

MIAC



locktronics™

Internet of Things



MATRIX

www.matrixtsl.com

Copyright 2019 Matrix Multimedia

Introduction

The Matrix TSL Internet of Things kit (MI0899) is designed to allow students to study the Internet of Things and Industry 4.0 for automation and control. The kit is based on a Raspberry Pi MIAC controller running Raspbian and Node-RED software. The MIAC controller has an LCD display, keypad, 8 analogue inputs, 4 transistor (FET) outputs and 4 relay outputs. It is housed in a plastic case with 4mm connectors that allows students to easily build a number of circuits with sensors (switches, potentiometer, light, temperature) and actuators (motor, solenoid, bulb). A Locktronics panel and several components are included so that circuits can be quickly put together.

The unit is supplied with MIAC specific Node-RED nodes that can be easily used to control the input and output signals. This associated briefing document shows students how to get the hardware and software up and running and gives suggested applications that the students must put together to learn how systems will be controlled using IP technology in the future.

Prerequisites

For internet access the MIAC will need to be configured to access your installation network. Hence the MIAC needs to be configured as a client to your Wi-Fi access point and DHCP router. This is explained in the Appendix “Initial setup of the MIAC Wi-Fi” section.

A host PC with connectivity to the same network as the MIAC and an internet browser is needed for access to the MIAC. A Linux based PC has advantages, but for this article we assume the use of a Windows PC.

A development environment is not required, however there are several tools that might be useful, and these can be optionally installed on the Windows PC.

- Mosquitto (MQTT broker)
- MQTT.fx (MQTT client)
- Putty (SSH client)
- Raspberry Pi “RPi Boot” (USB utility)
- Win32DiskImager (Disk image copy utility)

Internet of things

The Internet of things (IoT) is a term that describes the adoption of Internet connectivity onto physical devices and everyday objects. Internet connectivity extends into embedded electronics within many forms of hardware devices, which include control, monitoring and sensors, that can communicate and interact with others over the Internet. Additionally, they can be remotely monitored and controlled.

Industry 4.0 is a name given to the current trend of automation and data exchange in manufacturing technologies. It includes the Internet of things, machine control system and cloud computing that can combine to produce what is often referred to as a “smart factory”. Industry 4.0 is commonly considered as the fourth industrial revolution and involves the cooperation of physical systems, processes, machine and human communication in real-time both internally and across organisations.

Learning objectives

The Matrix IoT kit (MI0899) and this guide is designed to provide the following learning objectives:

- Sense and actuator circuits based on industrial controllers
- Node-RED software and flows
- Construction of browser-based GUIs using Node-RED
- Communications protocols including MQTT
- IOT data security
- Cloud based applications and data processing

The MIAC and Locktronics hardware

Central to Raspberry Pi based Matrix MIAC PLC unit is a Compute Module running Raspbian Linux. The MIAC unit contains a Wi-Fi module for network communication and the Linux environment supports numerous protocols for intranet and internet connections. This together with industrial type hardware input and output interfaces of the MIAC make it therefore ideal for IoT and Industry 4.0 learning.

Inputs and outputs of the MIAC

The MIAC unit has eight analogue inputs, marked I1 to I8, that have 12 bit analogue to digital convertors that give values of 0 to 960 across the 0 to 12 volt input range.

