

## Project 4 – LO2 – Technische analyse verslag

In dit project maak ik een systeem waarbij een Arduino kan praten met een webserver. Het doel is om sensordata van de Arduino zichtbaar te maken in een webpagina, én om vanuit diezelfde webpagina opdrachten terug te sturen naar de Arduino.

Om dit werkend te maken gebruik ik: **Arduino**, **Node.js**, **Socket.IO** en **MySQL**.

## **Arduino**

De communicatie met de computer loopt via de Wifi. Alles wat de Node.js doorstuurt, gaat naar de Arduino. Andersom kan ik via diezelfde seriële verbinding commando's terugsturen naar de Node.js.

## **Node.js**

Node.js is de "tussenlaag" in het systeem. Met behulp van het pakket serialport kan ik de seriële data van de Arduino uitlezen. Node.js ontvangt de berichten, verwerkt ze, en stuurt ze daarna door naar de webpagina of naar de database.

Ik gebruik Node.js omdat het snel werkt met real-time data en omdat de combinatie met Socket.IO heel makkelijk is.

## **Socket.IO**

Socket.IO zorgt voor real-time communicatie tussen de browser en de server. Hierdoor worden waardes van de Arduino meteen zichtbaar in de webpagina, zonder dat je hoeft te verversen.

De communicatie gaat ongeveer zo:

1. Arduino stuurt data → Node.js leest dit binnen.
2. Node.js stuurt dit meteen via Socket.IO naar de webbrowser.
3. De gebruiker klikt iets in de browser → Socket.IO stuurt dit terug naar Node.js.
4. Node.js vertaalt dat commando en stuurt het via serial naar de Arduino.

Zo ontstaat er een soort live verbinding tussen hardware en web.

## **MySQL**

MySQL gebruiken we om belangrijke data op te slaan, zoals sensormetingen of logs. Node.js maakt verbinding met de database en zet alle data netjes in tabellen. Later kan ik deze data gebruiken voor grafieken, statistieken of analyses.