UNIVERSIDAD RURAL DE GUATEMALA

FACULTAD DE INGENIERÍA
Ingeniería en Sistemas con especialización en IA y Ciberseguridad

Terminar la práctica 3, entender cómo viaja un byte por un cable y agregar el Arduino • Alternativa con PIC16F1829 en C

El Arduino recibe la letra

Electrónica Digital • FE731 • Sesión 4 de 4

Ing. Mynor A. Cifuentes

Universidad Rural de Guatemala • Laboratorio presencial • 8 de octubre de 2026

Agenda de la tarde

Última clase: el circuito de ayer, ahora con un microcontrolador

APERTURA

30 min	Cómo funciona el circuito de ayer

EL ARDUINO RECIBE LA LETRA

55 min	**A.** Terminar la práctica 3
15 min	Receso
35 min	**B.** Conectar el Arduino y la LCD
30 min	**C.** El Arduino recibe la letra
25 min	**D.** Reto: el Arduino es el emisor

CIERRE

25 min	Evaluación en Socrative
25 min	Desarmado y cierre del curso

El circuito de la práctica 3 no se desarma. Solo se le agregan cables: el Arduino escucha el mismo cable por el que viaja la letra y la escribe en la pantalla.

NOTA

Sin Arduino: termine la parte A y siga con los retos de la práctica 3. **Con PIC16F1829:** siga su guía; la explicación está al final de esta presentación.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Cuántas salidas activa a la vez el **74LS138**?	
2	¿Qué elige la dirección CBA del **74LS151**?	
3	¿Qué diferencia hay entre Y y W del **74LS151**?	
4	¿Qué es la U en ASCII?	
5	¿Por qué con 100 nF se veía la letra completa?	

LO QUE SIGUE

Ayer un byte viajó bit por bit por un cable y ocho LEDs lo mostraron.

Hoy un **microcontrolador** escucha ese mismo cable, arma la letra y la escribe en una pantalla.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Cuántas salidas activa a la vez el **74LS138**?	Una sola, en bajo
2	¿Qué elige la dirección CBA del **74LS151**?	
3	¿Qué diferencia hay entre Y y W del **74LS151**?	
4	¿Qué es la U en ASCII?	
5	¿Por qué con 100 nF se veía la letra completa?	

LO QUE SIGUE

Ayer un byte viajó bit por bit por un cable y ocho LEDs lo mostraron.

Hoy un **microcontrolador** escucha ese mismo cable, arma la letra y la escribe en una pantalla.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Cuántas salidas activa a la vez el **74LS138**?	Una sola, en bajo
2	¿Qué elige la dirección CBA del **74LS151**?	Cuál de los 8 bits sale
3	¿Qué diferencia hay entre Y y W del **74LS151**?	
4	¿Qué es la U en ASCII?	
5	¿Por qué con 100 nF se veía la letra completa?	

LO QUE SIGUE

Ayer un byte viajó bit por bit por un cable y ocho LEDs lo mostraron.

Hoy un **microcontrolador** escucha ese mismo cable, arma la letra y la escribe en una pantalla.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Cuántas salidas activa a la vez el **74LS138**?	Una sola, en bajo
2	¿Qué elige la dirección CBA del **74LS151**?	Cuál de los 8 bits sale
3	¿Qué diferencia hay entre Y y W del **74LS151**?	W es Y negado
4	¿Qué es la U en ASCII?	
5	¿Por qué con 100 nF se veía la letra completa?	

LO QUE SIGUE

Ayer un byte viajó bit por bit por un cable y ocho LEDs lo mostraron.

Hoy un **microcontrolador** escucha ese mismo cable, arma la letra y la escribe en una pantalla.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Cuántas salidas activa a la vez el **74LS138**?	Una sola, en bajo
2	¿Qué elige la dirección CBA del **74LS151**?	Cuál de los 8 bits sale
3	¿Qué diferencia hay entre Y y W del **74LS151**?	W es Y negado
4	¿Qué es la U en ASCII?	$0101\ \ 0101$ (55h)
5	¿Por qué con 100 nF se veía la letra completa?	

LO QUE SIGUE

Ayer un byte viajó bit por bit por un cable y ocho LEDs lo mostraron.

Hoy un **microcontrolador** escucha ese mismo cable, arma la letra y la escribe en una pantalla.

¿Lo recuerdan?

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Cuántas salidas activa a la vez el **74LS138**?	Una sola, en bajo
2	¿Qué elige la dirección CBA del **74LS151**?	Cuál de los 8 bits sale
3	¿Qué diferencia hay entre Y y W del **74LS151**?	W es Y negado
4	¿Qué es la U en ASCII?	$0101\ \ 0101$ (55h)
5	¿Por qué con 100 nF se veía la letra completa?	Los turnos eran más rápidos que el ojo

LO QUE SIGUE

Ayer un byte viajó bit por bit por un cable y ocho LEDs lo mostraron.

Hoy un **microcontrolador** escucha ese mismo cable, arma la letra y la escribe en una pantalla.

