

locktronics

Simplifying Electricity

CAN bus systems and operation



LK9893

MATRIX

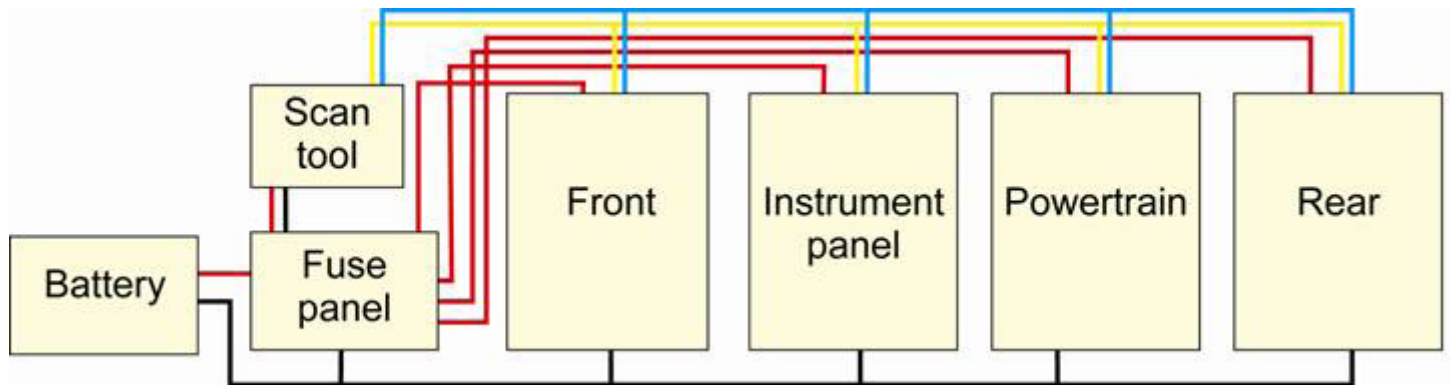
www.matrixmultimedia.com

Copyright © 2009 Matrix Multimedia Limited

Contents

Introduction	3
Overview	4
Building Node A	5
Building Node B	6
Building Node C	7
Building Node D	8
Building the Scan/Fuse Node	9
Wiring It All Up	10
Worksheet 1 - The Startup Routine	11
Worksheet 2 - Seeing the Messages	12
Worksheet 3 - Receiving Messages	14
Worksheet 4 - System Monitoring and 'Pinging'	15
Worksheet 5 - System Faults	16
Worksheet 6 - Circuit Faults	18
Worksheet 7 - Fault Diagnosis Using Charts	19
Worksheet 8 - Build Your Own Car!	20
Instructor Guide	21
Scheme of Work	25
Message Code Summary	31
Kvaser Analyser Set Up	32
System Graphic	33

Introduction



What is this all about?

The 'CAN' in CAN bus stands for 'Controller Area Network' and 'bus' a bundle of wires. CAN bus is a fast computer data bus similar to the network used for linking computers. Its use in cars greatly reduces the number and weight of wires used, simplifies construction, reduces cost and allows the use of advanced features, in troubleshooting and monitoring the automotive system.

The flexibility and power of the CAN standard brings many advantages to modern automotive systems. However it can seem like a 'black art'. This course is designed to introduce you to the technology involved in CAN bus systems, and reveal some of its magic!

The hardware:

The course looks at the role of Electronic Control Units (ECUs), linked by a CAN bus, in controlling automotive systems. Modern motor vehicles can contain over a hundred ECUs, controlling everything from the Anti-lock Braking System (ABS) to the position of the driver's seat.

An ECU contains a microprocessor, running software to control the devices attached to it, and to communicate, using the CAN protocol, with other ECUs.

In this course, MIACs (Matrix Industrial Automotive Controllers) are used as ECUs. Each MIAC is running a program that allows you to make it act as one of five types of node. These types are described in the following sheets.

