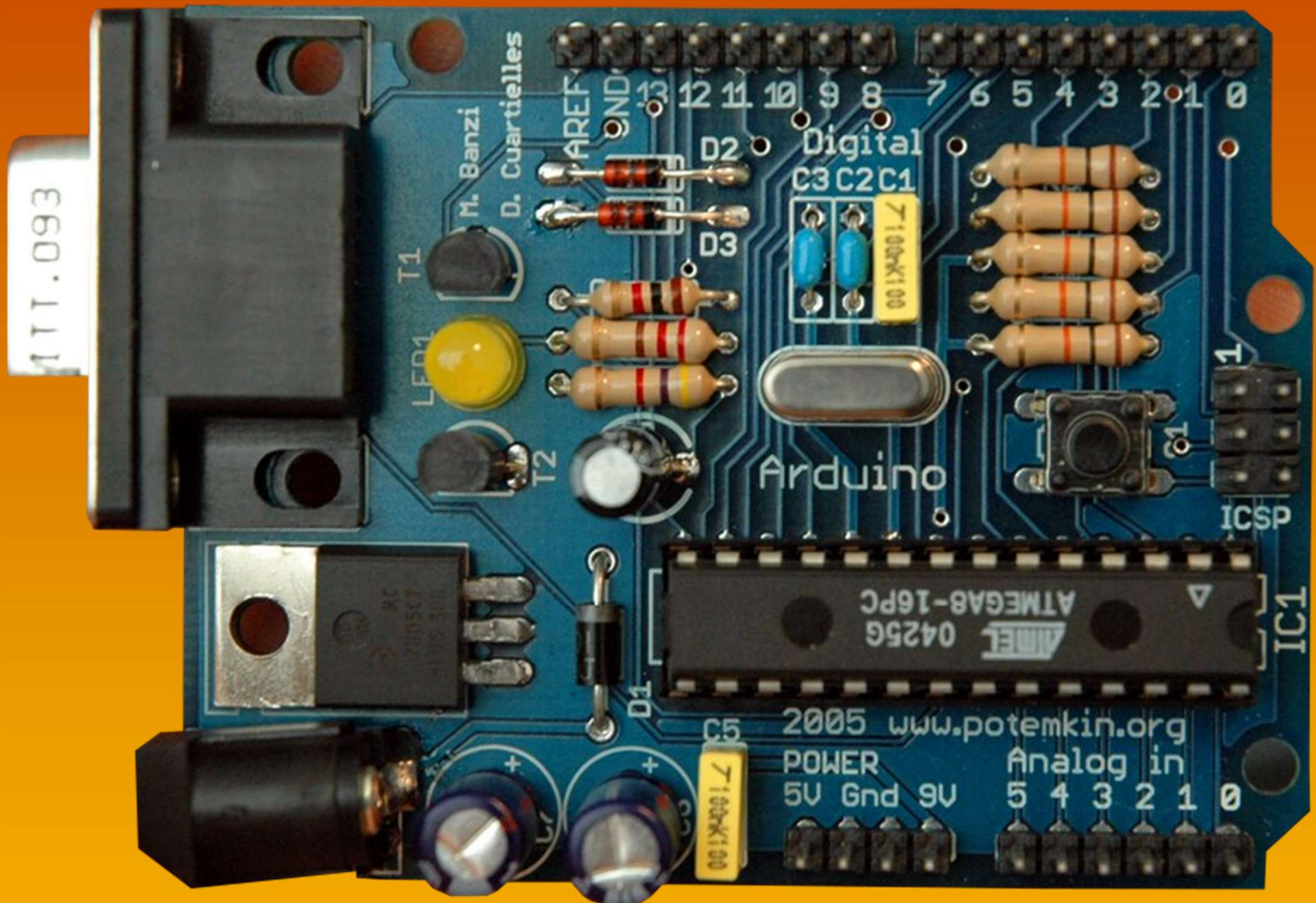




# Nerds Only

09.2023



Arduino

Source: Wikipedia

Massimo Banzi

## Arduino

It all started in Ivrea (Italy), and not at a technical faculty, but at the Institute of Interaction Design in Ivrea (IDII). There, Colombian student Hernando Barragán presents the Wiring project in his master's thesis. The aim of this project was to give designers and artists access to electronics and programming in a simplified form. He virtually created the predecessor of Arduino. His supervisor (doctoral advisor) Massimo Banzi develops from it in 2005 because the first Arduino, together with other founders, Hernando Barragán is somehow passed over [1]. However, in 2003 Massimo Banzi has already developed the board Programma2003, which from his point of view is the predecessor of Arduino.

