

lab report

by James Hunt

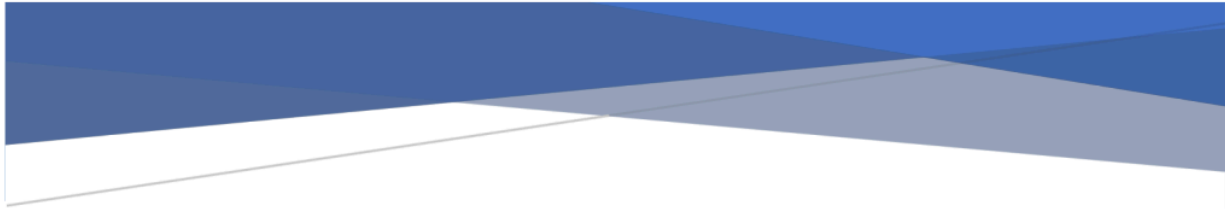
Submission date: 13-Nov-2024 09:20PM (UTC+0000)

Submission ID: 242585841

File name: 42360_James_Hunt_lab_report_523660_1101237506.docx (535.97K)

Word count: 2115

Character count: 12151



LAB REPORT

Distributed Intelligence Systems ACDA5018

Abstract

This report presents the design and implementation of a token-based distributed network aimed at controlling vehicle lights, focusing on the front left corner of the car. The system was developed using the Proteus Design Suite and uses RS232 for transmitting and receiving tokens. A single master node distributed signals to operate various light functions. The results demonstrate the effectiveness and potential scalability of the network for broader automotive applications.

James Hunt (2215832)
2215832@student.uwtsd.ac.uk

Contents

Contents	1
Table of Figures.....	1
Introduction	2
Background	2
Research.....	2
Quality Management Systems	2
Transmission Types	2
Point-to-Point Transmission	2
Point to Multipoint transmission	3
Multi-master transmission system	3
Objective	3
Method	3
Results.....	4
Discussion	5
Comparison to Traditional Systems	5
Potential Improvements	5
Other Transmission Standards	5
Code Improvements.....	6
Conclusion.....	6
References	7
Appendices.....	8
Flowchart	8
Code from Master Node	9
Code from Slave Node	11

Table of Figures

Figure 1 - Car Front Lights	3
Figure 2 - Token Table.....	3
Figure 3 - Screenshot of layout from Proteus.....	4

Introduction

As modern vehicles become more complex, traditional single signal-per-wire looms are no longer viable for controlling systems. This leads to using a distributed network to overcome some of these issues. This report looks at designing and implementing a simple token-based system within Proteus to control the various lights associated with the front left corner of a car. The network will demonstrate how a single master node can control multiple functions by using the RS232 protocol to transmit to a secondary node.

Background

These distributed networks are not a new concept in automobiles[1]. They have significant advantages over traditional wiring looms, such as improved reliability, ease of scalability, and signal noise tolerance.[1] A further advantage is the ability to remanufacture the parts to reduce both the environmental cost and the cost to the consumer. [2] A token-based network works by passing a token (a specific data packet) around all the nodes in the system. Each node is programmed to respond to only the token specific to its functions. This method ensures that all nodes have access to the network and only respond when required.

Research

Quality Management Systems

Quality Management Systems are used to provide clear guidelines and processes to be followed to ensure that regardless of industry the same quality can be expected. These standards can be either set internally or more commonly these are set by an external body. One prominent example of such an external body is the International Organisation for Standardisation (ISO). Importantly, many of the standards set are not just to maintain the quality of the end product, but also include references to regulations set to ensure the safety of both the environmental impact and health of individuals involved in the production process. [3]

Additionally, these standards can also form the basis for continuous improvement by identifying via audits, areas which are either performing above or below the standards guidelines. These audits can also create a data-driven review of either safety-critical systems or processes to discover any potential risks and hazards. This combined with a requirement in the quality management system to ensure continual training in best practices sets a reliable baseline and confirms an organisation's performance and reliability across both a local and potentially global setting.

