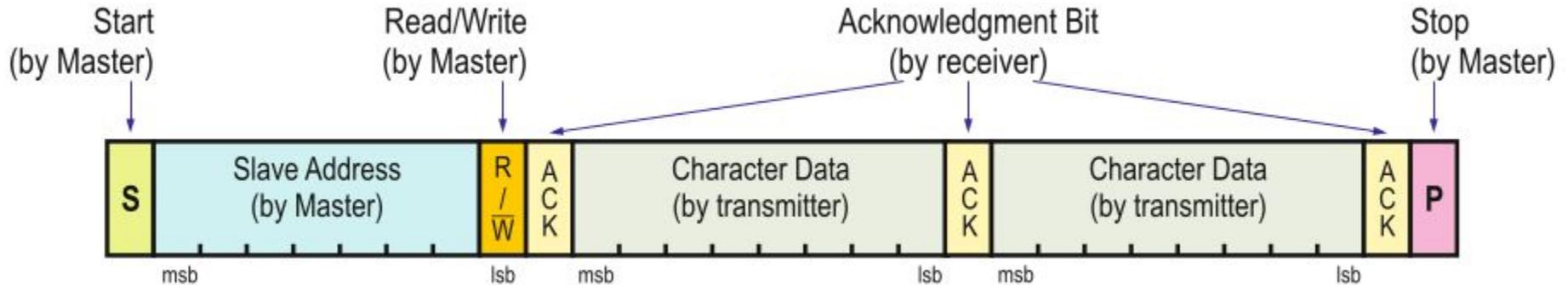
- La línea SDA (datos) es bidireccional. La línea SCL (clock) la controla el master.
- Un master, uno o más slaves (hasta 128).
- Protocolo basado en polling.



Solo el master controla el clock

Protocolo I²C

- Datos: no hay límite establecido en el protocolo de la cantidad de datos.



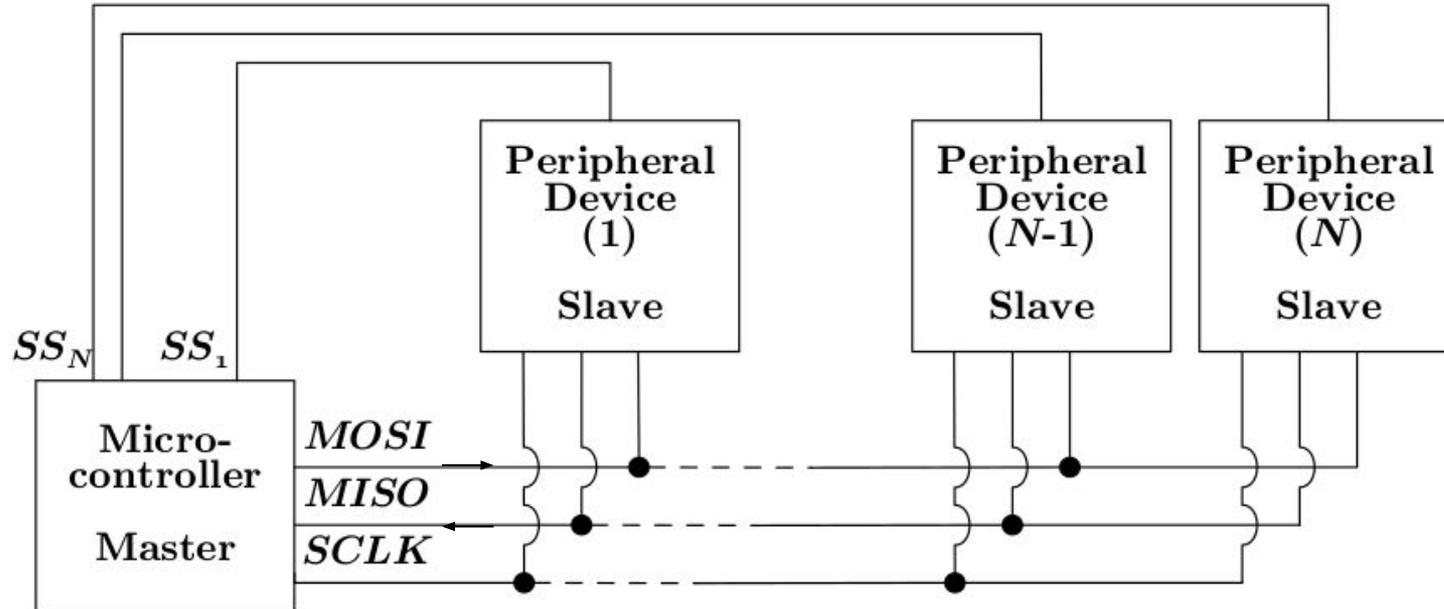
Protocolo I²C

- Direcciones: 7 bits
 - Los bits superiores usualmente fijos de fábrica según función.
 - Ejemplo: Conversores AD tienen direcciones 1001XXX.
 - Los bits inferiores pueden configurarse de diferentes maneras:
 - Grabados en una ROM o EPROM
 - Conectando pines a voltajes positivos para 1 y a 0 Volt para 0.
 - Direcciones 0000xxx y 1111xxx están reservadas.
 - Ejemplo: la dirección 000000 con RW=0 resetea todo el sistema.
 - Existen algunas extensiones (por ejemplo, direcciones de 10 bits).
- Curiosidad: El bus SMBus está basado en I²C.



Protocolo SPI (Serial Peripheral Interface)

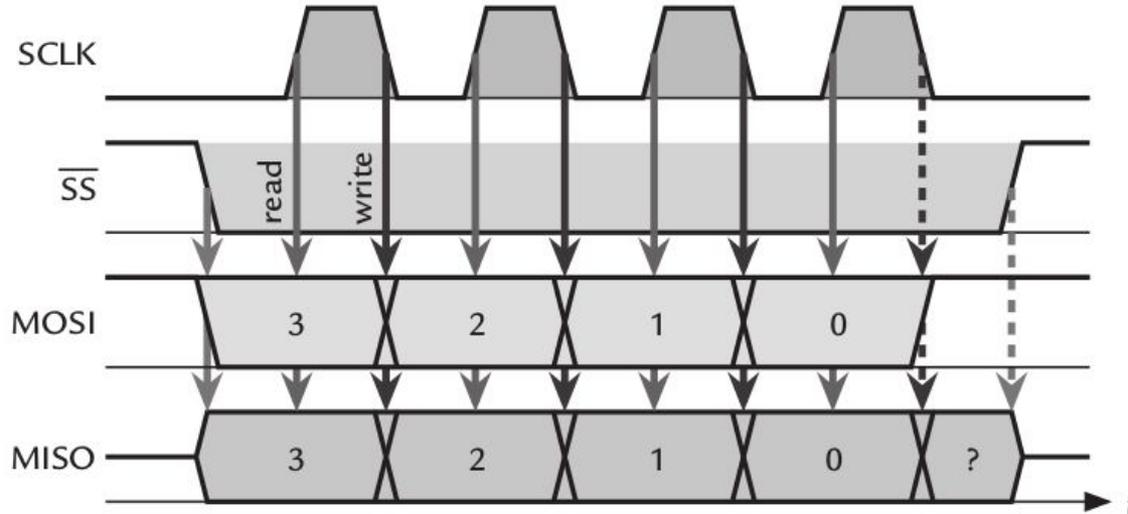
- Creado por Motorola.
- Bus de 3 cables full duplex.



SCLK: serial clock
MOSI:
master-out/slave-in
MISO:
master-in/slave-out
SS: slave-select

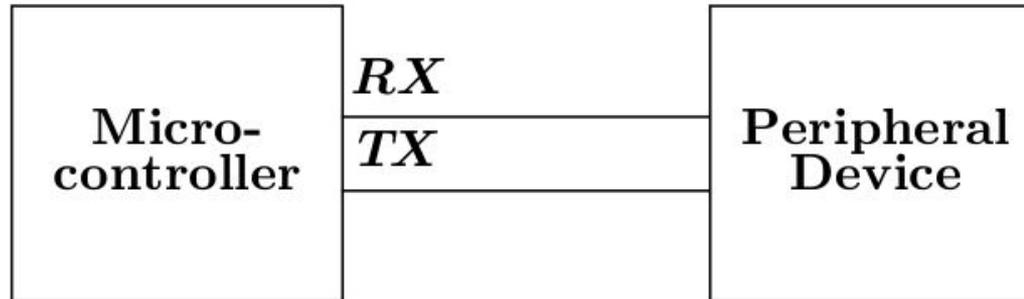
Protocolo SPI

- Muchas variaciones según los fabricantes.
- 8 Mbps (Arduino).
- Un master y uno o más slaves.
- El master elige el slave con el cual comunicarse a través de las líneas SSi.

