Práctica 10: Primera conexión con Arduino

Por fin vamos a conectar Arduino con Pure data y vamos a utilizar nuestro ritmo cardiaco para controlar el kick drum. Habéis visto en la lección de **Arduino** Práctica 4.1 Medimos n... | Librería CATEDU como medir vuestras pulsaciones y detectar un latido. Esta vez a parte de encender un led cuando se produzca un latido, vamos a mandar un mensaje a traves del **puerto Serial**.

¿Qué es un puerto Serial?

Es una interfaz de comunicación a través de la cual se envían datos de manera secuencial. Digamos que los datos van en una sola fila uno detrás de otro. Si has seguido el orden propuesto, ya habrás visto este apartado sobre el puerto serie.

Ahora en lugar de ser la app oscControl la que nos envíe datos a través de osc, será **Arduino** el que envíe esos datos a través del **puerto Serial**. En Arduino vamos a recoger y procesar los valores que el sensor de latidos nos dé. Tras procesar esos valores vamos a identificar cuándo **se produce un latido**, y cuando ese latido se produzca vamos a **enviar un mensaje** a Pure data gracias al puerto serie para activar el kick drum que hicimos en la Práctica 4: Kick drum ... | Librería CATEDU.

El ritmo de nuestro corazón se va escuchar en nuestros altavoces y sonará como un kick drum. Con este ritmo podríamos controlar cualquier evento de Pure Data; es un juego de números.

En Arduino cuando detectamos una pulsación aparte de encender el LED vamos a imprimir en el puerto Serial un 1.

El código en Arduino

Copyright 2026

A continuación, te dejamos el código que necesitará tu Arduino para reconocer cuándo se ha producido un latido. En los comentarios, puedes leer qué función tiene cada línea.

```
int leer = 0; //esta variable almacenará los valores leídos por nuestro sensor

void setup() {
 Serial.begin(9600); //iniciamos la comunicación con el puerto serie a 9600 baudios

}

void loop() {
leer = analogRead(A0); // almacenamos en la variable 'leer' los valores que detecta nuestro sensor

if(leer > 550){ // consideraremos que un valor >550 es un latido. Si se produce uno...
  Serial.println("1"); //enviaremos un '1' a Pure Data gracias al puerto serie.
}

Serial.println(); // enviamos una línea en blanco
delay(200); // esperamos 200 ms para realizar una nueva lectura
}
```

Conexiones en Arduino

Las conexiones que tienes que realizar entre el pulsómetro y tu Arduino, son las mismas que en esta práctica.

Recibir Serial en Pd

Como ya hemos visto, cada vez que Arduino detecte un latido enviará un '1' a Pure Data. Cuando recibamos ese '1' vamos a activar el kick drum.

Copyright 2026

¿Cómo vamos a recibir ese mensaje? Utilizando el objeto **"comport"**. En él tendremos que **configurar qué puerto serie** va a escuchar y la **velocidad de comunicación**. Tanto el **puerto** como la **velocidad** tienen que **coincidir** con las utiliza nuestro **Arduino**. En mi caso es el COM 5 y la velocidad son 9600 baudios:

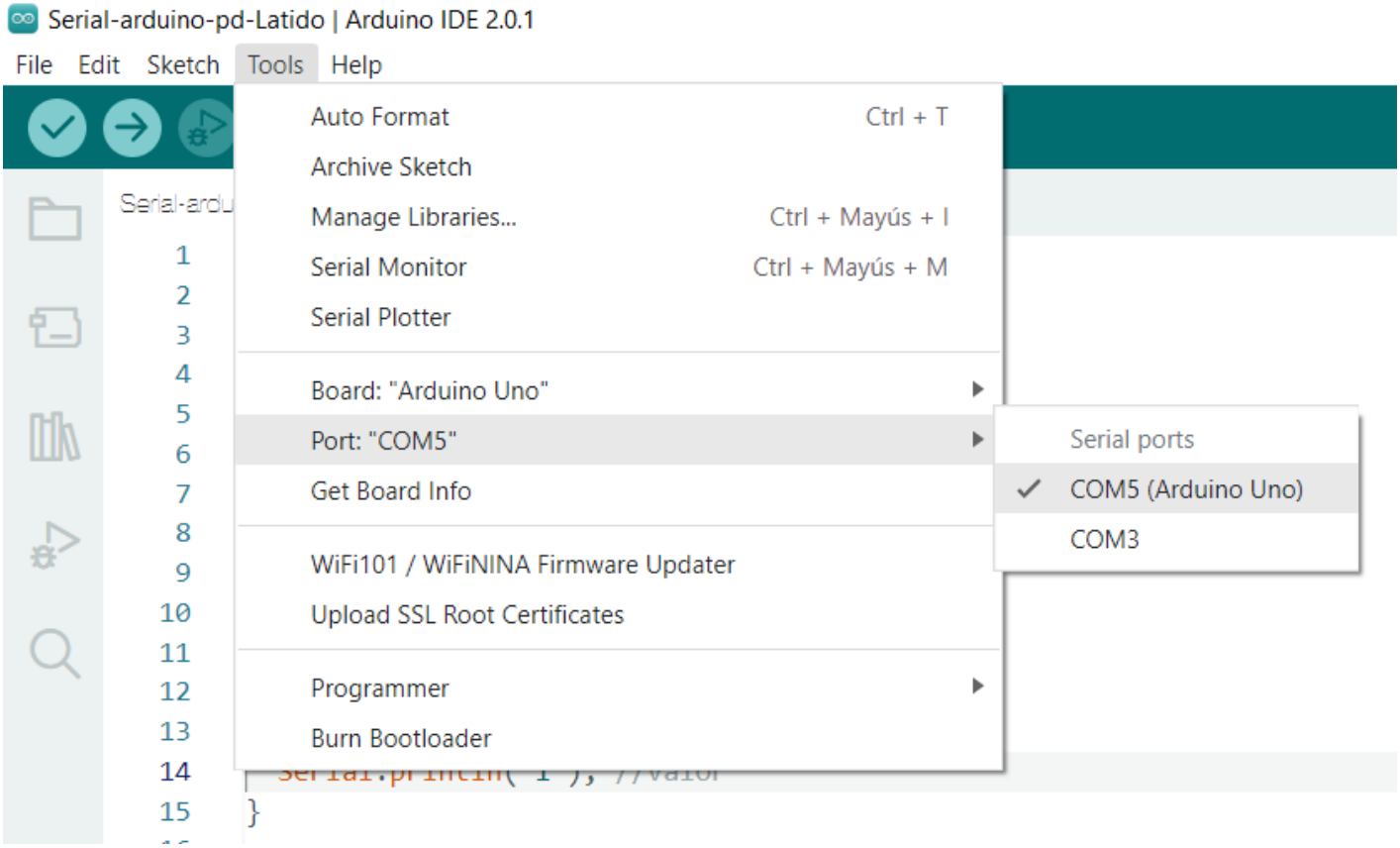


Figura 1. Arduino utilizando el puerto serial: "COM5"

```
void setup() {
 Serial.begin(9600);

}
```

Figura 2. Velocidad del puerto serial en Arduino configurada a 9600 baudios.

