ISSN 2347 - 3983
Volume 8. No. 3, March 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter45832020.pdf
https://doi.org/10.30534/ijeter/2020/45832020

A Hybridised Heterogeneous Embedded System Networking through Multi-Master Interface

E. Mounika[1], Dr. JKR Sastry[2], J. Rajasekhar[3]
[1]Koneru Lakshmaiah Education foundation, Vaddeswaram, mounikavvuradi@gmail.com
[2]Koneru Lakshmaiah Education foundation, Vaddeswaram, drsastry@kluniversity.in
[3]Koneru Lakshmaiah Education foundation, Vaddeswaram, rajasekharemb@kluniversity.in

ABSTRACT

Many Embedded system based implementations are forging forward in implementing hybridised networks that include a combination of subset of networks built using the Communication systems that include I^2C, CAN, USB, and RS485. Use of hybridised embedded networks is becoming most wanted necessity due to ongoing technology enhancements and the need for system integration.

Many Variations Exits among the heterogeneous embedded networks, which include protocol differences, speed variations, type of data packets etc. All these aspects are to be considered for building a system integration model. The internetworking is undertaken through multiple means using Single Master Interface, Multi-Master Interface, A Bridge and a Universal Bus.

In this paper an architectural model is presented that details system integration of heterogeneous Embedded Networks considering I2C and CAN based networking using Multi-Master Interface

Key words: Embedded Systems, Embedded Networks, Hybridisation of embedded Networks, Hybridisations through Multi-Master Communication

1. INTRODUCTION

Many kinds of embedded networks are in existence and in use for implementing different types of Applications. The Most important Embedded System networking being in use include the networks that were built using the communication systems that Include I2C, CAN, RS485, USB. But as they technologies are emerging, a necessity arises that require bridging the ES networks that were built around ES networking standards.

All ES networking standards differ in many ways that include network termination, device identification, Type and format of data packets, type of signals used, speeds of communication.

Hybridisation of Embedded Networks can be achieved through interconnecting different types of wired networks that include I2C, CAN, USB, and RS48. Serial communication takes place among the hybridised networks. Hybridisation can also be achieved through establishing wireless networks or through a combination of wireless and wired networks. The major issue in such kind of networking is the management of communication speeds and data rates. May architectures can be used for achieving the Hybridisation of the networks which include single master integrations, multi master integrations, use of hardware based bridge, mulimaster integration etc. The way a hybridised system works is dependent on the type of hybridisation method used. As on date Hybridisation through a Bridge device is proposed by [17]. Hybrdisation can be achieved through other methods that include Single master catering for the communication, using Multi master interface and by developing Universal bus that caters for most of ES communication Standards.

2. PROBLEM DEFINITION

The problem is to find a method using which Internetworking of heterogeneous embedded networks can be undertaken.

3. RELATED WORK

The speed of correspondence through I2C and RS232C get influenced because of the presence of complex electromagnetic condition as presence of such a domain will result into inappropriate disposal of the signals to the interfaces. Inappropriate transfer of electromagnetic will result into low speed and dependability. A few strategies are to be invented that help handling the issue of speed through legitimate cushion the executives [1].

Utilizing remote advances for the improvement of conveyed implanted frameworks is testing contrasted with wired systems because of presence of vulnerability and less unwavering quality that is caused because of attractive obstruction, blurring, reflection and so on. Consistency of remote correspondence is an issue. It turns out to be very unsafe for a framework when it needs to meet some security basic prerequisite by getting information through a remote system. The vulnerabilities existing with the remote correspondence can be settled through utilization of half breed models or undertaking hybridization of structural models. The idea of hybridization is increasingly common in the automotive area where everything must work consummately disregarding unsureness. Hybridization helps when the correspondence needs to occur in questionable circumstances. An engineering has been displayed that can be utilized to actualize applications requiring the idea of Hybridization. There are numerous interface gives that must be considered and took care of through a programming model which makes the framework hybridization-mindful [2].

Two gauges are as often as possible utilized which incorporate field bus and CAN bus for executing industry based applications, field bus benchmarks are not uniform and incredibly varies from industry to industry. Correspondence between the gadgets utilizing field bus in that capacity is entangled. In industry, both the transport based systems administration frameworks are often utilized. This has prompted a prerequisite of converting one kind of correspondence to the next which can be accomplished through convention transformation. Convention transformation can be planned and actualized at equipment level [3].

CAN transport based correspondence can be utilized for systems administration with frameworks. The engineers needs to comprehend the CAN convention, Interface, controller and Physical associations before the applications can be created utilizing CAN based correspondence. Actualizing CAN based correspondence at net root level is entangled and furthermore needs thorough testing. The advancement of utilizations utilizing CAN is made to be straightforward through CAN module. The CAN being a convention suite can be coordinated with any inserted framework Software. Sending and getting the information can be accomplished through CAN module [4]. When CAN is to be utilized alongside other correspondence conventions, for example, I^2C, protocol transformations can be actualized at work level rather than net root level. In the event that another module that modified works I^2C communication is grown, at that point convention transformation can be accomplished at work level.

