Introducción a los Sistemas Ciberfísicos - 2023

TRABAJO PRÁCTICO FINAL

Informe Pastillero Electrónico

Comisión: Única - Grupo: 1

Alumnos:

- Dayub, Mateo José
- Jurado Kokoyaczuk, Juan Manuel
- Mazzieri, Federico Nicolás
- Ponce, Baltazar
- Rufiner, Santiago
- Savat, Emanuel
- Silvero, Milton

Docentes:

- Borzone, Eugenio
- Carrique, Juan
- Molas Gimenez, Jose Tomas
- Zorzet, Bruno Jose

Fecha de entrega: Miércoles 22 de Noviembre

Resumen

En este trabajo final realizamos el modelado de un pastillero electrónico “inteligente”, el cual cuenta con una alarma para notificar al usuario la toma de sus medicamentos. Para esto abordamos la toma de decisiones referente a los componentes de hardware elegidos, el diseño del prototipo físico y la escritura del código utilizando el IDE de Arduino, con el fin de solucionar la problemática de tener muchos horarios diferentes para tomar medicamentos.

Introducción

Arduino es una plataforma de prototipado de hardware de código abierto que combina hardware y software de una manera intuitiva de usar, diseñada para personas con o sin experiencia en electrónica y programación, permitiendo crear proyectos de manera accesible. La plataforma se compone de una placa con un microcontrolador y un entorno de desarrollo integrado (IDE) que facilita la programación. Arduino se utiliza para crear proyectos electrónicos de todo tipo, desde proyectos sencillos para aficionados como puede ser prender un led con un switch y una resistencia, hasta proyectos mucho más complejos en entornos de desarrollo profesionales [1].

Para este trabajo final se nos dió a elegir entre dos posibles proyectos de final de cursado: un proyecto de trazabilidad térmica o la creación de un pastillero electrónico, decantándonos por este último, ya que de entrada nos gustó más el prototipo del pastillero expuesto por las profesoras de FADU en clases, y podíamos imaginarnoslo más a detalle, tanto su sistema físico como sus funcionalidades implementadas tanto en hardware como en software. Además, como condición de trabajo se nos pidió que utilicemos el IDE open source de Arduino, y se nos propuso utilizar componentes de hardware reutilizados en el taller de roboticlaje impartido por la cátedra, y cómo microprocesador las placas ESP32 también facilitadas por la cátedra o algún microprocesador que pudiéramos conseguir y que pueda cumplir con los requerimientos del proyecto.

En este trabajo final integrador aplicamos los conocimientos vistos en el cursado, tanto de programación en arduino como de componentes electrónicos, con el objetivo de modelar y construir un pastillero electrónico que sea capaz de comunicarle al usuario por medio de un disparador de alarma la hora a la que este debe consumir sus medicamentos.

Desarrollo

Descripción del sistema

Nuestro pastillero electrónico consta de 4 compartimientos (slots), cada uno con su propio led. Esto permite que el dispositivo maneje diferentes tipos de pastillas, y cada tipo tiene su propio compartimento. El led correspondiente se ilumina para indicar de qué compartimento se debe tomar una pastilla.

Como sensores, utilizamos un switch y un reloj/controlador interno. Por otro lado, los actuadores están constituidos por un parlante y leds, en ambos casos para alertar al usuario cuando y de donde debe tomar la/s pastilla/s.

Cuando llega el momento de tomar una pastilla (basado en el reloj interno), el arduino activa el parlante para alertar al usuario. Al mismo tiempo, activa un led específico para indicar qué tipo de pastilla debe ser tomada. Cuando el usuario presiona el botón el sistema de actuadores se apaga.

Todo este sistema está alimentado con baterías o con el microcontrolador desde una laptop.

Nuestro pastillero debe cumplir con las siguientes condiciones:

- Setear el horario de la/s alarma/s y controlar constantemente si el horario de esta/s coincide con la hora actual.
- Avisar al usuario de que debe consumir la pastilla.
- Sensar si el usuario tomó los medicamentos.

