

MATRIX

Electronic Flight Information Systems



MATRIX

CP5715

www.matrixtsl.com

Copyright © 2021 Matrix Technology Solutions Limited

Introduction	
Worksheet 1 - Building the system	5
Worksheet 2 - Adding PC-based diagnostics	7
Worksheet 3 - Understanding the Avionics Analyser	9
Worksheet 4 - CAN bus messages	10
Worksheet 5 - Sending CAN bus messages	12
Worksheet 6 - Embedded fault detection	14
Worksheet 7 - Node detection	15
Worksheet 8 - Diagnosing CAN faults with a multimeter	16
Worksheet 9 - Diagnosing faults with an oscilloscope	19
Worksheet 10 - Using E-blocks as a compass sensor	23
Reference	25
Notes for the instructor	38

As aircraft carry more and more passengers and goods at higher and higher speeds, accurate and timely information about the flight and factors relating to it becomes an ever more vital part of aeronautical engineering.

In an expanding field - jets, helicopters, military aircraft, satellites, space shuttles ..., involving an expanding range of techniques and technologies, aeronautical engineers research and develop new approaches to make flight and flying vehicles safer and more efficient.



© CanStockPhoto.com - csp37965146

Overview

This module examines the way in which data is transmitted from sensor to display in aeronautic systems using a controller area network (CAN).

CAN was developed as a means of reliable and flexible data transmission in electrically harsh environments, such as in automobiles and in power stations.

It allows the use of onboard diagnostics (OBD) to analyse, store and report problems that arise.

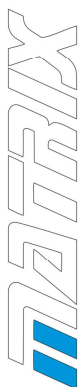
The diagram on the next page shows the system that you will build. Initially

It includes several real sensors, such as compass and attitude sensors, and several virtual sensors, in particular five potentiometers that generate voltage signals to simulate the output of real sensors, such as engine r.p.m. and fuel level sensors.

The system uses two kinds of display to present the output of these sensors. One is a software application called 'Avionics Analyser', used to monitor the performance of the system and to identify and log faults that occur. The other is a multifunction electronic flight information display, designed for use within the aircraft.

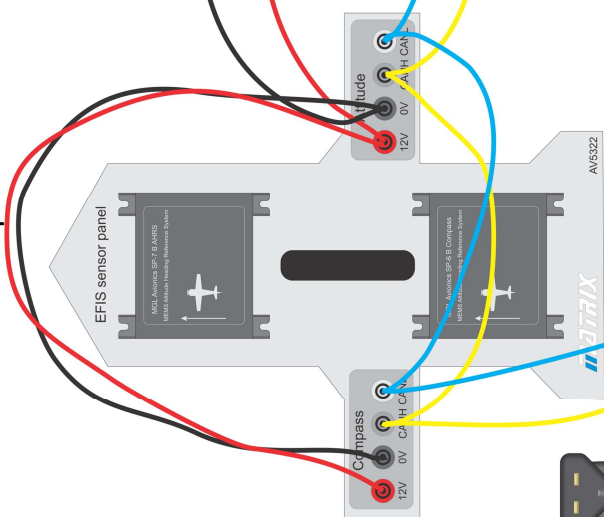
The sensors and displays are connected by a controller area network, consisting of two wires, called 'CAN High' and 'CAN Low'. You will be shown how to examine, decode and insert CAN signals on this network.

An E-blocks II development system will be used to create simulated data to illustrate aspects of the system's performance.

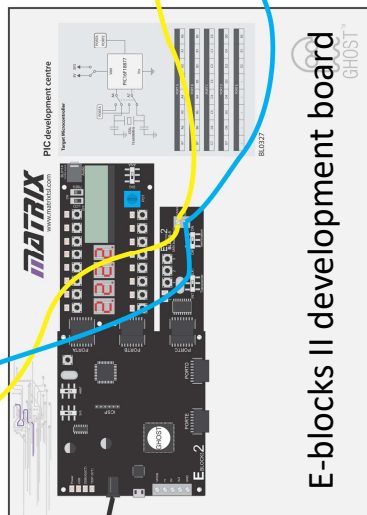
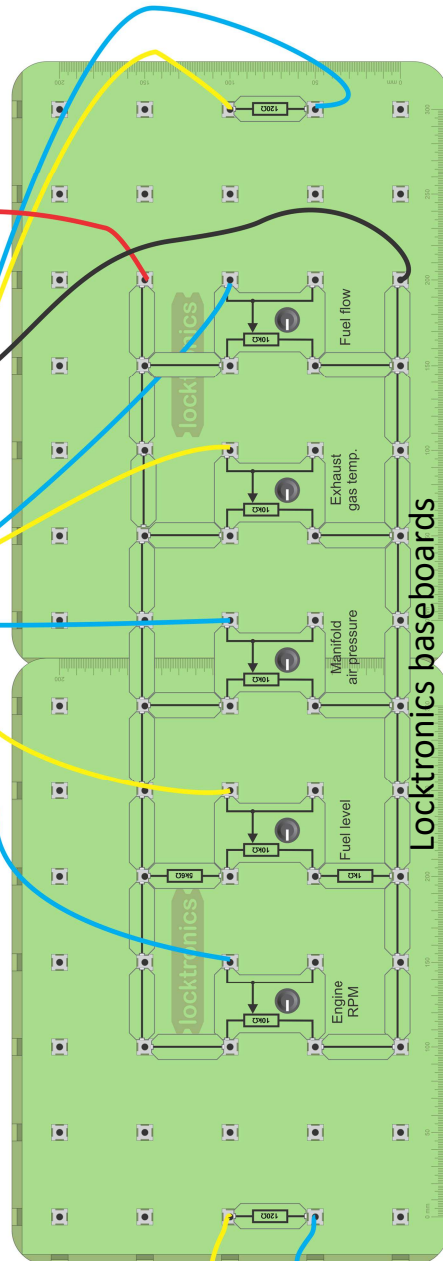
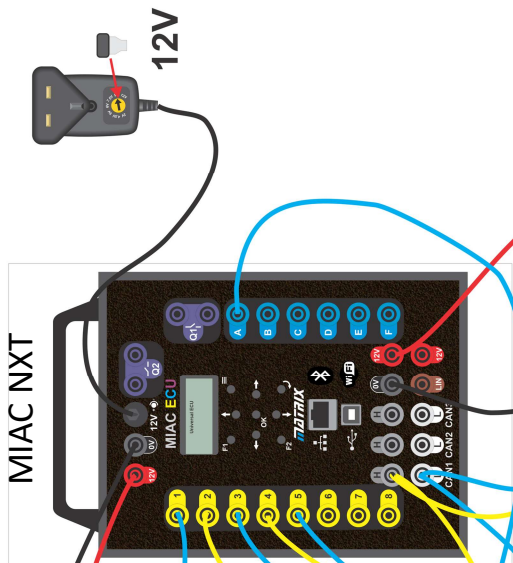
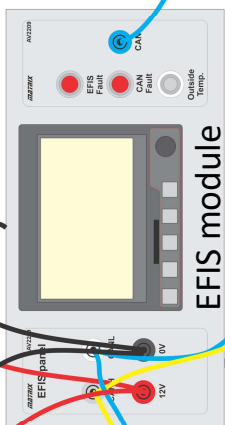


Full avionics system

Attitude and compass sensors



GPS sensor



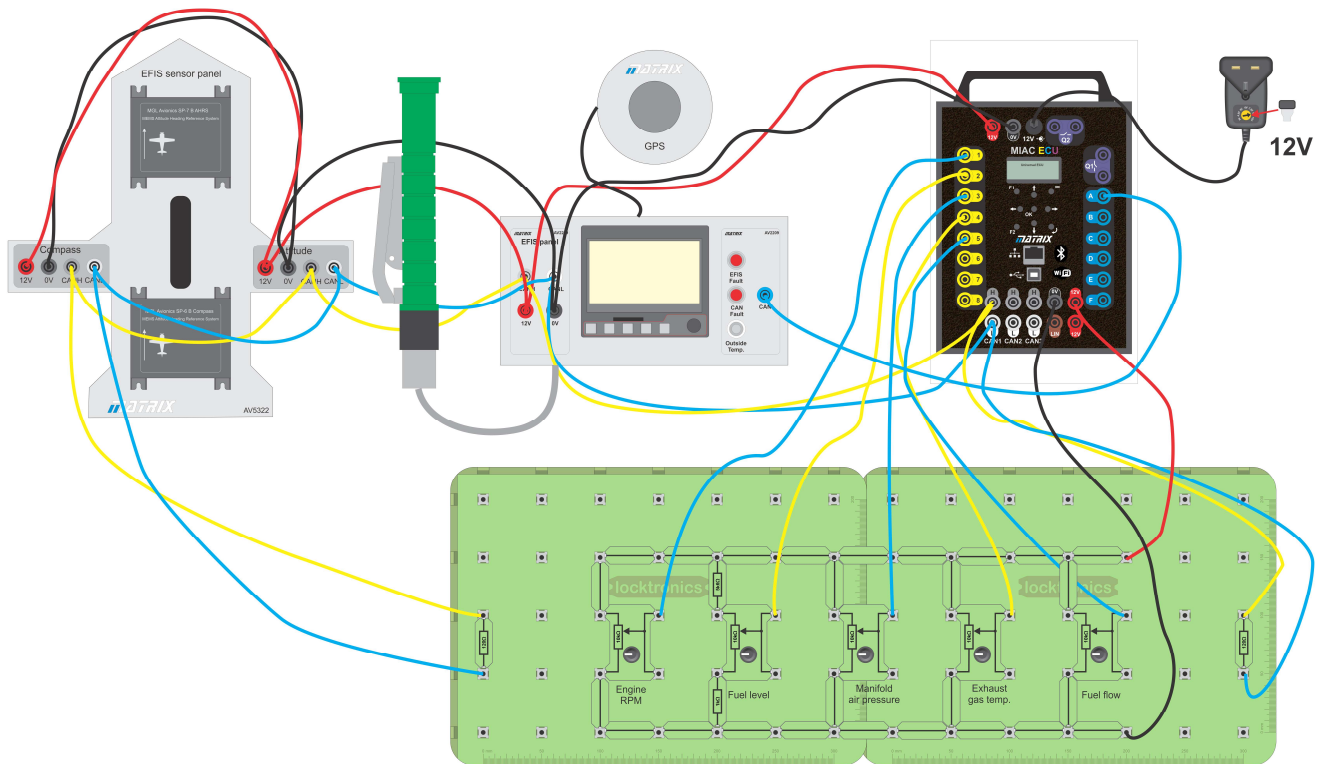
An Electronic Control Unit (ECU) is a rugged computer that is the backbone of computer systems in cars and in aircraft.

An ECU consists of a number of analogue and digital inputs and power and motor control outputs. They are rugged both electrically and mechanically and are connected together to form a network of control units in the aircraft.

The photograph shows an aircraft ECU.



Basic avionics system



Over to you:

- Familiarise yourself with the system components by reading through the Reference section.
- Build the system shown in the diagram above. To make this task easier, start by connecting the red and black power cables around the system. Then, connect up the CAN network and finally add the MIAC NXT connections to the Locktronics baseboards and to the EFIS module
- Turn the five pots into roughly central positions, set the power supply to 12V and switch it on.
- The EFIS module takes around 30s to initialise, displaying information about the maintenance and airframe timers. Wait until this is completed.
- Press the bottom left button on the EFIS module repeatedly and check what is shown on the different display screens. Refer to the information in the Reference section and the MGL avionics datasheet where needed.
- Examine the EFIS menu system, using the front soft keys and the rotary control. (The MGL avionics Xtreme manual describes these functions in detail.)

Over to you

Your system should now be fully functioning.

Verify the functionality as follows:

- Heading / attitude:
 - Check that rotating / tilting the compass and attitude sensor panel alters the EFIS module display.
- GPS:
 - Thirty-one satellites circling the earth form the basis of our Global Positioning System (GPS). The EFIS has a built-in GPS system. With the GPS antenna placed near a window to access satellite signals, make sure that the EFIS module can detect the presence of satellites. (This may take a few minutes to register.) Notice the latitude and longitude readings registered on the EFIS module. Now move the GPS antenna a few metres away, (but in a position where it is still able to pick up the satellite signal.) You should find that the latitude and longitudinal readings update.
- Air pressure:
 - The forward motion of the aircraft generates increased air pressure in the pitot tube. This is used to determine the aircraft's air speed. Slide the syringe piston in and out slowly to simulate this pressure change. You should see a change in the air speed reading shown on the EFIS module display.
- Air temperature:
 - Hold the thermistor on the front of the EFIS module and check that the temperature reading changes.
- Virtual sensors:
 - Adjust each potentiometer on the Locktronics baseboards in turn. Check that the corresponding EFIS quantity is updated as a result. The quantities are:
 - Engine RPM: 'RPM1';
 - Fuel Level: 'FL';
 - Manifold Air Pressure: 'MAP';
 - Exhaust Gas Temperature: ;EGT;
 - Fuel Flow: 'FF'.

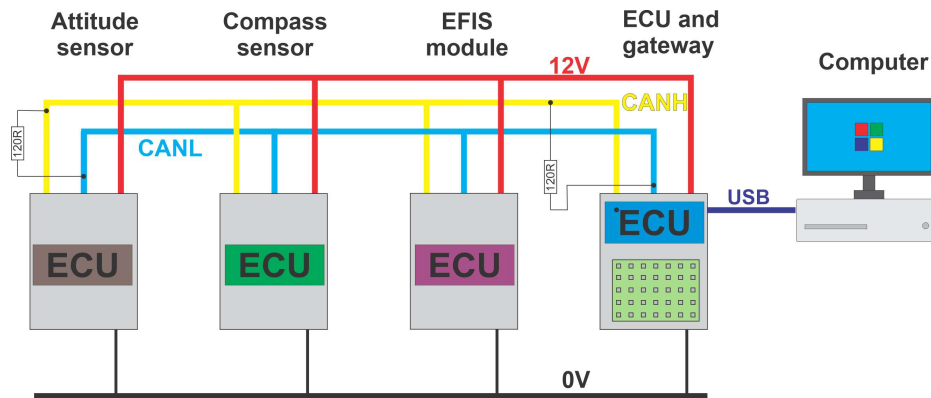
(Notice that the software uses a long time delay to 'damp' the measurement to stop erroneous fuel level readings caused by fuel slopping around in the tank.)

