DIGITAL TWIN MONITORING SYSTEM

By

Hanchi Ge

Kowshik Dey

Rongjian Chen

Final Report for ECE 445, Senior Design, Spring 2024

TA: Xinlong Huang

10 May 2024

Project No. 43

Abstract

Digital twin is a technology that connects the physical and digital realms. By creating a digital copy of the physical system, the technology can monitor, analyze, and predict the performance of the system in different scenarios in real time. The digital twin is constantly updated with data from sensors and other relevant sources, ensuring that it accurately reflects the current state of its physical counterpart. This paper describes our work on creating a digital twin analog bridge equipped with strain gauges to measure the weight of passing vehicles, which also incorporates real-time data and a traffic light-based alert system. This study elucidates new design methods for building physical models, as well as digital twins, and then integrating machine learning methods into the hardware.

Contents

1. Introduction

The focus of our project, the Digital Twin Monitoring System, is to develop a dynamic solution that bridges the gap between the physical and digital model through a real-time, data-driven replica of the physical model bridge. This system is designed to enhance our ability to monitor, analyze and predict the structural integrity and traffic efficiency on the bridge using a suite of integrated measurements and machine learning algorithms. The process of implementing a damage identification strategy for aerospace, civil and mechanical engineering infrastructure is referred to as structural health monitoring (SHM).[1] The development of strain gauges is also sufficient to prove its effectiveness. For example, the flexible skin Strain gauge in 2012[2], and the Silicon-on-Polymer Strain Gauges in 2013.[3]

The primary challenge of our project is the evident need for a more sophisticated, real-time monitoring system that can predict and alert for structural and traffic-related concerns on the bridge. This is especially critical for the prevention of accidents in the highway and ensuring timely maintenance resulting in saving thousands of dollars. Our proposed solution involves the use of digital twin simulation coupled with a traffic light alert system based on the weight of the crossing vehicles.

The Digital Twin Monitoring System is comprised of multiple subsystems, including the strain gauge module for mechanical stress measurement, a data processing controller, and a communication system for real-time data transmission and notifications. These subsystems work in conjunction to facilitate the ongoing monitoring and evaluation of the bridge's structural integrity and traffic patterns. Here we provide a brief overview of how each of the component contributes to the overarching goal of the project and how they are interconnected as follows:

- **Strain Gauge Module:** This component measures the mechanical strain on the bridge, providing data that is critical for accessing structural integrity. It is connected to the Wheatstone bridge circuit for accurate measurement conversion.
- **Controller:** The controller module comprises of a STM32 processor that uses the data from the strain gauge at critical points and other sensors, using a machine learning model (SVM) to predict the structural failures and to manage the traffic light system.
- **Traffic Light System:** Receives control signals from the controller to manage traffic flow based on the bridge's current load capacity.
- **Communication System:** This module ensures real-time data transfer between the bridge and the monitoring station/phones via Bluetooth module, facilitating immediate responses to detected issues.
- **Voltage Regulator:** Controls the voltage input to the strain gauge and all the other modules.

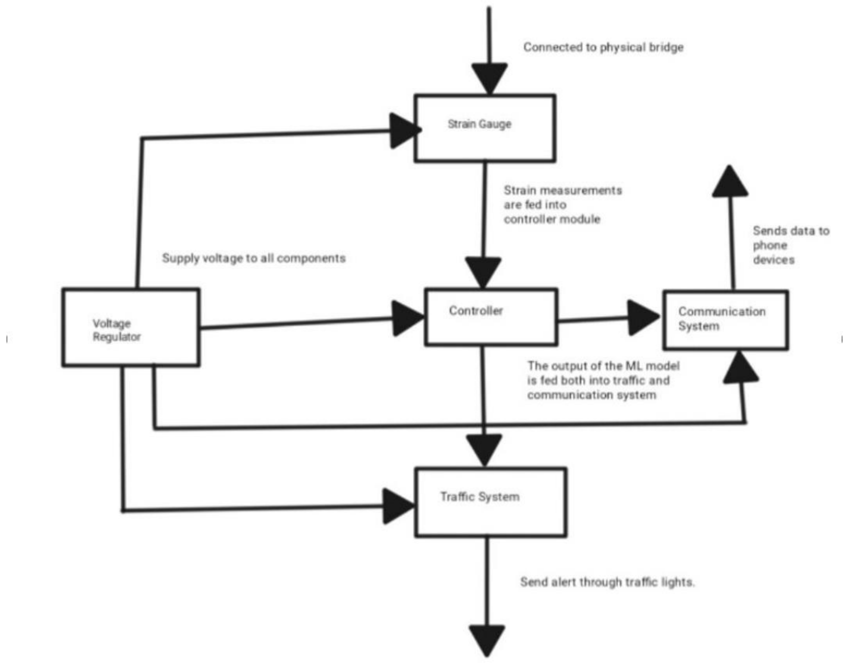


Figure 1. Top level diagram

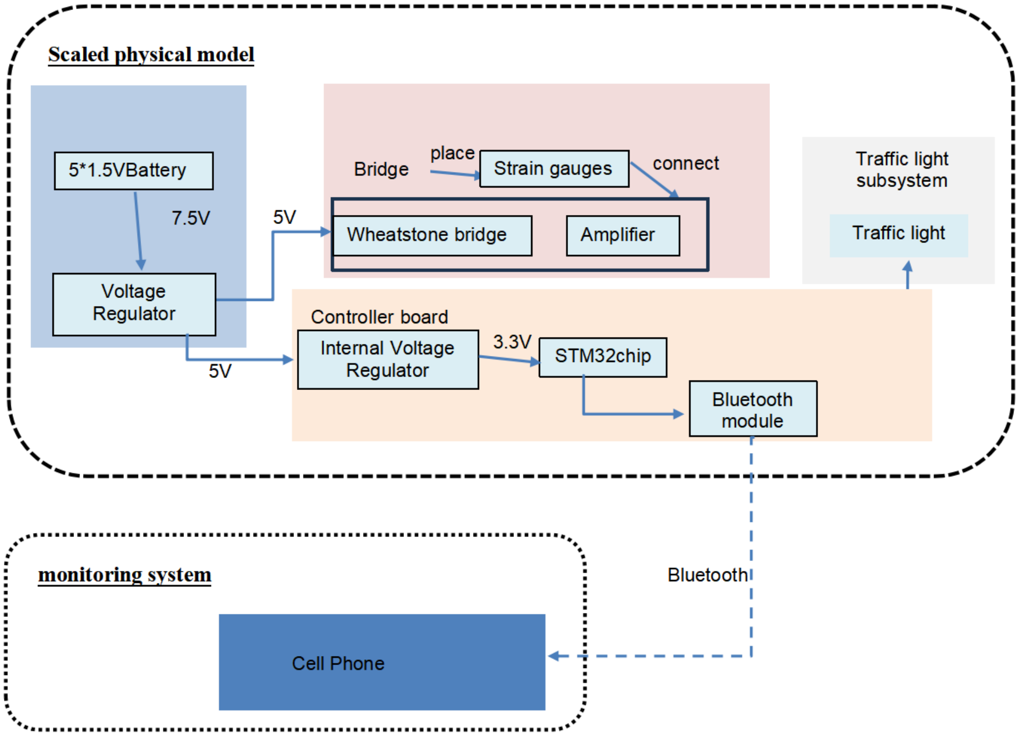


Figure 2. Block diagram

2 Design

Our system is a combination of subsystems. A key part of our system is the strain gauge module, which is designed to detect and measure mechanical strain with a precisely calibrated Wyeth bridge and amplified output to improve signal clarity. This input is fed into a control module powered by a dedicated voltage regulator to ensure the stable operation of the subsystem. The control module uses STM32F103C8T6 microcontroller to manage the data of strain gauge and execute the control command. Wireless communication via Bluetooth interface allows real-time data transfer to monitor data. The system's functionality uses machine learning to predict bridge structural safety. Below, we will delve into the details of each module, highlighting their individual contributions to the overall functionality of the system.