01 PARTE

ELECTRÓNICA DIGITAL

Cómo funciona el circuito de ayer

Cuatro bloques, cuatro trabajos

Como pasar lista en un salón

Bloque	En el salón	En el circuito
555	El reloj de la pared	Un pulso cada cierto tiempo: marca el paso
74LS193	El profesor que dice el número	Suma 1 en cada pulso; $Q_C Q_B Q_A$ (0 a 7) es el **turno**
74LS151	El alumno que contesta	Pone el bit del turno (D_0 a D_7) en **un solo cable**
74LS138	Quien anota en la pizarra	Enciende el LED del turno, **solo si** el bit es 1

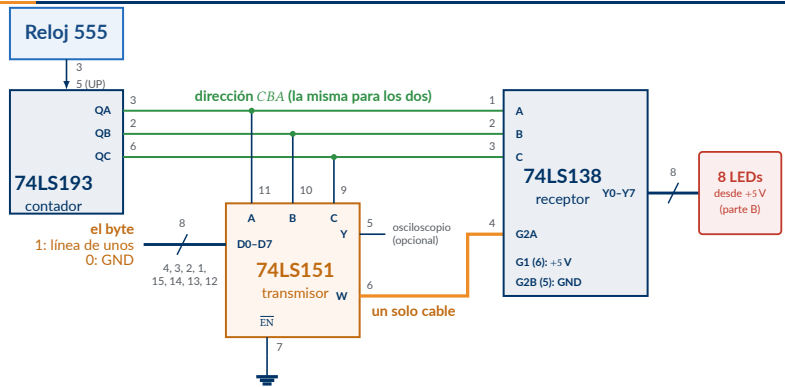
El secreto: emisor y receptor reciben **el mismo turno**. Los tres cables Q_A, Q_B y Q_C del contador van a los dos. Cuando el 74LS151 manda el bit 5, el 74LS138 sabe que es el bit 5.

NOTA

Ocho cables entran al 74LS151 y **uno** sale: paralelo a serie. El 74LS138 lo regresa a ocho LEDs: serie a paralelo.

El sistema completo

La dirección (verde) va a los dos integrados; el bit viaja por el cable naranja



Por qué el cable sale de W y llega a G2A

G2A es activa en bajo; W es el bit negado

Bit D_k	W = G2A	En el turno k
1	0	El 74LS138 trabaja: se enciende el LED k
0	1	El 74LS138 se apaga: ningún LED

⇄ «G2A PASA A W DEL 151»

Quite el cable de G2A (terminal 4 del 74LS138) que iba a GND y únalo, con un solo jumper, a W (terminal 6 del 74LS151).

La otra salida, Y (terminal 5), es el bit **sin negar**. Ayer quedó libre.

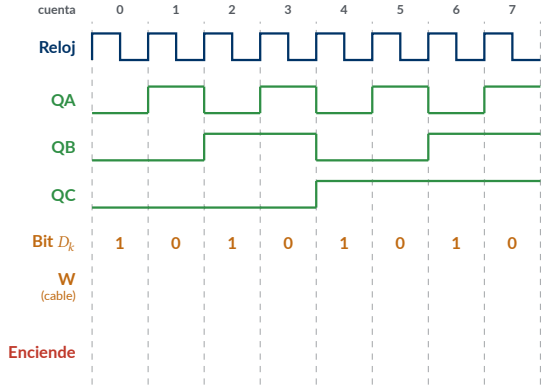
Hoy el Arduino la usa.

⚠ ADVERTENCIA

Si G2A va a Y y no a W, los LEDs muestran la letra negada: se encienden los turnos con bit 0.

La letra U, turno por turno

0101 0101: en cada turno sale un bit

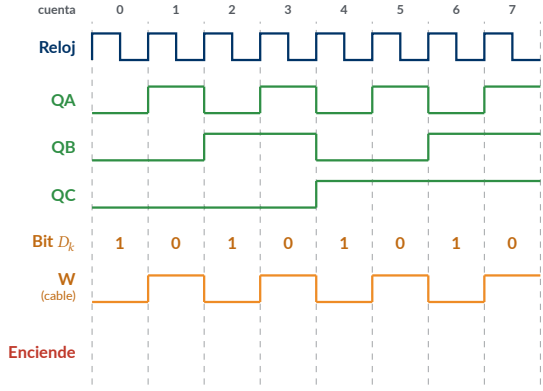


LÉALO ASÍ

1. El reloj sube y la cuenta avanza.
2. La cuenta elige el bit D_k.

La letra U, turno por turno

0101 0101: en cada turno sale un bit

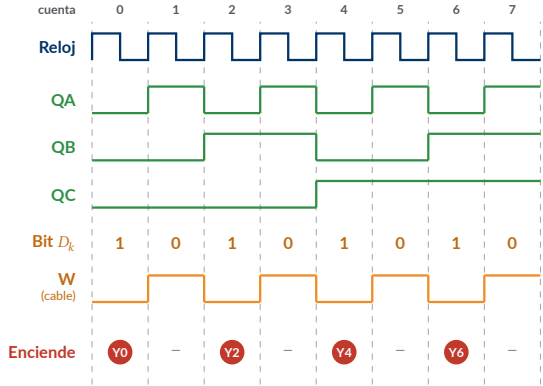


LÉALO ASÍ

1. El reloj sube y la cuenta avanza.
2. La cuenta elige el bit D_k.
3. W es ese bit negado.

La letra U, turno por turno

0101 0101: en cada turno sale un bit



LÉALO ASÍ

1. El reloj sube y la cuenta avanza.
2. La cuenta elige el bit D_k.
3. W es ese bit negado.
4. Si W vale 0, se enciende el LED del turno.

Se encienden Y0, Y2, Y4 y Y6, **uno a la vez**.

Por qué con 100 nF se ven todos a la vez

El mismo circuito, tres velocidades

C	f	Por turno	Lo que se ve
100 µF	0.69 Hz	1.46 s	La luz salta de un LED a otro
10 µF	6.9 Hz	0.15 s	Parpadeo rápido
100 nF	690 Hz	1.5 ms	La letra completa, fija

COMO EN EL CINE

Con 100 nF cada LED se enciende **86 veces por segundo**. El ojo no alcanza a ver que se apaga.