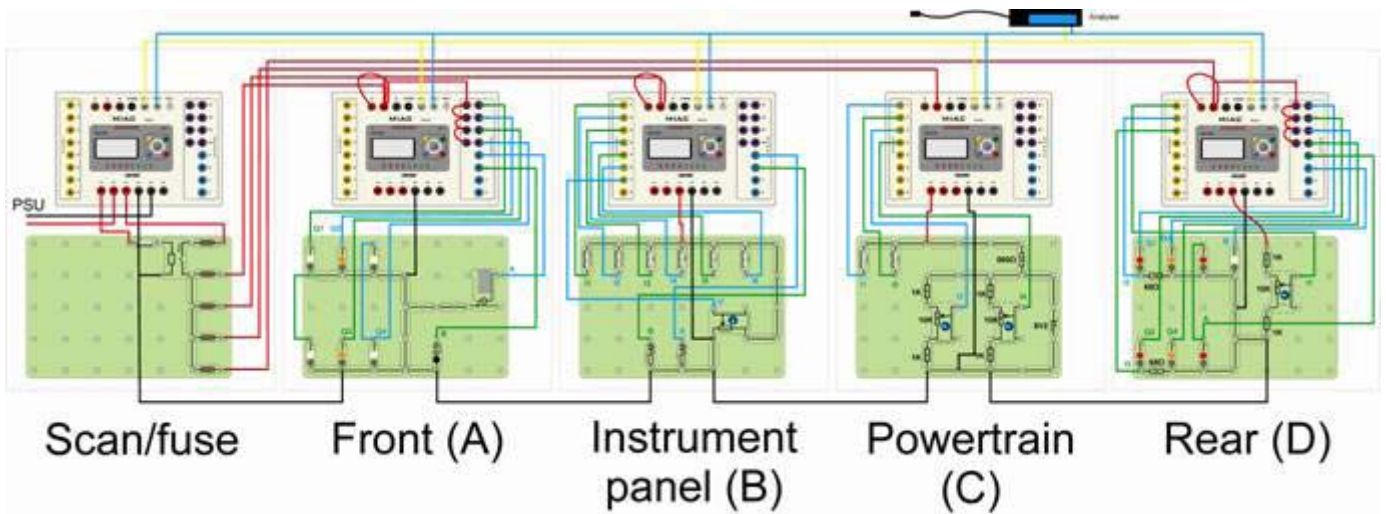
Using these, you will:

- construct and test a number of ECU nodes, using Locktronics equipment,
- wire these up into a single CAN bus system,
- carry out a range of tests on the whole system.

In doing so, you will learn about:

- the role of the ECU;
- the structure of CAN bus systems;
- the CAN bus message format
- the use of scan tools in fault diagnosis.

Overview



Node by node

Your group is going to build five nodes. These are:

Node A - the front ECU and circuit, (side lights, headlights, indicators, wipers and horn,)

Node B - the instrument panel ECU and circuit, (lighting switches, horn switch, wiper speed controller, indicator switches, and warning LEDs,)

Node C - the power train ECU and circuit, (reversing switch, brake light switch, engine temperature sensor and oil pressure sensor,)

Node D - the rear ECU and circuit, (side lights, indicators, brake lights, reversing light, fuel level sensor,)

Scan/fuse - the system controller and scan tool, fuse box and power relay

The number of nodes, and how they are configured, varies greatly from one car to another. Different manufacturers use different software, and different messages within their CAN bus systems. The system used in this course is broadly representa-

tive of systems found in cars.

Each node consists of an ECU and the associated circuitry required to control the devices attached to that node. Each node is supplied with power, ground and the two wires, called CAN High (CANH) and CAN Low (CANL). In motor vehicles, these CAN wires are twisted together.

Note - The Scan/fuse node is not present as such in a car. It is included to allow us to control the overall system and to provide tools to help you understand what is happening.

The whole system can be seen in the diagram above. Do not worry if this looks very complicated - all the wiring will be explained as the course progresses.

Your instructor will provide your group with a larger version of this picture.

Building Node A

The following pages describe how to build and test each of the five nodes, consisting of an ECU and the associated circuit to control the components attached to the node. All MIAC units are programmed with the same Flowcode program, called LK8902. Users will choose the function of each ECU unit from a menu offered on power up.

Over to you:

1. Wire up a MIAC unit and Locktronics equipment, as shown in the diagram on the right. This mimics the circuit you might find on the front of a car.
2. Set the voltage on the power supply to 13.5V.
3. Switch on. You will see a message:

```
Locktronics
LK8902
CAN bus systems
and operation
```