2.1 Strain gauge module

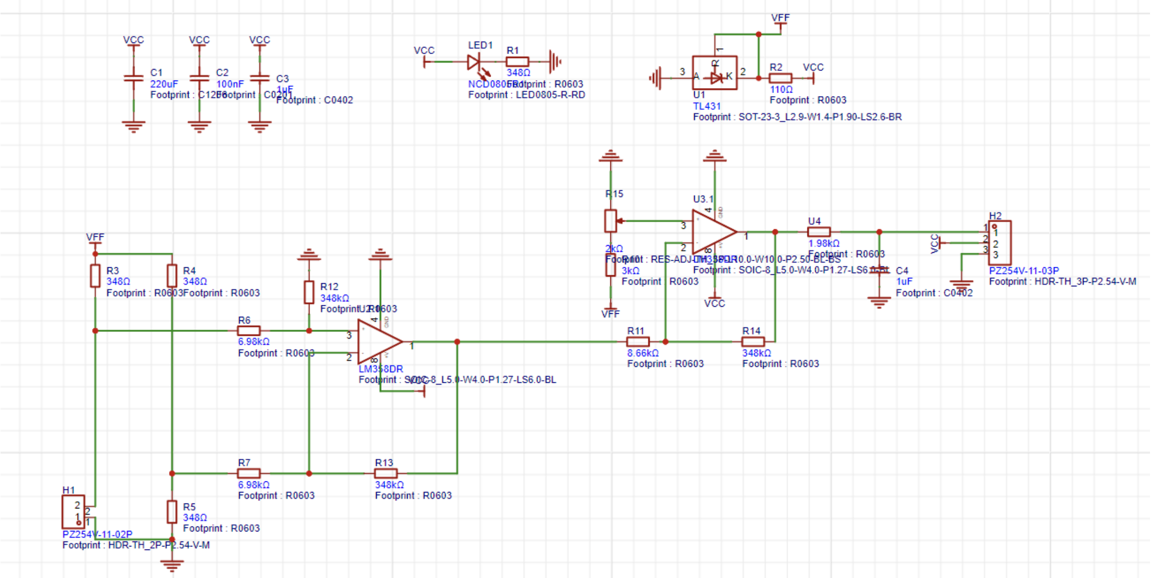


Figure3. Schematic of strain gauge module

2.1.1 Strain gauge

2.1.1.1Working principle

A strain gauge is a sensor whose resistance varies with applied force. It converts force, pressure, tension, weight, etc., into a change in electrical resistance which can then be measured. In the schematic, H1 connects to an external strain gauge. When attached to an object, the strain gauge will deform when the object is subjected to stress, and this deformation will change the electrical resistance of the gauge.

The gauge is attached to the object by a suitable adhesive, such as cyanoacrylate.[4] As the object is deformed, the foil is deformed, causing its electrical resistance to change. The strain gauge modules are critical components in the bridge monitoring system, designed to measure the deformation of the bridge structure under load. This section provides a detailed analysis of the design and operation of the strain gauge modules, including the principles behind their operation, the design criteria, and the specific choices made to optimize their performance.

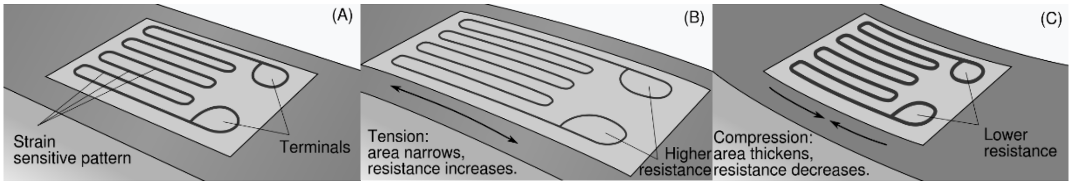


Figure4. Resistance of strain gauge

$$R = \rho\frac{L}{A} \tag{1}$$

$$dR = \frac{\rho}{A}dL - \rho\frac{LdA}{A^2} + \frac{L}{A}d\rho$$

By (1)

$$\frac{dR}{R} = \frac{dL}{L} - \frac{dA}{A} + \frac{d\rho}{\rho}$$

But

$$\frac{dA}{A} =- 2\sigma\frac{dL}{L}$$

$$\frac{dR}{R} = (1 + 2\sigma)\frac{dL}{L} + \frac{d\rho}{\rho}$$

$$gauge\ factor = G = \frac{dR/R}{dL/L} = 1 + 2\sigma + \frac{d\rho/\rho}{dL/L}$$

Foil strain gauges are widely used in various applications, and different applications have different requirements for strain gauges. In most cases, the durability of the strain gauge is very important. For example, strain gauges attached to a force sensor often need to remain stable for years or even decades, while those used for dynamic experiments may only need to be attached for a few days. In our experiment, the strain gauge was glued to the back of the bridge to ensure its stability over a relatively long period of time. However, due to the selection of non-professional glue, there may be errors within the allowable range of experimental data.

2.1.1.2 Ideal position

In our system, two strain gauges are adhered to the lower surface of the aluminum plate to monitor the deformation of the plate when a remote-controlled car with varying numbers of weights drives onto our bridge model. The more weights the remote-controlled car carries, the more pronounced the deformation of the aluminum plate. Consequently, the strain gauges affixed tightly to the lower surface of the aluminum plate deform along with the plate, causing a corresponding change in the output voltage of the strain gauges. This allows us to determine the degree of deformation of the bridge by processing the output voltage signals of the strain gauges. However, the degree of deformation of the bridge model is not uniform across every location on the aluminum plate. The most ideal position to affix the strain gauges is where the strain gauges' deformation and the corresponding output voltage do not exceed their threshold precisely at the point of maximum deformation of the bridge. This approach enables us to fully utilize the output range of the strain gauges, thereby minimizing the sensitivity requirements of the signal processing system.

Having outlined the logical approach to determining the position of the strain gauges, the next step is to calculate this ideal position. In the calculations, the bridge model is simplified by treating the two small stools supporting it as triangular support frames in the longitudinal direction of the aluminum plate. Additionally, the pressure exerted by the remote-controlled car on the aluminum plate is also idealized as concentrated at a single point. Consequently, all forces involved in the model calculations are simplified as forces concentrated at a single point. Furthermore, the centripetal acceleration generated by the bending deformation of the bridge is temporarily disregarded when the remote-controlled car traverses it. Based on this idealized model, the relationship between deflection and pressure and position indicates that the maximum deflection of the bridge model is 1/84 times the pressure exerted by the remote-controlled car on the aluminum plate, all in international standard units. Furthermore, according to the relationship between the deflections at various points on the bridge obtained from experiments using a high-speed camera when the remote-controlled car with the maximum number of weights traverses the bridge, the ideal installation position of the strain gauges is approximately 0.1723 m to the left or right of the midpoint between the two stools. Subsequently, we attempt to incorporate the centripetal acceleration generated by the remote-controlled car traversing the bridge model. This involves adding the centripetal force provided by the aluminum plate to the total gravitational force exerted by the remote-controlled car and weights. The centripetal force is calculated using the speed of the remote-controlled car measured by the high-speed camera and the radius of the arc formed by the bridge in the ideal model. The remaining calculation process is similar to that without considering centripetal force. The final result indicates that the ideal position is approximately 0.2297 m to the left or right of the midpoint between the two stools.

2.1.2 Wheatstone bridge

$$e_0 = \frac{R_3 R_5 - R_s R_4}{(R_3 + R_s)(R_5 + R_4)} \times VFF$$

A Wheatstone bridge is a circuit used to accurately measure resistance. In the application of strain gauges, Bridges help detect small changes in resistance. The voltage on the bridge changes as the resistance of the strain gauge changes, which occurs when strain is applied to the gauge.

To explain in more detail, each strain gauge is integrated into the Wheatstone bridge circuit to accurately measure the small resistance changes caused by strain. Wheatstone bridge is a balanced bridge circuit that converts resistance changes into a measurable voltage output. When the strain causes the resistance to change, the bridge is unbalanced, generating a voltage signal proportional to the resistance change, to achieve accurate measurement of the strain. This bridge structure greatly improves measurement accuracy and sensitivity and is suitable for a variety of strain measurement applications. The Wheatstone bridge structure used in this project is shown, which specifically shows the connection mode and function of strain gauge in the bridge. With this configuration, we can effectively capture and scale strain signals, ensuring the accuracy and reliability of measurement results.

2.1.3 Amplifier

An operational amplifier is used in the circuit to amplify the tiny voltage signal output by Wheatstone bridge. The output voltage of the Wheatstone bridge is very small, and sufficient gain from the op amp is needed to raise the signal level to make it suitable for subsequent analog-to-digital conversion or data acquisition. Operational amplifiers ensure accuracy and stability during signal amplification while minimizing noise and distortion. By using operational amplifiers, the weak bridge output signal can be amplified reliably, and high precision measurement and control can be achieved. How it works: A strain gauge is a sensor used to measure the strain of an object. They work on the principle that the resistance of the gauge is proportional to the amount of deformation of the object to which it is attached. This change in resistance is so small that it usually requires an accurate measuring system to detect it accurately.