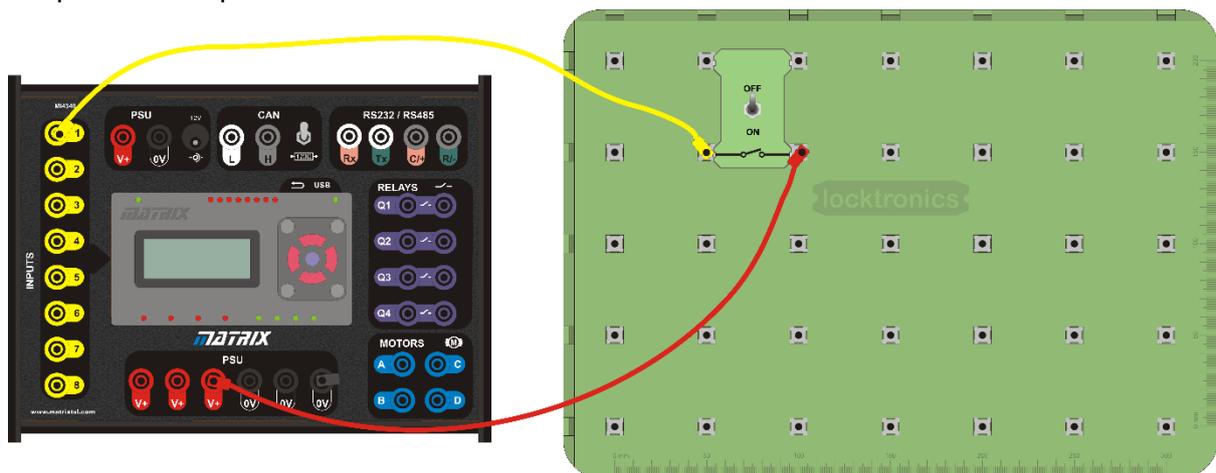
The outputs consist of four single-pole normally open relays, Q1 to Q4. Each pair of terminals provide access to the switch contacts of each relay which are isolated from each other. Each can switch up to 8A at 250VAC or 30VDC.

There are also four solid state outputs that can each sink or source up to 1 Amp and a total source current not exceeding 2.8 Amps. Inputs and outputs have an associated status LED.

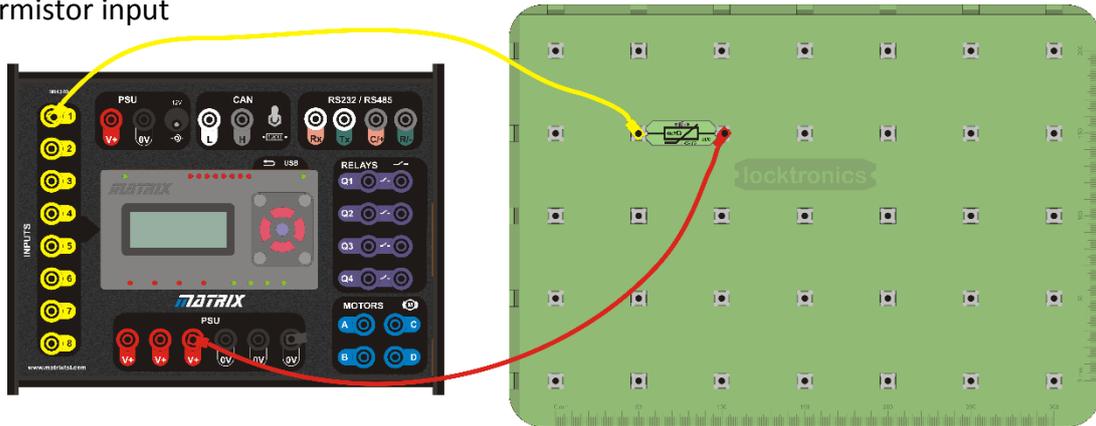
Example hardware interfacing

The Matrix IoT kit enables the demonstration of use of motors, sensors etc., in an industrial context. Here are some examples of use:

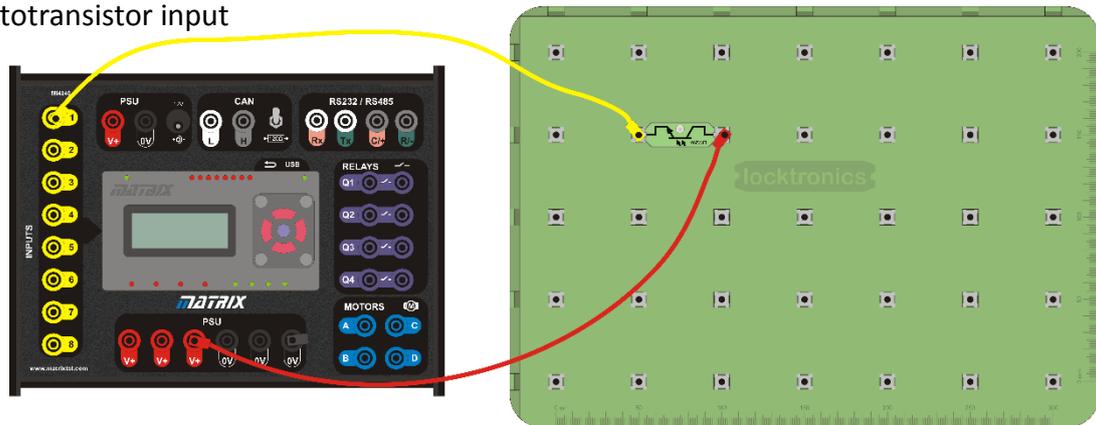
Simple switch input



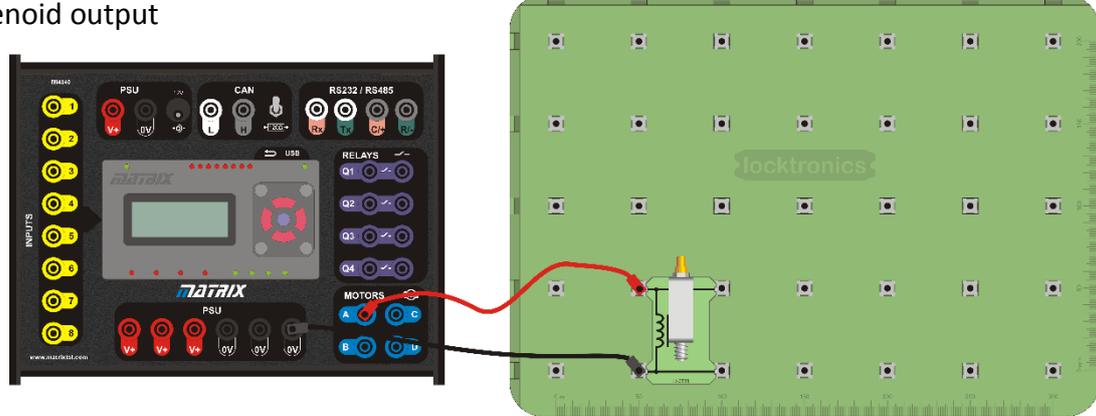
Thermistor input



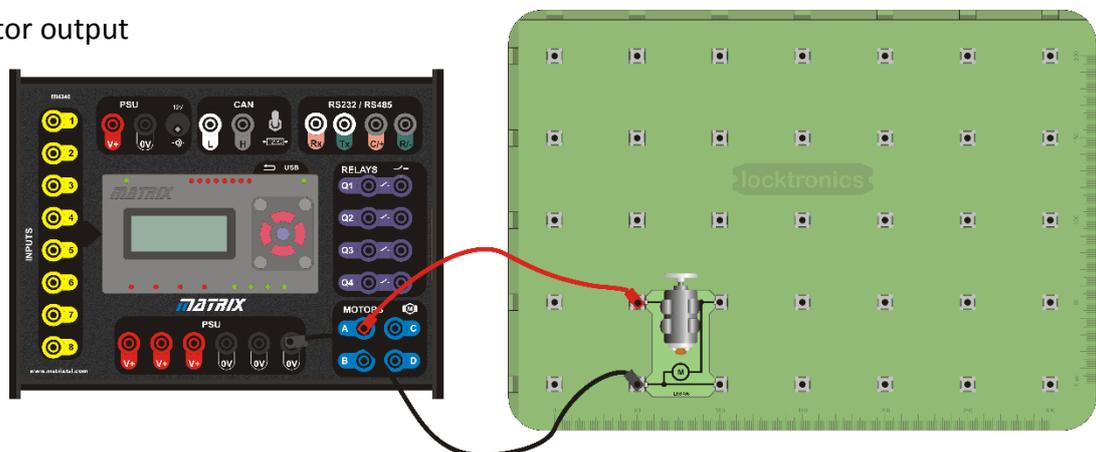
Phototransistor input



Solenoid output



Motor output



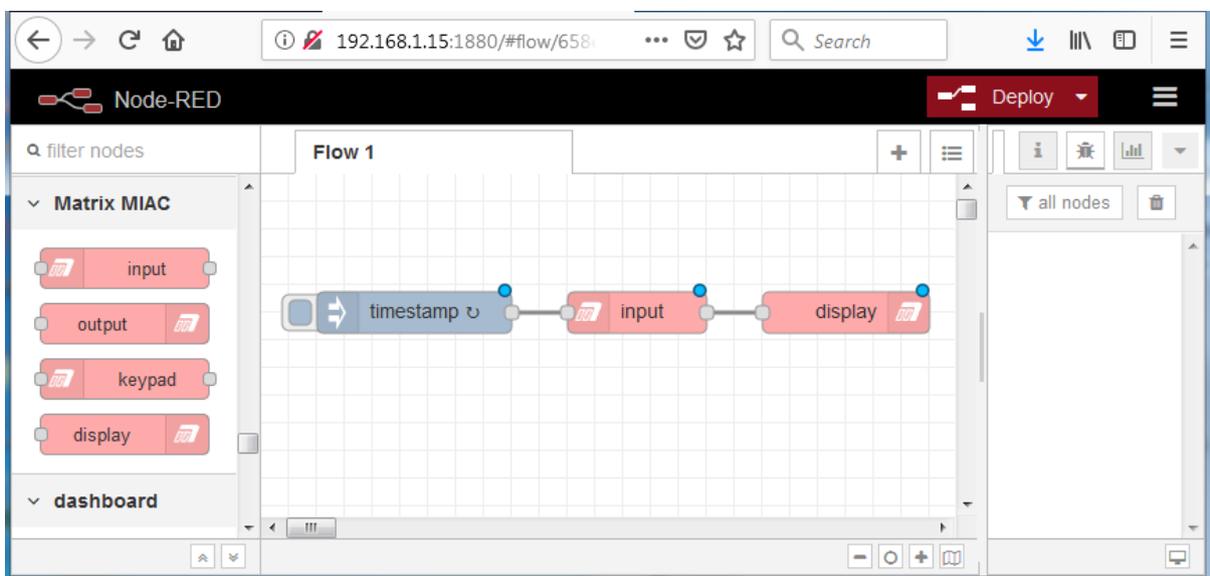
Connecting to the MIAC

The default boot script of the MIAC will display the Linux kernel version and the network IP address of the MIAC. Then in the PC browser go to the IP address of the MIAC and port 1880. For example, 192.168.0.101:1880

