



DesignNews

Designing Embedded Systems using the ESP32

DAY 2 : Setting up and Exploring the SDK

Sponsored by



Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Group Chat’ by maximizing the chat widget in your dock.
- Submit questions for the lecturer using the Q&A widget. They will follow-up after the lecture portion concludes.

Course Sessions

- Introduction to the ESP32 Wi-Fi Module
- **Setting up and Exploring the SDK**
- Programming and Writing the First Application
- It's all about Wi-Fi
- Jump-Starting Cloud Connectivity Applications with Amazon FreeRTOS

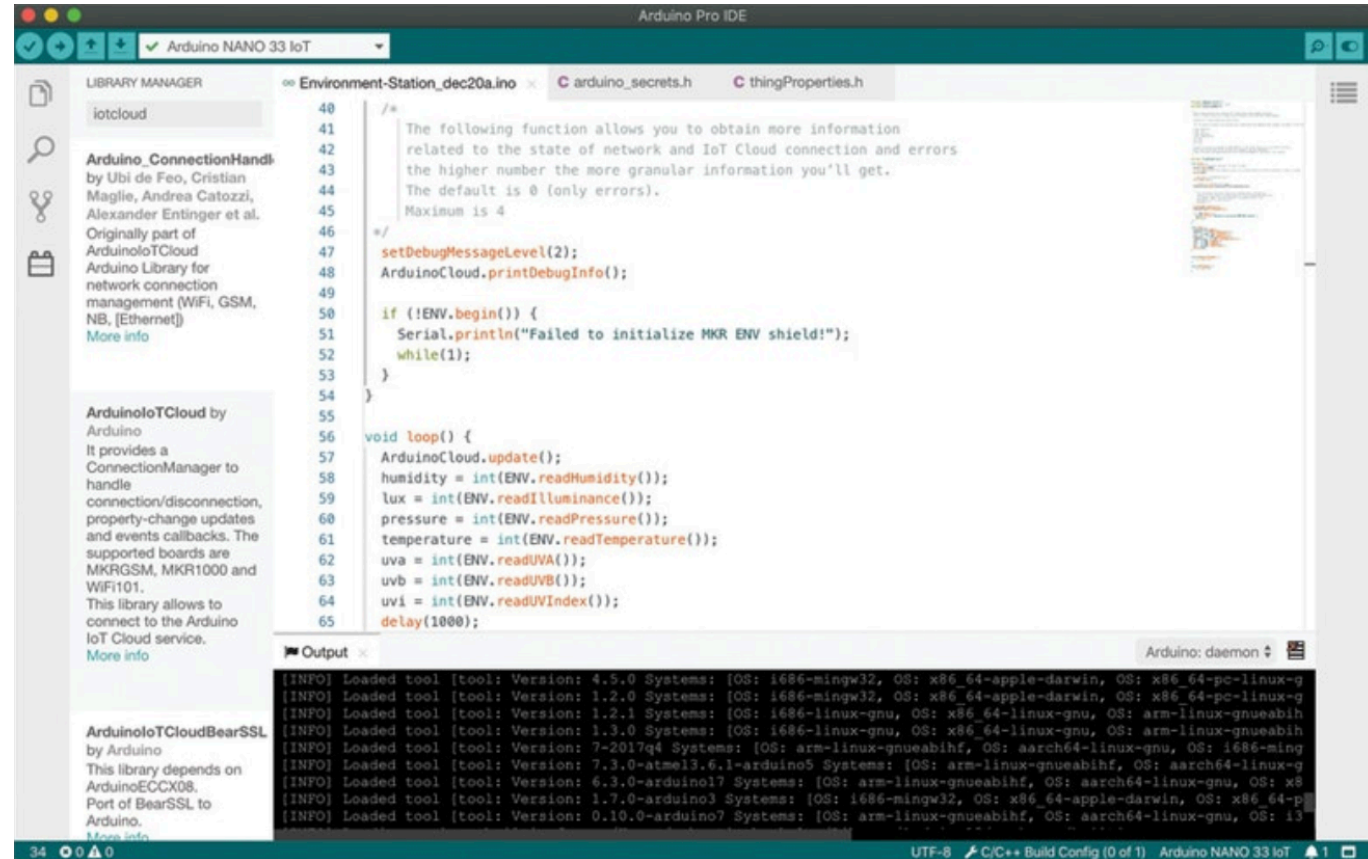
Arduino IDE

Pros

- Fast to setup
- Free
- Plenty of examples

Cons

- Slow compile times
- Rudimentary toolchain



```
Environment-Station_dec20a.ino
C arduino_secrets.h
C thingProperties.h

40
41
42 /*
43 The following function allows you to obtain more information
44 related to the state of network and IoT Cloud connection and errors
45 the higher number the more granular information you'll get.
46 The default is 0 (only errors).
47 Maximum is 4
48 */
49
50 setDebugMessageLevel(2);
51 ArduinoCloud.printDebugInfo();
52
53 if (!ENV.begin()) {
54   Serial.println("Failed to initialize MKR ENV shield!");
55   while(1);
56 }
57
58 void loop() {
59   ArduinoCloud.update();
60   humidity = int(ENV.readHumidity());
61   lux = int(ENV.readIlluminance());
62   pressure = int(ENV.readPressure());
63   temperature = int(ENV.readTemperature());
64   uva = int(ENV.readUVA());
65   uvb = int(ENV.readUVB());
66   uvi = int(ENV.readUVIndex());
67   delay(1000);
68 }
```

Output

```
Arduino: daemon