NOTA

$f = \dfrac{1.44}{(R_A + 2R_B)\,C}$ con $R_A = 1\,\text{k}\Omega$ y $R_B = 10\,\text{k}\Omega$.

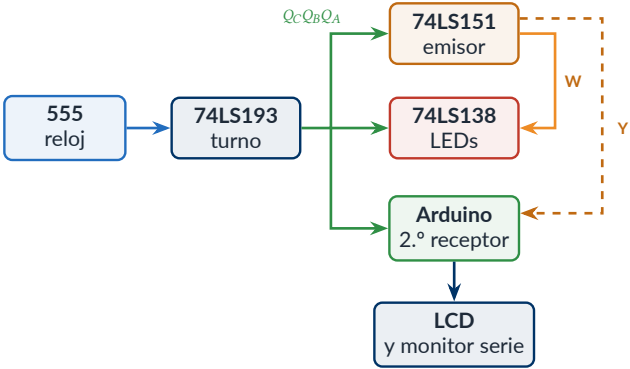
02 PARTE

ELECTRÓNICA DIGITAL

El Arduino recibe la letra

Dónde entra el Arduino

Un segundo receptor que escucha el mismo cable



OBJETIVOS

- Terminar y probar la práctica 3.
- Explicar cómo viaja un byte por un cable.
- Leer con el Arduino el turno y el bit, y armar la letra.
- Mostrarla en la LCD (I2C) y en la computadora (serie).
- Usar el Arduino como emisor.

NOTA

El Arduino **solo lee**. Si se desconecta, el circuito de ayer sigue igual.

Materiales y seguridad

Dos fuentes de 5 V y una sola tierra

MATERIALES Y EQUIPO

- El circuito de la práctica 3: 555, **74LS193**, **74LS151**, **74LS138** y ocho LEDs.
- Capacitores de 100 µF, 10 µF y 100 nF.
- **Arduino Nano o Uno** con cable USB y el Arduino IDE con la biblioteca `LiquidCrystal I2C`.
- **LCD 16x2** con módulo I2C y cuatro cables macho-hembra.
- Seis cables macho-macho, **MO-MB102** y multímetro.

NORMAS

- Apague el **MO-MB102** y desconecte el USB antes de mover cables.
- `GND` del Arduino al riel **`GND`**: sin tierra común no entiende los niveles.
- El `5V` del Arduino va **solo** a la LCD, nunca a los rieles.
- Una salida nunca va a otra salida: en el reto, quite el cable de `W` antes de poner el de `D9`.

Parte A. Terminar y probar la práctica 3

55 min Por bloques, en orden: no pase al siguiente si el anterior falla

	Bloque	Cómo se prueba
1	Alimentación	5 V entre 16 y 8 de cada 74LS; entre 8 y 1 del 555
2	Reloj (100 µF)	El LED del 555 destella cada 1.5 s
3	Contador y receptor	Con `G2A` a `GND`, la luz recorre Y0 a Y7
4	Emisor	**74LS151**: 16 a +5 V; 8 y 7 a `GND`; 11, 10, 9 a Q_A, Q_B, Q_C
5	La letra	Puentes de la U: `0101 0101`
6	El cable del dato	`G2A` (4 del 138) a `W` (6 del 151)

RECOMENDACIÓN

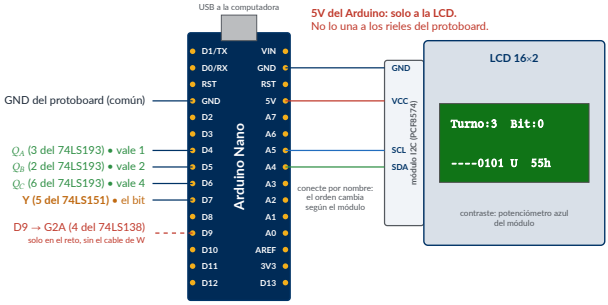
Punto de control A: con 100 nF se ven fijos Y6, Y4, Y2 y Y0. Con la R (`0101 0010`): Y6, Y4 y Y1.

NOTA

La guía *Cómo van conectados los circuitos* de la sesión 3 tiene cada cable dibujado.

Parte B. Conectar el Arduino y la LCD

35 min El cable va a un hueco libre de la misma columna del terminal



Arduino	Va a
GND	Riel **GND** del protoboard
D4	3 del **74LS193** (Q_A)
D5	2 del **74LS193** (Q_B)
D6	6 del **74LS193** (Q_C)
D7	**5 del 74LS151 (Y)**
5V, GND	**VCC** y **GND** de la LCD
A4, A5	**SDA** y **SCL** de la LCD

El Uno tiene los mismos nombres de terminales.

Parte B. Cargar el programa

`S04_Receptor_Letra.ino`, en la carpeta Arduino del curso

PASOS

1. Abra `S04_Receptor_Letra.ino` en el Arduino IDE.
2. Si falta, instale `LiquidCrystal I2C` (Frank de Brabander).
3. Placa: `Arduino Nano` con `ATmega328P (Old Bootloader)`, o `Arduino Uno`. Elija el puerto.
4. Pulse **Subir**.
5. Encienda el **MO-MB102** y abra el **Monitor serie** a **9600** baudios.

ADVERTENCIA

- ¿Falla al subir el Nano? Pruebe sin *Old Bootloader*.
- ¿No aparece el puerto? Controlador `CH340` u otro cable USB.
- ¿LCD sin letras? Gire el potenciómetro azul.
- ¿Solo cuadros? Cambie `0x27` por `0x3F`.

