# Bike Parking System for S-Trains

Phillip Ravn Boe Jensen
Email: s233478@student.dtu.dk

***Abstract*—The Bike Parking System for S-Trains project introduces a technologically advanced approach to managing bike parking spaces in urban train stations. By integrating networked embedded systems, this project aims to provide real-time data on bike parking availability, facilitating commuter decisions and promoting bicycle usage in city environments.**

## I. Introduction

In the bustling urban landscape of Copenhagen, S-Trains play a vital role in public transportation, featuring specialized bike wagons to accommodate cyclists. Despite this facility, cyclists face a significant challenge in locating available bike spots due to the rapid turnover at stations. This project addresses this challenge by developing a system to provide real-time information about bike parking availability in S-Trains, thereby enhancing commuter experience and promoting the integration of cycling into urban mobility

## II. Background

### *A. Arduino*

Arduino is an open-source electronics platform known for its simplicity and accessibility in hardware and software development. Primarily used for building digital devices and interactive objects that can sense and control physical devices, its ease of use is enhanced by a vast collection of libraries and examples, simplifying the integration of sensors, motors, and other peripherals.[1]

### *B. UART (Universal Asynchronous Receiver/Transmitter)*

UART, or Universal Asynchronous Receiver/Transmitter, is a critical hardware communication protocol used in the field of electronic devices and computing. It facilitates serial communication between devices by transmitting data in a sequential manner, bit by bit. This simplicity and efficiency make UART a popular choice for short-distance, low-speed, low-complexity data exchange between microcontrollers, computers, and peripheral devices. [2]

The fundamental characteristic of UART communication is its asynchronous nature, meaning it doesn't require the transmission and reception of data to be synchronized with a shared clock signal. This feature simplifies the design and reduces the hardware complexity, as only two wires are needed for the entire data transmission—one for sending (TX) and one for receiving (RX).

### *C. AWS Elastic Beanstalk*

AWS Elastic Beanstalk, a service by Amazon Web Services, significantly streamlines the deployment and scaling of web applications. The service's primary strength lies in its automation capabilities, handling tasks such as capacity provisioning, load balancing, auto-scaling, and application health monitoring. This automation allows developers to focus more on writing code rather than managing the underlying infrastructure. [3]

## III. System Architecture

The Bike Parking System is conceptualized as an integrated network of embedded systems, designed to provide real-time data on bike parking availability within trains. This section provides an overview of the system's architecture, focusing on its primary components and their interactions.

### *A. Components*

- **Sensing Units:** Each bike parking section on the train is equipped with sensing units. These units are responsible for detecting the occupancy status of bike spots.
- **Data Aggregators:** Data from the sensing units is collected by data aggregators. These aggregators are situated in each section and serve as intermediaries, compiling and processing sensor data.
- **Master Controller:** The master controller functions as the hub of the system. It receives processed data from the data aggregators and is responsible for orchestrating the overall system functionality.
- **Cloud Server:** A cloud server hosts the data processing and storage services. It also provides a user interface, accessible via web browsers, where commuters can view real-time information about bike spot availability.

### *B. Data Flow and Interaction*

As highlighted by Fig. 1, the system architecture diagram, the hierarchical and networked structure of the system has a natural flow of data:

1) Data flows from the sensing units, the ***Bike-Sensors***, to the ***Data Aggregator***.
2) The ***Master Controller*** receives the aggregated data and communicates with the cloud server.
3) The ***Cloud Server*** processes the data, updates the system's status, and displays it on the user interface.

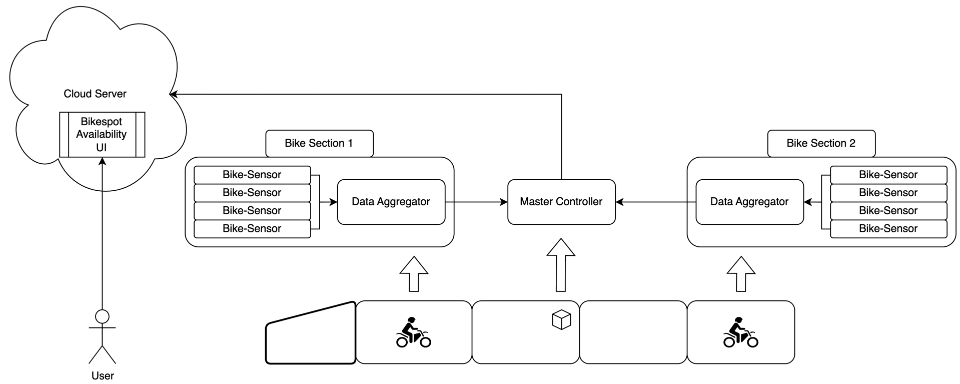


Fig. 1. Conceptual Diagram demonstrating the aspects of the Bike-Parking System, with a S-train below as reference

## IV. Implementation

The implementation of the Bike Parking System, though aligned with the abstract system architecture, differs in scale due to economic constraints. The implemented system demonstrates the core functionalities in a scaled-down model, where each bike section only consists of a single sensor. This section outlines the detailed implementation of the Bike Parking System.