Transmission Types

Transmission systems can generally be classified into the following categories.

Point-to-Point Transmission

This is a direct communication link between two distinct nodes. This type of network architecture ensures consistent bandwidth and reliable performance, commonly seen in applications like direct computer-to-printer connections of a USB cable or Bluetooth pairing like a set of headphones.

Point to Multipoint transmission

A single transmitter sends data to multiple receivers or nodes simultaneously, ideal for applications like wireless internet and traditional broadcast television. This system efficiently handles small data packets to multiple users or locations.

Multi-master transmission system

Allows multiple nodes to initiate communication and act as masters. In such systems, devices must cooperate to avoid conflicts. [4] This approach allows various network nodes to act as "masters" at different times, providing greater flexibility and redundancy. Multi-master systems are common in applications where multiple controllers or nodes need to collaborate or share information autonomously. Examples include industrial and automotive applications where multiple devices need to communicate and control various functions. However, this does increase the complexity as additional "handshakes" are required to ensure only one node is transmitting at a time.

Objective

The overall aim of this project is to show how a vehicle's front left lighting could be controlled by using the RS232 standard for the communication of tokens to control all the lights correctly. The system will control the left indicator, high/low beam functions and fog light.



Figure 1 - Car Front Lights

Method

This will be simulated within the Proteus design suite and various scenarios will be tested. These include activating the fog light only when the vehicle's lights are illuminated, a flashing indicator and toggling the main beam.

The code will be written in C for embedded systems using the CCS compiler to target PIC18F2525 microcontrollers.[5]

The process for controlling lights will be controlled via a token-based system, where a character will be used to control each of the light outputs.

Button Function	On	Off
Fog Light	F	f
Dipped lights	D	d
Left Indicator	L	l
Main Beam	M	m

Figure 2 - Token Table

These characters will be transmitted via RS232 the second node to then be decoded to activate the correct lamp.

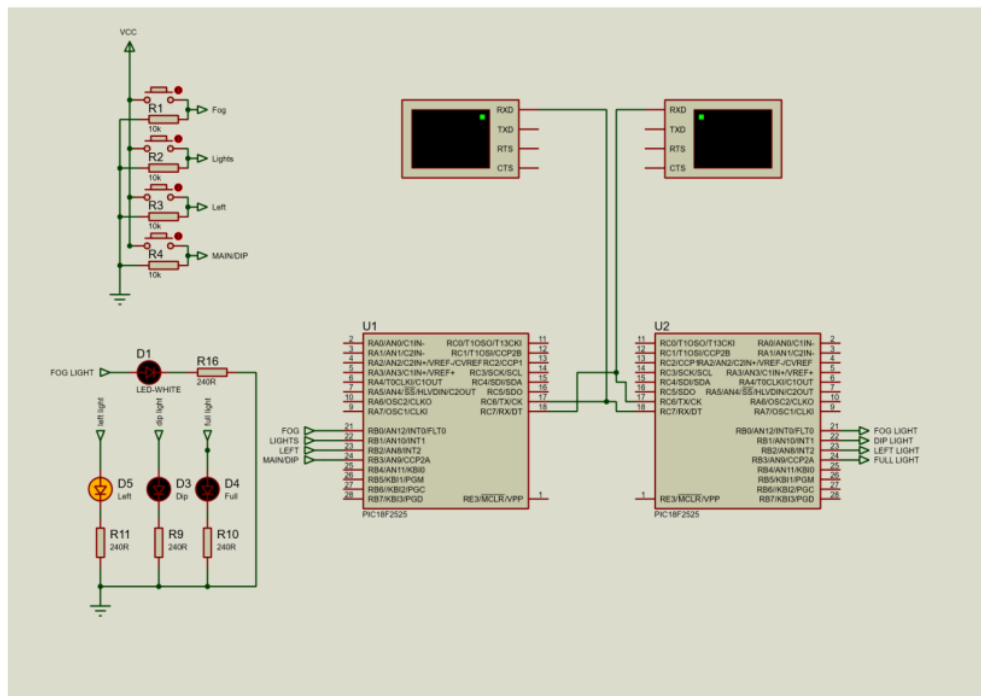


Figure 3 - Screenshot of layout from Proteus

Addition functionality of only allowing fog light and main beams to light when the dipped beam is 1st activated.