You should find that eventually, each pot triggers an EFIS module alarm, corresponding to the quantity being adjusted. The alarm can be cancelled by returning the relevant pot to its original position.

So what?

You have build a simple CAN bus based Electronic Flight information System - EFIS.

The wiring here looks complex, but isn't really. The diagram below shows a simplified representation:



The basic system consists of four ECUs:

- Attitude sensor;
- Compass sensor;
- EFIS module;
- MIAC NXT (acting as a gateway, connected to a PC).

These 'nodes' are each connected to the power rails, +12V and 0V, and network, CAN High and CAN Low rails.

Each end of the CAN bus network is terminated by a 120 ohm resistor.

Over to you:

- Use the EFIS module switches to explore the menu system. Make sure that you can locate the screens that display engine rpm, fuel level, manifold air pressure, exhaust gas temperature and fuel flow.
- Refer to the EFIS module reference section for additional help.

When diagnosing faults in multiplexed systems, another tool in addition to the usual multimeter and oscilloscope is needed.

Engineers need a diagnostic tool that shows sequential messages on the system in a text format. This allows them to compare what is happening on the system bus with a specification of what should happen.

These tools are either small handheld units with their own screen and keypad, or are based on laptops.

For this programme we have build a PC-based diagnostic tool, the Avionics Analyser, that allows you to view and send messages on the bus.

The photograph shows an ARINC diagnostic tool.



2D: Dashboard Panel

MATRIX CAN Bus Electronic Flight Information System

Time	ID	D0	D1	D2	D3	D4	D5	D6	D7
07:35.519	0012	01	00	06	00	07	00	00	00
07:33.292	0201	FF	00	E8	03	09	01	E8	03
07:35.786	0202	A8	01	14	02	4B	00	51	00
07:35.787	0203	80	00	00	01	80	01	00	02
07:35.787	0204	80	00	00	01	80	01	00	02
07:35.789	0205	80	00	40	01	80	00	00	02
07:35.789	0206	00	08	80	00	E9	00	E9	00
07:35.790	0207	11	00	75	02	00	00	00	00
07:35.983	0208	78	28	83	28	05	06	DE	0C
07:35.983	0241	6B	7C	A2	FE	BC	F3	01	FE
07:35.968	0281	FF	FF	0E	00	F1	03	F1	03
07:35.967	0282	FF	FF	00	00	03	00	FE	FF
07:35.968	0283	08	00	4A	00	47	00	00	04
07:35.923	0284	00	00	00	01	F2	03	F2	03

Click on decoded data to highlight messages.

Collated

MAP / "Hg

ENGINE RPM x1000

FUEL FLOW

TEMPERATURES

EGT 1: 424, EGT 2: 532, CHT 1: 75, CHT 2: 81

DECODED DATA

Measurements: -
 Fuel Flow = 33.0
 EGT 1 = 437
 EGT 2 = 548
 CHT 1 = 90
 CHT 2 = 97
 Oil Temp = 107
 Oil Press = 4
 Fuel Press = 3.7
 Coolant = 128
 Fuel Level = 43.9
 RPM 1 = 1726
 RPM 2 = 1728
 MAP = 31.7
 Heading: 318.5
 Roll = 0.08
 Pitch = 0.74

INJECT MESSAGE

0200 1 0 8
 CAN ID Repeats Period Length
 D0 D1 D2 D3 D4 D5 D6 D7
 Data

Zoom = 100.0% (601.24, 327.62)

Over to you:

- Connect the MIAC NXT to a computer using the USB lead, (no drivers are needed).
- Connect and switch on the power supply.
- Run the App "MI7009 Avionics Analyser.BAT" and press the **Go** button. You will see an App like the one shown above - the PC diagnostics screen.
- In the scrolling text box you see a sequence of messages, obtained from the CAN bus. You can pause the collection of messages to give you more time to view them.

So what?

You now have a mechanism that allows you to monitor the CAN bus system in the EFIS.

We are not using all the quantities that the EFIS can display, only the following subset:

- Engine RPM 1;
- Fuel Level (FL);
- Manifold Air Pressure (MAP);
- Exhaust gas temperature (EGT);
- Fuel flow (FF);
- Compass heading;
- Attitude - Pitch and Roll.

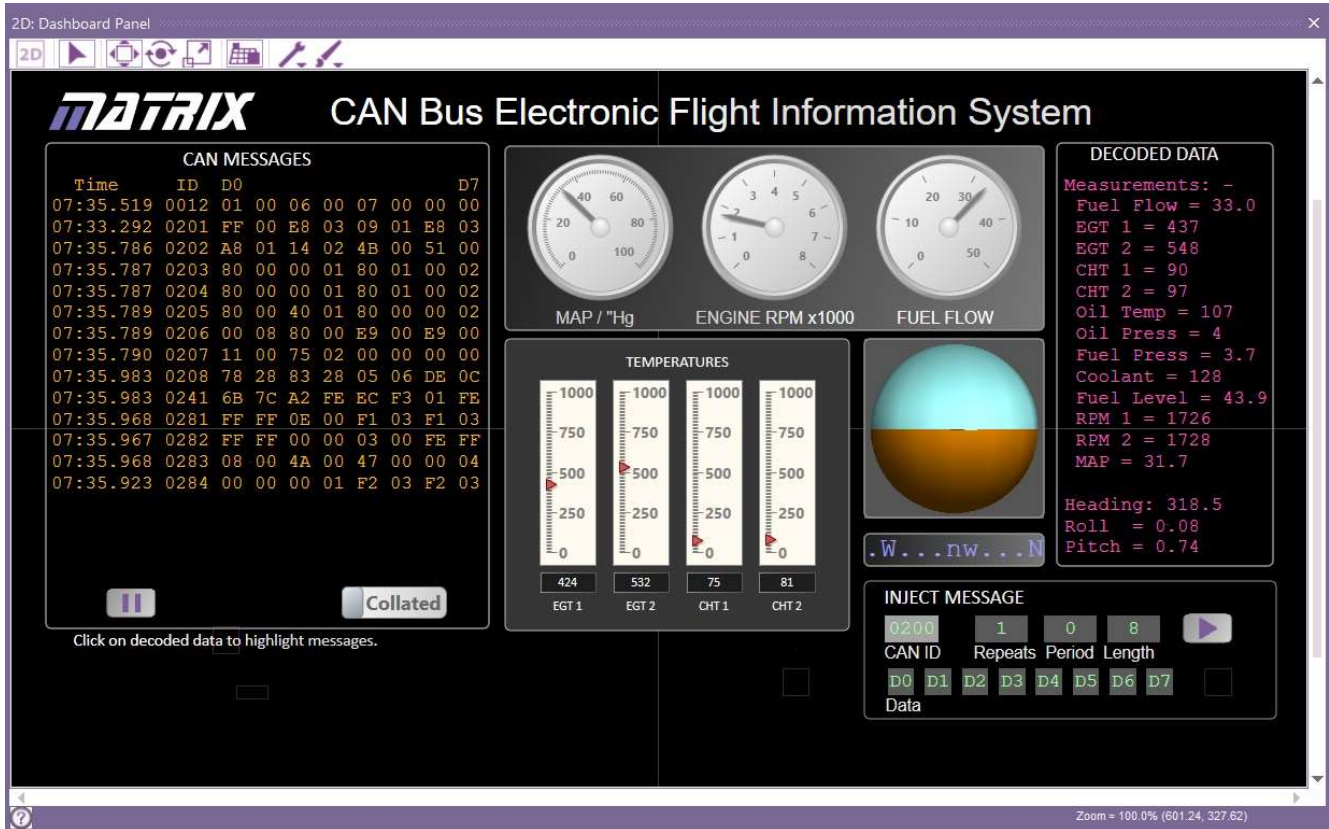
Other quantities that the EFIS can display include:

- Engine Gas Temperature 1 (EGT1);
- Engine Gas Temperature 2 (EGT2);
- Cylinder Head Temperature 1 (CHT1);
- Cylinder Head Temperature 2 (CHT2);
- Oil temperature (OT);
- Oil pressure (OP);
- Fuel pressure (FP);
- Coolant level (CL);
- Engine RPM 2.

Over to you:

- Using the Locktronics potentiometers, make sure that you can vary the following quantities on the EFIS portion of the PC diagnostic screen:
 - Engine RPM 1;
 - Fuel Level (FL);
 - Manifold Air Pressure (MAP);
 - Exhaust gas temperature (EGT);
 - Fuel flow (FF).
- Tilt and rotate the EFIS sensor panel. Verify that the compass and attitude display on the PC diagnostic screen change. If you find that **they** do not change then check your wiring.

Note: you may not get the compass readings you expect! This is linked to the angle of the attitude sensor.



CAN Messages section

The left-hand portion of the Avionics analyser display shows all the CAN messages present on the bus. There are two options: **Collated** and **All**.

- In **All** mode, the screen shows all messages on the bus, in the time order in which they arrive. You can pause this display.
- In **Collated mode**, the screen shows these messages collated into numerical order. There are not that many different messages on the CAN bus. You will use the **Collated** option most of the time.

Over to you:

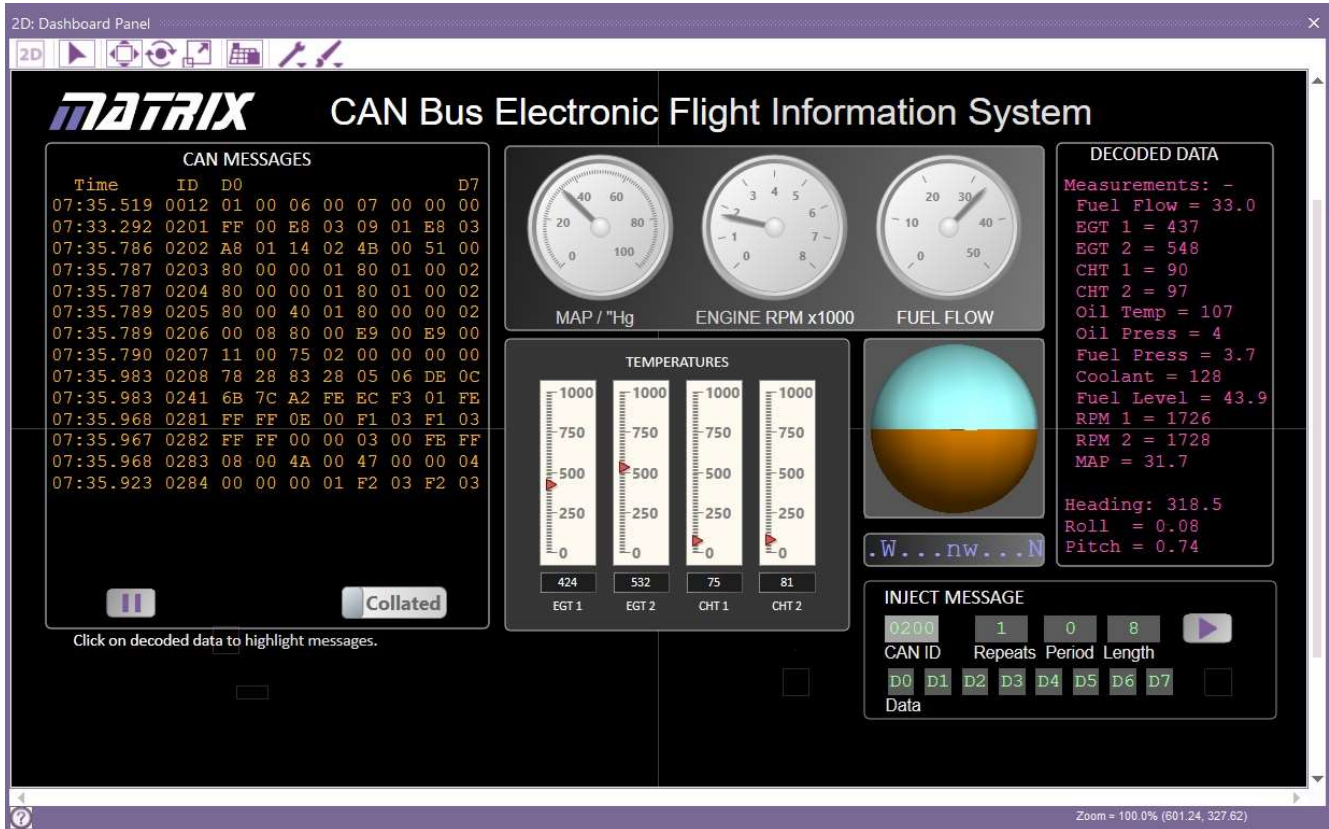
- Compare the output produced by both options - **All** and **Collated**.
- Press **GO** and **STOP** to see their effect on the CAN message data.

Instrument Mimics section

The Instrument Mimics section shows the quantities we can vary in the system in a graphical way - just like the EFIS itself.

Over to you:

- Vary the potentiometers and see the effects on the **User Interface** elements.
- Lift and rotate the EFIS sensor panels **to** see the compass and attitude display change.



Decoded Data section

The decoded data shows in numerical format all the quantities that the EFIS can display .

Over to you:

- With the CAN messages field in Collated mode, click on any line in the decoded data to see how that relates to the incoming CAN message.

Inject Message section

This allows you to set up a CAN data packet and then transmit that message a number of times at a set interval.

Over to you:

- With the CAN messages field in Collated mode, click on each of the fields in the CAN message and enter some random data. Set a number of repeats and an interval. Click on the **GO** button and check that the message appears in the CAN messages section.

Using the Console section

Over to you:

- Click on the Console button. This shows you the messages in a time-orientated way.
- Click on the *STOP* button - the data collection now stops. This allows you to examine a large amount of message history when diagnosing what is happening in the system.
- For now don't worry about the message information - we will look at that next.