The name "Arduino" was taken from a bar in Ivrea where some of the project founders used to meet (the bar itself was named after Arduin of Ivrea, who was also King of Italy from 1002 to 1014).

I haven't found an official explanation for the logo, but from my point of view + and - stand for voltage source, which is representative for hardware, and the lying eight wraps around it like a ribbon, which is representative for the SW, i.e. the code. And that's what Arduino represents, a board with associated programming environment.

The Arduinos are all based on microcontrollers from Atmel, starting with the ATmega8. Today the ATmega328P is used.

The Arduino board concept was later taken up by other companies, so in my basic version I use NodeMCU with the microcontroller ESP8266 from Espressif. More about this in another issue.

PS: If you zoom in on the photo of the Arduino, you can find "Arduino", "Banzi" and "ATmega8".

The story told by Hernando Barragán: <https://arduinohistory.github.io/de>

The development history as a poster: [https://blog.experientia.com/uploads/2013/10/Interaction\\_Ivrea\\_arduino.pdf](https://blog.experientia.com/uploads/2013/10/Interaction_Ivrea_arduino.pdf)

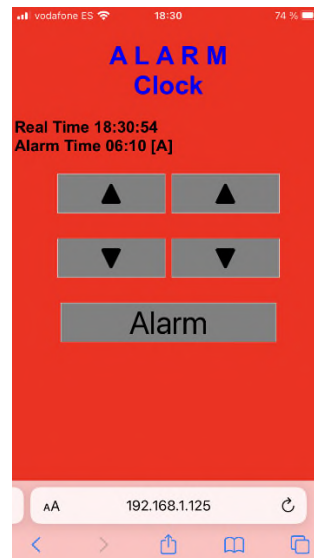
Massimo Banzi: [https://it.wikipedia.org/wiki/Massimo\\_Banzi](https://it.wikipedia.org/wiki/Massimo_Banzi)

Hernando Barragán: [https://en.wikipedia.org/wiki/Hernando\\_Barrag%C3%A1n](https://en.wikipedia.org/wiki/Hernando_Barrag%C3%A1n)

The King of Italy 1002-1015 (Arduin → Arduino): [https://de.wikipedia.org/wiki/Arduin\\_von\\_Ivrea](https://de.wikipedia.org/wiki/Arduin_von_Ivrea)

# Project

## Alarm Clock – Part 3 / 5



Last time we did part 1 and 2. You remember, the project is divided in five steps:

### 1.- WLAN – Basics

Show local IP on display

### 2.- WLAN – Basics & HTML

Show local IP on Webpage (Browser on Smartphone)

### 3.- Internet time

Show actual time on display and HTML

### 4.- Set Alarm Time

Show alarm time on Display and HTML, Modify hh and mm with buttons

### 5.- Realize Alarm Clock Funktion

Implement alarm clock logic, with optical and acoustical alarm

Let's continue with part 3: **Internet time**

## Introduction

### NTP and UDP

Of course there is a central internet server, where you can request the actual time & date. This server is called NTP, means Network Time Protocol. But there is a problem. The request needs some time to get from your ESP through the router of your wifi, through internet to get finally to this server. Then it takes the actual time & date, but the response has the same time delay, until it reaches your ESP. To reduce these time losses, instead of the normal internet protocol (TCP) a simpler, but faster is used: UDP, stand for User Datagram Protocol.

```
#include <NTPClient.h>
#include <WiFiUdp.h>
```

**NTP** = Network Time Protocol (Server that provides time and date information)  
**UDP** = User Datagram Protocol (a Internet protocol like TCP, but simpler and faster)

```
// Define NTP Client to get time
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

// Variables to save date and time
String formattedDate;
String dayStamp;
String timeStamp;
```

You need to setup your client to communicate with the server.

### timeStamp

The result of the request is stored in variable formattedDate, which is the string shown in the box below. With the substring command you cut first for date, then for time. They will be stored in the variable dayStamp and timestamp.

2023-08-06T17:30:57Z

Variable formattedDate contains this String, with the date before T and time between T and Z.