[INFO] Loaded tool [tool: Version: 4.5.0 Systems: [OS: i686-mingw32, OS: x86_64-apple-darwin, OS: x86_64-pc-linux-g
[INFO] Loaded tool [tool: Version: 1.2.0 Systems: [OS: i686-mingw32, OS: x86_64-apple-darwin, OS: x86_64-pc-linux-g
[INFO] Loaded tool [tool: Version: 1.2.1 Systems: [OS: i686-linux-gnu, OS: x86_64-linux-gnu, OS: arm-linux-gnueabi
[INFO] Loaded tool [tool: Version: 1.3.0 Systems: [OS: i686-linux-gnu, OS: x86_64-linux-gnu, OS: arm-linux-gnueabi
[INFO] Loaded tool [tool: Version: 7-2017q4 Systems: [OS: arm-linux-gnueabi, OS: aarch64-linux-gnu, OS: i686-ming
[INFO] Loaded tool [tool: Version: 7.3.0-atmel3.6.1-arduino5 Systems: [OS: arm-linux-gnueabi, OS: aarch64-linux-g
[INFO] Loaded tool [tool: Version: 6.3.0-arduino17 Systems: [OS: arm-linux-gnueabi, OS: aarch64-linux-gnu, OS: x8
[INFO] Loaded tool [tool: Version: 1.7.0-arduino3 Systems: [OS: i686-mingw32, OS: x86_64-apple-darwin, OS: x86_64-p
[INFO] Loaded tool [tool: Version: 0.10.0-arduino7 Systems: [OS: arm-linux-gnueabi, OS: aarch64-linux-gnu, OS: i3
```

MicroPython

Pros

- Fast to setup
- Free
- Quickly write apps in Python
- No low-level languages

Cons

- MicroPython is ~400 kB
- Features are not all supported

Libraries and external
classes

```
# main.py  
import pyb
```

Definitions and
Initialization

```
# define LED color constants  
LED_RED = 1  
LED_GREEN = 2  
LED_YELLOW = 3  
LED_BLUE = 4
```

```
# Defines the primary loop delay  
DELAY_1000MS = 1000
```

```
# Create an Led object assigned to the green LED  
Led = pyb.LED(LED_GREEN)
```

Primary program loop

```
# Main execution loop  
# Toggle the LED every  
while True:  
    Led.toggle()  
    pyb.delay(DELAY_1000MS)
```