followed by:

```
Select a node
A
```

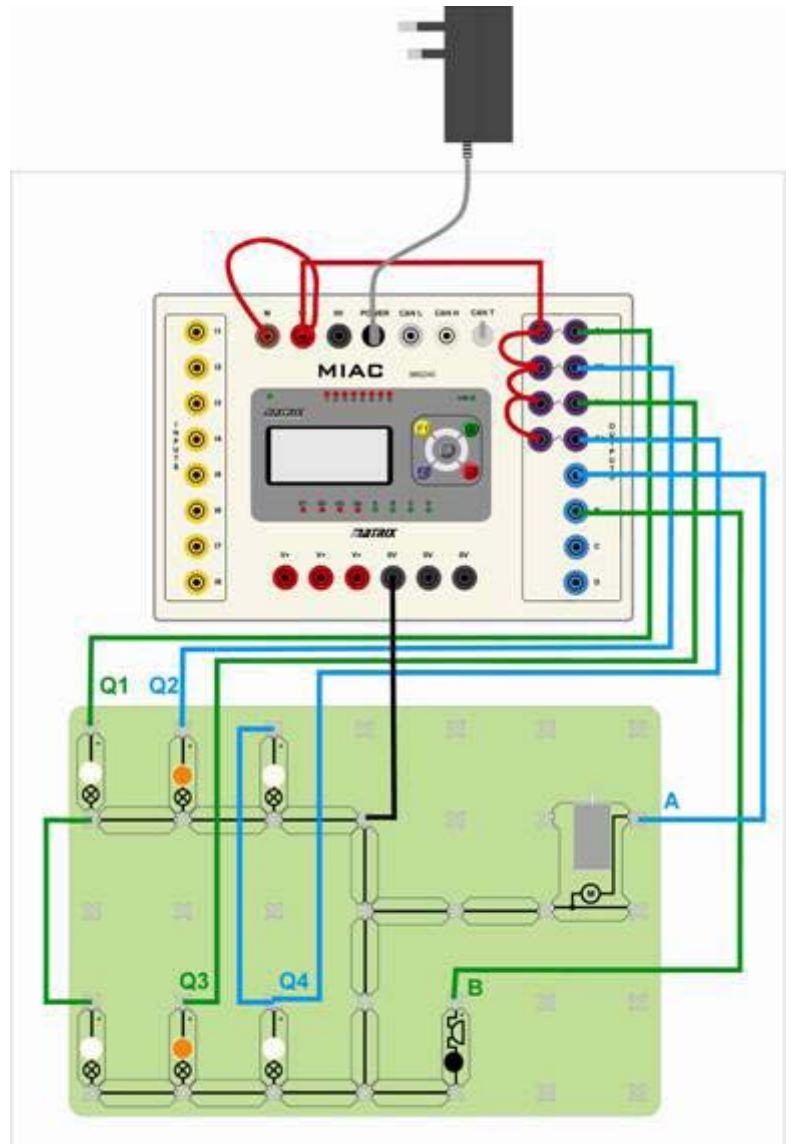
4. Press the OK button to select node A.
5. You should see a message:

```
Node A
F1: Node test
F2: CAN system
OK: Change node
```

Press the F1 button on the MIAC

6. The motor, each indicator and each pair of headlamps should come on in turn with the appropriate message on the MIAC unit. If the motor does not turn, or the lamps fail to light then check your wiring.
7. Press the reset button (top right on the MIAC) to stop the test.

Node A represents the front ECU. The lamps on this node mimic the side lights, headlights and indicators. The motor represents the windscreen wiper motor. The buzzer represents the horn.



Building Node B

This node represents the ECU that controls the instrument panel.

To help you keep track of the functions of the various switches you may wish to label each Locktronics switch carrier with its function.

The switches on this node mimic (left to right) side lights, headlights, left indicator, right indicator, horn and wipers.

The potentiometer controls wiper speed.

The left hand LED is the fault light which illuminates whenever a fault is detected. It will not go off until the fault is reset on the Scan/fuse node.

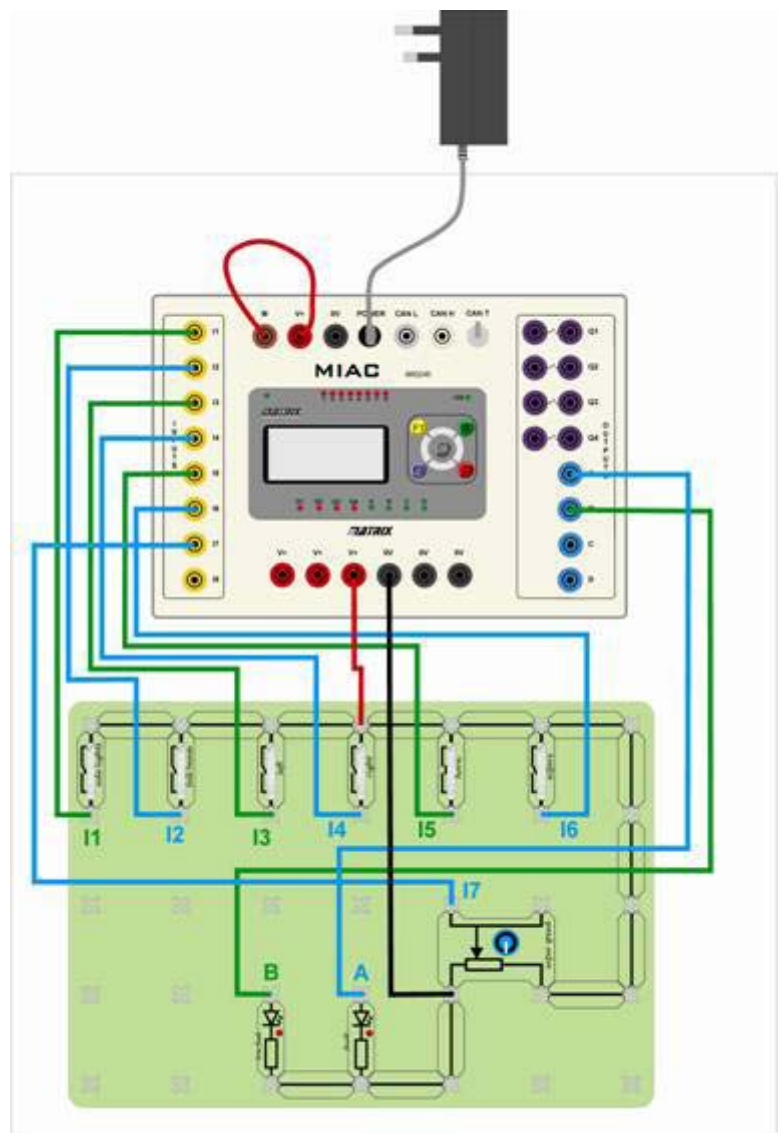
The LED on the right represents the low fuel warning light.