2.2 Outside Voltage Regulator

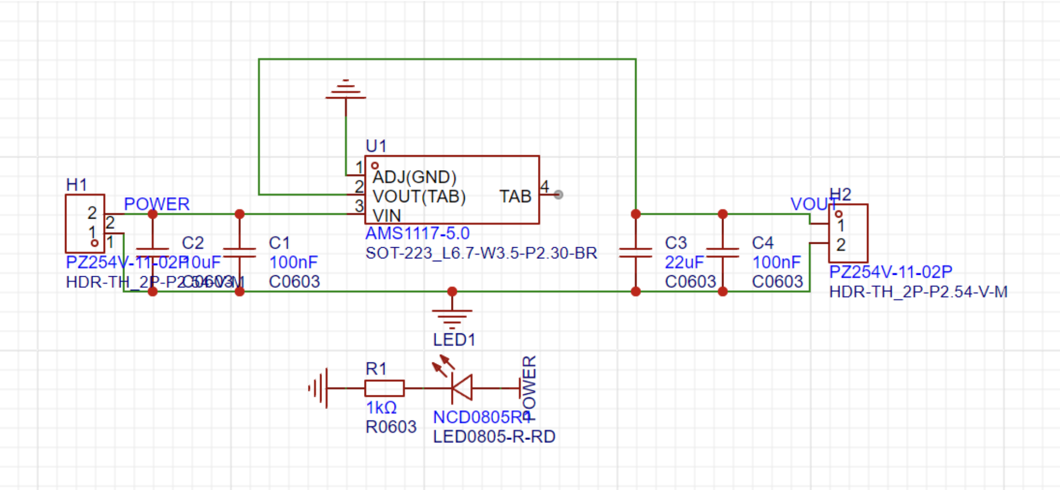


Figure5. Schematic of Voltage regulator

The power supply subsystem is the key to provide stable and reliable voltage for the various components of the Bridge Monitoring System. The subsystem consists of a battery pack, a voltage regulator circuit, and an internal voltage regulator circuit in the controller board. I will now introduce them separately.

Battery pack: The power supply of the system is a battery pack formed by 5 1.5 V batteries in series, with a total voltage of 7.5 V. This voltage is sufficient to power the entire system, but it must be adjusted to ensure the stable operation of sensitive electronics. It is important to note that battery capacity is a big limiting factor. When the energy stored in the battery is gradually exhausted, the input voltage will be less than 6v, and the energy supply subsystem will affect the operation of the entire system. Therefore, even replacing the battery is an important operation to protect the stability of the system.

Voltage regulator circuit: To provide a stable 5 V output, which is required for some parts of the Wheatstone bridge and controller board, a voltage regulator circuit is used. Our regulator of choice is AMS1117-5.0, a low-difference (LDO) linear regulator. AMS1117, with extremely low static current, varies with load, improving efficiency. The AMS1117 is a low leakage voltage regulator whose voltage control tube consists of a PNP driver NPN transistor defined as a leakage voltage.[5] The AMS1117 is available in fixed and adjustable versions with output voltages of 1.2 V, 1.5 V, 1.8 V, 2.5 V, 2.85 V, 3.0 V, 3.3 V and 5.0 V respectively. To ensure the stability of the adjustable voltage version of the AMS1117, the output needs to be connected to a capacitor of at least 22 μF. In this project, the external voltage regulator module adopts AMS1117 chip with 5 V output. This regulator was chosen because of its ability to maintain a constant 5V output when the input voltage changes slightly due to battery discharge. It includes an input capacitor (C1), which is a 100 μF capacitor placed at the input to stabilize the input voltage and filter out any high-frequency noise. The output capacitor (C2) has a 22 μF capacitor at the output to ensure a stable 5 V output and filter out any noise generated by the regulator itself. In addition, an LED indicator (LED1) indicates when the regulator is operating.

2.3 Controller

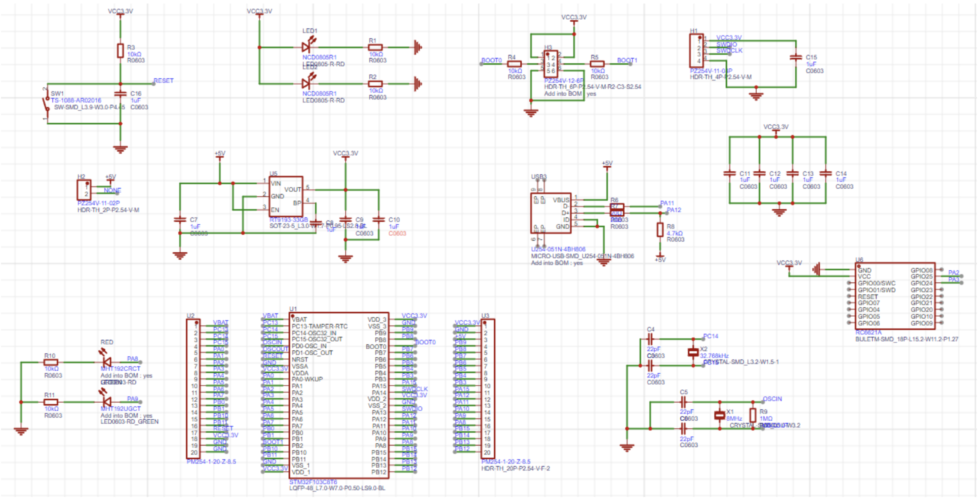


Figure6. Schematic of Controller

The controller board subsystem is a crucial part of the bridge monitoring system, responsible for processing the data collected from the strain gauges, running the machine learning algorithms, and managing communication with external devices. This section provides a detailed analysis of the design and operation of the controller board, including its components, circuitry, and the reasons behind the design choices. The core of the controller board is the STM32F103C8T6 microcontroller, chosen for its balance of performance, peripheral support, and cost. This microcontroller reads the analog signals from the strain gauges, processes these signals, applies the Support Vector Machine (SVM) algorithm to predict the structural safety of the bridge, and communicates the results via Bluetooth.

2.3.1 stm32f103c8t6

The STM32F103C8T6 microcontroller has a built-in analog/digital converter (ADC) capable of converting the 5V input voltage to the 3.3 V required for the microcontroller. The Bluetooth low power module is used for wireless communication, while the traffic light indicator consists of LEDs that provide a visual indication of the status of the bridge structure. The STM32F103C8T6 microcontroller features a 32-bit ARM Cortex-M3 core with enough computing power to handle the data processing and tasks required for this project. For analog/digital converters (ADCs) and data processing, the STM32F103C8T6 microcontroller integrates two 12-bit ADCs, each with 16 external channels, for single or continuous conversion.[6] In scan mode, it can automatically convert a selected set of analog inputs, simplifying circuit design without the need for an external ADC. The data collected by the ADC is transmitted to a predefined DMAT0buf, which takes an average from the DMA interrupt and puts it into the data. After the conversion is complete, the result is in the data register of the appropriate ADC converter. If the data is not captured in time, it will be overwritten by the new conversion results of the ADC converter. The 12-bit ADC provides high-resolution data conversion to convert the analog signal generated by the strain gauge into a digital signal, ensuring accurate strain measurement.

2.3.2 Bluetooth RC6621A

The new Bluetooth Low Power (BLE) standard was published by the Bluetooth Technology Alliance in 2010.[7] The Bluetooth communication subsystem is an important part of the bridge monitoring system, enabling real-time wireless data transmission between the microcontroller and external devices such as smartphones or computers running MATLAB. The main component of this subsystem is the Bluetooth Low Power (BLE) module, which was chosen for its low power consumption, reliable communication range, and easy integration with the STM32F103C8T6 microcontroller.

The BLE module operates at 3.3 V, which meets the voltage requirements of the single chip computer, thus simplifying the power supply design. The microcontroller communicates with the BLE module through a universal Asynchronous Transceiver (UART) interface. Set the baud rate of the UART interface to 9600bps, 8 data bits, 1 stop bit, and no parity check.

The flow of data in a subsystem consists of several steps. Firstly, STM32 microcontroller collects strain data from strain gauge and uses embedded SVM algorithm to process it to determine the structural state of the bridge. This processed data, including strain measurements and classification results, is then formatted into a string message suitable for transmission. The formatted data is sent to the BLE module via the UART interface and then transmitted wirelessly to the connected external device. External devices running MATLAB connect to the BLE module and receive the transmitted data. MATLAB is configured to interpret incoming data streams, allowing for real-time visualization and analysis. This setup enables continuous data monitoring, ensuring that any changes in the status of the bridge structure are detected and displayed in a timely manner.

To sum up, the Bluetooth communication subsystem realizes effective and reliable real-time data transmission through its BLE module and well-configured UART interface. This allows continuous monitoring and immediate response to structural health problems, enhancing the overall function and practicality of the bridge monitoring system. The integration of BLE module and MCU ensures efficient data transmission, while the real-time data visualization in MATLAB provides immediate and intuitive feedback on the state of the bridge structure, significantly improving the overall performance of the system.