Results

The front left corner lighting control was successfully tested in Proteus. The results for each light function are summarised below.

Indicator Light Response: The indicator light accurately responded to the RS232 token and the flash control worked at the pre-programmed intervals. Only the indicator token activated this function, confirming the effectiveness of this token-based control.

Fog Light Response: The fog light toggled on and off upon receiving its designated RS232 token. Each toggle command was transmitted but as specified the fog light only illuminated when the car's dipped lights were illuminated. However, the transmitted token was both received and then remembered by the slave node to activate the fog light when the dipped lights were called.

Main Beam/Dipped Beam Response: Upon receiving the Main or Dipped beam tokens, the corresponding lights were activated and in accordance with the state of the Dipped beams, the Main beam could only be toggled when the dipped beam was illuminated.

Discussion

The project demonstrated the feasibility and effectiveness of a token-based distributed network for controlling vehicle lights. The use of a single master node and RS232 communication simplified the system architecture while ensuring reliable signal distribution. The token-based approach proved to be efficient and ensured synchronised operation of all light functions.

Comparison to Traditional Systems

Traditional automotive lighting systems often rely on point-to-point wiring, which can be complex and prone to failures. In contrast, the token-based distributed network, supported by RS232 communication, offers a more robust and scalable solution. The project's success suggests that similar approaches could be applied to other automotive systems, such as brake lights, interior lighting, and infotainment systems.

Potential Improvements

While the current implementation focused on the front left corner of the vehicle, future work could expand the network to cover all corners and integrate additional functionalities, such as adaptive lighting using sensors, diagnostic capabilities for bulb failures or looking at displaying all the lights via a matrix system. [6]

Other Transmission Standards

I²C (Inter-Integrated Circuit)

Replacing RS232 with the I²C (Inter-Integrated Circuit) protocol could offer improvements with multiple master capabilities allowing for more redundancy and therefore safety in the system. However, although this is a reasonably simple hardware setup, both data speeds and transmission distances are still limited.

CAN (Controller Area Network) Bus

Due to the limitations of I²C most modern vehicles use a CAN protocol. This was developed by Bosch and is focused on its ability to provide reliable communication and faster data rates over greater distances. This also can support a multi-master configuration but has the advantage of built-in error detection to allow the system to work reliably even in electronically noisy environments.

Message Queuing Telemetry Transport (MQTT)

This is designed as a lightweight publish-subscribe machine-to-machine communication protocol.[7] It's used widely by Internet of Things (IoT) devices, as its quality of service (QoS) levels make it capable of handling unreliable networks. MQTT architecture is based around a central "Broker" server that handles message distribution. Each of the nodes within the system and then either publish a message, subscribe to a topic to receive a message or a combination of both.

LoRaWAN /WiFi

All the previous protocols have been designed to work predominantly on a wired network however, various wireless solutions could also be a possibility. LoRaWAN (Long Range Wide Area Network) is a low-power wireless protocol designed to use a chirp spread spectrum. [8] This spread spectrum technique allows for a very robust transmission that resists interference from noise. However, this technique does have a limited data rate. Therefore, as power consumption is not a limiting factor in a vehicle a WiFi-based protocol would allow for a greater data rate and therefore the possibility to control more complex devices.

Code Improvements

At this point a detailed breakdown of the code might help to up this report's work count. However, as the code is heavily commented, and the author has other papers due in, only a few highlights will be explored.

The use of AND (&) logic-based operations whilst computationally efficient, does cause issues if care is not taken to ensure only the correct bits are controlled. The logical bitwise OR (|=) or XOR (^=) operator could be used to ensure only the required bit is switched and thus reduce the likelihood of errors due to incorrect bit addition.

