Design of Inter-Networking Gateway for Message Transmission to the Ethernet for Elevator Monitoring

Ajit Pedha, Nivash.S

Abstract—**Recently, with the increasing use of elevators and the increasing height of the multi-storey buildings, there must be a system developed which continuously monitors the status of the elevators and reports for any problem to the authorities. There can be a monitoring system developed which uses the inter-networking gateway for transmission of messages to the authorities enquiring about the current status of the elevators. There must be a protocol developed for the transmission of CAN messages to be received by the Ethernet. This paper discusses a unique way of elevator monitoring and also the inter-networking gateway required to be developed for the CAN messages to be transmitted to the Ethernet.**

Index Terms—**Elevator, Inter-networking, CAN, Ethernet.**

I. Introduction

Elevator is most commonly used as a means of transportation for going up or down in multi-storey building or any other commercial buildings and is irreplaceable. If an elevator is present in a building then people tend to use it but sometimes there are some accidents concerned with the elevators which happens out of the negligence so the condition of the elevator must be maintained properly and the status or current status must be known for maintenance and it must be seen that no accidents occur. So some system must be designed in such a way that oversees the condition of the elevator and sends the status of the elevator over a periodic time to the maintenance department who looks after the condition. In previous system, the condition of the elevator is obtained but the monitoring personal could not get any data about the status of the elevators. Now there may be more than one elevators in a commercial area or a building. The maintenance department checks the condition of the elevators once in a while but sometimes due to their negligence some accidents happen. Various elevator monitoring systems areavailable nowadays but it is necessary to develop a smart and networked system which is more reliable. So a system designed which shares the status of the elevator to a maintenance department.CAN (Controller Area Network) Bus is important for serial communication, collecting sensor data and transmission of that data. CAN has many features such as speedy communication and long distance communication. CAN adopts multi-master mode of communication. Each controller attached to each node synchronises to the timing cycle of the CAN network. After the CAN bus acquires the data, it sends the data to the Ethernet. TCP/IP communication protocol is most commonly used nowadays for transmission of the data on the Ethernet. In summary the CAN bus collects the data from all the elevators and transmits this data over to the Ethernet. When the monitoring personal receive the data from the Ethernet, the data of the different elevators is displayed in a form understandable by the personal. If any elevator shows any signs of faults, error or not performing according to the ideal conditions then the monitoring personal can understand. This project focuses on the design of a protocol for the connection of the CAN Bus and the Ethernet. The synchronization of data is one of the most important parameters needed in the connection of the CAN and the Ethernet. It is called as internetworking gateway.

The status of the elevators is checked and if any error occurs then it sends a message to the personal of the building and they have to send the error to the monitoring personal who will pay a visit but the elevator will be shutdown till the maintenance is done. The new system proposed in this paper in which the status of the elevator such as power consumption, errors are directly sent to the monitoring personal who continuously check the status and if any value goes abnormal then the monitoring personal will inform and repair the elevator before any major problem occurs.

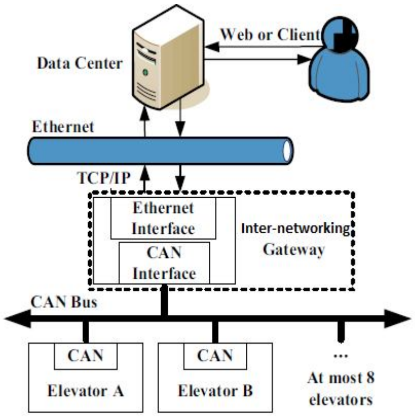


Fig.1 The structure of overall diagram

Manuscript received March 25, 2015.

Mr.Ajit Pedha, Department of Electronics and Communication Engineering (Embedded System Technology), SRM University, Chennai, India.

Mr. S. Nivash, Department of Electronics and Communication Engineering, SRM University, Chennai, India.