ECUs are dumb. They send out CAN bus messages triggered either by an event or by an internal timer.

They are programmed to look for specific messages on the bus and take certain actions as a result. But they don't 'think' - they are simply programmable units.

The system designer gives the system apparent intelligence by making it behave according to a set of rules. The messages into and out of ECUs govern its behaviour.

The photograph shows a digital aircraft compass.



Over to you:

CAN bus messages consist of an eleven bit ID and eight data bytes in hex format: ID D0 D1 D2 D3 D4 D5 D6 D7

- With CAN messages in **Collated** mode, click on the **RPM1** line in the Decoded Data section of the PC diagnostic.
- Vary the **RPM** potentiometer on the Locktronics board between fully anticlockwise and fully clockwise. *Make a note of your readings in the table.*

Pot position	ID	D0	D1	D2	D3	D4	D5	D6	D7
Fully clockwise									
Fully anticlockwise									

So what?

'RPM' is transmitted on the CAN bus as message ID 0208 in bytes D0 and D1.

(D1 is the most significant byte, D0 is the least significant byte.)

The numerical range should be from 0000 to around 09FF in hex.

The engine management system designer builds in the relationship between the voltage created by the engine speed sensor and its rpm.

The data in D1 and D0 is a hex representation of RPM, converted into decimal and displayed on the EFIS module and on the Avionics Analyser screen.

The ECUs in the system are not particularly intelligent. They either measure a quantity and send messages about it at regular intervals or they look at messages at regular intervals and act on a quantity encoded in the message.

Over to you

- Each of the potentiometers on the Locktronics board generates a signal that is displayed on the EFIS.
- For each of these, decide on the ID used in the CAN message and on which data bytes are used. Then complete the table with your results.
- Using the CAN codes table in the Reference section make a comment on how the quantity is scaled.

Quantity	ID	D0	D1	D2	D3	D4	D5	D6	D7
Fuel level									
Mass Air Pressure									
Exhaust Gas Temperature									
Fuel flow									

- Fuel level scaling _____
- MAP scaling _____
- EGT scaling _____
- Fuel flow scaling _____

So what?

In a CAN bus system, there can be many thousands of messages. A manual that shows the messages and gives their meaning, or a diagnostic tool that does the same thing, is essential.

This CAN bus system consists of four nodes: EFIS, Compass, Attitude sensor and the MIAC NXT.

The E-blocks system is a fifth node that is not always used.

As the diagnostic tool plugs into the MIAC ECU, it is often known as a 'gateway' as it allows access to the outside world.

Multiplexed systems consist of local circuit components connected by the power, CAN High and CAN Low rails. They often look complex because of the mass of wires in them.

Over to you:

- Click on the **Console** button and the '**Message list**' tab. This view shows you the history of all messages on the CAN bus with a time stamp. You can use **GO** and **STOP** to control the collection of data in this mode.
- Look for message 0208 - 'RPM' and 'MAP' data. How often is it sent? _____
- Look for message 0202 - Temperature messages. How often is it sent? _____

So what?

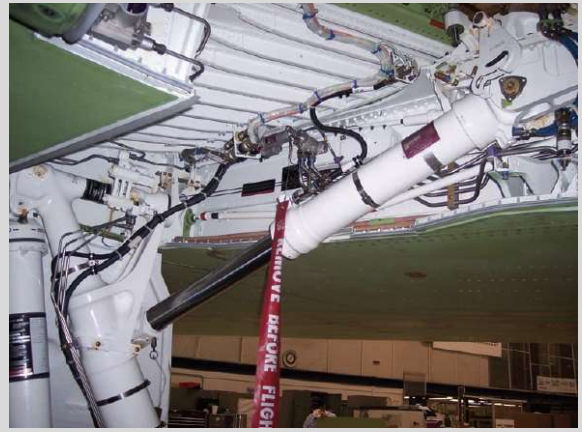
The system designer decides how often to send each kind of message, based on the need to have current data and based on how often the data is likely to change. This means that the frequency of transmission varies from one message type to another.

Messages are transmitted all the time - not just when a quantity changes. This means that there is always traffic on the bus and some kind of diagnostic tool is needed to decode it.

In a large commercial aircraft, there are hundreds of ECUs, with a complex messaging system between them.

Every subsystem has its own ECU. For example, the landing gear has several hydraulic pistons and associated valves and locking pins which are all computer-controlled via ECUs with feedback sensors to inform the pilot of the exact position of the landing gear.

The photograph shows landing gear on a commercial airliner.



Over to you:

- Remove the compass sensor from the system by disconnecting its CAN bus leads. (Make sure that the termination resistor is still connected to the CAN bus.)

The compass reading on the Xtreme display now has a red cross on it to indicate that this data is missing from the system.

- Using the Avionics Analyser, you are now going to recreate the message from the compass sensor manually. In the Avionics Analyser, create the following message:

	ID	D0	D1	D2	D3	D4	D5	D6	D7
Compass	0241	00	00	00	00	00	00	00	00

- Use the left-hand 'DISPLAY SCREEN' button on the EFIS module to show the compass screen, a graphical representation of the compass heading.
- On the 'Inject Messages' section of the Avionics Analyser, set the 'REPEATS' to be 1000 times and set the 'PERIOD' to be 50 - every 50ms. Press the Send button.
- You should see the compass graphic on the EFIS module alter to 0 degrees.
- Next, send the following messages:

	ID	D0	D1	D2	D3	D4	D5	D6	D7
Compass	0241	23	28	00	00	00	00	00	00
Compass	0241	46	50	00	00	00	00	00	00

- Looking at the Reference section devise a new message to get the compass to read 270 degrees.

So what?

The compass transmits the heading data on the CAN bus. The EFIS does not display the result directly but adjusts it depending on data taken from the Attitude sensor .

Over to you:

- Look at the 'All codes in the system' sheet in the Reference section. This lists all the CAN bus codes used in the system and shows how the quantities are transmitted.
- Use the left-hand button on the EFIS module to move to a screen where you can see the oil temperature and oil pressure settings.
- Set up the following message on the 'Inject Message' section of the PC Avionics software.

	<i>ID</i>	<i>D0</i>	<i>D1</i>	<i>D2</i>	<i>D3</i>	<i>D4</i>	<i>D5</i>	<i>D6</i>	<i>D7</i>
	0205	C8	00	80	01	C8	00	C8	00

- Set the 'REPEATS' to be 1000 times. Set the 'PERIOD' to '50' so that it send this message to the system every 50ms .
- You should see that the oil temperature is now 93C and the oil pressure is 5.1 bar.
- Determine the values of D0 and D1 for an oil temperature of 60C and send the corresponding message to test your understanding of this process.

So what?

The system designer decides how each of the message bytes are used.

In some cases, the data is in raw format.

In others it is scaled, or offset or adjusted to suit other parts of the system design.

To debug CAN systems you often need to know the details of the protocol.

With clever system design, it is possible to incorporate fault detection and diagnosis into the hardware and software.

This can include additional circuitry in and around each ECU that allows the ECU itself to detect faults.

Any faults that are detected can be reported by the in-flight display or through more specialised fault diagnosis tools like the Avionics analyser.

The photograph shows a hydraulic actuator.



Over to you:

- On the EFIS module, use the left-hand DISPLAY SCREEN button to move to a screen showing fuel level as a slide bar. This graphical representation of fuel level is controlled by a potentiometer on the Locktronics baseboards.
- Adjust the fuel level by turning the appropriate potentiometer from fully anticlockwise to fully clockwise. Notice that the EFIS module flashes orange when the fuel level is low. (The software uses a long time delay to stop erroneous readings caused by fuel slopping around in the tank.)
- Adjust the fuel level so that it is in the middle of its range and let it settle.
- Now remove the wire between the fuel level potentiometer and the MIAC.
- What reading do you get with the fuel level display when it settles? _____
- What warning do you get on the Avionics Analyser? _____
- Replace the wire. Add a short-circuit to the positive power rail.
- Clear the fault on the Avionics Analyser.
What reading do you get with the fuel level display ? _____
- What warning do you get on the Avionics analyser? _____
- Remove the short-circuit . Add a short-circuit to 0V. Clear the fault on the Avionics analyser.
- What reading do you get with the fuel level display ? _____
- What warning do you get on the Avionics Analyser? _____

So what?

The addition of two additional resistors to the fuel level sensor potential divider chain allows the MIAC to detect some faults in the system.

There is a pull-down resistor inside the MIAC. When the sensor is open circuit the voltage falls to 0V which is outside the normal reading range. The MIAC detects an error.

Similarly when the sensor is short-circuited to ground or to the positive rail, an error is detected.

A CAN bus - or ARINC bus - may be used to control a variety of systems including flight systems.

In addition, the aircraft may use more than one CAN or ARINC bus as there are advantages to separating messages onto different buses.

Ensuring that the system is fully functioning is very important. At start-up and periodically during operation the Gateway node usually uses a software integrity check routine to identify that a node is missing.

The photograph shows a camera control unit on a drone aircraft.



Over to you:

- Make sure that the EFIS system and Avionics Analyser software are connected and running properly.
 - Disconnect the Attitude sensor. What is the effect:
 - on the EFIS module;

 - on the Avionics Analyser ?
-
-
- Reconnect the Attitude sensor.
 - Clear the fault on the Avionics Analyser .

So what?

The Attitude sensor transmits a message onto the CAN bus every 50ms.

Both the EFIS module and the MIAC NXT have CAN node detection routines inside them. The EFIS module displays a red cross on the screen when the Attitude sensor CAN bus message is missing for around five seconds.

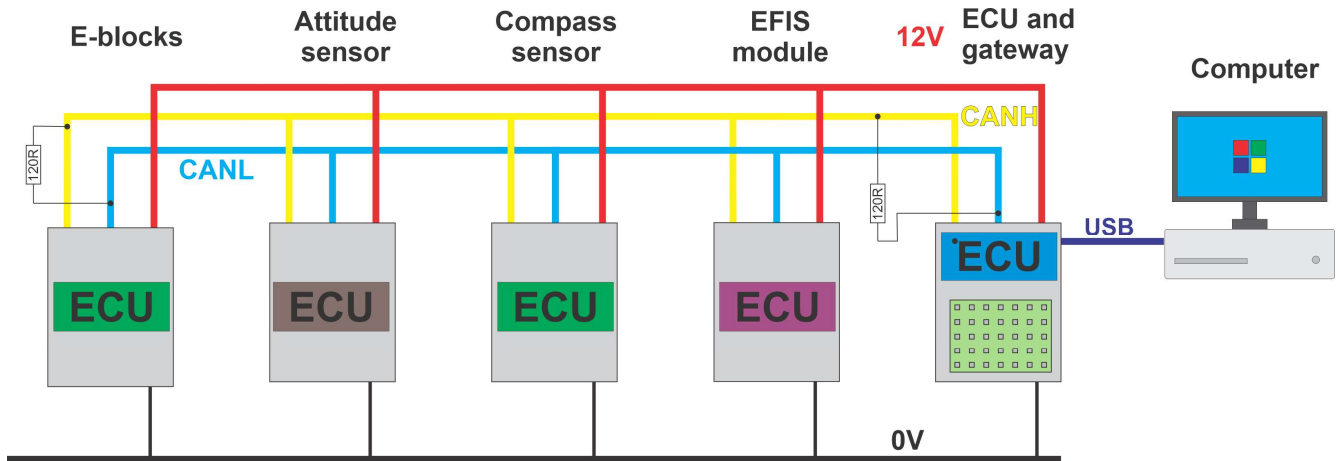
Should the Attitude CAN message reappear, the EFIS module recovers. Not so the MIAC however. This logs a fault which must be cleared using the Avionics Analyser software. Even if you power cycle the MIAC, the fault will still be logged so that a technician can take a look and see if there is loose wiring etc.

Detecting faults in multiplexed systems can be complex. Detecting faulty ECUs is usually requires a scan tool of some kind. Faults in wiring looms tend to occur where there is stress on the wiring, possibly caused by repeated movement.

This usually causes either an open circuit, a short circuit from one line on the bus to another, to the positive supply or to ground.

This section shows a sequence of investigations using a multimeter to detect this kind of fault in the CAN bus.

The photograph shows a radar system in a F/A 18 Hornet jet.



Over to you:

- Make sure that the power supply is turned off.
- The diagram above shows the schematic of the CAN bus in the system. Use a multimeter, on its resistance setting, to measure the resistance between the CAN High and CAN Low rails (using, for example, the MIAC CAN terminals).
What is this reading? _____ ohms
- Remove a connection to one of the CAN bus termination resistors.
What is the resistance reading now? _____ ohms

So what?

These results show you how to use a multimeter to check for an open circuit on the CAN bus.

Over to you:

- Replace the CAN bus connection.
 - Remove a CAN bus connection to the Compass sensor.
 - What is the resistance reading on the multimeter now? _____ phms
 - Restore the connection to the Compass sensor.
- Now short-circuit the CAN High and CAN Low connections on the Compass sensor.
- What is the resistance reading on the multimeter now? _____ ohms

So what?

A resistance of zero ohms between the CAN High and CAN Low rails is a pretty good indicator that there is a short circuit between these two wires at some point in the system.

In most systems, the 120 ohm termination resistors are inside the ECUs. A resistance reading of 60 ohms is an indication that the first and last ECUs on a CAN bus are correctly connected. It is not a guarantee that all nodes in the system are connected to the bus.

Over to you:

- Power up the system and start it.
- With the multimeter negative probe on 0V, make the following voltage measurements:

CAN Low voltage: _____ V

CAN High voltage: _____ V

So what?

CAN signals are based on around 2.5V.

CAN High pulses go between 2.5V and 5V so the voltage on CAN High rail is just above 2.5V.

CAN Low pulses go between 2.5V and 0V so the voltage on CAN Low rail is just below 2.5V.

Over to you:

- Power up the system and start it.
- With the multimeter negative probe on 0V, short circuit the CAN Low rail to 0V.

Make the following voltage measurements:

CAN Low voltage: _____ V

CAN High voltage: _____ V

So what?