As highlighted by Fig. 2, the implementation consisted of the following key components:

- **Sensing Units (HC-SR05 Ultrasonic Sensors):** For data collection, we utilized HC-SR05 Ultrasonic Sensors. These sensors were chosen primarily due to their cost-effectiveness and ease of use. Their ability to measure distance through ultrasonic waves makes them suitable for detecting the presence or absence of bikes. [4] Although they might be affected by noise and motion in a dynamic train environment, their performance is adequate for a proof-of-concept demonstration.
- **Data Aggregators:** Two Arduino Uno microcontrollers were used, each interfacing with a HC-SR05 sensor. The Arduino Uno was selected for its versatility, ease of programming, and widespread community support. [5] These microcontrollers are capable of handling the sensor data efficiently and are particularly suitable for prototyping IoT applications like ours.
- **Master Controller:** The Arduino Uno R4 WIFI acts as the central node for aggregating data from the Arduino Uno units. Due to its integrated WIFI capability, since it possesses a ESP8266 module, made it the ideal microcontroller for easily setting up and transmitting aggregated sensor data to the cloud server. [6]
- **Cloud Server:** The choice of AWS Elastic Beanstalk for hosting the cloud server stems from its scalability, reliability, and ease of deployment. This platform enables streamlined management of the web application, ensuring that the system can be scaled up for broader implementation without significant changes to the underlying architecture. [7]

## V. Experiment

### A. Methodology

The response time was measured from the moment a bike was simulated to be parked (or removed) in front of a sensor to the time when the status update was reflected on the user interface. A stopwatch was used to measure this duration. The test was repeated 10 times for each sensor to ensure accuracy and consistency in the measurements.

### B. Experiment Setup

The experiment was conducted in a controlled environment simulating a bike section of an S-Train. Two HC-SR05 Ultrasonic Sensors were installed to mimic real-life conditions of bike parking. These sensors were connected to the Arduino Uno microcontrollers, which in turn were linked to the Master Controller. The Master Controller was programmed to transmit data to the cloud server hosted on AWS Elastic Beanstalk upon detection of a bike's presence or absence. The server's user interface was monitored to record the time taken for the status update.

### C. Results

The following table presents the response times recorded during the experiment:

| Test Number | Response Time (seconds) |
|---|---|
| 1 | 1.2 |
| 2 | 1.1 |
| 3 | 1.2 |
| 4 | 1.3 |
| 5 | 0.9 |
| 6 | 1.4 |
| 7 | 1.3 |
| 8 | 1.1 |
| 9 | 1.3 |
| 10 | 1.1 |

TABLE I
Response times for bike detection and UI update

The data indicates that the response time ranged between 0.9 and 1.4 seconds.