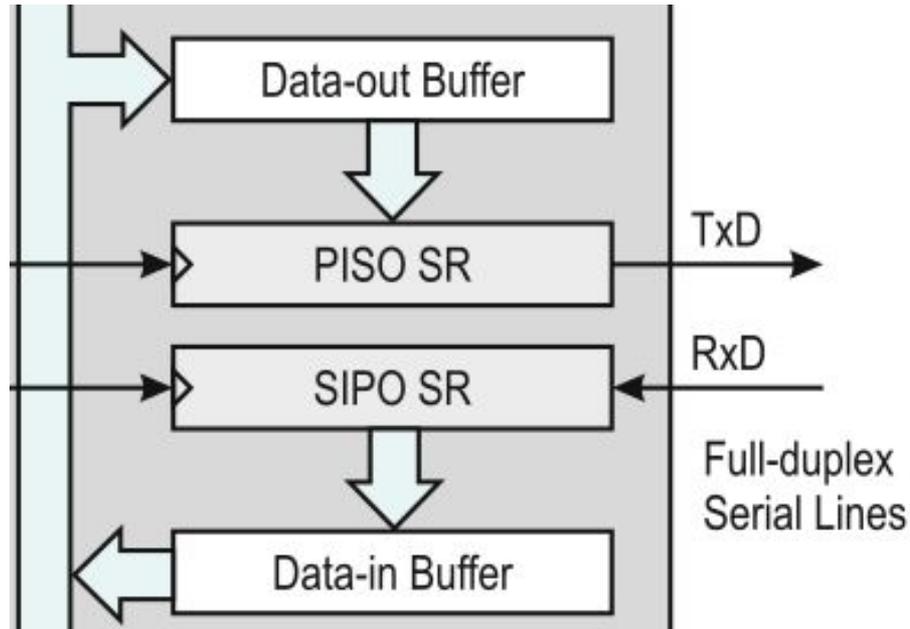


UART (universal asynchronous receiver/transmitter)

- No hay línea de clock. Con cada bits de start se sincronizan los relojes.
- Emisor y receptor son básicamente registros de desplazamiento.
- Transmisor y receptor deben acordar bps, longitud de palabra (7 u 8 bits), bits de stop y paridad.

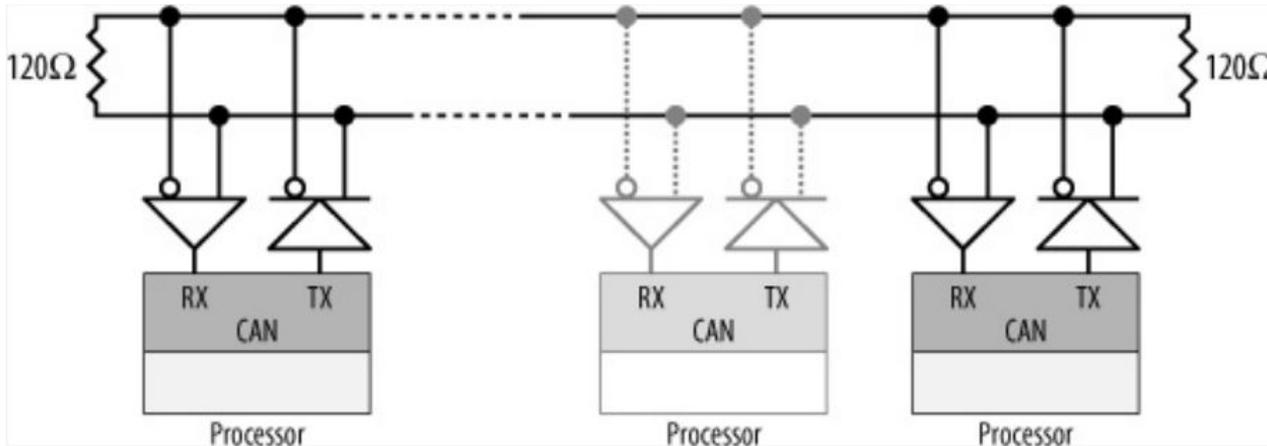


UART (universal asynchronous receiver/transmitter)



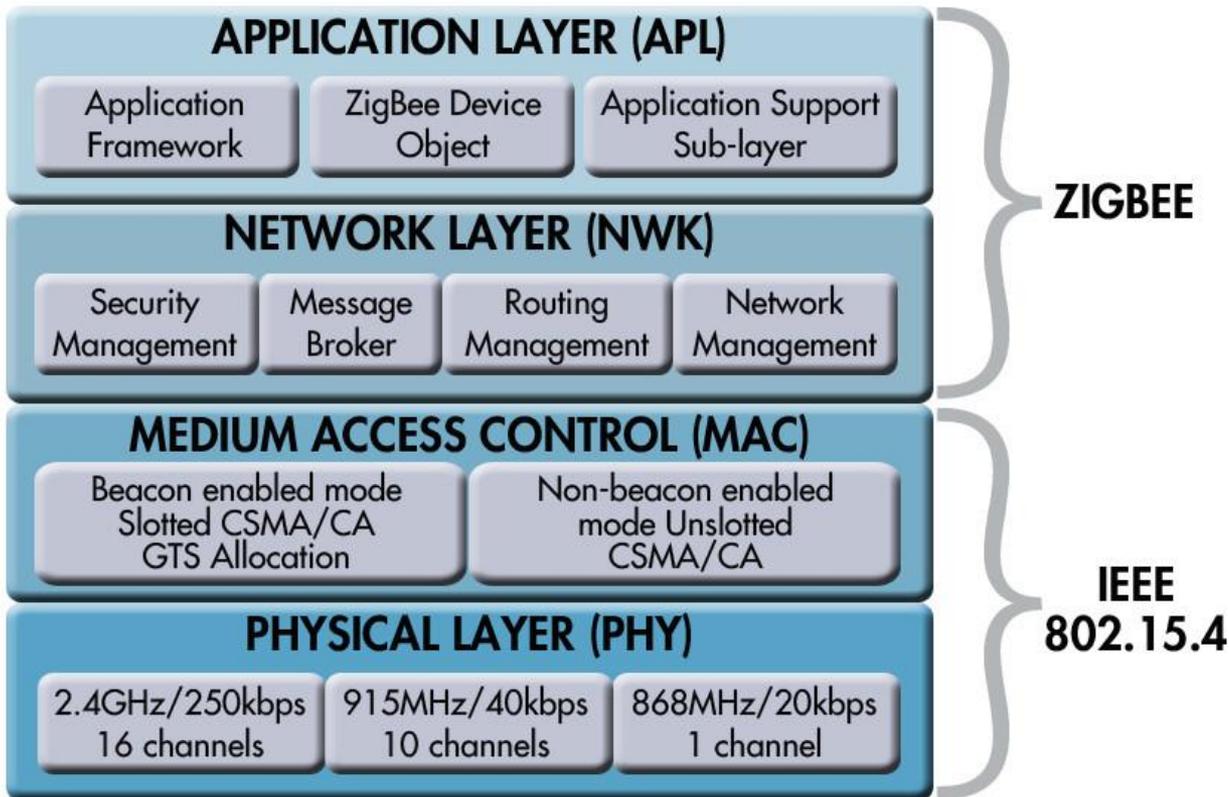
CAN (Controller Area Network)

- Red creada inicialmente para comunicar dispositivos dentro de un automóvil.
 - Su confiabilidad hizo que se adoptara en otros ámbitos (medicina, trenes, etc.)
- Especifica una capa física para red serial de dos alambres de hasta un 1 Mbps.