Qué hace el programa

Lee el turno, guarda el bit en su lugar y, con los ocho, tiene la letra

```
int leerTurno() {
  return digitalRead(QA) * 1 + digitalRead(QB)
       * 2
       + digitalRead(QC) * 4;
}

// en loop(), cuando cambia el turno:
int bit = digitalRead(BIT_Y);
bitWrite(recibido, turno, bit);  // en su lugar
bitSet(llegaron, turno);         // ya llego

if (turno == 7 && llegaron == 0xFF) {
  letra = recibido;              // los 8 bits
}
```

LA MISMA SUMA DE PESOS

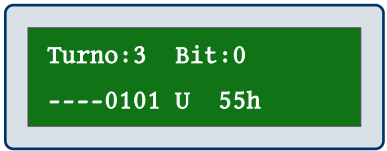
El turno es $4\,Q_C + 2\,Q_B + Q_A$, igual que en el **74LS138**. El bit del turno 0 es D_0, el del turno 7 es D_7.

NOTA

Si se salta un turno (un cable flojo), la letra empieza de nuevo: el Arduino solo acepta los ocho bits **seguidos**.

Parte C. Reloj lento: 100 µF

30 min La LCD dice el turno y va llenando la letra de derecha a izquierda



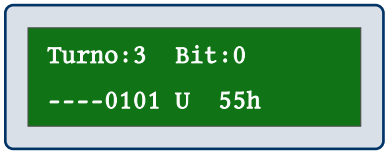
NOTA

- es un bit que todavía no llega. Al final de la línea 2 está la última letra completa y su código.

Turno	$Q_C Q_B Q_A$	Bit	Línea 2
0	000		
1	001		
2	010		
3	011		
7	111		

Parte C. Reloj lento: 100 µF

30 min La LCD dice el turno y va llenando la letra de derecha a izquierda



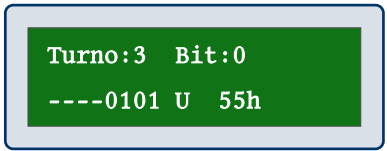
NOTA

- es un bit que todavía no llega. Al final de la línea 2 está la última letra completa y su código.

Turno	$Q_C Q_B Q_A$	Bit	Línea 2
0	000	1	-------1
1	001		
2	010		
3	011		
7	111		

Parte C. Reloj lento: 100 µF

30 min La LCD dice el turno y va llenando la letra de derecha a izquierda



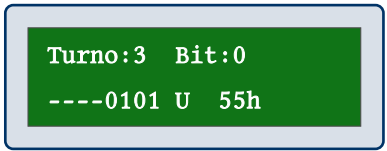
NOTA

- es un bit que todavía no llega. Al final de la línea 2 está la última letra completa y su código.

Turno	$Q_C Q_B Q_A$	Bit	Línea 2
0	000	1	-------1
1	001	0	------01
2	010		
3	011		
7	111		

Parte C. Reloj lento: 100 µF

30 min La LCD dice el turno y va llenando la letra de derecha a izquierda



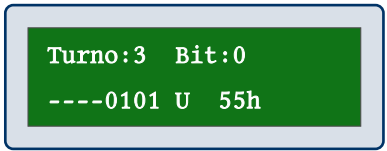
NOTA

- es un bit que todavía no llega. Al final de la línea 2 está la última letra completa y su código.

Turno	$Q_C Q_B Q_A$	Bit	Línea 2
0	000	1	-------1
1	001	0	------01
2	010	1	-----101
3	011		
7	111		

Parte C. Reloj lento: 100 µF

30 min La LCD dice el turno y va llenando la letra de derecha a izquierda



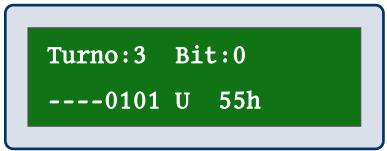
NOTA

- es un bit que todavía no llega. Al final de la línea 2 está la última letra completa y su código.

Turno	$Q_C Q_B Q_A$	Bit	Línea 2
0	000	1	-------1
1	001	0	------01
2	010	1	-----101
3	011	0	----0101
7	111		

Parte C. Reloj lento: 100 µF

30 min La LCD dice el turno y va llenando la letra de derecha a izquierda



NOTA

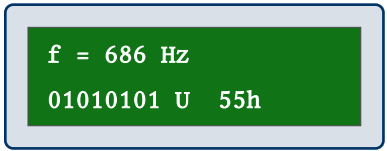
- es un bit que todavía no llega. Al final de la línea 2 está la última letra completa y su código.

Turno	$Q_C Q_B Q_A$	Bit	Línea 2
0	000	1	`-------1`
1	001	0	`------01`
2	010	1	`-----101`
3	011	0	`----0101`
7	111	0	`01010101 U 55h`

Una letra completa tarda unos **12 s**: 8×1.46 s.

Parte C. Reloj medio y rápido

Con 100 nF la LCD muestra la frecuencia del 555 y la última letra completa



C	Letras por segundo
10 µF	
100 nF	

PRUEBE CON 100 nF

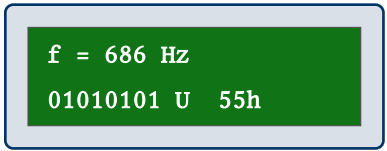
1. Cambie los puentes a la R (0101 0010), la A (0100 0001) y la L (0100 1100).
2. La LCD, el monitor serie y los LEDs deben mostrar la misma letra.
3. Cargue su **inicial**: la A es 41h, la B 42h, y así sucesivamente.

