

**embedis : a dictionary server
for embedded systems**

**Tom Moxon / David Turnbull
@PatternAgents**

What is embedis?

- **An Open Source C/C++ Library for creating key/value dictionaries in persistent storage**
- **Designed to be small, simple, and extensible**
- **Uses familiar Set/Get/Del command syntax**
- **Includes Pub/Sub commands**
- **Includes Hardware Read/Write commands**
- **Similar to redis.io (Perl/Python/Ruby bindings)**
- **<https://github.com/thingSoC/embedis/>**

Why embedis?

- **We needed to support a number of different persistent memory types with different interfaces and access methods, such as FRAM, NVSRAM, EEPROM, BB-SRAM, etc.**
- **We wanted a simple, consistent, command line interface (CLI) and APIs that would work with any of those persistent memory types**
- **We wanted to be able to reuse existing memory driver libraries for the Arduino IDE**

embedis : separate data from code

- **Embedis enables users to separate program code and data (meta-data) in a simple and easy way**
- **Properties such as device serial number, make, model, or other operating parameters like SSID, Gateway Address, etc. are stored independently from the device firmware (program) itself**
- **Embedis provides a simple, portable method for changing device configuration and provisioning data without modifying the device firmware image itself (tested/trusted firmware images)**

embedis : defining dictionaries

AVR Platform (Internal EEPROM available) :

```
Embedis::dictionary(  
    "EEPROM",  
    EEPROM_SIZE,  
    [](size_t pos) -> char { return EEPROM.read(pos); },  
    [](size_t pos, char value) { EEPROM.write(pos, value); }  
);
```



ESP8266 Platform (NO- Internal EEPROM available, emulated) :

```
Embedis::dictionary(  
    "EEPROM",  
    EEPROM_SIZE,  
    [](size_t pos) -> char { return EEPROM.read(pos); },  
    [](size_t pos, char value) { EEPROM.write(pos, value); },  
    []() { EEPROM.commit(); }  
);
```



Note : ESP8266 needs a "commit" to flush/write

embedis : defining dictionaries

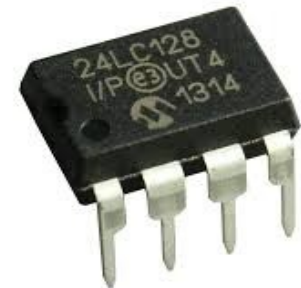
Due Platform (NO- Internal EEPROM available, emulated):

```
Embedis::dictionary(  
  "FLASH",  
  FLASH_EEPROM_SIZE,  
  [](size_t pos) -> char { return FlashDue.read(pos); },  
  [](size_t pos, char value) { FlashDue.write(pos, value); },  
  []() { }  
);
```



External I2C EEPROM device :

```
Embedis::dictionary(  
  "I2C_EEPROM",  
  I2C_EEPROM_SIZE,  
  [](size_t pos) -> char { return ext_i2c_eeprom.read(pos); },  
  [](size_t pos, char value) { ext_i2c_eeprom.write(pos, value); },  
  []() { }  
);
```



embedis : using dictionaries


```
String setting_led_pin() {  
    String led_pin;  
    if (!Embedis::get("led_pin", led_pin)) {  
        led_pin = String(F("13"));  
    }  
    led_pin.toLowerCase();  
    return led_pin;  
}  
  
String setting_sta_ssid() {  
    String sta_ssid;  
    if (!Embedis::get("wifi_ssid", sta_ssid)) {  
        // leave empty string...  
    }  
    return sta_ssid;  
}
```

Embedis : command line

COM53 (Arduino Due (Programming Port))

commands Send

```
[ Embedis Flip-n-Click (Arduino Due) Sketch ]  
[ Found SPI_FRAM ]  
[ Found I2C_EEPROM ]  
[ type 'commands' to get a list... ]  
*14  
+COMMANDS  
+PUBLISH  
+SUBSCRIBE  
+UNSUBSCRIBE  
+DICTIONARIES  
+GET  
+SET  
+DEL  
+KEYS  
+SELECT  
+pinMode  
+digitalWrite  
+digitalRead  
+analogRead
```



Both NL & CR 115200 baud

Autoscroll

Embedis : select dictionary

COM53 (Arduino Due (Programming Port))

select FLASH| Send

```
+COMMANDS
+PUBLISH
+SUBSCRIBE
+UNSUBSCRIBE
+DICTIONARIES
+GET
+SET
+DEL
+KEYS
+SELECT
+pinMode
+digitalWrite
+digitalRead
+analogRead
*3
+FLASH
+SPI_FRAM
+I2_EEPROM
+OK free = 1022
```



