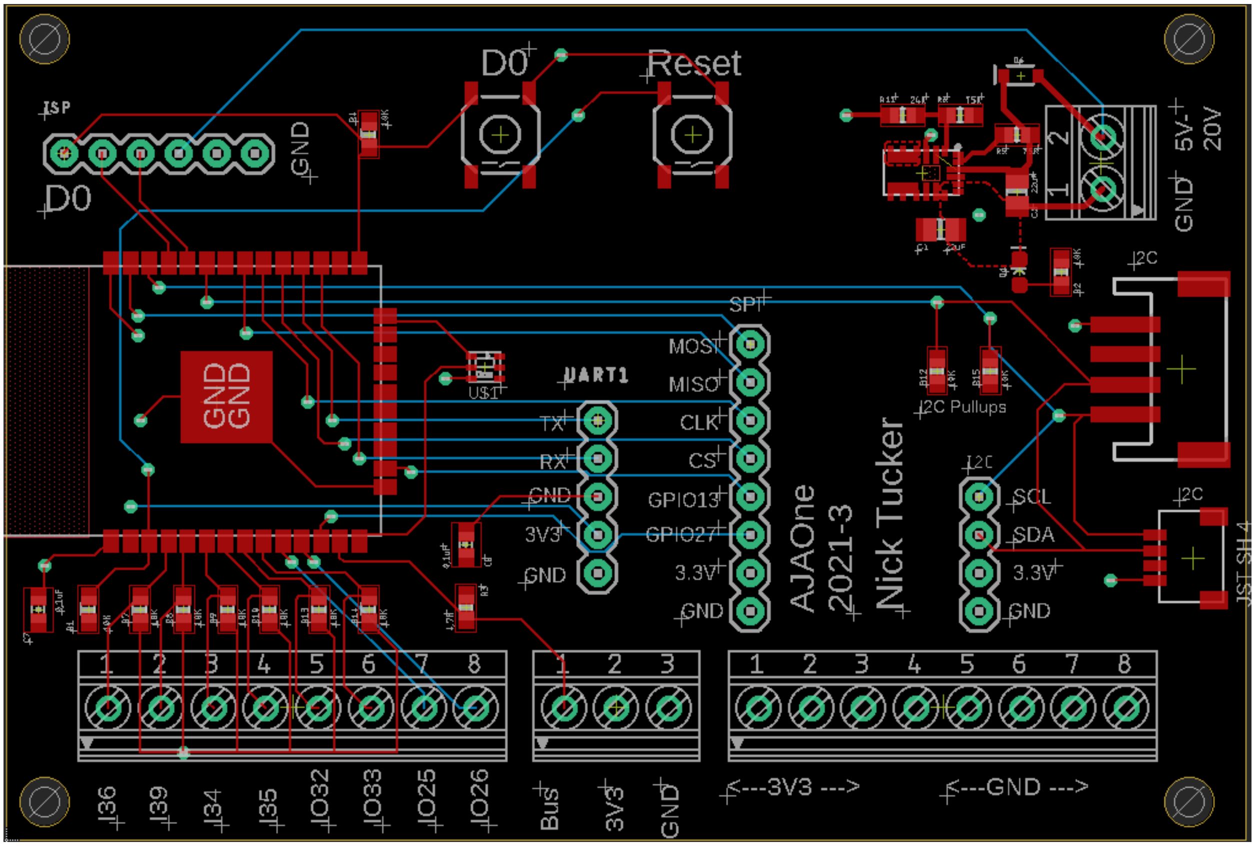
AJAOne

Sensor Hub

Version 2021-3



AJAOne Version 2021-3 Feb-2021
Nick Tucker

Contents

Introduction

AJAOne is part of the AJAOne family of devices that use the ESP32 and custom firmware to interface with an MQTT Broker. The AJAOne provides the following:

- 8 General ports, 4 are input only and 4 are input and output. 6 have pull down resistors with 2 floating. These can also be configured as analog input with minimal reprogramming.
- Dallas 1-Wire bus to interface with DS18B20 temperature sensors and similar.
- Serial port header (UART) for connecting serial devices.
- SPI header.
- I2C header.
- STEMMA and STEMMA-QT I2C connectors to interface with devices from Adafruit, Sparkfun and others.
- BME/BMP280 at address 0x76
- SSD1306 OLED at address 0x3C
- BH1750 Light sensor at address 0x23
- MFRC522 RFID card reader via SPI.

