MIAC-01

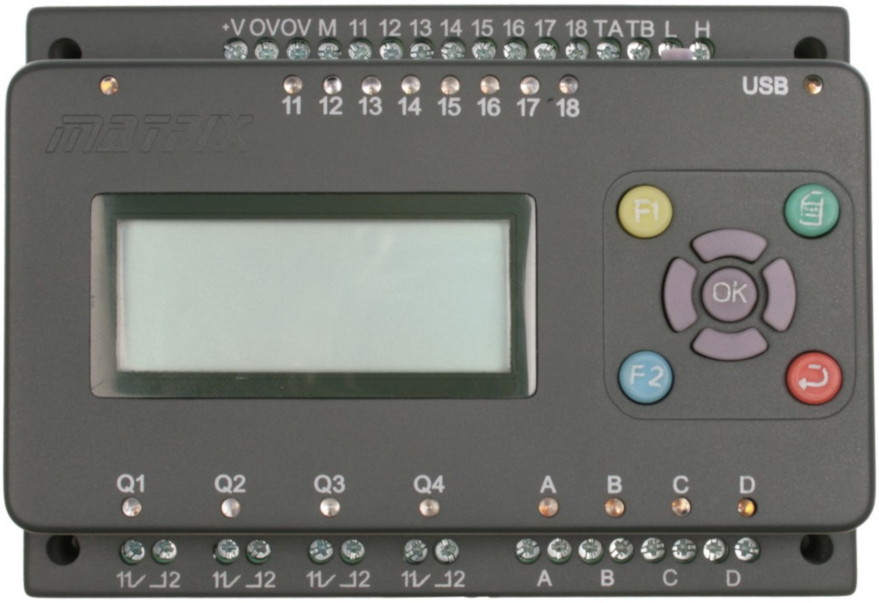
Now you are in control

Getting started guide

English
Spanish
Greek
Thai
Dutch
French
Romanian
Finish
Italian
Vietnamese
German
Slovenian
Danish
Hungarian
Korean
Slovak
Mandarin
Turkish
Portuguese
Japanese

MATRIX

Introduction



What does it do?

MIAC (Matrix Industrial Automotive Controller) is an industrial grade control unit which can be used to control a wide range of different electronic systems. It has a number of applications in industry and learning.

Benefits

- Flexible and expandable
- Easy to program with flowcharts, C or Assembly code
- Physically and electrically rugged

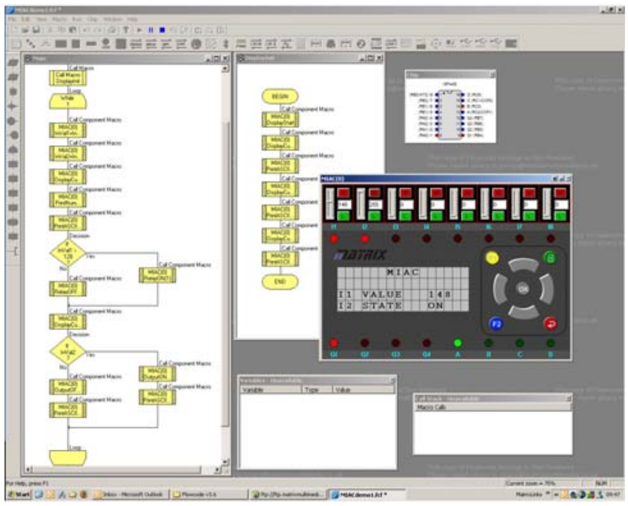
Features

- Programmable from USB
- Compatible with Flowcode and C
- 8 digital or analogue inputs
- 4 relay outputs, 4 motor outputs with speed control
- 4 line LCD display and control keys
- Lab View and Visual Basic compatible

Description

The MIAC is a fully specified industrial electronic controller designed to operate off 12 or 24V. It has 8 analogue or digital inputs, 4 high current relay outputs and 4 motor outputs. The MIAC is housed in an attractive, rugged, anthracite grey plastic moulding. It has two physical mounting options: it can be mounted onto a 30mm 'top hat' DIN rail, or it can be mounted directly onto any surface using the 4 screw holes provided.

The MIAC unit has screw terminal connector inputs across the top and bottom of the unit, has several input buttons for user control, and also has a 4 line 16 character alphanumeric display on the top of the unit to display system status and assist users.



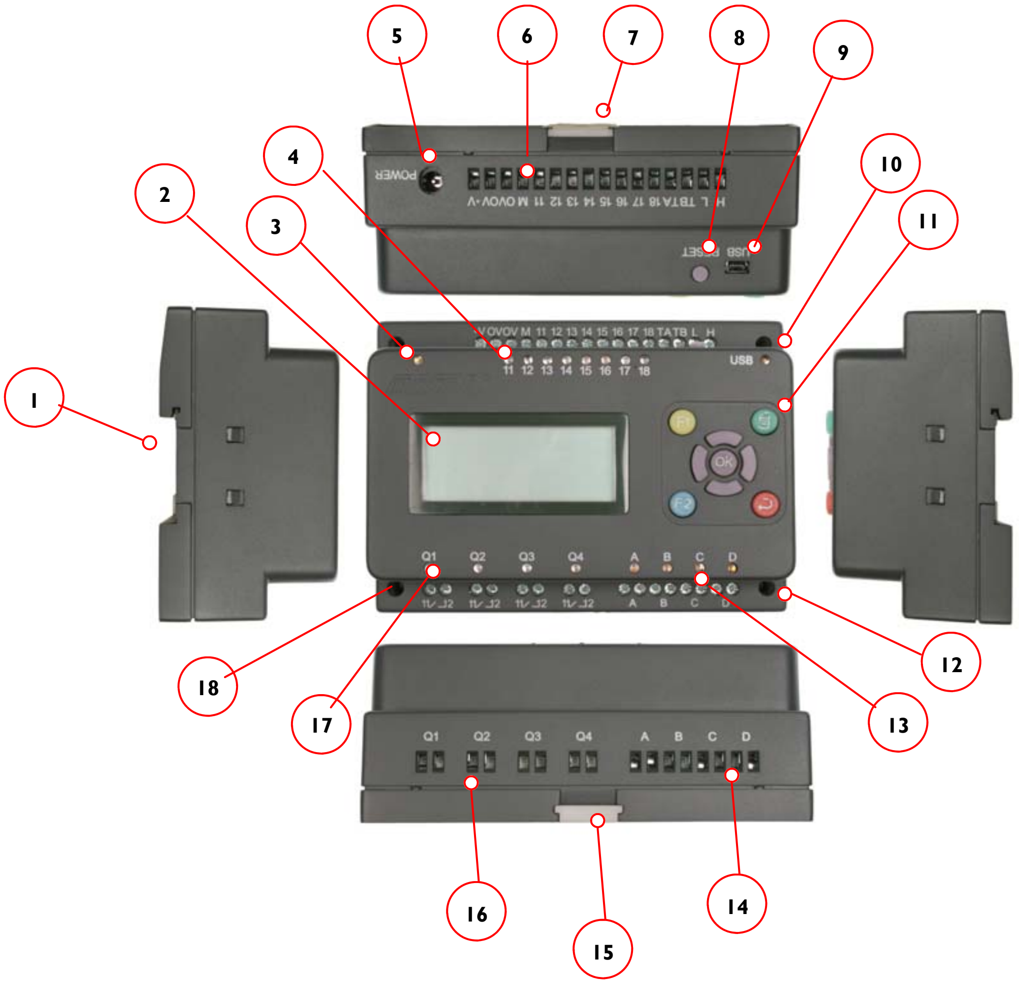
Flowcode- the graphical programming language supplied with MIAC

The unit is programmed directly from a PC's USB port and is compatible with the Flowcode graphical programming language. Users can develop a program using Flowcode, press the Reset button on the back of the unit, and the program will automatically download and start. The MIAC can also be programmed in C and assembly code, or any program that is compatible with PICmicro microcontrollers.