A different gadget can be structured and built up that has the capacity of doing convention change utilizing numerous Microcontroller based frameworks, fast double port RAM information sharing innovation, and continuous multi-entrusting framework μC/OS-II. A convention converter that helps correspondence somewhere in the range of RS232C and RS485 has been built up that can be actualized inside savvy instruments, information securing frameworks, etc. [5].The gadget can be interfaced with a remote checking framework through Ethernet based correspondence. Along these lines, sequential gadgets can be connected to the system control layer. This gadget built up along these lines has high unwavering quality and continuous execution and acknowledge information trade, information sharing and data handling among various Micro controller based frameworks

Field bus convention is nonstandard and has been actualized in various forms. There is additionally an issue of interconnecting distinctive field buses. At the point when two systems are manufactured utilizing distinctive field bus correspondence convention transformation is required. ARM controller can be used for accomplishing the change. The convention transformation is accomplished through advancement of a protocol utilizing a standard information parcel. The strategy isn't constrained to coordinated transformation and is free of the area of the transport and the sort of convention utilized by the field buses [6].

A USB and I^2C convention varies in numerous angles considering the manner in which the correspondence is attempted. The information bundle groups, length of the system, number of gadgets that can be associated, number of ace, transport discretion, synchronization strategy, and stream control and so on varies a great deal. A mapping between I^2C and USB has been done both at the equipment and product level. The product structure that can be utilized for accomplishing the change has been displayed [7].

Modbus and Profibus are two correspondence frameworks utilized for accomplishing the modern mechanization. Both the specialized strategies are generally utilized in modern control field. Anyway, these two transports can't be associated straightforwardly because of presence of extraordinary fluctuation between them. An entryway is required for interfacing two unique transports through which convention transformation can be conveyed. A passage is created utilizing AT89C52 Microcontroller. Profibus and Modbus are two progressively basic mechanical field transport, they were generally utilized in modern control field. Since the two transports can't interconnect with one another, a Profibus and Modbus convention conversionjs required. SPC3 is incorporated with the Micro Controller to achieve Profibus and Modbus conversions [8].

Numerous kinds of sub-frameworks are to be created and actualized, utilizing distinctive correspondence frameworks. For example, s Flight control, banking, therapeutic, and other high affirmation frameworks which should be executed most unequivocally. Since correspondence framework utilizes distinctive signalling, sheathing, commotion separating, signal disconnection and so forth., one should structure and build up the framework to such an extent that one sub-framework doesn't meddle with the other.

Following a data stream at equipment level is one strategy that can be utilized to distinguish and channel the differences. Door level in-arrangement stream following (GLIFT) framework is built up that provides a technique for testing data streams that happens inside I^2C and USB. Time division various access (TDMA) has been utilized that can confine a gadget on the BUS from the streams [9].

Mechanical Ethernet innovation (EPA) and Modbus correspondence innovation (MODBUS) are every now and again utilized correspondence frameworks to actualize modern procedures. No immediate communication in that capacity can be conveyed in the middle of these two frameworks as there is no immediate similarity between them. A correspondence passage is created for accomplishing the necessary interface between these two innovations. A passage has been created utilizing ARM based smaller scale controller and μCOS continuous working system. Bidirectional correspondence can be accomplished through utilization of the entryway. The correspondence entryway can give a steady, secure, constant and adaptable answer for process control of the power plants [10].

Two kinds of industry explicit correspondence frameworks are utilized in the power segments which are intended for National power dispatching and for controlling the breeze factories. For giving the coordinated vitality the board framework theories correspondence frameworks must be interfaced with one another. A door has been produced for to accomplish convention transformation that executes capacities like interconnectivity, convention information type, and configuration change and scaling, information approval, the board of neighbourhood/remote directions, recreation of information parcels transmission and solicitations, communication bundles investigation and repetitive correspondence joins [11].

Field buses are utilized in industry for trade of information between several microcontrollers and field gadgets through affecting communication among them. Numerous adaptations of the field bus communication frameworks exist. Structuring a correspondence framework for impacting correspondence among the gadgets that keep diverse fieldbus correspondence benchmarks is mind boggling and by and large, prompts convoluted usage of equipment and programming. An effective correspondence interface isthe requirement for executing a solid framework utilizing restrained field bus correspondence norms. CAN transport is additionally being utilized nowadays for actualizing a significant number of the mechanical procedure. There is a need to interconnect among CANBUS and MODBUS. The correspondence conventions are to be mapped considering various parts of correspondence and afterward an interface is required to be created. An interface that between associates both the transport based correspondence frameworks has been introduced. [12][13].

The way networking of the embedded systems is carried depends on the kind of communication standard used I2C [14] , USB[15], CAN[16] and RS485[17] and also the kind of Microcontroller based systems used for networking.