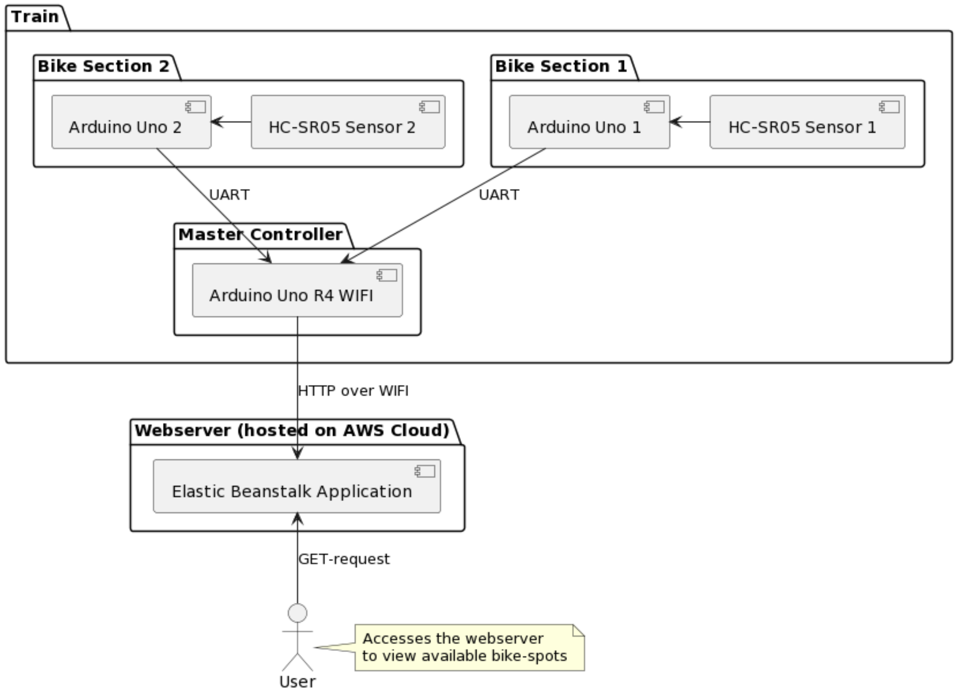


Fig. 2. Component Diagram of Bike-Parking System: The HC-SR05 sensors, connected to the Arduino Uno units, detected the presence or absence of bikes in their respective sections. The Arduino Uno units, collected the sensor data and transmitted it to the master controller via serial connection. The Arduino Uno R4 WIFI, as the master controller, received the aggregated data and relayed it to the cloud server through HTTP GET requests.

### *D. Analysis*

The results indicate that the Bike Parking System updates the user interface with a new bike parking status within an average of 1.19 seconds. This rapid response time is crucial for providing real-time information to commuters waiting at train stations. It allows them to make informed decisions about where to find available bike spots ahead of the train's arrival. However, further optimization might be necessary to account for potential delays due to network congestion or server performance under high load conditions.

## VI. Future Works

### *A. Advanced Sensing Technologies*

The current iteration of the Bike Parking System for S-Trains employs HC-SR05 Ultrasonic Sensors, which, while effective for a proof-of-concept, possess limitations in terms of accuracy and environmental adaptability. Future advancements in this project could explore the integration of advanced sensing technologies. The utilization of pressure sensors, for instance, could significantly enhance the system's accuracy in detecting bike occupancy. These sensors are less susceptible to environmental variables such as noise and movement, which are common in train settings. Furthermore, the incorporation of optical sensors could provide additional layers of data, enabling the system to differentiate between different types of bikes or to detect obstructions in the parking area. The amalgamation of these technologies would not only refine the system's operational efficacy but also pave the way for a more intelligent reliable bike parking solution.

## VII. Concluding Remarks

The development of the Bike Parking System for S-Trains signifies an important stride towards enhancing the synergy between cycling and public transportation in urban landscapes. By harnessing the power of technology to provide real-time insights into bike parking availability, the project directly addresses a key challenge for cyclists using S-Trains. Although the current implementation showcases the system's core capabilities, the future enhancements discussed promise to elevate its functionality, making it an indispensable tool in the pursuit of sustainable and efficient urban mobility. The envisioned advancements in sensing technologies and scalability are not just incremental improvements but are pivotal steps towards realizing a fully integrated, cyclist-friendly urban transportation ecosystem.

## References

[1] Arduino. "Arduino documentation - "what is arduino?"." Accessed: 28/11/23. (2018-02), [Online]. Available: https://www.arduino.cc/en/Guide/Introduction.

[2] Rohde & Schwarz, *Understanding uart*, https://www.rohde-schwarz.com/us/products/test-and-measurement/essentials-test-equipment/digital-oscilloscopes/understanding-uart_254524.html#gallery-7, Accessed: 2023-11-11.

[3] Amazon Web Services, *Aws elastic beanstalk*, https://aws.amazon.com/elasticbeanstalk/, Accessed: 2023-11-11.

[4] A. Name, "Study and analysis of an ultrasonic sensor for distance measurement," *International Journal of Research in Engineering, Science and Management*, vol. 3, no. 5, pp. XX–XX, 2020-05. DOI: XX. [Online]. Available: https://www.ijresm.com/Vol.3_2020/Vol3_Iss5_May20/IJRESM_V3_I5_15.pdf.

[5] A. Name, "An efficient approach for sensor data collection and communication in iot using arduino and mqtt protocol," *Sensors*, vol. 22, no. 3, p. 713, 2022. DOI: 10.3390/s22030713. [Online]. Available: https://www.mdpi.com/1424-8220/22/3/713.

[6] Arduino, *Arduino uno r4 wifi*, https://docs.arduino.cc/hardware/uno-r4-wifi, Accessed: 2023-11-11.

[7] Replo, *Replo cloud-based platform*, https://www.replo.app/cloud-based-platform, Accessed: 2023-11-11.