If there is a short circuit to 0V, it's the connection with the lower of the two voltages that is shorted to 0V

Over to you:

- Power up the system and start it.
- With the multimeter negative probe on 0V, short circuit the CAN Low rail on the CAN bus to the 12V rail.
Make the following voltage measurements:

CAN Low voltage: _____ V

CAN High voltage: _____ V

So what?

If there is a short circuit to 12V, it's the connection with the higher of the two voltages that is shorted to 12V.

Challenge!

Based on your findings, produce quick guides to finding faults using a multimeter with the following headings:

- How to spot a single missing terminating resistor with a multimeter.
- How to spot both terminating resistors missing with a multimeter.
- How to spot a CAN High short circuit to CAN Low with a multimeter.
- How to spot CAN High short circuit to +V with a multimeter.
- How to spot CAN High short circuit to 0V with a multimeter.
- How to spot CAN Low short circuit to +V with a multimeter.
- How to spot CAN Low short circuit to 0V with a multimeter.
- How to spot CAN High open circuit with a multimeter.
- How to spot CAN Low open circuit with a multimeter.

Fault-finding can be a lengthy and complicated process especially in multiplexed systems because it is difficult to know what is happening inside the system.

However it is possible to do some basic checks with an oscilloscope.

The photograph shows sensors on the front of an aircraft.

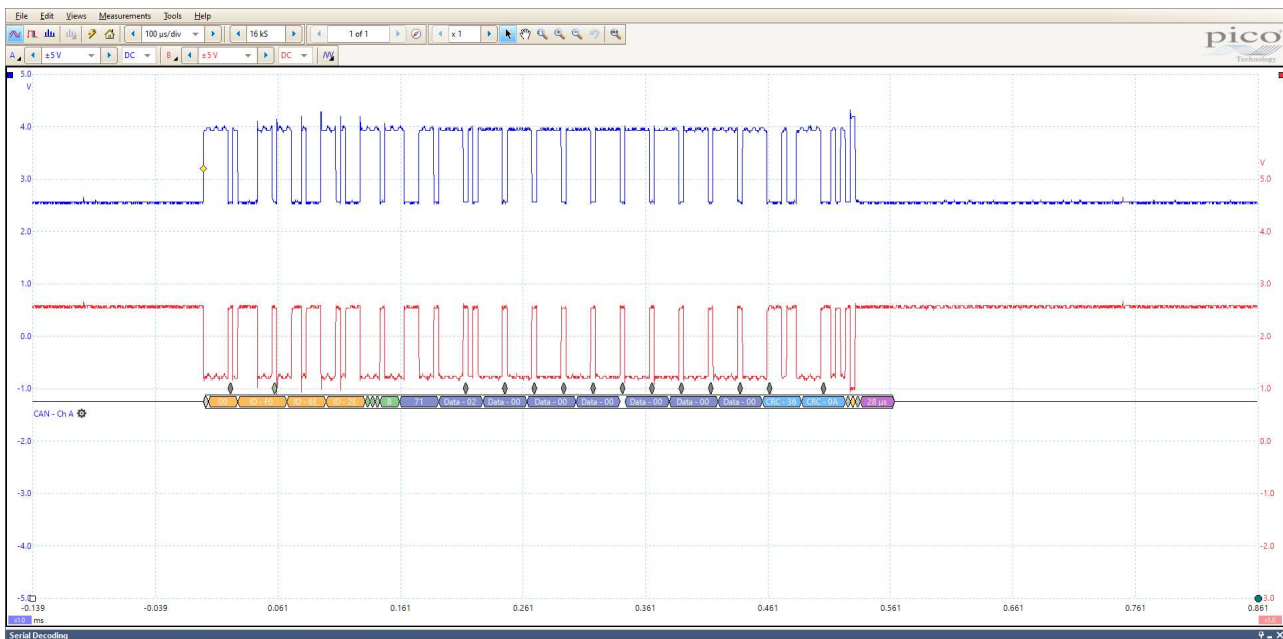


Over to you:

- Connect the oscilloscope ground probe to 0V.
- Connect oscilloscope channel A to the CAN High rail and channel B to the CAN Low rail.
- Trigger the oscilloscope so that it catches a single CAN bus waveform.
- Configure the oscilloscope to decode the CAN bus message.

If you are using a Picoscope, you can do this by selecting: TOOLS, SERIAL DECODING, CREATE, CAN. The baud rate for the CAN bus is 250kbaud and you can use either channel.

You should see something like this:



Note that the CAN High scale is on the left and CAN Low scale is on the right, on this image.

So what?

Notice:

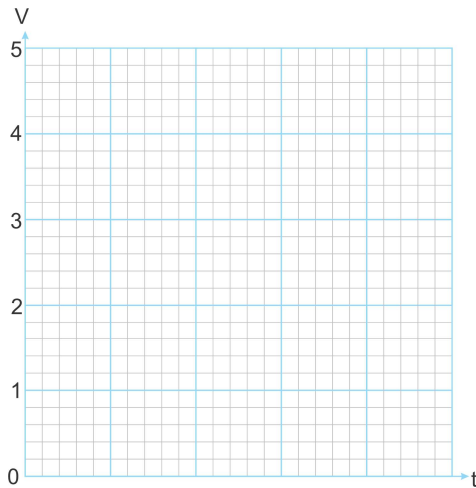
- when the CAN High voltage goes high, the CAN Low voltage goes low and vice-versa;
- quiescent CAN High sits at 2.5V and jumps between 2.5V and 4V with a message;
- quiescent CAN Low sits at 2.5V and jumps between 2.5V and 1V with a message.

Any significant variation from this could mean that there is an issue with the wiring.

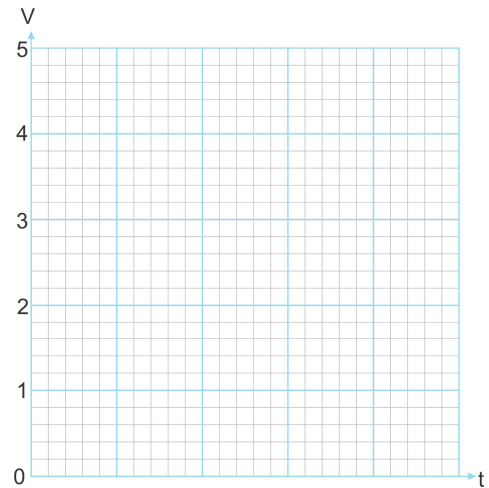
Over to you:

- Faults in an aircraft CAN bus system are either in wiring or in one of the nodes.
- Possible wiring faults include an open circuit in either CAN High or CAN Low, a short circuit between CAN High or CAN Low and ground, or the positive rail, or a short circuit from CAN High to CAN Low.
- Implement each of these faults in turn in your system. Don't worry about damaging the CAN bus circuits. The interface is designed to withstand such faults.
- Sketch the resulting waveforms for CAN High and CAN Low below. Concentrate on getting the high and low voltage levels right. The timing and message details are less important here.

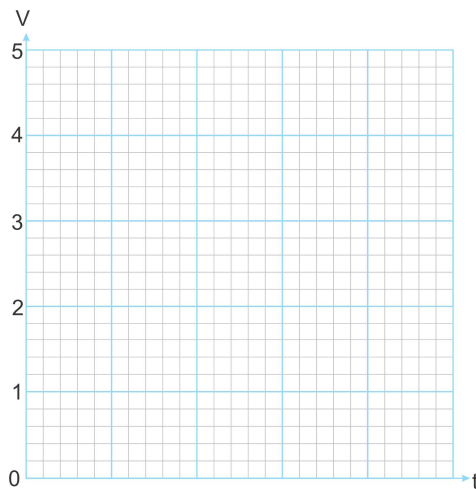
Fault 1: Single missing terminating resistor



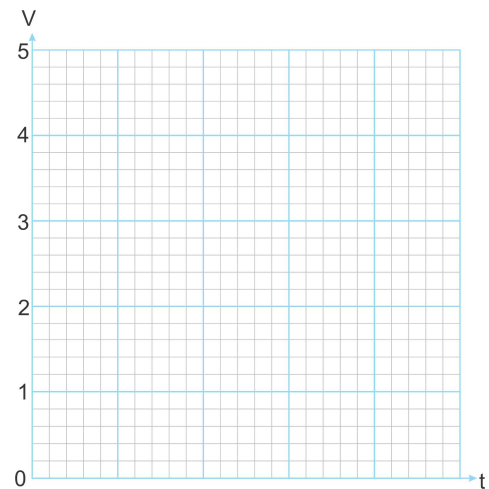
Fault 2: Both terminating resistors missing



