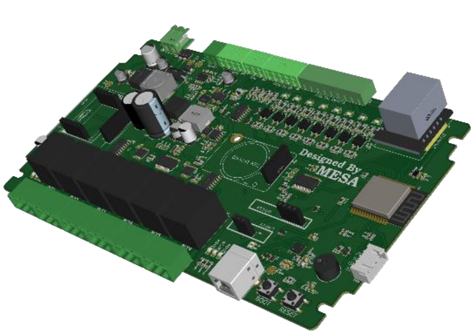
Cortex Link A8R-M ESP32 Smart Relay Board

Technical Manual



Manufactured by:
Microcode Embedded Systems and Automation (MESA)

Document Version: 1.0
Last Updated: April 2025

Table of Contents

1. Introduction
2. Technical Specifications
3. Board Layout & Components
4. Installation & Setup
5. Connection Interfaces
6. Programming Guide
7. Module Integration
8. Application Examples
9. Troubleshooting
10. Support & Resources
11. PIN Configurations

1. Introduction

The Cortex Link A8R-M ESP32 Board is an advanced IoT development platform designed and manufactured by Microcode Embedded Systems and Automation (MESA). Built around the powerful ESP32 dual-core processor, this versatile board offers extensive connectivity options and flexible I/O configurations, making it an ideal solution for smart home automation systems, industrial controls, and IoT projects.

Key Features

- **6 Relay Outputs**: Control relays with 12/24VDC 10A or 250VAC/10A capacity
- **8 Digital Inputs**: Optically isolated for monitoring switches and sensors
- **4 Analog Inputs**: 2 channels for 4-20mA and 2 channels for 0-5V DC inputs
- **2 Analog Outputs**: 0-10V DAC outputs
- **Multiple Communication Protocols**: Wi-Fi, Ethernet, Bluetooth, RS485/Modbus RTU
- **RF Communication**: 433MHz/315MHz transmitter and receiver interface
- **Cellular Connectivity**: Optional GSM module support (2G/4G)
- **Advanced Integration**: Compatible with Arduino IDE, ESPHome, Home Assistant, and other platforms

2. Technical Specifications

Microcontroller

- **Chip**: ESP32 dual-core processor
- **Architecture**: 32-bit
- **Clock Speed**: 240 MHz
- **Flash Memory**: 8 MB
- **SRAM**: 520 KB

Connectivity

- **Wi-Fi**: 802.11 b/g/n (2.4 GHz)
- **Ethernet**: RJ45 port for wired network connection (10/100 Mbps)
- **Bluetooth**: BLE 4.0 and Bluetooth Classic
- **RS485/Modbus**: For industrial communications
- **Optional GSM**: Support for external GSM module (2G/4G)

Digital I/O

- **Relay Outputs**: 6 channels (12/24V DC, 250VAC, 10A per channel)
- **Digital Inputs**: 8 optically isolated channels (dry N/O contacts)

Analog I/O

- **Analog Inputs**:
  - 2 channels: 4-20mA (Input Port 1 & 2)
  - 2 channels: 0-5VDC (Input Port 3 & 4)
  - 12-bit ADC resolution
- **Analog Outputs**:
  - 2 channels: 0-10V/0-5V DC (Programmable)

Additional Interfaces

- **I2C Interface**: For connecting external sensors and modules
- **1-Wire Interface**: 2 GPIO pins for temperature sensors
- **RF Interface**: 433MHz/315MHz transmitter and receiver

Power

- **Supply Voltage**: 9V-12V DC
- **Logic Voltage**: 5V DC
- **Relay Power**: 12V DC input
- **Power Consumption**: 1W (idle), 10W (under load)

Physical Specifications

- **Dimensions**: 200mm x 110mm x 45mm (L x W x H)
- **Operating Temperature**: -40°C to +85°C

Certifications

- CE Certified (European safety standards)
- RoHS Compliant

---

3. Board Layout & Components

The Cortex Link A8R-M board is organized into several functional sections:

Power Section

- DC Power input terminal (9-24V DC)
- Power status LED
- Power regulation circuitry