ADVERTENCIA

¿Sale 1010 1010 y ? en lugar de la U? D7 está en W (6): va en Y (5).

Parte C. Reloj medio y rápido

Con 100 nF la LCD muestra la frecuencia del 555 y la última letra completa



C	Letras por segundo
10 µF	Casi 1: $6.9 \div 8$
100 nF	Unas 86: $690 \div 8$

PRUEBE CON 100 nF

1. Cambie los puentes a la R (`0101 0010`), la A (`0100 0001`) y la L (`0100 1100`).
2. La LCD, el monitor serie y los LEDs deben mostrar la misma letra.
3. Cargue su **inicial**: la A es 41h, la B 42h, y así sucesivamente.

ADVERTENCIA

¿Sale `1010 1010` y `?` en lugar de la U? `D7` está en `W` (6): va en `Y` (5).

Parte D. Reto: el Arduino es el emisor

25 min Usted escribe una letra en la computadora y aparece en los LEDs

⇄ UN SOLO CABLE CAMBIA

1. Apague. **Quite** el cable de `W` (6 del 151) a `G2A` (4 del 138).
2. Ponga un cable de `D9` a `G2A` (4 del 138).
3. Con 100 nF, encienda: los LEDs muestran la U.
4. En el monitor serie escriba una letra y pulse Enter.

RIESGO ELÉCTRICO

Nunca deje `D9` y `W` unidos al mismo tiempo: dos salidas peleando son un cortocircuito.

Escribe	D7 … D0	LEDs
A		
K		

? PIÉNSELO

La LCD sigue mostrando la letra de los puentes. ¿Por qué?

Parte D. Reto: el Arduino es el emisor

25 min Usted escribe una letra en la computadora y aparece en los LEDs

UN SOLO CABLE CAMBIA

1. Apague. **Quite** el cable de `W` (6 del 151) a `G2A` (4 del 138).
2. Ponga un cable de `D9` a `G2A` (4 del 138).
3. Con 100 nF, encienda: los LEDs muestran la U.
4. En el monitor serie escriba una letra y pulse Enter.

RIESGO ELÉCTRICO

Nunca deje `D9` y `W` unidos al mismo tiempo: dos salidas peleando son un cortocircuito.

Escribe	D7 … D0	LEDs
A	0100 0001	Y6, Y0
K	0100 1011	Y6, Y3, Y1, Y0

PIÉNSELO

La LCD sigue mostrando la letra de los puentes. ¿Por qué?

Parte D. Reto: el Arduino es el emisor

25 min Usted escribe una letra en la computadora y aparece en los LEDs

UN SOLO CABLE CAMBIA

1. Apague. **Quite** el cable de `W` (6 del 151) a `G2A` (4 del 138).
2. Ponga un cable de `D9` a `G2A` (4 del 138).
3. Con 100 nF, encienda: los LEDs muestran la U.
4. En el monitor serie escriba una letra y pulse Enter.

RIESGO ELÉCTRICO

Nunca deje `D9` y `W` unidos al mismo tiempo: dos salidas peleando son un cortocircuito.

Escribe	D7 … D0	LEDs
A	0100 0001	Y6, Y0
K	0100 1011	Y6, Y3, Y1, Y0

PIÉNSELO

La LCD sigue mostrando la letra de los puentes. ¿Por qué?

El Arduino lee Y del 151, no lo que él manda por D9.

Si algo no funciona

Revise en este orden, sin cambiar varias cosas a la vez

Síntoma	Revise
LCD encendida, sin letras	El contraste: potenciómetro azul del módulo
LCD con cuadros en la primera línea	La dirección: `0x27` o `0x3F`
El turno no cambia o siempre es 0	El reloj; el puente BORRAR del **74LS193**; el `GND` común
El turno salta o va en desorden	`D4`, `D5` y `D6` van a los terminales 3, 2 y 6
La línea 2 nunca se completa	Un cable del turno flojo
Letra negada: `1010` `1010` y `?`	`D7` en **W** (6); va en **Y** (5)
LCD y LEDs con letras distintas	Puentes flojos; D0 es el terminal 4 y D7 el 12
En el reto, los LEDs no cambian	`D9` en el terminal 4 del 138; Enter después de la letra

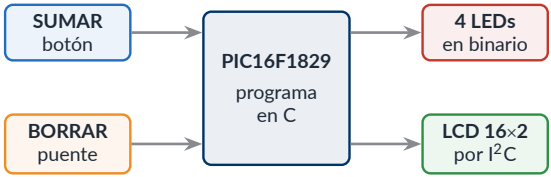
03 PARTE

ELECTRÓNICA DIGITAL

Alternativa: PIC16F1829 en C

Alternativa: contador binario con PIC16F1829

El mismo contador con LCD que los grupos con Arduino, programado en C



RIESGO ELÉCTRICO

El PIC16F1829 no es un PICAXE. El PICAXE-20M2 es ese mismo chip con un programa de fábrica. Un chip que dice PIC16F1829-I/P está vacío: el PICAXE Editor no lo reconoce.

NOTA

Cada vez que se presiona SUMAR la cuenta sube uno (0 a 15). BORRAR la regresa a 0. Sin el PICAXE Editor, sin el **CP2102** y sin el **74LS04**.

