# ESP32 開発環境

2020.03.03



#### <u>Windows10 版</u>



## 手順:

- •1. USBからArduinoファイルをデスクトップへコピーします。
- 2. そのファイルを展開します。そのファイルの中の
   ◎
- •3. セキュリティ警告画面が出るので許可します。
- 4. Arduino IDEが立ち上げるので、メニューから
   「ファイル」→「環境設定」を開き、
- 5.「追加のボードマネージャのURLは、 https://dl.espressif.com/dl/package\_esp32\_index.json と入力し「OK」をクリックします。



### 手順:

- 6.「ツール」→「ボード」→「ボードマネージャ」を開きます。
- 7. 「esp32」と入力し、「esp by Espressif Systems」 をインストールします。最新版は、1.0.5 です。
- 8. 「ツール」→「ボード」で、「DOIT ESP32 DEVKIT V1」 を選択します。
- これでESP32をArduino IDEを利用する準備ができました。 パソコンとESP32ボードをUSBケーブルで接続します。 今回は、ポートCOM3に認識されました。



## <u>Arduino 初期画面</u>

S IOTTest   Arduino 1.8.12	X
ファイル 編集 スケッチ ツール ヘルプ	
	20 - 20 - 20 - 20 - 20 - 20 - 20 - 20 -
IOTTest	
	^
int led = 23;	
<pre>void setup() {</pre>	
<pre>pinMode(led, OUTPUT);</pre>	
}	
void loop() {	
digitalWrite(led, HIGH);	
digitalWrite(led IOW):	
delav(1000):	
}	
	~
1	COM3のDOIT ESP32 DEVKIT V1, 80MHz, 115200, None



#### 赤波線のところに次ページの 🥯 URLを入力する。

💿 IOTTest   Arduino 1.8.12			- 🗆	$\times$
ファイル 編集 スケッチ ツール	ヘルプ			
				<b>1</b> 20-
IOTTest				
	環境設定 ×			^
	設定 ネットワーク			
int led = 23:	スケッチブックの保存場所:			
<pre>void setup(){</pre>	C.¥Users¥fakem¥Documents¥Arduino 参照			
pinMode(led,	言語設定: System Default v 変更の反映にはArduino IDEの再起動が必要			
- · · ·	エディタの文字の大きさ: 18			
}	インタフェースのスケール: ✓ 自動 100 € % 変更の反映にはArduino IDEの 再起動が必要			
<pre>void loop(){</pre>	テーマ: デフォルトのテーマ V 変更の反映にはArduino IDEの再起動が必要			
digitalWrite				
delay(1000);	□ 行番号を表示する □ コードの折り返しを有効に			
digitalWrite	2 書き込みを検証する 2 書き込みを検証する 1 外部のエディタを使用する			
delay(1000);	✓ 起動時に最新パージョンの有無をチェックする ✓ 検証または書き込みを行う前にスケッチを保存する			
}	Use accessibility features			
	追加のボーマネージャのURL: https://dl.espressif.com/dl/package.esp32.index.json			
	以下のJアイルを直接編集 9 AIは、よい少くの設定を行うことができま 9。 C:¥Users¥fakem¥AppData¥Local¥Arduino15¥preferences.txt			
	編集する際には、Arduino IDEを終了させておいてください。			
	OK キャンセル			
				J
1		COM30DOIT ESP32 DEVKIT	V1, 80MHz, 115200,	None



## <u>ライブラリー先URL</u>

#### 追加ボードマネージャのURL

#### https://dl.espressif.com/dl/package\_esp32\_index.json



## 開発環境・続き

環境設定		×
設定 ネットワーク		
スケッチブックの 保存	場所:	
C:¥Users¥fakem¥E	Documents¥Arduino	参照
言語設定:	System Default 🗸 🗸 🗸	変更の反映にはArduino IDEの再起動が必要
エディタの文字の大	💿 追加のボードマネージャのURL	×
インタフェースのスケ	追加のURLを1行ずつ入力	
テーマ: より詳細な情報を コンパイラの警告:	https://dl.espressif.com/dl/package_esp32_index	(.json
□ 行番号を表示	カルのリア非公式ボードなせポートするロロのリントを表示	
✓ 日日△の行使日 ✓ 起動時に最新 Use accessibin	ity reactices	OK キャンセル る
追加のボードマネー	ジャのURL: https://dl.espressif.com/dl/package_esp32_index.json	
以下のファイルを直接	接編集すれば、より多くの設定を行うことができます。	
C:¥Users¥fakem¥A	ppData¥Local¥Arduino15¥preferences.txt	
編集する際には、A	rduino IDEを終了させておいてください。	
		OK キャンセル