License Model

Power Requirements

Power can be supplied to either the 6 pin ISP header (5V) or via the 2 pin screw connector. If using the screw connector the power supply can be anywhere within 5v to 20v.

Do not connect power via the ISP and the screw terminals at the same time.

Initial start up

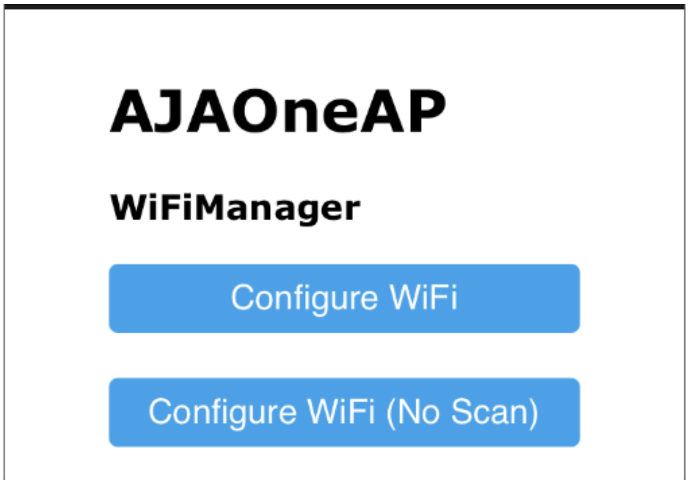
In order to use the ***AJAOne***, it must first be connected to the local WiFi followed by the MQTT Broker.

The information required will be:

1. WiFi SID (name of your WiFi access point)
2. WiFI password.
3. IP Address of the MQTT Broker.
4. MQTT Username (optional but recommended)
5. MQTT Password (optional but recommended).

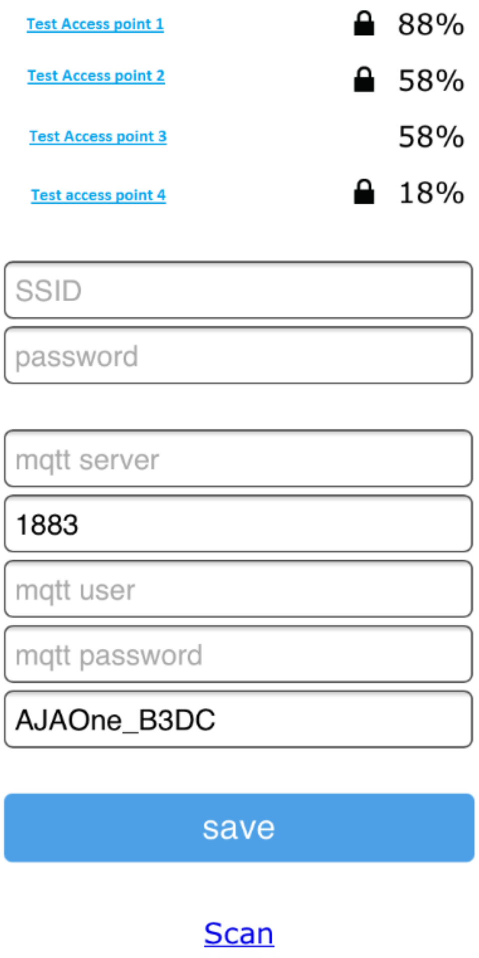
Configuration Steps

1. Connect the 5v power supply.
2. Wait approximately one minute for the system to configure itself and boot up.
3. Using a smart phone or tablet, go into Settings / WiFI and connect to the access point ***AJAoneAP***.
4. Depending on your device, the configuration page may show automatically. If not, open a browser to 192.168.4.1.