Desarrollo General: Paso a paso

1) **Diseño del prototipo de pastillero:** Durante las clases hicimos una puesta en común como iba a ser nuestro prototipo: ¿Que forma va a tener? ¿Con qué herramientas contamos para desarrollar un prototipo básico? ¿A qué clase de usuario va a estar dirigido? ¿De qué materiales va a estar hecho? Decidimos que, por cuestiones tanto

de simplicidad como de conveniencia, debíamos hacerlo con una forma simple y cuadrada. Dentro de nuestro grupo contamos con variedad de herramientas, decidimos también que nuestro público serían aquellos usuarios que tengan que tomar muchas pastillas en su dia a dia y quieran facilitar su vida. Finalmente también decidimos realizar la carcasa del pastillero con método de impresión 3D en PLA.

2) **Reciclado de componentes (Roboticlaje):** Realizamos un taller de reciclado y reutilización de chatarra electrónica donde se nos dio la oportunidad de reciclar materiales/componentes electrónicos para poder utilizarlos en nuestro proyecto, y de esta manera poder ahorrar gastos en la compra de componentes. Durante este proceso de reciclaje, se nos dio la oportunidad de pulir habilidades como soldar, desarmar placas y clasificar componentes.

3) **Simulación en simulIDE/Wokwi:** Utilizando el software Wokwi propuesto desde la cátedra para el desarrollo del tp, sumado a nuestro conocimiento ya aprendido en otras materias y junto con los nuevos conocimientos en programación arduino, trabajamos en conjunto para proyectar cómo sería la lógica de nuestro sistema, para luego poder plasmarlo en código. A lo largo de esta etapa nos topamos con diversos problemas: “¿Cómo hacemos para que nuestro sistema sepa qué hora es? ¿Cómo hacemos que este prenda los leds correspondientes cuando suene la alarma?”

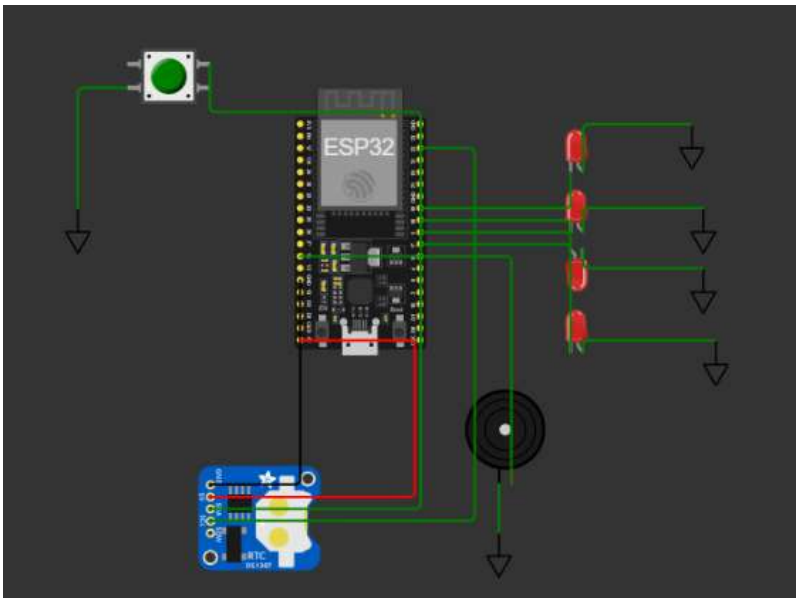


Imagen 1. Simulación de la parte electrónica hecha en el software Wokwi.

4) **Creación de código en Arduino IDE:** Después de haber terminado de proyectar la lógica en el simulador, nos pusimos a codear la solución del proyecto en el IDE de código abierto de Arduino, ideando varios algoritmos y probando distintas soluciones. Llegamos a que la solución más eficiente sería que los horarios recetados pudiesen ser configurados dentro de una estructura Switch-Case, lo que permitiría establecer múltiples combinaciones de recetas, es decir, se podrían combinar pastillas e incluso setear horarios múltiples para los mismos medicamentos. Se utiliza un librería que permite al bucle ser referenciado por un reloj o dispositivo RTC mediante wifi, que incluso puede ser mostrado al usuario si se agrega un display al dispositivo. Además está programado para que cuando se detecte un horario recetado, se enciendan LEDs junto con un buzzer para alertar al usuario, el cual los desactivará con un pulsador o en su defecto se desactivaran pasado 1 minuto.

5) **Proyecto físico:** Luego de realizado el código, probamos conectar los componentes en una protoboard y procurar que el algoritmo funcione. También ideamos una carcasa para el pastillero que fue mutando con el pasar de los días, para convertirse en un prototipo lo más sencillo y funcional posible.