Furthermore, defining the light bit control would improve the human readability within the main switch case-control. This would be achieved by either placing the definitions in the header file or at the top of the main c file.

Alongside the improvements to the code itself, additional functions could be added to more closely model typical vehicle functions. One example is the current code only allows the main beams to be illuminated if the dipped beams are first activated, this is not typically how these main beams work. Therefore, an adaptation could be made to allow the main beam to be "flashed" as long as the switch controlling them is depressed. A solution to this would be to transmit the "on" token when the button is first pressed. The "off" token would then be transmitted when the button is released. Managing the button pushes this way would avoid continually retransmitting the token and potentially "swamping" the bus line.

Conclusion

This project successfully implemented a token-based distributed network for automotive applications. The design and simulation of the systems were carried out using Proteus to design and simulate the system and RS232 for communication.

The primary function of this network was to efficiently control the light functions on the front left corner of the vehicle. The system successfully demonstrated the feasibility and potential advantages of this approach.

This report highlights the scalability, robustness, and potential advantages of token-based networks over traditional point-to-point wiring. Compared to conventional lighting control, this model offers a streamlined and fault-resistant setup that could be readily extended to additional corners of the vehicle.

Looking to the future, there is substantial potential for expanding this network beyond just lighting control. The design could be readily adapted for use in all the other corners of the vehicle, effectively managing more diverse functions like brake lights and reversing lights. Additionally, with the integration of a communication protocol that supports higher data rates, there is potential scope for combining this network with the vehicle's infotainment system, enhancing overall connectivity and functionality.

In summary, this project underscores the significant value of token-based distributed networks for controlling a wide array of systems. While this project's focus was on the automotive industry, the principles and advantages identified here show applications across various fields, paving the way for advancements in network design and implementation.

References

References

- [1] K. Pazul, "Controller Area Network (CAN) Basics," *Microchip AN713*, 1999. Available: <https://www.cl.cam.ac.uk/research/srg/han/Lambda/webdocs/an713.pdf>.
- [2] E. Sundin and O. Dunbäck, "Reverse logistics challenges in remanufacturing of automotive mechatronic devices," *Jnl Remanufactur*, vol. 3, (1), pp. 1–8, 2013. . DOI: 10.1186/2210-4690-3-2.
- [3] Anonymous (-11-11). *ISO - International Organization for Standardization*. Available: <https://www.iso.org/home.html>.
- [4] M. Escalante and N. Dimopoulos, "Assessing the Feasibility of Hardware Interface Designs in Microprocessor-based Systems," 1998. Available: https://www.researchgate.net/publication/2313393_Assessing_the_Feasibility_of_Hardware_Interface_Designs_in_Microprocessor-based_Systems.
- [5] M. Bates, *Programming 8-Bit PIC Microcontrollers in C : With Interactive Hardware Simulation*. (1st ed.) 2008.
- [6] G. Elger *et al*, "LED matrix light source for adaptive driving beam applications," in *2013 IEEE 63rd Electronic Components and Technology Conference*, 2013, . DOI: 10.1109/ECTC.2013.6575624.
- [7] Anonymous (). *MQTT - The Standard for IoT Messaging*. Available: <https://mqtt.org/>.
- [8] Anonymous "What-is-lorawan," LoRa Alliance, -11. 2015.