Amazon FreeRTOS

Pros

- Easy cloud connectivity
- Flexible dev environments
- Widely supported

Cons

- ESP-IDF synchronization
- Growing code size

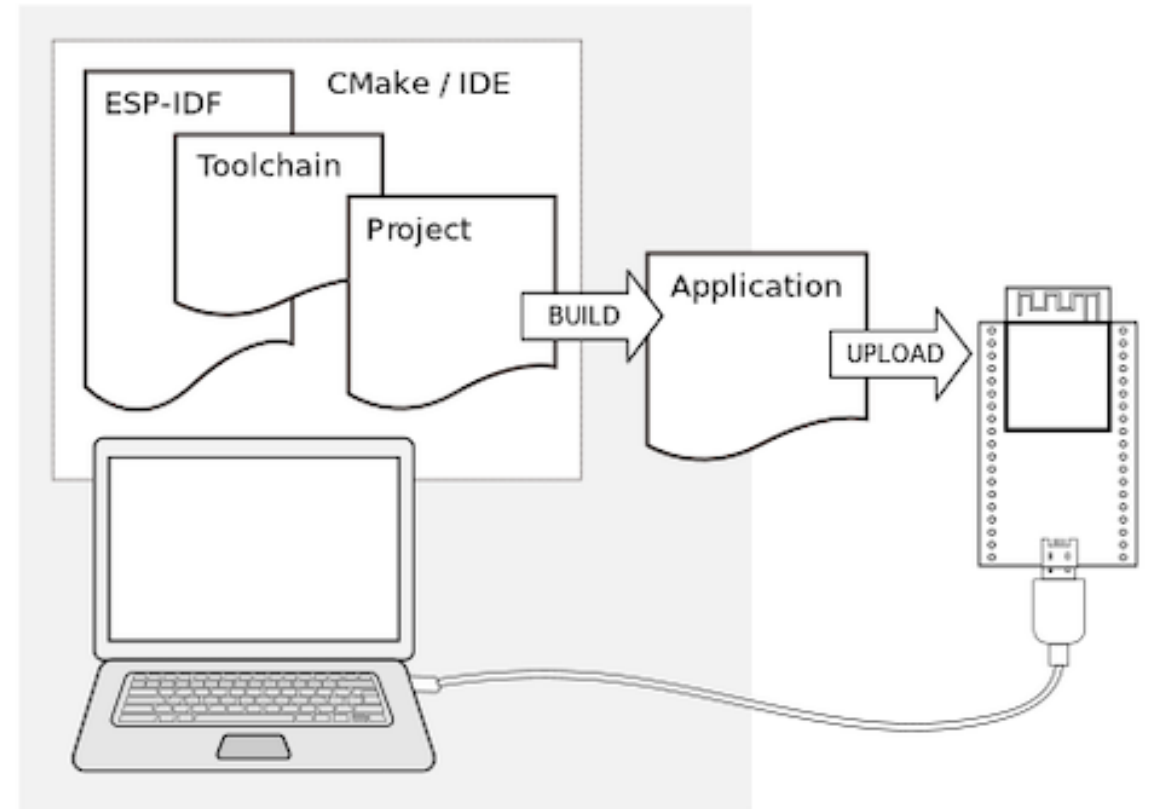
ESP-IDF

Pros

- Comprehensive device support
- Professional developer focused
- Integrated framework