Communication Interfaces

- Ethernet RJ45 port
- RS485/Modbus terminal (A-B connections)
- USB-B connector for programming and serial communication

Input/Output Sections

- 6 relay outputs with status LEDs (RLY1-RLY6)
- Terminal blocks for relay connections (NO, COM, NC)
- 8 digital input terminals with status LEDs
- 4 analog input terminals (2x 4-20mA, 2x 0-5V)
- 2 analog output terminals (0-10V)

Expansion Modules

- Socket for 3G SIM Module (SIM800L)
- Socket for 4G SIM Module (SIM7600E)
- RF433 Receiver Module Socket
- RF433 Transmitter Module Socket
- DS3231 RTC Module Connector
- I2C Extender

Control Buttons

- BOOT button
- RESET button

Address Select Jumpers

- U46 - GP8413 Address Select (Default: 0x58)
- U26 - MCP23017 Address Select for Output Relays (Default: 0x20)
- U8 - MCP23017 Address Select for Digital Inputs (Default: 0x21)

---

4. Installation & Setup

Power Connection

1. Connect a 9-24V DC power supply to the DC power input terminals, observing correct polarity (+ and -).
2. Verify that the DC Power Status LED illuminates, indicating proper power connection.

Network Connection

Ethernet Connection

1. Connect an Ethernet cable to the RJ45 port.
2. The Ethernet status LED will indicate connectivity.

Wi-Fi Connection

The board comes with built-in Wi-Fi connectivity. Configuration is done through the programming interface.

Initial Configuration

1. Connect the board to your computer using the provided USB cable.
2. Install the appropriate drivers if needed.
3. Configure the board using one of the supported development environments (Arduino IDE, ESPHome, ESP-IDF, or PlatformIO).

---

5. Connection Interfaces

Relay Outputs (RLY1-RLY6)

Each relay provides three connection points:

- **NO**: Normally Open contact
- **COM**: Common contact
- **NC**: Normally Closed contact

Relay specifications:

- Maximum switching voltage: 250VAC or 24VDC
- Maximum current: 10A per channel
- Control via MCP23017 GPIO expander (U26)

Digital Inputs (IN1-IN8)

- 8 channels of optically isolated digital inputs
- Designed for dry N/O contacts
- Status indicated by Digital Input Status LEDs
- Controlled via MCP23017 GPIO expander (U8)

Analog Inputs

- **IN1-IN2**: 4-20mA current loop inputs (industry standard)
- **IN3-IN4**: 0-5VDC voltage inputs
- 12-bit ADC resolution

Analog Outputs

- **OUT1-OUT2**: 0-10V outputs
- Control via GP8413 DAC chip (U46)

RS485/Modbus

- Terminal blocks labeled A and B for RS485 serial communication
- Used for Modbus RTU protocol communications with industrial devices

GPIO Pins

- Multiple GPIO pins available for custom functionality
- See the PIN Configurations section for details

---

6. Programming Guide

The Cortex Link A8R-M board can be programmed using several development environments:

Arduino IDE

1. Install the Arduino IDE from the official website.
2. Install ESP32 board support:
   - Go to File > Preferences
   - Add the following URL to Additional Boards Manager URLs: https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
   - Go to Tools > Board > Boards Manager and install ESP32
3. Select "ESP32 Dev Module" from the Tools > Board menu
4. Connect the board via USB and select the appropriate COM port
5. Required libraries:
   - For GP8413: https://github.com/adafruit/Adafruit-MCP23017-Arduino-Library.git
   - For MCP23017: https://github.com/adafruit/Adafruit-MCP23017-Arduino-Library.git

ESPHome

For Home Assistant integration:

1. Add the following to your ESPHome configuration:

```
esphome:
  name: cortexlink_a8r_m
  platform: ESP32
  board: esp32dev

# Wi-Fi configuration
wifi:
  ssid: "Your_SSID"
  password: "Your_Password"

# Enable API for Home Assistant
api:

# Enable OTA updates
ota:

# I2C Bus configuration
i2c:
  sda: GPIO21
  scl: GPIO22
  scan: true

# MCP23017 for relay outputs (U26)
mcp23017:
  - id: 'relay_mcp'
```

```
    address: 0x20

# MCP23017 for digital inputs (U8)
mcp23017:
  - id: 'input_mcp'
    address: 0x21

# Define relay outputs
switch:
  - platform: gpio
    name: "Relay 1"
    pin:
      mcp23017: relay_mcp
      number: 0
      mode: OUTPUT
    id: relay_1

# Define more relays as needed...

# Define binary sensors for the digital inputs
binary_sensor:
  - platform: gpio
    name: "Input 1"
    pin:
      mcp23017: input_mcp
      number: 0
      mode: INPUT
      inverted: true
    id: input_1


# Define more inputs as needed...


# Define analog sensors
sensor:
  - platform: adc
    pin: GPIO36
    name: "Analog Input 1"
    unit_of_measurement: "mA"
    accuracy_decimals: 2
    filters:
      - lambda: |-
          return (x * 20.0 / 4095.0) + 4.0; # Convert to 4-20mA range
```

ESP-IDF

For more advanced development:

1. Set up the ESP-IDF development environment following the Espressif documentation
2. Create a new project or use an example project as a base
3. Configure the project for the ESP32 target
4. Build and flash the project to the board

7. Module Integration

GSM Module

1. Insert a compatible SIM card into the SIM holder on the GSM module
2. Connect the GSM module to the appropriate socket:
   - SIM800L for 2G connectivity (Socket A/B)
   - SIM7600E for 4G connectivity (Socket C)
3. Use the provided library to initialize and control the GSM module

RF433 Modules

1. Connect the RF433 Receiver to Socket D
2. Connect the RF433 Transmitter to Socket E
3. Use the appropriate RF433 libraries to send and receive signals

RTC Module

The DS3231 RTC module is pre-integrated on the board, providing accurate timekeeping. Access it through the I2C interface.

---

8. Application Examples

Smart Home Control System

```
#include <Wire.h>
#include <Adafruit_MCP23017.h>

Adafruit_MCP23017 relayMCP;
Adafruit_MCP23017 inputMCP;

void setup() {
  Serial.begin(115200);

  // Initialize I2C
  Wire.begin();
  // Initialize relay MCP23017
  relayMCP.begin(0x20);
  for (int i = 0; i < 6; i++) {
    relayMCP.pinMode(i, OUTPUT);
    relayMCP.digitalWrite(i, LOW);
  }

  // Initialize input MCP23017
  inputMCP.begin(0x21);
  for (int i = 0; i < 8; i++) {
    inputMCP.pinMode(i, INPUT);
    inputMCP.pullUp(i, HIGH); // Enable pull-up resistors
  }
}

void loop() {
  // Example: Read digital input 1 and control relay 1
  int inputState = inputMCP.digitalRead(0);
  relayMCP.digitalWrite(0, inputState);

  delay(100);
}
```

Industrial Monitoring with Modbus

```
#include <ModbusMaster.h>
#include <Wire.h>
#include <Adafruit_MCP23017.h>
```

```
#define MAX485_RE_NEG 25  // Connect to GPIO25
#define MAX485_DE     26  // Connect to GPIO26

ModbusMaster modbus;
Adafruit_MCP23017 relayMCP;

void preTransmission() {
  digitalWrite(MAX485_RE_NEG, HIGH);
  digitalWrite(MAX485_DE, HIGH);
}

void postTransmission() {
  digitalWrite(MAX485_RE_NEG, LOW);
  digitalWrite(MAX485_DE, LOW);
}

void setup() {
  pinMode(MAX485_RE_NEG, OUTPUT);
  pinMode(MAX485_DE, OUTPUT);

  digitalWrite(MAX485_RE_NEG, LOW);
  digitalWrite(MAX485_DE, LOW);

  Serial2.begin(9600, SERIAL_8N1, 16, 17);  // RX=GPIO16, TX=GPIO17

  modbus.begin(1, Serial2);
  modbus.preTransmission(preTransmission);
  modbus.postTransmission(postTransmission);
```

```
 // Initialize relay MCP23017
 Wire.begin();
 relayMCP.begin(0x20);
 for (int i = 0; i < 6; i++) {
  relayMCP.pinMode(i, OUTPUT);
  relayMCP.digitalWrite(i, LOW);
 }
}

void loop() {
 // Read holding registers from a Modbus device
 uint8_t result = modbus.readHoldingRegisters(0, 10);

 if (result == modbus.ku8MBSuccess) {
  // Process data and control relays based on Modbus register values
  for (int i = 0; i < 6; i++) {
   if (modbus.getResponseBuffer(i) > 500) {
    relayMCP.digitalWrite(i, HIGH);
   } else {
    relayMCP.digitalWrite(i, LOW);
   }
  }
 }

 delay(1000);
}
```