MIAC is equipped with a fully operational CAN bus interface so that many MIACs can be networked together to form wide area electronic systems.

A DLL and sample programs are provided to enable MIAC to be used with PC based control programs like LabView, Visual Basic, C++ etc.

MIAC hardware



Key

1. Top hat rail mounting recess
2. 16 character x 4 line LCD display
3. Power LED
4. Input status LEDs
5. 2.1mm power jack
6. Screw terminal inputs
7. Top hat rail retainer clip - upper
8. Reset / run switch
9. USB socket
10. USB transfer LED
11. Control keys
12. M3 mounting holes
13. Motor status LEDs
14. Motor output screw terminals
15. Top hat rail retainer clip - lower
16. Relay output screw terminals
17. Relay output status LEDs
18. M2 mounting holes

MIAC is housed in a custom-made plastic moulded housing which can be mounted using standard M2.5 bolts, or can be mounted on a 'top hat' DIN rail. If using a DIN rail two retainer clips are used to lock the unit in place on the rail.

The unit has 8 analogue or digital inputs, 4 relay outputs, 4 transistor outputs, and a CAN bus interface. The unit is powered by an 18F4455 18 series PICmicro microcontroller from Arizona Microchip.

Inputs are fed into a signal conditioning circuit which allows them to be used as both analogue and digital inputs. Software dictates their operation as an analogue input or as a digital input. Inputs are not optically isolated. Signal conditioning powers the topside LEDs which show analogue inputs at the appropriate brightness relative to the voltage input. The input range is 0 to 10V which makes the MIAC compatible with industry standard sensors.

4 outputs from the PIC processor are fed into a power stage which provides current amplification before feeding them to 4 separate relays. Fusing for relay circuits should be provided externally.

4 additional outputs are fed into a motor driver stage, including current monitor with shutdown circuitry which limits the output current and protects the motor driver chip in case of short circuits. Output status is reflected by topside LEDs - where Pulse Width Modulation is used (only available on two of the 4 outputs) LED brightness reflects motor speed.

The internal processor also connects to a CAN bus driver circuit which allows many MIACs to be connected together to form an industrial control network.

Great care has been taken to ensure that the MIAC is electrically rugged - it is possible to short out any input to any output or any one output to any other without the unit failing. Relay contacts are not current limited and external fuses should be used to limit relay current to 8A.

Control and monitoring of processes is facilitated by a 4 line LCD display and a customisable keypad.

The MIAC is supplied with a version of Flowcode 3 professional (limited chip set) which allows users to program the device using flow charts.

Software for downloading hex programs designed for the PIC18F4455 in C/assembler etc., is supplied with the unit. Any programs should be designed to start at 0600 to allow for the bootloader supplied.

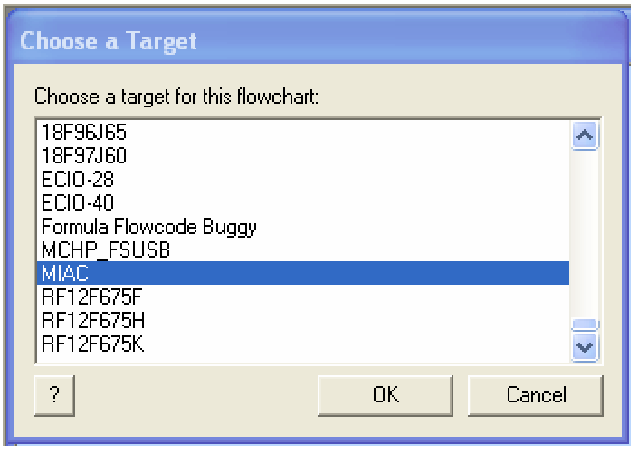
Using MIAC with Flowcode

Internally the MIAC is powered by a powerful 18 series PICmicro device which connects directly to the USB port for fast programming. The PIC device is pre-programmed with a bootloader and a Windows utility is provided with MIAC that allows programmers to download PIC compatible hex code into the device. This makes MIAC compatible with a wide range of compilers which generate Hex code for the 18F4455. Instructions on downloading hex files are given below.

To take advantage of the MIAC macros and simulation features Flowcode 3 users will need to upgrade to version 3.6 of Flowcode or to install the version of Flowcode 3.6 that is shipped with the MAIC unit.

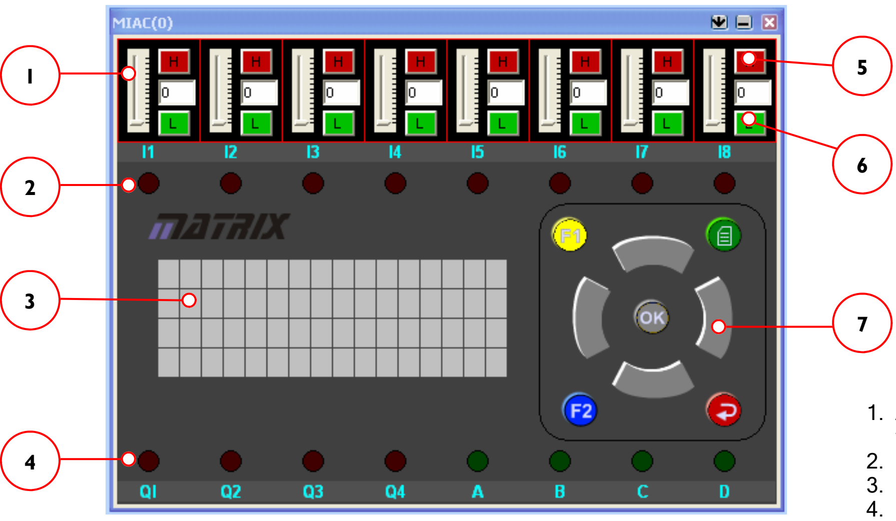
Getting started using Flowcode

Open a new flowchart in Flowcode. Select MIAC as the target device:

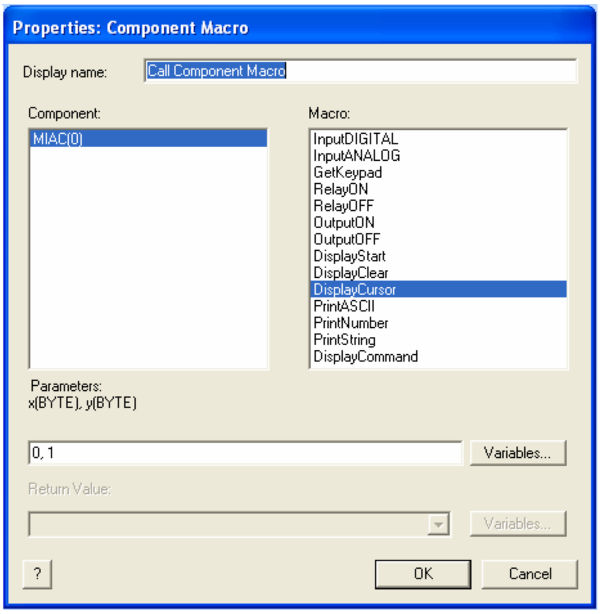


Selecting the MIAC as a target in Flowcode

Click on the MIAC icon on the tool bar to load the MIAC component onto the flowchart which will reveal the MIAC component shown below. Within Flowcode it is possible to call macros for all the basic input and output functions, to read the keypad, and to control the LCD display and to see the results on the simulation component.



The MIAC simulation component within Flowcode



Calling MIAC functions with Flowcode

Flowcode MIAC macros

InputDIGITAL
Return the logic level of the specified input.

InputANALOG
Return the analog voltage value from the specified input (20 x voltage)

GetKeypad
Scan the keypad and return the value of any key being pressed. Return 255 if no key is being pressed.

DisplayStart
Initialise the LCD (must be run before any other display command is used)

DisplayClear
Clear the LCD and return the cursor to the home position

DisplayCursor
Move the invisible LCD display cursor to the specified location

PrintASCII
Print the specified ASCII characters to the LCD, starting at the current cursor location

PrintNumber
Print a left-justified byte or integer value to the LCD, starting at the current cursor location

PrintString
Print a string to the LCD, starting at the current cursor location

DisplayCommand
Send a command/control character directly to the LCD

RelayON
Turn the specified relay on

RelayOFF
Turn the specified relay off

OutputON
Turn the specified transistor output on

OutputOFF
Turn the specified transistor output off

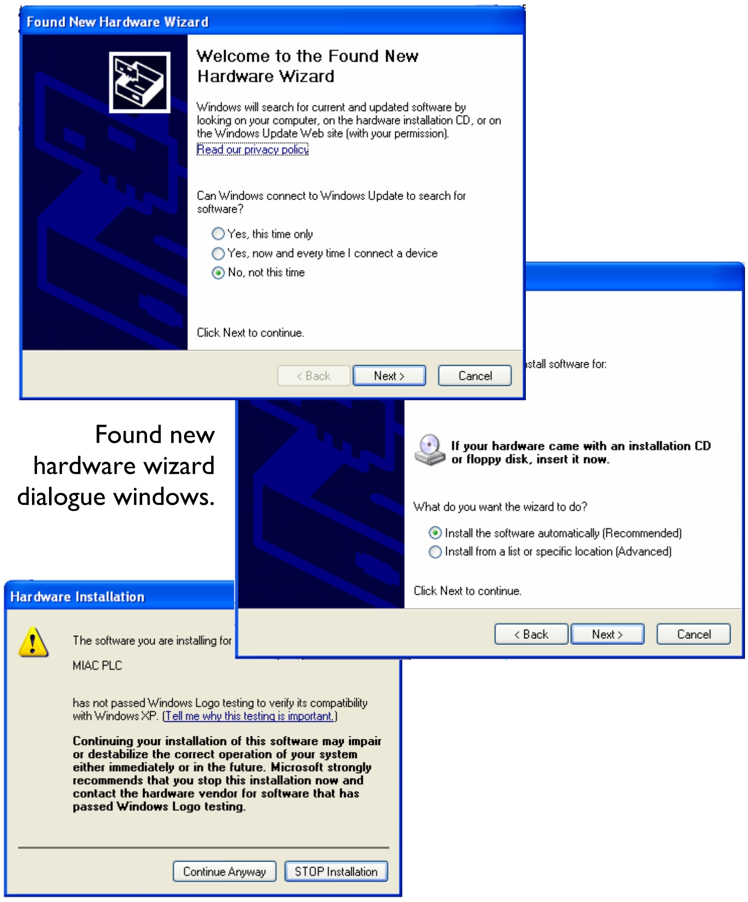
1. Analog slider controls and conversion value displays
2. Input status indicators
3. 16 x 4 character LCD
4. Output status indicators
5. Digital high buttons
6. Digital low buttons
7. Keypad

Installing drivers and software

Drivers

Before using the MIAC, driver software must be installed. To do this:

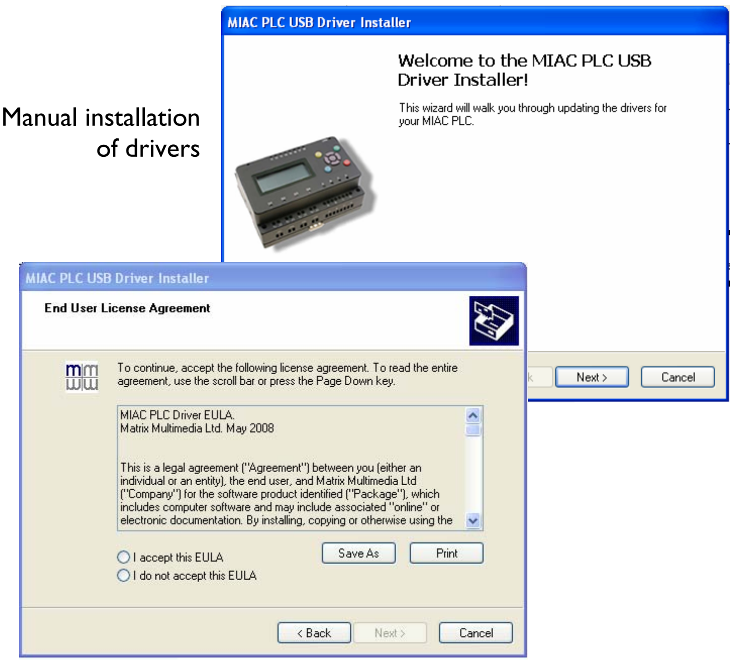
1. Power the MIAC using a suitable power source (see specifications). Note: The MIAC can not be powered via the USB connector
2. Connect a MIAC to the PC using a USB cable.
3. Press the RESET button on the MIAC to start the bootloader software and establish USB communications. If the MIAC driver is not installed, Windows will detect the MIAC as new hardware and begin the driver installation process.
4. Make sure the installation CD is in the drive. When the 'found new hardware' dialogue window appears select the options to disable internet searching and enable automatic installation of the software. If a compatibility warning appears, select the 'Continue Anyway' option. The driver is now installed.