Brief introduction to Node-RED

Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things. Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions. Elements of applications can be saved or shared for re-use. The runtime is built on Node.js.

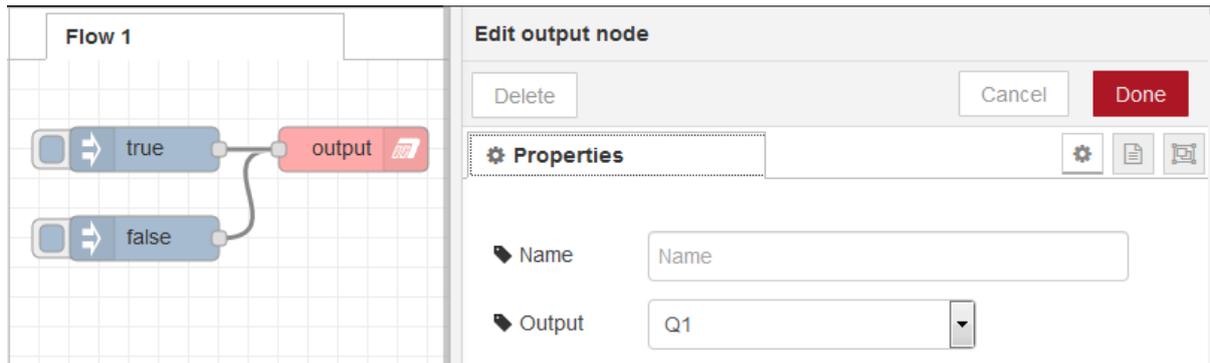
The flow editor can be seen in a browser such as below. To create a flow simply drag nodes from the pallet onto the window and connect with wires by dragging from the node connection points.