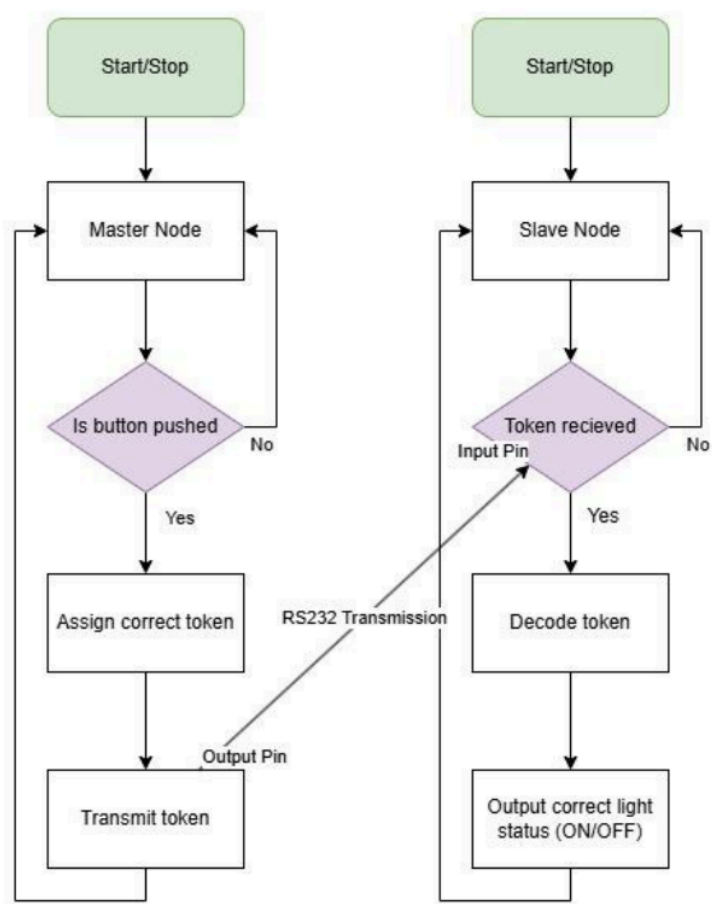
Appendices

Flowchart

Flow chart demoing a basic breakdown of data flow

Simple Flowchart

This briefly outlines the process between each of the microcontrollers.



Code from Master Node

```
#include <main.h>
// U1 Inputs (Master)

char read_buttons(void); // setup function for reading button pushes
char local_output;       // setup for a variable to remember light status
char txbyte;             // setup for transmitting tokens

void main()
{
    char buts = 0; // setup variable for reading buttons

    while (TRUE)
    {
        txbyte = 0; // setup and clear variable for transmitting tokens each loop

        buts = read_buttons(); // assigns variable buts to result of function
        switch (buts) // switch case to determine what to do following button press
        {
            case 1:
                if (local_output & 0x01) // set the fog lights on
                {
                    local_output -= 1; // turn bit off
                    txbyte = 'f';      // set token character
                }
                else
                {
                    local_output += 1; // turn bit on
                    txbyte = 'F';      // set token character
                }

                break;
            case 2:
                if (local_output & 0x02) // toggle the Dipped light on and off
                {
                    local_output -= 2;
                    txbyte = 'd';
                }
                else
                {
                    local_output += 2;
                    txbyte = 'D';
                }
                break;
            case 4:
```

```

        if (local_output & 0x04) // toggle the left indicator on and off
        {
            local_output -= 4;
            txbyte = 'l';
        }
        else
        {
            local_output += 4;
            txbyte = 'L';
        }
        break;
    case 8:
        if (local_output & 0x08) // toggle the Main beam on and off
        {
            local_output -= 8;
            txbyte = 'm';
        }
        else
        {
            local_output += 8;
            txbyte = 'M';
        }
        break;

    default:
        break; // break out of switch case
    }

    printf("%c", txbyte); // print to rs232
}
}

// functions
char read_buttons(void)
{
    char value = 0; // used to store button inputs
    while (!input_b())
        ; // waits for button press
    value = input_b(); // reads port b
    delay_ms(500); // timer for button release
    return (value); // returns value to main loop
}

```

Code from Slave Node

```
#include <main.h>
// U2 Outputs (slave)

char control_byte = 0;
char slave_output = 0;
unsigned int Lights = 0;
unsigned int fog = 0;
unsigned int fullbeam = 0;
unsigned int counter = 0;
unsigned int flash = 0;
unsigned int left = 0;

void flash_control(void);

#INT_TIMER2

void TIMER2_isr(void) // global timer used for indicator flash this is always running in
the background
{
    counter++; // add one to current count
    if (counter == 40) // if counter is equal to 40
    {
        counter = 0; //reset counter
        if (flash)
            flash = 0; // reset flash variable
        else
            flash = 1;
        flash_control(); // calls flash control routine
    }
}

void main()
{
    setup_timer_2(T2_DIV_BY_16, 249, 15); // 1.0 ms overflow, 15.0 ms interrupt
    enable_interrupts(INT_TIMER2); // allows the timer to interrupt the main loop - used
to control flashing indicators
    enable_interrupts(GLOBAL);

    char input = 0;

    while (TRUE)
    {
        if (kbhit()) // checks if keyboard has been hit(token received)
        {
            input = getc(); // wait for a token to be received
            printf("%c", input); // prints the value of the token used to visualise the
process
        }
    }
}
```

```

switch (input) // main switch case statement to control the car lights
{
case 'D': // switch on dipped lights
    Lights = 1; // used to remember light status
    if ((fullbeam) & (fog)) // checks to see status of full beam and fogs
lights when lights switched off
    {
        control_byte = control_byte += 0xB; // if both on set correct pins high
via bit control
    }

    else if (fog)
    {
        control_byte = control_byte += 0x3; // if fog set fog and dipped on
    }
    else if (fullbeam)
    {
        control_byte = control_byte += 0xA; // if full beam set full and
dipped on
    }
    else
    {
        slave_output += 0x2; // if no other lights status just switch on
dipped beam
        control_byte = control_byte += 0x2;
    }
    break;
case 'd':
    slave_output = slave_output & 0xf4; // lights off
    control_byte = control_byte & 0xf4;
    Lights = 0;
    break;
case 'L':
    slave_output = slave_output += 0x4; // left indicator
    left = 1; // set variable for indicator routine
    break;
case 'l':
    slave_output = slave_output & 0xfb;
    left = 0; //clear variable for indicator routine
    break;

```

```

        case 'F':
            fog = 1;    // remembers if fog light was on if main lights switched off
            if (lights) // fog light only when lights on
            {
                slave_output += 0x01;
                control_byte += 0x01;
            }
            break;
        case 'f':
            slave_output = slave_output & 0xfe; // bit control but ignores the status
of dipped lights
            control_byte = control_byte & 0xfe;
            fog = 0; // clears fog light variable
            break;
        case 'M':
            fullbeam = 1; // remembers if fog light was on if main lights switched
off
            if (lights) // checks to see if dipped beam on
            {
                slave_output += 0x08; // turn main beam on if lights on
                control_byte = control_byte += 0x08;
                fullbeam = 1; // remembers if fog light was on if main lights
switched off
            }
            break;
        case 'm':
            slave_output = slave_output & 0xf7; // switches off
            control_byte = control_byte & 0xf7;
            fullbeam = 0; // clears fullbeam variable
            break;

        default:
            break;
    }
    output_b(control_byte); // send data to port b
}
}

void flash_control(void) // function for controlling flashing of indicator
{
    if (left & flash) // checks to see if the two variables are 1
        output_high(pin_b2); // turn on indicator lamp
    else
        output_low(pin_b2); // turn off indicator lamp
}

```

lab report

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

70/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

GRADING FORM: SMU GENERIC_GJ

JAMES HUNT

69.80

RESEARCH-INFORMED LITERATURE



You have explained what point to point and point to multipoint and multimaster are correctly. 65%

13

KNOWLEDGE AND UNDERSTANDING



The design of your circuit is suitable for the task 75%

18.75

ANALYSIS



You have completed the programs for both devices and the circuit works well. 70%

17.5

PRACTICAL APPLICATION AND DEP.



Your explanation is good but does not acknowledge that both pics speak to memory IC in multimaster mode and not directly to each other 62%

9.3

SKILLS FOR PROFESSIONAL PRAC.



The report is well presented with sections to explain the code and include a reasonable number of references. 75%

11.25