Found new hardware wizard dialogue windows.

Manual installation of drivers

Alternatively, the driver can be manually pre-installed by selecting the appropriate installation .exe file on the CD. The files are located at <CD drive>:\etc\drivers\MIAC\. Separate files have been provided for 32-bit, 64-bit Intel, and 64-bit AMD processors:
Matrix_MIAC_Driver_install.exe
Matrix_MIAC_Driver_install (ia64).exe
Matrix_MIAC_Driver_install (amd64).exe



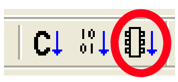
Manual installation of drivers

Programming MIAC

Each MIAC is shipped containing a small bootloader program that manages the transfer of programs into the internal flash memory via the USB port. The MIAC enters programming mode when a USB connection is detected during recovery from a RESET condition.
The two main causes of a RESET condition are connecting the power supply, and pressing the RESET button. If no USB connection is detected, MIAC will immediately run the program stored in its memory.
When in programming mode MIAC will wait for several seconds, with the USB LED flashing, trying to establish communications with a programming utility running on the host PC. If communications are not established, MIAC will run the program stored in its memory.

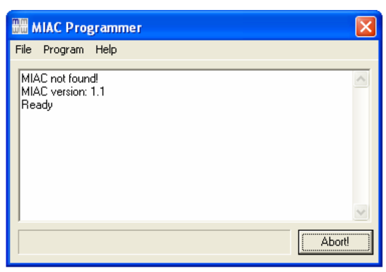
Programming from Flowcode

The MIAC programming utility (MIACprog.exe) is installed as part of Flowcode and will be automatically launched by Flowcode during the 'Compile to chip' operation. If the MIAC is not in programming mode, a message will be displayed with the necessary instructions.



Using MIACprog directly

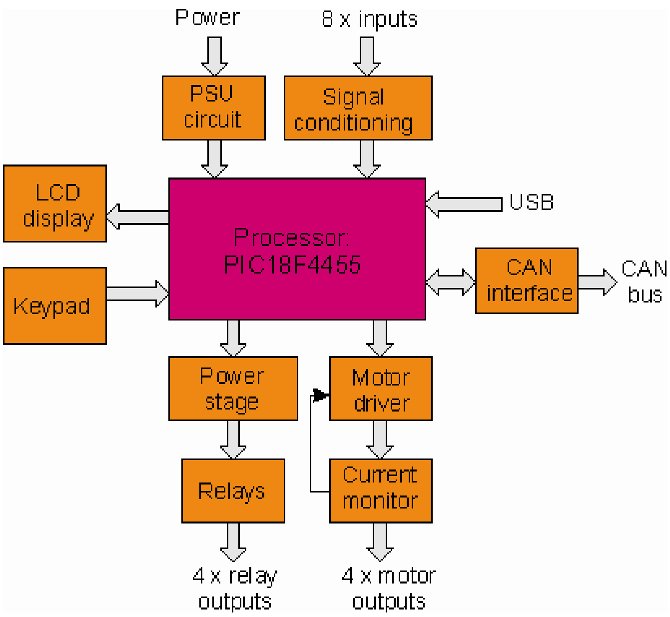
When not using Flowcode to download programs to MIAC, MIACprog can be launched manually.
If not installed as part of Flowcode, the MIACprog files can be found on the installation CD at:
<CD drive>:\etc\software\MIAC



MIACprog

Internal block schematic

Internally the MIAC is powered by a powerful 18 series PICmicro device which connects directly to the USB port for fast programming. The PIC device is pre-programmed with a bootloader and a Windows utility is provided which allows programmers to download PIC compatible hex code into the device.
Inputs are fed into a signal conditioning circuit which allows them to be used as both analogue and digital inputs. (Not optically isolated.) Signal conditioning powers the topside LEDs which show analogue inputs at the appropriate brightness.
The outputs from the PIC processor are fed into a power stage which provides current amplification before feeding them to 4 separate relays. Fusing for relay circuits should be provided externally. Additional outputs are fed into a motor driver stage and current monitor with shutdown circuitry which limits the output current and protects the motor driver chip in case of short circuits. Output status is reflected by topside LEDs - for motor outputs LED brightness reflects motor speed / PWM ratio.
The PIC processor also connects to a two wire CAN bus driver circuit which allows several MIACs to be connected together to form an industrial control network.
Control and monitoring of processes is facilitated by a 4 line LCD display and a customisable keypad.



Internal block schematic

Power supply circuit

Please refer to the schematic below.
MIAC can be powered with a DC supply voltage in the range 12V to 26V. The power can be supplied via the 2.1mm power jack (POWER), or the power supply terminals (V+, 0V). The power jack is fed into a bridge circuit and can therefore accept plugs wired with either connection polarity.

To prevent interference from high current spikes on the voltage rails the supply voltage is filtered inside the MIAC.

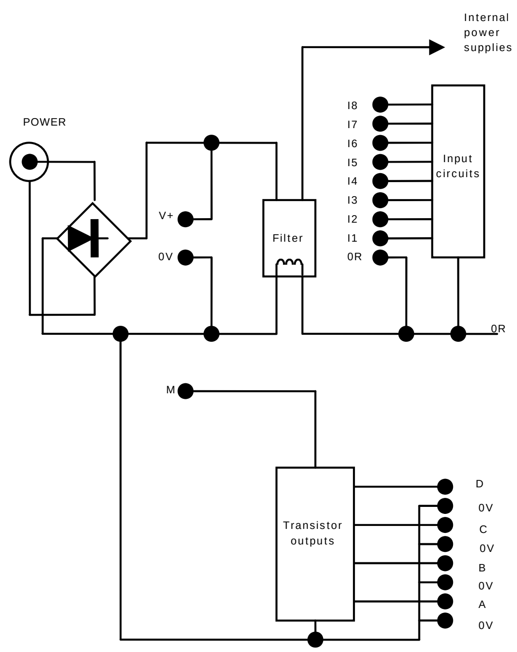
The 0R terminal is the common voltage for the internal logic circuits and is connected to the 0V terminals via the filter. This reduces the effects of the high current circuits on the control system, and provides an accurate reference voltage for input circuits, especially when being used as analogue inputs.

Notre that in the first 50 MIAC off the production line the filter can cause problems when using the PWM outputs as the inductive nature of the filter causes 0R to rise slightly. In this case use an external link to short 0R to)V externally and bypass the filter.

The transistor outputs are not powered internally. The M terminal is used to apply power to the transistors which allows a voltage other than the supply voltage to be used on the transistor outputs. If you wish to power the transistors from the same supply as the supply voltage then simply use a shorting link between the V+ and the M terminals.

The maximum value of M is nominally 12V but up to 36V can be used depending on the ambient temperature.

The transistor outputs are supplied by a single L298 device and can supply up to 2A each at 24V. Transistor outputs can be connected in parallel if more power is needed from out output.