II. INTER-NETWORKING HARDWARE DESIGN

This paper is designed with a focus of the developing the gateway for the communication of t CAN protocol and the TCP/IP stack. It is necessary to understand the hardware requirements which lay the foundations of the software requirements.

A. LPC2129 Microcontroller

LPC2129 is 32-bit Micro-controller which includes 256KB on-chip Flash ROM with In-System Programming (ISP), 16 KB RAM,Two UARTs, I2C serial interface, 2 SPI serial interfaces, PWM unit with up to 6 PWM outputs, 2 CAN channels. It can operate clock up to 60 MHz, it also contains On-chip crystal oscillator and an On-chip PLL. As it contains CAN interface external CAN transceiver is not needed.

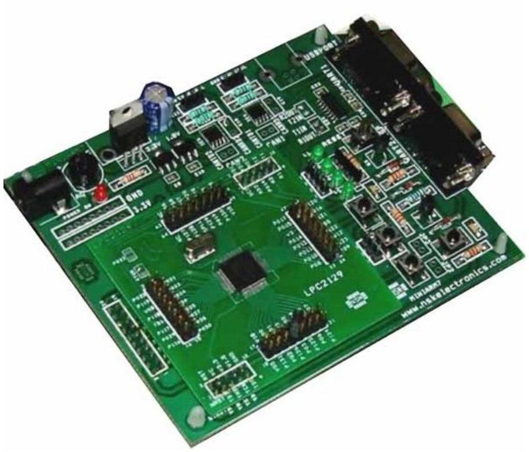


Fig.2 LPC2129 model

B. Raspberry Pi

Raspberry Pi(Model B) is an ARM11 using an ARMv6-architecture core with floating point, running at 700Mhz,a Video core 4 GPU and a Broadcom BCM2835 SOC. It has internal 512MB RAM. 2 USB ports, an Ethernet port.

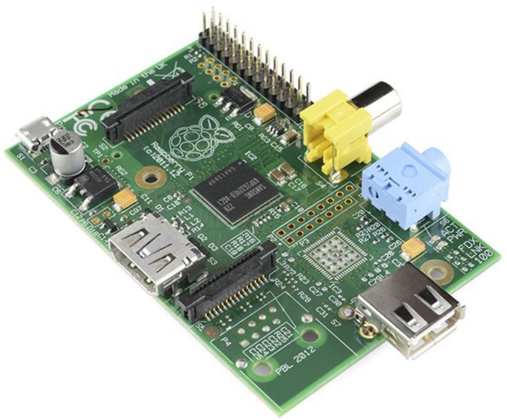


Fig.3 Raspberry Pi B model

C. CAN Transceiver

The MCP2551 is a high-speed CAN transceiver, the MCP2551 provides differential transmit and receive capability for the CAN protocol. It is the interface between micro-controller and CAN physical bus and is fully compatible with the ISO-11898 standard, including 24V requirements. Its operating speed is upto 1 Mb/s.

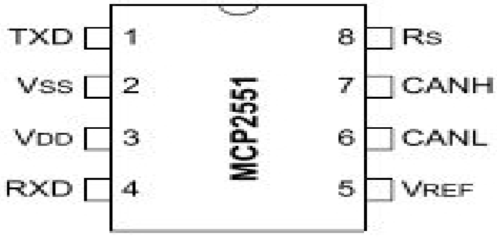


Fig.3 MCP2551 Pin-out diagram

D. CAN Controller

MCP2515 is used as a CAN controller. It converts the data received from the LPC2129 to the CAN data frame format.