5. Click on ***Configure WiFI***. It will take a few moments to scan for all the available SIDs. If none are displayed, then move the AJAOne closer to a WiFi access point.

6. Enter all the required information and click on ***Save***.



Test Access point 1	🔒	88%
Test Access point 2	🔒	58%
Test Access point 3		58%
Test access point 4	🔒	18%

1883

AJAOne_B3DC

7. The AJAOne will reboot and connect to the WiFI and the MQTT Broker.
8. If no Broker is found within five minutes, the AJAOne will reboot and try again.

When the AJAOne starts it's led will flash once to indicate the beginning of the boot sequence. Once the entire boot sequence is completed the led will flash three times.

If for some reason the MQTT Broker cannot be reached the LED will turn blue and remain on until a connection is made. If no connection is made in five minutes, it will reboot and try again.

Addressing

Each AJAOne will have a unique name assigned to it. This name will be in the format of ***AJAONE_XXXX*** where XXXX is the last four digits of the device MAC address. For example, ***AJAONE_AD34***.

LED Codes

The on board RGB LED will indicate the current state of operation.

Yellow	-	Booting.
Purple	-	WiFi Starting.
Blue	-	Attempting MQTT connection.
Solid red	-	Performing reset.
Blue-purple	-	A command was received.

Operation

Startup

When the AJAOne starts up and connects to the WiFi and the MQTT Broker it will send its name, IP Address and the firmware version.

MQTT Messages

Message prefixed with “cmd” are message sent FROM the MQTT broker, message without “cmd” are sent from the AJAOne TO the MQTT Broker.

Digital Ports

Each time a digital port changes state an MQTT message is sent to the MQTT Broker. “on” or “off” will be returned in the message payload.

- AJAOne_xxxx/status/port36
- AJAOne_xxxx/status/port39
- AJAOne_xxxx/status/port34
- AJAOne_xxxx/status/port35
- AJAOne_xxxx/status/port32
- AJAOne_xxxx/status/port33
- AJAOne_xxxx/status/port25
- AJAOne_xxxx/status/port26

Alternately the digital status can be individually requested:

- AJAOne_xxxx/cmd/status/port36
- AJAOne_xxxx/cmd/status/port39
- AJAOne_xxxx/cmd/status/port34
- AJAOne_xxxx/cmd/status/port35
- AJAOne_xxxx/cmd/status/port32
- AJAOne_xxxx/cmd/status/port33
- AJAOne_xxxx/cmd/status/port25
- AJAOne_xxxx/cmd/status/port26

Or can be requested all at once with:

- AJAOne_xxxx/cmd/status

Digital ports 32, 33, 25 and 26 can be used as outputs and could be used to drive relays and such. Send either “on” or “off” in the message payload.

- AJAOne_xxxx/cmd/outout32 {on|off}
- AJAOne_xxxx/cmd/output33 {on|off}
- AJAOne_xxxx/cmd/output25 {on|off}
- AJAOne_xxxx/cmd/output26 {on|off}

Each port can drive up to 40mA at 3.3v.

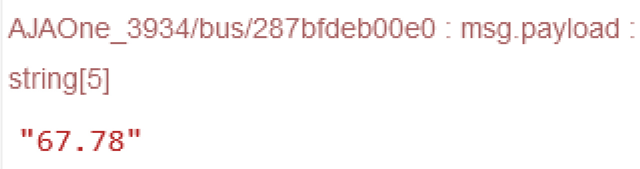
These same ports can also be toggles, on for 250ms and then off:

- AJAOne_xxxx/cmd/toggle32
- AJAOne_xxxx/cmd/toggle33
- AJAOne_xxxx/cmd/toggle25
- AJAOne_xxxx/cmd/toggle26

These can be useful when a quick “key press” is needed, for example to open a garage door.

