Run Azure IoT Edge Runtime on reTerminal DM device running Debian 11 (ARM32v7)

Table of Contents

- Introduction
- Step 1: Prerequisites
- Step 2: Prepare your Device
- Step 3: Manual Test for Azure IoT Edge on device
- Step 4: Additional information
- Step 5: Additional Links

Introduction

reTerminal DM is a Panel PC, HMI, PLC, IIoT Gateway all-in-one device powered by Raspberry Pi CM4, with 10.1” IP65 front panel and rich industrial interfaces, and natively integrated with Node-RED and supports Raspberry Pi-based software ecosystem.

This document describes how to connect reTerminal DM device running Debian 11 (ARM32v7) with Azure IoT Edge Runtime pre-installed and Device Management. This multi-step process includes:

- Configuring Azure IoT Hub
- Registering your IoT device
- Build and Deploy client component to test device management capability

Step 1: Prerequisites
You should have the following items ready before beginning the process:

- Create an Azure account
- Sign up to Azure Portal
- Setup your IoT hub
- Add the Edge Device
- Add the Edge Modules

**Step 2: Prepare your Device**

1. Power on the device. Connect a 12~24V power supply to the 2-pin power terminal block connector.

   ![Power Supply Connection](image)

2. Check the OS version, bit number and architecture.

   ```bash
   $ lsb_release -irc
   Distributor ID: Raspbian
   Release: 11
   Codename: bullseye
   $ getconf LONG_BIT
   32
   $ uname -m
   armv7l
   ```

   If your OS is older than Debian 11 (Bullseye) or LOG_BIT is 64, please install latest Raspberry Pi OS 32-bit. Refer to [Steps for Flashing Raspbian OS](#).

   If `uname -m` is aarch64, your OS running 64-bit kernel. Please change to 32-bit kernel. Refer to [32-bit OS driver](#).

**Step 3: Manual Test for Azure IoT Edge on device**

3.1 Edge Runtime Enabled

1. Register your device

2. View registered devices and retrieve provisioning information
3. Install IoT Edge

```bash
$ curl https://packages.microsoft.com/config/debian/11/packages-microsoft-prod.deb > ./packages-microsoft-prod.deb
$ sudo apt install ./packages-microsoft-prod.deb
$ rm ./packages-microsoft-prod.deb
```

4. Install a container engine

```bash
$ sudo apt-get update
$ sudo apt-get install moby-engine
$ sudo vi /etc/docker/daemon.json
$ sudo systemctl restart docker
```

Set the default logging driver to the local logging driver as shown in the example below.

```json
{
    "log-driver": "local"
}
```

5. Install the IoT Edge runtime

```bash
$ sudo apt-get update
$ sudo apt-get install aziot-edge defender-iot-micro-agent-edge
```

6. Provision the device with its cloud identity

```bash
$ sudo iotedge config mp --connection-string 'PRIMARY_CONNECTION_STRING'
$ sudo iotedge config apply
```

3.2 Check the iotedge daemon

Open the command prompt on your IoT Edge device, confirm that the Azure IoT Edge Daemon is under running state

```bash
sudo iotedge system status
```
Open the command prompt on your IoT Edge device, confirm that the module deployed from the cloud is running on your IoT Edge device

```
$ sudo iotedge list
```

On the device details page of the Azure, you should see the runtime modules - edgeAgent, edgeHub and SimulatedTemperatureSensor modules are under running status

---

**Step 4: Additional information**

- reTerminal DM Getting Started

**Step 5: Additional Links**
- What is Azure IoT Edge
- Azure IoT Edge supported platforms
- Develop your own IoT Edge modules