Many issues related to networking have been discussed [18][19][20][21][22][23][24] with reference to testing Distributed Embedded Systems. Various Methods and strategies have proposed by Rajasekhar et al., [25] for hybridising the networking of heterogeneous embedded networks. The interconnection between a I2C network and a CAN network can be achieved by developing a device that bridges both the networks. Speed Matching is one of the most important consideration that must be handled **[26]**.

4. GAP

The main GAP is nonexistence of any model, Architecture or Framework using which the Bridging of heterogeneous Embedded Networks can be done.

5. INVESTIGATIONS AND FINDINGS

5.1 Application Development Using CAN Interface

The ES Application developed through CAN Interface is related to Monitoring the temperature within an Engine and Humidity % as measured within a motor vehicle. The application specific functions implemented through CAN based networking is shown in Table1.

Table 1: CAN application description

Hardware Device Number	Hardware description	Interface description	Functional Description
1	STM 32 F401 RE	CAN	To receive Temperature Data from Slave
			To receive Humidity Data from Slave
			To Send Temperature data to I2C Master
			To Send Humidity data to I2C Master
			To Receive Distance data from I2C Master
			To send distance data to slave
2	ARDUINO UNO	CAN	To Receive distance data from master
			To Control the Lighting System
3	ARDUINO UNO	CAN	To sense the temperature
			To sense the Humidity
			To send Temperature to Master
			To send Humidity to the Master

The Networking Diagram for the Application is shown in Figure 1.

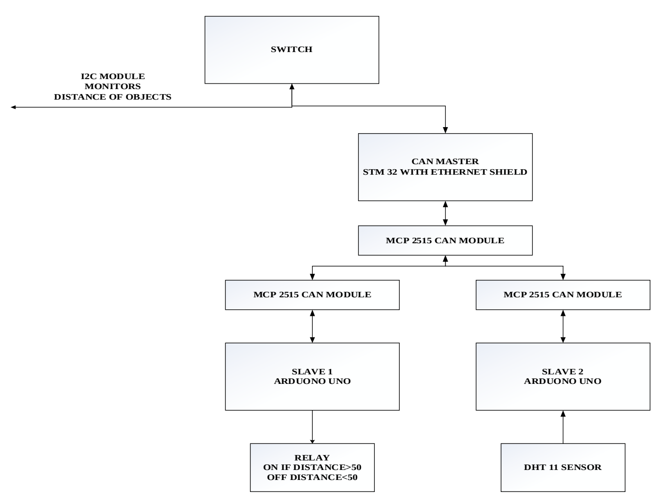

Figure 1:CAN Networking Diagram

CAN based networking is archived through one Master and two slaves. The slaves are implemented through Arduino-UNO and the Master is implemented through STEM 32, All the three Systems are connected through MCP 2515 CAN Module. The master is connected to a SWITCH through its native Ethernet port.

One of the slaves is connected with the DHT 11 sensor which is situated near the engine of an automobile system. The sensor continuously monitors the temperature of the engine and sends the information to the CAN master. CAN master will send the information to the I2C master via Ethernet. The I2C master sends the data to the I2C slave connected to a FAN. The FAN is controlled through a relay system. Whenever the temperature within the engine goes beyond a level, the FAN is switched on or otherwise switched off.

CAN or Controller Area Network is a two wired asynchronous, half-duplex fast sequential system innovation .Bus width of CAN is 217 bits .It is fundamentally utilized in correspondence among various gadgets in a low range distance for example, in a vehicle. A CAN convention is a CSMA-CD/ASM convention or transporter sense different access impact location mediation on message needs convention. CSMA guarantees every hub must hang tight for a given period before sending any message. The crash location guarantees that the impact is kept away from by choosing the messages dependent on their endorsed need. It gives a flagging rate from 125kbps to 1 Mbps. It accommodates 2048 diverse message identifiers.

CAN bus is the multi-master protocol. When the bus is idle any device can be attached to the CAN bus and starts messaging. The can bus is versatile so devices attached to the bus do not have addressing. Each device in CAN bus receives every message and it is up to the device whether to act according to the message or discard the message.

CAN bus provide remote transmission request (RTR), meaning that one node on the bus can request information from the other nodes So a request for information can be sent to the node instead of waiting for a node to send information continuously. Any device in CAN bus can identify the error that occurred on the bus while transmitting the data and generates the error frames. The node which identifies the error alerts all other nodes about the error. There are no limitations for attaching and detaching the devices to the CAN bus, so devices are easy to attach and detach. Depending on the bus delays time and electrical loads only we can decide the no of devices attached to the bus.

CAN protocol send messages in form of different types of data packets that include data frames, remote frames, error frames and over load frames shown Figure 2, Figure 3, Figure4, and Figure 5respectively. Data Frames are used to transmit data from Master to slave and vice versa. Remote frames are used to seek permission from other node to transmit messages, Error frames are used for transmitting the errors that occur due to transmission which can be classified as Biterrors, CRC errors, Form errors, Acknowledgement errors, and Stuffing errors. Overflow frames are used to create extra delay required for transmission of data or response.