1-Wire bus

Every 15 seconds the 1-Wire bus will be scanned. If devices are found the data will be read from each one and sent via individual MQTT messages. The message format will be ***<AJAOne Name>/bus/<1-Wire address>*** with the ***msg.payload*** being the data read.



```
AJAOne_3934/bus/287bfdeb00e0 : msg.payload :
string[5]
 "67.78"
```

Figure 1 1-Wire Bus example

OLED Display

An I2C OLED is supported and by default will display the AJAOne board name and IP address on lines zero and one upon boot up. Lines 2 through 5 can be written to via MQTT commands:

- AJAOne_xxxx/cmd/printline2
- AJAOne_xxxx/cmd/printline3
- AJAOne_xxxx/cmd/printline4
- AJAOne_xxxx/cmd/printline5

With the message payload being the text to write to the OLED.

BMP/BME280

If a BMP/BME280 is connected the message will be sent in the following format:

- AJAOne_xxxx/status/temperature
- AJAOne_xxxx/status/humidity
- AJAOne_xxxx/status/pressure (always zero in the case of the BMP280)

BH1750

If a BH1750 light sensor is connected the message will be sent in the following format:

- AJAOne_xxxx/status/light

Reboot

If needed the AJAOne can be rebooted via:

- AJAOne_xxxx/cmd/reboot.

Output Pin Headers

UART

The Universal Asynchronous Receiver Transmitter (UART) is connected to the Serial 2 port of the ESP32. The current firmware does not provide any functionality and is provided for user expansion.

SPI (VSPI)

The Serial Peripheral Interface (SPI) is connected to the VSPI port of the ESP32.

SPI Header		**ESP32 Pin**		**MFRC522 RFID card reader**
MOSI	→	GPIO23	→	MOSI
MISO	→	GPIO19	→	MISO
CLK	→	GPIO18	→	SCK
CS	→	GPIO5	→	SDA
GPIO13	→	GPIO13	→	RST
GPIO27	→	GPIO27	→	N/C
3.3V	→	3.3v bus	→	3.3
GND	→	Ground	→	GND

When initializing the SPI bus, pass the port parameters as follows:

```
SPI.begin(18,19,23,5);
```

I2C

The Inter-Integrated Circuit (I2C) is connected to the I2C port of the ESP32. There is a 4 pin header as well as a STEMMA and a STEMMA-QT JST connectors. All 3 I2C connectors are in parallel to each other and have 10K pull-up resistors on the SCL and SCD lines.

The included firmware supports the following:

- BMP/BME280 at address 0x76
- OLED display at address 0x3C using the SSD1306 chipset.
- BH1750 Light intensity at address 0x23

Other functionality can be added to the firmware.

STEMMA Connectors

I2C devices can also be connected via the STEMMA or STEMMA-QT connectors located on the right side of the board. The large connector is the STEMMA and the smaller one the STEMMA-QT.

STEMMA Pin outs

Top to bottom the pin outs are:

1. Ground
2. Power – 3.3V
3. SDA
4. SCL

STEMMA-QT Pin outs

Top to bottom the pin outs are:

1. SCL
2. SDA
3. Power – 3.3V
4. Ground

Technical

Buttons

Reset

As the name implies pressing the Reset button will reset the ESP32 and reboot it.

D0

In order to reset the AJAOne to factory defaults hold down the ***D0*** button for approximately 10 seconds until the LED turns a solid red. Wait for the LED to go out and then follow the "***Initial startup***" procedures.

Programming

Although the AJAOne can be used as is, it is also open for modification and expanded functionality. With the appropriate knowledge of C++ and the ESP32 in conjunction with the necessary hardware to program the AJAOne, extended functionality can be added

A standard ISP header is available and uses 5V logic. Please do not connect 5V power and 5v from the programmer at the same time.   Disconnect the power before plugging in the programmer.

Holding the ***D0*** button down and pressing the Reset button will put the ESP32 into "download mode". Cycle the power after programming.

OTA is enabled

Official software and Eagle schematics are available here on GITHub.

Sensors used

Waterproof Digital Temperature Temp Sensor Probe DS18b20

BME280 Temperature Humidity Barometric Pressure Sensor Module

0.96 Inch OLED Module 12864 128x64 Yellow Blue SSD1306

BH1750 BH1750FVI Light Intensity Illumination

Mifare RC522 RF IC Card Sensor Module