2.3.3 5v-3.3v voltage regulator

The voltage regulator circuit is designed to convert a 5 V power supply to 3.3 V, which is required for many components on the board, including the STM32F103C8T6 and Bluetooth modules. The regulator ensures that microcontrollers and other sensitive electronic components receive a stable power supply, providing the power necessary for reliable operation. The use of such regulators is essential to match the voltage requirements of various components to ensure module life and reliability.

2.3.4 Crystal oscillator circuit

The crystal oscillator circuit provides the precise clock signal for the microcontroller. It consists of two quartz crystals and four capacitors that help stabilize the oscillation frequency. This clock is critical for a microcontroller because it determines the speed at which it operates and ensures timing accuracy for tasks such as communication protocols, sensor readings, and time-related calculations. The correct and stable operation of the crystal oscillator is the basis of the reliable performance of the whole controller system.

2.3.5 Traffic light

In the circuit, the PA8 and PA9 pins of the stm32f103c8t6 control the LED red and LED green respectively. The light goes green when our program determines that the bridge structure is safe. When a heavy vehicle passes by, causing the bridge to deform too much, the red light goes on and stays on for five seconds.

3. Design Verification

To ensure the reliability of our project design, a validation process is implemented in all modules. This process includes functional testing under laboratory conditions and simulation scenarios. From the strain gauge module to the regulator and controller, each component is tested to ensure it meets operational requirements. These tests are essential to confirm the reliability of our systems. Below, we will detail the specific validation process and results for each subsystem.

3.1 Strain gauge module

To verify the strain gauge module, we first built a prototype on the PCB, and after soldering, made sure that all the connections were secure. We try forward and reverse bending strain gauges to provide stress. Under ideal conditions, $Vout = G \times \ (\epsilon \times S) \times e_0$, V is the output voltage, G is the multiplier, and S is the sensitivity. In the actual case, we only need to verify that the output signal of the strain gauge module always becomes larger or smaller when increasing stress is provided in one direction. We tested 126 types of ADC values passed from the strain gauge module to the MCU and then sent by the Bluetooth module when six different weight carts passed by. And we have selected some of them for display.

Table 1

ADCvalue in STM32	Car (green)	Car with 50 grams (green)	Car with 150 grams(green)	Car with 250 grams(green)	Car with 350 grams(red)	Car with 450 grams(red)
Right of center	(2702,1853)	(2671,1696)	(2681,1630)	(2668,1568)	(2654,1512)	(2674,1472)
Center	(2700,1512)	(2610,1679)	(2562,1646)	(2557,1538)	(2528,1471)	(2340,1392)
Left of center	(2398,1888)	(2430,1886)	(2400,1871)	(2282,1870)	(2336,1870)	(2144,1886)

In fact, in the test, because we cannot confirm how much stress is, we can only judge the size of the stress by the degree of bending, which is a deficiency in the test.

3.2 Voltage regulator

This is a relatively simple module. In the laboratory, we connected the generator to H1, provided input voltage, varied the input voltage within the range of 6.5-12 V, and tested that the output voltage was always 5V, confirming the success of the test.

In this test, we found that when the input voltage was 5-6 V, the output voltage did not work properly, so we chose 6.5-12 V as the test range.

3.3 Controller

When all PCBs are made, all components are soldered and properly connected, we begin verification.

Traffic light: When a remote-controlled car drove over our model bridge, the structure was safe, and the green light remained on. Gradually increase the weight loaded on the car until the red light comes on and remains off for five seconds after the car leaves, and the green light continues to come on after that. It's the same thing as the weight goes up.

Bluetooth: The TX of stm32 is connected with the RX of Bluetooth, and the RX of stm32 is connected with the TX of Bluetooth to complete asynchronous communication and realize serial communication. On the Bluetooth app of the mobile phone, the signal of the two strain gauge modules and the LED level signal can be received and shown. The test was successful.

4. Costs

4.1 Parts

Table 2 Parts Costs

Part	Manufacturer	Retail Cost ($)	Bulk Purchase Cost ($)	Actual Cost ($)
Aluminum Beam (Taobao)	Shengjili	118.8 yuan	118.8 yuan	118.8 yuan
LED (Taobao)	Zave	2.68 yuan	2.68 yuan	2.68 yuan
Strain Gauge (Taobao, BFH120-5AA)	Guantal Electronics	60 yuan	30 yuan	60 yuan
Data acquisition board	Lexus Electronics	339 yuan	339 yuan	339 yuan
Battery Case	Risym	2.76 yuan	2.76 yuan	2.76 yuan
PCB Board	Jiali Chuang Electronics	46.8 yuan	46.8 yuan	46.8 yuan
Strain Gauge Module	Jiali Chuang Electronics	46 yuan	46 yuan	46 yuan
Cntroller	Jiali Chuang Electronics	46 yuan	46 yuan	46 yuan
Weights	Yanheng	5.3 yuan	5.3 yuan	53 yuan
Total	**/**	751.94 yuan	721.94 yuan	799.64 yuan

4.2 Labor

Labor	Rongjian Chen	113630.1 yuan / year[8]	130 hours	1686.3 yuan
	Hanchi Ge	113630.1 yuan / year	130 hours	1686.3 yuan
	Kowshik Dey	113630.1 yuan / year	130 hours	1686.3 yuan

5. Conclusion

5.1 Accomplishments

The strain gauge module, built with a precisely calibrated Wheatstone bridge and dual LM324DR amplifiers, successfully converted mechanical strain into measurable electrical signals. This module reliably detected both small and significant changes in strain, thanks to its enhanced signal clarity and sensitivity. The module was tested under various stress conditions, confirming its capability to consistently output proportional changes in voltage with increasing stress, verifying the fundamental relationship.

Our custom-designed voltage regulator module, centered around the AMS1117-5.0 LDO regulator, successfully provided a stable 5 V output across a range of input voltages (6.5-12 V). This stability was crucial for powering the microcontroller and other sensitive components without interruption.

The STM32F103C8T6 microcontroller acted as the brain of our operation, effectively managing data from the strain gauge and executing control commands for real-time traffic light signaling based on structural data.

The Bluetooth module (RC6621A) enabled seamless wireless communication, allowing for the transmission of structural data and system alerts to mobile devices, facilitating remote monitoring and data analysis.

Traffic Light System successfully responded to varying loads on the model bridge. A remote-controlled car simulated different weights, triggering appropriate traffic light responses (green to red) as structural limits were approached and surpassed.

Using a support vector machine in Python, we developed a predictive model that learned from the strain gauge data to make decisions. This model was simplified and integrated into the microcontroller via C programming, demonstrating the successful blending of data analytics with embedded system responses.

The embedded system's capability to process and react to data in real-time was showcased through continuous monitoring and adaptive signal control based on the analyzed inputs from the structural sensors.

5.2 Uncertainties

Over the course of our project, while most of the results met expectations, there were also areas of uncertainty and where the results did not exactly match our original projections. Here, we provide a quantitative discussion of these unsatisfactory results, clarify the problems encountered, and support our interpretation with evidence from our testing and analysis.

Low voltage regulator performance: When the input voltage drops to the lower end of 6 V, the AMS1117-5.0 regulator does not achieve optimal performance. This is evidenced by the deviation of the output voltage, which is lower than the expected 5V threshold and can be as low as 4.7 V.

After analysis, it was observed during the test that when the input voltage was set to 5 V, the stability of the output voltage decreased, and the standard deviation of the output voltage was ±0.3 V, while the standard deviation of the output voltage under normal working conditions (6.5-12 V) was ±0.05 V. This also means that the low difference (LDO) characteristic of the regulator assumes very little difference between the input and output voltages; However, due to inherent inefficiency and a drop voltage characteristic of about 1.2 V at full load, it is difficult for the regulator to maintain a stable output when the input is close to the output voltage.

Sensitivity of strain gauges to environmental changes: Strain gauges exhibit sensitivity to environmental changes, particularly temperature fluctuations, which were not fully anticipated at the design stage. This results in slightly inaccurate strain measurements, with the resistance of the strain gauge changing by about 0.2% for every 1 °C change in temperature, a sensitivity that can result in a strain reading error of up to 5% but does not have a large impact on the experiment.

5.3 Ethical considerations

Collecting real-time data on vehicles crossing the bridge could potentially trigger privacy concerns, particularly if the gathered information includes identifiable details about individuals or vehicles. This might contravene the principle of upholding privacy as delineated in the IEEE Code of Ethic[9], which underscores the importance of safeguarding individuals' privacy and confidentiality. To address this, it's imperative to anonymize and aggregate the collected data to prevent the identification of individuals or specific vehicles. Additionally, stringent access controls and encryption protocols should be implemented to protect the integrity of the collected data.