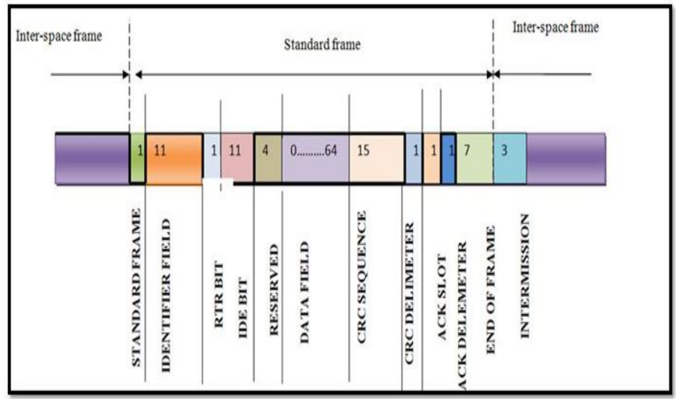

Figure 2: Data Frame

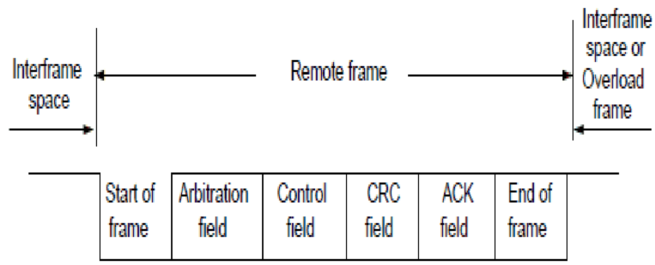

Figure 3: Remote Frames

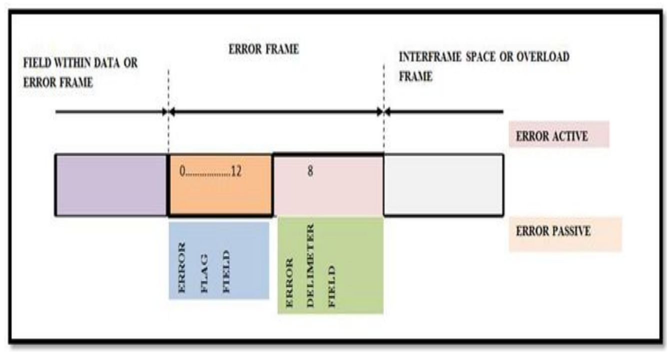

Figure 4: Error Frame

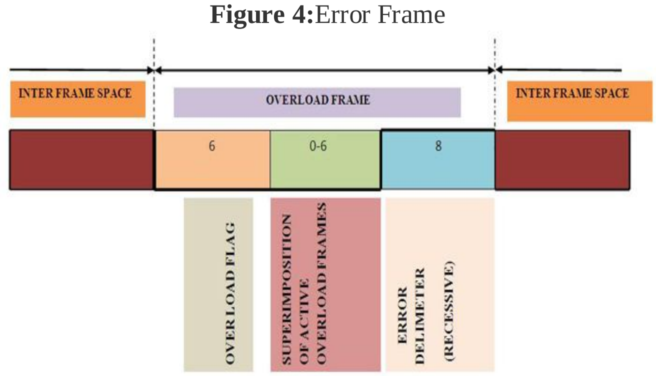

Figure 5: Error Frame

5.2 Application Development Using I²C Interface

The ES Application developed through I²C Interface is related to measuring the distance of the objects Located behind a motor Vehicle and also controlling the speed of a FAN fitted within the Engine based on the temperature existing in the Engine which is transmitted through CAN network. The application specific functions implemented through I²C based networking is shown in Table 2.

Table 2: I²C application descriptions

Hardware Device Number	Hardware description	Type of Device	Functional Description
1	**STM 32 F401 RE**	**Master**	To Receive Distance data from slave
			To receive Temperature data from CAN Master
			To receive Humidity data from CAN Master
			To send Distance data to CAN Master
			To send Temperature data to Slave
			To send Humidity data to slave
2	**STM 32 F301 RE**	**Slave**	To sense the distance of the object while reversing the car
			To send the Distance data to the master
3	**STM 32 F301 RE**	**Slave**	To receive Temperature data from the Master
			To receive Humidity data from Master
			To actuate the DC motor for controlling the FAN