El objeto **"comport"** no está incluido en Pure Data Vanilla, Forma parte de una librería externa que tendremos que instalar. Os dejo un video de como instalar librerías. No tenéis que instalar la librería que instalo yo en el video.

https://www.youtube.com/embed/pNG0yKeaNXM

Para esta práctica vamos a instalar dos **librerías**: "**comport**" y "**cyclone**"

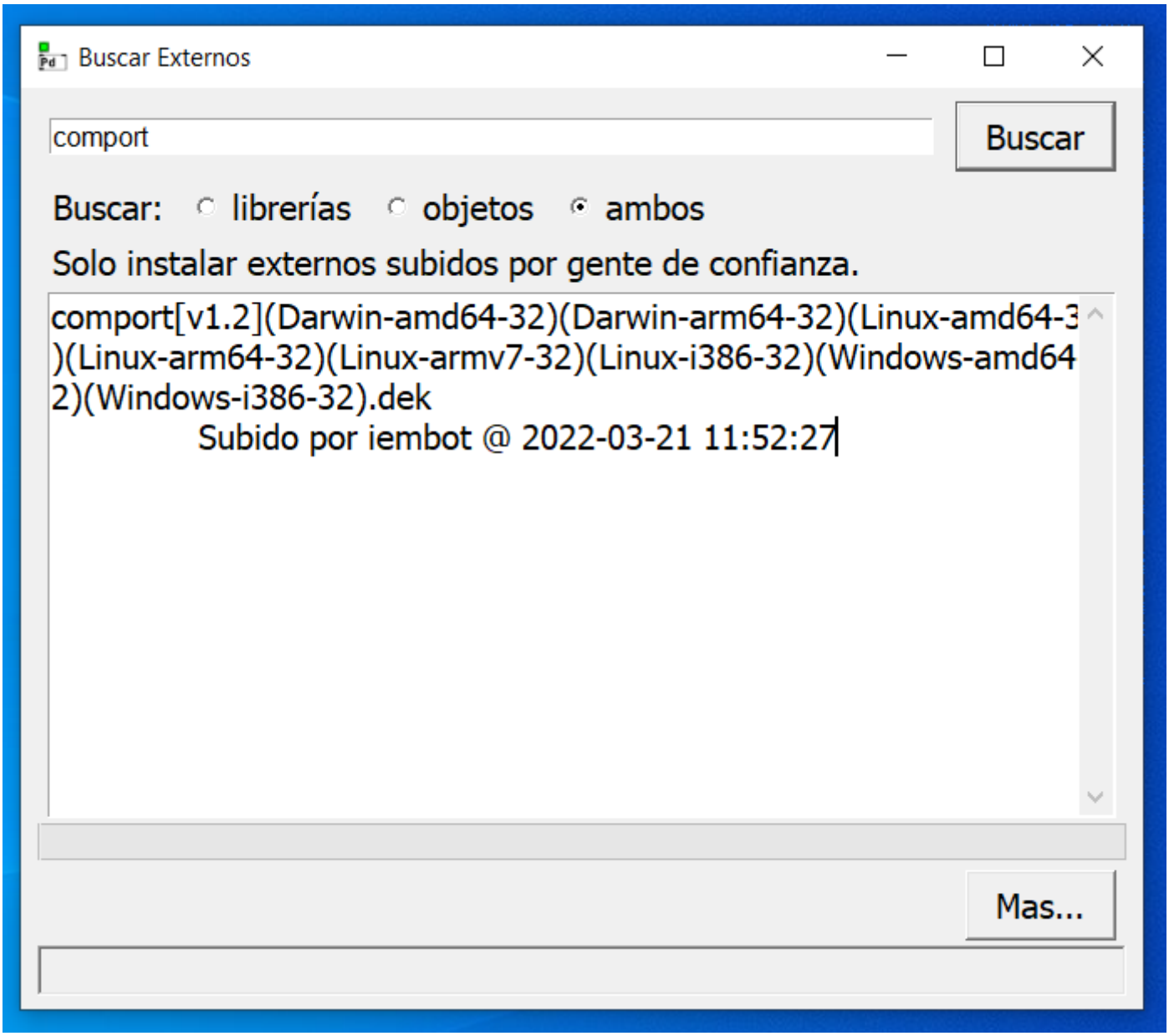


Figura 3. Proceso de instalación de librería comport.