Over to you:

1. Wire up a MIAC and the circuit shown opposite for Node B. This circuit mimics the switches and controls on the instrument panel.
2. Set the voltage on the power supply to 13.5V.
3. Switch on.
4. Select 'Node B' from the menu system using the up / down keys on the MIAC.
5. You will see the message:

```
Node B
F1: Node test
F2: CAN system
OK: Change node
```

6. Press the F1 button on the MIAC and look at the MIAC panel and check that each switch and the potentiometer are performing their appropriate functions. The potentiometer should give a readout of 0 (minimum) to 255 (maximum). If the circuit does not function as expected then check your wiring.
7. Press the reset button (top right on the MIAC) to stop the test.



Building Node C

This node represents the ECU that controls the power train. In motor vehicles, this ECU controls functions such as fuel injection and ignition timing.

The switches on this node (left to right) mimic the reverse switch in the gearbox, and the brake switch on the brake pedal.

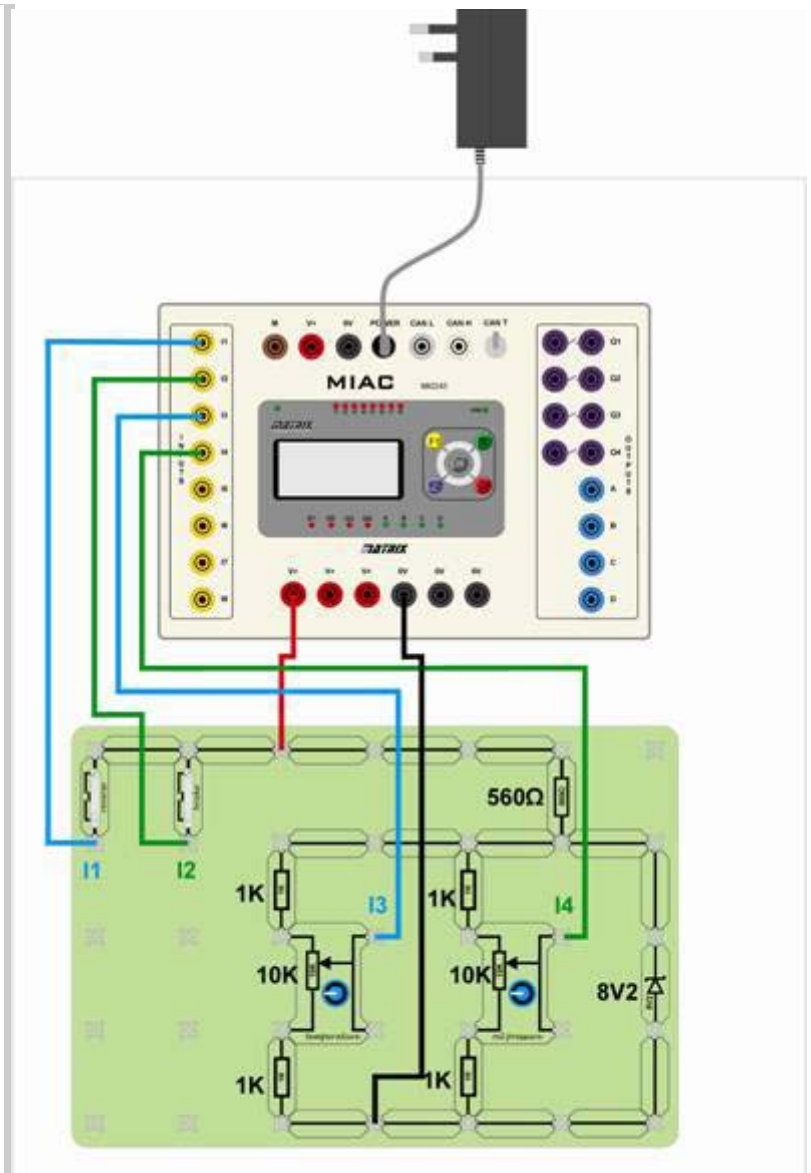
The two potentiometers represent the engine temperature sensor and oil pressure sensor.

Over to you:

1. Wire up a MIAC and the circuit shown opposite for Node C. This circuit mimics some of the switches and sensors that are connected to the power train ECU.
2. Set the voltage on the power supply to 13.5V.
3. Switch on.
4. Select 'Node C' from the menu system using the up / down keys on the MIAC.
5. You will see the message:

```
Node C
F1: Node test
F2: CAN system
OK: Change node
```

- Press the F1 button on the MIAC
6. Look at the MIAC panel and check that each switch and the potentiometer are mimicking their appropriate functions. The potentiometer should give a readout of around 21 (minimum) to 234 (maximum).
 7. Press the reset button (top right on the MIAC) to stop the test.