Programas que debe tener instalados

Gratuitos; instálelos antes de la clase porque piden permisos de administrador

Programa	Para qué sirve	Dónde
MPLAB X IDE (v6)	El editor: se escribe, compila y graba. Trae **MPLAB IPE**	microchip.com/mplab/mplab-x-ide
MPLAB XC8 (*Free*)	El compilador: convierte el C en un archivo `.hex`	microchip.com/mplab/compilers
PIC16F1xxxx_DFP	Los datos del chip. MPLAB X lo descarga solo, o en `Tools > Packs`	Dentro de MPLAB X
Software del programador	PICkit 3, 4, 5 o Snap: MPLAB IPE. Tarjeta **iCP03**: el software y el controlador de su fabricante	Con el programador

RECOMENDACIÓN

Compruebe en `Tools > Options > Embedded > Build Tools` que aparezca **XC8**. Con la **iCP03**, busque el **PIC16F1829** en la lista de chips de su software antes de la clase.

Compilar y grabar

Dos pasos con programas separados



COMPILAR

1. `File > New Project`: dispositivo **PIC16F1829**, compilador **XC8**.
2. Agregue el archivo `.c` del curso (carpeta `PIC`).
3. **Clean and Build**: **BUILD SUCCESSFUL**.

TARJETA iCP03

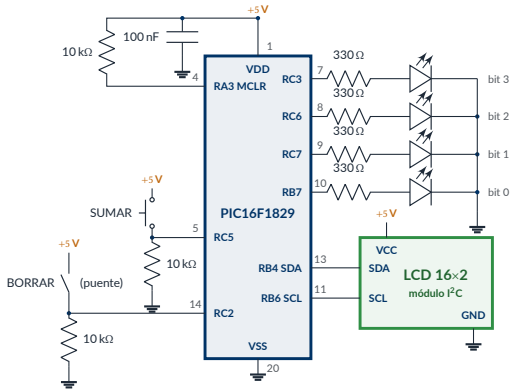
1. Selector en **8-20**; chip según la marca **20-PIN**.
2. Elija el PIC16F1829, lea el chip, cargue el `.hex`.
3. *Program* y *Verify*.

PICKIT O SNAP

En MPLAB X: **Make and Program Device**. O en MPLAB IPE: elija el chip, cargue el `.hex` y *Program*.

El circuito del PIC16F1829

El mismo contador; solo se agrega la resistencia de MCLR

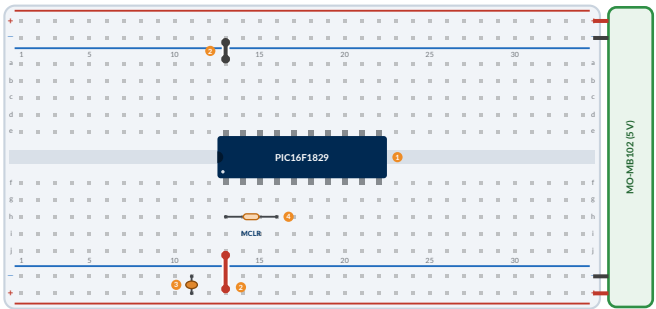


Terminal	Uso
1 / 20	VDD +5 V / VSS GND
4 (MCLR)	10 kΩ a +5 V
5 (RC5)	Botón SUMAR
14 (RC2)	Puente BORRAR
7, 8, 9, 10	LEDs: bit 3 a bit 0
13 / 11	SDA / SCL

ADVERTENCIA

VDD es el terminal **1** y VSS el **20**: al revés que en los 74LS. Sin la resistencia de MCLR, el PIC se queda reiniciado.

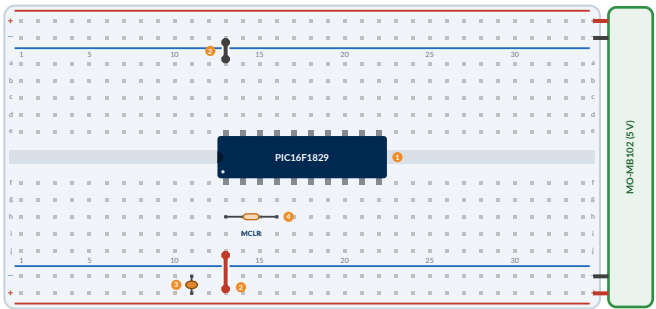
PIC16F1829: armado en el protoboard



PASO 1 • EL PIC

1. PIC **ya grabado**, columnas 13 a 22.
2. a13 al riel −; j13 al riel +.
3. 100 nF entre los rieles de abajo.
4. 10 kΩ de h13 a h16 (`MCLR`).

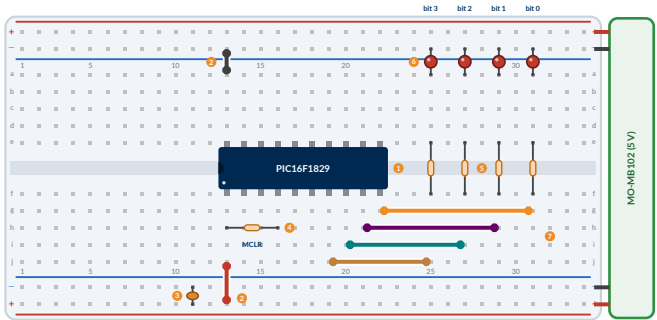
PIC16F1829: armado en el protoboard



PASO 1 • EL PIC

1. PIC **ya grabado**, columnas 13 a 22.
2. a13 al riel −; j13 al riel +.
3. 100 nF entre los rieles de abajo.
4. 10 kΩ de h13 a h16 (`MCLR`).