#### <u>アクセス許可を求められる。</u>



許可してください。



## <u>追加ボード情報のインストール</u>

💿 sketch_mar01a   Arduino	1.8.12	
ファイル 編集 スケッチ ツール ^	ルプ	
sketch_mar01a		
<pre>void setup()</pre>	{	
// put you	setup code here, to run once:	
[	छ	×
}	- タイプ 全て	
	esp32	^
<pre>void loop()</pre>	by Espressif Systems このパッケージに含まれているボード:	
// put you:	ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32. More Info	
	1.0.4 〜 インストール	
}		- 11
<b>ナノニノゴのシリフルギ</b>		
ネイティブのシリアルホー		
		×
	[a]	Va

タイプ欄に「esp32」と入力する。





### DOIT ESP32 DEVKIT V1を選択。

🔤   🗔 🍤	e	💿 IOTTest   Arduino 1	1.8.12			- [		×
ファイル	<b>π−</b> Δ	ファイル 編集 スケッチ ツ	ール ヘルプ					
	切り		自動整形	Ctrl+T				<b>.</b>
	שצ-		スグッチをアーカイフする					-
貼り付け		IOTTest	エノコーナイノクを1回止 ライブラリ友管理	Ctrl+Shift+I				
			シリアルモニタ	Ctrl+Shift+M				^
クリッノホ	1-1		シリアルプロッタ	Ctrl+Shift+L				
		<pre>int led = :</pre>	WiFi101 / WiFiNINA Firmware Updat	er	ボードマネージャ			
		void setup			<b>A</b>			
		pinMode(	The second secon	2	Widora AIR			
			Flash Frequency: "80MHz"	]	Electronic SweetPeas - ESP320			
		}	Core Debug Level: "なし"	>	Nano32			
		void loop(	シリアルボート	>	LOLIN D32			
		digitalW	ボード情報を取得		LOLIN D32 PRO			
		digitalw.	書込装置: "AVRISP mkll"	>	Dongsen Tech Pocket 32			
		delay(10	ブートローダを書き込む		"WeMos" WiFi&Bluetooth Battery			
		digitalW <sub>E</sub>	ICE(IEG, LOW),		ESPea32			
		delav(1000	0);		Noduino Quantum			~
		ボードへの書き込みが	完了しました。		Node32s			
		writing at u	XUUU18UUU (50 %)		Hornbill ESP32 Dev			^
		Writing at C	)x0001c000 (66 %)		Hornbill ESP32 Minima			
		Writing at C	)x00020000 (83 %)		IntoRobot Fig			
		Writing at 0	)x00024000 (100 %)		Onehorse ESP32 Dev Module			
		Wrote 192208	B bytes (95757 compre	ssed) at	Adafruit ESP32 Feather	onds (effective 179.2 kbit/s)		
		Hash of data	verified		NodeMCU-32S			
			2072 bytes to $144$		MH ET LIVE ESP32DevKIT			
		compressed 3	5072 bytes to 144		MH ET LIVE ESP32MiniKit			
					ESP32vn IoT Uno			
		Writing at O	)x00008000 (100 %)					
		Wrote 3072 b	oytes (144 compressed	) at 0x00	OLIMEX ESP32-GATEWAY	(effective 792.8 kbit/s)		
			a verified.		OLIMEX ESP32-PoE			
					ThaiEasyElec's ESPino32			
		Leaving			M5Stack-Core-ESP32			
		Hard resetti	ing via RTS pin		M5Stack-FIRE			
					ODROID ESP32			
		<			Heltec_WIFI_Kit_32			>
		1			ESPectro32	COM3のDOIT ESP3:	2 DEVKIT	V1