IEEE802.15.4 v ZiGBee





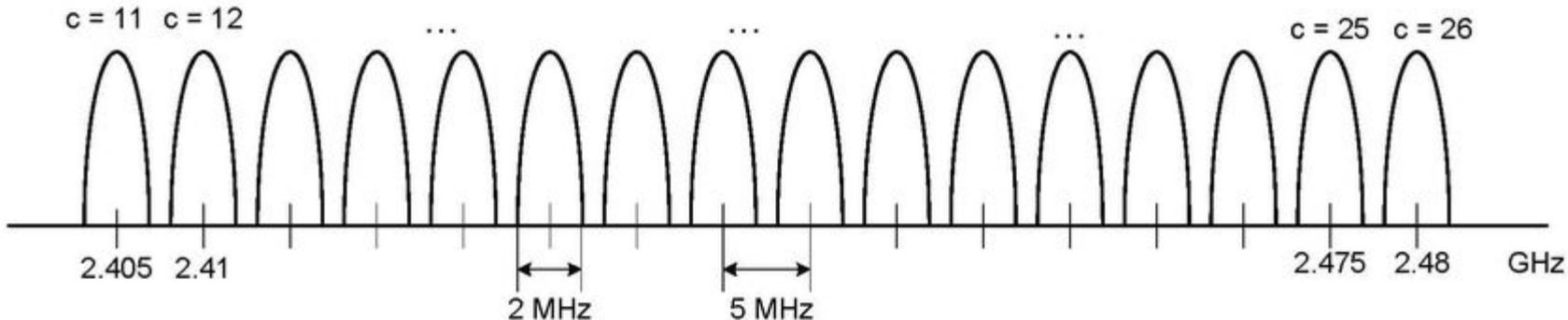
IEEE-802.15.4

- Comunicación WPAN o WLAN.
- Capas Física y MAC de baja velocidad (250 kbps).
- Utilizados por varios protocolos de ruteo de datos: ZigBee, 6LowPAN.
- Frecuencias de operación ISM: 868–868.6 MHz, 902–928 MHz, y 2400–2483.5 MHz.
- Tamaño máximo trama: 127 Bytes.
- Dos dispositivos:
 - RFD (Reduced Function Device). Solo actúa como dispositivo final.
 - FFD (Full Function Device): Pueden rutear datos o coordinar la red.
- Direccionamiento MAC de 64 bits

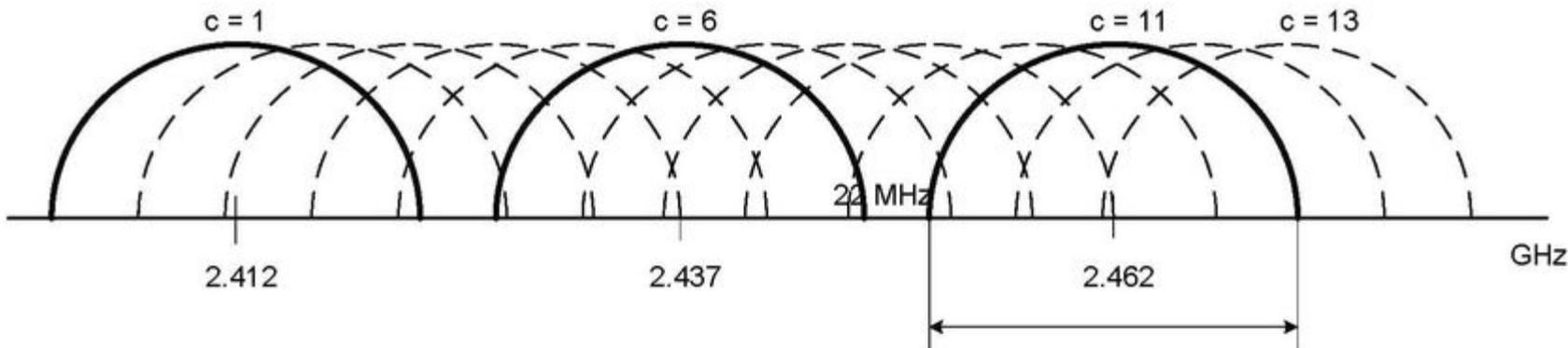


IEEE-802.15.4

IEEE 802.15.4 Channels



IEEE 802.11g Channels





ZigBee

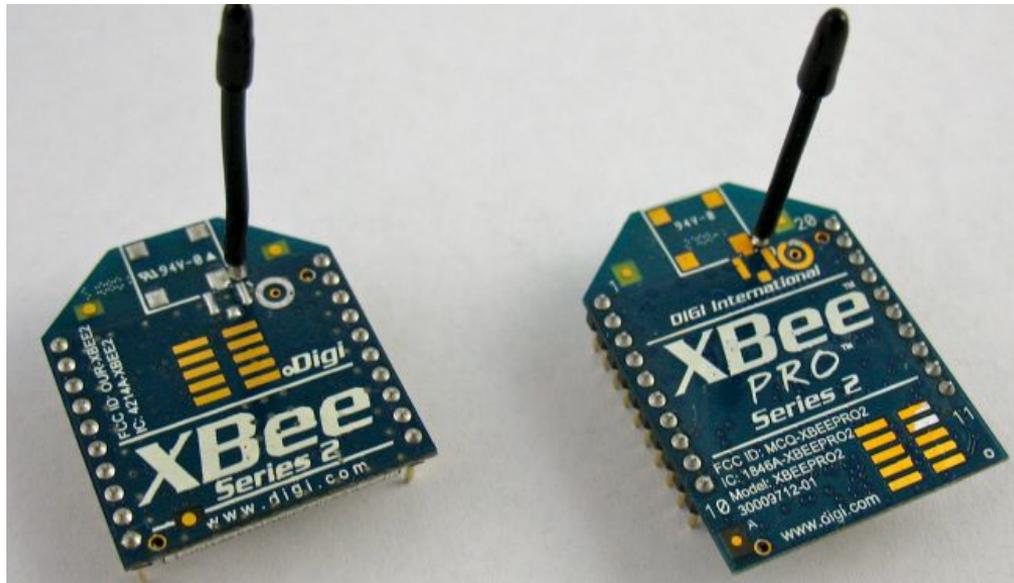
- Componentes de una red ZigBee:
 - Red: identificada mediante un PAN ID.
 - PAN Coordinador (FFD):
 - Elige un canal (dentro de una lista de canales habilitados y si la selección automática está habilitada).
 - Pone en marcha la red.
 - Acepta asociaciones de end-devices.
 - End devices (RFD)
 - Busca el coordinador de su red (con el PAN ID)
 - Pide asociación.
 - Router (FFD).
 - Misma funcionalidad que un End Device + Ruteo



Mediciones sobre 802.15.4

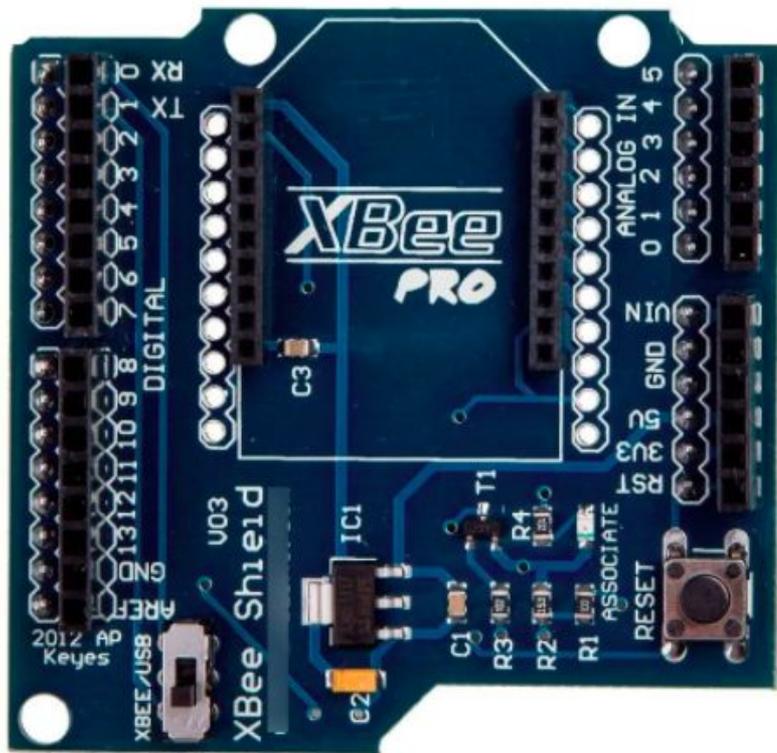
Escenario	Altura del nodo	Obstáculos	Distancia máxima
Referencia	1.5 m	Nada	170 m
1	0 m	Nada	40 m
2	0.15 m	Nada	76 m
3	1.5 m	Arboles	75 m
4	0 m	Arboles	40 m
5	1.5 m	Montículo	66 m
6	0 m	Montículo	40 m
7	1.5 m	Antena en posición incorrecta	86 m
8	0 m	Antena en posición incorrecta	36 m