Power supply schematic

Input circuits

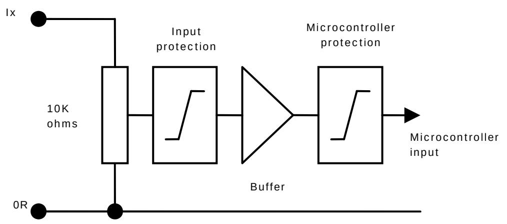
MIAC has eight identical inputs (I1 to I8). Each one can be read as an analogue or digital value. Input voltages from 0V to 12V can be read as either analogue or digital values. Input voltages from 12V to up to 24V are clamped internally, resulting in an unrepresentative analogue conversion, but produce a logic level 1 when read digitally.

Digital readings

Logic level switching occurs at input voltages between 3V and 8V. The logic levels are undefined for input voltages between these levels.

Digital inputs rely on the signal source switching cleanly between the two logic levels:

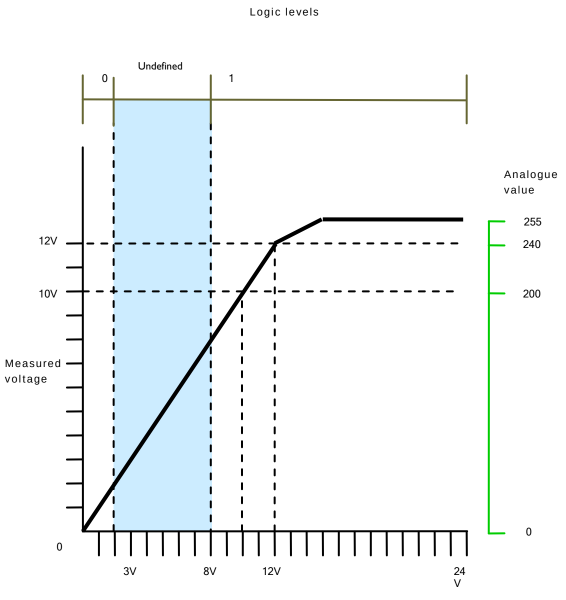
Logic 0: < 3V
Logic 1: > 8V



Schematic of each input

Analogue conversions

Analogue conversions have a resolution of 50mV (20 counts per Volt). This allows simple calculations to be used to convert the results of an analogue sample into a direct voltage representation - simply divide the input by 20.



Analog input conversion

Reading a digital input with Flowcode

In addition to the input/output commands supplied by the MIAC component, it is also possible to read digital inputs, and control any of the outputs, using the Flowcode Input and Output blocks.

Reading an input using other programming languages

Due to the internal allocation of microcontroller I/O for specific system functions, the external I/O is made up of individual pins from several ports:

INPUTS	
MIAC INPUT	MICROCONTROLLER PIN
I1	RA0
I2	RA1
I3	RA2
I4	RA4
I5	RE0
I6	RE1
I7	RE2
I8	RB2

Output circuits

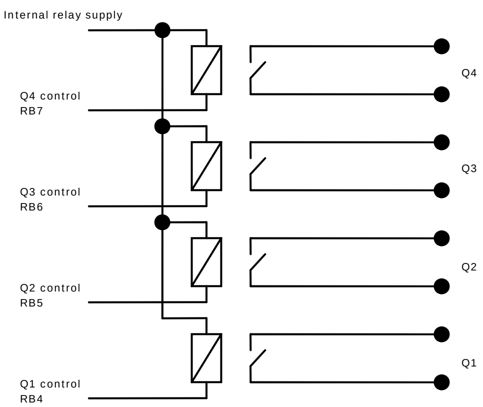
Relay outputs

MIAC contains four, single-pole, normally open relays. Q1 to Q4. Pairs of terminals provide access to the switch contacts of each relay.

The four pairs of relay contacts are isolated from each other, and from the MIAC control circuitry.

The relays are independently controlled by the MIAC. Each can switch up to 8A at 250VAC or 30VDC.

In order to retain the high isolation and low resistance switching properties of the relays, no protection devices have been added to the contact circuits. Care should be taken when switching loads that could exceed the voltage or current ratings of the contacts.



Relay outputs

Transistor outputs

MIAC provides four transistor controlled outputs. The four outputs have a common positive supply voltage (derived from the M terminal) and local 0V terminals.

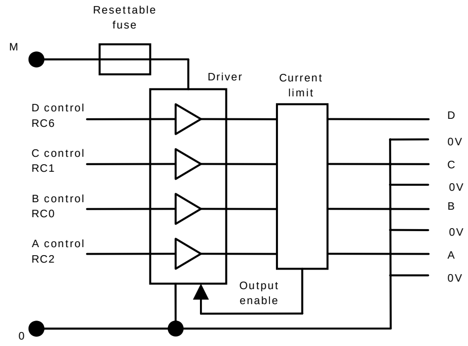
The transistor outputs include several protection devices to prevent accidental damage.

- Each output is current limited to approximately 1.75A sourcing or sinking.
- The supply from the M terminal is protected by a 4A resettable fuse.

The device driving the outputs contains a thermal protection circuit. Activation of any of the protection devices may cause all four transistor outputs to shut down for a short time period - depending on the nature of the fault

The transistor outputs are able to switch much faster than the relay output, allowing high frequency control signals to be generated. Two of the outputs (A and C) can be driven in hardware PWM mode, allowing high efficiency control of motor speeds, lamp brightness etc.

Each output can drive loads referenced to 0V, or to the positive supply voltage, or to another output (bridged connection). Bridged connection allows isolated loads, like many d.c. motors, to be driven bi-directionally.



Transistor outputs

Altering output status with Flowcode

The MIAC component within Flowcode has macros which allow users to directly adjust the status of the relay and transistor outputs. These can also be altered using the standard Output icon within Flowcode using the information in the tables below.

Altering output status with other programming languages

Due to the internal allocation of microcontroller I/O for specific system functions, the external I/O is made up of individual pins from several ports as can be seen in the tables below:

RELAY OUTPUTS	
MIAC OUTPUT	MICROCONTROLLER PIN
Q1	RB4
Q2	RB5
Q3	RB6
Q4	RB7

TRANSISTOR OUTPUTS	
MIAC OUTPUT	MICROCONTROLLER PIN
A	RC2
B	RC0
C	RC1
D	RC6

Using Pulse Width Modulation (PWM)

Two of the transistor outputs (A and C) can be configured to operate under the control of hardware PWM signal generators.

PWM functionality for the A and/or C output(s) can be included in a Flowcode program by adding the standard Flowcode PWM component:. Please refer to the Flowcode PWM documentation and help files for further information on the configuration and use of this component.

Controlling PWM using other compilers can be achieved by writing to the relevant PIC registers - see the datasheet on the 18F4455 for details.

PWM channel 1 = Transistor output A
PWM channel 2 = Transistor output C

Operator interface

The operator interface consists of:

- A status LED for each input.
- A status LED for each output.
- An indicator LED for the power supply.
- An indicator LED for the USB port.
- A 16 x 4 character LCD
- A 9-key keypad.