The development process must prioritize transparency regarding the system's capabilities and constraints. It is essential to establish clear accountability for any decisions derived from the collected data. This commitment resonates with the IEEE Code of Ethics, underscoring the significance of honesty and integrity in professional endeavors. It is imperative to comprehensively document the development process, encompassing the algorithms employed for data analysis and the criteria guiding alert generation. Stakeholders should receive lucid explanations about the system's functionality and its potential ramifications.

When assessing the force dynamics of the physical model of the bridge, an idealized model is employed, which simplifies by neglecting the impact of the self-weight of the steel plates situated both within and outside the piers serving as supports. This oversight extends to the deflection of the bridge. Additionally, the model fails to consider the width of the piers in relation to the bridge's longitudinal direction, contributing to discrepancies between simulated bridge data and results from actual experiments. To rectify this, experimental statistics are utilized to amend the relationship between the idealized and actual models. This correction is based on the average deviation between the outcomes derived from the ideal model and those observed. Notably, while the maximum transverse sheer force of aluminum 7075 significantly exceeds the maximum shear force calculated via the ideal model, this discrepancy in transverse shear force does not adversely impact the experimental findings.

Since the maximum weight limit for triggering the system to display a red light is set with respect to the next bridge, and routes between bridges typically aren't singular, this could potentially trigger unnecessary alerts.

5.4 Future work

For the future development of this project, we have several ideas for improvement.

One potential improvement is to expand the range of sensor types to include environmental types such as temperature, wind, and humidity, which would provide a more comprehensive understanding of the factors that affect the structural health of Bridges.

In addition, integrating advanced machine learning algorithms can improve the accuracy of predicting potential structural failures, so that proactive maintenance actions can be taken based on real-time data insights, and for feasibility first reasons, we only chose SVM for simple machine learning at design time.

Improving the traffic light communication system to make its interface more user-friendly will facilitate better interaction with stakeholders and ensure that important information is more accessible and actionable.

What’s more, ensuring seamless compatibility with other transportation and infrastructure management systems can make digital twins a key component of the broader smart city framework, thereby improving overall transportation efficiency and safety.

Object recognition is also a very feasible development direction. Machine learning models such as yolo algorithm can be used to transmit real-time effects with cameras on Bridges and identify and locate vehicles through object recognition technology, which is also of great help to the analysis of bridge safety.

Apart from that, we also explored other advanced techniques monitoring the state of the bridges to give us a direction of improvement. For example, Kang et al. introduced an innovative approach to bridge health monitoring marks a significant advancement in the field, combining multimedia knowledge and digital twin technology to revolutionize traditional monitoring method[10]. By harnessing the power of digital twins, which serve as virtual replicas of physical bridges, this method provides a dynamic and comprehensive view of the bridge's structural health in real time. Model-based structural health monitoring and diagnosis involves the process of refining the digital model by incorporating differences observed between measured data and the initial digital model information. Subsequently, the updated digital model is compared to either the original digital model or the physical bridge itself to identify any structural damage or assess its operational performance.[11][12] The process of model-based structural health monitoring and diagnosis is intricate, involving continuous refinement of digital models through the integration of measured data. This iterative process allows for the detection of structural damage and the evaluation of operational performance with precision and accuracy. However, the inherent complexity of model updating, compounded by the scarcity of measurement points, presents formidable challenges. Despite these obstacles, the potential benefits of this approach are immense, offering unprecedented insights into bridge behavior and condition.[13] To address the limitations of model updating, particularly in scenarios with limited measurement points, innovative strategies are required. Kang et al.'s method introduces a novel approach to monitor the anti-overturning behavior of box girder bridges. By leveraging measured traffic loads, this method enhances the assessment of structural stability and integrity, mitigating the risk of potential accidents and structural failures.[14][15] Furthermore, the development of a simulation approach for modeling traffic loads using machine vision technology represents a significant advancement in bridge monitoring techniques. This approach enables engineers to simulate and analyze various traffic scenarios with greater accuracy and efficiency, providing valuable insights into the bridge's response to different environmental conditions and loading scenario.[16]

By implementing these enhancements, projects can achieve higher levels of efficiency, safety, and proactive management, aligned with future smart infrastructure goals.

References

[1] Farrar, C. R., & Worden, K. (2007). An introduction to structural health monitoring. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 365(1851), 303-315.

[2] Pang, C., Lee, G., Kim, T., Kim, S. M., Kim, H. N., Ahn, S., & Suh, K. (2012). A flexible and highly sensitive strain-gauge sensor using reversible interlocking of nanofibres. Nature Materials, 11(9), 795–801. https://doi.org/10.1038/nmat3380

[3] Yang, S., & Lu, N. (2013). Gauge factor and stretchability of Silicon-on-Polymer strain gauges. Sensors, 13(7), 8577–8594. https://doi.org/10.3390/s130708577

[4] eFunda, Inc. (n.d.). Strain gage: materials. https://www.efunda.com/designstandards/sensors/strain_gages/strain_gage_selection_matl.cfm

[5] Video monitoring system of security based on Wi-Fi. (n.d.). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/7531471

[6] Yuan, Z. H., Xu, M. Y., & Qi, X. X. (2014). The bearing vibration signal collecting system based on STM32F103C8T6. Advanced Materials Research, 971–973, 1376–1379.

[7] Integrated healthcare system for Saline monitoring, patient communication, and heart disease prediction using IoT and machine learning algorithms. (n.d.). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/10497196

[8] Zhejiang University Average Salary and Salary Index of Fresh Graduates in 2023. (n.d.). https://www.xinchou.com/College/cn_21001_2023

[9] *IEEE Code of Ethics*. (n.d.). https://www.ieee.org/about/corporate/governance/p7-8.html

[10] J.-S. Kang, K. Chung and E. J. Hong, "Multimedia knowledge-based bridge health monitoring using digital twin", Multimedia Tools Appl., vol. 80, no. 26, pp. 34609-34624, Nov. 2021.

[11] T. Marwala, Finite-Element-Model Updating Using Computational Intelligence Techniques: Applications to Structural Dynamics, London, U.K.:Springer, 2010.

[12] A. J. Garcia-Palencia, E. Santini-Bell, J. D. Sipple and M. Sanayei, "Structural model updating of an in-service bridge using dynamic data", Struct. Control Health Monitor., vol. 22, no. 10, pp. 1265-1281, Oct. 2015.

[13] M. I. Friswell, "Damage identification using inverse methods" in Dynamic Methods for Damage Detection in Structures, Vienna, Austria:Springer-Verlag Wien, 2007.

[14] L. Ge, D. Dan and X. Yan, "Real time monitoring and evaluation of overturning risk of single-column-pier box-girder bridges based on identification of spatial distribution of moving loads", Eng. Struct., vol. 210, May 2020.

[15] D. Dan, X. Yu, X. Yan and K. Zhang, "Monitoring and evaluation of overturning resistance of box girder bridges based on time-varying reliability analysis", J. Perform. Constructed Facilities, vol. 34, no. 1, pp. 04019101-1-04019101-12, 2020.

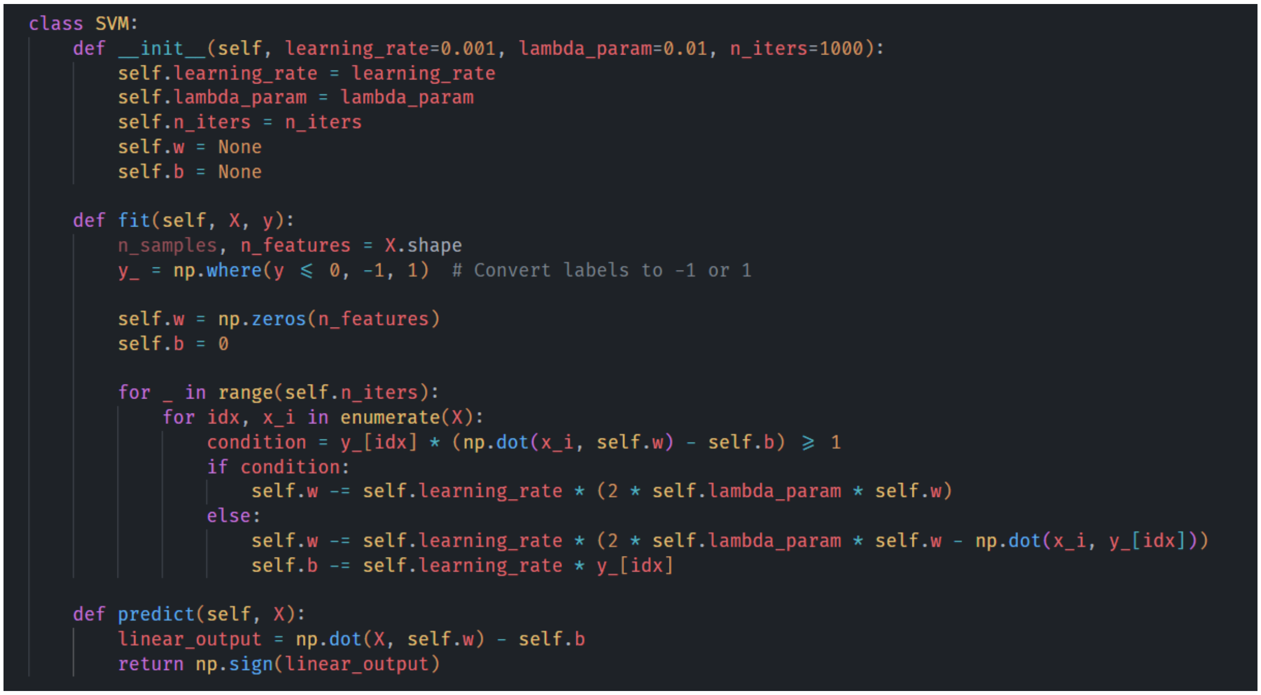
[16] L. Ge, D. Dan, Z. Liu and X. Ruan, "Intelligent simulation method of bridge traffic flow load combining machine vision and weigh-in-motion monitoring", 2020.

Appendix A Requirement and Verification Table

Table 2 System Requirements and Verifications

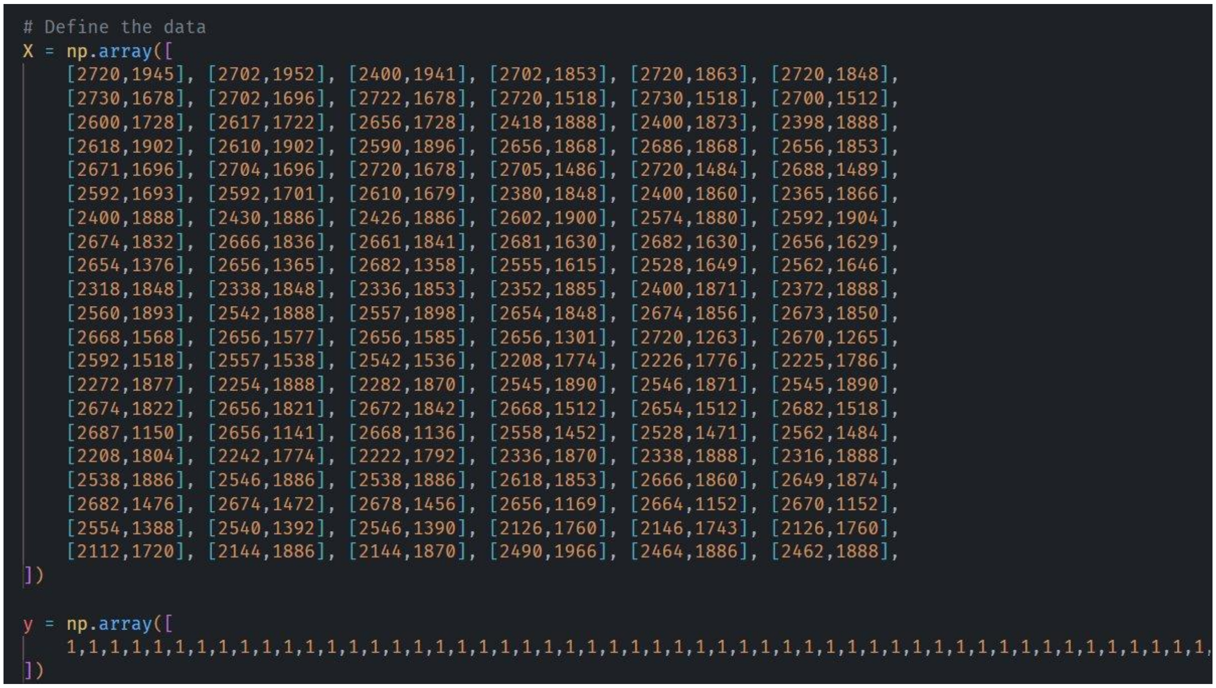
Requirement	Verification	Verification status (Y or N)
1. Voltage regulator requirement a. When the input voltage is 6.5-12V, the output voltage is 5V±0.05V.	1. Verification a. Measurements are made in the laboratory using generators and oscilloscopes.	Y
2. Strain gauge requirement a. When the strain gauge is bent in two directions, the resistance increases and decreases respectively. b. When the resistance value is small, the output voltage becomes smaller. c. The output voltage range is 0-5V.	2. Verification a. Use a resistance meter to measure. b. Measure the output voltage with an oscilloscope. c. Increase the bend in the range of 30 degrees and ensure that the output voltage is always below 5V.	Y
3. Controllerc requirement a. Must process sensor data in real-time with a latency of no more than 10ms. b. The traffic light system works according to the design logic.	3. Verification a. Verify that the error time is within the allowable range. b. Check traffic light status.	Y
4. Bluetooth requirement a. Must update the digital model in real-time with a maximum delay of 100ms from data receipt. b. Must simulate bridge behavior with an accuracy of 98% compared to the physical model. c. Must be capable of running predictive algorithms to forecast potential structural issues.	4. Verification a. Verify that the error time is within the allowable range. b. Confirm errors in pressure signal and strain gauge signal. c. Check the running status of the algorithm.	Y

Appendix B Code Snippets



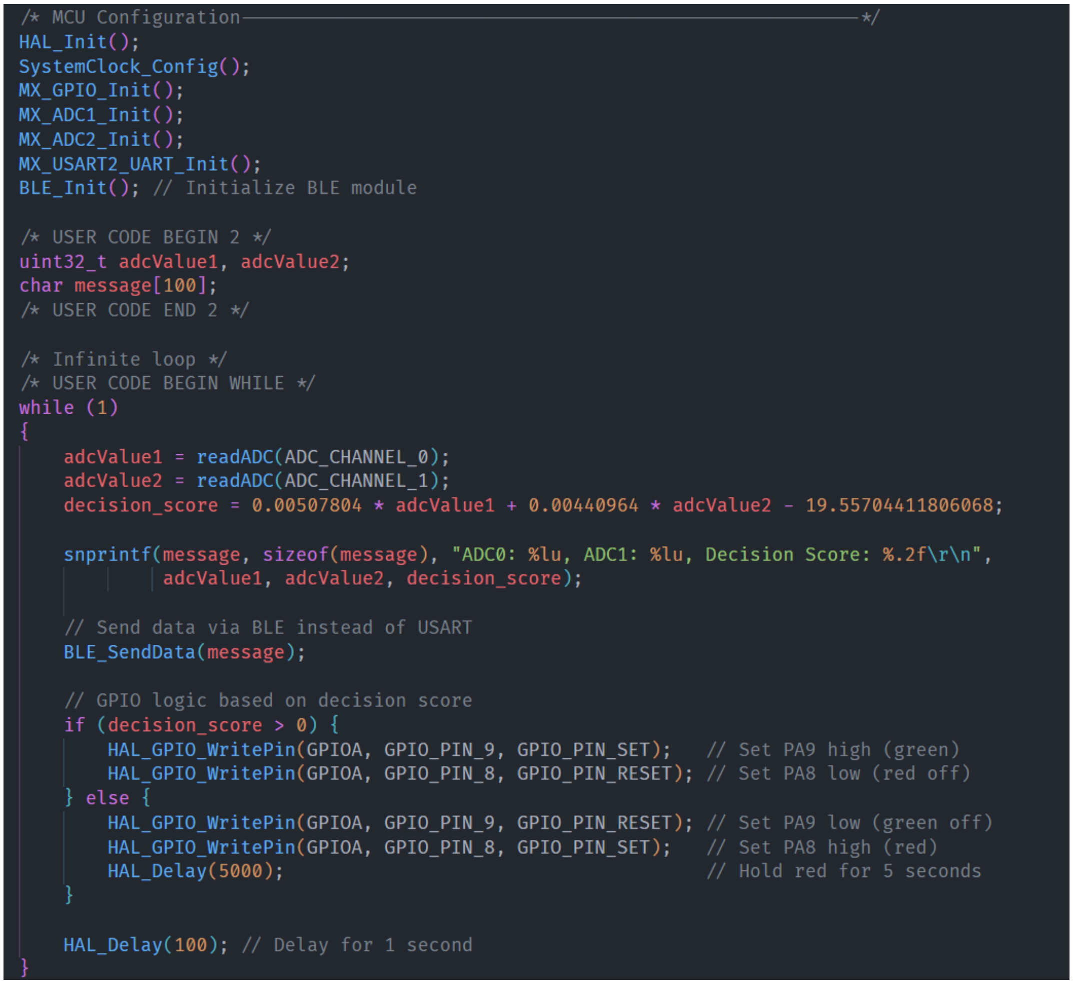
```
class SVM:
    def __init__(self, learning_rate=0.001, lambda_param=0.01, n_iters=1000):
        self.learning_rate = learning_rate
        self.lambda_param = lambda_param
        self.n_iters = n_iters
        self.w = None
        self.b = None

    def fit(self, X, y):
        n_samples, n_features = X.shape
        y_ = np.where(y ⩽ 0, -1, 1)  # Convert labels to -1 or 1

        self.w = np.zeros(n_features)
        self.b = 0

        for _ in range(self.n_iters):
            for idx, x_i in enumerate(X):
                condition = y_[idx] * (np.dot(x_i, self.w) - self.b) ⩾ 1
                if condition:
                    self.w -= self.learning_rate * (2 * self.lambda_param * self.w)
                else:
                    self.w -= self.learning_rate * (2 * self.lambda_param * self.w - np.dot(x_i, y_[idx]))
                    self.b -= self.learning_rate * y_[idx]

    def predict(self, X):
        linear_output = np.dot(X, self.w) - self.b
        return np.sign(linear_output)
```

Figure 7 Support Vector Machine class



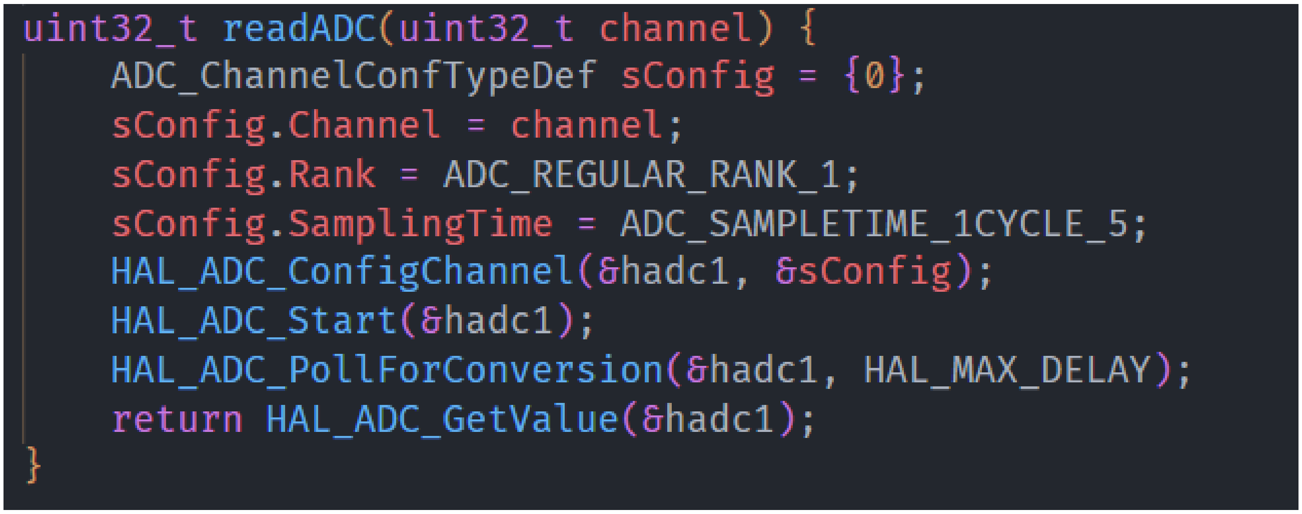
```
# Define the data
X = np.array([
    [2720,1945], [2702,1952], [2400,1941], [2702,1853], [2720,1863], [2720,1848],
    [2730,1678], [2702,1696], [2722,1678], [2720,1518], [2730,1518], [2700,1512],
    [2600,1728], [2617,1722], [2656,1728], [2418,1888], [2400,1873], [2398,1888],
    [2618,1902], [2610,1902], [2590,1896], [2656,1868], [2686,1868], [2656,1853],
    [2671,1696], [2704,1696], [2720,1678], [2705,1486], [2720,1484], [2688,1489],
    [2592,1693], [2592,1701], [2610,1679], [2380,1848], [2400,1860], [2365,1866],
    [2400,1888], [2430,1886], [2426,1886], [2602,1900], [2574,1880], [2592,1904],
    [2674,1832], [2666,1836], [2661,1841], [2681,1630], [2682,1630], [2656,1629],
    [2654,1376], [2656,1365], [2682,1358], [2555,1615], [2528,1649], [2562,1646],
    [2318,1848], [2338,1848], [2336,1853], [2352,1885], [2400,1871], [2372,1888],
    [2560,1893], [2542,1888], [2557,1898], [2654,1848], [2674,1856], [2673,1850],
    [2668,1568], [2656,1577], [2656,1585], [2656,1301], [2720,1263], [2670,1265],
    [2592,1518], [2557,1538], [2542,1536], [2208,1774], [2226,1776], [2225,1786],
    [2272,1877], [2254,1888], [2282,1870], [2545,1890], [2546,1871], [2545,1890],
    [2674,1822], [2656,1821], [2672,1842], [2668,1512], [2654,1512], [2682,1518],
    [2687,1150], [2656,1141], [2668,1136], [2558,1452], [2528,1471], [2562,1484],
    [2208,1804], [2242,1774], [2222,1792], [2336,1870], [2338,1888], [2316,1888],
    [2538,1886], [2546,1886], [2538,1886], [2618,1853], [2666,1860], [2649,1874],
    [2682,1476], [2674,1472], [2678,1456], [2656,1169], [2664,1152], [2670,1152],
    [2554,1388], [2540,1392], [2546,1390], [2126,1760], [2146,1743], [2126,1760],
    [2112,1720], [2144,1886], [2144,1870], [2490,1966], [2464,1886], [2462,1888],
])

y = np.array([
    1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
])
```

Figure 8 Custom Dataset



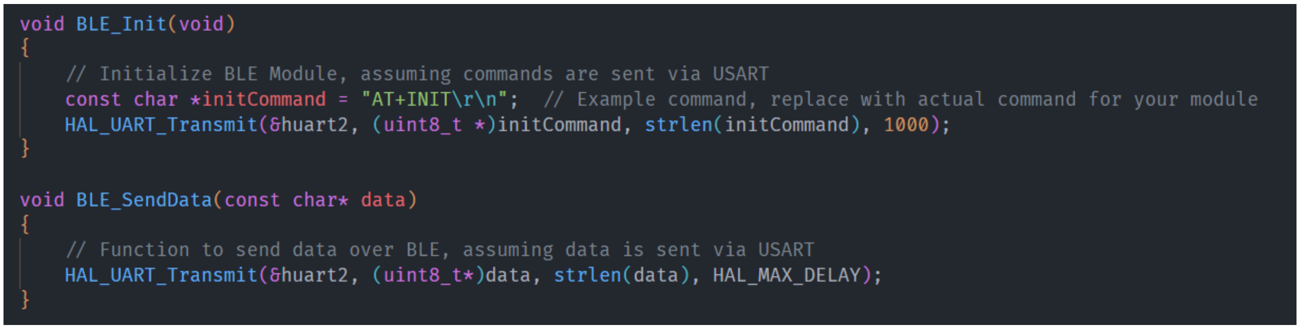
```c
/* MCU Configuration--------------------------------------------------------*/
HAL_Init();
SystemClock_Config();
MX_GPIO_Init();
MX_ADC1_Init();
MX_ADC2_Init();
MX_USART2_UART_Init();
BLE_Init(); // Initialize BLE module

/* USER CODE BEGIN 2 */
uint32_t adcValue1, adcValue2;
char message[100];
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    adcValue1 = readADC(ADC_CHANNEL_0);
    adcValue2 = readADC(ADC_CHANNEL_1);
    decision_score = 0.00507804 * adcValue1 + 0.00440964 * adcValue2 - 19.55704411806068;

    snprintf(message, sizeof(message), "ADC0: %lu, ADC1: %lu, Decision Score: %.2f\r\n",
             adcValue1, adcValue2, decision_score);

    // Send data via BLE instead of USART
    BLE_SendData(message);

    // GPIO logic based on decision score
    if (decision_score > 0) {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_SET);   // Set PA9 high (green)
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_RESET); // Set PA8 low (red off)
    } else {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, GPIO_PIN_RESET); // Set PA9 low (green off)
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, GPIO_PIN_SET);   // Set PA8 high (red)
        HAL_Delay(5000);                                      // Hold red for 5 seconds
    }

    HAL_Delay(100); // Delay for 1 second
}
```

Figure 9 MCU configuration



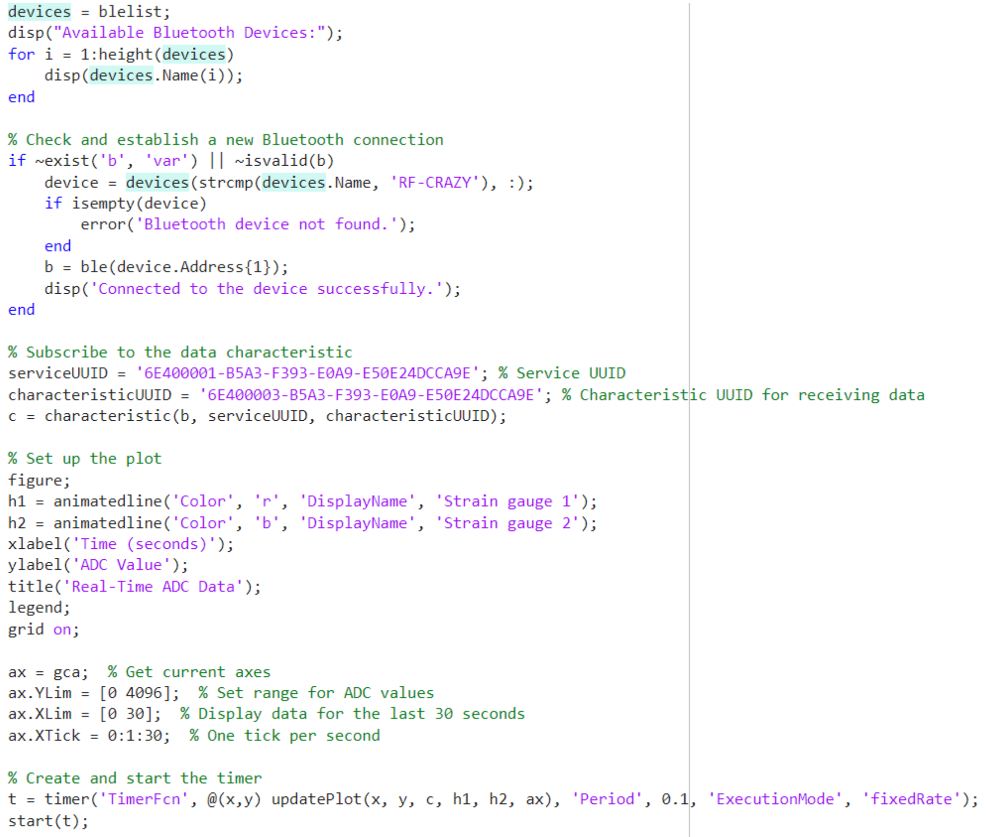
```c
uint32_t readADC(uint32_t channel) {
    ADC_ChannelConfTypeDef sConfig = {0};
    sConfig.Channel = channel;
    sConfig.Rank = ADC_REGULAR_RANK_1;
    sConfig.SamplingTime = ADC_SAMPLETIME_1CYCLE_5;
    HAL_ADC_ConfigChannel(&hadc1, &sConfig);
    HAL_ADC_Start(&hadc1);
    HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY);
    return HAL_ADC_GetValue(&hadc1);
}
```

Figure 10 Embedded ADC



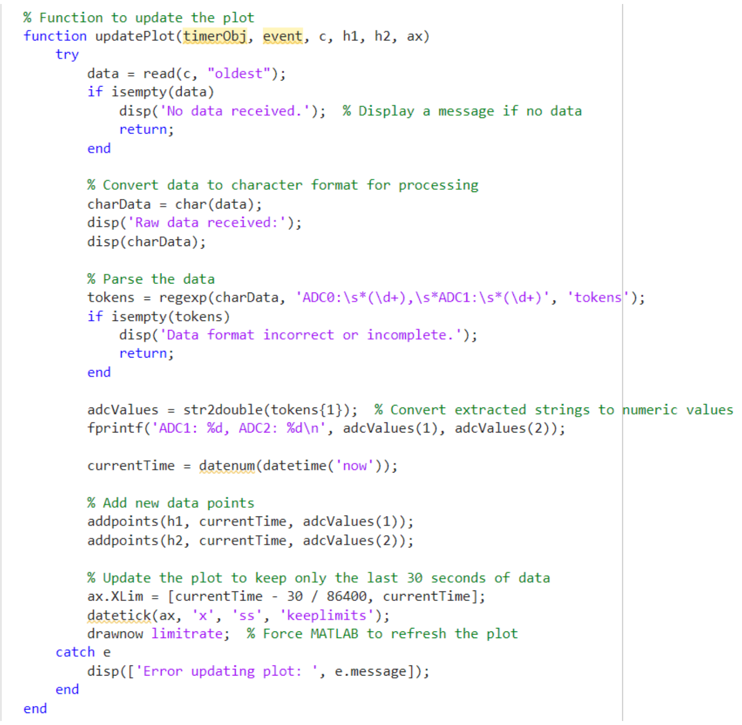
```c
void BLE_Init(void)
{
    // Initialize BLE Module, assuming commands are sent via USART
    const char *initCommand = "AT+INIT\r\n";  // Example command, replace with actual command for your module
    HAL_UART_Transmit(&huart2, (uint8_t *)initCommand, strlen(initCommand), 1000);
}

void BLE_SendData(const char* data)
{
    // Function to send data over BLE, assuming data is sent via USART
    HAL_UART_Transmit(&huart2, (uint8_t*)data, strlen(data), HAL_MAX_DELAY);
}
```

Figure 11 Bluetooth module



```
devices = blelist;
disp("Available Bluetooth Devices:");
for i = 1:height(devices)
    disp(devices.Name(i));
end

% Check and establish a new Bluetooth connection
if ~exist('b', 'var') || ~isvalid(b)
    device = devices(strcmp(devices.Name, 'RF-CRAZY'), :);
    if isempty(device)
        error('Bluetooth device not found.');
    end
    b = ble(device.Address{1});
    disp('Connected to the device successfully.');
end

% Subscribe to the data characteristic
serviceUUID = '6E400001-B5A3-F393-E0A9-E50E24DCCA9E'; % Service UUID
characteristicUUID = '6E400003-B5A3-F393-E0A9-E50E24DCCA9E'; % Characteristic UUID for receiving data
c = characteristic(b, serviceUUID, characteristicUUID);

% Set up the plot
figure;
h1 = animatedline('Color', 'r', 'DisplayName', 'Strain gauge 1');
h2 = animatedline('Color', 'b', 'DisplayName', 'Strain gauge 2');
xlabel('Time (seconds)');
ylabel('ADC Value');
title('Real-Time ADC Data');
legend;
grid on;

ax = gca;  % Get current axes
ax.YLim = [0 4096];  % Set range for ADC values
ax.XLim = [0 30];  % Display data for the last 30 seconds
ax.XTick = 0:1:30;  % One tick per second

% Create and start the timer
t = timer('TimerFcn', @(x,y) updatePlot(x, y, c, h1, h2, ax), 'Period', 0.1, 'ExecutionMode', 'fixedRate');
start(t);
```

Figure 12 Real time ADC reader Matlab



```
% Function to update the plot
function updatePlot(timerObj, event, c, h1, h2, ax)
    try
        data = read(c, "oldest");
        if isempty(data)
            disp('No data received.');  % Display a message if no data
            return;
        end

        % Convert data to character format for processing
        charData = char(data);
        disp('Raw data received:');
        disp(charData);

        % Parse the data
        tokens = regexp(charData, 'ADC0:\s*(\d+),\s*ADC1:\s*(\d+)', 'tokens');
        if isempty(tokens)
            disp('Data format incorrect or incomplete.');
            return;
        end

        adcValues = str2double(tokens{1});  % Convert extracted strings to numeric values
        fprintf('ADC1: %d, ADC2: %d\n', adcValues(1), adcValues(2));

        currentTime = datenum(datetime('now'));

        % Add new data points
        addpoints(h1, currentTime, adcValues(1));
        addpoints(h2, currentTime, adcValues(2));

        % Update the plot to keep only the last 30 seconds of data
        ax.XLim = [currentTime - 30 / 86400, currentTime];
        datetick(ax, 'x', 'ss', 'keeplimits');
        drawnow limitrate;  % Force MATLAB to refresh the plot
    catch e
        disp(['Error updating plot: ', e.message]);
    end
end
```

Figure 13 ADC plotter