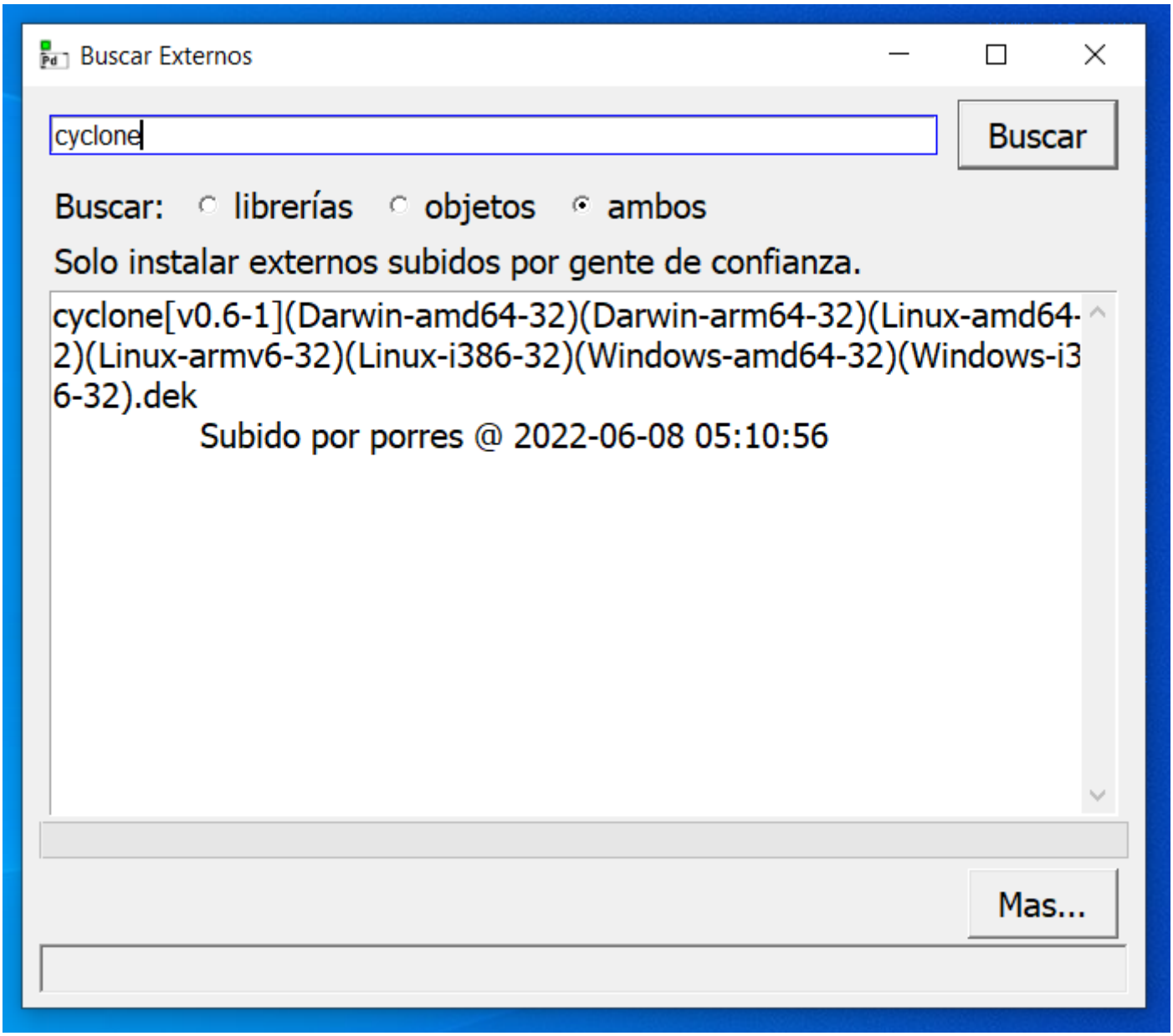