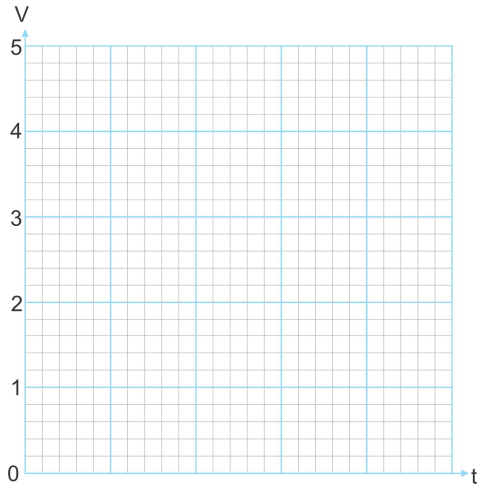
Fault 3: CAN High short circuit to CANL



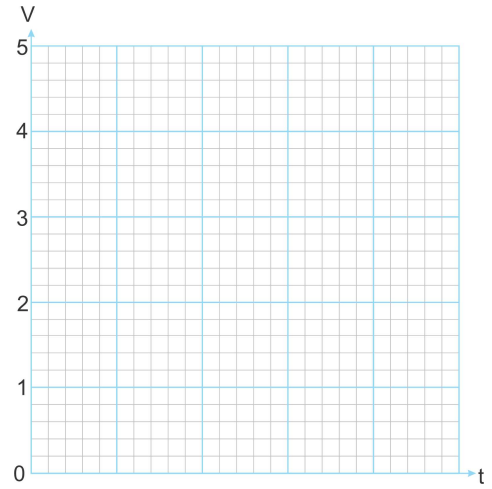
Fault 4: CAN High short circuit to +V



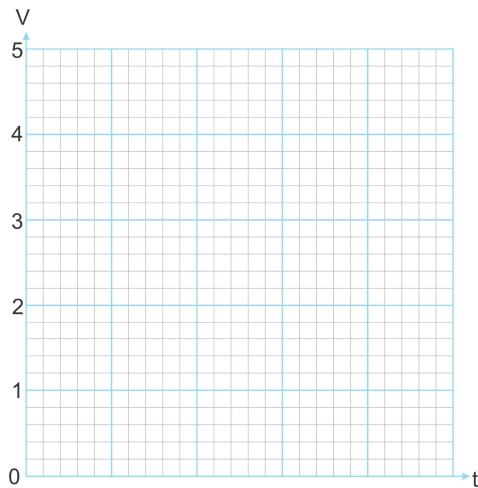
Fault 5: CAN High short circuit to 0V



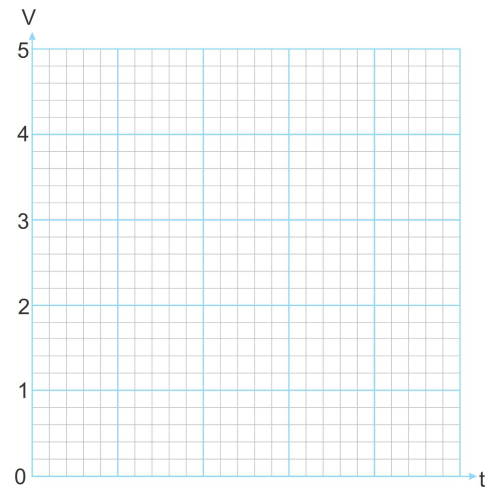
Fault 6: CAN Low short circuit to +V



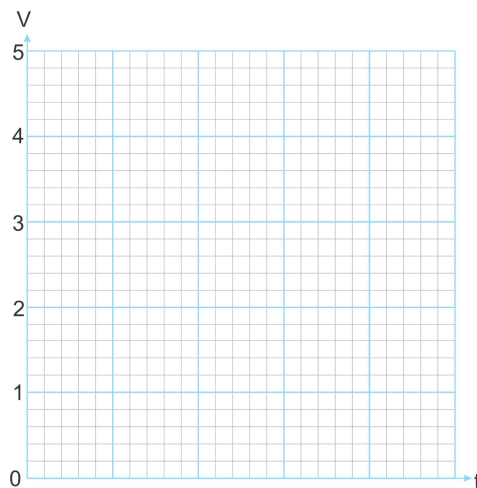
Fault 7: CAN Low short circuit to 0V



Fault 8: CAN High open circuit



Fault 9: CAN Low open circuit



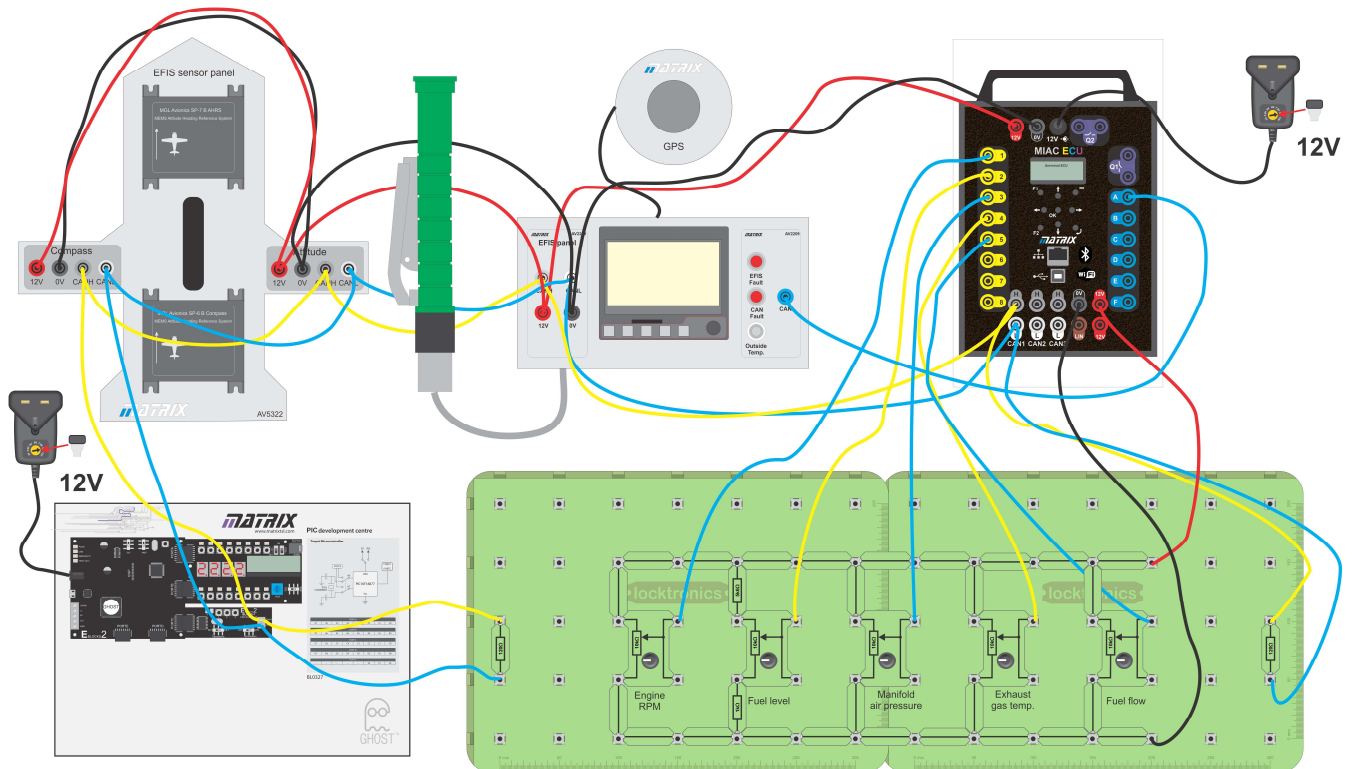
Challenge!

Based on your findings, produce quick guides to finding faults using a multimeter with the following headings:

- How to spot a single missing terminating resistor with a oscilloscope.
- How to spot both terminating resistors missing with a oscilloscope.
- How to spot a CAN High short circuit to CAN Low with a oscilloscope.
- How to spot CAN High short circuit to +V with a oscilloscope.
- How to spot CAN High short circuit to 0V with a oscilloscope.
- How to spot CAN Low short circuit to +V with a oscilloscope.
- How to spot CAN Low short circuit to 0V with a oscilloscope.
- How to spot CAN High open circuit with a oscilloscope.
- How to spot CAN Low open circuit with a oscilloscope.



Full avionics system



Over to you:

- Connect the E-blocks CAN bus development board to the CAN bus as shown in the diagram. Note that to get the leads to fit into the E-blocks board you will need to cut them and strip the ends. It may help if you tin the ends with a soldering iron.
- Navigate to a screen on the EFIS module that shows the compass as a graphical object.
- Disconnect the Compass from the sensor assembly. The EFIS module still has a red cross showing that a sensor is missing. The MIAC NXT warning light will come on.
- Power up the E-blocks system, running program 10, using a 12V supply. This is a simple compass simulator where the position of the potentiometer on the Combo board determines the angle of the compass .
- The EFIS module red cross should go out. Clear the 'Compass sensor missing' fault on the Avionics Analyser.
- Using the console, identify message 0241.
How does this differ from the message generated by the compass module?

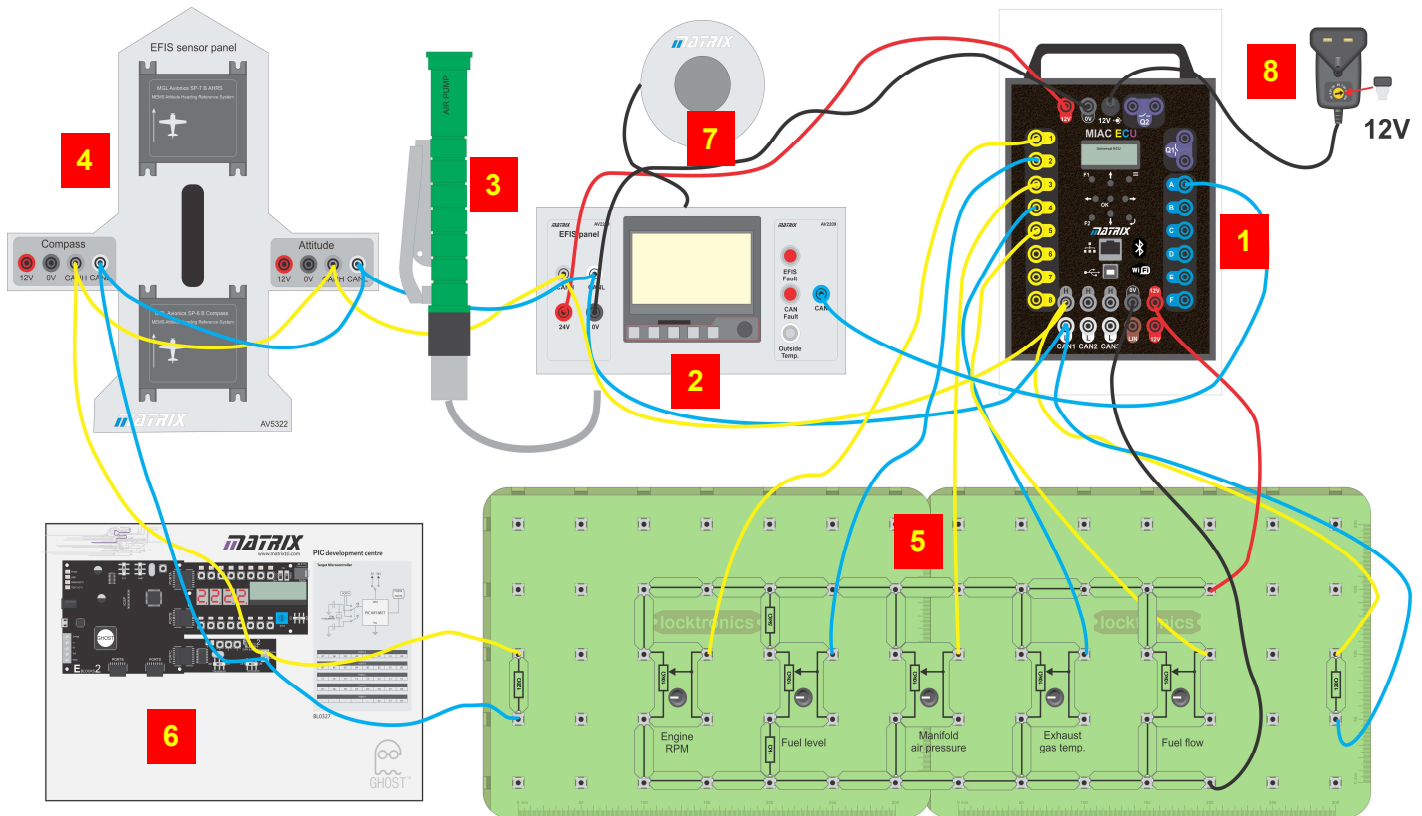
So what?

The E-blocks system simulates the behaviour of the compass module in that it generates messages that are correctly structured, sent on the CAN bus, with the right timing and with appropriate content.

The only difference is that rather than a compass sensor, it uses a potentiometer.

Whether CAN signals are created using an ECU, bare-board electronics, or a real sensor, the CAN signal has just the same structure and uses the same voltage ranges.

Reference



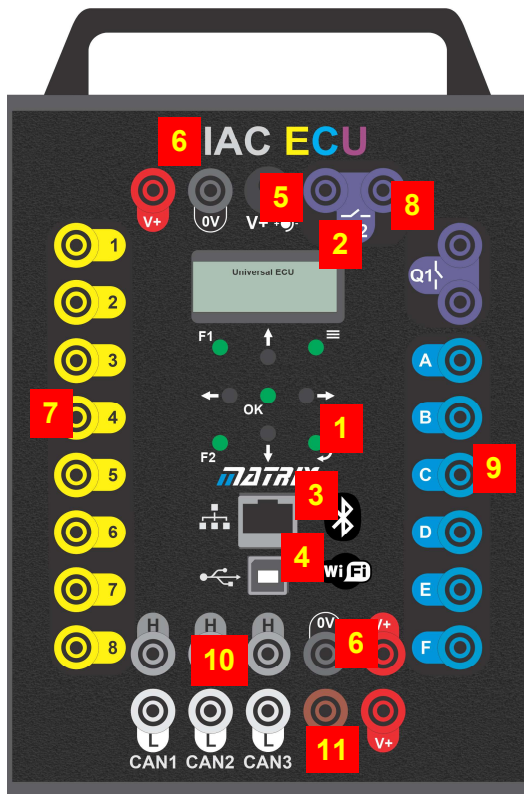
The diagram above shows all the parts of the EFIS system.

They are:

1. The MIAC NXT - that has eight analogue / digital inputs, two high power relays, six high power transistor outputs, three CAN bus interfaces and a LIN bus interface. It also has a LCD display, keypad, integrated USB, Wifi, Ethernet and Bluetooth communications systems. This is a rugged educational product designed by Matrix TSL for teaching and learning. Further details are given below.
2. The EFIS module includes the EFIS display itself, with 4mm connectors for power and CAN connections and for a warning indicator controlled by the MIAC NXT. See below for further details.
3. A syringe and tubing used to alter the air pressure on the EFIS module internal pressure sensor.
4. The EFIS sensor assembly with Attitude and Compass sensors. Each sensor has 4mm sockets for power, ground and CAN bus connections. More details are given below.
5. The Locktronics baseboards with components
6. An E-blocks II development system with microcontroller upstream board, Combo board and CAN bus interface board. Further details are given below.
(This is not integral to this module but is used in the module on 'Microcontroller Programming and Systems in Aviation' and provides alternative compass data via the CAN bus.)
7. The GPS antenna plugs directly into the back of the EFIS module using a micro BNC plug/socket.
8. A 12V power supply.

Not shown:

- the PC;
- the Avionics Analyser software (detailed below);
- Flowcode embedded software (detailed below).



Key

1. keypad
2. LCD display
3. RJ45 connector
4. USB connector
5. 12V 2.1mm positive inner connector
6. power/ ground connectors
7. eight digital or analogue inputs
8. two relay connectors
9. six transistor PWM power outputs
10. three CAN bus connectors
11. LIN connector

Not shown:

- Internal Bluetooth module
- Internal Wi-fi module

'MIAC' stands for Matrix Industrial Automotive Controller. 'NXT' stands for NeXT generation.

MIAC NXT is designed for educational use, allowing students to experiment with various types of control system.

Each MIAC has eight analogue / digital inputs, two relay outputs, and six transistor outputs.

It offers several methods of communications:

- three CAN bus interfaces and a LIN bus interface, aimed at communications in automotive applications;
- a USB interface, a RJ45 Ethernet interface and internal Bluetooth and Wi-fi modules.

The unit is controlled by a powerful PIC24 series microcontroller from Arizona Microchip.

The input signals are fed into a signal conditioning circuit. Software dictates their operation as either analogue or digital inputs. The input voltage range is 0 to 12V DC, making MIAC NXT compatible with industry standard sensors. The inputs are not optically isolated.

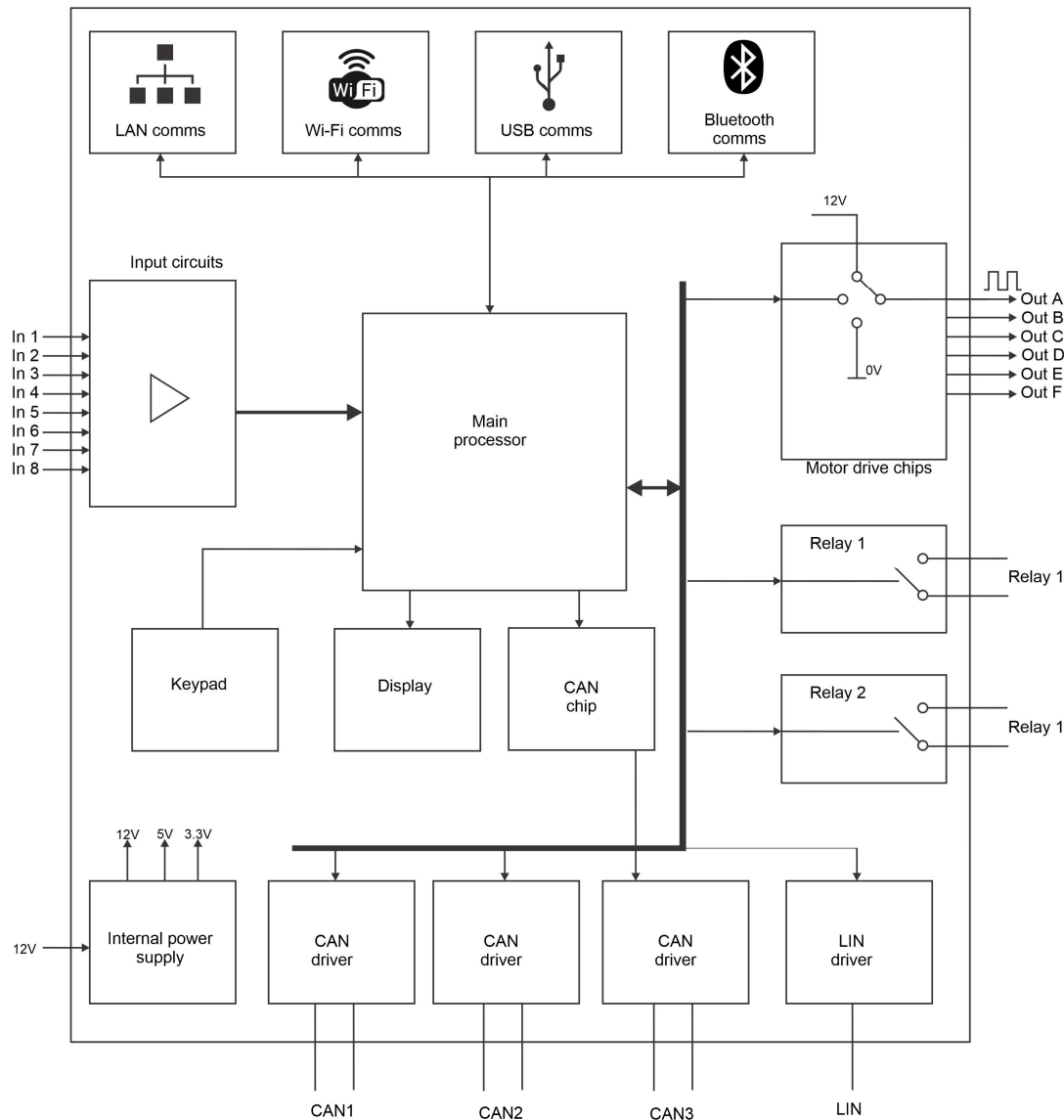
Two outputs from the PIC24 processor are fed into a power stage, providing current amplification, before they are connected to four separate relays. Fusing for these relay circuits should be provided externally to limit relay current to 8A AC or DC.