Keypad

The keypad consists of 9 keys. None of the keys have any specific software functions assigned to them, but are physically arranged to represent a range of frequently used operations:

- 4 cursor control/navigation buttons arranged in a circle.
- An OK (enter/select) button in the middle of the cursor keys.
- 4 function keys, individually coloured for ease of identification and referencing from the LCD.
- 2 of the function keys have been allocated useful icons (MENU and Go Back/Undo)
- 2 of the function keys are coloured red and green to represent STOP & START or ON & OFF controls. These are assigned the icons 'F1' and 'F2' implying Function 1 and function 2.

Using the keypad in Flowcode

Key	Icon	Value
Cursor centre	OK	4
Cursor up		5
Cursor down		3
Cursor left		1
Cursor right		7
Green	Menu	8
Red	Undo	6
Yellow	F1	2
Blue	F2	0

Key values returned by GetKeypad function

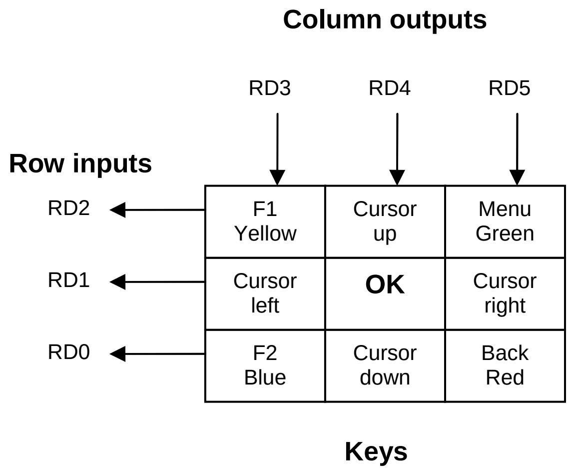
Within Flowcode the MIAC component includes macros for returning the number of the key pressed as can be seen in the table. The user program can freely define the functions allocated to each key, and the way the keypad interacts with the LCD.

Using the keypad with programs other than Flowcode

The keypad configuration is a standard 'output rows, scan columns' device. The keypad is connected as a 3 x 3 matrix. The keypad applies a weak pull-up to the common connections, so the keypad has been configured for active low column selection. A low level on any of the row connections indicates a corresponding key press in the selected column.

LED indicators

The input and output LEDs indicate the logic level present on their associated terminals. An input LEDs state is not valid when its input is being read as an analogue value. A LED could be weakly lit and yet not at a 'high' level. The need for the LED to indicate

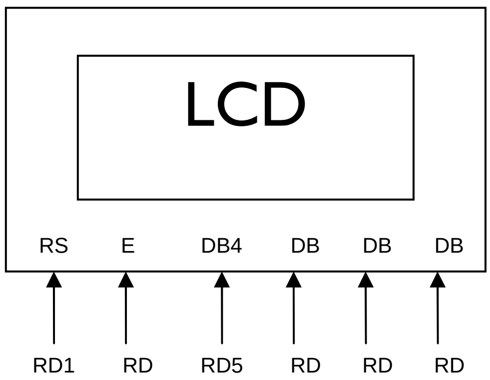


Keypad block diagram

analogue voltage dictates this.

The POWER LED is driven by the internal logic supply and indicates that both the power and logic supplies are present. The USB LED is illuminated when a USB connection is present, and will flash to indicate activity on the USB port.

LCD



LCD connections

The LCD is a general purpose alpha-numeric display consisting of 4 lines of 16 characters. It is fully accessible by programs running on the MIAC, and is supported in Flowcode by functions supplied as part of the MIAC component. The LCD E (Enable) signal is driven separately and allows the LCD to respond only to data intended for it. The display must be set to 4-bit data mode. The LCD R/W signal is connected to 0V (write mode) so data can not be retrieved from the display. The display is an SC1604A which uses a Samsung KS0066 controller and is compatible with the industry standard Hitachi HD44780 interface.

CAN bus interface

CAN (Control Area Network) is a standard serial communications bus used widely in both automotive and industrial applications. This bus can be used to network multiple MIAC units, and to communicate with other CAN equipped devices.

Each MIAC contains all the hardware necessary to operate as a CAN node. Successful transmission at high speed, and over long distances, requires good wiring practices to be observed:

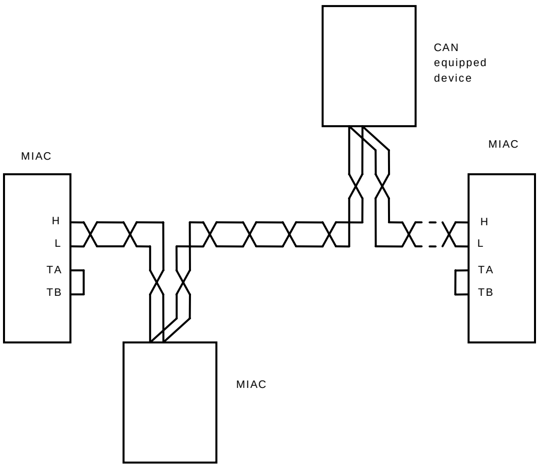
- Suitable twisted pair cable should be used.
- Branches to individual nodes must be kept as short as possible so the CAN bus has definite start and end nodes.
- Terminating resistors should be fitted at the start and end of the bus.

Each MIAC contains an internal terminating resistor that can be included in the CAN bus connections by adding a link between the TA and TB terminals.

CAN software functionality is included in a MIAC flowchart by adding the standard Flowcode CAN component:

Please refer to the Flowcode CAN documentation and help files for further information on the configuration and use of this component.

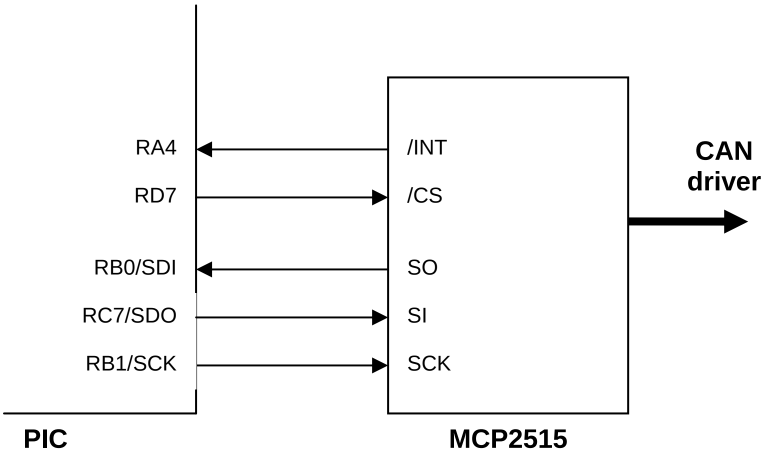
CAN offers the advantages of fast data transfer, long transmission lengths, high immunity from electrical noise, and the ability to support multiple devices sharing the same connections - making networks relatively simple to create.