Building Node D

This node represents the ECU that controls the functions at the back of the motor vehicle. These are usually mainly lighting components.

The lamps on this node mimic the various lamps controlled by the rear ECU: side lights, indicators, brake lights and a reversing light.

The potentiometer represents a fuel sensor.

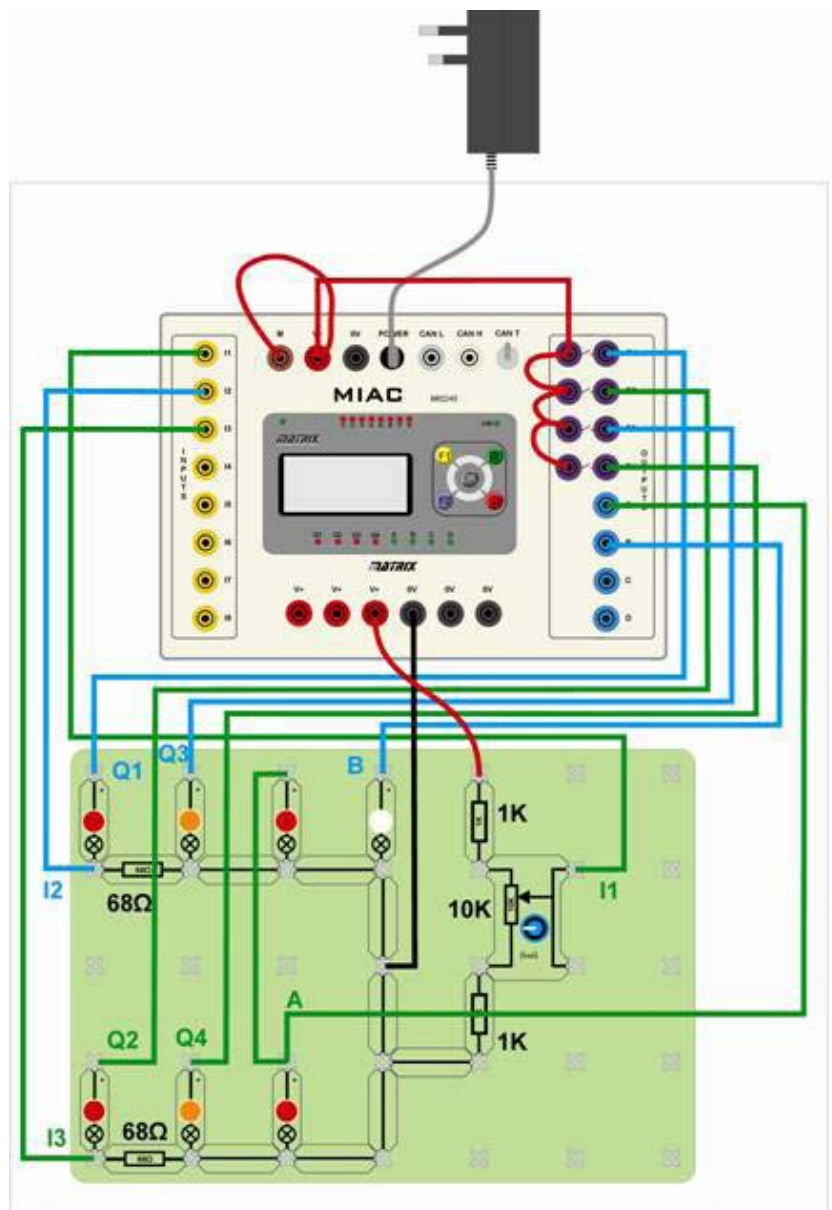
Over to you:

1. Wire up a MIAC and the circuit shown opposite for Node D, which mimics the indicators and sensors connected to the Rear ECU.
2. Set the voltage on the power supply to 13.5V.
3. Switch on.
4. Select 'Node D' from the menu system using the up / down keys on the MIAC.
5. You should see a message:

```
Node D
F1: Node test
F2: CAN system
OK: Change node
```

Press the F1 button on the MIAC

6. Look at the MIAC panel and check that the potentiometer is mimicking the fuel sensor. Check that all the lamps are working - if not then check your wiring.
7. Press the reset button (top right on the MIAC) to stop the test.



Building the Scan/fuse Node

This node is not found in the motor vehicle itself, but may be part of the troubleshooting equipment used by mechanics to locate faults.

The equipment attached to this ECU represents a basic fuse box.