#### <u>アップロードスピードの設定</u>

#### 💿 sketch\_mar01a | Arduino 1.8.12 ファイル 編集 スケッチ ツール ヘルプ 自動整形 Ctrl+T スケッチをアーカイブする sketch\_mar01a エンコーディングを修正 ライブラリを管理... Ctrl+Shift+I void setu シリアルモニタ Ctrl+Shift+M // put シリアルプロッタ Ctrl+Shift+L in once: WiFi101 / WiFiNINA Firmware Updater } ボード: "DOIT ESP32 DEVKIT V1" Upload Speed: "921600" 921600 Flash Frequency: "80MHz" 115200 void loop Core Debug Level: "なし" 256000 230400 tedly: シリアルポート: "COM4" // put ボード情報を取得 512000 書込装置: "AVRISP mkll" } ブートローダを書き込む



115200のスピードで設定。





ファイルド) 操作(A) 表示(M) ヘルブ(H)         ● ● □ □ □ □ ●         ● ● □ □ □ □ ●         ● ● □ □ □ □ □ ●         ● ● □ □ □ □ □ □ □ ●         ● ● □ □ □ □ □ □ □ □ □ □ □ □ ●         ● □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	<b>≞</b> 7	バイス	7ネージャー ―	×
<ul> <li>◆ ◆ I II II II II II II</li> <li>&gt; 3275 ム デバイス</li> <li>II 11 11 11 11 11 11 11 11 11 11 11 11 1</li></ul>	ファイノ	ν(F)	操作(A) 表示(V) ヘルプ(H)	
<ul> <li>&gt; システムデバイス</li> <li>              ◆システムデバイス          </li> <li>             ダンドウェアブパペス         </li> <li>             ダイスクドライブ         </li> <li>             ディスクレイアダプター         </li> <li>             ディスクレイアダプター         </li> <li>             ディスクレイアダプター         </li> <li>             ディスクレイクターフェイスデバイス         </li> <li>             ディスクレイクターフェイスデバイス         </li> <li>             プロセッサ         </li> </ul> <li>             プロセッサ         <ul> <li>             ディト (COM と LPT)             </li> <li>             Silicon Labs CP210x USB to UART Bridge (COM3)         </li> </ul> </li> <li> <ul> <li>             プロセッサ             </li> <li>             プロセッサ             </li> <li>             ディト (COM と LPT)             </li> <li>             Silicon Labs CP210x USB to UART Bridge (COM3)             </li> <li>                   マウスとそのほかのポインティング デバイス                  </li> <li>                  マウスとそのほかのポインティング デバイス</li></ul></li>				
<ul> <li>USB Composite Device</li> <li>USB Composite Device</li> <li>USB Composite Device</li> <li>USB ルート ハブ (USB 3.0)</li> <li>USB 大容量記憶装置</li> <li>汎用 SuperSpeed USB ハブ</li> <li>辺田 USB ハゴ</li> </ul>		<ul> <li>↓</li> <li>↓<th><ul> <li>(*) (*) (*) (*) (*) (*) (*) (*) (*) (*)</li></ul></th><th>^</th></li></ul>	<ul> <li>(*) (*) (*) (*) (*) (*) (*) (*) (*) (*)</li></ul>	^
♥ USB Composite Device ♥ USB ルート ハブ (USB 3.0) ♥ USB 大容量記憶装置 ♥ 汎用 SuperSpeed USB ハプ		Ŷ	USB Composite Device USB Composite Device	
♥ USB 大容量記憶装置 ● 汎用 SuperSpeed USB ハブ ● 汎用 LISP ハブ		Ŷ	USB Composite Device USB ルート ハブ (USB 3.0)	
		Ŷ	USB 大容量記憶装置 汎用 SuperSpeed USB ハブ	~