Cons

- Can be “complex” to setup

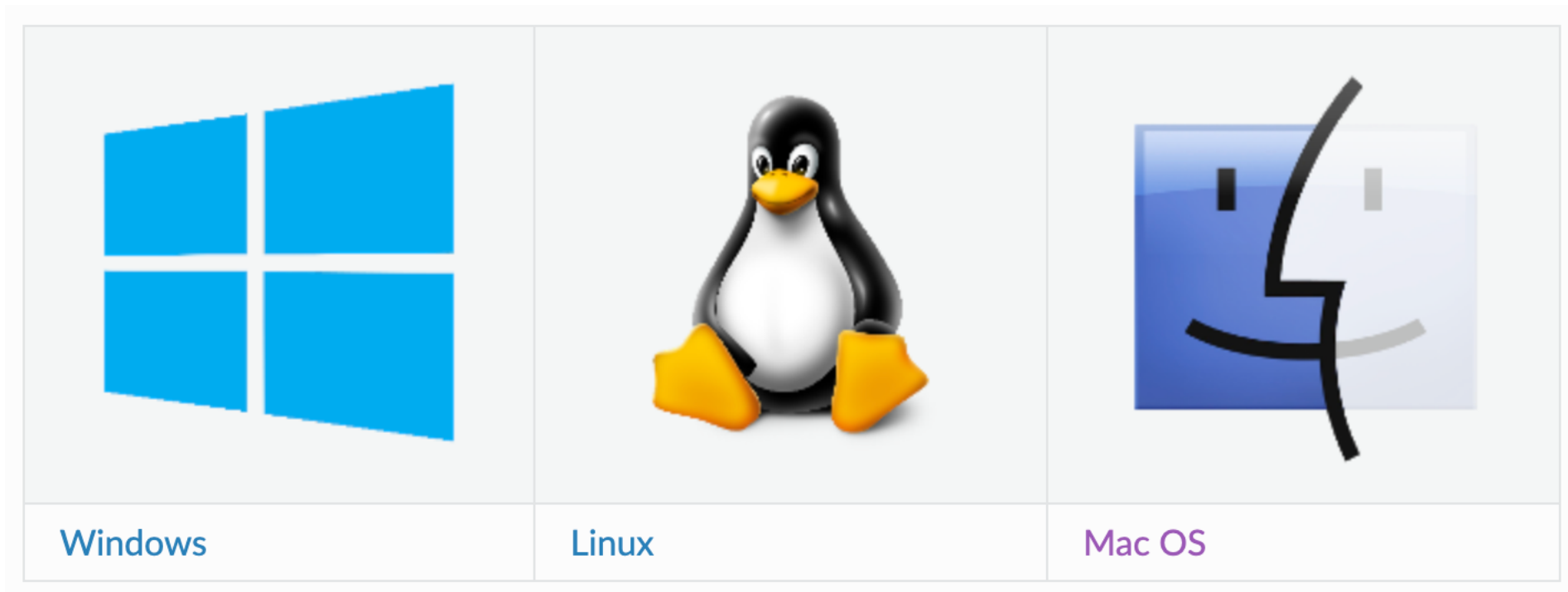


Which development environment are you most interested in trying?

- Arduino
- MicroPython
- Amazon FreeRTOS
- ESP-IDF

ESP-IDF Setup

1) Select your OS



ESP-IDF Setup

2) Get ESP-IDF

```
mkdir -p ~/esp
```

```
cd ~/esp
```

```
git clone --recursive https://github.com/espressif/esp-idf.git
```

ESP-IDF Setup

3) Set up the tools

```
cd ~/esp/esp-idf  
./install.sh
```

```
beningo@Jacobs-MacBook-Pro esp-idf % ./install.sh  
Installing ESP-IDF tools  
Installing tools: xtensa-esp32-elf, xtensa-esp32s2-elf, esp32ulp-elf, esp32s2ulp-elf, openocd-esp32  
Skipping xtensa-esp32-elf@esp-2020r3-8.4.0 (already installed)  
Skipping xtensa-esp32s2-elf@esp-2020r3-8.4.0 (already installed)  
Skipping esp32ulp-elf@2.28.51-esp-20191205 (already installed)  
Skipping esp32s2ulp-elf@2.28.51-esp-20191205 (already installed)  
Skipping openocd-esp32@v0.10.0-esp32-20200709 (already installed)  
Installing Python environment and packages  
Creating a new Python environment in /Users/benigo/.espressif/python_env/idf4.2_py3.9_env  
Installing virtualenv  
Collecting virtualenv  
  Using cached virtualenv-20.2.2-py2.py3-none-any.whl (5.7 MB)  
Collecting appdirs<2,>=1.4.3  
  Using cached appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)  
Collecting distlib<1,>=0.3.1  
  Using cached distlib-0.3.1-py2.py3-none-any.whl (335 kB)  
Collecting filelock<4,>=3.0.0  
  Downloading filelock-3.0.12-py3-none-any.whl (7.6 kB)  
Collecting six<2,>=1.9.0  
  Using cached six-1.15.0-py2.py3-none-any.whl (10 kB)  
Installing collected packages: six, filelock, distlib, appdirs, virtualenv  
WARNING: The script virtualenv is installed in '/Users/benigo/Library/Python/3.9/bin' which is not on PATH.  
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-loc  
Successfully installed appdirs-1.4.4 distlib-0.3.1 filelock-3.0.12 six-1.15.0 virtualenv-20.2.2  
created virtual environment CPython3.9.0.final.0-64 in 554ms  
creator CPython3Posix(dest=/Users/benigo/.espressif/python_env/idf4.2_py3.9_env, clear=False, no_vcs_ignore_re  
seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/User  
added seed packages: pip==20.3.1, setuptools==51.0.0, wheel==0.36.1  
activators BashActivator,CShellActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator  
Installing Python packages from /Users/benigo/esp/esp-idf/requirements.txt  
Ignoring None: markers 'sys_platform == "win32"' don't match your environment  
Requirement already satisfied: setuptools>=21 in /Users/benigo/.espressif/python_env/idf4.2_py3.9_env/lib/p  
nts.txt (line 4) (51.0.0)  
Collecting bitstring>=3.1.6  
  Using cached bitstring-3.1.7.tar.gz (195 kB)  
Collecting click>=5.0  
  Using cached click-7.1.2-py2.py3-none-any.whl (82 kB)  
Collecting cryptography>=2.1.4  
  Downloading cryptography-3.3.1-cp36-abi3-macosx_10_10_x86_64.whl (1.8 MB)  
  |████████████████████████████████████████| 1.8 MB 2.3 MB/s  
Collecting ecdsa>=0.16.0  
  Using cached ecdsa-0.16.1-py2.py3-none-any.whl (104 kB)  
Collecting future>=0.15.2  
  Using cached future-0.18.2.tar.gz (829 kB)  
Collecting gdbgui==0.13.2.0  
  Downloading gdbgui-0.13.2.0-py3-none-any.whl (878 kB)  
  |████████████████████████████████████████| 878 kB 13.5 MB/s  
Collecting pyelftools>=0.22
```