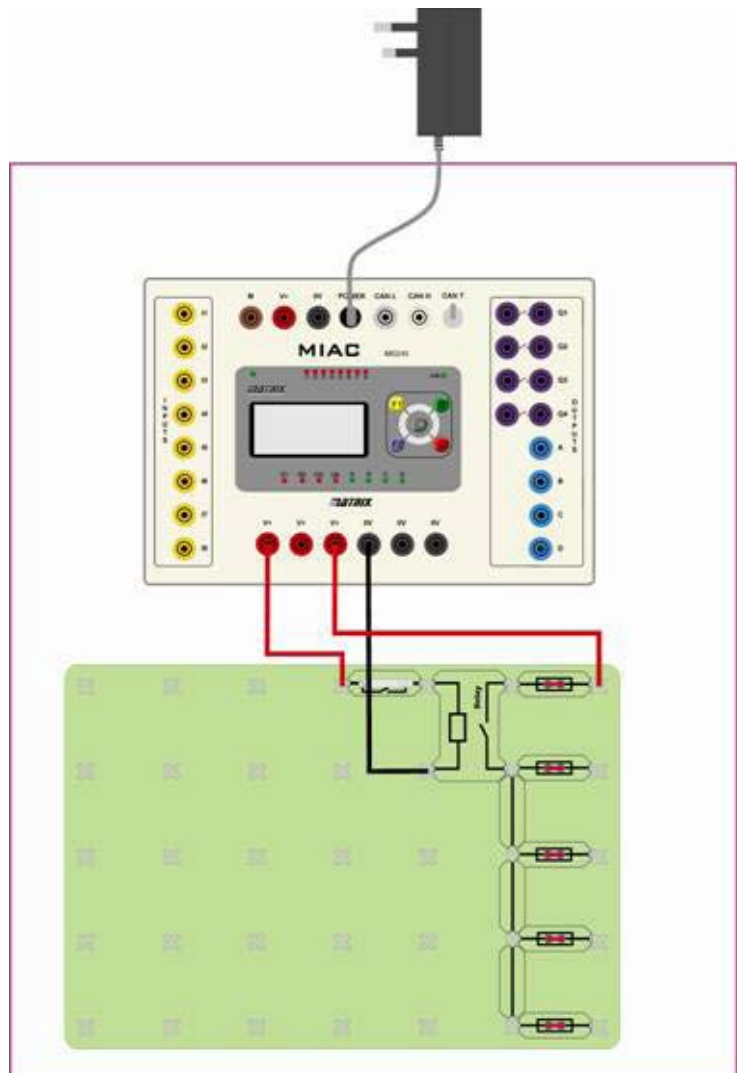
You would not find a single power relay like this in a car - but it helps to control this system. The first fuse limits the current into the overall system, and each additional fuse limits the current to a particular node.

Over to you:

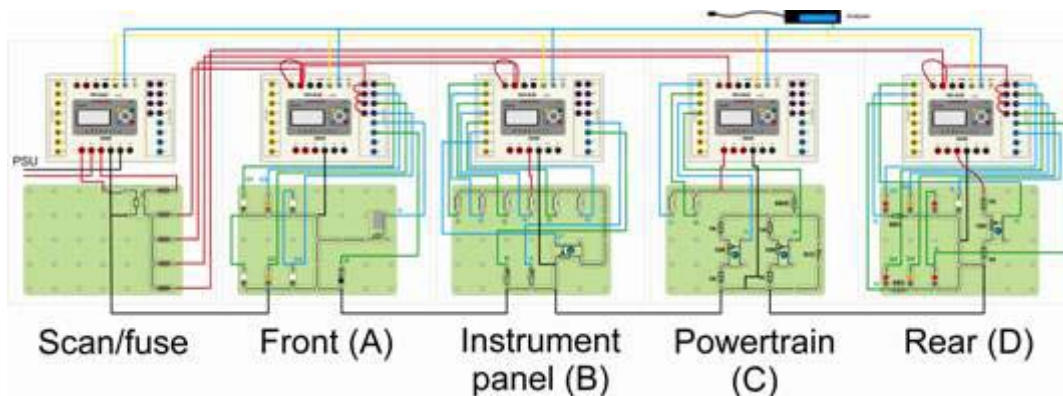
1. Wire up a MIAC and the circuit shown opposite for the Scan/fuse node. The circuit connected to the MIAC mimics the fuse box. The switch represents the ignition key which controls the power supplied, via the relay and fuses, to each of the ECUs in the system. This MIAC monitors the other nodes. It does not play a part in controlling any functions in the automotive system.

It will be used later on in the course to insert faults and to monitor CAN signals.

2. Set the voltage on the power supply to 13.5V.
3. Switch on.
4. Select 'Admin' from the menu system using the MIAC's up / down keys.
5. You will see the message:
`Admin node`
6. Check that you can scroll through the six options on the menu using the up / down keys.
7. Check that your circuit is working by checking the output voltages on the



Wiring It All Up



Your group now has wired up and tested the individual nodes of the CAN bus system. Next, you need to connect them all together.

Over to you:

The power supply:

All nodes will receive power from the 'fuse box' attached to the Scan/fuse node.