The networking Diagram for the I²C based Application is shown in Figure 6

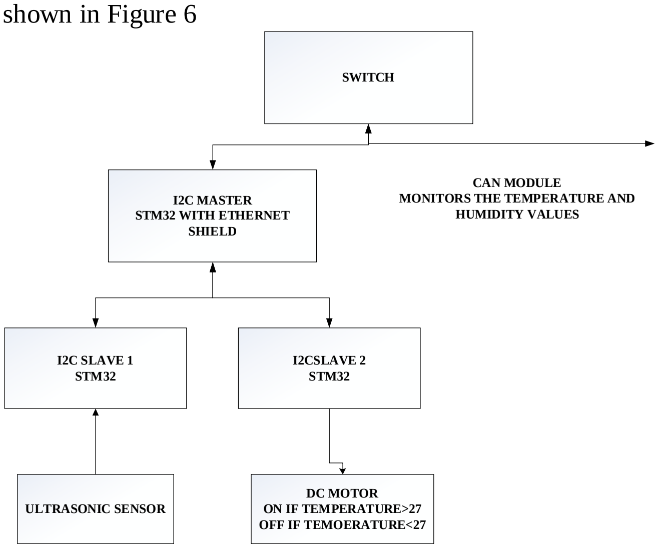

Figure 6: I2C Networking Diagram

I²C Network is built with a single master and two slaves. One slave is connected with an ultrasonic sensor which is used to monitor the distance of the nearer object to the car while the car is backing up. The monitored distances are sent to CAN master through Ethernet. The Second slave is connected with a DC Motor for controlling the speed of FAN which is connected to an engine. The controlling of the FAN is done based on the Temperature and Humidity data received from the CAN master. Two different protocols are used within I2C each for reading and writing data.

The following sequence of operations are carried when data transmitted by a slave is to be read by the Master

1. The master device sets the Read/Write bit to ‘1’ instead of ‘0’ which signals the targeted slave device that the master device is expecting data from it
2. The 8 bits corresponding to the data block are sent by the slave device and the ACK/NACK bit is set by the master device
3. Once the required data is received by the master device, it sends a NACK bit. Then the slave device stops sending data and releases the SDA line
4. If the master device to read data from specific internal location of a slave device, it first sends the location data to the slave device using the steps in previous scenario. It then starts the process of reading data with a repeated start condition.

The following sequence of operations takes place when a master device tries to send data to a particular slave device through I²C bus:

1. The master device sends the start condition
2. The master device sends the 7 address bits which corresponds to the slave device to be targeted
3. The master device sets the Read/Write bit to ‘0’, which signifies a write
4. Now two scenarios are possible:
5. If no slave device matches with the address sent by the master device, the next ACK/NACK bit stays at ‘1’ (default). This signals the master device that the slave device identification is unsuccessful. The master clock will end the current transaction by sending a Stop condition or a new Start condition
6. If a slave device exists with the same address as the one specified by the master device, the slave device sets the ACK/NACK bit to ‘0’, which signals the master device that a slave device is successfully targeted
7. If a slave device is successfully targeted, the master device now sends 8 bits of data which is only considered and received by the targeted slave device. This data means nothing to the remaining slave devices
8. If the data is successfully received by the slave device, it sets the ACK/NACK bit to ‘0’, which signals the master device to continue
9. The previous two steps are repeated until all the data is transferred
10. After all the data is sent to the slave device, the master device sends the Stop condition which signals all the slave devices that the current transaction has ended.

Data Frames

I^2C data is transferred in *messages*. Messages are broken into frames of data. Each message has an address frame that contains the binary address of the slave, and one or more data frames that contain the data being transmitted. The message also includes start and stop conditions, read/write bits, and ACK/NACK bits between each data frame. The format of the message used within a I2C system is shown in **Figure 7**.

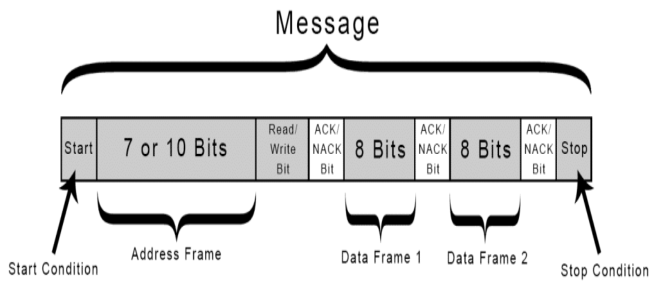

Figure 7: I^2C Message format

Start Condition is initiated by making theSDA line switched from a high voltage level to a low voltage level before the SCL line switches from high to low. The Stop condition is achieved through switching the SDA line from LOW voltage to HIGH voltage after SCL is switched from LOW to HIGH. The Bits 7-10 contain the address of the slave with which the master wants to communicate. The Read/Write Bit specify whether the master is sending data to the slave (low voltage level) or requesting data from it (high voltage level). The ACK/NACK Bit specify whether the master requires Acknowledgment from the slave or otherwise. If an address frame or data frame was successfully received, an ACK bit is returned to the sender from the receiving device

5.3 Comparison of Application Specific CAN and I^2C Networks

I^2C and CAN networks differs in many ways that include speeds, protocols used for transmission and reception of the data, addressing the devices within the networks, the data frames used for transmission and reception of the data, error control implemented etc. Making a device as slave to both the networks is cumbersome. There can be many ways of interconnecting both the networks which includes connectivity through single Master, Connectivity through Multiple Masters, Connectivity through a Bridge and by implementing a Universal Bus.