Home Assistant Integration with ESPHome

Configuration example for controlling relays and reading sensors:

```
# Control relay based on sensor value
automation:
 - alias: "Turn on relay when analog input exceeds threshold"
  trigger:
   platform: numeric_state
   entity_id: sensor.analog_input_1
   above: 10.0
  action:
   service: switch.turn_on
   target:
    entity_id: switch.relay_1
```

---

9. Troubleshooting

Power Issues

- **Problem**: Board does not power on (no LED indicators)
  - **Solution**: Check power supply voltage (9-24V DC), polarity, and connections

Communication Issues

- **Problem**: Cannot connect via USB
  - **Solution**: Ensure proper drivers are installed, try another USB cable
- **Problem**: Wi-Fi connection fails
  - **Solution**: Verify Wi-Fi credentials, check signal strength, restart the board
- **Problem**: Ethernet connection issues
  - **Solution**: Verify cable connection, check network settings

Relay Control Issues

- **Problem**: Relays not switching
  - **Solution**: Verify MCP23017 address settings (U26), check I2C connections, ensure relay power is connected

Input Sensing Issues

- **Problem**: Digital inputs not detecting
  - **Solution**: Check input connections, verify MCP23017 address (U8), test with simple test program

Programming Issues

- **Problem**: Cannot upload sketch
  - **Solution**: Hold BOOT button while initiating upload, then release

10. Support & Resources

Online Resources

- Arduino Libraries:
  - MCP23017: https://github.com/adafruit/Adafruit-MCP23017-Arduino-Library.git
  - GP8413: https://github.com/adafruit/Adafruit-MCP23017-Arduino-Library.git
- ESPHome Configuration Examples:
  - https://esphome.io/components/mcp23017.html
  - https://esphome.io/components/sensor/adc.html

Contact Information

For technical support, customization options, or ordering inquiries:

- Microcode Embedded Systems and Automation (MESA)
- Website: www. microcodeeng.com

11. PIN Configurations

ESP32 PIN Configuration

ESP32 Pin No	Pin Name	Function Control
3	EN	Reset Button
4	(SENSOR_VP) GPIO36	0-5V Analog Input Channel 1 (CN1 Pin 4)
5	(SENSOR_VN) GPIO39	0-5V Analog Input Channel 2 (CN1 Pin 3)
6	GPIO34	4-20mA Analog Input Channel 1 (CN1 Pin 2)
7	GPIO35	4-20mA Analog Input Channel 2 (CN1 Pin 1)
8	GPIO32	433MHz RF Transmitter TX
9	GPIO33	433MHz RF Receiver RX
10	GPIO25	GSM SIM800L/SIM7600E TX
11	GPIO26	GSM SIM800L/SIM7600E RX
12	GPIO27	MAX485 TX/RX Control pin for MODBUS (RS485)
13	GPIO14	U8 MCP23017 I2C INPUT Expander PORT A external Interrupt
14	GPIO12	Not Used
16	GPIO13	U8 MCP23017 I2C INPUT Expander PORT B external Interrupt (CN3.8 Digital Inputs)
23	GPIO15	DHT22/AM2302/Temperature/Humidity Sensor Input Channel 2 (CN1 Pin2)
24	GPIO2	BEEP
25	GPIO0	BOOT Enable
26	GPIO4	DHT22/AM2302/Temperature/Humidity Sensor Input Channel 1 (CN1 Pin1)