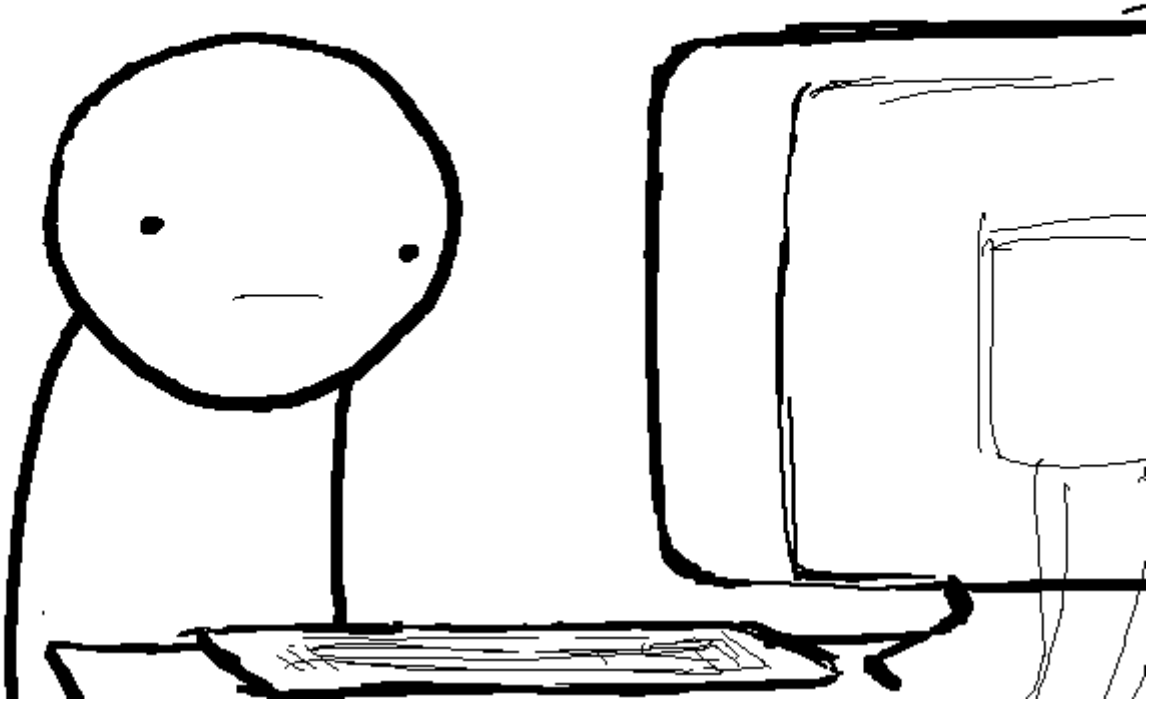