How to use the MIAC specific nodes

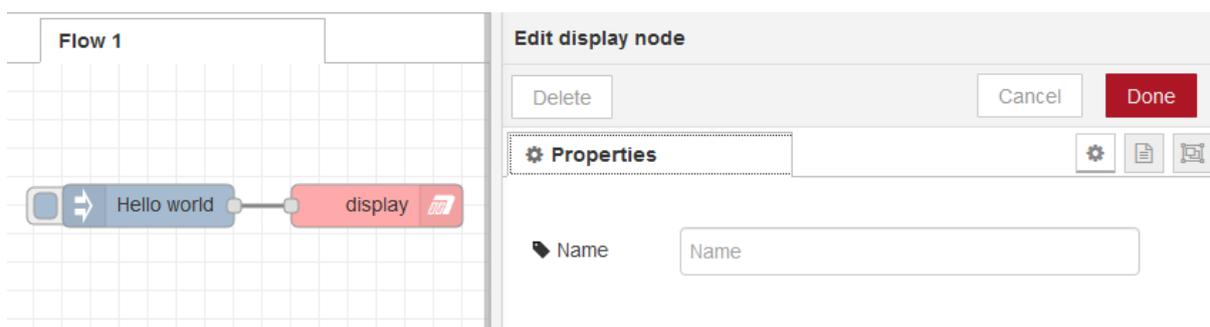
The Node-RED node pallet contains some MIAC specific nodes that allow the use of MIAC inputs and outputs easily within the flows.

Digital output (relay and solid-state driver) node



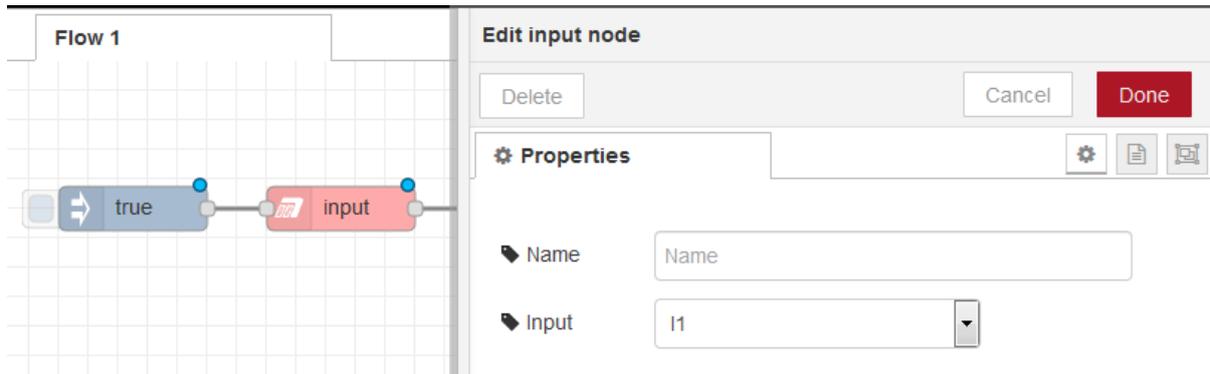
The MIAC relays and solid-state outputs are controlled with the “output” node. This node has an input that accepts Boolean or numeric message payloads and sets the output if the values are non-zero or true. The property “Output” determines the selected output.