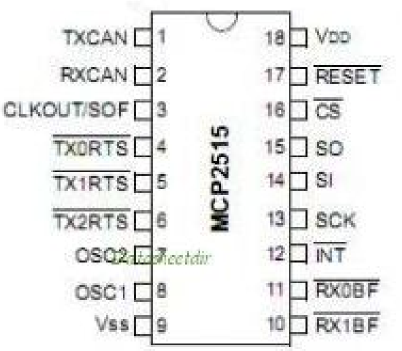


Fig.4 MCP2515 pin-out diagram

E. Overall Hardware Design

LPC2129 is a controller which is attached to the nodes of the elevators which collects the sensor values from the elevator and using its CAN interface on board it sends the data to the Raspberry Pi BROADCOM 2835 SOC. MCP2551 is used as a CAN transceiver for communication between microcontroller and Raspberry Pi. The direct CAN communication between LPC2129 and Raspberry Pi is not possible so a CAN Transceiver is used for communication between them. Normally the CAN Transceiver collects the data from the LPC2129 boards using SPI protocol and sends the data to the Raspberry Pi using another SPI protocol. The speciality of Raspberry Pi is that it directly converts the data bits from CAN to the TCP/IP format. So LPC2129 collects the data from the nodes and sends the data to the Raspberry Pi via the CAN bus. The Raspberry Pi sends the data over the Ethernet port for access through the internet.

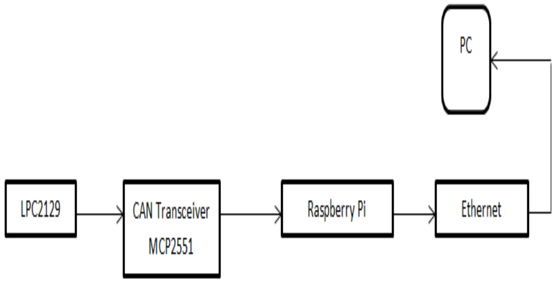


Fig.5 Project flow diagram

III. GATEWAY SOFTWARE DESIGN

A. *TCP/IP protocol*

Transmission Control Protocol / Internet Protocol (TCP/IP) is a set of protocol that defines how computers can communicate with each other using the internet. It is an open standard which can be implemented on any computer with the appropriate physical attributes. A network is basically a connection of computers or other hardware devices that are connected together and exchange data between them. These provide different functionality important to the exchange of data over the networks. Many operations such as the Domain Name System or the application that uses the network such as E-mail use TCP/IP.

Another related protocol is UDP (User Datagram Protocol) which also runs on top of the IP (Internet Protocol). The difference between TCP and UDP is that TCP is connection based protocol whereas UDP is connectionless. In TCP there needs to be a session setup between the hosts and the transfer for connection while in UDP each data packet is sent but there is no checking that it has been received, in short there is no acknowledgement or any method of resending within the network layers. Any application can have its own method of checking the receiving of the packet but there is no checking done by the networking stack. A common example of comparing these is to like comparing TCP to the telephone system and UDP to the postal service. With the telephone when you establish a connection with the other person, you know for certain that the user receives the message. If you were disconnected during the telephone conversation then you would know about it and be able to phone the other person again. With the postal system after you post the letter then you do not know for certain whether or not the mail will be received. After you have posted the letter it could be lost or destroyed on its way to its destination. Or if the person has moved house they may neverreceive the letter. From this discussion it may look like there is no reason to choose UDP over TCP but the only disadvantage with TCP is that, it has lot of overhead. For each data packet being sent a confirmation has to be generated and even if there is no data being sent there will often be some kind of keep alive signal. Whereas for some less important data you may just want to send and forget it with the hope it will reach the other end. It's also possible for the session to be handled higher up the networking stack.

The reliability of the internet connection between the data and the gateway is maintained by sending a heartbeat packet or a simple information indication that ‘I am online’ so that the connection is not lost.