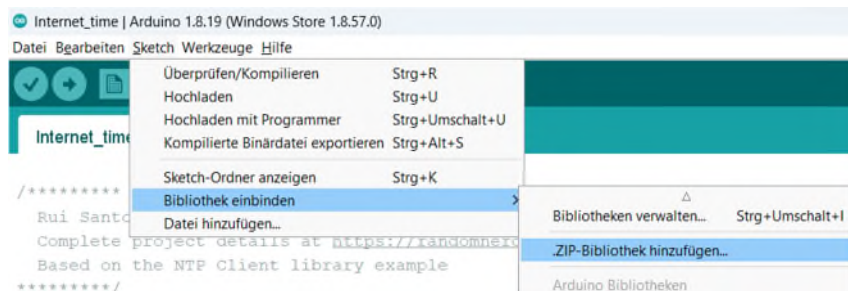
DATE: 2023-08-06  
HOUR: 17:30:57

dayStamp = cut until T  
timestamp = cut after T until Z

## Library

<https://github.com/taranais/NTPClient/archive/master.zip>

C:\Timo\_2022\Sketches\libraries



## Code

```
#include <LiquidCrystal_I2C.h>

// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27, 16, 2);

#include <ESP8266WiFi.h>;
#include <WiFiClient.h>;

#include <ESP8266mDNS.h>

#include <NTPClient.h>
#include <WiFiUdp.h>

// Replace with your network credentials
const char* ssid = "YOUR_SSID";
const char* password = "YOU_PSWD";

WiFiClient client;

#include <ESPAsyncWebServer.h>

AsyncWebServer server(80); // server port 80

void notFound(AsyncWebServerRequest *request)
{
  request->send(404, "text/plain", "Page Not found");
}

// Define NTP Client to get time
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP);

// Variables to save date and time
String formattedDate;
String dayStamp;
String timeStamp;

void setup() {
  // initialize the LCD
  lcd.begin();

  // Turn on the backlight and print a message.
  lcd.backlight();
  lcd.setCursor(0, 0); // Spalte, Zeile
  lcd.print("Connecting");
```

```

    // Connect to WiFi network
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        lcd.print(".");
    }

    // Print local IP on Display
    lcd.setCursor(0, 0); // Spalte, Zeile
    lcd.print("          ");
    lcd.setCursor(0, 0);
    lcd.print(WiFi.localIP());

// Initialize a NTPClient to get time
timeClient.begin();
// Set offset time in seconds to adjust for your timezone, for example:
// GMT +1 = 3600
// GMT +8 = 28800
// GMT -1 = -3600
// GMT 0 = 0
timeClient.setTimeOffset(7200);

}
void loop() {
    while(!timeClient.update()) {
        timeClient.forceUpdate();
    }
    // The formattedDate comes with the following format:
    // 2018-05-28T16:00:13Z
    // We need to extract date and time
    formattedDate = timeClient.getFormattedDate();

    // Extract date
    int splitT = formattedDate.indexOf("T");
    dayStamp = formattedDate.substring(0, splitT);

    // Extract time
    timeStamp = formattedDate.substring(splitT+1, formattedDate.length()-1);

    lcd.setCursor(0, 1);
    lcd.print(timeStamp);
    delay(100);
}

```

### Try yourself:

- 1.) Show instead of time, the date on display.
- 2.) The separators in the date are "-", substitute them by ".".
- 3.) Now change date from yyyy-mm-dd to dd.mm.yyyy style.
- 4.) Then display "dd.mm.yy hh:mm". Between yy and hh use max. 3 spaces.
- 5.) Finally you can blink the ":", means " " when the second is even, and ":" when the second is uneven.