1. With the power supply turned off, connect a 13.5V power supply to the MIAC and relay on the Scan/fuse node.
2. Connect wires from each fuse to power one of the nodes A to D. In this way, each ECU has its own fuse.
3. Connect the 0V terminals on all nodes together.
4. Switch on the power supply. If you have wired everything up correctly the power LED on each MIAC will light up. If not, then check your wiring.
5. Switch off.

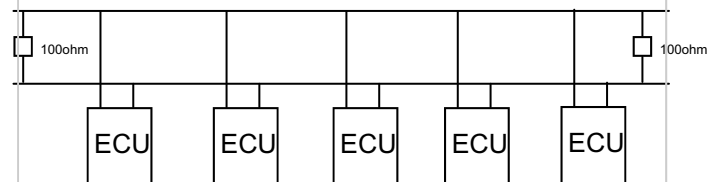
Connecting the CAN bus

The CAN bus consists of just two wires. CAN H (for CAN High) and CAN L (for CAN Low). In a motor vehicle, these are normally very fine wires that are twisted together - known as a 'twisted pair'. The nodes are wired in parallel, i.e. CAN H of node D is connected to CAN H of nodes A, B and C. CAN L is wired up in the same way.

Notice that there is no physical hierarchy in the CAN bus system - no central server. All nodes have equal status.

For clarity, use blue wires for CANH and yellow wires for CANL connections.

The CAN bus requires a 100Ω resistor, connected between CAN H and CAN L, at each end of the CAN bus, as shown in the diagram:

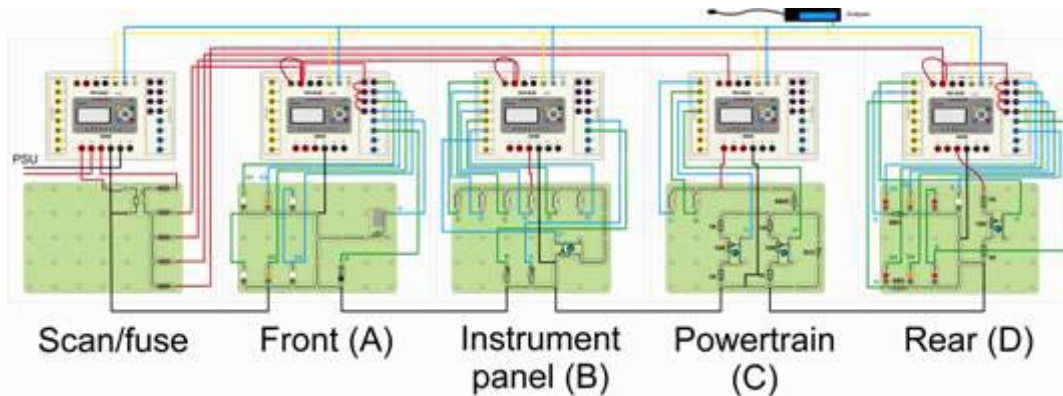


These resistors absorb the energy of the data signals, and prevent reflections back down the bus. The MIACs contain suitable resistors, connected internally between the CAN H and CAN L wires.

6. On the Scan/fuse node and on node D, turn the 'CAN termination' switch to the 'ON' position.
7. Check that the 'CAN termination' on the other MIACs is switched off.
8. The system is now ready for testing.

Worksheet 1

The Startup Routine



All MIAC units are running the same program. The hardware inside each MIAC is the same. They carry out different functions, as separate nodes, because of choices you made when the nodes were set up. These choices are stored in memory, so that the MIAC remembers what functions to carry out when power is restored.

Over to you:

1. Set the voltage on the power supply to 13.5V.
2. Switch on and check that the power LED on each MIAC is lit.
3. Each MIAC ECU should display:

```
Node X
F1: Node test
F2: CAN system
OK: Change node
```

Select F2 to make the MIAC mimic the preset CAN bus function.

Note: Carry out this step last on Node C, as node C runs the Startup routine.

In the Startup routine one ECU in the system is programmed to check that all of the other ECUs are present.

Node C - the power train ECU - is often given this task. It sends a message on the CAN bus to each node in turn and waits for a reply. It will display 'Checking Node X' and then verify the presence of each node in turn.

4. Check that each node is registered.
5. Switch off.

6. Remove the fuse carrier for Node D
7. Switch on and select F2 to initialise each node, but do Node C last.
8. Look at the messages displayed on Node C.
9. Switch off, and replace the fuse carrier.
10. Repeat step 7.
11. What is the status of the fault light on Node B?
12. Discuss what should happen when a fault like this is detected.
Should the car start?
Should it be disabled?
What level of fault should be tolerated?
What faults are critical?
13. On the Scan/fuse Node menu, use the up/down keys to select 'Release flts'.
14. You will then see a sub menu:
Get fault info
Clear all faults
15. Select 'Clear all faults' and then press F2. The fault light on Node B will go off.