ESP-IDF Setup

Package Issues:

You need to use Python 3. On MacOS, execute:

```
brew install python
```

```
ls -l /usr/local/bin/python*
```

```
ln -s -f /usr/local/bin/python3.9 /usr/local/bin/python
```

```
python --version
```

ESP-IDF Setup

4) Set up environment variables

```
. $HOME/esp/esp-idf/export.sh
```

You can also add to your terminal profile:

```
alias get_idf='. $HOME/esp/esp-idf/export.sh'
```

Did you follow along and install the toolchain?

- Yes
- No
- No but I plan to for tomorrow

ESP-IDF Overview

- Bluetooth
- Networking
- Peripheral
- Protocols
- Provisioning
- Storage
- System
- Configuration

ESP-IDF UART API

```
const int uart_num = UART2; uart_config_t uart_config =  
{ .baud_rate = 115200,  
  .data_bits = UART_DATA_8_BITS,  
  .parity = UART_PARITY_DISABLE,  
  .stop_bits = UART_STOP_BITS_1,  
  .flow_ctrl = UART_HW_FLOWCTRL_CTS_RTS,  
  .rx_flow_ctrl_thresh = 122, };
```

```
// Configure UART parameters ESP_ERROR_CHECK(uart_param_config(uart_num, &uart_config));
```

```
// Set UART pins(TX: IO16 (UART2 default), RX: IO17 (UART2 default), RTS: IO18, CTS: IO19)  
ESP_ERROR_CHECK(uart_set_pin(UART_NUM_2, UART_PIN_NO_CHANGE, UART_PIN_NO_CHANGE, 18, 19));
```


ESP-IDF UART Example

Parameter to Configure	Function
Baud rate	<code>uart_set_baudrate()</code>
Number of transmitted bits	<code>uart_set_word_length()</code> selected out of <code>uart_word_length_t</code>
Parity control	<code>uart_set_parity()</code> selected out of <code>uart_parity_t</code>
Number of stop bits	<code>uart_set_stop_bits()</code> selected out of <code>uart_stop_bits_t</code>
Hardware flow control mode	<code>uart_set_hw_flow_ctrl()</code> selected out of <code>uart_hw_flowcontrol_t</code>
Communication mode	<code>uart_set_mode()</code> selected out of <code>uart_mode_t</code>

ESP-IDF UART Example

```
// Setup UART buffered IO with event queue
```

```
const int uart_buffer_size = (1024 * 2);
```

```
QueueHandle_t uart_queue; // Install UART driver using an event queue here
```

```
ESP_ERROR_CHECK(uart_driver_install(UART2, uart_buffer_size, \uart_buffer_size, 10, &uart_queue, 0));
```

```
// Write data to UART.
```

```
char* test_str = "This is a test string.\n";
```

```
uart_write_bytes(uart_num, (const char*)test_str, strlen(test_str));
```

Do you plan to follow along live for the first application development tomorrow?

- Yes
- No

Thank you for attending

Please consider the resources below:

- www.beningo.com
 - Blog, White Papers, Courses
 - Embedded Bytes Newsletter
 - <http://bit.ly/1BAHYXm>

From www.beningo.com under

- Blog > CEC – Designing Embedded Systems using the ESP32





DesignNews

Thank You

Sponsored by

