ISSN 2277-2685
IJESR/Jan-Mar. 2025/ Vol-15/Issue-1s/252-260
Namitha Nagishetti *et. al.*, / International Journal of Engineering & Science Research

CAN Protocol Implementation for Data Communication

[1]Ms. Radhika Rayeekanti, [2]Namitha Nagishetti, [3]K Sai Vaishnavi, [4]Nemmala Sanjana
[1]Associate professor, Electronics and Communication Engineering, BRECW
[2,3,4]B.Tech Students, Department of Electronics and Communication Engineering, BRECW

ABSTRACT

The main aim of this project is the implementation of CAN protocol for transferring information from one place to another place.

CAN is a multi-master broadcast serial bus standard for connecting electronic control units (ECUs). Each node is able to send and receive messages, but not simultaneously: message(consisting primarily of an ID usually chosen to identify messagetype/sender and up to eight message bytes) is transmitted serially onto the bus, one bit after another this signal pattern codes the message (in NRZ) and is sensed by all nodes.

The devices that are connected by a CAN network are typically sensors, actuators and control devices. A CAN message never reaches these devices directly, but instead a host processor and a CAN controller are needed between these devices and the bus.

If the bus is free, any node may begin to transmit. If two or more nodes begin sending messages at the same time, the message with the more dominant ID (which has more dominant bits, i.e., bit 0) will overwrite other nodes' less dominant IDs, so that eventually (after this arbitration on the ID) only the dominant message remains and is received by all nodes. Bit rates up to 1 Mbit/s are possible at network lengths below 40 m. Decreasing the bit rate allows longer network distances (e.g. 125 kbit/s at 500 m).

The programming language used for developing the software to the microcontroller is Embedded/Assembly. The KEIL cross compiler is used to edit, compile and debug this program. Micro Flash programmer is used for burning the developed code on Keil in to the microcontroller Chip. Here in our application we are using AT89C51 microcontroller which is Flash Programmable IC. AT represents the Atmel Corporation represents CMOS technology is used for designing the IC. This IC is one of the versions of 8051

1- INTRODUCTION

Controller Area Network (CAN), is a robust and versatile communication protocol that can be used to send data between an ESP32(TX) to ESP32(RX) board and other devices in a networked environment. Originally developed by Bosch for automotive applications, CAN bus offers advantages in scenarios demanding robust, noise-resistant, and error-checked data transmission.

Communication via CAN is enabled through different CAN libraries and is dependent on the hardware used for the setup. It enables efficient, reliable, and error-resistant communication in environments with high electrical noise, making it ideal for critical systems. Implementing the CAN protocol involves configuring message frames, setting up filters and masks, and ensuring proper bus arbitration and error handling to facilitate seamless data transmission across nodes.

2-LITERATURE SURVEY

The Controller Area Network (CAN) protocol was developed by Bosch in the 1980s, primarily aimed at facilitating communication among multiple microcontrollers in automotive environments. Its

inception was driven by the need for a robust, costeffective solution that would reduce wiring complexity and enhance the reliability of data transmission. The foundational specifications outlined in ISO 11898 have since enabled widespread adoption in both automotive and industrial sectors, marking CAN as a significant advancement in communication technology. Research on the performance characteristics of CAN networks has provided valuable insights into their operational capabilities. Numerous studies indicate that CAN can support data transmission rates of up to 1 Mbps, with latency typically ranging from 1 to 5 milliseconds, depending on network load and configuration. The protocol's inherent error handling mechanisms—such as cyclic redundancy checks (CRC), acknowledgment bits, and automatic retransmission—significantly enhance communication reliability, particularly in electrically noisy environments. This robustness has been extensively documented in literature, with findings illustrating how these features contribute to CAN's effectiveness in critical applications.

Scalability is another key area of focus in the literature. Researchers have explored various network topologies for CAN implementations, including bus, star, and tree configurations. While bus topology remains the most common due to its simplicity and efficiency, alternative designs can provide enhanced scalability and reliability for specific applications. Studies suggest that CAN networks can effectively support up to 30 nodes without substantial performance degradation; however, increasing the number of nodes often leads to higher message collision rates and increased latency (Kurt & Müller, 2018). These findings underscore the importance of careful network design in optimizing CAN performance.

The application of CAN extends beyond automotive systems into industrial automation, robotics, and even the Internet of Things (IoT). Its reliability makes it a preferred choice for coordinating complex operations in manufacturing environments.

Recent studies highlight the potential of integrating CAN with IoT devices, which could leverage its robust communication capabilities to enhance smart home systems and industrial IoT applications (Patel & Desai, 2022). This expansion reflects a growing recognition of CAN's versatility and adaptability in a rapidly evolving technological landscape. In terms of advancements, the introduction of CAN Flexible Data-rate (CAN FD) represents a significant evolution in the protocol, allowing for greater data throughput and longer message lengths. This enhancement addresses the increasing demand for bandwidth in contemporary applications and positions CAN as a relevant technology in the context of modern communication needs (He et al., 2023). Ongoing research continues to focus on optimizing CAN performance through innovative techniques such as adaptive message scheduling and prioritization strategies, which promise to further enhance its capabilities and application scope. Its robust error handling, scalability, and adaptability make it a vital component in contemporary data communication systems. As research progresses, the continued evolution of CAN promises to keep it relevant in an increasingly interconnected world, highlighting the importance of ongoing exploration into its operational characteristics and potential applications.

3- PROPOSED MODEL

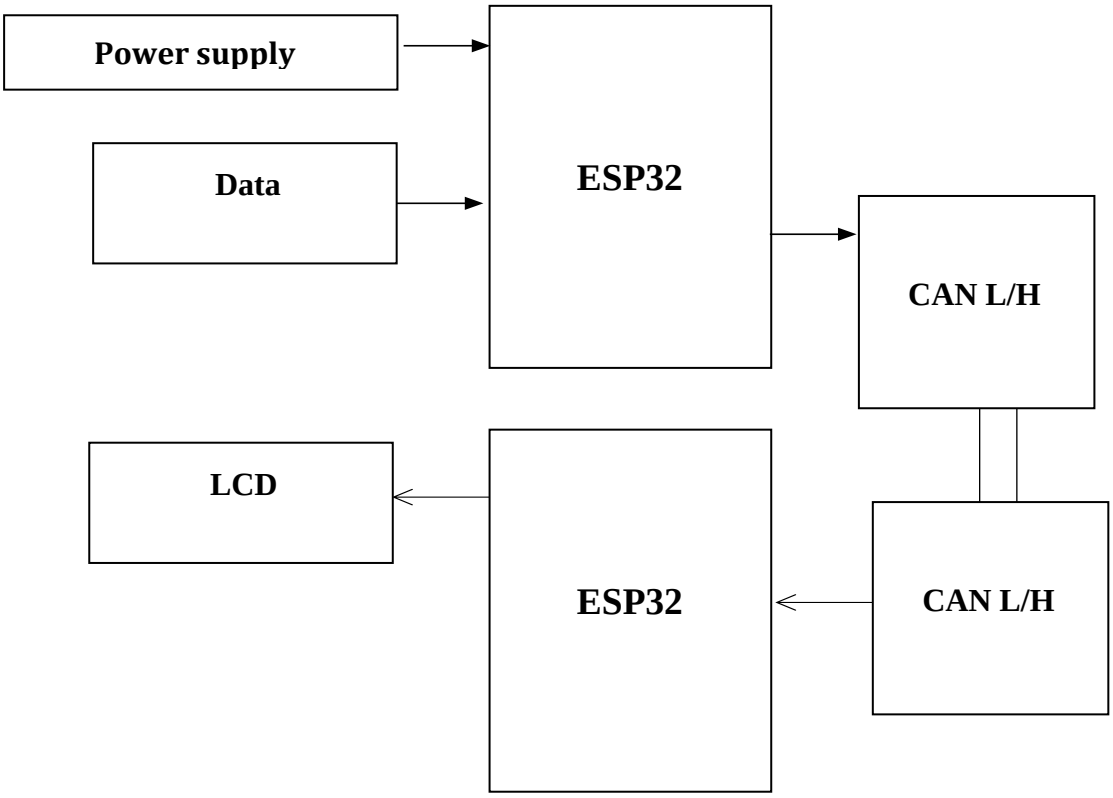


Fig : Proposed Model

The diagram in Figure 2.1 represents the proposed system architecture involving two ESP32 microcontrollers connected via a Controller Area Network (CAN) bus. The design aims to achieve data communication between the ESP32 modules with an external power supply and LCD display.

- Power Supply: The ESP32 microcontroller is powered by an external power source, ensuring stable voltage for operation. The ESP32 supports various low power modes, such as deep sleep, which allows for significant power savings during idle periods. This feature is crucial for battery-operated devices in a CAN network, enabling longer operational life without compromising communication readiness.
- Data Input: The first ESP32 module receives input data, which could come from sensors or any external devices. The ESP32 module processes the received input data efficiently, utilizing its dual-core architecture to handle multiple tasks simultaneously. This allows for real-time data manipulation and conversion before packaging the information into CAN messages for transmission to other nodes in the network.
- ESP32 Microcontrollers: The two ESP32 units are the core processing elements of the system. One ESP32 receives data input and communicates with the second ESP32 over the CAN bus. This setup allows for efficient data exchange, enabling coordinated actions between modules and enhancing the overall responsiveness and reliability of the system.
- CAN L/H (Controller Area Network): The CAN bus (high/low differential signalling) enables communication between the two ESP32 modules, allowing

for real-time data transfer.

- LCD Display: The second ESP32 drives an LCD display, likely intended for user interface or status output, by processing the received data from the first ESP32.

4- Hardware Description

Power Supply

The power supply section is the section which provides +5V for the components to work. IC LM7805 is used for providing a constant power of +5V.

The ac voltage, typically 220V, is connected to a transformer, which steps down the ac voltage down to the level of the desired dc output. A diode rectifier then provides a full-wave rectified voltage that is initially filtered by a simple capacitor filter to produce a dc voltage. This resulting dc voltage usually has some ripple or ac voltage variation.

A regulator circuit removes the ripples and also retains the same dc value even if the input dc voltage varies, or the load connected to the output dc voltage changes. This voltage regulation is usually obtained using one of the popular voltage regulator IC units.

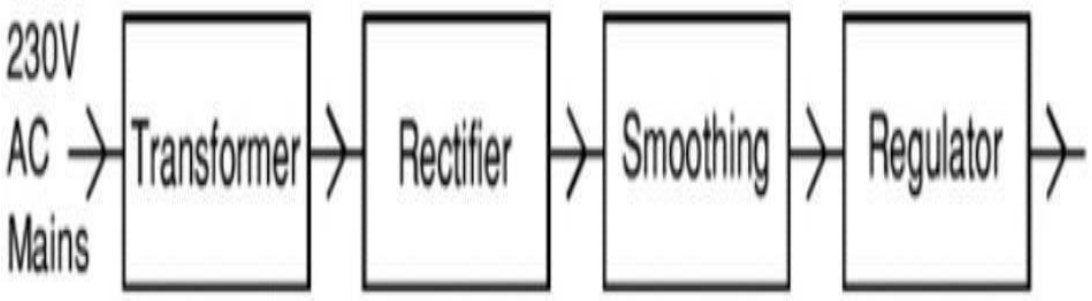


Fig 4.1: Block Diagram of Power Supply

Transformer

Transformers convert AC electricity from one voltage to another with little loss of power. Transformers work only with AC and this is one of the reasons why mains electricity is AC.

Step-up transformers increase voltage, step-down transformers reduce voltage. Most power supplies use a step-down transformer to reduce the dangerously high mains voltage (230V in India) to a safer low voltage.

The input coil is called the primary and the output coil is called the secondary. There is no electrical connection between the two coils; instead they are linked by an alternating magnetic field created in the soft-iron core of the transformer. Transformers waste very little power so the power out is (almost) equal to the power in. Note that as voltage is stepped down current is stepped up.

The transformer will step down the power supply voltage (0-230V) to (0- 6V) level. Then the secondary of the potential transformer will be connected to the bridge rectifier, which is constructed with the help of PN junction diodes. The advantages of using a bridge rectifier are it will give peak voltage output as DC.

Flowchart

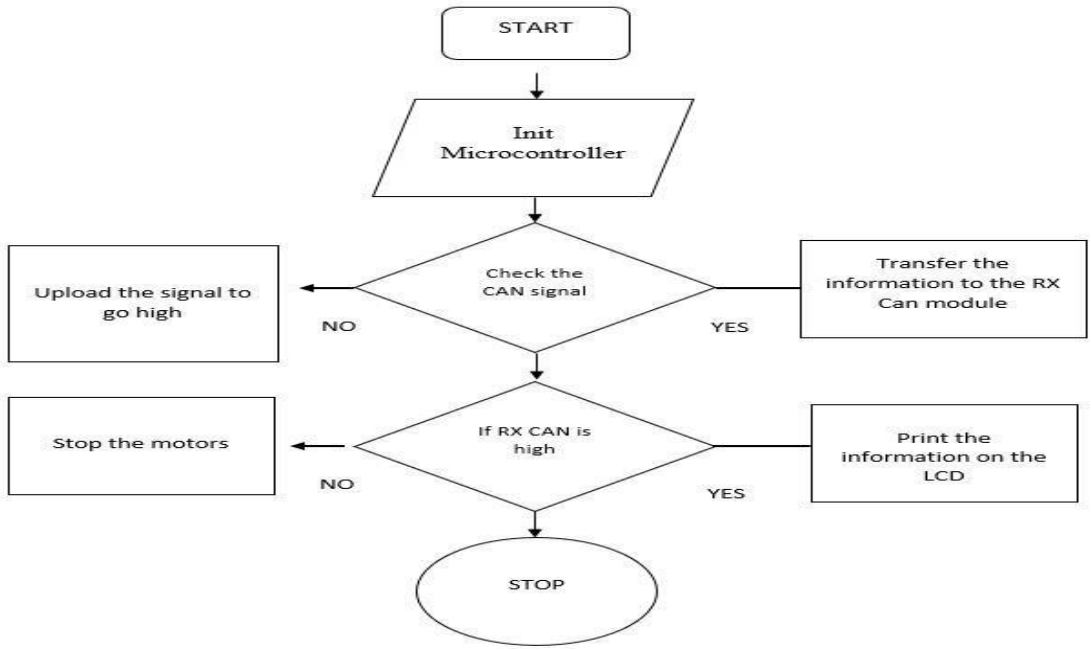


Fig: Flowchart

5-SOFTWARE DESCRIPTION

IDLE Arduino Software

You'll need to download the Arduino Software package for your operating system.

When you've downloaded and opened the application you should see something like this:

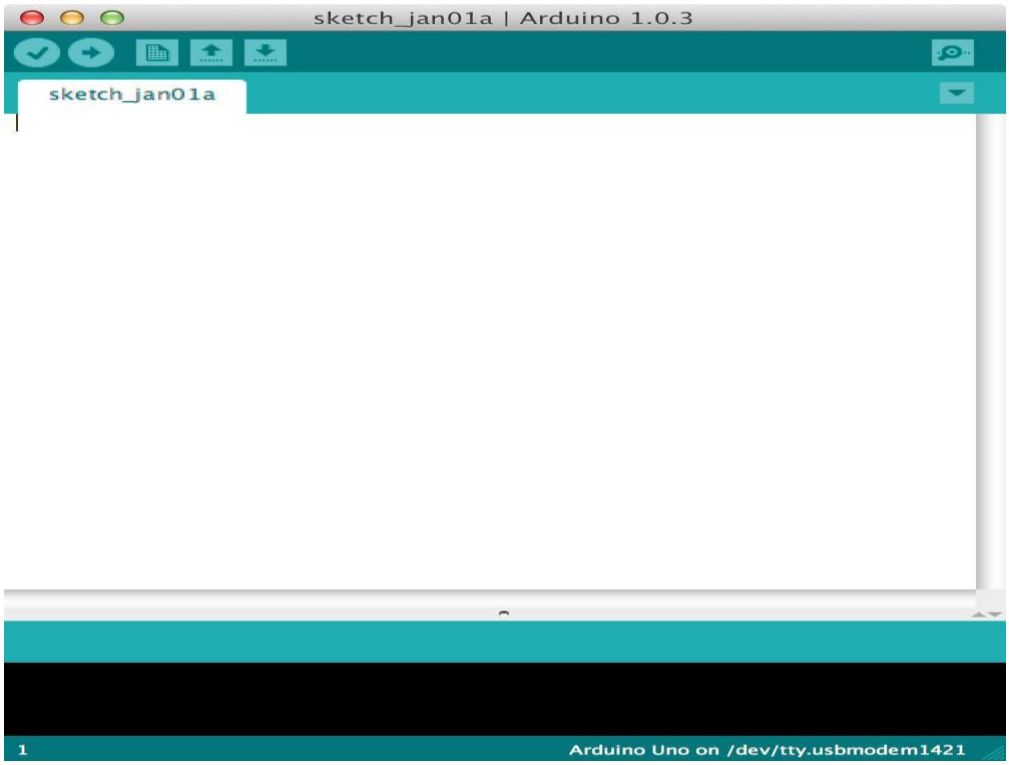


Fig 4.1: Arduino Software

• The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus.

• It connects to the Arduino hardware to upload programs and communicate with them.

• Next, you need to set up your Arduino IDE by installing the MCP_CAN library.

This library can be found in the Library Manager within the IDE.

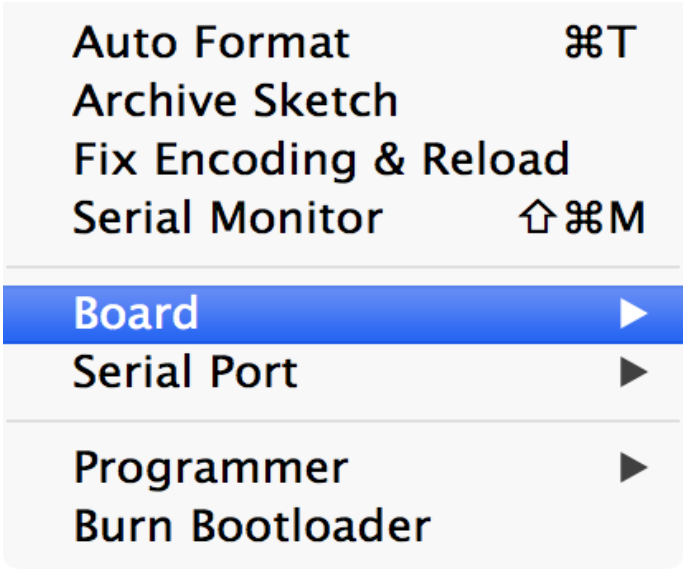
• To access it, go to **Sketch > Include Library > Manage Libraries...,** and then search for "MCP_CAN."

• By uploading the sender code to one Arduino and the receiver code to another, you can establish a simple CAN communication system.

• During the setup, initialize the CAN module by calling CAN.begin() with the desired baud rate (e.g., CAN_500KBPS). If the initialization is successful, you'll see a confirmation message in the Serial Monitor.

This is where you type the code you want to compile and send to the Arduino board.

The Initial Setup: We need to setup the environment to **Tools** menu and select **Board**.



Then select the type of Arduino you want to program, in our case it's the **Arduino Uno**.

6- RESULTS

We have finally reached our goal. We have to implement the hardware as all equipment is at our hands. So, in the whole procedure is as follows

Finally, we have successfully implemented the circuit. It can be easily implemented in elections and remote and local services. The results that the CAN protocol provides a reliable, efficient, and scalable solution for real-time data communication. The system's strengths in error handling and message arbitration contribute significantly to its robustness. Additionally, integrating the CAN network with IoT frameworks and cloud platforms may open new avenues for data analytics and remote monitoring, positioning the system for future advancements in communication technology.

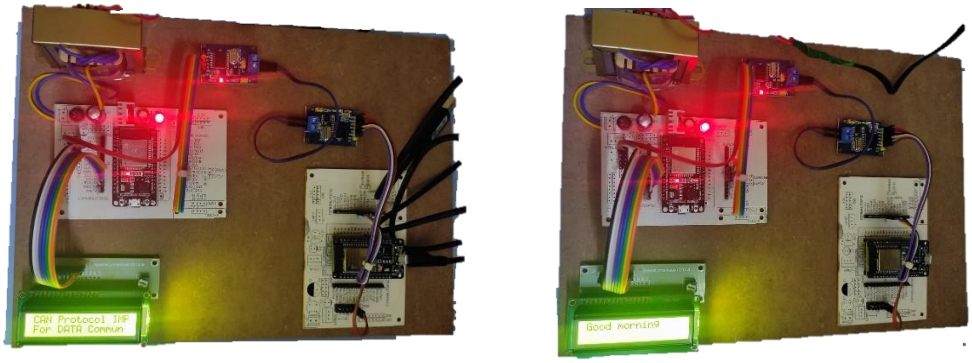


Fig 5.1: Output after Implementing CAN

7- ADVANTAGES, APPLICATIONS & LIMITATIONS

Advantages

1. Industrial automation and control is much easier to implementation.
2. This system is fast and accurate.
3. Cost-Effective.
4. Real time and easy to operate.
5. Reduced wiring, weight and errors.
6. Flexibility.

Applications

1. This system can be used in Industries.
2. Automotive
3. Medical instruments and equipment.
4. Railway Systems

Limitations of this project

- Requires CAN modules which might be getting expensive in some cases.
- CAN has a maximum data rate of 1 Mbps, which may be insufficient for highbandwidth applications.

8-CONCLUSION

The project "**CAN PROTOCOL IMPLEMENTATION FOR DATA COMMUNICATION**" has been successfully designed and tested. It has been developed by integrating features of all the hardware components used. Presence of every module has been reasoned out and placed carefully thus contributing to the best working of the unit. Secondly using highly advanced IC's and with the help of growing technology the project has been successfully implemented.

The implementation of the CAN protocol for data communication has proven to be a robust and efficient solution for a wide range of applications, particularly in automotive and industrial settings. By leveraging the protocol's inherent strengths— such as reliable error detection, real-time communication capabilities, and scalability—systems can achieve high levels of data integrity and responsiveness. The successful integration of CAN with microcontrollers, like the ESP32, demonstrates its versatility in enabling effective communication between multiple nodes. Overall, the project highlights the importance of meticulous design and careful consideration of network architecture to maximize the benefits of the CAN protocol.

REFERENCES

1) Ahuja, Vanita, and Shalini Priyadarshini. "Effective Communication Management for Urban Infrastructure Projects.

2) " Project Management National Conference, India. Academy of Management Review 9 (3):428-437. Bilczynska Wojcik, Aneta.

3) "Communication management within virtual teams in global projects." Dublin Business School. Caltrans. Project Communication Handbook.

4) "Effective communication: stone age to e-comm.

5) " Proceedings of the Project Management Institute Annual Seminars & Symposium.

6) Hoezen, MEL, IMMJ Reymen, and GPMR Dewulf. Effective communications for project management: CRC Press. Komi- Sirviö, Seija, and Maarit Tihinen.