Worksheet 2

Seeing the Messages



The Scan/fuse Node takes no part in controlling components in our CAN bus system. It is just another computer connected to the CAN bus, which helps us to see what is going on in the system.

It has some features that are found in the scan tools used in automotive diagnostics. These vary greatly in their features and complexity. Some just report faults. Others both report faults and allow you to reset them. Some connect to a PC, while others have a hand-held display.

Over to you:

1. Set the voltage on the power supply to 13.5V.
2. Switch on and check that the power LED on each MIAC is lit.
3. On the Scan/fuse Node, use the up / down keys to select the `Msg monitor` option.
The Scan/fuse MIAC unit will now display messages sent on the CAN bus.
4. On Nodes B and C, turn off all switches.
5. Next, press each switch in the system and hold it down.
6. As you do so, make a note of the **first** message - there will be more than one - on the Scan/fuse Node.
7. Complete the table shown opposite.

Sensor	ID	D0
Side lights (A1) switch on	8	1
Full beam (A2) on		
Left (A3) on		
Right (A4) on		
Horn (A5) on		
Wiper (A6) on		
Reverse (B1) on		

Worksheet 2

Seeing the Messages... continued...

Messages are at the heart of the CAN system. Without them, it's just a load of wires and circuit boards.

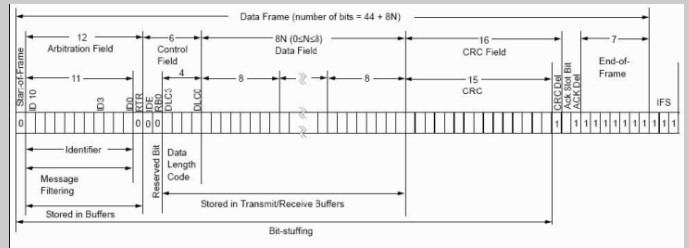
Each message contains an identifier (ID) and data. Actually, there's more – acknowledgment bits, checksums, transmission details etc. which need not worry us now.

A node will accept or reject a message, depending on its ID. Messages can contain data or be completely empty. Simple signals, such as 'turn on the horn' need little data. For others, e.g. engine temperature readings, more data is transmitted. Each message frame can contain up to eight bytes of data, (D0 to D7). (A byte is an eight bit binary number. It can identify up to 256 different conditions.)

Unfortunately, different manufacturers use different codes within this structure. Our CAN bus system is relatively simple and designed to illustrate principles. It uses only the **ID** and **D0** bytes, and does not populate the D1 to D7 bytes.

Each ECU puts a message onto the CAN bus when an event occurs.

In our system, when the horn switch is pressed, for example, the ECU on Node B transmits the message: **ID 5 D0 1** and, when released, sends the message :**ID 5 D0 0**



Sensor	ID	D0
Side lights (A1) switch on	8	1
Side lights (A1) switch off	8	0
Full beam (A2) on		
Full beam (A2) off		
Left (A3) on		
Left (A3) off		
Right (A4)		
Right (A4) on		
Horn (A5) on		
Horn (A5) off		
Wiper (A6) on		
Wiper (A6) off		
Reverse (B1) on		
Reverse (B1) off		

Sensor	ID	D0
Fuel level empty		
Fuel half full		
Full fuel tank		
Wiper half speed		

Over to you:

1. You are now going to investigate the message structure. Complete the table opposite for all switches and sensors, writing down the ID and D0 value for each of the 'on' (switch pressed) and 'off' (switch released) messages.
2. Discuss how D0 is used to indicate switch status.
3. Carry out a similar exercise for the fuel and other sensors in the system. How is D0 used to give an indication of the quantity being measured?

Worksheet 3

Receiving Messages



iStock_000003887872Small.jpg

You have seen that when a switch is pressed or released, or when a sensor value changes, the corresponding ECU sends an appropriate message to other parts of the system so that they can react.

In this assignment, we test an ECU's ability to receive messages. In our simplified system this is easy because we can see a light flashing, the horn comes on etc. However, we can also use the Scan/fuse Node to allow us to send messages to the

Over to you:

1. Set the voltage on the power supply to 13.5V.
2. Switch on and check that the power LED on each MIAC is lit.
3. On the Scan/fuse Node, use the up / down keys to select the `Send message` option.
4. Construct a 'switch horn on' message by setting `ID=5` and `D0=1`.
5. Press the 'OK' button to send this message.
6. What happens?
7. Press and release the horn switch on node B. Notice what happens.
8. Look up the codes to turn the brake lights on and off. Use the Scan/fuse node 'Send message' feature, to make the brake lights go on and off.
9. Turn the fuel sensor potentiometer to the 'full' position.
10. Use the Scan/fuse node 'Send message' feature to send a message

In this exercise you fooled Node A into believing that the horn switch on Node B was pressed. In doing so, you showed that Node A is receiving messages correctly.

We all know that computers are stupid. They do only what they are told. This also applies to ECUs. If they get the wrong message, they carry out the wrong action.

Good system design can - to some extent - prevent this.