Both NL & CR | 115200 baud

Autoscroll

The image shows a screenshot of a serial terminal window titled 'COM53 (Arduino Due (Programming Port))'. The input field contains 'select FLASH|' and a 'Send' button is to its right. The terminal output displays a list of commands and their responses, including '+COMMANDS', '+PUBLISH', '+SUBSCRIBE', '+UNSUBSCRIBE', '+DICTIONARIES', '+GET', '+SET', '+DEL', '+KEYS', '+SELECT', '+pinMode', '+digitalWrite', '+digitalRead', '+analogRead', '*3', '+FLASH', '+SPI_FRAM', '+I2_EEPROM', and '+OK free = 1022'. Below the text is a photograph of two Arduino Due boards. The left board is labeled 'Air quality click' and has a black sensor module on top. The right board is labeled 'Methane click' and has a black sensor module on top. Both boards are connected to a central USB cable. At the bottom of the terminal window, there are settings for 'Both NL & CR' and '115200 baud', and a checked 'Autoscroll' checkbox.

https://www.hackster.io/thingsoc

Many examples are ready at the hackster.io site :

The screenshot shows the Hackster.io website for the 'thingsoc' project. The page features a header with navigation links (Projects, Platforms, Challenges, Live) and a search bar. Below the header is a large image of a blue PCB with various components. The main content area is divided into sections: 'ABOUT THINGSOC' with a description of the open-source standard, 'THINGSOC PRODUCTS' featuring the 'Embedis' logo and a 'thingSoC NeoPixel Driver Embedded M...', and 'JOIN THE COMMUNITY' with social media icons. The footer contains links for 'More cool stuff', 'Legal things', 'About us', and 'We're fairly social people', along with the copyright notice 'Hackster, Inc. 2016'.

Embedis MQTT example

IO - Adafruit Edit project - Hackster.io

https://io.adafruit.com/patternagents/embedis-mqtt-aio

embedis_mqtt_aio

adc0

779
ADC_0

LAST UPDATED JUL 6, 2016 15:57

led0

ON

mqtt_embedis

more arbitrary datadata

pwm0

230

CONTACT
SUPPORT
DISTRIBUTORS
EDUCATORS
JOBS
FAQ
SHIPPING & RETURNS

"Achieving a goal provides immediate satisfaction; the process of achieving a goal is a lasting pleasure"
- Evelyn Berezin

embedis and DeviceTree

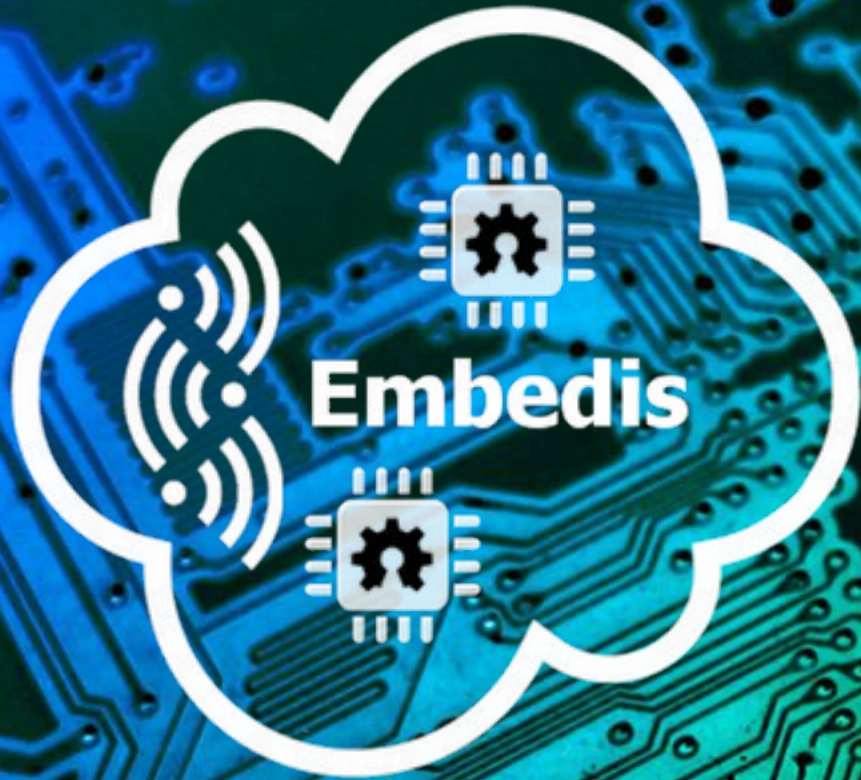
- **DeviceTree is used on Linux platforms like the Raspberry Pi and the Beaglebone Black to hold hardware configuration data and meta-data**
- **DeviceTree starts at low memory and goes up**
- **Embedis starts at high memory and goes down**
- **Embedis and DeviceTree can coexist in the same EEPROM/FLASH memory device**

embedis : Possible Future Plans

- **Dictionary to Dictionary Copy & Backup
(migrate settings to different platforms)**
- **Pub/Sub Feature Completion
(MQTT Integration)**
- **Device Tree Integration
(Present the Device Tree as a Dictionary)**
- **SDcard and eMMS Support
(multiple dictionaries as files)**

How can you help?

- **Comments and Feature Suggestions**
How are you using embedis?
What features would make it more useful?
- **We are looking for more programmers, so, if you have time and would like to help, please let us know...**



Thank You!

**This work
wouldn't
be possible
without you!**

**Tom Moxon / David Turnbull
@PatternAgents**