Display node



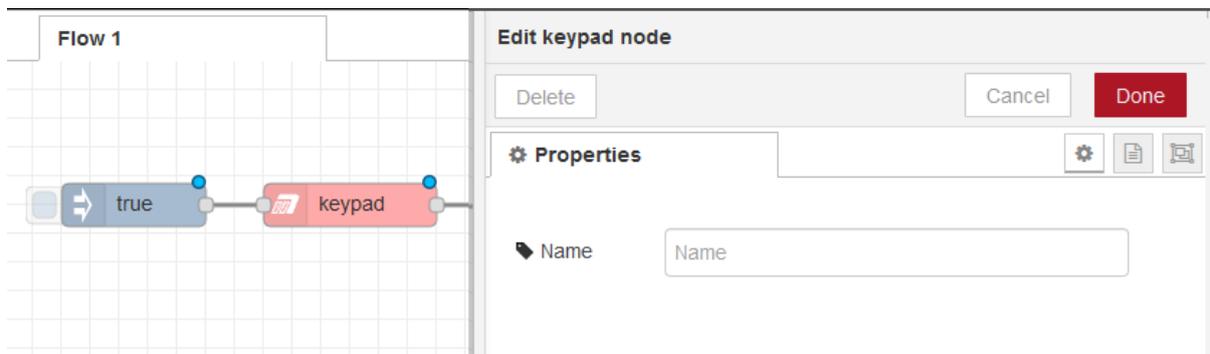
The “display” node is used to output text to the MIAC display screen. The message payload text is transferred to the display screen. To clear the screen the form feed character is supported. To clear the screen set the `msg.payload = String.fromCharCode(12)`

Analog input node



To take a reading from an analogue input add an “input” node to the flow. Set the “Input” property to the value I1 to I8 corresponding to the MIAC inputs. Trigger a reading by sending a message to the “input” node. An output message from the node will contain the ADC value as a string between “0000” and “1023”.

Keypad node



To get the current state of the MIAC keypad send a trigger message to the “keypad” node.

An output message from the node will be sent if the keypad state is currently different to the previously read state. In this case the payload string will contain a value from “0” to “9”, where “0” represents no key pressed and the values “1” to “9” represent the keys pressed as marked in the diagram.



IoT principles and experiments

The following section gives a brief introduction and overview of IoT principles and how they can be demonstrated by the student implementing the suggested experiments.

Control of an actuator from a Node-RED flow

Industrial control at its simplest involves the control of actuators such as solenoids and motors.

Create a flow that uses a trigger node to switch on a MIAC output to drive a solenoid or motor.

Input of sensor data using a Node-RED flow

We require the monitoring of sensor data such as a limit switch, a temperature sensor such as a thermistor or a light sensor.

Create a flow to read inputs to the MIAC and display the read values to the debug output window.

Communication between two IoT devices

Sensor input changes at one IoT device could require action at another IoT device.

Create a flow on one IoT MIAC that reads an input switch and sends that data to a second MIAC that runs a flow that then controls a relay based on the data. Hint: The UDP input and output nodes are a simple way of transferring data between IP connected devices.

Construction of a Node-RED GUI dashboard

Open a second browser window and connect to the same url as the flow editor with the addition of /ui For example 192.168.0.1:1880/ui and the dashboard will be visible.

Create flows that include dashboard nodes of switch control and sensor gauges to control and display data from the MIAC.

Introduction to IoT protocols

There are many protocols and wireless technologies, such as MQTT, CoAP, DDS, AMQP, LoRaWAN, Bluetooth, ZigBee, Wi-Fi and NFC, used by IoT applications. The MQTT connectivity protocol, in particular was designed as an extremely lightweight, low power, low bandwidth messaging transport. The MQTT protocol consists of three core components or mechanisms: Publisher, Broker and Subscriber. This provides for efficient distribution of information to one or many receivers. There are a number of freely available tools for testing, demonstrating and using the MQTT protocol:

- Eclipse Mosquitto is an open source message broker that implements the MQTT protocol, providing services to both Publisher and Subscriber.
- MQTT.fx is an open source MQTT client that can Publish and Subscribe to a broker.