CAN bus wiring showing termination

CAN bus wiring

The CAN bus is controlled by an MCP2515 CAN controller chip. The PIC communicates with it via its SPI peripheral hardware using the pins shown in the diagram. The output from the MCP2515 is fed into a CAN line driver chip and then fed to the CAN-H and CAN-L lines on the MIAC. Microchip provide plenty of information on driving the CAN bus in the MCP2515 datasheet.



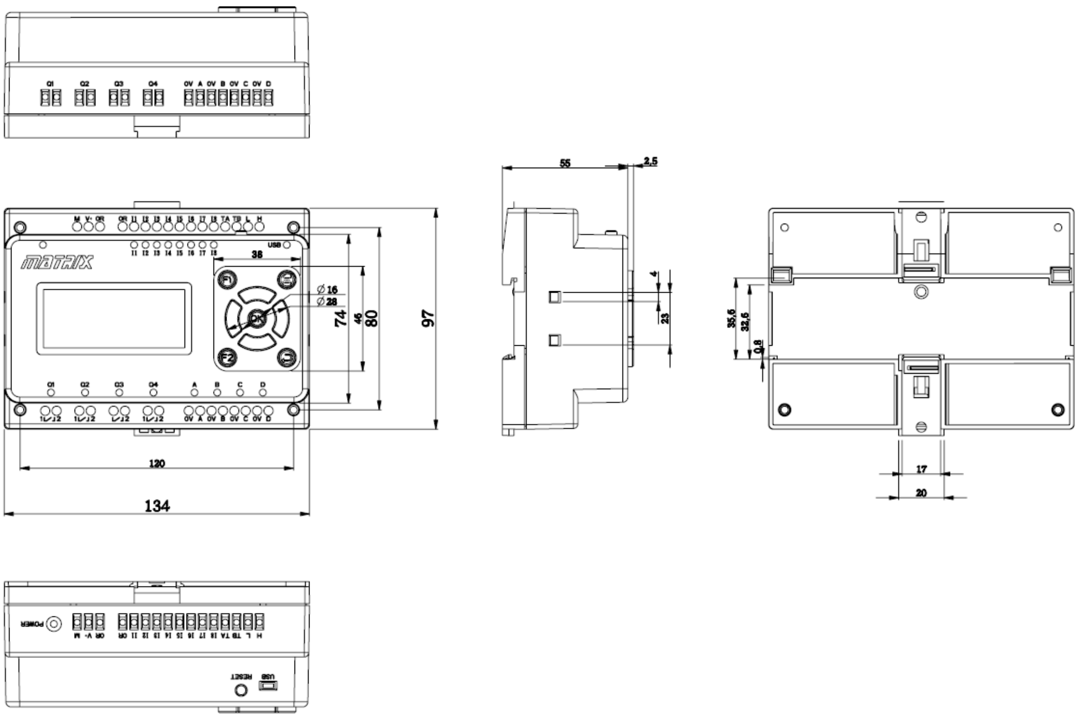
CAN bus structure

Technical specification

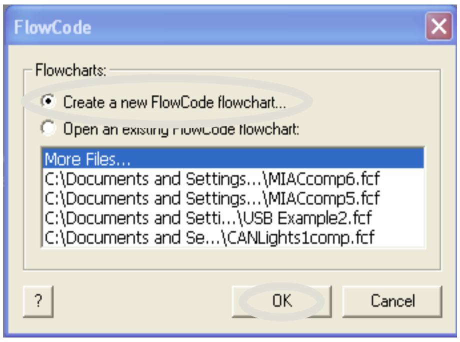
Power supply	12 - 16V, <2A
Inputs	8
Inputs usable as analogue inputs	8 - 0 to 12V
Analogue input sensitivity	10mV
Input impedance	10kΩ
Input Voltage Low	0V – 3V
Input Voltage High	>7.5V
Max input voltage range	-30V, +45V
Relay outputs	4
Relay output ratings	8A at 240VAC, 30VDC
Transistor outputs (source and sink)	4
Transistor output (per channel)	500mA
Max transistor output - all channels	1.75A
Transistor thermal shutdown	>500mA
PWM outputs, sensitivity	A, C, 0.4%
Power supply	12/24V at 100mA
Storage temperature	-40 to +70C
Transistor supply voltage (M)	6 - 24V, 4A
Operating temperature	-5 to 50C
Programming interface	USB
Processor	PICmicro 18F4455, 12K ROM, 2 K RAM @48MHz
CAN bus processor	MCP2515 @20MHz

Ordering codes

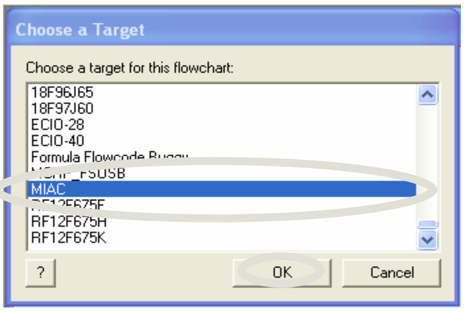
MIAC with Flowcode 3 MI0235
MIAC & 4mm shrouded sockets with Flowcode 3 MI0245
1A international power supply HP5328
Flowcode 4 TEFLCSI4



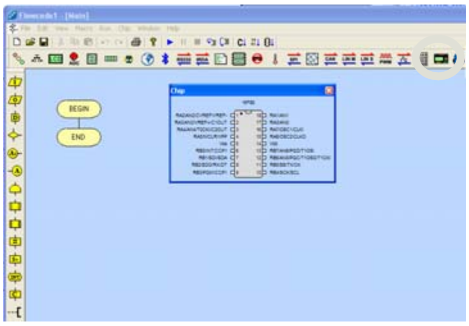
Programming example



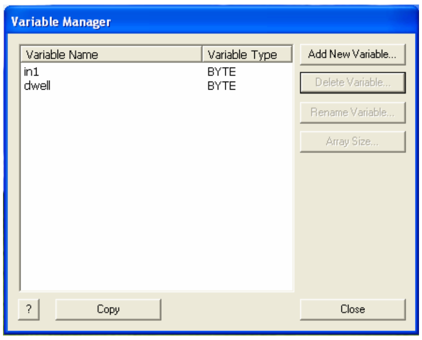
Start Flowcode and select the ‘**Create a new flowcode flow-chart**’ option