Ejemplo: Digi XBee

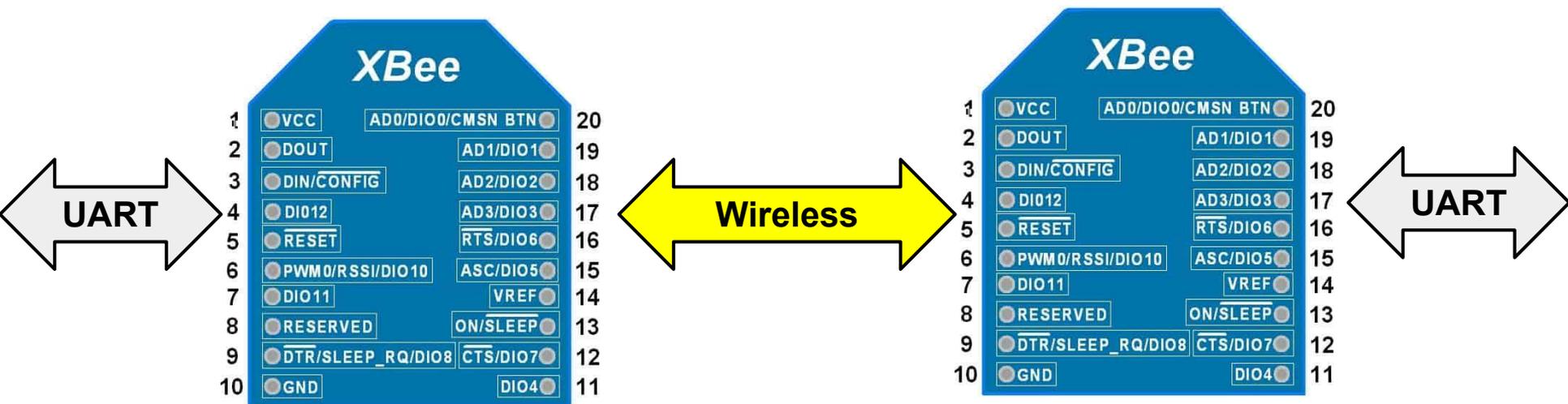




Ejemplo: XBee Shield



ZigBee PinOut Modo transparente



Debo configurar las direcciones MAC destino para que se comuniquen entre ellos

**ZigBee
Modo API**

Comando AT local (orden para un nodo local)

0x7E 1B	Longitud 2B	Tipo 0x08 1B	Frame ID 1B	Comando 2B	Parámetros Variable	Checksum 1B
7E	00 05	08	01	44 31	04	7D

Tipo: Comando local, comando remoto, respuesta a comando, reporte de datos, etc. Ejemplo: 08, comando AT local.

Longitud: sin contar los campos 7E, longitud y checksum

Comando: En código ASCII. Ejemplo: D1 (44 31), cambiar estado del pin D1

Parámetros: Depende del comando. Ejemplo: 04, Poner pin en estado bajo.

Trama ZigBee

Comando Remoto

SF 0x7E 1B	Longitud 2B	Tipo 0x17 1B	Frame ID 1B	Direcc. destino 8B	Direcc. destino 2B
--------------------------------	-----------------------	----------------------------------	-----------------------	------------------------------	------------------------------

Opciones 1B	Comando (ASCII) 2B	Parámetros Variable	Checksum 1B
-----------------------	----------------------------------	-------------------------------	-----------------------

7E 00 10 17 01 00 13 A2 00 40 7A BA D8 FF FE 02 44 34 05 6A
D4 High

Algunas opciones para el campo Tipo

- 0x08: Comando AT al dispositivo local (Conectado a la UART).
- 0x88: Respuesta a un comando AT local.
- 0x17: Comando AT al dispositivo remoto (Transmitido a través del medio inalámbrico).
- 0x92: Trama de muestreo de I/O
- 0x97: Respuesta a comando remoto.

Opciones (Cada bit configura una opción). Ejemplos.

- 0x01: Deshabilitar ACK (bit 0).
- 0x02: Aplicar cambios inmediatamente (bit 1).

Algunas opciones para el campo Comando y Parámetros

- D1, D2, D3, D4, D5, D6, D7, etc.: configurar el pin. Opciones:
 - 0x00: Desactivado.
 - 0x01: Función especial de cada pin (No todos los pines).
 - 0x02: ADC (No todos los pines)
 - 0x03: Digital Input
 - 0x04: Digital Output Low
 - 0x05: Digital Output High.
- IR: Tiempo entre muestras de los pines de entrada salida en milisegundos.
 - Valor entre 0x32 y 0xFFFF: Habilita el muestreo periódico de las I/O habilitadas. El valor indica el tiempo en milisegundos.
 - 0x00: Apaga el muestreo periódico.
- PL: Nivel de potencia (Valores entre 0x00 y 0x04).
- PM: Incrementa potencia de transmisión y sensibilidad.

Trama ZigBee
ZigBee Respuesta

SF 0x7E 1B	Longitud 2B	Tipo 0x97 1B	Frame ID 1B	Direcc. origen 8B	Direcc. origen 2B
--------------------------------	-----------------------	----------------------------------	-----------------------	-----------------------------	-----------------------------

Comando 1B	Status 1B	Checksum 1B
----------------------	---------------------	-----------------------

Status:
 0x00: OK
 0x01: Error
 0x02: Comando inválido
 0x03: Parámetro
 inválido

7E 00 0F 97 01 00 13 A2 00 40 3A 3B F8 FF FE 44 31 00 93

Trama ZigBee

Reporte periódico de datos

SF 0x7E 1B	Longitud 2B	Tipo 0x92 1B	Direcc. origen 8B	Direcc. origen 2B	Opciones 1B
--------------------------------	-----------------------	----------------------------------	-----------------------------	-----------------------------	-----------------------

N° muestras 1B	Digital Channel Mask 2B	Analog Channel Mask 1B	Muestras digitales 2B	Muestras analógicas 2B por muestra	Checksum 1B
--------------------------	---	--	-------------------------------------	---	-----------------------

7E 00 14 92 00 13 A2 00 40 3A 3B F8 2D 0E 41 01 00 10 08 00 10 03 6C F7

Trama ZigBee - Reporte periódico de datos

7E 00 14 92 00 13 A2 00 40 3A 3B F8 2D 0E 41 01 00 10 08 00 10 03 6C F7

Digital Channel Mask

- Pines digitales de los cuales se realiza un muestreo digital (mapa de bits en hex).
- Ejemplo: **00 10**: 0000 0000 0001 0000: Se muestrea el pin 4.

Muestras digitales

- Valores de los pines que se muestrean (mapa de bits en hexadecimal).
- Ejemplo: **00 10**: 0000 0000 0001 0000: El pin 4 está en alto.

Analog Channel Mask

- Pines digitales de los cuales se realiza un muestreo digital (mapa de bits en hex).
- Ejemplo: **08**: 0000 1000: Se muestrea el conversor AD 3.

Muestras Analógicas

- Valores analógicos de los pines muestreados. Indicados en secuencias de 2 bytes.
- Ejemplo: **03 6C**: Nivel 876.



7E 00 18 92 00 13 A2 00 40 3A 3B F8 FF FE 01 01 00 84 1C 00 80 02 33 02 31 03 33 4E

Length: 00 18 (24 bytes)

Frame type: 92 (Trama con muestreo de I/O)

Dirección origen de 64 bits: 00 13 A2 00 40 3A 3B F8

Dirección origen de 16bit: FF FE

Receive options: 01

Number of samples: 01

Digital channel mask: 00 84 (0000 0000 1000 0100: pines 7 y 2 habilitados)

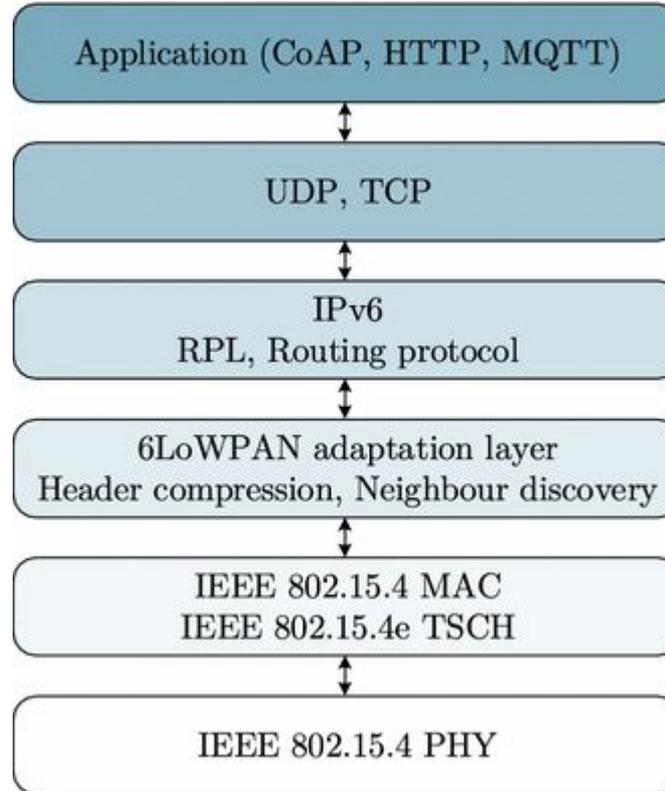
Analog channel mask: 1C (0001 1100: Conversores AD 4, 3 y 2 habilitados)