Create a Node-RED flow that uses the MQTT node to publish data messages to a broker, such as Mosquitto and examine the data by subscribing to the broker, for example with MQTT.fx

IoT security

IoT security is the technology concerned with safeguarding connected devices and the networks between them. Allowing devices to connect to the internet potentially exposes them to any number of serious vulnerabilities if they are not properly protected. Therefore, implementing security measures is critical to ensuring the safety of networks with IoT devices connected to them. It is essential not only to protect the integrity of data being sent from IoT devices to central systems, but also ensure only authorized devices connect with such systems. Basic to this protection is the use of encryption and authentication.

Publishing of data to Cloud based server

Create a test account with a cloud-based service, such as ThingSpeak, and create a Node-RED flow, using network protocol nodes, to send sensor data to the cloud. Take particular note of the need for authentication information such as ID and Keys.

Subscribing and display of server data

Create a Node-RED application using flows and dashboard components to connect with the cloud-based server, retrieve information and display on the Node-RED dashboard.

Appendix

Backup and restore of the MIAC operating system

The file system of the Raspberry Pi Compute Module based MIAC can be backed up to an image file on a host PC. This can be subsequently used to restore the MIAC.

The same restore procedure can be used with a clean factory image.

You will require the Raspberry Pi Foundation USB utility “RPi Boot” installed on a host PC. This utility is available from the Raspberry Pi Foundation web site.

Note that for a Linux host it is possible to mount all the partitions of the solid-state drive of the MIAC compute module. For a Windows host only the boot partition is directly accessible via the USB utility.

You will also require a disk image copy utility, such as Win32DiskImager, installed on the host PC.

Backup procedure:

1. Run the “RPi Boot” utility and a console window should appear with a waiting message.
2. Connect the MIAC to the PC via a USB cable.
3. Connect power to the MIAC.
4. The “RPi Boot” utility will make connection with the MIAC, the console window will close.
5. The boot partition of the MIAC should now be accessible as a drive on the PC, for example drive F:
6. The disk copy utility, such as Win32DiskImager, can now be run to “Read” the data from this drive into an image file on your PC. The image file will also contain the Linux partition that is not visible from the Windows PC. Note: this imaging process will take many minutes!
7. When complete, remove the USB cable and disconnect power from the MIAC.

Restore procedure:

The restore procedure is identical to the above backup procedure steps 1 to 5.

6. Use the disk copy utility, such as Win32DiskImager, to “Write” an image file to the drive allocated by the PC. Ensure that you select the correct drive! Note: this imaging process will take many minutes!
7. When complete, remove the USB cable and disconnect power from the MIAC.

Alternatively, the Wi-Fi access can be setup by editing the **wpa_supplicant.conf** file on the boot sector. See the previous section regarding use of the “RPi Boot” utility to access the boot sector.

The **wpa_supplicant.conf** file contents should contain your country and Wi-Fi login credentials:

```
country=GB
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
    ssid="your_ssid"
    psk="your_passkey"
}
```

Linux updates and the installation of Matrix specific drivers

If the Raspbian kernel version is updated by the user, then the Matrix drivers will require rebuilding. This is easily done by logging into Raspbian as the pi user and running the following commands. Note that this requires internet access.

```
cd drivers
./install.sh
```

Once built, the installation process will re-initialise the driver and the message “Booting” will appear on the MIAC display. This will confirm successful building and installation of the drivers.



Matrix Technology Solutions Ltd.

The Factory

33 Gibbet Street

Halifax, HX1 5BA, UK

t: +44 (0)1422 252380

e: sales@matrixtsl.com

www.matrixtsl.com

CP6844