Figura 4. Proceso de instalación de librería cyclone.

¿Ya tenéis las dos librerías instaladas y registradas en inicio para que se carguen al abrir Pd?

Copyright 2026



Al objeto **"comport"** le indicaremos el **puerto** en el **primer argumento** y la **velocidad en el segundo**. Recordar que el puerto va a ser la dirección donde Arduino enviara los datos, y sera en esa dirección donde Pure data tiene que ir a buscarlos. El mensaje **device** imprimirá en la ventanita de Pd la **lista de puertos** serial disponibles, tendréis que escoger aquel que coincida con el que utiliza vuestro Arduino. Para **abrir el puerto** enviaremos el **mensaje open**, en ese mensaje también podemos especificar el identificador del puerto que vamos a abrir, en mi caso es el 5. Para **cerrar la comunicación** enviaremos un mensaje con la palabra "**close**".

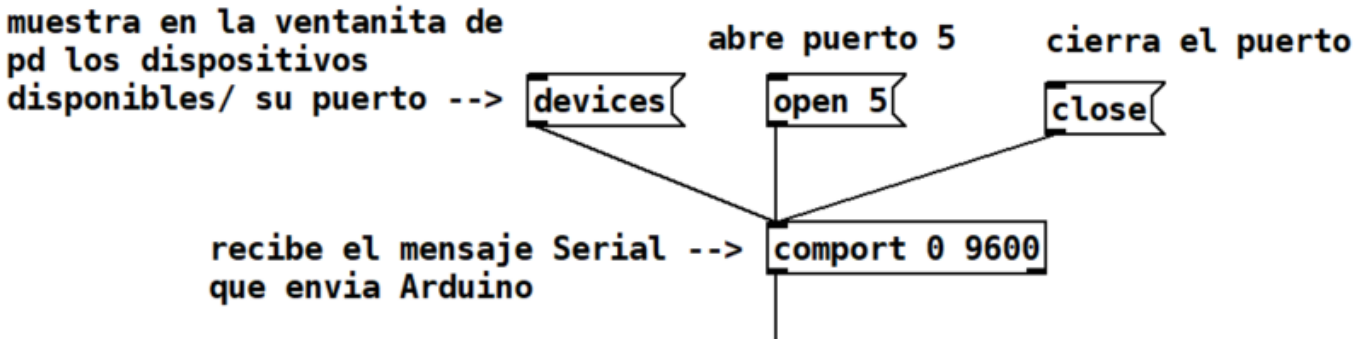


Figura 5. objeto "comport"

> Cerrar la ventana de Serial en Arduino para abrir la comunicación de pure data, si no, no os dejara. Para cargar el programa en Arduino también necesitaremos cerrar el puerto en pure data, de lo contrario nos dará error. No podemos tener dos elementos conectados a la vez al mismo puerto serial

Una vez recibido el mensaje en Pure Data tendremos que **decodificarlo**, para ello vamos a utilizar el siguiente grupo de objetos. Ahora no vamos a entrar a explicarlo. Simplemente lo utilizaremos ta

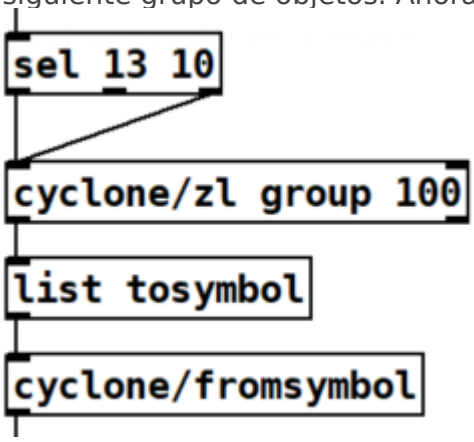


Figura 6. Grupo de decodificación para los datos que llegan a traves de serial a Pure Data.

A este grupo (figura 6) le llegará el **outlet** del objeto **"comport"**, Y de su ultimo objeto **"cyclone/fromsymbol"** obtendremos la información que Arduino envió por el puerto serie, en nuestro caso es el **número 1**. Con ese número activaremos un bang que activará nuestro kick drum. Lo haremos de la misma forma que en la práctica de osc pero esta vez utilizaremos un random para que con cada latido se active un uno de los 3 valores iniciales del envelope que regula la frecuencia del kick drum.