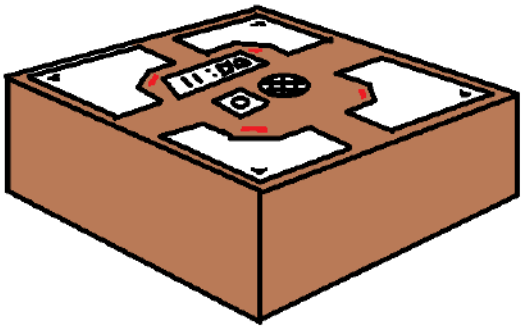


Imagen 2. Ilustración del prototipo físico de pastillero electrónico.

Listado de materiales seleccionados

componentes	cant.Necesitada
botones	4
leds	4
bocina	1
micro_BluePill stm32	1
resistores	5 - 1
carcasa	1

Imagen 3. Listado de componentes seleccionados para el desarrollo del pastillero electrónico.

Descripción de materiales

- ***Microcontrolador:*** STM32F103C8T6 Blue Pill

- Voltaje de operación: 3.3V.
- Entradas analógicas: 10.
- Pines de entrada/salida digitales: 37.
- Fuente/captación de corriente continua desde pines de entrada/salida: 6mA.
- Memoria flash (KB): 64/128.
- SRAM (memoria de acceso aleatorio): 20 KB.
- Frecuencia (velocidad del reloj): Máximo 72MHz.
- Comunicación: I2C, SPI, UART, CAN, USB.
- *Hojas de especificaciónes:* https://www.st.com/resource/en/datasheet/stm32f103c8.pdf

- ***LEDs:*** 5MM-R

Su consumo, color, voltaje de operación e intensidad dependen de la longitud de onda configurada.

- Voltaje de operación: 2V.
- Temperatura de operación: -40 a 70°C.
- Ángulo de apertura: 15° a 30°.
- Cantidad de pines: 2 o 4 (dependiendo el modelo).
  https://agelectronica.lat/pdfs/textos/L/LEDY5DYT.PDF

- ***Pulsador:*** A22-25mm

- Corriente nominal de carga: 10V.
- Operación momentánea: 30 minutos máx.
- Temperatura ambiente: -20°C a 70°C.

El resto de especificaciones depende de las decenas de sub modelos diferentes de este pulsador.

https://assets.omron.eu/downloads/datasheet/es/v3/a128_a22_pushbutton_switch_datasheet_es.pdf

- ***Buzzer:*** (2-10-18)

- Voltaje de funcionamiento: 3.3V ~ 5V.
- Corriente de operación: < 25mA.
- Salida de sonido min a 10 cm: 85 dB.
- Frecuencia emitida: 2300 ± 500 Hz.
- Material: PCB.
- Temperatura de funcionamiento: -27 a 70°c.
- Temperatura de almacenamiento: -30 a 105°c.
- Dimensiones: 1.9x1.5x1.2cm.
- Peso: 3g.

https://www.bolanosdj.com.ar/MOVIL/ARDUINO2/BuzzerActivo.pdf

- ***Resistencia:*** CCF-55

- Rango de valores: 10A 1M(serie E96).
- Disipación 1/4W a 70ºC (1/2W disponible sobre pedido).
- Temperatura de operación: -65C a 165ºC.
- Coeficiente de temperatura: 100 PPM/ºC.
- Máxima tensión de trabajo: 250 VRMS.
- Marcaje: estándar de 5 bandas.

https://www.juntadeandalucia.es/averroes/centros-tic/29009909/helvia/aula/archivos/_56/RESISTENCIAS-TABLAS.pdf

- ***Display:*** LCD 16x2 I2C.

- Dimensiones: 18 x 80 x 37 mm.
- Cantidad de líneas: 16.
- Comunicación: I2C.
- Conexión: 4 cables (2 de alimentación y 2 de datos).
- Voltaje de alimentación: 5V.

https://es.scribd.com/document/442952281/11-Pantalla-LCD-16x02-con-I2C

Proveedores

Al haber obtenido todos los componentes en el taller de roboticlaje y por prestación de algún conocido/amigo/familiar cercano, es que no tuvimos que gastar plata para conseguir ningún material, por lo que nuestros únicos proveedores fueron:

- Taller de roboticlaje (componentes varios reciclados en el taller, como resistencias, leds, interruptores, bocina).
- Familiar de uno de los integrantes del grupo (Blue pill stm32, protoboard y componentes varios).