Six additional outputs are fed into a motor-driver stage, including a current monitor with shutdown circuitry, to limit the output current and protect the motor-driver chip in case of short circuits. The motor driver circuits are all PWM compatible.

The internal processor also connects to three CAN bus driver circuits and a LIN bus circuit, allowing a number of MIAC NXTs to be connected together to form a control network.

Great care has been taken to ensure that the MIAC NXT is electrically rugged. It is possible to short circuit any input to any output or any output to another without the unit failing.

A four-line monochrome Graphical LCD display and a nine key keypad allow easy control and monitoring of processes.



Internally, the MIAC NXT is controlled by a powerful 24 series PICmicro device, which includes a USB port and filtered USB circuitry to allow fast programming and USB communications. This PIC device is pre-programmed to allow users to download PIC compatible hex code into the device.

Inputs are fed into signal-conditioning circuits which allow them to be used as either analogue or digital inputs.

The PIC processor includes two internal CAN bus driver circuits, fed to external CAN line driver circuits for CAN buses 1 and 2. An additional CAN driver chip and driver forms the third CAN bus. (Three CAN buses are needed for some automotive applications.) A simple LIN line driver circuit is included for LIN bus communications.

Bluetooth and Wi-fi modules provide communications for control, data transfer and reprogramming.

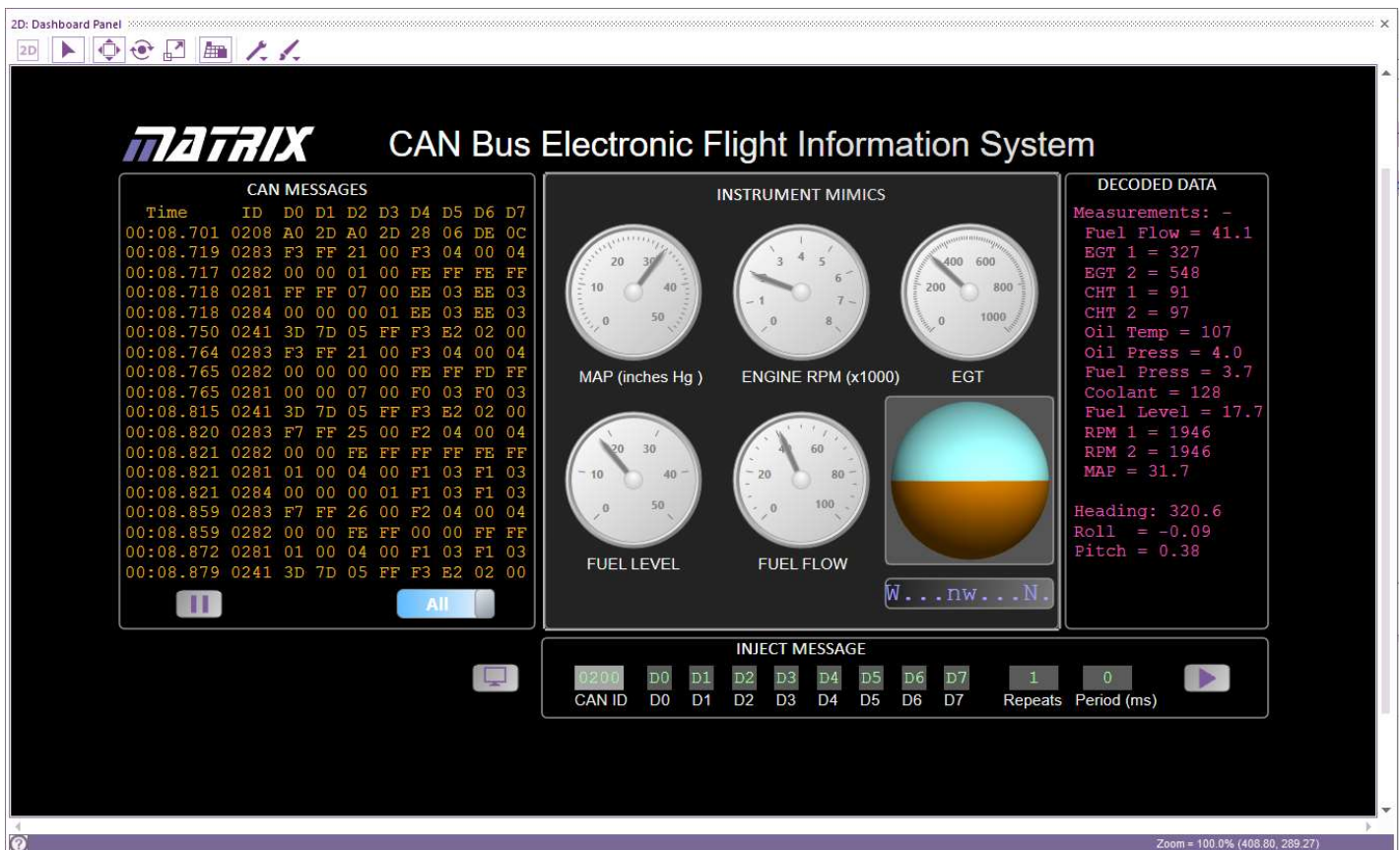
MIAC NXT can be powered from a 12V to 24V DC supply connected to the 2.1mm power jack (POWER), or to the power supply terminals (V+, 0V), wired in parallel with it.

Internal power supply circuitry provides 12V, 5V and 3.3V power rails to all parts of the MIAC NXT.

The Matrix part number is MI5550.

It is supplied with a HP2666 12V power supply with mains connectors for UK, EU, USA and a HUSB: USB A lead.

For further details on the MIAC NXT, please see the MI3728 datasheet.



The Avionics Analyser software displays data received from the CAN/bus system, using the MIAC NXT as an interface.

The MIAC NXT uses a HID USB system so that no drivers are needed to connect it to a PC.

(Note: MIAC NXT works only with Windows PCs.)

MIAC NXT is programmed via an Application Programming Interface (API), used by the Windows EFIS diagnostic software. This provides a number of standard functions for PC programs, called via USB, Wifi, or Bluetooth.

The API product code is MI8975 and it is available via the Matrix web site. (If you have bought the EFIS system, it comes pre-installed.)

You can program the MIAC NXT with the API using the 'Mloader' utility, available as a free download from the Matrix web site. Load the API into 'Mloader' and send the program to the MIAC NXT.

The PC needs the Flowcode App Developer firmware program (shown above,) to display CAN bus traffic on the EFIS system. This software is product code MI7009 also available as a free download from the Matrix web site.

To run the software download it to the PC and double click on the "MI7009 Avionics Analyser.BAT" file.

2D: Dashboard Panel

MATRIX CAN Bus Electronic Flight Information System

CAN MESSAGES

Time	ID	D0	D1	D2	D3	D4	D5	D6	D7
00:08.701	0208	A0	2D	A0	2D	28	06	DE	0C
00:08.719	0283	F3	FF	21	00	F3	04	00	04
00:08.717	0282	00	00	01	00	FE	FF	FE	FF
00:08.718	0281	FF	FF	07	00	EE	03	EE	03
00:08.718	0284	00	00	00	01	EE	03	EE	03
00:08.750	0241	3D	7D	05	FF	F3	E2	02	00
00:08.764	0283	F3	FF	21	00	F3	04	00	04
00:08.765	0282	00	00	00	00	FE	FF	FD	FF
00:08.765	0281	00	07	00	F0	03	F0	03	
00:08.815	0241	3D	7D	05	FF	F3	E2	02	00
00:08.820	0283	F7	FF	25	00	F2	04	00	04
00:08.821	0282	00	00	FE	FF	FF	FF	FE	FF
00:08.821	0281	01	00	04	00	F1	03	F1	03
00:08.821	0284	00	00	00	01	F1	03	F1	03
00:08.859	0283	F7	FF	26	00	F2	04	00	04
00:08.859	0282	00	00	FE	FF	00	00	FE	FF
00:08.872	0281	01	00	04	00	F1	03	F1	03
00:08.879	0241	3D	7D	05	FF	F3	E2	02	00

INSTRUMENT MIMICS

MAP (inches Hg) ENGINE RPM (x1000) EGT

FUEL LEVEL FUEL FLOW

DECODED DATA

Measurements: -
 Fuel Flow = 41.1
 EGT 1 = 327
 EGT 2 = 548
 CHT 1 = 91
 CHT 2 = 97
 Oil Temp = 107
 Oil Press = 4.0
 Fuel Press = 3.7
 Coolant = 128
 Fuel Level = 17.7
 RPM 1 = 1946
 RPM 2 = 1946
 MAP = 31.7

Heading: 320.6
 Roll = -0.09
 Pitch = 0.38

INJECT MESSAGE

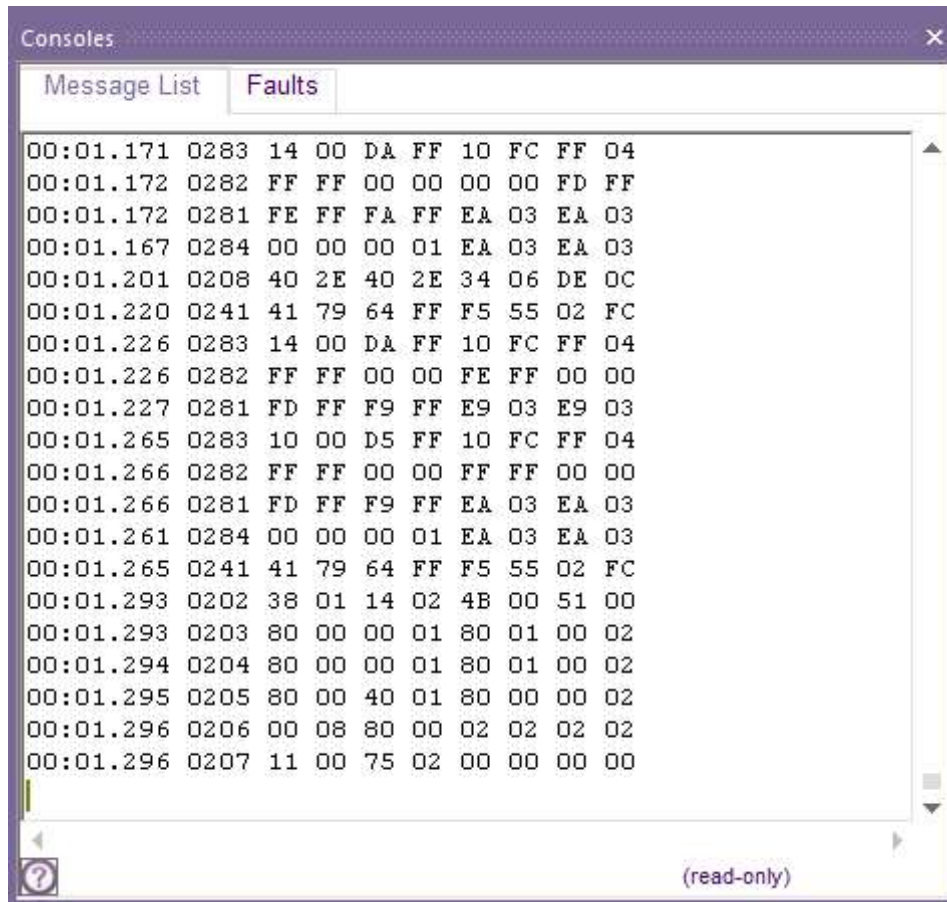
CAN ID: D0 D1 D2 D3 D4 D5 D6 D7 Repeats: 1 Period (ms): 0

Zoom = 100.0% (408.80, 289.27)

The Avionics Analyser software shows displays activity on the CAN bus . It is partly a diagnostic tool and partly a visual representation of the CAN bus activity and significant parameters.

The items displayed are as follows:

1. Virtual instruments displaying measured quantities in the EFIS system: MAP, Engine RPM, Exhaust Gas Temperature (EGT), Fuel Level and Fuel Flow.
2. Artificial horizon
3. Compass
4. Decoded data transmitted in the system.
5. Data display field - this can be set to display either 'All' or 'Collated' data using the 'Collated / All' button.
6. 'Collated / All' button - causes the data display field to show either all messages, with a time stamp, or to collate a list of all messages transmitted in the system.
7. Pause button
8. Console view/hide button
9. 'Inject message' section:
 Highlight the field and click on the ID and D0 to D7 fields individually to enter the data you want to send. Then set the number of times this is to be repeated, ('Repeats') and the time interval, ('Period'), between repetitions. Press the 'Send' button to inject this message into the CAN bus. The transmitted message appears in the Data Display field or on the Console.
10. Fault clear button: (only displayed when there is a fault).