Select **MIAC** as the target device

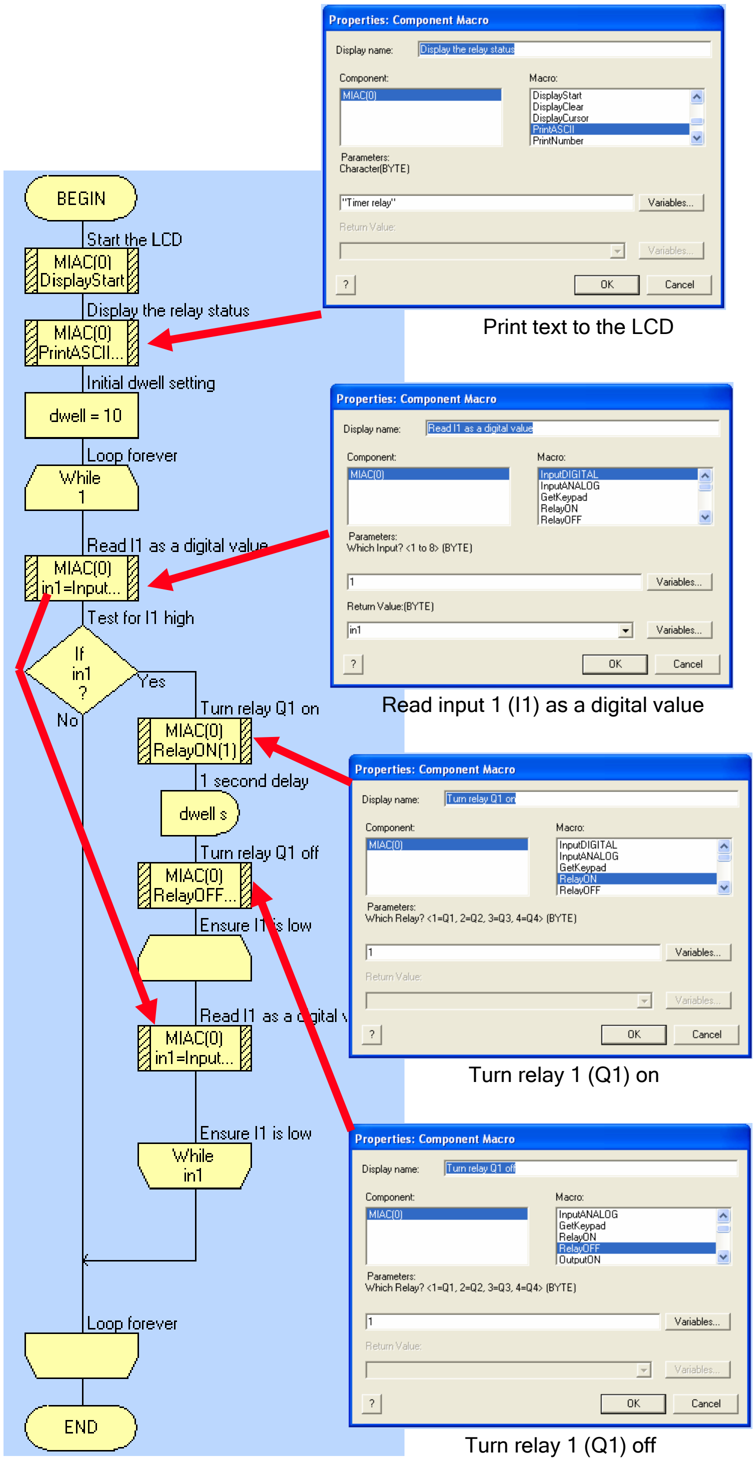



Select the MIAC component from the Flowcode component bar



Use the Edit -> Variables menu to create the two ‘byte’ variables required by the program.

Create the following timer relay example program..........



Print text to the LCD

Read input 1 (I1) as a digital value

Turn relay 1 (Q1) on

Turn relay 1 (Q1) off

Simulation

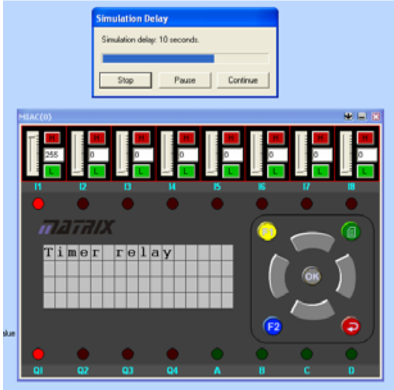
Run the program in simulation

Press the H button above I1 to:

- Illuminate the I1 LED
- Set the I1 analog value to 255
- Cause the Q1 LED to be illuminated for a simulated 10 seconds

Press the L button above I1 to:

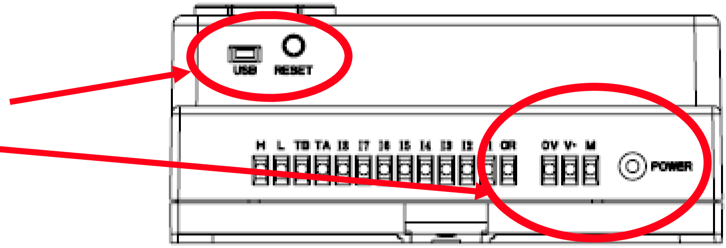
- Extinguish the I1 LED
- Set the I1 analog value to 0
- Allow the timer to be re-triggered if it has completed its previous delay



Programming

Connect a USB cable from the PC to the MIAC

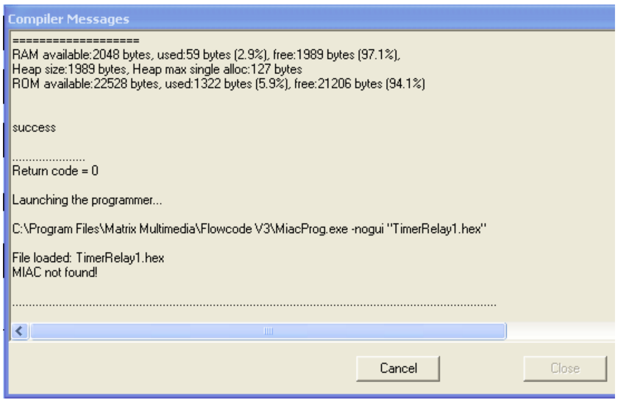
Connect the MIAC to suitable power supply.

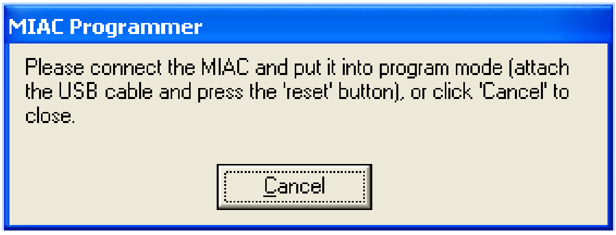


Use the Flowcode 'compile to chip' icon to start the download process.

If the MIAC is not ready to receive a new program the '**MIAC not found!**' message will be displayed and the MIACprog panel will provide instructions to connect the USB lead to the MIAC and press its RESET button. This will allow MIACprog to start communicating with the MIAC and download the program.

If the MIAC is already connected and in a reset condition, with the USB LED flashing, this process will be completed automatically





The MIAC will start running the program as soon as the download has been completed.

The timer can be triggered by making a momentary connection between the V+ terminal and the I1 terminal (a suitable voltage is available on the V+ terminal even if the MIAC is being powered from the POWER socket).

- The I1 LED will be illuminated for the duration of the input signal.
- The Q1 relay and LED will be energised for 10 seconds after each detection of a positive signal edge on the I1 terminal.
- Signals on the I1 terminal while the relay is energised will be ignored