Desde ya agradecemos la colaboración de todos ellos para poder llevar adelante el proyecto.

Criterios de selección de cada material listado

Para la selección de los materiales necesarios para nuestro proyecto, nos basamos en los siguientes criterios, en orden de importancia:

- **Disponibilidad:** priorizamos utilizar componentes y materiales que pudimos conseguir a partir de nuestro entorno más cercano.

- **Reciclaje y reutilización:** buscamos aprovechar al máximo los componentes que pudimos reciclar y reutilizar en el taller de roboticlaje. Esto contribuye no solo a la reducción de costos, sino también a la sostenibilidad ambiental del proyecto.

- **Precio:** priorizamos materiales y componentes que no implicaran un gasto significativo o, en su defecto, que tuvieran un costo mínimo.

- **Funcionalidad y compatibilidad:** evaluamos si los materiales cumplían con los requisitos funcionales específicos del proyecto, y verificamos que los materiales seleccionados sean compatibles entre sí.

- **Calidad:** evaluamos la resistencia y durabilidad de los materiales, priorizando aquellos que puedan soportar las condiciones de uso previstas.

Al seguir estos criterios en orden de importancia, buscamos minimizar costos, optimizar la eficiencia y garantizar la utilización de los materiales más adecuados para llevar adelante el pastillero electrónico.

Algoritmos principales

Código/pseudocódigo

Para el proyecto utilizamos la librería de RTC, es decir el módulo de reloj que nos permite obtener la hora para poder activar las alarmas en el momento correspondiente.

En un principio, establecemos las constantes que vamos a utilizar en el proyecto, como los pines correspondientes a los cuatro leds, el buzzer y el botón, además de aquellas necesarias para la configuración del buzzer, como la resolución y frecuencia. Por otro lado, también generamos una instancia de la clase RTC_DS3231, para poder hacer uso de nuestro módulo de reloj.
En este punto, también definimos un array de alarmas, el cual almacena el tiempo en el que debe sonar cada una de ellas además del pin correspondiente al led de ese compartimiento (cuatro en total).

En la función setup configuramos los sensores y actuadores a sus correspondientes pines y hacemos una pequeña validación para asegurarnos que el módulo rtc haya sido encontrado.

Finalmente, en la función loop iteramos sobre el array de alarmas y contrastamos las horas “objetivo” con la hora actual. Si alguna de ellas coincide, hacemos sonar el buzzer y encendemos el led correspondiente a ese compartimiento.
Por otro lado, también verificamos si el botón fue presionado, en cuyo caso apagamos el buzzer y los leds.

El código se visualiza a continuación (también se puede ver en el repositorio puesto en las Referencias):

```
#include <RTClib.h>

struct Alarm {
  int hour;
  int minute;
  int pin;
};


const long  gmtOffset_sec = -10800;
const int   daylightOffset_sec = 3600;


RTC_DS3231 rtc;

const int buttonPin = 2;
bool buttonPressed = false;

const int led1Pin = 19;
const int led2Pin = 18;
const int led3Pin = 5;
const int led4Pin = 13;
const int buzzerPin = 14;

const int buzzerFrequency = 500; // Frecuencia del zumbador en Hz
const int buzzerDuration = 500;   // Duración del zumbador en milisegundos
const int channel = 0;
const int resolution = 8;

Alarm alarms[4] = {{8,0,led1Pin},{12,0,led2Pin},{19,31,led3Pin},{23,0,led4Pin}};

bool ledcAttach(uint8_t pin, uint32_t freq, uint8_t resolution);
```