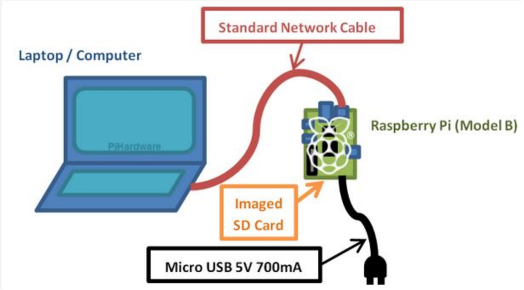
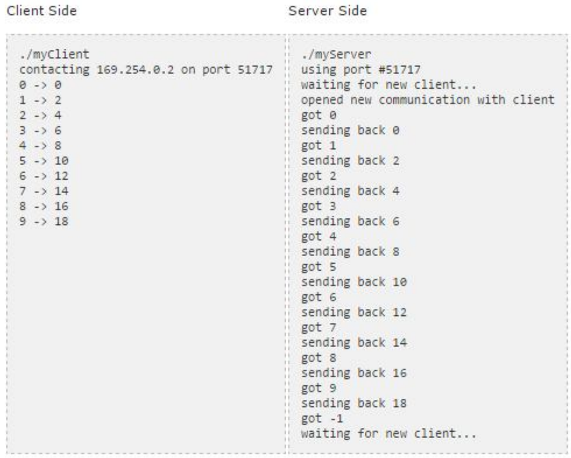


Fig.6 Connection of Raspberry Pi to the Ethernet

The above setup is required for the communication between the Raspberry Pi and the Ethernet. The following setting will be required for configuration of Ethernet. The Raspberry Pi runs a server that waits for connection from a laptop, and expects integers from it. It multiplies each integer by 2 and sends it back. The laptop runs a client that initiates a connection, sends a bunch of positive integers that it gets back multiplied by two, and closes the connection by sending a -1. Sending a -2 causes the server to stop.

Raspberry Pi is used as an Ethernet client to the web server. A C language code is written to communicate Ethernet client and web server. The sensor values of elevator will be transmitted through the Raspberry Pi Gateway. The data is sent in the heartbeat type manner that is the data is sent at a particular interval.



Client Side

```
./myClient
contacting 169.254.0.2 on port 51717
0 -> 0
1 -> 2
2 -> 4
3 -> 6
4 -> 8
5 -> 10
6 -> 12
7 -> 14
8 -> 16
9 -> 18
```

Server Side

```
./myServer
using port #51717
waiting for new client...
opened new communication with client
got 0
sending back 0
got 1
sending back 2
got 2
sending back 4
got 3
sending back 6
got 4
sending back 8
got 5
sending back 10
got 6
sending back 12
got 7
sending back 14
got 8
sending back 16
got 9
sending back 18
got -1
waiting for new client...
```

Fig 7 Example of client to server network

IV. DISPLAYING OUTPUT ON THE PC

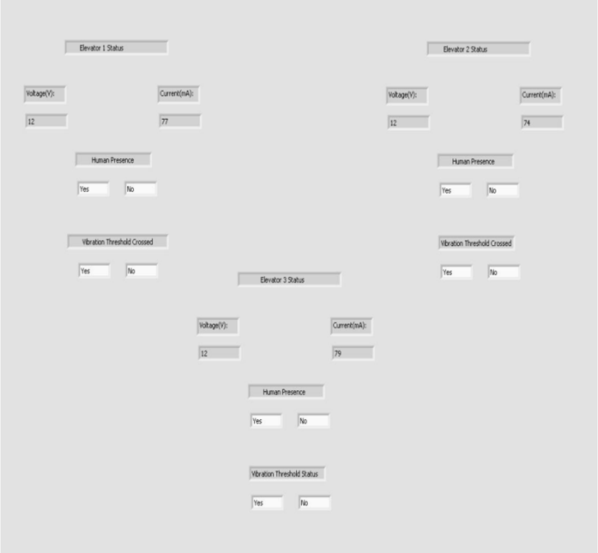
The monitoring personal must be able to receive the output in such a way that is understandable to them and also easy to detect the faults that are occurring. So LABVIEW software is used for displaying the output.