5.4 Interconnecting between CAN and I2C Networks through Multi-Master Interface

The way the interconnection between the I^2C network and CAN network is achieved through interfacing the MASTERS using an Ethernet Interface is shown in Figure8.

The Master on the I^2C network has both the I^2C and Ethernet Interface and similarly the master on the CAN network has both the CAN and Ethernet Interface. Whenever data from an I2C slave is to be transmitted to a CAN salve, the data is first transmitted to the I2C master using I2C protocol. The data packets are de-assembled and then assembled into Ethernet packets. The Ethernet packets are then transmitted to the CAN master through peer to peer connection established through Ethernet. The Ethernet packets are received by the CAN master and the packets are dissembled and assembled into CAN packets which are then transmitted the CAN slave. The process of transmission from a CAN slave to the I2C slave takes place in similar manner.

The entire process of communication involves the selection of the proper speeds considering I^2C, CAN and Ethernet communication system such that the communication is completed with the acceptable response time. The delay caused due to de-assembling and assembling of the packets must also be taken into account while calculating the response time.

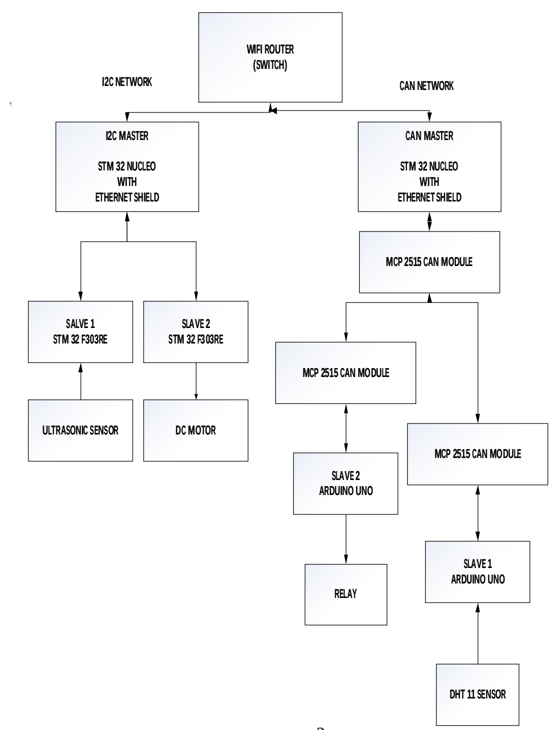

Figure 8: Interconnecting I^2C and CAN networks through Ethernet Interface

5.5 Architecture for Hybridised communication in between I^2C and CAN networks

The architecture for establishing communication among I^2C and CAN networks through Ethernet based Multi master Interface is shown in Figure 9. The most important issue is the arbitration among the I2C master and CAN master on the kind of speeds that must be used for effecting communications intra I2C, intra CAN and in between the masters using internet Interface. Speed matching is absolutely necessary so that the required response time is met. There should be time allowance for assembling and de-assembling the packets of different types. The software components contained within the slaves and the Masters shall carry designated functions as shown in the Architectural Diagram. The masters agree on speed based on the amount of data to be submitted considering the type of packets and the size of the packets to be transmitted and also considering the amount of delay time caused due to de-assembling and assembling processes. Once the speed agreements are achieved, the master shall communicate the agreed speeds to the respective slaves to follow the speeds especially helps in setting the slaves internal timers.

6. CONCLUSION

Interconnecting different heterogeneous embedded systems requires specific mechanism. Interconnecting the masters of I2C and CAN through Ethernet based peer to peer connectivity is one of the solution for achieving communication among heterogeneous ES networks. The Most important issue is deciding on the speeds with which communication must be affected considering I2C, CAN and Ethernet. The speeds must be decided based on the amount of data to be transmitted either way between CAN and I2C. Allowance to the delay caused due to the assembling and de-assembling must also be made.