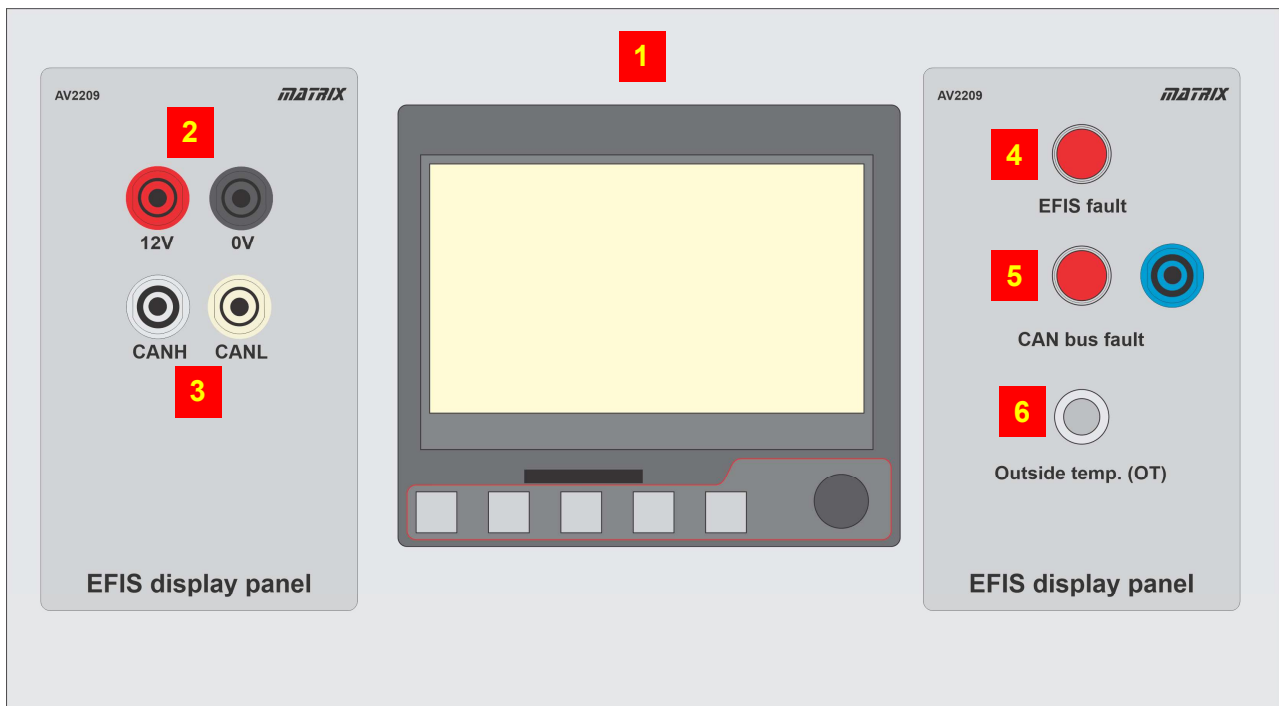


Selecting VIEW...CONSOLES gives a different view of the message data on the system, called the Console view.

This displays the messages on the CAN bus, but recorded over a much longer period, useful in debugging the system. You can use the scroll bar to view all CAN messages sent in the system.

Each message has a time stamp. This is useful in showing in detail the history of the messages on the bus.

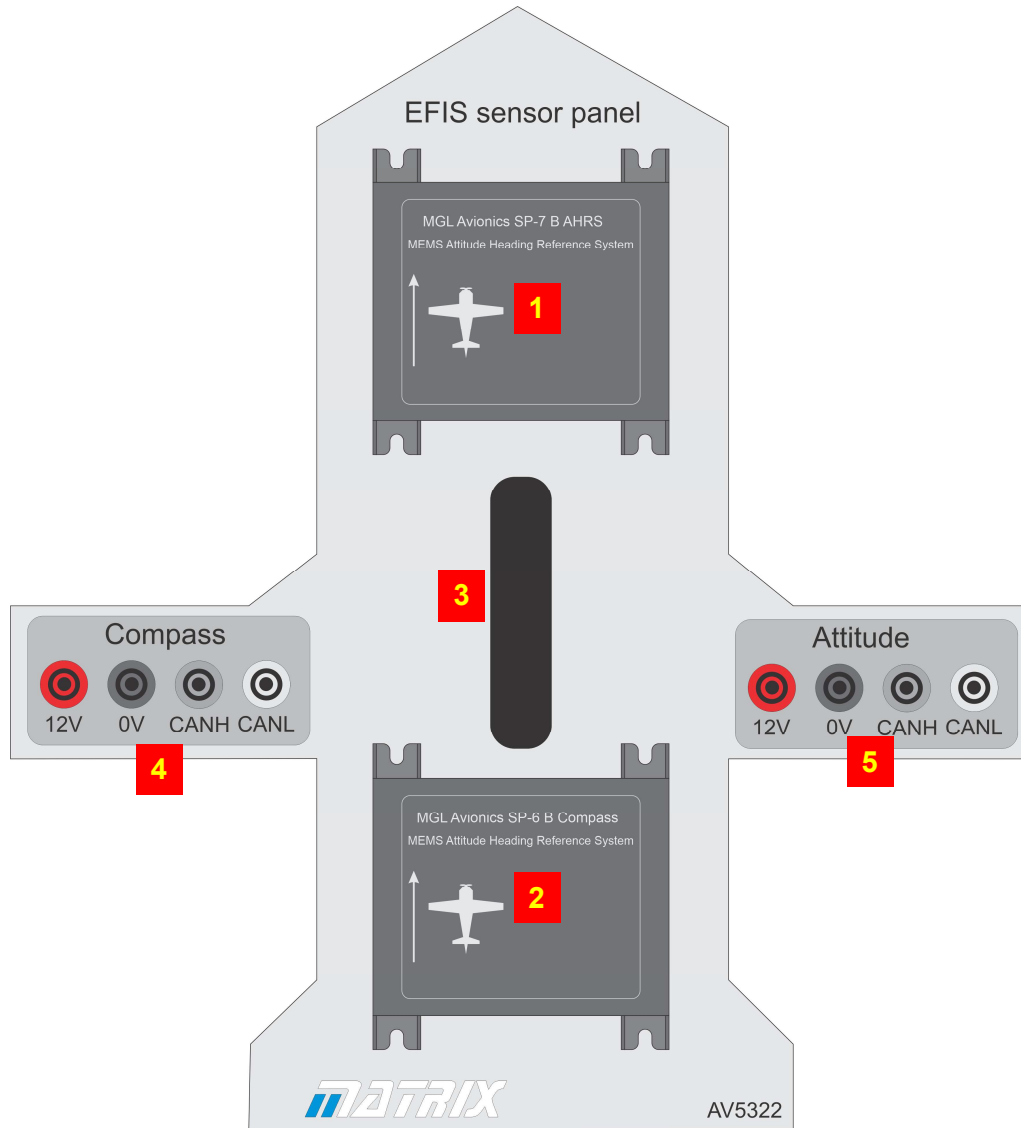
In particular, it shows how data is transmitted from one bus to another via the Gateway, for example from the LIN bus to the CAN bus.



The EFIS system consists of an MGL Avionics EFIS module, 4mm sockets, warning lights and a temperature sensor.

The parts are:

1. EFIS module - see below for further details;
2. Power sockets;
3. CAN bus sockets;
4. EFIS fault light - generated by the EFIS module itself;
5. System fault light - generated by the MIAC NXT, fed from Output 1 of the MIAC NXT via **the** 4mm socket;
6. Temperature sensor - connects behind the panel directly to the EFIS module.



The sensor assembly consists of an MGL Avionics Attitude sensor and an MGL Avionics Compass sensor

These are mounted on an aluminium carrier with a handle that allows you to adjust the angle and direction of the sensors easily .

The items are:

1. EFIS Attitude sensor;
2. EFIS Compass sensor;
3. Carrier;
4. Power and CAN bus sockets for Compass sensor;
5. Power and CAN bus sockets for Attitude sensor.



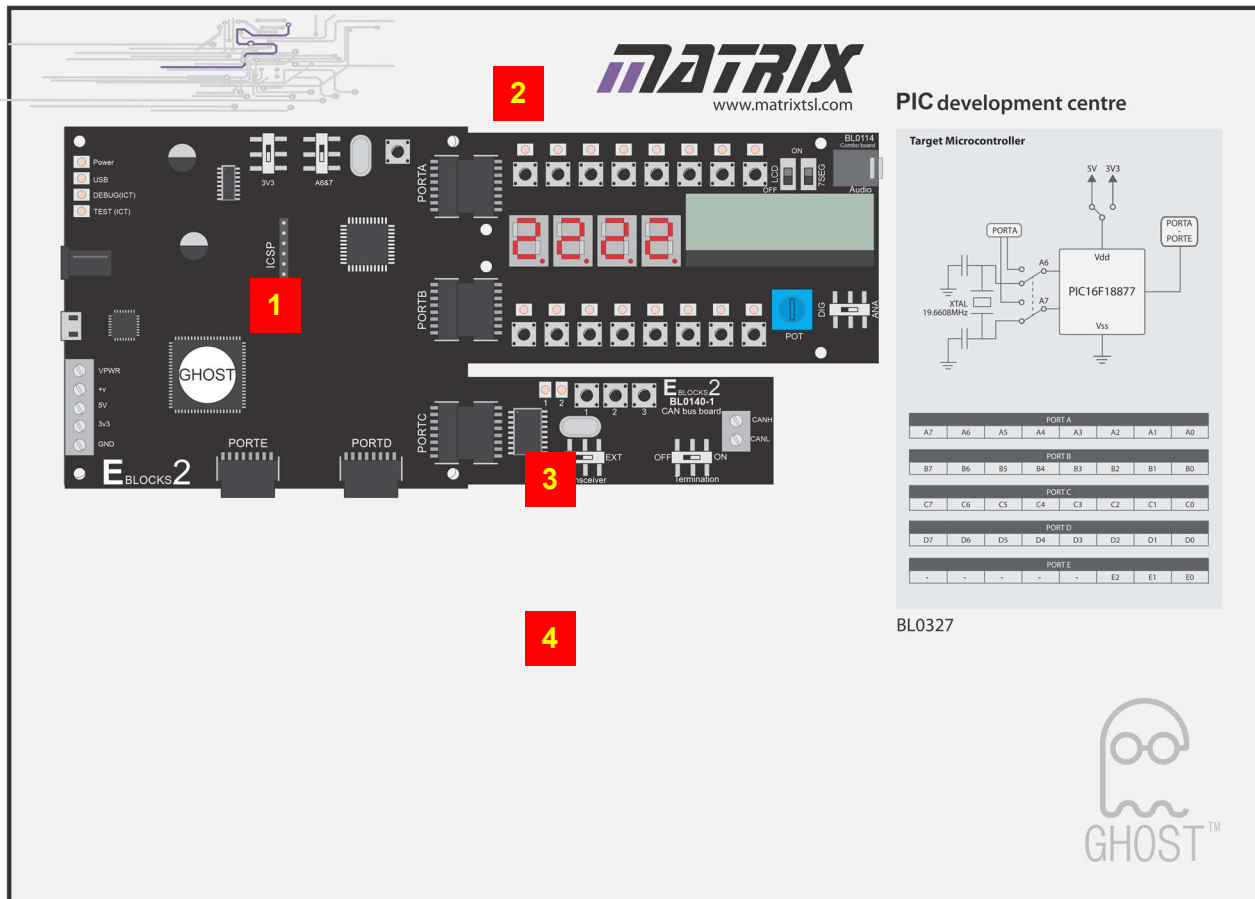
The XTreme is a compact, multifunction electronic flight information system intended as the main flight instrument in smaller aircraft or as a backup/secondary flight instrument in larger aircraft.

It is manufactured by MGL Avionics.

The XTreme offers all the functionality necessary to replace several flight and engine monitoring instruments. All information is displayed in an easy-to-read format on a high resolution, sunlight-readable 4.3" colour display, offering a wide viewing angle.

The XTreme's light weight, small size and high level of functionality makes it an excellent choice for all types of non-certified aircraft. The unit is very easy to use and students should investigate the functions of the buttons on the front panel.

For the full manual see <https://www.mglavionics.co.za/>



The E-blocks system allows you to create CAN bus messages at a microcontroller level.

The parts are:

1. PICmicro microcontroller upstream programming board.

This allows you to reprogram the resident PIC device using Flowcode.

It presents all PIC input/output pins on multiway headers so that other boards can be added to the system.

This board includes full in-circuit debug facilities and instrumentation to show what is happening on the PIC device pins.