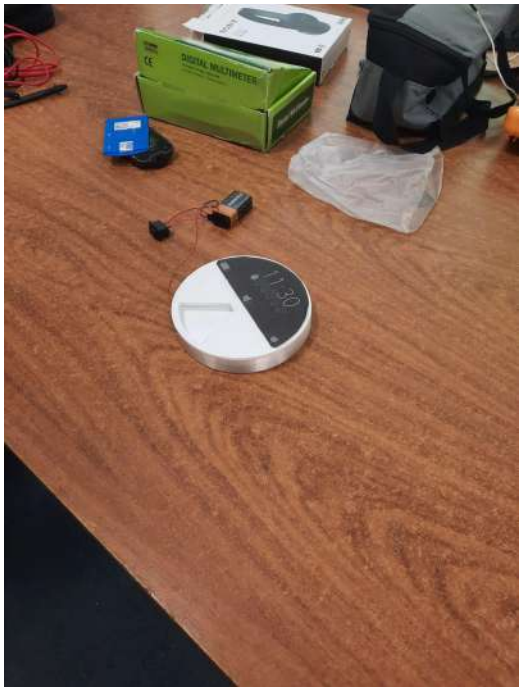
```
void activateBuzzer() {
  ledcAttachPin(buzzerPin, channel); // Asocia el pin del buzzer con el canal PWM
  ledcSetup(channel, buzzerFrequency, resolution); // Configura el canal PWM
  ledcWrite(channel, 255); // Inicia el tono (255 es el máximo valor para un ciclo de trabajo de 8 bits)
}


void deactivateBuzzer() {
  ledcWrite(channel, 0); // Establece el ciclo de trabajo en 0, deteniendo el tono
}


void removeAlerts() {
  deactivateBuzzer();
  digitalWrite(led1Pin, LOW);
  digitalWrite(led2Pin, LOW);
  digitalWrite(led3Pin, LOW);
  digitalWrite(led4Pin, LOW);
}

void displayAlert(const int ledPin) {
   digitalWrite(ledPin, HIGH);
   activateBuzzer();
}


void setup() {
  Serial.begin(115200);
  Wire.begin();
  rtc.begin();
```

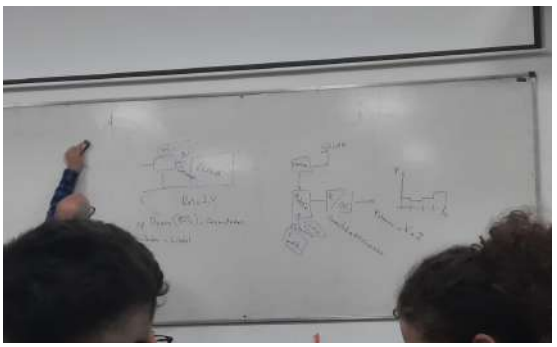
```
    if (! rtc.begin()) {
    Serial.println("Couldn't find RTC");
  }

  pinMode(led1Pin, OUTPUT);
  pinMode(led2Pin, OUTPUT);
  pinMode(led3Pin, OUTPUT);
  pinMode(led4Pin, OUTPUT);

  pinMode(buttonPin, INPUT_PULLUP);

  pinMode(buzzerPin, OUTPUT);
  }

void loop() {
  DateTime now = rtc.now();

  for (Alarm alarm : alarms) {
    if (now.hour() == alarm.hour && now.minute() == alarm.minute) {
      displayAlert(alarm.pin);
    }
  }


  if (digitalRead(buttonPin) == LOW) {
    if (!buttonPressed) {
      // Botón recién presionado, reiniciar el ESP32
      Serial.println("Button pressed, restarting ESP32");
      removeAlerts();
      buttonPressed = true;
      delay(60000);
    }
  } else {
    buttonPressed = false;  // Restablecer el indicador cuando el botón se suelta
  }
```

Imágenes 4, 5 y 6. Código del pastillero electrónico.

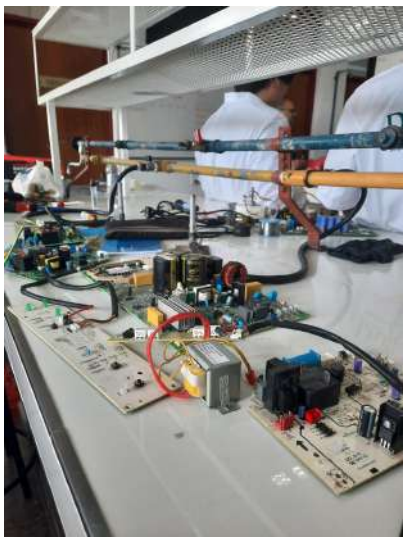
Evidencias del proceso de implementación y desarrollo del proyecto