REFERENCES

1. CHAI Yan-jie, SUN Ji-yin, GAO Jing,TAO Ling-jiao, JI Jing, BAO Fei-hu," Improvement of I2C Bus and RS-232 Serial Port un-der Complex Electromagnetic Environment" International Confer-ence on Computer Science and Software Engineering, 2008, PP 178 - 181. https://doi.org/10.1109/CSSE.2008.843
2. Antonio Casimiro, Jose Rufino, Luis Marques, Mario Calha, and Paulo Verissimo "Applying Architectural Hybridization in Net-worked Embedded Systems", IFIP International Federation for In-development Processing, 2009, PP 264-275.
3. Lou Guohuan,ZhangHao, Zhao Wei,"Research on Designing Method ofCAN Bus and Modbus Protocol Conversion Interface" International Conference on Future Bio Medical Information Engi-neering,2009, PP 180-182. https://doi.org/10.1109/FBIE.2009.5405874
4. Xiaoming Li, Mingxiong Li, "An Embedded CAN-BUS Communication Module for Measurement and Control System" International con-ference on ICEEE, 10.119/ICEEE.2010.5661248, 2010
5. Peng Daogang, Zhang Hao, Li Hui, Xia Fei," Development of the Communication Protocol Conversion Equipment Based on Embed-ded Multi-MCU and μC/OS-II", International Conference on Measuring Technology and Mechatronics Automation, 2010, PP 15-18. https://doi.org/10.1109/ICMTMA.2010.195
6. Tae-Won Kim, Jin Ho Kim, Do Eon Lee, Jun young Moon, Jae Wook Jeon," Development of Gateway based on BroadR-Reach for Application in Automation Network", International Conference on Computing, Communication and Automation, 2016, PP 421-426.
7. CHEN Huijuan," Heterogeneous Network Integration Based on Protocol Conversion" Control Conference (CCC), 2016, PP 6888-6893.
8. Chen Wei,WangXijun,SunWenxia," The Design of profinet- modbus protocol conversion Gateway Based on the ERTEC 200P", International Conference on Software, Knowledge, Information Management & Applications, 2016, PP 87-91.
9. Jason Oberg,WeiHu,AliIrturk," Information Flow Isolation in I2C and USB", Design Automation Conference, 2011, PP 254-259. https://doi.org/10.1145/2024724.2024782
10. Li Hui, Zhang Hao, Peng Daogang,"Design and Application of Communication Gateway of EPA and MODBUS on Electric Power System" International Conference on Future Electrical Power and Energy Systems, 2012, PP 286 – 292.
11. Gheorghe G.Florea,OanaI, Rohat, Monica G. Dragan, "Contributions to the Development of Communication Protocols Conversion Equipment", 2nd IFAC Workshop on Convergence of Information Technologies and Control Methods with Power Systems, 2013, VOLUME 46, Issue 6, PP 84-88.
12. Umesh Goyal, Gaurav Khurana," Implementing MOD bus and CAN bus Protocol Conversion Interface" IJETT, 2013, VOLUME 4, Issue 4, PP 630-635.
13. Roopak Sinha," Conversing at Many Layers: Multi-layer System-On-chip Protocol Conversion", 20th International Conference on Engineering of Complex Computer Systems, 2015, PP 170-173. https://doi.org/10.1109/ICECCS.2015.25
14. JKRSastry, J. Viswanadh Ganesh, and J. SasiBhanu, "I^{2}C based Networking for Implementing Heterogeneous Microcontroller based Distributed Embedded Systems", Indian Journal of Science and Technology, 2015, Vol 8(15), PP 1-10.
15. JKRSastry, Valluru Sai Kumar Reddy, Smt J SasiBhanu, " Net-working Heterogeneous Microcontroller based Systems through Universal Serial Bus", IJECE, 2015, Vol. 5, No. 5, PP. 992-1002.
16. JKR. Sastry, M. Vijaya Lakshmi and Smt J. SasiBhanu," Optimizing Communication Between Heterogeneous Distributed Embedded Systems Using CAN Protocol", ARPN Journal of Engineering and Applied Sciences, 2015, VOL. 10, NO. 18, PP 7900-7911.
17. JKRSastry, T. Naga Sai Tejasvi and J. Aparna, Dynamic scheduling of message flow within a distributed embedded system connected through RS485 network, ARPN Journal of Engineering and Applied Sciences, VOL. 12, NO. 9, MAY 2017
18. J. K. R. Sastry, A. Suresh, and Smt J. SasiBhanu, Building Heterogeneous Distributed Embedded Systems through RS485 Communication Protocol, ARPN Journal

of Engineering and Applied Sciences, issue. 16, vol.10, 2015

19. K. Chaitanya, SastryJKR, K. N. Sravani, D. Pavani, Ramya and K. Rajasekhara Rao, Testing Distributed Embedded Systems Using Assert Macros, ARPN Journal of Engineering and Applied Sciences, 2017, page no.3011-3021
20. SastryJKR, K. Chaitanya, K. Rajasekhara Rao, DBK Kamesh, Testing Distributed Embedded Systems Through Instruction Set Simulators, PONTE, International Journal of Sciences and Research, issue.7, vol.73, July 2017, page no.353-382 https://doi.org/10.21506/j.ponte.2017.7.57
21. JKRSastry, K. Chaitanya, K. Rajasekhara Rao, DBK Kamesh, An Efficient Method for Testing Distributed Embedded Systems using In-circuit Emulators, PONTE, International Journal of Sciences and Research, issue.7, vol.73, 2017, page no.390-422
22. K. Chaitanya, JKRSastry, K. Rajasekhara Rao, Testing Distributed Embedded Systems Using Logic, Analyzer, International Journal of Engineering and Technology, March 2018, page no. 297-302 https://doi.org/10.14419/ijet.v7i2.7.10601
23. K Chaitanya1, Dr. K Rajasekhra Rao, Dr. JKRSastry, A Framework for Testing Distributed Embedded Systems, International Journal advanced Trends in computer science and engineering, 2019, Volume 8, No.4, PP. 1104-1227
24. K Chaitanya, Dr. K Rajasekhra Rao, Dr. JKRSastry. A Formal and Enriched Framework for Testing Distributed Embedded Systems, International Journal Emerging Trends in Engineering Research, Volume 7, No. 12, 2019, PP. 867-878 https://doi.org/10.30534/ijeter/2019/247122019
25. J. Rajasekhar, Dr. JKRSastry, An approach to hybridization of embedded system networks, International Journal of Engineering & Technology, 7 (2.7) (2018) 384-389
26. Jammalamadaka Rajasekhar, JKRSastry, Building Composite Embedded Systems Based Networks Through Hybridization and Bridging I2C and CAN, Journal of Engineering Science and Technology, Vol. 15, No. 2 (2020) 858 - 881