2. E-blocks Combo board.

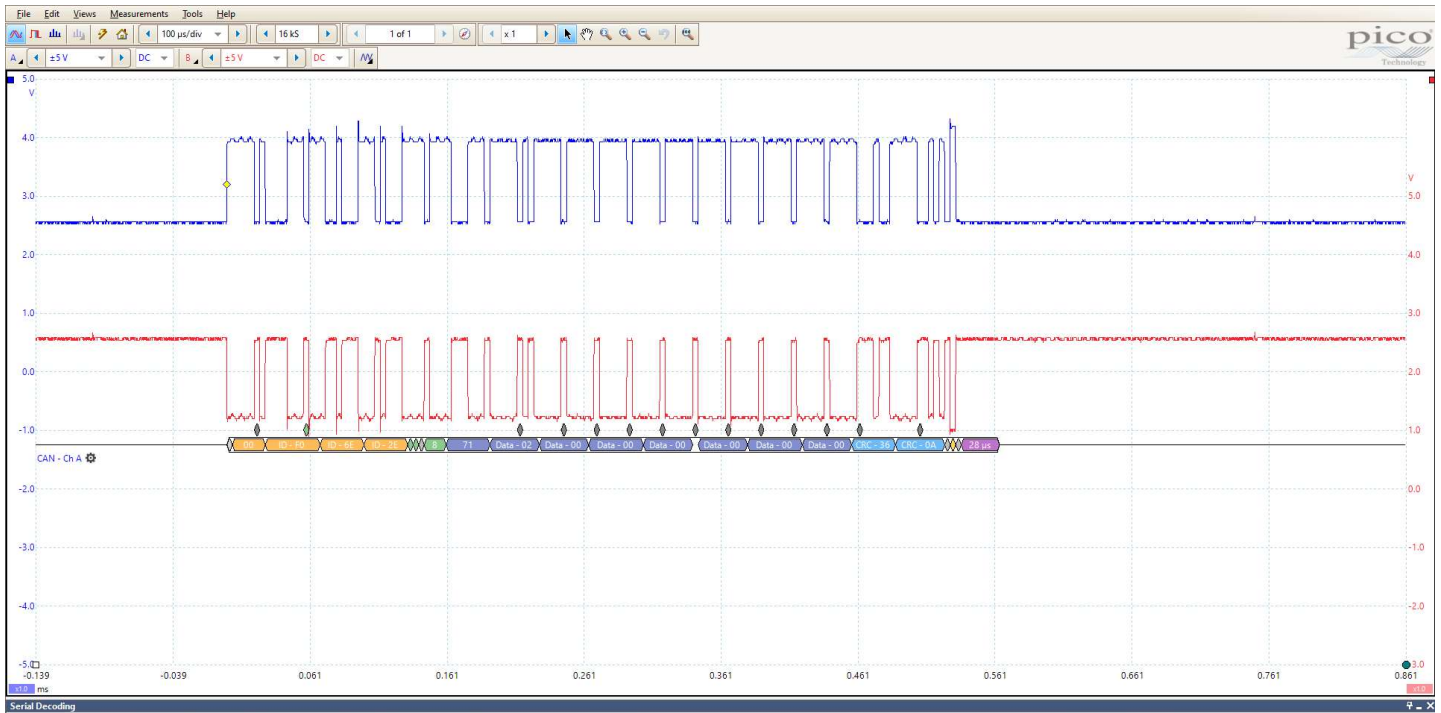
It allows you to create simple programs with a user interface.

It includes sixteen switches with LEDs, an LCD display, 7-segment displays and simple sensors.

3. CAN bus board.

This plugs into the serial port of the PICmicro device and allows you to create fully industrial-compatible CAN messages that interface to the system.

4. Backplane



The core part of a CAN bus message consists of a two byte identifier and eight bytes of data.

This is often described as: IA IB D0 D1 D2 D3 D4 D5 D6 D7

The trace above shows a CAN message on a Picoscope.

CAN High is shown in blue and CAN Low is shown in red.

Note that 0V on the Y-axis is different for each trace.

The decoded message is shown on the trace. The message here is:

```
IA IB D0 D1 D2 D3 D4 D5 D6 D7
F0 03 00 56 00 00 00 00 00 00
```

IA and IB are the first two parts of the identifier - often referred to as simply the 'ID', here eleven bits of data. In this case it is (hexadecimal) F003.

The CAN bus signal is transmitted on two wires, known as CAN High and CAN Low. In addition, two more cables provide the power supply. The CAN High signal has a voltage range between 2.5V and 4V (with respect to ground). The CAN Low signal is transmitted between 1V and 2.5V. The signals are 'differential', meaning that a logic signal 1 creates a high CAN High voltage at the same time as a low CAN Low voltage. A logic 0 signal is transmitted as both CAN High and CAN Low sitting at roughly the same voltage (around 2.5V). At the receiver the ECU uses the voltage difference between CAN High and CAN Low to decode it.

The codes used in this module are identified on the next page.

The image shows other parts of the message after the transmitted data (D0 to D7). These include start bits and CRC error checking bits, (not covered by this course.)

ID	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Message
11-bits (Hex)	Where quantities take up two bytes, the first is the LSB and the second is the MSB E.g. 0x1234 (4660 in decimal) would be sent as: 0x34 0x12								Frequency (ms)
Messages from EFIS									
0x0012	AHRS Bank angle x 0.1 degrees		AHRS Pitch angle x 0.1 degrees		AHRS YAW angle x 0.1 degrees		Ground Speed mph		500
Messages from RDAC (Simulated by MIAC)									
0x0201	Fuel Flow 1 ⁽¹⁾		Fuel Flow Ratio 1		Fuel Flow 2 ⁽¹⁾		Fuel Flow Ratio 2		4000
Flows are pulses / 400ms. If ratio = 1000 then each pulse is 0.1285 litres / hour									
0x0202	Exhaust Gas 1 ⁽²⁾ degrees Celsius ⁽³⁾		Exhaust Gas 2 degrees Celsius		Cylinder Head 1 degrees Celsius		Cylinder Head 2 degrees Celsius		500
0x0203	Thermocouple 5 degrees Celsius		Thermocouple 6 degrees Celsius		Thermocouple 7 degrees Celsius		Thermocouple 8 degrees Celsius		500
0x0204	Thermocouple 9 degrees Celsius		Thermocouple 10 degrees Celsius		Thermocouple 11 degrees Celsius		Thermocouple 12 degrees Celsius		500
0x0205	Oil Temperature (8)		Oil Pressure (9)		Aux 1 (8)		Aux 2 (8)		500
0x0206	Fuel Pressure (10)		Coolant Temp (8)		Fuel Level 1 ⁽⁴⁾ (5)		Fuel Level 2		500
0x0207	Cold Junction degrees Celsius ⁽³⁾		Supply Voltage x volts / 5.73758		Not Used				500
0x0208	Engine Speed 1 ⁽⁶⁾ x 6 RPM		Engine Speed 2 x 6 RPM		MAP ⁽⁷⁾ x 0.02 inches Hg		Current (Amps x 32) + 128		200
Messages from SP-6 Compass									
0x0241	Heading x 0.01 degrees		Magnetometer Data x,y,z - 12 bits each				Slip		50
0x07D2	Start up message - not relevant								50
Messages from SP-7 Attitude Heading Reference System									
0x0281	Accelerometer X x 0.001 G		Accelerometer Y x 0.001 G		Accelerometer Z x 0.001 G		Total G x 0.001 G		50
0x0282	Gyro Turn Rate x 0.1 deg / min		Gyro Bank Rate x 0.022 deg / sec		Gyro Pitch Rate x 0.022 deg / sec		Gyro Yaw Rate x 0.022 deg / sec		50
0x0283	Euler Roll x 0.01 degrees		Euler Pitch x 0.01 degrees		Euler Yaw x 0.01 degrees		Slip Status Word		50
0x0284	Not Used		Interface Version		Internal Temp		Voltage		50

Notes: -

- (1) MIAC Input 5 simulates fuel flow sensor.
- (2) MIAC Input 4 simulates exhaust gas temperature sensor.
- (3) Thermocouple readings are uncompensated. Display adds cold junction temperature from message 0x207.
- (4) MIAC Input 2 simulates fuel level sensor.
- (5) Fuel level scaling is piecewise linear. Calibration can be adjusted in the EFIS menu.
- (6) MIAC Input 1 simulates engine speed sensor.
- (7) MIAC Input 3 simulates manifold air pressure sensor.
- (8) Temperature Displays = $(21215 / (CV + 133.6)) + 27$ (CV = CAN Value)
- (9) Oil Pressure Display = $(CV^2 / 83333) + (CV / 110) - 0.16$
- (10) Fuel Pressure Display = $(CV / 512) - 0.3$

Scaling information for each quantity is given in the relevant spreadsheet cell.

Notes for the Instructor

This workbook is intended to reinforce the learning that takes place in EASA module 5 - Digital Techniques and Electronics Instrument Systems.

Coverage of module 5 is split across several Matrix products as follows:

	EASA Electronic fundamentals (LK9282)	Electronic Flight information systems (AV3737)	Microcontroller systems in Aviation (BL2976)
5.1 Electronic instrument systems			
5.2 Numbering systems		✓	✓
5.3 Data conversion			✓
5.4 Data buses		✓	✓
5.5 Logic circuits	✓		
5.6 Basic computer structure			✓
5.7 Microprocessors			✓
5.8 Integrated circuits			✓
5.9 Multiplexing		✓	✓
5.10 Fibre optics	✓		
5.11 Electronics displays		✓	✓
5.12 Electrostatic sensitive devices			
5.13 Software management and control		✓	
5.14 Electromagnetic environment			
5.15 Typical electronic / digital aircraft systems		✓	

Using this package, the expected learning outcomes are:

- hexadecimal, decimal and binary number systems;
- message multiplexing;
- interconnection of electronics modules in aircraft systems;
- sensors in aircraft;
- CAN bus systems in aircraft;
- CAN bus fault-finding with a multimeter;
- CAN bus fault-finding with an oscilloscope;
- built-in test, monitoring and fault-finding
- fly-by-wire systems;
- CAN bus message structure and protocol;
- Electronic Flight Information Systems;
- PC-based diagnosis software tools;
- Use of GPS.

Prior Knowledge

Students should have previously studied EASA Electronic Fundamentals and Microcontroller systems in Aviation.

Worksheet	Notes for the Tutor	Timing
1	<p>Following the instructions in this worksheet, the students build the system.</p> <p>The wiring is quite clear in the wiring diagram and students should be encouraged to view it as power, (V+ and 0V), CAN High, and CAN Low, fed to every node in parallel.</p> <p>They should then spend time with the EFIS module getting to know its structure. For most of that time, a simple trial-and-error approach is enough to learn the layout of the EFIS module but a full set of operator manuals for the EFIS module should be made available to them (from the MGL Avionics web site). The instructions include testing the pitot and GPS sensors. The instructor can adapt these to suit local conditions.</p> <p>Students should spend time reading the documents on the various parts of the system in the Reference section.</p> <p>They will also need introductory talks on CAN bus and the concepts of digital communications.</p>	60 minutes
2	<p>Students should add the PC-based diagnostic system and make sure that all the parts of the system are up and running.</p> <p>Students should make sure that all the potentiometers produce the required results on the EFIS and the Avionics Analyser screen.</p>	20 minutes
3	<p>The Avionics Analyser software is the main tool used for learning about the CAN bus. It is easy to use and no hardware installation is necessary.</p> <p>(Simply using an oscilloscope will not be sufficient.)</p> <p>Students should explore the Avionics Analyser software to understand how CAN messages are constructed and sent around the system.</p> <p>Once again, students should make use of the documents in the Reference section.</p>	40 minutes
4	<p>In this worksheet, students study the structure of the CAN messages and how sensors and ECUs make use of them to embed information.</p> <p>They need to understand that these messages are sent at regular intervals and that they can be displayed in various ways as required by the diagnostic tool provider.</p>	40 minutes

Worksheet	Notes for the Tutor	Timing
5	<p>In this worksheet, the students learn how the EFIS takes a message and manipulates it before it is displayed, and how different parts of the CAN bus packet are used for different quantities. They then construct their own CAN messages to alter the behaviour of the EFIS module.</p> <p>The students also learn about scaling, that some mathematical manipulation is involved when constructing these messages and that the numbers transmitted are not necessarily the actual quantities to be displayed.</p>	60 minutes
6	<p>Students learn that overall system design can help in fault-diagnosis and that ECUs can also provide information about likely faults in the circuitry connected to it.</p>	40 minutes
7	<p>Students learn that some ECU faults can be diagnosed by a master node and the relevant information passed to the technician to help in fault-finding.</p>	30 minutes
8,9	<p>In these sections, students learn what to look for when debugging a CAN bus with a multimeter or with an oscilloscope.</p> <p>As only a limited number of faults are possible in a CAN bus system the task is easy. They introduce a known fault and make measurements of the result.</p> <p>They then collate the findings so that they can investigate and identify a fault from its symptoms, using an oscilloscope or multimeter.</p>	70 minutes
10	<p>To understand the architecture and programming of CAN bus systems at a fundamental level, students add an E-blocks system with a CAN bus board and create an appropriate program to turns a potentiometer into a compass.</p>	40 minutes

There are two parts to this course:

1. CP5715 - EFIS for Aviation
2. CP7244 - Digital techniques in Aviation

CP5715 - EFIS for Aviation requires the following parts:

1	AV2209	EFIS display panel - with GPS antenna
1	AV5322	EFIS sensor panel
1	COM00170	Pipette filler
1	COM4177	4mm diameter tubing, 300mm
1	HP2666	Power supply
2	HP4039	Lid for plastic trays
2	HP5540	Deep plastic tray
1	HP7750	Locktronics daughter tray foam insert
1	HP9564	62mm daughter tray
1	HPUSB	USB lead
1	LK5202	Resistor - 1K, 1/4W, 5% (DIN)
2	LK5206	Resistor - 120 ohm, 1W 5% (DIN)
1	LK5209	Resistor - 5K6, 1/4W, 5% (DIN)
5	LK5214	Potentiometer, 10K (DIN)
28	LK5250	Connecting Link
2	LK5603	Lead - red - 500mm, 4mm to 4mm stackable
2	LK5604	Lead - black - 500mm, 4mm to 4mm stackable
2	LK5660	Lead - black - 1000mm, 4mm to 4mm stackable
2	Unknown	Lead - red - 1000mm, 4mm to 4mm stackable
8	LK5607	Lead - yellow - 500mm, 4mm to 4mm stackable
8	LK5609	Lead - blue - 500mm, 4mm to 4mm stackable
2	LK8900	7 x 5 baseboard with 4mm pillars
1	MI0550	MIAC NXT
1	HP2666	Adjustable power supply
2	LK5620	Lead, yellow, 1000mm 4mm to 4mm stackable
2	LK5640	Lead, blue, 1000mm, 4mm to 4mm stackable

CP7244 - Digital techniques in Aviation requires the following parts:

1	BL0127	E-blocks 2 actuators board
1	BL0140	E-blocks 2 CAN bus board
1	BL0161	Patch board
1	BL0562	PIC development centre and printed panel
1	FCXXX	Flowcode for education
1	HP2666	Adjustable power supply
2	HP4039	Tray Lid
1	HP5540	Deep tray
1	HP9564	62mm daughter tray
1	HPUSB	USB lead

Version control**Electronic Flight Information Systems**

14 07 22 First beta
13 12 22 First release