A. *LABVIEW*

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) form National Instruments is a system-design platform and development environment. LabVIEW is a development environment for a graphical programming language LabVIEW is an ideal platform for prototyping, designing, and deploying high-quality products to market fast. You can use one development environment to quickly iterate on your embedded hardware and software designs and then reuse the best parts in a final product. With limited resources and time LabVIEW can be used to quickly design and prototype the custom software embedded software application. LabVIEW can span your entire hardware array and across multiple instrumentation types, ranging from data acquisition devices to field-programmable gate arrays (FPGAs). It can be used for developing environment to quickly iterate on your embedded hardware and software designs and then reuse the best parts in a final product.

V. RESULTS

The sensor values from the LPC2129 microcontroller nodes will be sent to the CAN bus via the CAN transceivers and that data will be sent to the Raspberry Pi via another CAN transceiver to the Ethernet port where the data is received on PC and is displayed on LabVIEW, a software embedded application for maintenance.



Elevator 1 Status

Voltage(V): 12

Current(mA): 77

Human Presence

Yes No

Vibration Threshold Crossed

Yes No

Elevator 2 Status

Voltage(V): 12

Current(mA): 74

Human Presence

Yes No

Vibration Threshold Crossed

Yes No

Elevator 3 Status

Voltage(V): 12

Current(mA): 79

Human Presence

Yes No

Vibration Threshold Status

Yes No

Fig.8 LabVIEW output

VI. Conclusion

This design achieves the data transmission of the CAN Bus to the Ethernet, so by using this design each CAN nodes of the elevators can be connected to the Ethernet and hence successfully solving the connection of the internetworking of two heterogeneous network. The networking described in this paper is stable and reliable and is easy to implement. Also the GUI of the Elevator data is displayed in such a manner which is easy to interpret and understandable by the monitoring personal.

References

[1] Chen Zexi, Ye Xuan, Design of Interconnection Gateway in Elevator Remote Monitoring System, 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation.

[2] Bhuvaneswari.S, Sahaya Anselin Nisha.A, Implementation of Tcp/Ip on Embedded Webserver Using Raspberry Pi In Industrial Application, International Journal of Advanced Research in Computer and Communication Engineering.

[3] Ekiz H., Powner ET., Kutlu A. Performance Analysis of CAN in Bridged Systems, Proc. Of II Communication Conference,Manchester, June 1995.

[4] Adam Dunkels. The uIP1.0 Reference Manual. Swedish Institute of Computer Science, 2006: 34-37.

[5] NakulPadhye,Preet Jain Implementation of Arm Embedded Web Server for DAS using Raspberry Pi, Vol.3 No.4 April 2013.

[6] Jiang. J.N, Peng D.G, Zhang.H, (2008) Design and Realization of Embedded Web Server Based on ARM and Linux. Mechatronics, Vol.14 (10):37-40

[7] Dr. K.B. Khanchandani, International Journal of Engineering Science and Technology (IJEST), Embedded Web Server, SarikaChhatwani.

[8] SubhransuPadhee, Yaduvir Singh, “Data Logging and Supervisory Control of Process Using Lab VIEW”, Proceeding of the 2011 IEEE Students', Technology Symposium 14-16 January, 2011, lITKharagpur.

Mr. AjitPedha,received his Bachelor of Engineering in ECE from L.J Institute of Engineering and Technology, Gujarat, India. Currently pursuing M.Tech, in Embedded System Technology, SRM University Chennai, India. Area of interest: Embedded Systems and Electronics.

Mr.Nivash.Sreceived his Bachelor of Engineering in ECE from Priyadarshini College of Engineering and Technology, Anna University in2006. He received hisM.Tech degree in Embedded Systems from SRM University, Chennai in 2009. His research interests include Embedded Systems. Currently, he is an Assistant Professor (O.G) in the Department of Electronics and Communication Engineering at S.R.M. University, Kattankulathur, Chennai, India, working here since 2009.

.