## <u>シリアルポート確認</u>

🥺 IOTTest   Arduino 1.	8.12		
ファイル 編集 スケッチ <mark>ツ</mark> ー	ルヘルプ		
IOTTest	自動整形 スケッチをアーカイブする エンコーディングを修正 ライブラリを管理	Ctrl+T Ctrl+Shift+l	
	シリアルモニタ	Ctrl+Shift+M	
	シリアルプロック	Ctrl+Shift+I	
<pre>int led =</pre>	WiFi101 / WiFiNINA Firmware Update	r	
<pre>void setup   pinMode( } void loop(   digitalW   delay(10)</pre>	ボード: "DOIT ESP32 DEVKIT V1" Upload Speed: "115200" Flash Frequency: "80MHz" Core Debug Level: "なし" シリアルポート: "COM3" ボード情報を取得 書込装置: "AVRISP mkII"		シリアルポート <b>COM3</b>
digitalW	ブートローダを書き込む		
algitalwii	ce (rea, how),		
delav(1000	);		



#### <u>「ファイル」ー>「スケッチ例」ー>「01</u> <u>Basics」ー>「Blink」を選択する。</u>



実行例: 書き込みが完了すると ESP32ボード上にある 青色LEDが点滅する。



ファイルメニュー

ファイル       編集 スケッチ ツール ヘルプ         新規ファイル       Ctrl+N         開く       Ctrl+O         最近使った項目を開く       >         スケッチブック       >         スケッチブック       >
新規ファイル Ctrl+N 開く Ctrl+O 最近使った項目を開く > スケッチブック >
最近使った項目を開く         >           スケッチブック         >           スケッチブック         >
閉じる Ctrl+W 保存 Ctrl+S
名前を付けて保存 Ctrl+Shift+S
プリンタの設定     Ctrl+Shift+P     」);       印刷     Ctrl+P
環境設定 Ctrl+カンマ
終了 Ctrl+Q
<pre>digitalWrite(led, HIGH);</pre>
delay(1000);
<pre>digitalWrite(led, LOW);</pre>
delay(1000);
}



編集メニュー

🥺 IOTTest | Arduino 1.8.12

ファイル 編集 スケッチ ツール ヘルプ

	元に戻す	Ctrl+Z
		0.1.1/
	やり直し	Ctrl+Y
IOT		C. I. M.
	切り取り	Ctrl+X
	วษ์-	Ctrl+C
	フォーラム投稿形式でコピーする	Ctrl+Shift+C
	HTML形式でコピーする	Ctrl+Alt+C
Int	貼り付け	Ctrl+V
voi	全て選択	Ctrl+A
		C. I. I
p:	指定の行番号へ	Ctrl+L
	コメント化・復帰	Ctrl+スラッシュ
}	インデントを増やす	lab
woi	インデントを減らす	Shift+Tab
VOI		
d:	フォントサイズを大きく	Ctrl+プラス
-1	フォントサイズを小さく	Ctrl+マイナス
a		
d	検索	Ctrl+F
	次を検索	Ctrl+G
de		curro
	町を確安	Ctrl+Shitt+G



<u> ツールメニュー</u>

ファイル 編集 スケッチ ツール ヘルプ <ul> <li> <li> <ul> <li></li></ul></li></li></ul>
IOTTest自動整形Ctrl+TIOTTestスケッチをアーカイブするIOTTestフイブラリを管理Ottl+Shift+IシリアルモニタCtrl+Shift+MシリアルブロッタCtrl+Shift+LWiFi101 / WiFiNINA Firmware Updaterボード: "DOIT ESP32 DEVKIT V1"pinMode (Upload Speed: "115200"Flash Frequency: "80MHz"Core Debug Level: "なし"シリアルボートジリアルボートマoid 100p (digitalWdigitalWdelay (10digitalWTree (Text, Low),delay (1000);
int led = void setup pinMode ( WiFi101 / WiFiNINA Firmware Updater ポード: "DOIT ESP32 DEVKIT V1" Upload Speed: "115200" Flash Frequency: "80MHz" Core Debug Level: "なし" シリアルポート ポード情報を取得 書込装置: "AVRISP mkII" フートローダを書き込む delay (1000); }
void setup pinMode (       ボード: "DOIT ESP32 DEVKIT V1"       >>         Upload Speed: "115200"       >>         Flash Frequency: "80MHz"       >>         Core Debug Level: "なし"       >>         void loop (       ジリアルボート       >>         delay (10       ガートローダを書き込む       >>         delay (1000);       ;       >>
delay(10) digitalWirte(ieu, iow), delay(1000); }
<pre>delay(1000); }</pre>