Valores digitales: 00 80 (0000 0000 1000 000: pin 7 en alto, pin 2 en bajo)

Valores analógicos: CAD 2: 02 33 (563 niveles), CAD 3: 02 31 (561 niveles), CAD 4:
03 33 (819 niveles)

Checksum: 4E

6LowPAN



Truncar IPv6 y
fragmentar tramas

LoRa

- LoRa: Long Range (Lora Alliance).
- Modulación: Tecnología patentada por Semtech.
 - Elevada selectividad: -148 dBm.
 - Salida: 20 dBm.
 - Modulación: Chirp spread spectrum.
 - 915 MHZ/868 MHZ/433 MHZ.
- Consumo:
 - TX: hasta 120 mA.
 - RX: hasta 12 mA.
- Tramas de hasta 256 bytes con CRC.
- Dispositivos:
 - End devices.
 - Gateways.



LoRa

- **Topologías:**
 - Estrella. Requiere Gateway
 - Punto a punto. End device + Gateway o dos End Devices.
- **Encriptación AES.**
- **Distancias según hoja de datos: 200 km.**
 - Según foros: 40 km con línea de visión.
 - 4 km sin línea de visión.

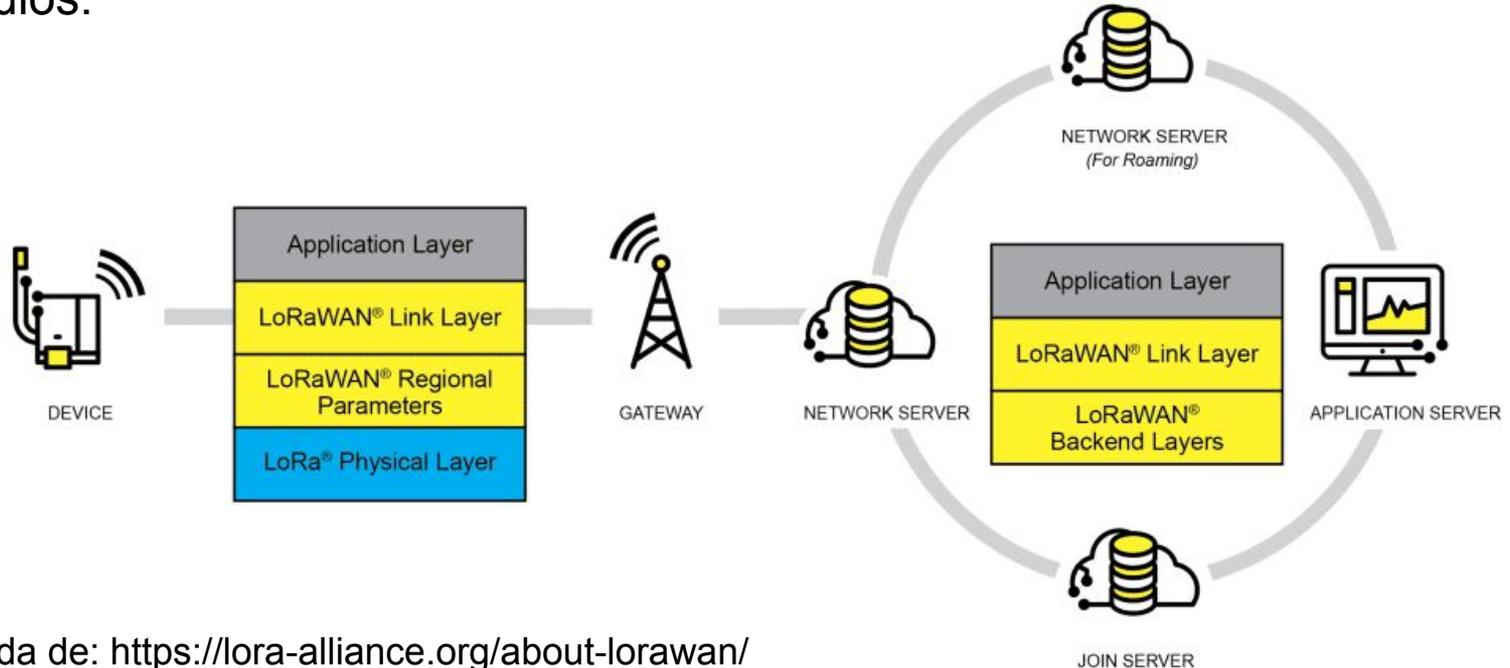


LoRa

	Europe	North America	China	Korea	Japan	India
Frequency band	867-869MHz	902-928MHz	470-510MHz	920-925MHz	920-925MHz	865-867MHz
Channels	10	64 + 8 +8	In definition by Technical Committee			
Channel BW Up	125/250kHz	125/500kHz				
Channel BW Dn	125kHz	500kHz				
TX Power Up	+14dBm	+20dBm typ (+30dBm allowed)				
TX Power Dn	+14dBm	+27dBm				
SF Up	7-12	7-10				
Data rate	250bps- 50kbps	980bps-21.9kpbs				
Link Budget Up	155dB	154dB				
Link Budget Dn	155dB	157dB				

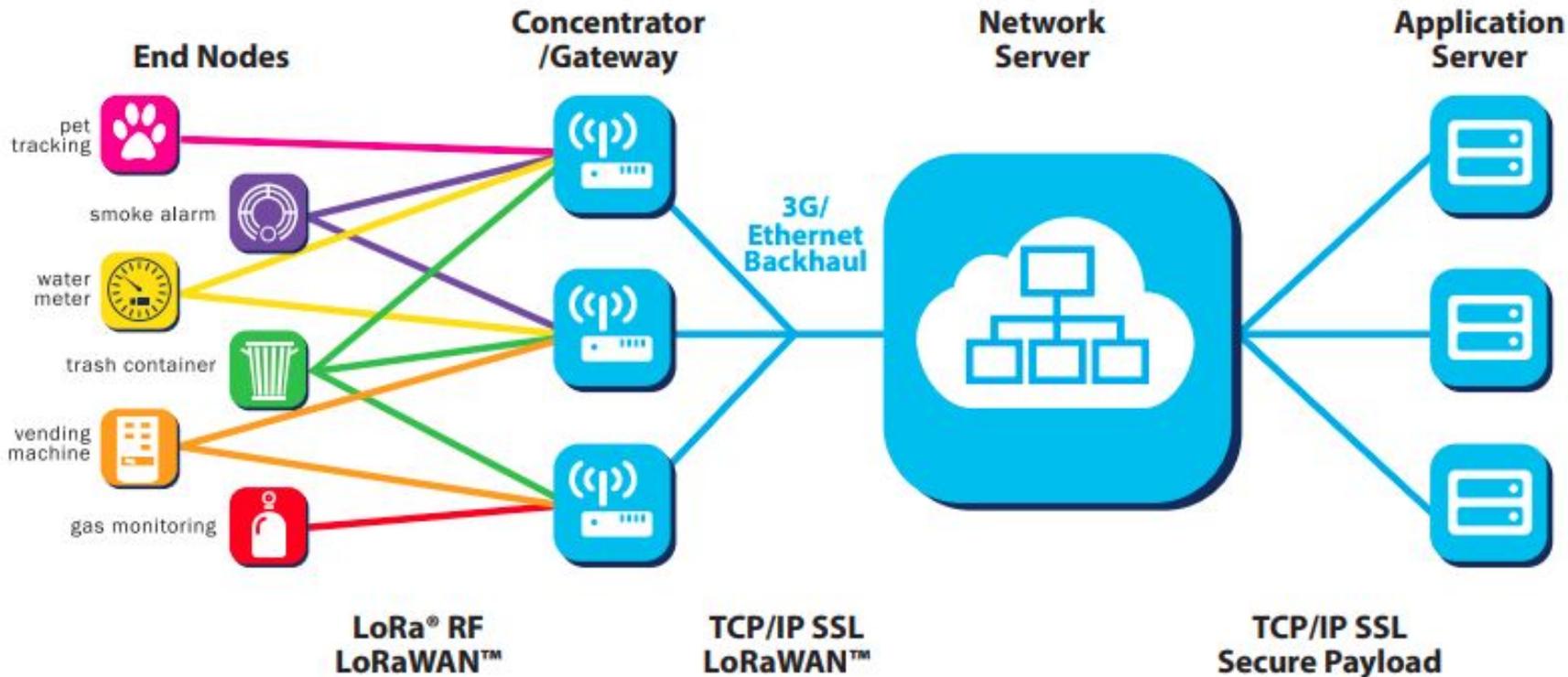
LoRa

- LoRaWAN: Arquitectura y plataforma de Low-Power, Wide-Area Networks (LPWAN): Radios LoRa + Gateway + plataforma tipo nube para manejar las radios.



LoRa

- LoRaWAN



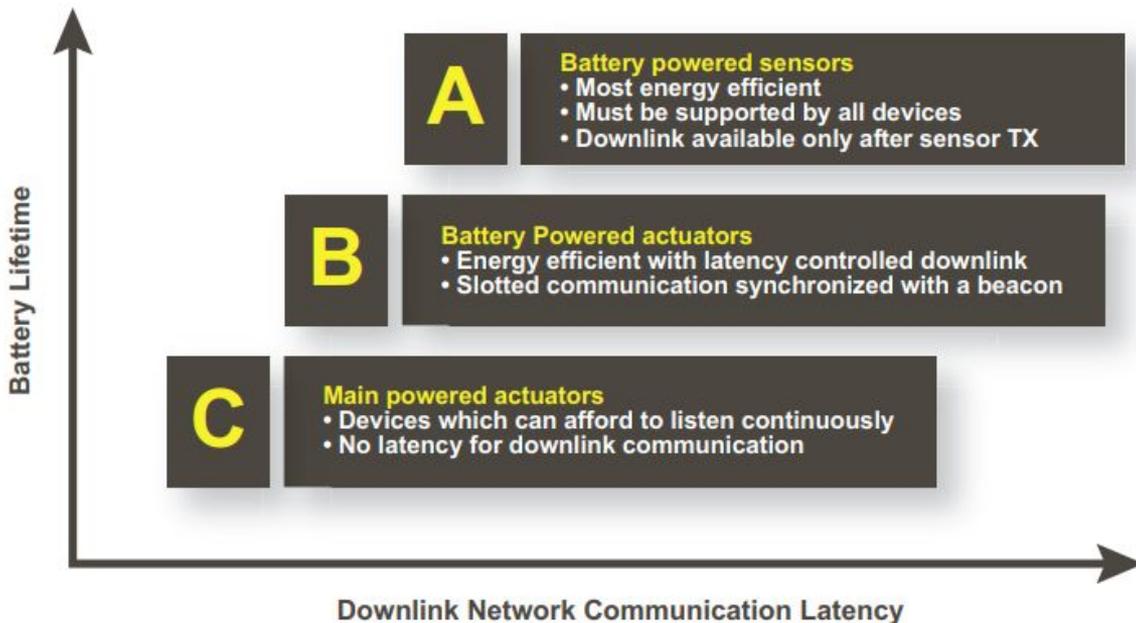


LoRaWAN

- El servidor lidia con paquetes repetidos (Varios gateways pueden recibir paquetes de un mismo nodo), envío de ACKs, etc.
- No es necesario handover.
- Red asíncrona. Los nodos envían datos cuando es necesario.
- Data rate adaptativo.
- Multicanal.

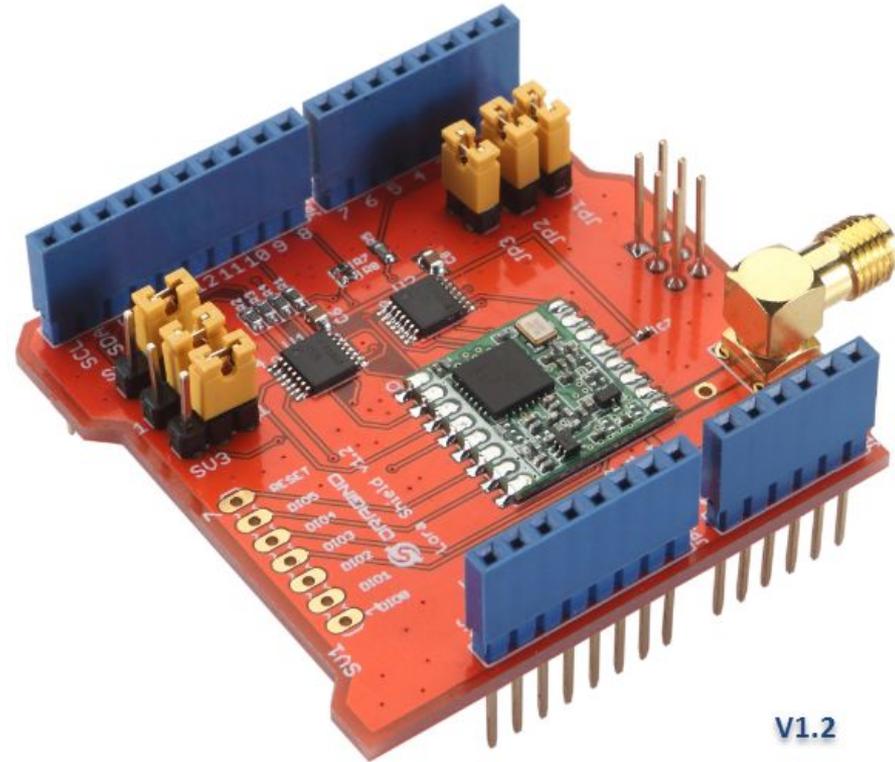
LoRaWAN

- Tipos de End Devices



LoRa

- Placas Lora Shield: Placas preparadas para trabajar con Arduino.
- Gran cantidad de librerías y ejemplos disponibles en Internet.
- Comunicación SPI.



V1.2

Pines azules: sin uso

LoRa

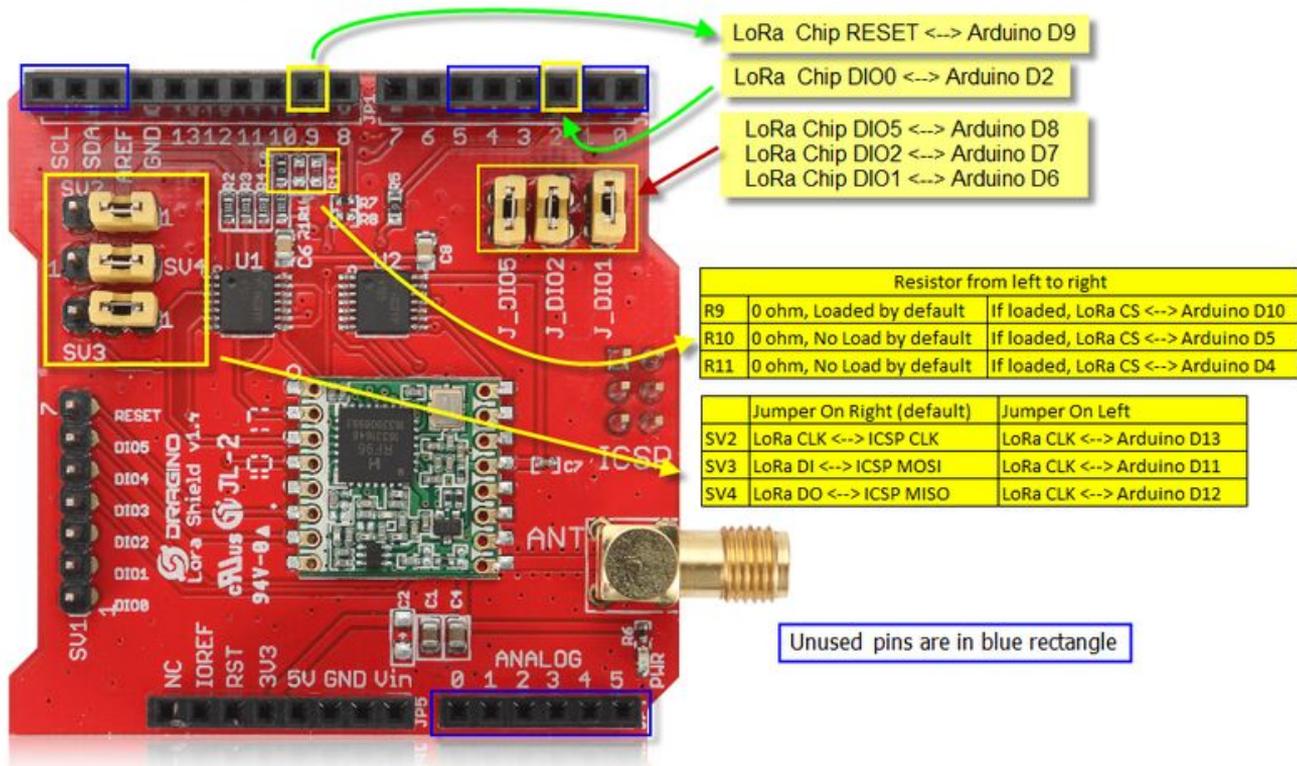


Figura obtenida de: <https://wiki.dragino.com/>

RFID y NFC

RFID (Radio-frequency identification) y NFC (Near-Field Communication)

- Permiten identificar y seguir dispositivos llamados “etiquetas” que se adjuntan a objetos. Las etiquetas son leídas por “lectores”.
- Las etiquetas pueden:
 - Ser solo un dispositivo de identificación (memoria ROM).
 - Poseer memoria que puede ser escrita y leída.
- Etiquetas pueden ser:
 - Activas: Poseen fuente de alimentación.
 - Pasivas: No poseen fuente de alimentación. Obtienen la energía de las señales electromagnéticas emitidas por el lector.
- RFID: distancias entre 1 m y 200 m según frecuencia.
- NFC: distancias máximas de 10 cm, típicas de 4 cm.

- Nace en el MIT 1999.
 - EPCglobal (EPC: Electronic Product Code)
- Objetivo: Reemplazar el código de barra (lectura a 10 m).
- El lector siempre transmite.
 - Si la etiqueta es pasiva, obtiene información de la señal transmitida por el lector.
- Cuando la etiqueta:
 - Recibe datos: Utiliza ACK.
 - Transmite datos:
 - El lector transmite una portadora sin modular (sin datos).
 - La etiqueta transmite sus datos.
 - Retrodispersión.
 - Inducción.

Frecuencias de RFID

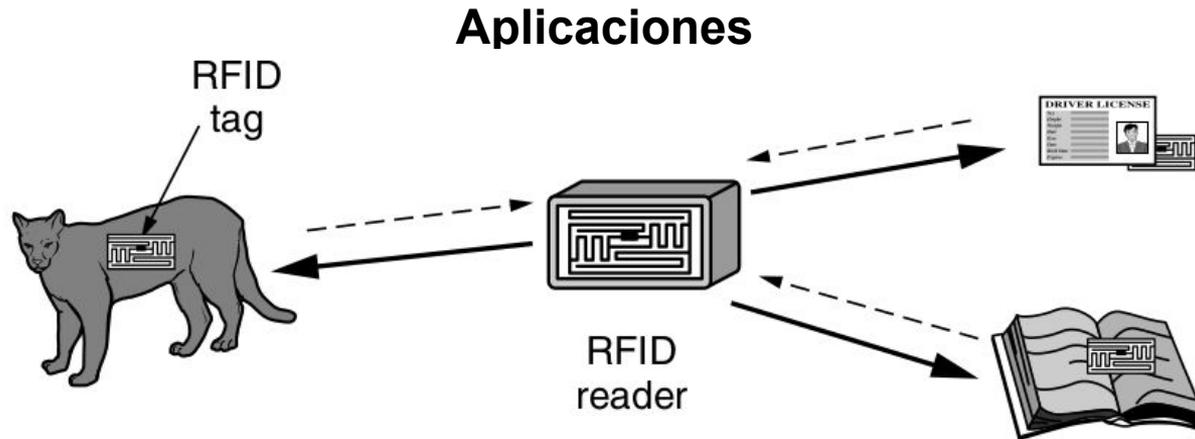
Band	Range	Data Speed	Tags
Low frequency (LF): 125–134.2 kHz	10 m	low	passive
High frequency (HF): 13.56 MHz	10 cm–1 m	low to moderate	passive
Ultra high frequency (UHF): 433 MHz	1–100 m	moderate	passive or active
Ultra high frequency (UHF): 856 MHz–960 MHz	1–12 m	moderate to high	passive or active
Microwave: 2.45–5.8 GHz	1–2 m	high	active
Microwave: 3.1–10 GHz	<200 m	high	active

Frecuencia
NFC



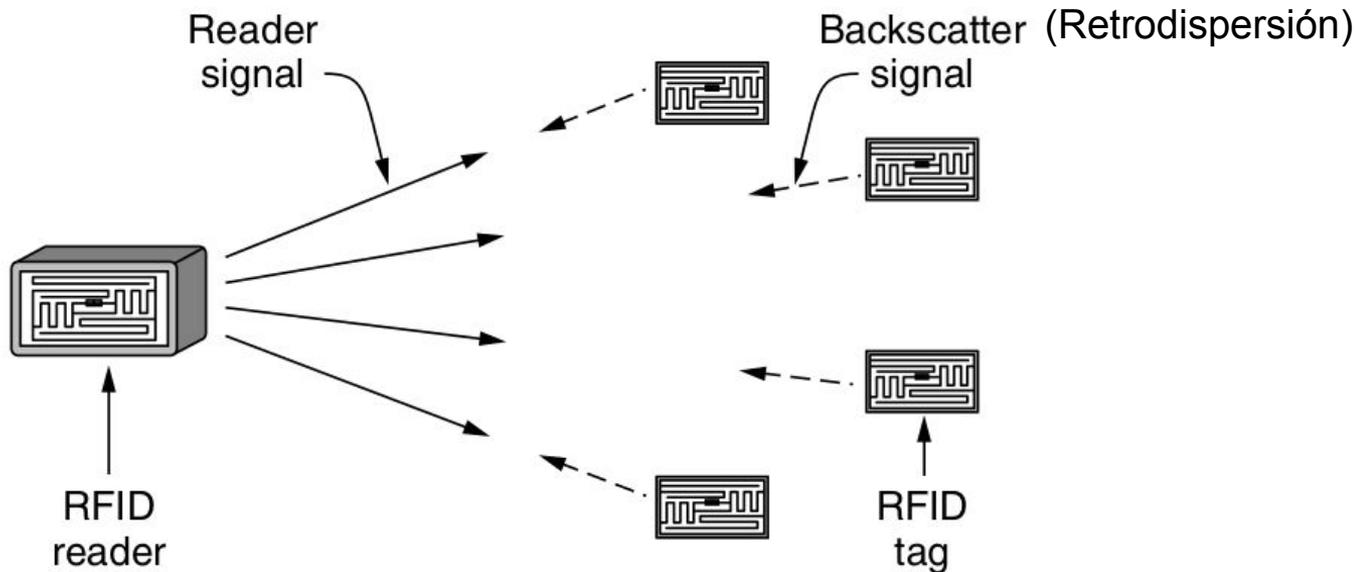
RFID formas de trabajar

- Retrodispersión: Las etiquetas reflejan la señal recibida pudiendo producir cambios en la señal reflejada.
- Inducción: La etiqueta obtiene energía eléctrica de la señal y trabaja como una computadora en si misma.



RFID

- Arquitectura EPC Gen 2:
 - 9600 bps, 115,2 kbps
 - Identificador único de 96 bits.
 - Memoria que puede ser escrita y leída.





RFID Acceso al medio

- El lector no sabe que etiquetas están presentes: Pide a las etiquetas que se reporten: pueden colisionar.
- Acceso Múltiple: Aloha ranurado.
- Mensaje Query: indica el comienzo del proceso.
- Mensajes QRepeat: Marcan el comienzo de cada ranura (beacon).
- Las etiquetas eligen aleatoriamente una ranura y transmiten un código de 16 bits (RN16).
- Si no hubo colisión, el lector responde (ACK), y la etiqueta transmite su ID completo (96 bits).

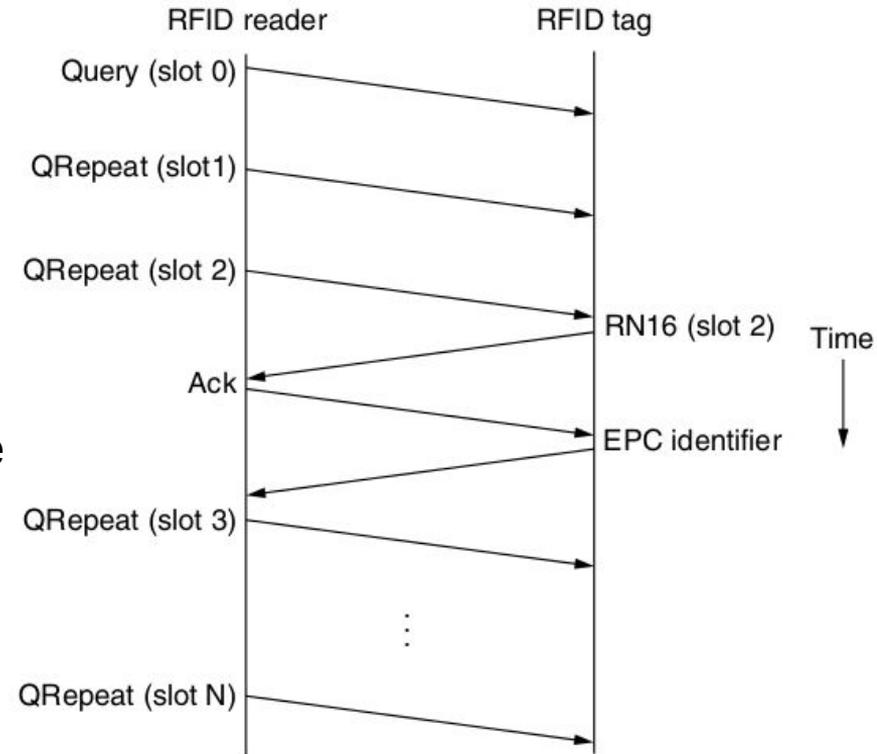


Figura obtenida de: A. Tanenbaum, D. Wetherall, "Redes de computadoras", Editorial Pearson, Quinta edición (2012), pag. 330



UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO



**FACULTAD
DE INGENIERÍA**

**Licenciatura en Ciencias de la
Computación**



Etiquetas RFID



Lector Arduino y Etiquetas



Bloqueadores RFID

NFC

- Norma ISO 14443-3:2011
- 106 - 848 Kbit/s
- Modulación:
 - Lector: ASK (modulación en amplitud)
 - BPSK (modulación en fase)
- Distancia máxima: 10 cm según estándar. Usualmente 4 cm.
- Memoria: 96 bytes a 1 MB según tipo.
- Topología punto a punto.



NFC - NDEF (NFC Data Exchange Format)

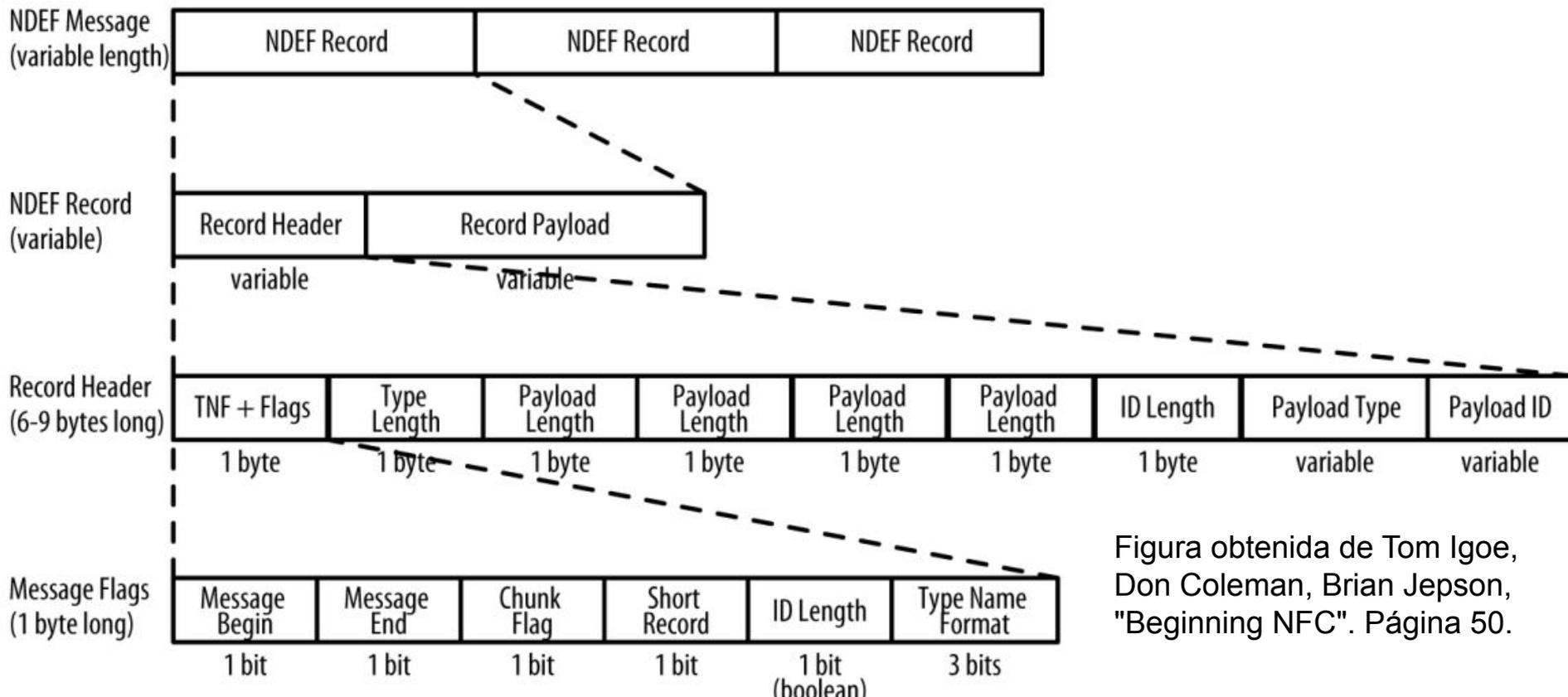


Figura obtenida de Tom Igoe, Don Coleman, Brian Jepson, "Beginning NFC". Página 50.

NFC - NDEF (NFC Data Exchange Format)

- TNF: Type Name Format. Indica el tipo del campo Payload Type. Algunos ejemplos:
 - Formatos definidos por el estándar NFC.
 - Formatos MIME (Internet media type).
 - URI.
 - Reservados para uso futuro
- Payload Type: Tipo de datos en el payload.



UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO

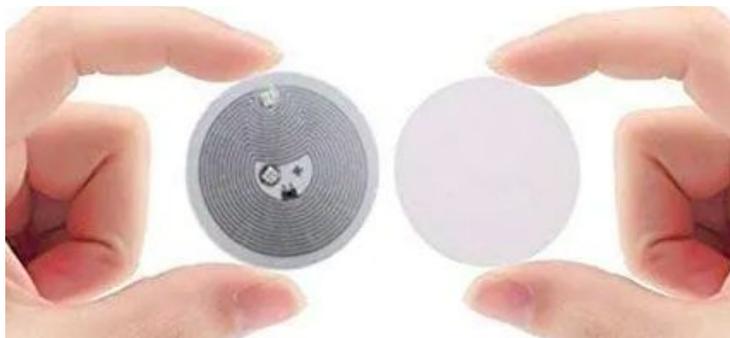


**FACULTAD
DE INGENIERÍA**

**Licenciatura en Ciencias de la
Computación**



NFC





UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO



**FACULTAD
DE INGENIERÍA**

**Licenciatura en Ciencias de la
Computación**

NFC



Bibliografía:

- Atmel, “8-bit AVR Microcontrollers. ATmega328-328P Datasheet complete”. 2016.
- Manuel Jiménez, Rogelio Palomera, Isidoro Couvertier, “Introduction to Embedded Systems Using Microcontrollers and the MSP430”. Springer. 2014.
- Peter Marwedel, "Embedded System Design. Embedded Systems, Foundations of Cyber-Physical Systems, and the Internet of Things". Springer. 2018.
- David Russel, "Introduction to Embedded System Using ANSI C and the Arduino Development Environment". Morgan y ClayPool Publishers series. 2010.
- John Davies, "MSP430 Microcontroller Basics". Elsevier. 2008.
- John Catsoulis. "Designing Embedded Hardware". O'Reilly. 2009.
- Perry Xiao , "Designing Embedded Systems and the Internet of Things (IoT) with the ARM Mbed". Wiley. 2018.



UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO



**FACULTAD
DE INGENIERÍA**

**Licenciatura en Ciencias de la
Computación**

Bibliografía:

- Chaves Osorio, José & Quintero, Edwin & Cortes, Jimy. (2011). Generación de señales senoidales mediante PWM y filtros activos de segundo orden.
- Tanenbaum, Wetherall. "Computer Networks". 5° edición .Prentice Hall. 2011.
- Tom Igoe, Don Coleman, Brian Jepson, "Beginning NFC". O'Reilly. 2014
- LoRa Alliance. "A technical overview of LoRa and LoRaWAN". 2015.