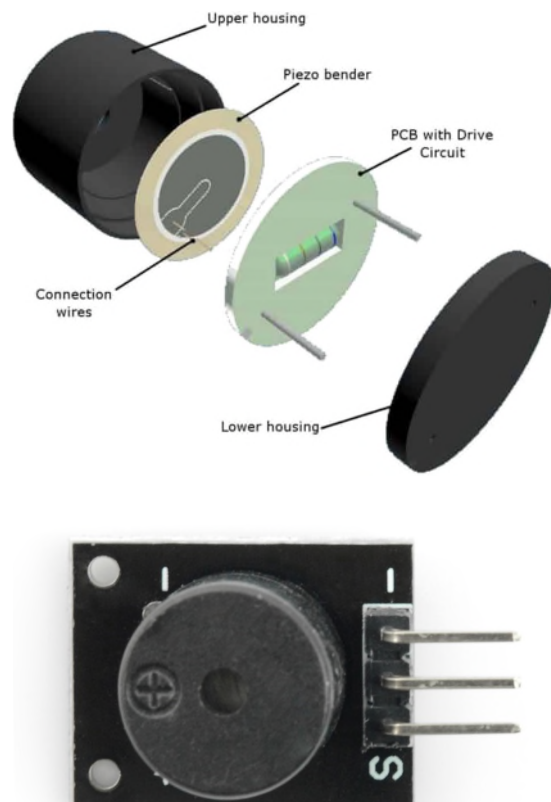
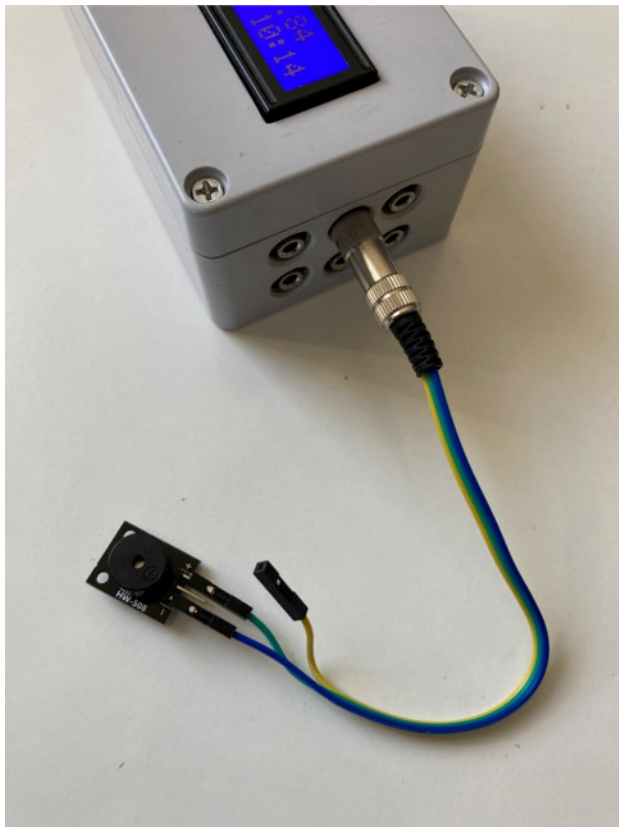


## Hardware

### Description

I use in my projects the KY-006 piezo buzzer. Simply connect – Pin to GND and S-Pin to digital Output.

The piezo will bend while he gets an electrical signal (ON), and go back to original shape, when no electrical signal (OFF) is send. Depending on how rapid or slow the ON-OFF sequence will be generated, we get a higher or lower ton out of the buzzer.



### Source

KY-006: <https://www.amazon.de/AZDelivery-KY-006-Passives-Buzzer-Arduino/dp/B07ZTR24HX/>

### Coding

```
tone(13, 1000); // IO Pin 13, frequency 1000 Hz
delay(1000);    // Wait 1sec

noTone(13);     // tone OFF
```

tone(IO pin, frequency)

noTone(IO pin)

### Try yourself:

- 1.) Try example.
- 2.) Modify code for police car sound (2 different tones alternating: ni-nu-ni-nu...)

## 3D Printing

### Description

From 5 different parts you solve a cube puzzle.



### Source

MyMiniFactory: <https://www.myminifactory.com/object/3d-print-3x3-puzzle-cube-65004>

### Try yourself:

- 1.) Print the standard version and find solution to mount it.
- 2.) Print inner parts in a different color.



Contact:

[nerdsonly.de/kontakt/](https://nerdsonly.de/kontakt/)

## Disclaimer

**Safety Awareness:** The content on this website is intended for educational purposes. Users should exercise caution and follow safety guidelines when engaging in electronic and 3D print projects to avoid potential risks.

**Accuracy Disclaimer:** While we strive for accuracy, the information and projects provided may change and might not always be up-to-date. Users should verify information before relying on it.

**Liability Limitation:** We are not responsible for any injuries, damages, or losses resulting from the use of information, projects, or materials provided on this website. This includes damages related to personal injury, property damage, data loss, or profits.

**External Links:** Links to external websites are not endorsements, and we are not accountable for the content or accuracy of those sites.

**Intellectual Property:** The intellectual property rights of projects and content belong to respective owners. Users can use the information for personal, non-commercial purposes, but commercial use requires explicit permission.