PIC16F1829: armado en el protoboard



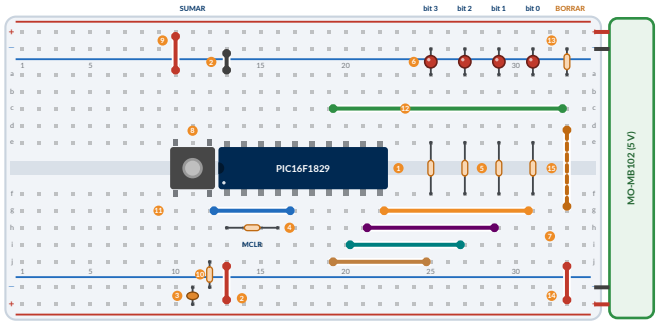
PASO 2 • LOS LEDS

5. 330 Ω cruzando la ranura en 25, 27, 29, 31.
6. LEDs: pata larga en la fila a.
7. Cables desde los terminales 7, 8, 9 y 10.

RECOMENDACIÓN

Al encender, la luz camina del bit 0 al bit 3: el PIC funciona.

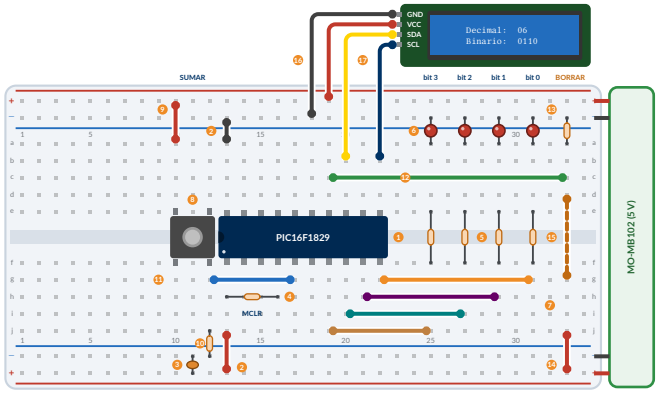
PIC16F1829: armado en el protoboard



PASO 3 • LAS ENTRADAS

8. Botón SUMAR en las columnas 10 y 12.
9. a10 al riel +.
10. 10 kΩ de j12 al riel −.
11. g12 a g17 (terminal 5).
12. c19 a c33 (terminal 14).
13. 10 kΩ de a33 al riel −.
14. j33 al riel +.
15. Puente BORRAR: d33 a g33.

PIC16F1829: armado en el protoboard



PASO 4 • LA LCD

16. `GND` y `VCC` a los rieles de arriba.
17. `SDA` a b20 (terminal 13) y `SCL` a b22 (terminal 11).

NOTA

¿Cuadros en la LCD? Cambie `0x4E` por `0x7E` en `DIR_LCD`, compile y grabe otra vez.

PIC16F1829: procedimiento y retos

Cada cambio del programa: compilar, grabar y volver a probar

PROCEDIMIENTO

- **A.** Compilar, grabar y conectar los LEDs.
- **B.** SUMAR y BORRAR: contar de 0 a 15 y volver a 0.
- **C.** La LCD: decimal y binario.
- **D.** Retos en el programa.

ADVERTENCIA

Para cambiar el programa: apague, saque el PIC sin doblar los terminales, grábelo en el zócalo y regréselo con la muesca a la izquierda.

RETO 1: CONTADOR DE DÉCADA

Que cuente de 0 a 9, como el **74LS90**.

RETO 2: CONTAR HACIA ABAJO

Que BORRAR reste uno y debajo de 0 siga el 15.

PIC16F1829: procedimiento y retos

Cada cambio del programa: compilar, grabar y volver a probar

PROCEDIMIENTO

- **A.** Compilar, grabar y conectar los LEDs.
- **B.** SUMAR y BORRAR: contar de 0 a 15 y volver a 0.
- **C.** La LCD: decimal y binario.
- **D.** Retos en el programa.

ADVERTENCIA

Para cambiar el programa: apague, saque el PIC sin doblar los terminales, grábelo en el zócalo y regréselo con la muesca a la izquierda.

RETO 1: CONTADOR DE DÉCADA

Que cuente de 0 a 9, como el **74LS90**.
Cambie `if (cuenta > 15)` por `if (cuenta > 9)`.

RETO 2: CONTAR HACIA ABAJO

Que BORRAR reste uno y debajo de 0 siga el 15.

PIC16F1829: procedimiento y retos

Cada cambio del programa: compilar, grabar y volver a probar

PROCEDIMIENTO

- **A.** Compilar, grabar y conectar los LEDs.
- **B.** SUMAR y BORRAR: contar de 0 a 15 y volver a 0.
- **C.** La LCD: decimal y binario.
- **D.** Retos en el programa.

ADVERTENCIA

Para cambiar el programa: apague, saque el PIC sin doblar los terminales, grábelo en el zócalo y regréselo con la muesca a la izquierda.

RETO 1: CONTADOR DE DÉCADA

Que cuente de 0 a 9, como el **74LS90**.
Cambie `if (cuenta > 15)` por `if (cuenta > 9)`.

RETO 2: CONTAR HACIA ABAJO

Que BORRAR reste uno y debajo de 0 siga el 15.

```
if (cuenta == 0) {
    cuenta = 15;
} else {
    cuenta = cuenta - 1;
}
```

04 PARTE
ELECTRÓNICA DIGITAL

Cierre

Preguntas de cierre

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Por qué el Arduino necesita Q_A, Q_B y Q_C además del bit?	
2	¿Por qué el Arduino lee Y y el **74LS138** recibe W?	
3	El USB manda bits por un cable sin los cables del turno. ¿Cómo sabe el receptor qué bit es cada uno?	
4	PIC: ¿qué diferencia hay entre compilar y grabar?	
5	PIC: ¿para qué sirve la resistencia del terminal 4?	

Preguntas de cierre

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Por qué el Arduino necesita Q_A, Q_B y Q_C además del bit?	El turno dice en qué lugar va el bit
2	¿Por qué el Arduino lee Y y el **74LS138** recibe W?	
3	El USB manda bits por un cable sin los cables del turno. ¿Cómo sabe el receptor qué bit es cada uno?	
4	PIC: ¿qué diferencia hay entre compilar y grabar?	
5	PIC: ¿para qué sirve la resistencia del terminal 4?	

Preguntas de cierre

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Por qué el Arduino necesita Q_A, Q_B y Q_C además del bit?	El turno dice en qué lugar va el bit
2	¿Por qué el Arduino lee Y y el **74LS138** recibe W?	Y es el bit tal cual; W va negado porque G2A se activa con 0
3	El USB manda bits por un cable sin los cables del turno. ¿Cómo sabe el receptor qué bit es cada uno?	
4	PIC: ¿qué diferencia hay entre compilar y grabar?	
5	PIC: ¿para qué sirve la resistencia del terminal 4?	

Preguntas de cierre

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Por qué el Arduino necesita Q_A, Q_B y Q_C además del bit?	El turno dice en qué lugar va el bit
2	¿Por qué el Arduino lee Y y el **74LS138** recibe W?	Y es el bit tal cual; W va negado porque G2A se activa con 0
3	El USB manda bits por un cable sin los cables del turno. ¿Cómo sabe el receptor qué bit es cada uno?	Velocidad acordada (9600 bits/s) y un bit de inicio
4	PIC: ¿qué diferencia hay entre compilar y grabar?	
5	PIC: ¿para qué sirve la resistencia del terminal 4?	

Preguntas de cierre

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Por qué el Arduino necesita Q_A, Q_B y Q_C además del bit?	El turno dice en qué lugar va el bit
2	¿Por qué el Arduino lee Y y el **74LS138** recibe W?	Y es el bit tal cual; W va negado porque G2A se activa con 0
3	El USB manda bits por un cable sin los cables del turno. ¿Cómo sabe el receptor qué bit es cada uno?	Velocidad acordada (9600 bits/s) y un bit de inicio
4	PIC: ¿qué diferencia hay entre compilar y grabar?	Compilar crea el `.hex`; grabar lo copia al chip
5	PIC: ¿para qué sirve la resistencia del terminal 4?	

Preguntas de cierre

Responda en voz alta antes de ver la respuesta

No.	Pregunta	Respuesta
1	¿Por qué el Arduino necesita Q_A, Q_B y Q_C además del bit?	El turno dice en qué lugar va el bit
2	¿Por qué el Arduino lee Y y el **74LS138** recibe W?	Y es el bit tal cual; W va negado porque G2A se activa con 0
3	El USB manda bits por un cable sin los cables del turno. ¿Cómo sabe el receptor qué bit es cada uno?	Velocidad acordada (9600 bits/s) y un bit de inicio
4	PIC: ¿qué diferencia hay entre compilar y grabar?	Compilar crea el `.hex`; grabar lo copia al chip
5	PIC: ¿para qué sirve la resistencia del terminal 4?	Mantiene MCLR en 1; en 0 el PIC se reinicia

Evaluación en Socrative

25 min Individual • 40 puntos

EVALUACIÓN EN LÍNEA (SOCRATIVE)

1. Desconecte el USB y apague el MO-MB102.
2. Abra `socrative.com` e ingrese como **Estudiante**.
3. Escriba el nombre de la sala.
4. Ingrese su **nombre completo**, tal como aparece en la lista.

NOMBRE DE LA SALA

_____



NOTA

Es la evaluación de cierre: las preguntas recorren las prácticas del curso y el circuito que usted acaba de terminar.

Lo que aprendieron en cuatro sesiones

De medir 5 voltios a un sistema que transmite y muestra una letra

5 oct	6 oct	7 oct	8 oct
S1	S2	S3	S4
Compuertas y álgebra	**El tiempo entra en juego**	**Elegir, contar y transmitir**	**El microcontrolador**
Multímetro, AND, OR, NOT, NAND y XOR; simplificar una función	Osciloscopio, reloj 555, flip-flops D y JK, contador de 2 bits	Multiplexores, decodificador y un byte por un cable	El Arduino recibe y envía la letra; LCD y puerto serie

Una computadora está hecha de estas mismas piezas: compuertas, relojes, contadores y buses serie. Hoy las vio trabajar juntas con un microcontrolador.

EN VOZ ALTA

¿Qué le costó más entender? ¿Qué sintió cuando su letra apareció en la pantalla?

¿Qué sigue?

Del circuito cableado al circuito programable

MICROCONTROLADORES

Hoy el Arduino hizo el trabajo del **74LS138** y del **74LS151** con un programa: cambiar lo que hace un circuito sin mover cables.

EN LA CARRERA

- El contador, el reloj y el bus reaparecen en arquitectura de computadoras.
- En **ciberseguridad** e **IA** el hardware también importa: UART, I2C, sensores e IoT.

CONSEJOS FINALES

- Mida antes de suponer: el instrumento no miente.
- Arme y pruebe por etapas.
- Documente con fotos, tablas y un `README` en GitHub.

RECOMENDACIÓN

Antes de salir: desarme el circuito, devuelva los componentes como los recibió y deje la estación ordenada para el próximo grupo.

UNIVERSIDAD RURAL DE GUATEMALA

¡Gracias!

Fue un gusto acompañarlos en estas cuatro sesiones de laboratorio.
¡Mucho éxito en la carrera!

github.com/MynorCifuentes