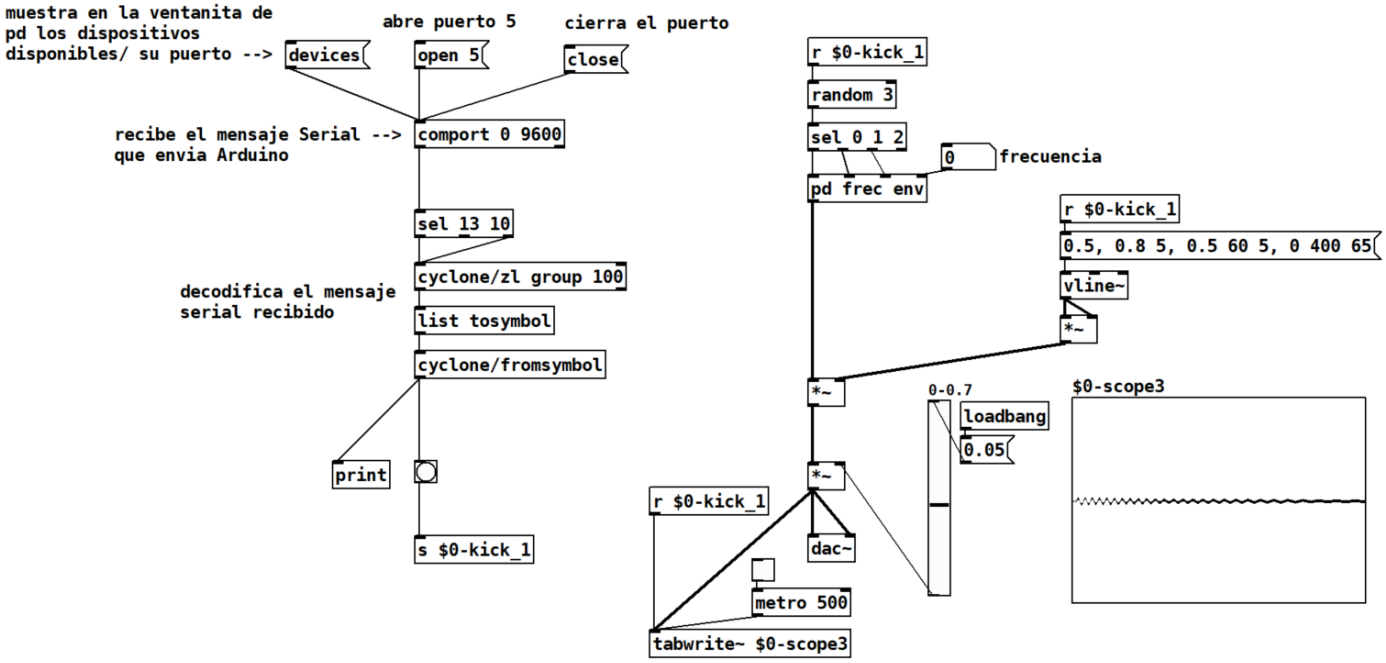


Figura 7. patch de Pd como el que tenéis que construir en esta práctica 10.

> Recordad que tenéis que tener encendido el DSP de pure data para procesar sonido.

¿Qué tenemos que entregar?

Sube a la carpeta del Moodle de la Práctica 10 los archivos de Arduino y Pure Data que has utilizado para convertir tu latido en un kick drum y escucharlo en tus altavoces. **Sube un video en el que se vea tu dedo en el sensor, el led encendiéndose con tus latidos y que se escuche el kickdrum también al ritmo de tus latidos.**

Archivos que tenéis que subir:

- patch Pd ***Practica10_Pd-vuestronombre_vuestroapellido.pd.***
- patch Arduino ***Practica10_Ardu-vuestronombre_vuestroapellido.ino.***
- video ***Practica10_Video-vuestronombre_vuestroapellido***

Copyright 2026

Figuras:

Figura 1. Arduino utilizando el puerto serial: "COM5"

Figura 2. Velocidad del puerto serial en Arduino configurada a 9600 baudios.

Figura 3. Proceso de instalación de librería comport.

Figura 4. Proceso de instalación de librería cyclone.

Figura 5. objeto "comport"

Figura 6. Grupo de decodificación para los datos que llegan a traves de serial a Pure Data.

Figura 7. patch de Pd como el que tenéis que construir en esta práctica 10.

Revision #23
Created 2022-11-03 12:13:17 CET by Julia del Río
Updated 2022-12-05 13:32:12 CET by Julia del Río

Copyright 2026