ESP32 Pin No	**Pin Name**	**Function Control**
27	GPIO16	RS485 MODBUS MAX485 RX pin (RXD)
28	GPIO17	RS485 MODBUS MAX485 TX pin (TXD)
29	GPIO5	ETHERNET W5500 MODULE SPI Chip Select
30	GPIO18	ETHERNET W5500 MODULE SPI SCLK
31	GPIO19	ETHERNET W5500 MODULE SPI MISO
33	GPIO21	I2C SDA
34	RXD0	Debug/Programming (USB) RX
35	TXD0	Debug/Programming (USB) TX
36	GPIO22	I2C SCK
37	GPIO23	ETHERNET W5500 MODULE SPI MOSI

MCP23017 PIN Configuration (INPUT Interface - U8)

MCP23017 Pin No	Pin Name	Pin Function
1	GPB0	Digital Input 1 (CN3 Pin8)
2	GPB1	Digital Input 2 (CN3 Pin7)
3	GPB2	Digital Input 3 (CN3 Pin7)
4	GPB3	Digital Input 4 (CN3 Pin6)
5	GPB4	Digital Input 5 (CN3 Pin5)
6	GPB5	Digital Input 6 (CN3 Pin4)
7	GPB6	Digital Input 7 (CN3 Pin3)
8	GPB7	Digital Input 8 (CN3 Pin2)
9	VDD	3.3V
10	VSS	GND
11	NC	NC
12	SCK	I2C SCK
13	SDA	I2C SDA
14	NC	NC
15	A0	A0 (ADDRESS Line 0)
16	A1	A1 (ADDRESS Line 1)
17	A2	A2 (ADDRESS Line 2)
18	Reset	Reset
19	INTB	MCP23017-PORTB Interrupt connected to ESP32 GPIO13
20	INTA	MCP23017-PORTA Interrupt connected to ESP32 GPIO14

MCP23017 Pin No	Pin Name	Pin Function
21	GPA0	NC
22	GPA1	NC
23	GPA2	NC
24	GPA3	NC
25	GPA4	NC
26	GPA5	U9 ETHERNET W5500 MODULE RESET
27	GPA6	NC
28	GPA7	U42 DS3231 RTC SQW PIN

MCP23017 PIN Configuration (OUTPUT Interface - U26)

MCP23017 Pin No	Pin Name	Pin Function
1	GPB0	NC
2	GPB1	NC
3	GPB2	NC
4	GPB3	NC
5	GPB4	NC
6	GPB5	NC
7	GPB6	NC
8	GPB7	NC
9	VDD	3.3V
10	VSS	GND

MCP23017 Pin No	Pin Name	Pin Function
11	NC	NC
12	SCK	I2C SCK
13	SDA	I2C SDA
14	NC	NC
15	A0	A0 (ADDRESS Line 0)
16	A1	A1 (ADDRESS Line 1)
17	A2	A2 (ADDRESS Line 2)
18	Reset	Reset
19	INTB	NC
20	INTA	NC
21	GPA0	K1 Relay 1 OUTPUT (Digital Output 1)
22	GPA1	K2 Relay 2 OUTPUT (Digital Output 2)
23	GPA2	K3 Relay 3 OUTPUT (Digital Output 3)
24	GPA3	K4 Relay 4 OUTPUT (Digital Output 4)
25	GPA4	K5 Relay 5 OUTPUT (Digital Output 5)
26	GPA5	K6 Relay 6 OUTPUT (Digital Output 6)
27	GPA6	NC
28	GPA7	NC

GP8413 PIN Configuration (Analog OUTPUT Interface - U46)

GP8413 Pin No	Pin Name	Pin Function
1	SCK	I2C SCK
2	SDA	I2C SDA
3	A0	A0 (ADDRESS Line 0)
4	A1	A1 (ADDRESS Line 1)
5	VCC	12V Supply
6	GND	GND
7	VOUT1	0-5V / 0-10V Analog Output Channel 1
8	VOUT2	0-5V / 0-10V Analog Output Channel 2
9	A4	A2 (ADDRESS Line 2)
10	VSS	GND