1) **Clase inicio del proyecto:**





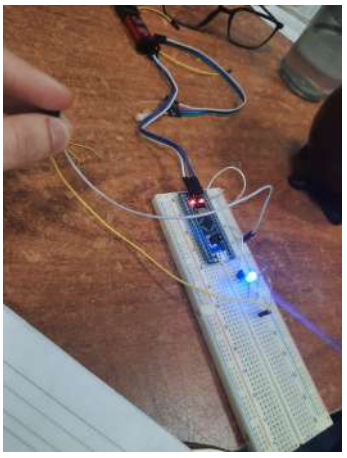
2) **Roboticlaje:**



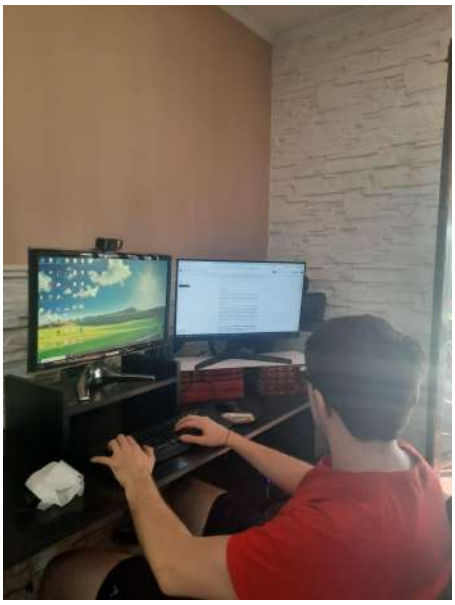
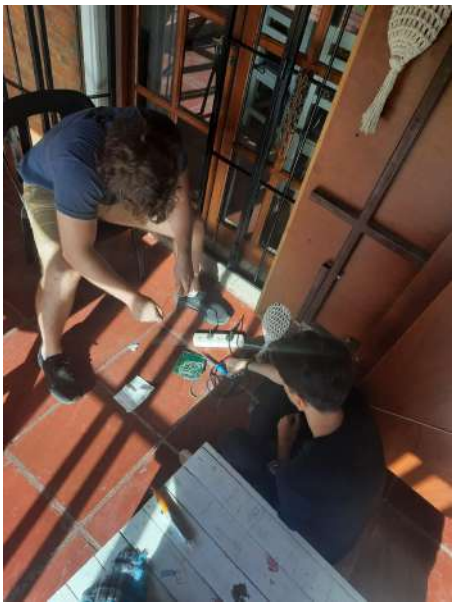
3) **Componentes:**

4) **Reciclaje en sinc(i) e inicio del código en Arduino:**

5) **Introducción en Protoboard (Trabajo en equipo con el grupo 3):**



6) **Juntada Grupal:**


Conclusiones y Aprendizajes

Como conclusión, creemos que el proyecto nos ayudó mucho a conocer y tocar las bases de la programación Arduino con microcontroladores, y que nos abrió la puerta a tener un mejor proceso de aprendizaje en las futuras asignaturas de la carrera.

Nuestra meta como grupo fue aprender a implementar un sistema ciberfísico desde cero, por lo que decidimos bajar la dificultad lo más posible debido a que es nuestra primera experiencia en esta área y ninguno de nosotros tenía conocimientos previos de electrónica, por lo que solo tuvimos las herramientas adquiridas en clase y en la bibliografía de la materia. Una vez podido resolver nuestro diseño evaluamos la posibilidad de seguir automatizando nuestro proyecto (tanto dentro como fuera de la materia).

Pensamos que pudimos cumplir con el desafío propuesto, llevando adelante un proceso multidisciplinario en el que cada integrante del grupo tuvo que aportar sus conocimientos para que estos se combinaran lo mejor posible con los aportes del resto y así llevar adelante el proyecto. Nos dimos cuenta de la cantidad de habilidades que tenemos y la cantidad de cosas que no sabemos, y que dentro del ámbito del trabajo en equipo, encontrar los puntos de carencia y fortaleza de cada uno es la clave del éxito de cualquier proyecto.

Referencias

Repositorio con el proyecto:

https://github.com/Baltazarky/Grupo1_SistemasCiberFisicos

Otras referencias

1. Fernández, Y. (2022, September 23). *Qué es Arduino, cómo funciona y qué puedes hacer con uno*. Xataka. Retrieved November 18, 2023, from https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno