International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-08, Issue-09, Dec 2022

Data Logger for Recording and Monitoring of Various Parameters of Electric Vehicles

[1]Pallvi , M.Tech ECE, Swami Vivekanand Institute of Engineering and Technology, Banur, Patiala, India. Pallvisharma576@gmail.com

[2]Dr. Shashi Jawla, HOD ECE, SVIET, Banur, India.

Abstract - The increase in greenhouse gases and growing pollution has predicted that electric vehicles will gradually be rising in the automobile industry in the upcoming future. It is important to develop an efficient data logger that fulfils the requirements of development in an electric vehicle. A data logger is an electronic automated device that monitors and registers data over a particular duration with the help of different sensors. Data loggers use a microcontroller, sensors to collect data and an internal or external memory for storing the data. This paper deals with the logging of data from pre-installed sensors in an Electric Vehicle. The CAN communication is used for lossless and high-speed data transmission. A test case consisting of some random data is successfully transmitted in CAN Normal mode. The same data is received on another microcontroller using CAN loopback mode. This data is analysed with the help of Digital Storage Oscilloscope (DSO).

LIST OF ABBREVIATION

EV	Electric Vehicle
CAN	Control Area Network
ECU	Electronics Component Unit
SPI	Serial Peripheral Interface
DC	Direct Current
LCD	Liquid Crystal Display
IoT	Internet of Things
MCU	Microcontroller Unit
API	Application Programming Interface
PC	Personal Computer

 DOI : 10.35291/2454-9150.2022.0495

1. INTRODUCTION

A surge in harmful secretions has made a shift in vehicles from a ignition engine vehicle to an electric vehicle that is powered by electricity rather than fuel. It means that it does not produce harmful emissions. The three-phase AC run a motor which will thus in turn drive wheel. Electric vehicles (EVs) use renewable source as a fuel, are safe to drive, and also cost-effective.

The development of EVs requires engineers to test their designed prototypes in real driving conditions. The measurement results provide an overview of the system performance and indicators of abnormal or faulty operations [1]. To measure the parameters of an electric vehicle model, a data logger presented in this paper with features of MMC based data storage and IoT connectivity. A data logger is an electronic automated device that monitors and registers data with help of sensors [2]. Depending on the type of sensor and signal conditioning used, a data logger can monitor temperature, humidity, pH, pressure, voltage and current levels etc's. Data loggers are either standalone or wireless. A standalone device stores data which can be reviewed later on PC. Wireless device transmits data in real time to connected devices. Microcontroller consists of components such as memory, peripherals, and processors. This microcontroller will work like a human brain that will regulate all activities Performed by the data logger system such as the task of regulating the sensors and sending the data from sensors into storage media [3].

1.1 LITERATURE REVIEW

A detailed literature review related to EVs is presented as follows: Basu *et al*. [1] presented an overall review of different type of sensors used in electric vehicles and the different kinds of EVs in the market. The authors concluded that under severe weather conditions position sensors are the best sensors. Cheng *et al*. [2] discussed major components in electric vehicle like battery technology (charger design), engine, steering, and brakes. The paper described the development of electric vehicles and comparison of different sensors. Nagar *et al*. [3] discussed the design and architecture of a device for recording electrical vehicle parameters. Electric vehicle data recorder (EVDR) allows registration of temperatures, currents and voltages in the powertrain, registration of ambient temperature; registration of GPS (Global Positioning System) data and supports data collection from CAN bus. Svendsen *et al*. [5] presented the importance of data loggers. Information from sensors has been accessed using the data logger. Thermocouple has been used to sense temperature and Hall Effect sensor for measuring vibrations. This data received is then sent for post analysis to a software or website. Padhee and Singh [6] discussed data logging and supervisory control in plant automation. The paper presents the design and simulation of a temperature data logger using CAN. Okwudibe *et al*. [7] designed and simulated a temperature data logger. The system consists of microcontroller and external Global navigation satellite system (GNSS), Radio Frequency Identification (RFID) and On-board Diagnostic (OBD) for port communication. This system retrieves the data from the car in which it is stored and transfers the data to a remote server using Wi-Fi.Fuentes *et al*. [8] developed a low-cost independent data logger for monitoring EV system using compliant Arduino. The parameters sensed were voltage, current, power and temperature. Alves *et al*. [9] discussed indirect ways to measure energy consumption in vehicles. The parameters used were speed, power, current, voltage. This work represents the development, validation and implementation of a new power consumption measurement system in BEV, based on the performance of a particular vehicle and the reliance on real-world data measured with a portable laboratory board. Nhivekar and Mudholker [10] developed embedded system based on microcontroller for data logging and remote monitoring. Key parameters used were temperature and humidity. Advantages were that storage capability is increased by interfacing the embedded system with external memory such as SD card or Multimedia card.Krogh *et al*. [11] analysed electric vehicle energy consumption using very large data sets.Sehgal *et al*. [12] developed data logger using ZigBee protocol. Key parameter worked upon was temperature. The proposed system has an advantage of wireless transmission of the data using ZigBee. Also, the system has very long battery life and supports a large number of nodes.Dey *et al*. [13] implemented data acquisition system using STM32 CAN architecture. CAN bus provides high quality data and is already in built in the systems. Smuts *et al*. [14] discussed issues while applying a data assimilation platform for electric vehicles by means of the Internet of Things. The results has shown that data assimilation concerns primarily relate to data availability data excellence, and interoperability between devices, IoT platforms, and EV service providers.Echavarría *et al*. [15] designed a monitoring system that collects various parameters of an EV through an OBD-II (The OBD term, stands from the acronym of "On Board Diagnosis (OBD)") port data-logger.

2. PROBLEM FORMULATION

The electric vehicle has different sensors like temperature and voltage sensors installed that communicate with each other through CANBUS.. The project proposes a general-purpose datalogger that can log data from the required sensors. The electronic component unit (ECU's) in Electric Vehicle communicate with each other using CANBUS and the data logger designed using Control Area Network (CANBUS). It is more robust due to less complexity because of common serial bus. STM32 ARM based microcontroller has advanced features

 DOI : 10.35291/2454-9150.2022.0495

of inbuilt CAN controller and high clock frequency.

2.1 OVERVIEW

A data logger is an electronic automated device that monitors and records data with the help of different sensors. Some of the important parameters that it can measure are temperature, voltage, pressure, current etc. Data loggers use a microcontroller, sensors to collect data and an internal or external memory for storing the data. Data loggers are either standalone or wireless. A standalone device stores data which can be reviewed later on PC. Wireless device transmits data in real time to connected devices. This project uses the CAN bus protocol to communicate between two microcontrollers, SD card and Internet of Things (IoT) interface to analyse and store data.

2.1.1 CONTROL AREA NETWORK

The electronic component units in the vehicle are connected with CAN bus, to interconnect individually , It does not cause any overload to the controller computer. The CAN bus system allows for central error diagnosis and configuration across all ECU's.

CAN is the main bus in the automobile. The CAN physical layer has two wires - CANH and CANL. The difference in the voltage between CANH and CANL is called V_d. A logic 1 is represented by the recessive state and is defined as V_d less than or equal to 0.5 V. A logic 0 is represented by the dominant state and is defined as V_d greater than or equal to 0.5 V. CAN is designed to be used with twisted pair cabling with 120 Ω characteristic impedance.

CAN Message Format-

A framework is a defined structure that contains data (bytes) within a network. CAN has four types of frames: Data Frame, Remote Frame, Error Frame, Overload frame.

Table 2.2 Extended Frame

Standard Frame and Extended Frame

Bosch developed the CAN protocol as a solution in 1986 and in 1991 published CAN 2.0 which included CAN 2.0A (11 bit) and CAN 2.0B (29 bit). CAN is an adopted international standard (ISO 11898). In 2012, Bosch released the CAN FD 1.0 which is CAN with flexible data rate. The CAN bus system will stay relevant and more so with rise of cloud computing, IoT and autonomous vehicles. Table 2.1 presents Standard Frame.

The extended Frame of CAN is almost similar to the standard frame except for its arbitration field. Table 2.2 presents Extended Frame for CAN.

The microcontroller used in this project is STM32F7. The microcontroller comprises of a CAN controller, which is based on 2.0B specification. The CAN controller is extended CAN Controller and that is why it is called BXCAN (Basic Extended CAN).

Table 2.1 Standard Frame

Field Name	Sub-Field	Length(bits)	Purpose
Start of Frame	SOF	1	Indicates the start of frame on CAN Bus(must be dominant (0))
Arbitration Field	Identifier	11	Decides the Message Priority(Arbitration) on CAN BUS
	RTR	1	Differentiate between Remote Frame or Request frame
Control Field	IDE	1	Tells about frame format : standard or extended For Standard : Always Dominant(0) For Extended : Always Recessive(1)
	R0	1	Reserved(must be dominant 0)
	DLC	4	Data length on the CAN bus
Data Field	D0-D8	64	Data
CRC Field	CRC	15	CRC
	CRC Delimiter	1	Must be recessive(1)
Ack field	Ack	1	Acknowledgment by receiving node.
	Ack Delimiter	1	Must be recessive(1)
End Of Frame	EOF	7	Indicates end of current frame(Must be recessive(1))

All these segments are required for transmission as well as reception purposes and these segments are actually decided by the specification. Width of the segments have to be adjusted properly to get desires bit rate on the CAN bus. Width of each segment is mentioned in terms of time quanta. The time quanta are smallest time unit for all configuration values.

2.1.2 Serial Peripheral Interface (SPI)

 DOI : 10.35291/2454-9150.2022.0495

ISSN : 2454-9150 Vol-08, Issue-09, Dec 2022

Field Name	Sub-Field	Length(bits)	Purpose
Start of Frame	SOF	1	Indicates the start of frame on CAN Bus(must be dominant (0))
Arbitration Field	Identifier	11	Decides the Message Priority(Arbitration) on CAN BUS
	SRR	1	Always Recessive
	IDE	1	Tells about frame format : standard or extended For standard: always dominant(0) For Extended: Always recessive(1)
	Identifier	18	Decides the Message Priority(Arbitration) on CAN BUS(Extended)
	RTR	1	Tells about remote or Data frame
Control Field	R0	1	Reserved(must be dominant 0)
	R1	1	Reserved(must be dominant 0)
	DLC	4	Data length on the CAN bus
Data Field	D0-D8	64	Data(As per DLC)
CRC Field	CRC	15	CRC
	CRC Delimiter	1	Must be recessive(1)
Ack field	Ack	1	Acknowledgment by receiving node.
	Ack Delimiter	1	Must be recessive(1)
End Of Frame	EOF	7	Indicates end of current frame(Must be recessive(1))

Figure 2.1 represents CAN Bit Data divided into four segments.

CAN Bit Timing

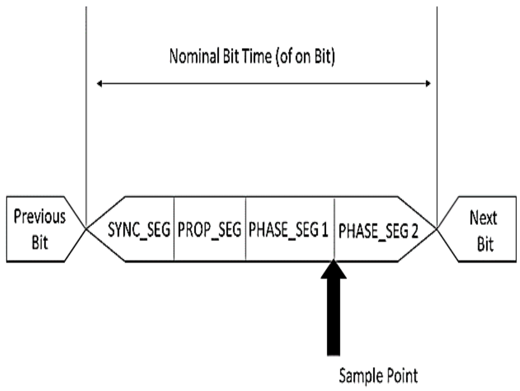

Figure 2.1. CAN Bit Data

2.1.3 Secure Digital Memory Card (SDC) Interfacing

Secure Digital Memory Card (SDC) is a memory storage chip commonly found in laptops, computer and mobile phones. They are connected with main computer using SPI interface. SD card is called solid state device because it requires no moving parts to function.

2.1.4 FAT File system basics

Microcontrollers have limited memory that restricts storage of large amounts of data. It is better to use external memory so that the microcontroller can access it during operation and the user can remove the memory and access it at any time.

Figure 2.5 presents SD card software stack.

Introduction

Serial Peripheral Interface is full duplex communication protocol which requires four wire to establish a communication.

The SPI communication method is shown in Figure 2.2. At simplest level, SPI communication consists of single bus master connected to a single bus slave. One device act as master and other as slave. In case of multiple slaves as shown in Figure 2.3, master must provide the dedicated chip select lines for each slave. The configuration is used in data acquisition system where multiple analog-to-digital (ADCs) and digital-to-analog converters (DACs) must be accessed individually.

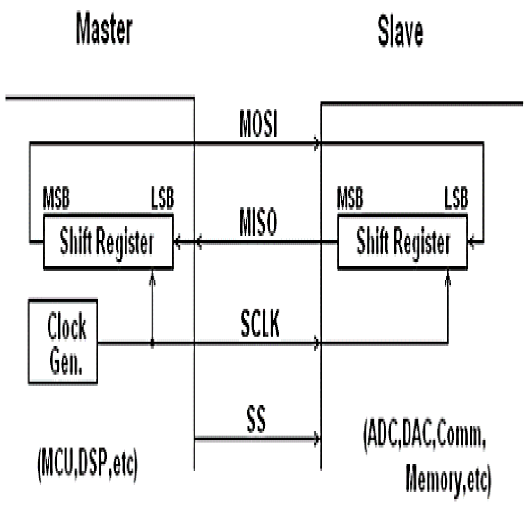

Figure 2.2 SPI Communication Method

2.2 CONSTRAINTS

- CANBUS can transfer data at 1Mbit/second.
- ARM Cortex-M7F has a maximum clock rate of 216 MHz.

2.3STANDARDS USED

- *ISO 11898-1:2015*

  It specifies the features for setting up digital information exchange between modules using the CAN data link layer. Defines the classic CAN frame format and the CAN Flexible Data Rate frame format. Old CAN stand format allows up bit bits of up to 1 Mbit / s and uploads up to 8 bytes per frame while Flexible Data Rate standalone format allows bit bits higher than 1 Mbit / s and loads longer than 8 byte each frame.

- *RS485 TIA-485 (-A) or EIA-485*

 DOI : 10.35291/2454-9150.2022.0495

ISSN : 2454-9150 Vol-08, Issue-09, Dec 2022

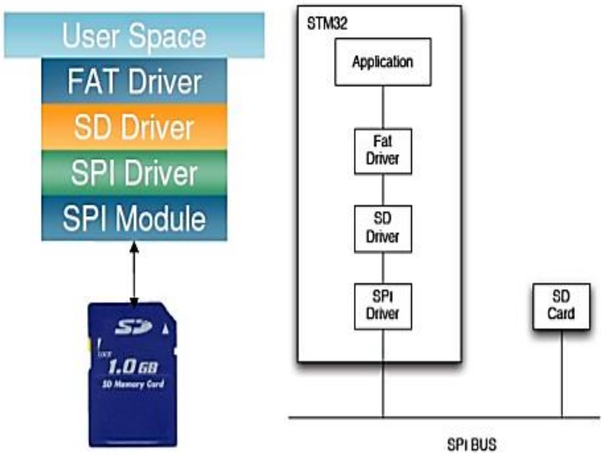

Figure 2.5 SD card software stack

2.1.5 Internet of Things (IoT)

Internet of Things (IoT) is a system of 'connected objects. Items usually include a embedded operating system and the ability to connect to the Internet or neighbouring objects. '

Figure 2.6 shows the IoT ecosystem.

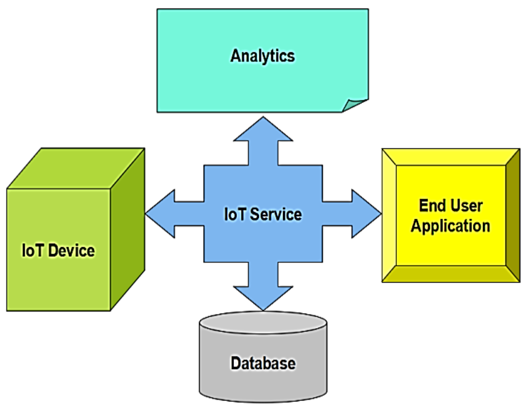

Figure 2.6 IoT Service

It is a standard that describes the electrical properties of serial lines to be used in serial communication systems. It can be used successfully over long distances and in noisy environments.

- *IEEE 802.11 standard*

IEEE 802.11 standards is a family of WLANs (Wireless Local Area Network). It is a standard that specifies the physical or MAC layer adapted. WIFI stands for Wireless Fidelity. Wi-Fi specifications are based on IEEE 802.11 standards.

DOI : 10.35291/2454-9150.2022.0495

3.1 CONCEPT MAP AND BLOCK DIAGRAM OF DATA LOGGER

Figure 3.1 represents the block diagram of data logger.

3.1.1 Control Area Network (CAN)

The communication between the microcontrollers is done by CAN protocol. The STM32 microcontroller already has CAN controller embedded in it. A CAN transceiver is connected with CAN controller to make the signals from single ended to differential signals.

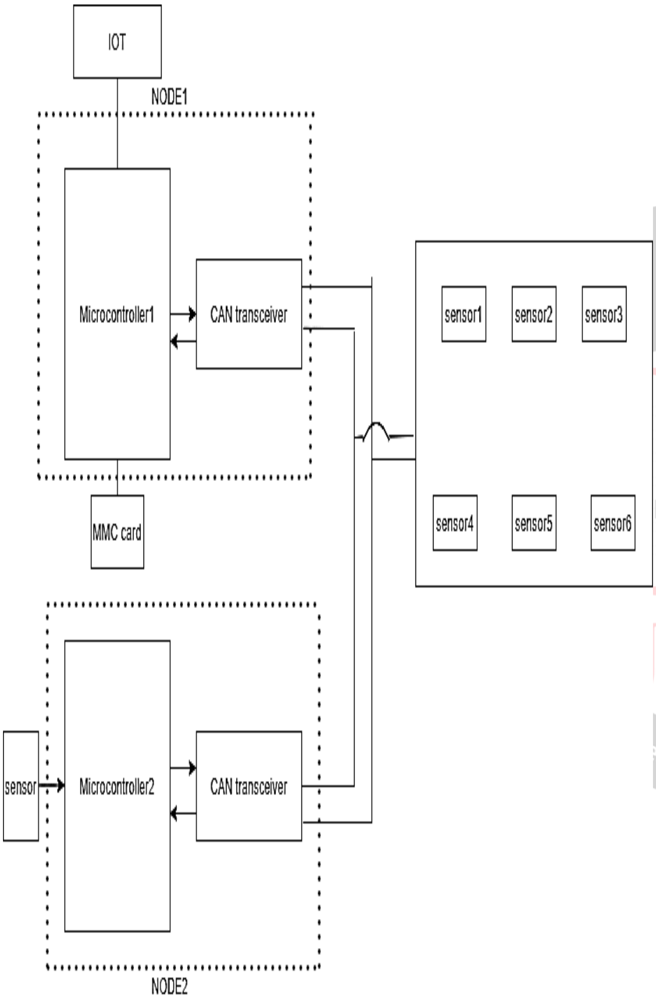

Figure 3.1 Block Diagram for Data Logger

BxCAN has 3 operating modes - NORMAL, SLEEP and INITIALISATION MODE.

Figure 3.2 shows the modes of operation when the different conditions are true in CAN registers.

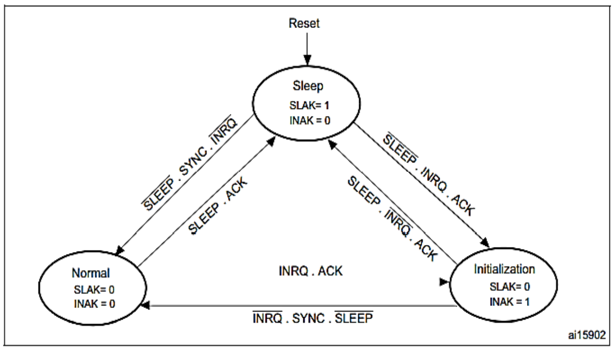

Figure 3.2 BxCAN Operating Modes

3.3 PROPOSED METHODOLOGY

In this project the BxCAN is used in NORMAL mode as shown in Figure 3.3. Here two nodes are required for operation; they are node 1 (N1) and node 2 (N2).

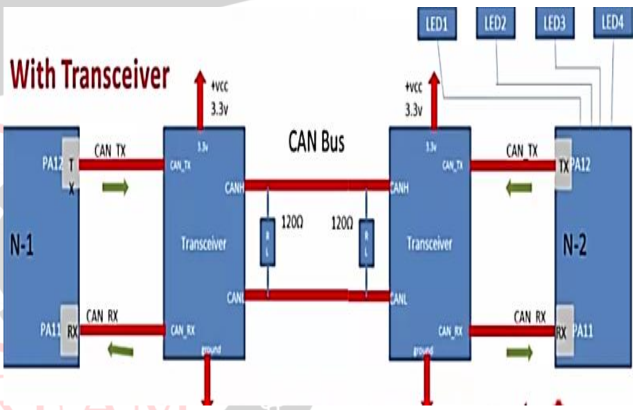

Figure 3.3. CAN Normal Mode (Courtesy: Fast bit Embedded Brain Academy)

Working:

- N1 sends a message, which is the signal to read from the particular sensor.
- Data is sent from the CAN Tx of node 1 to CAN Rx of node 2.
- When Node 2 starts receiving the packet of data, it compares received Cyclic Redundancy Check (CRC) from node 1 with calculated CRC of that packet. This is part of CAN Rx Filtering.
- If the CRC matches, then N2 concludes that the frame is correct. ACK bit is set to dominant bit which is 0 to confirm the message has been received.

As the car, is already installed with sensors, this is a proposed method to log any data from any sensor in the car by connecting the sensing microcontroller to it.

 DOI : 10.35291/2454-9150.2022.0495

ISSN : 2454-9150 Vol-08, Issue-09, Dec 2022

3.3.1 SD Interface with STM32F103

Figure 3.4- shows the interface logic between the STM32F103 and the SDC. The IRQ line, pin 8, is not connected. PA4 is connected to pin 1 which is card select. PA8 is used to detect if a card is inserted into the SD slot.

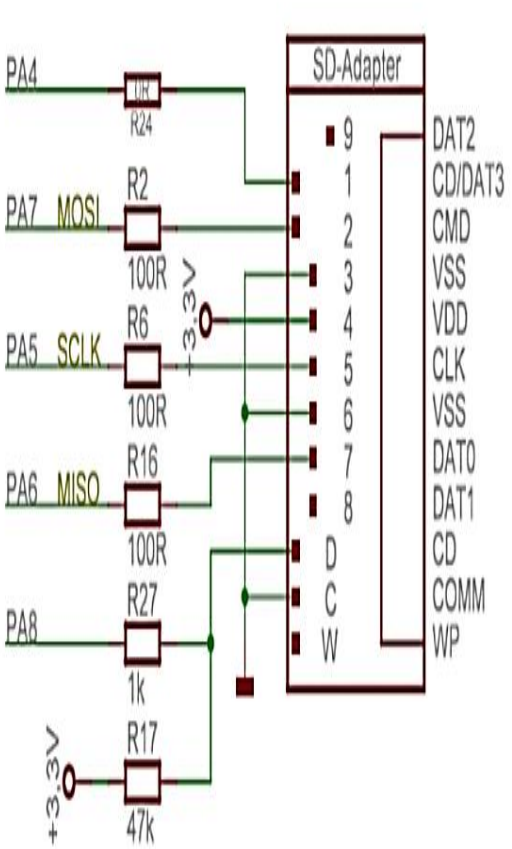

Figure 3.4. Schematic Diagram of SD Connection with STM32F103

3.3.2 ESP8266 Wi-Fi Module Interface for Internet Connectivity

For extra storage purposes, a MMC card is connected to actuator component of circuit. While it is suitable for small applications, but using IoT interfacing, huge amounts of data can be stored and analysed from a remote location.

Figure 3.5 represents another block diagram of data logger.

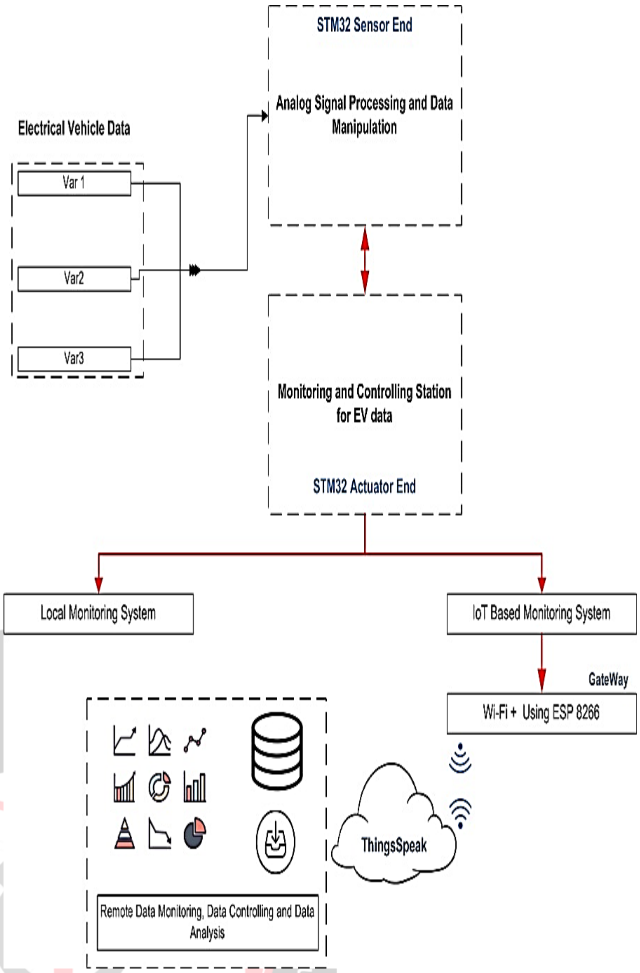

Figure 3.5. Block Diagram for Data Logger

Thing Speak is a platform providing various services targeted for building IoT applications. It offers the capabilities of real-time data collection, visualizing the collected data in the form of charts, ability to create plugins and apps for collaborating with web services, social network and other APIs.

The 16-digit API (Application Programming Interface) allows user to read from a private channel and write to a channel. The API key is entered in the code in IAR in function having esp8266 library.

3.4 MATHEMATICAL ANALYSIS AND CALCULATION

- Time Calculation for CANBUS
- Timer base Calculation

 DOI : 10.35291/2454-9150.2022.0495

International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-08, Issue-09, Dec 2022

Clock Rate is 32 MHz. This value is used at the first stage of Baud Rate Prescaler (BRP) where sample point is at 87.5%. Sync Jump Width (SJW) is 1 where it is small window of tolerance between CAN node baud rates.

Time Quanta (T_q)-Time quanta (t_q) is the basic unit of the bit time and its formula is –t_q=BRP/fsys

where,

BRP=Baud Rate Prescaler

fsys =MCU system Clock (typically =Fosc)

Duration of 1 time quanta (1 TQ) = PCLK1/CAN Pre-scaler = 0.03125 micro seconds

Bit Rate	accuracy	Pre-scaler	Number of time quanta	Seg 1 (Prop_Seg+Phase_Seg1)	Seg 2	Sample Point at	Register CAN_BTR
1000	0.0000	2	16	13	2	87.5	0x001c0001
1000	0.0000	4	8	6	1	87.5	0x00050003
800	0.0000	4	10	8	1	90.0	0x00070003
800	0.0000	5	8	6	1	87.5	0x00050004
500	0.0000	4	16	13	2	87.5	0x001c0003
500	0.0000	8	8	6	1	87.5	0x00050007
250	0.0000	8	16	13	2	87.5	0x001c0007
250	0.0000	16	8	6	1	87.5	0x0005000f
125	0.0000	16	16	13	2	87.5	0x001c000f
125	0.0000	32	8	6	1	87.5	0x0005001f
100	0.0000	20	16	13	2	87.5	0x001c0013
100	0.0000	32	10	8	1	90.0	0x0007001f
100	0.0000	40	8	6	1	87.5	0x00050027
83.333	0.0000	24	16	13	2	87.5	0x001c0017
83.333	0.0000	32	12	10	1	91.7	0x0009001f
83.333	0.0000	48	8	6	1	87.5	0x0005002f
50	0.0000	40	16	13	2	87.5	0x001c0027
50	0.0000	64	10	8	1	90.0	0x0007003f
50	0.0000	80	8	6	1	87.5	0x0005004f
20	0.0000	100	16	13	2	87.5	0x001c0063
20	0.0000	160	10	8	1	90.0	0x0007009f
20	0.0000	200	8	6	1	87.5	0x000500c7
10	0.0000	200	16	13	2	87.5	0x001c00c7
10	0.0000	320	10	8	1	90.0	0x0007013f
10	0.0000	400	8	6	1	87.5	0x0005018f

Figure 3.6. Design Calculation

Figure 3.7 shows the graph between clock period and time.

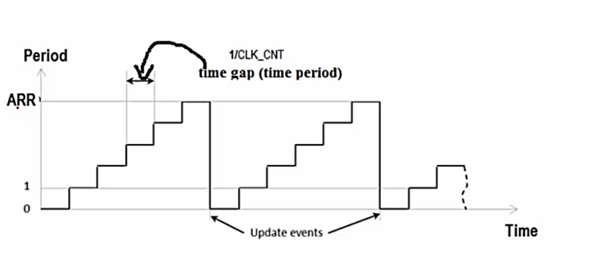

Figure 3.7. CAN Timing Sequence

4.1 SIMULATION RESULTS

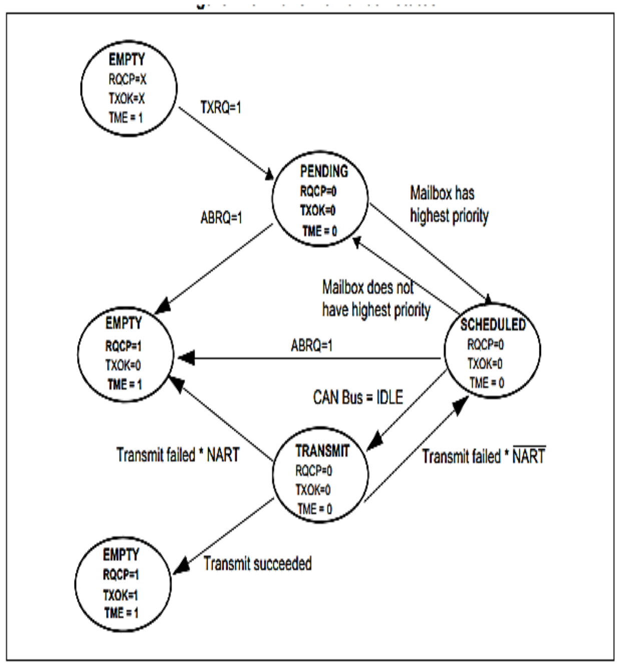

Figure 4.1 Transmit Mailbox States

- When SLEEP=0 and INRQ =0 then CAN comes into initialisation mode.
- Data is transmitted which is seen by bits TEC and LEC.LEC =0x03 - which is an acknowledgment error.
- When TXOK =1, TME = 1 and RQCP = 1 then the transmission is successful.

 DOI : 10.35291/2454-9150.2022.0495

ISSN : 2454-9150 Vol-08, Issue-09, Dec 2022

The received data is checked in CAN registers and results are verified. To receive data filters are added in code.

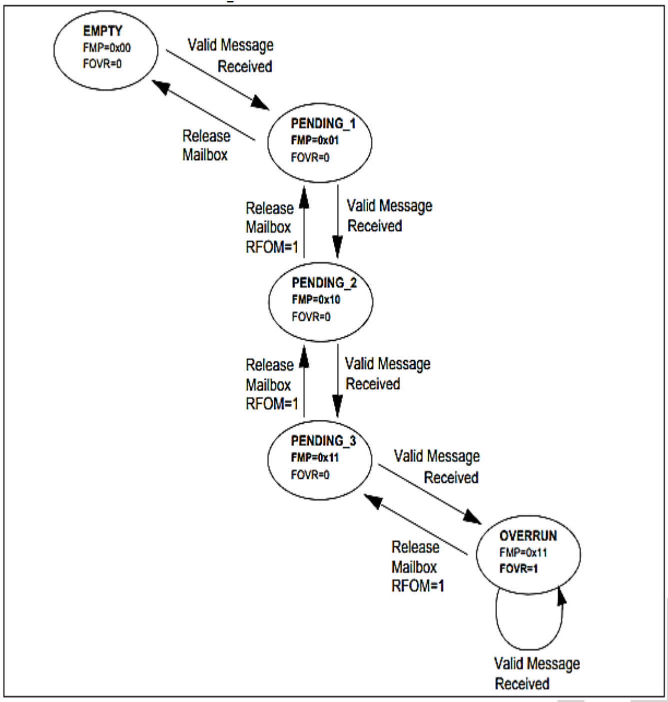

Figure 4.4. Receive Fifo States

If FMP =0x01 and FOVR = 0 valid message is received.

4.2 PRACTICAL RESULTS

The hardware connections are represented in Figure 4.5 and the transmitted signal is checked on digital storage oscilloscope (DSO).CAN Signal transmitted checked on DSO. The transmission is successful as values transmitted match on DSO.

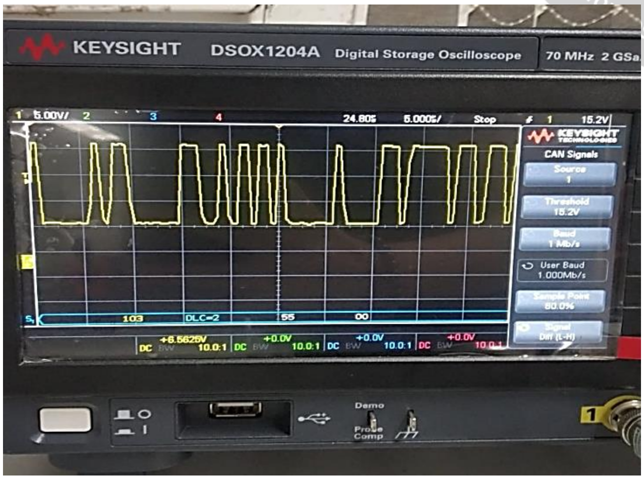

Figure 4.6. DSO Signal of Transmitted Message

5.1 CONCLUSION

This project proposes development of Data logger for Electric Vehicle. CAN is a robust vehicle bus standard designed to allow microcontrollers and devices to communicate with each other's applications without a host computer. The data is transmitted from one node to another by giving instructions to the microcontroller by the code written in IAR IDE in Hardware Abstraction Layer. Calculations were performed to find the best CAN bit rate timing, so that code gets transmitted successfully. To check if the status of microcontroller is made from sleep mode to initialisation mode and from initialisation mode to normal mode, the values in CAN register are checked to verify the results. Once that confirmation is received, the output is checked on DSO. The proposed model is able to transmit and receive data on CAN BUS and display it on Digital Scope Oscillator.

5.2 FUTURE WORK

The proposed work can be improved and envisioned as a small portable device to carry around to test and collect data from sensors. The data logger may also be built in a way to withstand high vibration thus perfect for any vehicle data logging.

REFERENCES

[1] Basu, A. K., Tatiya, S., & Bhattacharya, S. (2019). Overview of electric vehicles (EVs) and EV sensors. In *Sensors for Automotive and Aerospace Applications* (pp. 107-122). Springer, Singapore.

[2] Cheng, K. W. E. (2009, May). Recent development on electric vehicles. In 2009 3rd International Conference on Power Electronics Systems and Applications (PESA) (pp. 1-5). IEEE.

[3] Nagar, I., Rajesh, M., & Manitha, P. V. (2017, March). A low cost energy usage recording and billing system for electric vehicle. In 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 382-384). IEEE.

[4] Muratori, M., Alexander, M., Arent, D., Bazilian, M., Cazzola, P., Dede, E. M., ... & Ward, J. (2021). The rise of electric vehicles—2020 status and future expectations. *Progress in Energy*, *3*(2), 022002.

 DOI : 10.35291/2454-9150.2022.0495

[5] Svendsen, M., Winther-Jensen, M., Pedersen, A. B., Andersen, P. B., & Sørensen, T. M. (2014, December). Electric vehicle data acquisition system. In *2014) Engineering Applications* (pp. 363-374). Springer, Cham. *IEEE International Electric Vehicle Conference (IEVC* (pp. 1-7). IEEE.

[6] Padhee, S., & Singh, Y. (2011, January). Data logging and supervisory control of process using LabVIEW. In *IEEE Technology Students' Symposium* (pp. 329-334). IEEE.

[7] Okwudibe, C. D., & Akinloye, B. O. (2017). Design and simulation of temperature data logger. *American Journal of Engineering Research (AJER)*, *6*(12), 14-19.

[8] Fuentes, M., Vivar, M., Burgos, J. M., Aguilera, J., & Vacas, J. A. (2014). Design of an accurate, low-cost autonomous data logger for PV system monitoring using Arduino™ that complies with IEC standards. *Solar Energy materials and solar cells*, *130*, 529-543.

[9] Alves, J., Baptista, P. C., Gonçalves, G. A., & Duarte, G. O. (2016). Indirect methodologies to estimate energy use in vehicles: Application to battery electric vehicles. *Energy conversion and management*, *124*, 116-129.

[10] Nhivekar, G. S., & Mudholker, R. R. (2011). Data Logger and Remote Monitoring System for Multiple Parameter Measurement Applications. *e-Journal of Science & Technology*, *6*(3).

[11] Krogh, B., Andersen, O., & Torp, K. (2015, April). Analyzing electric vehicle energy consumption using very large data sets. In *International Conference on Database Systems for Advanced Applications* (pp. 471-487). Springer, Cham.

[12] Sehgal, V. K., Chauhan, D. S., & Sharma, R. (2008, November). Smart wireless temperature data logger using IEEE 802.15. 4/ZigBee protocol. In *TENCON 2008-2008 IEEE Region 10 Conference* (pp. 1-6). IEEE.

[13] Dey, D., Parida, A., Yamdagni, S., Sona, D. R., & Anand, N. Implementation of a cost-effective data acquisition system using STM32 and can architecture for electronic differential", *International Journal of Mechanical Engineering and Technology*, Vol. 9(2), pp. 417-431, 2018.

[14] Smuts, M., Scholtz, B., & Wesson, J. (2018, September). Issues in implementing a data integration platform for electric vehicles using the Internet of Things. In *IFIP International Internet of Things Conference* (pp. 160-177). Springer, Cham.

[15] Echavarría, S., Mejía-Gutiérrez, R., & Montoya, A. (2020, October). Development of an IoT platform for monitoring electric vehicle behaviour. In *Workshop on*

 DOI : 10.35291/2454-9150.2022.0495