**ESP8266 microchip-based Arrowhead Adaptor module**

v0.1

1. **What is an ESP8266?**

ESP is a low-cost microchip [1] [2], which is based on a Wi-Fi module combined with a 32-bit processor (Tensilica L106 32-bit microcontroller). The Wi-Fi part of the chip supports a full TCP/IP stack (only IPv4 support), and the libraries offers easy-to-use HTTP or HTTPs communications.

The maximum operation frequency is 160 MHz and the operating voltage is ~3.3V (80mA average operating current).

The package size is 5 mm x 5 mm (QFN32-pin), which offers several interfaces for sensor communication, e.g.: UART, SDIO, SPI, I2C, I2S.

Figure 1 represents some ESP8266-based development boards [3][4][5][6].

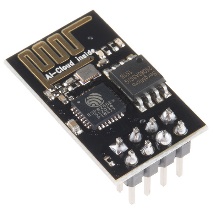
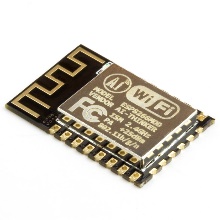
  

Figure 1 – ESP8266 Development Boards

1. **A simple sensor adaptor for ESP8266-based communication**

Since the ESP8266 chip supports several interfaces for sensor communication and the built-in HTTP support enables messaging to the Arrowhead Core System modules, it could be a perfect choice for a small adapter module. Using this adaptor, we could transform our sensor as an Arrowhead capable service.

Figure 2 represents a DHT11 Temperature sensor as an Arrowhead service, using the ESP8266-based adapter.

After power up the ESP module registers a predefined service into the Service Registry module and reads periodically the measured value from the sensor. After a successful registration the adaptor serves consumer requests with the latest measured value, packed in SenML [6] format.

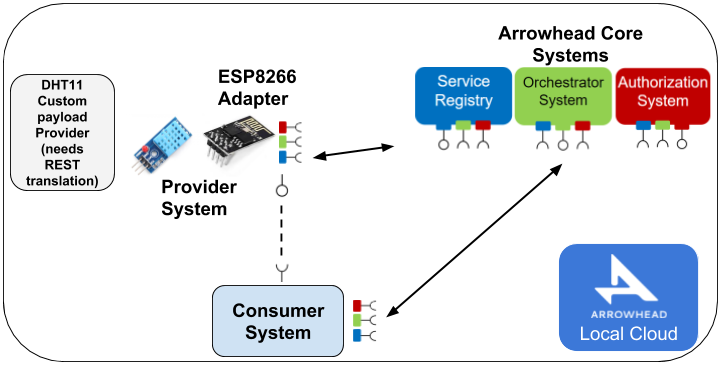


Figure 2 – Adapting a DHT11-Temperature sensor as an Arrowhead Service

1. **Installation guide**

The ESP8266 firmware is based on a C/C++ implementation, and we are using the Arduino SDK for compilation and uploading.

Installation steps:

1. Install the Arduino SDK: <https://www.arduino.cc/en/main/software>
2. Connect the dev-board to the computer
3. Download the ESP8266-Adapter project from github:   
   <https://github.com/arrowhead-f/adapter-cpp>
4. Add the following URL to the File/Preferences/Additional Boards Manager URLs section: <http://arduino.esp8266.com/stable/package_esp8266com_index.json>
5. Install the following libraries using the Tools/Manage libraries menu:
   1. Tools/Board/Boards Manager, install esp8266
   2. DHT sensor library for ESPx by beegee\_tokyo v1.0.9
   3. NTPClient by Fabrice Weinberg v3.1.0
   4. ArduinoJson by Benoit Blanchon v5.13.1
   5. <https://github.com/me-no-dev/ESPAsyncWebServer> download zip and add library to the sketch by Sketch/Include Library/Add .ZIP Library
   6. <https://github.com/me-no-dev/ESPAsyncTCP> download zip and add library to the sketch by Sketch/Include Library/Add .ZIP Library
6. Select the proper module settings:
   1. Tools/Board: Generic ESP8266 Module
   2. Upload Speed: 115200
   3. CPU Frequency: 80 MHz
   4. Flash Frequency: 40 MHz
7. Set the Wi-Fi connection parameters (ssid and password strings in the source code)
8. Compile and Upload the source code using Sketch/Upload
9. Open a serial monitor to check the debug messages, or just check the registered services in the Service Registry module.
10. **References**

[1] <https://www.espressif.com/en/products/hardware/esp8266ex/overview>

[2] <https://www.espressif.com/sites/default/files/documentation/esp8266_hardware_design_guidelines_en.pdf>

[3] <https://www.espressif.com/sites/default/files/documentation/ESP8266-DevKitC_getting_started_guide__EN.pdf>

[4] <http://www.microchip.ua/wireless/esp01.pdf>

[5] <http://wiki.ai-thinker.com/_media/esp8266/esp8266_series_modules_user_manual_v1.1.pdf>

[6] <https://components101.com/sites/default/files/component_datasheet/ESP8266%20Datasheet.pdf>

[3] <https://tools.ietf.org/html/rfc8428>