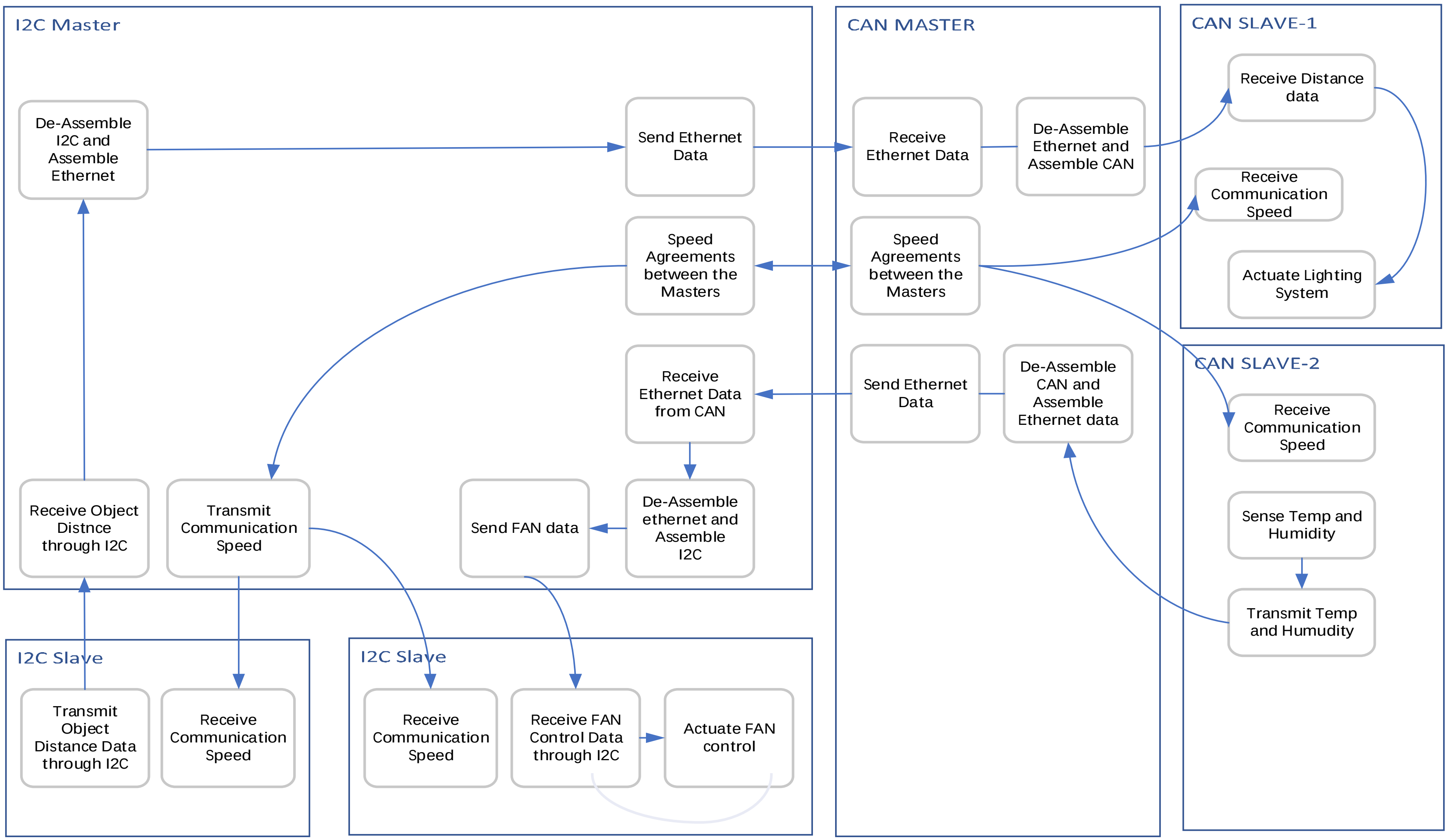

Figure 9: Architecture for Implementing MULTI-MASTER Ethernet based Interface for Interconnecting I2C and CAN Networks